

■ 데이터사전

데이터사전에는 데이터베이스의 데이터를 제외한 **DBMS가 관리하는 데이터**가 있다.

데이터 사전의 내용을 변경하는 권한은 시스템 사용자가 가진다

반면 일반 사용자에게는 단순 조회만 가능한 읽기 전용 테이블 형태가 제공된다

데이터를 제외한 **모든 정보**라는 점은 데이터의 데이터를 말한다,

데이터 사전은 **메타 데이터(Meta data)**로 구성되어 있음을 의미한다

■ 데이터 사전 내용

- 사용자 정보(user 정보)
- 데이터베이스 객체 (테이블, 인덱스, 뷰)
- 무결성 제약 상태
- 함수, 프로시저 및 트리거 정보

■ DDL 개념

스키마 : DBMS의 특징과 구축 환경에 기반한 데이터 구조

도메인 : 속성의 데이터 타입과 제약 조건을 등을 설정한 정보

테이블 : 데이터 저장 공간

뷰 : 1개 이상의 물리 테이블을 통해서 만드는 가상의 논리 테이블

인덱스 : 빠른 검색을 위한 데이터 구조

■ DDL의 유형

CREATE: 오브젝트 생성

ALTER : 오브젝트 변경

DROP : 오브젝트 삭제

TRUNCATE : 오브젝트 내용 삭제

-테이블변경

alter table 테이블명 **ADD 컬럼명 데이터 타입 (컬럼추가)**

alter table 테이블명 **MODIFY 컬럼명 데이터 타입 (컬럼 type변경)**

alter table 테이블명 **DROP 컬럼명 (컬럼삭제)**

- 테이블삭제 : drop table 테이블명

- 테이블내용삭제 : truncate table 테이블명

- 테이블명 변경 : (테이블명 변경방법 2가지)

RENAME table 이전테이블명 to 새로운테이블명

ALTER TABLE 이전테이블명 RENAME 새로운테이블명

테이블 생성에 사용되는 제약 조건

제약조건	설명
primary key	기본키정의함 not null, unique 제약사항이 설정됨
foreign key	외래키를 정의함 참조대상 테이블명 형식으로 작성해야 함 참조무결성이 위배되는 상황 발생시 다음 옵션으로 처리가능함 CASCADE, NO ACTION, SET NULL, SET DEFAULT
unique	테이블에서 해당하는 열값은 유일해야 함 (예: 전화번호, 이메일)
not null	테이블의 열의 값은 null이 불가능 반드시 입력되어야 하는 항목
check	사용자가 직접정의하는 제약조건

■다중테이블 조회

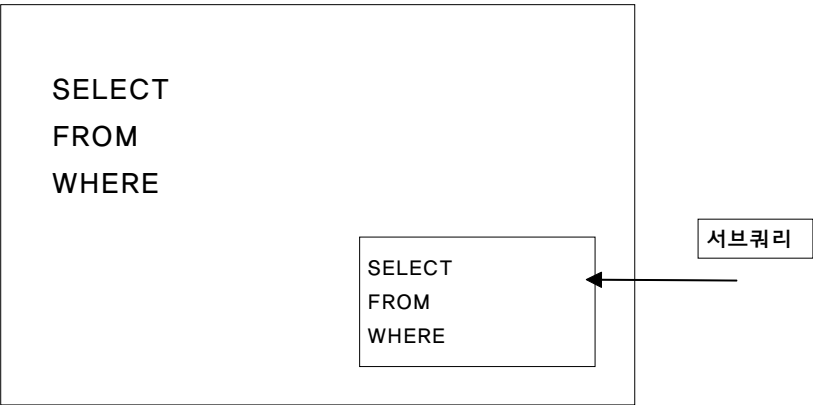
- 조인, 서브쿼리, 집합연산자

조인의 유형

조인유형	
내부조인(Inner Join)	두 테이블에 공통으로 있는 컬럼을 활용하는 유형(공통컬럼기반)
-동등조인(Equi Join)	공통으로 있는 컬럼값이 같은 경우에 레코드 추출
-자연조인(Natural Join)	두 테이블에 있는 동일한 컬럼명을 기준으로 모든 컬럼값이 같은 경우에 레코드 추출 (같은컬럼명으로 알아서 조인함, 주의를 기울려야 함)
-교차조인(cross join)	조인조건이 없는 모든 데이터의 조합을 추출 (조인시 on절이 없게되면 모든 경우의 조합의 조인됨) 가능한 모든행들의 조합을 결과로 만들어 냄
외부조인(Outer join)	
- left Outer join	왼쪽테이블의 모든데이터와 오른쪽테이블의 동일 데이터 추출
- right Outer join	오른쪽테이블의 모든데이터와 왼쪽 테이블의 동일 데이터 추출
- full Outer Join	양쪽테이블의 모든 데이터 추출(동일한 값이 없더라도)

■서브쿼리

:sql문안에
포함된 또다른 SQL
a문이다.



