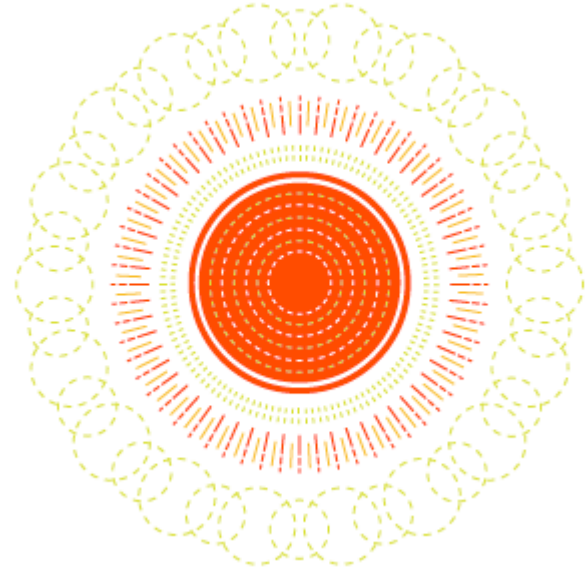


# JUST JAVA



객체지향에 충실한 자바다운 자바

## 데이터베이스 프로그래밍

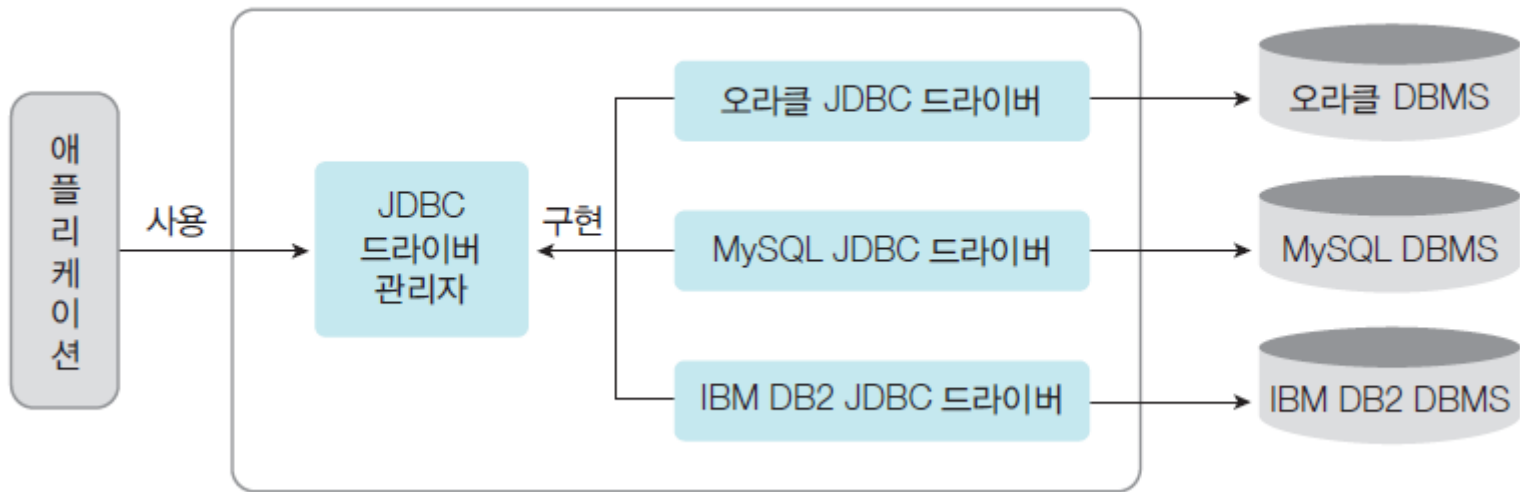
# 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

### ■ JDBC의 개념

- JDBC는 Java DataBase Connectivity의 약어로, 자바에서 데이터베이스 연동 프로그램을 개발하려고 만든 기술.

