

23-10-05

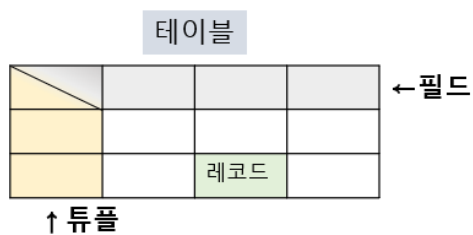
- Maria DB, mySQL 설치할 것
- DB 와 DB 명령어 내용에 대해 배움

<DB_DataBase>

DB는

- 데이터 저장하는 것이다.
- 엑셀시트 칸처럼 데이터 집어 넣는 것이다.
- 집어 넣을 수 있는 데이터가 문자뿐만 아니라 파일도 가능하다
- 일단 문자 데이터 형태로 집어 넣을 것이다.
- 사칙연산(+,-,x,/) 되지 않는 건 모두 문자다 -> 컴퓨터는 문자로 처리

DB는 수치자료와 문자열 자료를 저장하는 프로그램이다.



데이터 하나만 '레코드'라고 기억

묶음 전체 양을 '필드'라고 함

자료

1. 수치

123번 : 123은 숫자 + 번은 문자 -> 문자 => 수치자료

A. 정수 (int)

B. 실수 (허수, 복소수, 소수)

2. 문자

DB

1. 마리아DB
2. mySQL
3. 오라클DB -> 로그인 해야됨

<maria DB>

α => 사내 오류 검증 / 오류 검증이 안끝났다는 것 (사용하지 않는다.)

β => 사용자 오류 검증

RC

LT버전 => 프로그램 다운 받을 때 이 버전 받을 것

X86-64 : 호환계열 / 하위호환성 되기 때문에 64bit에 32bit 다운해도 구동 된다.

마리아DB 오픈소스 -> org로 가서 다운받아야 한다

버전 : 10.4.31

☒ **Use UTF8 as default 필수체크** -> 한글쓰려면 있어야 한다.

☒ **TCP port 번호 : 3306** //마리아 DB가 자료주고받을 때 통로의 번호이다.

MariaDB에 포트번호 3306잡았으면, mySQL에서는 3306을 포트번호로 사용할 수 없음.

HeidiSQL은 마리아DB 관리하기 위한 것이다.

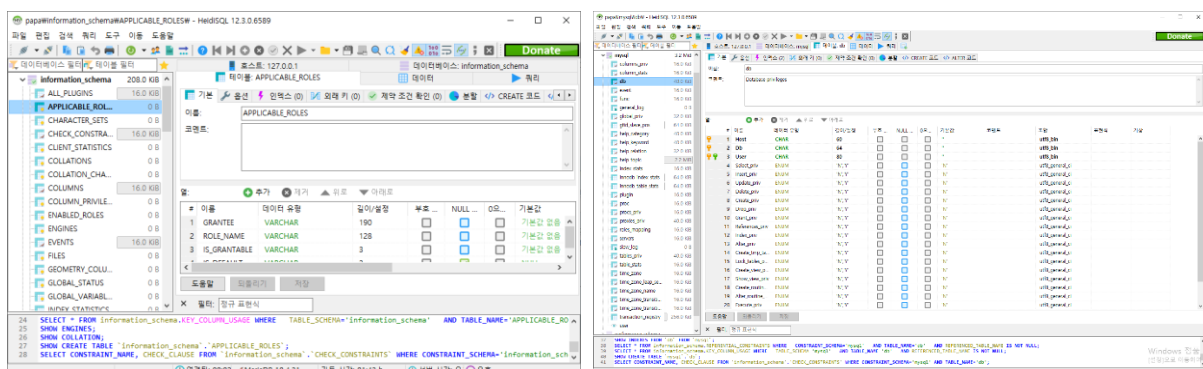
* 기술문서IEEE에 port입력"TCP/UDP의 포트목록" / [MariaDB Documetation 공식문서 열어보기](#)

포트는 기본 하나씩 사용가능

- 포트 몇 개까지 사용가능? / 0~65535번 까지 가능 (49152~65535는 동적포트)
- 픽스된 번호 존재? / 0~49151번은 픽스된 번호이다. (80이 http의 포트번호)
- 제너럴 포트존재?

