

JDBC



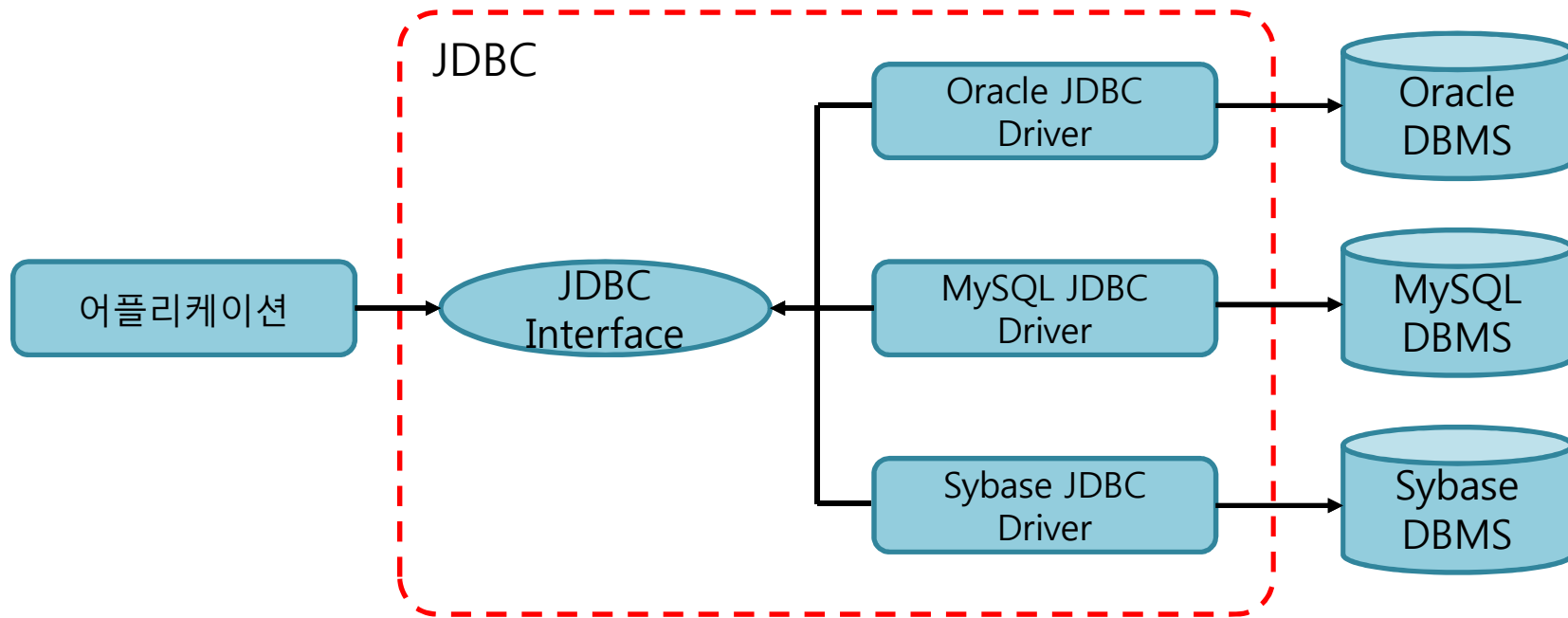


JDBC(Java DataBase Connectivity)란?

자바 언어에서 Database에 접근할 수 있게 해 주는 Programming API이다.

Java에서 DBMS연동에 필요한 메소드를 Connection인터페이스 제공
각 밴더(vendor)별 구현하여(jar파일) DBMS에 접속 가능하게 함.

Java.sql 패키지에서 관리



설치 및 세팅





OJDBC 다운로드

버전별 파일이름

- oracle 7 : classes111.zip / oracle 8 : classes12.zip
- oracle 9i : ojdbc14.jar / oracle 10g : ojdbc14.jar
- oracle 11g : ojdbc6.jar

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.

[ojdbc6g.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: d268a890a9a681cf498a9fe9c47e92ca06ac26fd)
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 >](#)



오라클에서 OJDBC파일 가져오기

- express버전

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

- Enterprise 버전

**C:\Wapp\W사용자계정\Wproduct\W11.2.0
Wdbhome_1\Wjdbc\Wlib\Wojdbc6**





jdk에 JDBC설치하기

다운받거나 복사한 **ojdbc**파일을 **jdk**폴더에 넣어
줘야 한다.

