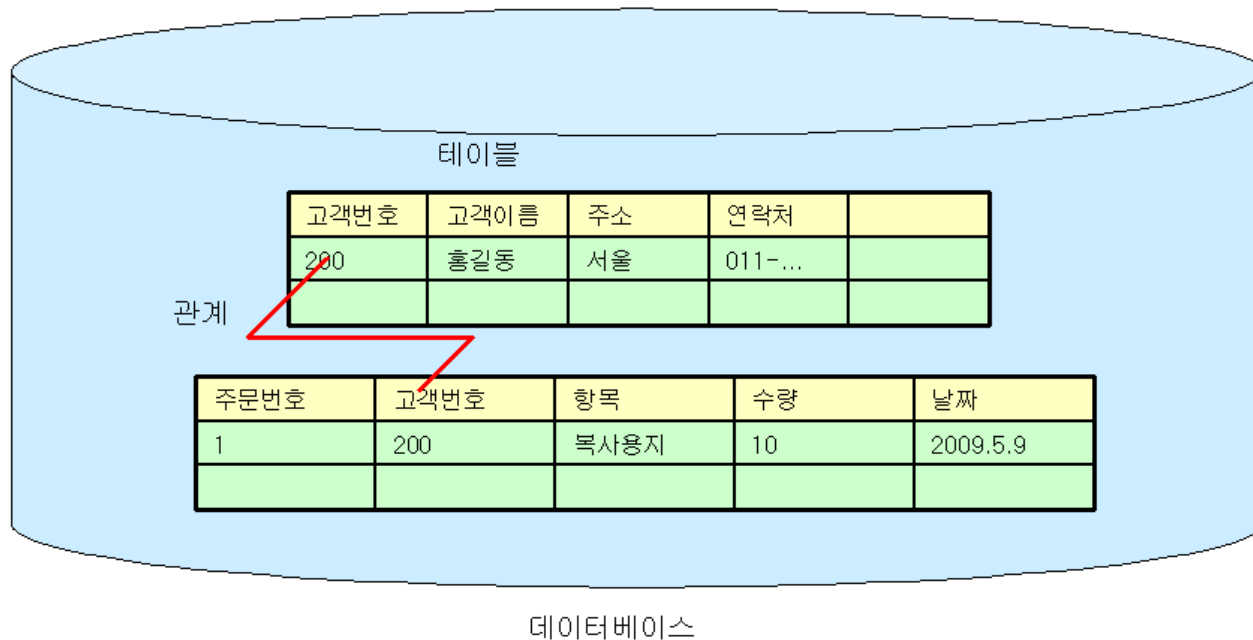


데이터베이스 프로그래밍

데이터베이스란?

- 관계형 데이터베이스(database)는 데이터를 여러 개의 테이블에 나누어서 저장한다.
- 가장 많이 사용되는 DBMS
 - ▣ 오라클, 마이크로소프트의 SQL Server, 사이베이스, MySQL



테이블

컬럼(column)

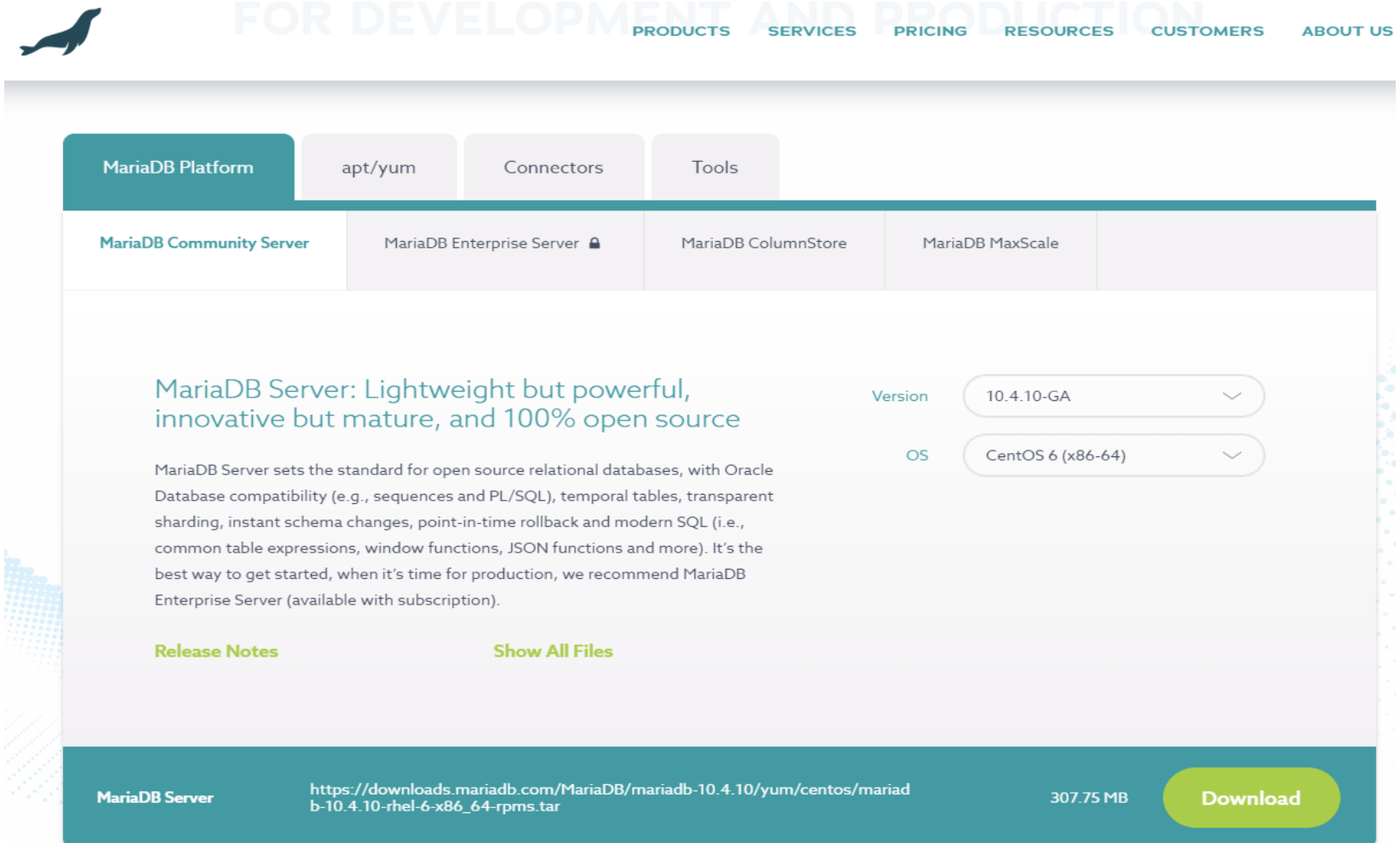
행(row)
또는
레코드(record)

book_id	title	publisher	year	price
1	Operating System Concepts	Wiley	2003	30700
2	Head First PHP and MYSQL	O'Reilly	2009	58000
3	C Programming Language	Prentice-Hall	1989	35000
4	Head First SQL	O'Reilly	2007	43000


