

K G 아 이 티 뱅 크

J A V A

J A V A

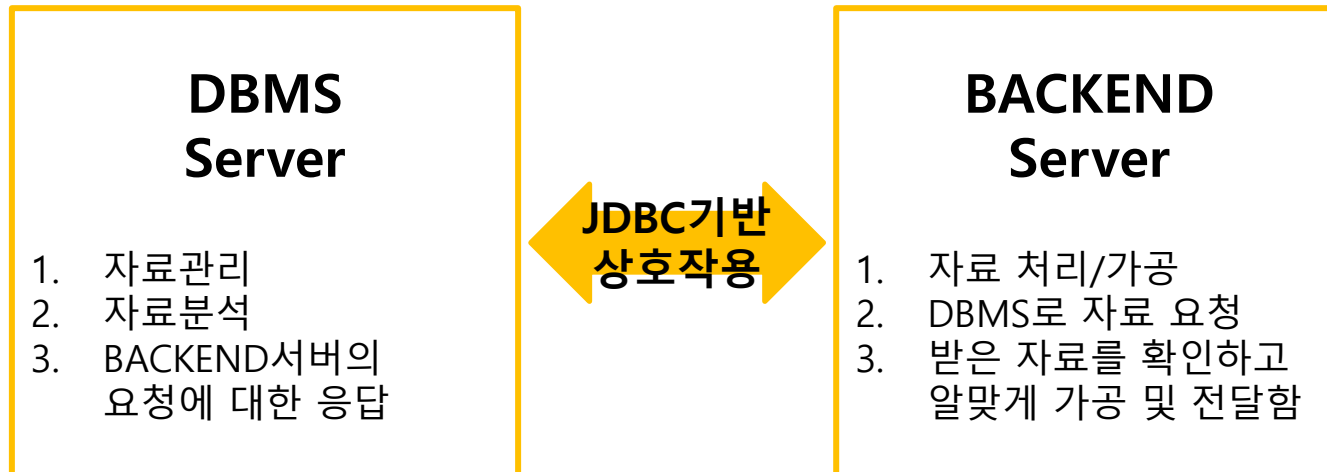
JDBC

❖ Java Database Connectivity : 데이터베이스 연결용 API

- Oracle DBMS도 JAVA로 작동하지만 구조자체가 너무 다름
- 이를 유효/적절하게 사용할 수 있게 해주는 도구
- 이를 통하여 Oracle DBMS와 상호작용을 수행하게 됨

