

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2022 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“ Дослідження та програмна реалізація інформаційно-аналітичної
системи для аналізу великих даних”

Виконав здобувач вищої освіти
II курсу, групи КІ-21М 1,4 _____
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Чабан О. О.
« ____ » _____ 2022 р.

Керівник проекту
доктор технічних наук, професор
_____ Мелешко Є. В.
« ____ » _____ 2022 р.

Рецензент Дреєв О. Н. _____

м. Кропивницький

Центральноукраїнський національний технічний університет
Факультет *Механіко-технологічний*
Кафедра *Кібербезпеки та програмного забезпечення*
Рівень вищої освіти *магістр*
Галузь знань . 12 *“Інформаційні технології”*
Спеціальність *123 “Комп’ютерна інженерія”*
Освітньо-професійна (освітньо-наукова) програма *“Комп’ютерна інженерія”*

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.

Олексій СМІРНОВ

“ ” 2022 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Чабана Олексія Олеговича

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація інформаційно-аналітичної системи для аналізу великих даних*

2. Керівник роботи *Мелешко Єлизавета Владиславівна, доктор технічних наук, професор*

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “17” серпня 2022 року № 19-13

3. Строк подання роботи до захисту *10.12.2022 р.*

4. Мета та завдання випускної кваліфікаційної роботи *Метою роботи є дослідження та програмна реалізація інформаційно-аналітичної системи для аналізу великих даних*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання. 7. Економічна ефективність

2. Перегляд аналогічних існуючих систем. розробленої програми.

3. Опис і обґрунтування проектних рішень. 8. Заходи з охорони праці та техніки

4. Етапи програмування системи. безпеки.

5. Впровадження системи в промислову експлуатацію. 9. Висновки.

6. *Наукова новизна*

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна 1 аркуш

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

Діаграма процесів 1 аркуш

Показники економічної ефективності 1 аркуш

6. Консультанти по роботі, із зазначенням розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В., к.т.н., доцент	25.10.2022	12.11.2022
Охорона праці	Оришака О.В., к.т.н., доцент	04.11.2022	20.11.2022

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів кваліфікаційної бакалаврської роботи	Примітка
1.	Аналіз існуючих систем	10.10.2022 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2022 р.	
3.	Розробка моделі компонента	20.10.2022 р.	
4.	Розробка структур даних	25.10.2022 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2022 р.	
6.	Програмування алгоритмів	10.11.2022 р.	
7.	Розрахунок економічної ефективності	13.11.2022 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2022 р.	
9.	Оформлення ПЗ	19.11.2022 р.	
10.	Попередній захист роботи	10.12.2022 р.	

Дата видачі завдання
«6» вересня 2022 р.

Підпис керівника

_____ (прізвище та ініціали)

Завдання прийнято до виконання
«__» _____ 2022 р.

Підпис здобувача

_____ (прізвище та ініціали)

АНОТАЦІЯ

Чабан О.О. Дослідження та програмна реалізація інформаційно-аналітичної системи для аналізу великих даних. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2022.

В даній магістерській роботі розроблено програмне забезпечення, яке призначено для реалізації інформаційно-аналітичної системи для аналізу великих даних.

Метою розробки є дослідження та програмна реалізація інформаційно-аналітичної системи для аналізу великих даних.

Об'єктом дослідження є процес роботи з великими даними у QlikSense.

Предметом дослідження є методи та алгоритми розробки інформаційно-аналітичного середовища для роботи з великими даними у QlikSense.

Методи дослідження базуються на теорії аналізу даних, теорії статистики теорії об'єктно-орієнтованого програмування, теорії алгоритмів, методах побудови аналітичного інтерфейсу.

Результат роботи – програмна реалізація інформаційно-аналітичної системи для аналізу великих даних.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на будь-яких гаджетах сучасного покоління(Android, IOS, PC).

Програму розроблено в середовищі Qlik.

Ключові слова: комп'ютерна інженерія, аналітичні системи, Big Data.

ABSTRACT

Chaban O.O. Research and software implementation of an information and analytical system for the analysis of big data. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2022.

In this master's thesis, software was developed, which is intended for the implementation of an information-analytical system for the analysis of big data.

The goal of development is the research and software implementation of an information-analytical system for analyzing big data.

The object of research is the process of working with big data in QlikSense.

The subject of the research is methods and algorithms for developing an information and analytical environment for working with big data in QlikSense.

The research methods are based on the theory of data analysis, the theory of statistics, the theory of object-oriented programming, the theory of algorithms, and methods of building an analytical interface.

The result of the work is the software implementation of an information and analytical system for the analysis of big data.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on any gadgets of the modern generation (Android, IOS, PC).

The program was developed in the Qlik environment.

Keywords: computer engineering, analytical systems, Big Data.

ЗМІСТ

ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	9
2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми кваліфікаційної магістерської роботи.....	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування	21
2.3 Розгорнута постановка завдання	31
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	33
3.1 Опис функціонування системи	33
3.2 Розробка структурної схеми.....	45
3.3 Розробка функціональної схеми	47
3.4 Розробка діаграми процесів.....	48
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	50
4.1 Блок-схеми та опис алгоритмів функціонування системи.....	50
4.2 Захист розробленого програмного забезпечення.....	58
5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	65
5.1 Впровадження середовища.....	65
5.2 Впровадження додатку	67
6 НАУКОВА НОВИЗНА	70

						ВКМР-123.22.0026.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата				
<i>Розроб.</i>	Чабан О. О.				Дослідження та програмна реалізація інформаційно-аналітичної системи для аналізу великих даних	Літ.	Аркуш	Аркушів
<i>Перев.</i>	Мелешко Є.В.					М	1	111
<i>Н.контр.</i>	Гермак В.С.					ЦНТУ КІ-21М 1,4		
<i>Затв.</i>	Смірнов О.А.							

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	733
7.1 Техніко-економічне обґрунтування теми магістерської роботи	73
7.2 Розрахунок трудомісткості розробки програмної продукції.....	75
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	77
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	81
7.5 Визначення собівартості розробки та ціни програмної продукції.....	86
7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції.....	89
7.7 Визначення експлуатаційних витрат.....	90
7.8 Визначення економічної ефективності програмної продукції	91
7.9 Висновки	93
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	94
8.1 Вступ.....	94
8.2 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста	96
8.3 Розрахункова частина	99
8.4 Висновки до розділу.....	102
9 ОСНОВНІ ВИСНОВКИ.....	104
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	106

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

BaRT	–	Bandits for Recommendations as Treatments
ЕОМ	–	Електрообчислювальна машина
ПЗ	–	Програмне забезпечення
ПК	–	Персональний комп'ютер
БД	–	База даних
ML	–	Machine Learning
MTML	–	multi-task multi-label

Кафедра _ КБПЗ _ 2022 рік

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. У зв'язку з стрімким зростанням обсягів оброблюваної інформації та через постійну потребу у детальній звірці старих даних спроможність людини підтримувати швидкодію та якість роботи з великими обсягами інформації стала незадовільною. Проте паралельно з ростом обсягів даних з'явилася наука роботи з великими даними.

Наука роботи з великими даними має у своїй основі дві основні цілі: швидкість та точність. Швидкість роботи рівно як і у людей обмежена, проте там де людина встигне обробити до десятка, а то і менше рядків даних кремнієві мізки встигнуть обробити до ста тисяч рядків, і на даний момент це є максимумом, який хоч і дуже рідко, але досягається. Проміжок реальної обробки даних може змінюватись від пари до п'ятдесяти тисяч рядків щосекунди, паралельно проводячи різноманітні обчислення які впливають на швидкодію. Точність роботи з BigData полягає у тому що завдяки сучасним технологіям можна виявити одиночні помилки, які були допущені при створенні баз даних та виправити їх за потреби, при цьому це не відобразиться на швидкості роботи додатка.

У проблемах реалізації інформаційно-аналітичних систем, які будуються на основі даних з BigData окрім проблеми швидкодії й достовірності оброблених даних ще є проблема донесення інформації до замовника – власника даних. Для цього існує багато середовищ розробки.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація інформаційно-аналітичної системи для аналізу великих даних.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих аналітично-інформаційних систем для аналізу великих даних.
- Дослідження аналітично-інформаційної системи для аналізу великих

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

даних.

– Програмна реалізація аналітично-інформаційної системи для аналізу великих даних.

Об'єктом дослідження є процес роботи з великими даними у QlikSense.

Предметом дослідження є методи та алгоритми розробки інформаційно-аналітичного середовища для роботи з великими даними у QlikSense.

Методи дослідження базуються на теорії аналізу даних, теорії статистики, теорії об'єктно-орієнтованого програмування, теорії алгоритмів, методах побудови аналітичного інтерфейсу.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

1. Удосконалено метод побудови аналітично-інформаційних систем, що відрізняється від існуючих можливістю обробляти інформацію незалежно-послідовним чином, що запобігає збою роботи додатку ціною актуальності найновіших даних, які вже надійшли у джерело даних.

2. Розроблено вітчизняний продукт інформаційно-аналітичної системи для аналізу великих даних, який має більш широкі можливості, на відміну від існуючих аналогів та може бути використаний для бізнес-аналітики.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі побудови інформаційно-аналітичних систем для бізнес-аналітики.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Таким чином, дослідження та програмна реалізація аналітично-інформаційної системи для аналізу великих даних, є актуальною задачею, яка потребує вирішення у даній кваліфікаційній роботі.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Довгий час перші реалізації аналітично-інформаційних систем являли собою табличну вибірку даних за певними критеріями, себто таблицю з фільтрами, яку можна було роздрукувати для презентації. Невдовзі завдяки розвитку комп'ютерних технологій стало можливим використання зведених таблиць. А з появою сучасного інтерфейсу користувача для створення аналітики у великих даних стало можливим використання різноманітних графіків та інших видів візуалізації для спрощення розуміння про загальну ситуацію.

Призначення аналітично-інформаційних систем полягає у створенні точного та швидкого засобу для роботи над даними й відображення ключових показників у інтуїтивно-зрозумілій формі. Це також передбачає створення інструкцій для розуміння що у продукті наявно та з чим воно пов'язано, щоб користуючись одним аналітично-інформаційним додатком люди, які займаються різними, пов'язаними з цими даними, питаннями змогли зрозуміти ситуацію у своїх відділах.

1.2 Область застосування

Подібне програмне забезпечення використовується різноманітними підприємствами (магазини, торговельні компанії, транспортні компанії), бібліотеками та кадровими реєстрами для сортування та надання інформації, яка повинна цікавити замовника даного аналітично-інформаційного продукту. Це допомагає власникам підприємства або ж іншої установи розуміти свої прибутки, популярність певних книг, аналізувати параметри кадрового складу.

Бувають ситуації, коли клієнт, який не розуміє призначення інформаційно-аналітичної системи може використовувати її у якості так званих «вітрини» чи

					VKMP-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

«меню». Подібну реалізацію було б доречно назвати просто інформаційною системою, проте існують задані у технічному завданні елементи які є результатом певного аналізу даних на різних етапах проектування та побудови аналітично-інформаційної системи. Основною проблемою при реалізації подібних «вітрин» є зниження швидкодії додатку через велику кількість активних обчислень. Для запобігання цьому використовується обмеження на обсяг інформації, яку відображає додаток.

У нинішні часи інформаційно-аналітичні системи використовують компанії для активного ознайомлення зі станом справ на підприємстві, що дозволяє активно вносити зміни до планів для запобігання збитків. Подібне фонове ознайомлення з ситуацією дозволяє клієнтам на зустрічах економити час, оскільки з інформацією про певні ключові моменти колектив вже буде ознайомлений.

Використання діаграм дозволить привернути увагу кінцевого клієнта до результуючого значення ключового елемента, завдяки різним формам відображення й кольоровим виділенням.

Наприклад:

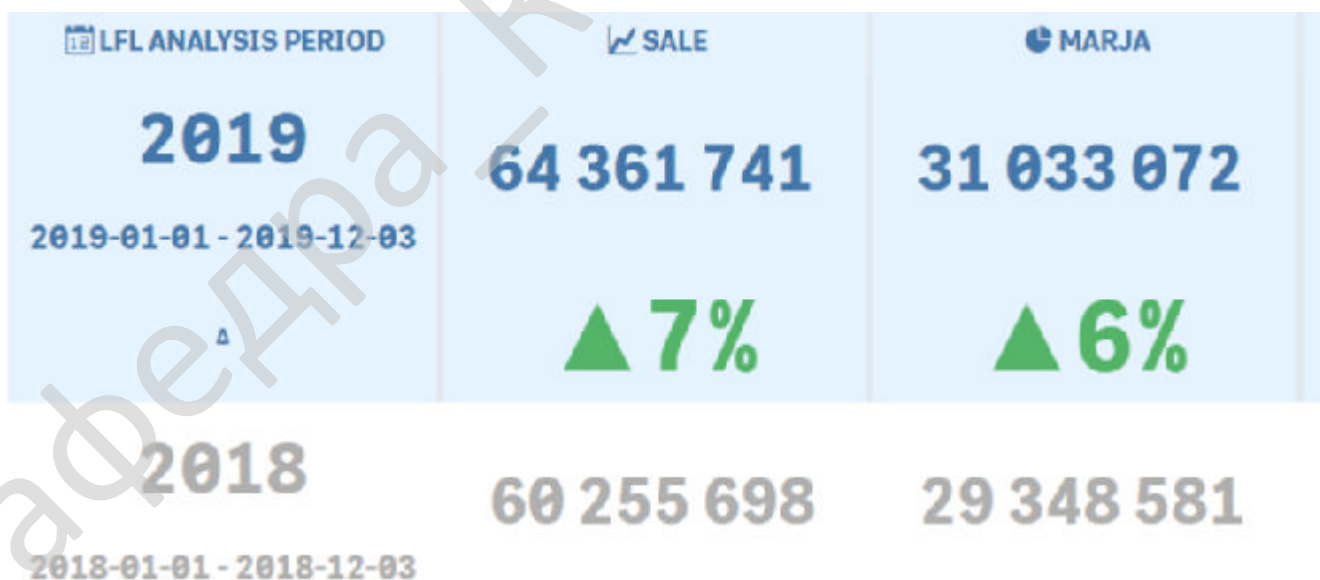


Рисунок 1.1 – виведення узагальнених LFL-показників

На рисунку 1.1 зображено ключові елементи даних за певний період у

числовому та у якісному вигляді. Даний приклад відображає загальну суму продажів та прибутків за два аналогічні періоди у різних проміжках часу і якісну зміну цих значень. Подібна візуалізація дає зрозуміти, що при збільшенні обсягу продажу, обсяг прибутку зріс слабше у порівнянні з обсягом продажів.

Для бізнесу ця ситуація не є позитивною. Подібні результати потребують додаткового аналізу, такого як: аналізу собівартості продуктів чи послуг, аналізу зростання ціни на оренду приміщень, аналіз комунальних витрат, аналіз кількості робітників та їх зарплатні, аналіз логістичних питань. Усі подібні аналізи будуть використані підприємцями для того, щоб у наступному проміжку часу відносний рівень чистого прибутку був більшим за відносний рівень загального прибутку.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми кваліфікаційної магістерської роботи

Огляд існуючих архітектур

Для інформаційно-аналітичних систем у світі цифрових торговельних відносин присвоєно назву Business intelligence.

Business intelligence (BI) – це збирання, зберігання й аналіз даних що утворюються при діяльності організації. Метою business intelligence є підтримка прийняття кращих управлінських рішень.

Термін «A Business Intelligence System» вперше ввів у 1958 дослідник фірми IBM Ганс Пітер Лун в однойменній статті.

У сучасному розумінні термін «Business Intelligence» вперше ввів Говард Дрезнер у 1989 році, як узагальнювальний термін для опису "концепцій та методів для покращення прийняття бізнес-рішень, використовуючи системи підтримки прийняття рішень, засновані на фактах" Після цього Говард Дрезнер більш ніж 10 років працював у консалтинговій компанії Gartner науковим співробітником і віцепрезидентом з Business Intelligence, що ймовірно стало суттєвою причиною закріплення оригінального та привабливо-таємничого терміну "Business Intelligence" у консалтингу і практиці ділової аналітики (Business Analytics).

У 1996 р. Gartner дала наступне визначення терміну Business Intelligence: "Програмні засоби, що функціонують в рамках підприємства і які забезпечують функції доступу та аналізу інформації, яка міститься в сховищі даних, а також забезпечують прийняття правильних і обґрунтованих управлінських рішень."

У першій декаді XXI ст. Gartner значно розширило поняття даного терміну: "BI – це узагальнювальний термін, що включає застосунки, інфраструктуру й

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

інструменти, а також кращі практики, які забезпечують доступ до інформації та її аналіз з метою оптимізації рішень і управління ефективністю.

Тож можна сказати, що ВІ-системи для аналізу даних - це методи та інструменти збору та структурування інформації для подальшого бізнес-аналізу.

Найчастіше системи ВІ застосовуються, коли необхідно:

- Підготувати звітність у прискореному режимі;
- Виявити якусь закономірність;
- Знайти ефективний метод для вирішування проблем бізнесу;
- Інтегрувати дані з кількох джерел одночасно;
- Надати співробітникам, які перебувають у різних локаціях, оперативний доступ до даних. Залежно від посади співробітники отримують інформацію різного рівня деталізації, необхідного для виконання конкретних завдань;
- Впровадити нові ІТ-рішення;
- Оновити існуючі алгоритми.

Технології для бізнес-аналітики трансформують масштабні обсяги даних компанії на зрозумілі структуровані вичавки. Основна мета ВІ – надати бізнесу всі дані для ухвалення обґрунтованого рішення. Тобто це означає, що система самостійно збирає дані, прибирає непотрібну інформацію та робить короткі вибірки даних, наголошуючи на найважливіших аспектах, факторах зростання ефективності. Наприклад, топ-менеджери компаній з різних сегментів використовують такі рішення, щоб розуміти, який напрямок їх підприємства працює більш ефективно у порівнянні з іншими, а також допомагає відстежувати причини успішності того чи іншого відділу. Допустимо, аналітик вводить до системи ключові параметри діяльності компанії, які він не буде змінювати без нагальної потреби. Інформаційно-аналітична система автоматично розраховує динаміку, а експертна система ще й пропонує варіанти дій у разі негативних зрушень. ВІ-технології дозволяють отримувати аналітику за заданими параметрами у режимі реального часу, і це чи не найголовніша перевага ВІ-систем.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Бізнес-аналітика застосовується у величезній кількості галузей. Вона використовується для прийняття рішень на різних рівнях управління. Так, великі компанії залучають ВІ-технології для прийняття обґрунтованих рішень, розвитку бізнесу, скорочення витрат, а також для допомоги у пошуку нових перспектив розвитку. Яскравим прикладом є компанія ABN AMRO Clearing Bank N.V., яка займається оцінкою ризиків угод чи вкладень. У період пандемії та особливої нестабільності ринку компанія стала використовувати ВІ-технології, які стали важливим інструментом прогнозування ризиків фінансових угод.

Керівникам ІТ-підрозділів бізнес-аналітика необхідна для виявлення не ефективних ланок в операційно-технічному супроводі бізнесу. Молодий сервіс Koodos, що займається підбором контенту аналізуючи емодзі. У своїй роботі він тестує багато різноманітних алгоритмів пошуку та підбору від історії пісень до асоціацій їхніх авторів. Використовуючи бізнес-аналітику, керівники безпосередньо оцінюють ефективність використання тих чи інших засобів, які допомагають розробникам удосконалювати свій продукт для того, щоб користувачам було легше використовувати подібні програмні рішення.

Підприємства громадського харчування, кафе та ресторани використовують системи бізнес-аналітики для прийняття стратегічно важливих рішень, серед яких виявлення неефективних точок мережі, знаходження непопулярних страв і, навпаки, найбільш затребуваних товарів та ін., за допомогою яких відбувалося обчислення загального виторгу ресторану, його продаж у зазначений час. Сервіс допоміг менеджерам ресторану відстежувати необхідні дані та дуже швидко оцінювати рівень роботи закладу.

Топ-менеджмент компаній виявляє збиткові та прибуткові напрямки бізнесу за допомогою ВІ-технологій, що допомагає спрогнозувати динаміку зростання виторгу компанії. Звідси впливає користь для відділу продаж та фінансів. Вони можуть планувати бюджет компанії, беручи за основу результати бізнес-аналізу. Так, при виробництві нових товарів компанія Brown-Forman завжди спирається на дані, зібрані за допомогою бізнес-аналітики. Саме з їх допомогою можна

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

оперативно оцінити успішність виробленого зовсім недавно продукту і прийняти рішення про зняття його з продажу або підвищення обсягів виробництва.

Окремо варто сказати про сегмент роздрібної торгівлі. Там ВІ допомагають проаналізувати продажі – визначити «ходові» товари, передбачати обсяг продажу того чи іншого товару тощо. Такими даними користується сервіс Stitch Fix, який допомагає своїм клієнтам за допомогою команди стилістів підбирати найвдаліші образи. За допомогою аналізу покупок одягу клієнтом, співробітники не просто підбирають максимально вдалі для нього фасони або моделі, але також можуть збирати дані про найбільш популярні та універсальні товари і менш популярні.

Щодо державного сектора, то найчастіше ВІ-технології допомагають реалізувати інструменти прогнозування та автоматизації бюджетування. Так, уряд штату Міссісіпі, США створив портал, який допомагає відстежувати стан доріг та мостів регіону. Він створює статистику стану, працездатності, дорожніх пригод, таким чином допомагаючи керівникам планувати бюджет, оцінюючи всі необхідні витрати на підтримку транспортних систем.

Аналогічні системи

На даний момент найбільш популярними аналітично інформаційними системами є:

- Power BI від Microsoft;
- Qlik View від шведської QlikTech;
- Tableau від американської Tableau Software, Inc.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

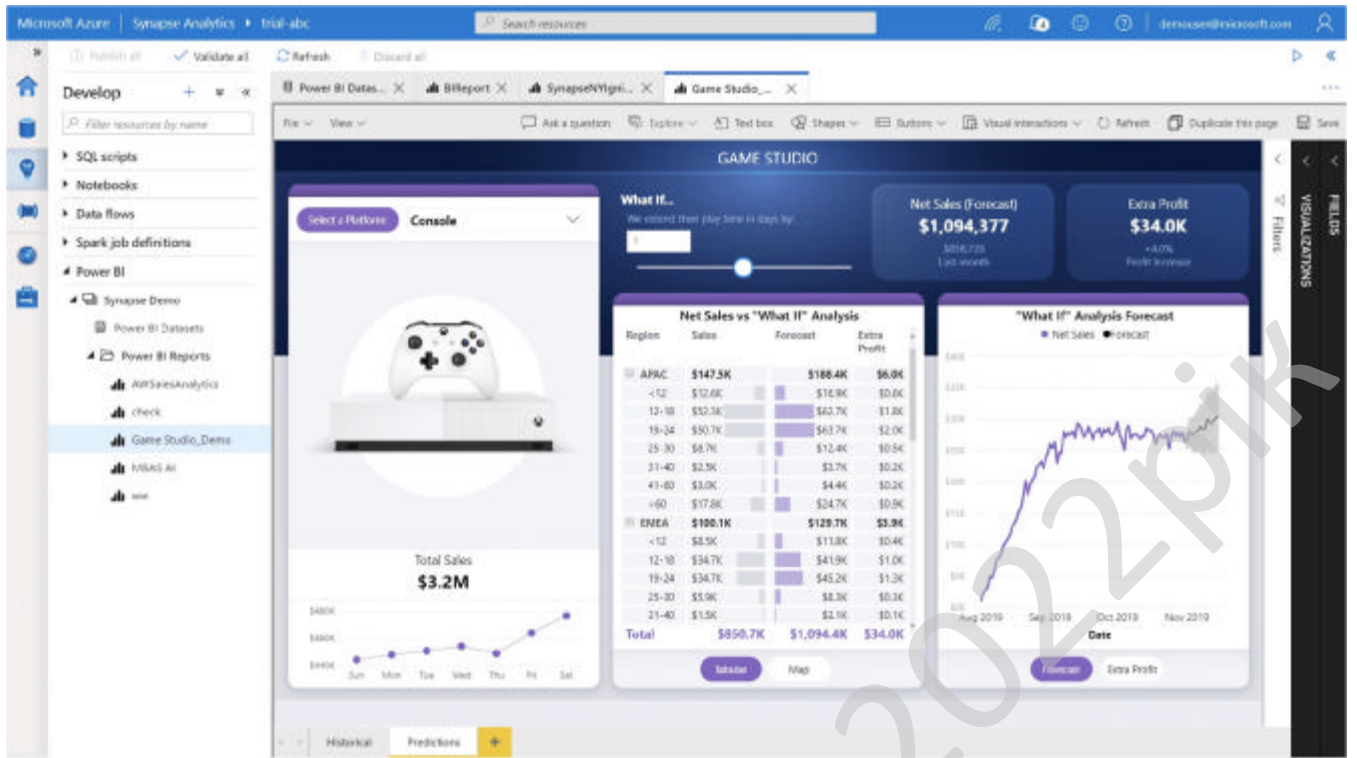


Рисунок 2.2 – інтерфейс побудови листів середовища Power BI на хмарній платформі Azure

Завдяки функціоналу, який включає велику кількість шлюзів (зв'язків), Power BI дозволяє зібрати воєдино інформацію з різних джерел, консолідувати її та проаналізувати. Такими джерелами можуть бути як власні розробки Microsoft (напр. Excel), так і внутрішні бази даних підприємств, установ, організацій.

Програму задумали Тьєррі Д'Херс та Амір Нец з команди служб звітування SQL Server у Microsoft. Спочатку сервіс був розроблений Роном Джорджем влітку 2010 року і названий Project Crescent. Project Crescent був доступний для загального завантаження 11 липня 2011 року в комплекті з кодовою назвою SQL Server Denali, пізніше перейменований у Power BI, а потім був представлений Microsoft у вересні 2013 року як Power BI для Office 365.

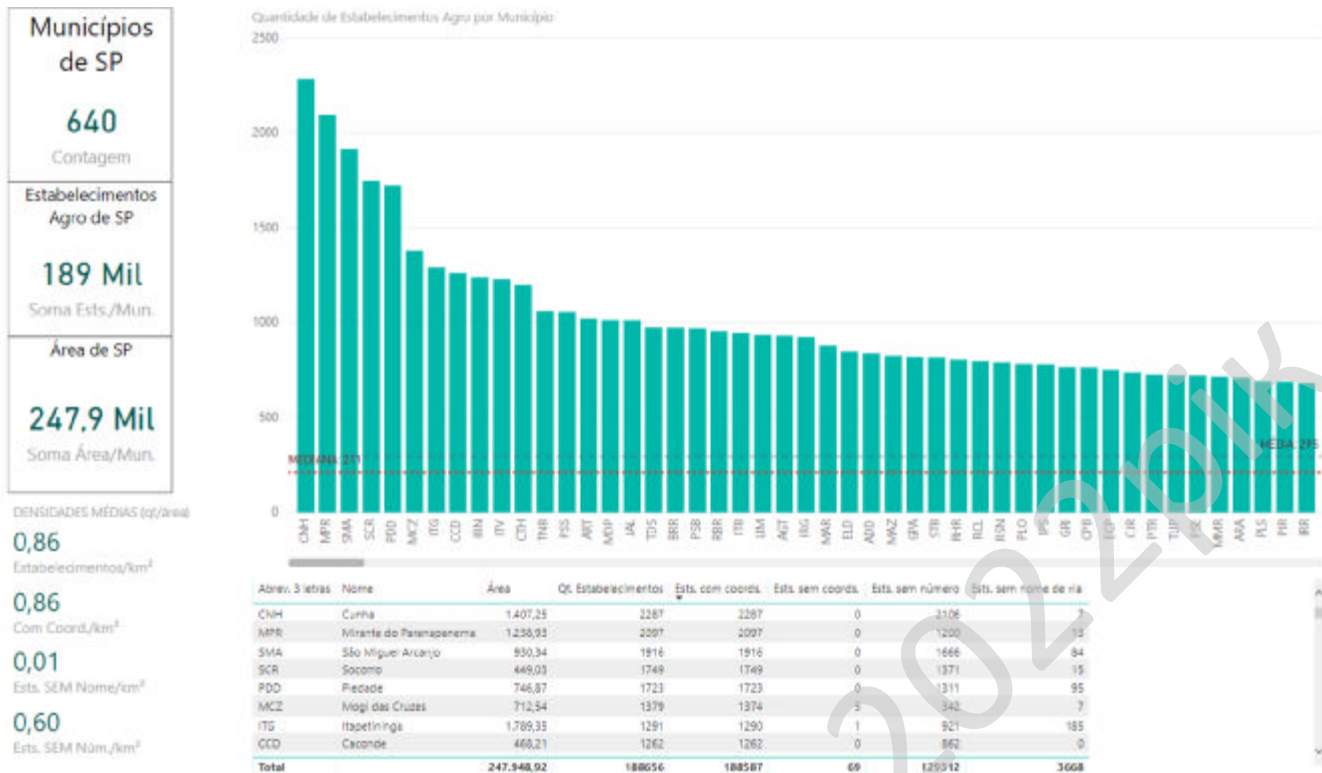


Рисунок 2.3 – інтерфейс листа у середовищі Power BI

Як можна побачити на рисунку 2.1, через те, що середовище для побудови інформаційно-аналітичних систем Power BI було розроблено корпорацією Microsoft, то дане середовище має високу сумісність з продуктами Microsoft Office.

Це дозволяє виводити результуючі дані одним натисканням клавіші миші та автоматично підключатись до хмар даних де, можуть розташовуватись бази даних. Подібна зручність одночасно є також недоліком, оскільки для роботи з базами даних від корпорації Microsoft у пакет Microsoft Office включена програма Access, яка підходить для ведення даних для навчання/ознайомлення з тим що таке бази даних та невеличкими фірмами які тільки починають працювати чи з тими фірмами для яких рівня можливостей даного програмного засобу достатньо або може бути, що власники підприємств просто не хочуть переходити на більш дорогі та більш зручні програми для ведення обліку.

Рисунок 2.3 демонструє нам аналітично-інформаційний лист низької якості.

Даний лист складається з 3 частин:

1. КПЕ, яке розташоване у лівій стороні листа;

2. Невелику за площею на листі таблицю;
3. Гістограму, яка займає левову частину листа.

Недоліки у даному прикладі присутні на кожному з елементів:

– КПЕ. Наявно 2 типи КПЕ, кожен тип має свій дизайн, подібні результуючі показники було б доречно зробити в єдиному стилі, чи надати можливість переключатись між ними. Також доречно було б помістити їх на «верхній рядок» листа.

– Таблиця та діаграма. Дуже велика гістограма, у той час, як таблицю зробили дуже малою, крім того, гістограма гарна, проте взагалі не інформативна, оскільки побудована лише на двох вимірах таблиці.

– Найкращим розташуванням об'єктів на листі було б наступне: Верхній рядок – найбільш важливі КПЕ, таблицю зробити максимально великою з шириною яка відповідає ширині використовуваної площі (у даному випадку звузити), перевернути гістограму вертикально й помістити її на місце яке залишилось.

Tableau

Tableau (Tableau Software, Inc.) – американська компанія, розробник однойменного програмного забезпечення для інтерактивної візуалізації даних та бізнес-аналітики. Заснована у січні 2003 року, 1 серпня 2019 року поглинена корпорацією Salesforce за \$15,7 млрд.

Засновники – дослідники факультету інформатики Стенфордського університету Крістіан Шабо, Пет Ханрахан та Кріс Столте, які спеціалізувалися на методах візуалізації для реляційних баз даних та кубів. У період з 1999 до 2002 року функціонувала як комерційний дослідницький центр при Стенфорді. Як самостійна компанія зареєстрована в 2003 році в Маунтін-В'ю, через рік штаб-квартира перенесена до району Фремонт у Сіетлі. До 2016 року компанію очолював Шабо, після чого генеральним директором було призначено Адама Селіпського. У березні 2017 року компанія збудувала нову будівлю штаб-квартири у Воллінгфорді, у 2018 році було відкрито новий офіс у Фремонті. У червні 2018 року компанія придбала

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

стартап Empirical Systems із Кембриджського Університету, що спеціалізується на технологіях глибокого навчання.

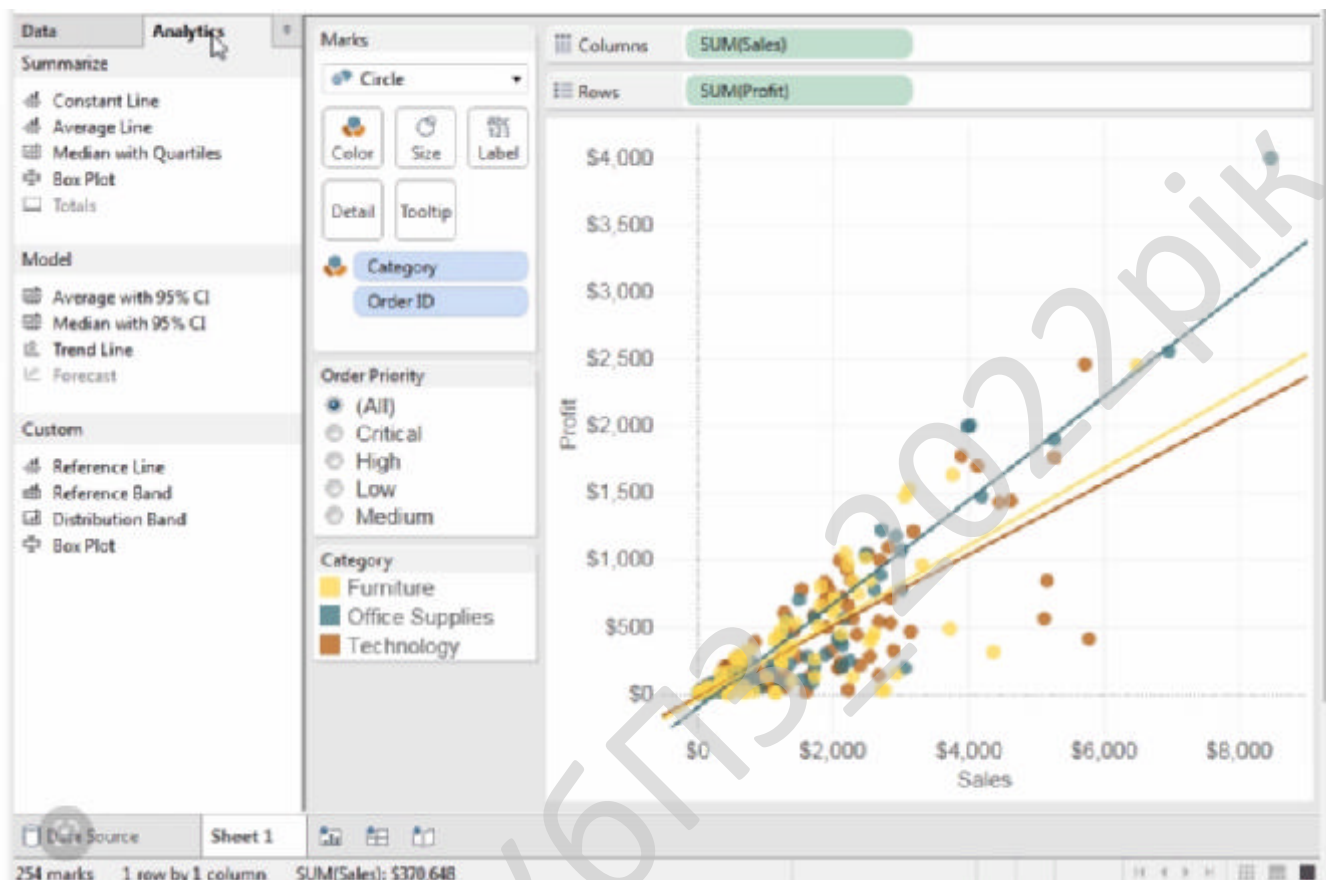


Рисунок 2.4 – інтерфейс розробки листа у Tableau

На рисунку 2.5 наведено коректний приклад реалізації листа з аналітикою. У верхньому рядку наведено єдиноформатне КПЕ, ліва частина листа виділена для фільтрів. Решта листа розділена на 2 половини, у лівій лінійчатій діаграмі з LFL, а у другій чотири LFL-гістаграми, які відображають коливання мір протягом періодів розбивки.

Sales Summary

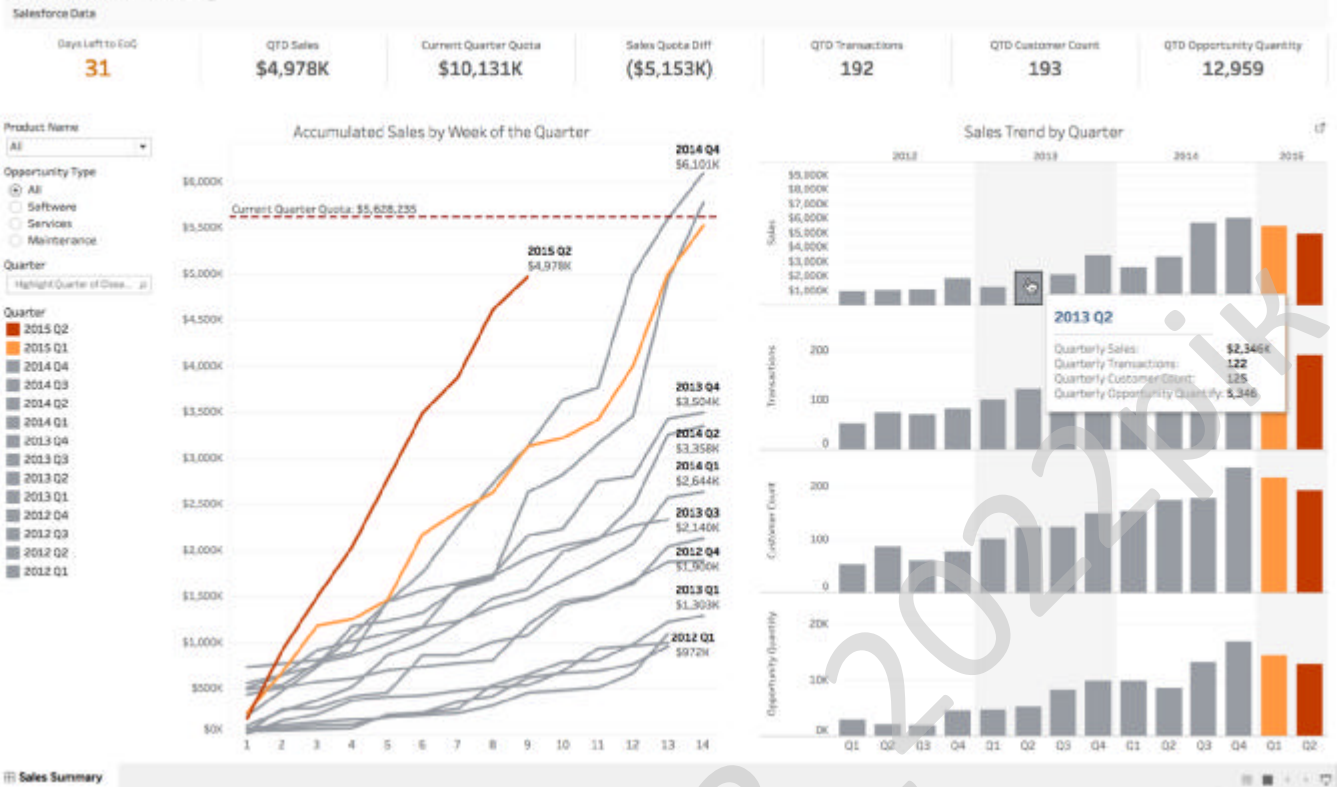


Рисунок 2.5 - інтерфейс листа у середовищі Tableau

QlikView

QlikView – це аналітичне рішення для швидкої розробки інтерактивних аналітичних програм високого рівня та панелей моніторингу, що забезпечують подання інформації з ділових завдань. Перша версія програми була представлена у 1993 році після 10 років розробки

Програма QlikView від компанії QlikTech дозволяє вільно досліджувати бізнес-дані, щоб дізнаватися факти і розуміти співвідношення даних у багатьох джерелах.

Програмне забезпечення QlikView забезпечує високу швидкість реагування на оточення бізнесу що змінюється, скорочує час для оцінки ситуації і дозволяє більш глибоко розуміти ситуацію.

Інтуїтивно зрозумілий інтерфейс програмного продукту QlikView дозволяє користувачам використовувати програму без попереднього тренування, що спрощує бізнес-аналітику та створення звітності, знижуючи навантаженість ІТ-

відділу та підвищуючи залученість до аналізу даних кінцевого користувача. Програма QlikView поєднує у собі засоби вилучення, перетворення, завантаження даних (англ. ETL), зберігання даних, багатовимірний аналіз та інтерфейс кінцевого користувача одному пакеті.

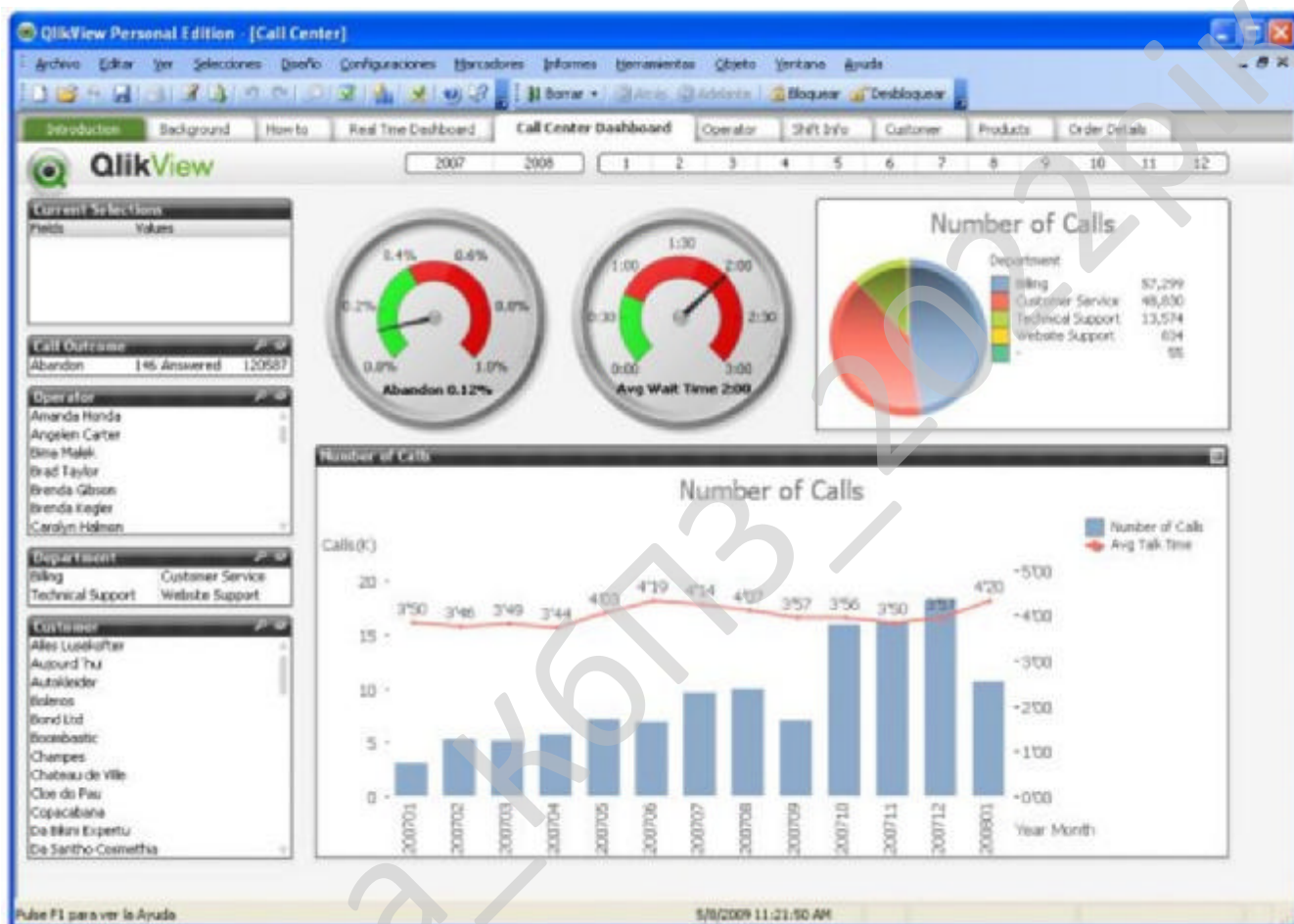


Рисунок 2.6 – інтерфейс листа QlikView

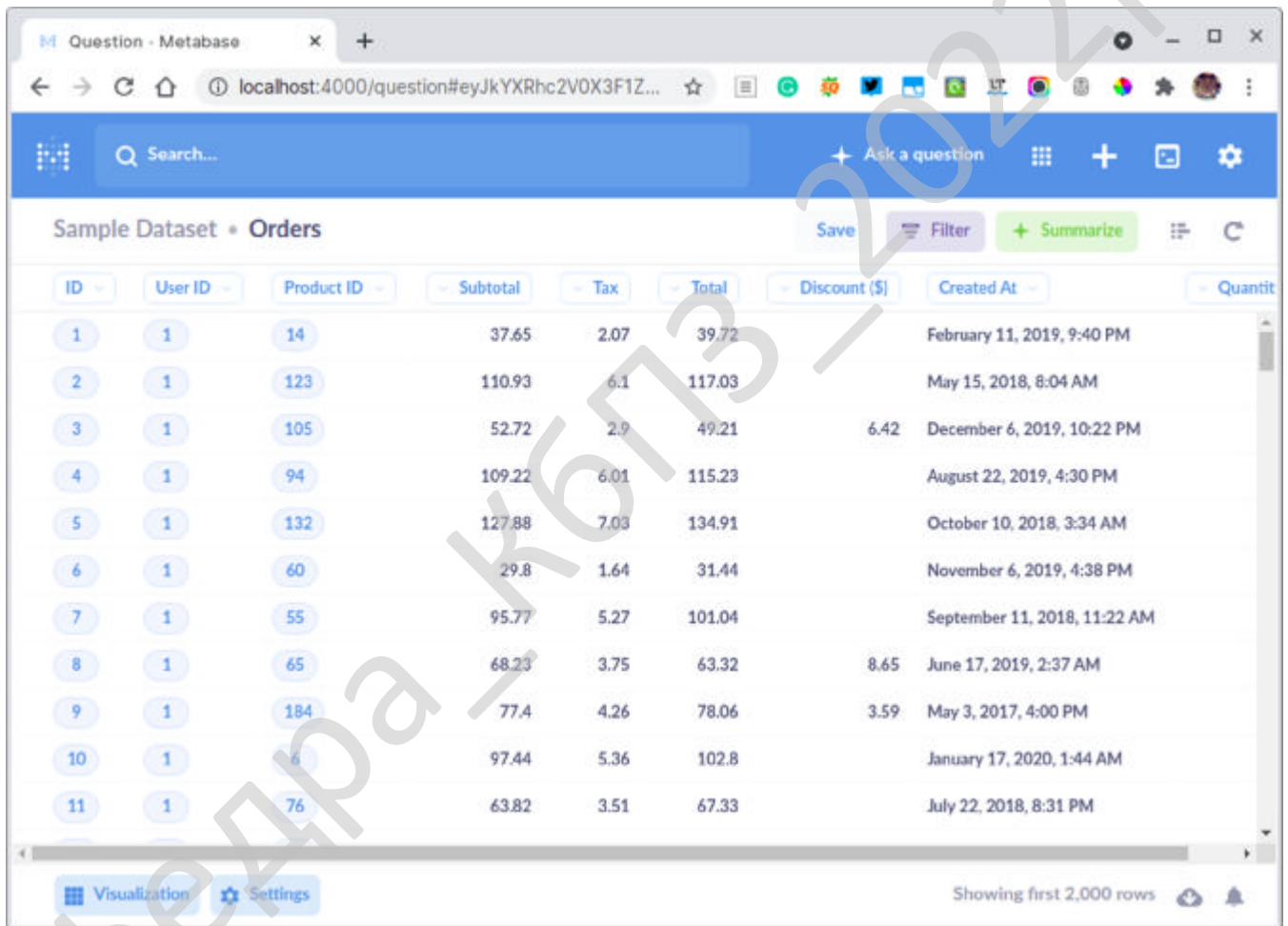
Даний програмний продукт є прикладом корпоративної платформи і підходить для проведення внутрішньої аналітики. Крім того система зав'язана на сталій формі відображення та не має можливостей оперативно виводити оновлену інформацію дистанційно. Себто це означає що дана платформа розробки підходить лише для внутрішнього використання продукту тільки на РС.

Розуміючи, що майбутнє за мобільністю компанія QlikTech розробляє середовище – спадкоємця Qlik Sense.

Metabase

Незважаючи на кількість вже створених середовищ для розробки ВІ, постійно створюють нові середовища. Складність розробки у середовищах може змінюватись, деякі будуть підходити «програмістам», а деякі створюють лише для «користувачів». Однією з таких платформ для «користувачів» є Metabase.

Metabase – це нове середовище розробки ВІ. Яке характеризується тим, що у ньому дані будуються без можливості значної зміни вхідних даних.



ID	User ID	Product ID	Subtotal	Tax	Total	Discount (\$)	Created At	Quantit
1	1	14	37.65	2.07	39.72		February 11, 2019, 9:40 PM	
2	1	123	110.93	6.1	117.03		May 15, 2018, 8:04 AM	
3	1	105	52.72	2.9	49.21	6.42	December 6, 2019, 10:22 PM	
4	1	94	109.22	6.01	115.23		August 22, 2019, 4:30 PM	
5	1	132	127.88	7.03	134.91		October 10, 2018, 3:34 AM	
6	1	60	29.8	1.64	31.44		November 6, 2019, 4:38 PM	
7	1	55	95.77	5.27	101.04		September 11, 2018, 11:22 AM	
8	1	65	68.23	3.75	63.32	8.65	June 17, 2019, 2:37 AM	
9	1	184	77.4	4.26	78.06	3.59	May 3, 2017, 4:00 PM	
10	1	6	97.44	5.36	102.8		January 17, 2020, 1:44 AM	
11	1	76	63.82	3.51	67.33		July 22, 2018, 8:31 PM	

Рисунок 2.7 – інтерфейс розробки Metabase

						ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			20

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Огляд середовища розробки Qlik Sense

Qlik Sense – програма для аналізу даних. За допомогою програми Qlik Sense Ви зможете 1 раз запрограмувати завантаження даних, налаштувати звіти та надалі кожен день аналізувати актуальні дані у програмі. При цьому система буде стабільно працювати.

На відміну від QlikView, Qlik Sense є інструментом для аналізу даних з адаптивною версткою (візуалізація будується на основі web принципів/механізмів).

Qlik Sense – це програма для побудови візуалізації, дослідження та моніторингу даних. Qlik Sense – новий продукт компанії Qlik, BI-платформа з асоціативним пошуком в оперативній пам'яті із вбудованими засобами ETL. Випущена в 2014 році та відрізняється від попередника можливостями до більш самостійної роботи з даними для бізнес-користувачів, бо на відміну від нового продукту, QlikView більшою мірою керували IT-фахівці.

Середовище розробки Qlik Sense використовує двигун другого покоління (QIX), візуалізація розроблена з використанням HTML5, CSS та JavaScript (за рахунок чого аналітичні програми мають адаптивну верстку і за рахунок цього їх можна використовувати на будь-яких пристроях).

Аналітична система Qlik Sense дозволяє зібрати інформацію з розрізнених джерел єдину інформаційну систему. При цьому для Qlik Sense не потрібне спеціальне сховище даних.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

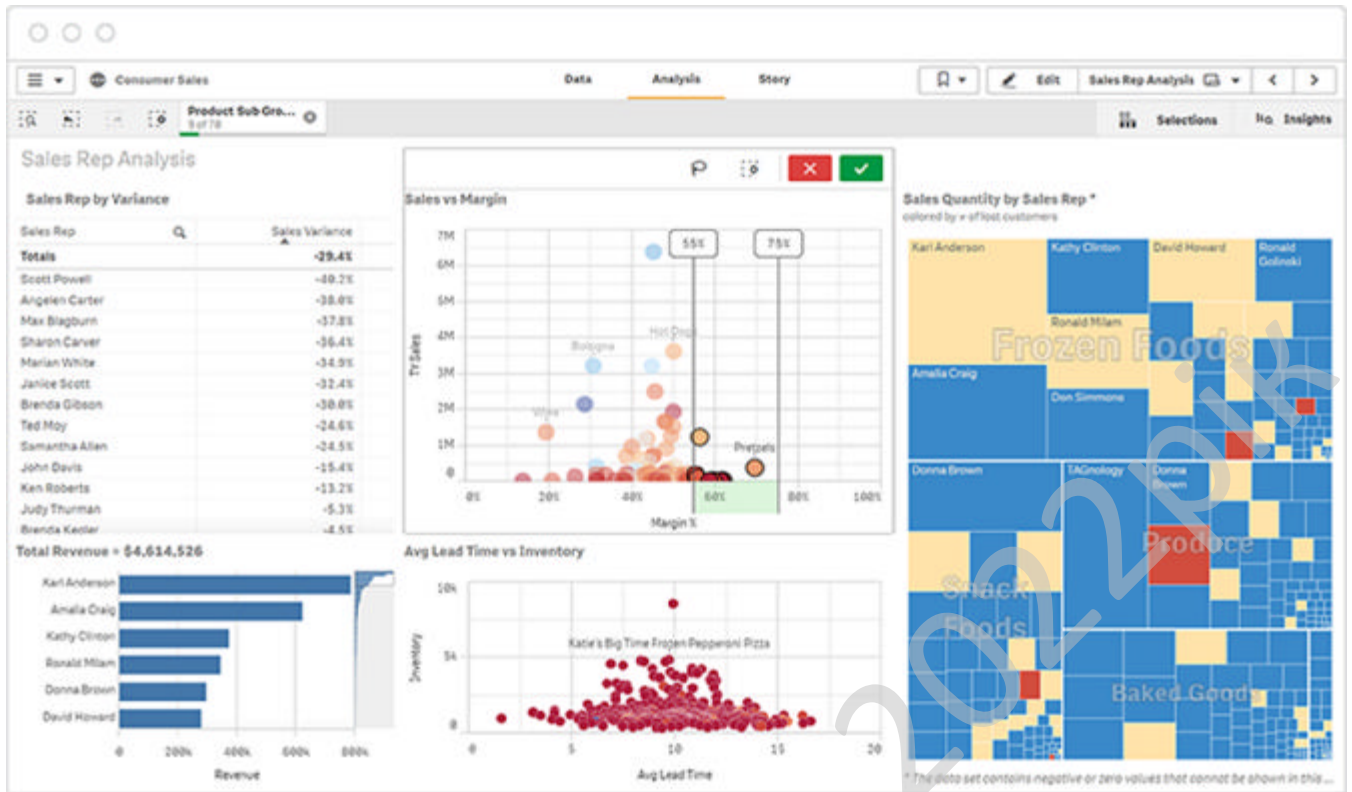


Рисунок 2.7 – інтерфейс листа Qlik Sense

Якщо Ваші вихідні дані таблиці Excel, то ви зможете навіть без допомоги IT-фахівців завантажити дані в додаток, просто перетягнувши файли в Qlik Sense.

Qlik Sense використовує асоціативну модель зв'язку даних. Перевага асоціативної моделі даних: динамічні асоціації, можливість дослідження даних у напрямі, розуміння взаємозв'язків даних, бачення всіх взаємозв'язків між даними.

Qlik Sense може імпортувати дані з різних джерел, агрегувати їх і розглядати за різними схемами і моделями даних. Забезпечується контакт практично з будь-якою базою даних через ODBC та DB OLE, включаючи джерела Big Data, такі як Cloudera, Hortonworks, Vertica та Teradata.