MariaDB 설치

4

□ <https://mariadb.com/downloads/>




The screenshot shows the MariaDB website's download section. At the top, there's a navigation bar with links: PRODUCTS, SERVICES, PRICING, RESOURCES, CUSTOMERS, and ABOUT US. Below this, a secondary navigation bar highlights 'MariaDB Platform' and includes links for 'apt/yum', 'Connectors', and 'Tools'. The main content area features a grid of product options: 'MariaDB Community Server' (selected), 'MariaDB Enterprise Server' (with a lock icon), 'MariaDB ColumnStore', and 'MariaDB MaxScale'. The 'MariaDB Community Server' section is expanded, showing a description: 'MariaDB Server: Lightweight but powerful, innovative but mature, and 100% open source'. It also includes a paragraph about its features and a recommendation for the Enterprise Server. To the right of the text are two dropdown menus: 'Version' set to '10.4.10-GA' and 'OS' set to 'CentOS 6 (x86-64)'. At the bottom, a teal bar contains the product name 'MariaDB Server', the download URL 'https://downloads.mariadb.com/MariaDB/mariadb-10.4.10/yum/centos/mariadb-10.4.10-rhel-6-x86_64-rpms.tar', the file size '307.75 MB', and a prominent green 'Download' button.

 FOR DEVELOPMENT AND PRODUCTION

PRODUCTS SERVICES PRICING RESOURCES CUSTOMERS ABOUT US

MariaDB Platform apt/yum Connectors Tools

MariaDB Community Server MariaDB Enterprise Server  MariaDB ColumnStore MariaDB MaxScale

MariaDB Server: Lightweight but powerful, innovative but mature, and 100% open source

MariaDB Server sets the standard for open source relational databases, with Oracle Database compatibility (e.g., sequences and PL/SQL), temporal tables, transparent sharding, instant schema changes, point-in-time rollback and modern SQL (i.e., common table expressions, window functions, JSON functions and more). It's the best way to get started, when it's time for production, we recommend MariaDB Enterprise Server (available with subscription).

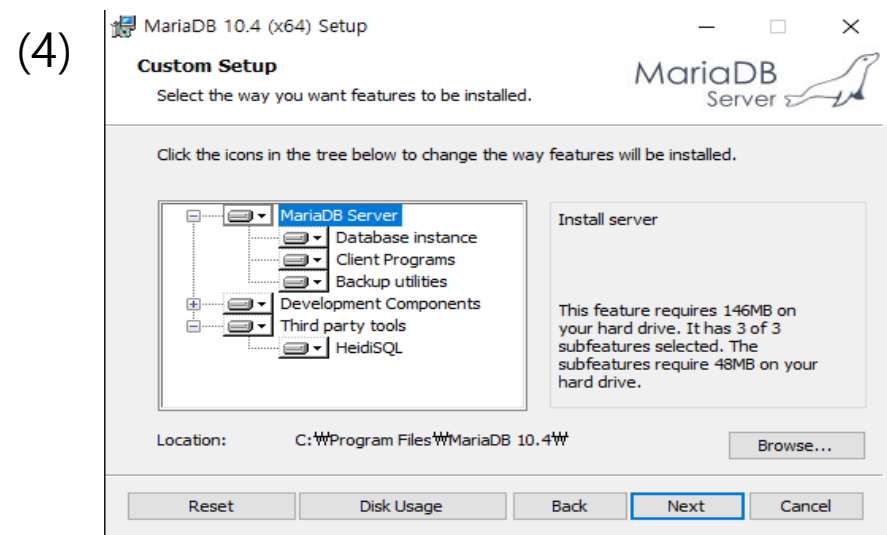
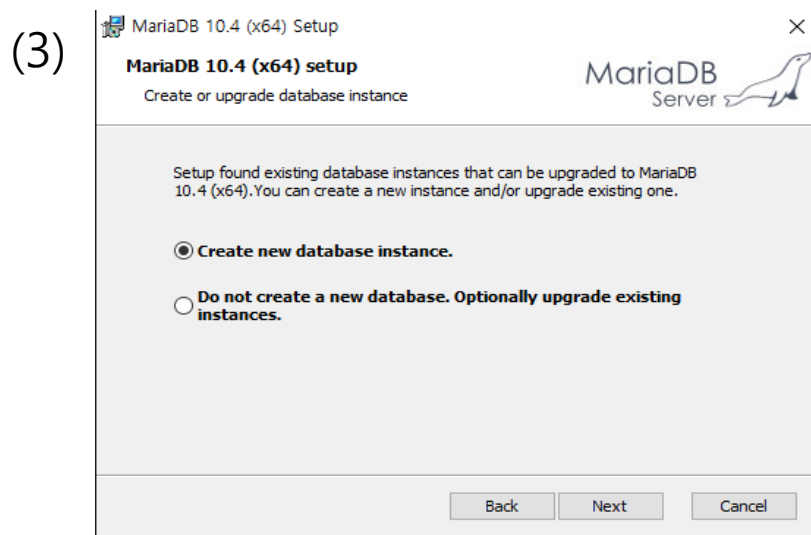
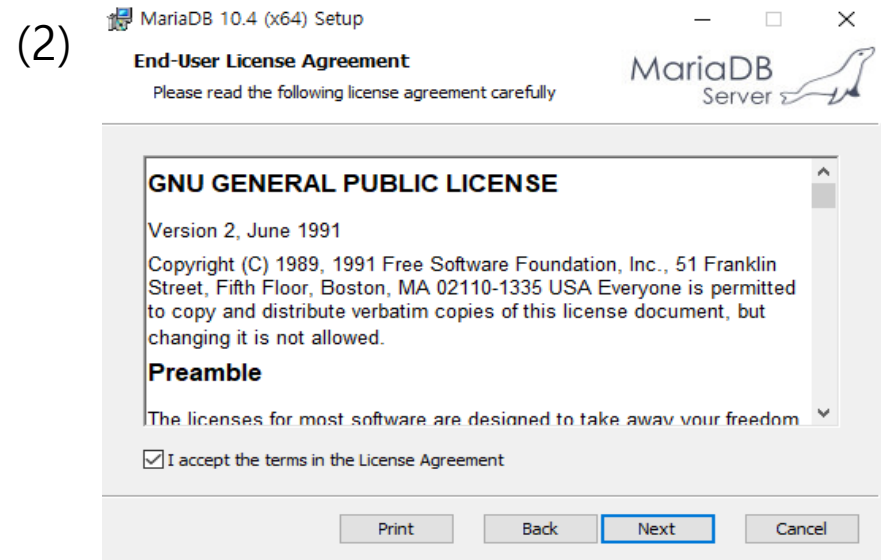
Version 10.4.10-GA

