MYSQL QUERY 문법

OneKick

Contents

OneKick

- MySQL에서 지원하는 데이터타입
- CREATE TABLE
- ALTER TABLE
- DROP TABLE
- SELECT
- INSERT
- UPDATE
- DELETE
- MYSQL 데이터베이스의 사용법

MySQL에서 지원하는 데이터타입

OneKick

- INT : 4bytes 정수
- CHAR(n) : 문자의 수가 n개인 문자열
- VARCHAE(n) : 문자의 수가 최대 n개인 문자열
- TEXT : 문자의 수가 최대 65535개인 문자열

CREATE DATABASE

OneKick

- 데이터베이스 생성
  > CREATE DATABASE <DATABASE NAME>;

- 데이터베이스 선택
  > USE <사용할 DB NAME>

```
mysql> CREATE DATABASE docmanager;
mysql> USE docmanager
Database changed
mysql> SHOW TABLES;
Empty set (0.00 sec)
mysql>
```

CREATE TABLE

OneKick

- create table 테이블명 (
  필드명 데이터타입 [NOT NULL]
  {, 필드명 데이터타입 [NOT NULL]}*
  [,PRIMARY KEY (필드명)])

```
mysql> CREATE TABLE  docmanager (title VARCHAR(40),
description
    -> VARCHAR(40),file_name VARCHAR(20),
    -> enrolled DATE, updated DATE, owner VARCHAR(20));
```

등록된 테이블의 확인

```
mysql> DESC <테이블명>
```

ALTER TABLE

OneKick

- ALTER TABLE(테이블구조 및 속성변경)
  - 기존의 테이블구조 변경
  - 필드 추가 생성
  - 필드 속성 변경
  - 필드 삭제

ALTER TABLE

OneKick

- 1. 필드추가생성
  - ALTER TABLE 테이블명 ADD [COLUMN] 필드명 필드타입
  - 예) ALTER TABLE userdb ADD COLUMN addr VARCHAR(100)

ALTER TABLE

OneKick

- 2. 필드속성변경
  - ALTER TABLE 테이블명 CHANGE [COLUMN] 기존필드명 새필드명 필드타입
  - 예)ALTER TABLE userdb ADD addr VARCHAE(100)
  - 예) ALTER TABLE userdb CHANGE COLUMN addr address VARCHAR(255)

ALTER TABLE

OneKick

- 3. 필드삭제
  - ALTER TABLE 테이블명 DROP [COLUMN] 필드명
  - 예) ALTER TABLE userdb DROP COLUMN address

DROP TABLE

OneKick

- 기존의 테이블삭제
- DROP TABLE 테이블명

SELECT

OneKick

- SELECT (데이터검색하기)
  - SELECT [DISTINCT] 테이블명 {, 테이블명}*
    FROM 필드명 {,필드명}*
    [WHERE 검색조건]
    [ORDER BY 필드명[ASC or DESC] {,필드명 [ASC or DESC]}*]
    [GROUP BY 필드명 {, 필드명}*]
    [HAVING 검색조건]

SELECT

OneKick

- 1. 일반적인 검색
  - SELECT name, id FROM userdb WHERE level = 'B'
  - SELCET * FROM userdb
  - SELECT name, id, email, age, milage, level FROM userdb

1-1. 일반적인 검색의 예제

mysql> SELECT * FROM docmanager;

title	decription	file_name	enrolled	updated	owner
MySQL Seminar	MySQL?	MySQL.ppt	2000-07-04	2000-07-06	please

mysql> SELECT * FROM docmanager WHERE owner = “please";
mysql> SELECT owner FROM docmanager;

형식: **SELECT** 컬럼명 ... **from** 테이블명 **[WHERE** 조건**];**

SELECT

OneKick

- 2. 결과레코드의 중복제거
  - SELECT level FROM userdb
  - SELECT DISTINCT level FROM userdb

