

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

**«ОСНОВИ УПРАВЛІННЯ БАЗАМИ ДАНИХ У
MICROSOFT ACCESS»**

**Методичні вказівки і завдання до лабораторних робіт
для студентів економічних спеціальностей**

*Розглянуто на засіданні кафедри
обчислювальної математики
і програмування.
Протокол №7 від 1 березня 2010 р.*

*Затверджено на навчально-видавничій
раді ДонНТУ.
Протокол №2 від 2 квітня 2010 р.*

Донецьк ДонНТУ 2010

УДК 681.3.06(071)

О 23

Основи управління базами даних у Microsoft Access / Методичні вказівки і завдання до лабораторних робіт для студентів економічних спеціальностей // Укл.: Зензеров В.І., Славінська Л.В., Перінська О.В. – Донецьк: ДонНТУ, 2010. – 80 с.

Методичні вказівки присвячені основним поняттям, що стосуються баз даних і систем керування базами даних. Зроблено короткий огляд можливостей додатку пакету Microsoft Office Access як системи керування реляційною базою даних, опис її робочого середовища й операцій, пов'язаних з створенням бази даних, відкриттям існуючій та управлінням нею. Теоретичний матеріал проілюстровано багатьом прикладів.

Містять приклад і варіанти завдань для виконання лабораторної роботи, що сприяє практичному засвоєнню матеріалу.

Матеріал корисний для студентів економічних спеціальностей, що вивчають курс "Інформатика і КТ", а також для викладачів, аспірантів і інших користувачів персональних комп'ютерів.

Автори:

Зензеров В.І., доцент

Славінська Л.В., ст. викладач

Перінська О.В., асистент

Відповідальний за випуск:

д.т.н., проф. Павлиш В.М.

ПЕРЕДМОВА

Система керування базами даних Microsoft Access є одним із самих популярних додатків у сімействі настільних СУБД. Всі версії Access мають у своєму арсеналі засобу, що значно спрощують уведення й обробку даних, пошук даних і надання інформації у вигляді таблиць, графіків і звітів. Починаючи з версії Access 2000, з'явилися також Web-сторінки доступу до даних, які користувач може переглядати за допомогою програми Internet Explorer. Крім цього, Access дозволяє використати електронні таблиці й таблиці з інших настільних і серверних баз даних для зберігання інформації, необхідної додатку. Приєднавши зовнішні таблиці, користувач Access буде працювати з базами даних у цих таблицях так, ніби це були таблиці Access. При цьому й інших користувачах можуть продовжувати працювати із цими даними в тім середовищі, у якій вони були створені.

Методичні вказівки адресовані тим користувачам, які тільки приступають до використання Access у своїй роботі. Тому ми даємо в ній основні поняття, що стосуються баз даних і систем керування базами даних. Ще ми хотіли б показати різницю між обробкою даних у форматі електронних таблиць, які стали вже звичними для широкого кола користувачів, і можливостями систем керування базами даних. Ми пояснимо також, коли треба від простого способу роботи з електронними таблицями перейти до більше складного додатка Access. Надалі ви переконаєтеся, що насправді не таке воно складне, якщо вміти користуватися тими помічниками, які представляє Access. Ми дамо короткий огляд можливостей Access як системи керування реляційною базою даних, опис її робітничого середовища, довідкової системи й операцій, пов'язаних з відкриттям уже існуючих додатків Access і створенням нового файлу бази даних.

Access - це система керування базами даних (СУБД), під якою розуміється комплекс програм, що дозволяє зберігати більші масиви даних у певному форматі, обробляти їх і представляти в зручному для користувача виді. Access дає можливість автоматизувати часто виконувані операції (наприклад, розрахунок заробітної плати, облік матеріальних цінностей і т.п.). З його допомогою можна розробляти зручні форми уведення й перегляду даних, а також становити складні звіти.

Access - реляційна база даних (можлива одночасна робота з декількома зв'язаними таблицями бази даних), у якій передбачене багато сервісних функцій. В Access убудовані майстри таблиць, форм, звітів, які полегшують їхнє створення з наявних заготівель. Макроси дозволяють автоматизувати багато процесів без програмування, тоді як убудована мова VBA (Visual Basic for Applications - діалект мови Basic для використання в додатках Microsoft Office) дає можливість програмувати складні процедури обробки даних. Microsoft Access містить безліч засобів для роботи в Internet, а також набір інструментів для керування базами даних, що включає в себе конструктори таблиць, форм, запитів і звітів. Потужність і гнучкість СУБД Access роблять її сьогодні однією із кращих програм для керування базами даних.


1. ТАБЛИЦІ

1.1. Створення таблиць

Всі складові бази даних (таблиці, звіти, запити, форми й об'єкти) зберігаються в єдиному файлі. Основним структурним компонентом бази даних є таблиця, у якій зберігаються вводять дані. Таблиця складається зі стовпців, названих *полями*, і рядків, названих *записами*. Кожний запис таблиці містить інформацію про окремий елемент бази даних.

Для створення бази даних відкрийте меню «Пуск» / «Програми»/ «Microsoft Access».

У вікні, що відкрилося, треба:

1. Нажати кнопку «Создать»  на панелі інструментів.
2. Вибрати «Новая база данных» (рис. 1.1).

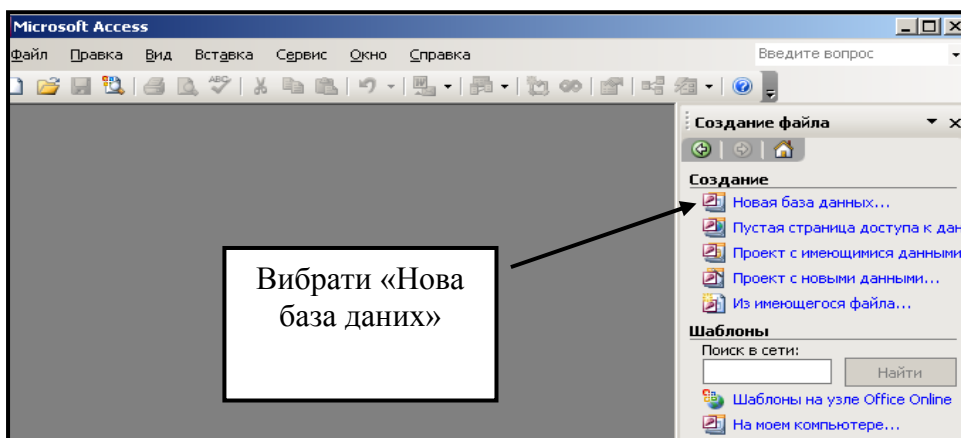


Рис. 1.1. Вікно створення нової бази даних

3. У діалоговому вікні «Файл новой базы данных» увести ім'я бази даних, указати її розташування, нажати кнопку «Создать» (рис. 1.2).

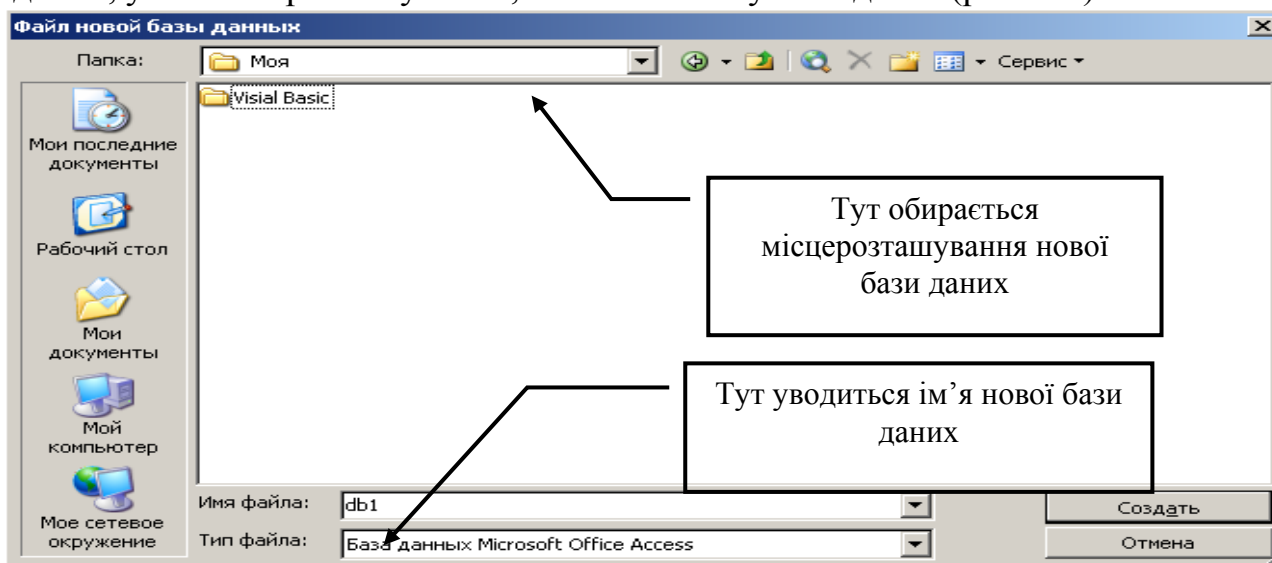


Рис. 1.2. Діалогове вікно присвоєння імені й вказівки місця розташування нової бази даних

Подивіться на корінці, представлені у вікні «База даних» (рис. 1.3), це основні об'єкти: таблиці, запити, форми, звіти, сторінки, макроси й модулі.

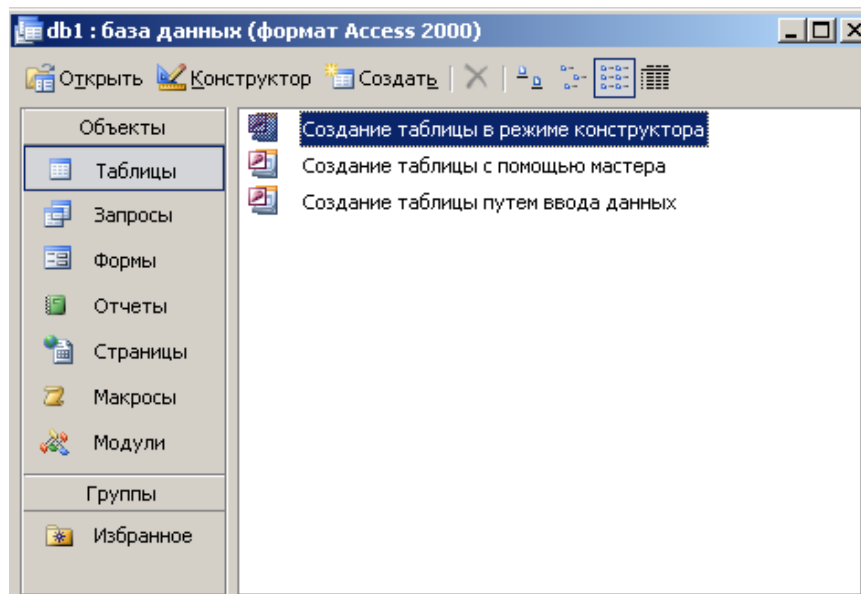


Рис. 1.3. Вікно знову створеної бази даних

- ✓ *Таблиця* - використовується для завдання структури бази й зберігання даних.
- ✓ За допомогою *запиту* проводять пошук інформації із критеріїв.
- ✓ *Форма* - об'єкт, призначений для уведення даних, розрахунків і прояву талантів дизайнера.
- ✓ *Звіт* - створення вихідної форми для перегляду й роздрукування.
- ✓ *Макрос* - використання макросів дозволяє автоматизувати повторювані операції.
- ✓ *Модулі* - містять VBA-код, використовуваний для написання процедур обробки подій, таких, наприклад, як натискання кнопки у формі або звіті, автоматизації дій користувачів.

Для створення таблиці можна скористатися декількома способами.

1. Створення таблиці в режимі Конструктора.
2. Створення таблиці за допомогою Майстра. Майстер баз даних створює нову базу, виходячи з Ваших відповідей на питання.
3. Створення таблиці шляхом уведення даних безпосередньо в порожню таблицю в режимі таблиці, щось нагадує уведення даних в Excel.

1.2. Створення нової бази даних

Для створення бази даних варто визначити структуру таблиці: кількість і назви полів, з яких вона повинна складатися, типи полів і їхні розміри. Вікно Конструктора (рис. 1.4) складається із двох частин. У верхній частині екрана задають структуру таблиці (назви полів, їхній тип і коментарі, якщо в них є необхідність), нижня частина екрана приділяється під властивості полів.

Кожному полю таблиці привласнюється ім'я, що не може містити більше 64 символів. У нижній частині екрана визначають тип поля й окремої його характеристики. Набір можливих характеристик визначається типом поля.

Типи полів:

- ✓ текстовий - для зберігання текстової інформації з максирисьною довжиною до 255 символів;
- ✓ числовий - для зберігання числових даних, використовуваних для проведення розрахунків;
- ✓ дата/час - для зберігання інформації про дату й час;
- ✓ грошовий - числові дані зображувані спільно зі знаком грошової одиниці;
- ✓ поле МЕМО - зберігання коментарів (до 65 535 символів);
- ✓ лічильник - спеціальне числове поле, у якому автоматично привласнюється порядковий номер кожного запису (значення поля обновити не можна);
- ✓ логічний - може мати одне із двох можливих значень (True/False);
- ✓ поле об'єкта OLE - об'єкт, зв'язаний або впроваджений у таблицю Access, призначений для зберігання зображень;
- ✓ гіперпосилання - рядок, що складається з букв і цифр і зображує собою адресі web-сторінок.

Властивість «Размер» визначається типом поля. Для текстових полів варто вказати число символів самого довгого тексту в поле. Для числових можливі різні варіанти (рис. 1.5), пов'язані з максирисьною величиною значення в поле. Байт - цілі числа від 0 до 255; ціле - цілі числа від -32768 до 32767; довге ціле - від $-2 \cdot 10^9$ до $2 \cdot 10^9$; із плаваючою крапкою – 4 байти - від $-3,4 \cdot 10^{38}$ до $3,4 \cdot 10^{38}$; із плаваючою крапкою – 8 байтів - від $-1,8 \cdot 10^{308}$ до $1,8 \cdot 10^{308}$.

Властивість «Формат поля» задає подання даних на екрані або печатці. Прийнято формати виду:

- ✓ основний;
- ✓ грошовий з використанням символу валюти;
- ✓ фіксований - два знаки після коми;
- ✓ с роздільниками розрядів - кома як роздільник тисяч;
- ✓ процентний;
- ✓ експонентний - подання чисел з порядком, причому E заміняє порядок, наприклад, 1,7E09 рівносильний запису 1,7(109).

Найбільш істотні властивості полів:

Для приклада на рис. 1.5 показане визначення формату поля типу «Дата».

Приклад.

Створити базу даних по наступних полях:

- Назва журналів
- Тиражі
- Продаж екземплярів за місяць

- Дата перевірки
- Продаж усього екземплярів
- Дата перевірки тиражу
- Ціна
- Тип

Приклад створення розглянутий на рис. 1.4, 1.5, 1.6

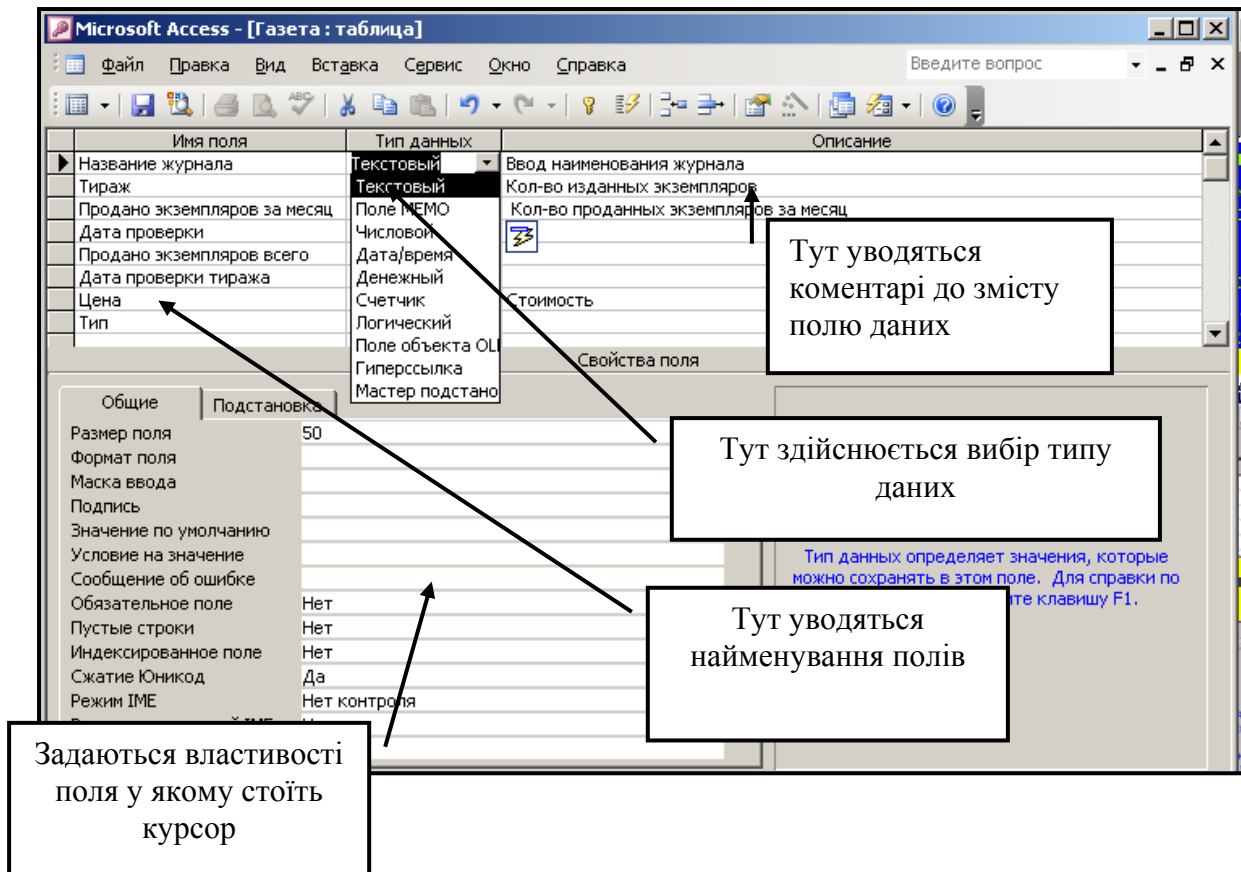


Рис. 1.4. Вікно Конструктора бази даних

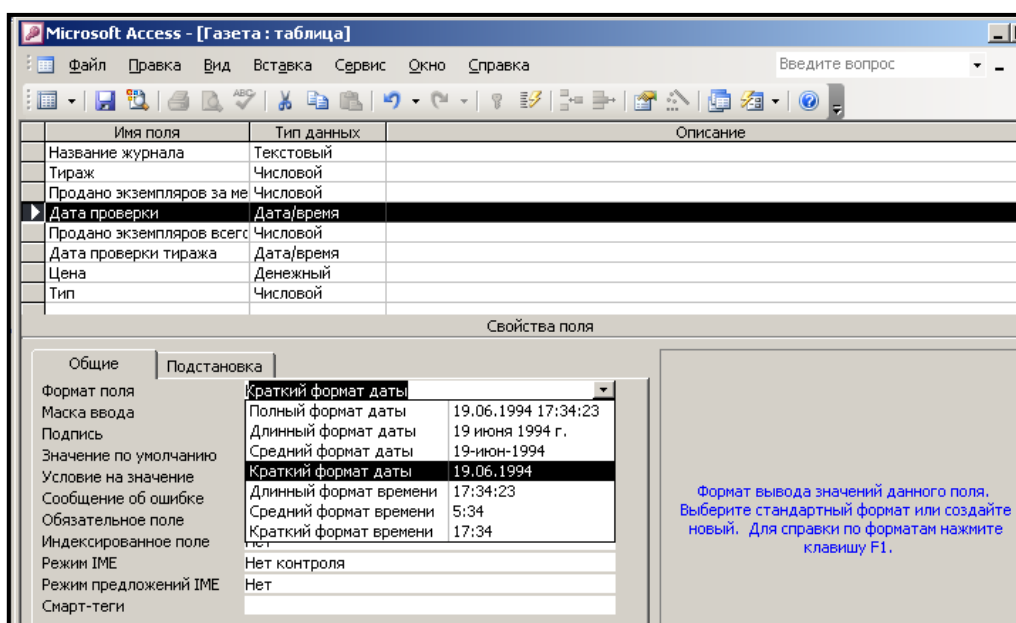


Рис. 1.5. Вибір розміру поля для даних типу «Дата»

Варто пам'ятати, що, вибравши будь-який формат, надалі дані необхідно вводити тільки в ньому. При спробі уведення даних в іншому форматі з'явиться повідомлення про помилку.

«Число десятичных знаков» задає точність подаваної інформації.

«Маска ввода» – використовується для перевірки відповідності шаблону даних, що вводять.

«Подпись» – використовується для зберігання повного імені поля, якщо при створенні таблиці ім'я поля було скорочено. Повний текст імені поля буде з'являтися у формі.

«Значения по умолчанию» – автоматично встановлює значення для кожного нового запису.

«Условие на значение» – умова, якій повинні задовольняти введенні значення.

«Сообщение об ошибке» – текст, що з'явиться, якщо умова, уведена в попередньому пункті, не буде виконано.

«Обязательное поле» – уведення даних у полі не є обов'язковим, якщо в графі зазначене «ні».

«Индексированное» - визначає індекс по полю – засіб, що прискорює пошук і сортування в таблиці. Індекс - внутрішня таблиця, що складається із двох стовпців: значення поля, по якому виробляється індексування, і місце розташування кожного запису таблиці.

Після опису всіх полів необхідно перейти в режим таблиці. Клацнувши по відповідній кнопці (рис. 1.6), вибираємо «Режим таблицы», при збереженні структури таблиці (дані ще не вводилися, визначалася кількість полів і їхньої властивості) уводять її назву (ім'я).

У режимі *Конструктора* можна створити нову таблицю або додати, видалити або настроїти поля існуючої таблиці.

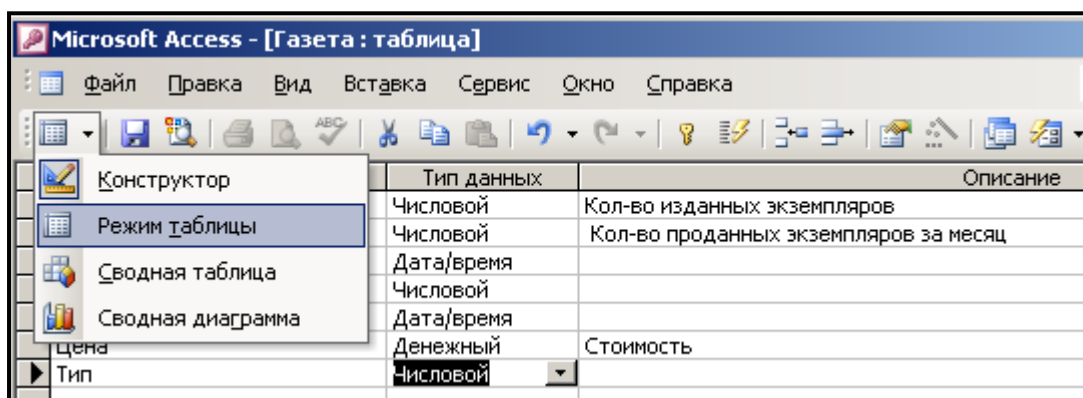


Рис. 1.6. Перехід з режиму *Конструктора* в режим таблиці

Уведення даних в осередки таблиці аналогічний звичайному набору. Для деяких типів даних (числова, грошовий, дата/час, логічний) Access автоматично перевіряє правильність уведення.

1.3. Перегляд і редагування таблиць

Для перегляду таблиці необхідно виконати наступні дії:

1. Відкрити базу даних.
2. Вибрати таблицю й клацнути по кнопці «Відкрити».

Базу даних можна відкрити в режимі конструктор для зміни структури таблиці або в режимі таблиці - для зміни даних, що зберігаються в ній.

Спочатку таблиці виводяться на екран зі стандартною висотою рядків і стовпців. Якщо ширина недостатня для одних полів і надлишкова для інших, її можна змінювати. Для цього виділити один або кілька стовпців і виконати команду «Формат»/ «Ширина стовпця». У вікні, що з'явилося (рис. 1.7) увести ширину стовпця або задати опцію «По ширині даних», що встановлює ширину стовпця по довжині самого довгого значення в стовпці. «Стандартна ширина» встановлюється автоматично.

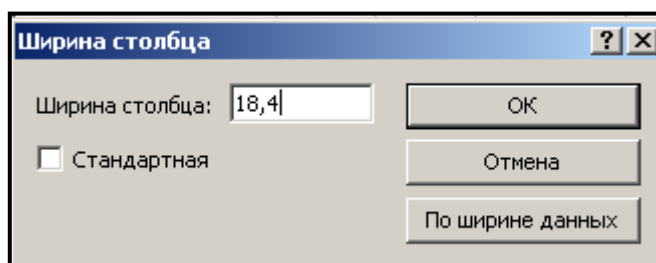


Рис. 1.7 - Вікно зміни ширини стовпця

Змінювати ширину стовпця можна вручну. Для цього варто вибрати потрібний стовпець, підвести курсор до його правої границі (при цьому форма курсору прийме вид перехрестя $\leftarrow \text{+} \rightarrow$), нажати ліву кнопку миші й, не відпускаючи її, плавно переміщати миша в потрібному напрямку. Впливаючи за мишею, буде переміщатися й вертикальна лінія - границя стовпця. По досягненні необхідної ширини кнопку миші відпускають.

Зміна висоти рядка відбувається аналогічно після вибору меню «Формат»/ «Висота строки».

Порядок проходження стовпців збігається з послідовністю визначення полів у таблиці. Для його зміни необхідно виділити стовпець, клацнувши по його заголовку, і, не відпускаючи лівої кнопки миші, перемістити курсор на нове місце.

Для того, щоб при прокручуванні аркуша вправо лівий крайній стовпець відображался на екрані необхідно виконати його закріплення. Для цього:

1. Виділити потрібний стовпець, клацнувши по імені його поля;
2. Виконати команду «Формат»/ «Закрепить столбцы».

При відкритті таблиці на екрані відображаються всі стовпці таблиці. Деякі з них при необхідності можна сховати. Для цього варто виконати команду «Формат»/ «Сховати стовпці», попередньо виділивши потрібний стовпець. Для відображення схованих стовпців виконується команда

«Формат»/ «Відобразити стовпці». У вікні, що відкрилося (рис. 1.8) відзначити прапорцем ті стовпці, які необхідно вивести на екран.

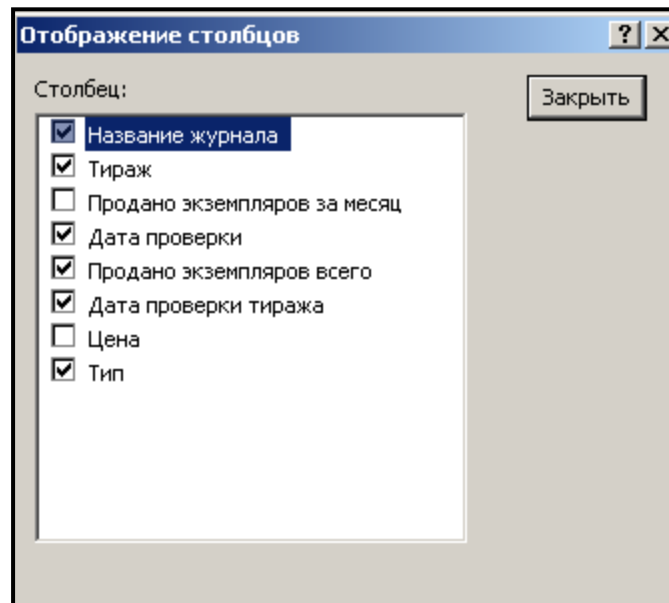


Рис. 1.8 - Вибір виведених на екран стовпців

Завжди при роботі з базою даних варто враховувати одну *нерисоважну особливість* - редагування в **Access** відрізняється від аналогічних операцій в **Word** і **Excel**.

Вся база даних - це один великий документ, що зберігається на диску у вигляді **mdb** файлу. Таблиці - самостійні документи, що входять до складу БД і зберігаються в її тілі. Структура кожної таблиці - це документ і він імплантований у загальний файл. Запити, форми, звіти - все це документи, замасковані в тілі mdb-файлу. Тому все, що ми робимо з кожним із цих елементів, змінює всю БД.

Всі зміни в таблицях зберігаються автоматично, у *режимі реального часу*. Це означає, що поки ми працюємо з таблицею, відбувається її безперервне збереження на диску. Як тільки кінчене уведення даних в один рядок і натиснута клавіша **Enter** для переходу до наступної, дані негайно додаються в базу й записуються на жорсткий диск. Особливо добре це видно при збереженні бази на дискету, кожний запис супроводжується записуванням і це видно візуально.

Цей принцип забезпечує надійність і безпека бази даних.

2. ЗВ'ЯЗУВАННЯ ТАБЛИЦЬ НА СХЕМІ ДАНИХ

Для того щоб було зручно переглядати, створювати, видаляти й модифікувати зв'язку між таблицями, в Microsoft Access використовується схема даних (Relationships) (рис. 2.1).

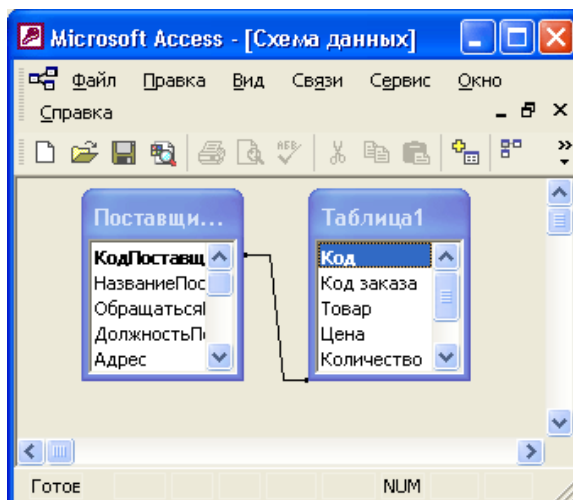


Рис. 2.1. Приклад схеми даних

Щоб відкрити схему даних, необхідно виконати команду Сервіс, Схема даних (Tools, Relationships). За замовчуванням схема буде містити всі таблиці зі зв'язками. Тому сюди буде включена, наприклад, наша нова таблиця "Список розсилання оповіщень", що була пов'язана з таблицею "Клієнти" (Customers) за допомогою Майстра підстановок (див. розд. "Використання Майстра підстановок" вище в даній главі).

Щоб забрати яку-небудь таблицю зі схеми даних, необхідно клацнути правою кнопкою миші на будь-якій місці цієї таблиці й з контекстного меню вибрати команду Сховати таблицю (Hide table). Наприклад, на рис. 2.1 таблиця "Список розсилання оповіщень" схована.

Видалення таблиці зі схеми даних не означає видалення її з бази даних, просто вона не присутня в схемі.