⇒ Port index table 찾으면 끝남

mariaDB 설치되어 있는 내 컴퓨터 ip -> 127.0.0.1



HeidiSQL 실행창

<mySQL>

버전 : 8.0.34 / web없는 경로 설치

5.7.43 버전은 mySQL이 단독개발하던 시점 -> 마리아DB 어떤 버전이든 호환한다, mySQL 만든 사람들이 마리아DB 만들었기 때문에

- > server only는 가상 머신에 설치할 것
- > custom은 원하는 것만 설치하는 것
- > Full은 모두 설치하는 것

환경이 달라지면 세팅값 다 설치해야 하고 안맞을 수 있다

<DB 문법 (mySQL과 mariaDB의 문법은 같다)>

이름 -> 컬럼에 해당한다	데이터유형 -> 자료형(실수, 정수, 문자)	길이 -> 자료의 크기이다 -> 레코드명	기본값

Null값 : 빈값이다 -> 0이 아니다. 컴퓨터에게 0은 값이 있는 것, null은 아무값도 들어있지 않음

각 부등호는 프로그램마다 표현하는 기호가 다르다 -> 컴퓨터에서 같지 않다.

3자이면 -> 0~999까지 사용가능

Information – schema가 데이터베이스 하나이다.

<MariaDB>

Mariadb의 명령프롬프트 = SQL

Mariadb -u root -p //mariaDB

-u는 사용자이름 적겠다, -p는 password

```
Command Prompt (MariaDB 10.4 (x64)) - mariadb -u root -p
Setting environment for MariaDB 10.4 (x64)
C:\Windows\System32>mariadb -u root -p
Enter password:
```

mySQL -u root -p //mySQL

→에러 : 마무리 안하고 다음줄로 넘어간 것, 뒤에 더 쓸게 있다는 것.

Ctrl + C : exit! / continue -> 브레이크의미와 계속하라는 의미 동시에 있다.

↑(화살표키) : 이것을 입력하면 이전에 입력한 명령어가 입력됨

➔ 임시저장소 기능이 있어서 작업하던 것을 임시로 기억하고 있다

한번 사용했던 것 다시 사용할 수 있다

➔ 꺾다 다시 키면 ↑ 눌러도 이전 명령어 나오지 않음, 프롬프트 작동할 때만 기억함

Show databases;

show schemas;

show 데이터 베이스 목록 보여주는 것, 각각 데이터 베이스 있는 데이터 베이스 목록 보여줌.

```
MariaDB [(none)]> show database
-> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'database' at line 1
MariaDB [(none)]> show database;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'database' at line 1
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| test           |
+-----+
4 rows in set (0.001 sec)
```

명령어가 길어지면 엔터(→)해서 연결하기

Use mysql;

mysql이라는 데이터베이스 안으로 들어가기 [(none)]에서 [mysql]로 된다.

Show tables;

테이블 보여주는 것.

Exit;

나가기

<mySQL>

바로 password 입력

- DB 구동하기 위해 필수 필요한 것들이라 아래는 건들이지 않는다.

Information-schema

mySQL

performance-schema

Show tables;

table 보여줌, mariaDB와 문법동일하다.

<mariaDB와 mySQL의 프로그램>

mySQL에는 workbench가 있고, mariaDB에는 HeidiSQL이 있다.

- Workbench

: ERD 작성 유리, DB 동기화가 편함

- HeidiSQL

: 연결 보는 것이 편함, 자료 입력 수정 변경 등의 확인이 용이함

- 명령프롬프트

: CLI => DB 환경 설정 시, 권한 설정 시, 리눅스 기반 설정 및 설치 할 때 사용

DB 사용하는 것은 중복되는 거 줄이도록 하는 것이다.

중복되는 것 빼고 각각의 키, 프라이머리키 지정하는 것이다.

앱을 지워도 조각(쿠키)이 디렉토리>program files에 남아있다. 그래서 싹 포맷하는 것.

<컴퓨터의 세상 3단계>



사용자 input 값(2+2) 입력 -> 엑셀로 보냄 -> 운영체제에 저장 -> 저장경로는 주기억장치인 RAM으로 전달 -> CPU에게 계산하게 보내줌, 결과 값 나오게끔(결과값 나온 것 CPU에는 저장이 안된다) -> CPU에서 계산된 내용을 무슨 계산의 결과 인지, 어떤 결과인지 값을 내고 -> RAM으로 결과 값을 보내준다(결과값을 CPU에는 저장이 안되서 RAM에 보내주고 저장 장치 SSD에 저장하는 것이다) -> CPU의 계산값을 RAM 어디에 저장하자 라는 명령을 내려줌 OS(운영체제)는 이 단계에 개입 -> SSD로 전달하여 SSD어디에 저장하라고 전달

- 가상화

JVM(자바가상머신) : os의 일정부분 위에 있는 부분

Window 도스 체계, 운영체제 다른데 똑같이 돌아간다.

왜 만들었을까?