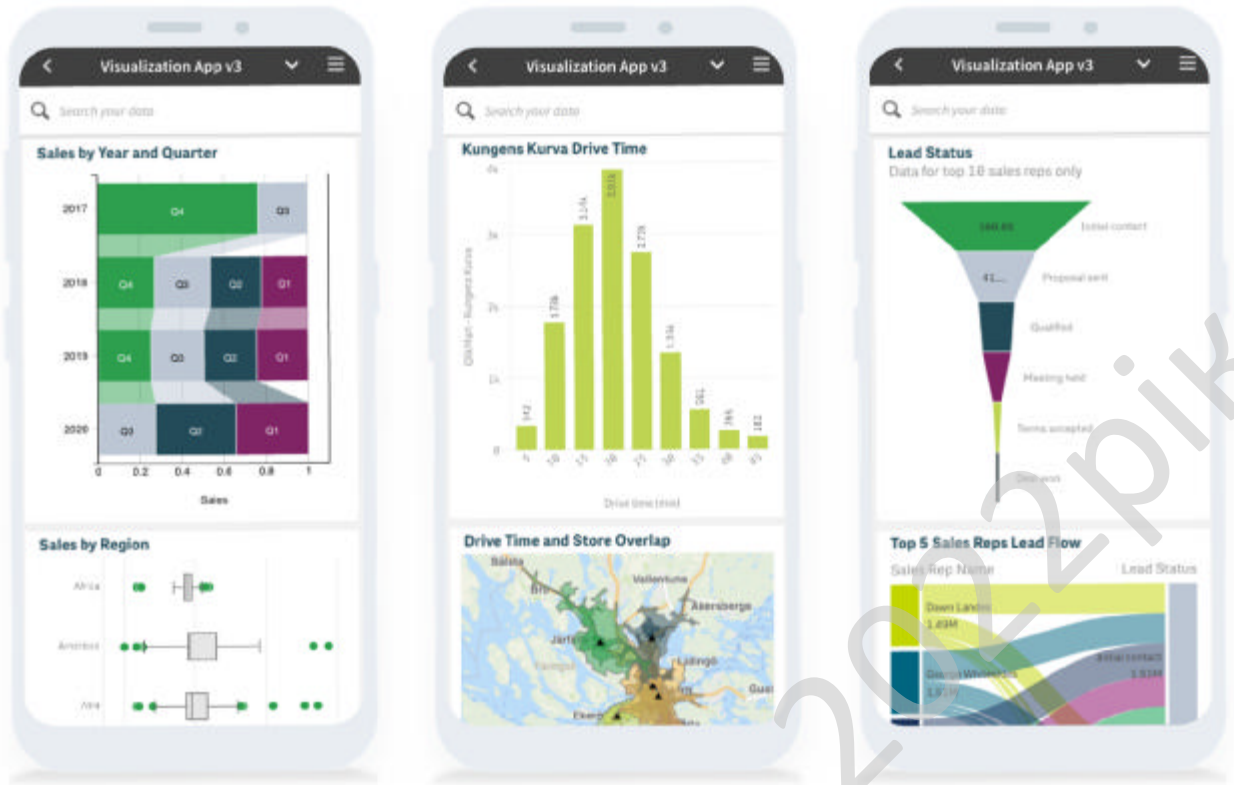


Рисунок 2.8 – мобільний інтерфейс листів додатків Qlik Sense

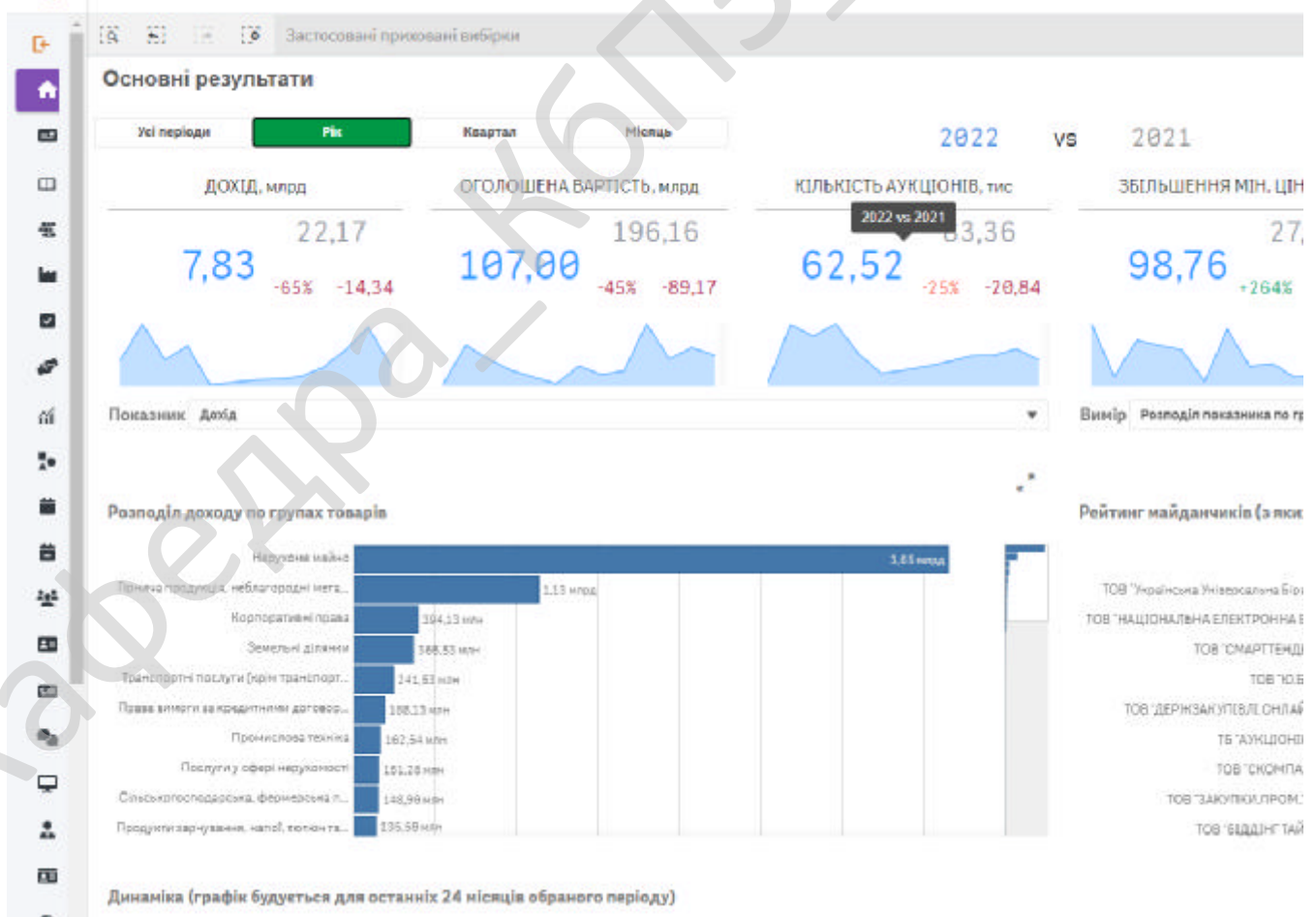


Рисунок 2.9 – інтерфейс мешапу

Завантаження даних із хмарних систем (API -> JSON) можна здійснювати за допомогою QLIK REST-конектора, а також за допомогою PHP у форматі CSV.

У середовище розробки користувацького інтерфейсу вбудовано автоматична адаптація листів у мобільному форматі, яку можна додатково налаштувати, приклад результату наведено на рисунку 2.8. Перетворення листа з широкоформатного розміру у мобільний підтримує усі встановлені на додаток плагіни, оскільки інтерфейс представляє собою сітку на яку «прикріплюються» елементи візуалізації.

Окрім використання «чистого» середовища Qlik Sense можливо створювати різноманітні мешапи.

Огляд принципів розробки мешапів у середовищі розробки Qlik Sense

Як можна побачити, то не зважаючи на повну схожість елементів впровадження для різноманітних елементів додатку принципи використання QlikSense відрізняються від QlikView.

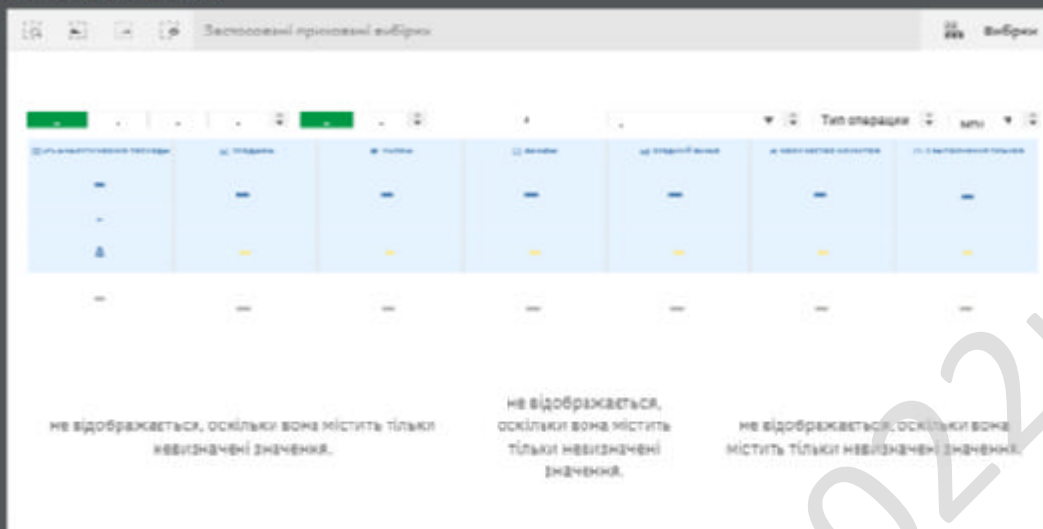
У випадку який наявний на рисунку 2.10 для успішного використання кругової, у даному випадку, діаграми нам потрібно створити окремий елемент: «Кругова діаграма "А"». Після чого дані про цей елемент ми передаємо далі.

Випадок на рисунку 2.9 потребує створення окремих елементів для кожного елементу діаграми як видимого користувачем, так і ні. Що, у свою чергу, дозволить елементу мешапу працювати немов годинник. Використання цілого листа у вигляді мешапу зазвичай не використовується через те що даний елемент мешапу оперує з цілим листом зберігаючи пропорції, через що площа для відображення потрібної інформації зменшується.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Вбудувати лист

Попередній перегляд



Ідентифікатор листа: f5192488-2aa4-4ddd-9ac7-8f2bd4097381

Рекомендується вбудовувати об'єкти тільки у внутрішні веб-сторінки.
Візуалізація буде оновлена одночасно з оновленням додатка.

Загальне

Дозволити взаємодію
Так

Включити контекстне меню
Так

Дозволити вибірки
Так

Мова

Тема

Інтернет-кадр для впровадження у веб-сторінку

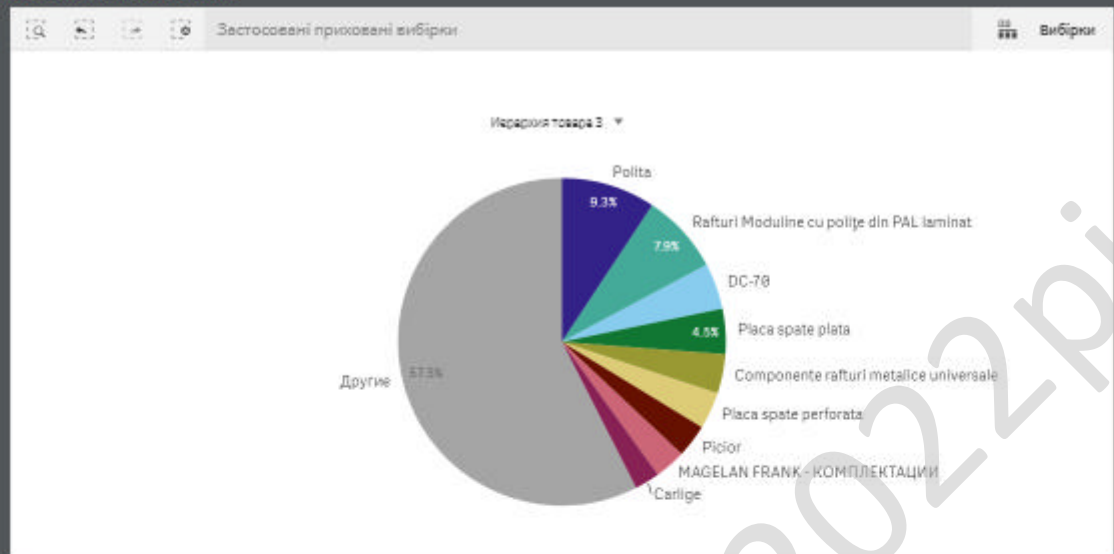
`PeriodId,Year" style="border:none;width:100%;height:100%;"></iframe>`

[Відкрити вікно на новій вкладці](#)

Рисунок 2.9 – елементи вбудовування цілого листа додатку з аналітикою

Впровадити

Попередній перегляд



Ідентифікатор об'єкту: msnzejm

Рекомендується вбудовувати об'єкти тільки у внутрішні веб-сторінки.
Візуалізація буде оновлена одночасно з оновленням додатка.

Загальне

Дозволити взаємодію

Так



Включити контекстне меню

Так



Дозволити вибірки

Так



Мова

За замовчуванням



Тема

Sense Класика



Інтернет-кадр для впровадження у веб-сторінку

Копіювати

[Відкрити вікно на новій вкладці](#)

Рисунок 2.10 – елементи вбудовування однієї з інформаційних діаграм листа додатку з аналітикою

Вим.	Арк.	№ докум.	Підпис	Дата

ВКМР-123.22.0026.00.00.ПЗ

Арк.

26

Реалізація мешапу відбувається за двома принципами:

1. Мешап являє собою вставку у сторінку. Гарний приклад цьому – це використання мап з Google maps. На таких картах ми можемо спокійно взаємодіяти, якщо це можливо, з основною частиною інтерфейсу Google maps залишаючись на потрібному нам сайті не полишаючи його. Зазвичай для інтеграції достатньо базових мов розмітки: HTML, CSS і JS. Проте усе залежить від того як саме розроблявся сайт.

2. Мешап являє собою окремий сайт. І даний елемент буде детально розглянуто нижче на прикладі використання інформаційно-аналітичного додатку розробленого у середовищі Qlik Sense.

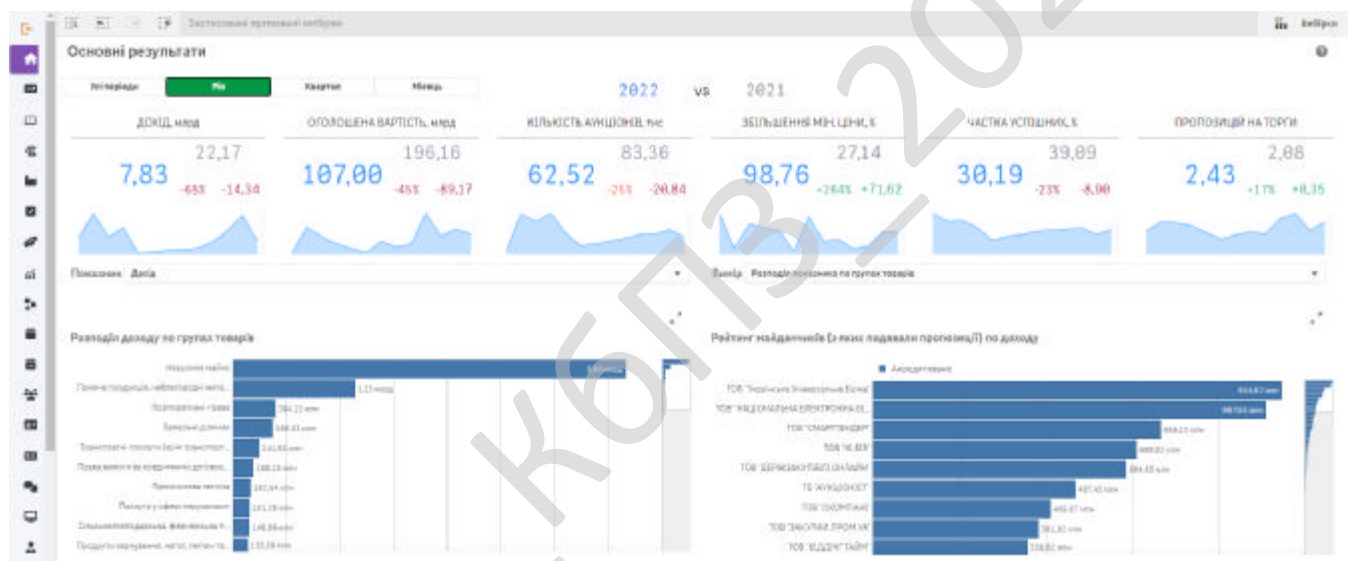


Рисунок 2.11 – Приклад інтерфейсу сайту-мешапу

Для створення будь-якого мешапу з елементами середовища Qlik Sense потрібно створити підключення до даних додатку на основі якого будується мешап.

У випадку з сайтом-мешапом для його розробки окрім HTML, CSS, JS та JSON використовують ще мову Qlik-ку. На мові Qlik-ку побудовано дизайн сайту, відбувається зв'язок з розширеннями Qlik Sense та з даними цільового додатку чи додатків.

```

package-lock.json > |> dependencies > |> @babel/core > |> @babel/core
20   "babel/core": {
21     "version": "7.12.1",
22     "resolved": "https://registry.npmjs.org/babel-core/-/babel-core-7.12.1.tgz",
23     "integrity": "sha512-V46401Y08uZBNp6D/01/YrUwHAX6U1FJelZhI6t701NGWd/01okjgE7oY4vjwKE1V4wrtiIcnUkPwyYl4Q==",
24     "requires": {
25       "babel-generator": "7.12.1",
26       "babel-helper-define-property": "7.12.1",
27       "babel-plugin-transform-runtime": "7.12.1",
28       "babel-preset-current-versions": "7.12.1",
29       "babel-preset-latest": "7.12.1",
30       "babel-runtime": "7.12.1",
31       "babel-traverse": "7.12.1",
32       "babel-types": "7.12.1",
33       "compare-versions": "4.1.0",
34       "debug": "4.1.0",
35       "gensync": "1.0.0-beta.2",
36       "json": "5.1.2",
37       "lodash": "4.17.19",
38       "resolve": "1.1.2",
39       "semver": "5.6.0"
40     }
41   },
42   "dependencies": {
43     "semver": {
44       "version": "5.6.0",
45       "resolved": "https://registry.npmjs.org/semver/-/semver-5.6.0.tgz",
46       "integrity": "sha512-P40eZt88h6a4E8y6b+b/dfbJfGsmoKy/mvMI9UZZhUyosqAa+0pUkgeiIm1Y3pl/oO/wWHt9+/YDp457Tj0="
47     }
48   },
49   "babel/generator": {
50     "version": "7.12.1",
51     "resolved": "https://registry.npmjs.org/babel-generator/-/babel-generator-7.12.1.tgz",
52     "integrity": "sha512-98CestkGJL3zLP2A9vkUsfVajvbg4mIFiaT86kw20CUP1qbo8YF/RTw74LnjMeAChW732EtToOJg3AjtXs1Q==",
53     "requires": {
54       "babel-types": "7.12.1",
55       "lodash": "4.17.19",
56       "source-map": "0.5.6"
57     }
58   }
59 }

```

Рисунок 2.12 – Приклад побудови JSON-структури.

```

import QlikCard from "components/QlikCard";
import QlikObject from "components/QlikObject";

export default function YearMonthFilters() {
  return (
    <div className="row mb-2">
      <div className="col-12">
        <QlikObject
          id="v100bv"
          height="35px"
          showLoader={false}
        />
      </div>
    </div>
    <div className="row">
      <div className="col-1g-12 col-12">
        { /* v100bvYearFilter */ }
        <QlikCard
          id="1f84c354-742d-49f8-b883-18859687626e"
          height="34px"
          showLoader={true}
          showHeader={false}
        />
      </div>
    </div>
  );
}

```

Рисунок 2.12 – Приклад інтеграції елементів на лист сайту.

Розташовування раніше перетворених елементів додатку Qlik Sense у

елементи сайту відбувається з а допомогою JS.

За допомогою вбудованого середовища DewHub можна зробити пробний мешап.

Огляд системних підключень у середовищі розробки Qlik Sense

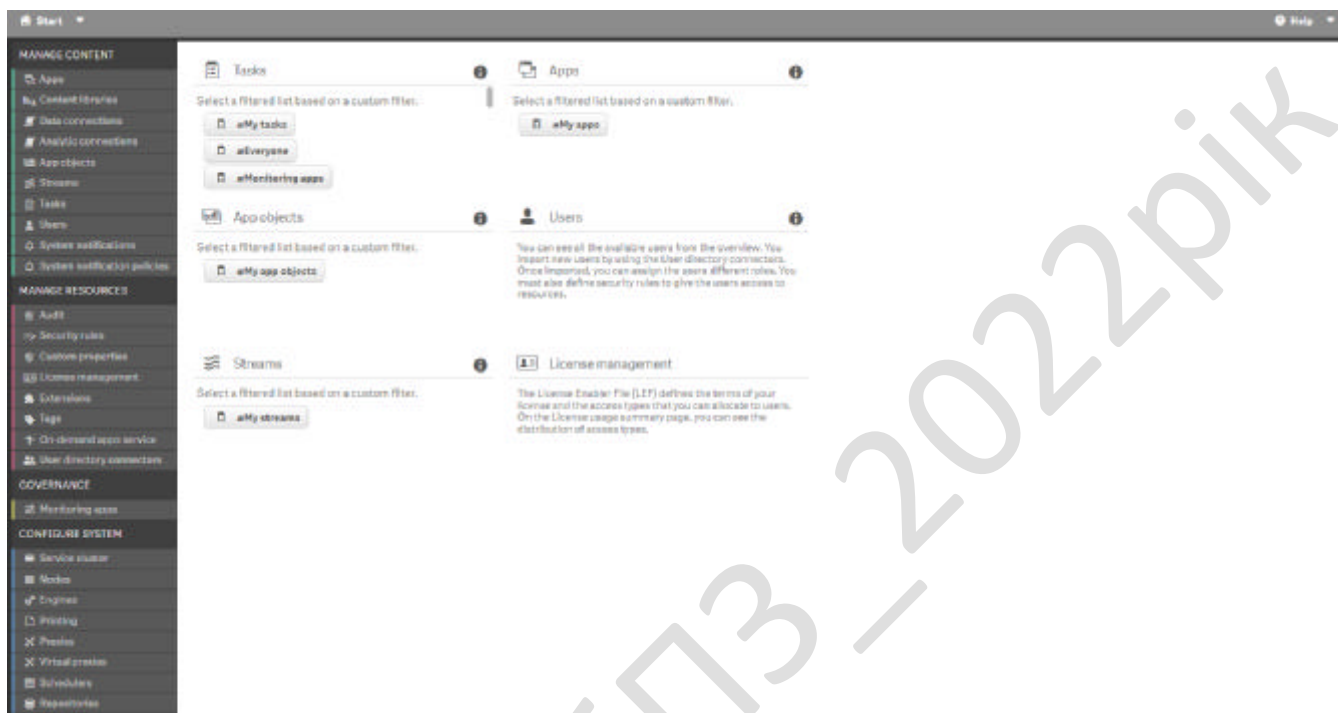


Рисунок 2.13 – інтерфейс системних підключень QMC

Для полегшення взаємодії між підключеннями додатків, серверів, джерел для зберігання та використання баз даних та доступів використовують вбудоване середовище QMC.

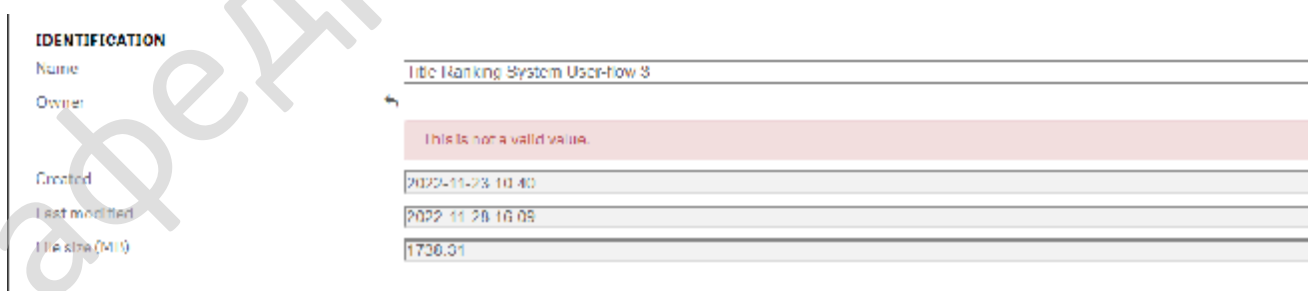


Рисунок 2.14 – стартове вікно огляду додатку QMC

Name	Type	Approved	Published	Last modified	App
Branch Clusters	sheet	Not approved	Not published	2022-11-23 10:40	Title Ranking System Use...
Cluster Analysis	sheet	Not approved	Not published	2022-11-23 10:40	Title Ranking System Use...
Plan & Settings	sheet	Not approved	Not published	2022-11-23 10:40	Title Ranking System Use...
Title Titles Coloring	sheet	Not approved	Not published	2022-11-23 10:40	Title Ranking System Use...
My new sheet	sheet	Not approved	Not published	2022-11-23 10:40	Title Ranking System Use...
My new sheet (1)	sheet	Not approved	Not published	2022-11-23 10:40	Title Ranking System Use...
My new sheet (2)	sheet	Not approved	Not published	2022-11-23 10:40	Title Ranking System Use...
My new sheet (3)	sheet	Not approved	Not published	2022-11-23 10:40	Title Ranking System Use...
My new sheet (4)	sheet	Not approved	Not published	2022-11-23 10:40	Title Ranking System Use...
My new sheet (5)	sheet	Not approved	Not published	2022-11-23 10:40	Title Ranking System Use...
No BISAC 2	sheet	Not approved	Not published	2022-11-23 10:40	Title Ranking System Use...
Test Rank	sheet	Not approved	Not published	2022-11-23 10:40	Title Ranking System Use...
Title Ranks	sheet	Not approved	Not published	2022-11-23 13:48	Title Ranking System Use...
Title Ranks Overview	sheet	Not approved	Not published	2022-11-23 10:40	Title Ranking System Use...
Titles without BISAC Lev...	sheet	Not approved	Not published	2022-11-23 10:40	Title Ranking System Use...
WorkID Analysis	sheet	Not approved	Not published	2022-11-23 10:40	Title Ranking System Use...
Моділювання історії	story	Not approved	Not published	2022-11-23 10:40	Title Ranking System Use...

Рисунок 2.15 – вікно контролю додатку

На рисунку 2.15 можна побачити загальний приклад сортування елементів певного додатку у середовищі підключень й контролю версій. При створенні мешапу для опрацювання елементів більш доречно використовувати це вікно через наступні чинники:

- Стабільність;
- Зручність;
- Лаконічність.

Тепер більш детально розберемо кожний з чинників:

- Лаконічність – У даному вікні користувач може контролювати які типи інформації про елементи додатку йому потрібні для опрацювання.
- Зручність – Відсутність сторонніх елементів які б заважали на пряму контролювати інформацію, чи потребували б додаткових дій.
- Стабільність – Оскільки у середині додатку проходять математичні обчислення різної складності можуть виникати ситуації коли швидкодія додатку буде сильно знижена. Що вплине на можливість контролю елементів.

IDENTIFICATION

Name

App

EXECUTION

Enabled

Partial reload

Task session timeout (minutes)

Max retries

Triggers

Name	Type	Enabled
No records found		

Actions

- Create new once-only trigger
- Create new hourly trigger
- Create new daily trigger
- Create new weekly trigger
- Create new monthly trigger
- Create new scheduled trigger
- Create new task event trigger
- Edit
- Delete

TAGS

CUSTOM PROPERTIES

Custom

Рисунок 2.16 – Вікно контролю додатку призначене для оновлення даних додатку

Підсередовище для контролю даних – QMC середовища розробки Qlik Sense дозволяє запланувати автоматичне оновлення додатку у будь-який час. Усі процеси які виконувались під час запуску додатку записуються у лог-файли.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на кваліфікаційну бакалаврську роботу, реалізації підлягає програмне забезпечення, яке призначено для аналітично-інформаційної системи.

В процесі розробки кваліфікаційної бакалаврської роботи необхідно виконати наступний обсяг роботи:

- провести аналіз існуючих систем-аналогів для виявлення їх позитивних і

негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Що таке інформаційно-аналітичні системи

Інформаційно-аналітичні системи визначаються як особливий клас інформаційних систем, призначених для аналітичної обробки даних, а не для автоматизації повсякденної діяльності організації. Інформаційно-аналітичні системи об'єднують, аналізують і зберігають як єдине ціле інформацію, яка вилучена як з баз даних організації, так і із зовнішніх джерел. Сховища даних що входять до складу інформаційно-аналітичних систем, забезпечують перетворення великих об'ємів деталізованих даних в узагальнену вивірену інформацію, яка придатна для прийняття обґрунтованих рішень. На відміну від звичайних баз даних сховища містять оброблене, впорядковане і зрозуміле керівникам представлення даних.; вони стають складальним конвеєром по підготовці інформації в інтегрованому, несуперечливому, наочному вигляді для підтримки прийняття управлінських рішень.

Створення інформаційно-аналітичних систем, що реально відповідають цілям і завданням організації, визнається як достатньо складний процес, що включає етапи формування концепції, проектування, розробки, впровадження і супроводу. Сам характер цього процесу вимагає попередньої розробки фіксованої технологічної схеми. Технологічна схема відповідає стандарту, що описує процеси життєвого циклу програмних засобів, послідовність робіт і завдань, що виконуються певними виконавцями. Таким чином, необхідна загальна методика створення інформаційно-аналітичних систем, що містить склад і послідовність робіт і завдань, склад ролевих функцій і породжуваних артефактів (документів, моделей, схем і ін.).

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Технологія і методика створення інформаційно-аналітичних систем охоплює наступні види діяльності:

- збір, аналіз і деталізацію вимог до інформаційно-аналітичної системи, визначення пріоритетів реалізації цих вимог і постановка завдань по їх реалізації, визначення вимог по архітектурі, надійності і захисту від несанкціонованого доступу і визначення складу даних;

- розробка проектних рішень по всіх аспектах побудови інформаційно-аналітичної системи, визначення складу джерел інформації, способів передачі даних, складу додатків організації доступу до даних, проектування архітектури, проектування баз даних;

- розробка аналітичних додатків, вибір і настройка інструментальних засобів збору, перетворення і очищення даних і організації доступу користувачів до даним, розробка метаданих, тестування, розробка документації для користувачів.

Рекомендації по виконанню робіт і завдань включають рекомендації з питань збору вимог, ідентифікації джерел даних, витягання і перетворення даних для розміщення в сховище, створення тематичних вітрин даних, розробки регламентних звітів, застосування засобів поглибленого аналізу даних, питання зберігання "історичних" даних.

В умовах розбудови в Україні нової економічної, а також законодавчої системи, яка базується на ідеях демократичного розвинутого суспільства, виникає необхідність створення прогресивних інформаційно-аналітичних і прогнозних технологій підтримки прийняття управлінських рішень, потужних інформаційних ресурсних центрів, комплексних систем опрацювання інформації як одного з найважливіших чинників досягнення у країні рівня сталого розвитку суспільства за рахунок запровадження стратегій випереджаючого розвитку.

У 2001 році міжвідомча комісія прийняла перший етап великого проекту, який стосується створення інтегрованої інформаційно-аналітичної системи органів (НАС) державної влади та органів місцевого самоврядування України, який

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

виконувався в межах Національної програми інформатизації на замовлення Держкомзв'язку групою організацій на чолі з Інститутом кібернетики ім. В. М. Глушкова НАН України. При цьому основною підсистемою інформаційно-аналітичної системи органів влади та ПАС в цілому є підсистема формування, інтеграції та використання інформаційних ресурсів.

Інформаційно-аналітичне забезпечення у сфері соціального захисту населення ґрунтується на таких основних положеннях: створення, впровадження і супроводження Інтегрованої інформаційно-аналітичної системи в централізованому порядку з дотриманням діючих стандартів; створення типових засобів інформатизації та інформаційних технологій Інтегрованої інформаційно-аналітичної системи з дотриманням єдиних методологічних, технічних, інформаційних та організаційних вимог; побудова типових засобів інформатизації та інформаційних технологій Інтегрованої інформаційно-аналітичної системи за модульним принципом; оперативний обмін даними про осіб, які перебувають на обліку в органах праці та соціального захисту населення, з авторизованим доступом до цих даних суб'єктів Інтегрованої інформаційно-аналітичної системи; формування єдиного інформаційного середовища, розподіленого в Головному центрі інформаційних технологій, інформаційно-аналітичних центрах, органах праці та соціального захисту населення, підприємствах, установах, підпорядкованих центральному уповноваженому органу виконавчої влади з питань праці та соціальної політики.

Основні інформаційні ресурси Інтегрованої інформаційно-аналітичної системи складають: база даних одержувачів всіх видів соціальної допомоги; база даних застрахованих осіб та одержувачів пенсійних виплат; база даних осіб, які перебувають на обліку в службі зайнятості населення; база даних одержувачів соціальних послуг; база даних недержавних та громадських організацій, які надають населенню соціальні послуги; Єдиний державний автоматизований реєстр осіб, які мають право на пільги; Централізований банк даних з проблем

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

інвалідності; Централізована база даних страхових виплат фондів соціального страхування.

В умовах постійного підвищення рівня інформатизації суспільства суттєвого значення набуває й інформаційно-аналітичне забезпечення органів державної влади. Це об'єктивно вимагає всебічного використання в органах державної влади сучасних інформаційних технологій та створення інформаційно-аналітичних систем.

Інфраструктуру Інтегрованої інформаційно-аналітичної системи становлять: центральний уповноважений орган виконавчої влади з питань праці та соціальної політики; органи праці та соціального захисту населення Автономної Республіки Крим, обласних, Київської та Севастопольської міських державних адміністрацій, а також підпорядковані їм інформаційно-аналітичні центри; органи праці та соціального захисту населення районних, районних в містах державних адміністрацій, органів місцевого самоврядування, установи, підприємства, що входять до сфери управління центрального уповноваженого органу виконавчої влади з питань праці та соціального захисту населення; Фонд соціального захисту інвалідів, Фонд соціального страхування з тимчасової втрати працездатності, Фонд соціального страхування від нещасного випадку на виробництві та професійного захворювання України, Фонд соціального страхування України на випадок безробіття, а також їх регіональні відділення; Державна служба зайнятості; Головний центр інформаційних технологій; науково-дослідні установи, навчальні заклади, підприємства протезної галузі, що входять до сфери управління центрального уповноваженого органу виконавчої влади з питань праці та соціальної політики; типові засоби інформатизації та інформаційні технології Інтегрованої інформаційно-аналітичної системи.

Необхідною умовою забезпечення системного управління країною, узгоджених та цілеспрямованих дій усіх ланок державної влади є наявність в органах управління всеосяжної, цілком вірогідної, позбавленої суб'єктивізму інформації щодо конкретних питань. На рівні управління державою ця інформація

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

має відображати не тільки реальний стан справ, але й тенденції, масштаби та очікувані наслідки розвитку процесів життєдіяльності держави та світу на близьку та віддалену перспективу. Тобто питання формування інформаційної й аналітичної бази для прийняття управлінських рішень на державному рівні є особливо актуальним.

Створення інформаційних систем державної влади України почалося з перших років незалежності. Однак створення інформаційно-аналітичної системи в органах влади продовжувалося відокремлено, технічні рішення між системами не узгоджувались, мала місце відсутність координації діяльності організацій-розробників цих систем. Тому до Національної програми інформатизації 2000 р. було включено проект "Створення інтегрованої інформаційно-аналітичної системи органів державної влади й органів місцевого самоврядування України". На першому етапі його реалізації проведено дослідження стану формування інформаційно-аналітичної системи в органах державної влади. Спираючись на здобуті дані, розроблена Концепція формування й функціонування інтегрованої інформаційно-аналітичної системи органів влади й органів місцевого самоврядування. В основі матеріалів також розроблено технічне завдання на створення інтегрованої інформаційно-аналітичної системи. Окрім того, для забезпечення проведення робіт в майбутньому - на другому етапі (пілот-проект), котрий повинен охоплювати верхні органи влади країни, розроблено відповідне технічне завдання й ескізний проект. Загальний рівень інформатизації аналітичної діяльності в органах влади України такий, що, в цей час треба вирішувати проблеми прискорення суттєвої модернізації самих об'єктів інтеграції. Вдосконаленню діяльності перешкоджає ряд об'єктивних факторів, до яких належать й неточність, й неповнота інформації, з якою працюють органи державної влади, й обмеження реального часу, на протязі якого мають бути прийняті управлінські рішення, та багатокритеріальність при прийнятті управлінських рішень, й відсутність належної культури виконання й використання інформаційно-аналітичних матеріалів. Проблеми, які виникають при створенні інтегрованої інформаційно-аналітичної

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

системи, підлягають вирішенню на найвищому державному рівні. Пріоритетними напрямами державної політики України відносно створення інтегрованої інформаційно-аналітичної системи є нормативно-правове забезпечення її створення й функціонування, координація створення та модернізація інтегрованої інформаційно-аналітичної системи органів державної влади, забезпечення використання Інтернету й розвиток "електронної демократії", забезпечення інформаційної безпеки всіх учасників інформаційних процесів в державі, сприяння в підготовці наукових та інженерно-технічних кадрів для розробки, експлуатації й розвитку систем інформаційно-аналітичної діяльності.

Проект НАС визначав інфраструктуру, стандарти, нормативно-методичне забезпечення цієї системи.

Загальною задачею є забезпечення умов як для прийняття зважених, узгоджених та ефективних рішень на всіх рівнях органів державної влади, так і для реалізації широких зв'язків владних структур з громадськістю. До напрямків функціональної інтеграції в інтегрованій інформаційно-аналітичній системі належать, насамперед, електронізація документообігу, а також процеси збору, первинної обробки інформації, введення та поповнення її джерел. Основним джерелом інформації для органів державної влади є звітність суб'єктів економічної діяльності та інша інформація, яку подають громадяни та підприємства до державних підприємств, тому вона повинна переводитися в електронну форму. Інтегрована інформаційно-аналітична система є багаторівневою розгалуженою глобальною організаційно-технічною системою, що будується за територіальними, галузевими та функціональними принципами. Основними територіальними й галузевими структурними одиницями інтегрованої інформаційно-аналітичної системи виступають інформаційно-аналітичні системи відповідних органів влади, які організаційно створюються ними ж, а також Центр управління Інтегрованою інформаційно-аналітичною системою. Основними інтеграційними складовими інтегрованої інформаційно-аналітичної системи є телекомунікаційне середовище, інтегрована система управління інформаційними ресурсами, інтегрована система

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

електронного документообігу, система управління аналітичною обробкою інформації, система зовнішніх зв'язків, система захисту інформації, нормативно-правове забезпечення. Функціональними складовими інтегрованої інформаційно-аналітичної системи є інформаційно-аналітичні центри (ІАЦ) інформаційно-аналітичні системи (ІАС), які репрезентують в інтегрованій інформаційно-аналітичній системі відповідний орган влади ІАЦ повинні збирати оперативну інформацію, вести нормативну документацію й класифікатори, проводить оперативний та ретроспективний аналіз інформації, захист даних та технологічну підтримку діяльності центру. Інтеграційно-комунікаційні властивості інформаційно-аналітичної системи в ІАЦ реалізують інтеграційно-комунікаційний компонент (ІКК), котрий забезпечує інтеграцію в інтегровану інформаційно-аналітичну систему. Ряд важливих задач відводиться Центру управління інтегрованої інформаційно-аналітичної системи, серед яких - управління доступом розгалуженого банку даних державних інформаційних ресурсів, управління обміном даними й електронним документообігом між ІАС, розробка єдиних методологічних основ, технологій аналітичних обчислювань й виконання аналітичних дослідів за міжгалузевими напрямками, управління й підтримка телекомунікаційного середовища й системи захисту інформації інтегрованої інформаційно-аналітичної системи. Загальні основи нормативно-правового забезпечення функціонування інтегрованої інформаційно-аналітичної системи, повинні складати законодавство у сфері громадських інформаційних відносин. Головними проблемами в галузі електронного документообігу є відсутність повної та достатньої системи державної стандартизації цієї ділянки, а також нормативного впровадження та експлуатації систем електронного документообігу.

Основними вимогами до єдиного телекомунікаційного середовища інтегрованої інформаційно-аналітичної системи, є забезпечення достатніх пропускних можливостей мережі та її дієздатності в умовах впливу різноманітних дестабілізуючих факторів. У зв'язку з високим державним статусом операцій по обміну інформацією в інтегрованій інформаційно-аналітичній системі набирає

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

особливу вагу інформаційна безпека та захист інформації. Проблема забезпечення інформаційної безпеки інтегрованої інформаційно-аналітичної системи є комплексною, та для її розв'язання необхідно об'єднати законодавчі, адміністративні, організаційні й програмно-технічні заходи.

Складність та масштабність проблеми впровадження інтегрованої інформаційно-аналітичної системи є безпрецедентною для України. Тому цей процес здійснюється поетапно, з вирішенням питань відносно вдосконалення та модернізації вже існуючих ІАС та їх інтеграції в інтегрованій інформаційно-аналітичній системі. Важливим початковим етапом створення НАС являється виконання пілот-проекту НАС за участі обраних органів державної влади центрального та регіонального рівня. Подальше формування та впровадження ПАС полягає в типізації проектних рішень для усіх рівнів ІАС галузевого, регіонального та функціонального напрямку й узгодженості та синхронізації впровадження ІАС. Реалізація ПАС може забезпечити якісно новий рівень державного управління завдяки доступності національних та світових інформаційних ресурсів для використання, оперативності, достовірності, повноти, важливості й своєчасності отриманої інформації для прийняття рішень, незалежно від складності вирішуваних питань та об'ємів обробленої інформації, підвищення понятійності та управлінської культури в функціонуванні органів державної влади.

Інформаційні ресурси Інтегрованої інформаційно-аналітичної системи формують Головний центр інформаційних технологій, інформаційно-аналітичні центри, місцеві органи праці та соціального захисту населення на підставі інформації, що: належить центральному уповноваженому органу виконавчої влади з питань праці та соціальної політики, підпорядкованим йому органам праці та соціального захисту населення, підприємствам, установам; одержана від органів виконавчої влади, органів місцевого самоврядування; одержана від осіб, які перебувають на обліку в органах праці та соціального захисту населення, або їх законних представників; одержана з інших джерел, які не суперечать чинному законодавству.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

Для формування і підтримки в актуальному стані інформаційних ресурсів органи виконавчої влади, органи місцевого самоврядування зобов'язані безкоштовно надавати Головному центру інформаційних технологій, інформаційно-аналітичним центрам, органам праці та соціального захисту населення в електронній формі документи й відомості, які в обов'язковому порядку супроводжують електронними підписами відповідних посадових осіб. Головний центр інформаційних технологій, інформаційно-аналітичні центри, органи праці та соціального захисту населення мають право перевіряти достовірність отриманої інформації.

Отже, пріоритетами Національної програми інформатизації є створення і розвиток інтегрованих систем інформаційно-аналітичного забезпечення органів державної влади, систем захисту інформації, створення системи національних інформаційних ресурсів, інтеграція web-сайтів органів влади до Урядового порталу як основи електронного уряду, розвиток української мережі Інтернет, удосконалення організаційного, нормативно-правового забезпечення процесу побудови інформаційного суспільства.

У суспільстві функціонують декілька рівнів соціально-економічного життя, які пов'язані між собою інформацією. Результатом взаємодії є прийняття різного роду управлінських рішень. У сфері науково-технічної діяльності формуються відповідні плани досліджень, приймаються науково-технічні, національні, міжгалузеві та галузеві програми, бізнес-проекти, бізнес-технології, відбувається інвестування тощо.

Стратегічною метою управління інформацією є вихід через прогнозно-аналітичну функцію на управління іншими ресурсами, зокрема трудовими, фінансовими і матеріальними. Саме тому роль держави в управлінні інформаційними ресурсами, як замовника і координатора, є вирішальним.

Маючи інтегровані інформаційні ресурси національного значення, можна робити аналіз і прогноз, розробити варіанти управлінських рішень. Національні інтегровані ресурси працюють і забезпечують верхній рівень управління -

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

глобальні економічні і соціально-політичні процеси. Але важливими є також інформаційні потоки, які створюються окремими фірмами, підприємствами, установами, тобто розгалужений інформаційний ресурс. А в цілому процес накопичення та використання інформаційного ресурсу відноситься у більшій мірі до стадії управління знаннями.

Управління знаннями - це система яка базується на інформаційно-аналітичному забезпеченні інтелектуального управління фірмами, підприємствами, установами тощо. Управління знаннями - це й освіта, як окремих осіб, так і організація навчання колективу, що забезпечує перехід до інтелектуального управління.

Формування інтелектуального управління, яке базується на інформаційно-аналітичному забезпеченні, передбачає не тільки більш повне використання інформації в діяльності фірм, але й зміну поглядів на її зміст і засоби переробки. Інформація перестає бути допоміжною, обслуговуючою і вже розглядається не тільки як середовище, в якому здійснюється діяльність фірми, а стає стратегічним ресурсом, що дозволяє вдало обґрунтувати і вибрати вектор розвитку як у поточному періоді, так і на майбутнє. Інформація стає основою, базою для виробничої функції фірми.

Плідна інформаційно-аналітична діяльність неможлива без нормативних документів, що регламентують технологію створення державними органами інформаційних ресурсів.

У результаті здійснення робіт по створенню розгалуженої інформаційно-аналітичної системи, які є складовою частиною Національної програми інформатизації України, передбачається досягти вирішення проблем:

- створення уніфікованої технології отримання інформаційно-аналітичної обробки і накопичення даних та розробки єдиної системи нормативних документів, що регламентує її функціонування;

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

– створення оперативного і повноцінного доступу користувачів до інформації національного та галузевого фондів стандартів та нормативних документів;

– актуалізації, оптимізації формування, зберігання, захисту та використання інформаційних ресурсів;

– входження до міжнародного інформаційного простору на основі єдиних наукових, організаційних, методологічних, технічних та технологічних засад.

Система підтримки прийняття рішень для складних соціальних об'єктів на базі технології ситуаційних оцінок створюється на науково-технологічних засадах моніторингу трансформаційних процесів соціальної системи, аналізу отриманої інформації та розробки вірогідних прогнозних сценаріїв розвитку.

Концептуальна модель інформаційно-аналітичної та прогнозної підтримки прийняття рішень системного геополітичного й економічного моніторингу включає такі етапи: проектування дослідження, інформаційно-аналітичне моделювання та створення прогнозних оцінок і сценаріїв, генерування підсумкових документів і пропозицій.

Інформаційний етап включає збір і опрацювання отриманої інформації за допомогою сучасних інформаційних технологій, перевірку достовірності отриманих даних, узагальнення експертних оцінок, ведення баз даних, формування багатофункціональних інформаційних середовищ, створення авторських програм і систем, формування розгалуженого середовища інформаційних ресурсних центрів.

Передбачається, що на цьому етапі користувач має можливість одержувати основні результати роботи у вигляді матеріалів стандартної форми яка використовується у світовій практиці при запровадженні стандартів сучасного менеджменту та підтримки організаційних процесів і роботи у інформаційно-комунікаційних мережах.

Аналітичний етап включає генерування масиву актуальної інформації для подальшого аналізу, вимір подій, рівня сил і впливів основних учасників подій, виявлення найбільш гострих і актуальних проблем та конфліктів у заданому

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

регіоні; генерацію матеріалів для політичного і економічного прогнозування (визначення найбільш імовірних шляхів розвитку ситуації в регіоні), а також оцінку рівнів прояву небажаних подій у політичній та економічній сферах. До виконання досліджень на цьому етапі залучаються висококваліфіковані експертні групи, провідні фахівці, представники апарату управління, досвідчені політики. Це дозволяє досягти комплексності при підготовці інформації для прийняття та ухвалення рішень. Передбачено також застосування загальноприйнятих методик відбирання експертів і оцінки їх компетентності.

Аналітична стадія роботи закінчується обчисленнями, які дозволяють одержати ймовірні оцінки, ступені ризику розвитку ситуації та прийняття відповідних рішень за напрямками діяльності відповідних державних установ.

Концептуальна модель інформаційно-аналітичної і прогнозної підтримки прийняття рішень реалізується в автоматизованій системі, яка базується на принципах системного підходу до аналізу соціально-політичних ситуацій.

Розробка та освоєння новітніх інформаційно-аналітичних та прогнозних технологій у сфері державного управління забезпечить вагомий внесок у процес формування менеджменту на вищому рівні прийняття рішень. У разі несвоєчасного одержання, опрацювання й аналізу інформації та розробки на її основі прогнозних оцінок у системі державного управління - мають місце величезні збитки та витрати. Тому необхідно просуватися в напрямі розробки й освоєння нових інформаційно-аналітичних технологій, щоб уникнути небажаних наслідків від неефективних рішень на державному рівні.

Принцип побудови інкрементально-незалежної системи

Сутність даної методики яку мені вбрело в голову спробувати реалізувати, при чому успішно, полягає у тому, що нам потрібно мати певний обсяг завантажених даних. Враховуючи розміри BigData деякі процеси можуть займати не оду годину завантаження, і якщо під час завантаження вирублять світло, то можна усе буде починати робити заново.

Тож перша частина моєї методики складається з дефрагментації додатка на

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

декілька менших для послідовного виконання завантаження, трансформації та відображення даних. Це дозволяє окрім незалежної роботи знизити витрати часу при неправильному завантаженні даних, разом з тим це також дозволяє чистим користувачам відображувати роботу лише одного доданка у якому немає ключових змін даних, що запобігає їх спотворенню через незграбність. Таким чином не надавши доступ на перезапис ми збережемо інформацію від людського фактору.

Другою частиною моєї методики є внесення загального ліміту завантажених даних за допомогою обмежень за часовою шкалою, яка впливає на загальний час та обсяг даних з якими буде працювати інформаційно-аналітична система. Також на основі часового проміжку створюються унікальні періоди на основі яких відбувається інкрементальне збереження даних. Та звичайно ж наявна перевірка факту їх існування, це використовується для ігнорування повторного завантаження ідентичних даних у бази даних, та допомагає виловити «втрачені» архіви.

3.2 Розробка структурної схеми

Структурна схема розробленої системи зображена на рисунку 3.1. На ній показано теоретичну структуру мого ПЗ у завершеному вигляді.

Основним елементом будь-якої аналітично-інформаційної системи є схема взаємодії «База даних – Обробник даних – Користувач». На основі цієї схеми можуть бути побудовані системи практично будь-якої складності з більшою кількістю елементів різних типів.

Основною ж задачею аналітично-інформаційної системи є швидка та точна видача аналітичної інформації, при цьому при збільшенні обсягів даних, які допомагають підвищити точність та обсяг аналітики, також підвищується затрачений час на цей обробіток даних. Тож перед розробником полягає задача у визначенні якості зв'язки між таблицями даних, та розробці методів які б змогли надати максимально точну інформацію за найкоротший час, при цьому

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

активізувавши якомога менше ресурсів, які в свою чергу при постійній активності крім аналізу даних використовуються на зборі та видачі різних типів інформації користувачам.

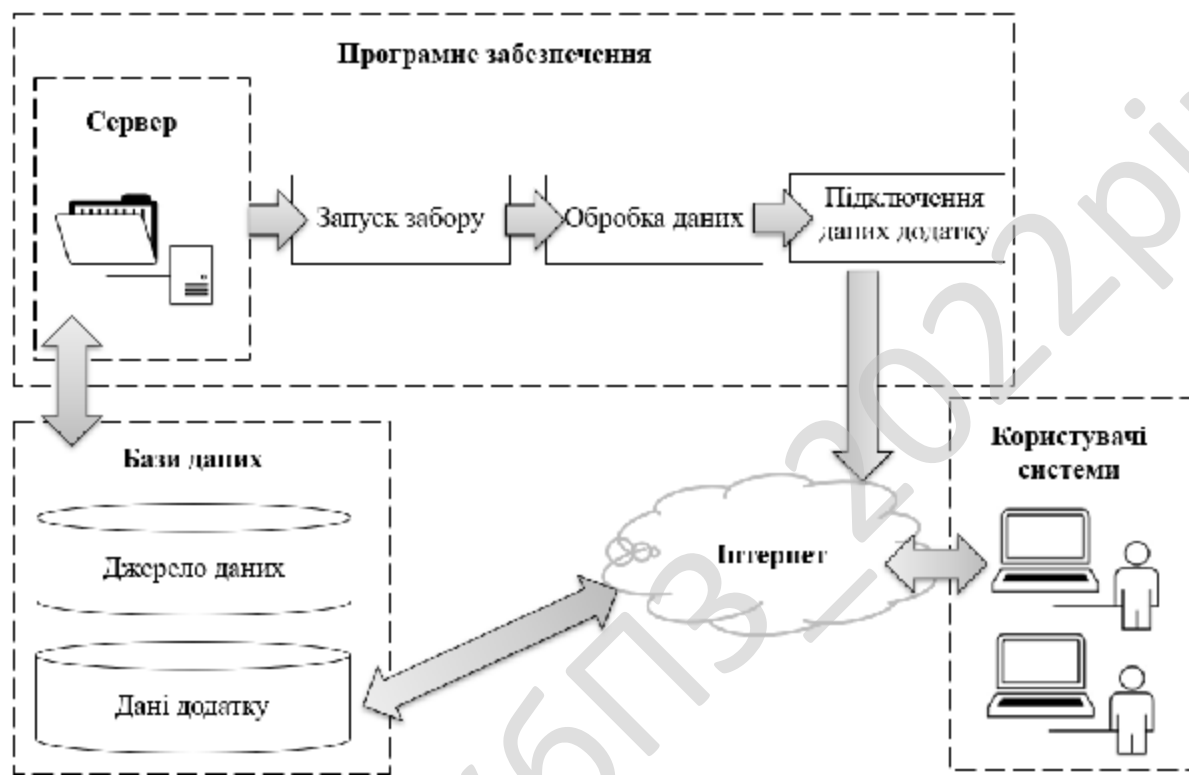


Рисунок 3.1 – Структурна схема системи

Розробник повинен визначити які зв'язки повинні бути використані між елементами моделі, наскільки сильно певні зв'язки повинні бути не сумісні з іншими, а які при своїй несумісності через певні збіжності все ж таки повинні знаходитись поруч. Крім того в залежності від типу, функціоналу системи у який вбудована аналітично-інформаційна система до особливостей виводу даних можна віднести тип даних, та час у який ту чи іншу тематику даних виводити.

Також в залежності від того наскільки пророблено ту чи іншу систему якість аналітичної інформації буде варіюватися через діяльність користувачів.

В залежності від активності системи у неї можуть бути різні терміни оновлення/додавання/вилучення даних з баз даних, тож при розробці системи потрібно врахувати те як буде поводити себе система підчас паралельної роботи та

оновлення даних, та як вона буде справлятися з наскоюванням певних даних на старі дані чи при одночасному наскоюванні.

Також треба брати до уваги сумісність розширень й коректність їх версій на різних серверах.

