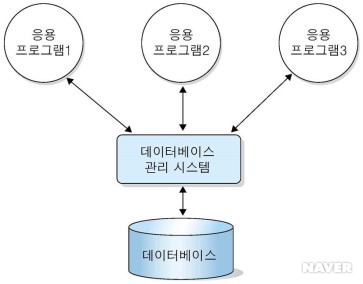
**DBMS와 그 종류**

데이터베이스를 직접 프로그램들이 조작하는 것이 아니라 데이터베이스를 조작하는 별도의 소프트웨어가 있는데

이를 데이터베이스 관리 시스템(DBMS : DataBase Management System)이라 한다.

즉 데이터베이스 관리 시스템이란 데이터베이스를 관리하며 응용 프로그램들이 데이터베이스를 공유하며 사용할

수 있는 환경을 제공하는 소프트웨어이다.

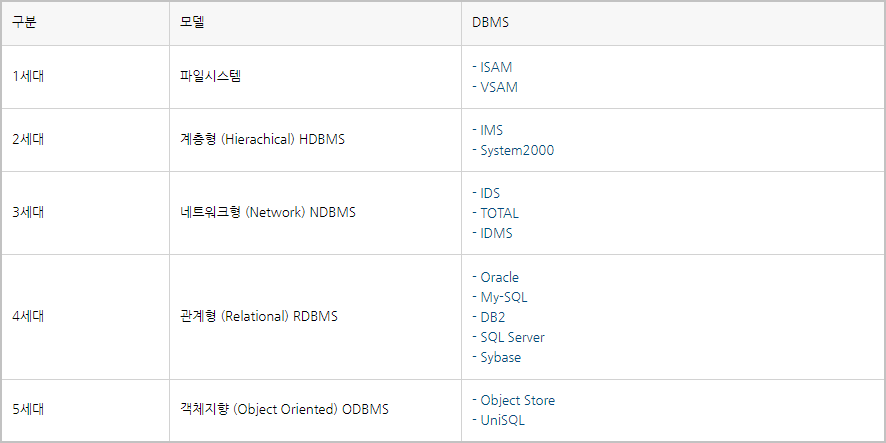


이런 데이터베이스 관리 시스템은 데이터베이스를 구축하는 틀을 제공하고, 효율적으로 데이터를 검색하고

저장하는 기능을 제공한다. 또한 응용 프로그램들이 데이터베이스에 접근할 수 있는 인터페이스를 제공하고,

장애에 대한 복구 기능, 사용자 권한에 따른 보안성 유지 기능 등을 제공 한다.

DBMS는 계층형과 네트워크형, 그리고 관계형으로 나눠지며 현재는 관계형이 DBMS의 주류를 이루고 있다.



1. 계층형(Hierachical) 데이터베이스 관리 시스템(HDBMS)

데이터가 계층적이며 상하 종속적인 관계로 구성

장점 : 데이터의 엑세스 속도가 빠르고, 데이터의 사용량을 쉽게 예측할 수 있다.

단점 : 상하 종속적인 관계로 구성되어 있어 초기 세팅 후 변화하는 프로세스를 수용하기가 쉽지 않다.

1. 네트워크(Network) 데이터베이스 관리 시스템(NDBMS)

데이터 구조를 네트워크상의 노드 형태로 논리적이게 표현한 모델로서 각각의 노드를 서로 대등한

관계로 구성한 시스템 (여기서의 노드란 시스템을 의미하는 것이 아닌 자료를 말한다.)

장점 : 계층형 데이터베이스 관리시스템의 문제점인 상하 종속적인 관계를 해결하였다.

단점 : 구성과 설계가 복잡하고 궁극적으로 데이터의 종속성을 해결하지 못하였다.

1. 관계형 데이터베이스 관리 시스템(RDBMS)

수학적 논리 관계를 테이블 형태로 구성한 구조로서 테이블 내의 컬럼 중 일부를 다른 테이블과 중복해

각 테이블 간의 상관관계를 정의

장점 : 업무 변화에 대한 적응력이 높아 변화하는 업무에 쉽게 활용하며 유지보수가 편리하다. 생산성 ↑

단점 : 다른 DBMS보다 더 많은 자원이 활용되어 시스템의 부하가 높다.