Щоб знову додати в цю схему згорнуту таблицю ("Список розсилання оповіщень") або таблицю, у якого зв'язки ще не встановлені, необхідно:

1. Клацнути правою кнопкою миші на вільному просторі схеми даних і з контекстного меню вибрати команду Додати таблицю (Show table).
2. У діалоговому вікні Додавання таблиці (Show table) розкрити вкладку Таблиці (Tables), вибрати зі списку таблицю "Список розсилання оповіщень", і нажати кнопку Додати (Add) (рис. 2.2).

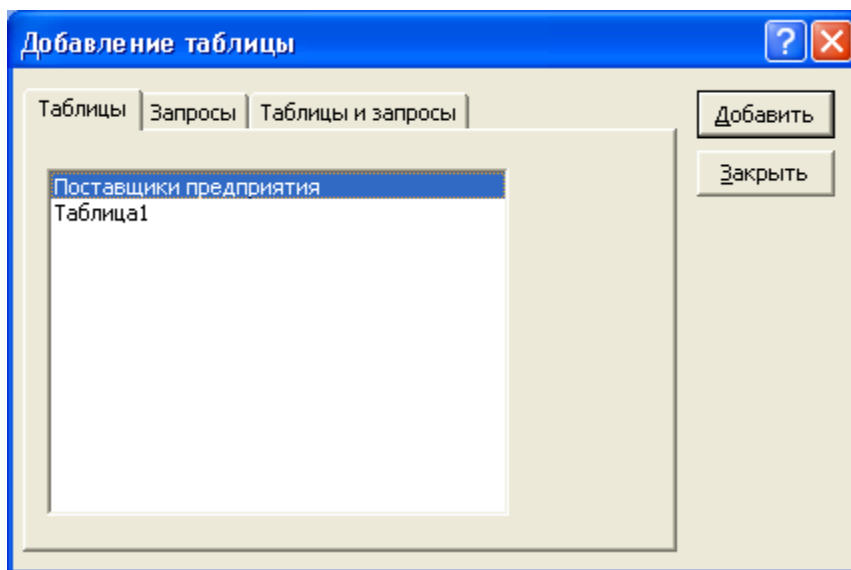
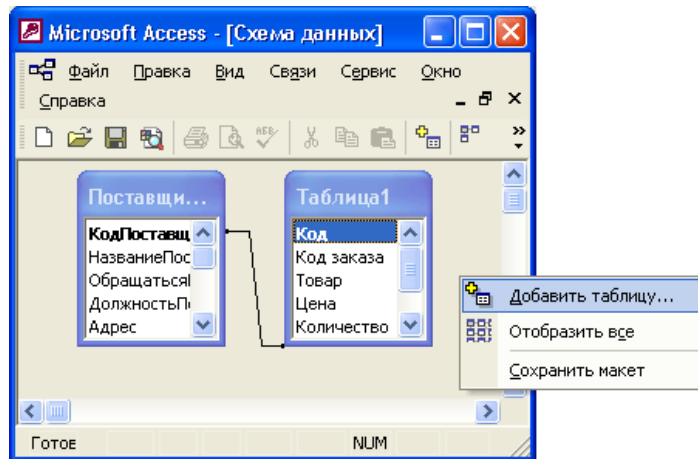


Рис. 2.2. Диалогове вікно Додавання таблиці

Якщо потрібно додати до схеми кілька таблиць, прийде повторити цю операцію для кожної таблиці. Нажати кнопку Закрити (Close), щоб закрити діалогове вікно Додавання таблиці (Show table).

У цьому випадку зв'язок між таблицями Access будує автоматично. Це відбувається тоді, коли дві таблиці мають однакові назви зв'язаних полів і погоджені типи даних, причому хоча б в одній з таблиць зв'язане поле є ключовим.

Погоджені типи даних означають наступне: якщо ключове поле має тип даних Лічильник (AutoNumber), те відповідне йому поле в іншій таблиці повинне мати тип Довге ціле (Long Integer). В інших випадках типи даних повинні просто збігатися.

Якщо потрібний зв'язок автоматично не створений, її можна створити двома шляхами. Перший шлях - графічний. Варто вибрати в головній таблиці поле для зв'язку, нажати ліву кнопку миші й перетягнути поле в другу таблицю. Відпустити ліву кнопку миші над тим полем підлеглої таблиці, з

яким установлюється зв'язок. Після цього з'явиться діалогове вікно Зміна зв'язків (Edit Relationships) (рис. 2.3).

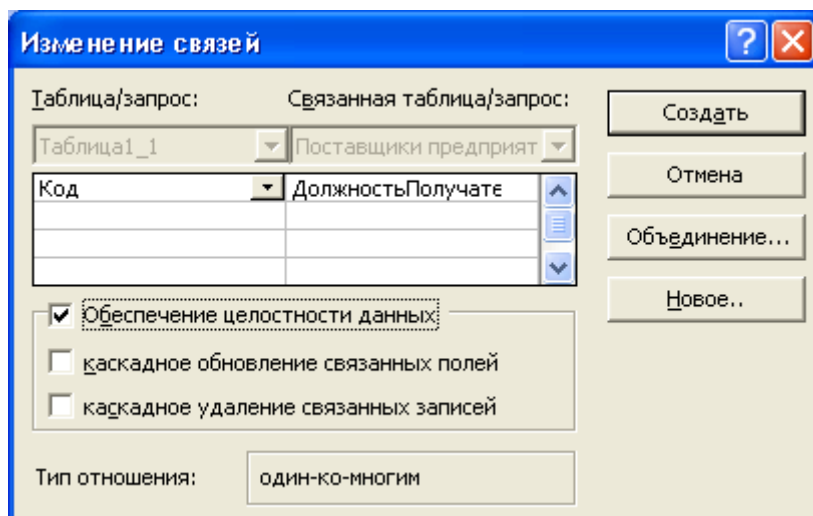


Рис. 2.3. Вікно Зміна зв'язків

Альтернативний варіант - вибрати команду Зв'язку, Змінити зв'язок (Relationships, Edit Relationships) з головного меню Access. Ця команда з'являється в меню, коли відкрите вікно Схема даних (Relationships).

У діалоговому вікні Зміна зв'язків (Edit Relationships) можна вибрати зі списків назви зв'язаних таблиць і полів для зв'язування. Якщо відношення між таблицями "один-до-багатьох", то ліворуч зі списку Таблиця/запит (Table / Query) вибирається головна таблиця й поле в цій таблиці, а праворуч зі списку Зв'язана таблиця/запит - підлегла їй відповідно поле в ній. Якщо відношення "один-до-одному", то порядок таблиць значення не має. Якщо ви встановлювали зв'язок графічно, то всі поля в списках уже обрані, і потрібно тільки визначити правила посилальної цілісності.

Для цього встановлюють прапорець Забезпечення цілісності даних (Enforce Referential Integrity) і один або обидва прапорці: каскадне відновлення зв'язаних полів (Cascade Update Related Fields) і каскадне видалення зв'язаних записів (Cascade Delete Related Records) (див. наступний розділ).

При необхідності можна змінити параметри об'єднання, для чого потрібно натиснути кнопку Об'єднання (Join Type) і вибрати один з перемикачів:

- Об'єднання тільки тих записів, у яких зв'язані поля обох таблиць збігаються;
- Об'єднання ВСІХ записів з <ім'я першої таблиці> і тільки тих записів з <ім'я другої таблиці>, у яких зв'язані поля збігаються;
- Об'єднання ВСІХ записів з <ім'я другої таблиці> і тільки тих записів з <ім'я першої таблиці>, у яких зв'язані поля збігаються.

За замовчуванням установлюється перший перемикач. Натиснути кнопку ОК.

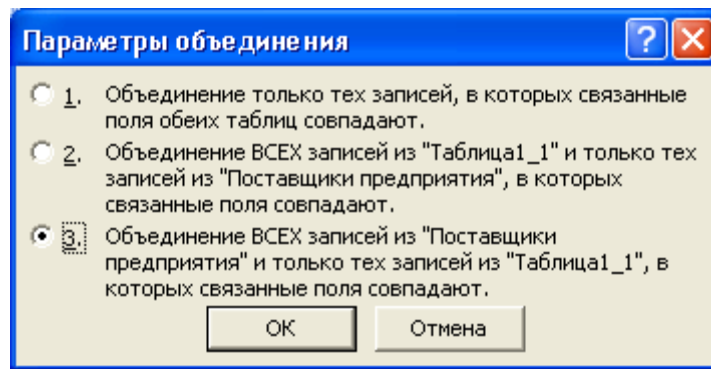


Рис. 2.4. Діалогове вікно Параметри об'єднання

Коли створюється новий зв'язок, можна також скористатися кнопкою Нове (Create New) і у вікні Створення (Create New) увести імена таблиць, що зв'язують, і імена полів, використовуваних для зв'язку. Нажати кнопку ОК.

Після установки всіх параметрів зв'язку необхідно нажати кнопку ОК у вікні Параметри об'єднання (Edit Relationship) і закрити вікно Схема даних (Relationship), нажавши на кнопку Закрити в правому верхньому куті вікна.

3. ПРОЕКТУВАННЯ ФОРМ І РОБОТА З НИМИ

Access надає можливість вводити дані як безпосередньо в таблицю, так і за допомогою форм. Форма в БД - це структуроване вікно, яке можна представити так, щоб воно повторювало форму бланка. Форми створюються з набору окремих елементів керування.

Зовнішній вигляд форми вибирається залежно від того, з якою метою вона створюється. Форми Access дозволяють виконувати завдання, які не можна виконати в режимі таблиці. Форми дозволяють обчислювати значення й виводити на екран результат. Джерелом даних для форми є записи таблиці або запиту.

Форма надає можливості для:

- уведення й перегляду інформації бази даних;
- зміни даних;
- печатки;
- створення повідомлень.

Способи створення форм:

- ✓ Конструктор форм (призначений для створення форми будь-якої складності)
- ✓ Майстер форм (дозволяє створювати форми різні як по стилі, так і по змісту)
- ✓ Автоформа: у стовпець (багатосторінкова - поля для запису виводяться в один стовпець, у формі одночасно відображаються дані для одного запису)
- ✓ Автоформа: стрічкова (всі поля записи виводяться в один рядок, у формі відображаються всі записи)
- ✓ Автоформа: таблична (відображення записів здійснюється в режимі таблиця)
- ✓ Автоформа: зведена таблиця
- ✓ Автоформа: зведена діаграма
- ✓ Діаграма (створюється форма з діаграмою, побудованої Microsoft Graph)
- ✓ Зведена таблиця (створюється форма Access, відображувана в режимі зведеної таблиці Excel)

Алгоритм створення форм наступний:

1. Відкрити вікно БД.
2. У вікні БД вибрати вкладку Форми.
3. Клацнути на піктограмі Створити, розташованій на панелі інструментів вікна БД.
4. У діалоговому вікні, що з'явилася, «Нова форма» вибрати спосіб створення форми й джерело даних (рис. 3.1).
5. Клацнути на кнопці ОК.

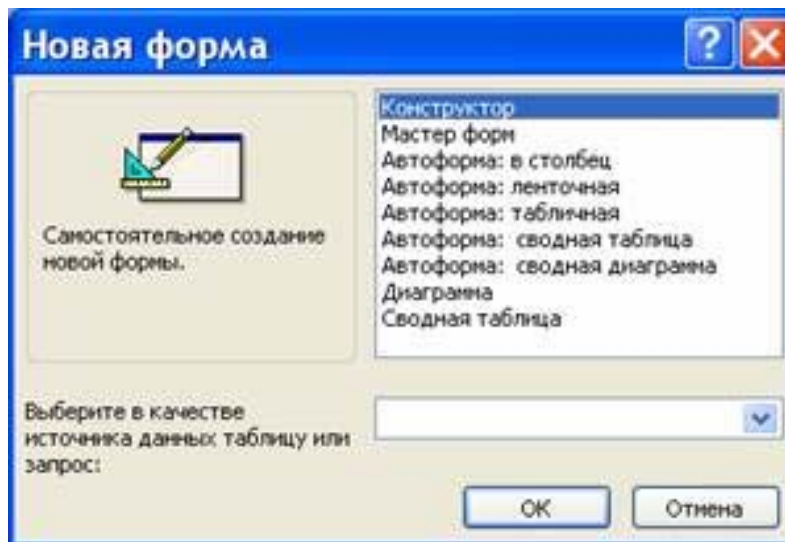


Рис. 3.1. Вікно «Нова форма»

Викликати Майстер форм (рис. 3.2) можна декількома способами. Один з них - вибрати Майстер форм у вікні діалогу Нова форма й клацнути на кнопці ОК. Відкриється вікно діалогу Створення форм, у якому необхідно відповідати на питання кожного поточного екрана Майстри й клацати на кнопці Далі.

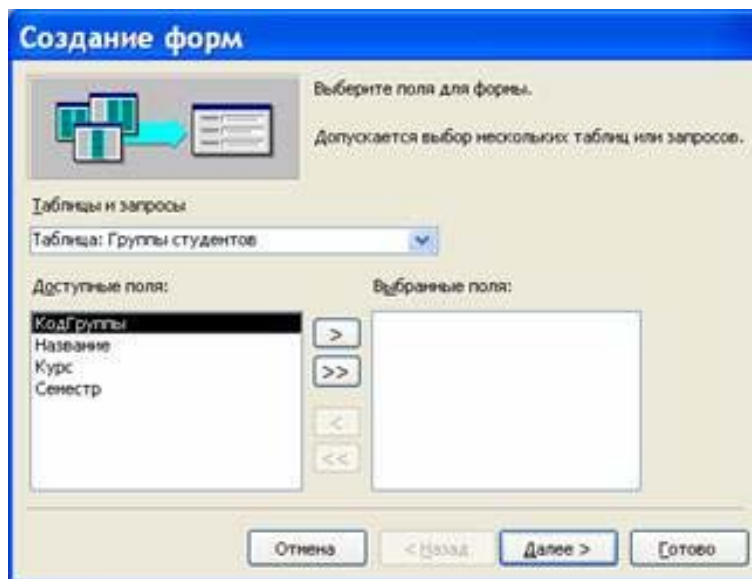


Рис. 3.2. Створення форми за допомогою Майстра (крок 1)

У першому вікні необхідно вибрати поля із джерела даних (таблиць або запитів). Для цього треба відкрити список Таблиці й запити, клацнувши на кнопку, праворуч. Потім доступні поля потрібно перевести в Обрані поля, виділивши їх і клацнувши на кнопку >> (рис. 3.3).

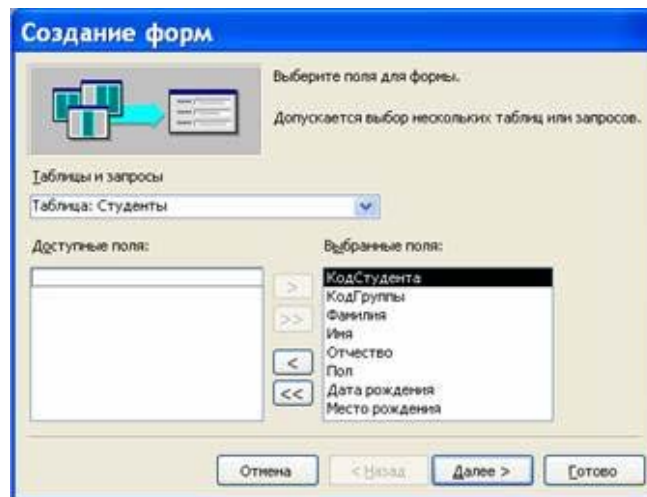


Рис. 3.3. Створення форми за допомогою Майстра (крок 2)

Наприклад, виберемо джерело - таблицю Студенти й всього її поля, а потім необхідно клацнути на кнопці Далі.

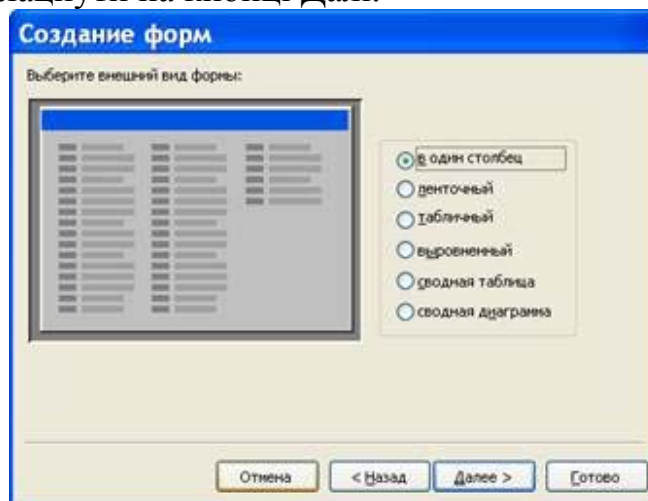


Рис. 3.4. Створення форми за допомогою Майстра (крок 3)

У цьому вікні треба вибрати зовнішній вигляд форми, наприклад в один стовпець і клацнути Далі.

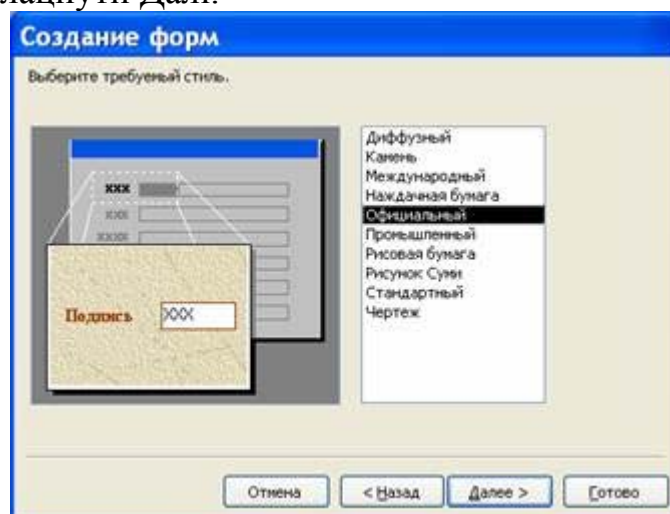


Рис.3.5. Створення форми за допомогою Майстра (крок 4)

Після вибору стилю форми (наприклад, офіційний), потрібно перейти в останнє вікно, клацнувши на кнопці Далі. В останнім вікні Майстра потрібно ввести ім'я форми й указати подальші дії: Відкрити форму для перегляду й введення даних; Змінити макет форми.

Після введення імені форми (наприклад, Студенти), вибору режиму: «Відкрити форму для перегляду й введення даних» і щиклика на кнопці Готово, одержимо наступну форму для введення й перегляду записів у таблицю Студенти (рис. 3.6).

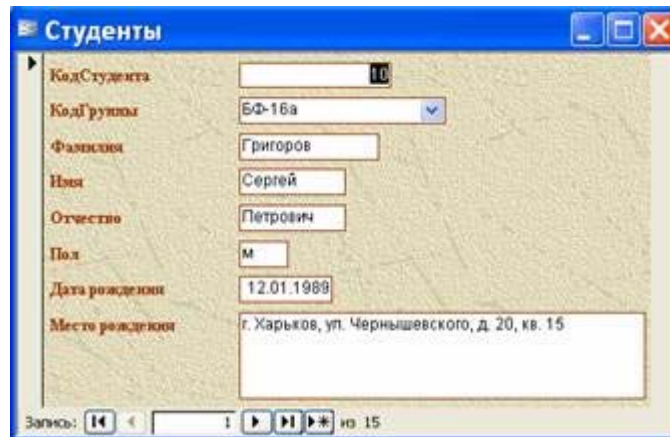


Рис.3.6. Створення форми за допомогою Майстра (крок 5)

Створення форми за допомогою Конструктора

Для створення форми Студенти необхідно виконати наступні дії:

1. Запустити програму Microsoft Access і відкрити БД.
2. У вікні БД вибрати вкладку Форми. Виконати щиклик по кнопці Створити. З'явиться діалогове вікно Нова форма. У цьому вікні необхідно вибрати зі списку пункт Конструктор. Потім у списку "Виберіть як джерело даних таблицю або запит" вибрати ім'я таблиці (наприклад, Студент). Виконати щиклик по кнопці ОК. На екрані з'явиться вікно Форма 1 (рис. 3.7).

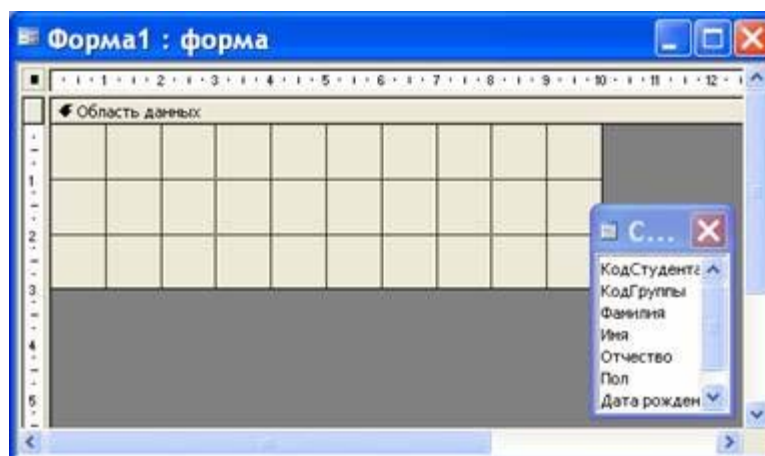


Рис. 3.7. Вікно Форма 1 Конструктора

3. Якщо на екрані відсутній список полів обраної для побудови форми таблиці, вибрати пункт меню Вид / Список полів.

4. Поля зі списку перемістити на форму (по одному або попередньо виділивши з використанням клавіші Shift і миші, для виділення всіх полів виконати подвійний щиглик мишею на заголовку вікна Список полів).

5. Розмістити поля на формі в потрібних місцях по розробленому зразку

6. Переміщення полів і їхніх імен за формою вироблятися в такий спосіб:

✓ Виділити об'єкт (поле з ім'ям) щигликом миші. Довкола нього з'являться маркери переміщення й зміни розмірів. Переміщати поле можна разом із прив'язаним до нього ім'ям або окремо від нього.

✓ Для переміщення помістити покажчик миші на квадратик, що перебуває в лівому верхньому куті елемента. Покажчик миші у вигляді долоні дозволяє переміщати об'єкт разом із прив'язаним до нього ім'ям, у вигляді долоні з витягнутим вказівним пальцем - переміщає один об'єкт.

✓ Нажати кнопку миші й, утримуючи її, буксирувати поле або його ім'я в потрібне місце у формі. Потім відпустити кнопку миші.

✓ Для зміни напису, пов'язаної з полем необхідно виконати на ній подвійний щиглик мишею. У діалоговому вікні, що відкрилося, Напис вибрати вкладку Макет і виконати необхідні зміни. Потім закрити вікно.

✓ Для зміни розмірів помістити курсор на розмірні маркери, при цьому курсор прийме вид двонаправленої стрілки. Нажати кнопку миші, буксирувати в потрібному напрямку, потім відпустити кнопку миші.

✓ Для видалення поля виділити його, нажати клавішу Delete або вибрати команду Виправлення / Видалити.

7. Зберегти форму, вибравши з меню Файл команду Зберегти як, і у вікні, що відкрилося, вибрати режим збереження «у поточній базі даних», потім щиглик по кнопці ОК.

8. Переглянути форму в режимі Конструктора, виконавши щиглика по кнопці Відкрити (рис. 3.8).

Студенты_конструктор : форма

Студенты

КодГруппы: BF-16a КодСтудента: 10

Фамилия: Григорев

Имя: Сергей

Отчество: Петрович

Дата рождения: 2.01.1989 Пол: м

Место рождения: г. Харьков, ул. Чернышевского, д. 20, кв. 15

Запись: 1 из 15

Рис. 3.8. Перегляд форми в режимі Конструктора

9. Якщо вид форми не задовольняє, відкрити форму в режимі Конструктор і внести необхідні зміни, потім зберегти форму Файл - Зберегти або виконати щиглика по піктограмі Зберегти.

4. РОБОТА З ДАНИМИ ЗА ДОПОМОГОЮ ЗАПИТІВ

4.1. Створення запиту на вибірку

За допомогою запитів можна переглядати, аналізувати й змінювати дані з декількох таблиць. Вони використовуються як джерело даних для форм і звітів. Запити дозволяють обчислювати підсумкові значення й виводити їх у компактному форматі, подібному до формату електронної таблиці, а також виконувати обчислення над групами записів.

Запити можна створювати самостійно й за допомогою майстрів. Самостійно розробити запити можна в режимі конструктора.

В Access можна створювати наступні типи запитів:

- ✓ запит на *вибірку*;
- ✓ запит з *параметрами* (критерій відбору задає користувач, увівши потрібний параметр при виклику запиту);
- ✓ *перехресний* запит (дозволяє створювати результуючі таблиці на основі результатів розрахунків, отриманих при аналізі групи таблиць);
- ✓ запит на *зміну* (видалення, відновлення й додавання) записів (дозволяє автоматизувати заповнення полів таблиць);
- ✓ запити *SQL* (на об'єднання, до сервера, що управляють, підлеглі), написані мовою запитів SQL.

Запит на вибірку використовується найбільше часто. При його виконанні дані, що задовольняють умовам відбору, вибираються з однієї або декількох таблиць і виводяться в певному порядку.

Для створення запиту варто перейти у вкладку «Запросы» у вікні бази даних, вибрати кнопку «Создать» (рис. 4.1), потім варіант «Конструктор».

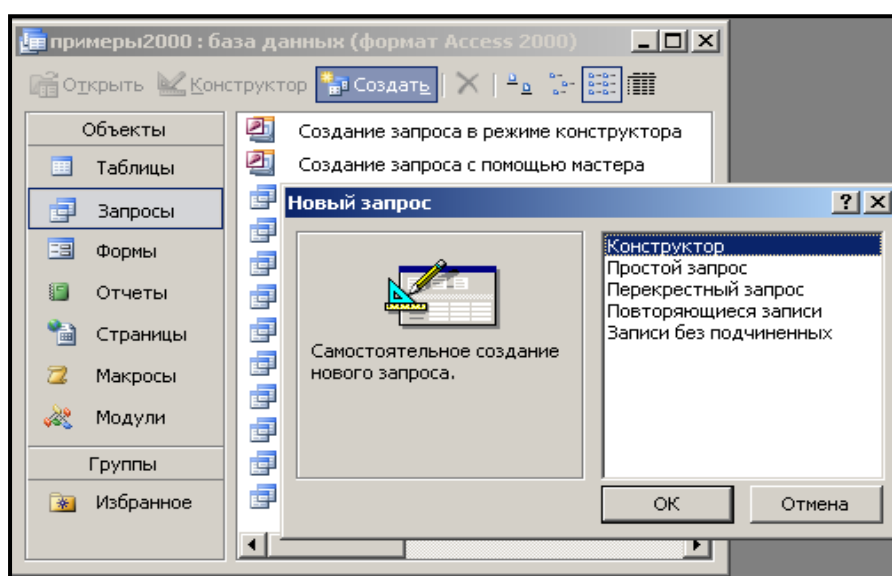


Рис. 4.1. Вікно створення нового запиту

Наступне вікно – «Добавление таблицы» використовується для вибору таблиці, на підставі якої створюється запит. Обрану таблицю варто висвітити курсором і натиснути спочатку на кнопку «Добавить», а потім «Закерить» (рис. 4.2).

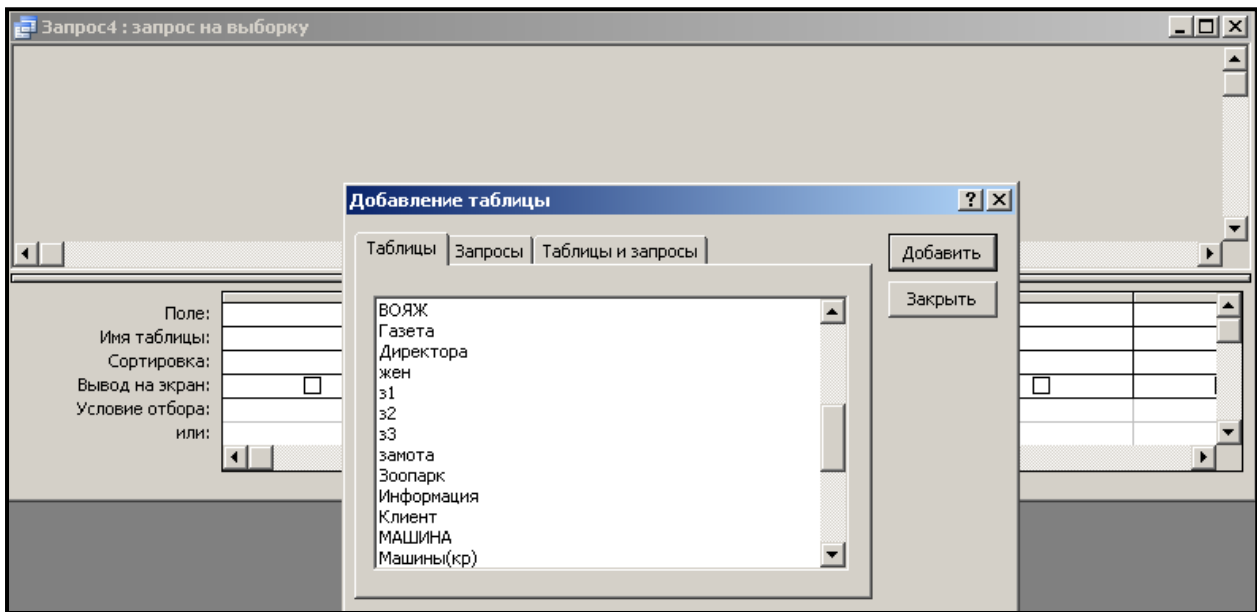


Рис. 4.2. Вікно «Добавление таблицы» для выбора таблицы

Вікно «Конструктора запросов» розділено на дві частини. У верхній частині - список полів таблиці у вигляді невеликого прямокутника, у нижній частині формується безпосередньо запит (рис. 4.3).

