

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ

Навчально–науковий інститут економіки та управління

Обліково–фінансовий факультет

Кафедра інформаційних систем і технологій



**ТЕХНОЛОГІЯ ПРОЕКТУВАННЯ ТА АДМІНІСТРУВАННЯ
БАЗ ДАНИХ І СИСТЕМ ДАНИХ**

Методичні рекомендації

до виконання практичних робіт для здобувачів вищої освіти ступеня
«бакалавр» 3 курсу напряму підготовки 6.030502 «Економічна
кібернетика» денної форми навчання

МИКОЛАЇВ

2017

Друкується за рішенням науково-методичної комісії обліково-фінансового факультету Миколаївського національного аграрного університету від «13» квітня 2017 р., протокол № 11.

Укладач:

Ю. В. Волосюк – канд. техн. наук, доцент, завідувач кафедри інформаційних систем і технологій Миколаївського національного аграрного університету

Рецензенти:

І. П. Атаманюк – д-р техн. наук, професор, завідувач кафедри вищої та прикладної математики Миколаївського національного аграрного університету;

Д. М. Самойленко – канд. ф.-м. наук, доцент кафедри електрообладнання суден та інформаційної безпеки Національного університету кораблебудування імені адмірала Макарова.

ЗМІСТ

ВСТУП.....	3
ЗАГАЛЬНІ ВИМОГИ ДО ВИКОНАННЯ ПРАКТИЧНИХ РОБІТ	4
ПРАКТИЧНІ РОБОТИ.....	6
Практична робота №1. СУБД Microsoft Access 2010.	
Характеристика та інтерфейс програми.....	6
Практична робота №2. Створення баз даних, таблиць.....	12
Практична робота №3. Створення запитів	24
Практична робота №4. Робота з формами	36
Практична робота №5. Створення звітів.....	43
Практична робота №6. Імпорт, експорт даних.....	51
Практична робота №7. Мова запитів SQL.....	53
ЗАПИТАННЯ ДЛЯ САМОКОНТРОЛЮ	58
ВАРІАНТИ ПРАКТИЧНИХ ЗАВДАНЬ	59
ЗРАЗКИ ТЕСТОВИХ ЗАВДАНЬ	64
КРИТЕРІЇ ОЦІНКИ ЗНАНЬ.....	66
РЕКОМЕНДОВАНА ЛІТЕРАТУРА	68

ВСТУП

На сьогодні основними напрямками використання засобів обчислювальної техніки та програмних продуктів для інформаційного забезпечення роботи підприємства являються:

- використання існуючих автоматизованих інформаційних систем та їх конфігурація з урахуванням типу підприємства і класу задач;
- розробка нових інформаційних систем для розв'язання конкретних задач.

Інформаційною системою називається програмно-апаратний комплекс, який забезпечує надійне збереження інформації в пам'яті комп'ютера, виконання специфічних операцій перетворення інформації, надання користувачу зручного інтерфейсу.

Ефективним способом розв'язання цих задач являються бази даних. Розробка бази даних вимагає серйозного підходу, забезпечуючи ефективне зростання продуктивності та високі конкурентні переваги перед іншими компаніями. Комплексний підхід до розробки і сучасні технології, що повинні використовуватися, дозволять підприємству досягнути нових результатів після впровадження бази даних. Для виконання цих завдань необхідні спеціалісти з відповідними знаннями та вміннями, що забезпечуються вивченням навчальної дисципліни «Технологія проектування та адміністрування БД і СД», яка є нормативною і входить до циклу дисциплін професійної підготовки.

Метою вивчення дисципліни «Технологія проектування і адміністрування БД і СД» є формування базових знань з технології проектування та адміністрування баз даних і сховищ даних та придбання практичних навичок розробки автоматизованих баз даних автоматизованих банків даних і сховищ даних.

Під час складання методичних рекомендацій було використано програмне забезпечення MS Access 2010 та MySQL 5.5, тому всі ілюстрації та команди відповідають зазначеному програмному забезпеченню. Зауважимо, що викладений у цих методичних рекомендаціях матеріал придатний до реалізації і в більш пізніх версіях MS Access та MySQL: відмінність ілюстрацій при цьому може полягати у нюансах інтерфейсів користувача раніших версій (можуть

відрізнятися назви та вигляд діалогових вікон, елементів керування; назви та шлях доступу до деяких команд). Однак зазначене не має істотного впливу на досягнення заявленої в практичних роботах мети.

ЗАГАЛЬНІ ВИМОГИ ДО ВИКОНАННЯ ПРАКТИЧНИХ РОБІТ

Цикл практичних робіт із дисципліни «Технологія проектування та адміністрування БД і СД» для студентів напряму підготовки 6.030502 «Економічна кібернетика» спрямований на поглиблення й закріплення теоретичних знань, набутих студентами під час лекційних занять.

У результаті виконання практичних робіт студент повинен отримати чітке уявлення про реалізацію теоретичних положень дисципліни на практиці:

- зокрема, на прикладі побудови бази даних для обраної предметної області та додатку роботи з базою даних сформувати навички створення БД і простих автоматизованих інформаційних систем (АІС) з використанням БД;
- сформувати вміння та навички проектування інфологічної та концептуальної моделі БД у формі DFD та ER-діаграми;
- набути знання, вміння та навички використання інструментальних засобів СУБД MS Access та MySQL щодо розв'язання економічних та управлінських задач.

Кожна практична робота складається з наступних складових:

- теми роботи;
- мети роботи;
- умови завдання, що містить опис задачі;
- покрокових інструкцій щодо побудови засобами СУБД MS Access та MySQL бази даних для обраної предметної області та додатку роботи з БД;
- приміток, зауважень щодо особливостей застосування пропонованих засобів зазначених СУБД;
- завдань для самостійної роботи;
- контрольних питань, що дозволяють студенту підготуватися до захисту практичної роботи.

За результатами виконання кожної практичної роботи студенти складають звіт, який повинен містити:

- номер практичної роботи;
- тему практичної роботи;
- мету виконання практичної роботи;
- короткі теоретичні відомості щодо засобів СУБД, використаних у роботі;
- результати виконання завдань у вигляді рисунків створених об'єктів додатку;
- відповіді на питання, що наведені в практичній роботі;
- висновки по роботі.

Виконана належним чином практична робота захищається шляхом співбесіди з викладачем. За результатами захисту викладач зараховує практичну роботу як виконану і виставляє відповідні бали.

ПРАКТИЧНІ РОБОТИ

ПРАКТИЧНА РОБОТА №1. СУБД Microsoft Access 2010.

Характеристика та інтерфейс програми.

Мета роботи:

Ознайомитись з інтерфейсом та функціональними можливостями системи управління базами даних Microsoft Access 2010.

Теоретичні відомості

Основні поняття реляційних баз даних.

Сучасні інформаційні системи характеризуються великими об'ємами даних, що зберігаються, їх складною організацією, а також високими вимогами до швидкості і ефективності обробки цих даних. Це стає можливим при використанні спеціальних програмних засобів – систем управління базами даних (СУБД).

***Бази даних (БД)** – це сукупність відомостей про реальні об'єкти, процеси, події або явища, що відносяться до певної предметної області, організована таким чином, щоб забезпечити зручне представлення цієї сукупності як в цілому, так і будь-якої її частини.*

***Система управління базами даних** – це комплекс програмних і мовних засобів, необхідних для структуризації інформації, розміщення її в таблицях, обробки баз даних і підтримки їх в актуальному стані.*

Основні **функції** СУБД — це визначення даних (опис структури баз даних), обробка даних і управління даними. В даний час налічується велика кількість СУБД для персональних комп'ютерів. До найпоширеніших типів СУБД відносяться: MS SQL Server, Oracle, MySQL, SQLite, Interbase, MS Access тощо.

Існують різноманітні моделі даних – списки (плоскі таблиці), реляційні бази даних, ієрархічні, мережні структури, об'єктно-орієнтовані, документо-орієнтовані, key-value модель та ін.

Протягом багатьох років переважно використовувалися плоскі таблиці типу списків в Excel, вся інформація яких містилася в одній таблиці, ніяк не зв'язаній з іншими таблицями, що розміщені в інших файлах.

В ієрархічних моделях будь-який об'єкт предметної області може підпорядковуватися лише одному об'єкту вищого рівня, тобто база даних описується як деревовидний граф.

Між об'єктами існують зв'язки, кожен об'єкт може включати в себе декілька об'єктів нижчого рівня. Такі об'єкти перебувають у відношенні

предка (об'єкт більш близький до кореня) до нащадка (об'єкт більш низького рівня), при цьому можлива ситуація, коли об'єкт-предок не має нащадків або має їх кілька, тоді як у об'єкта-нащадка обов'язково є один предок.

Наприклад, якщо ієрархічна база даних містить інформацію про покупців та їх замовленнях, то існує об'єкт "покупець" (батько) й об'єкт "замовлення" (дочірній). Об'єкт "покупець" пов'язаний з одним або декількома об'єктами "замовлення". Також прикладом ієрархічної бази даних є файлова система, що складається з кореневої директорії, в якій є ієрархія піддиректорій і файлів.

У мережних моделях прийнятий вільний зв'язок між елементами різних рівнів, тобто дочірні об'єкти можуть бути пов'язані з декількома батьківськими вузлами. У таких базах даних безпосередній доступ можливий до будь-якого об'єкту незалежно від рівня, на якому він знаходиться в моделі.

В даний час найбільше розповсюдження при розробці баз даних отримали реляційні моделі даних. Реляційна модель даних є сукупністю найпростіших двовимірних таблиць, що мають назву відношення (англ. relation) і забезпечують простоту, гнучкість структури, зручність реалізації. Цей спосіб організації даних дозволяє пов'язувати дані в одній таблиці з даними в інших таблицях через унікальні ідентифікатори (ключі) або ключові поля.

Реляційна база даних є множиною взаємозв'язаних таблиць, кожна з яких містить інформацію про об'єкти певного типу.

Таблиця – регулярна структура, що складається із кінцевого набору одно типових записів.

Для успішного функціонування бази даних важлива правильна організація даних в ній. При визначенні структури даних в базі виділяють такі основні поняття:

Клас об'єктів - сукупність об'єктів, що володіють однаковим набором властивостей. Для кожного окремого об'єкту з даного класу об'єктів у таблиці створюється окремий запис.

Властивість (атрибут) - певна частина інформації про деякий об'єкт. Зберігається у вигляді стовпця (поля) таблиці.

Зв'язок (відношення) - спосіб, яким зв'язана інформація про різні об'єкти.

Елементом відношення є кортеж – рядок таблиці, упорядкований набір даних з N елементів.

Ключ – це поле (або декілька полів), що додається до таблиці і дозволяє встановити зв'язок із записами в іншій таблиці та однозначно визначає будь-який запис в таблиці реляційної бази даних.

Кожний рядок таблиці включає дані про один об'єкт (наприклад, клієнти, автомобілі, документи), а стовпці таблиці містять різні характеристики цих об'єктів — атрибути (наприклад, найменування і адреси клієнтів, марки і ціни автомобілів). **Рядки** таблиці також називаються **записами**; всі записи мають однакову структуру — вони складаються з **полів**, в яких зберігаються атрибути об'єкту. Кожне поле запису містить одну характеристику об'єкту і має строго визначений тип даних (наприклад, текстовий рядок, число, дата). Всі записи мають ті ж самі поля, тільки в них містяться різні значення атрибутів (рис. 1.1).

	Поле_1	Поле_2	...	Поле_n
	Назва_поля_1	Назва_поля_2	...	Назва_поля_n
запис_1	атрибут11	атрибут12	...	атрибут1n
запис_2	атрибут11	атрибут22	...	атрибут2n
...
запис_k	атрибутk1	атрибутk2	...	атрибутkn

Рис. 1.1 – Структура таблиці реляційної бази даних

Проектування таблиць бази здійснюється так, щоб останні не містили дублюючих даних. Для кожної таблиці варто визначити структуру запису (перелік полів, тип записуваної в них інформації), ключові поля і поля для зв'язку з іншими таблицями. Процес організації даних у таблиці називається нормалізацією. Далі наведено основні правила нормалізації:

- у кожній таблиці БД не повинно бути полів, що повторюються;
- у кожній таблиці повинен бути унікальний ідентифікатор (первинний ключ);
- кожному значенню первинного ключа повинна відповідати достатня інформація про тип сутності або про об'єкт таблиці (наприклад, інформація про успішність, про групу або студентів);
- зміна значень у полей таблиці не повинна впливати на інформацію в інших полях (крім змін в полях ключа).

Існують **ключі двох типів: первинні та вторинні** (або зовнішні).

Первинний ключ – це одне або декілька полів, комбінація значень яких однозначно визначає кожний запис в таблиці. Значення первинного ключа не може бути нульовим та завжди повинно бути унікальним. Первинний ключ використовується для зв'язування таблиці з зовнішніми ключами в інших таблицях.

Зовнішній (вторинний) ключ – це поле у таблиці, що містить посилання на поле первинного ключа в іншій таблиці. Зовнішній ключ визначає спосіб з'єднання таблиць.

З двох логічно зв'язаних таблиць одну називають таблицею первинного ключа або головною таблицею, а іншу таблицею вторинного (зовнішнього) ключа або підлеглою таблицею. СУБД дозволяють зіставити споріднені записи з обох таблиць і спільно вивести їх у формі, звіті або запиті.

Існує три типи первинних ключів: ключові поля типу Лічильник, простий ключ або складений ключ.

Поле лічильника (тип даних «Лічильник»). Тип даних поля в базі даних, при якому для кожного запису, що додається таблицю, в поле автоматично заноситься унікальне числове значення.

Простий ключ. Якщо поле містить унікальні значення, такі як штрих-коди або інвентарні номери, то таке поле можна визначити як первинний ключ. У якості ключа можна визначити будь-яке поле, що містить дані, якщо це поле не містить значення **Null** або значень, що повторюються.

Складений ключ. У випадках, коли неможливо гарантувати унікальність значень кожного поля, існує можливість створити ключ, що складається з декількох полів. Частіше за все така ситуація виникає для таблиці, що використовується для скріплення двох таблиць «багато-до-багатьох».

Зв'язок встановлюється між двома загальними полями (стовбцями) двох таблиць. Існуючі зв'язки з відношенням **«один-до-одного»**, **«один-до-багатьох»** та **«багато-до-багатьох»**.

Відносини, що можуть існувати між записами двох таблиць:

1. один-до-одного, кожному запису з однієї таблиці відповідає один запис в іншій таблиці;
2. один-до-багатьох, кожному запису з однієї таблиці відповідає декілька записів іншої таблиці;
3. багато-до-одного, множині записів з однієї таблиці відповідає один запис в іншій таблиці;

4. багато-до-багатьох, множині записів з однієї таблиці відповідає декілька записів в іншій таблиці.

Тип відношень у зв'язку, що створюється, залежить від способу визначення тих полів, що зв'язуються.

Відношення «один-до-багатьох» створюється в тому випадку, коли тільки одне з полів є полем первинного ключа або унікального індексу.

Відношення «один-до-одного» створюється в тому випадку, коли обидва поля, що зв'язуються, є ключовими або мають унікальні індекси.

Відношення «багато-до-багатьох» фактично є двома відношеннями «один-до-багатьох» з третьою таблицею, первинний ключ якої складається з полів зовнішнього ключа двох інших таблиць.

СУБД Microsoft Access 2010. Характеристика та інтерфейс програми

Microsoft Access 2010 – одна з останніх розробок продукту корпорації Microsoft, що являє собою на даний момент найбільш розповсюджену у світі систему управління реляційними базами даних. Microsoft Access є настільною СУБД реляційного типу, яка має всі необхідні засоби для виконання перерахованих вище функцій. Перевагою Microsoft Access є те, що вона має дуже простий графічний інтерфейс, який дозволяє не тільки створювати власну базу даних, але і розробляти прості і складні додатки.

У файлі бази даних Microsoft Access використовують такі типи об'єктів:

таблиці для зберігання даних;

- запити для пошуку та виявлення необхідних даних;
- форми для перегляда, додавання і зміни даних в таблицях;
- звіти для аналізу і друку даних у певному форматі.

Microsoft Access 2010 зберігає всі об'єкти бази даних в одному файлі (з розширенням .accdb), хоча і розподіляє їх по різних таблицях. Можна створити скільки завгодно таблиць, використовуючи описані нижче правила.

Розглянемо нові покращені властивості Access 2010. Одним з основних нових властивостей є вкладка **Файл** і вікно представлення Backstage, що пов'язано з нею. За допомогою панелі вікна представлення Backstage можна відкривати нові файли, отримувати доступ до файлів, що недавно використовувались, зберігати і виводити на друк документи, закривати додатки, що виконуються та інше.

Для зміни налаштування стрічки достатньо перейти до вікна представлення Backstage, відкрити вікно **Параметри Access** і обрати вкладку **Налаштування ленти**. У вікні, що з'явиться, можна створювати нові вкладки і налаштовувати вже існуючі шляхом додавання і перестановки кнопок у відповідних групах стрічки.

Завдяки темам Office візуальне оформлення бази даних в цілому та її окремих компонентів значно спростилося. За допомогою різних тем можна змінювати кольори і шрифти, а також включати різні графічні елементи.

Попередньо розроблені шаблони баз даних, що доступні у вікні представлення Backstage, дозволяють створювати бази даних, що мають потрібні властивості.

Інструмент **Частини приложення** – це колекція різних компонентів баз даних, що створені раніше, які можна повторно використовувати в роботі.

В Access 2010 значно спростилася публікація в Інтернеті, у тому числі таких компонентів, як користувальницькі звіти, форми тощо.

Завдяки навігаційним формам спрощується реорганізація та відображення різних компонентів бази даних. Для відображення окремих фрагментів бази даних достатньо скористатися стандартним методом перетягування.

В Access 2010 використовуються події таблиць (тригери), що дозволяють генерувати нові дані на основі інформації, що знаходиться у таблицях. Також вдосконалені можливості макросів, що виконують обробку даних, завдяки яким забезпечуються зручні способи ефективної автоматизації бази даних. Значно підвищилась динамічність та гнучкість звітів Access 2010 завдяки застосуванню умовного форматування та гістограм.

Як бачимо, Access 2010 оснащено великою кількістю нових можливостей і потужних інструментів, які використовуються для створення таблиць, генерування звітів і форм, а також встановлення зв'язків між таблицями.

