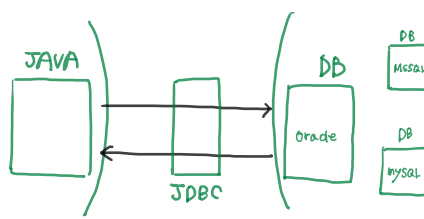


18강. 데이터 베이스 - II

- JDBC 살펴보기
- Statement객체 살펴보기

Lecturer Kim Myoung-Ho
Nickname 불스
blogstudy@naver.com

18-1. JDBC 살펴보기



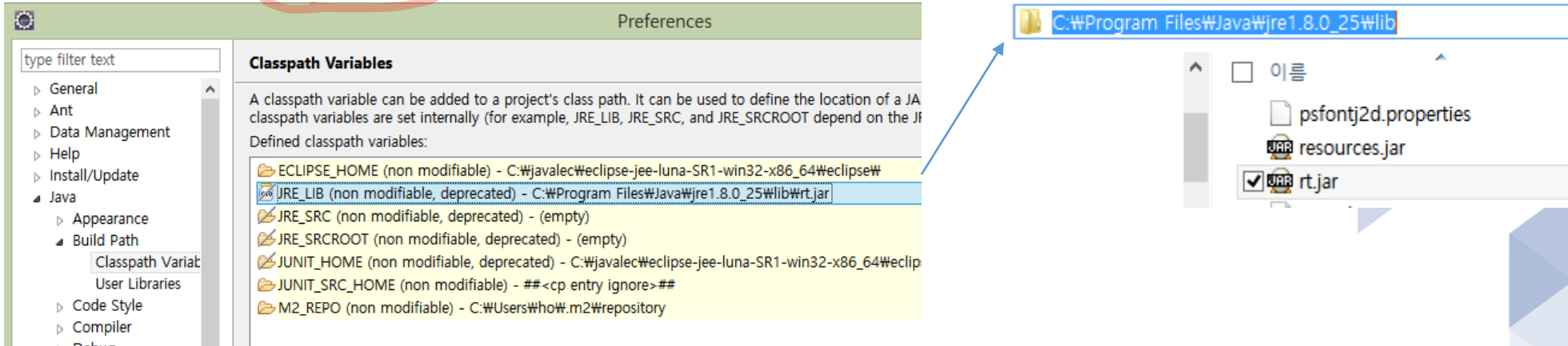
JAVA 프로그램에서 SQL문을 실행하여 데이터를 관리하기 위한 JAVA API입니다.

JDBC의 특징은 다양한 데이터 베이스에 대해서 별도의 프로그램을 만들 필요 없이, 해당 데이터 베이스의 JDBC를 이용하면 하나의 프로그램으로 데이터 베이스를 관리할 수 있습니다.

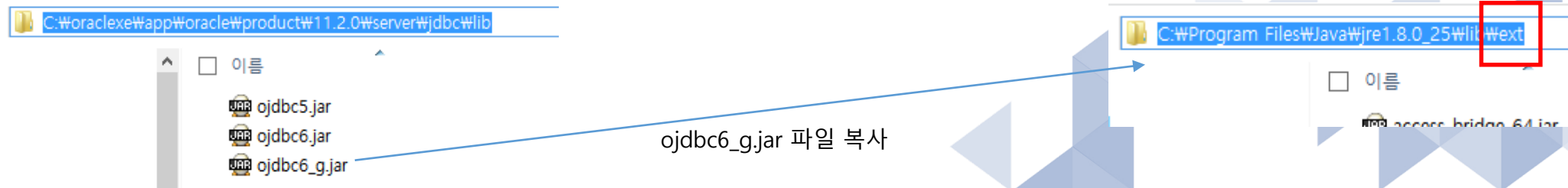
우리는 Oracle을 사용하므로, Oracle용 JDBC를 사용하며, 이것은 오라클을 설치하면 자동으로 설치 되고, 이클립스에서 해당 클래스 파일을 복사하면 됩니다.

