**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).

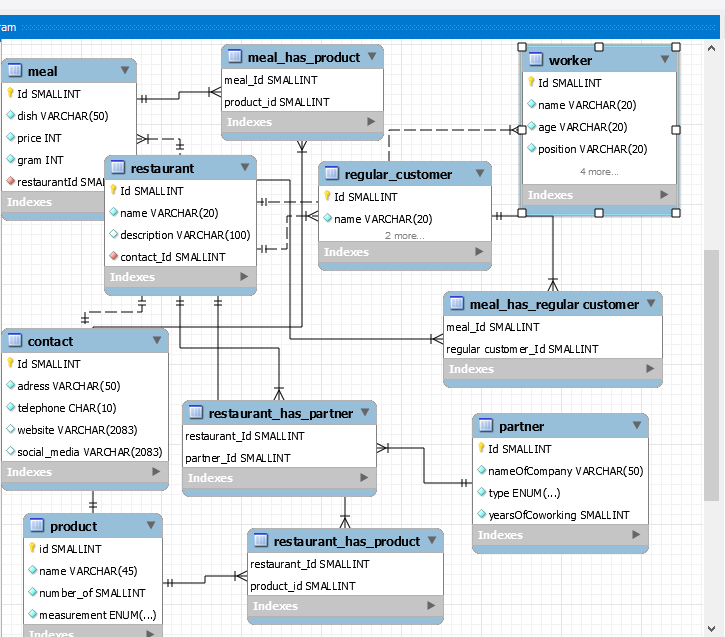
Нижче наведені деякі допоміжні команди для роботи в СУБД MySQL. Кожна команда і кожен запит в командному рядку повинні завершуватись розділяючим символом ";".

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 IF NOT EXISTS `lab2`;

USE `lab2` ;

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

-- Table `lab2`.`contact`

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

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

`Id` SMALLINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`adress` CHAR(50) NOT NULL,

`telephone` CHAR(10) NOT NULL,

`website` VARCHAR(2083) NULL DEFAULT '',

`social\_media` VARCHAR(2083) NULL DEFAULT '',

PRIMARY KEY (`Id`))

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

-- Table `lab2`.`restaurant`

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

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

`Id` SMALLINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`name` CHAR(20) NOT NULL,

`description` CHAR(100) NULL DEFAULT ‘’,

`contact\_Id` SMALLINT UNSIGNED NOT NULL,UNIQUE,

PRIMARY KEY (`Id`),

CONSTRAINT `fk\_restaurant\_contact1`

FOREIGN KEY (`contact\_Id`)

REFERENCES `lab2`.`contact` (`Id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

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

-- Table `lab2`.`meal`

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

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

`Id` SMALLINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`dish` CHAR(50) NOT NULL,

`price` INT NOT NULL,

`gram` INT NOT NULL,

`restaurantId` SMALLINT UNSIGNED NOT NULL,

PRIMARY KEY (`Id`),

INDEX `meal\_restaurant` (`restaurantId` ASC) VISIBLE,

CONSTRAINT `meal\_restaurant`

FOREIGN KEY (`restaurantId`)

REFERENCES `lab2`.`restaurant` (`Id`))

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

-- Table `lab2`.`partner`

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

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

`Id` SMALLINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`nameOfCompany` CHAR(50) NOT NULL,

`type` ENUM('provisioner', 'ads partner', 'supplier of chemistry') NOT NULL,

`yearsOfCoworking` SMALLINT UNSIGNED NOT NULL,

PRIMARY KEY (`Id`))

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

-- Table `lab2`.`worker`

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

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

`Id` SMALLINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`name` CHAR(20) NOT NULL,

`age` CHAR(20) NOT NULL,

`position` CHAR(20) NOT NULL,

`salary` INT NOT NULL,

`telephone` CHAR(10) NOT NULL,

`date\_of\_employment` DATE NOT NULL,

`restaurantId` SMALLINT UNSIGNED NOT NULL,

PRIMARY KEY (`Id`),

CONSTRAINT `worker\_restaurant`

FOREIGN KEY (`restaurantId`)

REFERENCES `lab2`.`restaurant` (`Id`))

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

-- Table `lab2`.`product`

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

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

`id` SMALLINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`name` CHAR(45) NOT NULL,

`number\_of` SMALLINT NOT NULL,

`measurement` ENUM(“units”,”kilograms”) NOT NULL ,

PRIMARY KEY (`id`))

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

-- Table `lab2`.`regular customer`

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

CREATE TABLE IF NOT EXISTS `lab2`.`regular customer` (

`Id` SMALLINT UNSIGNED NOT NULL AUTO\_INCREMENT,

`name` CHAR(20) NOT NULL,

`telephone` CHAR(10) NOT NULL,

`restaurant\_id` SMALLINT UNSIGNED NOT NULL,

PRIMARY KEY (`Id`),

CONSTRAINT `restaurant\_id`

FOREIGN KEY (`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,

PRIMARY KEY (`meal\_Id`, `regular customer\_Id`),

CONSTRAINT `fk\_meal\_has\_regular customer\_meal1`

FOREIGN KEY (`meal\_Id`)

REFERENCES `lab2`.`meal` (`Id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_meal\_has\_regular customer\_regular customer1`

FOREIGN KEY (`regular customer\_Id`)

REFERENCES `lab2`.`regular customer` (`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,

PRIMARY KEY (`meal\_Id`, `product\_id`),

CONSTRAINT `fk\_meal\_has\_product\_meal1`

FOREIGN KEY (`meal\_Id`)

REFERENCES `lab2`.`meal` (`Id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_meal\_has\_product\_product1`

FOREIGN KEY (`product\_id`)

REFERENCES `lab2`.`product` (`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`),

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)

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

-- 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,

PRIMARY KEY (`restaurant\_Id`, `product\_id`),

CONSTRAINT `fk\_restaurant\_has\_product\_restaurant1`

FOREIGN KEY (`restaurant\_Id`)

REFERENCES `lab2`.`restaurant` (`Id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_restaurant\_has\_product\_product1`

FOREIGN KEY (`product\_id`)

REFERENCES `lab2`.`product` (`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.