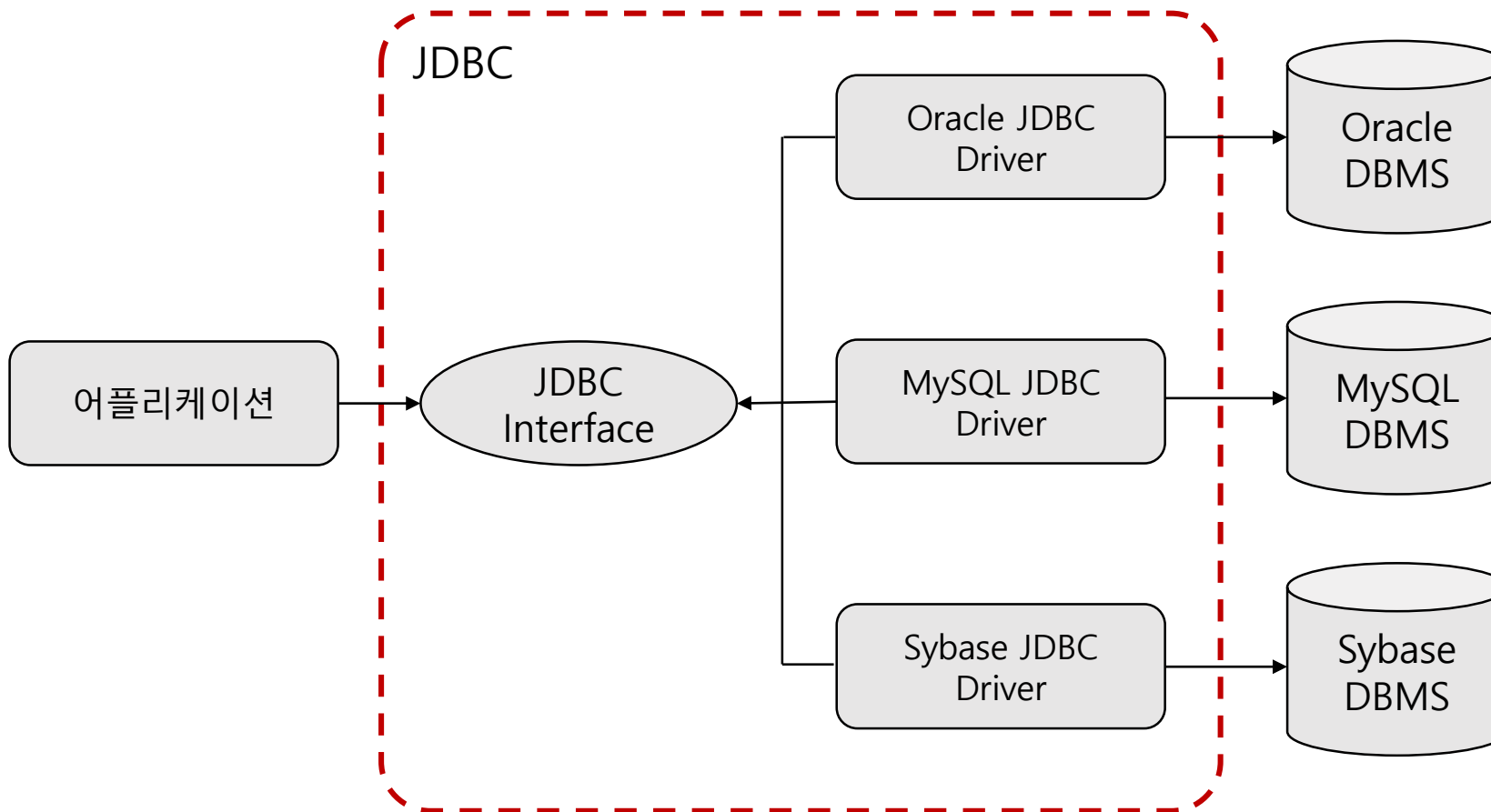




JDBC

