

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Запорізький національний технічний університет
Факультет інформатики та обчислювальної техніки

Кафедра комп'ютерних
систем та мереж

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт
з дисципліни “Організація баз даних”
для студентів спеціальності 123 “Комп’ютерна інженерія”
усіх форм навчання
частина VII

2019

Методичні вказівки до виконання лабораторних робіт з дисципліни “Організація баз даних” для студентів спеціальності 123 “Комп’ютерна інженерія” усіх форм навчання частина VII /Укл.: Т.О. Паромова, - Запоріжжя: ЗНТУ, 2019 – 22 с.

Укладачі: Т.О.Паромова, ст.викладач,

Рецензент: С.Д. Точилін, доцент, к.ф.-м.н.

Відповідальний за випуск: Паромова Т.О., ст.викладач.

Затверджено
на засіданні кафедри
“Комп’ютерні системи
та мережі ”
Протокол № 7
від “21” січня 2019.

Рекомендовано до видання
НМК факультету
комп’ютерних
наук і технологій
Протокол № 6
від “31” січня 2019.

ЗМІСТ

ЛАБОРАТОРНА РОБОТА № 5	4
1 ЗАГАЛЬНІ ВІДОМОСТІ	4
1.1. Початок з роботи MYSQL 5.7 Server	4
1.2. Отримання довідки з команд MySQL.....	5
1.3. Помилки при введенні	5
1.4. Команди створення об'єктів БД	6
2 ЛАБОРАТОРНЕ ЗАВДАННЯ.....	15
2.1 Початок роботи з СУБД MySQL	15
2.2 Створення нової бази даних	15
2.3 Створення таблиць	16
2.4 Редагування таблиць	17
2.5 Заповнення таблиць	18
2.6 Перегляд змісту створених таблиць	19
2.7 Самостійна розробка бази даних у середовищі СУБД MySQL	20
3 ЗМІСТ ЗВІТУ	21
4 КОНТРОЛЬНІ ПИТАННЯ	21
5 ЛІТЕРАТУРА.....	22

ЛАБОРАТОРНА РОБОТА № 5

СТВОРЕННЯ БАЗИ ДАНИХ У СЕРЕДОВИЩІ СУБД MYSQL

Мета: вивчення процесу створення бази даних у середовищі СУБД MySQL у консольному режимі

1 ЗАГАЛЬНІ ВІДОМОСТІ

1.1. Початок з роботи MySQL 5.7 Server

Завантажити **MySQL 5.7 Command Line Client** за допомогою Головного меню. Вікно **MySQL 5.7 Command Line Client** має вигляд звичайного консольного додатку, якій наведено на рисунку 2.1



Рисунок 1.1 - Початковий екран входу у клієнтський додаток

У вікні консольного додатку зазначити свій пароль. У разі введення правильного паролю в вікно програми має вигляд наведений на рисунку 1.2.

```
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.14-log MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Рисунок 1.2 - Початковий екран командного рядку

1.2. Отримання довідки з команд MySQL

Команди роботи з консоллю наведені у довідці, яка викликається командами `> \h` або `help` або `\?`

У вікні з'являється довідка про основні команди консолі.

Для отримання більш детальної інформації про доступні команди, які розділені по категоріях `> help contents;`

Отримання інформації по категоріях `> help <категорія>;`

Очищення введених команд `- \c;`

1.3. Помилки при введенні

Для MySQL ознакою завершення вираз є крапка з комою, а не кінець рядка (іншими словами, mysql приймає команди без форматування: рядки з командами збираються, але не виконуються до тих пір, поки програма не виявить крапку з комою).

Приклад нескладного виразу, що займає кілька рядків:

```
mysql> SELECT
-> USER ()
->,
-> CURRENT_DATE;
+ -----+ + -----+ +
| USER () | CURRENT_DATE |
+ -----+ + -----+ +
| joesmith @ localhost | 1999-03-18 |
+ -----+ + -----+ +
```

В залежності від дій користувача після введення першого рядка цього запиту мітка командного рядка змінюється (з `mysql>` на `->`). Зміна з `mysql>` на `->` показує, що завершення виразу не введено і очікується його повне введення. Ця мітка надає інформацію про стан програми, що треба вводити далі те, чого чекає mysql.

Для скасування виконання введеної команди `- \c`:

