

Chapter 17

# 네트워크 및 데이터베이스 프로그래밍

---

# Contents

---

01

데이터베이스 기초와 환경 설치

02

JDBC 프로그래밍

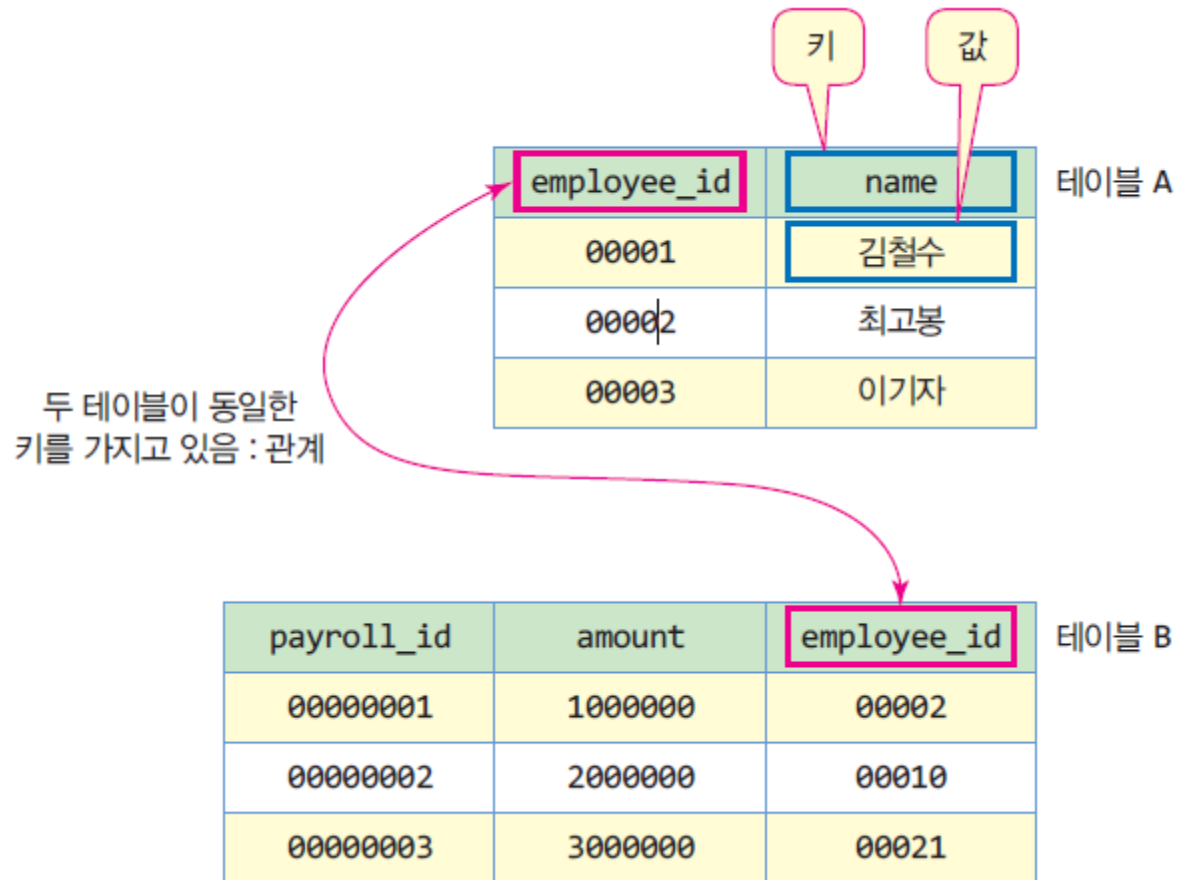
# 데이터베이스의 기초(1)

---

- 데이터베이스 관리 시스템(DBMS)은 크게 세 가지 종류인 계층형, 네트워크형, 관계형으로 구분
- 최근에는 관계형 DBMS가 주류
- 관계형 데이터베이스는 일련의 정형화된 테이블로 구현.
- 관계형 데이터베이스에서는 테이블을 재구성하지 않더라도 데이터를 다양한 방법으로 접근하거나 조합 가능
- 관계형 데이터베이스는 제작과 이용이 비교적 쉬울 뿐만 아니라 확장도 용이

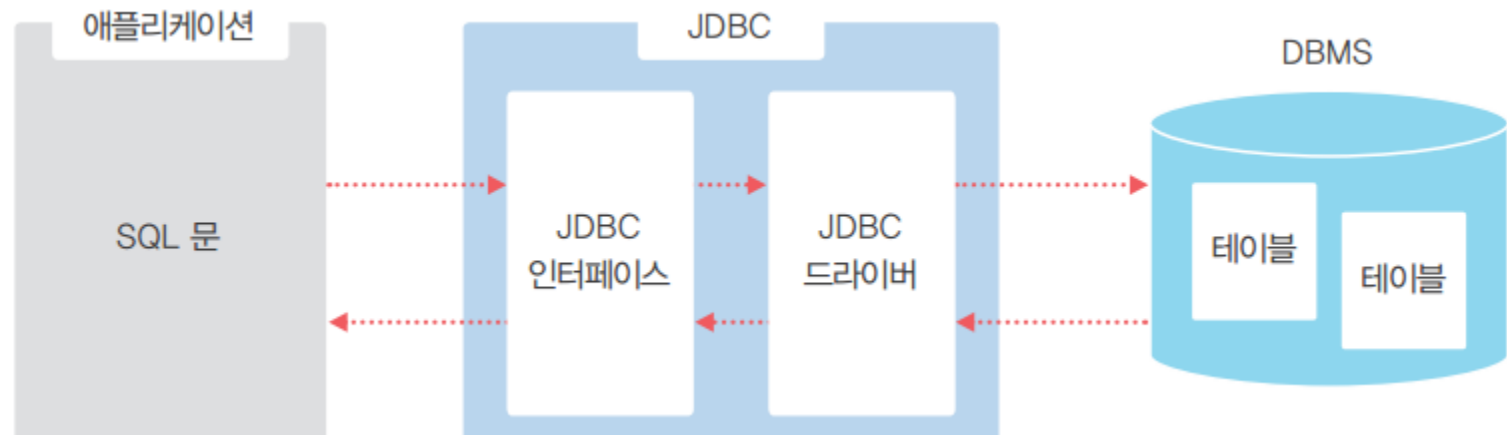
## 데이터베이스의 기초(2)

