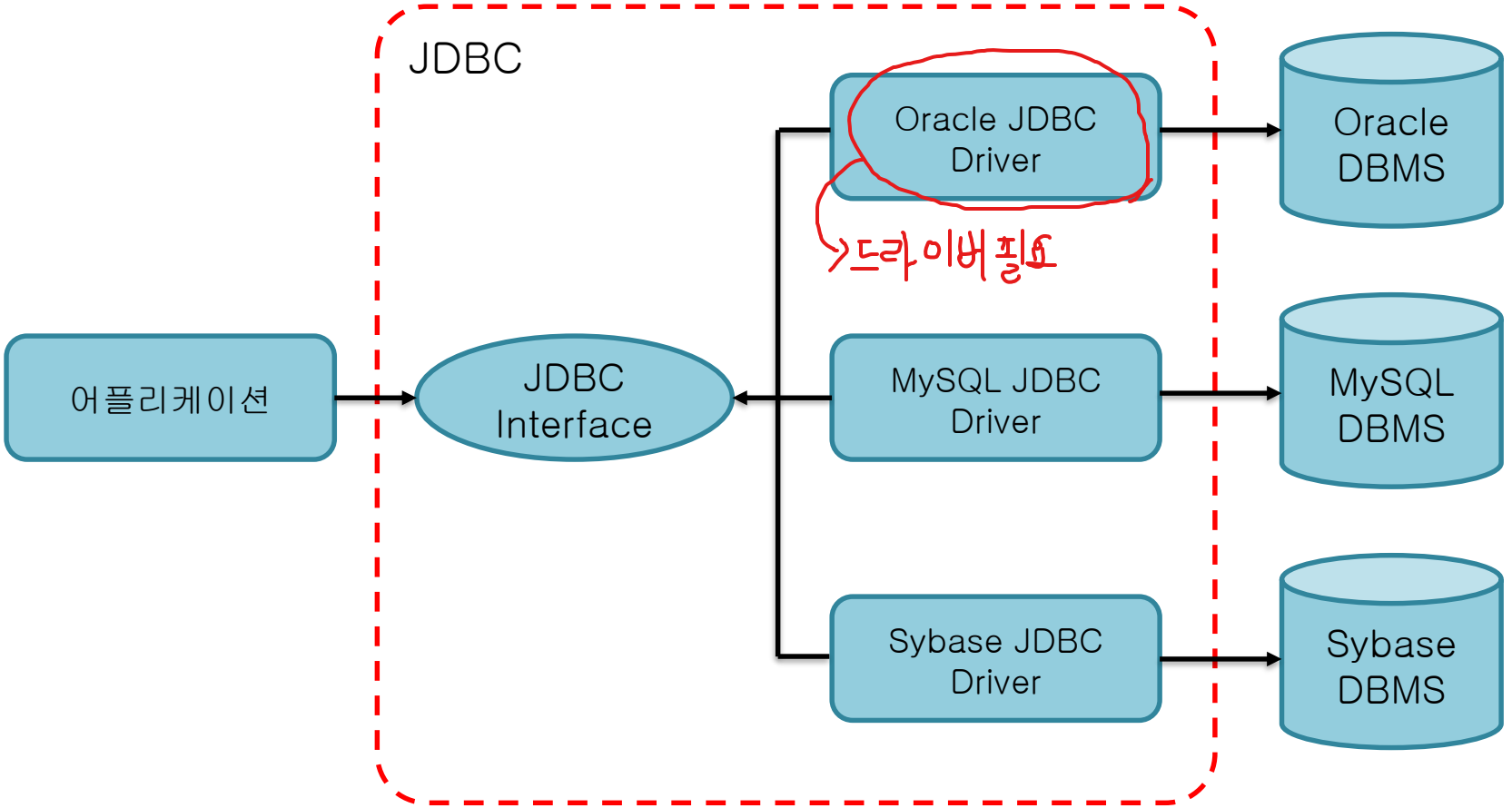
KH정보교육원

JDBC

