

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ  
УНІВЕРСИТЕТ

Механіко-технологічний факультет  
Кафедра кібербезпеки та програмного забезпечення

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ**

**до самостійної роботи**

**з дисципліни «Бази даних»**

**для студентів денної та заочної форм навчання за спеціальностями  
122/F3 «Комп'ютерні науки», 123/F7 «Комп'ютерна інженерія»,  
125/F5 «Кібербезпека та захист інформації», 172/G5 «Електроніка,  
електронні комунікації, приладобудування та радіотехніка»**

ЗАТВЕРДЖЕНО

На засідання кафедри кібербезпеки та  
програмного забезпечення,  
протокол від 25.08.2025 року № 1

КРОПИВНИЦЬКИЙ  
2025

Методичні рекомендації до самостійної роботи з дисципліни «Бази даних» для студентів денної та заочної форм навчання за спеціальностями 122/F3 «Комп'ютерні науки», 123/F7 «Комп'ютерна інженерія», 125/F5 «Кібербезпека та захист інформації», 172/G5 «Електроніка, електронні комунікації, приладобудування та радіотехніка» / уклад. В.В. Босько, Л.В. Константинова — Кропивницький: ЦНТУ, 2025. — 72 с.

Укладач: Босько В. В., Константинова Л. В.

Рецензенти: Смірнов О. А., д-р техн. наук, професор;  
Дрєєв О. М., канд. техн. наук.

© Босько В. В., Константинова Л.В., укладання, 2025  
© Центральноукраїнський національний  
технічний університет, 2025

## Теми самостійної роботи

з навчальної дисципліни Баз даних для студентів денної та заочної форм навчання за спеціальностями 122/F3 «Комп'ютерні науки», 123/F7 «Комп'ютерна інженерія», 125/F5 «Кібербезпека та захист інформації», 172/G5 «Електроніка, електронні комунікації, приладобудування та радіотехніка»

№з.с.	Назва теми
1	<b>Тема 1.</b> Загальні поняття інформаційних систем
2	<b>Тема 2.</b> Архітектура інформаційних систем
3	<b>Тема 3.</b> Моделювання даних. Реляційна модель даних. Загальна термінологія реляційної моделі даних
4	<b>Тема 4.</b> Проектування й застосування баз даних
5	<b>Тема 5.</b> Концептуальні моделі
6	<b>Тема 6.</b> Зв'язки між відношеннями
7	<b>Тема 7.</b> Розширена модель "сутність – зв'язок"
8	<b>Тема 8.</b> Історія мови SQL та огляд її можливостей
9	<b>Тема 9.</b> Етапи логічного проектування
10	<b>Тема 10.</b> Перетворення ER-діаграм в реляційні структури
11	<b>Тема 11.</b> Нормалізація та денормалізація
12	<b>Тема 12.</b> Фізична організація баз даних
13	<b>Тема 13.</b> Засоби для автоматизації проектування баз даних
14	<b>Тема 14.</b> Розподілена обробка даних
	<b>Робота над КР</b>
	<b>Разом:</b>

## **Самостійна робота №1**

**Тема:** Загальні поняття інформаційних систем

**Об'єктно-орієнтований підхід** являє собою метод, який дає можливість при розв'язанні складних задач в певній предметній області виділити та описати конкретні об'єкти, їх характеристики, зв'язки та взаємодію між ними.

**Предметна область (ПО)** - це цілеспрямоване первинне перетворення картини зовнішнього світу в деяку уможливлену картину, визначена частина якої втілюється в інформаційній системі в якості алгоритмічної моделі фрагмента дійсності [2]. ПО - частина реального світу, що розглядається в межах даного контексту, це сфера застосування конкретної бази даних (наприклад: медицина, освіта, залізничний транспорт тощо).

**База даних (БД) (DataBase (DB))** – сукупність спеціальним чином організованих даних, які зберігаються в пам'яті обчислювальної системи та відображають стан об'єктів та їх взаємозв'язки в даній ПО.

**Система керування базами даних (СКБД)** – комплекс мовних і програмних засобів, що призначені для створення, роботи та сумісного використання БД. (Наприклад: PostgreSQL, SQLite, MySQL, MS Access, Microsoft SQL Server, Oracle Database та ін.).

В СКБД входять такі компоненти (рисунок 1.1): ядро СКБД, підсистема засобів проектування й підсистема засобів обробки.

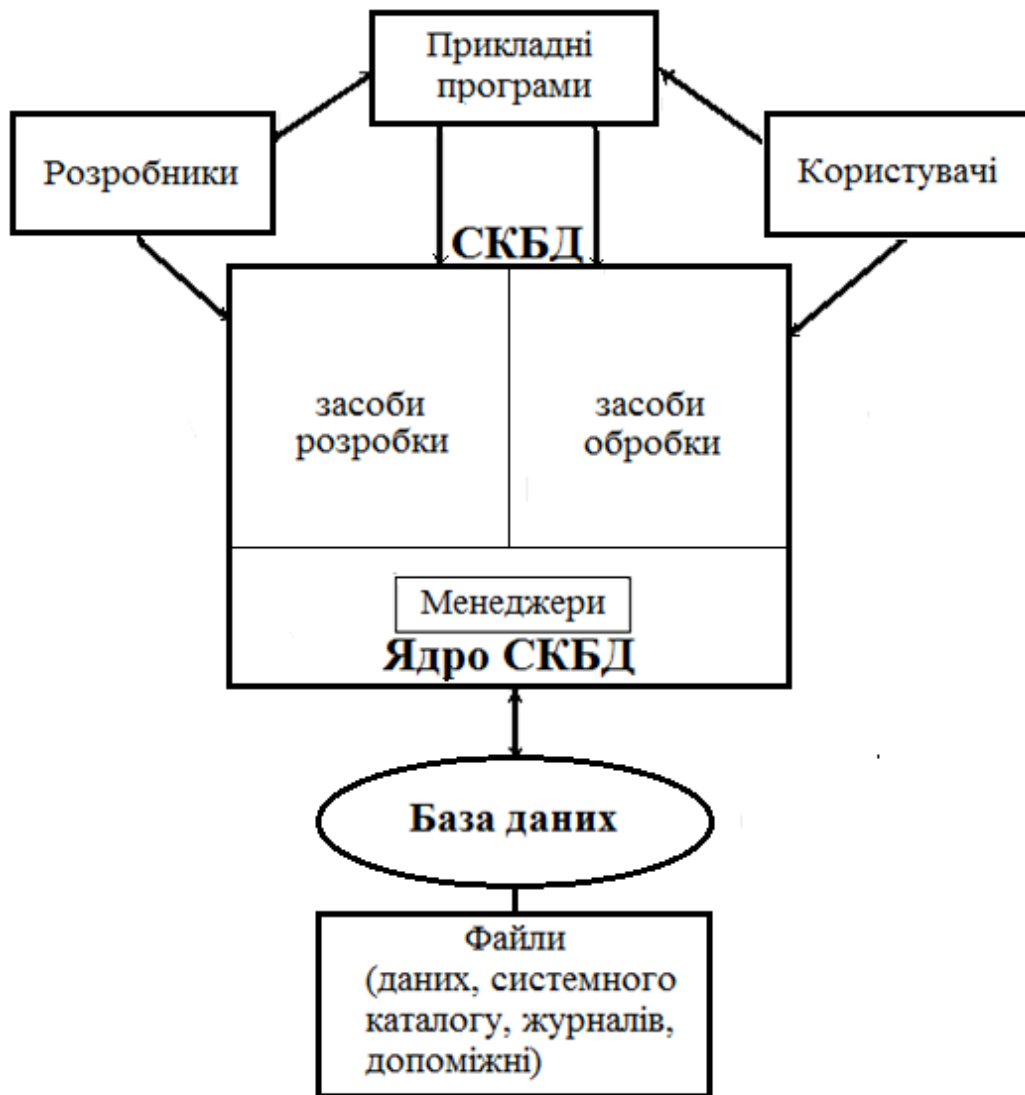


Рисунок 1.1 – Компоненти системи БД

**Ядро СКБД** – містить купу базових механізмів СКБД, які використовуються при будь-яких варіантах конфігурації системи. Ядро СКБД виконує функцію посередника між підсистемами засобів проектування й обробки та даними. Сучасні БД, у більшості, надають користувачу дані у вигляді таблиць. Ядро СКБД отримує запити від інших компонентів в термінах таблиць, стовпців, рядків і перетворює ці запити в команди операційної системи, які виконують запис і читання з фізичних носіїв інформації.

Крім того, ядро СКБД задіяне в управлінні транзакціями, блокуванні, резервному копіюванні й відновленні. В ядро входять менеджери буферів, даних, транзакцій, журналів.

**Менеджер буферів** – призначений для вирішення задач ефективної буферизації оперативної пам'яті.

**Менеджер даних** – призначений для керування зовнішньою пам'яттю, забезпечує створення структур для даних, що зберігаються, та допоміжних структур (таких як індекси й т.ін.).

**Менеджер транзакцій** – відповідає за підтримку механізмів фіксації й відміни транзакцій. Він пов'язаний з менеджером буферів оперативної пам'яті й забезпечує зберігання інформації, що потрібна після збоїв програми.

**Менеджер журналів** – реєструє відомості про виконання транзакцій, про наявних користувачів. Також забезпечує реєстрацію виконання застосування, доступи до структур даних й т. ін.

**Підсистема засобів проектування** це сукупність інструментів, які спрощують проектування й реалізацію БД. Як правило, ця сукупність містить засоби для створення об'єктів БД (таблиць, запитів, форм, звітів). СКБД також мають мови програмування й інтерфейси для роботи.

**Підсистема обробки** - виконує обробку компонентів застосування, що створюються за допомогою засобів проектування.

**Застосування БД** може складатись з таких об'єктів як форми, запити, меню, звіти й прикладні програми. Форми, звіти й запити можна створювати за допомогою засобів, що постачаються у комплексі з СКБД. Прикладні програми можуть бути написані або на вхідній мові СКБД, або іншій стандартній мові, а потім за допомогою СКБД з'єднані з БД.

**Інформаційна система (ІС (IS))** являє собою систему програмних, мовних, організаційних і технічних засобів, призначених для її централізованого зберігання та колективного використання даних.

Інформаційна система має функції збирання, зберігання, розповсюджує й обробляє інформацію.

**Банк даних (БнД)** – різновид ІС, в якій реалізовані функції централізованого зберігання та накопичення оброблюваної інформації, що об'єднується в одну або декілька БД.

**Словник даних (СД)** – це підсистема банку даних, що призначена для централізованого зберігання інформації про структури даних, взаємозв'язки

файлів БД, типи, формати даних, про коди захисту та розмежування доступу. Функції СД викликаються з головного меню системи та виконуються СКБД.

**Додаток** – це програма або комплекс програм, яка автоматизує обробку інформації для деякої прикладної задачі. Додатки бувають зовнішні та внутрішні. Для створення зовнішніх додатків використовуються будь яка мова програмування, яка має засоби доступу до БД. Внутрішні додатки пишуться на вбудованій мові програмування (наприклад, в MS Access це Visual Basic).

**Модель даних** – це набір правил, що визначають, як створюються та змінюються структури даних згодом, які операції дозволені над ними та обмеження цілісності, включаючи допустимі зв'язки, значення даних і послідовності їх змін. **Модель представлення даних** – це логічна структура даних, за якою побудована БД. Існують наступні основні моделі [2][4]:

1. Ієрархічна модель;
2. Мережева (або в деяких джерелах - мережна) модель;
3. Реляційна модель;
4. Об'єктно-орієнтована модель.

В залежності від моделі представлених даних, СКБД відповідно підрозділяються на: ієрархічні, мережеві (мережні), реляційні, об'єктно-орієнтовані та багатозначні (multi-value).

#### **Завдання. Дати відповіді на запитання:**

1. Що собою являє інформаційна система, автоматизована інформаційна система?
2. Дайте визначення банку даних.
3. Що представляє собою БД?
4. Дайте визначення СКБД.
5. Дайте визначення поняттю «Предметна область».
6. Що собою являє словник даних?
7. Назвіть основні моделі даних.
8. Які існують основні моделі даних?
9. Дайте визначення додатку.
10. Які компоненти входять в СКБД?

11. Які функції виконує інформаційна система?

## **Самостійна робота №2**

**Тема:** Архітектура інформаційних систем

### **Моделі даних**

**Модель даних** можна уявити, як абстракцію, в якій знаходять своє відображення більш важливі елементи функціонування визначеної ПО, а другорядні моменти не враховуються.

Модель даних виступає в ролі цільової моделі ПО. Розрізняють три головні складові моделі даних:

- **структурна частина** (визначає правила породження допустимих для даної СКБД видів структур даних);
- **керуюча частина** (відповідає за визначення допустимих операцій над певними структурами);
- **групи обмежень цілісності даних**, що засоби системи спроможні реалізувати.

**Моделювання даних** – це дії створення логічного подання структури БД.

Кожний рівень представлення інформації надає певну модель.

**Інфологічна модель** – дає уявлення про інформацію ПО природною мовою, незалежно від СКБД, що застосовується [5]. Ця модель подає інформаційно-логічний рівень абстрагування, що пов'язаний з визначенням об'єктів ПО, їх властивостей та взаємозв'язків між ними.

**Даталогічна модель** – відноситься до моделей логічного рівня, й дає уявлення про логічні зв'язки між елементами даних безвідносно до їх змісту й середовища зберігання даних. Ці моделі уподібнюють з логічними моделями.

**Фізична модель** – дає уявлення про спосіб зберігання даних на пристрої, представляючи інформацію про структуру записів, впорядкування, про шляхи доступу до інформації тощо.

**Модель "сутність-зв'язок" (ER-модель)** – визначає модель ПО й має у складі множину сутностей, множину зв'язків між сутностями, а також атрибутів сутностей. Модель також відображає обмеження цілісності даних, що пов'язано з двома множинами сутностей; та називається залежністю по

існуванню. За допомогою ER-моделей можливо графічно подавати моделі ПО. Вони входять в більшість CASE-продуктів.

**Семантична об'єктна модель** – це модель ПО, являє собою модель даних. Вона складається з частин - семантичних об'єктів, які містять набір атрибутів, що групуються за класами. Модель даних має більш розвинені засоби відображення семантики у порівнянні з іншими моделями (теоретико-множинними, теоретико-графовими).

**Теоретико-графова модель** – це модель даних, в якій дозволені структури даних можуть бути описані у вигляді графа. Наприклад, це може бути дерево. Необхідну допустиму групу операцій на мові маніпулювання даними, які ґрунтуються на цій моделі, представляють навігаційні операції.

**Теоретико-множинна модель** – це модель даних, в якій використовуються математичні засоби реляційної алгебри, реляційного числення, а також здійснюється маніпулювання таблицями як операції над даними.

**Фактографічні моделі** – це моделі, що містять відомості, що подаються як ретельно організовані групи формалізованих записів даних.

**Ієрархічна модель** – модель даних, що ґрунтується на ієрархічній, деревоподібній структурі даних. Якщо розглядати вершини цієї структури, то це - є записи, які складаються з простих елементів даних різного типу. Запису предку відповідає довільне число екземплярів нащадків записів кожного типу.

**Документальні моделі** – це моделі, у яких окремим елементом інформації є документ, що не ділиться, а не структурована інформація про нього, яка часто має обмежений формат, або не структурується взагалі. Ці моделі, здебільшого, представляють тексти, надані природною мовою, і працюють з документами у вільних форматах.

**Мережна (або в деяких джерелах мережева) модель** – це така модель, де структури даних надаються у вигляді графа, вершинами якого можуть бути дані різних типів (від атомарних елементів даних та до записів складної структури). Відрізняючись від ієрархічної моделі нащадок в цих моделях може наслідувати будь-яку кількість предків.