ПРАКТИЧНА РОБОТА №2. Створення баз даних, таблиць у середовищі Microsoft Access 2010.

Мета роботи:

Створення моделі «сутність-зв'язок» та розроблення таблиць бази даних відповідної предметної області.

Технологія виконання роботи

Створення бази даних починається з проектування, що включає наступні етапи:

- дослідження предметної області;
- аналіз даних (сутностей та їх атрибутів);
- визначення відносин між сутностями і визначення первинних і вторинних (зовнішніх) ключів.

У процесі проектування визначається структура реляційної БД (склад таблиць, їх структура і логічні зв'язки). Структура таблиці визначається складом стовпців, типом даних і ключами таблиці.

До базових понять **моделі БД «сутність–зв'язок»** відносяться: сутності, зв'язки між ними і їх атрибути (властивості).

Сутність – будь-який конкретний або абстрактний об'єкт в даній предметній області. Сутності – це базові типи інформації, які зберігаються в базі даних (в реляційній базі даних кожній сутності відповідає таблиця). До сутностей можуть відноситися: студенти, клієнти, підрозділи тощо. Екземпляр сутності тип сутності - це різні поняття. Поняття тип сутності відноситься до набору однорідних осіб, предметів або подій, виступаючих як ціле (наприклад, студент або клієнт). Екземпляр сутності відноситься, наприклад, до конкретної особи в наборі. Типом сутності може бути студент, а екземпляром – Петров, Сидоров тощо.

Атрибут – ця властивість сутності в предметній області. Його найменування повинне бути унікальним для конкретного типу сутності. Наприклад, для сутності студент можуть бути використані такі атрибути: прізвище, ім'я, по-батькові, дата і місце народження, паспортні дані та ін. У реляційній базі даних атрибути зберігаються в полях таблиць.

Зв'язок – взаємозв'язок між сутностями в предметній області. Зв'язки є з'єднаннями між частинами БД (у реляційній БД – це з'єднання між записами таблиць).

Сутності – це дані, які класифікуються за типом, а зв'язки показують, як ці типи даних співвідносяться один з іншим. Якщо

описати деяку предметну галузь в термінах сутність–зв'язок, то отримаємо модель сутність-зв'язок для цієї БД.

Розглянемо предметну область «Реалізація продукції публічним акціонерним товариством «НОРД».

У БД «Реалізація продукції ПАТ «НОРД» повинні зберігатися дані про товарні групи, товари, замовників, кількість реалізованої продукції тощо. Визначимо сутності, їх атрибути і основні вимоги до функцій бази даних.

Основними наочно-значущими сутностями БД «Реалізація продукції ПАТ «НОРД» є: Товари, Групи товарів, Замовники, Реалізація продукції.

Основні наочно-значущі атрибути сутностей:

- товари: модель, характеристика, товарна група, ціна; товарні групи: найменування групи;
- замовники: найменування замовника, юридична адреса, телефон, номер рахунку.

Основні вимоги до функцій бази даних:

- визначити загальну вартість реалізованої продукції за замовниками та датами;
- вибрати товари, що були реалізовані за певний проміжок часу.

З аналізу даних предметної області виходить, що кожній сутності необхідно співставити найпростішу двовимірну таблицю (відношення). Далі необхідно встановити логічні зв'язки між таблицями. Між таблицями Довідник товарів та Довідник товарних груп необхідно встановити такий зв'язок, щоб кожному запису з таблиці Довідник товарних груп відповідало декілька записів в таблиці Довідник товарів, тобто «один-до-багатьох», оскільки кожній товарній групі може відповідати декілька товарів.

Оскільки певний товар може бути проданий будь-якому замовнику, а певний замовник може придбати будь-який товар, то між сутностями Замовник та Товар існує зв'язок – багато-до-багатьох. Але такий тип не підтримується СУБД MS Access. Тому необхідно створити додаткову таблицю Реалізація та встановити логічний зв'язок між Довідник товарів – Реалізація та Довідник замовників – Реалізація як «один-до-багатьох».

На основі вищевикладеного складаємо модель «сутність-зв'язок» для БД «Реалізація продукції ПАТ «НОРД» (рис. 2.1).



■ стрілка є умовним позначенням зв'язку «один–до–багатьох».

Рис. 2.1 – Модель «сутність–зв'язок» для БД «Реалізація продукції ПАТ НОРД»

Будь-яка СУБД дозволяє виконувати такі найпростіші операції з даними:

- *додавати* в таблицю одну або декілька записів; *видаляти* з таблиці одну або декілька записів;
- *оновлювати* значення деяких полів в одній або декількох записах; *знаходити* одну або декілька записів, що задовольняють заданій умові.

Для виконання цих операцій використовується механізм *запитів*. Результатом виконання запитів є або відібрана за певними критеріями множина записів, або зміни в таблицях. Запити до бази формуються на спеціально створеній для цього мові, яка так і називається *мова структурованих запитів* (SQL – Structured Query Language).

Основою бази даних є **таблиці**, в яких зберігаються дані про сутності предметної області. Існує декілька способів створення таблиць Microsoft Access 2010:

- створення нової порожньої таблиці;
- створення таблиці на основі списку на вузлі SharePoint;
- створення таблиці за допомогою імпорту зовнішніх даних;
- створення таблиці за допомогою конструктору.

Для створення нової порожньої таблиці потрібно вибрати на другій вкладці стрічки **Создание** пункт **Таблица**. На екрані з'явиться порожня таблиця з двома стовбцями: **Код** і **Добавить поле**. Поле **Код** вміщує в себе унікальний номер кожного рядка цієї таблиці і заповнюється автоматично по мірі додавання записів в таблицю. Для стовпця **Добавить поле** привласнюється своє ім'я.

Для створення таблиці на основі списку на вузлі SharePoint необхідно вибрати на другій вкладці стрічки **Создание** пункт **Списки SharePoint**, після чого в меню шаблонів з пунктами **Контакти**, **Задачі**, **Вопросы** тощо вибрати потрібні, вказуючи URL-адреси і відкоригувати таблиці.

Створення таблиці за допомогою імпорту зовнішніх даних дозволяє імпортувати таблицю, що розташована в іншому місці. У цьому випадку в новій таблиці поточної бази даних буде створено копію імпортованих даних.

Для більш детальної розробки таблиці з встановленням визначених властивостей полів слід використовувати створення таблиці за допомогою конструктору. В режимі Конструктора таблиць задаються імена, типи і властивості полів для таблиці.

Кожний рядок в стовбці **Тип даних** є полем зі списком, елементами якого є типи даних Access. Тип поля визначається характером даних, що до нього заносяться. У табл. 2.1 наведено типи даних Microsoft Access 2010.

Серед типів даних Access є спеціальний тип – **Счетчик**. В поле цього типу Access автоматично нумерує рядки таблиці у зростаючій послідовності. Редагувати значення такого поля неможна.

Таблиця 2.1 – Типи даних MS Access 2010

Тип даних	Опис
Текстовый	Алфавітно-цифрові дані, що потребують проведення розрахунків (до 255 символів)
Поле Мемо	Алфавітно-цифрові дані - речення, абзаци, тексти (до 64 000 символів)
Числовой	Різні числові дані (має декілька форматів: ціле, довге ціле, з плаваючою точкою)
Денежный	Грошові суми, що зберігаються з 8 знаками у десятинній частині. В цілій частині кожен три розряди розділяються комами
Счетчик	Унікальне довге ціле, що створюється Access для кожного нового запису
Логический	Логічні дані, що мають значення Истина або Ложь
Поле объекта OLE	Картинки, діаграми, фотографії, звукозапис, векторна графіка та інші об'єкти OLE з додатків Windows
Дата / время	Дата або час в одному з форматів Access
Гиперссылка	В полях цього типу зберігаються гіперпосилання, які являють собою шлях до файлу на жорсткому диску, або

	адрес в мережі Internet
Вложение	Призначено для приєднання до запису різних файлів – документів Word, електронних таблиць Excel, презентацій PowerPoint або файлів інших Типів
Вычисляемое поле	Обирається, коли необхідно заповнення результатами розрахунку формули, яка використовує одне або декілька полів у тій же таблиці
Мастер подстановок	Обирається у тому випадку, коли потрібно в одній таблиці відобразити зміст іншої таблиці

Кожне поле має індивідуальний набір властивостей, за якими можна встановити, як повинні зберігатися, відображатися і оброблятися дані. Набір властивостей поля залежить від обраного типу даних. Для визначення властивостей поля використовується бланк **Свойства поля** у нижній частині вікна конструктора таблиць (табл. 2.2).

Таблица 2.2 – Загальні властивості полів

Назва властивості	Характеристика
Размер поля	Визначається тільки для текстових і Мемо-полів, вказує максимальну кількість символів в даному полі. За замовчуванням довжина текстового поля складає 50 символів
Формат поля	Визначається для полів числового, грошового типів, полів типу Счетчик і Дата/Время. Вибирається один з форматів представлення даних
Число десятичных знаков	Визначає кількість розрядів в дробовій частині числа
Маска ввода	Визначає шаблон для введення даних
Подпись поля	Містить напис, що може бути виведена рядом з полем в формі або звіті (даний напис може містити пояснюючі відомості)
Значение по умолчанию	Містить значення, що встановлюється за замовчання даному полі таблиці
Условие на значение	Визначає множину значень, які користувач може вводити в це поле при заповненні таблиці. Ця властивість дозволяє уникнути введення недопустимих у даному полі значень
Сообщение об ошибке	Визначає повідомлення, яке з'являється на екрані у випадку введення неприпустимого значення
Обязательное поле	Установка, що вказує на то, що дане поле потребує обов'язкового заповнення для кожного запису
Пустые строки	Установка, яка визначає, чи допускається введення в дане поле порожніх рядків
Индексированное поле	Визначає прості індекси для прискорення пошуку записів

Докладніше розглянемо технологію побудови маски введення, використання якої полегшує процес введення та запобігає введенню користувачем помилкових даних. Маска вводу визначає формат введення даних у поле за допомогою спеціальних знаків та символів.

Маски введення складаються з одного обов'язкового та двох необов'язкових компонентів, що розділені крапкою з комою:

1. Перший обов'язковий компонент – це послідовність знаків та символів маски з заповнювачами та літералами, наприклад, круглі скобки, крапки та дефіси.

2. Другий необов'язковий компонент визначає спосіб збереження вбудованих знаків маски в полі. Якщо компонент приймає значення 0, знаки зберігаються разом з даними, а якщо 1, то знаки відображаються без збереження. Вибравши значення 1, можна зекономити місце для зберігання бази даних.

3. Третій необов'язковий компонент визначає знак, що буде використовуватися у якості заповнювача (за замовчуванням це знак підкреслення «_»).

Наприклад, маска для введення номеру телефона: (999) 000-00-00;0;_ Розглянемо заповнювачі та літерали, що використовуються при створенні масок введення:

- 9 – користувач може ввести цифру (від 0 до 9);
- 0 – користувач повинен ввести цифру (від 0 до 9);
- # – користувач може ввести цифру, пробіл, знаки «+» або «-». Якщо нічого не ввести, то буде вставлений пробіл;
- L – користувач повинен ввести літеру;
- A – користувач повинен ввести цифру або літеру;
- ! – маска введення заповнюється зліва направо, а не справа наліво.

При накладенні умов на значення розрізняється синтаксис для різних типів полів:

- для числових значень використовуються оператори порівняння =, <, >, <> та оператори AND, OR BETWEEN, наприклад, запис *0 or >100* означає, що значення повинно або дорівнювати нулю, або бути більше 100;
- значення типу дата та час мають бути включені у дієзи, наприклад, якщо дата повинна бути у межах 2016 року, запишемо *>=#01.01.2016# AND <#01.01.2016#*;
- текстові дані мають бути у лапках, наприклад, *-так|| or -nil*.


Також умову на значення можна встановити не тільки для поля, але для запису в цілому. Для створення подібних умов використовують посилання на поля таблиці. Наприклад, якщо потрібно, щоб строк поставки товару не перевищив 30 днів, умова на значення для таблиці буде такою:


[ДатаПоставки]<=[ДатаЗамовлення]+30

Крім вище перелічених типів даних в списку є елемент **Подстановка**, який дозволяє представити значення полів у вигляді простого або комбінованого списку. Додаткові властивості такого поля представлені на вкладці **Подстановка** вікна конструктора таблиць.

- після визначення структури таблиць БД необхідно вказати Access спосіб, яким ці таблиці зв'язані. Встановлюючи зв'язки між таблицями слід враховувати наступні обмеження:
- зв'язувати можна тільки таблиці, що розташовані в одній базі даних
- таблицю можна зв'язати із запитом, необхідно вказати, як повинні бути зв'язані таблиці.

Для визначення зв'язків необхідно відкрити таблицю у режимі конструктора, перейти у групу **Связи** і клацнути по кнопці **Схема данных**. Щоб додати таблицю в схему даних необхідно клацнути по кнопці **Отобразить таблицу**. У вікні **Добавление таблицы** потрібно виділити імена таблиць, що будуть добавлені в схему даних, і натиснути на кнопку **Добавить**.

У вікні **Схема данных** імена первинних ключів помічені . Для зв'язку таблиць необхідно виділити ключове поле таблиці, натиснути ліву кнопку миші, перетягнути її курсор на аналогічне поле у другій таблиці, після чого кнопку миші відпустити. У результаті активізується діалогове вікно **Изменение связей**.

В цьому вікні відображаються імена полів, за якими зв'язуються таблиці. Для збереження цілісності даних у зв'язаних таблицях необхідно включити прапорець  **Обеспечение целостности данных**. Після цього СУБД Access зробить неможливим запис у таблицю такого значення загального поля, якого немає у ключовій таблиці.

Після установки цілісності даних СУБД Access включає дві додаткові опції: **Каскадное обновление связанных полей** і **Каскадное удаление связанных полей**. При зміні будь-якого значення ключового поля в ключовій таблиці автоматично оновлюються значення цього поля для відповідних записів у всіх зв'язаних таблицях, а видалення одного з записів в ключовій таблиці приведе до видалення тих записів, що мають таке ж значення ключа.

В результаті встановлення зв'язку між таблицями відображається лінія виду $\infty \text{ --- } 1$.

Розглянемо процес створення таблиць на прикладі бази даних «Реалізація продукції ПАТ "НОРД"».

Завантажуємо СУБД Access: **Пуск Все программы Microsoft Office Microsoft Access 2010**. Після цього клацаємо на піктограмі **Новая база данных**, вказуємо ім'я бази та місце її зберігання і натискаємо на кнопку **Создать**.

Створювана база даних повинна відповідати структурі, зображеній на рис. 2.2.



Рис. 2.2 – Схема даних бази «Реалізація продукції ПАТ "НОРД"»

Для створення нової порожньої таблиці на стрічці потрібно обрати вкладку **Создание**, клацнути по піктограмі **Таблица** у групі **Таблиці**. Перше поле **Код** перейменовуємо на **Код товарної групи**, а другий називаємо «Товарна група», клацнувши двічі по назві стовпця. Заповнити таблицю даними (рис. 2.3):

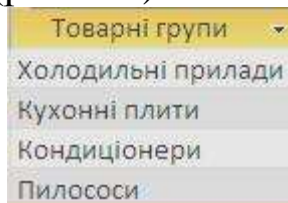


Рис. 2.3 – Дані про товарні групи

Зберегти таблицю під ім'ям «Довідник товарних груп». Поле **Код товарної групи** автоматично перетворюється на ключове. В результаті отримуємо таку таблицю (рис. 2.4):

Довідник товарних груп	
Код товарної групи ▾	Товарні групи ▾
	1 Холодильні прилади
	2 Кухонні плити
	3 Кондиціонери
	4 Пилососи
*	(№)

Рис. 2.4 – Таблиця «Довідник товарних груп» в режимі перегляду
Аналогічно створюємо таблицю «Довідник замовників» (рис. 2.5):


Довідник замовників				
Код замовник ▾	Замовник ▾	Юридична адреса ▾	Телефон ▾	Номер рахунку ▾
	1 Домотехніка	м. Донецьк	273-14-48	123656573
	2 Інтертехніка	м. Донецьк	285-96-37	253648975
	3 Донбас плюс	п. Молодіжне	44-55-66	326543657
	4 Норд-спрінт	м. Подольське	255-68-98	111236547
*	(№)			

Рис. 2.5 – Таблиця «Довідник замовників» в режимі перегляду

Наступну таблицю створимо за допомогою конструктору. На стрічці вибрати вкладку **Создание**, клацнути по піктограмі **Конструктор таблиц** у групі **Таблицы**. У вікні конструктора перерахувати поля та визначити їх властивості (рис. 2.6).

Имя поля	Тип данных
Код товару	Счетчик
Модель	Текстовый
Характеристика	Текстовый
Код групи	Числовой
Ціна	Денежный

Рис. 2.6 – Таблиця «Довідник товарів» в режимі конструктора

Для поля **Код товару** вказати, що воно є ключовим клацнувши по ньому правою кнопкою миші або натиснувши по кнопці  на вкладці **Конструктор** у групі **Сервис**.

Для поля **Характеристика** встановити у загальних властивостях розмір поля 100. Для поля **Код групи** виконати підстановку на вкладці «ПОДСТАНОВКА» (рис. 2.7):

Общие	Подстановка
Тип элемента управления	Поле со списком
Тип источника строк	Таблица или запрос
Источник строк	Довідник товарних груп
Присоединенный столбец	1
Число столбцов	2
Заглавия столбцов	Нет
Ширина столбцов	
Число строк списка	16
Ширина списка	Авто
Ограничиться списком	Нет
Разрешение нескольких значений	Нет
Разрешить изменение списка	Да
Форма изменения элементов	
Только значения источника	Нет

Рис. 2.7 – Зміст вкладки **Подстановка** в режимі конструктора для поля **Код групи**

Зберегти таблицю під ім'ям «Довідник товарів». Заповнити таблицю даними (рис. 2.8):

Довідник товарів				
Код товару	Модель	Характеристика	Код групи	Ціна
1	ДХ-241	двокамерний з верхнім розташуванням морозильної камери	1	650,25р.
2	Дх-244	двокамерний з верхнім розташуванням морозильної камери	1	550,25р.
3	ДХ-239-7	двокамерний з нижнім розташуванням морозильної камери	1	780,00р.
4	Дх-442-7	однокамерний з низькотемпературним відділенням	1	450,99р.
5	Дх-548-7	однокамерний	1	450,00р.
6	Норд-501	газова	2	654,20р.
7	ЕП-4	електрична	2	654,10р.
8	Норд-510	комбінована	2	1 025,00р.
9	KF®-20GW	роздільного типу холод-тепло	3	1 010,00р.
10	KF®-25GW	роздільного типу холод-тепло	3	1 000,00р.
11	Норд СН981	багаторівнева система фільтрації	4	638,00р.
12	Норд 982	шість ступенів фільтрації	4	1 034,00р.
*	(№)			

Рис. 2.8 – Таблиця «Довідник товарів» в режимі перегляду
Аналогічно створюємо таблицю «Реалізація» (рис. 2.9).

