**Створення запитів**

Хоча в режимі таблиці доступні самі різні операції зданими - перегляд, сортування, фільтрація, - запити є найкращим способом добору необхідних даних. Запити дозволяють проводити обчислення, а також відбирати необхідні дані з декількох таблиць.

**Створення запитів на вибірку**

Запити на вибірку створюються двома основними способами: за допомогою конструктора запитів і майстри запитів. Незалежно від того, який із способів використовується, для створення запиту на вибірку необхідно виконати кілька стандартних дій.

* 1. Спочатку потрібно вибрати джерело записів для запиту. Джерелом записів можуть бути одна або декілька таблиць, один або декілька запитів або поєднання цих елементів.
* 2. У джерелі записів слід вибрати нуля, які повинні бути включені в запит.
* 3. Додати в запити умови сортування, фільтрації та інші умови відбору.

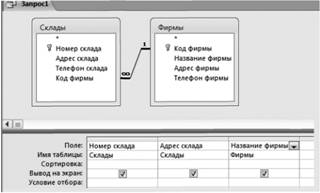
При виконанні запиту на вибірку MS Access виводить результати в таблицю. Результат роботи запиту називається набором записів, і з ними можна працювати точно так само, як з таблицею. Наприклад, можна додавати і змінювати дані, і неї внесені зміни будуть збережені в таблицях, які послужили джерелом записів для запиту.

Крім того, набір записів може служити джерелом даних для форм, звітів і навіть інших запитів.

Розглянемо на прикладі створення запиту в режимі конструктора.

Виберіть команду **Конструктор запитів** в групі **Запити** на вкладці Створення.

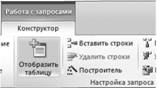
Вікно конструктора запитів (рис. 10.29) розділене на дві частини. У верхній частині повинні знаходитися макети таблиць або запитів, на основі яких створюється запит. У нижній частині бланка виконується вся робота по створенню запиту.



***Мал. 10.29.*** **Вікно запиту в режимі конструктора**

Щоб помістити таблиці на бланк запиту, скористайтеся діалоговим вікном **Додавання таблиці** (викликати дане вікно можна в режимі конструктора запиту за допомогою команди **Показати таблицю** в групі **Налаштування запиту** на вкладці **Робота з**запитами, рис. 10.30).

Кожен стовпець бланка представляє одне поле, що використовується в запиті. Вибрати поля можна кількома спосо-



***Мал.***10.30. **Кнопка Показати таблицю**

бами. Наприклад, потрібно виділити потрібне поле у верхній частині бланка запиту і двічі клацнути на ньому.

**Установка властивостей полів.** Поля, що виводяться в наборі записів запиту, успадковують властивості, задані для відповідних полів таблиці. Іноді, наприклад у випадку обчислюваних полів, виникає необхідність задати інші значення властивостей: формат поля, число десяткових знаків, підпис.

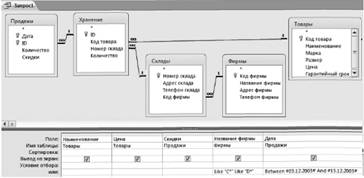
Щоб задати властивості поля, відкрийте запит в режимі конструктора, встановіть курсор в це поле в нижній частині бланка запиту, а потім натисніть на кнопку **Сторінка властивостей** в групі **Показати або приховати** на вкладці **Робота з запитами** (рис. 10.31).

Кнопка для виклику вікна властивостей

***Мал. 10.31.*** **Кнопка для виклику вікна властивостей**

**Введення умов відбору.** Введення умови відбору в запиті аналогічний завданням умови на значення для поля таблиці. Якщо необхідно відібрати записи, що задовольняють умові, введіть його в рядку "Умова відбору" в нижній частині бланка запиту.

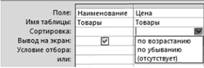
Як приклад розглянемо створення запиту, отображающего товари, продані фірмами Citilink і Dinikin з 5 по 15 грудня 2003, вказавши найменування товару, його ціну і розмір знижки (рис. 10.32).



