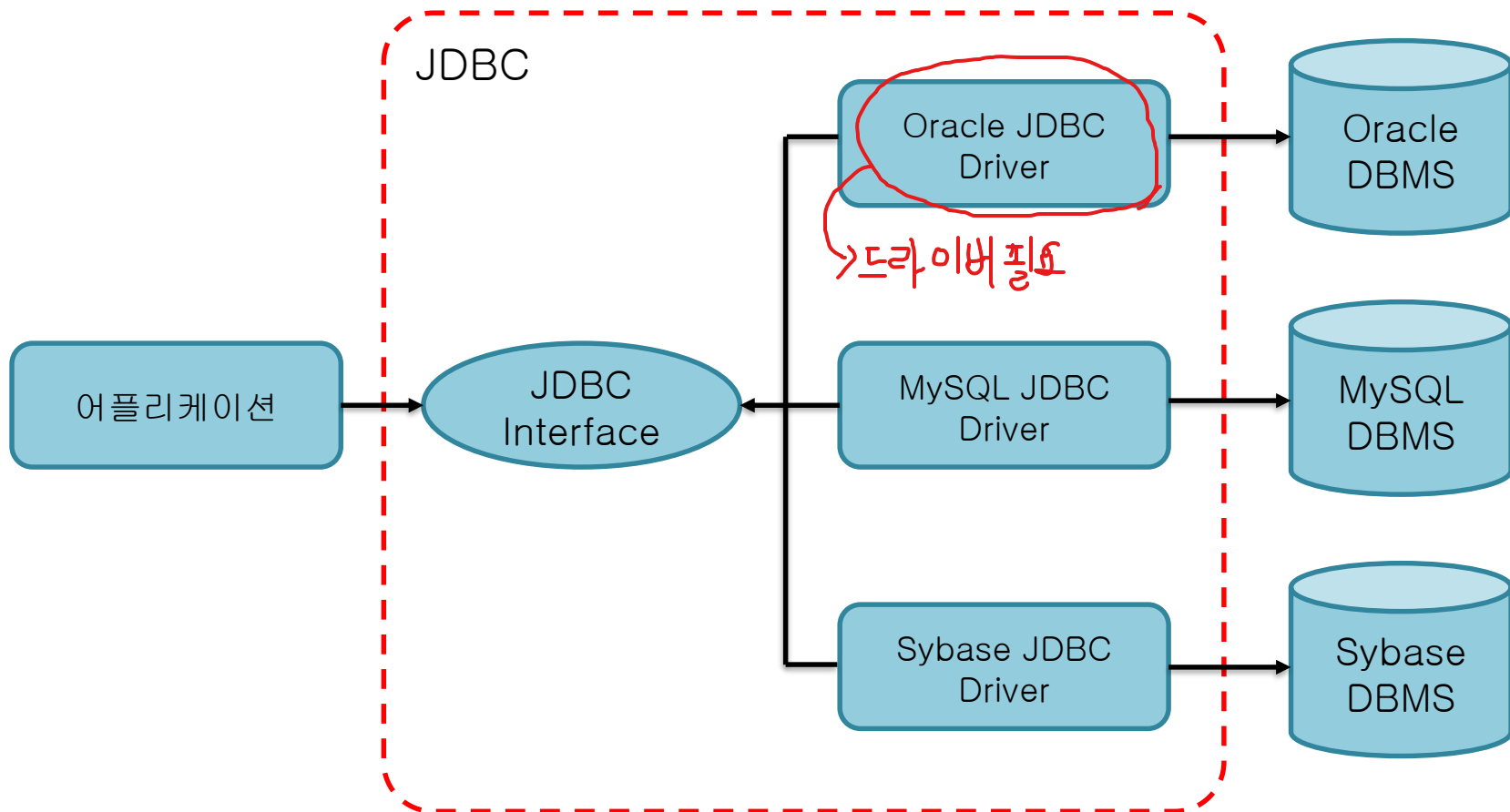