Реалізація	
Ім'я поля	Тип даних
Код	Счетчик
Код товару	Числовой
Код замовника	Числовой
Кількість	Числовой
Дата	Дата/время

Рис. 2.9 – Таблиця «Реалізація» в режимі перегляду

Для поля **Код товару** зробити підстановку на вкладці «ПОДСТАНОВКА» (рис. 2.10).

Общие	Подстановка
Тип элемента управления	Поле со списком
Тип источника строк	Таблица или запрос
Источник строк	Довідник товарів
Присоединенный столбец	1
Число столбцов	3
Заглавия столбцов	Нет
Ширина столбцов	1 см; 2 см; 6 см
Число строк списка	16
Ширина списка	Авто
Ограничиться списком	Нет
Разрешение нескольких значений	Нет
Разрешить изменение списка	Да
Форма изменения элементов	
Только значения источника	Нет

Рис. 2.10 – Зміст вкладки **Подстановка** в режимі конструктора для поля **Код товару**

Для поля **Код замовника** зробити підстановку на закладці «ПОДСТАНОВКА». Для поля **Кількість** визначити властивості на вкладці «Общие» (рис. 2.11):

Значение по умолчанию	1
Условие на значение	> 0

Рис. 2.11 – Властивості для поля **Кількість**

Для поля **Дата** визначити властивості на вкладці «Общие» (рис. 2.12).

Общие	Подстановка
Формат поля	Краткий формат даты
Маска ввода	00.00.00;0;0
Подпись	
Значение по умолчанию	
Условие на значение	> = #01.01.2009#
Сообщение об ошибке	Увага! Дата!

Рис. 2.12 – Властивості для поля **Дата**

Зберегти таблицю під ім'ям «Реалізація». Заповнити таблицю даними (рис. 2.13).

Для створення зв'язків між таблицями необхідно відкрити таблицю у режимі конструктора, перейти у групу **Связи** і клацнути по кнопці **Схема данных**. У вікні **Добавление таблицы** виділити всі таблиці і натиснути на кнопку **Добавить**.

Реалізація					
Код	Код товару	Код замовника	Кількість	Дата	
1	3	3	60	19.02.2009	
2	5	1	80	16.02.2016	
3	1	1	80	15.02.2016	
4	3	2	90	19.02.2016	
5	7	3	100	19.02.2016	
6	11	2	50	18.02.2016	
7	8	4	60	18.02.2016	
8	12	3	40	19.02.2016	
9	10	2	50	18.02.2016	
10	9	2	50	19.02.2016	
11	6	4	45	19.02.2016	
12	3	2	30	17.02.2016	
13	6	3	20	17.02.2016	
14	5	3	15	19.02.2016	
15	9	2	45	17.02.2016	
16	3	2	52	19.02.2016	
17	5	3	1	12.03.2016	
18	6	4	3	11.03.2016	
19	5	3	5	12.03.2016	
20	6	3	11	10.04.2016	
*	(№2)		1		

Рис. 2.13 – Таблиця «Реалізація» в режимі перегляду

Лівою кнопкою миші потягнути від первинного ключа до вторинного (див. рис. 2.2). В діалоговому вікні вказати (рис. 2. 14):

☒ Обеспечение целостности данных
☒ каскадное обновление связанных полей
☒ каскадное удаление связанных записей

Тип отношения: один-ко-многим

Рис. 2.14 – Фрагмент заповнення діалогового вікна «Схема даних»

Натиснути на кнопку **Создать**. Закрити вікно схеми даних, зберегти.

ПРАКТИЧНА РОБОТА №3. Створення запитів у середовищі Microsoft Access 2010.

Мета роботи:

Сформувати вміння та навички формування вихідної інформації за допомогою запитів до БД.

Технологія виконання роботи

При роботі з таблицями можна у будь-який момент вибрати з бази даних необхідну інформацію за допомогою запитів.

Запит – це звернення до бази даних для пошуку або зміни в базі даних інформації, що відповідає заданим критеріям.

За допомогою СУБД Access можуть бути створені такі типи запитів: запити на вибірку; підсумкові запити; запити на зміну; запити з параметром; перехресні запити.

Одним з найпоширеніших запитів є запит на вибірку, який виконує відбір даних з однієї або декількох таблиць за заданими користувачем критеріями, що не приводить до змін в самій базі даних.

Створення запитів може здійснюватися за допомогою двох режимів:

Мастер запросов і Конструктор запросов. За допомогою майстра запитів можна швидко створювати нескладні запити. Для цього необхідно виконати такі дії: у вікні бази даних вибрати вкладку стрічки **Создание**, клацнути по кнопці **Мастер запросов**, що знаходиться в групі **Запросы**. У вікні, що з'явиться (рис. 3.1), необхідно вибрати один з н типів запитів:

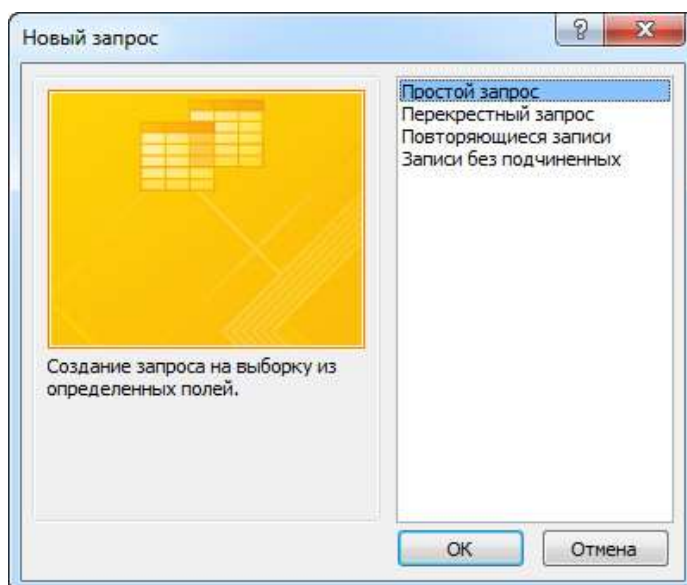


Рис. 3.1 – Вікно «Новый запрос»

- *простий запит* – дозволяє відобразити вибрану користувачем інформацію, що міститься у різних таблицях бази даних за допомогою зазначення необхідних полів;
- *перехресний запит* – клітинки перехресної таблиці містять узагальнюючі дані двох таблиць; рядками і стовпцями перехресної таблиці служать стовпці першої і другої таблиць; результуюча перехресна таблиця аналогічна зведеній таблиці Excel;
- *записи, що повторюються* – знаходить у таблицях записи, що повторюються;
- *запити без підпорядкованих* – знаходить незв’язані записи у двох таблицях, що мають загальні поля.

При виборі режиму **Конструктор запитів** через діалогове вікно **Добавление таблицы**(рис. 3.2), додаються імена таблиць, що необхідні для створення запиту, у вікно конструктора.

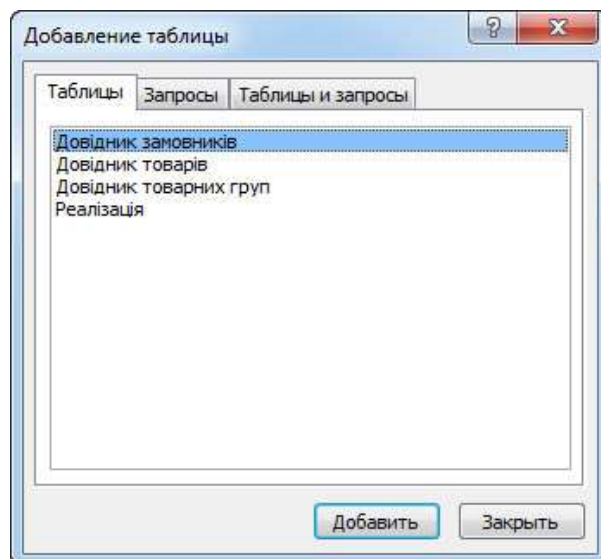


Рис. 3.2 – Діалогове вікно «Добавление таблицы»

Вікно **Добавление таблицы** складається з трьох вкладок, що містять перелік об’єктів, що пропонуються програмою для проектування запиту:

Таблицы, Запросы і Таблицы и запросы. У вікно конструктора запитів можна додати декілька об’єктів.

Вікно конструктора запитів розділено на дві частини (рис. 3.3). У верхній частині знаходяться вікна таблиць або запитів зі списками полів.

Нижня частина є бланком запиту, де вказуються параметри запиту і дані, які потрібно відібрати, а також визначається спосіб їх відображення на екрані.

Додати необхідні поля в бланк запиту можна шляхом перетягування їх імен зі списку, що знаходяться у верхній частині вікна конструктора у рядок бланка **Поле** (або подвійним клацанням по імені поля).

В рядку **Вывод на экран** прапорцем відмічаються ті поля, які повинні бути виведені на екран.

У загальному випадку поля, що виводяться в наборі записів запиту, успадковують властивості, задані для відповідних полів таблиці. Можна визначити інші значення властивостей, вибравши піктограму **Страница свойств**, що знаходиться у групі **Показать или скрыть** на вкладці **Конструктор**.

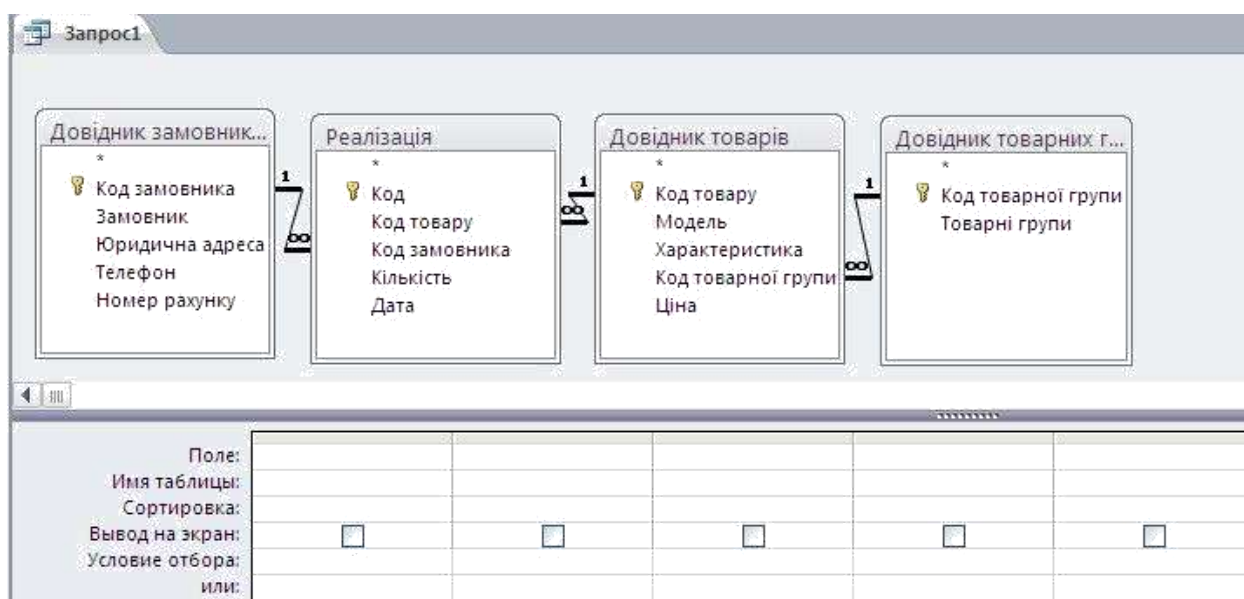


Рис. 3.3 – Вікно конструктора запитів

Видалити поле із бланка запиту можна клавішею [Delete] або за допомогою кнопки **Удалить столбцы** у групі **Настройка запроса**.

Встановлення критеріїв відбору записів. При створенні запиту можна задавати критерії у рядок **Условие отбора**, унаслідок чого за запитом буде здійснений відбір тільки необхідних записів. Оператори відбору наведено у табл. 3.1.

Таблица 3.1 – Основні оператори відбору

Оператор	Опис використання
And	Перевірка значення поля на відповідність декільком критеріям. Наприклад, >2 and<5
Or	Перевірка значення поля на відповідність хоча б одному критерію. Наприклад: 10 or 15
<>	Виключає групу даних зі складу аналізованих запитом записів. Наприклад: <>10

Between	Дозволяє задати діапазон значень. Наприклад: between 10 and 20
In	Дозволяє задавати використовуваний для порівняння списків значень. Наприклад: in (первый, второй, третий)
Like	Використовується для пошуку зразків в текстових полях з використання шаблонів: * – означає будь-яку кількість символів; ? – будь-який символ; # – вказує, що в даній позиції повинна бути цифра. Наприклад: для вибору прізвища, що починається з літери П і з закінченням – ов можна записати: like «П*ов»

При роботі з датою та часом значення повинні бути заключені між символами #. Наприклад:

#10 мая 1998# > #31.12.2016#

В СУБД Access використовується ряд інших функцій, які допоможуть задати умови відбору для дати та часу:


Day(дата) – повертає значення дня в діапазоні від 1 до 31.

Month(дата) – повертає значення місяця в діапазоні від 1 до 12.

Year(дата) – повертає значення року в діапазоні від 100 до 9999

Date() – повертає значення поточної дати.

Обчислювані поля. Над будь-якими полями можна задати обчислення та зробити обчислюване значення новим полем у запиті.

Для цього в рядку **Поле** бланку запита через **Построитель**  вводиться формула для обчислення. Ім'я полів беруться у квадратні дужки.

Наприклад:

Вартість замовлення: [Реалізація]![Кількість]
[Довідник товарів]![Ціна]*

У виразах можна використовувати такі оператори:

- арифметичні: * множення; + додавання; - віднімання; / ділення; ^ зведення в ступінь;
- з'єднання частин тексту за допомогою знаку &, наприклад: = [Прізвище] & " "&[Ім'я]

Підсумкові запити значно відрізняються від звичайних. В них поля поділяються на 2 типи:

- поля, за якими здійснюється угруповання даних;
- поля, для яких проводяться обчислення.

Для створення підсумкового запита, знаходячись у режимі конструктора, необхідно натиснути кнопку **Итоги**. В результаті чого в бланку запиту з'явиться рядок **Групповая операция**.

Групування в підсумковому запиті здійснюються тільки за одним полем. У решті необхідно визначити підсумкові функції.

Основні групові функції, які можна застосувати:

- Группировка – угруповання результатів запиту за значенням стовпця;
- Sum – загальна сума всіх значень заданого поля;
- Avg – визначення середнього значення поля;
- Min – визначення мінімального значення поля;
- Max – визначення максимального значення поля;
- Count – підрахунок кількості записів, що задовольняють критерію запиту за визначеним полем;
- StDev – стандартне відхилення значень стовпців від середнього у групі;
- Var – дисперсія значень стовпця;
- First – перше значення серед відібраних записів;
- Last – останнє значення серед відібраних записів;
- Выражение – обчислюване поле;
- Условие – клітинка, яка використовується для вибору записів, але не для узагальнення даних.

Перехресні запити. За допомогою перехресного запиту можна більш наглядно представити дані підсумкових запитів, що передбачають групування за декількома признаками.

В цьому випадку значення полів за першою ознакою групування можуть стати заголовками рядків, а за другою – заголовками стовбців.

Для створення перехресного запиту необхідно в режимі конструктора виконати команду **Запрос → Перекрестный**. В рядку **Перекрестная таблица** необхідно вказати як буде використано дане поле: у вигляді заголовка рядків, заголовка стовбців або значень.

Параметричний запит дозволяє користувачу задати критерій відбору через введення відповідного параметра при виклику запиту. При створенні запита в конструкторі для ключового поля в рядку **Условие отбора** необхідно в квадратних дужках ввести **[Введіть <назва_поля>]** (повідомлення, яке буде виводиться на екран при виконанні запита).

Виконання запита. Готовий запит виконується після клацання по кнопці



Выполнить, яка знаходиться у групі **Результаты** на вкладці **Конструктор**. У результаті буде отримана таблиця з відповіддю на задані умови.

Створення запитів на зміну (запити-дії).

Для швидкої зміни, вставки, створення або видалення наборів даних із бази даних використовуються запити на зміну. В режимі конструктору запитів необхідно вказати тип дії над відібраними записами. Існує всього чотири типи запитів-дій: створення таблиці, оновлення, видалення, додавання.

Запит на створення таблиці – дозволяє зберегти набір даних, що були відібрані у відповідь на запит, у вигляді таблиці.

Оновлення – використовується для того, щоб оновити значення деяких полів для всіх записів, що були відібрані в результаті виконання запиту.

Видалення – використовується для видалення всіх записів, що були відібрані в результаті виконання запиту.

Додавання – використовується для додавання даних у таблицю. У MS Access запити-дії відмічаються спеціальними значками:



Особливістю роботи з запитами на зміну є те, що з метою уникнення будь-яких випадковостей, необхідно перед виконанням запиту-дії створити резервну копію таблиці. Перш ніж клацнути по кнопці **Выполнить**, варто проглянути результат в режимі таблиці,



скориставшись кнопкою **Режим**.

Для бази даних «Реалізація продукції ПАТ "НОРД"» розглянемо процес створення запитів.

Приклад 1. Побудуємо простий запит на вибірку «*Замовлення*», за допомогою якого буде відібрана інформація про всі замовлення продукції ПАТ «НОРД» клієнтами за полями: «*Модель*», «*Характеристика*», «*Товарна група*», «*Замовник*», «*Дата*».

На вкладці **Создание** вибираємо піктограму **Конструктор запросов**. Додаємо у вікно конструктора всі 4 таблиці.

