

# 15. JDBC

15.1 JDBC 개요

15.2 JDBC 사용

15.3 SELECT/UPDATE

15.4 Statement/PreparedStatement

15.5 Connection Pool

## **15.1 JDBC 개요**

15.2 JDBC 사용

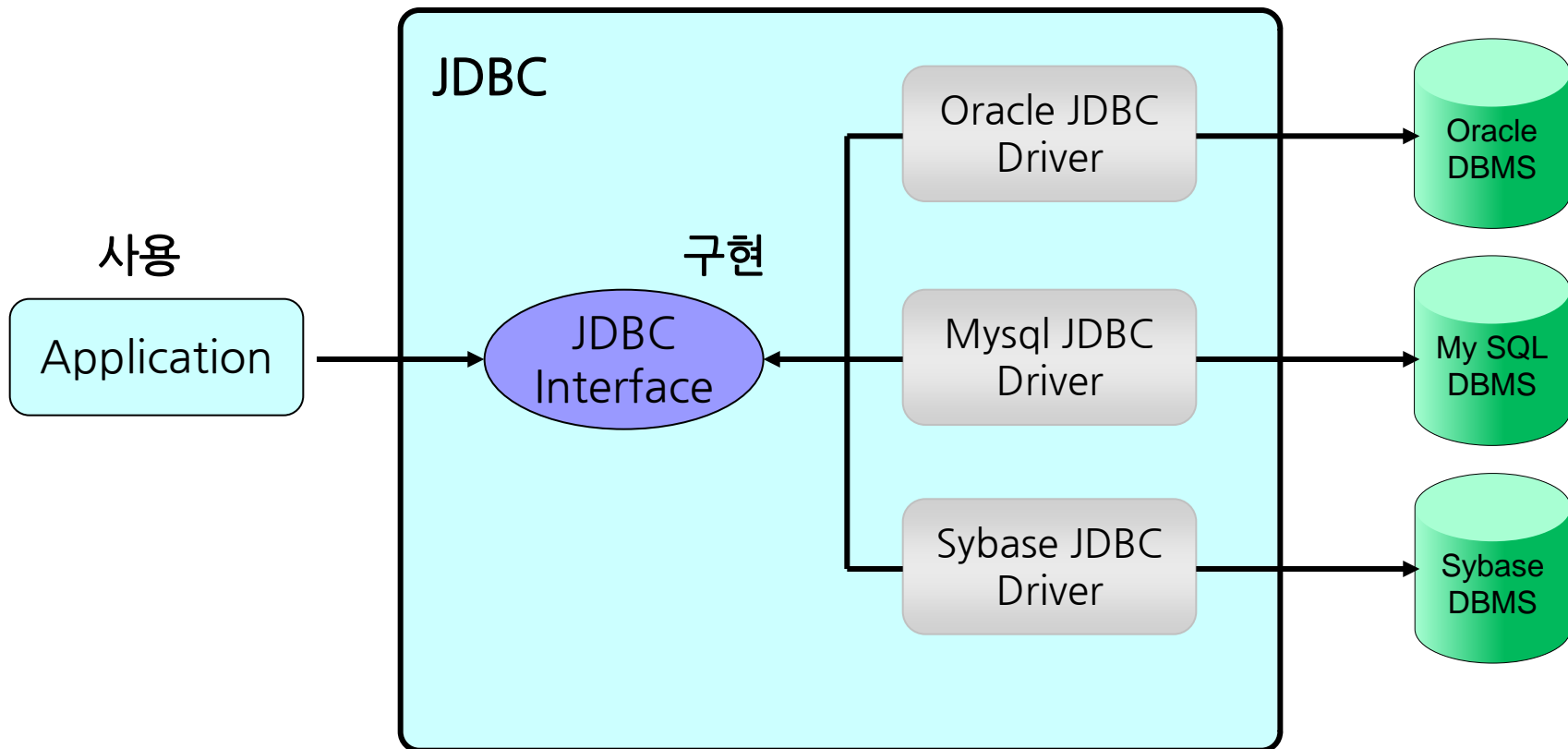
15.3 SELECT/UPDATE

15.4 Statement/PreparedStatement

15.5 Connection Pool

## ❖ JDBC의 개념

- 자바 언어에서 Database에 접근할 수 있게 해주는 Programming API



The screenshot shows a web browser window titled "Connection (Java Platform SE 6) - Windows Internet Explorer". The address bar shows the URL: <http://java.sun.com/javase/6/docs/api/index.html?overview-summary.html>. The browser's menu bar includes "파일(F)", "편집(E)", "보기(V)", "즐거찾기(A)", "도구(T)", and "도움말(H)". The "즐거찾기" (Favorites) bar shows a single entry: "Connection (Java Platform SE 6)".

