МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота №2

з дисципліни
«Організація баз даних та знань»

Виконав:
студент групи КН-209

Качмар Олексій
Викладач:
Мельникова Н.І.

Львів – 2020 р.

Лабораторна робота №2

Тема: “Створення таблиць бази даних засобами SQL”

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).

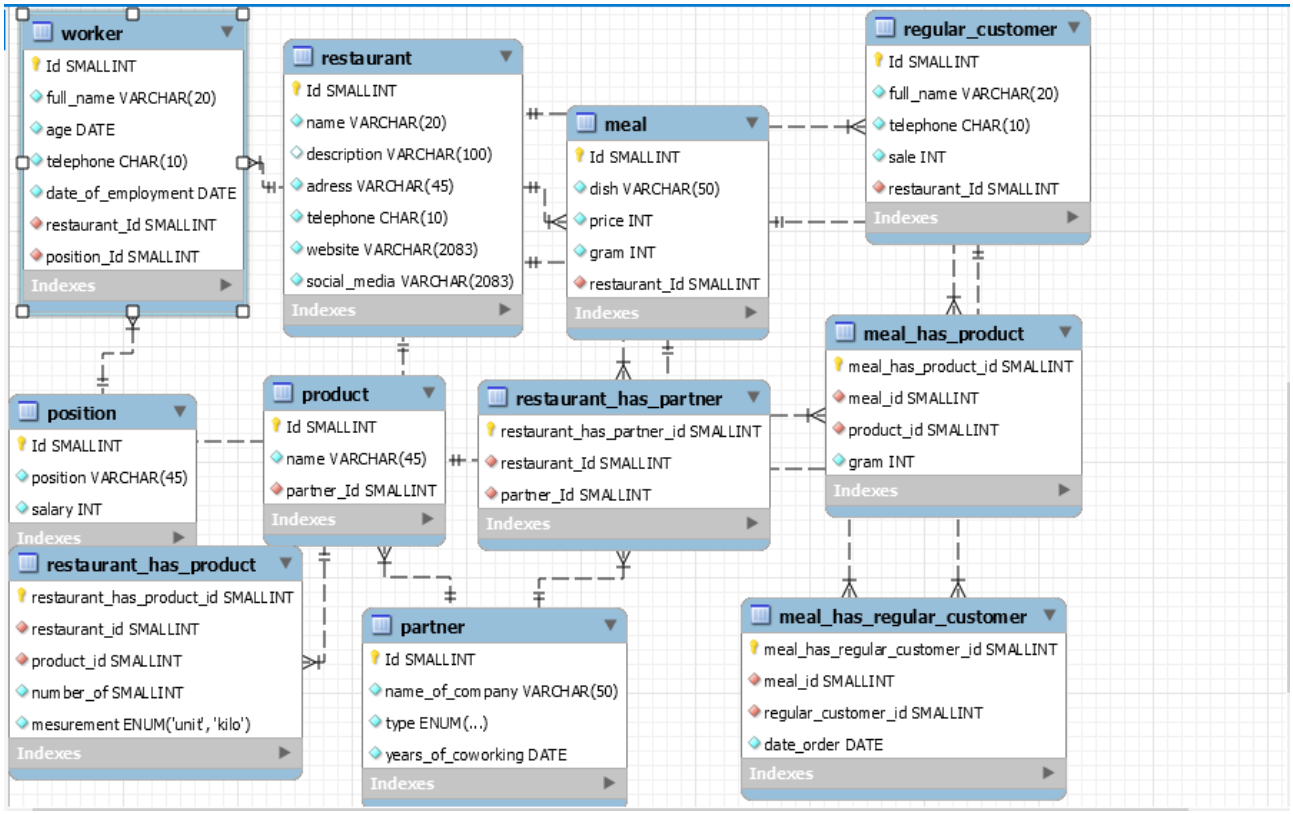
Нижче наведені деякі допоміжні команди для роботи в СУБД MySQL. Кожна команда і кожен запит в командному рядку повинні завершуватись розділяючим символом ";".

