

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).

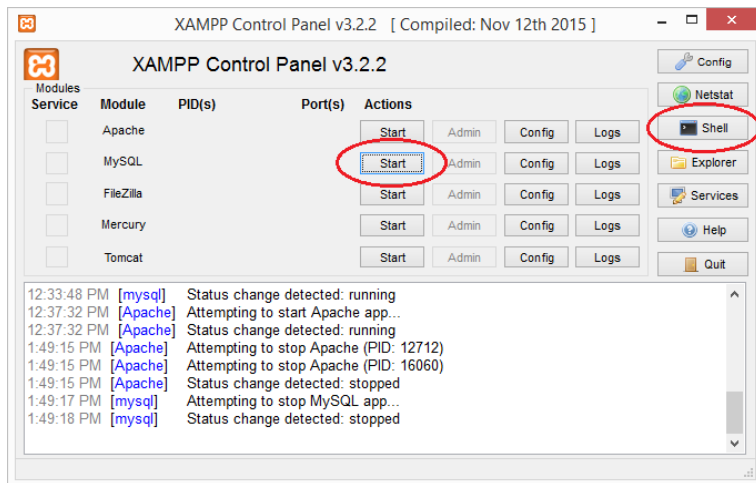
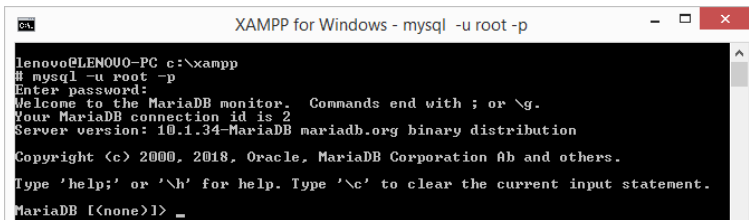


Рисунок 1.1 – Панель управління сервером XAMPP

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.



```
lenovo@LENOVO-PC c:\xampp
# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 2
Server version: 10.1.34-MariaDB mariadb.org binary distribution
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> _
```

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;
```

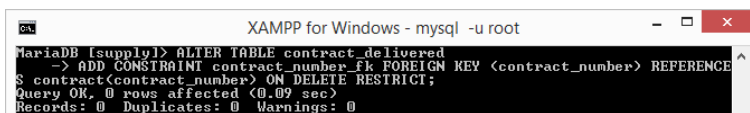
```
XAMPP for Windows - mysql -u root
MariaDB [supply]> CREATE TABLE contract_delivered (
  -> contract_number int NOT NULL,
  -> delivery_date timestamp NOT NULL,
  -> delivery_note varchar(100),
  -> PRIMARY KEY (contract_number)
  -> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0.04 sec)

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);
```

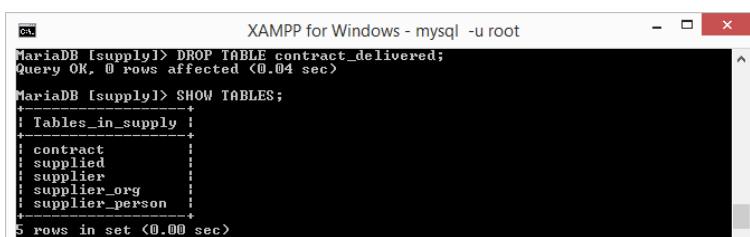


```
XAMPP for Windows - mysql -u root
MariaDB [supply]> ALTER TABLE contract_delivered
-> ADD CONSTRAINT contract_number_fk FOREIGN KEY (contract_number) REFERENCE
S contract(contract_number) ON DELETE RESTRICT;
Query OK, 0 rows affected (0.09 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

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?