

Data/Database

data: 관찰의 결과로 나타난 정량적 혹은 정성적 실제 값

database: 대량의 정보를 컴퓨터가 효율적으로 접근할 수 있도록 가공/저장한 것. 한 조직에 필요한 정보를 여러 응용 시스템에서 공유할 수 있도록 논리적으로 연관된 데이터를 모으고 중복을 최소화하여 구조적으로 통합/저장해둔 것.

database 특징 4가지

1. **실시간 접근성** - 사용자가 데이터를 요청하면 실시간으로 결과를 서비스함
2. **계속적인 변화** - 데이터값은 시간에 따라 늘 바뀜
3. **동시공유** - db는 서로 다른 업무, 또는 여러 사용자에게 동시공유됨
4. **내용에 따른 참조** - db에 저장된 데이터는 db의 물리적 위치가 아니라 db값에 따라 참조.

DB/ DBMS의 특징 / 관계형 데이터베이스

DB

- 일반적으로 컴퓨터 시스템에 전자적으로 저장되는 구조화된 정보/데이터의 조직화된 모음

DBMS란?

- 다수의 사용자들이 데이터베이스 내의 데이터를 접근할 수 있도록 해주는 소프트웨어 도구의 집합.
- CRUD(Create, Read, Update, Delete)의 기능을 가짐

RDB(Relational Database)

- 관계형 데이터베이스는 키, 값(키, 벨류)의 간단한 관계를 2차원 테이블형식으로 나타낸 데이터베이스. 하나의 데이터베이스 안엔 여러개의 테이블이 존재할 수 있음.

테이블

테이블

- 테이블은 행(ROW)과 열(COLUMN)로 이뤄져있다. 행은 가로, 열은 세로다.

column

name	age	major
Sophia	21	Economics
Isabella	22	Engineering
Jacob	22	Biology
Mason	23	Psychology

row
(record)

ROW(행)은 하나의 테이블에선 데이터(tuple, record)를 수백, 수천, 수만, 수백만개 저장하고 있음.

COLUMN(열)은 데이터의 많은 속성을 지닌다. 속성에 각자 구분하기 쉽게 이름을 붙여 분류한다. (아이디, 사원명, 핸드폰번호, 이메일 등등...) 각 칼럼은 특정한 데이터 타입을 가진다(number, varchar2)... 이런 데이터 타입이나 comment, 컬럼 이름들은 스키마에 테이블의 큰 정보를 저장한다.

RDBMS

- RDBMS는 Relational Database Management System의 약자. 관계형 db 관리 시스템이란 뜻.
- 정보를 저장할때 테이블이란 형태로 저장하는데, 이 테이블은 정보로 각각 구성된다. 이런 테이블 사이의 연관성을 관계성이라 함.
- RDBMS에선 키를 통해 연관된 정보를 조회하는데 특화되어있다.

RDBMS의 5가지 특징

1. 2차원 구조모델을 기반으로 수평적 구조.
2. 데이터 무결성, 완전성, 정확성 보장
3. SQL을 사용해 데이터 접근, CRUD 조작 허용
4. 안드로이드, ios, solaris, linux, window, unix 등 여러 os 플랫폼에서 사용 가능
5. rdbms는 사용자가 테이블을 분할하고 데이터 복제 가능.

SQL

SQL(structured query language) - 구조화된 질의 언어

query라는 글자가 있듯 말그대로 질의를 위한 언어. 관계형 데이터베이스에서 데이터를 조회하거나 조작하기 위해서 사용되는 표준 검색언어

SQL 종류 4가지

- **DDL(Data Definition Language)** - 데이터 정의어(ex: create, alter, drop, truncate)
- **DML(Data Manipulation Language)** - 데이터 조작어(ex: insert, update, delete, select)
- **DCL(Data Control Language)** - 데이터 제어어(ex: grant, revoke)
- **TCL(Transaction control language)** - 트랜잭션 제어어(ex: commit, rollback, savepoint)

데이터 타입

- null 여부 : **null**은 선택, **not null**은 필수라는 뜻

1. 문자형(가변형/고정형)

- 고정형: **char(2000byte)**
- 가변형: **varchar2(4000byte)** - 사용자가 정의한 데이터 길이만큼 기억공간 사용, 오라클에서만 사용되는 데이터 타입
- 가변길이

long: 가변길이 문자열을 저장할 때 사용

- 2gb크기의 자료저장 가능
- 한 테이블에 하나의 **long** 타입으로 선언 가능
- **select**문의 **select** 절이나 **update**의 **set**절, **insert**문의 **values**절에서 사용 가능

clob

- 가변길이 문자열 저장
- 최대 **4gb** 문자열 저장 가능
- **long** 타입을 개선한 데이터 타입
- 한 테이블에서 여러 개 칼럼에서 **clob** 사용 가능

nvarchar2, nclob

- 다국적 언어 지원형식
- **utf-8, utf-16** 형식의 코드로 변환해서 저장
- 나머지 기능은 **varchar2, clob**랑 동일

char, varchar2 차이

char형과 varchar2

	char(10)	varchar2(10)
korea	10byte로 저장	5byte로 저장
한국 xe버전 : 한글자당 3byte로 저장 ee버전 : 한글자당 2byte로 저장	10byte로 저장	6byte로 저장
대한민국	실제 데이터가 12byte라 입력 불가	실제 데이터가 12byte라 입력 불가

2. 숫자형

`number([p], [s])`

- 정수와 실수 모두 표현 가능.
- **p**: 표현할 수 있는 전체 자리수
- **s**: 소수점 이하 자리수

ex) 실제데이터 1234.567을 저장 시도

<code>number</code>	1234.567
<code>number(7,3)</code>	1234.567
<code>number(7)</code> or <code>number(7,0)</code>	1235
<code>number(7,1)</code>	1234.6
<code>number(7,-2)</code>	1200

0은 생략이 가능하다. -2면 반대로 10의자리 2자리임을 의미한다.

3. 날짜형

1. **date**:년월일 시분초('yyyy/MM/dd hh24:mi/ss')를 저장

날짜형 연산지원

- **date** + 숫자 : **date** + 숫자일 후의 날짜정보 리턴
- **date** - 숫자 : **date** - 숫자일 후의 날짜정보 리턴
- **date** - **date** : 날짜 차이를 리턴.

```
select
```

```
    to_char(sysdate+1,'yy/mm/dd hh24:mi:ss')tomorrow,  
    to_char(sysdate,'yy/mm/dd hh24:mi:ss')today,  
    to_char(sysdate-1,'yy/mm/dd hh24:mi:ss')yesterday,  
    to_char(sysdate+(25/24),'yy/mm/dd hh24:mi:ss') "25시간뒤",  
    (sysdate+1)-sysdate
```

```
from
```

```
    dual;
```

스크립트 출력 x 실행 결과 x

SQL | 인출된 모든 행: 1(0.003초)

TOMORROW	TODAY	YESTERDAY	25시간뒤	(SYSDATE+1)-SYSDATE
1 22/04/14 09:29:45	22/04/13 09:29:45	22/04/12 09:29:45	22/04/14 10:29:45	1

commit, rollback

commit

- 테이블을 생성하고 데이터를 삽입해도 그건 끝까지 저장된 게 아니다. db에 전달이 되지 않았기 때문. 따라서 **commit**을 해야 db에 적용되며, **commit**을 하면 이제 지워지지 않는다.(**rollback**을 해도 마찬가지)

rollback

- 메모리상 작업내용을 취소하는 방법. 저장 전으로 돌아간다.