

Міністерство освіти і науки України

КИРОВОГРАДСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Кафедра програмування та захисту інформації

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт

з дисципліни «Управління базами даних»

для студентів спеціальності 6.020105 – «Документознавство
та інформаційна діяльність»

Методичні вказівки до лабораторних робіт з дисципліни “Управління базами даних”/ К.М. Марченко – Кропивницький: КНТУ, 2016. – 43 с.

Для студентів напряму підготовки 6.020105 "Документознавство та інформаційна діяльність" при вивченні навчальної дисципліни “Управління базами даних”. Визначено тематику лабораторних робіт, подані приклади створення та обробки баз даних, приведені варіанти індивідуальних завдань.

Автор-укладач:

Марченко Костянтин Миколайович - канд. техн. наук, доцент кафедри програмування та захисту інформації

Схвалено на методичному
семінарі кафедри ПЗІ
29.08.2016 р.

Затверджено на засіданні
кафедри ПЗІ, протокол № 2 від
31.08.2016 р.

Зміст

Вступ	3
1. Створення бази даних та таблиць у СУБД Microsoft Access.....	4
2. Обробка записів та полів таблиці у СУБД Microsoft Access.	10
3. Робота з об'єднаними таблицями. Фільтрація даних.....	12
4. Створення та використання запитів у СУБД Microsoft Access	14
5. Створення звітів за базою даних	17
6. Створення форм для обробки даних у Microsoft Access	19
7. Вивчення інтерфейсу СУБД Microsoft Visual FoxPro. Створення бази даних та таблиць даних.	25
8. Visual FoxPro. Режими перегляду та редагування записів даних. Робота з полями таблиці. Використання індексних файлів.....	27
9. Visual FoxPro. Робота з декількома таблицями одночасно. Розрахунки у таблицях. Використання математичних функцій.....	29
10. Обробка даних символьного типу та типу дати.....	31
11. Створення запитів за базою даних.	34
12. Створення командних файлів	36
13. Створення меню команд для обробки бази даних.....	36
14. Програмування з використанням підпрогам	38
Рекомендована література.....	43

Вступ

У діловій сфері часто приходиться працювати з даними з різних джерел, кожне з яких пов'язане з певним видом діяльності. Для координації всіх цих даних необхідні певні знання й організаційні навички.

Електронною базою даних (БД) називається послідовність даних заданої структури, записана на магнітний диск комп'ютера.

Система управління базами даних (СУБД) поєднує відомості з різних джерел в одній реляційній базі даних. Створювані форми, запити і звіти дозволяють швидко й ефективно обновляти дані, отримувати відповіді на питання, здійснювати пошук потрібних даних, аналізувати дані, друкувати звіти, діаграми і поштові наклейки.

У світі існує безліч СУБД. Незважаючи на те, що вони можуть по-різному працювати з різними об'єктами і надають користувачу різні функції й засоби, більшість СУБД спираються на єдиний устояний комплекс основних понять. Це дає нам можливість розглянути одну систему й узагальнити її поняття, прийоми й методи на весь клас СУБД. В якості такого навчального об'єкта розглянемо СУБД Microsoft Access, що входить до пакета Microsoft Office.

Access 1.0, випущений у 1992 р., був одним із самих чудових продуктів, будь-коли розроблених для персональних комп'ютерів. Цей пакет установив нові стандарти для інтерфейсу, поліпшив систему звітів і збільшив швидкість керування даними. Завдяки цьому він став найбільш популярним пакетом СУБД для Windows і єдиним універсальним програмним продуктом, що задовольняє усіх - і кінцевих користувачів, і розробників повномасштабних додатків. В даний час широко використовуються версії Access 97 і Access 2000.

При розробці баз даних одночасно приходиться працювати з декількома різними структурними об'єктами. У Access реалізований зовсім новий формат збереження даних. Єдина уніфікована структура, що називається контейнер, містить у собі всі структурні елементи - таблиці, запити, програмні модулі на Access Basic і т.д. Стандартне розширення цих файлів .MDB (Microsoft Data Base). При відкритті файлу .MDB всі об'єкти бази даних виводяться у виді списку у вікні бази даних. Для вибору одного із зазначених списків використовуються вкладки.

Розглянемо призначення об'єктів Access.

Таблиця призначена для збереження даних у вигляді записів (рядків) і полів (стовпців). Таблиці складають основу бази даних. Саме в них зберігається інформація, організована певним чином.

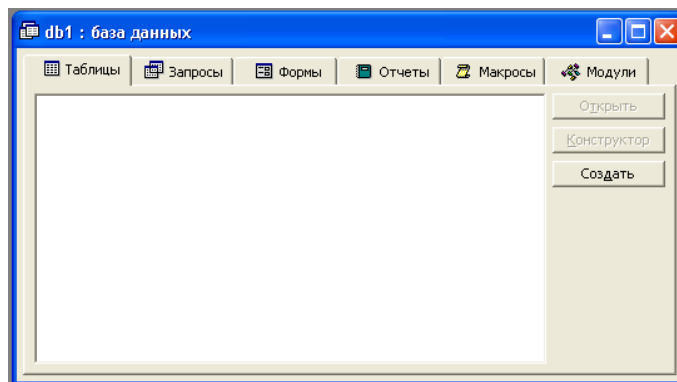


Рис. 1. Вікно бази даних у Access 97

Запит - дозволяє отримати потрібні дані з однієї чи декількох таблиць, розрахувати значення деяких даних за формулами.

Форма - об'єкт, призначений, як для введення, так і для ви-ведення даних у зручному для користувача вигляді. У формі можна розмістити елементи керування, які застосовуються для введення, зображення і зміни даних у полях таблиць.

Звіт - об'єкт, призначений для друку даних.

Макроси - засоби для автоматизації роботи з формами, звітами та ін.

Модулі - програмні модулі мовою Visual Basic.

Лабораторна робота № 1

Створення бази даних та таблиць у СУБД Microsoft Access

Основний структурний елемент Access - таблиця, в якій зберігається інформація. Усі інші об'єкти засновані на даних таблиць. По-дідно іншим базам даних Access використовує терміни *поле* (стовпець таблиці) і *запис* (рядок таблиці). Кожне поле має свій тип даних та властивості. Від властивостей полів залежить, які типи даних можна вносити у поле, а які ні, а також те, що можна робити з даними, що містяться у полі.

Типи полів:

1. Текстовий - використовується для збереження текстової інформації довжиною до 255 символів.

2. Мемо - використовується для збереження тексту великого обсягу до 65 535 символів. Особливість поля МЕМО полягає в тому, що реально ці дані зберігаються не в полі, а в іншому місці, а в полі зберігається тільки покажчик на те, де розташований текст.

3. Числовий - використовується для збереження числових значень, що можуть використовуватися в математичних розрахунках. Воно теж має розмір, але числові поля бувають різними, наприклад, для введення цілих чисел і для введення дійсних чисел. В останньому випадку крім розміру поля задається також розмір десяткової частини числа.

4. Дата/Час - значення вводяться до поля такого типу в стандартному форматі, наприклад, 1/1/99, 12:53:00 PM.

5. Грошовий - призначений для збереження точного значення дуже великих чисел і дозволяє проводити точні обчислення над ними. Грошові суми можна зберігати й у числовому полі, але в грошовому форматі з ними зручніше працювати. У цьому випадку комп'ютер зображує числа разом із грошовими одиницями, розрізняє карбованці і копійки, фунти і пенси, долари й центи.

6. Лічильник (Counter) - використовується для автоматичної нумерації запису, що додається. На перший погляд це звичайне числове поле, але воно має властивість автоматичного нарощування. Якщо в базі є таке поле, то при введенні нового запису в нього автоматично вводиться число, на одиницю більше, ніж значення того ж поля в попередньому записі.

7. Логічний - використовується для збереження логічних величин, що мають тільки два значення (Так чи Ні; 0 чи 1; Істина чи Неправда і т.п.). Довжина такого поля завжди дорівнює 1 біту.

8. Об'єкт OLE - дозволяє зберігати документи, таблиці, графіки, малюнки, звукові сигнали, відео- та іншу інформацію, створену додатками Windows, що підтримують технологію OLE.

9. Майстер підстановок - таке поле служить для створення зв'язків між таблицями СУБД Access 97.

Поля унікальні і ключові

Створення бази даних завжди починається з розробки структури її таблиць. Структура повинна бути такою, щоб при роботі з базою в неї було потрібно вводити якнайменше даних. Якщо введення якихось даних доводиться повторювати неодноразово, базу роблять з декількох зв'язаних таблиць. Структуру кожної таблиці розробляють окремо. Для того щоб зв'язки між таблицями працювали надійно, і за записом з однієї таблиці можна було однозначно знайти запис в іншій таблиці, треба передбачити в таблиці унікальні поля. **Унікальне поле** - це поле, значення в якому не можуть повторюватися. Якщо дані в полі повторюються і необхідно видавати повідомлення про це, то для таких даних використовують **ключове поле**.

Властивості полів

Крім основних елементів (Ім'я поля, Тип даних, Опис) поле може містити також набір властивостей. Кожен об'єкт у Access має власний набір властивостей. Властивості поля залежать від його типу.

Властивість поля	Призначення
Розмір поля (Field Size)	Визначає максимальну довжину текстового поля або обмежує

	припустимі значення для числових полів. За замовчуванням дорівнює 50 символів.
Формат (Format)	Визначає формат відображення даних. Його можна вибрати з уже створеного списку форматів або створити заново. У процесі вибору чи створення маски введення можна використовувати Access Mask Builder (Побудувач маски). Якщо як маску введення ввести Password (пароль), то в області поля для кожного символу буде виводитися зірочка.
Дробова частина (Decimal Places)	Визначає кількість цифр після десяткової коми. Число виведених десяткових знаків керує формою виведення чисел, але не обмежує їхнє введення, тобто якщо число десяткових знаків дорівнює 0, то в осередках будуть зберігатися все рівно повні числа.
Підпис (Caption)	Визначає мітку, що буде з'являтися у формах і звітах.
Значення за замовчуванням (Default Value)	Визначає встановлене за замовчуванням значення, що автоматично з'являється в поле при додаванні нових записів. Звичайно, такими значеннями є літеро-рали, такі як 0, для поля <i>Кількість</i> .
Умова на значення (Validation Rule)	Визначає умову, якій повинні задовольняти дані, що вводяться в поле. Перевірочні правила є виразами, використовуваними Access для визначення того, буде елемент, що вводиться, збережений як уміст поля.
Повідомлення про помилку (Validation Text)	Визначає повідомлення, що з'явиться в діалоговому вікні, якщо дані, що вводяться, не задовольняють умові. Якщо не ввести перевірочний текст, то при введенні невірних даних Access введе в діалогово-му вікні наступне повідомлення: The value you entered is prohibited by the validation rule set this field.
Обов'язкове поле (Required)	Визначає, чи обов'язково вводити в поле дані. Access не збереже запис доти, поки буде нульове значення хоча б одного поля, для якого встановлене значення Yes цієї властивості. Нульове значення - незаповнене поле (крім числових полів, для яких 0 - припустиме введення). У текстовому полі прогалина розглядається як нульове значення.
Допускається нульова довжина (Allow Zero Length)	Визначає, чи допускаються рядки нульової довжини (тобто порожні рядки). У деяких випадках розуміти різницю між полями типу <i>порожньо</i> (інформація відсутня) і <i>невідомо</i> (дані можуть бути введені пізніше й у полі вводиться слово Unknow чи None). Звичайно, якщо властивість має значення No, будь-яким незаповненими записам, що залишилися, авто-матично привласнюється нульове значення.
Індексоване поле (Indexed)	Визначає індекс для поля. Індокси використовують-ся для прискорення пошуку. Access підтримує два типові індокси: Індекс-первинний ключ (Primary key) (збіги не допускаються) Індекс-поле (збіги допускаються)

Створення і реорганізація таблиць

Створення таблиці складається в завданні її полів і призначенні їхніх властивостей. Воно починається з Клацання на кнопці **Создать** у вікні База даних. Є кілька способів створення нової таблиці, що відрізняються рівнем автоматизації:

1. Режим таблиці;
2. Конструктор;
3. Майстер таблиць;
4. Імпорт таблиць
5. Зв'язок із таблицями.

Самий "автоматичний" спосіб складається в **імпорті таблиць** з іншої бази, навіть створеної в іншій системі. У залежності від обставин з імпортованої таблиці може надійти структура полів,

їхні назви й властивості, а також і вміст бази. Якщо щось імпортується не зовсім так, як треба, необхідні коригування (наприклад, у властивості полів) вносять вручну.

У тих випадках, коли мова йде про чужу таблицю, що знаходиться на вилученому сервері і яку не можна імпортувати цілком, користуються режимом **Зв'язок із таблицями**. Це нагадує підключення до таблиці для спільного використання її даних.

Досвідчені розробники користуються **Майстром таблиць**. Це програма, що прискорює створення структури таблиці. Майстер задає ряд питань і, керуючись отриманими відповідями, створює структуру таблиці автоматично. Незважаючи на те що цей режим служить для спрощення роботи, початківцем користатися їм не рекомендується, оскільки, не володіючи всією термінологією, легко заплутатися в питаннях і відповідях. Перші таблиці варто спробувати створити вручну.

Пункт **Режим таблиці** відкриває заготовку, у якій усі поля мають формальні імена: Поле1, Поле2... і т.д. і один стандартний текстовий тип. Таку таблицю можна відразу наповнити інформацією.

Найбільш універсальний ручний метод надає пункт **Конструк-тор**. У цьому режимі можна самостійно задати імена полів, вибрати їхній тип і настроїти властивості.

Створення таблиці за допомогою Конструктора

1. У вікні бази даних вибрати вкладинку **Таблиця**.
2. Клацнути на кнопці **Создать**. З'явиться діалогове вікно конструктора таблиць. Вибрати режим **Конструктора** (рис. 2).
3. У стовпці **Імя поля** вводиться унікальне ім'я, що привласнюється даному полю. Імена полів можуть мати довжину не більш 64 символів. Бажано застосовувати короткі імена.
4. У стовпці **Тип даних** необхідно вибрати тип даних, що будуть зберігатися в полі.

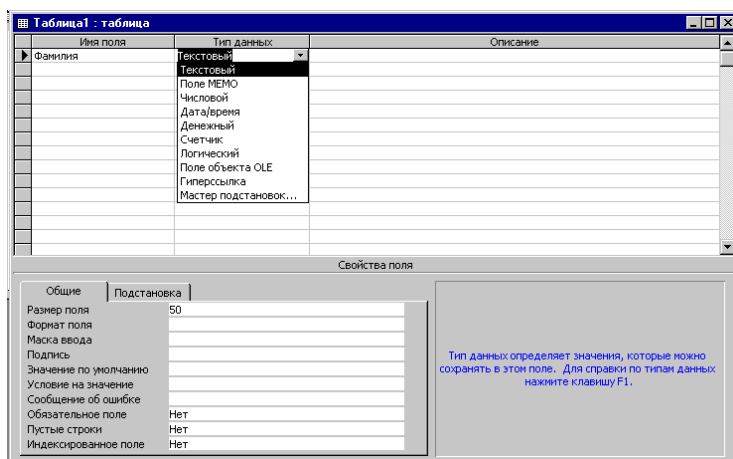


Рис. 2. Вікно Конструктора таблиць

5. У стовпці **Описание** вводиться необов'язковий опис поля, що забезпечує користувачів бази даних довідковою інформацією.

6. Ім'я поля, Тип даних і Опис вводиться для всіх полів.