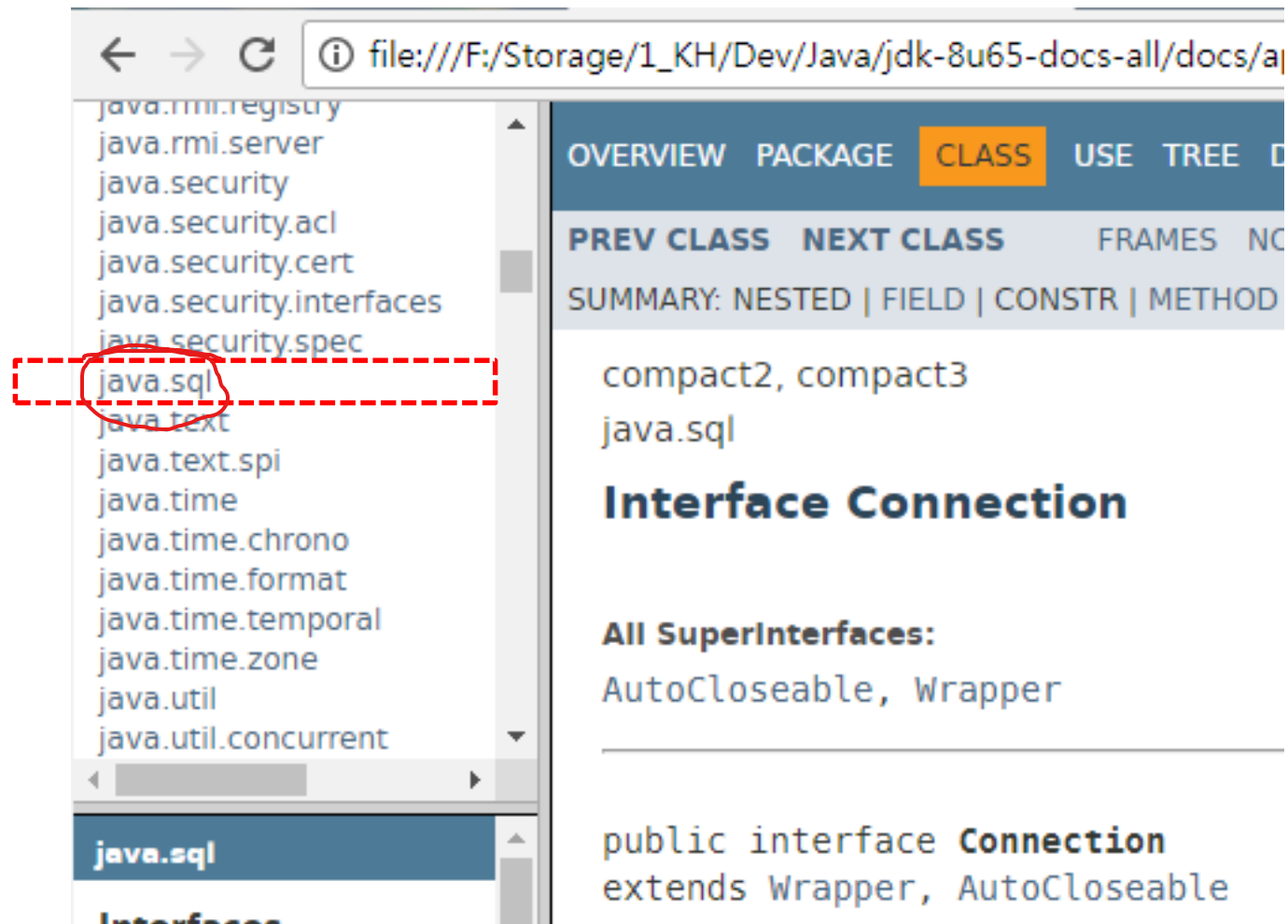
JDBC

JDBC란?

