

The background features a series of overlapping, wavy, ribbon-like shapes in various shades of green and white, creating a dynamic, flowing effect. The text is centered within a white horizontal band that cuts across the middle of the image.

JDBC Programming

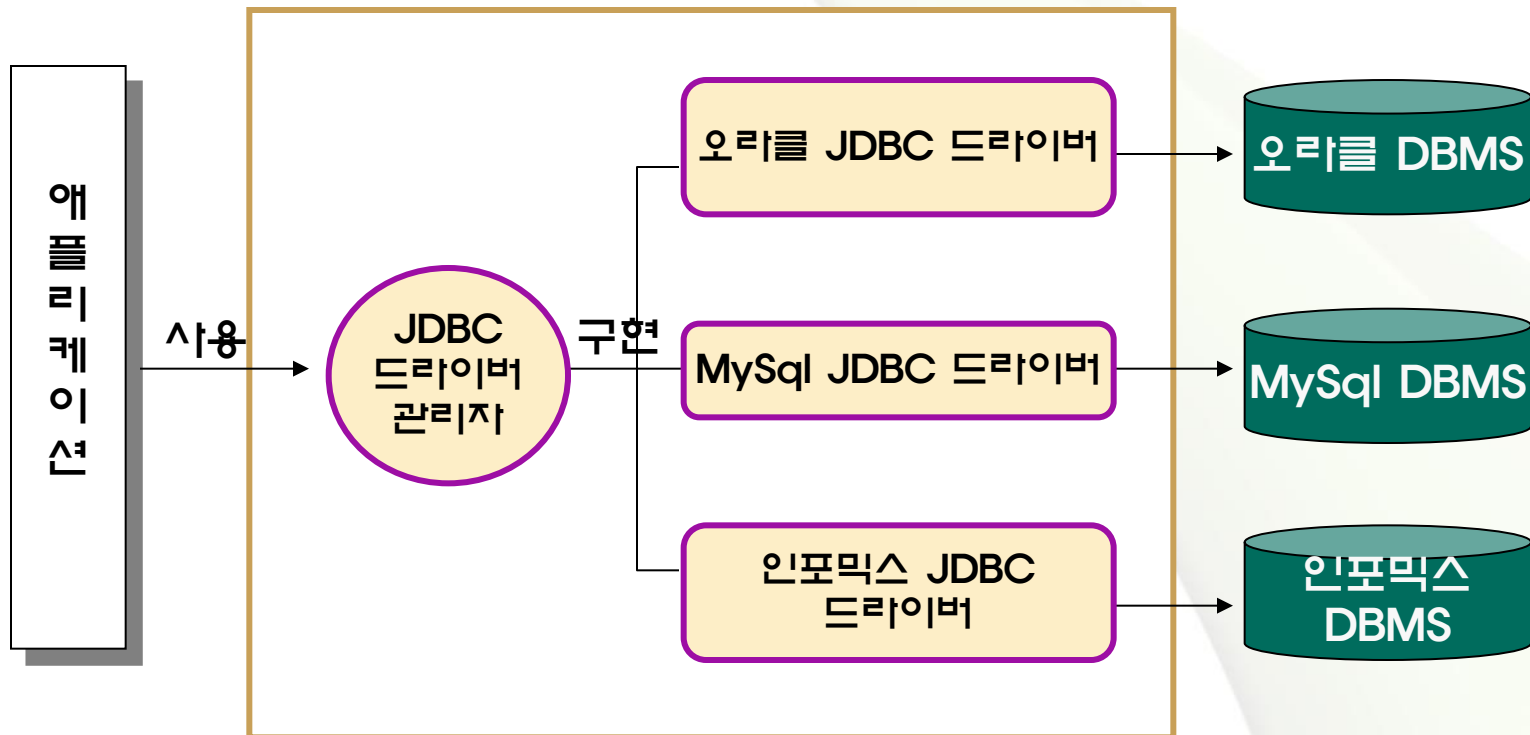
JDBC 개념과 역할

- 자바애플리케이션에서 표준화된 데이터베이스 접근 제공.
- 각 데이터베이스 접속에 대한 상세한 정보를 알 필요 없음.
- 이론적으로는 개발된 애플리케이션에서 DB 변경시 JDBC 드라이버만 교체하면됨.