7. Після цього виконується команда **Файл/Закреть**. У діалоговому вікні із запитом про ім'я таблиці ввести ім'я і клацнути на кнопці **ОК**.

Якщо в таблиці не визначений первинний ключ, то після введення імені таблиці відобразиться діалогове вікно, що рекомендує задати первинний ключ. Можна зберегти таблицю без первинного ключа (відповісти "ні") або використовувати як первинний ключ поле-лічильник (відповісти "так").

Перш ніж створювати нову таблицю, потрібно точно визначити, які поля вона буде містити. Необхідно визначити, скільки таблиць повинне бути створено, які поля (атрибути) будуть міститися в таблицях і яких типах даних будуть у цих полях. Крім того, необхідно виділити ключові поля, про роботу яких розповідається трохи пізніше.

Дайте відповіді на питання за темою «Таблиці»:

1. У яких об'єктах зберігаються дані бази?
2. Чим відрізняються поля й записи у таблиці?

3. Які типи полів таблиць ви знаєте?
4. Чи містить яку-небудь інформацію таблиця, у якій немає жодного запису?
5. Чи містить яку-небудь інформацію таблиця, у якій немає полів? Чи може існувати така таблиця?
6. У чому полягає особливість поля типу Лічильник?
7. У чому полягає особливість поля типу МЕМО?
8. Яке поле можна вважати унікальним?
9. Яким повинне бути поле первинного ключа?
10. У чому перевага поля, що є первинним ключем?
11. Яку базу даних називають реляційною?
12. У якому діалоговому вікні створюють зв'язки між полями таблиць бази даних?
13. Для чого призначений фільтр даних?
14. Чому при закритті таблиці програма Access 97 не пропонує виконати збереження внесених даних?

Завдання

1. Створити базу даних згідно варіантові.
2. Використовуючи Конструктор, створити першу таблицю бази даних. Задати ключове поле. Зберегти таблицю під указаним іменем. Внести до таблиці 15 записів даних.
3. Використовуючи Майстер таблиць, створити другу таблицю бази даних. Задати ключове поле. Зберегти таблицю під указаним іменем. Внести до таблиці 15 записів даних. Перенести співпадаючі дані з таблиці до таблиці шляхом копіювання.
4. Налагодити зв'язок між таблицями.

Варіанти індивідуальних завдань:

Варіант 1

Ім'я бази даних	Імена таблиць, полів таблиць	
	Кадри	Бухгалтерія
Співробітник	<ul style="list-style-type: none"> • Табельний номер • Прізвище, ім'я та по-батькові • Підрозділ • Посада • Рік народження • Стаж роботи • Домашня адреса • Фото 	<ul style="list-style-type: none"> • Табельний номер • Прізвище, ім'я та по-батькові • Підрозділ • Посада • Посадовий оклад • Ідентифікаційний код

Варіант 2

Ім'я бази даних	Імена таблиць, полів таблиць	
	Технічні дані	Бухгалтерія
Облік підйомно-транспортного обладнання	<ul style="list-style-type: none"> • Інвентарний номер • Найменування • Рік випуску • Вантажопідйомність • Маса • Потужність двигуна 	<ul style="list-style-type: none"> • Інвентарний номер • Найменування • Підрозділ, в якому використовується • Балансова вартість • Дата придбання • Кількість

Варіант 3

Ім'я бази даних	Імена таблиць, полів таблиць	
	Підприємство	Галузь

Облік електроенергії	<ul style="list-style-type: none"> • Код підприємства • Код галузі • Назва підприємства • Директор • Головний енергетик • Адреса • Обсяг споживання • Норма споживання 	<ul style="list-style-type: none"> • Код галузі • Назва галузі • Назва міністерства
----------------------	--	--

Варіант 4

Ім'я бази даних	Імена таблиць, полів таблиць	
	Поставки	Наявність
Облік запасних частин	<ul style="list-style-type: none"> • Код деталі • Найменування деталі • Номер партії • Обсяг постачання • Ціна за одиницю • Дата постачання • Постачальник • Обсяг попиту 	<ul style="list-style-type: none"> • Код деталі • Найменування деталі • Дата обліку • Обсяг постачання • Номер складу • Видано • Залишок

Варіант 5

Ім'я бази даних	Імена таблиць, полів таблиць	
	Поставки	Наявність
Облік промислових товарів	<ul style="list-style-type: none"> • Код товару • Найменування товару • Номер партії • Обсяг постачання • Ціна за одиницю • Дата постачання • Постачальник • Товарний знак 	<ul style="list-style-type: none"> • Код товару • Найменування товару • Дата обліку • Обсяг постачання • Номер складу • Видано • Залишок

Варіант 6

Ім'я бази даних	Імена таблиць, полів таблиць	
	Поставки	Наявність
Облік продовольчих товарів	<ul style="list-style-type: none"> • Код товару • Найменування товару • Номер партії • Обсяг постачання • Ціна за одиницю • Дата постачання • Постачальник • Товарний знак 	<ul style="list-style-type: none"> • Код товару • Найменування товару • Дата обліку • Обсяг постачання • Номер складу • Видано • Залишок

Варіант 7

Ім'я бази даних	Імена таблиць, полів таблиць	
	Особисті справи	Успішність
Студент	<ul style="list-style-type: none"> • Факультет • Спеціальність • Шифр групи • Рік вступу 	<ul style="list-style-type: none"> • Факультет • Шифр групи • Прізвище, ім'я та по-батькові • Предмет I

	<ul style="list-style-type: none"> • Прізвище, ім'я та по-батькові • Рік народження • Базова освіта • Домашня адреса • Фото 	<ul style="list-style-type: none"> • Предмет2 • Предмет3 • Предмет4
--	--	--

Варіант 8

Ім'я бази даних	Імена таблиць, полів таблиць	
	Абонемент	Відділ комплектації
Книга	<ul style="list-style-type: none"> • Шифр • Автор • Назва • Кількість за обліком • Кількість в наявності • Рівень попиту 	<ul style="list-style-type: none"> • Шифр • Автор • Назва • Рік видання • Кількість за обліком • Ціна • Обкладинка

Варіант 9

Ім'я бази даних	Імена таблиць, полів таблиць	
	Комерційні дані	Технічні дані
Автопродаж	<ul style="list-style-type: none"> • Марка автомобіля • Дата постачання • Номер партії • Кількість • Ціна закупки • Продажна ціна • Постачальник 	<ul style="list-style-type: none"> • Марка автомобіля • Призначення • Кількість місць • Потужність двигуна • Швидкість • Витрати палива • Тип палива • Країна-виробник • Фото

Варіант 10

Ім'я бази даних	Імена таблиць, полів таблиць	
	Автомобіль	Власник
Автореєстр	<ul style="list-style-type: none"> • Державний номер • Марка • Колір • Рік випуску • Дата останнього техогляду • Літраж двигуна • Пробіг 	<ul style="list-style-type: none"> • Державний номер • Марка • Прізвище та ініціали власника • Домашня адреса • Телефон

Варіант 11

Ім'я бази даних	Імена таблиць, полів таблиць	
	Кіоск	Продавець
Преса	<ul style="list-style-type: none"> • № кіоска • Назва • Адреса • Кількість найменувань видань, що реалізуються • Денна виручка • Час роботи 	<ul style="list-style-type: none"> • № кіоска • Назва • Прізвище, ім'я та по-батькові продавця • Стаж роботи

Варіант 12

Ім'я бази даних	Імена таблиць, полів таблиць	
	Загальні дані	Технічні дані
Кіровоградгаз	<ul style="list-style-type: none"> № рахунку Адреса Прізвище та ініціали власника Ціна 1 м³ газу Обсяг споживання Тариф при спожи-ванні у межах норми Наявність субсидії 	<ul style="list-style-type: none"> № рахунку Адреса Газова плита Колонка Котел Лічильник

Лабораторна робота № 2

Обробка записів та полів таблиці у СУБД Microsoft Access

Прийоми роботи з таблицями

Додавання, редагування й видалення даних у таблиці здійснюється в режимі таблиці. Необхідно двічі клацнути в потрібній таблиці у вікні бази чи даних виділити таблицю, а потім клацнути на кнопці **Открыть**.

Кожен запис має ліворуч кнопку (маркер запису). Клацання на цьому маркері виділяє весь запис і готує його до копіювання, переміщення, видалення. Клацання правою кнопкою на виділеному записі відкриває контекстне меню для операцій із записом.

Код п	Код от	Наимено	Директор	Энергетик	Адрес предприят	Норма расход
11	1	Тяжмаш	Иванов П.С.	Громов Л.А.	Днепропетровск, ул	25000
12	1	Станко	Ткачук С.А.	Свиридов Д.П.	Киев, пр. Победы,	18000
13	1	Инструме	Яновский М.В.	Кузнецов А.И.	Харьков, пр. Инжен	16000
21	2	Точприбо	Краснов Р.Б.	Искра П.В.	Львов, Ст. Бандери	13200
22	2	Электрон	Рабинович И.А.	Владимирцев	Винница, ул. Садо	15500
23	2	Электром	Трощенко В.Ю.	Белусов М.С.	Донецк, ул. Горняк	12400
24	2	Сегмент	Тимошенко С.Г.	Горбачев К.Д.	Кировоград, пер. Э	8400
31	3	Киянка	Сибирцева Л.А.	Степаненко В.Г.	Киев, пр. Науки, 87	12000
32	3	Синий ле	Петренко О.Д.	Корнев А.А.	Житомир, ул. К. Ме	9500
*						0

Рис. 3. Вікно роботи з даними в таблиці.

Маркер, що знаходиться в лівому верхньому куті таблиці, - це маркер таблиці. Клацання на ньому лівою клавішою миші виділяє всю таблицю, а клацання правою - відкриває контекстне меню для операцій з таблицею в цілому.

Якщо вміст поля не цілком уміщається в таблиці, стовпець мож-на розширити. При наведенні покажчика миші на границю між стовпцями покажчик змінює форму. Тепер границю можна переміщати методом перетаскування, а подвійне клацання, виконане у цей момент, автоматично встановлює ширину стовпця рівній довжині самого дов-гого значення в даному полі.

Контекстне меню запису дозволяє видаляти, копіювати і пере-міщувати запис і керувати висотою рядка.

Код п	Код от	Наимено	Директор	Энергетик	Адрес предприят	Норма расход
11	1	Тяжмаш	Иванов П.С.	Громов Л.А.	Днепропетровск, ул	25000
12	1	Станко	Ткачук С.А.	Свиридов Д.П.	Киев, пр. Победы,	18000
13	1	Инструме	Яновский М.В.	Кузнецов А.И.	Харьков, пр. Инжен	16000
21	2	Точприбо	Краснов Р.Б.	Искра П.В.	Львов, Ст. Бандери	13200
22	2	Электрон	Рабинович И.А.	Владимирцев	Винница, ул. Садо	15500
23	2	Электром	Трощенко В.Ю.	Белусов М.С.	Донецк, ул. Горняк	12400
24	2	Сегмент	Тимошенко С.Г.	Горбачев К.Д.	Кировоград, пер. Э	8400
31	3	Киянка	Сибирцева Л.А.	Степаненко В.Г.	Киев, пр. Науки, 87	12000
32	3	Синий ле	Петренко О.Д.	Корнев А.А.	Житомир, ул. К. Ме	9500
*						0

Рис. 4. Прийоми роботи із записами таблиці

Контекстне меню стовпця дозволяє сортувати записи, копію-вати, видаляти і переміщувати стовпці, керувати їх шириною й режимом відображення.

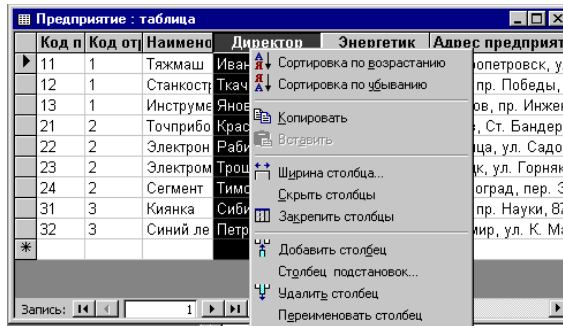


Рис. 5. Прийоми роботи з полями таблиці

Уведення даних до полів MEMO

У режимі таблиці можна вводити дані в поле MEMO, але побачити дані відразу, звичайно, неможливо. Натискання <Shift+F2> відкриває збільшене вікно введення, зручне для роботи з великими текстами.

Додавати дані у поля об'єкта OLE можна в такий спосіб:

1. У додатку Windows, що містить дані для вставки в поле, виділити потрібні дані.
2. Виконати команду **Правка/Копировать**.
3. Перейти в Access і помістити курсор у поле об'єкта OLE необхідного запису.
4. Виконати команду **Правка/Вставить**.

Ключі

Для реляційних СУБД, таких як Access, дані різних категорій зберігаються в різних таблицях. Це дозволяє виключити надмірність інформації. Однак між таблицями повинен бути встановлений зв'язок. Тільки тоді буде забезпечуватися правильна обробка даних. Вище вже говорилося про такі поняття, як унікальні поля і ключові поля. Для раціональної, правильної роботи з базою даних необхідно познайо-митися ще з поняттями первинного й вторинного ключа, а також індексованого поля.

Первинний ключ - один або кілька полів, за якими однозначно ідентифікуються записи в таблиці. Первинні ключі збільшують швидкодію й уможливають додавання зв'язків за замовчуванням. Для первинних ключів завжди необхідно використовувати дані, що ніколи не будуть повторюватися (номера покупців, рахунків, страховок, накладних). Іноді має сенс як первинний ключ використовувати комбі-націю полів. У цьому випадку говорять, що таблиця має подвійний (потрійний і т.д.) зчеплений ключ.

За замовчуванням первинні ключі в списку табличних даних відображаються першими.

У режимі конструктора таблиць первинний ключ можна задати в такий спосіб: вибрати одне чи кілька полів; клацнути на піктограмі "ключ" основної панелі інструментів Access. Ліворуч від імен полів з'явиться піктограма із зображенням ключа.

Індексовані поля

У Access підтримується два види індексів.

- Збіги не допускаються. Цей індекс працює подібно первин-ному ключу.
- Збіги допускаються. Можна створити індекс для полів із можливими повторюваними значеннями, включаючи і нульові.

Приклад виконання роботи

Відомі дані про співробітників:

табельний номер; прізвище, ім'я, по батькові; відділ; посада; дата народження; оклад; пільги по оподатковуванню; фото співробітника; додаткові відомості.

Спроекуємо таблиці бази даних, при цьому необхідно врахува-ти, що дані про посади і відділи зберігаються у відповідних довідни-ках: довіднику посад і довіднику відділів.

Створимо наступні таблиці:

1. **Довідник посад.** У вікні бази даних виберіть вкладинку **Таблицы**, натисніть кнопку **Создать** - виберіть режим **Конструктор/ ОК** і введіть структуру таблиці (рис. 6).

Виділіть поле Код_посади в області маркірування поля і натисніть кнопку "**Ключ**" на панелі інструментів.

Закрийте вікно конструктора таблиці і привласніть таблиці ім'я **Довідник посад**.

2. **Довідник відділів.** У вікні бази даних виберіть вкладинку **Таблицы**, натисніть кнопку

Создать - виберіть режим **Конструктор/ ОК** і введіть структуру таблиці.

Задайте ключ для поля Код_відділу. Закрийте таблицю і при-власніть їй ім'я Довідник відділів. Заповніть створені таблиці даними. Натисніть кнопку **Открыть** і уведіть вихідні дані.

3. Створіть структуру таблиці *Основні відомості*.

