

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

**МЕТОДИЧНІ ВКАЗІВКИ**  
**до виконання лабораторних робіт**  
**за темою «Вивчення основ роботи з СУБД MySQL:**  
**Основні засоби DDL та DML мови SQL»**

для студентів спеціальностей  
121 «Інженерія програмного забезпечення»  
122 «Комп'ютерні науки»  
126 «Інформаційні системи та технології»

Харків 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

**МЕТОДИЧНІ ВКАЗІВКИ**  
**до виконання лабораторних робіт**  
**за темою «Вивчення основ роботи з СУБД MySQL:**  
**Основні засоби DDL та DML мови SQL»**

для студентів спеціальностей  
121 «Інженерія програмного забезпечення»  
122 «Комп'ютерні науки»  
126 «Інформаційні системи та технології»

Затверджено  
Редакційно-видавничою  
радою університету,  
протокол № 2 від 29.06.2021

Харків  
НТУ «ХП»  
2021

Методичні вказівки до виконання лабораторних робіт за темою «Вивчення основ роботи з СУБД MySQL: Основні засоби DDL та DML мови SQL» для студентів спеціальностей 121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки» та 126 «Інформаційні системи та технології» / уклад. Д.Л. Орловський, А.М. Копп. – Харків: НТУ «ХПІ», 2021. – 31 с.

Укладачі: Д.Л. Орловський

А.М. Копп

Рецензент Малько М.М.

Кафедра програмної інженерії та інформаційних технологій  
управління

## ЗМІСТ

ВСТУП.....	4
ЛАБОРАТОРНА РОБОТА 1. СТВОРЕННЯ БАЗИ ДАНИХ ЗАСОБАМИ МОВИ SQL .....	5
ЛАБОРАТОРНА РОБОТА 2. МАНІПУЛЮВАННЯ ДАНИМИ ЗАСОБАМИ МОВИ SQL: ДОДАВАННЯ, ОНОВЛЕННЯ ТА ВИДАЛЕННЯ ДАНИХ.....	11
ЛАБОРАТОРНА РОБОТА 3. МАНІПУЛЮВАННЯ ДАНИМИ ЗАСОБАМИ МОВИ SQL: ЗАПИТИ SELECT ТА ЇХ ОСНОВНІ ОСОБЛИВОСТІ .....	17
ЛАБОРАТОРНА РОБОТА 4. СТВОРЕННЯ ТА ВИКОРИСТАННЯ ПРЕДСТАВЛЕНЬ.....	28
СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ.....	31

## ВСТУП

Система управління базами даних (СУБД) MySQL є вільно розповсюджуваною реляційною СУБД, розробку та підтримку якої здійснює корпорація Oracle. Від самого початку MySQL розроблялася шведською компанією MySQL AB, яка потім була придбана компанією Sun Microsystems, яка, в свою чергу, згодом була поглинена Oracle.

СУБД MySQL розповсюджується як під ліцензією вільного програмного забезпечення GNU General Public Licence (GPL), так і під власною комерційною ліцензією. За умовами GPL, програмне забезпечення, що використовує бібліотеки MySQL, також повинне розповсюджуватися за ліцензією GPL. Для випадків, коли розробники не бажають відкривати вихідний код свого програмного забезпечення, передбачена комерційна ліцензія. Перевагою комерційної ліцензії є якісна сервісна підтримка. У протиположності ліцензування MySQL компанією Oracle та для забезпечення вільного статусу СУБД, було створено відгалуження від MySQL, яке отримало назву MariaDB. Ця СУБД підтримує високу сумісність з MySQL, забезпечуючи точну відповідність інтерфейсу програмування, так званого API (Application Programming Interface), та команд MySQL.

СУБД MySQL є чудовим рішенням для малих, середніх, та інколи навіть великих програмних систем. Також MySQL є складовою частиною стеків розробки веб-застосунків WAMP (Windows, Apache, MySQL, PHP/Perl/Python) та LAMP (Linux, Apache, MySQL, PHP/Perl/Python). Дана СУБД включена до складу багатьох готових збірок серверів, призначених для розробки веб-застосунків, таких як XAMPP (який пропонується використовувати у даному лабораторному практикумі), OpenServer, Denwer тощо. Однак останнім часом, саме через підтримку відкритості, розробники збірок серверів та хостинг провайдери все частіше включають MariaDB у WAMP та LAMP стеки.

Зазвичай MySQL застосовується у ролі сервера, до якого звертаються локальні або віддалені клієнти. Проте дистрибутив містить і бібліотеку, що забезпечує розгортання внутрішнього сервера для автономних застосунків. У даному лабораторному практикумі розглядається знайомство з основними можливостями MySQL у ролі клієнт-серверної СУБД.

# ЛАБОРАТОРНА РОБОТА 1. СТВОРЕННЯ БАЗИ ДАНИХ ЗАСОБАМИ МОВИ SQL

**Мета роботи:** навчитися створювати та зв'язувати таблиці бази даних з використанням СУБД MySQL.

## Хід роботи

### 1. Встановити MySQL

Для спрощення установки та подальшого використання системи управління базами даних (СУБД) MySQL рекомендується встановити один з вільно розповсюджуваних WAMP (Windows, Apache, MySQL, PHP) або LAMP (Linux, Apache, MySQL, PHP) серверів, наприклад OpenServer або XAMPP. В подальших прикладах виконання лабораторних робіт буде використовуватися сервер XAMPP.

Після того, як сервер XAMPP буде встановлено, необхідно запустити панель управління компонентами серверу, запустити MySQL та відкрити командний рядок сервера (рисунок 1.1).

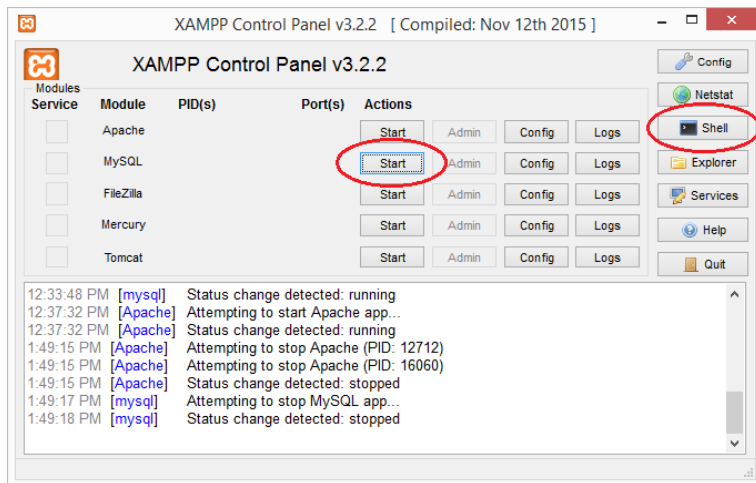


Рисунок 1.1 – Панель управління сервером XAMPP

### 2. Підключитися до MySQL та створити базу даних

У командному рядку сервера MySQL необхідно ввести команду:  
mysql -u root -p

Після цього необхідно ввести пароль (рисунок 1.2). Зазвичай пароль користувача root є порожнім, тому можна просто натиснути Enter.

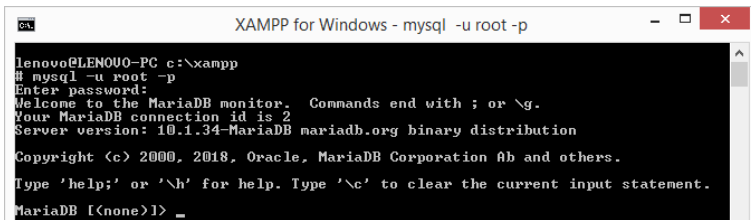


Рисунок 1.2 – Підключення до MySQL

Для того, щоб відключитися від MySQL необхідно використати команду exit.

Основними командами, які будуть використовуватися час від часу при роботі з MySQL є наступні:

- 1) USE database – вибрати базу даних (БД) для подальшої роботи;
- 2) SHOW DATABASES – отримати перелік баз даних;
- 3) SHOW TABLES – отримати перелік таблиць для обраної БД;
- 4) SHOW COLUMNS FROM table – отримати інформацію про таблицю;
- 5) SHOW INDEX FROM table – отримати інформацію про індекси, визначені для таблиці.

Створення бази даних виконується за допомогою команди:

```
CREATE DATABASE supply;
```

Виконання даної команди дозволить створити базу даних, робота з якою буде розглядатися у лабораторному практикумі. Перевірити створення бази даних можна за допомогою команди SHOW DATABASES.

### **3. Створити таблиці бази даних та зв'язати їх**

Для вивчення особливостей роботи з СУБД MySQL буде розглядатися база даних деякого підприємства, що придбає товари у різних постачальників. Придбання товарів виконується партіями та оформляється у вигляді договорів на поставку. Кожний договір має

унікальний номер та укладається лише з одним постачальником. У документах по кожному договору вказується назва, розмір поставленої партії та ціна (у грн.).