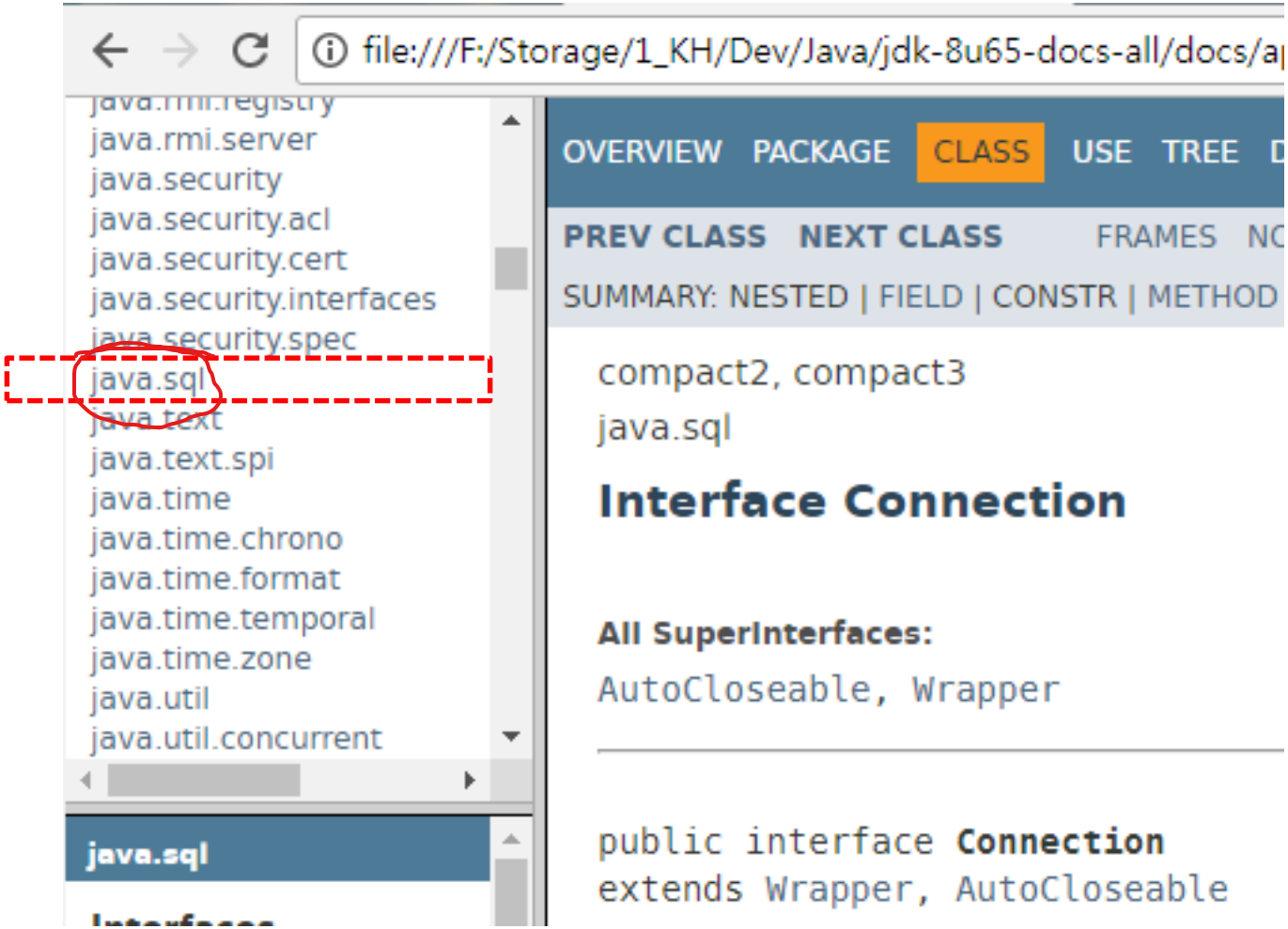
KH정보교육원

JDBC란?