Примітка. Для полів Код_посади і Код_ відділу виберіть Тип даних.

Майстер підстановок

Майстер підстановок створює в таблиці стовпець підстановок, що відображає список значень, які може вибрати користувач. Процес підстановки відбувається в 5 кроків (перехід до наступного здійснює-ться натисканням кнопки **Дальше**):

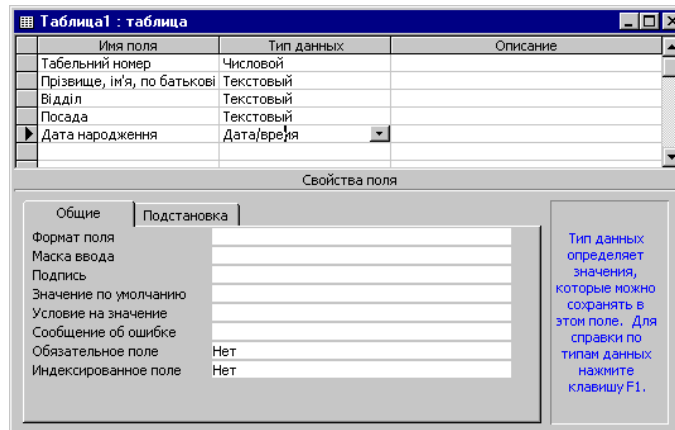


Рис. 6. Вікно Конструктора таблиць

Крок 1: Створюється стовпець підстановки. Стовпець підстановки використовує значення з таблиці або запиту.

Крок 2: Вибір таблиці або запиту. Вибираємо таблицю *Довідник відділів*.

Крок 3: Вибір полів. Вибираємо *Код_відділу* і *Найменування_відділу*.

Крок 4: Задайте ширину стовпців і установіть прапорець **“Спрятать ключевой столбец”**).

Крок 5: Задайте підпис поля - *Відділ*.

Перед створенням підстановки програма видає запит на збереження таблиці, тому заздалегідь установіть ключове поле у Вашій таблиці (якщо воно необхідно). Якщо ключове поле не задане, Access запропонує Вам створити таке поле, у випадку позитивної відповіді у Вашій таблиці з'явиться ключове поле Код, що має тип Лічильник.

Для поля *Код_посади* виберіть джерело *Довідник посад* і поля *Код_посади* і *Найменування_посади*.

Після введення всіх полів збережіть структуру таблиці. У даній таблиці використовуються підстановки для полів *код_посади* і *код_відділу* зі зв'язаних таблиць, у результаті підстановки, тип поля замінюється відповідним типом із зв'язаної таблиці.

Лабораторна робота №3

Робота з об'єднаними таблицями. Фільтрація даних

Установлення зв'язків

Основні СУБД реалізуються при роботі не з окремими таблицями, а з групами взаємозалежних таблиць. На практиці найбільш часто зустрічається зв'язок *один - до - багатьох*. При такому типі зв'язку кожному запису головної таблиці можуть бути поставлені у відповідність один або кілька записів так званої підлеглої таблиці.

Бази даних, що містять зв'язані таблиці, називають також реляційними базами даних. Для створення зв'язків між таблицями СУБД Access 97 має спеціальне діалогове вікно, що називається **Схема даних**.

1. Вікно **Схема даних** відкривають клацанням на однойменній кнопці панелі інструментів або командою **Сервис/Схема даних**.

2. Якщо раніше ніяких зв'язків між таблицями бази не було, то при відкритті вікна **Схема даних** одночасно відкривається вікно **Добавление таблицы**, у якому можна вибрати потрібні

таблиці для включення в структуру міжтабличних зв'язків.

3. Якщо зв'язки між таблицями вже були задані, то для введення в схему даних нової таблиці треба клацнути правою кнопкою миші на схемі даних і в контекстному меню вибрати пункт **Добавить таблицу**.

4. Увівши в схему даних усі таблиці, які треба зв'язати, можна приступати до створення зв'язків між полями таблиць.

5. Зв'язок між полями встановлюють шляхом перетаскування імені поля з однієї таблиці в іншу на відповідне йому зв'язане поле.

6. Після перетаскування відкривається діалогове вікно **Связи**, у якому можна задати властивості зв'язку, що утвориться (Рис. 7).

7. Включення прапорця **Обеспечение условия целостности данных** дозволяє захиститися від випадків видалення записів з однієї таблиці, при яких зв'язані з ними дані інших таблиць залишаться без зв'язку.

8. Прапорці **Каскадное обновление связанных полей** і **Кас-кадное удаление связанных записей** забезпечують одночасне від-новлення чи видалення даних у всіх підлеглих таблицях при їхній зміні в головній таблиці.

Щоб умова цілісності здійснювалася, поле основної таблиці повинне обов'язково бути ключовим, і обидва поля повинні мати однаковий тип.

Установлені зв'язки автоматично використовуються для зв'язування полів у створюваних запитах, формах і звітах. Цей засіб заощаджує час, гарантує отримання правильних результатів при розробці складних форм і звітів, підтримує цілісність бази даних. Дані автоматично захищаються від змін і видалень.

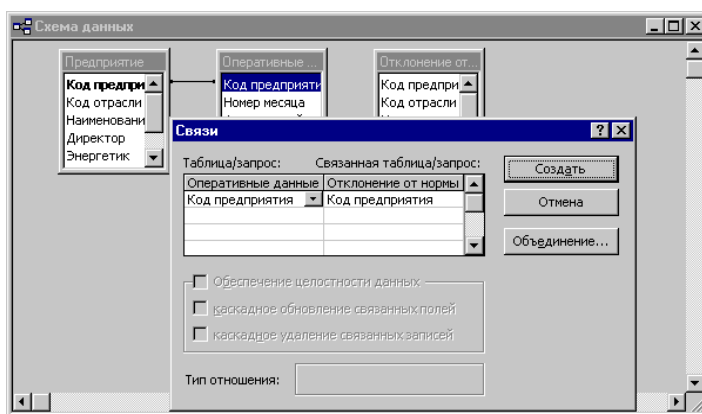


Рис. 7. Вікно **Схема даних**

Управління виведенням даних шляхом фільтрації

Для виведення на екран даних, які задовольняють певній умові необхідно подати команду **Записи/Фильтр/Расширенный фильтр** і ви-конати наступні дії (рис. 8):

1. Перетягнути ім'я поля або вибрати його у списку;
2. Задати при необхідності режим сортування при виведенні відфільтрованих даних
3. Ввести умову відбору даних вручну або викликати для її конструювання **Построитель выражений**.

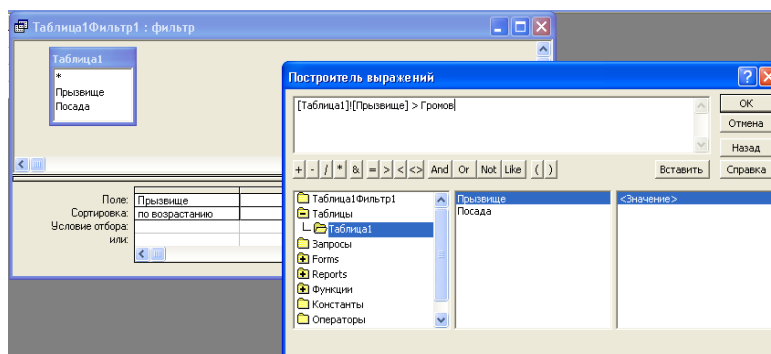


Рис. 8. Вікна фільтра даних і **Построителя выражений**

Лабораторна робота №4

Створення та використання запитів у СУБД Microsoft Access

Отримання відповідей на складні питання - одна з основних причин, за якою витрачаються кошти і час на створення та використання СУБД. Основний засіб для отримання цих відповідей - запити.

Запит (Query) - звертання за інформацією до бази даних. Запити в Access забезпечують швидкий і ефективний доступ до даних, що зберігаються в таблиці. Запити застосовуються для вибору стовпчиків і рядків, виведених на дисплей, для сортування рядків і для виконання звітів. За допомогою запитів можна також звести разом дані зі зв'язаних таблиць.

При роботі з таблицями виникають різні проблеми. У таблицях не дозволяється використовувати поля, що обчислюються. Крім того, Access майже завжди автоматично сортує таблиці відповідно до первинного ключа. Не існує також можливості вибрати інший критерій сортування, наприклад, по вторинному ключі. Усі ці проблеми легко вирішуються за допомогою запитів.

При виконанні запиту Access зчитує дані з таблиць і відображає результат виконання в режимі таблиці. При цьому варто враховувати, що результат виконання запиту не зберігається. Дані завжди зберігаються в таблицях. У запиті Access зберігає тільки інструкції про те, як повинні бути організовані дані в результаті виконання запиту. Однак варто пам'ятати про те, що при зміні даних у записах динамічного набору запиту змінюються і записи вихідних таблиць.

На основі запиту можна розробити форму або звіт (аналогічно створенню форми або звіту на основі таблиці). У Access використовуються різні види запитів: запити на вибірку, запити з параметрами, запити на зміну, запити з обчисленням, підсумкові запити і т.д., з деякими з яких ви познайомитеся нижче.

Запит-вибірка

1. У вікні БД виберіть вкладку "**Запрос**".
2. Натисніть кнопку **Создать**, вибрати **Конструктор**. Відкривається вікно "**Запрос на выборку**".
3. У вікні, що відкрилося, "**Добавление таблицы**" виберіть одну чи декілька потрібних для запиту таблиць для запиту і закрийте це вікно. На екрані залишається вікно конструктора запиту, що відображає списки полів обраних таблиць із зв'язками.
4. Відбуксируйте до бланка запиту необхідні поля даних.
5. Для кожного з полів, за якими повинне бути проведене сортування, виберіть спосіб сортування.

Якщо потрібна присутність усіх полів таблиці, то можна відбуксирувати в бланк запиту "зірочку" *.

Можна обрати "**Вывод на экран**" для поля сортування.

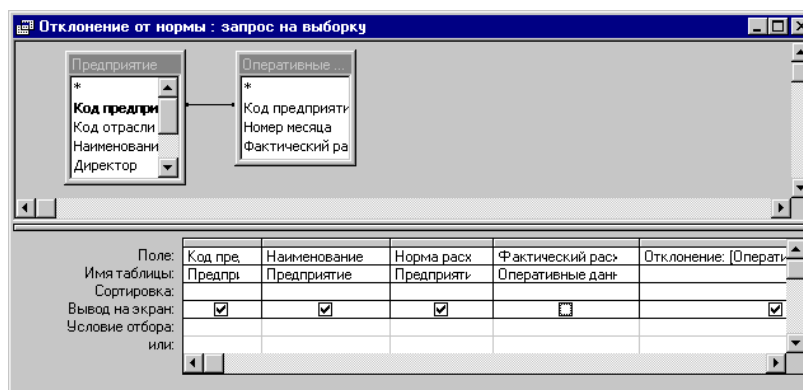


Рис. 9. Створення запиту в режимі Конструктора.

Створивши запит, можна:

- Відобразити на екрані результат запиту (піктограма "**Выполнить**");
- Зберегти для подальшого використання. Для цього при закритті запиту вводиться назва.

- Можна відмовитися від збереження внесених змін або від збереження запиту.

Запити - фільтри

При необхідності відобразити на екрані дані однієї або декількох таблиць відповідно до різних критеріїв використовуються запити з критеріями пошуку. Наприклад:

- відобразити на екрані відомості про співробітників, прізвище яких починається з букви «А»;
- відобразити на екрані відомості про співробітників, оклад яких менше або більше певної величини і т.д.

При виконанні запиту Access формує вікно для введення пара-метра, і результат виконання запиту з'являється на екрані тільки після введення значень всіх параметрів.

Запити до зв'язаних таблиць

У запиті повинна бути врахована інформація одночасно з декількох таблиць, причому автоматично враховуються зв'язки між таблицями. Наприклад, необхідно видати повну інформацію, що не містить кодів. Усі зв'язані дані беруться з різних таблиць.

Будемо вважати, що зв'язки між таблицями встановлені. У протилежному випадку їх треба установити за допомогою команди "Св'язи". Після того як зв'язки встановлені, можна формувати запит аналогічно вищевказаним запитам із фільтрами або з параметрами й ін., тобто відбуксирувати всі необхідні поля до бланка QBE і далі вказати умови вибору, сортування, обчислення і т.д. Уся необхідна інформація автоматично буде додана.

Що обчислюються в запитах

Access не дозволяє виконати обчислення над даними у числових полях таблиці. Для того щоб робити обчислення і відображати ці дані в табличному вигляді або у вигляді форми, використовуються запити з обчисленнями. В одному зі стовпців замість імені поля записують формулу. До формули входять взяті у квадратні дужки назви полів, що беруть участь у розрахунку, а також знаки математичних операцій.

Наприклад, для нарахування премії відхилення обсягу споживання електроенергії від норми потрібно:

- Активізувати вікно запитів. Створити запит, відбуксировавши поля *Назва підприємства*, *обсяг споживання* з таблиці «*Основні відомості*» до бланка запиту.
- У порожньому стовпці бланка запиту увести вираз для обчислення:

Рис. 10. Розрахункове поле у вікні Конструктора запитів

У квадратні дужки беруть тільки імена полів.

У вузький стовпець непросто записати довгу формулу, але якщо натиснути комбінацію клавіш **SHIFT+F2**, то відкривається допоміжне діалогове вікно, яке називається **Область вводу**. У ньому можна ввести яку завгодно довгу формулу, а потім клацанням на кнопці **ОК** перенести її до бланка запиту.

Сформувати поле, що обчислюється, можна також за допомогою **Побудователя**. Натисніть праву кнопку миші, виберіть у контекстному меню пункт **Побудувати**. У вікні, що відкрилося, введіть ім'я нового поля і сформулюйте арифметичне вираження, вибираючи поля запиту із середньої частини **Побудователя** кнопкою **Вставити** (якщо поля відсутні, збережіть запит, потім знову відкрийте його в режимі Конструктора і повторіть операцію).

Запити видалення

Видалення великої кількості записів вручну за допомогою команди **Правка/Удалить** займає дуже багато часу. У результаті деякі записи можуть виявитися вилученими по необережності. У цьому випадку користувач може скласти Запит видалення, що видалить усі записи за визначеним критерієм.

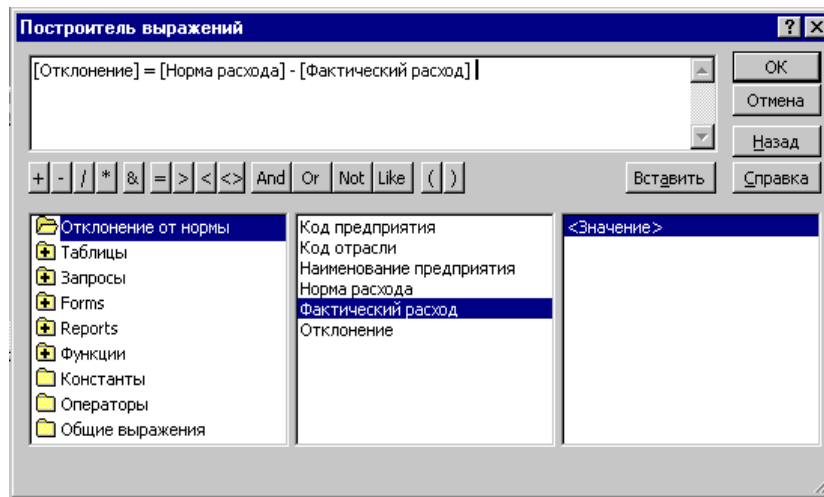


Рис. 11. Вікно Построителя выражений

- Виберіть таблицю, у якій пізніше буде проводитися видалення.

