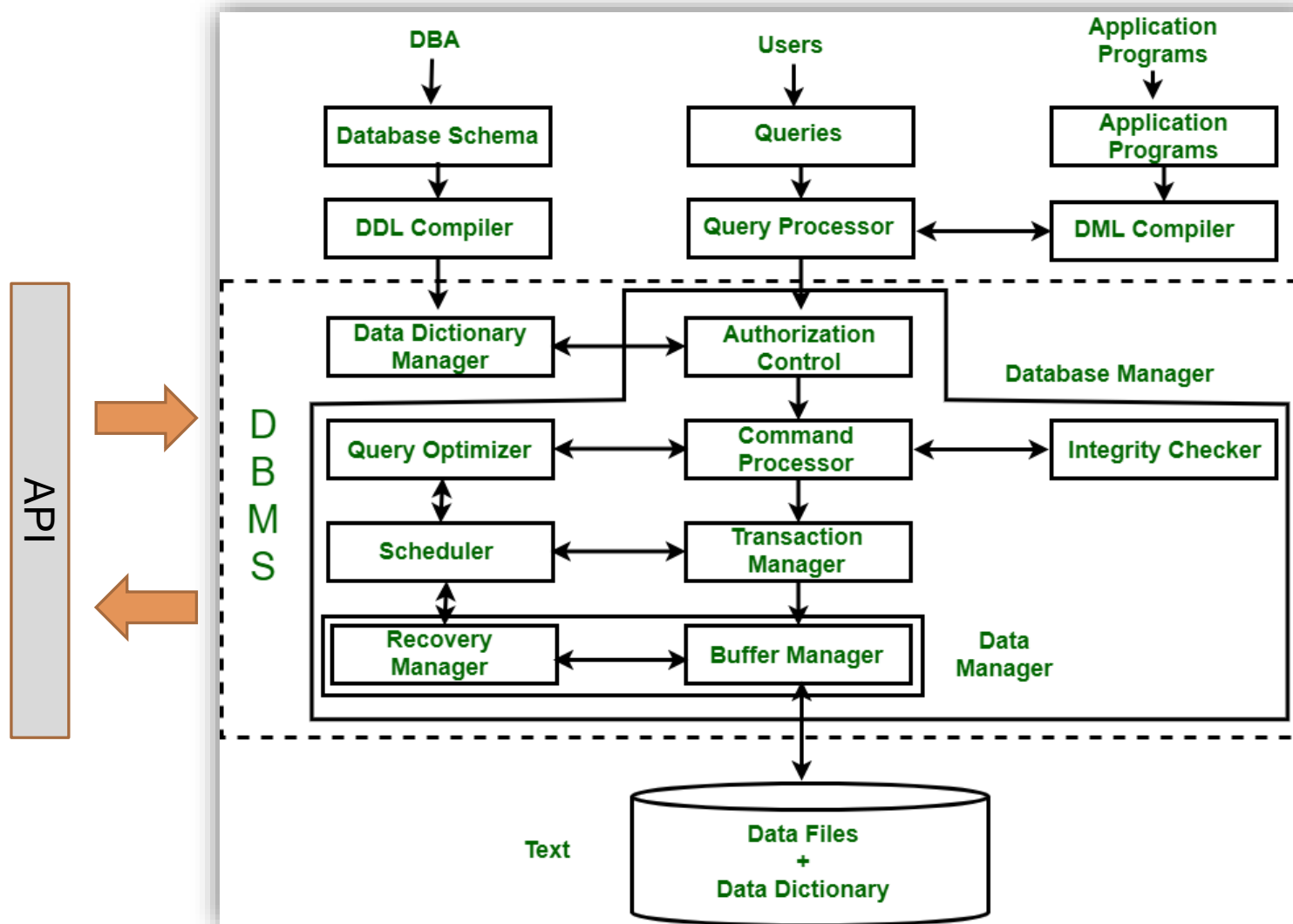


JDBC SQL 프로그래밍

지능물류빅데이터연구소 이상현

DBMS



이미지 출처 : <https://www.geeksforgeeks.org/structure-of-database-management-system/>

DBMS API

Oracle (Pro*C)

```
int main() {
    /* 데이터베이스 연결 */
    EXEC SQL CONNECT :username IDENTIFIED BY :password USING db_alias;

    /* 커서 생성 */
    EXEC SQL DECLARE emp_cursor CURSOR FOR
        SELECT emp_id, emp_name, salary FROM employees;

    /* 커서 열기 */
    EXEC SQL OPEN emp_cursor;

    struct Employee emp;

    /* 질의 결과 읽기 */
    while (1) {
        EXEC SQL FETCH emp_cursor INTO :emp;

        if (sqlca.sqlcode!=0) break; /* 더 이상 읽을 행이 없으면 */

        /* Process the fetched data */
        printf("Employee ID: %d, Name: %s, Salary: %.2f\n",
            emp.emp_id, emp.emp_name, emp.salary);
    }

    /* 커서 닫기 */
    EXEC SQL CLOSE emp_cursor;

    /* 데이터베이스 연결 해제 */
    EXEC SQL COMMIT WORK RELEASE;

    return 0;
}
```

개발 소스간 상호 호환 불가.

프로젝트 진행 중 DBMS 변경 시 비용 증가.