1. 객체지향(Object Oriented) 데이터베이스 관리 시스템(ODBMS)

멀티미디어 데이터의 원활한 처리와 RDBMS의 비즈니스형 데이터 타입만 처리되는 기본적인 제한점을

극복하고자 고안

DBMS 구조적 방법론에서 1~3세대는 4세대(관계형) DBMS로 빠른 전환이 이루어졌지만, 관계형 DBMS에서

객체지향 DBMS로는 전환이 되지 않고 있다. ( ODBMS에서는 쿼리 사용이 복잡해지기 때문 )

따라서 관계형 DBMS인 Oracle, SQL Server, MySQL, MariaDB, DB2등이 주로 사용되고 있다.

**RDBMS에서의 관계**

관계형 데이터베이스에서 관계라는 이름이 붙어진 이유는 각 테이블의 행과 행이 연결되는 관계를 맺을 수 있기

때문이다. 테이블 간의 관계는 1:1, 1:N, N:N의 관계가 있다.

우리는 하나의 테이블에 필요한 모든 필드를 넣고 모든 데이터 항목을 저장할 수 있다. 하지만 이렇게 하면

데이터들이 중복해서 저장되는 상황이 발생할 수 있어 다음 그림 테이블과 같이 비효율적이다.



그래서 관계형 데이터베이스 모델에서는 다음 그림과같이 테이블을 분리하여 행과 행을 연결할 수 있다.



그림과 같이 테이블 간의 관계는 기본 키(primary key)와 외래 키(foreign key)라는 개념을 사용하여 맺어진다.

기본 키는 고유한 ID 필드로 그림에서는 고객 번호 필드이다. 이 필드는 각 행이 중복된 값을 가질 수 없다.

외래 키는 기본 키를 참조하는 필드로 그림에서는 주문 테이블의 고객 번호 필드이고 각 테이블의 행을

연결시켜주는 역할을 한다. 이렇게 테이블을 분리하고 관계를 형성해 데이터를 효율적으로 관리할 수 있다.

( 테이블을 분리하고 중복 데이터를 제거하는 과정을 정규화라고 한다. )

**RDBMS 종류**

1. Oracle / Oracle

전 세계적으로 가장 많이 활용되는 관계형 데이터베이스 시스템이다. 오라클사에서 서비스하고 있으며

Unix, Linux, Windows 등 대부분의 OS를 지원한다. Oracle Rea Application Cluster(RAC)의 기능으로

데이터베이스 관리에 있어서 자원의 확장성과 장애에 대한 대처가 유연하다는 장점이 있다.

중견기업, 대기업 등 대형화된 데이터를 관리하기에 최적화되어 있다.

1. MS-SQL Server / Microsoft

마이크로소프트사의 대표적인 관계형 데이터베이스 시스템이다.

MS제품군이기 때문에 Window 서버에서만 구동이 되고 C#과는 가장 높은 호환성을 자랑하는 DBMS이다.

1. MySQL / Oracle (구 Sun)

썬 마이크로시스템즈에서 개발한 관계형 데이트베이스 시스템이다. Unix, Linux, Windows 운영 체제 등에서

사용할 수 있으며 무엇보다 오픈소스의 장점으로 많은 기업에서 홈페이지나 쇼핑몰 등 일반적인 웹 개발에

널리 이용되고 있다. 오라클에 비해 사용자 편의를 위한 기능, 사용자 실수 또는 재해에 대비한 기능,

성능 향상 등 기능적인 한계를 보이지만 오픈소스라는 장점 때문에 현재 많은 기업에서 활용하고 있다.

1. MariaDB / MariaDB재단

2009년에 발표되었으며 비교적 역사가 짧다. MySQL이 오라클에 인수되면서 시작되었다.

무료버전인 MySQL보다 주력 상품인 Oracle을 팔아야 하는 입장에서 오픈소스인 MySQL의 기능을

감소시키게 되었고 이에 MySQL을 개발한 Michael Monty Widenius가 오라클과의 의견 충돌로 회사를 나와

새롭게 개발한 것이 이 MariaDB이다. MySQL과 완벽하게 호환되며 기본적인 명령어나 사용방법까지

동일하기 때문에 개발자 또는 DBA가 쉽게 접할 수 있으며 MySQL에 비해 성능적인 부분에서 70%나

향상이 되었기 때문에 현재 MariaDB를 활용하는 기업이 많이 늘어나는 추세이다.

1. DB2 / IBM

대형화된 데이터 관리를 목적으로 만들어진 IBM의 관계형 데이터베이스 관리 시스템이다.

