

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ,  
МОЛОДІ ТА СПОРТУ УКРАЇНИ**

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ**

**Методичні рекомендації  
до проведення виробничої практики  
для студентів 3 курсу напрямку підготовки 6.030506  
"Прикладна статистика"  
денної форми навчання**

**Харків. Вид. ХНЕУ, 2012**

Затверджено на засіданні кафедри статистики та економічного прогнозування.

Протокол № 5 від 12.12.2011 р.

**Укладачі:** Раєвнєва О. В.  
Горохова О. І.  
Гольтяєва Л. А.

**М54**        Методичні рекомендації до проведення виробничої практики для студентів 3 курсу напряму підготовки 6.030506 "Прикладна статистика" денної форми навчання / укл. Раєвнєва О. В., Горохова О. І., Гольтяєва Л. А. — Х. : Вид. ХНЕУ, 2012. — 60 с. (Укр. мов.)

Подано структуру і зміст виробничої практики, а також методичні рекомендації щодо оформлення звіту з практики.

Рекомендовано для студентів 3 курсу напряму підготовки 6.030506 "Прикладна статистика".

## Вступ

**Метою** виробничої практики є набуття професійних компетенцій щодо обробки статистичної інформації на підставі використання програмних продуктів Microsoft Excel та Access.

Основними **завданнями** практики є набуття студентами навичок щодо:

роботи з програмними продуктами Microsoft Excel та Access;  
введення та обробки статистичних даних засобами даних пакетів;  
імпорту, експорту статистичної інформації за допомогою макросів Microsoft Excel;

перетворення інформації за допомогою макросів в різні форми її збереження;

використання макросів для автоматизації звичайних задач та створення функцій користувача програмного продукту;

роботи з базовими модулями СУБД Microsoft Access;  
побудови баз даних за допомогою СУБД Microsoft Access;  
використання елементарних засобів СУБД Access для аналізу даних.

**Етапи проходження** виробничої практики:

1. Знайомство з програмними продуктами Microsoft Excel та Access.
2. Знайомство з базовими модулями програм.
3. Ознайомлення та вивчення макросів для візуалізації та обробки статистичних даних.
4. Побудова та ознайомлення з існуючими БД в Microsoft Access.
5. Використання базових модулів Microsoft Access для аналізу та обробки статистичної інформації.
6. Virішення завдань за допомогою програмних продуктів Microsoft Excel та Access, підготовка звіту з практики.

У результаті проходження виробничої практики студенту необхідно набути таких **компетенцій** (табл. 1).

**Вимоги щодо компетенцій, які повинен здобути студент, після проходження виробничої практики**

Назва компетенцій	Зміст компетенцій	Вміння студента щодо даної компетенції
1. Обліково-статистична	1.1. Здатність до здійснення статистичних спостережень	1.1.1. Володіти можливостями програмних продуктів Microsoft Excel та Access для збору, обробки та поширення даних
	1.2. Здатність проводити статистичні розрахунки	1.2.1. Проводити розрахунок статистичних показників, використовуючи набір інструкцій Microsoft Excel. 1.2.2. Формувати інформаційний простір для статистичного дослідження
2. Аналітична	2.1. Здатність аналізувати та використовувати зведену інформацію	2.1.1. Збирати та обробляти інформацію відповідно до наукових методів її первинної обробки. 2.1.2. Здійснювати обмін необхідної інформації між вивчаємими пакетами прикладних програм. 2.2.2. Складати ґрунтовні аналітичні звіти, відповідно до поставленої мети дослідження
3. Інформаційна	3.1. Здатність використовувати сучасні інформаційні пакети для дослідження соціально-економічних процесів	3.1.1. Використання пакета Microsoft Excel для обробки статистичної інформації. 3.1.2. Здійснення обробки інформації на підставі проектування баз даних в Microsoft Access. 3.1.3. Застосування макросів для вирішення завдань відповідно до конкретної ситуації. 3.1.4. Використання засобів Microsoft Access для аналізу статистичної інформації. 3.1.5. Вирішення професійних задач за допомогою сучасних інформаційних систем та програмних продуктів. 3.1.6. Формування статистичних звітів на підставі сучасних інформаційних засобів візуалізації та обробки статистичної інформації

## 1. Загальні відомості про створення та використання макросів

**Макрос** – це набір інструкцій, які вказують послідовність дій, які MS Excel повинен виконати автоматично. Використовують макроси для автоматизації задач або операцій, які треба часто виконувати.

Макроси становлять комп'ютерні програми, але вони не виконуються незалежно, а працюють тільки в середовищі додатка, де вони створювались (MS Word, MS Excel тощо). Макроси пишуться мовою програмування VBA та зберігаються в спеціальних листах книги MS Excel — *листах модулів*. Лист модуля не відображається ярликом безпосередньо у вікні книги. Доступ до нього можна дістати тільки з вікна редактора VBA.

Макроси можна створювати:

- в автоматичному режимі з подальшим редагуванням у редакторі VBA;
- записувати мовою VBA на *Спеціальному листі модуля*.

Найчастіше використовують перший спосіб. Для автоматичного запису макросу використовується спеціальна програма *Macrorecorder*.

Запис макросу можна вести в *абсолютному* та *відносному* режимах. За замовчуванням Macrorecorder використовує **абсолютний** режим запису, тобто Macrorecorder у текст макросу записує абсолютні посилання на комірки, які ми вибираємо. Результат виконання макросу при ньому не залежить від комірки, з якої він запускався.

Під час запису макросу у відносному режимі Macrorecorder у текст макросу записує відносні посилання на комірки. Тому результат виконання макросу залежить від комірки, з якої запускався макрос.

Для перегляду значень змінних використовуються два вікна: *Watches* та *Locals*.

Вікно *Locals* показує значення всіх локальних змінних. Ці змінні, визначені всередині поточної процедури і, отже, непридатні до інших процедур.

Вікно *Watches* відображає значення всіх змінних або виразів, які були поміщені в нього. Додаток Excel автоматично оновлює вирази в цьому після виконання кожного коду рядка і діє згідно з вказаним типом контрольних значень.

Щоб помістити змінну або вираз у вікно *Watches*, використовується команда *Add Watch* меню *Debug*, після чого відкривається діалогове вікно (рис. 1.1)

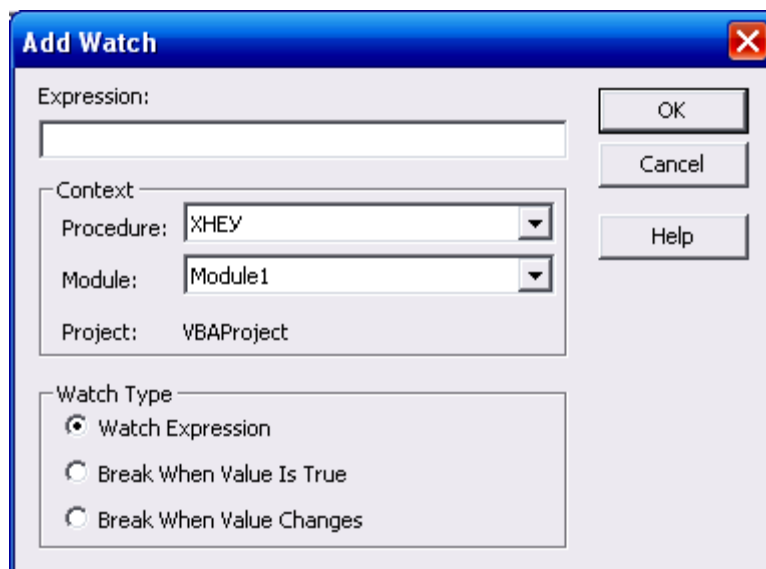


Рис. 1.1. Діалогове вікно *Add Watch*

Перемикач *Watch Type* установлює в одне з трьох положень:

*Watch Expression* – вираз поміщується у вікно *Watches*, що дає можливість стежити за його зміною у міру виконання коду;

*Break When Value Is True* – вираз у вікні *Watches* існує тільки з двома значеннями – *True* або *False* – залежно від величини параметру виразу. Якщо змінна набуває значення *True*, додаток переходить в режим переривання і зупиняє виконання коду;

*Break When Value Changes* – додаток переходить в режим переривання при будь-якій зміні контрольного значення, тобто відбувається перехід від *True* або *False* і навпаки.

**Для запису макросу в автоматичному режимі потрібно:**

1. Включити режим запису макросу:

*Сервіс / Макрос / Начать запись* => відкривається д/в *Запись макроса*, в якому зазначити ім'я макросу та встановити інші параметри (рис. 1.2).

2. Click кнопку **ОК** (рис. 1.1) => у результаті:

а) до робочої книги додається новий лист модуля, якому надається ім'я Модуль1;

б) *Macrorecorder* розпочне запис макросу;

с) на екрані з'явиться панель інструментів **Остановить запись** з двома кнопками: **Остановить запись** та **Относительная ссылка**.

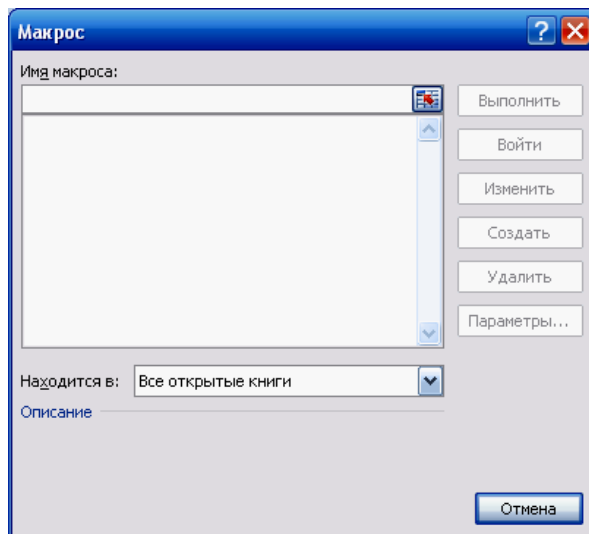


Рис. 1.2. Початкове діалогове вікно Макрос

3. Кнопка **Относительная ссылка** для запису макросу:

- в абсолютному режимі має бути не натиснута;
- у відносному режимі має бути натиснута.

4. В вікні параметри макросу (рис. 1.3) необхідно зазначити ім'я, короткий опис та де буде зберігатись макрос: якщо треба, щоб макрос був доступним з усіх книг MS Excel, вибрати "Личная книга", якщо тільки в поточній – вибрати "Эта книга".

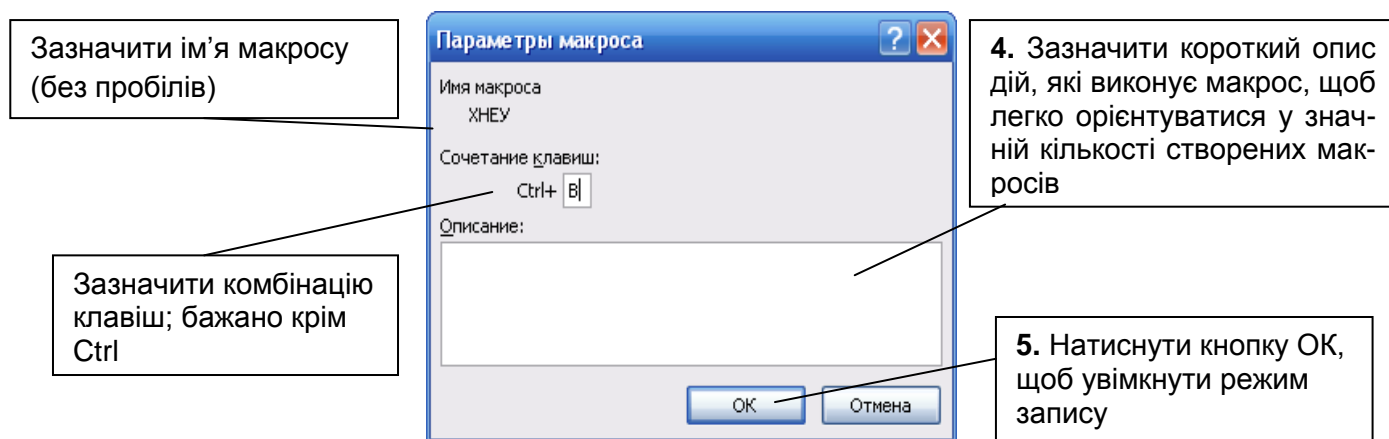



Рис. 1.3. Вікно параметрів запису макросу

5. На робочому листі виконати всі дії, які потрібно записати в макрос. Одночасно з цим Macrorecorder записує у створений модуль усі

дії, що виконуються (активізація комірок, команди, вказані за допомогою меню та кн. на п/і), ретранслюючи їх на мову програмування.

6. Після виконання всіх потрібних дій вимкнути режим запису макросу:

- *Сервіс / Макрос / Остановить запись*;
- клік на кн. **ОСТАНОВИТЬ ЗАПИСЬ**  на панелі інструментів

**Остановить запись.**

У результаті запис макросу буде припинено, п/і **Остановить запись** зникне.

Тексти записаних макросів на рис. 1.4 та 1.5.

```
ActiveCell.Offset(-24, 0).Range("A1").Select
ActiveCell.FormulaR1C1 = "55"
ActiveCell.Offset(0, 1).Range("A1").Select
ActiveCell.FormulaR1C1 = "22"
ActiveCell.Offset(0, 1).Range("A1").Select
ActiveCell.FormulaR1C1 = "=RC[-2]-RC[-1]"
```

Рис. 1.4. Тексти макросів, записаних в абсолютному режимі

```
Range("G4").Select
ActiveCell.FormulaR1C1 = "55"
Range("H4").Select
ActiveCell.FormulaR1C1 = "22"
Range("I4").Select
ActiveCell.FormulaR1C1 = "=RC[-2]-RC[-1]"
```

Рис. 1.5. Тексти макросів, записаних у відносному режимі

Перед запуском макросу, записаного у відносному режимі, треба активізувати комірку, з якої розпочнеться виконання макросу. Якщо макрос записаний у абсолютному режимі, цього робити не треба.

I спосіб. **Запуск макросу за допомогою меню**

1. Виконати команду меню *Сервіс / Макрос – Макросы* або клік на кнопку **Выполнить макрос >** на панелі інструментів Visual Basic.

2. У діалоговому вікні **Макрос**, що з'явилося, у зоні **Имя макроса**, виділити потрібний макрос і клік на кнопку **Выполнить** (або 2 клік на імені макросу).

У результаті будуть викопані дії, записані в макросі.

II спосіб. **Запуск макросу за допомогою кнопки на панелі інструментів**



Цей спосіб доцільно використовувати, коли макрос доступний з будь-якої книги, тобто записаний в особистій книзі макросів. Для запуску макросу в такий спосіб треба попередньо на будь-яку п/і винести кнопку, якій призначити потрібний макрос:

1) на панелі інструментів – "Настройка";

2) в діалоговому вікні "Настройка" – вкладка "Команды" вибрати "Макросы" – "Настраиваемая кнопка" – перетягнути на будь-яку панель інструментів (рис. 1.6);

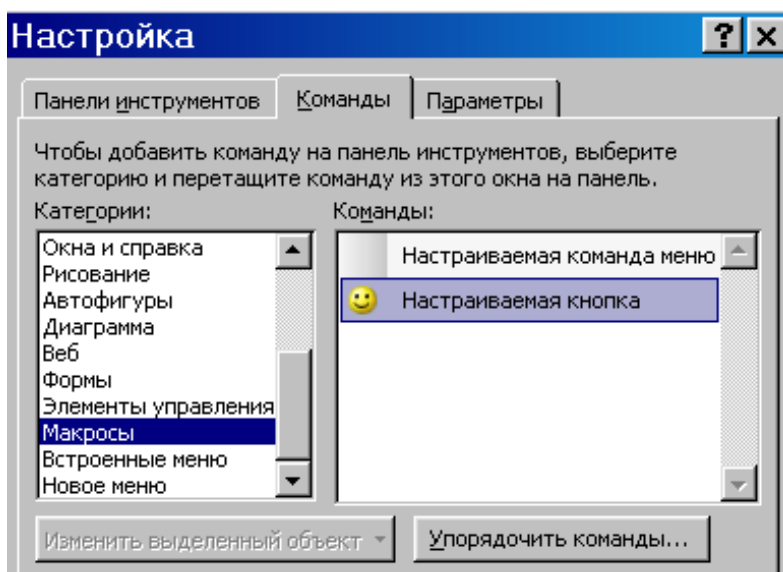


Рис. 1.6. Вікно створення кнопки на панелі інструментів

3) на цій кнопці на панелі інструментів – Назначить макрос;

4) діалогове вікно "Назначить макрос" у – зоні *Имя макроса* вибрати потрібний —> кнопку ОК (потім так само можна і змінити макрос, що призначений кнопці);

5) для заміни позначки на кнопці на іншу: на цій кнопці на панелі інструментів —> Изменить значок на кнопке... або Выбрать значок на кнопке, чи скопіювати позначку з іншої кнопки стандартним способом;

6) кнопка "Закреть" у діалоговому вікні Настройка.

Щоб запусити макрос за допомогою кнопки, достатньо клацнути на цій кнопці.