Створення таблиць виконується за допомогою оператора CREATE TABLE. Таким чином, для бази даних, що розглядається, необхідно створити наступні таблиці:

```
CREATE TABLE supplier (  
  supplier_id int NOT NULL,  
  supplier_address varchar(100) NOT NULL,  
  supplier_phone varchar(20) NOT NULL,  
  PRIMARY KEY (supplier_id)  
) ENGINE=InnoDB;  
  
CREATE TABLE supplier_person (  
  supplier_id int NOT NULL,  
  supplier_last_name varchar(20) NOT NULL,  
  supplier_first_name varchar(20) NOT NULL,  
  supplier_middle_name varchar(20) NOT NULL,  
  PRIMARY KEY (supplier_id),  
  FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id)  
) ENGINE=InnoDB;  
  
CREATE TABLE supplier_org (  
  supplier_id int NOT NULL,  
  supplier_org_name varchar(20) NOT NULL,  
  PRIMARY KEY (supplier_id),  
  FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id)  
) ENGINE=InnoDB;  
  
CREATE TABLE contract (  
  contract_number int NOT NULL AUTO_INCREMENT,  
  contract_date timestamp NOT NULL,  
  supplier_id int NOT NULL,  
  contract_note varchar(100),  
  PRIMARY KEY (contract_number),  
  FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id)  
) ENGINE=InnoDB;  
  
CREATE TABLE supplied (  
  contract_number int NOT NULL,  
  supplied_product varchar(20) NOT NULL,  
  supplied_amount decimal(4,0) NOT NULL,  
  supplied_cost decimal(8,2) NOT NULL,  
  PRIMARY KEY (contract_number, supplied_product),  
  FOREIGN KEY (contract_number) REFERENCES contract(contract_number)  
) ENGINE=InnoDB;
```

Перевірити створені таблиці у базі даних supply (рисунок 1.3).



```
XAMPP for Windows - mysql -u root
MariaDB [(none)]> USE supply; SHOW TABLES;
Database changed
+-----+
| Tables_in_supply |
+-----+
| contract          |
| supplied          |
| supplier          |
| supplier_org      |
| supplier_person   |
+-----+
5 rows in set (0.00 sec)
```

Рисунок 1.3 – Створені таблиці у базі даних

#### 4. Модифікація структури таблиць

Змінити структуру існуючої таблиці можна за допомогою оператора ALTER TABLE. Припустимо, що необхідно створити ще одну таблицю у базі даних supply, яка буде призначена для збереження даних про факти виконання поставок за договорами (рисунок 1.4).

```
CREATE TABLE contract_delivered (
    contract_number int NOT NULL,
    delivery_date timestamp NOT NULL,
    delivery_note varchar(100),
    PRIMARY KEY (contract_number)
) ENGINE=InnoDB;
```

```
XAMPP for Windows - mysql -u root
MariaDB [supply]> CREATE TABLE contract_delivered (
  -> contract_number int NOT NULL,
  -> delivery_date timestamp NOT NULL,
  -> delivery_note varchar(100),
  -> PRIMARY KEY (contract_number)
  -> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0.04 sec)

MariaDB [supply]> SHOW TABLES;
+-----+
| Tables_in_supply |
+-----+
| contract          |
| contract_delivered |
| supplied          |
| supplier          |
| supplier_org      |
| supplier_person   |
+-----+
6 rows in set (0.00 sec)
```

Рисунок 1.4 – Перевірка наявності створеної таблиці

Для того, щоб зв'язати створену таблицю contract\_delivered з таблицею contract необхідно застосувати команду ALTER TABLE (рисунок 1.5).

```
ALTER TABLE contract_delivered
ADD CONSTRAINT contract_number_fk FOREIGN KEY (contract_number)
REFERENCES contract(contract_number);
```

```
XAMPP for Windows - mysql -u root
MariaDB [supply]> ALTER TABLE contract_delivered
-> ADD CONSTRAINT contract_number_fk FOREIGN KEY (contract_number) REFERENCE
S contract(contract_number) ON DELETE RESTRICT;
Query OK, 0 rows affected (0.09 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Рисунок 1.5 – Виконання команди ALTER TABLE

## 5. Видалення таблиць

Видалити таблицю можна за допомогою оператора DROP TABLE. Оскільки створена таблиця contract\_delivered не буде використовуватися у подальшій роботі, її можна видалити за допомогою даної команди (рисунок 1.6).

```
XAMPP for Windows - mysql -u root
MariaDB [supply]> DROP TABLE contract_delivered;
Query OK, 0 rows affected (0.04 sec)
MariaDB [supply]> SHOW TABLES;
+-----+
| Tables_in_supply |
+-----+
| contract          |
| supplied         |
| supplier         |
| supplier_org     |
| supplier_person  |
+-----+
5 rows in set (0.00 sec)
```

Рисунок 1.6 – Перевірка результатів видалення таблиці

## 6. Оформити звіт з лабораторної роботи

У звіт включити основні етапи виконання лабораторної роботи та знімки екрану, що їх демонструють.

## 7. Питання для самоконтролю

1. Як отримати доступ до командного рядка сервера MySQL?
2. Як встановити з'єднання з сервером MySQL, використовуючи ім'я та пароль певного користувача?
3. Назвіть основні команди адміністрування сервера БД MySQL та їх призначення.
4. За допомогою якої команди створюється база даних? Як можна перевірити створення БД?
5. За допомогою яких операторів мови SQL виконується створення та зв'язування таблиць?
6. За допомогою якого оператора мови SQL виконується модифікація структури таблиці?
7. За допомогою якого оператора мови SQL виконується видалення таблиці з бази даних?

8. Яким чином можна перевірити наявність або відсутність створених або видалених таблиць відповідно?

9. Яким чином можна задати ім'я зовнішнього ключа під час зв'язування таблиць?

10. Які недоліки містить розглянута структура БД? Як їх усунути?

## ЛАБОРАТОРНА РОБОТА 2. МАНІПУЛЮВАННЯ ДАНИМИ ЗАСОБАМИ МОВИ SQL: ДОДАВАННЯ, ОНОВЛЕННЯ ТА ВИДАЛЕННЯ ДАНИХ

**Мега роботи:** навчитися використовувати оператори мови SQL, призначені для додавання, оновлення та видалення даних, на прикладі СУБД MySQL.

### Хід роботи

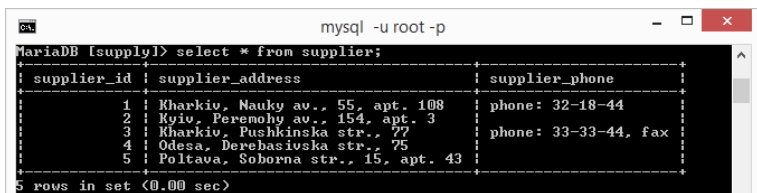
#### 1. Додавання даних до створеної бази даних

Для додавання даних використовується оператор INSERT.

Наступні команди дозволяють заповнити дані про постачальників у створеній базі даних:

```
INSERT INTO supplier (supplier_id, supplier_address, supplier_phone)
VALUES (1, 'Kharkiv, Nauky av., 55, apt. 108', 'phone: 32-18-44');
INSERT INTO supplier (supplier_id, supplier_address, supplier_phone)
VALUES (2, 'Kyiv, Peremohy av., 154, apt. 3', '');
INSERT INTO supplier (supplier_id, supplier_address, supplier_phone)
VALUES (3, 'Kharkiv, Pushkinska str., 77', 'phone: 33-33-44, fax: 22-12-33');
INSERT INTO supplier (supplier_id, supplier_address, supplier_phone)
VALUES (4, 'Odesa, Derebasivska str., 75', '');
INSERT INTO supplier (supplier_id, supplier_address, supplier_phone)
VALUES (5, 'Poltava, Soborna str., 15, apt. 43', '');
```

Перевірити записи, створені у таблиці supplier (рисунок 2.1).



```
mysql -u root -p
MariaDB [supply]> select * from supplier;
+-----+-----+-----+
| supplier_id | supplier_address | supplier_phone |
+-----+-----+-----+
| 1 | Kharkiv, Nauky av., 55, apt. 108 | phone: 32-18-44 |
| 2 | Kyiv, Peremohy av., 154, apt. 3 | |
| 3 | Kharkiv, Pushkinska str., 77 | phone: 33-33-44, fax: 22-12-33 |
| 4 | Odesa, Derebasivska str., 75 | |
| 5 | Poltava, Soborna str., 15, apt. 43 | |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Рисунок 2.1 – Записи, створені у таблиці supplier

Наступні команди дозволяють заповнити дані про постачальників-фізичних осіб у створеній базі даних:

```

INSERT INTO supplier_person (supplier_id, supplier_last_name,
supplier_first_name, supplier_middle_name) VALUES (1, 'Petrov', 'Pavlo',
'Petrovych');
INSERT INTO supplier_person (supplier_id, supplier_last_name,
supplier_first_name, supplier_middle_name) VALUES (3, 'Ivanov', 'Illia',
'Illych');
INSERT INTO supplier_person (supplier_id, supplier_last_name,
supplier_first_name, supplier_middle_name) VALUES (5, 'Sydorov', 'Serhii',
'Stepanovych');

```

Перевірити записи, створені у таблиці `supplier_person` (рисунок 2.2).

```

mysql -u root -p
MariaDB [supply]> select * from supplier_person;
+-----+-----+-----+-----+
| supplier_id | supplier_last_name | supplier_first_name | supplier_middle_name |
+-----+-----+-----+-----+
| 1 | Petrov | Pavlo | Petrovych |
| 3 | Ivanov | Illia | Illych |
| 5 | Sydorov | Serhii | Stepanovych |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

Рисунок 2.2 – Записи, створені у таблиці `supplier_person`

Наступні команди дозволяють заповнити дані про постачальників-юридичних осіб у створеній базі даних:

```

INSERT INTO supplier_org (supplier_id, supplier_org_name) VALUES (2,
'Interfruit Ltd. ');
INSERT INTO supplier_org (supplier_id, supplier_org_name) VALUES (4,
'Transservice LLC');

```

Перевірити записи, створені у таблиці `supplier_org` (рисунок 2.3).

