

Summary

DB(DataBase) : 데이터의 집합체 RDBMS(Relation DB Management System) - Oracle, mysql-- >maria, MS-sql -GUI(Graphic User Interface) tool : SQL Developer -Oracle : EnterpriseEdition(서버-orcl) // ExpressEdition(경량-XE) 11g

SQL(Structured Query Language) : 구조적 질의 언어

- DB에 적재된 데이터들을 다루는 언어

=====

DDL(Data Definition Language) : 데이터 정의어

- Create Select Alter Drop
create table 테이블명 (empno number(7) primary key, -- PK
col2 varchar2(10),
deptno number referenes dept(deptno)); --FK
- Primary Key : 모든 데이터에서 유일하게 구분되어지는 값/컬럼
- Foreign Key : 남의 테이블의 PK를 참조해서 사용하는 값/컬럼
- RDB(Relation 관계) - 일반적으로 PK-FK, 두 테이블에 간에 관계를 맺는 것

Drop table 테이블명; --테이블객체 자체를 지운다

Truncate table 테이블명; -- 백업 로그를 남기지 않고 순삭

delete from 테이블명 cascade; -- 딸린 자식이 있어도 지운다

DML(Data Manipulation Language) : 데이터 조작어

- CRUD : Insert Select Update Delete
- Insert Into 테이블 values(1,'aa',99);
- Insert Into 테이블(col1, col2) values(1,'aa');
-----null : unknown value : 값 취급을 하지 않는다.-----
- 1. Insert Into 테이블(col1, col2) values(1,'aa'); --col3 취급하지 않는다.
- 2. Insert Into 테이블(col1, col2) values(1,");
- 3. Insert Into 테이블(col1, col2) values(1,null);
- 4. select * from 테이블 where col3 is not null; ----- col3 is not null
- 5. update 테이블 set col3=" where col1 = 1;
- 6. update 테이블 set col3=null where col1 = 1; ----- col3=null

Read

- 1) select * from 테이블;
- 2) select * from 테이블 where col1=1 and/or col2 is null;
- 3) select col1, col2 from 테이블 where col1=1 and/or col2 is null;

-----subquery : 값을 대체할 수 있는 어느 곳이든 관계없다.

: from 절에 놓이는 subquery를 inline view 라고 한다.

4) select * from emp where job = (select job from emp where empno=7799);

5) select * from emp where (job, sal)

= (select job, sal from emp where empno=7799);

6) subquery 결과 값이 멀티일경우 : any all in

select * from emp where (job, sal)

in (select job, sal from emp where deptno=10);

-----join : from 절에 2개이상 테이블 이름을 명시한 경우

1) JOIN (inner join / self join)

select e.ename, d.dname from emp e, dept d where e.deptno = d.deptno;

select e.empno, cp.empno from emp e, emp cp where e.mar = cp.empno;

select e.ename, d.dname from emp e JOIN dept d ON e.deptno = d.deptno;

2) OUTER JOIN (left outer join , right outer join)

--- 부족한 쪽에 (+)

select e.ename, d.dname from emp e, dept d where e.deptno(+) = d.deptno;

--- 종류(distinct)를 더 많이 가진 쪽이 기준

select e.ename, d.dname from emp e

RIGHT OUTER JOIN dept d ON e.deptno = d.deptno;

-----Group : group by ~ having

1) 그룹함수 : min(값/컬럼), max() count() avg() sum()

2) 그룹함수와 일반컬럼을 select에 같이 쓸겨우 반드시 일반컬럼을 group by 에 넣어라

select sum(sal) from emp;

select deptno, sum(sal) from emp group by deptno;

3) 그룹함수의 결과값 : multi(3건) .. single(1건) ----- 예러

select deptno, max(sum(sal)) from emp group by deptno;

4) having : 그룹의 조건절 == select..where

그룹함수에 대한 조건은 무조건 having에

select sum(sal) from emp where sum(sal) > 2000; ---- 예러 : 그룹함수는 where에 사용 불가

select deptno, sum(sal) from emp group by sal having sum(sal) > 2000;

