



인하공업전문대학
INHA TECHNICAL COLLEGE

서버구축실습

2023년 2학기



서버 구축 실습 STACK

리눅스 네트워크
서비스 관리

리눅스 서버

네트워크 설정

텔넷 SSH

FTP

DB

Apache(웹서버)

NFS & SAMBA

리눅스
시스템 관리

디스크관리

패키지 관리

사용자 관리

리눅스
기본 명령

디렉터리와
파일

문서
편집

셸사용

접근
권한
설정

프로세스
관리

리눅스
개요 및 설치

리눅스 개요

AWS 리눅스
인스턴스 생성

명령 사용 환경

01 데이터베이스

01. 데이터베이스

■ 데이터베이스

- 서로 관련성 있는 데이터들을 데이터 간의 중복성을 최소화해서 체계적으로 모아놓은 것
- 데이터를 테이블 구조로 구성하는 관계형 데이터베이스를 기준으로 학습

■ 관계형 데이터베이스의 기본 용어

- 데이터: 각 항목에 저장되는 값
- 테이블: 데이터를 체계화하여 행과 열의 형태로 구성한 것으로 테이블 이름 지정
- 데이터베이스: 데이터는 테이블 형태로 저장되며 하나 이상의 테이블이 있을 수 있음
- 필드: 테이블의 열을 의미하며 칼럼이라고도 함 **col**
- 레코드: 테이블에 저장한 한 행의 정보를 의미하며 튜플 tuple 이라고도 함 **ROW**
- 키: 레코드를 구성하는 필드에서 각 레코드를 구분할 수 있는 필드 값을 의미
 - 예: 학생 데이터를 구성하는 레코드라면 학번 필드가 키가 될 수 있음
 - 키로 선택된 필드의 값은 중복될 수 없으며 기본 키 primary key 와 그 외 다른 키가 있을 수 있음