- 관계형 데이터베이스 테이블의 예



# JDBC 드라이버

- DBMS는 종류가 다양하며, 그 구조와 특징도 다 다르다. 따라서 자바 표준 API를 확장해 DBMS에 접근한다는 것은 자바의 기본 개념에 맞지 않을 뿐만 아니라 개발자에게도 고통
- 자바는 모든 DBMS에서 공통으로 사용할 수 있는 인터페이스와 클래스로 구성된 JDBC를 제공
- 각 DBMS 벤더는 JDBC를 구현한 JDBC드라이버를 개발자에게 제공
- JDBC가 데이터베이스에 접근하는 과정



# MySQL 다운로드 및 설치(1)

- MySQL 다운로드하기(<http://dev.mysql.com/downloads/mysql/>)

The screenshot shows the MySQL download page with the following elements:

- Browser Address Bar:** <https://dev.mysql.com/downloads/mysql/>
- Page Title:** MySQL Community
- Select Operating System:** Microsoft Windows
- Select OS Version:** All
- Recommended Download:** MySQL Installer 5.7.18
- Download Table:**

| Operating System   | Version | Size   | Action                   |
|--|---------|--------|--------------------------|
| Windows (x86, 32-bit), MSI Installer   | 5.7.18  | 18.5M  | <a href="#">Download</a> |
| (mysql-installer-web-community-5.7.18.1.msi) MD5: acc7a5ec61c2dafbe97501ffd8c8ed90   <a href="#">Signature</a> |         |        |                          |
| Windows (x86, 32-bit), MSI Installer   | 5.7.18  | 405.8M | <a href="#">Download</a> |
- File Name:** mysql-installer-community-5.7.18.1.msi
- MD5 Hash:** 82093d29c73beeb6 | [Signature](#)
- Verify Integrity:** [verify the integrity](#)
- Login or Sign Up:**
  - [Login »](#) using my Oracle Web account
  - [Sign Up »](#) for an Oracle Web account
- MySQL.com Notice:** MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.
- Footer:** [No thanks, just start my download.](#)

**3 클릭** (3 clicks) are indicated for the download process: 1. Clicking the 'Download' button for the 18.5M installer. 2. Clicking the 'Download' button for the 405.8M installer. 3. Clicking the 'No thanks, just start my download.' link.

