

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»
АВТОМОБІЛЬНО-ДОРОЖНИЙ ІНСТИТУТ

«ЗАТВЕРДЖУЮ»
Директор АДІ ДВНЗ «ДонНТУ»
М. М. Чальцев
13.03.2012 р.

Кафедра «Інформаційні системи в економіці»

**МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИКОНАННЯ ПРАКТИЧНИХ РОБІТ З ДИСЦИПЛІНИ
«СИСТЕМИ ОБРОБЛЕННЯ ЕКОНОМІЧНОЇ ІНФОРМАЦІЇ»
(ДЛЯ СТУДЕНТІВ СПЕЦІАЛЬНОСТІ 6.030502
«ЕКОНОМІЧНА КІБЕРНЕТИКА»)**

7/46 – 2012 – 02

«РЕКОМЕНДОВАНО»
Навчально-методична комісія
факультету
«Економіка та управління»
протокол № 4 від 18.01.2012 р.

«РЕКОМЕНДОВАНО»
Кафедра «Інформаційні системи в
економіці»
протокол № 5 від 17.12.2011 р.

УДК 681.5(07)

Методичні вказівки до виконання практичних робіт з дисципліни «Системи оброблення економічної інформації» (для студентів спеціальності 6.030502 «Економічна кібернетика») [Електронний ресурс] / укладачі: Н. В. Гуменюк, Д. В. Ніколаєнко. – Електронні дані. – Горлівка: ДВНЗ «ДонНТУ» АДІ, 2012. – 1 електрон. опт. диск (CD-R); 12 см. – Систем. вимоги: Pentium; 32 MB RAM; WINDOWS 98/2000/NT/XP, Linux, FreeBSD; MS Word 2000. – Назва з титул. екрану.

Вказівки містять практичні рекомендації по виконанню основних видів робіт, що передбачені програмою дисципліни «Системи оброблення економічної інформації». Практичні завдання надають можливість набути та закріпити навички роботи в програмах Microsoft Excel та Microsoft Access шляхом виконання типових та індивідуальних завдань курсу.

Укладачі:

Гуменюк Н. В.
Ніколаєнко Д. В., к.т.н., доц.

Відповідальний за випуск:

Ніколаєнко В. Л., к.т.н., доц.

Рецензент:

Дуднікова Н. М., к.т.н., доц.

ЗМІСТ

ВСТУП.....	4
ЗАГАЛЬНІ ПОЛОЖЕННЯ ОФОРМЛЕННЯ ЗВІТУ З ПРАКТИЧНОЇ РОБОТИ.....	5
ПРАКТИЧНА РОБОТА № 1 ПОБУДОВА ІНФОРМАЦІЙНОЇ СИСТЕМИ В СЕРЕДОВИЩІ EXCEL	6
ПРАКТИЧНА РОБОТА № 2 АГРЕГУВАННЯ ДАНИХ В MS EXCEL.....	16
ПРАКТИЧНА РОБОТА № 3 СТВОРЕННЯ ФОРМИ КОРИСТУВАЧА ДЛЯ ПОБУДОВИ БАЗИ ДАНИХ	21
ПРАКТИЧНА РОБОТА № 4 РОБОТА З ШАБЛОНАМИ MS ACCESS....	28
ПРАКТИЧНА РОБОТА № 5 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПЕРСОНАЛОМ	35
ПРАКТИЧНА РОБОТА № 6 РОЗРОБКА БАЗИ ДАНИХ ДЛЯ ОБЛІКУ РУХУ ТОВАРІВ НА СКЛАДІ. ЗАГАЛЬНЕ ПОНЯТТЯ МОВИ SQL, СТВОРЕННЯ ЗАПИТІВ	49
СПИСОК ЛІТЕРАТУРИ	60

ВСТУП

Широкий розвиток комп'ютерної техніки й телекомунікацій дозволив збирати, зберігати, обробляти й передавати інформацію в таких об'ємах і з такою оперативністю, які були не допустимі раніше. Завдяки новим інформаційним технологіям виробнича й невиробнича діяльність людини, її повсякденна сфера спілкування розширюються за рахунок залучення досвіду, знань і духовних цінностей, вироблених світовою цивілізацією. Внаслідок цього економіка в меншій мірі характеризується як виробництво матеріальних благ та в більшій – як створення й поширення інформаційних продуктів і послуг.

Саме тому володіння навичками роботи з сучасними програмами обробки економічної інформації стає однією з основних складових в професійній підготовці фахівця з економічної кібернетики.

Широкий асортимент програмних продуктів, представлений на інформаційному ринку, дозволяє задовольнити потреби практично будь-якої сфери управлінської діяльності. Незважаючи на це, багато фахівців віддають перевагу перевіреним програмам пакету Microsoft Office, надійним, відносно недорогим, зручним і зрозумілим, і, що дуже важливо, сумісним практично з усіма програмами для фінансово-економічної сфери діяльності.

Метою розробки методичних вказівок є надання студентам рекомендацій для закріплення практичних навичок обробки масивів економічної інформації за допомогою програм Microsoft Excel, Microsoft Access.

Запропонований курс розглядає можливості застосування пакету прикладних програм Microsoft Office в процесі здійснення управлінської діяльності. Володіючи основними навичками роботи в Microsoft Office, студенти мають можливість удосконалювати особистий апарат аналізу й прогнозування ситуацій при ухваленні рішень в процесі управління.

Курс освоєння прийомів роботи в програмах пакету Microsoft Office передбачає детальний розбір прикладів завдань і команд, призначених для аналізу економічної інформації. В кінці кожної практичної роботи наведені варіанти індивідуальних завдань та питання для самоконтролю.

ЗАГАЛЬНІ ПОЛОЖЕННЯ ОФОРМЛЕННЯ ЗВІТУ З ПРАКТИЧНОЇ РОБОТИ

Залежно від змісту й особливостей виконання практичної роботи, звіт складають у вигляді тексту, ілюстрацій, таблиць або їх сполучень.

Звіт оформлюють на аркушах формату А4 (210 x 297 мм).

Текст звіту виконується шрифтом Times New Roman 14 pt, через 1,5 міжрядкових інтервали до 30 рядків на сторінці.

Текст звіту слід друкувати, додержуючись таких розмірів берегів: верхній і нижній – 20 мм, лівий – 30 мм, правий – 10 мм.

Під час оформлення звіту необхідно дотримуватись рівномірної щільності, контрастності й чіткості зображення впродовж усього звіту. У звіті мають бути чіткі, нерозпливчасті лінії, літери, цифри та інші знаки. Всі лінії, літери, цифри й знаки повинні бути однаково чорними впродовж усього звіту.

Прізвища, назви установ, організацій, фірм та інші власні назви у звіті наводять мовою оригіналу.

Скорочення слів і словосполучень у звіті – відповідно до чинних стандартів з бібліотечної та видавничої справи.

Абзацний відступ повинен бути однаковим впродовж усього тексту звіту й дорівнювати 1,25 см.

Звіт має містити такі структурні елементи:

- відомості про студента (прізвище, ім'я, група, варіант завдання);
- тему та мету роботи;
- перелік завдань;
- теоретичні відомості до виконання практичних завдань;
- хід роботи, в якому описуються етапи та послідовність виконання завдань практичної роботи;
- висновки по роботі.

ПРАКТИЧНА РОБОТА № 1

ПОБУДОВА ІНФОРМАЦІЙНОЇ СИСТЕМИ В СЕРЕДОВИЩІ EXCEL

Мета: здобути практичні навички з розробки бази даних, провести аналіз, структурування та імпорт даних за допомогою процесора MS Excel.

Завдання:

1. Створити таблицю заданої структури (згідно з варіантом). Введення даних здійснити за допомогою форми вводу.
2. Виконати сортування даних за певним критерієм.
3. Додати до стандартного переліку списків новий список користувача.
4. Виконати фільтрацію даних за декількома умовами з використанням автофільтру та розширеного фільтру.
5. Відкрити новий документ Excel. Скориставшись вбудованим редактором Microsoft Query, створити запит із раніше створеної бази Excel за декількома критеріями та зберегти його результати.




Методичні вказівки до виконання практичної роботи

Створення форми вводу. Електронний процесор Excel розглядає таблицю даних як список, з яким можна здійснювати низку операцій, якщо дані в таблиці організовані за ознакою однорідності, тобто кожен стовпець містить дані одного типу. З точки зору баз даних кожен стовпець є полем даних, а кожен рядок – записом для окремого об'єкта бази даних. Excel розпізнає список автоматично. При цьому перший рядок списку програма розглядає як заголовки стовпців списку (назви полів) й не включає його до результатів обробки списку. Якщо активною є комірка в списку й формат комірок першого рядка відрізняється від формату комірок інших рядків, за допомогою команди «*Форма*» з пункту меню «*Дані*» Excel автоматично створює форму для цього списку (з назвою робочого аркуша), використовуючи перший рядок списку як назви полів форми.

Крім зручного вводу та перегляду даних форма дає змогу шукати записи за певними критеріями. Для цього потрібно натиснути «*Критерії*». Якщо заповнити відповідні поля форми, кнопки «*Назад*» і «*Далі*» показуватимуть лише ті записи таблиці, що задовольняють введеним умовам. Для числових полів можна вводити математичні умови порівняння («=», «<», «>», «<>», «<=», «>=»). У текстових полях можна використовувати символи шаблону («*») – будь-яка кількість будь-яких


припустимих символів, «?» – будь-який один довільний символ).


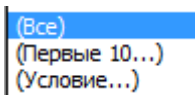
Сортування. Виконується за допомогою команди «Сортування» пункту меню «Дані». Програма автоматично розпізнає таблицю даних (коли активною є комірка всередині списку) й відкриває діалогове вікно «Сортування діапазону».

Якщо перший рядок відформатований інакше або містить об'єкти, що відрізняються від об'єктів інших рядків, у діалоговому вікні пропонується здійснити сортування за назвами цих стовпців. В іншому разі Excel використовує безпосередні назви стовпців, наприклад . Змінити автоматичний вибір програми можна за допомогою функції «Ідентифікувати поля за». Діалогове вікно дає змогу вибрати черговість сортування за стовпцями й напрям сортування кожного із стовпців. Якщо список треба відсортувати тільки за одним полем списку, потрібно зробити активною одну з комірок цього стовпця та скористатися відповідними кнопками панелі інструментів «Стандартна»: за зростанням  й за спаданням .

Створення списку користувача. В Excel можливе сортування за встановленими в програмі списками (дні тижня, місяці року тощо), а також за списками користувача, що створюються за допомогою вкладки «Списки» діалогового вікна «Параметри», яке викликається однойменною командою пункту меню *Сервіс*.

Список користувача можна ввести в поле «Елементи списку» в окремих рядках, а потім натиснути «Додати». Якщо список уже міститься в певному діапазоні комірок робочого аркуша, достатньо ввести цей діапазон у поле «Імпорт списку з комірок» і натиснути «Імпорт». Список користувача, що задає порядок сортування, потрібно вибрати у діалоговому вікні «Параметри сортування» (кнопка «Параметри» у діалоговому вікні «Сортування діапазону») у полі «Сортування по першому ключу».

Фільтрація даних. Для простих умов фільтрації даних найзручніше скористатися командою «Автофільтр» з підменю «Фільтр» пункту меню «Дані». Якщо помістити курсор всередину таблиці даних (списку) й виконати цю команду, то справа від кожного заголовка стовпця з'явиться кнопка зі стрілкою вниз , за допомогою якої можна виконувати різні команди фільтрації списку.

Кнопка  розкриває команди автофільтра, а також список усіх значень комірок стовпця в алфавітному порядку,  що дає можливість з'ясувати весь набір значень і які зі значень програма розглядає як неоднакові. Якщо вибрати один з елементів цього списку,

таблиця даних буде відфільтрована за цим елементом, і на екрані відобразяться лише ті записи, що відповідають заданому фільтру. На відміну від сортування порядок відфільтрованих рядків не порушується. Кожен список, що розкривається, містить елементи *«Пусті»* і *«Непусті»*, які дають змогу відфільтрувати записи, в яких значення зазначеного поля не введені або введені.

Команди автофільтра:

- *«Все»* – відміняє будь-які введені умови фільтрації й повертає таблицю даних до початкового вигляду;

- *«Перші 10...»* – призначена для стовпців з числовими даними. За цією командою відкривається діалогове вікно *«Накладення умови за списком»*;

- *«Умова»* – дає змогу задавати складні критерії фільтрації. За цією командою відкривається діалогове вікно *«Автофільтр користувача»*, в якому визначають одну або дві умови (перший і другий рядки), що можуть бути зв'язані між собою логічними операторами *«І»* (елементи стовпця повинні задовольняти як першу, так і другу умову) або *«Або»* (елементи мають задовольняти одну з цих двох умов).

Якщо стовпець містить числові дані, для них можна встановити звичайні умови порівняння (*«Дорівнює»*, *«Не дорівнює»*, *«Менше»*, *«Менше або дорівнює»* та ін.). Для текстових даних можуть бути застосовані умови *«Починається з»*, *«Не починається з»*, *«Закінчуються на»*, *«Містить»* тощо, а також використані підстановочні знаки: *«?»*, *«*»*.

Для деактивації автофільтра слід повторно виконати команду *«Автофільтр»* з підменю *«Фільтр»* пункту меню *«Дані»*.

У разі застосування *«Розширеного фільтра»* можна задати складні умови відбору записів зі списку. Для розширеного фільтра умови фільтрації для списку вводяться в окремий діапазон комірок. Після виконання команди *«Розширений фільтр»* з підменю *«Фільтр»* пункту меню *«Дані»* відкривається діалогове вікно з такою ж назвою, в якому потрібно встановити діапазони комірок для таблиці даних, умов відбору і результатів фільтрації, а потім натиснути кнопку *«ОК»*.

Для формулювання умов відбору в розширеному фільтрі треба в будь-якому вільному місці робочого аркуша створити додатковий рядок із заголовками стовпців, а наступні кілька рядків використати для введення в них умов відбору записів таблиці даних. Умови відбору в комірках відповідних стовпців можна записувати у кілька рядків. Критерії в одному рядку з'єднуються логічним оператором *«Та»*, в різних рядках – логічним оператором *«Або»*.

У найпростішому випадку, коли зі списку треба отримати тільки записи з одним значенням в одному стовпці (наприклад, виділити лише

записи, що мають значення «Васильківська» для стовпця «Вулиця»), достатньо для діапазону умов вказати тільки дві комірки – з назвою

Вулиця
Васильківська

стовпця й нижче з потрібним значенням. Якщо ж цікавлять два значення з одного стовпця (логічний оператор «Або» – або «Васильківська», або «Антоновича»), в діапазон умов слід нижче додати

Вулиця
Васильківська
Антоновича

ще одну комірку з відповідним значенням.

У разі необхідності можна обмежити відібрані записи додатковою умовою для іншого стовпця (цікавлять з вулиці «Васильківська» лише записи з прізвищем «Петренко», тобто з'єднані логічним оператором «Та»), у тих самих рядках слід додати назву стовпця та відповідне

Вулиця	Прізвище
Васильківська	Петренко

значення. Для іншої ситуації, коли до відібраних записів потрібно додати записи з умовою для іншого стовпця (цікавлять усі з вулиці «Васильківська» та всі з прізвищем «Петренко», тобто зв'язані логічним оператором «Або»), значення для другого стовпця записують у

Вулиця	Прізвище
Васильківська	
	Петренко

наступному рядку.

Щоб повернутися до початкового вигляду списку записів, слід використовувати команду «Відобразити всі» з підменю «Фільтр». При виконанні команди «Розширений фільтр» програма Excel переглядає всю таблицю даних, а не тільки відфільтровану її частину, тому не потрібно виконувати команду «Відобразити всі» перед новою фільтрацією списку записів.

Крім критеріїв точної рівності деякому значенню можна використовувати інші критерії порівняння: «<», «<=», «<>», «>», «>=». Для текстових критеріїв слід враховувати певні правила. Якщо в комірці записати одну літеру, при фільтрації будуть знайдені всі значення, що починаються на цю літеру. За умовою «>Л» після фільтрації залишаться записи, що починаються з літер від «М» до «Я», для «<Л» – від «А» до «К». Крім того символи шаблону («+», «?») обробляються так само, як в автофільтрі.


Використання Microsoft Query. Підсистема Microsoft Query (MS Query), що встановлюється разом із додатком Excel, дає можливість підключатися до зовнішніх джерел даних, баз даних різних форматів, текстових файлів, файлів Excel, відразу аналізувати ці дані та автоматично оновлювати звіти й підсумкові значення в Excel у разі зміни вихідної бази даних.

Програму MS Query доцільно використовувати, якщо джерело даних велике за обсягом, а потрібно проаналізувати лише частину з них або

виконати спеціалізовані завдання, такі як фільтрація рядків і стовпців, сортування даних або об'єднання кількох таблиць перед їх перенесенням в Excel. Це може бути також створення параметричного запиту (запит, перед виконанням якого визначається один або кілька параметрів чи умов).

З точки зору MS Query зовнішні дані – це діапазон даних, розташованих не в Excel, а наприклад, у базі даних, текстовому файлі або таблиці зовнішнього файлу Excel.

Інструментом аналізу даних у MS Query є *запит* – засіб пошуку в джерелі даних тих елементів, що відповідають певному критерію, певній умові. Запити можна створювати самостійно, а можна скористатися майстром.

Для створення запиту за допомогою функції «Майстер запитів» потрібно виконати команду «Створити запит» з підпункту «Імпорт зовнішніх даних» пункту меню «Дані» або натиснути кнопку «Створити запит»  панелі інструментів Microsoft Query та у вікні «Вибір джерела даних» встановити прапорець режиму «Використати майстер запитів».

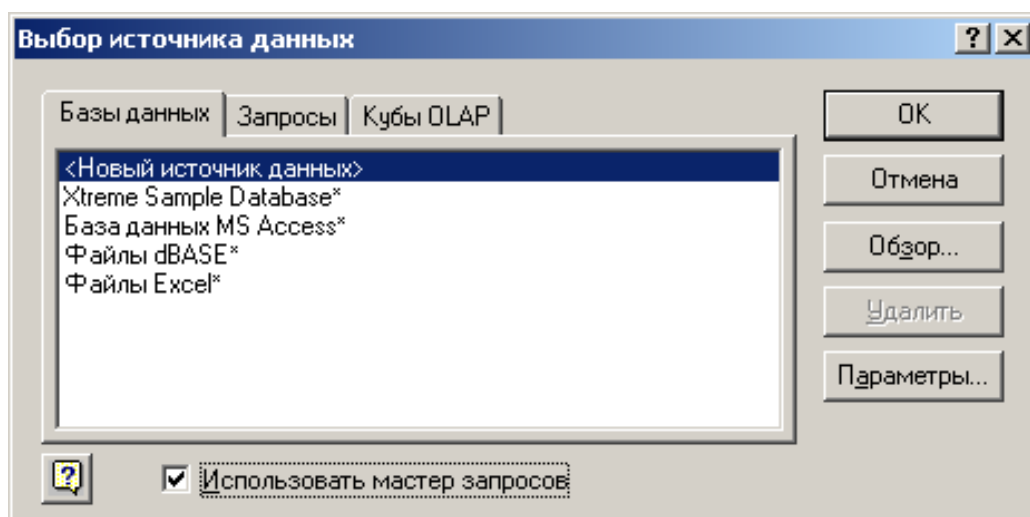





Рисунок 1.1 – Вікно вибору джерела зовнішніх даних

Для обробки таблиць з файлів Excel необхідно вибрати рядок «Файли Excel*». Після натискання кнопки «OK» з'являється напис «Підключення до джерела даних» і відкривається вікно «Вибір книги».

Визначивши файл, що має бути джерелом даних для майбутнього запиту, Excel аналізує дані й у діалоговому вікні «Створення запиту» виводить усі стовпці таблиці даних.

