МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

**Лабораторні роботи № 1 – 4**

з дисципліни

«Організація баз даних та знань»

Виконав:

студент групи КН-208

Келемен С. Й.

**Викладач:**

Мельникова Н. І.

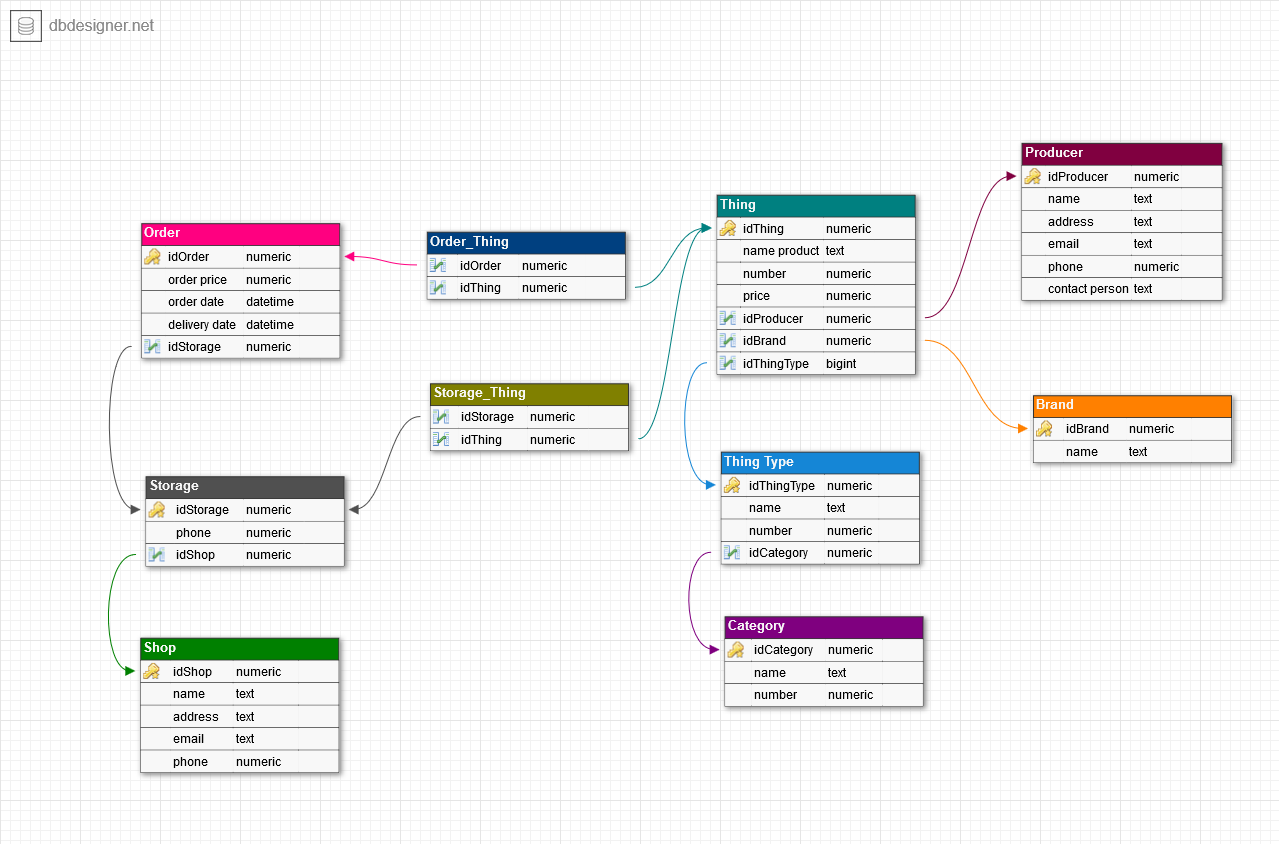
Львів – 2019 р

**Предметна область:** магазин.

**Лабораторна робота №1**

**Тема**: проектування баз даних.

**Мета**: визначити предметну область бази даних, визначити об’єкти, що підлягають представленню в базі даних, побудувати формалізований опис об’єктів, визначити первинні та зовнішні ключі, побудувати контекстну діаграму предметної області.