- 설치경로

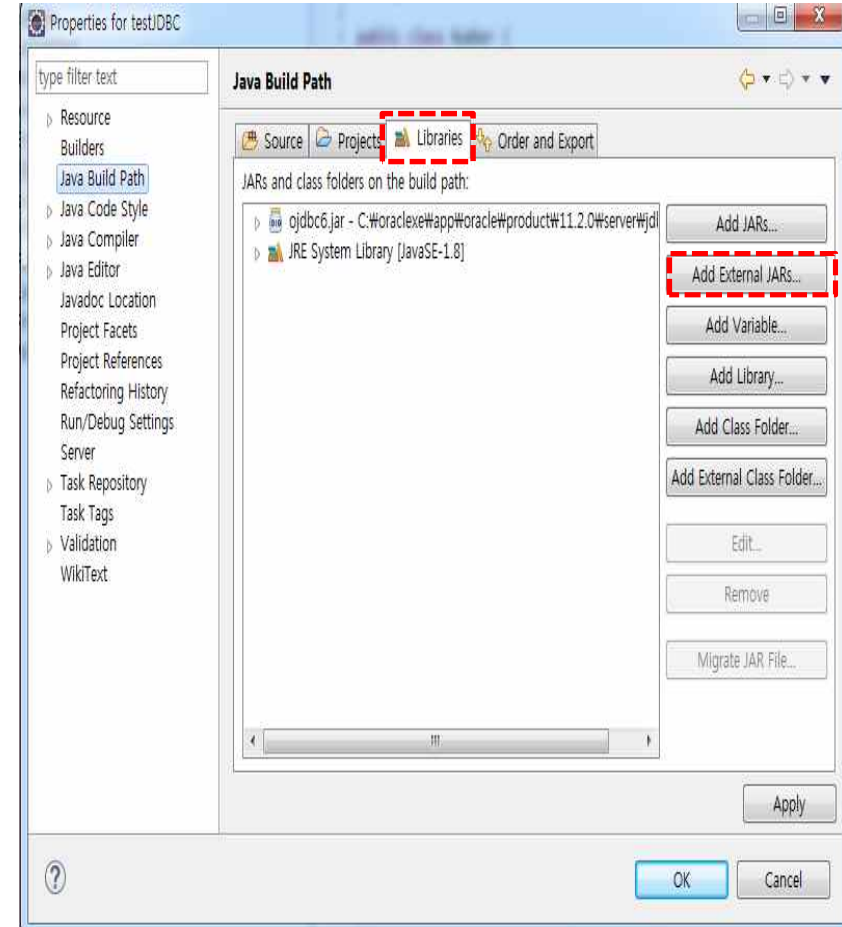
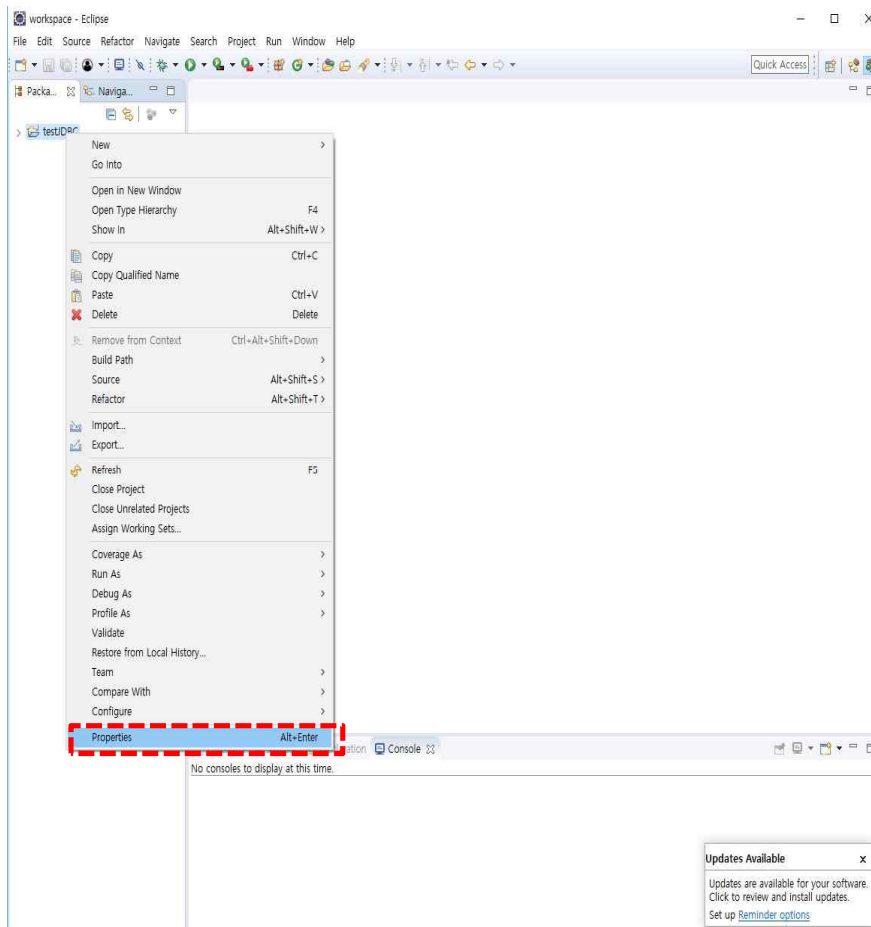
C:\Program Files\Java\jdk1.8.0_144
\jre\lib\ext





