

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра систем штучного інтелекту



ЗВІТ ДО ЛАБОРАТОРНОЇ РОБОТИ № 3

з курсу “ОБДЗ”

на тему:

“Модифікація структури таблиць бази даних засобами SQL”

Виконала:

студентка групи КН-211

Лаврик Юліана

Викладач:

Якимишин Х.М.

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 бази даних використовується наступний синтаксис 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

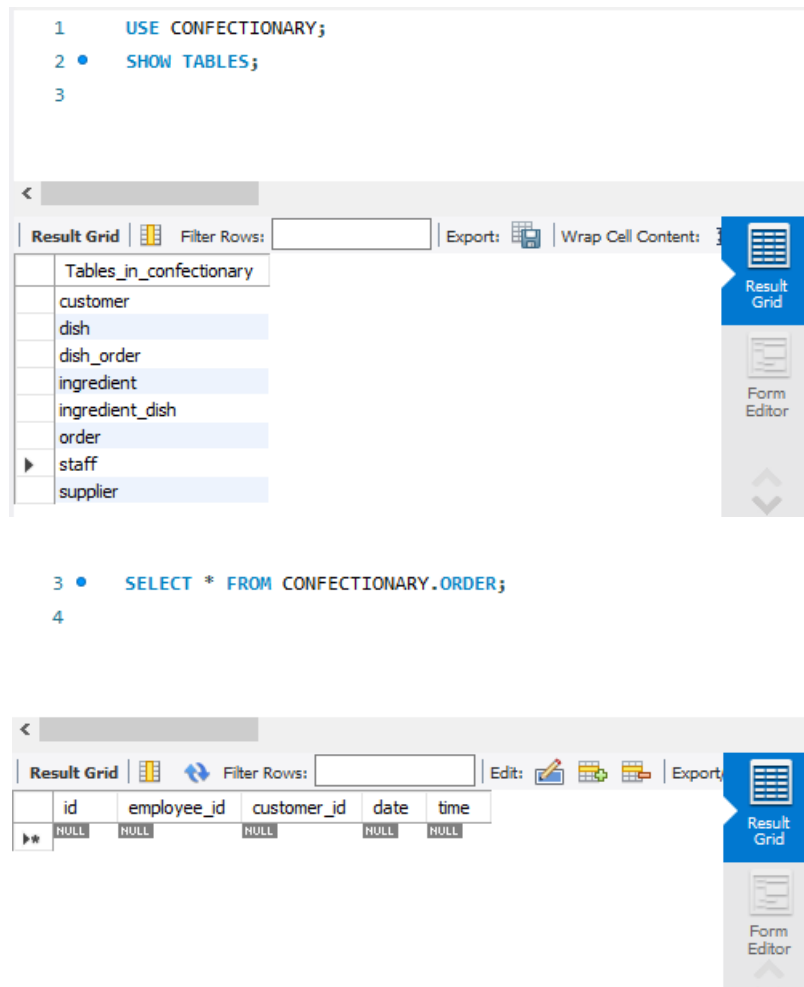
{DEFAULT|DYNAMIC|FIXED|COMPRESSED|REDUNDANT|COMPACT}

Опис аргументів :

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.

Хід роботи

За допомогою команд `SHOW TABLES` та `SELECT * FROM Confectionary.order` переглянемо ,які таблиці наявні у спроектованій базі даних та які поля містить таблиця `order` :



Розробимо SQL запити для зміни структури таблиць бази даних ,а саме виконаємо такі операції :

1. Вилучимо поле `date` в таблиці `order` за допомогою операторів `ALTER TABLE` та `DROP COLUMN` :

```
1 • ALTER TABLE Confectionary.order  
2 DROP COLUMN date;  
3  
4  
5  
6  
7
```

І перевіримо чи справді видалилось поле , використовуючи запит `SELECT * FROM :`

```
1 • SELECT * FROM Confectionary.order;
```



2. Змінимо поле `telephone` в таблиці `staff` .

