

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту



Звіт до лабораторної роботи №3

з дисципліни
“ОБДЗ”

Виконав:
ст. гр. КН-211
Ільків Андрій

Викладач:
Яхимишин Х. М.

Львів – 2020

“Модифікація структури таблиць бази даних засобами

SQL”

Мета роботи: Розробити SQL запити для зміни структури таблиць бази даних (включення нових полів, вилучення полів таблиць, зміни опису полів, зміни обмежень).

Короткі теоретичні відомості.

Для зміни структури таблиць використовується наступна директива мови SQL.

Синтаксис:

ALTER [IGNORE] **TABLE** *ім'я_таблиці*
опис_зміни [, опис_зміни ...]

опис_зміни:

[додаткові_параметри ...] | **ADD** [**COLUMN**] *назва_поля опис_поля* [FIRST | AFTER *назва_поля*] | **ADD** [**COLUMN**] (*назва_поля опис_поля,...*) | **ADD** {**INDEX**|**KEY**} [*ім'я_обмеження*] (*назва_поля,...*)[*тип_обмеження*] | **ADD** [**CONSTRAINT** [*ім'я_обмеження*]] **PRIMARY KEY** (*назва_поля,...*) [*тип_обмеження*] | **ADD** [**CONSTRAINT** [*ім'я_обмеження*]] **UNIQUE** [**INDEX**|**KEY**]

[*ім'я_обмеження*](*назва_поля,...*)[*тип_обмеження*] | **ADD** [**FULLTEXT**|**SPATIAL**]

[**INDEX**|**KEY**] [*ім'я_обмеження*] (*назва_поля,...*) [*тип_обмеження*] | **ADD** [**CONSTRAINT** [*ім'я_обмеження*]] **FOREIGN KEY** [*ім'я_обмеження*] (*назва_поля,...*) опис_зв'язку | **ALTER** [**COLUMN**] *назва_поля* {**SET DEFAULT** *значення* | **DROP DEFAULT**} | **CHANGE** [**COLUMN**] *стара_назва_поля нова_назва_поля опис_поля* [FIRST|AFTER *назва_поля*] | **MODIFY** [**COLUMN**] *назва_поля опис_поля* [FIRST | AFTER *назва_поля*] | **DROP** [**COLUMN**] *назва_поля* | **DROP PRIMARY KEY** | **DROP** {**INDEX**|**KEY**} *ім'я_обмеження* | **DROP FOREIGN KEY** *ім'я_обмеження* | **RENAME** [TO] *нова_назва_таблиці* | **ORDER BY** *назва_поля* [,*назва_поля*] ... | **CONVERT TO CHARACTER SET** *кодування* [**COLLATE** *набір_правил*] | [**DEFAULT**] **CHARACTER SET** [=] *кодування* [**COLLATE** [=] *набір_правил*]

додаткові_параметри:

{**ENGINE**|**TYPE**} [=] *тип_таблиці* |

AUTO_INCREMENT [=] *початкове_значення* |
AVG_ROW_LENGTH [=] *значення* | [DEFAULT]
CHARACTER SET [=] *кодування*
| CHECKSUM [=] {0 | 1} | [DEFAULT] COLLATE [=] *набір_правил* | COMMENT [=]
'*коментар до таблиці*' | DELAY_KEY_WRITE [=] {0 | 1} | MAX_ROWS [=] *значення* |
MIN_ROWS [=] *значення* | ROW_FORMAT
{DEFAULT|DYNAMIC|FIXED|COMPRESSED|REDUNDANT|COMPACT}

Аргументи:

тип_обмеження

Вказує тип індексу для поля: USING {BTREE | HASH | RTREE}. IGNORE

Ігнорувати помилки дублювання значень в ключових полях, тобто залишати в таблиці перше значення, і не вставляти дублікат. За замовчуванням, при такій помилці вся дія над таблицею відміняється. ALTER *назва_поля*

Використовується для зміни значення за замовчуванням для поля. MODIFY *назва_поля*

Змінити тип поля.

CHANGE *назва_поля*

Змінити назву і тип поля.

FIRST | AFTER

Змінити розташування поля в таблиці. За замовчуванням, нове поле додається в кінець таблиці. FIRST додасть його на початок; AFTER – одразу після вказаного поля. DROP *назва_поля*

Видалення поля, обмеження чи індексу. ORDER BY

Відсортувати рядки в таблиці за зростанням значень у вказаному полі (полях). AUTO_INCREMENT

При зміні початкового значення для приросту поля-лічильника, слід вказати значення більше за ті, які вже містяться в цьому полі. CONVERT TO

Задає кодування, в яке потрібно перевести таблицю. RENAME TO

Змінити назву таблиці на вказану. Для зміни назви таблиці, або для переносу таблиці в іншу базу, також можна використати директиву RENAME TABLE ім'я_бази_1.назва_таблиці_1 TO ім'я_бази_2.назва_таблиці_2.

Хід роботи.

Виконаємо такі операції над схемою бази даних.

1. Вилучимо зв'язок file_author.
2. Вилучимо і змінимо поля в таблиці File.
3. Змінимо поле в таблиці Author.
4. Додамо поле і нове обмеження унікальності до таблиці Message.
5. Змінимо тип обмеження цілісності для зв'язку file_message.

1. Видалення зв'язку між таблицями шляхом видалення обмеження для таблиці:

```
alter table studentsgroup  
drop foreign key group_teacher,  
drop index group_teacher;
```

Для перевірки результату можна виконати команди SHOW INDEX FROM studentsgroup

| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null |
|-----------------|------------|----------------|--------------|-------------|-----------|-------------|----------|--------|------|
| ▶ studentsgroup | 0 | PRIMARY | 1 | group_id | A | 0 | NULL | NULL | |
| studentsgroup | 1 | Group_category | 1 | category_id | A | 0 | NULL | NULL | |

2. Видалення поля і зміна розмірності поля в таблиці studentsgroup:

```
alter table studentsgroup  
drop column lectionId,  
modify column teacher_id int(22) null;
```

3. Зробимо поле first_name в таблиці student необов'язковим, і змінимо його розмірність:

```
alter table student  
  modify column first_name varchar(200)  
  null;
```

4. Додамо до таблиці car_student поле, яке дозволить знайти студента, який навчається їздити на певній машині.

```
alter table car  
  add column car_student int(20) unsigned not null after car_id,  
  add constraint unique student_on_car(car_id, car_student);
```

5. Змінимо обмеження цілісності student_group так, щоб при видаленні повідомлення, з бази видалялись записи про всі приєднані до нього файли. Для цього потрібно видалити обмеження і створити нове, оскільки MySQL не підтримує директиви ALTER CONSTRAINT.

```
alter table student drop foreign key student_group;
```

```
alter table student add constraint student_group foreign key (group_id)  
references studentsgroup (group_id) on delete no action;
```

Висновок: на цій лабораторній роботі було розглянуто методи модифікації структури таблиць БД засобами SQL та проведено модифікацію полів, зв'язків і обмежень.