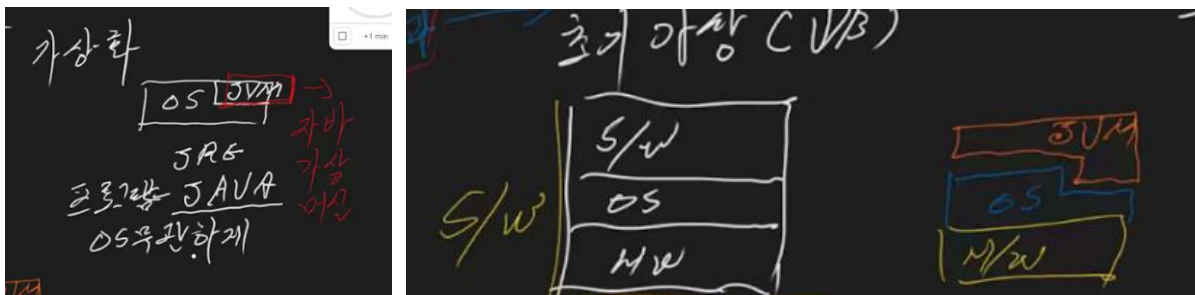
-> 가상화면은 OS와 S/W 사이에 걸쳐있다. 어디 운영체제 다 작동하는 것이다. S/W 안에 가상화면으로 H/w, OS, S/W가 있는 것이다. / 처음컴퓨터는 호환성이 없었다.

-> Java를 어디든 실행하게 해주는 것이 JVM이다.

-> JVM이 어떤 OS든 상관없이 통신함.

-> JVM은 기계장치지만 가상의 기계장치(물리적으로 존재하지 않지만, 논리적으로는 존재한다.)

-> S/W로 만들어 놓은 것, 다 없는데도 컨트롤하고 관리할 수 있다.



⇒ 이렇게 해서 JVM이 나온 것.

JVM -> JRE -> JAVA 가 나옴

⇒ 프로그램이 OS하고 무관하게 어떤 운영체제든 상관없이 JVM이 설치되면 JAVA가 실행됨

JVM : 자바 가상머신은 운영체제(window,리눅스)에 상관없이 자바프로그램을 실행시키기 위한 것, 운영체제가 달라도 프로그램이 실행됨.

<TCP/UDP의 포트목록>

막히면 공식문서부터 열어보기

Git에 문서에 단어들 다 나와있다.

블로그 신뢰하지 말기, 공식문서가 아니다

➔ 맞는 것도 틀린 것도 있을 것.

위키백과

무의 도두의 백과사전

위키백과 검색

검색

백지 [문기]

저위

잘 알려진 포트

동독된 포트

TCP/UDP의 포트 목록

23개 언어

문서 토론

위키백과, 우리 모두의 백과사전

잘 알려진 포트(well-known port)는 특정한 쓰임새를 위해서 IANA에서 할당한 TCP 및 UDP 포트 번호의 일부이다. 일반적으로 포트 번호는 다음과 같이 세 가지로 나눌 수 있다.

- 0번 ~ 1023번: 잘 알려진 포트 (well-known port)
- 1024번 ~ 49151번: 등록된 포트 (registered port)
- 49152번 ~ 65535번: 동적 포트 (dynamic port)

잘 알려진 포트 [편집]

대부분의 유닉스 계열 운영 체제의 경우, 잘 알려진 포트를 알려진 루트 권한이 있어야 한다.

이 번호는 강제적으로 지정된 것은 아니며, IANA의 권고사항 뿐이다. 가끔 각 포트 번호를 그대로 사용하지 않고 다른 용도로 사용하기도 한다. 이 경우, 트로이 목마와 같은 프로그램들이 악의적인 목적으로 포트를 변경하여 사용하는 경우도 있다.

포트	TCP	UDP	설명	상태
0		UDP	예약됨; 사용하지 않음	공식
1	TCP		TCPMUX (TCP 포트 서비스 멀티플렉서)	공식
7	TCP	UDP	ECHO 프로토콜	공식
9	TCP	UDP	DISCARD 프로토콜	공식
13	TCP	UDP	DAYTIME 프로토콜	공식
17	TCP		QOTD (Quote of the Day) 프로토콜	공식
19	TCP	UDP	CHARGEN (Character Generator) 프로토콜 - 원격 오류 수정	공식
20	TCP		파일 전송 프로토콜 (FTP: File Transfer Protocol) - 데이터 포트	공식
21	TCP		파일 전송 프로토콜 (FTP: File Transfer Protocol) - 제어 포트	공식
22	TCP		시큐어 쉘 (SSH, Secure Shell) - ssh scp, sftp같은 프로토콜 및 포트 포워딩	공식
23	TCP		텔넷 프로토콜 (Telnet Protocol) - 암호화되지 않은 텍스트 통신	공식