The left sidebar displays a list of API packages and classes. The package `java.sql` is highlighted with a red rectangle. Below the package list, the `java.sql` package is expanded, showing a list of interfaces: `Array`, `Blob`, `CallableStatement`, and `Clob`.

The main content area displays the "Overview" tab for the `java.sql` package. The navigation bar includes links for "Overview", "Package", "Class", "Use", "Tree", "Deprecated", "Index", and "Help". The "Class" tab is selected. The page title is "Interface Connection". The "All Superinterfaces:" section lists `Wrapper`. The "public interface Connection" is defined as extending `Wrapper`. The description states: "A connection (session) with a specific database. SQL statements of a connection."

Oracle JDBC Drivers release 10.2.0.1.0 Downloads - Windows Internet Explorer

http://www.oracle.com/technology/software/tech/java/sqlj\_jdbc/htdocs/jdbc\_10201.html

파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(I) 도움말(H)

★ 즐겨찾기 Oracle JDBC Drivers release 10,2,0,1,0 Downl...

**ORACLE**  
TECHNOLOGY NETWORK

**PRODUCTS**  
Database  
Middleware  
Developer Tools  
Enterprise Management  
Applications Technology  
Products A-Z

**TECHNOLOGIES**  
BI & Data Warehousing  
Embedded  
Java  
Linux  
.NET  
PHP  
Security  
Technologies A-Z

**ARCHITECTURE**  
Enterprise Architecture  
Enterprise 2.0  
Grid  
Service-Oriented Architecture

shortcuts GETTING STARTED **DOWNLOADS** DOCUMENTATION FORUMS ART

Downloads Index

Database

Fusion Middleware

Enterprise Manager

JDeveloper

SQL Developer

More...

**Oracle Database 10g Release 2 (10.2.0.1.0) JDBC Drivers**

Thank you for accepting the OTN License Agreement. To download this software.

☐ [JavaDoc](#) (4,383,404 bytes)

☐ [README](#)

**JDBC Thin for All Platforms**

☐ [classes12.jar](#) (1,609,607 bytes) - for use with JDK 1.2 and JDK 1.3

☐ [classes12\\_q.jar](#) (2,055,921 bytes) - same as classes12.jar, except that classes were con and contain some tracing information.

☐ [classes12dms.jar](#) (1,617,382 bytes) - same as classes12.jar, except that it contains addi

The screenshot shows a Microsoft Internet Explorer browser window displaying the MySQL Developer Zone website. The address bar shows <http://www.mysql.org/>. The page has a blue header with the MySQL logo and navigation links: MySQL.com, MySQL Network, Developer Zone (selected), Partners, and Online Shop. A secondary navigation bar includes links like Downloads (highlighted with a red box), Documentation, Forums, Lists, Bugs, Events, User Groups, Guilds, Blogs, Support, Resources, and Books. Below this, the 'Download Connector/J 5.0' link is also highlighted with a red box. The main content area features a 'NOTE' about the GPL license and a section for downloading the driver. It lists two options: 'Source and Binaries (tar.gz)' and 'Source and Binaries (zip)', both for version 5.0.0-beta (8.0M and 8.1M respectively). Each option includes a 'Pick a mirror' link and an MD5 checksum. A 'Related pages' sidebar on the right lists links to Product Information, Documentation, and Connector/J 3.1 and 3.0 Downloads. At the bottom, there is a section to 'Subscribe to the monthly MySQL Newsletter'.

http://www.mysql.org/ - Microsoft Internet Explorer

파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)

주소(D) <http://www.mysql.org/>

MySQL.com MySQL Network **Developer Zone** Partners Online Shop

Downloads Documentation Forums Lists Bugs Events User Groups Guilds Blogs Support Resources Books

FAQ

**Download Connector/J 5.0**

Overview Database Server Cluster MaxDB Migration Toolkit Administrator Query Browser

Mirror Sites

**NOTE:** By downloading the software from this page, you acknowledge that the software available from here is licensed under the GPL. We advise that you review the [GPL](#) before downloading.

If you need commercial, non-GPL, licenses, you can order them [online](#).

[MySQL Connector/J](#) is the official JDBC driver for MySQL. On this page are downloads of the current development release, and [the latest production version](#) is available on another page. A [list of changes](#) is available in the documentation.