За допомогою кнопки «Перегляд» можна переглянути дані вибраного стовпця та відповідними кнопками із стрілками додати .

вибраний стовпець до запиту, або видалити  вже доданий. Кнопка  дає змогу видалити всі вже вибрані до запиту стовпці.

Наступне вікно майстра запитів призначене для встановлення критеріїв відбору для всіх стовпців запиту. Для кожного стовпця можна встановити кілька умов, кожна з яких може бути з'єднана логічним оператором «Та», «Або». У разі потреби можна вибрати порядок сортування відібраних записів, а потім визначитися з місцем збереження даних, а також використати можливість збереження запиту у файлі з розширенням DQY (файли запитів *.dqy).

Використання MS Query без майстра запитів дає змогу створювати складні види запитів. Якщо у вікні «Вибір джерела даних» вимкнути режим «Використати майстер запитів», після виконання команди відразу запускається програма MS Query з вікном відкриття файлів. У разі вибору файлів Excel відкриється вікно «Вибір книги», в якому слід вибрати потрібний файл.

Вікно запитів MS Query складається з двох частин: ділянки таблиць у верхній частині та ділянки даних у нижній. Якщо вибрати зі списку деякої таблиці поле, в ділянці даних під назвою цього поля буде виведено стовпець його значень, а поряд з'явиться новий список, що розкривається. Для вибору деякого поля будь-якої таблиці крім подвійного клацання можна використати перетягування назви поля з ділянки таблиць до ділянки даних або розкрити список порожнього ще стовпця в ділянці даних, що містить поля всіх доданих таблиць, і вибрати з цього списку потрібне поле.


Ще один спосіб – виконати команду «Додати стовпець» пункту меню «Записи» програми MS Query та використати діалогове вікно «Додавання стовпця». За допомогою елемента «*» у списку полів можна вивести всі поля цієї таблиці.



Після того, як у ділянці даних будуть виведені всі потрібні поля, ділянку таблиць можна сховати за допомогою кнопки «Відображення таблиць» або команди «Таблиці» пункту меню «Вид». Поля записів у ділянці даних можна легко міняти місцями за допомогою миші. Для цього спочатку треба виділити мишею стовпець, клацнувши в зоні заголовку стовпця, а потім перетягнути заголовок стовпця на нове місце. Так само просто можна видалити непотрібний стовпець, попередньо його виділивши, за допомогою клавіші «Delete» або команди «Видалити стовпець» пункту меню «Записи». Також заголовок виділеного стовпця можна змінити, виконавши команду «Змінити стовпець» з того ж пункту меню.


Виведені в ділянці даних записи розташовані у тому ж порядку, в якому вони зберігаються у початковому джерелі даних. Ці записи можна

впорядкувати, якщо використати кнопки «Сортувати за зростанням» та «Сортування за спаданням», або команду «Сортувати» пункту меню «Записи».

Імпортування даних у програму MS Query пов'язане здебільшого з можливістю створення деякого запиту відносно записів введених даних, тобто фільтрації записів за певними критеріями.

Найпростішим типом фільтрації є умова точного збігання значень поля із заданим пошуковим значенням. Для виконання такої фільтрації необхідно виділити те значення поля, якому повинні дорівнювати всі відфільтровані записи, та скористатися кнопкою «Фільтр за виділеним» .

За замовчуванням MS Query автоматично оновлює записи в ділянці даних після кожної зміни умов їх подання. Індикатором автоматичного оновлення є позначка рядка «Автоматичний режим» у пункті меню «Записи» або вдавнена кнопка з тією ж назвою  на панелі інструментів. Якщо цей режим вимкнтий, виконати запит можна за допомогою команди «Виконати запит» пункту меню «Записи» або кнопки з тією ж назвою .

Для фільтрації даних потрібно встановити певний критерій для записів стовпця в ділянці критеріїв, яка викликається за допомогою кнопки «Відображення умов»  або команди «Умови» пункту меню «Вид».

Ділянка критеріїв подібна інтервалу критеріїв розширеного фільтра системи Excel: верхній рядок містить заголовки полів, а нижче вводяться

Условие:	Фамилия	Имя
Значение:	'Василенко'	'Евгений'
умови:	или:	

Умови можна вводити безпосередньо в комірки ділянки критеріїв, а також за допомогою команд MS Query.

Як і в розширеному фільтрі, відбір записів можна продовжити, задаючи критерії в інших стовпцях. Кожний доданий критерій з'єднується з попереднім логічним оператором «Та». Якщо додаються нові умови для того самого поля, то їх записують у наступний рядок того самого стовпця та з'єднують логічним оператором «Або».


Для видалення критерію або скасування умов фільтрації потрібно виділити відповідний стовпець умов і натиснути клавішу «Delete». Команда «Видалити всі умови» пункту меню «Умови» відміняє всі критерії фільтра і повертає список записів у початковий стан.

Програма MS Query дає можливість створювати критерії порівняння. Для цього використовується діалогове вікно «Додавання умови», що

викликається однойменною командою «Додати умову» пункту меню «Умова».

Діалогове вікно «Властивості запиту», що викликається однойменною командою з пункту меню «Вид», дає змогу вилучити з таблиці результатів запиту всі записи, що повторюються, у разі ввімкнення режиму «Тільки унікальні записи».

Крім аналізу окремих таблиць даних MS Query має можливість одержувати дані з кількох зв'язаних таблиць. Для об'єднання (зв'язування) таблиць вони повинні мати загальні поля.

Відфільтровані дані з MS Query можна повернути до Excel за допомогою кнопки «Повернути дані»  або команди «Повернути дані» в MS Excel пункту меню «Файл».

За замовчуванням у діалоговому вікні «Імпорт даних» пропонується помістити дані у поточну комірку робочого аркуша або визначити нове місце розташування даних.

Кнопка «Змінити запит» дає можливість повернутися до MS Query й встановити новий варіант фільтрації даних, а кнопка «Властивості» – визначити режими збереження даних запиту. Зокрема, якщо встановлено режим «Зберегти визначення запиту», дані в Excel автоматично оновлюватимуться по мірі їх змін у зовнішньому джерелі даних.

Варіанти завдань

1. Нарахування заробітної плати працівникам малого підприємства (можливий набір полів: прізвище, ім'я, по батькові, посада, ставка за місяць, податки, до видачі).

2. Ведення обліку договорів страхування автомобілів (можливий набір полів: марка авто, прізвище, ім'я, по батькові страхувальника, ціна авто, дата укладання договору, тариф, страхова сума).

3. Ведення обліку договорів страхування побутової техніки (можливий набір полів: товар, прізвище, ім'я, по батькові страхувальника, ціна товару, дата укладання договору, тариф, страхова сума).

4. Ведення обліку договорів страхування нерухомості (можливий набір полів: вид житла, прізвище, ім'я, по батькові страхувальника, ціна нерухомості, дата укладання договору, тариф, страхова сума).

5. Ведення обліку трудових ресурсів малого підприємства (можливий набір полів таблиці: прізвище, ім'я, по батькові, посада, вік, стаж роботи).

6. Ведення обліку вантажоперевезень (можливий набір полів таблиці: марка авто, вантажопідйомність, ціна перевезень за 1 км, відстань, вид вантажу, вартість).

7. Нарахування стипендій (прізвище, ім'я, по батькові студента, група, курс, середній бал, розмір стипендії).

8. Оподаткування (тип підприємства, система оподаткування, прибуток підприємства, розмір податку на прибуток, сума податку).

9. Таблиця даних щодо обстеження попиту на промислові товари (можливий набір полів таблиці: товар, дата, постачальник, товарні запаси, продано товару).

10. Таблиця даних щодо обстеження попиту на продовольчі товари (можливий набір полів таблиці: товар, характеристика товару, дата, товарні запаси, продано товару).

11. Таблиця даних щодо обстеження попиту на спортивні товари (можливий набір полів таблиці: товар, код магазину, дата, товарні запаси, продано товару).

12. Таблиця даних щодо обстеження попиту на фототовари (можливий набір полів таблиці: товар, рік, місяць, товарні запаси, продано товару).

13. Таблиця даних щодо обстеження попиту на товари кондитерської фабрики (можливий набір полів таблиці: товар, рік, місяць, менеджер, магазин, продано товару).

14. Таблиця даних щодо динаміки руху товарів на складах (можливий набір полів таблиці: склад, товар, рік, місяць, товарні запаси, продано товару).

15. Таблиця динаміки продажу комп'ютерів (можливий набір полів таблиці: місяць, модель комп'ютера, магазин, кількість, ціна, вартість).

16. Пропозиція послуг туристичної фірми (можливий набір полів таблиці: країна, термін відпочинку, сезон, тип готелю, вартість).

17. Ведення обліку випуску продукції меблевої фабрики (можливий набір полів таблиці: вид виробу, матеріал, ґатунок виробу, термін виготовлення, вартість виробу).

18. Ведення обліку договорів кредитування автомобілів (можливий набір полів таблиці: марка авто, рік випуску, оціночна вартість, процентна ставка, термін кредитування).

19. Виробництво промислових товарів (можливий набір полів таблиці: вид деталі, витрати енергоресурсів, витрати часу на виробництво 1 шт., вага деталі, вартість).

20. Ведення обліку випуску продукції фабрики трикотажного полотна (можливий набір полів таблиці: вид виробу, матеріал, ґатунок виробу, термін виготовлення, вартість виробу).

Питання для самоконтролю

1. Що таке список з точки зору Excel?
2. Поняття поля даних і запису у списку Excel.
3. За яких умов Excel автоматично створить форму для списку?
4. Яке основне призначення форм для списків в Excel?
5. Як здійснити пошук записів, що задовольняють певні умови?
6. Які перетворення можна виконувати з даними у списку в Excel?
7. Що таке список користувача та як його створити в Excel?
8. Які можливості надає Excel для фільтрації даних?

За якими правилами «Автофільтр» виводить дані списку певного стовпця?

10. Скільки умов можна накласти на дані одного стовпця списку за допомогою вікна «Автофільтр користувача».

11. Як сформулювати умови фільтрації для розширеного фільтра?

12. Як скасувати результати фільтрації при використанні розширеного фільтра?

13. Яке основне призначення Microsoft Query?

14. Які типи даних можна проаналізувати за допомогою Microsoft Query?

15. Як виконується фільтрація за виділенням у Microsoft Query?

16. Для чого призначений «Автоматичний режим» у Microsoft Query?

17. За якою умовою об'єднуються критерії в одному рядку та у різних рядках?

ПРАКТИЧНА РОБОТА № 2 АГРЕГУВАННЯ ДАНИХ В MS EXCEL

Мета: здобути практичні навички створення зведених та консолідованих таблиць, засвоїти функції агрегування даних засобами MS Excel.

Завдання:

1. Розробити макет таблиці, занести початкові дані та розрахункові формули до комірок електронної таблиці.
2. Створити зведену таблицю за допомогою «Майстра зведених таблиць».
3. Навчитися працювати зі зведеною таблицею:
 - змінювати макет таблиці;
 - сортувати дані;
 - фільтрувати дані;
 - змінювати параметри полів;
 - оновлювати дані зведеної таблиці при зміні вихідних даних.
4. Використовуючи довідкову систему MS Excel, ознайомитися з вбудованими функціями для агрегування підсумків у зведених таблиці.
6. Навчитися формувати проміжні підсумки (пункт меню «Дані → Підсумки»). Майте на увазі, що для обчислення підсумків необхідно згрупувати рядки, за якими необхідно підвести підсумки.
7. Провести агрегування даних на основі консолідації (пункт меню «Дані → Консолідація»).

Методичні вказівки до виконання практичної роботи

Порядок побудови зведеної таблиці:

1. Розробити макет таблиці, занести вихідні дані та розрахункові формули в комірки електронної таблиці.
2. Для агрегування даних вибрати меню «Дані→ Зведена таблиця...».
3. На першому кроці роботи майстра зведених таблиць необхідно вказати джерело даних, яким може бути список або база даних MS Excel, зовнішнє джерело даних, діапазони консолідації або інша зведена таблиця. Також треба зазначити вид звіту: зведена таблиця або зведена діаграма.
4. Для переходу до наступного кроку роботи майстра необхідно натиснути кнопку «Далі».

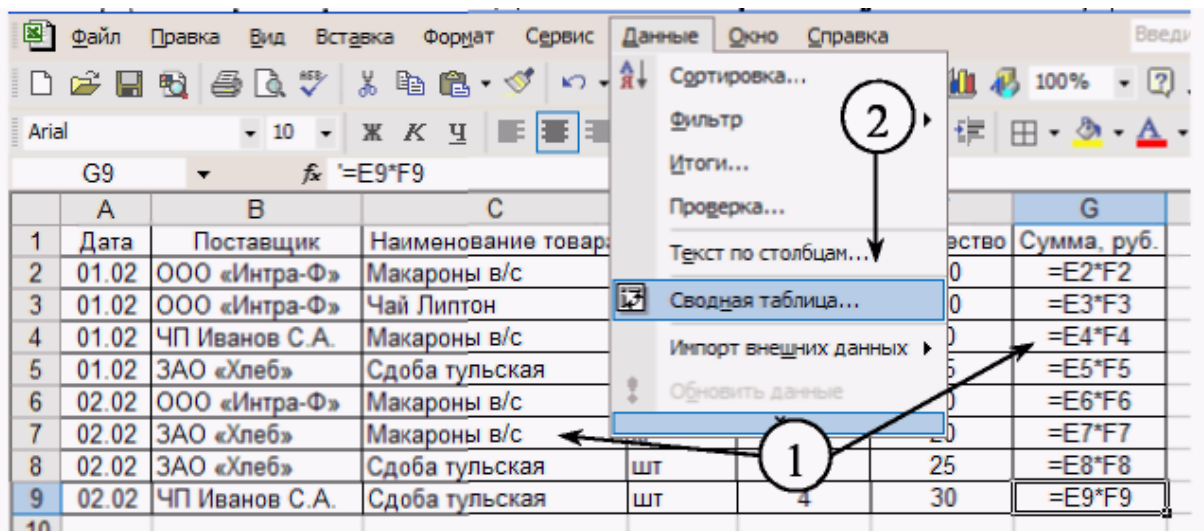


Рисунок 2.1 – Форматування таблиці та занесення вихідних даних

5. На другому кроці роботи майстра потрібно вказати діапазон, що містить вихідні дані. Діапазон можна задати в рядку введення з клавіатури, вказавши адресу лівої верхньої комірки й через двокрапку – адресу правої нижньої клітинки діапазону; можна виділити діапазон за допомогою миші безпосередньо в таблиці. Якщо вікно майстра заважає виділенню, його можна згорнути, натиснувши на кнопку «Звернути вікно» поруч з полем вводу, а потім відновити вікно майстра за допомогою кнопки «Розвернути вікно». Якщо дані знаходяться в іншому файлі, необхідно попередньо відкрити його, використовуючи кнопку «Огляд...», яка знаходиться на вікні майстра. Після того, як діапазон задано, переходимо до наступного кроку, натиснувши кнопку «Далі».

6. На останньому кроці роботи майстра необхідно вказати, де буде розміщена зведена таблиця – на новому аркуші книги або на існуючому аркуші. Якщо зведена таблиця розміщується на існуючому аркуші, то у полі введення потрібно задати посилання на клітинку, з позиції якої буде вставлена таблиця.

7. Також на третьому кроці роботи майстра можна створити макет зведеної таблиці. Для цього треба натиснути кнопку «Макет...» у вікні майстра.

8. Для формування макету необхідно перетягнути кнопки полів в області зведеної таблиці. Наприклад: в область сторінок перетягнемо дату, у рядки таблиці – найменування товару, в стовпці – постачальника, в область даних – кількість і суму (за їх значеннями буде проводитися агрегування).

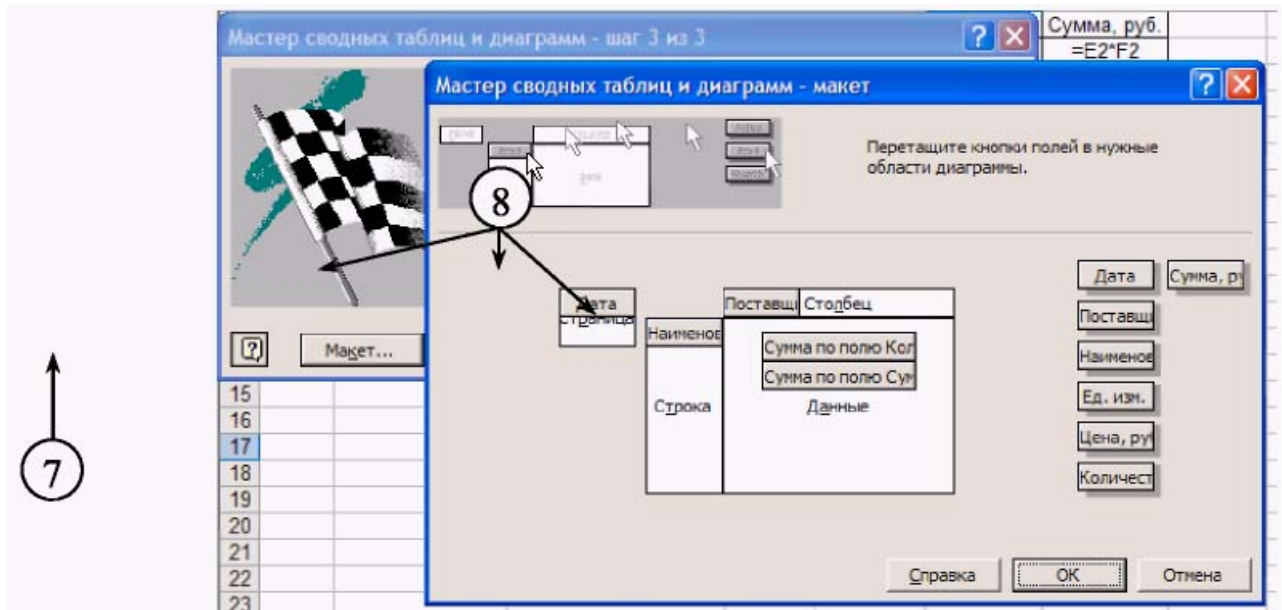


Рисунок 2.2 – Макетування зведеної таблиці

9. Для завершення роботи майстра натиснути кнопку «Готово». В результаті, на новому аркуші робочої книги вихідні дані будуть агреговані у формі зведеної таблиці.

10. Для керування зведеною таблицею можна скористатися кнопками панелі інструментів «Зведена таблиця». Якщо вона відсутня на екрані, необхідно вибрати пункт меню «Вид → Панелі інструментів → Зведені таблиці».

11. Якщо необхідно змінити операцію агрегування даних в зведеній таблиці, необхідно встановити курсор у поле даних, для якого змінюється операція, та натиснути кнопку «Параметри поля» на панелі інструментів зведеної таблиці. Відчиниться вікно обчислення полів зведеної таблиці.

12. В цьому вікні треба вибрати необхідну операцію агрегування (наприклад, кількість значень, середнє значення тощо) та натиснути кнопку «ОК».

13. Для відображення агрегованих даних по конкретній даті натиснути на кнопку списку поруч з полем «Дата» (область сторінки зведеної таблиці) та вибрати потрібну дату.

14. Можна приховати окремі рядки та стовпці зведеної таблиці. Для цього треба натиснути на кнопку списку в заголовках колонок або рядків зведеної таблиці та зняти позначки виділення для прихованих даних у списку.

15. Якщо вихідні дані у списку були змінені, то для оновлення значень у зведеній таблиці необхідно натиснути кнопку «Оновити дані».

