



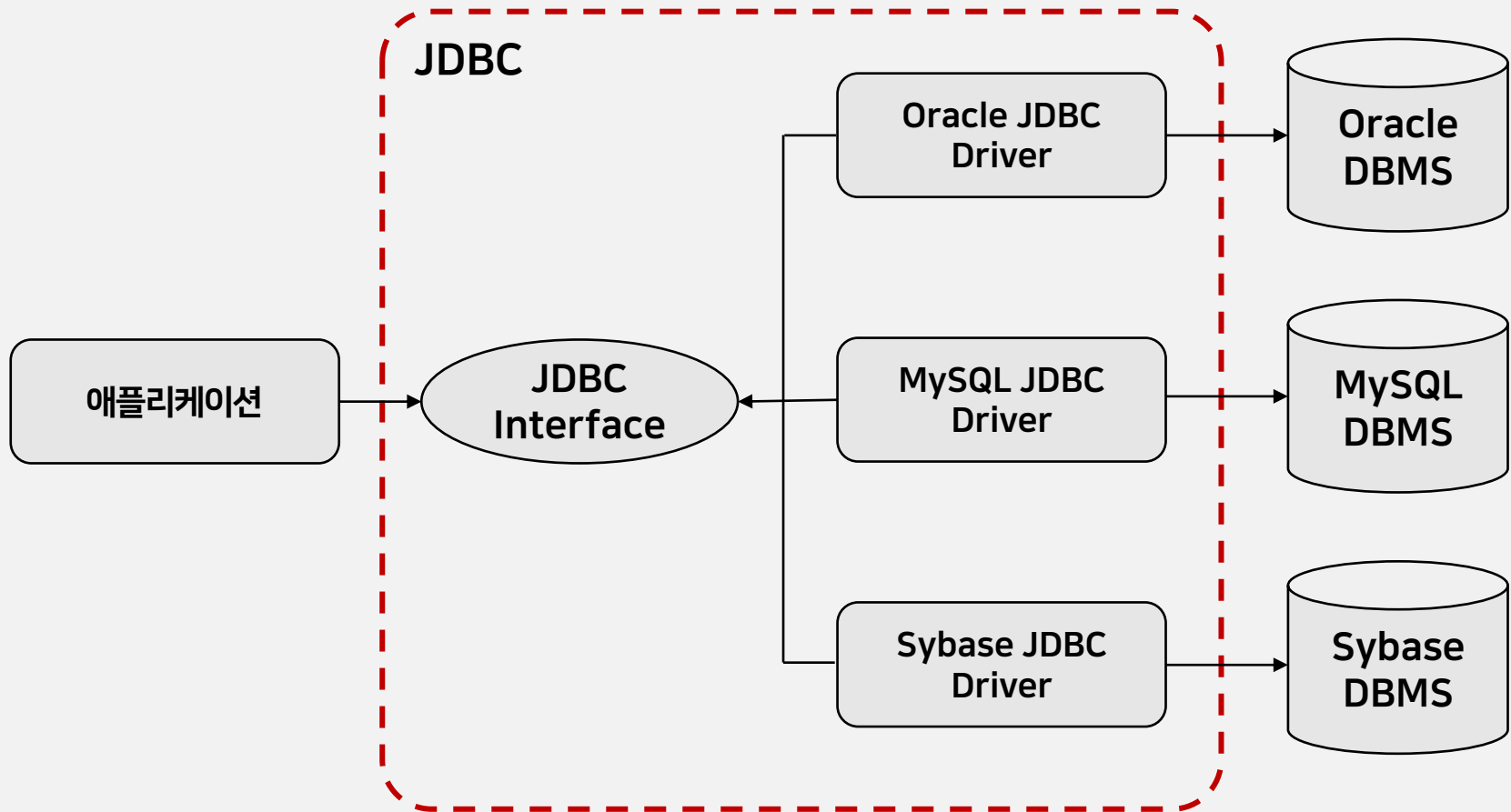
Today's lecture



```
System.out.print('JDBC');
```

