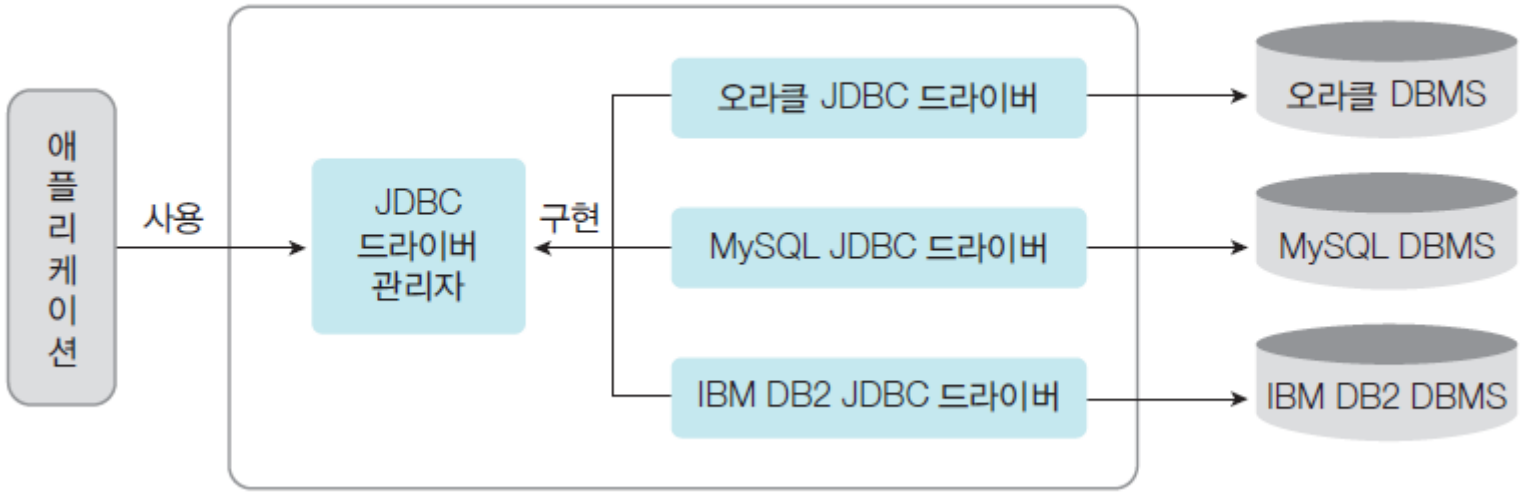
02. JDBC의 이해

JDBC의 개념

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

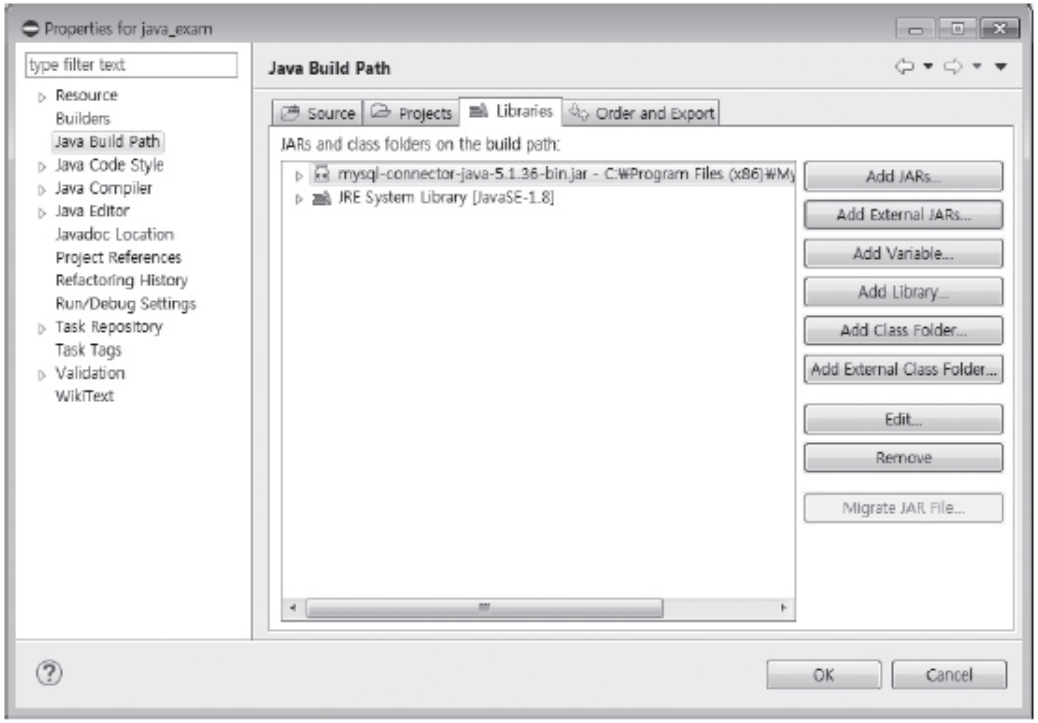
그림 10-22 JDBC의 동작 구조



