



데이터베이스 프로그래밍



한국기술교육대학교
온라인평생교육원

학습내용

- 데이터베이스 이해하기
- 데이터베이스 활용하기

학습목표

- 데이터베이스의 개요와 SQL에 대하여 이해하고, JDBC에 대하여 설명할 수 있다.
- JavaDB를 이용하여 데이터베이스를 조작하는 응용 애플리케이션을 작성할 수 있다.

데이터베이스 이해하기

1 데이터베이스(database) 개요

① 데이터베이스란?

- 1 공유하여 사용하기 위한 중복되지 않는 데이터의 집합

② 데이터베이스 관리 시스템 (DBMS : Database Management System)

- ## 1 데이터베이스를 효율적으로 관리하기 위한 시스템

- 2 종류 : Oracle, MS-SQL, MariaDB , My-SQL등

③ 데이터베이스 사용한 자바 프로그램

- ## 1 DBMS 설치

- ## 2 DBMS 접근을 위한 드라이버 설치

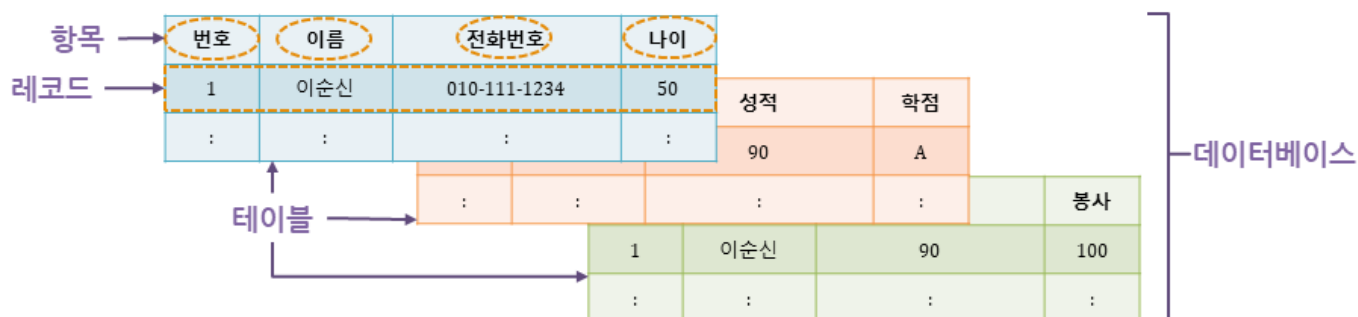
1 데이터베이스(database) 개요

④ 데이터베이스 구성

- 1 데이터의 기본 항목들이 모여서 하나의 레코드를 구성

- 2 레코드들을 모아서 표와 같은 형태의 테이블을 구성

- 3 관련 테이블들을 모아서 데이터베이스를 구성



데이터베이스 이해하기

2 MariaDB 다운로드 및 설치

① MariaDB 개요

- 1 세계에서 가장 인기 있는 DBMS중에 하나
- 2 관계형 데이터베이스
- 3 오픈 소스 소프트웨어

2 MariaDB 다운로드 및 설치

② MariaDB 다운로드

- 1 홈페이지 : <https://mariadb.org/download>
- 2 최신 버전의 설치 파일을 다운로드

Download MariaDB Server

MariaDB Server is one of the world's most popular open source relational databases and is available in the standard repositories of all major Linux distributions. Look for the package mariadb-server using the package manager of your operating system. Alternatively you can use the following resources:

