**Міністерство освіти і науки України**

**Національний університет харчових технологій**

Кафедра інформаційних технологій, штучного

інтелекту та кібербезпеки

**Звіт**

із лабораторної роботи № **7**

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

на тему: «DDL. Індексація, обмеження, тригери, представлення, події.»

Варіант **13**

Виконав:

Студент групи КН-1-3М

Кучерявий М. В.

Перевірив:

асистент Струзік В.А.

**Київ — 2024**

**Хід роботи:**

1. Створити базу даних з кодуванням utf8mb4 та порівнянням utf8mb4\_unicode\_ci.

**CREATE** **DATABASE** **IF** **NOT** **EXISTS** test\_DB7

**CHARACTER** **SET** utf8mb4

**COLLATE** utf8mb4\_unicode\_ci

2. Користуючись навчальною базою даних за своїм варіантом створити у новій базі даних три таблиці з мінімум трьома полями різних типів даних.

3. Користуючись запитом INSERT SELECT наповнити створені таблиці даними.

**CREATE** **TABLE** customer (

customer\_id **INT**,

customer\_short\_name **CHAR**(5) **NOT** **NULL**,

customer\_region\_code **CHAR**(5) **NOT** **NULL**

)

**CREATE** **TABLE** orderreg (

order\_id **INT**,

customer\_id **INT**,

order\_date **DATETIME**,

oreder\_type **CHAR**(15)

)

**CREATE** **TABLE** orderitems (

order\_item\_id **INT**,

item\_no **INT**,

order\_id **INT**,

order\_item\_quantity **INT**,

order\_item\_rate **FLOAT**,

oreder\_item\_total\_amount **FLOAT**

)

**INSERT** **INTO** test\_DB7.customer

**SELECT** \*

**FROM** salesorder.customer *c*

**INSERT** **INTO** test\_DB7.orderreg

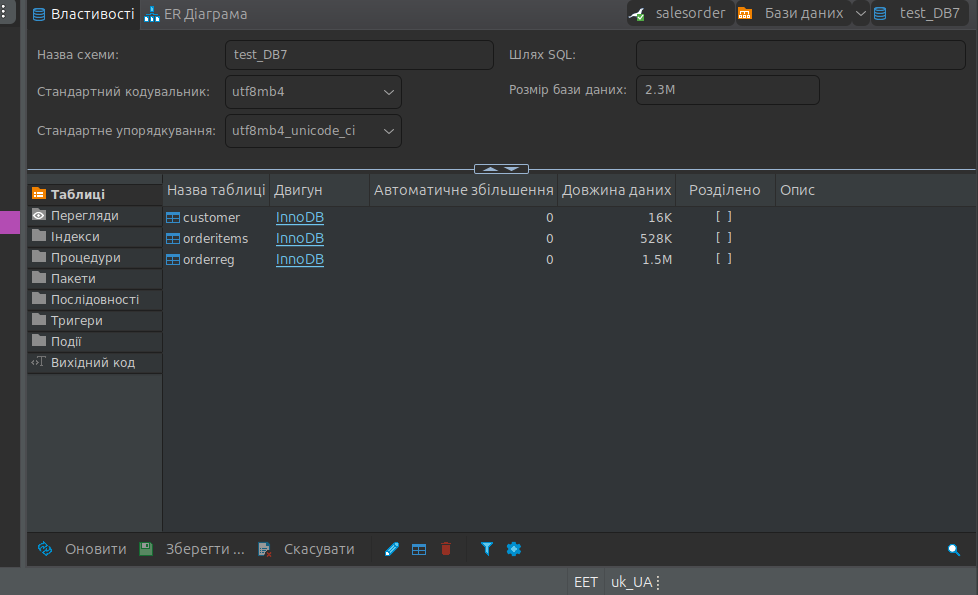
**SELECT** \*

**FROM** salesorder.orderreg

**INSERT** **INTO** test\_DB7.orderitems

**SELECT** \*

**FROM** salesorder.orderitems *o*



4. Написати запити на створення та видалення PRIMARY KEY з одним та декількома полями.