자바 언어에서 Database에 접근할 수 있게 해 주는 Programming API이다.

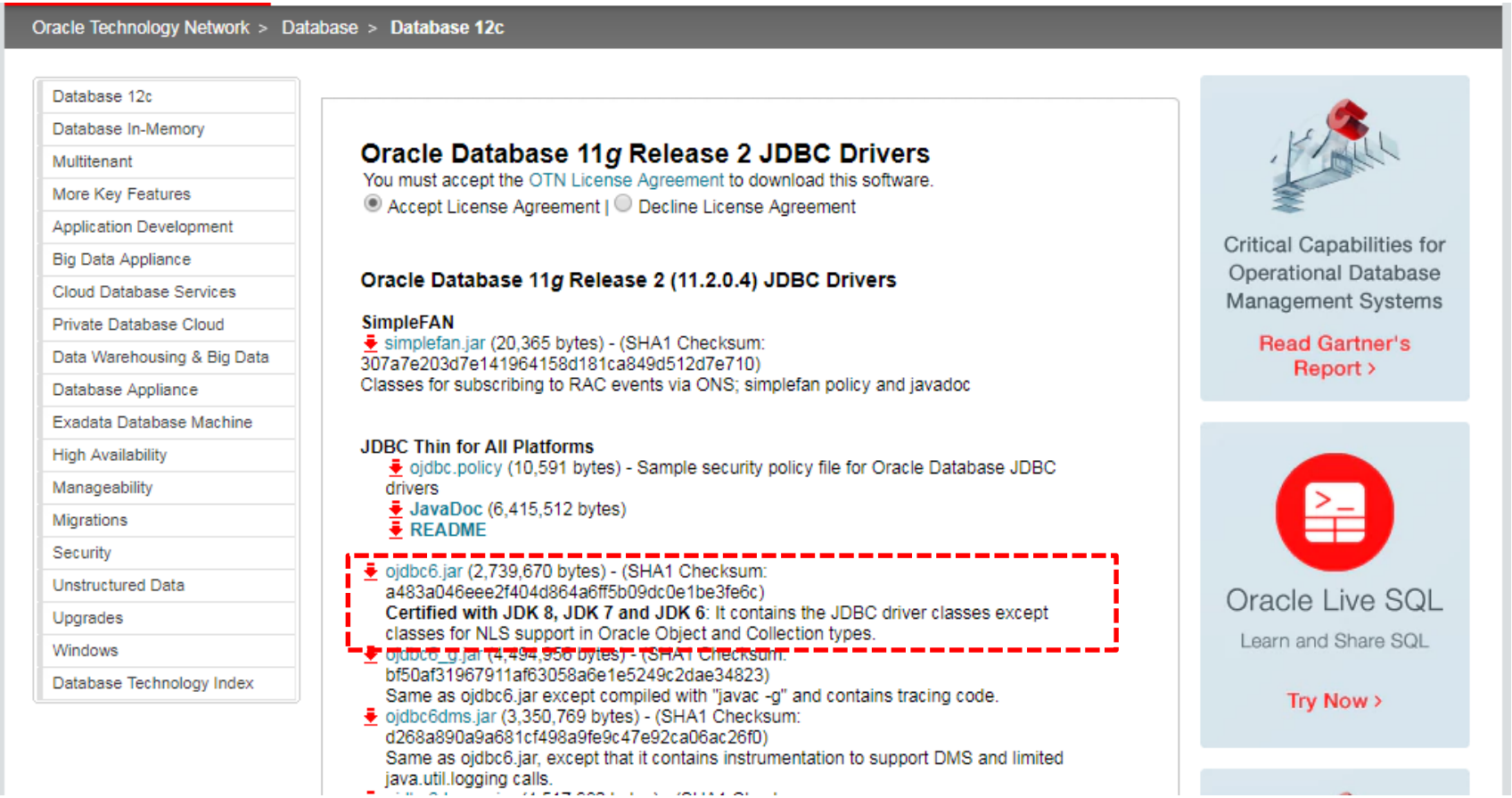


java.sql 패키지

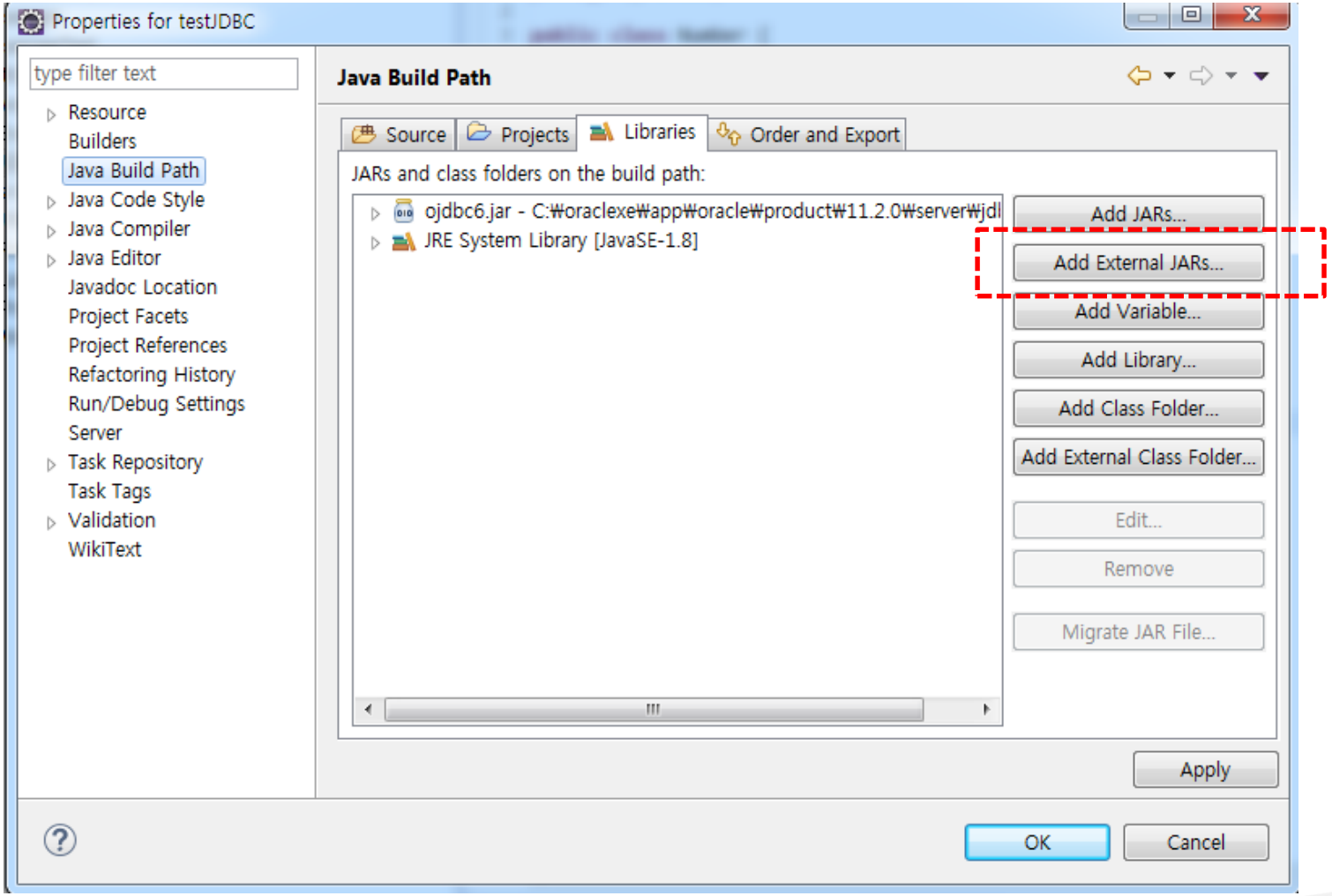

file:///F:/Storage/1_KH/Dev/Java/jdk-8u65-docs-all/docs/a
java.rmi.server
java.security
java.security.acl
java.security.cert
java.security.interfaces
java.security.spec
java.sql
java.text
java.text.spi
java.time
java.time.chrono
java.time.format
java.time.temporal
java.time.zone
java.util
java.util.concurrent
java.sql
