Лабораторна робота № 1

Тема: Отримання даних із таблиць. Сортировка.

Мета: Здобути навички з отримання даних с таблиця за допомогою SQL-запитів.

Інструкція SELECT

Інструкції SQL складаються із звичайних англійських термінів. Ці терміни називаються *ключовими словами*, і кожна інструкція SQL містить одне або кілька ключових слів. Найчастіше ви стикатиметеся з інструкцією SELECT, яка призначена для отримання інформації з однієї або декількох таблиць.

Ключове слово
Зарезервоване слово є частиною SQL. Ніколи не називайте таблицю чи стовпець таким словом.

Щоб за допомогою інструкції SELECT витягти дані з та потрібно вказати як мінімум дві речі: що саме ви хочете витягти і звідки.

Запит

```
SELECT name
FROM products;
```

Аналіз

У цьому прикладі інструкція SELECT витягує один стовпець під назвою name із таблиці products. Ім'я стовпця вказується відразу після ключового слова SELECT, а ключове слово FROM вказує на ім'я таблиці, з якої витягуються дані. Результат виконання інструкції буде таким.

Результат

```
products
- - - - - - - - - - - -
Fish bean bag toy
Bird bean bag toy
Rabbit bean bag toy
8 inch teddy bear
12 inch teddy bear
18 inch teddy bear
```

Якщо ви спробуєте виконати цей запит самостійно, то можете побачити, що дані часто відображаються в іншому порядку. Якщо результати запиту не відсортовані явно, то дані повертатимуться у довільному порядку. Це може бути порядок, у якому рядки заносилися в таблицю, або якийсь інший порядок. Головне щоб запит повертав те саме число рядків.

Для отримання кількох стовпців з таблиці застосовується та сама інструкція SELECT. Відмінність полягає в тому, що після ключового слова SELECT необхідно через кому вказати кілька імен стовпців.

Запит

```
SELECT id, name, price
FROM products;
```

Аналіз

Як і в попередньому прикладі, тут для отримання даних з таблиці products застосовується інструкція SELECT. У цьому випадку перераховані три імені стовпця, розділені комами.

Крім вилучення зазначених стовпців (одного чи кількох), за допомогою інструкції SELECT можна запросити всі стовпці, не перозраховуючи кожен із них. Для цього замість імен стовпців вказується груповий символ "зірочка" (*). Це робиться в такий спосіб.

Запит

```
SELECT *
FROM products;
```

Аналіз

За наявності групового символу (*) повертаються усі стовпці. Зазвичай (але не завжди) стовпці повертаються в тому порядку, якому вони зазначені у визначенні таблиці. Втім, табличні дані рідко виводяться у вигляді, у якому зберігаються у базі дані

Отримання унікальних рядків. Як ви переконалися, інструкція SELECT повертає всі рядки, які відповідають критерію відбору. Але якщо вам не потрібні абсолютно всі значення? Припустимо, наприклад, що необхідно дізнатися про ідентифікатори всіх постачальників з таблиці products.

Запит

```
SELECT vendor
FROM products;
```

Результат

```
vendor
- - - - - - - - - - - -
BRS01
BRS01
BRS01
DLL01
DLL01
DLL01
DLL01
FNG01
FNG01
```

Інструкція SELECT повернула 9 рядків (хоча у списку всього три постачальники), тому що в таблиці Products вказано 9 товарів. Як отримати список унікальних значень?

Рішення полягає у застосуванні ключового слова DISTINCT, яке, як неважко припустити, змушує СУБД повернути лише унікальні значення.

Запит

```
SELECT DISTINCT vendor
FROM products;
```

Результат

```
vendor
- - - - - - - - - - -
BRS01
DLL01
FNG01
```

Обмеження результатів запиту

Інструкція SELECT повертає всі рядки, що відповідають критерієм відбору. Найчастіше це всі рядки таблиці. Але що якщо необхідно отримати лише перший рядок чи задану кількість рядків? Таке цілком можливо, проте доводиться з жалем констатувати, що це одна з тих рідкісних ситуацій, коли різні реалізації SQL поводяться по-різному.

У Microsoft SQL Server і Microsoft Access можна скористатися ключовим словом TOP, щоб отримати лише кілька перших записів, як показано нижче.

Запит

```
SELECT TOP 5 name
FROM products;
```

В MySQL, MariaDB, PostgreSQL и SQLite можно воспользоваться предложением LIMIT .

Запит

```
SELECT name
FROM products
LIMIT 5;
```

Аналіз

У цій інструкції вилучається єдиний стовпець. Інструкція LIMIT 5 змушує СУБД повернути не більше п'яти рядків. Якщо потрібно одержати наступні п'ять рядків, задайте початкову точку вилучення та потрібну кількість рядків, як показано нижче.

Запит

```
SELECT name
FROM products
LIMIT 5 OFFSET 5;
```

Аналіз

Інструкція LIMIT 5 OFFSET 5 змушує СУБД повернути п'ять рядків, починаючи з рядка 5. Перше число - це кількість рядків для отримання, а друге - початкова точка.

Сортування отриманих даних

Для сортування даних, що витягуються інструкцією SELECT, призначена пропозиція ORDER BY. У ньому вказується ім'я одного або кількох стовпців, за якими сортуються результати запиту. Розглянемо наступний приклад.

Запит

```
SELECT *
FROM products
ORDER BY name ASC;
```

Інструкція ORDER BY змушує СУБД відсортувати дані в алфавітному порядку за стовпцем name.