```

mysql -u root -p
MariaDB [supply]> select * from supplier_org;
+-----+-----+
| supplier_id | supplier_org_name |
+-----+-----+
| 2 | Interfruit Ltd. |
| 4 | Transservice LLC |
+-----+-----+
2 rows in set (0.00 sec)

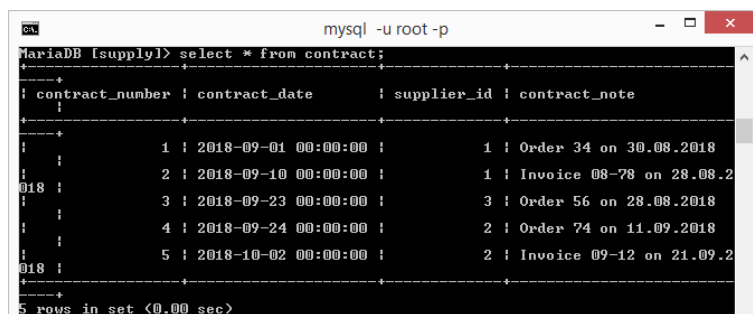
```

Рисунок 2.3 – Записи, створені у таблиці `supplier_org`

Наступні команди дозволяють заповнити дані про укладені договори у створеній базі даних:

```
INSERT INTO contract (contract_date, supplier_id, contract_note) VALUES ('2018-09-01', 1, 'Order 34 on 30.08.2018');
INSERT INTO contract (contract_date, supplier_id, contract_note) VALUES ('2018-09-10', 1, 'Invoice 08-78 on 28.08.2018');
INSERT INTO contract (contract_date, supplier_id, contract_note) VALUES ('2018-09-23', 3, 'Order 56 on 28.08.2018');
INSERT INTO contract (contract_date, supplier_id, contract_note) VALUES ('2018-09-24', 2, 'Order 74 on 11.09.2018');
INSERT INTO contract (contract_date, supplier_id, contract_note) VALUES ('2018-10-02', 2, 'Invoice 09-12 on 21.09.2018');
```

Перевірити записи, створені у таблиці contract (рисунок 2.4).



```
mysql -u root -p
MariaDB [supply]> select * from contract;
+-----+-----+-----+-----+
| contract_number | contract_date       | supplier_id | contract_note                |
+-----+-----+-----+-----+
| 1 | 2018-09-01 00:00:00 | 1 | Order 34 on 30.08.2018      |
| 2 | 2018-09-10 00:00:00 | 1 | Invoice 08-78 on 28.08.2018 |
| 3 | 2018-09-23 00:00:00 | 3 | Order 56 on 28.08.2018      |
| 4 | 2018-09-24 00:00:00 | 2 | Order 74 on 11.09.2018      |
| 5 | 2018-10-02 00:00:00 | 2 | Invoice 09-12 on 21.09.2018 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Рисунок 2.4 – Записи, створені у таблиці contract

Наступні команди дозволяють заповнити дані про поставлені товари у створеній базі даних:

```

INSERT INTO supplied (contract_number,supplied_product,supplied_amount,
supplied_cost) VALUES (1, 'TV', 10, 1300);
INSERT INTO supplied (contract_number,supplied_product,supplied_amount,
supplied_cost) VALUES (1, 'Audio Player', 25, 700);
INSERT INTO supplied (contract_number,supplied_product,supplied_amount,
supplied_cost) VALUES (1, 'Video Player', 12, 750);
INSERT INTO supplied (contract_number,supplied_product,supplied_amount,
supplied_cost) VALUES (2, 'Stereo System', 11, 500);
INSERT INTO supplied (contract_number,supplied_product,supplied_amount,
supplied_cost) VALUES (2, 'Audio Player', 5, 450);
INSERT INTO supplied (contract_number,supplied_product,supplied_amount,
supplied_cost) VALUES (2, 'Video Player', 8, 450);
INSERT INTO supplied (contract_number,supplied_product,supplied_amount,
supplied_cost) VALUES (3, 'TV', 52, 900);
INSERT INTO supplied (contract_number,supplied_product,supplied_amount,
supplied_cost) VALUES (3, 'Audio Player', 11, 550);
INSERT INTO supplied (contract_number,supplied_product,supplied_amount,
supplied_cost) VALUES (3, 'Monitor', 85, 550);
INSERT INTO supplied (contract_number,supplied_product,supplied_amount,
supplied_cost) VALUES (4, 'TV', 56, 990);
INSERT INTO supplied (contract_number,supplied_product,supplied_amount,
supplied_cost) VALUES (4, 'Audio Player', 22, 320);
INSERT INTO supplied (contract_number,supplied_product,supplied_amount,
supplied_cost) VALUES (4, 'Printer', 41, 350);
INSERT INTO supplied (contract_number,supplied_product,supplied_amount,
supplied_cost) VALUES (5, 'TV', 14, 860);
INSERT INTO supplied (contract_number,supplied_product,supplied_amount,
supplied_cost) VALUES (5, 'Audio Player', 33, 580);
INSERT INTO supplied (contract_number,supplied_product,supplied_amount,
supplied_cost) VALUES (5, 'Video Player', 17, 850);

```

Перевірити записи, створені у таблиці supplied (рисунок 2.5).

```

mysql -u root -p
MariaDB [supply]> select * from supplied;
+-----+-----+-----+-----+
| contract_number | supplied_product | supplied_amount | supplied_cost |
+-----+-----+-----+-----+
| 1 | Audio Player | 25 | 700.00 |
| 1 | TV | 10 | 1300.00 |
| 1 | Video Player | 12 | 750.00 |
| 2 | Audio Player | 5 | 450.00 |
| 2 | Stereo System | 11 | 500.00 |
| 2 | Video Player | 8 | 450.00 |
| 3 | Audio Player | 11 | 550.00 |
| 3 | Monitor | 85 | 550.00 |
| 3 | TV | 52 | 900.00 |
| 4 | Audio Player | 22 | 320.00 |
| 4 | Printer | 41 | 350.00 |
| 4 | TV | 56 | 990.00 |
| 5 | Audio Player | 33 | 580.00 |
| 5 | TV | 14 | 860.00 |
| 5 | Video Player | 17 | 850.00 |
+-----+-----+-----+-----+
15 rows in set (0.00 sec)

```

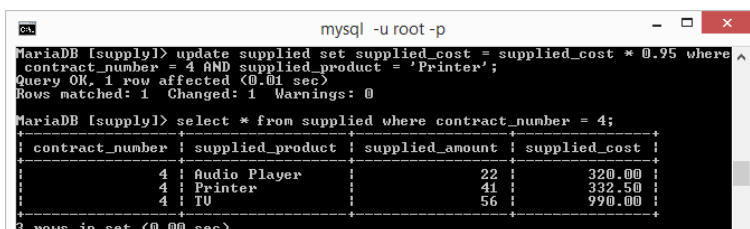
Рисунок 2.5 – Записи, створені у таблиці supplied

## 2. Оновлення даних у базі даних

Оновлення даних (зміна значень полів у існуючих записах) у БД виконується за допомогою оператора UPDATE.

Наприклад, якщо необхідно зменшити вартість принтеру який був поставлений за договором з номером 4, на 5%, команда буде наступною (рисунок 2.6):

```
UPDATE supplied
SET supplied_cost = supplied_cost * 0.95
WHERE contract_number = 4 AND supplied_product = 'Printer';
```



```
mysql -u root -p
MariaDB [supply]> update supplied set supplied_cost = supplied_cost * 0.95 where
contract_number = 4 AND supplied_product = 'Printer';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [supply]> select * from supplied where contract_number = 4;
+-----+-----+-----+-----+
| contract_number | supplied_product | supplied_amount | supplied_cost |
+-----+-----+-----+-----+
| 4 | Audio Player | 22 | 320.00 |
| 4 | Printer | 41 | 332.50 |
| 4 | TU | 56 | 990.00 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

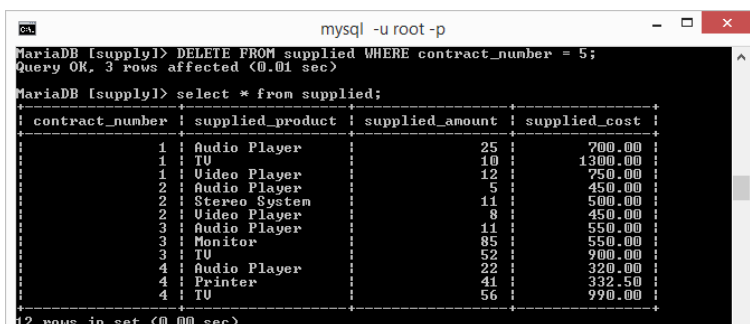
Рисунок 2.6 – Результати виконання запиту UPDATE

## 3. Видалення даних з бази даних

Для видалення даних з таблиць бази даних застосовується оператор DELETE.

Наприклад, для видалення поставлених товарів, які відповідають договору з номером 5, необхідно виконати наступну команду (Рисунок 2.7):

```
DELETE FROM supplied WHERE contract_number = 5;
```



```
mysql -u root -p
MariaDB [supply]> DELETE FROM supplied WHERE contract_number = 5;
Query OK, 3 rows affected (0.01 sec)