**Реляційна модель** – модель даних, що ґрунтується на понятті відношення (математичному) й поданні відношень у вигляді таблиць.

**Постреляційна модель** – це реляційна модель, що розширена та без обмежень неподільності даних, що накладаються на записи таблиць. Допускається застосовувати багатозначні поля (тобто, значення таких складаються з підзначень). Набір значень в багатозначних полях розглядається, як самостійна таблиця, яка вбудовується в основну таблицю. Часто ці моделі порівнюють з об'єктно-реляційними моделями.

**Об'єктно-орієнтована модель** – це модель даних, що базується на понятті об'єкта (або сутності) що описує стан і поведінку. За допомогою атрибутів визначається стан об'єкта, а наявна сукупність операцій чи методів визначають поведінку цього об'єкта. До того ж є можливість між типами об'єктів організувати зв'язки.

**Багатомірна модель.** Ця модель даних дозволяє оперувати багатомірним представленням даних (у формі гіперкубу) та дозволяє підтримку аналізу даних. В цій моделі є можливим конструювання різних агрегатних операцій над даними у межах гіперкубу, побудова всяких його проєкцій (підмножин гіперкубу), деталізація й обертання даних тощо.

**Тезаурусна модель** – це така модель, що подає документ за допомогою описувачів (дескрипторів) та вагомих відношень між лексичними одиницями (ціле-частина, рід-вид, клас-підклас тощо). Завдяки цим моделям дозволяється збільшити продуктивність дескрипторних моделей коштом кращого відображення ПО.

**Дескрипторна модель** – це модель, в якій описується кожен документ за допомогою деякого дескриптора. Сам дескриптор має певну структуру та являє собою множину якихось лексичних одиниць (термінів, слів, словосполучень), що необхідні при роботі з документами. При чому дескриптори не пов'язані між собою.

**Гіпертекстова модель** – це модель, яка ґрунтується на розмітці документа зі спеціальними навігаційними конструкціями, які відтворюють смислові

зв'язки між різними документами або в межах окремих фрагментів одного документа.

Подібні конструкції створюють якусь семантичну мережу в документальній базі.

Трирівнева модель архітектури СКБД була заініційована спеціальною організацією, Комітетом планування стандартів і норм SPARC (Standarts Planning and Requirements Committee) Американського національного інституту стандартів ANSI (American National Standarts Institute) в 1971 р. [1][2] (концептуальний, зовнішній та внутрішній рівні).

Опис структури даних на будь-якому рівні називається **схемою**. Розрізняється три типи схем БД, що визначені відповідно до рівнів абстракції архітектури СКБД.

В мережі при інтеграції комп'ютерів є можливість розподілу додатків, що працюють з єдиною БД, та самої бази даних також. Найбільш поширеною є схема, при якій кожен користувач має свою персональну БД (КБД) і звертається до серверної БД (СБД) за інформацією, що спільно використовується багатьма користувачами.

Під **сервером** розуміється комп'ютер або програма, які керують певними ресурсами. **Клієнт** – це теж комп'ютер або програма, що користується цим ресурсами.

Таким чином є можливість поєднувати всі переваги централізованого зберігання даних та індивідуальної роботи користувачів.

Вибір моделі даних залежить від ряду факторів [5], таких як :

1. Обсяг інформації, який потрібно обробити і зберегти.
2. Складність задач, що вирішуються за допомогою моделі даних.
3. Наявність технічного та програмного оснащення для реалізації моделі даних.
4. Засоби маніпулювання даними, які використовуються для операцій з інформацією.
5. Цілісність та захист даних, що є важливими аспектами при обробці і зберіганні інформації.

6. Критерії якості, такі як надійність і точність моделі.
7. Можливість розвитку й масштабування моделі даних для відповіді на майбутні потреби.

Також варто зауважити, що вибір моделі даних часто визначає вибір системи керування базами даних, що буде використовуватися для роботи з даними.

**Завдання. Дати відповіді на запитання:**

1. Як називають процес побудови логічного подання структури БД?
2. Як можна визначити поняття модель даних?
3. Які основні складові моделі даних?
4. Що відображає інфологічна модель?
5. Що представляє собою даталогічна модель даних?
6. Що відображає фізична модель?
7. Що представляє семантична модель?
8. Що відображає модель «сутність-зв'язок»?
9. Що передбачають документальні моделі?
10. Яка модель називається теоретико-множинною, а яка теоретико-графовою?
11. Що можна сказати про фактографічну модель?
12. Що представляє собою ієрархічна модель?
13. Що відображає мережна модель?
14. На чому ґрунтується реляційна модель?
15. Що відомо про об'єктно-орієнтовану модель?
16. Яку модель називають багатомірною?
17. Що відомо про трирівневу модель архітектури СКБД?
18. Що називають схемою?
19. Що розуміють під локальним поданням даних?
20. Що розуміють під концептуальним поданням даних?
21. Що розуміють під формалізованим поданням даних?
22. Що розуміють під внутрішнім поданням даних?

### Самостійна робота №3

**Тема:** . Моделювання даних. Реляційна модель даних. Загальна термінологія реляційної моделі даних

**Реляційна модель даних (РМД)** деякої ПО є набором відношень, що змінюються з часом. Під час побудови інформаційної системи множина відношень дає можливість зберігати інформацію про об'єкти предметної області та створювати зв'язки між ними.

В 1970 році Едгар Кодд запропонував концепцію реляційної моделі даних для розв'язування такої задачі: забезпечення незалежності подання опису даних від прикладних програм.

РМД ґрунтується на понятті «відношення» (relations), що подається у формі таблиці з певними умовами обмежень, яких потрібно дотримуватись.

Основні елементи РМД можна надати у вигляді деякої таблиці:

Таблиця 3.1 – Основні елементи РМД

Елементи РМД	Форма подання
Відношення	Таблиця
Схема відношення	Рядок заголовків стовпців таблиці
Кортеж	Рядок таблиці (запис)
Сутність	Опис властивостей об'єкту
Атрибут	Заголовок стовпця таблиці (поле)
Домен	Множина допустимих значень атрибутів
Значення атрибута	Значення поля в записі
Первинний ключ	Один або декілька атрибутів
Тип даних	Тип значень елементів таблиці

**Відношення** – це основне поняття БД, що являє собою двовимірну таблицю, в якій можуть міститись певні дані.

Відношення РБД можна розділити на дві групи:

- об'єктні відношення;
- зв'язні відношення.

Про **об'єктне відношення**: зберігає дані, що стосуються об'єкта (екземпляри сутності). В об'єктному відношенні є один або декілька атрибутів, що однозначно ідентифікують об'єкт. Їх називають **ключем** відношення (або первинним ключем). Основне обмеження в реляційній МД, що забезпечує цілісність даних, наступне: атрибути в об'єктному відношенні не повинні дублюватись.

**Зв'язне відношення** – це відношення, що зберігає ключі двох або більше об'єктних відношень. Завдяки цим ключам встановлюються зв'язки між об'єктами відношень.

**Сутність** – об'єкт будь-якого походження, дані про який зберігаються в БД. Інформація про певну сутність зберігаються у певному відношенні.

**Атрибути** – це властивості, завдяки яким можна характеризувати сутність.

В структурі таблиці кожен елемент має назву, яка відповідає заголовку певного стовпця у таблиці.

**Кортеж** являє собою рядок у таблиці, тобто один екземпляр таблиці (запис). Порядок кортежів у відношенні є не визначеним. Проте порядок атрибутів у відношенні є визначеним, і якщо поміняти їх місцями, отримаємо нове відношення.

**Домен** - це набір всіх можливих значень певного атрибута у відношенні.

**Схема відношення** або заголовок відношення – це перелік імен атрибутів.

**Тіло відношення** – так називають набір кортежів у відношенні.

На рисунку 3.1 надається приклад таблиці, яка відображає тип об'єкта реального світу, а кожен рядок – конкретний екземпляр об'єкта.

**Стовпець таблиці** – це набір значень певного атрибута об'єкта. Ці значення вибираються з набору всіх можливих значень атрибутів об'єкта, який називається доменом.

Країна	Столиця	Площа, км 2	Грошова одиниця
Австрія	Відень	83871	євро
Україна	Київ	603628	гривна
Франція	Париж	551695	євро

Рисунок 3.1 – Елементи РМД таблиці «Атлас світу»

Ключ – є важливим поняттям РМД. Існують наступні поняття пов’язані з ключем:

**Потенційний ключ** – це мінімальний піднабір атрибутів відношення, який однозначно ідентифікує кортежі даного відношення.

**Первинний ключ** – це потенційний ключ, що обрали для унікальної ідентифікації кортежів відношення.

**Вторинний ключ** – ключ, кожному значенню якого може відповідати більше ніж один екземпляр індексованих даних, може бути унікальним для деякої підмножини даних, але може мати повторювані значення в інших частинах таблиці.

**Зовнішній ключ** – це сукупність атрибутів одного відношення, значення яких є одночасно також значеннями первинного або потенційного ключа іншого відношення.

В кожному відношенні обов’язково має бути атрибут або комбінація атрибутів, що може виконувати функцію ключа. Бувають відношення, які мають кілька комбінацій атрибутів, кожна з яких однозначно визначає всі кортежі відношення та претендують на роль ключа відношення. Можна вибирати любий з можливих варіантів ключів, як первинний. Первинний ключ називають не надмірним, якщо він складається з мінімально необхідної множини атрибутів.

**Ключі застосовують для досягнення наступних цілей:**

1. Запобігають дублюванню значень в ключових полях [2] (інші атрибути не враховуються);

2. Впорядкування кортежів, що стає можливим при збільшенні або зменшенні значень всіх ключових полів;
3. Пришвидження роботи з кортежами відношення;
4. Необхідні для операцій поєднання таблиць.

Нехай відношення  $R$  має не ключовий атрибут  $A$ , значення якого є значеннями ключового атрибута іншого відношення, тоді говорять, що атрибут  $A$  відношення  $R$  є **зовнішнім ключем** (рисунок 3.2).

В нашому прикладі є два відношення:  $R\_Студенти$  з первинним ключем «ПІБ» і  $R\_Предмети$  з первинним ключем «Код предмета», які пов'язані з відношенням  $R\_Успішність$ , в якому атрибути «ПІБ» і «Код предмета» є зовнішніми ключами по відношенню до відношень  $R\_Студент$  і  $R\_Предмети$  відповідно.

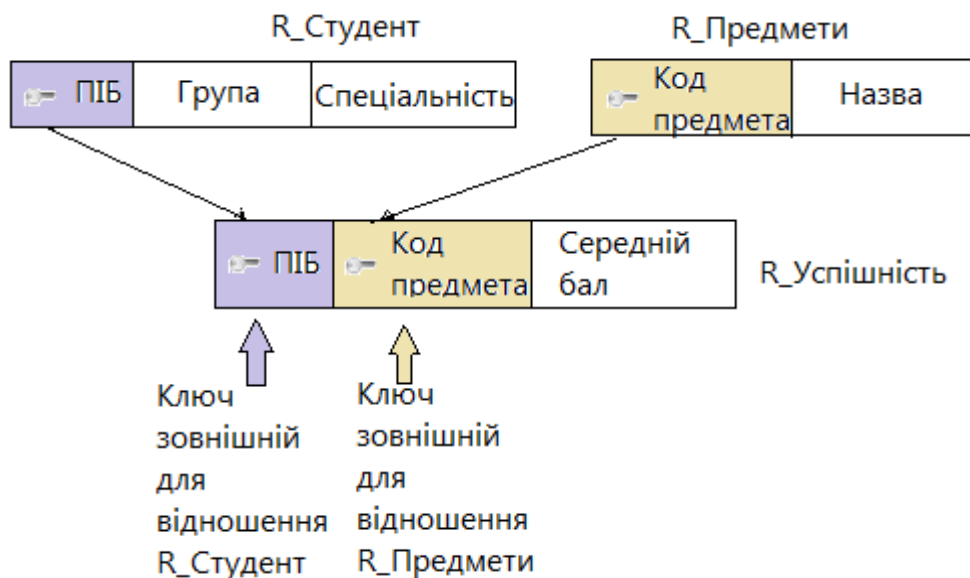


Рисунок 3.2 – Приклад зовнішніх ключів

У РМД є деякі властивості: це природність та простота структур даних, що використовуються й операцій маніпуляції даними; незалежність від середовища зберігання даних; підтримка віртуальних, а не фізичних зв'язків між даними.

Реляційна БД може складатись з наступних частин:

- **масиви інформації** (у вигляді таблиць, індексів);
- **інформація системи** (структура БД, обмеження цілісності, правила);
- **прикладні програми** (модулі, процедури, тригери, макроси, API, тощо).

Операційні можливості відношення базуються на **реляційній алгебрі** та **реляційному обчисленні**.

Існують мови запитів реляційної алгебри та мови реляційного обчислення, принципи побудови яких запропонував Е.Кодд. На відміну від теоретичних мов, реальні сучасні мови запитів, такі як SQL або QBE забезпечують не тільки функції відповідної теоретичної мови, але і реалізують деякі додаткові арифметичні операції, операції друку, тощо.

### **Цілісність баз даних**

**Цілісність баз даних** – властивість даних, що визначає повноту і коректність інформації, яка вміщується в БД.

В підтримку цілісності БД входить такі складові:

- структурна цілісність;
- обмеження реальних значень даних;
- посилальна цілісність.

При **структурній цілісності** вимагається дотримання наступних умов:

- існування тільки реляційних відношень (серед структур даних);
- виключення кортежів, що дублюються;
- обов'язкове існування первинного ключа в кожному відношенні;
- обмеження доменів, яке передбачає визначення кожного атрибуту на своєму домені;
- можливість застосування значень NULL (невизначених значень) (позначає відсутність будь-якого значення атрибуту).

**Обмеження реальних значень даних** – це вимога, щоб значення поля відповідали деякому інтервалу значень, або задовільняли певне арифметичне співвідношення між значеннями певних полів таблиці. Обмеження значень можуть включати також визначення певних форматів для полів, задоволення значень полів певним статистичним умовам, бізнес правилам предметної області й т.ін.

**Посилальна цілісність** вказує, що зміни в таблицях повинні виконуватися синхронно, а зміст двох пов'язаних таблиць має відповідати таким правилам:

- кожному запису головної таблиці відповідає нуль або більше записів підпорядкованої таблиці;

— в підпорядкованій таблиці немає записів, які не відповідають записам в головній таблиці;

— Кожний запис підпорядкованої таблиці має тільки один батьківський запис головної таблиці.

З умовами цілісності даних визначається, які дані можуть бути записані в БД у результаті внесення, видалення або редагування даних (правила можна оглянути у таблицях 3.2 і 3.3).

Таблиця 3.2 - Правила видалення даних в БД

RESTRICT	Заборона вилучення рядка з батьківської таблиці, якщо він має нащадків в підлеглій таблиці.
CASCADE	При видаленні рядка з батьківської таблиці в підлеглій таблиці всі рядки-нащадки автоматично видаляються
SET NULL	При видаленні рядка з батьківської таблиці в підлеглій таблиці всім зовнішнім ключам рядків-нащадків автоматично привласнюється значення NULL
SET DEFAULT	При видаленні рядка з батьківської таблиці в підлеглій таблиці всім зовнішнім ключам рядків-нащадків автоматично привласнюється певне значення встановлене за замовчуванням

Таблиця 3.3 - Правила оновлення даних в БД

RESTRICT	Заборона зміни первинного ключа в рядку батьківської таблиці, якщо в підлеглій таблиці цей рядок має нащадків
CASCADE	При зміні первинного ключа в рядку батьківської таблиці в підлеглій таблиці відповідні значення зовнішнього ключа також автоматично змінюються у всіх рядках-нащадках для того, щоби відповідати новому значенню первинного ключа
SET NULL	При зміні первинного ключа в рядку батьківської таблиці в підлеглій таблиці відповідні значення зовнішнього ключа також автоматично змінюються у всіх рядках-нащадках і їм привласнюється значення NULL
SET DEFAULT	При зміні первинного ключа в рядку батьківської таблиці в підпорядкованій таблиці відповідні значення зовнішнього ключа також автоматично змінюються у всіх рядках-нащадках і їм привласнюється певне значення встановлене за замовчуванням

Також можливо виконання правила NONE – не виконуються ніякі дії і правила NULL ALLOWED – дозволяються невизначені значення.