그림 10-22 JDBC의 동작 구조

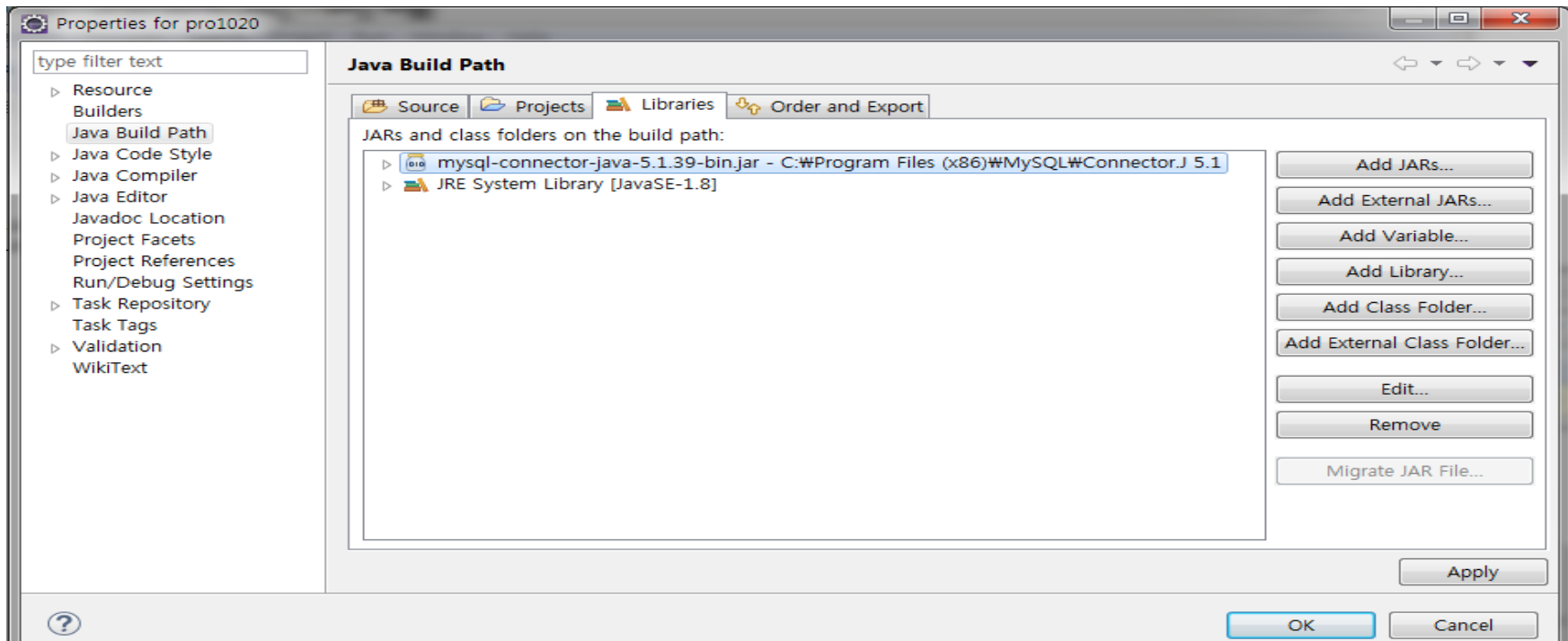


# 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

### ■ JDBC의 개념

- JDBC는 Java DataBase Connectivity의 약어로, 자바에서 데이터베이스 연동 프로그램을 개발하려고 만든 기술.
- JDBC는 데이터베이스 연결 및 쿼리에서 표준화된 인터페이스를 정의한다. 데이터베이스 회사에서는 자신들의 데이터베이스에 맞는 JDBC 드라이버(Driver)를 개발하여 배포하기 때문에 개발자들은 데이터베이스 회사와 상관 없이 표준화된 API를 이용하여 프로그램을 개발할 수 있다.

