**МІНІСТЕРСТВО ОСВІТИ І НАУКИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

****

**Звіт до лабораторної роботи №2**

З предмету “Організація баз даних та знань”

**Виконав**:

ст. гр. КН-211

Шебеко Андрій

**Викладач:**

Якимишин Х.М.

Львів – 2020

**Лабораторна робота №2**

**Створення 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.

**Короткі теоретичні відомості:**

Щоб створити нову базу даних у командному рядку клієнта MySQL (mysql.exe) слід виконати команду CREATE DATABASE, опис якої подано нижче. Тут і надалі, квадратні дужки позначають необов’язковий аргумент команди, символ "|" позначає вибір між аргументами.

CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] ім’я\_бази

[[DEFAULT] CHARACTER SET кодування]

[[DEFAULT] COLLATE набір\_правил]

ім’я\_бази – назва бази даних (латинські літери і цифри без пропусків);

кодування – набір символів і кодів (koi8u, latin1, utf8, cp1250 тощо);

набір\_правил – правила порівняння рядків символів (див. результат команди show collation).

Нижче наведені деякі допоміжні команди для роботи в СУБД MySQL. Кожна команда і кожен запит в командному рядку повинні завершуватись розділяючим символом ";".

1. Перегляд існуючих баз даних:

SHOW DATABASES

1. Вибір бази даних для подальшої роботи:

USE DATABASE ім’я\_бази

1. Перегляд таблиць в базі даних:

SHOW TABLES [FOR ім’я\_бази]

1. Перегляд опису таблиці в базі:

DESCRIBE ім’я\_таблиці

1. Виконати набір команд з зовнішнього файлу:

SOURCE назва\_файлу

1. Вивести результати виконання подальших команд у зовнішній файл:

\T назва\_файлу

Для роботи зі схемою бази даних існують такі основні команди:

ALTER DATABASE – зміна опису бази даних;

CREATE TABLE – створення нової таблиці;

ALTER TABLE – зміна структури таблиці;

DELETE TABLE – видалення таблиці з бази даних;

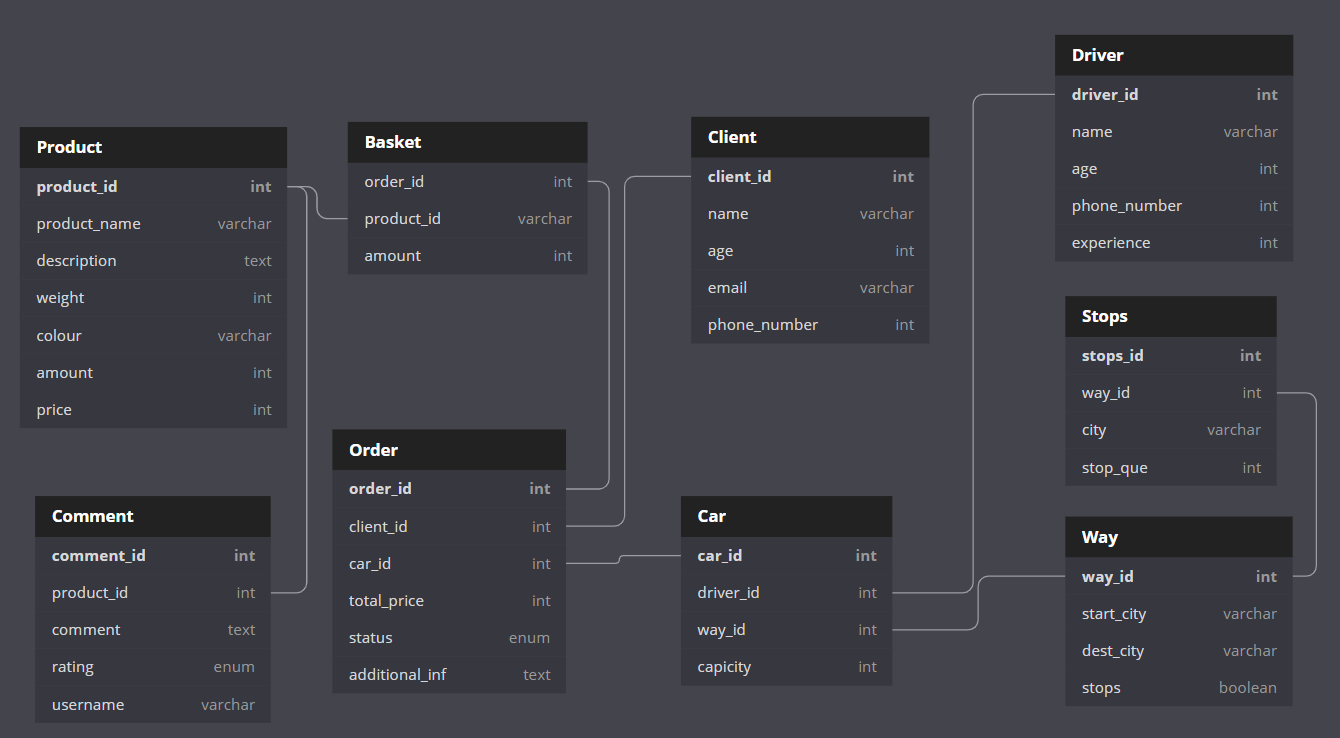
CREATE INDEX – створення нового індексу (для швидкого пошуку даних);