MariaDB [supply]> select * from supplied;
+-----+-----+-----+-----+
| contract_number | supplied_product | supplied_amount | supplied_cost |
+-----+-----+-----+-----+
| 1 | Audio Player | 25 | 700.00 |
| 1 | TU | 10 | 1300.00 |
| 1 | Video Player | 12 | 750.00 |
| 2 | Audio Player | 5 | 450.00 |
| 2 | Stereo System | 11 | 500.00 |
| 2 | Video Player | 8 | 450.00 |
| 3 | Audio Player | 11 | 550.00 |
| 3 | Monitor | 85 | 550.00 |
| 3 | TU | 52 | 900.00 |
| 4 | Audio Player | 22 | 320.00 |
| 4 | Printer | 41 | 332.50 |
| 4 | TU | 56 | 990.00 |
+-----+-----+-----+-----+
12 rows in set (0.00 sec)
```

Рисунок 2.7 – Результати виконання запиту DELETE



Відновіть видалені записи за допомогою команд INSERT.

#### **4. Оформити звіт з лабораторної роботи**

У звіт включити основні етапи виконання лабораторної роботи та знімки екрану, що їх демонструють.

#### **5. Питання для самоконтролю**

1. Наведіть структуру та приклади використання команди INSERT.

2. Наведіть структуру та приклади використання команди UPDATE.

3. Наведіть структуру та приклади використання команди DELETE.

4. Яким чином можна оновити усі записи у таблиці БД?

5. Яким чином можна видалити усі записи з таблиці БД?

6. Яким чином можна видалити останні 20 укладених договорів?

7. Яким чином можна збільшити ціну на 15% для 5 найдешевших товарів, які були поставлені за певним договором?

8. Якою повинна бути структура команди INSERT для того, щоб нові записи з дублюючим ключем відкидалися без генерації помилки?

# ЛАБОРАТОРНА РОБОТА 3. МАНІПУЛЮВАННЯ ДАНИМИ ЗАСОБАМИ МОВИ SQL: ЗАПИТИ SELECT ТА ЇХ ОСНОВНІ ОСОБЛИВОСТІ

**Мега роботи:** навчитися використовувати оператор SELECT мови SQL, призначений для вибірки даних, на прикладі СУБД MySQL.

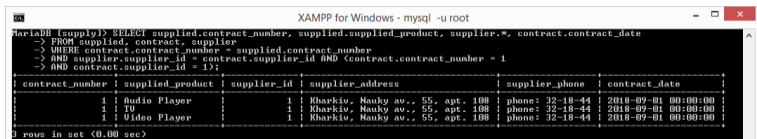
## Хід роботи

### 1. Створити та виконати запити SQL SELECT

#### Запит 1

Сформувати список товарів, поставлених постачальником 1 (Петров П. П.) за договором 1 (рисунок 3.1).

```
SELECT supplied.contract_number, supplied.supplied_product, supplier.*, contract.contract_date
FROM supplied, contract, supplier
WHERE contract.contract_number = supplied.contract_number
AND supplier.supplier_id = contract.supplier_id AND (contract.contract_number = 1
AND contract.supplier_id = 1);
```



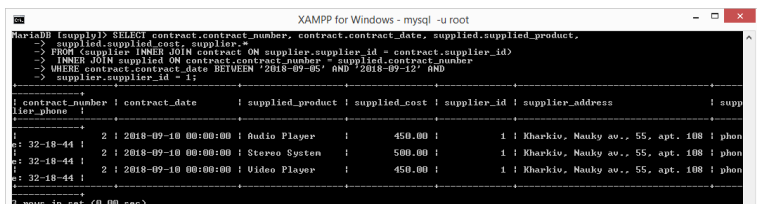
contract_number	supplied_product	supplier_id	supplier_address	supplier_phone	contract_date
1	Audio Player	1	Kharkiv, Nauky av., 55, apt. 108	phone: 32-18-44	2018-09-01 00:00:00
1	TV	1	Kharkiv, Nauky av., 55, apt. 108	phone: 32-18-44	2018-09-01 00:00:00
1	Video Player	1	Kharkiv, Nauky av., 55, apt. 108	phone: 32-18-44	2018-09-01 00:00:00

Рисунок 3.1 – Результат виконання запиту 1

#### Запит 2

Сформувати список товарів, які були поставлені постачальником 1 (Петров П. П.) у період з 2018-09-05 по 2018-09-12 (рисунок 3.2).

```
SELECT contract.contract_number, contract.contract_date, supplied.supplied_product,
supplied.supplied_cost, supplier.*
FROM (supplier INNER JOIN contract ON supplier.supplier_id = contract.supplier_id)
INNER JOIN supplied ON contract.contract_number = supplied.contract_number
WHERE contract.contract_date BETWEEN '2018-09-05' AND '2018-09-12' AND
supplier.supplier_id = 1;
```



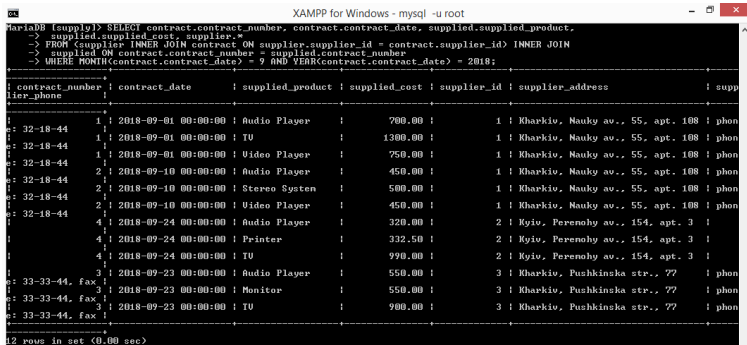
contract_number	contract_date	supplied_product	supplied_cost	supplier_id	supplier_address	supplier_phone
2	2018-09-10 00:00:00	Audio Player	450.00	1	Kharkiv, Nauky av., 55, apt. 108	phone: 32-18-44
2	2018-09-10 00:00:00	Stereo System	500.00	1	Kharkiv, Nauky av., 55, apt. 108	phone: 32-18-44
2	2018-09-10 00:00:00	Video Player	450.00	1	Kharkiv, Nauky av., 55, apt. 108	phone: 32-18-44

Рисунок 3.2 – Результат виконання запиту 2

### Запит 3

Сформувати список товарів, які були поставлені у 9 місяці 2018 року з виводом назви постачальника та дати поставки (рисунок 3.3).

```
SELECT contract.contract_number, contract.contract_date, supplied.supplied_product,  
       supplied.supplied_cost, supplier.*  
FROM (supplier INNER JOIN contract ON supplier.supplier_id = contract.supplier_id) INNER JOIN  
     supplied ON contract.contract_number = supplied.contract_number  
WHERE MONTH(contract.contract_date) = 9 AND YEAR(contract.contract_date) = 2018;
```



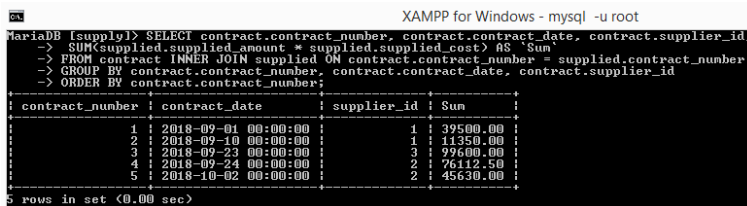
```
XAMPP for Windows - mysql -u root  
MariaDB [supply] > SELECT contract.contract_number, contract.contract_date, supplied.supplied_product,  
-> supplied.supplied_cost, supplier.*  
-> FROM (supplier INNER JOIN contract ON supplier.supplier_id = contract.supplier_id) INNER JOIN  
-> supplied ON contract.contract_number = supplied.contract_number  
-> WHERE MONTH(contract.contract_date) = 9 AND YEAR(contract.contract_date) = 2018;  
+-----+-----+-----+-----+-----+-----+-----+  
| contract_number | contract_date | supplied_product | supplied_cost | supplier_id | supplier_address | supplied_amount |  
+-----+-----+-----+-----+-----+-----+-----+  
| 32-18-44 | 2018-09-01 00:00:00 | Audio Player | 700.00 | 1 | Kharkiv, Nauky av., 55, apt. 108 | phon | 1 |  
| 32-18-44 | 2018-09-01 00:00:00 | TV | 1300.00 | 1 | Kharkiv, Nauky av., 55, apt. 108 | phon | 1 |  
| 32-18-44 | 2018-09-01 00:00:00 | Video Player | 750.00 | 1 | Kharkiv, Nauky av., 55, apt. 108 | phon | 1 |  
| 32-18-44 | 2018-09-10 00:00:00 | Audio Player | 450.00 | 1 | Kharkiv, Nauky av., 55, apt. 108 | phon | 2 |  
| 32-18-44 | 2018-09-10 00:00:00 | Stereo System | 500.00 | 1 | Kharkiv, Nauky av., 55, apt. 108 | phon | 1 |  
| 32-18-44 | 2018-09-10 00:00:00 | Video Player | 450.00 | 1 | Kharkiv, Nauky av., 55, apt. 108 | phon | 2 |  
| 32-18-44 | 2018-09-24 00:00:00 | Audio Player | 320.00 | 2 | Kyiv, Pevnohy av., 154, apt. 3 | | 1 |  
| 32-18-44 | 2018-09-24 00:00:00 | Printer | 332.50 | 2 | Kyiv, Pevnohy av., 154, apt. 3 | | 1 |  
| 32-18-44 | 2018-09-24 00:00:00 | TV | 990.00 | 2 | Kyiv, Pevnohy av., 154, apt. 3 | | 1 |  
| 32-18-44 | 2018-09-23 00:00:00 | Audio Player | 550.00 | 3 | Kharkiv, Puchkynska str., 77 | phon | 1 |  
| 33-33-44, Fax | 2018-09-23 00:00:00 | Monitor | 550.00 | 3 | Kharkiv, Puchkynska str., 77 | phon | 1 |  
| 33-33-44, Fax | 2018-09-23 00:00:00 | TV | 900.00 | 3 | Kharkiv, Puchkynska str., 77 | phon | 1 |  
+-----+-----+-----+-----+-----+-----+-----+  
12 rows in set (0.00 sec)
```

Рисунок 3.3 – Результат виконання запиту 3

### Запит 4

Сформувати список договорів (номер, дата, назва) та загальну суму за кожним договором (розмір партії помножити на ціну за одиницю та просумувати за договором). Список повинен бути відсортований за зростанням номерів договорів (рисунок 3.4).