При додаванні даних в таблицю (INSERT) необхідно дотримуватися такої послідовності введення: спочатку дані вводяться в батьківську таблицю, а потім в підпорядковану.

**Алгеброю** називають множину об'єктів із заданою на ній купою математичних операцій та правил, які замкнені відносно цієї множини.

Основною множиною в реляційній алгебрі є множина відношень. Варіант реляційної алгебри, запропонований Коддом, містить наступні **основні операції**: об'єднання, декартів добуток, різниця, перетин, проекція, селекція, ділення, з'єднання.

### **Контрольні запитання:**

1. Як можна визначити реляційну модель даних (РМД) деякої ПО?
2. Що лежить в основі концепції РМД?
3. Як можна визначити поняття відношення для РБД?
4. Яке відношення називають об'єктним?
5. Яке відношення називають зв'язним?
6. Що означає поняття сутність?
7. Що означає поняття атрибут?
8. Що означає поняття кортеж?
9. Що означає поняття домен?
10. Що означає поняття схема відношення?
11. Із чого складається тіло відношення?
12. Що означає поняття первинний ключ?
13. Що означає поняття зовнішній ключ?
14. Що означає цілісність БД?
15. Що означає структурна цілісність?
16. Що передбачає посилальна цілісність?
17. Що означає поняття алгебра для БД?

## **Самостійна робота №4**

**Тема:** Проектування й застосування баз даних

**Життєвий цикл БД** складається з наступних етапів:

1. Планування БД;
2. Визначення вимог до БД;
3. Проектування БД (концептуальне, логічне, фізичне);
4. Розробка застосувань;
5. Реалізація й завантаження даних;
6. Тестування;
7. Експлуатація.

Кожний етап значною мірою залежить від складності програмного продукту, що розробляється. Для невеликих ІС кількість етапів може бути зменшена.

**Що розглядається в кожному етапі проектування:**

1. **Етап планування** БД передбачає створення загального плану, який дозволить ефективно реалізувати етапи життєвого циклу. Тут вирішуються такі питання:

- аналіз існуючих інформаційних систем;
- доцільність зміни існуючої ІС;
- обсяг робіт і ресурсів, вартість проекту;
- визначення технічного завдання для проекту БД;
- визначення технічних вимог;
- розробка методології збору даних, визначення їх формату;
- визначення необхідної документації;
- визначення послідовності проектування і реалізації застосувань.

2. На **етапі визначення вимог** до бази даних вирішуються такі задачі:

- визначення діапазону дії і меж застосувань БД;
- визначення складу користувачів і сфер застосування;
- визначення представлень користувачів, що підтримуються БД.

На цьому етапі також збираються та аналізуються вимоги користувачів:

- опис даних, що застосовуються (вхідні й вихідні документи);

- детальні відомості про транзакції;
- відомості про засоби застосування даних.

Грунтуючись на всій попередній інформації створюються специфікації вимог користувачів.

3. Процес **проектування БД** являє собою послідовність переходів від неформального мовного опису інформаційної структури предметної області до формалізованого опису об'єктів ПО в термінах деякої моделі. Проектування БД складається з таких етапів:

- системний аналіз ПО;
- концептуальне проектування;
- логічне проектування;
- фізичне проектування.

4. **Застосування** – це програма, яка призначена для вирішення деякої сукупності задач в даній ПО, або яка являє собою типовий інструментарій, що застосовується в різних областях.

На цьому етапі вирішуються такі задачі:

- проектування транзакцій;
- проектування інтерфейсів користувачів.

5. Для етапу **реалізації** характерні вирішення наступних питань:

- встановлюється технічне і програмне забезпечення СКБД;
- реалізується проект БД;
- реалізуються прикладні програми;
- реалізуються форми введення/виведення даних і звіти;
- наповнення БД даними;
- захист БД від несанкціонованого втручання;
- підтримка цілісності БД.

#### **Контрольні запитання:**

1. З яких етапів складається життєвий цикл БД?
2. Що спільного у життєвих циклах інформаційної системи і бази даних?
3. Яка різниця між функціональним і предметним підходами в

проектуванні БД?

4. У чому полягає спільність і різниця трьох етапів проектування БД?
5. Назвати етапи проектування БД.
6. З чого складається етап планування БД?
7. У чому полягає аналіз вимог до БД?
8. Яку роль відіграють застосування в проектуванні БД?
9. Які задачі вирішуються на етапі експлуатації?

## Самостійна робота №5

### Тема: Концептуальні моделі

Концептуальна модель представляє загальний погляд на дані. Існують два головних підходи до моделювання даних при концептуальному проектуванні:

- семантичні моделі;
- об'єктні моделі.

*Семантичні моделі* загалом відображають структури даних. Найбільш поширеною семантичною моделлю є модель "сутність – зв'язок" (Entity Relationship model, ER-модель). *ER-модель* складається із сутностей, зв'язків, атрибутів, доменів атрибутів, ключів. Моделювання даних відображає логічну структуру даних (схоже як блок-схеми алгоритмів відображають логічну структуру програми).

*Об'єктні моделі*, головним чином, описують поведінку об'єктів даних та засоби маніпуляції даними. Першорядне поняття таких моделей – це об'єкт, або сутність, що описується станом і поведінкою. Стан об'єкта визначають множиною його атрибутів, а поведінка об'єкта визначається набором операцій, що специфіковані саме для нього.

Ці моделі знаходять свою реалізацію в Extended Entity Relationship model (EER-модель), тобто в розширеному ER- моделюванні.

### Модель "сутність-зв'язок"

ER-моделювання являє собою низхідний підхід до проектування БД, який починається з визначення найбільш важливих даних, які називаються *сутностями* (entities), і *зв'язків* (relationships) між даними, які повинні бути подані в моделі. Наступною дією в модель заноситься інформація про властивості сутностей та зв'язків, що називають *атрибутами*, а також обмеження, що відносяться до сутностей, зв'язків й атрибутів. Завдяки ER-моделі маємо графічне подання логічних об'єктів і їх відношень в структурі бази даних.

### Завдання:

1. Закінчити твердження:

1). Найбільш поширеною семантичною моделлю є модель ....

2). *ER-модель* складається із ....

2. Відповісти на запитання:

1). Що описують головним чином об'єктні моделі?

2). Що являє собою ER-модельовання?

## Самостійна робота №6

Тема: Зв'язки між відношеннями

### Зв'язки

Після перетворення концептуальної моделі залишаються такі типи зв'язків:

1. "один до одного";
2. "один до багатьох";
3. рекурсивні зв'язки;
4. суперклас – підклас.

В реляційній моделі зв'язки між відношеннями реалізуються шляхом застосування первинних і зовнішніх ключів.

#### 1. "Один до одного"

В концептуальних моделях даних визначають наступні **обмеження ступеня участі сутностей**:

- обов'язкова участь для обох сутностей;
- обов'язкова участь для однієї сутності;
- необов'язкова участь для обох сутностей.

Перетворення до реляційної моделі будуть різнитись в залежності від визначених обмежень.

#### 2. "Один до багатьох"

Кожне відношення, що відповідає підлеглій (дочірній) сутності, має набір атрибутів з основної (батьківської) сутності, це - первинний ключ основної сутності. У відношенні, що відповідає дочірній сутності, цей набір атрибутів перетворюється в зовнішній ключ (FOREIGN KEY, FK).

Атрибути, що відповідають зовнішньому ключу набувають властивості допустимості невизначених значень (тобто NULL) при моделюванні необов'язкового типу зв'язку. У разі обов'язкового типу зв'язку атрибути набувають властивості відсутності невизначених значень (NOT NULL).

#### 3. "Багато до багатьох"

Кожний зв'язок M:N потребує створення додаткового відношення, яке представляє цей зв'язок та додати в нього всі атрибути, які складають цей зв'язок. В якості зовнішніх ключів у новому відношенні беруться копії

атрибутів первинного ключа сутностей, які беруть участь у зв'язку. Ці зовнішні ключі представляють також первинний ключ нового відношення.

### **Інші види зв'язків**

**4. Рекурсивні зв'язки 1:1** виконуються згідно правил визначених раніше для зв'язку між двома сутностями 1:1. Для рекурсивного зв'язку 1:1 з обов'язковою участю двох сторін, реляційна схема представляється у вигляді одного відношення з двома копіями первинного ключа. Одна копія відповідає зовнішньому ключу. Для рекурсивного зв'язку 1:1 з обов'язковою участю тільки однієї сторони створюється або одне відношення, або нове відношення, яке відображає цей зв'язок. Для рекурсивного зв'язку 1:1 з необов'язковою участю обох сторін створюється нове відношення.

Для **складних типів зв'язків** вводиться нове відношення, яке відображає цей зв'язок і складається з атрибутів, які входять в цей зв'язок.

### **5. Зв'язки "суперклас – підклас"**

Щоб виконати трансформацію зв'язку типу суперклас – підклас у реляційну модель потрібно звернути увагу на обмеження ступеня участі у зв'язку (Optional, Mandatory) та обмеження неперетинання (Or, And). Є можливим виконати чотири сполучення, перетворення яких дає чотири реляційні схеми. Має вплив на схему також те, чи беруть участь підкласи в різних зв'язках, число сутностей у зв'язку тощо. Діапазон можливих варіантів рішення є достатньо великим і конкретна схема вибирається в кожному конкретному випадку з урахуванням множини факторів.

#### **Завдання:**

1. Як відображається зв'язок "багато до багатьох" на реляційній схемі?
2. Як відображаються складні зв'язки і зв'язки з атрибутами на реляційній схемі?
3. Як відображаються рекурсивні зв'язки в реляційній схемі?
4. Яким чином подається зв'язок "один до одного" залежно від рівня участі сутностей на реляційній схемі?
5. Як відображається зв'язок "один до багатьох" залежно від рівня участі сутностей на реляційній схемі?

6. Як відображається зв'язок суперклас-підклас залежно від ступеня участі сутностей і обмеження неперетинання на реляційній схемі?

## Самостійна робота №7

**Тема:** Розширена модель "сутність – зв'язок"

Для більш складних застосувань, що розширюють можливості, в семантичне моделювання були введені додаткові концепції у вигляді **розширеної ER-моделі** (Enhanced Entity Relationship, EER-модель). Окрім всіх загальних концепцій ER-моделі вона містить концепції узагальнення, уточнення, агрегування та композиції. Ці додаткові концепції за підґрунтя мають поняття **суперклас** та **підклас**. Суперклас може утримувати кілька підкласів. Як у прикладі, де підкласи **Керівник**, **Викладач** і **Лаборант** є учасниками суперкласу **Співробітник**. Тобто, кожен екземпляр підкласу одночасно є екземпляром суперкласу. Зв'язок між суперкласом та підкласом відноситься до зв'язку типу 1:1.

Поняття суперклас і підклас допомагають визначити для підкласів власні атрибути і атрибути, що наслідуються від суперкласу, наприклад, підклас **Викладач** повинен мати однакові атрибути, що і всі **Співробітники**. Але він має і власні атрибути, які не визначені для інших категорій працівників ВУЗу. Серед атрибутів можуть бути наступні: вчене звання, науковий ступінь, номер диплому про вчене звання, кількість наукових робіт, кількість навчально-методичних праць тощо. Якщо підкласи для об'єкту **Співробітник** відсутні, то треба було б створити атрибути, що мали б невизначене значення для певних співробітників (наприклад для інженерів). Підклас може мати власні зв'язки, що не підходять для всіх учасників суперкласу. Як приклад, **Викладач** може мати підкласи **Професор**, **Доцент**, **Старший викладач**, **Асистент**. Підклас наслідує не тільки атрибути, але й зв'язки суперкласу.

Термін **уточнення** означає процес збільшення різниці між екземплярами об'єкта завдяки визначенню їх характеристик, що не мають подібності. Цей процес - низхідний. Наприклад, кроки від об'єкта **Співробітник** до об'єктів **Викладач** та **Керівник**.

**Узагальнення** означає процес зменшення відмінностей між об'єктами до мінімуму завдяки виділенню їх спільних характеристик. Цей процес -

висхідний. Як у прикладі, кроки від об'єктів **Керівник** і **Викладач** до об'єкту **Співробітник**.

Під час виконання процесів узагальнення або уточнення можуть застосовуватися обмеження не перетинання й ступеню участі.

Може бути що підкласи з наборів сутностей перетинаються, а може й не перетинаються. У випадку коли підкласи суперкласу не перетинаються, то це вказує, що кожен екземпляр сутності може бути елементом тільки для одного з підкласів (Or). Зв'язки, що не перетинаються, мають позначку "G". До прикладу, співробітник має можливість працювати або на посаді професора, або на посаді доцента, й не може бути одночасно на обох посадах.

У випадку коли перетинаються підкласи суперкласу, то це вказує, що будь-який екземпляр сутності може бути одночасно елементом кількох підкласів (And). А зв'язки, що перетинаються, мають позначку "Gs". Зустрічається, наприклад, що завідувач кафедри проводить заняття та, одночасно, виконує обов'язки керівника та викладача.

Завдання:

1. Наведіть приклад діаграми з використанням понять суперклас та підклас.
2. Наведіть приклад діаграми зв'язку суперкласу з підкласом з обов'язковою участю.

## **Самостійна робота №8**

**Тема:** Історія мови SQL та огляд її можливостей.

SQL - найрозповсюдженіша мова запитів серед мов програмування. Адже СКБД - MySQL, PostgreSQL і MicrosoftSQLServer - поширені повсюдно: у великому і малому бізнесі, в лікарнях, банках, університетах і так далі. В принципі, SQL не обмежується тільки настільними девайсами: СКБД SQLite з успіхом зайняла своє місце на Android-смартфонах і мобільних пристроях компанії Apple. Однак були часи, коли не було смартфонів, а ця мова вже існувала. Історія SQL - це не роки, але десятиліття. Повірили в нього не відразу.

### **Система R і IBM**

Перші згадки про цю мову датуються 1974 роком. SQL створювалася в рамках проекту експериментальної реляційної СКБД SystemR. Займалася цим проектом компанія IBM.

Спочатку мова називалася SEQUEL (StructuredEnglishQueryLanguage), але потім слово «англійська» пропало з цього словосполучення, а аббревіатура набула того вигляду, до якого ми давно вже звикли. З одного боку, SQL був орієнтований на зручне і зрозуміле користувачам формулювання запитів до реляційних БД. З іншого боку, практично з самого початку вона була так званою «повною мовою БД». Це означає, що SQL включала:

- засоби визначення та маніпулювання схемою БД;
- засоби визначення обмежень цілісності і тригерів;
- засоби визначення подань БД;
- засоби визначення структур фізичного рівня, що підтримують ефективне виконання запитів;
- засоби авторизації доступу до відносин і їхніх полів;
- засоби визначення точок збереження транзакції і виконання фіксації і відкатів транзакцій.

Правда, в ньому не були реалізовані засоби синхронізації доступу до об'єктів БД з боку паралельно виконуваних транзакцій. Справа в тому, що розробники спочатку розраховували, що необхідну синхронізацію неявно виконує СКБД.