SELECT

- 3. 조건검색
  - SELECT name, id, email FROM userdb WHERE milage > 3000 AND gender = 'M'

- 4. 검색결과의 정렬
  - SELECT name, id, milage FROM userdb WHERE milage >= 3000 ORDER BY milage DESC

SELECT

- 5. 검색결과에 대한 산술계산 및 문자열처리
  - SELECT name, '님의 마일리지는 ', milage, '점입니다.' FROM userdb WHERE milage >= 3000 ORDER BY milage
  - SELECT name, id, email, milage+300 FROM userdbWHERE id = 'soony'

SELECT

OneKick

- 6. 그룹함수(group function)를 이용한 검색
  - SELECT count(*) FROM userdb WHERE gender = 'F'
  - SELECT avg(milage) FROM userdb WHERE gender='F'

- 7. GROUP BY를 이용한 검색
  - SELECT level, max(milage), min(milage), avg(milage) FROM userdb GROUP BY level
  - SELECT gender, max(milage), min(milage), avg(milage) FROM userdb GROUP BY gender

<GROUP BY를 이용한 예제>

Table : ***Store_Information***

store_name	Sales	Date
Los Angeles	$1500	Jan-05-1999
San Diego	$250	Jan-07-1999
Los Angeles	$300	Jan-08-1999
Boston	$700	Jan-08-1999

- **SELECT store_name, SUM(Sales)**
  FROM Store_Information GROUP BY store_name

store_name	SUM(Sales)
Los Angeles	$1800
San Diego	$250
Boston	$700

- 8. HAVING을 이용한 검색
  - SELECT gender, max(milage), min(milage), avg(milage) FROM userdb GROUP BY gender HAVING gender = 'F'

< HAVING을 이용한 예제>

Table : ***Store_Information***

store_name	Sales	Date
Los Angeles	$1500	Jan-05-1999
San Diego	$250	Jan-07-1999
Los Angeles	$300	Jan-08-1999
Boston	$700	Jan-08-1999

- **SELECT store_name, SUM(Sales) FROM Store_Information GROUP BY store_name HAVING SUM(sales) > 1500**

store_name	SUM(Sales)
Los Angeles	$1800

- 9. BETWEEN 연산자를 이용한 검색
  - SELECT name, email, mailage, gender FROM userdb WHERE gender = 'F' AND milage BETWEEN 3000 AND 4000
  - SELECT name, email, milage, gender FROM userdb WHERE gender = 'F' AND milage >= 3000 AND milage <= 4000
  - SELECT name, gender, milage FROM userdb WHERE milage NOT IN (3300,3500,3700)

SELECT

OneKick

- 10. LIKE를 사용한 검색
  - SELECT name FROM userdb WHERE name, LIKE '%현%'
  - SELECT name, id, FROM userdb WHERE id LIKE 'm%'

SELECT

OneKick

- 11. NULL값을 갖는 데이터검색
- SELECT name, id FROM userdb WHERE email IS NULL
- SELECT name, id FROM userdb WHERE email IS NOT NULL

INSERT

OneKick

- INSERT (새로운 데이터의 삽입)
- INSERT INTO 테이블명 [(필드명{[, 필드명]}*)] VALUES (필드값)
- INSERT INTO userdb (name, id, email, gender, milage, level) VALUES ('이완철', 'yunka1018','sspark@kornet.net'.'M','3500','B')
- INSERT INTO userdb VALUES('이완철', 'yunka1018', 'sspark@kornet.net', 'M','3500','B')
- INSERT INTO userdb (id, name) VALUES('yunka1018','정양미')

UPDATE

OneKick

- UPDATE 테이블명 SET 필드명=필드값 또는 산술식 {, 필드명 = 필드값 또는 산술식}*[WHERE 검색조건]
  - UPDATE userdb SET gender = 'F' WHERE name = '김현아'
  - UPDATE userdb SET gender = 'F'
- 만일 위와 같이 WHERE절로 검색조건을 명시하지 않은 경우에는 해당테이블내의 모든레코드의 필드값이 설정한값으로 수정되므로 유의해야한다.
  - UPDATE userdb SET gender = 'F'
  - UPDATE userdb SET milage = milage + 300