```
SELECT contract.contract_number, contract.contract_date, contract.supplier_id,  
       SUM(supplied.supplied_amount * supplied.supplied_cost) AS `Sum`  
FROM contract INNER JOIN supplied ON contract.contract_number = supplied.contract_number  
GROUP BY contract.contract_number, contract.contract_date, contract.supplier_id  
ORDER BY contract.contract_number;
```



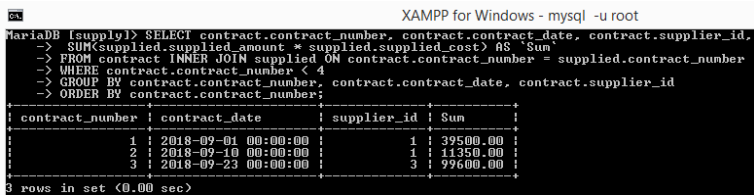
```
XAMPP for Windows - mysql -u root  
MariaDB [supply] > SELECT contract.contract_number, contract.contract_date, contract.supplier_id,  
-> SUM(supplied.supplied_amount * supplied.supplied_cost) AS `Sum`  
-> FROM contract INNER JOIN supplied ON contract.contract_number = supplied.contract_number  
-> GROUP BY contract.contract_number, contract.contract_date, contract.supplier_id  
-> ORDER BY contract.contract_number;  
+-----+-----+-----+-----+  
| contract_number | contract_date | supplier_id | Sum |  
+-----+-----+-----+-----+  
| 1 | 2018-09-01 00:00:00 | 1 | 39500.00 |  
| 2 | 2018-09-10 00:00:00 | 1 | 11350.00 |  
| 3 | 2018-09-23 00:00:00 | 3 | 99600.00 |  
| 4 | 2018-09-24 00:00:00 | 2 | 76112.50 |  
| 5 | 2018-10-02 00:00:00 | 2 | 45630.00 |  
+-----+-----+-----+-----+  
5 rows in set (0.00 sec)
```

Рисунок 3.4 – Результат виконання запиту 4

### Запит 5

Сформувати список договорів (номер, дата, назва) та загальну суму за кожним договором (розмір партії помножити на ціну за одиницю та просумувати за договором). Список повинен бути відсортований за зростанням загальних сум за кожним договором. Після цього на список повинна бути накладена умова фільтрації, яка полягає у виключенні з результату запиту записів, для яких номер договору менший 4 (рисунок 3.5).

```
SELECT contract.contract_number, contract.contract_date, contract.supplier_id,
       SUM(supplied.supplied_amount * supplied.supplied_cost) AS `Sum`
FROM contract INNER JOIN supplied ON contract.contract_number = supplied.contract_number
WHERE contract.contract_number < 4
GROUP BY contract.contract_number, contract.contract_date, contract.supplier_id
ORDER BY contract.contract_number;
```



```

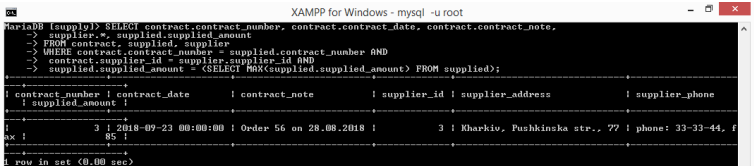
XAMPP for Windows - mysql -u root
MariaDB [supply]> SELECT contract.contract_number, contract.contract_date, contract.supplier_id,
-> SUM(supplied.supplied_amount * supplied.supplied_cost) AS `Sum`
-> FROM contract INNER JOIN supplied ON contract.contract_number = supplied.contract_number
-> WHERE contract.contract_number < 4
-> GROUP BY contract.contract_number, contract.contract_date, contract.supplier_id
-> ORDER BY contract.contract_number;
+-----+-----+-----+-----+
| contract_number | contract_date | supplier_id | Sum |
+-----+-----+-----+-----+
| 1 | 2018-09-01 00:00:00 | 1 | 39500.00 |
| 2 | 2018-09-10 00:00:00 | 1 | 11350.00 |
| 3 | 2018-09-23 00:00:00 | 3 | 99600.00 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Рисунок 3.5 – Результат виконання запиту 5

### Запит 6

Вивести інформацію про найбільшу за розміром партію товару в усіх договорах із зазначенням постачальника, а також номера та дати договору (рисунок 3.6).

```
SELECT contract.contract_number, contract.contract_date, contract.contract_note,
       supplier.*, supplied.supplied_amount
FROM contract, supplied, supplier
WHERE contract.contract_number = supplied.contract_number AND
       contract.supplier_id = supplier.supplier_id AND
       supplied.supplied_amount = (SELECT MAX(supplied.supplied_amount) FROM supplied);
```



```

XAMPP for Windows - mysql -u root
MariaDB [supply]> SELECT contract.contract_number, contract.contract_date, contract.contract_note,
-> supplier.*, supplied.supplied_amount
-> FROM contract, supplied, supplier
-> WHERE contract.contract_number = supplied.contract_number AND
-> contract.supplier_id = supplier.supplier_id AND
-> supplied.supplied_amount = (SELECT MAX(supplied.supplied_amount) FROM supplied);
+-----+-----+-----+-----+-----+
| contract_number | contract_date | contract_note | supplier_id | supplier_address | supplier_phone |
+-----+-----+-----+-----+-----+
| 3 | 2018-09-23 00:00:00 | Order 56 on 28.08.2018 | 3 | Kharkiv, Pushkinska str., 77 | phone: 33-33-44, f |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Рисунок 3.6 – Результат виконання запиту 6

### Запит 7

Сформувати список постачальників (назва та код), з якими не було укладено жодного договору (рисунок 3.7).

```
SELECT * FROM supplier
WHERE supplier_id NOT IN (SELECT supplier_id FROM supplier);
```

```
cmd XAMPP for Windows
MariaDB [supply]> SELECT * FROM supplier
-> WHERE supplier_id NOT IN (SELECT supplier_id FROM supplier);
Empty set (0.00 sec)
```

Рисунок 3.7 – Результат виконання запиту 7

### Запит 8

Сформувати список назв поставлених товарів із зазначенням середньої ціни поставки за одиницю (незалежно від постачальника) (рисунок 3.8).

```
SELECT supplied_product, AVG(supplied_cost) AS `Average cost`
FROM supplied
GROUP BY supplied_product;
```

```
cmd XAMPP for Windows - mysql -u root
MariaDB [supply]> SELECT supplied_product, AVG(supplied_cost) AS `Average cost`
-> FROM supplied
-> GROUP BY supplied_product;
+-----+-----+
| supplied_product | Average cost |
+-----+-----+
| Audio Player    | 520.000000  |
| Monitor         | 550.000000  |
| Printer         | 332.500000  |
| Stereo System   | 500.000000  |
| TV              | 1012.500000 |
| Video Player    | 683.333333  |
+-----+-----+
6 rows in set (0.00 sec)
```

Рисунок 3.8 – Результат виконання запиту 8

### Запит 9

Сформувати список товарів (назва, кількість та ціна, постачальник), для яких ціна за одиницю більше середньої (рисунок 3.9).

```
SELECT supplied_product, supplied_amount, supplied_cost, supplier.*
FROM (supplier INNER JOIN contract ON supplier.supplier_id = contract.supplier_id)
INNER JOIN supplied ON contract.contract_number = supplied.contract_number
WHERE supplied_cost > (SELECT AVG(supplied_cost) FROM supplied);
```

```

XAMPP for Windows - mysql -u root
MariaDB [supply]> SELECT supplied_product, supplied_amount, supplied_cost, supplier.*
-> FROM (supplier INNER JOIN contract ON supplier.supplier_id = contract.supplier_id)
-> INNER JOIN supplied ON contract.contract_number = supplied.contract_number
-> WHERE supplied_cost > (SELECT AVG(supplied_cost) FROM supplied);
+-----+-----+-----+-----+-----+-----+
| supplied_product | supplied_amount | supplied_cost | supplier_id | supplier_address | supplier_phone |
+-----+-----+-----+-----+-----+-----+
| Audio Player    | 25              | 700.00        | 1           | Kharkiv, Nauky av., 55, apt. 108 | phone: 32-18-44 |
| TV              | 10              | 1300.00       | 1           | Kharkiv, Nauky av., 55, apt. 108 | phone: 32-18-44 |
| Video Player   | 12              | 750.00        | 1           | Kharkiv, Nauky av., 55, apt. 108 | phone: 32-18-44 |
| TV              | 56              | 920.00        | 2           | Kyiv, Paremohy av., 154, apt. 3  | |
| TV              | 14              | 860.00        | 2           | Kyiv, Paremohy av., 154, apt. 3  | |
| Video Player    | 52              | 900.00        | 3           | Odesa, Derebasivska str., 75     | phone: 33-33-44, fax: |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

```

Рисунок 3.9 – Результат виконання запиту 9

### Запит 10

Вивести інформацію про п'ять найдорожчих товарів (назва, ціна за одиницю, постачальник) (рисунок 3.10).