Мова реалізована в переважній більшості СКБД - як в реляційних, так і нереляційних. Метою розробки було створення простої непроцедурної мови, якою міг скористатися будь-який користувач, який навіть не має навичок програмування.

Розробкою мови запитів займалися Дональд Чемберлін (Donald D. Chamberlin) і Рей Бойс (RayBoyce).

Шлях до комерційної реалізації SQL, який пройшла IBM, називають рухом «зверху вниз».

Oracle, Informix і Sybase пішли іншим шляхом - «від низу до верху»: в перших версіях цих систем, випущених на ринок, використовувалося істотно обмежена підмножина SQL System R. А далі вони почали поступово розширюватися. Однак у першій комерційній реалізації SQL в СКБД Oracle в операторах вибірки не допускалося використання вкладених підзапитів і була відсутня можливість формулювання запитів із з'єднаннями декількох відносин.

Зростаюча зацікавленість ринку в якнайшвидшому переході до реляційних систем управління базами даних дозволила розробникам перерахованих вище компаній домогтися комерційного успіху. Тепер Oracle, Informix, Sybase і Microsoft SQL Server підтримують досить потужні діалекти SQL.

Поява численних діалектів SQL і їх розростання мало призвести до проблем сумісності та до інших суперечностей.

Однак діяльність по стандартизації мови SQL почалася дуже вчасно - практично одночасно з появою його перших комерційних реалізацій. У 1982 році комітету по базам даних Американського національного інституту стандартів (ANSI) було доручено розробити специфікацію стандартної мови реляційних баз даних.

Після відхилення ряду невдалих версій стандарту в 1986 році експерти прийшли до єдиного знаменника. А в 1987 році стандарт SQL / 86 був схвалений Міжнародною організацією зі стандартизації (ISO).

Тим не менш, можна стверджувати, що базовий набір операторів SQL, що включає оператори визначення схеми БД, вибірки та маніпулювання даними, авторизації доступу до даних, підтримки вбудови SQL в мови програмування та

оператори динамічного SQL, в комерційних реалізаціях склався та більш-менш відповідає стандарту.

Стандарти SQL (ISO/IEC 9075) – це офіційні специфікації мови запитів до баз даних, що розробляються ANSI та ISO, які визначають синтаксис та семантику SQL; вони регулярно оновлюються (остання версія — SQL:2023) і є основою всім систем керування базами даних, хоча кожна СКБД (Oracle, Microsoft SQL Server, PostgreSQL та інших.) має розширення й діалекти (наприклад, T-SQL, PL/SQL).

Завдання:

Відповісти на запитання:

1. Якими роками датуються перші згадки про SQL?
2. Яка компанія займалася проектом СКБД SystemR?
3. Який стандарт став першим всесвітньо прийнятим стандартом мови SQL?
4. Який основний набір операторів SQL?

## Самостійна робота №9

### Тема: Етапи логічного проектування

Логічне проектування для реляційної моделі даних полягає у створенні реляційної схеми, визначенні кількості та структури таблиць, формуванні запитів до БД, визначенні типів звітних документів, розробці алгоритмів обробки інформації, створенні форм для введення й редагування даних в БД і вирішенні інших задач. Таким чином концептуальні моделі завдяки певним правилам трансформуються в логічні моделі даних.

Задачею логічного етапу проектування є відображення об'єктів ПО в об'єкти моделі даних, що використовується, щоб це відображення не суперечило семантиці предметної області й було за можливістю зручним, ефективним, найкращим.

За допомогою *правил нормалізації* можливо впевнитись в структурній узгодженості, логічній цілісності й мінімальній збитковості прийнятої моделі даних. Також модель проходить перевірку на виявлення можливостей виконання транзакцій, які будуть задаватися користувачами. Проектування являє собою циклічний процес. Етапи логічного проектування наведені на рисунку 9.1.



Рисунок 9.1 - Логічне проектування БД поетапно

### Етап спрощення концептуальної моделі

Спочатку спрощення концептуальної моделі є попередні перетворення з метою видалення зв'язків, які є несумісними з реляційною моделлю.

Цей етап передбачає виконання операцій вилучення:

- двосторонніх зв'язків M:N;
- складних зв'язків;
- багатозначних атрибутів;
- рекурсивних зв'язків;
- зв'язків з атрибутами.

Спрощення концептуальної моделі передбачає також вилучення збиткових зв'язків. Збиткові зв'язки характеризуються тим, що одна і та ж інформація може бути отримана не тільки через них, але і через інші зв'язки.

Після спрощення концептуальна модель має містити тільки наступні елементи:

- об'єкти й атрибути;
- зв'язки типів 1:1 і 1:M;
- суперклас-підклас тип зв'язків.

Завдання:

1. Наведіть основні етапи логічного проектування.
2. Наведіть основну задачу логічного проектування.
3. Які елементи має містити концептуальна модель після етапу спрощення?

## Самостійна робота №10

**Тема:** Перетворення ER-діаграм в реляційні структури

Щоб перетворити ER-модель в реляційну модель даних необхідно слідувати алгоритму певних дій та додержуватись відповідних **правил**:

### 1. Сутності та атрибути

Сутностям відповідають відношення, а кожен атрибут сутності є атрибутом відповідного відношення.

Первинний ключ сильних сутностей стає PRIMARY KEY (PK) відповідного відношення.

Первинний ключ слабких сутностей частково або повністю залежить від ключа сутності власника (декількох власників). Спочатку треба визначити всі РК сутностей власників, а потім визначається РК слабкої сутності.

### 2. Зв'язки

Після перетворення концептуальної моделі залишаються такі типи зв'язків:

1. "один до одного";
2. "один до багатьох";
3. "багато до багатьох"
4. рекурсивні зв'язки;
5. суперклас – підклас.

Можливі варіанти:

Обов'язкова участь для обох сутностей

Обов'язкова участь для однієї сутності

Необов'язкова участь для обох сутностей

Завдання:

1. Наведіть за своїм варіантом приклад перетворення для зв'язку один до одного
2. Наведіть за своїм варіантом приклад перетворення для зв'язку один до багатьох.
3. Наведіть за своїм варіантом приклад перетворення для зв'язку багато до багатьох.

## Самостійна робота №11

### Тема: Нормалізація та денормалізація

Набір відношень логічної моделі БД, що отримали на попередніх етапах, повинен бути перевірений на коректність об'єднання атрибутів у кожному відношенні. Шляхом застосування процедури послідовної нормалізації до кожного відношення. Допомога нормалізації в тому, що модель, що отримується не буде містити протиріч, а збитковість буде мінімальною. В результаті нормалізації атрибути будуть згруповані відповідно до існуючих між ними логічних зв'язків. Якщо виявились відношення, які не відповідають вимогам нормалізації, необхідно повернутися на попередні етапи проектування й перебудувати помилково створені елементи моделі, щоб забезпечити коректність логічної моделі.

Нормальна форма - властивість зв'язку в реляційній моделі даних, що характеризує його з точки зору надмірності, потенційно призводить до логічно помилкових результатів вибірки або зміни даних. Нормальна форма визначається як сукупність вимог, яким має задовільняти відношення.

Під **нормалізацією** розуміють механізм трансформації відносин БД до того виду, що відповідає нормальним формам. Вона призначається для приведення структури БД до стану мінімальної логічної надмірності, та не має на меті збільшення чи зменшення продуктивності роботи або фізичного обсягу бази даних. Нормалізація виконується з метою зменшення потенційної суперечливості збереженої в БД інформації. Як зазначає К. Дейт, загальне призначення процесу нормалізації полягає в наступному:

- виключення деяких типів надмірності;
- усунення деяких аномалій оновлення;
- розробка проекту бази даних, який є досить «якісним» поданням реального світу, інтуїтивно зрозумілий і може служити хорошою основою для подальшого розширення;
- спрощення процедури застосування необхідних обмежень цілісності.

Позбавлення надлишковостей відбувається, зазвичай, за допомогою декомпозиції відносин таким чином, щоб в кожному відношенні зберігалися тільки первинні факти, що не виводяться з інших збережених фактів.

Процес проектування БД з використанням методу нормальних форм є ітераційним і полягає в послідовному переведенні відношень з першої нормальної форми до бінормальних форм більш високого порядку за певними правилами. Кожна наступна нормальна форма обмежує певний тип функціональної залежності, усуває відповідні аномалії при виконанні операцій над відношеннями БД і зберігає властивості попередніх нормальних форм.

Надлишковість даних призводить до збільшення об'єму пам'яті й уповільнення роботи БД. Тому є небажаною обставиною для БД і є наслідком, передусім, дублювання даних. Розрізняють *збиткове* та *незбиткове* дублювання даних. Зовсім лишатись надлишковості немає необхідності, бо в такому випадку буде неможливо підтримувати базу даних як щось ціле. Можна тільки зробити мінімум надлишковості та залишити необхідне дублювання даних.

Дублювання даних викликає проблеми під час виконання операцій (редагування, додавання інформації або вилучення) з базою даних.

Ситуація, що виникає в БД, що призводить до протиріч та/або суттєво ускладнює обробку даних називається **аномаліями**.

Зустрічаються аномалії додавання, вилучення та модифікації.

**Аномалія додавання** може виникнути при додаванні інформації, наприклад, нового студента, тобто необхідно вводити інформацію, яка вже існує в БД: назва факультету, прізвище декана. Крім того неможливо створити нову групу поки не введено студентів, що в ній навчаються.

**Аномалія модифікації** може статись при спробі змінити значення, наприклад, прізвище декана; в такому випадку необхідно переглянути всі кортежі. Якщо осяг БД великий, то це затребує значного часу, крім того можливі помилки при модифікації прізвища, які можуть порушити цілісність БД.

**Аномалія вилучення** виникає при спробі вилучити дані про студента, що в групі доданий один, наприклад. В такому разі зникне інформація про групу.

Застосування декомпозиції відношень допомагає позбутися вищезгаданих аномалій.

Існує наступна послідовність нормальних форм:

- перша нормальна форма (1НФ);
- друга нормальна форма (2НФ);
- третя нормальна форма (3НФ);
- посилена третя нормальна форма, або нормальна форма Бойса-Кодда (БКНФ);
- четверта нормальна форма (4НФ);
- п'ята нормальна форма (5НФ);
- доменно-ключева нормальна форма (ДКНФ);
- шоста нормальна форма (6НФ).

**Денормалізація** – це навмисний процес модифікації реляційної моделі, при якому ступінь нормалізації модифікованого відношення стає нижчим.

Денормалізацію використовують в таких ситуаціях, якщо вже нормалізована БД не задовільняє вимогам, які висуваються до продуктивності роботи системи. Денормалізацію застосовують у наступних випадках:

- при об'єднанні таблиць зі зв'язками "один до одного";
- при дублюванні неключових атрибутів у зв'язках "один до багатьох" (для зменшення кількості з'єднань);
- при дублюванні атрибутів зовнішнього ключа у зв'язках "один до багатьох" (для зменшення кількості з'єднань);
- при дублюванні атрибутів "багато до багато" (для зменшення кількості з'єднань);
- при створенні таблиць з даних, які містяться в інших таблицях;
- при введенні груп полів, які повторюються.

Перед виконанням денормалізації слід визначитись з користю її й негативними наслідками, та якщо переваг більше, то з обережністю застосовувати. До переваг відносять у тому числі збільшення пропускну здатності, рівня задоволеності клієнтів та продуктивності. Слід пам'ятати, про негативні наслідки цього процесу

Завдання:

Згідно індивідуального завдання (додаток 1) необхідно перевірити можливі помилки й недоліки в структурі БД за допомогою нормалізації та усунути їх.

## **Самостійна робота №12**

**Тема:** Фізична організація баз даних

В БД фізична організація даних відповідає за їх зберігання, керування, форми подання й структури даних.

**Фізичне проектування** - це процес визначення характеристик сховища даних та доступу до них в БД. Властивості сховища даних залежать від пристроїв зберігання, засобів доступу до даних, що підтримуються системою та від СКБД. Етап фізичного проектування має на увазі визначення розташування даних на пристроях зберігання та загальної продуктивності системи.

В реляційних БД складні фізичні процеси організації даних приховані від користувача, але вони важливі для продуктивності роботи з БД.

Починається робота з визначення необхідного запису, для знаходження якого викликається диспетчер файлів. Диспетчер файлів визначає сторінку з записом та викликає диспетчер дисків для її отримання. Диспетчер дисків визначає фізичне розташування потрібної сторінки на диску та передає її диспетчеру файлів, а той – СКБД. Головними одиницями операцій обміну системи СКБД - БД є сторінки даних. БД виглядає як набір записів з точки зору СКБД, та як набір сторінок - з точки зору диспетчера файлів. Кожна сторінка пам'яті має унікальний ідентифікатор. Кожен запис зберігається повністю на одній сторінці.

### **Контрольні запитання:**

1. За що відповідає фізична організація даних?
2. Що представляє собою індекс?
3. Який файл називають індексним?
4. Чим відрізняється послідовна, індексно-послідовна й пряма організація файлів?
5. Чому розподіл записів по блоках впливає на швидкість роботи системи?
6. В чому різниця між організацією пошукових структур за первинним ключем і за вторинним ключем?
7. Що представляє собою сторінкова організація даних в СКБД?
8. Що собою являє технологія хешування та яке її застосування в БД?

9. У яких випадках необхідно застосовувати індекси, а у яких краще не використовувати?

## Самостійна робота №13

**Тема:** Засоби для автоматизації проектування баз даних

CASE-засоби (Computer Aided Software Engineering) - це засоби спеціального класу, що покликані вирішувати проблеми автоматизації проектування інформаційних систем. CASE-технологія являє собою сукупність методів проектування ПЗ, а також набір інструментів, що дозволяють візуально зручно моделювати ПО, аналізувати цю модель на всіх стадіях розробки й супроводження ПЗ та створювати додатки відповідно до інформаційних потреб користувачів.

RAD-технології та компонентно-орієнтовані технології

Для швидшої розробки застосувань використовуються **RAD-технології**. Головні ознаки RAD (Rapid Application Development, середовища швидкої розробки застосувань):

- наявність об'єктно-орієнтованої мови програмування;
- застосування візуальних засобів розробки;
- підтримка стандартних протоколів обміну даними між додатками, що дозволяє створювати багаторівневі, незалежні від джерела даних застосування.

Компонентно-орієнтовані технології – це технології, що ґрунтуються на застосування готових програмних компонентів, що попередньо були розроблені. В цій ситуації широко застосовуються бібліотеки класів, а застосування готового модуля виконується за допомогою його інтерфейсу. Специфікації, що визначають інтерфейс, відокремлені від модуля, а внутрішні деталі сховані від користувача. Компоненти постачаються в скомпільованій формі.

Технології штучного інтелекту

Роль штучного інтелекту (ШІ), як технології для проектування БД дуже важлива. Існують різні інструменти на основі ШІ, що допомагають у проектуванні БД.

Розглянемо деякі з них: SuperDuperDB, Taskade AI, PostgreSQL, Airtable AI, Towhee, MongoDB Atlas та інші.

Застосування технологій ШІ при проектуванні та дизайні баз даних допомагають досягати мети, при цьому, полегшує та пришвидшує роботу. Інновації просуваються вперед. Вони дозволяють оптимізувати створення схем БД, роблячи їх більш масштабованими та стійкими до потреб зазначених даних. Такі функції, як генерація на основі ШІ та можливості оптимізації власної схеми, підвищують ефективність робіт.

