

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 комп'ютерного практикуму № 4 з дисципліни
«Бази даних»

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

Виконала

IT-01 Дмитрієва Ірина
(шифр, прізвище, ім'я, по батькові)

Перевірила

Марченко О.І.
(прізвище, ім'я, по батькові)

Київ 2021

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.

Теоретичні основи:

Викладені в лекційному матеріалі.

Постановка задачі комп'ютерного практикуму № 4:

При виконанні комп'ютерного практикуму необхідно виконати наступні дії:

1) Створити наступні запити:

- a. запит з використанням функції COUNT;
- b. запит з використанням функції SUM;
- c. запит з використанням функцій UPPER, LOWER;
- d. запит з використанням функцій для роботи з датами;
- e. запит з використанням групування по одному стовпцю;
- f. запит з використанням групування по декільком стовпцям;
- g. запит з використанням умови відбору груп HAVING;
- h. запит з використанням HAVING без GROUP BY;
- i. запит з використанням сортування по стовпцю.

2) Робота з представленнями (view):

- a. створити представлення, котре містить дані з декількох таблиць;
- b. створити представлення, котре містить дані з декількох таблиць та посилання, котре створене в п.а;
- c. модифікувати запити з використанням команди ALTER VIEW;

d. отримати довідникову інформацію про ці представлення з використанням вбудованих процедур (наприклад в MsSQL `sp_help`, `sp_helptext` и `sp_depends`).

Варіант індивідуального завдання – 7:

7) Програмне забезпечення готелю.

Основна задача програмного забезпечення, котре проєктується - є відстеження фінансової сторони роботи готелю. Діяльність організована наступним чином: готель надає номери клієнтам на певний термін. Кожен номер характеризується місткістю, комфортністю (люкс, напівлюкс, звичайний , тощо) і ціною. Клієнтами є різні особи,, по яким збирається певна інформація (прізвище, ім'я, по-батькові, номер та серія паспорту або іншого документу, що посвідчує особу, додаткова інформація). Здача номера клієнтові проводиться за наявності вільних місць в номерах, за параметрами, котрі вказав клієнт. При поселенні фіксується дата поселення, при виїзді - дата звільнення. номеру. Необхідно не лише зберігати інформацію за фактом здачі номера клієнтові, але і здійснювати бронювання номерів. Крім того, для постійних клієнтів, а також для певних категорій клієнтів передбачена система знижок. Знижки можуть підсумовуватися.

1 частина завдання:

-- ----- кількість номерів по комфортабельності -----

```
SELECT room.comfort AS "Комфортабельність",  
COUNT(*) AS "Кількість кімнат", ROUND(AVG(room.default_price))  
AS "Середня вартість"  
FROM hotel.room GROUP BY room.comfort;
```

-- ----- сума до сплати усіх гостей готелю -----

```
SELECT COUNT(*) AS "Кількість гостей", SUM(r.default_price -  
cs.discount - ct.discount) AS "До сплати"  
FROM hotel.client AS cl  
RIGHT JOIN hotel.booking AS b  
ON cl.id = b.client_id  
LEFT JOIN hotel.room AS r  
ON r.id = b.room_id  
LEFT JOIN client_status AS cs  
ON cs.id = cl.client_status_id  
LEFT JOIN category_type AS ct  
ON ct.id = cl.category_type_id WHERE b.status != "Canceled"  
ORDER BY r.number;
```

-- ----- список заселених гостей разом зі статусом їхнього проживання -----

```
SELECT UPPER(cl.first_name) AS "Ім'я", LOWER(cl.last_name) AS  
"Прізвище", r.number AS "№:", b.status AS "Статус бронювання:"  
FROM hotel.client AS cl  
RIGHT JOIN hotel.booking AS b  
ON cl.id = b.client_id  
LEFT JOIN hotel.room AS r  
ON r.id = b.room_id WHERE b.status != "Canceled" ORDER BY  
r.number;
```