MySQL

```
int main() {
    /* 데이터베이스 연결 */
    MYSQL *conn = mysql_init(NULL);
    mysql_real_connect(conn, "localhost", "username", "password", "dbname", 0, NULL, 0);

    /* 질의 실행 */
    mysql_query(conn, "SELECT emp_id, emp_name, salary FROM employee");

    /* 질의 결과 저장 */
    MYSQL_RES *result = mysql_store_result(conn);

    /* 질의 결과 읽기 */
    while (1) {
        MYSQL_ROW row = mysql_fetch_row(result);

        if (row == NULL) break; /* 더 이상 읽을 행이 없으면 */

        sscanf(row[0], "%d", &emp.emp_id);
        sscanf(row[1], "%s", emp.emp_name);
        sscanf(row[2], "%f", &emp.salary);
        printf("Employee ID: %d, Name: %s, Salary: %.2f\n",
            emp.emp_id, emp.emp_name, emp.salary);
    }

    /* 데이터베이스 연결 해제 */
    mysql_close(conn);

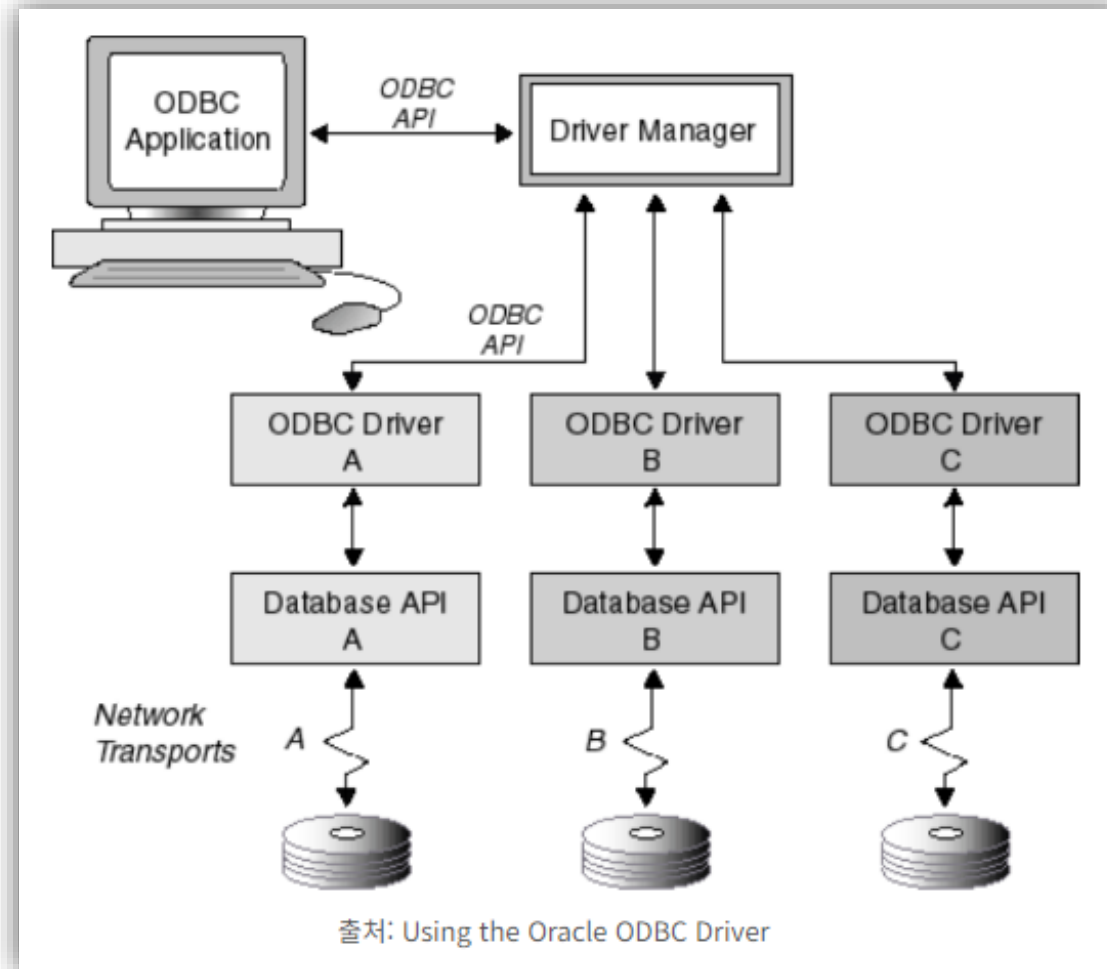
    return 0;
}
```

해제 */
result);

ODBC - Open Database Connectivity

- ODBC 드라이버

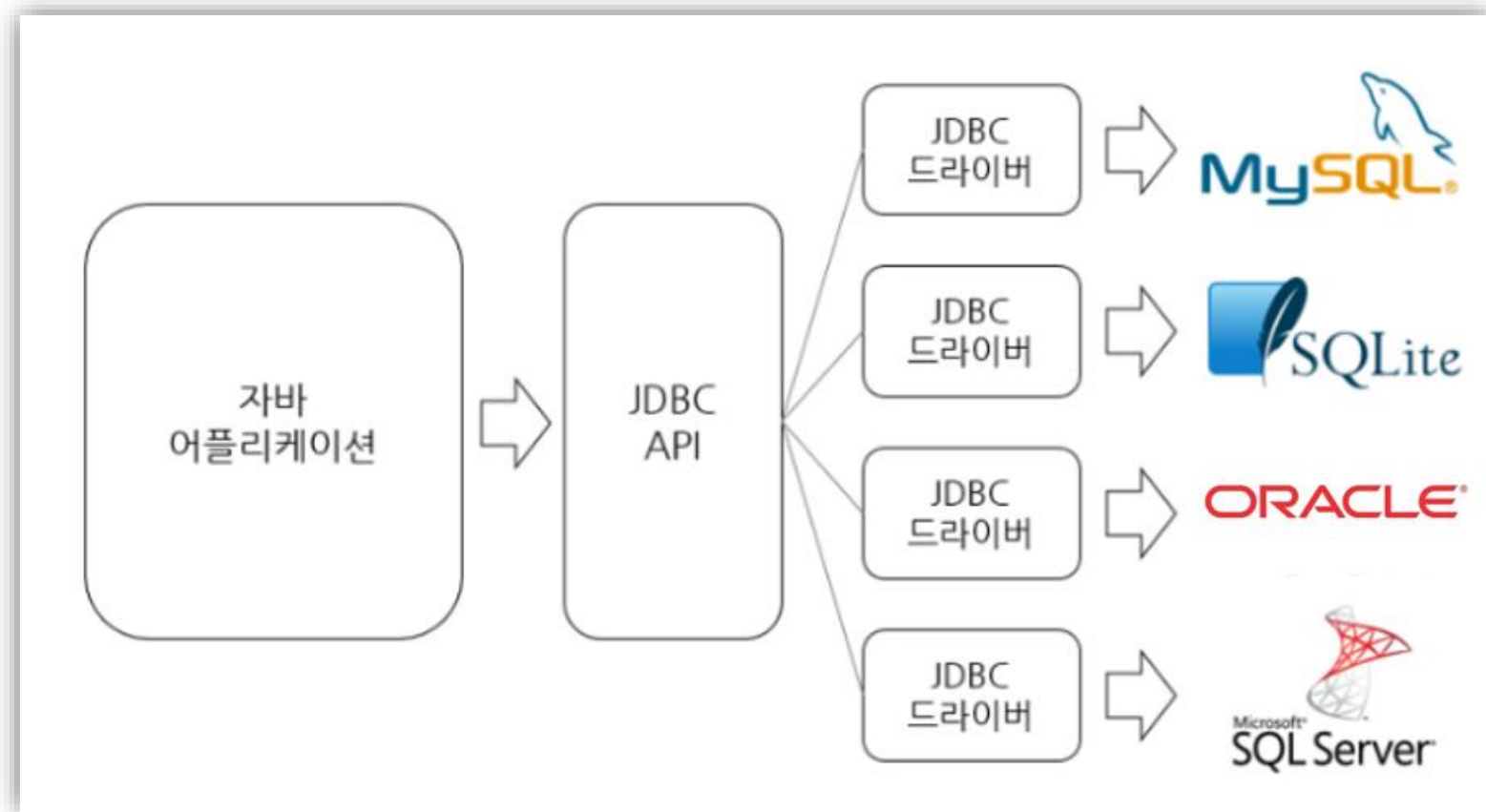
- DATABASE를 이용하기 위한 **표준 API**를 정해두고 DATABASE Vendor들이 API를 구현한 드라이버