3.3 Розробка функціональної схеми

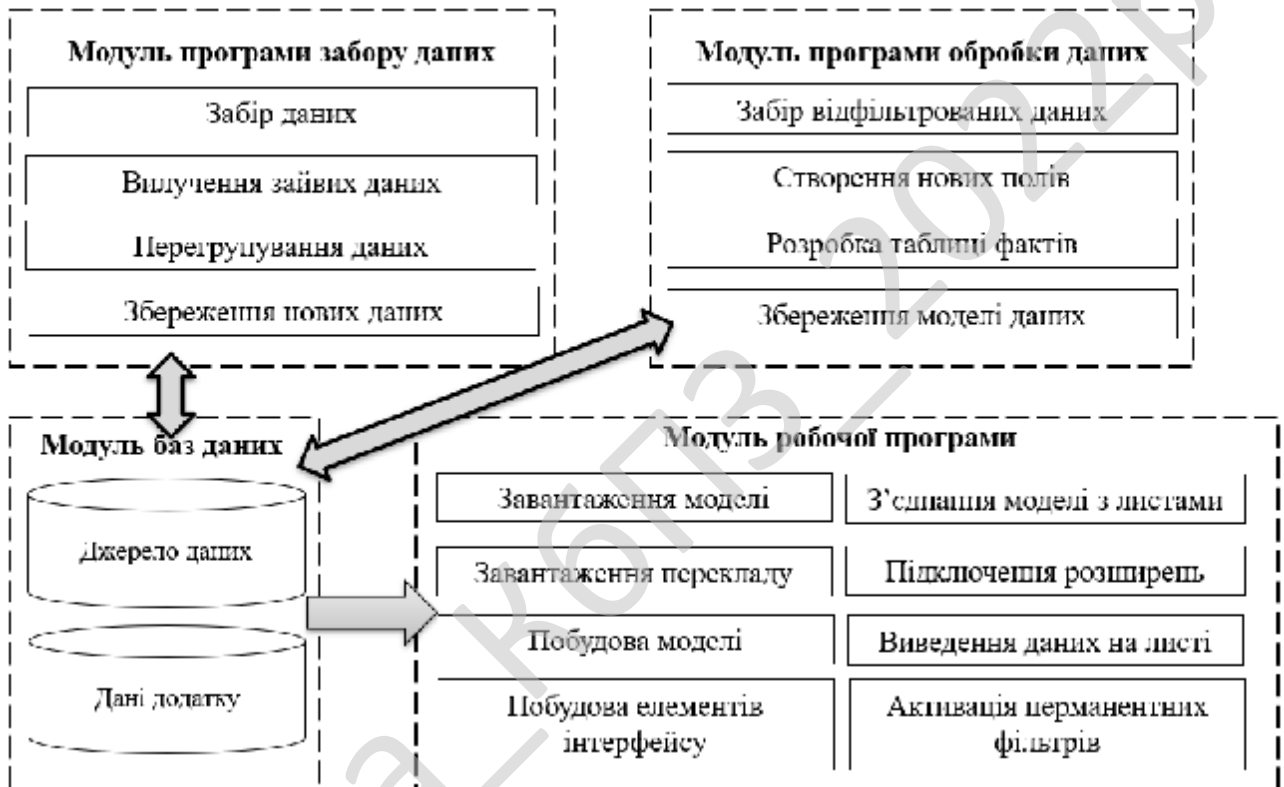


Рисунок 3.2 – Функціональна схема програми

Функціональна схема розробленої системи зображена на рисунку 3.2.

З рисунку видно, що розроблена система складається з наступних частин:

1. Модулів програми:

- Модуля програми для забору сирих даних;
- Модуля оброки даних;
- Модуля кінцевого додатку.

2. Модуля баз даних.

Розглянемо більш детально кожний з блоків розробленої програми.

У модулі баз даних спочатку у нас знаходяться джерела - бази даних, з їх допомогою ми можемо побудувати файли даних які буде використовувати додаток.

До модуля програми для забору сирих даних відносяться початкові бази даних, які ми можемо обробити і отримати відфільтровані файли для роботи інших додатків.

До модуля програми для обробітку відфільтрованих даних відносяться створені нами бази даних, які ми обробляємо та будуємо між ними коректні зв'язки. Після чого вони зберігаються як окремі таблиці.

Модуль кінцевої програми забирає дані які обробив модуль обробки даних та частину інформації яку обробив модуль забору даних. Будує та корегує таблицю фактів на основі якої будується модель даних. Також підключаються таблиці перекладів за допомогою яких будуються певні елементи інтерфейсу. При коректній роботі моделей стає можливою робота над листами з аналітично-інформаційною інформацією.

3.4 Розробка діаграми процесів

Розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей. Діаграма процесів розробленої системи зображена на рисунку 3.3.

Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

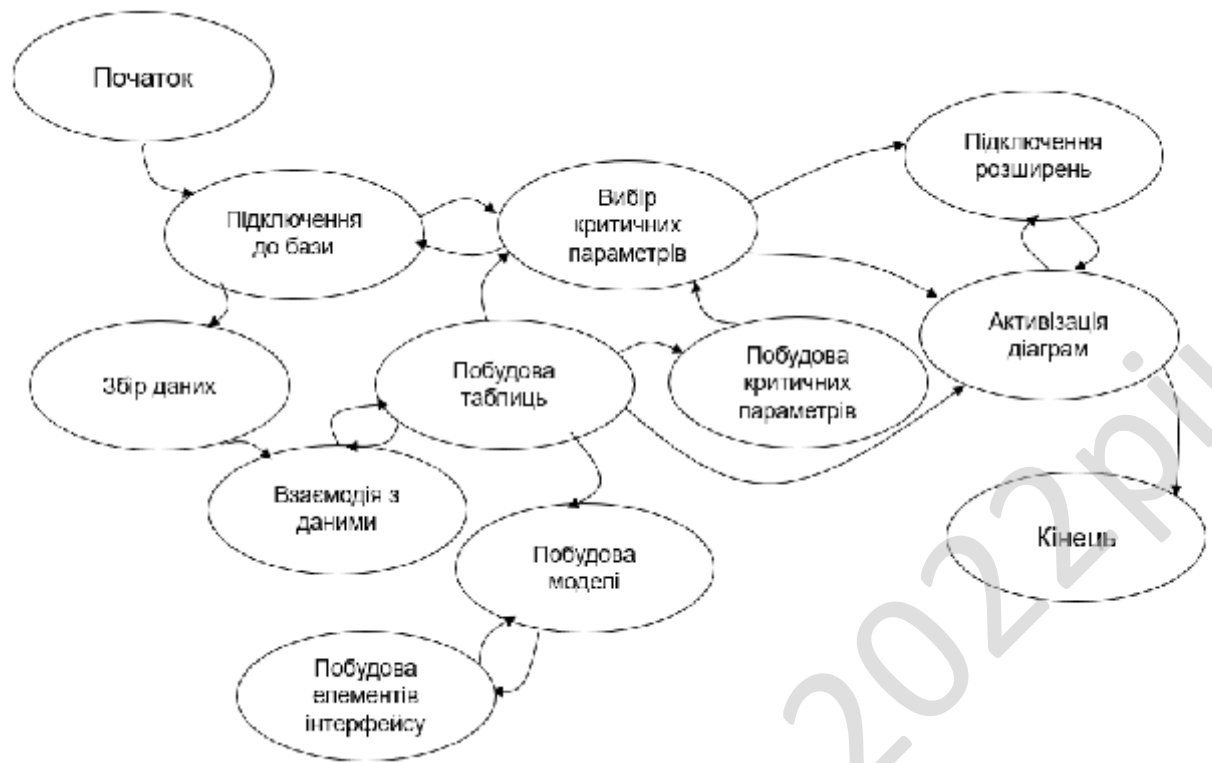


Рисунок 3.3 – Діаграма взаємодії процесів

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

Діаграми потоків даних містять три типи елементів:

- Зовнішні по відношенню до системи сутності;
- Процеси які являють собою трансформацію даних в рамках описуваної системи;
- Бази даних.

При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо алгоритм роботи основної програми. Його блок-схема зображена на рисунку 4.1.



Рисунок 4.1 – Блок-схема роботи програмної частини

З рисунку 4.1 та програмного коду наведеного нище видно, що під час запуску програми відбувається безліч заватажувальних процесів.

```
SET ThousandSep=' ';
SET DecimalSep=',';
SET MoneyThousandSep=' ';
SET MoneyDecimalSep=',';
SET MoneyFormat='# ##0,00€;-# ##0,00€';
SET TimeFormat='h:mm:ss';
SET DateFormat='DD.MM.YYYY';
SET TimestampFormat='DD.MM.YYYY h:mm:ss[.fff]';
SET DayNames='Пн;Вт;Ср;Чт;Пт;Сб;Нд';
SET LongMonthNames=
'Sічень;Лютий;Березень;Квітень;Травень;Червень;Липень;Серпень;Вересень;Жовтень;Лис
топад;Грудень';
SET LongDayNames='понеділок;вівторок;серeda;четвер;п'ятниця;субота;неділя';
SET FirstWeekDay=0;
SET BrokenWeeks=0;
SET ReferenceDay=0;
SET FirstMonthOfYear=1;
SET CollationLocale='uk-UA';

SET HidePrefix = '_';
SET ShortDateFormat = 'DD.MM.YY';
LET vLoadTime = now(1);
Товар:
LOAD
    Nomenclature_KEY,
    "Product ID",
    "Код товару",
    Товар,
    Товар as Product,
    "Код товару" & ' | ' & Товар as ID_Product,
    "Код товару" & ' | ' & Товар as Код_Товару,
    if (not IsNull("Product hierarchy 1"), "Product hierarchy 1", '_Product
hierarchy 1 not specified_') as "Product hierarchy 1",
    if (not IsNull("Product hierarchy 1"), "Product hierarchy 1", '_Ієрархія
товару 1 не вказана_') as "Ієрархія товару 1",
    if (not IsNull("Product hierarchy 2"), "Product hierarchy 2", '_Product
hierarchy 2 not specified_') as "Product hierarchy 2",
    if (not IsNull("Product hierarchy 2"), "Product hierarchy 2", '_Ієрархія
```

									ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата						51

```

товару 2 не вказана_') as "Ієрархія товару 2",
    if (not IsNull("Ієрархія товару 3"), "Ієрархія товару 3", '_Product
hierarchy 3 not specified_') as "Product hierarchy 3",
    if (not IsNull("Ієрархія товару 3"), "Ієрархія товару 3", '_Ієрархія
товару 3 не вказана_') as "Ієрархія товару 3",
    if (IsNull("Номенклатурна група"), '_Номенклатурна група не вказана_',
"Номенклатурна група") as "Номенклатурна група",
    if (IsNull("Номенклатурна група"), '_Nomenclature group not specified_',
"Номенклатурна група") as "Nomenclature group",
    if (not IsNull("Група товару"), "Група товару", '_Група товару не
вказана_') as "Група товару",
    if (not IsNull("Група товару"), "Група товару", '_Product group не
указана_') as "Product group",
    if (not IsNull("Вид товару"), "Вид товару", '_Вид товару не вказано_')
as "Вид товару",
    if (not IsNull("Вид товару"), "Вид товару ", '_Product type not
specified_') as "Product kind",
    if (not IsNull("Країна товару"), "Країна товару", '_Product country not
specified_') as "Product country",
    if (not IsNull("Країна товару"), "Країна товару", '_Країна товару не
вказана_') as "Країна товару",
    if (IsNull("Постачальник"), '_Supplier not listed_', "Постачальник") as
"Supplier",
    if (IsNull("Постачальник"), '_Постачальник не вказаний_',
"Постачальник") as "Постачальник"
FROM [$ (vQVD_D)/Товар.qvd] (qvd)
where not IsNull("Товар");

```

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52


```

NoConcatenate
LOAD
    id as _TranslationId, *
RESIDENT Language;

DROP TABLE Language;

LET T = replace('_T_$_1_@(=MaxString({$_} _Language))', '@', '$');

_Languages:
LOAD Concat(distinct _Language, ',') as _Languages
RESIDENT Translations;

LET vLanguages = chr(39) & replace(peek('_Languages'), ',', chr(39)
&', '&chr(39)) & chr(39);

DROP TABLE _Languages;

LET vTranslationsRows = NoOfRows('Translations');

FOR i = 0 TO vTranslationsRows - 1
    LET _vTranslationId = Peek('_TranslationId', i, 'Translations');
    LET _vTranslation = Peek('_Translation', i, 'Translations');
    LET _vTrLanguage = Peek('_Language', i, 'Translations');

    LET _vTrName = '_T_$_(_vTranslationId)_$_(_vTrLanguage)';
    LET $_(_vTrName) = '$(_vTranslation)';

    // TRACE $_(_vTrName);

    LET _vTranslationId =;
    LET _vTranslation =;
    LET _vTrLanguage =;
    LET _vTrName =;
NEXT

```

Звдяки якій програма буде модель інтерфейсних елементів (рисунок 4.3).

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

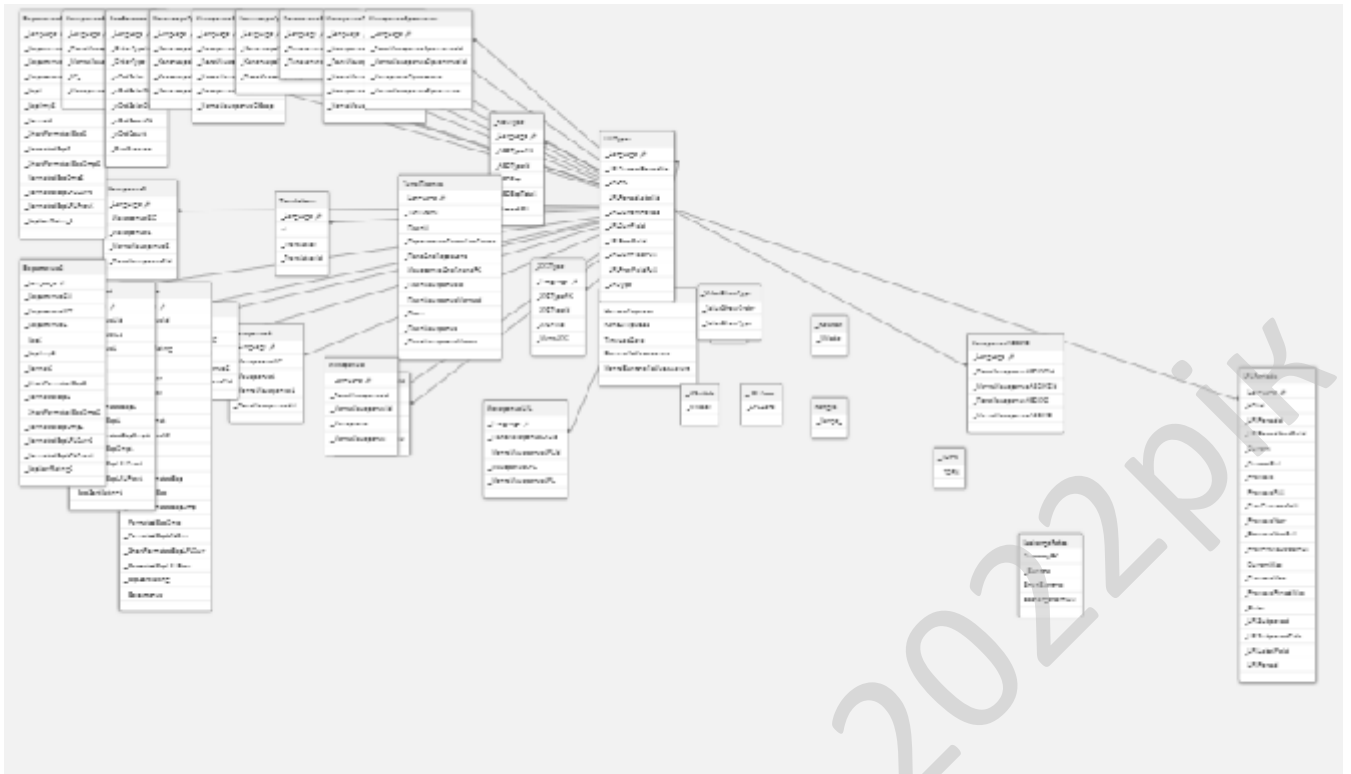


Рисунок 4.3 – Модель інтерфейсу

Нижче мною було описано принцип роботи коду, який відповідає за те, щоб моє ПЗ побудувало рекомендацію (рисунок 4.4).

У першому додатку ми проводимо завантаження сирих даних з джерела даних та їх обробку та фільтрування. Під час цього процесу створюються критичні параметри такі як: мова за замовчуванням, періоди календаря та валюта за замовчуванням. На основі критичних параметрів відбувається фільтрація частини даних, які завантажуються. Після обробки додатком, додаток займається зберіганням таблиць у зручному для середовища форматі.

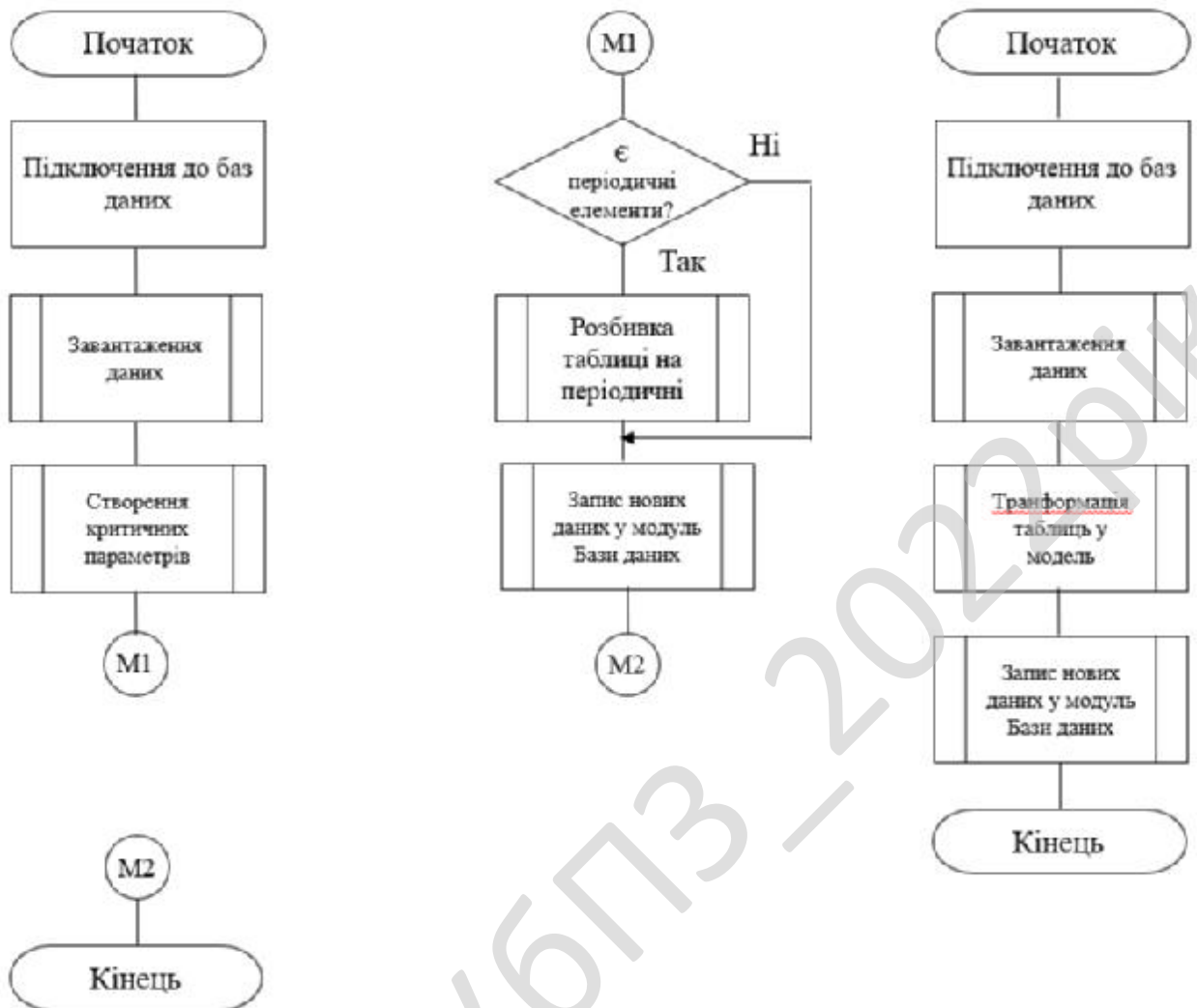


Рисунок 4.4 – Блок-схема роботи програм для забору та обробки даних.

Другий додаток виконує функцію обробника даних. Він завантажує разом з відфільтрованими даними й критичні параметри. Сам додаток проводить безліч конкатенацій одних полів, кон'юнкцію інших, та за деякими випадками диз'юнкцію. За допомогою таких дій будується модель даних. Окрім побудови моделі даних також проводиться створення інтерфейсу багатомовності

Translations:

```

NoConcatenate
load "Index",
Reference,
UA,
EN
  
```

```

FROM [$(vQVD_D)/Translation.qvd] (qvd);

Language:
CROSSTABLE (Language, Translation) LOAD
    Index,
    UA,
    EN
RESIDENT Translations;

_Languages:
LOAD Concat(distinct Language, ',') as _Languages
RESIDENT Language;

LET vLanguages = chr(39) & replace(peek('_Languages'), ',', chr(39)
&', '&chr(39)) & chr(39);

DROP TABLE _Languages;

LanguageList:
Load distinct _Language,
    AutoNumber(_Language) AS Index_
;
LOAD Distinct
    Language as _Language
RESIDENT Language
WHERE Language > 0;
DROP TABLE Translations; //drops after multilang calendar creation
LET T = replace('_T_$(=FirstSortedValue(distinct _Language, Index_))',
'@', '$');
LET vTranslationsRows = NoOfRows('Language');
FOR i = 0 TO vTranslationsRows - 1
    LET _vTranslationId = Peek('Index', i, 'Language');
    LET _vTranslation = Peek('Translation', i, 'Language');
    LET _vTrLanguage = Peek('Language', i, 'Language');

    LET _vTrName = '_T_$_vTranslationId_$_vTrLanguage';
    LET $_vTrName = '$(_vTranslation)';
    LET _vTranslationId =;
    LET _vTranslation =;
    LET _vTrLanguage =;
    LET _vTrName =;
NEXT;

```

					БКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

```

SET NumConvertUA = "if($1 > pow(10, 15), '> 1 квлн', Pick(Floor(LOG10($1))+1,
NUM($1, '# ##0'), NUM($1, '# ##0'), NUM($1, '# ##0'), NUM($1/1000, '# ##0,00 тис',
',', ' '), NUM($1/1000, '# ##0,00 тис', ',', ' '), NUM($1/1000, '# ##0,00 тис', ',',
' '), NUM($1/1000000, '# ##0,00 млн', ',', ' '), NUM($1/1000000, '# ##0,00 млн',
',', ' '), NUM($1/1000000, '# ##0,00 млн', ',', ' '), NUM($1/1000000000, '# ##0,00
млрд', ',', ' '), NUM($1/1000000000, '# ##0,00 млрд', ',', ' '), NUM($1/1000000000,
'# ##0,00 млрд', ',', ' '), NUM($1/1000000000000, '# ##0,00 трлн', ',', ' '),
NUM($1/1000000000000, '# ##0,00 трлн', ',', ' '), NUM($1/1000000000000, '# ##000,00
трлн', ',', ' '), NUM($1/1000000000000000, '# ##0,00 квлн', ',', ' '),
NUM($1/1000000000000000, '# ##0,00 квлн', ',', ' '), NUM($1/1000000000000000, '#
##0,00 квлн', ',', ' '))";

```

```

SET NumConvertEN = "if($1 > pow(10, 15), '> 1 qdln', Pick(Floor(LOG10($1))+1,
NUM($1, '# ##0'), NUM($1, '# ##0'), NUM($1, '# ##0'), NUM($1/1000, '# ##0,00 k',
',', ' '), NUM($1/1000, '# ##0,00 k', ',', ' '), NUM($1/1000, '# ##0,00 k', ',', '
'), NUM($1/1000000, '# ##0,00 mln', ',', ' '), NUM($1/1000000, '# ##0,00 mln', ',',
' '), NUM($1/1000000, '# ##0,00 mln', ',', ' '), NUM($1/1000000000, '# ##0,00 bln',
',', ' '), NUM($1/1000000000, '# ##0,00 bln', ',', ' '), NUM($1/1000000000, '# ##0,00
bln', ',', ' '), NUM($1/1000000000000, '# ##0,00 trln', ',', ' '),
NUM($1/1000000000000, '# ##0,00 trln', ',', ' '), NUM($1/1000000000000, '# ##000,00
trln', ',', ' '), NUM($1/1000000000000000, '# ##0,00 qdln', ',', ' '),
NUM($1/1000000000000000, '# ##0,00 qdln', ',', ' '), NUM($1/1000000000000000, '#
##0,00 qdln', ',', ' '))";

```

```

STORE [Language] INTO
[$(vQVD_D)Language.qvd] (qvd);
STORE [LanguageList] INTO
[$(vQVD_D)LanguageList.qvd] (qvd);

Drop Tables Language, LanguageList;

```

4.2 Захист розробленого програмного забезпечення

Механізм захисту Qlik може бути заданий двома різними способами: Він може бути вбудований у скрипт документа Qlik, або заданий за допомогою використання Qlik Publisher. Елементами захисту ПЗ є такі пункти:

1) Перевірка справжності та авторизація

Перевірка справжності – це процес, у якому підтверджується особистість

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

людини. Qlik може або дозволити ОС Windows провести автентифікацію, або запросити ідентифікатор користувача та пароль (відмінні від ідентифікатора та пароль користувача Windows), або використовувати ліцензійний ключ Qlik як метод простої автентифікації.

Авторизація - це перевірка наявності у вже ідентифікованої людини прав на використання ресурсу. Qlik може або дозволити Windows провести авторизацію, або провести авторизацію самостійно. В останньому випадку в скрипт має бути вбудована таблиця безпеки.

2) Захист за допомогою Qlik Publisher

Якщо Qlik Publisher налаштований на обробку безпеки, то кожен файл Qlik буде розділений на кілька файлів, кожен з яких міститиме дані, що стосуються певного користувача або групи користувачів. Дані файли будуть зберігатися в папках з поточними налаштуваннями безпеки ОС, тобто Qlik дозволяє операційній системі провести автентифікацію та авторизацію.

Однак у сам файл захист не вбудований, тому на завантаженому файлі захист відсутній.

Розміри файлів, як правило, будуть меншими, оскільки один файл буде розділений на декілька, користувач відкриває файл тільки зі своїми даними. Однак це також означає, що сервер Qlik може використовувати більше пам'яті, ніж якби всі дані знаходилися в одному файлі, оскільки іноді завантажуються кілька файлів, що містять однакові дані.

3) Захист за допомогою доступу до секції у скрипті Qlik

Якщо Section Access у скрипті Qlik налаштований на обробку безпеки, один файл може містити дані для декількох користувачів або груп користувачів. Qlik буде використовувати інформацію Section Access для автентифікації та авторизації, а також для динамічного скорочення даних, щоб користувач міг бачити тільки свої дані.

Захист вбудований у сам файл, тому навіть завантажений файл певною мірою захищений. Однак, якщо потреби захисту високі, слід запобігати

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

завантаженню файлів та використання в автономному режимі. Файли повинні публікуватися лише з Qlik Server.

Оскільки всі дані знаходяться у файлі, розмір файлу може бути дуже великим.

Документи Qlik можна зробити невидимими в автономному режимі. Щоб зробити невидимим документ користувача, який знаходиться в автономному режимі, додайте наступний атрибут до розділу відомостей про документ за допомогою QMC:

Ім'я: True

Значення: True

Вся інформація, вказана нижче, відноситься до методу захисту, який використовує Section Access у скрипті Qlik.

4) Розділи скрипту

Керування доступом на рівні рядка здійснюється за допомогою однієї або декількох таблиць безпеки, завантажених так само, як програма зазвичай завантажує дані. Таким чином, можна зберігати ці таблиці у звичайній базі даних або електронній таблиці. Оператори скрипта, що управляють таблицями безпеки, дано у розділі авторизації, який у скрипті запускається оператором Section Access.

Якщо в скрипті визначено розділ авторизації, то частина скрипту, що завантажує «звичайні» дані, має бути поміщена до іншого розділу, який запускає оператор Section Application.

5) Рівні доступу до секції

Доступ до документів Qlik можна авторизувати для вказаних користувачів або груп користувачів. У таблиці безпеки користувачам можуть бути призначені рівні доступу ADMIN або USER. Якщо рівень доступу не призначений, користувач зможе відкрити документ Qlik.

Користувач з доступом ADMIN може виконувати будь-які зміни у документі. За допомогою сторінки Безпека в діалогових вікнах Властивості документа та Властивості аркуша користувач з доступом ADMIN може

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

обмежувати права інших користувачів змінювати документ. Користувач із привілеями USER не має доступу до сторінок Безпека.

б) Системні поля доступу до секції

Рівні доступу призначаються користувачам в одній або кількох таблицях, завантажених у доступ до секції. Дані таблиці можуть містити різні користувача системні поля (як правило, USERID і PASSWORD) і поле, що визначає рівень доступу (ACCESS). Усі системні поля Секції доступу використовуються для автентифікації або авторизації. Нижче наведено повний набір системних полів доступу до секції.

У доступ до секції можна завантажити жодного, всі чи будь-яку комбінацію полів налаштувань безпеки. Таким чином, використовувати USERID не обов'язково, авторизація може бути здійснена за допомогою інших полів, наприклад тільки серійного номера.

Qlik порівнює серійний номер Qlik з полем SERIAL, ім'я користувача та групи Windows NT з NTNAME, SID домену Windows NT з NTDOMAINSID та Windows NT SID з NTSID. Далі він запросить ідентифікатор користувача та пароль та порівняти їх з полями USERID та PASSWORD.

Якщо знайдене поєднання ідентифікатора користувача та властивостей оточення також буде знайдено у таблиці доступу до секції, документ відкривається з відповідним рівнем доступу. Якщо з'єднання не знайдено, Qlik відмовить користувачу в доступі до документа. Якщо ідентифікатор користувача та/або пароль введено неправильно тричі поспіль, необхідно повністю повторити процедуру входу.

Оскільки та ж внутрішня логіка, яка є відмінною особливістю Qlik, також використовується і в доступі до секції, поля безпеки можуть бути розміщені в різних таблицях. (Таким чином, системний адміністратор може створити документ Qlik поза таблицями безпеки. У цьому випадку вірний серійний номер, пароль і т. д. будуть створені при натисканні відповідного значення поля.)

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

При виконанні процедури входу Qlik спочатку перевіряє SERIAL, NTNAME, NTDOMAINSID і NTSID, щоб перевірити, чи достатньо цієї інформації для надання користувачеві доступу до документа. У разі надання доступу Qlik відкриє документ, не запитуючи ідентифікатор користувача та пароль.

Якщо завантажено лише деякі поля доступу, використовуються відповідні вимоги наведені вище.

Усі поля, перелічені в операторах Load або Select у доступі до секції, мають бути написані у ВЕРХНОМУ РЕЄСТРІ. Ім'я будь-якого поля в базі даних, що містить літери в нижньому регістрі, має бути перетворено на верхній регістр за допомогою функції upper до читання за допомогою операторів Load або Select.

Однак для ідентифікатора користувача та пароля, що вводяться кінцевим користувачем, який відкриває документи Qlik, враховується регістр.

Знак підстановки (*) інтерпретується як усі (перераховані) значення цього поля, тобто значення, вказане у якомусь іншому місці у цій таблиці. При використанні в одному із системних полів (USERID, PASSWORD, NTNAME або SERIAL) у таблиці, завантаженої у доступ до секції скрипта, інтерпретується як усі (також і не перелічені) можливі значення цього поля.

7) Змішані середовища

Якщо планується використовувати ту саму таблицю авторизації в Qlik і Qlik Sense SaaS, необхідно враховувати наступне:

- USERID має різні значення Qlik і Qlik Sense SaaS і при використанні може викликати проблеми з безпекою. Натомість використовуйте NTNAME або об'єднайте його з SERIAL, як описано нижче.

- GROUP і поля, що починаються з 'USER.', такі як 'USER.NAME' і 'USER.EMAIL' і т. д., є (або будуть) полями автентифікації в Qlik Sense Enterprise SaaS. При використанні цих полів у Section Access доступ може бути закритий Qlik Sense SaaS.

- PASSWORD, NTSID і NTDOMAINSID не можна використовувати у Qlik Sense SaaS. У доступі буде відмовлено, якщо не використовується знак

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

підстановки.

– SERIAL не можна використовувати для перевірки номера ліцензії Qlik Sense SaaS. Однак, якщо це поле містить рядок 'QLIKCLOUD' або 'QLIK', доступ може бути наданий. Це означає, що можна мати таблицю авторизації, подібну до наступної, де рядок 1 надаватиме доступ до Qlik (але не Qlik Sense SaaS), а рядок 2 надаватиме доступ до Qlik Sense SaaS (але не Qlik).

8) Обмеження функцій Qlik

Елементи керування на сторінці Властивості документа: Безпека та Властивості аркуша: Безпека дозволяє заборонити доступ до певних елементів меню, а також заборонити вносити зміни до макету. Якщо ці параметри будуть використовуватися як дійсно захисний захід, важливо, щоб користувачі увійшли до системи як USER. Всі користувачі, які увійшли до ADMIN, можуть змінювати параметри безпеки в будь-який час.

У користувача, який відкрив документ із правами USER, відсутні сторінки Безпека у діалогових вікнах «Властивості».

9) Динамічне скорочення даних

Qlik та сервер Qlik підтримують функцію, що дозволяє приховати деякі дані в документі від користувача на основі входу до доступу до секції.

Насамперед, поля (стовпці) можна приховати, використовуючи системне поле OMIT.

Крім того, записи (рядки) можна приховати шляхом зв'язку даних доступу до секції з реальними даними: Вибір значення для відображення/виключення управляється за допомогою одного або кількох полів із загальними іменами у доступі до секції та програми секції. Після входу користувача в систему Qlik спробує скопіювати вибірки в полях у доступі до секції в будь-які поля в додатку секції з такими ж іменами (імена полів повинні бути у ВЕРХНОМУ РЕЄСТРІ). Після створення вибірок Qlik постійно приховуватиме всі дані, виключені цими вибірками, від користувача.

Для застосування цієї процедури необхідно встановити параметр Початкове скорочення кількості даних з урахуванням доступу до секції на сторінці Властивості документа: Відкриття. Якщо ви використовуєте цю функцію в документах, які повинні поширюватися іншими способами, а не через Qlik Server, необхідно встановити параметр Заборона бінарного завантаження на цій же сторінці діалогового вікна «Властивості документа» для забезпечення захисту даних.

10) Успадковані обмеження доступу

Бінарне завантаження спричинить успадкування обмежень доступу новим документом Qlik. Користувач із правами ADMIN для нового документа може змінити права доступу цього документа, додавши нову секцію доступу. Користувач із правами USER може виконувати та змінювати скрипт, таким чином додаючи власні дані до файлів із бінарним завантаженням. Користувачі з правами USER не можуть змінювати права доступу. Також це дозволяє адміністратору бази даних керувати правами доступу до документів Qlik із бінарним завантаженням.

11) Шифрування

Зв'язок між сервером Qlik та клієнтом Qlik для Windows шифрується. Однак під час використання клієнта AJAX зв'язок не шифрується.

Крім того, всі документи Qlik шифруються, що робить інформацію недоступною для перегляду, відладчиків і т.п.

Секретні дані у файлах QVD можна також зашифрувати за допомогою пар ключів клієнта, які дозволяють вам керувати доступом до даних.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Впровадження програмного рішення у експлуатацію складається з двох частин:

1. Впровадження середовища.
2. Впровадження додатку.

Нижче буде більш детально розглянуто елементи впровадження.

5.1 Впровадження середовища

Для загального впровадження системи Qlik спочатку треба встановити її. Після встановлення системи та її активізації можна починати керувати доступами за допомогою Qlik Sense SaaS.

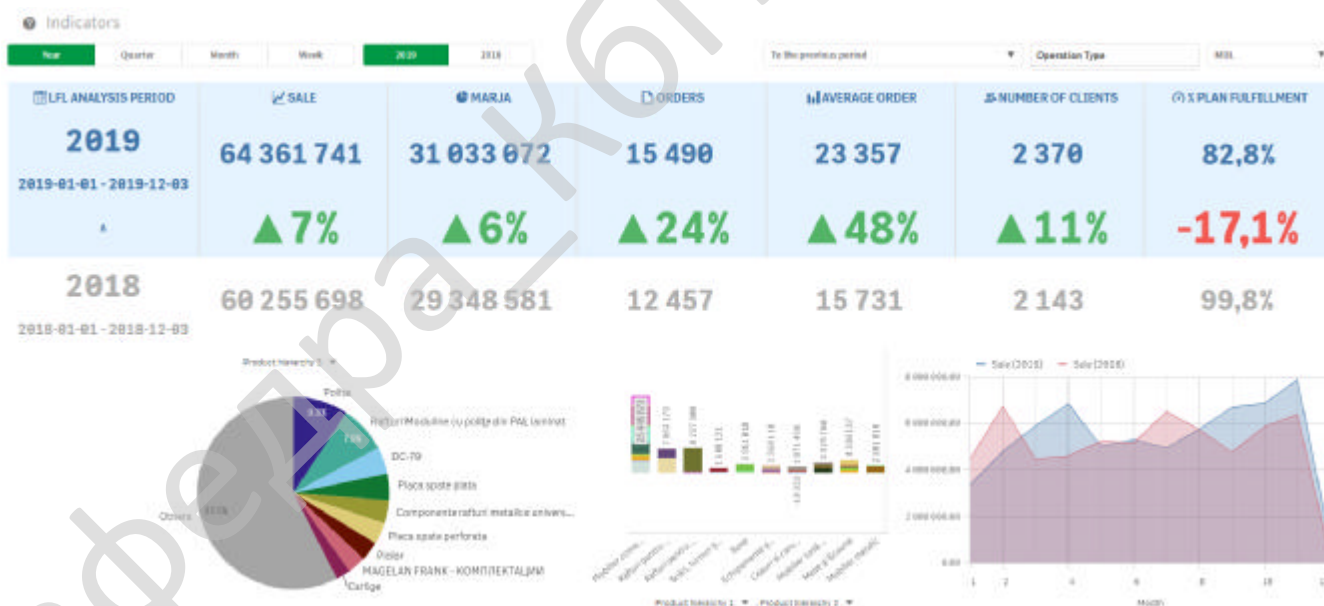


Рисунок 5.1 – Перший лист додатку

Qlik Sense SaaS пропонує кілька способів організувати користувачів, які додаються до клієнта. Після додавання користувачів до клієнта, їх можна

відсортувати за групами відповідно до того, чи будуть вони користувачами програми, чи створюватимуть вміст у клієнті. Для цього застосовуються розподіли користувачів. Можна призначити ролі на основі завдань користувачам так, щоб деякі з них мали певні дозволи в клієнті. Для цього застосовуються користувальницькі ролі. На рисунку 5.3 показані кроки щодо додавання користувачів до клієнта та організації згідно з ролями, зміни ці налаштування можливо у будь-який час.

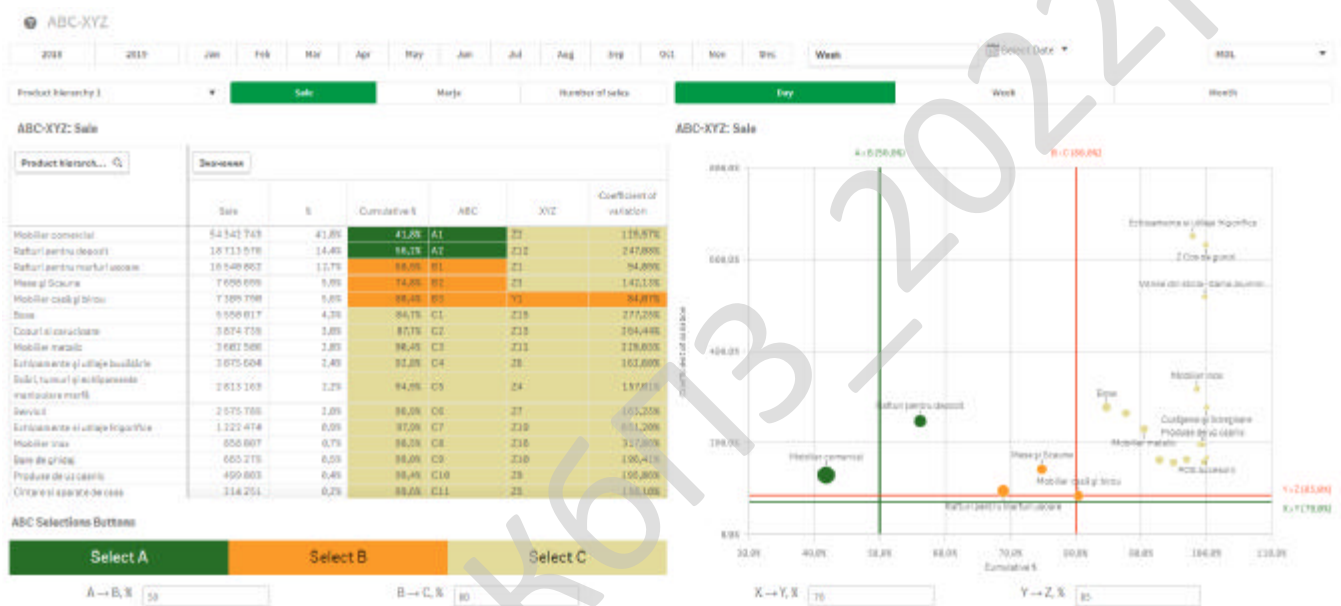


Рисунок 5.2 – Лист ABC-XYZ

Користувальницькі ролі надають користувачам низку дозволів на рівні клієнта за рамками загальних дозволів, отриманих під час розподілу користувачів. За допомогою ролей користувача можна надати користувачеві певну роль або функцію в клієнті. Призначити ролі користувачам може лише адміністратор клієнта, і це може бути лише з Management Console. Користувальницькі ролі встановлюються за бажанням, і призначати їх користувачам необов'язково. У користувачів без користувальницької ролі дозволу засновані на їхньому розподілі.

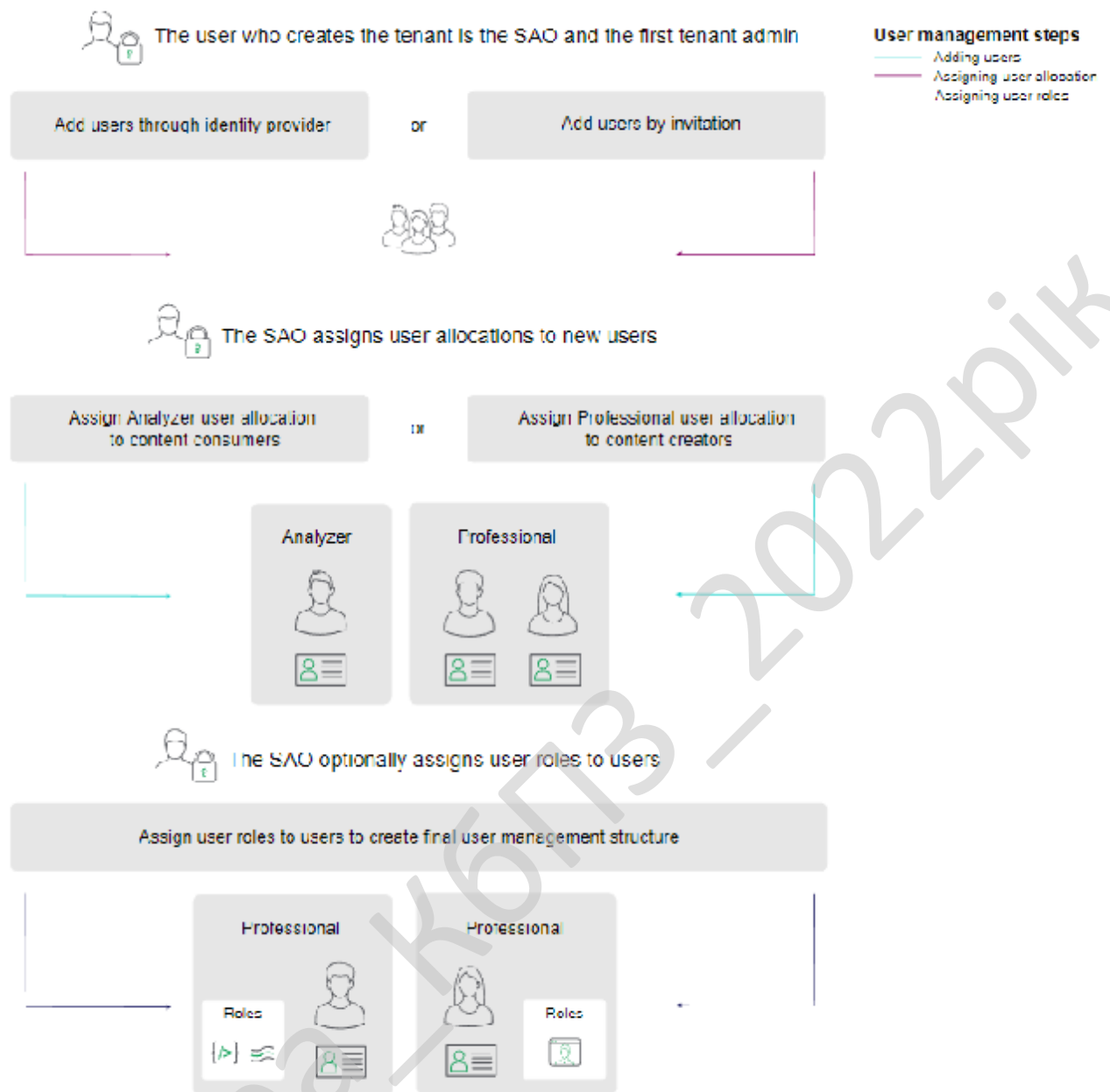


Рисунок 5.3 – Блок-схема роботи програм для забору та обробки даних.

Впровадження додатку

Для впровадження додатку потрібно розмістити три додатки на сервері, де розгорнуто середовище Qlik Sense. Після чого адміністратори мережі повинні побудувати таймінг активації доданків та перевірити коректність шляхів файлів у системі, після чого можна перевіряти якість роботи додатку на робочому місці.

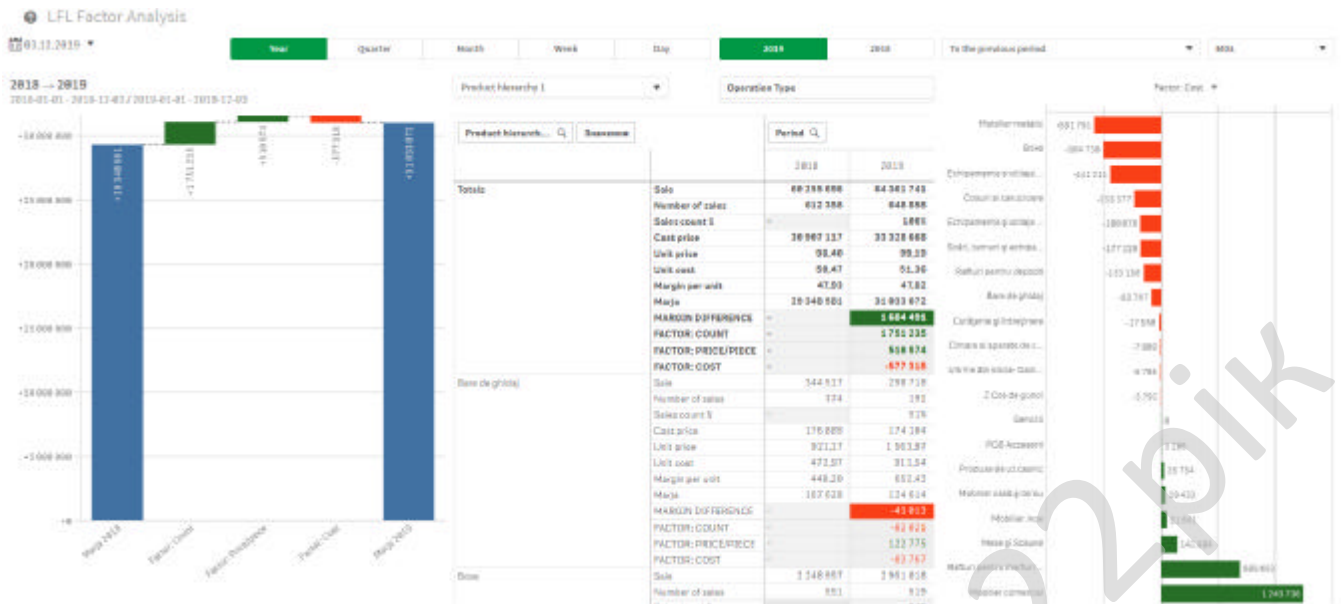


Рисунок 5.4 – Лист LFL-аналізу

Подальші кроки залежать від того яким чином буде використовуватись додаток: у якості мешапу, чи як окремий додаток.

Якщо інформаційно-аналітичний додаток буде використовуватися як окремий додаток, то адміністратори мережі можуть надати дозвіл на огляд додатку у якості користувача співробітникам які цього потребують. Надати право на редагування вони не можуть через те що це буде рівносильно передачі мавпі гранати.

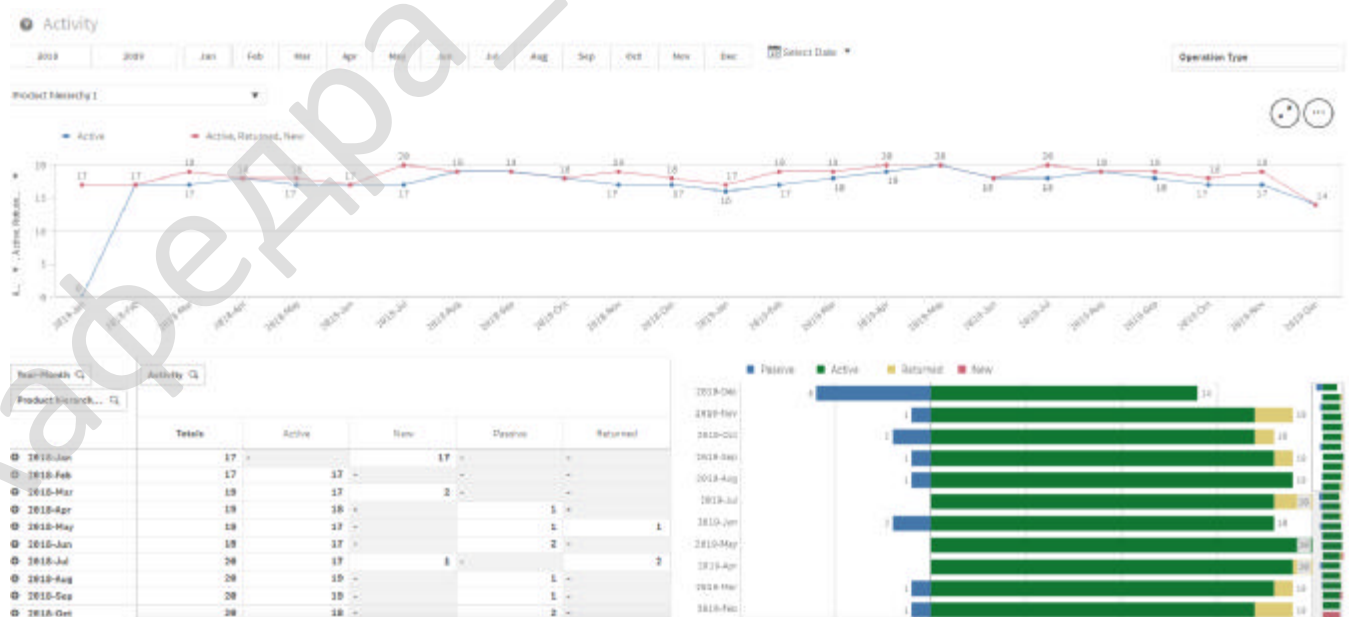


Рисунок 5.5 – Лист активності

У тому випадку коли додаток використовується у якості мешапу, то права на користувацький огляд нікому не надаються. Під час реалізації даного випадку додаток сам фактично стає джерелом даних для сайту який являється мешапом даного додатка. Дозволи на редагування сайту мешапу обговорюються між сторонами, зазвичай право на редагування залишається у розробника.

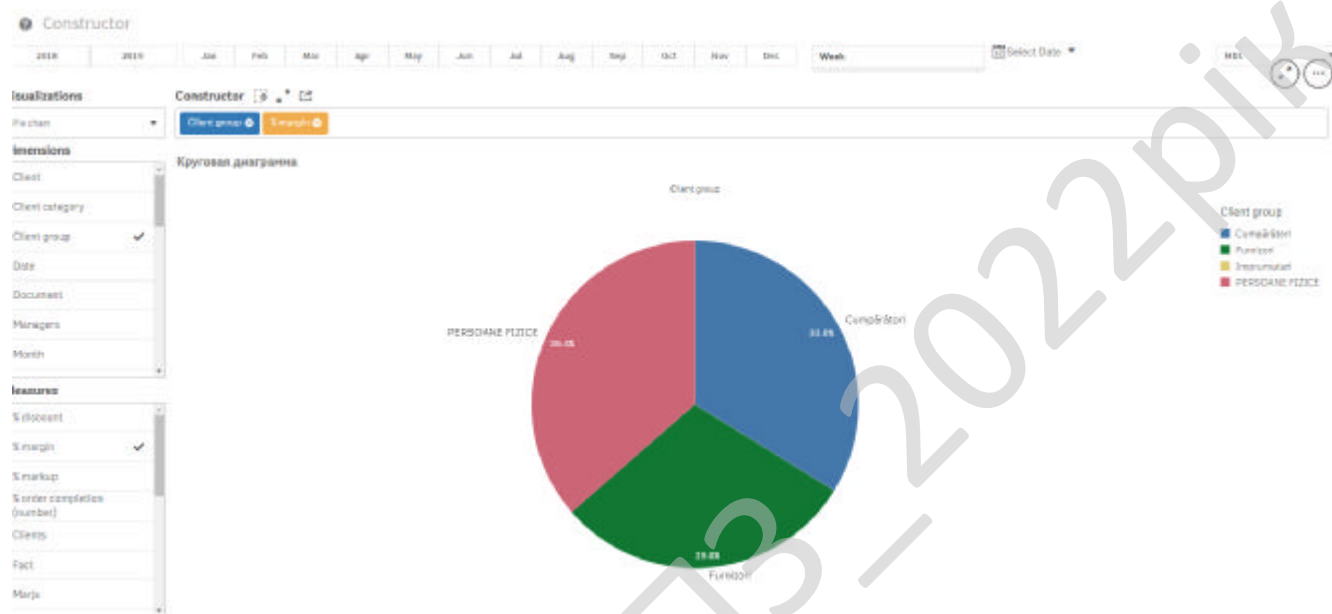


Рисунок 5.5 – Лист конструктора

6 НАУКОВА НОВИЗНА

Метою роботи є дослідження та програмна реалізація інформаційно-аналітичної системи для аналізу великих даних.

Об'єктом дослідження є процес роботи з великими даними у QlikSense.

Предметом дослідження є методи та алгоритми розробки інформаційно-аналітичного середовища для роботи з великими даними у QlikSense.

Методи дослідження базуються на теорії аналізу даних, теорії статистики теорії об'єктно-орієнтованого програмування, теорії алгоритмів, методах побудови аналітичного інтерфейсу.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

1. Удосконалено метод побудови аналітично-інформаційних систем, що відрізняється від існуючих можливістю обробляти інформацію незалежно-послідовним чином, що запобігає збою у роботі програми ціною актуальності найновіших даних, які вже надійшли у джерело даних.

2. Розроблено вітчизняний продукт інформаційно-аналітичної системи для аналізу великих даних, який має більш широкі можливості, на відміну від існуючих аналогів та може бути використаний для бізнес-аналітики.

Науковою новизною результатів розробки є створення відчизняного програмного забезпечення аналітично-інформаційної системи, що використовує BigData у якості джерела даних, за допомогою інкрементально-незалежного підходу для представлення багатофункціональної аналітично-інформаційної системи, з можливістю побудувати власну аналітику, окрім тої, яка була мною створена, у вигляді WEB-сторінки, що дозволить користуватись нею не встановлюючи програмне забезпечення аналітично-інформаційної системи на інструмент роботи, такий як: планшет, смартфон, сенсорний мобільний телефон, ноутбук, персональний комп'ютер та смарт-годинник.

Інкрементально-незалежна методика заключає у собі розділення єдиного

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

додатку, який має безліч функцій та процесів, на N-ну кількість елементів, кожен з яких послідовно виконує свої процеси.

Незалежність дає змогу у разі виникнення помилки на будь-якому етапі нової ітерації роботи додатків спокійно зігнорувати факт помилки та продовжити роботу. В іншому разі додаток не зміг би нормально функціонувати. Також через те, що проміжні додатки генерують проміжні бази інформації у зручних для роботи місцях, у системи немає постійної прив'язки до баз даних. Також подібні проміжні файли дають змогу використовувати лише кінцевий додаток у місці розташування аналітично-інформаційної системи.