**ALTER** **TABLE** customer

**ADD** **PRIMARY** **KEY** (customer\_id)

**ALTER** **TABLE** customer

**ADD** **PRIMARY** **KEY** (customer\_id, customer\_short\_name)

5. Написати запити на створення та видалення UNIQUE KEY з одним та декількома полями.

**CREATE** **UNIQUE** **INDEX** customerINDEX **ON** customer(customer\_id)

**CREATE** **UNIQUE** **INDEX** customerINDEX **ON** customer(customer\_id, customer\_short\_name)

6. Написати запити на створення та видалення KEY з одним та декількома полями.

**ALTER** **TABLE** customer

**DROP** **CONSTRAINT** customer.**PRIMARY**

**DROP** **INDEX** customerINDEX **ON** customer

7. Написати запити на створення та видалення обмежень CHECK.

**ALTER** **TABLE** customer

**ADD** **CONSTRAINT** c\_name **CHECK**(**LENGTH**(customer\_short\_name) <= 3)

**ALTER** **TABLE** customer

**DROP** **CONSTRAINT** c\_name

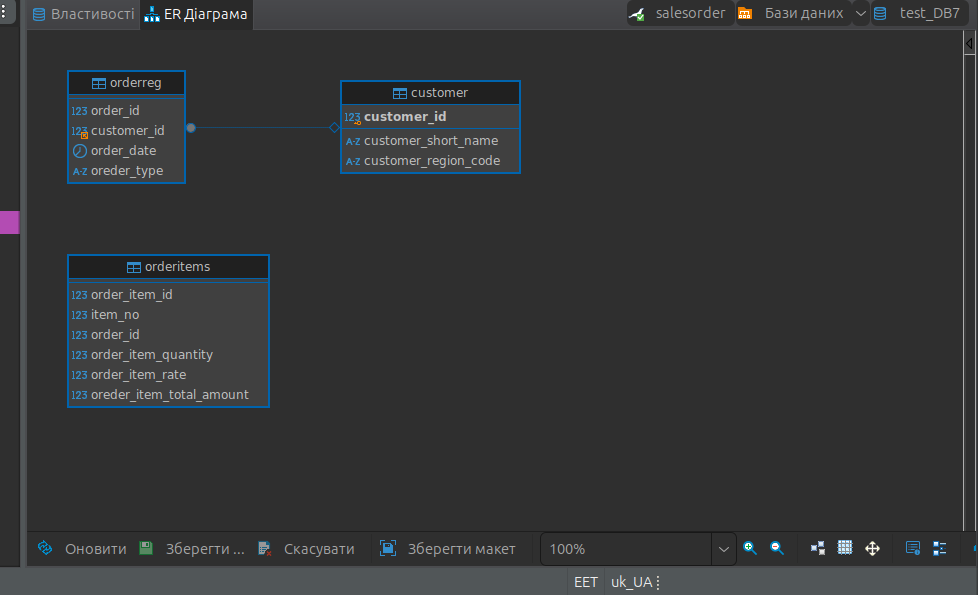
8. Написати запити на створення та видалення обмежень FOREIGN KEY.

**ALTER** **TABLE** orderreg

**ADD** **CONSTRAINT** cf\_index **FOREIGN** **KEY** (customer\_id) **REFERENCES** customer(customer\_id)

**ALTER** **TABLE** orderreg

**DROP** **CONSTRAINT** cf\_index



9. Використовуйте [репозиторій з навчальними базами даних](https://gitlab.com/nuft3/da-sql).

10. До довільної таблиці з навчальної бази даних створити таблицю, що буде зберігати історію вставок, оновлень та видалень записів обраної таблиці. Запис історії виконати за допомогою тригерів.