We suggest that you [use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download](#).

|   |            |      |                               |
|---|------------|------|-------------------------------|
| Source and Binaries (tar.gz)  | 5.0.0-beta | 8.0M | <a href="#">Pick a mirror</a> |
| MD5: <a href="#">8b673dde79ba5539f49183d2a410b1fb</a>   <a href="#">Signature</a> |            |      |                               |
| Source and Binaries (zip)   | 5.0.0-beta | 8.1M | <a href="#">Pick a mirror</a> |
| MD5: <a href="#">30557d6553b35343055400110-0004</a>   <a href="#">Signature</a>   |            |      |                               |

**Related pages:**

- [Product Information](#)
- [Documentation](#)
- [Connector/J 3.1 Downloads](#)
- [Connector/J 3.0 Downloads](#)

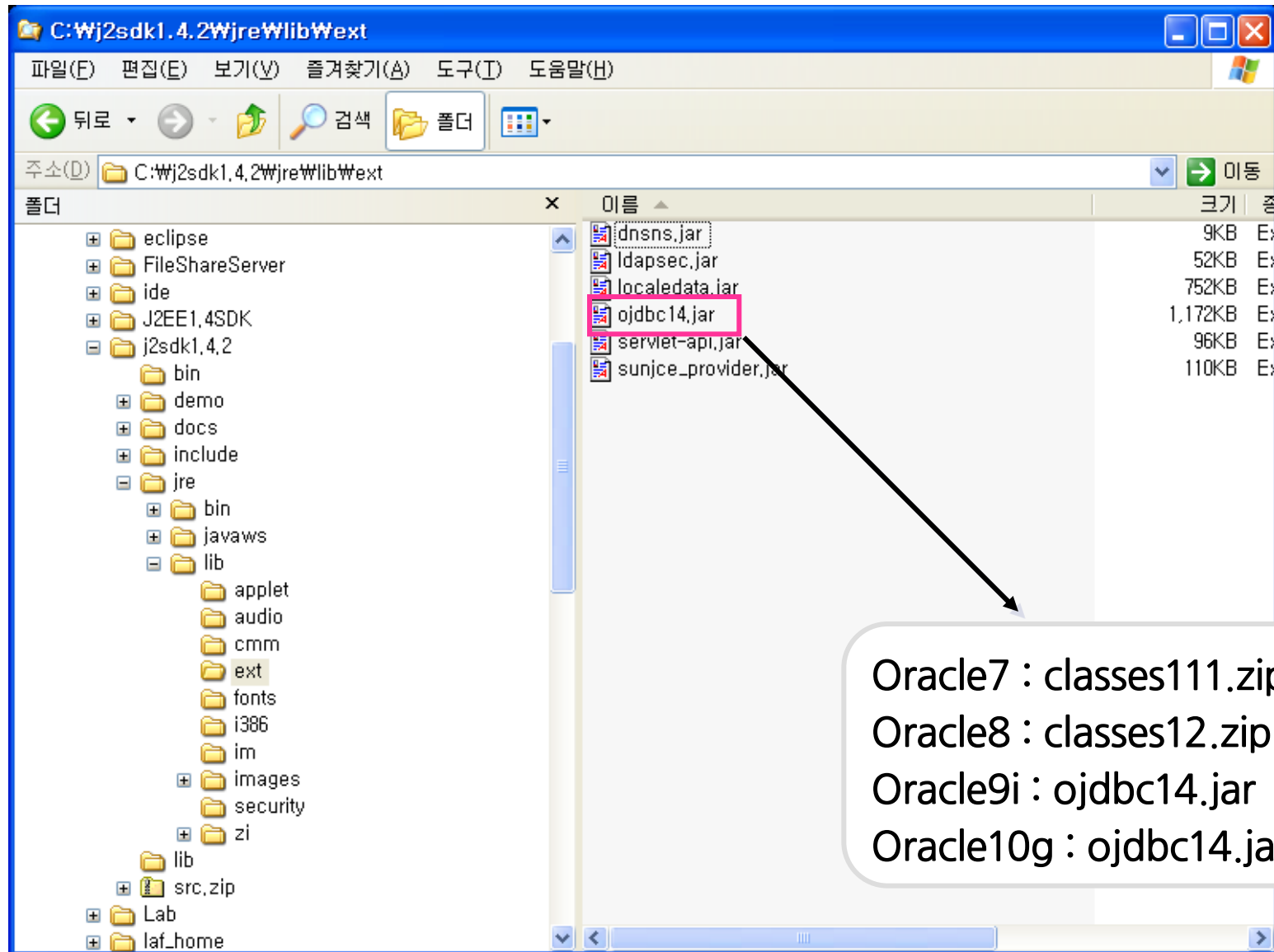
[Check out the Pogo Linux DataWare 2600:](#) a premium out-of-the-box database solution with superior performance at a low price-point.

Learn about new MySQL releases, technical articles, events and more.