OS CentOS 6 (x86-64)

Release Notes Show All Files

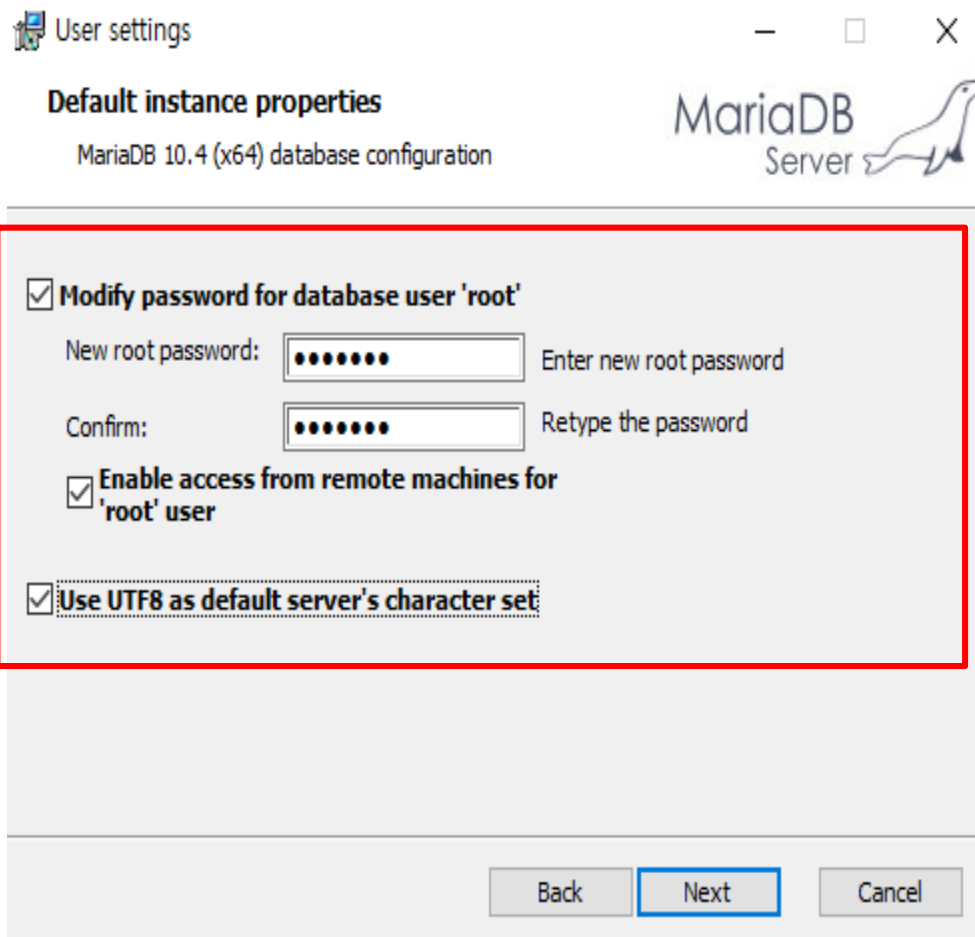
MariaDB Server https://downloads.mariadb.com/MariaDB/mariadb-10.4.10/yum/centos/mariadb-10.4.10-rhel-6-x86_64-rpms.tar 307.75 MB [Download](#)

MariaDB 설치하기



MariaDB 설치하기

□ 비밀번호 설정 및 원격접속 허용



The image shows a Windows-style window titled "User settings" for "MariaDB 10.4 (x64) database configuration". The window has a title bar with standard minimize, maximize, and close buttons. The main content area is divided into sections. The first section, "Default instance properties", includes the MariaDB logo and a seal icon. Below this, a red rectangular box highlights the "Modify password for database user 'root'" section. This section contains two password input fields labeled "New root password:" and "Confirm:", both filled with dots. Below these are two checked checkboxes: "Enable access from remote machines for 'root' user" and "Use UTF8 as default server's character set". At the bottom of the window, there are three buttons: "Back", "Next" (which is highlighted with a blue border), and "Cancel".

User settings

Default instance properties

MariaDB 10.4 (x64) database configuration

MariaDB Server

☒ Modify password for database user 'root'

New root password: Enter new root password

Confirm: Retype the password

☒ Enable access from remote machines for 'root' user

☒ Use UTF8 as default server's character set

Back Next Cancel

MariaDB 설치하기

Database settings

Default instance properties
MariaDB 10.4 (x64) database configuration

☒ **Install as service**
Service Name:

