

(문제 1) test2.sql 과 test6.sql을 합쳐 pack1 패키지를 작성


SQL>

<명세부>

```
create or replace package pack1 is
  procedure test2
    (v_stu_no in student.stu_no%type,
     v_stu_grade in student.stu_grade%type);
  function test6
    (v_enr_grade in number)
    return char;
end;
```

<몸체부>

```
create or replace package body pack1 is
  procedure test2
    (v_stu_no in student.stu_no%type,
     v_stu_grade in student.stu_grade%type)
  is
  begin
    update student
      set stu_grade = v_stu_grade
      where stu_no = v_stu_no;
  end test2;
  function test6
    (v_enr_grade in number)
    return char
  is
    enr_score char;
  begin
    if v_enr_grade >= 90 then enr_score := 'A';
    elsif v_enr_grade >= 80 then enr_score := 'B';
    elsif v_enr_grade >= 70 then enr_score := 'C';
    elsif v_enr_grade >= 60 then enr_score := 'D';
    else enr_score := 'F';
    end if;
    return (enr_score);
  end test6;
end;
```



PACK1.TEST6(85)
B

(문제 2) 이름으로 학번 검색, 학번으로 이름 검색

SQL>

<명세부>

```
create or replace package pack2 is g_stu_dept varchar2(20);
  procedure test12(v_stu_no in student.stu_no%type);
  procedure test12(v_stu_name student.stu_name%type);
End;
```

<몸체부>

```
create or replace package body pack2 is
  procedure test12(v_stu_no in student.stu_no%type)
  is
    v_stu_name student.stu_name%type;
  begin
    select stu_name into v_stu_name from student
    where stu_no = v_stu_no and stu_dept= g_stu_dept;
    dbms_output.put_line(v_stu_name);
  exception
    when no_data_found then
      dbms_output.put_line('컴퓨터정보과에학생이없습니다.');
```

```
end test12;
  procedure test12(v_stu_name in student.stu_name%type)
  is
    v_stu_nostudent.stu_no%type;
  begin
    select stu_no into v_stu_no from student
    where stu_name = v_stu_nameand stu_dept = g_stu_dept;
    dbms_output.put_line(v_stu_no);
  exception
    when no_data_foundthen
      dbms_output.put_line('컴퓨터정보과에학생이없습니다.');
```

```
end test12;
Begin
  g_stu_dept:= '컴퓨터정보';
End;
```

Line	Position	Message
1		김종현

Line	Position	Message
1		20191001

(문제 3) 학생 테이블에 변경작업이 일어날 때 동작하는 트리거 생성

```
SQL> create or replace trigger tri1
after update on student
begin
    insert into tmp_tbl1
    values(user, sysdate, 'U');
end tri1;
```

```
SQL> update student set stu_weight = stu_weight * 0.9;
```

USERID	WORKDATE	BIGO
A202044011	2023-12-02 오후 3:26:38	U

(문제 4) 행단위 트리거

```
SQL> CREATE OR REPLACE TRIGGER TRI2
after update on student
for each row
begin
    insert into tmp_tbl1 values(user, sysdate, 'U');
end tri2;
SQL> update student
set stu_weight = stu_weight * 0.9;
```

USERID	WORKDATE	BIGO
A202044011	2023-12-02 오후 3:30:45	U
A202044011	2023-12-02 오후 3:30:45	U
A202044011	2023-12-02 오후 3:30:45	U
A202044011	2023-12-02 오후 3:30:45	U
A202044011	2023-12-02 오후 3:30:45	U
A202044011	2023-12-02 오후 3:30:45	U
A202044011	2023-12-02 오후 3:30:45	U
A202044011	2023-12-02 오후 3:30:45	U
A202044011	2023-12-02 오후 3:30:45	U
A202044011	2023-12-02 오후 3:30:45	U
A202044011	2023-12-02 오후 3:30:45	U

(문제 5) 조건에 의한 트리거

```
SQL> create or replace trigger tri3
after update on student
for each row
when (old.stu_weight > 70)
begin
    insertinto tmp_tbl1
    values(user, sysdate, 'U');
end tri3;
```

