**데이터베이스 시스템 과제물 제출**

**1826074 오현진**

**지도교수 : 오염덕교수님**

**제출일자 : 22.10.31**



**[문제 1] 테이블 생성하기**

SQL> select \* from cat\_a;

N N

- -

1 A

2 B

SQL> select \* from cat\_b;

N N

- -

1 C

2 D

SQL> select \* from cat\_c;

N N

- -

1 E

2 F

**테이블이 이미 생성되어 있어 해당 테이블을 검색했습니다.**

**[문제2] 카티션 곱 생성하기**

SQL> select \* from cat\_a, cat\_b;

N N N N

- - - -

1 A 1 C

1 A 2 D

2 B 1 C

2 B 2 D

SQL> select \* from cat\_a, cat\_b, cat\_c;

N N N N N N

- - - - - -

1 A 1 C 1 E

1 A 1 C 2 F

1 A 2 D 1 E

1 A 2 D 2 F

2 B 1 C 1 E

2 B 1 C 2 F

2 B 2 D 1 E

2 B 2 D 2 F

**카티션 곱 결과, 2x2 의 경우는 4개가 나왔으며, 2x2x2의 경우, 6개가 나왔습니다.**

**[문제3] JOIN 과 카티션 곱**

SQL> select a.name, b.name, c.name

2 from cat\_a a, cat\_b b, cat\_c c

3 where a.no = b.no and a.no = c.no;

N N N

- - -

A C E

B D F

SQL> select a.name, b.name, c.name

2 from cat\_a a, cat\_b b, cat\_c c

3 where a.no = b.no;

N N N

- - -

A C E

A C F

B D E

B D F

**결과가 원하는 대로 나왔습니다.**

**[문제4] 카티션 곱을 사용한 집합 세트 구성**

SQL> select \* from

2 (select empno, ename, job, sal

3 from emp

4 where deptno = 10

5 ), (select level c1 from dual connect by level <= 3);

EMPNO ENAME JOB SAL C1

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

7839 KING PRESIDENT 5000 1

7782 CLARK MANAGER 2450 1

7934 MILLER CLERK 1300 1

1000 Tiger 3600 1

1000 Tiger 3600 1

7839 KING PRESIDENT 5000 2

7782 CLARK MANAGER 2450 2

7934 MILLER CLERK 1300 2

1000 Tiger 3600 2

1000 Tiger 3600 2

7839 KING PRESIDENT 5000 3

EMPNO ENAME JOB SAL C1

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

7782 CLARK MANAGER 2450 3

7934 MILLER CLERK 1300 3

1000 Tiger 3600 3

1000 Tiger 3600 3

**level별로 개의 집합 세트가 나왔습니다.**

**[문제5] 양쪽 모두에 해당하는 칼럼의 경우**

SQL> select empno, ename, deptno, dname

2 from emp e, dept d

3 where e.deptno = d.deptno;

select empno, ename, deptno, dname

\*

ERROR at line 1:

ORA-00918: column ambiguously defined

SQL>

SQL> select empno, ename, d.deptno, dname

2 from emp e, dept d

3 where e.deptno = d.deptno;

EMPNO ENAME DEPTNO DNAME

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

7839 KING 10 ACCOUNTING

7698 BLAKE 30 SALES

7782 CLARK 10 ACCOUNTING

7566 JONES 20 RESEARCH

7788 SCOTT 20 RESEARCH

7902 FORD 20 RESEARCH

7369 SMITH 20 RESEARCH

7499 ALLEN 30 SALES

7521 WARD 30 SALES

7654 MARTIN 30 SALES

7844 TURNER 30 SALES

EMPNO ENAME DEPTNO DNAME

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

7876 ADAMS 20 RESEARCH

7900 JAMES 30 SALES

7934 MILLER 10 ACCOUNTING

1000 Tiger 10 ACCOUNTING

1000 Tiger 10 ACCOUNTING

2000 Cat 30 SALES

**양 테이블 모두에 해당하는 칼럼인 deptno를 d.deptno로 명시하여 해결하였습니다.**

**[문제6] oracle 조인과 ansi 조인 실습**

(1) oracle 조인과 between 구문을 이용한 경우,

SQL> select c.name "CUST\_NAME", to\_char(c.point,'999,999') "POINT",

2 g.name "GIFT\_NAME"