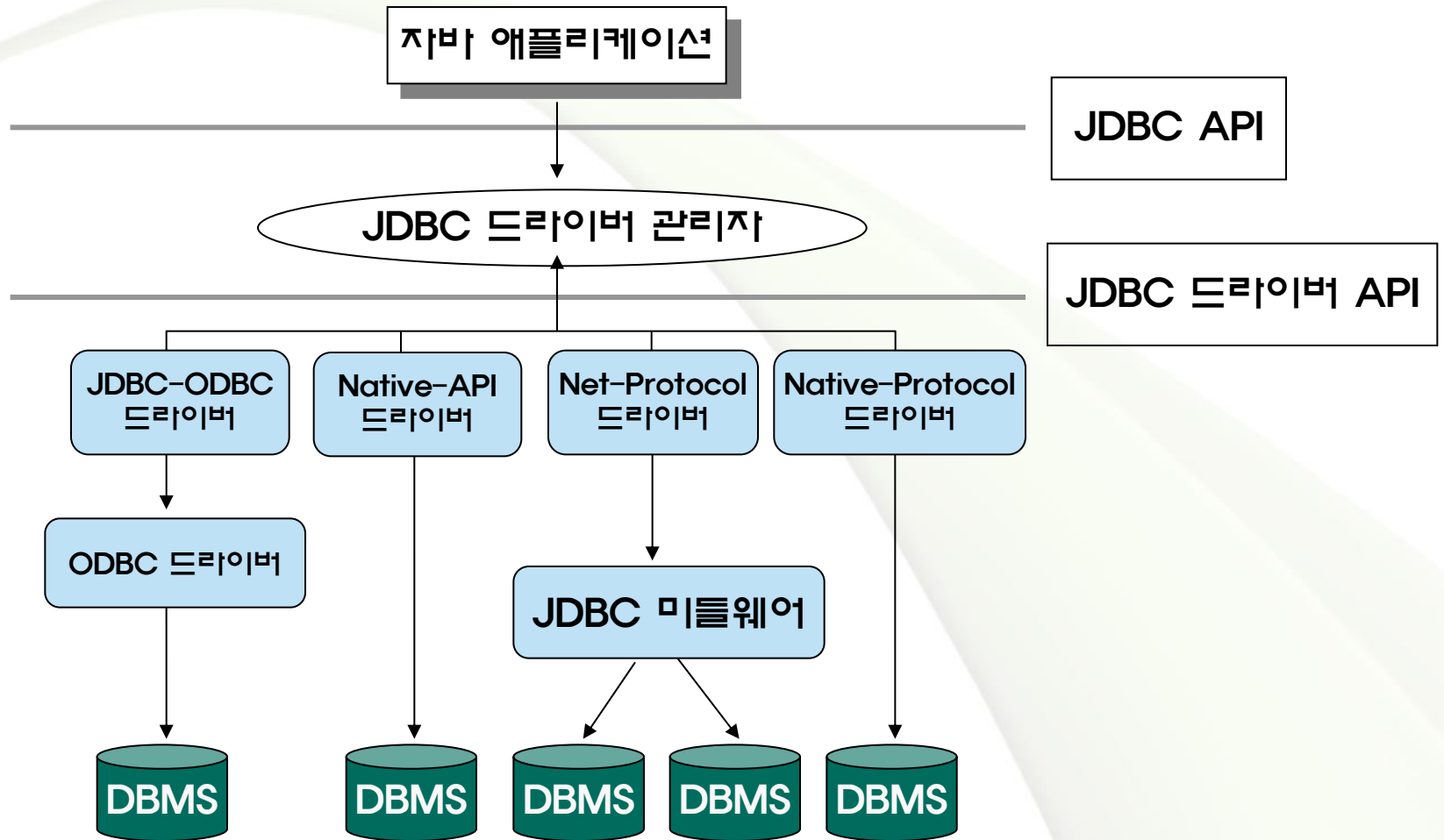
JDBC 개념과 역할

▪ JDBC 구조

- JDBC(Java Database Connectivity)
- 자바에서 데이터베이스를 표준화 된 방법으로 접속할 수 있도록 만든 API 규격
- 데이터베이스 벤더와 상관없이 동일한 개발이 가능함