01. 데이터베이스

■ 관계형 데이터베이스의 예

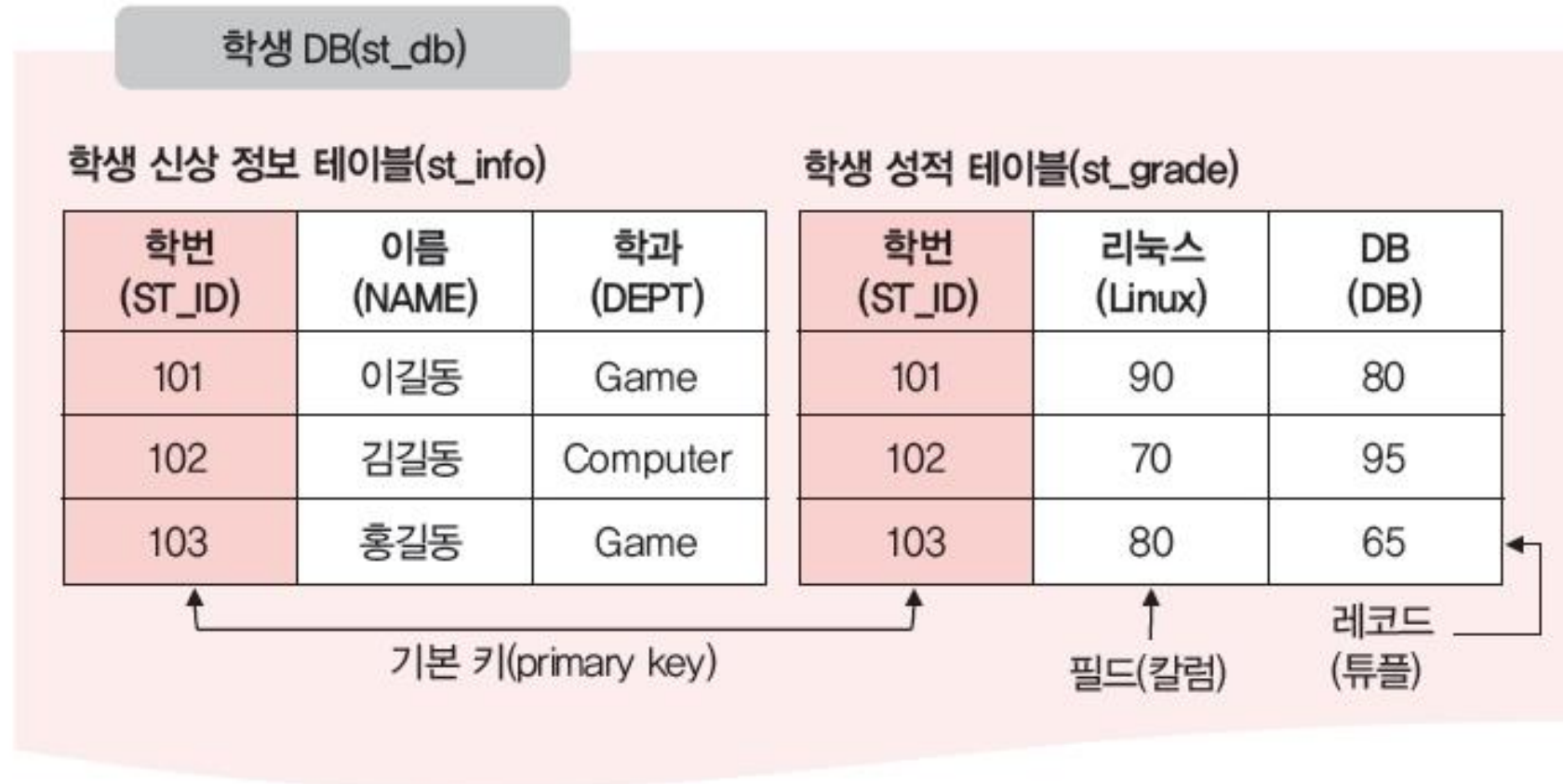


그림 13-1 학생 데이터베이스의 예

01. 데이터베이스

■ SQL 기초

- 관계형 데이터베이스를 생성하고, 테이블을 만들고, 데이터를 입력·삭제·수정할 때 사용하는 언어

■ 데이터베이스 관련 SQL

- 데이터베이스 목록 확인: 기존 데이터베이스 목록을 출력
- 데이터베이스 생성: 새로운 데이터베이스를 생성
- 데이터베이스 삭제: 지정한 데이터베이스를 삭제
- 사용할 데이터베이스 지정: 여러 데이터베이스 중에서 작업할 데이터베이스를 선택

```
show databases;
```

```
create database 데이터베이스명;
```

```
drop database 데이터베이스명;
```

```
use 데이터베이스명;
```

01. 데이터베이스

■ 테이블 관련 SQL

- 테이블 목록 확인: 현재 데이터베이스에 있는 테이블의 목록을 출력
- 테이블 생성: 필드에 사용할 수 있는 자료형을 알아야 함

```
show tables;
```

```
create table 테이블명(필드명1 필드명1의 자료형, 필드명2 필드명2의 자료형, ...);
```

이름 varchar(10)

표 13-1 테이블 필드의 자료형

자료형	설명	예
varchar(n)	최대 n개의 크기를 가진 가변 문자열	varchar(10)
char(n)	n개의 크기를 가진 고정 문자열	char(5)
int	정수형 숫자	10
float	실수형 숫자	10.1
date	날짜	2021-08-09
time	시간	02:56:30

01. 데이터베이스

- 테이블 구조 확인: 지정한 테이블의 구조를 출력

```
explain 테이블명;
```

- 테이블 수정: 테이블의 구조를 변경

```
alter table 테이블명 수정 명령;
```

- 테이블에 필드 추가: alter table 테이블명 add 필드명 자료형

예) alter table st_info add AGE int;

- 필드의 자료형 변경: alter table 테이블명 modify 필드명 자료형

예) alter table st_info modify AGE float;

- 필드 삭제: alter table 테이블명 drop column 필드명

예) alter table st_info drop column AGE;

- 기본 키 추가: alter table 테이블명 add constraint 제약 조건명 primary key (필드명)

예) alter table st_info add constraint pk_stinfo primary key (ST_ID);

제약 조건명은 사용자가 정하면 된다. 처음 테이블을 생성하면서 기본 키를 지정할 때 다음과 같이 한다.