Із таблиць двійним клацанням по полю вибираємо потрібні поля: **Модель**,

Характеристика, Товарні групи, Замовник, Дата (рис. 3.4).

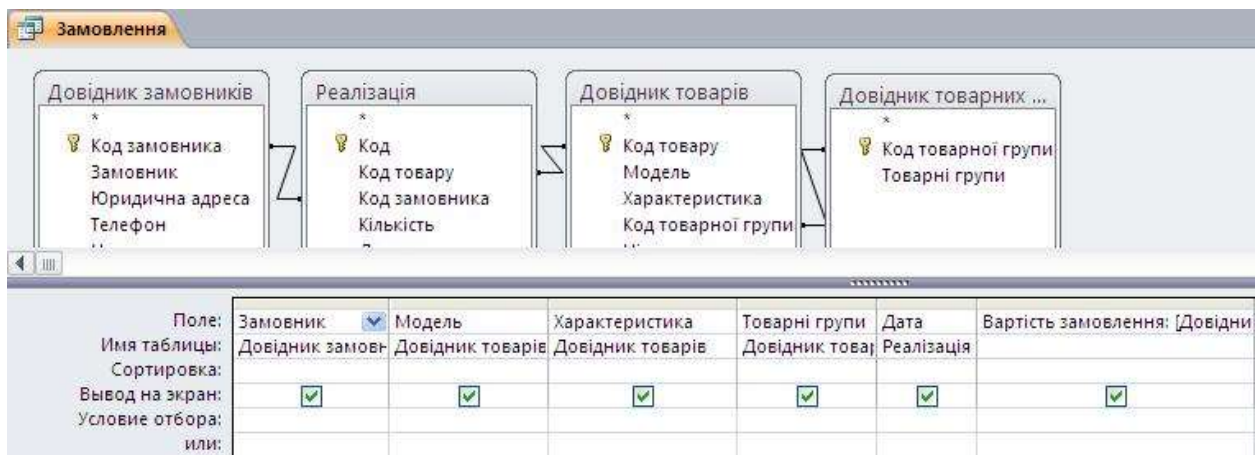


Рис. 3.4 – Запит «Замовлення» в режимі конструктора

Створюємо поле **Вартість замовлення**. Для нового порожнього поля викликаємо **Побудувач** і будуємо такий вираз:

Вартість замовлення: [Реалізація]![Кількість][Довідник товарів]![Ціна]*

Зберігаємо запит під ім'ям «**Замовлення**». Більш складними запитами є запити на вибірку з умовами.

Приклад 2. Відберемо замовлення тільки від 19 лютого 2016 р. Запит строюємо на основі таблиць. Із таблиць подвійним клацанням по полю обираємо необхідні поля: **Модель, Характеристика, Товарні групи, Замовник, Дата** (рис. 3.5).

Поле Вартість_замовника розраховуємо через Побудувач:

*Вартість_замовника:[Реалізація]![Кількість] * [Довідник товарів]![Ціна]*

На поле Дата встановлюємо **Умову вибірки**: #19.02.2016# Поле Дата не виводимо на екран – знімаємо прапорець **Вивод на екран** для цього поля.

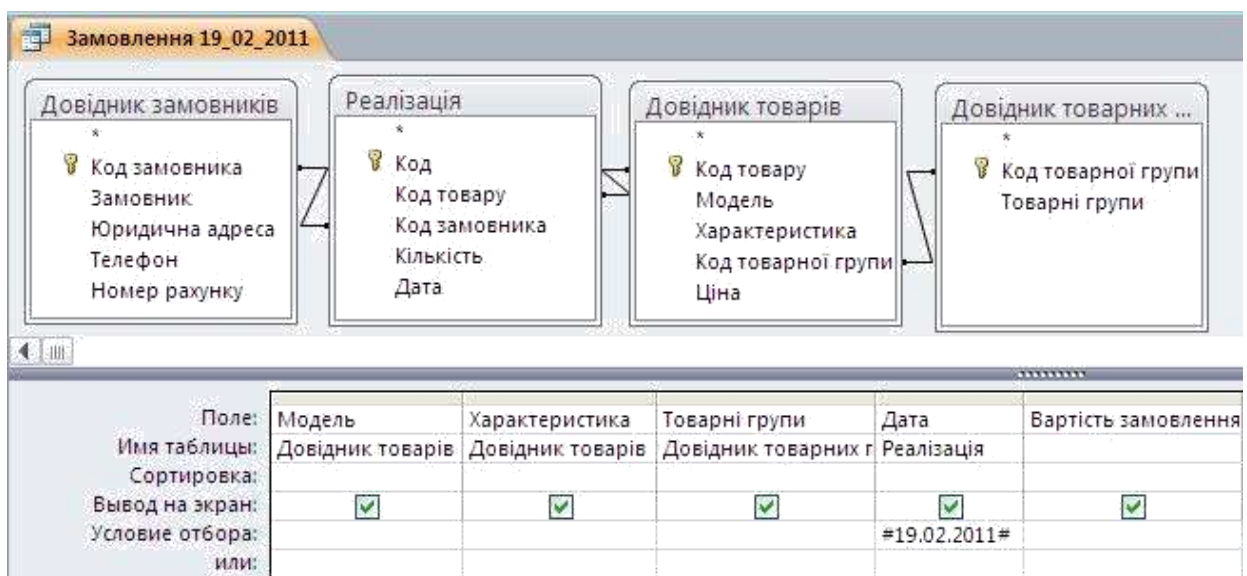


Рис. 3.5 – Запит «Замовлення 19.02.2016» в режимі конструктора

Зберігаємо запит під ім'ям «**Замовлення 19_02_2016**».

Приклад 3. За допомогою запиту на вибірку можна відібрати тільки замовлення на двокамерні холодильники. Запит слід створювати на основі запиту «Замовлення», вибрати всі поля запиту.

Для поля **Характеристика** ставимо Условие отбора:

*Like —*двокамер**

Поле **Характеристика** не виводимо на екран (рис. 3.6)

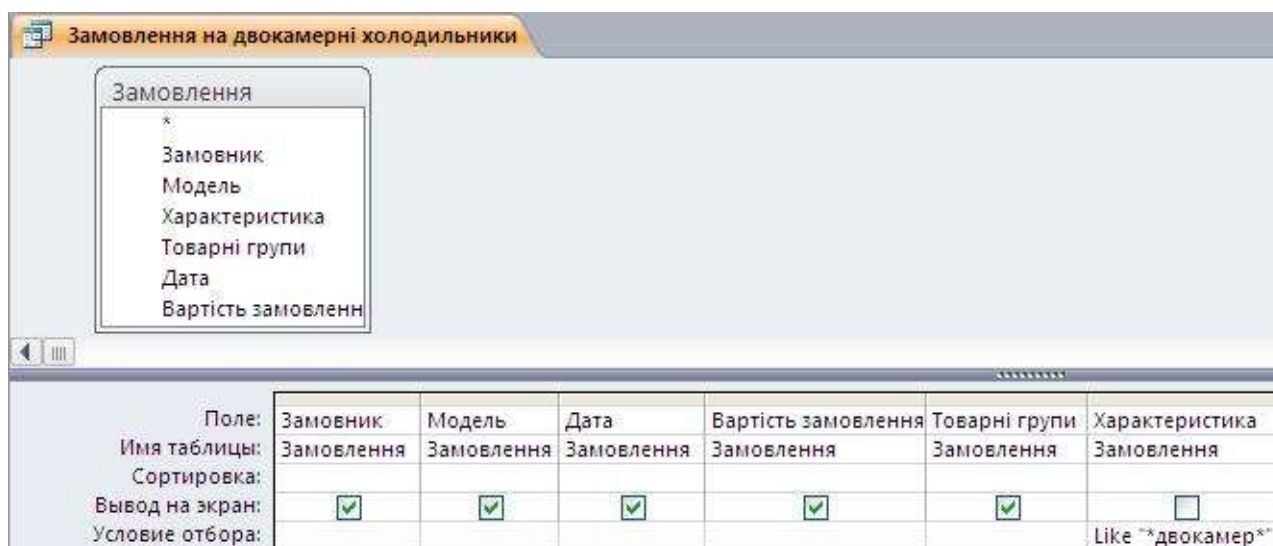


Рис. 3.6 – Запит «Замовлення на двокамерні холодильники» в режимі конструктора

Зберігаємо запит під ім'ям «**Замовлення на двокамерні холодильники**». **Приклад 4.** Запит на вибірку «**Замовлення поточного місяця**» створюємо на основі запиту «Замовлення». Вибираємо всі поля запиту.

Розраховуємо поле Місяць через **Построитель**

Місяць: Month([Дата])

Розраховуємо поле Рік через **Построитель** : *Рік: Year([Дата])*

Для поля Місяць ставимо **Условие отбора**: *Month(Date())*

Для поля Рік ставимо **Условие отбора**: *Year(Date())*

Поля **Рік** і **Місяць** не виводимо на екран (рис. 3.7).

Зберігаємо запит під ім'ям «**Замовлення поточного місяця**».

Приклад 5. Підсумковий запит «**Загальна вартість замовлень за товарними групами**» будуємо на основі таблиць. Із таблиць подвійним клацанням по полю вибираємо потрібні поля: **Товарна група, Сума**.

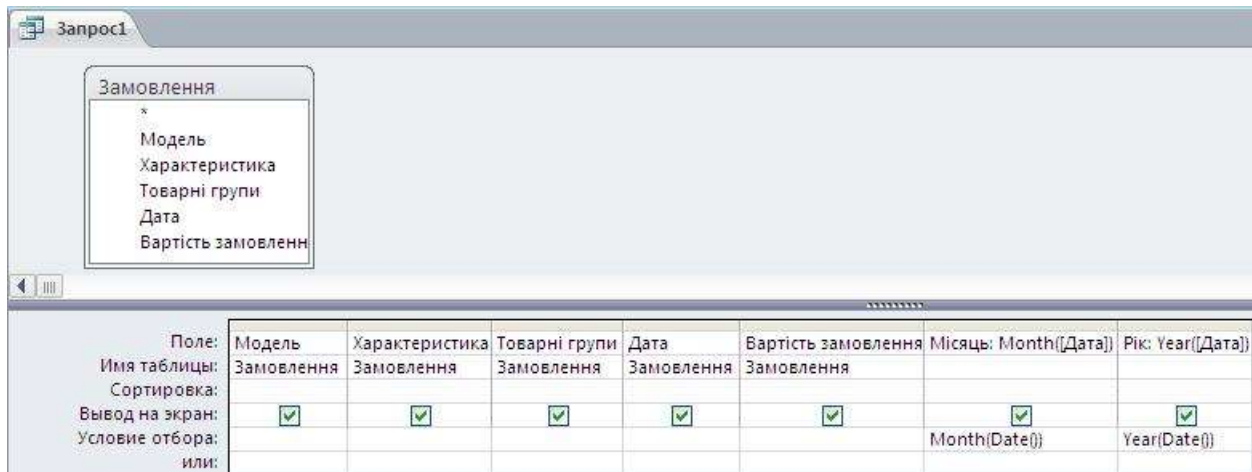


Рис. 3.7 – Запит «Замовлення поточного місяця» в режимі конструктора

Поле **Сума** розраховуємо через **Побудоваль**:

*Сума: [Реалізація]![Кількість] * [Довідник товарів]![Ціна]*

Встановлюємо угруповання за допомогою піктограми Σ . Для поля Сума вибираємо **Групповую операцію** – **Sum** (рис. 3.8).

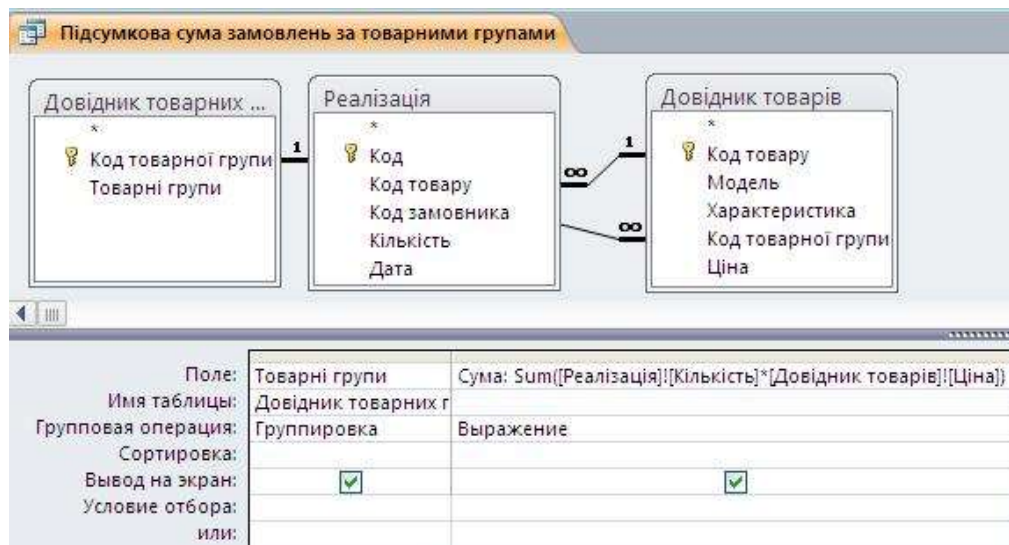


Рис. 3.8 – Запит «Підсумкова сума замовлень за товарними групами» в режимі конструктора

Зберігаємо запит під ім'ям «**Загальна вартість замовлень за товарними групами**». Отримуємо (рис. 3.9):

Підсумкова сума замовлень за товарними групами	
Товарні групи	Сума
Кондиціонери	145 950,00р.
Кухонні плити	178 591,80р.
Пилососи	73 260,00р.
Холодильні прилади	278 430,00р.

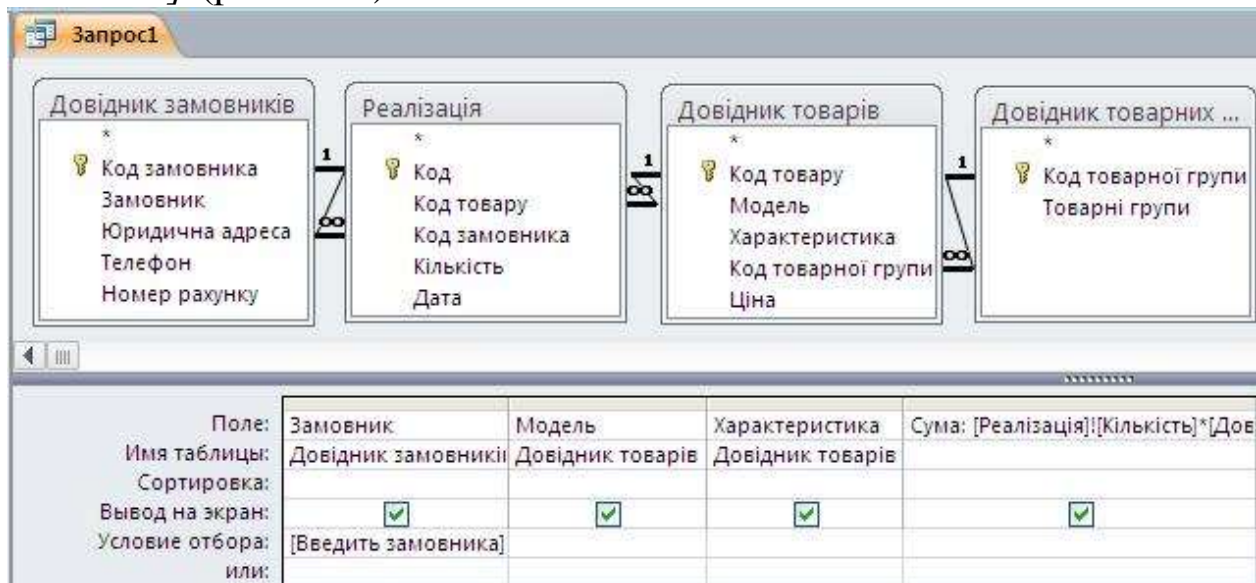
Рис. 3.9 – Запит «Підсумкова сума замовлень за товарними групами» в режимі перегляду

Приклад 6. Параметричний запит «Інформація про замовлення конкретного клієнта» створюємо на основі таблиць, із яких подвійним клацанням по полю вибираємо потрібні поля: **Замовник, Модель, Характеристика.**

Поле **Сума** розраховуємо через Построитель :

*Сума: [Реалізація]![Кількість] * [Довідник товарів]![Ціна]*

Для поля **Замовник** визначаємо Условие отбора: *[Введіть замовника]* (рис. 3.10):



Поле:	Замовник	Модель	Характеристика	Сума: [Реалізація]![Кількість]*[Дов
Имя таблицы:	Довідник замовників	Довідник товарів	Довідник товарів	
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	[Введіть замовника]			
или:				

Рис. 3.10 – Запит «Інформація про замовлення конкретного клієнта» в режимі конструктора

Зберігаємо запит під ім'ям «Інформація про замовлення конкретного клієнта».

Приклад 7. Запит на оновлення «Збільшення ціни на товари групи **Пилососи** на **10%**».

Нагадаємо, що запити такого типу змінюють дані початкової таблиці. Тому перед тим, як створювати запит, слід зробити копію таблиці «Довідник товарів» Додаємо таблиці «Копія Довідник товарів» і «Довідник товарних груп» у конструктор запитів і обираємо тип запиту **Обновление**.

Вибираємо поля **Ціна** і **Товарні групи**. Для поля **Товарні групи** встановлюємо умову відбору «Пилососи». Для поля **Ціна** встановлюємо оновлення: *[Ціна]*1,1* (рис. 3.11).