JDBC

Eclipse Library 등록



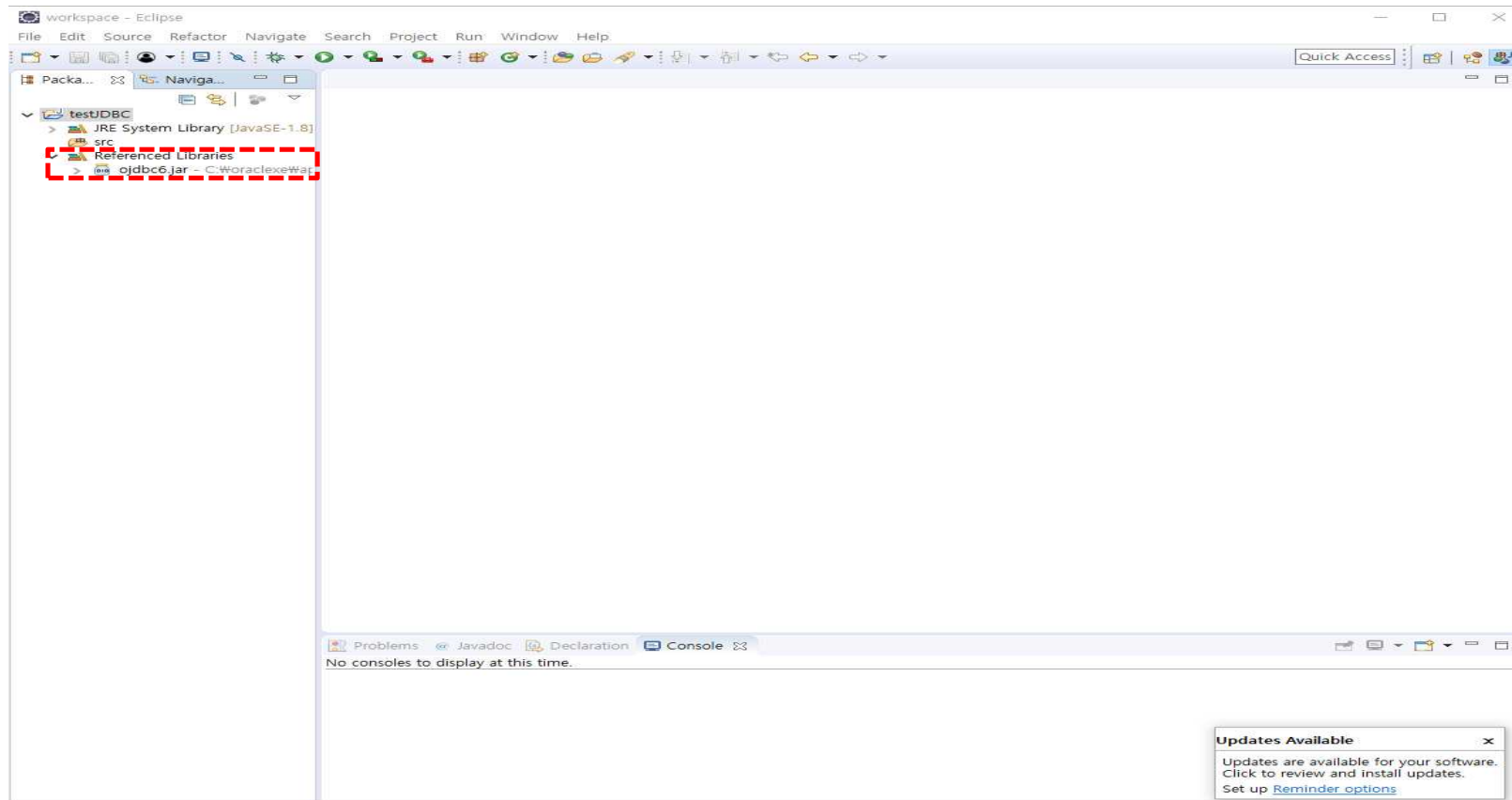
프로젝트에서 우클릭하여 properties클릭

Libraries선택 후 Add External JARs.. 클릭 후 ojdbc6파일이 있는 폴더로
가서 파일 선택 apply클릭



JDBC

Eclipse Library 등록

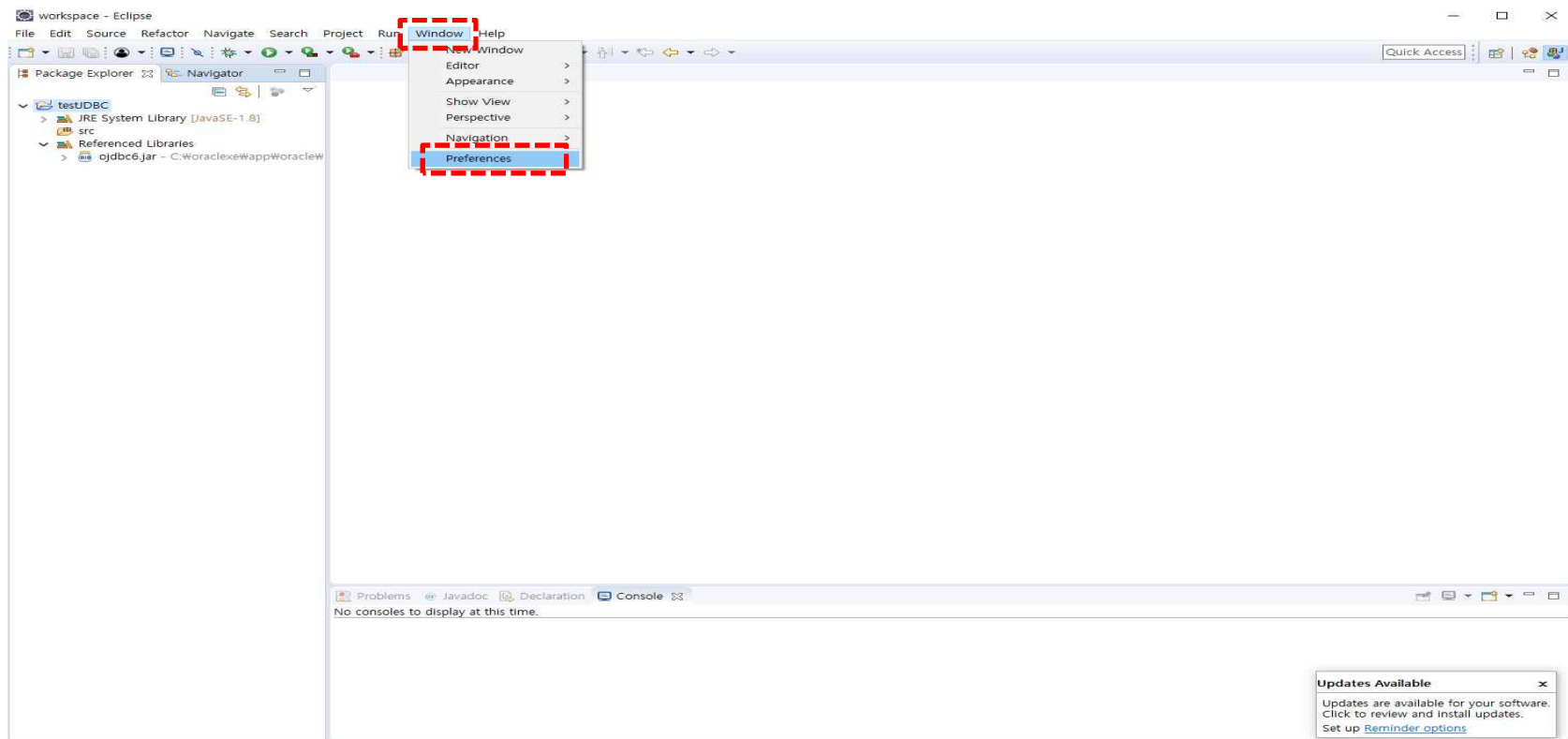


package Explorer에 Referenced Libraries가 생기고 ojdbc6.jar가 생성됨.



이클립스 Character Set설정

문자 인코딩 방식이 맞지 않으면 해당 문자가 제대로 출력되지 않아 이클립스 작업파일에 대한 문자 셋을 일치시켜줘야 한다. (UTF-8)