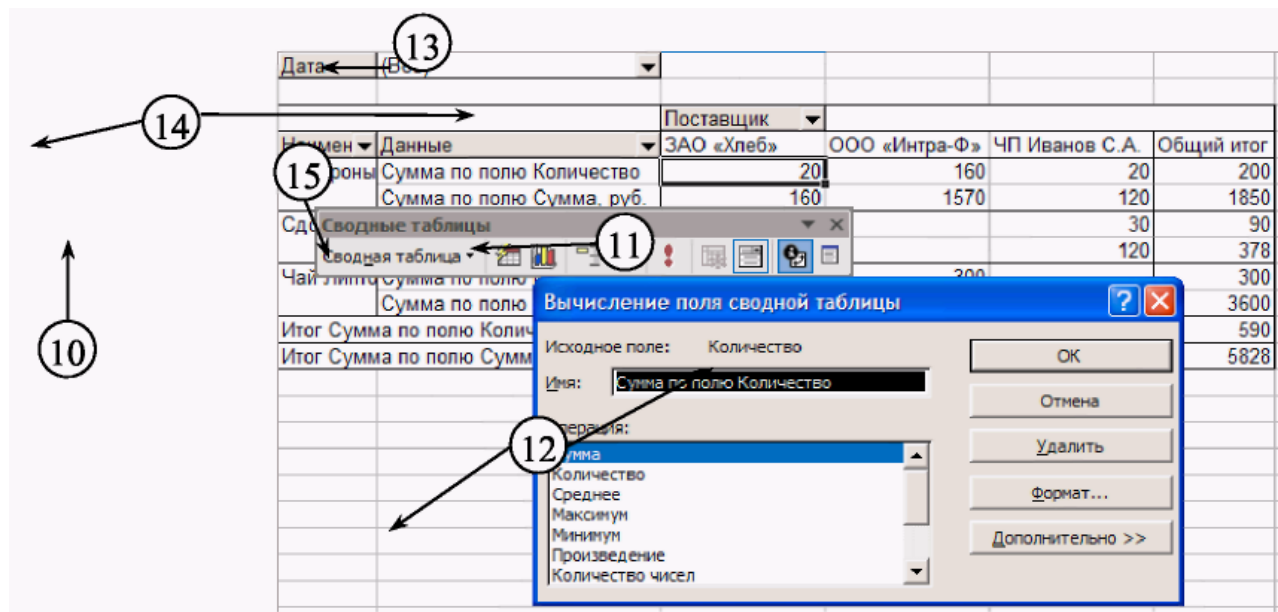


Рисунок 2.3 – Керування зведеною таблицею

16. Зберегти результати у файлі, обравши пункт меню «Файл → Зберегти», вказавши ім'я файлу.

Варіанти завдань:

Макет таблиці вихідних даних має наступний вигляд:

Дата	Постачальник	Найменування товару	Одиниці виміру	Ціна грн	Кількість, од.	Сума, грн
------	--------------	------------------------	-------------------	-------------	-------------------	--------------

Необхідно заповнити таблицю товарами категорій згідно з варіантом.

1. Продукти харчування.
2. Побутова техніка.
3. Жіночий одяг.
4. Будівельні матеріали.
5. Побутова хімія.
6. Меблі.
7. Мобільні пристрої.
8. Медичні препарати.
9. Канцелярські вироби.
10. Спортивні товари.
11. Промислові товари.
12. Фототовари.
13. Кондитерські вироби.

14. Комп'ютерна техніка.
15. Книги.
16. Чоловічий одяг.
17. Обігрівачі.
18. Іграшки.
19. Посуд.
20. Керамічні вироби.

Таблиця має містити дані 5 – 7 постачальників відповідних груп товарів. До складу групи товарів входить 5 – 10 найменувань.

Питання для самоконтролю

1. Дайте визначення терміну «Зведена таблиця».
2. Для чого використовується сортування даних?
3. Дайте визначення поняттю «Агрегування даних». Поясніть, для чого воно використовується.
4. Дайте визначення поняттю «Консолідація даних». Поясніть, для чого використовується консолідація?
5. Дайте визначення поняттю «Проміжні результати». Для чого використовується проміжні результати?
6. Для чого використовується зведена таблиця?
7. Дайте визначення поняттю «Макет таблиці».
8. Як змінити операцію агрегування даних в зведеній таблиці?
9. Як оновити дані в зведеній таблиці, якщо були змінені вихідні дані?
10. Для чого використовується фільтрація?
11. Як використовувати розширений фільтр?
12. Що може бути джерелом даних для зведеної таблиці?
13. Як вибрати джерело даних зведеної таблиці?

ПРАКТИЧНА РОБОТА № 3

СТВОРЕННЯ ФОРМИ КОРИСТУВАЧА ДЛЯ ПОБУДОВИ БАЗИ ДАНИХ

Мета: набути практичні навички в розробці форм користувача занесення до бази даних інформації різних типів.

Завдання: освоюючи принципи розробки програмних додатків в середовищі VBA, необхідно засвоїти наступні операції:

- послідовне заповнення бази даних;
- програмування приміток;
- програмування текстових полів (TextBox);
- використання перемикачів та прапорців (CheckBox, OptionButton);
- створення заголовку вікна додатку;
- програмне закріплення області на робочому аркуші.

Методичні вказівки до виконання практичної роботи

Розглянемо створення діалогового вікна вводу даних реєстрації клієнтів туристичної фірми.

Для заповнення бази даних на робочому аркуші за допомогою форми користувача створити діалогове вікно виду:

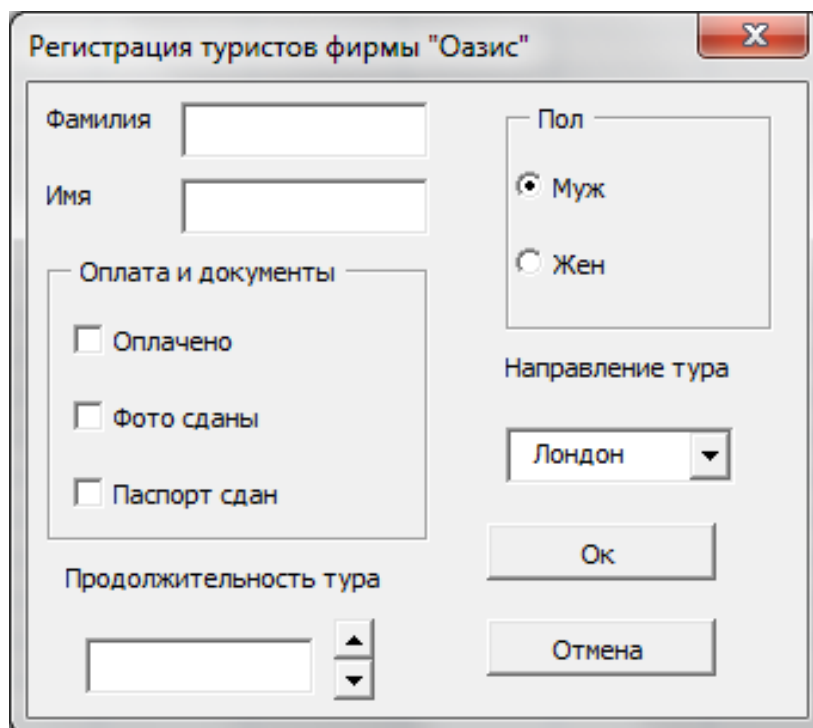


Рисунок 3.1 – Діалогове вікно

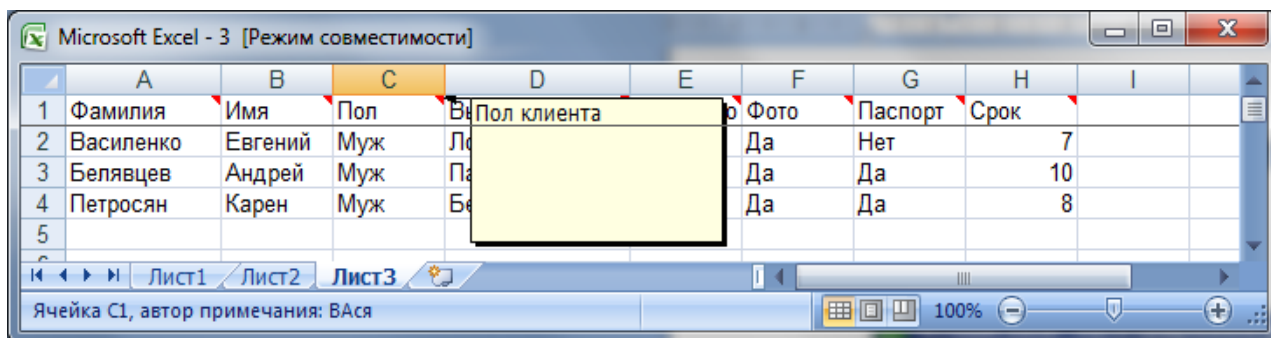


Рисунок 3.2 – База даних на робочому листі

До комірок з назвами стовпців бази та кнопок діалогового вікна необхідно програмним способом додати коментарі (примітки), що з'являються при наведенні курсору на об'єкт.

При ініціалізації діалогового вікна програма перевіряє, чи є назви полів створюваної бази даних реєстрації туристів. Якщо цих назв немає, то програма автоматично їх створює та додає коментарі.

Таблиця 3.1 – Основні процедури програмної моделі

Процедура	Дії процедури
UserForm_Initialize()	<ol style="list-style-type: none"> 1. Активізує діалогове вікно 2. Призначає клавіші «Esc» функцію кнопки «Відміна», а клавіші «Enter» – «OK». 3. Призначає кнопкам спливаючі підказки. 4. Закріплює першу строку так, щоб вона завжди відображалась на екрані. 5. Створює назви полів бази, якщо вони ще не були створені. 5. Заповнює список. 6. Встановлює текст назви вікна додатку.
Натиснення кнопки «OK» запускає на виконання процедуру CommandButton1_Click()	<ol style="list-style-type: none"> 1. Визначає номер першого пустого рядка в базі даних. 2. Зчитує дані з діалогового вікна. 3. Заносить їх до першого пустого рядку.
Натиснення кнопки «Відміна» запускає на виконання процедуру CommandButton2_Click()	Закриває діалогове вікно. Встановлює назву додатку, використаний за замовчуванням, тобто видаляє назву створену при активізації форми.
SpinButton1_Change()	Вводить значення в поле «Тривалість туру».
Заголовок Робочого аркушу()	Створює назви полів бази даних. Ці назви відображаються з примітками.

Програмний код додатку складається з наступних процедур.

Процедура «Private Sub CommandButton1_Click()» відповідає за зчитування інформації з діалогового вікна та запис її до бази даних на робочому листі.

Зміст змінних однозначно визначений їх назвами, число після змінної показує максимальну кількість знаків, яку може містити змінна.

Наприклад:

```
Dim Прізвище As String * 20
Dim Ім'я As String * 20
Dim Стать As String * 3
Dim Обраний тип As String * 20
Dim Сплачено As String * 3
Dim Фото As String * 3
Dim Паспорт As String * 3
Dim Термін As String * 3
Dim Номер рядку As String * 3
```

Номер рядку – це номер першої пустої строки робочого аркушу.

Номер рядку = Application.CountA(ActiveSheet.Columns(1)) + 1

Зчитування інформації з діалогового вікна описується наступним чином:

```
With UserForm1
Прізвище = .TextBox1.Text
Ім'я = .TextBox2.Text
Термін = .TextBox3.Text
If .OptionButton1.Value = True Then
Стать = «Чол.»
Else
Стать = «Жін.»
End If
If .CheckBox1.Value = True Then
Сплачено = «Так»
Else
Сплачено = «Ні»
End If
If .CheckBox2.Value = True Then
Фото = «Так»
Else
Фото = «Ні»
End If
If .CheckBox3.Value = True Then
Паспорт = «Так»
```

```
Else
Паспорт = «Hi»
End If
Обраний тип = .ComboBox1.List(.ComboBox1.ListIndex, 0)
End With
```

Введення даних у рядок з номером «Номер рядку» робочого аркушу виконується в наступній послідовності:

```
With ActiveSheet
.Cells(НомерСтроки, 1).Value = Прізвище
.Cells(НомерСтроки, 2).Value = Ім'я
.Cells(НомерСтроки, 3).Value = Стать
.Cells(НомерСтроки, 4).Value = Обраний тип
.Cells(НомерСтроки, 5).Value = Сплачено
.Cells(НомерСтроки, 6).Value = Фото
.Cells(НомерСтроки, 7).Value = Паспорт
.Cells(НомерСтроки, 8).Value = Термін
End With
End Sub
```

Наступна процедура відповідає за зачинення діалогового вікна. Вона спочатку приховує діалогове вікно (UserForm1.Hide), потім робить заголовок форми порожнім (Application.Caption = Empty), після чого видаляє всі програмні об'єкти з робочого аркушу.

```
Private Sub CommandButton2_Click()
UserForm1.Hide
Application.Caption = Empty
ActiveSheet.DrawingObjects.Delete
End Sub
```

Наступна процедура описує введення значення лічильника до поля вводу:

```
Private Sub SpinButton1_Change()
With UserForm1
.TextBox3.Text = CStr(.SpinButton1.Value) оператор CStr конвертує
дані в тип String
End With
End Sub
Private Sub TextBox3_Change() – це процедура, яка встановлює
лічильник з поля вводу.
Private Sub TextBox3_Change()
With UserForm1
```


.SpinButton1.Value = CInt(TextBox3.Text) – оператор CInt конвертує дані в тип Integer, якщо це можливо, у випадку неспівпадання типів може виникнути помилка.

End With

End Sub

Процедура виклику діалогового вікна та завдання елементів списку, що розкривається має наступний вигляд:

Private Sub UserForm_Initialize()

Заголовок Робочого аркушу. Для завдання користувальницького заголовку вікна додатку необхідно ввести наступний рядок програмного коду:

Application.Caption = «Реєстрація. База даних туристів»

Рядок Application.DisplayFormulaBar = False – закриває рядки формул вікна Excel.

Далі описується завдання елементів списку, що розкривається, заповнення ComboBox даними та активізація діалогового вікна:

With CommandButton1

.Default = True

.ControlTipText = «Введення інформації до бази даних»

End With

With CommandButton2

.Cancel = True

.ControlTipText = «Кнопка відміни»

End With

OptionButton1.Value = True

With ComboBox1

.List = Array(«Лондон», «Париж», «Берлін»)

.ListIndex = 0

End With

UserForm1.Show

End Sub

Остання процедура створює заголовки полів бази даних. Для цього вона спочатку перевіряє, чи існують вони. Якщо заголовки вже введені, то програма автоматично закривається, якщо ні – програма створює заголовки для кожного стовпчика бази даних та додає примітки. Ця процедура в програмі на мові VBA має наступний вигляд:

Sub Заголовок Робочого аркушу()

If Range(«A1»).Value = «Прізвище» Then

Range(«A2»).Select

Exit Sub

End If

```

ActiveSheet.Cells.Clear
Range(«A1:H1»).Value = Array(«Прізвище», «Ім'я», «Стать»,
«Обраний тур», «Сплачено», «Фото», «Паспорт», «Термін»)
Range(«A:A»).ColumnWidth = 12
Range(«D:D»).ColumnWidth = 14.4
Range(«2:2»).Select
ActiveWindow.FreezePanels = True
Range(«A2»).Select
Range(«A1»).AddComment
Range(«A1»).Comment.Visible = False
Range(«A1»).Comment.Text Text:=«Прізвище клієнта»
Range(«B1»).AddComment
Range(«B1»).Comment.Visible = False
Range(«B1»).Comment.Text Text:=«Ім'я клієнта»
Range(«C1»).AddComment
Range(«C1»).Comment.Visible = False
Range(«C1»).Comment.Text Text:=«Стать клієнта»
...
Range(«G1»).AddComment
Range(«G1»).Comment.Visible = False
Range(«G1»).Comment.Text Text:=«Наявність паспорта» & Chr(10) &
«Так/Ні»
Range(«H1»).AddComment
Range(«H1»).Comment.Visible = False
Range(«H1»).Comment.Text Text:=«Тривалість» & Chr(10) &
«поїздки»
End Sub

```

Варіанти завдань:

Тематика створюваної бази даних та форми введення інформації обирається згідно з варіантом:

1. Нарахування заробітної плати працівникам малого підприємства.
2. Ведення обліку договорів страхування автомобілів.
3. Ведення обліку договорів страхування побутової техніки.
4. Ведення обліку договорів страхування нерухомості .
5. Ведення обліку трудових ресурсів малого підприємства.
6. Ведення обліку вантажоперевезень.
7. Нарахування стипендій студентам.
8. Оподаткування малого бізнесу
9. Таблиця даних щодо обстеження попиту на промислові товари.

10. Таблиця даних щодо обстеження попиту на продовольчі товари.
11. Таблиця даних щодо обстеження попиту на спортивні товари.
12. Таблиця даних щодо обстеження попиту на фототовари.
13. Таблиця даних щодо обстеження попиту на товари кондитерської фабрики.
14. Таблиця даних щодо динаміки руху товарів на складах на промислових товарів.
15. Таблиця динаміки продажу комп'ютерів.
16. Пропозиція послуг туристичної фірми.
17. Ведення обліку випуску продукції меблевої фабрики.
18. Ведення обліку договорів кредитування автомобілів.
19. Виробництво промислових товарів.
20. Ведення обліку випуску продукції фабрики трикотажного полотна.

Питання для самоконтролю

1. Якою подією викликається процедура «UserForm_Initialize()»?
2. Чим відрізняються такі елементи управління як «OptionButton» та «CheckBox»?
3. Для чого використовується «ComboBox»?
4. Як за допомогою програмного коду додати коментарі до комірки MS Excel?
5. Як за допомогою програмного коду заповнити «ComboBox»?
6. Як за допомогою програмного коду внести данні у комірку MS Excel?
7. Для чого використовується такий елемент управління, як «Frame»?
8. Що робить метод «Show»?
9. Що зберігає властивість «Caption»?
10. Що зберігає властивість «Name»?
11. «Dim Прізвище As String * 20» що означає «20» у цьому записі?
12. Якою подією викликається процедура «CommandButton_Click»?

ПРАКТИЧНА РОБОТА № 4 РОБОТА З ШАБЛОНАМИ MS ACCESS

Мета: здобути практичні навички у створенні баз даних за допомогою шаблонів, вбудованих в додатку MS Access.

Завдання:

1. Створити шаблон бази даних (згідно з номером варіанту).
2. Заповнити базу умовними даними.
3. Охарактеризувати функціональні об'єкти створеної бази даних.

Методичні вказівки до виконання практичної роботи

Програма MS Access – одна з сучасних реляційних систем керування базами даних (СКБД), яку можуть використовувати фахівці різних галузей економіки, промисловості та науки. MS Access створена для забезпечення комп'ютеризації кадрової, виробничої діяльності, процесів обліку у фінансовій сфері тощо. MS Access, як і будь-яка система керування базами даних, має досить багато різних можливостей для аналізу та обробки баз даних.

Сукупність даних, типових для конкретної інформаційної області, можна структурно зобразити у вигляді множини пов'язаних між собою окремих інформаційних частин. До будь-якої такої сукупності даних у загальному випадку можна застосувати термін «база даних». Наприклад, базою даних можна вважати сукупність інформації про кадровий склад підприємства, розклад занять у навчальному закладі, родинні зв'язки, бібліотечний фонд тощо. Поняття «база даних» стосовно інформаційних систем, розміщених на комп'ютерних носіях даних, означає набір записів і файлів, організованих у спеціальний спосіб. Спеціальне програмне забезпечення, яке потрібне для використання та модифікації баз даних користувачем, називається системою керування базами даних.

Основні функції СКБД:

- забезпечення користувача інструментарієм, що дає можливість оперувати даними в абстрактних термінах, не пов'язаних із способом збереження даних в ЕОМ;
- забезпечення секретності та пріоритетності доступу до даних;
- захист цілісності даних;
- синхронізація доступу до даних;
- захист даних від відмови та відновлення даних.