*Контекстна діаграма*

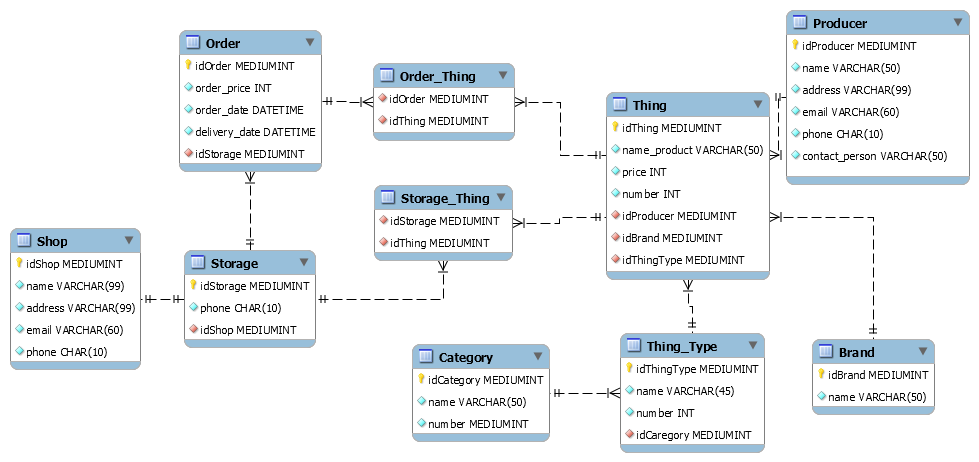
**Висновок**

На цій лабораторній роботі було спроектовано контекстну діаграму бази даних для системи постачання товарів від постачальника до магазину. Зокрема, створено таблиці відношень, які містять атрибути, первинні та зовнішні ключі, що виражають зв’язки між поданими таблицями.

**Лабораторна робота №2**

**Тема**: cтворення таблиць бази даних засобами SQL.

**Мета роботи**:побудувати даталогічну модель бази даних; визначити типи, розмірності та обмеження полів; визначити обмеження таблиць; розробити SQL запити для створення спроектованих таблиць.

*****Даталогічна модель*

*SQL-запити*

**CREATE** **DATABASE** `Shop`;

**CREATE** **TABLE** `Shop`.`Producer` (

`idProducer` **MEDIUMINT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT**,

`name` **VARCHAR**(**50**) **NOT** **NULL**,

`address` **VARCHAR**(**99**) **NOT** **NULL**,

`email` **VARCHAR**(**60**) **NOT** **NULL**,

`phone` **CHAR**(**10**) **NOT** **NULL**,

`contact\_person` **VARCHAR**(**50**) **NOT** **NULL**,

**PRIMARY** **KEY** (`idProducer`)

);

**CREATE** **TABLE** `Shop`.`Shop` (

`idShop` **MEDIUMINT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT**,

`name` **VARCHAR**(**99**) **NOT** **NULL**,

`address` **VARCHAR**(**99**) **NOT** **NULL**,

`email` **VARCHAR**(**60**) **NOT** **NULL**,

`phone` **CHAR**(**10**) **NOT** **NULL**,

**PRIMARY** **KEY** (`idShop`)

);

**CREATE** **TABLE** `Shop`.`Storage` (

`idStorage` **MEDIUMINT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT**,

`phone` **CHAR**(**10**) **NOT** **NULL**,

`idShop` **MEDIUMINT** **UNSIGNED** **NOT** **NULL**,

**PRIMARY** **KEY** (`idStorage`),

**INDEX** `Storage\_Shop` (`idShop`),

**CONSTRAINT** `Storage\_Shop`

**FOREIGN** **KEY** (`idShop`)

**REFERENCES** `Shop`.`Shop` (`idShop`)

**ON** **DELETE** NO ACTION

**ON** **UPDATE** NO ACTION

);

**CREATE** **TABLE** `Shop`.`Order` (

`idOrder` **MEDIUMINT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT**,

`order\_price` **INT** **NOT** **NULL**,

`order\_date` **DATETIME** **NOT** **NULL**,

`delivery\_date` **DATETIME** **NOT** **NULL**,

`idStorage` **MEDIUMINT** **UNSIGNED** **NOT** **NULL**,

**PRIMARY** **KEY** (`idOrder`),

**INDEX** `Order\_Storage` (`idStorage`),

**CONSTRAINT** `Order\_Storage`

**FOREIGN** **KEY** (`idStorage`)

**REFERENCES** `Shop`.`Storage` (`idStorage`)

**ON** **DELETE** NO ACTION

**ON** **UPDATE** NO ACTION

);

**CREATE** **TABLE** `Shop`.`Brand` (

`idBrand` **MEDIUMINT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT**,

`name` **VARCHAR**(**50**) **NOT** **NULL**,

**PRIMARY** **KEY** (`idBrand`),

**UNIQUE** **INDEX** `name\_UNIQUE` (`name`)

);

**CREATE** **TABLE** `Shop`.`Category` (

`idCategory` **MEDIUMINT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT**,

`name` **VARCHAR**(**50**) **NOT** **NULL**,

number **MEDIUMINT** **UNSIGNED** **NOT** **NULL**,

**PRIMARY** **KEY** (`idCategory`),

**UNIQUE** **INDEX** `name\_UNIQUE` (`name`)

);

**CREATE** **TABLE** `Shop`.`Thing\_Type` (

`idThingType` **MEDIUMINT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT**,