```
mysql> SELECT
-> USER ()
-> \c
mysql>
```

Після введення команди `\c` мітка запрошення програми знову набуває вигляду `mysql>`, показуючи, що програма перейшла в режим очікування команд.

Мітки `'>'` і `'> i '>` використовуються при введенні рядків. В MySQL рядки можна укладати як в одинарні (' '), так і в подвійні (" ") лапки

(наприклад, припустимо написати 'hello' або "goodbye"), до того ж, mysql дозволяє вводити рядкові вирази, що складаються з кількох рядків тексту. Мітка '>' або '>' позначає, що введено рядок, що відкривається символом лапок ' ' або " ", але ще не введено завершальні лапки, які закривають строковий вираз.

Приклад

```
mysql> SELECT * FROM my_table WHERE name = "Smith
AND age <30;
">
```

У таблиці 1.1 наведено всі можливі варіанти виду мітки командного рядка і відповідний їм стан mysql - мітка значення.

Таблиця 1.1 - Варіанти виду мітки командного рядка і відповідні їм стану mysql

mysql>	Очікування нової команди – ввести команду
->	Очікування наступного рядка багаторядкової команди – продовжити введення наступного рядку команди, для закінчення команди ввести ;
'>	Очікування наступного рядка, що починається з одиночної лапки (' ') - продовжити введення наступного строкового рядку, для закінчення команди ввести ;
">	Очікування наступного рядка, що починається з подвійної лапки ("")- - продовжити введення наступного строкового рядку, для закінчення команди ввести ;

1.4. Команди створення об'єктів БД

Створення бази даних

CREATE DATABASE < ім'я бази даних >;

На екрані з'являється повідомлення

```
mysql> create database pet ;
Query OK, 1 row affected (0.02 sec)
```

Рисунок 1.3 – Результат створення нової БД

Перевірка наявності БД на сервері

SHOW DATABASES;

На екрані з'являється повідомлення, у якому перелічуються бази даних, що існують на сервері MySQL

У переліку наявних БД з'являється створена БД Pet.

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| performance_schema |
| pet        |
| test       |
+-----+
5 rows in set (0.14 sec)
```

Рисунок 1.4 – Результат перевірки наявних БД

Перехід до потрібної БД

USE < ім'я бази даних >;

На екрані з'являється повідомлення яке свідчить, що робочою є обрана БД.

```
mysql> use pet;
Database changed
```

Рисунок 1.5 – Перехід до потрібної БД

Перегляд наявних таблиць у робочій БД

SHOW TABLES;

Якщо у робочій БД не має таблиць з'являється повідомлення

```
mysql> use pet;
Database changed
mysql> show tables;
Empty set (0.01 sec)
```

Рисунок 1.6 – Повідомлення при відсутності таблиць у робочій БД

При наявності таблиць відображається їх перелік

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| pet |
| test |
+-----+
5 rows in set (0.14 sec)
```

Рисунок 1.7 – Перелік наявних БД на сервері

Створення таблиці

CREATE TABLE <ім'я таблиці>

(<ім'я атрибуту1> <тип даних> <додаткові параметри>,

.....
< ім'я атрибуту n> <тип даних><додаткові параметри>

PRIMARY KEY (<ім'я атрибуту>)

FOREIGN KEY (<ім'я атрибуту>)

REFERENCE <ім'я таблиці>. <ім'я атрибуту>

);

Можливі додаткові параметри атрибуту

Дозвіл значення NULL -NULL /NOT NULL

Значення по замовченню -

Первинний ключ PRIMARY KEY (Ім'я атрибуту)

Зовнішній ключ - FOREIGN KEY

Обмеження на значення атрибуту

Приклад

```
mysql> create table gruppaa(
-> id_gr int(6) auto_increment,
-> gruppaa varchar(10) not null,
-> primary key(id_gr));
Query OK, 0 rows affected (0.61 sec)
```

Рисунок 1.8 – SQL –код створення таблиці

Визначення структури таблиці**DESCRIBE** <ім'я таблиці>; або**SHOW COLUMNS** <ім'я_таблиці>;

Наприклад:

