DB: 데이터의 집합. 엑셀 파일도 db임

DBMS: 데이터를 효과적으로 관리하기 위한 시스템

관계형 DB 구성요소

1. 계정: 데이터의 접근 제한을 위한 업무/시스템별 계정
2. 테이블
3. 스키마: 테이블이 어떤 구성으로 되있는지, 어떤 정보를 가지고 있는지에 대한 구조 정의

### 테이블

데이터를 저장하는 최소 단위

물리 모델에서 속성은 컬럼이라고 함

하나의 테이블은 반드시 하나의 유저가 소유해야 함

테이블명은 중복될 수 없지만 소유자가 다른 경우 다른 이름으로 생성 가능

특징

분류, 정렬, 탐색 빠름

신뢰성 높고 데이터의 무결성 보장

작성된 스키마 수정 어려움

DB의 부하 분석 어려움

### 데이터 무결성

정확성, 일관성을 유지. 결손 및 부정합이 없음을 보증하는 것

1. 개체 무결성: 테이블의 기본키를 구성하는 속성은 NULL, 중복 X
2. 참조 무결성: 외래키의 값은 널이거나 참조 테이블의 기본키 값과 동일해야 함
3. 도메인 무결성: 주어진 속성이 도메인에 속한 값이어야 함
4. NULL 무결성: 특정 속성에 대해 null 허용하지 않는 특징
5. 고유 무결성: 특정 속성에 대해 값이 중복되지 않는 특징
6. 키 무결성: 하나의 관계에 최소 하나의 키가 존재해야 함

SQL 종류

1. DDL(Data Definition Language): CREATE, ALTER, DROP, TRUNCATE
2. DML(Data Manipulation L): INSERT, DELETE, UPDATE, MERGE
3. DCL(Data Control L): GRANT, REVOKE
4. TCL(Transaction Control L: COMMIT, ROLLBACK
5. DQL(Data Query L): SELECT

### SELECT

6개의 절로 구성

각 절의 순서대로 작성해야 함

SELECT FROM WHERE GROUP BY HAVING ORDER BY

GROUP BY, HAVING이 바뀔 수는 있지만 잘 사용 안함

FROM, WHERE, GROUP BY, HAVING, SELECT, ORDER BY 순으로 실행

가져오고, 필터링, 그룹, 그룹 단위 필터링, 속성/연산 선택, 정렬

SELECT보다 늦게 수행되는 ORDER BY에서만 별칭 사용 가능

한글 사용 가능, 존재하는 예약어는 별칭 사용 불가

별칭에 공백, 특문, 별칭 그대로 전달할 경우 큰따음표 사용

### FROM

테이블명 혹은 뷰명

테이블 여러개 전달 가능 -> 조인 조건 없이 테이블만 나열 시 카티시안 곱 발생

테이블 별칭 선언 가능, AS 안씀

ORACLE에서는 FROM 생략 불가. 23c 버전부터 생략 가능

MSSQL은 생략 가능

### 함수

FROM 절 제외한 모든 곳에서 사용 가능

기본적인 쿼리문을 강력하게

계산, 데이터 수정, 타입 변경

단일행, 복수행 함수로 나뉨(avg, count 등)

DECODE(대상, 값1, 반환1, 값2, 반환2 … 이외반환)

대상의 값에 따라 == 값1 -> 반환1, == 값2 -> 반환2 … 마지막 반환

NVL(대상, 값)

대상이 NULL이면 값 반환

NVL2(대상, 값1, 값2)

대상이 NULL이면 2, 아니면 1 반환

COALESCE(대상1, 대상2…이외반환)

대상1이 NULL이면 대상2, 2가 NULL이면 3… 쭉 가다가 모두 NULL이면 이외반환

### WHERE

데이터 필터링

NULL 조회 시 IS NULL, IS NOT NULL 연산자 사용해야 함

### GROUP BY

특정 조건에 따라 그룹으로 분리하는 식

그룹에 대한 조건은 WHERE에서 불가능. HAVING에서 해야 함

오라클은 GROUP BY로 데이터가 요약되므로 요약되기 전 데이터와 함께 출력할 수 없음

SELECT NUM, SUM(PRICE), NAME GROUP BY NUM: 집계 함수가 있는 상태에, NAME이 집계연산이 아니므로 불가능

### 

### HAVING

그룹 단위로 필터링할 때 사용

SELECT보다 먼저라서 SELECT의 별칭 사용 불가

### ORDER BY

출력 정렬

### JOIN

여러 테이블의 데이터를 사용해 동시 출력, 참조할 경우 사용

FROM절에 테이블 나열

ORACLE은 테이블 나열 순서 중요하지 않음. ANSI 표준은 OUTER JOIN시 순서 중요

ORACLE은 조인은 WHERE에서 기술, ANSI 표준은 조인조건은 ON, 일반 조건만 WHERE

종류

조건의 형태에 따라

EQUI JOIN: JOIN 조건이 ==일 경우

NON EQUI JOIN: 조건이 ==가 아닌 경우(>, <, between 등)

조인 결과에 따라

INNER JOIN: JOIN 조건 성립하는 데이터만 출력

OUTER JOIN: 성립하지 않는 데이터도 출력. LEFT, RIGHT, FULL OUTER JOIN 등

NATURAL JOIN: 조건 생략 시 같은 이름의 속성으로 자연 연결되는 조인

CROSS JOIN: 조건 생략 시 두 테이블의 발생 가능한 모든 행 출력

SELF JOIN: 하나의 테이블을 두번 이상 참조해 연결

표준조인

ANSI 표준으로 작성되는 INNER JOIN, CROSS JOIN, NATURAL JOIN, OUTER JOIN

INNER JOIN

조인 조건이 일치하는 행만 추출. ORCALE 조인 기본

ANSI의 경우 FROM 절에 INNER JOIN이나 JOIN을 명시, USING이나 ON 조건절을 필수로 사용

USING: 조인할 컬럼명이 같을 경우 사용, 괄호 필수. SELECT \* FROM T1 JOIN T2 USING (COLUMN);

NATURAL JOIN

동일한 이름을 가지는 모든 컬럼에 대해 EQUI JOIN 수행

데이터 유형 동일해야 하고, 접두사 사용 불가

SELECT \* FROM T1 NATURAL JOIN T2;

CROSS JOIN

JOIN 조건이 없는 경우 출력 가능한 모든 조합(카타시안 곱) 출력

ANSI 표준은 CROSS JOIN이라고 명시해야 함

OUTER JOIN

JOIN 조건에서 동일한 값이 없는 행도 반환할 때 사용

LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN

ORACLE 표준은 WHERE 절의 조인 조건을 작성함

SELECT \* FROM S, P WHERE S.ID = P.ID(+)

WHERE 절에서 기준이 되는 테이블 반대 테이블 컬럼 뒤에 (+)를 붙임

SELECT \* FROM S LEFT OUTER JOIN P ON S.ID = P.ID

오라클에서는 UNION을 이용해야 함

SELECT \* FROM S, P WHERE S.ID = P.ID(+)

UNION

SELECT \* FROM S, P WHERE S.ID(+) = P.ID