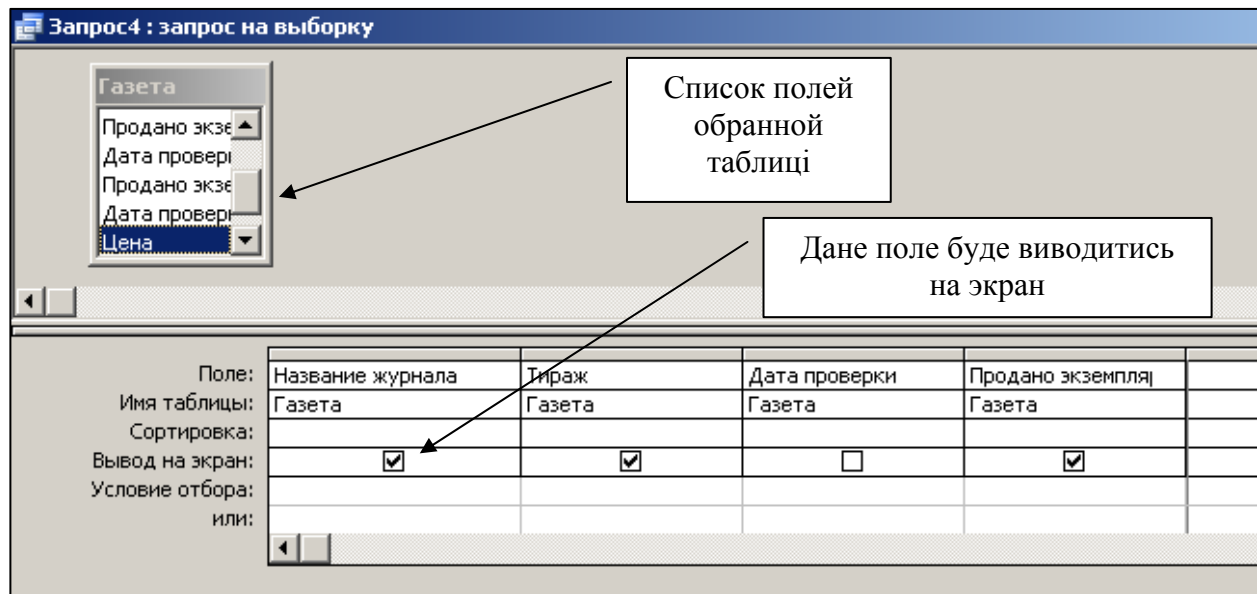


Рис. 4.3. Вікно запиту в режимі Конструктора

Спочатку визначають, які поля потрібні в запиті. Це можуть бути просто інформаційні поля й поля пошукові. Наприклад, Вас цікавлять всі співробітники, чий оклад перевищує, допустимо, тисячу грн. Пошуковим полем буде поле «Оклад». Але якщо при виконанні запиту з'являться тільки цифри, навряд чи він буде досить інформативний. От якщо додати поле «Фаміліи», то стане зрозуміло, кому й скільки нараховує бухгалтерія. Це поле інформаційного типу.

Перенести потрібні поля в нижню частину екрана можна подвійним щигликом по їхній назві в прямокутнику з назвами полів. При цьому назви полів з'являються в рядку «Поле» і автоматично заповнюється рядок «Имя таблицы», фіксуючи, з якої таблиці або запиту взяте поле.

Якщо в третьому рядку задане сортування для даного поля, то знайдені записи будуть відсортовані у встановленому порядку. Прапорці в рядку «Вывод на экран» означають ті поля, інформація з яких буде виведена на екран, після цього їх можна переглядати, друкувати й т.д. Якщо прапорець не включений, поле бере участь у конструюванні запиту, але не виводиться візуально.

Правила проектування умов для відбору записів тісно пов'язані з типом полів.

Текстові поля. Якщо відомо все слово, його приводять повністю в рядку «Условие отбора»; якщо частина, то використовують шаблони * і ? , перший заміняє будь-яка кількість символів, другий – один символ. Наприклад, **“Вас*”** дозволить знайти в поле «Фамилия» людей із прізвищем Васин, Васильєв, Васисуалий, Васько й т.д.

Запис **“М???до”** дозволить знайти в поле «Фамилия» людей із прізвищем, що починається на букву **М** і завершується на **-до**, при цьому складається з 6 символів.

Для запису складних критеріїв використовуються логічні операції.

Приклад.

Магазини продають цукерки. Знайти ціни на цукерки певних назв.

Для знаходження ціни на цукерки «Белочка» і «Коровка» в умові відбору використовують логічну операцію **IN** і в дужках перераховують необхідні найменування (рис. 4.4).

| | | | |
|-------------------------|-------------------------------------|-------------------------------------|--|
| Поле: | Наименование | Цена | |
| Имя таблицы: | Магазин | Магазин | |
| Сортировка: | | | |
| Вывод на экран: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| Условие отбора: или: | [n ("Белочка";"Коровка")] | | |

Рис. 4.4. Приклад використання логічної операції **IN**

Числові поля. При записі умов використовуються знаки порівняння, логічні оператори, символи шаблона.

>2 - більше 2;

Більш 2 або менш 20.

>2 AND < 20

Інший варіант завдання умови більш 2

BETWEEN 2 AND 20

або менш 20.

<=3 - менше або дорівнює 3;

>=30 OR <>0 - більше або дорівнює 30 або не дорівнює 0

Поля типу «Дата/час». При складанні запитів з полями зазначених типів при пошуку конкретної дати, неї вводять в «Умову відбору», при запуску запиту автоматично з'являться обмежуючі значки #, а для значення часу з'являться й секунди. Умови відбору можуть бути наступними:

>01.01.05 AND <01.04.05 - 1 квартал 2005 року; <>8.03.2006 - за винятком 8 березня 2006 року;

Для здійснення можливості багаторазового виконання запиту при зміні умов відбору, необхідно в нього додати параметр. Щоб визначити параметр, у рядок «Умова відбору» вводиться ім'я або фраза, укладені у квадратні дужки, як це показано на рис. 4.5.

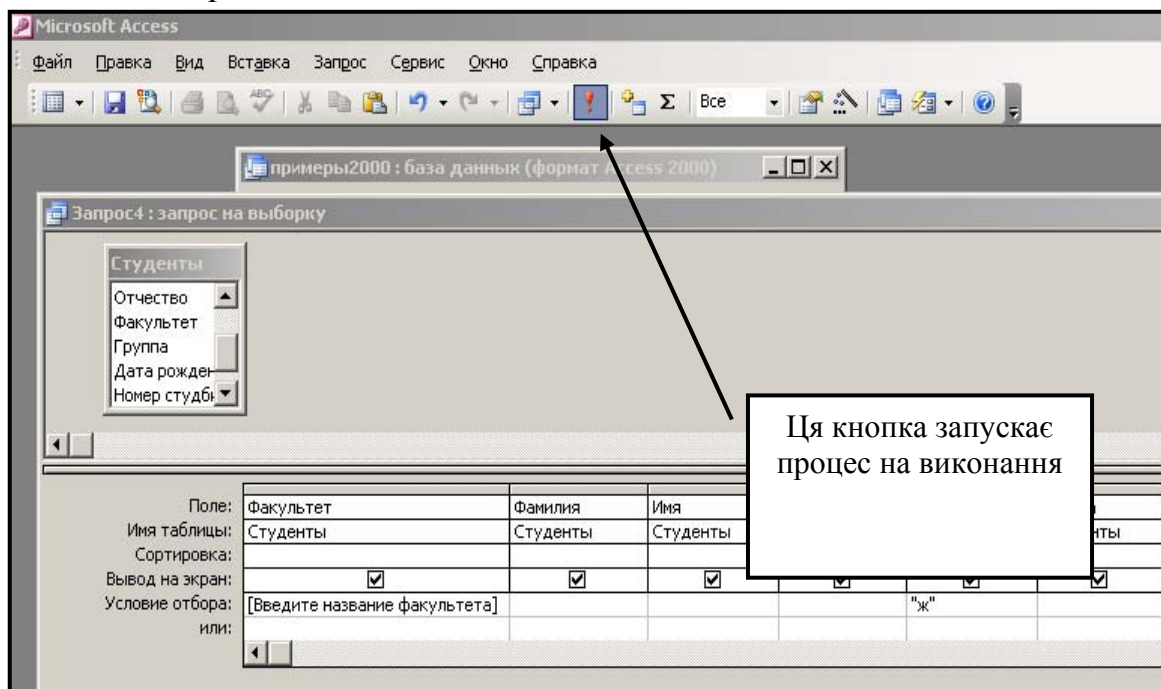



Рис. 4.5. Вікно конструктора параметричного запиту

Для запуску на виконання запиту виконують команду «Запрос»/«Запуск» або клацають по кнопці  на панелі інструментів.

Після цього з'являється діалогове вікно, показане на рис. 4.6. У вікні вводять пошуковий критерій, допустимо слово «экономический» для пошуку даних по економічному факультеті.

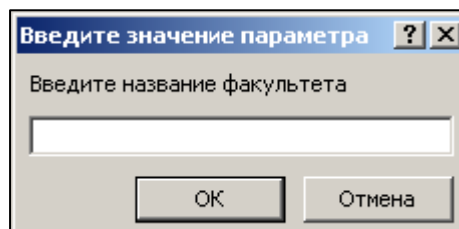



Рис. 4.6. Діалогове вікно параметричного запиту

Для збереження запиту виконати команду «Файл»/«Зберегти запит». Запити можна використати багаторазово, викликаючи їх по іменах.

4.2. Обчислювальні поля

За допомогою запиту можна створювати обчислювальні поля. Це тимчасові поля, у які заносяться результати обчислень над значеннями інших полів.

Для створення обчислює поля, що, необхідно:

1. Установити курсор у рядок «Поле» вільного стовпця;
2. На панелі інструментів натиснути кнопку  **Побудувати**, відкривши вікно побудовника виражень (рис. 1.23);
3. У вікні «Построитель» увести назву нового поля, відокремивши його двокрапкою від розрахункової формули.
4. У нижній частині вікна Побудовника перебувають три списки. У лівому списку виводяться папки, що містять об'єкти бази даних, функції, константи, оператори, які можна використати у вираженнях. При виборі необхідної папки, у середньому списку відображаються вхідні в неї елементи. Вибравши елемент у середньому списку, у правому вибирають значення (якщо вони існують). Наприклад, якщо вибрати в лівому списку «Убудовані функції», у середньому - потрібну категорію функції, то праворуч буде виведений список всіх убудованих функцій обраної категорії.

Приклад.

Знайти споживану потужність машини.

Приклад розглянутий на рис. 4.7.

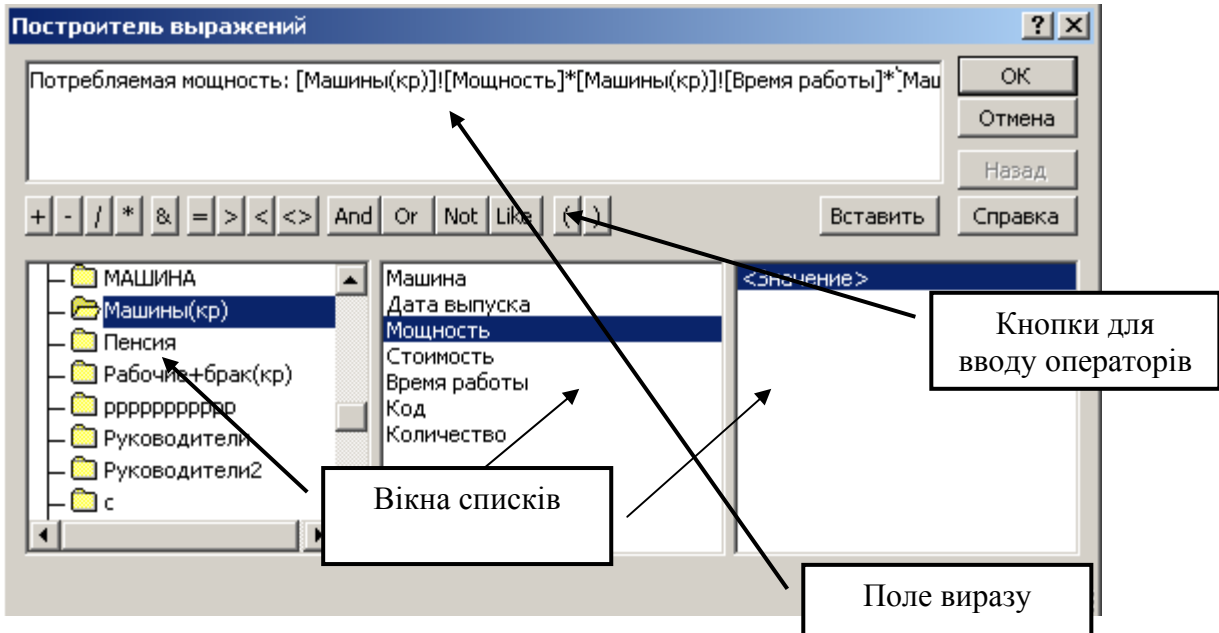


Рис. 4.7. Вікно Побудовника виражень

При введенні формули небажано друкувати назви полів вручну, досить їхнім подвійним щикликом, висвітлюючи в середньому списку.

Розглянемо приклад обчислює поля, що. Нехай у таблиці є інформація про потужності, споживану кожною машиною в годину, їхній кількості й часі роботи. Для розрахунку «Расходование энергии» всіх машин за увесь час

необхідно помножити потужність кожної машини на їхню кількість і час роботи в годинниках. Для цього в побудовник виражень заноситься наступний запис:

Споживана потужність: [Машини]![Потужність]*[Машини]![Час роботи]*[Машини]![Кількість]

Зверніть увагу, полючі таблиці з'являються у квадратних дужках у супроводі назви таблиці й відділяються від імен таблиць знаком оклику.

Іноді потрібно об'єднати текстові поля. Наприклад, у Вашій таблиці окремо втримуються поля «Город», «Улица», «Дом» і «Квартира». Потрібно створити поле «Адрес». Створюємо запит, у який виносимо поле «Фамилия» і викликаємо побудовник виражень.

На рис. 4.8 представлена вихідна таблиця, на підставі якої створюється запит.

| | Фамилия | Имя | Результ | Дата | Город | Улица | Дом | КВ |
|---|---------|--------|---------|----------|--------|-----------------|-----|----|
| ▶ | Васин | Костя | 12 | 12.12.96 | Тула | Садовая | 35 | 10 |
| | Васькин | Вася | 12 | 11.01.89 | Донецк | Университетская | 30А | 7 |
| | Кикина | Варвар | 0 | 08.03.99 | Донецк | Доменная | 88 | 9 |
| | Минин | Миша | 34 | 31.12.97 | Орел | Овражная | 12 | 8 |

Рис. 4.8. Вихідна таблиця

Таблиця, що містить необхідні поля, називається «Экзамены». Поля потрібно приписати друг до друга. Як з'єднувач полів використаний знак конкатенації & . Ми не дали новому полю назва й автоматично з'явився напис «Выраз1» (рис. 4.9).

| | Фамилия | Выражение1 |
|---|---------|---------------------------|
| ▶ | Васин | ТулаСадовая3510 |
| | Васькин | ДонецкУниверситетская30А7 |
| | Кикина | ДонецкДоменная889 |
| | Минин | ОрелОвражная128 |
| * | | |

Рис. 4.9. Створення поля «Адрес» поки під ім'ям «Выраз1»

Результат виконання запиту представлений на рис. 4.10.

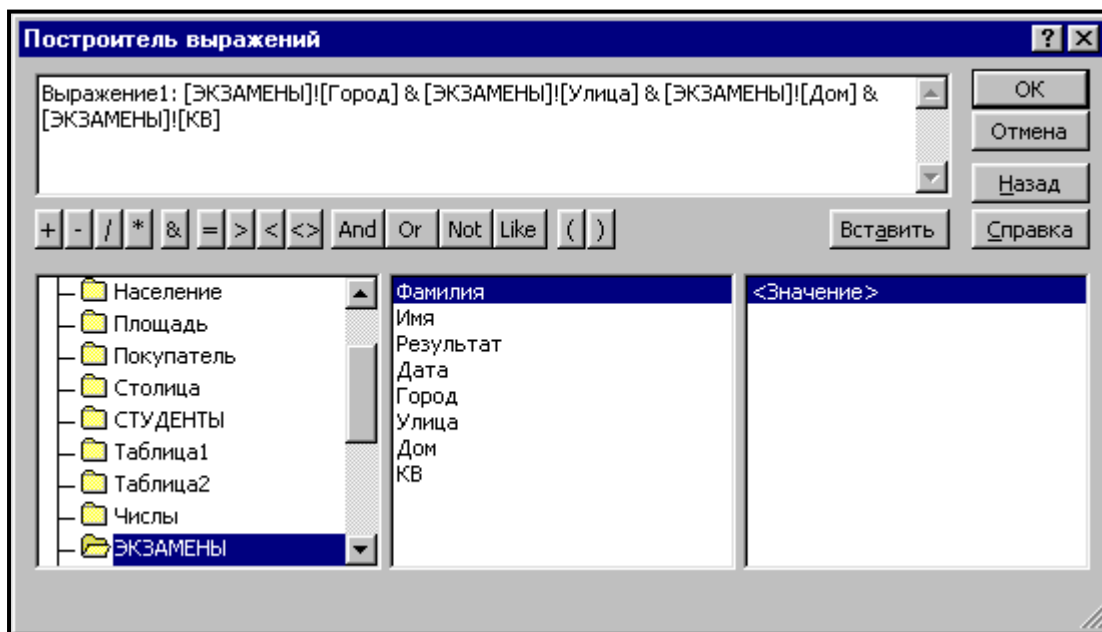


Рис. 4.10. Результат виконання запиту

Зовнішній вигляд, отриманий у результаті виконання такого запиту, досить далекий від ідеалу. Модифікуємо вираження, додавши в нього назва поля й знаки роздільника між складовими адреси. Вікно «Побудовника» буде виглядати в такий спосіб (рис. 4.11):

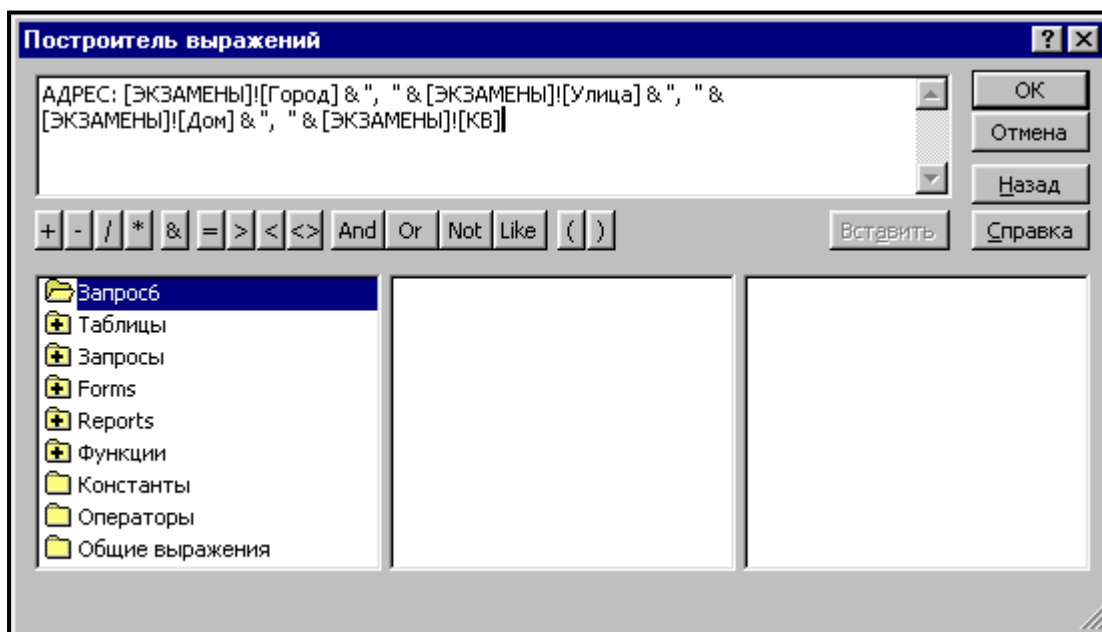


Рис. 4.11. Вікно Побудовника вираження з модифікованим запитом

У верхнім вікні Побудовника записана наступна формула:

АДРЕСА: [ИСПИТИ]![Місто] & ", " & [ИСПИТИ]![Булиця] & ", " & [ИСПИТИ]![Будинок] & ", " & [ИСПИТИ]![КВ]

Знак & з'єднує окремі елементи, у лапках - роздільники. Результат виконання запиту, представлений на рис. 4.12, відповідає правилам орфографії. Ви можете за допомогою знака конкатенації з'єднувати існуючі поля й довільний текст, що вводиться в подвійних лапках ". У приведенному прикладі, це кома й кілька пробілів, для відділення назв друг від друга.

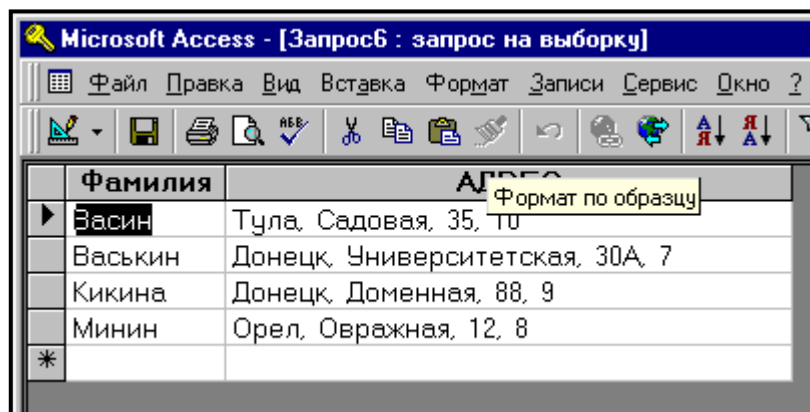



Рис. 4.12. Остаточний результат виконання запиту

4.3. Підсумкові запити

На практиці іноді виникає необхідність виконати запит для груп даних, або одержати підсумкові значення по групах рядків. Такі запити називаються «Ітоговими».

Наприклад, необхідно підрахувати число травмованих робітників по кожному цеху за поточний рік. У цьому випадку використаються підсумкові запити.

Для створення групового запиту необхідно виконати команду «Вид»/ «Групповые операции» або натиснути на панелі інструментів кнопку, , у результаті чого у вікні

Конструктора запитів з'явиться новий рядок «Групповая операция». Вибравши поле, по якому буде виконуватися групова операція, варто замовити одну зі статистичних функцій, список яких приводиться в табл. 4.1.

Таблиця 4.1

Основні групові функції

| Позначення | Пояснення |
|------------|---|
| SUM | Використається для визначення суми всіх значень обраного поля в кожній групі |
| AVG | Використається для визначення середнього арифметичного значення обраного поля в кожній групі |
| MIN, MAX | Використається для визначення відповідно мінімального й максимального значення, знайденого в обраному полі усередині кожної групи |
| COUNT | Використається для визначення кількості записів у поле |
| STDEV | Використається для визначення статистичного стандартного відхилення всіх значень даного поля в групі |

| Позначення | Пояснення |
|-------------|--|
| VAR | Використається для визначення статистичної дисперсії |
| FIRST, LAST | Використається для визначення відповідно першого й останнього значення в групі |

Якщо необхідно зробити розрахунки не по всіх записах, а тільки по певним, у рядку «Условия отбора» задаються відповідні умови. При необхідності вказується й порядок сортування.

Якщо в запит додається обчислювальне поле, використовуючи одну або кілька статистичних функцій для рас чету підсумкових значень, необхідно вибрати в осередку «Групповая операция» цього поля значення «Вираз» (рис. 4.13).

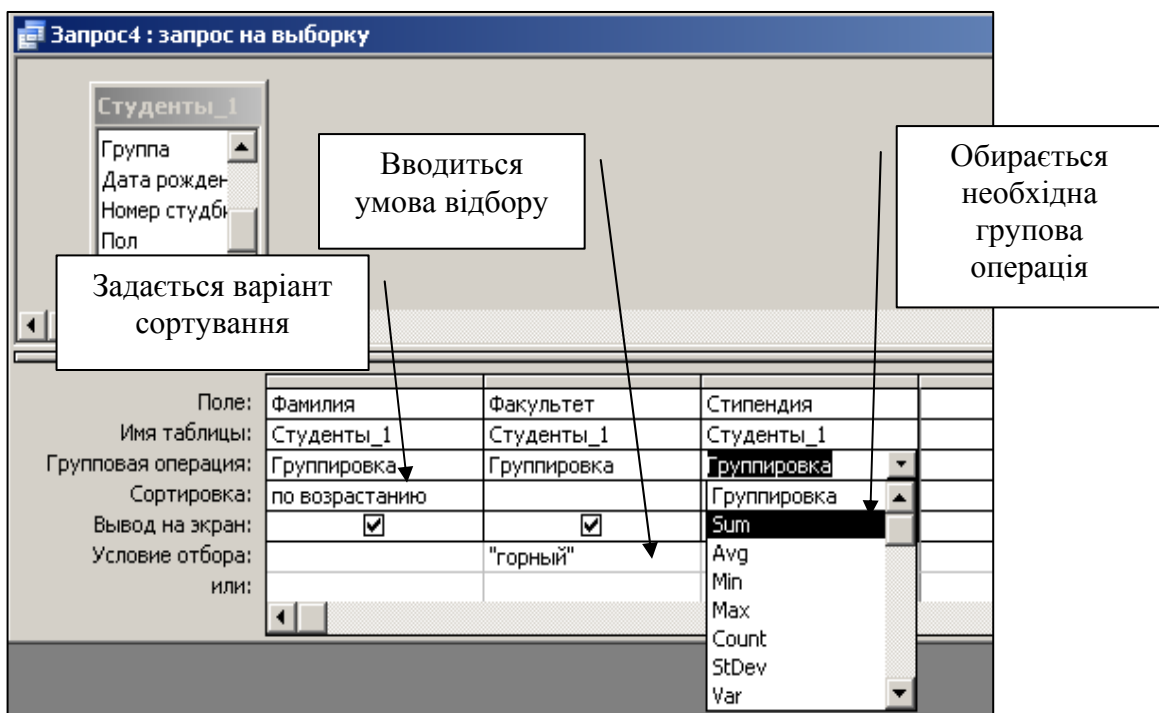


Рис. 4.13. Режим конструктора при створенні підсумкового запиту

4.4. Використання функцій дати/часу

Допустимо, нами була створена база даних, утримуючого поля, зазначені на (рис. 4.14) , причому поля «Дата іспиту» і «Дата народження» мають «Короткий формат дати», а поля «Время начала экзамена» і «Время окончания» - «Короткий формат времени».

Приклад.

Знайти всіх студентів певного факультету.

Приклад розглянутий на рис. 4.13, 4.14.

| Имя поля | Тип данных | |
|-----------------------|------------------------|-------------------|
| Фамилия | Текстовый | |
| Имя | Текстовый | |
| Оценка | Числовой | |
| Дата экзамена | Дата/время | |
| Время начала экзамена | Дата/время | |
| ▶ Время окончания | Дата/время | |
| Дата рождения | Дата/время | |
| | | |
| | | |
| Свойства поля | | |
| Общие | | |
| Подстановка | | |
| Формат поля | Краткий формат времени | |
| Маска ввода | Полный формат даты | 19.06.94 17:34:23 |
| Подпись | Длинный формат даты | 19 июня 1994 г. |
| Значение по умолчанию | Средний формат даты | 19-июн-94 |
| Условие на значение | Краткий формат даты | 19.06.94 |
| Сообщение об ошибке | Длинный формат времени | 17:34:23 |
| Обязательное поле | Средний формат времени | 5:34 |
| Индексированное поле | Краткий формат времени | 17:34 |

Рис. 4.14. Створення таблиці бази даних

При складанні запитів з полями зазначених типів при пошуку конкретної дати, неї вводять в «Условие отбора», при цьому при запуску запиту з'являться обмежуючі значки #, наприклад, при уведенні умови = #1.2.06# з'являться всі іспити, що доводяться на цю дату. Однак, такий варіант роботи має обмежені можливості. Якщо потрібно знайти всіх, хто, допустимо, народився в травні, навряд чи це вдасться зробити за допомогою однієї умови. Для створення складних запитів, пов'язаних з полями зазначених типів, використовуються убудовані функції.

Найбільш популярні функції:

DATE() - повертає поточну системну дату комп'ютера у вигляді 1.7.2006;

DAY([дата]) - повертає день місяця. Обов'язковий аргумент [дата] може бути будь-яким значенням типу «ДАТА», наприклад, ім'ям поля відповідного типу або просто довільною датою.

MONTH([дата]) - повертає номер місяця;

YEAR([дата]) - повертає рік;

WEEKDAY([дата]) - повертає номер дня тижня, причому нумерація днів починається з неділі (№ 1), понеділок - №2, вівторок - №3 і т.д.

HOURL([час]) - використовується для визначення години, що відповідає зазначеному часу;

MINUTE ([час]) - використовується для визначення хвилин, що відповідає зазначеному часу;

SECOND([час]) - використовується для визначення секунд, що відповідають зазначеному часу;

TIME() - повертає поточний системний час;

NOW() - повертає поточну дату й час.

Приведемо кілька прикладів використання функції типу **Дата/час**.

Приклад 1. Для визначення кількості днів, що пройшли із часу іспиту, використається вираження, записане за допомогою Побудовника виражень (рис. 4.15).

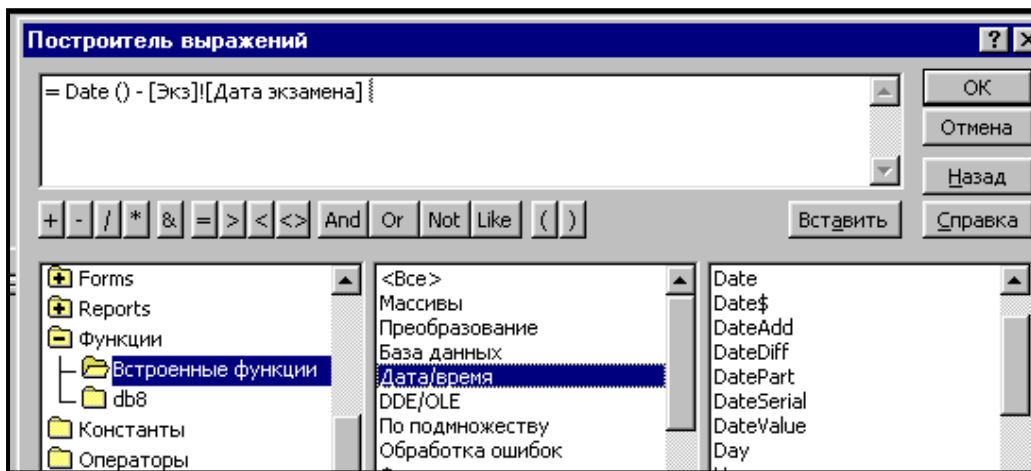


Рис. 4.15. Обчислюємо дні, що пройшли від дня іспиту

Приклад 2. Для визначення прізвищ тих, хто здавав іспит у п'ятницю **WEEKDAY[Дата іспиту]=6** і 13-ого числа **DAY[Дата іспиту]=13** використаємо запит, конструктор якого наведений на (рис. 4.16).

| | | |
|-----------------|-------------------------------------|--|
| Поле: | Дата экзамена | Тяжелый день: Day([Экз]![Дата экзамена]) |
| Имя таблицы: | Экз | |
| Сортировка: | | |
| Вывод на экран: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Условие отбора: | | =13 And Weekday([Экз]![Дата экзамена])=6 |
| или: | | |

Рис. 4.16. П'ятниця, 13-ої число

Спробуйте за аналогією знайти всіх, хто народився у вихідні дні.

Приклад 3. Для визначення студентів, що народилися у високосні роки, створюється запит, наведений на (рис. 4.17).

| | | |
|-----------------|-------------------------------------|--|
| Поле: | Фамил | Дата рождения |
| Имя таблицы: | Экз | Экз |
| Сортировка: | | |
| Вывод на экран: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Условие отбора: | | Int(Year([Экз]![Дата рождения])/4)=Year([Экз]![Дата рождения])/4 |
| или: | | |

Рис. 4.17. Пошук високосного років

Якщо буде потрібно знайти всіх, хто народився в роки, кратні п'яти, що зміниться у формулі?

