**Міністерство освіти і науки України**

**Національний університет харчових технологій**

Кафедра інформаційних технологій, штучного

інтелекту та кібербезпеки

**Звіт**

із лабораторної роботи № **3**

з дисципліни «Менеджмент баз та сховищ даних, їх аналіз»

на тему: «DML. Об’єднання та функції.»

Варіант **13**

Виконав:

Студент групи КН-1-3М

Кучерявий М. В.

Перевірив:

асистент Струзік В.А.

**Київ — 2024**

**Завдання:**

1. Використовуйте репозиторій з навчальними базами даних.

2. Створити запит, з використанням об’єднання INNER JOIN.

3. Створити запит, з використанням об’єднання LEFT JOIN.

4. Створити запит, з використанням об’єднання RIGHT JOIN.

5. Створити запит, з використанням об’єднання CROSS JOIN.

6. Створити запит, що буде містити в собі відразу декілька видів

об’єднання.

7. Створити запит, що буде містити об’єднання UNION.

8. В запиті, створеному у пункті 6 цієї лабораторної роботи, замінити

UNION на UNION ALL та порівняти отримані результати з

результатами, отриманими у пункті 6.

9. Створити запити, що будуть загалом містити мінімум три різні

рядкові функції.

10.Створити запити, що будуть загалом містити мінімум три різні

функції дати-часу.

11.Створити запити, що будуть загалом містити мінімум три різні

агрегаційні функції.

12.Створити запити, що будуть загалом містити мінімум три різні

числові функції.

13. Створити запит, з використанням функції управління потоком

CASE.

14. Створити запит, з використанням функції управління потоком IF.

15. Створити запит, з використанням функції управління потоком

IFNULL.

**Хід роботи:**

2. **SELECT** *p*.payment\_id, *i*.invoice\_id

**FROM** invoice *i*

**INNER** **JOIN** payments *p* **ON** *i*.invoice\_id = *p*.invoice\_id

Повертає таблицю рахунків та оплат по ним

3. **SELECT** *i*.invoice\_id, *p*.payment\_id

**FROM** invoice *i*

**LEFT** **JOIN** payments *p* **ON** *p*.invoice\_id = *i*.invoice\_id

**WHERE** **ISNULL**(p.payment\_id)

Повертає таблицю неоплачених рахунків

4. **SELECT** *i*.invoice\_id, *c*.customer\_id, *c*.customer\_short\_name

**FROM** invoice *i*

**RIGHT** **JOIN** customer *c* **ON** *c*.customer\_id = *i*.customer\_id

**WHERE** **ISNULL**(i.invoice\_id)

Повертає таблицю клієнтів, по яким не створені рахунки

5. **SELECT** \*

**FROM** customer *c*

**CROSS** **JOIN** orderreg *o*

**WHERE** *c*.customer\_id = *o*.customer\_id

Повертає таблицю клієнтів, та їх замовлень

6. **SELECT** \*

**FROM** customer *c*

**RIGHT** **JOIN** orderreg *o* **ON** *c*.customer\_id = *o*.customer\_id

**RIGHT** **JOIN** invoice *i* **ON** *o*.order\_id = *i*.order\_id

Повертає клієнтів замовлень та рахунків

7 — 8 (**SELECT** *c*.customer\_short\_name **FROM** customer *c*)

**UNION**

(**SELECT** *c2*.customer\_short\_name **FROM** orderreg *o* **LEFT** **JOIN** customer *c2* **ON** *c2*.customer\_id = *o*.customer\_id)

(**SELECT** *c*.customer\_short\_name **FROM** customer *c*)

**UNION** **ALL**

(**SELECT** *c2*.customer\_short\_name **FROM** orderreg *o* **LEFT** **JOIN** customer *c2* **ON** *c2*.customer\_id = *o*.customer\_id)

Запити демонструють роботу union та я union all. Вони роблять таблицю клієнтів які робили замовлення. Різниця полягає в тому, що union дає лише унікальні значення, а union all включає в себе велику кількість копій  
  
9. **SELECT**

**LENGTH**(*c*.customer\_short\_name) **as** *"length"*,

**REVERSE**(*c*.customer\_short\_name) **as** *"reverce"*,

**CONCAT**(*c*.customer\_short\_name, **"country"**) **as** *"name"*

**FROM** customer *c*   
  
Повертає таблицю видозмінених назв клієнтів: довжина назви, назва навпаки, назва + рядок «country»

10. **SELECT**

**EXTRACT**(**YEAR** **FROM** o.order\_date) **as** **"year"**,

**DATEDIFF**(**CURRENT\_DATE**(), o.order\_date) **as** **"difrance"**

**FROM** orderreg o

**ORDER** **BY** **DATEDIFF**(**CURRENT\_DATE**(), o.order\_date) **DESC**

Повертає дату створення замовлення а саме: рік, та скільки часу пройшло від створення замовлення дотепер

11. **SELECT** **MAX**(*i*.invoice\_date) **as** *"Last date"*,

**COUNT**(*i*.invoice\_id) **as** *"count"*,

**ROUND**( **AVG**(*i*.invoice\_total))**as** *"AVG invoce price"*

**FROM** invoice *i*

Повертає найновішу дату створення рахунку, кількість рахунків та середню ціну рахунку

12. **SELECT**

*i*.invoice\_total,

**POW**(*i*.invoice\_total, 2) **as** *"2pow"* ,

**SQRT**(*i*.invoice\_total) **as** *"root"*,

**PI**() **as** *"pi"*

**FROM** invoice *i*

Повертає ціну рахунку з математичними модифікацієми а також число пі

13. **SELECT** **CASE** *c*.customer\_region\_code

**WHEN** **"ch"** **THEN** **"china"**

**WHEN** **"de"** **THEN** **"denmark"**

**ELSE** **"other"**

**END** **as** *"name"*

**FROM** customer *c*

Перетворює коди країн на реальні назви за певною умовою

14. **SELECT**

**IF** (*i*.invoice\_total > 1000, **"High"**, **"Low"**) **as** *"type"*,

*i*.invoice\_total

**FROM** invoice *i*

Перетворює ціну вище чи нижче за 1000

15. **SELECT** **IFNULL**(*i*.invoice\_id, **"unknown"**), *c*.customer\_id, *c*.customer\_short\_name

**FROM** invoice *i*

**RIGHT** **JOIN** customer *c* **ON** *c*.customer\_id = *i*.customer\_id

Повертає список клієнтів та їх рахунків. Оскільки не всі клієнти мають рахунки то в поля рахунків вписується «unknown»