Інкрементальний підхід використовується для покращення швидкодії роботи при записі-зчитуванні проміжних даних джерела даних, які мають певні критерії розбивки за періодами, можна розбити на менші джерела даних, які при роботі системи не будуть постійно змінюватись через те, що ці дані за своєю природою не потребують внесення будь-яких змін у зв'язку з їх структурою. Це дозволяє виконувати запис даних у пришвидженому форматі. Подібний підхід дозволяє оперувати дефрагментованою інформацією яка захищена від випадкових змін. Також орієнтуючись на подібний підхід можна корегувати обсяг (давність) інформації яка буде виводитись на екран кінцевого користувача. Разом з тим, що цей підхід дозволяє швидко завантажувати нові дані та лімітувати обсяг наявних, подібний взаємозв'язок також впливає на швидкість роботи фінального додатку як при завантаженні даних так і при фактичній роботі з програмним продуктом.

У результаті розробки були реалізовані та досліджені існуючі концепції побудови аналітично-інформаційних систем, також були протестовані власні методики, які за теоретичним значенням придатні до практичного застосування, проте їх реальна коректність та можливість подальшого використання потребує «випробування боєм», оскільки аналітично-інформаційні системи у більшості випадків розробляються для використання у вузькоспеціалізованих колах через те, що можуть містити конфіденційну, для замовника, інформацію, частина якої як раз і оголошується після ознайомлення з інформаційно-аналітичною системою.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

У зв'язку з тим що готові програмні рішення являють собою джерело конфіденційної інформації здобувачем вищої освіти на основі теоретичних даних було реалізовано концепції побудови зручного для користувача додатка, який дозволяє обробляти інформацію незалежно-послідовним чином, що запобігає збоєм ціною актуальності найновіших даних, які вже надійшли у джерело даних. Також цей підхід дозволить маючи оброблені дані використовувати лише завершальний додаток для відображення аналітично-інформаційної системи.

Структура роботи, виклад матеріалу і оформлення здійснені відповідно до вимог, які пред'являються до дипломів на здобуття наукового ступеня магістра з комп'ютерної інженерії.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми магістерської роботи

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 24 днів (один місяць).

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи кібербезпеки силової інфраструктури ЦОД.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;
- г) зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	60
3. Запланований термін розробки, днів	Fpq	24 (1 місяць)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	В
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	8
8. Кількість форм вихідної інформації.	–	6
9. Мова програмування (1-6)	–	2
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	1
13. Рівень спрацьованості колективу (1-6)	–	3
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	1
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	1
17. Складність кінцевого програмного продукту (1-6)	–	3
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	3
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	6
23. Професійний рівень аналітиків (1-6)	–	3
24. Професійний рівень програмістів (1-6)	–	4
25. Постійність складу команди розробників (1-6)	–	1
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	3
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	70000
33. Норматив додаткової зарплати, % :	Н _д	10
34. Норматив відрахувань у соціальні фонди, %	Н _с	22
35. Норматив загальногосподарських витрат, %	Н _г	15
36. Норматив витрат на освоєння нових мов програмування, %	Н _п	15
37. Рівень рентабельності програмної продукції, %	Р _е	30
38. Ставка податку на додану вартість, %	Н _{дв}	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B, \quad (7.1)$$

де: A – коефіцієнт Боема, $A = 2,45$;

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

Size – загальний об'єм відлагодженого програмного коду, тис. рядків;

B – показник ступеня, що визначається співвідношенням:

$$B = 1,01 + 0,001 \sum W_i, \quad (7.2)$$

де: W_i – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(3,24 + 3,64 + 0 + 1,98 + 2,73) = 1,021.$$

$$T_{ном} = 2,45 \cdot 3,772^{1,021} = 9,5 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} \prod V_j, \quad (7.3)$$

де: $\prod V_j$ – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 9,5 \cdot (0,88 \cdot 1,19 \cdot 1,149 \cdot 0,89 \cdot 1 \cdot 1 \cdot 0,87 \cdot 0,67 \cdot 0,87 \cdot 1 \cdot 0,88 \cdot 0,81 \cdot 0,84 \cdot 0,72 \cdot 1 \cdot 1) = 2,88 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкість програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{рп} = 0,3 C T_{уточн}^{0,33 + 0,2(B-1,01)} S, \quad (7.4)$$

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4);

S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%.

$$T_{рп} = 0,3 \cdot 2,75 \cdot 2,88^{0,33 + 0,2(1,021-1,01)} \cdot 60 = 70 \text{ люд/день.}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	5	Д5
Ескізний проект	5	Д6
Технічний проект	7	Д7
Робочий проект	48	Ф 7.1-7.4
Впровадження	5	Д13
Всього	70	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою:

$$Ч = \frac{T_{нз}N}{F_{pq} - H_{ев}}, \quad (7.5)$$

де: F_{pq} – плановий фонд робочого часу одного спеціаліста, днів;

$T_{нз}$ – трудомісткість розробки програмного забезпечення люд-дні.

$$Ч = \frac{60 \cdot 1}{24 \cdot 3} = 2,86 \text{ ставки.}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3.

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	9	810	13,5
Монітор	60	9	540	9
Клавіатура	30	9	270	4,5
Маніпулятор «мишка»	30	9	270	4,5
Принтер матричний	60	1	60	1
Принтер лазерний	120	2	240	4
Принтер струминний	60	1	60	1
Сканер	20	2	40	0,(6)
Концентратор–маршрутизатор	30	2	60	1
Кабельні господарства ЛВС на 1 м. п.	2,5	60	150	2,5
Копіювальний апарат	140	1	140	2,(3)
Усього за рік:			3 _ч	44

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%.

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{др}^c = \frac{3_{ч} \cdot n_{mic}}{1,2}, \quad (7.6)$$

$$\Phi_{др}^c = \frac{44 \cdot 1}{1,2} = 36, (6) \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{\Phi_{др}^c}{F_{др} \cdot T_{зм}}, \quad (7.7)$$

$$Ч_{ел} = 36,(6)/(24 \cdot 8) = 0,2 \text{ ставки.}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (ОС FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2019, серверу доступу ADSL (ОС Linux), налаштування ADSL, VPN, PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (СМТS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	2	0,5
	Підтримка постійних клієнтів	1	
	Оформлення договорів, ведення тендерів	0,5	
	Контроль взаєморозрахунків з постачальниками	0,5	
Всього		4	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,5
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	14000	14000
Адміністратор загальної мережі	0,5	15000	7500
Інженер - електронщик	0,2	10000	2000
Продакт-менеджер	0,5	12000	6000
Інженер-програміст	3	12000	36000
Інженер верстальник	0,5	10000	5000
Системний програміст	0,5	10000	5000
Бухгалтер-економіст	0,5	10000	5000
Всього за період розробки	$R_{cn} = 6,7$	-	$\Phi_{роб} = 80500$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{cd} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де: $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$z_{cd} = \frac{80500}{6,7 \cdot 24} = 500 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі:

$$B_{y0} = R_{cn}^1 S_y C_{nl}, \quad (7.9)$$

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 6,7 робочих місць;

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

S_y – питома площа на одне робоче місце, m^2 ;

$C_{пл}$ – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград, вул. Глинка 16) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 250...1600 у.о./ m^2 . Враховуючи, що курс у.о. складає 25 приймаємо для розрахунку вартість одного метра квадратного рівною 6560 грн./ m^2 . На кожне робоче місце у середньому потрібно $8m^2$. З урахуванням цього:

$$B_{уд} = 6,7 \cdot 8 \cdot 6560 = 351616 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 35161,6 грн.

Балансова вартість інвентарю розраховується за нормою 6000 грн. на одне робоче місце. Тобто:

$$I_{нв} = R_{сп}^1 \cdot C_m, \quad (7.10)$$

де: C_m – ціна меблів для одного робочого місця, грн.

$$I_{нв} = 6,7 \cdot 6000 = 40200 \text{ грн.}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу фірми Brain за 28.10.21 – джерело <https://brain.com.ua>.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок	VINGA ADVANCED B0166 (R3M8INT.B0166)	7347
Процесор	AMD Ryzen 3, 2100GE, 4 ядра, 4 потоки, Частота процесора, 3.2 ГГц, Частота в Boost, 3.6 ГГц	-
Системна плата	MB AM4, Чіпсет AMD A320, 1 x Headphone, 4 x USB 3.0, 3 x Audio, 1 x Microphone, 4 x USB 2.0, 2 x PS/2, 1 x HDMI, 1 x VGA, 1 x RJ45, Realtek ALC887, 10/100/1000 Мбіт/с	-
Відеокарта	Вбудована AMD Radeon Vega 8	-
Жорсткий диск	SSD 120 GB	-
Оперативна пам'ять	8 ГБ DIMM, DDR4-2666 MHz, PC4-21300	-
DVD-привод	Не комплектується	-
Корпус	ATX Vinga CS108B, PSU 350W(FSP Brand: ATX-400PNR, 12cm), (front bezel – black+light silver; body material – 0.6mm), Air Duct chassis design,Thermally Advantaged Chassis	-
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-B, int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	220
Інше	Клавіатура, мишка	Подарунок

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Монітор	PHILIPS 223V5LSB2/10, 21.5", TN, 1920 x 1080, 16:9, WLED, матове покриття, 5мс	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	4	10947	16420,5	60208,5
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	73360

Витрати на транспорт, монтаж та випробування прийняті в межах до 10% від оптової ціни.

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	351616	-	-
2. Передавальні пристрої	35161,6	-	-
Всього по групі	386 777	5	19338
Група 4			
3. Обчислювальна техніка	73360	-	-
Всього по групі	73360	50	36 680
4. Нематеріальні активи	70000	50	35000
Група 5, 6			
5. Вимірювальні пристрої	9031	25	2257,75
6. Транспортні засоби	143000	20	28600
7. Господарський інвентар	45500	25	11375
Всього по групі	197531	-	42232,75
Разом	К _р = 727 668		А _р = 133250,75

Примітка: вартість автомобіля Sens (Standard+) взята по даним з автосалону «Кіровоград-Авто», джерело [http://kirovograd-avto.ukravto.ua /catalog/tm-9/model-80/description](http://kirovograd-avto.ukravto.ua/catalog/tm-9/model-80/description), складає 143000 грн.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{сд} \cdot T_{нз}}{N_e}, \quad (7.11)$$

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 500 \cdot 70 / 60 = 583 \text{ грн.}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%:

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де: H_q – норматив додаткової зарплати, %.

$$Z_d = 583 \cdot 10 \cdot 0,01 = 58,3 \text{ грн.}$$

Відрахування на соціальні потреби за нормативом $H_c = 22\%$ від суми основної та додаткової зарплати:

$$C_{оц} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де: H_c – відрахування на соціальні потреби, %.

$$C_{оц} = 0,01 \cdot 22(500 + 50) = 121 \text{ грн.}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_z = 15\%$ від основної зарплати:

$$G_{осн} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де: H_z – загальногосподарські витрати, %.

$$G_{осн} = 500 \cdot 15 \cdot 0,01 = 45 \text{ грн.}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де: Z_{M1} – вартість паперу, грн.;

Z_{M2} – вартість запам'ятовуючих пристроїв, грн.;

Z_{M3} – вартість фарби, картриджей, тонеру, грн.;

N_e – кількість екземплярів програм, шт.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

Згідно виданих викладачем норм приймаємо 0,5 пачки паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n = 200$ грн., визначаємо вартість паперу за період розробки $N_m = 1$ міс:

$$Z_{M1} = C_n \cdot N_m. \quad (7.16)$$

$$Z_{M1} = 200 \cdot 1 \cdot 0,5 = 100 \text{ грн.}$$

Згідно виданих викладачем норм до вартості запам'ятовуваних пристроїв входить вартість CD дисків в кількості, що дорівнює 30 екземплярів програм та одного DVD диска для збереження резервної копії програми:

$$Z_{M2} = \sum C_d, \quad (7.17)$$

де: C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 3 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 9 грн./шт.

$$Z_{M2} = 30 \cdot 9 + 9 = 279 \text{ грн.}$$

Згідно виданих викладачем норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_z, \quad (7.18)$$

де: C_z – вартість розхідних матеріалів друкуючих пристроїв: картридж для CANON LBP-3010 Black Canon 712 – 574 грн.; картридж для EPSON STYLUS PHOTO R390 – 558 грн.; картридж для CANON IR-1022A – LJ Q2612A Cart. HP LJ 1010/1012/1015/3015/3020/3030 (2500 стр.) – 570 грн.

$$Z_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$Z_M = (100 + 289 + 1702) / 60 = 34,85 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($H_n = 15\%$) від основної зарплати виконавців:

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де: H_n – норматив витрат на освоєння нових мов програмування, %.

$$O_n = 500 \cdot 15 \cdot 0,01 = 45 \text{ грн.}$$

Визначимо витрати на амортизацію фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм:

$$A_m = \frac{A_p \cdot N_{mic}}{N_e \cdot 12}, \quad (7.20)$$

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

де: A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 133250,75 \cdot 1 / (60 \cdot 12) = 185,07 \text{ грн.}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_M + O_n + A_m. \quad (7.21)$$

$$C_n = 500 + 50 + 121 + 45 + 34,85 + 45 + 185,07 = 980,92 \text{ грн.}$$

Визначимо плановий прибуток за рівнем рентабельності (P_n) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 40%.

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 40 \cdot 980,92 = 392,37 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн.
1	2	3
1. Основна зарплата виконавців	Z_o	500
2. Додаткова зарплата виконавців	Z_d	50
3. Відрахування на соціальні потреби	C_{oc}	121
4. Загальногосподарські витрати	Γ_{ocn}	45
5. Витрати на матеріали	Z_M	34,85
6. Освоєння нових операційних систем, мов програмування	O_n	45

Продовження таблиці 7.9

1	2	3
7. Амортизація основних фондів	A_m	185,07
8. Повна собівартість програмного забезпечення	C_n	980,92
9. Плановий прибуток	P_p	392,37
10. Ціна підприємства $C_n = C_n + P_p$	C_n	1 373,29
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{ог} \cdot C_n$	$ПДВ$	274,66
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	1 647,95

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.9.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	1648
Всього капітальних витрат	–	1648

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на обслуговування	Z_p	10048,9	6280,56
2. Витрати на електроенергію	$Z_{ел}$	756	472,5
3. Витрати на амортизацію	$Z_{ам}$	0	1538
Всього витрат за рік	I	10804,9	8 291,06

Витрати на обслуговування системи:

$$Z_p = T_p \cdot Z_2 \cdot (1 + 0,01 \cdot N_q) \cdot (1 + 0,01 \cdot N_c), \quad (7.23)$$

де: T_p – кількість годин обслуговування за рік, год.;

Z_2 – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 240 годин на рік до 150 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 240 \cdot 10,4 \cdot 1,1 \cdot 1,22 \cdot 3 = 10048,9 \text{ грн},$$

до:

$$Z_{p \text{ нов}} = 150 \cdot 10,4 \cdot 1,1 \cdot 1,22 \cdot 3 = 6280,56 \text{ грн}.$$

Витрати на електроенергію визначаються з урахуванням споживаємої потужності ($P_{ел}$) в кіловатах, часу експлуатації технічних засобів (T_p) в годинах та ціни однієї кіловат-години ($C_{ел}$):

$$Z_{ел} = P_{ел} \cdot T_p \cdot C_{ел}. \quad (7.24)$$

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

$$Z_{ел баз} = 7 \cdot 0,15 \cdot 240 \cdot 3 = 756 \text{ грн.}$$

$$Z_{ел нов} = 7 \cdot 0,15 \cdot 150 \cdot 3 = 472,5 \text{ грн.}$$

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн., за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	50	–	1648	–	824
Всього відрахувань	-	–	1648	–	824

7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою:

$$E_v = (C_n - C_n) \cdot N_e - E_n \cdot K_p, \quad (7.25)$$

де: K_p – балансова вартість основних фондів розробника, грн.

$$E_v = (1373,29 - 980,92) \cdot 60 - 0,15 \cdot \frac{727668}{12} = 16\,246,35 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_e = \frac{K_p}{(C_n - C_n) \cdot N_e}, \quad (7.26)$$

$$T_e = \frac{727668}{(1373,29 - 980,92) \cdot 60 \cdot 12/1} = 2, (57) \text{ року.}$$

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\bar{o}} - I_n) - E_n(K_n - K_{\bar{o}}), \quad (7.27)$$

де: $I_{\bar{o}}, I_n$ – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

$K_{\bar{o}}, K_n$ – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = 19\,095,96 - 824 = 18\,271,96 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{cn} = \frac{K_n - K_{\bar{o}}}{I_{\bar{o}} - I_n}, \quad (7.28)$$

$$T_{cn} = \frac{1648}{19095,96} = 0,09 \text{ року.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	60
2. Повна собівартість розробленої програми	Грн.	980,92
3. Ціна розробленої програми	Грн.	1 373,29
4. Плановий прибуток від реалізації розробленої програми	Грн.	667,03
5. Рентабельність програмної продукції	%	40
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	727668
7. Загальний прибуток від реалізації програмної продукції	Грн.	65 918

Продовження таблиці 7.13

Найменування показників	Одиниця виміру	Величина
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	16 246,35
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	2,(57)
10.Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	1 647,95
11.Величина економічного ефекту у користувача програмної продукції	Грн.	18 271,96
12.Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,09

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Охорона праці – це: система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини в процесі трудової діяльності;

Охорона праці є складовою частиною безпеки життєдіяльності [53, 55].

Законом України “Про охорону праці” [53] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [55], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

Розглянемо шкідливі чинники роботи програмістів керуючись наступними нормативно-правовими актами: “Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин” ДСанПіН 3.3.2-007-98, та “Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями” НПАОП 0.00-7.15-18.

Умови праці програміста включають наступні фактори:

- вентиляція приміщення;
- освітлення приміщення;

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

- параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення негативного впливу комп'ютера на організм людини визначимо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.[71]

Штучне занулення.

Ця річ являє собою підключення захисного проводу до металевих складових конструкції, які не проводять електричний струм. Можливо також приєднання до нуля, використовуваним джерелом напруги або до іншого заземленого проводу.

Нульовий захисний дріт прокладають від стійки групового розподільчого щита, розподільчого пункту до розеток живлення. Не допускається підключення на щиті до одного контактного затискача нульового робочого та нульового захисного провідників. Площа перерізу нульового робочого та нульового захисного провідника в груповій трьох-провідній мережі повинна бути не менше площі перерізу фазового провідника.

Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальна машина (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють люди (у тому числі програмісти) необхідно створити належний мікроклімат, параметри якого регламентуються Державними санітарними правилами і нормами, зокрема ДсанПіН 3.3.2.007-98.

При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території;
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

- електромагнітні (у тому числі високочастотні) електромагнітні випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- монотонність праці;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шум;
- статистичні навантаження на кістково-м'язовий апарат.

8.2 Аналіз умов праці на робочому місці фахівця

Дослідження та аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці:

Таблиця 8.1 — Розміри приміщення

Найменування	
Ширина	
Довжина	
Висота	

Таблиця 8.2 - Площа та обсяг приміщення, на одного працюючого

Гео	О	Но	Фактичне значення
мет	д	рм	
рич	и	ат	
на	н	ив	

хар акт ери сти ка	и ц я в и мі р у	не зн ач ен ня	
Пл ощ а, S	м	не ме н ше	
Об' єм,	м	не ме н ше	

Нормованим параметром природного освітлення являється коефіцієнт природного освітлення (КПО). КПО встановлюється в залежності від розряду виконуваних зорових робіт.

Робота оператора ПК відноситься до робіт середньої точності (IV розряд зорових робіт, мінімальний розмір об'єкту розрізнення складає 0,5 – 1,0 мм), для яких при використанні бокового освітлення КПО=1,5 %.

Для штучного освітлення нормованим параметром виступає Емін – мінімальний рівень освітленості, та Кп – коефіцієнт пульсації світлового потоку, який не повинний бути більшим ніж 20%.

Мінімальна освітленість встановлюється в залежності від розряду виконуваних зорових робіт. Для IV розряду зорових робіт вона складає 300...500 лк.

Була проведена перевірка освітленості робочого місця користувача ПК на відповідність розряду зорової роботи. За даними вимірювань рівень природної освітленості поверхні, де розташований ПК, складає 200 лк за освітленості тієї же поверхні відкритим небосхилом в 20000 лк, тобто КПО = 1%, що не відповідає нормативному КПО. Для штучного освітлення у приміщенні використовуються люмінесцентні лампи.[68]

Розташовані у приміщенні 3 ПК є джерелами тепловиділень, крім того для підтримання у приміщенні в холодний період року оптимальних параметрів мікроклімату використовуються нагріті поверхні опалювальної системи. Нормованим показником ІЧВ являється гранично допустима густина потоку енергії $I_{гд}$, Вт/м², яка встановлюється в залежності від площі опромінюваної поверхні тіла людини ($S_{опр}$). Нормовані рівні складають:

$$I_{гд} = 35 \text{ Вт/м}^2 \text{ за } S_{опр} > 50\%;$$

$$I_{гд} = 70 \text{ Вт/м}^2 \text{ за } S_{опр} \sim 25-50\%;$$

$$I_{гд} = 100 \text{ Вт/м}^2 \text{ за } S_{опр} < 25\%$$

Нормування параметрів проводиться в залежності від періоду року та категорії важкості виконуваних робіт. Для постійних робочих місць, якими є робочі місця операторів ПК, встановлені оптимальні параметри мікроклімату, а за неможливості їх дотримання використовують допустимі параметри. Робота оператора ПК за енерговитратами відноситься до категорії легких робіт Ia, Ib. В таблиці 8.3[68]. наведені оптимальні параметри мікроклімату в приміщеннях, де виконуються роботи операторського типу.

Таблиця 8.3 - Оптимальні і фактичні значення параметрів мікроклімату

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

П о р а р о к у Х о л о д н а Т е п л а	Оптимальні для Іа			Фактичні		
	Температура, °С	Вологість,%	Швидкість повітря, м/с	Температура, °С	Вологість%	Швидкість повітря, м/с

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року.

В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

Одним з найважливіших факторів, які впливають на ефективність трудової діяльності людини та попереджають травматизм і професійні захворювання програмістів, є освітлення на робочому місці.

Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, було отримано наступні результати:

- мікроклімат відповідає нормативному значенню;
- акустичні умови роботи не становлять вище нормативного значення;
- розміри приміщення, у розрахунку на одного працівника є відповідними до норми.

Тому можна припустити, що основною причиною можливо зниження працездатності програміста є вплив на психофізіологічний стан, через це є наступні пропозиції: дотримання позитивної психологічної атмосфери в колективі, розмірений режим праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані наступні заходи: За потреби особливої концентрації уваги під час виконання робіт суміжні робочі місця операторів необхідно відділяти одне від одного перегородками висотою 1,5 – 2м. Конструкція робочого місця користувача персонального комп'ютера має забезпечити підтримання оптимальної робочої пози офісного працівника. Конструкція робочого столу має відповідати сучасним вимогам ергономіки і забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання (дисплея, клавіатури, принтера) і документів. Висота робочої поверхні робочого столу має регулюватися в межах 68-80 см, а ширина і глибина – забезпечувати можливість виконання операцій у зоні досяжності моторного поля (рекомендовані розміри: 60-140 см, глибина – 80-100 см). Робочий стіл повинен мати простір для ніг заввишки не менше ніж 60 см, завширшки не менше ніж 50 см, завглибшки (на рівні колін) не менше ніж 45 см, на рівні простягнутої ноги не менше ніж 65 см. Робочий стілець має бути підйомно-поворотним, регульованим за висотою, з кутом і нахилу сидіння та спинки і за

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

відстанню від спинки до переднього краю сидіння поверхня сидіння має бути плоскою, передній край – заокругленим.[58]

8.3 Розрахункова частина

Розрахунок занулення

Початкові дані

1. Потужність електродвигуна, який підлягає зануленню : $P = 10$ кВт.
2. Кількість електродвигунів: $m = 4$.
– Потужність освітлювальних приладів : $P_o = 20$ кВт.
3. Довжина магістрального кабеля: $L_M=60$ м.
4. Довжина розгалудження (від щита до електродвигуна) : $l=20$ м.
5. Матеріал провідників кабеля–алюміній.
6. Лінійна напруга $U=360$ В.
7. Фазна напруга $U_f=220$ В

1.1 Сила номінального струму електроустановки:

$$I_{\text{ном}} = \frac{10 \cdot 1000}{\sqrt{3} \cdot 360 \cdot 0,85} = 18,9$$

1.2 Сила пускового струму:

п
у
·
у

1.3 Номінальна сила струму апарата захисту:

н
л
н
·
п
у
с

З таблиці 1 [67] вибирається запобіжник ПН2-100 з плавкою вставкою $I_{\text{ном}} = 40$ А.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

1.4 Найменше допустиме по умовам спрацьовування захисту значення сили струму короткого замикання, А:

$$37,7 \cdot 3 = 113 \text{ А}$$

1.5 Переріз дроту розгалуження з умови допустимого нагрівання:

$$d_{\text{доп}} \geq 18,9$$

Площа перерізу узгоджується з номінальним струмом плавкої вставки запобіжника із умови:

$$d \geq d_0$$

2 Параметри магістрального кабелю:

$$r = r_0$$

2.1 Струм короточасного перевантаження магістрального кабелю:

$$I_{\text{б}} \cdot P \cdot T \cdot E$$

2.2 Струм спрацювання кабельного або розеткового розчеплювача автоматичного вимикача:

$$I_{\text{спр}} \geq I_{\text{роб}} \geq 60,14 \text{ А} \quad (9)$$

2.3 Максимальний струм перевантаження лінії:

$$\frac{K}{\sqrt{3}} \cdot I_{\text{н}} = 80 \text{ А}$$

Вимикач: А3714Б

2.4 Площа перерізу магістрального кабелю:

$$S_{\text{н}} = S_{\text{д}} = S_{\text{ф}}$$

3. Потужність трансформатора:

$$P_{\text{н}} = P_{\text{д}}$$

Вим.	Арк.	№ докум.	Підпис	Дата	ВКМР-123.22.0026.00.00.ПЗ	Арк.
						102

$$I_{TP} = \frac{0,7 \cdot 180}{0,8} = 157,5 \text{кВ} \cdot \text{А}$$

4. Активний та індуктивний опір фазного і нульового захисного провідників на ділянках 1 та 2:

$$r_{\pi} = 0,028 \cdot \left(\frac{85}{95}\right) + 0,028 \cdot \left(\frac{20}{3}\right) = 0,234 \text{Ом}$$

5. Дійсне значення струма однофазного короткого замикання

$$I_{kr} = \frac{220}{\frac{0,141}{3} + \sqrt{(0,165 + 0,234)^2}} = 493,27 \text{А}$$

6. Максимальна напруга на корпусі обладнання відносно землі при замиканні фази на корпус:

$$U_{kmax} = 493,27 \cdot 0,234 = 115,42 \text{В} > 36 \text{В}$$

Умова не виконується, необхідно замінити запобіжник з плавкою вставкою на автоматичний вимикач із струмовим реле, таке рішення дає можливість зменшити час замикання на корпус і підвищити допустиму напругу на корпусі або застосувати повторне заземлення нульового захисного провідника.

Необхідний опір нульового захисного провідника:

$$36 \cdot 2 / ((493,27 \cdot 0,165) - 36) = 1,59 \text{Ом}$$

8.4 Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного занулення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з охорони праці.

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання кваліфікаційної магістерської роботи, призначено для аналітично-інформаційно системи.

В межах України в недостатній мірі представлені у вільному доступі вітчизняні розробки в цій області, оскільки вони являються або власністю певних фірм або є продуктом побудови псевдоаналітично-інформаційної системи (MS Excel має певні елементи побудови аналітично-інформаційних систем).

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих аналітично-інформаційних систем.
- Досліджені аналітично-інформаційні системи.
- На основі отриманих результатів досліджень створена програмна реалізація аналітично-інформаційної системи.

Розроблені під час виконання кваліфікаційної магістерської роботи алгоритми дозволяють успішно вирішувати завдання по розробці та впровадженню аналітично-інформаційних систем.

Розроблене програмне забезпечення має цікавий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань. Також наявні підказки у яких детально розказано про інтерфейс листів.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

Програма реалізована на мові Qlik, яка поєднує у собі елементи C++, SQL та Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для аналітично-інформаційної системи. Це дозволило мінімізувати термін розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку, разом з тим середовище Qlik зарекомендувало себе як якісний та швидкий обробник даних, який постійно покращується. Запропоноване програмне забезпечення поєднує у собі які елементи загального програмного забезпечення, що поставляється із засобами обчислювальної техніки так й спеціального програмного забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програмна реалізація відбувається у двох частинах перший – це сервер розгортання, а другий – це онлайн сторінка у будь-якому браузері останніх версій.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у певних галузях.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Лутц М. Вивчаємо Python, 4-е видання. - Пер. з англ. - СПб .: Символ-Плюс, 2011. - 1280 с.
2. Златопольский Д.М. Основи програмування на мові Python. - М .: ДМК Пресс, 2017. - 284 с.
3. Лутц М. Програмування на Python, том I, 4-е видання. - Пер. з англ. - СПб .: Символ-Плюс, 2011. - 992 с.
4. Лутц М. Програмування на Python, том II, 4-е видання. - Пер. з англ. - СПб .: Символ-Плюс, 2011. - 992 с.
5. Геддіс Т. Починаємо програмувати на Python. - 4-е вид.: 2019. - 768 с.
6. Лучано Рамальо Python. До вершин майстерності. - М .: ДМК Пресс, 2016. - 768 с.
7. Свейгарт, Ел. Автоматизація рутинних завдань за допомогою Python: практичне керівництво для початківців. Пер. з англ. - М .: Вільямс, 2016. - 592 с.
8. Рейтц К., Шлюссер Т. Автостопом по Python. - СПб .: Пітер, 2017. - 336 с .: іл. - (Серія «Бестселери O'Reilly»).
9. Любановіч Білл Простий Python. Сучасний стиль програмування. - СПб .: Пітер, 2016. - 480 с .: - (Серія «Бестселери O'Reilly»).
10. Шелудько, В. М. Мова програмування високого рівня Python. Функції, структури даних, додаткові модулі: навчальний посібник 2017. - 107 с. - ISBN 978-5-9275-2648-2.
11. Доусон М. Програмуємо на Python. - СПб .: Пітер, 2014. - 416 с.
12. Прохоренко Н.А. Python 3 і PyQt. Розробка додатків. - СПб .: БХВ-Петербург, 2012. - 704 с.
13. Selenium with Python [Електронний ресурс] – Режим доступу: <https://selenium-python.readthedocs.io/>
14. Pandas [Електронний ресурс] – Режим доступу:

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

<https://ru.wikipedia.org/wiki/Pandas>

15. Колаборативна фільтрація: як переваги інших користувачів впливають на твої рекомендації [Електронний ресурс] – Режим доступу: <https://vc.ru/marketing/153788-kollaborativnaya-filtraciya-kak-predpochteniya-drugih-polzovateley-vliayut-na-tvoi-rekomendacii#:~:text=Коллаборативная фильтрация – технология прогнозирования предпочтений,а конкретный человек – еще нет.>

16. Колаборативна фільтрація [Електронний ресурс] – Режим доступу: https://ru.wikipedia.org/wiki/Коллаборативная_фильтрация

17. Колаборативна фільтрація [Електронний ресурс] – Режим доступу: <https://habr.com/ru/post/150399/>

18. З чого складається рекомендаційна система YouTube [Електронний ресурс] – Режим доступу: <https://neurohive.io/ru/novosti/iz-chego-sostoit-rekomendatelnaia-sistema-youtube/>

19. Свіжі подробиці про те, як працює рекомендаційна система YouTube в 2020 році [Електронний ресурс] – Режим доступу: <https://indexcall.com/content/svezhie-podrobnosti-o-tom-kak-rabotaet-rekomendatelnaia-sistema-youtube-v-2020-godu>

20. Як працюють рекомендації YouTube? [Електронний ресурс] – Режим доступу: <https://androidinsider.ru/soft/kak-rabotayut-rekomendaczii-youtube.html>

21. Звідки Spotify знає, що нам подобається? [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/imi-journal/откуда-spotify-знает-что-нам-нравится-53ad76db6044>.

22. Від Spotify до власної рекомендаційної системи [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/nuances-of-programming/от-spotify-к-собственной-рекомендательной-системе-bb98593c00c4>.

23. Як Apache Cassandra, Kafka, Storm і Hadoop формують рекомендації користувачам Spotify [Електронний ресурс] – Режим доступу до ресурсу: <https://www.bigdataschool.ru/blog/cassandra-use-case-spotify-recommendation-system.html>.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

24. Рекомендаційні системи: ідеї, підходи, завдання [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/company/jetinfosystems/blog/453792/>.
25. Введення в рекомендаційні системи [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/476222/>.
26. Як працює система рекомендацій Netflix [Електронний ресурс] – Режим доступу до ресурсу: <https://help.netflix.com/ru/node/100639>.
27. Парсинг безкоштовно: 30 програм, сервісів і фреймворків [Електронний ресурс] – Режим доступу до ресурсу: <https://vc.ru/services/174802-parsing-besplatno-30-programm-servisov-i-freymvorkov#:~:text=Парсинг – это автоматизированный процесс сбора,нужно собрать большой объем информации.>
28. Що таке парсинг і як правильно парсити [Електронний ресурс] – Режим доступу до ресурсу: <https://blog.calltouch.ru/chto-takoe-parsing/>.
29. Парсинг даних з сайтів: що це і навіщо він потрібен [Електронний ресурс] – Режим доступу до ресурсу: <https://blog.ringostat.com/ru/parsing-dannyh-s-saytov-chto-eto-i-zachem-on-nuzhen/>.
30. Python's documentation, tutorials, and guides are constantly evolving. – [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.python.org/doc/>
31. Downey A., Elkner J., Meyers Ch. How to Think Like a Computer Scientist: Learning with Python. - Wellesley, Massachusetts: Green Tea Press, 2002. - 290 pp.
32. Tutorialspoint / Python – [Електронний ресурс]. – Режим доступу до ресурсу: <http://www.tutorialspoint.com/python/>
33. Temperley D., Sleator D., Lafferty J. Link Grammar Documentation. CarnegieMellon Univ., Pittsburgh, PA, USA. – [Електронний ресурс]. – Режим доступу до ресурсу: <http://www.link.cs.cmu.edu/link/dict/index.html>
34. New Protege Short Course // Stanford Center for Biomedical Informatics Research. Stanford Univ., California, USA. – [Електронний ресурс]. – Режим доступу до ресурсу: <http://protege.stanford.edu/>
35. Kumar N., Srinathan K., Varma V. Using graph based mapping of co-occurring words and closeness centrality score for summarization evaluation // CICLing

Proceedings of the 13th International Conference on Computational Linguistics and Intelligent Text Processing. – 2012. – Vol. 2, P. 353–365.

36. X-Parser Light. – [Електронний ресурс]. – Режим доступу до ресурсу: <https://skladchik.com/threads/x-parser-light>

37. Обробляємо csv файли - Модуль csv. – [Електронний ресурс]. – Режим доступу до ресурсу: <https://python-scripts.com/import-csv-python>

38. lxml - XML and HTML with Python. – [Електронний ресурс]. – Режим доступу до ресурсу: <http://lxml.de/index.html>

39. Beautiful Soup Documentation. – [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

40. Requests: HTTP for Humans. – [Електронний ресурс]. – Режим доступу до ресурсу: <http://docs.python-requests.org/en/master/#>

41. Вікіпедія. Selenium. – [Електронний ресурс]. – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/Selenium/>

42. Інтернет. – [Електронний ресурс]. – Режим доступу до ресурсу: <https://ua.wikipedia.org/wiki/Інтернет>

43. Zaharia M., Chowdhury M., Franklin M. J., Shenker S., Stoica I. Spark: cluster computing with working sets // HotCloud, 2010. Vol. 10, P. 10

44. Aggarwal C. C. Recommender Systems. The Textbook. Springer International Publishing, 2016. 498 p

45. Ekstrand M. D., Riedl J. T., Konstan J. A. Collaborative Filtering Recommender Systems // Foundations and Trends® in Human-Computer Interaction, 2011. Vol. 4, No. 2. P. 81-173.

46. Gabriel H. H., Spiliopoulou M., Nanopoulos A. Eigenvector-based clustering using aggregated similarity matrices // Proceedings of the 2010 ACM Symposium on Applied Computing, 2010. P. 1083-1087

47. Schedl M., Flexer A., Urbano J. The neglected user in music information retrieval research // Journal of Intelligent Information Systems, 2013. Vol. 41, Issue 3. P. 523-539.

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		109

48. Aggarwal C. C. Data mining. The Textbook. Springer International Publishing, 2015. 734 p.

49. Ricci F., Rokach L., Shapira B., Kantor P.B. Recommender Systems Handbook. Springer US, 2011. 842 p

50. Celma O. Music Recommendation and Discovery. Springer-Verlag Berlin Heidelberg, 2010. 194 p

51. Природне і штучне освітлення” – ДБН В.2.5-28:2018 – [Електронний ресурс]. – Режим доступу до ресурсу: https://ledeffect.com.ua/images/_branding/dbn2018.pdf

52. Vucetic S., Obradovic Z. Performance Controlled Data Reduction for Knowledge Discovery in Distributed Databases // Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications, 2000. P. 29-39.

53. Закон України “Про охорону праці”. – [Електронний ресурс]. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12#Text>

54. Qlik help. – [Електронний ресурс]. – Режим доступу до ресурсу: <https://help.qlik.com/>

55. «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» – [Електронний ресурс]. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508-18#Text>

56. Методичні рекомендації до виконання розділу "Заходи з охорони праці та техніки безпеки" випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти для здобувачів вищої освіти спеціальностей 123 "Комп’ютерна інженерія" та 122 "Комп’ютерні науки" / М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т, каф. кібербезпеки та програм. забезпечення; [укл. О.В. Оришака, К.М. Марченко]. - Кропивницький: ЦНТУ, 2022. — 19 с. [Електронний ресурс]. – Режим доступу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12240> (дата звернення 19.09.22).

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

57. Інформаційно-аналітична система (ІАС) – [Електронний ресурс]. – Режим доступу до ресурсу:
<https://www.kspu.edu/About/DepartmentAndServices/DigitalInfrastructure/Products/IAS.aspx?lang=uk>

58. Сакулін В.П., Шептовіцький В.М. Безпека праці під час монтажу та експлуатації електроустановок / В.П.Сакулін, В.М.Шептовіцький. - Л.: "Колос", 1973. - 238 с.

59. Інформаційно-аналітична система – [Електронний ресурс]. – Режим доступу до ресурсу:
https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%BE-%D0%B0%D0%BD%D0%B0%D0%BB%D1%96%D1%82%D0%B8%D1%87%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0

60. Qlick View – [Електронний ресурс]. – Режим доступу до ресурсу:
https://help.qlik.com/qlikview/May2022/Content/QV_HelpSites/what-is.htm

61. Qlick Sense – [Електронний ресурс]. – Режим доступу до ресурсу:
https://help.qlik.com/qliksense/May2022/Content/QS_HelpSites/what-is.htm

62. Tableau – [Електронний ресурс]. – Режим доступу до ресурсу:
<https://www.tableau.com/>

63. Power BI – [Електронний ресурс]. – Режим доступу до ресурсу:
<https://powerbi.microsoft.com/>

64. Metabase – [Електронний ресурс]. – Режим доступу до ресурсу:
<https://www.metabase.com/>

65. Wang A. The Shazam music recognition service // Communications of the ACM, 2006. P. 44-48

66. Інтегрована інформаційно-аналітична система: проект та концепція – [Електронний ресурс]. – Режим доступу до ресурсу:
https://pidru4niki.com/1031062453915/informatika/integrovana_informatsiyno-analitichna_sistema_proekt_kontseptsiya

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		111

67. Методичні вказівки по виконанню розрахунків з використанням персональних ЕОМ IBM–сумісного типу, 2–ге видання, перероблене та доповнене [Електронний ресурс] – Режим доступу до ресурсу: http://dspace.kntu.kr.ua/jspui/bitstream/123456789/8769/1/Zanul_2019_pub.pdf.(дата звернення: 30.11.2022)

68. Аналіз умов праці на робочому місці користувача ПК [Електронний ресурс] – Режим доступу до ресурсу: https://cpo.stu.cn.ua/Oksana/dipl_bak/140.html.(дата звернення: 30.11.2022)

69. Розрахунок опору природного заземлювача. Розрахунок заземлюючих пристроїв. Послідовне розташування електродів [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: <https://tigerdoor.ru/uk/dizajjn/raschet-soprotivleniya-estestvennogo-zazemlitelya-raschet/>.(дата звернення: 30.11.2022)

70. Розрахунки з електробезпеки [Електронний ресурс] – Режим доступу до ресурсу: https://cpo.stu.cn.ua/Oksana/rozrah_rozd_OP_DP_bak_spec_mag/90.html.(дата звернення: 30.11.2022)

71. Вимоги до електробезпеки у офісних приміщеннях з комп’ютерною технікою [Електронний ресурс] – Режим доступу до ресурсу: <https://cpo.stu.cn.ua/Oksana/posibnik/1140.html>.(дата звернення: 30.11.2022)

72. Охорона праці в офісі. Вимоги до робочого місця офісного працівника [Електронний ресурс] – Режим доступу до ресурсу: <https://gc.ua/uk/oxorona-praci-v-ofisi-vimogi-do-robochogo-miscya-ofisnogo-pracivnika/>.(дата звернення: 30.11.2022)

					ВКМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

Технічне завдання

Зміст

1	Найменування та область застосування.....
2	Підстава для розробки.....
3	Мета та призначення розробки.....
4	Джерела розробки.....
5	Технічні вимоги.....
5.1	Вміст проекту.....
5.2	Показники призначення.....
5.3	Вимоги до функціональних характеристик.....
5.4	Вимоги до архітектури.....
5.5	Вимоги до надійності.....
5.6	Умови експлуатації.....
5.7	Вимоги до складу та параметрів технічних засобів.....
5.8	Вимоги до інформаційної і програмної сумісності.....
5.8.1	Обладнання.....
5.8.2	Мова програмування.....
5.8.3	Вхідні дані.....
5.8.4	Вихідні дані.....
1. 6	Вимоги до програмної документації.....
7	Економічні вимоги.....
8	Вимоги щодо охорони праці.....
9	Перелік документів, що розробляються.....
10	Етапи розробки.....
11	Порядок контролю та приймання.....

KMP-123.22.0026.00.00.ПЗ

Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Чабан О. О.			Дослідження та програмна реалізація інформаційно- аналітичної системи для аналізу великих даних	Літ.	Аркуш	Аркушів
Перев.		Мелешко Є.В.				М	1	1
Н.контр.		Гермак В.С.			ЦНТУ КІ-21М			
Затв.		Смірнов О.А.						

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку аналітично-інформаційного додатку - аналізу продажів.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (наказ № 19-13 від 17.08.2022р.).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є підвищення ефективності роботи системи бухгалтерського обліку на основі впровадження нових інформаційних технологій і застосування сучасних засобів програмування.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є бази даних, які беруться у компанії “ViTra”

5 Технічні вимоги

5.1 Вміст проекту

Складовими розробки є:

– аналіз існуючих комп’ютеризованих аналітично-інформаційних систем на предмет їхньої відповідності сучасним вимогам;

					KMP-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

- вибір і обґрунтування методики побудови додатків і засобів їхньої реалізації;
- розробка структур даних і механізму їхньої взаємодії, робочих форм, засобів і правил бізнесу;
- техніко-економічне обґрунтування доцільності прийнятого до розробки аналітично-інформаційного додатку;
- аналіз умов праці програміста в лабораторії К505;
- розробка програми, яка реалізує алгоритми и методики надання та створення аналітично-інформаційних послуг.

5.2 Показники призначення

Система повинна забезпечувати:

- простий, інтуїтивно зрозумілий інтерфейс з користувачем;
- цілісність даних в таблицях БД.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення повинно буди максимально влучним, дозаяти порівнювати періоди у різні проміжки часу та усі обчислення мусять буди швидкими.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати архітектуру клієнт/сервер, реляційні БД і об'єктно-орієнтовані засоби розробки.

5.5 Вимоги до надійності

Компонент повинен використати існуючі угоди по стандартним викликам

					КМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

процедур, функцій, засобів і форм, визначених технічною документацією на середовище розробки.

5.6 Умови експлуатації

Автоматизовані робочі місця користувачів системи повинні задовольняти наступним умовам експлуатації:

- температура повітря: 10-40⁰ С;
- відносна вологість повітря при 20⁰ С до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу і параметрів технічних засобів

Компонент повинен бути реалізований таким чином щоб без проблем можна було користуватись додатком як і за допомогою ПК, так і за допомогою смартфонів.

5.8 Вимоги до інформаційної та програмної сумісності

Сумісність програмного забезпечення повинна бути забезпечена за рахунок його реалізації засобами об'єктно-орієнтованої СУБД, працюючої під управлінням ОС Windows 10 (рекомендується СУБД Mysql).

5.8.1 Обладнання

Будь-який комп'ютер з доступом у мережу інтернет

5.8.2 Мова програмування

Середовище Qlik Sense

					КМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена в вигляді опису структури даних, інструкції користувача, а також текстів вхідних модулів програмного забезпечення в відповідності з ЄСПД.

7 Економічні вимоги

7.1 Для програми ядра виконавчої системи моделі необхідно виробити функціонально-вартісний аналіз варіантів розробки

7.2 Виконати розрахунок кошторису витрат на реалізацію прийнятого варіанту з урахуванням розцінок на 1 жовтня 2022 р.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинен бути наведений аналіз умов праці програміста в лабораторії К505 ЦНТУ і вироблений розрахунок категорії важкості праці програміста.

9 Перелік документів, які необхідно розробити

- | | |
|-------------------------------|------------|
| – Наукова новизна | – 1 аркуш. |
| – Структурна схема системи | – 1 аркуш. |
| – Функціональна схема системи | – 1 аркуш. |
| – Діаграма процесів | – 1 аркуш. |

					КМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 107 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок-схем алгоритмів роботи програмного забезпечення компоненту.

10.4 Побудова схем взаємодії структур даних.

10.5 Створення прототипу компоненту. Створення програмного продукту.

10.6 Відлагодження компоненту, аналіз отриманих результатів.

10.7 Робота над питаннями охорони праці і техніки безпеки.

10.8 Розрахунки по техніко-економічному обґрунтуванню (остаточні).

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю і приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12. 2022 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 20.12.2022 р.

					КМР-123.22.0026.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи

за другим (магістерським) рівнем вищої освіти

_____ **Мелешко Є.В.**

**Дослідження та програмна реалізація інформаційно-аналітичної
системи для аналізу великих даних**

Лістинг програми

Код документу 12

Носій: DVD-диск

Загальна кількість аркушів: 77

Літера: РП

Кропивницький – 2022 року

Лістинг №1 - Додаток для забору даних

```
SET ThousandSep = '';
SET DecimalSep = ',';
SET MoneyThousandSep = '';
SET MoneyDecimalSep = ',';
SET MoneyFormat = '# ##0,00 €;-# ##0,00 €';
SET TimeFormat = ' hh:mm:ss ';
SET DateFormat = 'DD.MM.YYYY';
SET TimestampFormat = 'DD.MM.YYYY hh:mm:ss [. fff ]';
SET FirstWeekDay = 0;
SET BrokenWeeks = 1;
SET ReferenceDay = 0;
SET FirstMonthOfYear = 1;
SET CollationLocale = ' uk -UA';
SET CreateSearchIndexOnReload =1;
SET
MonthNames='січ.;лют.;бер.;квіт.;трав.;черв.;лип.;серп.;вер.;жовт.;лист.;груд
.';
SET
LongMonthNames='січень;лютий;березень;квітень;травень;червень;липень;серпень;
вересень;жовтень;листопад;грудень';
SET DayNames = ' пн;вт;ср;чт;пт;сб;нд ';
SET LongDayNames='понеділок;вівторок;середа;четвер;п'ятниця;субота;неділя';
SET NumericalAbbreviation='3:k;6:M;9:G;12:T;15:P;18:E;21:Z;24:Y;-3:m;-6:μ;-
9:n ;-12:p;-15:f;-18:a;-21:z;-24:y';
SET DB_XLS_PATH =
'lib://QlikViewDocs_Folder/chaban/ViTRA/04_DataBase/01_DataFiles/Excel/';
SET BASE_QVD_PATH =
'lib://QlikViewDocs_Folder/chaban/ViTRA/04_DataBase/01_DataFiles/';
SET BASE_QVDTr_PATH = ' lib :// QlikData_chaban / ViTRA /04_DataBase/
Translation /';
SET QVD_DB_PATH =
'lib://QlikViewDocs_Folder/chaban/ViTRA/04_DataBase/02_QVD_DB/01_Dictionary/'
;
LET vStartPeriod = YearStart ( AddYears ( Date ( Now ()), -4));
LET vEndPeriod = Date ( Now () -1);
LET vCurrentDate = Date ( Now () -1);
SQL set dateformat dmy ;
set DBName = ' vitra ';
Cnp_Currencies :
load *;
SQL select
    _IDRef                as Currencies_KEY
    ,_Version             as ВерсіяДаних
    ,_Marked              as МіткаВидаленняCurrencies
    ,_IsMetadata          as ПередвизначеньCurrencies
    ,_Code                as КодCurrencies
    ,_Description        as Currencies
    ,_Fld360              as FullDescription
    ,_Fld361              as SpellingParametersRO
    ,_Fld362              as SpellingParametersEN
FROM "$( DBName )" .dbo. "_Reference26" as Cnp_Currencies ;
Store * from [ Cnp_Currencies ] into '$(BASE_QVD_PATH)01_Dictionary/
Cnp_Currencies.qvd '
    ( qvd );
Let vCurrentdate = Today ();
Let vStartPeriod = MakeDate ( Year ( Today ())-4);
Let vEndPeriod = Today ();
Let vRollBackMonths = 6;
LET vDefaultCurrency = Peek ( ' Currencies ', 0, '
Cnp_Currencies ');
LET vDefaultCurrencyLabel = Peek ( ' FullDescription ', 0, '
Cnp_Currencies ');
LET vBaseCurrency = Peek ( ' Currencies ', 0, '
```

```

Спр_Currencies ');
Параметри:
Load
'$( vStartPeriod )' as Початок Періоду ,
'$( vEndPeriod )' as КінецьПеріоду ,
'$( vCurrentdate )' as Поточна Дата ,
'$( vRollBackMonths )' as vRollBackMonths ,
'$( vDefaultCurrency )' as Валюта За Замовчуванням ,
'$( vDefaultCurrencyLabel )' as Мітка Валюти За Замовчуванням ,
'$( vBaseCurrency )' as Валюта Базова
AutoGenerate 1;
Store Параметри into [$(BASE_QVD_PATH)01_Dictionary/ Параметри.qvd ] ( qvd );

// Exit Script ;
Drop Tables Параметри , Спр_Currencies ;
load *;
SQL select
  _IDRRef      as  Nomenclature_KEY //!
, _Version    as  ВерсіяДаних
, _Marked     as  МіткаВидаленняNomenclature
, _IsMetadata as  ЗумовленьNomenclature
, _ParentIDRRef as  БатькоNomenclature_KEY //!
, _Folder     as  ЦеГрупаNomenclature //!
, _Code       as  КодNomenclature //!
, _Description as  Nomenclature //!
, _Fld393     as  FullDescription
, _Fld394RRef as  NomenclatureType_KEY
, _Fld4573    as  ExpandByWeight
, _Fld401     as  ExpandBySeries
, _Fld5206RRef as  CustomGroup_KEY
, _Fld397RRef as  ReportMeasure_KEY
, _Fld4702RRef as  NomenclatureSectors_KEY
, _Fld403RRef as  OriginCountry_KEY //!
, _Fld4707    as  SupplierCode
, _Fld395RRef as  BaseMeasure_KEY
, _Fld5237    as  EnterDate
, _Fld405RRef as  NomenclatureGroup_KEY //!
, _Fld5415    as  OldParent
, _Fld399     as  Comment
, _Fld5631RRef as  BSOGroup_KEY
, _Fld396RRef as  BalanceMeasure_KEY
, _Fld407RRef as  VATRate_KEY
, _Fld398     as  MarkingGoods
, _Fld400     as  ExpandByCharacteristics
, _Fld402RRef as  Picture_KEY
, _Fld404     as  Weight
, _Fld406     as  RefreshPricesOnReceipt
, _Fld408     as  DistributeRound
, _Fld4131    as  OfficialCode
, _Fld3747    as  CurrentPrice
, _Fld4706    as  SupplierName
, _Fld4738RRef as  ShowOnWebSite_KEY
, _Fld5240RRef as  Creator_KEY
, _Fld5630    as  BSO
, _Fld5948    as  Complect
, _Fld9729    as  InternetLinkToPicture
, _Fld9730RRef as  Picture1_KEY
, _Fld9731RRef as  Picture2_KEY
, _Fld9732RRef as  Picture3_KEY
, _Fld9733RRef as  Picture4_KEY
, _Fld9734    as  InternetLinkToPicture1
, _Fld9735    as  InternetLinkToPicture2
, _Fld9736    as  InternetLinkToPicture3
, _Fld9737    as  InternetLinkToPicture4

```

```

    ,_Fld9738 as      DescriptionRu
    ,_Fld9739 as     DescriptionEn
    ,_Fld9740 as     SpecificationRo
    ,_Fld9741 as     SpecificationRu
    ,_Fld9742 as     SpecificationEn
    ,_Fld9914 as     DoNotUnloadFolderOnWebSite
    ,_Fld9915 as     AlwaysShowOnWebSite
    ,_Fld9916 as     InstallPrice
    ,_Fld9927 as     Warranty
    ,_Fld9928 as     PlanningAndProject
    ,_Fld9929 as     DoNotShowOnWebSite
    ,_Fld9976 as     Production
    ,_Fld9977 as     Rack
    ,_Fld9978 as     GroupIDWebSite
    ,_Fld9979 as     GroupNameWebSite
    ,_Fld9980 as     ModifiedDate
    ,_Fld10023 as    PAL
    ,_Fld10504RRef as      PromoFolder_KEY
    ,_Fld11923RRef as      Supplier_KEY //!
FROM "$ ( DBName ) ".dbo."_Reference37" as Cnp_Nomenclature ;
Store * from [ Cnp_Nomenclature ] into '$ (BASE_QVD_PATH) 01_Dictionary/
Cnp_Nomenclature.qvd ' ( qvd );
Drop Table Cnp_Nomenclature ;
Cnp_NomenclatureGroups :
load *;
SQL select
    _ IDRRef      as      NomenclatureGroups_KEY
    ,_ Version    as      ВерсіяДаних
    ,_ Marked     as      МіткаВидаленняNomenclatureGroups
    ,_ IsMetadata as      ЗумовленьNomenclatureGroups
    ,_ Code       as      КодNomenclatureGroups
    ,_ Description as      NomenclatureGroups
FROM "$ ( DBName ) ".dbo."_Reference39" as Cnp_NomenclatureGroups ;
Store * from [ Cnp_NomenclatureGroups ] into '$ (BASE_QVD_PATH) 01_Dictionary/
Cnp_NomenclatureGroups.qvd ' ( qvd );
Drop Table Cnp_NomenclatureGroups ;
Cnp_NomenclatureTypes :
load *;
SQL select
    _ IDRRef      as      NomenclatureTypes_KEY
    ,_ Version    as      ВерсіяДаних
    ,_ Marked     as      МіткаВидаленняNomenclatureTypes
    ,_ IsMetadata as      ЗумовленьNomenclatureTypes
    ,_ Code       as      КодNomenclatureTypes
    ,_ Description as      NomenclatureTypes
    ,_Fld452RRef as      NomenclatureType_KEY
FROM "$ ( DBName ) ".dbo."_Reference43" as Cnp_NomenclatureTypes ;
Store * from [ Cnp_NomenclatureTypes ] into '$ (BASE_QVD_PATH) 01_Dictionary/
Cnp_NomenclatureTypes.qvd ' ( qvd );
Drop Table Cnp_NomenclatureTypes ;
Cnp_CountriesClassifier :
load
    *,
    Num ( КодCountriesClassifier , '000') as КодCountries
;
SQL select
    _ IDRRef      as      CountriesClassifier_KEY
    ,_ Version    as      ВерсіяДаних
    ,_ Marked     as      МіткаВидаленняCountriesClassifier
    ,_ IsMetadata as      ЗумовленьCountriesClassifier
    ,_ Code       as      КодCountriesClassifier
    ,_ Description as      CountriesClassifier
FROM "$ ( DBName ) ".dbo."_Reference25" as Cnp_CountriesClassifier ;