**CREATE** **TRIGGER** on\_customer\_add

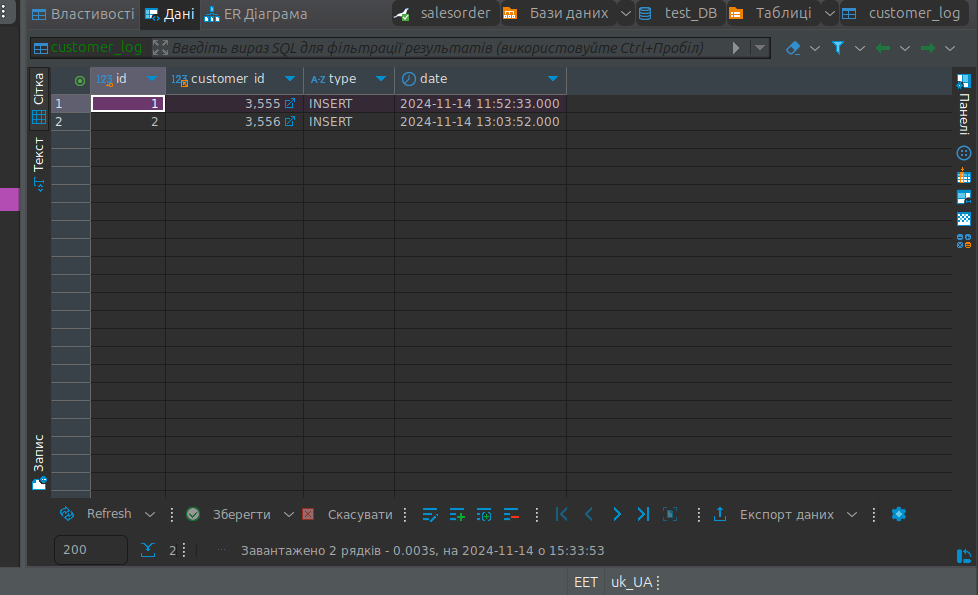
**AFTER** **INSERT**

**ON** test\_DB.customer

**FOR** **EACH** **ROW**

**INSERT** **INTO** customer\_log (**type**, **date**, customer\_id)

**VALUES** ('INSERT', **NOW**(), **NEW**.customer\_id)

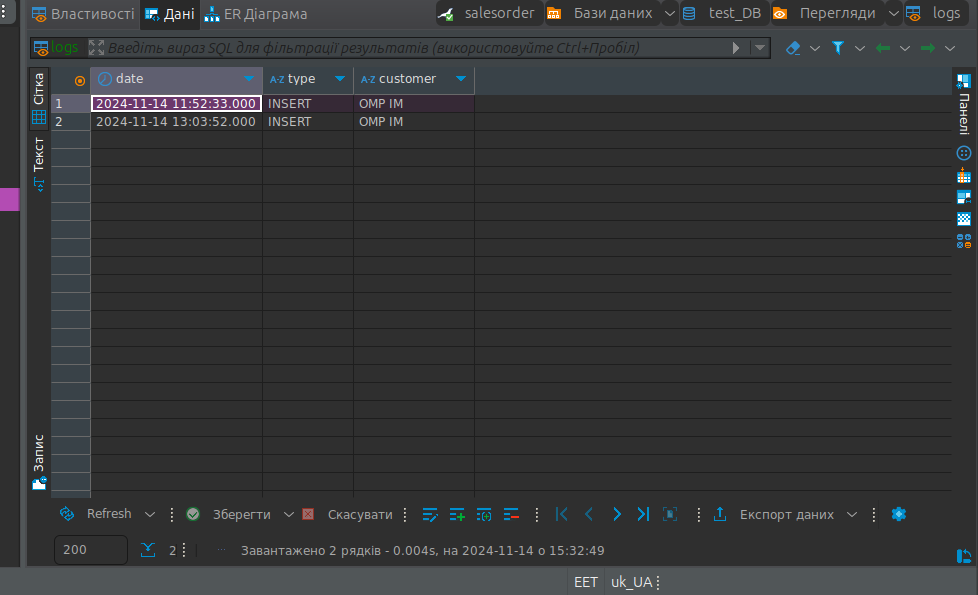


11.  Написати запити на створення та видалення представлення з використанням підзапиту.