-- ----- бронювання на певну дату -----

```
SELECT booking.`check-in_date` AS "Дата чек-іну",  
booking.`check-out_date` AS "Дата чек-ауту",  
booking.status AS "Статус", booking.arrival_date AS "Дата  
заїзду", booking.departure_date AS "Дата виїзду" FROM  
hotel.booking  
WHERE DATE(departure_date) = "2021-03-25";
```

-- ----- список бронювань певного діапазону років -----

```
SELECT booking.`check-in_date` AS "Дата чек-іну",  
booking.`check-out_date` AS "Дата чек-ауту", booking.status AS  
"Статус" FROM hotel.booking  
WHERE YEAR(booking.`check-in_date`) >= 2015 AND  
YEAR(booking.`check-in_date`) <= 2025;
```

-- ----- кількість претендентів на знижку постійного користувача -----

```
SELECT ct.type AS "Категорія", COUNT(*) AS "Кількість  
користувачів"  
FROM category_type AS ct  
RIGHT JOIN hotel.client AS cl  
ON ct.id = cl.category_type_id GROUP BY cl.category_type_id;
```

-- ----- список номерів "не для бідних" -----

```
SELECT room.number AS "№", room.comfort AS "Комфортабельність",  
room.capacity AS "Місткість", room.default_price AS "Вартість"  
FROM hotel.room WHERE room.comfort NOT LIKE "Common" ORDER BY  
room.default_price;
```

-- --- кількість претендентів на знижку згідно до студентського статусу -----

```
SELECT cs.name AS "Статус", COUNT(*) AS "Кількість користувачів"
```

```
FROM client_status AS cs
RIGHT JOIN hotel.client AS cl
ON cs.id = cl.client_status_id GROUP BY cl.client_status_id
HAVING COUNT(*) > 2;
```

-- ----- знайти готельні номери з початковою ціною, більшою за 200 -----

```
SELECT * FROM hotel.room HAVING room.default_price > 200;
```

2 частина завдання

-- ----- кількість гостей, що претендують на певну знижку -----

```
CREATE VIEW view1 AS
SELECT cl.first_name AS "Ім'я", cl.last_name AS "Прізвище",
cs.name AS "Статус клієнта", ct.type AS "Категорія користувача"
FROM hotel.client AS cl
LEFT JOIN client_status AS cs
ON cs.id = cl.client_status_id
LEFT JOIN category_type AS ct
ON ct.id = cl.category_type_id;
```

```
CREATE OR REPLACE VIEW view1 AS
SELECT cl.first_name AS "Ім'я", cl.last_name AS "Прізвище",
cl.document_type AS "Тип документа", concat(cl.document_series,
' ', cl.document_number) AS "Документ", r.number AS "№",
r.comfort AS "Комфортність"
FROM hotel.client AS cl
RIGHT JOIN hotel.booking AS b
ON cl.id = b.client_id
LEFT JOIN hotel.room AS r
ON r.id = b.room_id WHERE r.number IS NOT NULL;
```

```
ALTER VIEW view1 AS
SELECT cl.first_name AS "Ім'я", cl.last_name AS "Прізвище",
cl.document_type AS "Тип документа", concat(cl.document_series,
' ', cl.document_number) AS "Документ", r.number AS "№",
r.comfort AS "Комфортність"
FROM hotel.client AS cl
RIGHT JOIN hotel.booking AS b
ON cl.id = b.client_id
LEFT JOIN hotel.room AS r
ON r.id = b.room_id WHERE r.number IS NOT NULL ORDER BY
r.number;
```

```
SELECT * FROM view1;
```

```
DROP VIEW IF EXISTS view1;
```

```
SHOW FULL TABLES
WHERE table_type = 'VIEW';
```

```
SHOW FULL TABLES IN hotel
WHERE table_type = 'VIEW';
```

```
SHOW FULL TABLES IN sys
WHERE table_type='VIEW';
```

Висновок:

при виконанні даного комп'ютерного практикуму був вивчений оператор, котрий використовується в реляційних СУБД, для вибірки даних з таблиць, групування та сортування даних; було опановано використання вбудованих функцій у запитах; було вивчено призначення представлень (view) баз даних, синтаксису та семантики команд SQL для їх створення, зміни та видалення, системних збережених процедур для отримання інформації про представлення.