МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

ІНСТИТУТ ПІСЛЯДИПЛОМНОЇ ОСВІТИ

КАФЕДРА CИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ



ЗВІТ ДО ЛАБОРАТОРНОЇ РОБОТИ №2

*з курсу* “ОБДЗ”

*на тему:*

***“*** ***Створення таблиць бази даних засобами SQL ”***

**Підготувала:**

Студентка групи КН-209

Кульчицька Олена

**Викладач:**

Мельникова Н.І.

2020

Лабораторна робота №2

*з курсу* “ОБДЗ”

*на тему:*

***“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]

**опис\_таблиці**: назва\_поля опис\_поля

| [CONSTRAINT [ім’я\_обмеження]] PRIMARY KEY (назва\_поля,...) [тип\_обмеження]

| {INDEX|KEY} [ім’я\_обмеження] (назва\_поля,...)[ тип\_обмеження]

| [CONSTRAINT [ім’я\_обмеження]] UNIQUE [INDEX|KEY] [ім’я\_обмеження](назва\_поля,...) [тип\_обмеження]

| {FULLTEXT|SPATIAL} [INDEX|KEY] [ім’я\_обмеження] (назва\_поля,...)

[тип\_обмеження]

| [CONSTRAINT [ім’я\_обмеження]] FOREIGN KEY [ім’я\_обмеження] (назва\_поля,...) опис\_зв’язку

| CHECK (вираз)

**опис\_поля:**

тип\_даних [NOT NULL | NULL] [DEFAULT *значення\_за\_замовчуванням*] [AUTO\_INCREMENT] [UNIQUE [KEY] | [PRIMARY] KEY]

**опис\_зв’язку:**

REFERENCES *ім’я\_таблиці* (назва\_поля, ...) [ON DELETE дія]

[ON UPDATE дія]

**дія:**

CASCADE

Одночасне видалення, або оновлення відповідного значення у зовнішній таблиці.

RESTRICT

Аналог NO ACTION. Дія над значенням поля ігнорується, якщо існує відповідне йому значення у зовнішній таблиці. Опція задана за замовчуванням.

SET NULL

При дії над значенням у первинній таблиці, відповідне значення у зовнішній таблиці замінюється на NULL.

**додаткові\_параметри:**

{ENGINE|TYPE} [=] тип\_таблиці

| AUTO\_INCREMENT [=] *значення\_приросту\_лічильника*

| AVG\_ROW\_LENGTH [=] *значення*

| [DEFAULT] CHARACTER SET [=] кодування

| CHECKSUM [=] {0 | 1}

| [DEFAULT] COLLATE [=] *набір\_правил*

| COMMENT [=] 'коментар до таблиці'

| DATA DIRECTORY [=] 'абсолютний шлях'

| DELAY\_KEY\_WRITE [=] {0 | 1}

| INDEX DIRECTORY [=] 'абсолютний шлях'

| MAX\_ROWS [=] *значення*

| MIN\_ROWS [=] *значення*

| ROW\_FORMAT {DEFAULT|DYNAMIC|FIXED|COMPRESSED|REDUNDANT|COMPACT}

**вибірка\_даних:**

[IGNORE | REPLACE] [AS] SELECT ... (вибір даних з інших таблиць)

**вираз:**

Логічний вираз, що повертає TRUE або FALSE.

**Опис аргументів:**

ім’я\_таблиці

Назва таблиці. Або назва\_бази.назва\_таблиці.

тип\_таблиці

В MySQL крім типів таблиць MyISAM та InnoDB існують типи MEMORY, BDB, ARCHIVE тощо.

тип\_обмеження

Задає тип індексу для ключового поля: USING {BTREE | HASH | RTREE}.

TEMPORARY

Створення тимчасової таблиці, яка буде знищена після завершення зв’язку з сервером.

CONSTRAINT

Вказує на початок оголошення PRIMARY KEY, UNIQUE, або FOREIGN KEY

обмеження.

NULL | NOT NULL

Директива, що дозволяє/забороняє null-значення для даного поля.

PRIMARY KEY

Вказує, що дане поле буде первинним ключем в таблиці.