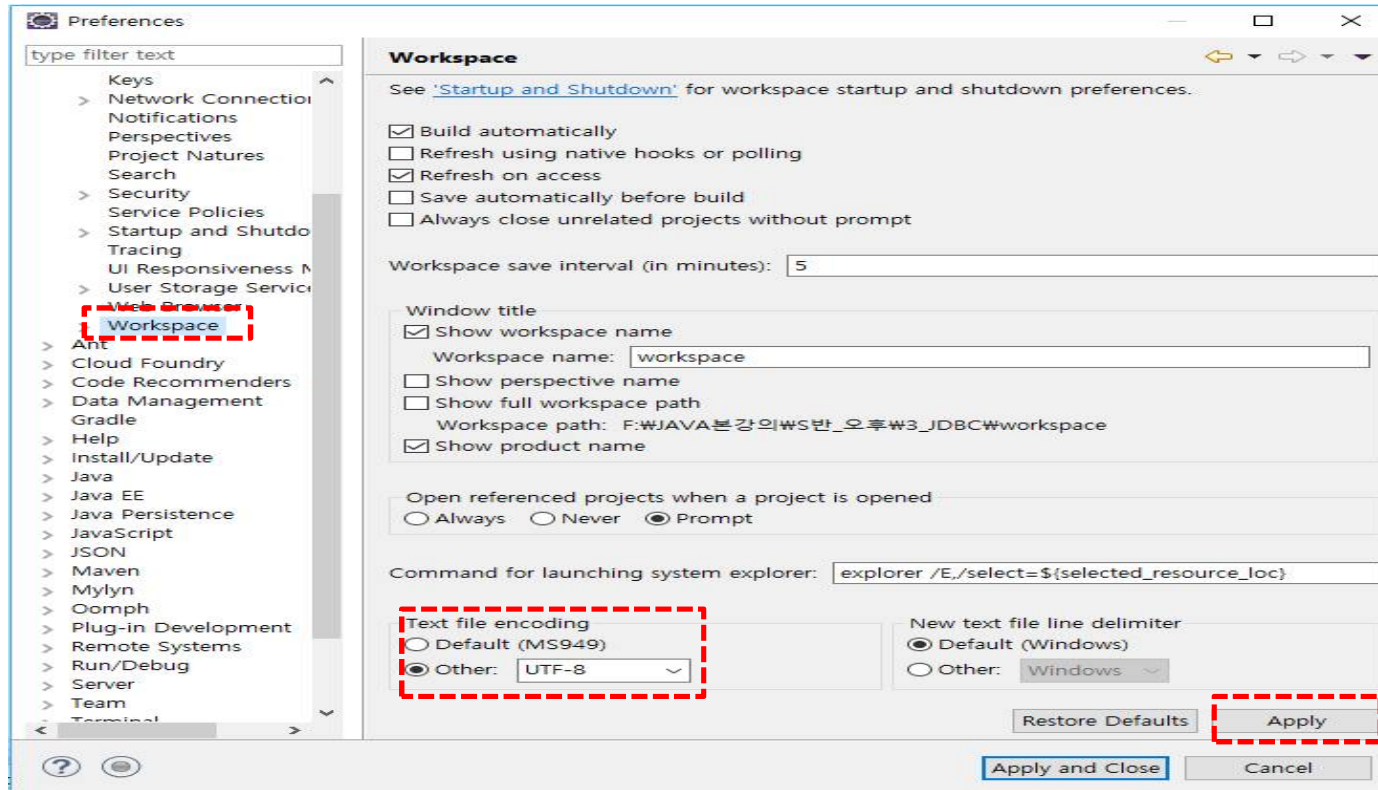


window – preferences 클릭



이클립스 Character Set

- 기본 문자 인코딩



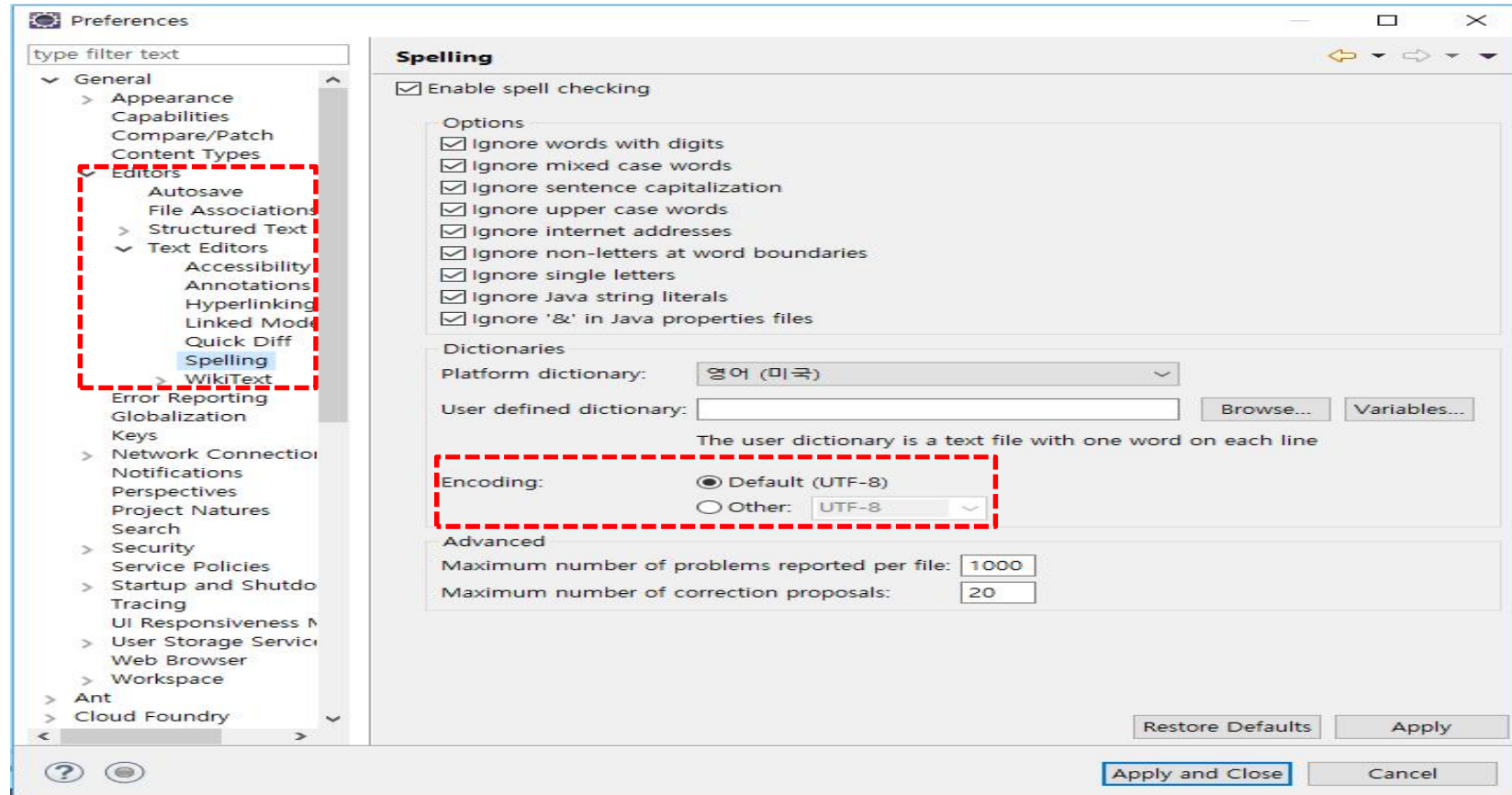
General - Workspace – Text file encoding – other에서 UTF-8 선택