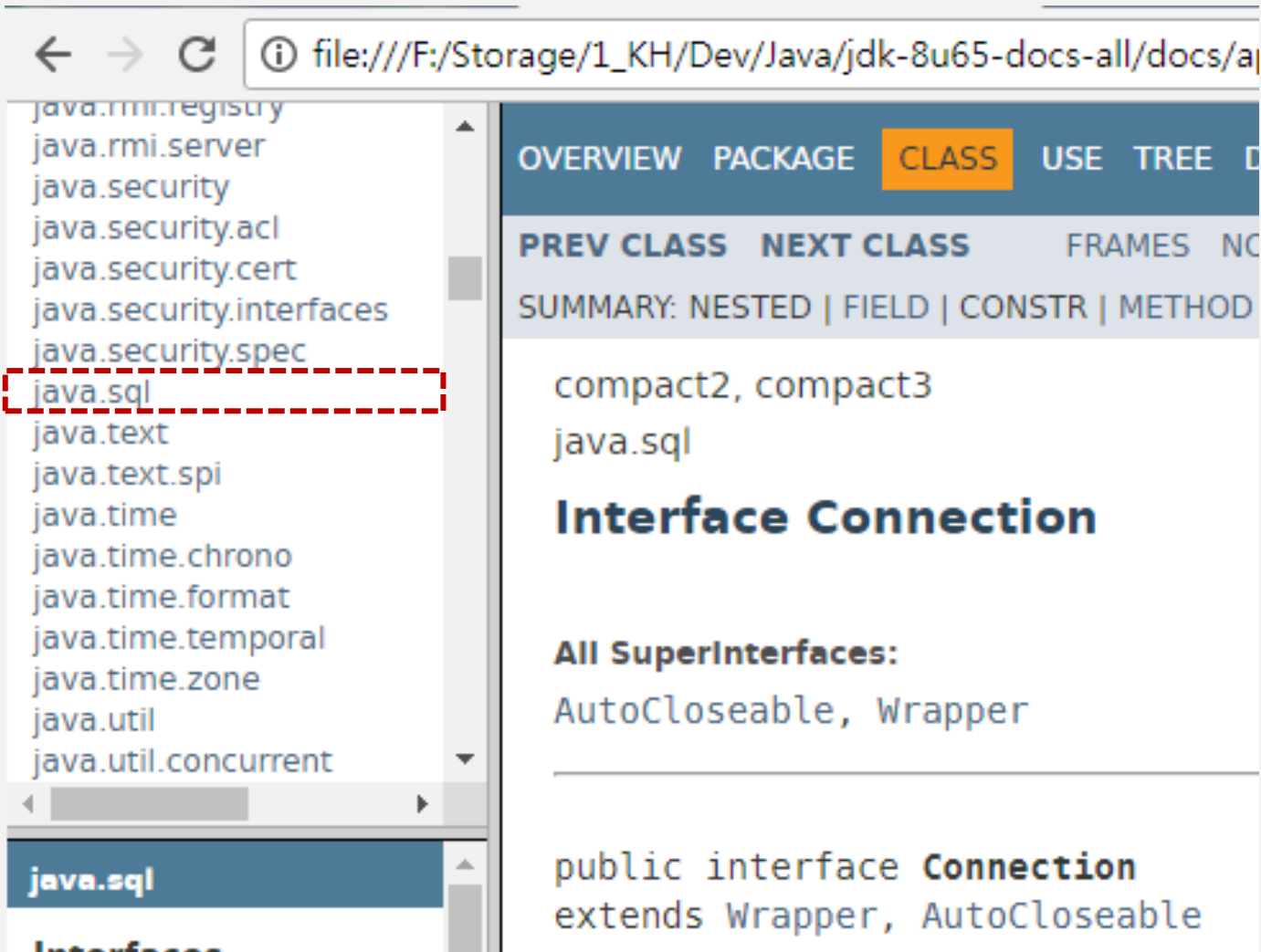
▶ JDBC(Java DataBase Connectivity)

자바에서 데이터베이스에 접근할 수 있게 해주는 Programming API



▶ JDBC(Java DataBase Connectivity) KH정보교육원

✓ 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 the title "Interface Connection" and the declaration "public interface Connection extends Wrapper, AutoCloseable".

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

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

▶ OJDBC 다운로드

✓ 홈페이지 다운로드

<https://www.oracle.com/technetwork/database/enterprise-edition/jdbc-112010-090769.html>

Oracle Technology Network / Database / Database Downloads

Database Downloads
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 A - Z

Oracle Database 11.2.0.4 JDBC Driver & UCP Downloads

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 & UCP Downloads

Zipped JDBC Driver and Companion JARs

📄 [ojdbc-full.tar.gz](#) (6,761,477 bytes) - (SHA1 Checksum:

1ce3d1055b94ee1c6148d74a440c937d0a2df30e)

The TAR archive contains the latest 11.2.0.4 JDBC Thin driver ([ojdbc6.jar](#) and [ojdbc5.jar](#)), Universal Connection Pool ([ucp.jar](#)), other companion jars, and README that has more information about the contents of the tar file.

OR

Unzipped JDBC Driver and Companion JARs

The JARs included in the [ojdbc-full.tar.gz](#) are also available as individual downloads in this section.

📄 [ojdbc6.jar](#) (2,739,670 bytes) - (SHA1 Checksum: a483a046eee2f404d864a6ff5b09dc0e1be3fe6c)
Certified with JDK8, JDK7, and JDK6:

▶ OJDBC 다운로드

✓ 오라클에서 복사

- Express 버전

C:\Woraclexe\Wapp\Woracle\Wproduct\W11.2.0\server
Wjdbc
Wlib\Wojdbc6

- Enterprise 버전

C:\Wapp\W사용자계정\Wproduct\W11.2.0\dbhome_1
Wjdbc\Wlib\Wojdbc6

c 드라이브에 dev 폴더 만들어서 붙여넣기!!!!