Приклад 4. Для визначення прізвищ всіх, що здали іспити в 2006 році, використають вираження: **Year([Экз]![Дата іспиту])=2006**, що вводять у рядок «Умова відбору». Для виділення місяця й року з поля застосовують вираження:

Місяць: **Month([Экз]![Дата іспиту]) ;» **Рік:** **Year([Экз]![Дата іспиту])****

Якщо необхідно довідатися, скільки днів пройшло між двома датами, визначають їхню різницю. Але іноді потрібно визначити кількість місяців, що пройшли між двома датами, вік людини в літах, строки виконання роботи у кварталах. Access надає для цього спеціальні функції.

Визначення різниці між датами роблять за допомогою функції **DATEDIFF(інтервал; початкова дата; кінцева дата)**. На першому місці задається інтервал (у чому вимірюється різниця). Якщо це рік, то використовують уууу, **q** -квартали; m- місяці; d- у днях; ww – у тижнях; h – у годинниках; n – у хвиликах; s – у секундах.

✓ **Вік1 : DateDiff ("уууу"; [Экз]![Дата народження]; [Экз]![Дата іспиту])** - визначає вік студента на момент здачі іспиту в літах;

✓ **Вік2: DateDiff("m"; [Экз]![Дата народження]; [Экз]![Дата іспиту])** - визначає вік студента на момент здачі іспиту в місяцях.

Складіть вираження, що дозволяє визначити стаж роботи, якщо відома дата надходження на роботу й дата відходу на пенсію.

Функція **DATEADD(інтервал, число, дата)** призначена для додавання або вирахування зазначеного тимчасового інтервалу зі значення дати. Наприклад, за допомогою цієї функції можна обчислити дату, на 30 днів більше пізню, чим поточна, або час, на 45 хвилин більше пізніше, ніж сьогодні.

Для додавання днів до аргументу дата можна задавати часовий інтервал як день року ("y"), день місяця ("d") або день тижня ("w").

Приклад 1. Для визначення дати, що була **через півтора року після дня народження** в побудовнику виражень варто ввести наступний запис:

Дата: DateAdd("m";18;[Экз]![Дата народження])

Приклад 2. Поле «Нова дата» визначає дату через 10 років після дня народження, для чого в побудовнику виражень уводиться наступний запис:

Нова дата: DateAdd("уууу";10;[Экз]![Дата народження]).

Приклад 3. Для визначення всіх, що прийшли на іспит до 11 годин і збіглих не пізніше 14, необхідно створити запит, конструктор і результат виконання якого наведений на (рис. 4.18).

| | | |
|-----------------|---|-----------------------------------|
| Поле: | Время начала экзамена | Время окончания |
| Имя таблицы: | Экз | Экз |
| Сортировка: | | |
| Вывод на экран: | <input checked="" type="checkbox"/> | |
| Условие отбора: | Hour([Экз]![Время начала экзамена])<=11 | Hour([Экз]![Время окончания])<=14 |
| или: | | |

| | Фамилия | Время начала | Время окончания |
|---|---------|--------------|-----------------|
| ▶ | Васин | 10:45 | 12:34 |
| | Ильин | 11:23 | 13:45 |
| | Кошина | 11:34 | 14:45 |

Рис. 4.18. Конструктор запиту й результат виконання

4.5. Використання функцій керування при створенні запитів

Функції «Керування» використовуються для перевірки умов і виконання набору деяких дій залежно від результатів перевірки. Для їхнього застосування використовують «Побудовник» (рис. 4.19).

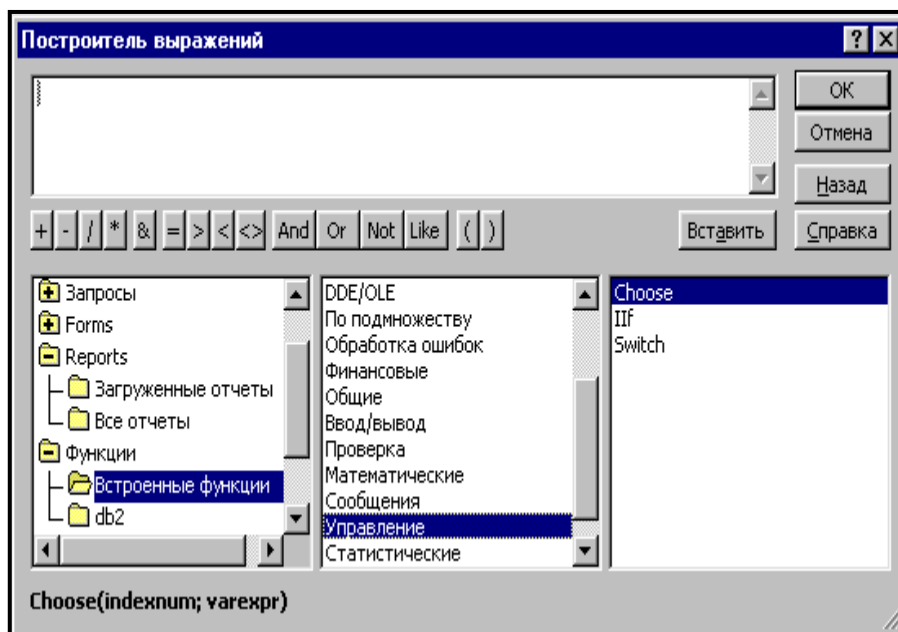


Рис. 4.19. Список функцій керування

Принципи дій операторів вивчимо на прикладі таблиці «ВОЯЖ» (рис. 4.20). Допустимо, потрібно знайти нову вартість квитків, якщо їхньої зміни відбулися по наступному принципі: для поїздок, що доводяться на період із травня по жовтень, вартість квитка варто збільшити на 25%; на інші місяці зменшити на 10%.

| | Страна | Вид транспорта | Стоимость билета | Стоимость проживания | Дата отъезда | Дата приезда |
|---|----------|----------------|------------------|----------------------|--------------|--------------|
| | Франция | Самолет | 500,00р. | 34,00р. | 04.08.99 | 15.08.99 |
| | Франция | Поезд | 234,00р. | 12,00р. | 06.06.99 | 15.09.99 |
| | Германия | Самолет | 400,00р. | 23,00р. | 05.05.98 | 06.11.98 |
| ▶ | Болгария | Поезд | 100,00р. | 45,00р. | 06.09.97 | 12.07.97 |
| | Турция | Море | 78,00р. | 12,00р. | 04.08.99 | 09.08.99 |

Рис. 4.20. Таблица вихідних даних

Для виконання поставленого завдання використаємо функцію ІФ (рис. 4.21). Синтаксис: **Иф**(умова; дія по «ТАК» ; дія по «Ні»)

Призначення: перевірити умова й залежно від результатів його перевірки, чи виконати та дія по «ТАК», чи та дія по «Ні».

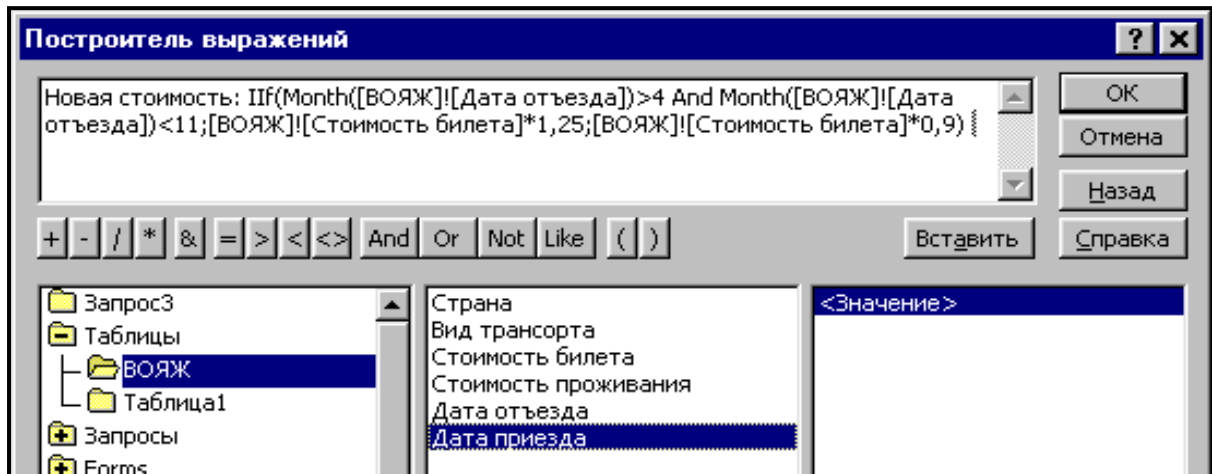


Рис. 4.21. Побудовник вираження при використанні функції керування

Нова вартість:

IIf(Month([ВОЯЖ]![Дата від'їзду])>4 And Month([ВОЯЖ]![Дата від'їзду])<11; [ВОЯЖ]![Вартість квитка]*1,25;[ВОЯЖ]![Вартість квитка]*0,9).

Коментарі до формули ми привели нижче.

| Нова вартість: | Назва нового поля |
|---|-------------------------------|
| Month([ВОЯЖ]![Дата від'їзду])>4 And Month([ВОЯЖ]![Дата від'їзду])<11 | Умова перевірки номера місяця |
| [ВОЯЖ]![Вартість квитка]*1,25 | дія по «ТАК» |
| ВОЯЖ]![Вартість квитка]*0,9 | дія по «НІ» |

Функція **IIf** може бути використана в укороченому варіанті: **IIf**(умова; дія по «ТАК»). Наприклад, потрібно підвищити вартість проживання тільки у Франції. Новий запит виконується на базі вже наявного - це робиться в такий спосіб:

Нова вартість: **IIf**([ВОЯЖ]![Країна]= Франція";
[ВОЯЖ]![Вартість проживання]*2).

Для того, щоб можна було використати поля базового запиту, його потрібно попередньо зберегти.

Можна використати вкладені функції **IIf**, які перевіряють кілька умов. Наприклад:

ПОБАЖАННЯ: IIf([ДНІ]<10;"Рано їдете!"; **IIf**([ДНІ]<20; "Приїдьте ще!";**IIf**([ДНІ]<30;"Може побачимо!";".....!!!!"))

У запиті розписані можливі варіанти невисловлених думок хазяїв, залежно від строку проживання гостей.

Якщо кількість перевірок більше двох або трьох, функцією **IIf** стає незручно користуватися й замість її застосовують **SWITCH**.

Синтаксис: **SWITCH**(умова 1; дія 1 ; умова 2; дія 2; умова 3; дія 3...)

Призначення: містить пари виражень і значень. Вираження обчислюються в тім порядку, у якому вони включені в список (ліворуч праворуч). Вертається значення, що відповідає першому щирому вираженню в списку.

Приклад: Країна = **SWITCH**([CityName] = "Лондон", "Англія", [CityName] = "Рим", "Італія", [CityName] = "Москва", "Росія") - якщо в поле [CityName] зазначений Лондон, то Ви перебуваєте в Англії.

У наступному прикладі функція **SWITCH** використовується для визначення сезону по відомих датах. Нагадуємо, що **Month()** використовується для виділення місяця з дати. Зверніть увагу на слово ІСТИНА вираження, що приводить наприкінці. Воно використовується в тому випадку, коли потрібно виконати яка-небудь дія, якщо всі попередні умови виявилися нездійсненними. Так, якщо аналізований місяць перебуває поза розглянутими межами, те цей сезон називається «ЗИМА». Слово ІСТИНА вводиться без лапок.

Сезон: **SWITCH**(Month([Екскурсія]![Дата])<6 And Month([Екскурсія]![Дата])>2; "весна";Month([Екскурсія]![Дата])<9 And Month([Екскурсія]![Дата])>5;"літо"; Month([Екскурсія]![Дата])<12 And Month([Екскурсія]![Дата])>8;"осінь";Истина;"зима").

Спробуйте скласти вираження, у якому ціна вартості квитків збільшується на 35% у літній сезон, падає на 15% в осінне - весняний і збільшується на 5% у зимовий.

Наступною функцією керування є функція **CHOOSE**.

Синтаксис: **CHOOSE**(покажчик; дія 1 ; дія 2; дія 3...)

Призначення: повертає дія зі списку, обрана на підставі значення аргументу - покажчика. Покажчик завжди зважає на 1. Якщо покажчик рівняється 1, вертається перший елемент списку, якщо покажчик рівняється 2, вертається другий елемент списку й т.п. Найчастіше застосовують для заміни числових виражень текстовими еквівалентами.

Приклад. Допустимо, що при уведенні даних у таблицю «Екскурсія», ми не стали записувати назву транспорту, а ввели кодування. Якщо вводилася 1, то це відповідало морському транспорту, 2 - поїзду, 3 - автобусу. У запиті поле транспорт визначали як:

Транспорт(текст): **Choose**([Екскурсія]![Транспорт]; "море"; "поїзд"; "автобус").

4.6. Формування складних запитів

Запити, які вивчалися нами раніше, називаються *запитами-вибірками*. При їхньому збереженні в пам'яті машини зберігаються тільки задані умови пошуку. Щораз при запуску запиту на виконання виникає так званий *динамічний набір даних*. Він не існує реально.

Запит на зміну - це запит, що за одну операцію вносить зміни в кілька записів. Існує чотири типи запитів на зміну: на видалення, на відновлення, додавання записів, на створення таблиці.

Запит на видалення використовується для автоматичного видалення групи записів, що задовольняють певним умовам з однієї або декількох таблиць. Причому можна видаляти тільки весь запис, а не окремі поля усередині неї.

Операцію видалення записів скасувати неможливо, тому рекомендується перед виконанням створити резервну копію таблиці.

Послідовність дій:

1. У режимі конструктора запитів увійти в меню «Запрос» і вибрати команду «Удаление». З'явиться додатковий рядок «Выделения».
2. У рядку «Умова відбору» указати, які записи повинні віддалятися.
3. Спочатку рекомендується виконати команду «Запрос»/«Выполнить», щоб переглянути видаляють записи, що, а потім «Запрос»/«Удалить».

На (рис 4.22) показаний вікно конструктора, у якому зі складу студентів, що одержують стипендію, будуть виключені ті, хто здав іспити на два. Зверніть увагу, у тій версії, що реалізована в нас, стипендії втрачають студенти, що одержали двійки по трьох іспитах, якби умови здачі на два по англійській мові ми перенесли на один рядок нижче (рядок «аб»), а двійка по математиці була б записана ще нижче, те це означало б пошук студентів, що одержали двійку хоча б по одному іспиті.

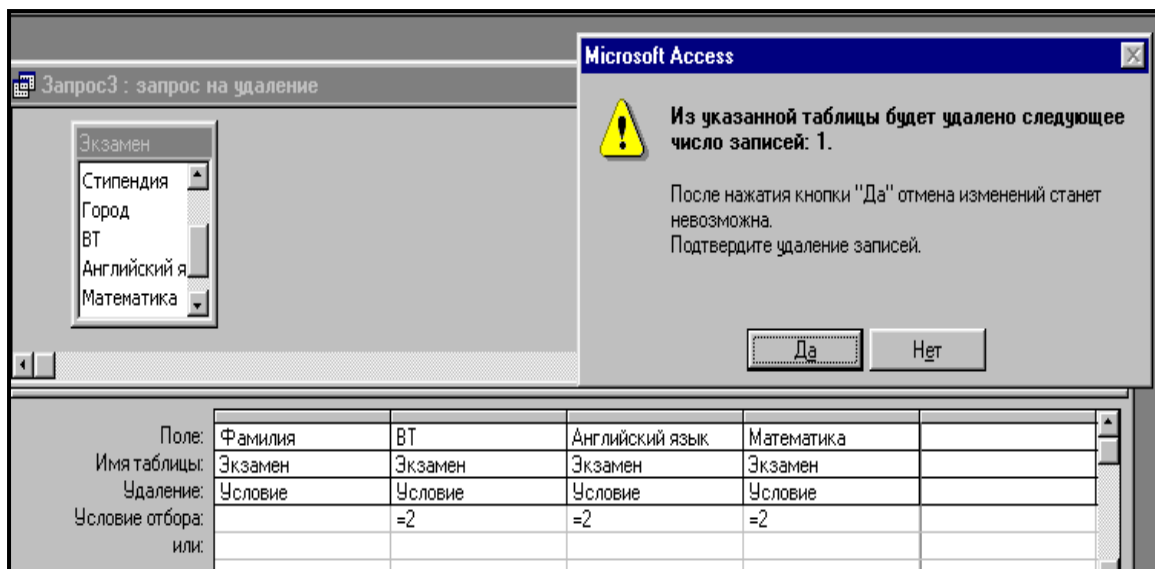


Рис. 4.22. Вікно конструктора запиту на видалення

Запит на відновлення записів. Вносить загальні зміни в групу записів, які відбираються за допомогою умов, в одну або кілька таблиць. Запит цього типу дозволяє змінювати дані в існуючих таблицях.

Допустимо, спочатку студентам була призначена стипендія в розмірі 34 грн. (рис. 4.23). Потім стипендію вирішили збільшити на 50грн. Це можна зробити за допомогою запиту на відновлення.

У нижній частині бланка з'явиться новий рядок «Обновления», у якій варто ввести вираження для розрахунку (рис. 4.23).

| Фамилия | Дата рожд | Стипендия | Город | ВТ | Английски | Математика |
|-----------|-----------|-----------|----------|----|-----------|------------|
| Котов | 03.12.78 | 34р. | Донецк | 3 | 4 | 2 |
| Кошкина | 01.01.79 | 34р. | Артемовс | 2 | 3 | 3 |
| Кисин | 02.09.78 | 34р. | Донецк | 5 | 4 | 5 |
| Кисочкина | 03.11.80 | 34р. | Харцызск | 5 | 5 | 5 |
| Кошак | 07.03.80 | 34р. | Донецк | 2 | 2 | 2 |
| Брысин | 04.05.75 | 34р. | Луганск | 2 | 4 | 5 |
| Мышеловки | 30.04.79 | 34р. | Донецк | 3 | 3 | 5 |

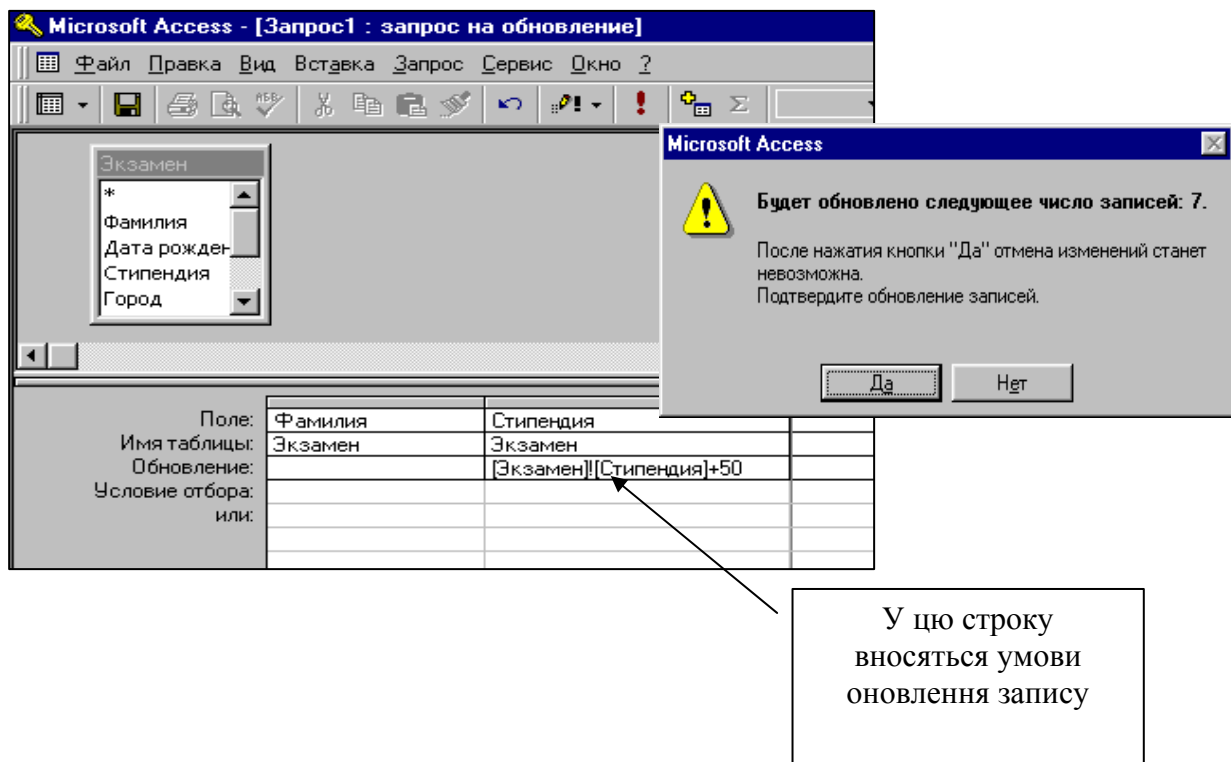


Рис. 4.23. Вікно конструктора запиту на відновлення

При запуску запиту на виконання з'явиться повідомлення, яке варто підтвердити. Після того, як Ви підтвердите відновлення записів, зовнішній вигляд екрана не зміниться, всі зміни відбуваються тільки з таблицею. Щоб переглянути їх, відкрийте таблицю.

5. СТВОРЕННЯ ЗВІТУ ЯК ОБ'ЄКТА БАЗИ ДАНИХ

Звіт - це форматоване подання даних, що виводиться на екран, до друку або файл. Вони дозволяють витягти з бази потрібні відомості й представити їх у вигляді, зручному для сприйняття, а також надають широкі можливості для узагальнення й аналізу даних.

При печатці таблиць і запитів інформація видається практично в тім виді, у якому зберігається. Часто виникає необхідність представити дані у вигляді звітів, які мають традиційний вигляд і легко читаються. Докладний звіт включає всю інформацію з таблиці або запиту, але містить заголовки й розбитий на сторінки із вказівкою верхніх і нижніх колонтитулів.

Структура звіту в режимі Конструктора.

Microsoft Access відображає у звіті дані із запиту або таблиці, додаючи до них текстові елементи, які спрощують його сприйняття.

До числа таких елементів ставляться:

Заголовок. Цей розділ друкується тільки у верхній частині першої сторінки звіту. Використається для висновку даних, таких як текст заголовка звіту, дата або частина, що констатує, тексту документа, які варто надрукувати один раз на початку звіту. Для додавання або видалення області заголовка звіту необхідно вибрати в меню Вид команду Заголовок/примітка звіту.

Верхній колонтитул. Використається для висновку даних, таких як заголовки стовпців, дати або номери сторінок, що друкуються зверху на кожній сторінці звіту. Для додавання або видалення верхнього колонтитула необхідно вибрати в меню Вид команду Колонтитули. Microsoft Access додає верхній і нижній колонтитули одночасно. Щоб сховати один з колонтитулів, потрібно задати для його властивості Висота значення 0.

Область даних, розташована між верхнім і нижнім колонтитулами сторінки. Містить основний текст звіту. У цьому розділі з'являються дані, що роздруковують для кожного з тих записів у таблиці або запиті, на яких заснований звіт. Для розміщення в області даних елементів керування використають список полів і панель елементів. Щоб сховати область даних, потрібно задати для властивості роздязнула Висота значення 0.

Нижній колонтитул. Цей розділ з'являється в нижній частині кожної сторінки. Використається для висновку даних, таких як підсумкові значення, дати або номери сторінки, що друкуються знизу на кожній сторінці звіту.

Примітка. Використається для висновку даних, таких як текст висновку, загальні підсумкові значення або підпис, які варто надрукувати один раз наприкінці звіту. Незважаючи на те, що в режимі Конструктора розділ "Примітка" звіту перебуває внизу звіту, він друкується над нижнім колонтитулом сторінки на останній сторінці звіту. Для додавання або видалення області приміток звіту необхідно вибрати в меню Вид команду Заголовок/примітка звіту. Microsoft Access одночасно додає й видаляє області заголовка й приміток звіту.

Способи створення звіту.

В Microsoft Access можна створювати звіти різними способами (рис. 5.1):

- ✓ Конструктор
- ✓ Майстер звітів
- ✓ Автозвіт: у стовпець
- ✓ Автозвіт: стрічковий
- ✓ Майстер діаграм
- ✓ Поштові наклейки

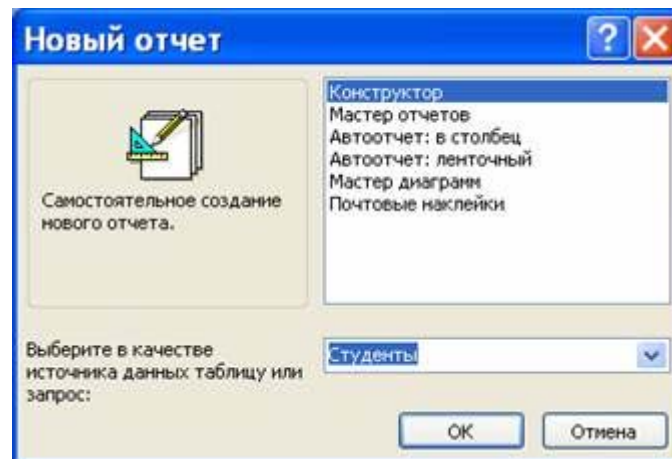


Рис. 5.1. Вікно початку створення звіту

Майстер дозволяє створювати звіти з угрупованням записів і являє собою найпростіший спосіб створення звітів. Він поміщає обрані поля у звіт і пропонує шість стилів його оформлення. Після завершення роботи Майстри отриманий звіт можна доробити в режимі Конструктора. Скориставшись функцією Автозвіт, можна швидко створювати звіти, а потім вносити в них деякі зміни.

Для створення Автозвіта необхідно виконати наступні дії:

- У вікні бази даних клацнути на вкладці Звіти й потім клацнути на кнопці Створити. З'явиться діалогове вікно Новий звіт.
- Виділити в списку пункт Автозвіт: у стовпець або Автозвіт: стрічковий.
- У поле джерела даних клацнути на стрілці й вибрати як джерело даних таблицю або запит.
- Клацнути на кнопці ОК.

Майстер Автозвіта створює Автозвіт у стовпець або стрічковий (на вибір користувача), і відкриває його в режимі Попереднього перегляду, що дозволяє побачити, як буде виглядати звіт у роздрукованому виді.

У меню Файл клацнути на команді Зберегти. У вікні Збереження в поле Ім'я звіту вказати назва звіту й клацнути на кнопці ОК.

Зміна масштабу відображення звіту

Для зміни масштабу відображення користуються покажчиком - лупою. Щоб побачити всю сторінку цілком, необхідно клацнути в будь-якій місці звіту. На екрані відобразиться сторінка звіту в зменшеному масштабі.

Знову клацнути на звіті, щоб повернутися до збільшеного масштабу відображення. У збільшеному режимі подання звіту, крапка, на якій ви клацнули, виявиться в центрі екрана. Для пролистання сторінок звіту користуються кнопками переходу вниз вікна.

Печатка звіту.

Для печатки звіту необхідно виконати наступне:

- У меню Файл клацнути на команді Печатка.
- В області Друкувати клацнути на варіанті Сторінки.
- Щоб надрукувати тільки першу сторінку звіту, уведіть 1 у поле "з" і 1 у поле "по".
- Клацнути на кнопці ОК.

Перш ніж друкувати звіт, доцільно переглянути його в режимі Попереднього перегляду, для переходу до якого в меню Вид потрібно вибрати Попередній перегляд.

Якщо при печатці наприкінці звіту з'являється порожня сторінка, переконаєтеся, що параметр Висота для приміток звіту має значення 0. Якщо при печатці порожні проміжні сторінки звіту, переконаєтеся, що сума значень ширини форми або звіту й ширини лівого й правого полів не перевищує ширину аркуша паперу, зазначений у діалоговому вікні Параметри сторінки (меню Файл).

При розробці макетів звіту керуйтеся наступною формулою: ширина звіту + ліве поле + праве поле \leq ширина паперу.

Для того щоб підігнати розмір звіту, необхідно використати наступні прийоми:

- змінити значення ширини звіту;
- зменшити ширину полів або змінити орієнтацію сторінки.

Створення звіту

1. Запустите програму Microsoft Access. Відкрийте БД (наприклад, навчальну базу даних «Деканат»).

2. Створіть Автозвіт: стрічковий, використовуючи як джерело даних таблицю (наприклад, Студенти). Звіт відкривається в режимі Попереднього перегляду, що дозволяє побачити, як буде виглядати звіт у роздрукованому виді (рис. 5.2).

3. Перейдіть у режим Конструктора й виконаєте редагування й форматування звіту. Для переходу з режиму попереднього перегляду в режим конструктора необхідно клацнути команду Закрити на панелі інструментів вікна додатка Access. На екрані з'явиться звіт у режимі Конструктора (рис. 5.3).

Редагування:

1) видалите поля код студента у верхньому колонтитулі й області даних;

2) перемістите вліво всі поля у верхньому колонтитулі й області даних.

3) Змініте напис у заголовку сторінки

У розділі Заголовок звіту виділити напис Студенти.

Студенты

| КодСтудента | КодГруппы | Фамилия | Имя | Отчество | Пол | Дата рождения | Место рождения |
|-------------|-----------|-------------|-----------|---------------|-----|---------------|--|
| 10 | Б0-16a | Григорен | Сергей | Петрович | м | 12.01.1989 | Харьков |
| 11 | Б0-16a | Краснова | Мария | Васильевна | ж | 25.05.1989 | г. Харьков |
| 12 | Б0-16b | Ильин | Илья | Ильич | м | 13.03.1989 | с. Красная Волчанского р-на Харьковской обл. |
| 13 | Б0-20a | Павлова | Зоя | Владимирова | ж | 01.05.1989 | г. Днепродзержинск |
| 14 | Б0-20b | Воробейна | Степан | Иванович | ж | 29.06.1989 | г. Люботин Харьковской обл. |
| 15 | Б0-30a | Луговой | Иван | Федорович | м | 17.08.1989 | с. Красная Октябрьского р-на Донецкой обл. |
| 16 | Б0-30b | Синельников | Александр | Владимирович | м | 12.02.1989 | с. Петровская Богодуховского р-на Харьковской обл. |
| 17 | Б0-40a | Мальцова | Наталья | Варисовна | ж | 14.06.1989 | г. Полтава |
| 18 | Б0-50a | Сидорова | Виктория | Павловна | ж | 09.04.1989 | г. Харьков |
| 19 | Б0-40b | Сидорова | Віра | Александрівна | ж | 30.05.1989 | г. Киев |
| 20 | Б0-20b | Корень | Виталий | Глебович | м | 31.08.1989 | г. Запорожье |
| 21 | Б0-30a | Навродова | Гарриет | Гарриетовна | м | 07.07.1989 | г. Ужгород |
| 22 | Б0-20a | Степанова | Елена | Николаевна | ж | 10.10.1989 | с. Васильевка Харьковской р-на Харьковской области |
| 23 | Б0-40a | Прозорова | Наталья | Николаевна | ж | 22.11.1989 | г. Харьков |
| 24 | Б0-16b | Иванова | Иван | Иванович | м | 20.09.1989 | г. Киев |

25 октября 2007 г.