이클립스 Character Set

- 기본 문자 인코딩



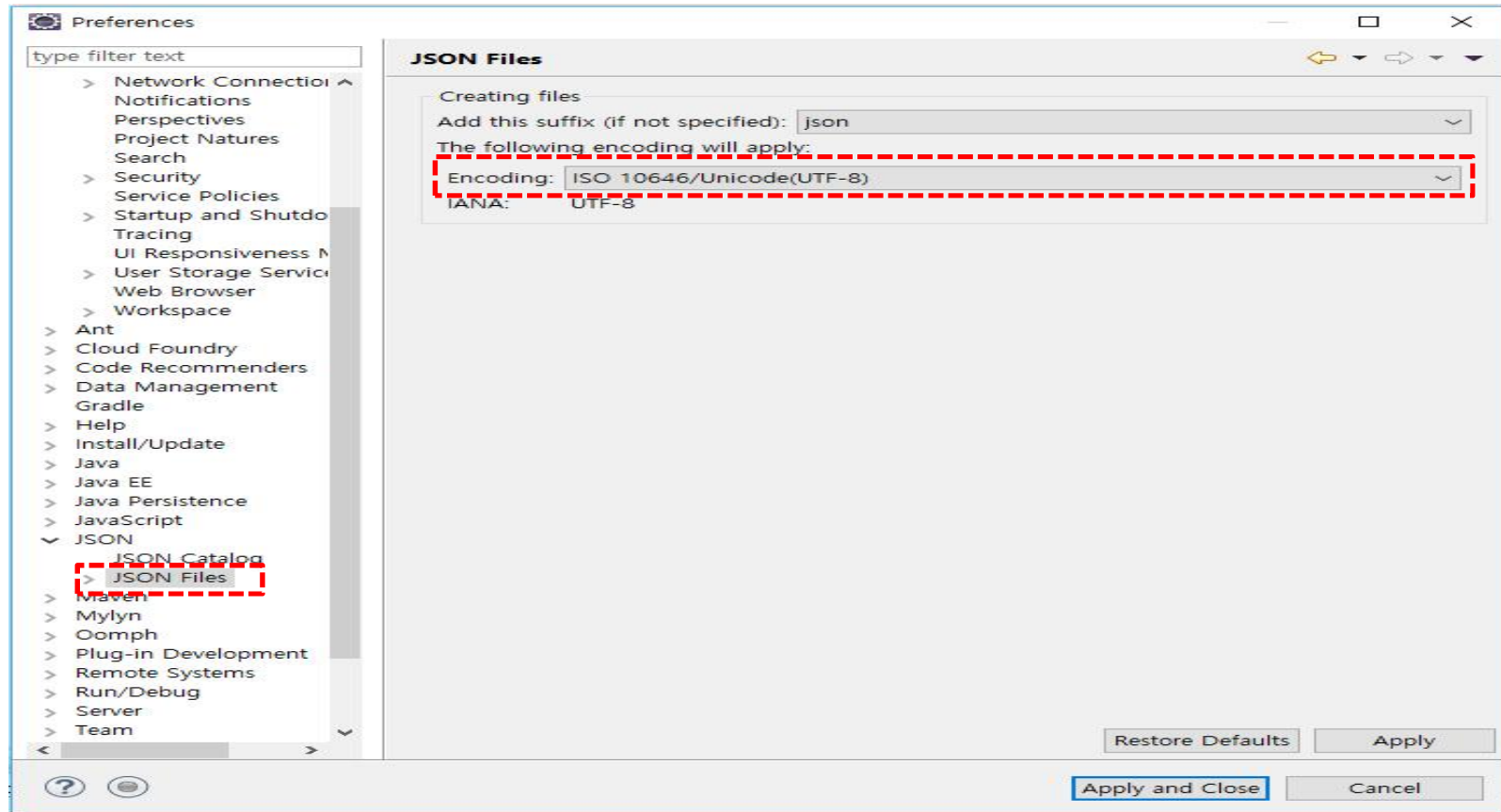
General - Editors – Text Editors – Spelling UTF-8 변경





이클립스 Character Set

- JSON 인코딩



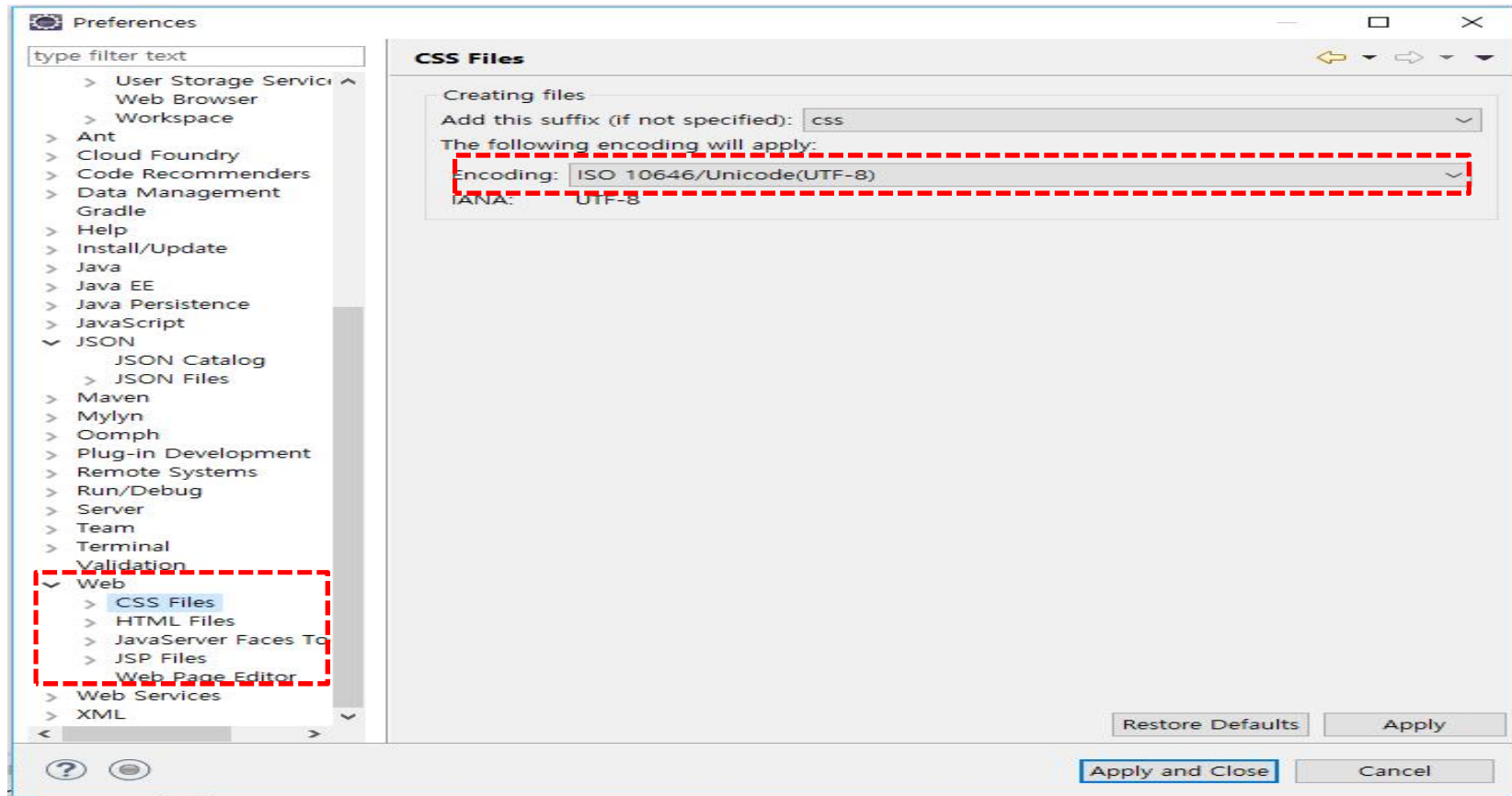
#JSON – JSON Files – Encoding – ISO 10646/Unicode(UTF-8) 선택





이클립스 Character Set

- Web 인코딩



CSS Files, HTML Files, JSP Files도 수정



DataBase 생성





연결을 위한 Test DB

- 사용자 계정 생성 및 권한 설정

계정명 : student / student

권 한 : connect, resource

- DB생성(테이블명 : MEMBER)