USERID	WORKDATE	BIGO
A202044011	2023-12-02 오후 3:34:14	U
A202044011	2023-12-02 오후 3:34:14	U
A202044011	2023-12-02 오후 3:34:14	U
A202044011	2023-12-02 오후 3:34:14	U
A202044011	2023-12-02 오후 3:34:14	U
A202044011	2023-12-02 오후 3:34:14	U
A202044011	2023-12-02 오후 3:34:14	U
A202044011	2023-12-02 오후 3:34:14	U
A202044011	2023-12-02 오후 3:34:14	U
A202044011	2023-12-02 오후 3:34:14	U
A202044011	2023-12-02 오후 3:34:14	U
A202044011	2023-12-02 오후 3:34:14	U
A202044011	2023-12-02 오후 3:34:14	U
A202044011	2023-12-02 오후 3:34:14	U
A202044011	2023-12-02 오후 3:34:14	U

(문제 6) LOG 파일을 생성하기 위한 트리거 TRI4

```
SQL> create or replace trigger tri4
after insert or update or delete on enrol
for each row
begin
    if inserting then
        insert into tmp_tbl2(u_id, wdate, n_sub_no, n_stu_no, n_enr_grade, bigo)
        values(user, sysdate, :new.sub_no, :new.stu_no, :new.enr_grade, 'I');
    elsif updating then
        insert into tmp_tbl2(u_id, wdate, o_sub_no, o_stu_no, o_enr_grade,
        n_sub_no, n_stu_no, n_enr_grade, bigo)
        values(user, sysdate, :old.sub_no, :old.stu_no, :old.enr_grade, :new.sub_no,
        :new.stu_no, :new.enr_grade, 'U');
    elsif deleting then
        insert into tmp_tbl2(u_id, wdate, o_sub_no, o_stu_no, o_enr_grade, bigo)
        values(user, sysdate, :old.sub_no, :old.stu_no, :old.enr_grade, 'D');
    end if;
end tri4;
```

SQL> insert into enrol values ('101', '20211062', 40);

U_ID	WDATE	N_SUB_NO	N_STU_NO	N_ENR_GRADE	O_SUB_NO	O_STU_NO	O_ENR_GRADE	BIGO
A202044011	2023-12-02 오후 3:41:04	101	20211062	40				I

SQL> update enrol set enr_grade = 80
where sub_no = '101' and stu_no = '20211062';

U_ID	WDATE	N_SUB_NO	N_STU_NO	N_ENR_GRADE	O_SUB_NO	O_STU_NO	O_ENR_GRADE	BIGO
A202044011	2023-12-02 오후 3:41:04	101	20211062	40				I
A202044011	2023-12-02 오후 3:43:52	101	20211062	80	101	20211062	40	U

SQL>

U_ID	WDATE	N_SUB_NO	N_STU_NO	N_ENR_GRADE	O_SUB_NO	O_STU_NO	O_ENR_GRADE	BIGO
A202044011	2023-12-02 오후 3:41:04	101	20211062	40				I
A202044011	2023-12-02 오후 3:43:52	101	20211062	80	101	20211062	40	U
A202044011	2023-12-02 오후 3:44:55				101	20211062	80	D

(문제 1) 0부터 100까지를 10개의 구간으로 나눈 후 92가 몇 번째 구간에 속하는지 구함

SQL> select width_bucket(92,0,100,10) "score"from dual;

score
10

(문제 2) 전체 데이터 중 순위를 반환하는 함수

SQL> select empno, sal, rank() over(order by sal desc) rank from emp;

EMPNO	SAL	RANK
7839	5000	1
7788	3000	2
7902	3000	2
7566	2975	4
7698	2850	5
7782	2450	6
7499	1600	7
7844	1500	8
7934	1300	9
7521	1250	10
7654	1250	10
7876	1100	12
7900	950	13
7369	800	14

(문제 3) 전체 데이터 중 순위를 반환하는 함수

SQL> select empno, deptno, sal, avg(sal) over(partition by deptno) "deptavg" from emp;

EMPNO	DEPTNO	SAL	deptavg
7934	10	1300	2916.66666666667
7782	10	2450	2916.66666666667
7839	10	5000	2916.66666666667
7369	20	800	2175
7788	20	3000	2175
7902	20	3000	2175
7876	20	1100	2175
7566	20	2975	2175
7900	30	950	1566.66666666667
7844	30	1500	1566.66666666667
7654	30	1250	1566.66666666667
7521	30	1250	1566.66666666667
7698	30	2850	1566.66666666667
7499	30	1600	1566.66666666667

