

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
Ометюх Р.Б.
Прийняв:
Мельникова Н.І.

Львів – 2018 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.

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 спроектованої бази даних. Для зв'язку коментарів і повідомлень встановлено обмеження цілісності «каскадне оновлення». Для полів status у таблицях MESSAGE та COMMENT визначено такий домен – (“опубліковане”, “неопубліковане”, “видалене”).

Створимо нову базу даних, виконавши такі команди:

```
-- MySQL Workbench Forward Engineering
```

```
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_ZE  
RO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
-- -----
```

```
-- Schema Ksions_db
```

```
-- -----
```

```
-- -----
```

```
-- Schema Ksions_db
```

```
-- -----
```

```
CREATE SCHEMA IF NOT EXISTS `Ksions_db` DEFAULT CHARACTER SET utf8 ;
```

```
-- -----
```

```
-- Schema db1
```

```
-- -----
```

```
-- -----
```

```
-- Schema db1
```

```
-- -----
```

```
CREATE SCHEMA IF NOT EXISTS `db1` ;
```

```
-- -----
```

```
-- Schema ksions_db
```

```
-- -----
```

```
-- -----
```

```
-- Schema ksions_db
```

```
-- -----
```

```
CREATE SCHEMA IF NOT EXISTS `ksions_db` DEFAULT CHARACTER SET utf8mb4 COLLATE  
utf8mb4_0900_ai_ci ;
```

```
USE `Ksions_db` ;
```

```
-- -----
```

```
-- Table `ksions_db`.`office`
```

```
-- -----
```

```
CREATE TABLE IF NOT EXISTS `ksions_db`.`office` (  
  `idOffice` INT(11) NOT NULL,  
  `Address` VARCHAR(45) NULL DEFAULT NULL,  
  `Work_start` TIME NULL DEFAULT NULL,  
  `Work_end` TIME NULL DEFAULT NULL,  
  PRIMARY KEY (`idOffice`))
```

```
ENGINE = InnoDB
```

```
DEFAULT CHARACTER SET = utf8mb4
```

```
COLLATE = utf8mb4_0900_ai_ci;
```

```
-- -----
```

```
-- Table `Ksions_db`.`Item`
```

```
-- -----
```

```

CREATE TABLE IF NOT EXISTS `Ksions_db`.`Item` (
  `idItem` INT NOT NULL,
  `Item_name` VARCHAR(45) NULL,
  `Price` INT NULL,
  `office_idOffice` INT(11) NOT NULL,
  PRIMARY KEY (`idItem`, `office_idOffice`),
  INDEX `fk_Item_office1_idx` (`office_idOffice` ASC) VISIBLE,
  CONSTRAINT `fk_Item_office1`
    FOREIGN KEY (`office_idOffice`)
      REFERENCES `ksions_db`.`office` (`idOffice`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `Ksions_db`.`Customer`
-----

```

```

CREATE TABLE IF NOT EXISTS `Ksions_db`.`Customer` (
  `idCustomer` INT NOT NULL,
  `Name` VARCHAR(45) NULL,
  `Surname` VARCHAR(45) NULL,
  `Document_id` VARCHAR(45) NULL,
  PRIMARY KEY (`idCustomer`))
ENGINE = InnoDB;

```

```
-- -----  
-- Table `Ksions_db`.`Customer_has_Item`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `Ksions_db`.`Customer_has_Item` (  
  `Customer_idCustomer` INT NOT NULL,  
  `Item_idItem` INT NOT NULL,  
  PRIMARY KEY (`Customer_idCustomer`, `Item_idItem`),  
  INDEX `fk_Customer_has_Item_Item1_idx` (`Item_idItem` ASC) VISIBLE,  
  INDEX `fk_Customer_has_Item_Customer_idx` (`Customer_idCustomer` ASC) VISIBLE,  
  CONSTRAINT `fk_Customer_has_Item_Customer`  
    FOREIGN KEY (`Customer_idCustomer`)  
    REFERENCES `Ksions_db`.`Customer` (`idCustomer`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Customer_has_Item_Item1`  
    FOREIGN KEY (`Item_idItem`)  
    REFERENCES `Ksions_db`.`Item` (`idItem`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
USE `db1` ;
```

```
USE `ksions_db` ;
```

```
-- -----  
-- Table `ksions_db`.`appointment`
```

```

-----
CREATE TABLE IF NOT EXISTS `ksions_db`.`appointment` (
  `idAppointment` INT(11) NOT NULL,
  `Appointment` VARCHAR(45) NULL DEFAULT NULL,
  `Salary` INT(11) NULL DEFAULT NULL,
  `Permissions` INT(11) NULL DEFAULT NULL,
  PRIMARY KEY (`idAppointment`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