MariaDB Server

MariaDB Repositories

Connectors

MariaDB Server Version
 MariaDB Server 10.6.4

Display older releases: ☐

Operating System
 Windows

Architecture
 x86_64

Package Type
 MSI Package

Download

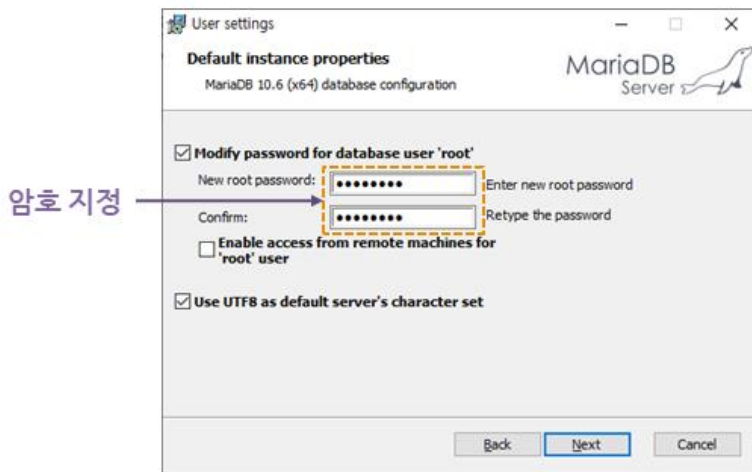
Mirror
 Yngbok.net - South Korea

데이터베이스 이해하기

2 MariaDB 다운로드 및 설치

③ MariaDB 설치

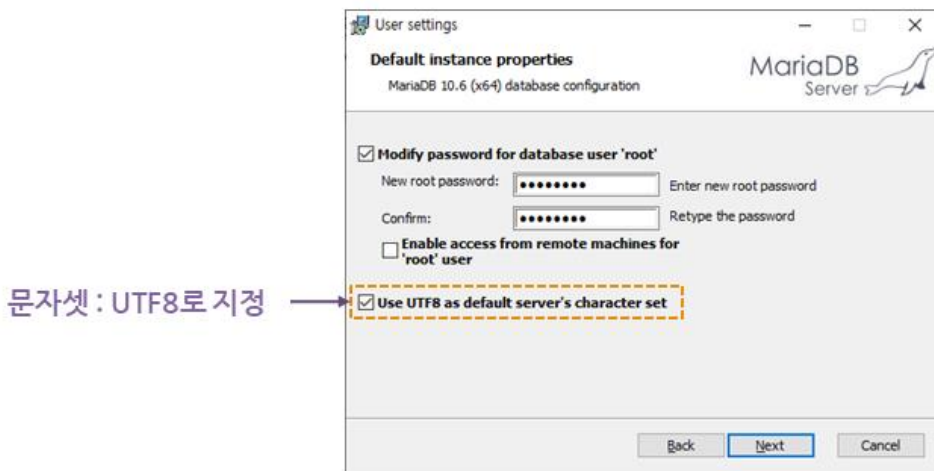
1 DBMS 관리자(root)의 암호 지정



2 MariaDB 다운로드 및 설치

③ MariaDB 설치

2 문자셋 지정 : UTF-8

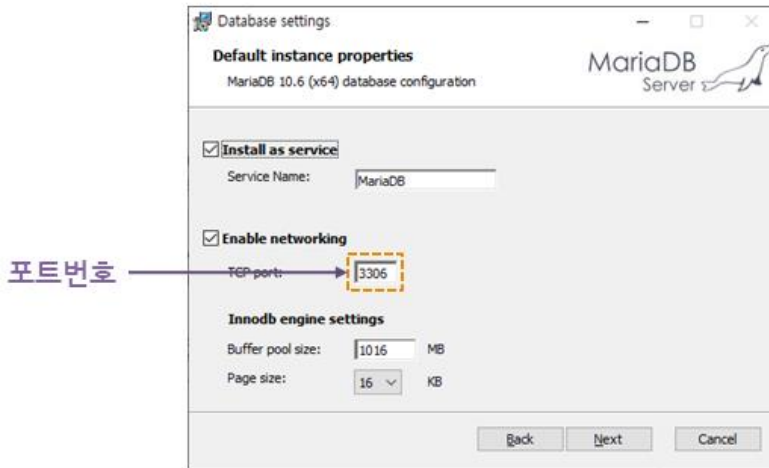


데이터베이스 이해하기

2 MariaDB 다운로드 및 설치

3 MariaDB 설치

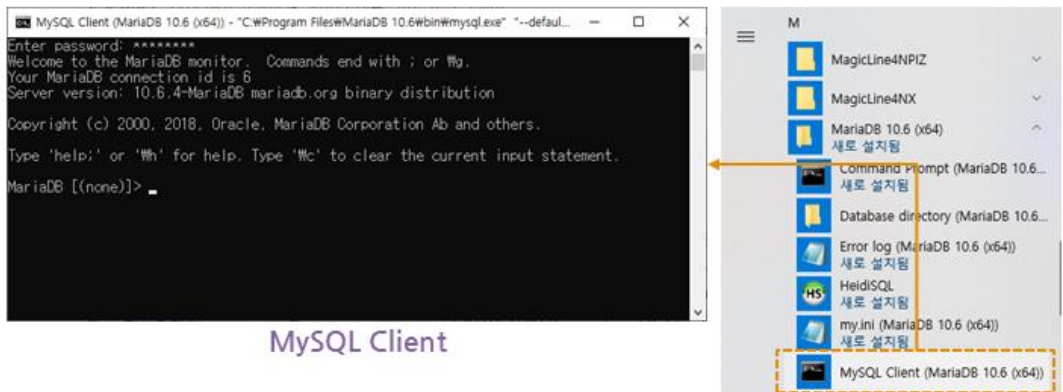
3 포트 번호 : 3306



2 MariaDB 다운로드 및 설치

4 데이터베이스 관리

1 MySQL Client 프로그램 사용



MySQL Client

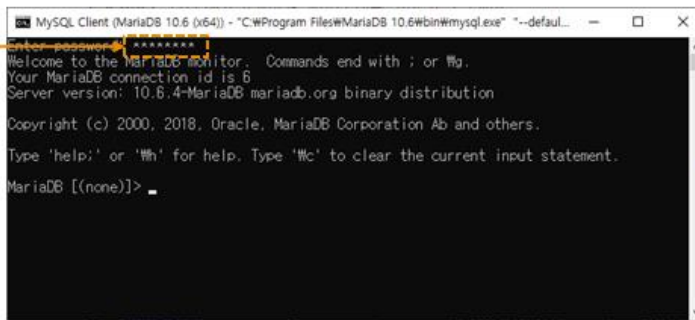
데이터베이스 이해하기

2 MariaDB 다운로드 및 설치

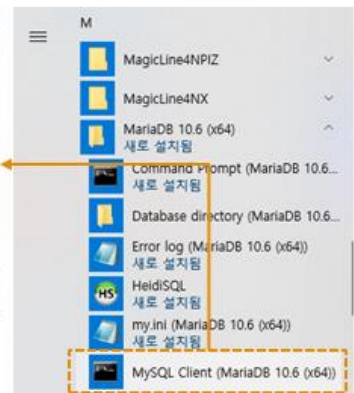
④ 데이터베이스 관리

1 MySQL Client 프로그램 사용

암호 입력



MySQL Client

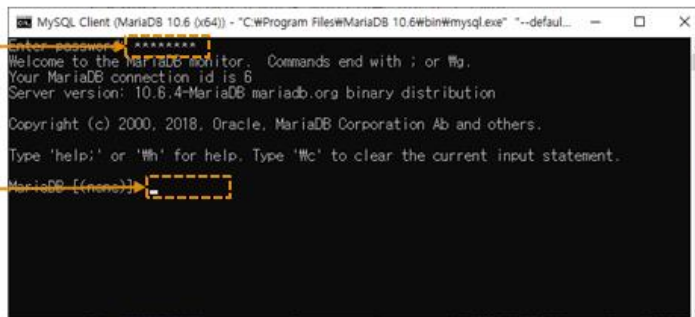


2 MariaDB 다운로드 및 설치

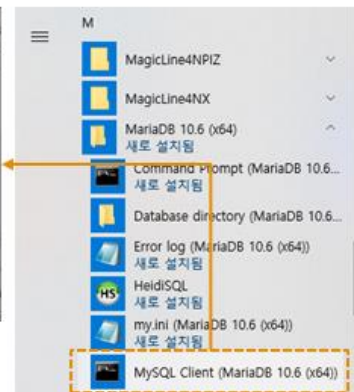
④ 데이터베이스 관리

1 MySQL Client 프로그램 사용

암호 입력



MySQL Client



데이터베이스 이해하기

2 MariaDB 다운로드 및 설치

④ 데이터베이스 관리

- 1 MySQL Client 프로그램 사용
- 2 주요 명령어

데이터베이스 현황 보기 : show databases;

데이터베이스 만들기 : create database 데이터베이스이름;

데이터베이스 지우기 : drop database 데이터베이스이름;

데이터베이스 연결 하기 : connect 데이터베이스이름;

데이터베이스의 테이블 현황 보기 : show tables;

3 SQL(Structured Query Language)