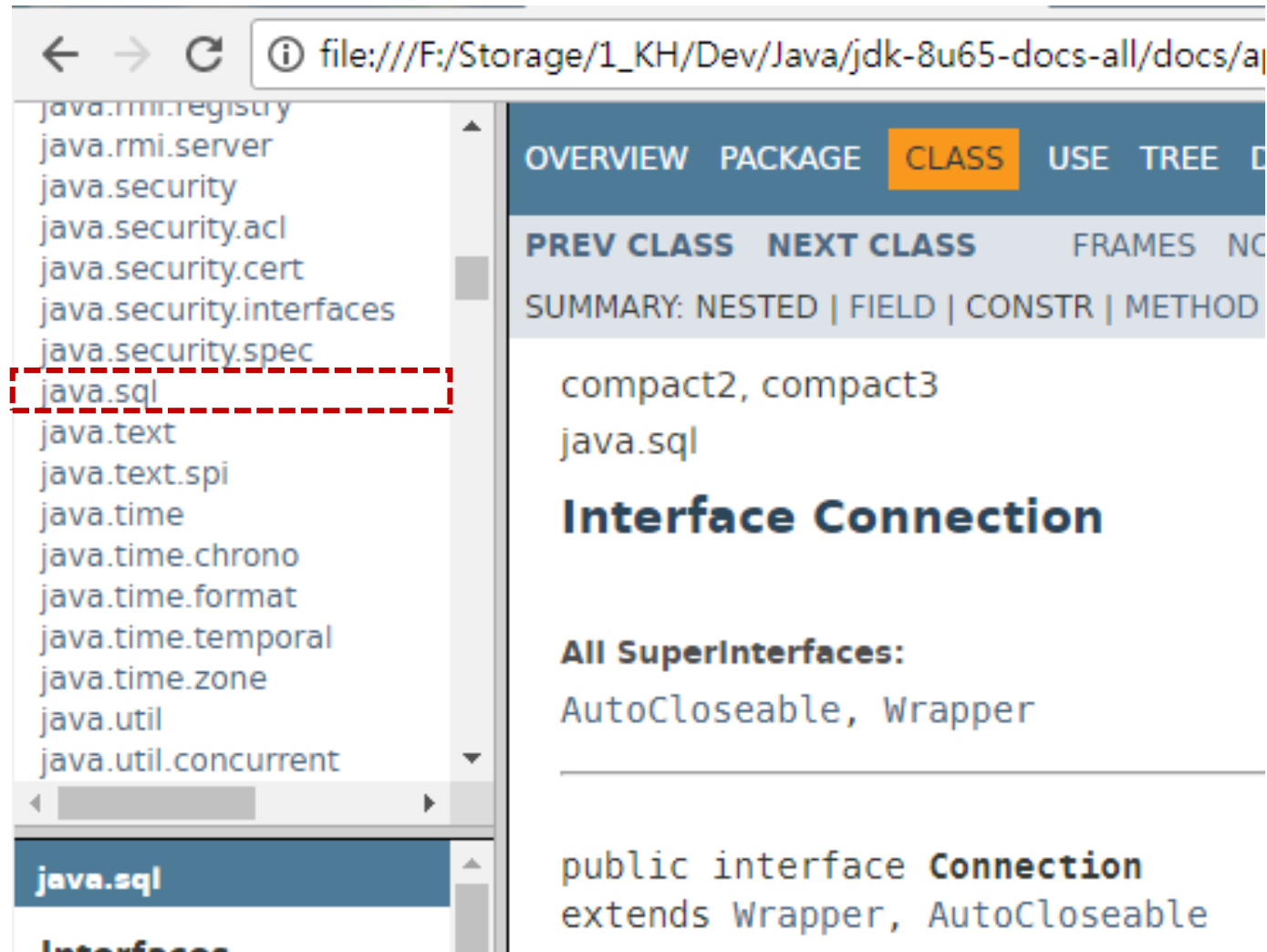
▶ JDBC(Java DataBase Connectivity)