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. Розглянемо команду створення таблиці в MySQL та її основні аргументи. CREATE [TEMPORARY] TABLE [IF NOT EXISTS] ім’я_таблиці [(опис

_таблиці,...)] [додаткові_параметри] ... [вибірка_даних]



Сформуємо цю базу даних виконавши такі команди :

```
CREATE DATABASE `lab2`;

USE `lab2` ;


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

-- Table `lab2`.`restaurant`

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

CREATE TABLE IF NOT EXISTS `lab2`.`restaurant` (

 `Id` SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,

 `name` VARCHAR(20) NOT NULL,

 `description` VARCHAR(100) NULL DEFAULT NULL,

 `adress` VARCHAR(45) NOT NULL,

 `telephone` CHAR(10) NOT NULL,

 `website` VARCHAR(2083) NOT NULL,

 `social_media` VARCHAR(2083) NOT NULL,

 PRIMARY KEY (`Id`))
```

```
-- -----------------------------------------------------
-- Table `lab2`.`meal`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab2`.`meal` (
  `Id` SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
  `dish` VARCHAR(50) NOT NULL,
  `price` INT NOT NULL,
  `gram` INT NOT NULL,
  `restaurant_Id` SMALLINT UNSIGNED NOT NULL,
  PRIMARY KEY (`Id`, `restaurant_Id`),
  INDEX `fk_meal_restaurant1_idx` (`restaurant_Id` ASC) VISIBLE,
  CONSTRAINT `fk_meal_restaurant1`
    FOREIGN KEY (`restaurant_Id`)
    REFERENCES `lab2`.`restaurant` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
 -----------------------------------------------------
-- Table `lab2`.`partner`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab2`.`partner` (
  `Id` SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
  `name_of_company` VARCHAR(50) NOT NULL,
  `type` ENUM('provisioner', 'ads partner', 'supplier of chemistry', 'advertiser') NOT NULL,
  `years_of_coworking` DATE NOT NULL,
  PRIMARY KEY (`Id`))
-- -----------------------------------------------------
-- Table `lab2`.`product`
```

```
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab2`.`product` (
  `Id` SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NOT NULL,
  `partner_Id` SMALLINT UNSIGNED NOT NULL,
  PRIMARY KEY (`Id`, `partner_Id`),
  INDEX `fk_product_partner1_idx` (`partner_Id` ASC) VISIBLE,
  CONSTRAINT `fk_product_partner1`
    FOREIGN KEY (`partner_Id`)
    REFERENCES `lab2`.`partner` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)


-- -----------------------------------------------------
-- Table `lab2`.`meal_has_product`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab2`.`meal_has_product` (
  `meal_id` SMALLINT UNSIGNED NOT NULL,
  `product_id` SMALLINT UNSIGNED NOT NULL,
  `gram` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`meal_id`, `product_id`),
  INDEX `fk_meal_has_product_product1_idx` (`product_id` ASC) VISIBLE,
  INDEX `fk_meal_has_product_meal1_idx` (`meal_id` ASC) VISIBLE,
  CONSTRAINT `fk_meal_has_product_meal1`
    FOREIGN KEY (`meal_id`)
    REFERENCES `lab2`.`meal` (`Id`),
  CONSTRAINT `fk_meal_has_product_product1`
    FOREIGN KEY (`product_id`)
```

```
    REFERENCES `lab2`.`product` (`Id`))


-- -----------------------------------------------------
-- Table `lab2`.`regular_customer`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab2`.`regular_customer` (
  `Id` SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(20) NOT NULL,
  `telephone` CHAR(10) NOT NULL,
  `sale` INT NOT NULL,
  `restaurant_Id` SMALLINT UNSIGNED NOT NULL,
  PRIMARY KEY (`Id`, `restaurant_Id`),
  INDEX `fk_regular_customer_restaurant1_idx` (`restaurant_Id` ASC) VISIBLE,
  CONSTRAINT `fk_regular_customer_restaurant1`
    FOREIGN KEY (`restaurant_Id`)
    REFERENCES `lab2`.`restaurant` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)


-- -----------------------------------------------------