Відповідні функції, що виконує СКБД, можна розподілити за типом на групи:

- визначення даних (структура, тип, зв'язки між даними, елементи підстановок для значень тощо);
- обробка даних за допомогою різних методів, причому виконуються такі дії, як фільтрація й сортування;
- керування даними, що дає змогу розподілити дані за користувачами, надаючи їм право доступу до них, причому як з наданням можливості коригувати їх, так і без неї (режим перегляду даних).

Кожна конкретна інформаційна область за своєю структурою належить до деякого типу. Найпоширенішими типами базами є ієрархічні, мережні, реляційні.

Якщо інформаційна область даних пов'язана з ієрархією її частин, таку структуру називають ієрархічною. До ієрархічних інформаційних областей належать, наприклад, дані, пов'язані з родинними зв'язками, кадрові та виробничі структури міністерств тощо. Графічно дані з ієрархічною структурою подають у вигляді так званого дерева або сукупності дерев.

Якщо інформаційна область даних пов'язана з деякою мережею, то відповідну інформаційну структуру називають мережною. Прикладом мережних інформаційних областей є дані про стан електричних мереж, шляхів транспорту тощо. Графічно дані з мереженою структурою зображують у вигляді так званого графа.

Нарешті, якщо інформаційна область має вигляд пов'язаних між собою двовимірних таблиць, то одержану структуру даних можна назвати реляційною (від англ. *relation* – відношення). За типом інформаційних структур розрізняють і системи керування базами даних. Найпоширенішими у використанні виявились реляційні СКБД, прикладом яких є MS FoxPro, MS Access тощо.

Перевага реляційних СКБД пояснюється легкістю їх використання та ефективністю реалізації. Слід зазначити, що будь-яку інформаційну структуру можна визначити за допомогою моделі іншої структури, але саме СКБД реляційного типу дає можливість зробити це з найменшими втратами.

Рядки двовимірних таблиць називаються записами, стовпці – полями. У термінах реляційних СКБД такі таблиці називаються відношеннями, записи, що вони містять, – кортежами відношень, поля – атрибутами відношень. Саме тому СКБД такого типу називають реляційними.

Основними об'єктами ACCESS є:

– *таблиця* – об'єкт, обумовлений і використовуваний для збереження даних. Кожна таблиця містить інформацію про об'єкти визначеного виду;

– *запит* – об'єкт, що дозволяє користувачеві одержати потрібні дані з однієї або декількох таблиць, або інших запитів. Можна створити запит на вибірку, відновлення, видалення або створення нових таблиць;

– *форма* – об'єкт, призначений в основному для введення даних, відображення їх на екрані або керування роботою додатка;

– *звіт* – об'єкт, призначений для створення документа, який можна роздрукувати або включити в документ іншого додатка. Форми й звіти одержують дані як безпосередньо з таблиць, так і через запити;

– *макрос* – об'єкт, що є структурованим описом дій виконуваних у відповідь на визначену подію. Це будь-яка зміна стану об'єкта Microsoft Access. Приклад макросів: пошук в іншій таблиці; перевірка значення поля при його введенні або зміні;

– *Модуль* – об'єкт, що містить програми написані на Visual Basic for Applications (VBA). Модулі можуть бути незалежними об'єктами, що містять функції, які можна викликати з будь-якого місця додатка: макросу, форми або звіту. Ці функції можна застосовувати у вираженнях, використовуваних у запитах, а також умовах на значення при роботі з таблицями й звітами. Але модулі можуть бути безпосередньо «прив'язані» до окремих форм або звітів для реакції на ті або інші зміни, що відбуваються в них, тобто події.

Записи в базі відрізняються за номером, а поля – за іменем. Основні умови, яким повинні задовольняти таблиці:

- однакові записи заборонені;
- усі записи повинні мати одну й ту саму кількість полів;
- значення полів атомарні, тобто таблиця не може містити інші таблиці як свої компоненти.

Порядок створення бази даних з використанням шаблону:

1. Відкрити область задач «Створення файлу» («*New file*»), якщо вона ще не була відкрита.

2. З розділу «Створення за допомогою шаблону» («*New from template*») вибрати команду «Загальні шаблони» («*General templates*»). Після чого з'явиться діалогове вікно «Шаблони» («*Templates*»).

3. Вкладка «Бази даних» («*Databases*») дозволяє використовувати як основу один із шаблонів найбільш популярних баз даних. Для цього необхідно вибрати ярлик шаблону (рис. 4.1), натиснути кнопку «OK».

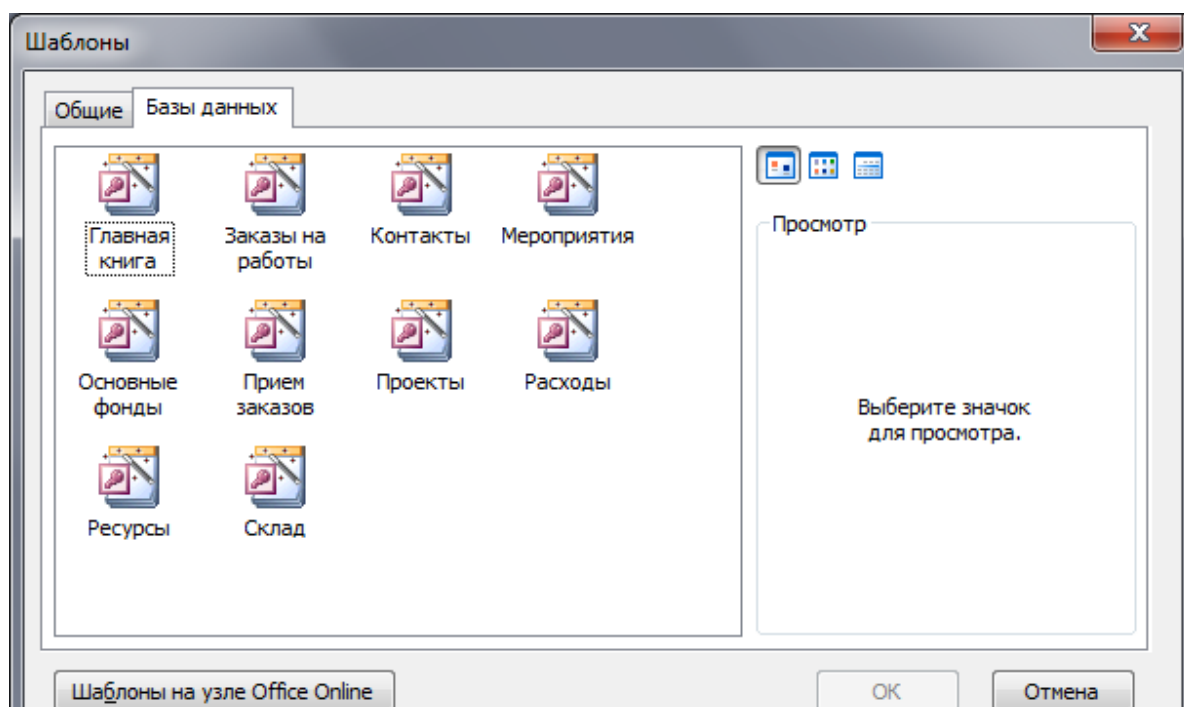


Рисунок 4.1 – Вкладка «База даних»

4. Далі необхідно додержуватися вказівок «*Майстра шаблонів*». При роботі з «*Майстром шаблонів*» спочатку необхідно задати назву нової бази даних, після чого вона буде відкрита. Потім приводяться короткі зведення про зміст бази. Потрібно натиснути на кнопку «Далі» («*Next*»). Після цього будуть показані таблиці й поля, що використовуються в базі (рис. 4.2).

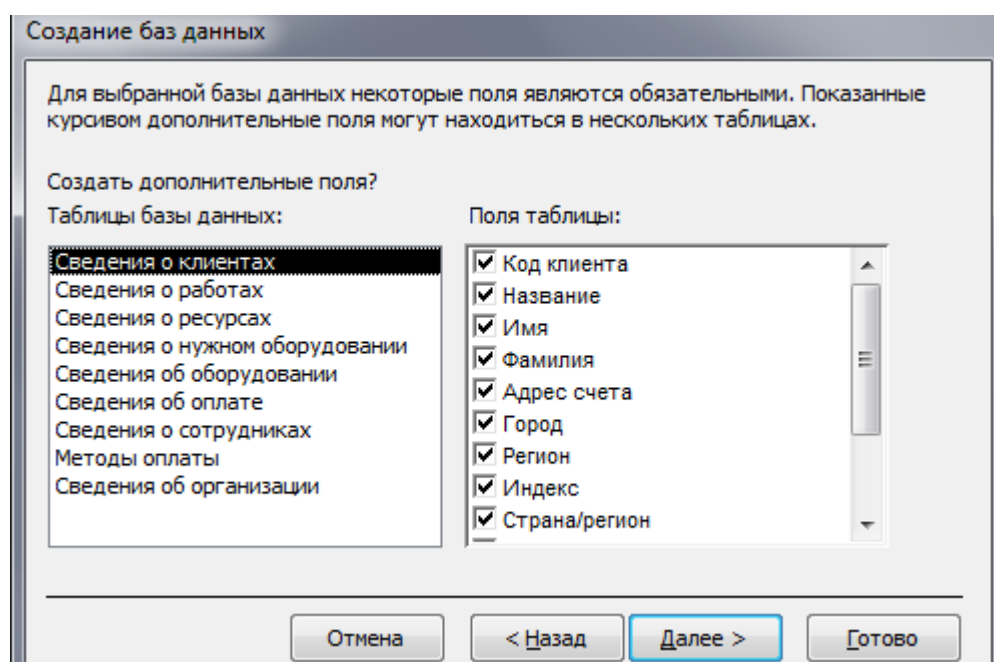


Рисунок 4.2 – Діалогове вікно вибору таблиць і полів

5. Наступні кроки (перехід до кожного наступного кроку здійснюється натисканням кнопки «Далі» («Next»)) дозволяють вибрати стильове оформлення форм (рис. 4.3) і звітів (рис. 4.4), а також задати заголовок головного меню нової бази даних і файл малюнку, що використовується при друкуванні звітів (рис. 4.5).

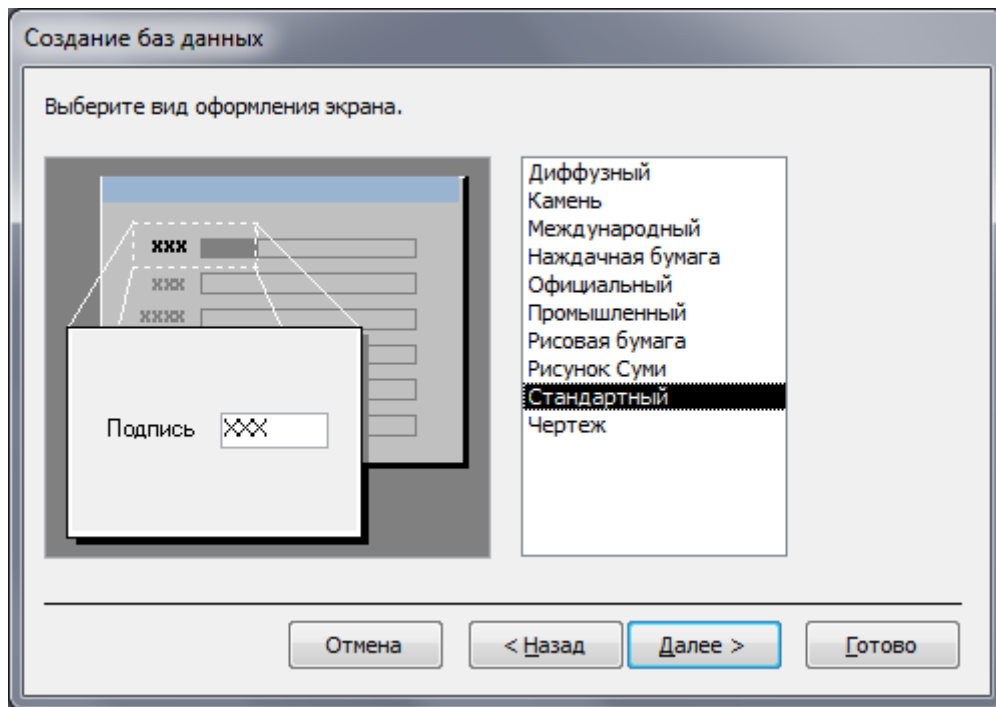


Рисунок 4.3 – Діалогове вікно стильового оформлення форм

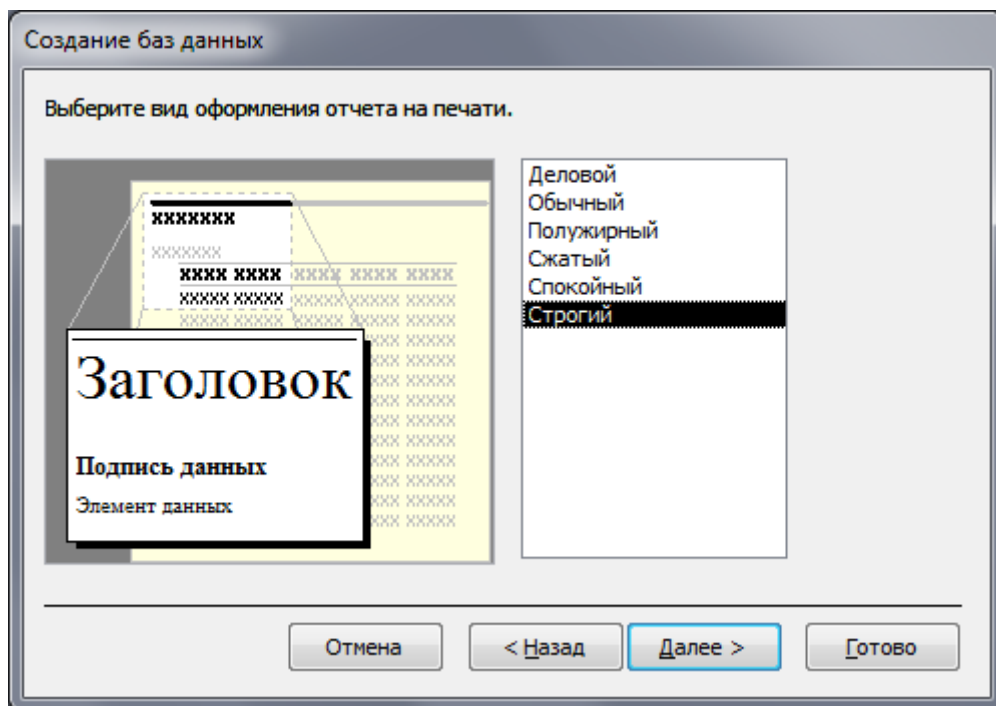


Рисунок 4.4 – Діалогове вікно стильового оформлення звітів

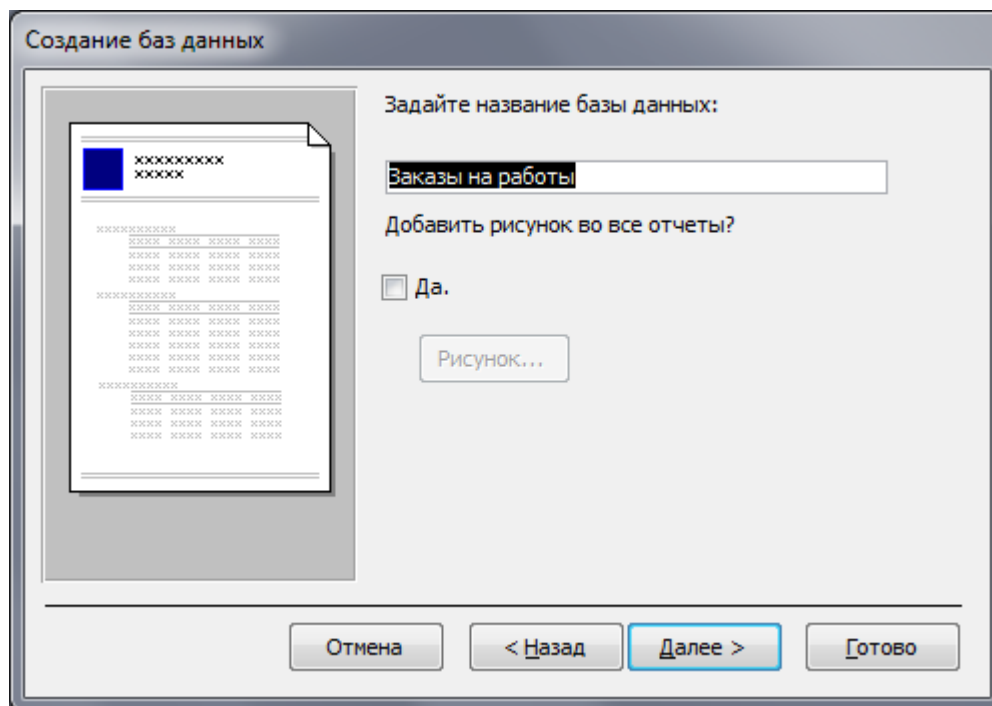


Рисунок 4.5 – Діалогове вікно «Майстра шаблонів», що задає заголовок бази даних та файл логотипу звітів

6. Нарешті, на останньому кроці можна повідомити Access про відмовлення негайного відкриття знову створеної бази (для цього необхідно зняти прапорець «Так» («Yes») в останньому вікні майстра), після чого натиснути кнопку «Готово» («Finish»).

Варіанти завдань:

1. Шаблон «Головна книга»
2. Шаблон «Замовлення»
3. Шаблон «Контакти»
4. Шаблон «Заходи»
5. Шаблон «Основні фонди»
6. Шаблон «Прийом замовлень»
7. Шаблон «Проекти»
8. Шаблон «Витрати»
9. Шаблон «Ресурси»
10. Шаблон «Склад»

Питання для самоконтролю

1. Які існують інформаційні моделі?
2. Що таке база даних?
3. Що таке система керування базами даних?
4. Які основні функції СКБД?
5. Які об'єкти належать до бази даних MS Access?
6. Дати визначення таблиці.
7. Дати визначення запиту.
8. Дати визначення звіту.
9. Дати визначення модулю.
10. Дати визначення форми.
11. Дати визначення макросу.
12. Назвіть послідовність дій при створенні бази даних за шаблоном.

ПРАКТИЧНА РОБОТА № 5

РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПЕРСОНАЛОМ

Мета: здобути практичні навички у розробці та проектуванні баз даних, побудові кнопоквих форм та оформленні звітів в середовищі MS Access.

Завдання: розробити базу даних, передбачивши в ній:

1. Кілька таблиць, зв'язаних між собою відношенням «один до багатьох».
2. Форми для введення даних.
3. Запити для вибірки інформації, що відповідає певним критеріям.
4. Звіти для виведення даних у заданому форматі.
5. Кнопокві форми з можливостями швидкого доступу до складових баз даних, функціональність яких полягає у відкритті форм, звітів і виконанні запитів.
6. Розрахункові поля мають бути забезпечені макросами або запрограмованими процедурами.

Методичні вказівки до виконання практичної роботи

Структуру бази даних «Зарплата» складають три таблиці, при чому серед довідкових таблиць слід визначити дві: «Посади» і «Працівники», а третя – «Зарплата», має обліковий характер. Нехай ці таблиці містять зв'язані між собою поля (жирним шрифтом позначені ключеві поля).