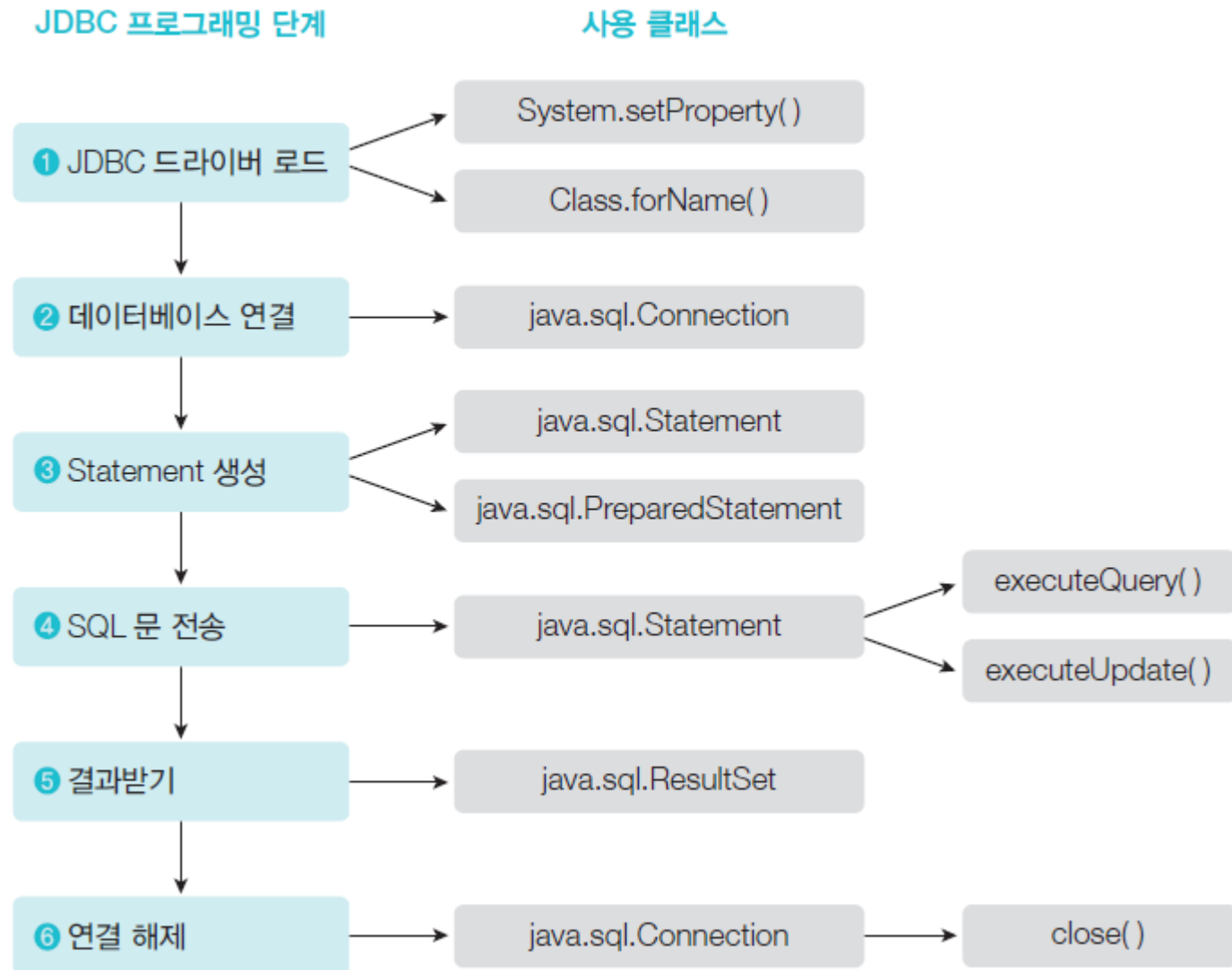


### 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

#### ■ JDBC의 개념

그림 10-24 JDBC 프로그래밍의 단계(JSP)



# 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

### ■ JDBC의 설치 및 연결 설정 : 클래스 패스

- 자바 가상머신은 프로그램을 실행할 때 이런 클래스 라이브러리의 경로 정보를 바탕으로 해당 라이브러리를 참조하는데, 이것을 클래스 패스Class Path라고 한다.

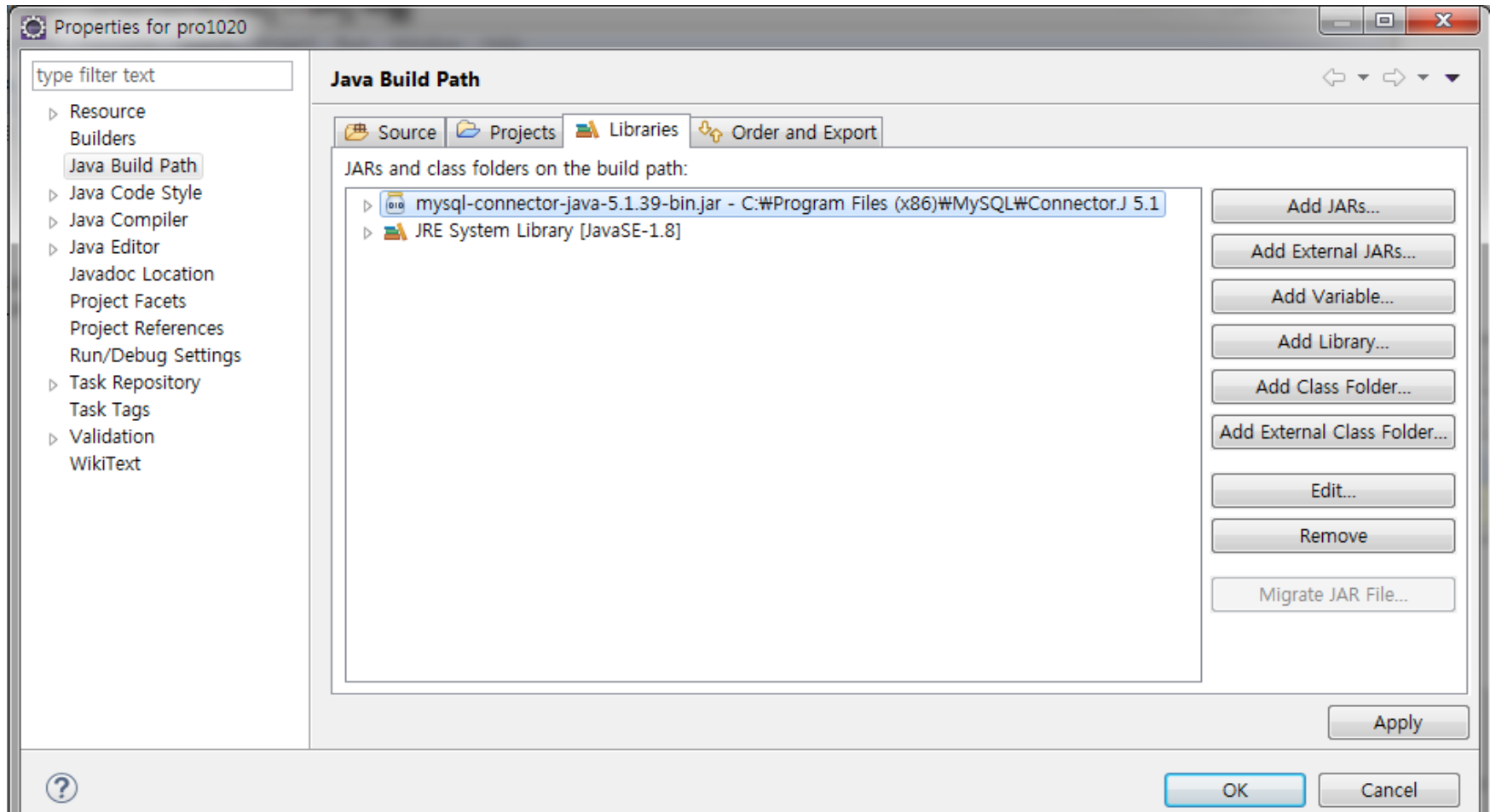
#### [클래스 패스의 관리법]

