

**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 9

**Виконав(ла)**

ІП-35 Євстаф'єв Даніїл Євгенович  
(шифр, прізвище, ім'я, по батькові)

**Перевірив**

Марченко Олена Іванівна  
(прізвище, ім'я, по батькові)

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

**Варіант 9**

Облік внутрішньо офісних витрат компанії. Співробітники компанії мають можливість здійснювати дрібні покупки для потреб компанії, надаючи у бухгалтерію товарний чек. Необхідно відстежувати внутрішньо офісні витрати. Компанія складається з відділів. Кожен відділ має назву. У кожному відділі працює певна кількість співробітників. Співробітники можуть здійснювати покупки відповідно до видів витрат. Кожен вид витрат має назву, деякий опис і граничну суму коштів, які можуть бути витрачені в певний місяць. При кожній купівлі співробітник оформляє документ, де вказує вид витрати, дату, суму і відділ. Дані про витрати необхідно зберігати не лише в цілому по відділу, але і по окремих співробітниках. Нормативи по витрачання засобів встановлюються не в цілому,, а по кожному відділу за кожен місяць. Не використані в поточному місяці гроші можуть бути використані пізніше.

**Постановка задачі лабораторної роботи No 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.

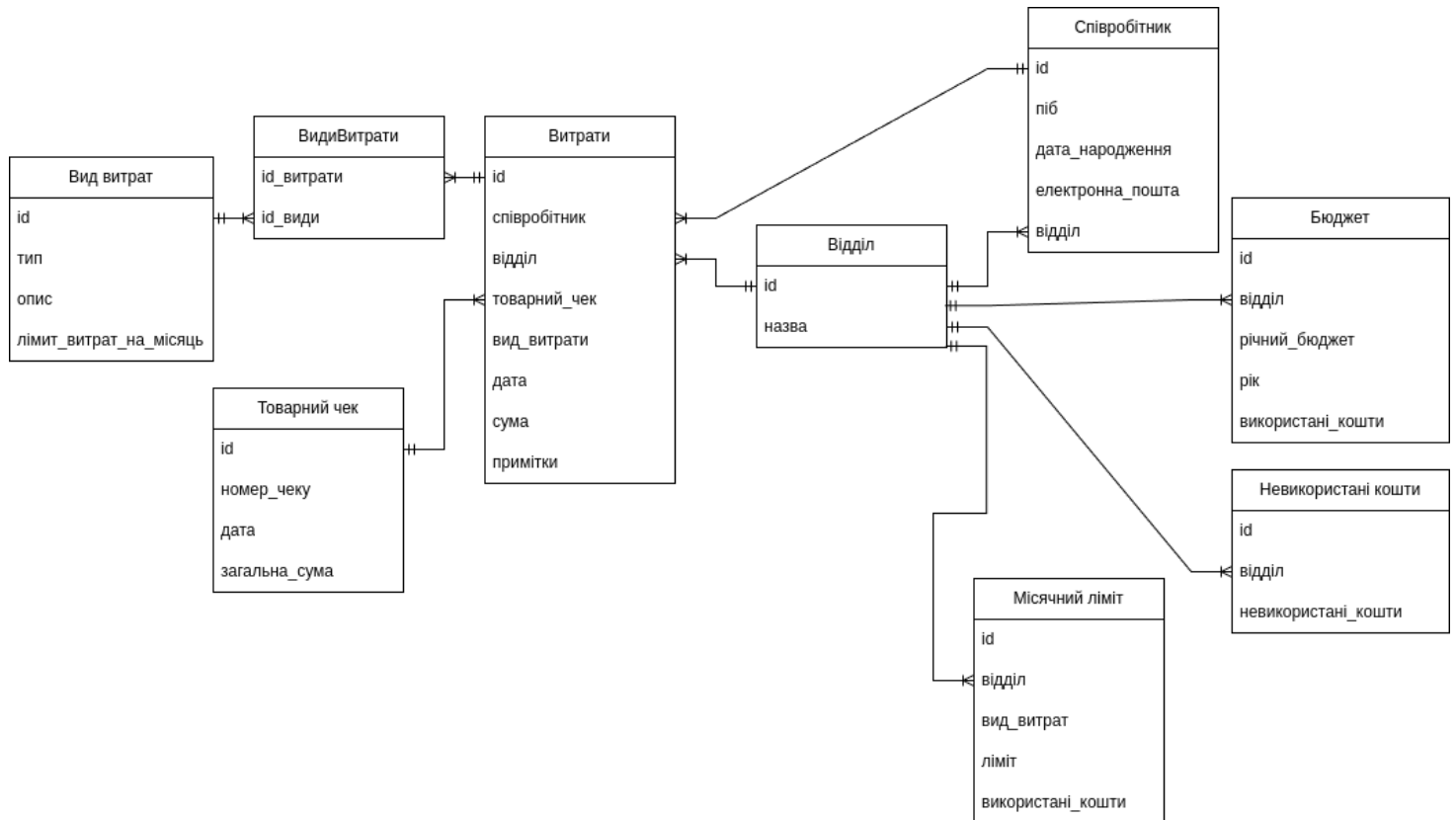
2) Робота з представленнями (view):