III спосіб. **Запуск макросу за допомогою графічного об'єкта Кнопка**

Цей спосіб доцільно використовувати, коли макрос доступний тільки в поточній книзі, тобто записаний у модулі цієї книги і запускається тільки на листі, на якому створюється кнопка. Для запуску макросу таким

способом треба попередньо на робочому листі створити іменований графічний об'єкт **КНОПКА**, натискання на який запускає макрос:

1) активізувати панель інструментів **Форми** (рис. 1.7);



Рис. 1.7. Панель інструментів **Форма**

2) клік на кнопку "Кнопка" на панелі інструментів "Форми";

3) намалювати в потрібному місці робочого листа прямокутник => з'явиться нова кнопка з назвою **Кнопка 1**, на екрані з'явиться діалогове вікно "Назначить макрос объекту" (рис. 1.8);

4) виділити ім'я потрібного макросу і кнопку **ОК**;

5) для того, щоб змінити напис **Кнопка 1**, треба відразу клік на напису на кнопці та набрати текст. Для запуску макросу за допомогою кнопки достатньо клацнути на цій кнопці.

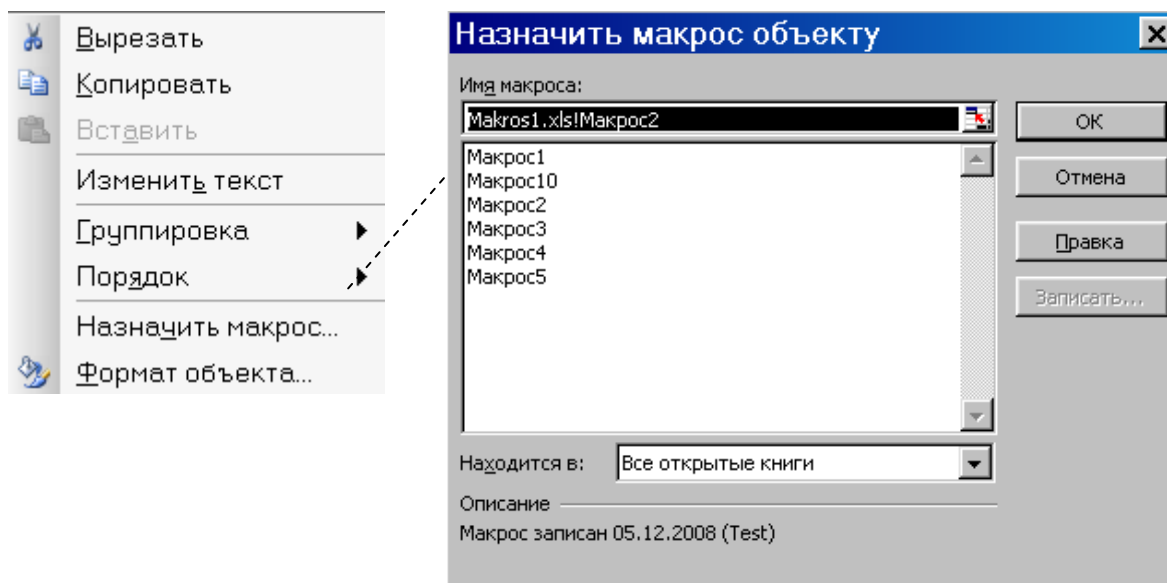


Рис. 1.8. Етапи призначення макросу кнопці

Для зміни призначеного макросу: визиваємо діалогове вікно **Назначить макрос**. Для зміни тексту: **Изменить текст**.

#### IV спосіб. **Запуск макросу за допомогою комбінації клавіш**

Якщо, створюючи макрос, призначили йому комбінацію клавіш, то для запуску макросу цим способом необхідно натиснути призначену макросу комбінацію клавіш. **Змінити** призначену комбінацію можна так:

- 1) *Сервис / Макрос / Макроси*;
- 2) у діалоговому вікні Макрос клік на кнопці Параметри;
- 3) у діалоговому вікні Параметры Макросы в зоні Сочетание клавиш установити потрібну; можна також змінити опис макросу – кнопка **ОК**.

### *Перегляд та редагування макросу в редакторі VBA*

Для того, щоб переглянути та відредагувати записаний у редакторі VBA макрос, необхідно:

- 1) *Сервис / Макрос / Макроси*;
- 2) у діалоговому вікні Макрос клацнути на кнопці "Изменить".

У результаті відкриється *вікно редактора VBA*, яке матиме вигляд, приклад якого поданий на рис. 1.9.

Редагуючи макрос, можна організувати циклічність його виконання чи додати діалог тощо. Після проведення потрібного редагування для збереження змін треба натиснути на кнопку Сохранить. Для того, щоб виконати макрос з вікна редактора VBA, слід натиснути на кнопку Run.

## **Редактор VBA: загальні відомості**

### *Завантаження редактора VBA*

Для завантаження редактора VBA без завантаження модуля, що містить певний макрос (спосіб описано раніше), треба з додатка MS Excel виконати команду *Сервис / Макрос / Редактор Visual Basic*.

Редактор VBA завантажується з вікна додатка MS Office, тому після його закриття закривається і редактор VBA.

Вікно редактора VBA, як і будь-яке вікно додатка, має стандартний набір елементів: рядок заголовка, рядок меню, панелі інструментів та робочу зону (рис. 1.10).

У робочій зоні в її лівій частині можна виводити *вікно проєктів* та *вікно властивостей* об'єкта. У правій частині у *вікнах коду* відбувається перегляд програмного коду вибраних (2 клік) об'єктів.

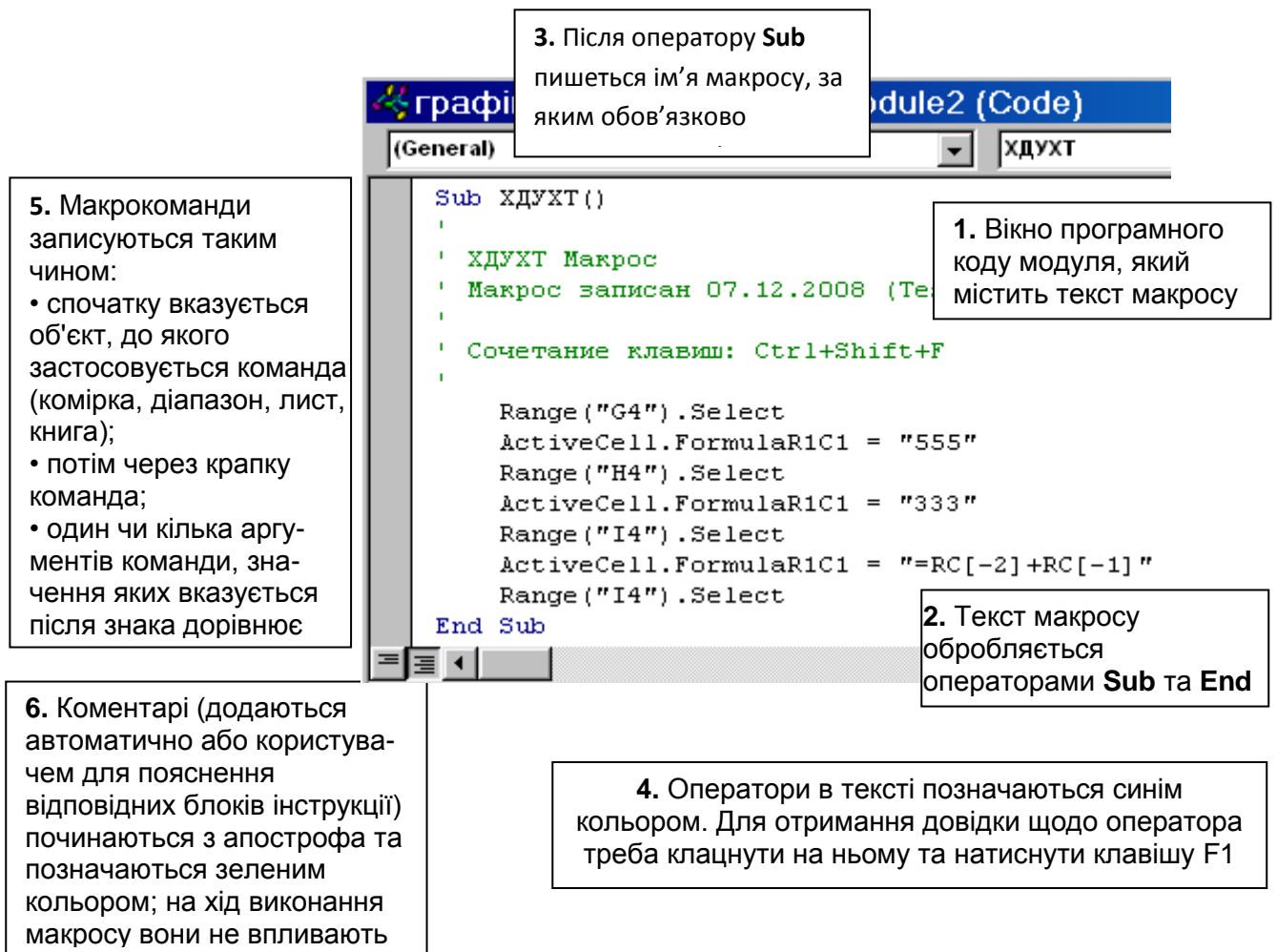


Рис. 1.9. Текст макросу мовою VBA

### Екранний інтерфейс редактора VBA

Вікно проектів відображає ієрархічну структуру об'єктів, з якими в даний момент працює VBA:

I. Файли, що автоматично завантажуються разом з MS Excel і які є закритими для доступу, наприклад, *книга Fupcres.xls*.

II. VBA-проект *книги Personal.xls* складається з таких об'єктів:

а) об'єктів MS Excel, тобто самої книги та її робочого листа;

б) форм (якщо вони є) – діалогових вікон, що їх створив користувач;

в) модулів, в яких містяться тексти процедур та функцій, що доступні з будь-якої книги MS Excel.

III. VBA-проекти всіх відкритих книг, які мають таку саму структуру, як і VBA-проект книги *Personal.xls*.

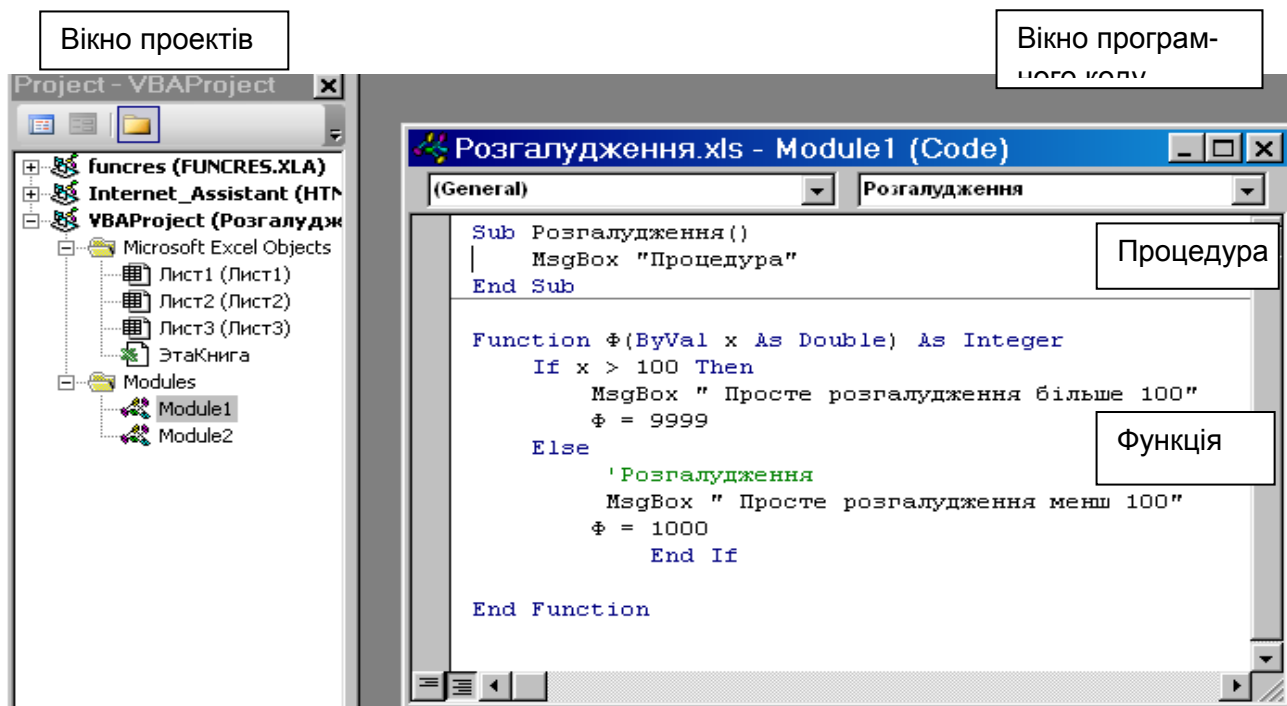


Рис. 1.10. Елементи вікна редактора VBA

Програмний код об'єкта містить набір *процедур*. Процедура – це код, написаний мовою програмування, який включає послідовність логічних кроків для виконання певної дії. Інакше кажучи, процедура – це сукупність операторів, які виконують дію або обчислення, мають власне ім'я й аргументи.

Існують такі види процедур: *підпрограми* та *функції* (рис. 1.10).

**Підпрограма** – це процедура, яка:

- 1) обрамляється операторами Sub і End Sub;
- 2) не повертає значення в основну програму;
- 3) не використовується у виразах та операціях присвоювання;
- 4) має таку структуру:

**Функція** – це процедура, яка:

- 1) обрамляється операторами Function і End Function;
- 2) завжди повертає значення в основну програму;
- 3) використовується у виразах та операціях присвоювання змінній значення функції;
- 4) має таку структуру:

*Макрос* – це підпрограма з порожнім списком аргументів, а тому його можна викликати не тільки з іншої підпрограми чи функції, а й за допомогою комбінації клавіш, команд меню, кнопок на п/і тощо.

Функції поділяють на *вбудовані* та *створені користувачем*.

Програма може складатися з кількох підпрограм та функцій, які можуть розташовуватися в одному чи кількох модулях. Модулі групуються в проекти, при цьому в одному проекті можуть бути кілька програм, які використовують спільні модулі або процедури.

Формула в комірці G6 – Ф(G4). Згідно зі змістом комірки G4 в комірці G6 з'являться значення 9999 чи 1000.

## Основні принципи створення функцій користувача

**Функція користувача** – це програма, записана користувачем мовою VBA в модулі поточної книги або особистої книги макросів, яка належить до класу процедур *Функція* (тому обрамляється операторами *Function* та *End Function*) та дозволяє провадити обчислення над значеннями її аргументів.

### Правила створення та використання функції користувача:

1. Функції користувача пишуться мовою VBA, тому що спочатку треба завантажити *Сервіс / Макрос / Редактор Visual Basic*.

2. Функцію користувача можна зробити:

- або з загальним доступом (тобто створити та зберегти на листі модуля особистої книги макросів);
- або доступною тільки в поточній книзі (тобто створити та зберегти на листі модуля поточної книги).

Для цього треба розгорнути у вікні проекту шапку потрібної книги (особистої книги макросу чи поточної).

3. Функції користувача пишуться в модулях, тому в шапці потрібної книги треба розгорнути шапку з модулями та:

- або відкрити існуючий модуль 2 click на назві модуля;
- або вставити новий: меню *Вставка / Модуль*.

4. У тексті коду модуля функція обрамляється двома операторами *Function* та *End Function*

5. Ім'я функції:

- записується без пробілів будь-якою мовою;
- не повинно збігатися з назвою жодної функції та не збігатися з назвами операторів VBA;

6. Аргументи функції:

- імена записуються будь-якою мовою та без пробілів;
- записуються через кому;
- максимальна кількість – 29.

7. Можна додати коментарі, відокремлюючи їх від програмного коду апострофом.

8. Функція користувача може містити:

- одну чи кілька інструкцій VBA, які виконують розрахунки на основі значень аргументів функції;
- управляючі конструкції для організації розрахунків, вибору та циклів.

9. Можна використовувати вбудовані функції:

- функції VBA пишуться з великої букви, аргументи відокремлюються комами;
- якщо у VBA немає потрібної функції, її викликають з MS Excel.

10. Для того, щоб повернути результат обчислень формули, яка включає в себе функцію, наприкінці треба присвоїти результат змінній, ім'я якої збігається з ім'ям цієї функції.

Для **введення створеної функції в будь-яку формулу робочого листа** треба вставити її стандартним способом; користувацькі функції містяться в категорії "Определенные пользователем".

### *Управляючі конструкції розгалуження, вибору та циклів*

Для організації **простого розгалуження** використовують оператор розгалуження IF-Then.

```
If <умова> Then
    <блок операторів_1>
Else:
    <блок операторів_2>
End if
```

Для організації **складного розгалуження** використовують ланцюжок операторів розгалуження IF-Then:

```

If <умова1> Then
<блок операторів_1>
Else If <умова2> Then
    <блок операторів_2>
    .....
Else If <умоваN> Then
    <блок операторів_N>
Else
    <блок операторів_Else>
End if
End if
End if

```

Для організації циклів використовують оператори **циклів з лічильником** (For-To), **циклів масиву** (For Each-In) та оператори **циклів з умовою** (Do –While та Do-Until).