1. [java 설치 디렉터리\jre8\lib\ext]에 라이브러리를 복사하는 방법
2. 운영체제의 환경 변수에 CLASSPATH를 설정하는 방법
3. 이클립스 프로젝트 속성에 빌드 경로를 추가하는 방법

# 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

### ■ JDBC의 설치 및 연결 설정 : MySQL JDBC 드라이버 설치

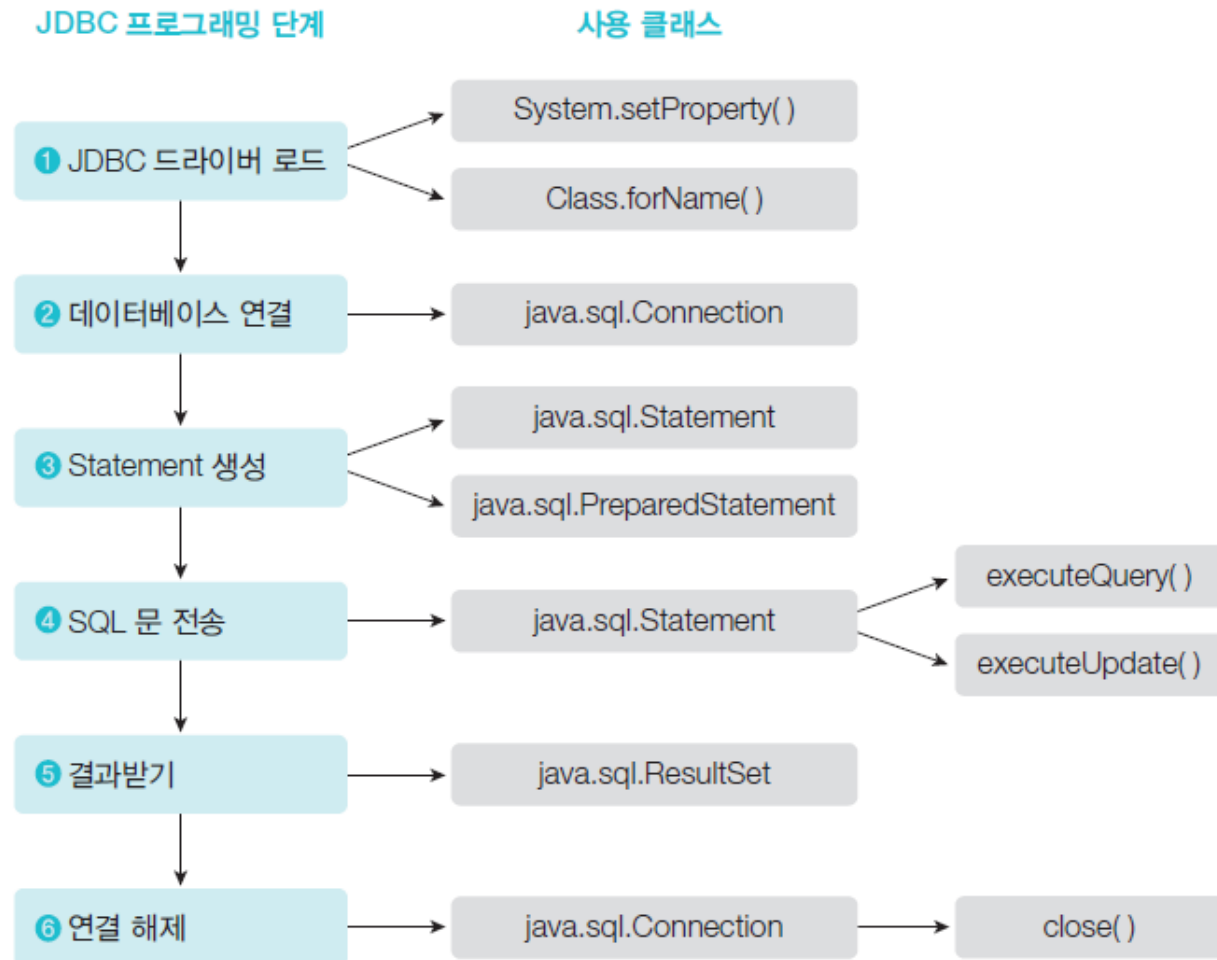


# 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

### ■ JDBC 프로그램 개발 절차

그림 10-24 JDBC 프로그래밍의 단계(JSP)



# 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

### ■ JDBC 프로그램 개발 절차

#### 1단계 : JDBC 드라이버 로드

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

#### 2단계 : 데이터베이스 연결

##### ▶ JDBC URL

```
jdbc:<서브 프로토콜>:<데이터 원본 식별자>
```

```
jdbc:mysql://DB 서버의 IP 주소/스키마:PORT(옵션임)
```

①

②

③

① **IP 주소** : MySQL 데이터베이스가 설치된 컴퓨터의 IP 주소 또는 도메인 이름이다.