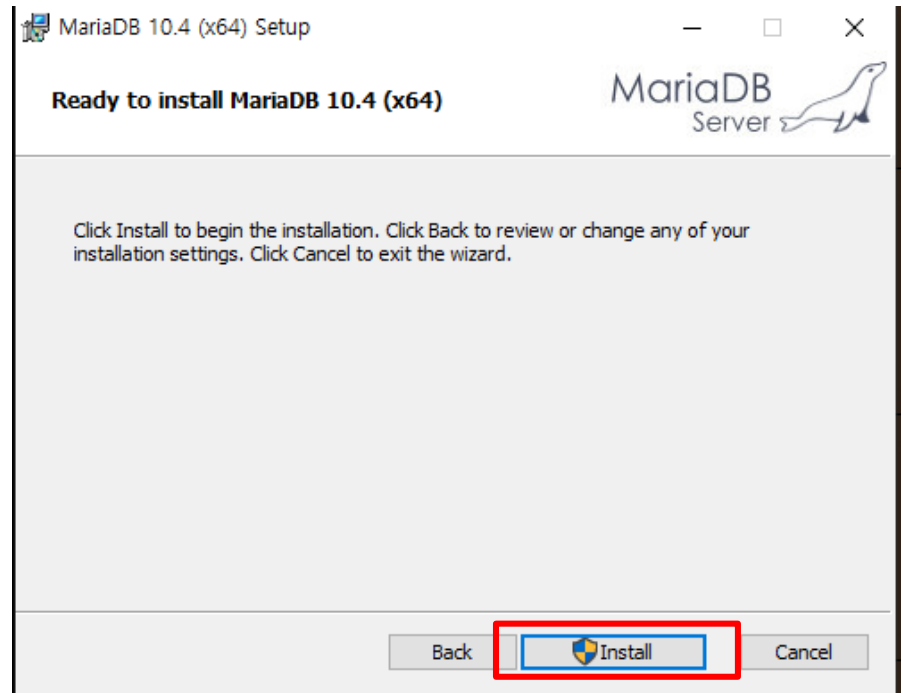
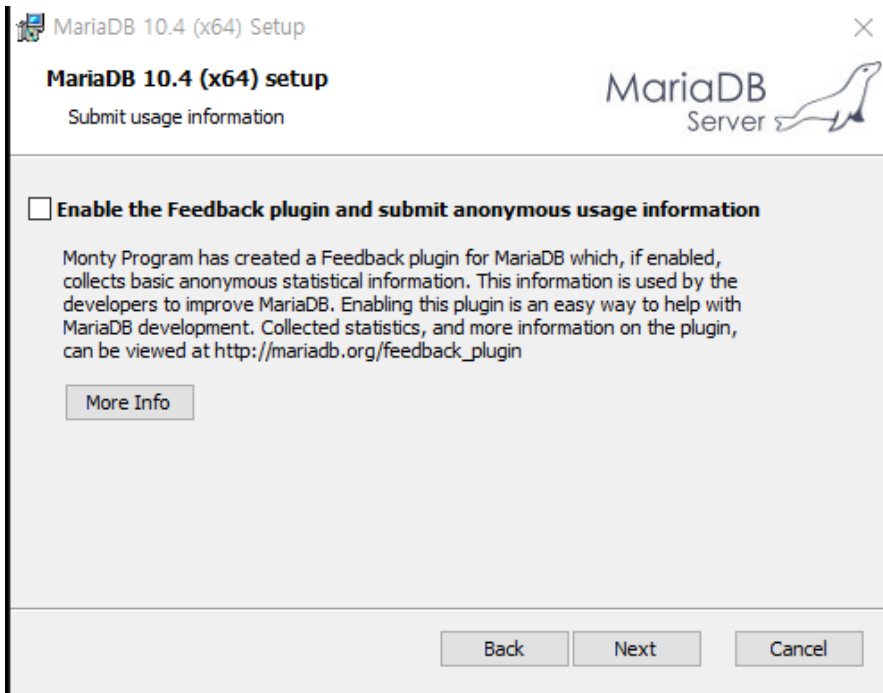
☒ **Enable networking**
TCP port:

InnoDB engine settings
Buffer pool size: MB
Page size: KB

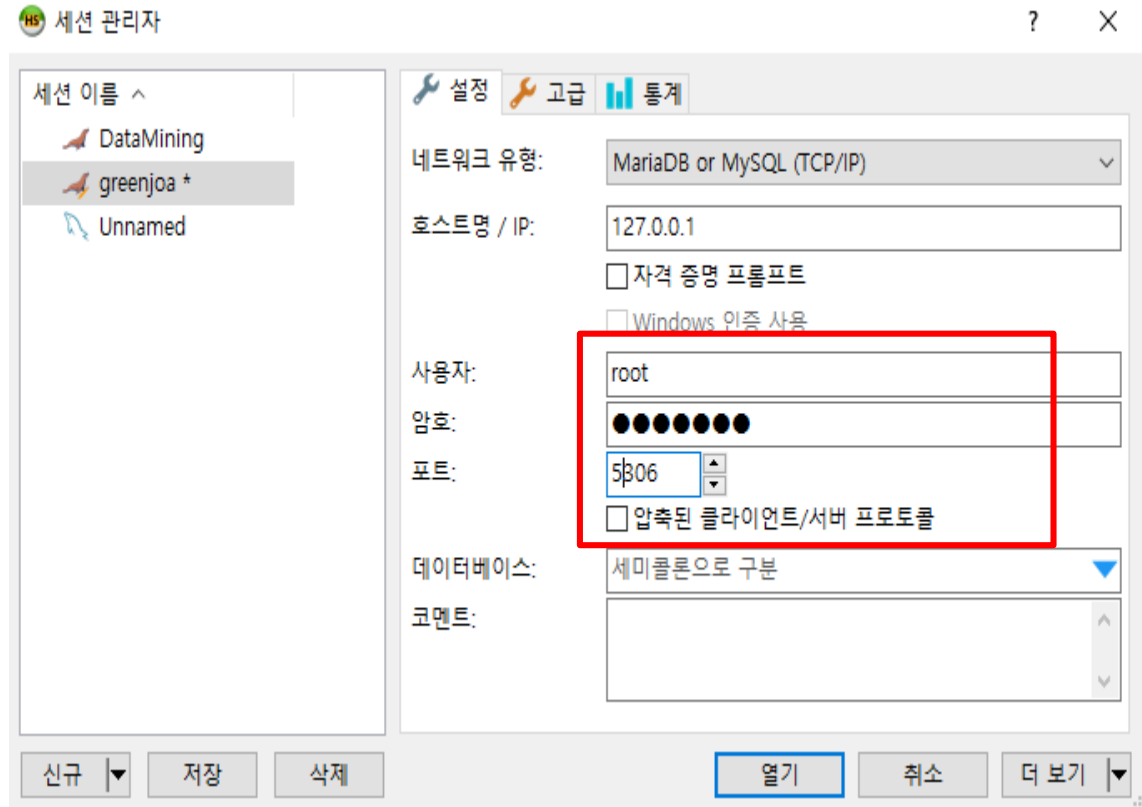
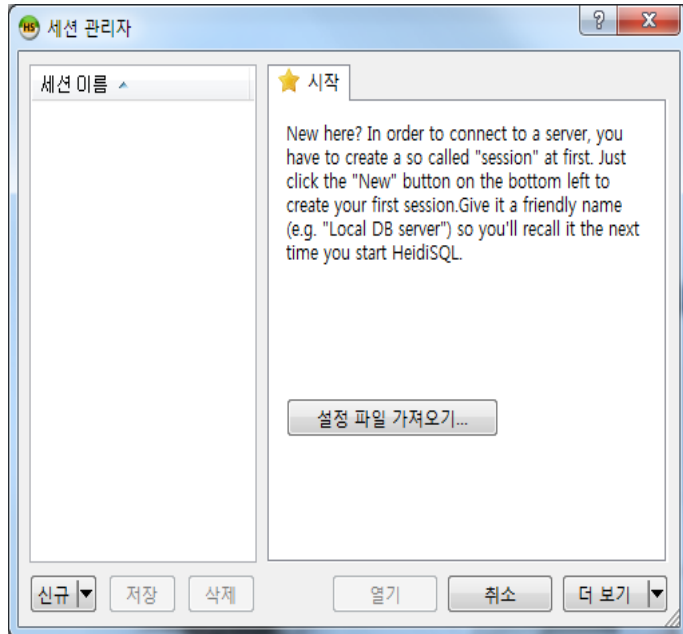
Back Next Cancel

