

# DSLab SQL세션

2019 02 28

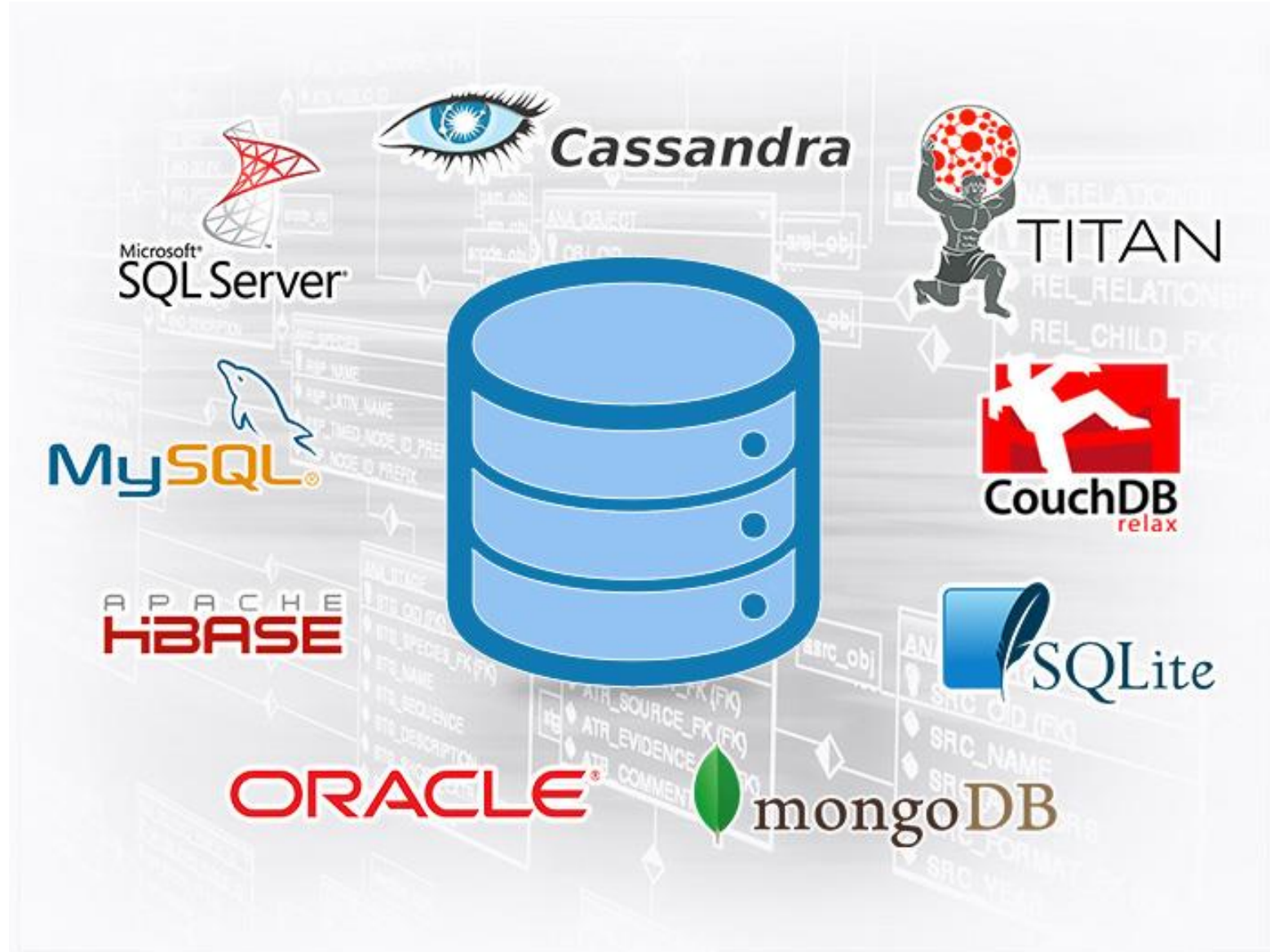
# 목차

1. SQL 이란
2. SQL의 구조
3. SQL의 데이터 타입
4. 정의어 조작용 제어어
5. SQL 실습
6. 파이썬 실습

# SQL이란

- SQL (구조화 질의어, S-Q-L)는 관계형 데이터베이스 관리 시스템(RDBMS)의 데이터를 관리하기 위해 설계된 특수 목적의 프로그래밍 언어이다. 관계형 데이터베이스 관리 시스템에서 자료의 검색과 관리, 데이터베이스 스키마 생성과 수정, 데이터베이스 객체 접근 조정 관리를 위해 고안되었다. 많은 수의 데이터베이스 관련 프로그램들이 SQL을 표준으로 채택하고 있다.