예) create table st_info (ST_ID int Not NULL primary key, NAME varchar(20),
DEPT varchar(25));

테이블 삭제: 지정한 테이블을 삭제

```
drop table 테이블명;
```


01. 데이터베이스

■ 레코드 삽입·삭제·수정 관련 SQL

- 레코드 삽입: 테이블에 새로운 레코드를 추가 `insert into 테이블명 values (값1, 값2, ...);`
 - 예: `insert into st_info values (101, '이길동', 'Game');`
- 레코드 수정: 기존 레코드를 수정
`update 테이블명 set 필드명1=수정할 값1, 필드명2=수정할 값2, ... where 조건;`
 - 예: `update st_info set DEPT='Computer' where ST_ID=103;`
- 레코드 삭제: 테이블에서 해당 레코드를 삭제 `delete from 테이블명 where 조건;`
 - 예: `delete from st_info where ST_ID=103;`

01. 데이터베이스

■ 레코드 검색 관련 SQL: select 문 사용

```
select 필드명1, 필드명2, ... from 테이블명 where 조건;
```

완전 중요

- ❶ select * from st_info;
- ❷ select NAME from st_info where ST_ID=101;
- ❸ select Linux, DB from st_grade where ST_ID=101;

01. 데이터베이스

■ 접근 권한 관련 SQL

- 데이터베이스에 대한 모든 권한을 해당 사용자에게 부여

```
grant all privileges on DB명.* to 사용자 계정@localhost identified by '패스워드';
```

DB명의 모든 테이블

- 예:

```
grant all privileges on st_db.* to user1@localhost identified by '123456';
```

02 MariaDB 설치와 사용

02. MariaDB 설치와 사용

■ MariaDB 서버는 apt 명령으로 설치

```
user1@myubuntu:~$ sudo apt install mariadb-server
```

■ MariaDB 활성화하기

```
user1@myubuntu:~$ systemctl status mariadb.service
● mariadb.service - MariaDB 10.5.13 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset:
enabled)
   Active: active (running) since Thu 2022-01-06 19:56:14 KST; 1min 12s ago
     Docs: man:mariadb(8)
          https://mariadb.com/kb/en/library/systemd/
  Process: 46342 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/
run/mysqld (c)
  Process: 46344 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_
POSITION (>
```

02. MariaDB 설치와 사용

■ MariaDB 시작하기

```
user1@myubuntu:~$ mysql
ERROR 1045 (28000): Access denied for user 'user1'@'localhost' (using password: NO)
user1@myubuntu:~$ mariadb
ERROR 1045 (28000): Access denied for user 'user1'@'localhost' (using password: NO)
user1@myubuntu:~$ sudo mariadb
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 46
Server version: 10.5.13-MariaDB-0ubuntu0.21.10.1 Ubuntu 21.10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

mysql 종료는 exit 입력

02. MariaDB 설치와 사용

■ 데이터베이스 목록 확인하기

```
MariaDB [(none)]> show databases;
```

```
+-----+
```

```
| Database |
```

```
+-----+
```

```
| information_schema |
```

```
| mysql |
```

```
| performance_schema |
```

```
+-----+
```

```
3 rows in set (0.002 sec)
```

02. MariaDB 설치와 사용

■ 데이터베이스 생성 및 사용하기

```
MariaDB [(none)]> create database st_db;  
Query OK, 1 row affected (0.001 sec)
```

```
MariaDB [(none)]> show databases;
```

```
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| st_db |  
+-----+  
4 rows in set (0.001 sec)
```

```
MariaDB [(none)]> use st_db;  
Database changed  
MariaDB [st_db]>
```


02. MariaDB 설치와 사용

■ 테이블 생성 및 구조 확인하기

```
MariaDB [st_db]> show tables;  
Empty set (0.001 sec)
```

■ 테이블 생성하기

```
MariaDB [st_db]> create table st_info (ST_ID int, NAME varchar(20), DEPT varchar(25))  
default charset=utf8;  
Query OK, 0 rows affected (0.006 sec)
```

```
MariaDB [st_db]> create table st_grade (ST_ID int, Linux int, DB int);  
Query OK, 0 rows affected (0.004 sec)
```

02. MariaDB 설치와 사용

- show tables 명령을 사용하여 테이블이 정상적으로 생성되었는지 확인

```
MariaDB [st_db]> show tables;
```

```
+-----+
```

```
| Tables_in_st_db |
```

```
+-----+
```

```
| st_grade      |
```

```
| st_info       |
```

```
+-----+
```

```
2 rows in set (0.001 sec)
```

02. MariaDB 설치와 사용

- 테이블의 구조를 확인

```
MariaDB [st_db]> explain st_info;
```

Field	Type	Null	Key	Default	Extra
ST_ID	int(11)	YES		NULL	
NAME	varchar(20)	YES		NULL	
DEPT	varchar(25)	YES		NULL	

3 rows in set (0.004 sec)