Update

- 1) update emp set sal=2000, detpno=30;
- 2) update emp set sal=2000, detpno=30 where emptno=777;

Delete

- 1) delete from emp;
- 2) delete from emp where emptno=777;
- 3) select * from emp where emptno=777;

DCL(Data Control Language) : 데이터 제어어

commit/rollback ... savepoint MYPOINT;rollback to MYPOINT;

DDL(Data Definition Language): 데이터 정의어

1. USER
create user ai identified by "0000";
grant dba to ai;
grant connect, resources to ai;
2. TABLE
create table emp(empno number, ename varchar2(20));
3. FUNCTION
create function myfunc(prm varchar2(2))
is ~~~~~;
4. INDEX
create index myindx ~~~~~;
5. SEQUENCE : 자동번호표 == PK
where seq = 5; (pk로 쓸만하게 없으면, 예시 - 게시판)
 - 구분만 되면 된다.
 - 이쁘게 정렬될 필요가 없다.create sequence myseq
6. VIEW @- 가상의 테이블
create view myview~~~~~;

인덱스란?

: 검색 속도를 향상

: 랜덤 액세스하기 위한 목적

create index 이름 on 테이블(컬럼1); _____ 단일 인덱스

create index 이름 on 테이블(컬럼1, 컬럼2); _____ 결합 인덱스

: 수정/삭제가 빈번한 테이블일 경우 인덱스를 사용하게 되면 속도가 느려진다.
: 전체 데이터 중에서 10~15% 이내의 데이터를 검색하는 경우 효율적이다.

함수 기반 인덱스

create index 이름 on 테이블(sal*12);

뷰의 개념

: 메모리에서 주소만 가르키는 것이지 디스크 사용량은 없다.
: 여기저기서 막 가져와야 할 때,
: 뷰는 테이블처럼 쓰인다.

테이블 복제

create table test_222 AS (select * from emp where deptno=10); ---구조 + 데이터

create table test_222 AS (select * from emp where 1=0); ---구조 + 데이터

create view (뷰 이름) AS (subquery)

create view test_222 AS (select * from emp where deptno=10);-----참조

create or replace → 기존에 있으면 덮어써라..!

뷰의 장점

: 데이터를 보호하기 위한 보안
: 사용자 편의성

인라인 뷰(inline view)

: from 절에서 사용하는 것.

create or replace

: 매번 drop하기 귀찮으므로 덮어쓰기를 위한 코드.
: 이게 매우 편하다.

시퀀스(Sequence)

: 유일한 숫자를 생성해주는 오라클 객체

: 시퀀스를 생성하면 기본키와 같이 순차적으로 증가하는 컬럼을 자동적 생성할 수 있음.

: 테이블과 독립적으로 저장 및 생성

(하나의 시퀀스를 여러 테이블에서 사용 가능)

시퀀스 생성

스키마(A): AI

이름(B): MYSEQ

속성 DDL

다음으로 시작(C): 1

증분(D): 1

최소값(E): 1

최대값(F): 9999999

캐시(G): <지정되지 않음>

캐시 크기(I):

주기(J): <지정되지 않음>

정렬(K): <지정되지 않음>

도움말(H) 확인 취소

```
44 select MYSEQ.nextval from dual; --증이를 한번 빼야한다.
45 select MYSEQ.currval from dual;
```

스크립트 출력 x | 질의 결과 x

SQL | 인출된 모든 행: 1(0.001초)

CURRVAL
1

⇒ 번호표 뽑고 현재 시퀀스 확인하는 코드

```
drop sequence MYSEQ;
```

⇒ 시퀀스 삭제

```
alter sequence MYSEQ increment by 2; --2씩 증가시키겠다.
```

⇒ 시퀀스 2씩 증가

```
insert into test_3(empno) values(MYSEQ.nextval);  
insert into test_3(empno) values(MYSEQ.nextval);  
  
insert into test_3(empno, ename, deptno) values(MYSEQ.nextval, 'adam' || MYSEQ.nextval, 10);
```

⇒ 시퀀스 insert 해주기