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 MEMBER_ID	VARCHAR2(15 BYTE)	No	(null)	1	회원아이디
2 MEMBER_PWD	VARCHAR2(15 BYTE)	No	(null)	2	회원비밀번호
3 MEMBER_NAME	VARCHAR2(20 BYTE)	No	(null)	3	회원이름
4 GENDER	CHAR(1 BYTE)	Yes	(null)	4	성별
5 AGE	NUMBER	No	(null)	5	나이
6 EMAIL	VARCHAR2(30 BYTE)	Yes	(null)	6	이메일
7 PHONE	CHAR(11 BYTE)	Yes	(null)	7	전화번호
8 ADDRESS	VARCHAR2(500 BYTE)	Yes	(null)	8	주소
9 HOBBY	VARCHAR2(50 BYTE)	Yes	(null)	9	취미
10 ENROLL_DATE	DATE	Yes	sysdate	10	가입날짜





연결을 위한 Test DB

- 제약조건

CONSTRAINT_TYPE	SEARCH_CONDITION	COLUMN_NAME
C	"MEMBER PWD" IS NOT NULL	MEMBER PWD
C	"MEMBER NAME" IS NOT NULL	MEMBER NAME
C	"AGE" IS NOT NULL	AGE
C	gender in ('M','F')	GENDER
P	(null)	MEMBER ID

- 데이터

MEMBER_ID	MEMBER_PWD	MEMBER_NAME	GENDER	AGE	EMAIL	PHONE	ADDRESS	HOBBY	ENROLL_DATE
1 admin	admin	관리자	M	30	admin@iei.or.kr	0101234678	서울시 강남구 역삼동 테헤란로 7	기타, 독서, 운동	16/03/15
2 user11	pass11	홍길동	M	23	hong@kh.org	01017778888	경기도 수원시 팔달구 팔달동 77	운동, 등산, 기타	17/09/21
3 user22	pass22	신사임당	F	48	shin50@kh.org	01050005555	강원도 강릉시 오죽헌 5	독서, 그림, 요리	17/05/05
4 user77	user77	이순신	M	50	ditnstls@naver.com	01021226374	경기도 시흥시	음악	17/12/08
5 lsj	lsj	이선주	F	24	dltjsw@naver.com	01021226377	경기도 안산시	운동, 음악, 댄스	17/08/25
6 seonn	seonn	김공부	F	28	studyl1@naver.com	01021226312	경기도 성남시	공부, 책읽기	17/11/08



JDBC class와 interface





JDBC class/interface

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.

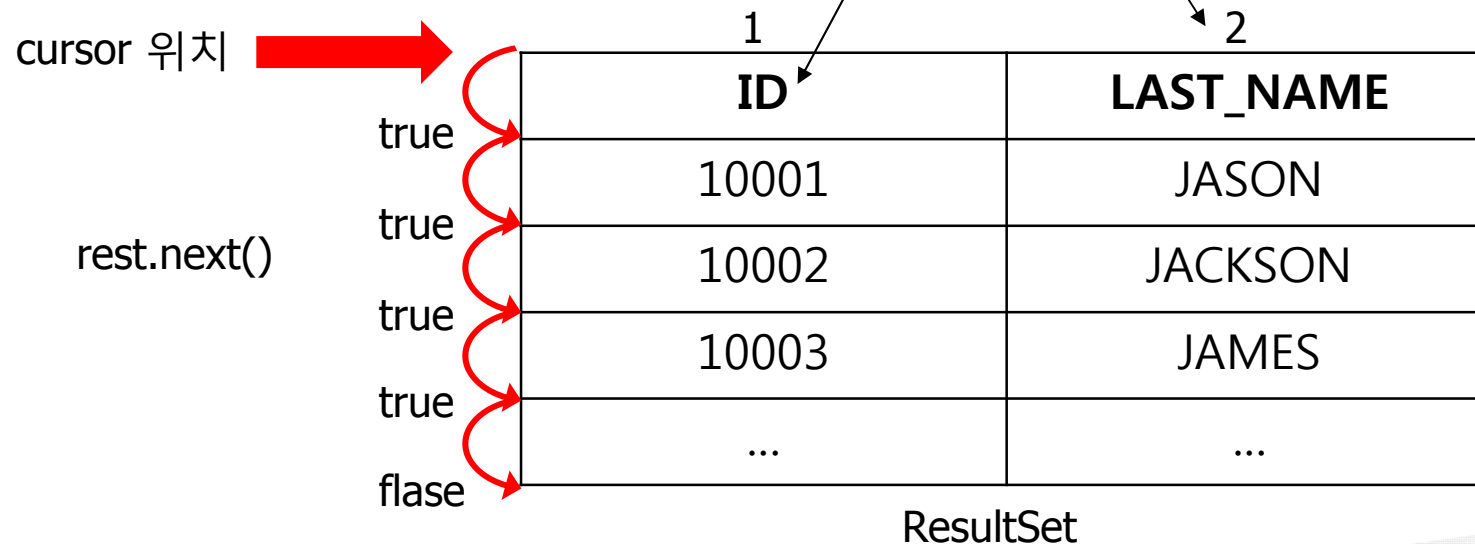
```
예) try{  
    String query = "insert into member value (?, ?);  
    pstmt = conn.prepareStatement(query);  
    pstmt.setString(1, id);  
    pstmt.setString(2, password);  
  
    }catch(SQLException e){  
    e.printStackTrace();  
    }
```