(문제 4) CROSSTAB같은 보고서 만들때 사용

SQL> select stu_dept, grouping(stu_dept), stu_gender, grouping(stu_gender), Count(*) from student group by rollup(stu_dept, stu_gender);

STU_DEPT	GROUPING(STU_DEPT)	STU_GENDER	GROUPING(STU_GENDER)	COUNT(*)
기계	0	F	0	2
기계	0	M	0	1
기계	0		1	3
전기전자	0	F	0	1
전기전자	0	M	0	2
전기전자	0		1	3
컴퓨터정보	0	F	0	1
컴퓨터정보	0	M	0	3
컴퓨터정보	0		1	4
	1		1	10

(문제 5) 조직도와 같은 LEVEL을 나타내는 함수

```
SQL> select LPAD(' ', 3*level - 3)||ename, LEVEL, empno, mgr
from emp start with ename = 'KING'
connect by prior empno = mgr;
```

LPAD(' ', 3*LEVEL-3) ENAME	LEVEL	EMPNO	MGR
KING	1	7839	
JONES	2	7566	7839
SCOTT	3	7788	7566
ADAMS	4	7876	7788
FORD	3	7902	7566
SMITH	4	7369	7902
BLAKE	2	7698	7839
ALLEN	3	7499	7698
WARD	3	7521	7698
MARTIN	3	7654	7698
TURNER	3	7844	7698
JAMES	3	7900	7698
CLARK	2	7782	7839
MILLER	3	7934	7782

(문제 6) 여러 개의 서브쿼리가 하나의 메인쿼리에서 사용될 때 사용

```
SQL> WITH
dept_costs as (select deptno, sum(sal) as dept_total from emp group by deptno),
avg_cost as (select (sum(dept_total)/count(*)) as dept_avg from dept_costs)
select * from dept_costs
where dept_total > (select dept_avg from avg_cost);
```

DEPTNO	DEPT_TOTAL
20	10875

(문제 7) MULTIPLE INSERT

```
SQL> insert all
into stu_gender values(stu_no, stu_name, stu_gender)
into stu_body values(stu_no, stu_name, stu_height, stu_weight)
select * from student;
```

STU_NO	STU_NAME	STU_GENDER
20213075	육한빛	M
20213088	이태연	F
20203054	유가인	F
20212088	조민우	M
20202021	심수정	F
20192003	박희철	M
20211062	김인중	M
20201007	전현무	M
20191001	김종현	M
20191025	육성우	F

STU_NO	STU_NAME	STU_HEIGHT	STU_WEIGHT
20213075	육한빛	177	72
20213088	이태연	162	45
20203054	유가인	154	42.3
20212088	조민우	188	81
20202021	심수정	168	40.5
20192003	박희철		56.7
20211062	김인중	166	60.3
20201007	전현무	174	57.6
20191001	김종현		64.8
20191025	육성우	172	56.7

(문제 8) MULTIPLE INSERT 문

```
SQL> insert all when stu_dept = '컴퓨터정보' then
into stu_gender values(stu_no, stu_name, stu_gender)
when stu_dept = '전기전자' then
into stu_body values(stu_no, stu_name, stu_height, stu_weight)
select * from student;
```

STU_NO	STU_NAME	STU_GENDER
20211062	김민준	M
20201007	전현우	M
20191001	김종현	M
20191025	육성우	F

STU_NO	STU_NAME	STU_HEIGHT	STU_WEIGHT
20212088	조민우	188	81
20202021	심수정	168	40.5
20192003	박희철		56.7

(문제 9) DECODE 문

```
SQL> select deptno, count(DECODE(TO_CHAR(HIREDATE, 'MM'), 01, 1)) "1월",
count(DECODE(TO_CHAR(HIREDATE, 'MM'), 02, 1)) "2월",
count(DECODE(TO_CHAR(HIREDATE, 'MM'), 03, 1)) "3월",
count(DECODE(TO_CHAR(HIREDATE, 'MM'), 04, 1)) "4월",
count(DECODE(TO_CHAR(HIREDATE, 'MM'), 05, 1)) "5월",
count(DECODE(TO_CHAR(HIREDATE, 'MM'), 06, 1)) "6월",
count(DECODE(TO_CHAR(HIREDATE, 'MM'), 07, 1)) "7월",
count(DECODE(TO_CHAR(HIREDATE, 'MM'), 08, 1)) "8월",
count(DECODE(TO_CHAR(HIREDATE, 'MM'), 09, 1)) "9월",
count(DECODE(TO_CHAR(HIREDATE, 'MM'), 10, 1)) "10월",
count(DECODE(TO_CHAR(HIREDATE, 'MM'), 11, 1)) "11월",
count(DECODE(TO_CHAR(HIREDATE, 'MM'), 12, 1)) "12월"
from emp group by deptno order by 1;
```

