

따라 하면서 배우는 IT

SQL

# 목차

## INDEX

SQL

DDL, DML, DCL

따라  
學IT

SQL이란?

DDL  
DML  
DCL

DDL 사용해보기  
DML 사용해보기

따라 하면서 배우는 IT

SQL

# SQL

## SQL이란?

//

Structured Query Language  
SQL

SQL은 1970년대 초 IBM에서 관계형 데이터베이스 관리 시스템에서 저장된 데이터를 다루기 위해서 개발되었다.

관계 대수와 관계해석을 기초로한 고급 데이터 언어  
데이터 정의, 데이터 조작, 데이터 제어 기능을 제공하는 비절차적 언어

SQL의 종류는 DDL, DML, DCL이 있다.

//

```
CREATE TABLE IF NOT EXISTS `mydb`.`member` (  
  `mid` VARCHAR(45) NOT NULL,  
  `mpw` VARCHAR(45) NULL,  
  `mname` VARCHAR(45) NULL,  
  PRIMARY KEY (`mid`))  
ENGINE = InnoDB;
```

따라 하면서 배우는 IT

**DDL, DML, DCL**

# DDL, DML, DCL

## DDL

//

데이터 정의어  
DDL

//

스키마, 도메인, 테이블, 뷰, 인덱스 등 데이터의 구조를 정의하거나 제거

CREATE, ALTER, DROP이 있다.

```
CREATE TABLE IF NOT EXISTS `mydb`.`member` (  
  `mid` VARCHAR(45) NOT NULL,  
  `mpw` VARCHAR(45) NULL,  
  `mname` VARCHAR(45) NULL,  
  PRIMARY KEY (`mid`))  
ENGINE = InnoDB;
```

# DDL, DML, DCL

## DDL

---

//

데이터 정의어  
DDL

//

---

스키마, 도메인, 테이블, 뷰, 인덱스 등 데이터의 구조를 정의하거나 제거

```
CREATE TABLE [테이블_이름] (  
    [속성_이름]      [타입],  
    [속성_이름]      [타입],  
    [속성_이름]      [타입],  
    [속성_이름]      [타입],  
    ...  
    [옵션]  
);
```

# DDL, DML, DCL

## DDL

---

//

데이터 정의어  
DDL

스키마, 도메인, 테이블, 뷰, 인덱스 등 데이터의 구조를 정의하거나 제거

ALTER TABLE [테이블\_이름] ADD [속성\_이름] [타입]  
ALTER TABLE student ADD hpnum INTEGER;

ALTER TABLE [테이블\_이름] DROP (속성\_이름) [CASCADE]  
ALTER TABLE student DROP (hpnum);

//

---

ALTER TABLE [테이블\_이름] RENAME COLUMN [속성\_이름] to [변경\_이름]  
ALTER TABLE student RENAME COLUMN name TO full\_name;

ALTER TABLE [테이블\_이름] MODIFY (속성\_이름 타입)  
ALTER TABLE student MODIFY full\_name VARCHAR2(20);



# DDL, DML, DCL

## DDL

---

//

데이터 정의어  
DDL

스키마, 도메인, 테이블, 뷰, 인덱스 등 데이터의 구조를 정의하거나 제거

DROP TABLE 테이블\_이름 [CASCADE or RESTRICTED]

\*CASCADE : 개체를 변경/삭제할 때 다른 개체가 변경/삭제할 개체를 참조하고 있을 경우 함께 변경/삭제

---

//

RESTRICTED : 개체를 변경/삭제할 때 다른 개체가 변경/삭제할 개체를 참조하고 있을 경우 변경/삭제가 취소(제한)

# DDL, DML, DCL

## DML

---

//

데이터 조작어  
DML

//

---

실직적으로 데이터를 조회, 삽입, 삭제, 수정할 때 사용하는 SQL

SELECT [속성명] FROM [테이블명] WHERE [조건];

SELECT full\_name FROM student WHERE age >= 27;

SELECT full\_name FROM student WHERE age >= 27 AND full\_name='test';

SELECT full\_name FROM student WHERE age >= 27 OR full\_name='test';

# DDL, DML, DCL

## DML

---

//

데이터 조작어  
DML

실직적으로 데이터를 조회, 삽입, 삭제, 수정할 때 사용하는 SQL

INSERT INTO 테이블명 (속성명, 속성명...) VALUES (값, 값...)

INSERT INTO 테이블명 VALUES (값, 값...)

//

---

# DDL, DML, DCL

## DML

---

//

데이터 조작어  
DML

실직적으로 데이터를 조회, 삽입, 삭제, 수정할 때 사용하는 SQL

DELETE FROM [테이블명] WHERE [조건]

DELETE FROM student WHERE full\_name='SIMSIM';

//

---

# DDL, DML, DCL

## DML

---

//

데이터 조작어  
DML

//

---

실직적으로 데이터를 조회, 삽입, 삭제, 수정할 때 사용하는 SQL

UPDATE 테이블명 SET 속성\_이름=값, 속성\_이름=값 WHERE 조건

UPDATE student SET bundan=1 WHERE full\_name='LEE JINWOO';

UPDATE student SET jul=1 WHERE full\_name='LEE JINWOO';

UPDATE student SET bundan=1,jul=1 WHERE full\_name='LEE JINWOO';

# DDL, DML, DCL

## DCL

---

//

데이터 제어어  
DCL

//

---

데이터베이스에 접근하고 사용할 수 있도록 권한을 주고 회수하는 명령어

GRANT, REVOKE가 있다.

# DDL, DML, DCL

## DCL

---

//

데이터 제어어  
DCL

데이터베이스에 접근하고 사용할 수 있도록 권한을 주고 회수하는 명령어

GRANT [권한명] on [DB].[TABLE] to [계정명];

GRANT ALL PRIVILEGES ON \*.\* TO 'user01'@'%' WITH GRANT OPTION;

//

---

# DDL, DML, DCL

## DCL

---

//

데이터 제어어  
DCL

데이터베이스에 접근하고 사용할 수 있도록 권한을 주고 회수하는 명령어

REVOKE [권한명] on [DB].[TABLE] FROM [계정명];

//

---



# DDL, DML, DCL

## DCL

---

//

데이터 제어어  
DCL

//

---

권한의 종류

CREATE, ALTER, DROP : 테이블 생성, 변경, 삭제

SELECT, INSERT, UPDATE, DELETE : 테이블의 레코드 조회, 입력, 수정, 삭제

ALL : 모든 권한 허용

USAGE : 권한 없이 계정만 생성

따라 하면서 배우는 IT

실습

## 1. SQL에서 DDL 사용해보기

SQL을 이용해서 데이터 구조 만들어보기 DB 설계해보기

## 2. SQL에서 DML 사용해보기

SQL을 이용해서 데이터 삽입, 조회, 수정, 삭제 해보기