자바에서 데이터베이스에 접근할 수 있게 해주는 Programming API



▶ JDBC(Java DataBase Connectivity)

✓ java.sql 패키지



The screenshot shows the Java API documentation for the `java.sql` package. The left sidebar lists various Java packages, with `java.sql` highlighted by a red dashed box. The main content area displays the package overview for `java.sql`, including navigation tabs (OVERVIEW, PACKAGE, CLASS, USE, TREE, D), navigation links (PREV CLASS, NEXT CLASS, FRAMES, NC), and a summary of nested classes, fields, constructors, and methods. The `Interface Connection` section is visible, showing that it extends `Wrapper` and `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 D

PREV CLASS NEXT CLASS FRAMES NC

SUMMARY: NESTED | FIELD | CONSTR | METHOD

compact2, compact3
java.sql

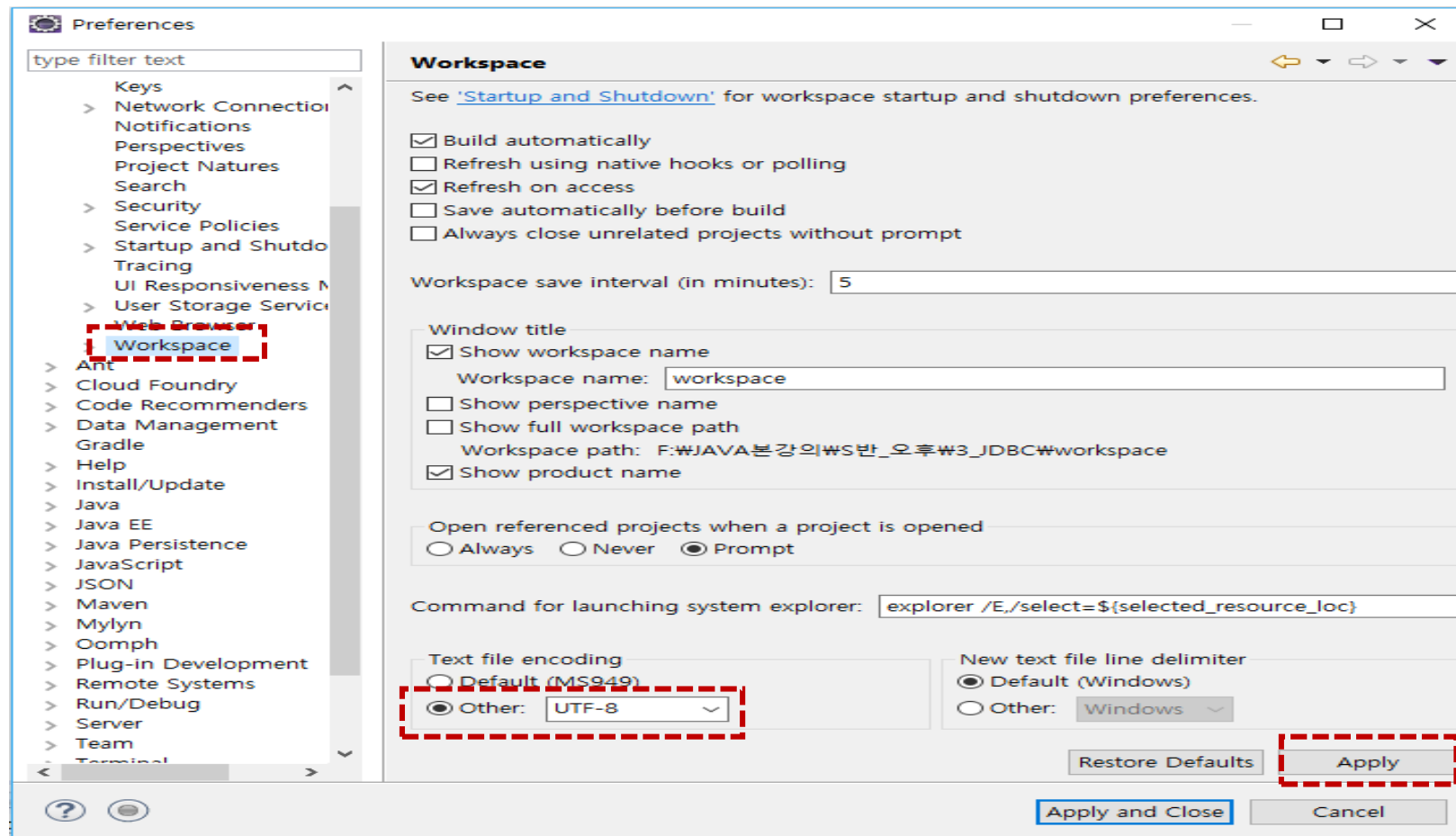
Interface Connection

All SuperInterfaces:
AutoCloseable, Wrapper