사용자들이 서로 관계된 여러 개의 데이터베이스를 동시에 접근할 수 있다.

DB2의 특징은 각 워크로드(업무)의 특성에 맞게 시스템이 최적화될 수 있으며 자가 최적화, 자가 치유,

자가 구성, 워크로드 관리, 확장된 자동화 기능 등 다양한 기능을 구현할 수 있다. 또한 데이터 압축 기술이

좋기 때문에 대형화된 데이터를 다루는데 최적화되어 있다. 때문에 많은 중견기업 대기업에서 활용하고 있다.

1. Sysbase / Sysbase

사이베이스 사에서 개발한 관계형 데이터베이스 시스템이다.

2011년에 사이베이스 IQ 15.3이 출시되어 컬럼 단위 데이터 처리로 I/O 속도를 90%까지 향상시켰으며

데이터 압축 저장, 스토리지 공간의 활용도를 높이게 되었다. 또한 오라클에 비해 비교적 저럼하기에 많은

기업에서 관심을 가지고 있지만 아직까지 국내에서는 오라클에 비해 밀리는 추세이다.

Maraidb – Jsp 연동 절차

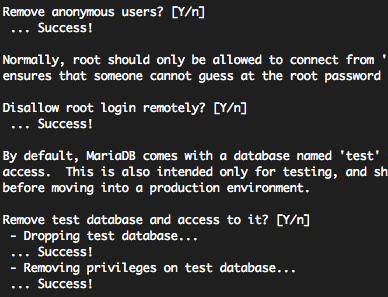
1. Maraidb 설치

# yum install mariadb

# yum install mariadb-server

# service mariadb start // MariaDB 시작

# /usr/bin/mysql\_secure\_installation // MaraiDB 기본 환경설정 (비밀번호만 설정 후 나머지 Y)



# service mariadb restart // MariaDB 재시작 (변경사항 적용)

1. 리눅스 보안강화 모듈 해제

# vi /etc/sysconfig/selinux

SELINUX=enforcing -> SELNIUX=disabled 로 변경

SELinux(Security-Enhanced Linux)란? 직역하면 보안을 강화한 리눅스이다.

과거 리눅스는 소스코드가 공개되어 있기 때문에 보안에 취약하다는 평을 많이 받았다.

그 중 취약한 이유는 설치 이후 보안에 필요한 설정이 되어있지 않았기 때문이다.

이러한 취약점을 보안하기 위한 모델이 SELinux으로, 미국 국방부 스타일의 강제 접근 제어(MAC)를 포함한

접근 제어 보안 정책을 지원하는 매커니즘을 제공하는 리눅스 커널 보안 모듈이다.

SELinux는 매우 정밀한 상세 설정을 포함하여 어떠한 활동을 시스템이 개별 사용자, 프로세스, 데몬에 허용할

것인지를 잠재적으로 제어할 수 있다. 일반적으로 사용자 프로세스는 주로 SELimux에 제한되지 않는 영역에서

실행되지만 고전적인 리눅스 접근 권한에 제한을 받는다.

이러한 SELinux의 상태는 3가지가 있다. Enforcing(강제), permissive(허용), disabled(비활성화)이다.

최초 설치 시 SELinux는 enforcing 되어 있다.

하지만 이 SELinux로 특정 서비스가 동작하지 않게 되는 경우가 많다. 이러할 때는 SELinux를 끄기보다는 해당 서비스가 SELinux 하에서 잘 동작하도록 설정하는 것이 보안적으로 가장 좋지만 이러한 보안 정책 설정은

복잡하고 방화벽을 통해 따로 접근 제어를 하기 때문에 보통 서버 OS를 설치한 후 바로 disabled 시킨다.

( 다른 보안 정책을 이용해 보안 강화 ex : Iptables )