Щоб виконати сортування кількома стовпцями, вкажіть їх імена через кому. У наступному прикладі витягуються три стовпці, а результат сортується за двома: спочатку — за ціною, потім — за назвою.

Запит

```
SELECT id, name, price
FROM products
ORDER BY price, name;
```

Результат

id	price	name
BNBG02	3.49	Bird bean bag toy
BNBG01	3.49	Fish bean bag toy
BNBG03	3.49	Rabbit bean bag toy
RGAN01	4.99	Raggedy Ann
BR01	5.99	8 inch teddy bear
BR02	8.99	12 inch teddy bear
RYL01	9.49	King doll
RYL02	9.49	Queen doll
BR03	11.99	18 inch teddy bear

Вказівка напрямку сортування

Дані можна сортувати не лише за зростанням (від А до Я). Такий порядок заданий за замовчуванням, але в пропозиції ORDER BY також можна вказувати порядок за спаданням (від Я до А). Для цього призначено ключове слово DESC.

Запит

```
SELECT id, name, price
FROM products
ORDER BY price DESC;
```

Результат

id	price	name
BR03	11.99	18 inch teddy bear
RYL01	9.49	King doll
RYL02	9.49	Queen doll
BR02	8.99	12 inch teddy bear
BR01	5.99	8 inch teddy bear
RGAN01	4.99	Raggedy Ann
BNBG02	3.49	Bird bean bag toy
BNBG01	3.49	Fish bean bag toy
BNBG03	3.49	Rabbit bean bag toy

Але як бути у разі сортування кількома стовпцями? Ключове слово DESC застосовується тільки до того стовпця, після якого воно вказано.

Запит

```
SELECT id, name, price
FROM products
ORDER BY price DESC, name ASC;
```

ЗАВДАННЯ:

Для виконання роботи необхідно імпортувати таблицю cities.

Необхідно написати наступні запити:

1. Написати запит на отримання другої десятки міст України за кількістю населення
2. Отримати список міст відсортованим за назвою в зворотньому порядку (від Я до А). Обмежити запит 30 запасами.
3. Отримати список міст відсортованих за регіоном та кількістю населення (спочатку найнаселеніші міста регіону).
4. Отримати список регіонів вказаних в таблиці.
5. Отримати список міст відсортованих за регіоном у зворотньому порядку та за назвою у межах регіону також у зворотньому порядку

Усі запити мають бути збережені у файл в форматі .sql

Таблиця cities

```
SET NAMES utf8;
SET time_zone = '+00:00';
SET foreign_key_checks = 0;
SET sql_mode = 'NO_AUTO_VALUE_ON_ZERO';

DROP TABLE IF EXISTS `cities`;
CREATE TABLE `cities` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  `population` int(10) unsigned DEFAULT NULL,
  `region` varchar(5) COLLATE utf8_unicode_ci DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

INSERT INTO `cities` (`id`, `name`, `population`, `region`) VALUES
(1,     'Київ',2888470,    'N'),
(2,     'Харків',      1444540,    'E'),
(3,     'Одеса',       1010000,    'S'),
(4,     'Дніпро',      984423,     'C'),
(5,     'Донецьк',  932562,     'E'),
(6,     'Запоріжжя',       758011,     'E'),
(7,     'Львів',       728545,     'W'),
(8,     'Кривий Ріг',      646748,     'S'),
(9,     'Миколаїв',494381,     'S'),
(10,    'Маріуполь',       458533,     'S'),
(11,    'Луганськ', 417990,     'E'),
```

(12, 'Севастополь', 412630, 'S'),
(13, 'Вінниця', 372432, 'W'),
(14, 'Макіївка', 348173, 'E'),
(15, 'Сімферополь', 332608, 'S'),
(16, 'Херсон', 296161, 'S'),
(17, 'Полтава', 294695, 'E'),
(18, 'Чернігів', 294522, 'N'),
(19, 'Черкаси', 284459, 'C'),
(20, 'Суми', 268409, 'E'),
(21, 'Житомир', 268000, 'N'),
(22, 'Хмельницький', 267891, 'W'),
(23, 'Чернівці', 264427, 'W'),
(24, 'Горлівка', 250991, 'E'),
(25, 'Рівне', 249477, 'W'),
(26, 'Кам\'янське', 240477, 'C'),
(27, 'Кропивницький', 232052, 'C'),
(28, 'Івано-Франківськ', 229447, 'W'),
(29, 'Кременчук', 224997, 'C'),
(30, 'Тернопіль', 217950, 'W'),
(31, 'Луцьк', 217082, 'W'),
(32, 'Біла Церква', 211080, 'N'),
(33, 'Краматорськ', 160895, 'E'),
(34, 'Мелітополь', 156719, 'S'),
(35, 'Керч', 147668, 'S'),
(36, 'Сєвєродонецьк', 130000, 'E'),
(37, 'Хрустальний', 124000, 'E'),
(38, 'Нікополь', 119627, 'C'),
(39, 'Бердянськ', 115476, 'S'),

```
(40, 'Слов\'янськ', 115421, 'E'),
(41, 'Ужгород', 115195, 'W'),
(42, 'Алчевськ', 111360, 'E'),
(43, 'Павлоград', 110144, 'E'),
(44, 'Євпаторія', 106115, 'S'),
(45, 'Лисичанськ', 103459, 'E'),
(46, 'Кам\'янець-Подільський', 101590, 'W'),
(47, 'Бровари', 100374, 'N'),
(48, 'Дрогобич', 98015, 'W'),
(49, 'Кадіївка', 92132, 'E'),
(50, 'Конотоп', 92000, 'E');
```