`name` **VARCHAR**(**45**) **NOT** **NULL**,

`number` **INT** **UNSIGNED** **NOT** **NULL**,

`idCategory` **MEDIUMINT** **UNSIGNED** **NOT** **NULL**,

**PRIMARY** **KEY** (`idThingType`),

**UNIQUE** **INDEX** `name\_UNIQUE` (`name`),

**INDEX** `Thing\_Type\_Category` (`idCategory`),

**CONSTRAINT** `Thing\_Type\_Category`

**FOREIGN** **KEY** (`idCategory`)

**REFERENCES** `Shop`.`Category` (`idCategory`)

**ON** **DELETE** NO ACTION

**ON** **UPDATE** NO ACTION

);

**CREATE** **TABLE** `Shop`.`Thing` (

`idThing` **MEDIUMINT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT**,

`name\_product` **VARCHAR**(**50**) **NOT** **NULL**,

`price` **INT** **UNSIGNED** **NOT** **NULL**,

`number` **INT** **UNSIGNED** **NOT** **NULL**,

`idProducer` **MEDIUMINT** **UNSIGNED** **NOT** **NULL**,

`idBrand` **MEDIUMINT** **UNSIGNED** **NOT** **NULL**,

`idThingType` **MEDIUMINT** **UNSIGNED** **NOT** **NULL**,

**PRIMARY** **KEY** (`idThing`),

**INDEX** `Thing\_Producer` (`idProducer`),

**INDEX** `Thing\_Brand` (`idBrand`),

**INDEX** `Thing\_Thing\_Type` (`idThingType`),

**CONSTRAINT** `Thing\_Producer`

**FOREIGN** **KEY** (`idProducer`)

**REFERENCES** `Shop`.`Producer` (`idProducer`),

**CONSTRAINT** `Thing\_Brand`

**FOREIGN** **KEY** (`idBrand`)

**REFERENCES** `Shop`.`Brand` (`idBrand`)

**ON** **DELETE** NO ACTION

**ON** **UPDATE** NO ACTION,

**CONSTRAINT** `Thing\_Thing\_Type`

**FOREIGN** **KEY** (`idThingType`)

**REFERENCES** `Shop`.`Thing\_Type` (`idThingType`)

**ON** **DELETE** NO ACTION

**ON** **UPDATE** NO ACTION

);

**CREATE** **TABLE** `Shop`.`Order\_Thing` (

`idOrder` **MEDIUMINT** **UNSIGNED** **NOT** **NULL**,

`idThing` **MEDIUMINT** **UNSIGNED** **NOT** **NULL**,

**INDEX** `Order\_Thing\_Order` (`idOrder`),

**INDEX** `Order\_Thing\_Thing` (`idThing`),

**CONSTRAINT** `Order\_Thing\_Order`

**FOREIGN** **KEY** (`idOrder`)

**REFERENCES** `Shop`.`Order` (`idOrder`)

**ON** **DELETE** NO ACTION

**ON** **UPDATE** NO ACTION,

**CONSTRAINT** `Order\_Thing\_Thing`

**FOREIGN** **KEY** (`idThing`)

**REFERENCES** `Shop`.`Thing` (`idThing`)

**ON** **DELETE** NO ACTION

**ON** **UPDATE** NO ACTION

);