JDBC - Java Database Connectivity

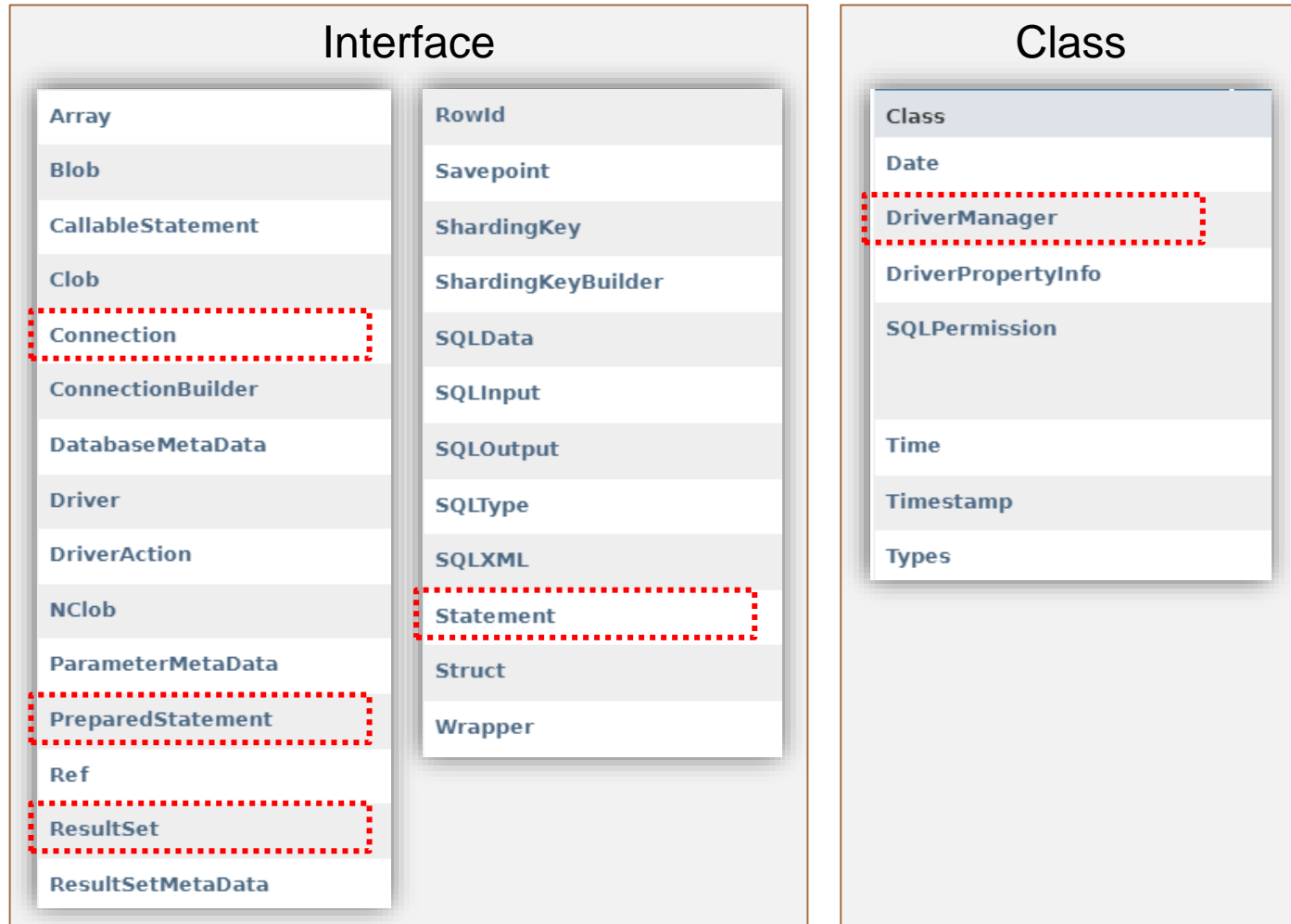
- JDBC 드라이버

- DATABASE를 이용하기 위한 **자바 표준 API**를 정해두고 각 DATABASE Vendor들이 API를 Java 언어로 구현한 드라이버



JDBC Interface

- Module java.sql
- Package java.sql
- DB 기능 처리 객체
- 주로 interface들 사용



<https://docs.oracle.com/en/java/javase/21/docs/api/java.sql/java/sql/package-summary.html#class-summary>