다른 DBMS 가 설치된 경우 변경

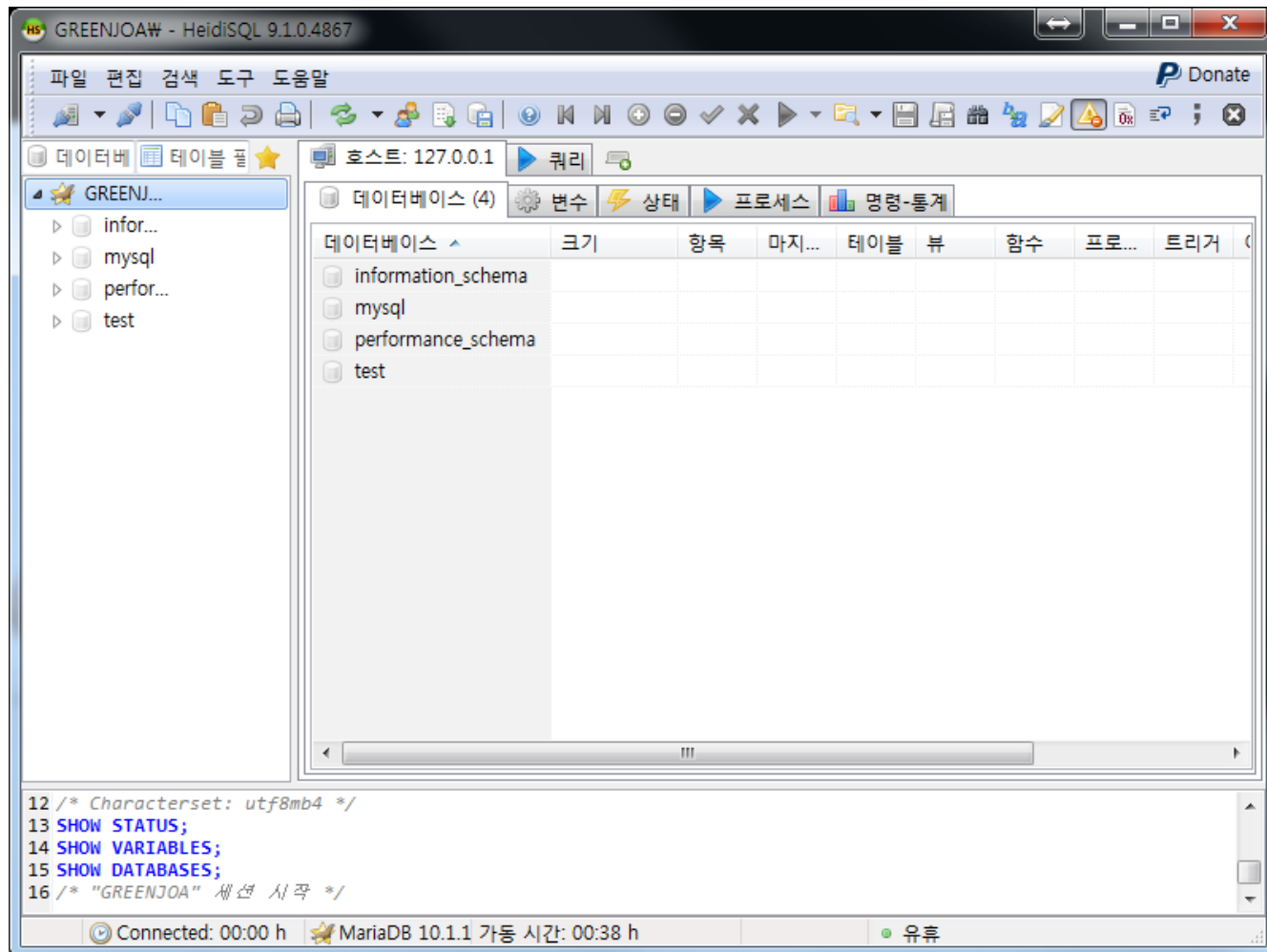
MariaDB 설치하기



HeidiSQL



HeidiSQL



DB 생성

11

- 팝업메뉴 >> 새로 생성 >> 데이터베이스

데이터베이스 생성...