```

```

Store * from [ Cnp_CountriesClassifier ] into '$(BASE_QVD_PATH)01_Dictionary/
Cnp_CountriesClassifier.qvd '          ( qvd );
Drop Table Cnp_CountriesClassifier ;
Cnp_ObjectsCategory :
load *;
SQL select
  _IDRRef      as  ObjectsCategory_KEY
  ,_ Version   as  ВерсіяДаних
  ,_ Marked    as  МіткаВидаленняObjectsCategory
  ,_ IsMetadata as  ЗумовленьObjectsCategory
  ,_ Code      as  КодObjectsCategory
  ,_ Description as  ObjectsCategory
  ,_Fld453RRef as  CategoryDestination_KEY
FROM "$ ( DBName )".dbo."_Reference44" as Cnp_ObjectsCategory ;

Store * from [ Cnp_ObjectsCategory ] into '$ (BASE_QVD_PATH) 01_Dictionary /
Cnp_ObjectsCategory.qvd '          ( qvd );
Drop Table Cnp_ObjectsCategory ;
PC_ObjectsCategory :
load *;
SQL select
  _Fld2791_TYPE as  Object_TYPE
  ,_Fld2791_RTRef as  Object_RTRef
  ,_Fld2791_RRRef as  Object_KEY
  ,_Fld2792RRef as  Category_KEY
  ,_ SimpleKey    as  _ SimpleKey
FROM "$ ( DBName )".dbo."_InfoRg2790" as PC_ObjectsCategory ;
Store * from [ PC_ObjectsCategory ] into '$(BASE_QVD_PATH)01_Dictionary/
PC_ObjectsCategory.qvd '          ( qvd );

Drop Table PC_ObjectsCategory ;
Cnp_Contractors :
load *;
SQL select
  _IDRRef      as  Contractors_KEY
  ,_ Version   as  ВерсіяДаних
  ,_ Marked    as  МіткаВидаленняContractors
  ,_ IsMetadata as  ЗумовленьContractors
  ,_ ParentIDRRef as  БатькоContractors_KEY
  ,_ Folder    as  ЦеГрупаContractors
  ,_ Code      as  КодContractors
  ,_ Description as  Contractors
  ,_Fld321RRef as  PersonType_KEY
  ,_Fld322 as  CodeFiscal
  ,_Fld323 as  OfficialName
  ,_Fld324RRef as  DefaultContract_KEY
  ,_Fld325RRef as  DefaultAccount_KEY
  ,_Fld326RRef as  DefaultContact_KEY
  ,_Fld327 as  Comment
  ,_Fld328 as  DocumentSeries
  ,_Fld329 as  DocumentDate
  ,_Fld330 as  DocumentNumber
  ,_Fld331 as  DocumentOrganization
  ,_Fld4126 as  IDCardSeries
  ,_Fld332 as  VATCode
  ,_Fld333 as  IDCardNumber
  ,_Fld334 as  IssuingOffice
  ,_Fld335 as  IssuingDate
  ,_Fld336 as  RegistrationNumber
  ,_Fld337 as  RegistrationDate
  ,_Fld338RRef as  Employee_KEY
  ,_Fld339 as  NonResident
  ,_Fld4127RRef as  IdentityDocumentType_KEY
  ,_Fld4128 as  VATRegistrationDate

```

```

, _Fld5203 as Buyer
, _Fld5204RRef as MainResponsiblePerson_KEY
, _Fld5384 as EnterDate
, _Fld4674RRef as Creator_KEY
, _Fld9390RRef as CRM_ТипВідносин_KEY
, _Fld9391RRef as ОсновнийВидДіяльності_KEY
, _Fld9392RRef as ОсновнеКонтактнеОбличчя_KEY
, _Fld9393RRef as Region_KEY
, _Fld11922RRef as ContractorCategory_KEY
//, _Fld11932RRef as OriginCountry_KEY
FROM "$( DBName )".dbo."_Reference22" as Cnp_Contractors ;
Store * from [ Cnp_Contractors ] into '$(BASE_QVD_PATH)01_Dictionary/
Cnp_Contractors.qvd ' ( qvd );
Drop Table Cnp_Contractors ;
Cnp_Managers :
load *;
SQL select
_ IDRRRef as Managers_KEY
, _ Version as ВерсіяДаних
, _ Marked as МіткаВидаленняManagers
, _ IsMetadata as ЗумовленийManagers
, _ ParentIDRRRef as БатькоManagers_KEY
, _ Folder as ЦеГрупаManagers
, _ Code as КодManagers
, _ Description as Managers
, _Fld3754 as ManagerCode
, _Fld3755RRef as Employee_KEY
, _Fld4710 as TelephoneNumbersMobile
FROM "$( DBName )".dbo."_Reference3751" as Cnp_Managers ;
Store * from [ Cnp_Managers ] into '$(BASE_QVD_PATH)01_Dictionary/
Cnp_Managers.qvd ' ( qvd );
Drop Table Cnp_Managers ;
PC_ExchangeRates :
load
Date ( AddYears ( Date ( ПеріодExchangeRates ), -2000)) as
ПеріодExchangeRates ,
Currency_KEY ,
Rate ,
Factor
;
SQL select
_Period as ПеріодExchangeRates
, _Fld2734RRef as Currency_KEY
, _Fld2735 as Rate
, _Fld2736 as Factor
FROM "$( DBName )".dbo."_InfoRg2733" as PC_ExchangeRates ;
Store * from [ PC_ExchangeRates ] into '$(BASE_QVD_PATH)01_Dictionary/
PC_ExchangeRates.qvd ' ( qvd );
Drop Table PC_ExchangeRates ;
Cnp_OperationsTypes :
load *;
SQL select
_ IDRRRef as OperationsTypes_KEY
, _ Version as ВерсіяДаних
, _ Marked as МіткаВидаленняOperationsTypes
, _ IsMetadata as Зумовлений OperationsTypes
, _ Code as КодOperationsTypes
, _ Description as OperationsTypes
FROM "$( DBName )".dbo."_Reference45" as Cnp_OperationsTypes ;
Store * from [ Cnp_OperationsTypes ] into '$(BASE_QVD_PATH)01_Dictionary/
Cnp_OperationsTypes.qvd ' ( qvd );
Drop Table Cnp_OperationsTypes ;
Cnp_SaleTypes :
load *;

```

```

SQL select
    _ IDRRef      as  SaleTypes_KEY
    ,_ Version    as  ВерсіяДаних
    ,_ Marked     as  МіткаВидаленняSaleTypes
    ,_ IsMetadata as  ЗумовленийSaleTypes
    ,_ Code       as  КодSaleTypes
    ,_ Description as  SaleTypes
FROM "$ ( DBName ) ".dbo."_Referencell924" as Cnp_SaleTypes ;
Store * from [ Cnp_SaleTypes ] into '$(BASE_QVD_PATH)01_Dictionary/
Cnp_SaleTypes.qvd '          ( qvd );
Drop Table Cnp_SaleTypes ;
PH_NomenclatureCategoriesPlanSales :
LOAD
" Denumire ( categorie )" as Category _
" Ianuarie 2018" as January ,
" Februarie 2018" as February ,
" Martie 2018" as March ,
" Aprilie 2018" as April ,
" Mai 2018" as May ,
" Iunie 2018" as June ,
" Iulie 2018" as July ,
" August 2018" as August ,
" Septembrie 2018" as September ,
" Octombrie 2018" as Жовтень ,
" Noiembrie 2018" as November ,
" Decembrie 2018" as December ,
'2018' as ПiкПлану
FROM [$(DB_XLS_PATH)План 2018.xlsx]
( ooxml , embedded labels , table is [План за категоріями])
Where not IsNull (" Denumire ( categorie )" ) i " Denumire ( categorie )" <>
'TOTAL'
;
Concatenate ( PH_NomenclatureCategoriesPlanSales )
LOAD
" Denumire ( categorie )" as Category _
" Ianuarie 2019" as January ,
" Februarie 2019" as February ,
" Martie 2019" as March ,
" Aprilie 2019" as April ,
" Mai 2019" as May ,
" Iunie 2019" as June ,
" Iulie 2019" as July ,
" August 2019" as August ,
" Septembrie 2019" as September ,
" Octombrie 2019" as Жовтень ,
" Noiembrie 2019" as November ,
" Decembrie 2019" as December ,
'2019' as ПiкПлану
FROM [$(DB_XLS_PATH)План 2019.xlsx]
( ooxml , embedded labels , header is 1 lines , table is [План за
категоріями])
Where not IsNull (" Denumire ( categorie )" ) i " Denumire ( categorie )" <>
'TOTAL'
;
Store * from PH_NomenclatureCategoriesPlanSales into
'$(BASE_QVD_PATH)/04_Plans/PH_NomenclatureCategoriesPlanSales.qvd'
(qvd );
Drop Table PH_NomenclatureCategoriesPlanSales ;
PH_ManagersPlanSales :
LOAD
    Manager ,
    ianuarie as January ,
    februarie as February ,
    martie as March ,

```

```

    aprilie as April ,
    mai as May ,
    iunie as June ,
    iulie as July ,
    august as August ,
    septembrie as September ,
    жовтень as ЖОВТЕНЬ ,
    noiembrie as November ,
    decembrie as December ,
'2018' as ПикПлану
FROM [$(DB_XLS_PATH)План 2018.xlsx]
( ooxml , embedded labels , header is 1 lines , table is [План менеджерів])
Where not IsNull ( Manager ) and Manager <> ' Total ' ;
Concatenate ( PH_ManagersPlanSales )
LOAD
    Manager ,
    ianuarie as January ,
    februarie as February ,
    martie as March ,
    aprilie as April ,
    mai as May ,
    iunie as June ,
    iulie as July ,
    august as August ,
    septembrie as September ,
    жовтень as ЖОВТЕНЬ ,
    noiembrie as November ,
    decembrie as December ,
'2019' as ПикПлану
FROM [$(DB_XLS_PATH)План 2019.xlsx]
( ooxml , embedded labels , header is 1 lines , table is [План менеджерів])
Where not IsNull ( Manager ) and Manager <> ' Total ' ;
Store * from PH_ManagersPlanSales into '$(BASE_QVD_PATH)/04_Plans/
PH_ManagersPlanSales.qvd ' ( qvd );
Drop Table PH_ManagersPlanSales ;
let vIncPeriod = vStartPeriod ;
Do
    let DateFrom = vIncPeriod ;
    let DateTo = AddMonths ( vIncPeriod , 1);
    let vPeriod = Year ( vIncPeriod ) & '_' & text ( num ( Month ( vIncPeriod
), '00')));
    let InRollBack = vIncPeriod >= AddMonths ( vEndPeriod , - vRollBackMonths
);
    if InRollBack or
isnull(FileSize('$(BASE_QVD_PATH)02_SourceData\01_Sales\PH_Продажи_$(vPeriod)
.qvd')) then
        TRACE PH_Продажи_$( vPeriod );
        let vDateFrom = Addyears ( DateFrom , 2000);
        let vDateTo = Addyears ( DateTo , 2000);
[ PH_Sales ]:
    load
        *,
        Date ( Floor ( Addyears ( ПеріодDiscountsAnalisy , -2000))) as
Дата;
        SQL select
            _Period as ПеріодDiscountsAnalisy
, _RecorderTRef as РеєстраторDiscountsAnalisy_TRef
, _RecorderRRef as РеєстраторDiscountsAnalisy_KEY
, _LineNo as НомерСтрокаDiscountsAnalisy
, _Active as АКТИВНІСТЬDiscountsAnalisy
, _Fld4056RRef as Division_KEY
, _Fld4057RRef as Nomenclature_KEY
, _Fld4058RRef as Characteristic_KEY
, _Fld4059RRef as Warehouse_KEY

```

```

, _Fld4060RRef      as  Contract_KEY
, _Fld4061RRef      as  CardOwner_KEY
, _Fld4062RRef      as  DiscountCard_KEY
, _Fld4063          as  DiscountPercent
, _Fld4064_TYPE     as  Document_TYPE
, _Fld4064_RTRef    as  Document_RTRef
, _Fld4064_RRRef    as  Document_KEY
, _Fld5652RRef      as  Manager_KEY
, _Fld4065          as  Quantity
, _Fld4066          as  Sum_
, _Fld4067          as  SumManagement
, _Fld4068          as  Cost
, _Fld4069          as  CostManagement
, _Fld4070          as  VATSum
, _Fld4071          as  VATSumManagement
, _Fld4072          as  DiscountSumManagement
, _Fld4073          as  DiscountSum
, _Fld4074RRef      as  OperationType_KEY
, _Fld4075          as  SumByCurrencyOperations
FROM "$ ( DBName ) ".dbo."_AccumRg4055" as PH_DiscountsAnalysis
where PH_DiscountsAnalysis ._ Period >='$( vDateFrom )' and PH_
DiscountsAnalysis ._ Period <'$( vDateTo )';
Store * from [ PH_Sales ] into
[$(BASE_QVD_PATH)02_SourceData\01_Sales\PH_Продажі_$(vPeriod).qvd] (
qvd );
Drop Table [ PH_Sales ];
end if
LET vIncPeriod = AddMonths ( vIncPeriod , 1);
LOOP WHILE NOT InMonth ( vIncPeriod , vEndPeriod , 1);
let vIncPeriod = vStartPeriod ;
Do
let DateFrom = vIncPeriod ;
let DateTo = AddMonths ( vIncPeriod , 1);
let vPeriod = Year ( vIncPeriod ) & '_' & text ( num ( Month ( vIncPeriod
), '00'));
let InRollBack = vIncPeriod >= AddMonths ( vEndPeriod , - vRollBackMonths
);
if InRollBack or
isnull(FileSize('$(BASE_QVD_PATH)02_SourceData\02_Orders\PH_Замовлення_$(vPer
iod).qvd')) then
TRACE PH_Замовлення_$( vPeriod );
let vDateFrom = Addyears ( DateFrom , 2000);
let vDateTo = Addyears ( DateTo , 2000);
[ PH_CustomerDocumentOrders ]:
load
*,
Date ( Floor ( Addyears ( ПеріодCustomerDocumentOrders , -2000)))
as Дата;
SQL select
_Period      as  ПеріодCustomerDocumentOrders
, _RecorderTRef  as  РеєстраторCustomerDocumentOrders_TRef
, _RecorderRRef  as  РеєстраторCustomerDocumentOrders_KEY
, _LineNo       as  НомерСтрокаCustomerDocumentOrders
, _Active       as  АктивністьCustomerDocumentOrders
, _RecordKind   as  Вид Руху
, _Fld2988RRef  as  DocumentOrder_KEY
, _Fld2989RRef  as  Nomenclature_KEY
, _Fld2990RRef  as  Characteristic_KEY
, _Fld2991      as  Price
, _Fld2992RRef  as  VATRate_KEY
, _Fld2993RRef  as  Measure_KEY
, _Fld2994      as  DiscountByHand
, _Fld2995      as  DiscountAutomatic
, _Fld2996RRef  as  ConditionDiscountAutomatic_KEY

```

```

, _Fld2997 as ValueOfCondition
, _Fld2998 as DiscountRestriction
, _Fld2999 as Quantity
, _Fld3000 as Sum _
, _Fld3001 as SumManagement
FROM "$( DBName )".dbo."AccumRg2987" as PH_CustomerDocumentOrders
where PH_CustomerDocumentOrders ._ Period >='$( vDateFrom )' and PH_
CustomerDocumentOrders ._ Period <'$( vDateTo )';
Store * from [ PH_CustomerDocumentOrders ] into
[$(BASE_QVD_PATH)02_SourceData\02_Orders\PH_Замовлення_$(vPeriod).qvd]
( qvd );
Drop Table [ PH_CustomerDocumentOrders ];
end if
LET vIncPeriod = AddMonths ( vIncPeriod , 1);
LOOP WHILE NOT InMonth ( vIncPeriod , vEndPeriod , 1);
let vIncPeriod = vStartPeriod ;
Do
let DateFrom = vIncPeriod ;
let DateTo = AddMonths ( vIncPeriod , 1);
let vPeriod = Year ( vIncPeriod ) & '_' & text ( num ( Month ( vIncPeriod
), '00')));
let InRollBack = vIncPeriod >= AddMonths ( vEndPeriod , - vRollBackMonths
);
let vDateFrom = Addyears ( DateFrom , 2000);
let vDateTo = Addyears ( DateTo , 2000);
if InRollBack or
isnull(FileSize('$(BASE_QVD_PATH)02_SourceData\01_Sales\Док_ПродажТовару_$(vP
eriod).qvd')) then
TRACE Док_ПродажТовару _$( vPeriod );
[ Док_GoodsSale ]:
load
*,
Date ( Floor ( Addyears ( ДатаGoodsSale , -2000))) as Дата;

SQL select
_ IDRRef as GoodsSale_KEY
, _ Version as ВерсіяДаних
, _ Marked as МіткаВидаленняGoodsSale
, _ Date_Time as ДатаGoodsSale
, _ NumberPrefix as _ NumberPrefix
, _ Number as Номер
, _ Posted as Проведено
, _Fld1434RRef as Organization_KEY
, _Fld1435RRef as Creator_KEY
, _Fld1437RRef as Division_KEY
, _Fld1438RRef as OperationType_KEY
, _Fld1439RRef as Warehouse_KEY
, _Fld1440RRef as Contractor_KEY
, _Fld1441RRef as Contract_KEY
, _Fld1442 as Sum _
, _Fld1443_TYPE as PriceType_TYPE
, _Fld1443_RTRef as PriceType_RTRef
, _Fld1443_RRRef as PriceType_KEY
, _Fld1444 as ContractCurrencyRate
, _Fld1445RRef as DocumentCurrency_KEY
, _Fld1446 as IncomingNumber
, _Fld1447 as IncomingDate
, _Fld1448RRef as AccountContractor_KEY
, _Fld1449 as PaymentDate
, _Fld1450_TYPE as BasisDocument_TYPE
, _Fld1450_RTRef as BasisDocument_RTRef
, _Fld1450_RRRef as BasisDocument_KEY
, _Fld1451 as WarehouseInTabularSection
, _Fld1452RRef as CashRegister_KEY

```

```

, _Fld1453RRef      as   GoodsInventory_KEY
, _Fld1454      as   OperationComment
, _Fld4207RRef      as   DiscountCard_KEY
, _Fld4208      as   ProvideRetailSaleByContractor
, _Fld4611      as   CheckIsPrinted
, _Fld4612RRef      as   PaymentType_KEY
, _Fld5214RRef      as   InformationSource_KEY
, _Fld3658RRef      as   CashType_KEY
, _Fld3834RRef      as   CashierPlace_KEY
, _Fld3835      as   PaymentSum
, _Fld3841      as   AutomaticallyCreated
, _Fld3846RRef      as   Manager_KEY
, _Fld4209      as   DisableCard
, _Fld4210      as   UseSpecialDiscount
, _Fld4211      as   ValueOfSpecialDiscount
, _Fld4318      as   VIPSale
, _Fld4320      as   RestSum
, _Fld4488      as   MonyInPledge
, _Fld4711RRef      as   ForcedVATRate_KEY
, _Fld4712      as   InstallForcedVATRate
, _Fld4720RRef      as   POSTerminal_KEY
, _Fld5453      as   FillAnnex2InDeclarationTVA12
, _Fld5645      as   WrongBSO
, _Fld5848      as   DisableSale
, _Fld1436      as   Comment
, _Fld5993RRef      as   WarehouseOrder_KEY
, _Fld10495      as   UseVouchers
, _Fld11483      as   NotDelivery
, _Fld11501      as   Efactura
, _Fld11928RRef      as   SaleType_KEY
//, [_Fld11936RRef]      as   SaleType_KEY
FROM "$ ( DBName ) ".dbo."_Document90" as Док_GoodsSale
where Док_GoodsSale ._ Date_Time >='$( vDateFrom )' and Док_
GoodsSale ._ Date_Time <'$( vDateTo )';
Store * from [ Док_GoodsSale ] into
[$(BASE_QVD_PATH)02_SourceData\01_Sales\Док_ПродажТовару_$(vPeriod).qvd]
( qvd );
Drop Table [ Док_GoodsSale ];
end if
if InRollBack or
isnull(FileSize('$(BASE_QVD_PATH)02_SourceData\01_Sales\Док_ПродажТоваруБауче
pi_$(vPeriod).qvd')) then
TRACE Док_ПродажТоваруБаучери _$( vPeriod );
Док_GoodsSaleVouchers :
load
*;
SQL select
_LineNo10497      as   НомерСтрокаGoodsSaleVouchers
, _Fld10498RRef      as   Voucher_KEY
, _Fld10499      as   Nominal
, _Fld10500RRef      as   VoucherOperationType_KEY
, _Document90_IDRRef      as   GoodsSale_KEY
, _KeyField      as   _KeyField
FROM "$ ( DBName ) ".dbo."_Document90_VT10496" as h
inner join
"$ ( DBName ) ".dbo."_Document90" d
on
h."_Document90_IDRRef" = d."_IDRRef "
where d._ Date_Time >='$( vDateFrom )' and d._ Date_Time <'$( vDateTo
)';

Store * from [ Док_GoodsSaleVouchers ] into
[$(BASE_QVD_PATH)02_SourceData\01_Sales\Док_ПродажТоваруБаучери_$(vPeriod).qv
d]
( qvd );
Drop Table [ Док_GoodsSaleVouchers ];

```

```

end if
if InRollBack or
isnull (FileSize ('$(BASE_QVD_PATH)02_SourceData\03_Receipt\Док_НадходженняТовару_$(vPeriod).qvd')) then
TRACE Док_НадходженняТовару_$( vPeriod );
[ Док_GoodsReceipt ]:
load
*,
Date ( Floor ( Addyears ( ДатаGoodsReceipt , -2000)) as Дата;
SQL select
_IDRRef as GoodsReceipt_KEY
,_ Version as ВерсіяДаних
,_ Marked as МіткаВидаленняGoodsReceipt
,_ Date_Time as ДатаGoodsReceipt
,_ NumberPrefix as _ NumberPrefix
,_ Number as Номер
,_ Posted as Проведено
,_ Fld1362RRef as Organization_KEY
,_ Fld1363RRef as Creator_KEY
,_ Fld1365RRef as OperationType_KEY
,_ Fld1366RRef as Warehouse_KEY
,_ Fld1367RRef as Contractor_KEY
,_ Fld1368RRef as Contract_KEY
,_ Fld1369 as Sum
,_ Fld1370_TYPE as PriceType_TYPE
,_ Fld1370_RTRef as PriceType_RTRef
,_ Fld1370_RRRef as PriceType_KEY
,_ Fld1371 as ContractCurrencyRate
,_ Fld1372RRef as DocumentCurrency_KEY
,_ Fld1373 as IncomingNumber
,_ Fld1374 as IncomingDate
,_ Fld1375RRef as AccountContractor_KEY
,_ Fld1376_TYPE as BasisDocument_TYPE
,_ Fld1376_RTRef as BasisDocument_RTRef
,_ Fld1376_RRRef as BasisDocument_KEY
,_ Fld1377RRef as Division_KEY
,_ Fld1378RRef as DistributeExpensesMethod_KEY
,_ Fld1379RRef as Account_KEY
,_ Fld1380_TYPE as ExtDimension1_TYPE
,_ Fld1380_RTRef as ExtDimension1_RTRef
,_ Fld1380_RRRef as ExtDimension1_KEY
,_ Fld1381_TYPE as ExtDimension2_TYPE
,_ Fld1381_RTRef as ExtDimension2_RTRef
,_ Fld1381_RRRef as ExtDimension2_KEY
,_ Fld1382_TYPE as ExtDimension3_TYPE
,_ Fld1382_RTRef as ExtDimension3_RTRef
,_ Fld1382_RRRef as ExtDimension3_KEY
,_ Fld1383 as PaymentDate
,_ Fld1384 as OperationComment
,_ Fld3496 as WarehouseInTabularSection
,_ Fld4203RRef as IncludedServicesDocument_KEY
,_ Fld4608RRef as DiscountCard_KEY
,_ Fld4312 as DisableCard
,_ Fld4313 as UseSpecialDiscount
,_ Fld4314 as ValueOfSpecialDiscount
,_ Fld4315RRef as CashRegister_KEY
,_ Fld4319 as VIPSale
,_ Fld4322 as CheckIsPrinted
,_ Fld4323 as PaymentSum
,_ Fld4324 as RestSum
,_ Fld5376RRef as PaymentType_KEY
,_ Fld5377RRef as POSTerminal_KEY
,_ Fld5452 as FillAnnex2InDeclarationTVA12
,_ Fld5641 as WrongBSO

```

```

, _Fld1364 as Comment
, _Fld5952RRef as GoodsAcceptance_KEY
, _Fld11482 as NotDelivery
, _Fld11927RRef as SaleType_KEY
//, _Fld11935RRef as SaleType_KEY
FROM "$( DBName )".dbo."_Document89" as Док_GoodsReceipt
where Док_GoodsReceipt ._ Date_Time >='$( vDateFrom )' and Док_
GoodsReceipt ._ Date_Time <'$( vDateTo )';
Store * from [ Док_GoodsReceipt ] into
[$(BASE_QVD_PATH)02_SourceData\03_Receipt\Док_НадходженняТовару_$(vPeriod).qvd]
( qvd );
Drop Table [ Док_GoodsReceipt ];
end if
LET vIncPeriod = AddMonths ( vIncPeriod , 1);
LOOP WHILE NOT InMonth ( vIncPeriod , vEndPeriod , 1);
let vIncPeriod = vStartPeriod ;

Do
let DateFrom = vIncPeriod ;
let DateTo = AddMonths ( vIncPeriod , 1);
let vPeriod = Year ( vIncPeriod ) & '_' & text ( num ( Month ( vIncPeriod
), '00'));
let InRollBack = vIncPeriod >= AddMonths ( vEndPeriod , - vRollBackMonths
);
let vDateFrom = Addyears ( DateFrom , 2000);
let vDateTo = Addyears ( DateTo , 2000);
if InRollBack or
isnull(FileSize('$(BASE_QVD_PATH)02_SourceData\02_Orders\Док_Замовлення_$(vPe
riod).qvd')) then
TRACE Док_Замовлення_$( vPeriod );
[ Док_CustomerDocumentOrder ]:
load
*,
Date ( Floor ( Addyears ( ДатаCustomerDocumentOrder , -2000)) ) as
Дата;
SQL select
_ IDRRef as CustomerDocumentOrder_KEY
, _ Version as ВерсіяДаних
, _ Marked as МіткаВидаленняCustomerDocumentOrder
, _ Date_Time as ДатаCustomerDocumentOrder
, _ Number as Номер
, _ Posted as Проведено
, _Fld1069RRef as Organization_KEY
, _Fld1070RRef as Creator_KEY
, _Fld1072RRef as Division_KEY
, _Fld1073RRef as Warehouse_KEY
, _Fld1074RRef as Contractor_KEY
, _Fld1075RRef as Contract_KEY
, _Fld1076 as Sum
, _Fld1077RRef as PriceType_KEY
, _Fld1078 as ContractCurrencyRate
, _Fld1079RRef as DocumentCurrency_KEY
, _Fld1080_TYPE as CashBankAccount_TYPE
, _Fld1080_RTRef as CashBankAccount_RTRef
, _Fld1080_RRRef as CashBankAccount_KEY
, _Fld1081RRef as ContractorBankAccount_KEY
, _Fld1082 as PaymentDate
, _Fld1083 as ShipmentDate
, _Fld4188RRef as DiscountCard_KEY
, _Fld3737 as Order
, _Fld3744 as CheckFilling
, _Fld3745 as CheckIsPrinted
, _Fld3828RRef as Manager_KEY
, _Fld4189 as DisableCard

```

```

, _Fld4190 as UseSpecialDiscount
, _Fld4191 as ValueOfSpecialDiscount
, _Fld4487 as CheckFillingPledge
, _Fld4677RRef as CashRegister_KEY
, _Fld5640 as CheckOtherOptions
, _Fld5751 as Project
, _Fld5752 as DeliverTerm
, _Fld1071 as Comment
, _Fld10028 as CheckFillingOrderProduction
, _Fld10489 as UseVouchers
, _Fld11919RRef as ContractForSale_KEY
, _Fld11925 as ServiceOrder
FROM "$( DBName )" .dbo. "_Document82" as
Док_CustomerDocumentOrder
where Док_CustomerDocumentOrder ._ Date_Time >='$( vDateFrom )' and
Док_CustomerDocumentOrder ._ Date_Time <'$( vDateTo )';
Store * from [ Док_CustomerDocumentOrder ] into
[$(BASE_QVD_PATH)02_SourceData\02_Orders\Док_Замовлення_$(vPeriod).qvd]
( qvd );
Drop Table [ Док_CustomerDocumentOrder ];
end if
LET vIncPeriod = AddMonths ( vIncPeriod , 1);
LOOP WHILE NOT InMonth ( vIncPeriod , vEndPeriod , 1);
Translation :
LOAD
A as [ Index ],
"B" as [ Reference ],
C as [UA],
D as [EN]
FROM
[lib://QlikData_chaban/ViTRA/04_DataBase/Translation/TranslationTable.xlsx]
( ooxml , no labels , table is Лист1);
Store Translation into [$(BASE_QVD_PATH)01_Dictionary/ Translation.qvd ] (
qvd );
Drop Table Translation ;
Exit Script ;

```

Лістинг №2 – Додаток для трансформації даних

```
SET ThousandSep = '';
SET DecimalSep = ',';
SET MoneyThousandSep = '';
SET MoneyDecimalSep = ',';
SET MoneyFormat = '# ##0,00 €;-# ##0,00 €';
SET TimeFormat = ' hh:mm:ss ';
SET DateFormat = 'DD.MM.YYYY';
SET TimestampFormat = 'DD.MM.YYYY hh:mm:ss [. fff ]';
SET FirstWeekDay = 0;
SET BrokenWeeks = 1;
SET ReferenceDay = 0;
SET FirstMonthOfYear = 1;
SET CollationLocale = ' uk -UA';
SET CreateSearchIndexOnReload = 1;
SET
MonthNames='січ.;лют.;бер.;квіт.;трав.;черв.;лип.;серп.;вер.;жовт.;лист.;груд
.';
SET
LongMonthNames='січень;лютий;березень;квітень;травень;червень;липень;серпень;
вересень;жовтень;листопад;грудень';
SET DayNames = ' пн;вт;ср;чт;пт;сб;нд ';
SET LongDayNames='понеділок;вівторок;середа;четвер;п'ятниця;субота;неділя';
SET NumericalAbbreviation='3:k;6:M;9:G;12:T;15:P;18:E;21:Z;24:Y;-3:m;-6:u;-
9:n ;-12:p;-15:f;-18:a;-21:z;-24:y';
//SET DB_PATH =
'lib://QlikViewDocs_Folder/chaban/ViTRA/04_DataBase/01_DataFiles/';
SET vQVD_D =
'lib://QlikViewDocs_Folder/chaban/ViTRA/04_DataBase/01_DataFiles/01_Dictionar
y';
Set vQVD_S =
'lib://QlikViewDocs_Folder/chaban/ViTRA/04_DataBase/01_DataFiles/02_SourceDat
a';
Set vQVD_T =
'lib://QlikViewDocs_Folder/chaban/ViTRA/04_DataBase/01_DataFiles/03_Transform
Data';
Set vQVD_P =
'lib://QlikViewDocs_Folder/chaban/ViTRA/04_DataBase/01_DataFiles/04_Plans';
Параметри:
LOAD
    Початок Періоду ,
    КінецьПеріоду ,
    Поточна Дата ,
    vRollBackMonths ,
    Валюта За Замовчуванням ,
    Мітка Валюті За Замовчуванням ,
    Валюта Базова
FROM [$( vQVD_D )/ Параметри.qvd ] ( qvd );
LET vStartPeriod = Peek (' НачалоПеріоду ', -1, 'Параметри');
LET vEndPeriod = Peek (' КінецьПеріоду ', -1, 'Параметри');
LET vCurrentDate = Peek (' ПоточнаДата ', -1, 'Параметри');
LET vDefaultCurrency = Peek (' ВалютаЗаЗамовчуванням ', -1,
'Параметри');
LET vDefaultCurrencyLabel = Peek (' МіткаВалютіЗаЗамовчуванням ', -1,
'Параметри');
LET vBaseCurrency = Peek (' ВалютаБазова ', -1,
'Параметри');
Спр_ Nomenclature :
LOAD
    Nomenclature_KEY ,
    БатькоNomenclature_KEY as БатькоНоменклатура_Key ,
    ЦеГрупаNomenclature ,
```

```

    КодNomenclature as [Код товару],
    Nomenclature as [Товар],
    NomenclatureType_KEY ,
    OriginCountry_KEY ,
    NomenclatureGroup_KEY ,
    Supplier_KEY
FROM [$( vQVD_D ) / Спр_Nomenclature.qvd ]      ( qvd );

// Exit Script ;
Left Join ( Спр_Nomenclature )

//e3efefr:
LOAD
    NomenclatureGroups_KEY      as NomenclatureGroup_KEY ,
    NomenclatureGroups          as "Номенклатурна група"
FROM [$( vQVD_D ) / Спр_NomenclatureGroups.qvd ]      ( qvd );
Left Join ( Спр_Nomenclature )
LOAD
    Nomenclature_KEY            as БатькоНоменклатура_Key ,
    [Товар]                     as "Група товару"
Resident Спр_Nomenclature ;

// exit Script ;
Товари_tmp :
Hierarchy
( Nomenclature_KEY, БатькоНоменклатура_Key, Товар,, Товар, Товари_Шлях, '/', Товари_
Рівень)
load
    Nomenclature_KEY ,
    БатькоНоменклатура_Key ,
Товар
resident Спр_Nomenclature ;

Store Товари_tmp into '$( vQVD_D )/Товари_tmp01.qvd' ( qvd );
Left Join ( Спр_Nomenclature )
Load *
Resident Товари_tmp ;
Drop Table Товари_tmp ;
Товари_tmp_01:
Load
    Nomenclature_KEY ,
    [Код товару],
    Товар,
    Товар1,
    Товар2      as "Ієрархія товару 1",
    if ( IsNull (Товар3), Товар2, Товар3) as "Ієрархія товару 2",
    Товар4,
    "Група товарів",
    "Номенклатурна група",
    NomenclatureType_KEY ,
    NomenclatureGroup_KEY ,
    OriginCountry_KEY ,
    Supplier_KEY
Resident Спр_Nomenclature

//; Exit Script ;
Where
    Upper (Товар1) = 'MARFURI-VITRA' and
    ЦеГрупаNomenclature = '01';

Drop Table Спр_Nomenclature ;

// Exit Script ;
Store Товари_tmp_01 into '$( vQVD_D )/Товари_tmp02.qvd' ( qvd );

```

```

Left Join (Товари_tmp_01)
LOAD
    NomenclatureTypes_KEY      as NomenclatureType_KEY ,
    NomenclatureTypes як      " Вигляд товару "

// NomenclatureType_KEY
FROM [$( vQVD_D ) / Спр_NomenclatureTypes.qvd ]      ( qvd );
Left Join (Товари_tmp_01)
LOAD
    Contractors_KEY            as Supplier_KEY ,
    Contractors                as Постачальник
FROM [$( vQVD_D ) / Спр_Contractors.qvd ]      ( qvd );
Left Join (Товари_tmp_01)
LOAD
    CountriesClassifier_KEY    as OriginCountry_KEY ,
    CountriesClassifier        as "Країна товару"
FROM [$( vQVD_D ) / Спр_CountriesClassifier.qvd ]      ( qvd );
PC_ObjectsCategory :
LOAD
    Object_KEY                as Nomenclature_KEY ,
    FirstValue ( Category_KEY ) as ObjectsCategory_KEY
FROM [$( vQVD_D ) / PC_ObjectsCategory.qvd ]      ( qvd )
Group by Object_KEY ;

Left Join ( PC_ObjectsCategory )
LOAD
    ObjectsCategory_KEY ,
    ObjectsCategory            as "Категорія товару"
FROM [$( vQVD_D ) / Спр_ObjectsCategory.qvd ]      ( qvd );

Left Join (Товари_tmp_01)
LOAD
    Nomenclature_KEY ,
    "Категорія товару"
Resident PC_ObjectsCategory ;

Drop Table PC_ObjectsCategory ;

Товари :
Load
    Nomenclature_KEY ,
    [Код товару],
    Товар,
    "Ієрархія товару 1",
    "Ієрархія товару 2",
    if ( IsNull (Товар4), "Ієрархія товару 2", Товар4) as "Ієрархія товару
3",
    "Номенклатурна група",
    "Група товарів",
    "Вигляд товару",
    "Категорія товару",
    "Країна товару",
    "Постачальник"
Resident Товари_tmp_01;
Drop Table Товари_tmp_01;

// Exit Script ;
Store Товари into '$ ( vQVD_D ) / Товари.qvd ' ( qvd );
Товари_tmp :
Load
    Nomenclature_KEY

```

```

Resident Товари;

Drop Tables Товари;
Спр_Contractors :
LOAD
    Contractors_KEY ,
    БатькоContractors_KEY ,
    Contractors      as Контрагент,
    Region_KEY ,
    ContractorCategory_KEY
    //, OriginCountry_KEY
FROM [$( vQVD_D ) / Спр_Contractors.qvd ]      ( qvd );
Контрагенти_tmp :
Hierarchy
(Contractors_KEY,БатькоContractors_KEY,Контрагент,,Контрагент,Контрагенти_Шля
х, '/',Контрагенти_Рівень)
load
    Contractors_KEY ,
    БатькоContractors_KEY ,
Контрагент
resident Спр_Contractors ;
Left Join ( Спр_Contractors )
Load
    *
Resident Контрагенти_tmp ;

Drop Table Контрагенти_tmp ;

Контрагенти:
NoConcatenate
Load
    Contractors_KEY  as Contractor_KEY ,
Контрагент,
Контрагент1      as "Група контрагента"//,

Resident Спр_Contractors ;

Store Контрагенти into '$( vQVD_D )/ Контрагенти.qvd '( qvd );

Drop Tables Спр_Contractors , Контрагенти;

// PH Продажі

let vIncPeriod = vStartPeriod ;

Do
    let DateFrom = vIncPeriod ;
    let DateTo = AddMonths ( vIncPeriod , 1);
    let vPeriod = Year ( vIncPeriod ) & '_' & text ( num ( Month ( vIncPeriod
), '00'));

    if not isnull ( FileSize ('$( vQVD_S )/01_Sales/ PH_Продажі _$( vPeriod ).
qvd ')) then

TRACE PH_Продажі _$( vPeriod );

    PH_Продажі_tmp :
LOAD
        Дата,
        РеєстраторDiscountsAnalisys_TRef ,
        РеєстраторDiscountsAnalisys_KEY as GoodsSale_KEY ,
        НомерСтрокаDiscountsAnalisys ,
        АктивністьDiscountsAnalisys ,
        Division_KEY ,

```

```

        Nomenclature_KEY ,
        Warehouse_KEY ,
        Manager_KEY ,
        Quantity ,
        Sum ,
        SumManagement ,
        Cost ,
        DiscountSumManagement ,
        OperationType_KEY
FROM [$( vQVD_S )/01_Sales/ PH_Продажі_$( vPeriod ). qvd ] ( qvd )
// Where Exists ( Nomenclature_KEY )
;

        else
        TRACE !!!!!!!! Файл відсутній: $( vQVD_S )/01_Sales/ PH_Продажі_$(
vPeriod );
        endif ;

        LET vIncPeriod = AddMonths ( vIncPeriod , 1);

LOOP WHILE NOT InMonth ( vIncPeriod , vEndPeriod , 1);

PH_Продажі :
NoConcatenate
Load
*
Resident PH_Продажі_tmp
Where АКТИВНІСТЬDiscountsAnalisys = '01'
Order By
        Дата,
        GoodsSale_KEY ,
        НомерСтрокаDiscountsAnalisys
;

Drop Table PH_Продажі_tmp ;

Store * from [ PH_Продажі ] into '$( vQVD_T )/01_Sales/PH_Продажі_tmp00.qvd'
( qvd );

Док_ПродажіТоваруВаучері :
NoConcatenate
load *
inline [ GoodsSale_KEY ];

Док_ПродажіТовару :
NoConcatenate
load *
inline [ GoodsSale_KEY ];

Док_НадходженняТовару :
NoConcatenate
load *
inline [ GoodsSale_KEY ];

// Док Продажі

let vIncPeriod = vStartPeriod ;

Do
let DateFrom = vIncPeriod ;
let DateTo = AddMonths ( vIncPeriod , 1);

```

```

    let vPeriod = Year ( vIncPeriod ) & '_' & text ( num ( Month ( vIncPeriod
), '00')));

    if not
isnull(FileSize('$ (vQVD_S)/01_Sales/Док_ПродажіТоваруВаучери_$(vPeriod).qvd')
) then

TRACE Док_ПродажіТоваруВаучери _$( vPeriod );

Concatenate ( Док_ПродажіТоваруВаучери )
LOAD
    Nominal ,
    GoodsSale_KEY
FROM [$( vQVD_S )/01_Sales/ Док_ПродажіТоваруВаучери _$( vPeriod ). qvd ]
( qvd );

else
TRACE !!!!!!!!!!! Немає файлу : $( vQVD_S )/01_Sales/
Док_ПродажіТоваруВаучери _$( vPeriod );
endif ;

    if not
isnull(FileSize('$ (vQVD_S)/01_Sales/Док_ПродажіТовару_$(vPeriod).qvd')) then

TRACE Док_ПродажіТовару _$( vPeriod );

Concatenate ( Док_ПродажіТовару )
LOAD
    GoodsSale_KEY ,
Номер,
    Contractor_KEY ,
    BasisDocument_TYPE ,
    BasisDocument_RTRef ,
    BasisDocument_KEY ,
    SaleType_KEY
FROM [$( vQVD_S )/01_Sales/ Док_ПродажіТовару _$( vPeriod ). qvd ]
( qvd );

else
TRACE !!!!!!!!!!! Файл відсутній: $( vQVD_S )/01_Sales/ Док_ПродажіТовару
_$( vPeriod );
endif ;

    if not
isnull(FileSize('$ (vQVD_S)/03_Receipt/Док_НадходженняТовару_$(vPeriod).qvd'))
then

TRACE Док_НадходженняТовару _$( vPeriod );

Concatenate ( Док_НадходженняТовару )
LOAD
    GoodsReceipt_KEY as GoodsSale_KEY ,
Номер,
    Contractor_KEY ,
    BasisDocument_TYPE ,
    BasisDocument_RTRef ,
    BasisDocument_KEY ,
    SaleType_KEY
FROM [$( vQVD_S )/03_Receipt/ Док_НадходженняТовару _$( vPeriod ). qvd ]
( qvd );

```

```

else
TRACE !!!!!!!!!!! Немає файлу : $( vQVD_S )/03_Receipt/
Док_НадходженняТовару_$( vPeriod );
endif ;

LET vIncPeriod = AddMonths ( vIncPeriod , 1);

LOOP WHILE NOT InMonth ( vIncPeriod , vEndPeriod , 1);

// Приєднання ваучерів у Продажі

Left Join ( PH_Продажі )
LOAD
Sum ( Nominal ) as [Номінал ваучера],
1 as НомерСтрокаDiscountsAnalisys ,
GoodsSale_KEY
Resident Док_ПродажіТоваруВаучери
Group By
GoodsSale_KEY ;

Drop Table Док_ПродажіТоваруВаучери ;

Store * from [ PH_Продажі ] into '$( vQVD_T )/01_Sales/PH_Продажі_tmp01.qvd'
( qvd );

TRACE Групування: PH_Продажі ;

PH_Продажі2:
Load
Дата,
РеєстраторDiscountsAnalisys_TRef ,
GoodsSale_KEY ,
Nomenclature_KEY ,
Warehouse_KEY ,
Manager_KEY ,
OperationType_KEY ,
Concat ( Номер_РядкаDiscountsAnalisys , ',' ) as "№ рядка",
Sum ( Quantity ) as Кількість,
Sum ( Sum_ ) as "Сума без ПДВ",
Sum ( SumManagement ) as "Сума",
Sum ( Cost ) as Собівартість,
Sum ( DiscountSumManagement ) as "Сума знижки",
Sum ( "Номінал ваучера" ) as "Номінал ваучера"
Resident PH_Продажі
Group By
Дата,
РеєстраторDiscountsAnalisys_TRef ,
GoodsSale_KEY ,
Nomenclature_KEY ,
Warehouse_KEY ,
Manager_KEY ,
OperationType_KEY
;

Drop Table PH_Продажі ;

Rename Table PH_Продажі2 to PH_Продажі ;

Store * from [ PH_Продажі ] into '$( vQVD_T )/01_Sales/PH_Продажі_tmp02.qvd'
( qvd );

```

```
Док_Продажі :
Load
    GoodsSale_KEY ,
Номер,
    Contractor_KEY ,
    SaleType_KEY ,
    BasisDocument_TYPE ,
    BasisDocument_RTRef ,
    BasisDocument_KEY ,
    Sum ([ GoodsSale_KEY ])          as Sm
Resident Док_ПродажіТовару
Group by
    GoodsSale_KEY ,
Номер,
    Contractor_KEY ,
    SaleType_KEY ,
    BasisDocument_TYPE ,
    BasisDocument_RTRef ,
    BasisDocument_KEY ;
```

```
Drop Table Док_ПродажіТовару ;
```

```
PH_Продажі_tmp :
Load
    GoodsSale_KEY ,
Номер,
    Contractor_KEY ,
    SaleType_KEY ,
    BasisDocument_TYPE ,
    BasisDocument_RTRef ,
    BasisDocument_KEY ,
' Док_ПродажіТовара ' as "Документ-Джерело"
Resident Док_Продажі ;
```

```
Drop Table Док_Продажі ;
```

```
Док_Надходження :
Load
    GoodsSale_KEY ,
Номер,
    Contractor_KEY ,
    SaleType_KEY ,
    BasisDocument_TYPE ,
    BasisDocument_RTRef ,
    BasisDocument_KEY ,
    Sum ([ GoodsSale_KEY ])          as Sm
Resident Док_НадходженняТовара
Group by
    GoodsSale_KEY ,
Номер,
    Contractor_KEY ,
    SaleType_KEY ,
    BasisDocument_TYPE ,
    BasisDocument_RTRef ,
    BasisDocument_KEY ;
```

```
Drop Table Док_НадходженняТовара ;
```

```
Concatenate ( PH_Продажі_tmp )
Load
```

```

        GoodsSale_KEY ,
Номер,
        Contractor_KEY ,
        SaleType_KEY ,
        BasisDocument_TYPE ,
        BasisDocument_RTRef ,
        BasisDocument_KEY
Resident Док_Надходження ;

Drop Table Док_Надходження ;

Store * from [ PH_Продажі_tmp ] into '$( vQVD_T
)/01_Sales/PH_Продажі_tmp02_2.qvd'      ( qvd );

Left Join ( PH_Продажі )
Load
    *
Resident PH_Продажі_tmp ;

Drop Table PH_Продажі_tmp ;

Store * from [ PH_Продажі ] into '$( vQVD_T )/01_Sales/PH_Продажі_tmp03.qvd'
    ( qvd );

Транзакції:
LOAD
Дата,
    if ( IsNull (Номер), '_БН_', Номер) as Номер,
"№ рядки",
    РеєстраторDiscountsAnalysis_TRef ,
    if ( РеєстраторDiscountsAnalysis_TRef ='0000005A', 'Продаж',
        if ( РеєстраторDiscountsAnalysis_TRef = '00000059', 'Повернення', '
Коригуюча ')) as "Тип Транзакції",
    GoodsSale_KEY ,
    Nomenclature_KEY ,
    Warehouse_KEY ,
    Manager_KEY ,
    OperationType_KEY ,
    Contractor_KEY ,
    SaleType_KEY ,
    BasisDocument_TYPE ,
    BasisDocument_RTRef ,
    BasisDocument_KEY as DocumentOrder_KEY ,
"Документ-джерело",
Кількість,
Сума,
"Сума без ПДВ",
Собівартість,
"Сума знижки",
" Номінал ваучера",
    RangeSum (Сума, -Собівартість) as Маржа
Resident PH_Продажі ;

Drop Table PH_Продажі ;

// Exit Script ;
Store * from [Транзакції] into '$( vQVD_T )/01_Sales/Транзакції_tmp00.qvd'
    ( qvd );

Left Join (Транзакції)
LOAD
    OperationsTypes_KEY as OperationType_KEY ,

```

```
КодOperationsTypes as "Код типу операції",
OperationsTypes    as "Тип операції "
FROM [$( vQVD_D ) / Спр_OperationsTypes.qvd ]      ( qvd );
```

Left Join (Транзакції)

```
LOAD
    SaleTypes_KEY          as SaleType_KEY ,
    КодSaleTypes          as "Код типу Продажі",
    SaleTypes              as "Тип Продажу"
FROM [$( vQVD_D ) / Спр_SaleTypes.qvd ]          ( qvd );
```

Drop Fields OperationType_KEY , SaleType_KEY від транзакції;

Транзакції2:

NoConcatenate

Load

*

Resident Транзакції

Order By

Дата,

"Тип транзакції",

Номер,

"№ рядка";

Drop Table Транзакції;

Rename Table Транзакції2 to Транзакції;

```
Store * from [Транзакції] into '$( vQVD_T )/01_Sales/Транзакції_tmp01.qvd'
( qvd );
```

// PH Заовлення

```
let vIncPeriod = vStartPeriod ;
```

Do

```
let DateFrom = vIncPeriod ;
```

```
let DateTo = AddMonths ( vIncPeriod , 1);
```

```
let vPeriod = Year ( vIncPeriod ) & '_' & text ( num ( Month ( vIncPeriod
), '00'));
```

if not

```
isnull(FileSize('$ (vQVD_S)/02_Orders/PH_Заовлення_$(vPeriod).qvd')) then
```

```
TRACE PH_Заовлення $( vPeriod );
```

```
PH_Заовлення_tmp :
```

LOAD

```
Дата as "Дата заовлення",
```

```
// ПеріодCustomerDocumentOrders ,
```

```
РеєстраторCustomerDocumentOrders_TRef ,
```

```
РеєстраторCustomerDocumentOrders_KEY ,
```

```
НомерСтрокаCustomerDocumentOrders as "№ рядки",
```

```
АктивністьCustomerDocumentOrders ,
```

```
Вид Руху ,
```

```
DocumentOrder_KEY ,
```

```
Nomenclature_KEY ,
```

```
Quantity as "Кількість заовлення",
```

```
SumManagement as "Сума заовлення"
```

```
FROM [$( vQVD_S )/02_Orders/ PH_Заовлення_$( vPeriod ). qvd ]      ( qvd )
Where Exists ( Nomenclature_KEY );
```

```

else
TRACE !!!!!!!!!!! Файл відсутній: $( vQVD_S )/02_Orders/ PH_Замовлення _$(
vPeriod );
endif ;

LET vIncPeriod = AddMonths ( vIncPeriod , 1);

LOOP WHILE NOT InMonth ( vIncPeriod , vEndPeriod , 1);

PH_Замовлення :
NoConcatenate
Load
*
Resident PH_Замовлення_tmp
Where
АктивністьCustomerDocumentOrders = '01';

Drop Table PH_Замовлення_tmp ;

Store * from [ PH_Замовлення ] into '$( vQVD_T
)/01_Sales/PH_Замовлення_tmp00.qvd' ( qvd );

Док_Замовлення :
NoConcatenate
load *
inline [ CustomerDocumentOrder_KEY ];

// Док_Замовлення
let vIncPeriod = vStartPeriod ;

Do
let DateFrom = vIncPeriod ;
let DateTo = AddMonths ( vIncPeriod , 1);
let vPeriod = Year ( vIncPeriod ) & '_' & text ( num ( Month ( vIncPeriod
), '00' ));

if not
isnull(FileSize('$ (vQVD_S)/02_Orders/Док_Замовлення_$(vPeriod).qvd')) then
TRACE Док_Замовлення _$( vPeriod );

Concatenate ( Док_Замовлення )
LOAD
CustomerDocumentOrder_KEY ,
Номер as "№ замовлення",
Contractor_KEY ,
Manager_KEY ,
ServiceOrder
FROM [$( vQVD_S )/02_Orders/ Док_Замовлення _$( vPeriod ). qvd ]
( qvd );

else
TRACE !!!!!!!!!!! Файл відсутній: $( vQVD_S )/02_Orders/ Док_Замовлення
_$( vPeriod );
endif ;

LET vIncPeriod = AddMonths ( vIncPeriod , 1);

```

```
LOOP WHILE NOT InMonth ( vIncPeriod , vEndPeriod , 1);
```

```
TRACE Угрупування: РН_Замовлення ;
```

```
РН_Замовлення2:
```

```
NoConcatenate
```

```
Load
```

```
    "Дата замовлення",  
    РеєстраторCustomerDocumentOrders_TRef ,  
    РеєстраторCustomerDocumentOrders_KEY ,  
    Concat ("№ рядка", ',') as "№ рядка",  
    Вид Руху ,  
    DocumentOrder_KEY ,  
    Nomenclature_KEY ,  
    Sum ("Кількість замовлення") as "Кількість замовлення",  
    Sum ("Сума замовлення") as "Сума замовлення"
```

```
Resident РН_Замовлення
```

```
Group By
```

```
    "Дата замовлення",  
    РеєстраторCustomerDocumentOrders_TRef ,  
    РеєстраторCustomerDocumentOrders_KEY ,  
    Вид Руху ,  
    DocumentOrder_KEY ,  
    Nomenclature_KEY ,
```

```
;
```

```
Drop Table РН_Замовлення ;
```

```
Rename Table РН_Замовлення2 to РН_Замовлення ;
```

```
Store * from [ РН_Замовлення ] into '$( vQVD_T  
) /01_Sales/РН_Замовлення_tmp01.qvd' ( qvd );
```

```
TRACE Угрупування: Док_Замовлення ;
```

```
Док_Замовлення2:
```

```
Load
```

```
    CustomerDocumentOrder_KEY ,  
    "№ замовлення",  
    Contractor_KEY ,  
    Manager_KEY ,  
    ServiceOrder ,  
    Count ( CustomerDocumentOrder_KEY )
```

```
Resident Док_Замовлення
```

```
Group by
```

```
    CustomerDocumentOrder_KEY ,  
    "№ замовлення",  
    Contractor_KEY ,  
    Manager_KEY ,  
    ServiceOrder ;
```

```
Drop Table Док_Замовлення ;
```

```
Left Join ( РН_Замовлення )
```

```
Load
```

```
    CustomerDocumentOrder_KEY as DocumentOrder_KEY ,  
    "№ замовлення",  
    Contractor_KEY ,  
    Manager_KEY ,
```

```

ServiceOrder
Resident Док_Замовлення2;

Drop Table Док_Замовлення2;

Rename Field "№ замовлення" до "Номер замовлення";

Замовлення:
NoConcatenate
Load
    *
Resident РН_Замовлення
Where
    ВидРуху = '0' // Тільки за документами Замовлення
покупця , без Витратних накладних
    and ServiceOrder = '00' // Не сервісні Замовлення
Order By
"Дата замовлення",
"Номер замовлення",
"№ рядки"
;

Drop Table РН_Замовлення ;

Store * from [Замовлення] into '$( vQVD_T )/01_Sales/РН_Замовлення_tmp02.qvd'
    ( qvd );
ПродЗкази :
Load
    DocumentOrder_KEY ,
    Nomenclature_KEY ,
    "Код типу операції",
    "Тип операції ",
    1 as "Ознака наявність продажу",
    Sum ( DocumentOrder_KEY ) as Sm
Resident Транзакції
Where
    Not IsNull ( DocumentOrder_KEY ) and
    Not IsNull ( Nomenclature_KEY ) and
    "Тип Транзакції"=' Продаж '
Group by
    DocumentOrder_KEY ,
    Nomenclature_KEY ,
    "Код типу операції",
    "Тип операції ";

Store * from [ПродЗкази] into '$( vQVD_T
) /01_Sales/РН_Замовлення_tmp03.qvd' ( qvd );

Left Join (Замовлення)
Load
    DocumentOrder_KEY ,
    Nomenclature_KEY ,
    "Код типу операції",
    "Тип операції ",
    "Ознака наявність продажу"
Resident ПродЗкази ;

Drop Table ПродЗкази ;

Замовлення Agr :
Load
    DocumentOrder_KEY as Замовлення_KEY ,
    Nomenclature_KEY ,