# MySQL 다운로드 및 설치(2)

- MySQL 설치하기

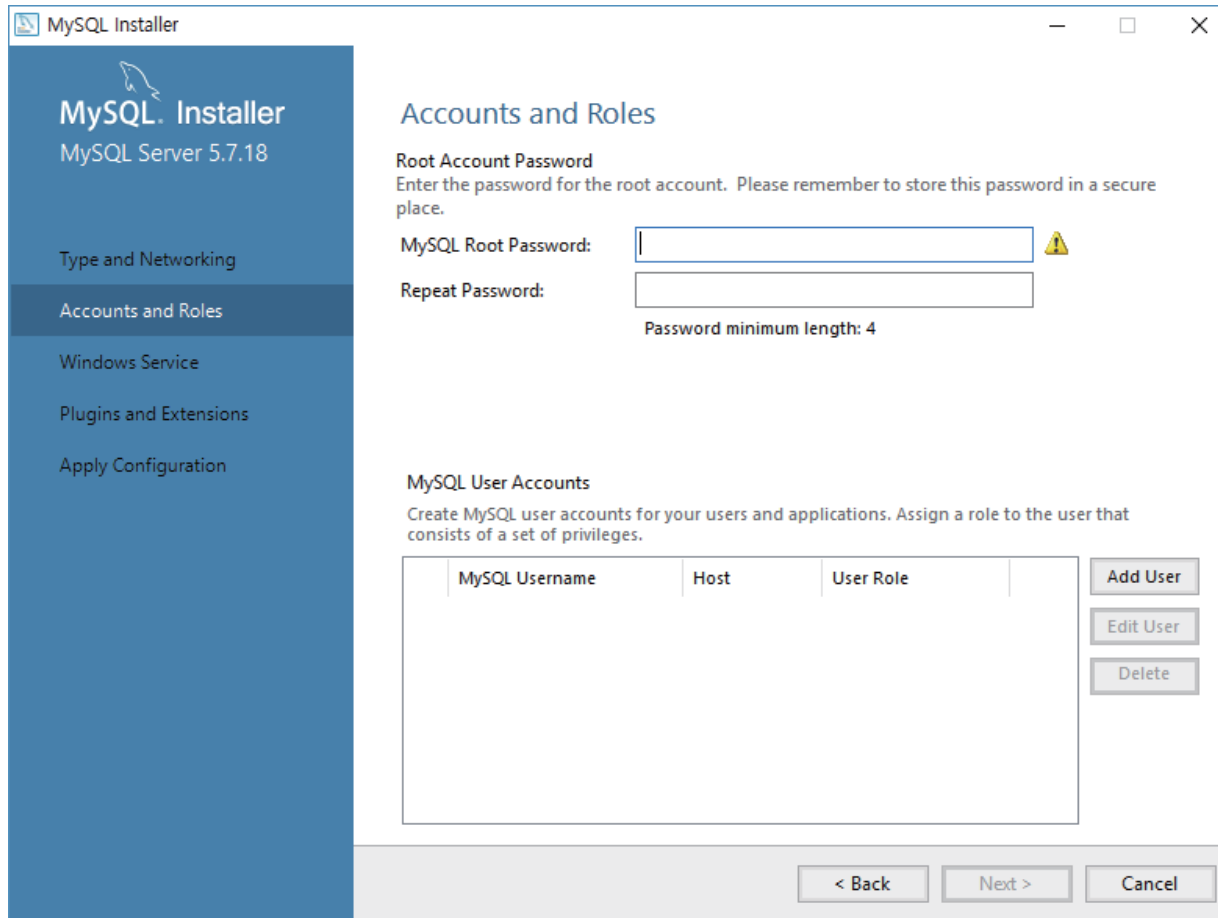
## Check Requirements

The following products have failing requirements. The installer will attempt to resolve some of this automatically. Requirements marked as manual cannot be resolved automatically. Click on those items to try and resolve them manually.

| For Product   | Requirement                               | Status |
|---|---|--------|
| <input type="radio"/> MySQL Server 5.7.18           | Microsoft Visual C++ 2013 Runtime...      |        |
| <input type="radio"/> MySQL Workbench 6.3.9         | Microsoft Visual C++ 2015 Runtime...      |        |
| <input type="radio"/> MySQL For Excel 1.3.6         | Visual Studio Tools for Office 2010 is... |        |
| <input type="radio"/> MySQL for Visual Studio 1.2.7 | Visual Studio version 2012, 2013, 20...   | Manual |
| <input type="radio"/> MySQL Utilities 1.6.5         | Microsoft Visual C++ 2013 Runtime...      |        |
| <input type="radio"/> MySQL Shell 1.0.9             | Microsoft Visual C++ 2013 Runtime...      |        |
| <input type="radio"/> MySQL Router 2.1.3            | Microsoft Visual C++ 2015 Runtime...      |        |
| <input type="radio"/> Connector/Python (3.4) 2.1.6  | Python 3.4 is not installed               | Manual |

# MySQL 다운로드 및 설치(3)

- 사용자 계정 설정하기



The screenshot shows the 'Accounts and Roles' step of the MySQL Installer for MySQL Server 5.7.18. The left sidebar contains navigation options: 'Type and Networking', 'Accounts and Roles' (selected), 'Windows Service', 'Plugins and Extensions', and 'Apply Configuration'. The main area is titled 'Accounts and Roles' and contains two sections. The first section, 'Root Account Password', instructs the user to enter a password for the root account, with a warning icon and a note that the password must be at least 4 characters long. It includes two input fields: 'MySQL Root Password:' and 'Repeat Password:'. The second section, 'MySQL User Accounts', instructs the user to create MySQL user accounts and assign roles. It features a table with columns for 'MySQL Username', 'Host', and 'User Role'. To the right of the table are three buttons: 'Add User', 'Edit User', and 'Delete'. At the bottom of the window are three buttons: '< Back', 'Next >', and 'Cancel'.

MySQL Installer  
MySQL Server 5.7.18

Type and Networking  
Accounts and Roles  
Windows Service  
Plugins and Extensions  
Apply Configuration

### Accounts and Roles

Root Account Password  
Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password:

Repeat Password:

Password minimum length: 4

### MySQL User Accounts

Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

| MySQL Username | Host | User Role |
|----------------|------|-----------|
|----------------|------|-----------|