② **스키마** : 데이터베이스에서 생성한 스키마(데이터베이스) 이름이다.

③ **PORT** : 기본 설정값인 3306 포트를 사용할 때는 생략할 수 있다.



# 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

### ■ JDBC 프로그램 개발 절차

#### 2단계 : 데이터베이스 연결

##### ▶ Connection 클래스 인스턴스 레퍼런스 얻기

```
Connection conn = DriverManager.getConnection(JDBC_URL, "아이디", "비밀번호");
```

①

②

① JDBC\_URL : 해당 데이터베이스에 맞게 미리 정의한 문자열이다.

② 아이디와 비밀번호 : 시스템에 로그인하는 계정이 아닌 데이터베이스 자체에서 관리하는 계정

# 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

### ■ JDBC 프로그램 개발 절차

#### 3단계 : Statement 생성

##### ▶ executeQuery( )

- SELECT 문을 수행할 때 사용

```
String sql = "select * from test";  
Statement stmt = conn.createStatement();  
stmt.executeQuery(sql);
```

##### ▶ executeUpdate( )

- UPDATE 문, DELETE 문 등을 수행할 때 사용한다.

```
Statement stmt = conn.createStatement();  
stmt.executeUpdate("insert into test values(' "+getUsername()+" ', ' "+getEmail()+" ');
```

# 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

### ■ JDBC 프로그램 개발 절차

#### 3단계 : Statement 생성

##### ▶ PreparedStatement

- PreparedStatement는 SQL 문을 미리 만들어 두고 변수를 따로 입력하는 방식이므로, 효율성이 나 유지보수 측면에서 유리하다.

```
PreparedStatement pstmt = conn.prepareStatement("insert into test values(?, ?)");  
pstmt.setString(1, getUsername());  
pstmt.setString(2, getEmail());  
pstmt.executeUpdate();
```

##### ▶ Statement의 close( )

```
stmt.close();  
pstmt.close();
```

# 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

### ■ JDBC 프로그램 개발 절차

#### 4단계 : SQL 문 전송

- 데이터를 입력·수정·삭제하려고 SQL 문을 만들 때는 PreparedStatement를 사용하여 변수와 적절히 조합.
- 데이터의 입력, 수정, 삭제는 Statement나 PreparedStatement의 executeUpdate( ) 메서드를 사용

```
int count = pstmt.executeUpdate();
```

```
pstmt.executeUpdate();
```