### **Контрольні запитання:**

1. Які засоби та які технології називають CASE-засобам і CASE-технологіями?
2. Які основні переваги CASE-технологій для розробки інформаційних систем?
3. З яких компонентів складаються інструментальні CASE-засоби?
4. Що являє собою методологія функціонального моделювання?
5. Яка технологія називається RAD-технологією?
6. Яка технологія називається компонентно-орієнтованою?
7. Які відмінності CASE-систем, що базуються на структурному й об'єктно-орієнтованому підходах?
8. Що спільного у CASE-систем, що базуються на структурному й об'єктно-орієнтованому підходах?
9. Що означає поняття репозиторій?
10. Які засоби ШІ застосовують для проектування БД?

## Самостійна робота №14

### Тема: Розподілена обробка даних

Якщо до БД мають доступ одночасно декілька користувачів, то СКБД повинна забезпечувати коректну паралельну роботу всіх користувачів з даними. Розрізняють розподілену обробку і розподілені БД (рисунок 14.1).

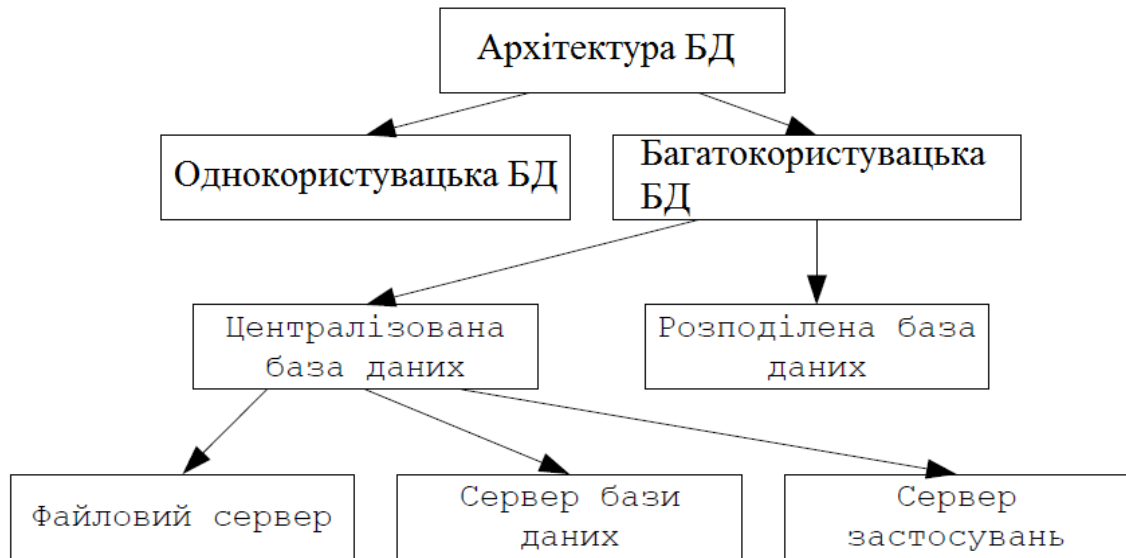


Рисунок 14.1 – Види режимів роботи з БД

**Розподілена обробка** - це робота з обробки даних з використанням централізованої БД, доступ до якої мають різні комп'ютери мережі. В цій системі одні вузли (комп'ютери) - клієнти, а інші - сервери.

**Сервер** - комп'ютер, що надає деякі послуги іншим комп'ютерам, обмін повідомленнями з якими відбувається за допомогою мережі.

**Клієнт** – це комп'ютер чи процес, що надсилає запит на обслуговування.

**Розподілена БД** - це набір логічно зв'язаних між собою роздільних даних і їх описів, які фізично розподілені в мережі.

Тобто, розподілена БД застосовується на різних просторово розосереджених обчислювальних засобах, разом з організаційними, технічними й програмними засобами її створення й обслуговування.

В дійсності розподілена БД є віртуальною БД, компоненти якої фізично зберігаються на декількох різних реальних БД на декількох різних вузлах.

**Розподілена СКБД** – це програмна система, яка призначена для керування розподіленими БД і яка забезпечує прозорий доступ користувачів до інформації.

**Гомогенною** (однорідною) називається розподілена система, в якій всі вузли використовують той самий тип СКБД.

**Гетерогенною** (різномірною) називається розподілена система, в якій всі вузли використовують різні типи СКБД, які обробляють різні моделі даних.

### **Керування паралельною обробкою**

В багатокористувацьких системах до БД одночасно мають доступ декілька користувачів або прикладних програм. Виникає необхідність захисту БД від можливих випадкових чи спланованих ситуацій, коли існує вирогідність втрати даних. Для збереження цілісності даних і забезпечення безпеки в цих умовах застосовуються транзакції, які забезпечують роботу кожного користувача з узгодженим станом БД. Підтримка механізму транзакцій – показник рівня розвинутості СКБД.

**Транзакція** - послідовність операторів маніпулювання даними (читання, видалення, вставки, модифікації), яка розглядається СКБД, як одне ціле. Транзакція або успішно виконується, і СКБД фіксує зміни БД, які були виконані транзакцією, у зовнішній пам'яті, або, у разі невдачі, жодна зміна не відбувається у БД.

Щоб механізм транзакцій забезпечував цілісність даних й ізолюваність користувачів, транзакція повинна мати такі властивості: атомарність (Atomicity), узгодженість (Consistency), ізолюваність (Isolation), довготерміновість (Durability). Транзакції, які мають ці властивості називаються ACID-транзакціями:

-Атомарність або нерозривність означає, що транзакція виконується, як єдина операція доступу до БД і виконується або повністю або не виконується зовсім (за забезпечення нерозривності відповідає підсистема відновлення СКБД);

-Узгодженість гарантує взаємну цілісність даних, тобто виконання обмежень цілісності БД після завершення роботи транзакції (відповідальність за забезпечення властивості узгодженості покладається на СКБД та на розробників додатку);

-Ізолюваність можна означити так, що транзакції, які конкурують за

доступ до БД, проходять послідовну фізичну обробку, ізольовано одна від одної, але мають виглядає для користувачів, начебто, паралельно виконуються;

-Довготерміновість або стійкість означає, що коли транзакція виконана успішно, то всі зміни, які вона зробила в даних, не будуть втрачені ні за яких обставин (за забезпечення стійкості відповідає підсистема відновлення).

Для паралельних транзакції застосовується серіалізація транзакцій і метод тимчасових міток.

**Серіалізація транзакцій** - це процедура, яка забезпечує підтримку незалежного виконання транзакцій. Іншими словами, дія двох паралельно діючих транзакцій буде як і їх послідовна дія: спочатку одна, а потім інша, або навпаки, спочатку друга, а потім перша. У ході виконання транзакції користувач видно тільки узгоджені дані, а неузгоджені проміжні залишаються непомітними. Для підтримки паралельної роботи складається спеціальний план.

#### **Контрольні запитання:**

1. Що означає термін «Розподілена обробка»?
2. Що означає термін «Розподілена БД»?
3. Що означає термін «Розподілена СКБД»?
4. Які системи називають Гомогенними?
5. Які системи називають Гетерогенними?
6. Який комп'ютер називають сервером?
7. Що називають клієнтом при роботі з БД?
8. Що називають транзакцією?
9. Які властивості повинна мати транзакція, щоб механізм транзакцій забезпечував цілісність даних й ізольованість користувачів?
10. Яку процедуру називають серіалізація транзакцій?
11. Для яких цілей застосовують механізм блокувань при роботі з БД?
11. Яким чином визначається порядок у виконанні операцій транзакцій?
12. З яких компонентів можуть складатись розподілені СКБД?
13. Які системи називають мультибазовими системами?
14. Які розподілені системи називають зосередженими?

15. Які розподілені системи називають розосередженими?
16. Які основні правила організації розподілених БД?
17. Які існують інтерфейси, що застосовуються для доступу до БД?
18. Що означає реплікація БД?

## Додаток №1

### Теми індивідуальної роботи

№	Назва, зміст та обсяг завдань для індивідуальної роботи	Прим
1	<p>Склад бази даних «Інститут»: таблиця «НДІ», таблиця «Кадри», таблиця «Довідник відділів», таблиця «Довідник тем».</p> <p>Структура таблиці «НДІ»: номер відділу; табельний номер; номер теми, над якою працює працівник; тривалість роботи над темою (у місяцях). <b>Примітка:</b> Одну тему можуть розробляти працівники кількох відділів.</p> <p>Структура таблиці «Кадри»: номер відділу; табельний номер; прізвище та ініціали; код посади; розмір зарплати.</p> <p>Структура таблиці «Довідник відділів»: номер відділу; назва відділу; табельний номер керівника.</p> <p>Структура таблиці «Довідник тем»: номер теми; назва теми; дата початку розробки; дата завершення роботи; табельний номер керівника.</p> <p>Використовуючи інформаційно-логічну модель:</p> <p>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</p> <p>б) визначити ключові реквізита та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</p> <p>в) побудувати форми для перегляду та заповнення таблиць даними.</p> <p>Використовуючи зазначену базу даних та засоби мови SQL:</p> <p>1. Визначити теми (у порядку зростання їх номерів), що розробляються в НДІ, кількість зайнятих працівників та суми коштів, які витрачені на зарплату.</p> <p>2. Отримати інформацію про працівників, зайнятих у розробці тем різних відділів, та визначити загальну суму зарплати, яка їм сплачується.</p> <p>3. Отримати інформацію про чисельність працівників у відділах та працюючих над окремими темами у розрізі посад.</p> <p>4. Отримати інформацію про середній зарібок працівників різних відділів.</p> <p>5. Визначити обсяг робіт (кількість людино-місяців) за темами та відділами.</p> <p>6. Отримати інформацію про тривалість роботи працівників за темами.</p> <p>7. Визначити в цілому по НДІ й окремо по кожній темі чисельність зайнятих працівників та суму коштів, витрачених на зарплату.</p> <p>8. Визначити чисельність працівників кожного відділу у розрізі посад, які вони займають.</p> <p>9. Визначити загальну тривалість роботи над кожною темою (у людино-місяцях) з виділенням даних по відділах та в цілому по НДІ.</p> <p>10. Скласти список працівників по відділах, тривалість роботи яких над темою не перевищує заданої користувачем величини, та підрахувати їх кількість по відділах та НДІ в цілому.</p> <p>Примітка: Тут і далі «*» означає спрощений варіант задачі.</p>	

2	<p>Склад бази даних «Матеріали»: таблиця «Замовлення», таблиця «Ліміт», таблиця «Довідник підприємств», таблиця «Прейскурант».</p> <p>Структура таблиці «Замовлення»: дата замовлення; код підприємства; номенклатурний номер матеріалу, який замовляється.</p> <p>Примітка: У таблиці містяться відомості за поточний місяць.</p> <p>Структура таблиці «Ліміт»: номенклатурний номер матеріалу; розмір ліміту на місяць.</p> <p>Структура таблиці «Довідник підприємств»: код підприємства; назва.</p> <p>Структура таблиці «Прейскурант»: номенклатурний номер матеріалу; назва матеріалу; код одиниці вимірювання; ціна за одиницю.</p> <p><b>Використовуючи інформаційно-логічну модель:</b></p> <p>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</p> <p>б) визначити ключові реквізита та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</p> <p>в) побудувати форми для перегляду та заповнення таблиць даними.</p> <p><b>Використовуючи зазначену базу даних та засоби мови SQL</b></p> <ol style="list-style-type: none"> <li>1. Отримати інформацію про потреби підприємства в матеріалах.</li> <li>2. Визначити сумарну потребу в матеріалах по кожному номенклатурному номеру та співвідношення потреби і ліміту.</li> <li>3. Отримати інформацію по підприємствах, які надіслали замовлення на визначені види матеріалу в конкретний період часу.</li> <li>4. Отримати інформацію про задоволення замовлень підприємств.</li> <li>5. Визначити кількість замовлень, що надійшли від підприємств на визначені види матеріалів в різні періоди часу.</li> <li>6. Визначити сумарну потребу в матеріалах кожного підприємства та загальну потребу всіх підприємств по кожному номенклатурному номеру.</li> <li>7. Визначити перелік підприємств, які надіслали замовлення не пізніше дати, що цікавить користувача.</li> <li>8. Визначити кількість замовлень, які надійшли від кожного підприємства на визначений матеріал за звітний період та загальну кількість замовлень на нього.</li> </ol>	
3	<p>Склад бази даних «Торгівля»: таблиця «Товар», таблиця «Магазибн», таблиця «Прейскурант».</p> <p>Структура таблиці «Товар»: номер магазину, номер секції; номер чека; дата продажу; артикул товару; кількість одиниць товару; ціна за одиницю; загальна вартість.</p> <p>Примітка: У таблиці містяться відомості за звітний місяць, реквізити, ціна за одиницю та загальна вартість. Вони використовуються для контролю за введенням даних. Структура таблиці «Магазин»: номер магазину; назва магазину. Структура таблиці «Прейскурант»: артикул товару; назва товару; одиниця вимірювання; ціна за одиницю.</p> <p><b>Використовуючи інформаційно-логічну модель:</b></p> <p>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</p>	

	<p>б) визначити ключові реквізита та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</p> <p>в) побудувати форми для перегляду та заповнення таблиць даними.</p> <p><b>Використовуючи зазначену базу даних та засоби мови SQL:</b></p> <ol style="list-style-type: none"> <li>1. Визначити обсяг товарообігу магазинів за відповідний період часу.</li> <li>2. Визначити обсяг товарообігу секцій магазинів за відповідний період часу.</li> <li>3. Визначити виручку від реалізації відповідного виду товару в різних магазинах.</li> <li>4. Отримати інформацію про обсяг денної виручки магазинів,</li> <li>5. Отримати інформацію про динаміку денної виручки магазинів протягом відповідного періоду (відрізку) часу.</li> <li>6. Визначити магазини з найкращими та найгіршими показниками обсягу товарообігу.</li> <li>7. Отримати інформацію про секції магазинів, які забезпечать найбільший та найменший обсяг реалізації відповідного виду товару.</li> <li>8. Отримати інформацію про виконання плану товарообігу магазинами.</li> <li>9. Отримати інформацію про магазини, які перевиконали та не виконали план товарообігу.</li> <li>10. Визначити обсяг товарообігу по управлінню в цілому та по кожному магазину за період, що минув.</li> <li>11. Визначити обсяг товарообігу магазинів в цілому та по кожній секції за відповідний день.</li> <li>12. Визначити обсяг продажу відповідних товарів по кожному магазину та по всіх магазинах в цілому.</li> <li>13. Визначити магазини, які забезпечують найбільший і найменший обсяг денної виручки та обсяг виручки по кожній секції в цих магазинах.</li> <li>14. Визначити виконання плану (у відсотках) по кожному магазину та в цілому по управлінню.</li> </ol>	
4	<p>Склад бази даних «Фабрика»: таблиця «Замовлення», таблиця «Фабрика», таблиця «Потужність».</p> <p>Структура таблиці «Замовлення»: номер фабрики-пральні; номер квитанції; дата прийому; дата виконання; вага; вартість.</p> <p><b>Примітка:</b> У таблиці містяться відомості за поточний місяць.</p> <p>Структура таблиці «Фабрика»: номер фабрики-пральні; район міста; адреса.</p> <p>Структура таблиці «Потужність»: номер фабрики-пральні; пропускна спроможність (кг за добу).</p> <p><b>Використовуючи інформаційно-логічну модель:</b></p> <ol style="list-style-type: none"> <li>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</li> <li>б) визначити ключові реквізита та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</li> <li>в) побудувати форми для перегляду та заповнення таблиць даними.</li> </ol>	