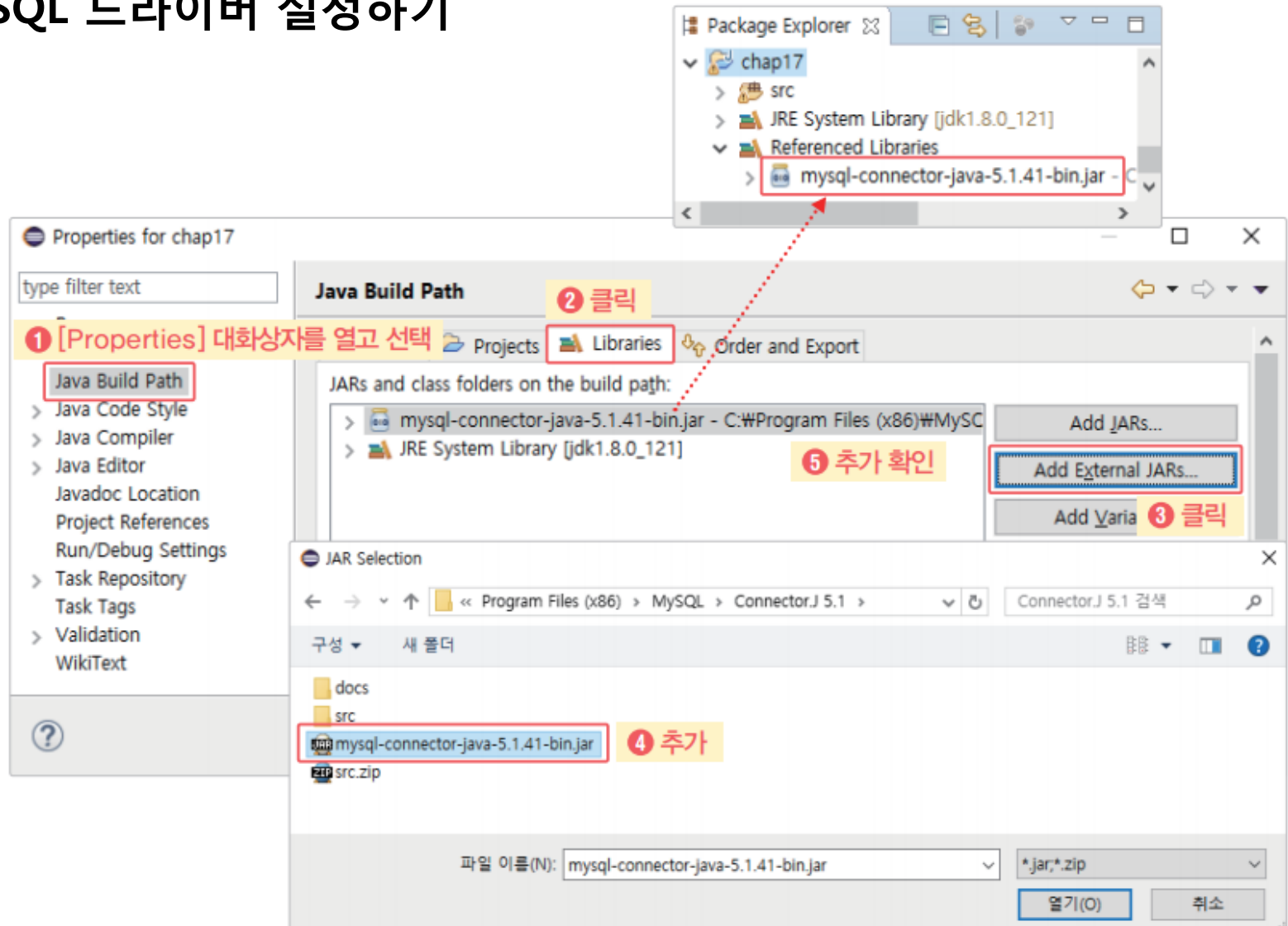
Add User  
Edit User  
Delete

< Back   Next >   Cancel



## MySQL 다운로드 및 설치(4)

- MySQL 드라이버 설정하기



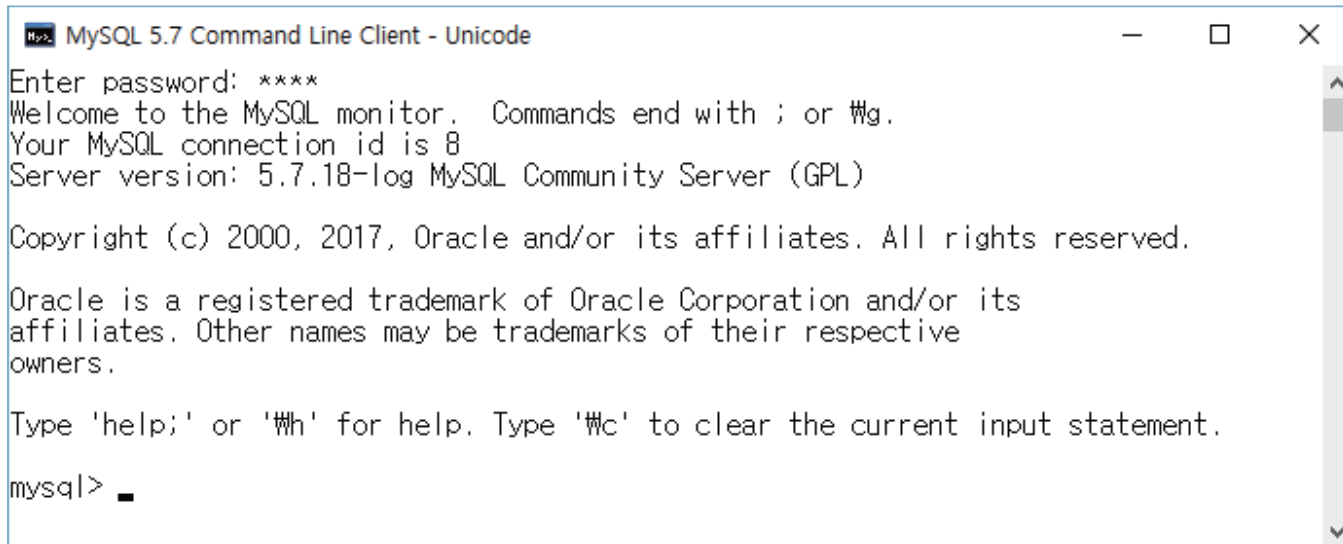
# SQL 명령어(1)