***Мал. 10.31.*** **Приклад створення простого запиту**

**Сортування блоків даних у запиті.** Блоки даних в запиті можуть бути розсортовані алфавітним або числовим способом в зростаючій (А - Я, 0 - 9) або спадної (Я - А, 9 - 0) послідовності по вмісту окремих полів. Можна одночасно проводити сортування по вмісту декількох полів (до десяти):

* • клацніть мишею в рядку "Сортування" того стовпця (поля), за яким необхідно провести сортування;
* • вкажіть спосіб сортування (рис. 10.33).

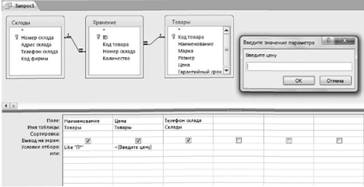


***Мал. 10.33.*** **Сортування даних в запиті**

# Запит з параметром (параметричний запит)

Як правило, запити з параметром (рис. 10.34) створюються в тих випадках, коли передбачається виконувати цей запит багаторазово, змінюючи лише умови відбору.

Щоб визначити параметр, введіть у рядок "Умова відбору" замість конкретного значення ім'я або фразу, укладену в квадратні дужки ([]). Те, що укладено всередині квадратних дужок, Access розглядає як ім'я параметра. Воно виводиться в вікні діалогу при виконанні запиту, тому в якості імені параметра розумно використовувати змістовну фразу. У запиті можна задати



***Мал. 10.34.*** **Приклад створення запиту з параметром**

декілька параметрів; при цьому ім'я кожного з них має бути унікальним і інформативним.

Наприклад, потрібно створити параметричний запит для відображення процесорів вартістю до певної суми, їх найменування, ціни і телефону складу, де вони зберігаються (параметром є ціна товару).

# Перехресний запит

Перехресний запит - особливий тип підсумкового запиту. Він дозволяє вивести обчислювані значення в перехресної таблиці, яка нагадує електронну.

Для побудови перехресного запиту виконайте такі дії.

У режимі конструктора сформуйте запит на вибірку.

Змініть тип запиту на **Перехресний** (вкладка **Робота з**запитами, група **Тип**запиту) (рис. 10.35).

Кнопка Перехресний запит

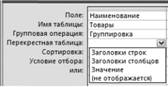
***Мал. 10.35.*** **Кнопка Перехресний запит**

У нижній частині бланка запиту з'являться рядки "Групова операція" і "Перехресна таблиця" (рис. 10.36).

У рядку "Перехресна таблиця" для кожного поля запиту виберіть одну з чотирьох установок: ***Заголовки рядків, Заголовки стовпців. Значення*** (виведене в осередках перехресної таблиці) або ***не відображається.***

Для перехресного запиту визначте поле (можна декілька) в якості заголовків рядків. Визначте одне (і тільки одне) поле, яке буде використовуватися в якості заголовків стовпців.

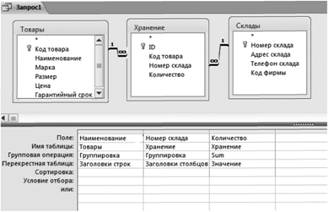
Визначте одне (і тільки одне) поле значень. Це поле повинно бути підсумковим (тобто в рядку "Групова операція" повинна бути задана одна з підсумкових функцій, на-



***Мал. 10.36.*** **Конструктор Перехресного запиту**

приклад підсумовування (Sum), визначення середнього значення (Avg) або кількості (Count)).

Як приклад розглянемо створення перехресного запиту, отображающего кількість товарів, що зберігаються на складах. У заголовках рядків вкажемо найменування товарів, в заголовках стовпців - номери складів (рис. 10.37).



***Мал. 10.37.*** **Приклад створення перехресного запиту**

# Обчислювані поля в запитах

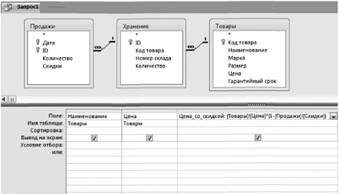
Запит можна використовувати для виконання розрахунків і підбиття підсумків з вихідних таблиць.

Для створення обчислюваних полів можна використовувати арифметичні оператори і будь вбудовані функції Access.

Обчислюване поле може також містити результат конкатенації (об'єднання) значень текстових полів. Для цього укладіть текст в лапки, в якості оператора конкатенації використовується символ "&". Наприклад, можна створити поле, яке буде містити результат об'єднання поля [Прізвище] і поля [Ім'я].