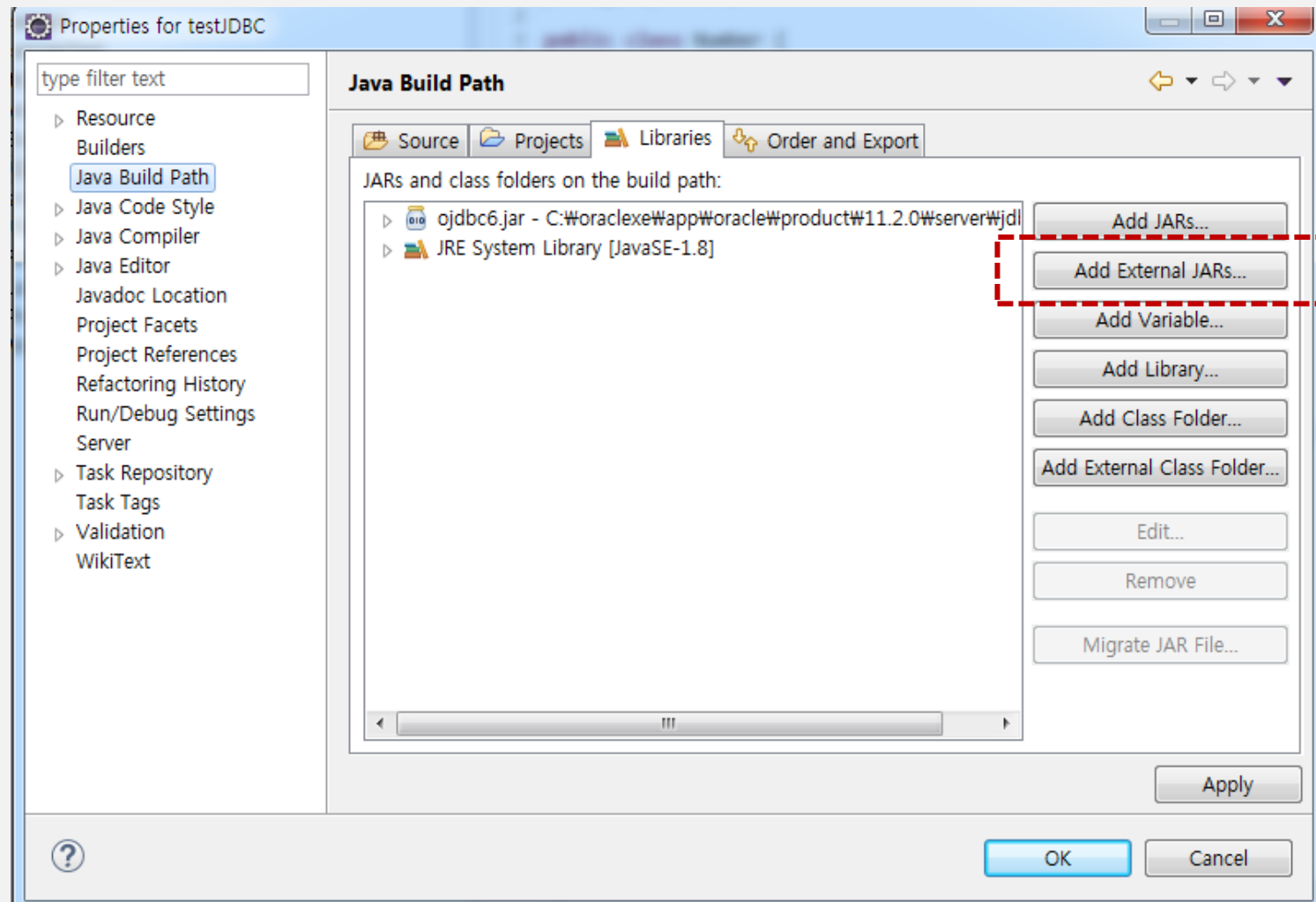
▶ JDBC 설치

다운 받거나 복사한 ojdbc파일을 dev폴더에 넣어주기

D:\dev\

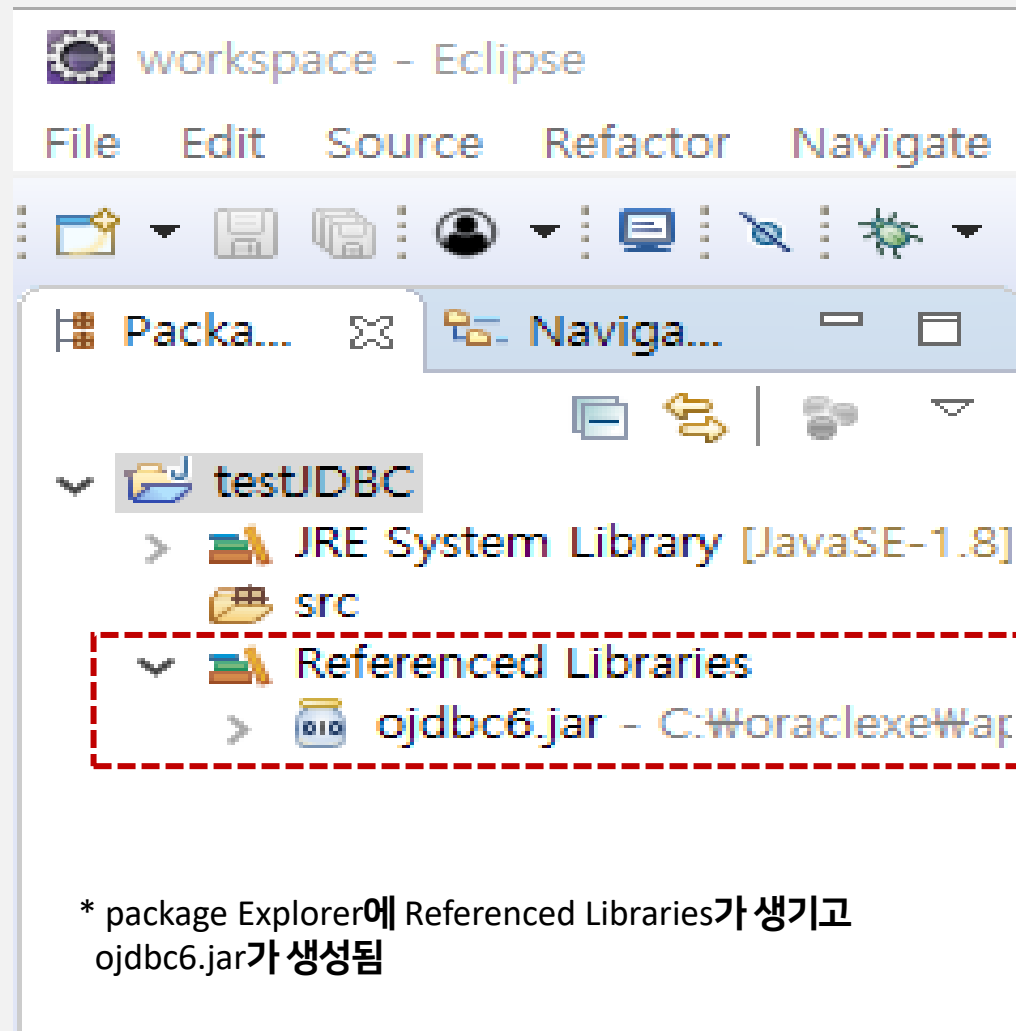
▶ Library 등록

✓ Properties로 등록



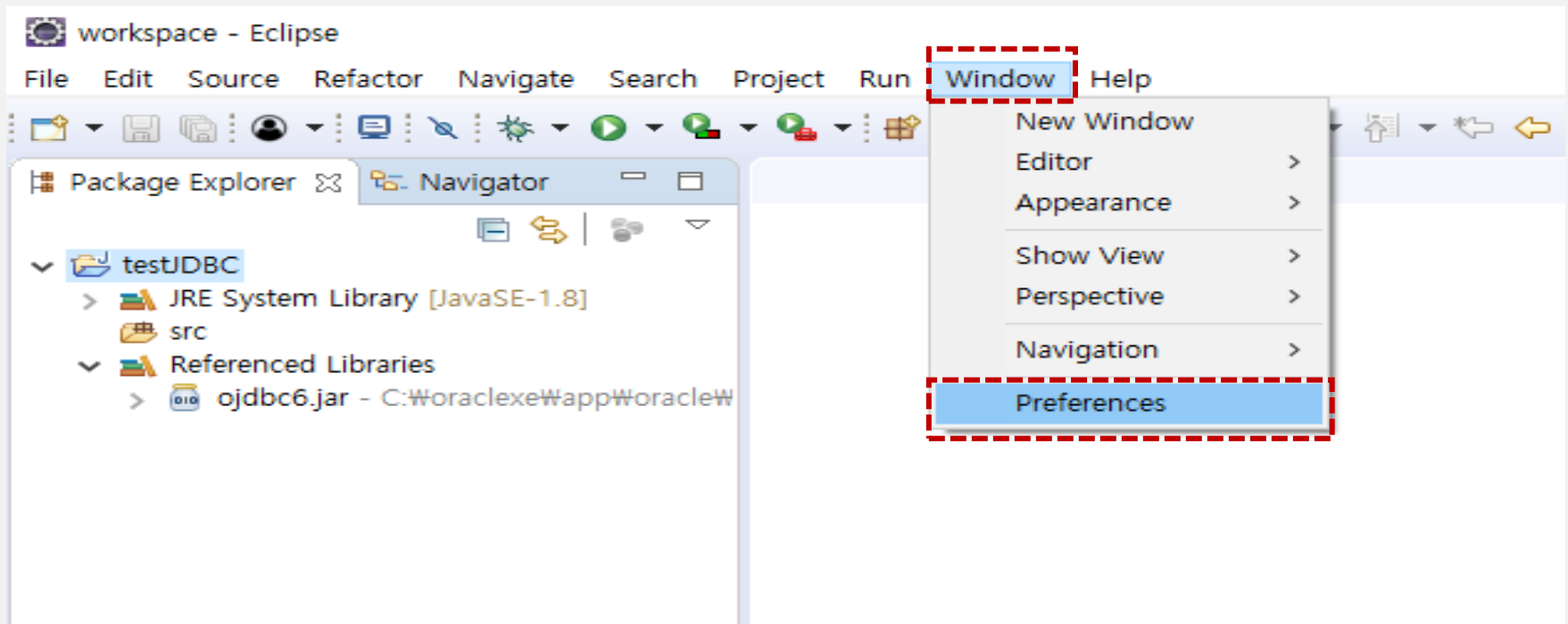
▶ Library 등록

✓ Eclipse Library로 등록



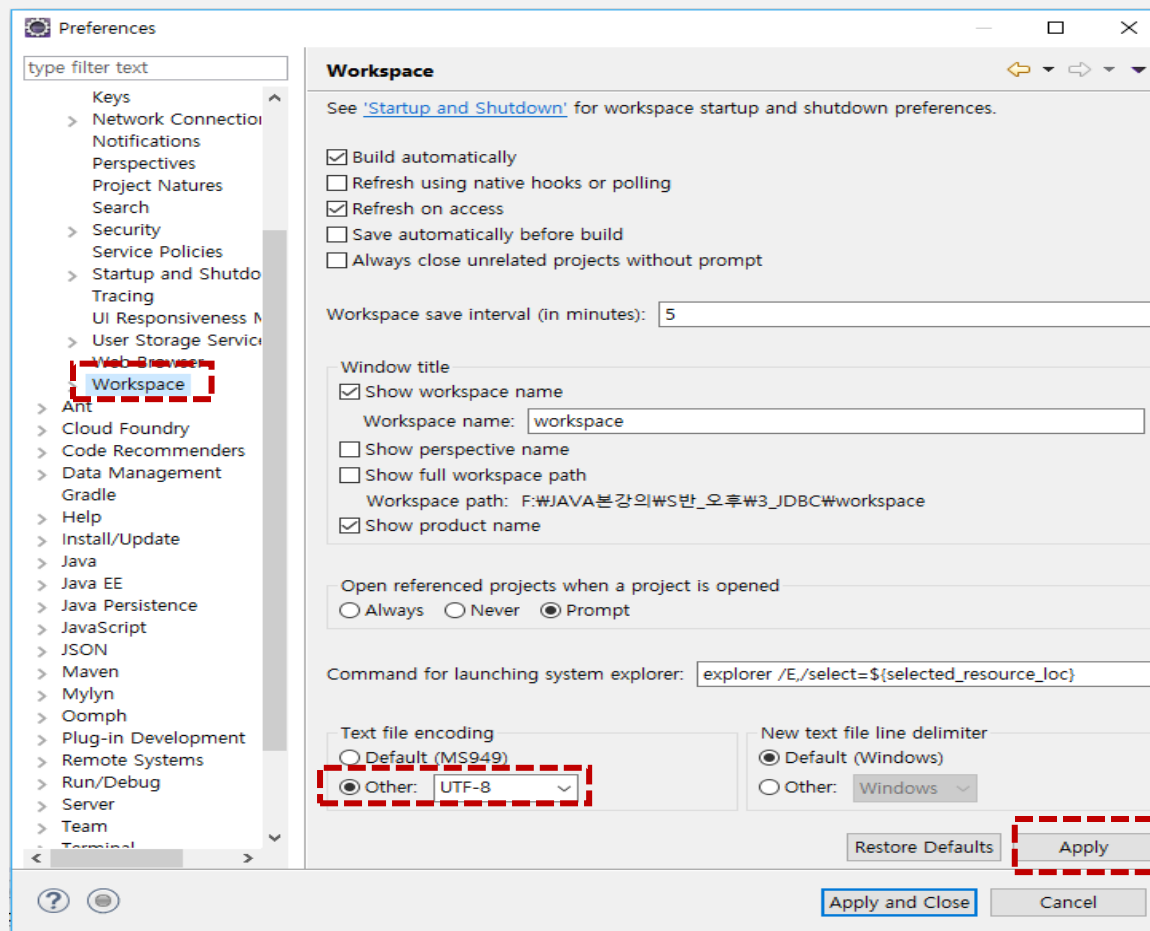
▶ Character Set 설정

