처음에 Run SQL command line 접속해서 페이지 설정할 때

set linesize 300; 과 set pagesize 300; 을 같이 설정해주어야 테이블이 잘 나온다.

조인(JOIN)

두 개의 테이블에서 원하는 데이터를 가져다 쓰는 것을 조인(join)이라고 한다.

두 테이블의 연관성은 deptno로 연관되어 있는데 이처럼 연관이 되어 있어야만 join이 가능하다.

즉, primary key, foreign key 관계에서만 사용이 가능하다.

조인조건

두 개 이상의 테이블에서 하나의 컬럼을 뽑아서 원하는 컬럼을 하나의 테이블로 만들고 싶을 때 사용한다.

select dname,ename,sal

from emp,dept

where emp.deptno = dept.deptno;

기본 조인 방식이 내부 조인과 외부 조인이 있다.

내부 조인은 정확하게 일치하는 데이터를 가져올 때 사용한다.

조인, 서브쿼리, DML, DDL

오후에 여태까지 했던 거 다 정리할 예문

데이터가 늘어나는데 중복 데이터들이 계속 쌓이면 분리를 해내는 작업이 필요하다.

1. 조인

-정규화된 테이블이나 혹은 일반적으로 작성된 여러 테이블의 컬럼을 이용해서 데이터를 조회하는 것을 조인이라고 한다.

-조인은 관계형 데이터베이스에 반드시 알아야 하는 개념

-기본키와 외래키의 관계를 이용해서 테이블을 조인

외래키를 가지고 기본키 테이블에서 값을 비교하여 작업이 진행된다.

-조인을 하는 경우 무조건 where절에 조인조건을 정의해야 한다.

-테이블을 여러 개 사용하는 경우 모든 테이블들의 조인조건을 정의해야 하며 select절에서 사용하지 않고 조건으로만 사용한다고 하더라도 조인조건은 정의해야 한다.

[조인작성방법]

1. from절에 조회하고 싶은 데이터가 저장된 테이블들을 모두 명시

2. 조인을 하는 경우 컬럼이 어떤 테이블의 컬럼인지 명확하게 정의하기 위해 “테이블명.컬럼명”으로 액세스를 한다.

3. from절에 테이블명읠 정의할 때 alias를 함께 추가하여 alias를 통해 액세스하도록 한다.

select alias1.컬럼명, alias2.컬럼명 …

from 테이블1 alias1, 테이블2 alias2

4. where절에는 반드시 조인조건을 추가하며 조인조건에는 두 테이블의 값을 비교하기 위해 정의하는 것이므로 외래키와 기본키를 정의한다.

외래키테이블(child 테이블)에 정의된 컬럼값을 기본키테이블(parent 테이블)에서 비교하여 정확하게 일치하는 경우 값을 가져온다.

(테이블이 두 개 -> 조인조건 1개, 테이블이 세 개 -> 조인조건2개)

[실습예제]

부서별 인원수 구하기 (단, 부서명으로 출력하기)

select d.dname, count(e.empno)

from emp e, dept d

where e.deptno = d.deptno;

group by d.dname;

컬럼명을 찾을 때에는 desc 테이블명 해서 찾아야 한다.

버림

select floor(125.8) from dual;

125

select trunc(125.8,0) from dual;

125

소수점 올림

select ceil(125.2) from dual;

126

지금까지 한 조인은 정확히 일치하는 테이블에 대한 것이었다.

지금부터 조인의 종류들을 확인할 것이다.

2) 조인 종류

-equi join(inner join): 두 테이블에서 정확하게 일치하는 컬럼에 대한 데이터만 조인

-outer join: 두 개 이상의 테이블에 조인을 적용했을 때 join조건을 만족하지 않아도 데이터를 조회하고 싶은 경우 사용

조인조건에 (+)를 추가한다.

join조건을 만족하지 않아도 한쪽 테이블의 모든 데이터를 출력하고 싶을 때 사용하는 조인방식으로 정보가 부족한 테이블에 컬럼에 (+)를 추가한다.

select 테이블 alias.컬럼명….

from 테이블 alias

where 테이블 alias.컬럼명 = 테이블 alias.컬럼명(+)

or

where 테이블 alias.컬럼명(+) = 테이블 alias.컬럼명

=>(+)기호는 한쪽에만 사용한다.

데이터가 확장되어야 하는 곳에 추가한다.

select dname,ename,hiredate

from emp e, dept d

where e.deptno=d.deptno(+);

컬럼명 뒤에 +기호를 붙여 보고 싶은 데이터 출력한다.

채워져야 되는데 채워지지 않은 곳에 +기호를 붙여서 확장해야 하는 곳에 +기호를 붙인다.