3 from customer c, gift g

4 where c.point between g.gstart and g.gend;

CUST\_NAME POINT GIFT\_NAME

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

jack 164,230 tabletPC

mike 76,520 cpu

mike 76,520 keyboard

chris 34,250 keyboard

chris 34,250 mouse

(2) ansi 조인과 between 구문을 이용한 경우,

SQL> select c.name "CUST\_NAME", to\_char(c.point,'999,999') "POINT",

2 g.name "GIFT\_NAME"

3 from customer c join gift g

4 on c.point between g.gstart and g.gend;

CUST\_NAME POINT GIFT\_NAME

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

jack 164,230 tabletPC

mike 76,520 cpu

mike 76,520 keyboard

chris 34,250 keyboard

chris 34,250 mouse

(3) oracle 조인을 이용하고, between 구문을 이용하지 않은 경우,

SQL> select c.name "CUST\_NAME", to\_char(c.point,'999,999') "POINT",

2 g.name "GIFT\_NAME"

3 from customer c, gift g

4 where c.point >= g.gstart

5 and c.point <= g.gend;

CUST\_NAME POINT GIFT\_NAME

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

jack 164,230 tabletPC

mike 76,520 cpu

mike 76,520 keyboard

chris 34,250 keyboard

chris 34,250 mouse

**oracle조인과 ansi조인 및 between 구문의 사용 여부에 따른 차이를 실습했습니다.**

**[문제7] oracle 비등가 조인과 ansi 비등가 조인 실습**

(1) student, score, hakjum 테이블 생성 및 조회

SQL> select \* from student;

STUDNO NAME PROFNO

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

1 jack 52

2 mike 78

3 chris 89

4 wayne 93

5 devio 46

6 califa 67

7 antonio 74

8 cledor 59

9 kantona 92

9 rows selected.

SQL> select \* from score;

STUDNO TOTAL

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

1 85

2 71

3 48

4 51

5 77

6 92

7 36

8 74

9 54

9 rows selected.

SQL> select \* from hakjum;

GR MIN MAX

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

A+ 91 100

A0 81 90

B+ 71 80

B0 61 70

C+ 51 60

C0 41 50

D+ 31 40

D0 21 30

F 0 20

9 rows selected.

(2) oracle 비등가 join

SQL> select s.name "STU\_NAME", o.total "SCORE", h.grade "CREDIT"

2 from student s, score o, hakjum h

3 where s.studno = o.studno

4 and o.total >= h.min

5 and o.total <= h.max;

STU\_NAME SCORE CR

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

califa 92 A+

jack 85 A0

devio 77 B+

cledor 74 B+

mike 71 B+

kantona 54 C+

wayne 51 C+

chris 48 C0

antonio 36 D+

(3) ansi 비등가 join

SQL> select s.name "STU\_NAME", o.total "SCORE", h.grade "CREDIT"

2 from student s join score o

3 on s.studno = o.studno

4 join hakjum h

5 on o.total >= h.min

6 and o.total <= h.max;

STU\_NAME SCORE CR

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

califa 92 A+

jack 85 A0

devio 77 B+

cledor 74 B+

mike 71 B+

kantona 54 C+

wayne 51 C+

chris 48 C0

antonio 36 D+

**테이블의 경우, 기존에 생성되어 있던 테이블을 이용하였으며, oracle 조인과 ansi 조인의 차이를 실습하였습니다 결과는 동일했습니다.**

**[문제8] oracle outer 조인과 ansi outer 조인 실습**

(1) oracle outer 조인

SQL> select s.name "STU\_NAME", p.name "PROF\_NAME"

2 from student s, professor p

3 where s.profno = p.profno(+);