- SQL 명령어는 반드시 세미콜론( ; )으로 종료

| SQL 명령어 | 설명     |                   |
|---------|--------|-------------------|
| DDL     | CREATE | 테이블을 생성한다.        |
|         | DROP   | 테이블을 삭제한다.        |
|         | ALTER  | 테이블 구조를 변경한다.     |
|         | USE    | 사용할 데이터베이스를 지정한다. |
| DML     | INSERT | 데이터를 추가한다.        |
|         | UPDATE | 데이터를 갱신한다.        |
|         | DELETE | 데이터를 삭제한다.        |
|         | SELECT | 데이터를 조회한다.        |
| DCL     | GRANT  | 테이블에 권한을 부여한다.    |

## SQL 명령어(2)

- 대부분의 DBMS는 이런 SQL명령어를 직접 실행할 수 있는 명령 창을 제공



```
MySQL 5.7 Command Line Client - Unicode
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.7.18-log MySQL Community Server (GPL)

Copyright (c) 2000, 2017, 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> _
```

## SQL 명령어(3)

---

```
CREATE DATABASE 데이터베이스_이름;
```

```
CREATE TABLE 테이블_이름 (열_이름, 열_타입, ... )
```

```
CREATE DATABASE contacts; # contacts 데이터베이스를 생성한다.
```

```
USE contacts; # 사용할 데이터베이스로 변경한다.
```

```
CREATE TABLE person ( # contacts 데이터베이스에 person 테이블을 생성한다.
```

```
name VARCHAR(20), # person 테이블은 name, phone, email 열로 구성되며
```

```
phone VARCHAR(20),
```

```
email VARCHAR(20),
```

```
PRIMARY KEY(phone)); # phone을 기본키로 사용한다.
```

## SQL 명령어(4)

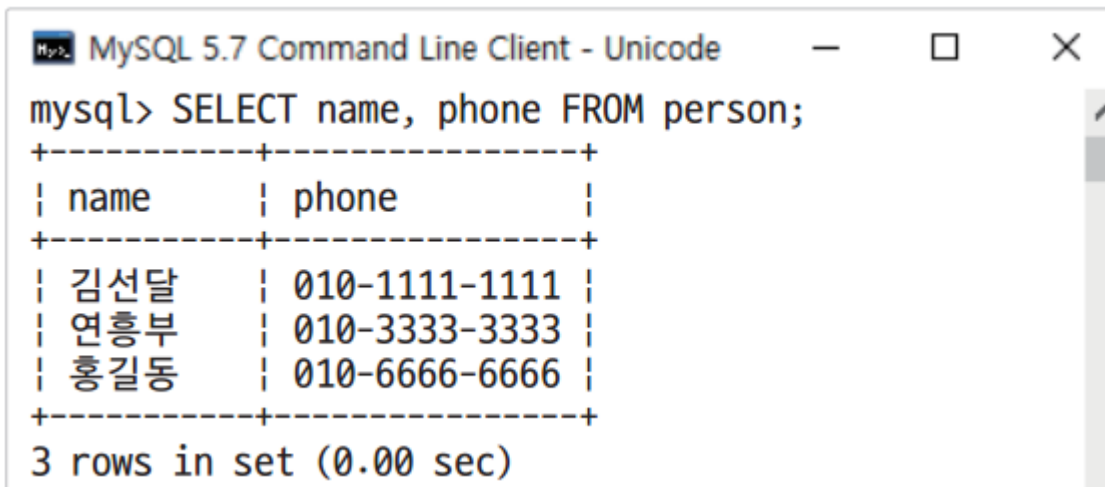
- SQL 명령어

```
INSERT INTO 테이블_이름 [(열_이름, ...)] VALUES(값, ...);
```

```
INSERT INTO person (name, phone, email) VALUES ('김선달', '010-1111-1111',  
'kim@one.com');
```

```
SELECT 열_이름, ... FROM 테이블_이름 [WHERE 조건] [ORDER BY 정렬 방식];
```

```
SELECT name, phone FROM person;
```



The screenshot shows a terminal window titled "MySQL 5.7 Command Line Client - Unicode". The prompt is "mysql>". The user has entered the command "SELECT name, phone FROM person;". The output is a table with two columns: "name" and "phone". There are three rows of data: "김선달" with phone "010-1111-1111", "연홍부" with phone "010-3333-3333", and "홍길동" with phone "010-6666-6666". Below the table, it says "3 rows in set (0.00 sec)".

```
mysql> SELECT name, phone FROM person;  
+-----+-----+  
| name      | phone      |  
+-----+-----+  
| 김선달     | 010-1111-1111 |  
| 연홍부     | 010-3333-3333 |  
| 홍길동     | 010-6666-6666 |  
+-----+-----+  
3 rows in set (0.00 sec)
```

## SQL 명령어(5)

DELETE FROM 테이블\_이름 [WHERE 조건];

DELETE FROM person WHERE name='김선달';

UPDATE 테이블\_이름 SET 열\_이름=값, ... [WHERE 조건];

UPDATE person SET phone='010-5555-5555' WHERE name='홍길동';

MySQL 5.7 Command Line Client - Unicode

```
mysql> SELECT name, phone FROM person;
```

| name | phone         |
|------|---------------|
| 연홍부  | 010-3333-3333 |
| 홍길동  | 010-5555-5555 |

2 rows in set (0.00 sec)

```
mysql>
```

## JDBC프로그래밍 과정

---

- ① JDBC드라이버 로드 및 데이터베이스를 연결한다.
- ② SQL문을 실행한다.
- ③ ResultSet객체를 처리한다.
- ④ 자원 객체를 닫는다.