	<p><b>Використовуючи зазначену базу даних та засоби мови SQL:</b></p> <ol style="list-style-type: none"> <li>1. Отримати інформацію про порушення термінів виконання замовлень.</li> <li>2. Визначити обсяг виконаних робіт.</li> <li>3. Визначити дні, у які фабрика має найбільшу і найменшу завантаженість.</li> <li>4. Отримати інформацію про фабрики з максимальною та мінімальною кількістю порушень термінів виконання замовлень.</li> <li>5. інформацію про використання потужності фабрик.</li> <li>6. Визначити кількість порушень термінів виконання замовлень по кожній фабриці та загальну кількість по всіх фабриках міста.</li> <li>7. Визначити перелік фабрик, на яких кількість порушень термінів виконання замовлення не перевищує заданої величини.</li> <li>8. Визначити обсяг денного приймання кожної фабрики та кількість фабрик, на яких цей обсяг не досягає визначеної величини.</li> </ol>	
5	<p>Склад бази даних «Виробництво»: таблиця «Наряд», таблиця «Довідник кадрового складу», таблиця «Довідник операцій», таблиця «Довідник цехів».</p> <p>Структура таблиці «Наряд»: номер наряду; номер цеху; номер ланки; табельний номер робітника; код операції; розряд роботи; кількість виготовлених деталей; кількість прийнятих деталей.</p> <p>Примітка: У таблиці містяться відомості за поточний місяць.</p> <p>Структура таблиці «Довідник кадрового складу»: номер цеху; табельний номер робітника, прізвище та ініціали.</p> <p>Структура таблиці «Довідник операцій»: код операції; назва операції; норма часу на виготовлення деталі (хв.). Структура таблиці «Довідник цехів»: номер цеху; назва цеху.</p> <p><b>Використовуючи інформаційно-логічну модель:</b></p> <ol style="list-style-type: none"> <li>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</li> <li>б) визначити ключові реквізита та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</li> <li>в) побудувати форми для перегляду та заповнення таблиць даними.</li> </ol> <p><b>Використовуючи зазначену базу даних та засоби мови SQL:</b></p> <ol style="list-style-type: none"> <li>1. Отримати інформацію про обсяг браку на виробництві.</li> <li>2. Отримати інформацію про рівень браку на різних операціях.</li> <li>3. Отримати інформацію про результати роботи конкретного робітника.</li> <li>4. Отримати інформацію про втрати часу, пов'язані з виробництвом бракованих деталей.</li> <li>5. Отримати інформацію щодо робітників, які допустили випуск бракованих деталей на визначених операціях.</li> <li>6. Визначити перелік операцій в кожному цеху, на яких допущено випуск найбільшої чи найменшої кількості бракованих деталей.</li> <li>7. Визначити кількість бракованих деталей в цілому по заводу та по кожному цеху.</li> </ol>	

	<p>8. Визначити кількість бракованих деталей, випущених робітниками, табельні номери яких знаходяться в заданому інтервалі.</p> <p>9. Скласти список робітників, які допустили випуск бракованих деталей на операції із зазначеним кодом по всіх цехах, де ця операція виконується.</p>	
6	<p>Склад бази даних «Простої»: таблиця «Простій», таблиця «Робітник», таблиця «Довідник цехів», таблиця «Довідник причин простою».</p> <p>Структура таблиці «Простій»: номер документа; дата; номер цеху; номер ланки; табельний номер робітника; код причини і винуватця простою; % оплати; тривалість простою (хв.).</p> <p>Примітка: У таблиці містяться відомості за поточний місяць.</p> <p>Структура таблиці «Робітник»: код цеху; табельний номер; прізвище та ініціали; середня заробітна плата за місяць. Структура таблиці «Довідник цехів»: номер цеху; назва цеху.</p> <p>Структура таблиці «Довідник причин простою»: код причини чи винуватця простою; назва.</p> <p>Використовуючи інформаційно-логічну модель:</p> <p>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</p> <p>б) визначити ключові реквізита та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</p> <p>в) побудувати форми для перегляду та заповнення таблиць даними.</p> <p>Використовуючи зазначену базу даних та засоби мови SQL:</p> <ol style="list-style-type: none"> <li>1. Отримати інформацію про тривалість простоїв у цехах.</li> <li>2. Отримати інформацію про тривалість простоїв за кодами причин та винуватців.</li> <li>3. Отримати інформацію про простої, що перевищили відповідну величину.</li> <li>4. Отримати інформацію про найбільшу та найменшу тривалість простоїв.</li> <li>5. Отримати інформацію про тривалість простоїв окремих робітників.</li> <li>6. Визначити величину втрат, викликаних оплатою простоїв.</li> <li>7. Отримати інформацію про динаміку простоїв протягом відповідного періоду часу.</li> <li>8. Визначити питому вагу простоїв у загальному фонді робочого часу.</li> <li>9. Визначити тривалість простоїв в цілому по заводу та по кожному цеху.</li> <li>10. Визначити тривалість простоїв по кожному цеху за кодами причин та винуватців.</li> <li>11. Скласти список цехів, у яких тривалість простоїв за день не перевищує задану величину.</li> <li>12. Визначити ділянки кожного цеху, на яких тривалість простоїв була найбільшою чи найменшою.</li> <li>13. Визначити суму втрат, викликаних оплатою простоїв по кожному цеху та заводу в цілому.</li> </ol>	

7

Склад бази даних «Студент»: таблиця «Студент», таблиця «Довідник факультетів».

Структура таблиці «Студент»: код факультету; курс; вік; код статі; код родинного стану; середній дохід сім'ї; час на дорогу до інституту (хв.).

Структура таблиці «Довідник факультетів»: код факультету; назва факультету.

Використовуючи інформаційно-логічну модель:

а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;

б) визначити ключові реквізита та побудувати схему даних, яка відповідає інформаційно-логічній моделі;

в) побудувати форми для перегляду та заповнення таблиць даними.

Використовуючи зазначену базу даних та засоби мови SQL:

1. Отримати інформацію про час (в середньому), який студенти витрачають на дорогу до університету.
2. Отримати інформацію про віковий склад студентів.
3. Отримати інформацію про чисельний склад студентів.
4. Визначити середній дохід в сім'ях студентів.
5. Отримати інформацію про студентів, середній прибуток в сім'ях яких не досягає заданої користувачем величини.
6. Отримати інформацію про студентів, вік яких перевищує заданий користувачем рівень.
7. Отримати інформацію про сімейний стан студентів.
8. Визначити середній дохід студентів, які перебувають у шлюбі.
9. Визначити чисельний склад студентів, які перебувають у шлюбі.
10. Визначити віковий склад студентів, які перебувають у шлюбі.
11. Визначити питому вагу студенток у загальній кількості студентів.
12. Отримати інформацію про середній дохід в сім'ях студентів, які належать до різних вікових груп.
13. Отримати інформацію про співвідношення студентів та студенток, які перебувають у шлюбі.
14. Визначити середній час, який витрачають студенти кожного факультету на дорогу до університету.
15. Визначити віковий склад студентів у цілому по університету та по кожному факультету за віковими групами: до 25 років; 25-30 років; після 30 років.
16. Визначити кількість студентів та студенток на кожному факультеті та по університету в цілому.
17. Визначити середній дохід в сім'ях студентів на кожному факультеті з виділенням даних по курсах.
18. Визначити кількість студентів на кожному курсі факультету, які мають мінімальний та максимальний вік.
19. Визначити кількість студентів, що перебувають у шлюбі, по університету в цілому та по кожному факультету.

	<p>20. Визначити віковий склад студентів, які перебувають у шлюбі, по кожному факультету та курсу.</p> <p>21. Визначити середній дохід студентів, які перебувають у шлюбі, по кожному курсу університету.</p> <p>22. Визначити чисельність студентів та студенток на кожному курсі та по університету в цілому (за віковими групами: до 20 років; 20—25 років; після 25 років).</p> <p>23. Визначити середній дохід студентів різних вікових груп по університету (за віковими групами: до 20 років; 20-25 років; після 25 років).</p>	
8	<p>Склад бази даних «Обстеження»: таблиця «Анкета», таблиця «Довідник рівнів освіти».</p> <p>Структура таблиці «Анкета»: рік обстеження; код групи обстеження; код рівня освіти; код родинного стану; заробітна плата; кількість дітей.</p> <p>Структура таблиці «Довідник рівнів освіти»: код рівня освіти; повна назва.</p> <p><b>Використовуючи інформаційно-логічну модель:</b></p> <p>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</p> <p>б) визначити ключові реквізита та побудувати схему даних, яка відповідає інформаційно-логічній моделі нарахованої і виданої кожному працівникові цеху, який цікавить користувача.</p> <ol style="list-style-type: none"> <li>1. Отримати інформацію про суму прибуткового податку, який утримано із заробітної плати працюючих.</li> <li>2. Отримати інформацію про питому вагу прибуткового податку в загальній сумі утримань.</li> <li>3. Отримати інформацію про динаміку заробітної плати.</li> <li>4. Отримати інформацію про питому вагу суми утримань в нарахованій заробітній платі.</li> <li>5. Визначити межові значення нарахованої заробітної плати.</li> <li>6. Отримати інформацію про працюючих, що мають заробітну плату (суму до виплати) нижче відповідної величини.</li> <li>7. Отримати інформацію про питому вагу робітників, що перераховують заробітну плату до ощадних кас, та про суму перерахувань.</li> <li>8. Визначити суму зарплати, виданої працюючим кожного цеху за місяць.</li> <li>9. Визначити заробітну плату, нараховану і виплачену кожному робітникові цеху, що цікавить користувача, та загальну суму по цеху.</li> <li>10. Визначити суму прибуткового податку, утриманого із зарплати працюючих в цілому по підприємству та по кожному цеху.</li> <li>11. Визначити питому вагу суми утримань в нарахованій зарплаті кожного робітника цеху, що цікавить користувача, та в цілому по цеху.</li> </ol>	

	<p>12. Визначити найбільшу і найменшу суму зарплати, нарахованої по кожному цеху та по підприємству в цілому.</p> <p>13. Скласти список робітників по кожному цеху, що мають зарплату менше вказаної величини, та визначити їх кількість по кожному цеху.</p>	
9	<p>Склад бази даних «Успішність»: таблиця «Сесія», таблиця «Довідник факультетів», таблиця «Довідник спеціальностей», таблиця «Довідник дисциплін».</p> <p>Структура таблиці «Сесія»: код факультету; код спеціальності; курс; група; номер залікової книжки; код дисципліни; оцінка.</p> <p>Таблиці «Довідник факультетів», «Довідник спеціальностей», «Довідник дисциплін» мають однакову структуру: код реквізиту; назва реквізиту.</p> <p><b>Використовуючи інформаційно-логічну модель:</b></p> <p>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</p> <p>б) визначити ключові реквізита та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</p> <p>в) побудувати форми для перегляду та заповнення таблиць даними.</p> <p><b>Використовуючи зазначену базу даних та засоби мови SQL:</b></p> <p>1. Отримати інформацію про кількість відмінних та незадовільних оцінок, виставлених в період сесії.</p> <p>2. Отримати інформацію про студентів-відмінників та про їх кількість.</p> <p>3. Отримати інформацію про студентів, які мають дві і більше оцінок «незадовільно» за результатами сесії.</p> <p>4. Визначити питому вагу студентів, що мають незадовільні оцінки, в загальній кількості студентів.</p> <p>5. Отримати інформацію про дисципліни, з яких на іспитах було виставлено найбільшу і найменшу кількість оцінок «відмінно» та «добре».</p> <p>6. Отримати інформацію за результатами сесії конкретних студентів.</p> <p>7. Визначити кількість оцінок «відмінно» і «незадовільно», отриманих в період сесії в цілому по університету та з кожної спеціальності.</p> <p>8. Скласти список студентів-відмінників з кожної спеціальності та визначити кількість студентів-відмінників в цілому по університету.</p> <p>9. Визначити кількість студентів, що мають дві і більше оцінок «незадовільно» за результатами сесії, з кожної спеціальності з переліком прізвищ студентів та дисциплін, з яких отримані незадовільні оцінки, а також визначити загальну кількість таких студентів в цілому по університету.</p> <p>10. Визначити питому вагу студентів, які мають оцінки «незадовільно» в загальній кількості студентів з кожної спеціальності та в цілому по університету.</p> <p>11. Визначити дисципліни, з яких на іспитах було виставлено найбільшу і найменшу кількість оцінок «відмінно» та «добре».</p>	
10	<p>Склад бази даних «Виробництво»: таблиця «Напої», таблиця «Довідник заводів», таблиця «План».</p>	

	<p>Структура таблиці «Напої»: дата звітного періоду; код заводу; обсяг виробництва: соків; газових напоїв; мінеральної води; квасу.</p> <p>Структура таблиці «Довідник заводів»: код заводу; назва заводу.</p> <p>Структура таблиці «План»: код заводу; плановий обсяг виробництва всього, у тому числі: соків; газових напоїв; мінеральної води; квасу.</p> <p><b>Використовуючи інформаційно-логічну модель:</b></p> <p>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</p> <p>б) визначити ключові реквізита та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</p> <p>в) побудувати форми для перегляду та заповнення таблиць даними.</p> <p><b>Використовуючи зазначену базу даних та засоби мови SQL:</b></p> <ol style="list-style-type: none"> <li>Отримати інформацію про обсяг виробництва безалкогольних напоїв.</li> <li>Отримати інформацію про питому вагу виробництва тих напоїв, що цікавлять користувача, в загальному обсязі виробництва.</li> <li>Отримати інформацію про виконання плану щодо виробництва безалкогольних напоїв.</li> <li>Отримати інформацію про заводи, на яких обсяг виробництва того напою, що цікавить користувача, не перевищує заданої величини.</li> <li>Отримати інформацію про динаміку виробництва безалкогольних напоїв.</li> <li>Отримати інформацію про максимальний та мінімальний рівень виробництва безалкогольних напоїв.</li> <li>Визначити обсяг виробництва кожного виду напою заводами міста та загальний обсяг виробництва безалкогольних напоїв.</li> <li>Визначити питому вагу виробництва кожного виду напою в загальному обсязі виробництва по місту.</li> <li>Визначити заводи, які досягли максимального виробництва кожного виду напою.</li> <li>Визначити перелік заводів, на яких обсяг виробництва мінеральної води не перевищує заданої величини.</li> </ol>	
11	<p>Склад бази даних «Кадри»: таблиця «Кадрові відомості», таблиця «Довідник підрозділів», таблиця «Довідник категорій», таблиця «Довідник посад», таблиця «Довідник рівнів освіти».</p> <p>Структура таблиці «Кадрові відомості»: код підрозділу; табельний номер робітника; прізвище та ініціали; код категорії; код посади; розряд; дата прийому на завод; загальний стаж роботи; стаж роботи на заводі; рік народження; код статі; код рівня освіти; код родинного стану;</p> <p>домашня адреса; середня заробітна плата.</p> <p>Таблиці «Довідник підрозділів», «Довідник категорій», «Довідник посад», «Довідник рівня освіти» мають однакову структуру: код реквізиту; назва реквізиту.</p>	