JDBC Driver

- java.sql의 인터페이스들을 상속하여 구현한 파일
- 데이터베이스 제작사에서 제공
- 데이터베이스별 드라이버



mysql-connector-java-8.4.0.jar

c:\Program Files (x86)\MySQL\Connector J 8.0



h2-2.1.214.jar

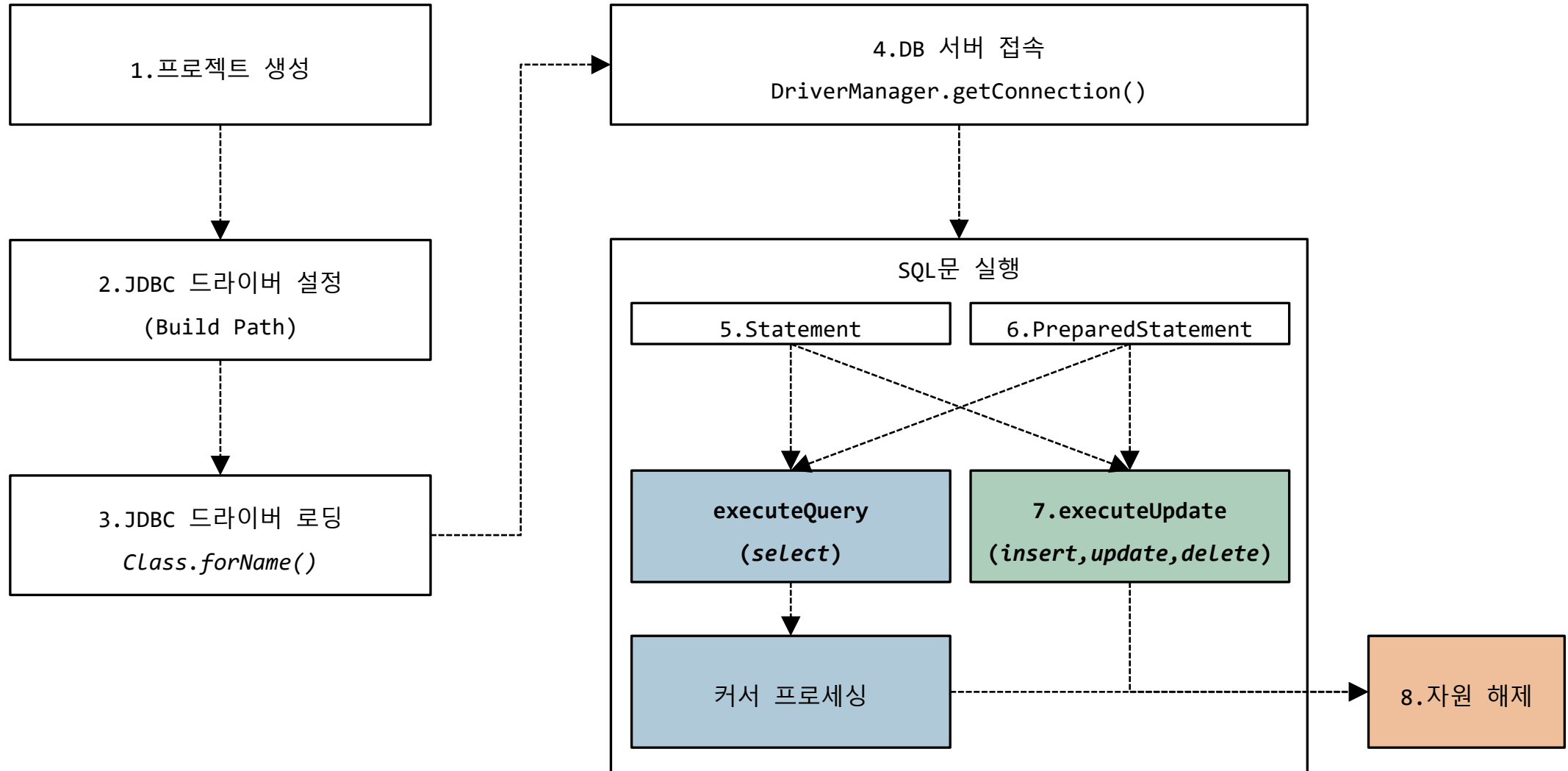
c:\Program Files (x86)\H2\bin



ojdbc6.jar

c:\oracle\app\oracle\product\11.2.0\server\jdbc\lib

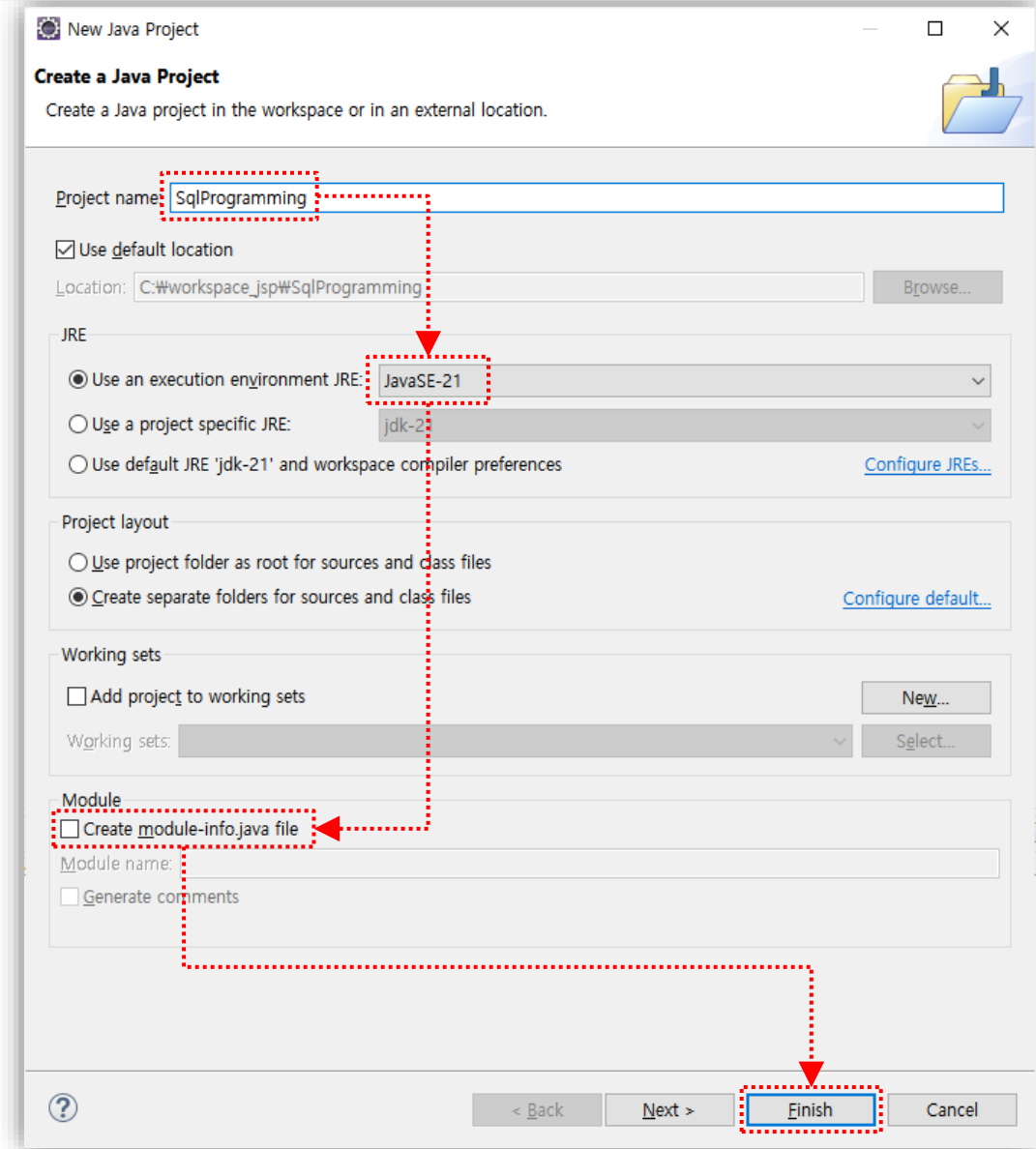
JDBC SQL Programming 구현 순서



1. 프로젝트 생성

○ 프로젝트 생성

- 프로젝트 명 : SqlProgramming
- JRE 버전 확인
- Create module-info.java file
 - 체크 해제



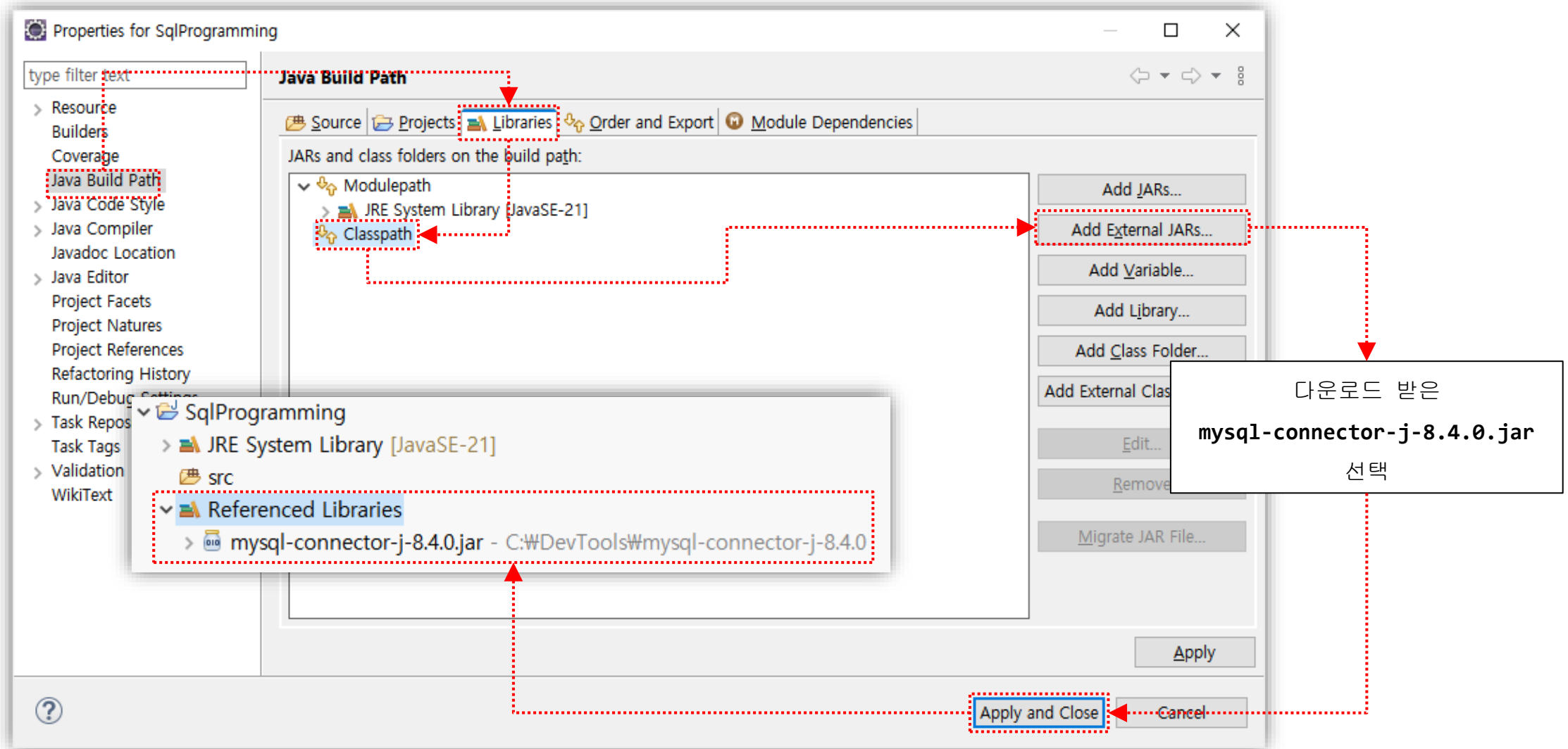
2. JDBC 드라이버 설정

- MySQL JDBC 드라이버 다운로드

- ① 웹 브라우저를 열어 MySQL 홈페이지에 접속 `https://www.mysql.com`
- ② 상단 메뉴에서 [Downloads]를 클릭
- ③ 화면 하단 [MySQL Community (GPL) Downloads >>]를 클릭
- ④ [Connector/J]를 클릭
- ⑤ [Archives] 탭을 클릭
- ⑥ Product version: 8.4, Operating System: Platform Independent