```
MariaDB [st_db]> explain st_grade;
```

Field	Type	Null	Key	Default	Extra
ST_ID	int(11)	YES		NULL	
Linux	int(11)	YES		NULL	
DB	int(11)	YES		NULL	

3 rows in set (0.001 sec)

02. MariaDB 설치와 사용

- 기본 키 추가하기

```
MariaDB [st_db]> alter table st_info modify ST_ID int Not Null;
```

```
Query OK, 0 rows affected (0.012 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [st_db]> alter table st_grade modify ST_ID int Not Null;
```

```
Query OK, 0 rows affected (0.010 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

- ST_ID 필드를 기본 키로 설정

```
MariaDB [st_db]> alter table st_info add constraint pk_stinfo primary key(ST_ID);
```

```
Query OK, 0 rows affected (0.011 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [st_db]> alter table st_grade add constraint pk_stgrade primary key(ST_ID);
```

```
Query OK, 0 rows affected (0.013 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

02. MariaDB 설치와 사용

■ 레코드 입력하기

```
MariaDB [st_db]> insert into st_info values (101, '이길동', 'Game');  
Query OK, 1 row affected, 1 warning (0.002 sec)
```

```
MariaDB [st_db]> insert into st_info values (102, '김길동', 'Computer');  
Query OK, 1 row affected, 1 warning (0.002 sec)
```

텍스트

```
MariaDB [st_db]> insert into st_info values (103, '홍길동', 'Game');  
Query OK, 1 row affected, 1 warning (0.001 sec)
```

```
MariaDB [st_db]> insert into st_grade values (101, 90, 80);  
Query OK, 1 row affected (0.001 sec)
```

```
MariaDB [st_db]> insert into st_grade values (102, 70, 95);  
Query OK, 1 row affected (0.002 sec)
```

```
MariaDB [st_db]l> insert into st_grade values (103, 80, 65);  
Query OK, 1 row affected (0.001 sec)
```

02. MariaDB 설치와 사용

■ 레코드 검색하기

- st_info 테이블의 전체 레코드를 출력

```
MariaDB [st_db]> select * from st_info;
```

ST_ID	NAME	DEPT
101	이길동	Game
102	김길동	Computer
103	홍길동	Game

```
3 rows in set (0.001 sec)
```

02. MariaDB 설치와 사용

- 학번이 101인 학생의 이름과 학과를 검색

```
MariaDB [st_db]> select NAME, DEPT from st_info where ST_ID=101;
```

```
+-----+-----+
```

```
| NAME      | DEPT |
```

```
+-----+-----+
```

```
| 이길동    | Game |
```

```
+-----+-----+
```

```
1 row in set (0.001 sec)
```

02. MariaDB 설치와 사용

- 학번이 101인 학생의 리눅스 성적을 검색

```
MariaDB [st_db]> select Linux from st_grade where ST_ID=101;
+-----+
| Linux |
+-----+
|    90 |
+-----+
1 row in set (0.001 sec)
```


02. MariaDB 설치와 사용

- 학번이 101인 학생의 이름과 학과, DB 성적을 한 번에 검색

```
MariaDB [st_db]> select st_info.NAME, st_info.DEPT, st_grade.DB  
-> from st_info, st_grade  
-> where st_info.ST_ID=101 and st_grade.ST_ID=101;
```

```
+-----+-----+-----+  
| NAME      | DEPT | DB  |  
+-----+-----+-----+  
| 이길동    | Game | 80  |  
+-----+-----+-----+  
1 row in set (0.000 sec)
```

02. MariaDB 설치와 사용

■ 데이터 수정하기

```
MariaDB [st_db]> update st_grade set DB=90 where ST_ID=101;  
Query OK, 1 row affected (0.002 sec)  
Rows matched: 1   Changed: 1   Warnings: 0
```

```
MariaDB [st_db]> select * from st_grade where ST_ID=101;  
+-----+-----+-----+  
| ST_ID | Linux | DB   |  
+-----+-----+-----+  
| 101   | 90    | 90   |  
+-----+-----+-----+  
1 row in set (0.000 sec)
```

02. MariaDB 설치와 사용

■ MariaDB 관리

mysqladmin

- **기능** MariaDB 서버를 관리한다.
- **형식** mysqladmin [옵션]
- **옵션** version: MariaDB 서버의 버전 정보를 출력한다.
status: MariaDB 서버의 현재 상태 정보를 출력한다.
password 암호: 계정의 암호를 지정한다.
- **사용 예** mysqladmin password "123456"

02. MariaDB 설치와 사용

■ MariaDB 상태 정보 출력하기