	<p><b>Використовуючи інформаційно-логічну модель:</b></p> <p>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</p> <p>б) визначити ключові реквізита та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</p> <p>в) побудувати форми для перегляду та заповнення таблиць даними.</p> <p><b>Використовуючи зазначену базу даних та засоби мови SQL:</b></p> <ol style="list-style-type: none"> <li>1. Отримати інформацію про прийняття робітників на завод.</li> <li>2. Отримати інформацію про освітній рівень робітників.</li> <li>3. Отримати інформацію про стаж працюючих.</li> <li>4. Отримати інформацію про сімейний стан працюючих.</li> <li>5. Отримати інформацію про склад посад працюючих.</li> <li>6. Отримати інформацію про віковий склад працюючих.</li> <li>7. Отримати інформацію про місячний фонд заробітної плати.</li> <li>8. Отримати інформацію про співвідношення працюючих різних категорій.</li> <li>9. Отримати інформацію про межові значення розмірів середньої заробітної плати.</li> <li>10. Отримати інформацію про чисельність працюючих в структурних підрозділах.</li> <li>11. Отримати інформацію про відповідність освітнього рівня працюючих займаним посадам.</li> <li>12. Отримати інформацію про працюючих пенсіонерів.</li> <li>13. Отримати інформацію про "середню заробітну плату різних вікових груп працюючих.</li> <li>14. Отримати інформацію про чисельність працюючих чоловіків та жінок по різних категоріях.</li> </ol>	
12	<p>Склад бази даних «Навчальний процес»: таблиця «Практика», таблиця «Довідник факультетів», таблиця «Довідник дисциплін», таблиця «Довідник рівня завершеності».</p> <p>Структура таблиці «Практика»: код факультету; курс; код залікової книжки студента; код дисципліни; номер лабораторної роботи з дисципліни; дата видачі завдання; термін завершення роботи; код рівня завершеності; кількість виходів на машину.</p> <p>Таблиці «Довідник факультетів», «Довідник дисциплін», «Довідник рівня завершеності» мають однакову структуру: код реквізиту; назва реквізиту.</p> <p><b>Використовуючи інформаційно-логічну модель:</b></p> <p>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</p> <p>б) визначити ключові реквізита та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</p> <p>в) побудувати форми для перегляду та заповнення таблиць даними.</p> <p><b>Використовуючи зазначену базу даних та засоби мови SQL:</b></p> <ol style="list-style-type: none"> <li>1. Отримати інформацію про рівень завершеності визначеної лабораторної роботи.</li> </ol>	

	<p>2. Отримати інформацію про середню кількість виходів на машину під час виконання визначеної лабораторної роботи.</p> <p>3. Отримати інформацію про питому вагу налагоджених програмна певну дату.</p> <p>4. Отримати інформацію про завантаженість студентів лабораторними роботами.</p> <p>5. Отримати інформацію про хід виконання графіка лабораторних робіт.</p> <p>6. Визначити кількість студентів у кожній групі по курсу, заданому користувачем, які виконали чергову лабораторну роботу й підготували текст програми до наступної роботи.</p> <p>7. Визначити середню кількість виходів на машину студентів певної групи під час виконання лабораторної роботи з дисципліни, що цікавить користувача.</p> <p>8. Визначити кількість лабораторних робіт, які виконуються студентами факультету, що цікавить користувача, за певний період.</p> <p>9. Визначити кількість лабораторних робіт, які виконуються на ЕОМ студентами певного факультету на кожному курсі.</p>	
13	<p>Склад бази даних «Моя колекція музики»: таблиця «Групи», таблиця «Альбоми», таблиця «Треки».</p> <p>Структура таблиці «Групи»: група (ключове поле); кількість альбомів; країна; стиль.</p> <p>Структура таблиці «Альбоми»: альбом; група; число записів; час звучання; рік випуску; примітки; ідентифікатор (ключове поле).</p> <p>Структура таблиці «Треки»: номер запису; трек; альбом; час звучання; Kbps (Кб/с); частота дискретизації (Гц); якість; розмір; формат файлу; замітки; ідентифікатор; ідентифікатор альбому.</p> <p><b>Використовуючи інформаційно-логічну модель:</b></p> <p>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</p> <p>б) Визначити ключові реквізити та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</p> <p>в) побудувати форми для перегляду та заповнення таблиць даними.</p> <p><b>Використовуючи зазначену базу даних та засоби мови SQL:</b></p> <ol style="list-style-type: none"> <li>1. Вивести інформацію по всіх групах.</li> <li>2. У визначеному альбомі визначити найкоротший трек.</li> <li>3. Вивести інформацію про те, якій групі належить який альбом.</li> <li>4. Вивести інформацію про групи.</li> <li>5. Вивести інформацію про «наймолодший» альбом.</li> <li>6. Вивести інформацію про «найстарший» альбом.</li> <li>7. Визначити загальну довжину звучання вказаного альбому.</li> <li>8. Вивести назви груп у порядку від найстаршої до наймолодшої.</li> </ol>	

	<p>9. Керуючись інформацією про треки, виведіть 10 найкращих робіт. 10.Провівши аналіз по базі даних дайте висновок про найпопулярнішу групу і чому.</p>	
14	<p>Склад бази даних «Відділ кадрів»: таблиця «Кадри», таблиця «Посади», таблиця «Штатний розклад», таблиця «Стан штату», таблиця «Планові робочі дні».</p> <p>Структура таблиці «Кадри»;табельний номер; ПІБ; дата народження; стать; ідентифікаційний код; паспортні дані; адреса; освіта; форма працевлаштування; підрозділ.</p> <p>Структура таблиці «Посади»: код посади; посада; тариф оплати за 1 год.; надбавка.</p> <p>Структура таблиці «Штатний розклад»: код посади; код підрозділу, кількість штатних одиниць, верхня межа окладу, нижня межа окладу.</p> <p>Структура таблиці «Стан штату»: табельний номер; код посади; код підрозділу, рік, місяць, код форми працевлаштування, оклад, кількість відпрацьованих днів.</p> <p>Структура таблиці «Планові робочі дні»: рік, місяць, кількість робочих днів.</p> <p><b>Використовуючи інформаційно-логічну модель:</b></p> <p>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</p> <p>б) Визначити ключові реквізити та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</p> <p>в) побудувати форми для перегляду та заповнення таблиць даними.</p> <p><b>Використовуючи зазначену базу даних та засоби мови SQL:</b></p> <ol style="list-style-type: none"> <li>Отримати інформацію по наймолодшому працівнику.</li> <li>Вивести інформацію по працівникам з вищою освітою.</li> <li>Отримати інформацію по всім працюючим на підприємстві.</li> <li>Отримати інформацію про чисельність працюючих чоловіків та жінок окремо.</li> <li>Вивести інформацію по одній вказаній посаді.</li> <li>Користуючись інформацією по вказаній базі даних скласти вікову діаграму працівників підприємства.</li> <li>Вказати кількість хворих за поточний місяць.</li> <li>Скласти список робітників по кожній посаді, що мають зарплату менше вказаної величини.</li> <li>Визначити суму прибуткового податку, утриманого з зарплати працюючих в цілому по підприємству.</li> <li>Отримати інформацію про працюючих, що мають заробітну плату нижче відповідної величини.</li> </ol>	
15	<p>Склад бази даних «Абітурієнт»: таблиця «Абітурієнти», таблиця «Паспортні дані», таблиця «Відомості про освіту», таблиця «Результати вступних іспитів».</p>	

	<p>Структура таблиці «Абітурієнти»: порядковий номер (ключове поле), ШБ, дата народження, дата подачі документів, спеціальність, основа навчання, форма навчання.</p> <p>Структура таблиці «Паспортні дані»: порядковий номер (ключове поле), серія, номер, ким виданий, дата видачі, сімейний стан.</p> <p>Структура таблиці «Результати вступних іспитів»: фізика, математика, диктант, порядковий номер (ключове поле). Використовуючи інформаційно-логічну модель:</p> <p>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</p> <p>б) Визначити ключові реквізити та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</p> <p>в) побудувати форми для перегляду та заповнення таблиць даними.</p> <p><b>Використовуючи зазначену базу даних та засоби мови SQL:</b></p> <p>вказаної літери.</p> <p>7. Визначити, який товар замовив вказаний клієнт.</p> <p>8. Видалити усіх клієнтів, що не зробили жодного замовлення, але перебувають у списку.</p> <p>9. Додати до переліку товарів на складі будь-який новий товар.</p> <p>10. Після видалення прізвищ клієнтів, поновити ці дані.</p>	
16	<p>Склад бази даних «Бібліотека»: таблиця «Бібліотекарі», таблиця «Каталог», таблиця «Формуляр читача», таблиця «Читальний зал», таблиця «Читач».</p> <p>Структура таблиці «Бібліотекарі»: особистий номер (ключове поле), ШБ, дата народження, освіта, адреса, телефон, паспортні дані, коли і ким виданий.</p> <p>Структура таблиці «Каталог»: номер відділу, назва відділу, інвентарний номер (ключове поле), автор, заголовок.</p> <p>Структура таблиці «Формуляр читача»: № формуляра (ключове поле), дата видачі, інвентарний номер, дата здачі, видав, прийняв.</p> <p>Структура таблиці «Читальний зал»: № п/п (ключове поле), № формуляру, дата видачі, час видачі, інвентарний номер, час здачі, число здачі, сума, сплачено, видав, прийняв.</p> <p>Структура таблиці «Читач»: № формуляру (ключове поле), ПІБ, дата народження, освіта, професія, місце роботи, навчальний заклад, домашня адреса, телефон, паспортні дані, ким і коли виданий, читач бібліотеки з (число).</p> <p><b>Використовуючи інформаційно-логічну модель:</b></p> <p>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</p> <p>б) Визначити ключові реквізити та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</p> <p>в) побудувати форми для перегляду та заповнення таблиць даними.</p> <p><b>Використовуючи зазначену базу даних та засоби мови SQL:</b></p> <p>1. Вивести дані по всіх працівниках бібліотеки.</p>	

	<ol style="list-style-type: none"> <li>2. Вивести дані по усіх боржниках.</li> <li>3. Вивести дані по книжці, вказавши її інвентарний номер.</li> <li>4. Вивести дані про читача по прізвищу.</li> <li>5. Вивести інформацію по читачу, що взяв вказану книжку.</li> <li>6. Вивести список читачів, що записались до бібліотеки у липні минулого року.</li> <li>7. Вивести список усіх книжок, що знаходяться «на руках».</li> <li>8. Дати інформацію, які книжки видавав вказаний бібліотекар.</li> <li>9. Вивести список книжок, назви яких починаються з вказаної літери.</li> <li>10. Вивести список літератури по вказаному відділу.</li> </ol>	
17	<p>Склад бази даних «Автоматизована система торгівлі»: таблиця «Товари», таблиця «Постачальники», таблиця «Клієнти», таблиця «Замовлення», таблиця «Замовлено товару», таблиця «Типи доставки», таблиця «Типи оплати».</p> <p>Структура таблиці «Товари»: код, найменування, кількість на складі, ціна за одиницю товару, опис, код постачальника.</p> <p>Структура таблиці «Постачальники»: код, ПШ, контактний телефон, адреса, факс, e-mail.</p> <p>Структура таблиці «Клієнти»: код, ПБ, адреса, контактний телефон, e-mail.</p> <p>Структура таблиці «Типи оплати»: код, назва типу, опис.</p> <p>Структура таблиці «Типи доставки»: код, тип доставки, опис.</p> <p>Структура таблиці «Замовлення»: код замовлення, замовлено товару на (грн.), вартість доставки, коментар, тип доставки, тип оплати.</p> <p>Структура таблиці «Замовлено товару»: код, код замовлення, код товару, куплено за (ціна), кількість.</p> <p><b>Використовуючи інформаційно-логічну модель:</b></p> <ol style="list-style-type: none"> <li>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</li> <li>б) Визначити ключові реквізити та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</li> <li>в) побудувати форми для перегляду та заповнення таблиць даними.</li> </ol> <p><b>Використовуючи зазначену базу даних та засоби мови SQL:</b></p> <ol style="list-style-type: none"> <li>1. Вивести інформацію по замовленню кожного клієнта.</li> <li>2. Вивести повну інформацію по клієнту.</li> <li>3. Вивести інформацію по вказаному товару.</li> <li>4. Вивести інформацію лише по типу доставки.</li> <li>5. Вивести динаміку по замовленнях.</li> <li>6. Вивести інформацію про те, який клієнт яким способом оплачує замовлення та суму, на яку було здійснене замовлення.</li> <li>7. Вивести усіх клієнтів, що зробили замовлення вище вказаної суми.</li> <li>8. За вказаним найменуванням товару вивести інформацію про його постачальника. Вивести інформацію по усіх постачальниках та товарах, що вони постачають.</li> <li>9. Дізнатись кількість замовленого товару одного найменування.</li> </ol>	

18	<p>Склад бази даних «Туристична агенція»: таблиця «Замовлення», таблиця «Клієнти», таблиця «Путівки».</p> <p>Структура таблиці «Замовлення»: № замовлення, дата замовлення, дата відправлення, код клієнта, код путівки, кількість білетів.</p> <p>Структура таблиці «Клієнти»: код клієнта, ПІБ клієнта, № страховки, адреса, телефон, паспортні дані.</p> <p>Структура таблиці «Путівки»: код путівки, країна, вид відпочинку, проїзд, дата відправлення, термін, ціна.</p> <p><b>Використовуючи інформаційно-логічну модель:</b></p> <p>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</p> <p>б) Визначити ключові реквізити та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</p> <p>в) побудувати форми для перегляду та заповнення таблиць даними.</p> <p><b>Використовуючи зазначену базу даних та засоби мови SQL:</b></p> <ol style="list-style-type: none"> <li>1. Вивести інформацію по клієнтах.</li> <li>2. Вивести ціну по вказаній путівці.</li> <li>3. Вивести інформацію по вказаному клієнту.</li> <li>4. Вивести номери страховок всіх клієнтів.</li> <li>5. Вказати тривалість зазначеної путівки.</li> <li>6. Визначити найпопулярнішу країну відпочинку.</li> <li>7. Ввівши прізвище клієнта, отримати інформацію про його путівку.</li> <li>8. Показати усі замовлення за вказаний місяць.</li> <li>9. Визначити, путівка до якої країни є найтривалішою.</li> <li>10. Вивести усіх клієнтів та дані по ним у алфавітному порядку.</li> </ol>	
19	<p>Склад бази даних «Факультет»: таблиця «Список студентів», таблиця «Оцінки», таблиця «Іспити», таблиця «Викладачі».</p> <p>Структура таблиці «Список студентів»: прізвище, рік народження, адреса, державна основа, контрактна основа, спеціальність.</p> <p>Структура таблиці «Оцінки»: прізвище, № з/к, група, код предмету, оцінка.</p> <p>Структура таблиці «Іспити»: код іспиту, назва предмету, викладач, табельний №.</p> <p>Структура таблиці «Викладачі»: табельний №, ПІБ, дата народження, адреса, телефон, вчений ступінь.</p> <p><b>Використовуючи інформаційно-логічну модель:</b></p> <p>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</p> <p>б) Визначити ключові реквізити та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</p> <p>в) побудувати форми для перегляду та заповнення таблиць даними.</p> <p><b>Використовуючи зазначену базу даних та засоби мови SQL:</b></p> <ol style="list-style-type: none"> <li>1. По табельному номеру вивести дані по викладачу.</li> </ol>	