DEPTNO	1월	2월	3월	4월	5월	6월	7월	8월	9월	10월	11월	12월
10	1	0	0	0	0	1	0	0	0	0	1	0
20	0	0	0	1	0	0	2	0	0	0	0	2
30	0	2	0	0	1	0	0	0	2	0	0	1

(문제 10) DECODE 문

```
SQL> select deptno, decode(count(DECODE(TO_CHAR(HIREDATE, 'MM'), 01, 1)),
'0', 'zero', '1', 'one', '2', 'two') "1월",
decode(count(DECODE(TO_CHAR(HIREDATE, 'MM'), 02, 2)),
'0', 'zero', '1', 'one', '2', 'two') "2월",
count(DECODE(TO_CHAR(HIREDATE, 'MM'), 03, 1)) "3월",
count(DECODE(TO_CHAR(HIREDATE, 'MM'), 04, 1)) "4월",
count(DECODE(TO_CHAR(HIREDATE, 'MM'), 05, 1)) "5월",
count(DECODE(TO_CHAR(HIREDATE, 'MM'), 06, 1)) "6월",
count(DECODE(TO_CHAR(HIREDATE, 'MM'), 07, 1)) "7월",
count(DECODE(TO_CHAR(HIREDATE, 'MM'), 08, 1)) "8월",
```

```

count(DECODE(TO_CHAR(HIREDATE,'MM'), 09, 1)) "9월",
count(DECODE(TO_CHAR(HIREDATE,'MM'), 10, 1)) "10월",
count(DECODE(TO_CHAR(HIREDATE,'MM'), 11, 1)) "11월",
count(DECODE(TO_CHAR(HIREDATE,'MM'), 12, 1)) "12월"
from emp group by deptno order by 1;

```

DEPTNO	1월	2월	3월	4월	5월	6월	7월	8월	9월	10월	11월	12월
10	one	zero	0	0	0	1	0	0	0	0	1	0
20	zero	zero	0	1	0	0	2	0	0	0	0	2
30	zero	two	0	0	1	0	0	0	2	0	0	1

(문제 11) CASE 문

```

SQL> select dept.deptno, dname,
count(case when (TO_CHAR(HIREDATE,'MM')) = '01' then 1 end) "1월" ,
count(case when (TO_CHAR(HIREDATE,'MM')) = '02' then 1 end) "2월" from emp
full outer join dept on emp.deptno = dept.deptno
group by dept.deptno, dname order by 1;

```

DEPTNO	DNAME	1월	2월
10	ACCOUNTING	1	0
20	RESEARCH	0	0
30	SALES	0	2
40	OPERATIONS	0	0

(문제 12) DECODE 문

```

SQL> select d.dname,
count(decode(job, 'CLERK', 1)) as CLERK,
count(decode(job, 'SALESMAN', 1)) as SALESMAN,
count(decode(job, 'MANAGER', 1)) as MANAGER,
count(decode(job, 'ANALYST', 1)) as ANALYST,
count(decode(job, 'PRESIDENT', 1)) as PRESIDENT
from emp e, dept d where e.deptno = d.deptno
group by d.dname order by d.dname;

```

DNAME	CLERK	SALESMAN	MANAGER	ANALYST	PRESIDENT
ACCOUNTING	1	0	1	0	1
RESEARCH	2	0	1	2	0
SALES	1	4	1	0	0

(문제 13) DECODE 문

```

SQL> select * from (select deptno, job, sal from emp)
pivot(avg(sal) for job
in ('CLERK', 'SALESMAN', 'MANAGER', 'ANALYST', 'PRESIDENT'))
order by 1;

```

DEPTNO	'CLERK'	'SALESMAN'	'MANAGER'	'ANALYST'	'PRESIDENT'
10	1300		2450		5000
20	950		2975	3000	
30	950	1400	2850		