```

SELECT supplied_product, supplied_cost, supplier.*
FROM (supplier INNER JOIN contract ON supplier.supplier_id = contract.supplier_id)
INNER JOIN supplied ON contract.contract_number = supplied.contract_number
ORDER BY supplied_cost DESC
LIMIT 1;

```

```

XAMPP for Windows - mysql -u root
MariaDB [supply]> SELECT supplied_product, supplied_cost, supplier.*
-> FROM (supplier INNER JOIN contract ON supplier.supplier_id = contract.supplier_id)
-> INNER JOIN supplied ON contract.contract_number = supplied.contract_number
-> ORDER BY supplied_cost DESC
-> LIMIT 1;
+-----+-----+-----+-----+-----+
| supplied_product | supplied_cost | supplier_id | supplier_address | supplier_phone |
+-----+-----+-----+-----+-----+
| TV              | 1300.00       | 1           | Kharkiv, Nauky av., 55, apt. 108 | phone: 32-18-44 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

Рисунок 3.10 – Результат виконання запиту 10

### Запит 11

Сформувати список постачальників із зазначенням коду, адреси та даних постачальника. При формуванні даних постачальника для фізичних осіб вивести прізвище та ініціали, а для юридичних осіб – назву (рисунок 3.11).

```

SELECT supplier.supplier_id, supplier.supplier_address,
IFNULL(supplier_org.supplier_org_name, CONCAT(RTRIM(supplier_person.supplier_last_name), ' ',
SUBSTRING(supplier_person.supplier_first_name, 1, 1), ' ',
SUBSTRING(supplier_person.supplier_middle_name, 1, 1), ' ')) AS `Supplier`
FROM (supplier LEFT JOIN supplier_person ON supplier.supplier_id = supplier_person.supplier_id)
LEFT JOIN supplier_org ON supplier.supplier_id = supplier_org.supplier_id;

```

```

XAMPP for Windows - mysql -u root
MariaDB [supply]> SELECT supplier.supplier_id, supplier.supplier_address,
-> IFNULL(supplier_org.supplier_org_name, CONCAT(RTRIM(supplier_person.supplier_last_name), ' ',
-> SUBSTRING(supplier_person.supplier_first_name, 1, 1), ' ',
-> SUBSTRING(supplier_person.supplier_middle_name, 1, 1), ' ')) AS `Supplier`
-> FROM (supplier LEFT JOIN supplier_person ON supplier.supplier_id = supplier_person.supplier_id)
-> LEFT JOIN supplier_org ON supplier.supplier_id = supplier_org.supplier_id;
+-----+-----+-----+
| supplier_id | supplier_address | Supplier |
+-----+-----+-----+
| 1           | Kharkiv, Nauky av., 55, apt. 108 | Petrov P. P. |
| 2           | Kyiv, Paremohy av., 154, apt. 3 | Interfruit Ltd. |
| 3           | Kharkiv, Pushkinska str., 77 | Ivanov I. I. |
| 4           | Odesa, Derebasivska str., 75 | Transservice LLC |
| 5           | Poltava, Soborna str., 15, apt. 43 | Sydorov S. S. |
+-----+-----+-----+
5 rows in set (0.00 sec)

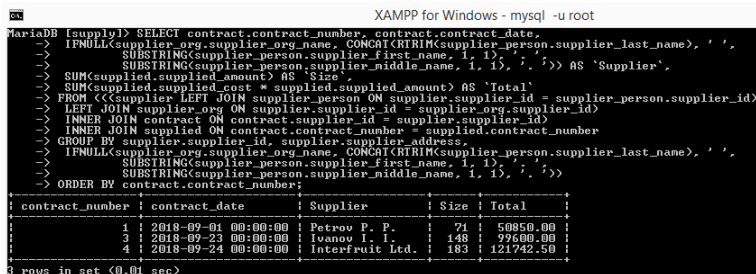
```

Рисунок 3.11 – Результат виконання запиту 11

## Запит 12

Сформувати список договорів (із зазначенням номеру, дати постачання та даних про постачальника), загальну кількість поставлених товарів та загальну суму за кожним договором. При формуванні даних постачальника для фізичних осіб вивести прізвище та ініціали, а для юридичних осіб – назву. В результат повинні бути включені тільки ті договори, на основі яких товари дійсно поставлялись (тобто в результат запити не повинні потрапити так звані «пусті» договори) (рисунок 3.12).

```
SELECT contract.contract_number, contract.contract_date,
IFNULL(supplier.org.supplier_org_name, CONCAT(RTRIM(supplier_person.supplier_last_name), ' ',
SUBSTRING(supplier_person.supplier_first_name, 1, 1), ' ',
SUBSTRING(supplier_person.supplier_middle_name, 1, 1), ' ') AS `Supplier`,
SUM(supplier.supplied_amount) AS `Size`,
SUM(supplier.supplied_cost * supplier.supplied_amount) AS `Total`
FROM (((supplier LEFT JOIN supplier_person ON supplier.supplier_id = supplier_person.supplier_id)
LEFT JOIN supplier_org ON supplier.supplier_id = supplier_org.supplier_id)
INNER JOIN contract ON contract.supplier_id = supplier.supplier_id)
INNER JOIN supplied ON contract.contract_number = supplied.contract_number
GROUP BY supplier.supplier_id, supplier.supplier_address,
IFNULL(supplier.org.supplier_org_name, CONCAT(RTRIM(supplier_person.supplier_last_name), ' ',
SUBSTRING(supplier_person.supplier_first_name, 1, 1), ' ',
SUBSTRING(supplier_person.supplier_middle_name, 1, 1), ' '))
ORDER BY contract.contract_number;
```



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

contract_number	contract_date	Supplier	Size	Total
1	2018-09-01 00:00:00	Petrov P. P.	74	50850.00
3	2018-09-23 00:00:00	Ivanov I. I.	148	99600.00
4	2018-09-24 00:00:00	Interfruit Ltd.	183	121742.50

Рисунок 3.12 – Результат виконання запити 12

## Запит 13

Сформувати список товарів (із зазначенням номеру договору та дати постачання), поставлених постачальниками 1 (Петров П. П.) та 2 («Інтерфрут») (рисунок 3.13).

```

SELECT supplied.contract_number, contract.contract_date,
       supplied.supplied_product, supplier.supplier_id
FROM supplied, contract, supplier
WHERE contract.contract_number = supplied.contract_number
      AND supplier.supplier_id = contract.supplier_id AND contract.supplier_id = 1
UNION
SELECT supplied.contract_number, contract.contract_date,
       supplied.supplied_product, supplier.supplier_id
FROM supplied, contract, supplier
WHERE contract.contract_number = supplied.contract_number
      AND supplier.supplier_id = contract.supplier_id AND contract.supplier_id = 2
ORDER BY supplier_id, contract_number;

```

```

C:\> XAMPP for Windows - mysql -u root
MariaDB [supply]> SELECT supplied.contract_number, contract.contract_date,
-> supplied.supplied_product, supplier.supplier_id
-> FROM supplied, contract, supplier
-> WHERE contract.contract_number = supplied.contract_number
-> AND supplier.supplier_id = contract.supplier_id AND contract.supplier_id = 1
-> UNION
-> SELECT supplied.contract_number, contract.contract_date,
-> supplied.supplied_product, supplier.supplier_id
-> FROM supplied, contract, supplier
-> WHERE contract.contract_number = supplied.contract_number
-> AND supplier.supplier_id = contract.supplier_id AND contract.supplier_id = 2
-> ORDER BY supplier_id, contract_number;
+-----+-----+-----+-----+
| contract_number | contract_date | supplied_product | supplier_id |
+-----+-----+-----+-----+
| 1 | 2018-09-01 00:00:00 | Audio Player | 1 |
| 1 | 2018-09-01 00:00:00 | TV | 1 |
| 1 | 2018-09-01 00:00:00 | Video Player | 1 |
| 2 | 2018-09-10 00:00:00 | Audio Player | 1 |
| 2 | 2018-09-10 00:00:00 | Stereo System | 1 |
| 2 | 2018-09-10 00:00:00 | Video Player | 1 |
| 4 | 2018-09-24 00:00:00 | Printer | 2 |
| 4 | 2018-09-24 00:00:00 | TV | 2 |
| 4 | 2018-09-24 00:00:00 | Audio Player | 2 |
| 5 | 2018-10-02 00:00:00 | Audio Player | 2 |
| 5 | 2018-10-02 00:00:00 | TV | 2 |
| 5 | 2018-10-02 00:00:00 | Video Player | 2 |
+-----+-----+-----+-----+
12 rows in set (0.00 sec)

```

Рисунок 3.13 – Результат виконання запиту 13

#### Запит 14

Сформувати номенклатуру товарів (тобто список назв товарів), які поставлялись тільки постачальником 1 (Петров П. П.), або тільки постачальником 2 («Інтерфрут»), або і постачальником 1, і постачальником 2 (рисунок 3.14).

```

SELECT DISTINCT supplied.supplied_product
FROM supplied, contract
WHERE contract.contract_number = supplied.contract_number AND contract.supplier_id = 1
UNION
SELECT DISTINCT supplied.supplied_product
FROM supplied, contract
WHERE contract.contract_number = supplied.contract_number AND contract.supplier_id = 2
ORDER BY supplied_product;

```



```

XAMPP for Windows - mysql -u root
MariaDB [supply]> SELECT DISTINCT supplied.supplied_product
-> FROM supplied, contract
-> WHERE contract.contract_number = supplied.contract_number AND contract.supplier_id = 1
-> UNION
-> SELECT DISTINCT supplied.supplied_product
-> FROM supplied, contract
-> WHERE contract.contract_number = supplied.contract_number AND contract.supplier_id = 2
-> ORDER BY supplied_product;
+-----+
| supplied_product |
+-----+
| Audio Player    |
| Printer         |
| Stereo System   |
| TU              |
| Video Player    |
+-----+
5 rows in set (0.00 sec)

```

Рисунок 3.14 – Результат виконання запиту 14

### Запит 15

Сформувати список товарів, який повинен відображати частоту поставок. У список включити тільки товари, які поставлялись більше одного разу. Список повинен бути відсортований за зменшенням частоти поставок (рисунок 3.15).

```

SELECT supplied_product, COUNT(supplied_product) AS `SupplyFrequency`
FROM supplied
GROUP BY supplied_product
HAVING COUNT(supplied_product) > 1
ORDER BY COUNT(supplied_product) DESC;

