

ILE3-032 DBMS

데이터를 관리하기 위하여 사용되는 DBMS에 대하여 학습

Database Management System [↗](#)

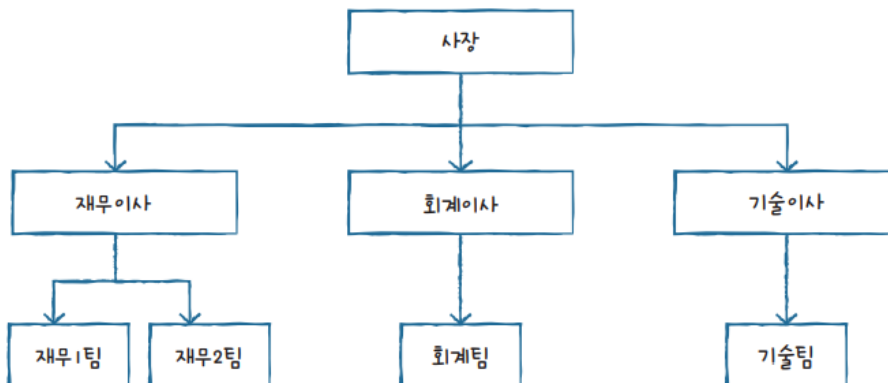
데이터베이스라는 데이터의 집합을 만들고, 저장 및 관리할 수 있는 기능들을 제공하는 응용 프로그램

기능 [↗](#)

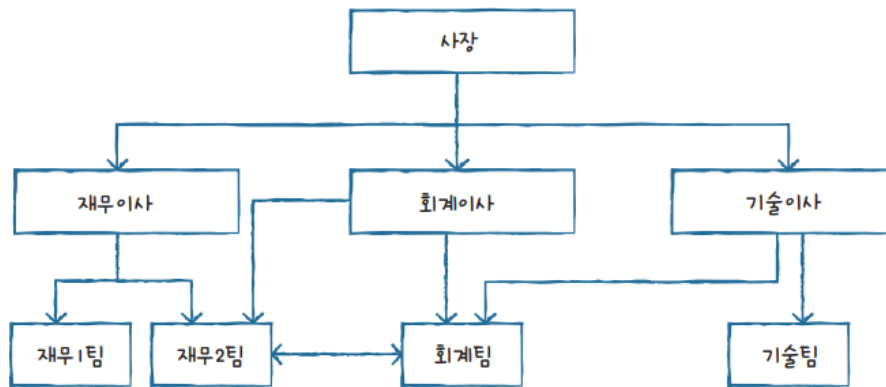
- 중복 제어
중복되는 데이터가 저장되지 않게 한다
중복되는 데이터가 존재하는 경우 저장공간이 낭비되고 데이터의 일관성이 손실 될 수 있음
- 접근 통제
각 사용자에게 권한을 부여하여, 권한에 따라 데이터에 대한 접근을 제어
- 인터페이스 제공
데이터를 조작하기 위한 인터페이스를 제공
- 무결성
데이터의 일관성을 유지하기 위하여 제약 조건을 정의/검사하는 기능을 제공

분류 [↗](#)

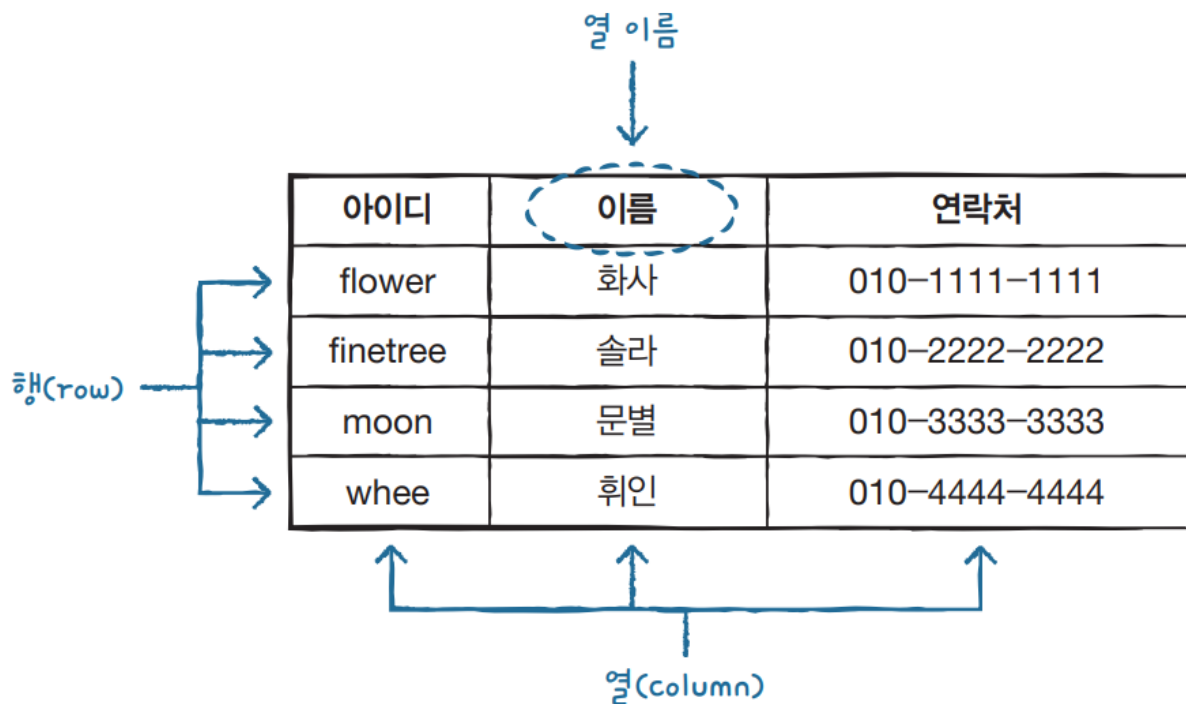
- 계층형
최초로 등장한 DBMS 개념
트리 형태로 데이터를 구성
최초 구성 후 변경이 까다로움
현재는 사용하지 않는 형태