- ⑦ [Platform Independent(Architecture Independent), ZIP Archive] Download
- ⑧ 다운받은 파일의 압축을 풀어서 C:\DevTools에 복사

2. JDBC 드라이버 설정

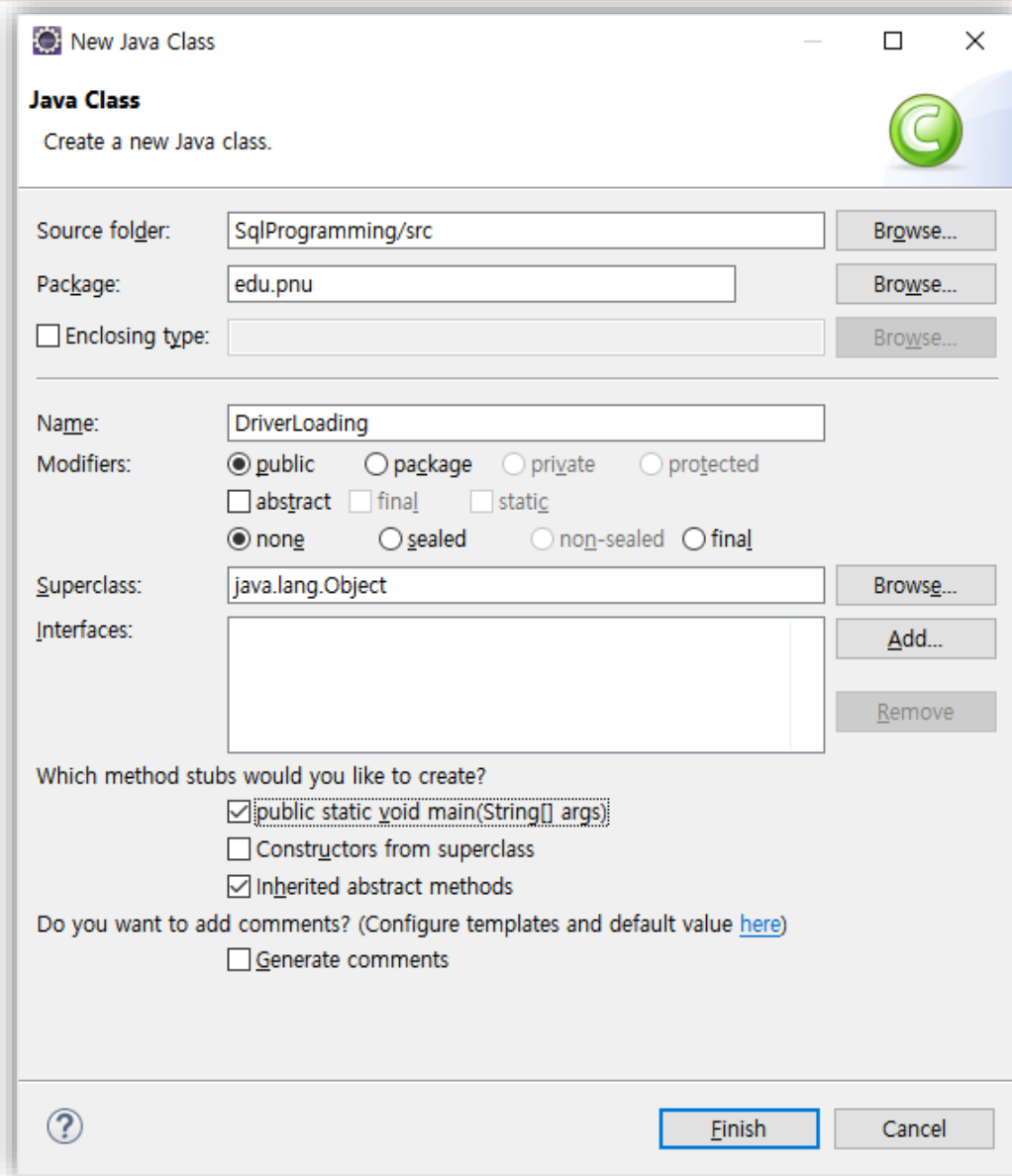
- 드라이버 설정 방법 : [Build Path] - [Configure Build Path...]



3. JDBC 드라이버 로딩

- `java.lang.Class` 의 `forName()` 메서드 사용
 - `static Class<?> forName(string className)`
 - `forName()` 메서드의 인자 값으로 `Driver`를 상속하는 클래스이름을 지정
- 사용 예
 - MySQL인 경우 : `Class.forName("com.mysql.cj.jdbc.Driver");`
 - H2인 경우 : `Class.forName("org.h2.Driver");`
 - Oracle인 경우 : `Class.forName("oracle.jdbc.driver.OracleDriver");`
- Java 6(JDBC 4.0) 부터 자바 프로젝트에서는 명시적으로 지정하지 않아도 가능
 - 하지만, 프로젝트에 따라 오류가 나는 경우가 있으므로 지정하는 것이 오류를 줄일 수 있음.

3. JDBC 드라이버 로딩



New Java Class

Java Class
Create a new Java class.

Source folder: SqlProgramming/src Browse...

Package: edu.pnu Browse...

☐ Enclosing type: Browse...