① SQL이란?

- 1 데이터베이스를 조작하거나 접근할 수 있는 표준 언어
- 2 데이터베이스와 상관없이 사용 가능

구분	설명
DDL(Data Definition Language)	데이터베이스를 정의하거나 변경하는 언어 • CREATE, DROP, ALTER 등
DML(Data Manipulation Language)	데이터베이스의 데이터를 조작하는 언어 • INSERT, SELECT, UPDATE, DELETE 등
DCL(Data Control Language)	데이터베이스 권한을 제어하는 언어 • GRANT, REVOKE

• SQL에 대한 학습 추천 : 데이터베이스(2018년), SQL활용

데이터베이스 이해하기

3 SQL(Structured Query Language)

② 데이터 정의 언어 (DDL)

1 테이블의 생성과 삭제 및 수정을 위한 명령

구분	명령어	설명
생성	CREATE	CREATE TABLE 테이블명 (필드명 데이터타입 [PRIMARY KEY] [NOT NULL][UNIQUE]);
삭제	DROP	DROP TABLE 테이블명;
수정	ALTER	ALTER TABLE ADD (필드명 데이터타입 [PRIMARY KEY] [NOT NULL][UNIQUE][DEFAULT]); ALTER TABLE DROP 필드명; ALTER TABLE ALTER (필드명 데이터타입);

• SQL에 대한 학습 추천 : 데이터베이스(2018년), SQL활용

3 SQL(Structured Query Language)

③ 데이터 조작 언어 (DML)

1 데이터의 추가, 검색, 수정, 삭제를 하는 명령

구분	명령어	설명
검색	SELECT	SELECT 필드명 ALL DISTINCT FROM 테이블명 WHERE 조건 [GROUP BY 필드명...] [HAVING 조건] [ORDER BY 필드명 ASC DESC];
삭제	DELETE	DELETE FROM 테이블명 WHERE 조건;
수정	UPDATE	UPDATE 테이블명 SET 필드명=데이터 WHERE 조건;
추가	INSERT	INSERT INTO 테이블명(필드명...) VALUES (데이터...);

• SQL에 대한 학습 추천 : 데이터베이스(2018년), SQL활용

데이터베이스 이해하기

4 JDBC(Java Database Connectivity)

① JDBC의 개요

- 1 자바에서 제공하는 데이터베이스 접근 API
- 2 SQL을 지원하는 데이터베이스에 일관된 접근
DBMS와 상관없이 프로그램 작성 가능
- 3 DBMS에서 제공하는 JDBC 드라이버 사용
- 4 JDBC 패키지 : `java.sql.*`