-위키피디아



# SQL의 구조

- 개체(Entity)
  - 데이터 표현 대상에 대한 논리적 표현 (즉, 구별이 가능한 것을 말함)
- 관계(Relation) [수학적 용어] = 테이블(Table) [데이터베이스 용어]
  - 튜플들의 집합
  - 기본 테이블 : 명시적으로 사용되며, 저장장치에 생성 기록되는 테이블
  - 임시 테이블 : 임시 목적으로 메모리에 일시 저장 (데이터베이스 재기동시 사라짐)
  - 가상 테이블 : 뷰(View)를 말함 (테이블 처럼 사용되지만 실제 데이터를 갖지 않음)

- 튜플(Tuple) = 행(Row)
  - 값의 나열(리스트) 즉, 테이블의 행 (때로는, 레코드 라고도 함)
- 속성(Attribute) = 열(Column)
  - 관계 테이블의 열(column)
- 도메인(Domain)
  - 하나의 속성이 취할 수 있는 허가된 값의 집합 (그 속성의 가능한 타입)
- 키(Key)
  - 릴레이션 내에서 튜플을 유일하게 구별할 수 있는 속성 모음
- 널값(Null)
  - 존재하지 않는 값을 의미하는 특별한 값

- 뷰(View)
  - 1 이상의 테이블로부터 유도되는 가상 테이블
  - 테이블 처럼 동작하지만 실제로는 데이터를 갖지 않는 테이블
  - 뷰가 참조하는 테이블 변경이 일어나면 뷰도 같이 변경됨
  - 외부 스키마에서 기본 테이블 및 뷰가 정의됨
  - 일단 정의된 뷰는 변경이 불가능하고 삭제 후 다시 생성해야만 됨
  - 뷰를 통한 검색은 가능하지만, 삽입,갱신,삭제 등은 제약을 받음
  - 사용형식) create view 뷰명 as select문;
- 질의 (Query)
  - 사용자의 정보 요구 (문장 또는 문자열 또는 프로그램 등)

- 기본키(primary key) : DB 설계자에 의해 직접 선택된 하나의 후보키
  - 개체를 고유하게 식별할 수 있는 속성
  - 중복을 허용 않고, Null 값도 허용 않음
- 기본키와 관련된 키
  - 대체키(alternate key) : 2 이상의 후보키 중 기본키를 제외한 나머지 후보키들
  - 외래키(foreign key) : 다른 릴레이션(테이블)의 기본키를 갖게되는 키
  - 즉, 다른 테이블의 기본키 값과 대응되는 키



# SQL의 데이터 타입

## MySQL DATA TYPES

DATE TYPE	SPEC	DATA TYPE	SPEC
CHAR	String (0 - 255)	INT	Integer (-2147483648 to 2147483647)
VARCHAR	String (0 - 255)	BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
TINYTEXT	String (0 - 255)	FLOAT	Decimal (precise to 23 digits)
TEXT	String (0 - 65535)	DOUBLE	Decimal (24 to 53 digits)
BLOB	String (0 - 65535)	DECIMAL	"DOUBLE" stored as string
MEDIUMTEXT	String (0 - 16777215)	DATE	YYYY-MM-DD
MEDIUMBLOB	String (0 - 16777215)	DATETIME	YYYY-MM-DD HH:MM:SS
LONGTEXT	String (0 - 4294967295)	TIMESTAMP	YYYYMMDDHHMMSS
LOBLOB	String (0 - 4294967295)	TIME	HH:MM:SS
TINYINT	Integer (-128 to 127)	ENUM	One of preset options
SMALLINT	Integer (-32768 to 32767)	SET	Selection of preset options
MEDIUMINT	Integer (-8388608 to 8388607)	BOOLEAN	TINYINT(1)

# 데이터 정의어 (DDL)

- DDL(Data Definition Language) – 객체의 생성, 변경, 삭제 명령어 (CREATE , ALTER , DROP, RENAME 등)
  - SCHEMA, DOMAIN, TABLE, VIEW, INDEX를 정의하거나 변경 또는 삭제할 때 사용하는 언어
  - 데이터베이스 관리자나 데이터베이스 설계자가 사용함

# 데이터 조작용어 (DML)

- DML(Data Manipulation Language)- 레코드 제어 명령어 (SELECT, INSERT, UPDATE ,DELETE )
- 데이터베이스 사용자가 응용 프로그램이나 질의어를 통하여 저장된 데이터를 실질적으로 처리하는데 사용하는 언어
  - 데이터베이스 사용자와 데이터베이스 관리 시스템 간의 인터페이스 제공

# 데이터 제어어 (DCL)

- DCL(Data Control Language) – 객체 권한 부여 등의 제어어 (GRANT 등)
- 데이터의 보안, 무결성, 데이터 회복, 병행 수행 제어 등을 정의하는 데 사용하는 언어
  - 데이터베이스 관리자가 데이터 관리를 목적으로 사용함