- 망형
계층형 DBMS의 문제점을 개선하기 위해 등장
다른 노드에 존재하는 하위 노드에도 연결이 가능
구조가 복잡하여 현재는 거의 사용되지 않음



- 관계형
현재 거의 대다수의 DBMS가 사용하는 형태
행/열로 이루어진 2차원 형태 테이블을 이용하여 구성



- 객체지향형
확장성이 부족한 관계형 데이터 베이스를 개선하기 위하여 사용자가 데이터 타입을 정의하여 사용하는 형태

Object-Oriented Model

Object 1: Maintenance Report Object 1 Instance

Date	
Activity Code	
Route No.	
Daily Production	
Equipment Hours	
Labor Hours	

01-12-01
24
I-95
2.5
6.0
6.0

Object 2: Maintenance Activity

Activity Code	
Activity Name	
Production Unit	
Average Daily Production Rate	

- 객체 관계형
관계형에 객체지향형의 특징을 추가한 DBMS
객체지향형과 유사하게 사용자 정의 타입을 지원

SQL (Structured Query Language)

관계형 데이터베이스에서 사용하는 언어

국제 표준화 기구(ISO)에서 표준을 정해서 발표하고 있으나, DBMS별로 다른 구문이 존재(SQL 방언)

SQL 방언의 예

특정 개수의 Row만 읽어오고 싶은 경우

- ORACLE

```
SELECT * FROM table WHERE ROWNUM <= 10
```

- MySQL

```
SELECT * FROM table LIMIT 10
```

수행하는 작업에 따라 DML, DDL, DCL, TCL로 분류

- **DDL** (데이터 정의어, Data Definition Language)
Data의 Schema를 정의 및 편집하는데 사용되는 언어
CREATE, ALTER, DROP이 포함
- **DML** (데이터 조작어, Data Manipulation Language)
데이터를 읽고 쓰는데 사용되는 언어
SELECT, INSERT, UPDATE, DELETE가 포함
- **DCL** (데이터 제어어, Data Control Language)
접근을 위한 권한을 관리하는 언어
GRANT, REVOKE가 포함
- **TCL** (트랜잭션 제어어, Transaction Control Language)
논리적인 작업(DML의 묶음) 단위를 조작하는 언어
BEGIN, COMMIT, ROLLBACK이 포함

DDL

- CREATE

Schema와 연관된 객체를 생성할 때 사용.

```
CREATE DATABASE [DB_NAME]
```

- CREATE TABLE

Table을 생성할 때 사용하는 구문

Column을 1개 이상 정의가 필수

```
1 CREATE TABLE Persons(  
2     PersonID int,  
3     LastName varchar(255),  
4     FirstName varchar(255),  
5     Address varchar(255),  
6     City varchar(255)  
7 );
```

- DROP

Schema와 연관된 객체를 제거할 때 사용

```
DROP DATABASE [DB_NAME]
```

```
DROP TABLE [TABLE_NAME]
```

- ALTER

Schema와 연관된 객체를 편집 할 때 사용

```
1 -- Persons Table에 Age Column 추가  
2 ALTER TABLE Persons ADD Age int;
```

DML

- SELECT

데이터를 읽어올 때 사용

```
SELECT * FROM [TABLE_NAME] WHERE [COND]
```

- INSERT

데이터를 삽입 할 때 사용

```
INSERT [TABLE_NAME] VALUES([VAL1], [VAL2], ... )
```

- UPDATE

데이터를 변경할 때 사용

```
UPDATE [TABLE_NAME] [COLUMN_NAME]=[VAL] WHERE [COND]
```

- DELETE

데이터를 삭제할 때 사용

```
DELETE FROM [TABLE_NAME] WHERE [COND]
```

과제

POI 정보를 저장 할 Table을 만들려고 한다.

아래의 정보를 저장할 수 있는 CREATE TABLE 구문을 작성

- POI ID
- 이름
- 위치 (경위도)
- 종별
- 전화번호
- 주소