4 JDBC(Java Database Connectivity)

② JDBC 드라이버

- 1 MariaDB JDBC 드라이버 다운로드

<https://downloads.mariadb.org/connector-java/>

안정화된 최신 버전 다운로드

데이터베이스 이해하기

4 JDBC(Java Database Connectivity)

② JDBC 드라이버

1 MariaDB JDBC 드라이버 다운로드

파일명 : mariadb-java-client-2.7.3.jar

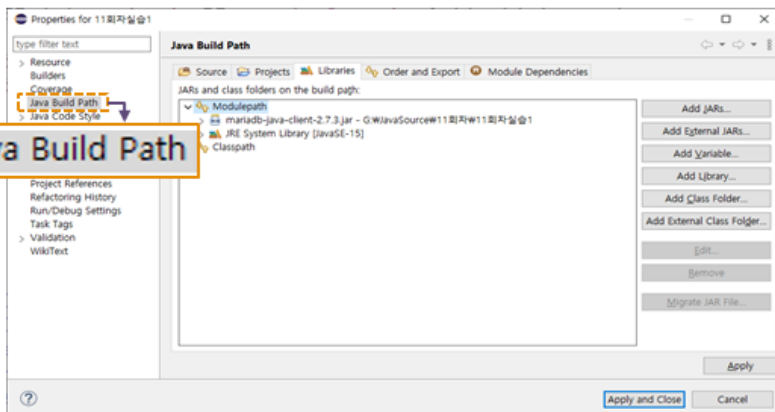


4 JDBC(Java Database Connectivity)

② JDBC 드라이버

2 프로젝트에 등록

[Project]-[Properties]



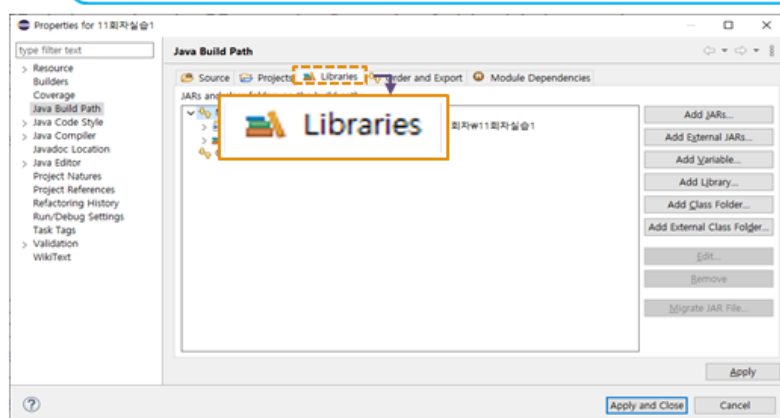
데이터베이스 이해하기

4 JDBC(Java Database Connectivity)

② JDBC 드라이버

2 프로젝트에 등록

[Project]-[Properties]

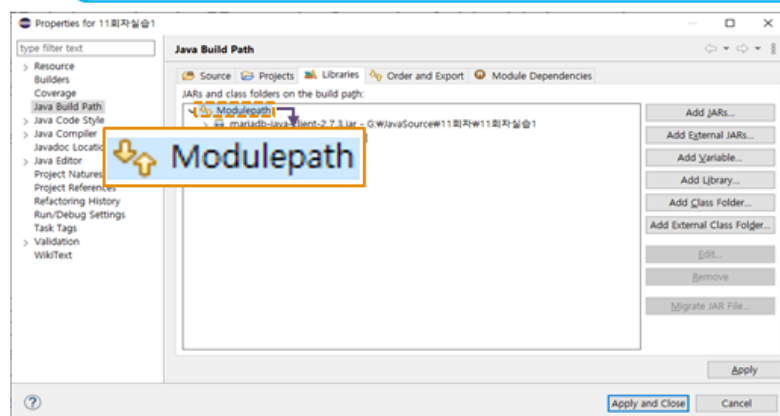


4 JDBC(Java Database Connectivity)

② JDBC 드라이버

2 프로젝트에 등록

[Project]-[Properties]



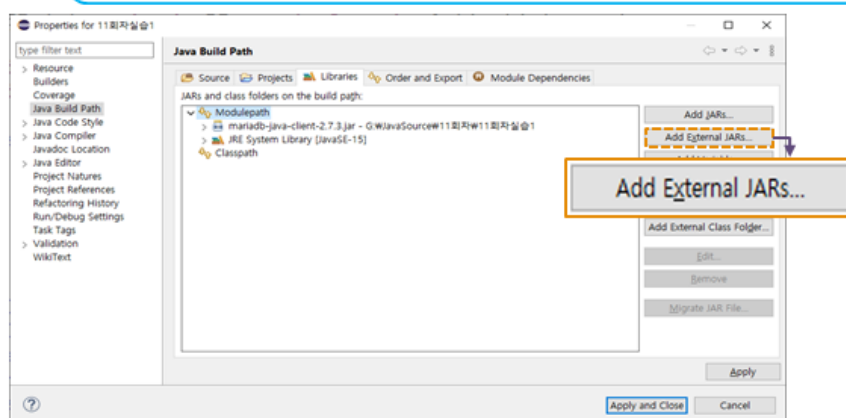
데이터베이스 이해하기

4 JDBC(Java Database Connectivity)

② JDBC 드라이버

2 프로젝트에 등록

[Project]-[Properties]