Name: DriverLoading

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☒ public static void main(String[] args);

☐ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

Finish Cancel

edu.pnu.DriverLoading.java

```
package edu.pnu;

public class DriverLoading {

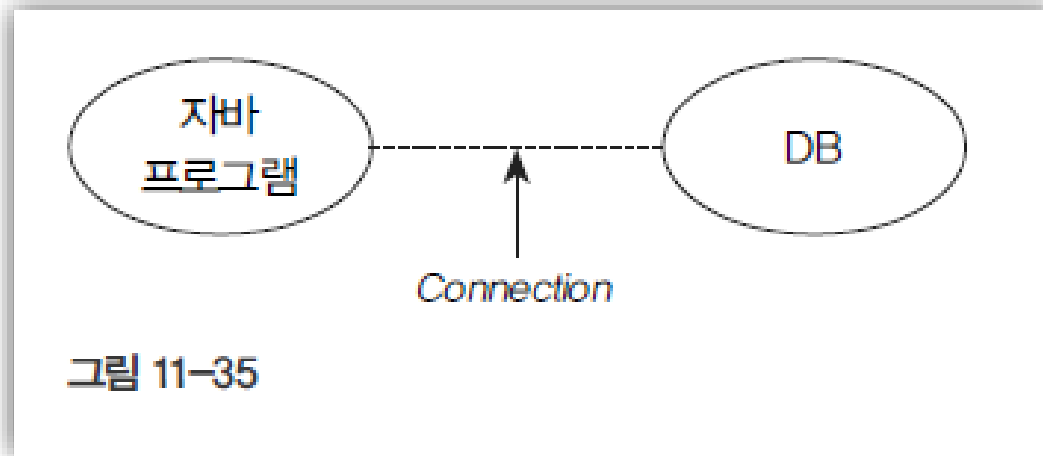
    public static void main(String[] args) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            System.out.println("로딩 성공");
        } catch (Exception e) {
            System.out.println("로딩 실패 : " + e.getMessage());
        }
    }
}
```

4.DB 서버 접속

- `java.sql.DriverManager`를 이용해서 연결 객체 생성
 - `static Connection getConnection(String url, String username, String password)`
 - `url` → 접속할 DB 서버의 URL 지정
 - `username` → 로그인할 DB 계정
 - `password` → 로그인할 DB 계정의 비밀번호
- 사용 예
 - `String url = "jdbc:mysql://localhost:3306/world";`
 - `String username = "scott";` musthave
 - `String password = "tiger";`
 - `Connection con = DriverManager.getConnection(url, username, password);`

4.DB 서버 접속

- Connection 객체
 - getConnection()의 리턴 값
 - DB 서버와의 연결 정보를 가지고 있는 객체



4.DB 서버 접속

edu.pnu.DriverLoading.java

```
package edu.pnu;

import java.sql.Connection;
import java.sql.DriverManager;

public class DBConnect {
    public static void main(String[] args) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");

            // static Connection getConnection(String url, String user, String password)
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/world", "scott", "tiger");
            System.out.println("연결 성공");
            con.close();
        } catch (Exception e) {
            System.out.println("연결 실패 : " + e.getMessage());
        }
    }
}
```


5. SQL문 실행 - Statement

◦ Statement

- Connection상에서 SQL문을 처리하는 객체
- Connection의 `createStatement()` 메서드를 이용해서 객체 생성
 - `Statement stmt = conn.createStatement();`
- Statement의 메서드 사용
 - `ResultSet executeQuery(String sql)` - select 질의
 - `int executeUpdate(String sql)` - select를 제외한 나머지 질의

5. SQL문 실행 - Statement

- ResultSet - select문의 질의 결과값을 가지는 객체

```
C:\Windows\system32\cmd.exe
C:\>mysql -uroot -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 276
Server version: 8.0.30 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use world;
Database changed
mysql> select * from city limit 10;
+----+-----+-----+-----+-----+
| ID | Name      | CountryCode | District | Population |
+----+-----+-----+-----+-----+
| 1  | Kabul     | AFG         | Kabul    | 1780000    |
| 2  | Qandahar  | AFG         | Qandahar | 237500     |
| 3  | Herat     | AFG         | Herat    | 186800     |
| 4  | Mazar-e-Sharif | AFG      | Balkh    | 127800     |
| 5  | Amsterdam | NLD         | Noord-Holland | 731200    |
| 6  | Rotterdam | NLD         | Zuid-Holland | 593321    |
| 7  | Haag      | NLD         | Zuid-Holland | 440900    |
| 8  | Utrecht   | NLD         | Utrecht  | 234323    |
| 9  | Eindhoven | NLD         | Noord-Brabant | 201843    |
| 10 | Tilburg   | NLD         | Noord-Brabant | 193238    |
+----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> exit;
Bye
C:\>
```

ResultSet rs

id	Name	CountryCode	District	Population
시작 빈행				
1	Kabul	AFG	Kabul	1780000
2	Qandahar	AFG	Qandahar	237500
...				
10	Tilburg	NLD	Noord-Brabant	193238
끝 빈행				

← cursor

rs.next() ↓

◆ Cursor Processing

- 질의 객체를 실행해서 얻은 질의 결과를 순서대로 읽는 것
- Cursor는 시작 빈행을 가리키고 있는 상태가 된다.
- rs.next() 메서드를 호출하면 다음 row의 데이터를 서버에서 읽어 오며, 메서드의 리턴값이 false가 될 때 까지 반복

5. SQL문 실행 - Statement

edu.pnu.QueryByStatement.java

```
package edu.pnu;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class QueryByStatment {
    public static void main(String[] args) {
        Connection con = null;
        Statement st = null;
        ResultSet rs = null;
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            String url = "jdbc:mysql://localhost:3306/world";
            con = DriverManager.getConnection(url, "scott", "tiger");
            st = con.createStatement();
            rs = st.executeQuery("select id, name, countrycode, "
                                + "district, population from city");
```

```
        while(rs.next()) {
            System.out.print(rs.getInt("id") + ",");
            System.out.print(rs.getString("name") + ",");
            System.out.print(rs.getString("countrycode") + ",");
            System.out.print(rs.getString("district") + ",");
            System.out.print(rs.getInt("population") + "\n");
        }
    } catch (Exception e) {
        System.out.println("연결 실패:" + e.getMessage());
    } finally {
        try {
            if (rs != null) rs.close();
            if (st != null) st.close();
            if (con != null) con.close();
        } catch (Exception e) {
            System.out.println("오류:" + e.getMessage());
        }
    }
}
```

5. SQL문 실행 - Statement

◉ ResultSet Methods

- **void afterLast()**
 - 끝 빈행으로 커서를 이동한다.
- **void beforeFirst()**
 - 시작 빈행으로 커서를 이동한다.
- **boolean next()**
 - 현재 커서 다음의 레코드 유무를 판단한다.
 - true인 경우 커서를 다음으로 이동시킨다.
- **XXX getXXX(String column명) / getXXX(int index)**
 - 커서가 위치한 레코드의 컬럼 값을 반환함 (XXX 는 데이터 타입), index를 사용할 경우 1부터 시작
 - `System.out.print(rs.getInt("id") + ",");`
 - `System.out.print(rs.getInt(1) + ",");`

<https://docs.oracle.com/en/java/javase/21/docs/api/java.sql/java/sql/ResultSet.html#method-summary>

Statement 실습 - QueryStatement.java

- ◆ 아래 질의들을 Scanner를 통해 입력 받아서 실행하는 형태로 구현하되, while 루프를 이용해서 질의 번호와 값을 입력 받아 반복적으로 질의를 수행할 수 있도록 구현하세요. 질의 번호가 0이면 프로그램을 종료합니다.
- ① 인구 수를 입력 받아서 그보다 많은 인구를 가진 도시를 출력하세요. (City)
- ② 국가 코드를 입력 받아서 해당 국가의 도시의 이름과 인구를 출력하세요. (City)
 - 국가 명의 일부를 입력 받아서 해당 국가의 도시의 이름과 인구를 출력하세요. (City, Country Join)
 - 국가 코드 또는 국가 명의 일부를 입력 받아서 해당 국가의 도시명과 인구 출력하세요. (City, Country Join)
- ③ 대륙을 입력 받아서 해당 대륙에 위치한 국가를 출력하세요. (Country.Continent)
- ④ 넓이를 입력 받아서 입력 값보다 작은 면적을 가진 국가의 이름과 면적을 출력하세요. (Country.SurfaceArea)
- ⑤ 대한민국의 District를 입력 받아서 해당 지역에 있는 모든 도시를 출력하세요. (예: 'Kyonggi', City)
- ⑥ 언어를 입력 받아서 해당 언어가 국가 공식 언어인 국가를 출력하세요. (예: 'Spanish', CountryLanguage)
 - 언어를 입력 받아서 해당 언어가 국가 공식 언어인 국가명을 출력하세요. (Country, CountryLanguage Join)
- ⑦ CountryLanguage에서 사용자가 입력 비율 이상인 언어의 국가 코드와 비율을 출력하세요.