**Subscribe to the monthly MySQL Newsletter**

알 수 없는 영역 (혼합)

❖ JAVA\_HOME \jre\lib\ext 에 driver를 추가해야 함 : ojdbc14.jar



15.1 JDBC 개요

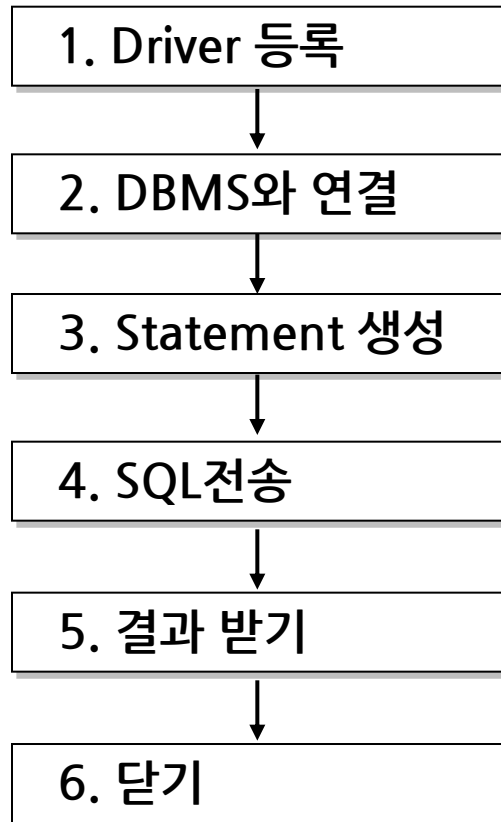
 **15.2 JDBC 사용**

15.3 SELECT/UPDATE

15.4 Statement/PreparedStatement

15.5 Connection Pool





## 1. DriverManager에 해당 DBMS Driver 등록

1. Driver 등록

2. DBMS와 연결

3. Statement 생성

4. SQL전송

5. 결과 받기

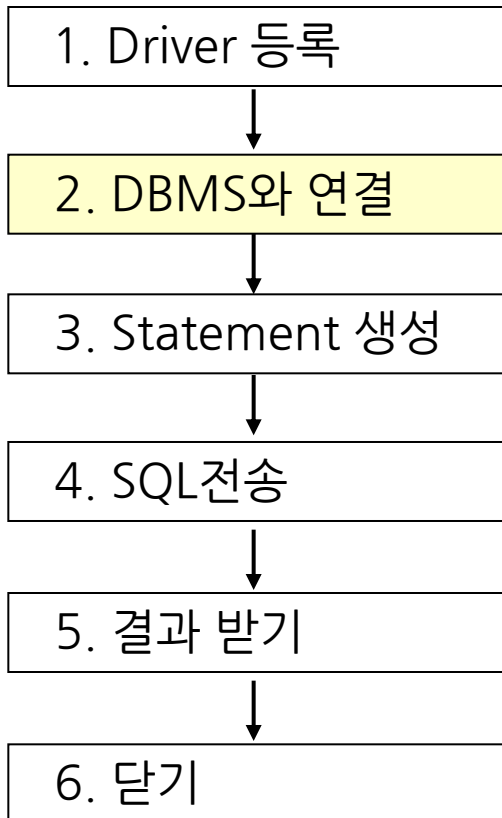
6. 닫기



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

```
cf)  
Class.forName( "com.microsoft.jdbc.sqlserver.SQLServerDriver" );  
Class.forName( "org.gjt.mm.mysql.Driver" );
```

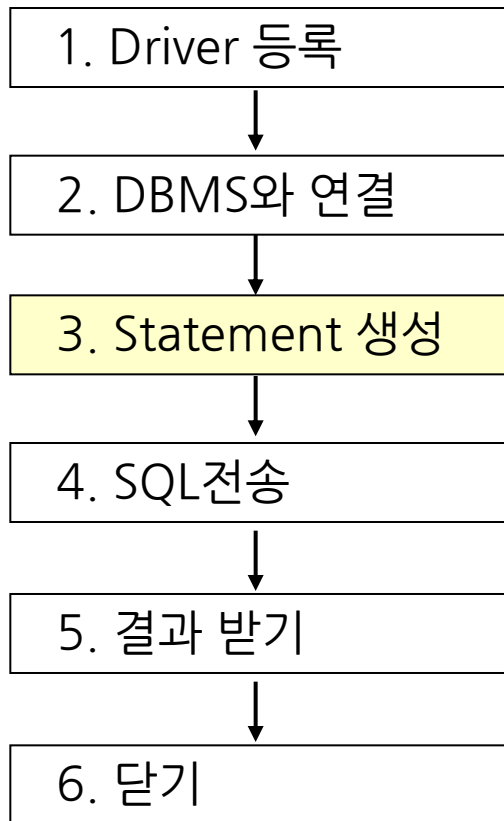
## 2. 해당 Driver로부터 Connection instance를 획득



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

```
Connection conn =
    DriverManager.getConnection(
        "jdbc:oracle:thin:@192.168.0.200:1521:VCC",
        "SEXXXXX",
        "SEXXXXX" );
```

