

МІНІСТЕРСТВО ОСВІТИ І НАУКИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА  
ПОЛІТЕХНІКА»

Кафедра систем штучного інтелекту



Лабораторна робота №2  
З курсу “Організація баз даних та знань”

Виконав:  
студент групи КН-210  
Марій Павло

Викладач:  
Мельникова Наталя Іванівна

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

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

### Теоретичні відомості

Щоб створити нову базу даних у командному рядку клієнта MySQL (mysql.exe) слід виконати команду **CREATE DATABASE**.

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.

1. Слід використовувати якомога менший тип даних для полів таблиць.

Наприклад, для зберігання чисел від 1 до 64 краще використати тип

TINYINT(6) замість SMALLINT. Це впливає на швидкість пошуку і вибірки даних.

2. Слід використовувати рядки фіксованої довжини, якщо це можливо. Для цього всі поля таблиці повинні бути фіксованої довжини. Тобто, варто уникати типів VARCHAR, TEXT і BLOB . Це пришвидшить вибірку даних з середини рядків, оскільки ці дані будуть мати фіксовану адресу. При потребі використання полів з типами TEXT або BLOB, їх можна виділити в окрему таблицю.
3. Якщо можливо, варто завжди використовувати поля з обмеженням NOT NULL . Хоча це може збільшувати об'єм бази на диску.
4. MySQL дозволяє використовувати різні типи таблиць в одній базі даних. Слід використовувати переваги різних типів (MyISAM, INODB тощо) залежно від характеру майбутнього використання таблиці.
5. Потрібно створювати індекси, які пришвидшать пошук і вибірку даних.
6. В рідкісних випадках можна денормалізувати схему з метою зменшення кількості операцій з об'єднання таблиць при складних запитах. Але при цьому ускладнюється задача збереження цілісності бази даних.

## Хід роботи

Згідно з моделлю бази даних з попередньої лабораторної роботи, я написав SQL-скрипт, який створює цю базу даних в середовищі MySQL.

Код скрипту:

```
-----
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DAT
E,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-----

-- Schema mydb
-----

-----

-- Schema mydb
-----

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;

-----

-- Table `mydb`.`user`
-----

CREATE TABLE IF NOT EXISTS `mydb`.`user` (
```

```

`id` INT NOT NULL AUTO_INCREMENT,
`name` VARCHAR(45) NOT NULL,
`surname` VARCHAR(45) NOT NULL,
`telephone_number` VARCHAR(12) NULL,
`email` VARCHAR(40) NULL,
PRIMARY KEY (`id`),
UNIQUE INDEX `id_UNIQUE` (`id` ASC) )
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`type`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`type` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `type` VARCHAR(45) NOT NULL,
  `description` VARCHAR(200) NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) )
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`mark`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`mark` (
  `id` INT NOT NULL ,
  `name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) )
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`model`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`model` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NOT NULL,
  `country` VARCHAR(45) NOT NULL,
  `mark_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC),
  INDEX `fk_model_mark_idx` (`mark_id` ASC) ,
  CONSTRAINT `fk_model_mark`
    FOREIGN KEY (`mark_id`)
      REFERENCES `mydb`.`mark` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

-----  
-- Table `mydb`.`auto`  
-----

```
CREATE TABLE IF NOT EXISTS `mydb`.`auto` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `volume_engine` FLOAT NOT NULL,  
  `volume_power` INT NOT NULL,  
  `transmission` VARCHAR(45) NOT NULL,  
  `type_id` INT NOT NULL,  
  `model_id` INT NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE INDEX `id_UNIQUE` (`id` ASC),  
  INDEX `fk_auto_type1_idx` (`type_id` ASC),  
  INDEX `fk_auto_model1_idx` (`model_id` ASC),  
  CONSTRAINT `fk_auto_type1`  
    FOREIGN KEY (`type_id`)  
      REFERENCES `mydb`.`type` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `fk_auto_model1`  
    FOREIGN KEY (`model_id`)  
      REFERENCES `mydb`.`model` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-----  
-- Table `mydb`.`advertisement`  
-----

```
CREATE TABLE IF NOT EXISTS `mydb`.`advertisement` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `price` INT NULL,  
  `mileage` INT NULL,  
  `color` VARCHAR(15) NULL,  
  `condition` VARCHAR(20) NULL,  
  `pubdate` DATETIME NOT NULL,  
  `address` VARCHAR(200) NULL,  
  `year_auto` INT NULL,  
  `text` VARCHAR(400) NULL,  
  `auto_id` INT NOT NULL,  
  `user_id` INT NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE INDEX `id_UNIQUE` (`id` ASC),  
  INDEX `fk_advertisement_auto1_idx` (`auto_id` ASC),  
  INDEX `fk_advertisement_user1_idx` (`user_id` ASC),  
  CONSTRAINT `fk_advertisement_auto1`  
    FOREIGN KEY (`auto_id`)
```

```

REFERENCES `mydb`.`auto` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_advertisement_user1`
FOREIGN KEY (`user_id`)
REFERENCES `mydb`.`user` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`comment`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`comment` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `text` VARCHAR(200) NOT NULL,
  `advertisement_id` INT NOT NULL,
  `user_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC),
  INDEX `fk_comment_advertisement1_idx` (`advertisement_id` ASC),
  INDEX `fk_comment_user1_idx` (`user_id` ASC),
  CONSTRAINT `fk_comment_advertisement1`
    FOREIGN KEY (`advertisement_id`)
    REFERENCES `mydb`.`advertisement` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_comment_user1`
    FOREIGN KEY (`user_id`)
    REFERENCES `mydb`.`user` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`advertisement_user`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`advertisement_user` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `advertisement_id` INT NOT NULL,
  `user_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `id_UNIQUE` (`id` ASC),
  INDEX `fk_advertisement_user_user1_idx` (`user_id` ASC),
  INDEX `fk_advertisement_user_advertisement1_idx` (`advertisement_id` ASC),
  CONSTRAINT `fk_advertisement_user_advertisement1`
    FOREIGN KEY (`advertisement_id`)
    REFERENCES `mydb`.`advertisement` (`id`)