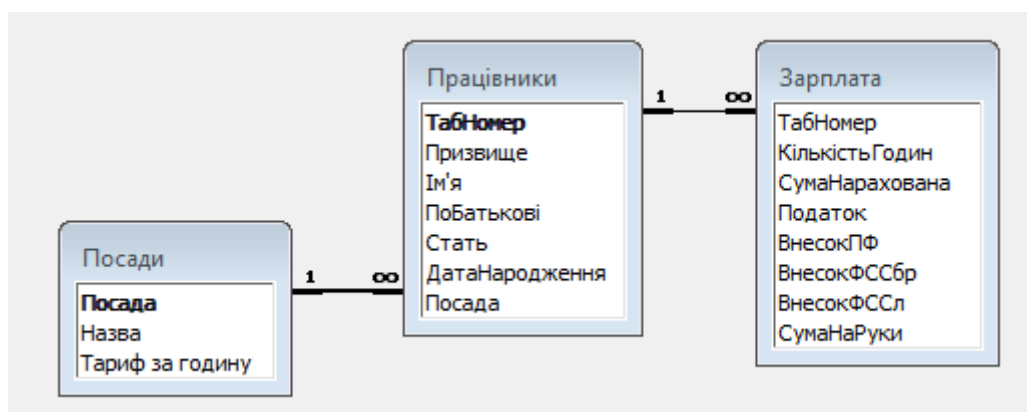


Рисунок 5.1 – Схема даних

Типи полів та їх розмір (максимальна кількість символів для даних поля) зведені до таблиці 5.1.

Таблиця 5.1 – Типи полів та їх розміри

Таблиця	Поле	Тип	Розмір
ПОСАДИ	Посада	текстовий	3
	Назва	текстовий	40
	ТарифЗаГодину	грошовий	
ПРАЦІВНИКИ	ТабНомер	текстовий	4
	Прізвище	текстовий	16
	Ім'я	текстовий	12
	ПоБатькові	текстовий	16
	Стать	текстовий, «Ч» або «Ж»	1
	ДатаНародження	Дата	
	Посада	текстовий	3
	ТабНомер	текстовий	4
ЗАРПЛАТА	КількістьГодин	числовий, цілий	
	СумаНарахована	грошовий	
	Податок	грошовий	
	ВнесокПФ	грошовий	
	ВнесокФССбр	грошовий	
	ВнесокФССл	грошовий	
	СумаНаРуки	грошовий	

У деяких рядках цієї таблиці не вказано розмір, тому що в MS Access він визначається автоматично. Наприклад, розмір поля типу «дата» дорівнює 8.

Обумовимо правила обчислення розрахункових полів, вказуючи перед ними назву відповідної таблиці.

ЗАРПЛАТА. СумаНарахована – загальна сума нарахованої зарплати, обчислюється як добуток кількості відпрацьованих годин і тарифу оплати праці за годину.

ЗАРПЛАТА. Податок – прибутковий податок обчислюється за шкалою залежно від суми нарахованої зарплати S (грн). Спочатку треба відкинути дробову частину нарахованої зарплати, позначити результат як S , після чого розрахувати податок, сума якого залежить від інтервалу, до якого належить S (табл. 5.2).

Таблиця 5.2 – Залежність податку від суми заробітної плати

Інтервал, до якого належить S	%
[0;17]	0
(17;85]	10
(85; 170]	15
(170; 1020]	20
(1020;1700]	30
(1700; ∞)	40

ЗАРПЛАТА. Внесок ПФ – внесок до пенсійного фонду обчислюється за шкалою залежно від суми нарахованої зарплати S грн (табл. 5.3).

Таблиця 5.3 – Залежність внесків до пенсійного фонду від суми заробітної плати

Інтервал, до якого належить S	Внесок, грн.
$[0;150]$	$S*1\%$
$(150;2200]$	$S*2\%$
$(2200; \infty)$	32

ЗАРПЛАТА. Внесок ФССбр – внесок до фонду соціального страхування по безробіттю обчислюється за шкалою залежно від суми нарахованої зарплати S грн (табл. 5.4).

Таблиця 5.4 – Залежність розміру внеску до фонду соціального страхування по безробіттю від суми заробітної плати

Інтервал, до якого належить S	Внесок, грн.
$[0;2200]$	$S*0,5\%$
$(2200; \infty)$	8

ЗАРПЛАТА. Внесок ФССл – внесок до фонду соціального страхування з непрацездатності обчислюється за шкалою залежно від суми нарахованої зарплати S грн (табл. 5.5).

Таблиця 5.5 – Залежність розміру внеску до фонду соціального страхування по непрацездатності від суми заробітної плати

Інтервал, до якого належить S	Внесок, грн.
$[0;150]$	$S*0,25\%$
$(150;2200]$	$S*0,5\%$
$(2200; \infty)$	8

ЗАРПЛАТА. Сума На Руки – заробітна плата, яку працівник отримує через касу, обчислюється як різниця між загальною нарахованою зарплатою S і сумою податків і внесків.

Побудова таблиць бази даних. Насамперед потрібно побудувати таблиці, поля яких не містять значень полів з інших таблиць. Такою в розгляданій задачі є таблиця «Посади». Натиснувши кнопку «Створити», слід відкрити вікно нової таблиці, де можна вибрати один з кількох режимів побудови таблиці, зокрема режим «Конструктор», вікно

якого складається з трьох стовпців для введення назви поля, типу даних поля й приміток (рис. 5.2).

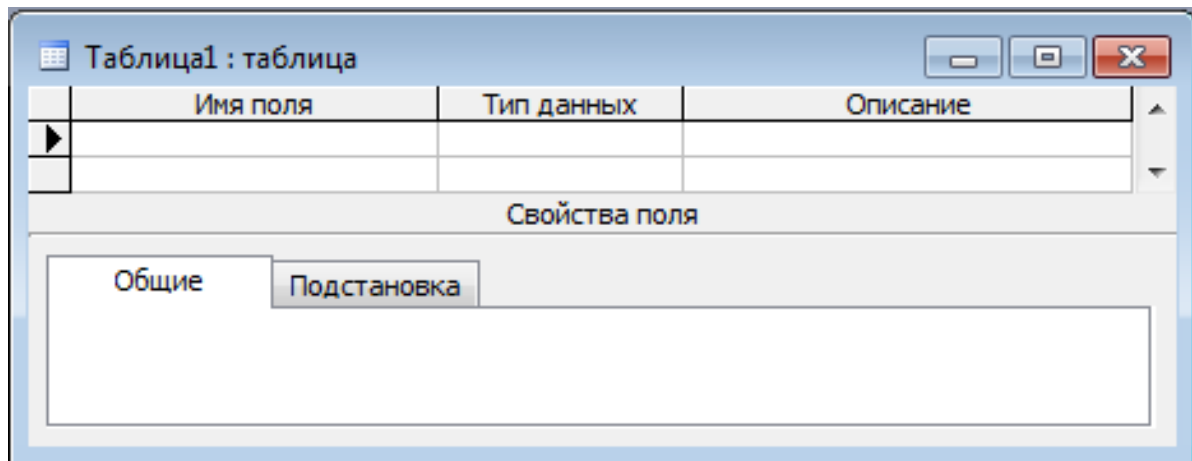


Рисунок 5.2 – Зовнішній вигляд конструктора таблиць

Сюди необхідно ввести зазначені вище поля, їх тип. Якщо поле ключове, то слід позначити його піктограмою з ключем, розташованою на панелі конструктора таблиць (повторне клацання відмінює ознаку ключового поля). Для таблиці «Посади» ключовим є поле «Посада». В нижній частині вікна конструктора побудови таблиці у вкладці «Загальні» слід ввести розмір поля, зокрема для поля «Посада» це буде 3.

Після введення всіх полів та їх розмірів потрібно зберегти структуру таблиці «Посади». Під час цієї дії система пропонує за замовчуванням назву «Таблиця», а користувач може ввести свою назву (у цьому випадку – «Посади»). Після закриття вікна у списку з'явилось ім'я «Посади». У разі необхідності повернутися до вікна побудови структури таблиці (щоб виправити помилки або переглянути структуру таблиці) можливо натисненням кнопки «Конструктор».

При створенні таблиці «Працівники» крім дій, що були виконані під час створення таблиці «Посади», потрібно також забезпечити підстановку поля «Посада». Це можливо зробити за допомогою «Майстра підстановок», який викликається в розділі типів даних вікна побудови структури таблиці. На першому кроці для обробки підстановки поля «Стать» необхідно вибрати режим «Буде введено фіксований набір значень» для подальшого забезпечення підстановки значеннями «Ч» і «Ж» (рис. 5.3).

На другому кроці, вибравши кількість стовпців – 1, необхідно ввести значення «Ч» і «Ж» відповідно в першому та другому рядках стовпця 1 (рис. 5.4).

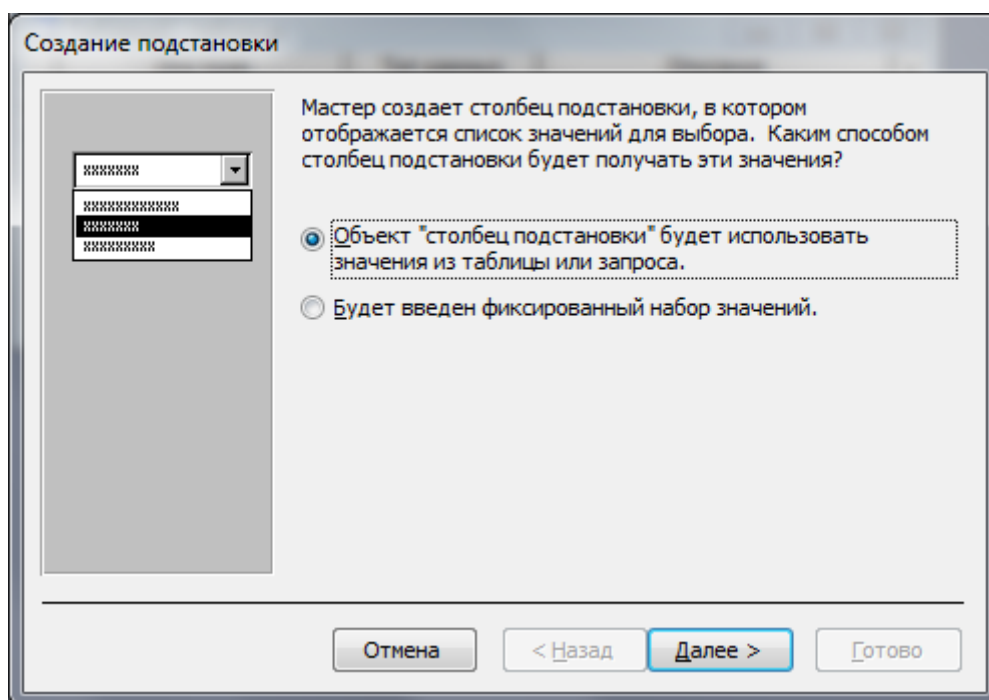


Рисунок 5.3 – Створення підстановки (вибір способу отримання даних)

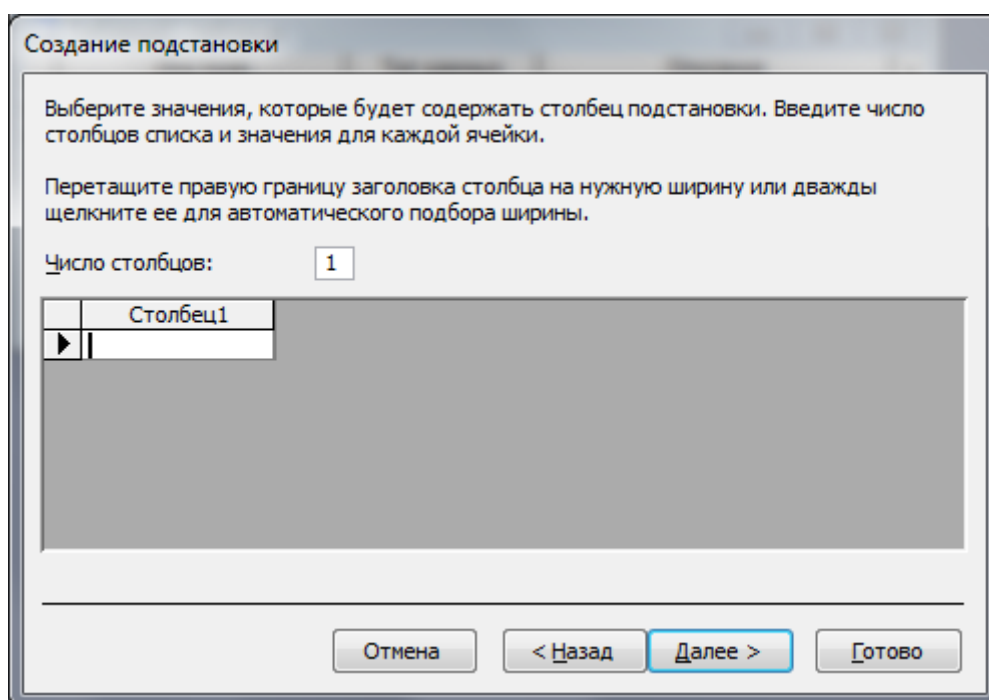


Рисунок 5.4 – Створення підстановки (введення даних)

На третьому кроці слід погодитись із назвою підпису (або змінити цю назву) й завершити цим кроком увесь процес організації підстановки (рис. 5.5).

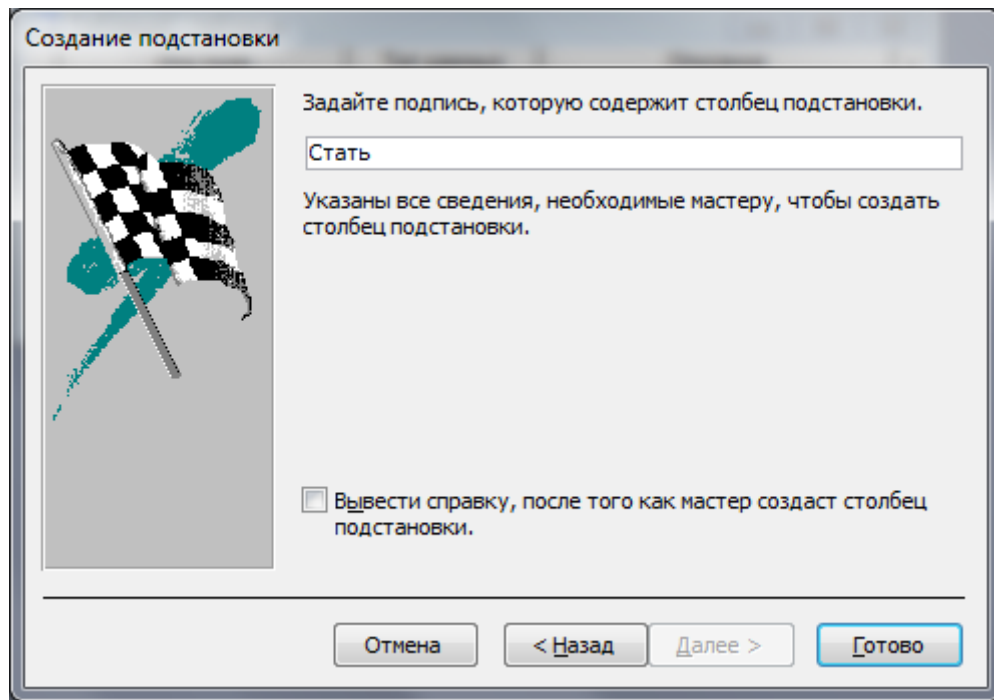


Рисунок 5.5 – Створення підстановки (підпис стовпця підстановки)

Якщо потрібно виконати підстановку поля «Посада», то на першому кроці слід вибрати режим «Об'єкт «стовбець підстановки» буде використовувати значення із таблиці або запиту». В цьому разі на другому кроці треба вибрати таблицю, поля якої надають дані для підстановки (у задачі такою таблицею є таблиця «Посади», рис. 5.6).

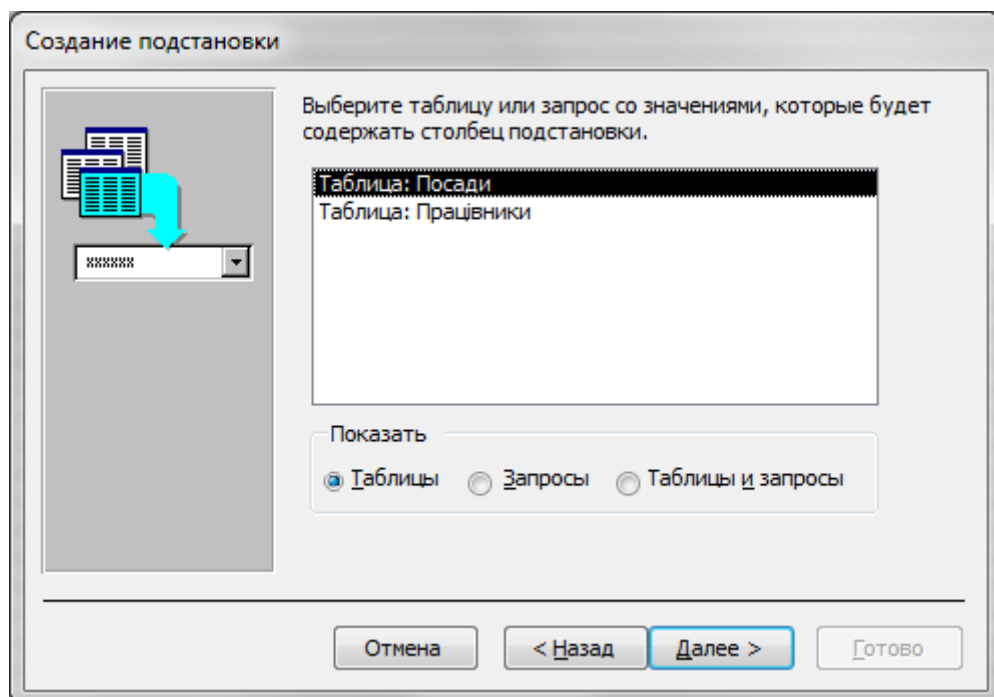


Рисунок 5.6 – Підстановка даних з іншої таблиці

На третьому кроці також є можливість з поля «Доступні поля» у поле «Обрані поля» перемістити ті поля, що допоможуть у подальшому правильно вибрати той або інший запис. У цьому прикладі можна обмежитись полями «Посада» і «Назва» (рис. 5.7).

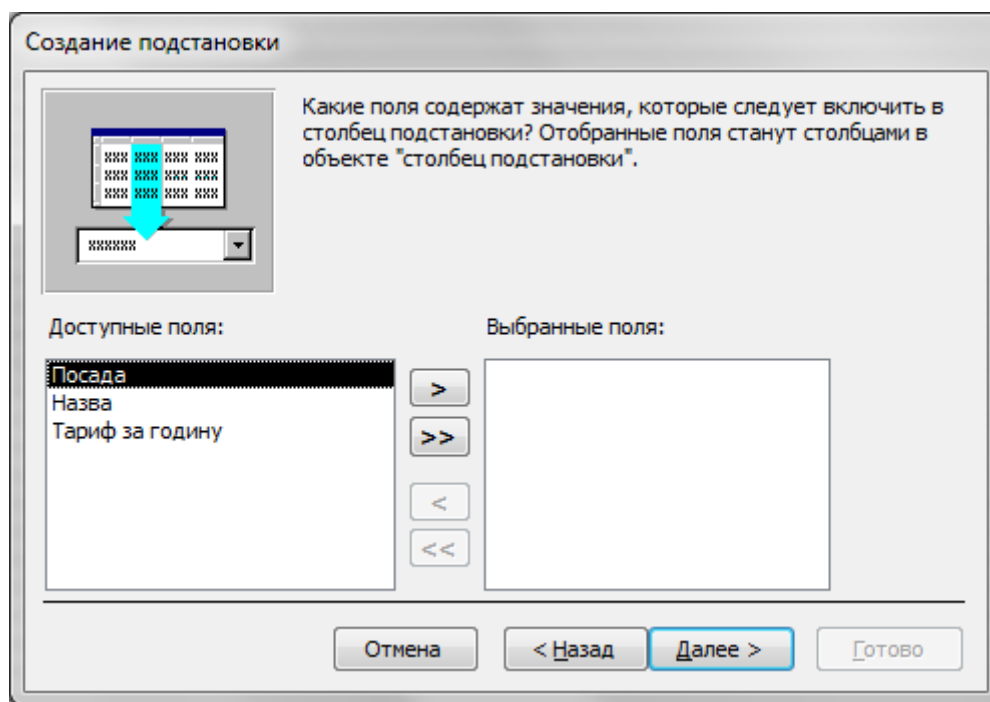



Рисунок 5.7 – Створення підстановки (вибір полів)

Четвертий крок підстановки значень полів нагадує другий крок фіксованої підстановки. Відмінність полягає в тому, що з'являється прапорець керування візуалізацією ключового стовпця. За замовчуванням режим «Приховати ключовий стовбець (рекомендується)» вимкнено, але це не завжди буває зручно, і досить часто існує потреба у візуалізації значення ключового поля.