**Цикли з лічильником** використовують, коли зазначену послідовність операторів необхідно повторити певну кількість разів. При цьому використовується спеціальна тимчасова змінна, яка називається **лічильником**, значення якої збільшується або зменшується з кожним кроком виконання циклу на зазначену величину (*крок*, або *присіст*)

```

For <лічильник> = <нач знач> To <кін знач>
    крок<блок операторів>
Next [<лічильник>]

```

**Цикли масивів** використовують, коли вказану послідовність операторів необхідно виконати для кожного елемента масиву, який є аргументом функції з типом Range.

```

For Each <елемент масиву> In < масив>
    <блок операторів>
Next [<елемент масиву>]

```

**Цикли з умовою** використовують, коли невідомо, скільки разів треба повторити виконання блоку операторів, які утворюють *тіло циклу*.



Такі цикли продовжують роботу, поки зазначена умова виконується (Do While), або ж навпаки, поки вказана умова не буде виконана (Do Until). Умову можна перевіряти або перед виконанням блоку операторів циклу, або після.

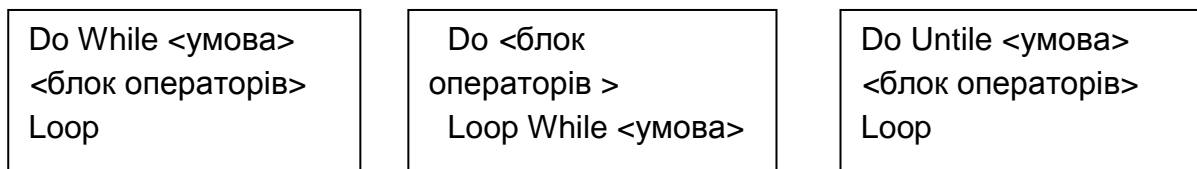


Рис. 1.11. Перелік циклів, які прововжують роботу

Використовуючи макроси, можна значно (в десятки, а то і в сотні разів) збільшити ефективність роботи з додатком. Крім виконання набору вбудованих в додаток команд, макрос дозволяє проводити обробку зовнішніх файлів, завантаження і передачу файлів і даних через Інтернет, читання і зміна налаштувань операційної системи. У більшості випадків макрорекодер (режим запису дій користувача у вигляді макросу) видає код, що потребує доопрацювання. Але, тим не менше, і такий макрос значно скорочує час, потрібний на виконання рутинних операцій.

## 2. Загальні відомості про системи управління базами даних

При розв'язанні багатьох статистичних задач сьогодні широко використовуються системи управління базами даних (СУБД) реляційного типу. У цих СУБД реалізується реляційна модель даних – зображення їх у табличному вигляді. Рядок такої таблиці еквівалентний запису файла бази даних (БД), а графа – полю запису. Доступ до елемента даних здійснюється з допомогою зв'язку потрібного рядка (запису) з потрібною графою (полем).

Існує два типи систем управління БД: *файлами* і *реляційними базами даних*. У *системах управління файлами (File management systems)*, іноді їх ще називають *плоскими файловими базами даних*, дані вводяться і зберігаються без індексування, а при пошуку і створенні вихідних форм обробляються послідовно. Тому ці системи недостатньо гнучкі при маніпулюванні даними. *Системи управління реляційними базами даних (Relational database management systems)* дозволяють

користувачам маніпулювати даними більш вільно завдяки збереженню всіх даних тільки один раз і опису зв'язків між ними. Зв'язки становлять загальні елементи даних, таких як прізвище покупця або номер його чекової книжки. Дані кожного набору інформації можна знайти, відобразити і відновити на підставі даних іншого набору.

До складу багатьох СУБД, призначених для роботи на персональних комп'ютерах, входять три основні компоненти: командна мова, інтерпретуюча система або компілятор для перетворення команд до виду, придатного до виконання, і засоби взаємодії користувача із СУБД (інтерфейс користувача).

Командна мова слугує для виконання і забезпечення потрібних операцій з даними, дає змогу створювати прикладні програми, екранні форми введення і виведення інформації, структури файлів БД і маніпулювати даними. Структура і можливості мови визначають вигляд і можливості конкретної СУБД.

У СУБД можна виконувати команди по одній або групами в автоматичному режимі, заздалегідь записаними у спеціальний (програмний) файл. Команди мови СУБД записуються в текстовій формі, близькій до звичайної мови. Для того, щоб персональний комп'ютер міг виконати таку команду, її треба перетворити на команду, що виконується комп'ютером.

Структура і вміст компонентів СУБД визначає її призначення і коло потенційних користувачів.

Останні версії сім'ї СУБД **Access 2010** для **Windows**, розроблені фірмою Microsoft, відповідають вимогам користувачів.

Зберігання інформації – одна з найважливіших функцій комп'ютера. Одним з найпоширеніших засобів такого зберігання є бази даних.

**База даних** – це файл спеціального формату, який містить інформацію, структуровану певним чином. Комплекс програмних засобів, які призначені для створення структури нової бази, наповнення її вмістом, редагування вмісту і візуалізації інформації, називають **системою управління базою даних (СУБД)**. У даній роботі ми познайомимося з СУБД Microsoft Access і створимо за допомогою неї базу даних (файл).

База даних може мати такі об'єкти: таблиці, запити, форми, звіти. Створення бази даних завжди повинно починатися з розробки структури її таблиць. Таблиці – це основні об'єкти бази даних, у них зберігаються

дані. У табличній структурі адрес даних визначається перетином рядків та стовпців. У базах даних стовпці називаються *полями*, а рядки – *записами*. Поля утворюють структуру бази даних, а записи складають інформацію, яка в ній міститься.

Поля – це основні елементи структури бази даних. Вони мають свої властивості. Від властивостей полів залежить, які типи даних можна вносити у поле, а які ні, а також те, що можна робити з даними, які містяться у полі.

Основні *властивості полів* таблиць баз даних на прикладі СУБД Microsoft Access:

- **ім'я поля** – визначає, як варто звертатися до даних цього поля при автоматичних операціях з базою (за замовчуванням імена полів використовуються як заголовки стовпців таблиць);
- **тип поля** – визначає тип даних, що можуть міститися в даному полі (текстовий, числовий, дата/час, поле MEMO, грошовий, лічильник, логічний, поле об'єкта OLE, гіперпосилання);
- **розмір поля** – визначає граничну довжину (у символах) даних, що можуть розміщатися в даному полі, основна властивість текстового поля;
- **формат поля** – визначає спосіб форматування даних в комірках, що належать полю, наприклад, для типу поля дата/час – короткий формат 19.06.11;
- **маска введення** – визначає форму, в якій вводяться дані в поле (засіб автоматизації введення даних);
- **підпис** – визначає заголовок стовпця таблиці для даного поля (якщо підпис не зазначений, то як заголовок стовпця використовується властивість Ім'я поля);
- **значення за замовчуванням** – те значення, що вводиться в комірки поля автоматично (засіб автоматизації введення даних);
- **умова на значення** – обмеження, використовуване для перевірки правильності введення даних (засіб автоматизації введення, що використовується, як правило, для даних, що мають числовий тип, грошовий чи тип дата/час);
- **повідомлення про помилку** – текстове повідомлення, що видається автоматично при спробі введення в поле помилкових даних (перевірка помилковості виконується автоматично, якщо задана властивість);

- **обов'язкове поле** – властивість, що визначає обов'язковість заповнення даного поля при наповненні бази.

Вказані властивості полів відносяться, в основному, до полів текстового типу. Поля інших типів можуть мати або не мати ці властивості, але можуть додавати до них і свої. Наприклад, для даних, що представляють дійсні числа, важливою властивістю є кількість десятичних знаків.

Microsoft Access працює з такими типами даних, які можуть міститися у полі:

- **текстовий** – тип даних, який використовується для збереження звичайного неформатованого тексту обмеженого розміру (до 255 символів);

- **поле MEMO** – спеціальний тип даних для збереження великих обсягів тексту (до 65 535 символів);


- **числовий** – тип даних для збереження числових даних. Воно також має свій розмір (байт, ціле, довге ціле, дійсне). Залежно від розміру, поле має свої властивості, наприклад, для дійсного необхідного вказати розмір десяткової частини числа;

- **дата/час** – тип даних для збереження календарних дат і поточного часу. Мають свої формати (повний, довгий, середній, короткий);

- **грошовий** – тип даних для збереження грошових сум. У цьому випадку комп'ютер відображає числа разом з грошовими одиницями;

- **лічильник** – спеціальний тип даних для унікальних (не повторюваних у поле) натуральних чисел з автоматичним нарощуванням. Природне використання – для порядкової нумерації запису;

- **поле об'єкта OLE** – спеціальний тип даних, який призначений для зберігання об'єктів OLE, наприклад мультимедійних (картинки, музичні кліпи та відеозаписи).

З організаційної точки зору у роботі з будь-якою базою даних є два режими: 1) за допомогою якого проектується структура об'єкта – **режим конструктора** та 2) експлуатаційний (користувацький) – **режим таблиці**. Перший режим запускається за допомогою команди "**Создать**" об'єкт (таблицю, запит, звіт, форму) за допомогою конструктора або якщо об'єкт створено за допомогою кнопки . Цей режим дозволяє

правити не зміст, а структуру. Другий режим запускається подвійним кліком по назві об'єкта та дозволяє наповнювати об'єкт змістом.

При створенні структури таблиць одне поле (чи одну комбінацію полів) треба назначити **ключовим**. З ключовими полями комп'ютер працює особливо. Він перевіряє їх унікальність та швидше виконує сортування за такими полями.

Комп'ютер і комп'ютерні програми слід сприймати як робочий інструмент, який щоденно використовується в роботі. До пакету Microsoft Office входить набір таких програм-інструментів, скориставшись якими, користувач зможе автоматизувати виконання щоденних завдань. Тому корисно знати, для яких завдань необхідно використовувати СУБД Access. Бази даних призначені для збереження інформації. Розглянемо кілька типових прикладів інформації, яка може зберігатися в комп'ютерних БД:

- інформація про працівників установи;
- облік продукції;
- робочі плани;
- відомості про покупців;
- облік продажів;
- постачальники;
- накладні.

Зверніть увагу на те, що в усіх наведених прикладах підкреслюється функція зберігання і маніпулювання даними, а не розрахунки. У БД можна виконувати багато фінансових, статистичних та інших розрахунків, але останні здійснюються не так швидко, як, наприклад, в електронних таблицях. Наприклад, СУБД – не найкраща програма для автоматизації розрахунків податку з прибутку: для розв'язування цієї задачі більш придатна електронна таблиця. Однак якщо користувачеві необхідно відслідковувати, аналізувати і зберігати інформацію про податки за кілька років, то БД краще знадобиться для збирання та збереження цієї інформації.

Працюючи з Access, користувач може розв'язувати такі завдання:  
розбивати дані на логічно пов'язані між собою частини;  
вводити, змінювати і знаходити необхідні дані;  
знаходити підмножини даних за заданими умовами;  
створювати форми і звіти;

автоматизувати виконання стандартних завдань;  
 графічно встановлювати зв'язки між даними;  
 вставляти рисунки у форми і звіти;  
 створювати особисті, готові до роботи з БД програми, які вміщують меню, діалогові вікна і командні кнопки.

Кожна БД Access складається з таких об'єктів: таблиць, запитів, форм, звітів, макрокоманд і модулів. У табл. 2.1 наведені основні об'єкти в Access.

Перш ніж приступати до створення таблиць, форм і звітів, користувачеві необхідно витратити якийсь час на проектування БД. Далі подані деякі важливі поради, яких слід дотримуватися при розробці БД.

Почніть з аналізу існуючих БД (створених вручну або за допомогою комп'ютера). Прогляньте форми і звіти, що використовуються. Визначте джерело даних (якщо це комп'ютер, з'ясуйте, чи можна імпортувати дані).

Таблиця 2.1

### Об'єкти БД

Об'єкт	Опис
Таблиця	Вміщує дані в форматі таблиці; схожа на електронну таблицю
Запит	Вибирає дані з таблиці на підставі заданої умови. Запити дозволяють бачити дані з кількох таблиць в одній
Форма	Відображує дані з таблиці або запиту на підставі описаного користувачем формату. Форми дозволяють проглядати, редагувати і друкувати дані. У формі можна відображати дані, взяті з кількох таблиць або запитів
Звіт	Відображує і друкує дані з таблиці або запиту на підставі описаного користувачем формату. У звіті дані редагувати не можна. Звіти можуть вміщати дані, взяті з кількох таблиць або запитів
Макрос	Автоматизує стандартні дії на підставі вибраних користувачем команд і подій
Програмний модуль	Автоматизує складні операції, які не можна описати за допомогою макрокоманд. Програмні модулі – це процедури, написані мовою програмування Access Basic

Спілкуйтеся з іншими людьми, які використовують вміщену в БД інформацію, говоріть проблеми і побажання, які виникають у зв'язку з цим. Прогляньте задачі, які розв'язуються за допомогою БД (і обміркуйте ті, які необхідно розв'язати в майбутньому), наприклад, створення щотижневих звітів, експортування даних, сортування і аналіз (обробка).

Ідентифікувавши дані і задачі, які необхідно розв'язувати, поділіть їх на групи (наприклад, розмежуйте інформацію про клієнтів та інформацію про платіжні документи). Ці групи потім стануть таблицями.

Визначте типи даних, які будуть зберігатися в кожній таблиці (таблиця клієнтів, наприклад, може вміщувати прізвища клієнтів, їх адреси і номери телефонів). Ці групи даних таблиці називаються *полями*.

Прогляньте загальні елементи таблиць (прізвище клієнта може бути загальним елементом таблиці клієнтів і таблиці накладних). Ці загальні елементи називаються *ключовими полями*.

Обміркуйте оформлення форм і звітів.

Визначте умови вибірки для запитів.

Обміркуйте, як автоматизувати стандартні задачі БД, такі як виконання запитів і друкування звітів.

Обміркуйте проблеми безпеки даних, такі як стратегія створення копій, поділ даних і обмеження доступу при роботі в мережі.

Access зберігає таблиці, запити, форми і звіти в одному файлі. Перед створенням таблиці необхідно створити БД, в якій будуть зберігатися таблиці та інші об'єкти Access.

Під час запуску Access з'явиться діалогове вікно (рис. 2.1).

Головне вікно Access – це робоча область, де користувач маніпулює різними об'єктами Access (наприклад, таблицями). У цьому вікні виберіть БД, з якою будете працювати, або створіть нову. Створити нову БД можна з допомогою одного з двох методів.

Активізуйте пункт "**Новая база данных**" і натисніть мишкою на кнопці "**ОК**", щоб відкрити нову БД.

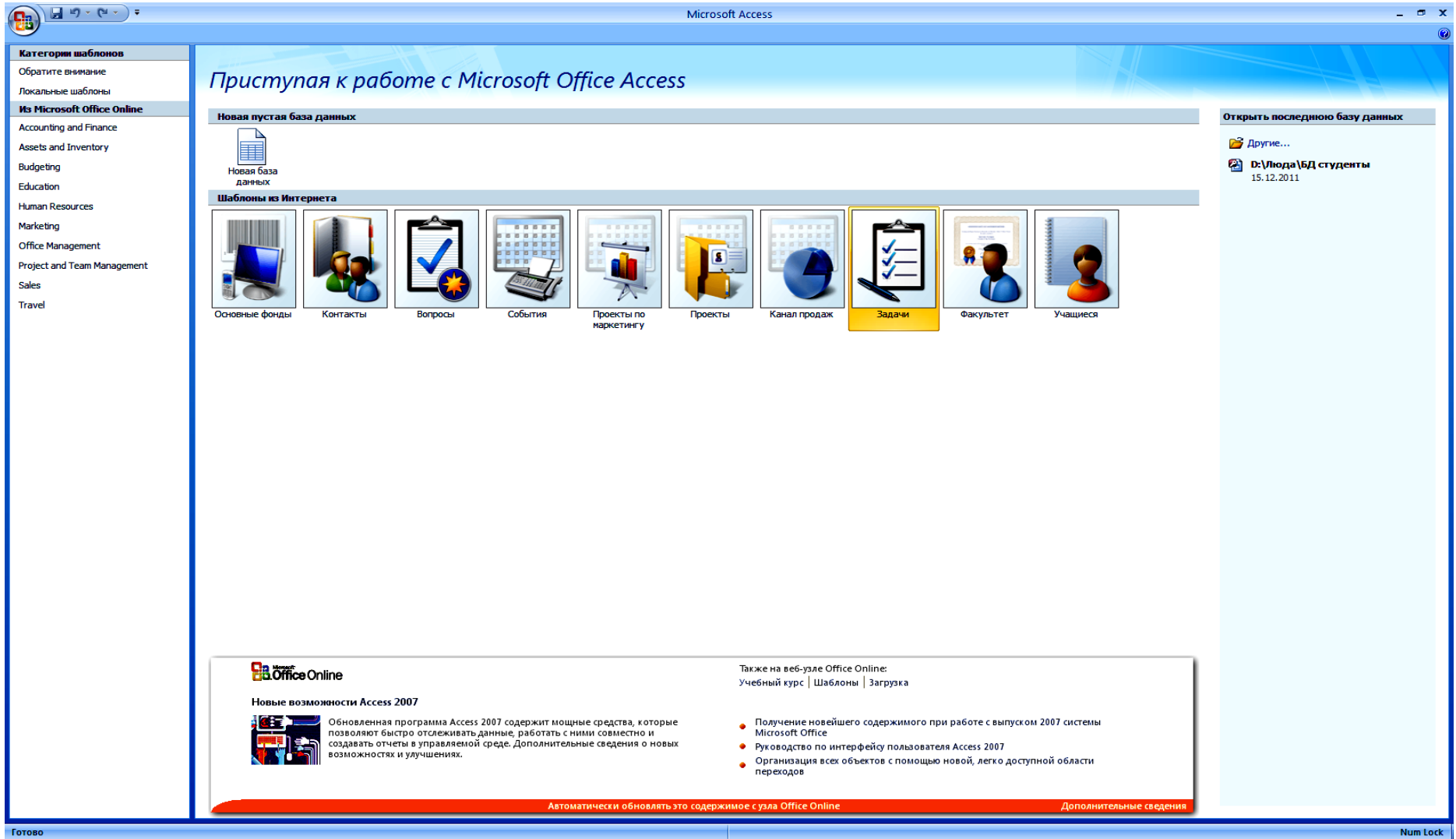


Рис. 2.1. Початкове діалогове вікно Access



Активізуйте пункт **"Запуск мастера"** і натисніть мишкою на кнопці **"ОК"**, щоб запустити **"Мастер базы данных"**.

