

**JDBC**

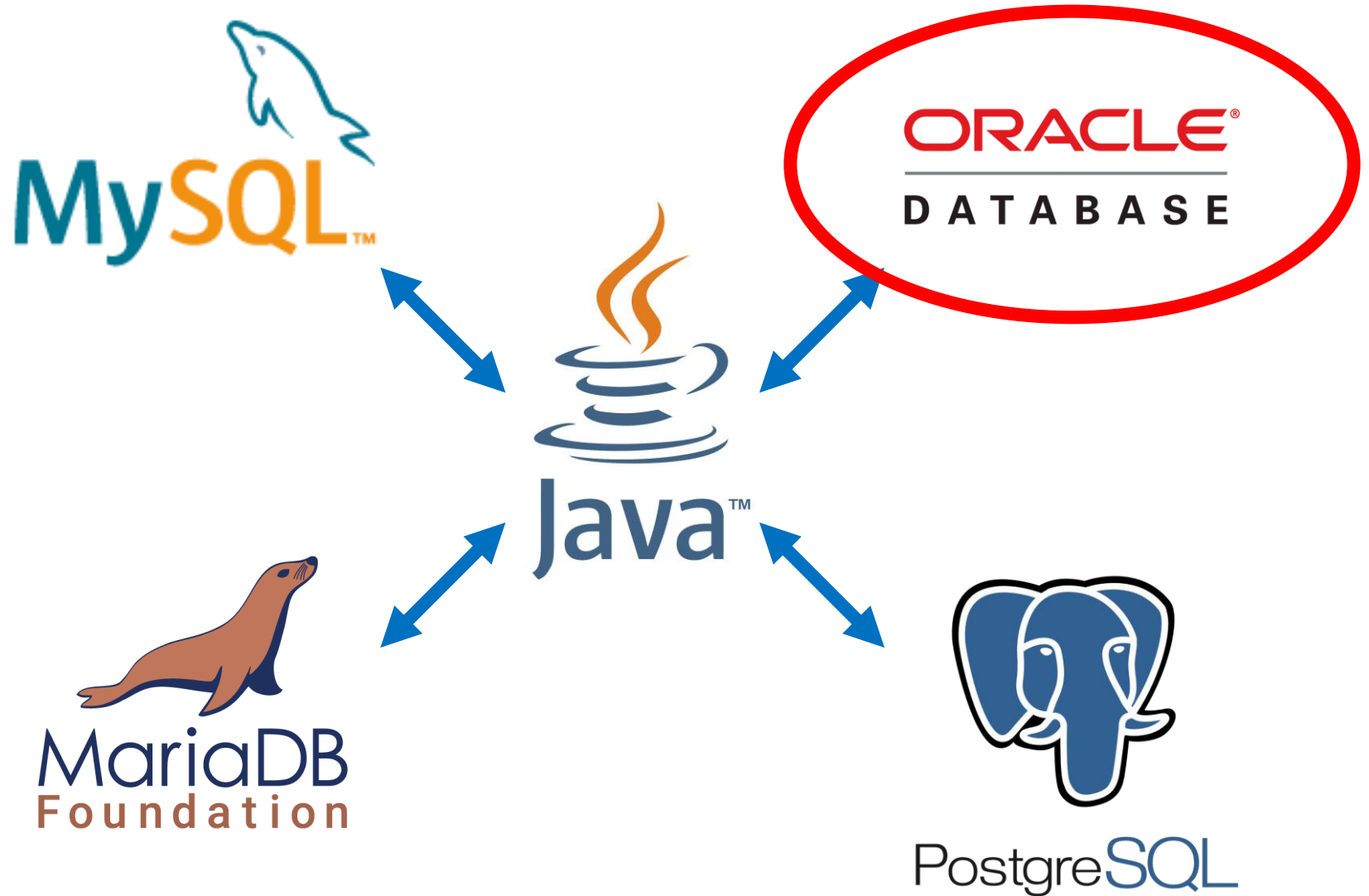
# JDBC?

---

## Java Database Connectivity

자바와 데이터베이스를 연결하는 API

# JDBC?



# JDBC 설정

---

**JDK8**  
+  
**Oracle Database 11g Release 2**  
↓  
**ojdbc6.jar**

C:\Woraclexe\Wapp\Woracle\Wproduct\W11.2.0\server\Wjdbc\Wlib

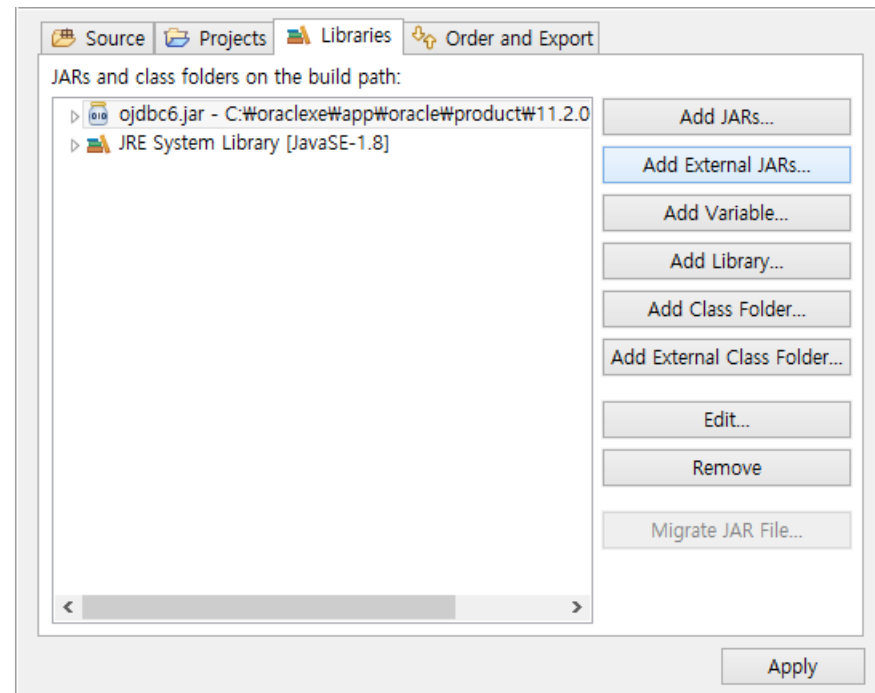
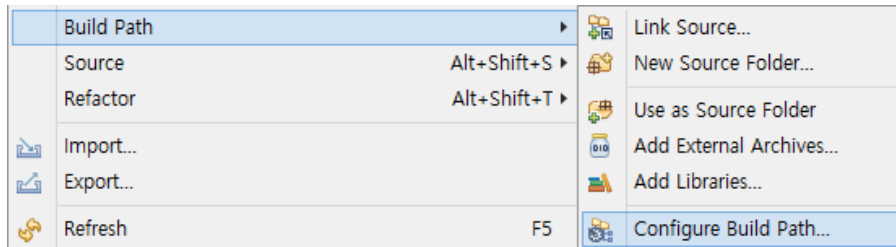
혹은