• Створіть запит вибору і перегляньте результат на екрані - чи всі записи обрані за визначеним Вами критерієм.

• Якщо в результаті запиту присутні тільки записи, що підлягають видаленню, варто перетворити запит вибору в запит видалення. Для цього в рядку меню виберіть команду **Запрос/Удаление**. Access залишає в бланку запит для всіх полів рядка.

• Виконайте запит. При цьому Access не відобразить на екрані результат виконання запиту. З'являється тільки повідомлення про кількість записів, що буде видалено у вихідній таблиці. Можна перервати процес видалення натисканням кнопки **"Отмена"**.

Запити заміни

Внесення однакових змін у велику кількість записів займає багато часу і може з'явитися причиною численних помилок. Запити заміни автоматично виконують заміну даних у ряді записів.

- Створіть новий запит-вибірку.
- Уведіть критерії для вибору змінюваних записів.
- Перегляньте результат на екрані - чи всі записи обрані правильно за критерієм.
- Перетворіть запит-вибірку в запит заміни. Для цього виберіть команду **Запрос/Обновление**

• У рядок **"Обновление"** уведіть нові вирази для заміни записів. Тип даних у виразах повинний збігатися з типом даних поля вихідної таблиці.

- Виконайте запит.

Підсумкові запити

Запити дозволяють не тільки відбирати потрібну інформацію з таблиць і обробляти її шляхом створення нових (що обчислюються) полів, але і робити так звані *підсумкові обчислення*.

Прикладом підсумкового обчислення може служити сума всіх значень у якійсь *групі* записів або їхнє середнє значення, хоча крім *суми* і *середнього значення* існують і інші підсумкові функції.

Оскільки підсумкові функції для одного запису не мають змісту й існують тільки для *групи записів*, то попередньо записи треба *згрупувати* за якою-небудь ознакою.

Припустимо, необхідно одержати відомості про максимальну, мінімальну, середню заробітну плату в цілому по відділах. Для цього записи потрібно згрупувати по відділах і для кожної з груп можна провести підсумкове обчислення по полю *оклад*.

Підсумкові запити створюють на основі відомого нам бланка запиту за зразком, тільки тепер у ньому з'являється додатковий рядок – **Групировка**.

- Для введення цього рядка в бланк треба клацнути на кнопці **Групповые операции** на панелі інструментів програми Access 97.

- У тих полях, по яких виробляється угруповання, треба установити (чи залишити) функцію **Группировка**.

- У тих полях, за якими варто провести підсумкове обчислення, треба в рядку **Группировка** розкрити список і вибрати одну з де-кількох підсумкових функцій.

- Клацання, на кнопці Вид запускає запит і видає результуючу таблицю з необхідними

підсумковими даними.

У рядку **Групування** можна вказати лише одну підсумкову функцію. А як бути, якщо треба знайти і суму, і середнє, і максимальне значення, і ще щось? Рішення просте: те саме поле можна включити в бланк запиту за зразком кілька разів, як показано на прикладі.

Дайте відповіді на питання за темою «Запити»

1. Для чого призначені запити?
2. Запити баз даних звернені до таблиць. Як називаються таблиці, на базі яких заснований запит?
3. У результаті роботи запиту утвориться таблиця. Як називається ця таблиця?
4. Як називається категорія запитів, призначена для вибору даних із таблиць бази?
5. Як називається бланк, заповненням якого створюється струк-тура запиту?
6. Як називається поле, в якому розміщуються результати роз-рахунку за участю даних, що містяться в інших полях?
7. Як називається категорія запитів, що виконують підсумкові обчислення?
8. Які підсумкові функції ви знаєте?
9. Для виконання підсумкових обчислень записи в таблицях повинні бути згруповані по якій-небудь ознаці. У той же час, у бланку запиту за зразком немає рядка для команди угруповання. Як внести в нього цей рядок?
10. Кожному полю таблиці в бланку запиту за зразком відпові-дає один стовпець. У ньому можна дати команду на розрахунок однієї підсумкової функції. Як зробити чотири підсумкові обчислення за одним полем, наприклад, визначити сумарну, середню, максимальну і мінімальну ціну товарів у деякій групі? (Задача часто зустрічається на практиці.)
11. Як за допомогою запиту провести сортування записів у таблиці за яким-небудь полем?

Завдання

Створити запити за зв'язаними таблицями своєї бази даних: вибіркового за текстовими та числовими даними, у тому числі за складними умовами, видалення, розрахунковий, підсумковий, заміни. Зберегти запити для подальшої роботи з базою даних.

Лабораторна робота № 5

Створення звітів за базою даних

Звіти дозволяють представити й роздрукувати дані відповідно до вимог користувача. Можливості оформлення даних і виведення на друк настільки ж гнучкі, як і можливості їхнього відображення на екрані.

Звіти призначені винятково для виведення даних на друк. У них відсутня необхідність наявності керуючих елементів для введення даних. Тому у звітах можна відмовитися від використання списків, прапорців. У них не можна установити режим таблиці чи форми. Для звіту можна вибрати тільки режим Конструктора і попереднього перегляду.

Автозвіти

Зі звітами зручніше знайомитися в режимі автоматичного створення. Створіть на основі будь-якої таблиці автозвіт у стовпець або стрічковий. Операція настільки проста, що зводиться до одного клацання лівої кнопки миші.

Вибравши в діалоговому вікні База даних вкладку Звіти і клацнувши на кнопці **Создать**, ми відкриваємо діалогове вікно **Новый отчет**, що дозволяє створити звіт автоматично (автоотчет), за допомогою Майстра або вручну.

Звіт за допомогою Майстра створюється в три кроки:

Перейдіть до вкладки **Отчеты**. Натисніть кнопку **Создать**.

Крок 1. Виберіть **Мастер отчетов** і джерело даних (таблицю або запит), натисніть **ОК**.

Крок 2. Виберіть необхідні для звіту поля.

Крок 3. Дозволяє додати рівні угруповання.

Крок 4. Задайте порядок сортування й обчислень, якщо необхідно зробити підрахунок за визначеними полями, натисніть кнопку **Ітоги** й установіть прапорці функції SUM тим полям, за якими необхідно отримати проміжні підсумки, натисніть **Дальше**.

Крок 5. Задайте вигляд Макета звіту.

Крок 6. Задайте стиль звіту.

Крок 7. Задайте ім'я Звіту.

Звіти призначені для виведення інформації на принтер, тому для розрахунку розташування даних на друкованій сторінці програма Access 97 повинна "знати" усе необхідне про особливості принтера. Ці дані Access отримує від операційної системи. Відповідно, принтер у системі повинний бути встановлений.

При відсутності принтера звіти створювати все-таки можна. Досить виконати програмну установку за допомогою команди операційної системи: **Пуск/Настройка/Принтери/Установка принтера**, після чого зареєструвати драйвер принтера, або взявши його з гнучко-го диска, або вибравши один із драйверів, що додаються до самої операційної системи.

Структура звіту

Звіти складаються з розділів, а розділи можуть містити елементи керування.

Зі структурою звіту найпростіше ознайомитися, створивши який-небудь автозвіт, а потім відкривши його в режимі Конструктора.

1. Як видно з рис. 12, структура звіту складається з п'яти розділів: *заголовка звіту, верхнього колонтитула, області даних, нижнього колонтитула й примітки звіту.*

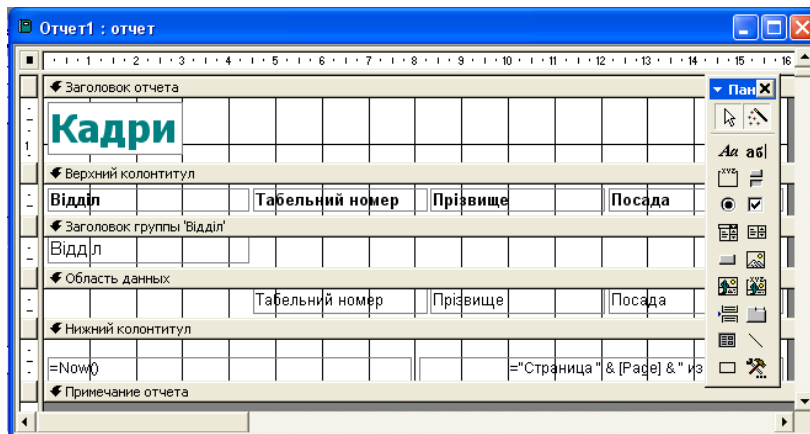


Рис. 12. Звіт у режимі Конструктора

2. Розділ *заголовка* служить для друку загального заголовка звіту.

3. Розділ *верхнього колонтитула* можна використовувати для друку підзаголовків, якщо звіт має складну структуру і займає багато сторінок. Тут можна також поміщати і *колонцифри* (номера сторінок), якщо це не зроблено в нижньому колонтитулі.

4. В *області даних* розміщують елементи керування, зв'язані з вмістом полів таблиць бази. У ці елементи керування видаються дані з таблиць для друку на принтері.

5. Розділ *нижнього колонтитула* використовують для тих самих цілей, що і розділ верхнього колонтитула. У нашому випадку в ньому розміщені два елементи керування.

6. У першому елементі керування виводиться поточна дата. Для цього використана убудована в Access 97 функція Now(). Вона повертає поточну дату і поміщає її в поле, а звіт відтворює її під час друку.

7. У другому елементі керування виводиться номер сторінки і загальна кількість сторінок. Для їхнього визначення використані вбудовані функції Pages(). Той текст, що записаний у лапках, відтворюється "буквально", а оператор & служить для "склеювання" тексту, укладеного в лапки, із значеннями, що повертаються функціями. Оператор & називається *оператором конкатенації*.

8. Розділ *примітки* використовують для розміщення додаткової інформації. У нашому прикладі він не використаний.

Режим перегляду звіту

У режимі перегляду звіту можна переглянути всі сторінки готового звіту, налагодити параметри сторінки друку (п-т меню **Файл/ Параметри сторінки**), змінити масштаб області, що

переглядається, перейти в режим Конструктора звіту,

На панелі інструментів режиму Перегляду звіту присутня кнопка **Связь с Office**. Отриманий звіт можна експортувати в текстовий редактор Word або табличний процесор Excel.

Установки принтера зберігаються разом із звітом, тобто опції для установки кількості стовпчиків і т.д. знаходяться саме в діалого-вому вікні для налагодження друку. Це важливо при необхідності друку звітів на принтерах різних типів.

Як правило, Access автоматично виконує розбивку по сторінках. Правила розбивки встановлюються у властивостях звіту. Однак, використовуючи піктограму **Конец страницы**, можна здійснювати перехід на нову сторінку вручну, тобто виконати розбивку на сторінки відповідно до вимог користувача.

Кожна з областей звіту має свої властивості. **Не разрывать/Да** - намагається надрукувати все на одній сторінці. **Вывод на экран** - відображає на екран дану область звіту. **Расширение и Сжатие/Да** - автоматично підганяється розмір відповідно керуючого елемента.

Дайте відповіді на питання за темою «Звіти»:

1. Для чого призначені звіти?
2. На який пристрій комп'ютерної системи відбувається ви-ведення даних при роботі звіту?
3. З чого складаються звіти?
4. Які розділи звітів ви знаєте?
5. Які способи автоматичної генерації звітів ви знаєте?
6. Що варто вказати програмі Access 97 для автоматичної генерації звіту? (На базі чого виконується звіт?)

7. З якими ще об'єктами крім таблиць, запитів, форм і звітів працює СУБД Access 97? Для чого вони призначені?

Завдання

Створити звіти за своєю базою даних: автозвіт, звіт за допомогою Майстра, Конструктора. Виконати необхідні угруповання даних, розрахунки підсумків за даними. Зберегти звіти для подальшого використання.

Лабораторна робота № 6

Створення форм для обробки даних у Microsoft Access

Форма являє собою електронний бланк, у якому містяться поля для введення даних. Користувач вводить дані в ці поля, і дані автоматично заносяться в таблиці бази. Форми дуже схожі на звіти, у них однакові:

- панелі інструментів;
- панелі елементів;
- засоби для розміщення й роботи з елементами керування;
- можливість використовувати ті ж області, що і при створенні звітів: заголовка, примітки, області верхнього й нижнього колонтитулів, області даних можливість включати області для полів, за якими здійснюється угруповання записів.

Створення форм

Як і інші об'єкти Access 97, форми можна створювати за допомогою майстра форми або вручну, відкривши порожню форму і перетягнувши елементи форми в потрібне місце. Можна сполучити автоматизовану і ручну розробку. Майстер форми задає ряд питань про створювану форму, а потім будує форму в залежності від відповідей. У Access 97 існують наступні можливості створення форм:

- Конструктор;
- Майстер форм;
- Автоформа: у стовпець;
- Автоформа: стрічкова;
- Автоформа: таблична;

- Діаграма (2-х чи багатомірні кругові, лінійні чи штрихові діаграми);
- Зведена таблиця.

Розглянемо послідовність дій для створення форми за допомогою Майстра. Такі форми в основному використовуються для введення даних у таблицю. Перейдіть до вкладки **Форми**, натисніть кнопку **Создать**, виберіть режим **Мастер Форм** зі списку, що розгортається, виберіть джерело даних для форми. За допомогою Майстра форма створюється в чотири етапи:

- 1) вибір полів, до яких можна буде вводити дані у формі;
- 2) вибір зовнішнього вигляду форми (в один стовпець, стрічко-вий, табличний, вирівняний);
- 3) вибір фонового малюнка форми (один із десяти);
- 4) завдання імені форми.

Готову форму можна відразу ж використовувати для перегляду існуючих записів або для введення нових.

Розділи форми

Якщо відкрити форму в режимі Конструктора (рис. 13), то у структурі форми можна чітко виділити три розділи:

- **розділ заголовка** форми (звичайно містить назву форми, назву підзадачі чи пункту меню, може містити логотип фірми, дату і т.д.)

- **область даних**. Елементи керування, що містяться в області даних найчастіше є представленням одного запису таблиці (чи запиту). У цьому випадку говорять про тип форми «у стовпець». Якщо необхідно відобразити одночасно кілька записів, то в цьому випадку використовують стрічкову форму. Усе, що міститься в області даних, є елементами керування і з ними ви познайомитеся в наступному розділі. Фоновий малюнок, що лежить під елементами керування, показує розмір *робочого поля форми*.

- **розділ примітки** форми (може містити підсумкові відомості, пояснення до області даних, спеціальні кнопки навігації, посилання на інші форми і т.д.)

Розміри розділів і розміри робочого поля форми можна змінювати за допомогою миші. При наведенні на границю розділу покажчик змінює форму, після чого границю можна переміщати методом перетаскування.

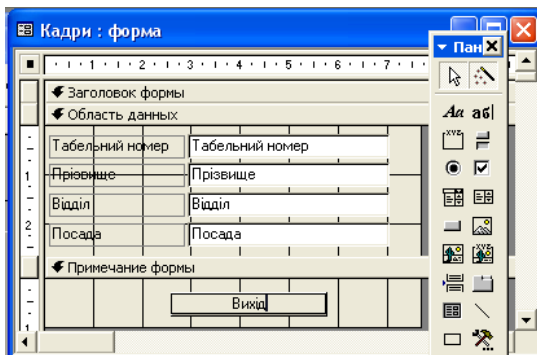


Рис. 13. Форма в режимі Конструктора

Елементи форми

У «формі можна розмістити в довільному порядку ряд елементів. Елементи форми можуть бути зв'язаними й незв'язаними. Зв'язані -прив'язані до вихідної таблиці поля чи запиту. Незв'язані (вільні) - як правило, відображають результат обчислень або є даними (текстами, малюнками й ін.) які залишається незмінними незалежно від того, який запис у цей момент переглядають у формі.