- 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.

## Модель бази даних:



1) Створити наступні запити (в запитах повинні використовуватись 2 та більше таблиць):

а. запит з використанням функції COUNT;

```
--а. запит з використанням функції COUNT;--
SELECT
    e.full_name AS employee_name,
    COUNT(ex.id) AS expense_count
FROM
    employee e
    JOIN
    expenses ex 1<->1..n: ON e.id = ex.employee_id
GROUP BY
    e.full_name;
```

**Опис:** Запит підраховує кількість витрат для кожного працівника.

employee_name	expense_count
Doyle Kunze	2
Mattie Koelpin	2
Jared Runolfsdottir	1
Loren Gottlieb	1
Miss Eleanor Rath	1
Mr. Robert Smith	1

b. запит з використанням функції SUM;

```
--b. запит з використанням функції SUM;--
SELECT
    d.name AS department_name,
    SUM(ex.amount) AS total_expenses
FROM
    department d
    JOIN
    expenses ex 1<->1..n: ON d.id = ex.department_id
GROUP BY
    d.name;
```

**Опис:** Запит підсумовує загальні витрати для кожного підрозділу.

department_name	total_expenses
Hirthe - Predovic	23235
Bashirian - Waelchi	24923
Hahn, Heller and Nikolaus	67074
Lowe - Grady	77819
Nader - Skiles	15772
Hyatt - Ward	92355
Roob and Sons	46306

с. запит з використанням групування по декільком стовпцям;

```
--с. запит з використанням групування по декільком стовпцям;--
SELECT
    d.name AS department_name,
    et.type AS expense_type,
    SUM(ex.amount) AS total_expenses
FROM
    department d
    JOIN
    expenses ex 1<->1..n: ON d.id = ex.department_id
    JOIN
    expense_expense_type eet 1<->1..n: ON ex.id = eet.expense_id
    JOIN
    expense_type et 1..n<->1: ON eet.expense_type_id = et.id
GROUP BY
    d.name, et.type;
```

**Опис:** Групує дані за назвою підрозділу та типом витрат, виводячи загальну суму витрат для кожної комбінації.

department_name	expense_type	total_expenses
Kling - Konopelski	Shoes	15824
Shanahan Group	Electronics	83066
Schaden - Hermiston	Books	47037
Mayer, White and Ullrich	Tools	60993
Corkery - Flatley	Kids	73037
Breitenberg - Rempel	Clothing	70217
Klocko, Howe and Hauck	Automotive	75723
Kozey, Treutel and Schumm	Movies	95946

d. запит з використанням умови відбору груп HAVING;

```
--d. запит з використанням умови відбору груп HAVING;--
SELECT
    d.name AS department_name,
    SUM(ex.amount) AS total_expenses
FROM
    department d
    JOIN
    expenses ex 1<->1..n: ON d.id = ex.department_id
GROUP BY
    d.name
HAVING
    SUM(ex.amount) > 100000;
```

**Опис:** Запит відбирає підрозділи, у яких загальні витрати перевищують 100000.

<input type="checkbox"/> department_name ▾	<input type="checkbox"/> total_expenses ▾
Little Inc	249712
Waelchi and Sons	111988
Emard LLC	129193
Tremblay LLC	143247
Sporer and Sons	114959
Sipes and Sons	155425
Stark Inc	171761

e. запит з використанням HAVING без GROUP BY;

```
--e. запит з використанням HAVING без GROUP BY;--
SELECT
    SUM(ex.amount) AS total_expenses
FROM
    expenses ex
HAVING
    SUM(ex.amount) > 50000;
```