Створити або відкрити БД можна, закривши діалогове вікно (натиснувши мишкою на кнопці **"Отмена"**). Для створення БД виконайте команду **"Файл"** та її опцію **"Создать базу данных"**, або натисніть мишкою на кнопці **"Создать базу данных"** на панелі інструментів.

Для відкриття раніше створеної БД виконайте команду **"Файл"** та її опцію **"Открыть базу данных"**, або натисніть мишкою на кнопці **"Открыть базу данных"** на панелі інструментів.

Якщо буде активізовано пункт **"Новая база данных"**, програма зробить запит про ім'я створюваної БД (рис. 2.1). У вікно **"Имя файла"** необхідно ввести ім'я файла.

Наприклад, необхідно розробити інформаційну систему "Споживчий кошик" для відділу управління обстежень домогосподарств Головного управління статистики, яка дозволить спростити зберігання та редагування інформації для аналізу рівня життя населення. Як відомо, рівень життя населення залежить від співвідношення доходів сім'ї та цін на продукти споживання. Тому база даних повинна складатися з 3-х таблиць: "Продукти", "Доходи" та "Споживання". Таблиці мають таку структуру:

"Продукти":

код продукту (ключове поле);  
найменування;  
одиниці виміру.

"Доходи":

місяць, рік (ключове поле);  
сукупний дохід.

"Споживання":

код продукту (ключове поле);  
місяць, рік (ключове поле);  
кількість;  
ціна.

У вікні "Файл новой базы данных" відкрийте папку для зберігання БД, визначте файлове змістовне ім'я та натисніть кнопку "Создать". Наступним кроком необхідно створити об'єкти бази даних, для чого натиснути кнопку "Создать" та спочатку створити таблицю БД. Для опису структури таблиці необхідно задати поле й вказати його тип (рис. 2.2).

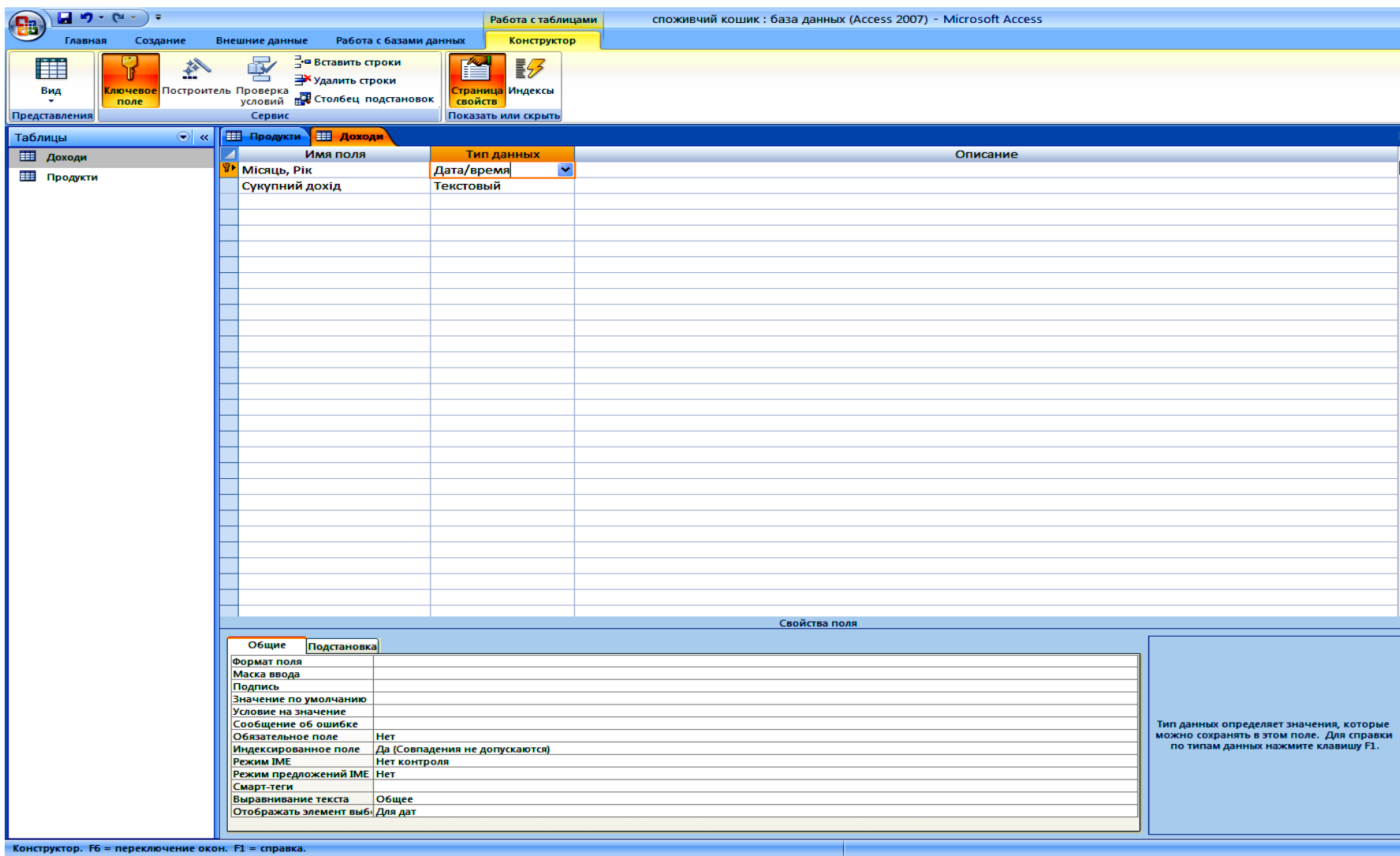


Рис. 2.2. Вікно таблиці в конструкторі

Після цього в розділі "Свойства поля" задати необхідні властивості поля. Наприклад, для текстових полів основною властивістю є його довжина. Наступним кроком необхідно задати ключове поле. Для цього виділяємо строку поля таблиці та помічаємо його як ключове, натиснувши на панелі інструментів "Ключевое поле". У лівій частині рядка з'явиться значок ключа. Роль ключового поля можуть грати декілька полів таблиці, для цього необхідно виділити ці поля натиснувши клавішу Ctrl та натиснути кнопку "Ключевое поле". Завершити опис структури таблиці закриттям вікна опису структури. На питання, чи зберігати зміни макета структури таблиці, натискаємо кнопку "Так". Задайте змістовне ім'я таблиці відповідно до її призначення та натисніть "ОК". Якщо ключове поле не було задано, то Access запропонує завдати ключове поле. Якщо воно не заплановане, то відповідайте "Ні". Якщо відповісти "Так", то Access автоматично вставляє ключове поле типу лічильник. У цій ситуації більш кращою є відміна закриття вікна опису структури та завдання ключового поля вручну.

Заповніть таблиці записами. Для цього потрібно виділити необхідну таблицю у вікні БД та натисніть "Открыть". Введіть дані в поля таблиці (10 – 20 записів, залежно від сенсу таблиці). Для ключових полів, тип яких не є лічильником, розробіть систему кодів. Код повинен бути коротким та певною мірою відображати вміст запису. Найпростіший код для ключа – лічильник – може бути вставлений автоматично.

Після заповнення таблиця виглядає таким чином (рис. 2.3).

Якщо початкова інформація оброблялась до цього в іншій програмі управління БД чи програмою розрахунку електронних таблиць, то можливо перенести вже створені набори даних в Access – таблиці, що заощадить час, особливо якщо мова йде про досить великі БД, що містять в собі велику кількість записів.

Для імпорту набору даних в меню "**Файл**" вибираємо підменю "**Внешние данные**" команду "**Импорт**". Ця команда переносить дані з текстового файла, програми обробки електронних таблиць в Access-таблицю. Вона також дозволяє перенести об'єкти з іншої БД Access в активну БД.

Дані можливо імпортувати, якщо вони зберігаються в файлах одного із таких форматів:

- MS Access;
- Текстовий;

код продук	найменування	одиниці виміру
101	хліб	шт.
102	молоко	літр
103	яйця	шт.
104	м'ясо	кг.
105	вода	літр
106	овочі	кг.
107	фрукти	кг.
108	ковбасні вироби	кг.
109	риба	кг.
110	макаронні вироби	кг.
111	оція	літр
112	цукор	кг.
113	солодощі	кг.
114	сир	кг.
115	кава	г.
116	чай	г.
117	масло	кг.
118	кефір	літр
119	сік	літр
120	бобові	кг.
121	рис	кг.
122	гречка	кг.
123	консерви	шт.
124	сметана	літр
125	сіть	кг.

Рис. 2.3. Внесення інформації в таблицю

- Microsoft Excel;
- Paradox;
- Exchange;
- Outlook;
- Документи HTML;
- dBASE;
- dBASE IV;
- dBASE 5;
- Таблиці Lotus 1-2-3;
- Документи XML;
- ODBC Database.

Припустимо, що дані про споживання товарів зберігались в Excel - таблиці. З метою подальшої обробки її необхідно імпортувати в Access. Після виклику команди імпорту (рис. 2.4) потрібно знайти файл, який необхідно імпортувати та місце призначення даних (рис. 2.5).

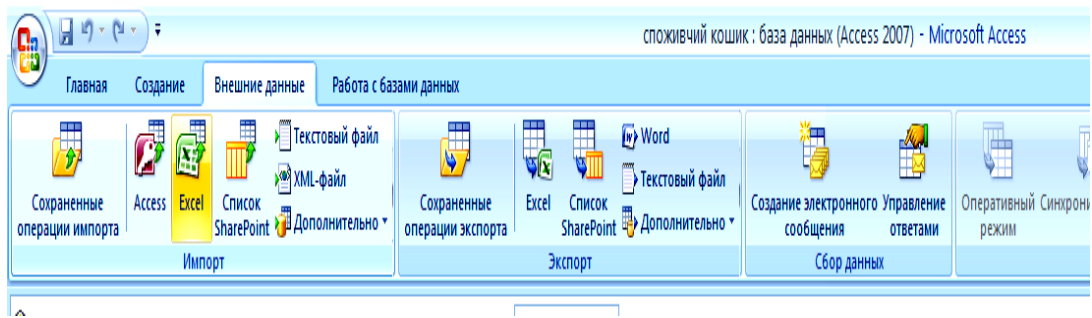


Рис. 2.4. Команда імпортування даних

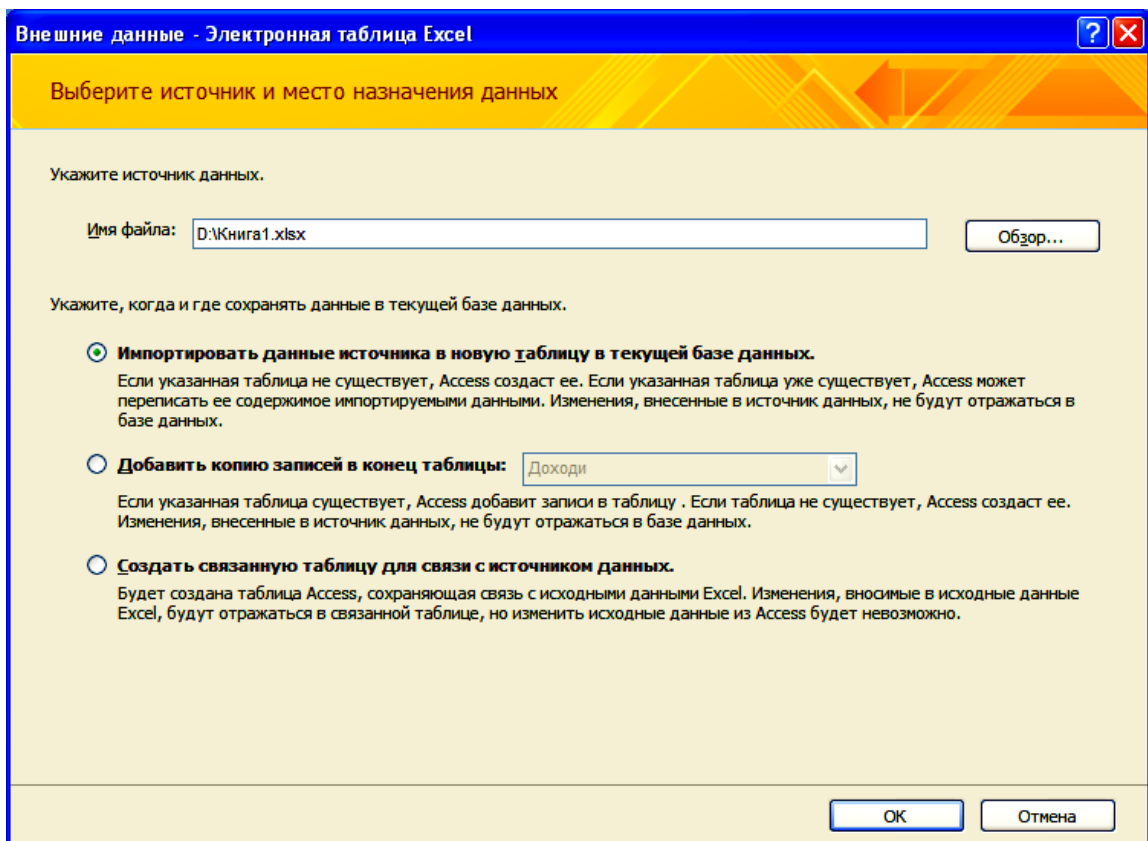


Рис. 2.5. Вікно вибору даних для імпортування

Далі автоматично електронна таблиця перетвориться в таблицю Access за декількома етапами. Спочатку відкриється діалогове вікно, в якому відобразяться дані імпортованої таблиці Excel (рис. 2.6).

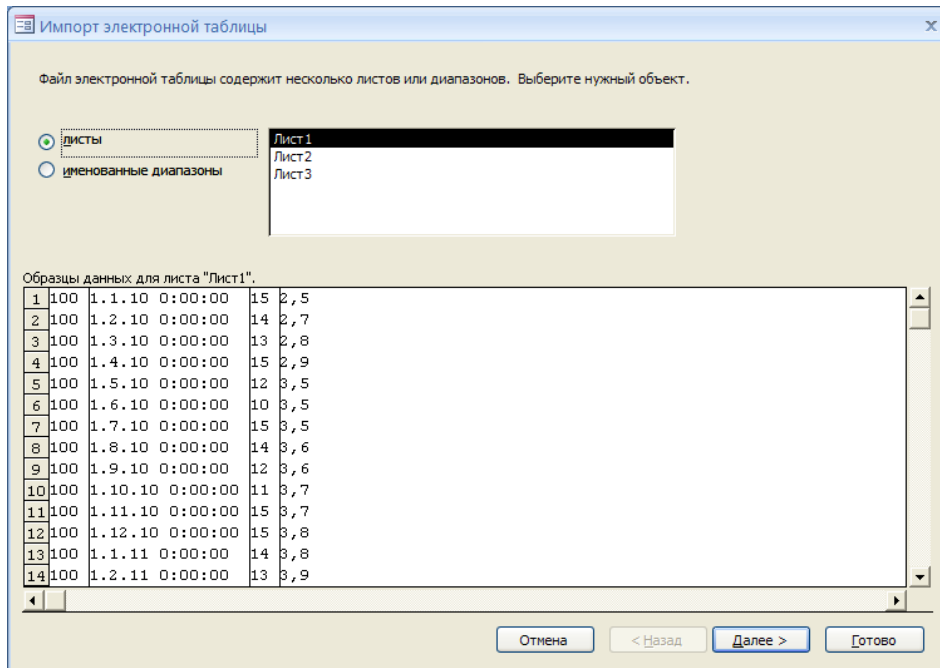


Рис. 2.6. Диалогове вікно імпортованої таблиці

Після цього в наступному вікні пропонується вибрати в якості імен полів першу строку таблиці Excel, яка містить заголовки стовпців (рис. 2.7).