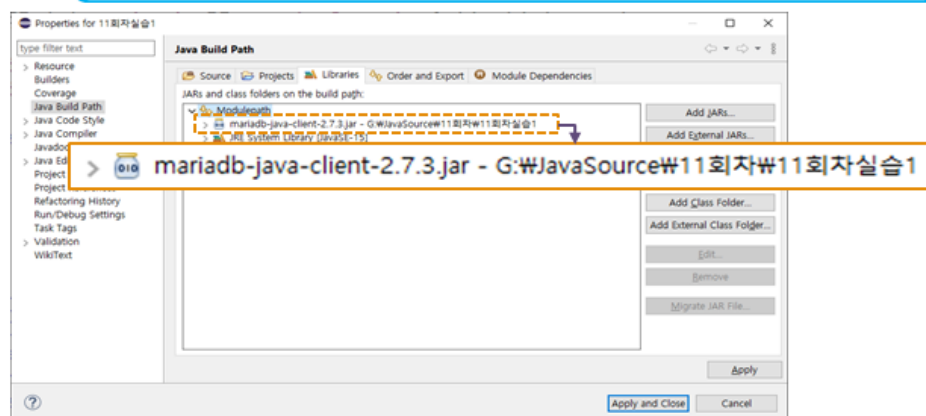


4 JDBC(Java Database Connectivity)

② JDBC 드라이버

2 프로젝트에 등록

[Project]-[Properties]



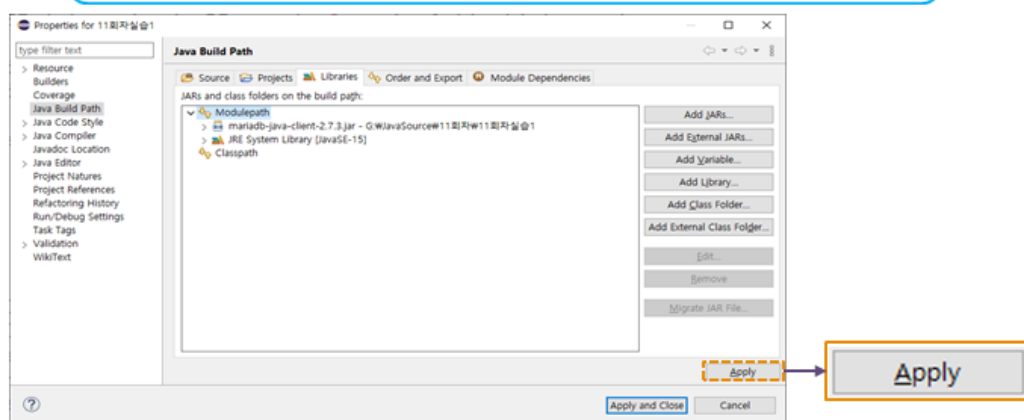
데이터베이스 이해하기

4 JDBC(Java Database Connectivity)

② JDBC 드라이버

2 프로젝트에 등록

[Project]-[Properties]

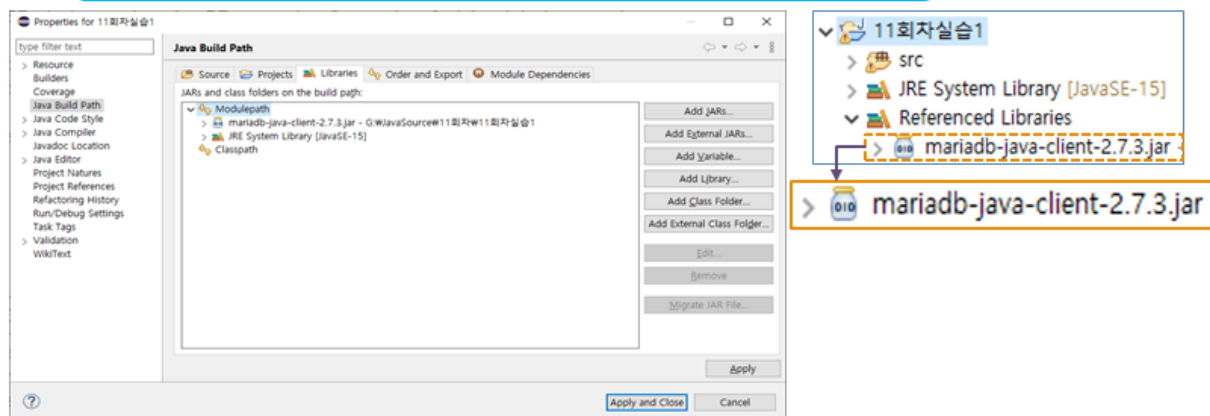


4 JDBC(Java Database Connectivity)

② JDBC 드라이버

2 프로젝트에 등록

[Project]-[Properties]



데이터베이스 이해하기

4 JDBC(Java Database Connectivity)

② JDBC 드라이버

③ MariaDB JDBC 드라이버 로딩 및 데이터베이스 연결 방법

구분	설명
JDBC 드라이버 로딩	<pre>String DB_DRIVER = "org.mariadb.jdbc.Driver"; Class.forName(DB_DRIVER);</pre> ClassNotFoundException 예외 발생
데이터베이스 연결	<pre>String DB_URL = "jdbc:mariadb://localhost:3306/students"; Connection db = DriverManager.getConnection(DB_URL, "root", "12345678");</pre> SQLException 예외 발생

4 JDBC(Java Database Connectivity)

② JDBC 드라이버

③ MariaDB JDBC 드라이버 로딩 및 데이터베이스 연결 방법

구분	설명
JDBC 드라이버 로딩	<pre>String DB_DRIVER = "org.mariadb.jdbc.Driver"; Class.forName(DB_DRIVER);</pre> ClassNotFoundException 예외 발생
데이터베이스 연결	<pre>String DB_URL = "jdbc:mariadb://localhost:3306/students"; Connection db = DriverManager.getConnection(DB_URL, "root", "12345678");</pre> SQLException 예외 발생