일시적으로 비활성화 하는 법

# setenforce 0

다시 활성화 하는 법

# setenforce 1

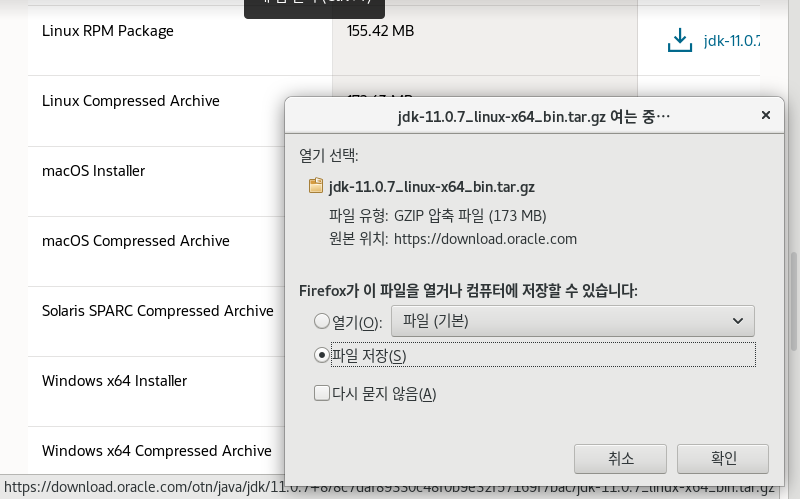
영구적으로 비활성화 하는 법

# vi /etc/sysconfig/selinux

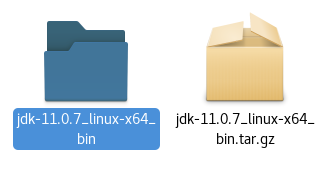
SELINUX=enforcing -> SELNIUX=disabled 로 변경

1. Jdk 설치

<https://www.oracle.com/java/technologies/javase-jdk11-downloads.html> 에서 Linux Compress Archive 다운로드



다운로드 폴더에 가서 jdk-11.0.7\_linux-x64\_bin.tar.gz 파일의 압축해제



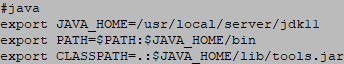
선택된 jdk-11.0.7\_linux-x64\_bin 폴더 안의 jdk11.0.7 폴더를 /usr/local/server 로 옮겨준다.

# cp -r /home/사용자/다운로드/jdk-11.0.7\_linux-x64\_bin/jdk-11.0.7 /usr/local/server (사용자에는 본인 사용자)

설정을 편하게 하기 위해 이름을 간편히 바꾸어준다.

# mv jdk-11.0.7 jdk11 (경로 /usr/local/server에서)

# vi /etc/profile



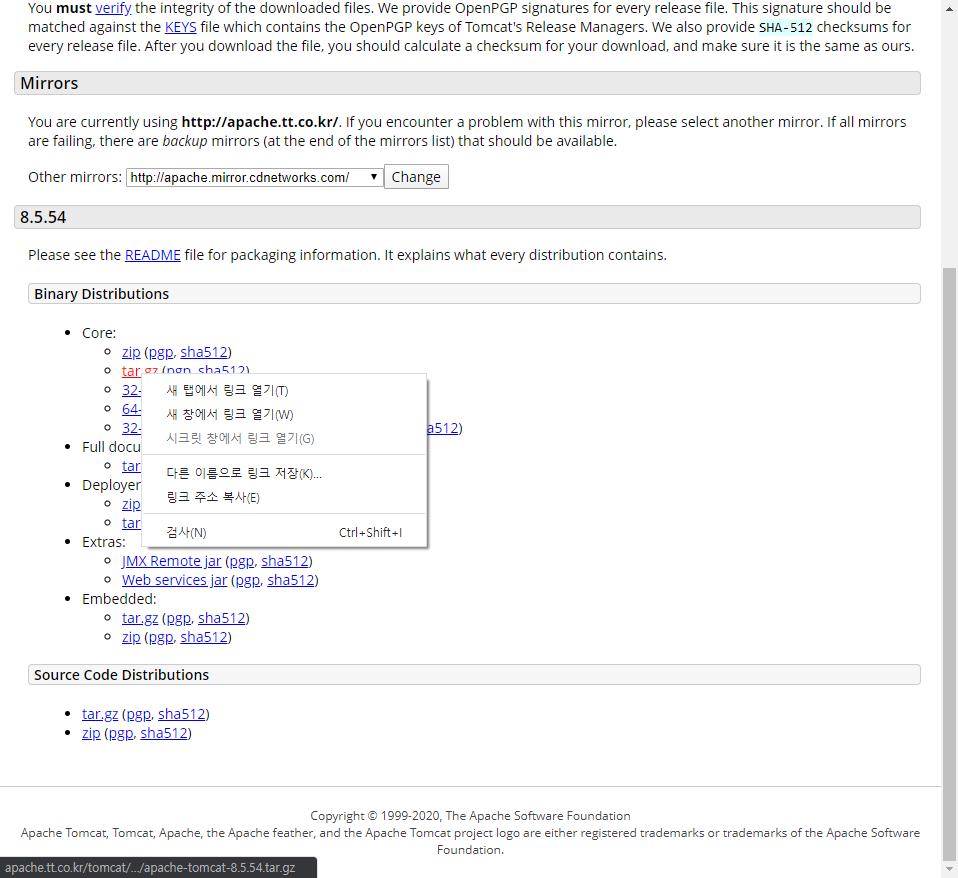
위의 내용을 맨 하단에 추가하고 저장해준다.

