

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”  
ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ

Кафедра систем штучного інтелекту

Лабораторна робота №2  
із дисципліни  
Бази даних

Виконала:  
Ст. групи КН-207  
Ювченко Ю.  
Прийняла:  
Мельникова Н.І.

Львів – 2019 р.

**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;

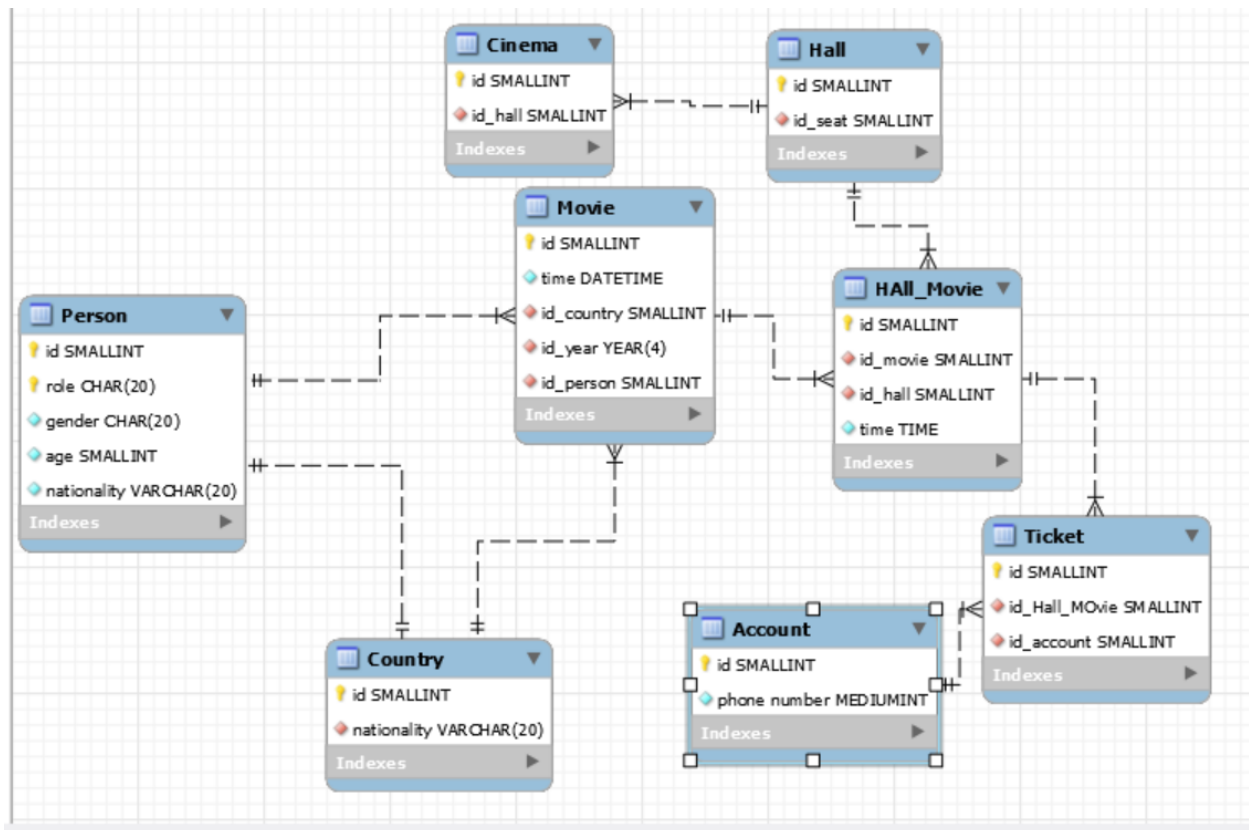
CREATE INDEX – створення нового індексу (для швидкого пошуку даних);

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 проєктованої бази даних. Оскільки дані всіх параметрів виданого паспорту не можна змінювати то на всіх зв'язках таблиць встановлено обмеження цілісності «Restrict».