자바 언어에서 Database에 접근할 수 있게 해 주는 Programming API이다.



java.sql 패키지



The screenshot shows a Java IDE interface. On the left, a package explorer lists various Java packages. The `java.sql` package is highlighted with a red dashed rectangle. On the right, the details pane for the `java.sql` package is displayed. It includes tabs for OVERVIEW, PACKAGE, CLASS, USE, and TREE. The CLASS tab is selected. Below the tabs, there are links for PREVIOUS CLASS, NEXT CLASS, FRAMES, and NO FRAMES. A summary section shows 'compact2, compact3' and 'java.sql'. The main section is titled 'Interface Connection'. Under 'All SuperInterfaces:', it lists 'AutoCloseable, Wrapper'. At the bottom, it shows the declaration: 'public interface Connection extends Wrapper, AutoCloseable'.

file:///F:/Storage/1_KH/Dev/Java/jdk-8u65-docs-all/docs/api/

java.rmi.registry
java.rmi.server
java.security
java.security.acl
java.security.cert
java.security.interfaces
java.security.spec
java.sql
java.text
java.text.spi
java.time
java.time.chrono
java.time.format
java.time.temporal
java.time.zone
java.util
java.util.concurrent

OVERVIEW PACKAGE **CLASS** USE TREE

PREV CLASS NEXT CLASS FRAMES NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHOD

compact2, compact3
java.sql

Interface Connection

All SuperInterfaces:
AutoCloseable, Wrapper

public interface **Connection**
extends Wrapper, AutoCloseable

OJDBC 다운로드

Oracle Technology Network > Database > Database 12c

Database 12c
Database In-Memory
Multitenant
More Key Features
Application Development
Big Data Appliance
Cloud Database Services
Private Database Cloud
Data Warehousing & Big Data
Database Appliance
Exadata Database Machine
High Availability
Manageability
Migrations
Security
Unstructured Data
Upgrades
Windows
Database Technology Index

Oracle Database 11g Release 2 JDBC Drivers

You must accept the [OTN License Agreement](#) to download this software.

☒ Accept License Agreement | ☐ Decline License Agreement

Oracle Database 11g Release 2 (11.2.0.4) JDBC Drivers

SimpleFAN

↓ [simplefan.jar](#) (20,365 bytes) - (SHA1 Checksum: 307a7e203d7e141964158d181ca849d512d7e710)

Classes for subscribing to RAC events via ONS; simplefan policy and javadoc

JDBC Thin for All Platforms

↓ [ojdbc.policy](#) (10,591 bytes) - Sample security policy file for Oracle Database JDBC drivers

↓ [JavaDoc](#) (6,415,512 bytes)

↓ [README](#)

↓ [ojdbc6.jar](#) (2,739,670 bytes) - (SHA1 Checksum: a483a046eee2f404d864a6ff5b09dc0e1be3fe6c)

Certified with JDK 8, JDK 7 and JDK 6: It contains the JDBC driver classes except classes for NLS support in Oracle Object and Collection types.

↓ [ojdbc6_g.jar](#) (4,494,956 bytes) - (SHA1 Checksum: bf50af31967911af63058a6e1e5249c2dae34823)

Same as ojdbc6.jar except compiled with "javac -g" and contains tracing code.

↓ [ojdbc6dms.jar](#) (3,350,769 bytes) - (SHA1 Checksum: d268a890a9a681cf498a9fe9c47e92cad6ac26f0)

Same as ojdbc6.jar, except that it contains instrumentation to support DMS and limited java.util.logging calls.



Critical Capabilities for
Operational Database
Management Systems

[Read Gartner's
Report >](#)

