

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»

«Створення запитів на групування, сортування, використання вбудованих
функцій. Створення та керування представленнями.»

Варіант 1

Виконав студент ІП-13 Ал Хадам Мурат Резгович

(шифр, прізвище, ім'я, по батькові)

Перевірів Марченко Олена Іванівна

(прізвище, ім'я, по батькові)

Лабораторна робота №4

Варіант 1

Тема: Створення запитів на групування, сортування, використання вбудованих функцій. Створення та керування представленнями.

Мета:

- 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

Постановка задачі

При виконанні лабораторної роботи необхідно виконати наступні дії:

1) Створити наступні запити:

- а. запит з використанням функції COUNT;
- б. запит з використанням функції SUM;
- в. запит з використанням функцій UPPER, LOWER;
- г. запит з використанням функцій для роботи з датами;
- д. запит з використанням групування по декільком стовпцям;
- е. запит з використанням умови відбору груп HAVING;
- ж. запит з використанням HAVING без GROUP BY;
- з. запит з використанням функцій row_number() over;
- и. запит з використанням сортування по декільком стовпцям.

2) Робота з представленнями (view):

- а. створити представлення, котре містить дані з декількох таблиць;
- б. створити представлення, котре містить дані з декількох таблиць та використовує представлення, котре створене в п.а;
- в. модифікувати представлення з використанням команди ALTER VIEW;
- г. отримати довідникову інформацію про ці представлення з використанням вбудованих процедур (наприклад в MsSQL sp_help, sp_helptext та sp_depends).

Виконання завдання

Створені запити:

task1.sql

```
use `db-labs`;
# a
SELECT COUNT(*)
FROM patients
WHERE gender = 'чоловік';
# b
SELECT SUM(sum_to_pay)
FROM conclusions;
# c
SELECT UPPER(last_name)
FROM patients;

SELECT LOWER(first_name)
FROM admins;
# d
SELECT CONCAT(first_name, ' ', last_name) AS full_name,
       DATE_FORMAT(FROM_DAYS(DATEDIFF(NOW(), birthday)), '%Y') + 0 AS age
FROM patients
ORDER BY age;
# e
SELECT COUNT(shift_number), shift_number
FROM admins
GROUP BY shift_number;
# f
SELECT COUNT(patient_id) AS patient_amount,
       CONCAT(d.first_name, ' ', d.last_name) AS doctor_name
FROM applications
       JOIN doctors d on d.doctor_id = applications.doctor_id
GROUP BY doctor_name
HAVING patient_amount > 1;
# g
SELECT AVG(sum_to_pay)
FROM conclusions
HAVING AVG(sum_to_pay) > 1;
# h
SELECT ROW_NUMBER() OVER (ORDER BY c.height DESC) AS position,
       CONCAT(p.first_name, ' ', p.last_name) AS full_name,
       c.height
FROM patients p
       JOIN cards c on p.patient_id = c.patient_id;
# i
SELECT CONCAT(p.first_name, ' ', p.last_name) AS full_name,
       c.weight
FROM patients p
       JOIN cards c on p.patient_id = c.patient_id
ORDER BY c.weight DESC;
```

task2.sql

```
USE `db-labs`;

# a
CREATE OR REPLACE VIEW patient_info AS
SELECT p.patient_id,
       CONCAT(p.first_name, ' ', p.last_name) AS full_name,
       c.height,
       c.weight,
       c.medical_group
FROM patients AS p
      JOIN cards c on p.patient_id = c.patient_id
ORDER BY p.patient_id;

SELECT *
FROM patient_info;

# b
CREATE OR REPLACE VIEW patient_doctor AS
SELECT p.patient_id,
       CONCAT(p.first_name, ' ', p.last_name) AS patient_name,
       pi.medical_group,
       CONCAT(d.first_name, ' ', d.last_name) AS doctor_name,
       possible_diagnosis,
       d.specialization
FROM patients p
      JOIN applications a on p.patient_id = a.patient_id
      JOIN doctors d on d.doctor_id = a.doctor_id
      JOIN patient_info pi on pi.patient_id = a.patient_id
ORDER BY doctor_name;

SELECT *
FROM patient_doctor;

# c
ALTER VIEW patient_info AS
SELECT p.patient_id,
       CONCAT(p.first_name, ' ', p.last_name) AS full_name,
       c.height,
       c.weight,
       c.medical_group,
       ROUND(c.weight / POW((c.height / 100), 2), 2) AS bmi
FROM patients AS p
      JOIN cards c on p.patient_id = c.patient_id
ORDER BY p.patient_id;

SELECT *
FROM patient_info;

# d
/*
MySQL command to view the definition of a view or table,
similar to sp_helptext in Microsoft SQL Server
*/
SHOW CREATE VIEW patient_info;

SELECT *
FROM information_schema.TABLES
WHERE TABLE_TYPE = 'VIEW'
      AND TABLE_SCHEMA = 'db-labs';
```

Висновок:

На даній лабораторній роботі я попрактикувався у використанні функцій для групування та сортування даних в запитах. Також, навчився працювати з представленнями, їх створенням та модифікацією.