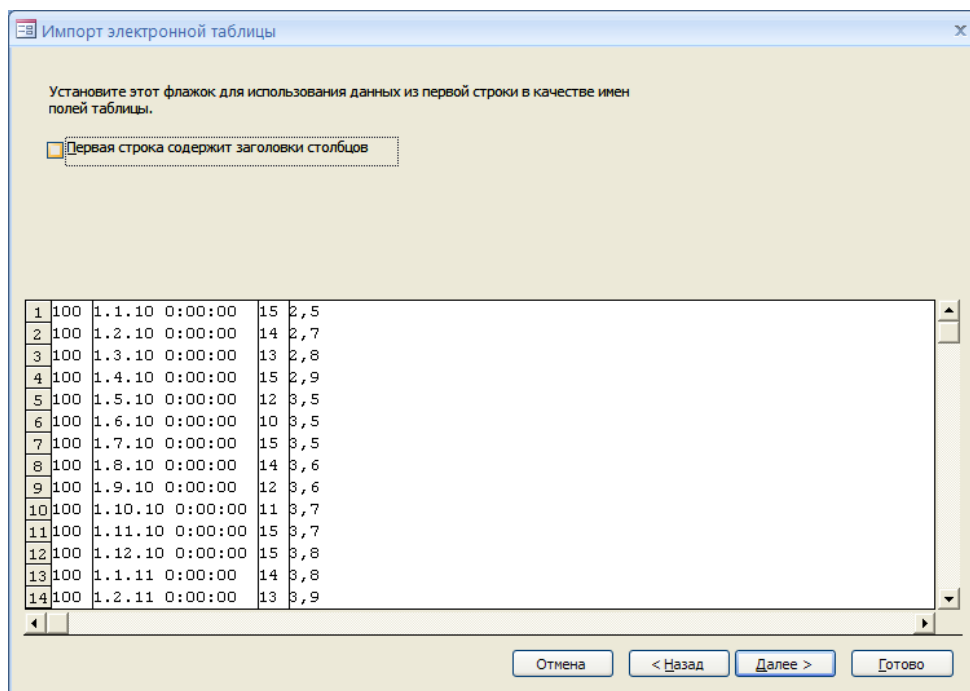


Рис. 2.7. Вікно майстра імпорту

Для цього необхідно встановити опцію **"Первая строка содержит заголовки столбцов"**. Імена полів, а також інші

характеристики можливо відкоригувати у наступному діалоговому вікні, для чого натиснути кнопку "Далее" (Рис. 2.8). У цьому вікні можливо змінити ім'я поля, та, якщо можливо, його тип, а також визначити, чи слід його імпортувати, встановивши опцію "Не импортировать поле". Крім того, можливо визначити, чи потребує індексації це поле.

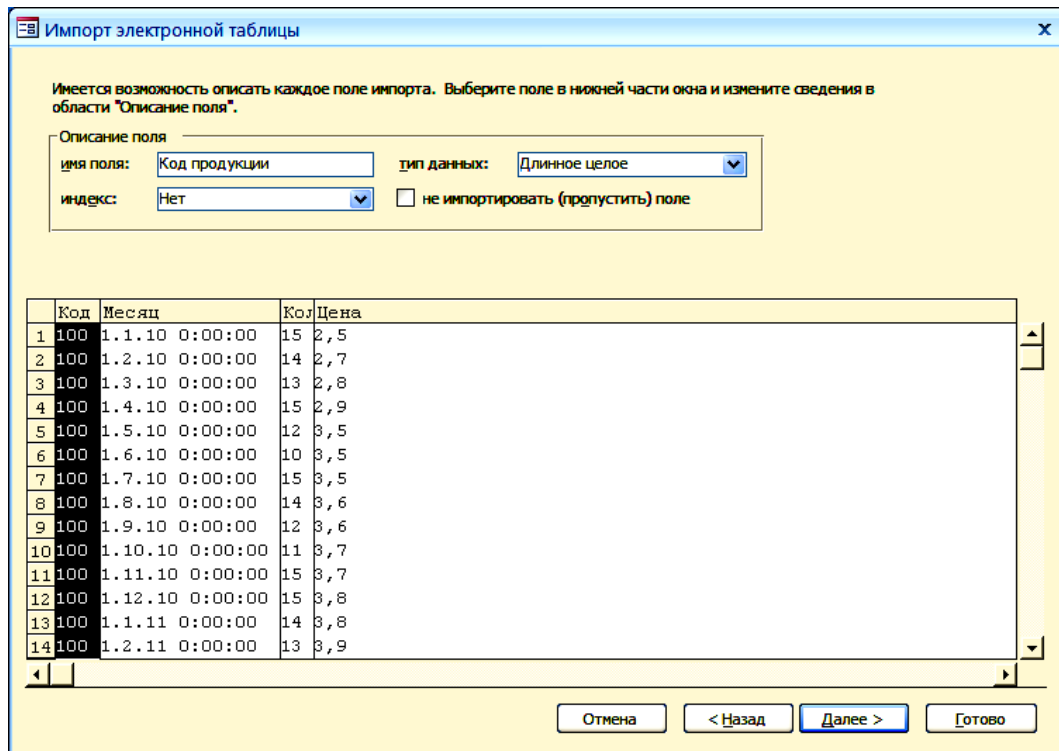


Рис. 2.8. Вікно вибору полів та зміни їх формату

У наступному вікні (рис. 2.9) необхідно встановити первинний ключ для таблиці.

Первинний ключ встановлюється, виходячи з таких запропонованих варіантів:

"автоматически создать ключ" – програма Access сама додає поле з первинним ключем;

"определить следующий ключ" – поле, яке виконуватиме роль первинного ключа з переліку доступних полів вибирає користувач;

"не создавать ключ" – первинний ключ не присвоюється.

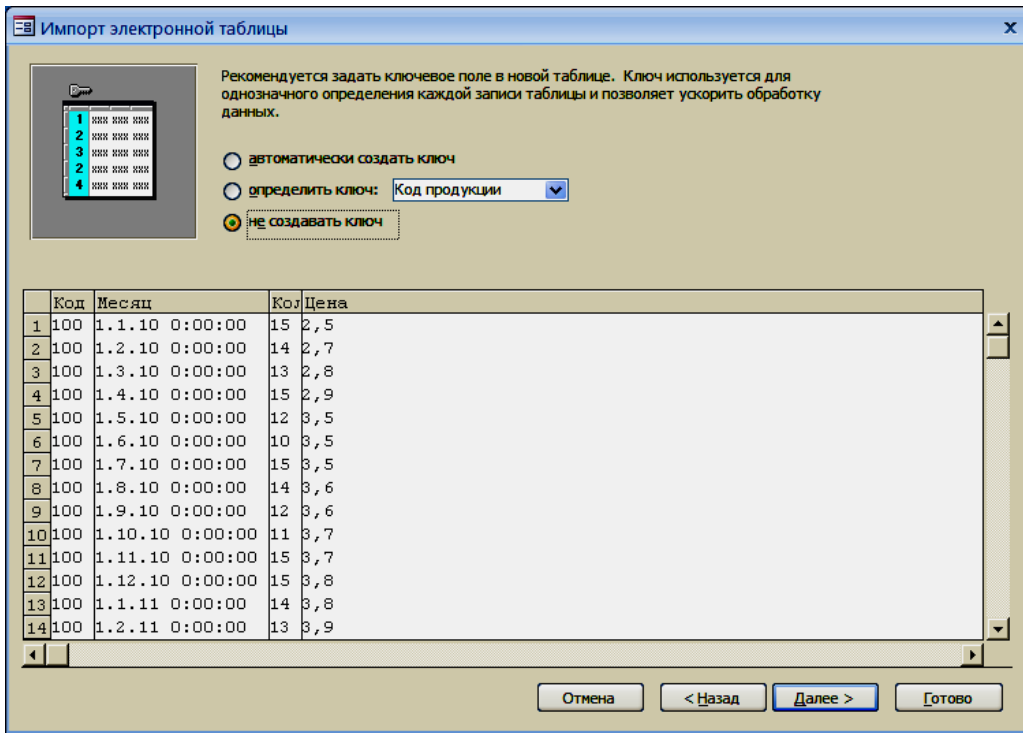


Рис. 2.9. Вибір первинного ключа при імпортуванні даних

Далі треба перейти до останнього діалогового вікна майстра та завдати ім'я імпортованій таблиці (рис. 2.10).

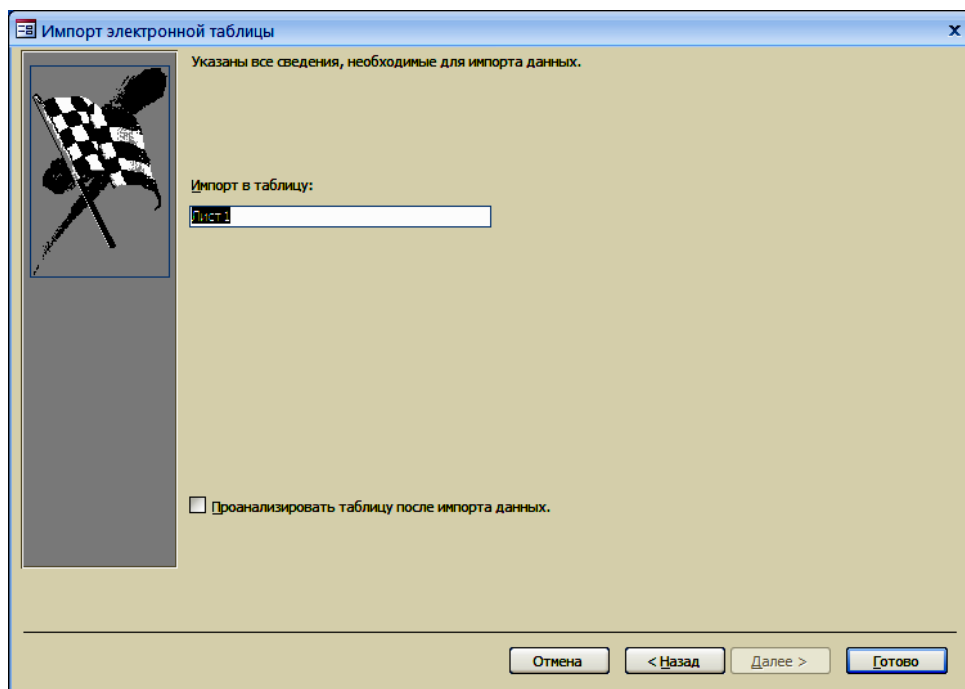


Рис. 2.10. Вікно завдання імені поля



У цьому вікні можливо встановити опцію, яка дозволить після імпорту таблиці запустити програму аналізу даних ("**Проанализировать таблицу после импорта данных**"). Отже, імпортована таблиця має такий вигляд (рис. 2.11)

Код продукту	Місяць, рік	Кількість	Ціна
100	01.01.2010	15	2,5
100	01.02.2010	14	2,7
100	01.03.2010	13	2,8
100	01.04.2010	15	2,9
100	01.05.2010	12	3,5
100	01.06.2010	10	3,5
100	01.07.2010	15	3,5
100	01.08.2010	14	3,6
100	01.09.2010	12	3,6
100	01.10.2010	11	3,7
100	01.11.2010	15	3,7
100	01.12.2010	15	3,8
100	01.01.2011	14	3,8
100	01.02.2011	13	3,9
100	01.03.2011	12	3,9
100	01.04.2011	11	4
100	01.05.2011	15	4

Рис. 2.11. Імпортована таблиця "Споживання"

Далі створюємо таблиці за аналогією або імпортуємо з інших програм.

Окрім імпорту даних з Access, можливо експортувати таблиці в текстові файли, електронні таблиці, файли інших прикладних програм управління БД, а також в іншу БД. Для виконання експорту маркуйте експортовану таблицю (наприклад "Доходи") та активізуйте в меню "Внешние данные" команду "Экспорт". Вікно, яке при цьому з'явиться, виглядатиме аналогічно вікну імпорту (рис. 2.5). У ньому слід вибрати формат файла чи бази даних, в яку необхідно експортувати дані. Формати експорту ідентичні форматам імпорту. Залежно від того, який формат був обраний користувачем, майстер експорту, що запускається автоматично або в діалозі з користувачем, проводить експорт обраної таблиці. Оскільки процедура експорту ідентична процедурі імпорту, детально на ній зупинятися не будемо.

Також існує можливість зберегти таблицю в існуючий БД, але у вигляді іншого об'єкта. Для цього таблицю необхідно виділити та

активізувати команду "Сохранить как" головного меню. У діалоговому вікні збереження необхідно ввести ім'я копії файлу, а також вибрати, в який об'єкт при збереженні буде перетворена таблиця. Наприклад, копію таблиці "Доходи" збережемо як запит (рис. 2.12).

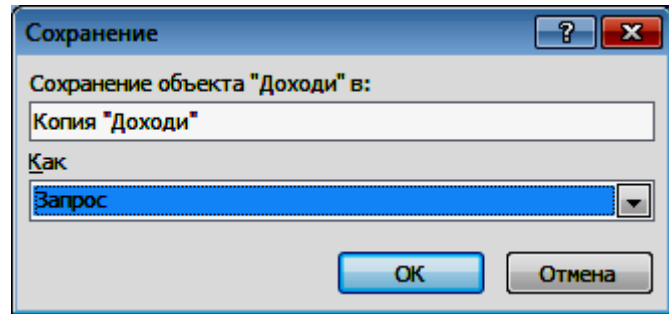


Рис. 2.12. Вікно перетворення таблиці в запит

Наступним кроком є встановлення зв'язків між створеними таблицями. Для цього необхідно натиснути кнопку "Схема даних" на основній панелі інструментів. На вкладці "Таблицы" у вікні "Добавление таблицы" виділіть таблиці БД та натисніть "Добавить" (рис. 2.13).

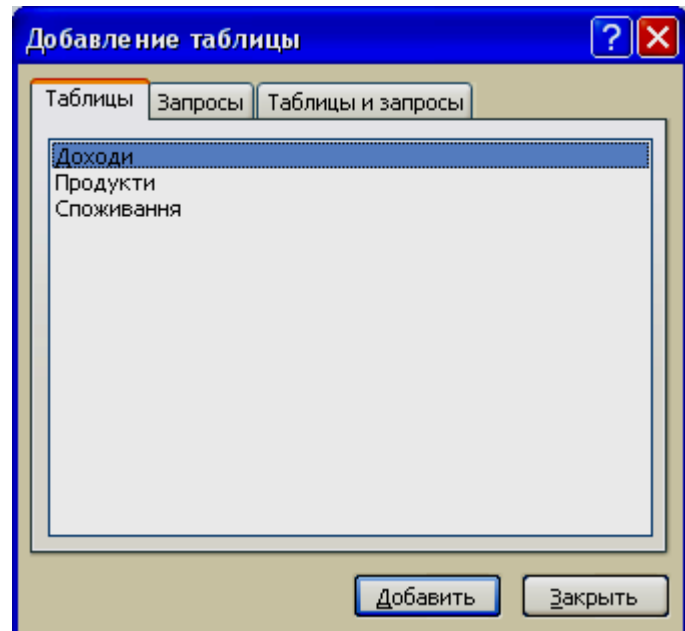


Рис. 2.13. Вікно додавання таблиці

У результаті чого з'явиться вікно з таблицями, які входять до БД (рис. 2.14).

