**Міністерство освіти і науки України**

**Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”**

**Факультет інформатики та обчислювальної техніки Кафедра інформатики та програмної інженерії**

Звіт

з лабораторної роботи №4 з дисципліни

“Бази даних”

**“Побудова простих запитів”**

Варіант 32

ВиконалаІП-21 Скрипець Ольга Олександрівна

ПеревірилаМарченко О. І.

**Київ 2023**

**ЛАБОРАТОРНА РОБОТА №4**

**Побудова простих запитів**

**Мета роботи:**

* Вивчити оператор, котрий використовується в реляційних СУБД, для вибірки даних з таблиць, групування та сортування даних
* Навчитись використовувати вбудовані функції в запитах
* Вивчити призначення представлень (view) баз даних, синтаксису та семантики команд SQL для їх створення, зміни та видалення, системних збережених процедур для отримання інформації про представлення
* Навчитись створювати запити згідно їх словесного опису

**Постановка задачі лабораторної роботи № 4**

При виконанні лабораторної роботи необхідно виконати наступні дії: 1) Створити наступні запити:

a. запит з використанням функції COUNT;

b. запит з використанням функції SUM;

c. запит з використанням групування по декільком стовпцям;

d. запит з використанням умови відбору груп HAVING;

e. запит з використанням HAVING без GROUP BY;

f. запит з використанням функцій row\_number() over ….;

g. запит з використанням сортування по декільком стовпцям;

h. запити згідно варіанту завдання.

2) Робота з представленнями (view):

a. створити представлення, котре містить дані з декількох таблиць;

b. створити представлення, котре містить дані з декількох таблиць та використовує представлення, котре створене в п.a;

c. модифікувати представлення з використанням команди ALTER VIEW;

d. отримати довідникову інформацію про ці представлення з використанням вбудованих процедур (наприклад в MsSQL sp\_help, sp\_helptext).

3) Для кожного з запитів п.1 та п.2 навести їх словесний (сутнісний) опис.

**Варіант 32:**

**32) “** Тролейбусне депо **“**

Програмне забезпечення «Тролейбусне депо» . База даних містить інформацію: відомості про водіїв (табельний номер; ПІБ ; дата , час початку і закінчення роботи на маршруті), відомості про тролейбуси (номер ; марка ; кількість пасажирських місць; дата останнього технічного огляду), відомості про маршрути (номер ; час початку і кінця роботи тролейбусів ; початковий пункт ; кінцевий пункт ; список зупинок ; тривалість маршруту). Кожен водій працює на одному тролейбусі і на одному маршруті . На одному маршруті працює кілька водіїв. Кількість пасажирських місць залежить тільки від марки тролейбуса.

**Запити:**

а) Визначить маршрути, на яких працює найбільша кількість тролейбусів.

б) Визначить зупинки, на яких зупиняється найбільша кількість тролейбусних маршрутів.

**Виконання:**

1) **Створити наступні запити:**

**a. запит з використанням функції COUNT;**

-- Підрахунок кількості автобусних зупинок для кожного маршруту

SELECT route\_number, COUNT(bus\_stopID) AS num\_bus\_stops

FROM bus\_stop\_list

GROUP BY route\_number;

**b. запит з використанням функції SUM;**

-- Підрахунок загальної кількості пасажирів на всіх тролейбусах

SELECT SUM(number\_of\_passengers) AS total\_passengers

FROM trolleybus;

**c. запит з використанням групування по декільком стовпцям;**

-- Підрахунок кількості тролейбусів для кожної марки та країни виробника

SELECT car\_brand\_name, producing\_country, COUNT(troll\_number) AS num\_trolleybuses

FROM trolleybus

JOIN car\_brand ON trolleybus.car\_brandID = car\_brand.car\_brandID

GROUP BY car\_brand\_name, producing\_country;

**d. запит з використанням умови відбору груп HAVING;**

-- Пошук маршрутів, де кількість зупинок більше 2

SELECT route\_number, COUNT(bus\_stopID) AS num\_bus\_stops

FROM bus\_stop\_list

GROUP BY route\_number

HAVING num\_bus\_stops > 1;

**e. запит з використанням HAVING без GROUP BY;**

-- Пошук тролейбусів, які мають більше 30 пасажирів та доступ до Wi-Fi

SELECT troll\_number

FROM trolleybus

WHERE number\_of\_passengers > 30 AND wifi\_available = TRUE;