오라클 드라이버를 사용하기 위한 **ojdbc6_g.jar** 파일 복사

이클립스 설정되어 있는 **JAVA클래스 패스** : C:\Program Files\Java\jre1.8.0_25\lib → **여기 밑의 ext 디렉터리 밑에 복사**

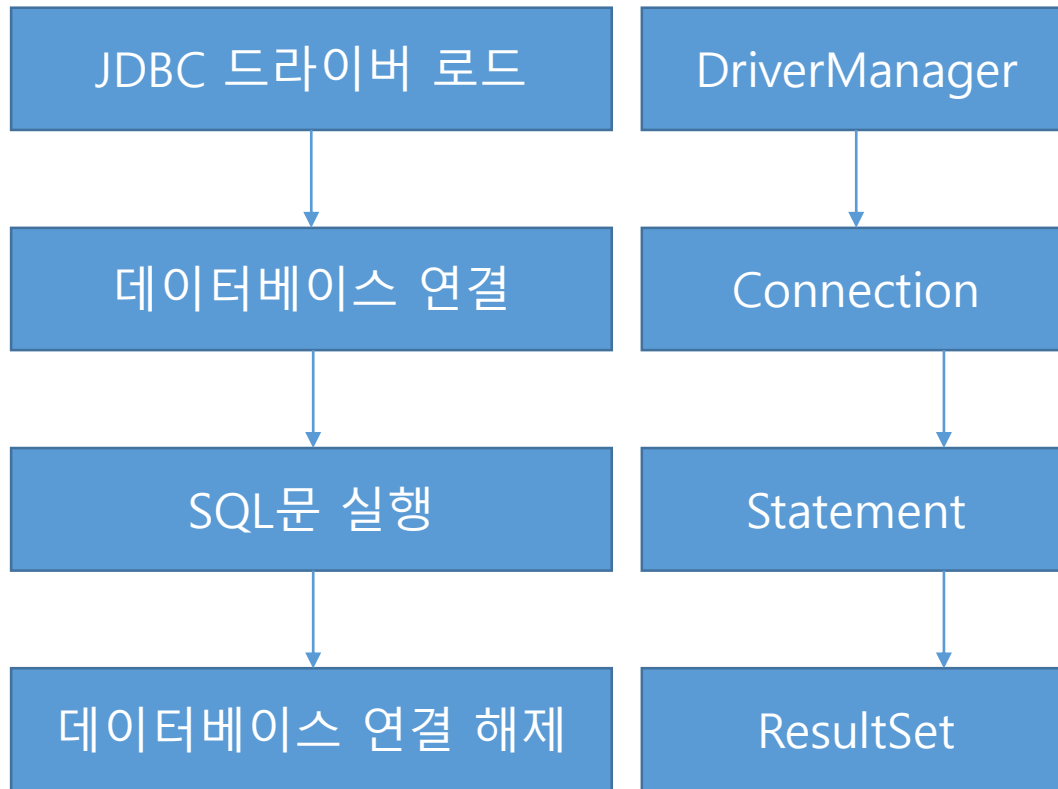


오라클 JDBC 클래스 패스 : C:\Woraclexe\app\oracle\product\11.2.0\server\jdbc\lib\ojdbc6_g.jar



18-1. JDBC 살펴보기

데이터 베이스 연결 순서



`Class.forName("oracle.jdbc.driver.OracleDriver");`
: 메모리에 OracleDriver가 로드 됩니다.