Страница 2 из 2

Рис. 5.2. Видяг звіту, створеного за допомогою Автозвіт

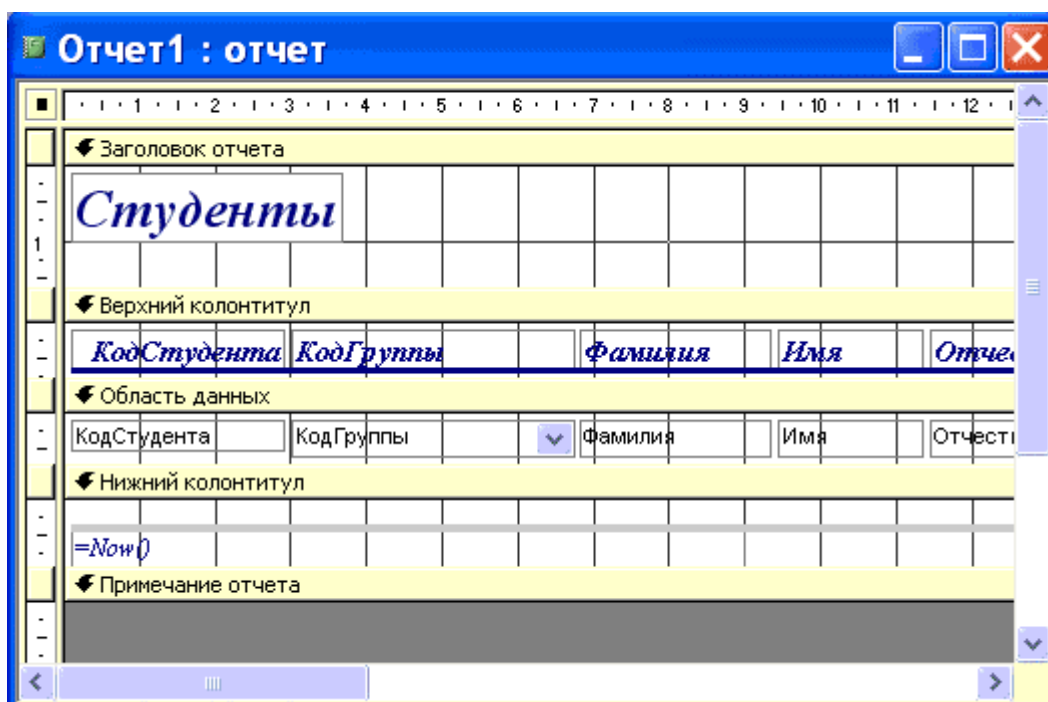


Рис. 5.3. Звіт у режимі Конструктора

Помістіть покажчик миші праворуч від слова Студенты, так щоб покажчик прийняв форму вертикальної риси (курсору уведення), і клацніть у цій позиції.

Уведіть НТУ «ХПІ» і натисніть Enter.

4) Перемістіть Напис. У Нижньому колонтитулі виділити поле =Now() і перетягнути його в Заголовок звіту під назву Студенты. Дата буде відображатися під заголовком.

5) На панелі інструментів Конструктор звітів клацнути на кнопці Попередній перегляд, щоб переглянути звіт

Форматування:

1) Виділіть заголовок Студенти НТУ «ХПІ»

2) Змініть гарнітуру, накреслення й кольори шрифту, а також кольори заливання тла.

3) На панелі інструментів Конструктор звітів клацнути на кнопці Попередній перегляд, щоб переглянути звіт (рис. 5.4).

Студенти НТУ "ХПІ"

26 октября 2007 г.

| КодГруппы | Фамилия | Имя | Отчество | Пол | Дата рождения | Место рождения |
|-----------|------------|-----------|---------------|-----|---------------|--|
| БФ-14а | Григорьев | Сергей | Петрович | ж | 12.01.1989 | г. Харьков |
| БФ-14а | Красавина | Ирина | Васильевна | ж | 25.05.1989 | г. Харьков |
| БФ-14б | Ильин | Илья | Ильич | ж | 13.03.1989 | с. Красное Волчанского р-на Харьковской обл. |
| БФ-24а | Петрова | Юли | Викторовна | ж | 01.05.1989 | г.Дзержинск |
| БФ-24б | Ворожова | Светла | Ивановна | ж | 29.04.1989 | г.Львов |
| БФ-34а | Лугинский | Иван | Федорович | ж | 17.08.1989 | с. Красное Славянского р-на Донецкой обл. |
| БФ-54б | Славкошина | Александр | Владимиров | ж | 12.02.1989 | с. Петровское Богородского р-на Харьковской обл. |
| БФ-44а | Маликова | Катерина | Борисовна | ж | 14.04.1989 | г. Николаев |
| БФ-54а | Сорокина | Валентина | Павловна | ж | 08.04.1989 | г. Харьков |
| БФ-44б | Сорокина | Вера | Александровна | ж | 30.05.1989 | г. Киев |
| БФ-24б | Корова | Виктория | Петровна | ж | 31.08.1989 | г. Харьков |
| БФ-34а | Кадарова | Георгий | Георгиевич | ж | 07.07.1989 | г. Ужгород |
| БФ-24а | Славина | Ирина | Марковна | ж | 10.10.1989 | с. Васильеве Харьковской р-на Харьковской обл. |
| БФ-44а | Приходко | Наталья | Ивановна | ж | 22.11.1989 | г. Харьков |
| БФ-14б | Иванов | Иван | Иванович | ж | 20.09.1989 | г. Киев |

Страница 1 из 1

Рис. 5.4. Звіт після редагування та форматування

Зміна стилю.

Для зміни стилю виконаєте наступне:

1) На панелі інструментів Конструктора звітів клацнути на кнопці Автоформат, відкриється діалогове вікно Автоформат.

2) У списку Стилів об'єкта "звіт - автоформат" клацнути на пункті Строгий і потім клацнути на кнопці ОК. Звіт буде відформатований у стилі Строгий.

3) Перемкнеться в режим Попередній перегляд. Звіт відобразиться в обраному вами стилі. Надалі всі звіти створені за допомогою функції Автозвіт будуть мати стиль Строгий, поки ви не задасте інший стиль у вікні Автоформат.

4) Зберегти й закрити звіт.

6. ЕЛЕМЕНТ КЕРУВАННЯ КНОПКА

Елемент керування Кнопка (Command Button) є дуже важливим елементом форми, оскільки саме із кнопками зв'язані різні дії, виконувані користувачем у додатку (збереження уведених даних, виклик іншої форми, висновок на печатку документа й т.д.). Звичайно кнопка вибирається, коли користувач клацає по ній мишею. Однак можна вибрати (нажати, тому що візуально це саме так і виглядає) кнопку, використовуючи тільки клавіатуру. Для цього потрібно помістити на неї фокус, тобто зробити активної, натискаючи клавішу <Tab>, а потім нажати клавіші <Пробіл> або <Enter>. Для цієї ж мети можна призначити кнопці спеціальну комбінацію клавіш.

Створити кнопку набагато зручніше за допомогою Майстра кнопок, оскільки він виконає за вас більшу роботу й навіть напише програму, що буде виконувати дії, для яких дана кнопка призначена. Наприклад, розглянемо процес створення кнопки, що буде відкривати нову форму:

Створіть у режимі Конструктора порожню форму.

Виберіть на панелі елементів елемент керування Кнопка (CommandButton) (кнопка Майстра (Control Wizards) на панелі елементів повинна бути натиснута).

У діалоговому вікні, що з'явилося, ви побачите два списки: лівий список містить категорії дій, а правий - самі дії. Виберіть у списку Категорії (Categories) значення Робота з формою (Form Operations) (рис. 6.1).

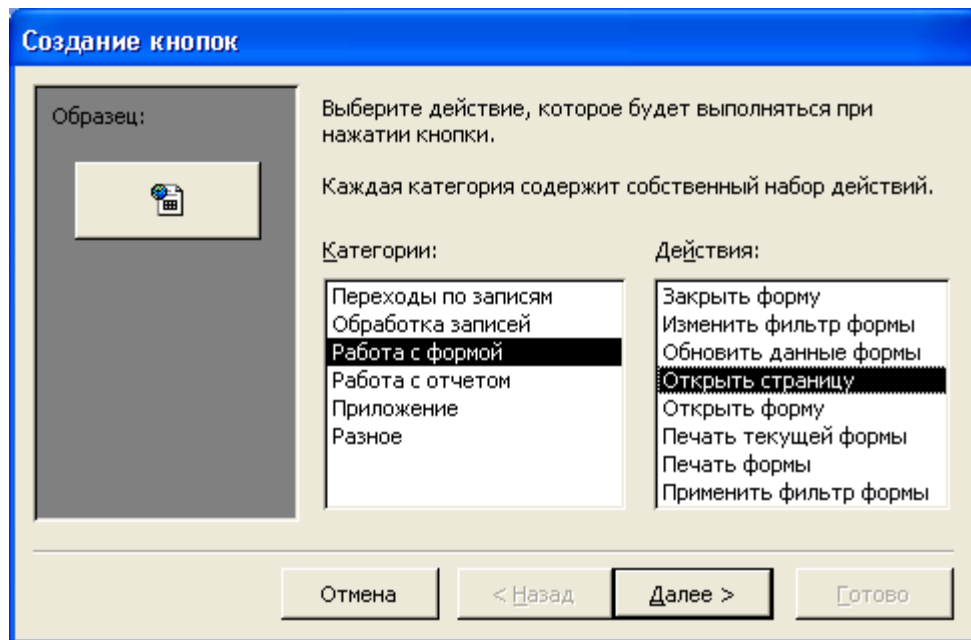


Рис. 6.1. Перше діалогове вікно Майстра кнопок

У поле Дії при цьому з'явиться список дій, що ставляться до цієї категорії. Виберіть значення Відкрити форму (Open Form) і натисніть кнопку Далі (Next).

У наступному вікні майстра потрібно вибрати форму, яку потрібно відкривати за допомогою кнопки. Виберіть одну з форм, наприклад "Замовлення" (Orders) (рис. 6.2).

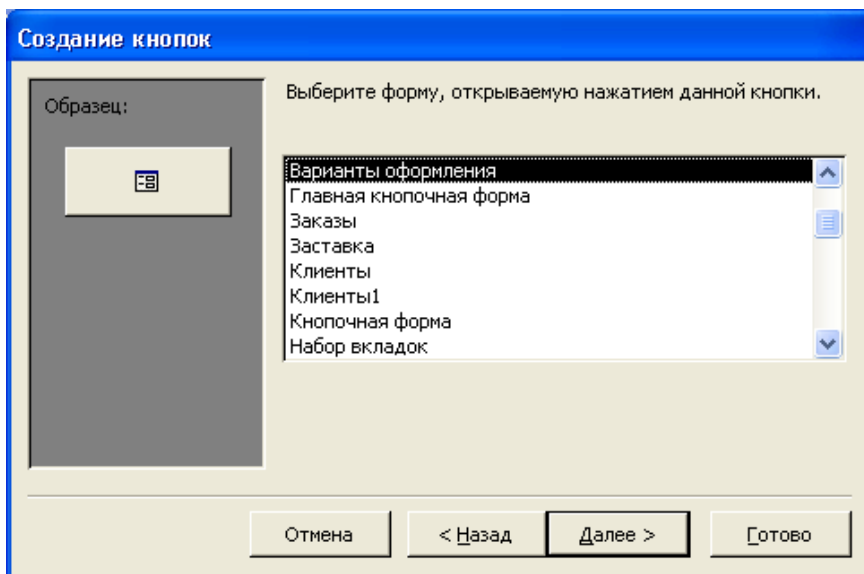


Рис. 6.2. Друге діалогове вікно Майстра кнопок

У третім діалоговому вікні Майстра кнопок потрібно вибрати, чи всі записи будуть відображатися у відкриваємій формі, або відібрати записи, пов'язані із записами в даній формі (рис. 6.3). Оскільки форма не містить поки жодного поля, ми не зможемо задати критерії відбору. Відповідно, варто вибрати другий перемикач: Відкрити форму й показати всі записи (Open the form and show all the records), після чого натиснути кнопку Далі (Next).

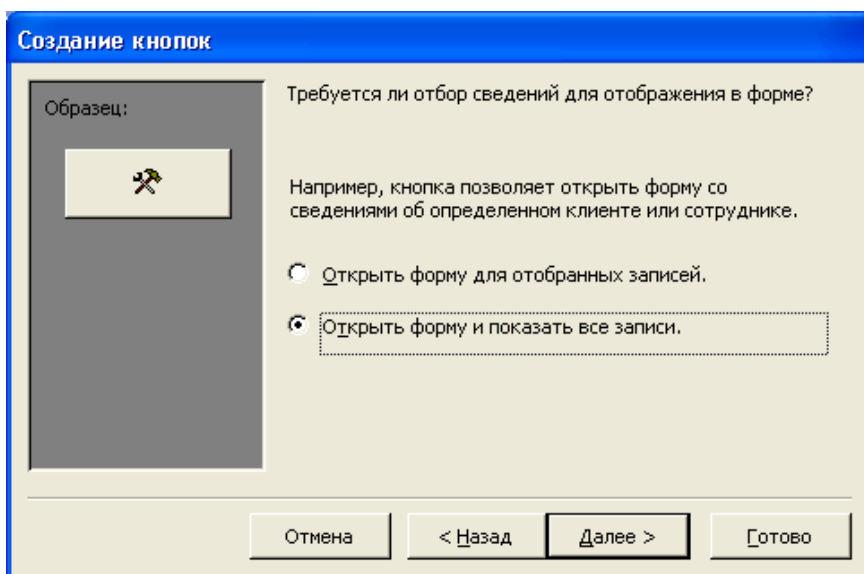


Рис. 6.3. Третє діалогове вікно Майстра кнопок

У наступному вікні пропонується вибрати малюнок, що ви хочете помістити на кнопку, або задати напис (рис. 6.4). Щоб побачити список всіх готових малюнків, необхідно встановити прапорець Показати всі малюнки (Show All Pictures). Щоб відобразити напис на кнопці, виберіть перемикач Текст (Text) і введіть текст напису, що, наприклад, збігає з ім'ям форми: "Замовлення". Натисніть кнопку Далі (Next).

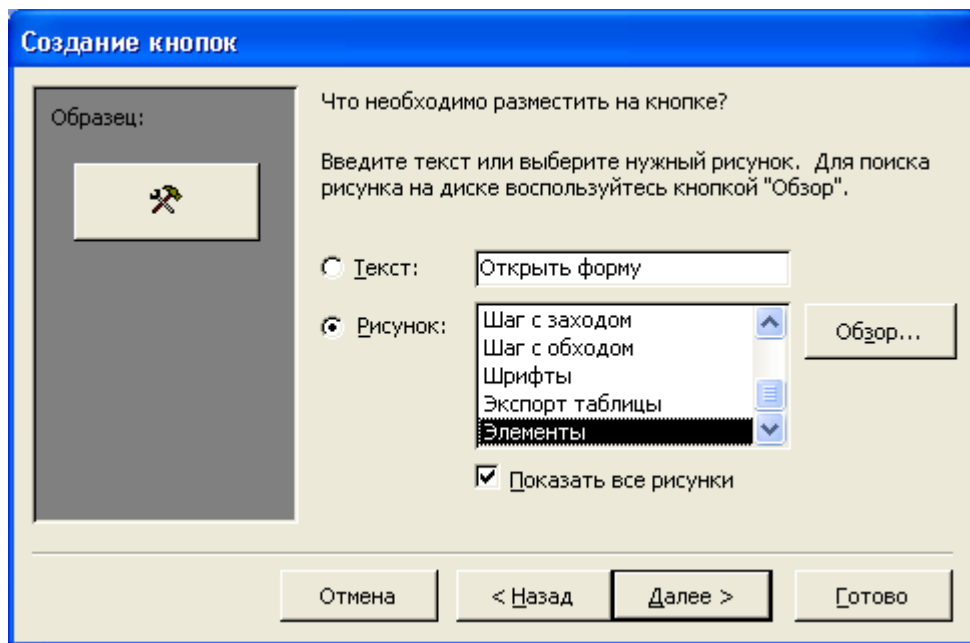


Рис.6.4. Четверте діалогове вікно Майстра кнопок

В останнім вікні Майстра кнопок потрібно ввести ім'я кнопки. Воно може не збігатися з написом на кнопці, але повинне бути коротким і зрозумілим. У цьому випадку можна залишити той же текст, що й у написі: Замовлення (рис. 6.5). Натисніть кнопку Готове (Finish).

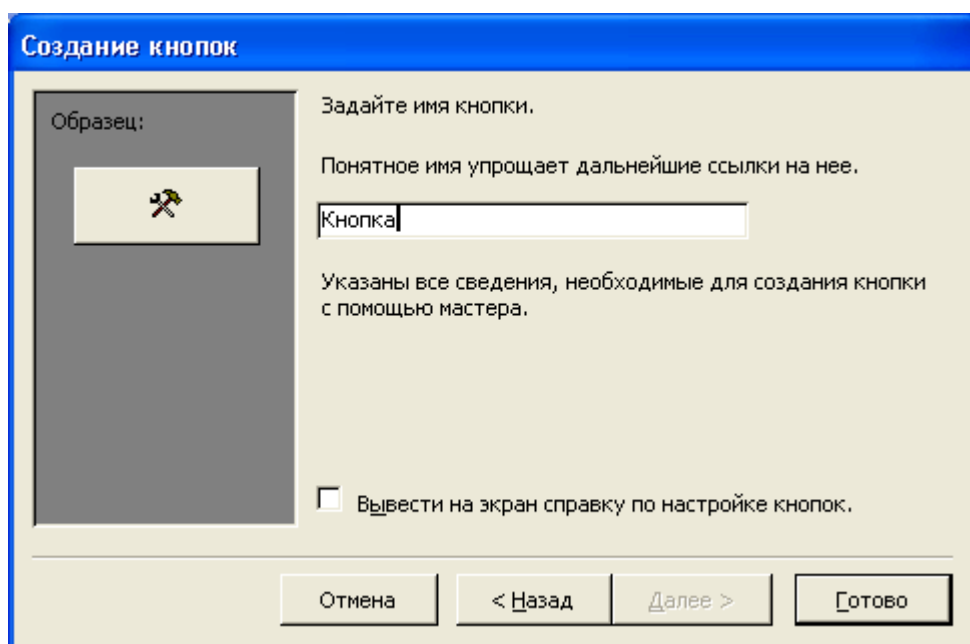


Рис. 6.5. П'яте діалогове вікно Майстра кнопок

На формі з'явилася кнопка. Якщо ви потім вирішите все-таки помістити на кнопку яку-небудь картинку, це можна зробити, установивши властивість Малюнок (Picture), для чого ви можете скористатися Побудовником, що пропонує або вибрати одну зі стандартних картинок, або використати графічний файл, наприклад створений у графічному редакторі Paint в Windows (рис. 6.6).

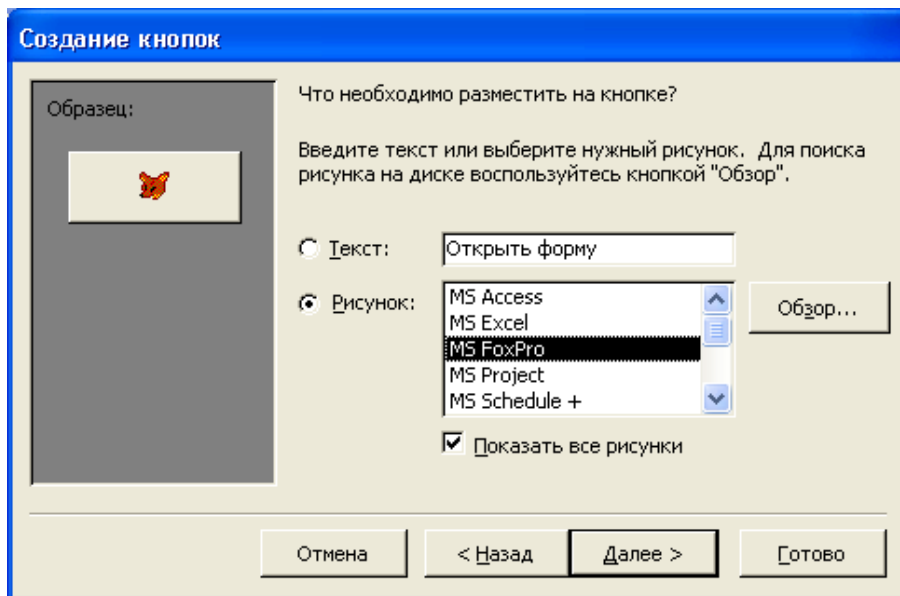


Рис. 6.6. Розміщення рисунка на кнопці

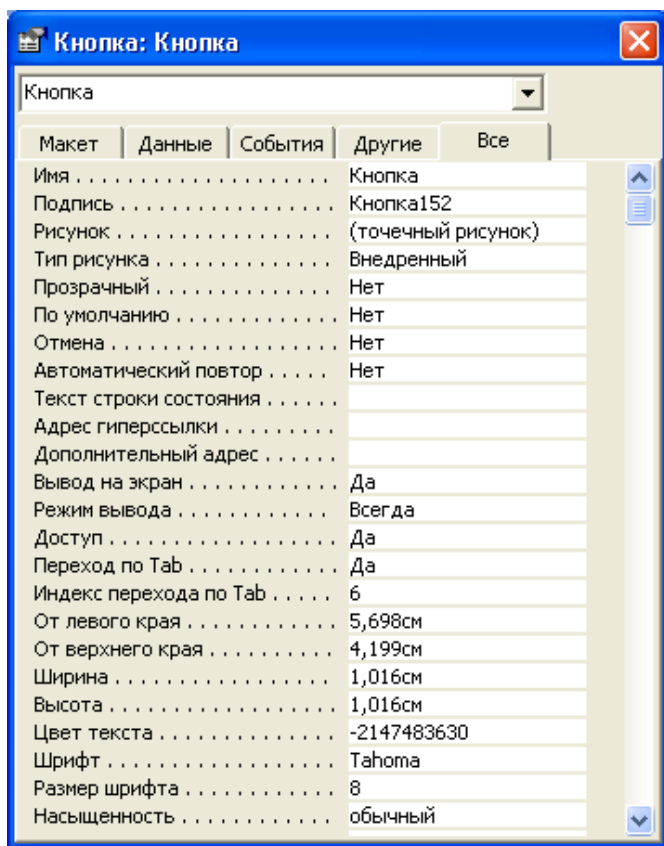


Рис. 6.7. Призначення кнопці комбінації клавiш

Можна призначити елементу керування Кнопка (Command Button) комбінацію клавiш. Це буває виправдано, наприклад, якщо користувач працює із клавіатурою і йому незручно для натискання кнопки перемикатися на мишу, а також, якщо у формі багато елементів і для активізації кнопки доводиться багато разів натискати клавiшу <Tab>. Щоб призначити кнопці комбінацію клавiш, у властивості Підпис (Caption), що визначає напис на кнопці, вибирають одну букву (вона необов'язково повинна бути першою) і додають перед нею знак амперсанта (&) (рис. 6.7). Обрана буква в написі на кнопці буде підкреслена. Тоді для того, щоб вибрати кнопку, досить просто натиснути комбінацію

клавiш <Alt>+<обрана буква>, у нашому прикладі - це <Alt>+<3>.

Для будь-якої форми можна створити дві спеціальні кнопки:

- Одна буде "натискатися" щораз, коли користувач натисне клавiшу <Enter> і незалежно від того, на якому елементі форми встановлений фокус. Така кнопка визначає дію за замовчуванням, і для неї властивості За замовчуванням (Default) привласнюють значення Так (Yes).

- Друга кнопка "натискається" щораз, коли користувач натисне клавiшу <Esc>, і знову незалежно від того, на якому елементі форми встановлений фокус.

Така кнопка відіграє роль скасування за замовчуванням. Для неї повинне бути привласнене значення Так (Yes) властивості Скасування (Cancel).

Приклад зображення вікна кнопкової форми на (рис. 6.8).

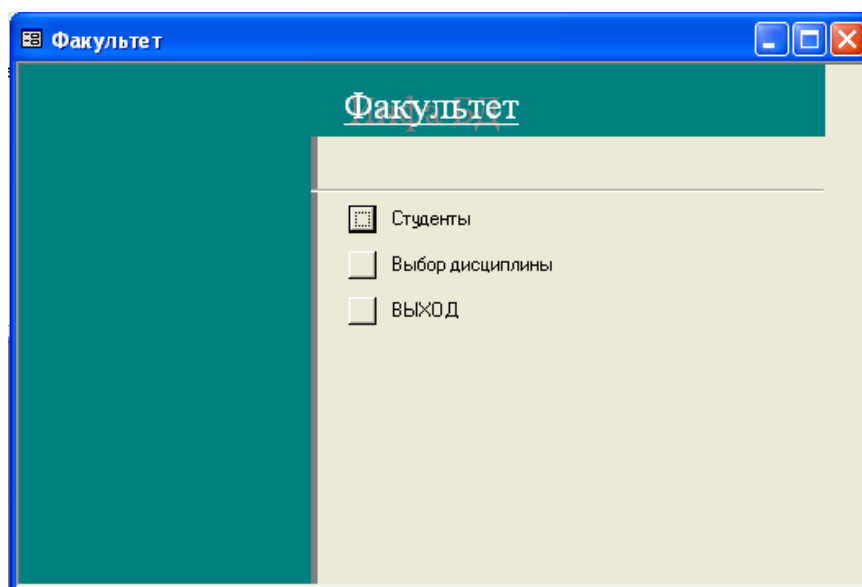


Рис. 6.8. Приклад кнопкової форми

Приклад зображення вікна після того, як нажата кнопка „Студент” на (рис. 6.9).

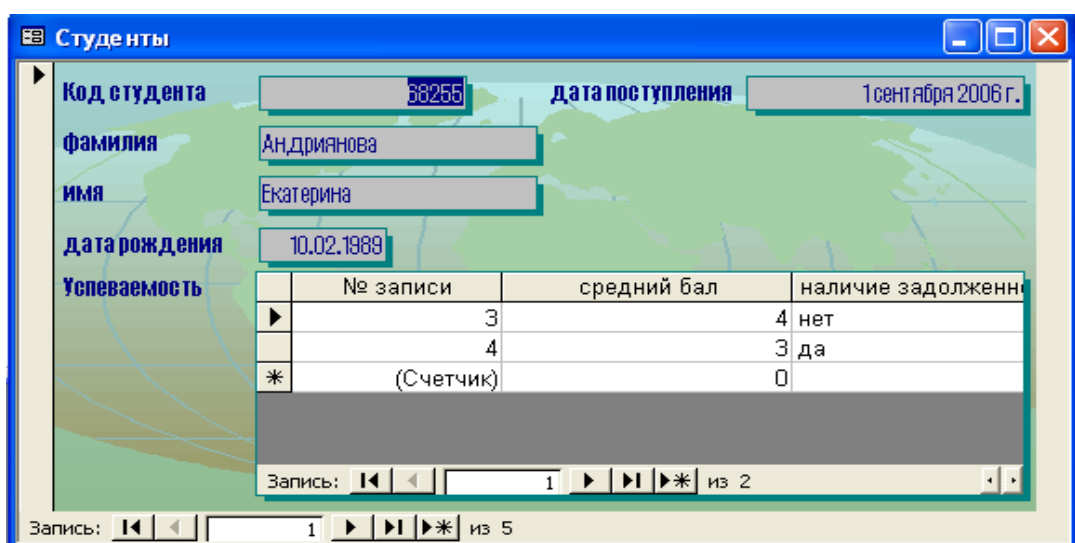


Рис. 6.9. Вікно кнопкової форми після включення кнопки „Студент”

Приклад зображення вікна після того, як нажата кнопка „Выбор дисциплины” на (рис. 6.10).

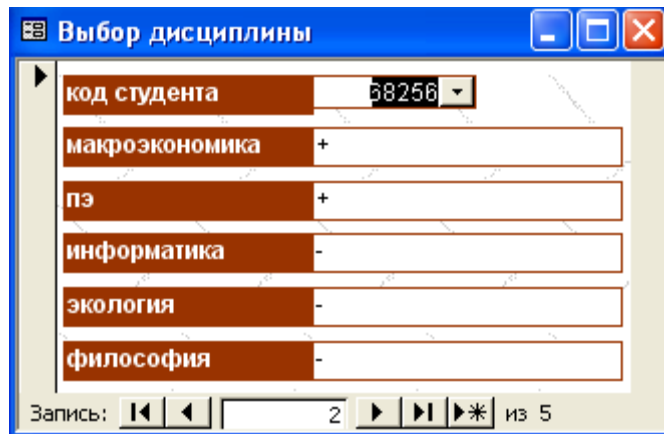


Рис. 6.10 Вікно кнопкової форми після включення кнопки „Выбор дисциплины”

7. МАКРОСИ

Основний набір засобів Microsoft Access, що ми розглядали в попередніх главах книги, орієнтований на користувачів, що не володіє мовами програмування. Для програмістів же до цих засобів додані макроси - невеликі програми мовою макрокоманд системи Access і модулі - процедури мовою Visual Basic for Application (VBA). З їхньою допомогою можна істотно розширити функціональні можливості створюваного вами додатка й настроїти його на потреби конкретних користувачів. У цій главі ми познайомимося з одним із цих засобів - макросами й одним з мов програмування Access - мовою макрокоманд

Використання макросів у базі даних Microsoft Access.

За допомогою макросів можна виконати практично всі дії над об'єктами Access з тих, які минулого описані в попередніх главах.

Макрос в Access являє собою структуру, що складається з однієї або декількох макрокоманд, які виконуються або послідовно, або в порядку, заданому певними умовами. Набір макрокоманд в Access дуже широкий, за допомогою макросів можна реалізувати багато чого з того, що дозволяють зробити процедури на VBA. Кожна макрокоманда має певне ім'я й, можливо, один або кілька аргументів, які задаються користувачем. Наприклад, при використанні макрокоманди Відкрити форму (OpenForm) як аргументу необхідно задати, принаймні, ім'я відкриваємої форми, що і виводить її на екран.