STU\_NAME PROF\_NAME

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

jack kim

mike lee

chris lim

wayne nam

devio jeong

califa park

antonio yong

cledor hong

kantona kwack

(2) ansi outer 조인

SQL> select s.name "STU\_NAME", p.name "PROF\_NAME"

2 from student s LEFT OUTER JOIN professor p

3 on s.profno = p.profno;

STU\_NAME PROF\_NAME

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

jack kim

mike lee

chris lim

wayne nam

devio jeong

califa park

antonio yong

cledor hong

kantona kwack

**두 경우의 결과가 동일함을 확인하였습니다.**

**[문제9] outer join의 유의사항**

SQL> select d.deptno, d.dname, d.loc, e.empno, e.ename, e.sal

2 from dept d, emp e

3 where d.deptno = e.deptno(+) and e.deptno = 20

4 order by 1;

DEPTNO DNAME LOC EMPNO ENAME SAL

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

20 RESEARCH DALLAS 7566 JONES 2975

20 RESEARCH DALLAS 7788 SCOTT 3000

20 RESEARCH DALLAS 7902 FORD 3000

20 RESEARCH DALLAS 7369 SMITH 800

20 RESEARCH DALLAS 7876 ADAMS 1100

SQL> select d.deptno, d.dname, d.loc, e.empno, e.ename, e.sal

2 from dept d, emp e

3 where d.deptno = e.deptno(+) and e.deptno(+) = 20

4 order by 1;

DEPTNO DNAME LOC EMPNO ENAME SAL

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

10 ACCOUNTING BUSAN

20 RESEARCH DALLAS 7566 JONES 2975

20 RESEARCH DALLAS 7788 SCOTT 3000

20 RESEARCH DALLAS 7902 FORD 3000

20 RESEARCH DALLAS 7369 SMITH 800

20 RESEARCH DALLAS 7876 ADAMS 1100

30 SALES CHICAGO

40 OPERATIONS BOSTON

50 ????

60 DBLAB SOFT212

80 AAA dddd

**위의 경우는 해당 부서와 연결되는 경우만을 출력했으나, 아래의 경우에는 해당 부서와 연결되는 사원이 없어도 부서의 정보를 출력하였습니다.**

**[문제10] oracle self 조인과 ansi self 조인 실습**

(1) oracle self 조인

SQL> select e1.ename "ENAME", e2.ename "MGR\_ENAME"

2 from emp e1, emp e2

3 where e1.mgr = e2.empno;

ENAME MGR\_ENAME

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

JONES KING

CLARK KING

BLAKE KING

JAMES BLAKE

TURNER BLAKE

MARTIN BLAKE

WARD BLAKE

ALLEN BLAKE

MILLER CLARK

FORD JONES

SCOTT JONES

ENAME MGR\_ENAME

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

ADAMS SCOTT

SMITH FORD

(2) ansi self 조인

SQL> select e1.ename "ENAME", e2.ename "MGR\_ENAME"

2 from emp e1 join emp e2

3 on e1.mgr = e2.empno;

ENAME MGR\_ENAME

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

JONES KING

CLARK KING

BLAKE KING

JAMES BLAKE

TURNER BLAKE

MARTIN BLAKE

WARD BLAKE

ALLEN BLAKE

MILLER CLARK

FORD JONES

SCOTT JONES

ENAME MGR\_ENAME

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

ADAMS SCOTT

SMITH FORD

**결과가 동일함을 확인하였습니다.**

**[문제11] join 실습하기**

(1) oracle 조인 구문

SQL> select e1.empno, e1.ename, e1.hiredate,

2 count(nvl2(e2.empno, e1.empno, null)) "CNT"

3 from emp e1, emp e2

4 where e1.hiredate > e2.hiredate(+)

5 group by e1.empno, e1.ename, e1.hiredate

6 order by cnt;