OVERVIEW PACKAGE CLASS USE TREE
PREV CLASS NEXT CLASS FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD
compact2, compact3
java.sql
Interface Connection
All SuperInterfaces:
AutoCloseable, Wrapper
public interface Connection
extends Wrapper, AutoCloseable

KH정보교육원

OJDBC 다운로드



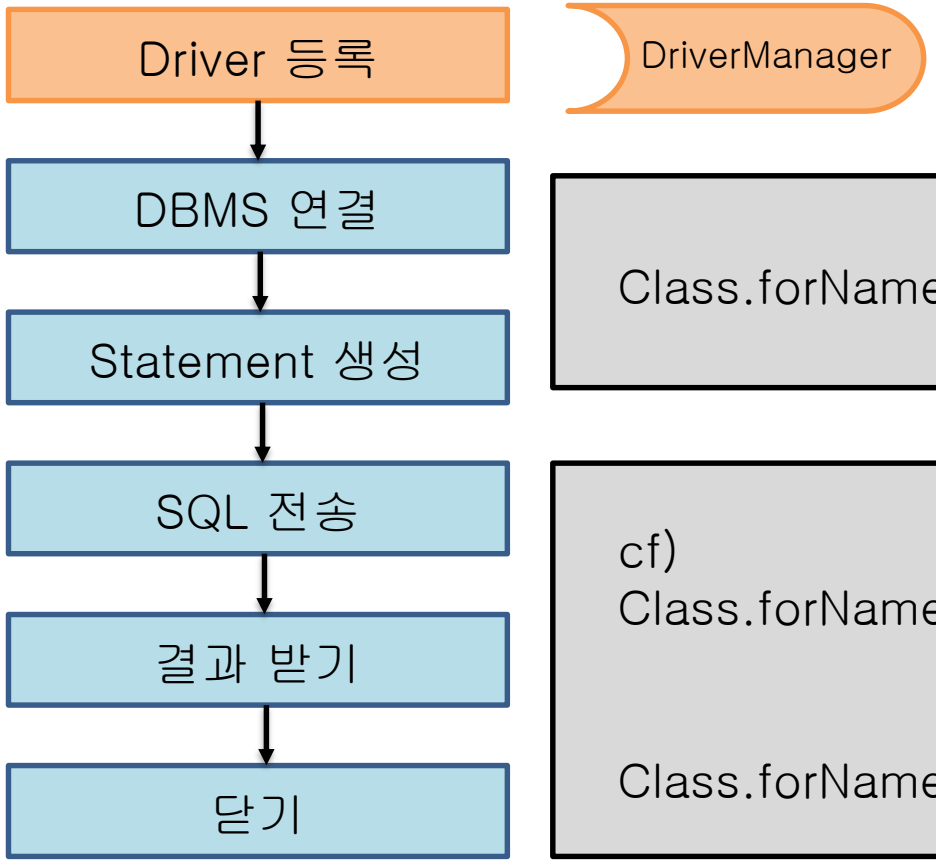
Library 등록


Properties for testJDBC
type filter text
Resource
Builders
Java Build Path
Java Code Style
Java Compiler
Java Editor
Javadoc Location
Project Facets
Project References
Refactoring History
Run/Debug Settings
Server
Task Repository
Task Tags
Validation
WikiText
Java Build Path
Source
Projects
Libraries
Order and Export
JARs and class folders on the build path:
ojdbc6.jar - C:₩oraclexe₩app₩oracle₩product₩11.2.0₩server₩jdl
JRE System Library [JavaSE-1.8]
Add JARs...
Add External JARs...
Add Variable...
Add Library...
Add Class Folder...
Add External Class Folder...
Edit...
Remove
Migrate JAR File...
Apply
OK
Cancel