Поле, вміст якого є результатом розрахунку але вмісту інших полів, називається ***обчислюваним полем.*** Обчислюване поле існує тільки в результуючій таблиці.

Створення обчислюваного поля здійснюється шляхом простого введення вирази для обчислення в осередок "Поле" порожнього стовпця бланка запиту (рис. 10.38).

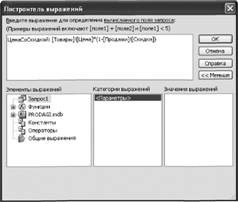


***Мал. 10.38.*** **Створення обчислюваного поля**

Після виконання запиту в результуючій таблиці з'явиться нове иоле з назвою "Вираз1", використовуваним як імені обчисленого виразу.

У режимі конструктора запиту змініть ім'я "Вираження!" на більш змістовне. У прикладі "Вираження!" змінено на "ЦенаСоСкідкой".

Для того щоб ввести складні обчислення, використовуйте вікно П **остроітель виразів** (рис. 10.39), яке викликається командою, розташованої на вкладці **Робота з запитами** (в режимі конструктора) у групі **Налаштування**запиту, або відповідною командою контекстного меню.



***Мал. 10.39.*** **Будівник виразів**

У верхній частині вікна розташована порожня область введення, призначена для створення вирази; в нижній знаходяться три списки, призначені для пошуку необхідних полів і функцій.

Будівник допоможе правильно побудувати вираз. Клацніть на кнопці **ОК, і введене вираз буде перенесено в бланк запиту**

**Підсумкові запити**

Для отримання підсумкових значень по групах даних використовуються ***підсумкові запити.*** Для завдання обчислень підсумкових значень відкрийте запит в режимі конструктора, клацніть на кнопці **Підсумки** в групі **Показати або приховати** на вкладці **Робота з**запитами, щоб в бланку запиту з'явився рядок "Групова операція". Тоді записи по кожному полю будуть групуватися. Для обчислення підсумків замініть значення **Угруповання** в рядку "Групова операція" на конкретну підсумкову функцію.

MS Access надає дев`ять функцій, що забезпечують виконання групових операцій (табл. 10.4).

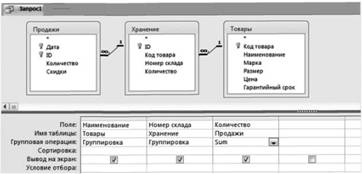
Наприклад, необхідно обчислити, скільки всього було продано даного товару з даного складу за весь час продажів.

***Таблиця 10.4***

**Функції для групових операцій**

|  |  |
| --- | --- |
| Функція | Опис |
| Sum | Підсумовування значень певного поля |
| Avg | Обчислення середнього значення даних певного поля |
| Min | Обчислення мінімального значення поля |
| Max | Обчислення максимального значення поля |
| Count | Обчислення кількості записів, відібраних запитом за умовою |
| First | Визначається перше значення у вказаному нулі записів, відібраних запитом |
| Last | Визначається останнє значення у вказаному полі записів, відібраних запитом |
| StDev | Обчислюється стандартне відхилення значень даного поля для всіх записів, відібраних запитом |
| Var | Обчислюється варіація значень даного поля для всіх записів, відібраних запитом |

Так як один і той же товар міг продаватися не один раз, використовуємо угруповання по полях [Найменування] і [НомерСклада], а по полю [Кількість] вибираємо функцію підсумовування (рис. 10.40).



***Мал. 10.40.*** **Підсумковий запит**

**Модифікація даних за допомогою запитів**

Щоб убезпечити себе від випадковостей, перед створенням запитів на модифікацію даних створіть резервні копії таблиць, дані яких будуть модифікуватися.