문자 인코딩 방식이 맞지 않으면 해당 문자가 제대로 출력되지 않아
이클립스 작업 파일에 대한 문자셋을 UTF-8로 일치시켜야 함



▶ Character Set 설정

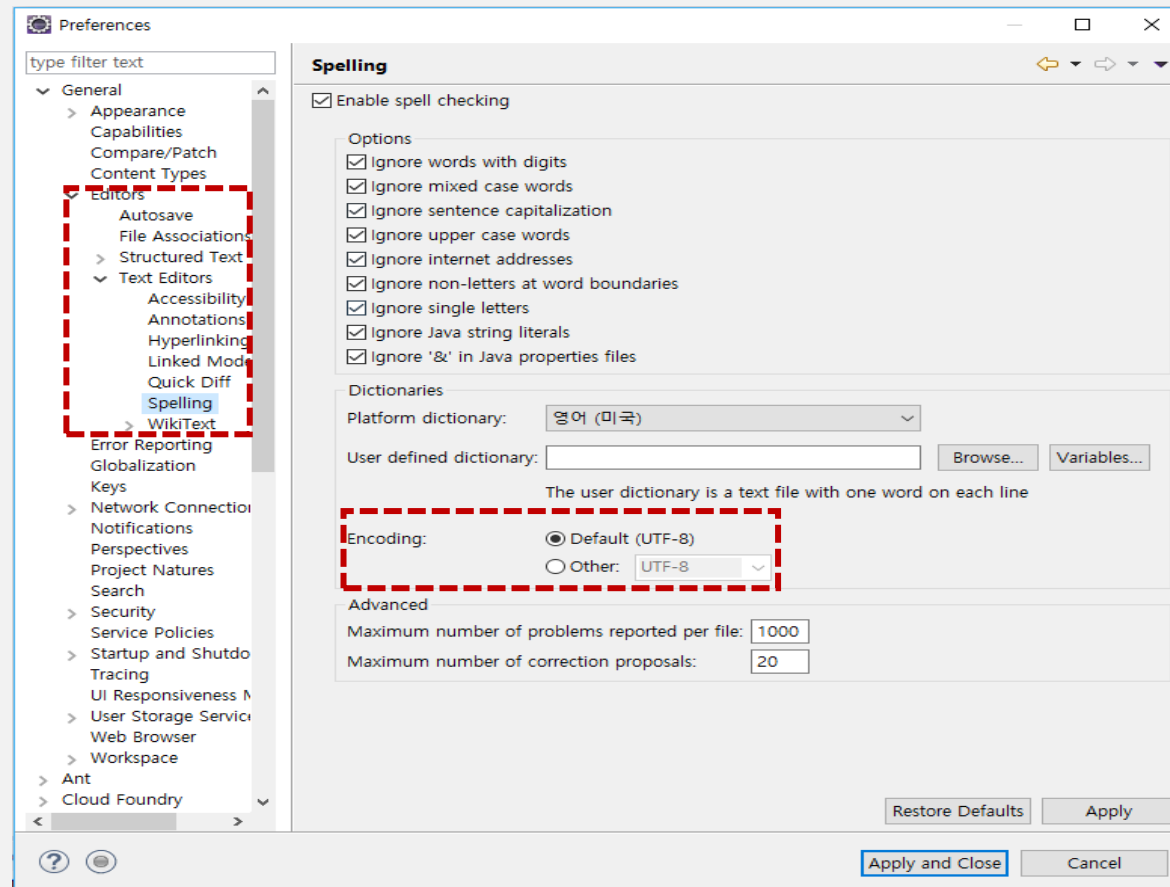
✓ 기본 문자 인코딩



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

▶ Character Set 설정

✓ 기본 문자 인코딩



* General – Editors – Text Editors – Spelling UTF-8 변경

▶ 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 호

출하여 생성

SQL문장을 실행시키기 전에 우선 Connection 객체가 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 SQL 질의 수행

▶ JDBC 사용 객체

✓ 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 시간 동안 인수 값을 위한 공간을
확보한다는 점에서 Statement와 다름
각 인수에 대해 위치홀더(?)를 사용해 SQL문장을 정의할 수 있게 함