Редагування форм полягає в створенні нових чи зміні наявних елементів керування, а також у зміні їхнього взаємного розташування. Для кожного з елементів форми можна установити свій колір, розмір, місце розташування, а також ряд властивостей, характерних саме для даного елемента. Повний список усіх властивостей елементів можна отримати, відкривши вікно властивостей шляхом вибору однойменної піктограми. Якщо виділити кілька елементів вікна, можна відразу змінити їхні загальні властивості. Можна додати ілюстрації й лінії різного кольору.

Більшість із властивостей елементів вікна можна визначити, використовуючи панель керування, палітру кольорів і техніку буксирування.

Для форматування елемента керування його треба спочатку виділити. Для цього служить інструмент **Выбор объектов**. При виділенні елемента керування докола нього утвориться рамка

з вісьма марке-рами (по кутах і по центрах сторін рамки). Рамку можна розтягувати чи стискати методом перетаскування границь. При наведенні на мар-кер покажчик миші змінює форму, приймаючи зображення відкритої долоні. У цей момент рамку можна переміщувати.

При створенні зв'язаного поля разом із ним одночасно утворює-ться ще один елемент керування - приєднаний надпис. Елемент напис служить для пояснення до даних, відображуваних у поле. Він пере-міщається разом зі зв'язаним полем і утворить із ним єдине ціле. Для того щоб за допомогою буксирування по окремоті переміщувати підписи чи поле, необхідно "захопити" великий квадрат у лівому верхньому куті. При наведенні на нього покажчик миші приймає форму вказівного пальця. У цей момент зв'язане поле можна відірвати від приєданого напису і переміщувати окремо.

Щоб елементи керування розташовувалися у формі рівними рядами, існують спеціальні команди вирівнювання. Спочатку треба виділити групу елементів керування (група вибирається при натисну-тій клавіші *Shift*), а потім дати команду **Формат/Вирівнять** і вибрати метод вирівнювання. Переміщати елементи керування і змінювати їхні розміри за допомогою миші не дуже зручно. Набагато зручніше ви-користовувати для цієї мети курсорні клавіші в комбінації з клавішами *Shift* або *Ctrl*. У першому випадку відбувається зміна розмірів елемен-та керування, а в другому - його розташування.

Крім розглянутих вище елементів керування **Надпись** і **Поле**, існує ще ряд елементів, які ми розглянемо.



Рис. 14. Панель інструментів форми

Щоб при роботі з елементами керування використовувалися програми-майстри, натисніть кнопку

• **Група перемикачів** - використовується для введення значень у числові поля, якщо вони можуть приймати тільки визначені значення. Послідовність дій при створенні цього елемента:

- вибрати елемент *Aa*;
 - відзначити лівий верхній кут рамки й установити необхідний розмір групи;
 - задати поле у Властивості «Дані» групи;
 - розмістити в групі кілька перемикачів;
- для кожного перемикача групи в поле «значення параметра» ввести необхідне значення. Це значення буде введено в поле таблиці при виборі відповідного перемикача.
- **Вимикачі** особливо зручні при використанні в групах. У такій групі легко бачити, який з вимикачів натиснутий. Замість під-пису на вимикач можна помістити малюнок.
- **Перемикачі** звичайно використовуються в групі для відобра-ження набору параметрів, з яких необхідно вибрати один. З цими елементами можна зв'язати команди, наприклад, що виконують фільтрацію.

• **Прапорці** використовуються для введення значень у логічні поля. Якщо прапорець позначено галочкою , то значення логічного поля = Істина (1 ,Так). У протилежному випадку = Неправда (0, Ні).

Прапорці також можна розмістити в групі. Вони діють аналогіч-но перемикачам, але на відміну від них, допускають множинний вибір.

• **Списки й Поля зі списком** використовуються для організації введення даних, вибираючи значення зі списку чи значень таблиці даних. Ці елементи дуже зручні при введенні даних із зв'язаних таблиць. Послідовність дій при створенні полів із списком наступна:

вибрати піктограму "Поле із списком" (або відповідно "Список");

- слідувати вказівкам Майстра;
- "Джерело рядків" визначає походження даних, що будуть відо-бражатися в списку.

Поля зі списком займають менше місця на формі, а список їхніх значень виводиться на екран тільки за вказівкою користувача. Списки займають - більше місця, однак у них завжди відображається трохи можливих для введення значень.

• **Прямокутник** служить для виділення підпису поля разом із полем, для виділення декількох елементів форми. Після вибору пікто-грами навколо обраних об'єктів малюється рамка.

Для того щоб прямокутник не закривав розташовані під ним елементи, із меню варто вибрати команду **Формат/На задній план**. Після цього можна змінити розташування, колір, розмір прямокутника.

• **Командні кнопки.** Додавання їх у форму можна здійснювати за допомогою майстра командних кнопок:

- вибрати елемент "Кнопка";
 - клацнути в тім місці, де необхідно помістити верхній лівий кут командної кнопки;
- слідувати вказівкам Access для розміщення потрібного елемента керування.

Ці кнопки використовуються для виконання операцій навігації, керування даними й ін. (наприклад, перейти на новий запис, чи відкрити, закрити форму, видалити запис)

• **Малюнки й інші об'єкти.** Прості графічні об'єкти - лінії, прямокутники включаються у форму за допомогою панелі інструментів. OLE технологія дозволяє включити у форму ілюстрації, фото-графії, навіть відеозапис, і музичний фрагмент, голосове повідомлення.

Існують два типи полів для розміщення об'єктів OLE: "Вільна рамка об'єкта" і "Приєднана рамка об'єкта". У першому випадку рамка не зв'язана ні з яким полем таблиць бази даних. Об'єкт, що знаходиться в ній, виконує роль ілюстрації і служить для оформлення форми. З "Приєднаною рамкою" зв'язане одне з полів таблиці. У ній відображається вміст цього поля. Цей вміст може мінятися при переході від одного запису до іншого.

Наприклад, щоб малюнок вміщався в рамку поля, виділіть поле OLE, натисніть праву кнопку миші й у контекстному меню виберіть пункт **Свойства**. У вікні властивостей виберіть пункт **Установка розміров**, із списку, що розкривається, виберіть значення **"По розміру рамки"**. Закрийте форму і підтвердіть збереження змін.

• **Вкладинки.** Дозволяють розмістити багато інформації на обмеженій площі. На вкладинках розміщують інші елементи керування.

На рисунку представлена форма з елементами керування.

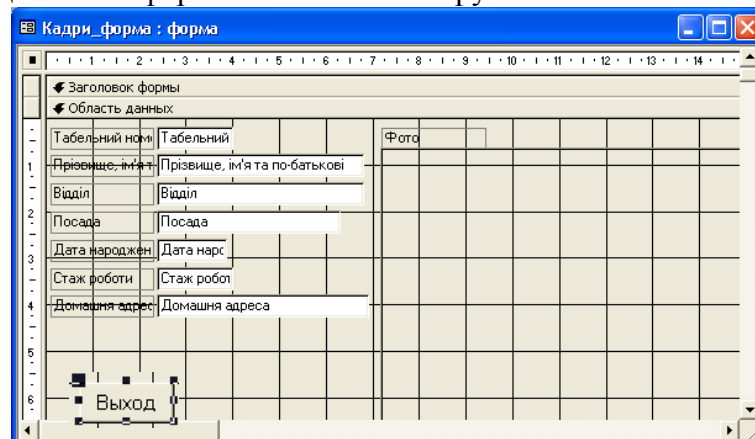


Рис. 15. Оформлення форми елементами керування

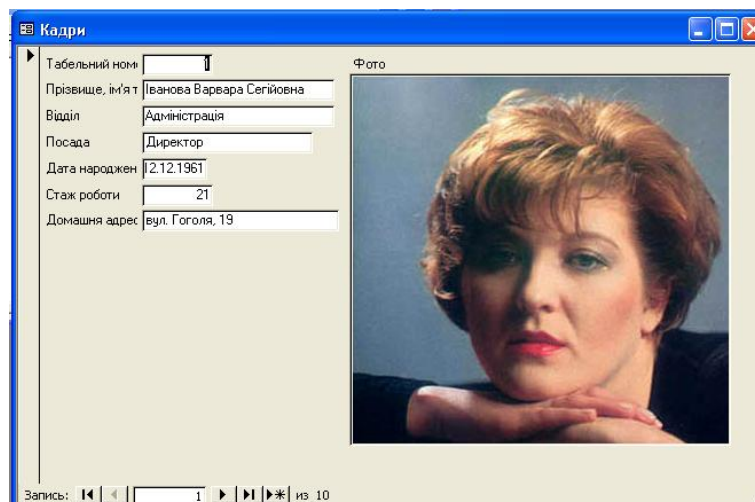


Рис. 16. Та ж форма в режимі представлення

Дайте відповіді на питання за темою «Форми»:

1. Для чого призначені форми?
2. Які методи автоматичного створення форм ви знаєте?
3. З чого складаються форми?
4. Які розділи форм ви знаєте?
5. Які елементи керування форм ви знаєте?
6. При створенні елемента керування форми поруч із ним авто-матично утвориться елемент керування, що називається приєднаним надписом. Як відірвати цей надпис від елемента керування, якщо розроблювач не хоче, щоб він розташовувався ліворуч, а бажає, наприклад, помістити його нижче елемента керування?
7. Якою командою можна вирівняти елементи керування форми, розташовані неакуратно?
8. При заповненні таблиць зручно не вводити часто повторювані дані, а вибирати їх із задалегідь заготовленого списку. Це не тільки полегшує роботу користувачів, але і дозволяє уникнути помилок у найбільш відповідальних місцях. Як називаються елементи керування форми, що дозволяють це зробити?
9. Чи можна використовувати форми не тільки для введення, а і для виведення даних? Якщо так, то, на який пристрій комп'ютерної системи виконується це виведення?

Створення меню для роботи з об'єктами бази даних

При створенні бази даних розробник може організувати меню програми, що допомагає переміщатися по базі даних. Існує два способи організації таких форм:

- за допомогою створення порожньої форми й розташування на ній керуючих елементів (кнопок) для виклику форм, уведення даних або відкриття звітів;
- за допомогою диспетчера кнопочних форм, що дозволяє створювати кнопочні форми, подібні тим, що створює майстер баз даних.

Розглянемо запропоновані способи організації меню на наступному прикладі.

Припустимо, у базі даних «Кадри» розроблені:

- форми введення «Основні відомості», «Довідник посад», «До-відник відділів»;
- запити;
- звіти.

Необхідно створити меню, що дозволяло б легко переміщуватися між формами й звітами бази даних.

Організація меню на основі порожньої форми

Перейдіть до вкладки **Форми**, натисніть кнопку **Создать**, виберіть режим створення форми **Конструктор**.

1. На панелі елементів виберіть елемент керування **Кнопка**, розмістіть його в розділі **Область даних** порожньої форми. Запускає-ться програма-майстер.

Крок 1 - виберіть Категорії - **Работа с формой**; Дії – **Открытие формы**, натисніть **Дальше**;

Крок 2 - задайте ім'я форми, яка буде викликана шляхом натискання даної кнопки, наприклад, «Основні відомості», натисніть **Дальше**;

Крок 3 - задайте: **Открыть форму і Показать все записи**, натисніть **Дальше**;

Крок 4 - задайте підпис для кнопки (наприклад, «Довідник посад») або виберіть малюнок;

Крок 5 - задайте ім'я кнопки (ім'я змінної Access привласнює автоматично, однак ви можете привласнити своє, використовуючи правила написання імен змінних), натисніть **Готово**.

2. Створіть аналогічно кнопку для виклику форми «Довідник відділів».

3. Створіть кнопки для виклику звіту «Список співробітників»:

Крок 1 - виберіть Категорії - **Работа с отчетом**, Дії - **Просмотр отчета**, натисніть **Дальше**;

Крок 2 - виберіть ім'я звіту, що буде викликаний натиском даної кнопки, наприклад, «Список співробітників», натисніть **Дальше**;

Крок 3 - задайте підпис для кнопки (наприклад «Список спів-робітників») або виберіть малюнок;

Крок 4 - задайте ім'я кнопки (ім'я перемінної Access привласнює автоматично, однак ви можете привласнити своє), натисніть **Готово**.

4. Створіть кнопки для виклику інших звітів.

5. Створіть кнопку для виходу з додатка:

Крок 1 - виберіть Категорії – **Додаток**, Дії - **Выход из приложения**, натисніть **Дальше**;

Крок 2 - задайте підпис для кнопки (наприклад, «Вихід») або виберіть малюнок;

Крок 3 - задайте ім'я кнопки, натисніть **Готово**.

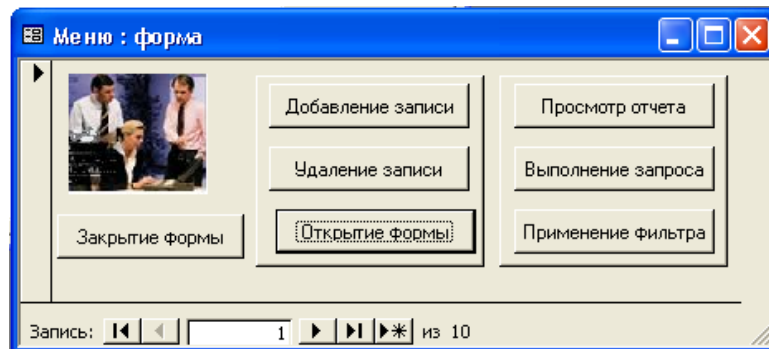


Рис. 17. Готова форма-меню на основі пустої форми

Організація форми меню за допомогою Диспетчера кнопочних форм

За допомогою майстра автоматично створюється кнопочка форма. На кнопочній панелі розміщують кнопки, при натисканні яких відкриваються форми чи звіти (чи відкриваються інші кнопочні форми, за допомогою яких відкриваються інші форми чи звіти), здійснюється вихід з Microsoft Access чи змінюється сама кнопочка форма.

Для створення кнопочної форми за допомогою диспетчера кнопочних форм виберіть у меню **Сервис** команду **Надстройки** й під-команду **Диспетчер кнопочных форм**. Якщо виводиться запрошення підтвердити створення кнопочної форми, натисніть кнопку **Да**.

Для нашого приклада будемо використовувати три сторінки кнопочної форми, на яких розмістимо кнопки для виклику відповідних об'єктів. Сторінки «Вихідні дані» і «Звіти» створюються за допомогою кнопки **Создать**.

1. У діалоговому вікні **Диспетчер кнопочных форм** виберіть потрібну сторінку і натисніть кнопку **Изменить**.

2. У діалоговому вікні **Изменение страницы** кнопочної форми введіть ім'я в поле **Название кнопочной формы** і натисніть кнопку **Создать**.

3. У діалоговому вікні **Изменение страницы** кнопочної форми введіть текст підпису для першої кнопки і виберіть команду в полі зі списком **Команда**. Наприклад, уведіть текст Основні відомості про співробітників, а потім виберіть у поле **Команда** команду **Открытие формы в режиме редактирования**.