# 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

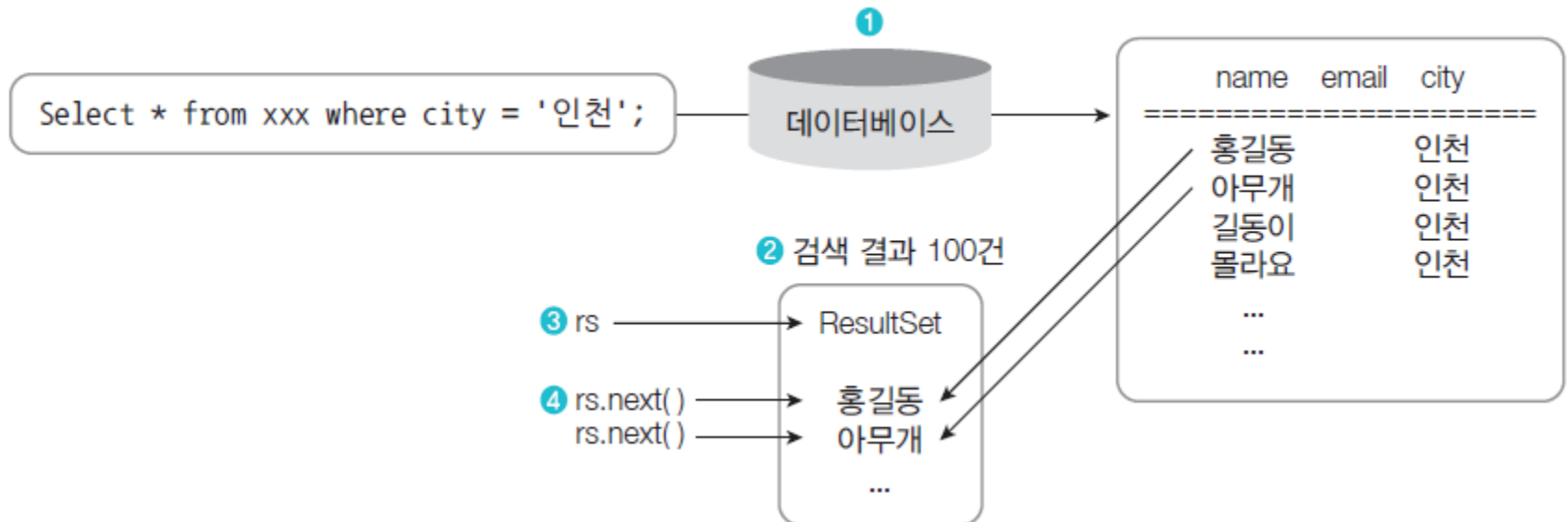
### ■ JDBC 프로그램 개발 절차

#### 5단계 : 결과받기

- Statement나 PreparedStatement의 executeQuery( )를 사용한다.
- 입력, 수정, 삭제와 달리 데이터를 가져올 때는 가져온 결과 데이터를 처리하는 ResultSet 객체가 필요하다.

```
ResultSet rs = pstmt.executeQuery();
```

그림 10-25 SQL 처리 결과와 ResultSet의 관계



# 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

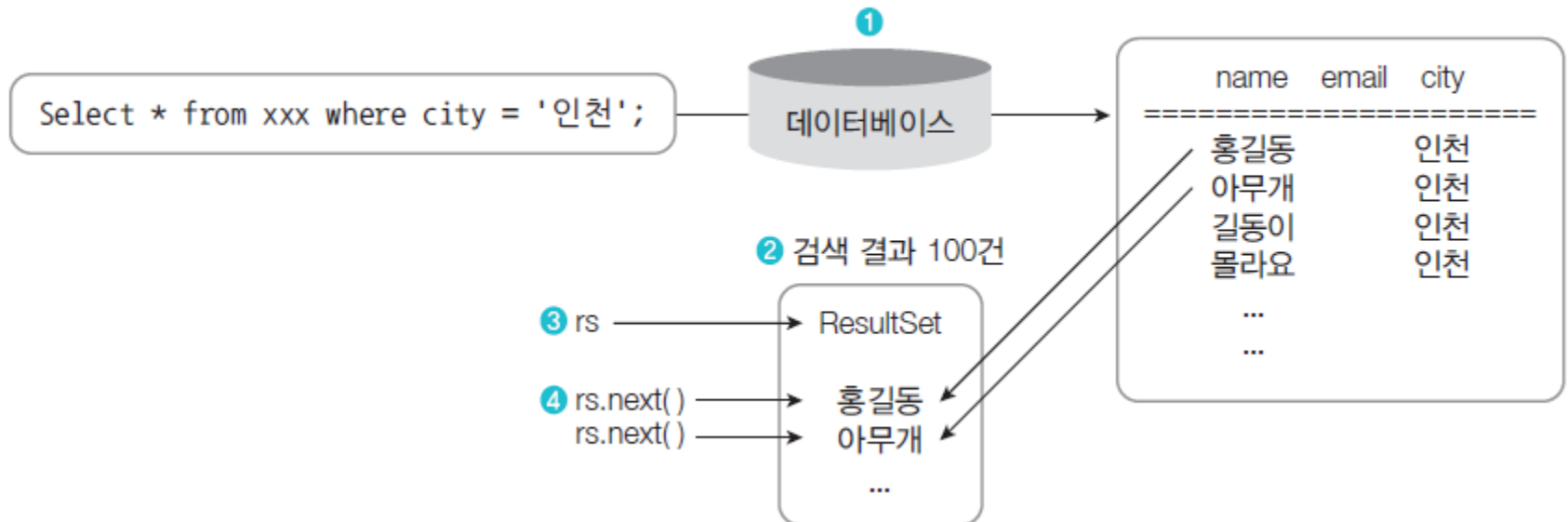
### ■ JDBC 프로그램 개발 절차

#### 5단계 : 결과받기

- Statement나 PreparedStatement의 executeQuery( )를 사용한다.
- 입력, 수정, 삭제와 달리 데이터를 가져올 때는 가져온 결과 데이터를 처리하는 ResultSet 객체가 필요하다.

```
ResultSet rs = pstmt.executeQuery();
```

그림 10-25 SQL 처리 결과와 ResultSet의 관계



# 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

### ■ JDBC 프로그램 개발 절차

#### 5단계 : 결과받기

- 로우에서 각 컬럼값을 가져오려면 rs.getXxx( ) 메서드를 사용

```
ResultSet rs = pstmt.executeQuery();
while(rs.next()) {
    name = rs.getString(1);    // or rs.getString("name");
    age = rs.getInt(2);        // or rs.getInt("email");
}
rs.close();
```

### 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

### ■ JDBC 프로그램 개발 절차

#### 6단계 : 연결 해제

데이터베이스 연결은 해당 연결을 이용한 작업이 모두 끝나는 시점에서 close( ) 메서드를 사용하여 해제해야 한다.