Створимо нову базу даних, виконавши такі команди:

```
CREATE DATABASE `mydb`;
```

```
USE `mydb` ;
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Hall` (  
  `id` SMALLINT NOT NULL,  
  `id_seat` SMALLINT NOT NULL,  
  PRIMARY KEY (`id`),  
  INDEX `Hall_fk0` (`id_seat` ASC) VISIBLE,  
  CONSTRAINT `Hall_fk0`  
    FOREIGN KEY (`id_seat`)  
    REFERENCES `mydb`.`Seat` (`id`));
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Cinema` (  
  `id` SMALLINT NOT NULL ,  
  `id_hall` SMALLINT NOT NULL,  
  PRIMARY KEY (`id`),  
  INDEX `Cinema_fk0` (`id_hall` ASC) VISIBLE,  
  CONSTRAINT `Cinema_fk0`  
    FOREIGN KEY (`id_hall`)  
    REFERENCES `mydb`.`Hall` (`id`)  
    ON DELETE RESTRICT  
    ON UPDATE CASCADE);
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Person` (  
  `id` SMALLINT NOT NULL,  
  `role` CHAR(20) NOT NULL,  
  `gender` CHAR(20) NOT NULL,  
  `age` SMALLINT NOT NULL,  
  `nationality` VARCHAR(20) NOT NULL,  
  PRIMARY KEY (`id`, `role`));
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Country` (  
  `id` SMALLINT NOT NULL AUTO_INCREMENT,  
  `nationality` VARCHAR(20) NOT NULL,  
  PRIMARY KEY (`id`),  
  INDEX `Country_fk0` (`nationality` ASC) VISIBLE,  
  CONSTRAINT `Country_fk0`  
    FOREIGN KEY (`nationality`)  
    REFERENCES `mydb`.`Person` (`nationality`));
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Movie` (  
  `id` SMALLINT NOT NULL,  
  `time` DATETIME NOT NULL,  
  `id_country` SMALLINT NOT NULL,  
  `id_year` YEAR(4) NOT NULL,  
  `id_person` SMALLINT NOT NULL,  
  PRIMARY KEY (`id`),  
  CONSTRAINT `Movie_fk2`  
    FOREIGN KEY (`id_year`)  
    REFERENCES `mydb`.`Year` (`id`),
```

```

CONSTRAINT `Movie_fk3`
  FOREIGN KEY (`id_person`)
  REFERENCES `mydb`.`Person` (`id`),
CONSTRAINT `Movie_fk1`
  FOREIGN KEY (`id_country`)
  REFERENCES `mydb`.`Country` (`id`)
  ON DELETE RESTRICT
  ON UPDATE RESTRICT);

CREATE TABLE IF NOT EXISTS `mydb`.`Account` (
  `id` SMALLINT NOT NULL,
  `phone number` MEDIUMINT NOT NULL,
  PRIMARY KEY (`id`);

CREATE TABLE IF NOT EXISTS `mydb`.`Hall_Movie` (
  `id` SMALLINT NOT NULL,
  `id_movie` SMALLINT NOT NULL,
  `id_hall` SMALLINT NOT NULL,
  `time` TIME NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `Hall_Movie_fk0`
    FOREIGN KEY (`id_movie`)
    REFERENCES `mydb`.`Movie` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `Hall_Movie_fk1`
    FOREIGN KEY (`id_hall`)
    REFERENCES `mydb`.`Hall` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE);

CREATE TABLE IF NOT EXISTS `mydb`.`Ticket` (
  `id` SMALLINT NOT NULL,
  `id_Hall_Movie` SMALLINT NOT NULL,
  `id_account` SMALLINT NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `Ticket_fk0`
    FOREIGN KEY (`id_Hall_Movie`)
    REFERENCES `mydb`.`Hall_Movie` (`id`),
  CONSTRAINT `Ticket_fk1`
    FOREIGN KEY (`id_account`)
    REFERENCES `mydb`.`Account` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE);

```

**Висновок:** на цій лабораторній роботі було завершено моделювання і засобами SQL створено базу даних, що складається з 11 таблиць.