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);
```



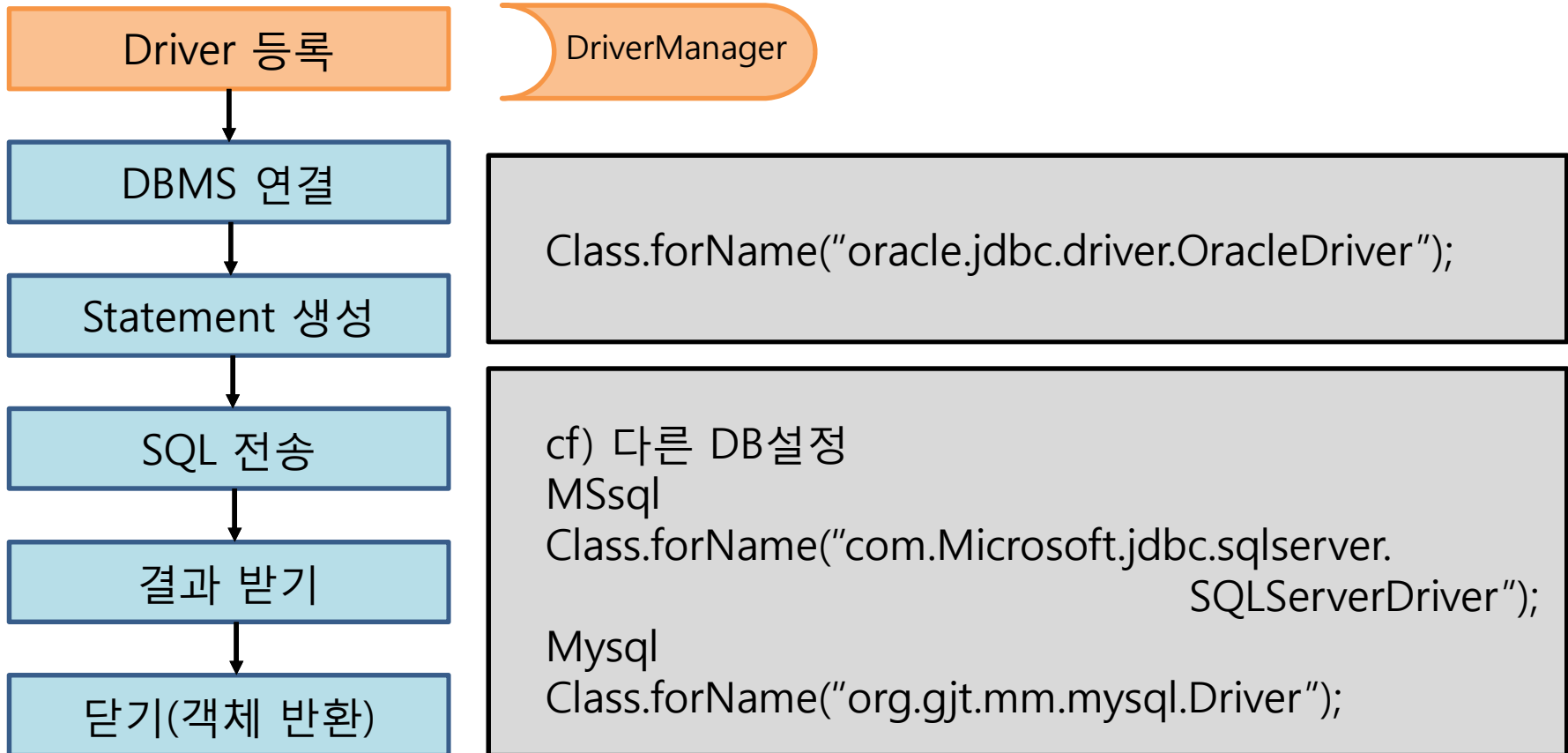
작성 절차





JDBC Coding 절차

1. DriverManager에 해당 DBMS Driver 등록

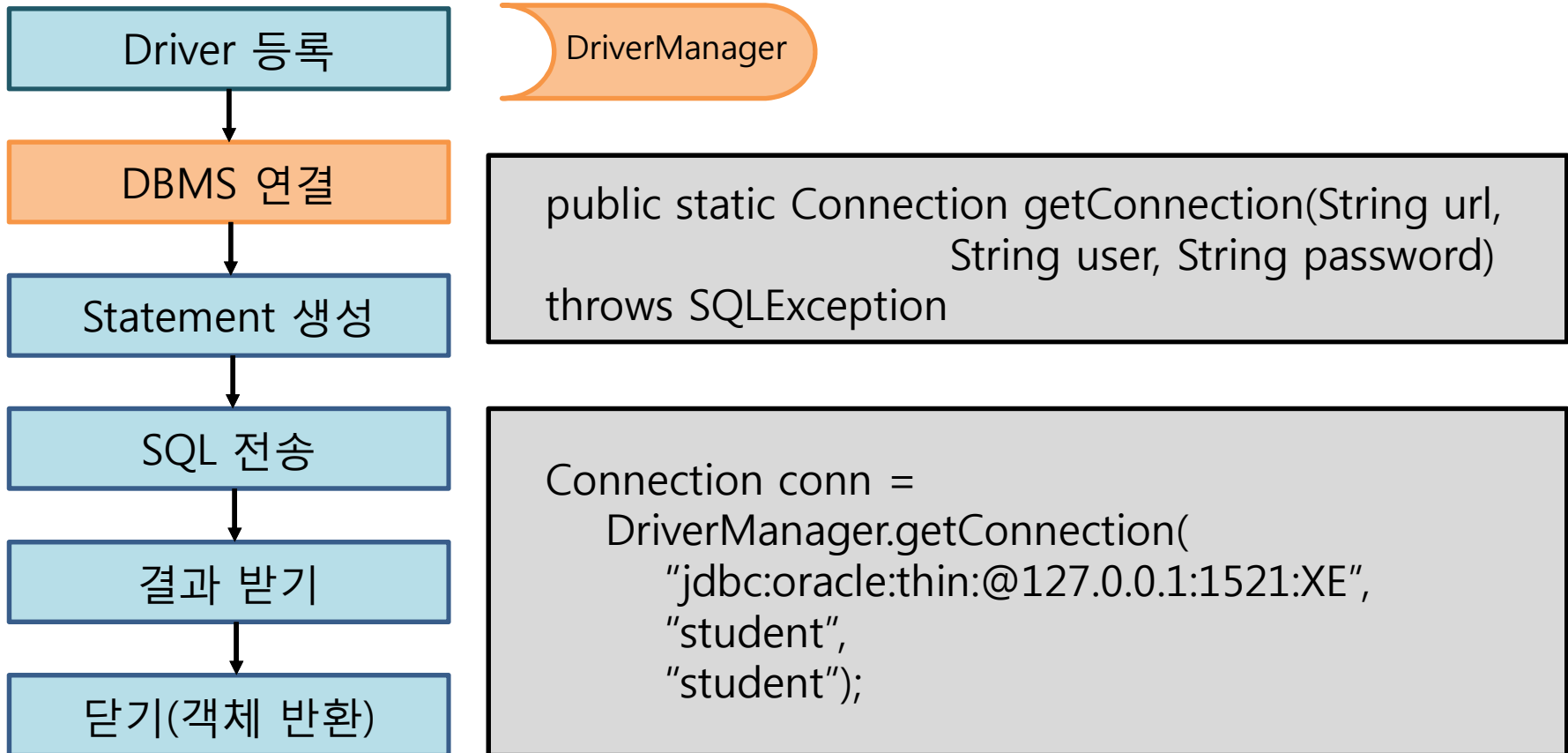


반드시 **ClassNotFoundException 처리** 해줘야한다.



JDBC Coding 절차

2. 해당 Driver로부터 Connection instance 획득



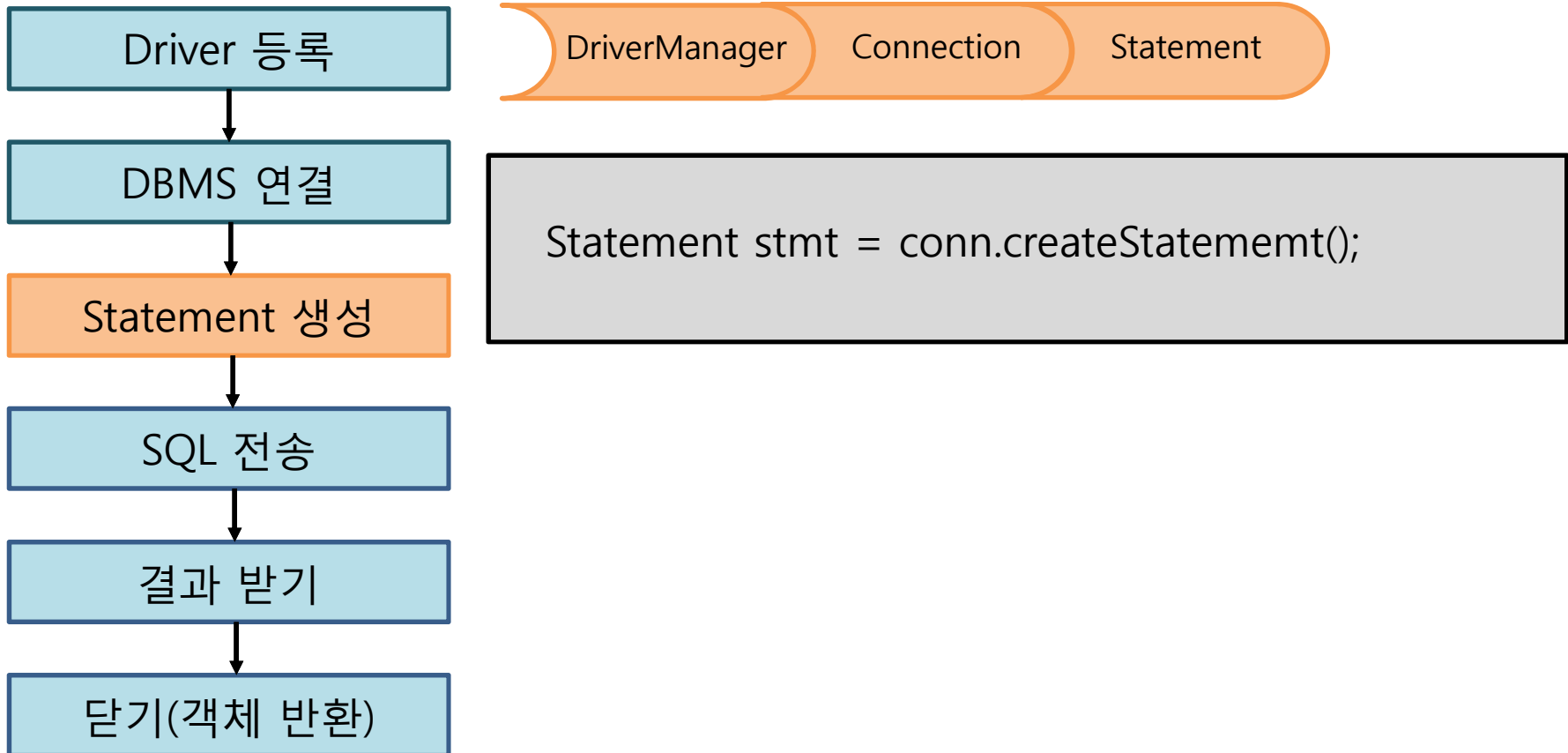
반드시 **SQLException 처리** 해줘야한다.





JDBC Coding 절차

3. Connection instance로부터 Statement instance 획득



반드시 **SQLException 처리** 해줘야한다.

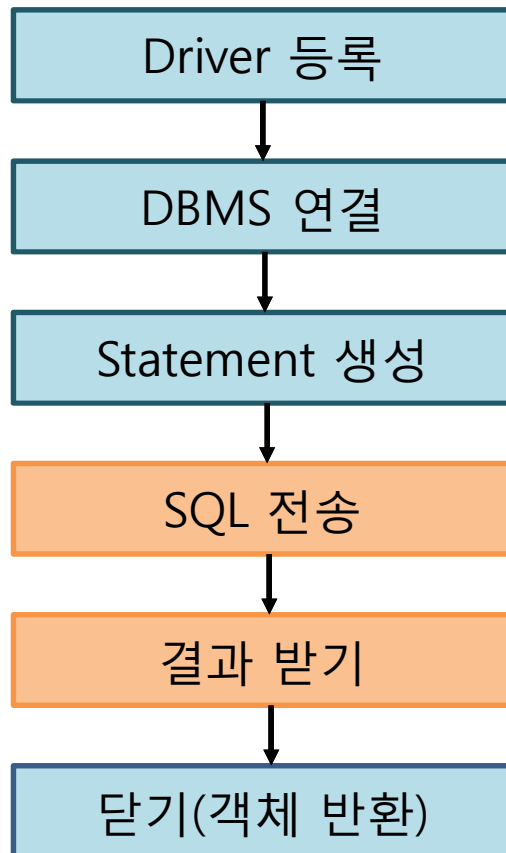




JDBC Coding 절차

4. Statement method를 이용하여 SQL문 실행

5. 실행 후 결과를 ResultSet(Selete)혹은 int형 변수(DML)로 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);
```