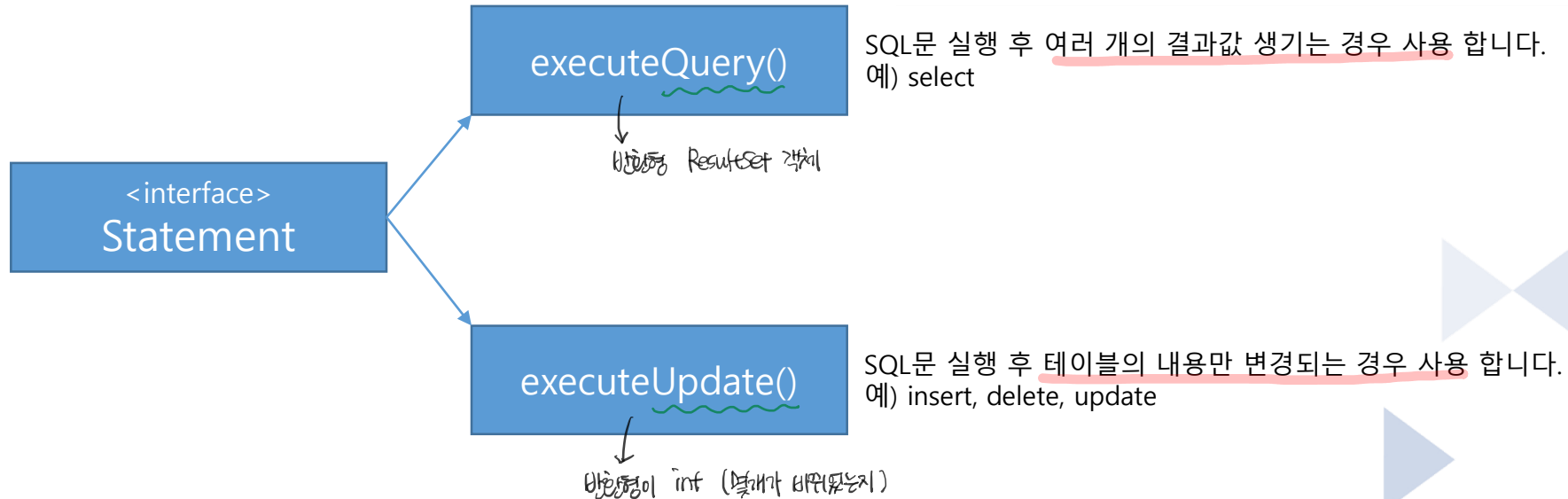
`DriverManager.getConnection(JDBC URL, 계정아이디, 비밀번호);`
: Connection객체 생성 합니다.

`connection.createStatement();`
: Statement객체를 통해 SQL문이 실행 됩니다.

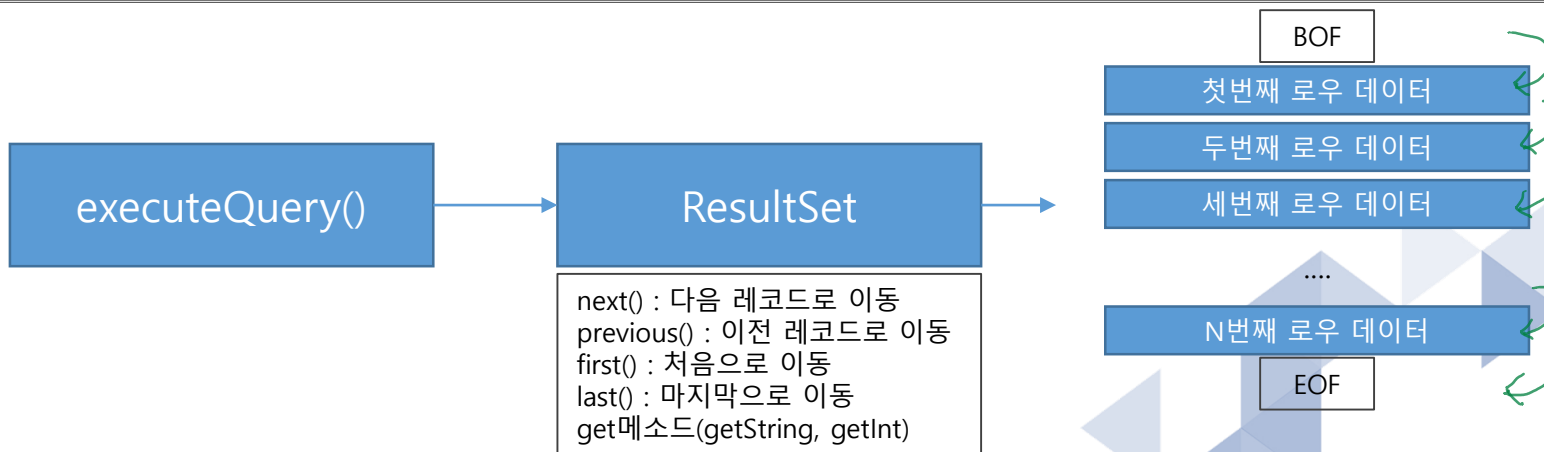
`statement.executeQuery(), statement.executeUpdate()`
: SQL문의 결과값을 ResultSet객체로 받습니다.

