

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

Виконав:
Ст. групи КН-207
Букін С.А.
Прийняв:
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;

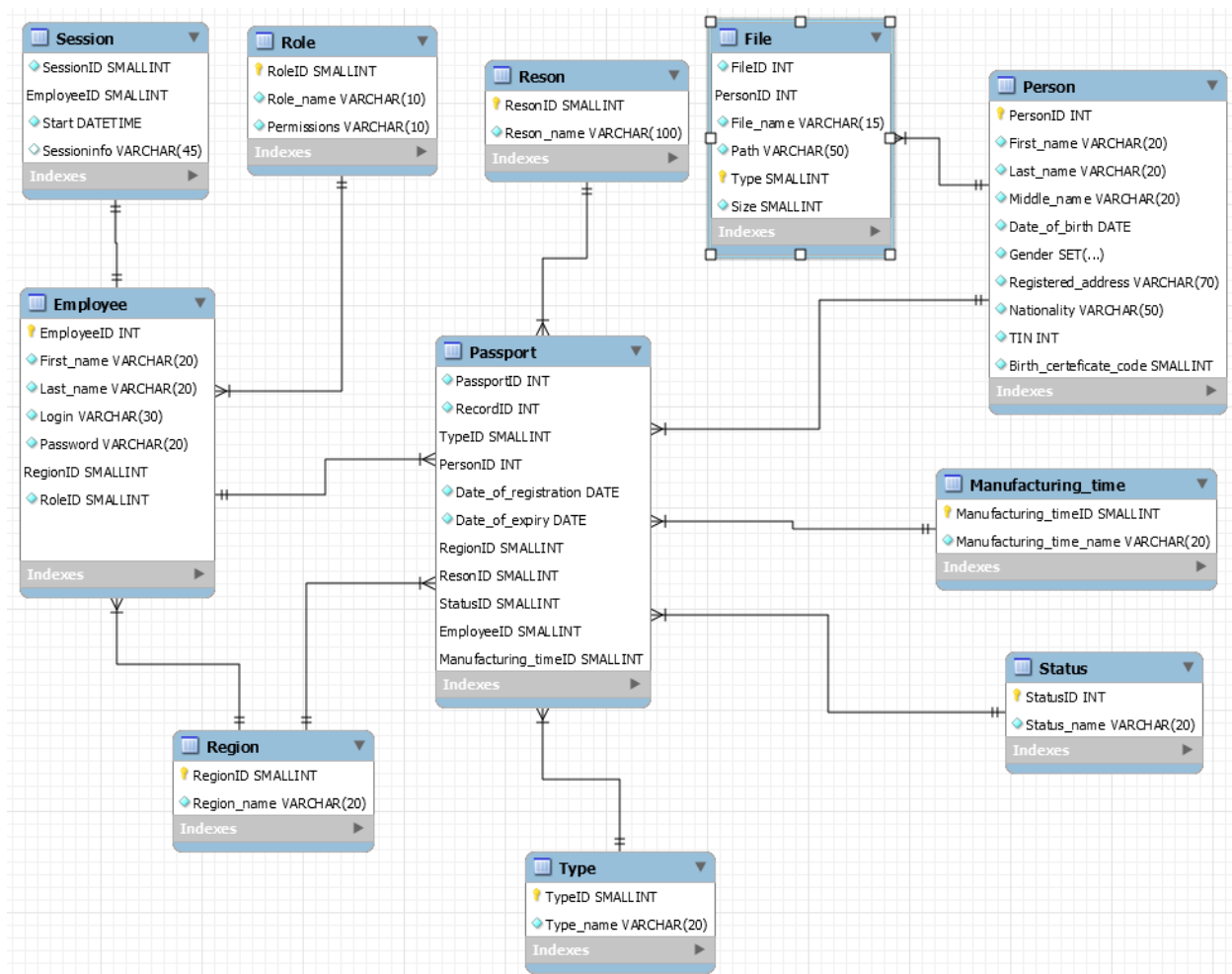
CREATE INDEX – створення нового індексу (для швидкого пошуку даних);

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 passportdb;
```

```
USE passportdb;
```

```
CREATE TABLE IF NOT EXISTS `passportdb`.`Region` (  
  `RegionID` SMALLINT NOT NULL,  
  `Region_name` VARCHAR(20) NOT NULL,  
  PRIMARY KEY (`RegionID`));
```