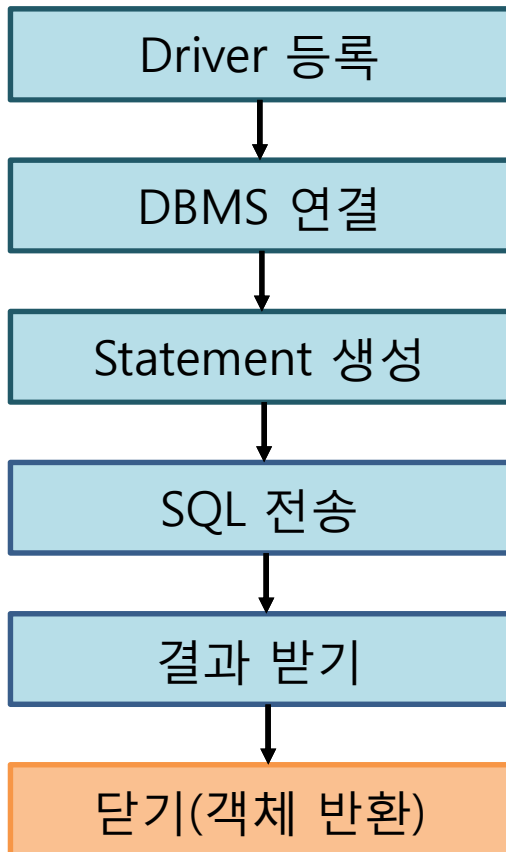
반드시 **SQLException 처리** 해줘야한다.





JDBC Coding 절차

6. 사용한 자원 반납



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

반드시 **SQLException 처리** 해줘야한다.

