

ЛАБОРАТОРНА РОБОТА №6
Розробка запитів і селекторів для тематичних баз даних

Мета: закріпити отримані навички із розробки SQL-запитів.

Програмне забезпечення: СКБД на вибір студента.

ПРАКТИЧНЕ ЗАВДАННЯ

1) Відповідно до розробленої вами БД в ЛР№2 розробити систему запитів для додавання даних.

Запити для додавання даних повинні забезпечити наповнення таблиць бази даних щонайменше 15 записами кожену (за виключенням довідникових таблиць, де кількість записів логічно обмежена).

2) Відповідно до розробленої вами БД в ЛР№2 розробити систему запитів для видалення даних.

Запити для видалення даних з таблиць необхідно розробити для вирішення типових завдань:

- 2.1. видалення всіх даних з таблиці (розробити для кожної таблиці, у кожного студента за умовою ЛР№1 повинно було бути не менше 6-и сутностей, а отже і таблиць);
- 2.2. видалення даних з таблиць згідно з певними умовами (наприклад, видалити записи, у яких значення поля Дата менше за певне, або Ціна менша середньої тощо) – таких запитів необхідно розробити не менше по одному для кожної таблиці (тобто не менше 6-и), умови на розсуд студента.

3) Відповідно до розробленої вами БД в ЛР№2 розробити систему запитів для пошуку даних.

Запити для пошуку даних повинні дозволяти формувати вибірки даних для типових задач предметної області (Додаток Г). Необхідно розробити не менше 3-х запитів для кожної з наведених категорій (тобто всього не менше 9-и):

- 3.1. звичайні запити (вибірка даних з обмеженнями на значення деяких стовпців);
- 3.2. вкладені запити (вибірка даних з обмеженнями, які формуються як результат вкладеного запиту до цієї або іншої таблиці);
- 3.3. запити з обмеженням на результати підсумкових функцій (наприклад, на мінімальне, максимальне, середнє тощо).

ПРИМІТКА: в додатку Г наведено декілька умов для розробки запитів пункту 3. Студент їх повинен реалізувати, визначивши, яку категорію (3.1-3.3) наведена умова покриває. Решта умов запитів – на розсуд студента.

4) При розробці системи запитів пунктів 1-3 навести використання:

- умова WHERE.
- пропозиція DISTINCT.
- реляційні операції (=, >, <, >=, <=, <>).
- булеві операції (AND, OR, NOT).
- оператори IN, BETWEEN, LIKE.

- агрегатні функції (COUNT, SUM, AVG, MAX, MIN) з одиночними полями або з аргументами, які складаються з скалярних виразів, що включають більше одного поля.
- пропозиції GROUP BY (з одним або з декількома полями) та HAVING.
- упорядкування виводу полів ORDER BY за одним або за декількома полями.
- упорядкування агрегатних груп (поєднання ORDER BY та GROUP BY).
- оператор LIMIT з одним або з двома параметрами.
- у підзапитах операторів IN, ALL, ANY, EXISTS.
- оператор UNION та з'єднання даних з різних таблиць (Inner Join, Outer Join).

ПРИМІТКА: Приклади встановлення обмеження в таблицях.

- Використання обмежень для виключення порожніх (NULL) значень

```
CREATE TABLE Salespeople  
(snum integer NOT NULL,  
sname char (10) NOT NULL,  
city char (10),  
comm decimal);
```

- Унікальність, як обмеження стовпця

```
CREATE TABLE Salespeople  
(snum integer NOT NULL UNIQUE,  
sname char (10) NOT NULL UNIQUE,  
city char (10),  
comm decimal);
```

- Унікальність, як обмеження таблиці

```
CREATE TABLE Customers  
(cnum integer NOT NULL,  
cname char (10) NOT NULL,  
city char (10),  
rating integer,  
snum integer NOT NULL,  
UNIQUE (cnum, snum));
```

- Перевірка значень полів (CHECK)

```
CREATE TABLE Salespeople  
(snum integer NOT NULL PRIMARY KEY,  
sname char(10) NOT NULL UNIQUE,  
city char(10),  
comm decimal CHECK (comm < 1));
```

- Перевірка умов, які базуються на декількох полях

```
CREATE TABLE Salespeople  
(snum integer NOT NULL UNIQUE,  
sname char (10) NOT NULL UNIQUE,  
city char (10),  
comm decimal  
CHECK (comm < 0.15 OR city = 'Barcelona'));
```

- Установка значень за замовчуванням

```
CREATE TABLE Salespeople  
(snum integer NOT NULL UNIQUE,  
sname char(10) NOT NULL UNIQUE,  
city char(10) DEFAULT = 'New York',  
comm decimal CHECK (comm < 1));
```