JDBC 드라이버 구성

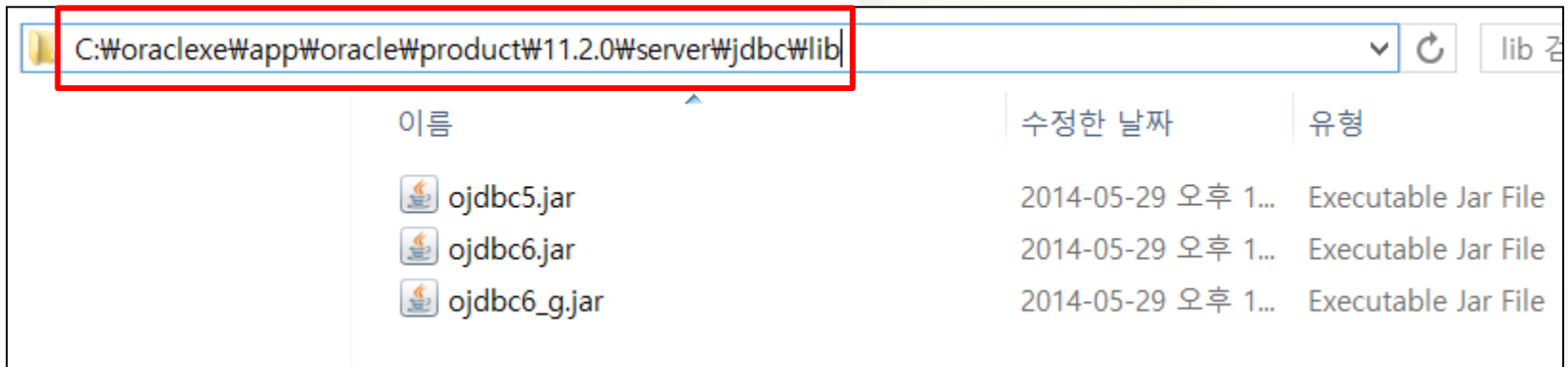


JDBC 드라이버 다운로드

- JDBC 드라이버 선택

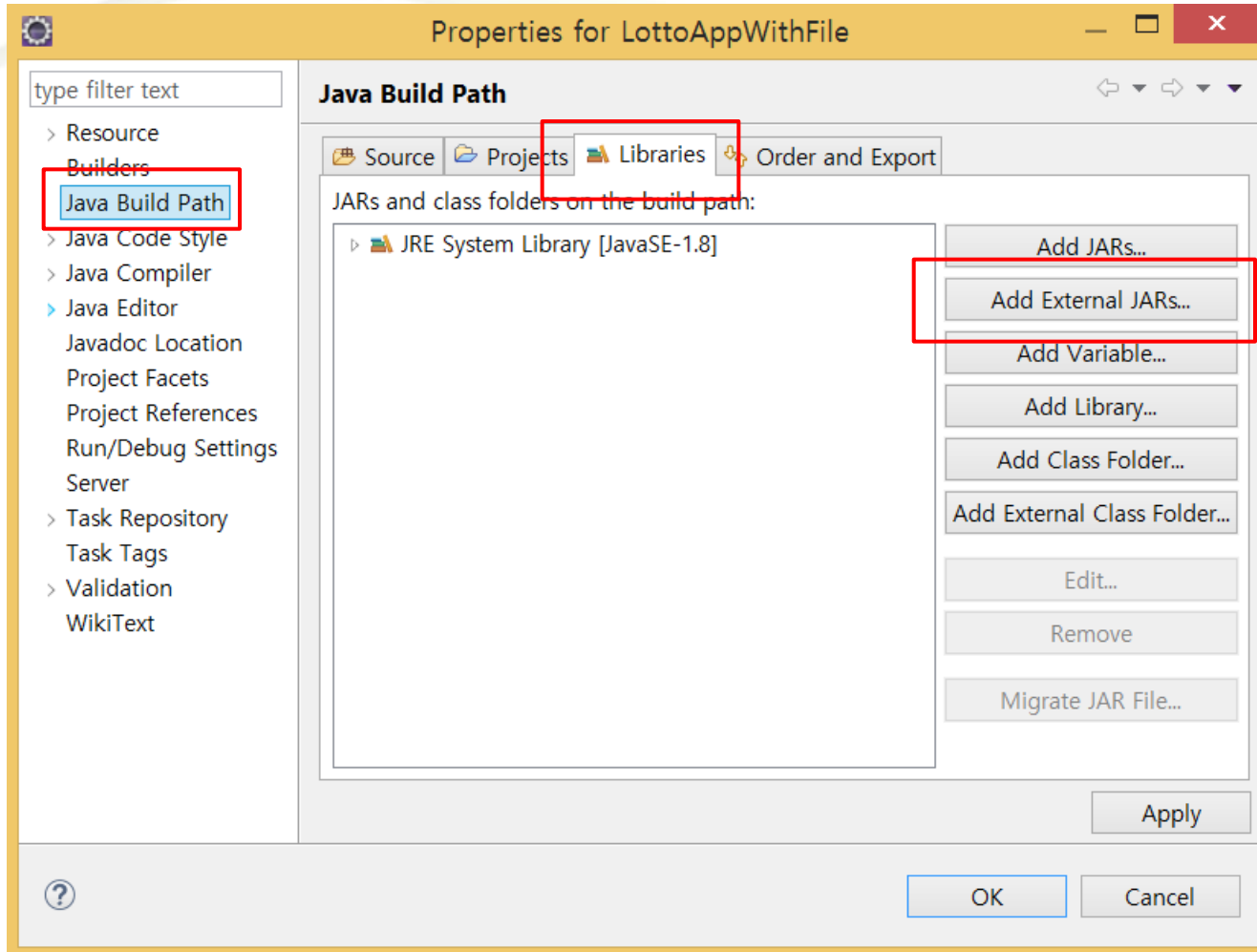
- JDBC 드라이버는 사용하고자 하는 데이터베이스 벤더 별로 제공 됨