이름(N) mydb

조합(O) utf8_general_ci

서버 기본값: utf8_general_ci

확인 취소

CREATE 코드:

```
CREATE DATABASE `mydb` /*!40100 COLLATE 'u'
```

테이블 생성

12

□ DB>>팝업메뉴>>새로 생성 >> 테이블

greenjoa#mydb#student# - HeidiSQL 10.2.0.5599

파일 편집 검색 도구 ...로 이동 도움말

호스트: 127.0.0.1 데이터베이스: mydb 테이블: student 데이터 쿼리

기본 옵션 인덱스 외래 키 분할 CREATE 코드 ALTER 코드

이름: student **테이블 이름**

코멘트:

열: **+** 추가 **-** 제거 ▲ 위로 ▼ 아래로

#	이름	데이터 유형	길이/설정	부호 없음	NULL 허용	0으로 채움	기본값
1	name	VARCHAR	50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
2	kor	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
3	eng	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

도움말 되돌리기 저장

X 필터: 정규 표현식

```
46 SHOW TRIGGERS FROM `mydb`;
47 SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM information_schema.`EVENTS` WHERE `EVENT_SCHEMA`='mydb';
48 SHOW CREATE TABLE `mydb`.`student`;
49 /* "greenjoa" 세션 시작 */
50 SHOW CREATE TABLE `mydb`.`student`;
```

연결됨: 00:39 MariaDB 10.4.10 가동 시간: 00:41 h 서버 시간: 오기 유후

데이터 저장

13

□ 데이터 탭에서 행 추가 후 데이터 입력

파일 편집 검색 도구 ...로 이동 도움말

데이터베이스 필터 테이블 필터

호스트: 127.0.0.1 데이터베이스: mydb 테이블: student 데이터 쿼리

greenjoa

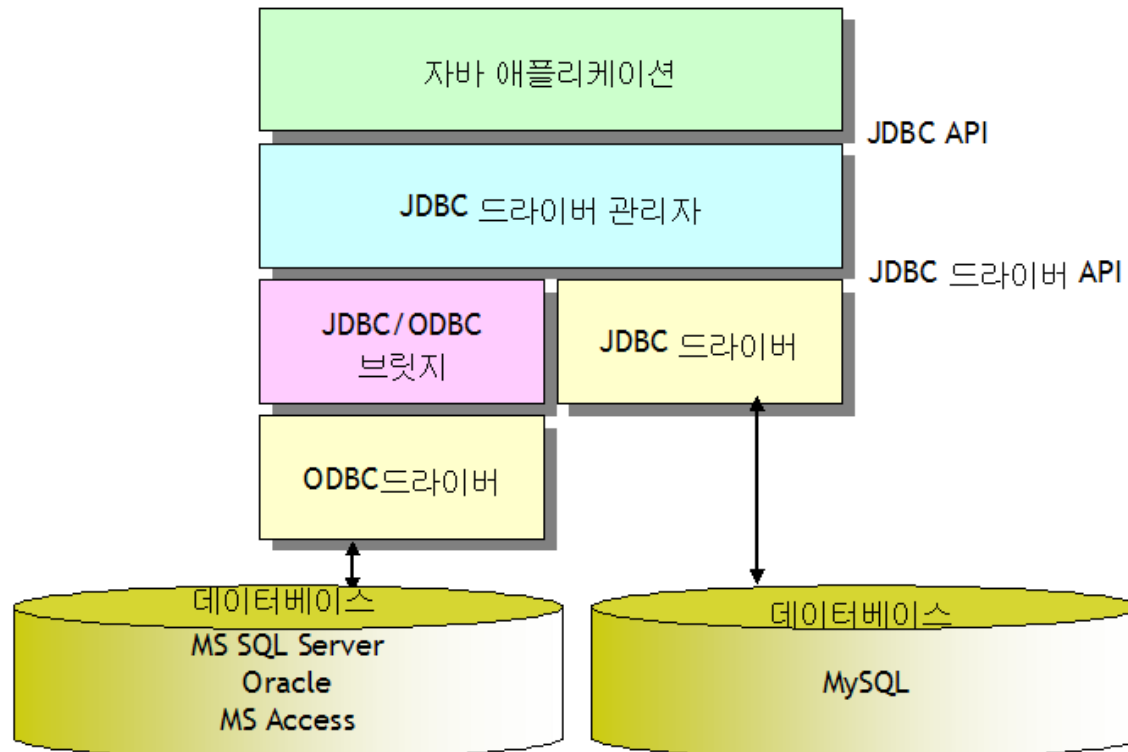
- information_schema 192.0 KiB
- mydb 16.0 KiB
 - student 16.0 KiB
 - mysql
 - performance_schema
 - test

mydb.student: 4 행 (총) (대략적)

name	kor	eng
홍길동	89	80
김길동	75	60
이길동	60	50
고길동	100	94

자바와 데이터베이스

- JDBC(Java Database Connectivity)는 자바 API의 하나로서 데이터베이스에 연결하여서 데이터베이스 안의 데이터에 대하여 검색하고 데이터를 변경할 수 있게 한다.



데이터베이스 프로그램 개발 절차

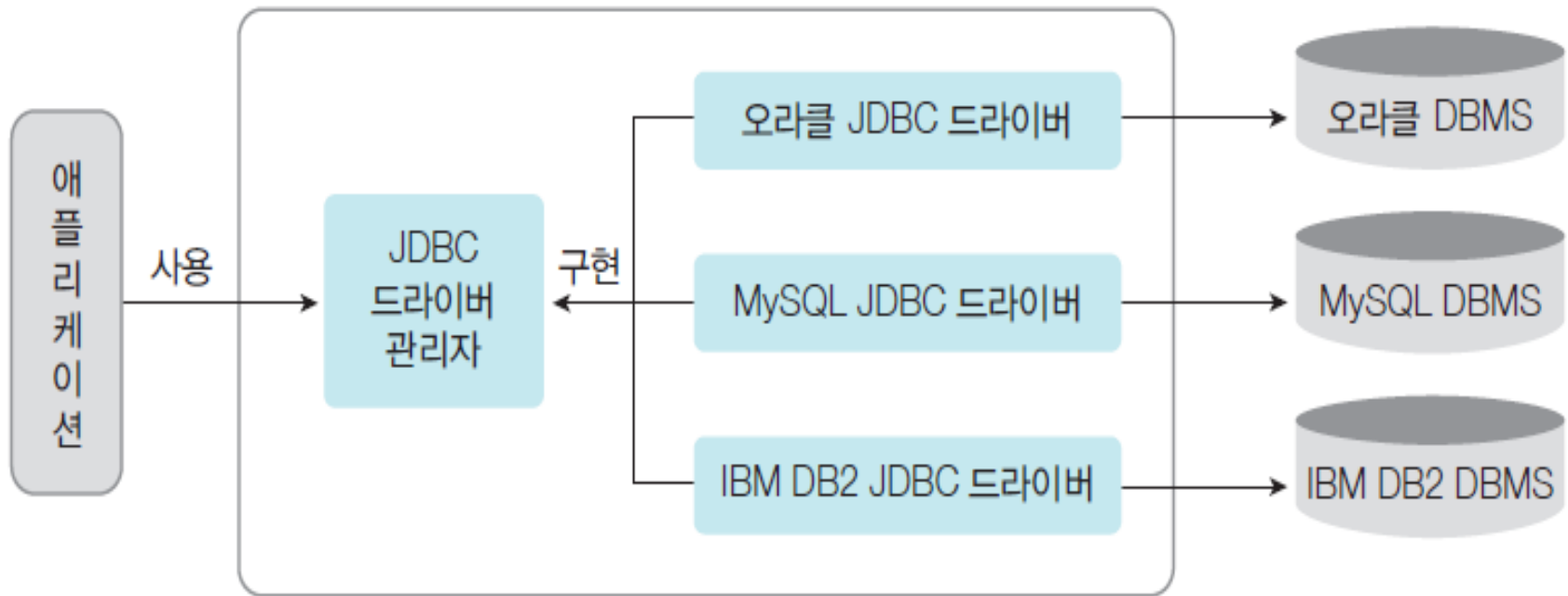
- ① DBMS(DataBase Management System)를 설치
- ② 자신이 설치한 DBMS에 필요한 JDBC 드라이버를 다운로드
- ③ JDBC가 제공하는 기능을 이용하여 데이터베이스 응용 프로그램을 개발