■서브쿼리종류: 비연관서브쿼리 (독자적실행) , 연관서브쿼리

■서브쿼리의 종류 :

Single row(단일행)서브쿼리 : 서브쿼리의 결과가 항상1건 이하인 서브쿼리 (- , < , > ,<>)

multiple row(다중행) 서브쿼리 : 서브쿼리 실행 결과가 여러 건인 서브쿼리

다중 행의 비교 연산자에는 IN, EXISTS, ALL , ANU등이 사용

multiple column(다중컬럼) 서브쿼리 :

서브쿼리의 실행결과가 2개 이상 컬럼으로 반환 되는 쿼리

■집합연산자 (행으로 합치기)

UNION : 2개 이상 SQL문의 실행결과에 대한 중복을 제거한 합집합

UNION ALL : 2개 이상 SQL문의 실행 결과에 대한 중복을 제거하지 않은 합집합

INTERSECTION : 2개이상 SQL문의 실행결과에 대한 중복을 제거한 교집합

EXCEPT : 선행 SQL문의 실행 결과와 후행 SQL문의 실행결과 사이의 중복을 제거한 차집합 (MINUS)

■DCL개념

DCL조작대상 : 사용자권한(접근통제), 트랜잭션(안전하고 무결한 거래 보장)

트랜잭션 : 동시 다발적으로 발생하는 다수 작업을 독립적이고 안전하게 처리하기 위한 데이터베이스 작업단위

DCL : GRANT , REVOKE

TCL :

COMMIT : 작업거래 내용 확정

ROLLBACK :작업거래 내용 취소

CHECKPOINT(작업거래의 저장 시점 설정: Rollback위치지정)

트랜잭션

: 트랜잭션은 논리적인 연산단위이다

하나 이상의 sql문이 포함된다

트랜잭션은 거래를 의미한다

거래의 모든 결과가 모두 반영되거나 모두 취소되어야 한다

분해되지 않는 최소 단위이다

■ 집합연산으로 다중 테이블을 조회한다

(행단위로 여러 테이블을 합치는 개념) , 조인은 가로로 합치기, union은 행으로 합치기

select column1, column2

union

select column1, column2

두 집합 사이에서 집합 연산자가 작동하도록 select문 사이에 연산자를 배치한다

- union으로 연결된 select문에 나타나는 컬럼의 숫자는 같아야 한다

- union으로 대응되는 각 컬럼의 데이터 타입이 같아야 한다

■ 트랜잭션 (transaction) 제어 관련 dcl을 수행한다

트랜잭션의 특징:

원자성 (Atomicity) :트랜잭션 내에서 모두 실행되거나 모두 실행되지 않음 All or nothing

일관성 (Consistency) : 트랜잭션의 실행 전/후 결과가 동일해야 함 무결성

고립성 (Isolation) : 트랜잭션이 실행되는 도중에 다른 트랜잭션의 영향을 받지 않음 독립성

지속성(Durability) : 트랜잭션의 결과는 지속적으로 저장되어 보존됨 장애 대응성

TCL 명령어의 메모리 관점 동작

Commit: 메모리의 내용을 하드디스크에 저장함

Rollback: 메모리의 내용을 하드디스크에 저장하지않고 버림

CheckPoint : rollback범위설정을 위해 메모리상 경계를 설정함

트랜잭션 시나리오

1트랜잭션 시작

2 insert

3 savepoint p1

4 update

5 savepoint p2

6 delete

7 현재위치에서 rollback 예정 (복구지점 사용에 따라 결과 달라짐)

■오라클 데이터사전영역

DBA_TABLES :모든 테이블목록확인

DBA_INDEXES : 모든 인덱스 목록 확인

USER_TABLES :자기 계정의 테이블 목록 확인

■인덱스와 뷰 작성

-인덱스의 개념

저장된 데이터를 빠르게 검색할 수 있는 수단이자, 테이블에 대한 조회속도를 높여 주는 자료구조를 말한다

=> 테이블 조회시 full scan함(테이블의 데이터는 정렬되어 있지 않음 ,처음부터 순차적으로 다 찾음): 찾고자 하는 데이터를 찾기위해서 처음부터 전체를 찾아봐야 함

정렬된 index테이블을 통해서 찾고하는 하는 항목을 빠르게 찾은 후 전체데이터를 빠르게 찾아옴

인덱스 생성

CREATE INDEX <인덱스명> ON 테이블명 < 컬럼명 나열> ;

인덱스 삭제

`DROP INDEX <인덱스명> ;`

인덱스 변경

`ALTER INDEX <인덱스명> ON 테이블명 <컬럼명>;`

■ 뷰

뷰장점: 논리적독립성제공, 사용자 데이터 관리 용이, 데이터 보안용이

뷰단점: 뷰의 인덱스 사용 불가, 뷰구조변경 불가, 데이터변경 제약 존재

뷰생성:

`create view <뷰이름> <컬럼목록> as <뷰를 통해 보여줄 데이터 조회용 쿼리문>`

뷰삭제 및 변경 : (뷰구조 변경이 불가능)

`drop view <view이름>`

뷰내용변경은 가능하다.

■ 인덱스 설계과정

1. 접근경로수집 (검색조건을 의미)
2. 분포도 조사를 통한 후보컬럼 설정
3. 접근경로 결정
4. 컬럼조합 및 순서결정
5. 적용 시험