▶ JDBC 사용 객체

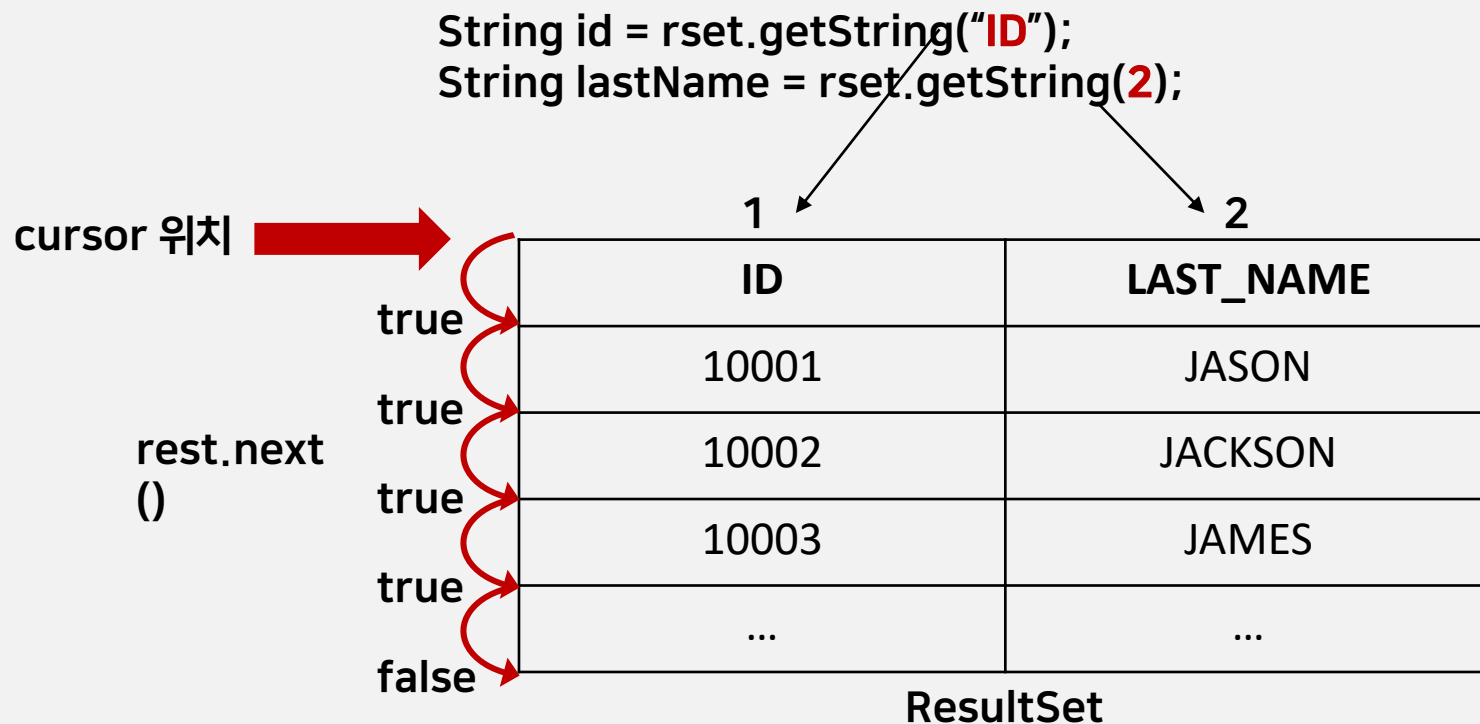
✓ 예시

```
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

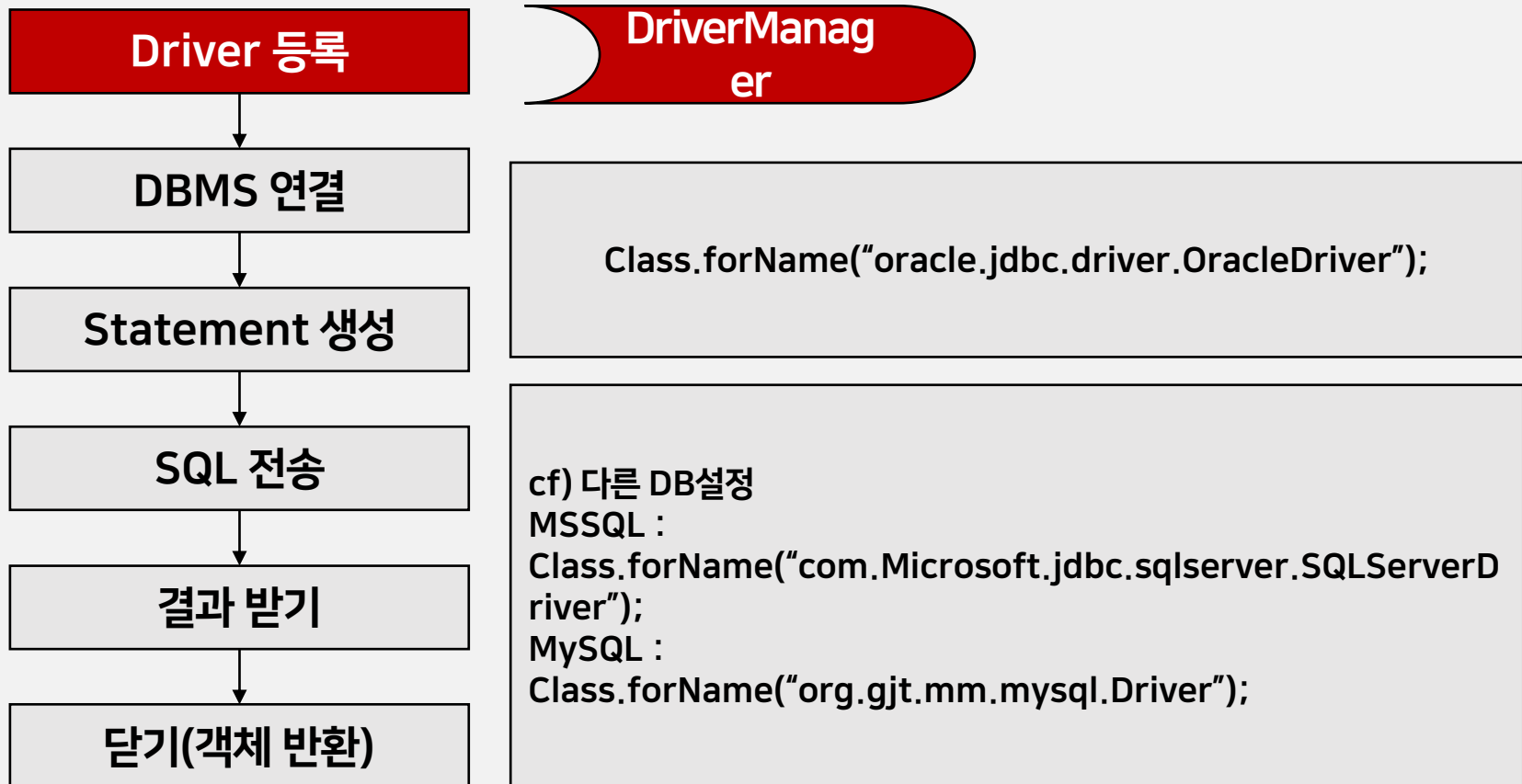
✓ ResultSet

SELECT문을 사용한 질의 성공 시 ResultSet반환
SQL질의에 의해 생성된 테이블을 담고 있으며
커서(cursor)로 특정 행에 대한 참조 조작



▶ JDBC 코딩 절차

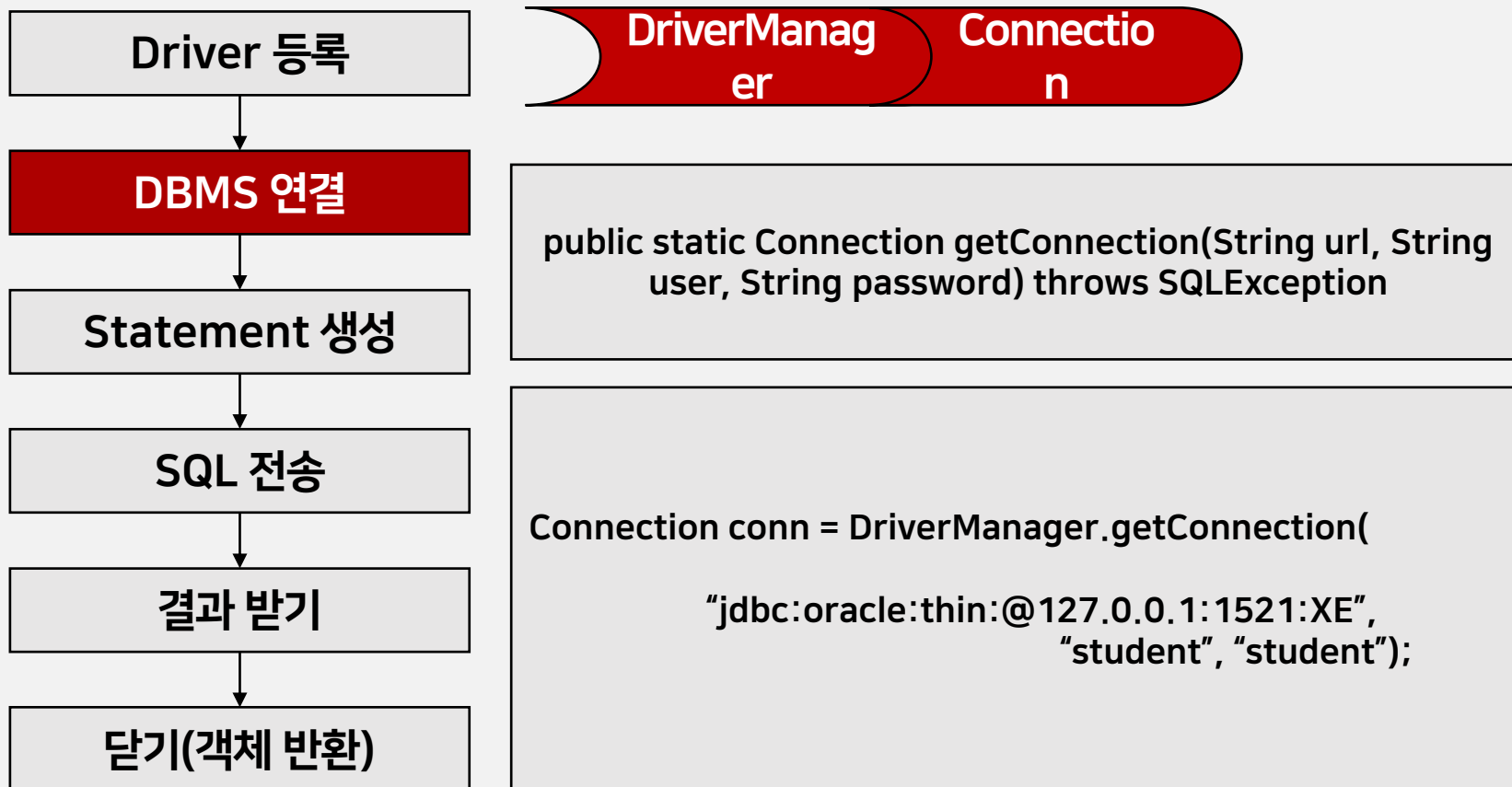
✓ DriverManager에 해당 DBMS Driver 등록