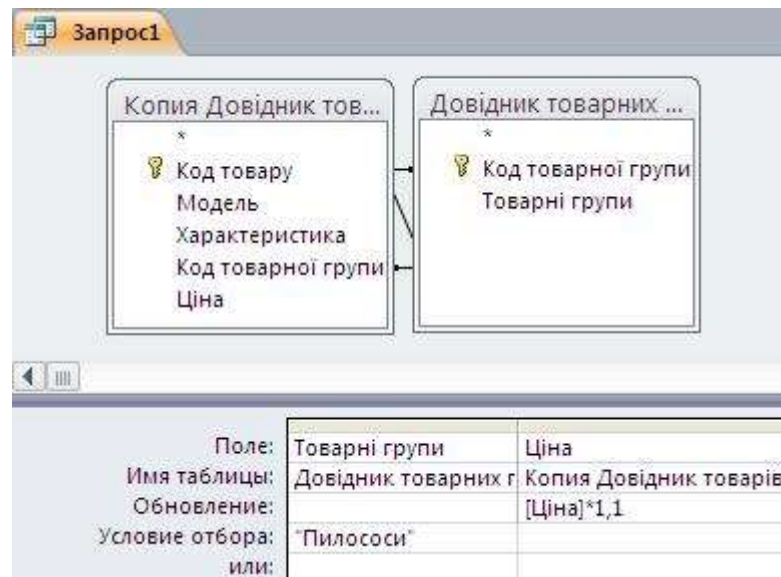


Рис. 3.11 – Запит «Збільшення ціни на товари групи Пилососи на 10%» в режимі конструктора

Зберігаємо запит. Він створюється із знаком . Запускаємо запит на виконання. Запит приведе до перерахунку цін у таблиці «Копія Довідник товарів».

За аналогічною схемою створюються запити на додавання і видалення. Результати відбору даних за допомогою запиту можна зберегти у вигляді таблиці. Побудуємо запит на створення таблиці на основі запиту на вибірку «Замовлення». Виберемо всі поля запиту.

Обираємо тип запиту за допомогою піктограми Создание таблицы. У діалогове вікно ввести ім'я таблиці «**Всі замовлення**» (рис. 3.12). Зберігаємо запит, запускаємо на виконання.

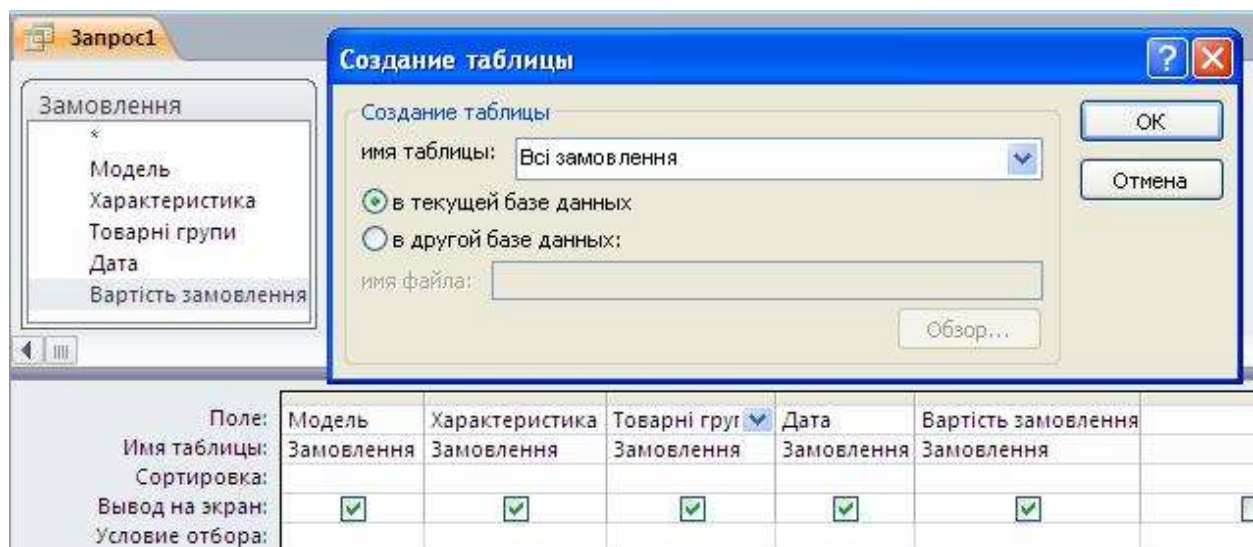


Рис. 3.12 – Запит на створення таблиці «всі замовлення» у режимі конструктора

Запит приведе до створення нової таблиці з ім'ям «**Всі замовлення**».

Приклад 8. Перехресний запит, що містить дані про вартість реалізації продукції за товарними групами та датами, створюємо за допомогою **Мастера запросов**. У вікні **Новый запрос** вибираємо **Перекрестный запрос**. На наступному етапі вибираємо запит «Замовлення», на основі якого будуватимемо перехресний запит. У наступному вікні вибираємо поля: **Товарні групи**, **Дата**, **Вартість замовлення**. В якості заголовків рядків вибираємо поле **Товарні групи**, для заголовків стовпців вибираємо поле **Дата**, а для розрахунку сумарної вартості замовлення обираємо функцію **Сумма**.

Зберігаємо запит під ім'ям «**Замовлення_перехресний**», запускаємо на виконання і отримуємо табличний вигляд (рис. 3.13).

Замовлення_перекрестный											
Товарні групи	Итоговое	10_04_2016	11_03_2016	12_03_2016	15_02_2016	16_02_2016	17_02_2016	18_02_2016	19_02_2009	19_02_2016	
Кондиціонери	145 950,00р.						45 450,00р.	50 000,00р.		50 500,00р.	
Кухонні плити	178 591,80р.	7 196,20р.	1 962,60р.				13 084,00р.	61 500,00р.		94 849,00р.	
Пилососи	73 260,00р.							31 900,00р.		41 360,00р.	
Холодильні прилади	278 430,00р.			2 700,00р.	52 020,00р.	36 000,00р.	23 400,00р.		46 800,00р.	117 510,00р.	

Рис. 3.13 – Перехресний запит у режимі перегляду

ПРАКТИЧНА РОБОТА №4. Робота з формами у середовищі Microsoft Access 2010.

Мета роботи:

Сформувати вміння та навички створення оболонок для баз даних.

Теоретичні відомості.

Форми є основним засобом організації інтерфейсу користувача у додатках MS Access. Добре розроблені форми дозволяють працювати з додатком навіть непідготовленому користувачу.

Найчастіше форми створюються з метою:

- введення і редагування даних у зручному для користувача вигляді;
- управління ходом виконання додатку – у цьому випадку форми використовуються для запуску макросів;
- виведення повідомлень – за допомогою форм можна вивести на екран інформацію, попередження або повідомлення про помилки;
- друк інформації.

Для створення форми слід у вікні бази даних перейти на вкладку **Создать** і звернутися до групи **Форми**.

У залежності від задачі, що вирішується, форму в Microsoft Access 2010 можна створити одним з наступних способів:

- створення форми за допомогою інструмента **Мастер форм**;
- створення форми за допомогою інструмента **Конструктор форм**;
- створення форми за допомогою інструмента **Форма**;
- створення форми за допомогою інструмента **Пустая форма**;
- створення Web-форми, у якій відображаються декілька записів, за допомогою інструмента **Несколько инструментов**;
- створення розділеної форми за допомогою інструмента **Разделенная форма**;
- створення розділеної форми за допомогою інструмента **Модальное диалоговое окно**;
- створення форми у вигляді зведеної діаграми за допомогою інструмента **Сводная диаграмма**;
- створення форми у вигляді зведеної таблиці за допомогою інструмента **Сводная таблица**.

Будь-яка форма може включати такі розділи:

- розділ **Заголовок форми** визначає верхню частину форми. Додається у форму разом з розділом примітки форми. В область заголовку можна помістити текст, графіку та інші елементи управління;

- розділ **Верхний колонтитул** визначає верхній колонтитул сторінки при друкуванні форми. Цей розділ додається в форму разом з розділом, що визначає нижній колонтитул сторінки, і відображається тільки коли форма відкрита у режимі попереднього перегляду;
- розділ **Область данных** визначає основну частину форми, що містить дані, отримані з джерела. Може містити елементи управління, що відображають дані з таблиць та запитів, а також незмінні дані, наприклад, написи;
- розділ **Нижний колонтитул** визначає нижній колонтитул сторінки при друкуванні форми. Звичайно містить номер сторінки, дату тощо;
- розділ **Примечание формы** визначає нижню частину форми. Відображається внизу останньої екранної сторінки форми.

Технологія виконання роботи

Розглянемо процес створення форми для редагування даних таблиці «Довідник товарів» за допомогою інструмента **Мастер форм**, користуючись такою схемою:

1-й крок. Вказуємо джерело (таблиця «Довідник товарів»), вибираємо поля, які будуть відображатися на формі (Модель, Характеристика, Код товарної групи, Ціна) (рис. 4.1).

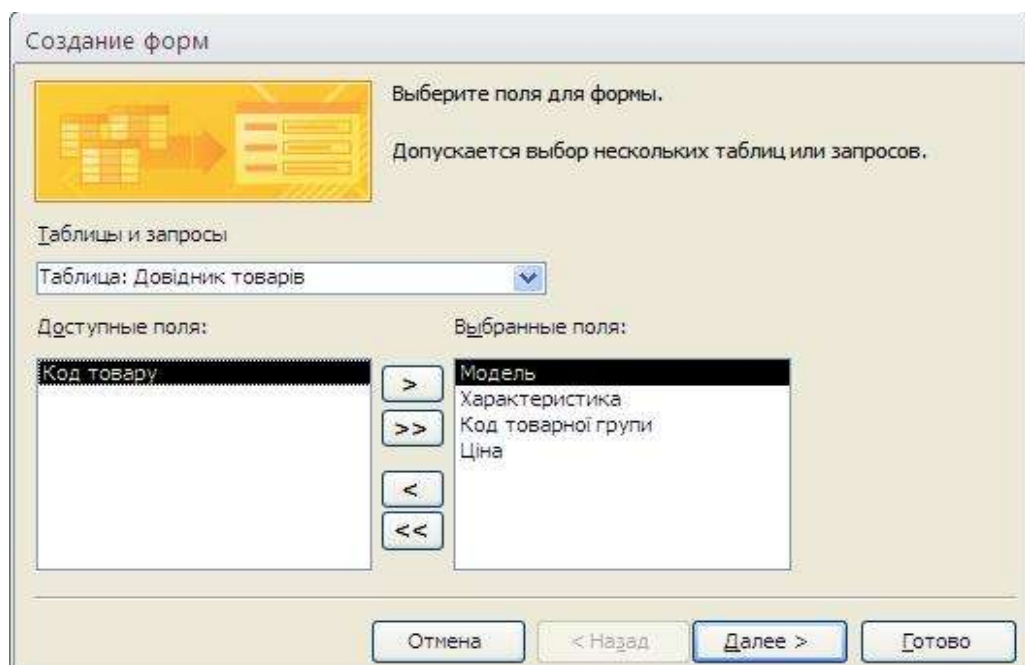




Рис. 4.1 – Діалогове вікно майстра форм

2-й крок. Обираємо зовнішній вид форми: в один столбец, ленточный, табличный, выровненный.

3-й крок. Визначаємо ім'я форми «Довідник товарів» Готово.
У результаті отримуємо форму, що зображена на рис. 4.2.

Рис 4.2 – Форма «Довідник товарів» у режимі перегляду

Для здійснення переходу на наступний, перший або останній запис слід використовувати кнопки , для додавання запису слід натиснути кнопку .

Для зміни зовнішнього вигляду форми необхідно перейти в **Режим макета**, натиснувши правою кнопкою миші, і використати інструменти, що входять до складу групи **Темы**, а також інструменти зі вкладки **Формат**.

Наступну форму для редагування даних таблиці «Довідник замовників» створюємо за допомогою інструмента **Конструктор**, користуючись такою схемою:

1-й крок. На стрічці вибираємо вкладку **Создание**, натискаємо кнопку **Конструктор форм**, у групі **Сервіс** натискаємо кнопку **Добавить поля**. Із списку полів переносимо на форму поля «Замовник», «Юридична адреса», «Телефон», «Номер рахунку». Змінюємо тип та розмір шрифту, колір тексту і заливки (за допомогою вкладки **Формат**) (рис. 4.3).

Рис. 4.3 – Розділ «Область данных» конструктора форм

2-й крок. Додаємо області заголовку та примітки форми за допомогою кнопки **Заголовок** групи **Колонтитулы** і створюємо назву форми «ЗАМОВНИКИ» за допомогою елемента управління **Аа** (Надпись). Елементом управління називають будь-який об'єкт, розташований на формі: напис, поле, список, прапорець тощо. У примітці форми створюємо поле для виводу поточної дати за допомогою елемента **abl** (Поле) і напису з ПІБ відповідального **Аа** (Надпись) (рис 4.4).

Заголовок формы									
				ЗАМОВНИКИ					
Область данных									
	Замовник				Замовник				
	Юридична адреса				Юридична адреса				
	Телефон				Телефон				
	Номер рахунку				Номер рахунку				
Примечание формы									
	Дата:	=Date()						Іванов В. О.	

Рис. 4.4 - Заповнення розділу **Заголовок/Примечание** в конструкторі форми

3-й крок. Область даних оформлюємо рисунком за допомогою елемента **Вставить изображение**, де визначаємо файл з рисунком.

4-й крок. Відкриваємо властивості форми (правою кнопкою миші по формі). Визначаємо значення властивостей:

Полосы прокрутки – Отсутствуют

Область выделения – Нет

Разделительные линии – Нет

Проглядаємо форму в режимі форми.

Для аналізу даних Microsoft Access 2010 пропонує інструмент – діаграма. Для створення діаграми аналізу загальних сум замовлень підприємствами за датами обираємо об'єкт бази даних, на основі якого необхідно побудувати діаграму. В нашому випадку – це запит «Замовлення». На головній стрічці натискаємо на кнопку **Создание**, вибираємо об'єкт **Другие формы**.

Вибираємо поля, які необхідно відобразити на діаграмі і переносимо їх на макет таким чином: в якості полів даних вибираємо поле **Вартість** замовлення, в якості полів рядів і категорій – поля **Товарні категорія** і **Дата** відповідно. Тип діаграми можна вибрати за допомогою контекстного меню.

З метою навігації базою даних створюється головна форма. Вона використовується як головне меню бази даних. Елементами головної кнопкової форми є об'єкти форм і звітів. Для однієї бази даних можна створити декілька форм. Кнопки слід групувати на сторінках кнопкової форми так, щоб користувачу було зрозуміло, які дії можна виконувати на кнопкових формах (перегляд, введення або редагування даних). На підлеглих формах слід розмістити кнопки для повернення в головну форму.

Отже, створимо головну форму, з якою безпосередньо працюватиме користувач бази даних. Створення форми проводимо в режимі конструктора. Додаємо область заголовка форми, де за допомогою елемента управління «Надпись» створюємо заголовок форми.

У властивостях форми вказуємо оформлення за допомогою **Рисунка** та заповнюємо вкладку **Макет** за зразком, що зображено на рис. 4.5.

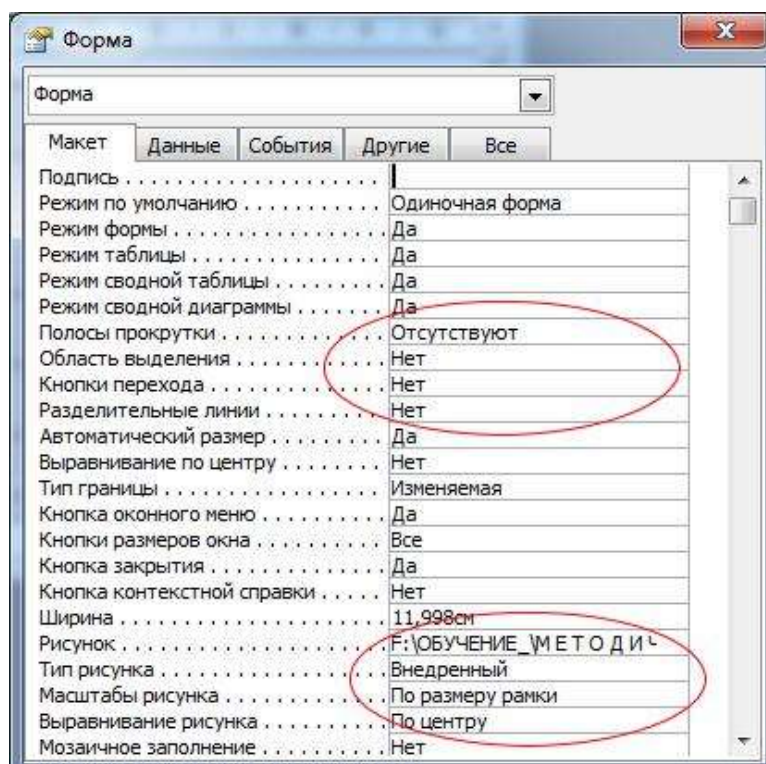


Рис. 4.5 – Зразок заповнення властивостей форми

За допомогою елементів управління, що знаходяться на вкладці **Конструктор**, створюємо на формі кнопку для перегляду змісту форми **Довідник товарів**.

У майстрі створення кнопок вказуємо дію, що буде виконуватися при натисненні на кнопку **Работа с формой – Открыть форму** (рис. 4.6).

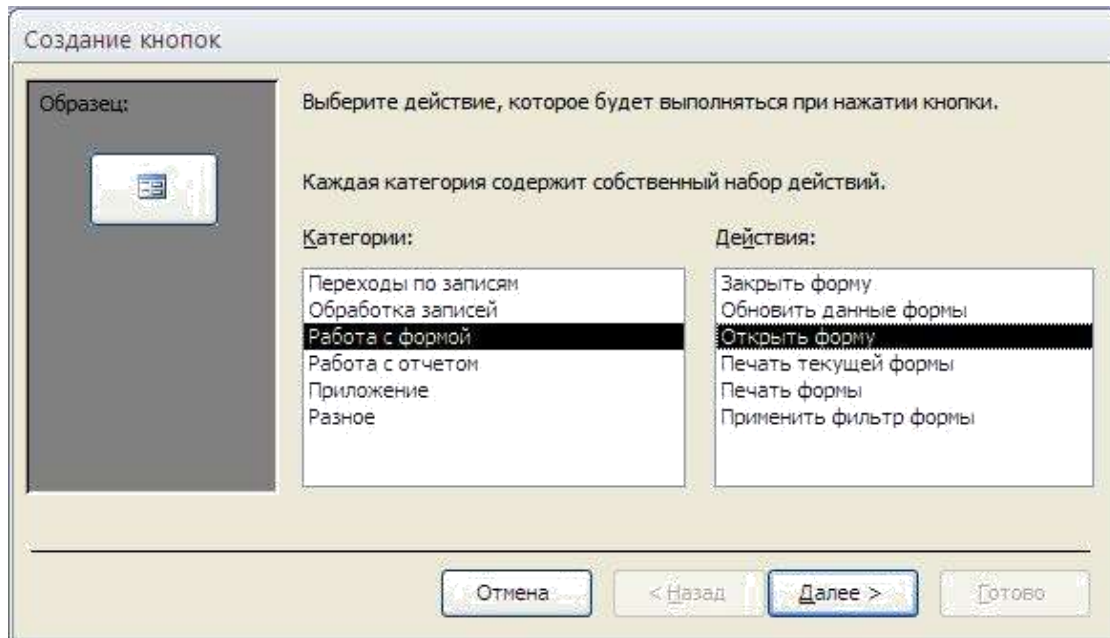


Рис. 4.5 – Діалогове вікно для створення кнопки переходу

Вказуємо ім'я форми, що повинна відкритися. Встановлюємо перемикач у положення – **Открыть форму и показать все записи**.

