

**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(ла)**      ІП-35 Адаменко Арсен Богданович      \_\_\_\_\_

**Перевірів**      Марченко Олена Іванівна      \_\_\_\_\_

Київ 2024

## Лабораторна робота №4

### 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 лабораторної роботи №4

При виконанні лабораторної роботи необхідно виконати наступні дії:

- 1) Створити наступні запити (в запитах повинні використовуватись 2 та більше таблиць):
  - 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, в котрому значення одного зі стовпців таблиці будуть виведені в рядок через кому;
  - h. запит з використанням сортування по декільком стовпцям в різному порядку;
  - i. запити згідно варіанту завдання.
- 2) Робота з представленнями (view):
  - a. створити представлення з конкретним переліком атрибутів, котрі обираються, та котре містить дані з декількох таблиць;
  - b. створити представлення, котре містить дані з декількох таблиць та використовує представлення, котре створене в п.а;
  - c. модифікувати представлення з використанням команди ALTER VIEW;
- 3) Для кожного з запитів та завдань п.1 та п.2 навести їх словесний (сутнісний) опис та призначення.
- 4) Оформити звіт з роботи. В звіт включити запити, їх словесний опис та результати виконання.

#### Варіант 1

Програмне забезпечення військової частини. Військові частини округу розквартировані по різних місцях дислокації, причому в одному місці можуть розташовуватися кілька частин. Кожна військова частина складається з рот, роти з взводів, взводи з відділень, в свою чергу військові частини об'єднуються в

дивізії, корпуси або бригади, а ті в армії. Військовий округ представлений офіцерським складом (генерали, полковники, підполковники, майори, капітани, лейтенанти) і рядовим і сержантським складом (старшини, сержанти, прапорщики, рядові). Кожна з перерахованих категорій військовослужбовців може мати характеристики, властиві тільки цій категорії: для генералів це може бути дата закінчення академії, дата присвоєння генеральського звання і т.д. Кожне з підрозділів має командира, причому військовослужбовці офіцерського складу можуть командувати будь-яким з перерахованих вище підрозділів, а військовослужбовці рядового і сержантського складу тільки взводом і відділенням. Всі військовослужбовці мають одну або кілька військових спеціальностей. Кожна військова частина має бойову і транспортну техніку: БМП, тягачі, автотранспорт тощо. і озброєння: карабіни, автоматична зброя, артилерія, ракетне озброєння тощо. Кожна з перерахованих категорій бойової техніки і озброєння також має специфічні, притаманні лише їй атрибути і по кожній категорії може бути кілька видів техніки і озброєння. Треба мати можливість отримувати інформацію про всі частини військового округу, дані про офіцерський, рядовий та сержантський склад, отримувати місця дислокації, дані про наявне озброєння тощо.

#### **Запити:**

- а) Яка частина має найбільшу кількість офіцерського складу та найменшу кількість озброєння одночасно.
- б) Визначить яким типом підрозділу найчастіше командують офіцери зі званням підполковника.

#### **а. Запит із використанням функції COUNT (Знайти кількість службовців у кожній військовій частині):**

```
SELECT mu.name_ AS military_unit_name, COUNT(sa.serviceman_id) AS
servicemen_count
FROM serviceman_affiliation sa JOIN military_unit mu ON sa.unit_id = mu.id_
WHERE sa.unit_type = 'military_unit' GROUP BY mu.name_ ;
```

	military_unit_name character varying (64)	servicemen_count bigint
1	Bratya	8
2	Mayonsk	7
3	Vera	8

#### **б. Запит із використанням функції SUM (Підрахувати загальну кількість ресурсів, прикріплених до кожної військової частини):**

```
SELECT mu.name_ AS military_unit_name, SUM(1) AS total_resources
FROM resource_military_unit rmu JOIN military_unit mu ON rmu.military_unit_id
= mu.id_;
```

GROUP BY mu.name\_;

	military_unit_name character varying (64)	total_resources bigint
1	Compatsk	5
2	Mayonsk	5
3	Vzvod	5
4	Bratya	6
5	Vladiminsk	5
6	Vera	6