KH정보교육원

JDBC Coding 절차

1. DriverManager에 해당 DBMS Driver 등록



DriverManager

```
Class.forName("oracle.jdbc.driver.OracleDriver");
```

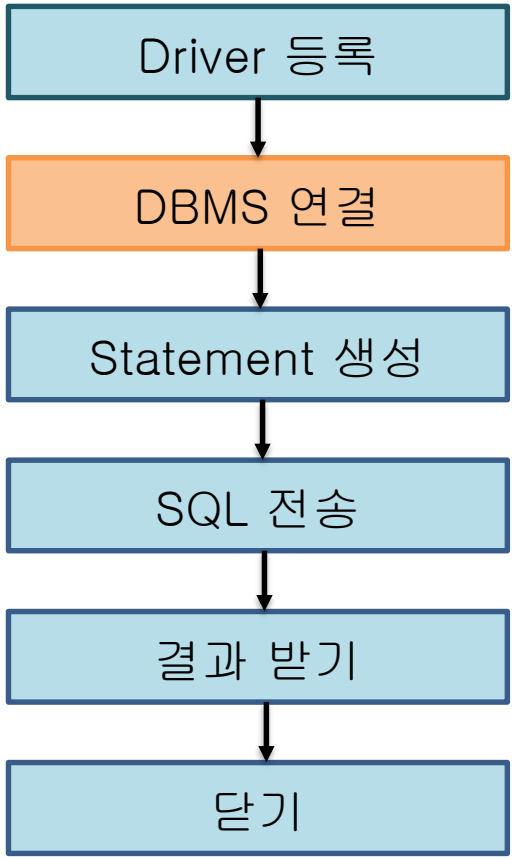
cf)

```
Class.forName("com.Microsoft.jdbc.sqlserver.
                                    SQLServerDriver");

Class.forName("org.gjt.mm.mysql.Driver");
```

JDBC Coding 절차

2. 해당 Driver로부터 Connection instance 획득



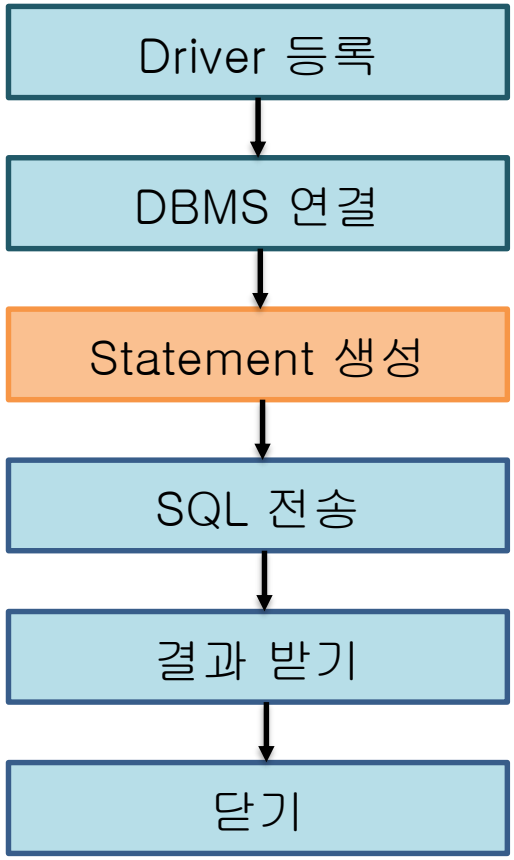
DriverManager

```
public static Connection getConnection(String url,
                                       String user, String password)
throws SQLException
```

```
Connection conn =
    DriverManager.getConnection(
        “jdbc:oracle:thin:@127.0.0.1:1521:XE”,
        “student”,
        “student”);
```