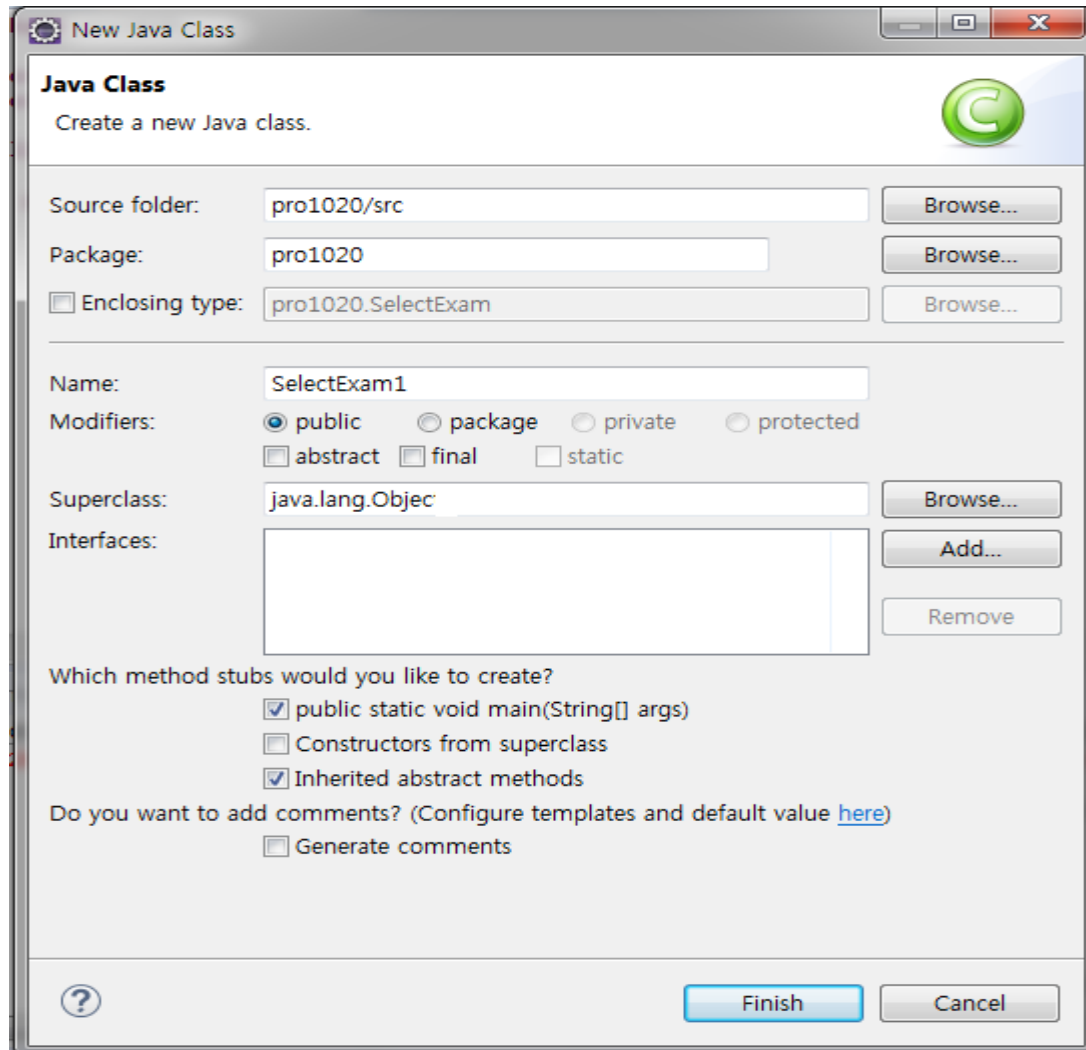
```
conn.close();
```



# 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

- JDBC를 활용한 database 연결하기
- ✓ Class 생성



# 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

- JDBC를 활용한 database 연결하기
- ✓ Class 코드 작성

```
3 import java.sql.*;
4 public class student {
5
6     public static void main(String[] args) {
7         String jdbcDriver = "com.mysql.cj.jdbc.Driver";
8         String jdbcUrl = "jdbc:mysql://localhost/yangjung?serverTimezone=UTC";
9         Connection conn = null;
10        PreparedStatement pstmt = null;
11        ResultSet rs = null;
12
13        String sql = "Select * from student";
14        try {
15            Class.forName(jdbcDriver);
16            conn = DriverManager.getConnection(jdbcUrl,"root","1234");
17            pstmt = conn.prepareStatement(sql);
18
19            rs = pstmt.executeQuery();
20
```