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 상관없이 표준화된 API를 이용하여 프로그램을 개발할 수 있다.

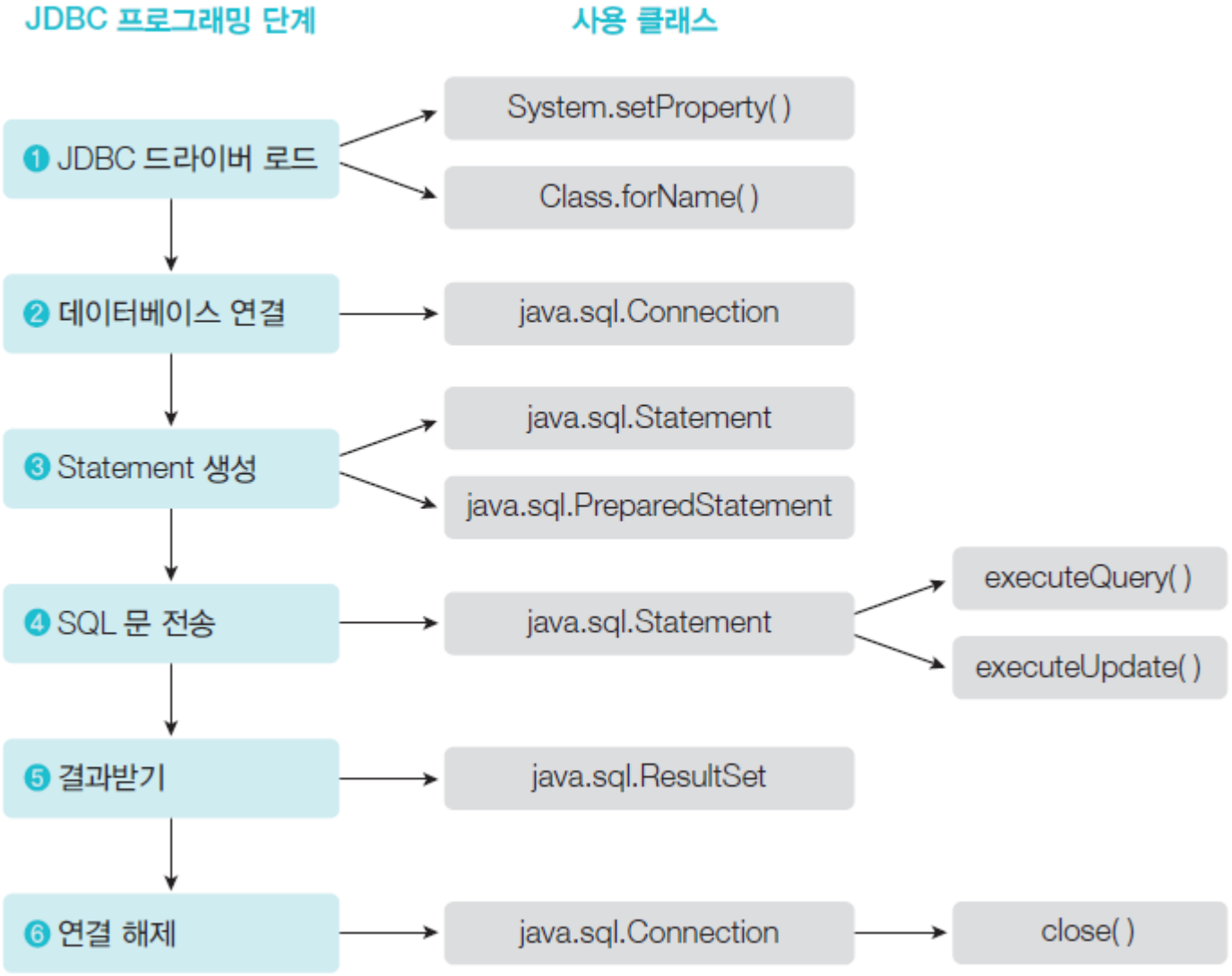
그림 10-23 빌드 경로에 MySQL JDBC 드라이버 추가



02. JDBC의 이해

- JDBC의 개념

그림 10-24 JDBC 프로그래밍의 단계(JSP)