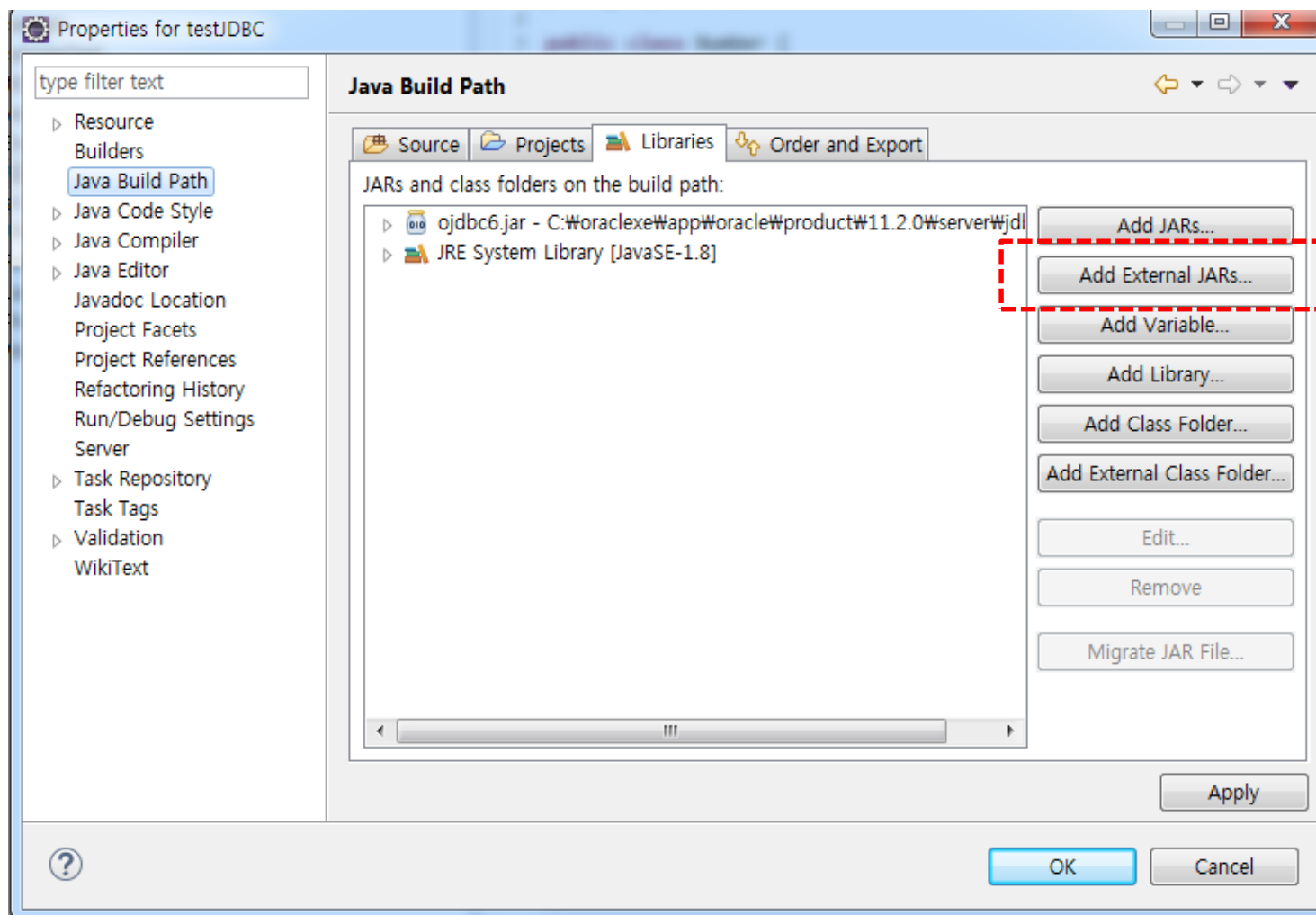


Oracle Live SQL