EMPNO ENAME HIREDATE CNT

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

1000 Tiger 0

2000 Cat 0

7499 ALLEN 20-FEB-81 0

7521 WARD 22-FEB-81 1

7900 JAMES 12-MAR-81 2

7902 FORD 12-MAR-81 2

7844 TURNER 09-AUG-81 4

7654 MARTIN 28-SEP-81 5

7698 BLAKE 05-JAN-91 6

7839 KING 17-NOV-96 7

7782 CLARK 06-SEP-99 8

EMPNO ENAME HIREDATE CNT

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

7566 JONES 04-FEB-01 9

7934 MILLER 23-JAN-03 10

7788 SCOTT 17-JUN-03 11

7369 SMITH 01-DEC-07 12

7876 ADAMS 23-MAY-87 13

(2) ansi 조인 구문

SQL> select e1.empno, e1.ename, e1.hiredate,

2 count(nvl2(e2.empno, e1.empno, null)) "CNT"

3 from emp e1 left join emp e2

4 on e1.hiredate > e2.hiredate

5 group by e1.empno, e1.ename, e1.hiredate

6 order by CNT;

EMPNO ENAME HIREDATE CNT

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

1000 Tiger 0

2000 Cat 0

7499 ALLEN 20-FEB-81 0

7521 WARD 22-FEB-81 1

7900 JAMES 12-MAR-81 2

7902 FORD 12-MAR-81 2

7844 TURNER 09-AUG-81 4

7654 MARTIN 28-SEP-81 5

7698 BLAKE 05-JAN-91 6

7839 KING 17-NOV-96 7

7782 CLARK 06-SEP-99 8

EMPNO ENAME HIREDATE CNT

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

7566 JONES 04-FEB-01 9

7934 MILLER 23-JAN-03 10

7788 SCOTT 17-JUN-03 11

7369 SMITH 01-DEC-07 12

7876 ADAMS 23-MAY-87 13

**두 구문 모두 동일하게 원하는 결과를 얻었습니다.**

**[문제12] 임시 테이블 생성하기**

(1) delete ROWS 방식과 preserve ROWS 방식의 COMMIT 이후 차이 확인

SQL> create global temporary table ex1

2 (no NUMBER,

3 name VARCHAR2(10))

4 ON COMMIT delete ROWS;

Table created.

SQL> insert into ex1 VALUES(1,'AAAAA');

1 row created.

SQL> select \* from ex1;

NO NAME

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

1 AAAAA

SQL> COMMIT;

Commit complete.

SQL> select \* from ex1;

no rows selected

SQL> create global temporary table ex2

2 (no NUMBER,

3 name VARCHAR2(10))

4 ON COMMIT preserve ROWS;

Table created.

SQL> insert into ex2 values(1,'AAAAA');

1 row created.

SQL> select \* from ex2;

NO NAME

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

1 AAAAA

SQL> COMMIT;

Commit complete.

SQL> select \* from ex2;

NO NAME

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

1 AAAAA

(2) delete ROWS 적용 시, 서로 다른 세션에서 테이블 조회가 가능한지 확인

SQL> create global temporary table temp001

2 (no NUMBER,

3 name VARCHAR2(10))

4 ON COMMIT delete ROWS;

Table created.

SQL> INSERT INTO temp01 VALUES(1,'AAAA');

1 row created.

SQL> INSERT INTO temp001 VALUES(1,'AAAA');

1 row created.

SQL> select \* from temp001;

NO NAME

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

1 AAAA

텍스트이(가) 표시된 사진

자동 생성된 설명

**delete ROWS의 경우, COMMIT 이후 행이 삭제되었으며, preserve ROWS의 경우 COMMIT 이후에도 행이 유지되었습니다. 또한 서로 다른 세션에서는 임시 테이블은 조회할 수 없었습니다.**

**[문제13] 임시 테이블 조회하기**

SQL> create global temporary table tp01

2 (no NUMBER,

3 name VARCHAR2(10))

4 on commit delete ROWS;

Table created.

SQL> insert into tp01 values(1,'AAAA');

1 row created.

SQL> select \* from tp01;

NO NAME

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

1 AAAA

SQL> select temporary, duration

2 from user\_tables

3 where table\_name = 'TP01';

T DURATION

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

Y SYS$TRANSACTION

**임시로 생성한 테이블을 조회하였습니다.**