На п'ятому кроці необхідно вибрати ключове поле, назва якого може з'явитися в процедурах. Отже, вибравши в прикладі поле «Посада», можна зробити останній крок, що за змістом такий самий, як у разі підстановки фіксованих значень, але з іншим значенням підпису.

Побудова схеми даних. Для забезпечення цілісності бази даних необхідно побудувати схему даних. Для цього, залишаючись у вкладці «Таблиця», за допомогою кнопки «Схема даних»  панелі інструментів слід відкрити вікно «Схема даних» із зображенням таблиць бази даних.

Якщо таблиць у вікні не виявиться, відкриється діалогове вікно «Додавання таблиці», за допомогою якого можна вибрати потрібні таблиці й запити бази даних (рис. 5.8).

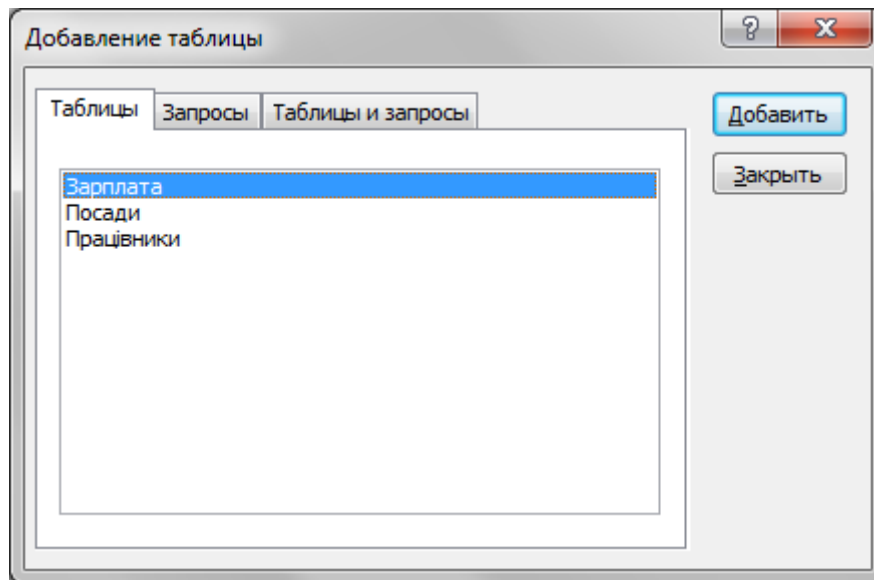


Рисунок 5.8 – Додавання таблиці

У тому разі, коли таблиця зайва, її потрібно виділити, а потім видалити, натиснувши клавішу «Delete». Якщо зв'язки у схемі відсутні, їх можна встановити лівою кнопкою миші за допомогою уявного з'єднання полів. Наприклад, розташувати курсор на полі «Посада» таблиці «Посади» і, утримуючи кнопку миші, перетягнути його до відповідного поля «Посада» таблиці «Працівники». У результаті такої дії відкриється вікно зв'язків. Якщо зв'язок існує, виділивши його за допомогою контекстного меню, можна відкрити вікно «Зв'язки» (рис. 5.9).

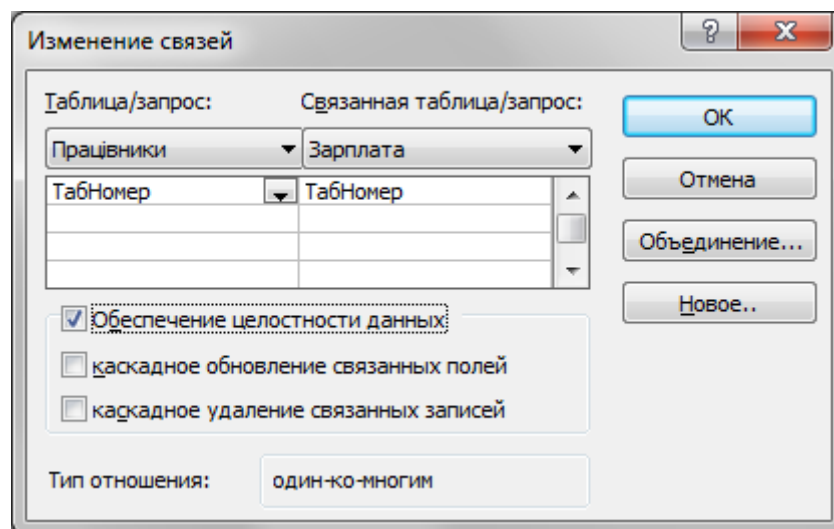


Рисунок 5.9 – Зв'язки

У цьому вікні необхідно увімкнути режим «Забезпечення цілісності даних», що активізує ще два режими каскадного оновлення й вилучення даних. Ці режими зазвичай вмикати не треба. Вийти з вікна «Зв'язки»

«Системи оброблення економічної інформації»

можна за допомогою кнопки «Створити». Після виконання цих дій буде побудована схема даних (рис. 5.1).

Побудова форм для введення даних. Для побудови форм для введення даних слід перейти у вкладку «Форми». Побудова об'єкта здійснюється за допомогою кнопки «Створити». Під час виконання цієї процедури зручно скористатись майстром форм. Вибравши варіант технології побудови форми, потрібно визначити базову таблицю. На першому кроці майстер форм пропонує визначитись з множиною полів для вибраної таблиці або таблиць.

При побудові форми для таблиці «Посади» слід вибрати всі поля. На другому кроці майстер дає можливість задати структуру форми. Можна вибрати, наприклад, вигляд для форми «В один стовпець».


На третьому кроці майстер пропонує визначитись з дизайном форми. Після чого майстер пропонує дати назву формі й варіанти подальших дій користувача: або відкрити форму для накопичення (редагування) даних, або редагувати форму, що побудована майстром.

Аналогічно будується форма для таблиці «Працівники».

Форма для таблиці «Зарплата» будується за тими самими правилами, що й попередні. Але тут виникає необхідність здійснити певні обчислення. Спочатку в середовищі майстра треба сформувати множину полів форми: це всі поля таблиці «Зарплата», поля «Назва» і «ТарифЗаГодину» таблиці «Посади», а також поля «Прізвище», «Ім'я» та «ПоБатькові» таблиці «Працівники».

Після завершення звичайної роботи майстра побудови форм слід ще налаштувати поля, що обчислюються. Тобто вже побудовану форму для таблиці «Зарплата» потрібно відредагувати, для чого її необхідно відкрити за допомогою «Конструктора».

У формі «Зарплата» необхідно забезпечити обчислення поля «Сума нарахована». Значення цього поля залежить від значень двох полів: «КількістьГодин» і «ТарифЗаГодину». У властивостях елементу форми «КількістьГодин» необхідно вибрати вкладку «Події», у полі «Після оновлення» – значення «Процедура обробки події» (рис. 5.10).

Далі за допомогою кнопки  потрібно відкрити вікно модуля, де слід відредагувати відповідну процедуру з назвою «КількістьГодин_AfterUpdate» (рис. 5.11).

Вигляд цієї процедури буде наступним:

СумаНарахована = КількістьГодин * ТарифЗаГодину.

Так само потрібно побудувати процедуру обробки події оновлення другого аргументу поля «Сума нарахована», а саме аргументу «ТарифЗаГодину».

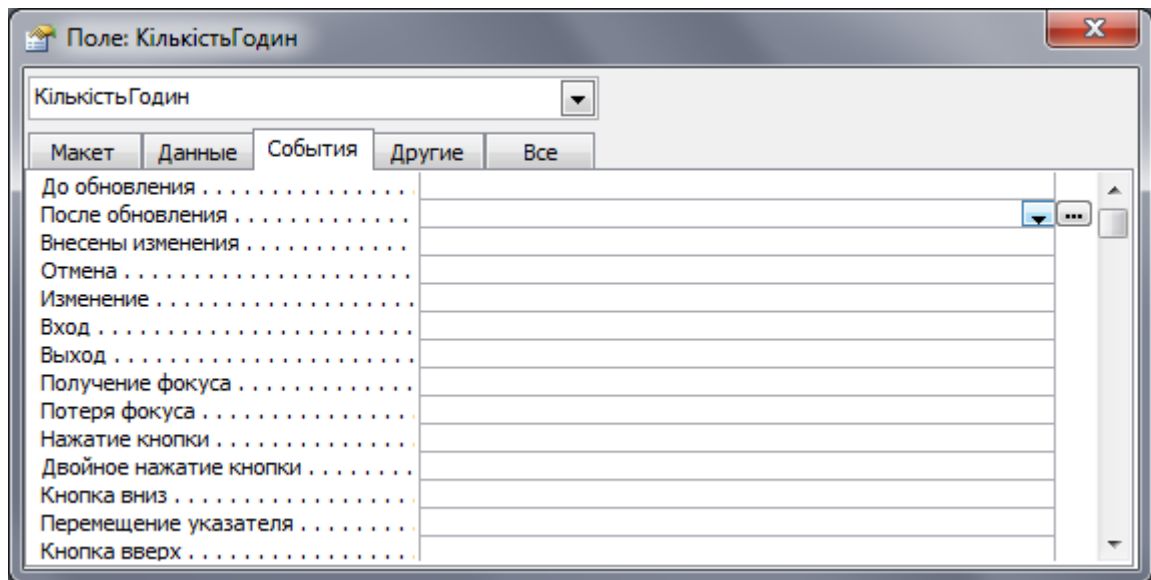


Рисунок 5.10 – Події поля «КількістьГодин»

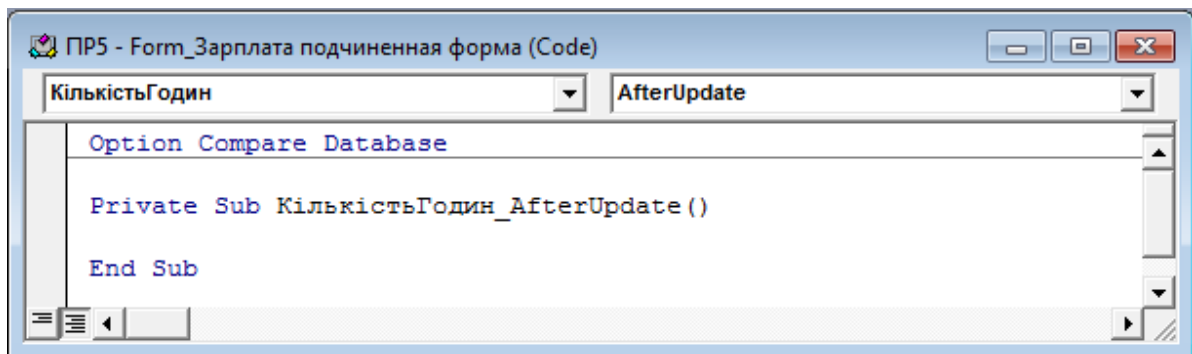


Рисунок 5.11 – Редагування процедури

Тепер слід побудувати другу процедуру з назвою «ТарифЗаГодину_AfterUpdate», де тіло процедури збігається з тілом процедури «КількістьГодин_AfterUpdate».

Значення інших полів, що обчислюються, також залежать від аргументів «КількістьГодин» і «ТарифЗаГодину». Це означає, що визначені вище процедури треба редагувати, додаючи нові інструкції. Редагування процедур виконують у тому ж вікні модуля, в яке можна потрапити через властивості поля й обробку подій. З урахуванням уже побудованого алгоритму:

СумаНарахована = КількістьГодин * ТарифЗаГодину

Dim s As Integer

s = Int(СумаНарахована)

Податок = $\min(85 - 17, \max(s - 17, 0)) * 10 / 100 + \min(170 - 85, \max(s - 85, 0)) * 15 / 100 + \min(1020 - 170, \max(s - 170, 0)) * 20 / 100 + \min(1700 - 1020, \max(s - 1020, 0)) * 30 / 100 + \max(s - 1700, 0) * 40 / 100$

$\text{ВнесокПФ} = \text{If}(\text{СумаНарахована} \leq 150, \text{СумаНарахована} * 1 / 100, \min(\text{СумаНарахована} * 2 / 100, 32))$

$\text{ВнесокФССбр} = \min(\text{СумаНарахована} * 0,5 / 100, 8)$

$\text{ВнесокФССл} = \text{If}(\text{СумаНарахована} \leq 150, \text{СумаНарахована} * 0,25 / 100, \min(\text{СумаНарахована} * 0,5 / 100, 8))$

$\text{СумаНаРуки} = \text{СумаНарахована} - (\text{Податок} - \text{ВнесокПФ} - \text{ВнесокФССбр} - \text{ВнесокФССл})$

Крім зазначених процедур у цьому ж модулі потрібно записати функції max і min:

Private Function max(a, b)

max = If(a >= b, a, b)

End Function

Private Function min(a, b)

min = If(a >= b, b, a)

End Function

Побудова звітів. Для розглянутої задачі звіт необхідно побудувати групуванням за полем «Посада» і сортуванням за полем «Прізвище». Повернувшись у вікно бази даних, слід перейти у вкладку «Звіти» й за допомогою кнопки «Створити» розпочати роботу з побудови звіту. Скористаємося майстром і після вибору варіанта технології побудови звіту визначимо базову таблицю.

На першому кроці майстер звіту пропонує визначити множину полів для вибраної таблиці або таблиць (рис. 5.12). Крім усіх полів таблиці «Зарплата» треба вибрати поля «Назва» й «ТарифЗаГодину» з таблиці «Посади», а також поля «Прізвище», «Ім'я», «ПоБатькові» і «Посада» з таблиці «Працівники».

На другому кроці майстер пропонує вибрати базову таблицю. Її вже вибрано на початку побудови звіту.

Третій крок – групування даних, для якого можна вибрати, наприклад, поле «Посада», що належить таблиці «Працівники».

Четвертий крок за майстром – сортування даних. Можна обмежитись одним рівнем сортування, за полем «Прізвище». На цьому етапі можна скористатися можливістю підбити підсумки за полями, що обчислюються, натиснувши кнопку «Підсумки» (рис. 5.13).

На п'ятому кроці обирається форма звіту, в тому числі його орієнтація. Оскільки кількість вибраних полів велика, тому, щоб розташувати рядки звіту на одній сторінці формату А4, слід вибрати альбомну орієнтацію звіту. Макет користувач вибирає самостійно.

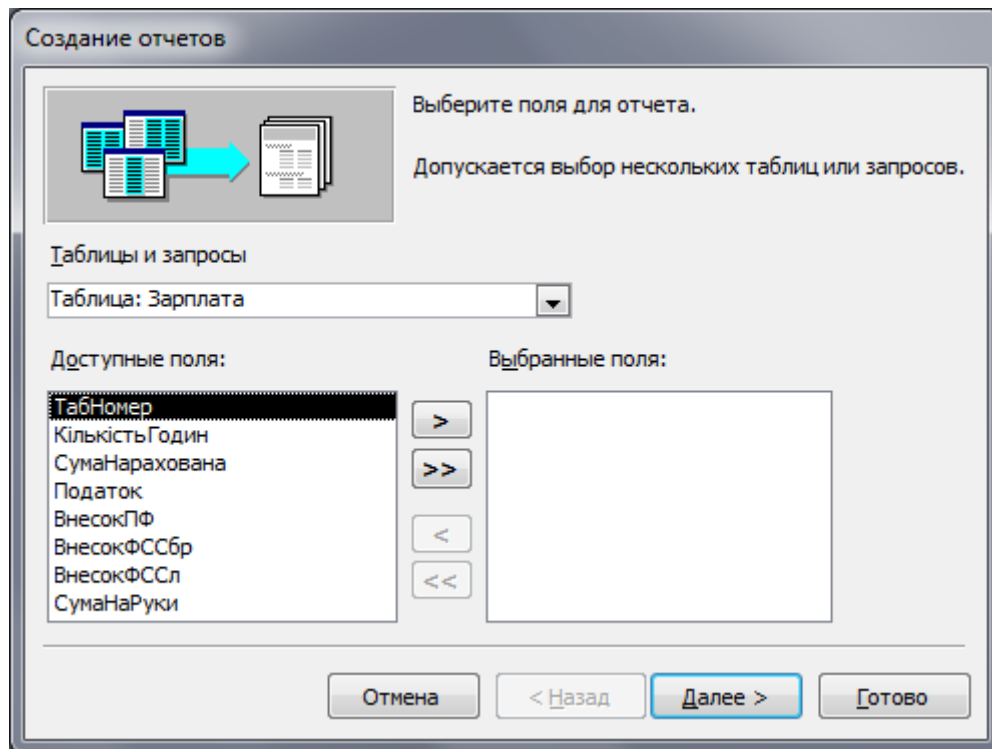


Рисунок 5.12 – Створення звітів

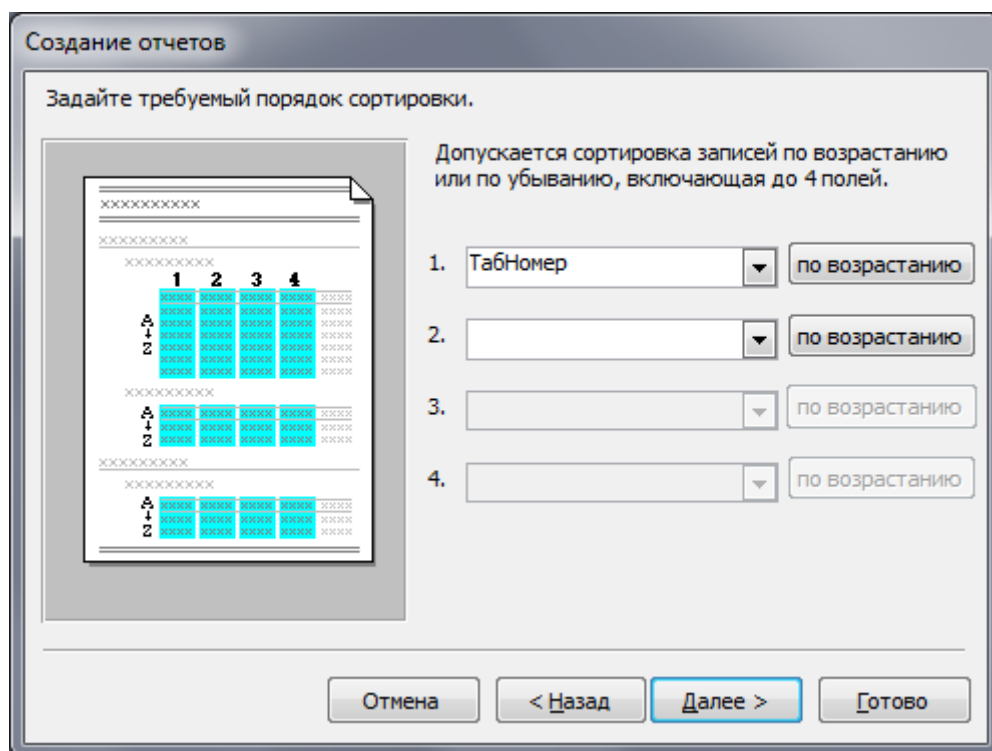


Рисунок 5.13 – Створення звітів (сортування)

Вибравши на шостому кроці майстра потрібний стиль, можна перейти до фінальної стадії діалогу, що складається з визначення назви звіту й вибору таких дій: або перегляд звіту, або його редагування.