UNIQUE

Вказує на те, що в даному полі будуть зберігатися унікальні значення.

FOREIGN KEY ... REFERENCES

Створює зовнішній ключ, зв’язаний із вказаним полем (полями).

AVG\_ROW\_LENGTH

Приблизне значення середньої довжини рядків зі змінною довжиною.

DATA DIRECTORY

Вказує шлях, за яким таблиця має зберігатись у файловій системі.

CHECKSUM

Якщо параметр = 1, то для рядків таблиці буде рахуватись контрольна сума. Це сповільнює оновлення таблиці, але робить легшим пошук пошкоджених таблиць.

ROW\_FORMAT

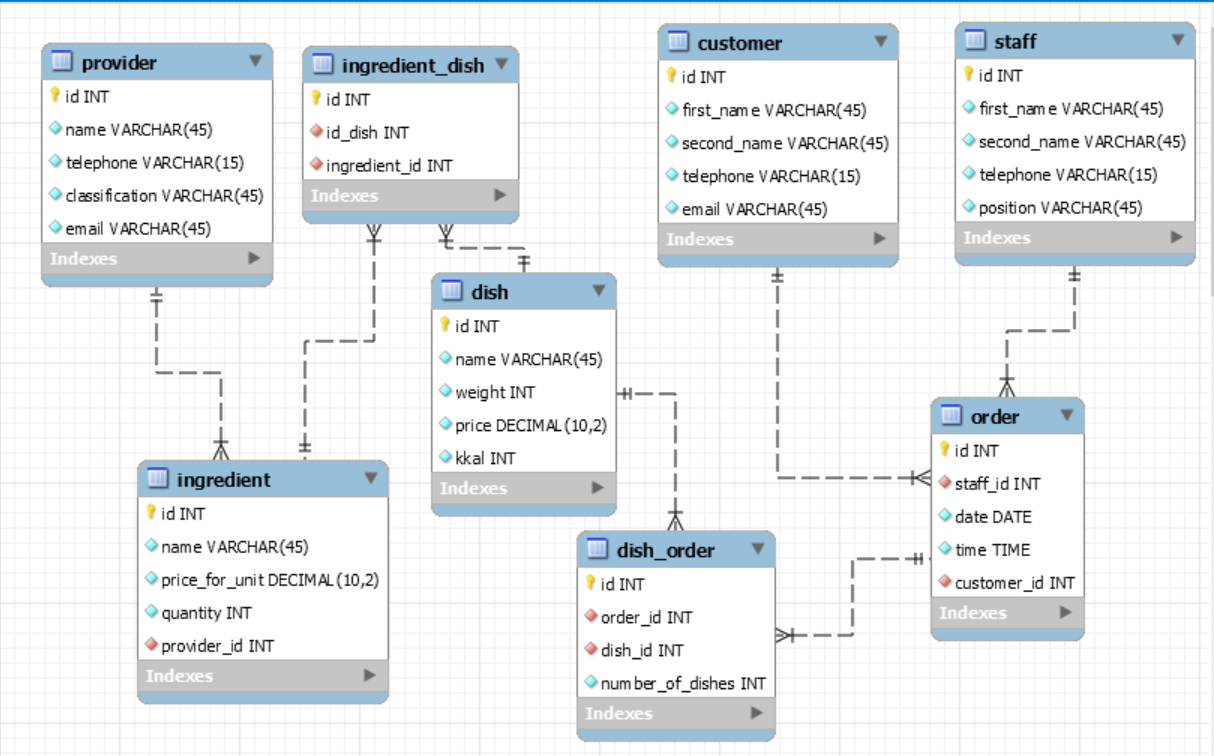
Вказує на спосіб зберігання рядків таблиці (залежно від типу таблиці).

FULLTEXT|SPATIAL

Тип індексу (повнотекстовий/просторовий; тільки для таблиць типу MyISAM).

**Хід роботи.**

Даталогічна модель вимагає визначення конкретних полів бази даних, їхніх типів, обмежень на значення, тощо. На рисунку зображено даталогічну модель проектованої бази даних.