	<ol style="list-style-type: none"> <li>2. Вивести список усіх студентів.</li> <li>3. Вказати студентів, що навчаються на державній і контрактній основі окремо.</li> <li>4. При введенні групи, вивести список студентів цієї групи.</li> <li>5. Дізнатись який викладач читає який предмет.</li> <li>6. Вивести інформацію про успіхи студентів вказаної групи з кожного предмету за даними іспитів.</li> <li>7. За прізвищем студента вивести його оцінки за іспит з кожного предмету.</li> <li>8. Отримати інформацію по кожному викладачу окремо.</li> <li>9. Ввівши прізвище викладача, отримати інформацію про предмет та групи, в яких цей предмет читався.</li> <li>10. Вказавши предмет і групу дізнатись за списком оцінки за іспит з цього предмету студента кожної групи.</li> </ol>	
20	<p>Склад бази даних «Аеропорт»: таблиця «Квитки», таблиця «Пасажири», таблиця «Замовлення квитка», таблиця «Розклад авіарейсів».</p> <p>Структура таблиці «Квитки»: код квитка, напрям, дата вильоту, тип літака, термінал, номер рейсу, код напрямку, вартість квитка.</p> <p>Структура таблиці «Пасажири»: код пасажира, ПІБ, клас, номер рейсу, код напрямку.</p> <p>Структура таблиці «Замовлення квитка»: номер замовлення, дата замовлення, код квитка, код пасажира, термінал.</p> <p>Структура таблиці «Розклад авіарейсів»: дні вильоту, код напрямку, час зльоту, напрям, час посадки, тип літака. Використовуючи інформаційно-логічну модель: а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі; б) Визначити ключові реквізити та побудувати схему даних, яка відповідає інформаційно-логічній моделі; в) побудувати форми для перегляду та заповнення таблиць даними.</p> <p><b>Використовуючи зазначену базу даних та засоби мови SQL:</b></p> <ol style="list-style-type: none"> <li>1. По коду пасажира вивести дані по замовленому квитку.</li> <li>2. Вивести розклад усіх авіарейсів.</li> <li>3. Визначити куди летить вказаний пасажир.</li> <li>4. Визначити дату вильоту та час рейсу.</li> <li>5. Вивести інформацію про усіх пасажирів.</li> <li>6. Вказати з якого терміналу і коли здійснюється виліт.</li> <li>7. Вказавши країну і час вильоту отримати список пасажирів.</li> <li>8. Дізнатись, які квитки були замовлені вказаного числа.</li> <li>9. Використовуючи дані по пасажирам отримати інформацію, хто летить першим класом.</li> <li>10. Вказавши клас польоту вивести список усіх пасажирів, що летять даним класом.</li> </ol>	10
21	Склад бази даних «Успішність у школі»: таблиця «Список учнів»,	

	<p>таблиця «Список вчителів», таблиця «Список кабінетів».</p> <p>Структура таблиці «Список учнів»: № п/п, ПІБ, дата народження, клас.</p> <p>Структура таблиці «Список вчителів»: табельний номер, ПІБ, дата народження, освіта.</p> <p>Структура таблиці «Список кабінетів»: № кабінету, найменування кабінету, телефон, відповідальний.</p> <p><b>Використовуючи інформаційно-логічну модель:</b></p> <p>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</p> <p>б) Визначити ключові реквізити та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</p> <p>в) побудувати форми для перегляду та заповнення таблиць даними.</p> <p><b>Використовуючи зазначену базу даних та засоби мови SQL:</b></p> <ol style="list-style-type: none"> <li>1. Вказати який викладач відповідає за який кабінет.</li> <li>2. Вивести дані про класне керівництво.</li> <li>3. По вказано прізвищу вивести дані про учня.</li> <li>4. Вивести дані по усіх викладачах.</li> <li>5. При введенні класу вивести список учнів цього класу.</li> <li>6. Вивести дані по усіх учнях школи.</li> <li>7. По табельному номеру отримати всю можливу інформацію по викладачу.</li> <li>8. Вказавши клас отримати список учнів, та прізвище класного керівника.</li> <li>9. Отримати вписок учнів, що народились вказаного року.</li> <li>10. Підрахувати кількість учнів у школі.</li> </ol>	
22	<p>Склад бази даних «Магазин»; таблиця «Товари», таблиця «Постачальники», таблиця «Клієнти», таблиця «Склад».</p> <p>Структура таблиці «Товари»: код товару, ціна, код постачальника, найменування.</p> <p>Структура таблиці «Постачальники»: код постачальника, місто, телефон, ПІБ.</p> <p>Структура таблиці «Клієнти»: код клієнта, місто, ПІБ, телефон.</p> <p>Структура таблиці «Склад»: код товару, код клієнту, наявність на складі. <b>Використовуючи інформаційно-логічну модель:</b></p> <p>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</p> <p>б) Визначити ключові реквізити та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</p> <p>в) побудувати форми для перегляду та заповнення таблиць даними.</p> <p><b>Використовуючи зазначену базу даних та засоби мови SQL:</b></p> <ol style="list-style-type: none"> <li>1. Вивести на екран замовлення вказаного клієнту.</li> <li>2. Визначити наявність та кількість вказаного товару на складі.</li> <li>3. По номеру замовлення вивести дані по цьому замовленню.</li> <li>4. Вивести дані про клієнтів.</li> </ol>	10

	<ol style="list-style-type: none"> <li>5. Вивести ціну на вказаний товар.</li> <li>6. По коду постачальника визначити який товар він постачає.</li> <li>7. Визначити найдорожчий та найдешевший товар у магазині.</li> <li>8. Вивести список постачальників з вказаного міста.</li> <li>9. Вивести список клієнтів з вказаного міста.</li> <li>10. Вивести дані по усіх постачальниках.</li> </ol>	
23	<p>Склад бази даних «Читальний зал»: таблиця «Бібліотекар», таблиця «Каталог», таблиця «Читач».</p> <p>Структура таблиці «Бібліотекар»: табельний №, ПІБ, дата народження, освіта, адреса, телефон, паспортні дані, ким і коли виданий.</p> <p>Структура таблиці «Каталог»: код напрямку, назва напрямку, інвентарний номер, автор, назва.</p> <p>Структура таблиці «Читач»: № формуляру, ПІБ, дата народження, освіта, професія, місце роботи, учбовий заклад, адреса, телефон, паспортні дані, ким і коли виданий.</p> <p><b>Використовуючи інформаційно-логічну модель:</b></p> <ol style="list-style-type: none"> <li>а) Побудувати та заповнити даними таблиці бази даних, що відповідають об'єктам моделі;</li> <li>б) Визначити ключові реквізити та побудувати схему даних, яка відповідає інформаційно-логічній моделі;</li> <li>в) побудувати форми для перегляду та заповнення таблиць даними.</li> </ol> <p><b>Використовуючи зазначену базу даних та засоби мови SQL:</b></p> <ol style="list-style-type: none"> <li>1. Вивести інформацію по усіх читачах.</li> <li>2. Вивести інформацію по вказаному бібліотекарю.</li> <li>3. По номеру формуляра отримати дані по читачу.</li> <li>4. Отримати список читачів, що працюють і де.</li> <li>5. Отримати список читачів, що навчаються і де.</li> <li>6. Отримати список літератури за вказаним напрямком.</li> <li>7. Отримати дані про книжки, написані вказаним автором.</li> <li>8. Вивести особистий номер вказаного бібліотекаря.</li> <li>9. Вивести список читачів у алфавітному порядку.</li> <li>10. Отримати список бібліотекарів, що не мають вищої освіти.</li> </ol>	10

## Рекомендовані джерела інформації

1. Бази даних: навч. посіб. / Босько В.В., Константинова Л.В., Поліщук Л.І., Коноплицька-Слободенюк О.К.; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. - Кропивницький : ЦНТУ, 2024. – 226 с.
2. Гайна Г.А. Основи проектування баз даних: Навчальний посібник. – К.: КНУБА, 2005. – 204 с.
3. Сидоренко В.В., Константинова Л.В., Смірнов С.А. Організація баз даних Навчальний посібник. - Кропивницький: ЦНТУ, 2018. – 274 с. [Електронний ресурс] - Режим доступу: <http://dspace.kntu.kr.ua/jspui/bitstream/123456789/10527/1/NavPosOBD.pdf> (дата звернення: 7.11.2023).
4. Пасічник В.В., Резніченко В.А. Організація баз даних та знань.-К: Видавнича група ВНУ, 2006.-384с.:іл.
5. Моделювання даних та маніпулювання даними /Навчальні ресурси СумДУ University online learning ecosystem. СумГУ, 2015 [Електронний ресурс] - Режим доступу: [https://elearning.sumdu.edu.ua/free\\_content/lectured:a8104441b8e00905159c1ff04257b014dd456247/20151109195846/162252/index.html](https://elearning.sumdu.edu.ua/free_content/lectured:a8104441b8e00905159c1ff04257b014dd456247/20151109195846/162252/index.html) (дата звернення: 20.03.2024).
6. Петух А.М., Романюк О.В., Романюк О.Н. Бази даних. Мови запитів, управління транзакціями, розподілена обробка даних, ВНТУ, 2016 [Електронний ресурс] - Режим доступу: [https://web.posibnyku.vntu.edu.ua/fitki/11petuh\\_bazdanyh\\_movy\\_zalitiv/index.htm](https://web.posibnyku.vntu.edu.ua/fitki/11petuh_bazdanyh_movy_zalitiv/index.htm) (дата звернення: 12.03.2024).
7. Клієнт-серверна архітектура. QATestLab. 28.05.2020. [Електронний ресурс] - Режим доступу: <https://training.qatestlab.com/blog/technical-articles/client-server-architecture/> (дата звернення: 20.03.2024).
8. Database Replication 101: Everything You Need To Know. January 24th, 2024. [Електронний ресурс] - Режим доступу: <https://www.astera.com/type/blog/database-replication-101/> (дата звернення: 20.03.2024).

9. ISO/IEC 9075-1:2023(en). Information technology - Database languages SQL [Електронний ресурс] - Режим доступу: <https://www.iso.org/obp/ui/en/#iso:std:76583:en> (дата звернення: 25.03.2024).
10. Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни «Бази даних» (1 частина) для студентів денної та заочної форми навчання за спеціальностями F3 «Комп'ютерні науки», F7 «Комп'ютерна інженерія», F5 «Кібербезпека та захист інформації», G5 «Електроніка, електронні комунікації, приладобудування та радіотехніка» / уклад. В.В. Босько, Л.В. Константинова — Кропивницький: ЦНТУ, 2025. — 85 с.
11. Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни «Бази даних» (2 частина) для студентів денної та заочної форми навчання за спеціальностями F3 «Комп'ютерні науки», F7 «Комп'ютерна інженерія», F5 «Кібербезпека та захист інформації», G5 «Електроніка, електронні комунікації, приладобудування та радіотехніка» / уклад. В.В. Босько, Л.В. Константинова, Л.І. Поліщук — Кропивницький: ЦНТУ, 2025. — 61 с.
12. Абрахам Зильбершцац, С. Сударшан Database System Concepts, Seventh Edition, McGraw-Hill, 2019.
13. John J. Patrick. SQL Fundamentals. [Електронний ресурс]/ - Режим доступу: [https://api.pageplace.de/preview/DT0400.9780137156139\\_A23587394/preview-9780137156139\\_A23587394.pdf](https://api.pageplace.de/preview/DT0400.9780137156139_A23587394/preview-9780137156139_A23587394.pdf) (дата звернення: 20.03.2025).
14. Elmasri, Navathe Fundamentals of Database Systems. [Електронний ресурс]/ - Режим доступу: <https://people.inf.elte.hu/kiss/DB/fundamentals-of-database-systems.pdf> (дата звернення: 20.03.2025).
15. Alan Beaulieu. Learning SQL [Електронний ресурс]/ - Режим доступу: [http://www.r-5.org/files/books/computers/languages/sql/mysql/Alan\\_Beaulieu-Learning\\_SQL-EN.pdf](http://www.r-5.org/files/books/computers/languages/sql/mysql/Alan_Beaulieu-Learning_SQL-EN.pdf) (дата звернення: 20.03.2025).
16. Features of SQL databases / LinkedIn. 27.05.2023. [Електронний ресурс]/ - Режим доступу: <https://www.linkedin.com/pulse/features-sql-databases-database-designer-sql-mysql> (дата звернення: 24.03.2024).

17. Stephane Faroult with Peter Robson «The Art of SQL». Sebastopol, Calif.: O'Reilly Media Inc. (2006).
18. 10 SQL Skills for Programmers and Developers / Indeed. 11.03.2023 [Електронний ресурс] - Режим доступу: <https://www.indeed.com/career-advice/resumes-cover-letters/sql-skills> (дата звернення: 25.03.2024).
19. Paul DuBois, MySQL, 5th Edition, Mar 29, 2013 by Addison-Wesley Professional.
20. SQL Підручник [Електронний ресурс] – Режим доступу: <https://w3schoolsua.github.io/sql/index.html#gsc.tab=0> (дата звернення: 7.04.2024).
21. Методичні вказівки до лабораторних робіт з дисципліни «Бази даних і знань» для студентів напряму підготовки «Управління інформаційною безпекою» / [уклад. : Ю. Є. Яремчук, Д. П. Присяжний, І. О. Дьогтева, О. В. Салієва] ; ВНТУ [Електронний ресурс] - Режим доступу: [https://web.posibnyku.vntu.edu.ua/fmib/37yaremchuk\\_metodvkaз\\_labrob\\_bazi\\_dani\\_h\\_znan\\_upravlinnya\\_informacijnoyu\\_bezpekoju/07.html](https://web.posibnyku.vntu.edu.ua/fmib/37yaremchuk_metodvkaз_labrob_bazi_dani_h_znan_upravlinnya_informacijnoyu_bezpekoju/07.html) (дата звернення: 7.04.2024).
22. Documentation. MySQL, 2024. [Електронний ресурс] - Режим доступу: <https://dev.mysql.com/doc/> (дата звернення: 21.03.2024).
23. Бази даних: метод. вказівки до виконання комп'ютерного практикуму для студентів спеціальності "Електронні комунікації та радіотехніка " / Уклад.: Суліма С.В., Глоба Л.С., Скулиш М.А.. – К.: КПІ ім. Ігоря Сікорського, 2023. – 54 с.
24. Efficient MySQL Performance: Best Practices and Techniques. 1st Ed. Daniel Nicher. O'Reilly, 2021. 276p.
25. Learning MySQL: Get a Handle on Your Data. 2nd Ed. Vinicius M. Grippa, Sergey Kuzmichev. O'Reilly, 2021. 550 p.
26. 97 Things Every Data Engineer Should Know: Collective Wisdom from the Experts. Tobias Macey. O'Reilly, 2021. 264 p.
27. Beginning Spring Data: Data Access and Persistence for Spring Framework 6 and Boot 3 1st ed. Edition. Andres Sacco. Apress, 2022. 439 p.
28. Learning PHP, MySQL & JavaScript. A Step-by-Step Guide to Creating

Dynamic Websites. 6th Ed. Robin Nixon. O'Reilly, 2021. 826 p.

29. PHP & MySQL: Novice to Ninja 7th Edition. Tom Butler. SitePoint, 2022. 686 p.

30. PostgreSQL Query Optimization: The Ultimate Guide to Building Efficient Queries. Anna Bailliekova, Henrietta Dombrovskaya, Boris Novikov. Apress, 2021. 344 p.

31. Суліма С. В., Єрмолаєв О. Д. Метод оптимізації SQL запитів системи управління базами даних. Київ : КПІ ім. І. Сікорського. Системи управління навігації та зв'язку Збірник наукових праць – 2023. – Вип. 2 (72). – С. 151-157. – doi:10.26906/SUNZ.2023.2.151.

32. Праворська Н.І., Яшина О.М., Нетребя І.В., Доміна А.Р., Кириченко О. М. Метод конструювання програмного забезпечення згідно аналізу помилок SQL-запитів. Вісник ХНУ: Технічні науки. – 2023. Вип. 3, 2023 (321). – С. 302-307. – URL: <http://journals.khnu.km.ua/vestnik/wp-content/uploads/2023/06/vknu-ts-2023-n3321-302-307.pdf>.

33. Смірнов О., Константинова Л., Коноплицька-Слободенюк, О., Козірова, Н., Якименко, Н., Доренський, О., & Буравченко, К. (2025). Дослідження інструментів штучного інтелекту для роботи з базами даних та аналізу даних. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 3(27), 429–448. [Електронний ресурс] - Режим доступу: <https://doi.org/10.28925/2663-4023.2025.27.763>