[실습예제]

1. 부서별 인원수를 출력하기 (단, 부서명으로 출력하고 인원이 없는 부서명도 모두 출력한다.)

select d.deptno, count(e.empno)

from emp e, dept d

where e.deptno=d.deptno(+)

group by d.deptno;

2. 부서별 인원수를 출력하기

단, 부서명을 출력하고 부서가 배치되지 않은 사원은 “신입사원”으로 표시한다.

select nvl(to\_char(d.dname),'신입사원'), count(e.empno)

from emp e, dept d

where e.deptno=d.deptno(+)

group by d.dname;

-self조인: 두 개 이상의 테이블에서 조인하지 않고 같은 테이블의 컬럼들을 이용해서 조인(하나를 가상테이블로)

select e.empno, e.ename, e.mgr, m.ename

from emp e, emp m

where e.mgr = m.empno;

[실습]

관리자별 인원수를 구하시오.

관리자명 인원수

select m.ename 관리자명, count(e.empno) 인원수

from emp m, emp e

where e.mgr = m.empno(+)

group by m.ename;

관리자가 배정 안 된 사원까지 포함해서 출력

select m.ename 관리자명, count(e.empno) 인원수

from emp m, emp e

where e.mgr = m.empno(+)

group by m.ename;

view는 내가 단지 원하는 조건만 뽑아내서 사용할 수 있다.

view는 실험결과로 보이게 하는 가상의 테이블이다.

1. 서브쿼리

-SQL문에 삽입된 Query

-select문에서 주로 사용하고 select문에 삽입된 select문

바깥쪽의 query를 main query, 안쪽에 삽입된 query를 sub query라고 한다.

-sub query는 괄호로 묶어 주어야 한다.

-sub query 메인 쿼리가 실행되기 전에 한 번 실행되며 그 실행결과를 메인 쿼리에서 사용한다.

create view ???

select avg(sal)

from emp

where deptno=10;

select ename,sal

from emp

where sal> 2916.66667

select deptno, avg(sal)

from emp

group by deptno;

conn system/manager

grant create view to scott;

conn scott/tiger

create view countdata

as

select deptno, avg(sal) empcount

from emp

group by deptno;

select \* from countdata;

[실습]

10번 부서의 평균 급여보다 급여를 많이 받는 사원들을 조회

select ename, sal

from emp

where sal>(select avg(sal)

from emp

where deptno=10)

쿼리 안에 쿼리를 넣어 사용하는 것을 서브쿼리라고 한다.

[서브쿼리의 종류]

1)단일행서브쿼리 – 결과가 1행 1열인 서브쿼리

>,>=,<,<= 연산자와 함께 사용할 수 있다.

비교 연산자 오른쪽에 서브쿼리를 추가해야 한다.

[실습]

-전체 평균보다 높은 급여를 받는 사원의 목록(ename,sal)

-smith와 같은 job을 갖고 있는 사원의 목록(ename,job,hiredate)

-10번 부서에 근무하는 사원들의 job과 같은 job을 갖고 있는 사원의 목록

select ename, sal

from emp

where sal>(select avg(sal)

from emp);

select ename,job,hiredate

from emp

where job = (select job

from emp

where ename='SMITH');

이런 경우를 사용하고 싶은 경우

select \*

from emp

where job = (select job

from emp

where deptno=10);

위와 같이 했을 때 아래와 같은 에러가 발생한다.

single-row subquery returns more than one row

이 에러를 해결하기 위해서 = 부분을 IN으로 대체한다.

select \*

from emp

where job IN (select job

from emp

where deptno=10);

2) 다중행 서브쿼리

-서브쿼리의 실행결과가 열 하나의 행이 여러개 인 경우

- =연산자와 같은 비교연산자를 사용할 수 없다.

-in: 컬럼의 값이 정확하게 일치하는 경우

-<any: 서브쿼리 결과의 최대값보다 작은 값을 구하는 경우

select ename,sal

from emp

where sal < any(select sal

from emp

where deptno=10);

->any: 서브쿼리 결과의 최소값보다 큰 값을 구하는 경우

select ename,sal

from emp

where sal >any(select sal

from emp

where deptno=10);

-<all: 서브쿼리 결과의 최소값보다 작은 값을 구하는 경우

select ename,sal

from emp

where sal < all(select sal

from emp

where deptno=10);

ENAME SAL

-------------------- ----------

WARD 1250

MARTIN 1250

ADAMS 1100

JAMES 950

SMITH 800

->all: 서브쿼리 결과의 최대값보다 큰 값을 구하는 경우

select ename,sal

from emp

where sal > all(select sal

from emp

where deptno=10);

-10번 부서에 근무하는 사원들의 job과 같은 job을 갖고 있는 사원의 목록

3)다중 컬럼 서브쿼리

-두 개 이상의 컬럼과 다중행을 반환하는 서브쿼리

-메인쿼리 비교 컬럼의 개수,종류가 서브쿼리의 반환 결과와 동일

….

where (컬럼1,컬럼2) in (select 컬럼1,컬럼2

……….)

