**Assignment pl/sql**

Ans1.

a. legal

b. not legal as many variables have a single varchar

c. legal

d. not legal as boolean must take true or false

Ans2.

a. valid and date

b. valid and varchar

c. valid and currency

d. valid and boolean

e. valid and boolean

f. valid and boolean

Ans3.

set serveroutput on;

declare

v\_send varchar2(25);

begin

v\_send :='my pl/sql block works';

dbms\_output.put\_line( v\_send);

end;

Ans 4.

declare

v\_char varchar2(50) ;

v\_num varchar2(50);

begin

v\_char := '42 is the answer';

v\_num := substr(v\_char,1,2);

dbms\_output.put\_line(v\_char);

dbms\_output.put\_line(v\_num);

end;

Ans5.

a. 2, number

b. western europe, varchar

c. 601, number

d. product 10012, varchar

e. western, varchar

Ans6.

a. 201, number

b. unisports, varchar

c. excellent, varchar

d. womansport, varchar

e. null

Ans 7.

set serveroutput on;

define p\_num1 = 2;

define p\_num2 = 4;

declare

var1 number:=&p\_num1;

var2 number:=&p\_num2;

res number(7,2);

begin

res :=(var1/var2)+var2;

dbms\_output.put\_line('result: ' || res);

end;

/

Ans8.

declare

v\_sal number:=50000;

v\_annsal number;

v\_bonus number:=10;

res number;

begin

if v\_sal is null

then

v\_bonus:=0;

end if;

v\_annsal:=12\*v\_sal;

res := v\_annsal+(v\_annsal\*(v\_bonus/100));

dbms\_output.put\_line(res);

end;

Ans9.

declare

g\_max\_deptno number(8);

begin

select max(deptno)

into g\_max\_deptno

from dept;

dbms\_output.put\_line('g\_max\_deptno: ' || g\_max\_deptno);

end;

Ans10.

declare

v\_dname dept.dname%type:= 'education';

v\_deptno dept.deptno%type:=60;

begin

insert into dept (dname,deptno) values (v\_dname,v\_deptno);

dbms\_output.put\_line(v\_deptno);

dbms\_output.put\_line(v\_dname);

end;

Ans11.

define p\_deptno=280

define p\_loc=1700

declare

var1 number:=&p\_deptno ;

var2 number:=&p\_loc ;

deptno number;

loc number;

begin

update dept set loc=var2

where deptno=var1;

dbms\_output.put\_line(deptno || loc);

end;

Ans12.

define p\_deptno=280

define p\_loc=1700

declare

var1 number:=&p\_deptno ;

var2 number:=&p\_loc ;

deptno number;

loc number;

begin

delete from dept

where deptno=var1;

end;

Ans13.

begin

for i in 1..10 loop

if i not in (6,8) then

dbms\_output.put\_line(i);

end if;

end loop;

end;

/

Ans14.

define p\_empno=7839

declare

var1 number:=&p\_empno ;

v\_empno emp.empno%type;

begin

select emp.sal

into v\_empno

from emp

where empno= var1;

if emp.sal<5000

then dbms\_output.put\_line(emp.sal\*0.10);

else if emp.sal>5000 and emp.sal<10000

then dbms\_output.put\_line(emp.sal\*0.15);

else if emp.sal>10000

then dbms\_output.put\_line(emp.sal\*0.20);

else if emp.sal is null

then dbms\_output.put\_line('0');

end if;

end;

Ans15.

create table employees as select \* from emp;

alter table employees add stars varchar2(50);

Ans16.

define p\_empno=7566

declare

v\_asterisk VARCHAR2(25) null;

v\_empno employees.empno%TYPE;

v\_sal employees.sal%TYPE;

var1 NUMBER;

var2 NUMBER:=1000;

BEGIN

select sal

into v\_sal

from employees

where empno=&p\_empno;

var1:= v\_sal/var2;

for i in 1..var1 loop

v\_asterisk:=v\_asterisk || '\*';

end loop;

update employees SET stars=v\_asterisk

where empno=&p\_empno;

END;

/

SELECT empno,sal,stars

FROM employees

where empno IN(7698,7566,7839);

Ans17.

declare

v\_sal number(8,2);

cursor employee

is

select sal

into v\_sal

from emp;

begin

open employee;

loop

fetch employee into v\_sal;

insert into top\_dogs values v\_sal;

exit when employee%notfound;

end loop;

close employee;

end;

Ans18.

