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
Романко С.А.
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 проектованої 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 DB1

-- -----------------------------------------------------



-- -----------------------------------------------------

-- Schema DB1

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `DB1` DEFAULT CHARACTER SET utf8 ;

USE `DB1` ;



-- -----------------------------------------------------

-- Table `DB1`.`Worker`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `DB1`.`Worker` (

  `idWorker` INT NOT NULL,

  `Name` VARCHAR(45) NULL,

  `Surname` VARCHAR(45) NULL,
```

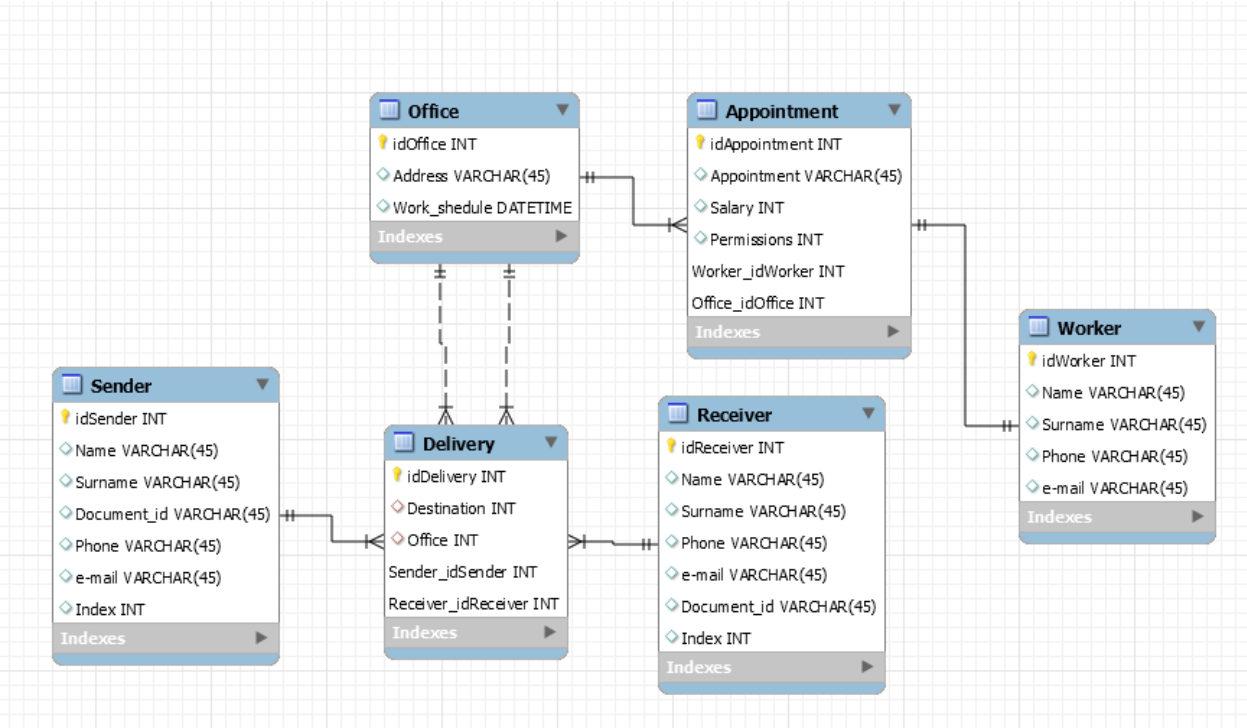
```
  `Phone` VARCHAR(45) NULL,
  `e-mail` VARCHAR(45) NULL,
  PRIMARY KEY (`idWorker`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `DB1`.`Office`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `DB1`.`Office` (
  `idOffice` INT NOT NULL,
  `Address` VARCHAR(45) NULL,
  `Work_shedule` DATETIME NULL,
  PRIMARY KEY (`idOffice`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `DB1`.`Appointment`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `DB1`.`Appointment` (
  `idAppointment` INT NOT NULL,
  `Appointment` VARCHAR(45) NULL,
  `Salary` INT NULL,
  `Permissions` INT NULL,
  `Worker_idWorker` INT NOT NULL,
```

```
  `Office_idOffice` INT NOT NULL,
  PRIMARY KEY (`idAppointment`, `Worker_idWorker`, `Office_idOffice`),
  INDEX `fk_Appointment_Worker1_idx` (`Worker_idWorker` ASC) VISIBLE,
  INDEX `fk_Appointment_Office1_idx` (`Office_idOffice` ASC) VISIBLE,
  CONSTRAINT `fk_Appointment_Worker1`
    FOREIGN KEY (`Worker_idWorker`)
    REFERENCES `DB1`.`Worker` (`idWorker`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Appointment_Office1`
    FOREIGN KEY (`Office_idOffice`)
    REFERENCES `DB1`.`Office` (`idOffice`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;



-- -----------------------------------------------------
-- Table `DB1`.`Sender`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `DB1`.`Sender` (
  `idSender` INT NOT NULL,
  `Name` VARCHAR(45) NULL,
  `Surname` VARCHAR(45) NULL,
  `Document_id` VARCHAR(45) NULL,
  `Phone` VARCHAR(45) NULL,
```

```
  `e-mail` VARCHAR(45) NULL,
  `Index` INT NULL,
  PRIMARY KEY (`idSender`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `DB1`.`Receiver`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `DB1`.`Receiver` (
  `idReceiver` INT NOT NULL,
  `Name` VARCHAR(45) NULL,
  `Surname` VARCHAR(45) NULL,
  `Phone` VARCHAR(45) NULL,
  `e-mail` VARCHAR(45) NULL,
  `Document_id` VARCHAR(45) NULL,
  `Index` INT NULL,
  PRIMARY KEY (`idReceiver`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `DB1`.`Delivery`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `DB1`.`Delivery` (
  `idDelivery` INT NOT NULL,
```

```
`Destination` INT NULL,
`Office` INT NULL,
`Sender_idSender` INT NOT NULL,
`Receiver_idReceiver` INT NOT NULL,
PRIMARY KEY (`idDelivery`, `Sender_idSender`, `Receiver_idReceiver`),
INDEX `fk_Delivery_Sender_idx` (`Sender_idSender` ASC) VISIBLE,
INDEX `fk_Delivery_Office1_idx` (`Destination` ASC) VISIBLE,
INDEX `fk_Delivery_Office2_idx` (`Office` ASC) VISIBLE,
INDEX `fk_Delivery_Receiver1_idx` (`Receiver_idReceiver` ASC) VISIBLE,
CONSTRAINT `fk_Delivery_Sender`
  FOREIGN KEY (`Sender_idSender`)
  REFERENCES `DB1`.`Sender` (`idSender`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Delivery_Office1`
  FOREIGN KEY (`Destination`)
  REFERENCES `DB1`.`Office` (`idOffice`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Delivery_Office2`
  FOREIGN KEY (`Office`)
  REFERENCES `DB1`.`Office` (`idOffice`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Delivery_Receiver1`
  FOREIGN KEY (`Receiver_idReceiver`)
```

```
    REFERENCES `DB1`.`Receiver` (`idReceiver`)
    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;
```



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