Побудова запитів. Відповідно до постановки задачі потрібно створити запит щодо вибірки зарплати жінок. Повернувшись у вікно побудови бази даних, слід перейти у вкладку «Запити». Натиснувши кнопку «Створити», можна побудувати запит за допомогою «Конструктора». В результаті з'являється вікно «Запити: запит на вибірку», у верхній частині якого треба розташувати таблиці з потрібною інформацією, додавати таблиці можна у вікні «Додавання таблиць» (рис. 5.14).

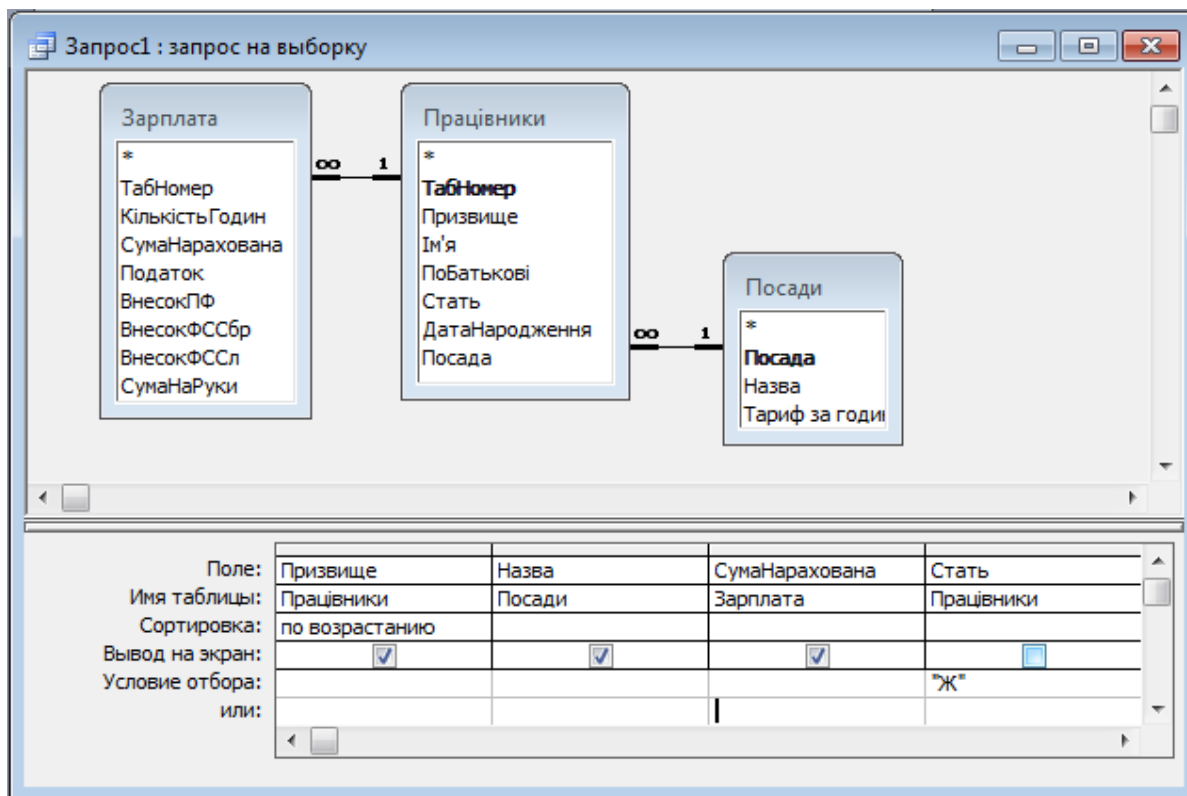


Рисунок 5. 14 – Створення запиту

У нижній частині слід вказати поля, яких стосується запит. Для прикладу вибрано чотири поля: «*Прізвище*», «*Назва*», «*СумаНарахована*» і «*Стать*». Сортування здійснюється за полем «*Прізвище*», а вибірка – за значенням «Ж» поля «*Стать*», яке на екран не виводиться. Після визначення назви запиту (наприклад, «*Жінки*») його можна відкрити, отриманий результат буде подано у вигляді таблиці.

Побудова кнопок форм. Необхідно побудувати кнопочку форму, з якої можна відкривати форми, звіти, запити. Можна розробити таку кнопочку форму, яка відкриватиме інші кнопочкові форми або безпосередньо розроблені об'єкти. Нехай головна кнопочка форма під назвою «*Приклад*» міститиме три кнопки: «*Довідники*», «*Облік*», «*Документ*». Нехай кнопка «*Довідники*» головної кнопочкової форми

відкриватиме іншу кнопкову форму (під назвою «Довідники»), яка міститиме дві кнопки «Посади» й «Працівники», функція яких – відкриття однойменних розроблених вище форм. Нарешті, нехай кнопки «Облік» і «Документ» головної кнопкової форми забезпечуватимуть відкриття форми «Зарплата» і запиту «Жінки».

Першою слід розробити форму «Довідники», оскільки вже до цієї форми звертатимуться з форми «Приклад». Відкривши вкладку «Форми», за допомогою кнопки «Створити» можна починати розробку. Необхідно вибрати режим «Конструктор», не вказуючи таблицю або запит.

Потім потрібно розмістити кнопку «Посади». Для цього слід ввімкнути панель інструментів, вибрати елемент «Кнопка», а також вказати мишею місце його розташування на кнопковій формі. Після цього майстер побудови елемента керування запропонує діалог.

Перший крок – вибір категорії і відповідної дії. Необхідно вибрати категорію «Робота з формою» й дію «Відкриття форми».

Другий крок – вибір форми. Слід вибрати форму «Посади».

Третій крок – вибір множини записів. Слід вибрати «Відкрити форму та показати всі записи».

Четвертий крок – вибір об'єкта розміщення на кнопці: текст або малюнок. Вибравши текст, слід змінити текст «Відкриття форми» на текст «Посади».

П'ятий крок – визначення назви елемента керування.

У результаті виконаних дій кнопка «Посади» готова. Аналогічно будуються кнопки та форми для зазначених елементів управління.

Питання для самоконтролю

1. Як створити підстановку з фіксованим набором значень?
2. Як додати таблиці чи запити до схеми даних?
3. Чи обов'язкове існування ключового поля у таблиці?
4. Для чого потрібні ключові поля?
5. Розкрийте основні властивості ключових полів таблиці.
6. Яким чином можливо автоматизувати обчислення значень полів?
7. Опишіть послідовність дій побудови запиту.
8. Як при побудові запиту можна задати видимість полів?
9. Як змінити фон кнопкової форми?
10. Як додати кнопку на кнопкову форму?
11. Для чого використовуються звіти?

ПРАКТИЧНА РОБОТА № 6

РОЗРОБКА БАЗИ ДАНИХ ДЛЯ ОБЛІКУ РУХУ ТОВАРІВ НА СКЛАДІ. ЗАГАЛЬНЕ ПОНЯТТЯ МОВИ SQL, СТВОРЕННЯ ЗАПИТІВ

Мета: здобути практичні навички у створенні запитів за допомогою команд мови SQL.

Завдання:

1. Створити базу даних обліку руху товарів на складі згідно із запропонованою схемою даних (рис. 6.1).

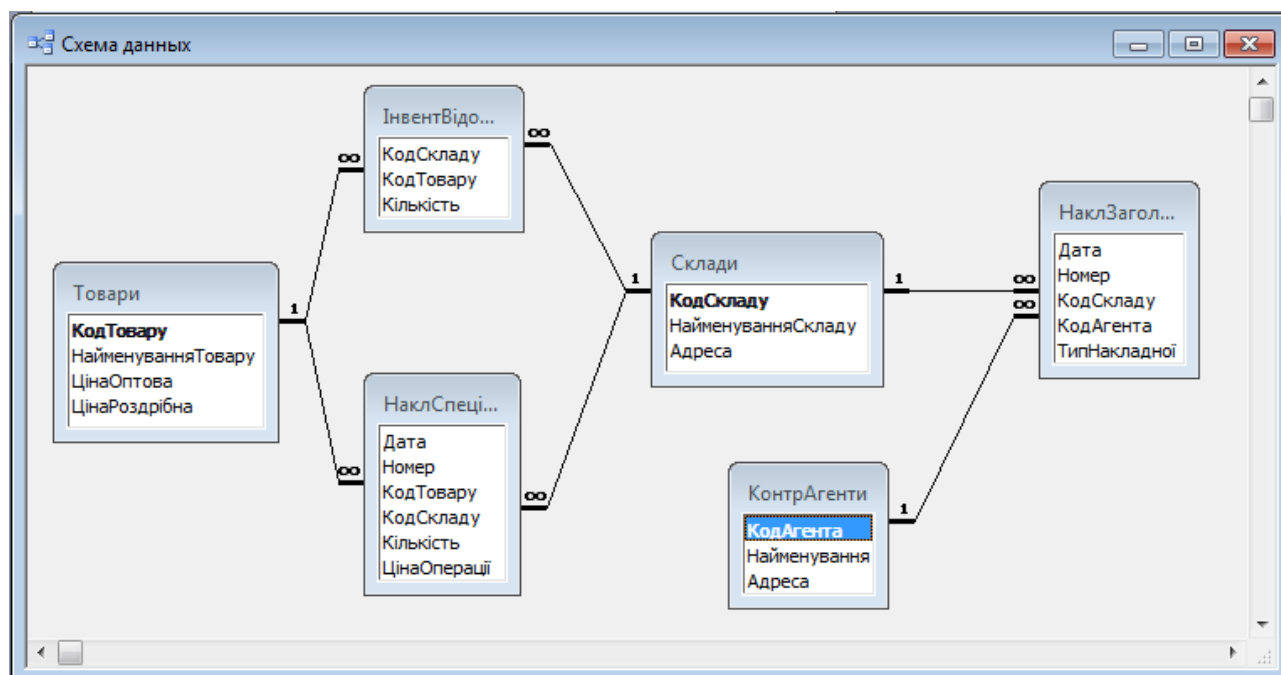


Рисунок 6.1 – Схема даних

2. Заповнити базу довільними даними згідно зі специфікою варіанта.
3. Ознайомитись з основними командами мови SQL та створити запити використовуючи всі розглянуті функції операторів.

Методичні вказівки до виконання практичної роботи

В ANSI SQL передбачено шість основних типів команд:

– команди мови визначення даних (data definition language – DDL) дозволяють створювати нові таблиці в базі даних, додавати індекси тощо, основні команди мови визначення даних представленні у таблиці 6.1.

Таблиця 6.1 – Команди «*DDL*»

Команди	Що дозволяють
CREATE TABLE	Створити таблицю
ALTER TABLE	Модифікувати таблицю
DROP TABLE	Видалити таблицю
CREATE INDEX	Створити індекс
ALTER INDEX	Модифікувати індекс
DROP INDEX	Видалити індекс

– команди мови обробки даних (data manipulation language – DML) використовуються для додавання, коректування та видалення рядків у таблицях (табл. 6.2).

Таблиця 6.2 – Команди «*DML*»

Команди	Що дозволяють
INSERT	Вставити дані в таблицю
UPDATE	Обновити дані
DELETE	Видалити дані

– команди мови запитів даних (data query language – DQL) використовується для отримання даних із таблиць й визначення форми представлення цих даних (табл. 6.3).

Таблиця 6.3 – Команди «*DQL*»

Команди	Що дозволяють
SELECT	Виконати запит із таблиць бази

– команди мови керування даними (data control language – DCL) визначають доступ користувачів і груп користувачів до об'єктів бази за допомогою повноважень, наданих і скасованих командами (табл. 6.4).

Таблиця 6.4 – Команди «*DCL*»

Команди	Що дозволяють
GRANT	Надати повноваження
REVOKE	Скасувати повноваження

– команди мови обробки транзакцій (transaction processing language – TPL) забезпечують оновлення всіх рядків, що використовуються в операторі «*DML*» (табл. 6.5).

Таблиця 6.5 – Команди «TPL»

Команди	Що дозволяють
BEGIN TRANSACTION	Почати транзакцію
COMIT TRANSACTION	Закінчити транзакцію
SAVE TRANSACTION	Створити точку збереження в транзакції

– команди мови керування курсором (cursor control language – CCL) виконують операції з окремими рядками однієї або декількох таблиць і включають: «*DECLARE CURSOR*», «*FETCH INTO*» і «*UPDATE WHERE CURRENT*».

Для роботи з базами даних у Visual Basic використовується мова запитів, що включає єдину інструкцію «*SELECT*». Розглянемо деякі питання, що пов'язані з її використанням.

Синтаксис інструкції «*SELECT*» у SQL Jet наступний:

```
SELECT [ALL | DISTINCT | DISTINCTROW | TOP] { * | table.* | [table. ]field1 [AS alias1] [, [table.]field2 [AS alias2] [, ...]] }
FROM table1 [table1Alias] [, table2 [table2Alias] [, ...]
[WHERE criteria]
[GROUP BY groupfieldlist ]
[HAVING groupcriteria]
[ORDER BY field1 [ASC | DESC ][, field2 [ASC | DESC ][, ...]]]
[WITH OWNERACCESS OPTION]
```

Інструкція «*SELECT*» включає наступні основні елементи:

– «*SELECT*» означає, що з деяких таблиць бази даних необхідно вибрати набір полів або таблицю даних;

– необов'язкові слова «*ALL*», «*DISTINCT*», «*DISTINCTROW*» і «*TOP*» називаються предикатами (predicates) і визначають вибір у такий спосіб:

а) «*ALL*» указує, що в набір передаються всі рядки (навіть з повторюваними значеннями);

б) «*DISTINCT*» указує, що в набір передаються тільки неповторювані рядки;

в) «*DISTINCTROW*» указує, що в набір буде включений кожний рядок, в якому є різниця в значенні будь-якого поля вихідних таблиць;

г) «*TOP*» використовується для відображення деякої кількості (точно або у відсотках) початкових або кінцевих записів з результатного набору;

– список { * | table.* | [table.]field1 [AS alias1], [table.]field2 [AS alias2] [, ...] } (дужки { } тут позначають список) складається з імен полів таблиць запиту. Зірочка «*» означає вибір усіх полів таблиці. Якщо в запиті указуються кілька таблиць, то для визначення поля використовується

назва таблиці, відокремлюване від імені поля крапкою «.». Поле може отримати «аліасне» ім'я за допомогою ключового слова «AS».

– після слова «FROM» указуються таблиці, з яких вибираються раніше вказані поля. Тут «Tableexpression» – це ім'я таблиці (або таблиць), що містить дані, «Externaldatabase» – ім'я бази даних, якщо не використовується поточна база.

Для тестування SQL-інструкцій у середовищі Access виберіть у лівій частині Головного вікна Access у меню «Об'єкти» опцію «Запити» й викликати команду «Створення запиту в режимі конструктора». Вікно «Додавання таблиці», що з'явилося, у даному випадку не потрібно, тому його варто закрити.

Не звертаючи уваги на засоби Access, призначені для створення запиту (у нижній частині діалогового вікна), виберіть команду «Режим SQL» в меню «Вид», щоб одержати доступ до вікна, у якому можна буде набирати SQL-інструкції.

Приклад 1.

```
SELECT *
FROM table
```

Тут знак «*» означає вибір значень усіх полів, а «Table» – ім'я таблиці, з якої вибираються значення. Тому найпростіша SELECT-інструкція однієї з розглянутих нами таблиць може виглядати так:

```
SELECT * FROM Товари
```

Уведіть цю інструкцію у вікно «Запит 1»: запит на вибірку й клацніть кнопку «Запуск» на панелі «Конструктор запитів».

Вибір у SQL-запиті визначених полів. Частіше усього не потрібно одержувати набір із усіма полями однієї (або більше) таблиці. Наприклад, оскільки поле «Код товару» в таблиці «Товари» призначено в основному для зв'язку з іншими таблицями, то не часто його слід включати в SQL-запит. У SELECT-інструкції можна вказати конкретні найменування полів деякої таблиці, використовуючи наступний синтаксис:

```
SELECT field1 [field2 [, ....]]
FROM table
```

Приклад 2.

```
SELECT НайменуванняТовару, ЦінаОптова
FROM Товари
```

Для обраних у запиті полів можна вказати відмінні від найменувань полів заголовки. Щоб вказати назви полів результуючого запиту, необхідно використовувати наступний синтаксис:

```
SELECT field1 [As alias1] [field2 [As alias2] [, ....]]
FROM table
```

Приклад 3.

```
SELECT НайменуванняТовару [Найменування товару] [,ЦінаОптова
[Ціна Оптова]
FROM Товари
```

Вибір у SQL-запиті визначених записів. Ключове слово «*WHERE*» у синтаксисі інструкції «*SELECT*» дозволяє вказувати визначені типи записів, що повинні попадати в набір. При цьому інструкція «*SELECT*» має наступний синтаксис:

```
SELECT { * |field1 [AS alias1] [,field2 [AS alias2] [, ...]]}
FROM table
[WHERE criteria]
```

Приклад 4. Вибрати тільки ті товари, ціни яких не перевищують 100 грошових одиниць:

```
SELECT НайменуванняТовару as [Найменування товару],
ЦінаОптова as [Ціна Оптова]
FROM Товари
WHERE ЦінаОптова < 100
```

Використання у SQL-запиті функцій для виведення певних записів. В області слова «*WHERE*» можна відобразити досить складний вираз з використанням знаків логічних операцій.

Приклад 5. Вибрати товари, оптова ціна яких більше 50 та довжина назви яких не перевищує 20 символів:

```
SELECT НайменуванняТовару AS [Найменування товару],
ЦінаОптова as [Ціна Оптова]
FROM Товари
WHERE ЦінаОптова > 90 AND Len(НайменуванняТовару) < 20
```

Крім операцій «>» і «<», в інструкції «*SELECT*» можна використовувати операції «=» («дорівнює»), «< =» («менше або дорівнює») і «> =» («більше або дорівнює»), а також «*AND*», «*OR*» і «*NOT*». Великі можливості по добору необхідних записів надають такі оператори, як «*IS NULL*», «*BETWEEN*», «*IN*» і «*LIKE*».

Приклад 6. Оператор «*IS NULL*» дозволяє знайти в таблиці записи, у полях яких не вказані дані:

```
SELECT НайменуванняТовару AS [Найменування товару],
ЦінаОптова AS [Ціна Оптова]
FROM Товари
WHERE ЦінаРоздрібна IS NULL OR ЦінаОптова < 50
```

Приклад 7. Оператор «*BETWEEN*» дозволяє вказати діапазон, у якому знаходяться дані якогось поля:

```
SELECT НайменуванняТовару as [Найменування товару],
ЦінаОптова as [Ціна Оптова]
```

```
FROM Товари
WHERE ЦінаОптова BETWEEN 42 AND 120
```

Приклад 8. Оператор «*IN*» дозволяє вказати список, у якому знаходяться дані деякого поля:

```
SELECT НайменуванняТовару as [Найменування товару],
ЦінаОптова as [Ціна оптова]
FROM Товари
WHERE ЦінаОптова IN (42, 105, 750)
```

Приклад 9. Використання оператору «*LIKE*»:

```
SELECT НайменуванняТовару as [Найменування товару],
ЦінаОптова as [Ціна оптова]
FROM Товари
WHERE НайменуванняТовару LIKE «(VCD)*»
```

Цей запит дозволяє одержати тільки ті записи, у яких у найменуванні перші три символи збігаються з рядком «(VCD)».