# source /etc/profile // 변경사항 적용

* wget을 쓰지 않은 이유 : 파일이 깨져서 다운로드 받아짐

1. Tomcat 설치

<https://tomcat.apache.org/download-80.cgi> 다운로드 링크 주소 복사



# wget <http://apache.tt.co.kr/tomcat/tomcat-8/v8.5.54/bin/apache-tomcat-8.5.54.tar.gz> // 다운로드

# tar xvfz apache-tomcat-8.5.54.tar.gz // 압축해제

# mv apache-tomcat-8.5.54 /usr/local/server/tomcat8.5.54 // 파일이동

# vi /etc/profile // 환경변수 설정



위의 내용을 맨 하단에 추가하고 저장해준다.

# source /etc/profile // 변경사항 적용

# cd /usr/local/server/tomcat8.5.54/webapps

# mkdir test // jsp를 테스트할 디렉토리 생성

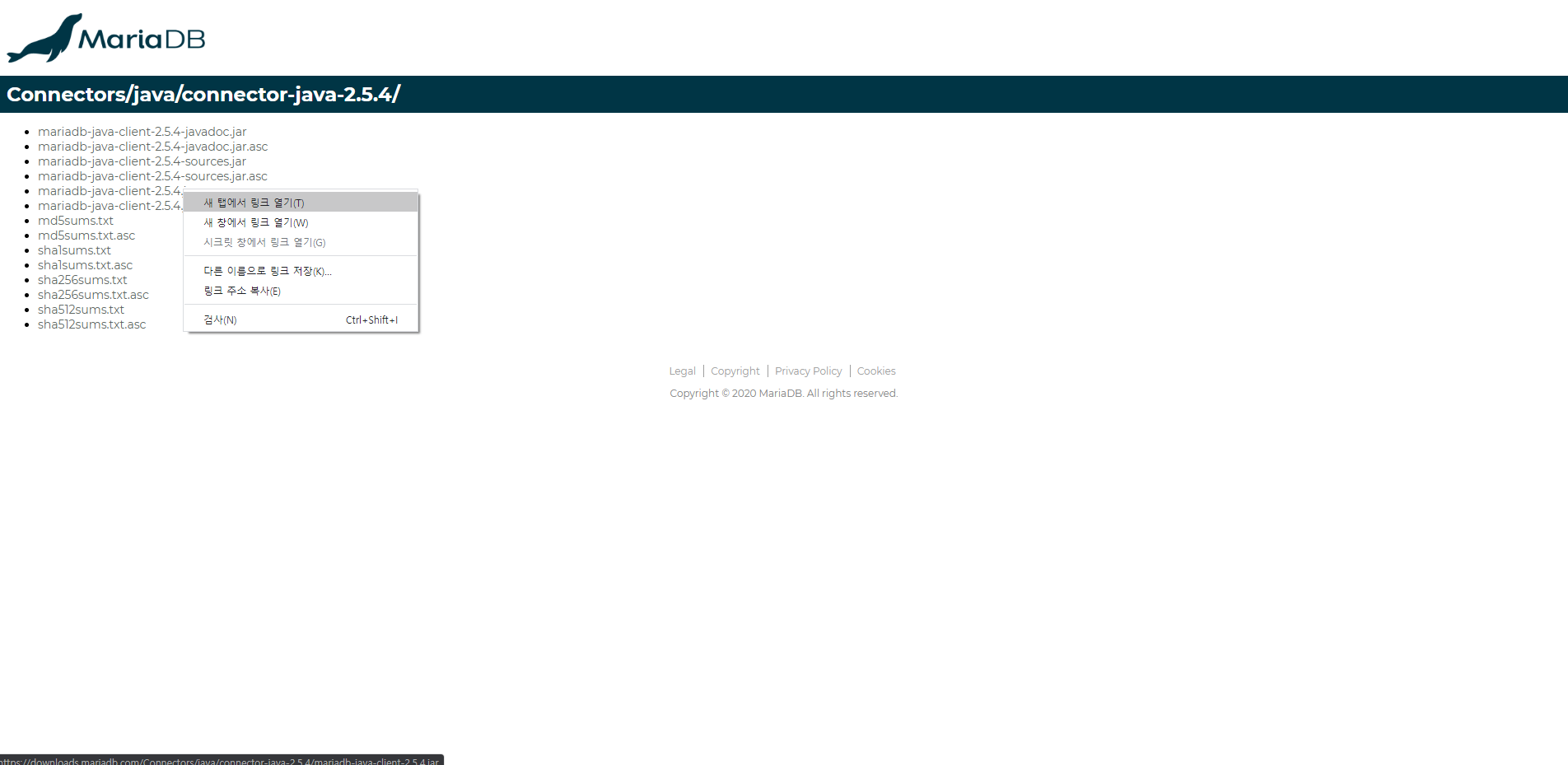
# cd ROOT

# cp -r WEB-INF ../test // jsp를 위해 반드시 필요한 WEB-INF를 test 디렉토리에 복사

# /usr/local/server/tomcat8.5.54/bin/startup.sh // 톰캣 실행

1. Driver 설치 (Connector/J)

<https://downloads.mariadb.org/connector-java/2.5.4/> 다운로드 링크 주소 복사



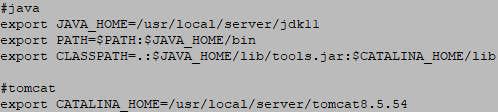
# wget https://downloads.mariadb.com/Connectors/java/connector-java-2.5.4/mariadb-java-client-2.5.4.jar

# mv mariadb-java-client-2.5.4.jar /usr/local/server/tomcat8.5.54/lib // 톰캣의 라이브러리 폴더로 이동

# cd /usr/local/server/tomcat8.5.54/lib

# jar -xvf mariadb-java-client-2.5.4.jar // 압축해제

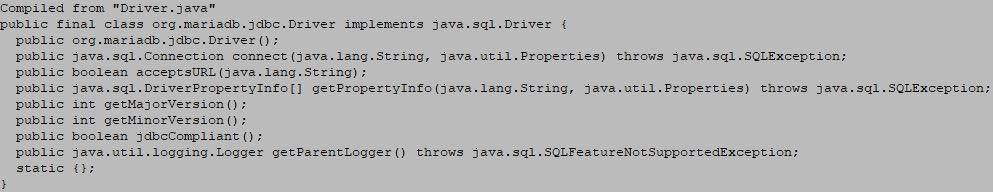
# vi /etc/profile



CLASSPATH에 $CATALINA\_HOME/lib 경로를 추가

# source /etc/profile // 변경사항 적용

# javap org.mariadb.jdbc.Driver // 드라이브 체크



1. Test

# mysql -u root -p

Enter password :