**Опис:** Запит визначає загальну суму витрат, яка перевищує 50000.

<input type="checkbox"/> total_expenses ▾
1 502326158

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

```
--f. запит з використанням функцій row_number() over ....;--
SELECT
    e.full_name AS employee_name,
    d.name AS department_name,
    SUM(ex.amount) AS total_expenses,
    ROW_NUMBER() OVER (PARTITION BY d.name ORDER BY SUM(ex.amount) DESC) AS row_num
FROM
    employee e
    JOIN
    expenses ex 1<->1..n: ON e.id = ex.employee_id
    JOIN
    department d 1..n<->1: ON e.department_id = d.id
GROUP BY
    e.full_name, d.name
ORDER BY
    d.name, row_num;
```

**Опис:** Присвоює кожному працівнику у межах одного підрозділу номер залежно від суми витрат (від найбільшої до найменшої).

	employee_name	department_name	total_expenses	row_num
1	Terence Lubowitz	Abbott and Sons	85553	1
2	Roy Christiansen V	Abbott - Beahan	68575	1
3	Miss Lila VonRueden	Abbott - Beahan	65606	2
4	Camille McDermott	Abbott - Berge	93359	1
5	William Armstrong	Abbott - Berge	32230	2
6	Carla Spencer	Abbott - Fadel	43680	1
7	Ms. Donna Bednar	Abbott, Gorczany and Shields	97057	1
8	Kristi McGlynn	Abbott, Gorczany and Shields	91707	2
9	Cesar Osinski	Abbott Group	79478	1



г. запит, в котрому значення одного зі стовпців таблиці будуть виведені в рядок через кому;

```
--г. запит, в котрому значення одного зі стовпців таблиці будуть виведені в рядок через кому;--
SELECT
    d.name AS department_name,
    STRING_AGG(et.type, ', ') AS expense_types
FROM
    department d
    JOIN
    expenses ex 1<->1..n: ON d.id = ex.department_id
    JOIN
    expense_expense_type eet 1<->1..n: ON ex.id = eet.expense_id
    JOIN
    expense_type et 1..n<->1: ON eet.expense_type_id = et.id
GROUP BY
    d.name;
```

**Опис:** Запит об'єднує типи витрат для кожного підрозділу в один рядок через кому.

department_name	expense_types
Abbott and Sons	Jewelery, Books, Grocery
Abbott - Beahan	Music
Abbott - Berge	Electronics
Abbott, Christiansen and Bins	Movies
Abbott, Cole and Toy	Games
Abbott - Collier	Books
Abbott - Dicki	Automotive
Abbott - Fadel	Jewelery
Abbott, Feeney and Hartmann	Beauty

h. запит з використанням сортування по декільком стовпцям в різному порядку;

```
--h. запит з використанням сортування по декільком стовпцям в різному порядку;--
SELECT
    d.name AS department_name,
    e.full_name,
    ex.date,
    ex.amount
FROM
    expenses ex
    JOIN employee e 1..n<->1: ON ex.employee_id = e.id
    JOIN department d 1..n<->1: ON ex.department_id = d.id
ORDER BY
    d.name ASC,
    ex.amount DESC;
```

**Опис:** Запит сортує підрозділи за назвою у зростаючому порядку, а витрати — у спадному.

<input type="checkbox"/> department_name ▾	÷	<input type="checkbox"/> full_name ▾	÷	<input type="checkbox"/> date ▾	÷	<input type="checkbox"/> amount ▾	÷
Abbott and Sons		Dr. Lucille Murray-Weimann		2024-11-05		59000	
Abbott and Sons		Rosa Okuneva		2024-11-05		7869	
Abbott and Sons		Dianne Langosh		2024-11-05		1901	
Abbott - Beahan		Duane Sanford		2024-11-05		57101	
Abbott - Berge		Dominic Bergstrom		2024-11-05		26855	
Abbott, Christiansen and Bins		Alex Steuber		2024-11-05		4110	
Abbott, Cole and Toy		Wilbert Gleichner		2024-11-05		43397	
Abbott - Collier		Allison Wintheiser		2024-11-05		7906	
Abbott - Dicki		Charlotte Hackett		2024-11-05		74699	

i. запити згідно варіанту завдання.

a) Чи всі підрозділи вклались в бюджет по кожному виду витрат минулого місяця?

```
--i. запити згідно варіанту завдання.--  
--a) Чи всі підрозділи вклались в бюджет по кожному виду витрат минулого місяця?--  
SELECT  
    d.name AS department_name,  
    b.monthly_budget,  
    b.used_funds,  
    CASE  
        WHEN b.used_funds <= b.monthly_budget THEN 'Within Budget'  
        ELSE 'Over Budget'  
    END AS status  
FROM  
    budget b  
    JOIN department d ON b.department_id = d.id  
WHERE  
    DATE_PART('month', b.date) = DATE_PART('month', CURRENT_DATE) - 1  
    AND DATE_PART('year', b.date) = DATE_PART('year', CURRENT_DATE);
```