```
user1@myubuntu:~$ sudo mysqladmin status
Uptime: 1321  Threads: 1  Questions: 519  Slow queries: 0  Opens: 177  Open tables:
30  Queries per second avg: 0.392
```

- Uptime: 서버가 동작한 시간을 초 단위로 표시
- Threads: 현재 동작 중인 MariaDB 서버의 스레드 수
- Questions: 서버가 동작한 이후 처리한 질의 수
- Slow queries: 일정 시간보다 처리 시간이 길어진 질의 수
- Opens: 서버가 열었던 테이블 수
- Flush tables: flush, refresh, reload 명령을 수행한 횟수
- Open tables: 현재 열려 있는 테이블 수

02. MariaDB 설치와 사용

- MariaDB 버전 정보 출력하기

```
user1@myubuntu:~$ sudo mysqladmin version
mysqladmin  Ver 9.1 Distrib 10.5.13-MariaDB, for debian-linux-gnu on x86_64
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Server version          10.5.13-MariaDB-0ubuntu0.21.10.1
Protocol version        10
Connection              Localhost via UNIX socket
UNIX socket             /run/mysqld/mysqld.sock
Uptime:                 22 min 36 sec

Threads: 1  Questions: 520  Slow queries: 0  Opens: 177  Open tables: 30  Queries
per second avg: 0.383
```

02. MariaDB 설치와 사용

- MariaDB 서버 암호 설정하기

```
user1@myubuntu:~$ sudo mysqladmin password '123456'
```

- 암호를 입력하기 위해 -u 옵션으로 계정 이름을 지정하고, -p 옵션으로 암호를 입력할 것임을 알려줌

```
user1@myubuntu:~$ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 52
Server version: 10.5.13-MariaDB-0ubuntu0.21.10.1 Ubuntu 21.10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

02. MariaDB 설치와 사용

- MariaDB 계정 생성하기

- 내부ip(localhost) 접속 가능 계정

CREATE USER '계정아이디'@'localhost' IDENTIFIED BY '비밀번호';

ex) create user 'testId1'@'localhost' identified by 'testPw1';

- 외부ip 접속 가능 계정

CREATE USER '계정아이디'@'%' IDENTIFIED BY '비밀번호';

ex) create user 'testId1'@'%' identified by 'testPw1';

02. MariaDB 설치와 사용

■ 계정 권한 부여

- 모든 DB에 모든 권한 부여

GRANT ALL PRIVILEGES ON *.* TO '계정아이디'@'호스트';

ex) grant all privileges on *.* to 'testId1'@'loalhost';

- 특정 DB에 모든 권한 부여

GRANT ALL PRIVILEGES ON 데이터베이스명.* TO '계정아이디'@'호스트';

ex) grant all privileges on board.* to 'testId1'@'loalhost';

- 특정 DB에 특정 권한 부여

GRANT SELECT, INSERT, UPDATE ON 데이터베이스명.* TO '계정아이디'@'호스트';

ex) grant select, insert, update on board.* to 'testId1'@'loalhost';

- 권한 적용 꼭 해야함

FLUSH PRIVILEGES;

- 권한 부여 확인

SHOW GRANTS FOR '계정아이디'@'호스트';

02. MariaDB 설치와 사용

- 계정 삭제

DROP USER '계정아이디'@'호스트';

ex) drop user 'testId1'@'localhost';

DELETE FROM USER WHERE USER = '계정아이디';

ex) delete from user where user = 'testId1';

688p. 따라해보기: 테이블 생성 및 레코드 입력/검색/수정하기

테이블 이름: st_janghak

- ① st_db 데이터베이스를 그대로 사용
- ② 새로운 테이블을 추가
- ③ st_janghak 테이블의 구조를 확인
- ④ 테이블에 데이터 입력
- ⑤ st_janghak 테이블의 모든 레코드를 검색하여 데이터가 제대로 입력되었는지 확인
- ⑥ st_info 테이블에 나이(Age) 필드를 추가
- ⑦ 새로 추가한 Age 필드에 값을 입력
- ⑧ 학번, 이름, 나이, 리눅스 성적, 장학금이 출력되도록 검색

학번(ST_ID)	장학금(Money)
101	500000
102	1000000
103	1500000