DBMS 네트워크 주소

데이터베이스 이해하기

4 JDBC(Java Database Connectivity)

② JDBC 드라이버

③ MariaDB JDBC 드라이버 로딩 및 데이터베이스 연결 방법

구분	설명
JDBC 드라이버 로딩	<pre>String DB_DRIVER = "org.mariadb.jdbc.Driver"; Class.forName(DB_DRIVER);</pre> ClassNotFoundException 예외 발생
데이터베이스 연결	<pre>String DB_URL = "jdbc:mariadb://localhost:3306/students"; Connection db = DriverManager.getConnection(DB_URL, "root", "12345678");</pre> SQLException 예외 발생

포트 번호

4 JDBC(Java Database Connectivity)

② JDBC 드라이버

③ MariaDB JDBC 드라이버 로딩 및 데이터베이스 연결 방법

구분	설명
JDBC 드라이버 로딩	<pre>String DB_DRIVER = "org.mariadb.jdbc.Driver"; Class.forName(DB_DRIVER);</pre> ClassNotFoundException 예외 발생
데이터베이스 연결	<pre>String DB_URL = "jdbc:mariadb://localhost:3306/students"; Connection db = DriverManager.getConnection(DB_URL, "root", "12345678");</pre> SQLException 예외 발생

데이터베이스 이름

데이터베이스 이해하기

4 JDBC(Java Database Connectivity)

② JDBC 드라이버

③ MariaDB JDBC 드라이버 로딩 및 데이터베이스 연결 방법

구분	설명
JDBC 드라이버 로딩	<pre>String DB_DRIVER = "org.mariadb.jdbc.Driver"; Class.forName(DB_DRIVER);</pre> <p>ClassNotFoundException 예외 발생</p>
데이터베이스 연결	<pre>String DB_URL = "jdbc:mariadb://localhost:3306/students"; Connection db = DriverManager.getConnection(DB_URL, "root", "12345678");</pre> <p>SQLException 예외 발생</p>

DBMS 관리자 ID

4 JDBC(Java Database Connectivity)

② JDBC 드라이버

③ MariaDB JDBC 드라이버 로딩 및 데이터베이스 연결 방법

구분	설명
JDBC 드라이버 로딩	<pre>String DB_DRIVER = "org.mariadb.jdbc.Driver"; Class.forName(DB_DRIVER);</pre> <p>ClassNotFoundException 예외 발생</p>
데이터베이스 연결	<pre>String DB_URL = "jdbc:mariadb://localhost:3306/students"; Connection db = DriverManager.getConnection(DB_URL, "root", "12345678");</pre> <p>SQLException 예외 발생</p>

암호

데이터베이스 이해하기

5 테이블(Table) 생성

① 테이블 구성

1 테이블 이름 : INFO

2 항목

번호(NUMBER) : 숫자, 주 키(Primary Key)

NUMBER
1
:
:

테이블 이름 : INFO

5 테이블(Table) 생성

① 테이블 구성

1 테이블 이름 : INFO

2 항목

이름(HNAME) : 문자열

NUMBER	HNAME
1	김장군
:	:
:	:

테이블 이름 : INFO

데이터베이스 이해하기

5 테이블(Table) 생성

① 테이블 구성

1 테이블 이름 : INFO

2 항목

전화번호(TELNO) : 문자열

NUMBER	HNAME	TELNO
1	김장군	010-111-1111
:	:	:
:	:	:

테이블 이름 : INFO

5 테이블(Table) 생성

① 테이블 구성

1 테이블 이름 : INFO

2 항목

나이(AGE) : 숫자

NUMBER	HNAME	TELNO	AGE
1	김장군	010-111-1111	30
:	:	:	:
:	:	:	:

테이블 이름 : INFO

데이터베이스 이해하기

5 테이블(Table) 생성

② 테이블 생성 SQL문

CREATE TABLE INFO (NUMBER INT NOT NULL PRIMARY KEY,
HNAME VARCHAR(20) NOT NULL,
TELNO VARCHAR(15) NOT NULL,
AGE INT NOT NULL)

↑
테이블 생성 명령
CREATE TABLE

5 테이블(Table) 생성

② 테이블 생성 SQL문

CREATE TABLE INFO (NUMBER INT NOT NULL PRIMARY KEY,
HNAME VARCHAR(20) NOT NULL,
TELNO VARCHAR(15) NOT NULL,
AGE INT NOT NULL)

↑
테이블 이름
INFO

데이터베이스 이해하기

5 테이블(Table) 생성

② 테이블 생성 SQL문

```
CREATE TABLE INFO (NUMBER INT NOT NULL PRIMARY KEY, ← 번호(NUMBER)
                    HNAME VARCHAR(20) NOT NULL,
                    TELNO VARCHAR(15) NOT NULL,
                    AGE INT NOT NULL)
```

5 테이블(Table) 생성

② 테이블 생성 SQL문

```
CREATE TABLE INFO (NUMBER INT NOT NULL PRIMARY KEY,
                    HNAME VARCHAR(20) NOT NULL, ← 이름(HNAME)
                    TELNO VARCHAR(15) NOT NULL,
                    AGE INT NOT NULL)
```

