

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут комп'ютерних наук та інформаційних технологій
Кафедра «Системи штучного інтелекту»



Лабораторна Робота №2
З предмету: «Організація баз даних та знань»

Виконав
студент групи КН-211
Турик Олександр
Прийняла :
Якимішин Х.М..

Львів-2020

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 розділюючим 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

Створюємо базу даних і прописуємо, що її будемо використовувати :

CREATE DATABASE `Calendar`;

USE `Calendar`;

Створюємо таблицю користувача так як він є основною сутністю :

CREATE TABLE `User` (
 `id_User` INT NOT NULL AUTO_INCREMENT,

```

        `Name` VARCHAR(50) NOT NULL,
        `Surname` VARCHAR(100) NOT NULL,
        `Position` VARCHAR(100),
        `E-mail` VARCHAR(100) NOT NULL UNIQUE,
        `Password` VARCHAR(100) NOT NULL UNIQUE,
        `Is_Admin` bool NOT NULL,
        PRIMARY KEY (`id_user`)
    );

```

Створюємо таблицю Типів подій :

```

CREATE TABLE `Types` (
    `id_Type` INT NOT NULL AUTO_INCREMENT,
    `Name` VARCHAR(100) NOT NULL,
    `Description` VARCHAR(1000),
    PRIMARY KEY (`id_Type`)
);

```

Створюємо таблицю Подій :

```

CREATE TABLE `Event` (
    `id_Event` INT NOT NULL AUTO_INCREMENT,
    `Name` VARCHAR(200) NOT NULL,
    `Creator` VARCHAR(100) NOT NULL,
    `Description` VARCHAR(10000) NOT NULL,
    `date` VARCHAR(30) NOT NULL,
    `time` VARCHAR(30) NOT NULL,
    `Link` VARCHAR(200),
    `Status` VARCHAR(100) NOT NULL,
    `id_type` INT,
    PRIMARY KEY (`id_Event`),
    FOREIGN KEY (`id_type`) REFERENCES `Types` (`id_Type`)
    ON DELETE CASCADE ON UPDATE CASCADE
);

```

Створюємо таблицю Компанії :

```

CREATE TABLE `Company` (
    `id_company` INT NOT NULL,
    `Name` VARCHAR(50) NOT NULL,
    `Description` VARCHAR(10000) NOT NULL,
    PRIMARY KEY (`id_company`)
);

```

Створюємо таблицю Зовнішні учасники :

```

CREATE TABLE `Participants` (
    `id_Participant` INT NOT NULL AUTO_INCREMENT,
    `Name` VARCHAR(100) NOT NULL,
    `Surname` VARCHAR(100) NOT NULL,
    `Position` VARCHAR(100) NOT NULL,
    `E-mail` VARCHAR(100) NOT NULL UNIQUE,
    `Login` VARCHAR(100) NOT NULL UNIQUE,
    `Password` VARCHAR(100) NOT NULL UNIQUE,
    `id_company` INT,
    PRIMARY KEY (`id_participant`)
);

```

Створюємо таблицю Пропоновані події :

```

CREATE TABLE `Request` (
    `id_Request` INT NOT NULL AUTO_INCREMENT,

```

```

        `id_participant` INT NOT NULL,
        `Name_Event` VARCHAR(50) NOT NULL,
        `Description` VARCHAR(1000) NOT NULL,
        `Date` VARCHAR(30),
        `Time` VARCHAR(10),
        `Link` VARCHAR(200),
        `Status` VARCHAR(100),
        PRIMARY KEY (`id_request`),
        CONSTRAINT `Request_fk0` FOREIGN KEY (`id_participant`) REFERENCES `Participants`
        (`id_Participant`)
        ON DELETE CASCADE ON UPDATE CASCADE
    );

```

Створюємо таблицю яка відображає зв'язок багато до багатьох між користувачем і подіями :

```

CREATE TABLE `User_has_Events` (
    `id_user` int NOT NULL,
    `id_Events` int NOT NULL,
    CONSTRAINT `User_has_Events_fk0` FOREIGN KEY (`id_user`) REFERENCES `User` (`id_User`)
    ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT `User_has_Events_fk1` FOREIGN KEY (`id_Events`) REFERENCES `Event` (`id_Event`)
    ON DELETE CASCADE ON UPDATE CASCADE
);