public interface **Connection**
extends Wrapper, AutoCloseable

▶ Character Set 설정

✓ 기본 문자 인코딩



* General – Workspace – Text file encoding – other에서 UTF-8 선택

▶ JDBC 사용 객체

✓ DriverManager

데이터 원본에 JDBC드라이버를 통하여 커넥션을 만드는 역할

Class.forName() 메소드를 통해 생성되며 반드시 예외처리를 해야 함

직접 객체 생성이 불가능하고 getConnection() 메소드를 사용하여 객체 생성 가능

✓ Connection

특정 데이터 원본과 연결된 커넥션을 나타내며 Statement객체를 생성할 때도

Connection객체를 사용하여 createStatement() 메소드를 호출하여 생성

SQL문장을 실행시키기 전에 우선 Connection객체가 있어야 함

▶ JDBC 사용 객체

✓ Statement

Connection객체에 의해 프로그램에 리턴되는 객체에 의해 구현되는 일종의 메소드 집합 정의
Connection클래스의 `createStatement()` 메소드를 호출하여 얻어지며
생성된 Statement객체로 질의문장을 String객체에 담아 인자로 전달하여
`executeQuery()` 메소드를 호출하여 SQL질의 수행

✓ 예시

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

▶ JDBC 사용 객체

✓ PreparedStatement

Connection객체의 `prepareStatement()` 메소드를 사용하여 객체 생성
SQL문장이 미리 컴파일 되고 실행 시간동안 인수 값을 위한 공간을 확보한다는 점에서
Statement와 다름