MariaDB [(none)]> CREATE DATABASE test; // 데이터베이스 생성

MariaDB [(none)]> USE test; // test 데이터베이스 사용

MariaDB [test]> quit

<%@ page language=*"java"* contentType=*"text/html;charset=UTF-8"* pageEncoding=*"UTF-8"* import=*"java.sql.\*"*%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>

<title>DB Connection Test</title>

</head>

<body>

       <%

              String DB\_URL = "jdbc:mariadb://localhost:3306/test";

              String DB\_USER = "root";

              String DB\_PASSWORD = "본인 DB 패스워드";

              Connection conn;

              Statement stmt;

              PreparedStatement ps;

              ResultSet rs;

**try** {

                     Class.forName("org.mariadb.jdbc.Driver");

                     conn = DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

                     stmt = conn.createStatement();

                     conn.close();

                     out.println("MySQL JDBC Driver Connection Test Success!!!");

              } **catch** (Exception e) {

                     out.println("MySQL JDBC Driver Connection Test Failed!!!");

              }

       %>

</body>

</html>

# cd /usr/local/server/tomcat8.5.54/webapps/test

# vi test.jsp

위의 코드를 넣고 저장한다.

인터넷 창을 열고 localhost:8080/test/test.jsp 페이지를 접속해 MySQL JDBC Driver Connection Test Success!!!

의 문구가 뜬다면 정상적으로 연동이 된 것이다.