DELETE

OneKick

- DELETE (데이터의 삭제)
- DELETE FROM 테이블명 [WHERE 검색조건]
- DELETE FROM userdb WHERE id = 'king94'
- DELETE FROM userdb
- 만일 위와 같이 검색조건을 지정하지 않으면 해당테이블내의 모든레코드가 삭제되므로 주의해야한다.

명령어 정리

OneKick

MariaDB 명령어

OneKick

- **접속하기/끝내기**
  - $ mysql -u 유저명 –p
    - 접속하기
  - MariaDB[(none)] quit
    - 끝내기

- **Database**
  - MariaDB[(none)] show databases ;
    - DB목록보기
  - MariaDB[(none)] create database DB명 ;
    - DB생성
  - MariaDB[(none)] drop database DB명 ;
    - DB삭제
  - MariaDB[(none)] use DB명;
    - 사용할 DB선택하기

MariaDB 명령어

OneKick

- **Table 명령어**
  - MariaDB[DB명] show tables;
    - 테이블목록보기
  - MariaDB[DB명] create table 테이블명 (컬럼명1 데이터타입1, 컬럼명2 데이터타입2, ...);
    - 테이블생성
  - MariaDB[DB명] drop table 테이블명;
    - 테이블삭제

MariaDB 명령어

OneKick

- **Column 명령어**
  - MariaDB[DB명] show columns from 테이블명;
    - 컬럼목록보기1
  - MariaDB[*] show columns from 테이블명 from DB명;
    - 컬럼목록보기2
  - MariaDB[*] show columns from DB명.테이블명;
    - 컬럼목록보기3
  - MariaDB[DB명] show columns from 테이블명 where type like '타입명';
    - 컬럼목록보기 (특정타입)
    - ex) show columns from 테이블명 where type like 'varchar%';
  - MariaDB[DB명] alter table 테이블명 add 컬럼명 데이터타입 옵션;
    - 컬럼 추가
  - MariaDB[DB명] alter table 테이블명 drop 컬럼명;
    - 컬럼 삭제
  - MariaDB[DB명] alter table 테이블명 change 컬럼명 변경할컬럼명 varchar(12);
    - 컬럼명 변경/타입변경
  - MariaDB[DB명] alter table 테이블명 modify 컬럼명 변경할타입명;
    - 컬럼 타입변경 (ex. varchar(10)

MariaDB 명령어

OneKick

- **Count 명령어**
  - MariaDB[DB명] select count(*) from 테이블명;
    - 전체 레코드 수 (row count) 구하기
  - MariaDB[DB명] SELECT TABLE_NAME AS "Tables", round(((data_length + index_length) / 1024 / 1024), 2) "Size in MB" FROM information_schema.TABLES WHERE table_schema = "DB_NAME" ORDER BY (data_length + index_length) DESC;
    - DB사이즈 구하기
    - DB_NAME항목에 알고자 하는 DB명 입력

MariaDB 명령어

- 추가하기

  insert into 테이블이름 values(‘2017’, 1);

- **Etc**
  - MariaDB[DB명] show warnings;
    - 경고창 확인
  - MariaDB[DB명] flush privileges ;
    - 저장하기
  - MariaDB[DB명] repair table user user_frm;
    - 오류 시 실행
  - MariaDB[*] update user set password = password(' 암호 ') ;
    - 비밀번호 변경
  - MariaDB[*] show processlist ;
    - 현재 작업 중인 프로세스들의 진행정도 확인
  - MariaDB[*] kill " id번호 " ;
    - 현재 작업 중인 프로세스들 중 특정 프로세스를 종료
    - id번호는 show processlist에서 확인할 수 있음.