```

```
"Код типу операції",
"Тип операції ",
"Дата замовлення",
"Номер замовлення",
    Contractor_KEY ,
    Manager_KEY ,
    Concat ("№ рядка", ',') as "№ рядка замовлення",
    Sum ("Кількість замовлення") as "Кількість замовлення",
    Sum ("Сума замовлення") as "Сума замовлення",
"Ознака наявності продажу"
Resident Замовлення
Group By
    DocumentOrder_KEY ,
    Nomenclature_KEY ,
"Код типу операції",
"Тип операції ",
"Дата замовлення",
"Номер замовлення",
    Contractor_KEY ,
    Manager_KEY ,
"Ознака наявності продажу";
```

```
Drop Table Замовлення;
```

```
Store * from [ ЗамовленняАгр ] into '$( vQVD_T
)/01_Sales/PH_Замовлення_tmp04.qvd' ( qvd );
```

```
Left Join ( Замовлення Агр )
LOAD
    Nomenclature_KEY ,
" Категорія товару "
FROM [$( ( vQVD_D ) / Товари.qvd ] ( qvd );
```

```
Left Join ( Замовлення Агр )
LOAD
    Managers_KEY as Manager_KEY ,
    Managers as "Менеджер"
FROM [$( ( vQVD_D ) / Спр_Managers.qvd ] ( qvd );
```

```
Store * from [ ЗамовленняАгр ] into '$( vQVD_T )/01_Sales/ Замовлення.qvd '
( qvd );
```

```
// Прокид у транзакції продажів посилань на Замовлення, які є в PH_Замовлення
( що б очистити від " Повернення товару із застави" )
```

```
ЗамовленняАгр_tmp :
Load
    Замовлення_KEY ,
    Nomenclature_KEY ,
    Count ( Замовлення_KEY )
Resident ЗамовленняАгр
Group by
    Замовлення_KEY ,
    Nomenclature_KEY ;
```

```
Drop Table Замовлення Агр ;
```

```
Store * from [ ЗамовленняАгр_tmp ] into '$( vQVD_T
```

```

)/01_Sales/PH_Замовлення_tmp05.qvd'                ( qvd );

Left Join (Транзакції)
Load
    Замовлення_KEY ,
    Замовлення_KEY as DocumentOrder_KEY ,
    Nomenclature_KEY
Resident Замовлення Agr_tmp ;

Drop Table Замовлення Agr_tmp ;

// Drop Table Транзакції;

// Exit Script ;
Left Join (Транзакції)
LOAD
    Nomenclature_KEY ,
    " Категорія товару "
FROM [$( vQVD_D ) / Товари.qvd ]                ( qvd );

Left Join (Транзакції)
LOAD
    Managers_KEY                as Manager_KEY ,
    Managers                    as "Менеджер"
FROM [$( vQVD_D ) / Спр_Managers.qvd ]        ( qvd );

Store * from [Транзакції] into '$( vQVD_T )/01_Sales/ Транзакції.qvd '
    ( qvd );
trace Завантаження курсів валют;

Спр_Валюти :
LOAD
    Currencies_KEY            as Валюта_KEY ,
    Currencies                as Валюта
FROM [$( vQVD_D ) / Спр_Currencies.qvd ]        ( qvd )
where not ( Currencies = 'RUB' ); //( Currencies = 'USD') or ( Currencies =
'EUR');

// курси валют
Курси TMP :
LOAD
    ПеріодExchangeRates as Дата,
    Currency_KEY            as Валюта_KEY ,
    Num (" Rate ") / Factor    as [Курс]
FROM [$( vQVD_D ) / PC_ExchangeRates.qvd ]        ( qvd );

Store * від [ Курси TMP ] into [$( vQVD_T ) \ 02_Currency \
КурсиВалют_tmp01.qvd ]        ( qvd );

// доповнюємо назвами
Left Join ( Курси TMP )
LOAD
    Валюта_KEY ,
    Валюта
Resident Спр_Валюти ;
Drop Table Спр_Валюти ;

```

```

// робимо курси на кожен день з транзакцій
КурсВалют_tmp :
Load
    [Валюта],
Дата as [ ДатаОт ],
    if ([Валюта]= Previous (Валюта) , Date ( Previous ([Дата])-1), makedate
(3000)) as ДатаДо ,
[Курс]
Resident Курси TMP
Order By [Валюта], [Дата] desc ;
Drop Table Курси TMP ;

// генеруємо лічильник списку дат від vStartPeriod до vEndPeriod
Курс Валют :
NoConcatenate
load
    date ('$( vStartPeriod )' + RowNo ()-1) as [Дата]
    AutoGenerate 1
WHILE NOT InMonth ('$( vStartPeriod )' + RowNo ()-1, Date ( floor ( YearEnd
('$( vEndPeriod )'))), 1);

// дублюємо лічильник на кожну з завантажених валют
left join ( КурсВалют )
load
    Distinct [Валюта]
Resident КурсВалют_tmp ;

// До кожної дати ( згенерованій у лічильнику ) прокидаємо відповідну ДатуОт
і ДатуДо та відповідний курс валют

// Результат приєднується ( доджуїнується )
inner join ( КурсВалют )
IntervalMatch ([Дата], [Валюта])
load
[ ДатаОт ],
[ ДатаДо ],
[Валюта]
Resident КурсВалют_tmp ;
left join ( КурсВалют )
load
[ ДатаОт ],
[ ДатаДо ],
[Валюта],
[Курс]
Resident КурсВалют_tmp ;
Drop Fields [ ДатаОт ], [ ДатаДо ];
Drop Table КурсВалют_tmp ;
Store * From [ КурсВалют ] into [$( vQVD_T )\02_Currency \ Курси Валют.qvd ]
( qvd );
Drop Table Курс Валют ;
MapMonth :
mapping
LOAD *
Inline [
    _ Month ,          MonthNum
January ,          1
February ,         2
March ,            3
April ,            4
May ,              5
June ,             6
July ,             7
August ,           8
September ,       9

```

```

Жовтень , 10
November , 11
December , 12
];
MapMonth2:
NoConcatenate

// mapping
LOAD *
Inline [
  _ Month ,      MonthNum
  January , 1
  February , 2
  March , 3
  April , 4
  May , 5
  June , 6
  July , 7
  August , 8
  September , 9
  Жовтень , 10
  November , 11
  December , 12
];
NomenclatureCategoriesPlanSales :
LOAD
  Category _      Пік Плану ,      January ,      February ,      March ,
  April ,      May ,      June ,      July ,      August ,      September ,
  Жовтень ,      November ,      December
FROM [$( vQVD_P )/ PH_NomenclatureCategoriesPlanSales.qvd ]
qvd );
CategoryPlan_tmp :
Crosstable ( _ Month , CPlan , 2)
LOAD
  *
Resident NomenclatureCategoriesPlanSales ;
Drop Table NomenclatureCategoriesPlanSales ;
Left Join ( CategoryPlan_tmp )
LOAD
  _ Month ,
  MonthNum
Resident MapMonth2;
Плани :
Load
  Category _
  Місяць Плану ,
  Пік Плану ,
  '01.'&num( MonthNum ) &'.'& ПікПлану      as Дата,
  0      as "Тип плану ",
План
;
LOAD
  Category _
  _ Month ,
  MonthNum ,
  _ Month / * ApplyMap ( ' MapMonth ', _ Month , '-' ) * / as Місяць Плану ,
  Пік Плану ,
  Num ( CPlan )      as План
Resident CategoryPlan_tmp ;
Drop Table CategoryPlan_tmp ;
Left Join ( Плани )
LOAD
  ObjectsCategory_KEY ,
  ObjectsCategory      as Category

```

```

FROM [$( vQVD_D ) / Спр_ObjectsCategory.qvd ] ( qvd );
ПланиПовні :
Load
Дата,
    "Тип плану ",
План,
    Місяць Плану ,
    Рік Плану ,
    if ( IsNull ( " Category " ), '_ Категорія товару не вказано_', " Category
") as " Категорія товару "
Resident Плани ;
rop Table Плани ;
ManagersPlanSales :
LOAD
    Manager ,    Рік Плану ,    January ,    February ,    March ,    April ,
    May ,    June ,    July ,    August ,    September ,    Жовтень ,
November ,    December
FROM [$( vQVD_P ) / PH_ManagersPlanSales.qvd ] ( qvd );
ManagersPlan_tmp :
Crosstable ( _ Month , MPlan , 2)
LOAD
*
Resident ManagersPlanSales ;

// Drop Table ManagersPlanSales ;
Left Join ( ManagersPlan_tmp )
LOAD
    _ Month ,
    MonthNum
Resident MapMonth2;
ManagersPlan :
Load
    Manager ,
    Місяць Плану ,
    Рік Плану ,
    '01.'&num( MonthNum ) &'.'& РікПлану as Дата,
    1 as "Тип плану ",
План
;
LOAD
    Manager ,
    _ Month ,
    MonthNum ,
    _ Month / * ApplyMap ( ' MapMonth ', _ Month , '-' ) * / as Місяць Плану ,
    Рік Плану ,
    Num ( MPlan ) as План
Resident ManagersPlan_tmp ;
Drop Table ManagersPlan_tmp ;
Left Join ( ManagersPlan )
LOAD
    Managers_KEY as Manager_KEY ,
// Managers as "Менеджер"
    Managers as Manager
FROM [$( vQVD_D ) / Спр_Managers.qvd ] ( qvd );
CorrMan :
LOAD
В as "Менеджер",
"A" as Manager
FROM
[lib://QlikData_chaban/ViTRA/04_DataBase/Translation/TranslationTable.xlsx]
( ooxml , no labels , table is Лист2);
Left join ( ManagersPlan )
load *
Resident CorrMan ;

```

```

Drop Table CorrMan ;
Concatenate ( ПланиПовні )
Load
Дата,
    "Тип плану ",
План,
    Manager_KEY ,
    Місяць Плану ,
    Рік Плану ,
    if ( IsNull ("Менеджер"), '_Менеджер не вказано_', "Менеджер") as
"Менеджер"
Resident ManagersPlan ;
Drop Table ManagersPlan ;
Store * from [ ПланиПовні ] into '$( vQVD_T )/03_Plans/ Плани.qvd ' (
qvd );

// Drop Table Транзакції;
Drop Table ManagersPlanSales ;

// Drop Table ПланиПовні ;
Drop Table ПланиПовні ;

// Drop Table Транзакції;
Translations :
NoConcatenate
load " Index ",
    Reference _
EN,
/UA
FROM [$( vQVD_D ) / Translation.qvd ] ( qvd );
Language :
CROSSTABLE ( Language , Translation ) LOAD
    Index ,
EN,
UA
RESIDENT Translations ;
_ Languages :
LOAD Concat ( distinct Language , ',' ) as _ Languages
RESIDENT Language ;
LET vLanguages = chr (39) & replace ( Peek ('_ Languages '), ',', chr (39)
&','& chr (39)) & chr (39);
DROP TABLE _ Languages ;
LanguageList :
Load distinct _ Language ,
    AutoNumber ( _ Language ) AS Index _
;
LOAD Distinct
    Language as _ Language
RESIDENT Language
WHERE Language > 0;
DROP TABLE Translations ; // drops after multilang calendar створення

// Змінні для локалізації інтерфейсу
LET T = replace ('_T_$_1_@ (= FirstSortedValue ( distinct _ Language , Index
_))', '@', '$');
LET vTranslationsRows = NoOfRows (' Language ');
FOR i = 0 TO vTranslationsRows - 1
    LET _ vTranslationId = Peek (' Index ', i, ' Language ');
    LET _ vTranslation = Peek (' Translation ', i, ' Language ');
    LET _ vTrLanguage = Peek (' Language ', i, ' Language ');
    LET _ vTrName = '_T_$_(_ vTranslationId )$_(_ vTrLanguage)';
    LET $_ vTrName = '$(_ vTranslation)';
    LET _ vTranslationId =;
    LET _ vTranslation =;

```

```

    LET _ vTrLanguage =;
    LET _ vTrName =;
NEXT;

// SET NumConvertRu = " Pick ( RangeSum ( Floor (LOG10($1)),1), NUM($1,
'###0'), NUM($1, '###0'), NUM($1, ') ###0'), NUM($1/1000,
'###0,00$(_T_1_Ru)', ',', ' '), NUM($1/1000, '###0,00$(_T_1_Ru )', ',', ' '),
NUM($1/1000, '# ##0,00 $(_T_1_Ru)', ',', ' '), NUM($1/1000000, '# ##0, 00
$(_T_2_Ru)', ',', ' '), NUM($1/1000000, '###0,00 $(_T_2_Ru)', ',', ' '),
NUM($1/1000000, '# ##0,00 $(_T_2_Ru)', ',', ' '), NUM($1/1000000000, '#
##0,00 $(_T_3_Ru)', ',', ' '), NUM($1/ 1000000000, '# ##0,00 $(_T_3_Ru)',
',, ' '), NUM($1/1000000000, '# ##0,00 $(_T_3_Ru)', ',', ' '),
NUM($1/1000000000000, '###0,00$(_T_4_Ru)', ',', ' '), NUM($1/1000000000000,
'###0,00$(_T_4_Ru)', ',', ' '), NUM($1/1000000000000, '# ##0,00 $(_T_4_Ru)',
',, ' '))";
// SET NumConvertEn = " Pick ( RangeSum ( Floor (LOG10($1)),1), NUM($1,
'###0'), NUM($1, '###0'), NUM($1, ') # ##0'), NUM($1/1000, '# ##0,00
$(_T_1_En)', ',', ' '), NUM($1/1000, '# ##0,00 $(_T_1_En)', ',', ' '),
NUM($1/1000, '# ##0,00 $(_T_1_En)', ',', ' '), NUM($1/1000000, '# ##0, 00
$(_T_2_En)', ',', ' '), NUM($1/1000000, '# ##0,00 $(_T_2_En)', ',', ' '),
NUM($1/1000000, '# ##0,00 $(_T_2_En)', ',', ' '), NUM($1/1000000000, '#
##0,00 $(_T_3_En)', ',', ' '), NUM($1/ 1000000000, '# ##0,00 $(_T_3_En)',
',, ' '), NUM($1/1000000000, '# ##0,00 $(_T_3_En)', ',', ' '),
NUM($1/1000000000000, '###0,00$(_T_4_En)', ',', ' '), NUM($1/1000000000000,
'###0,00$(_T_4_En)', ',', ' '), NUM($1/1000000000000, '# ##0,00 $(_T_4_En)',
',, ' '))";
SET NumConvertUA = " if ($1 > pow (10, 15), '> 1 квлн', Pick ( Floor
(LOG10($1))+1, NUM($1, '###0'), NUM($1, ') ###0'), NUM($1, '###0'),
NUM($1/1000, '###0,00 тис', ',', ' '), NUM($1/1000, '# ##0,00 тис', ',',
'), NUM($1/1000, '###0,00 тис', ',', ' '), NUM($1/1000000, '# # #0,00 млн',
',, ' '), NUM($1/1000000, '###0,00 млн', ',', ' '), NUM($1/1000000, '###0 ,00
млн', ',', ' '), NUM($1/1000000000, '###0,00 млрд', ',', ' '),
NUM($1/1000000000, '###0,00 млрд', ',', ' '), NUM($1/1000000000, '###0,00
млрд', ',', ' '), NUM($1/1000000000000, '###0,00 трлн', ',', ' '),
NUM($1/1000000000000, '###0,00 трлн', ',', ' '), NUM($1/1000000000000,
'###000,00 трлн', ',', ' '), NUM($1/1000000000000000, '###0,00 квлн', ',',
'), NUM($1/1000000000000000, '###0,00 квлн', ',', ' '),
NUM($1/1000000000000000, '# ##0,00 квлн', ',', ' '))));";
SET NumConvertEN = " if ($1 > pow (10, 15), '> 1 qdln', Pick ( Floor
(LOG10($1))+1, NUM($1, '###0'), NUM($1, ') ###0'), NUM($1, '###0'),
NUM($1/1000, '###0,00 k', ',', ' '), NUM($1/1000, '# ##0,00 k', ',', ' '),
NUM($1/1000, '# ##0,00 k', ',', ' '), NUM($1/1000000, '# # #0,00 mln', ',',
'), NUM($1/1000000, '###0,00 mln', ',', ' '), NUM($1/1000000, '###0 ,00 mln
', ',', ' '), NUM($1/1000000000, '# ##0,00 bln', ',', ' '),
NUM($1/1000000000, '# ##0,00 bln', ',', ' '), NUM($1/1000000000, '# ##0,00
bln', ',', ' '), NUM($1/1000000000000, '# ##0,00 trln', ',', ' '),
NUM($1/1000000000000, '###0,00 trln', ',', ' '), NUM($1/1000000000000,
'###000,00 trln', ',', ' '), NUM($1/1000000000000000, '# ##0,00 qdln',
',, ' '), NUM($1/1000000000000000, '# ##0,00 qdl', ',', ' '),
NUM($1/1000000000000000, '# ##0,00 qdln', ',', ' '))));";
STORE [ Language ] INTO [ $ (
vQVD_D ) Language.qvd ] ( qvd );
STORE [ LanguageList ] INTO [ $ ( vQVD_D
) LanguageList.qvd ] ( qvd );
Drop Tables Language , LanguageList , Параметри;
Exit Script ;

```

Лістинг №3 – Додаток з аналітично-інформаційною системою

```
SET ThousandSep = '';
SET DecimalSep = ',';
SET MoneyThousandSep = '';
SET MoneyDecimalSep = ',';
SET MoneyFormat = '###0,00€;-###0,00€';
SET TimeFormat = ' h:mm:ss ';
SET DateFormat = 'DD.MM.YYYY';
SET TimestampFormat = 'DD.MM.YYYY h:mm:ss[.fff]';
SET DayNames = ' Пн;Вт;Ср;Чт;Пт;Сб;Нд ';
SET LongMonthNames = 'Січень;Лютий;Березень;Квітень;Майл;Червень;Июль;
SET LongDayNames='понеділок;вівторок;середа;четвер;п'ятниця;субота;неділя';
SET FirstWeekDay = 0;
SET BrokenWeeks = 0;
SET ReferenceDay = 0;
SET FirstMonthOfYear = 1;
SET CollationLocale = ' uk -UA';
SET HidePrefix = '_';
SET ShortDateFormat = 'DD.MM.YY';
LET vLoadTime = now (1);

// Exit Script ;
SET vVersion = '2.0';
Language :
LOAD
" Index " as id ,
    Language as _ Language ,
    Translation as _ Translation
FROM [$( vQVD_D ) Language.qvd ]
( qvd );
Translations :
NoConcatenate
LOAD
    id as _ TranslationId , *
RESIDENT Language ;

DROP TABLE Language ;
// Змінні для локалізації інтерфейсу
LET T = replace ('_T_$1_@(= MaxString ({$} _ Language ))', '@', '$');
_Languages :
LOAD Concat ( distinct _ Language , ',' ) as _ Languages
RESIDENT Translations ;
LET vLanguages = chr (39) & replace ( Peek ('_ Languages '), ',', chr (39)
&','& chr (39)) & chr (39);
DROP TABLE _ Languages ;

// Формуємо змінні на підставі таблиці Translations для реалізації
мультимовного інтерфейсу
LET vTranslationsRows = NoOfRows (' Translations ');
FOR i = 0 TO vTranslationsRows - 1
    LET _ vTranslationId = Peek ('_ TranslationId ', i, ' Translations ');
    LET _ vTranslation = Peek ('_ Translation ', i, ' Translations ');
    LET _ vTrLanguage = Peek ('_ Language ', i, ' Translations ');

    LET _ vTrName = '_T_$(_ vTranslationId )$_ vTrLanguage ';
    LET $_ vTrName ) = '$(_ vTranslation)';

    // TRACE $_ vTrName );

    LET _ vTranslationId =;
    LET _ vTranslation =;
    LET _ vTrLanguage =;
```

```

        LET _ vTrName =;
NEXT
Товари:
LOAD
    Nomenclature_KEY ,
    "Код товару" as " Product ID",
    "Код товару" as "Код товару",
    Товар,
    Товар as Product ,
    "Код товару" & ' | ' & Товар as ID_Product ,
    "Код товару" & ' | ' & Товар as Код_Товару ,
    if ( not IsNull ("Ієрархія товару 1"), "Ієрархія товару 1", '_ Product
hierarchy 1 not specified _')      as " Product hierarchy 1",
    if ( not IsNull ("Ієрархія товару 1"), "Ієрархія товару 1", '_ Ієрархія
товару 1 не вказана_') as "Ієрархія товару 1",

    if ( not IsNull ("Ієрархія товару 2"), "Ієрархія товару 2", '_ Product
hierarchy 2 not specified _')      as " Product hierarchy 2",
    if ( not IsNull ("Ієрархія товару 2"), "Ієрархія товару 2", '_ Ієрархія
товару 2 не вказана_') as "Ієрархія товару 2",

    if ( not IsNull ("Ієрархія товару 3"), "Ієрархія товару 3", '_ Product
hierarchy 3 not specified _')      as " Product hierarchy 3",
    if ( not IsNull ("Ієрархія товару 3"), "Ієрархія товару 3", '_ Ієрархія
товару 3 не вказана_') as "Ієрархія товару 3",

    if ( IsNull ("Номенклатурна група"), '_ Номенклатурна група не вказана_',
"Номенклатурна група") as "Номенклатурна група",
    if ( IsNull ( " Номенклатурна група " ), '_ Nomenclature group not
specified _', " Номенклатурна група") as " Nomenclature group ",

    if ( not IsNull (" Група товару"), "Група товару", '_ Група товару не
вказана_') as "Група товару",
    if ( not IsNull (" Група товару"), "Група товару", '_ Product group не
вказана_') as " Product group ",

    if ( not IsNull ("Вигляд товару"), "Вид товару", '_ Вигляд товару не
вказано_') as "Вид товару",
    if ( not IsNull ("Вигляд товару"), "Вигляд товару", '_ Product type not
specified _')      as " Product kind ",

    if ( not IsNull (" Країна товару"), " Країна товару", '_ Product country
not specified _') as " Product country ",
    if ( not IsNull (" Країна товару"), " Країна товару", '_ Країна товару не
вказана_') as "Країна товару",
    if ( IsNull ("Постачальник"), '_ Supplier not listed _', " Постачальник
") as " Supplier ",
    if ( IsNull ("Постачальник"), '_ Постачальник не вказаний_', "
Постачальник ") as "Постачальник"
FROM [$( vQVD_D )/ Товари.qvd ]          ( qvd )
where not IsNull ("Товар");

// Drop Table Замовлення_tmp ;
Транзакції_tmp :
LEFT KEEP ( Товари )
LOAD

// Manager_KEY ,
    Nomenclature_KEY ,
    Contractor_KEY ,

// DocumentOrder_KEY ,
    Замовлення_KEY ,
Дата,

```

```

Номер,
"Тип транзакції",
"№ рядки",

// Division_KEY ,

// Warehouse_KEY ,

// Document_RTRef ,

// BasisDocument_TYPE ,

// BasisDocument_RTRef ,

// "Документ-джерело",
Кількість,
Сума,
"Сума без ПДВ",
Собівартість,
"Сума знижки",
"Номинал ваучера",
Маржа,

// "Код типу операції",
"Тип операції", / / "Код типу продажу",
"Тип продаж",

// "Дата замовлення",

// "№ замовлення",

// " Кількість замовлення",

// " Сума замовлення",
"Категорія товару",
Менеджер
FROM [$( vQVD_T )/01_Sales/Транзакції.qvd ] (          qvd ) ;
Транзакції:
NoConcatenate

// Keep ( Товари )
LOAD
    Nomenclature_KEY ,
    if ( IsNull ( Contractor_KEY ), '00000000000000000000000000000000',
Contractor_KEY ) as Contractor_KEY ,
Дата,
//Дата as Data ,
Номер as Документ,
Номер as Document ,
"Тип транзакції",
    if ("Тип Транзакції"=' Повернення ','Повернення', if ("Тип Транзакції"='
Продаж ','Продаж')) as "Тип транзакції",
    if ("Тип Транзакції"=' Повернення ',' Return ', if ("Тип Транзакції"='
Продаж ',' Sale ')) as " Transaction type ",
"№ рядка" as "№ line ",
"№ рядка" as "№ рядка",
Кількість,
Кількість as Quantity ,
Сума,
Сума as Amount ,
"Сума без ПДВ" as " Amount без VAT",
"Сума без ПДВ" as "Сума без ПДВ",
    Собівартість as Собівартість,
Собівартість as " Cost price ",

```

```

" Сума знижки" as " Discount amount ",
" Сума знижки" as "Сума знижки",
"Номинал ваучера",
" Номинал ваучера" as " Voucher value ",
Маржа,
Маржа as Margin ,

// "Код типу операції",
    if ( IsNull ("Тип операції"), '_Тип операції не вказано_', if ("Тип операції " = ' Продаж товарів ', 'Продаж товарів', if ("Тип операції" = ' Надання послуг', 'Надання послуг'))    as "Тип операції",
    if ( IsNull ( " Тип операції " ), '_ Operation type not specified _', if ("Тип операції " = ' Продаж товарів ', ' Sale of goods ', if ("Тип операції" = ' Надання послуг', ' Provision of services '))) as " Operation type ",
    if ( IsNull ("Тип продаж"), ' Return of goods ', if ("Тип продажу" = 'Опт', ' Wholesale ', ' Online sales '))    as " Sale type ",
    if ( IsNull ("Тип продаж"), 'Повернення товару', if ("Тип продажу" = ' Опт', 'Опт', 'Онлайн продажу'))    as "Тип продаж",
// if ( IsNull ("Тип продажу"), '_Тип продажу не вказано_', "Тип продажу")
    as "Тип продажу",

// "Дата замовлення",

// "№ замовлення",

// " Кількість замовлення",

// " Сума замовлення",
    Замовлення_KEY ,
    if ( IsNull ("Категорія товару"), '_ Product category not specified _', "
Категорія товару ") as " Product category ",
    if ( IsNull ("Категорія товару"), '_Категорія товару не вказана_', "
Категорія товару ") as "Категорія товару",
    if ( IsNull ("Менеджери"), '_ Менеджера не вказано_', "Менеджер") as
"Менеджери",
    if ( IsNull ("Менеджери"), '_ Manager not listed _', "Менеджер") as "
Managers "
Resident Транзакції_tmp
; // Where not IsNull (Дата);
Drop Table Транзакції_tmp ;

// Exit Script ;

Замовлення_tmp :
LEFT KEEP ( Товари )
LOAD
    Manager_KEY ,
    Nomenclature_KEY ,

// "Код типу операції",
"Тип операції ",
    Замовлення_KEY ,
"Дата замовлення",
"Номер замовлення",
    Contractor_KEY ,
"№ рядки замовлення",
" Кількість замовлення",
" Сума замовлення",
"Ознака наявності продажу",
" Категорія товару ",
Менеджер
FROM [$( vQVD_T )/01_Sales/ Замовлення.qvd ]          ( qvd );

Concatenate (Транзакції)

```

```

Load
    Manager_KEY ,
    Nomenclature_KEY ,

// "Код типу операції",
    if ("Тип операції " = ' Продаж товарів ', ' Sale of goods ', if ("Тип операції " = ' Надання послуг', ' Provision of services ')) as " Operation type ",
    if ("Тип операції " = ' Продаж товарів ', 'Продаж товарів', if ("Тип операції " = ' Надання послуг', 'Надання послуг')) as "Тип операції",
    Замовлення_KEY ,
    "Дата замовлення" as " Order date ",
    "Дата замовлення" as "Дата замовлення",
    "Дата замовлення" as Дата,
    "Номер замовлення" as " Order number ",
    "Номер замовлення" as "Номер замовлення",
    if ( IsNull ( Contractor_KEY ), '00000000000000000000000000000000', Contractor_KEY ) as Contractor_KEY ,
    "№ рядка замовлення" as "№ рядка замовлення",
    "№ строки замовлення" as " Order Line No ",
    " Кількість замовлення" as "Кількість замовлень",
    " Кількість замовлення" as " Order Quantity ",
    " Сума замовлення" as "Сума замовлення",
    " Сума замовлення" as " Order amount ",
    "Ознака наявності продажу" as "A sign of a sale ",
    "Ознака наявності продажу" as "Ознака наявності продаж",
    if ( IsNull ( " Категорія товару " ), '_ Product category not specified _', " Категорія товару ") as " Product category ",
    if ( IsNull ( " Категорія товару " ), '_ Категорія товару не вказана_', " Категорія товару ") as "Категорія товару",
    if ( IsNull ("Менеджер"), '_ Менеджера не вказано_', "Менеджер") as "Менеджери",
    if ( IsNull ("Менеджер"), '_ Manager not listed _', "Менеджер") as " Managers ",
    '_ The type of sale is not specified _' as " Sale type ",
    '_Тип продаж не вказано_' as "Тип продаж"
Resident Замовлення_tmp ;
DIRECTORY $( vQVD_D );

// Exit Script ;
Клієнти_tmp :
LEFT KEER (Транзакції)
LOAD
    Contractor_KEY ,
Контрагент as Клієнт ,
Група контрагента as Група _ клієнта "/*,
" Категорія контрагента" as " Категорія клієнта ",
    Region */
FROM [ Контрагенти.qvd ] ( qvd )
Where Exists ( Contractor_KEY );

Concatenate ( Клієнти_tmp )
LOAD * INLINE [
Contractor_KEY
00000000000000000000000000000000
];
Клієнти :
NoConcatenate
Load
    Contractor_KEY ,
    if ( not IsNull ( Клієнт ), Клієнт , '_ Клієнт не вказано _') as Клієнт ,
    if ( not IsNull ( Клієнт ), Клієнт , '_ Client not specified _') as Client ,

```

```

        if ( not IsNull ( Клиент ), Клиент , '_Клієнта не вказано_' )      as
Клиент,
        if ( not IsNull (" Група клієнта " ), " Група клієнта " , '_ Група клієнта
не вказана_' )      as " Група клієнта " ,
        if ( not IsNull (" Група клієнта " ), " Група клієнта " , '_ Client group
not specified_' )      as " Client group " ,
        if ( not IsNull (" Група клієнта " ), " Група клієнта " , '_Група клієнта
не вказана_' )      as "Група клієнта",
'_ Категорія клієнта не вказано_' as " Категорія клієнта " ,
'_ Client category not specified_' as " Client category " ,
'_ Категорія клієнта не вказана_' as "Категорія клієнта",
'_ Region клієнта не вказано_' as Region ,
'_ Client region not specified_' as , _
'_ Region клієнта не вказано_' as Region
Resident Клиенти_tmp ;
Drop Table Замовлення_tmp ;
Drop Table Клиенти_tmp ;

// Exit Script ;
ТипТранзакції :
LOAD * INLINE [
Тип Транзакції, _ ТипТранзакціїМіткаId
Продаж , Sales
Повернення , Returns
];
for each _ Lang in FieldValueList ('_ Language ')

    LEFT JOIN ( ТипТранзакції )
    LOAD
        id as _ ТипТранзакціїМіткаId ,
        _ Translation as "_ ТипТранзакціїМітка $_ Lang )"
RESIDENT Translations
WHERE _ Language = '$_ Lang )';
    next
LET _ Lang = null ();
Concatenate (Транзакції)

// Плани :
LOAD
    Дата,
"Тип плану " as [Тип плану ],
План,
' Продаж товарів ' as "Тип операції ",
    if ( IsNull ( " Категорія товару " ), '_ Категорія товару не вказано_', "
Категорія товару ") as " Категорія товару " ,
    if ( IsNull ( " Категорія товару " ), '_ Product category not specified_',
" Категорія товару ") as " Product category " ,
    if ( IsNull ( " Категорія товару " ), '_ Категорія товару не вказана_', "
Категорія товару ") as "Категорія товару",
    if ( IsNull ("Менеджер"), '_ Менеджер не вказано_', "Менеджер") as
"Менеджера",
    if ( IsNull ("Менеджер"), '_ Менеджера не вказано_', "Менеджер") as
"Менеджери",
    if ( IsNull ("Менеджер"), '_ Manager not listed_', "Менеджер") as "
Managers "
FROM [$( vQVD_T )/03_Plans/ Плани.qvd ] ( qvd );
LET vMinDate = vStartPeriod ; // Date ( Peek ( ' MinDate ', -1, ' MinMaxDates
') );

// Для прогнозу виконання планів - календар генеруємо до кінця року
LET vMaxDate = Date ( floor ( YearEnd ( vEndPeriod ))); // Date ( floor (
YearEnd ( Peek ( ' MaxDate ', -1, ' MinMaxDates ')))));
SET CALENDAR_TABLE = ' Календар ';

```

```

$(CALENDAR_TABLE):
LOAD
Дата,
    num (Дата) as DateNum ,
    Day (Дата) as "День",
    Year (Дата) as "Рік",
    Month (Дата) as " Місяць ",
    WeekDay (Дата) as "День тижня ",
    Year (Дата)& num ( Month (Дата),'00') as " РікМісяць ",
    Dual ( Year (Дата)&'-'& Month (Дата), Year (Дата) * 100 + Month (Дата)
as "Рік- Місяць ",
    Dual ( Month (Дата) & '-' & Year (Дата), Month (Дата) * 10000 + Year
(Дата)) as " Місяць -Рік",
    Dual ( WeekName (Дата,0), WeekYear (Дата) * 100 + Week (Дата)) as "Рік-
Тиждень ",
    WeekYear (Дата) & Num ( Week (Дата), '00') as РікТиждень ,
    // Week (Дата) & '-' & WeekYear (Дата) as " Тиждень -Рік",
    Week (Дата) as " Тиждень ",
    'К' & Ceil ( Month (Дата)/3) AS "Квартал",
    Dual ( Year (Дата)&'-'&'К' & Ceil ( Month (Дата)/3), Year (Дата) & num (
Ceil ( Month (Дата)/3), '00')) as "Рік -Квартал",
    Year (Дата) & num ( Ceil ( Month (Дата)/3), '00') as РікКвартал ,
    Dual ('К' & Ceil ( Month (Дата)/3) & '-' & Year (Дата), Num ( Ceil (
Month (Дата)/3), '00') & Year (Дата)) as "Квартал -Рік",
    'П' & Ceil ( Month (Дата)/6) AS " Півріччя ",
    Dual ( Year (Дата)&'-'&'П' & Ceil ( Month (Дата)/6), Year (Дата) & num (
Ceil ( Month (Дата)/6), '00')) as "Рік - півріччя ",
    Dual ('П' & Ceil ( Month (Дата)/6) & '-' & Year (Дата), Num ( Ceil (
Month (Дата)/6), '00') & Year (Дата)) as ' Півріччя -Рік',
    if ( num ( WeekDay (Дата)) = 5 або num ( WeekDay (Дата)) = 6, 1, 0) as
IsWeekend

, if ( InYear (Дата,'$( vCurrentDate )',0), 1, 0) AS CY // Прапор поточного
року
, if ( InYear (Дата, '$( vCurrentDate )',-1), 1, 0) AS PY // Прапор
попереднього року

, if ( InMonth (Дата,'$( vCurrentDate )',0), 1, 0) AS CM // Прапор поточного
місяця
, if ( InMonth (Дата, '$( vCurrentDate )',-1), 1, 0) AS PM // Прапор
попереднього місяця
, if ( InMonth (Дата,'$( vCurrentDate )',-12), 1, 0) AS PYM // прапор
аналогічного місяця в попередньому році
, if ( InQuarter (Дата,'$( vCurrentDate )',0), 1, 0) AS CQ // Прапор
поточного кварталу
, if ( InQuarter (Дата, '$( vCurrentDate )',-1), 1, 0) AS PQ // Прапор
попереднього кварталу
, if ( InQuarter (Дата,'$( vCurrentDate )',-4), 1, 0) AS PYQ // прапор
аналогічного кварталу попереднього року
, if ( InWeek (Дата,'$( vCurrentDate )',0), 1, 0) AS CW // прапор поточної
тижня
, if ( InWeek (Дата, '$( vCurrentDate )',-1), 1, 0) AS PW // прапор
попередньої тижня
, if ( InYear (Дата,'$( vCurrentDate )',-1) and week (Дата) = week ('$(
vCurrentDate )'), 1, 0) AS PYW // прапор аналогічний тижня у попередньому
році
, if ( InYearToDate (Дата,'$( vCurrentDate )',0), 1, 0) AS CYD // прапор
ToDate в поточному році
, if ( InYearToDate (Дата,'$( vCurrentDate )',-1), 1, 0) AS PYD // прапор
ToDate у попередньому році
, if ( InMonthToDate (Дата,'$( vCurrentDate )',0), 1, 0) AS CMD // прапор
ToDate в поточному місяць
, if ( InMonthToDate (Дата,'$( vCurrentDate )',-1), 1, 0) AS PMD // прапор
ToDate у попередньому місяць

```

```

, if ( InMonthToDate (Дата,'$( vCurrentDate )',-12), 1, 0) AS PYMD // прапор
ToDate аналогічного місяця в попередньому році

, if ( InQuarterToDate (Дата,'$( vCurrentDate )',0), 1, 0) AS CQD // прапор
ToDate у поточному кварталі
, if ( InQuarterToDate (Дата,'$( vCurrentDate )',-1), 1, 0) AS PQD // прапор
ToDate у попередньому кварталі
, if ( InQuarterToDate (Дата,'$( vCurrentDate )',-4), 1, 0) AS PYQD // прапор
ToDate аналогічного кварталу попереднього року _

, if ( InWeekToDate (Дата,'$( vCurrentDate )',0), 1, 0) AS CWD // прапор
ToDate в поточній тижні
, if ( InWeekToDate (Дата,'$( vCurrentDate )',-1), 1, 0) AS PWD // прапор
ToDate у попередній тижні
, if ( InYear (Дата,'$( vCurrentDate )',-1) and week (Дата)= week ('$(
vCurrentDate )') and weekday (Дата)<= weekday ('$( vCurrentDate )') , 1, 0)
AS PYWD // прапор ToDate аналогічною тижня у попередньому році

,1 as DS
,1 as DE
, if (Дата = date ( floor ( MonthStart (Дата))), 1, 0) as MS
, if (Дата = date ( floor ( MonthEnd (Дата))), 1, 0) as ME
, if (Дата = date ( floor ( QuarterStart (Дата))), 1, 0) as QS
, if (Дата = date ( floor ( QuarterEnd (Дата))), 1, 0) as QE
, if (Дата = date ( floor ( YearStart (Дата))), 1, 0) as YS
, if (Дата = date ( floor ( YearEnd (Дата))), 1, 0) as YE
, if (Дата <= '$( vCurrentDate )', 1, 0) as FP // Період " транзакцій " (
крім майбутніх періодів для планів _

, If ( DayNumberOfYear (Дата) <= DayNumberOfYear ('$( vCurrentDate )'), 1, 0)
AS [ InYTD ]
, Year ('$( vCurrentDate )')- Year (Дата) AS [ YearsAgo ]
//, If ( DayNumberOfQuarter ($1) <= DayNumberOfQuarter ( Today ()),1,0) AS [
InQTD ]
,4*Year('$(vCurrentDate)')+Ceil(Month('$(vCurrentDate)')/3)-4*Year(Дата)-
Ceil(Month(Дата)/3) AS [ QuartersAgo ]
//, Ceil ( Month ( Today ())/3)- Ceil ( Month ($1)/3) AS [ QuarterRelNo ]
//, If ( Day ($1)<= Day ( Today ()),1,0) AS [ InMTD ]
,12*Year('$(vCurrentDate)')+Month('$(vCurrentDate)')-12*Year(Дата)-
Month(Дата) AS [ MonthsAgo ]
//, Month ( Today ())- Month ($1) AS [ MonthRelNo ]
//, If ( WeekDay ($1)<= WeekDay ( Today ()),1,0) AS [ InWTD ]
,( WeekStart ('$( vCurrentDate )')- WeekStart (Дата))/7 AS [ WeeksAgo ]
//, Week ('$( vCurrentDate )')- Week (Дата) AS [ WeekRelNo ]

// , Year (Дата) as " Year "
;
LOAD Date ('$( vMinDate )' + RecNo () - 1) as Дата
AutoGenerate vMaxDate - vMinDate + 1;

_LFLДата :
load
[Дата] as [_ LFLДата ] //для вибору дат LFL
Resident $(CALENDAR_TABLE)
where FP=1;

[ autoCalendar ]:
DECLARE FIELD DEFINITION Tagged ('$ date ')
FIELDS
Dual ( Year ($1), YearStart ($1)) AS [ Year ] Tagged ('$ axis ', '$ year
'),

```

```

Dual ('Q' & Num (Ceil (Num (Month ($1)) / 3)), Num (Ceil (NUM (Month ($1)) / 3), 00)) AS [
Quarter ] Tagged ('$ quarter ', '$ cyclic '),
Dual ( Year ($1) & '-' & Num ( Ceil ( Num ( Month ($1)) / 3)), QuarterStart
($1)) AS [ YearQuarter ] Tagged ('$ yearquarter ', '$ qualified '),
Dual ('Q' & Num ( Ceil ( Num ( Month ($1)) / 3)), QuarterStart ($1)) AS [_
YearQuarter ] Tagged ('$ yearquarter ', '$ hidden ', '$ simplified '),
Num ( Month ($1), '00') AS [ Month ] Tagged ('$ month ', '$ cyclic '),
Dual ( Year ($1) & '-' & Num ( Month ($1), '00'), monthstart ($1)) AS [
YearMonth ] Tagged ('$ axis ', '$ yearmonth ', '$ qualified '),
Dual ( Num ( Month ($1), '00'), monthstart ($1)) AS [_ YearMonth ] Tagged
('$ axis ', '$ yearmonth ', '$ simplified ', '$ hidden '),
Dual ('W' & Num ( Week ($1), 00), Num ( Week ($1), 00)) AS [ Week ] Tagged ('$
weeknumber ', '$ cyclic '),
Date ( Floor ($1)) AS [ Date ] Tagged ('$ axis ', '$ date ', '$ qualified
'),
Date ( Floor ($1), 'D') AS [_ Date ] Tagged ('$ axis ', '$ date ', '$
hidden ', '$ simplified ');

//DERIVE FIELDS FROM FIELDS Дата USING [ autoCalendar ];
DERIVE FIELDS FROM EXPLICIT TAGS '$ date ' USING [ autoCalendar ];

// Алгоритм локалізації календаря
DateFormat :
MAPPING LOAD DISTINCT
_ Language , _ Translation
RESIDENT Translations
WHERE _ TranslationId = ' CalendarDateFormat ';

FOR EACH vLang in $( vLanguages )

    // TRACEDate $ ( vLang );
LET DF = ApplyMap ( ' DateFormat ', '$( vLang )');
TRACE DF: $ (DF);

    // Перелік полів календаря, що підлягають локалізації . TranslateKey -
ключ для перекладу . _ _
    // Якщо Generate = 1, поля генеруються на підставі локалізованих
значень.
    // Наприклад , для ПоляКалендаря " Місяць " створюємо за значенням у
полі ФормулаГенераціїПоля нове поле на відповідному мовою , з ключем у полі
КлючДляПерекладу
    // У полі GenerationFormula $1 - це поле з локалізованими значеннями
// Dual ( WeekName (Дата, 0), WeekYear (Дата) * 100 + Week (Дата))

// Якщо Generate = 2, то поле генерується на підставі простий формули !
_ ДляЛокалізаціяКалендаря :
LOAD * INLINE [
ПоляКалендаря ; TranslateKey ; Generate ; GenerationFormula
Місяць ; Month ; 0;
День тижня ; WeekDay ; 0;
квартир; Quarter ; 0;
Півріччя ; Halfyear ; 0;
Рік - Тиждень ; YearWeek ; 0;
Рік; Year; 0;
Тиждень; Week; 0;
Рік - Тиждень ; YearWeek ; 1; Дата, Dual ( WeekName (Дата, 0), WeekYear
(Дата) * 100 + Week (Дата))
Місяць ; YearMonth ; 1; Дата, Dual ( Year (Дата) & '-' & "$1", Year
(Дата) * 100 + Month (Дата))
Місяць ; MonthYear ; 1; Dual ("$1" & '-' & Year (Дата), Month (Дата) *
10000 + Year (Дата))
квартир; YearQuarter ; 1; Дата, Dual (Рік & '-' & "$1", Рік & num ( Ceil

```

```

( Month (Дата)/3), '00'))
    квартал; QuarterYear ; 1; Dual ("§1" & '-' & Пік, Num ( Ceil ( Month
(Дата)/3), '00') & Пік)
    Півріччя ; YearHalfyear ; 1; Дата, Dual (Пік & '-' & "§1", Пік & num (
Ceil ( Month (Дата)/6), '00'))
    Півріччя ; HalfyearYear ; 1; Dual ("§1" & '-' & Пік, Num ( Ceil ( Month
(Дата)/6), '00') & Пік)
Дата; Date; 2; Дата, Date (Дата, '$(DF)')
] ( delimiter is ';');

// без полів для генерації
_ Temp :
LOAD distinct chr (39) & Concat ( distinct Trim ( ПолеКалендаря ), chr
(39) & ',' & chr (39)) & chr (39) as CalendarLocalizationFieldsStrings
RESIDENT _ ДляЛокалізаціяКалендаря
WHERE Generate = 0; // без полів для генерації

LET vCalendarLocalizationFieldsStrings = Peek ('
CalendarLocalizationFieldsStrings ');

// TRACE vCalendarLocalizationFieldsStrings : $(
vCalendarLocalizationFieldsStrings );

DROP TABLE _ Temp ;

LEFT JOIN ( _ ДляЛокалізаціяКалендаря )
LOAD DISTINCT
    _ TranslationId as [ TranslateKey ]
    , _ Translation as [ CalendarField ]
RESIDENT Translations
WHERE _ Language = '$( vLang )';

CalendarFieldMap $( vLang ):
MAPPING LOAD ПолеКалендаря , CalendarField
RESIDENT _ ДляЛокалізаціяКалендаря
WHERE Generate = 0; // без полів для генерації

CalendarFieldGenerationMap $( vLang ):
MAPPING LOAD ПолеКалендаря , Concat ( GenerationFormula & ' as [' &
CalendarField & ']', ', ', ') as CalendarGenerationFields
RESIDENT _ ДляЛокалізаціяКалендаря
WHERE Generate = 1 // Поля для генерації
GROUP BY ПолеКалендаря ;

//DROP TABLE _ ДляЛокалізаціяКалендаря ;

FOR EACH vCalendarField IN $( vCalendarLocalizationFieldsStrings )

    TranslatedField :
    LOAD ApplyMap ( ' CalendarFieldMap $( vLang )', '$( vCalendarField
)') as TranslatedField
    AutoGenerate 1;

    LET vTranslatedField = Peek ( ' TranslatedField ');

    DROP TABLE TranslatedField ;

TRACE $( vTranslatedField ) $( vCalendarField );

IF vTranslatedField <> vCalendarField THEN
    //
    // Враховуємо Dual поля ( текстова складову локалізуємо )
    //
    [$( vCalendarField ) Локалізація ]:

```

```

LOAD DISTINCT [$( vCalendarField )], text ([$(
vCalendarField )]) as [_ TrID_$( vCalendarField )],
num ([$( vCalendarField )]) as [$( vCalendarField ) Numeric ]
RESIDENT $(CALENDAR_TABLE);

LEFT JOIN ([$( vCalendarField ) Локалізація ])
LOAD DISTINCT
    _ TranslationId as [_ TrID_$( vCalendarField )]
    , _ Translation as [$( vCalendarField )
Translation ]

RESIDENT Translations
WHERE _ Language = '$( vLang )';

LEFT JOIN ($(CALENDAR_TABLE))
LOAD [$( vCalendarField )]
    , if ( IsNum ([$( vCalendarField ) Numeric ]),
Dual (
    if ( not IsNull ([$( vCalendarField ) Translation ]), [$(
vCalendarField ) Translation ], [$( vCalendarField )]), [$( vCalendarField )
Numeric ]),
    if ( not IsNull ([$( vCalendarField ) Translation ]), [$(
vCalendarField ) Translation ], [$( vCalendarField )])) as [$(
vTranslatedField )]
    //,[$( vCalendarField ) Translation ] as [$(
vTranslatedField )]
RESIDENT [$( vCalendarField ) Локалізація ];

DROP TABLE [$( vCalendarField ) Локалізація ];

//
// На підставі локалізованих полів , генеруємо додаткові
поля ( наприклад , YearMonth і т.п. )
//
CalendarFieldGeneration :
LOAD ApplyMap (' CalendarFieldGenerationMap $( vLang
)', '$( vCalendarField )', null ()) as CalendarGenerationFields AutoGenerate
1;
LET vCalendarGenerationFields = Peek ('
CalendarGenerationFields ');
DROP TABLE CalendarFieldGeneration ;

IF not IsNull ( vCalendarGenerationFields ) THEN

TRACE $( vCalendarGenerationFields ) : $(
vCalendarGenerationFields ($( vTranslatedField )));

LEFT JOIN ($(CALENDAR_TABLE))
LOAD $( vCalendarGenerationFields ($(
vTranslatedField ))) RESIDENT $(CALENDAR_TABLE);

ENDIF

ENDIF

NEXT

// Проста генерація полів з використанням формули 2 (!)
CalendarG2FieldMap$( vLang ):
MAPPING LOAD ПолеКалендаря , CalendarField
RESIDENT _ ДляЛокалізаціяКалендаря
WHERE Generate = 2;

```

```

CalendarG2FieldGenerationMap$( vLang ):
MAPPING LOAD ПолеКалендаря , Concat ( GenerationFormula & ' as [' &
CalendarField & ']', ', ' ) as CalendarGenerationFields
RESIDENT _ ДляЛокалізаціяКалендаря
WHERE Generate = 2
GROUP BY ПолеКалендаря ;

//Для локалізації найменувань полів
" CalFieldsTo $( vLang )":
MAPPING LOAD DISTINCT
[ id ], [ _ Translation ]
RESIDENT Translations
WHERE [ _ Language ] = '$( vLang )'
and match ( [ id ], ' fldDate ', ' fldMonth ', ' fldWeekDay ', '
fldQuarter ',
' fldHalfyear ', ' fldYearWeek ', ' fldYear ', ' fldWear ', ' fldYearWeek ',
' fldYearMonth ', ' fldMonthYear ',
' fldYearQuarter ', ' fldQuarterYear ', ' fldYearHalfyear ', '
fldHalfyearYear ', ' fldYear_Month ') > 0;

_ Temp :
LOAD distinct chr (39) & Concat ( distinct Trim ( ПолеКалендаря ), chr
(39) & ',' & chr (39)) & chr (39) as CalendarLocalizationFieldsStrings
RESIDENT _ ДляЛокалізаціяКалендаря
WHERE Generate = 2; // генерація із застосуванням простий формули

LET vCalendarLocalizationFieldsStrings = Peek ( '
CalendarLocalizationFieldsStrings ');

DROP TABLE _ Temp ;

DROP TABLE _ ДляЛокалізаціяКалендаря ;

FOR EACH vCalendarField IN $( vCalendarLocalizationFieldsStrings )

TranslatedField :
LOAD ApplyMap ( 'CalendarG2FieldMap$( vLang )', '$( vCalendarField
)' ) as TranslatedField
AutoGenerate 1;

LET vTranslatedField = Peek ( ' TranslatedField ');

DROP TABLE TranslatedField ;

IF vTranslatedField <> vCalendarField THEN
//
// На підставі локалізованих полів , генеруємо додаткові
поля
//
CalendarFieldGeneration :
LOAD ApplyMap ( 'CalendarG2FieldGenerationMap$( vLang
)', '$( vCalendarField )', null () ) as CalendarGenerationFields AutoGenerate
1;

LET vCalendarGenerationFields = Peek ( '
CalendarGenerationFields ');
DROP TABLE CalendarFieldGeneration ;

IF not IsNull ( vCalendarGenerationFields ) THEN

TRACE $( vCalendarGenerationFields ) : $(
vCalendarGenerationFields $( vTranslatedField ));

LEFT JOIN ( $( CALENDAR_TABLE))

```

```

                                LOAD $( vCalendarGenerationFields $(
vTranslatedField ))) RESIDENT $(CALENDAR_TABLE);

                                ENDIF

ENDIF

NEXT;

NEXT

// Exit Script ;
Курси Валют :
LOAD
Дата,
'$( vDefaultCurrency )' as Валюта,
1 as Курс
RESIDENT $(CALENDAR_TABLE);

Курси Валют :
LOAD
Валюта,
Дата,
Курс
FROM [$( vQVD_T )/02_Currency/ Курси Валют.qvd ]           ( qvd );

// Формуємо курс остаточно року (на підставі останнього значення )
ExchTmp :
LOAD Валюта, Max (Дата) as MaxDate RESIDENT КурсиВалют
WHERE Валюта <> '$( vDefaultCurrency )'
GROUP BY Валюта;

LEFT JOIN ( ExchTmp )
LOAD Валюта, Дата as MaxDate , Курс RESIDENT Курси Валют ;

LEFT JOIN ( ExchTmp )
LOAD Дата
RESIDENT $(CALENDAR_TABLE);

Concatenate ( Курси Валют )
LOAD Валюта, Курс, Дата
RESIDENT ExchTmp
WHERE Дата > MaxDate ;

DROP TABLE ExchTmp ;

// Перекидаємо курси до таблиці транзакцій
// У таблиці транзакцій для кожної валюти буде створено поле, що має назва
Курс<Валюта>,

// де замість <Валюта> буде підставлено код відповідної валюти .

// Для валюти за замовчуванням ( тобто валюти транзакцій ) поле не
створюється .

SET ExchangeFormulaPK = 1;
ExchangeRates :
LOAD * INLINE [

```

```
CurrencyPK , _Валюта, ЗнакВалюти , ExchangeFormula
$( ExchangeFormulaPK ), $( vDefaultCurrency ), $( vDefaultCurrencyLabel ),
];
```

Валюти :

```
LOAD Concat ( Distinct chr (39) & Валюта & chr (39), ',') as Валюти
RESIDENT КурсиВалют
WHERE Валюта <> '$( vDefaultCurrency )';
```

```
LET vExchs = peek ( ' Валюти ', -1, ' Валюти ');
DROP TABLE Валюти ;
```

```
FOR EACH vCurrency IN $( vExchs )
```

```
    LET ExchangeFormulaPK = ExchangeFormulaPK + 1;
```

```
    LEFT JOIN ( Транзакції)
    LOAD Дата, Курс as 'Курс$( vCurrency )'
    RESIDENT КурсиВалют
    WHERE Валюта = '$( vCurrency )';
```

```
    Concatenate ( ExchangeRates )
    LOAD
$( ExchangeFormulaPK ) as CurrencyPK ,
    '$( vCurrency )' as _Валюта, // код валюти
    '$( vCurrency )' as Знак Валюти // грн , $, €
    ' / Курс$( vCurrency )' as ExchangeFormula
    AutoGenerate 1;
```

```
NEXT
```

```
LET ExchangeFormulaPK = null ();
```

```
DROP TABLE Курси Валют ;
SET EXCH = '(= FirstSortedValue ({$} ExchangeFormula , CurrencyPK ))';
SET vMoneySign = ' FirstSortedValue ( ЗнакВалюти , CurrencyPK )';
SET vIsCurrency = ' if ( _Валюта = '$1', true (), false ())';
```

```
_ ABCType :
LOAD AutoNumber ( _ ABCTypeId , '_ ABCTypeId ') as _ ABCTypePK , * INLINE [
    _ ABCTypeId , _ ABCExp , _ ABCExpTotal
Sales , vSales , vSalesTotal
Margin , vMargin , vMarginTotal
SalesCount , vSalesCount , vSalesCountTotal
];
```

```
LEFT JOIN ( _ ABCType )
LOAD _ Language , id as _ ABCTypeId , _ Translation as _ МіткаABC RESIDENT
Translations ;
```

```
_ XYZType :
LOAD AutoNumber ( _ XYZTypeId , '_ XYZTypeId ') as _ XYZTypePK , * INLINE [
    _ XYZTypeId , _ XYZField
Day , Дата
Week , Рік- Тиждень
Month , Рік Місяць
];
```

```
LEFT JOIN ( _ XYZType )
LOAD _ Language , id as _ XYZTypeId , _ Translation as _ МіткаXYZ RESIDENT
Translations ;
```

```
//
```

```

// Використовується для аналізу активності ( клієнтів , SKU і т.д. )

//
SET vNewColor = RGB(204, 102, 119);
SET vReturnColor = RGB (221, 204, 119);
SET vActiveColor = RGB(17, 119, 51);
SET vPassiveColor = RGB(68, 119, 170);

LEFT JOIN (Транзакції)
LOAD
    Дата,
    Рік Місяць as РікМісяцьID
RESIDENT $(CALENDAR_TABLE)
WHERE FP = 1;

Додатковий календар :
LOAD Distinct
    Рік Місяць as Рік МісяцьID ,
// Year (Дата) as Год _
    Dual ( Year (Дата) & '-' & Month (Дата), Рік Місяць ) as Рік_місяць _
    0 as CALF // Рік МісяцьID відповідає Рік_Місяць _
RESIDENT $(CALENDAR_TABLE)
WHERE FP = 1;

// Попередній місяць
LET vEndMonth = Year ( AddMonths ('$( vEndPeriod )', -1)) & num ( Month (
AddMonths ('$( vEndPeriod )', -1)), '00');

Concatenate ( Додатковий календар )
LOAD Distinct
    Рік Місяць as Рік МісяцьID ,
    Dual ( Year ( AddMonths (Дата, 1)) & '-' & Month ( AddMonths (Дата, 1)),
Year ( AddMonths (Дата, 1)) & num ( Month ( AddMonths (Дата, 1)), '00 ') as
Рік_місяць _
    1 as CALF // Рік_Місяць це наступний місяць по відношенню до РікМісяцьID
RESIDENT $(CALENDAR_TABLE)
WHERE Рік Місяць <= $( vEndMonth );

// Усі попередні місяця
Додатковий календарТмп :
LOAD Distinct
    Рік Місяць as РікМісяцьID
RESIDENT $(CALENDAR_TABLE)
WHERE FP = 1;

LEFT JOIN ( ДопКалендарТмп )
LOAD РікМісяць as Рік_Місяць
RESIDENT $(CALENDAR_TABLE)
WHERE FP = 1;

Concatenate ( Додатковий календар )
LOAD РікМісяцьID , Рік_Місяць , 2 as CALF
RESIDENT ДодатковийКалендарТмп
WHERE РікМісяцьID < Рік_Місяць and Рік_Місяць <> '2018-січ';

DROP TABLE ДодатковийКалендарТмп ;

// Додатковий КалендарМісяць_ENG :
left join ( ДопКалендар )
Load

