

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**  
**Інститут комп'ютерних наук та інформаційних технологій**  
**Кафедра систем штучного інтелекту**



**Звіт до практичної роботи №3**  
**з дисципліни “ ОБДЗ ”**

**На тему: «Модифікація структури таблиць бази даних засобами SQL»**

**Виконав:**

ст. гр. КН-211

Качмарик Віктор

**Викладач:**

Якимишин Х. М.

### Лабораторна робота №3

**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 використовуються наступна деректива 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 команди: show tables, describe(customer):

```
mysql> use pharmacy;
Database changed
mysql> show tables;
+-----+
| Tables_in_pharmacy |
+-----+
| all_medicine        |
| customer            |
| delivery            |
| delivery_man        |
| medicine            |
| order               |
| order_item          |
| pharmacy            |
+-----+
8 rows in set (0.00 sec)

mysql> describe customer;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| customer_id | int       | NO   | PRI | NULL    | auto_increment |
| first_name  | varchar(40) | NO   |     | NULL    |              |
| last_name   | varchar(40) | NO   |     | NULL    |              |
| phone       | varchar(30) | NO   |     | NULL    |              |
| email       | varchar(50) | YES  |     | NULL    |              |
| street      | varchar(50) | YES  |     | NULL    |              |
| city        | varchar(50) | YES  |     | Lviv    |              |
| state       | varchar(50) | YES  |     | NULL    |              |
| zip_code    | varchar(50) | YES  |     | NULL    |              |
| login       | varchar(150) | NO   |     | NULL    |              |
| password    | varchar(150) | NO   |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

Виконаємо такі операції над схемою бази даних:

1. Вилучимо зв'язок pharmacy\_id(from order).
2. Вилучимо поле в таблиці customer(state).
3. Змінимо поле в таблиці pharmacy.
4. Додамо поле і нове обмеження унікальності до таблиці delivery.
5. Змінимо тип обмеження цілісності для зв'язку pharmacy\_if(in medicine).

2. Вилучимо поле в таблиці customer(state).

Використовуючи функції ALTER TABLE та DROP COLUMN :

```
mysql> describe customer
-> ;
```

Field	Type	Null	Key	Default	Extra
customer_id	int	NO	PRI	NULL	auto_increment
first_name	varchar(40)	NO		NULL	
last_name	varchar(40)	NO		NULL	
phone	varchar(30)	NO		NULL	
email	varchar(50)	YES		NULL	
street	varchar(50)	YES		NULL	
city	varchar(50)	YES		Lviv	
state	varchar(50)	YES		NULL	
zip_code	varchar(50)	YES		NULL	
login	varchar(150)	NO		NULL	
password	varchar(150)	NO		NULL	

11 rows in set (0.00 sec)

```
mysql> ALTER TABLE pharmacy.customer
-> DROP COLUMN state;
```

Query OK, 0 rows affected (0.07 sec)

Records: 0 Duplicates: 0 Warnings: 0

```
mysql> describe customer;
```

Field	Type	Null	Key	Default	Extra
customer_id	int	NO	PRI	NULL	auto_increment
first_name	varchar(40)	NO		NULL	
last_name	varchar(40)	NO		NULL	
phone	varchar(30)	NO		NULL	
email	varchar(50)	YES		NULL	
street	varchar(50)	YES		NULL	
city	varchar(50)	YES		Lviv	
zip_code	varchar(50)	YES		NULL	
login	varchar(150)	NO		NULL	
password	varchar(150)	NO		NULL	

10 rows in set (0.00 sec)

### 3. Змінимо поле в таблиці pharmacy.

Сперше перевіримо самі поля:

```
mysql> describe pharmacy;
```

Field	Type	Null	Key	Default	Extra
pharmacy_id	int	NO	PRI	NULL	auto_increment
pharmacy_name	varchar(200)	NO		NULL	
phone	varchar(30)	NO		NULL	
email	varchar(50)	NO		NULL	
street	varchar(50)	NO		NULL	
city	varchar(50)	NO		NULL	
state	varchar(50)	NO		NULL	

```
7 rows in set (0.00 sec)
```

Зробимо поле state на не обов'язкове та змінимо його розмірність, використовуючи функцію MODIFY COLUMN:

```
mysql> ALTER TABLE pharmacy
-> MODIFY COLUMN state varchar(70)
-> ;
Query OK, 0 rows affected (0.10 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> describe pharmacy;
```

Field	Type	Null	Key	Default	Extra
pharmacy_id	int	NO	PRI	NULL	auto_increment
pharmacy_name	varchar(200)	NO		NULL	
phone	varchar(30)	NO		NULL	
email	varchar(50)	NO		NULL	
street	varchar(50)	NO		NULL	
city	varchar(50)	NO		NULL	
state	varchar(70)	YES		NULL	

```
7 rows in set (0.00 sec)
```

### 4. Додамо поле і нове обмеження унікальності до таблиці delivery.

Для цього використаємо команду: ADD COLUMN:

```
mysql> describe delivery;
```

Field	Type	Null	Key	Default	Extra
delivery_id	int	NO	PRI	NULL	auto_increment
order_id	int	NO	UNI	NULL	
delivery_status	tinyint(1)	NO		NULL	
delivery_price	int	NO		NULL	

```
4 rows in set (0.00 sec)
```

```
mysql> ALTER TABLE delivery
-> ADD delivery_day varchar(100);
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> describe delivery;
```

Field	Type	Null	Key	Default	Extra
delivery_id	int	NO	PRI	NULL	auto_increment
order_id	int	NO	UNI	NULL	
delivery_status	tinyint(1)	NO		NULL	
delivery_price	int	NO		NULL	
delivery_day	varchar(100)	YES		NULL	

5 rows in set (0.00 sec)

5. Змінимо тип обмеження цілісності для зв'язку pharmacy\_id(from medicine).

```
mysql> ALTER TABLE medicine
-> ADD CONSTRAINT UNIQUE pharmacy(pharmacy_id);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> describe medicine;
```

Field	Type	Null	Key	Default	Extra
medicine_id	int	NO	PRI	NULL	auto_increment
pharmacy_id	int	NO	UNI	NULL	
medicine_name	varchar(150)	NO	MUL	NULL	
quantity	int	NO		NULL	
price	int	NO		NULL	

5 rows in set (0.00 sec)

1. Вилучимо зв'язок pharmacy\_id(from order).

```
mysql> SHOW INDEX FROM delivery;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
delivery	0	PRIMARY	1	delivery_id	A	0	NULL	NULL	NULL	BTREE			YES	NULL
delivery	0	order_id	1	order_id	A	0	NULL	NULL	NULL	BTREE			YES	NULL

2 rows in set (0.00 sec)

```
mysql> ALTER TABLE delivery
-> DROP FOREIGN KEY delivery_fk0;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

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