Вводимо підпис кнопки або визначаємо рисунок, який буде на ній відображатися.

Аналогічним чином створюємо кнопку для перегляду змісту форми «Довідник замовників».

Створюємо кнопку для виходу з додатку. У майстрі створення кнопок укажуємо дію, що буде виконуватися при натисненні на кнопку (**Приложение – Выйти из приложения**) (рис. 4.7).

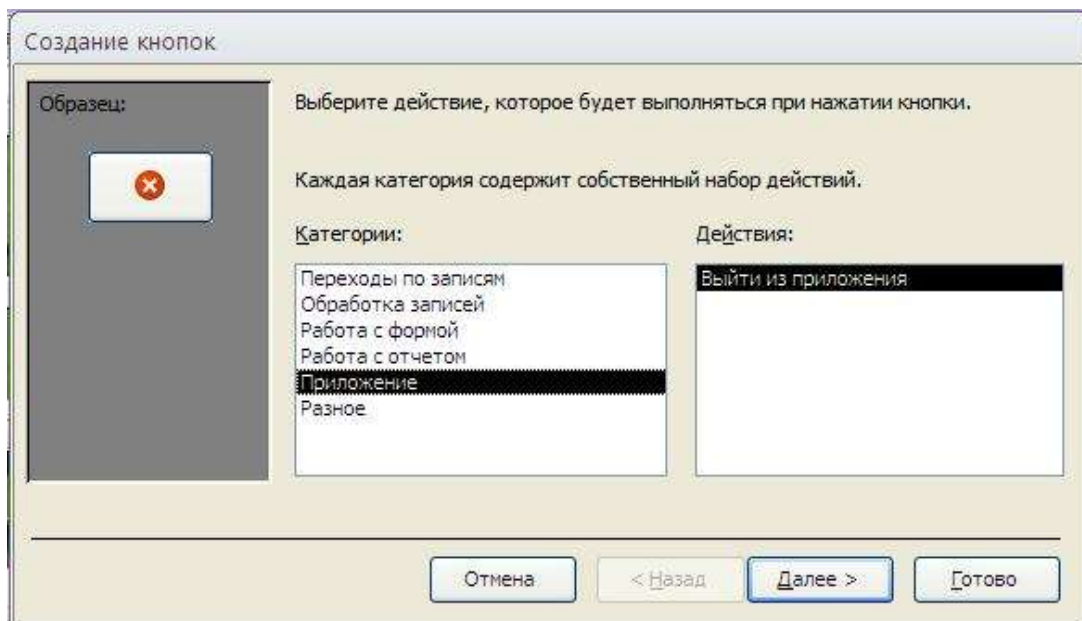


Рис. 4.7 – Діалогове вікно для створення кнопки виходу

Вибираємо рисунок. Готово. Зберігаємо форму під ім'ям «Головна». Для того, щоб форма запускалася автоматично при відкритті бази даних необхідно вибрати вкладку **Файл**, в лівому куту натиснути кнопку **Параметри**. У вікні, що з'явиться, обрати пункт **Текущая база** (рис. 4.8).

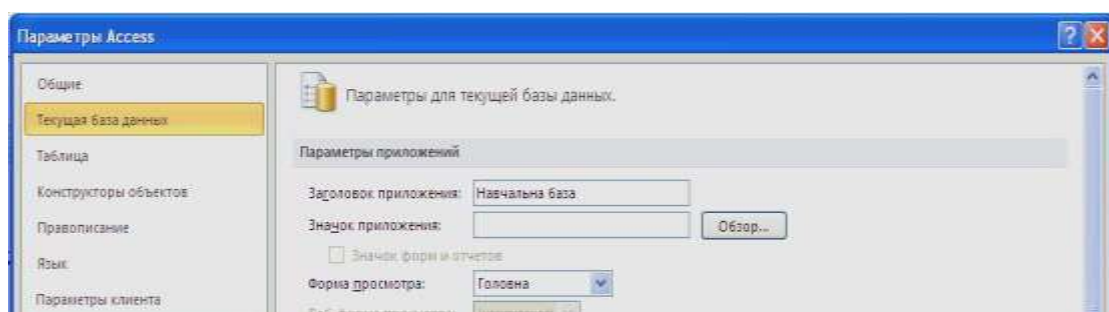


Рис. 4.8 – Діалогове вікно «Параметры Access»

Поле зі списком **Форма просмотра** дозволяє вибрати форму, яка буде виводитися на екран при відкритті бази даних, у нашому випадку – це форма «Головна». Необхідно задати заголовок і значок додатка.

ПРАКТИЧНА РОБОТА №5. Створення звітів у середовищі Microsoft Access 2010.

Мета роботи:

Сформувати вміння та навички організації виведення інформації за допомогою звітів.

Теоретичні відомості.

Звіт – це відформатоване представлення інформації із бази даних у вигляді друкованого документу.

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

Звіти надають дві принципові переваги:

- звіти надають широкі можливості для угруповання і обчислення проміжних і загальних підсумків для великих наборів даних;
- звіти можуть бути використані для отримання красиво оформлених матеріалів, наклеює і інших документів.

Таким чином, запити дозволяють вибрати з бази потрібні відомості, а звіти

– представити ці відомості у вигляді, зручному для сприйняття, а також надають широкі можливості для узагальнення та аналізу даних.

В Access 2010 існують такі режими створення звітів:

Отчет – найшвидший спосіб створення звіту, формується відразу без уточнення додаткової інформації.

Мастер отчетов – засіб Access 2010, що створює звіт крок за кроком на основі відповідей, отриманих на задані користувачу запитання. Розробник має можливість вказати спосіб угруповання і сортування даних, включити в звіт поля з декількох таблиць або запитів при умові, задані відношення між цими таблицями і запитами.

Пустой отчет – виводить на екран порожній макет майбутнього звіту і дозволяє користувачу самостійно визначити структуру звіту.

Конструктор – призначений для конструювання звітів з урахуванням побажань користувача щодо оформлення, структурних елементів, а також додаткових елементів управління, що можуть знаходитись у звіті.

Звіт в режимі конструктору має таку структуру:

Заголовок. Цей розділ друкується тільки у верхній частині першої сторінки звіту. Використовується для виводу даних, таких як текст

заголовка звіту, дата або констатуюча частина тексту документа, які слід надрукувати один раз на початку звіту.

Верхний\нижний колонтитул. Використовується для виводу даних, таких як заголовки стовбців, дати або номери сторінок, що друкуються зверху\знизу на кожній сторінці звіту.

Область данных – розташована між верхнім та нижнім колонтитулами сторінки. Містить основний текст звіту, що формується із полів, перерахованих у **Списку полів**.

Примечание. Використовується для виведення даних, таких як текст висновку, загальні підсумкові значення або підпис, які слід надрукувати один раз в кінці звіту. Примітки друкуються над нижнім колонтитулом сторінки на останній сторінці звіту.

Для додавання або видалення кожного елементу структури необхідно вибрати відповідну команду в контекстному меню або на головній стрічці в групі **Колонтитулы**.

Технологія виконання роботи.

Розглянемо процес створення звіту за допомогою майстра. На основі запиту «Замовлення» створити звіт з групуванням за **Товарними групами**, у середині за **Моделлю**. В якості підсумкових значень розрахувати загальну суму замовлення за групою.

Для виконання цього завдання необхідно перейти на вкладку **Создание** і у групі **Отчеты** вибрати режим **Мастер отчетов**. На першому етапі створення вказуємо, на основі якої таблиці або звіту слід побудувати звіт та обрати необхідні поля. У нашому випадку джерелом даних є запит «Замовлення», з якого ми вибираємо всі поля (рис. 5.1).

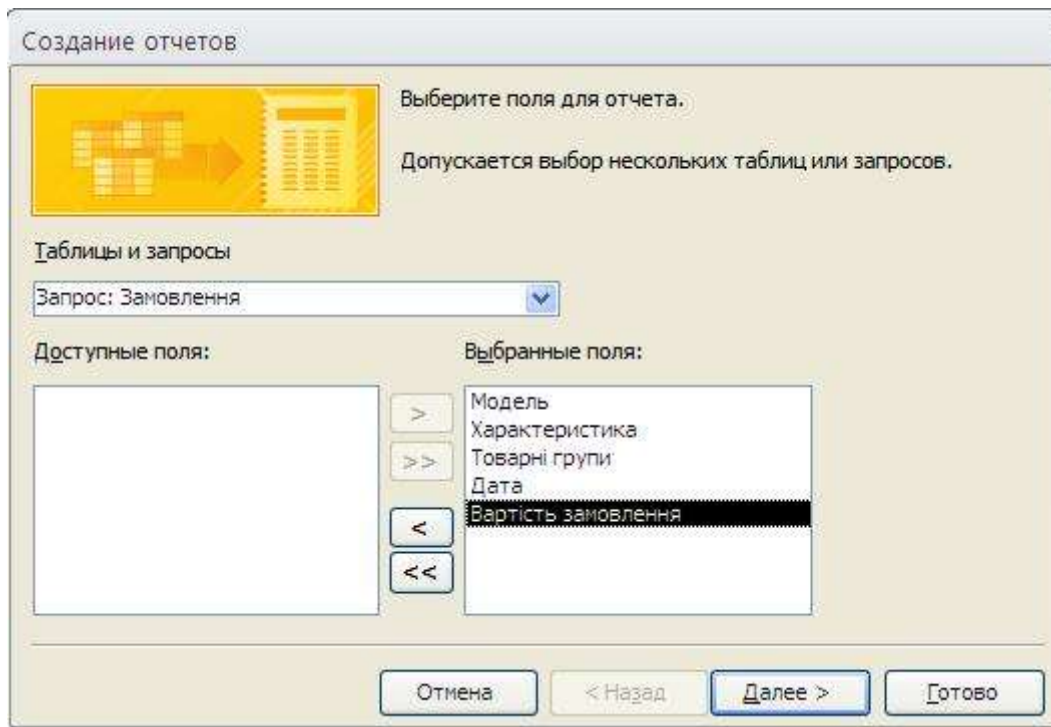


Рис. 5.1 – Діалогове вікно «Создание отчета»

Згідно умови додаємо рівні угруповання, переносимо вправо поля **Товарні групи**, **Модель** (рис. 5.2), натискаємо кнопку **Далее**.

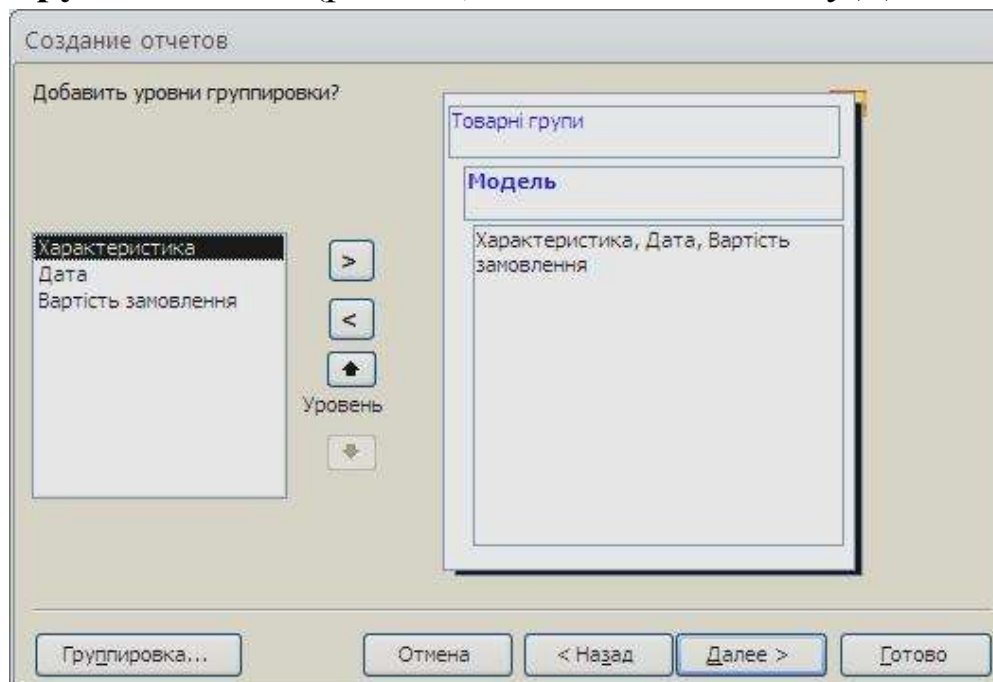


Рис. 5.2 – Діалогове вікно майстра звітів

Вказуємо сортування (рис. 5.3).



Рис. 5.3 – Зразок заповнення сортування

При перегляді звіту «Замовлення» перша сторінка з трьох виглядатиме таким чином, як зображено на рис. 5.4.

Рис. 5.4 – Режим перегляду звіту «Замовлення»

Для виконання даного завдання необхідно перейти на вкладку **Создание** і у групі **Отчеты** вибрати режим **Конструктор**. Після цього натискаємо на кнопку **Страница свойств** і заповнюємо вікно таким чином: у списку **Тип выделенного элемента** вказуємо **Отчет**, переходимо на вкладку **Данные** і в полі **Источник записей** вибираємо запит «Замовлення на двокамерні холодильники» (рис. 5.5).

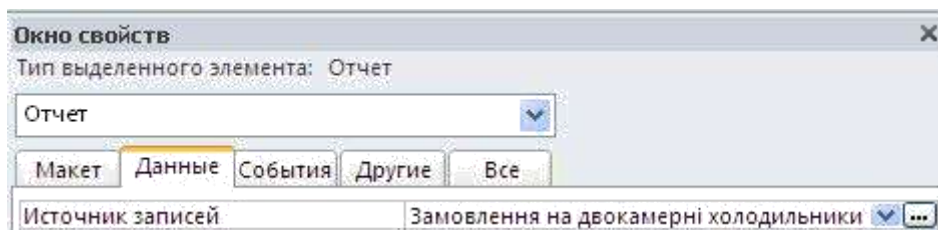


Рис. 5.5 – Фрагмент заповнення вікна «Свойства» в режимі конструктора звіту

На наступному кроці слід обрати список полів, що необхідні для створення даного запиту. Для цього натискаємо кнопку **Добавить поля**, в результаті чого у вікні **Список полей** будуть відображені всі доступні поля даного запиту.

Якщо при завантаженні конструктора відображаються не всі структурні елементи, необхідно додати їх за допомогою контекстного меню (рис. 5.6).

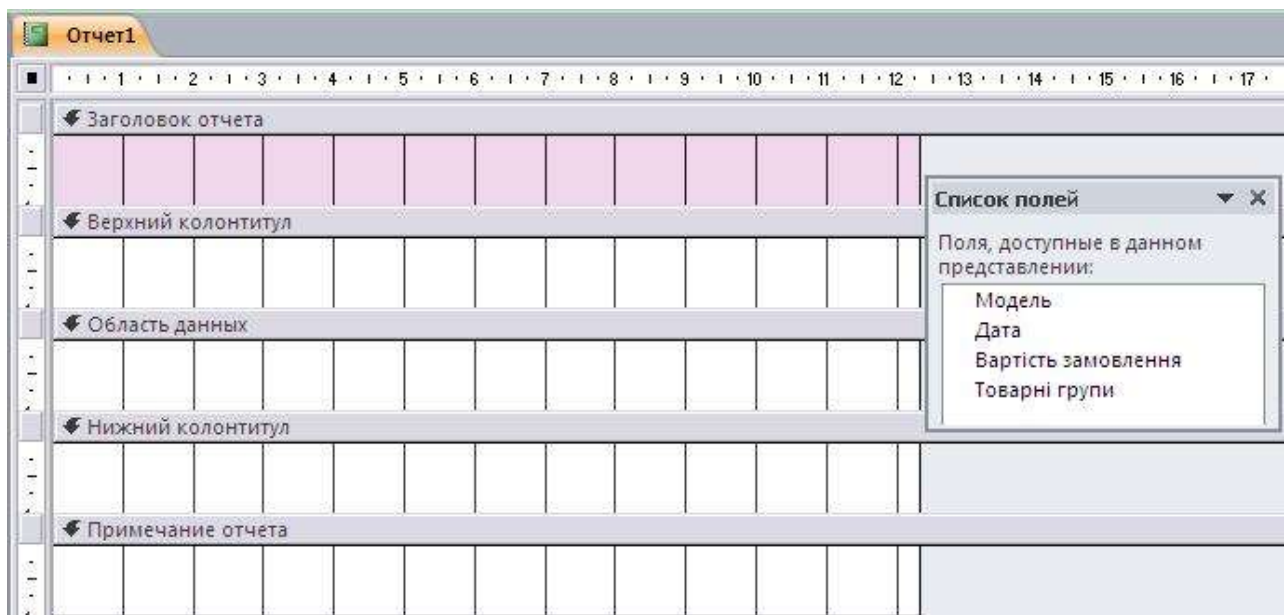


Рис. 5.6 – Структурні елементи звіту в режимі конструктора

У розділі **Заголовок отчета** за допомогою елементу управління **Аа** створюємо напис «Замовлення двокамерних холодильників» і форматуємо його за бажанням за допомогою інструментів, що розташовані на вкладці **Формат**.

Встановлюємо групування за допомогою кнопки **Группировка**, що знаходиться на вкладці **Конструктор**. Вибираємо рівні групування за Замовниками та Датами (рис. 5.7).

Після цих дій у конструкторі звіту з'являться ще два розділи: **Заголовок группы «Замовник»** і **Заголовок группы «Дата»**.