- Oracle을 설치한 경우 설치 경로에서 복사



JDBC 드라이버 참조

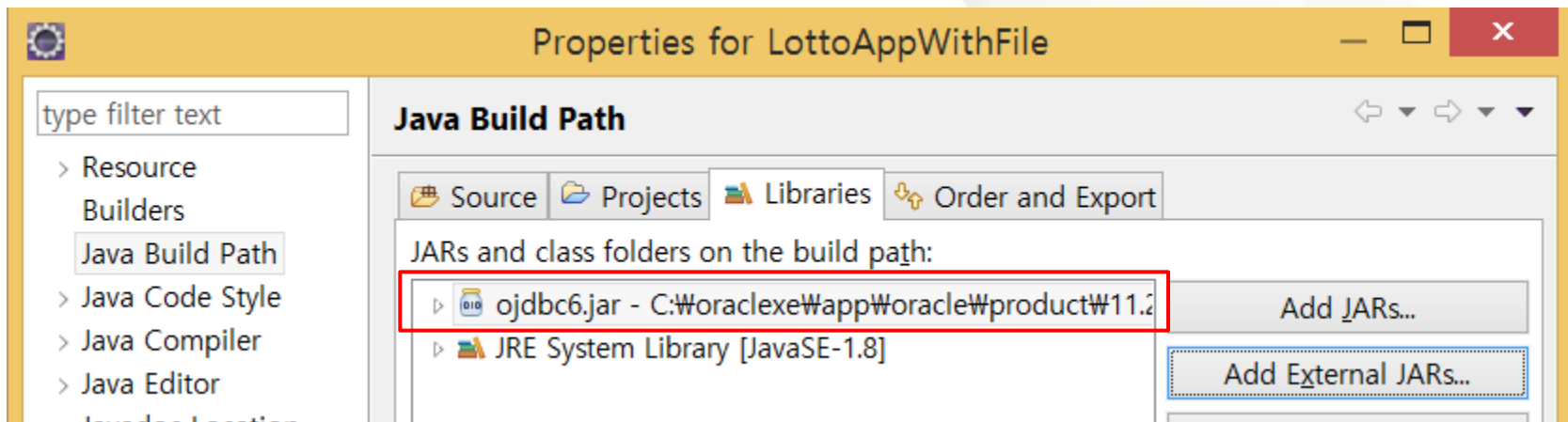
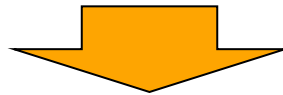
- eclipse에서 프로젝트 선택 → 마우스 우측 버튼 클릭 → Properties 선택