a. declare

v\_no number(5);

v\_sal emp.sal%type;

counter number(4):=0;

cursor top\_employee

is

select sal

from emp

order by sal desc;

begin

open top\_employee;

while(counter<&v\_no) loop

fetch top\_employee into v\_sal;

dbms\_output.put\_line(v\_sal);

counter:=counter+1;

end loop;

close top\_employee;

end;

b. declare

v\_no number(5);

v\_sal emp.sal%type;

counter number(4):=0;

cursor top\_employee

is

select distinct(sal)

from emp

order by sal desc;

begin

open top\_employee;

while(counter<&v\_no) loop

fetch top\_employee into v\_sal;

dbms\_output.put\_line(v\_sal);

counter:=counter+1;

end loop;

close top\_employee;

end;

c. declare

v\_no number(5);

v\_sal emp.sal%type;

counter number(4):=0;

cursor top\_employee

is

select distinct(sal)

from emp

order by sal desc;

begin

open top\_employee;

while(counter<&v\_no) loop

fetch top\_employee into v\_sal;

insert into top\_dogs values(v\_sal);

dbms\_output.put\_line(v\_sal);

counter:=counter+1;

end loop;

close top\_employee;

end;

Ans19.

declare

v\_deptno number(5);

v\_ename emp.ename%type;

v\_mgr emp.mgr%type;

v\_sal emp.sal%type;

cursor employee

is

select ename,sal,mgr

from emp

where deptno=&v\_deptno;

begin

open employee;

loop

fetch employee into v\_ename,v\_sal,v\_mgr;

dbms\_output.put\_line(v\_ename||' '||v\_sal||' '||v\_mgr);

exit when employee%notfound;

end loop;

close employee;

end;

c.declare

v\_deptno number(5);

v\_ename emp.ename%type;

v\_mgr emp.mgr%type;

v\_sal emp.sal%type;

cursor employee

is

select ename,sal,mgr

from emp

where deptno=&v\_deptno;

begin

open employee;

loop

fetch employee into v\_ename,v\_sal,v\_mgr;

if v\_sal<5000 and v\_mgr in (101,124) then

dbms\_output.put\_line(v\_ename||' due for a raise.');

else

dbms\_output.put\_line(v\_ename||' not due for a raise.');

end if;

--dbms\_output.put\_line(v\_ename||' '||v\_sal||' '||v\_mgr);

exit when employee%notfound;

end loop;

close employee;

end;

Ans20.

declare

v\_ename emp.ename%type;

v\_sal emp.sal%type;

cursor employee

is

select ename into v\_ename

from emp

where sal=&v\_sal;

begin

open employee;

loop

fetch employee into v\_ename;

exit when employee%notfound;

dbms\_output.put\_line(v\_ename);

end loop;

close employee;

end;

c.

declare

v\_ename emp.ename%type;

v\_sal emp.sal%type;

begin

select ename into v\_ename

from emp

where sal=&v\_sal;

dbms\_output.put\_line(v\_ename);

exception

when no\_data\_found then

dbms\_output.put\_line('no employee with a salary of '||v\_sal);

end;

e.

declare

v\_ename emp.ename%type;

v\_sal emp.sal%type;

begin

select ename into v\_ename

from emp

where sal=&v\_sal;

dbms\_output.put\_line(v\_ename);

exception

when no\_data\_found then

dbms\_output.put\_line('no employee with a salary of '||v\_sal);

when others then

dbms\_output.put\_line ('some other exception occurred');

end;

Ans21.

declare

v\_deptno dept.deptno%type;

v\_loc dept.loc%type;

begin

select loc into v\_loc

from dept

where deptno=&v\_deptno;

dbms\_output.put\_line(v\_loc);

exception

when no\_data\_found then

dbms\_output.put\_line('no department with a deptid of '||v\_deptno);

--when others then

--dbms\_output.put\_line ('some other exception occurred');

end;

c.

declare

v\_deptno dept.deptno%type;

v\_loc dept.loc%type;

v\_ref dept.deptno%type;

begin

v\_ref := &v\_deptno;

select loc, deptno into v\_loc, v\_ref

from dept

where deptno=v\_ref;

dbms\_output.put\_line(v\_loc);

exception

when no\_data\_found then

dbms\_output.put\_line('no department with a deptid of '||v\_ref);

when others then

dbms\_output.put\_line ('department'||v\_ref||' is an invalid department');

end;

Ans22.

declare

v1\_sal emp.sal%type;

v\_sal number(5);

counter number(5);

no\_emp exception;

begin

v\_sal:=&v1\_sal;

select count(empno)into counter

from emp

where sal between (v\_sal-100) and (v\_sal+100);

dbms\_output.put\_line(counter);

if counter=0 then

raise no\_emp;

end if;

exception

when no\_emp then

dbms\_output.put\_line('no employee in given range');

when others then

dbms\_output.put\_line('some error occoured.');

end;

Ans23.

create or replace procedure add\_job

(j\_id in jobs.job\_id%type,title in jobs.job\_title%type) is

begin

insert into jobs values (j\_id,title,'87','13');

end;

execute add\_job('it\_dba','database administrator');

c.

execute add\_job('st\_man','stock manager');

Ans24.