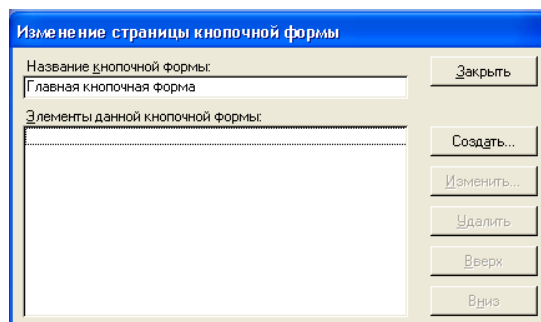


Рис. 18. Вікно Диспетчера кнопочных форм

4. Для більшості обраних команд під полем **Команда** відкривається нове поле зі списком. При необхідності, виберіть потрібний елемент у цьому полі. Наприклад, якщо обрана одна з команд роботи з формами, то в поле зі списком **Форма** варто вибрати ім'я форми, що відкривається.

5. Повторіть кроки з 4 по 6 для створення всіх потрібних кнопок кнопочної форми. Щоб змінити чи видалити яку-небудь із створених кнопок, виберіть ім'я цієї кнопки в списку **Элементы данной кнопочной формы** і натисніть кнопку **Изменить** або **Удалить**. При необхідності змінити порядок кнопок у списку виберіть елемент і натисніть кнопку **Вверх** або **Вниз**.

6. Закінчивши створення кнопкової форми, натисніть кнопку **Закрить**.

Диспетчер кнопкових форм дозволяє створювати кнопку форму, кнопки якої відкривають інші кнопкові форми. За допомогою описаних вище дій можуть бути створені кілька кнопкових форм. Для того щоб зробити кнопку форму стартовою кнопковою формою, що відкривається за замовчуванням при запуску бази даних, виберіть ім'я кнопкової форми у діалоговому вікні **Диспетчер кнопкових форм** і натисніть кнопку **По умовчанням**. Для того щоб відкрити з кнопкової форми іншу кнопку форму, виберіть на кроці 5 команду **Переход к кнопковій формі** вкажіть ім'я кнопкової форми, на яку потрібно перейти.

При створенні кнопкової форми за допомогою Диспетчера кнопкових форм Microsoft Access створіть таблицю **Елементи кнопкової форми**, що утримує опис кнопок, що виводяться у формі, і дій, що ними виконуються. Спроба змінити кнопку форму в режимі конструктора форми, може привести до того, що додаток перестане працювати. Якщо потрібно внести в кнопку форму істотні зміни, рекомендується створити кнопку форму заново, а потім указати нову форму в якості стартової форми.

Завдання:

1. Створити форму за своєю базою даних, що служить для редагування й додавання записів даних, використовуючи автоформу.
2. Створити форму для виконання запитів вибірки й розрахунків за допомогою Конструктора форм. Передбачити у формі необхідні поля для введення та виведення даних, елементи керування, необхідні пояснення та оформлення.
3. Зберегти створені форми для подальшої роботи з базою даних.
4. Створити кнопку форму-меню для сумісної роботи з об'єктами бази даних, що були створені у попередніх лабораторних роботах.

Лабораторна робота №7

Вивчення інтерфейсу СУБД Microsoft Visual FoxPro.

Створення бази даних та таблиць даних

Основні максимальні можливості системи Visual FoxPro

Найменування граничної величини	Гранична величина
Кількість записів у файлі таблиці	1 мільярд
Розмір файлу таблиці	2 гігабайти
Кількість символів в одному записі	65500
Кількість полів в одному записі	255
Кількість одночасно відкритих таблиць	255
Кількість символів у полі таблиці	254
Кількість відкритих індексних файлів для однієї таблиці	не обмежено
Кількість відкритих індексів у всіх робочих областях	не обмежено
Кількість зв'язків	не обмежено
Довжина виражень зв'язку	не обмежена
Розмір символічних полів	254
Розмір числових полів	20
Кількість символів в імені поля у вільній таблиці	10
Кількість символів в імені поля в таблиці, що міститься в базі даних	128
Діапазон цілих чисел	+ 2 147 483 647
Точність у числових обчисленнях	16 цифр до 9007199254740992 (2^{53})

Дійсні числа	до 10^{308} або 2^{1023}
Кількість змінних за замовчуванням	16384
Кількість змінних	65000
Кількість масивів	65000
Кількість елементів у масиві	65000
Кількість рядків у вихідних програмних файлах	не обмежено
Розмір модуля програми, що компілюється	64 кілобайти
Розмір процедур у файлі	не обмежений
Кількість вкладених операторів DO	128
Кількість вкладених операторів READ	5
Кількість переданих параметрів	27
Кількість об'єктів у звіті	не обмежено
Довжина опису звіту	20 дюймів
Кількість рівнів угруповання	74
Довжина символічних змінних у звіті	255
Кількість відкритих вікон (усіх типів)	не обмежено
Кількість відкритих вікон BROWSE	255
Кількість символів у символічному рядку або у змінній пам'яті	16 777 184
Кількість символів у командному рядку	8192
Кількість відкритих файлів	можливості ОС
Кількість натискань клавіш у макро	1024
Кількість полів в одному запиті SQL	255

Основні елементи інтерфейсу СУБД Visual FoxPro представлені на рис. 1:

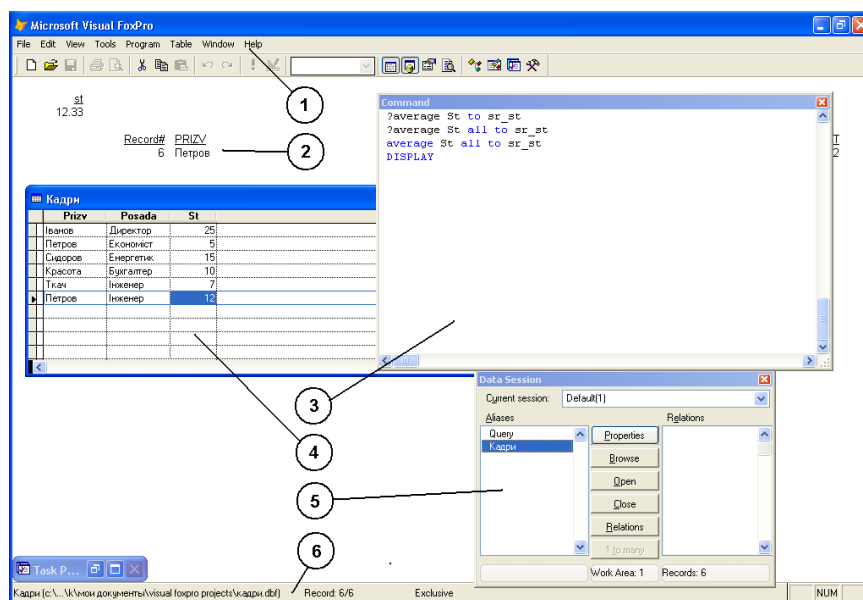


Рисунок 1 – Елементи інтерфейсу СУБД Visual FoxPro:

1 – меню команд; 2 – головне вікно; 3 – вікно команд (Command); 4 – таблиця даних; 5 – вікно сеансу даних (Data Session); 6 – рядок стану.

Завдання:

1. Ознайомитися з інтерфейсом СУБД Visual FoxPro: меню команд, панелі інструментів, головне вікно, вікно команд, рядок стану.
2. Створити базу даних та першу таблицю даних згідно варіантам:
 - 2.1. Командою File/ New створити послідовно базу даних та таблицю. Дати базі даних та

таблиці імена, вказані в індивідуальному завданні (див. лабораторну роботу №1). Всі об'єкти бази даних зберігати в окремій створеній папці.

2.2. Описати структуру таблиці у вікні Table Designer. Увести короткі та зрозумілі імена полів латиницею. Вибрати тип даних для кожного з полів таблиці. Для числових полів вказати точність (Decimals – кількість знаків після десятичної крапки). Натиснути ОК.

2.3. У вікні Command подати команду APPEND та ввести почерзі 12 записів, що відповідають змісту таблиці, взявши дані на власний розсуд.

2.4. Закрити таблицю командою USE та відкрити її знову командою USE шлях\ім'я таблиці, наприклад, USE c:\Студент\Лобода\ Співробітник\Кадри.dbf SHARED (SHARED – повний доступ).

3. Виконати переміщення за таблицею такими способами:

Задача переходу	Меню команд	Вікно Command
Перший запис таблиці	Table/ Go To Record/ Top	GO 1
Останній запис таблиці	Table/ Go To Record/ Bottom	GO BOTTOM
Запис 5	Table/ Go To Record/ Record # ...	GO 5
Наступний запис	Table/ Go To Record/ Next	
Попередній запис	Table/ Go To Record/ Previous	
На три записи вперед		SKIP 3
На чотири записи назад		SKIP -4

4. Командою File/ New/ Text створити текстовий файл. Скопіювати команди з вікна Command у текстовий файл та зберегти його під довільним ім'ям.

Питання для самоконтролю:

1. Призначення й основні характеристики СУБД FoxPro ?
2. Які типи полів використовуються при створенні файлу БД ?
3. Як здійснюється створення файлу БД із використанням меню?
4. Як здійснюється створення файлу БД із використанням командного вікна ?
5. Як здійснюється створення структури файлу БД ?
6. Як здійснюється введення даних ?
7. Як здійснюється перегляд записів файлу БД ?
8. Як здійснюється доповнення записів у файл БД?
9. Як здійснюється переклад покажчика запису?
10. Як здійснюється перегляд конкретного запису?
11. Як здійснюється перегляд визначеної кількості записів, починаючи з конкретної ?
12. Як здійснюється перегляд записів до кінця БД, починаючи з конкретної?
13. Як здійснюється перегляд усіх записів по конкретних полях?
14. Як здійснюється редагування записів по конкретних умовах?

Лабораторна робота №8

Visual FoxPro. Режими перегляду та редагування записів даних. Робота з полями таблиці. Використання індексних файлів

Завдання:

1. Переглянути таблицю у різних режимах командами меню View/ Browse та View/ Edit, з вікна Command командою LIST. Вивести у головне вікно дані у заданих діапазонах:

Діапазон виведення	Команда у вікні Command
Поточний запис	DISPLAY
Перший запис	LIST RECORD 1
Наступні 5 записів	LIST NEXT 5
Структура таблиці	LIST STRUCTURE

2. Відредагувати дані у наступних діапазонах:

Діапазон редагування	Команда у вікні Command
----------------------	-------------------------

Структура таблиці	MODIFY STRUCTURE
Перший запис	EDIT 1
Наступні 5 записів	EDIT NEXT 5
Вибрані поля таблиці	CHANGE FIELDS Prizv, Oklad
Вибрані поля у вибраному записі	CHANGE FIELDS Prizv RECORD 8

3. Виконати пошук записів, що задовольняють заданій умові за допомогою команди Table/ Go to Record/ Locate... або LOCATE FOR <умова> у вікні Command. Вивести знайдений запис у головне вікно командою DISPLAY. Продовжити пошук командою CONTINUE. Виконати пошук за числовими та за символічними даними.

4. Вивчити режими та методи вилучення записів даних з таблиці:

4.1. Командою меню Table/ Delete Records або DELETE FOR <умова> у вікні Command відзначити декілька записів для вилучення у різних діапазонах (RECORD, NEXT, REST) із вказанням умов для даних числового та символічного типу.

4.2. Відновити записи командою Table/ Recall Records або RECALL FOR <умова> у вікні Command.

2.3. Відзначити один із записів для вилучення за допомогою миші. Вилучити відзначений запис остаточно командою Table/ Remove Deleted Records або PACK у вікні Command.

5. Виконати упорядкування табличних даних за двома полями у порядку зростання значень та у зворотньому порядку:

SORT TO t1 ON Prizv, Oklad

SORT TO t2 ON Prizv/d, Posada/d.

Відкрити вікно Data Session одноіменною командою меню Window. Відкрити у робочих областях 2 і 3 отримані таблиці. Переглянути таблиці, перевірити результати сортування.

6. Виконати вибірку даних, що задовольняють: а) числовій умові; б) співпадають з заданою послідовністю символів:

COPY TO t5 FOR Oklad>500 FIELDS Prizv, Oklad

COPY TO t6 FOR Posada="інженер" FIELDS Prizv, Posada.

Переглянути отримані таблиці, перевірити результати вибірки.

7. Проіндексувати таблицю за ключовим полем та за будь-яким числовим полем:

INDEX ON Prizv TO t7

INDEX ON Oklad TO t8

8. Виконати швидкий пошук символічних та числових даних по індексованим полям:

SEEK "Карпенко"

FIND 1200

Контрольні питання:

1. Як здійснюється активізація файлу БД ?
2. Як здійснюється перегляд записів по конкретним даної числових полів ?
3. Як здійснюється видалення полів файлу БД ?
4. Як здійснюється додавання полів у структуру файлу БД ?
5. Як здійснюється перегляд структури файлу БД ?
6. Як здійснюється створення файлу БД, відсортованого за даними символічного поля за абеткою?
7. Як здійснюється створення файлу БД, відсортованого за даними символічного поля за алфавітом у зворотньому порядку ?
8. Як здійснюється створення індексного файлу, відсортованого за даними числового поля за зростаючому порядку ?
9. Як здійснюється активізація індексних файлів ?
10. Як здійснюється пошук першого запису в індексному файлі за даними числових полів?
11. Як здійснюється пошук першого запису в індексному файлі за даними символічних полів?
12. Як здійснюється вставка нового запису після активної? Як здійснюється вставка нового запису перед активної? Як здійснюється вставка одного порожнього запису?
13. Як здійснюється позначка на видалення всіх записів БД?

14. Як здійснюється позначка на видалення конкретного запису БД?
15. Як здійснюється позначка на видалення декількох записів БД, починаючи з активної?
16. Як здійснюється відмовлення від видалення позначених записів БД?
17. Як здійснюється фізичне видалення позначених записів БД?

Лабораторна робота №9

Тема: Робота з декількома таблицями одночасно. Виконання розрахунків у таблицях Використання математичних функцій

Завдання:

1. Створити другу за завданням таблицю шляхом копіювання структури першої таблиці у межах співпадаючих полів:

`COPY STRUCTURE TO` кадри FIELDS prizv,posada

2. У другій робочій області відкрити створену таблицю:

`SELECT 2`
`USE` кадри

3. Модифікувати структуру створеної таблиці. Додати у структуру поля, яких не вистачає:

`MODIFY STRUCTURE`

Додати до поточної таблиці дані з першої таблиці:

`APPEND FROM` бухгалтерія FIELDS prizv,posada

Заповнити у другій таблиці даними інші поля.