# CREATE

- CREATE DATABASE [DB\_NAME];
- USE [DB\_NAME];
- DROP DATABASE [DB\_NAME];
  
- CREATE TABLE [TABLE\_NAME] ( 컬럼 상세 );
- DROP TABLE [TABLE\_NAME];
  
- DELETE TABLE [TABLE\_NAME];
- TRUNCATE TABLE [TABLE\_NAME];

# CREATE

```
CREATE TABLE DEPT (  
    DEPTNO DECIMAL(2),  
    DNAME VARCHAR(14),  
    LOC VARCHAR(13),  
    CONSTRAINT PK_DEPT PRIMARY KEY (DEPTNO)  
);
```

# SQL 문법

- SELECT 칼럼명1, 칼럼명2
- [FROM 테이블명 ]
- [WHERE 조회조건 ]
- [GROUP BY 칼럼명]
- [HAVING 조건]
- [ORDER BY 칼럼명 [ASC | DESC]]
- [LIMIT offset, 조회 할 행의 수]

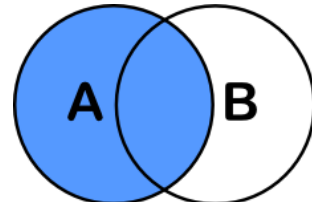
# 집계함수

- 집계함수는 계산을 수행하고 단일 값을 반환 합니다.
- Count()함수를 제외한 집계함수는 Null값을 무시하며 SELECT문 혹은 HAVING 절에만 사용할 수 있습니다.

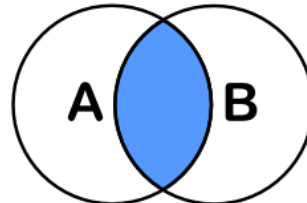
APPROX_COUNT_DISTINCT	MIN
AVG	STDEV
CHECKSUM_AGG	STDEVP
COUNT	STRING_AGG
COUNT_BIG	SUM
GROUPING	VAR
GROUPING_ID	VARP
MAX	



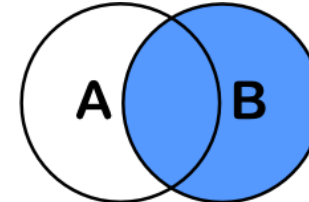
# JOIN



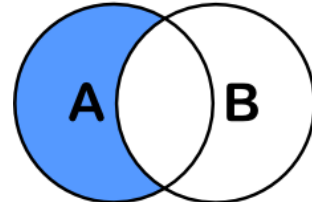
```
SELECT <auswahl>
FROM tabelleA A
LEFT JOIN tabelleB B
ON A.key = B.key
```



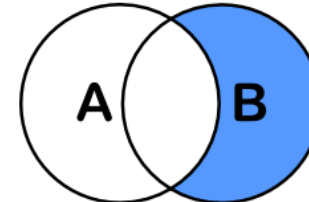
```
SELECT <auswahl>
FROM tabelleA A
INNER JOIN tabelleB B
ON A.key = B.key
```



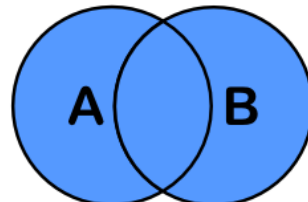
```
SELECT <auswahl>
FROM tabelleA A
RIGHT JOIN tabelleB B
ON A.key = B.key
```



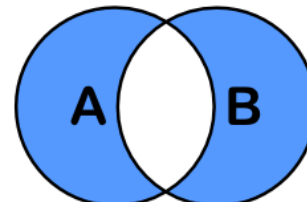
```
SELECT <auswahl>
FROM tabelleA A
LEFT JOIN tabelleB B
ON A.key = B.key
WHERE B.key IS NULL
```



```
SELECT <auswahl>
FROM tabelleA A
RIGHT JOIN tabelleB B
ON A.key = B.key
WHERE A.key IS NULL
```



```
SELECT <auswahl>
FROM tabelleA A
FULL OUTER JOIN tabelleB B
ON A.key = B.key
```



```
SELECT <auswahl>
FROM tabelleA A
FULL OUTER JOIN tabelleB B
ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL
```

# JOIN

- ORACLE 조인

```
SELECT E.ENAME,D.DNAME  
FROM EMP E,DEPT D  
WHERE E.DEPTNO = D.DEPTNO;
```

- ANSI 조인

```
SELECT E.ENAME, M.ENAME  
FROM EMP E INNER JOIN EMP M ON E.MGR = M.EMPNO;
```

# JOIN의 방식

- Nested Loop Join
- Merge Join
- Hash Join

실습