**CREATE** **TABLE** `Shop`.`Storage\_Thing` (

`idStorage` **MEDIUMINT** **UNSIGNED** **NOT** **NULL**,

`idThing` **MEDIUMINT** **UNSIGNED** **NOT** **NULL**,

**INDEX** `Storage\_Thing\_Storage` (`idStorage`),

**INDEX** `Storage\_Thing\_Thing` (`idThing`),

**CONSTRAINT** `Storage\_Thing\_Storage`

**FOREIGN** **KEY** (`idStorage`)

**REFERENCES** `Shop`.`Storage` (`idStorage`)

**ON** **DELETE** NO ACTION

**ON** **UPDATE** NO ACTION,

**CONSTRAINT** `Storage\_Thing\_Thing`

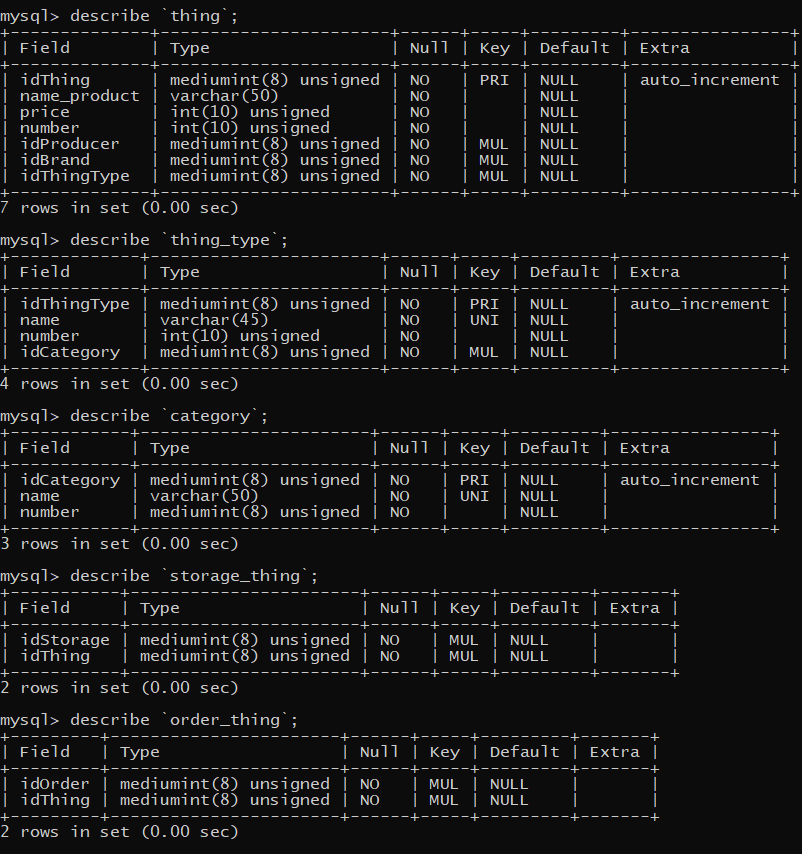
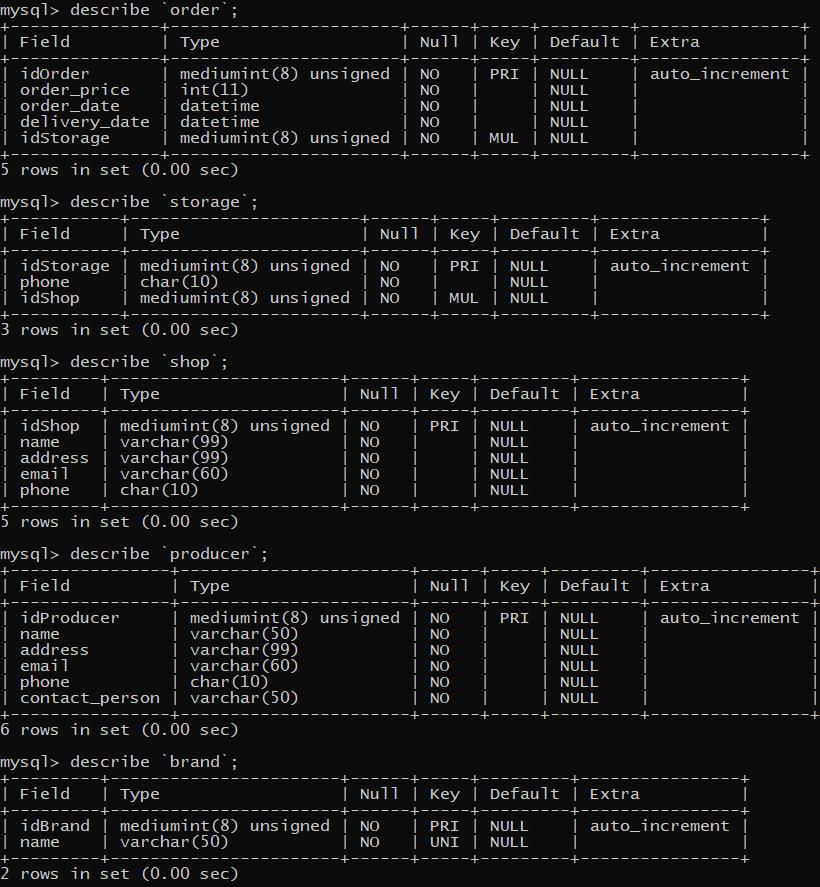
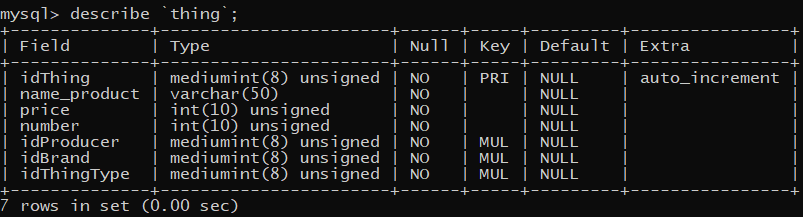
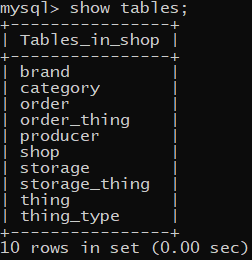
**FOREIGN** **KEY** (`idThing`)

**REFERENCES** `Shop`.`Thing` (`idThing`)

**ON** **DELETE** NO ACTION

**ON** **UPDATE** NO ACTION

);



*Елгьяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяяятаблиці*

**Висновки**

Була побудована даталогічна модель бази даних на основі контекстної діаграми, визначено типи, розмірності та обмеження полів, розроблено запити SQL для створення спроектованих таблиць.

**Лабораторна робота №3**

**Тема:** модифікація структури таблиць бази даних засобами sql.

**Мета:** розробити sql запити для зміни структури таблиць бази даних  
(включення нових полів, вилучення полів таблиць, зміни опису полів, зміни обмежень).