✓ 예시

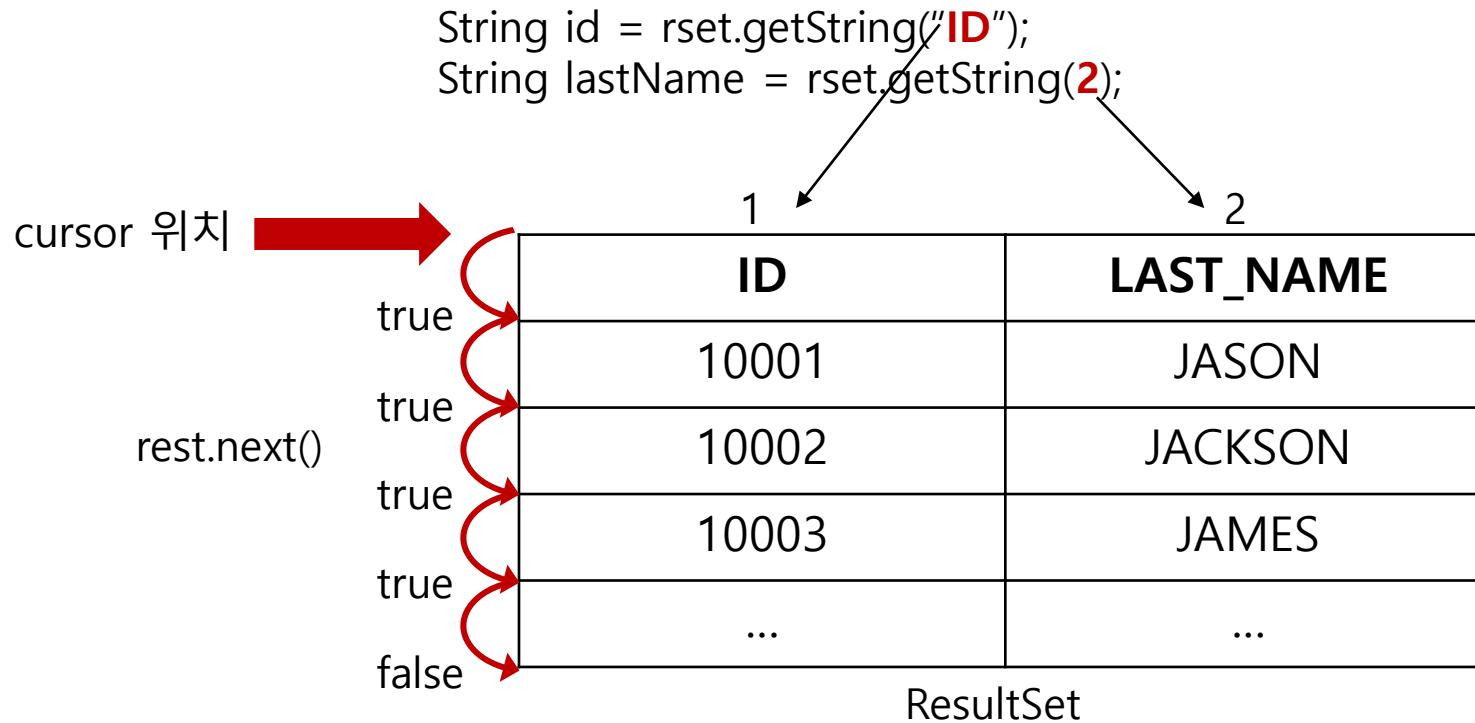
```
try{
    String query = "INSERT INTO MEMBER VALUES(?, ?)";
    pstmt = conn.prepareStatement(query);
    pstmt.setString(1, id);
    pstmt.setString(2, password);
} catch(SQLException e){
    e.printStackTrace();
}
```