4. Проіндексувати обидві таблиці по ключовому полю:

`SELECT 1`
`INDEX ON` prizv `TO` t1
`SELECT 2`
`INDEX ON` prizv `TO` t3

5. Налагодити зв'язок між таблицями по ключовому полю. Використати вікно Data Session або команди:

`SELECT 1`

`SET RELATION TO` бухгалтерія.prizv `INTO` кадри `ADDITIVE`

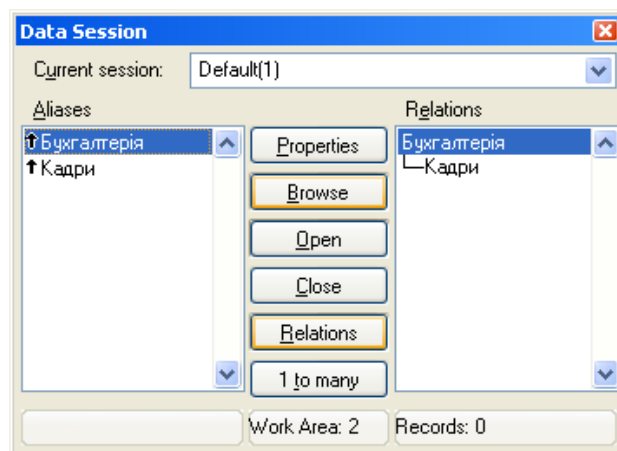


Рисунок 2 – Налаштування зв'язку між таблицями

6. Виконати заміни символічних та числових даних у полях таблиці. Виконати розрахунки у розрахункових полях:

`REPLACE` avans `WITH` oklad*0.4
`BROWSE LAST`

REPLACE ALL avans WITH oklad*0.4
REPLACE ALL prizv WITH "Сидоренко" FOR prizv="Сидоров"

7. Виконати розрахунки за числовими полями таблиці. Визначити суми, середні значення та кількості записів, що задовольняють заданим діапазнам та умовам:

```
SUM oklad TO s
SELECT 1
SUM oklad TO s
?s
AVERAGE oklad TO sr
?sr
?s
GO 2
SUM oklad NEXT 2 TO s1
?s1
SUM oklad FOR oklad>1500
SUM oklad FOR oklad>1500 TO s2
GO 3
AVERAGE NEXT 2 TO sr1
AVERAGE oklad NEXT 2 TO sr1
COUNT TO k
?k
COUNT FOR oklad>1500 TO k1
?k1
?s/k
COUNT FOR posada="Інженер" TO k2
?k2
SUM oklad FOR posada="Інженер" TO s4
?s4/k2
```

8. Вивчити математичні функції та арифметичні оператори. Виконати з їх допомогою розрахунки:

Числові оператори:

Оператор	Дія	Запис
()	Групування виразів	(4-3) * (12/nVar2)
**, ^	Піднесення до ступеню	? 3 ** 2 ? 3 ^ 2
*, /	Добуток та ділення	? 2 * 7 ? 14 / 7
%	Залишок від ділення	? 15 % 4
+, -	Додавання та віднімання	? 4 + 15

```
?SIN(30)
?s4/k2
?(35/5+25/5)*10
?sqr(2)
?3.5^2.8
?(35/5+25/5)*10
?TAN(60)^2
? MOD(36,10)
? MOD((4*9), (90/9))
? 36 % 10
```

Контрольні питання:

1. Як здійснюється копіювання структури файлу БД?

3. Як здійснюється копіювання окремих полів файлу БД?
4. Як здійснюється копіювання записів файлу БД по конкретних умовах символічних полів?
5. Як здійснюється копіювання записів файлу БД по конкретних умовах числових полів?
7. Як здійснюється видалення файлу БД із використанням команди ERASE?
8. Як здійснюється видалення файлу БД із використанням команди DELETE FILE?
9. Як здійснюється зміна даних числових полів усіх записів файлу БД?
10. Як здійснюється зміна даних числових полів файлу БД по заданих умовах?
11. Якими командами необхідно користатися при редагуванні записів файлу БД командою BROWSE?

Лабораторна робота № 10

Обробка символічних даних і даних типу дата

Символьні дані - константи й змінні - визначаються як дані типу CHARACTER.

Символьна константи обмежуються знаком апострофу або подвійних лапок. Наприклад: ' 55-30-48', "Об'ява".

Символьна значення можуть порівнюватися між собою. Порівняння здійснюється зліва направо й припиняється, коли знаходяться неоднакові символи. Значення, яке має "більший" наступний символ, вважається більшим, із двох символів "більшим" вважається той, який розміщується далі в кодовій таблиці:

символи ! " # \$ % @ ' () * + , =

цифри 0...9

знаки : ; < = > ? @

прописні літери латинського алфавіту. A...Z

рядкові літери латинського алфавіту: a...z

прописні літери кирилиці: А... Я

рядкові літери кирилиці: а...я.

Крім цього діє правило: теля введення команди SET EXACT ON додання будь-якого додаткового символу до одного із рівних значень робить його більшим. По умовчанню встановлюється SET EXACT OF. У цьому режимі кількість символів, що порівнюються, визначається довжиною символічного рядка, розміщеного в логічній умові праворуч. Наприклад, у порівнянні "Іванов" = "Ів" аналізуються лише два символи із результатом "Т" - True (істина).

Приклад: у режимі SET EXACT OF
 Порівняння: ? 'A B C D E' = 'A B'
 Результат: .Т.
 у режимі: SET EXACT ON
 Порівняння: ? 'A B C D E' = 'A B'
 Результат: .F.

Над символічними значеннями може виконуватись операція зчеплення (конкатенація), яка позначається знаком +.

Приклад : ? 'Іванов' + 'С. '+'А.'

Іванов С. А.

NAME = "Коваленко С.І."

Коваленко С, І.

GR = "ФК 05-1"

ФК 05-1

? NAME + " - " + GR

Коваленко С. І. - ФК 05-1

Для обробки символічних значень широко використовуються спеціальні функції.

Для приведених прикладів по вивченню функцій використовується база даних INFO.DBF:

USE INFO.DBF
LIST STRUCTURE

Структура БД: C:\ INFO.DBF

Число записів даних: 4

Дата останнього відновлення: 11/29/07

Поле	ім'я поля	Тип	Ширина
1	PR	CHARACTER	15
2	IM	CHARACTER	15
3	PO	CHARACTER	15
4	RN	DATE	8
5	GR	CHARACTER	8

Зміст БД INFO.DBF:

LIST

№ запису	PR	IM	PO	RN	GR
1	Коваленко	Леонід	Миколайович	01/13/87	ФК 05-1
2	Демура	Сергій	Леонідович	04/30/88	ОА 04-1
3	Сергатий	Олег	Васильович	09/11/87	ФК 05-2
4	Коваль	Сергій	Іванович	02/25/87	ФК 04-2

Функція LEN() дозволяє визначити довжину символьного рядка, який є аргументом цієї функції,

Приклади: C = "ОА 04 - 1"

ОА 04 - 1

? LEN(C)

6

USE INFO.DBF

? LEN(PR)

15

Функції для обробки символьних значень:

TRIM знищує в символьному рядку "хвостові" прогалини:

Приклад: USE INFO.DBF

GOTO 4

? PR+IM+PO

Коваль Сергій Іванович

? TRIM (PR) +' ' + TRIM (IM)+' ' +PO

Коваль Сергій Іванович

SUBSTR дозволяє із рядка-аргумента (аргумент 1) виділити, починаючи з символу під номером M (аргумент 2), підрядок довжиною L символів (аргумент 3): SUBSTR (<рядок>,M,L).

Приклад: B="знання-сила"

знання-сила

? SUBSTR(B,8,4)

сила

AT дозволяє визначити вхід підрядка (аргумент 1) у рядок (аргумент 2): AT(<підрядок>,<рядок>);

Приклад: A="сила"

сила

B="знання-сила"

знання-сила

? AT(A,B)

8

UPPER використовується для заміни в даному рядку-аргументі рядкових літер на прописні;

Приклад: A="Ткаченко"

Ткаченко

? UPPER(A)

ТКАЧЕНКО

LOWER виконує дно, протилежну дії функції UPPER;

Приклад: ? LOWER('ТОВАР')

товар

SPACE використовується для формування символного значення із заданої кількості прогалін

Приклад: ?'H'+SPACE(9)+'K'

Н К

Функції для обробки даних типа ДАТА:

STOD призначена для перетворення символного значення в значення типу DATE:

Приклад: ? STOD('01/01/08')

01/01/06 (у форматі мм/дд/рр)

DATE () дозволяє одержати значення дати, яку було введено при завантаженні операційної системи:

Приклад: ? DATE ()

04/15/06

CDOW формує для заданої дати символне значення дня тижня:

Приклад: ? CDOW (A)

Monday

DOW формує числове значення номера дня тижня:

Приклад: ? DOW (A)

2

DAY виділяє із заданої дати символи, які вказують на день місяця (дд) і переводить ці символи в числове значення:

Приклад: B=STOD('04/16/06')

04/16/06 ? DAY(B)

16

MONTH формує числове значення номера місяця в році:

Приклад: ? MONTH (B)

4

СМОНТН формує для заданої дати символічне значення назви місяця:

Приклад: ? СМОНТН (B)

April

YEAR виділяє із заданої дати два останніх символи року (pp) і формує числове значення року у форматі 20pp:

Приклад: ? YEAR (A)

2006

DТОС виконує протилежну дію.

Завдання:

1. Створити структуру таблиці, яка використовується для прикладу в методичних порадах цієї лабораторної роботи з іменем АТЕ5Т1.dbf.
2. Увести 3 записи з даними про себе і про студентів, які сидять ліворуч і праворуч від вас.
3. Модифікувати створену таблицю, доповнивши її полями, які дадуть змогу ввести оцінки по п'ятибальній системі за результатами першої атестації, а також підрахувати середній бал і успішність.
4. Відредагувати таблицю АТЕ5Т1.dbf і підрахувати середній бал та успішність.
5. Виконати всі приклади, які приведені в методичних вказівках по даній роботі.

Контрольні питання:

1. Який тип даних описує послідовність символів?
2. Який тип даних описує дати?
3. Як вилучити кінцеві прогалини у послідовності символів?
4. Як підрахувати кількість символів у послідовності символів?
5. Як змінити рядкові символи на прописні та навпаки?
6. Як перетворити символічний вираз у дату?
7. Як із дати отримати день тижня, місяць?
8. Якою функцією визначити поточну дату?

Лабораторна робота № 11

Створення запитів за базою даних

Завдання:

1. Створити запити за таблицею даних:
 - а) за одним числовим полем;
 - б) за одним символічним полем;
 - в) за комбінацією символічних та числових полів таблиці.Приклад:
 - 1) Подаємо в меню команд команду File/ New/ Query або у вікні команд CREATE QUERY;
 - 2) Вибираємо у списку таблиць потрібну таблицю для запита;
 - 3) Кнопкою Add вибираємо поля таблиці, за якими будуть вказуватися критерії пошуку;
 - 4) Вибираємо вкладинку Filter. У стовпчику Field Name вибираємо ім'я потрібного поля, у стовпчику Criteria – логічний оператор (=, >, < і т.д.), у стовпчику Example – зразок для порівняння;
 - 5) Подати команду Query/ Run Query;
 - 6) Зберегти запит.

	Prizv	Posada	St
Іванов	Директор		25
Петров	Економіст		5
Сидоров	Енергетик		15
Красота	Бухгалтер		10
Ткач	Інженер		7
Петров	Інженер		12

Рисунок 3 – Початкова таблиця

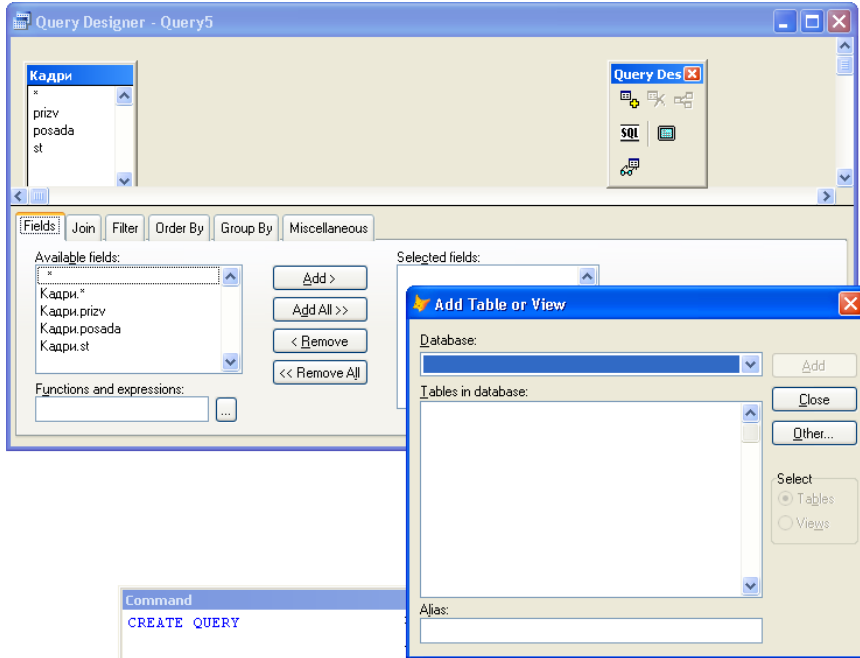


Рисунок 4 – Вибір таблиці та полів для запити

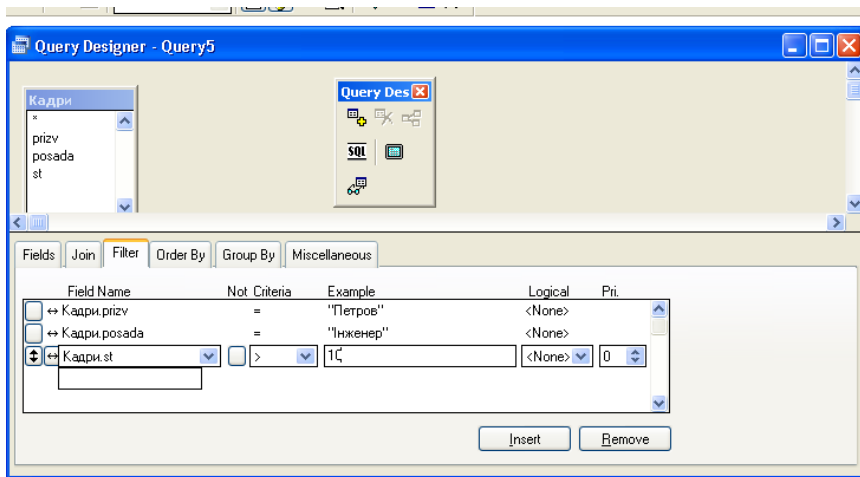


Рисунок 5 – Створення запити

	Prizv	St	Posada
Петров		12	Інженер

Рисунок 6 – Результат виконання запити

Лабораторна робота №12

Створення командних файлів

Завдання:

Створити програму (командний файл) командою File/ New/ Program або MODIFY COMMAND у вікні команд. Увести текст програми. У програмі передбачити операції обробки числових даних, заміни даних, форматного виведення результатів на екран.

Приклад:

```
CLOSE ALL                (Закрити всі файли)
USE "c:\program files\visual foxpro projects\кадри.dbf" exclusive
                        (Відкрити свою таблицю даних)
CLEAR                    (Очистити екран)
LIST                     (Вивести записи даних на екран)
COUNT TO kilk           (Підрахувати кількість записів даних)
AVERAGE St TO sr        (Розрахувати середнє)
@ 11,5 say "Кількість співробітників="
                        (Вивести результати на екран)

@ 12,5 say kilk
@ 13,5 say "Середній стаж роботи="
@ 14,5 say sr picture '99.9'
REPLACE ALL st WITH st+1 (Замінити дані у полі)
LIST
RETURN                  (Повернутися з програми)
```

Виконати програму командою Program/ Do або DO ім'я програми у вікні команд.
Зберегти програму.

Контрольні питання:

1. Для чого призначений запит до бази даних?
2. Що зберігається у файлі запиту?
3. Який порядок створення запиту?
4. Що називається командним файлом?
5. Як створити новий командний файл?
6. Як запустити командний файл на виконання?
7. Як повернутися із командного файлу до діалогового режиму?
8. Як вивести значення змінної у рядок 10 і колонку 10 екрану?
9. Як виконати форматне виведення даних?
10. Які є способи введення даних у програмах?