Опис: Перевіряє, чи підрозділи залишилися у межах бюджету за попередній місяць.

department_name	monthly_budget	used_funds	status
Bernier Inc	693398	753164	Over Budget
Turner LLC	306942	363475	Over Budget
Boyer - Brakus	702755	848288	Over Budget
Wolff - Beier	456576	271627	Within Budget

б) Визначить відділи з найбільшими витратами за минулі два місяця.

```
--б) Визначить відділи з найбільшими витратами за минулі два місяця--  
SELECT  
    d.name AS department_name,  
    SUM(ex.amount) AS total_expenses  
FROM  
    expenses ex  
    JOIN department d 1..n<->1: ON ex.department_id = d.id  
WHERE  
    ex.date >= DATE_TRUNC('month', CURRENT_DATE) - INTERVAL '2 months'  
GROUP BY  
    d.name  
ORDER BY  
    total_expenses DESC  
LIMIT 5;
```

**Опис:** Визначає підрозділи з найбільшими витратами за останні два місяці.

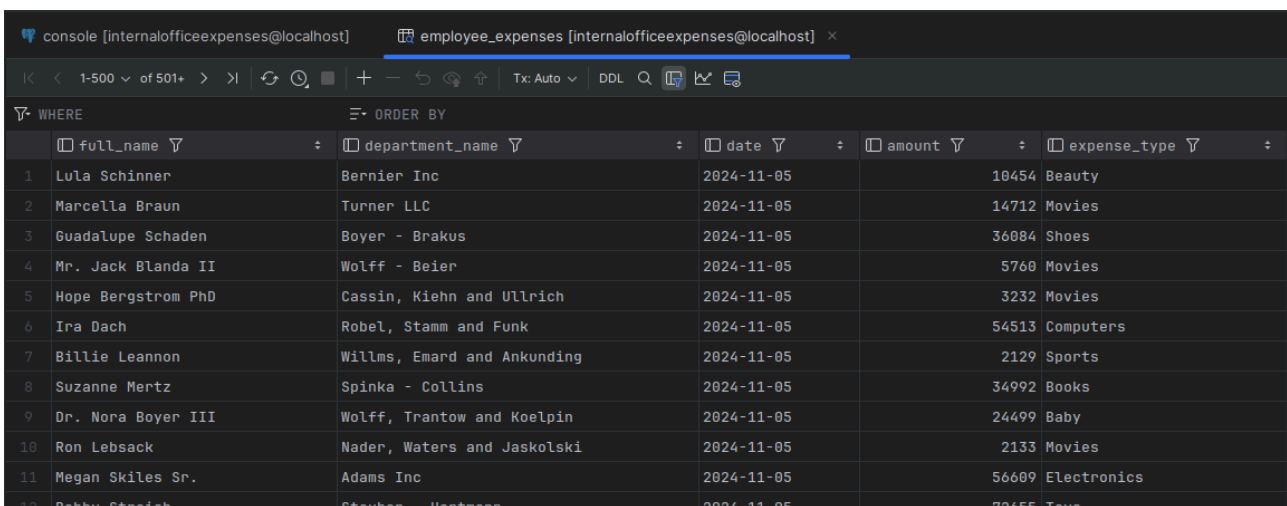
	department_name	total_expenses
1	Koepp Group	468581
2	Littel Group	427225
3	Harvey Inc	417046
4	Marks Inc	403095
5	Bode Inc	395820

## 2) Робота з представленнями (view):

- а. створити представлення з конкретним переліком атрибутів, котрі  
обираються, та котре містить дані з декількох таблиць;

```
--2) Робота з представленнями (view):--  
--а. створити представлення з конкретним переліком атрибутів, котрі обираються, та котре містить дані з декількох таблиць;--  
CREATE VIEW employee_expenses AS  
SELECT  
    e.full_name,  
    d.name AS department_name,  
    ex.date,  
    ex.amount,  
    et.type AS expense_type  
FROM  
    expenses ex  
    JOIN employee e 1..n<->1: ON ex.employee_id = e.id  
    JOIN department d 1..n<->1: ON ex.department_id = d.id  
    JOIN expense_expense_type eet 1<->1..n: ON ex.id = eet.expense_id  
    JOIN expense_type et 1..n<->1: ON eet.expense_type_id = et.id;
```

