

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:

Якимішин Х.М.

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
2. Вибір бази даних для подальшої роботи:
USE DATABASE ім'я_бази
3. Перегляд таблиць в базі даних:
SHOW TABLES [FOR ім'я_бази]
4. Перегляд опису таблиці в базі:
DESCRIBE ім'я_таблиці
5. Виконати набір команд з зовнішнього файлу:
SOURCE назва_файлу
6. 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;

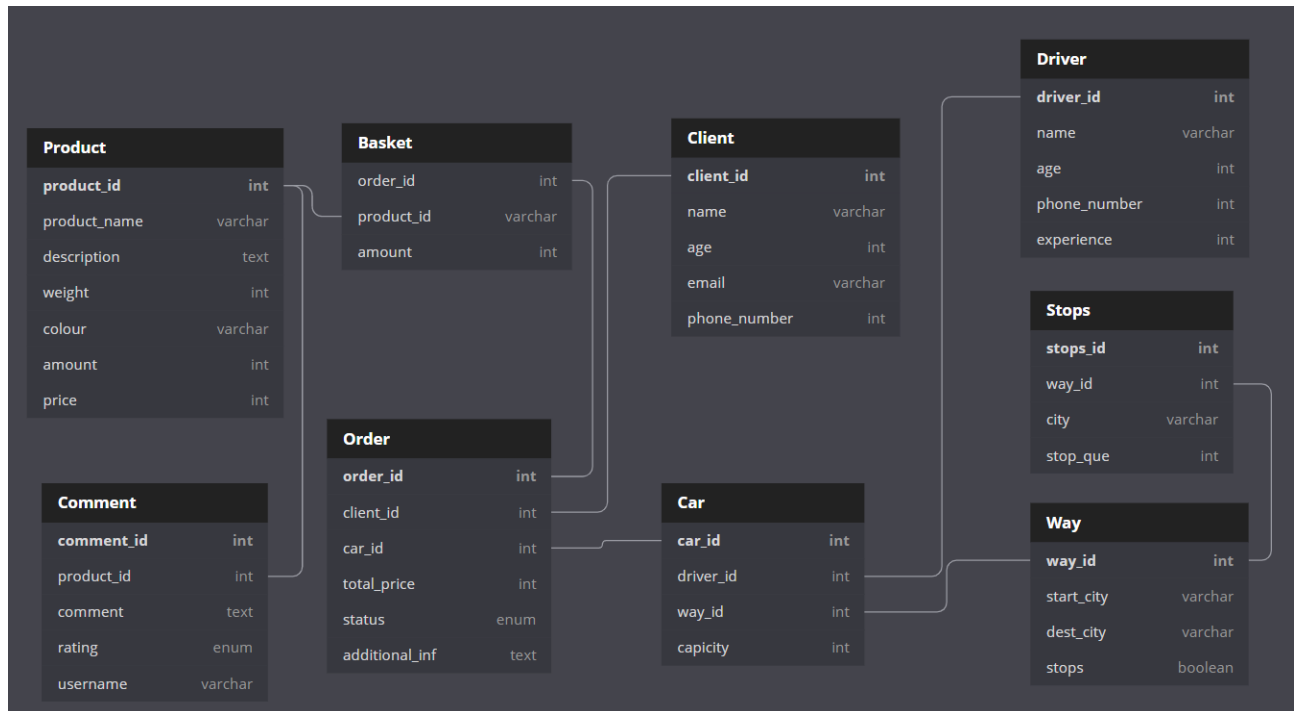
CREATE INDEX – створення нового індексу (для швидкого пошуку даних);

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,  
  `capacity` 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.