Learn and Share SQL

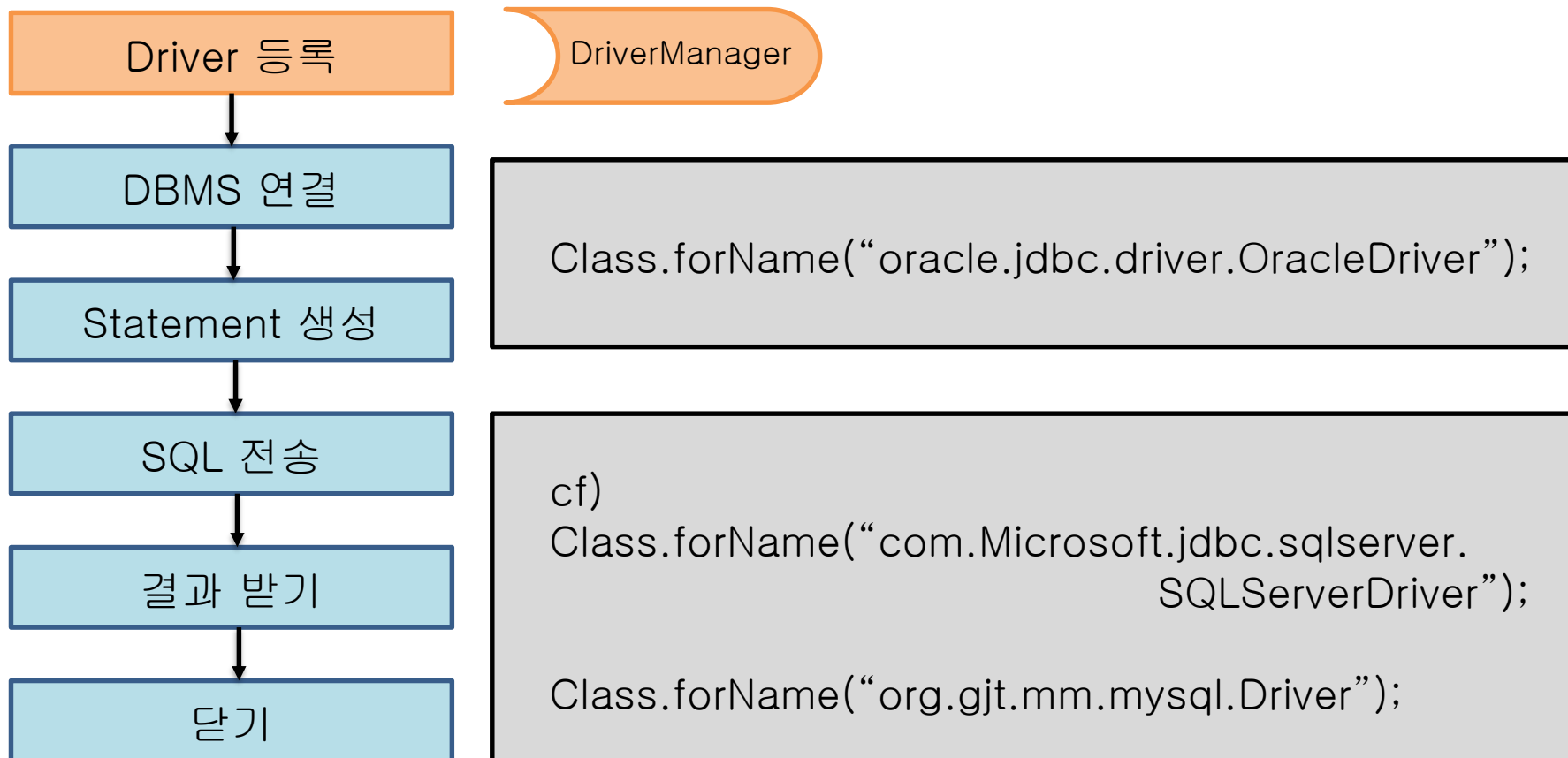
[Try Now >](#)

