

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту



Звіт до лабораторної роботи №2

з дисципліни
“ОБДЗ”

Виконав:

ст. гр. КН-211

Ткачук Орест

Викладач:

Якимишин Х.М.

Лабораторна робота №2

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 (koі8u, 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

Створимо нову базу даних, виконавши такі команди:

```
CREATE TABLE `Theater` (
  `id_theater` INT NOT NULL AUTO_INCREMENT,
  `theater_name` VARCHAR(100) NOT NULL,
  `address` VARCHAR(100) NOT NULL,
  `city` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`id_theater`)
);

CREATE TABLE `Hall` (
  `id_hall` INT NOT NULL AUTO_INCREMENT,
  `theaterID` INT NOT NULL,
  `hall_name` VARCHAR(100) NOT NULL,
  `capacity` INT NOT NULL,
  PRIMARY KEY (`id_hall`),
  CONSTRAINT `Hall_fk0` FOREIGN KEY (`theaterID`) REFERENCES
`Theater`(`id_theater`) ON DELETE CASCADE ON UPDATE NO ACTION
);

CREATE TABLE `Repertoire` (
  `id_repertoire` INT NOT NULL AUTO_INCREMENT,
  `performance_name` VARCHAR(100) NOT NULL,
  `length` TIME NOT NULL,
  `description` TEXT NOT NULL,
  `director` VARCHAR(50),
  `artist` VARCHAR(50),
  PRIMARY KEY (`id_repertoire`)
);

CREATE TABLE `Staff` (
  `id_staff` INT NOT NULL AUTO_INCREMENT,
  `position` VARCHAR(50) NOT NULL,
  `staff_name` VARCHAR(100) NOT NULL,
  `salary` INT NOT NULL,
  `birthday` DATE NOT NULL,
  `email` VARCHAR(100) NOT NULL,
  `phone` VARCHAR(13) NOT NULL,
  `theaterID` INT NOT NULL,
  PRIMARY KEY (`id_staff`),
  CONSTRAINT `Staff_fk0` FOREIGN KEY (`theaterID`) REFERENCES
`Theater`(`id_theater`) ON DELETE CASCADE ON UPDATE NO ACTION
);

CREATE TABLE `Customer` (
  `id_customer` INT NOT NULL AUTO_INCREMENT,
  `customer_name` VARCHAR(100) NOT NULL,
  `customer_email` VARCHAR(100),
  `customer_mobile` VARCHAR(13) NOT NULL,
  PRIMARY KEY (`id_customer`)
);

CREATE TABLE `Seance` (
  `id_seance` INT NOT NULL AUTO_INCREMENT,
  `date` DATE NOT NULL,
  `time` TIME NOT NULL,
  `hallID` INT NOT NULL,
  `repertoireID` INT NOT NULL,
  PRIMARY KEY (`id_seance`),
  CONSTRAINT `Seance_fk0` FOREIGN KEY (`hallID`) REFERENCES
`Hall`(`id_hall`) ON DELETE CASCADE ON UPDATE NO ACTION,
  CONSTRAINT `Seance_fk1` FOREIGN KEY (`repertoireID`) REFERENCES
`Repertoire`(`id_repertoire`) ON DELETE CASCADE ON UPDATE NO ACTION
);
```

```

CREATE TABLE `Ticket` (
  `id_ticket` INT NOT NULL AUTO_INCREMENT,
  `seanceID` INT NOT NULL,
  `customerID` INT,
  `type` VARCHAR(20) NOT NULL,
  `place` INT(200) NOT NULL UNIQUE,
  `price` INT NOT NULL,
  `isReserved` BINARY NOT NULL,
  PRIMARY KEY (`id_ticket`),
  CONSTRAINT `Ticket_fk0` FOREIGN KEY (`seanceID`) REFERENCES
`Seance`(`id_seance`) ON DELETE CASCADE ON UPDATE NO ACTION,
  CONSTRAINT `Ticket_fk1` FOREIGN KEY (`customerID`) REFERENCES
`Customer`(`id_customer`) ON DELETE CASCADE ON UPDATE NO ACTION
);

```

```

CREATE TABLE `Role` (
  `id_role` INT NOT NULL AUTO_INCREMENT,
  `role_name` VARCHAR(255) NOT NULL,
  `role_description` TEXT NOT NULL,
  `repertoireID` INT NOT NULL,
  PRIMARY KEY (`id_role`),
  CONSTRAINT `Role_fk0` FOREIGN KEY (`repertoireID`) REFERENCES
`Repertoire`(`id_repertoire`) ON DELETE CASCADE ON UPDATE NO ACTION
);

```

```

CREATE TABLE `ActorRole` (
  `id_act_role` INT NOT NULL AUTO_INCREMENT,
  `seanceID` INT NOT NULL,
  `roleID` INT NOT NULL,
  `staffID` INT NOT NULL,
  PRIMARY KEY (`id_act_role`),
  CONSTRAINT `ActorRole_fk0` FOREIGN KEY (`seanceID`) REFERENCES
`Seance`(`id_seance`) ON DELETE CASCADE ON UPDATE NO ACTION,
  CONSTRAINT `ActorRole_fk1` FOREIGN KEY (`roleID`) REFERENCES
`Role`(`id_role`) ON DELETE CASCADE ON UPDATE NO ACTION,
  CONSTRAINT `ActorRole_fk2` FOREIGN KEY (`staffID`) REFERENCES
`Staff`(`id_staff`) ON DELETE CASCADE ON UPDATE NO ACTION
);

```

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