Лабораторна робота № 13

Створення меню команд для обробки бази даних

Завдання:

Створити програму для обробки таблиці даних за своїм варіантом. У програмі передбачити меню для вибору варіантів обробки даних та команди, що забезпечують відповідні операції над даними.

Приклад:

```
CLOSE ALL
USE "c:\Program Files\visual foxpro projects\Бухгалтерія.dbf" exclusive
CLEAR
LIST
WAIT
DO WHILE .t.             (Відкриття циклу «Поки істина»)
vibor=0                  (Задаємо змінну меню)
```

@ 11,5 SAY "Виберіть режим обробки даних:" (Повідомлення)
 @ 13,5 PROMPT "Додати новий запис" (Варіанти режимів обробки)
 @ 15,5 PROMPT "Редагувати дані"
 @ 17,5 PROMPT "Упорядкувати дані"
 @ 19,5 PROMPT "Пошук даних"
 @ 21,5 PROMPT "Вихід"
 MENU TO vibor (Присвоєння номеру вибраного варіанта змінній)
 DO CASE (Вибір операцій обробки у відповідності до вибраного рядка меню)
 CASE vibor=1
 APPEND
 CASE vibor=2
 BROWSE
 CASE vibor=3
 SORT TO t1 ON Posada, oklad FIELDS Prizv, Posada, Oklad
 SELECT 2
 USE "c:\Program Files\visual foxpro projects\t1" exclusive
 BROWSE
 CASE vibor=4
 ACCEPT "Уведіть прізвище співробітника" TO pr
 LOCATE FOR Prizv=pr
 DISPLAY
 CASE vibor=5
 CLOSE all (Закрити всі файли)
 RETURN (Повернутися з програми у діалоговий режим)
 ENDCASE (Кінець розгалуження)
 ENDDO (Кінець циклу)

Режими обробки бази даних:

- 1 – відкриття таблиці;
- 2 – редагування даних;
- 3 – пошук даних за критерієм;
- 4 – додати новий запис;
- 5 – упорядкувати таблицю за певними полями та переглянути результат упорядкування;
- 6 – проіндексувати таблицю за певними полями та виконати швидкий пошук даних;
- 7 – виконати розрахунки за певними полями таблиці шляхом заміни;
- 8 – обчислити підсумки та середні значення за числовими полями;
- 9 – вибрати дані, що задовольняють умові та переглянути результат вибірки;
- 10 – закрити таблицю, вийти з програми.

Варіанти завдань

№ варіанта	Режими обробки (дивись вище)				
1	1	2	3	7	10
2	1	4	5	8	10
3	1	2	6	9	10
4	1	4	3	8	10
5	1	2	6	8	10
6	1	4	5	6	10
7	1	2	7	9	10
8	1	4	3	5	10
9	1	2	5	7	10
10	1	4	3	9	10
11	1	2	6	7	10
12	1	4	5	9	10

Контрольні питання:

1. Для чого призначене меню команд, що створюється у командних файлах?
2. Які є способи створення меню команд у Visual FoxPro?
3. Який оператор управління реалізує вибір послідовності дій у командному файлі?
4. Які оператори призначені для організації циклічних дій?
5. Яку дію виконує оператор CLEAR?
6. Як увести числові дані з клавіатури?
7. Як увести символні дані з клавіатури?
8. Як відкрити таблицю даних для повного доступу?

Лабораторна робота № 14

Програмування з використанням підпрограм

Завдання:

Створити командний файл для виконання завдання за варіантом з використанням операторів управління та підпрограм:

Варіант 1

Створити таблицю “Квартира” нарахування сплати за квартиру, яка містить інформацію: № квартири, її площа, кількість мешканців. Увести до таблиці 7 записів.

Відомо, що 1 м^2 площини у межах норми коштує X коп., норма площини на одну людину встановлює $S\text{ м}^2$, сім’ї, де норма проживання не витримана, сплачують на $K\%$ менше від нарахованої суми.

Підрахувати, скільки кожна сім’я повинна сплачувати за площину, визначити загальну плату.

Варіант 2

Створити таблицю “Електроенергія” нарахування сплати за спожиту електроенергію, яка містить інформацію: прізвище споживача, категорія пільг, попереднє показання лічильника, теперішнє показання лічильника. Увести до таблиці 7 записів

Відомо, що існує дві категорії споживачів. Для першої категорії нарахування за спожиту електроенергію здійснюється за звичайним тарифом, тобто, по X коп. за одиницю, для другої категорії є льготний тариф – на Y коп. менше тарифного.

Підрахувати кількість спожитої електроенергії, розмір платні, визначити загальну плату.

Варіант 3

Створити таблицю “Вода” нарахування сплати за спожиту воду, яка містить прізвище споживача, кількість спожитої гарячої води (м^3), кількість спожитої холодної води. Увести до таблиці 7 записів

Відомо, що 1 м^3 холодної води коштує M_1 коп., 1 м^3 гарячої води коштує M_2 коп.

Підрахувати кількість спожитої води, розмір платні за гарячу воду, за холодну воду, загальну плату за спожиту воду кожним споживачем, та загальну кількість спожитої води.

Варіант 4

Створити таблицю “Опалення” нарахування сплати за спожите опалювання, яка містить інформацію: прізвище споживача, площа квартири, категорія пільг. Увести до таблиці 7 записів

Відомо, що для споживачів 1 категорії пільг сплата встановлена у розмірі N_1 коп. за 1 м^2 , для споживачів 2 категорії – N_2 коп. за 1 м^2 .

Підрахувати сплату за опалення кожним споживачем, визначити загальну та середню сплату.

Варіант 5

Створити таблицю “Газ” нарахування сплати за спожитий газ, яка містить інформацію: № квартири, кількість мешканців, кількість спожитого газу. Увести до таблиці 7 записів

Відомо, що норма споживання газу на одну людину встановлює $X\text{ м}^3$ і сплата за 1 м^3 – Y

коп. Якщо спожитий газ перевищує норму, то сплата збільшується на U_1 коп. за кожний m^3 газу, спожитого поза норму.

Підрахувати сплату за опалення кожним споживачем та загальну. Вказати мінімальну та максимальну сплату.

Варіант 6

Створити таблицю “Заробітна плата” нарахування заробітної платні робітникам, яка містить інформацію: прізвище робітника, оклад, кількість відпрацьованих днів. Увести до таблиці 7 записів

Підрахувати розмір премії, яка становить $M\%$ від окладу, якщо робітник відпрацював місячну кількість днів N , а в інших випадках розмір премії нараховується за формулою: $[\text{оклад} / N] * (\text{кількість відпрацьованих днів}) * M$.

Підрахувати загальну суму нарахувань, найбільшу та середню суму нарахувань.

Варіант 7

Створити таблицю “Електрика” нарахування сплати за спожиту електроенергію, яка містить інформацію: прізвище, категорія пільг, попереднє показання лічильника, теперішнє показання лічильника. Увести до таблиці 7 записів

Відомо, що споживачам першої категорії пільг сплата встановлена N коп. за одиницю спожитої електроенергії, для другої категорії пільг: за нормою споживання M кВт сплата становить 50% тарифа, по-за нормою – за тарифом.

Підрахувати кількість спожитої електроенергії, суму до сплати та загальну суму. Визначити мінімального платника.

Варіант 8

Створити таблицю “Заробітна плата” нарахування заробітної платні робітникам, яка містить інформацію: прізвище робітника, розряд, кількість виготовлених деталей. Увести до таблиці 7 записів

Відомо, що робітники 1 розряду за кожну виготовлену деталь отримують N_1 коп., 2 розряду – N_2 коп., 3 розряду – N_3 коп.

Вивести загальну суму нарахувань, загальну кількість виготовлених деталей та середню заробітну платню.

Варіант 9

Створити таблицю “Допомога” нарахувань матеріальної допомоги робітникам, яка містить такі відомості: прізвище робітника, заробітну платню, кількість дітей. Увести до таблиці 7 записів

Відомо, що мінімальний прожитковий мінімум становить X крб. Якщо заробітна платня робітника менше цього прожиткового мінімуму, то матеріальна допомога на кожну дитину становить U крб.

Підрахувати розмір допомоги кожному робітнику, загальну суму допомоги, найменшу та найбільшу суми допомоги.

Варіант 10

Створити таблицю “Податок” нарахування податків за бездітність, яка містить відомості: прізвище податкоплатника, кількість дітей, прибуток. Увести до таблиці 7 записів

Відомо, що податківець, у якого немає дітей, сплачує податки в розмірі $N_1\%$ від прибутку, у якого не більше 2-х дітей – $N_2\%$ від доходу, сім'я, в якій 3 і більше дитини, звільняється від цього податку.

Нарахувати податок, загальну суму податку, найбільший прибуток та його податок.

Варіант 11

Створити таблицю “Квартира” нарахування сплати за квартиру, яка містить інформацію: прізвище квартиронаймача, площа, категорія пільг. Увести до таблиці 7 записів

Відомо, що 1 категорія сплачує за $1m^2$ – N коп., 2 категорія – на $K\%$ нижче, а 3 категорія має 100% пільги.

Підрахувати сплату кожної сім'ї, визначити загальну плату.

Варіант 12

Створити таблицю "Телефон" нарахування сплати за телефонні розмови, яка містить відомості: номер телефону, кількість хвилин міських розмов, міжміських та міжнародних розмов окремо. Увести до таблиці 7 записів

Відомо, що кожна хвилина міської розмови становить N_1 коп., міжміської – N_2 коп., міжнародної – N_3 коп.

Обчислити суму сплати окремо за кожен тип розмови та сумарну платню для кожного абонента. Знайти загальну суму платні, середню суму 1 хвилини.

Приклад:

Створити таблицю "Нова ціна" розрахунку нової ціни на продукти, яка містить відомості: назва продукту, ціна за одиницю продукту, належність до категорії. Увести до таблиці 7 записів

Відомо, що продовольчі товари мають три категорії. За один день зберігання продукти 1 категорії мають знижку на $N_1\%$ від початкової ціни, 2 категорії – $N_2\%$, а третьої – $N_3\%$.

Підрахувати нову ціну продуктів на другий та третій день після виготовлення. Знайти загальну суму знижки на всі продукти.

Програма має вигляд:

Основна програма

CLOSE ALL

USE "c:\documents and settings\k\мои документы\visual foxpro projects\nova_cina.dbf" excl

DO while.t.

v=0

@ 5,5 prompt "Створити таблицю"

@ 7,5 prompt "Уведення даних"

@ 9,5 prompt "Редагування даних"

@ 11,5 prompt "Розрахунок знижки"

@ 13,5 prompt "Розрахунок ціни по днях"

@ 15,5 prompt "Вихід"

menu to v

DO case

CASE v=1

CREATE

CASE v=2

APPEND

CASE v=3

BROWSE

CASE v=4

DO "c:\documents and settings\k\мои документы\visual foxpro projects\rozrahunok"

(Запускаємо на виконання процедуру)

CASE v=5

DO "c:\documents and settings\k\мои документы\visual foxpro projects\znizhka"

(Запускаємо на виконання процедуру)

CASE v=6

CLOSE ALL

RETURN

ENDCASE

ENDDO

Підпрограма rozrahunok

GO 1

DO while.not.eof().

```
IF (kat=1)
nc=cina*0.15
ENDIF
IF (kat=2)
nc=cina*0.1
ENDIF
IF (kat=3)
nc=cina*0.05
ENDIF
replace n with nc
SKIP
ENDDO
BROWSE
RETURN
```

Підпрограма znizhka

```
GO 1
DO while.not.eof().
replace day1 with cina
replace day2 with cina-n
replace day3 with cina-2*n
SKIP
ENDDO
BROWSE
RETURN
```

Контрольні питання:

1. Для чого призначені підпрограми?
2. Як створити підпрограму?
3. Як ініціювати виконання підпрограми?
4. Як описати цикл за всіма записами таблиці до кінця файла?
5. Як організувати розгалужені розрахунки у програмі?
6. Записати синтаксис оператора розгалуження.
7. Записати синтаксис оператора вибору.
8. Записати синтаксис циклічного оператора за передумовою.
9. Де записується текст підпрограми?
10. Як підключити підпрограму до основної програми?

Рекомендована література

1. Базы данных: Учеб. для вузов / Под ред. А. Д. Хомоненко. СПб.: КОРОНА принт, 2000. 416 с.
2. Дейт К. Дж. Введение в системы баз данных: Пер. с англ. 7-е изд. М.; СПб.; Киев: Вильямс, 2001. 1071 с.
3. Кошелев, В.Е. Access 2003. Практическое руководство / В.Е. Кошелев. – М. : ООО «Бином-Пресс», 2005. – 464 с.
4. Бекаревич, Ю.Б. Создание реляционной базы данных и запросов. MS Access 2007: учебное пособие / Ю.Б. Бекаревич, Н.В. Пушкина. - Санкт-Петербург: СПбГУЭФ, 2012. - 87 с.
5. Бекаревич Ю.Б., Пушкина Н.В. Самоучитель Microsoft Access 2009. – СПб.: БХВ – Петербург, 2009. – 720 с.
6. Карпова Т.С. Базы данных: модели, разработка, реализация. – СПб.: Питер, 2001. – 304 с.
7. Харрингтон Д.Л. Проектирование баз данных. Просто и доступно. – М.: Лори, 2000. – 230 с.
8. Гринченко Н. Н., Гусев Е. В., Макаров Н. П. Проектирование баз данных. СУБД Microsoft Access, 2004. – 364 с.
9. Михеева В., Харитоновна И. Microsoft Access 2003. Издательство: БХВ-Петербург, 2004. – 1072 с.
10. Федунец А.Д. Пакети програм офісного призначення. – Кіровоград: Народне слово, 2000. – 272 с.
11. Лебедев А.Н. Самоучитель Visual FoxPro 9.0. Издательство: НТ-Пресс, 2005. - 329 с.
12. Бартенев О. В. Microsoft Visual FoxPro. Учебно-справочное пособие. Издательство: Диалог-МИФИ, 2005. - 427 с.
13. Омельченко Л. Самоучитель Visual FoxPro 8. Издательство: БХВ-Петербург, 2003. - 389 с.
14. Шапоров Д. Visual FoxPro. Уроки программирования. Издательство: БХВ-Петербург, 2005. - 276 с.
15. Мусина Т. В. Visual FoxPro 8.0. Учебный курс. Издательство: Век +, Корона-Принт, НТИ, 2004. - 359 с.
16. Грег Риккарди. Системы баз данных. Теория и практика использования в Internet и среде Java. Издательство: Вильямс, 2001. - 297 с.
17. Кириллов В. В., Громов Г. Ю. Введение в реляционные базы данных. Издательство: БХВ-Петербург, 2009. - 226 с.
18. Глушаков С. В., Ломотько Д. В. Базы данных. Издательство: АСТ, 2002. - 386 с.
19. Рудикова Л. В. Базы данных. Разработка приложений. Издательство: БХВ-Петербург, 2006. - 394 с.
20. Г. Гарсиа-Молина, Д. Ульман, Д. Уидом. Издательство Системы баз данных. Полный курс.: Вильямс, 2003. - 723 с.
21. Гайдамакин Н. А. Автоматизированные информационные системы, базы и банки данных. Вводный курс. Издательство: Гелиос АРВ, 2002. - 456 с.
22. Базиян Менахем и др. Использование Visual FoxPro 6. М.: Вильямс, 2003. — 928 с.
23. Гайна Г.А. Основи проектування баз даних. Навч. посібник. К.: Кондор, 2007. - 208 с.
24. Ситник Н. В. Проектування баз і сховищ даних: Навч. посібник – К.: КНЕУ, 2004. – 348 с.