```

```
    "Рік - Місяць " as " Рік_Місяць ",
    " Year-Month " as " Year_Month "
Resident $(CALENDAR_TABLE);
```

```
//
```

```
// Показники
```

```
//
```

```
// Змінна EXCH використовується для підтримки мультивалютності ( див. стор .
Мультивалюту ).
```

```
// ПОЛЯ:
```

```
SET vManager = 'Менеджер';
SET vSalesField = ' Сума ';
SET vSalesVATField = '[ Сума без ПДВ]';
SET vSalesDiscountField = '[ Сума знижки]';
SET vSalesPlanField = 'План';
SET vSalesDocField = 'Документ';
SET vOrderSalesField = '[ Сума замовлення]';
//SET vOrderNumberField = '[№ замовлення]';
SET vOrderNumberField = ' Замовлення_KEY ';
SET vOrderLineNumberField = '[№ рядка]';
SET vOrderSalesCountField = '[ Кількість замовлення]';
SET vSalesCountField = ' Кількість ';
SET vMarketSalesField = ' СумаРинок ';
SET vMarketCountField = ' КількістьРинок ';
SET vMarginField = 'Маржа';
SET vCostField = 'Собівартість';
//SET vSKUModelField = '[Код товару ]';
SET vProductCodeField = '[ Nomenclature_KEY ]';
SET vClientCodeField = '[ Contractor_KEY ]';
SET vTransTypeField = '[Тип транзакції]';
SET vPlaTypeField = '[Тип плану ]';
SET vInflCoefField = ' КоефІнфл ';
SET vProductField = 'Товар';
SET vProductIDField = '[Код товару | Товар]';
SET vUnitsCoeffField = ' КоефДля ОдиниціВиміри ';
SET vUnitsField = '[ Одиниця Виміри]';
//SET vUnitsFormatField = ' Формат Одиниці ';
//SET vBaseUnitField = '[ Базова одиниця Виміри]';
```

```
SET vYearField = 'Рік';
SET vQuarterField = 'Квартал';
SET vMonthField = ' Місяць ';
SET vYearMonthField = '[Рік- Місяць ]';
SET vYearMonthNField = '[ РікМісяць ]';
SET vYearQuarterNField = '[ РікКвартал ]';
SET vYearWeekNField = '[ РікТиждень ]';
SET vDateField = 'Дата';
SET vDateNField = ' DateNum ';
SET vDateSField = 'DS';
```

```
SET SALES_TRANS = ' Продаж '; // транзакція типу " Продаж "
SET RETURN_TRANS = ' Повернення ';
SET ORDER_WITH_SALES = '[Ознака наявність продажу]={1}'; //
використовується в сет-аналізі для виділення замовлень з продажами.
SET ORDER_INTIME = ' OrderInTime ={1}'; // використовується в
сет-аналізі для виділення замовлень виконаних " вчасно ".
SET ORDER_CLOSED = ' OrderClosed ={1}'; // Використовується в
сет-аналізі для виділення " виконаних " замовлень .
SET POSSIBLE_ORDER = ' Замовлення_KEY =P()'; //
```

```

використовується в сет-аналізі для виділення продажів із замовленнями
SET TO_SERVICES = '[Тип операції] = {`Надання послуг`}';
// використовується в сет-аналізі для виділення послуг

// SET TO_SALES = '[Тип операції]--={`Надання послуг`}'; //
використовується в сет-аналізі для виділення НЕ послуг
SET TO_SALES = '[Тип операції] = P()';
// використовується в сет-аналізі для виділення всіх можливих типів
операцій
SET TO_PRODUCT_SALES = '[Тип операції]--={`Надання послуг`}'; //
використовується в сет-аналізі для виділення НЕ послуг

SET CP = 'FP={1}'; // Поточний період (за поточною датою). Формування
календаря
//SET CAL_IGNORE = 'Рік=; Year =; Півріччя =; [ Half Year] =; Квартал =;
Quarter =; Місяць =; Month =; Рік-Місяць=; [Year-Month]=; [Рік-
Тиждень]=; [Year-Week]=; Тиждень=; Week=; Дата=; Date=';

SET CAL_IGNORE = '';
SET _EXCL_CAL_FIELDS = 'DateNum', 'IsWeekend', 'CY', 'PY', 'CM', 'PM', 'PYM',
'CQ', 'PQ', 'PYQ', 'CW', 'PW',
'PYW', 'CYD', 'PYD', 'CMD', 'PMD', 'PYMD', 'CQD', 'PQD', 'PYQD', 'CWD', 'PWD', 'PYWD',
'DS', 'DE', 'MS', 'ME', 'QS', 'QE', 'YS', 'YE', 'FP', 'InYTD', 'YearsAgo', 'QuartersAgo',
'MonthsAgo', 'WeeksAgo';

LET _calFields = NoOfFields (CALENDAR_TABLE);
FOR i = 1 TO _calFields
    LET _calFieldName = FieldName (i, CALENDAR_TABLE);
    IF match (_calFieldName, $_EXCL_CAL_FIELDS) = 0 THEN
        IF len (CAL_IGNORE) > 0 THEN
            LET CAL_IGNORE = CAL_IGNORE & ' ';
        ENDIF
    LET CAL_IGNORE = CAL_IGNORE & '['_calFieldName'] =';
    ENDIF
NEXT
LET _EXCL_CAL_FIELDS = null ();

LET _CAL_IGNORE_ = replace (CAL_IGNORE, ';', ',');
SET POSSIBLE_DATES = '$(vDateField)=P()';
SET POSSIBLE_MONTH = '$(vYearMonthNField)=P()';

// Можливість передавати параметри сет-аналізу у формулу через точку з
комою .

// Наприклад, vSales (Рік =; Місяць =). Крапка з комою замінюється на кому .

// Для передачі другого параметра використовується роздільник кома .

// Наприклад, для макророзстановки з двома параметрами ($1 і $2) можна
писати так: $(vSalesPlanForecast %(FP={1};CYD={1}, CY={1}))

// Перший параметр FP={1};CYD={1} буде перетворений на FP={1},CYD={1} і
підставлений замість $1. Замість $2 буде підставлено CY={1}.
SET PARAMS = '(= replace ('$1', ';', ','))';

// Формули із суфіксом C1 використовуються в порівняльному аналізі,
реалізованому за допомогою альтернативного стану C1.

// ПРОДАЖІ
SET _SALES_ = 'SUM({<$ (TO_SALES), $'&PARAMS'>} $(vSalesField) $'&EXCH&')';
SET _SALES_C1_ = 'SUM({[C1]<$ (TO_SALES), $'&PARAMS'>} $(vSalesField)
$'&EXCH&')';
SET _SALES_VAT_ = 'SUM({<$ (TO_SALES), $'&PARAMS'>} $(vSalesVATField)

```

```

$'&EXCH&')';
SET _SALES_VAT_C1_ = 'SUM({[C1]<$ (TO_SALES), $'&PARAMS&'>} $( vSalesVATField
) $'&EXCH&')';
SET _SALES_TOTAL_ = 'SUM({<$ (TO_SALES), $'&PARAMS&'>} total $( vSalesField )
$'&EXCH&')';
SET _SALES_ONLY_ = 'SUM({<$ (TO_SALES), $( vTransTypeField )={$(SALES_TRANS)},
$'&PARAMS&'>} $( vSalesField ) $'&EXCH&')';
SET _SALES_ONLY_C1_ = 'SUM({[C1]<$ (TO_SALES), $( vTransTypeField
)={$(SALES_TRANS)},$'&PARAMS&'>} $( vSalesField ) $'&EXCH&')';
SET _SALES_DISCOUNT_ = 'SUM({<$ (TO_SALES), $'&PARAMS&'>} $(
vSalesDiscountField ) $'&EXCH&')';
SET _SALES_DISCOUNT_C1_ = 'SUM({[C1]<$ (TO_SALES), $'&PARAMS&'>} $(
vSalesDiscountField ) $'&EXCH&')';
SET _SALES_DISCOUNT_TOTAL_ = 'SUM({<$ (TO_SALES), $'&PARAMS&'>} total $(
vSalesDiscountField ) $'&EXCH&')';

// Продажі послуг
SET _SALES_S_ = 'SUM({<$ (TO_SERVICES), $'&PARAMS&'>} $( vSalesField )
$'&EXCH&')';
SET _SALES_S_C1_ = 'SUM({[C1]<$ (TO_SERVICES), $'&PARAMS&'>} $( vSalesField )
$'&EXCH&')';
SET _SALES_VAT_S_ = 'SUM({<$ (TO_SERVICES), $'&PARAMS&'>} $( vSalesVATField )
$'&EXCH&')';
SET _SALES_VAT_S_C1_ = 'SUM({[C1]<$ (TO_SERVICES), $'&PARAMS&'>} $(
vSalesVATField ) $'&EXCH&')';
SET _SALES_DISCOUNT_S_ = 'SUM({<$ (TO_SERVICES), $'&PARAMS&'>} $(
vSalesDiscountField ) $'&EXCH&')';
SET _SALES_DISCOUNT_S_C1_ = 'SUM({[C1]<$ (TO_SERVICES), $'&PARAMS&'>} $(
vSalesDiscountField ) $'&EXCH&')';

// Тип транзакції повернення
SET _RETURNS_ONLY_ = ' fabs (SUM({<$ (TO_SALES)), $( vTransTypeField
)={$(RETURN_TRANS)},$'&PARAMS&'>} $( vSalesField ) $'&EXCH&')';
SET _RETURNS_C1_ONLY_ = ' fabs (SUM({[C1]<$ (TO_SALES)), $( vTransTypeField
)={$(RETURN_TRANS)},$'&PARAMS&'>} $( vSalesField ) $'&EXCH&')';

SET _SALES_COUNT_ = 'SUM({<$ (TO_SALES), $'&PARAMS&'>} $( vSalesCountField
))';
SET _SALES_COUNT_C1_ = 'SUM({[C1]<$ (TO_SALES), $'&PARAMS&'>} $(
vSalesCountField ))';
SET _SALES_COUNT_S_ = 'SUM({<$ (TO_SERVICES), $'&PARAMS&'>} $(
vSalesCountField ))';
SET _SALES_COUNT_S_C1_ = 'SUM({[C1]<$ (TO_SERVICES), $'&PARAMS&'>} $(
vSalesCountField ))';

SET _SALES_COUNT_ONLY_ = 'SUM({<$ (TO_SALES), $( vTransTypeField
)={$(SALES_TRANS)}, $'&PARAMS&'>} $( vSalesCountField ))';
SET _SALES_COUNT_C1_ONLY_ = 'SUM({[C1]<$ (TO_SALES), $( vTransTypeField
)={$(SALES_TRANS)}, $'&PARAMS&'>} $( vSalesCountField ))';

// ЗАМОВЛЕННЯ
SET _ORDER_SALES_ = 'SUM({<$'&PARAMS&'>} $( vOrderSalesField ) $'&EXCH&')';
//'SUM({<$ ( vTransTypeField )={$(SALES_TRANS)},$'&PARAMS&'>} $(
vOrderSalesField ))';
SET _ORDER_SALES_ONLY_ = 'SUM({<$ ( vTransTypeField
)={$(SALES_TRANS)},$'&PARAMS&'>} $( vOrderSalesField ) $'&EXCH&')';
SET _ORDER_SALES_C1_ = 'SUM({[C1]<$'&PARAMS&'>} $( vOrderSalesField )
$'&EXCH&')'; // 'SUM({[C1]<$ ( vTransTypeField
)={$(SALES_TRANS)},$'&PARAMS&'>} $( vOrderSalesField ))';
SET _ORDER_SALES_ONLY_C1_ = 'SUM({[C1]<$ ( vTransTypeField
)={$(SALES_TRANS)},$'&PARAMS&'>} $( vOrderSalesField ) $'&EXCH&')';

SET _ORDER_SALES_WS_ = 'SUM({<$ (ORDER_WITH_SALES), $'&PARAMS&'>} $(
vOrderSalesField ) $'&EXCH&')'; // 'SUM({<$ ( vTransTypeField

```

```

)={$(SALES_TRANS)},$'&PARAMS&'>} $( vOrderSalesField ));

SET _ORDER_SALES_COUNT_ = 'SUM({<$'&PARAMS&'>} $( vOrderSalesCountField ))';
//'SUM({<$ ( vTransTypeField )={$(SALES_TRANS)},$'&PARAMS&'>} $(
vOrderSalesCountField ))';
SET _ORDER_SALES_COUNT_TOTAL_ = 'SUM({<$'&PARAMS&'>} total $(
vOrderSalesCountField ))'; // 'SUM({<$ ( vTransTypeField
)={$(SALES_TRANS)},$'&PARAMS&'>} total $( vOrderSalesCountField ))';
SET _ORDER_SALES_COUNT_C1_ = 'SUM({[C1]<$'&PARAMS&'>} $(
vOrderSalesCountField ))';

SET _ORDER_SALES_COUNT_WS_ = 'SUM({<$ (ORDER_WITH_SALES), $'&PARAMS&'>} $(
vOrderSalesCountField ))';

SET _ORDERS_COUNT_ = 'COUNT({<$'&PARAMS&'>} distinct $( vOrderNumberField
))'; // 'COUNT({<$ ( vTransTypeField )={$(SALES_TRANS)},$'&PARAMS&'>}
distinct $( vOrderNumberField ))';
SET _ORDERS_COUNT_TOTAL_ = 'COUNT({<$'&PARAMS&'>} total distinct $(
vOrderNumberField ))';
// 'COUNT({<$ (vTransTypeField)={$(SALES_TRANS)},$'&PARAMS&'>}total
distinct $( vOrderNumberField ))';
SET _ORDERS_COUNT_C1_ = 'COUNT({[C1]<$'&PARAMS&'>} distinct $(
vOrderNumberField ))'; // 'COUNT({[C1]<$ ( vTransTypeField
)={$(SALES_TRANS)},$'&PARAMS&'>} distinct $( vOrderNumberField ))';

SET _ORDERS_COUNT_WS_ = 'COUNT({<$ (ORDER_WITH_SALES), $'&PARAMS&'>} distinct
$( vOrderNumberField ))'; // 'COUNT({<$ ( vTransTypeField
)={$(SALES_TRANS)},$'&PARAMS&'>} distinct $( vOrderNumberField ))';

// ПРОДАЖУ ПИHKY
SET _MARKET_SALES_ = 'SUM({<$'&PARAMS&'>} $( vMarketSalesField ) $'&EXCH&'');
SET _MARKET_SALES_TOTAL_ = 'SUM({<$'&PARAMS&'>} total $( vMarketSalesField )
$'&EXCH&'');

// ПРОДАЖУ (KИЛЬKИCTЬ) ПИHKY
SET _MARKET_SALES_COUNT_ = 'SUM({<$'&PARAMS&'>} $( vMarketCountField ))';

// MAPЖA
SET _MARGIN_ = 'SUM({<$ (TO_SALES), $'&PARAMS&'>} $( vMarginField )
$'&EXCH&'');
SET _MARGIN_ONLY_ = 'SUM({<$ (TO_SALES), $( vTransTypeField
)={$(SALES_TRANS)}, $'&PARAMS&'>} $( vMarginField ) $'&EXCH&'');
SET _MARGIN_ONLY_TOTAL_ = 'SUM({<$ (TO_SALES), $( vTransTypeField
)={$(SALES_TRANS)},$'&PARAMS&'>} total $( vMarginField ) $'&EXCH&'');
SET _MARGIN_TOTAL_ = 'SUM({<$ (TO_SALES), $'&PARAMS&'>} total $( vMarginField
) $'&EXCH&'');
SET _MARGIN_C1_ = 'SUM({[C1]<$ (TO_SALES), $'&PARAMS&' >} $( vMarginField )
$'&EXCH&'');

// COBIBAPTICTЬ
SET _COST_ = 'SUM({<$ (TO_SALES), $'&PARAMS&'>} $( vCostField ) $'&EXCH&'');
SET _COST_C1_ = 'SUM({[C1]<$ (TO_SALES), $'&PARAMS&' >} $( vCostField )
$'&EXCH&'');

// Kилькiсть днiв у календарi
SET _DAYS_ = ' rangesum (MAX({<$'&PARAMS&'>} $( vDateField )), -
MIN({<$'&PARAMS&'>} $( vDateField )), 1)';
SET _DAYS_COUNT_ = ' count ({{<$'&PARAMS&'>} distinct total $( vDateField ))';
// 'SUM({<$'&PARAMS&'>} total $(
vDateSField ))';
SET _DAYS_COUNT_CALENDAR_ = ' count ({{<$'&PARAMS&'>} distinct total $(
vDateField ))'; // 'SUM({<$'&PARAMS&'>} total $(

```

```

vDateSField ))';
SET _DAYS_COUNT_CALENDAR_C1_ = ' count ({C1<$'&PARAMS&'>} distinct total $(
vDateField ))';          //'SUM({C1<$'&PARAMS&'>} total $( vDateSField
));
SET _DAYS_COUNT_TOTAL_CALENDAR_ = ' count ({1<$'&PARAMS&'>} distinct total
<2> $( vDateField ))';          //'SUM({1<$'&PARAMS&'>} total <2> $(
vDateSField ))';
SET _DAYS_COUNT_C1_ = ' count ({C1<$'&PARAMS&'>} distinct total $( vDateField
));          //'SUM({[C1]<$'&PARAMS&'>} total $( vDateSField
));

LET vSales = $ (_SALES_);
LET vSalesC1 = $ (_SALES_C1_);
LET vSalesVAT = $ (_SALES_VAT_);
LET vSalesVATC1 = $ (_SALES_VAT_C1_);
LET vSalesDiscount = $ (_SALES_DISCOUNT_);
LET vSalesDiscountC1 = $ (_SALES_DISCOUNT_C1_);
LET vSalesDiscountTotal = $ (_SALES_DISCOUNT_TOTAL_);
LET vMaxSalesDiscount = 'MAX({<$ (TO_SALES), '$'&PARAMS&'>} $(
vSalesDiscountField ) '$&EXCH&')';
LET vSalesTotal = $ (_SALES_TOTAL_);
LET vSalesAll = 'SUM({1< '$'&PARAMS&'>} $( vSalesField ) '$&EXCH&')';

LET vSalesS = $ (_SALES_S_);
LET vSalesSC1 = $ (_SALES_S_C1_);
LET vSalesVATS = $ (_SALES_VAT_S_);
LET vSalesVATSC1 = $ (_SALES_VAT_S_C1_);
LET vSalesDiscountS = $ (_SALES_DISCOUNT_S_);
LET vSalesDiscountSC1 = $ (_SALES_DISCOUNT_S_C1_);

LET vSalesDiscount % = $ (_SALES_DISCOUNT_) &'/'& $ (_SALES_);
LET vSalesDiscountC1% = $ (_SALES_DISCOUNT_C1_) &'/'& $ (_SALES_C1_);

LET vSalesDiscount%S = $ (_SALES_DISCOUNT_S_) &'/'& $ (_SALES_S_);
LET vSalesDiscountC1%S = $ (_SALES_DISCOUNT_S_C1_) &'/'& $ (_SALES_S_C1_);

// Ринок '$'&PARAMS&'
LET vMarketSales = $ (_MARKET_SALES_);
LET vMarketSalesTotal = $ (_MARKET_SALES_TOTAL_);
LET vMarketSalesCount = $ (_MARKET_SALES_COUNT_);

// Тип Транзакції "Продажі"
LET vSalesOnly = $ (_SALES_ONLY_);
LET vSalesOnlyTotal = 'SUM({<$ (TO_SALES)}, $( vTransTypeField
)={$(SALES_TRANS)},'$'&PARAMS&'>} total $( vSalesField ) '$&EXCH&')';

// Тип Транзакції " Повернення "
LET vReturns = $ (_RETURNS_ONLY_);
LET vReturnsC1 = $ (_RETURNS_C1_ONLY_);

LET vMaxReturns = 'MAX({<$ (TO_SALES), $( vTransTypeField
)={$(RETURN_TRANS)}},'$'&PARAMS&'>} fabs $( vSalesField )) '$&EXCH&')';

LET vSalesAnalyticsTotal = 'SUM({<$ (TO_SALES), '$'&PARAMS&'>} total <
AnalyticsGroup > $( vSalesField ) '$&EXCH&')';

// Кількість проданих товарів :
LET vSalesCount = $ (_SALES_COUNT_);
LET vSalesCountC1 = $ (_SALES_COUNT_C1_);

//LET vAvgSalesCount = 'AVG({<$ (TO_SALES), '$'&PARAMS&'>} $( vSalesCountField
));

```

```

LET vAvgSalesCount = ' Sum ({<$ (TO_SALES), $'&PARAMS&'>} $( vSalesCountField
)) / Count ({<$ (TO_SALES), $'&PARAMS&'>} $( vSalesDocField ))';
LET vSalesCountTotal = 'SUM({<$ (TO_SALES), $'&PARAMS&'>} total $(
vSalesCountField ))';
LET vSalesCountOnly = $ (_SALES_COUNT_ONLY_);
LET vSalesCountOnlyTotal = 'SUM({<$ (TO_SALES), $( vTransTypeField
)={$(SALES_TRANS)}, $'&PARAMS&'>} total $( vSalesCountField ))';

LET vSalesCountS = $ (_SALES_COUNT_S_);
LET vSalesCountSC1 = $ (_SALES_COUNT_S_C1_);

// Кількість проданих товарів у вказаних одиницях
//SET vIsUnitsSelected = ' Only $( vBaseUnitField )<>'1';
SET vUnit = $( vUnitsField )=${1};
LET vSalesCountInUnits = 'SUM({<$ (TO_SALES), $'&PARAMS&'>} $(
vSalesCountField ) * $( vUnitsCoeffField ))';

LET vMargin = $ (_MARGIN_);
LET vMarginOnly = $ (_MARGIN_ONLY_);
LET vMarginOnlyTotal = $ (_MARGIN_ONLY_TOTAL_);
LET vMarginTotal = $ (_MARGIN_TOTAL_);
LET vMarginC1 = $ (_MARGIN_C1_);

// Собівартість:
LET vCost = $ (_COST_);
LET vCostC1 = $ (_COST_C1_);

// Кількість SKU:

//LET vSKUs = 'COUNT({<$'&PARAMS&'>} distinct $( vProductCodeField ))';
LET vSKUs = 'COUNT({<$ (TO_SALES), $( vTransTypeField
)={$(SALES_TRANS)},$(RETURN_TRANS)}, $'&PARAMS&'>} distinct $(
vProductCodeField ))'; // Підрахунок кол -ва SKU тільки з продажу та
повернення
LET vSKUsOnly = 'COUNT({<$ (TO_SALES), $( vTransTypeField )={$(SALES_TRANS)},
$'&PARAMS&'>} distinct $( vProductCodeField ))';
LET vSKUsOnlyTotal = 'COUNT({<$ (TO_SALES)}, $( vTransTypeField
)={$(SALES_TRANS)}, $'&PARAMS&'>} total distinct $( vProductCodeField ))';
LET vSKUsC1 = 'COUNT({[C1]<$ (TO_SALES), $'&PARAMS&'>} distinct $(
vProductCodeField ))';

//% націнки
LET vMarkup % = $ (_MARGIN_) &'/'& $ (_COST_);
LET vMarkupC1% = $ (_MARGIN_C1_) &'/'& $ (_COST_C1_);

//% собівартості
LET vCost % = $ (_COST_) &'/'& $ (_SALES_);
LET vCostC1% = $ (_COST_C1_) &'/'& $ (_SALES_C1_);

//% маржі
LET vMargin % = $ (_MARGIN_) &'/'& $ (_SALES_);
LET vMarginC1% = $ (_MARGIN_C1_) &'/'& $ (_SALES_C1_);

// Прогноз продажів // @2 - весь період , @1 - фактичний період
LET vSalesForecast = replace ('@ ( vSales (@1) ) * (@ (vDaysCountCalendar (@2)) /
@ (vDaysCountCalendar (@1)) )', '@', '$');

```

```

// Кількість днів у періоді
LET vDays = $ ( _DAYS_ );
LET vDaysCount = $ ( _DAYS_COUNT_ );
LET vDaysCountCalendar = $ ( _DAYS_COUNT_CALENDAR_ ); // Кількість днів у календарі , незалежно від відборів
LET vDaysCountCalendarC1 = $ ( _DAYS_COUNT_CALENDAR_C1_ );
LET vDaysCountTotalCalendar = $ ( _DAYS_COUNT_TOTAL_CALENDAR_ );
LET vDaysCountC1 = $ ( _DAYS_COUNT_C1_ );

// Сума замовлення продажу:
LET vOrderSales = $ ( _ORDER_SALES_ ); // 'SUM({<[Тип Транзакції]={ Продаж },$1>} [ Сума замовлення] )';
LET vOrderSalesC1 = $ ( _ORDER_SALES_C1_ ); // 'SUM({[C1]<[Тип Транзакції]={ Продаж }>} [ Сума замовлення] )';
LET vOrderSalesWS = $ ( _ORDER_SALES_WS_ ); // Замовлено Сума , тільки якщо є продаж

//% виконання замовлення (за сумою ):
LET vOrderSales % = $ ( _SALES_ ) & '/' & $ ( _ORDER_SALES_ );
LET vOrderSalesC1 % = $ ( _SALES_C1_ ) & '/' & $ ( _ORDER_SALES_C1_ );
LET vOrderSalesWS % = $ ( _SALES_ ) & '/' & $ ( _ORDER_SALES_WS_ );

// Кількість у замовленні продажу (без урахування повернень ):
LET vOrderSalesCount = $ ( _ORDER_SALES_COUNT_ );
LET vOrdersSalesCountInUnits = ' Num (SUM({<$ (TO_SALES), $' & PARAMS & '>} $ ( vOrderSalesCountField ) * $ ( vUnitsCoeffField ), $ ( vUnitsFormatField ))';
LET vOrderSalesCountWS = $ ( _ORDER_SALES_COUNT_WS_ ); // Замовлено кількість , тільки якщо є продаж

//% виконання замовлення (за кількістю ):
LET vOrderCount % = $ ( _SALES_COUNT_ ) & '/' & $ ( _ORDER_SALES_COUNT_ );
LET vOrderCountC1 % = $ ( _SALES_COUNT_C1_ ) & '/' & $ ( _ORDER_SALES_COUNT_C1_ );
LET vOrderCountWS % = $ ( _SALES_COUNT_ ) & '/' & $ ( _ORDER_SALES_COUNT_WS_ );

// Кількість замовлень типу продаж :
LET vOrdersCount = $ ( _ORDERS_COUNT_ );
LET vOrdersCountTotal = $ ( _ORDERS_COUNT_TOTAL_ );
LET vOrdersCountC1 = $ ( _ORDERS_COUNT_C1_ );
LET vOrdersCountWS = $ ( _ORDERS_COUNT_WS_ ); // Кількість замовлень з продажами

// Середня сума замовлення
LET vAvgSalesOrder = $ ( _ORDER_SALES_ ) & '/' & $ ( _ORDERS_COUNT_ );
LET vAvgSalesOrderC1 = $ ( _ORDER_SALES_C1_ ) & '/' & $ ( _ORDERS_COUNT_C1_ );

// Мінімальне / максимальне замовлення (без урахування повернень )
LET vMinSalesOrder = ' Min ( aggr ( ' & $ ( _ORDER_SALES_ONLY_ ) & ' , $ ( vOrderNumberField )) )';
LET vMaxSalesOrder = ' Max ( aggr ( ' & $ ( _ORDER_SALES_ONLY_ ) & ' , $ ( vOrderNumberField )) )';
LET vMinSalesOrderC1 = ' Min ( {[C1]} aggr ( ' & $ ( _ORDER_SALES_ONLY_C1_ ) & ' , $ ( vOrderNumberField )) )';
LET vMaxSalesOrderC1 = ' Max ( {[C1]} aggr ( ' & $ ( _ORDER_SALES_ONLY_C1_ ) & ' , $ ( vOrderNumberField )) )';

LET vMinSalesOrderFor = ' Min ( aggr ( ' & $ ( _ORDER_SALES_ONLY_ ) & ' , $ (

```

```

vOrderNumberField ), $2))';
LET vMaxSalesOrderFor = ' Max ( aggr ( ' & $_ORDER_SALES_ONLY_ ) & ', $(
vOrderNumberField ), $2))';
LET vMinSalesOrderForC1 = ' Min ([[C1]] aggr ( ' & $_ORDER_SALES_ONLY_C1_ ) &
', $( vOrderNumberField , $2))';
LET vMaxSalesOrderForC1 = ' Max ([[C1]] aggr ( ' & $_ORDER_SALES_ONLY_C1_ ) &
', $( vOrderNumberField , $2))';

// Середнє кількість замовлення
LET vAvgOrderSalesCount = $_ORDER_SALES_COUNT_ & '/' & $_ORDERS_COUNT_;
//% виконання замовлень вчасно :
LET vOrdersInTime % = 'COUNT({<$ ( vTransTypeField )={$(SALES_TRANS)}},
$(ORDER_INTIME), $'&PARAMS&'>} distinct $( vOrderNumberField )) / COUNT({<$ (
vTransTypeField )={$( SALES_TRANS)}, $'&PARAMS&'>} distinct $(
vOrderNumberField ))';
LET vOrdersInTimeC1% = 'COUNT({[C1]<$ ( vTransTypeField )={$(SALES_TRANS)}},
$(ORDER_INTIME), $'&PARAMS&'>} distinct $( vOrderNumberField )) /
COUNT({[C1]<$ ( vTransTypeField )={$(SALES_TRANS)}, $'&PARAMS&'>} distinct $(
vOrderNumberField ))';
//% закритих замовлень :
LET vOrdersClosed % = 'COUNT({<$ ( vTransTypeField )={$(SALES_TRANS)}},
$(ORDER_CLOSED), $'&PARAMS&'>} distinct $( vOrderNumberField )) / COUNT({<$ (
vTransTypeField )={$( SALES_TRANS)}, $'&PARAMS&'>} distinct $(
vOrderNumberField ))';
LET vOrdersClosedC1% = 'COUNT({[C1]<$ ( vTransTypeField )={$(SALES_TRANS)}},
$(ORDER_CLOSED), $'&PARAMS&'>} distinct $( vOrderNumberField )) /
COUNT({[C1]<$ ( vTransTypeField )={$(SALES_TRANS)}, $'&PARAMS&'>} distinct $(
vOrderNumberField ))';
// Кількість клієнтів :
//LET vClientsCount = 'COUNT({<$'&PARAMS&'>} distinct $( vClientCodeField
))';
LET vClientsCount = 'COUNT({<$ (TO_SALES), $( vTransTypeField
)={$(SALES_TRANS),$(RETURN_TRANS)}}, $'&PARAMS&'>} distinct $(
vClientCodeField ))'; // Підрахунок кількості клієнтів _ тільки з
продажу та повернення
//LET vClientsCountC1 = 'COUNT({[C1]<$'&PARAMS&'>} distinct $(
vClientCodeField ))';
LET vClientsCountC1 = 'COUNT({[C1]<$ (TO_SALES), $( vTransTypeField
)={$(SALES_TRANS),$(RETURN_TRANS)}}, $'&PARAMS&'>} distinct $(
vClientCodeField ))'; // Підрахунок кількості клієнтів _ тільки з продажу та
повернення

// Агрегування : Кількість позицій у замовленні
SET vAggrOrdersLines = ' aggr ( count ( $( vOrderLineNumberField ),$(
vOrderNumberField ))';
// Для розрахунку % звернень до товарів (FMR)
//SET vFMRtotal = ' sum ( total aggr (COUNT( distinct $( vOrderNumberField
)), $( vProductField )))';
LET vFMRtotal = ' sum ( total aggr (COUNT({<$ (TO_SALES), $( vTransTypeField
)={$(SALES_TRANS),$(RETURN_TRANS)}}, $'&PARAMS&'>} distinct $(
vOrderNumberField )), $( vProductField )))';
LET vFMRtotal2 = ' sum ( total aggr (COUNT({<$ (TO_SALES), $( vTransTypeField
)={$(SALES_TRANS),$(RETURN_TRANS)}}, $'&PARAMS&'>} distinct $(
vOrderNumberField )), $( vProductIDField )))';

// Перша дата продажу
LET vFirstSalesDate = replace ('MIN({<$ ( vTransTypeField )={$(SALES_TRANS)}},
$'&PARAMS&'>} [ @ (= @ ( T ( fldDate ) ) ) ] )', '@', '$ ');

// Остання дата продажу
LET vLastSalesDate = replace ('MAX({<$ (CP), $( vTransTypeField
)=$ (SALES_TRANS)}}, $'&PARAMS&'>} [ @ (= @ ( T ( fldDate ) ) ) ] )', '@', '$ ');

```

```

// Активних (на періоді аналізу ): CALF={1} - продажі в попередньому місяці ,
CALF={0} - продажі в поточному місяці

//LET vActive = replace (' if
(@ (vSales (CALF={1})>*<Пік_Місяць=P(ПікМісяцьID)))> 0 and
@(vSales (CALF={0})>*<Пік_Місяць=P(ПікМісяцьID) ))> 0, 1, 0)', '@', '$');
LET vActive = replace (' if
(@ (vSalesOnly (CALF={1};Пік_Місяць=P(ПікМісяцьID)))> 0 and
@(vSalesOnly (CALF={0};Пік_Місяць=P(ПікМісяцьID)))> 0 , 1, 0) ', '@', '$');
LET vPassive = replace (' if
(@ (vSalesOnly (CALF={0};Пік_Місяць=P(ПікМісяцьID))) = 0 and
@(vSalesOnly (CALF={2};Пік_Місяць=P(ПікМісяцьID)))> 0 , 1, 0) ', '@', '$');

// Повернулися : CALF={0} є продажі в поточному місяці , CALF={1} немає
продажів у попередньому місяці , CALF={2} але є продажі в попередніх місяцях
LET vReturn = replace (' if
(@ (vSalesOnly (CALF={0};Пік_Місяць=P(ПікМісяцьID)))> 0 and
@(vSalesOnly (CALF={1};Пік_Місяць=P(ПікМісяцьID))) = 0 and
@(vSalesOnly (CALF={2};Пік_Місяць=P(ПікМісяцьID)))> 0, 1, 0)', '@', '$');

// Нових (на періоді аналізу ): CALF={2} - продажі у всіх попередніх місяцях
, CALF={0} - продажі поточного місяця

//>*<
LET vNew = replace (' if (@ (vSalesOnly (CALF={2};Пік_Місяць=P(ПікМісяцьID)))
= 0 and @ (vSalesOnly (CALF={0};Пік_Місяць=P(ПікМісяцьID)))> 0 , 1, 0) ', '@',
'$');

// Активних (на періоді аналізу ): CALF={1} - продажі в попередньому місяці ,
CALF={0} - продажі в поточному місяці

//LET vActive = replace (' if
(@ (vSales (CALF={1})>*<Пік_Місяць=P(ПікМісяцьID)))> 0 and
@(vSales (CALF={0})>*<Пік_Місяць=P(ПікМісяцьID) ))> 0, 1, 0)', '@', '$');

// LET vActive = replace (' if
(@ (vSalesOnly (CALF={1};Пік_Місяць=P(ПікМісяцьID)))> 0 and
@(vSalesOnly (CALF={0};Пік_Місяць=P(ПікМісяцьID)))> 0, 1, 0) ', '@', '$');

//// Пасивних ( на періоді аналізу ): CALF={0} - продажі в поточному місяці ,
CALF={2} - продажі у всіх попередніх місяцях

// LET vPassive = replace (' if
(@ (vSalesOnly (CALF={0};Пік_Місяць=P(ПікМісяцьID))) = 0 and
@(vSalesOnly (CALF={1};Пік_Місяць=P(ПікМісяцьID)))> 0 and
@(vSalesOnly (CALF={2};Пік_Місяць=P(ПікМісяцьID)))> 0 , 1, 0)', '@', '$');

// // Повернувшись : CALF={0} є продажі в поточному місяці , CALF={1} немає
продажів у попередньому місяці , CALF={2} але є продажі в попередніх місяцях

// LET vReturn = replace (' if
(@ (vSalesOnly (CALF={0};Пік_Місяць=P(ПікМісяцьID)))> 0 and
@(vSalesOnly (CALF={1};Пік_Місяць=P(ПікМісяцьID)))> 0 and
@(vSalesOnly (CALF={2};Пік_Місяць=P(ПікМісяцьID)))> 0 , 1, 0)', '@', '$');

// // Нових (на періоді аналізу ): CALF={2} - продажі у всіх попередніх
місяцях , CALF={0} - продажі поточного місяця

// // >*<

// LET vNew
= replace (' if
(@ (vSalesOnly (CALF={2};Пік_Місяць=P(ПікМісяцьID))) = 0 and
@(vSalesOnly (CALF={0};Пік_Місяць=P(ПікМісяцьID)))> 0, 1, 0) ', '@', '$');

```

```

// //LET vReturn = replace (' if
(@ (vSalesOnly (CALF={0};Pік_Місяць=P (PікМісяцьID))) > 0 and
@ (vSalesOnly (CALF={1};Pік_Місяць=P (PікМісяцьID))) > 0 and
@ (vSalesOnly (CALF={2};Pік_Місяць=P (PікМісяцьID))) > 0 , 1, 0)', '@', '$');

// // Активні (на періоді аналізу ): CALF={1} - продажі в попередньому місяці
, CALF={0} - продажі в поточному місяці

// //LET vActive = replace (' if
(@ (vSalesOnly (CALF={1};Pік_Місяць=P (PікМісяцьID))) > 0 and
@ (vSalesOnly (CALF={0};Pік_Місяць=P (PікМісяцьID))) > 0, 1, 0)', '@', '$');

// //LET vActive = replace (' if
(@ (vSalesOnly (CALF={1};Pік_Тиждень=P (PікТижденьID))) > 0 and
@ (vSalesOnly (CALF={0};Pік_Тиждень=P (PікТижденьID))) > 0, 1, 0)', '@', '$');

// LET vActive =
replace ('if (@ (vSalesOnly (@ (= [CALF])) = {1}; @ (= [KA_F]) = P (@ (= [KA_Id]))) > 0 and
@ (vSalesOnly (@ (= [CALF])) = {0}; @ (= [KA_F]) = P (@ (= [KA_Id]))) > 0, 1, 0)', '@', '$
');

//// Пасивних (на періоді аналізу ): CALF={0} - продажі в поточному місяці ,
CALF={2} - продажі у всіх попередніх місяцях

// LET vPassive = replace (' if
(@ (vSalesOnly (CALF={0};Pік_Місяць=P (PікМісяцьID))) = 0 and
@ (vSalesOnly (CALF={2};Pік_Місяць=P (PікМісяцьID))) > 0, 1, 0)', '@', '$');

// LET vPassive = replace (' if
(@ (vSalesOnly (CALF={0};Pік_Тиждень=P (PікТижденьID))) = 0 and
@ (vSalesOnly (CALF={2};Pік_Тиждень=P (PікТижденьID))) > 0, 1, 0)', '@', '$');

// LET vPassive =
replace ('if (@ (vSalesOnly (@ (= [CALF])) = {0}; @ (= [KA_F]) = P (@ (= [KA_Id]))) = 0 and
@ (vSalesOnly (@ (= [CALF])) = {2}; @ (= [KA_F]) = P (@ (= [KA_Id]))) > 0, 1, 0)', '@', '$
');

// Повернувшись : CALF={0} є продажі в поточному місяці , CALF={1} немає
продажів у попередньому місяці , CALF={2} але є продажі в попередніх місяцях

// LET vReturn = replace (' if
(@ (vSalesOnly (CALF={0};Pік_Місяць=P (PікМісяцьID))) > 0 and
@ (vSalesOnly (CALF={1};Pік_Місяць=P (PікМісяцьID))) = 0 and
@ (vSalesOnly (CALF={2};Pік_Місяць=P (PікМісяцьID))) > 0, 1, 0)', '@', '$');

// LET vReturn = replace (' if
(@ (vSalesOnly (CALF={0};Pік_Тиждень=P (PікТижденьID))) > 0 and
@ (vSalesOnly (CALF={1};Pік_Тиждень=P (PікТижденьID))) = 0 and
@ (vSalesOnly (CALF={2};Pік_Тиждень=P (PікТижденьID))) > 0, 1, 0)', '@', '$');

// LET vReturn =
replace ('if (@ (vSalesOnly (@ (= [CALF])) = {0}; @ (= [KA_F]) = P (@ (= [KA_Id]))) > 0 and
@ (vSalesOnly (@ (= [CALF])) = {1}; @ (= [KA_F]) = P (@ (= [KA_Id]))) = 0 and
@ (vSalesOnly (@ (= [CALF])) = {2}; @ (= [KA_F]) = P (@ (= [KA_Id]))) > 0, 1, 0)', '@',
'$');
LET vAnalyticsFromDefault = replace ('[@ (=getcurrentfield (AnalyticsGroup))]
= @::[@ (=getcurrentfield (AnalyticsGroup))]', '@', '$');
LET vAnalyticsFromDefault = replace ('[@ (=getcurrentfield (AnalyticsGroup))]
= @::[@ (=getcurrentfield (AnalyticsGroup))]', '@', '$');
SET vLFLMonthQuarteYearEnable = " if ( match ( Only ( _ LFLPeriod ), '$(
vYearField )', '$( vQuarterField )', '$( vMonthField )') > 0, true (), false
())";
LET vCurrentYear = replace ('$( vYearField )=@ (=Max ($ (vYearField)))');

```

```

FP={1}', '@', '$');
LET vCurrentYearFull = replace ('$( vYearField )={@ (=Max ($ (vYearField)))}',
 '@', '$');
SET vCurrentYearLabel = ' Max ($ ( vYearField ))';

// Попередній рік
LET vPreviousYear = replace ('$( vYearField )={@ (=Max ($ (vYearField))-1)}',
 '@', '$');
SET vPreviousYearLabel = ' ( Max ($ ( vYearField )) - 1)';
LET vYearLabel = ' Max ({<$' &PARAMS &'>}$( vYearField ))';
LET IsMaxYear = ' Max ({1}$ ( vYearField ))= Max ($ ( vYearField ))';

// Додаткові змінні ( аналіз множин )
LET vMinDate = ' Min ({<$' &PARAMS &'>}$( vDateField ))';
LET vMaxDate = ' Max ({<$' &PARAMS &'>}$( vDateField ))';
LET vMinDateN = ' Min ({<$' &PARAMS &'>}$( vDateNField ))';
LET vMaxDateN = ' Max ({<$' &PARAMS &'>}$( vDateNField ))';

// Формати використовуються при відображенні на сторінці показники
SET vPlan%Format = '###0%;-###0%';
SET vGrowth%Format = '▲# ##0%;▼# ##0%';//'+###0%;-###0%';
SET vPlanForecastFormat = '###0,0%;-###0,0%'; //→

// Вираз умови для розрахунку показника , розрахункове значення умови в альт.
стан для запобігання перерахунку всіх об'єктів аркуша
LET CondExp = ' SubStringCount ( Concat ( { DashBoard } _ ВиразДашбоарда №,
 ''|'' ), $' &'1)';
// Поле _ ТипПлану - унікальне , числове ,
// Але значення в полі повинні відповідати значенням "Тип плану " для планів
// Поле _ ТипПлану - Always One Selected Value
// _ ПланId - ідентифікатор рядка у файлі translation.xlsx
// План за замовчуванням
SET DEFAULT_PLANTYPE = 0;
// Вимір за замовчуванням
SET DEFAULT_PLANDIM = 1;
// Для аркуша " Показники " - відображення " основного " плану
SET saDEFAULT_PLANTYPE = '$ ( vPlaTypeField )={$( DEFAULT_PLANTYPE )}';
SET sa_DEFAULT_PLANTYPE = ' _ ТипПлану ={$ ( DEFAULT_PLANTYPE )}';
SET saDEFAULT_PLANDIM = ' _ ВиміриДляПлануPK ={$ ( DEFAULT_PLANDIM )}';.
Типи Планів :
LOAD
_ ТипПлану ,
_ ПланId ,
_ Змінна Факта Для Плану ,
_ ПолеДляПерерахунку
INLINE [
_ ТипПлану , _ ПланId , _ ЗміннаФактаДляПлану , _
ПолеДляПерерахунку
0, PlanType1, vSales ,
ExchangeFormula
1, PlanType2, vSales ,
ExchangeFormula
];
// Поля, для відповідного типу плану (для план- фактного аналізу )
// _ ПланВимірId - підтримка мультимовних планів . Ключ у файлі
translation.xlsx
// _ ВиміриДляПлануPK - числове унікальне ! Використовується в
FirstSortedValue як sort_weight
LEFT JOIN ( Типи Планів )
LOAD * INLINE [
_ ТипПлану , _ ВиміриДляПлануPK , _ ПланВимірId , _ ПланВимірМіткаId
0, 1, fldMonth , Month
0, 2, fldProductCategory , ProductCategory
1, 3, fldMonth , Month

```

```

1,          4,          fldManager ,          Manager
];
LEFT JOIN ( Типи Планів )
LOAD _ Language , id as _ ПланId , _ Translation as _План RESIDENT
Translations ;
LEFT JOIN ( Типи Планів )
LOAD _ Language , id as _ ПланВимірId , _ Translation as _ ПланВимір RESIDENT
Translations ;
LEFT JOIN ( Типи Планів )
LOAD _ Language , id as _ ПланВимірМіткаId , _ Translation as _
ПланВимірМітка RESIDENT Translations ;
SET _PLANTYPE_ = '(= FirstSortedValue ( _ ТипПлану , _ ВиміриДляПлануPK ))';
// '(= FirstSortedValue ( _ ТипПлану , _ ТипПлану ))'
LET PLAN_DIMENSION = replace ('@(@T(@=FirstSortedValue( _ПланВимірId, _
ВиміриДляПлануPK ))))', '@', '$');
SET IS_PLAN_MONTH_DIM = _ ПланВимірId = ' fldMonth '; // _ ВиміриДляПлануPK =
1;
SET _PLANEXCH_ = '(= FirstSortedValue ( _ ПолеДляПерерахунку , _
ВиміриДляПлануPK ))'; // '(= FirstSortedValue ( _ ПолеДляПерерахунку , _
ТипПлану ))'
LET PLANEXCH = '(= FirstSortedValue ($'_PLANEXCH_', CurrencyPK ))';
SET _PLAN_FACT_VAR = ' FirstSortedValue ( _ ЗміннаФактаДляПлану , _
ВиміриДляПлануPK )'; // ' FirstSortedValue ( _ Змінна ФактаДляПлану , _
ТипПлану )'

// Для аркуша "ПОКАЗНИКИ"
LET vPlanFactVar = ' FirstSortedValue ( {<$'&PARAMS&'>} _ ЗміннаФактаДляПлану ,
_ ВиміриДляПлануPK )';

// ПЛАН ПРОДАЖІВ
SET _SALES_PLAN_ = 'SUM({<$ ( vPlaTypeField )=>{$'&_PLANTYPE_'}, $'&PARAMS&'>}
$( vSalesPlanField ) $'&PLANEXCH&')';
SET _SALES_PLAN_TOTAL_ = 'SUM({<$ ( vPlaTypeField )=>{$'&_PLANTYPE_'},
$'&PARAMS&'>} total $( vSalesPlanField ) $'&PLANEXCH&')';
SET _SALES_PLAN_C1_ = 'SUM({[C1]<$ ( vPlaTypeField )=>{$'&_PLANTYPE_'},
$'&PARAMS&'>} $( vSalesPlanField ) $'&PLANEXCH&')';

// План продажів:
LET vSalesPlan = $( _SALES_PLAN_ );
LET vSalesPlanTotal = $( _SALES_PLAN_TOTAL_ );
LET vSalesPlanC1 = $( _SALES_PLAN_C1_ );

// Факт - у залежності від обраного типу плану
LET vFact = replace ('@(@= $( _PLAN_FACT_VAR )) (@1)', '@', '$');
LET vFactC1 = replace ('@(@= $( _PLAN_FACT_VAR )) C1 (@1)', '@', '$');
LET vSalesForecastForPlan = replace ('@(@= $( _PLAN_FACT_VAR )) (@1) * (
@(vDaysCountCalendar (@2)) / @(vDaysCount (@1)) )', '@', '$');

//LET vSalesPlanForecastForPlan % = replace
('@(@=Only( _ЗміннаФактаДляПлану )) (@1) / ( @(vSalesPlan (@2)) /
@(vDaysCountCalendar (@2)) * @(vDaysCountCalendar (@1)) ) )', '@', '$');

//% виконання плану
LET vSalesPlan % = replace ('@(@= $( _PLAN_FACT_VAR )) (@1)', '@',
'$') & '/' & $( _SALES_PLAN_ );
LET vSalesPlanC1 % = replace ('@(@= $( _PLAN_FACT_VAR )) C1 (@1)', '@',
'$') & '/' & $( _SALES_PLAN_C1_ );

// Прогноз виконання плану , %
LET vSalesPlanForecast % = replace (' (@(vFact (@1))) / (@(vSalesPlan (@2))) *
if (@(IS_PLAN_MONTH_DIM), @(vDaysCountTotalCalendar ($ (POSSIBLE_DATES); @2, [
(PLAN_DIMENSION)])) / @(vDaysCountTotalCalendar ( $ (POSSIBLE_DATES); @1,
[(PLAN_DIMENSION)])) ), @(vDaysCountCalendar (@2)) / @(vDaysCountCalendar ( @1
))))', '@', '$');

```

```

LET vMaxPlanMonth = replace ('$( vYearMonthNField )={@ (=Max({<FP={1}>} $(
vYearMonthNField )))}', '@', '$');
LET vMaxPlanMonthLabel = replace (' Only ({<$ ( vYearMonthNField
))={@ (=Max({<FP={1}>} $( vYearMonthNField )))>} $( vYearMonthField ) )',
'@', '$');
LET vLFLCurrStartDate = vStartPeriod ;
LET vLFLCurrEndDate = '= num ( max ([ _ LFLData ]))';
SET vPrevStr = '@ (= FirstSortedValue (@ (=FirstSortedValue(_LFLPrevField,
_LFLPK)), _ Order ))';
SET vCurrStr = '@ (= FirstSortedValue (@ (=FirstSortedValue(_LFLCurrField,
_LFLPK)), _ Order ))';
SET vCurrFullStr = '@ (= FirstSortedValue
(@ (=FirstSortedValue(_LFLCurrFieldFull, _LFLPK)), _ Order ))';
SET vPrevFullStr = '@ (= FirstSortedValue
(@ (=FirstSortedValue(_LFLPrevFieldFull, _LFLPK)), _ Order ))';
SET vPrevYearOnlyStr =
'@ (=FirstSortedValue (@ (=FirstSortedValue({<_LFLPeriodLabelId={LFLPreviousYear
}>} _LFLPrevField, _LFLPK)), _ Order ))';
LET vCurrFull = replace ('$( vCurrFullStr )', '@', '$'); // Поточний період
( повний : до кінця періоду )
LET vPrevFull = replace ('$( vPrevFullStr )', '@', '$'); // Попередній
період ( повний : до кінця періоду )
LET vCurr = replace ('$( vCurrStr )', '@', '$'); // Поточний
період (за поточною датою)
LET vPrev = replace ('$( vPrevStr )', '@', '$'); // Попередній
( аналогічний ) період
LET vCurrYear = replace ('@ (= Only ({<_ LFLPeriod ={Pik}>})
@ (=FirstSortedValue(_LFLCurrField, _LFLPK) ))', '@', '$');
LET vPrevYear = replace ('@ (= Only ({<_ LFLPeriod ={Pik}>}
@ (=FirstSortedValue(_LFLPrevField, _LFLPK) ))', '@', '$');
LET vCurrYearFull = replace ('@ (= Only ({<_ LFLPeriod ={Pik}>})
@ (=FirstSortedValue(_LFLCurrFieldFull, _LFLPK) ))', '@', '$');
LET vCurrentMax = replace ('@ (= FirstSortedValue ( _ CurrentMax , _ Order ) )',
'@', '$');
LET vCurrentMaxLabel = replace ( ' MaxString
({<@ (=FirstSortedValue( _ CurrentMax, _ Order
))>})@ (=Only({<@ (=FirstSortedValue( _ CurrentMax, _ Order ) ) }) _ LFL ', '@',
'$');
LET vPreviousMax = replace ('@ (= FirstSortedValue ( _ PreviousMax , _ Order
))', '@', '$');
LET vPreviousPeriodMax = replace ('@ (= FirstSortedValue ( _ PreviousPeriodMax
, _ Order ) )', '@', '$');
LET vPrevLabel = replace (' if ( len ( MaxString ({1<$ ( vPrevStr ),
$'&PARAMS&'>}) [@ (=FirstSortedValue(_LFLLabelField, _ Order ))]) > 0, text
( MaxString ({1 <$ ( vPrevStr ), $'&PARAMS&'>}
[@ (=FirstSortedValue(_LFLLabelField, _ Order ))]), @ (vPrevPeriod))', '@',
'$');
LET vCurrLabel = replace (' if ( len ( MaxString ({1<$ ( vCurrStr ),
$'&PARAMS&'>}) [@ (=FirstSortedValue(_LFLLabelField, _ Order ))]) > 0, text (
MaxString ({1 <$ ( vCurrStr ), $'&PARAMS&'>}
[@ (=FirstSortedValue(_LFLLabelField, _ Order ))]), @ (vCurrPeriod))', '@',
'$');
LET vLabelFor = replace (' text ( MaxString ({1<_ Language =P(),
$'&PARAMS&'>} total [@ (=FirstSortedValue([_LFLLabelField], _ Order ))])',
'@', '$');
LET vPrevPeriod = replace (' Date ( Min ({1<$ ( vPrevStr ), $'&PARAMS&'>} $(
vDateField ) ), '& chr (39) &'@ (@ (T (CalendarShortDateFormat)))'& chr (39) &')',
'@', '$') & chr (38) & chr (39) & '-' & chr (39) & chr (38) & replace (' Date
( Max ({1 <$ ( vPrevStr ), $'&PARAMS&'>} $( vDateField ) ), '& chr
(39) &'@ (@ (T (CalendarShortDateFormat)))'& chr (39) &')', '@', '$');
LET vCurrPeriod = replace (' Date ( Min ({1<$ ( vCurrStr ), $'&PARAMS&'>} $(
vDateField ) ), '& chr (39) &'@ (@ (T (CalendarShortDateFormat)))'& chr (39) &')',
'@', '$') & chr (38) & chr (39) & '-' & chr (39) & chr (38) & replace (' Date
( Max ({1 <$ ( vCurrStr ), $'&PARAMS&'>} $( vDateField ) ), '& chr