JDBC를 통하여
데이터베이스에
연결되면 그 다음
에는 **SQL** 명령어
를 데이터베이스
에 전달하면 됩니
다.

JDBC : Java Database Connectivity

16



JDBC 설치 및 연결 설정

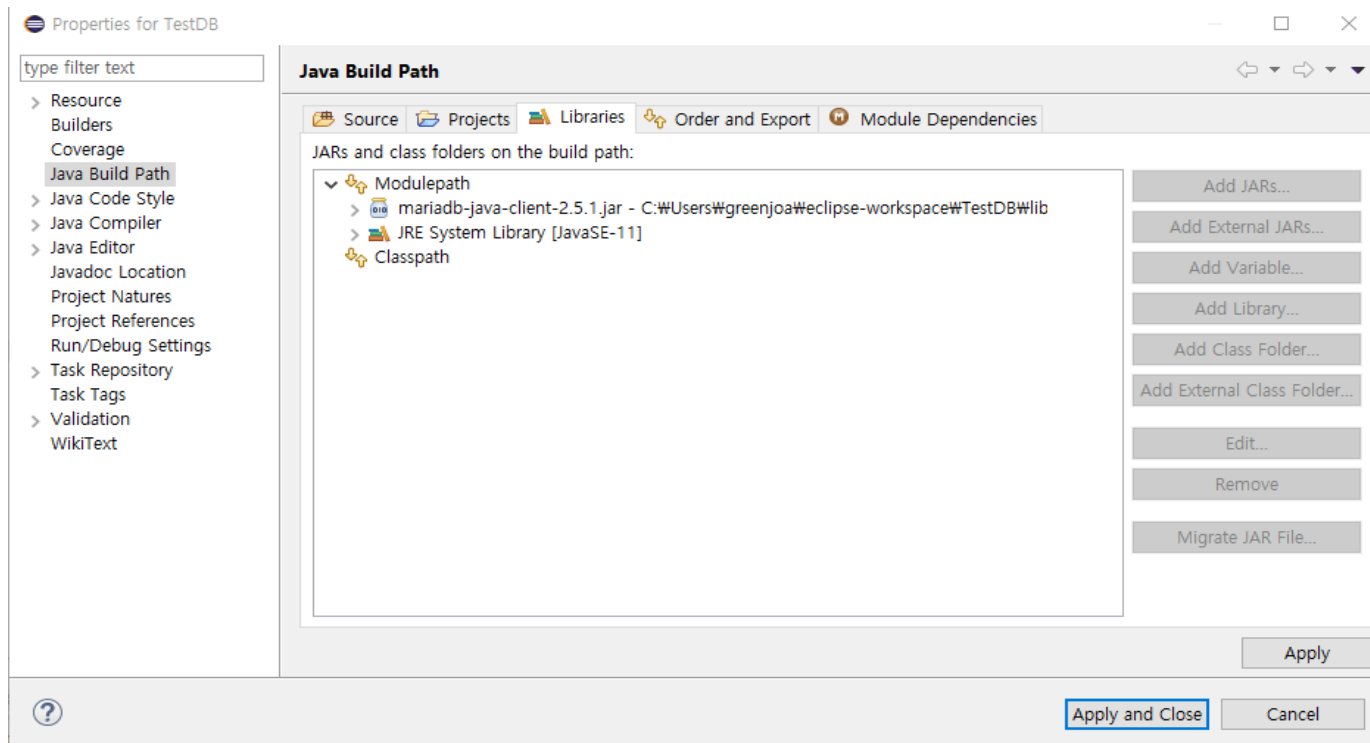
□ JDBC의 설치 및 연결 설정 방법

▣ JDBC 다운로드 받기

▣ <https://downloads.mariadb.com/Connectors/java/connector-java-2.5.1/>

■ Mariadb-java-client-2.5.1.jar

▣ 이클립스 프로젝트 속성에 빌드 경로에 ModulePath에 추가



JDBC 프로그램 개발 절차

□ 1단계 : JDBC 드라이버 로드

```
Class.forName("com.mysql.jdbc.Driver");
```

org.mariadb.jdbc.Driver

2단계 : 데이터베이스 연결

▶ JDBC URL

- ① IP 주소 : MySQL 데이터베이스가 설치된 컴퓨터의 IP 주소 또는 도메인이름.
- ② 스키마 : 데이터베이스에서 생성한 스키마(데이터베이스) 이름.
- ③ PORT : 기본 설정값인 3306 포트를 사용할 때는 생략할 수 있다.

```
jdbc:<서브 프로토콜>:<데이터 원본 식별자> "jdbc:mariadb://127.0.0.1:5306/mydb"
```

```
jdbc:mysql://DB 서버의 IP 주소/스키마:PORT(옵션임)
```

①

②

③

▶ Connection 클래스 인스턴스 레퍼런스 얻기

roo 1234

t

```
Connection conn = DriverManager.getConnection(JDBC_URL, "아이디", "비밀번호");
```

①

②

① JDBC_URL : 해당 데이터베이스에 맞게 미리 정의한 문자열.