**CREATE** **VIEW** customer\_info **AS**

**SELECT** c.fullName **AS** **"name"**, (**SELECT** **COUNT**(order\_id) **AS** **"count"** **FROM** orderreg o **WHERE** o.customer\_id = c.customer\_id) **AS** **"count"**

**FROM** customer c



12.  Написати запити на створення та видалення представлення з використанням об’єднання.

**CREATE** **VIEW** logs **AS**

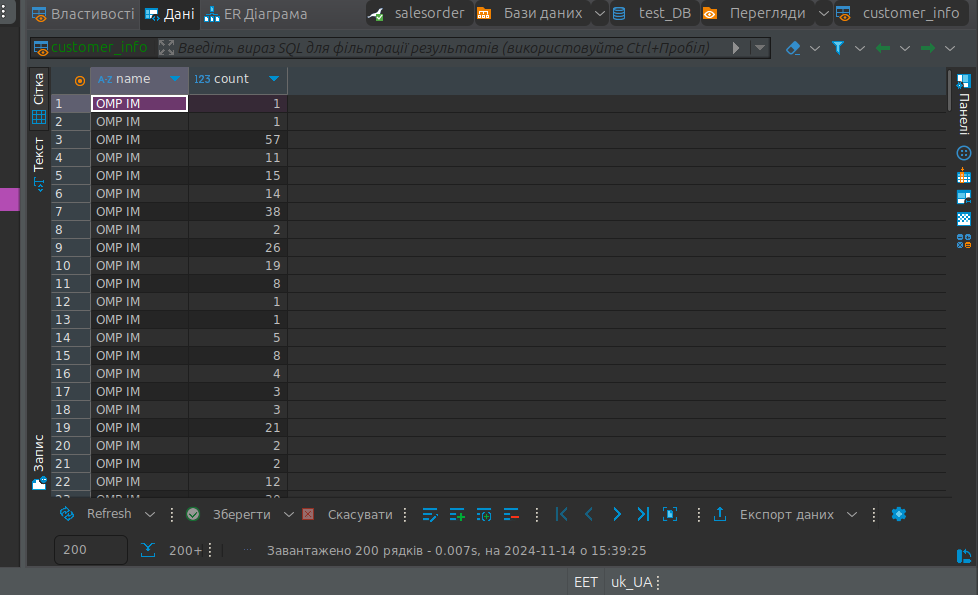
**SELECT** cl.date **AS** **"date"**, cl.type **AS** **"type"**, c.fullName **AS** **"customer"**

**FROM** customer\_log cl

**RIGHT** **JOIN** customer c

**ON** cl.customer\_id = c.customer\_id

**WHERE** cl.type **IS** **NOT** **NULL**



13. Створити подію, котра щогодини зберігає у таблицю інформацію про кількість записів по кожній іншій таблиці бази даних.

**CREATE** EVENT update\_count

**ON** SCHEDULE **EVERY** 1 **MINUTE**

DO

**INSERT** **INTO** tables\_row\_count\_report (**date**, customer\_count, customer\_log\_count, order\_items\_conut, oreder\_reg\_count)

**VALUES** (**NOW**(),

(**SELECT** **COUNT**(customer\_id) **FROM** customer),

(**SELECT** **COUNT**(**date**) **FROM** customer\_log),

(**SELECT** **COUNT**(order\_item\_id) **FROM** orderitems),

(**SELECT** **COUNT**(order\_id) **FROM** orderreg)

)