▶ JDBC 사용 객체

✓ ResultSet

SELECT문을 사용한 질의 성공 시 Result Set 반환

SQL질의에 의해 생성된 테이블을 담고 있으며 커서(cursor)로 특정 행에 대한 참조 조작



▶ JDBC 코딩 절차

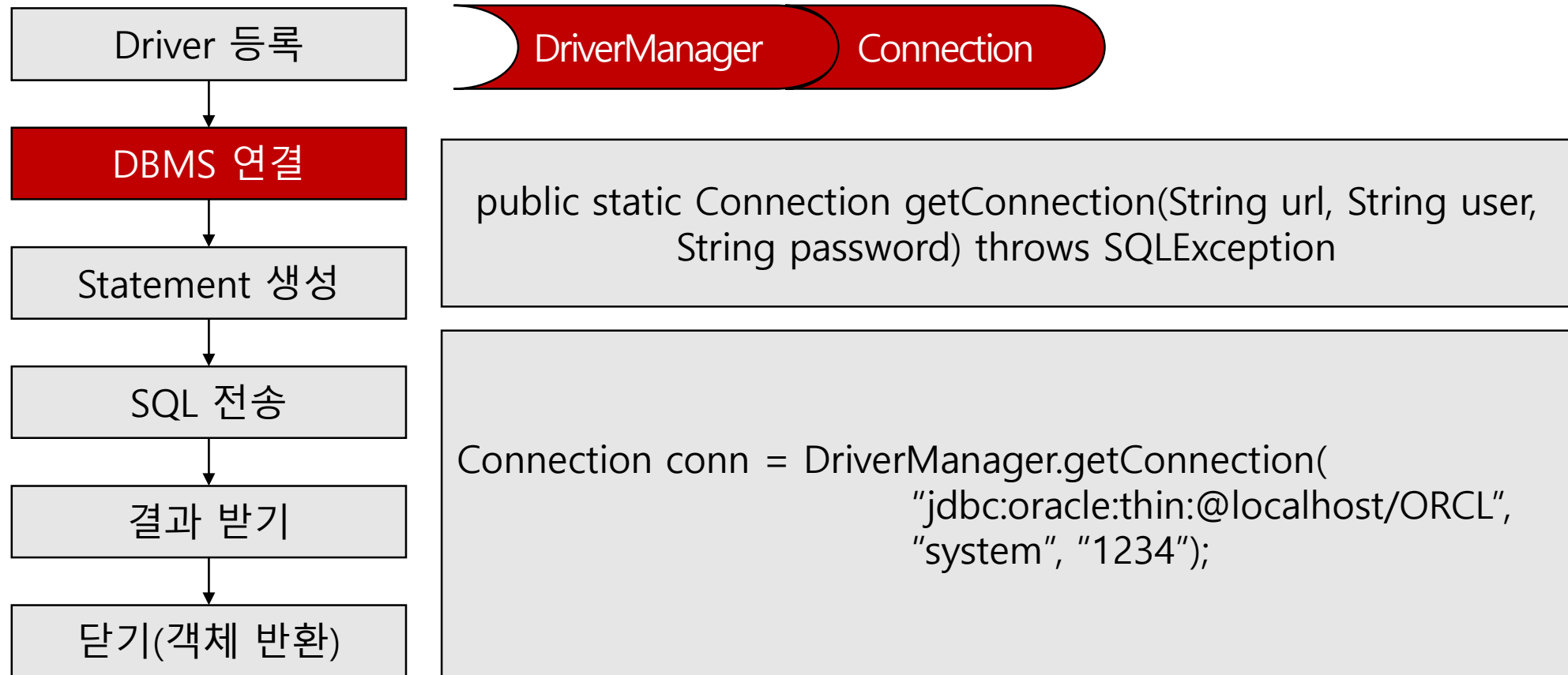
✓ DriverManager에 해당 DBMS Driver 등록



* 반드시 `ClassNotFoundException` 처리를 해야 함

▶ JDBC 코딩 절차