Макрокоманди Microsoft Access

| Категорія | Призначення | Макрокоманда |
|-----------------------------------|---|--|
| Робота з даними у формах і звітах | Відбір даних | ЗастосуватиФільтр (ApplyFilter) |
| | Переміщення за даними | Наступнийзапис (FindNext) Знайтизапис (FindRecord) Доелементу керування (GoToControl) Насторінку (GoToPage) Назапис (GoToRecord) |
| | Відновлення даних або екрана | Відновлення (Requery) Показативсізаписи (ShowAllRecords) |
| Виконання | Виконання команди | Виконати Команду (RunComraand) |
| | Виконання макросу, процедури або запиту | ЗапускМакроса (RunMacro) ЗалускПрограми (RunCode) ВідкритиЗапит |

| Категорія | Призначення | Макрокоманда |
|--------------------|--|---|
| | | (OpenQuery) ЗапускЗапроса SQL (RunSQL) |
| | Виконання іншого додатка | ЗапускДодатка (RunApp) |
| | Переривання виконання | СкасуватиПодію (CancelEvent) ЗупинитиУсіМакроси (StopAllMacros) Остановитьмакрос (StopMacro) |
| | Вихід з Microsoft Access | Вихід (Quit) |
| Імпорт/експорт | Передача об'єктів Microsoft Access в інші додатки | Вивести У Форматі (OutputTo) Відправитиоб'єкт (SendObject) |
| | Перетворення формату даних | Перетворити БазуДанных (Transfer Database) ПереносБазыДанныхSQL (TransferSQLDatabase) Перетворити ЭлектроннуТаблицю (Transfer Spreadsheet) ПеретворитиТекст (TransferText) |
| Робота з об'єктами | Копіювання, перейменування й збереження об'єкта | Копіюватиоб'єкт (CopyObject) Копіювати Файл Базиданных (CopyDatabase File) Перейменувати (Rename) Зберегти (Save) |
| Інші | Вивід на екран або приховання убудованої або спеціальної панелі інструментів Подача звукового сигналу | Панельинструментов (ShowToolbar) Сигнал (Beep) |

Як видно із цієї таблиці, дії, які можуть бути виконані за допомогою макросів, дуже різноманітні. Навіть якщо ви не будете використати макроси у своїх додатках, має сенс вивчити ці макрокоманди, оскільки їх можна використати й у процедурах VBA, як буде показано далі.

У порівнянні з попередньою версією Access, в Microsoft Access 2002 з'явилися три нові макрокоманди:

ПереносбазиданныхSQL (TransferSQLDatabase) - переносить базу даних Microsoft SQL Server 7.0 і вище на інший сервер;

Копіюватифайлбазиданих(CopyDatabaseFile) - виконує копіювання поточної бази даних Microsoft SQL Server 7.0 і вище, приєднаної до проекту Access;

Відкритифункцію (OpenFunction) - виконує функцію, певну користувачем, у проекті Access. Ця функція - являє собою запит, що, використовуючи вхідні параметри, повертає результати так само, як і збережена процедура.

Використання макросів виправдане тим, що їх легко створювати, і для цього не потрібно вивчати синтаксис мови програмування. Як буде показано в наступному розділі, для створення макросу потрібно знати лише основні прийоми роботи в Microsoft Access і Windows, такі як перетаскування об'єктів з вікна База даних (Database) у спеціальне вікно - Конструктор макросів (Macro Design), вибір дії зі списку й уведення виражень як аргументи макрокоманди. Тому, якщо дуже не хочеться вивчати синтаксис мови VBA або це представляється занадто важким, сміло застосовуйте макроси, і ви одержите досить функціональний додаток.

Як уже говорилося вище, основне призначення макросів - це створення зручного інтерфейсу додатка: щоб форми й звіти відкривалися при натисканні кнопок у формі або на панелі інструментів або ж звичним вибором команди меню; щоб при відкритті додатка користувач бачив на екрані не вікно База даних (Database), наповнене безліччю таблиць, запитів, форм і звітів, а якусь зрозумілу форму, за допомогою якої можна було б відразу робити бажані дії й т. буд.

На (рис. 7.1) проілюстровано вибір макрокоманди застосувати Фільтр (ApplyFilter). На панелі аргументів є два поля, які допоможуть задати ім'я фільтра й умова відбору записів. Для того щоб задати умова відбору записів, можна скористатися побудовником виразів.

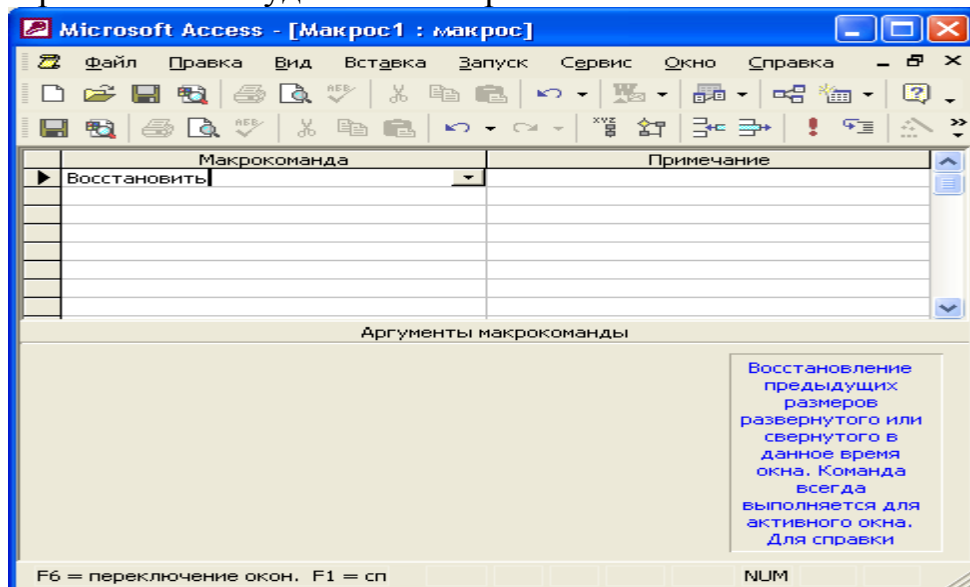


Рис. 7.1. Конструктор макросів і панель аргументів

Такий макрос можна використати у формі або звіті для того, щоб обмежити число відображуваних записів.

Щоб створити групу макросів:

- ✓ Відкрийте Конструктор макросів.
- ✓ Натисніть кнопку Імена макросів (Macro Name) на панелі інструментів. На панелі описів у вікні Конструктора з'явиться ще один стовпець - "Ім'я макросу" (Macro Name) (рис. 7.2).
- ✓ У цей стовпець уведіть ім'я першого макросу. Інші поля даного рядка краще залишити порожніми - це полегшить переміщення й копіювання макрокоманд.
- ✓ Починаючи з наступного рядка, уведіть всі макрокоманди макросу й відповідні аргументи для кожної макрокоманди.
- ✓ Пропустите один рядок.
- ✓ Повторіть кроки 3-5 для кожного макросу.

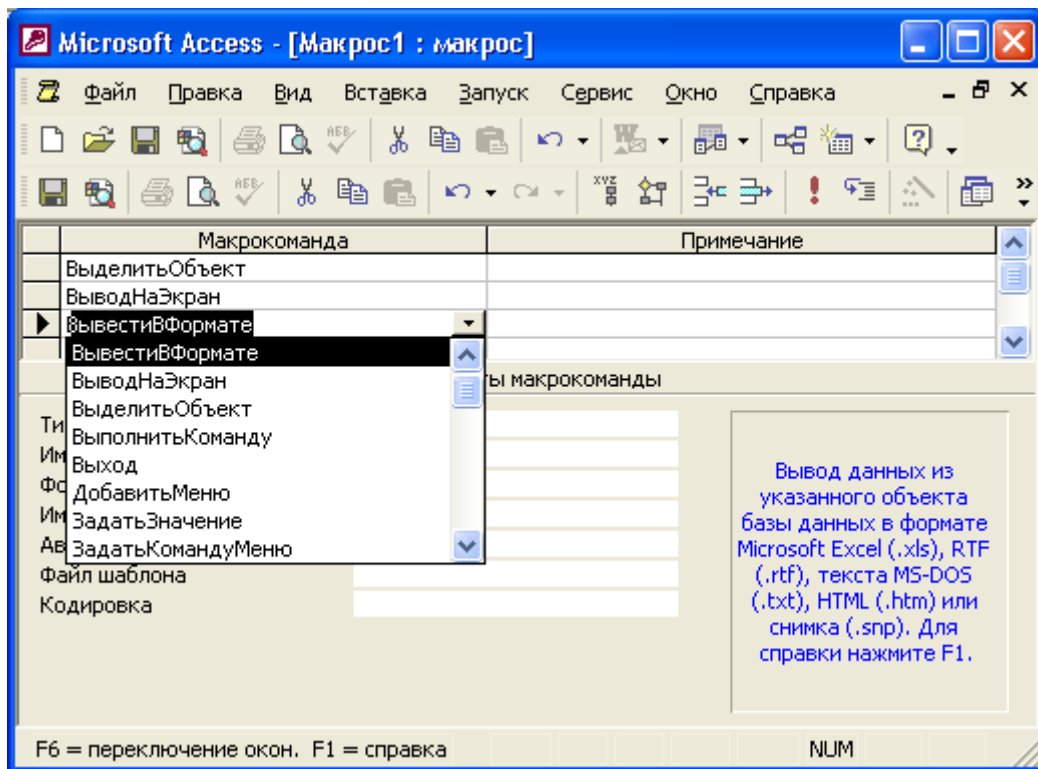


Рис.7.2. Створення групи макросів

На (рис. 7.2) у вікні Конструктора представлений макрос "Співробітники" (Employees) бази даних (Northwind). Для вказівки кінця макросу не потрібно окремої макрокоманди. Access продовжує виконання макросу, що входить у групу, поки не виявить рядок, що містить ім'я в стовпці "Ім'я макросу" (Macro Name), або поки не дійде до останнього рядка вікна макросів.

При обробці макросу порожні рядки ігноруються, тому їх зручно використати для поділу макросів у групі. Цей прийом, а також введення коментарів для складних макросів, що складаються з великої кількості макрокоманд, дозволяє поліпшити читаність макросів.

Для запуску одного з макросів групи використовується повне ім'я для посилання на макрос. Повне ім'я макросу формується в такий спосіб: `имягруппы.имямакроса`.

Способів запуску макросів дуже багато. Серед них є прямі способи, коли макрос запускається безпосередньо користувачем. Ці способи використовуються звичайно при тестуванні макросів. Коли макрос налагоджений, створюють спеціальну команду меню або кнопку на панелі інструментів або зв'язують макрос із якою-небудь подією у формі або звіті.

Універсальним способом запуску макросу є використання команди меню Сервіс, Макрос, Виконати макрос (Tools, Macro, Run Macro) (рис. 7.3).

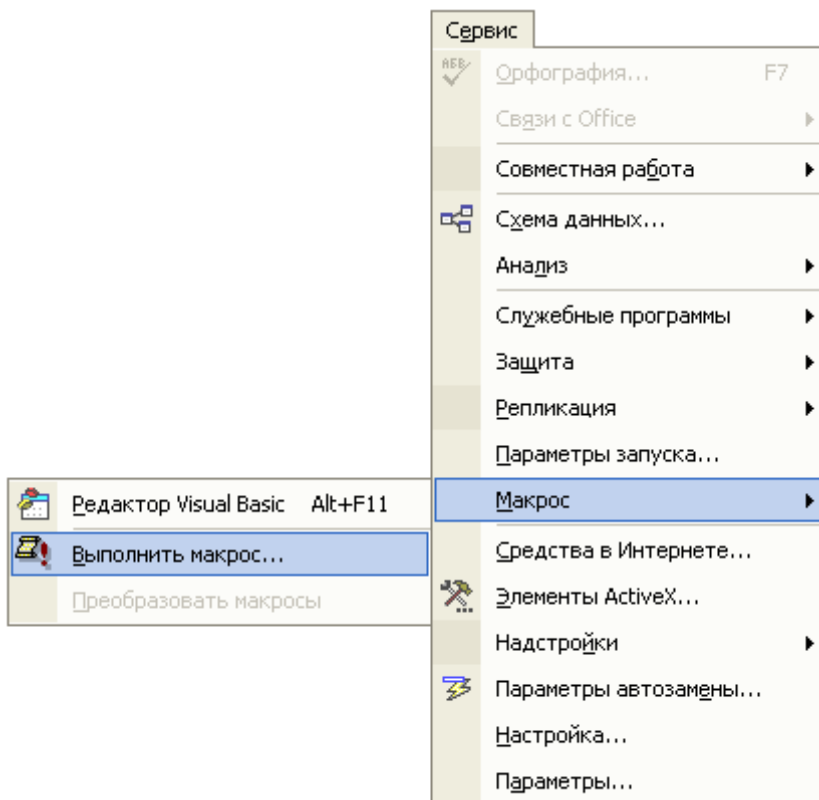


Рис. 7.3. Запуск макросу за допомогою команди меню

У діалоговому вікні, що з'явився при виборі цієї команди, потрібно вказати (увести або вибрати зі списку) повне ім'я макросу, тобто `имягруппы.имямакроса` (рис. 7.4). Цей спосіб дозволяє виконати макрос із будь-якого активного вікна, будь те вікно макросів, вікно бази даних або, наприклад, Конструктор форм.

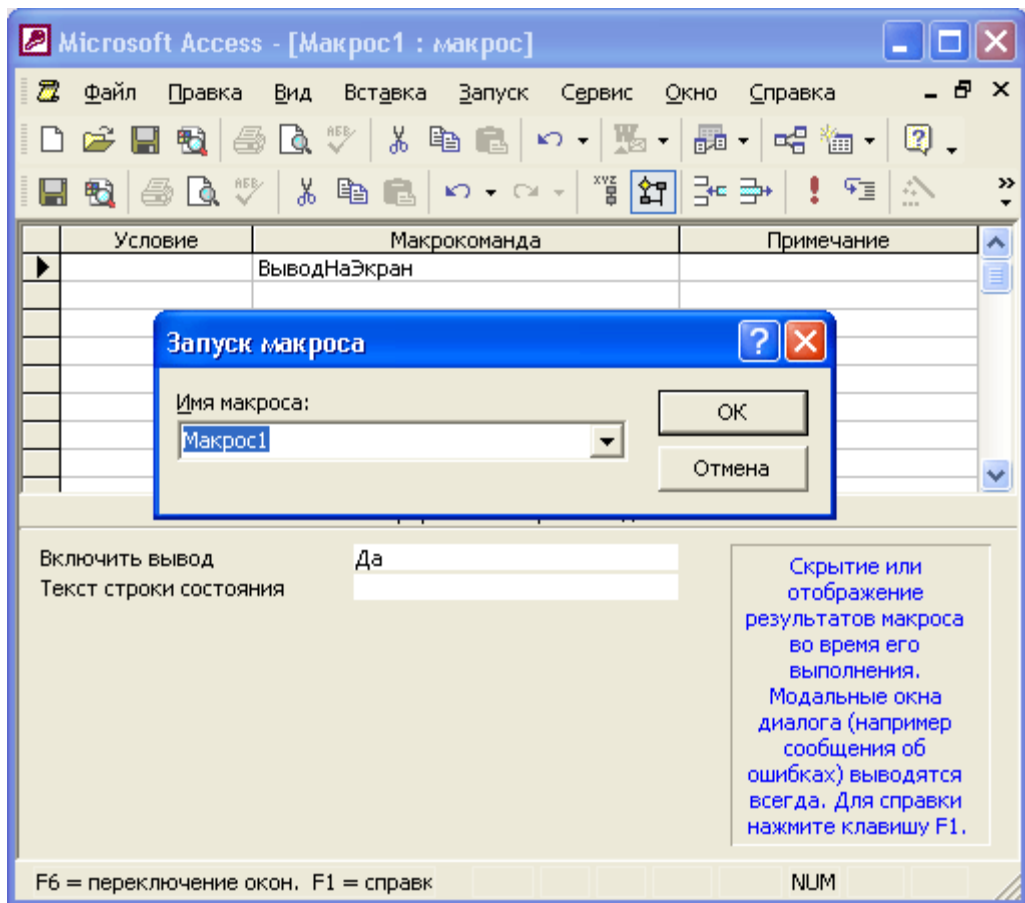


Рис.7.4. Диалогове вікно Запуск макросу

8. ПРИКЛАД РОЗРОБКИ ТА УПРАВЛІННЯ БАЗОЮ ДАНИХ

Вихідні дані:

- **Збирачі:** табельний номер, прізвище ім'я, по батькові, дата народження.
- **Виробі:** шифр виробу, тип виробу, складність зборки (середня, висока);
- **Журнал реєстрації:** кількість зібраних виробів по днях тижня понеділок, вівторок,..., субота;

Визначити:

- загальну кількість деталей, зібраних збирачами цеху;
- прізвище збирача, що зібрав найбільше число виробів, та день, коли він досяг найвищої продуктивності праці;

Створити таблицю, що містить наступну інформацію: прізвище збирача й загальна кількість деталей, зібрана їм за тиждень.

Виконання завдання:

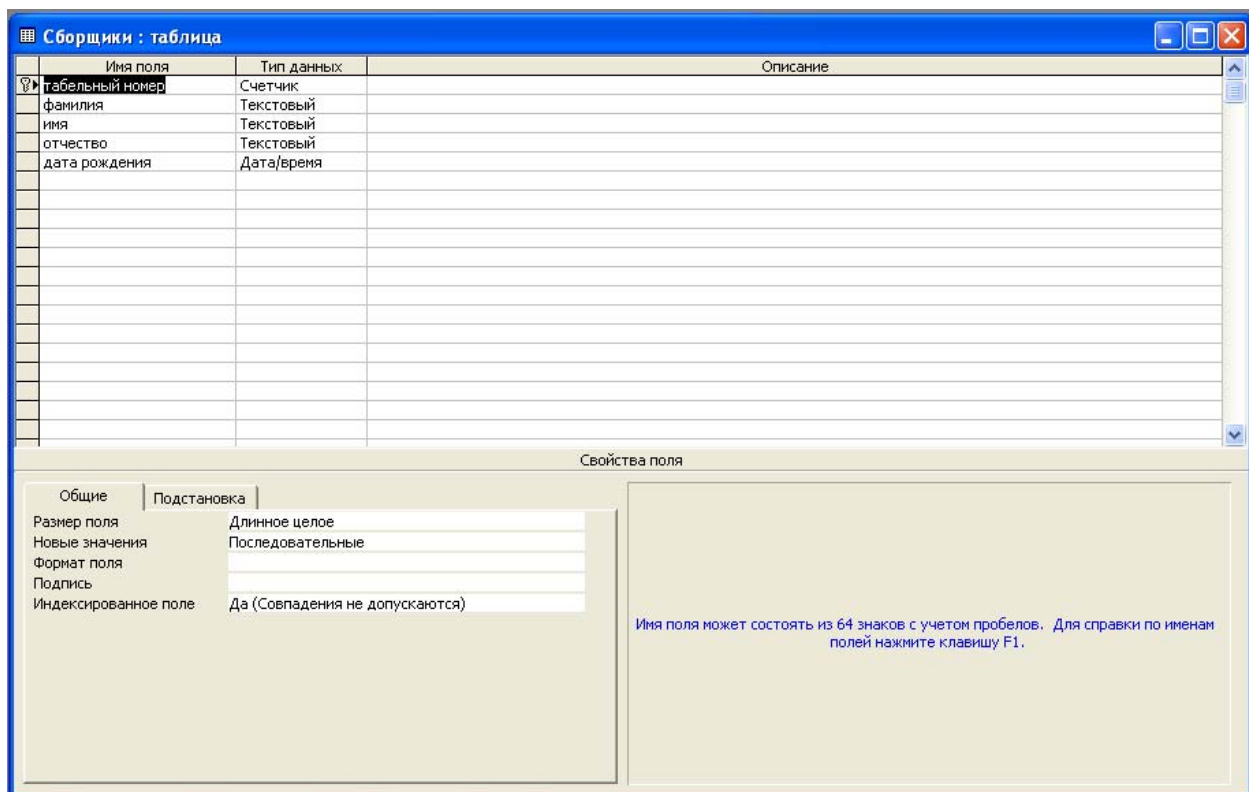


Рис.8.1. Створення у Конструкторі структури таблиці *Збирачі*

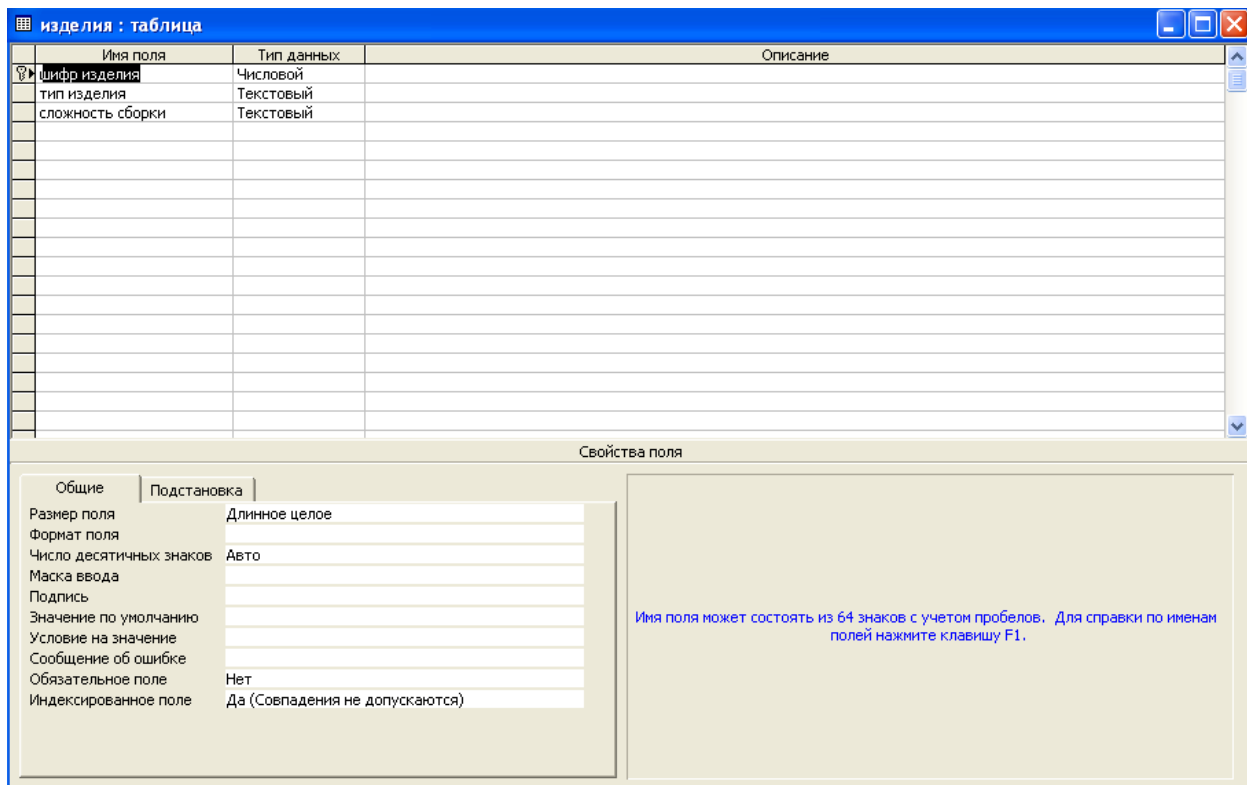


Рис.8.2. Створення у Конструкторі структури таблиці *Виробі*

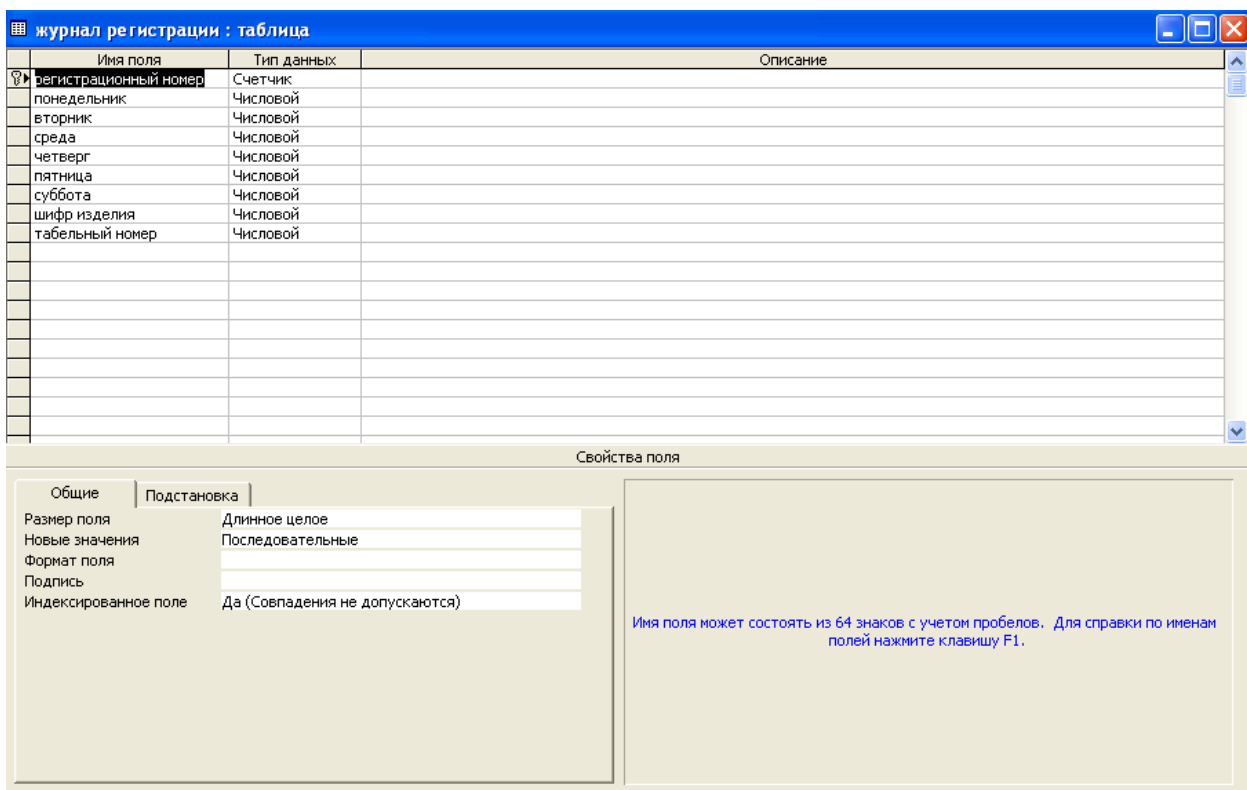


Рис. 8.3. Створення у Конструкторі структури таблиці *Журнал реєстрації*

| Сборщики : таблица | | | | | | |
|--------------------|---|-----------------|-----------|----------|---------------|---------------|
| | | табельный номер | фамилия | имя | отчество | дата рождения |
| ▶ | + | 1 | Доброва | Анна | Владимировна | 17.09.1989 |
| | + | 2 | Сидоренко | Светлана | Андреевна | 04.08.1989 |
| | + | 3 | Морозова | Олег | Юрьевич | 20.10.1982 |
| | + | 4 | Матвиенко | Андрей | Викторович | 05.01.1985 |
| | + | 5 | Иванов | Вадим | Владиславович | 14.05.1979 |
| * | | (Счетчик) | | | | |

Рис. 8.4. Ввід даних у таблицю *Збирачі*

| изделия : таблица | | | | |
|-------------------|---|--------------|-------------|------------------|
| | | шифр изделия | тип изделия | сложность сборки |
| ▶ | + | 1001 | холодильник | высокая |
| | + | 1002 | утюг | средняя |
| | + | 1003 | телевизор | высокая |
| | + | 1004 | компьютер | высокая |
| | + | 1005 | миксер | средняя |
| * | | | | |

Рис. 8.5. Ввід даних у таблицю *Вироби*

| журнал регистрации : таблица | | | | | | | | | | |
|------------------------------|---|---------------|-------------|---------|-------|---------|---------|---------|--------------|---------------|
| | | регистрационн | понедельник | вторник | среда | четверг | пятница | суббота | шифр изделия | табельный нот |
| ▶ | + | 1 | 2 | 4 | 5 | 3 | 2 | 1 | 1001 | 1 |
| | + | 2 | 4 | 5 | 3 | 6 | 4 | 2 | 1002 | 1 |
| | + | 3 | 1 | 3 | 2 | 4 | 2 | 1 | 1003 | 2 |
| | + | 4 | 2 | 1 | 3 | 1 | 3 | 2 | 1004 | 3 |
| | + | 5 | 5 | 4 | 6 | 7 | 3 | 5 | 1005 | 2 |
| | + | 6 | 4 | 3 | 5 | 2 | 4 | 6 | 1002 | 3 |
| | + | 7 | 8 | 7 | 10 | 5 | 8 | 5 | 1004 | 4 |
| | + | 8 | 9 | 7 | 4 | 6 | 8 | 5 | 1001 | 5 |
| | + | 9 | 4 | 2 | 4 | 3 | 1 | 3 | 1005 | 5 |
| | + | 10 | 6 | 3 | 2 | 5 | 4 | 2 | 1003 | 4 |
| * | | (Счетчик) | | | | | | | | |

Рис. 8.6. Ввід даних у таблицю *Журнал реєстрації*

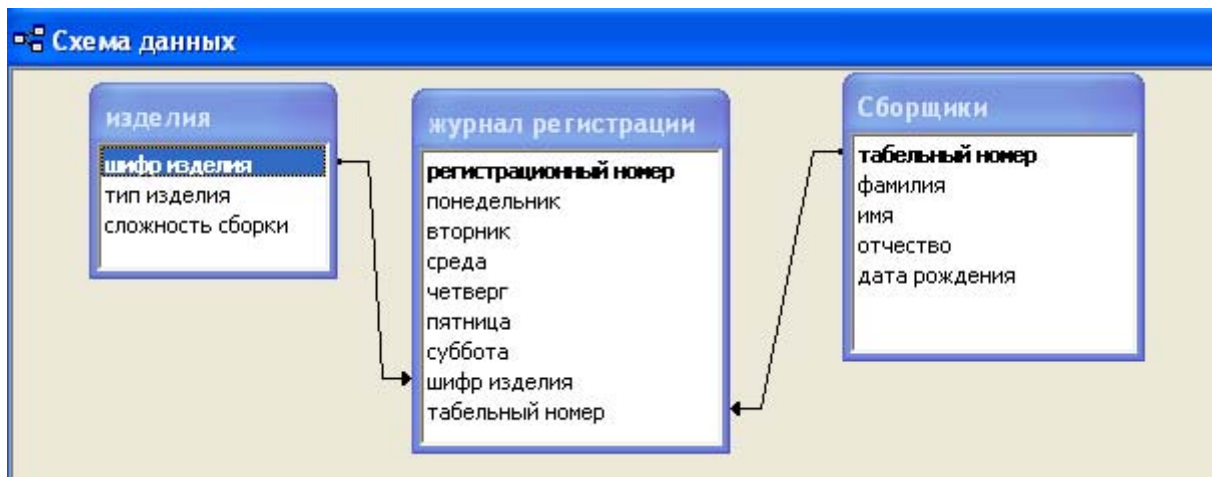


Рис. 8.7. Створення зв'язку між таблицями даних

| | |
|-----------------------|----------------|
| регистрационный номер | регистрационнь |
| понедельник | понедельник |
| вторник | вторник |
| среда | среда |
| четверг | четверг |
| пятница | пятница |
| суббота | суббота |
| шифр изделия | шифр изделия |

Рис. 8.8. Створення підлеглої форми таблиці *Журнал реєстрації*

| | |
|------------------|------------------|
| изделия | |
| шифр изделия | шифр изделия |
| тип изделия | тип изделия |
| сложность сборки | сложность сборки |