Для цього спершу перевіряємо наявні параметри даного поля (`int`; `NULL`):

Column	Type	Default Value	Nullable	Chara
◇ first_name	varchar(100)		NO	utf8
◇ id	int		NO	
◇ last_name	varchar(100)		NO	utf8
◇ post	varchar(100)		NO	utf8
◇ telephone	int		YES	

Зробимо поле обов'язковим та змінимо його тип та розмірність . Це можна зробити за допомогою команд `ALTER TABLE` та `MODIFY COLUMN` :

```
1 • ALTER TABLE Confectionary.staff
2   MODIFY COLUMN telephone varchar(20) NOT NULL
3
```

Перевіримо чи відбулись зміни у таблиці `staff` :

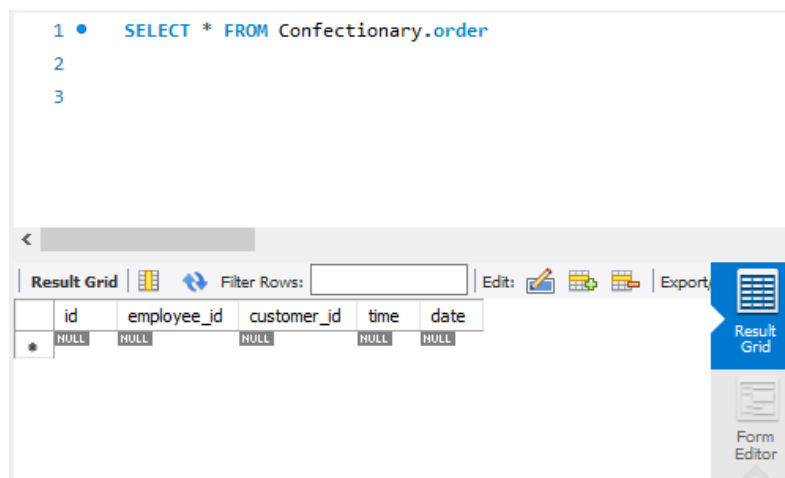
Column	Type	Default Value	Nullable	Chara
◇ first_name	varchar(100)		NO	utf8
◇ id	int		NO	
◇ last_name	varchar(100)		NO	utf8
◇ post	varchar(100)		NO	utf8
◇ telephone	varchar(20)		NO	utf8

3. Додамо поле date після поля time до таблиці order , використовуючи команди ALTER TABLE та ADD COLUMN:

```
1 • ALTER TABLE Confectionary.order
2     ADD COLUMN date DATE NOT NULL AFTER time ;
3
4
```

Перевіримо :

```
1 • SELECT * FROM Confectionary.order
2
3
```



	id	employee_id	customer_id	time	date
*	NULL	NULL	NULL	NULL	NULL

4. Вилучимо зв'язок між таблицями шляхом видалення обмеження для таблиці, а саме зв'язок між таблицями order та customer з використанням операторів ALTER TABLE, DROP FOREIGN KEY та DROP INDEX :

```
1 • ALTER TABLE Confectionary.order
2     DROP FOREIGN KEY customer_id,
3     DROP INDEX customer_id;
4
```

Перевіряємо результат за допомогою операторів SHOW INDEX FROM і SHOW CREATE TABLE :

До запитів :

```
1 SHOW INDEX FROM confectionary.order
```

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation
▶	order	0	PRIMARY	1	id	A
	order	1	customer_id	1	customer_id	A
	order	1	employee_id	1	employee_id	A

Після :

```
1 • SHOW INDEX FROM Confectionary.order
2
```

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation
▶	order	0	PRIMARY	1	id	A
	order	1	employee_id	1	employee_id	A

```
1 • SHOW CREATE TABLE Confectionary.order
```

Create Table:

```
CREATE TABLE `order` (
  `id` int NOT NULL AUTO_INCREMENT,
  `employee_id` int NOT NULL,
  `customer_id` int NOT NULL,
  `time` time NOT NULL,
  `date` date NOT NULL,
  PRIMARY KEY (`id`),
  KEY `employee_id` (`employee_id`),
  CONSTRAINT `employee_id` FOREIGN KEY
(`employee_id`) REFERENCES `staff` (`id`) ON DELETE
CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

Створимо нове обмеження унікальності для полів (customer_id,employee_id) :

```
1 ALTER TABLE confectionary.order
2 ADD CONSTRAINT UNIQUE confectionary (customer_id, employee_id);
```

#	Time	Action
1	04:49:43	ALTER TABLE confectionary.order ADD CONSTRAINT UNIQUE confectionary (customer_id, e...

Висновок : під час виконання даної лабораторної роботи я здобула навички розробки SQL запитів для зміни структури таблиць бази даних , а саме включення нових полів, вилучення полів таблиць, зміни опису полів, зміни обмежень.