<https://www.oracle.com/technetwork/database/enterprise-edition/jdbc-112010-090769.html>

# JDBC 설정

- 프로젝트에 ojdbc6.jar 추가하기

1. 프로젝트 우클릭
2. Build Path ► Configure Build Path...
3. Libraries 탭의 Add External JARs...
4. ojdbc6.jar 추가 후 Apply



# Java – Database 연동

---

- **Java에서 Database 연동하기**

1. JDBC 드라이버 로드
2. DB 연결
3. Statement 생성
4. Query 수행
5. 결과 처리
6. Resources 반환

# Java – Database 연동

## 1. JDBC 드라이버 로드

2. DB 연결
3. Statement 생성
4. Query 수행
5. 결과 처리
6. Resources 반환

- JDBC 드라이버를 메모리에 적재하는 단계
- `oracle.jdbc.driver.OracleDriver` 객체 사용
- **`Class.forName("드라이버 객체명");`**
- `ClassNotFoundException` 처리

# Java – Database 연동

1. JDBC 드라이버 로드
2. **DB 연결**
3. Statement 생성
4. Query 수행
5. 결과 처리
6. Resources 반환

- 데이터베이스에 접속하여 Connection 객체를 생성하는 단계
- DriverManager.getConnection() 메소드 사용
- URL, 유저명, 비밀번호 필요
- URL의 형태: jdbc:oracle:thin:@localhost:1521:xe

thin 타입 오라클  
JDBC 드라이버

접속할  
데이터베이스 주소

\*Listener 포트 번호

\*SID

\*Listener: 오라클 Database에서 외부와의 연결을 관리하는 컴포넌트

\*SID: 오라클 Database 인스턴스가 사용하는 이름



# Java – Database 연동

1. JDBC 드라이버 로드
2. DB 연결
3. **Statement 생성**
4. Query 수행
5. 결과 처리
6. Resources 반환

- Query를 수행할 Statement 객체를 생성하는 단계
- Connection 객체의 createStatement() 메소드 사용
- SQLException 처리

# Java – Database 연동

1. JDBC 드라이버 로드
2. DB 연결
3. Statement 생성
4. **Query 수행**
5. 결과 처리
6. Resources 반환

- Statement 객체를 이용하여 실질적으로 데이터베이스에 Query를 수행하는 단계
- SELECT의 경우:
  - executeQuery("Query 문장") 메소드 사용
  - ResultSet 객체 반환
- INSERT, UPDATE, DELETE의 경우:
  - executeUpdate("Query 문장") 메소드 사용
  - 영향을 받은 레코드의 수(int) 반환

# Java – Database 연동

1. JDBC 드라이버 로드
2. DB 연결
3. Statement 생성
4. Query 수행
- 5. 결과 처리**
6. Resources 반환

- SELECT Query를 수행한 후 반환된 ResultSet 객체에서 데이터를 꺼내는 단계
- ResultSet 객체의 주요 메소드
  - next(): 다음 레코드가 존재할 경우 true, 아니면 false를 반환
  - getInt(컬럼 번호 또는 "컬럼명"): NUMBER 자료형에 대한 데이터 선택
  - getDouble(컬럼 번호 또는 "컬럼명"): NUMBER 자료형에 대한 데이터 선택
  - getString(컬럼 번호 또는 "컬럼명"): VARCHAR2 자료형에 대한 데이터 선택
  - getDate(컬럼 번호 또는 "컬럼명"): DATA 자료형에 대한 데이터 선택
  - getMetaData(): 테이블에 대한 정보를 담은 객체를 반환

# Java – Database 연동

1. JDBC 드라이버 로드
2. DB 연결
3. Statement 생성
4. Query 수행
5. 결과 처리
6. **Resources 반환**

- ResultSet, Statement, Connection 객체 순으로 자원 반환
- close() 메소드 사용, SQLException 처리
- ResultSet 객체는 Statement 객체가 닫힐 때 자동으로 닫힘
- Statement 객체는 Connection 객체가 닫힐 때 자동으로 닫힘
- 하지만 직접 닫아주는 것을 권장

# 또 다른 Statement

- **PreparedStatement**
  - 매개변수의 값을 동적으로 할당하기 위해 사용
  - Query를 여러 번 실행하기 위해 사용
  - 생성할 때 Query 문장을 미리 입력받음
  - 가독성 및 유지보수가 좋음