데이터베이스 이해하기

5 테이블(Table) 생성

② 테이블 생성 SQL문

```
CREATE TABLE INFO (NUMBER INT NOT NULL PRIMARY KEY,
                    HNAME VARCHAR(20) NOT NULL,
                    TELNO VARCHAR(15) NOT NULL, ← 전화번호(TELNO)
                    AGE INT NOT NULL)
```

5 테이블(Table) 생성

② 테이블 생성 SQL문

```
CREATE TABLE INFO (NUMBER INT NOT NULL PRIMARY KEY,
                    HNAME VARCHAR(20) NOT NULL,
                    TELNO VARCHAR(15) NOT NULL,
                    AGE INT NOT NULL) ← 나이(AGE)
```


데이터베이스 이해하기

5 테이블(Table) 생성

③ 테이블 생성 프로그램 순서

순서	코딩/설명
JDBC 드라이버 로딩	String DB_DRIVER = "org.mariadb.jdbc.Driver"; Class.forName(DB_DRIVER);
데이터베이스 연결	String DB_URL = "jdbc:mariadb://localhost:3306/students"; Connection db = DriverManager.getConnection(DB_URL, "root", "12345678")
SQL문 객체 생성	Statement statement = db.createStatement(); SQLException 예외 발생

5 테이블(Table) 생성

③ 테이블 생성 프로그램 순서

순서	코딩/설명
SQL문 실행	String sql = "CREATE TABLE INFO (... "; statement.execute(sql); SQLException 예외 발생
SQL문 객체 닫기	statement.close(); SQLException 예외 발생
데이터베이스 연결 종료	db.close(); SQLException 예외 발생

데이터베이스 이해하기



데이터베이스 프로그래밍

데이터베이스 이해하기



실습하기



데이터베이스 프로그래밍

데이터베이스 이해하기

실습순서

1. MariaDB 다운로드 및 설치 실습
2. 데이터베이스 생성 실습
3. MariaDB JDBC 드라이버 다운로드 실습
4. 테이블 생성 프로그래밍 실습



유의사항

- JDK와 이클립스를 설치한 후 실습이 가능함
- 본인이 원하는 작업 폴더를 미리 정해 놓은 다음 실습하기
- 작업 폴더는 C드라이브에 지정하기 보다는 D드라이브나 외장하드디스크를 활용하는 것을 추천함

※ 제공되는 실습 코드를 다운받아 실습해보시기 바랍니다.

데이터베이스 활용하기

1 활용 클래스

① Statement 클래스

- 1 Connection 객체의 `createStatement()` 메소드로 객체 생성
- 2 SQL문을 실행하기 위한 메소드 제공

SQLException 예외 발생

주요 메소드	설 명
<code>boolean execute(String sql)</code>	지정한 SQL문 실행 • CREATE문 실행
<code>int executeUpdate(String sql)</code>	지정한 SQL문 실행 • INSERT, UPDATE, DELETE문 실행
<code>ResultSet executeQuery(String sql)</code>	지정한 SQL문 실행하고 ResultSet 객체를 반환 • SELECT문 실행

1 활용 클래스

② PreparedStatement 클래스

- 1 Connection 객체의 `prepareStatement(String sql)` 메소드로 객체 생성

인수로 미완성의 SQL문을 지정함

- 2 SQL문을 실행하기 위한 메소드 제공

SQLException 예외 발생

- 3 SQL문을 만들기 위한 메소드 제공

복잡한 SQL문을 작성할 때 사용

미완성된 SQL문을 완성시킴

데이터베이스 활용하기

1 활용 클래스

② PreparedStatement 클래스

③ SQL문을 만들기 위한 메소드 제공

주요 메소드	설 명
boolean execute() int executeUpdate() ResultSet executeQuery()	SQL문 실행
void setInt(int index, int x)	index 위치에 정수값을 지정
void setString(int index, String x)	Index 위치에 문자열을 지정

1 활용 클래스

② PreparedStatement 클래스

④ SQL문 만들기

미완성된 SQL문 : "UPDATE INFO SET HNAME=?,TELNO=?,AGE=? WHERE NUMBER=?"

데이터베이스 활용하기

1 활용 클래스

② PreparedStatement 클래스

④ SQL문 만들기

미완성된 SQL문 : "UPDATE INFO SET HNAME=?, TELNO=?, AGE=? WHERE NUMBER=?"

↑ index 1
 ↑ index 2
 ↑ index 3
 ↑ index 4

1 활용 클래스

② PreparedStatement 클래스

④ SQL문 만들기

setString(2, "010-111-1111")

setString(1, "김장군")

미완성된 SQL문 : "UPDATE INFO SET HNAME=?, TELNO=?, AGE=? WHERE NUMBER=?"

데이터베이스 활용하기

1 활용 클래스

② PreparedStatement 클래스

④ SQL문 만들기

미완성된 SQL문 : "UPDATE INFO SET HNAME=?, TELNO=?, AGE=? WHERE NUMBER=?"

setInt(3, 25) → AGE=?
setInt(4, 100) → NUMBER=?