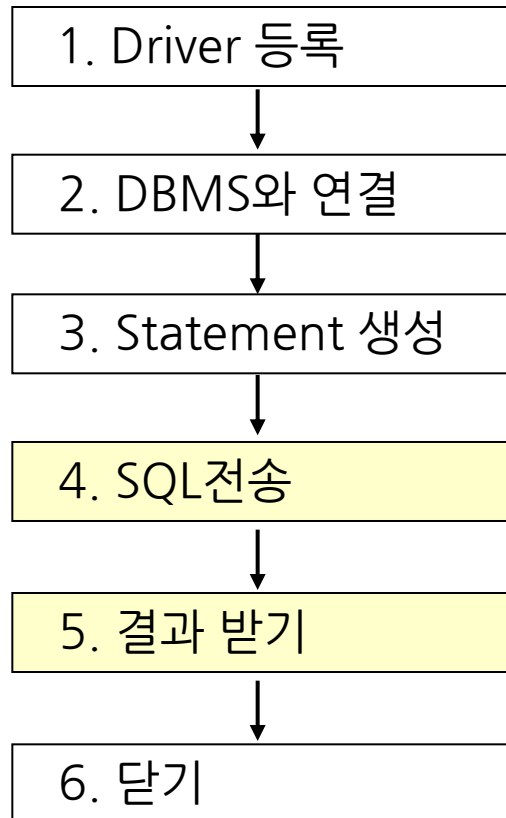
### 3. Connection instance로부터 Statement instance 획득



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

4. Statement method를 이용하여 SQL 실행

5. 실행 후 결과를 ResultSet(SELECT) 혹은 int형 변수(DML)로 받아 처리



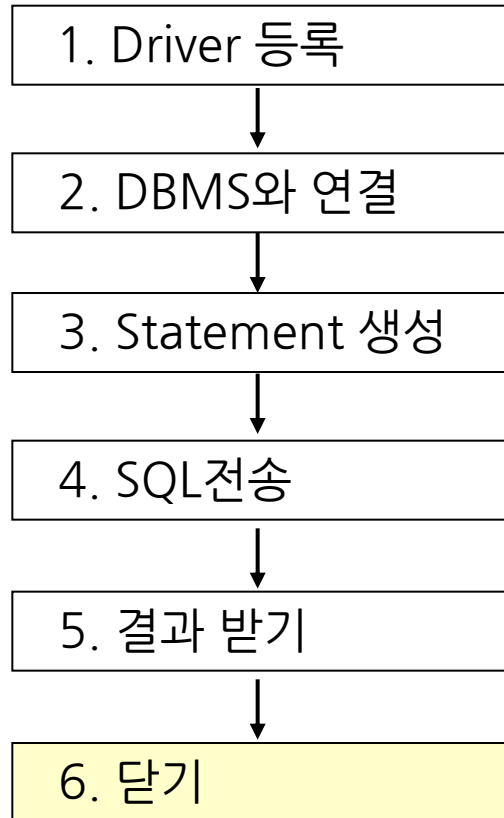
#### Select

```
String query = "SELECT ID, LAST_NAME FROM EMP";  
ResultSet rset = stmt.executeQuery( query );  
  
while ( rset.next() ) {  
    System.out.println( rset.getString( "ID" ) + " " +  
                        + rset.getString( 2 ) );  
}
```

#### DML

```
String query = "UPDATE EMP "  
              + " SET LAST_NAME = 'KIM' "  
              + " WHERE ID = '100000' ";  
int result = stmt.executeUpdate( query );
```

## 6. 사용한 자원 반납



Select

```
rset.close();  
stmt.close();  
conn.close();
```

DML

```
stmt.close();  
conn.close();
```

15.1 JDBC 개요

15.2 JDBC 사용

 **15.3 SELECT/UPDATE**


15.4 Statement/PreparedStatement

15.5 Connection Pool

```
Connection conn = null;
Statement stmt = null;
ResultSet rset = null;
String url = "jdbc:oracle:thin:@192.168.0.200:1521:VCC";
// 1. DBMS Driver 로딩
Class.forName( "oracle.jdbc.driver.OracleDriver" );
// 2. Connection 객체 획득
conn = DriverManager.getConnection( url , "student01" , "student01" );
// 3. Statement 객체 생성
stmt = conn.createStatement();
// 4. SQL 실행
String query = "SELECT ID, LAST_NAME FROM EMP " ;
rset = stmt.executeQuery( query );
// 5. ResultSet을 이용한 결과 처리
while( rset.next() ){
    System.out.println( rset.getString( "ID" ) + "WtWtWt" +
        rset.getString( 2 ) );
}
// 6. 사용할 Resource 반납
rset.close();
stmt.close();
conn.close();
```