# JDBC드라이버 로드 및 데이터베이스 연결

---

- MySQL 드라이버의 경우 다음 실행문으로 드라이버를 메모리에 로드

```
Class.forName("com.mysql.jdbc.Driver");
```

- 데이터베이스와 연결해 Connection 객체 생성

```
Connection con = DriverManager.getConnection(url, user_id, passwd);
```

- 여기서 url은 다음 형식 사용

```
jdbc:subprotocol:subname
```



- 예제 : [예제 17-5]

```
1 import java.sql.Connection;
2 import java.sql.DriverManager;
3 import java.sql.SQLException;
4
5 public class DB1Demo {
6     public static void makeConnection() {
7         String url = "jdbc:mysql://localhost/contacts?characterEncoding=UTF-8&serverTimezone=UTC";
8
9         Connection con = null;
10        try {
11            Class.forName("com.mysql.cj.jdbc.Driver");
12            System.out.println("데이터베이스 연결 중...");
13            con = DriverManager.getConnection(url, "root", "pass");
14            System.out.println("데이터베이스 연결 성공");
15        } catch (ClassNotFoundException ex) {
16            System.out.println(ex.getMessage());
17        } catch (SQLException ex) {
18            System.out.println("SQLException: " + ex.getMessage());
19        }
20    }
21
22    public static void main(String[] args) {
23        makeConnection();
24    }
25
26
27 }
```

|                |
|----------------|
| 데이터베이스 연결 중... |
| 데이터베이스 연결 성공   |

# SQL 문 실행

- SQL문을 DBMS에 전송할 수 있는 Statement 객체 생성

```
Statement stmt = con.createStatement();
```

- Statement 인터페이스가 제공하는 주요 메서드

| 메서드                                | 설명  |
|------------------------------------|---|
| boolean execute(String sql)        | SQL 문을 수행한다. 실행 결과가 ResultSet 객체라면 true, 아니면 false를 반환한다. |
| ResultSet executeQuery(String sql) | SQL 문을 수행한다. ResultSet 객체를 반환한다.                          |
| int executeUpdate(String sql)      | SQL 문을 수행한다. 영향을 받은 레코드의 개수를 반환한다.                        |

- 예제 : [예제 17-6]

```
1⊕ import java.sql.Connection;
5
6 public class DB2Demo {
7⊖     public static Connection makeConnection() {
8 String url = "jdbc:mysql://localhost/contacts?characterEncoding=UTF-8&serverTimezone=UTC";
9
10         Connection con = null;
11         try {
12             Class.forName("com.mysql.cj.jdbc.Driver");
13             System.out.println("데이터베이스 연결 중...");
14             con = DriverManager.getConnection(url, "root", "pass");
15             System.out.println("데이터베이스 연결 성공");
16         } catch (ClassNotFoundException ex) {
17             System.out.println(ex.getMessage());
18         } catch (SQLException ex) {
19             System.out.println("SQLException: " + ex.getMessage());
20         }
21
22         return con;
23     }
24 }
25
```

...

```

26 public static void main(String[] args) throws SQLException{
27     Connection con = makeConnection();
28     Statement stmt = con.createStatement();
29
30     String sql = "INSERT INTO person (name, phone, email) VALUES "
31         + "('임꺽정', '010-4444-4444', 'lim@four.com')";
32
33     if(stmt.executeUpdate(sql) == 1)
34         System.out.println("레코드 추가 성공");
35     else
36         System.out.println("레코드 추가 실패");
37
38     con.close();
39     stmt.close();
40 }
41
42 }

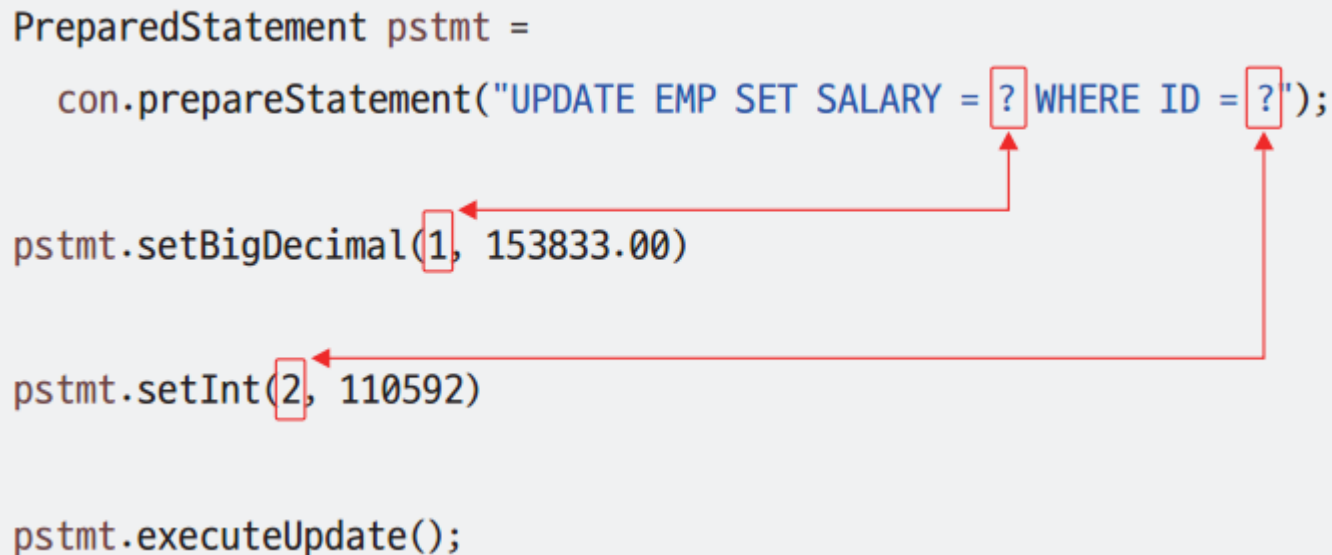
```