**Запит на створення**таблиці. БД на фізичному рівні зберігає тільки таблиці. Набір записів запитів фізично не існує в БД. MS Access створить його з даних таблиць тільки під час виконання запиту. Іноді виникає необхідність зберегти витягувані за допомогою запиту на вибірку дані в новій таблиці. Для цього:

* • створіть новий запит на вибірку тих записів, з яких повинна складатися створювана за допомогою запиту таблиця;
* • перевірте правильність відбору записів, перейшовши в режим таблиці;
* • перетворіть запит на вибірку в запит на створення таблиці. Для цього, повернувшись в режим конструктора, виберіть кнопку **Створення таблиці** у групі **Тип запиту** на вкладці **Робота з запитами** (рис. 10.41);
* • у вікні введіть ім'я нової таблиці;
* • виконайте запит (кнопка).

Кнопка Створення таблиці

***Мал. 10.41.*** **Кнопка Створення таблиці**

**Запит на оновлення.** Використовуючи цей тип запиту, можна змінити в базовій таблиці групу блоків даних, відібрану па основі певних критеріїв:

* • створіть новий запит на вибірку і перевірте його коректність, перейшовши в режим таблиці;
* • перетворіть запит на вибірку в запит на оновлення; для цього, повернувшись в режим конструктора, виберіть команду **Оновлення** (вкладка **Тип**запиту);
* • в з'явилася в бланку запиту рядку "Оновлення" у шпальтах задайте нові значення полів таблиці. В якості таких можуть виступати й обчислювані значения; у разі потреби скористайтеся інструментом "Будівник виразів";
* • виконайте запит.

**Запит на додавання записів.** За допомогою цього типу запиту блоки даних однієї таблиці (всі або відібрані запитом) можна приєднати в кінець іншої таблиці:

* • створіть новий запит на вибірку тих блоків даних, які будуть додані в деяку таблицю, і перевірте його коректність, перейшовши в режим таблиці ;.
* • перетворіть запит на вибірку в запит на додавання; для цього, повернувшись в режим конструктора, виберіть команду **Додавання ...** (вкладка **Тип**запиту);
* • у вікні введіть ім'я таблиці, до якої потрібно приєднати дані, і натисніть кнопку ОК;
* • виконайте запит.

**Запит на видалення записів.** За допомогою даного типу запиту можна видалити з базової таблиці групу блоків даних, відібраних за певними критеріями. При цьому слід ретельно проаналізувати критерії відбору, оскільки цю операцію не можна скасувати. Для здійснення запиту потрібні наступні дії:

* • створіть новий запит на вибірку видаляються блоків даних; відбір блоків даних виконується відповідно до заданими в рядку "Умова" критеріями;
* • перевірте коректність сформульованих умов, перейшовши в режим таблиці;
* • перетворіть запит на вибірку в запит на видалення записів; для цього, повернувшись в режим конструктора, виберіть команду **Видалити** (вкладка **Тип**запиту);
* • в з'явилася рядку "Видалити" встановіть критерії відбору;
* • виконайте запит.

Розглянемо створення запитів на модифікацію на прикладі.

**Приклад 10.3.** Створимо ***запит на створення таблиці*** [Монітори], отриманої з таблиці [Товари]. Таблиця [Монітори] повинна містити всі поля таблиці [Товари] і тільки ті записи, в яких найменування товару - Монітор.

***Рішення***

1. Створимо запит на вибірку даних про моніторах з таблиці [Товари] (рис. 10.42), попередньо створивши копію таблиці [Товари].

Запит на вибірку даних про моніторах

***Мал. 10.42.*** **Запит на вибірку даних про моніторах**

Для вибору всіх полів таблиці [Товари] (рис. 10.43) використана "\*". Умова відбору задано для поля [Найменування]. З цслио уникнути дублювання поля | Найменування! в таблиці [Монітори] відсутня "галочка" в рядку ***Вивід на екран.***

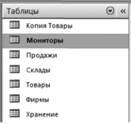


***Мал. 10.43.*** **Запит на створення таблиці**

* 2. Змінимо тип запиту на вибірку в запит на створення таблиці, при цьому з'явиться вікно для введення імені створюваної таблиці (див. Рис. 10.44).
* 3. Виконаємо запит, при цьому з'явиться попереджувальне вікно (див. Рис. 10.44).
* 4. Підтвердимо створення нової таблиці з вибраними записами, в результаті в списку таблиць БД побачимо нову таблицю [Монітори] (рис. 10.45).