**с. Запит із групуванням по декільком стовпцям (Отримати кількість службовців по рангах і підрозділах):**

```
SELECT s.rank_ AS rank, mu.name_ AS military_unit_name, COUNT(s.id_) AS  
servicemen_count  
FROM serviceman s JOIN serviceman_affiliation sa ON s.id_ = sa.serviceman_id  
JOIN military_unit mu ON sa.unit_id = mu.id_  
WHERE sa.unit_type = 'military_unit'  
GROUP BY s.rank_, mu.name_;
```

	rank character varying (32)	military_unit_name character varying (64)	servicemen_count bigint
1	flagman	Bratya	3
2	flagman	Mayonsk	3
3	general	Vera	1
4	major	Bratya	1
5	major	Mayonsk	1
6	officer	Bratya	2

**d. Запит із умовою відбору груп HAVING (Отримати військові підрозділи, у яких більше 10 службовців):**

```
SELECT mu.name_ AS military_unit_name, COUNT(sa.serviceman_id) AS  
servicemen_count  
FROM serviceman_affiliation sa JOIN military_unit mu ON sa.unit_id = mu.id_  
WHERE sa.unit_type = 'military_unit' GROUP BY mu.name_  
HAVING COUNT(sa.serviceman_id) > 3;
```

	military_unit_name character varying (64)	servicemen_count bigint
1	Bratya	8
2	Vera	8

**f. Запит із використанням row\_number() OVER (Пронумерувати службовців у кожному підрозділі):**

```
SELECT s.fio, sa.unit_id, ROW_NUMBER() OVER (PARTITION BY sa.unit_id
ORDER BY s.fio) AS row_num
FROM serviceman s JOIN serviceman_affiliation sa ON s.id_ = sa.serviceman_id;
```

	fio character varying (64)	unit_id integer	row_num bigint
1	Anas.	1	1
2	Arsen	1	2
3	Borya	1	3
4	Dmitro	1	4
5	Mia	1	5
6	Olena	1	6

**g. Запит, де значення стовпця виводяться через кому (Отримати список службовців у кожному підрозділі через кому):**

```
SELECT mu.name_ AS military_unit_name, STRING_AGG(s.fio, ', ') AS
servicemen_list
FROM serviceman s JOIN serviceman_affiliation sa ON s.id_ = sa.serviceman_id
JOIN military_unit mu ON sa.unit_id = mu.id_
WHERE sa.unit_type = 'military_unit' GROUP BY mu.name_;
```

	military_unit_name character varying (64)	servicemen_list text
1	Bratya	FictBot, Denys, Piata, Vadim, Vladik, Ilya, Yarik, De...
2	Mayonsk	Loleps, KPI, Kolya, Maksik, Braga, Andrii, Danya
3	Vera	Sasha, Anas., Arsen, Mia, Popov, Olena, Borya, D...

**h. Запит із сортуванням по кількох стовпцях у різному порядку (Вивести службовців із сортуванням спочатку за ранком (зростання), потім за датою народження (спадання)):**

```
SELECT fio, rank_, birth_date
```

FROM serviceman  
ORDER BY rank\_ ASC, birth\_date DESC;

	fio character varying (64)	rank_ character varying (32)	birth_date date
1	Danya	flagman	2008-01-01
2	Braga	flagman	2006-01-01
3	Derii	flagman	2005-01-01
4	Yarik	flagman	2005-01-01
5	Maksik	flagman	2005-01-01
6	FictBot	flagman	2000-01-01
7	Arsen	general	2006-01-01
8	Denys	major	2007-01-01
9	Loleps	major	2005-01-01
10	Ilya	officer	2008-01-01
11	KPI	officer	2006-01-01
12	Piata	officer	2004-01-01
13	Vadim	private	2007-01-01
14	Borya	private	2007-01-01
15	Mia	private	2006-01-01
16	Vladik	private	2006-01-01
17	Popov	private	2006-01-01
18	Dmitro	private	2006-01-01
19	Sasha	private	2006-01-01
20	Kolya	private	2003-01-01
21	Anas.	private	2003-01-01

**а. Створення представлення з даними з кількох таблиць (Створимо представлення, яке показує список військових частин із загальною кількістю службовців у кожній частині):**

```
CREATE VIEW military_unit_servicemen_count AS
SELECT
    mu.id_ AS military_unit_id,
    mu.name_ AS military_unit_name,
    COUNT(sa.serviceman_id) AS servicemen_count
FROM military_unit mu
LEFT JOIN serviceman_affiliation sa ON mu.id_ = sa.unit_id AND sa.unit_type =
'military_unit'
GROUP BY mu.id_, mu.name_;
```

	military_unit_id integer	military_unit_name character varying (64)	servicemen_count bigint
1	5	Vzvod	0
2	4	Compatsk	0
3	6	Vladiminsk	0
4	2	Bratya	8
5	1	Vera	8
6	3	Mayonsk	7