# SQL 문 실행

- **PreparedStatement 인터페이스**

- Statement의 자식 인터페이스
- SQL문을 미리 만들어 두고 사용할 수 있기 때문에 효율성이나 유지 보수 면에서 유리

```
PreparedStatement pstmt =  
    con.prepareStatement("UPDATE EMP SET SALARY = ? WHERE ID = ?");  
  
pstmt.setBigDecimal(1, 153833.00)  
  
pstmt.setInt(2, 110592)  
  
pstmt.executeUpdate();
```



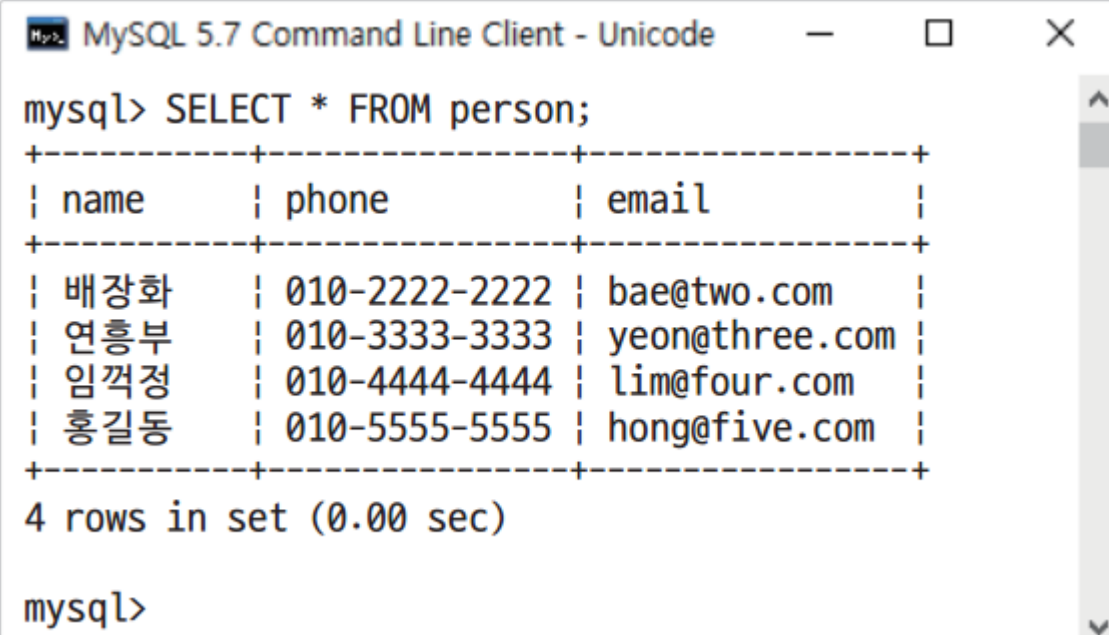
- 예제 : [예제 17-7]

```
1 import java.sql.Connection;
6
7 public class DB3Demo {
8     public static Connection makeConnection() {
9         String url = "jdbc:mysql://localhost/contacts?characterEncoding=UTF-8&serverTimezone=UTC";
10
11         Connection con = null;
12         try {
13             Class.forName("com.mysql.cj.jdbc.Driver");
14             System.out.println("데이터베이스 연결 중...");
15             con = DriverManager.getConnection(url, "root", "pass");
16             System.out.println("데이터베이스 연결 성공");
17         } catch (ClassNotFoundException ex) {
18             System.out.println(ex.getMessage());
19         } catch (SQLException ex) {
20             System.out.println("SQLException: " + ex.getMessage());
21         }
22
23         return con;
24     }
25 }
```

```

20
27 public static void main(String[] args) throws SQLException{
28     Connection con = makeConnection();
29     StringBuilder sql = new StringBuilder();
30     sql.append("INSERT INTO person (name, phone, email)");
31     sql.append("VALUES (?, ?, ?)");
32     PreparedStatement pstmt = con.prepareStatement(sql.toString());
33     pstmt.setString(1, "배장화");
34     pstmt.setString(2, "010-2222-2222");
35     pstmt.setString(3, "bae@two.com");
36
37     pstmt.execute();
38
39     con.close();
40     pstmt.close();
41 }
42
43 }