**Хід роботи**

1. Видалення зв’язку між таблицями шляхом видалення обмеження для таблиці:

**ALTER TABLE** Shop.Storage

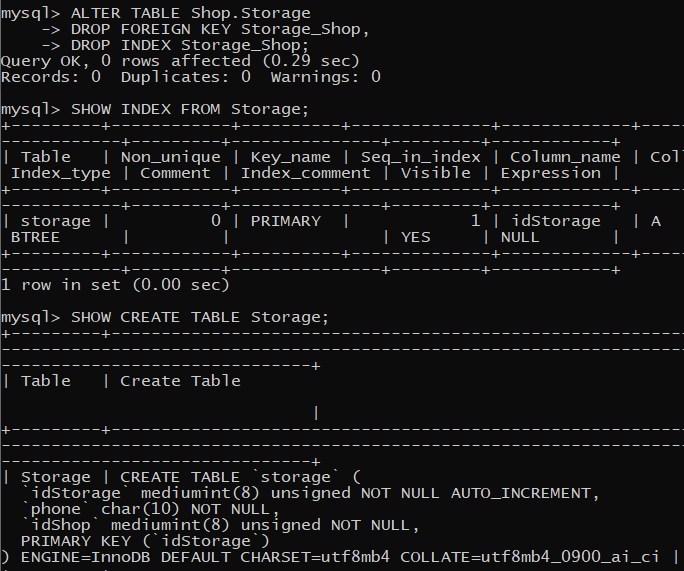
**DROP** **FOREIGN KEY** Storage\_Shop,

**DROP INDEX** Storage\_Shop;

Для перевірки результату можна виконати команди

**SHOW INDEX FROM** Storage

**SHOW CREATE TABLE** Storage

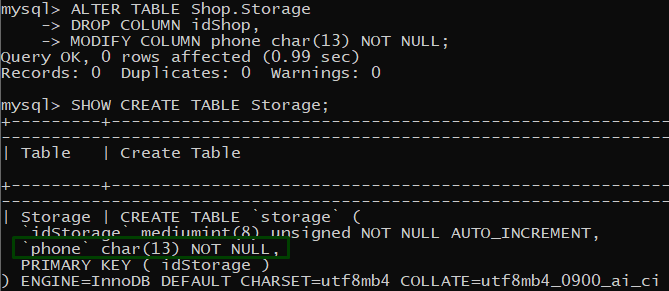


1. Видалення поля і зміна розмірності поля в таблиці Storage:

**ALTER TABLE** Shop.Storage

**DROP COLUMN** idShop,

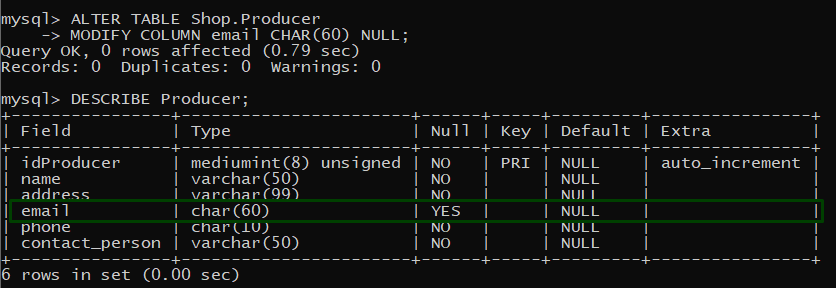
**MODIFY COLUMN** phone char(13) NOT NULL;



1. Зробимо поле email в таблиці Producer необов’язковим, і змінимо його розмірність:

**ALTER TABLE** Shop.Producer

**MODIFY COLUMN** email CHAR(60) NULL;



1. Додамо до таблиці Producer поле idBrand, щоб знати з якими брендами працює виробник. Спочатку змінимо поле idProducer, забравшилічильник (властивість auto\_increment). Після додавання поля idBrand, створимо нове обмеження унікальності для полів (idProducer, idBrand).