```

```

ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_advertisement_user_user1`
FOREIGN KEY (`user_id`)
REFERENCES `mydb`.`user` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

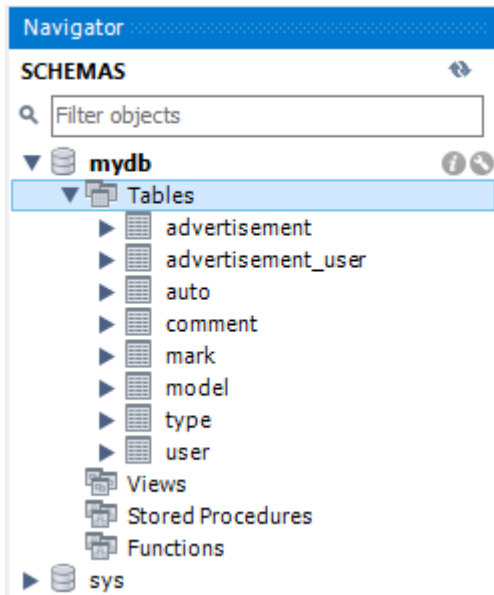
Команди SET на початку і в кінці скрипта встановлюють певні параметри: не перевіряти унікальність значень певних полів, не перевіряти відповідність зовнішніх ключів, не дозволяти нулі в полях дати, а також нульову дату, викликати помилку при діленні на нуль.

Командою CREATE SCHEMA створюю нову базу даних. Командою CREATE TABLE створюю нові таблиці. Командою PRIMARY KEY вказую на первинні ключі таблиці. Командою CONSTRAINT вказую на існування зв'язків між таблицями. Командою FOREIGN KEY вказую на зовнішній ключ. Командою REFERENCES вказую на поле таблиці, якої стосується зовнішній ключ. Командою ON DELETE і ON UPDATE задаю, як має поводитись зовнішній ключ при видаленні чи оновленні поля, якого цей ключ стосується. Командою ENGINE вказую на тип таблиці.

SQL-скрипт виконується правильно, без помилок:

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
✓ 1	11:16:43	SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0	0 row(s) affected	0.000 sec
✓ 2	11:16:43	SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY...	0 row(s) affected	0.000 sec
✓ 3	11:16:43	SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE=ONLY_FULL_GROUP_BY,S...	0 row(s) affected	0.016 sec
⚠ 4	11:16:43	CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8	1 row(s) affected, 1 warning(s): 3719 'utf8' is currently an alias for the character set UTF8...	0.188 sec
✓ 5	11:16:44	USE `mydb`	0 row(s) affected	0.000 sec
✓ 6	11:16:44	CREATE TABLE IF NOT EXISTS `mydb`.`user` ( `id` INT NOT NULL AUTO_INCREME...	0 row(s) affected	1.328 sec
✓ 7	11:16:45	CREATE TABLE IF NOT EXISTS `mydb`.`type` ( `id` INT NOT NULL AUTO_INCREME...	0 row(s) affected	0.641 sec
✓ 8	11:16:46	CREATE TABLE IF NOT EXISTS `mydb`.`mark` ( `id` INT NOT NULL , `name` VARCH...	0 row(s) affected	0.625 sec
✓ 9	11:16:46	CREATE TABLE IF NOT EXISTS `mydb`.`model` ( `id` INT NOT NULL AUTO_INCREM...	0 row(s) affected	0.859 sec
✓ 10	11:16:47	CREATE TABLE IF NOT EXISTS `mydb`.`auto` ( `id` INT NOT NULL AUTO_INCREME...	0 row(s) affected	0.719 sec
✓ 11	11:16:48	CREATE TABLE IF NOT EXISTS `mydb`.`advertisement` ( `id` INT NOT NULL AUTO_I...	0 row(s) affected	0.922 sec
✓ 12	11:16:49	CREATE TABLE IF NOT EXISTS `mydb`.`comment` ( `id` INT NOT NULL AUTO_INCR...	0 row(s) affected	0.922 sec
✓ 13	11:16:50	CREATE TABLE IF NOT EXISTS `mydb`.`advertisement_user` ( `id` INT NOT NULL AU...	0 row(s) affected	0.625 sec
⚠ 14	11:16:50	SET SQL_MODE=@OLD_SQL_MODE	0 row(s) affected, 1 warning(s): 3135 'NO_ZERO_DATE', 'NO_ZERO_IN_DATE' and 'ER...	0.000 sec
✓ 15	11:16:51	SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS	0 row(s) affected	0.016 sec
✓ 16	11:16:51	SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS	0 row(s) affected	0.000 sec

База даних “mydb” створена, в ній присутні всі задані таблиці:



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