DROP INDEX – видалення індексу;

DROP DATABASE – видалення бази даних.

**Хід роботи:**

Даталогічна модель вимагає визначення конкретних полів бази даних, їхніх типів, обмежень на значення, тощо. На рисунку зображено даталогічну модель проектованої бази даних. Для зв’язку коментарів і продуктів встановлено обмеження цілісності «каскадне оновлення».



Створимо нову базу даних, таблиці та їх вміст, виконавши такі команди:

drop database if exists transport;

create database if not exists transport;

USE transport;

CREATE TABLE `Client` (

`client\_id` int PRIMARY KEY AUTO\_INCREMENT,

`first name` varchar(30),

`second name` varchar(30),

`age` tinyint,

`email` varchar(40),

`phone\_number` int,

`register\_date` DATETIME DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE `Order` (

`order\_id` int PRIMARY KEY AUTO\_INCREMENT,

`client\_id` int,

`car\_id` int,

`total\_price` int,

`status` enum('Active','Completed','Canceled'),

`additional\_inf` text(500)

);

CREATE TABLE `Basket` (

`basket\_id` int PRIMARY KEY AUTO\_INCREMENT,

`order\_id` int,

`product\_id` int,

`amount` smallint

);

CREATE TABLE `Product` (

`product\_id` int PRIMARY KEY AUTO\_INCREMENT,

`product\_name` varchar(60),

`description` text(500),

`weight` float,

`colour` varchar(255),

`amount` smallint,

`price` int

);

CREATE TABLE `Comment` (

`comment\_id` int PRIMARY KEY AUTO\_INCREMENT,

`product\_id` int,

`comment` text(500),

`rating` enum('1','2','3','4','5'),

`username` varchar(50)

);

CREATE TABLE `Car` (

`car\_id` int PRIMARY KEY AUTO\_INCREMENT,

`driver\_id` int,

`way\_id` int,

`capicity` float

);

CREATE TABLE `Driver` (

`driver\_id` int PRIMARY KEY AUTO\_INCREMENT,

`name` varchar(255),

`age` tinyint,

`phone\_number` int,

`experience` tinyint

);

CREATE TABLE `Way` (

`way\_id` int PRIMARY KEY AUTO\_INCREMENT,

`start\_city` varchar(60),

`dest\_city` varchar(60),

`stops` boolean

);

CREATE TABLE `Stops` (

`stops\_id` int PRIMARY KEY AUTO\_INCREMENT,

`way\_id` int,

`city` varchar(60),

`stop\_que` tinyint

);

ALTER TABLE `Order` ADD FOREIGN KEY (`client\_id`) REFERENCES `Client` (`client\_id`) ON DELETE NO ACTION ON UPDATE CASCADE;

ALTER TABLE `Order` ADD FOREIGN KEY (`car\_id`) REFERENCES `Car` (`car\_id`) ON DELETE NO ACTION ON UPDATE CASCADE;

ALTER TABLE `Basket` ADD FOREIGN KEY (`order\_id`) REFERENCES `Order` (`order\_id`) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `Basket` ADD FOREIGN KEY (`product\_id`) REFERENCES `Product` (`product\_id`) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `Comment` ADD FOREIGN KEY (`product\_id`) REFERENCES `Product` (`product\_id`) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE `Car` ADD FOREIGN KEY (`driver\_id`) REFERENCES `Driver` (`driver\_id`) ON DELETE NO ACTION ON UPDATE CASCADE;

ALTER TABLE `Car` ADD FOREIGN KEY (`way\_id`) REFERENCES `Way` (`way\_id`) ON DELETE NO ACTION ON UPDATE CASCADE;

ALTER TABLE `Stops` ADD FOREIGN KEY (`way\_id`) REFERENCES `Way` (`way\_id`) ON DELETE CASCADE ON UPDATE CASCADE;

**Висновок:** на цій лабораторній роботі було завершено моделювання і засобами SQL створено базу даних, що складається з восьми таблиць.