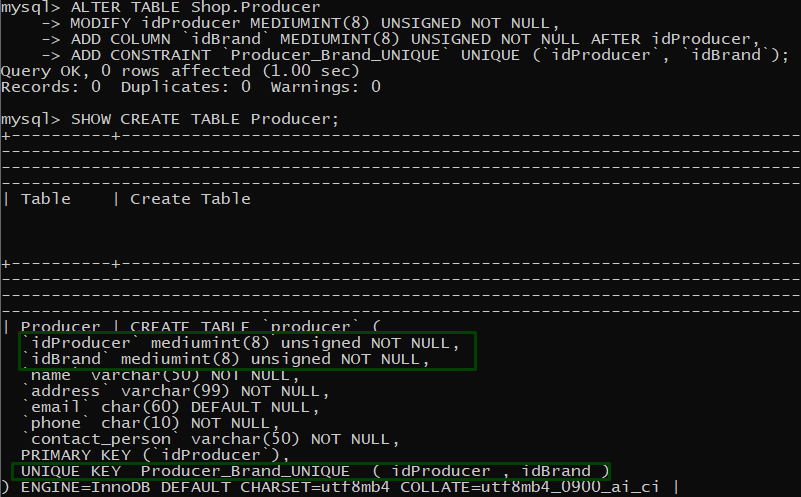
**ALTER TABLE** Shop.Producer

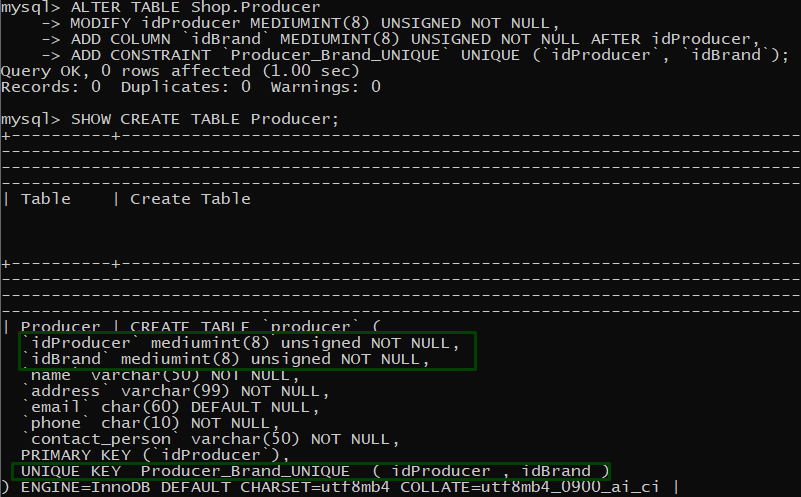
**MODIFY** idProducer MEDIUMINT(8) UNSIGNED NOT NULL,  
**ADD COLUMN `**idBrand` MEDIUMINT(8) UNSIGNED NOT NULL

**AFTER** idProducer,

**ADD CONSTRAINT `**Producer\_Brand\_UNIQUE` **UNIQUE**

(`idProducer`, `idBrand`);



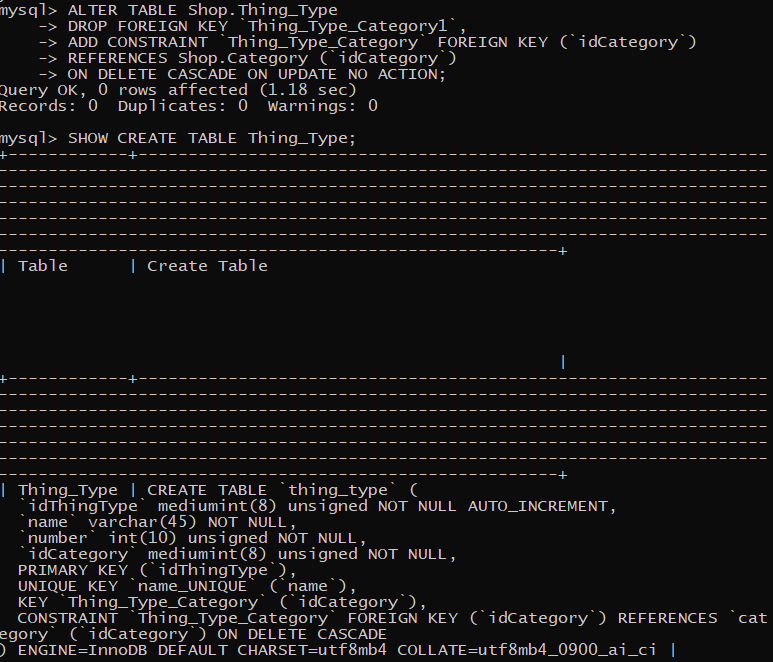
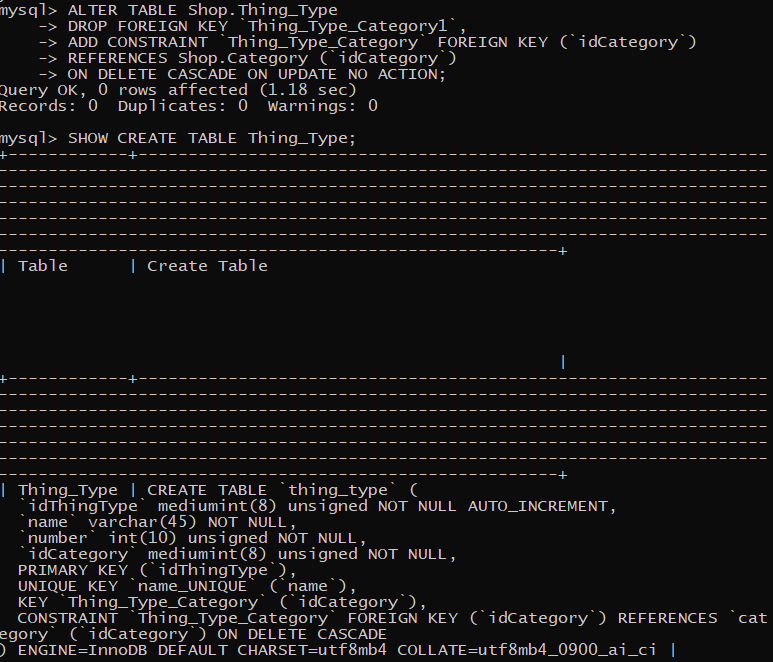


1. Змінимо обмеження цілісності Thing\_Type так, щоб при видаленні  
   категорії з бази видалявся запис про приєднаний до нього вид продукту. Для цього потрібно видалити обмеження і створити нове, оскільки MySQL не підримує директиви ALTER CONSTRAINT.