Розташовуємо поле **Замовник** в області **Заголовки** групи «Замовник».

Розташовуємо поле **Дата** в області **Заголовки** групи «Дата».

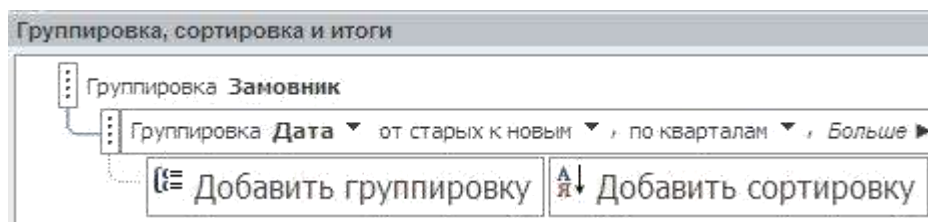


Рис. 5.7 – Групування даних у звіті

Решту полів (**Модель**, **Товарні групи** і **Вартість замовлення**) розташовуємо в області даних.

Встановимо підсумки для груп **Замовник** та **Дата**. Для цього в області **Группировка, сортировка и итоги** натиснути на пункт **больше**, обрати **без итогов** та встановити.

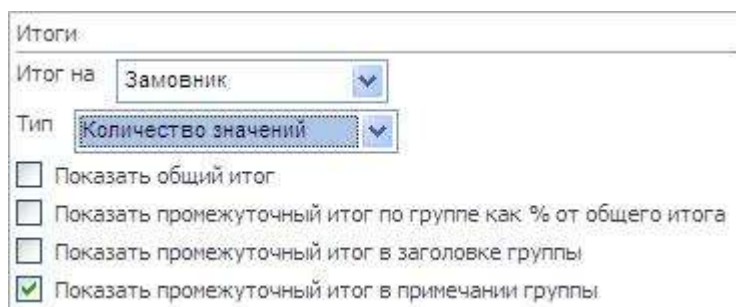


Рис. 5.8 – Встановлення підсумків для груп

Аналогічно зробити для поля **Дата**. Для розрахунку підсумкових значень:

В області **Примечание группы** «Замовник» створюємо об'єкт **а61** (Поле), де вказуємо:

Кількість записів за замовником:=Count([Модель])

В області **Примечание группы** «Дата» створюємо об'єкт **а61** (Поле), де вказуємо:

Кількість записів за датою:=Count([Модель])

В області **Примечание отчета** створюємо об'єкт **а61** (Поле), де вказуємо:

Загальна сума:=Sum([Вартість замовлення])

Змінюємо для об'єктів тип та розмір шрифту, колір тексту та заливки (за власним смаком). Встановимо тло у вигляді рисунка.

Для цього потрібно клацнути на звіті правою кнопкою миші, у контекстному вікні обрати **Свойства**, у верхній частині діалогового вікна обрати тип елемента Отчет, на закладці Все указати ім'я рисунка, що буде у якості тла.

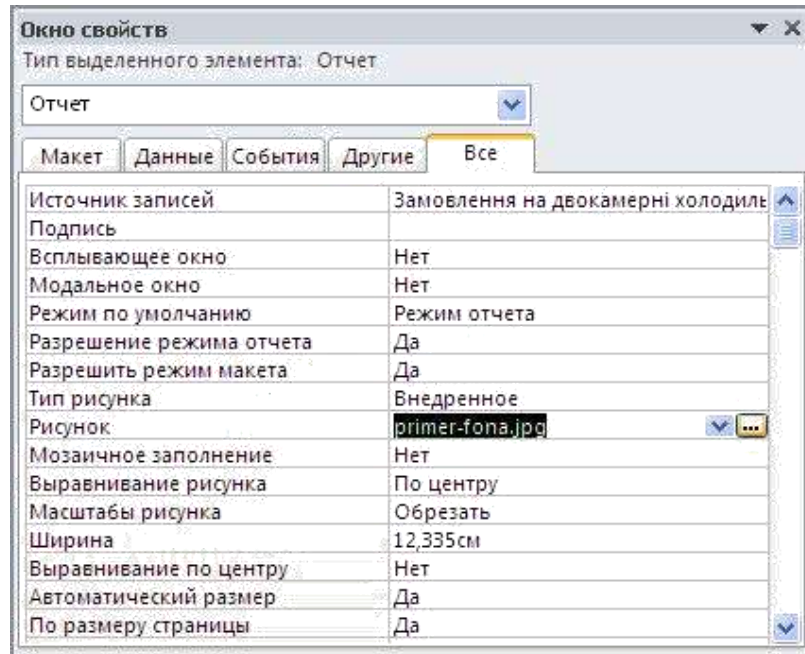


Рис. 5.9 – Встановлення рисунка у якості тла для звіту

Отримуємо в конструкторі таблицю, що зображено на рис. 5.10.

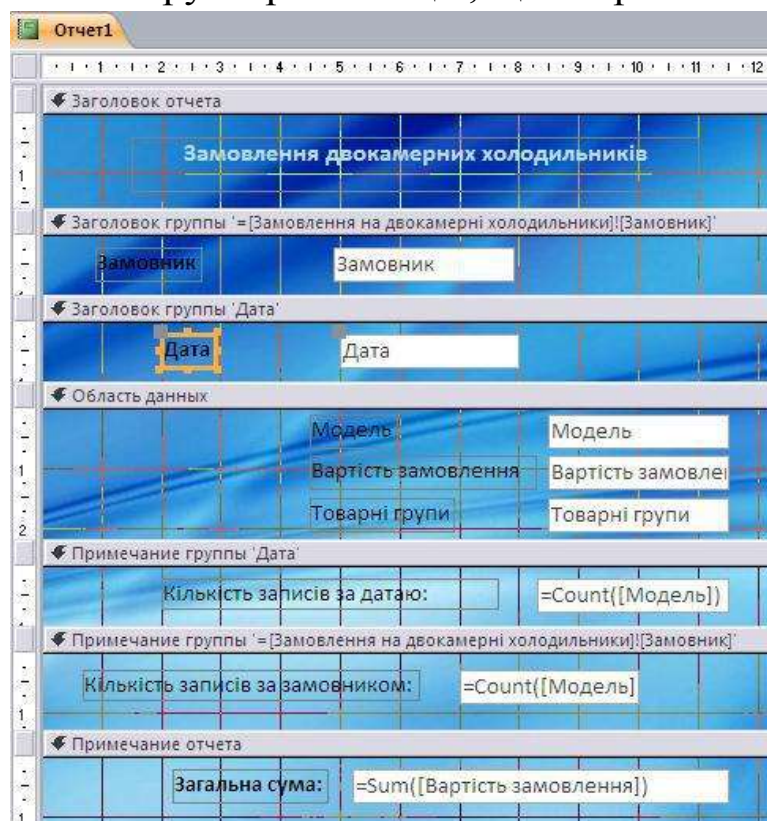



Рис. 5.10 – Звіт «Замовлення холодильників» в режимі конструктора

У примітці звіту можна також створити поле з поточною датою за допомогою елементу **abl** (Поле) та напис з ПІБ відповідального за підготовку документа **Aa** (Надпись) (рис. 5.11).



*Рис. 5.11– Створення елементів **Поле** та **Надпись***

В області заголовку або примітки звіту можна також створити рисунок за допомогою елементу  (Рисунок). Для зміни характеристик рисунку відкриваємо його властивості правою кнопкою миші:

Установка размеров – По размерам рамки

Выравнивание рисунка – По центру

У режимі **Представления отчета** отримуємо (рис. 5.12):

Рис. 5.12 – Звіт «Замовлення двокамерних холодильників» у режимі перегляду

ПРАКТИЧНА РОБОТА №6. Імпорт, експорт даних.

Мета роботи:

Сформувати вміння та навички імпорту та експорту даних між СУБД Access та іншими форматами файлів.

Теоретичні відомості.

Досить часто виникає ситуація, коли для бази даних Access потрібна інформація, яка зберігається у файлах інших форматів, або навпаки, необхідно розмістити інформацію бази в іншу програму. Процес переписування інформації Access у файли інших форматів називається *імпортом*, а інформації, що зберігається в інших форматах, в базу даних Access – експортом.

Програма Access підтримує два способи вилучення даних з інших додатків: імпорт, в процесі якого дані із зовнішнього джерела спочатку перетворюються у формат Access, а потім перетворені дані додаються в таблицю Access; зв'язування, при якому встановлюється тимчасовий канал комунікації між базою даних і зовнішнім джерелом.

Незалежно від того імпортуються дані чи зв'язуються, Access сприймає лише окремі формати файлів, які називаються сумісними (табл.6.1).

Таблиця 6.1 – Формати файлів сумісних з Access

Програма	Розширення файлів
Excel	.xls, .xlsx, .xlsb
Текст	.txt
XML	.xml
HTML	.htm, .html
Access Web-служби	.mdb, .adp, .mda, .mde, .ade, .accdb, .accda, .accde
Список SharePoint	–
ODBC	–
Outlook/Exchange	–
dBASE	.dbf

При імпорті файлу електронної таблиці кожен її стовбець стає стовпцем таблиці Access. Перший рядок електронної таблиці, що містить заголовки стовпців, можна перетворити в імена полів. Для цього необхідно встановити в Access прапорець **Первая строка содержит заголовки столбцов** у другому вікні майстра імпорту. Кожен наступний рядок електронної таблиці перетворюється у запис таблиці Access. В стовпці електронної таблиці повинні зберігатися дані тільки одного типу.

Якщо потрібно виконати імпортування із книги Excel, де на одному листі розміщені дані для декількох таблиць, то необхідно попередньо дати кожному діапазону даних ім'я. У вікні **Импорт электронной таблицы** вказати об'єкти імпорту – **именованные диапазоны**.

Інколи виникає необхідність імпортувати файл у форматі, що не підтримується Access. У таких випадках можна спочатку імпортувати джерело в текстовий файл, а потім – в Access.

Технологія виконання роботи.

Щоб імпортувати або зв'язати дані джерела з базою даних Access, необхідно виконати такі дії:

1. Відкрити базу даних Access, в яку повинна поступити інформація.
2. На стрічці активізувати вкладку **Внешние данные**.
3. Клацнути на інструменті, що призначений для початкового формату.
4. Вибрати джерело даних, яке необхідно імпортувати або зв'язати з Access.
5. Вибрати метод зберігання даних.
6. Пройти всі етапи імпорту або зв'язування, що залишилися.

Програма Access може експортувати дані в усі формати, з яких вона може імпортувати інформацію, а також у файли PDF і Microsoft Word. Щоб експортувати таблицю або результат запиту у файл іншого формату, необхідно виконати такі дії:

1. Відкрити базу даних. Клацнути на імені таблиці або запиту, результати якого потрібно експортувати.
2. Активізувати вкладку **Внешние данные**.
3. Клацнути на інструменті експорту, призначеному для формату, в який необхідно експортувати дані.
4. Виконати інструкції майстра, що залежать від вибраного цільового формату.
5. Встановити прапорець **Сохранить шаги экспорта**, щоб при необхідності швидко повторити експорт.

ПРАКТИЧНА РОБОТА №7. Мова запитів SQL.

Мета роботи:

Сформувати вміння та навички використання мови запитів SQL.

Теоретичні відомості.

Мова SQL (*Structured Query Language*) – одна з мов, що з'явилася в результаті розробки реляційної моделі даних. На даний момент ця мова отримала широке розповсюдження і перетворилася у стандартну мову реляційних баз даних. Ця мова дає можливість розробнику баз даних:

- створювати бази даних і таблиці, повністю описуючи їхню структуру;
- виконувати маніпулювання даними, використовуючи операції додавання, видалення і модифікації;
- створювати і виконувати прості та складні запити до бази даних.

Мова SQL достатньо проста і легкодоступна для вивчення. До її складу входять два основні компоненти:

- мова DML (Data Manipulation Language) – використовується для вибірки даних та їх оновлення;
- мова DDL (Data Definition Language) – використовується для описання структур баз даних і управління доступом до них.

Оператором, що найбільш використовується для створення запитів на вибірку є оператор SELECT. Результатом роботи цього оператора в MS Access 2010 є вибірка необхідних рядків з бази даних. Синтаксис оператора такий:

SELECT <Список стовпців>

FROM <Список таблиць> – визначає імена таблиць, які є джерелом записів;

[WHERE <Умова вибірки>] – дозволяє задавати вирази умови, що приймають значення істина або хибно;

[GROUP BY <Список полів групи>] – об'єднує записи з однаковим значенням, що знаходяться у вказаному списку полів, в один запис;

[HAVING <Умова>] – визначає згруповані записи, які повинні відображатися в операторі SELECT с GROUP BY;

[ORDER BY <Список атрибутів>] – упорядковує виведення запиту згідно значенням в тій чи іншій кількості вибраних стовпців.

Для вибірки з декількох таблиць слід використовувати зв'язку ключових слів INNER – JOIN – ON, що дозволяє попередити появу

дубльованих записів, за рахунок визначення зв'язків таблиць за окремими полями.

Технологія виконання роботи.

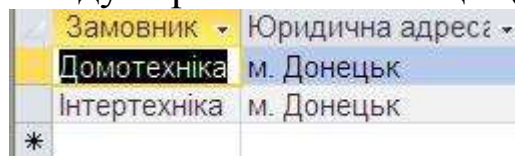
Приклад 1. Розглянемо приклад створення запиту на вибірку, що дозволить вибрати з бази даних «Реалізація продукції ПАТ "НОРД"» інформацію про донецьких замовників. Для цього скористуємося конструктор запитів таким чином:

1. Відкрити вкладку **Создание** на стрічці Access 2010.
2. Запустити конструктор запитів і закрити вікно **Добавление таблицы**.
3. Вибрати першу піктограму вкладки **Режим SQL**.
4. У вікні, що з'явилося набрати такий текст:

```
SELECT [Довідник замовників].Замовник,  
[Довідник замовників].[Юридична адреса] FROM [Довідник замовників]  
WHERE [Довідник замовників].[Юридична адреса]='м. Донецьк';
```

5. Закрити і зберегти запит.

В результаті перегляду отримаємо таблицю (рис. 7.1).



Замовник	Юридична адреса
Домотехніка	м. Донецьк
Інтертехніка	м. Донецьк

Рис. 7.1 – Запит на вибірку в режимі перегляду

Приклад 2. Запит, що дозволить вибрати інформацію про кількість замовленої продукції по датах та замовниках. Для цього у вікні режиму SQL слід ввести такий текст:

```
SELECT [Довідник замовників].Замовник,  
Реалізація.Дата, Реалізація.Кількість  
FROM [Довідник замовників] INNER JOIN Реалізація ON  
[Довідник замовників].[Код замовника]=Реалізація.[Код замовника]  
ORDER BY Реалізація.Дата;
```

Другий рядок говорить про те, що таблиці «Довідник замовників» і «Реалізація» зв'язані за допомогою поля **Код замовника**. Останній рядок дозволив відсортувати записи за датою. У результаті виконання даного запиту отримаємо таблицю (рис. 7.2):

Для запобігання появи записів, що повторюються використовують предикат **DISTINCT**. Наприклад, у таблиці «Реалізація» повторюються записи щодо придбаних товарів замовниками. Для виключення таких записів слід набрати такий текст у вікні режиму SQL:


```
SELECT DISTINCT Реалізація.[Код товару],
Реалізація.[Код замовника] FROM Реалізація;
```

Приклад 3. Вибір з бази даних інформацію про п'ять замовників, вартість замовлення яких найбільша. Для виконання даного завдання при створенні запиту слід використати предикат TOP. Такий запит в режимі SQL буде мати такий вигляд:

```
SELECT DISTINCT TOP 5 Замовлення.Замовник, Замовлення.[Вартість
замовлення], Замовлення.Дата
FROM Замовлення
ORDER BY [Замовлення.Вартість замовлення];
```

Замовник	Дата	Кількість
Донбас плюс	19.02.2009	60
Домотехніка	16.02.2010	80
Домотехніка	15.02.2011	80
Інтертехніка	17.02.2011	45
Донбас плюс	17.02.2011	20
Інтертехніка	17.02.2011	30
Інтертехніка	18.02.2011	50
Інтертехніка	18.02.2011	50
Норд-спрінт	18.02.2011	60
Інтертехніка	19.02.2011	90
Донбас плюс	19.02.2011	100
Донбас плюс	19.02.2011	40
Норд-спрінт	19.02.2011	45
Донбас плюс	19.02.2011	15
Інтертехніка	19.02.2011	52
Інтертехніка	19.02.2011	50
Норд-спрінт	11.03.2011	3
Донбас плюс	12.03.2011	1
Донбас плюс	12.03.2011	5
Донбас плюс	10.04.2012	11

Рис. 7.2 – Результуюча таблиця запиту на вибірку з декількох таблиць

Звернемо увагу, що в якості таблиці ми вказали запит «Замовлення». В результаті отримаємо таблицю, що показана на рис.7.3.

Замовник	Вартість зам	Дата
Донбас плюс	450,00р.	12.03.2011
Норд-спрінт	1 962,60р.	11.03.2011
Донбас плюс	2 250,00р.	12.03.2011
Донбас плюс	6 750,00р.	19.02.2011
Донбас плюс	7 196,20р.	10.04.2012

Рис. 7.3 – Результуюча таблиця запиту на вибірку з умовою

Якщо необхідно відобразити замовників з найменшою вартістю замовлення, слід змінити порядок сортування:

```
SELECT DISTINCT TOP 5 Замовлення.Замовник, Замовлення.[Вартість  
замовлення], Замовлення.Дата  
FROM Замовлення  
ORDER BY [Замовлення.Вартість замовлення] DESC;
```

Приклад 4. Для створення запитів на додавання записів використовується оператор INSERT, який дозволяє копіювати зміст однієї таблиці в іншу, або додавати записи в таблицю і заповнювати ці записи значенням. Синтаксис першої форми оператора має такий вигляд:

```
INSERT INTO <Ім'я таблиці>  
SELECT <Інструкція>
```

Наприклад, для копіювання всіх товарних груп з таблиці «Довідник товарних груп» у таблицю «Довідник товарів» можливо здійснити за допомогою такого запиту:

```
INSERT INTO [Довідник товарів]  
SELECT * FROM [Довідник товарних груп];
```

Якщо необхідно додати у таблицю записи зі значеннями, слід використовувати другу форму оператора INSERT, яка має вигляд:

```
INSERT INTO <Ім'я таблиці> [(Стовбець 1[, Стовбець 2[, ...]])] VALUES  
(Значення 1[, Значення 2[, ...]])
```

Наприклад, для додавання у таблицю «Довідник товарних груп» нову товарну групу необхідно створити такий запит:

```
INSERT INTO [Довідник товарних груп] ( [Довідник товарних груп].[Код  
товарної групи], [Довідник товарних груп].[Товарні групи] )  
VALUES (5, 'Дрібнабитоватехніка');
```

Приклад 5. Для створення запитів на зміну використовується оператор UPDATE, синтаксис якого має такий вигляд:

```
UPDATE <Ім'я таблиці >  
SET Стовбець1=Вираз1] [, Стовбець2=Вираз2] [...]  
WHERE <Умова відбору>
```

Слід зауважити ще раз, що запити на зміну призводять до змін в таблицях, які не можливо потім виправити.