* 반드시 `ClassNotFoundException` 처리를 해야 함

▶ JDBC 코딩 절차

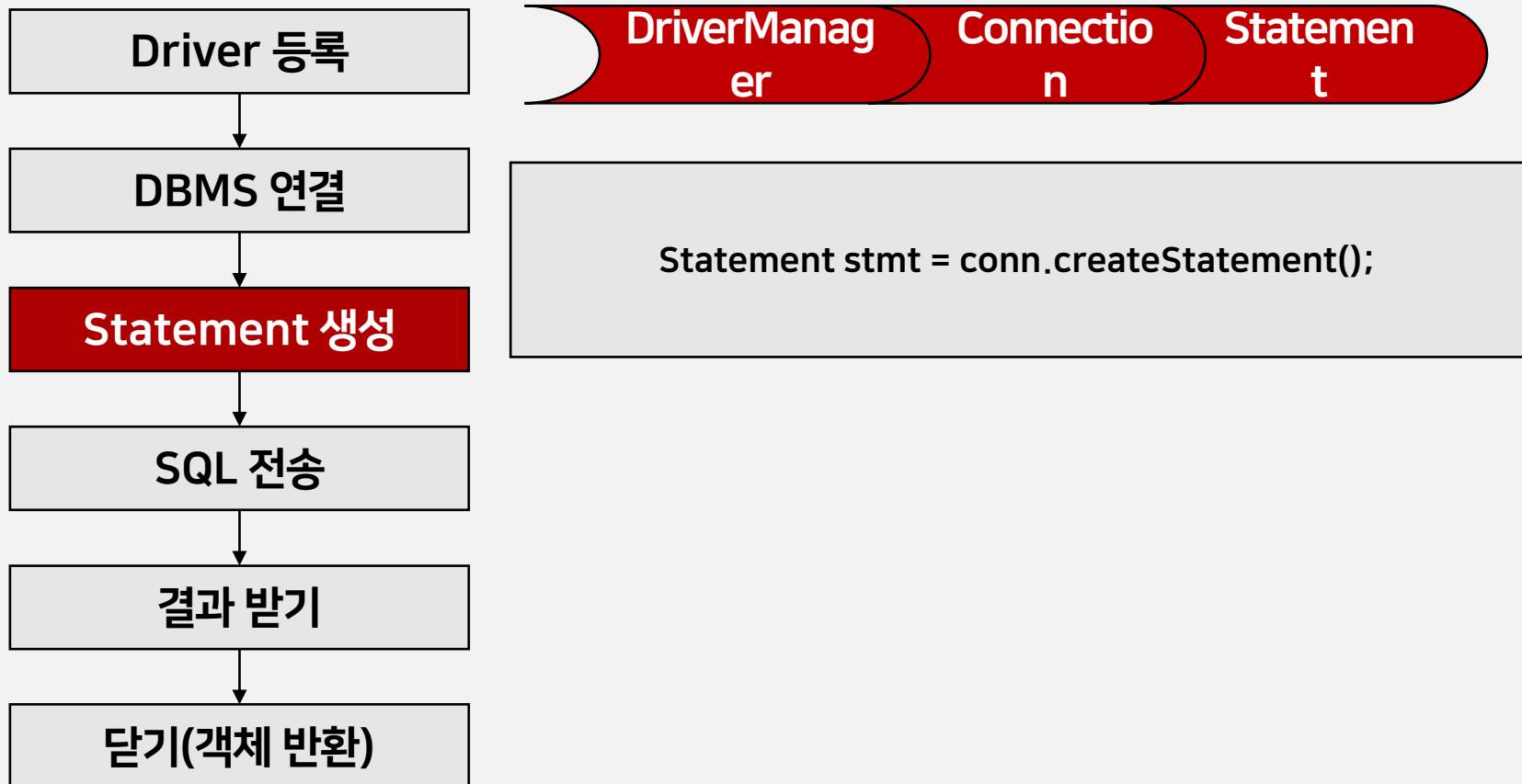
✓ 해당 Driver로부터 Connection instance 획득



* 반드시 SQLException처리를 해야 함

▶ JDBC 코딩 절차

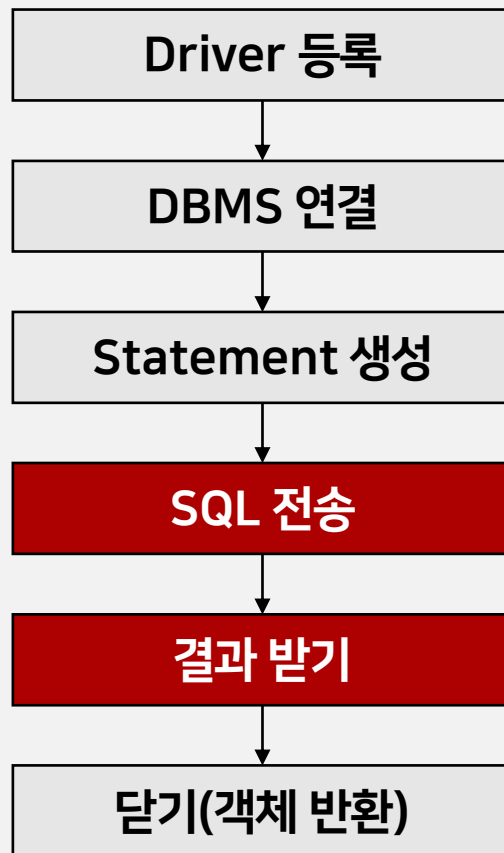
✓ 해당 Driver로부터 Connection instance 획득



* 반드시 SQLException처리를 해야 함

▶ JDBC 코딩 절차

- ✓ Statement method를 이용하여 SQL문 실행
- ✓ 실행결과를 ResultSet 혹은 int형 변수로 받아서 처리