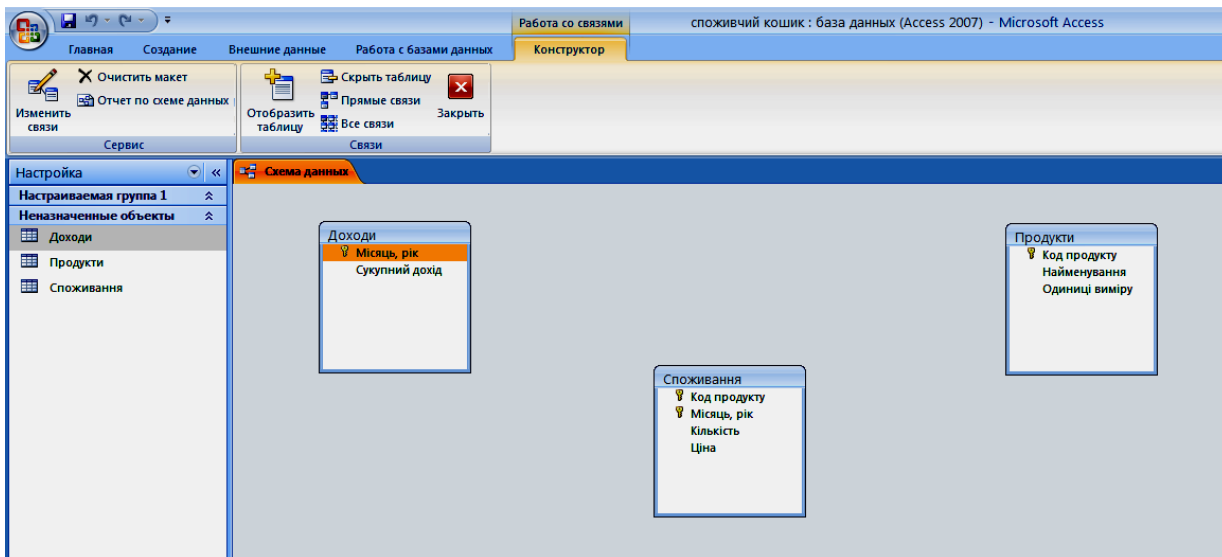


Рис. 2.14. Вікно схеми даних

У вікні "Схема данных" зв'язки встановлюються по черзі між парами таблиць. Спочатку необхідно визначити головну таблицю в парі. Потім виділити в схематичному зображенні головної таблиці ключове поле та перетягнути його на поле у підпорядкованій таблиці, з яким встановлюється зв'язок. У вікні "Изменение связей" встановіть позначку "Обеспечение целостности данных" (рис. 2.15)

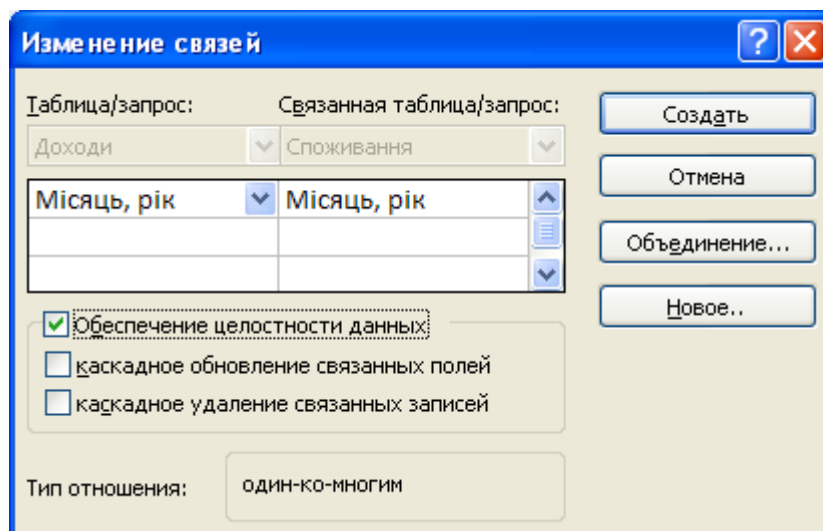


Рис. 2.15. Вікно встановлення зв'язків між таблицями

Отже, в основному вікні схеми даних зображення таблиць будуть пов'язані лініями, що показують тип зв'язку (рис. 2.16). Бази даних, які мають зв'язані таблиці, називають реляційними базами даних.

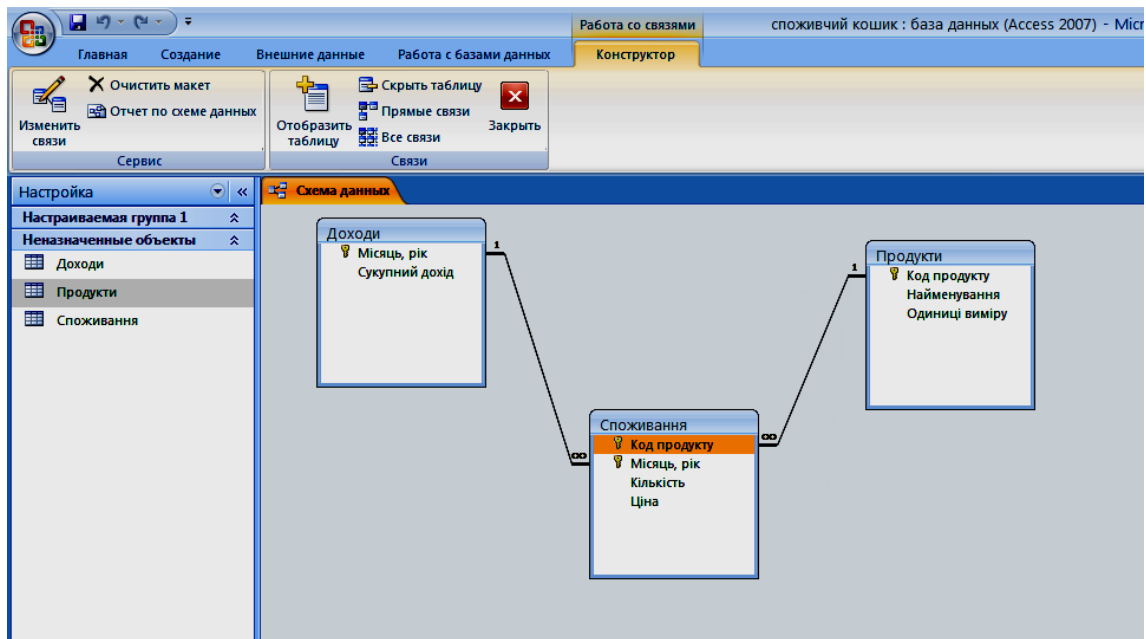


Рис. 2.16. **Схема даних**

Зв'язок між таблицями має дві характеристики: ступінь зв'язку та клас належності таблиці до зв'язку. Значення цих характеристик можуть бути визначені з аналізу зв'язку між полями таблиць. Ступінь зв'язку показує, скільки полів однієї таблиці можуть бути зв'язані з кожним екземпляром іншої таблиці. Виділяють три ступені зв'язків між таблицями у схемі:

"один до одного" (1:1) – одному значенню поля в одній таблиці відповідає тільки одне значення поля в іншій;

"один до багатьох" (1:∞) – одному значенню поля в одній таблиці відповідає декільком (одній чи більше) значень в іншій, а кожне поле другої таблиці може бути зв'язане тільки з одним полем першої таблиці;

"багато до багатьох" (∞:∞) – кожне поле першої таблиці може бути зв'язане з декількома екземплярами другої таблиці та навпаки.

Клас належності таблиці до зв'язку може бути обов'язковим та необв'язковим. При обов'язковому класі належності кожне поле таблиці повинно бути зв'язане з іншим полем, при необв'язковому класі належності не вимагається, щоб кожне поле таблиці було зв'язане з яким-небудь полем іншої таблиці. Іншими словами, в БД зберігається інформація про всі поля таблиць.

Отже, тільки найпростіші бази даних містять тільки таблиці. Якби інформація зберігалась би у таких простих структурах, то для роботи з

нею можливо було б обійтись без спеціальних систем управління базами даних. Основні переваги систем управління базами даних реалізуються при роботі не з окремими таблицями, а з групами взаємопов'язаних таблиць. Тому наступним кроком в створенні БД є формування запитів.

Залежно від подальшого використання або способу отримання даних, отриманих за допомогою запиту, в Access існує 4 типи запитів.

Запити на вибірку – відбирають потрібні поля даних з записів, що задовольняють заданій умові з однієї або декількох таблиць та/або інших запитів БД.

Перехресні запити формують дані з таблиць і запитів у вигляді електронної таблиці (матриці). У такій таблиці заголовки рядків утворюються з унікальних значень одного поля, заголовки стовпців – з унікальних значень іншого поля, а на їх перетині обчислюються які-небудь сумарні значення для відібраної множини записів.

Запити на зміну діляться на 4 види:

на створення нової таблиці;

на додавання нових записів в таблицю;

на видалення відібраних записів з таблиці;

на зміну значень яких-небудь полів у відібраних записах таблиці.

Запити з параметрами – це такі запити, в яких критерії відбору задаються користувачем при кожному запуску запиту. Тобто запит з параметрами можна отримати з будь-якого з трьох попередніх типів запитів, ввівши в нього параметри для відбору записів. При запуску такого запиту на екран виводиться вікно, в яке слід ввести значення параметра вибірки, для кожного з параметрів.

Наприклад, для створення запиту на вибірку необхідно у вікні БД перейти на вкладку "Создание" та натиснути "Мастер запросов". Потім додати всі таблиці в запит – у верхній частині вікна запиту повинна бути відображена схема даних. Після цього необхідно додати в строку "Поле" в нижній частині вікна запиту імена полів таблиць БД, що відбираються цим запитом. Для цього натисніть в клітинці першого рядка та зі списку оберіть потрібне поле. У переліку зірочками відмічені назви таблиць, а без них – поля таблиць. Рекомендується обирати поля. Порядок проходження полів зліва направо повинен бути за зменшенням важливості інформації (рис. 2.17).

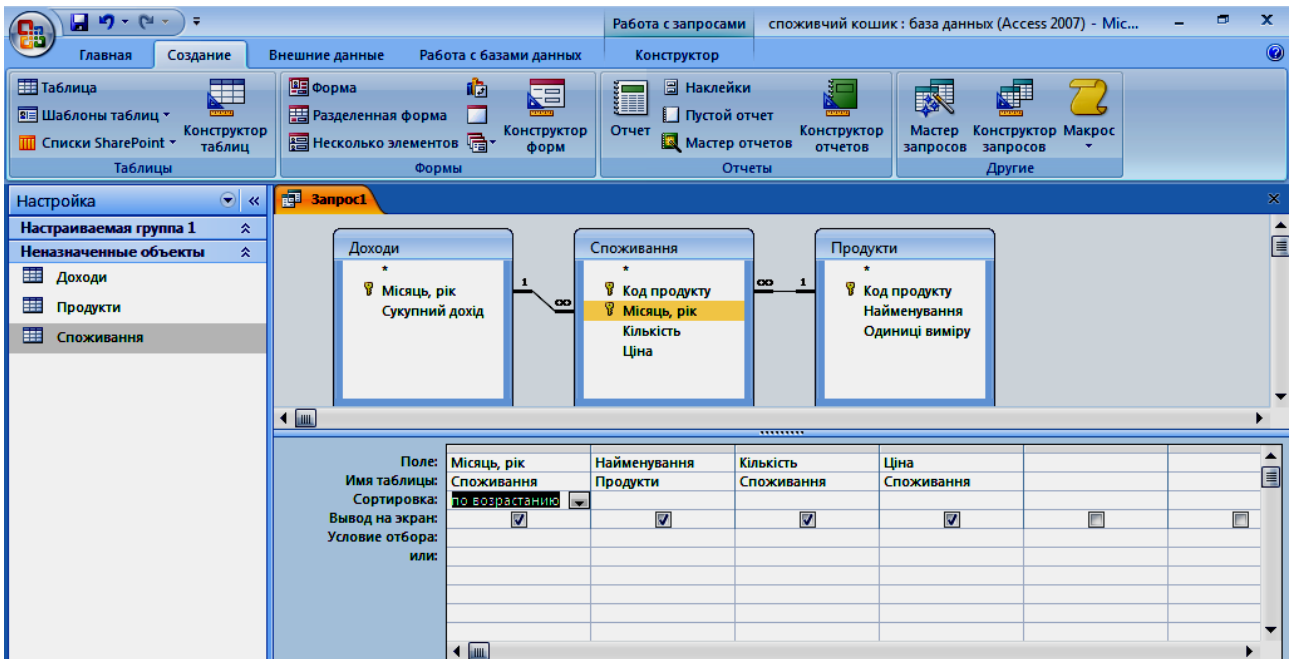


Рис. 2.17. Вікно формування запиту

Після збереження запиту та завдання йому змістовного ім'я можливо переглянути запит, натиснув "Открыть" у вікні БД (рис. 2.18).

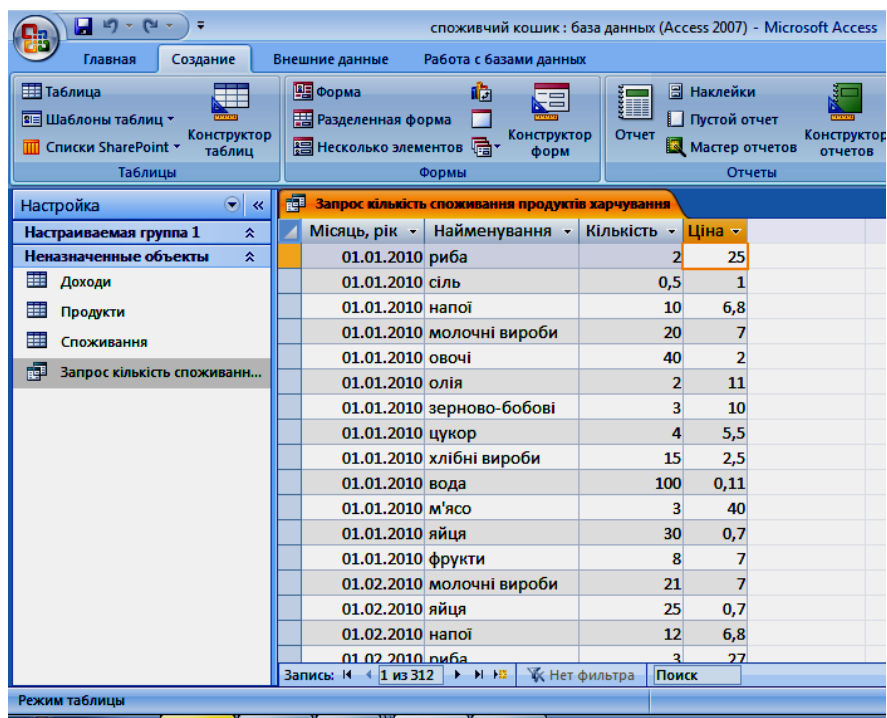


Рис. 2.18. Запит на споживання продуктів харчування

Наступним об'єктом бази даних є форма. Форми – це об'єкти, за допомогою яких у базу вводять нові дані, переглядають наявні, а також здійснюють редагування даних, розміщених у таблиці. Форма подібна до бланка чи карточки. Прикладами форми є будь-яким бланком, карточкою з досьє чи бібліографічною карточкою на книжку в бібліотеці. Одна форма містить дані лише з одного запису.

Access дозволяє швидко створювати форми з допомогою "Мастера форм". Можна також розробляти форми вручну, відкривши пусту форму і перетягнувши об'єкти (наприклад, поля, текст і графіку) в потрібне місце форми. Можна поєднувати автоматизовану і ручну розробку. Наприклад, за допомогою Мастера форм можна створити основу форми, а потім відкрити форму в режимі "Конструктор форми" і внести зміни вручну.

Якщо користувачеві потрібно управляти процесом створення форм і витратити на це менше зусиль, то краще скористатися допомогою "Мастера форм". Як і інші майстри Access, "Мастер форм" задає низку запитань про форму, яка створюється, а потім будує форму залежно від відповідей користувача. На формі можливо розташувати:

- поля типу OLE з картинками, фотографіями тощо;

- елементи керування: кнопки, перемикачі тощо;

- надписи: заголовки форми, рубрик (а також розрисувати форму чи задати фоновий рисунок-заставку);

- обчислювальні поля (це також елементи керування) для відображення результатів обчислень, виконаних на базі наявних полів;

- закладки (багатосторінкові форми, де поля групують за змістом на різних закладках);

- підпорядковані форми тощо.

Наприклад, для створення головної форми необхідно через пункт меню "Создание" вкладку "Конструктор форм" створити макет форми. Потім нанести на неї необхідні кнопки для роботи з БД. При нанесенні кнопок на форму програма запропонує вибрати дії, які потрібні виконуватися при натисканні створених кнопок. Для головної форми це може бути "Открытие формы в режиме редактирования" (для кнопок, які відкривають форму чи команду) та "Выход из приложения" (для кнопки завершення роботи) та ін. (рис. 2.19).

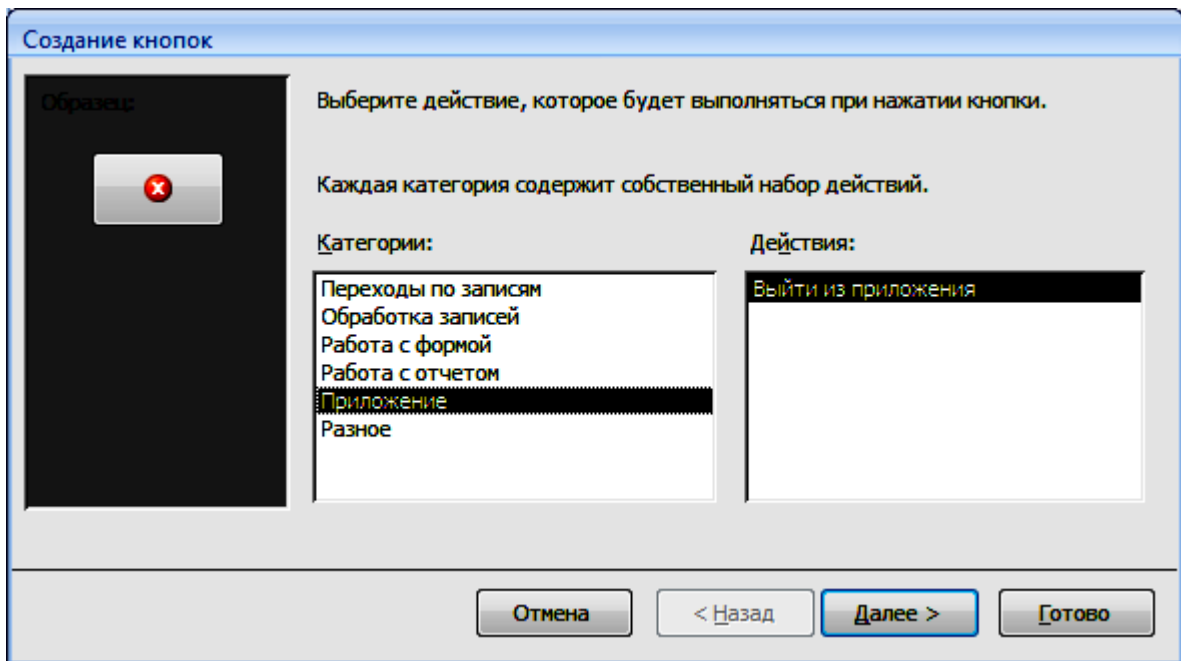


Рис. 2.19. **Діалогове вікно створення кнопок форми**

Отже, форма відкриття запиту може мати такий вигляд (рис. 2.20).