```

Створюємо таблицю яка відображає зв'язок багато до багатьох між користувачем і пропонованою подією :

```

CREATE TABLE `User_has_request` (
    `id_user` INT NOT NULL,
    `id_request` INT NOT NULL,
    CONSTRAINT `User_has_requets_fk0` FOREIGN KEY (`id_user`) REFERENCES `User` (`id_User`)
    ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT `User_has_requests_fk1` FOREIGN KEY (`id_request`) REFERENCES `Request`
    (`id_Request`)
    ON DELETE CASCADE ON UPDATE CASCADE
);

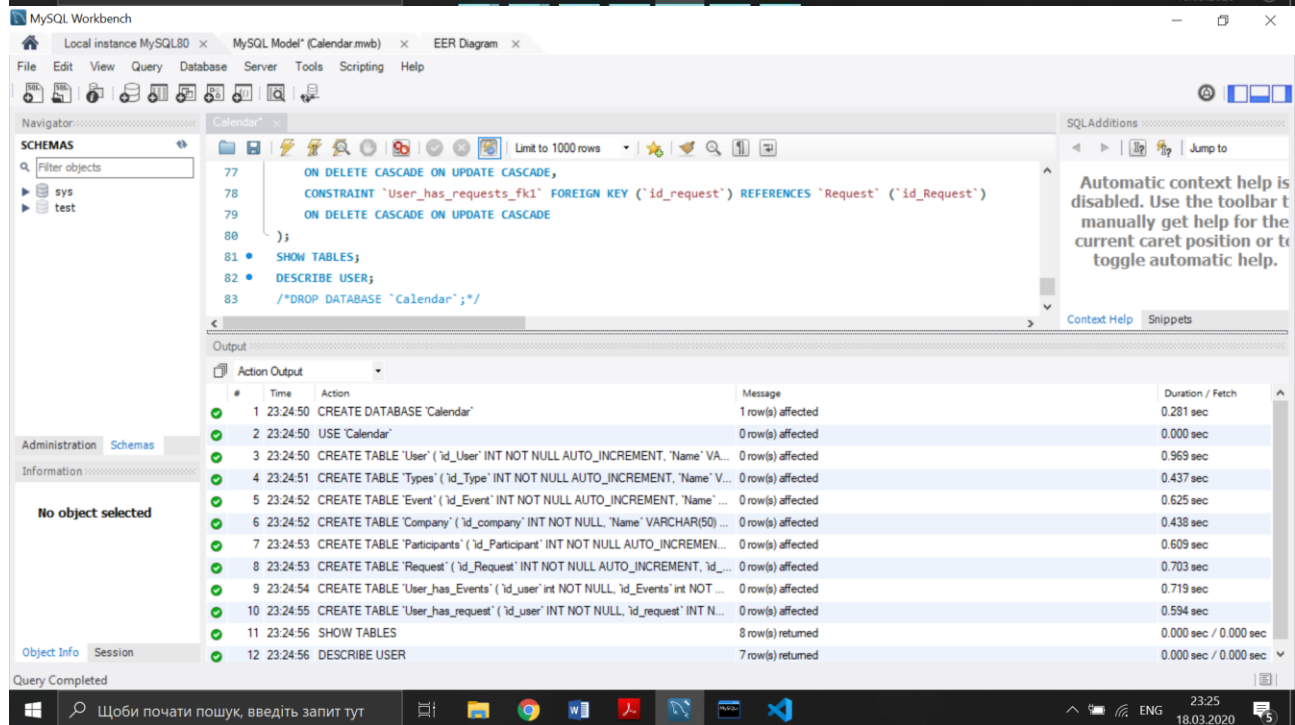
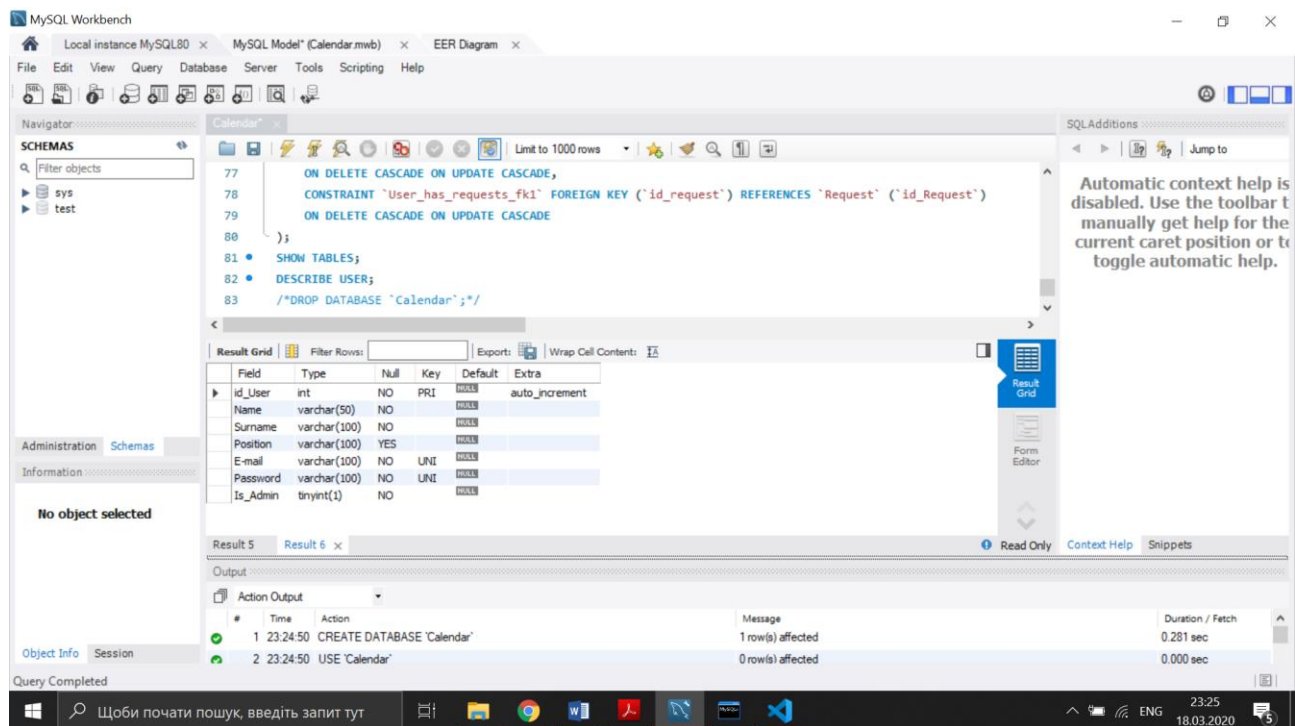
```

Показую таблиці :

```
SHOW TABLES;
```

Описати таблицю користувача :

```
DESCRIBE USER;
```



Щоб видалити базу даних пропишу і виконаю команду

DROP DATABASE `Calendar`;

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