```
mysql> describe gruppa;
```

Field	Type	Null	Key	Default	Extra
id_gr	int(6)	NO	PRI	NULL	auto_increment
gruppa	varchar(10)	NO		NULL	

```
2 rows in set (0.11 sec)
```

```
mysql> show columns from gruppa;
```

Field	Type	Null	Key	Default	Extra
id_gr	int(6)	NO	PRI	NULL	auto_increment
gruppa	varchar(10)	NO		NULL	

```
2 rows in set (0.01 sec)
```

Рисунок 1.9 – SQL –код перегляду структури таблиці

Перегляд лістингу створеної таблиці

SHOW CREATE TABLE <i'мя_таблиці>;

Наприклад

```
mysql> show create table gruppa;
+-----+-----+
| Table | Create Table |
+-----+-----+
| gruppa | CREATE TABLE `gruppa` (
  `id_gr` int(6) NOT NULL AUTO_INCREMENT,
  `gruppa` varchar(10) NOT NULL,
  PRIMARY KEY (`id_gr`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-----+-----+
1 row in set (0.00 sec)
```

Рисунок 1.10 – SQL-код перегляду лістингу створення таблиці

Вставлення даних у всі стовпці таблиці

INSERT INTO <імя_таблиці> **VALUES** (' значення_першого_стовпця', 'значення_другого_стовпця', ..., 'значення_останнього_стовпця');

```
mysql> insert into gruppa values (1, "КНТ-516");
Query OK, 1 row affected (0.12 sec)

mysql> select * from gruppa;
+-----+-----+
| id_gruppa | gruppa |
+-----+-----+
|          1 | КНТ-516 |
+-----+-----+
1 row in set (0.00 sec)
```

Рисунок 1.11 – SQL –код вставлення даних у таблицю

Вставлення даних у окремі стовпці таблиці

INSERT INTO ім'я_таблиці (ім'я_стовпця, ім'я_стовпця)
VALUES ('значення_першого_стовпця', 'значення_другого_стовпця'.....);

```
mysql> insert into gruppa(gruppa) values ("КНТ-526");
Query OK, 1 row affected (0.09 sec)

mysql> select * from gruppa;
+-----+-----+
| id_gruppa | gruppa |
+-----+-----+
|          1 | КНТ-516 |
|          15 | КНТ-526 |
+-----+-----+
2 rows in set (0.00 sec)
```

Рисунок 1.12 – SQL –код вставлення окремих даних у таблицю

Якщо необхідно вставити декілька рядків, треба вказати кожен рядок зі значеннями через кому у дужках, які теж розділюються комою.

INSERT INTO <ім'я_таблиці> **VALUES** (значення всіх атрибутів першого рядка через кому), (значення всіх атрибутів другого рядка через кому) (значення всіх атрибутів N-ого рядка через кому);

У разі задання у ключового атрибуту параметру **AUTO_INCREMENT** значення цього стовпця можна не задавати.

Наприклад:

```
mysql> insert into gruppa values (2, "КНТ-526"),
-> (3, "КНТ-616"), (4, "КНТ536cn");
Query OK, 3 rows affected (0.07 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> select * from gruppa;
+-----+-----+
| id_gruppa | gruppa |
+-----+-----+
|          1 | КНТ-516 |
|          2 | КНТ-526 |
|          3 | КНТ-616 |
|          4 | КНТ536cn |
|          15 | КНТ-526 |
+-----+-----+
```

Рисунок 1.13—SQL-код вставлення декількох рядків у таблицю

Редагування таблиці***Додавання стовпця в таблицю*****ALTER TABLE** <ім'я_таблиці> **ADD COLUMN**

<ім'я_стовпця> <тип>

Додавання стовпця в таблицю з визначенням місця стовпця**ця****ALTER TABLE** <ім'я_таблиці> **ADD COLUMN**<ім'я_стовпця> <тип> **AFTER** <ім'я_стовпця>

mysql> alter table gruppa add column curs int (4);

Query OK, 0 rows affected (0.64 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> describe gruppa;

Field	Type	Null	Key	Default	Extra
id_gr	int(6)	NO	PRI	NULL	auto_increment
gruppa	varchar(10)	NO		NULL	
curs	int(4)	YES		NULL	

3 rows in set (0.02 sec)

Рисунок 1.14—SQL-код вставлення додавання стовпця у таблицю.

Зміна параметрів стовпця**ALTER TABLE** <ім'я_таблиці> **MODIFY** <ім'я_стовпця> <

нові параметри стовпця>

Наприклад, за умовами попереднього прикладу змінити параметри стовпця curs – зменшити розмір поля int до 4 та додати обмеження not null.