**ALTER TABLE** Shop.Thing\_Type

**DROP FOREIGN KEY** **`**Thing\_Type\_Category1`,  
**ADD CONSTRAINT `**Thing\_Type\_Category` **FOREIGN KEY** (`idCategory`)

**REFERENCES** Shop.Category (`idCategory`) **ON DELETE** CASCADE **ON UPDATE** NO ACTION;



**Висновки**

Було розглянуто методи модифікації структури таблиць БД засобами SQL та проведено модифікацію полів, зв’язків і обмежень, зокрема розроблено sql запити для включення нових полів, вилучення полів таблиць, зміни опису полів, зміни обмежень.

**Лабораторна робота №4**

**Тема:** запити на додавання, зміну та вилучення даних.

**Мета:** розробити SQL-запити для внесення нових значень в таблиці в режимі одиничного та групового доповнення; розробити SQL-запити для внесення змін в рядки таблиць; розробити SQL-запити для вилучення вибраних рядків.

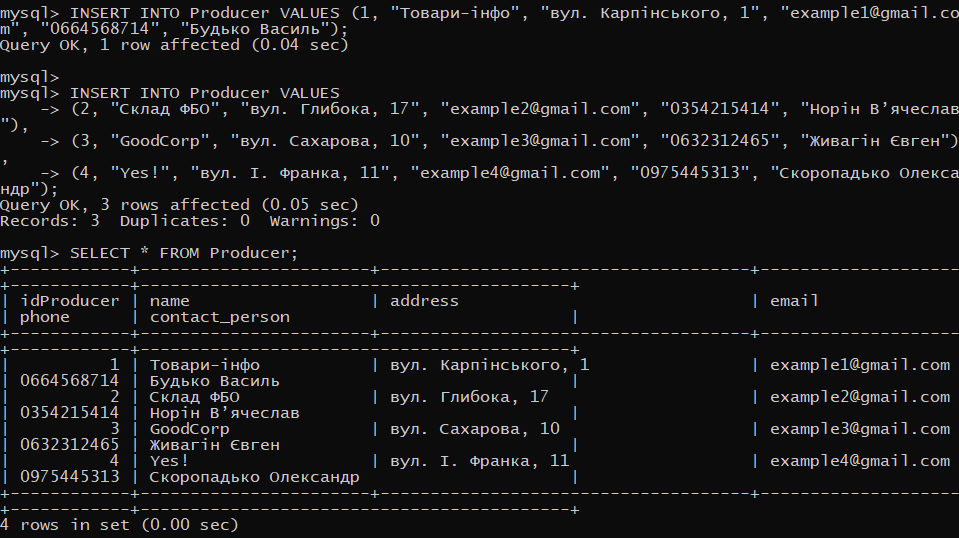
**Хід роботи**

1. Внесення нових значень в таблицю Producer в режимі одиничного доповнення:

**INSERT INTO** Producer **VALUES** (1, ”Товари-інфо”, ”вул. Карпінського, 1”, ”example1@gmail.com”, ”0664568714”, ”Будько Василь”);

Внесення нових значень в таблицю Producer в режимі групового доповнення:

**INSERT INTO** Producer **VALUES**(2, ”Склад ФБО”, ”вул. Глибока, 17”, ”example2@gmail.com”, ”0354215414”, ”НорінВ’ячеслав”),**(**3, ”GoodCorp”, ”вул. Сахарова, 10”, ”example3@gmail.com”, ”0632312465”, ”Живагін Євген”), **(**4, ”Yes!”, ”вул. І. Франка, 11”, ”example4@gmail.com”, ”0975445313”, ”Скоропадько Олександр”);