Рис. 8.9. Створення форми таблиці *Вироби*

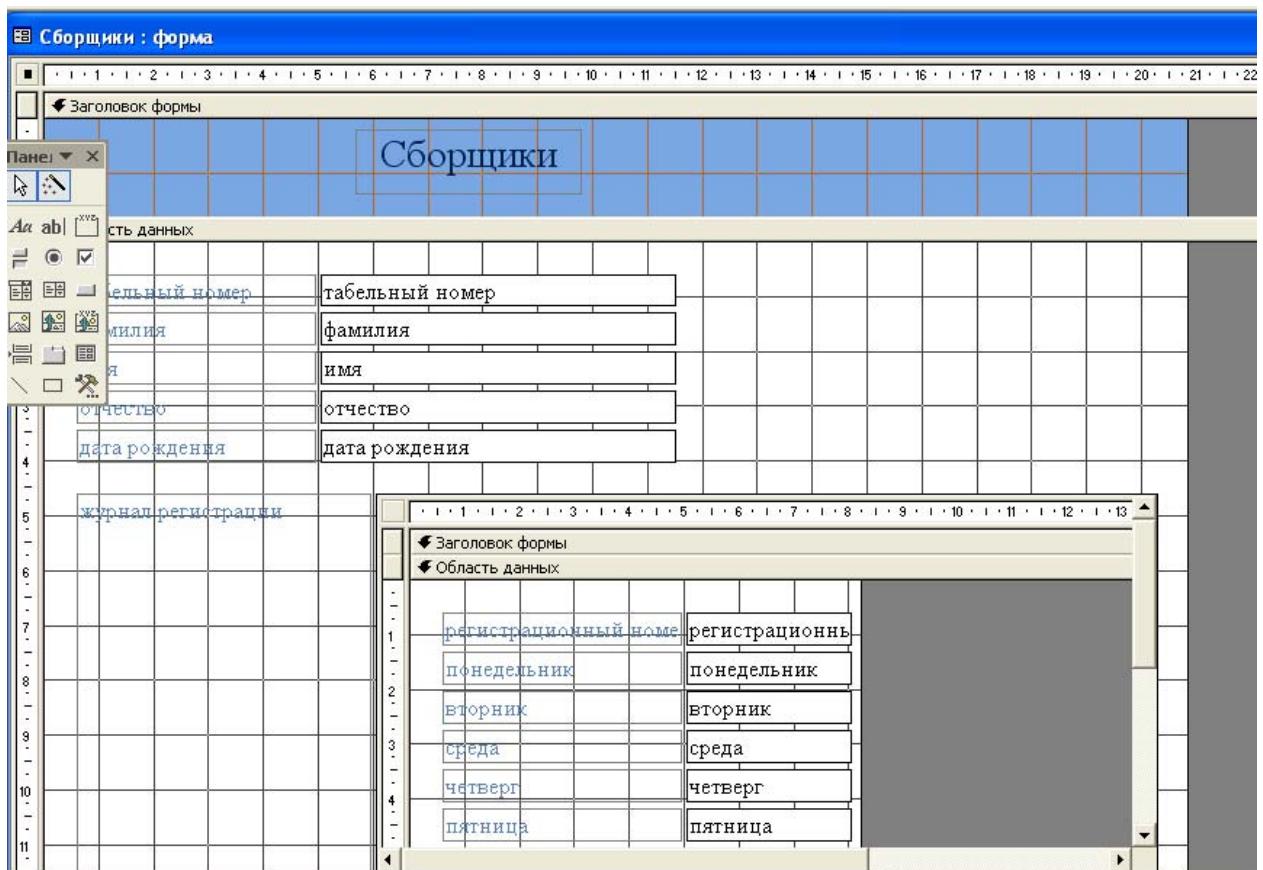


Рис. 8.10. Створення форми таблиці *Збирачі*

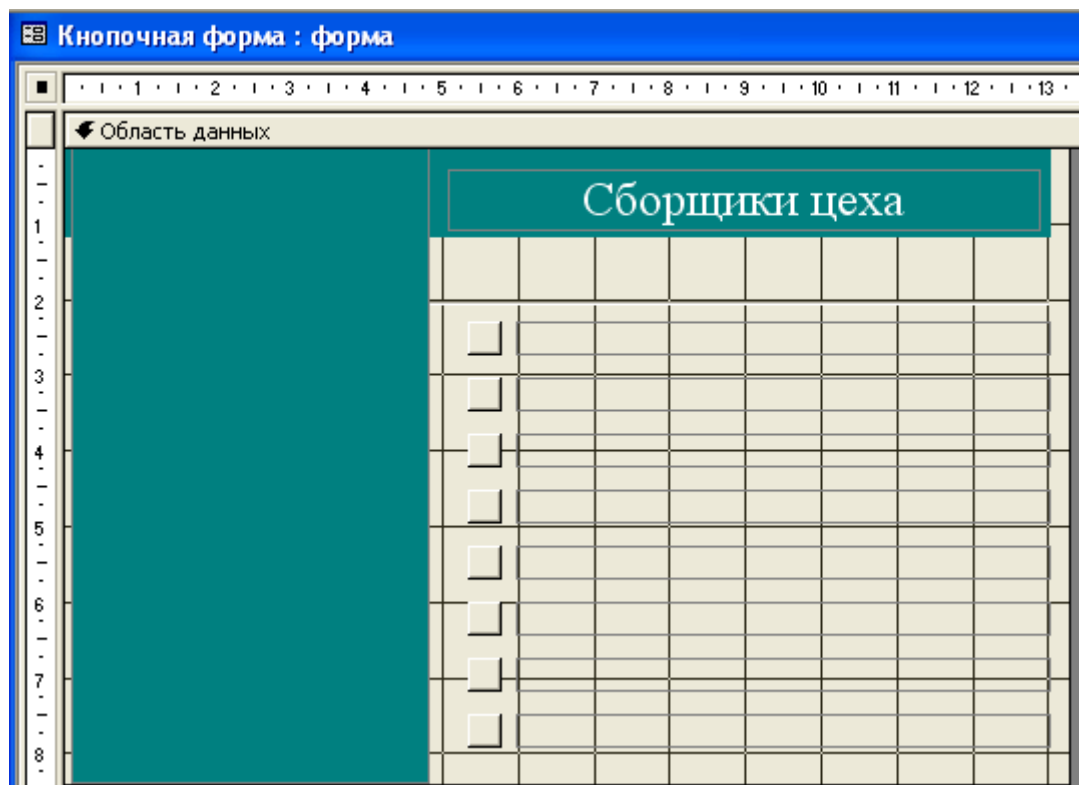


Рис. 8.11. Створення кнопкової форми

| журнал регистрации подчиненная форма | | | | | | | |
|--------------------------------------|---------|---------|-------|---------|---------|---------|--------------|
| регистрационны | понедел | вторник | среда | четверг | пятница | суббота | шифр изделия |
| 1 | 2 | 4 | 5 | 3 | 2 | 1 | 1001 |
| 2 | 4 | 5 | 3 | 6 | 4 | 2 | 1002 |
| 3 | 1 | 3 | 2 | 4 | 2 | 1 | 1003 |
| 4 | 2 | 1 | 3 | 1 | 3 | 2 | 1004 |
| 5 | 5 | 4 | 6 | 7 | 3 | 5 | 1005 |
| 6 | 4 | 3 | 5 | 2 | 4 | 6 | 1002 |
| 7 | 8 | 7 | 10 | 5 | 8 | 5 | 1004 |
| 8 | 9 | 7 | 4 | 6 | 8 | 5 | 1001 |
| 9 | 4 | 2 | 4 | 3 | 1 | 3 | 1005 |
| 10 | 6 | 3 | 2 | 5 | 4 | 2 | 1003 |
| * (Счетчик) | | | | | | | |

Рис. 8.12. Форма таблиці *Журнал реєстрації*

изделия1

изделия

шифр изделия: 1001

тип изделия: Холодильник

сложность сборки: Высокая

Запись: 1 из 5

Рис. 8.13. Форма таблиці *Виробі*

Сборщики

Сборщики

табельный номер: 1

фамилия: Дюброва

имя: Анна

отчество: Владимировна

дата рождения: 17.09.1989

журнал регистрации

| регистрационны | понедел | вторник | среда | четверг | пятница | суббота | шиф |
|----------------|---------|---------|-------|---------|---------|---------|------|
| 1 | 2 | 4 | 5 | 3 | 2 | 1 | 1001 |
| 2 | 4 | 5 | 3 | 6 | 4 | 2 | 1002 |
| * (Счетчик) | | | | | | | |

Запись: 1 из 2

Запись: 1 из 5

Рис. 8.14. Форма таблиці *Збирачі*

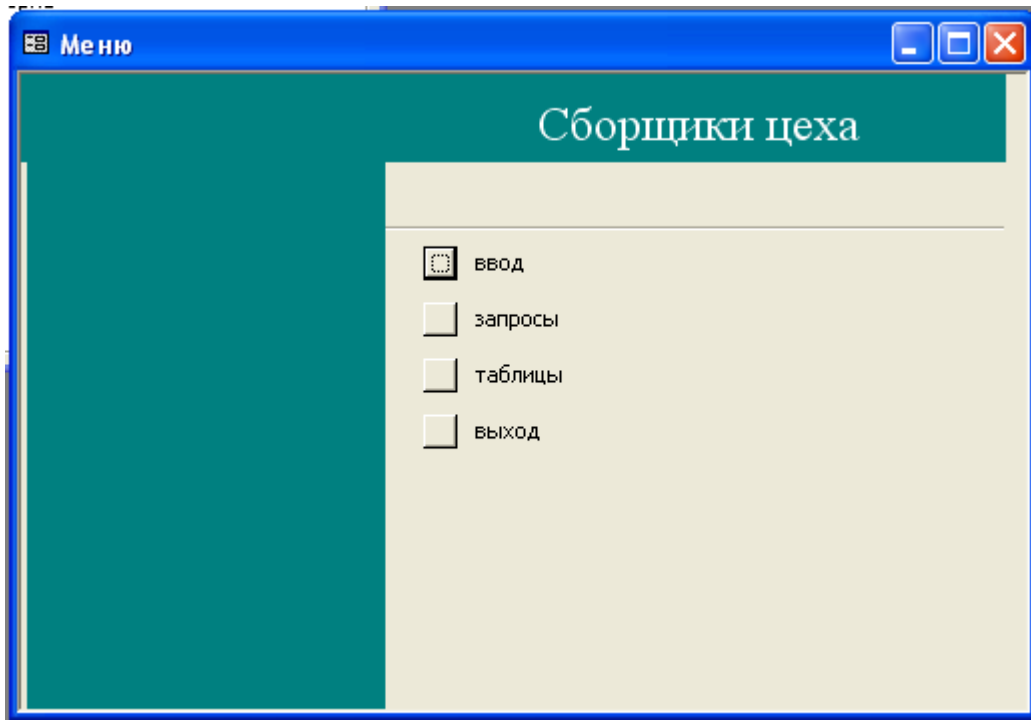


Рис. 8.15. Головна кнопочка форма

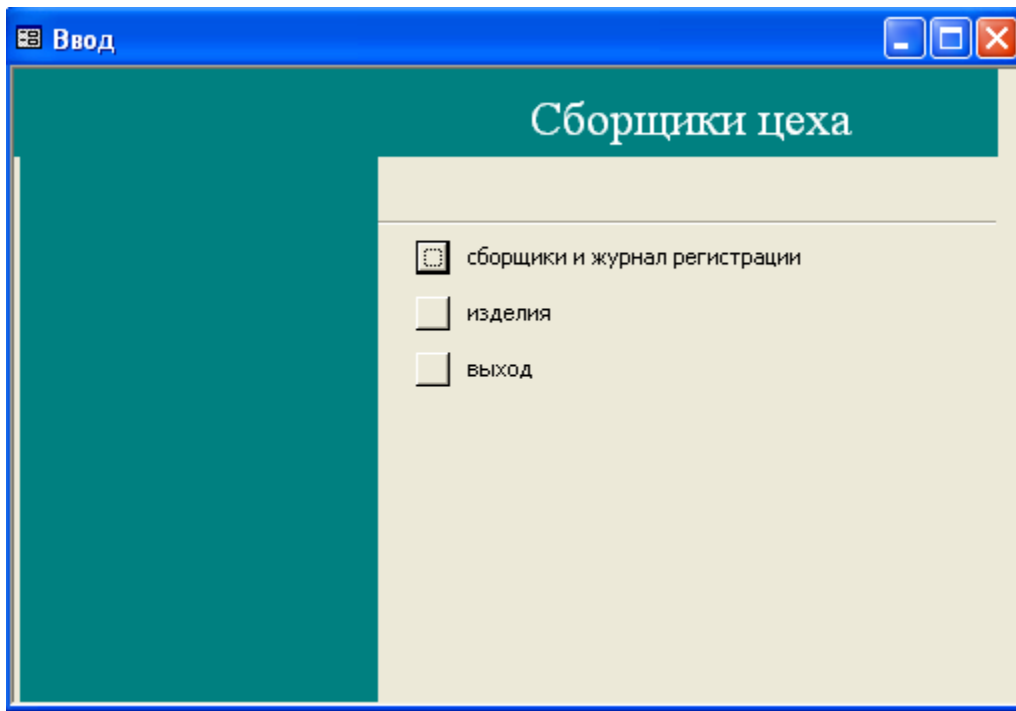


Рис. 8.16. Кнопочка форма "Ввід"

Сборщики

табельный номер:

фамилия:

имя:

отчество:

дата рождения:

журнал регистрации

| регистрационн | понедел | вторник | среда | четверг | пятница | суббота | шиф |
|---------------|---------|---------|-------|---------|---------|---------|------|
| ▶ 1 | 2 | 4 | 5 | 3 | 2 | 1 | 1001 |
| 2 | 4 | 5 | 3 | 6 | 4 | 2 | 1002 |
| * (Счетчик) | | | | | | | |

Запись: из 2

Запись: из 5

Рис. 8.17. Ввід даних о збирачах та журналу реєстрації

изделия1

изделия

шифр изделия:

тип изделия:

сложность сборки:

Запись: из 5

Рис. 8.18. Ввід даних о виробах

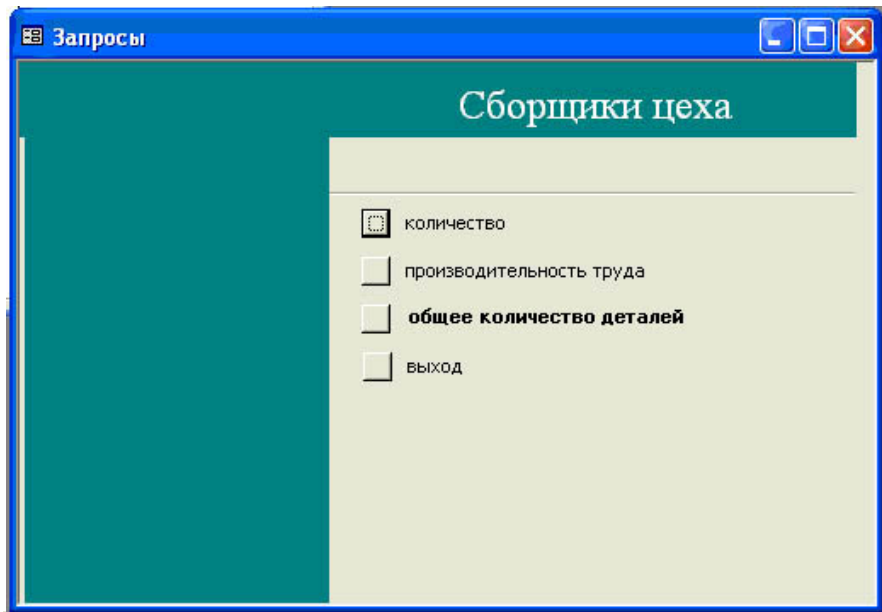


Рис. 8.19. Кнопковая форма "Запит"

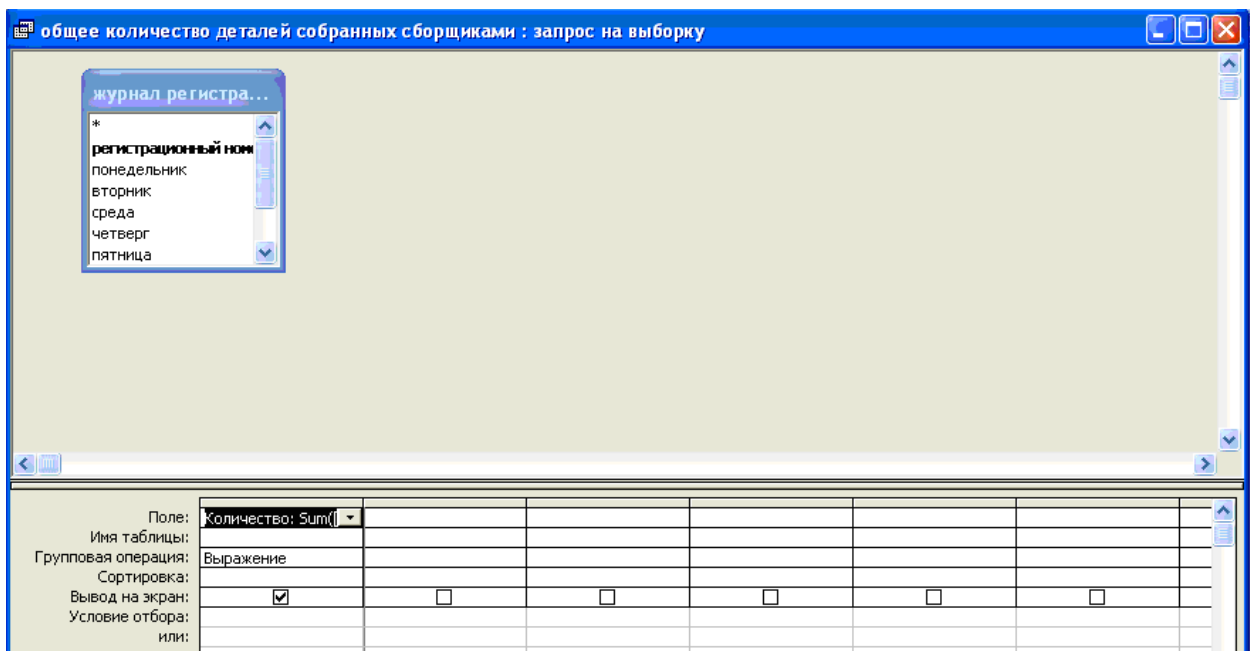


Рис. 8.20. Разробка запиту "Кількість" у Конструкторі

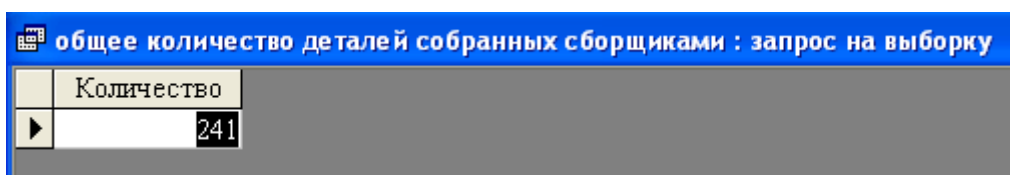


Рис. 8.21. Запит на вибірку "Кількість"

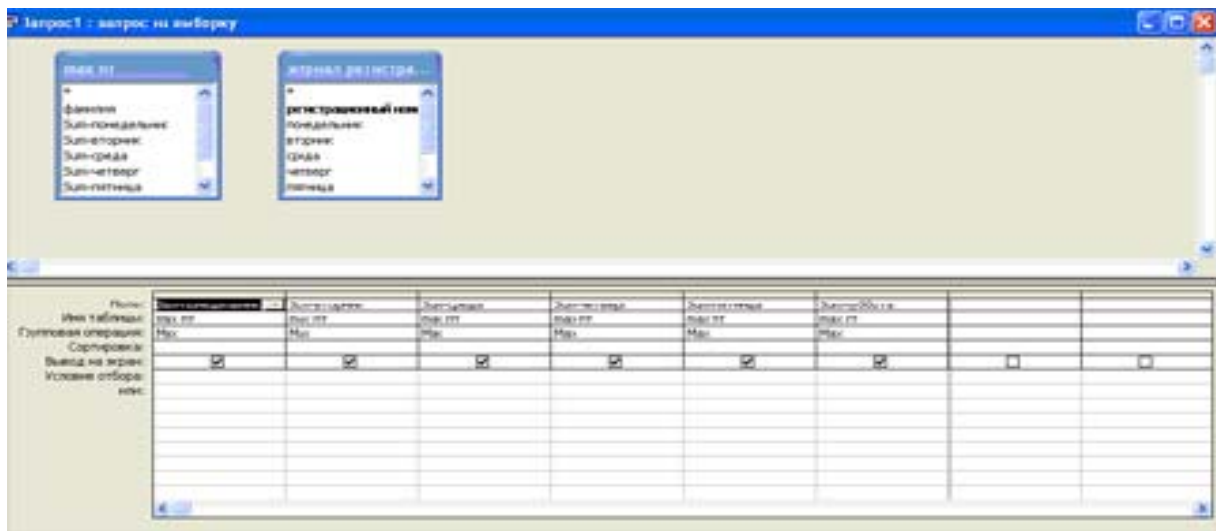


Рис. 8.22. Запит 1 у Конструкторі до пункту "Продуктивність праці"

| Запрос1 : запрос на выборку | | | | | | |
|-----------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | Мах-Sum-поне | Мах-Sum-втор | Мах-Sum-сред | Мах-Sum-четв | Мах-Sum-пятн | Мах-Sum-субб |
| ▶ | 14 | 10 | 12 | 11 | 12 | 8 |

Рис. 8.23. Результат Запиту 1

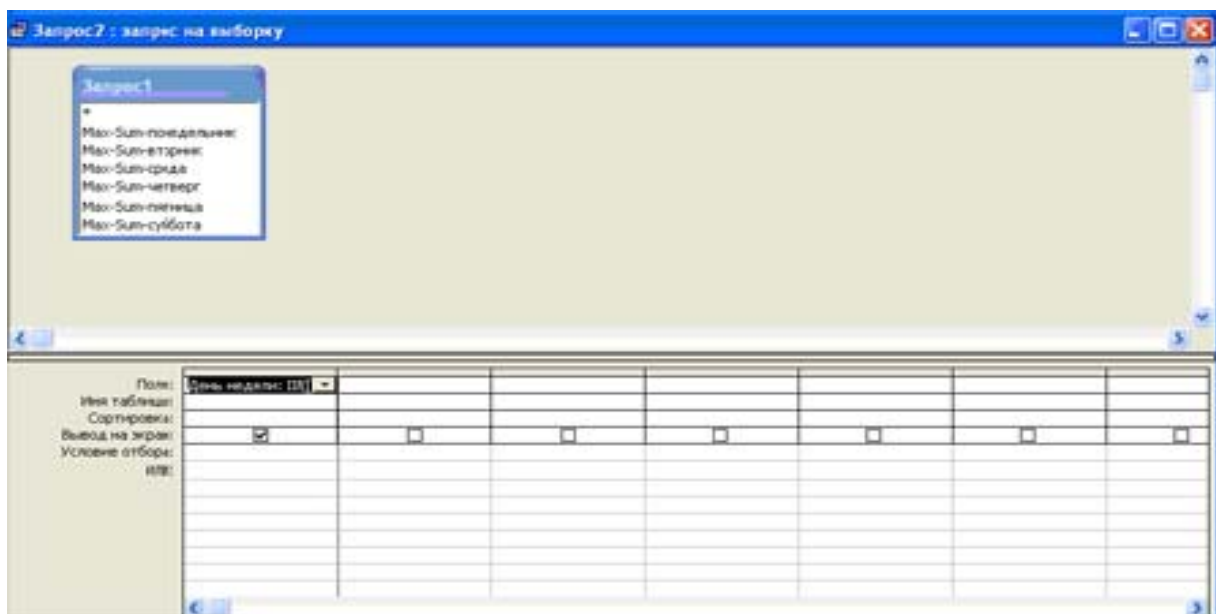


Рис. 8.24 Запит 2 у Конструкторі до пункту "Продуктивність праці"

| Запрос2 : запрос на выборку | |
|-----------------------------|-------------|
| | День недели |
| ▶ | понедельник |

Рис. 8.25. Результат Запиту 2

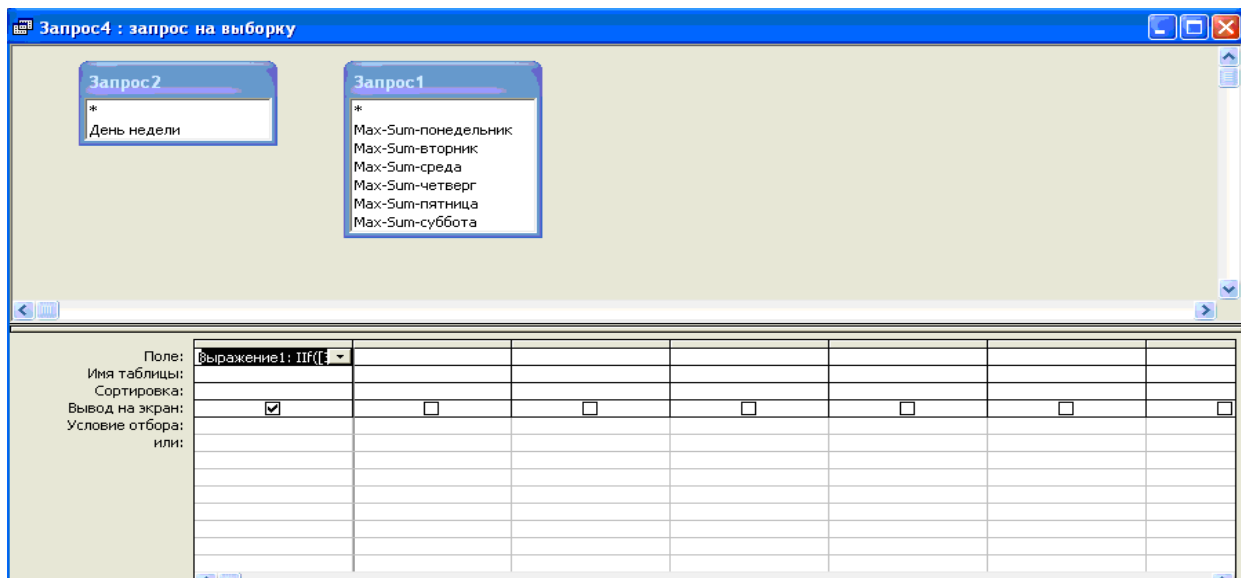


Рис. 8.26. Запит 4 у Конструкторі до пункту "Продуктивність праці"

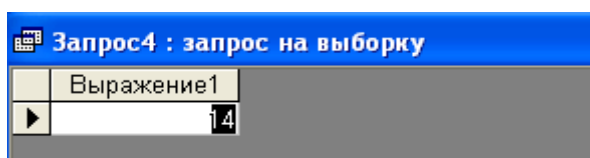


Рис. 8.27. Результат Запиту 4

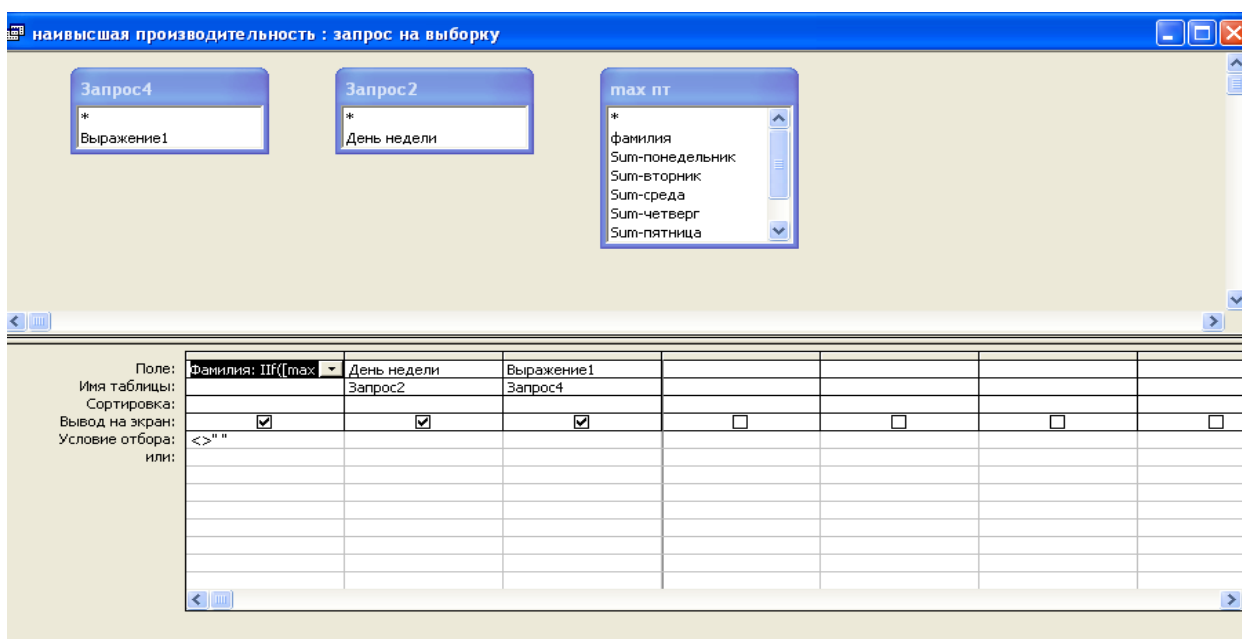


Рис. 8.28. Запит у Конструкторі до визначення найвищої продуктивності праці

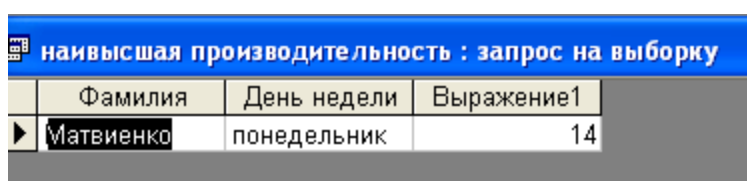


Рис. 8.29. Запит на вибірку "Продуктивність праці"

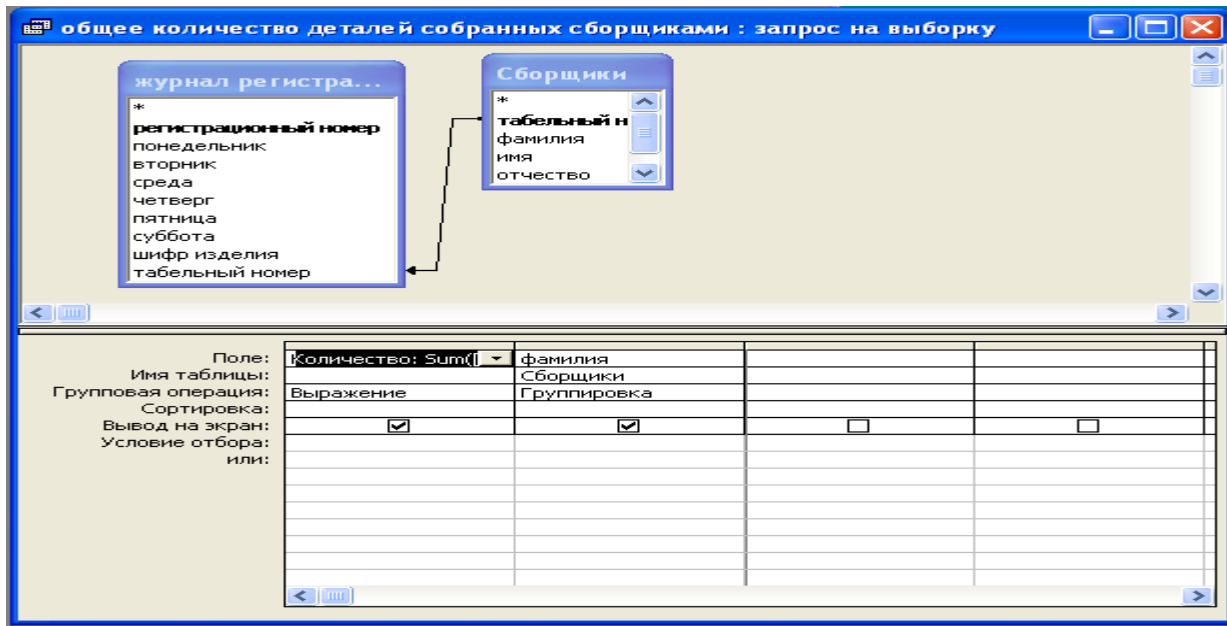


Рис. 8.30. Запит у Конструкторі до визначення загальної кількості деталей зібраних збирачами

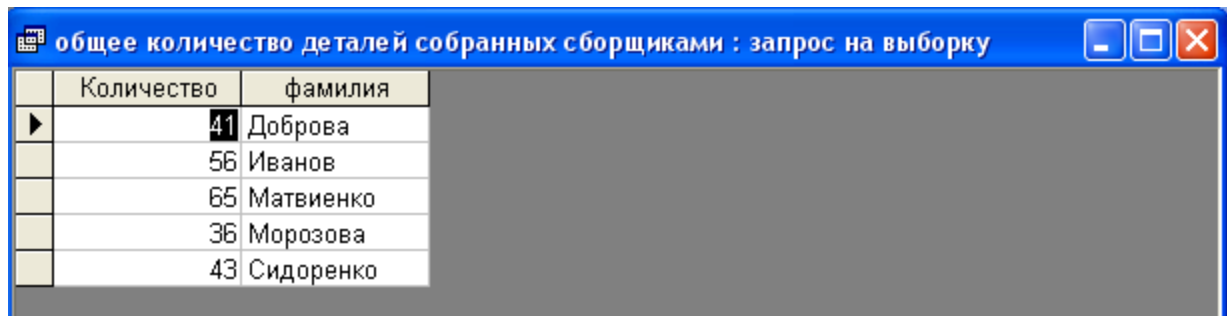


Рис. 8.31. Запит на вибірку "Загальна кількість деталей"

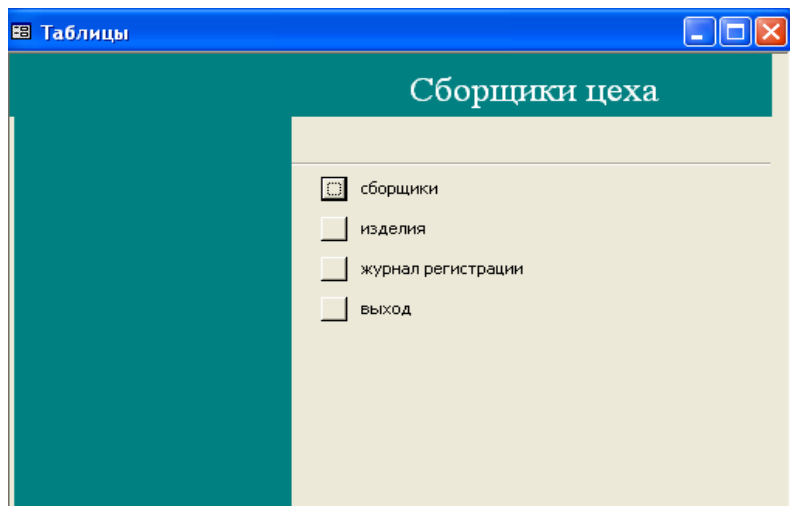


Рис. 8.32. Кнопковая форма "Таблицы"

Від таблиць "Збирачі", "Вироби" та "Журнал реєстрації" які виводяться при натисненні відповідних кнопок зображені на рис. 8.4, 8.5 і 8.6.