```
mysql> alter table gruppa modify curs int (4) not null;
Query OK, 0 rows affected (0.49 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> describe gruppa;
```

Field	Type	Null	Key	Default	Extra
id_gr	int(6)	NO	PRI	NULL	auto_increment
gruppa	varchar(10)	NO		NULL	
curs	int(4)	NO		NULL	

```
3 rows in set (0.02 sec)
```

Рисунок 1.15—SQL-код зміни параметрів стовпця у таблиці

Видалення стовпця

ALTER TABLE <ім'я_таблиці> **DROP COLUMN**

<ім'я_стовпця>

Більш докладно про можливості оператора **ALTER TABLE** можна передивитись за посиланням <http://www.mysql.ru/docs/man>

Перегляд змісту створених таблиць

Для перегляду змісту твореної таблиці необхідно використувати оператор

SELECT * FROM <ім'я_таблиці>.

Видалення записів з таблиці

Видалити потрібний рядок з таблиці можна за допомогою SQL-запиту

DELETE FROM <ім'я_таблиці>. **WHERE** умови відбору

Запит **DELETE FROM** <ім'я_таблиці> дозволяє знищити всі записи у таблиці, при цьому структура таблиці зберігається у БД.

Копіювання таблиці з одної БД у другу БД одного користувача

CREATE TABLE db2.table2 **SELECT * FROM** db1.table1

CREATE TABLE test.gr **SELECT * FROM** proba.gruppa; - копіювання таблиці gruppa з БД proba у таблицю gr БД test.

```
mysql> use test;
Database changed
mysql> create table test.gr select * from proba.gruppa;
Query OK, 5 rows affected (0.25 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

Рисунок 1.16—SQL-код створення копії таблиці у інший БД

mysql> use test;		mysql> select * from gr;	
Database changed			
mysql> show tables;			
-----+		-----+	
Tables_in_test		id_gruppa gruppa	
-----+		-----+	
gr		1	KHT-516
		2	KHT-526
		3	KHT-616
		4	KHT536сн
-----+		-----+	
row in set (0.00 s)		15	KHT-526

Рисунок 1.17—SQL-код перевірки правильності результатів копіювання

Вставлення нової БД з SQL-файлу

source path/to/file/dump_file_name.sql
 де path/to/file- шлях до файлу дампа
 dump_file_name.sql- ім'я файла дампу

2 ЛАБОРАТОРНЕ ЗАВДАННЯ

2.1 Початок роботи з СУБД MySQL

Завантажити **MySQL 5.7 Command Line Client**

Визначити довідкову інформацію про команди **SELECT DATABASES** та **CREATE**. Записати інформацію у звіт.

Визначити наявність баз даних у на сервері за допомогою команди

SHOW DATABASES; або SELECT DATABASES;

У вікні консольного додатку відображається список баз даних, які доступні користувачеві

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| decanat   |
| mysql     |
| performance_schema |
| sys       |
+-----+
5 rows in set (0.00 sec)
```

Рисунок 2.1 □ Відображення БД, які наявні на сервері

Для закінчення сеансу роботи ввести команду **EXIT** або **QUIT** або \q.

2.2 Створення нової бази даних

Для створення нової бази даних у командному рядку ввести команду

CREATE DATABASE PROBA;

Не забувайте закінчувати рядок символом “;”.

У разі правильно введеної команди на екрані з’являється повідомлення.

