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] ... [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:

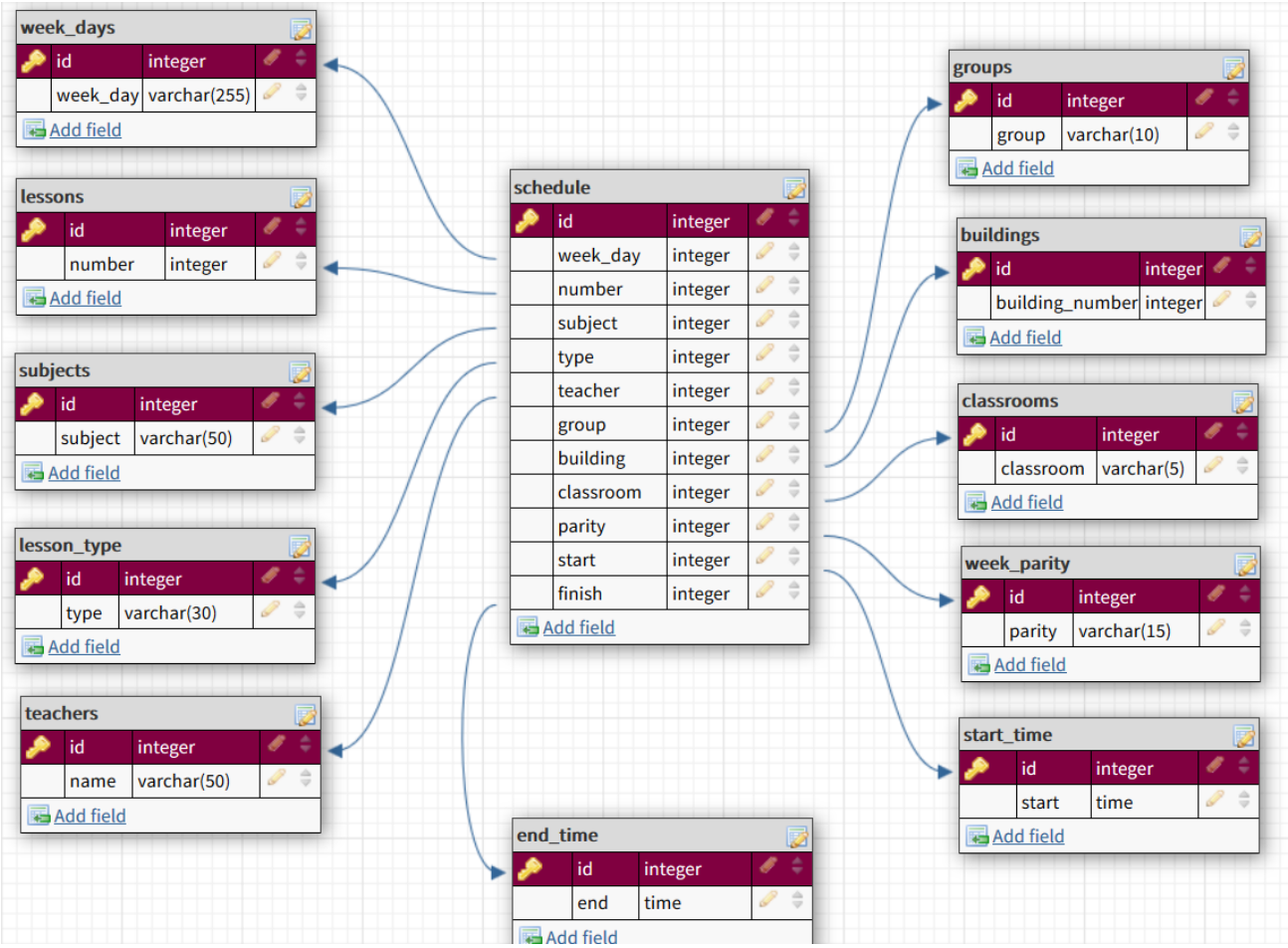
CASCADE Одночасне видалення, або оновлення відповідного значення у зовнішній таблиці. RESTRICT Аналог NO ACTION. Дія над значенням поля ігнорується, якщо існує відповідне йому значення у зовнішній таблиці. Опція задана за замовчуванням. SET NULL При дії над значенням у первинній таблиці, відповідне значення 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}

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




```
drop table if exists week_days;
create table week_days (
        id integer primary key auto_increment,
        week_day varchar(255) not null
);

drop table if exists lessons;
create table lessons(
        id integer primary key auto_increment,
         number integer not null
);
```

```
drop table if exists subjects;
create table subjects(
        id integer primary key auto_increment,
        subject varchar(50) not null
);

drop table if exists lesson_types;
create table lesson_types(
        id integer primary key auto_increment,
         type varchar(30) not null
);

drop table if exists teachers;
create table teachers(
        id integer primary key auto_increment,
         name varchar(50) not null
);

drop table if exists groupes;
create table groupes(
        id integer primary key auto_increment,
        groupe varchar(10) not null
);

 drop table if exists buildings;
 create table buildings(
        id integer primary key auto_increment,
```

```
        building_number integer not null
);

drop table if exists classrooms;
create table classrooms(
        id integer primary key auto_increment,
        classroom varchar(5) not null
);

drop table if exists week_parity;
create table week_parity(
        id integer primary key auto_increment,
        parity varchar(5) not null
);

drop table if exists start_time;
create table start_time(
        id integer primary key auto_increment,
        start time not null
);

drop table if exists end_time;
create table end_time(
        id integer primary key auto_increment,
        end time not null
);
drop table if exists schedule;
```

```
create table schedule (
        id int primary key auto_increment,
        week_day integer,
        number integer,
         subject integer,
         type integer,
        teacher integer,
        groupe integer,
        building integer,
        classroom integer,
        parity integer,
         start integer,
        end integer,
        foreign key (week_day) references week_days(id),
        foreign key (number) references lessons(id),
        foreign key (subject) references subjects(id),
        foreign key (type) references lesson_types(id),
        foreign key (teacher) references teachers (id),
        foreign key (groupe) references groupes(id),
        foreign key (building) references buildings(id),
        foreign key (classroom) references classrooms(id),
        foreign key (parity) references week_parity(id),
        foreign key (start) references start_time(id),
        foreign key (end) references end_time(id)
   );
```

Висновок: на цій лабораторній роботі було завершено моделювання і засобами SQL створено базу даних розкладу кафедри, що складається з 12 таблиць.