```

```

XAMPP for Windows - mysql -u root
MariaDB [supply]> SELECT supplied_product, COUNT(supplied_product) AS `SupplyFrequency`
-> FROM supplied
-> GROUP BY supplied_product
-> HAVING COUNT(supplied_product) > 1
-> ORDER BY COUNT(supplied_product) DESC;
+-----+-----+
| supplied_product | SupplyFrequency |
+-----+-----+
| Audio Player    | 5               |
| TU              | 4               |
| Video Player    | 3               |
+-----+-----+
3 rows in set (0.00 sec)

```

Рисунок 3.15 – Результат виконання запиту 15

### Запит 16

Сформувати дані про кількісну динаміку поставок товарів протягом 2018 року. Дані повинні бути агреговані за місяцями та представлені у вигляді таблиці, рядками якої є назви товарів, а стовпчиками – номери місяців 2018 року. На перетині рядка і стовпчика повинна відображатися кількість даного товару, поставленого у даному місяці (рисунок 3.16).

```

SELECT supplied_product, SUM(IF(MONTH(contract_date) = 1, supplied_amount, 0)) AS `Jan`,
SUM(IF(MONTH(contract_date) = 2, supplied_amount, 0)) AS `Feb`,
SUM(IF(MONTH(contract_date) = 3, supplied_amount, 0)) AS `Mar`,
SUM(IF(MONTH(contract_date) = 4, supplied_amount, 0)) AS `Apr`,
SUM(IF(MONTH(contract_date) = 5, supplied_amount, 0)) AS `May`,
SUM(IF(MONTH(contract_date) = 6, supplied_amount, 0)) AS `Jun`,
SUM(IF(MONTH(contract_date) = 7, supplied_amount, 0)) AS `Jul`,
SUM(IF(MONTH(contract_date) = 8, supplied_amount, 0)) AS `Aug`,
SUM(IF(MONTH(contract_date) = 9, supplied_amount, 0)) AS `Sep`,
SUM(IF(MONTH(contract_date) = 10, supplied_amount, 0)) AS `Oct`,
SUM(IF(MONTH(contract_date) = 11, supplied_amount, 0)) AS `Nov`,
SUM(IF(MONTH(contract_date) = 12, supplied_amount, 0)) AS `Dec`
FROM contract, supplied
WHERE contract.contract_number = supplied.contract_number AND YEAR(contract_date) = 2018
GROUP BY supplied_product
ORDER BY supplied_product;

```

```

mysql> SELECT supplied_product, SUM(IF(MONTH(contract_date) = 1, supplied_amount, 0)) AS `Jan`,
-> SUM(IF(MONTH(contract_date) = 2, supplied_amount, 0)) AS `Feb`,
-> SUM(IF(MONTH(contract_date) = 3, supplied_amount, 0)) AS `Mar`,
-> SUM(IF(MONTH(contract_date) = 4, supplied_amount, 0)) AS `Apr`,
-> SUM(IF(MONTH(contract_date) = 5, supplied_amount, 0)) AS `May`,
-> SUM(IF(MONTH(contract_date) = 6, supplied_amount, 0)) AS `Jun`,
-> SUM(IF(MONTH(contract_date) = 7, supplied_amount, 0)) AS `Jul`,
-> SUM(IF(MONTH(contract_date) = 8, supplied_amount, 0)) AS `Aug`,
-> SUM(IF(MONTH(contract_date) = 9, supplied_amount, 0)) AS `Sep`,
-> SUM(IF(MONTH(contract_date) = 10, supplied_amount, 0)) AS `Oct`,
-> SUM(IF(MONTH(contract_date) = 11, supplied_amount, 0)) AS `Nov`,
-> SUM(IF(MONTH(contract_date) = 12, supplied_amount, 0)) AS `Dec`
-> FROM contract, supplied
-> WHERE contract.contract_number = supplied.contract_number AND YEAR(contract_date) = 2018
-> GROUP BY supplied_product
-> ORDER BY supplied_product;

```

supplied_product	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Audio Player	0	0	0	0	0	0	0	0	63	33	0	0
Monitor	0	0	0	0	0	0	0	0	85	0	0	0
Printer	0	0	0	0	0	0	0	0	41	0	0	0
Stereo System	0	0	0	0	0	0	0	0	11	0	0	0
TU	0	0	0	0	0	0	0	0	118	14	0	0
Video Player	0	0	0	0	0	0	0	0	20	17	0	0

```

6 rows in set (0.00 sec)

```

Рисунок 3.16 – Результат виконання запиту 16

### Запит 17

Сформувати список поставлених товарів. Для кожного товару у цьому списку повинні бути вказані наступні дані: номер договору, назва товару, кількість одиниць, ціна за одиницю, дата поставки, назва місяця та номер року (рисунку 3.17).

```

SELECT supplied.contract_number, supplied.supplied_product,
supplied.supplied_amount, supplied.supplied_cost,
contract.contract_date,
MONTHNAME(contract.contract_date) AS `Month`,
YEAR(contract.contract_date) AS `Year`
FROM supplied, contract
WHERE contract.contract_number = supplied.contract_number;

```

```

XAMPP for Windows - mysql -u root
MariaDB [supply]> SELECT supplied_contract_number, supplied_contract_product,
-> supplied_contract_amount, supplied_contract_cost,
-> contract_contract_date,
-> MONTHNAME(contract_contract_date) AS 'Month',
-> YEAR(contract_contract_date) AS 'Year'
-> FROM supplied_contract
-> WHERE contract_contract_number = supplied_contract_number;

```

contract_number	supplied_contract_product	supplied_contract_amount	supplied_contract_cost	contract_date	Month	Year
1	Audio Player	25	700.00	2018-09-01 00:00:00	September	2018
1	TU	10	1300.00	2018-09-01 00:00:00	September	2018
1	Video Player	12	750.00	2018-09-01 00:00:00	September	2018
2	Audio Player	5	450.00	2018-09-10 00:00:00	September	2018
2	Screen System	11	500.00	2018-09-10 00:00:00	September	2018
2	Video Player	8	450.00	2018-09-10 00:00:00	September	2018
3	Audio Player	11	550.00	2018-09-23 00:00:00	September	2018
3	Monitor	85	550.00	2018-09-23 00:00:00	September	2018
3	TU	52	900.00	2018-09-23 00:00:00	September	2018
4	Audio Player	22	320.00	2018-09-24 00:00:00	September	2018
4	Printer	41	332.50	2018-09-24 00:00:00	September	2018
4	TU	56	990.00	2018-09-24 00:00:00	September	2018
5	Audio Player	33	580.00	2018-10-02 00:00:00	October	2018
5	TU	14	860.00	2018-10-02 00:00:00	October	2018
5	Video Player	17	850.00	2018-10-02 00:00:00	October	2018

```

15 rows in set (0.00 sec)

```

Рисунок 3.17 – Результат виконання запиту 17

## 2. Оформити звіт з лабораторної роботи

У звіт включити основні етапи виконання лабораторної роботи та знімки екрану, що їх демонструють.

## 3. Питання для самоконтролю

1. Який оператор мови SQL використовується для вибору даних з однієї або декількох таблиць?
2. Навести загальну структуру оператора SELECT.
3. Як можна записати SQL SELECT запит, якщо необхідно вивести усі стовпці таблиці?
4. Яка конструкція використовується для вибору записів, що задовольняють критеріям пошуку?
5. Яке ключове слово використовується для виключення повторень?
6. Яка конструкція використовується для сортування значень за одним або кількома стовпцями?
7. Яким чином реалізується зворотний порядок сортування?
8. За допомогою якого ключового слова виконується обмеження вибірки?
9. За допомогою якої конструкції виконується групування рядків, що вилучаються?
10. Назвати функції агрегації, їх призначення та основні особливості.
11. Яким чином стовпцю можна призначити нове ім'я?
12. Назвіть призначення та відмінність ключового слова HAVING від WHERE?
13. Назвати основні арифметичні, логічні та оператори порівняння, їх призначення та приклади використання.

14. Призначення функції MONTH та приклади її використання.
15. Призначення функції YEAR та приклади її використання.
16. Призначення функції IFNULL та приклади її використання.
17. Призначення функції CONCAT та приклади її використання.
18. Призначення функції RTRIM та приклади її використання.
19. Призначення функції SUBSTRING та приклади її використання.
20. Призначення функції IF та приклади її використання.
21. Який оператор використовується для об'єднання результатів двох запитів.