1 활용 클래스

③ ResultSet 클래스

- 1 SELECT문을 실행한 결과인 레코드의 집합을 표현한 클래스
- 2 executeQuery() 메소드에서 객체를 반환
- 3 SQLException 예외 발생

주요 메소드	설 명
boolean next()	레코드의 집합에서 레코드의 위치를 다음으로 이동 false가 반환되면 레코드의 마지막 위치
int getInt(String columnLabel)	지정한 항목의 이름에 해당하는 값을 int로 반환
int getString(String columnLabel)	지정한 항목의 이름에 해당하는 값을 String 객체로 반환

데이터베이스 활용하기



2 데이터베이스 조작

① 레코드 추가 프로그램 작성 순서

순서	코딩/설명
JDBC 드라이버 로딩	<code>String DB_DRIVER = "org.mariadb.jdbc.Driver"; Class.forName(DB_DRIVER);</code>
데이터베이스 연결	<code>String DB_URL = "jdbc:mariadb://localhost:3306/students"; Connection db = DriverManager.getConnection(DB_URL, "root", "12345678");</code>
SQL문 객체 생성	<code>Statement statement = db.createStatement();</code>

2 데이터베이스 조작

① 레코드 추가 프로그램 작성 순서

순서	코딩/설명
SQL문 실행	<code>String sql = "INSERT INTO INFO VALUES(...)"; statement.executeUpdate(sql);</code>
SQL문 객체 닫기	<code>statement.close();</code>
데이터베이스 연결 종료	<code>db.close();</code>

데이터베이스 활용하기

2 데이터베이스 조작

② 레코드 검색 프로그램 작성 순서

순서	코딩/설명
JDBC 드라이버 로딩 → 데이터베이스 연결	
SQL문 객체 생성	<code>Statement statement = db.createStatement();</code>
SQL문 실행	<code>String sql="SELECT * FROM INFO"; ResultSet rs = statement.executeQuery(sql);</code>

2 데이터베이스 조작

② 레코드 검색 프로그램 작성 순서

순서	코딩/설명
레코드들의 내용을 화면에 출력	<code>while(rs.next()) { rs.getInt("NUMBER"); rs.getString("HNAME"); rs.getString("TELNO"); rs.getInt("AGE"); }</code>
Statement 객체 닫기 → 데이터베이스 연결 종료	

데이터베이스 활용하기

2 데이터베이스 조작

③ 레코드 수정 프로그램 작성 순서

순서	코딩/설명
JDBC 드라이버 로딩 → 데이터베이스 연결	
SQL문 객체 생성	String sql="UPDATE INTO SET HNAME=?, ..."; PreparedStatement statement = db.prepareStatement(sql);
SQL문 실행	statement.setString(1, "김장군"); statement.setString(2, "010-111-1111"); statement.setInt(3, 25); statement.setInt(4, 100); statement.executeUpdate();
PreparedStatement 객체 닫기 → 데이터베이스 연결 종료	

2 데이터베이스 조작

④ 레코드 삭제 프로그램 작성 순서

순서	코딩/설명
JDBC 드라이버 로딩 → 데이터베이스 연결	
SQL문 객체 생성	String sql="DELTE FROM INFO WHERE NUMBER=?"; PreparedStatement statement = db.prepareStatement(sql);
SQL문 실행	statement.setInt(1, 100); statement.executeUpdate();
PreparedStatement 객체 닫기 → 데이터베이스 연결 종료	

데이터베이스 활용하기



데이터베이스 프로그래밍

데이터베이스 활용하기



실습하기



데이터베이스 프로그래밍

데이터베이스 활용하기

실습순서

1. 레코드 추가, 검색, 수정, 삭제 프로그래밍 실습



유의사항

- JDK와 이클립스를 설치한 후 실습이 가능함
- 본인이 원하는 작업 폴더를 미리 정해 놓은 다음 실습하기
- 작업 폴더는 C드라이브에 지정하기 보다는 D드라이브나 외장하드디스크를 활용하는 것을 추천함



※ 제공되는 실습 코드를 다운받아 실습해보시기 바랍니다.

응용문제

데이터베이스 프로그래밍 응용문제

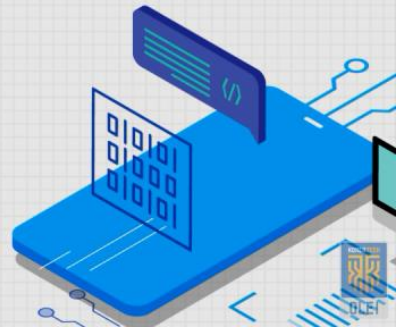
다음 실행화면과 조건에 맞게 프로그램을 작성하시오.

조건

- 1 레코드 추가, 검색, 수정, 삭제를 하나의 화면에 구현
- 2 전체 목록 보기 버튼 추가

프로젝트명: DBTotal

제공되는 실습 소스코드를 다운받아 실습해보시기 바랍니다.



데이터베이스 프로그래밍 응용문제

실행화면

JDBC-MariaDB TOTAL

DB Connect

DB Disconnect

CLEAR

List All

번호 2

이름 최고집

전화 번호 010-222-2222

나이 45

INSERT

SELECT

UPDATE

DELETE

번호	이름	전화번호	나이
1	김장군2	010-111-11112	5312
2	최고집	010-222-2222	45
4	김고려	010-444-4444	25