[실습]

-각 부서별로 최소급여를 받는 사원의 정보를 출력(사원명,부서코드,급여,입사일)

select ename,deptno,sal,hiredate

from emp

where sal in (select min(sal)

from emp

group by deptno);

위와 같이 하면 sal만 비교했기 때문에 아래와 같이 결과가 나온다.

ENAME DEPTNO SAL HIREDATE

-------------------- ---------- ---------- --------

SMITH 20 800 80/12/17

JAMES 30 950 81/12/03

MILLER 10 1300 82/01/23

장동건 20 1300 19/12/20

select ename,deptno,sal,hiredate

from emp

where (deptno,sal) in (select deptno,min(sal)

from emp

group by deptno);

그래서 아래와 같이 두 컬럼을 비교하여 중복될 요소를 제거한다.

ENAME DEPTNO SAL HIREDATE

-------------------- ---------- ---------- --------

SMITH 20 800 80/12/17

JAMES 30 950 81/12/03

MILLER 10 1300 82/01/23

select ename,deptno,sal,hiredate

from emp

where (deptno,sal) in (select deptno,min(sal)

from emp

group by deptno)

order by deptno;

4) 상관형서브쿼리(상호연관서브쿼리)

-메인 쿼리의 값이 서브쿼리에서 사용되는 경우

-메인쿼리 한 row에 대해 서브쿼리가 한 번씩 실행된다.

-메인쿼리의 값이 어떤 값이냐에 따라 서브쿼리의 결과가 달라진다.

[실행]

1. 메인쿼리에서 비교할 값을 가져온다.

2. 메인쿼리에서 받은 값을 이용해서 서브쿼리가 실행된다.

3. 서브쿼리의 실행결과로 메인쿼리가 실행된다.

4. 메인쿼리의 레코드수만큼 반복된다.

[실습]

소속급여의 급여 평균보다 급여가 많은 사원들의 정보를 출력(ename,deptno,sal)

select ename,deptno,sal

from emp outer

where sal >(select avg(sal)

from emp e

where e.deptno = outer.deptno);

현재 작업중인 부서코드가 와야 한다.

\*검산하는 방법

select deptno,avg(sal)

from emp

group by deptno;

DEPTNO AVG(SAL)

------ ----------

30 1566.66667

20 2029.16667

10 2916.66667

5) from절에서 사용하는 서브쿼리(inline view)

-from절에 서브쿼리를 추가해서 사용

-서브쿼리 결과를 가상 테이블로 사용하겠다는 의미

-from절에 추가되는 서브쿼리는 alias를 정의해야 한다.

-from절에 추가되는 서브쿼리 내부의 컬럼은 실제 컬럼처럼

메인쿼리에서 사용해야 하므로 컬럼도 컬럼명이 존재하거나 alias를 정의해야 한다.

select 컬럼명1,…….

from ( select 컬럼…..

from 테이블명

where…

group by….)alias

소속부서의 급여 평균보다 급여가 많은 사원들의 정보를 출력

=> 조인과 from절에 추가하는 서브쿼리를 이용해서 작업

select deptcode, countdata

from (select deptno as deptcode, count(empno) as countdata

from emp

group by deptno) mytable;

select e.ename, e.deptno, e.sal, d.avgsal

from emp e, (select deptno, avg(sal) as avgsal

from emp

group by deptno) d

where e.deptno=d.deptno and e.sal > d.avgsal;

DEPTNO ENAME SAL AVGSAL

------ -------------------- ---------- ----------

30 BLAKE 2850 1566.66667

30 ALLEN 1600 1566.66667

20 FORD 3000 2029.16667

20 SCOTT 3000 2029.16667

20 JONES 2975 2029.16667

10 KING 5000 2916.66667

create table member(

create table member(

id varchar2(10),

pass varchar2(10),

addr varchar2(20));

insert into member values('jang','1234','서울');

insert all

into member values('lee','1234','인천')

into member values('kang','1234','안산')

into member values('hong','1234','수원')

select \* from dual;

insert into member values('jjang',null,null);

for문으로 돌려서 값을 만들 것

확인

select \* from member;

delete member; 테이블 삭제할 때 사용한다.

※ 다시 테이블을 되돌리고 싶을 때;

rollback; 다시 되돌리다.

delete from member

where id='jjang';

select \* from member;

id='jang'인 사람의 addr을 id='lee'인 사람의 주소로 수정하고 싶을 때

update member

set addr=(select addr

from member

where id='jang')

where id='lee';

확인

select \* from member;

delete from member

where addr = (select addr

from member

where id='jang');

select \* from member;