### 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

- JDBC를 활용한 database 연결하기
- ✓ Class 코드 작성

```

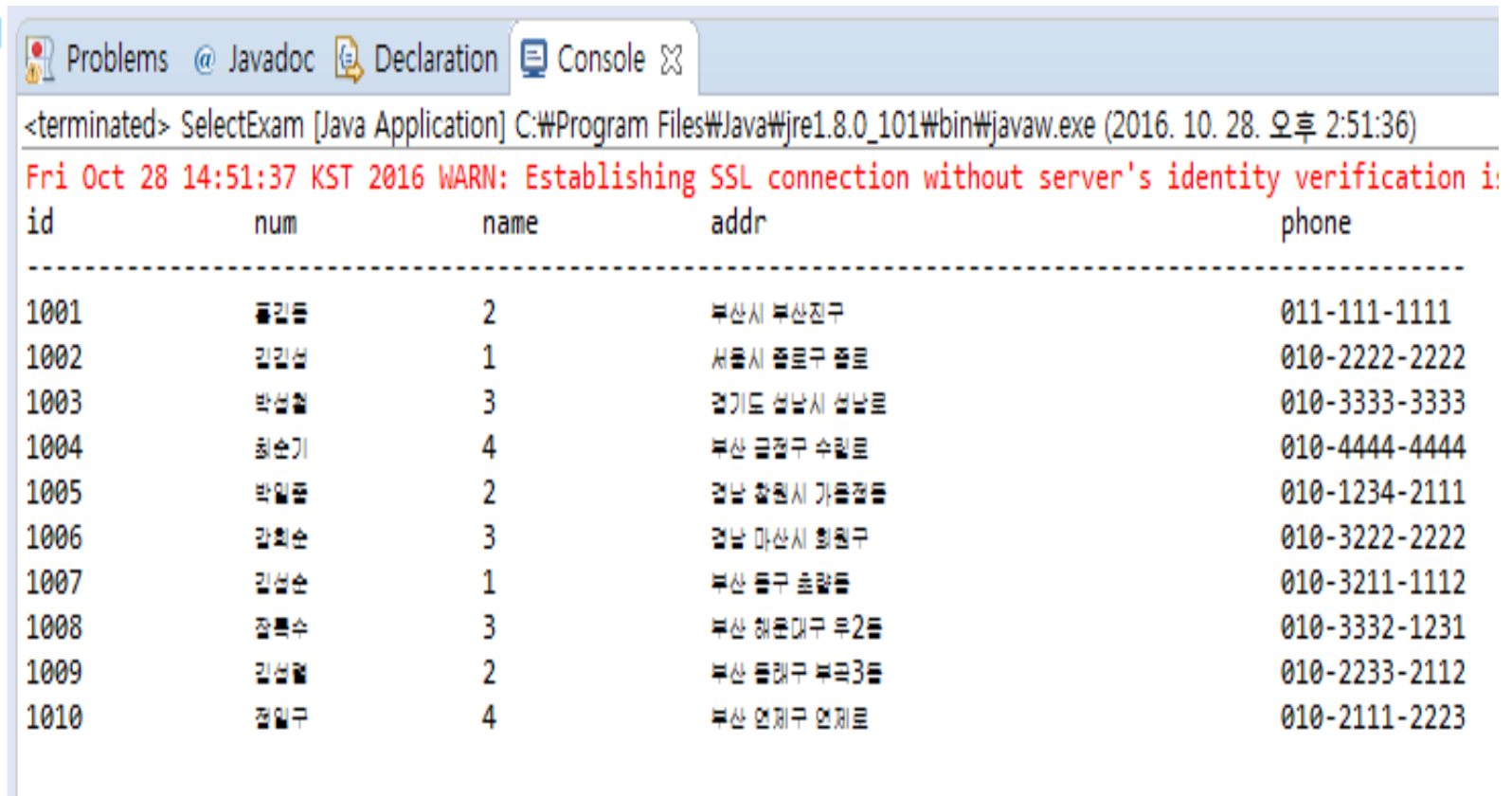
26 System.out.println("id\t\tnum\t\ttname\t\ttaddr\t\t\t\t\ttphone");
27 System.out.println("-----");
28     + "-----");
29 while(rs.next( )) {
30     System.out.print(rs.getLong("hakbun") + "\t\t");
31     System.out.print(rs.getString("name") + "\t\t");
32     System.out.print(rs.getString("hakyen") + "\t\t");
33     System.out.print(rs.getString("address") + "\t\t\t\t");
34     System.out.print(rs.getString("hp") + "\n");
35 }
36 }
37 catch(Exception e) {
38     System.out.println("데이터베이스 연결 실패!");
39     e.printStackTrace( );
40 }
41
42 finally {
43     try {
44         if(rs != null) rs.close( );
45         if(pstmt != null) pstmt.close( );
46         if(conn != null) conn.close( );
47     }
48     catch(Exception e) {
49         System.out.println(e.getMessage( ));
50     }
51 }
52 }
53
54 }
55

```

# 3. 서버 프로그램 구현하기(LM2001020203\_14v2.3)

## 02. JDBC의 이해

- JDBC를 활용한 database 연결하기
- ✓ 실행결과



```
<terminated> SelectExam [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (2016. 10. 28. 오후 2:51:36)
Fri Oct 28 14:51:37 KST 2016 WARN: Establishing SSL connection without server's identity verification i
id          num      name      addr          phone
-----
1001        돌김돌      2        부산시 부산진구      011-111-1111
1002        김김삼      1        서울시 종로구 종로      010-2222-2222
1003        박김철      3        경기도 성남시 성남로      010-3333-3333
1004        최순기      4        부산 금정구 수림로      010-4444-4444
1005        박일돌      2        경남 창원시 가동정로      010-1234-2111
1006        김희순      3        경남 마산시 회원구      010-3222-2222
1007        김성순      1        부산 동구 을왕로      010-3211-1112
1008        장복수      3        부산 해운대구 루2로      010-3332-1231
1009        김성철      2        부산 동래구 부곡3로      010-2233-2112
1010        정일구      4        부산 연제구 연제로      010-2111-2223
```