

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»  
Інститут комп'ютерних наук та інформаційних технологій

Кафедра «Системи штучного інтелекту»



**Лабораторна робота №2**  
**З дисципліни:**  
**«Організація баз даних та знань»**

Виконав:  
Студент групи КН-208  
Лукаш О.В.  
Викладач:  
Мельникова Н.І.

## Тема: 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;

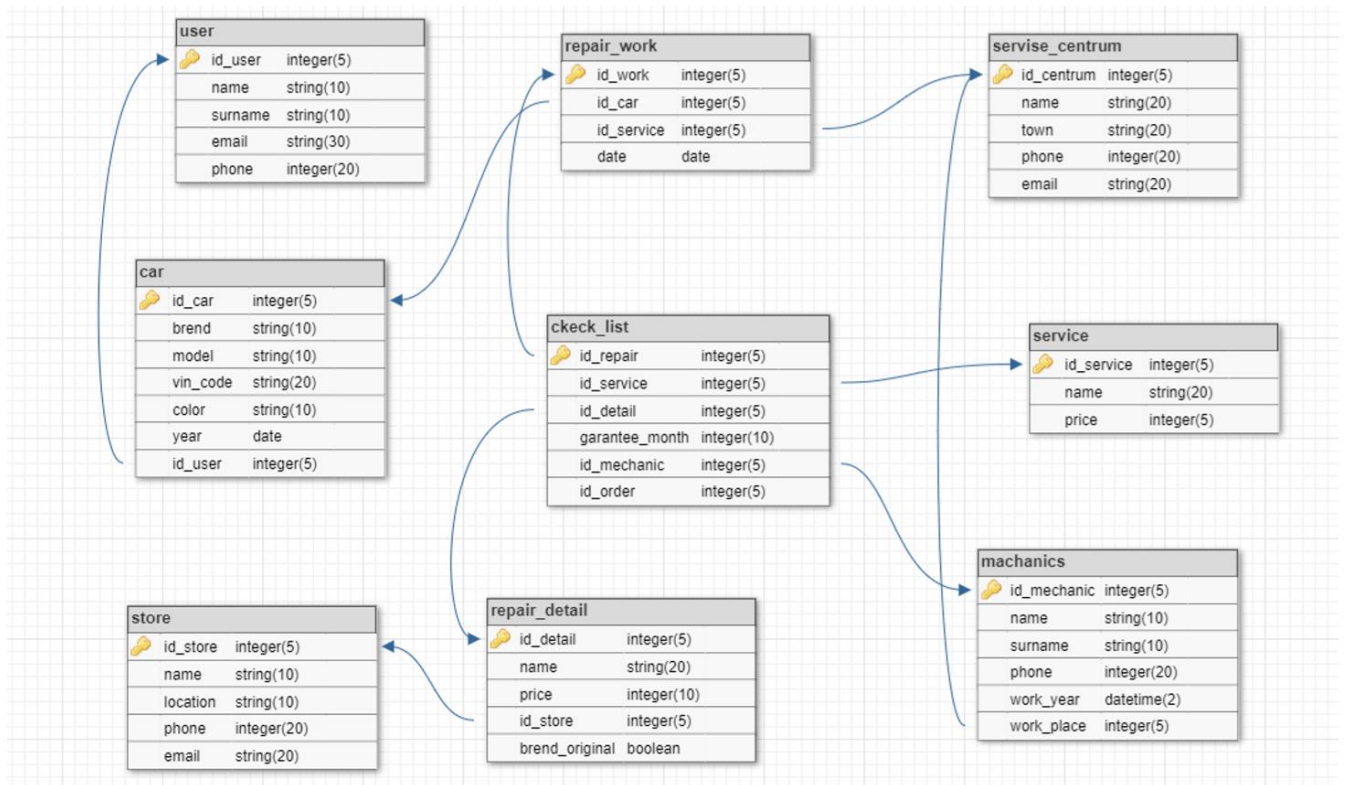
**CREATE INDEX** – створення нового індексу (для швидкого пошуку даних);