**Опис:** Представлення об'єднує інформацію про працівників, їхні витрати, підрозділи та типи витрат.



	full_name	department_name	date	amount	expense_type
1	Lula Schinner	Bernier Inc	2024-11-05	10454	Beauty
2	Marcella Braun	Turner LLC	2024-11-05	14712	Movies
3	Guadalupe Schaden	Boyer - Brakus	2024-11-05	36084	Shoes
4	Mr. Jack Blanda II	Wolff - Beier	2024-11-05	5760	Movies
5	Hope Bergstrom PhD	Cassin, Kiehn and Ullrich	2024-11-05	3232	Movies
6	Ira Dach	Robel, Stamm and Funk	2024-11-05	54513	Computers
7	Billie Leannon	Willms, Emard and Ankunding	2024-11-05	2129	Sports
8	Suzanne Mertz	Spinka - Collins	2024-11-05	34992	Books
9	Dr. Nora Boyer III	Wolff, Trantow and Koelpin	2024-11-05	24499	Baby
10	Ron Lebsack	Nader, Waters and Jaskolski	2024-11-05	2133	Movies
11	Megan Skiles Sr.	Adams Inc	2024-11-05	56609	Electronics
12	Bobby Steich	Steuber - Hartmann	2024-11-05	72655	Toys

- б. створити представлення, котре містить дані з декількох таблиць та  
використовує представлення, котре створене в п.а;

```
--б. створити представлення, котре містить дані з декількох таблиць та використовує представлення, котре створене в п.а;--  
CREATE VIEW department_expenses_summary AS  
SELECT  
    department_name,  
    expense_type,  
    SUM(amount) AS total_expenses  
FROM  
    employee_expenses  
GROUP BY  
    department_name, expense_type;
```

**Опис:** Узагальнює дані з попереднього представлення, виводячи загальні витрати по типах для кожного підрозділу.

console [internalofficeexpenses@localhost]    department...es\_summary [internalofficeexpenses@localhost] ×

1-500 of 501+    Tx: Auto    DDL    🔍    📄    📊    📁

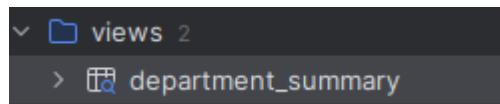
WHERE    ORDER BY

	department_name	expense_type	total_expenses
1	Kling - Konopelski	Shoes	15824
2	Shanahan Group	Electronics	83066
3	Schaden - Hermiston	Books	47037
4	Mayer, White and Ullrich	Tools	60993
5	Corkery - Flatley	Kids	73037
6	Breitenberg - Rempel	Clothing	70217
7	Klocko, Howe and Hauck	Automotive	75723
8	Kozey, Treutel and Schumm	Movies	95946
9	Hermann, Stark and Davis	Grocery	32504
10	Boyer - Brakus	Shoes	36084
11	Witting - McDermott	Health	28665
12	Cripe and Sons	Kids	86686

с. модифікувати представлення з використанням команди ALTER VIEW;

```
--с. модифікувати представлення з використанням команди ALTER VIEW;--
ALTER VIEW department_expenses_summary RENAME TO department_summary;
```

**Опис:** Команда змінює назву існуючого представлення на нову.



**Висновок:** У ході виконання лабораторної роботи були вивчені основні оператори та функції SQL, які застосовуються в реляційних СУБД для вибірки, групування та сортування даних. Зокрема, опрацьовано 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 розглянуто функцію **ROW\_NUMBER() OVER**, яка дозволяє нумерувати записи в межах груп, а також методи об'єднання значень стовпця в рядок через кому. Здійснено практику сортування результатів запитів за декількома стовпцями в різному порядку.

Також увагу було приділено роботі з представленнями (view), які забезпечують спрощений доступ до даних із декількох таблиць, дозволяють створювати логічні подання даних, зручні для повторного використання. Було опрацьовано створення представлень із конкретним набором атрибутів, створення представлень на основі інших представлень, а також їх модифікація за допомогою команди **ALTER VIEW**.

Результатом роботи стало створення низки SQL-запитів, які відповідають вимогам завдання, а також глибше розуміння принципів роботи з реляційними базами даних. Практичне використання вбудованих функцій, умов фільтрації, групування та представлень сприяє формуванню навичок оптимального проектування та використання SQL-запитів у реальних задачах.