Library 등록



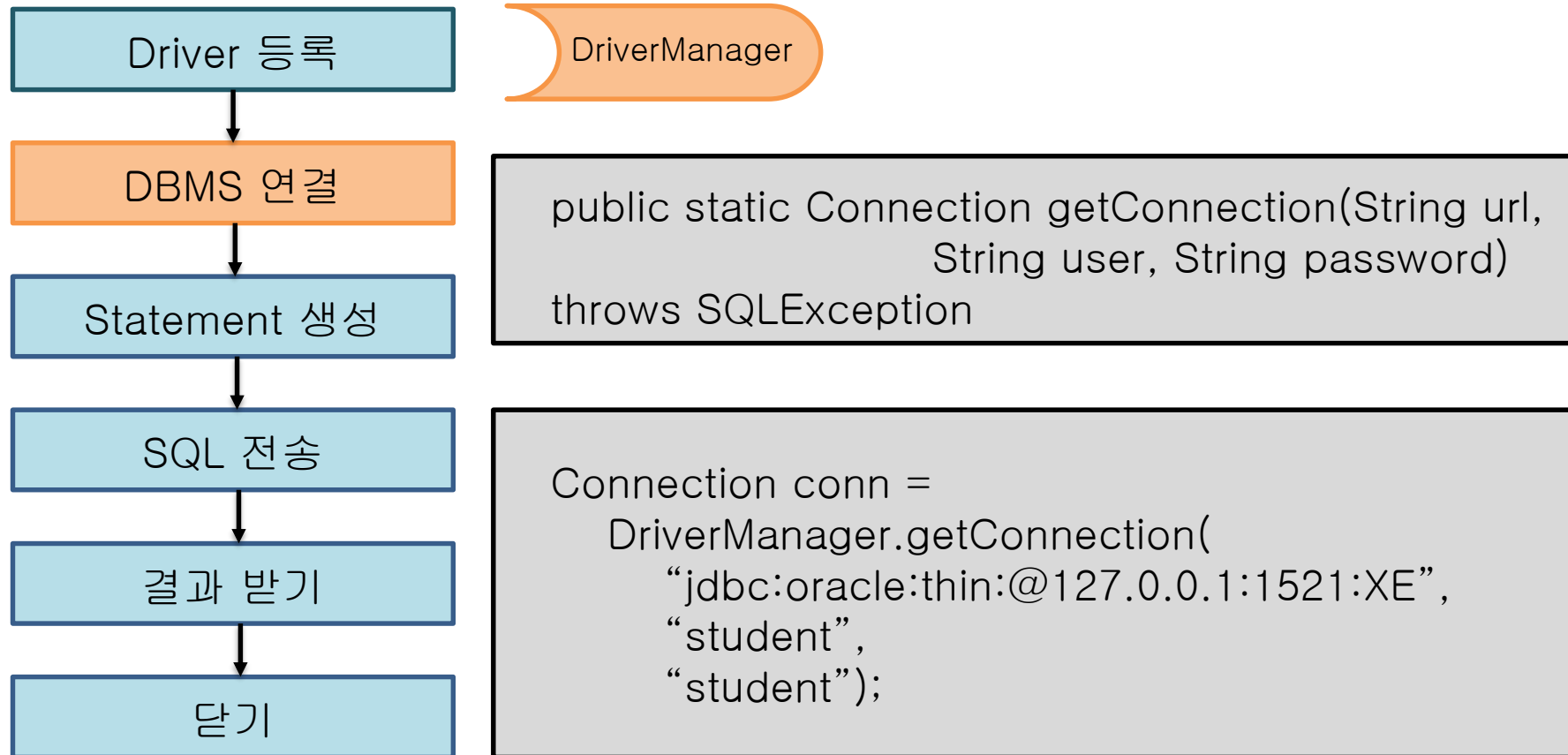
JDBC Coding 절차

1. DriverManager에 해당 DBMS Driver 등록



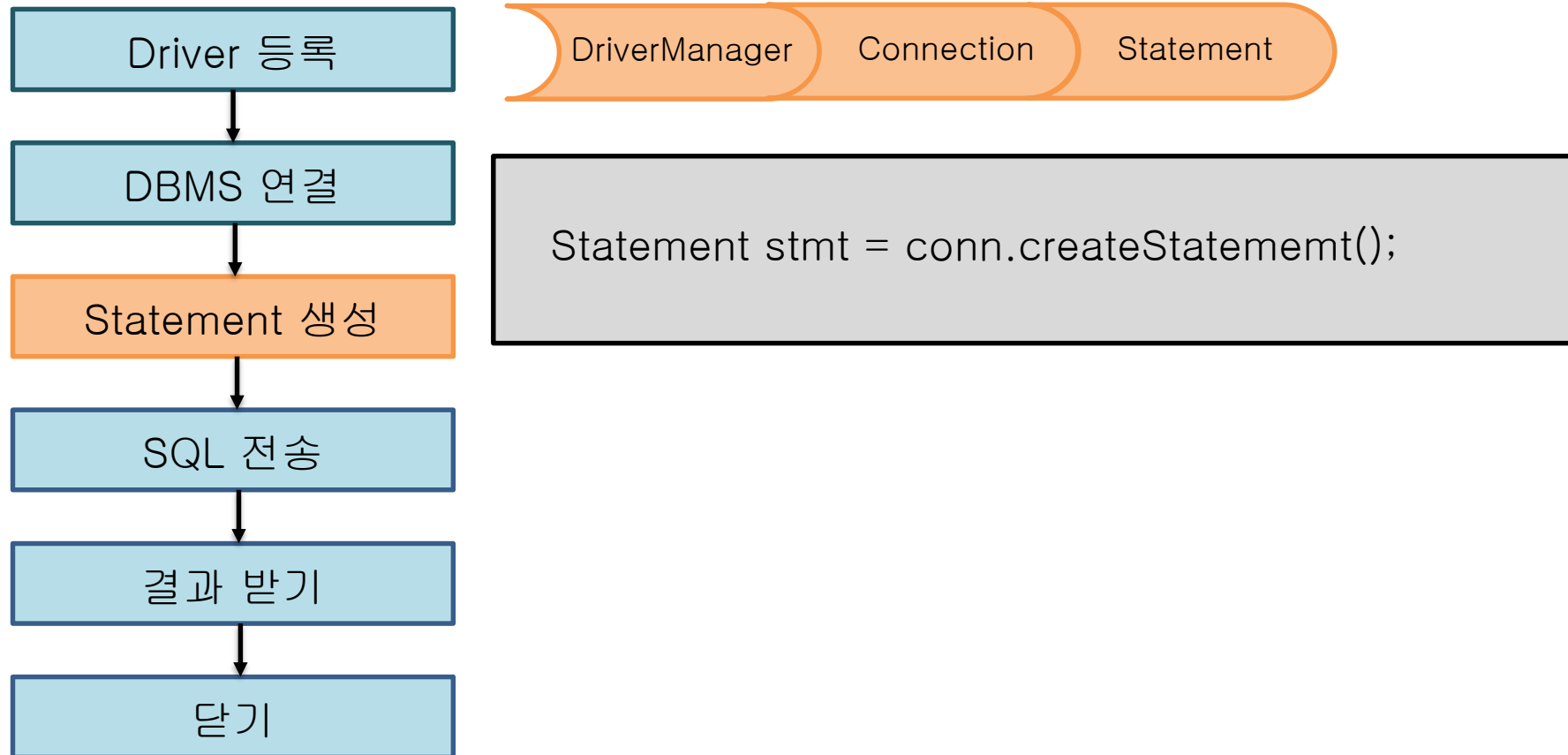
JDBC Coding 절차

2. 해당 Driver로부터 Connection instance 획득



JDBC Coding 절차

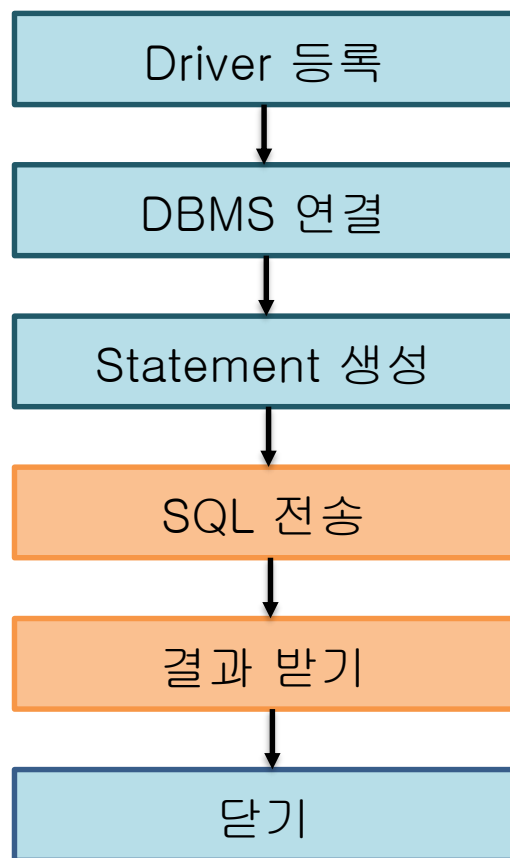
3. Connection instance로부터 Statement instance 획득



JDBC Coding 절차

4. Statement method를 이용하여 SQL문 실행

5. 실행 후 결과를 ResultSet(Selete)혹은 int형 변수(DML)로 받아서 처리



DriverManager

Connection

Statement

ResultSet

```
String query = "SELECT ID, LAST_NAME FROM EMP";  
ResultSet rset = stmt.executeQuery(query);
```