JDBC의 설치 및 연결 설정 : 클래스 패스

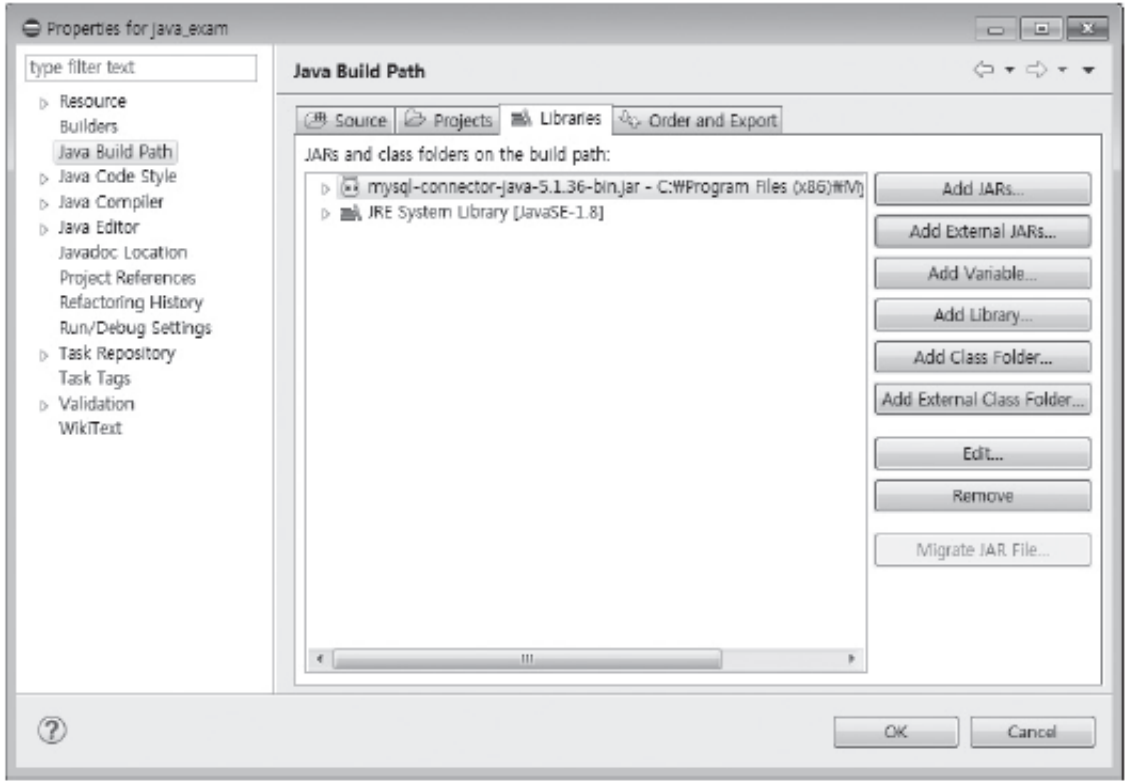
- 자바 가상머신은 프로그램을 실행할 때 이런 클래스 라이브러리의 경로 정보를 바탕으로 해당 라이브러리를 참조하는데, 이것을 클래스 패스Class Path라고 한다.