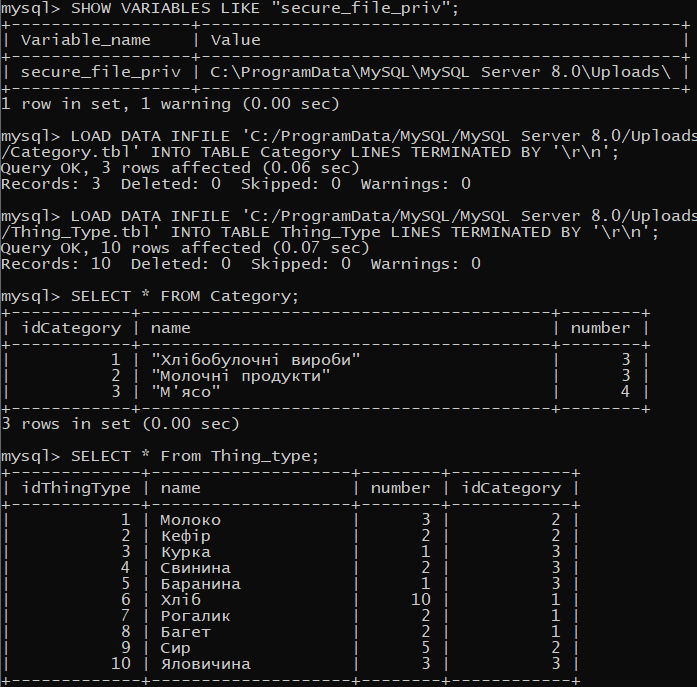
1. Створимо файли з даними для таблиць Category і Thing\_Type. Значення полів повинні розділятись символом табуляції, а кожен рядок таблиці повинен починатися з нового рядка у файлі. Спочатку дізнаємось, куди необхідно зберігати файли, щоб опція secure-file-priv дозволила їх прочитати за допомогою команди:

**SHOW VARIABLES LIKE** "secure\_file\_priv".

Після цього виконаємо дві наступні команди:

**LOAD DATA INFILE** ' C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Thing\_Type.tbl' **INTO TABLE** Category LINES TERMINATED BY '\r\n';

**LOAD DATA INFILE** ' C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Category.tbl' **INTO TABLE** Thing\_Type LINES TERMINATED BY '\r\n';

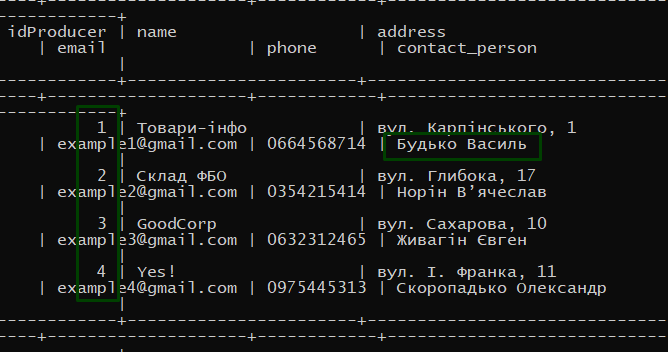


1. Внесення змін в записи таблиці. Змінимо пароль користувача user1 і збільшимо значення ключового поля на одиницю. Для зміни значень ключа потрібно змінити порядок сортування рядків, щоб уникнути суперечності.

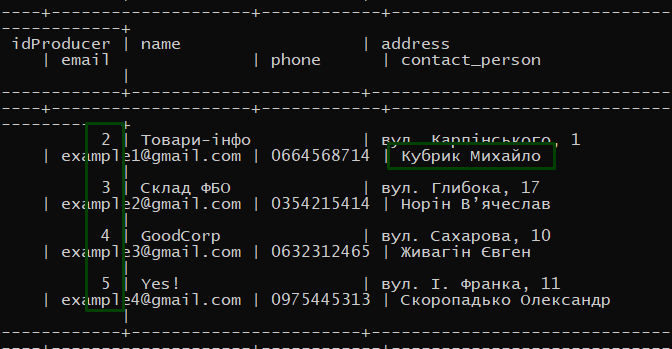
**UPDATE** Producer **SET** contact\_person = “Кубрик Михайло” **WHERE** name= “Товари-інфо”;

**UPDATE** Producer **SET** idProducer = idProducer + 1  
**ORDER BY** idProducer DESC;

Producer до виконання запитів:



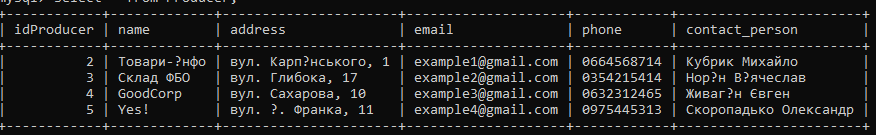
Producer після виконання запитів:



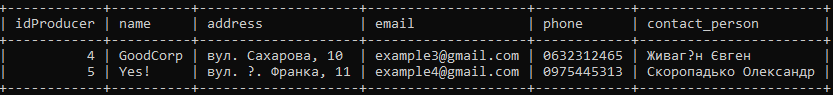
1. Запит для вилучення старих повідомлень користувачів з таблиці Producer.

**DELETE FROM** Producer **WHERE** (email=”[example1@gmail.com](mailto:example1@gmail.com)” || email=”[example2@gmail.com](mailto:example2@gmail.com)”);

Таблиця Producer до видалення інформації:



Таблиця Producer після видалення інформації:



**Висновок**

Було розглянуто способи наповнення і модифікації даних в таблицях БД та проведено модифікацію даних у двох таблицях, зокрема розроблено SQL запити для внесення нових значень в таблиці в режимі одиничного та групового доповнення, змін в рядки таблиць та для вилучення вибраних рядків.