Діаграми в Access – це тип форми. Microsoft Graph дозволяє використовувати такі типи діаграм: з областями, лінійчаті, гістограми, графіки, колові, кільцеві, лепесткові і крапкові. Кожен тип діаграм має підтипи. Далі подані різні типи діаграм.

"Диаграмма с областями" подає співвідношення величин протягом деякого часу. Хоча така діаграма подібна графіку, вона, скоріше за все, демонструє обсяг змін, а не зміни у часі та їх швидкість. Для того, щоб зосередити увагу на швидкості змін, краще використовувати графік.

"Линейчатая диаграмма" відтворює окремі значення у визначений момент часу або відтворює співвідношення компонентів. У цій діаграмі категорії розташовуються на вертикалі, а значення – на горизонталі. Таким чином, більше уваги приділяється співставленню компонентів і менше – змінам у часі.

"Гистограмма" відтворює зміни на протязі деякого часу або відтворює співвідношення компонентів. *Гистограмма* схожа з *лінійчатою діаграмою*, але категорії розташовуються на горизонталі, а значення – по вертикалі.

"График" відтворює тенденції або реальну зміну даних за рівні відрізки часу. Він схожий з діаграмою областями, але більшою мірою відтворює зміни в часі та їх швидкість, а не обсяг.



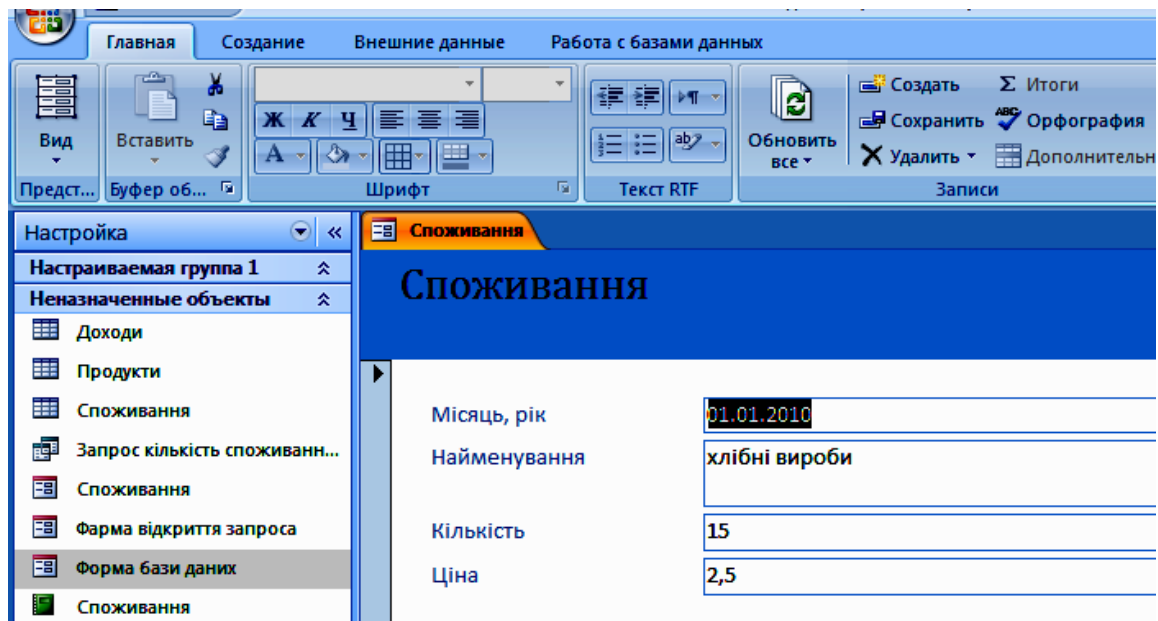


Рис. 2.20. Форма на відкриття запити

Якщо бажано показати тенденції або зміни даних за неоднакові відрізки часу, то більш підійде точкова діаграма.

"Круговая диаграмма" відтворює співвідношення частей і цілого. Цей тип діаграм є найбільш підходящим, коли є потреба виділити який-небудь суттєвий компонент.

"Кольцевая диаграмма" схожа з *круговою*. Основна відмінність кільцевої діаграми від кругової, крім кругового отвору по середині, є те, що вона може відтворювати декілька рядів даних.

"Лепестковая диаграмма" (або діаграма типу "радар") подає зміни (або частотність рядів) даних по відношенню до центральної точки і одного ряду до іншого. Кожна категорія має особисту вісь координат, яка виходить з центральної точки. Усі маркери даних одного ряду з'єднуються прямими.

"Точечная диаграмма" відтворює взаємозв'язок (або ступінь цього взаємозв'язку) між числовими значеннями у декількох рядах даних або подає дві групи чисел у вигляді одного ряду в координатах XY. Точкова діаграма відтворює нерівномірність діапазонів або груп даних. Вона часто використовується для відтворення даних наукового характеру.

Також важливим об'єктом бази даних є звіт. Звіти в Access використовуються для подання підсумкової інформації у вигляді, сформованому користувачем. Звіти і форми можуть бути дуже схожі, але вони мають різне призначення. Завдання користувача – визначити, що більш придатне для конкретного випадку.

Оскільки в Access можна друкувати як форми, так і звіти, користувачу іноді дуже важко визначити, що краще роздрукувати.

Тому бажано виходити з таких міркувань: форми призначені для інтерактивного використання, тоді як звіти – для відтворення або друкування окремих і підсумкових значень для груп записів; введення даних та їх редагування виконується тільки у формах. У звітах записи можна відтворювати або друкувати в групах, а форми не дають змоги це робити.

Практично всі звіти, які можна створити в Access, поділяються на три категорії.

"Группы/итоги". У цих звітах, які ще називаються табличними, дані відтворюються за рядками, причому дані кожного поля вміщуються в окрему графу.

"Колоночные отчеты". У цих звітах всі поля відтворюються в одній графі і вирівнюються по лівому краю. Звіти дуже схожі на форми, але вони використовуються тільки для відтворення або друкування даних, а не для їх введення і редагування.

"Почтовые наклейки". Ці звіти використовуються для друкування поштових наклеюк (тобто поштових адрес тих, кому адресована пошта). У Access є функція Мастер отчета, призначена спеціально для створення поштових наклеюк в найрізноманітніших форматах.

Для створення звіту необхідно виконати такі дії. По-перше, визначити джерело даних. Як джерело в звітах Access можна використовувати одну або кілька таблиць або запитів. Щоб відібрати дані і подати їх у визначеному порядку, доцільно використовувати запити (рис. 2.21).

Використовуючи "Мастер отчета" або вікно "Конструктор отчета", можливо розмістити у звіті необхідні поля і мітки. Після завершення розробки звіт можна переглянути на екрані, через головне вікно бази даних (рис. 2.22).

Підводячи підсумки, можна сказати, що в роботі представлені основні рекомендації щодо проектування бази даних, створення таблиць, запитів, форм та звітів. Поетапно розглянуті можливості експорту та імпорту даних БД Access. Лаконічно наведено основний теоретичний матеріал, який може бути використаний при практичних розробках. Також представлений приклад сформованої бази даних з інструкціями користувача, який є наглядним помічником при створенні нової БД.

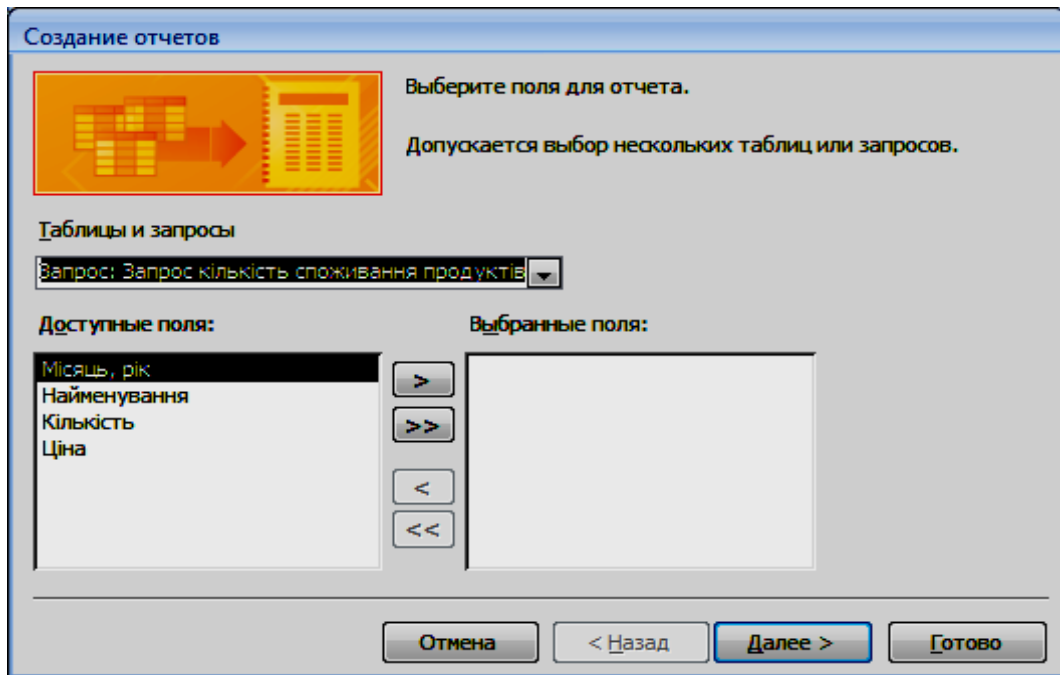


Рис. 2.21. Діалогове вікно створення звіту

При проектуванні студентами унікальної БД, за допомогою якої вирішуватимуться статистичні завдання, необхідно використовувати персональні дані, бажано реальні. Обов'язковим є повне або часткове використання вхідної інформації з Excel при формуванні БД, встановлення взаємозв'язків між вивчаємими командами пакетів та набутих знань щодо роботи в пакеті Excel.

Найменування	Кількість	Ціна	Місяць, рік
вода	100	0,11	01.01.2010
	100	0,2	01.03.2010
	100	0,4	01.12.2010
	100	0,58	01.11.2011
	102	0,58	01.12.2011
	105	0,45	01.07.2011

Рис. 2.22. Сформований звіт

### **3. Завдання для виконання за допомогою Microsoft Excel**

#### ***Завдання 1***

Вивчаючи статистичну сукупність, створіть дві функції користувача, за допомогою яких будуть знайдені середні значення в сукупності, що вивчається. Перевірте, що знову створені формули з'явилися в загальному списку статистичних функцій, викликавши, наприклад майстер функцій в MS Excel.

#### ***Завдання 2***

Маємо дані щодо року та дати народження людини, а також рік і номер місяця сьогоднішнього дня. Визначити вік людини (число повних років). У разі збігу вказаних місяців вважати, що пройшов весь рік.

#### ***Завдання 3***

Написати процедуру, яка в одновимірному масиві визначає мінімальний і максимальний розмір заробітної плати та знаходить її середнє значення. Обсяг вибірки ввести за допомогою оператора InputBox використовуючи наявну інформаційну базу. Значення елементів вибірки визначити датчиком випадкових чисел.

#### ***Завдання 4***

Написати процедуру, яка формує двовимірний масив розмірністю 5X5 за такою схемою для визначення попиту на товар А в мережі супермаркетів

1 2 3 4 5

1 4 9 16 25

1 8 27 64 125

1 16 81 256 625

1 32 243 1024 3125.

#### ***Завдання 5***

Маємо формуляр, що складається з назви книг, прізвища автора, назви видання і року видання. Сформувати список, впорядкований за прізвищем автора. Результат представити у вигляді таблиці в новому файлі. Для введення даних використовувати форму і елементи електронної бази бібліотеки ХНЕУ.

#### ***Завдання 6***

Підприємство з метою визначення економії власних коштів надало завдання своїм системним адміністраторам створити систему моніторингу плати за розмови. Вихідними даними є розмови до А хвилин в місяць по 8 грн, а розмова понад встановлену норму оплачується з

розрахунку 3 грн за хвилину. Написати програму, що обчислює плату за користування телефоном для введеного часу розмов за місяць.

### ***Завдання 7***

Маємо формуляр, що складається з назви книг, прізвища автора, назви видання і року видання. Надрукувати список, впорядкований за роками видання. Результат представити у вигляді таблиці в новому файлі. Для введення даних використовувати форму і елементи електронної бази бібліотеки ХНЕУ.

### ***Завдання 8***

У хмарочосі  $N$  поверхів і всього 1 під'їзд; на кожному поверсі по 3 квартири; ліфт може зупинятися тільки на непарних поверхах. Людина сідає в ліфт і набирає № квартири  $M$ . На який поверх ліфт повинен доставити пасажир?

### ***Завдання 9***

Створіть масив даних для дослідження інвестицій за регіонами України, який має вигляд:  $x_1, x_2, \dots, x_n$ . Визначити регіони з найбільшим обсягом інвестицій. Розмірність масиву і задане число вводити за допомогою оператора MsgBox.

### ***Завдання 10***

Дано дві матриці руху населення в Україні. Зменшити максимальну кількість елементів у кожній з цих матриць в 10 разів.

### ***Завдання 11***

Скласти програму, що імітує застосування комп'ютера в книжковому магазині. Комп'ютер запрошує вартість книг, суму грошей, внесену покупцем: якщо здачі не вимагається, друкує на екрані "дякую": Якщо грошей внесено більше, ніж необхідно, то друкує: "Візьміть здачу" і вказує суму здачі. Якщо грошей недостатньо, то друкує повідомлення про це і вказує недостатню суму.

### ***Завдання 12***

Маємо  $K$  заявок на отримання житлової площі, в кожній вказано термін подачі заяви, площа займаної квартири і кількість членів сім'ї. Сформувати список подачі заяв за датою їх подання. Для введення даних використовувати форму і елементи управління.

### ***Завдання 13***

Маємо анкетні дані співробітників кафедри. Сформувати реєстр у якості ознаки взяти їх посаду та стаж роботи. Отримані результати представити на екрані у вигляді таблиці. Для введення даних використовувати форму і елементи управління.

## **Завдання 14**

Маємо список групи з оцінками екзаменаційної сесії. Підрахувати кількість відмінників в групі і в полі "Є заборгованість" студентам, що мають двійки надрукувати "так", а всім останнім "ні". Результати сформувати у вигляді таблиці, впорядкованої за алфавітом. Для введення даних використовувати форму і елементи управління.

## **4. Завдання для виконання за допомогою Microsoft Access**

### **Завдання 1**

Розробити СУБД "Абітурієнт" для автоматизації роботи регіонального центру оцінювання якості освіти. БД повинна містити три таблиці: анкети абітурієнтів, дані по дисциплінах і результати іспитів.

#### **Анкета включає такі дані про абітурієнта:**

- реєстраційний номер (ключове поле);
- прізвище, ім'я, по батькові;
- дата народження;
- який навчальний заклад закінчили (назва, номер, населений пункт);
- дата закінчення навчального закладу;
- наявність червоного диплому або золотої/срібної медалі;
- адреса проживання (місто, вулиця, номер будинку, телефон).

#### **Дані по дисциплінах містять:**

шифр дисципліни (ключове поле);  
назва дисципліни.

#### **Результати іспитів містять:**

реєстраційний номер абітурієнта;  
шифр дисципліни;  
екзаменаційна оцінка;  
середній бал атестата.

### **Завдання 2**

Розробити статистичну інформаційну систему "Зарплата" для розрахунку середньої погодинної заробітної плати по підприємству. База даних складається з трьох таблиць, що містять відомості про працівників, ставки погодинної оплати і табель відпрацьованих годин.

#### **На кожного працівника зберігаються такі дані:**

особистий номер (ключове поле);  
прізвище, ім'я, по батькові;  
відділ;  
посада;  
розряд.

**Тарифна сітка для погодинної оплати:**

посада (ключове поле разом з розрядом);  
розряд (від 7 до 15);  
ставка (грн/год).