```



MySQL 5.7 Command Line Client - Unicode

```
mysql> SELECT * FROM person;
```

| name | phone         | email          |
|------|---------------|----------------|
| 배장화  | 010-2222-2222 | bae@two.com    |
| 연흥부  | 010-3333-3333 | yeon@three.com |
| 임걱정  | 010-4444-4444 | lim@four.com   |
| 홍길동  | 010-5555-5555 | hong@five.com  |

```

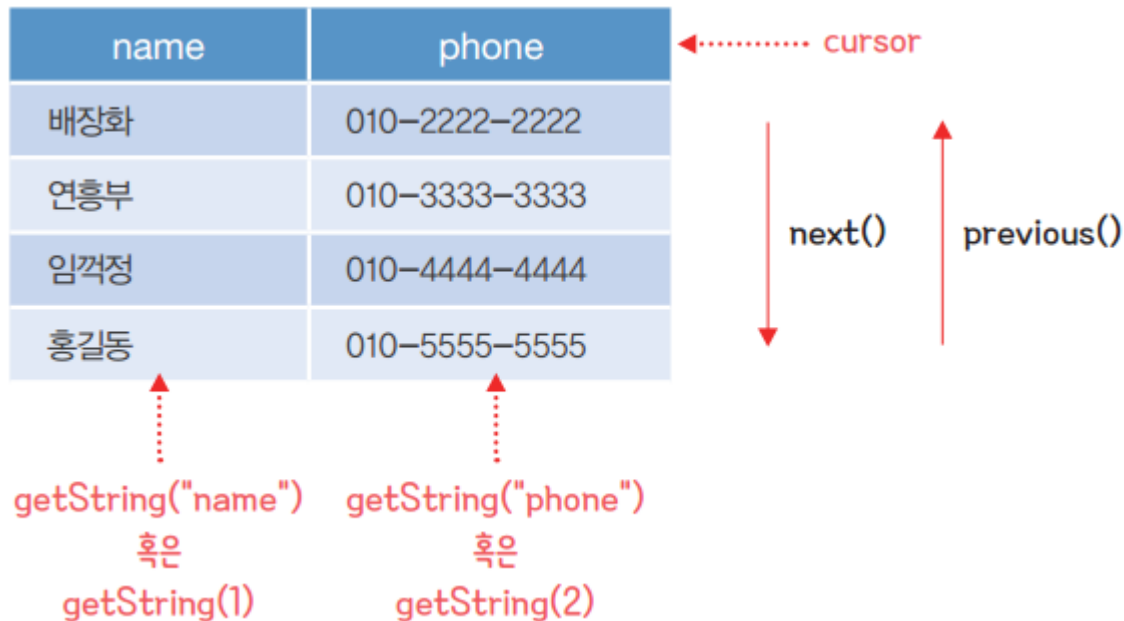
4 rows in set (0.00 sec)

mysql>

```

## 결과집합의 처리(1)

- ResultSet은 SQL질의문의 결과물을 추상화한 인터페이스
- 예를 들어 SELECT name, phone FROM person;의 결과집합





## 결과집합의 처리(2)

- ResultSet 인터페이스가 제공하는 주요 메서드

| 메서드   | 설명  |
|---|---|
| void afterLast( )   | 커서를 마지막 행 다음으로 이동한다.                      |
| void beforeFirst( )   | 커서를 첫 행 이전으로 이동한다.                        |
| void close( )   | ResultSet 객체를 닫는다.                        |
| boolean first( )  | 커서를 첫 행으로 이동한다.                           |
| int getInt(int columnIndex)<br>long getLong(String columnLabel)           | 현재 레코드에서 주어진 열의 값을 int 혹은 long 타입으로 반환한다. |
| int getRow( )   | 현재 레코드 번호를 반환한다.                          |
| String getString(int columnIndex)<br>String getString(String columnLabel) | 현재 레코드에서 주어진 열의 값을 String 타입으로 반환한다.      |
| boolean last( )   | 커서를 마지막 행으로 이동한다.                         |
| boolean next( )   | 커서를 다음 행으로 이동한다.                          |
| boolean previous( )   | 커서를 이전 행으로 이동한다.                          |

- 예제 : [예제 17-8]

```
1 import java.sql.Connection;
6
7 public class DB4Demo {
8     public static Connection makeConnection() {
9         String url = "jdbc:mysql://localhost/contacts?characterEncoding=UTF-8&serverTimezone=UTC";
10
11         Connection con = null;
12         try {
13             Class.forName("com.mysql.cj.jdbc.Driver");
14             System.out.println("데이터베이스 연결 중...");
15             con = DriverManager.getConnection(url, "root", "pass");
16             System.out.println("데이터베이스 연결 성공");
17         } catch (ClassNotFoundException ex) {
18             System.out.println(ex.getMessage());
19         } catch (SQLException ex) {
20             System.out.println("SQLException: " + ex.getMessage());
21         }
22
23         return con;
24     }
25 }
```

```

20
27 public static void main(String[] args) throws SQLException{
28     Connection con = makeConnection();
29
30     String sql = "SELECT * FROM person;";
31     PreparedStatement pstmt = con.prepareStatement(sql);
32     ResultSet rs = pstmt.executeQuery();
33
34     while(rs.next()) {
35         System.out.print("name : " + rs.getString(1) + "\t");
36         System.out.println("phone : " + rs.getString("phone"));
37     }
38     con.close();
39 }
40
41
42 }

```

데이터베이스 연결 중...

데이터베이스 연결 성공

|            |                       |
|------------|-----------------------|
| name : 김선달 | phone : 010-1111-1111 |
| name : 배장화 | phone : 010-2222-2222 |
| name : 연홍부 | phone : 010-3333-3333 |
| name : 임찍정 | phone : 010-4444-4444 |
| name : 홍길동 | phone : 010-6666-6666 |