```

```

-----
-- Table `ksions_db`.`worker`
-----

```

```

CREATE TABLE IF NOT EXISTS `ksions_db`.`worker` (
  `idWorker` INT(11) NOT NULL AUTO_INCREMENT,
  `Name` VARCHAR(45) NULL DEFAULT NULL,
  `Surname` VARCHAR(45) NULL DEFAULT NULL,
  `Phone` VARCHAR(45) NULL DEFAULT NULL,
  `e-mail` VARCHAR(45) NULL DEFAULT NULL,
  `appointment_idAppointment` INT(11) NOT NULL,
  `office_idOffice` INT(11) NOT NULL,
  PRIMARY KEY (`idWorker`, `appointment_idAppointment`, `office_idOffice`),
  INDEX `fk_worker_appointment_idx` (`appointment_idAppointment` ASC) VISIBLE,
  INDEX `fk_worker_office1_idx` (`office_idOffice` ASC) VISIBLE,
  CONSTRAINT `fk_worker_appointment`

```

```

FOREIGN KEY (`appointment_idAppointment`)
REFERENCES `ksions_db`.`appointment` (`idAppointment`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk_worker_office1`

FOREIGN KEY (`office_idOffice`)

REFERENCES `ksions_db`.`office` (`idOffice`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8mb4

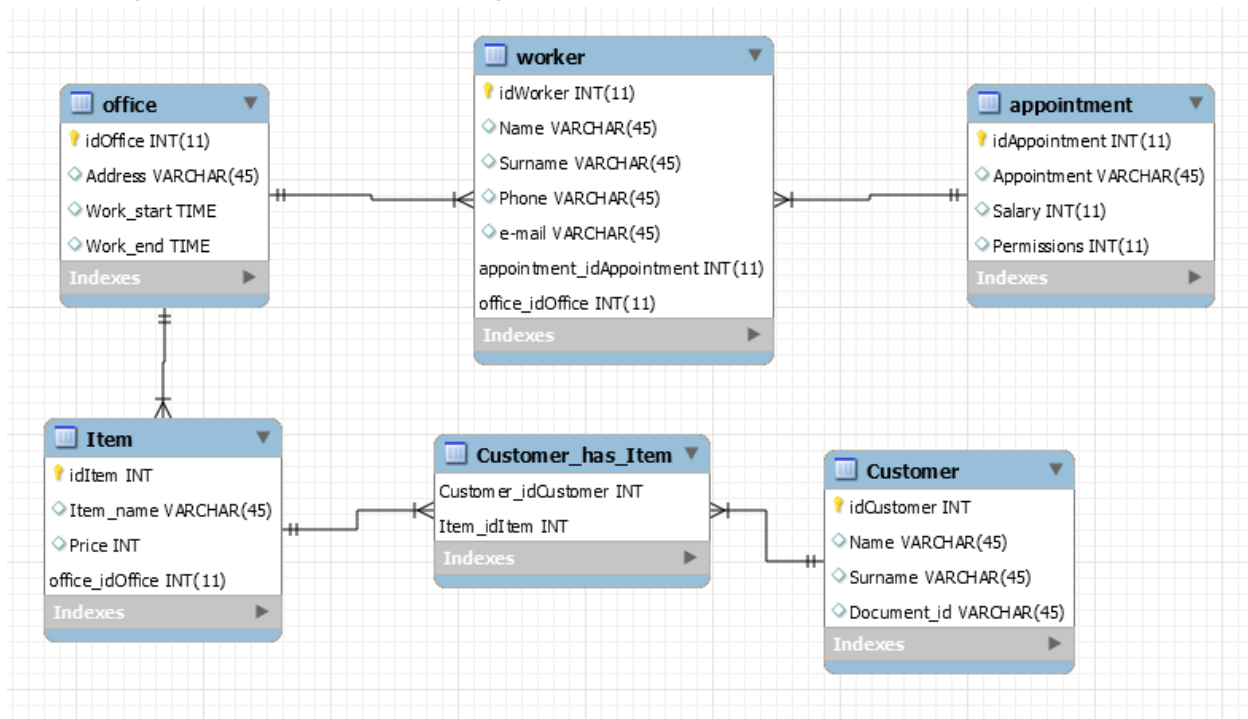
COLLATE = utf8mb4_0900_ai_ci,

SET SQL_MODE=@OLD_SQL_MODE;

SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;

SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```



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