9. ЗАВДАННЯ ДО САМОСТІЙНОГО ВИКОНАННЯ

Завдання на створення бази даних MS Access і вказівки до їхнього виконання.

- Створити логічну модель бази даних: для заданої предметної області, представленої у вигляді переліку реквізитів, сформувати таблиці, визначити в них ключові поля, описати імена, типи й властивості полів і створити зв'язку між таблицями.
- Створити форми для уведення даних у таблиці (1 проста й складна форма). Заповнити таблиці. Кожна таблиця повинна містити не менш 5 рядків.
- Створити запит згідно 3 пункту завдання по темі «Обробка табличних даних в MS Access відповідно до варіанта.
- Створити кнопкову форму для виклику створених об'єктів.

Варіанти завдань

Варіант №1.

Вихідні дані:

- **Робітники:** табельний номер, прізвище ім'я, по батькові, дата народження;
- **Цеху:** найменування цеху, категорія виробництва (основне, керування, допоміжне);
- **Рух по службі:** посада, оклад, тип роботи (штатний, сумісник, погодинник);
 - Обчислити загальну суму виплат за місяць по обраному цеху, а також середньомісячний заробіток цього цеху;
 - Створити відомість для нарахування заробітної плати робітників цього цеху.

Варіант №2.

Вихідні дані:

- **Співробітники:** прізвище, ім'я, по батькові, дата народження, дата надходження на роботу;
- **Оплата праці:** посада, оклад;
- **Відділи:** номер відділу, прізвище співробітника.
 - Визначити:
 - вік співробітників (кількість повного років) при надходженні на роботу й на поточний час;
 - кількість співробітників заданого відділу молодше 30 років;
 - мінімальний розмір окладу.
 - Створити таблицю, що містить відомості про 5-ох самих високооплачуваних співробітників фірми.

Варіант №3.

Вихідні дані:

- **Працівники:** прізвище ім'я, по батькові, цех;
- **Виробі:** найменування виробу, категорія виробу (А, В, З), вартість виготовлення;
- **Підсумки:** шифр збирача, кількість виготовлених виробів по категоріях.
 - Розрахувати:
 - загальна кількість виробів кожної категорії;
 - загальна кількість виробів, зібраних всіма робітниками заданого цеху;
 - Створити відомість для нарахування заробітної плати робітників заданого цеху середній розмір заробітної плати працівників цього цеху.

Варіант №4.

Вихідні дані:

- **Абоненти:** прізвище, ім'я, по батькові, телефон, дата установки;
- **Розцінки:** тип замовлення (по місту, область, Україна, Європа,...), ціна 1 хв. розмови;
- **Замовлення:** телефон виклику, викликуваний пункт, час у хвилинах.
 - Розрахувати загальна кількість телефонів, установлених починаючи із заданого року по сьогоднішній. Видавати по вводить фамілії абонента номер його телефону.
 - Створити таблицю, що містить прізвища заборжників і їхні телефони.

Варіант №5.

Вихідні дані:

- **Поставка:** постачальник, дата поставки, обсяг поставки;
- **Іграшки:** артикул, найменування, ціна, нижні й верхня вікові границі;
- **Чеки:** номер чека, дата продажу, сума.
 - Визначити вартість найдорожчої іграшки і її найменування. За введеним значенням А, В і Х, видавати назва іграшки, що за вартістю не перевищує Х і пасує дитині від А до В років.
 - Створити таблицю, що містить наступну інформацію: найменування іграшок, які пасують дітям від 1 до 3 років і їхньої ціни.

Варіант №6.

Вихідні дані:

- **Студенти:** прізвище, ім'я по батькові студента; код групи, дата народження;
- **Предмети:** найменування предмета, категорія предмета (фундаментальна, професійно-орієнтована, на вибір), тип атестації (залік, іспит);
- **Журнал:** оцінки по 5 іспитам, ознака участі в суспільній роботі.
- Визначити загальне число активістів у списку.
- Створити таблицю, що містить відомість нарахування стипендії студентам заданої групи. Розрахувати розмір стипендії по наступному принципі. Студентові, що одержав всі оцінки «5» і активно бере участь у суспільній роботі, призначається підвищена стипендія - доплата 50%. Студентові, що одержав «4» і «5», призначається звичайна стипендія - її необхідно задати. Студентові, що одержав одну оцінку «3», але активно займається суспільною роботою, також призначається звичайна стипендія. Іншим студентам стипендія не призначається.

Варіант №7.

Вихідні дані:

- **Автори:** прізвище, ім'я по батькові, назва книги;
- **Книги:** рік видання, кількість екземплярів;
- **Місце розташування:** шифр книги, номер стелажа, номер шафи, номер полиці.
- Визначити загальна кількість книг у колекції, а також число книг заданого року видання. По заданому авторі й назві книги видати інформацію про місцезнаходження книги.
- Створити таблицю, що містить інформацію про книги заданого автора, що перебувають у колекції.

Варіант №8.

Вихідні дані:

- **Група:** факультет, шифр групи, прізвище куратора, посада;
- **Студент:** прізвище студента, шифр групи, номер залікової книжки, дата народження, дата надходження;
- **Результати сесії:** оцінки по 5 іспитам і результати здачі 5 заліків («З» - залік, «Н» - незалік).
- Обчислити середній бал, отриманий кожним студентом заданої групи, і середній бал цієї групи по кожному предметі. Визначити загальна кількість заборгованостей (по іспитам і залікам у сумі) кожного студента заданої групи й загальне число студентів-боржників тієї ж групи.
- Створити таблицю, що містить відомості про невстигаючих студентів :групу, прізвище й кількість заборгованостей.

Варіант №9.

Вихідні дані:

- **Рейси:** номер рейса, пункт призначення, час вильоту, час прибуття, вартість квитка;
- **Літаки:** шифр літака, марка, кількість посадкових місць, термін служби;
- **Квитки:** дата вильоту, кількість вільних місць у літаку.
- Визначити:
 - номера рейсів і час відправлення літаків у задане місто;
 - по заданому місту і часу відправлення наявність вільних місць на рейс;
 - загальна кількість рейсів за добу в задане місто.
- Створити таблицю, що містить номери рейсів і час відправлення літаків у задане місто

Варіант №10.

Вихідні дані:

- **Поставка:** артикул взуття (артикул починається з букви Д – для дамського взуття, М – для чоловічий, П – для дитячого взуття, наприклад, А0321), найменування, обсяг поставки;
- **Взуття:** кольори, вартість;
- **Наявність:** розмір, кількість.
- Визначити:
 - вартість взуття заданого артикула, і які розміри є в наявності;
 - загальна кількість пар дитячого взуття, наявної в магазині і її сумарній вартості.
- Створити таблицю, що містить інформацію про всі моделі дамського взуття

Варіант №11.

Вихідні дані:

- **Гравці:** прізвище, ім'я, по батькові, рік народження, назва футбольний клубу;
- **Футбольний клуб:** назва клубу, прізвище директора, прізвище головного тренера;
- **Результати:** шифр гравця, число занедбаних їм шайб, число зроблених їм гольових передач, зароблене штрафний час.
- Обчислити загальне число шайб, забитих хокеїстами кожної команди, і сумарний штрафний час.
- Створити таблицю, що містить прізвища шести кращих гравців з обох команд, і суму окупів кожного гравця (голи + передачі).

Варіант №12.

Вихідні дані:

- **Студенти:** прізвище, ім'я, дата народження дата надходження;
- **Вибір дисципліни:** код студента, найменування п'яти дисциплін (обирана дисципліна відзначається символом «1», інакше - пробіл);
- **Успішність:** середній бал, наявність заборгованості.
 - Обчислити кількість слухачів кожної дисципліни. Визначити число слухачів заданої дисципліни, у яких середній бал перевищує заданий.
 - Створити таблицю, що містить прізвище, групу й середній бал всіх слухачів заданої дисципліни. Якщо число їх перевищить задане, то відібрати студентів, що мають більше високий середній бал успішності.

Варіант №13.

Вихідні дані:

- **Рейси:** номер поїзда, станція призначення, час відправлення, час прибуття, вартість квитка у вагони кожного виду окремо;
- **Поїзда:** кількість посадкових місць у купейних вагонах, плацкартних, кількість місць у вагони підвищеної комфортності;
- **Квитки:** дата відправлення, номер поїзда, наявність квитків у вагони кожного виду окремо.
 - Визначити:
 - кількість вільних місць у купейні вагони поїзда із заданим номером;
 - кількість поїздів, що впливають до заданої станції призначення.
 - Створити таблицю, що містить інформацію про поїзди, які відправляються до заданої станції в заданому інтервалі часу (часовий інтервал задати двома значеннями, наприклад 13:00 і 18:30).

Варіант №14.

Вихідні дані:

- **Співробітники:** табельний номер прізвище, ім'я, по батькові, дата народження, дата надходження на роботу;
- **Відділи:** номер відділу, кількість співробітників, прізвище начальника;
- **Рух по службі:** посада, оклад, тип роботи (штатний, сумісник, погодинник).
 - Розрахувати стаж роботи всіх співробітників; середній стаж роботи співробітників заданого відділу; кількість співробітників з окладом нижче заданого.
 - Створити таблицю, що містить список співробітників пенсійного віку (на сьогоднішній день) із вказівкою стажу роботи. Урахувати різницю в пенсійному віці жінок і чоловіків.

Варіант №15.

Вихідні дані:

- **Пацієнти:** прізвище, ім'я. По батькові, підлога, дата народження, місце проживання (місто);
- **Палата:** кількість місць у палаті;
- **Облік:** дата надходження, діагноз надходження, уточнений діагноз, дата виписки, якщо виписався).
 - Визначити:
 - кількість іногородніх, прибулих у клініку;
 - кількість пацієнтів із заданим діагнозом;
 - кількість пацієнтів пенсійного віку.
 - Створити таблицю, що містить список пацієнтів старше заданого віку із заданим діагнозом.

Варіант №16.

Вихідні дані:

- **Пасажир:** прізвище, шифр багажу;
- **Багаж:** кількість речей, вага.
- **Облік:** дата здачі, час здачі, номер секції, номер осередку.
- Визначити:
 - загальна середня вага однієї речі;
 - багаж, у якого середня вага однієї речі відрізняється не більше ніж на 0,3 кг від загальної середньої ваги однієї речі;
 - кількість пасажирів, що мають більше 2 речей.
- Створити таблицю, що містить інформацію про багаж, вага якого перевищує заданий

Варіант №17.

Вихідні дані:

- **Компанія:** найменування компанії, дата створення компанії, прізвище директора, номінал акції;
- **Курс:** дата, продаж, покупка;
- **Облік:** кількість проданих акцій, кількість куплених акцій.
- Визначити:
 - середня кількість проданих і куплених акцій;
 - максимальну різницю між курсом продажу й покупки акцій;
 - сумарна кількість акцій, проданих всіма фірмами, і загальну суму, на яку вони продані.
- Створити таблицю, що містить найменування фірми й вартість проданих акцій.

Варіант №18.

Вихідні дані:

- **Замовник:** прізвище, адреса, телефон;
- **Ремонт:** номер замовлення, найменування встаткування, вид ремонту, вартість;
- **Облік:** прізвище майстра, дата початку ремонту, дата закінчення ремонту.
- Визначити:
 - сумарну вартість всіх замовлень;
 - кількість замовлень на ремонт заданого виду;
 - мінімальна вартість ремонту.
- Створити таблицю, що містить відомості про тривалість ремонту замовлень, оформлених навесні: номер замовлення, прізвище замовника, найменування встаткування, тривалість замовлення.

Варіант №19.

Вихідні дані:

- **Абоненти:** прізвище, ім'я, по батькові, телефон, дата установки;
- **Розцінки:** код міста, вартість 1 хвилини розмови;
- **Замовлення:** дата розмови, телефон виклику, викликуваний пункт, тривалість у хвилинах.
- Визначити:
 - максимальну вартість розмови;
 - сумарну вартість всіх розмов;
 - загальна кількість розмов у місто із заданим кодом.
- Створити таблицю, що містить відомості про вартість розмов, які відбулися в інтервалі між двома заданими датами.

Варіант №20.

Вихідні дані:

- **Товари:** шифр товару, найменування товару, категорія (А, В, З), країна-виробник;
- **Поставка:** дата поставки, постачальник (найменування фірми), обсяг, оптова ціна;
- **Облік:** дата продажу, роздрібна ціна, кількість проданого товару.
- По заданому шифрі товару видавати інформацію про нього.
- Визначити:
 - сумарний прибуток від продажу всіх товарів;
 - найменування товарів, продаваних по найвищій і найнижчій ціні.
- Створити таблицю, що містить найменування товару й сумарний виторг.

Варіант №21.

Вихідні дані:

- **Продукція:** номер цеху виготовлювача, найменування виробу;
- **Вартість:** код виробу, собівартість;
- **Облік:** дата виготовлення, кількість виготовлених виробів, ціна.
- Визначити:
 - сумарну різницю між собівартістю й ціною всіх виготовлених виробів;
 - загальна кількість виробів, виготовлених раніше заданої дати
 - ціну виробу по заданому найменуванню.
- Створити таблицю, що містить відомості про товари, виготовлених у заданому цеху

Варіант №22.

Вихідні дані:

- **Робітники:** прізвище, ім'я, по батькові, дата народження, дата надходження на роботу, номер цеху;
- **Розцінки:** розряд, вартість однієї години;
- **Облік:** дата, кількість відпрацьованих годин.
- Визначити:
 - середня кількість годин, відпрацьованих за день;
 - максимальну вартість однієї години;
 - по заданій даті кількість відпрацьованих годин.
- Створити таблицю, що містить наступні відомості про працівників заданого цеху:

Варіант №23.

Вихідні дані:

- **Клієнти:** прізвище, адреса, телефон;
- **Замовлення:** номер замовлення, найменування виробу, прізвище майстра;
- **Облік:** дата прийому, дата виконання замовлення, вартість замовлення.
- Визначити:
 - кількість замовлень, виконаних майстром із заданим прізвищем;
 - вартість самого дорогого замовлення;
 - середню вартість замовлень.
- Створити таблицю, що містить прізвище клієнта, номер замовлення й тривалість його виконання.

Варіант №24.

Вихідні дані:

- **Імпортери:** фірма-імпортер, країна, найменування товару;
- **Поставка:** шифр товару, обсяг партії в штуках, вартість 1 штуки в У. Е.,
- **Облік:** дата поставки, дата одержання, підтвердження приймання партії.
- Визначити:
 - сумарний обсяг товарів, імпортованих заданою країною;
 - сумарну вартість партії товару по заданому шифрі
 - мінімальну вартість товару.
- Створити таблицю, що містить відомості про вартість товарів, імпортованих заданою країною. Таблиця повинна містити найменування товару й сумарну вартість партії.

Варіант №25.

Вихідні дані:

- **Робітники:** прізвище, ім'я, по батькові, дата народження, дата надходження на роботу, номер цеху;
- **Оплата:** розряд, оплата за виготовлення 1 якісної деталі;
- **Облік:** дата, кількість виготовлених деталей, кількість бракованих деталей.
- Визначити:
 - загальна кількість бракованих деталей, виготовлених всіма майстрами заданого цеху;
 - суму штрафу за кожен браковану деталь становить 20% від оплати за якісну;
 - прізвище майстра, що виготовив максимальну кількість якісних деталей.
- Створити таблицю, що містить відомості про оплату праці робітників. Таблиця повинна містити прізвище робітника, номер цеху й суму до виплати з урахуванням штрафу й податку (податок становить 13% від вартості оплати).

Варіант №26.

Вихідні дані:

- **Відділення:** номер відділення, прізвище завідуючого, номер корпусу, поверх;
 - **Лікування:** шифр хвороби, тривалість;
 - **Оплата:** діагностика, вартість 1 дня лікування, витрати на ліки.
- сумарна вартість відпрацьованого їм часу.
- Визначити:
 - за назвою хвороби витрати на ліки;
 - середню вартість 1 дня лікування;
 - розрахувати сумарну вартість лікування кожної хвороби, включаючи витрати на ліки.
 - Створити таблицю, що містить відомості про вартість лікування в заданому відділенні.

Варіант №27.

Вихідні дані:

- **Книги:** найменування книги, прізвище автора, видавництво, рік видання, тираж;
- **Магазини:** шифр книги, номер магазину, ціна;
- **Облік продажів:** код продажу, продано, залишок.
- Визначити:
 - кількість проданих книг у заданому магазині;
 - сумарну вартість всіх непроданих книг;
 - середню ціну однієї книги.
- Створити таблицю, що містить сумарну вартість книг, проданих кожним магазином

Варіант №28.

Вихідні дані:

- **Деталі:** найменування деталі, цех-виготовлювач;
- **Виготовлення:** шифр матеріалу, шифр деталі, витрата матеріалу на 1 деталь;
- **Облік:** дата виготовлення, кількість виготовлених деталей, кількість шлюбу.
- Визначити:
 - для всіх деталей сумарна витрата матеріалу на шлюб;
 - кількість якісних деталей;
 - деталь, на яку витрачається найбільше матеріалу.
- Створити таблицю, що містить шифр деталі й відсоток браку.

Варіант №29.

Вихідні дані:

- **Ліки:** назва ліки, категорія (антибіотик, жарознижуюче, вітамін, протизапальне, вітамін, антидепресант), дата виготовлення, дата витікання строку;
- **Вартість:** шифр ліки, код аптеки, ціна 1 упакування;
- **Продаж:** кількість проданих упакувань, залишок.
- Визначити:
 - сумарна кількість упакувань ліків, проданих всіма аптеками й приналежної заданої категорії.
 - вартість всіх непроданих упакувань;
 - середню вартість ліків.
- Створити таблицю, що містить інформацію про прострочені ліки.: номер аптеки, назву, категорію й дату витікання строку .

Варіант №30.

Вихідні дані:

- **Продавці:** табельний номер, прізвище ім'я, по батькові, дата народження.
- **Товари:** шифр товару, тип товару, складність продажу (середня, висока);
- **Журнал реєстрації:** кількість проданих товарів по днях тижня понеділок, вівторок, ... , субота;
 - Визначити:
 - загальна кількість товарів, проданих кожним продавцем;
 - прізвище продавця, що продав найбільше число товарів, і визначити день, коли він досяг найвищої продуктивності праці.
 - Створити таблицю, що містить наступну інформацію: прізвище продавця й загальна кількість товарів, продана їм за тиждень.

ВИСНОВОК

У методичних вказівках зібрані основні відомості по системам керування базами даних програмного продукту сімейства MICROSOFT OFFICCE ACCESS, викладені елементи інтерфейсу, загальні принципи керування базами даних, організація запитів, різних за формою і змістом.

Всі можливості запитів, які основані на одній таблиці, відповідають і створенню багатотабличних запитів. Ці запити здатні відбирати дані з декількох таблиць, що особливо корисно при створенні таких компонентів систем керування базами даних, як форми і звіти. Для полегшення роботи користувача, особливо початківця, рекомендоване створення запиту за допомогою майстра запитів.

Невеликий об'єм методичних вказівок не дозволив торкнутися цілого ряду питань. Проте, більшості користувачів з цими розділами доводиться зустрічатися досить рідко. При знайомстві з цим пакетом на більш детальному рівні бажано використовувати інші джерела, які наведені у списку рекомендованої літератури, або вирішувати питання за допомогою довідкової системи пакету ACCESS.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ:

1. Дженинкс Р. Использование Access 97. - СПб.: Питер, 1998. - 1072с.
2. Баркер Ф., Скот Ф. Использование Access 97 - К.: Диалектика 2002. - 336с.
3. Золотова С.И. Практикум по Access. - М: Финансы и статистика, 2008. - 144с.
4. Кауфельд Д. Microsoft Office Access 2003 для "чайников". - М.: Диалектика, 2004. - 320с.
5. Проектирование баз данных. СУБД Microsoft Access / Учебное пособие // Н.Н. Гринченко, Е.В. Гусев, Н.П. Макаров. - М.: Горячая Линия – Телеком, 2004. - 240с.
6. Гончаров А.Ю. Access 97 в примерах. - СПб: Питер, 1997. - 320
7. Степанов В. Microsoft Access 2003 для начинающих. - М.: Аквариум-Принт, Дом печати – Вятка, 2004. - 128с.
8. Вейскас Д. Эффективная работа с MS Access 97. – СПб: Питер, 1999. - 976с.
9. Руденко В.Д., Макаручук О.М., Патланжоглу М.О. Практичный курс информатики. - К: ДиаСофт, 1997. - 300с.
10. Послед Б. Access 2000. Базы данных и приложения. Лекции и упражнения. – К: ДиаСофт, 2000. - 512с.
11. Майкл Р. Ирвин, Керри Н. Праг. Access 2002. Библия пользователя. – М.: Диалектика, 2003. – 1216с.
12. Начальный курс информатики для пользователей персональных компьютеров / Уч.-метод. пособие // Павлыш В.Н., Анохина И.Ю., Кононенко И.Н., Зензеров В.И. – Донецк: ДонНТУ, 2005. – 235 с.
13. Иллюстрированный самоучитель по Access 2002. / [Электронный ресурс] http://computers.plib.ru/office/Access_2002/index.html

ЗМІСТ

| | |
|---|----|
| Передмова..... | 3 |
| 1. Таблиці | 4 |
| 1.1. Створення таблиць..... | 4 |
| 1.2. Створення нової бази даних | 5 |
| 1.3. Перегляд і редагування таблиць..... | 9 |
| 2. Зв'язування таблиць на схемі даних | 11 |
| 3. Проектування форм і робота з ними | 15 |
| 4. Робота з даними за допомогою запитів..... | 20 |
| 4.1. Створення запиту на вибірку | 20 |
| 4.2. Обчислювальні поля | 24 |
| 4.3. Підсумкові запити..... | 27 |
| 4.4. Використання функцій дати /часу..... | 28 |
| 4.5. Використання функцій керування при створенні запитів | 32 |
| 4.6. Формування складних запитів | 34 |
| 5. Створення звіту як об'єкта бази даних | 37 |
| 6. Елемент керування Кнопка | 42 |
| 7. Макроси..... | 48 |
| 8. Приклад розробки та управління базою даних | 54 |
| 9. Завдання до самостійного виконання..... | 66 |
| Висновок..... | 77 |
| Список рекомендованої літератури..... | 78 |

«ОСНОВИ УПРАВЛІННЯ БАЗАМИ ДАНИХ У MICROSOFT ACCESS»

**Методичні вказівки і завдання до лабораторних робіт
для студентів економічних спеціальностей**

Автори: *Зенеров Володимир Іванович, доцент*
Славінська Людмила Василівна, ст. викладач
Перінська Олена Володимирівна, асистент

Підп. до друку 31.05.2010 р. Формат 60x84¹/₁₆. Папір офсетний.
Різографічний друк. Ум.-др. арк. 5,0. Тираж 100 прим.
Замовл. №0531.

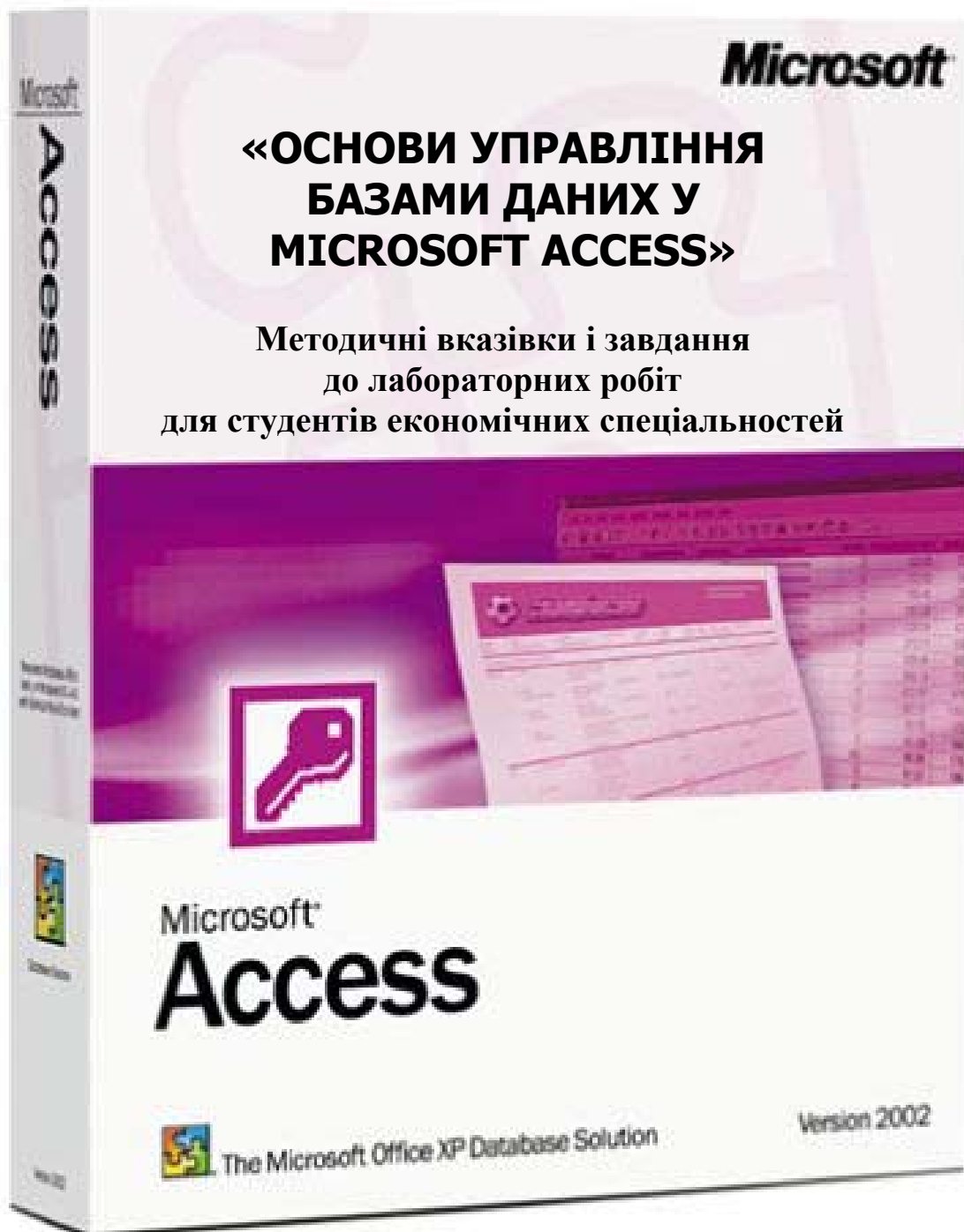
Донецький національний технічний університет
83000 м.Донецьк-00, вул. Артема, 58

Друк з оригінал-макету МПП "ВІК"
Свідоцтво про реєстрацію ДК №382 від 26.03.2001 р.
83059, м. Донецьк, вул. Разенкова, 12/17, тел. (062) 381-70-87



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ
ТЕХНІЧНИЙ УНІВЕРСИТЕТ



Донецьк 2010