[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

- JDBC의 설치 및 연결 설정 : JDBC 드라이버 설치

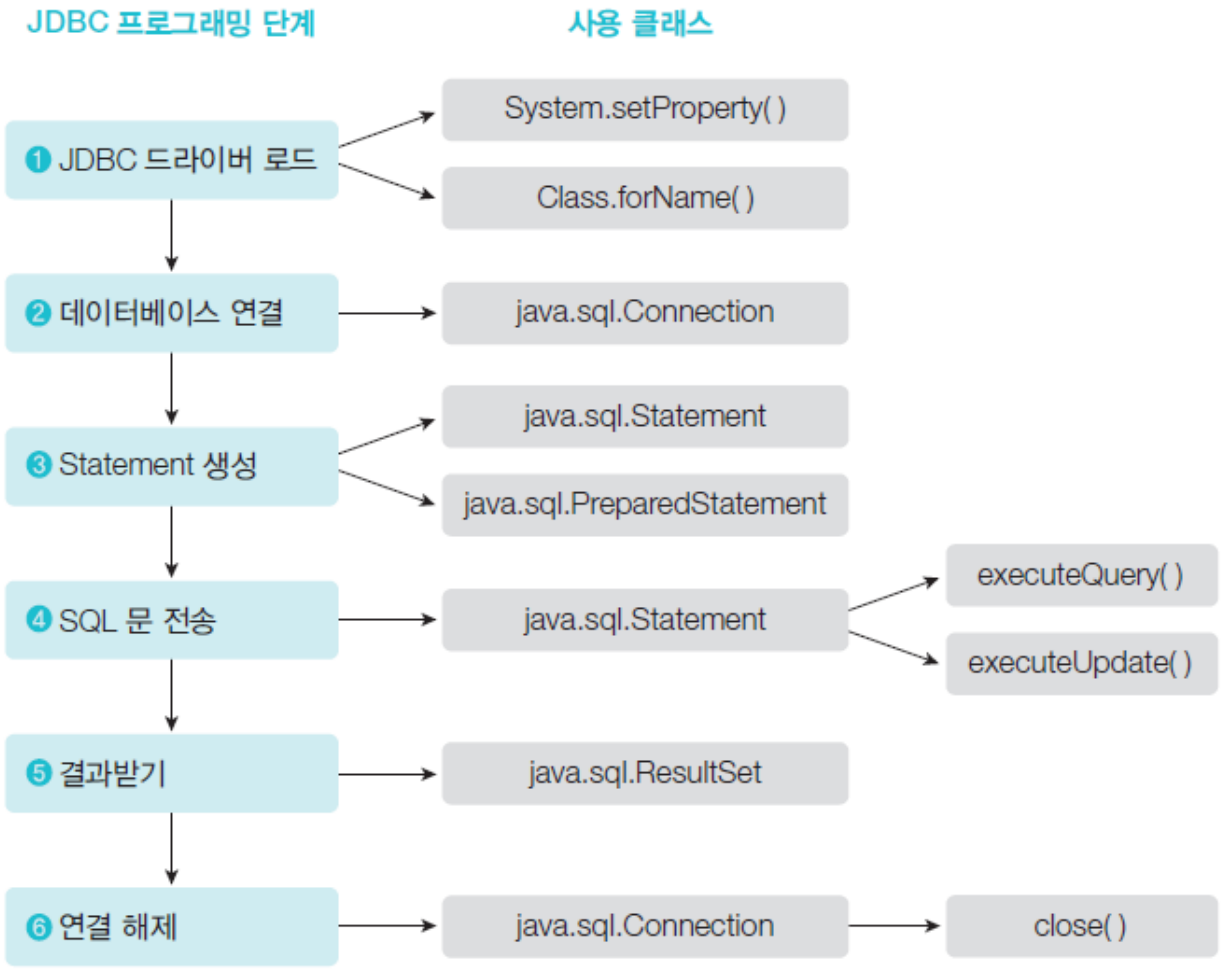
그림 10-23 빌드 경로에 MySQL JDBC 드라이버 추가



02. JDBC의 이해

- JDBC 프로그램 개발 절차

그림 10-24 JDBC 프로그래밍의 단계(JSP)



JDBC 프로그램 개발 절차

1단계 : JDBC 드라이버 로드

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

2단계 : 데이터베이스 연결

▶ **JDBC URL**

```
jdbc:<서브 프로토콜>:<데이터 원본 식별자>
```

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

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.

JDBC 프로그램 개발 절차

2단계 : 데이터베이스 연결

▶ **Connection 클래스 인스턴스 레퍼런스 얻기**

① JDBC_URL : 해당 데이터베이스에 맞게 미리 정의한 문자열이다.

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

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

JDBC 프로그램 개발 절차

3단계 : Statement 생성

▶ **executeQuery( )**

- SELECT 문을 수행할 때 사용

▶ **executeUpdate( )**

- UPDATE 문, DELETE 문 등을 수행할 때 사용한다.

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

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

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.

▶ **Statement의 close( )**

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

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

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();
```

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