Приклад 10. Оператор «*NOT*», що інвертує логічне вираження, може використовуватися з операторами «*IS NULL*», «*BETWEEN*», «*IN*», «*LIKE*». При цьому, практично використовувати оператори «*IS NOT NULL*», «*NOT BETWEEN*», «*NOT IN*» і «*NOT LIKE*»:

```
SELECT НайменуванняТовару AS [Найменування товару],
ЦінаОптова AS [Ціна оптова]
FROM Товари
WHERE ЦінаРоздрібна IS NOT NULL OR ЦінаОптова < 50
```

Вибір даних з більш ніж однієї таблиці. Маючи базу даних з декількох таблиць, ми дотепер одержували за допомогою інструкції «*SELECT*» набори даних тільки з однієї таблиці. Розглянемо задачу вибору з бази даних товарів деякого складу. Перед тим як роздивимось використання слова «*WHERE*» для зв'язку таблиць, помітимо, що в інструкції «*SELECT*» можна перед ім'ям поля вказувати ім'я таблиці, що відокремлюється від імені поля крапкою:

```
SELECT { * | [table1. ] field1 [AS alias1] [, [table2 . ] field2 [AS
alias2] [,...]] }
FROM table1 [table1Alias] [, table2 [table2Alias] [, ...]
[WHERE criteria]
```

Приклад 11.

```
SELECT Товари.НайменуванняТовару AS [Найменування товару],
Товари.ЦінаОптова AS [Ціна оптова]
FROM Товари
```

Приклад 12.

```
SELECT а.НайменуванняТовару AS [Найменування товару],
а.ЦінаОптова AS [Ціна оптова]
```

FROM Товари a

В прикладі 12 використане альтернативне ім'я таблиці (локальний псевдонім), але обидві інструкції видають той самий результуючий набір.

Приклад 13. Одержати найменування (з таблиці «Товари») й відповідну кількість товару (з таблиці «ІнвентВідомість»):

```
SELECT Товари.НайменуванняТовару AS [Найменування товару],
ІнвентВідомість.Кількість AS [Кількість]
```

```
FROM Товари, ІнвентВідомість
```

```
WHERE Товари.КодТовару=ІнвентВідомість.КодТовару
```

ЗВЕРНІТЬ УВАГУ!

Вираз «Товари.КодТовару = ІнвентВідомість.КодТовару»

використовується для зв'язку таблиць «Товари» й «ІнвентВідомість», по ключовому полю «КодТовару».

Приклад 14. Одержати найменування (з таблиці «Товари») й кількість (із таблиці «ІнвентВідомість») тільки для тих кодів товарів, записи яких в таблиці «ІнвентВідомість» у полі «Код_складу» містять рядок «0429»:

```
SELECT Товари.НайменуванняТовару AS [Найменування товару],
ІнвентВідомість.Кількість AS [Кількість]
```

```
FROM Товари, ІнвентВідомість
```

```
WHERE ІнвентВідомість.Код_складу='0429' AND Товари.КодТовара
= ІнвентВідомість.КодТовара
```

За допомогою локальних псевдонімів можна скоротити попередню інструкцію:

```
SELECT a.НайменуванняТовару AS [Найменування товару],
b.Кількість AS [Кількість]
```

```
FROM Товари a, ІнвентВідомість b
```

```
WHERE b.Код_складу='0429' AND a.КодТовару=b.КодТовару
```

Варто помітити, що взагалі ж коди, як товарів, так і складів, не призначені для використання клієнтами. Вони необхідні для зв'язку між таблицями. Часто менеджери, що використовують бази даних, навіть не знають про наявність кодів у таблицях баз даних. По цій або подібній їй причині слід позбавитися від наглядного використання коду складу в SELECT-інструкції, при чому інструкція стане дещо складнішою.

Приклад 15.

```
SELECT a.НайменуванняТовару AS [Найменування товару],
b.Кількість AS [Кількість]
```

```
FROM Товари a, ІнвентВідомість b, Склади c
```

```
WHERE a.КодТовару=b.КодТовару AND b.КодСкладу= c.КодСкладу
AND c.НайменуванняСкладу= «Склад(Мультимедія)»
```

У цій інструкції використана таблиця «Склади», а за допомогою поля «КодСкладу» ця таблиця зв'язана з таблицею «ІнвентВідомість».

Представлення обраних даних у визначеному порядку. Найменування товарів у таблицях і виведених наборах не відсортовані. Це буває особливо помітно, коли даних досить багато (наприклад, у вікні списку), а потрібно знайти тільки деякі з них.

Для сортування даних в інструкції «*SELECT*» є ключове слово «*ORDER BY*»:

```
SELECT { *| [table.]field1 [AS alias1] [, [table.]field2 [AS alias2] [, ...]] }
FROM table1 [table1Alias] [, table2 [table2Alias] [, ...]
[WHERE criteria]
[ORDER BY field1 [ASC | DESC] [, field2 [ASC | DESC] ][,...]]
```

Використовуючи слова «*ASC*» і «*DESC*», можна змінювати «напрямок» сортування («за зростанням» і «за спаданням»). Сортувати можна по декількох полях і навіть по різних елементах того самого поля з використанням функцій (як убудованих, так і користувацьких).

Приклад 16. Сортування виведеного набору за найменуваннями:

```
SELECT a.НайменуванняТовару AS [Найменування товару],
b.Кількість AS [Кількість]
FROM Товари a, ІнвентВідомість b
WHERE a.КодТовару=b.КодТовару AND b.Код_складу='0429'
ORDER BY a.НайменуванняТовару
```

Агрегуючі функції в інструкції «*SELECT*» визначають кількість записів, обчислюють суми всіх значень полів у наборі, знаходять мінімальні або максимальні, а також середні значення. До агрегуючих функцій відносяться «*COUNT*», «*SUM*», «*MAX*», «*MIN*», «*AVG*».

Функції «*COUNT*» використовуються для визначення кількості записів у запиті.

Приклад 17. Знаходження кількості записів в таблиці «Товари»:

```
SELECT COUNT (НайменуванняТовару) AS [Всього найменувань]
FROM Товари
```

Приклад 18. Знаходження кількості записів, у яких значення поля «ЦінаОптова» менше 100:

```
SELECT COUNT(НайменуванняТовару) AS [Усього найменувань]
FROM Товари
WHERE ЦенаОптовая < 100
```

Приклад 19. Знаходження підсумкове значення поля:

```
SELECT SUM(Кількість) AS [Загальна кількість товарів]
FROM ІнвентВідомість
```

Приклад 20. Обмеження кількості записів при використанні функції «*SUM*» за допомогою умови:


```
SELECT SUM(Кількість) AS [Кількість на складу]
FROM ІнвентВідомість
WHERE КодСклада='0429'
```

Функція «*AVG*» в інструкції «*SELECT*» дозволяє знайти середнє значення рядків, що входять у запит.

Приклад 21. Визначення середньої ціни на товари, найменування яких починаються з набору символів «(VCD)»:

```
SELECT AVG (ЦінаОптова) AS [Середнє значення]
FROM Товари
WHERE НайменуванняТовару LIKE '(VCD)*'
```

Приклад 22. Знаходження максимального значення ціни на товари, найменування яких починаються з набору символів «(VCD)»:

```
SELECT MAX (ЦінаОптова) AS [Максимальна ціна]
FROM Товари
WHERE НайменуванняТовару LIKE '(VCD)*'
```

Групування даних в інструкції «SELECT». Повторювані значення можна поєднувати в групи, використовуючи вираз «*GROUP BY*».

Приклад 23. Формування списку складів і сумарної кількості товарів на кожному складі:

```
SELECT с.НайменуванняСкладу, SUM (b.Кількість) AS Кількість
FROM ІнвентВідомість b, Склади с
WHERE b.КодСкладу=с.КодСкладу
GROUP BY с.НайменуванняСкладу
```

Приклад 24. Формування списку складів і знаходження сумарної кількості товарів на них. При чому в список включаються тільки ті склади, на котрих сумарна кількість товарів менше, ніж 1000:

```
SELECT с.НайменуванняСкладу, SUM (b.Кількість) AS Кількість
FROM ІнвентВідомість b, Склади с
WHERE b.КодСкладу=с.КодСкладу
GROUP BY с.НайменуванняСкладу
HAVING SUM (b.Кількість) <1000
```

Підзапит інструкції «SELECT» – це запит, що розміщується усередині іншого запиту, є частиною виразу «*WHERE*» (або «*HAVING*») основного запиту і береться у дужки «()». Іноді підзапити називають *вкладеними запитами*. Запит, що містить підзапит є *складним запитом*, при виконанні якого спочатку виконується вкладений, а потім – основний запит.

Вихідні дані підзапиту звичайно використовуються при оцінці виразу «*WHERE*» (або «*HAVING*») основного запиту з використанням таких операторів, як «*=*», «*>*», «*<*», «*<>*», «*IN*», «*NOT IN*», «*AND*», «*OR*».

Правила написання складних запитів:

- вираз «*SELECT*» звичайно містить тільки один стовпець, з чого випливає, що результатом може бути масив однорідних значень (тільки рядкові, тільки цілі тощо) або єдине значення;
- вираз «*ORDER BY*» варто використовувати тільки в основному запиті. У підзапиті для упорядкування даних застосовують вираз «*GROUP BY*»;
- неприпустиме використання в основному запиті виразу «*BETWEEN*».

Простий синтаксис підзапиту для оператора «*SELECT*» наступний:

```
SELECT { * | table.* | [table.] field1 [AS alias1] [, [table. ] field2
[AS alias2] [,...]]}
FROM table1 [table1Alias] [, table2 [table2Alias] [, ...]
WHERE field1 operator (SELECT { * | table.* | [table.]field1 [AS
alias1] [, [table.]field2 [AS alias2] [, ...]]} FROM table1 [table1Alias] [, table2
[table2Alias] [, ...] WHERE [criteria])
```

Приклад 25. Одержання списку товарів складу, на якому їх більше всього. Спочатку створимо запит, котрий поверне код складу з найбільшою кількістю товарів:

```
SELECT TOP 1 КодСкладу
FROM ІнвентВідомість GROUP BY КодСкладу ORDER BY SUM
(Кількість) DESC
```

Вираз «*TOP 1*» указує на те, що в результуючий набір потрапляє тільки один запис, а оскільки набір сортується (за спаданням) по сумарній кількості товарів, то в результаті одержуємо код складу з найбільшою кількістю товарів. Залишається використати раніше розглянутий запит, що повертає список товарів для зазначеного коду складу, але якщо раніше вказували цей код, то тепер визначаємо його в підзапиті:

```
SELECT a.НайменуванняТовару AS [Найменування товару], b.
Кількість AS Кількість
FROM Товари AS a, ІнвентВідомість AS b
WHERE a.КодТовару=b.КодТовару AND b.КодСкладу = (SELECT
TOP 1 КодСкладу FROM ІнвентВідомість GROUP BY КодСкладу
ORDER BY SUM(Кількість) DESC)
```

Варіанти завдань

1. Продукти харчування.
2. Побутова техніка.
3. Жіночий одяг.
4. Будівельні матеріали.

5. Побутова хімія.
6. Меблі.
7. Мобільні пристрої.
8. Медичні препарати.
9. Канцелярські вироби.
10. Спортивні товари.
11. Промислові товари.
12. Фототовари.
13. Кондитерські вироби.
14. Комп'ютерна техніка.
15. Книги.
16. Чоловічий одяг.
17. Обігрівачі.
18. Іграшки.
19. Посуд.
20. Керамічні вироби.

Питання для самоконтролю

1. Назвіть шість основних типів команд «*ANSI SQL*».
2. Перелічіть команди «*DDL*», та назвіть що вони дозволяють зробити.
3. Для чого використовуються команди «*DML*»?
4. Що дозволяє зробити команда «*SELECT*»? До якого типу команд вона відноситься?
5. Назвіть команди мови керування даними. Для чого вони використовуються?
6. Який тип команд забезпечує оновлення всіх строк, що використовуються в операторі «*DML*»?
7. Назвіть основні елементи інструкції «*SELECT*».
8. Які логічні операції можна використовувати в інструкції «*SELECT*»?
9. Перелічіть агрегуючі функції інструкції «*SELECT*».
10. Який вираз в інструкції «*SELECT*» використовується для об'єднання значень у групи?
11. Розкрийте суть поняття «Підзапит».
12. Розкажіть, які правила написання складних запитів Ви знаєте.

СПИСОК ЛІТЕРАТУРИ

1. Ананьев О. М. Інформаційні системи і технології в комерційній діяльності: підручник / О. М. Ананьев, В. М. Бшик, Я. А. Гончарук. – Л.: Новий Світ-2000, 2006. – 584 с.
2. Оксанич А. П. Інформаційні системи і технології маркетингу: навч. посіб. / А. П. Оксанич, В. Р. Петренко, О. П. Костенко. – К.: Видавничий дім «Професіонал», 2008. – 320 с.
3. Інформаційні системи і технології в економіці: посібник для студентів вищих навчальних закладів / за ред. В. С. Пономаренка. – К.: Видавничий центр «Академія», 2002. – 544 с.
4. Інформаційні системи в менеджменті: навчальний посібник / С. В. Глівенко, Є. В. Лапін, О. О. Павленко. – Суми: ВТД «Університетська книга», 2005. – 407 с.
5. Рогач І. Ф. Інформаційні системи в фінансово-кредитних установах: навч. посібник. – 2-ге вид. / І. Ф. Рогач, М. А. Сендзюк, В. А. Антонюк. – К.: КНЕУ, 2001. – 239 с.
6. Исаев Г. Н. Информационные системы в экономике: учебное пособие / Г. Н. Исаев. – М.: Омега-Л, 2006. – 462 с.
7. Інформаційні системи і технології: навч. посіб. для студ. вищ. навч. закл. / С. Г. Карпенко, В. В. Попов, Ю. А. Тарнавський, Г. А. Шпортюк. – К.: МАУП, 2004. – 192 с.
8. Білик В. М. Інформаційні технології та системи: навч. посіб. / В. М. Білик, В. С. Костирко. – К.: Центр навчальної літератури, 2006. – 232 с.
9. Система управління базами даних Microsoft Access для самостійного вивчення: навч. посібник / Н. В. Баловсяк, І. А. Григоришин, Л. В. Кулібаба. – К.: Дакор, КНТ, 2006. – 156 с.
10. Информационные системы в экономике: учебник для студентов вузов / под ред. Г. А. Титоренко. – 2-е изд., перераб. и доп. – М.: ЮНИТИ, 2008. – 463 с.
11. Герасимов В. В. Информационные технологии производственных систем: учеб. пособие / В. В. Герасимов, Л. С. Минина, А. В. Васильева. – Новосибирск: НГАСУ, 2001. – 74 с.
12. Бобцов А. А. Банки и базы данных. Основы работы в MS Access. Часть 1 (для пользователей): учебное пособие / А. А. Бобцов, В. В. Шиегин. – СПб., 2005. – 93 с.
13. Бобцов А. А. Банки и базы данных. Основы работы в MS Access. Часть 2 (для разработчиков): учебное пособие / А. А. Бобцов, В. В. Шиегин. – СПб., 2005. – 57 с.
14. Золотова С. И. Практикум по Access / С. И. Золотова. – М.: Финансы и статистика, 2003. – 144 с.

15. Гончаров А. Ю. ACCESS 2003. Самоучитель с примерами / А. Ю. Гончаров. – М.: КУДИЦ-ОБРАЗ, 2004. – 272 с.
16. Райордан Р. Основы реляционных баз данных / Р. Райордан; пер. с англ. – М.: Издательско-торговый дом «Русская Редакция», 2001. – 384 с.
17. Харрингтон Д. Л. Проектирование реляционных баз данных / Д. Л. Харрингтон; пер. с англ. И. Дранишников. – М.: Лори, 2006. – 230 с.
18. Корнеев И. К. Информационные технологии в управлении / И. К. Корнеев, В. А. Машурцев. – М.: ИНФРА-М, 2001. – 158 с.
19. Тютюнник А. В. Информационные технологии в банке / А. В. Тютюнник, А. С. Шевелев. – К.: Издательская группа «БДЦ-пресс», 2003. – 224 с.
20. Лепа Р. Н. Информационные технологии в финансовом менеджменте. Курс лекций. Часть 1 / Р. Н. Лепа. – Донецк: ДИЭХП, 2001. – 86 с.
21. Информационные технологии управления: учебное пособие / под ред. Ю. М. Черкасова. – М.: ИНФРА-М, 2001. – 216 с.
22. Карпенко М. Ю. Інформаційні системи в менеджменті. Курс лекцій (для студентів 4 курсу за напрямом 0502 (6.030601) «Менеджмент») / М. Ю. Карпенко, В. Б. Уфимцева. – Х.: ХНАМГ, 2008. – 89 с.
23. Игнатьева А. В. Исследование систем управления: учеб. пособие для вузов / А. В. Игнатьева, М. М. Максимцов. – М.: ЮНИТИ-ДАНА, 2000. – 157 с.
24. Шуремов Е. Л. Информационные технологии финансового планирования и экономического анализа: практическое пособие. – 2-е изд. / Е. Л. Шуремов. – М.: ООО «1С-Паблишинг», 2003. – 195 с.
25. Смирнова Г. Н. Проектирование экономических информационных систем: учебник / Г. Н. Смирнова, А. А. Сорокин, Ю. Ф. Тельнов; под ред. Ю. Ф. Тельнова. – М.: Финансы и статистика, 2003. – 512 с.
26. Кузнецов С. Д. СУБД (системы управления базами данных) и файловые системы / С. Д. Кузнецов. – М.: Букпресс, 2006. – 174 с.
27. Йегер И. Упростите свои информационные технологии. Новые шансы для вас и вашего предприятия / И. Йегер, В. Т. Кюстенмахер, Р. Шуманн; пер. с нем. – СПб.: ДИЛЯ, 2007. – 128 с.
28. Олсон Г. Цифровые системы автоматизации управления / Г. Олсон, Д. Пиани. – СПб.: Невский Диалект, 2001. – 557 с.
29. Гламаздин Е. С. Управление корпоративными программами: информационные системы и математические модели / Е. С. Гламаздин, Д. А. Новиков, А. В. Цветков. – М.: ИПУ РАН, 2003. – 159 с.
30. Перешивкин С. А. Информационные технологии управления персоналом: учебное пособие / С. А. Перешивкин. – СПб.: Изд-во СПбГУЭФ, 2010. – 48 с.

31. Плотоненко Ю. А. Автоматизированные системы обработки экономической информации: учебное пособие. Ч.1 / Ю. А. Плотоненко. – Тюмень: Издательство Тюменского государственного университета, 2008. – 136 с.
32. Плотоненко Ю. А. Автоматизированные системы обработки экономической информации: учебное пособие. Ч.2 / Ю. А. Плотоненко. – Тюмень: Издательство Тюменского государственного университета, 2008. – 156 с.
33. Системи оброблення економічної інформації: навчально-метод. посіб. для самост. вивч. дисц. / за заг. ред. В. Ф. Ситника. – К.: КНЕУ, 2004. – 332 с.
34. Страхарчук Я. А. Інформаційні системи і технології в банках: навч. посіб. / Я. А. Страхарчук, В. П. Страхарчук. – К.: УБС НБУ: Знання, 2010. – 515 с.

ЕЛЕКТРОННЕ НАВЧАЛЬНО-МЕТОДИЧНЕ ВИДАННЯ

Гуменюк Наталія Володимирівна
Ніколаєнко Денис Володимирович

**МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИКОНАННЯ ПРАКТИЧНИХ РОБІТ З ДИСЦИПЛІНИ
«СИСТЕМИ ОБРОБЛЕННЯ ЕКОНОМІЧНОЇ ІНФОРМАЦІЇ»
(ДЛЯ СТУДЕНТІВ СПЕЦІАЛЬНОСТІ 6.030502
«ЕКОНОМІЧНА КІБЕРНЕТИКА»)**

Підписано до випуску 13.03.2012 р. Гарнітура Times New.
Умов. друк. арк. 3,94 Зам. № 86.

Державний вищий навчальний заклад
«Донецький національний технічний університет»
Автомобільно-дорожній інститут
84646, м. Горлівка, вул. Кірова, 51
E-mail: druknf@rambler.ru
Редакційно-видавничий відділ

Свідоцтво про внесення до Державного реєстру видавців, виготовників і
розповсюджувачів видавничої продукції ДК № 2982 від 21.09.2007 р.

