

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА
ПОЛІТЕХНІКА»

Кафедра систем штучного інтелекту



Лабораторна робота №2
З курсу “Організація баз даних та знань”
Варіант 18

Виконав:
ст.гр. КН-210
Петров Кирило
Перевірила:
Мельникова Н. І.

Львів – 2020

Тема: “Створення таблиць бази даних засобами SQL”.

Мета: 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] ... [вибір_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]

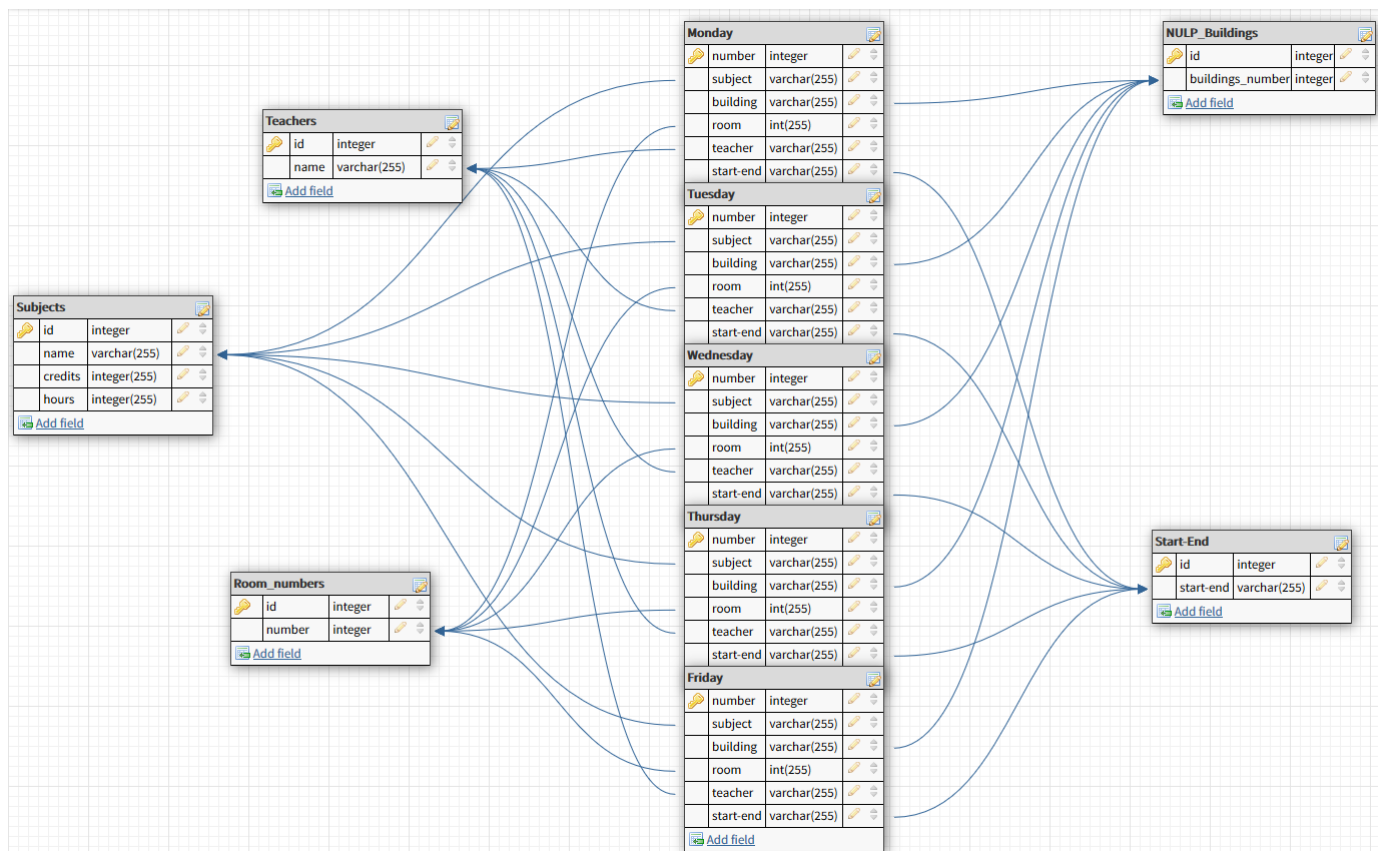
дія:

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}

Завдання: обудувати модель бази даних **розкладу кафедри**; визначити типи, розмірності та обмеження полів; визначити обмеження таблиць; розробити SQL запити для створення спроектованих таблиць.



```
CREATE TABLE Teachers (
    id INT PRIMARY KEY NOT NULL,
    name varchar(255) NOT NULL UNIQUE,
);
```

```
CREATE TABLE Subjects (
    id INT PRIMARY KEY NOT NULL,
    name varchar(255) NOT NULL UNIQUE,
    credits INT(255) NOT NULL,
    hours INT(255) NOT NULL,
);
```

```
CREATE TABLE Room_numbers (
    id INT PRIMARY KEY NOT NULL,
```

```
        number INT NOT NULL UNIQUE,  
    );
```

```
CREATE TABLE NULP_Buildings (  
    id INT PRIMARY KEY NOT NULL ,  
    buildings_number INT NOT NULL UNIQUE,  
);
```

```
CREATE TABLE Start_End (  
    id INT PRIMARY KEY NOT NULL,  
    start_end varchar(255) NOT NULL UNIQUE,  
);
```

```
CREATE TABLE Monday(  
    number INT PRIMARY KEY NOT NULL,  
    subject varchar(255) NOT NULL,  
    building INT NOT NULL,  
    room varchar(255) NOT NULL,  
    teacher varchar(255) NOT NULL,  
    start_end varchar(255) NOT NULL,  
    FOREIGN KEY (subject) REFERENCES Subjects(name),  
    FOREIGN KEY (building) REFERENCES  
NULP_Buildings(buildings_number),  
    FOREIGN KEY (room) REFERENCES Room_numbers(number),  
    FOREIGN KEY (teacher) REFERENCES Teachers(name),  
    FOREIGN KEY (start_end) REFERENCES Start_End(start_end)  
);
```

```
CREATE TABLE Tuesday (  
    number INT PRIMARY KEY NOT NULL,  
    subject varchar(255) NOT NULL,  
    building INT NOT NULL,  
    room varchar(255) NOT NULL,  
    teacher varchar(255) NOT NULL,  
    start_end varchar(255) NOT NULL,  
    FOREIGN KEY (subject) REFERENCES Subjects(name),  
    FOREIGN KEY (building) REFERENCES  
NULP_Buildings(buildings_number),  
    FOREIGN KEY (room) REFERENCES Room_numbers(number),  
    FOREIGN KEY (teacher) REFERENCES Teachers(name),  
    FOREIGN KEY (start_end) REFERENCES Start_End(start_end)  
);
```

```
CREATE TABLE Wednesday (  
    number INT PRIMARY KEY NOT NULL,  
    subject varchar(255) NOT NULL,  
    building INT NOT NULL,  
    room varchar(255) NOT NULL,  
    teacher varchar(255) NOT NULL,  
    start_end varchar(255) NOT NULL,  
    FOREIGN KEY (subject) REFERENCES Subjects(name),  
    FOREIGN KEY (building) REFERENCES  
NULP_Buildings(buildings_number),  
    FOREIGN KEY (room) REFERENCES Room_numbers(number),  
    FOREIGN KEY (teacher) REFERENCES Teachers(name),
```

```
FOREIGN KEY (start_end) REFERENCES Start_End(start_end)
);
```

```
CREATE TABLE Thursday (
    number INT PRIMARY KEY NOT NULL,
    subject varchar(255) NOT NULL,
    building INT NOT NULL,
    room varchar(255) NOT NULL,
    teacher varchar(255) NOT NULL,
    start_end varchar(255) NOT NULL,
    FOREIGN KEY (subject) REFERENCES Subjects(name),
    FOREIGN KEY (building) REFERENCES
NULP_Buildings(buildings_number),
    FOREIGN KEY (room) REFERENCES Room_numbers(number),
    FOREIGN KEY (teacher) REFERENCES Teachers(name),
    FOREIGN KEY (start_end) REFERENCES Start_End(start_end)
);
```

```
CREATE TABLE Friday (
    number INT PRIMARY KEY NOT NULL,
    subject varchar(255) NOT NULL,
    building INT NOT NULL,
    room varchar(255) NOT NULL,
    teacher varchar(255) NOT NULL,
    start_end varchar(255) NOT NULL,
    FOREIGN KEY (subject) REFERENCES Subjects(name),
```

```
FOREIGN KEY (building) REFERENCES  
NULP_Buildings(buildings_number),  
FOREIGN KEY (room) REFERENCES Room_numbers(number),  
FOREIGN KEY (teacher) REFERENCES Teachers(name),  
FOREIGN KEY (start_end) REFERENCES Start_End(start_end)  
);
```

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