❖ Oracle DBMS와는 네트워크를 경유하여 상호작용을 함

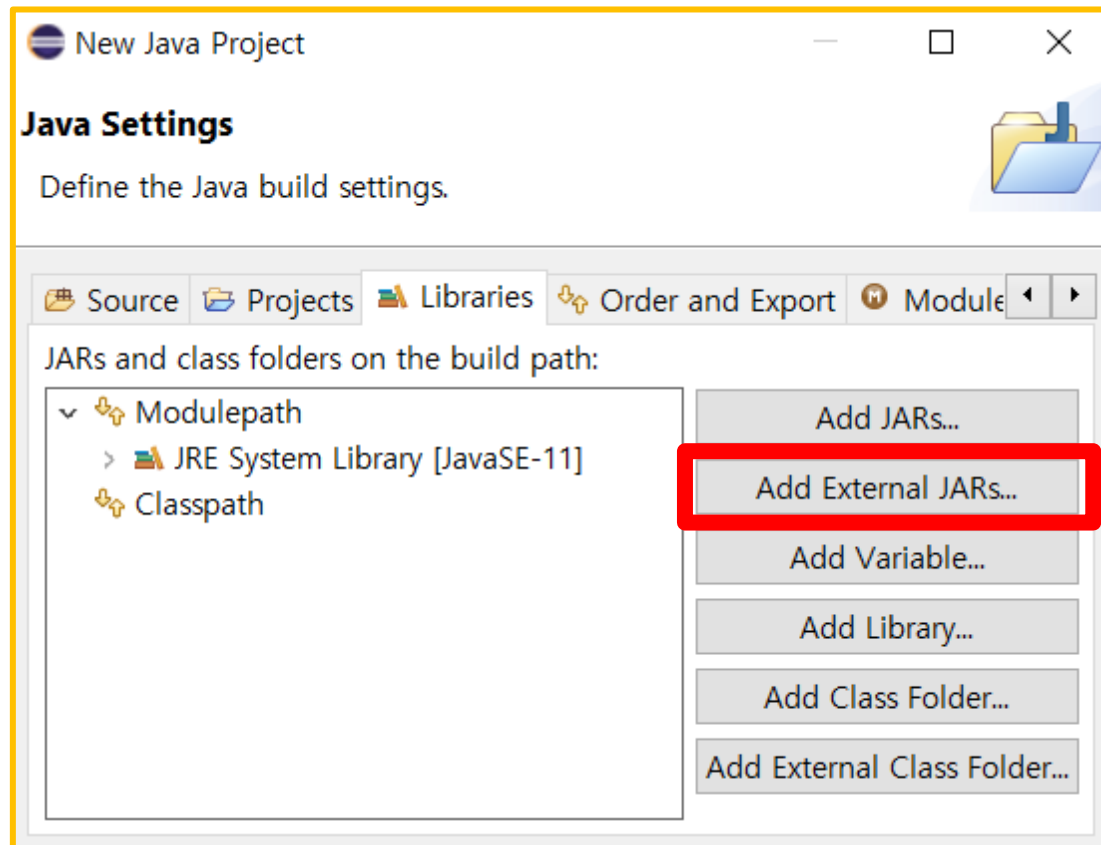
- 상호작용시 쿼리문을 보내고, 그 결과를 받는 방식이 됨
- 네트워크를 통하는 편이 접속 및 사용내역 로깅에 유리함



JDBC

❖ JDBC API는 내장된 것이 아니기에 사용하기 전에 주의해야 함

- ① JDBC API가 포함된 파일을 먼저 불러와야 함
- ② 올바르게 불러와서 인식되도록 추가적인 코드 실행이 필요함



상호작용하기

상호작용하기

❖ 불러온 API는 이하의 과정을 거쳐 DBMS와 연결됨

- ① 연결에 필요한 드라이버를 로딩함
- ② 접속정보를 구성하여 접속을 수행함
- ③ DBMS와 상호작용을 수행함
- ④ 상호작용이 끝났으면 연결을 종료함

```
try {
    Class.forName("oracle.jdbc.driver.OracleDriver");
    String url = "jdbc:oracle:thin:@localhost:1521:XE";
    String id = "HR";
    String pw = "1234";
    Connection conn = DriverManager.getConnection(url, id, pw);
    /* 상호작용 수행 */
    conn.close();
} catch (ClassNotFoundException e) {
    System.out.println(e);
} catch (SQLException e) {
    System.out.println(e);
}
```

상호작용하기

❖ 상호작용시 PreparedStatement 객체를 생성하여 사용함

- 어떤 쿼리문인지에 따라 실행메서드를 달리 사용함

```
PreparedStatement psmt = null;  
String query = null;
```

```
// excuteQuery() : SELECT에 대응됨
```

```
query = "SELECT * FROM regions";  
psmt = conn.prepareStatement(query);  
ResultSet records = psmt.executeQuery();
```

```
// excuteUpdate() : INSERT, UPDATE, DELETE에 대응됨
```

```
query = "DELETE FROM regions WHERE region_id = 1";  
psmt = conn.prepareStatement(query);  
int count = psmt.executeUpdate();
```

```
// excute() : 범용이지만 응용하기 어려운 구조를 가짐
```

```
query = "SELECT * FROM regions";  
psmt = conn.prepareStatement(query);  
boolean rsExist = psmt.execute();
```