↓
Close 를 통해 자원을 해제

18-2. Statement 객체 살펴보기



executeQuery() 실행 후 반환 되는 레코드 셋



18-2. Statement객체 살펴보기

JDBC예제 (jsp_18_2_ex1_jdbcex)

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\who>sqlplus

SQL*Plus: Release 11.2.0.2.0 Production on 화 1월 6 14:18:11 2015

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Enter user-name: scott
Enter password:

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> create table member (
  2 id varchar2(20),
  3 pw varchar2(20),
  4 name varchar2(20),
  5 phone varchar2(20));

Table created.
```

```
SQL> insert into member (id, pw, name, phone) values ('abc', '123', '홍길동', '010-1234-5678');

1 row created.

SQL> insert into member (id, pw, name, phone) values ('def', '456', '홍길순', '010-9012-3456');

1 row created.

SQL> insert into member (id, pw, name, phone) values ('ghi', '789', '홍길자', '010-7890-1234');

1 row created.

SQL> insert into member (id, pw, name, phone) values ('jkl', '234', '홍길목', '010-5678-9012');

1 row created.

SQL> commit
  2 ;

Commit complete.
```

18-2. Statement객체 살펴보기

JDBC예제 (jsp_18_2_ex1_jdbcex)

```
<%!
Connection connection;
Statement statement;
ResultSet resultSet;

String driver = "oracle.jdbc.driver.OracleDriver";
String url = "jdbc:oracle:thin:@localhost:1521:xe";
String uid = "scott";
String upw = "tiger";
String query = "select * from member";

%>

<%
try{
    Class.forName(driver); //JDBC 드라이버 로드
    connection = DriverManager.getConnection(url, uid, upw); // 데이터베이스 연결
    statement = connection.createStatement(); // SQL 실행
    resultSet = statement.executeQuery(query);
    // 이 객체에 정보들이 담겼다.
    while(resultSet.next()){
        String id = resultSet.getString("id");
        String pw = resultSet.getString("pw");
        String name = resultSet.getString("name");
        String phone = resultSet.getString("phone");

        out.println("아이디 : " + id + ", 비밀번호 : " + pw + ", 이름 : " + name + ", 전화번호 : " + phone + "<br />");
    }
}
```

DB와 관련된 작업은 try, catch 문을 사용

memberData.jsp

Insert title here

http://localhost:8181/jsp_18_2_ex1_jdbcex/memberData.jsp

아이디 : abc, 비밀번호 : 123, 이름 : 홍길동, 전화번호 : 010-1234-5678
 아이디 : def, 비밀번호 : 456, 이름 : 홍길순, 전화번호 : 010-9012-3456
 아이디 : ghi, 비밀번호 : 789, 이름 : 홍길자, 전화번호 : 010-7890-1234
 아이디 : jkl, 비밀번호 : 234, 이름 : 홍길목, 전화번호 : 010-5678-9012

MSSQL JDBC 사용법.

```
String driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";
String url = "jdbc:sqlserver://localhost:1433;database=DB이름;integratedSecurity=true";
```

↑ 내사 연결된 TCP/IP 포트