```
CREATE TABLE IF NOT EXISTS `passportdb`.`Role` (  
  `RoleID` SMALLINT NOT NULL,  
  `Role_name` VARCHAR(10) NOT NULL,  
  `Permissions` VARCHAR(70) NOT NULL,  
  PRIMARY KEY (`RoleID`));
```

```
CREATE TABLE IF NOT EXISTS `passportdb`.`Employee` (  
  `EmployeeID` SMALLINT NOT NULL AUTO_INCREMENT,  
  `First_name` VARCHAR(20) NOT NULL,  
  `Last_name` VARCHAR(20) NOT NULL,  
  `Login` VARCHAR(30) NOT NULL,  
  `Password` VARCHAR(20) NOT NULL,  
  `RegionID` SMALLINT NOT NULL,  
  `RoleID` SMALLINT NOT NULL,  
  PRIMARY KEY (`EmployeeID`, `RegionID`, `RoleID`),
```

```
UNIQUE INDEX `Login_UNIQUE` (`Login` ASC) VISIBLE,  
INDEX `RoleID_fk_idx` (`RoleID` ASC) VISIBLE,  
CONSTRAINT `RegionID_fk1`  
    FOREIGN KEY (`RegionID`)  
    REFERENCES `passportdb`.`Region` (`RegionID`)  
    ON DELETE RESTRICT  
    ON UPDATE RESTRICT,  
CONSTRAINT `RoleID_fk`  
    FOREIGN KEY (`RoleID`)  
    REFERENCES `passportdb`.`Role` (`RoleID`)  
    ON DELETE RESTRICT  
    ON UPDATE RESTRICT);
```

```
CREATE TABLE IF NOT EXISTS `passportdb`.`Session` (  
    `SessionID` INT NOT NULL,  
    `EmployeeID` SMALLINT NOT NULL,  
    `Start` DATETIME NOT NULL,  
    `Sessioninfo` VARCHAR(45) NULL,  
    INDEX `EmployeeID_fk_idx` (`EmployeeID` ASC) VISIBLE,  
    PRIMARY KEY (`SessionID`, `EmployeeID`),  
    CONSTRAINT `EmployeeID_fk1`  
        FOREIGN KEY (`EmployeeID`)  
        REFERENCES `passportdb`.`Employee` (`EmployeeID`)  
        ON DELETE CASCADE
```

ON UPDATE CASCADE);

```
CREATE TABLE IF NOT EXISTS `passportdb`.`Person` (  
  `PersonID` INT NOT NULL,  
  `First_name` VARCHAR(20) NOT NULL,  
  `Last_name` VARCHAR(20) NOT NULL,  
  `Middle_name` VARCHAR(20) NOT NULL,  
  `Date_of_birth` DATE NOT NULL,  
  `Gender` SET('Male', 'Female') NOT NULL,  
  `Registered_address` VARCHAR(70) NOT NULL,  
  `Nationality` VARCHAR(50) NOT NULL,  
  `TIN` INT UNSIGNED NOT NULL,  
  `Birth_certificate_code` SMALLINT NOT NULL,  
  PRIMARY KEY (`PersonID`),  
  UNIQUE INDEX `TIN_UNIQUE` (`TIN` ASC) VISIBLE,  
  UNIQUE INDEX `Birth_certificate_code_UNIQUE` (`Birth_certificate_code` ASC)  
  VISIBLE);
```

```
CREATE TABLE IF NOT EXISTS `passportdb`.`File` (  
  `FileID` INT NOT NULL,  
  `PersonID` INT NOT NULL,  
  `File_name` VARCHAR(15) NOT NULL,  
  `Path` VARCHAR(50) NOT NULL,
```

```
`Type` SMALLINT NOT NULL,  
`Size` SMALLINT NOT NULL,  
INDEX `PersonID_fk_idx` (`PersonID` ASC) VISIBLE,  
PRIMARY KEY (`PersonID`, `Path`),  
CONSTRAINT `PersonID_fk1`  
FOREIGN KEY (`PersonID`)  
REFERENCES `passportdb`.`Person` (`PersonID`)  
ON DELETE RESTRICT  
ON UPDATE RESTRICT);
```