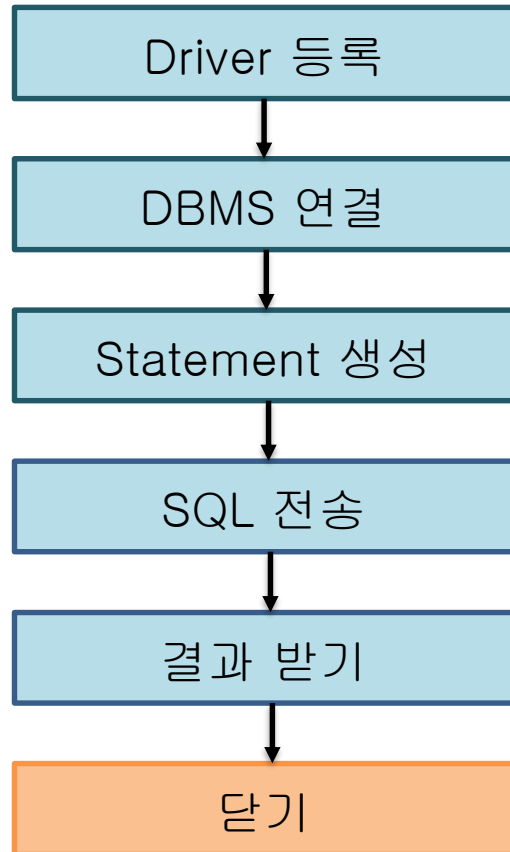
```
while(rset.next()){  
    System.out.println(rset.getString("ID") + "Wt"  
        + rset.getString(2));  
}
```

```
String query = "UPDATE EMP"  
    + " SET LAST_NAME = 'KIM'"  
    + " WHERE ID = '10000'";
```

```
int result = stmt.executeUpdate(query);
```

JDBC Coding 절차

6. 사용한 자원 반납



```
rset.close(); //ResultSet 사용한 경우 반납처리  
stmt.close();  
conn.close();
```

DriverManager

데이터 원본에 JDBC 드라이버를 통하여 커넥션은 만드는 역할을 한다.
Class.forName()메소드를 통해 생성되며, 반드시 예외처리를 해야 한다.
직접 객체 생성이 불가능하고, getConnection()메소드를 사용하여 객체를 생성할 수 있다.

Connection

Connection객체는 특정 데이터 원본과 연결된 커넥션을 나타낸다.
Statement객체를 생성할 때도 Connection객체를 사용하여
createStatement()메소드를 호출하여 생성한다.
SQL문장을 실행시키기 전에 우선 Connection객체가 있어야 한다.

Statement

Connection객체에 의해 프로그램에 리턴되는 객체에 의해 구현되는 일종의 메소드 집합을 정의한다. Connection클래스의 createStatement()메소드를 호출하여 얻어지며, 생성된 Statement객체로 질의문장을 String객체에 담아 인자로 전달하여 executeQuery()메소드를 호출하여 SQL 질의를 수행한다.

```
예) try{  
    String query = "SELECT ID, LAST_NAME FROM EMP"  
    stmt = conn.createStatement();  
    rset = stmt.executeQuery(query);  
  
}catch(SQLException e){  
    e.printStackTrace();  
}
```

PreparedStatement

Connection 객체의 `prepareStatement()` 메소드를 사용하여 객체를 생성한다. SQL문장이 미리 컴파일되고, 실행시간동안 인수 값을 위한 공간을 확보할 수 있다는 점에서 Statement와 다르다.

각각의 인수에 대해 위치홀더(?)를 사용하여 SQL문장을 정의할 수 있게 해준다.

```
예) try{  
    String query = "insert into member value (?, ?);  
    pstmt = conn.prepareStatement(query);  
    pstmt.setString(1, id);  
    pstmt.setString(2, password);  
  
    }catch(SQLException e){  
    e.printStackTrace();  
    }  
}
```

ResultSet

SELECT문을 사용한 질의 성공시 ResultSet을 반환한다. ResultSet은 SQL질의에 의해 생성된 테이블을 담고 있으며 '커서(cursor)라는 것을 가지고 특정 행에 대한 참조를 조작한다.

```
String id = rset.getString("ID");  
String lastName = rset.getString(2);
```