Розглянемо приклад, в якому необхідно змінити адресу замовника «Норд-спрінт» на м. Донецьк. В режимі SQL такий запит буде мати такий вигляд:

```
UPDATE [Довідник замовників] SET [Довідник замовників].[Юридична  
адреса] = "м.Донецьк"  
WHERE [Довідник замовників].[Юридична адреса]="м. Подольське";
```

Приклад 6. Для створення запиту на видалення використовують оператор DELETE, який видаляє тільки дані, що відповідають заданим умовам, а структура таблиці і її властивості залишаються без змін. Синтаксис даного оператора виглядає таким чином:

```
DELETE <Ім'я таблиці > WHERE <Умова відбору>;
```

Для видалення інформації про реалізацію продукції з кількістю менш ніж 50 одиниць необхідно у вікні режиму SQL вести такий запис:

```
DELETE *  
FROM Реалізація  
WHERE Реалізація.Кількість<50;
```

Запитання для самоконтролю.

1. Що таке база даних?
2. Які основні функції системи управління базами даних?
3. Які моделі баз даних існують?
4. У чому полягає призначення таблиць, запитів, форм та звітів?
5. З яких елементів складається структура таблиці реляційної бази даних?
6. Що таке ключове поле? Які типи ключових полів Вам відомі?
7. Чи містить будь-яку інформацію таблиця, у якій немає жодного запису?
8. Які відносини між таблицями існують?
9. Які типи даних підтримує MS Access 2010?
10. Як встановити зв'язок між таблицями?
11. Які способи створення таблиць у MS Access 2010 Вам відомі?
12. Для чого призначені властивості полів на вкладці «ПОДСТАНОВКА»?
13. На основі яких об'єктів створюються запити?
14. Які типи запитів Вам відомі?
15. Які особливості запитів на зміну?
16. З яких етапів складається процес створення запитів?
17. Які оператори відбору значень при створенні запитів Вам відомі?
18. Які функції дати та часу для задачі умови відбору Вам відомі?
19. Які основні групові функції можна застосувати у підсумкових запитах? Які способи створення форм Вам відомі?
20. На базі яких об'єктів Access формується форми?
21. З яких частин складається структура форми?
22. Як оформити форму за допомогою рисунка?
23. Як видалити область виділення з форми?
24. Які режими створення звітів існують у Access 2010?
25. Яку структуру має звіт в режимі конструктору?
26. Які дані можуть розміщуватися в області *Примечание отчета* у звіті?
27. У чому полягає призначення імпорту, експорту даних у Access 2010?
28. Які можливості надає мова розробнику баз даних?

Варіанти практичних завдань для самостійної роботи.

Варіант 1

Створити базу даних «Агентство з продажів авіаквитків «Мрія». Агентство «Мрія» займається продажами авіаквитків на різні рейси, веде

облік проданих квитків і облік пасажирів, що придбали квитки. Тому виникає необхідність у зберіганні та обробці відповідних даних. Згідно запропонованої структури бази даних необхідно побудувати такі таблиці:

Розклад рейсів

номер рейсу	тип літака	пункт відправлення	пункт призначення	час відльоту	час польоту	кількість місць	ціна квитка
				Умова на значення	Умова на значення	Умова на значення	Умова на значення

Інформація про пасажирів

номер паспорта	ПІБ	адреса	телефон	пільги
			Маска введення	Значення: Так/Ні

Придбання квитків

код	номер рейсу	номер паспорта	кількість квитків	дата
	Підстановка з визначенням джерела рядків	Підстановка з визначенням джерела рядків	Умова на значення	Умова на значення, маска введення

Зв'язати таблиці бази даних та заповнити їх даними. Створити такі запити:

1. Створити запит на вибірку з наступною інформацією: тип літака, пункт відправлення, пункт призначення, ПІБ, вартість квитків.

2. Створити запит на вибірку, в результаті якого можна отримати дані (тип літака, пункт відправлення, пункт призначення, ПІБ, кількість та вартість квитків) про клієнтів, що придбали квитки до *Москви* у поточному місяці.

3. Створити підсумковий запит, який би відображав загальну кількість проданих білетів на рейси поточного місяця.

4. Створити параметричний запит, в результаті якого буде відображена інформація (тип літака, пункт відправлення, пункт

призначення, ПІБ, кількість та вартість квитків) за конкретними пунктами відправлення та призначення.

5. Створити запит, що здійснить видалення інформації про клієнтів, що придбали білети на рейси від певної дати.

6. На основі першого запиту створити запит на створення таблиці.

7. Створити перехресний запит по рейсу, даті, середній вартості квитків.

Розробити форми:

1. В режимі майстра створити форму «Довідник рейсів» на основі таблиці «Розклад рейсів». За її допомогою додати до таблиці 5 записів.

2. В режимі конструктора на основі таблиці «Клієнти агентства» створити форму «Інформація про пасажирів». За її допомогою додати до таблиці 5 нових записів.

Розробити звіти:

1. За допомогою майстра звітів створити звіт «Реалізація квитків». Здійснити угруповання даних по пункту відправлення, підсумкові значення для груп і звіту в цілому повинні містити загальну вартість проданих квитків.

2. За допомогою конструктора створити звіт «Реалізація квитків у поточному місяці». У звіті виконати угруповання по рейсах, всередині по даті. Підсумкові значення для груп і звіту в цілому повинні містити мінімальну та середню кількість проданих квитків.

Варіант 2

Створити базу даних «**Проектна організація**».

Проектна організація проектуванням будівель як для промислових, так і для непромислових цілей. Тому виникає необхідність у зберіганні та обробці даних про співробітників, проекти та терміни їх виконання. У виконанні одного проекту можуть приймати участь декілька співробітників. Необхідно вести облік виконання етапів проекту.

Згідно запропонованої структури необхідно побудувати такі таблиці:

Довідник посад

код посади	найменування	оклад
		<i>Умова на значення</i>

Довідник співробітників

код співробітника	ПІБ	код посади	дата народження	дата прийому на роботу	відділ
		<i>Підстановка з визначенням джерела рядків</i>	<i>Умова на значення, маска введення</i>	<i>Умова на значення, маска введення</i>	

Довідник проектів

код проекту	назва проекту	керівник	дата початку	дата закінчення	бюджет
			<i>Умова на значення, маска введення</i>	<i>Умова на значення, маска введення</i>	<i>Умова на значення</i>

Виконання проектів

код	код проекту	код співробітника	опис етапу роботи	дата закінчення етапу
	<i>Підстановка з визначенням джерела рядків</i>	<i>Підстановка з визначенням джерела рядків</i>		<i>Умова на значення, маска введення</i>

Зв'язати таблиці бази даних та заповнити їх даними. Створити такі запити:

1. Створити запит на вибірку з такою інформацією: назва проекту, керівник, дата початку і закінчення, ПІБ співробітника, найменування посади, опис етапу роботи, дата закінчення етапу.
2. Створити запит на вибірку, в результаті якого можна отримати дані (назва проекту, керівник, дата початку і закінчення, ПІБ співробітника, найменування посади, опис етапу роботи, дата закінчення етапу) про співробітників, що закінчують виконання етапів роботи над проектами у поточному місяці.
3. Створити підсумковий запит, який би відображав загальну кількість співробітників, що працюють над проектами.
4. Створити параметричний запит, в результаті якого буде відображена інформація (назва проекту, дата початку і закінчення, ПІБ співробітника, опис етапу роботи, дата закінчення етапу) за конкретним керівником.
5. Створити запит, що збільшить посадовий оклад співробітників на 250 грн.

6. На основі першого запиту створити запит на створення таблиці.
7. Створити перехресний запит по проекту, керівника, середньому бюджету.

Розробити форми:

1. В режимі майстра створити форму «Реєстр співробітників» на основі таблиці «Довідник співробітників». За її допомогою додати до таблиці 5 записів.
2. В режимі конструктора на основі таблиці «Довідник проектів» створити форму «Проекти». За її допомогою додати до таблиці 5 нових записів.

Розробити звіти:

1. За допомогою майстра звітів створити звіт «Виконання проектів». Здійснити угруповання даних по назві проекту, посаді, підсумкові значення для груп і звіту в цілому повинні містити загальну кількість співробітників.
2. За допомогою конструктора створити звіт «Виконання проектів у поточному місяці». У звіті виконати угруповання назві проекту, посаді. Підсумкові значення для груп і звіту в цілому повинні містити мінімальну та середню кількість задіяних співробітників.

Варіант 3

Створити базу даних «Спортивні змагання». Спортсмени під керівництвом тренерів займаються окремими видами спорту та приймає участі у різних спортивних заходах. При цьому у рамках одного змагання спортсмен може брати участі у декількох секціях. Всі результати змагань заносяться у таблицю. За результатами участі спортсменів у змаганнях відбувається нагородження.

Згідно запропонованої структури бази даних необхідно побудувати такі таблиці:

Секції

код секції	секція	опис

Спортивні заходи

код заходу	назва заходу	місто проведення	дата проведення
			Умова на значення, маска введення

Учасники

Код учасника	ПІБ учасника	ПІБ тренера	Вік	Клуб
			Умова на значення	

Результати змагання

Код	Код учасника	Код секції	Код заходу	Зайняте місце
	Підстановка з визначенням джерела рядків	Підстановка з визначенням джерела рядків	Підстановка з визначенням джерела рядків	Умова на значення

Зв'язати таблиці бази даних та заповнити їх даними. Створити такі запити:

1. Створити запит на вибірку з наступною інформацією: П.І.Б. учасника, тренера, клуб, назва заходу, секція, зайняте місце.
2. Створити запит на вибірку, в результаті якого можна отримати дані (назва заходу, П.І.Б. учасника, тренера, зайняте місце, секція) про конференції, що відбулися у поточному році.
3. Створити підсумковий запит, який би відображав загальну кількість учасників по кожній секції.
4. Створити параметричний запит, в результаті якого буде відображена інформація (назва заходу, секція, місце) по конкретному місту проведення.
5. Створити запит, в результаті якого в таблиці «Спортивні заходи» для заходів, що відбудуться у наступному місяці місто проведення м. Донецьк зміниться на м. Маріуполь.
6. Створити перехресний запит по назві заходу, секції.

Розробити форми:



1. В режимі майстра створити форму «Список спортивних заходів» на основі таблиці «Спортивні заходи». За її допомогою додати до таблиці 5 записів.
2. В режимі конструктора на основі таблиці «Учасники» створити форму «Відомості про учасників». За її допомогою додати до таблиці 5 нових записів.

Розробити звіти:

1. За допомогою майстра звітів створити звіт «Учасники змагань». Здійснити угруповання даних по секції, підсумкові значення для груп і звіту в цілому повинні містити загальну кількість учасників.
2. За допомогою конструктора створити звіт «Учасники змагань у поточному році». У звіті виконати угруповання по назві заходу, всередині по секції. Підсумкові значення для груп і звіту в цілому повинні містити загальну кількість учасників.

Зразки тестових завдань

1. Таблиці служать для:
 - a) зберігання інформації
 - b) ведення розрахунків
 - c) вибірки інформації
 - d) виводу інформації на друк
2. Укажіть способи створення таблиць:
 - a) режим конструктора
 - b) за допомогою майстра
 - c) шляхом вводу даних
 - d) за допомогою макросу
3. Властивість автоматичного нарощування має поле:
 - a) числове
 - b) лічильник
 - c) логічне
 - d) МЕМО
4. Текстове поле дозволяє занести інформацію до:
 - a) 256 символів
 - b) 20 символів
 - c) 65536 символів
 - d) 56 символів
5. Яку умову відбору потрібно задати, щоб вибрати всі товари за ціною, більше або рівної 350 грн?
 - a) ≤ 350
 - b) ≥ 350
 - c) $=> 350$
 - d) $\leq=> 350$
6. Як називається категорія запитів, призначених для вибору даних з таблиць бази?
 - a) запит на вибірку
 - b) підсумковий запит
 - c) запит з параметром
 - d) запит на обчислення
7. Вкажіть, для полів яких типів даних в умовах відбору використовують оператор Like?
 - a) для текстових полів
 - b) для числових полів
 - c) для грошових полів
 - d) для поля МЕМО

8. Як встановити, щоб форма запускалася автоматично при відкритті бази даних?
- а) виконати команду головного меню: Сервис – Параметры
 - а) у поле Выводформы/страницы ввести ім'я головної форми.
 - б) виконати команду головного меню: Сервис – Параметры – закладка Вид. Встановити прапорець *Начинать с текущей*
 - с) Натиснути на кнопку  на панелі інструментів. Ввести ім'я головної форми.
9. На базі яких об'єктів Access формується звіт?
- а) запити
 - б) форми
 - с) таблиці
 - д) макроси
 - е) модулі
10. Для групування даних у звіті необхідно зробити...
- а) Вид – Сортировка и группировка
 - б) Вибрати  на панелі інструментів
 - с) Виділити поля та в контекстному меню обрати *Группировать*

Критерії оцінки знань

Контроль знань і умінь студентів (поточний і підсумковий) з навчальної дисципліни «Технологія проектування і адміністрування БД і СД» здійснюють згідно з кредитно-модульною системою організації навчального процесу.

Поточний – під час виконання практичних робіт, індивідуальних завдань (описових робіт, написання рефератів). Контроль за засвоєнням певного модуля (модульний контроль) проводять у вигляді тестового контролю знань із змістового модуля навчальної дисципліни.

Підсумковий – включає залік.

Навчальна дисципліна «Технологія проектування і адміністрування БД і СД» складається з 2-х модулів. Кожен модуль оцінюється в умовних балах пропорційно обсягу часу, відведеному на засвоєння матеріалу цього модуля. Максимально можлива кількість умовних балів за навчальні заняття студента становить 100%.

Студент може збільшити свій рейтинг за навчальну роботу на величину додаткового рейтингу ($R_{др}$), визначену лектором. Навчальну роботу вводять за рішенням кафедри під час виконання робіт, що не передбачені навчальним планом, але сприяють підвищенню кваліфікації студентів із навчальної дисципліни (доповідь на студентській конференції, здобуття призового місця на II-му етапі всеукраїнської олімпіади, виготовлення макетів, підготовку наочних посібників тощо). Рейтинг із додаткової роботи ($R_{др}$) може становити до 10 балів. $R_{др}$ додається до $R_{нр}$ (рейтинг з навчальної роботи).

Рейтинг штрафний ($R_{штр}$) віднімається від $R_{нр}$ і може становити до 5 балів. $R_{штр}$ визначає лектор і вводять за рішенням кафедри для студентів, які невчасно засвоїли матеріал модуля, не дотримувалися графіка роботи, пропускали заняття тощо.

Для допуску до атестації (заліку) студенту необхідно набрати не менше 50% балів від рейтингу з навчальної роботи $R_{нр}$. Це означає, що в цілому студенту необхідно виконати такий мінімум робіт:

- виконати всі заплановані практичні завдання (лабораторні роботи, розрахункові завдання, домашні завдання, контрольні роботи);

Рейтинг з атестації Rat включає рейтинг з заліку Rіс і визначається кількістю балів, отриманих студентом на атестації з дисципліни і передбачених робочим навчальним планом.

Студенти, які протягом семестру набрали необхідну кількість балів (не менше 60% від розрахункового рейтингу дисципліни, тобто 60 балів), мають можливість:

- не складаючи залік отримати залік – автоматично, відповідно до набраної за семестр кількості балів;
- скласти залік з метою підвищення рейтингу з дисципліни.

Критерії оцінки виконання навчальних завдань є одним з основних способів перевірки знань, умінь і навичок студентів з дисципліни «Технологія проектування і адміністрування БД і СД». При оцінці завдань за основу слід брати повноту і правильність їх виконання. Необхідно враховувати наступні навички і вміння студентів (студент вміє):

- диференціювати, інтегрувати та уніфікувати отримані знання;
- викладати матеріал логічно й послідовно;
- користуватися додатковою літературою.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Дейт К. Введение в системы баз данных / К. Дейт. – М. : Наука, 2011. – 630 с.
2. Гурвиц Г. А. Microsoft Access 2010. Разработка приложений на реальном примере / Г. А. Гурвиц. – СПб : БХВ-Петербург, 2010. – 484 с.
3. Мейер М. Теория реляционных баз данных : пер. с англ. / М. Мейер. – М. : Мир, 2011. – 608 с.
4. Тиори Т. Проектирование структур баз данных / Т. Тиори. – М. : Мир, 2012. – 496 с.
5. Сеннов А. С. Access 2010 / А. С. Сеннов – СПб. : Питер, 2010. – 288 с.
6. Бекаревич Ю. Самоучитель. Access 2010 / Ю. Бекаревич, Н. Пушкина. – СПб. : БХВ-Петербург, 2013. – 432 с.
7. Англо-український тлумачний словник з обчислювальної техніки, Інтернету і програмування. – 2-ге вид. – К. : Видавничий дім "СофтПрес", 2006. – 824 с.

Навчальне видання

Технологія проектування та адміністрування БД і СД

Методичні рекомендації

Укладач: **Волосюк** Юрій Вікторович

Формат 60x84 1/16 Ум. друк. арк. 2,6.

Тираж 30 прим. Зам. №_____

Надруковано у видавничому відділі
Миколаївського національного аграрного університету.
54020 м. Миколаїв, вул. Георгія Гонгадзе, 9

Свідоцтво суб'єкта видавничої справи ДК № 4490 від 20.02.2013 р.