```
mysql> create database PROBA;  
Query OK, 1 row affected (0.00 sec)
```

Рисунок 2.2 – Створення нової БД на сервері

Перевірити створення нової БД за допомогою команди

SHOW DATABASES;

Переконатись у наявності створюваної БД.

Для видалення створеної БД PROBA ввести команду

DROP DATABASE PROBA

Створити нову базу даних **PET** (тварини)

CREATE DATABASES PET;

Перейти до БД **PET**

USE PET;

2.3 Створення таблиць

Та переконатись, що БД PET не має таблиць.

SHOW TABLES;

або **SHOW TABLES FROM PET;**

На екрані з'явиться повідомлення, що у БД PET не має таблиць.

Створити таблицю PET

CREATE TABLE PET

(id_pet INT(10) **AUTO_INCREMENT**,

PET VARCHAR(25) **NOT NULL**,

PRIMARY KEY(id_pet));

Для перевірки створення таблиці за вимогами ввести оператор **DESCRIBE PET;**

Результати виконання введених команд відображені на рисунку 2.5

```
mysql> create table pet
-> ( id_pet int(10) auto_increment,
-> pet varchar (10) not null,
-> primary key (id_pet)
-> );
Query OK, 0 rows affected (0.52 sec)
```

Рисунок 2.3 – Створення таблиці PET

```
mysql> describe pet;
```

Field	Type	Null	Key	Default	Extra
id_pet	int(10)	NO	PRI	NULL	auto_increment
pet	varchar(10)	NO		NULL	

```
2 rows in set (0.16 sec)
```

Рисунок 2.4 –Перевірка структури таблиці PET

Зверніть увагу, одну команду можна писати в кілька рядків, використовуючи клавішу Enter (MySQL автоматично підставляє символ нового рядка ->), і тільки після роздільник (крапки з комою) натискання клавіші Enter призводить до виконання запиту.

Створити таблицю Food зі стовпцями id_food –ціле, автоматичне збільшення, первинний ключ, food текстове поле розміром 25.

Переконайтесь у правильності створення таблиці, переглянути лістинг створення таблиці та відповідність їх стовпців зазначеним умовам.

Самостійно створити таблицю Food(id_food, food), id_food – первинний ключ.

2.4 Редагування таблиць

Додати стовпець id_food у таблицю PET –числовий , зовнішній ключ з таблиці food.

ALTER TABLE PETS

ADD COLUMN id_food INT(10) NOT NULL;

Та переглянути результат внесених змін

```
mysql> describe pets;
```

Field	Type	Null	Key	Default	Extra
id_pets	int(10)	NO	PRI	NULL	auto_increment
pets	varchar(25)	NO		0	
id_food	int(10)	NO	MUL	0	

Рисунок 2.5 –Перевірка зміни структури таблиці PET

Видалення стовпця

ALTER TABLE PETS DROP COLUMN id_food;

```
mysql> describe pets;
```

Field	Type	Null	Key	Default	Extra
id_pets	int(10) unsigned	NO	PRI	NULL	auto_increment
pets	varchar(25)	NO		0	
id_food	int(10) unsigned	NO	MUL	0	

```
mysql> alter table pets drop column id_food;
```

```
Query OK, 0 rows affected (0.87 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> describe pets;
```

Field	Type	Null	Key	Default	Extra
id_pets	int(10) unsigned	NO	PRI	NULL	auto_increment
pets	varchar(25)	NO		0	

Рисунок 2.6 –Перевірка видалення стовпця зі таблиці PET

2.5 Заповнення таблиць

Для внесення даних у таблицю використовується оператор

INSERT

INSERT INTO FOOD VALUES (1, 'мясо');

```
mysql> insert into food values ( 1, 'мясо');
```

```
Query OK, 1 row affected (0.09 sec)
```

Рисунок 2.7 –Команда додавання даних у таблицю FOOD

Внесення інформації у окремі стовпці

INSERT INTO FOOD (food) VALUES ('рыба');

Тому що перший стовпець є первинним ключем, його можна не визначати.

Щоб додати відразу кілька рядків, треба просто перераховувати дужки зі значеннями через кому:

INSERT INTO FOOD VALUES

(7,'молоко'), (8,'хлеб');

Вводити дані, що додаються у таблицю, необхідно по рядках.