***Мал. 10.44.*** **Вікно підтвердження створення таблиці**

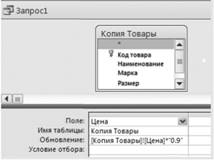


***Мал. 10.45.*** **Поява нової таблиці в списку таблиць**

**Приклад 10.4.** Створимо запит на оновлення цін товарів з урахуванням сезонних знижок на 10%.

***Рішення***

* 1. Попередньо створимо копію таблиці [Товари], присвоївши їй ім'я [Копія Товари].
* 2. Створимо запит на вибірку (рис. 10.46), що відображає поле [Ціна] (вибираємо тільки те поле, значення якого повинні бути оновлені).
* 3. Змінимо тип запиту на вибірку в запит на оновлення.
* 4. У рядку **Оновлення** введемо вираз, оновлююче значення поля, за допомогою Будівника виразів.
* 5. Виконаємо запит.



***Мал. 10.40.*** **Запит на оновлення**

# Створення запитів з використанням мови SQL

Насправді будь-який запит в MS Access реалізується за допомогою мови SQL. Більшість запитів можна побудувати, користуючись засобами, які Access надає в режимі конструктора, і в цьому випадку вони будуть зберігатися у вигляді інструкцій SQL. Але деякі типи запитів можна побудувати, тільки використовуючи мову SQL.

**Синтаксис інструкції SELECT в MS Access.** Ядром мови SQL є інструкція SELECT. Вона використовується для відбору рядків і стовпців з таблиць бази даних і містить п'ять основних пропозицій. У загальному випадку її синтаксис можна представити в наступному вигляді:

SELECT <список-полів> [[1]](https://stud.com.ua/53372/informatika/stvorennya_zapitiv_vikoristannyam_movi" \l "srcannot_1)

FROM <список-таблиць>

[WHERE <специфікація-відбору-рядків>] [[2]](https://stud.com.ua/53372/informatika/stvorennya_zapitiv_vikoristannyam_movi" \l "srcannot_2) [GROUP BY <специфікація-угруповання>]

[HAVING <специфікація-відбору-груп>]

[ORDER BY <специфікація-сортування>]

У MS Access реалізовані важливі засоби розширення мови: інструкція TRANSFORM, що дозволяє будувати перехресні запити; пропозиція IN, за допомогою якого задається зв'язок з віддаленою базою даних або вказуються імена стовпців перехресного запиту, та ін. Синтаксис:

[[[] {ім'я-таблиці | ім'я-запиту-на-вибірку | псевдонім}!]].] [[] ім'я-поля []] [[3]](https://stud.com.ua/53372/informatika/stvorennya_zapitiv_vikoristannyam_movi" \l "srcannot_3)

***Ім'я-таблиці, ім'я-запиту-на-вибірку*** і ***псевдонім,*** уточнююче ім'я поля, обов'язково повинні бути присутніми в пропозиції FROM запиту. Якщо таблиця або запит має псевдонім, необхідно використовувати саме його, а не реальне ім'я таблиці або запиту.

Першу частину імені (включаючи точку) можна опустити, якщо поле є тільки в одній з таблиць пропозиції FROM.

Імена, що містять пробіли, обов'язково повинні полягати в квадратні дужки.

При визначенні списку полів використання символу "\*" замість імені поля вказує, що потрібно відобразити всі стовпці даної таблиці. Якщо в якості списку полів використаний символ "\*", то відбираються всі стовпці всіх таблиць, зазначених у пропозиції FROM.

**Приклад 10.5.** Наступний вираз задає нулі з таблиці [Товари], ім'я якого Найменування:

Товари. Найменування

**Пропозиція FROM.** Задає таблиці або запити, службовці джерелами даних для створюваного запиту. Синтаксис:

FROM {ім'я-таблиці **[[AS]** псевдонім] I

ім'я-запиту-на-вибірку [[AS] псевдонім] I

<таблиця-об'єднання>] ...

**[IN** сспеціфікація джерела>]

де <таблиця-об'єднання>:

{ім'я-таблиці **[[AS]** псевдонім] I

ім'я-запиту-на-вибірку **[[AS]** псевдонім] I

(<таблиця-об'єднання>)}

{INNER I LEFT I RIGHT] JOIN

{ім'я-таблиці **[[AS]** псевдонім] I

ім'я-запиту-на-вибірку **[[AS]** псевдонім] I

(<таблиця-об'єднання>)}

**ON** <умову об'єднання>

Для кожної таблиці і запиту можна визначити альтернативне ім'я. Воно використовується як псевдонім замість повного імені таблиці при завданні імен стовпців у списку полів, умови об'єднання і реченні WHERE.