create or replace procedure upd\_job

(jobid jobs.job\_id%type,

title jobs.job\_title%type)

is

counter jobs.max\_salary%type := 0;

id\_not\_found exception;

cursor c1 is

select job\_id

from jobs;

begin

for jobsid in c1 loop

if jobsid.job\_id = jobid then

counter := counter + 1;

end if;

end loop;

if counter = 1 then

update jobs

set

job\_title = title

where

job\_id = jobid;

else

raise id\_not\_found;

end if;

exception

when id\_not\_found then

dbms\_output.put\_line('job id not in table');

end;

Ans26.

create or replace procedure query\_emp

( eid in emp.empno%type,

esal out emp.sal%type,

ejob out emp.job%type)

is

begin

select sal,job

into esal,ejob

from emp

where empno=eid;

dbms\_output.put\_line(esal||' '||ejob);

end query\_emp;

b.

declare

id emp.empno%type:=7499;

sala emp.sal%type;

jobt emp.job%type;

begin

query\_emp(id,sala,jobt);

end;

c.

no data found

Ans27.

create or replace procedure upd\_job

(jobid jobs.job\_id%type,

title jobs.job\_title%type)

is

counter jobs.max\_salary%type := 0;

id\_not\_found exception;

cursor c1 is

select job\_id

from jobs;

begin

for jobsid in c1 loop

if jobsid.job\_id = jobid then

counter := counter + 1;

end if;

end loop;

if counter = 1 then

update jobs

set

job\_title = title

where

job\_id = jobid;

else

raise id\_not\_found;

end if;

exception

when id\_not\_found then

dbms\_output.put\_line('job id not in table');

end;

Ans28.

create or replace package body emp\_pack as

procedure new\_emp is

vale boolean;

invalid\_deptid exception;

begin

vale :=valid\_deptid(15);

if vale=false then

raise invalid\_deptid;

else

dbms\_output.put\_line('true');

insert into emp values('arpit','sa\_rep',7316,sysdate,4000,145,30);

dbms\_output.put\_line('true');

end if;

exception

when invalid\_deptid then

dbms\_output.put\_line('invalid department id');

when others then

dbms\_output.put\_line('some other errors');

end;

function valid\_deptid(dept emp.deptno%type)

return boolean

is

deptno emp.deptno%type;

begin

select count(\*)

into deptno

from emp

where deptno=dept;

if

deptno =0 then

return false;

else

return true;

end if;

end;

end emp\_pack;

create or replace package emp\_pack as

procedure new\_emp;

function valid\_deptid(dept emo.deptno%type) return boolean;

end emp\_pack;

Ans29.

create or replace package body chk\_pack as

procedure chk\_hiredate(p\_date varchar2) as

custom\_null\_error exception;

type datechar is table of data;

countval integer(5,2);

datechars datechar;

retval varchar2(50);

begin

retval :=test\_date(p\_date);

if retval ='false' then

raise custom\_null\_error;

end if;

select hire\_date bulk collect into datechars from emp where p\_date between (sysdate-18000) and (sysdate+90) group by hire\_date;

select count(\*) into countval from emp where p\_date between (sysdate-90) and (sysdate+90);

for i in 1..countval

loop

dbms\_output.put\_line('work maadi '||countval||' '||datechars(i));

end loop;

exception

when custom\_null\_error then

dbms\_output.put\_line('either null data or incorrect date format has been passed,please supply input date in dd-mon-yy format');

when others then

dbms\_output.put\_line('date format is not correct');

end;

function test\_date(p\_date varchar2) return varchar2 is

l\_date date;

begin

if p\_date is null then

return 'false';

end if;

l\_date := to\_date(p\_date,'dd-mon-yy');

return 'true';

exception

when others then

return 'false';

end;

procedure chk\_dept\_mgr(p\_empid emp.empno%type,p\_mgr emp.mgr%type) as

invalid exception;

p\_deptid emp.empno%type;

m\_deptid emp.deptno%type;

title jobs.job\_title%type;

begin

select department\_id into p\_deptid from emp where empno=p\_empid;

select department\_id into m\_deptid from emp where empno=p\_mgr;

if m\_deptid=p\_deptid then

dbms\_output.put\_line('manager id and employee id belong to same department');

select job\_id into title from emp where empno=p\_mgr and job\_id like '%man';

else

raise invalid;

end if;

exception

when invalid then

dbms\_output.put\_line('manager id and employee id doesnt belong to same department');

when no\_data\_found then

dbms\_output.put\_line('manager id doesnt have a manager role');

end;

end chk\_pack;

Ans30.

create or replace procedure secure\_dml

is

begin

if (to\_char(sysdate,'hh24:mi') not between '08:45' and '17:30')

or to\_char (sysdate, 'dy') in ('sat', 'sun')

then

raise\_application\_error (-20500,'you may insert into employees table only during business hours.');

end if;

end secure\_dml;

Ans31.

a. create or replace trigger sec\_job

before insert or update or delete on jobhist

begin

secure\_dml;

end sec\_job;

b. insert into jobhist(empno,job) values(152, 'marketing');