```
mysql> insert into food values
-> (7,'молоко'),
-> (8,'хлеб');
Query OK, 2 rows affected (0.10 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

Рисунок 2.8 –Додавання даних у таблицю FOOD до декількох рядків

2.6 Перегляд змісту створених таблиць

Переглянути зміст таблиці **food** можна за допомогою SQL-запиту

SELECT * FROM FOOD

Видалення даних з таблиці

Видалити потрібний рядок з таблиці можна за допомогою SQL-запиту

DELETE FROM FOOD WHERE id_food =7

```
mysql> select * from food;
+-----+-----+
| id_food | food |
+-----+-----+
|      1 | мясо |
|      7 | молоко |
|      8 | хлеб |
|     10 | рыба |
+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> delete from food where id_food = 7;
Query OK, 1 row affected (0.12 sec)
```

```
mysql> select * from food;
+-----+-----+
| id_food | food |
+-----+-----+
|      1 | мясо |
|      8 | хлеб |
|     10 | рыба |
+-----+-----+
```

Рисунок 2.9 – Видалення даних з таблиці FOOD

2.7 Самостійна розробка бази даних у середовищі СУБД MySQL

1. Створити базу даних Decanat та перейти до неї.
2. У створеній БД Decanat створити таблиці Кафедра, Посада, Звання, Група та Іспити відповідно схемі бази даних, параметри полів таблиць визначити самостійно. Імена таблиць визначати латиницею.



Рисунок 2.10 □ Схема бази даних Деканат

3. У таблицю Група вставити записи КНТ-513, КНТ-523, КНТ-623, КНТ-613

У таблицю Кафедра вставити записи фізика, радіотехніка, математика, ксм.

У таблицю Посада вставити дані доцент, професор, старший викладач.

У таблицю Звання вставити записи доцент, професор.

У таблицю Іспити вставити записи ОБД, Програмування, Архітектура ПК, Іноземна мова, Інформатика, Комп'ютерні мережі, Лінійна алгебра

4. * Створити таблиці Студенти, Викладачі, Сесія та Гуртожиток згідно заданої схеми

5. *Таблицю Студенти, Викладачі, Сесія та Гуртожиток заповнити за власним розсудом трьома записами.

6. Переглянути зміст створених таблиць

7. Створити нову БД Decanat1
8. Скопіювати до створеної БД таблиці Кафедра, Посада, Звання, Група та Іспити з БД Decanat, переглянути результати копіювання.
9. Пред'явити роботу викладачу та вилучити БД Decanat.
10. Вставити нову БД з файлу dump.sql, якій знаходиться у папці ...\\common\\OBD.
11. Переглянути зміст нової БД movedb та переконатись у невідповідності кодування БД movedb з кодуванням серверу Mysql.

3 ЗМІСТ ЗВІТУ

- 3.1. Тема та мета лабораторної роботи.
- 3.2. SQL – коди створення та редагування таблиць
- 3.3. Відповіді на контрольні питання.

4 КОНТРОЛЬНІ ПИТАННЯ

- 4.1. Команди перегляду наявних баз даних
- 4.2. Створення бази даних
- 4.3. Перехід у зазначену базу даних
- 4.4. Створення таблиці
- 4.5. Параметри атрибутів при створенні таблиці
- 4.6. Перевірка створення таблиці
- 4.7. Вставлення зазначених даних у рядок
- 4.8. Перегляд змісту таблиці
- 4.9. Вставлення даних у рядок таблиці
- 4.10. Вставлення зазначених даних у рядок, одночасно декілька рядків
- 4.11. Вставлення зазначених даних
- 4.12. Видалення даних з таблиці
- 4.13. Очищенні таблиці
- 4.14. Видалення таблиці
- 4.15. Перегляд коду створення таблиць
- 4.16. Копіювання змісту таблиць з однієї БД у іншу одного користувача
- 4.17. Перегляд структури таблиці
- 4.18. Перегляд кодування даних

5 ЛИТЕРАТУРА

- 5.1. <http://www.site-do.ru/db/sql3.php>
- 5.2. <https://dev.mysql.com/doc>
- 5.3. <http://kafinfor.petrso.ru/mysql/index.htm>
- 5.4. <http://www.site-do.ru/db/sql3.php>
- 5.5. <http://phpclub.ru/mysql/doc>
- 5.6. <http://www.sql.ru/>
- 5.7. <http://www.cyberforum.ru>