[실습] chap15WEmpList.java





| BOF   | ID    | LAST_NAME |
|-------|-------|-----------|
| ROW 1 | 10001 | BOSS      |
| ROW 2 | 10002 | JACKSON   |
| ROW 3 | 10003 | HITE      |
| ...   | ...   | ...       |
| EOF   |       |           |

rs.next() // true 리턴

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

ID = 10001

LAST\_NAME = BOSS

|       | 1     | 2         |
|-------|-------|-----------|
| BOF   | ID    | LAST_NAME |
| ROW 1 | 10001 | BOSS      |
| ROW 2 | 10002 | JACKSON   |
| ROW 3 | 10003 | HITE      |
| ...   | ...   | ...       |
| EOF   |       |           |

rs.next() // true 리턴

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

ID = 10002

LAST\_NAME = JACKSON

|       | 1     | 2         |
|-------|-------|-----------|
| BOF   | ID    | LAST_NAME |
| ROW 1 | 10001 | BOSS      |
| ROW 2 | 10002 | JACKSON   |
| ROW 3 | 10003 | HITE      |
| ...   | ...   | ...       |
| EOF   |       |           |

rs.next() // false 리턴

| BOF   | ID    | LAST_NAME |
|-------|-------|-----------|
| ROW 1 | 10001 | BOSS      |
| ROW 2 | 10002 | JACKSON   |
| ROW 3 | 10003 | HITE      |
| ...   | ...   | ...       |
| EOF   |       |           |

### ❖Exception Handling 로직 추가

```
try {  
    // 1. DBMS Driver 로딩  
    Class.forName( "oracle.jdbc.driver.OracleDriver" );  
    ...  
    // 4. SQL 실행  
    String query = "SELECT ID, LAST_NAME FROM EMP";  
    rs = stmt.executeQuery( query);  
    .....  
} catch( ClassNotFoundException ce){  
    ce.printStackTrace();  
} catch( SQLException se){  
    se.printStackTrace();  
} finally {  
    // 6. 사용할 Resource 반납  
    try {  
        rs.close();  
        stmt.close();  
        conn.close();  
    } catch ( SQLException e ) {  
        e.printStackTrace();  
    }  
}
```

[실습] chap15WEmpListWithTryCatch.java

```
public static void main( String[] args ) {
    Connection conn = null;
    Statement stmt = null;
    String url = "jdbc:oracle:thin:@192.168.0.200:1521:VCC";
    String query = "UPDATE EMP SET LAST_NAME = 'HITE' WHERE ID = '10004'";
    try {
        Class.forName( "oracle.jdbc.driver.OracleDriver" );
        conn = DriverManager.getConnection( url , "student01" , "student01" );
        conn.setAutoCommit( false );
        stmt = conn.createStatement();
        stmt.executeUpdate( query.toString() );
        conn.commit();
    } catch( ClassNotFoundException ce){
        ce.printStackTrace();
    } catch( SQLException se){
        conn.rollback();
        se.printStackTrace();
    } finally {
        try {
            .....
        } catch ( SQLException e ) {
            .....
        }
    }
}
```

[실습] chap15WEmpUpdate.java

15.1 JDBC 개요

15.2 JDBC 사용

15.3 SELECT/UPDATE

 **15.4 Statement/PreparedStatement**

---

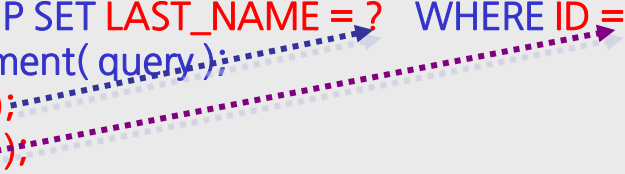
15.5 Connection Pool

```
public static void main( String[] args ) {  
    Connection conn = null;  
    Statement stmt = null;  
    String url = "jdbc:oracle:thin:@192.168.0.200:1521:VCC";  
  
    try {  
        Class.forName( "oracle.jdbc.driver.OracleDriver" );  
        conn = DriverManager.getConnection( url , "student01" , "student01" );  
        conn.setAutoCommit( false );  
        stmt = conn.createStatement();  
        String query = "UPDATE EMP SET LAST_NAME = 'HITE2' WHERE ID = '10005' ";  
        stmt.executeUpdate( query );  
        conn.commit();  
    } catch( ClassNotFoundException ce){  
        ce.printStackTrace();  
    } catch( SQLException se){  
        conn.rollback();  
        se.printStackTrace();  
    } finally {  
        try {  
            stmt.close();  
            conn.close();  
        } catch ( SQLException e ) {  
            e.printStackTrace();  
        }  
    }  
}
```