```

```

(39)'@(@ (T(CalendarShortDateFormat)))&chr(39)&')', '@', '$');
LET vPeriod = replace (' Date ( Min ({<$&PARAMS>} $( vDateField )), '& chr
(39)'@(@ (T(CalendarShortDateFormat)))&chr(39)&')', '@', '$') & chr (38) &
chr (39) & ' - ' & chr (39) & chr (38) & replace (' Date ( Max
({<$&PARAMS>}) $( vDateField )), '& chr
(39)'@(@ (T(CalendarShortDateFormat)))&chr(39)&')', '@', '$');

// replace (' Date (@(=Only(_LFLPeriodStartField))), '& chr (39)&'$(
ShortDateFormat )& chr (39)&')', '@', '$') & chr ( 38) & chr (39) & '-' &
chr (39) & chr (38) & replace (' Date ( vLFLCurrEndDate , '& chr (39)'$(
ShortDateFormat )& chr (39)&')', '@', '$');
LET vCurrFullPeriod = replace (' Date ( Min ({1<$( vCurrFullStr ),
$&PARAMS>} $( vDateField )), '& chr
(39)'@(@ (T(CalendarShortDateFormat)))&chr (39)&')', '@', '$') & chr (38) &
chr (39) & '-' & chr (39) & chr (38) & replace (' Date ( Max ({1<$(
vCurrFullStr ), $&PARAMS>} $( vDateField )), '& chr
(39)'@(@ (T(CalendarShortDateFormat)))&chr(39)&')', '@', '$');
LET vPrevFullPeriod = replace (' Date ( Min ({1<$( vPrevFullStr ),
$&PARAMS>} $( vDateField )), '& chr
(39)'@(@ (T(CalendarShortDateFormat)))&chr (39)&')', '@', '$') & chr (38) &
chr (39) & '-' & chr (39) & chr (38) & replace (' Date ( Max ({<< $(
vPrevFullStr ), $&PARAMS>} $( vDateField )), '& chr
(39)'@(@ (T(CalendarShortDateFormat)))&chr(39)&')', '@', '$ ');
LET vLFLDim = replace (' ValueList ( Dual
(''@=(vCurrLabel(@ (CAL_IGNORE))))', 1), Dual
(''@=(vPrevLabel(@ (CAL_IGNORE))))', 2))', '@', '$');
LET vLFLDim2 = replace (' ValueList ( Dual
(''@=(vCurrLabel(@ (CAL_IGNORE))))', 1), Dual (''+'', 2), Dual (''%',
3), Dual (''@=(vPrevLabel(@ (CAL_IGNORE))))', 4))', '@', '$');
LET vCurrentPeriodLabel = replace (' pick ( Only ({1<_ Language =P(>)} )
LFLCurrentPeriodNo ), @ (vLabelFor(@ (Only(_ Current))))),
@ (vLabelFor(@ (Only(_ PreviousFull))))', '@', '$');
LET leYTD = replace ('$( vDateNField )={ "<=@ (=Num(vLFLCurrEndDate))" }', '@',
'$');
LET lePYTD = replace ('$( vDateNField )={ "<=@ (=Num(AddYears (vLFLCurrEndDate,
-1*@(=Max(Pik)-Max(Pik, 2)))))" }', '@', '$');
LET lePYWTD =
replace ('$( vDateNField )={ "<=@ (=Num(MakeWeekDate (Year (AddYears (vLFLCurrEndDate
, -1*@(=Max(Pik)-Max(Pik, 2))))))", Week ( vLFLCurrEndDate ), WeekDay (
vLFLCurrEndDate ))))" }', '@', '$');
LET saYTDYearsAgo = replace (' YearsAgo ={@ (=Min({<@(leYTD)>}YearsAgo) +
$1)}', '@', '$');
LET saMaxTwoYearsDiff = replace ('@ (= Max (Pik) - Max (Pik, 2))', '@', '$');
SET _LFLPeriod = ' FirstSortedValue ( _LFLPeriod , _ Order )';
SET LFLPeriodCurrent = _LFLCurrentPeriod = ' Поточний ';
SET LFLPeriodPrevious = _LFLCurrentPeriod = ' Попередній ';
SET LFLPeriodLabelPrevYear = _LFLPeriodLabelId = ' LFLPreviousYear ';
Виміри LFL :
LOAD * INLINE [
_ ПолеВиміриLFLId , _ МіткаВиміриLFLId ,
fldProductHierarchy1, ProductHierarchy1
fldProductHierarchy2, ProductHierarchy2
fldProductHierarchy3, ProductHierarchy3
fldClientCategory , ClientCategory
fldClientGroup , ClientGroup
fldClient , Client
fldManager , Manager
fldSupplier , Supplier
fldProductKind , ProductKind
fldProductCategory , ProductCategory
fldNomenclatureGroup , NomenclatureGroup
fldProductGroup , ProductGroup
fldProductCountry , ProductCountry
fldProduct , Product

```

```

];
LEFT JOIN ( ВиміриLFL )
LOAD _ Language , id as _ ПолеВиміриLFLId , _ Translation as _ ВимірLFL
RESIDENT Translations ;
LEFT JOIN ( ВиміриLFL )
LOAD _ Language , id as _ МіткаВиміриLFLId , _ Translation as _
МіткаВиміриLFL RESIDENT Translations ;

// Тип порівняння LFL: поточний період , попередній період
LFLTypes :
LOAD
Autonumber ( _ LFLCurrentPeriod , ' _ LFLCurrentPeriod ' ) as _
LFLCurrentPeriodNo ,
* INLINE [
_LFLPK, _ LFLPeriodLabelId , _ LFLCurrentPeriod , _ LFLCurrField ,
_LFLPrevField , _ LFLCurrFieldFull , _
LFLPrevFieldFull
1, _ LFLPreviousYear , Поточний ,
Current , _ PreviousYear , _ CurrentFull ,
_PreviousYearFull
2, _ LFLPreviousPeriod , Поточний ,
Current , _ Previous , _ CurrentFull ,
_PreviousFull
3, _ LFLPreviousYear , Попередній ,
PreviousFull , _ PrevPreviousYearFull , _ PreviousFull ,
_PrevPreviousYearFull
4, _ LFLPreviousPeriod , Попередній ,
PreviousFull , _ PrevPreviousFull , _ PreviousFull , _
_PrevPreviousFull
];
LEFT JOIN ( LFLTypes )
LOAD _ Language , id as _ LFLPeriodLabelId , _ Translation as _ LFLType
RESIDENT Translations ;
LFLPeriods :
LOAD
' fldYear ' as _ LFLId
, ' Year ' as _ LFLPeriodId
, ' YearStart ( vLFLCurrEndDate ) ' as _ LFLPeriodStartField
, replace ( '$ (vDateNField)={ ">=@ (=num (YearStart (vLFLCurrEndDate))) <=@ (=num (vLFL
CurrEndDate))" }', '@', '$') as _ Current
, replace ( '$ (vDateNField)={ ">=@ (=num (YearStart (vLFLCurrEndDate))) <=@ (=Floor (Ye
arEnd (vLFLCurrEndDate)))" }', '@', '$') as _ CurrentFull
, replace ( '$ (vDateNField)={ ">=@ (=num (YearStart (AddYears (vLFLCurrEndDate, -
1)))) <=@ (=num (AddYears (vLFLCurrEndDate, -1)))" }', '@', '$') as _ Previous
, replace ( '$ (vDateNField)={ ">=@ (=num (YearStart (AddYears (vLFLCurrEndDate, -
1)))) <=@ (=Floor (YearEnd (AddYears (vLFLCurrEndDate, -1))))" }', '@', '$') as _
PreviousFull
, replace ( '$ (vDateNField)={ ">=@ (=num (YearStart (AddYears (vLFLCurrEndDate, -
2)))) <=@ (=Floor (YearEnd (AddYears (vLFLCurrEndDate, -2))))" }', '@', '$') as _
PrevPreviousFull
, replace ( '$ (vDateNField)={ ">=@ (=num (YearStart (AddYears (vLFLCurrEndDate, -
1)))) <=@ (=num (AddYears (vLFLCurrEndDate, -1)))" }', '@', '$') as _ PreviousYear
, replace ( '$ (vDateNField)={ ">=@ (=num (YearStart (AddYears (vLFLCurrEndDate, -
1)))) <=@ (=Floor (YearEnd (AddYears (vLFLCurrEndDate, -1))))" }', '@', '$') as _
PreviousYearFull
, replace ( '$ (vDateNField)={ ">=@ (=num (YearStart (AddYears (vLFLCurrEndDate, -
2)))) <=@ (=Floor (YearEnd (AddYears (vLFLCurrEndDate, -2))))" }', '@', '$') as _
PrevPreviousYearFull
, replace ( '$ (CAL_IGNORE); $ ( vYearField )={ @ (=Max ( {<$ (vCurrStr), FP={1},
$_CAL_IGNORE_>} $ ( vYearField ))) }; $ ( vDateNField
)={ "<=@ (=num (vLFLCurrEndDate))" }', '@', '$') as _ CurrentMax
, replace ( '$ (CAL_IGNORE); $ ( vYearField )={ @ (=Max ( {<$ (vPrevYearOnlyStr),
$_CAL_IGNORE_>} $ ( vYearField ))) }; $ ( vDateNField )={ "<=
@ (=num (AddYears (vLFLCurrEndDate, -1)))" }', '@', '$') as _ PreviousMax

```

```

, replace ('$(CAL_IGNORE); $( vYearField )={@ (=Max ({<$ (vPrevStr),
$_CAL_IGNORE_> } $( vYearField ))})', '@', '$') as _ PreviousPeriodMax
,5 as _ Order
, ' fldMonth ' as _ LFLSubperiod
, ' Month ' as _ LFLSubperiodTitle
Autogenerate 1;
Concatenate ( LFLPeriods )
LOAD
' fldYearQuarter ' as _ LFLId
, ' Quarter ' as _ LFLPeriodId
, ' QuarterStart ( vLFLCurrEndDate )' as _ LFLPeriodStartField
, replace ('$(vDateNField)={">=@ (=num(QuarterStart (vLFLCurrEndDate)))<=@ (=num(vLFLCurrEndDate))"}', '@', '$') as _ Current
, replace ('$(vDateNField)={">=@ (=num(QuarterStart (vLFLCurrEndDate)))<=@ (=Floor(QuarterEnd(vLFLCurrEndDate)))"}', '@', '$') as _ CurrentFull
, replace ('$(vDateNField)={">=@ (=num(QuarterStart (AddMonths (vLFLCurrEndDate,-3))))<=@ (=num(AddMonths (vLFLCurrEndDate,-3)))"}', '@', '$') as _ Previous
, replace ('$(vDateNField)={">=@ (=num(QuarterStart (AddMonths (vLFLCurrEndDate,-3))))<=@ (=Floor(QuarterEnd (AddMonths (vLFLCurrEndDate,-3))))"}', '@', '$')
as _ PreviousFull
, replace ('$(vDateNField)={">=@ (=num(QuarterStart (AddMonths (vLFLCurrEndDate,-6))))<=@ (=Floor(QuarterEnd (AddMonths (vLFLCurrEndDate,-6))))"}', '@', '$')
as _ PrevPreviousFull
, replace ('$(vDateNField)={">=@ (=num(QuarterStart (AddYears (vLFLCurrEndDate,-1))))<=@ (=num(AddYears (vLFLCurrEndDate,-1)))"}', '@', '$') as _ PreviousYear
, replace ('$(vDateNField)={">=@ (=num(QuarterStart (AddYears (vLFLCurrEndDate,-1))))<=@ (=Floor(QuarterEnd (AddYears (vLFLCurrEndDate,-1))))"}', '@', '$') as
_ PreviousYearFull
, replace ('$(vDateNField)={">=@ (=num(QuarterStart (AddYears (AddMonths (vLFLCurrEnd
ndDate,-3), -1))))<=@ (=Floor(QuarterEnd (AddYears (AddMonths (vLFLCurr) ,-3), -
1))))"}', '@', '$') as _ PrevPreviousYearFull
, replace ('$(CAL_IGNORE); $( vYearQuarterNField )={@ (=Max ({<$ (vCurrStr),
FP={1}, $_CAL_IGNORE_>> } $( vYearQuarterNField ))}); $( vDateNField
)={"<=@ (=num(vLFLCurrEndDate))"}', '@', '$') as _ CurrentMax
, replace ('$(CAL_IGNORE); $( vYearQuarterNField
)={@ (=Max ({<$ (vPrevYearOnlyStr), $_CAL_IGNORE_> } $( vYearQuarterNField
))}) $( vDateNField )={ @ (=num(AddYears (vLFLCurrEndDate,-1)))"}', '@', '$')
as _ PreviousMax
, replace ('$(CAL_IGNORE); $( vYearQuarterNField )={@ (=Max ({<$ (vPrevStr),
$_CAL_IGNORE_> } $( vYearQuarterNField ))})', '@', '$') as _
PreviousPeriodMax
,10 as _ Order
, ' fldWeek ' as _ LFLSubperiod
, ' Week ' as _ LFLSubperiodTitle
Autogenerate 1;
Concatenate ( LFLPeriods )
LOAD
' fldYearMonth ' as _ LFLId
, ' Month ' as _ LFLPeriodId
, ' MonthStart ( vLFLCurrEndDate )' as _ LFLPeriodStartField
, replace ('$(vDateNField)={">=@ (=num(MonthStart (vLFLCurrEndDate)))<=@ (=num(vLFLCurrEndDate))"}', '@', '$') as _ Current
, replace ('$(vDateNField)={">=@ (=num(MonthStart (vLFLCurrEndDate)))<=@ (=Floor(MonthEnd(vLFLCurrEndDate)))"}', '@', '$') as _ CurrentFull
, replace ('$(vDateNField)={">=@ (=num(MonthStart (AddMonths (vLFLCurrEndDate,-1))))<=@ (=num(AddMonths (vLFLCurrEndDate,-1)))"}', '@', '$') as _ Previous
, replace ('$(vDateNField)={">=@ (=num(MonthStart (AddMonths (vLFLCurrEndDate,-1))))<=@ (=Floor(MonthEnd (AddMonths (vLFLCurrEndDate,-1))))"}', '@', '$') as
_ PreviousFull
, replace ('$(vDateNField)={">=@ (=num(MonthStart (AddMonths (vLFLCurrEndDate,-2))))<=@ (=Floor(MonthEnd (AddMonths (vLFLCurrEndDate,-2))))"}', '@', '$') as
_ PrevPreviousFull
, replace ('$(vDateNField)={">=@ (=num(MonthStart (AddYears (vLFLCurrEndDate,-1))))<=@ (=num(AddYears (vLFLCurrEndDate,-1)))"}', '@', '$') as _ PreviousYear

```

```

, replace ('$(vDateNField)={">=@ (=num (MonthStart (AddYears (vLFLCurrEndDate, -
1))))<=@ (=Floor (MonthEnd (AddYears (vLFLCurrEndDate, -1))))" }, '@', '$') as
_ PreviousYearFull
, replace ('$(vDateNField)={">=@ (=num (MonthStart (AddYears (AddMonths (vLFLCurrEnd
Date, -1), -1))))<=@ (=Floor (MonthEnd (AddYears (AddMonths (vLFLCurrEnd
Date, -1), -1))))" }, '@', '$') as _ PrevPreviousYearFull
, replace ('$(CAL_IGNORE); $( vYearMonthNField )={@ (=Max ({<$ (vCurrStr),
FP={1}, $_CAL_IGNORE_>>} $( vYearMonthNField )))); $( vDateNField
)={"<=@ (=num (vLFLCurrEndDate))" }, '@', '$') as _ CurrentMax
, replace ('$(CAL_IGNORE); $( vYearMonthNField
)={@ (=Max ({<$ (vPrevYearOnlyStr), $_CAL_IGNORE_>} $( vYearMonthNField ))));
$( vDateNField )={" @ (=num (AddYears (vLFLCurrEndDate, -1)))" }, '@', '$') as _
PreviousMax
, replace ('$(CAL_IGNORE); $( vYearMonthNField )={@ (=Max ({<$ (vPrevStr),
$_CAL_IGNORE_>} $( vYearMonthNField ))}}', '@', '$') as _
PreviousPeriodMax
, 20 as _ Order
, ' fldDay ' as _ LFLSubperiod
, ' Day ' as _ LFLSubperiodTitle
Autogenerate 1;
Concatenate ( LFLPeriods )
LOAD
' fldYearWeek ' as _ LFLId
, ' Week ' as _ LFLPeriodId
, ' WeekStart ( vLFLCurrEndDate )' as _ LFLPeriodStartField
, replace ('$(vDateNField)={">=@ (=num (WeekStart (vLFLCurrEndDate))<=@ (=num (vLFL
CurrEndDate))" }, '@', '$') as _ Current
, replace ('$(vDateNField)={">=@ (=num (WeekStart (vLFLCurrEndDate))<=@ (=Floor (We
ekEnd (vLFLCurrEndDate)))" }, '@', '$') as _ CurrentFull
, replace ('$(vDateNField)={">=@ (=num (WeekStart (vLFLCurrEndDate-
7)))<=@ (=num (vLFLCurrEndDate-7))" }, '@', '$') as _ Previous
, replace ('$(vDateNField)={">=@ (=num (WeekStart (vLFLCurrEndDate-
7)))<=@ (=Floor (WeekEnd (vLFLCurrEndDate-7)))" }, '@', '$') as _ PreviousFull
, replace ('$(vDateNField)={">=@ (=num (WeekStart (vLFLCurrEndDate-
14)))<=@ (=Floor (WeekEnd (vLFLCurrEndDate-14)))" }, '@', '$') as _
PrevPreviousFull
, replace ('$(vDateNField)={">=@ (=num (MakeWeekDate (Year (vLFLCurrEndDate) -1,
Week ( vLFLCurrEndDate ), 0)))== (=num (MakeWeekDate (Year (vLFLC) 1, Week (
vLFLCurrEndDate ), WeekDay ( vLFLCurrEndDate )))}}', '@', '$') as _
PreviousYear
, replace ('$(vDateNField)={">=@ (=num (MakeWeekDate (Year (vLFLCurrEndDate) -1,
Week ( vLFLCurrEndDate ), 0)))== (=Floor (MakeWeekDate (Year (vLFL) 1, Week (
vLFLCurrEndDate ), 6)))" }, '@', '$') as _ PreviousYearFull
, replace ('$(vDateNField)={">=@ (=num (MakeWeekDate (Year (vLFLCurrEndDate-7) -1,
Week (vLFLCurrEndDate-7), 0))<=@ (=Floor (MakeWeekDate (Year (vLFLCurrEndDate-
7) -1, Week ( vLFLCurrEndDate - 7), 6)))" }, '@', '$') as _
PrevPreviousYearFull
, replace ('$(CAL_IGNORE); $( vYearWeekNField )={@ (=Max ({<$ (vCurrStr),
FP={1}, $_CAL_IGNORE_>>} $( vYearWeekNField )))); $( vDateNField
)={"<=@ (=num (vLFLCurrEndDate))" }, '@', '$') as _ CurrentMax
, replace ('$(CAL_IGNORE); $( vYearWeekNField
)={@ (=Max ({<$ (vPrevYearOnlyStr), $_CAL_IGNORE_>} $( vYearWeekNField ))));
$( vDateNField )={" @ (=num (AddYears (vLFLCurrEndDate, -1)))" }, '@', '$') as _
PreviousMax
, replace ('$(CAL_IGNORE); $( vYearWeekNField )={@ (=Max ({<$ (vPrevStr),
$_CAL_IGNORE_>} $( vYearWeekNField ))}}', '@', '$') as _ PreviousPeriodMax
, 30 as _ Order
, ' fldDay ' as _ LFLSubperiod
, ' Day ' as _ LFLSubperiodTitle
Autogenerate 1;
Concatenate ( LFLPeriods )
LOAD
' fldDate ' as _ LFLId
, ' Day ' as _ LFLPeriodId

```

```

, ' Date ( vLFLCurrEndDate )' as _ LFLPeriodStartField
, replace ('$( vDateNField )={@ (=Num(vLFLCurrEndDate))}', '@', '$') as _
Current
, replace ('$( vDateNField )={@ (=Num(vLFLCurrEndDate))}', '@', '$') as _
CurrentFull
, replace ('$( vDateNField )={@ (=Num(vLFLCurrEndDate-1))}', '@', '$') as _
Previous
, replace ('$( vDateNField )={@ (=Num(vLFLCurrEndDate-1))}', '@', '$') as _
PreviousFull
, replace ('$( vDateNField )={@ (=Num(vLFLCurrEndDate-2))}', '@', '$') as _
PrevPreviousFull
, replace ('$( vDateNField )={@ (=Num(AddYears (vLFLCurrEndDate, -1)))}', '@',
'$') as _ PreviousYear
, replace ('$( vDateNField )={@ (=Num(AddYears (vLFLCurrEndDate, -1)))}', '@',
'$') as _ PreviousYearFull
, replace ('$( vDateNField )={@ (=Num(AddYears (vLFLCurrEndDate-1, -1)))}',
 '@', '$') as _ PrevPreviousYearFull
, replace ('$(CAL_IGNORE); $( vDateNField )={@ (=Max({<$ (vCurrStr), FP={1},
$_CAL_IGNORE_>} $( vDateNField )))}; $( vDateNField
)={"<=@ (=num(vLFLCurrEndDate))"}', '@', '$') as _ CurrentMax
, replace ('$(CAL_IGNORE); $( vDateNField )={@ (=Max({<$ (vPrevYearOnlyStr),
$_CAL_IGNORE_>} $( vDateNField )))}; $( vDateNField )={"<=
@ (=num(AddYears(vLFLCurrEndDate,-1)))"}', '@', '$') as _ PreviousMax
, replace ('$(CAL_IGNORE); $( vDateNField )={@ (=Max({<$ (vPrevStr),
$_CAL_IGNORE_>} $( vDateNField )))}', '@', '$') as _ PreviousPeriodMax
, 40 as _ Order
, ' fldDay ' as _ LFLSubperiod
, ' Day ' as _ LFLSubperiodTitle
Autogenerate 1;
LEFT JOIN ( LFLPeriods )
LOAD _ Language , id as _ LFLId , _ Translation as _ LFLLabelField RESIDENT
Translations ;
LEFT JOIN ( LFLPeriods )
LOAD _ Language , id as _ LFLPeriodId , _ Translation as _ LFLPeriod RESIDENT
Translations ;
Параметри Інфляції :
LOAD * INLINE [
_ InflId , FI
ConsiderInflation , 1
DontConsiderInflation , 0
];
LEFT JOIN ( Параметри Інфляції )
LOAD _ Language , id as _ InflId , _ Translation as _ ПараметрІнфляції
RESIDENT Translations ;
SET vIsInflation = _ InflId = ' ConsiderInflation ';
LET vInflPerStart =
replace ('Num (MonthStart (AddMonths (Max ({<$ (vPrevStr), $_CAL_IGNORE_>} $(
vDateNField ), 1)))', '@', '$');
LET vInflPerEnd = replace (' Num ( Max ( {<$ ( vCurrStr ), $_CAL_IGNORE_>} $(
vDateNField )))', '@', '$');
LET vInflPeriod =
replace ('$(vDateNField)={">=@ (=@(vInflPerStart))<=@ (=@(vInflPerEnd))"}', '@',
'$');
LET vKI = replace ('( exp (SUM({<@(vInflPeriod)>}) log ($ ( vInflCoefField
))))-1)', '@', '$');
LET vFI = replace (' Max (FI) * if (@(vInflPerStart)<=@(vInflPerEnd), 1, 0)',
 '@', '$');
SET _Валюта = 'EUR'; // Валюта за замовчуванням
LET vCardExpr = replace ('@( vSales )', '@', '$');
LET vCardExprTitle = replace ('@(T( Sales ))', '@', '$');
SET vLFLMoneyFormat = '###0;-###0;0';
SET vLFLDiffMoneyFormat = '+# ##0;-# ##0;0';
SET vLFLCountFormat = '###0;-###0;0';
SET vLFLAvgFormat = '###0,0;-###0,0;0,0';

```

```

SET vLFLDiffAvgFormat = '+# ##0,0;-# ##0,0;0,0';
SET vLFLDiffCountFormat = '+# ##0;-# ##0;0';
SET vPlan%Format = '###0%;-###0%';
SET vGrowth%Format = '▲# ##0%;▼# ##0%;0%';//'+###0%;-###0%';
SET vPlanForecastFormat = '###0,0%;-###0,0%'; //→
LET vGoodKPIColor = rgb (39, 110, 39); // RGB (29, 128, 15); // RGB (98, 172, 30); // rgb (0,128,0); // RGB (102,153,51); // RGB (99, 172, 31); // Green ();
LET vBadKPIColor = rgb (249, 63, 23); // rgb (255,0,0); // LightRed ();
LET vMiddleKPIColor = rgb (251, 154, 41); // RGB (255, 255, 0);
LET vMiddle2KPIColor = rgb (236, 236, 0);
LET vKPIColor = rgb (150,150,150); // DarkGray ();

// LET vPlanBackColor = rgb (236, 238, 240);
LET vPlanColor = rgb (109, 128, 150);
LET vKPINameColor = rgb (109, 128, 150);
LET vCompColor = rgb (57, 90, 128);
LET vTitleColor = rgb (91,91,91);
LET vFontColor = rgb (59,59,59);

// Кольори для діапазонів
SET vColorRanges =
'0 <= c < 0,1',
'0,1 <= c < 0,2',
'0,2 <= c < 0,3',
'0,3 <= c < 0,4',
'0,4 <= c < 0,5',
'0,5 <= c < 0,6',
'0,6 <= c < 0,7',
'0,7 <= c < 0,8',
'0,8 <= c < 0,9',
'0,9 <= c < 1',
'1 <= c < 1,1',
'1,1 <= c < 1,2';
SET vColorsForRanges =
/* Red */
rgb (224, 0, 0),
rgb (232, 0, 0),
rgb (240, 0, 0),
rgb (248, 0, 0),
rgb (255, 0, 0),
rgb (255, 51, 0),
rgb (255, 102, 0),
rgb (255, 153, 0),
rgb (255, 204, 0),
/* Yellow */
rgb (255, 255, 0),
/* Green */
rgb (0, 255, 0),
rgb (51, 204, 0);

Ranges :
LOAD * INLINE [
Range
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

```

```

1.0
1.1
];

// Колірна схема
SET vColor = pick ( rowno (), '#9999CC', '#6666FF', '#009999', '#66CC99',
'#6633FF', '#6600CC', '#330099', '#663399 #009966', '#339966', '#009933',
'#006600', '#666633', '#666600', '#333300', '#FF9900', '#FF9933', '# # 996633',
'# 663366', '# 330033');
SET vTopN = 10;

// Парето
SET vPareto = 0.8; // 80% значення за замовчуванням

// FMR- аналіз
SET vF = 80; //%, F -> M
SET vM = 95; //%, M -> R
LET vFmrMsg = replace ('@ (=@(T(FMRSelectionsButtons)))', '@', '$');

// RFM- аналіз
LET vRDays = ' round (('&$_( _DAYS_COUNT_CALENDAR_)&') / 5)'; // ' round ( ( $ (
vDaysCount (FP={1})) ) / 5)'
/ Межі RFM (за замовчуванням )

// Кордони R (в днях):
LET R5 = 0; // починаючи із зазначеної кількості днів - R5
LET R4 = 31; // - R4
LET R3 = 91; // - R3
LET R2 = 181; // - R2
LET R1 = 365; // - R1

// Кордони F (за кількістю замовлень ):
LET F5 = 200; // [F5 .. і більше )
LET F4 = 190; // [F4.. F5)
LET F3 = 150; // [F3.. F4)
LET F2 = 50; // [F2.. F3)
LET F1 = 0; // [F1.. F2)

// Межі M (за сумою продажу):
LET M5 = 15000000; // [M5 .. і більше )
LET M4 = 10000000; // [M4.. M5)
LET M3 = 8000000; // [M3.. M4)
LET M2 = 5000000; // [M2.. M3)
LET M1 = 0; // [M1.. M2)

_ RSwitch :
LOAD * INLINE [
_ RMode
Автоматично
Вручну
];
_ FSwitch :
LOAD * INLINE [
_ FMode
Автоматично
Вручну
];
_ MSwitch :
LOAD * INLINE [
_ MMode
Автоматично
Вручну
];
SET vABCType = ' FirstSortedValue ( _ ABCExp , _ ABCTypePK )';
SET vABCTypeTotal = ' FirstSortedValue ( _ ABCExpTotal , _ ABCTypePK )';

```

```

SET vABCLabel = ' FirstSortedValue ( _ МіткаABC , _ ABCTypePK )';
SET vXYZType = ' FirstSortedValue ( _ XYZField , _ XYZTypePK )';
SET vXYZLabel = ' FirstSortedValue ( _ МіткаXYZ , _ XYZTypePK )';
SET vClassA = 50;
SET vClassB = 80;
SET vClassX = 70;
SET vClassY = 85;
LET vAColor = vGoodKPIColor ;
LET vBColor = vMiddleKPIColor ;
LET vCColor = 'RGB(227, 219, 154)';
LET _ABC =
replace ('if (rangesum (above ([@ (= @ (T (CumulativePercent)))))) < @ (vClassA) / 100
, 'A' ,
      if (rangesum (above ([@ (= @ (T (CumulativePercent)))))) < @ (vClassB) / 100 , 'B'
,
      'C'
      )))
, '@' , '$');
LET _XYZ =
replace ('if (fabs ([@ (= @ (T (VariationCoefficient)))))) < @ (vClassX) / 100 , 'X' ,
      if (fabs ([@ (= @ (T (VariationCoefficient)))))) < @ (vClassY) / 100 , 'Y' , 'Z'))
, '@' , '$');
LET _RankX = replace ( ' Num ( rank ( if ((@ (_XYZ)) = 'X' , -
[ @ (= @ (T (VariationCoefficient))) ] , null () , 4 , 1 ) )' , '@' , '$');
LET _RankY = replace ( ' Num ( rank ( if ((@ (_XYZ)) = 'Y' , -
[ @ (= @ (T (VariationCoefficient))) ] , null () , 4 , 1 ) )' , '@' , '$');
LET _RankZ = replace ( ' Num ( rank ( if ((@ (_XYZ)) = 'Z' , -
[ @ (= @ (T (VariationCoefficient))) ] , null () , 4 , 1 ) )' , '@' , '$');
LET _RankXYZ = replace ( ' alt (@ (_RankX) , @ (_RankY) , @ (_RankZ))' , '@' , '$');
LET vMsg = replace ('@ (= @ (T (ABCSelectionsButtons)))' , '@' , '$'); //
Виміри:
LOAD * INLINE [
_ ПолеВиміриId , _ МіткаВиміриId ,
fldProductHierarchy1 , ProductHierarchy1
fldProductHierarchy2 , ProductHierarchy2
fldProductHierarchy3 , ProductHierarchy3
fldClientCategory , ClientCategory
fldClientGroup , ClientGroup
fldClient , Client
fldManager , Manager
fldSupplier , Supplier
fldProductKind , ProductKind
fldProductCategory , ProductCategory
fldNomenclatureGroup , NomenclatureGroup
fldProductGroup , ProductGroup
fldProductCountry , ProductCountry
fldProduct , Product
fldProduct_ID , Product_ID
fldSaleType , SaleType
fldTransactionType , TransactionType
fldDocument , Document
fldYearQuarter , YearQuarter
fldYearMonth , YearMonth
fldYear , Year
fldQuarter , Quarter
fldMonth , Month
fldDate , Date
];
LEFT JOIN (Виміри)
LOAD _ Language , id as _ ПолеВиміриId , _ Translation as _ Вимір RESIDENT
Translations ;
LEFT JOIN (Виміри)
LOAD _ Language , id as _ МіткаВиміриId , _ Translation as _ МіткаВиміри
RESIDENT Translations ;
FOR i = 1 TO 3

```

```

Виміри$(i):
  NoConcatenate
LOAD
  _ Language
, AutoNumber ( _ ПолеВиміриId , ' _ ПолеВиміриId ' ) as [ _ Виміри$(i)№ ]
, _ Виміри as [ _ Виміри$(i) ]
, _ МіткаВиміри as [ _ МіткаВиміри $(i) ]
, _ ПолеВиміриId as [ _ ПолеВиміри $(i) Id ]
RESIDENT Виміри;
NEXT
Вирази:
LOAD
*
, AutoNumber ( _ Expr , ' _ Expr ' ) as _ Вираз№
, replace ( _ Expr , '@', '$') as _ Exp
, replace ( _ ExprCmp , '@', '$') as _ ExpCmp
, ' Num (' & replace ( _ Expr , '@', '$') & ', ' & chr (39) & _ ShortFormat &
chr (39) & ')' as _ ShortFormattedExp
, ' Num (' & replace ( _ Expr , '@', '$') & ', ' & chr (39) & _ Format & chr
(39) & ')' as _ FormattedExp
, ' Num (' & replace ( _ ExprCmp , '@', '$') & ', ' & chr (39) & _ ShortFormat
& chr (39) & ')' as _ ShortFormattedExpCmp
, ' Num (' & replace ( _ ExprCmp , '@', '$') & ', ' & chr (39) & _ Format &
chr (39) & ')' as _ FormattedExpCmp
, ' Num (' & replace ( _ ExplFLCurr , '@', '$') & ', ' & chr (39) & _ Format &
chr (39) & ')' as _ FormattedExplFLCurr
, ' Num (' & replace ( _ ExplFLCurr , '@', '$') & ', ' & chr (39) & _
ShortFormat & chr (39) & ')' as _ ShortFormattedExplFLCurr
, ' Num (' & replace ( _ ExplFLPrev , '@', '$') & ', ' & chr (39) & _ Format &
chr (39) & ')' as _ FormattedExplFLPrev
, replace ( ' Dual ( Num (' & replace ( _ Expr , '@', '$') & ', ' & chr (39) &
_ Format & chr (39) & ') , ' & if ( len ( trim ( _ ExprSortRating ) ) > 0 , _
ExprSortRating , _ Expr ) & ') , '@', '$') as _ ExprSortRating
;
LOAD * INLINE [
  _ ВиразId          | _ Expr          | _ ExprSortRating
                    | _ ExprCmp      | _ ExplFLCurr
                    | _ Format        | _ ExplFLPrev
                    |                | _ ShortFormat
Sales               | @ (vSales (FP={1})) |
                    | @ (vSalesC1 (FP={1})) |
@ (vSales (@ (vCurr); @ (CAL_IGNORE))) | @ (vSales (@ (vPrev); @ (CAL_IGNORE)))
                    | # ##0,00      | # ##0
SalesVAT            | @ (vSalesVAT (FP={1})) |
                    | @ (vSalesVATC1 (FP={1})) |
@ (vSalesVAT (@ (vCurr); @ (CAL_IGNORE))) |
@ (vSalesVAT (@ (vPrev); @ (CAL_IGNORE))) | # ##0,00 | # ##0
SalesCount         | @ (vSalesCount (FP={1})) |
                    | @ (vSalesCountC1 (FP={1})) |
@ (vSalesCount (@ (vCurr); @ (CAL_IGNORE))) |
@ (vSalesCount (@ (vPrev); @ (CAL_IGNORE))) | # ##0,00 | # ##0
Returns            | @ (vReturns (FP={1})) | -@ ( vReturns (FP={1}))
                    | @ (vReturnsC1 (FP={1})) |
@ (vReturns (@ (vCurr); @ (CAL_IGNORE))) |
@ (vReturns (@ (vPrev); @ (CAL_IGNORE))) | # ##0,00 | # ##0
Discount           | @ (vSalesDiscount (FP={1})) |
                    | @ (vSalesDiscountC1 (FP={1})) |
@ (vSalesDiscount (@ (vCurr); @ (CAL_IGNORE))) |
@ (vSalesDiscount (@ (vPrev); @ (CAL_IGNORE))) | # ##0,00 | # ##0
Margin             | @ (vMargin (FP={1})) |
                    | @ (vMarginC1 (FP={1})) |
@ (vMargin (@ (vCurr); @ (CAL_IGNORE))) | @ (vMargin (@ (vPrev); @ (CAL_IGNORE)))
                    | # ##0,00      | # ##0
Cost               | @ (vCost (FP={1})) | -@ ( vCost

```

```

(FP={1})) | @ (vCostC1 (FP={1})) |
@ (vCost (@ (vCurr);@ (CAL_IGNORE))) |
@ (vCost (@ (vPrev);@ (CAL_IGNORE))) | # ##0,00 | # ##0
Margin % | @ (vMargin% (FP={1})) |
| @ (vMarginC1% (FP={1})) |
@ (vMargin% (@ (vCurr);@ (CAL_IGNORE))) |
@ (vMargin% (@ (vPrev);@ (CAL_IGNORE))) | # ##0,00% | #
##0,00%
Markup % | @ (vMarkup% (FP={1})) |
| @ (vMarkupC1% (FP={1})) |
@ (vMarkup% (@ (vCurr);@ (CAL_IGNORE))) |
@ (vMarkup% (@ (vPrev);@ (CAL_IGNORE))) | # ##0,00% | #
##0,00%
Discount % | @ (vSalesDiscount% (FP={1})) |
| @ (vSalesDiscountC1% (FP={1})) |
@ (vSalesDiscount% (@ (vCurr);@ (CAL_IGNORE))) |
@ (vSalesDiscount% (@ (vPrev);@ (CAL_IGNORE))) | # ##0,00% | # ##0,00%
SalesS | @ (vSalesS (FP={1})) |
| @ (vSalesSC1 (FP={1})) |
@ (vSalesS (@ (vCurr);@ (CAL_IGNORE))) | @ (vSalesS (@ (vPrev);@ (CAL_IGNORE)))
| # ##0,00 | # ##0
SalesVATS | @ (vSalesVATS (FP={1})) |
| @ (vSalesVATSC1 (FP={1})) |
@ (vSalesVATS (@ (vCurr);@ (CAL_IGNORE))) | @ (vSalesVATS (@ (vPrev);@ (CAL_IGNORE)))
| # ##0,00 | # ##0
SalesCounts | @ (vSalesCountS (FP={1})) |
| @ (vSalesCountSC1 (FP={1})) |
@ (vSalesCountS (@ (vCurr);@ (CAL_IGNORE))) |
@ (vSalesCountS (@ (vPrev);@ (CAL_IGNORE))) | # ##0,00 | # ##0
DiscountS | @ (vSalesDiscountS (FP={1})) |
| @ (vSalesDiscountSC1 (FP={1})) |
@ (vSalesDiscountS (@ (vCurr);@ (CAL_IGNORE))) |
@ (vSalesDiscountS (@ (vPrev);@ (CAL_IGNORE))) | # ##0,00 | # ##0
Discount%S | @ (vSalesDiscount%S (FP={1})) |
| @ (vSalesDiscountSC1% (FP={1})) |
@ (vSalesDiscount%S (@ (vCurr);@ (CAL_IGNORE))) |
@ (vSalesDiscount%S (@ (vPrev);@ (CAL_IGNORE))) | # ##0,00% | # ##0,00%
SKUCount | @ (vSKUs (FP={1})) |
| @ (vSKUsC1 (FP={1})) |
@ (vSKUs (@ (vCurr);@ (CAL_IGNORE))) |
@ (vSKUs (@ (vPrev);@ (CAL_IGNORE))) | # ##0 | # ##0
ClientsCount | @ (vClientsCount (FP={1})) |
| @ (vClientsCountC1 (FP={1})) |
@ (vClientsCount (@ (vCurr);@ (CAL_IGNORE))) |
@ (vClientsCount (@ (vPrev);@ (CAL_IGNORE))) | # ##0 | # ##0
AverageOrder | @ (vAvgSalesOrder (FP={1})) |
| @ (vAvgSalesOrderC1 (FP={1})) |
@ (vAvgSalesOrder (@ (vCurr);@ (CAL_IGNORE))) |
@ (vAvgSalesOrder (@ (vPrev);@ (CAL_IGNORE))) | # ##0,00 | # ##0
OrdersCount | @ (vOrdersCount (FP={1})) |
| @ (vOrdersCountC1 (FP={1})) |
@ (vOrdersCount (@ (vCurr);@ (CAL_IGNORE))) |
@ (vOrdersCount (@ (vPrev);@ (CAL_IGNORE))) | # ##0 | # ##0
] ( delimiter is '|');

// Orders % | @ (vOrderSales% (FP={1})) |
| @ (vOrderSalesC1% (FP={1})) |
@ (vOrderSales% (@ (vCurr);@ (CAL_IGNORE))) |
@ (vOrderSales% (@ (vPrev);@ (CAL_IGNORE))) | # ##0,00% | # ##0,00%

// OrdersCount % | @ (vOrderCount% (FP={1})) |
| @ (vOrderCountC1% (FP={1})) | @ (vSKUs (@ (vCurr);@ (CAL_IGNORE)))
| @ (vSKUs (@ (vPrev);@ (CAL_IGNORE))) | # ##0,00%

```

```

| # ##0,00%

// Plan % | @(vSalesPlan%(FP={1})) |
@(vSalesPlanC1%(FP={1})) | @(vSalesPlan%(@ (vCurr);@(CAL_IGNORE))) |
@(vSalesPlan%(@ (vPrev);@(CAL_IGNORE))) | # ##0,0% | # ##0,0%
LEFT JOIN ( Вирази )
LOAD _ Language , id as _ ВиразId , _ Translation as _Вираз RESIDENT
Translations ;
SET _Вираз = ' FirstSortedValue ( _Вираз, _Вираз№)';
SET _ FormatedExp = ' FirstSortedValue ( _ FormatedExp , _Вираз№)';
SET _ ShortFormatedExp = ' FirstSortedValue ( _ ShortFormatedExp , _Вираз№)';
SET _ FormatedExpCmp = ' FirstSortedValue ( _ FormatedExpCmp , _Вираз№)';
SET _ ShortFormatedExpCmp = ' FirstSortedValue ( _ ShortFormatedExpCmp ,
_Вираз№)';
SET _ FormatedExpLFLCurr = ' FirstSortedValue ( _ FormatedExpLFLCurr ,
_Вираз№)';
SET _ ShortFormatedExpLFLCurr = ' FirstSortedValue ( _ ShortFormatedExpLFLCurr
, _Вираз№)';
SET _ FormatedExpLFLPrev = ' FirstSortedValue ( _ FormatedExpLFLPrev ,
_Вираз№)';
SET _ ExpSortRating = ' FirstSortedValue ( _ ExpSortRating , _Вираз№)';
FOR i = 1 TO 3
SET _Вираз$(i) = ' FirstSortedValue ( _Вираз$(i), _Вираз№$(i))';
SET _ FormatedExp $(i) = ' FirstSortedValue ( _ FormatedExp $(i),
_Вираз№$(i))';
SET _ ShortFormatedExp $(i) = ' FirstSortedValue ( _ ShortFormatedExp $(i),
_Вираз№$(i))';
SET _ FormatedExpCmp $(i) = ' FirstSortedValue ( _ FormatedExpCmp $(i),
_Вираз№$(i))';
SET _ ShortFormatedExpCmp $(i) = ' FirstSortedValue ( _ ShortFormatedExpCmp
$(i), _Вираз№$(i))';
SET _ FormatedExpLFLCurr $(i) = ' FirstSortedValue ( _ FormatedExpLFLCurr
$(i), _Вираз№$(i))';
SET _ ShortFormatedExpLFLCurr $(i) = ' FirstSortedValue ( _
ShortFormatedExpLFLCurr $(i), _Вираз№$(i))';
SET _ FormatedExpLFLPrev $(i) = ' FirstSortedValue ( _ FormatedExpLFLPrev
$(i), _Вираз№$(i))';
SET _ Format $(i) = ' FirstSortedValue ( _ Format $(i), _Вираз№$(i))';
SET _ ExpSortRating $(i) = ' FirstSortedValue ( _ ExpSortRating $(i),
_Вираз№$(i))';
Вирази $(i):
LOAD _ Language
, _ ВиразId as [ _Вираз$(i) Id ]
, _Вираз№ as [ _Вираз№$(i) ]
, _Вираз as [ _Вираз$(i) ]
, _ Exp as [ _ Exp $(i) ]
, _ ExpCmp as [ _ ExpCmp $(i) ]
, _ Format as [ _ Format $(i) ]
, _ ShortFormatedExp as [ _ ShortFormatedExp $(i) ]
, _ FormatedExp as [ _ FormatedExp $(i) ]
, _ ShortFormatedExpCmp as [ _ ShortFormatedExpCmp $(i) ]
, _ FormatedExpCmp as [ _ FormatedExpCmp $(i) ]
, _ FormatedExpLFLCurr as [ _ FormatedExpLFLCurr $(i) ]
, _ FormatedExpLFLPrev as [ _ FormatedExpLFLPrev $(i) ]
, _ ExpSortRating as [ _ ExpSortRating $(i) ]
RESIDENT Вирази ;
NEXT
ВиміриЗамовлення :
LOAD * INLINE [
_ ПолеВиміриЗамовленняId , _ МіткаВиміриЗамовленняId , _№_
fldProductHierarchy1, ProductHierarchy1, 1
fldProductHierarchy2, ProductHierarchy2, 2
fldProductHierarchy3, ProductHierarchy3, 3
fldClientCategory , ClientCategory , 4

```

```

fldClientGroup ,      ClientGroup , 5
fldClient ,          Client , 6
fldManager ,        Manager,7
fldSupplier ,       Supplier , 8
fldProductKind ,   ProductKind , 9
fldProductCategory , ProductCategory , 10
fldNomenclatureGroup , NomenclatureGroup , 11
fldProductGroup ,   ProductGroup , 12
fldProductCountry , ProductCountry , 13
fldProduct ,        Product , 14
fldProduct_ID ,     Product_ID , 15
fldOrderNumber ,    OrderNumber , 16
fldOrderDate ,      OrderDate , 17
fldDocument ,       Document , 18
fldYearQuarter ,    YearQuarter , 19
fldYearMonth ,      YearMonth , 20
fldYear ,           Year , 21
fldQuarter ,        Quarter , 22
fldMonth ,          Month , 23
fldDate ,           Date , 24
];
LEFT JOIN ( ВиміриЗамовлення )
LOAD _ Language , id as _ ПолеВиміриЗамовленняId , _ Translation as _
ВимірЗамовлення RESIDENT Translations ;
LEFT JOIN ( ВиміриЗамовлення )
LOAD _ Language , id as _ МіткаВиміриЗамовленняId , _ Translation as _
МіткаВиміриЗамовлення RESIDENT Translations ;
ВидЗамовлень :
LOAD * INLINE [
  _ OrderTypeId , _ OrderType , _ vOrdSales ,
vOrdSales % , _ vOrdSalesCount , _ vOrdCount % , _ vOrdCount
0 ,
  vOrderSales % , vOrderSalesCount , vOrderCount % , vOrdersCount
1 ,
  OrdersWithSales , vOrderSalesWS , vOrderSalesWS
% , vOrderSalesCountWS , vOrderCountWS % , vOrdersCountWS
];
LEFT JOIN ( ВидЗамовлень )
LOAD _ Language , id as _ OrderType , _ Translation as _ ВидЗамовлень
RESIDENT Translations ;
Календар Група :
LOAD * INLINE [
  _ КалендарГрупаId , _ КалендарГрупаМіткаId
fldYear , Year
fldYearQuarter , YearQuarter
fldYearMonth , YearMonth
fldYearWeek , YearWeek
fldDate , Date
];
LEFT JOIN ( КалендарГрупа )
LOAD _ Language , id as _ КалендарГрупаId , _ Translation as _ КалендарГрупа
RESIDENT Translations ;
LEFT JOIN ( КалендарГрупа )
LOAD _ Language , id as _ КалендарГрупаМіткаId , _ Translation as _
КалендарГрупаМітка RESIDENT Translations ;

// Змінні для аналізу множин
LET saCalGroupMax =
replace (' [ @ ( = _КалендарГрупа ) ] = { " @ ( =Only ( { < [ @ ( = _КалендарГрупа ) ] ] = { " = [ @ ( = _КалендарГрупа ) ] = ' ' @ ( =Max ( { < @ ( CP ) > } [ @ ( = _КалендарГрупа ) ] ) ) ' ' " } > } [ @ ( = _КалендарГрупа ) ] ) ) " ' , ' @ ' , ' $ ' ) ;
LET saCalGroupMax2 =
replace (' [ @ ( = _КалендарГрупа ) ] = { " @ ( =Only ( { < [ @ ( = _КалендарГрупа ) ] ] = { " = [ @ ( = _КалендарГрупа ) ] = ' ' @ ( =Max ( { < @ ( CP ) > } [ @ ( = _КалендарГрупа ) ] , 2 ) ) ' ' " } > } [ @ ( = _КалендарГрупа ) ] ) ) " ' , ' @ ' , ' $ ' ) ;

```

```

// _ ПолеВиміриОглядId - Always One Selected
ВиміриОгляд :
LOAD
    AutoNumber ( _ ПолеВиміриОглядId , '_ ПолеВиміриОглядId ') as _
ВиміриОгляд №, *
INLINE [
    _ ПолеВиміриОглядId , _ МіткаВиміриОглядId ,
fldProductHierarchy1, ProductHierarchy1
fldProductHierarchy2, ProductHierarchy2
fldProductHierarchy3, ProductHierarchy3
fldClientCategory , ClientCategory
fldClientGroup , ClientGroup
fldClient , Client
fldManager , Manager
fldSupplier , Supplier
fldProductKind , ProductKind
fldProductCategory , ProductCategory
fldNomenclatureGroup , NomenclatureGroup
fldProductGroup , ProductGroup
fldProductCountry , ProductCountry
fldProduct , Product
];
LEFT JOIN ( ВиміриОгляд )
LOAD _ Language , id as _ ПолеВиміриОглядId , _ Translation as _ ВиміриОгляд
RESIDENT Translations ;
LEFT JOIN ( ВиміриОгляд )
LOAD _ Language , id as _ МіткаВиміриОглядId , _ Translation as _
МіткаВиміриОгляд RESIDENT Translations ;

// _КалендарГрупа2Key - Always One Selected
КалендарГрупа2:
LOAD AutoNumber ( _КалендарГрупа2Key, 'КалендарГрупа2') as _КалендарГрупа2PK,
_КалендарГрупа2Key INLINE [
_КалендарГрупа2Key
Year
Quarter
Month
Week
];
LEFT JOIN (КалендарГрупа2)
LOAD _ Language , id as _КалендарГрупа2Key, _ Translation as
_ПолеКалендарГрупа2 RESIDENT Translations ;
SET _КалендарГрупа2Вибраний = '_КалендарГрупа2Key = '$1'';
SET _КалендарГрупа2НеВибраний = '_КалендарГрупа2Key <> '$1'';
Показники Огляд :
LOAD * INLINE [
_ ПоказникОглядKey
Sales
Margin
Orders
Clients
SKUCount
SalesCount
Cost
Discount
SalesVAT
];
LEFT JOIN ( Показники Огляд )
LOAD _ Language , id as _ ПоказникОглядKey , _ Translation as _ ПоказникОгляд
RESIDENT Translations ;
SET _ ПоказникОглядВибраний = '|' & Concat ( distinct _ ПоказникОглядKey ,
'|') & '|' like '*|$1|*';
LET clrHeatmap = replace ('Colormix2( if ($1>0, rangemin ($1, 1), rangemax (-

```

```

1, $1))), rgb (243, 88, 81), rgb (84, 180, 104), rgb ( 255, 255, 255))', '@',
'$');
LET clrHeatmapReversed = replace ('Colormix2( if ($1>0, rangemin ($1, 1),
rangemax (-1, $1))), rgb (84, 180, 104), rgb ( 243, 88, 81), rg 255, 255,
255))', '@', '$');
ВиміриРейтинги :
LOAD
    AutoNumber ( _ ПолеВиміриРейтингиId , '_ ПолеВиміриРейтингиId ') as
    _ВимірРейтинги № , *
    INLINE [
        _ ПолеВиміриРейтингиId , _ МіткаВиміриРейтингиId ,
        fldProductHierarchy1, ProductHierarchy1
        fldProductHierarchy2, ProductHierarchy2
        fldProductHierarchy3, ProductHierarchy3
        fldClientCategory , ClientCategory
        fldClientGroup , ClientGroup
        fldClient , Client
        fldManager , Manager
        fldSupplier , Supplier
        fldProductKind , ProductKind
        fldProductCategory , ProductCategory
        fldNomenclatureGroup , NomenclatureGroup
        fldProductGroup , ProductGroup
        fldProductCountry , ProductCountry
        fldProduct , Product
        fldYearQuarter , YearQuarter
        fldYearMonth , YearMonth
        fldYear , Year
        fldQuarter , Quarter
        fldMonth , Month
        fldDate , Date
    ];
LEFT JOIN ( ВиміриРейтинги )
LOAD _ Language , id as _ ПолеВиміриРейтингиId , _ Translation as _
ВимірРейтинги RESIDENT Translations ;
LEFT JOIN ( ВиміриРейтинги )
LOAD _ Language , id as _ МіткаВиміриРейтингиId , _ Translation as _
МіткаВиміриРейтинги RESIDENT Translations ;
ВиміриПорівняння :
LOAD * INLINE [
    _ ПолеВиміриПорівнянняId , _ МіткаВиміриПорівнянняId ,
    fldProductHierarchy1, ProductHierarchy1
    fldProductHierarchy2, ProductHierarchy2
    fldProductHierarchy3, ProductHierarchy3
    fldClientCategory , ClientCategory
    fldClientGroup , ClientGroup
    fldClient , Client
    fldManager , Manager
    fldSupplier , Supplier
    fldProductKind , ProductKind
    fldProductCategory , ProductCategory
    fldNomenclatureGroup , NomenclatureGroup
    fldProductGroup , ProductGroup
    fldProductCountry , ProductCountry
    fldProduct , Product
    fldSaleType , SaleType
    fldTransactionType , TransactionType
    fldYearQuarter , YearQuarter
    fldYearMonth , YearMonth
    fldYear , Year
    fldQuarter , Quarter
    fldMonth , Month
    fldDate , Date
];

```

```

LEFT JOIN ( ВиміриПорівняння )
LOAD _ Language , id as _ ПолеВиміриПорівнянняId , _ Translation as _
ВимірПорівняння RESIDENT Translations ;
LEFT JOIN ( ВиміриПорівняння )
LOAD _ Language , id as _ МіткаВиміриПорівнянняId , _ Translation as _
МіткаВиміриПорівняння RESIDENT Translations ;
Виміри ABCXYZ :
LOAD * INLINE [
_ ПолеВиміриABCXYZId , _ МіткаВиміриABCXYZId ,
fldProductHierarchy1, ProductHierarchy1
fldProductHierarchy2, ProductHierarchy2
fldProductHierarchy3, ProductHierarchy3
fldClient , Client
fldManager , Manager
fldSupplier , Supplier
fldProductCategory , ProductCategory
fldNomenclatureGroup , NomenclatureGroup
fldProduct_ID , Product_ID
fldYearQuarter , YearQuarter
fldYearMonth , YearMonth
fldYear , Year
fldQuarter , Quarter
fldMonth , Month
];
LEFT JOIN ( ВиміриABCXYZ )
LOAD _ Language , id as _ ПолеВиміриABCXYZId , _ Translation as _
ПолеВиміриABCXYZ RESIDENT Translations ;
LEFT JOIN ( ВиміриABCXYZ )
LOAD _ Language , id as _ МіткаВиміриABCXYZId , _ Translation as _
МіткаВиміриABCXYZ RESIDENT Translations ;
SET vIfValueShowType = if ( _ ValueShowType = '%', '+', '%');
_ ValueShowType :
LOAD * INLINE [
_ ValueShowOrder , _ ValueShowType
2%
1, ±
];
_TOPN:
LOAD * INLINE [
_TOPN
5
10
25
50
100
];
SET vStepB = 150;
SET vStepB1 = 2;
SET vPrice2 = 0;
SET vCost2 = 0;
SET vQty2 = 0;
SET vDiscount2 = 0;
//SET vModelDays = 10;
LET vAvgPrice = 'AVG([Середня ціна продажу] $'&EXCH&')';
LET vAvgCost = 'AVG([Середня ціна закупівлі] $'&EXCH&')';
LET vAvgQty = 'AVG([Середня кількість])';
LET vLoadTime = Interval ( now (1) - vLoadTime );
Exit Script ;

```