**Приклад 10.6.** Наступний запит відображає всі поля таблиці Товари:

SELECT Товари. \*

FROM Товари; [[4]](https://stud.com.ua/53372/informatika/stvorennya_zapitiv_vikoristannyam_movi" \l "srcannot_4)

**Приклад 10.7.** Наступний запит відображає товари, ціна яких більше 100 дол, і гарантійний термін яких 36 міс:

SELECT Товари.Наіменованіе, Товари.Цена

FROM Товари

WHERE Товари.Цена> 100 AND Товари. [Гарантійний термін] = 36;

Великі можливості мови SQL в чому засновані на його здатності об'єднувати інформацію з декількох таблиць або запитів.

Для завдання типу об'єднання таблиць в логічний набір записів, з якого буде вибиратися необхідна інформація, в пропозиції FROM використовується операція JOIN.

Операція INNER JOIN використовується для отримання всіх рядків з обох логічних таблиць, що задовольняють умові об'єднання.

Операція LEFT JOIN повертає всі рядки з першої логічної таблиці, об'єднані з тими рядками з другої, для яких виконується умова об'єднання.

Аналогічно, операція RIGHT JOIN повертає всі рядки з другої логічної таблиці, об'єднані з тими рядками з першої таблиці, для яких виконується умова об'єднання.

<Умова об'єднання> - вираз, в якому поля першої таблиці порівнюються з полями другої таблиці. У бланку запиту QBE в умови об'єднання використовується тільки оператор одно (=). Запит на основі об'єднання таблиць за нерівністю (<,>, <>, <= або> =) можна створити тільки в режимі SQL.

**Приклад 10.8.** Наступний запит відображає відомості про складах фірми Citii i nk:

SELECT Склади.Номерсклада, Склади.Адрес,

Склади .Телефон

FROM Фірми INNER JOIN Склади

ON Фірми.Кодфірми = Склади.Кодфірми

WHERE Фірми.Названіе = "Citilink";

**Приклад 10.9.** Запит відображає список товарів, які ще не продавалися, із зазначенням найменування товарів і номерів складів, на яких вони зберігаються:

SELECT Товари.Наіменованіе, Храненіе.Номерсклада

FROM (Товари INNER JOIN Зберігання

ON Товари.КодТовара = Храненіе.КодТовара)

LEFT JOIN Продажі ON Храненіе.ID = Продажі.ID

WHERE ((Продажі.Колічество) Is Null);

**Приклад 10.10.** Наступна операція поверне всі рядки з таблиць "Товари" і "Зберігання", для яких збігаються значення в поле КодТовара:

Товари INNER JOIN Зберігання ON Товари.КодТовара =

Храненіе.КодТовара

Операція LEFT JOIN поверне всі рядки з таблиці, отриманої в результаті попередньої операції INNER JOIN, об'єднані з тими рядками з таблиці "Продажі", для яких виконується умова об'єднання. Тоді в с Троках товару, який не продавався, в поле [Кількість проданого товару] буде значення Null, що буде умовою відбору для шуканого списку.

**Пропозиція GROUP BY.** В інструкції SELECT задає стовпці, що використовуються для формування груп з обраних рядків. Синтаксис:

GROUP BY ім'я-стовпця, ...

Оператор GROUP BY розділяє розглянуту таблицю па такі групи, що всередині кожної із цих груп все рядки містять однакові значення у вказаному стовпці.