[실습] chap15\WEmpUpdate.java



```
public static void main( String[] args ) {  
    Connection conn = null;  
    PreparedStatement pstmt = null;  
    String url = "jdbc:oracle:thin:@192.168.0.200:1521:VCC";  
  
    try {  
        Class.forName( "oracle.jdbc.driver.OracleDriver" );  
        conn = DriverManager.getConnection( url , "student01" , "student01" );  
        conn.setAutoCommit( false );  
  
        String query = "UPDATE EMP SET LAST_NAME = ? WHERE ID = ? ";  
        pstmt = conn.prepareStatement( query );  
        pstmt.setString( 1, "HITE2" );  
        pstmt.setString( 2, "10005" );  
        pstmt.executeUpdate();  
        conn.commit();  
    } catch( ClassNotFoundException ce){  
        .....  
    } finally {  
        .....  
        pstmt.close();  
        .....  
    }  
}
```



15.1 JDBC 개요

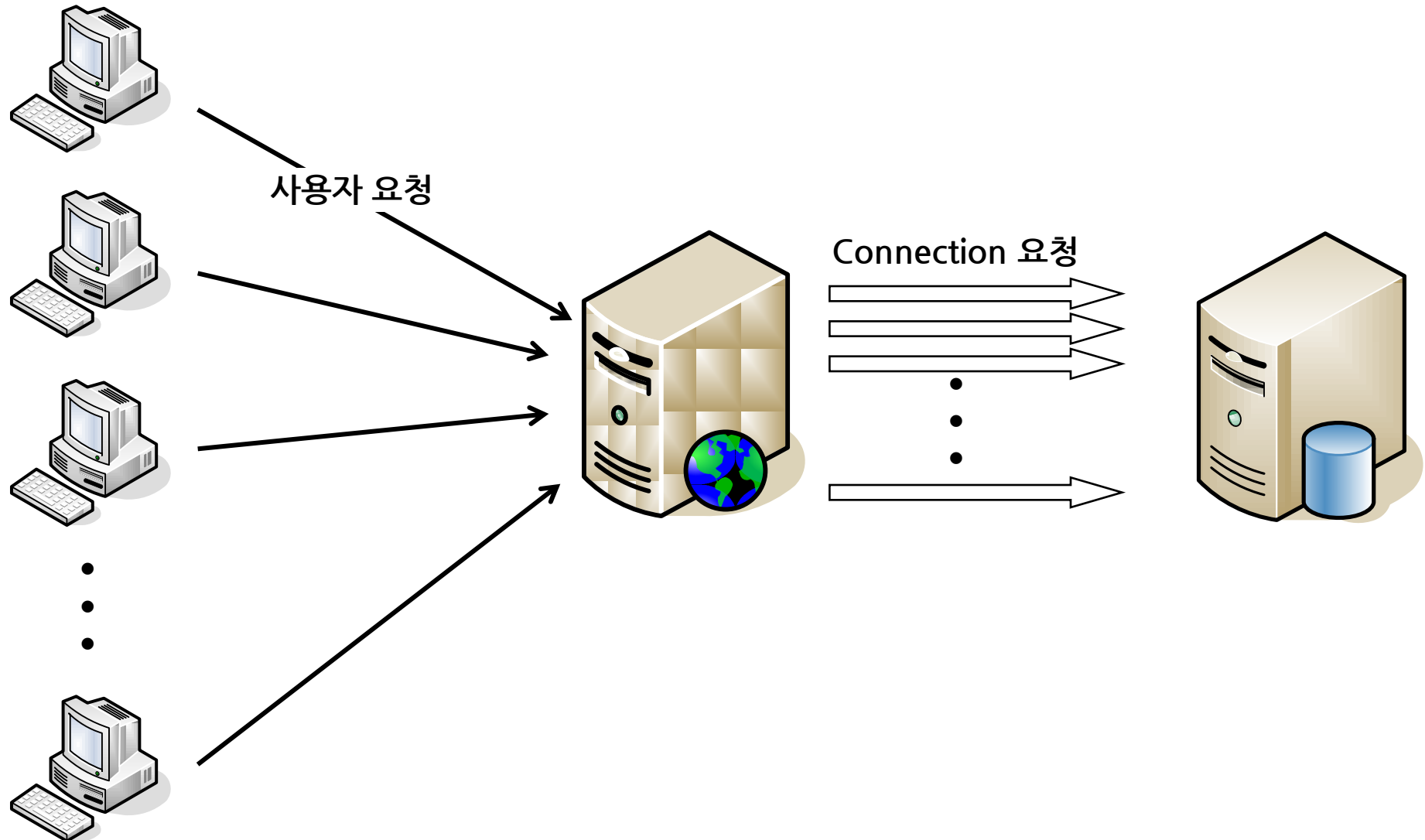
15.2 JDBC 사용

15.3 SELECT/UPDATE

15.4 Statement/PreparedStatement

 **15.5 Connection Pool**

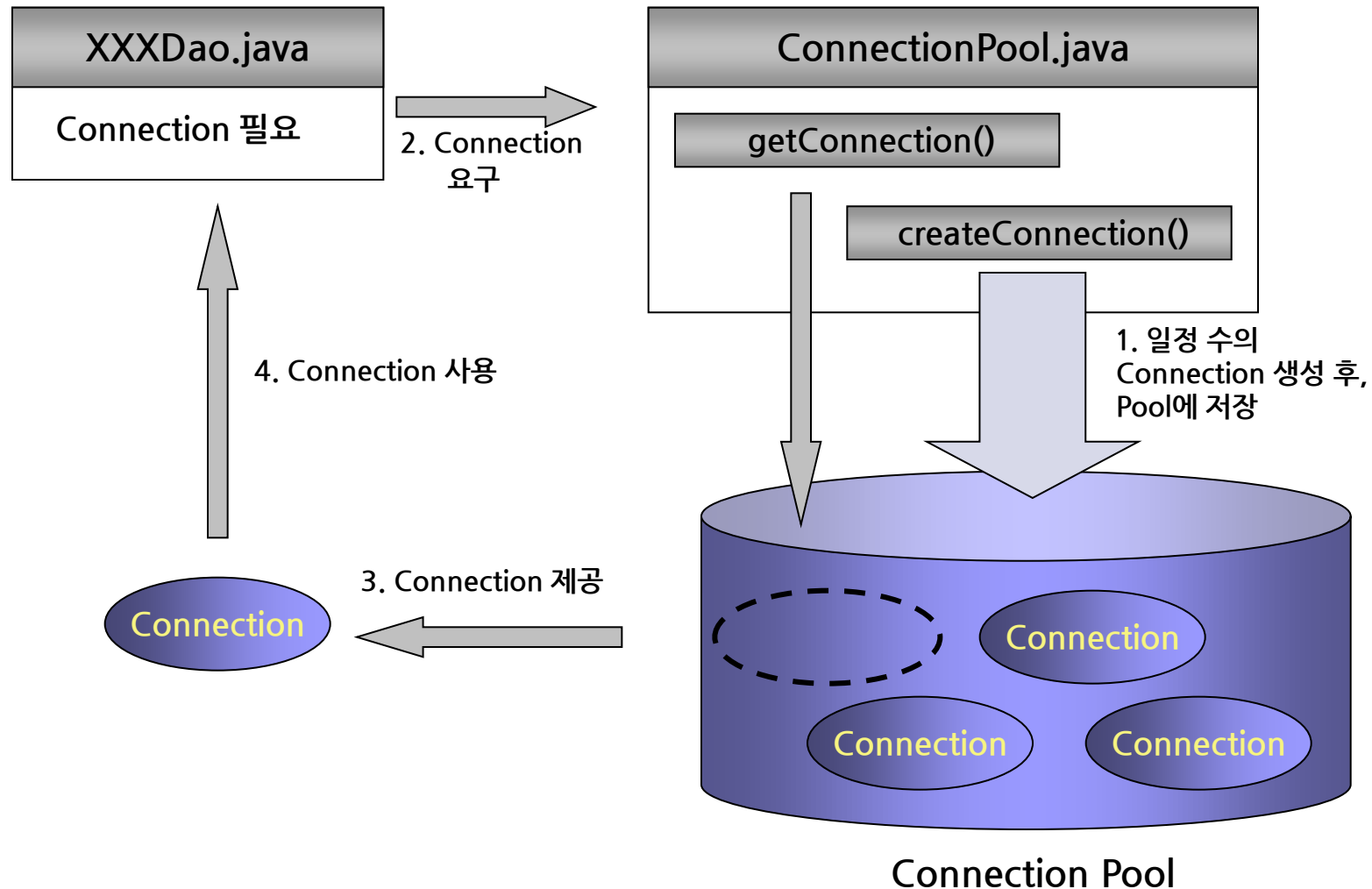
❖ 동시에 100명의 사용자가 요청한다면?



### ❖ DB Connection Pooling

- Pooling 기법이란, 미리 데이터베이스 Connection을 여러 개 만들어서 특정 공간에 저장해 놓고, 여러 사용자가 필요할 때 마다 하나씩 꺼내서 사용하고 다시 집어 넣는 방식을 말한다.
- Pooling 기법
  1. Connection을 미리 생성해서 보관
  2. Connection에 대한 요청이 들어오면, 보관 중인 Connection 중 하나를 넘겨줌
  3. 사용이 끝난 Connection을 다시 보관
- Connection Pooling의 장점
  1. 속도 향상
  2. 자원의 효율적인 활용
  3. Connection 객체 수 제어 가능

### ❖ Connection 객체의 보관과 사용



### ❖ Connection 객체의 반환 및 저장