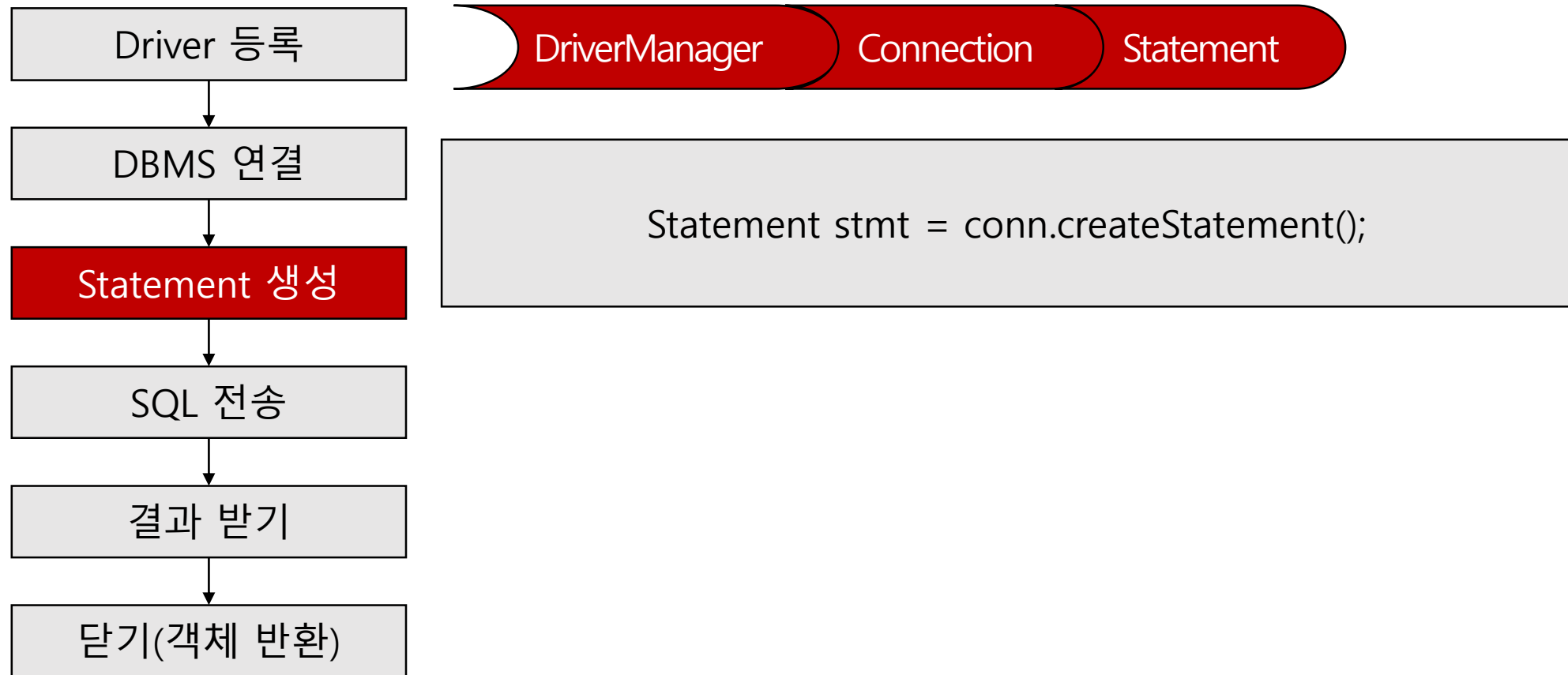
✓ 해당 Driver로부터 Connection instance 획득



* 반드시 SQLException 처리를 해야 함

▶ JDBC 코딩 절차

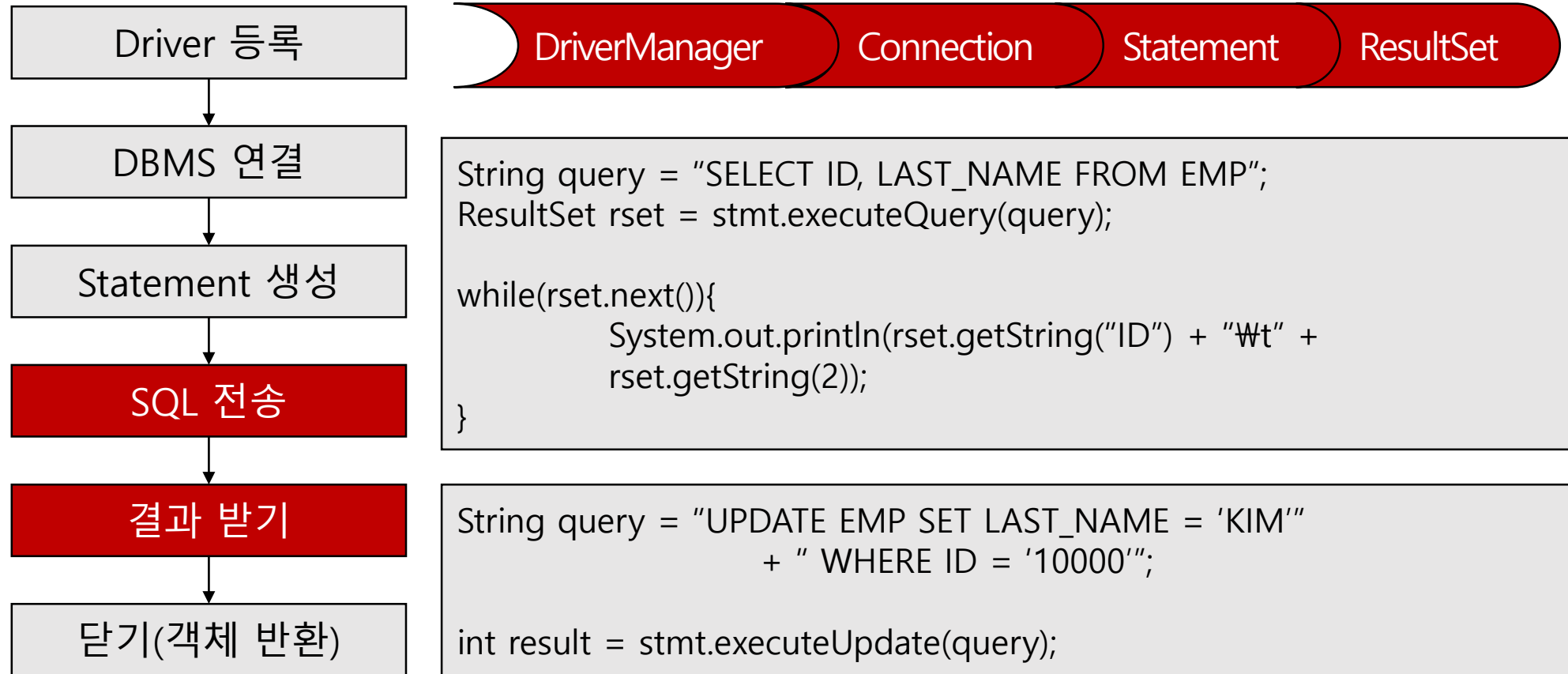
- ✓ Connection instance로부터 Statement instance 획득



* 반드시 SQLException 처리를 해야 함

▶ JDBC 코딩 절차

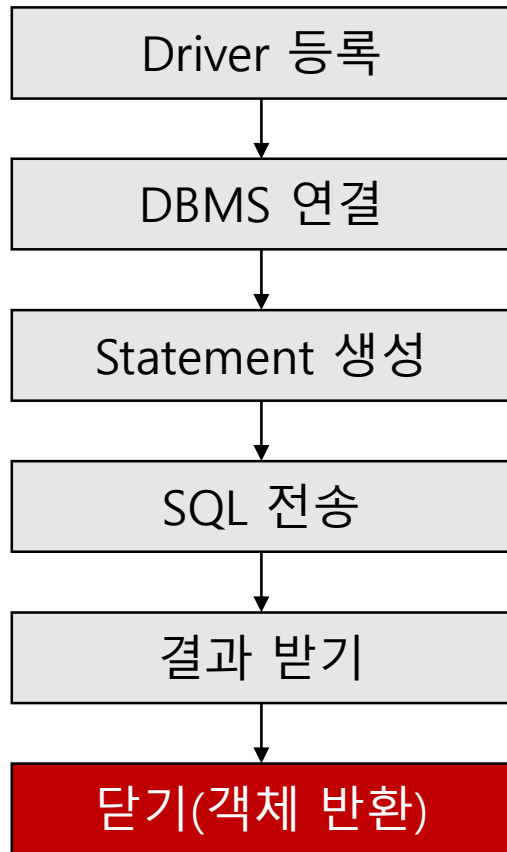
- ✓ Statement method를 이용하여 SQL문 실행
- ✓ 실행결과를 ResultSet(Select) 혹은 int형 변수(DML)로 받아서 처리



* 반드시 SQLException 처리를 해야 함

▶ JDBC 코딩 절차

- ✓ Connection instance로부터 Statement instance 획득



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

* 반드시 SQLException 처리를 해야 함