```
CREATE TABLE IF NOT EXISTS `passportdb`.`Type` (  
`TypeID` SMALLINT NOT NULL,  
`Type_name` VARCHAR(20) NOT NULL,  
PRIMARY KEY (`TypeID`));
```

```
CREATE TABLE IF NOT EXISTS `passportdb`.`Reson` (  
`ResonID` SMALLINT NOT NULL,  
`Reson_name` VARCHAR(100) NOT NULL,  
PRIMARY KEY (`ResonID`));
```

```
CREATE TABLE IF NOT EXISTS `passportdb`.`Manufacturing_time` (  
`Manufacturing_timeID` SMALLINT NOT NULL,
```

```
`Manufacturing_time_name` VARCHAR(20) NOT NULL,  
PRIMARY KEY (`Manufacturing_timeID`));
```

```
CREATE TABLE IF NOT EXISTS `passportdb`.`Status` (  
  `StatusID` SMALLINT NOT NULL,  
  `Status_name` VARCHAR(20) NOT NULL,  
  PRIMARY KEY (`StatusID`));
```

```
CREATE TABLE IF NOT EXISTS `passportdb`.`Passport` (  
  `PassportID` INT NOT NULL,  
  `RecordID` INT UNSIGNED NOT NULL,  
  `TypeID` SMALLINT NOT NULL,  
  `PersonID` INT NOT NULL,  
  `Date_of_registration` DATE NOT NULL,  
  `Date_of_expiry` DATE NOT NULL,  
  `RegionID` SMALLINT NOT NULL,  
  `ResonID` SMALLINT NOT NULL,  
  `StatusID` SMALLINT NOT NULL,  
  `EmployeeID` SMALLINT NOT NULL,  
  `Manufacturing_timeID` SMALLINT NOT NULL,  
  INDEX `ResonID_fk_idx` (`ResonID` ASC) VISIBLE,  
  INDEX `Manufacturing_timeID_fk_idx` (`Manufacturing_timeID` ASC) VISIBLE,  
  INDEX `RegionID_fk_idx` (`RegionID` ASC) VISIBLE,
```



```

INDEX `EmployeeID_fk_idx` (`EmployeeID` ASC) VISIBLE,

PRIMARY KEY (`EmployeeID`, `RegionID`, `Manufacturing_timeID`, `PersonID`,
`StatusID`, `TypeID`, `ResonID`),

INDEX `StatusID_fk_idx` (`StatusID` ASC) VISIBLE,

UNIQUE INDEX `RecordID_UNIQUE` (`RecordID` ASC) VISIBLE,

UNIQUE INDEX `PassportID_UNIQUE` (`PassportID` ASC) VISIBLE,

CONSTRAINT `ResonID_fk`

FOREIGN KEY (`ResonID`)

REFERENCES `passportdb`.`Reson` (`ResonID`)

ON DELETE RESTRICT

ON UPDATE RESTRICT,

CONSTRAINT `TypeID_fk`

FOREIGN KEY (`TypeID`)

REFERENCES `passportdb`.`Type` (`TypeID`)

ON DELETE RESTRICT

ON UPDATE RESTRICT,

CONSTRAINT `PersonID_fk2`

FOREIGN KEY (`PersonID`)

REFERENCES `passportdb`.`Person` (`PersonID`)

ON DELETE RESTRICT

ON UPDATE RESTRICT,

CONSTRAINT `Manufacturing_timeID_fk`

FOREIGN KEY (`Manufacturing_timeID`)

REFERENCES `passportdb`.`Manufacturing_time` (`Manufacturing_timeID`)

ON DELETE RESTRICT

```

```
ON UPDATE RESTRICT,  
CONSTRAINT `RegionID_fk2`  
FOREIGN KEY (`RegionID`)  
REFERENCES `passportdb`.`Region` (`RegionID`)  
ON DELETE RESTRICT  
ON UPDATE RESTRICT,  
CONSTRAINT `EmployeeID_fk2`  
FOREIGN KEY (`EmployeeID`)  
REFERENCES `passportdb`.`Employee` (`EmployeeID`)  
ON DELETE RESTRICT  
ON UPDATE CASCADE,  
CONSTRAINT `StatusID_fk`  
FOREIGN KEY (`StatusID`)  
REFERENCES `passportdb`.`Status` (`StatusID`)  
ON DELETE RESTRICT  
ON UPDATE RESTRICT);
```

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