JDBC Coding 절차

3. Connection instance로부터 Statement instance 획득



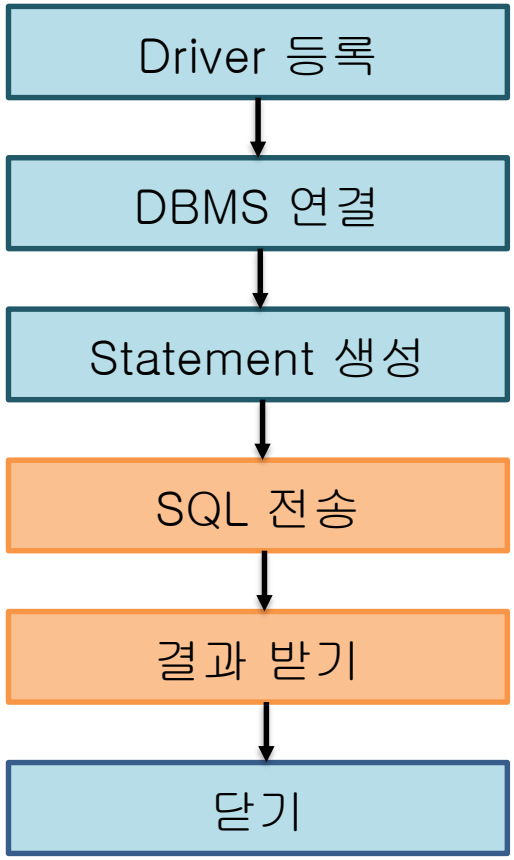
DriverManager → Connection → Statement

```
Statement stmt = conn.createStatememt();
```

KH정보교육원

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") + "₩t"
                        + rset.getString(2));
}
```

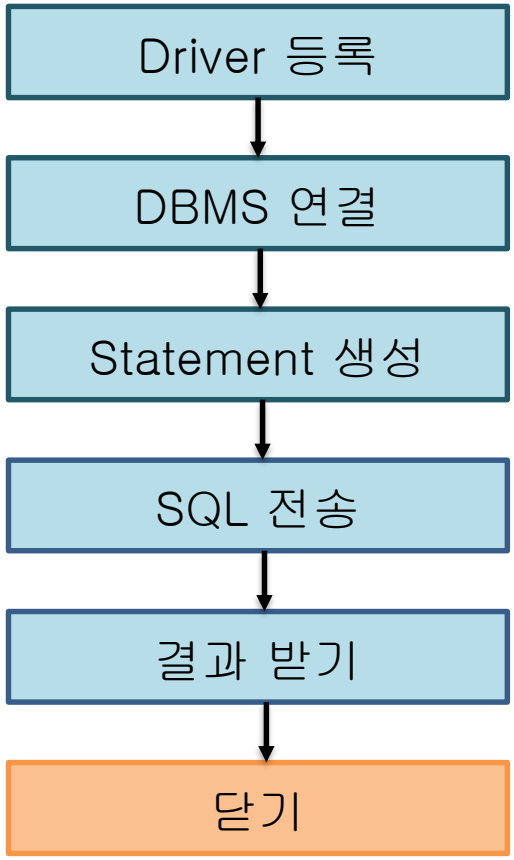
```
String query = "UPDATE EMP"
                + " SET LAST_NAME = 'KIM'"
                + " WHERE ID = '10000'";

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

KH정보교육원

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 excuteQuery()메소드를 호출하여 SQL 질의를 수행한다.

예)
```
try{
        String query = "SELECT ID, LAST_NAME FROM EMP"
        stmt = conn.createStatement();
        rset = stmt.excuteQuery(query);

    }catch(SQLException e){
        e.printStackTrace();
    }
```

PreparedStatement

Connection 객체의 preparedStatement()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.preparedStatement(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.