6장 4절

**데이터의 삽입 : INSERT** -> 테이블에 데이터를 삽입하는 명령어

예시 SQL:

USE sqldb;

CREATE TABLE testTbl1 (id int, userName char(3), age int);

INSERT INTO testTbl1 VALUES (1, '홍길동', 25);

**AUTO\_INCREMENT** : 빈 정보의 값을 자동으로 입력해주는 기능

예시 SQL

USE sqldb;

CREATE TABLE testTbl2

(id int AUTO\_INCREMENT PRIMARY KEY,

userName char(3),

age int );

INSERT INTO testTbl2 VALUES (NULL, '지민', 25);

INSERT INTO testTbl2 VALUES (NULL, '유나', 22);

INSERT INTO testTbl2 VALUES (NULL, '유경', 21);

SELECT \* FROM testTbl2;

만약 특정 값을 AUTO\_INCREMENT에 지정하고 싶다면 아래의 SQL 구문 이용

예시 SQL

ALTER TABLE testTbl3 AUTO\_INCREMENT=1000; -- 초기 값을 1000으로 지정

SET @@auto\_increment\_increment=3; -- 이 3은 증가값을 변경해주는 구문임

**대량의 샘플 데이터 생성** -> SELECT문의 결과를 INSERT!

* 대회 대용량 데이터를 처리하는데 편리할 듯

예시 SQL

USE sqldb;

CREATE TABLE testTbl4 (id int, Fname varchar(50), Lname varchar(50));

INSERT INTO testTbl4

SELECT emp\_no, first\_name, last\_name -- 원하는 조건의 SELECT 문 생성

FROM employees.employees ;

**데이터의 수정 : UPDATE** -> 일반적으로 WHERE절을 이용 특정 조건만 UPDATE시킴

형식

UPDATE 테이블이름

SET 열1=값1, 열2=값2

WHERE 조건;

**데이터의 삭제 : DELETE FROM** -> UPDATE와 마찬가지로 특정 WHERE 조건을 걸어줌

DELETE – 트랜잭션 로그를 기록, 시간이 오래걸림

DROP – 테이블 자체를 삭제, 트랜잭션을 발생시키지 않음,

TURNCATE - 트랜잭션 로그를 발생X, 테이블은 남겨 놓음 DROPP에 비해 시간이 빠르다.

실습3 : 대용량 테이블을 삭제하자

참고 : Edit – Preferences – SQL Editor 에서 safe 모드 해제가능

실습 4 : 조건부 데이터 입력 변경

**WITH절과 CTE(Common Table Expression)**

형식

WITH CTE\_테이블이름(열 이름)

AS

(

<쿼리문>

)

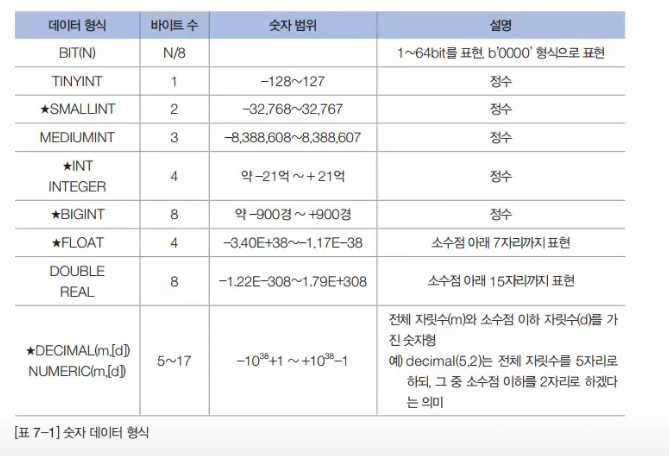
SELECT 열 이름 FROM CTE\_테이블이름 ;

더 자세한 내용은 6장-CTE 스크립트를 참고하기!

**7장 SQL 고급**

7.1 MYSQL의 데이터 형식 (\*)는 중요하다는 의미

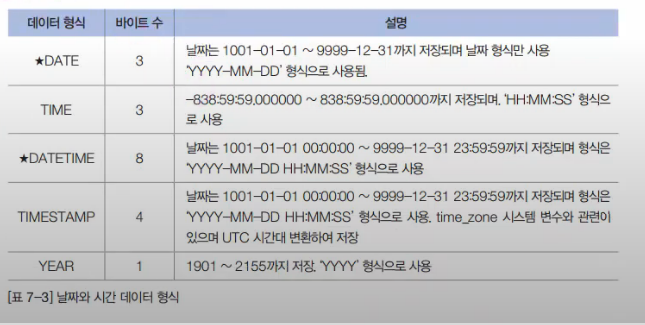
**숫자 데이터 형식**



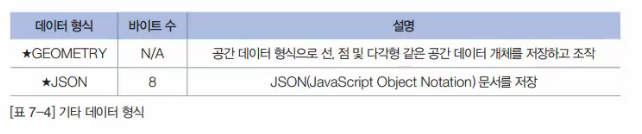
**문자 데이터 형식**



**날짜 데이터 형식**



**기타 데이터 형식**



GEOMETRY 형식은 14장에서 다룸! -> 우리 데이터 다룰시 이용가능하다고 생각됨

**참고**

LONGTEXT, LONGBLOB는 최대 4GB까지 저장 가능하므로 영화 대본, 영화 동영상 같은 파일을 쉽게 저장 가능하다는 장점이 있다.

**변수의 사용**

형식

SET @변수이름 = 변수의 값; -- 변수의 선언 및 값 대입

SELECT @변수이름 ; -- 변수의 값 출력

더 자세한 내용은 7장-실습1 파일 참고