**Приклад 10.11.** Наступний запит відображає загальна кількість проданого товару кожного найменування:

SELECT Товари.Наіменованіе, Sum (Продажі.Колічество)

AS [Sum-Кількість]

FROM (Товари INNER JOIN Зберігання

ON Товари.КодТовара = Храненіе.КодТовара)

INNER JOIN Продажі ON Храненіе.Ю = Продажі.ID

GROUP BY Товари.Наіменованіе;

**Пропозиція HAVING.** Це спеціальна форма фрази WHERE. Вона відноситься пе до окремих рядків, а до груп: предикат у фразі HAVING завжди посилається (за допомогою спеціальних бібліотечних функцій, таких як, наприклад, SET) на властивості груп, а не рядків, і на основі цього предиката групи цілком або вибираються, або відкидаються . Синтаксис:

HAVING "умова відбору>

У разі відсутності пропозиції GROUP BY умова відбору застосовується до всієї логічної таблиці, визначеної інструкцією SELECT.

**Приклад 10.12.** Наступний запит відображає загальна кількість проданого товару по днях із 10 по 20 грудня 2003 р .:

SELECT Продажі.Дата, ЕІТ (Продажі.Колічество)

AS [Sum-Кількість]

FROM Продажі

GROUP BY Продажі.Дата

HAVING ((Продажі.Дата) Between # 10.12.03 # AND

# 20.12.03 #);

**Пропозиція ORDER BY.** Задає порядок розташування рядків, що повертаються інструкцією SELECT. Синтаксис:

ORDER BY {ім'я-стовпця I номер-стовпця [ASC I DESC]}, ...

Оператор ORDER BY визначає сортування результату вибірки в порядку зростання (ASC) або убування (DESC) значень атрибута. У пропозиції ORDER BY можна вказати декілька стовпців. Список сортується спочатку за значеннями стовпця, ім'я якого зазначено першим.

**Приклад 10.13.** Наступний запит відображає список товарів, упорядкованих за ціною:

SELECT Товари.Наіменованіе, Товари.Цена

FROM Товари

ORDER BY Товари.Цена DESC;

**Інструкція SELECT.** Виконує операції вибору та об'єднання для створення логічної таблиці (набору записів) на базі інших таблиць або запитів. Вище було розглянуто синтаксис інструкції SELECT в загальному вигляді. Розглянемо його більш докладно:

SELECT [ALL I DISTINCT I DISTINCTROW I TOP число

[PERCENT]] <список-полів>

Предикати ALL, DISTINCT, DISTINCTROW, TOP ***n*** або TOP ***n*** PERCENT уточнюють остаточний набір записів запиту.

За замовчуванням діє предикат ALL, при якому в набір записів включаються всі рядки, що задовольняють умовам відбору, у тому числі дублікати.

Предикат DISTINCT вимагає, щоб запит повернув лише рядки, що відрізняються від всіх інших. Якщо інструкція SELECT містить предикат DISTINCTROW, то в набір записів включаються тільки ті рядки, в яких конкатенація первинних ключів з усіх таблиць, що беруть участь у формуванні повертаються стовпців, є унікальною. Залежно від того, які стовпці представлені в наборі записів, іноді можна побачити повторювані рядки, але навіть у цьому випадку кожна рядок запиту утворена з унікальної (DISTINCT) комбінації рядків (ROWS) базових таблиць.

Щоб результуючий набір містив тільки перші ***п*** або перші ***п*** відсотків записів, використовуйте предикат ТОР ***п*** або TOP ***п*** PERCENT. Параметр ***п*** повинен бути цілим числом, що не перевищує 100, якщо використовується ключове слово PERCENT.

**Приклад 10.14.** Наступний запит відображає список 10 найдорожчих товарів:

SELECT ТОР 10 Товари.Наіменованіе, Товари.Цена

FROM Товари

ORDER BY Товари.Цена DESC;

**Підлеглий запит (вкладена вибірка).** Мова SQL дозволяє використовувати в умові відбору результати інший вибірки. Рівнів вкладеності може бути декілька.

**Приклад 10.15.** Наступний запит відображає список процесорів, ціна яких нижча середньої ціни процесорів:

SELECT Товари.Наіменованіе, Товари.Цена

FROM Товари

WHERE ((Товари.Наіменованіе) Like "проц \*") AND

((Товари.L (eHa) <(SELECT Ауд (Товари.Цена) AS СредняяЦена

FROM Товари

WHERE ((Товари.Наіменованіе) Like "проц \*")));

У наведеному прикладі внутрішній підлеглий запит обчислює середню ціну процесорів, зовнішній запит відбирає процесори, ціна яких нижча середньої ціни процесорів