-- Table `lab2`.`meal_has_regular customer`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab2`.`meal_has_regular_customer` (
  `meal_id` SMALLINT UNSIGNED NOT NULL,
  `regular_customer_id` SMALLINT UNSIGNED NOT NULL,
  `date_order` DATE NOT NULL,
  PRIMARY KEY (`meal_id`, `regular_customer_id`),
  INDEX `fk_meal_has_regular customer_regular customer1_idx` (`regular_customer_id` ASC)
VISIBLE,
```

```
  INDEX `fk_meal_has_regular customer_meal1_idx` (`meal_id` ASC) VISIBLE,
  CONSTRAINT `fk_meal_has_regular customer_meal1`
    FOREIGN KEY (`meal_id`)
    REFERENCES `lab2`.`meal` (`Id`),
  CONSTRAINT `fk_meal_has_regular customer_regular customer1`
    FOREIGN KEY (`regular_customer_id`)
    REFERENCES `lab2`.`regular_customer` (`Id`))

-- -----------------------------------------------------
-- Table `lab2`.`restaurant_has_product`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab2`.`restaurant_has_product` (
  `restaurant_id` SMALLINT UNSIGNED NOT NULL,
  `product_id` SMALLINT UNSIGNED NOT NULL,
  `number_of` SMALLINT UNSIGNED NOT NULL,
  `mesurement` ENUM("unit", "kilo") NOT NULL,
  PRIMARY KEY (`restaurant_id`, `product_id`),
  INDEX `fk_restaurant_has_product_product1_idx` (`product_id` ASC) VISIBLE,
  INDEX `fk_restaurant_has_product_restaurant1_idx` (`restaurant_id` ASC) VISIBLE,
  CONSTRAINT `fk_restaurant_has_product_product1`
    FOREIGN KEY (`product_id`)
    REFERENCES `lab2`.`product` (`Id`),
  CONSTRAINT `fk_restaurant_has_product_restaurant1`
    FOREIGN KEY (`restaurant_id`)
    REFERENCES `lab2`.`restaurant` (`Id`))

-- -----------------------------------------------------
-- Table `lab2`.`position`
```

```
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab2`.`position` (
  `Id` INT NOT NULL AUTO_INCREMENT,
  `position` VARCHAR(45) NOT NULL,
  `salary` INT NOT NULL,
  PRIMARY KEY (`Id`))


-- -----------------------------------------------------
-- Table `lab2`.`worker`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab2`.`worker` (
  `Id` SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
  `full_name` VARCHAR(20) NOT NULL,
  `age` DATE NOT NULL,
  `telephone` CHAR(10) NOT NULL,
  `date_of_employment` DATE NOT NULL,
  `restaurant_Id` SMALLINT UNSIGNED NOT NULL,
  `position_Id` INT NOT NULL,
  PRIMARY KEY (`Id`, `restaurant_Id`, `position_Id`),
  INDEX `fk_worker_restaurant1_idx` (`restaurant_Id` ASC) VISIBLE,
  INDEX `fk_worker_position1_idx` (`position_Id` ASC) VISIBLE,
  CONSTRAINT `fk_worker_restaurant1`
    FOREIGN KEY (`restaurant_Id`)
    REFERENCES `lab2`.`restaurant` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_worker_position1`
    FOREIGN KEY (`position_Id`)
```

```
  REFERENCES `lab2`.`position` (`Id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)


-- -----------------------------------------------------
-- Table `lab2`.`restaurant_has_partner`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `lab2`.`restaurant_has_partner` (
  `restaurant_Id` SMALLINT UNSIGNED NOT NULL,
  `partner_Id` SMALLINT UNSIGNED NOT NULL,
  PRIMARY KEY (`restaurant_Id`, `partner_Id`),
  INDEX `fk_restaurant_has_partner_partner1_idx` (`partner_Id` ASC) VISIBLE,
  INDEX `fk_restaurant_has_partner_restaurant1_idx` (`restaurant_Id` ASC) VISIBLE,
  CONSTRAINT `fk_restaurant_has_partner_restaurant1`
    FOREIGN KEY (`restaurant_Id`)
    REFERENCES `lab2`.`restaurant` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_restaurant_has_partner_partner1`
    FOREIGN KEY (`partner_Id`)
    REFERENCES `lab2`.`partner` (`Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
```

Висновок : я побудував даталогічну модель бази даних; визначив типи, розмірності та обмеження полів; визначив обмеження таблиць; розробив SQL запити для створення спроектованих таблиць.