② 아이디와 비밀번호 : 시스템에 로그인하는 계정이 아닌 데이터베이스 자체에서 관리하는 계정

JDBC 프로그램 개발 절차

19

□ 연결 테스트

*패키지명 : java.sql.Connection
*모듈에 requires 구문 추가
* requires java.sql

```
public void connectDB() {  
    try {  
        Class.forName("org.mariadb.jdbc.Driver");  
        // Connection conn; 멤버 변수로 선언 후  
        conn = DriverManager.getConnection(  
            "jdbc:mariadb://127.0.0.1:6306/mydb",  
            "root",  
            "1234");  
        if( conn != null ) {  
            System.out.println("DB 접속 성공");  
        }  
    } catch (ClassNotFoundException e) {  
        System.out.println("JDBC 드라이버 로드 실패");  
    } catch (SQLException e) {  
        System.out.println("DB 접속 실패");  
        e.printStackTrace();  
    }  
}
```

JDBC 프로그램 개발 절차

3단계 : Statement 생성

- ▶ `executeQuery()` : SELECT 문을 수행할 때 사용

```
String sql = "select * from test";  
Statement stmt = conn.createStatement();  
stmt.executeQuery(sql);
```

- ▶ `executeUpdate()` : UPDATE 문, DELETE 문 등을 수행할 때 사용한다.

```
Statement stmt = conn.createStatement();  
stmt.executeUpdate("insert into test values(' "+getUsername()+" ', ' "+getEmail()+" ');");
```

- ▶ **PreparedStatement**

SQL 문을 미리 만들어 두고 변수를 따로 입력하는 방식으로, 효율성이 나 유지보수 측면에서 유리함

```
PreparedStatement pstmt = conn.prepareStatement("insert into test values(?, ?);");  
pstmt.setString(1, getUsername());  
pstmt.setString(2, getEmail());  
pstmt.executeUpdate();
```

```
stmt.close();  
pstmt.close();
```

JDBC 프로그램 개발 절차

4단계 : SQL 문 전송

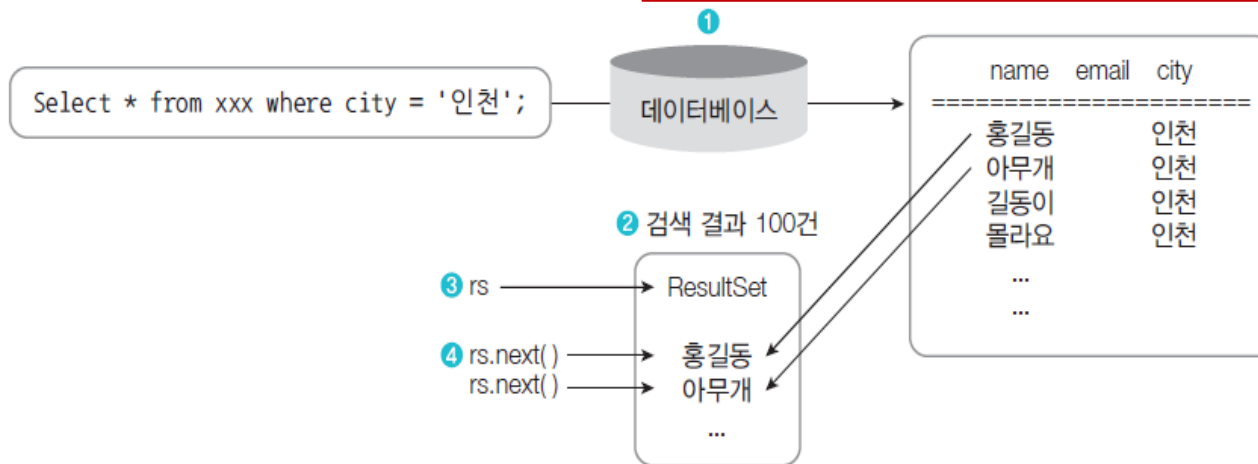
- 데이터를 입력·수정·삭제하려고 SQL 문을 만들 때는 PreparedStatement를 사용하여 변수와 적절히 조합.
- 데이터의 입력, 수정, 삭제는 Statement나 PreparedStatement의 executeUpdate() 메서드를 사용

```
int count = pstmt.executeUpdate();
```

```
pstmt.executeUpdate();
```

5 단계 : 결과 받기

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



JDBC 프로그램 개발 절차

5단계 : 결과받기

- Statement나 PreparedStatement의 executeQuery()를 사용한다.
- 입력, 수정, 삭제와 달리 데이터를 가져올 때는 가져온 결과 데이터를 처리하는 ResultSet 객체가 필요하다.

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

```
ResultSet rs = pstmt.executeQuery();
while(rs.next()) {
    name = rs.getString(1);    // or rs.getString("name");
    age = rs.getInt(2);        // or rs.getInt("email");
}
rs.close();
```

6단계 : 연결해제

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

DB 관련 멤버 추가

23

□ 멤버 추가

```
Connection conn;  
PreparedStatement pstmt;  
ResultSet rs;
```

□ 클래스 import

- ▣ java.sql.~~~

□ 모듈 정보 등록

- ▣ requires java.sql

데이터 삽입

24

```
public void registerStudent() {
    System.out.print("학생 정보(이름 국어 영어)를 입력해 주세요 : ");
    String name = scan.next();
    int kor = scan.nextInt();
    int eng = scan.nextInt();
    String sql = "insert into student values(?,?,?)";
    if(conn!=null) {
        try {
            pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, name);
            pstmt.setInt(2, kor);
            pstmt.setInt(3, eng);
            pstmt.execute();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```


데이터 리스트 획득

25

```
public void printDB() {
    String sql = "select * from student";
    if(conn!=null) {
        try {
            pstmt = conn.prepareStatement(sql);
            rs = pstmt.executeQuery();
            while(rs.next()) {
                String name = rs.getString(1);
                int kor = rs.getInt(2);
                int eng = rs.getInt(3);
                System.out.println(name + ", "+kor+", "+eng);
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

DB 연결 해제

26

```
public void closeDB() {  
    try {  
        pstmt.close();  
        rs.close();  
        conn.close();  
    } catch (SQLException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
}
```