**Табель містить:**

особистий номер;  
місяць;  
кількість годин, відпрацьованих за місяць.

**Завдання 3**

Розробити систему "Оптова база" для статистичного обліку товарів на складі. На оптовій базі зберігаються товари, що отримуються від постачальників. Споживачами є організації, попередньо направляють на базу заявки на товар. Відділ постачання бази веде облік руху товарів.

БД повинна складатися з чотирьох таблиць: "Склад", "Товари", "Заявки" і "Відпустка товарів". Таблиці мають таку структуру:

**Склад:**

код товару;  
кількість;  
дата надходження.

**"Товари":**

код товару (ключове поле);  
назва товару;  
одиниця виміру;

**"Заявки":**

код заявки (ключове поле);  
назва організації;  
код товару;  
необхідну кількість.

**"Відпуск товарів":**

код заявки (ключове поле);  
код товару;  
відпущений кількість;  
дата відпустки товару.

**Завдання 4**

Розробити інформаційну систему "Споживчий кошик" для аналізу рівня життя в родині. Рівень життя залежить від співвідношення доходів сім'ї та цін на споживані продукти.

БД системи містить 3 таблиці: "Продукти", "Доходи" і "Споживання".  
Таблиці мають таку структуру:

**"Продукти":**

код продукту (ключове поле);  
найменування;  
одиниці виміру.

**"Доходи":**

рік, місяць (ключове поле);  
сукупний дохід за місяць.

**"Споживання":**

рік, місяць;  
код продукту;  
кількість;  
ціна.

**Завдання 5**

Розробити статистичну інформаційну систему "Бібліотека" для обліку збереженої і виданої читачам літератури. БД системи складається з трьох таблиць з такою структурою:

**"Книги":**

шифр книги (ключове поле);  
автор;  
назва;  
рік видання;  
кількість примірників.

**"Читачі":**

читацький квиток (ключове поле);  
прізвище та ініціали;  
відділ (адреса).

**"Видача":**

шифр книги;  
читацький квиток;  
кількість примірників;  
дата видачі;  
дата повернення;  
дата фактичного повернення.



## **Завдання 6**

Розробити статистичну інформаційну систему "Єдиний державний реєстр підприємств і організацій України" для підрахунку кількості СПДФО. База даних складається з трьох таблиць:

### **"Підприємства":**

назва;  
податковий номер (ключове поле);  
шифр галузі;  
форма власності.

### **"Відомості про відкриття":**

реєстраційний номер;  
дата відкриття;

### **"Відомості про закриття":**

реєстраційний номер;  
дата закриття;  
відмітка про не виплату податків.

## **Завдання 7**

Розробити статистико-логістичну систему "Перевезення" для транспортного підприємства. На підприємстві є пакет заявок від інших організацій на перевезення різних вантажів. БД системи складається з трьох таблиць: "Транспорт", "Заявки", "Доставка", що мають таку структуру:

### **"Транспорт":**

марка автомобіля;  
державний номер (ключове поле);  
витрата палива (літрів на 100 км).

### **"Заявки":**

код заявки (ключове поле);  
дата;  
пункт відправлення;  
пункт призначення;  
назва вантажу;  
одиниця виміру;  
кількість вантажу.

### **"Доставка":**

дата і час відправлення;  
дата і час повернення;  
державний номер автомобіля;

код заявки;  
одиниці вимірювання;  
кількість фактично перевезеного вантажу;  
пройдена відстань.

### **Завдання 8**

Розробити статистичну інформаційну систему "Сесія" для аналізу успішності на факультеті з конкретної спеціальності. БД системи складається з чотирьох таблиць: "Студенти", "Іспити", "Заліки" і "Дисципліни" за такою структурою:

#### **"Студенти":**

шифр студента (ключове поле);  
прізвище, ім'я, по батькові;  
курс;  
група.

#### **"Іспити":**

шифр студента;  
дата;  
шифр дисципліни;  
оцінка.

#### **"Заліки":**

шифр студента;  
дата;  
шифр дисципліни;  
залік.

#### **"Дисципліни":**

шифр дисципліни (ключове поле);  
назва дисципліни.

### **Завдання 9**

Розробити статистичну інформаційну систему "Навчальне навантаження" для обліку навантаження викладача ВНЗ та автоматизації звіту про виконання навантаження навчальної частини ВНЗ. БД системи складається з 3 таблиць за такою структурою.

#### **Таблиця "Дисципліни":**

код дисципліни (ключове поле);  
назва дисципліни;  
спеціальність;  
курс.

**Таблиця "Види навантаження"** (лекції, лабораторні роботи, семінари, індивідуальна робота, заліки, іспити ін.):

тип заняття (ключове поле);

назва навантаження.

**Таблиця "Виконання навантаження":**

дата;

номер пари;

номер групи (підгрупи);

код дисципліни;

назва теми;

тип заняття (код);

кількість годин.

### **Завдання 10**

Розробити СУБД "Програмне забезпечення відділу статистики фінансів Держкомстату" для зберігання та обробки відомостей про файли на CD-ROM. БД системи повинна складатися з файлів "CD-ROM", "Власники", "Файли" з такою структурою:

**"CD-ROM":**

шифр диска (ключове поле);

назва диска;

дата випуску;

шифр власника.

**"Власники":**

шифр власника (ключове поле);

прізвище, ім'я, по батькові;

адреса;

телефон.

**"Файли":**

назва файлу (пакета);

обсяг у кбайтах;

шифр диска;

пояснення про призначення і властивості.

### **Завдання 11**

Розробити статистичну інформаційну систему "Куратор групи" для спрощення ведення обліку успішності студентів. Система містить три таблиці: "Студенти", "Успішність" і "Дисципліни" за такою структурою.

**"Студенти":**

прізвище, ім'я;  
шифр студента (ключове поле);  
домашня адреса;  
телефон;  
прізвище, ім'я, по батькові батьків.

**"Успішність":**

шифр дисципліни;  
шифр студента;  
оцінка;  
пропуск заняття;  
дата.

**"Дисципліни":**

шифр дисципліни (ключове поле);  
назва дисципліни.

**Завдання 12**

Розробити інформаційну систему "Адміністратор готелю" для спрощення обслуговування клієнтів туристичного бізнесу. БД системи містить чотири таблиці: "Готельні номери", "Проживання", "Заявки на бронювання" та "Інформація про місця".

**"Готельні номери":**

номер кімнати (ключове поле);  
кількість місць.

**"Проживання":**

прізвище, ім'я, по батькові;  
серія та номер паспорта;  
дата початку проживання;  
дата закінчення проживання;  
номер кімнати;  
номер місця.

**"Заявки на бронь":**

номер заявки;  
організація або особа, яка подала заявку;  
заявляється початок проживання;  
заявляється закінчення проживання;  
кількість заявлених місць.

### **"Інформація про місце":**

номер кімнати (ключове поле разом з номером місця);

номер місця;

стан (зайнято, вільно або заброньовано);

номер заявки (якщо заброньовано).

### **Завдання 13**

Розробити інформаційну систему "Товарообіг" для промтоварного магазину. БД системи складається з чотирьох таблиць: "Товари", "Надходження товарів", "Продаж товарів" та "Постачальники" з такою структурою.

#### **"Товари":**

код товару (ключове поле);

найменування товару;

одиниця вимірювання кількості товару.

#### **"Надходження товарів":**

код товару;

дата надходження;

ціна придбання товару за одиницю виміру;

код постачальника.

#### **"Продаж товарів":**

код товару;

місяць продажу;

продане кількість за місяць;

ціна продажу товару.

#### **"Постачальники":**

код постачальника (ключове поле);

назва постачальника;

адреса постачальника;

телефон постачальника.

### **Завдання 14**

Розробити інформаційну систему "Податкова статистика промисловості регіону". БД системи складається з трьох таблиць: "Промислові підприємства", "Сплата податків", "Прибуток", "Види податків" з такою структурою.

#### **"Промислові підприємства":**

код підприємства (ключове поле);

назва підприємства;

адресу підприємства;  
форма власності;  
продукція, що випускається.

**"Сплата податків":**

код підприємства;  
код податку;  
сума податку;  
фактична дата сплати;  
планована дата сплати.

**"Прибуток":**

код підприємства;  
сума прибутку за рік.

**"Види податків" (передбачити 3 види податків):**

код податку (ключове поле);  
назва податку.

## ***5. Звіт з виробничої практики***

За результатами проходження виробничої практики кожний студент повинен представити керівнику практики звіт.

Звіт з виробничої практики має таку структуру:

вступ, який містить мету виробничої практики, перелік завдань, які необхідно виконати. Обсяг вступу 1 – 2 сторінки;

перший розділ, який складається з теоретичних та функціональних положень програмних продуктів, що вивчаються. Обсяг першого розділу 7 – 10 сторінок;

другий розділ, який містить постановку завдання та його реалізацію за допомогою Microsoft Excel та Access. Обсяг другого розділу 10 – 15 сторінок;

висновок, який містить економічну інтерпретацію вирішених завдань за допомогою інструментів вивчаємих програм. Обсяг висновку 1 – 2 сторінки.

Додатки (якщо це необхідно), в яких представляються громіздкі рисунки та таблиці.

***Оцінювання результатів*** проходження практики.

Оцінювання практики здійснюється керівником виробничої практики під час захисту звіту за 12-бальною шкалою і формується за таким правилом.

За виробничу практику студент отримує від 1 до 12 балів:

**1 бал** – сформовано файли з початковими даними, відсутність звіту.

**2 бали** – практичне завдання вирішено в повному обсязі, але немає якісно оформленого звіту.

**3 бали** – практичне завдання вирішено в повному обсязі, але оформлено 30 % звіту.

**4 бали** – практичне завдання вирішено в повному обсязі, але оформлено 70 % звіту.

**5 балів** – практичне завдання вирішено в повному обсязі, але оформлено 80 % звіту.

**6 балів** – звіт виконано в повному обсязі, але студент не відповідає на поставлені запитання під час захисту.

**7 балів** – практичне завдання виконано правильно, якісно оформлений звіт, наведена доцільність використання того чи іншого статистичного інструментарію, але в результаті захисту студент відповідає обґрунтовано не на всі поставлені питання і мають місце неточності у розрахунках, які несуттєво впливають на загальний результат.

**8 балів** – практичне завдання виконано в повному обсязі, якісно оформлений звіт, наведена доцільність використання того чи іншого статистичного інструментарію в аналізі запропонованих ситуацій, студент під час захисту відповідає обґрунтовано на всі поставлені питання, але мають місце недоліки у розрахунках, які не впливають на кінцевий результат.

**9 балів** – практичне завдання виконано в повному обсязі, якісно оформлений звіт, доведено доцільність використання обраного статистичного інструментарію в аналізі запропонованих ситуацій, студент відповідає обґрунтовано на всі поставлені питання під час захисту, але мають місце неточності в обґрунтуванні висновків, які не впливають на загальний висновок.

**10 балів** – практичне завдання виконано в повному обсязі, якісно оформлений звіт, наведена доцільність використання того чи іншого статистичного інструментарію в аналізі запропонованих ситуацій, за результатами розрахунків зроблені аргументовані аналітичні висновки.

**11 балів** – практичне завдання виконано бездоганно, якісно оформлений звіт, доведено засвоєння матеріалу та рекомендованої літератури; використано весь необхідний статистичний інструментарій

для аналізу запропонованих ситуацій, наведено пояснення до розрахунків; зроблено аргументовані аналітичні висновки та доведено вміння використовувати їх для виконання конкретних практичних завдань, розв'язання ситуацій та прийняття зважених управлінських рішень з використанням сучасних інформаційних продуктів.

**12 балів** – глибоке засвоєння інтерфейсів прикладних пакетів та статистичного інструментарію; звіт виконано бездоганно, без жодної помилки, якісно оформлено, проведено порівняльний аналіз того чи іншого статистичного інструментарію для розв'язання практичних ситуацій, за результатами розрахунків зроблені аргументовані аналітичні висновки та узагальнення.

Підсумкова оцінка з дисципліни згідно з Методикою переведення показників успішності знань студентів університету в систему оцінювання за шкалою ECTS конвертується в підсумкову оцінку за шкалою ECTS (табл. 5.1).

Таблиця 5.1

**Переведення показників успішності знань студентів  
у систему оцінювання за шкалою ECTS**

Відсоток студентів, які зазвичай успішно досягають відповідної оцінки	Оцінка за шкалою ECTS		Оцінка за шкалою, що використовується в ХНЕУ	Оцінка за національною шкалою
1	2		3	4
10	відмінне виконання	A	12 – 11	відмінно
25	вище середнього рівня	B	10	
30	взагалі робота правильна, але з певною кількістю помилок	C	9 – 7	добре
25	непогано, але зі значною кількістю недоліків	D	6	задовільно
10	виконання задовольняє мінімальні критерії	E	5 – 4	



1	2		3	4
–	потрібне повторне перекладання	FX	3	незадовільно
–	повторне вивчення дисципліни	F	2 – 1	

## 6. Рекомендована література

1. Бекаревич Ю. Б. Создание реляционной базы данных и запросов в MS Access 2007 / Ю. Б. Бекаревич, Н. В. Пушкина. – СПб. : ГУЭФ, 2010. – 343 с.

2. Вуколов Э. А. Основы статистического анализа : практикум по статистическим методам и исследованию операций с использованием пакетов STATISTICA и EXCEL / Э. А. Вуколов. – М. : Форум: ИНФРА-М, 2004. – 464 с.

3. Гарбер Г. З. Основы программирования на Visual Basic и VBA в Excel 2007 / Г. З. Гарбер. – М. : СОЛОН-ПРЕСС, 2008. – 192 с.

4. Гринченко Н. Н. Проектирование баз данных. СУБД Microsoft Access : учебн. пособ. для вузов / Н. Н. Гринченко, Е. В. Гусев, Н. П. Макаров. – М. : Высшая школа, 2002. – 436 с.

5. Информатика : практикум по технологии работы на компьютере / под ред. Н. М. Макаровой. – М. : Финансы и статистика, 2002. – 255 с.

6. Информатика : учебник для вузов / под ред. Н. В. Макаровой. – М. : Финансы и статистика, 2001. – 315 с.

7. Каратыгин С. Access 2000. Руководство пользователя с примерами / С. Каратыгин. – М. : Лаборатория базовых знаний, 2000. – 376 с.

8. Карчевский Е. М. Access 2010 в примерах / Е. М. Карчевский, И. Е. Филиппов. – М. : "Казанский федеральный университет", 2011. – 118 с.

9. Кашаев С. М. Программирование в Microsoft Excel на примерах / С. М. Кашаев. – СПб. : БХВ-Петербург, 2007. – 320 с.

10. Кошелев В. Е. ACCESS 2007 / В. Е. Кошелев. – М. : Бином-Пресс, 2008. – 217 с.

11. Лялин В. С. Статистика: теория и практика в Excel : учебн. пособ. / В. С. Лялин, И. Г. Зверева, Н. Г. Никифорова. – М. : Финансы и статистика ; ИНФРА-М, 2010. – 448 с.
12. Мак-Дональд М. Access 2007. Недостающее руководство / М. Мак-Дональд. – СПб. : БХВ-Петербург, 2007. – 521 с.
13. Пылькин А. Н. Проектирование баз данных. СУБД Microsoft Access / Пылькин А. Н., Цуканова Н. И. – М. : Горячая линия-Телеком, 2004. – 240 с.
14. Ревунков Г. И. "Базы и банки данных и знаний" / Г. И. Ревунков, Э. Н. Самохвалов, В. В. Чистов. – М. : Высшая школа, 1992.
15. Савельев А. Е. ПК для всех / А. Е. Савельев, Б. А. Сазонов, С. Э. Лукьянов. – М. : Высшая школа, 1991. – 208 с.
16. Саймон Дж. Анализ данных в Microsoft Office Excel: наглядный курс создания отчетов, диаграмм и сводных таблиц / Дж. Саймон. – М. : Издательский дом "Вильямс", 2004. – 528 с.
17. Семенова И. И. Манипулирование данными / И. И. Семенов. – СПб. : СИБАДИ, 2008. – 59 с.
18. Тюрин Ю. Н. Анализ данных на компьютере / Ю. Н. Тюрин, А. А. Макаров ; под ред. В. Э. Фигурнова. – 3-е изд. – М. : ИНФРА-М, 2003. – 544 с.
19. Хоффбыуэр М. Создание и использование баз данных / М. Хоффбыуэр, Шпильманн. – М. : Высшая школа, 1991. – 326 с.
20. Руководство по продукту MicrosoftAccess 2010 [Электронный ресурс]. – Режим доступа : <http://msdn.microsoft.com/en-us/office/cc907897.aspx>.

## Зміст

Вступ	3
1. Загальні відомості про створення та використання макросів	5
2. Загальні відомості про системи управління базами даних	17
3. Завдання для виконання за допомогою Microsoft Excel	44
3 Завдання для виконання за допомогою Microsoft Access	46
5. Звіт з виробничої практики	54
6. Рекомендована література	57



**Методичні рекомендації  
до проведення  
виробничої практики  
для студентів 3 курсу  
напряму підготовки 6.030506  
"Прикладна статистика"  
денної форми навчання**