# ЛАБОРАТОРНА РОБОТА 4. СТВОРЕННЯ ТА ВИКОРИСТАННЯ ПРЕДСТАВЛЕНЬ

**Мета роботи:** навчитися створювати та застосовувати представлення (view) у базі даних на прикладі СУБД MySQL.

## Хід роботи

### 1. Створити представлення, що дозволяє при перегляді списку договорів бачити назву постачальника

Створення представлень здійснюється за допомогою оператора CREATE VIEW. Таким чином, створити представлення, яке дозволить переглядати список договорів із зазначенням назви постачальника, можна на основі наступного запиту (рисунок 4.1).

```
SELECT contract.contract_number, contract.contract_date, supplier.supplier_id,
supplier_org.supplier_org_name, supplier_person.supplier_last_name,
supplier_person.supplier_first_name, supplier_person.supplier_middle_name
FROM contract INNER JOIN supplier ON contract.supplier_id = supplier.supplier_id
LEFT OUTER JOIN supplier_org ON supplier.supplier_id = supplier_org.supplier_id
LEFT OUTER JOIN supplier_person ON supplier.supplier_id = supplier_person.supplier_id;
```

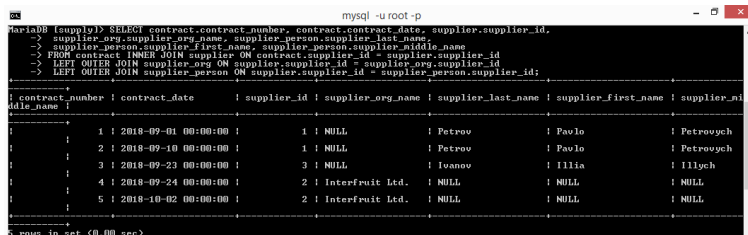


Рисунок 4.1 – Результат виконання представлення, що дозволяє при перегляді списку договорів бачити назву постачальника

Результат такого запиту має певний недолік – дані постачальників – юридичних та фізичних осіб знаходяться у різних стовпцях, а також присутні значення NULL. Цей недолік можна виправити за допомогою наступного запиту (рисунок 4.2).

```
SELECT contract.contract_number, contract.contract_date, supplier.supplier_id,
IFNULL(supplier_org.supplier_org_name, CONCAT(supplier_person.supplier_last_name, ' ',
supplier_person.supplier_first_name, ' ', supplier_person.supplier_middle_name)) AS `Supplier`
FROM contract INNER JOIN supplier ON contract.supplier_id = supplier.supplier_id
LEFT OUTER JOIN supplier_org ON supplier.supplier_id = supplier_org.supplier_id
LEFT OUTER JOIN supplier_person ON supplier.supplier_id = supplier_person.supplier_id;
```

```

mysql -u root -p
MariaDB [supply]> SELECT contract.contract_number, contract.contract_date, supplier.supplier_id,
-> IFNULL(supplier_org.supplier_org_name, CONCAT(supplier_person.supplier_last_name, ' ',
-> supplier_person.supplier_first_name, ' ', supplier_person.supplier_middle_name)) AS 'Supplier'
-> FROM contract INNER JOIN supplier ON contract.supplier_id = supplier.supplier_id
-> LEFT OUTER JOIN supplier_org ON supplier.supplier_id = supplier_org.supplier_id
-> LEFT OUTER JOIN supplier_person ON supplier.supplier_id = supplier_person.supplier_id;
+-----+-----+-----+-----+
| contract_number | contract_date | supplier_id | Supplier |
+-----+-----+-----+-----+
| 1 | 2018-09-01 00:00:00 | 1 | Petrov Pavlo Petrovych |
| 2 | 2018-09-10 00:00:00 | 1 | Petrov Pavlo Petrovych |
| 3 | 2018-09-23 00:00:00 | 3 | Ivanov Ilia Illych |
| 4 | 2018-09-24 00:00:00 | 2 | Interfruit Ltd. |
| 5 | 2018-10-02 00:00:00 | 2 | Interfruit Ltd. |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

Рисунок 4.2 – Результат роботи модифікованого запиту

Тепер можна створити дане представлення з назвою `contract_supplier` за допомогою відповідної команди мови SQL (рисунок 4.3).

```

mysql -u root -p
MariaDB [supply]> SHOW TABLES;
+-----+
| Tables_in_supply |
+-----+
| contract |
| contract_supplier |
| supplied |
| supplier |
| supplier_org |
| supplier_person |
+-----+
6 rows in set (0.00 sec)

MariaDB [supply]> SELECT * FROM contract_supplier;
+-----+-----+-----+-----+
| contract_number | contract_date | supplier_id | Supplier |
+-----+-----+-----+-----+
| 1 | 2018-09-01 00:00:00 | 1 | Petrov Pavlo Petrovych |
| 2 | 2018-09-10 00:00:00 | 1 | Petrov Pavlo Petrovych |
| 3 | 2018-09-23 00:00:00 | 3 | Ivanov Ilia Illych |
| 4 | 2018-09-24 00:00:00 | 2 | Interfruit Ltd. |
| 5 | 2018-10-02 00:00:00 | 2 | Interfruit Ltd. |
+-----+-----+-----+-----+
5 rows in set (0.01 sec)

```

Рисунок 4.3 – Результат виконання модифікованого представлення

## 2. Створити представлення, що дозволяє користувачу працювати з обмеженими даними про постачальників

Припустимо, що для певних користувачів повинна бути доступна не уся загальна інформація про постачальників (збережена у таблиці `supplier`), а лише інформація про код та адресу постачальника. При цьому користувач повинен мати можливість бачити дані постачальника як суб'єкта підприємницької діяльності (для юридичних осіб – назва, для фізичних – прізвище, ім'я, по батькові) (рисунок 4.4).

```

CREATE VIEW supplier_info AS
SELECT supplier.supplier_id, supplier.supplier_address,
IFNULL(supplier_org.supplier_org_name, CONCAT(supplier_person.supplier_last_name, ' ',
supplier_person.supplier_first_name, ' ', supplier_person.supplier_middle_name)) AS 'Info'
FROM supplier LEFT OUTER JOIN supplier_org ON supplier.supplier_id = supplier_org.supplier_id
LEFT OUTER JOIN supplier_person ON supplier.supplier_id = supplier_person.supplier_id;

```

```
mysql -u root -p
MariaDB [supply]> select * from supplier_info;
+-----+-----+-----+
| supplier_id | supplier_address | Info |
+-----+-----+-----+
| 1 | Kharkiv, Nauky av., 55, apt. 108 | Petrov Paolo Petrovych |
| 2 | Kyiv, Peremohy av., 154, apt. 3 | Interfruit Ltd. |
| 3 | Kharkiv, Pushkinska str., 77 | Ivanov Iliia Ilyych |
| 4 | Odesa, Derebasiuska str., 75 | Transservice LLC |
| 5 | Poltava, Soborna str., 15, apt. 43 | Sydarov Serhii Stepanovych |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Рисунок 4.4 – Результат роботи представлення, що дозволяє користувачу працювати з обмеженими даними про постачальників

В разі виникнення необхідності, видалити представлення можна за допомогою оператора DROP VIEW.

### 3. Оформити звіт з лабораторної роботи

У звіт включити основні етапи виконання лабораторної роботи та знімки екрану, що їх демонструють.

### 4. Питання для самоконтролю

1. Що таке представлення?
2. Назвати переваги та недоліки представлень.
3. Який оператор мови SQL використовується для створення представлень?
4. Який оператор мови SQL використовується для видалення представлень?
5. Яким чином можна перевірити наявність представлення у базі даних?
6. Як вказати список стовпців при створенні представлення?
7. Що таке вертикальне представлення?
8. Що таке горизонтальне представлення?

## СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. SQL Tutorial. URL: <https://www.w3schools.com/sql/default.asp> (дата звернення: 30.09.2020).
2. SQL. URL: <https://www.tutorialspoint.com/sql/index.htm> (дата звернення: 30.09.2020).
3. MySQL. URL: <https://www.tutorialspoint.com/mysql/index.htm> (дата звернення: 30.09.2020).
4. MySQL Documentation. URL: <https://dev.mysql.com/doc/> (дата звернення: 30.09.2020).
5. MySQL Tutorial – Learn MySQL Fast, Easy, and Fun. URL: <https://www.mysqltutorial.org/> (дата звернення: 30.09.2020).
6. MySQL Tutorial for Beginners Learn in 7 Days. URL: <https://www.guru99.com/mysql-tutorial.html> (дата звернення: 30.09.2020).
7. MySQL by Examples for Beginners. URL: [https://www3.ntu.edu.sg/home/ehchua/programming/sql/MySQL\\_Beginner.html](https://www3.ntu.edu.sg/home/ehchua/programming/sql/MySQL_Beginner.html) (дата звернення: 30.09.2020).
8. Drake M. An Introduction to Queries in MySQL. URL: <https://www.digitalocean.com/community/tutorials/introduction-to-queries-mysql> (дата звернення: 30.09.2020).
9. Мулеса О.Ю. Основи мови запитів SQL. Ужгород, 2015. 48 с.
10. Балик Н.Р., Мандзюк В.І. MySQL: лабораторний практикум. Тернопіль: Навчальна книга–Богдан, 2008. 88 с.
11. Балик Н.Р., Мандзюк В.І. Бази даних MySQL: Навчальний посібник. Тернопіль: Навчальна книга–Богдан, 2010. 160 с.
12. Берко А.Ю., Верес О.М., Пасічник В.В. Системи баз даних та знань. Книга 2. Системи управління базами даних та знань: навч. посібник. Львів : «Магнолія-2006», 2013. 584 с.
13. Ульман Л. MySQL: Руководство по изучению языка. Litres, 2019. 352 с.
14. Фиайли К. SQL: Руководство по изучению языка. Litres, 2019. 456 с.
15. Шварц Б., Зайцев П., Ткаченко В. MySQL по максимуму. 3-е издание. Издательский дом «Питер», 2018. 864 с.
16. Гольцман В.И. MySQL 5.0. Библиотека программиста. Издательский дом «Питер», 2009. 258 с.
17. Кузнецов М.В. MySQL 5. БХВ-Петербург, 2010. 1024 с.



Навчальне видання

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт  
за темою «Вивчення основ роботи з СУБД MySQL:  
Основні засоби DDL та DML мови SQL»  
для студентів спеціальностей  
121 «Інженерія програмного забезпечення»  
122 «Комп'ютерні науки»  
126 «Інформаційні системи та технології»

Укладачі:

ОРЛОВСЬКИЙ Дмитро Леонідович  
КОПП Андрій Михайлович

Відповідальний за випуск проф. Годлевський М.Д.  
Роботу до видання рекомендував проф. Гамаюн І.П.

План 2021 р., поз. 209

Підп. до друку 28.12.2021. Гарнітура Times New Roman.  
Ум. друк. арк. 0,4.

---

Видавничий центр НТУ «ХП»,  
вул. Кирпичова, 2, м. Харків, 61002  
Свідоцтво про державну реєстрацію ДК № 3478 від 21.08.2017 р.

---

Самостійне електронне видання