**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.



Створимо нову базу даних, виконавши такі команди:

```
CREATE DATABASE car_servise;
USE car_servise;
```

```
CREATE TABLE `user` (
  `id_user` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(10) NOT NULL,
  `surname` VARCHAR(10) NOT NULL,
  `email` VARCHAR(20) UNIQUE NOT NULL,
  `phone` VARCHAR(10) UNIQUE NOT NULL,
  `age` DATE NOT NULL,
  PRIMARY KEY(`id_user`)
);
```

```
CREATE TABLE `car` (
  `id_car` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `brend` VARCHAR(10) NOT NULL,
  `model` VARCHAR(20) NOT NULL,
  `vin_code` VARCHAR(20) UNIQUE NOT NULL,
  `color` VARCHAR(10) NOT NULL,
  `year` INT NOT NULL,
  `id_user` INT UNSIGNED NOT NULL,
  PRIMARY KEY(`id_car`),
  CONSTRAINT `user_fk` FOREIGN KEY(`id_user`)
  REFERENCES `user`(`id_user`) ON DELETE NO ACTION ON UPDATE NO ACTION
);
```

```
CREATE TABLE `repair_work` (
  `id_work` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `id_car` INT UNSIGNED NOT NULL,
  `id_centrum` INT UNSIGNED NOT NULL,
  `status` ENUM ( 'not_started', 'in_progres', 'completed') DEFAULT 'not_started',
  `price` INT UNSIGNED NOT NULL,
  PRIMARY KEY(`id_work`),
  CONSTRAINT `service_centrum_fk` FOREIGN KEY(`id_centrum`)
```

```

REFERENCES `service_centrum`(`id_centrum`) ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT `car_fk` FOREIGN KEY(`id_car`)
REFERENCES `car`(`id_car`) ON DELETE NO ACTION ON UPDATE NO ACTION
);

CREATE TABLE `service_centrum` (
`id_centrum` INT UNSIGNED NOT NULL AUTO_INCREMENT,
`name` VARCHAR(30) NOT NULL,
`towm` VARCHAR(20) NOT NULL,
`phone` VARCHAR(10) UNIQUE NOT NULL,
`email` VARCHAR(20) UNIQUE NOT NULL,
PRIMARY KEY(`id_centrum`)
);

CREATE TABLE `mechanic` (
`id_mechanic` INT UNSIGNED NOT NULL AUTO_INCREMENT,
`name` VARCHAR(10) NOT NULL,
`surname` VARCHAR(10) NOT NULL,
`phone` VARCHAR(10) UNIQUE NOT NULL,
`work_year` DATE NOT NULL,
`work_place` INT UNSIGNED NOT NULL,
PRIMARY KEY(`id_mechanic`),
CONSTRAINT `service_centrum_fk0` FOREIGN KEY(`work_place`)
REFERENCES `service_centrum`(`id_centrum`) ON DELETE NO ACTION ON UPDATE NO ACTION
);

CREATE TABLE `check_list` (
`id_repair` INT UNSIGNED NOT NULL AUTO_INCREMENT,
`id_service` INT UNSIGNED NOT NULL,
`garante_month` INT UNSIGNED NOT NULL,
`id_detail` INT UNSIGNED,
`id_mechanic` INT UNSIGNED NOT NULL,
PRIMARY KEY(`id_repair`),
CONSTRAINT `repair_work_fk` FOREIGN KEY(`id_repair`)
REFERENCES `repair_work`(`id_work`) ON DELETE NO ACTION ON UPDATE NO ACTION ,
CONSTRAINT `service_fk` FOREIGN KEY(`id_service`)
REFERENCES `service`(`id_service`) ON DELETE NO ACTION ON UPDATE NO ACTION ,
CONSTRAINT `repair_detail_fk` FOREIGN KEY(`id_detail`)
REFERENCES `repair_detail`(`id_detail`) ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT `mechanic_fk` FOREIGN KEY(`id_mechanic`)
REFERENCES `mechanic`(`id_mechanic`) ON DELETE NO ACTION ON UPDATE NO ACTION
);

CREATE TABLE `service` (
`id_service` INT UNSIGNED NOT NULL AUTO_INCREMENT,
`name` VARCHAR(20) NOT NULL,
`price` INT UNSIGNED NOT NULL,
PRIMARY KEY(`id_service`)
);

CREATE TABLE `repair_detail` (
`id_detail` INT UNSIGNED NOT NULL AUTO_INCREMENT,
`name` VARCHAR(20) NOT NULL,
`price` INT UNSIGNED NOT NULL,
`id_store` INT UNSIGNED NOT NULL,
`brend_original` BOOL NOT NULL,
PRIMARY KEY(`id_detail`),
CONSTRAINT `store_fk` FOREIGN KEY(`id_store`)
REFERENCES `store`(`id_store`)
);

CREATE TABLE `store` (
`id_store` INT UNSIGNED NOT NULL AUTO_INCREMENT,
`name` INT UNSIGNED NOT NULL,
`phone` VARCHAR(10) UNIQUE NOT NULL,
`email` VARCHAR(50) UNIQUE NOT NULL,
PRIMARY KEY(`id_store`)
);

```

**Висновок:** на цій лабораторній роботі було завершено моделювання і засобами SQL створено бази даних, що складається з дев'яти таблиць.