```
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));
}
```

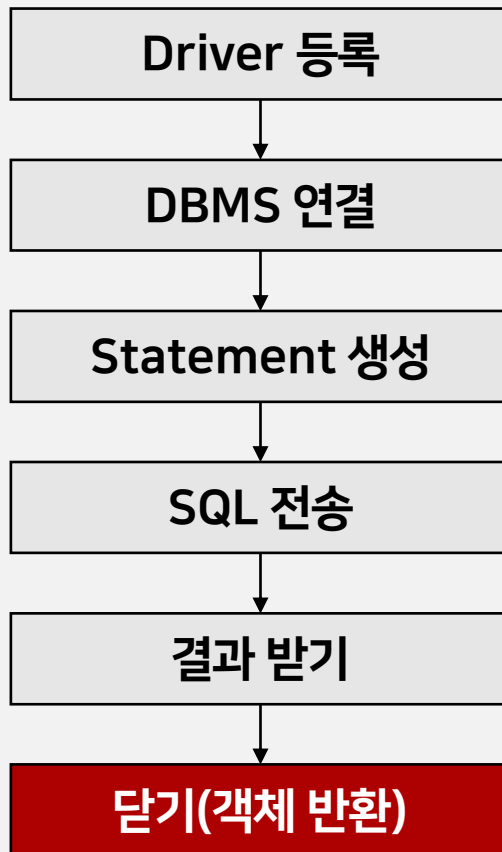
```
String query = "UPDATE EMP SET LAST_NAME = 'KIM'
    + " WHERE ID = '10000'";
```

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

* 반드시 SQLException처리를 해야 함

▶ JDBC 코딩 절차

- ✓ Statement method를 이용하여 SQL문 실행
- ✓ 실행결과를 ResultSet 혹은 int형 변수로 받아서 처리



DriverManager Connection Statement ResultSet

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

* 반드시 SQLException처리를 해야 함