**б. Створення представлення, що використовує перше (Створимо представлення, яке додатково включає загальну кількість ресурсів, закріплених за кожною військовою частиною. Для цього використовуватимемо попереднє представлення):**

```
CREATE VIEW military_unit_details AS
SELECT
    mus.military_unit_id,
    mus.military_unit_name,
    mus.servicemen_count,
    COUNT(rmu.resource_id) AS resource_count
FROM military_unit_servicemen_count mus
LEFT JOIN resource_military_unit rmu ON mus.military_unit_id =
rmu.military_unit_id
GROUP BY mus.military_unit_id, mus.military_unit_name, mus.servicemen_count;
```

	military_unit_id integer	military_unit_name character varying (64)	servicemen_count bigint	resource_count bigint
1	2	Bratya	8	6
2	6	Vladiminsk	0	5
3	5	Vzvod	0	5
4	4	Compatsk	0	5
5	3	Mayonsk	7	5
6	1	Vera	8	6

**с. Модифікація представлення (Перейменування представлення):**

```
ALTER VIEW military_unit_details RENAME TO unit_interesting_details;
```

```

1 select * from unit_interesting_details;
2 |

```

	military_unit_id integer	military_unit_name character varying (64)	servicemen_count bigint	resource_count bigint
1	2	Bratya	8	6
2	6	Vladiminsk	0	5
3	5	Vzvod	0	5
4	4	Compatsk	0	5
5	3	Mayonsk	7	5
6	1	Vera	8	6

**а. Яка частина має найбільшу кількість офіцерського складу та найменшу кількість озброєння одночасно:**

```

WITH OfficerCount AS (
    SELECT
        sa.unit_id AS unit_id,
        COUNT(*) AS officer_count
    FROM serviceman_affiliation sa
    WHERE sa.is_commander = true
    GROUP BY sa.unit_id
),
WeaponCount AS (
    SELECT
        rmu.military_unit_id AS unit_id,
        COUNT(*) AS weapon_count
    FROM resource_military_unit rmu
    INNER JOIN resource r ON rmu.resource_id = r.id_
    WHERE r.type_ = 'weapon'
    GROUP BY rmu.military_unit_id
)
SELECT
    o.unit_id,
    o.officer_count,
    COALESCE(w.weapon_count, 0) AS weapon_count
FROM OfficerCount o
LEFT JOIN WeaponCount w ON o.unit_id = w.unit_id
ORDER BY o.officer_count DESC, w.weapon_count ASC

```



LIMIT 1;

	unit_id integer	officer_count bigint	weapon_count bigint
1	3	1	2

**б. Визначить яким типом підрозділу найчастіше командують офіцери зі званням підполковника:**

```
WITH colonel_commands AS (  
    SELECT  
        sa.unit_type,  
        COUNT(*) AS command_count  
    FROM serviceman s  
    JOIN serviceman_affiliation sa ON s.id_ = sa.serviceman_id  
    WHERE s.rank_ = 'major' AND sa.is_commander = TRUE  
    GROUP BY sa.unit_type  
)  
SELECT  
    unit_type,  
    command_count  
FROM colonel_commands  
ORDER BY command_count DESC  
LIMIT 1;
```

	unit_type unit_type	command_count bigint
1	military_unit	2

## **Висновок**

Під час виконання лабораторної роботи нарешті було вивчено основні оператори та функції SQL, які застосовуються для вибірки, групування та сортування даних. Також, було використано функції агрегування, такі як COUNT і SUM, синтаксис групування даних за декількома стовпцями, а також використання умов відбору груп за допомогою HAVING з GROUP BY та без нього.

Також були вивчені представлення (view), які забезпечують спрощений та більш структурований спосіб для збереження даних для подальшої обробки та опрацювання іншими запитам SQL. Також було розглянуто модифікацію представлення за допомогою ALTER VIEW.

Результатом стало створення нових запитів SQL, для яких вже треба нові вимоги, наприклад представлення даних в табличному вигляді і не тільки. Практичне використання вбудованих функцій, умов фільтрації, групування та представлень сприяє покращення навичок будування та супроводження SQL скриптів.