JDBC 드라이버 참조

▪ JDBC Driver 파일 경로에서 파일 선택

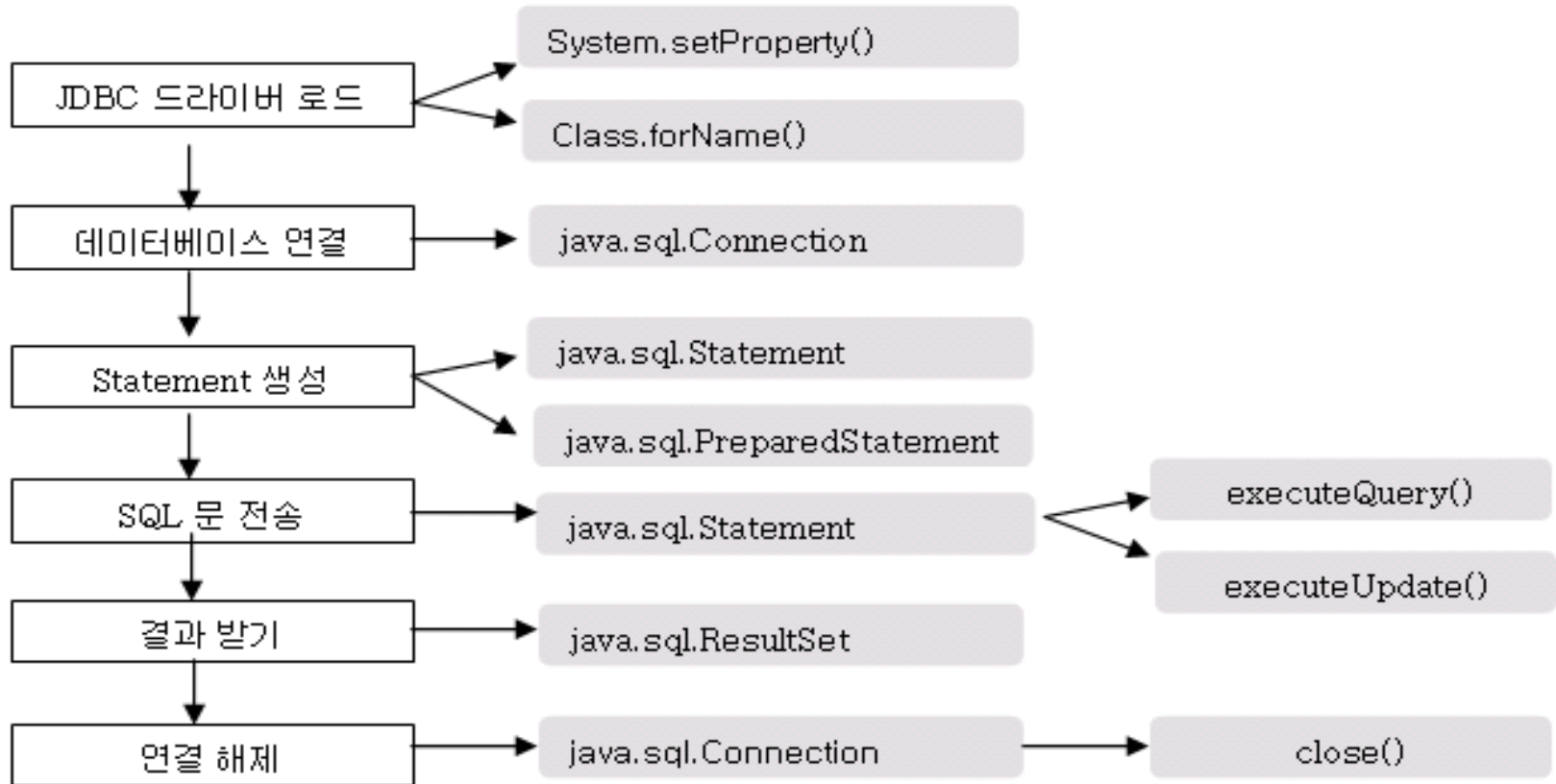
이름	수정한 날짜	유형	크기
 ojdbc5.jar	2014-05-29 오후 1...	Executable Jar File	1,983KB
 ojdbc6.jar	2014-05-29 오후 1...	Executable Jar File	2,102KB
 ojdbc6_g.jar	2014-05-29 오후 1...	Executable Jar File	3,782KB