**f. запит з використанням функцій row\_number() over ….;**

-- Нумерація рядків для водіїв з сортуванням за прізвищем у зростаючому порядку

SELECT \*, ROW\_NUMBER() OVER (ORDER BY last\_name) AS row\_num

FROM driver;

**g. запит з використанням сортування по декільком стовпцям;**

-- Сортування тролейбусів за кількістю пасажирів у спадаючому порядку, а потім за датою останньої інспекції

SELECT troll\_number, number\_of\_passengers, date\_last\_tech\_insp

FROM trolleybus

ORDER BY number\_of\_passengers DESC, date\_last\_tech\_insp;

**h. запити згідно варіанту завдання.**

-- Знаходження маршрутів з найбільшою кількістю тролейбусів

SELECT route\_number, COUNT(troll\_number) AS num\_trolleybuses

FROM driver

GROUP BY route\_number

ORDER BY num\_trolleybuses DESC

LIMIT 1;

-- Знаходження зупиноки з найбільшою кількістю тролейбусних маршрутів

SELECT bs.bus\_stopID, bs.bus\_stop\_name, COUNT(DISTINCT bsl.route\_number) AS num\_routes

FROM bus\_stop\_list bsl

JOIN bus\_stop bs ON bsl.bus\_stopID = bs.bus\_stopID

GROUP BY bs.bus\_stopID, bs.bus\_stop\_name

ORDER BY num\_routes DESC

LIMIT 1;

**2) Робота з представленнями (view):**

**a. створити представлення, котре містить дані з декількох таблиць;**

CREATE VIEW combined\_data AS

SELECT d.DriverID, d.last\_name, d.first\_name, d.troll\_number, r.route\_number

FROM driver d

JOIN route r ON d.route\_number = r.route\_number;

**b. створити представлення, котре містить дані з декількох таблиць та використовує представлення, котре створене в п.a;**

-- Створення представлення, що містить дані з декількох таблиць та використовує інше представлення

CREATE VIEW extended\_combined\_data AS

SELECT cd.\*, bs.bus\_stop\_name

FROM combined\_data cd

JOIN bus\_stop\_list bsl ON cd.route\_number = bsl.route\_number

JOIN bus\_stop bs ON bsl.bus\_stopID = bs.bus\_stopID;

**c. модифікувати представлення з використанням команди ALTER VIEW;**

-- Модифікація представлення, додавання нового стовпця

ALTER VIEW combined\_data AS

SELECT d.DriverID, d.last\_name, d.first\_name, d.troll\_number, r.route\_number, s.weekday

FROM driver d

JOIN route r ON d.route\_number = r.route\_number

JOIN schedule s ON d.scheduleID = s.scheduleID;

**d. отримати довідникову інформацію про ці представлення з використанням вбудованих процедур (наприклад в MsSQL sp\_help, sp\_helptext).**

-- Використання вбудованої процедури для отримання інформації про представлення

SHOW CREATE VIEW combined\_data;

SHOW CREATE VIEW extended\_combined\_data;

DESCRIBE combined\_data;

DESCRIBE extended\_combined\_data;

**ВИСНОВОК**

У ході виконання лабораторної роботи були створені та опрацьовані прості SQL-запити для отримання інформації з бази даних "Тролейбусне депо". Кожен запит включав в себе використання різних функцій та операцій для забезпечення необхідного аналізу даних. Було використано функції COUNT та SUM для підрахунку кількості записів та суми значень відповідно. Використано групування для агрегації даних по певних критеріях. Створено представлення, яке об'єднує дані з декількох таблиць. Створено ще одне представлення, яке використовує вже створене представлення, доповнюючи його додатковими даними. Модифіковано одне з представлень з використанням команди ALTER VIEW.

Лабораторна робота дозволила ознайомитися з основними функціями SQL для вибірки, групування та обробки даних у реляційних базах даних. Робота з представленнями (view) надає можливість абстрагуватися від складної структури бази даних і отримувати необхідні дані за допомогою зручних іменованих "віртуальних таблиць". Використання вбудованих функцій та операцій SQL дозволяє ефективно обробляти та аналізувати дані в базах даних.