****

Створимо нову базу даних, виконавши такі команди:

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='TRADITIONAL,ALLOW\_INVALID\_DATES';

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

-- Schema Confectionary

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

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

-- Schema Confectionary

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

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

USE `Confectionary` ;

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

-- Table `Confectionary`.`dish`

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

CREATE TABLE IF NOT EXISTS `Confectionary`.`dish` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NOT NULL,

`weight` INT NOT NULL,

`price` DECIMAL(10,2) NOT NULL,

`kkal` INT NOT NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB;

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

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

-- Table `Confectionary`.`customer`

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

CREATE TABLE IF NOT EXISTS `Confectionary`.`customer` (

`id` INT NOT NULL AUTO\_INCREMENT,

`first\_name` VARCHAR(45) NOT NULL,

`second\_name` VARCHAR(45) NOT NULL,

`telephone` VARCHAR(15) NOT NULL,

`email` VARCHAR(45) NOT NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB;

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

-- Table `Confectionary`.`staff`

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

CREATE TABLE IF NOT EXISTS `Confectionary`.`staff` (

`id` INT NOT NULL AUTO\_INCREMENT,

`first\_name` VARCHAR(45) NOT NULL,

`second\_name` VARCHAR(45) NOT NULL,

`telephone` VARCHAR(15) NOT NULL,

`position` VARCHAR(45) NOT NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB;

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

-- Table `Confectionary`.`provider`

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

CREATE TABLE IF NOT EXISTS `Confectionary`.`provider` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NOT NULL,

`telephone` VARCHAR(15) NOT NULL,

`classification` VARCHAR(45) NOT NULL,

`email` VARCHAR(45) NOT NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB;

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

-- Table `Confectionary`.`ingredient`

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

CREATE TABLE IF NOT EXISTS `Confectionary`.`ingredient` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NOT NULL,

`price\_for\_unit` DECIMAL(10,2) NOT NULL,

`quantity` INT NOT NULL,

`provider\_id` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `provider\_idx` (`provider\_id` ASC),

CONSTRAINT `provider`

FOREIGN KEY (`provider\_id`)

REFERENCES `Confectionary`.`provider` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

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

-- Table `Confectionary`.`order`

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

CREATE TABLE IF NOT EXISTS `Confectionary`.`order` (

`id` INT NOT NULL AUTO\_INCREMENT,

`staff\_id` INT NOT NULL,

`date` DATE NOT NULL,

`time` TIME NOT NULL,

`customer\_id` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `customer\_idx` (`customer\_id` ASC),

INDEX `staff\_idx` (`staff\_id` ASC),

CONSTRAINT `customer`

FOREIGN KEY (`customer\_id`)

REFERENCES `Confectionary`.`customer` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `staff`

FOREIGN KEY (`staff\_id`)

REFERENCES `Confectionary`.`staff` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

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

-- Table `Confectionary`.`dish\_order`

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

CREATE TABLE IF NOT EXISTS `Confectionary`.`dish\_order` (

`id` INT NOT NULL AUTO\_INCREMENT,

`order\_id` INT NOT NULL,

`dish\_id` INT NOT NULL,

`number\_of\_dishes` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `dish\_idx` (`dish\_id` ASC),

INDEX `order\_idx` (`order\_id` ASC),

CONSTRAINT `dish`

FOREIGN KEY (`dish\_id`)

REFERENCES `Confectionary`.`dish` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `order`

FOREIGN KEY (`order\_id`)

REFERENCES `Confectionary`.`order` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

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

-- Table `Confectionary`.`ingredient\_dish`

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

CREATE TABLE IF NOT EXISTS `Confectionary`.`ingredient\_dish` (

`id` INT NOT NULL AUTO\_INCREMENT,

`id\_dish` INT NOT NULL,

`ingredient\_id` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `ingredient\_idx` (`ingredient\_id` ASC),

INDEX `dish\_idx` (`id\_dish` ASC),

CONSTRAINT `ingredientid`

FOREIGN KEY (`ingredient\_id`)

REFERENCES `Confectionary`.`ingredient` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `dishid`

FOREIGN KEY (`id\_dish`)

REFERENCES `Confectionary`.`dish` (`id`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;