JDBC 프로그래밍 과정

JDBC 프로그래밍 단계

사용 클래스



JDBC 프로그래밍 단계

- 데이터베이스 드라이버 로드

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

- 데이터베이스 연결

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

- JDBC_URL 구성 = jdbc:mysql://ip:port/db-name

JDBC 프로그래밍 단계

- **연결정보**

- **IP 주소**

- DBMS가 설치된 컴퓨터의 IP 주소 혹은 도메인 이름

- **포트**

- DBMS에서 네트워크를 통한 접속을 처리하기 위해 실행되어 있는
리스너의 사용 포트

JDBC 프로그래밍 단계

- PreparedStatement 생성 및 쿼리 실행

- PreparedStatement 객체 생성시 SQL 문장을 미리 생성하고 변수 부는 별도의 메서드로 대입하는 방식으로 성능과 관리 면에서 모두 권장 되는 방식임

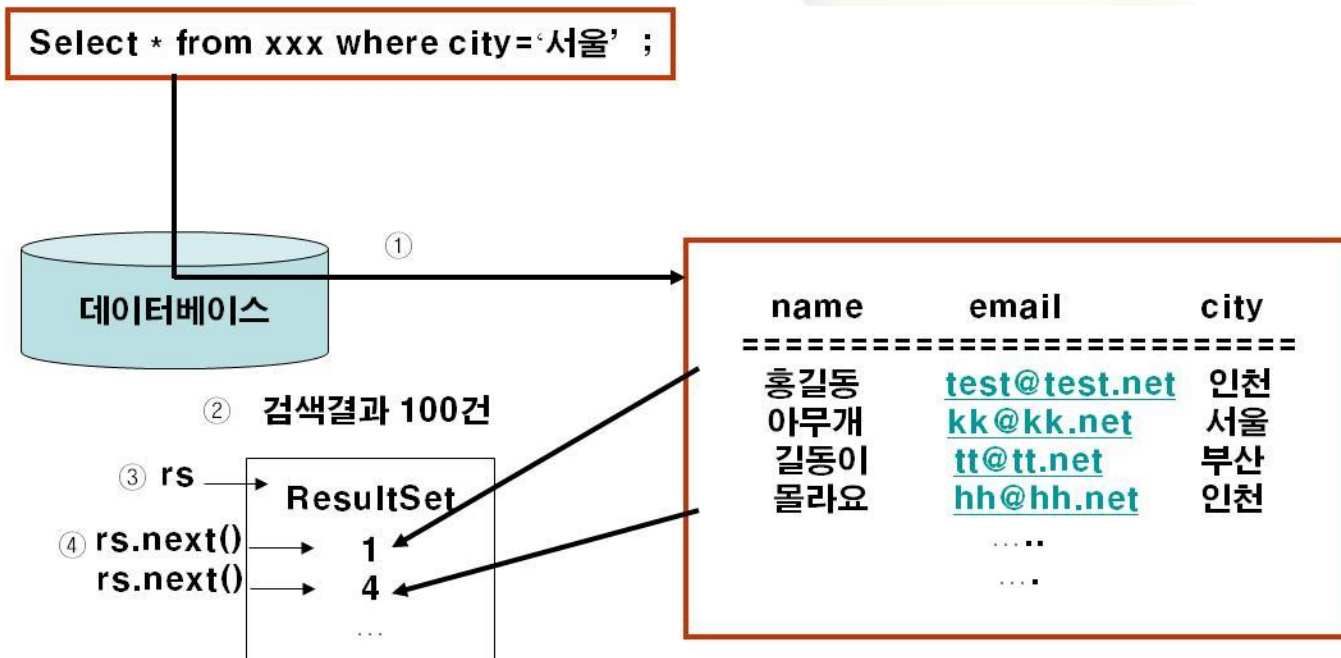
```
PreparedStatement pstmt = conn.prepareStatement( "insert into test  
values(?,?)" );  
pstmt.setString(1,request.getParameter("username");  
pstmt.setString(2, request.getParameter("email");  
pstmt.executeUpdate();
```

JDBC 프로그래밍 단계

▪ 결과 받기

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

- ResultSet은 커서 개념의 연결 포인터
- 기본적으로 next()메서드를 통해 행 이동



JDBC 프로그래밍 과정

▪ 결과 받기

```
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();
```

JDBC 프로그래밍 단계

▪ 연결해제

- Connection 을 close() 해주지 않으면 사용하지 않는 연결이 유지됨.
- DB 자원을 낭비하게 됨.

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

JDBC 프로그래밍 단계 요약