6. SQL문 실행 - PreparedStatement

◦ PreparedStatement

- Connection상에서 SQL문을 처리하는 객체
- Connection의 `prepareStatement()` 메서드를 이용해서 객체 생성
- `PreparedStatement pstmt = conn.prepareStatement(queryString);`
- 파라미터로 입력되는 `queryString` 내에 "?" 기호를 사용하여 **미완성의 SQL문을 생성하여 사용**

6. SQL문 실행 - PreparedStatement

- void setXXX(int parameterIndex, XXX value)

↓
Date Type

↓
1부터 시작

↓
Date Type

// Statement를 사용한 코드 예제

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

```
ResultSet rs = stmt.executeQuery("select Continent,Name,Population from country where population > 100000000");
```

// PreparedStatement를 사용한 코드 예제 1

```
PreparedStatement pstmt = con.prepareStatement("select Continent,Name,Population from country where population > ?");
```

```
pstmt.setInt(1, 100000000);
```

```
ResultSet rs = pstmt.executeQuery();
```

// PreparedStatement를 사용한 코드 예제 2

```
PreparedStatement pstmt = con.prepareStatement("select Continent,Name,Population from country where name like ? and population > ?");
```

```
pstmt.setString(1, "A%");
```

```
pstmt.setInt(2, 100000000);
```

```
ResultSet rs = pstmt.executeQuery();
```

PreparedStatement 실습 – QueryPreparedStatement.java

◆ Statement 실습 예제를 PreparedStatement로 구현하세요.

- ① 인구 수를 입력 받아서 그보다 많은 인구를 가진 도시를 출력하세요. (City)
- ② 국가 코드를 입력 받아서 해당 국가의 도시의 이름과 인구를 출력하세요. (City)
 - 국가 명의 일부를 입력 받아서 해당 국가의 도시의 이름과 인구를 출력하세요. (City, Country Join)
 - 국가 코드 또는 국가 명의 일부를 입력 받아서 해당 국가의 도시명과 인구 출력하세요. (City, Country Join)
- ③ 대륙을 입력 받아서 해당 대륙에 위치한 국가를 출력하세요. (Country.Continent)
- ④ 넓이를 입력 받아서 입력 값보다 작은 면적을 가진 국가의 이름과 면적을 출력하세요.
(Country.SurfaceArea)
- ⑤ 대한민국의 District를 입력 받아서 해당 지역에 있는 모든 도시를 출력하세요. (예: 'Kyonggi', City)
- ⑥ 언어를 입력 받아서 해당 언어가 국가 공식 언어인 국가를 출력하세요. (예: 'Spanish', CountryLanguage)
 - 언어를 입력 받아서 해당 언어가 국가 공식 언어인 국가명을 출력하세요. (Country, CountryLanguage Join)
- ⑦ CountryLanguage에서 사용자가 입력 비율 이상인 언어의 국가 코드와 비율을 출력하세요.

7. SQL문 실행 - executeUpdate (insert)

- QueryExecuteUpdateInsert.java

- url을 "jdbc:mysql://localhost:3306/myfirstdb"로 변경;

```
// PreparedStatement 사용
String sql = "insert into phonebook(name, mobile) values(?,?)";
PreparedStatement psmt = con.prepareStatement(sql);
psmt.setString(1, name);
psmt.setString(2, mobile);
psmt.executeUpdate();
```

```
// Statement 사용
String sql = "insert into phonebook(name, mobile) values ";
Statement stmt = con.createStatement();
stmt.executeUpdate(sql + String.format("( '%s', '%s' )", name, mobile));
```

```
INSERT INTO `myfirstdb`.`phonebook` (`name`, `mobile`, `home`, `company`, `email`)
VALUES ('홍길동', '010-1111-2222', '051-123-4567', '051-123-1234', 'hongkd@korea.com');
```

7. SQL문 실행 - executeUpdate (update)

- QueryExecuteUpdateUpdate.java

```
// PreparedStatement 사용
String sql = "update phonebook set home=? where id=?";
PreparedStatement psmt = con.prepareStatement(sql);
psmt.setString(1, home);
psmt.setInt(2, id);
psmt.executeUpdate();
```

```
// Statement 사용
String sql = "update phonebook set ";
Statement stmt = con.createStatement();
stmt.executeUpdate(sql + String.format("home='%s' where id=%d", home, id));
```

```
UPDATE `myfirstdb`.`phonebook` SET `mobile` = '010-1111-9999' WHERE (`id` = '1');
```

7. SQL문 실행 - executeUpdate (delete)

- QueryExecuteUpdateDelete.java

```
// PreparedStatement 사용
String sql = "delete from phonebook where id=?";
PreparedStatement psmt = con.prepareStatement(sql);
psmt.setInt(1, id);
psmt.executeUpdate();
```

```
// Statement 사용
Statement stmt = con.createStatement();
stmt.executeUpdate("delete from phonebook " + String.format("where id=%d", id));
```

```
DELETE FROM `myfirstdb`.`phonebook` WHERE (`id` = '1');
```

7. SQL문 실행 - executeQuery (select)

- QueryExecuteQuery.java

```
// PreparedStatement 사용
String sql = "select * from phonebook where id > ?";
PreparedStatement psmt = con.prepareStatement(sql);
psmt.setInt(1, id);
ResultSet rs = psmt.executeQuery();
```

```
// Statement 사용
String sql = "select * from phonebook ";
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery(sql + String.format("where id > %d", id));
```

8. 자원 해제

- ◉ JDBC 프로그램 실행 시 사용했던 모든 객체 자원 해제
 - Resultset 해제
 - Statement 또는 PreparedStatement 해제
 - Connection 해제
- ◉ 해제 메서드 : `close()`
 - `con.close()`
 - `stmt.close()` 또는 `pstmt.close()`
 - `con.close()`
 - try-catch-finally 또는 try-with-resource(p545) 이용 가능

9. 통합 실습 - 코드를 완성하세요.

edu.pnu.PhoneBookApp.java

```
public class PhoneBookApp {  
    private static Scanner sc = new Scanner(System.in);  
    private static String url = "jdbc:mysql://localhost:3306/myfirstdb";  
    public static void main(String[] args) throws Exception {  
        Connection con = DriverManager.getConnection(url, "scott", "tiger");  
        boolean flag = true;  
        while(flag) {  
            System.out.print("[I]nsert/[U]pdate/[D]elete/[S]elect/[N]ative/[Q]uit:");  
            char c = sc.next().toUpperCase().charAt(0);  
            switch(c) {  
                case 'I': insertPhonebook(con);        break;  
                case 'U': updatePhonebook(con);        break;  
                case 'D': deletePhonebook(con);        break;  
                case 'S': selectALLPhonebook(con);     break;  
                case 'N': nativeQuery(con);            break;  
                case 'Q': flag = false;                break;  
            }  
        }  
        System.out.println("Bye~");  
    }  
}
```

추가

사용자 계정 생성 및 권한 설정

- root 권한으로 Workbench 로그인
- [Navigator] - [Administration Tab] - [Users and Privileges]

User

User	From Host
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
root	localhost

Add Account Delete Refresh

Login Account Limits Administrative Roles Schema Privileges

Login Name: musthave

Authentication Type: Standard

Limit to Hosts Matching: %

Password: *****

Weak password.

Confirm Password: *****

Expire Password

Login Name : musthave
Password : tiger

사용자 계정 생성

Schema Privileges

Schema Privileges

Schema and Host fields may use % and _ wildcards.
The server will match specific entries before wildcarded ones.

Revoke All Privileges Delete Entry Add Entry...

권한 설정

New Schema Privilege Definition

Select the Schema for which the user 'newuser' will have the privileges you want to define.

Schema

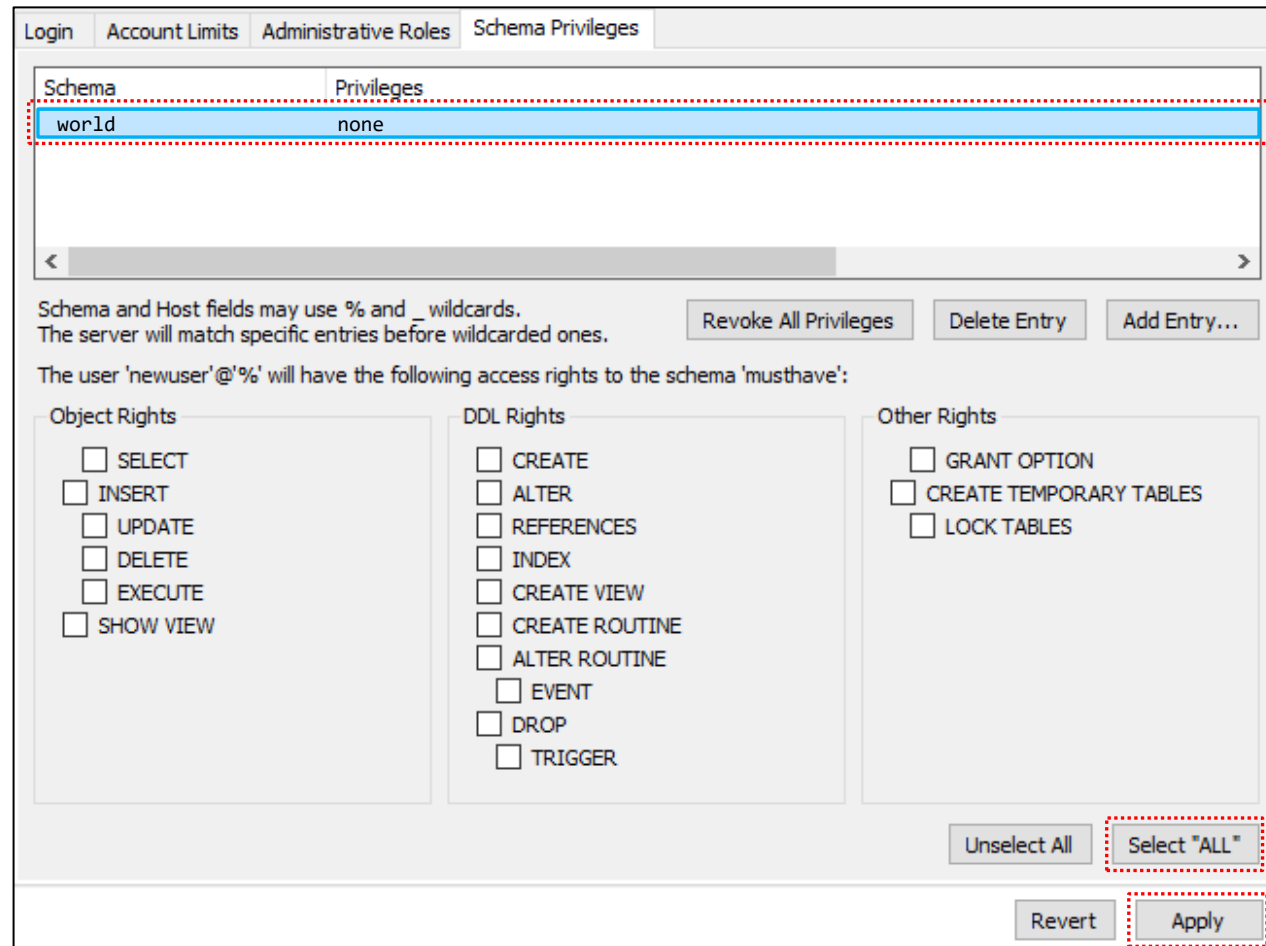
☐ All Schema (%) This rule will apply to any schema name.

☐ Schemas matching pattern: This rule will apply to schemas that match the given name or pattern. You may use _ and % as wildcards in a pattern. Escape these characters with \ in case you want their literal value.

☒ Selected schema: world Select a specific schema name for the rule to apply to.

Cancel OK

사용자 계정 생성 및 권한 설정



The image shows the 'Schema Privileges' tab in a MySQL configuration window. At the top, there are tabs for 'Login', 'Account Limits', 'Administrative Roles', and 'Schema Privileges'. Below these, a table lists schema privileges. The first row, 'world' with 'none' privileges, is highlighted in blue and enclosed in a red dashed box. Below the table, a text box explains that Schema and Host fields can use wildcards. To the right of this text are three buttons: 'Revoke All Privileges', 'Delete Entry', and 'Add Entry...'. Below this is a text box stating: 'The user 'newuser'@'%' will have the following access rights to the schema 'musthave':'. This is followed by three columns of checkboxes for 'Object Rights', 'DDL Rights', and 'Other Rights'. At the bottom right, there are buttons for 'Unselect All', 'Select "ALL"', 'Revert', and 'Apply'. The 'Select "ALL"' button is highlighted with a red dashed box.

Schema	Privileges
world	none

Schema and Host fields may use % and _ wildcards.
The server will match specific entries before wildcarded ones.

The user 'newuser'@'%' will have the following access rights to the schema 'musthave':

Object Rights

- ☐ SELECT
- ☐ INSERT
- ☐ UPDATE
- ☐ DELETE
- ☐ EXECUTE
- ☐ SHOW VIEW

DDL Rights

- ☐ CREATE
- ☐ ALTER
- ☐ REFERENCES
- ☐ INDEX
- ☐ CREATE VIEW
- ☐ CREATE ROUTINE
- ☐ ALTER ROUTINE
- ☐ EVENT
- ☐ DROP
- ☐ TRIGGER

Other Rights

- ☐ GRANT OPTION
- ☐ CREATE TEMPORARY TABLES
- ☐ LOCK TABLES

Unselect All Select "ALL" Revert Apply