DBMS

PL/SQL

LAB ASSIGNMENT – 3



Ananya Agarwal

102083036

2CO14

1. Write a PL/SQL program to demonstrate following exceptions: • Too Many Rows • No Data Found.

create table try1(rn number(3));

declare

x number;

begin

select rn into x from try1;

dbms\_output.put\_line(x);

exception

when too\_many\_rows then

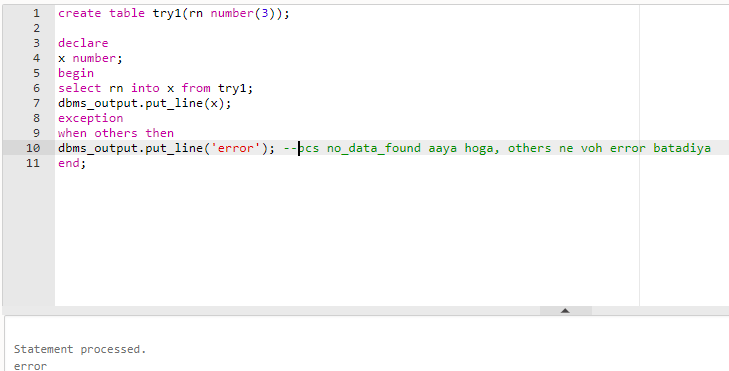
dbms\_output.put\_line('too many rows are there!');

when no\_data\_found then

dbms\_output.put\_line('no data exist');

end;

C:\Users\User\Desktop\3_1.PNG



2. Write a PL/SQL code to display a message to check whether the record is deleted or not.

create table emp(eno number(3) primary key,ename char(20),sal number(20));

insert into emp values(1,'a',2);

insert into emp values(2,'b',3);

insert into emp values(3,'c',2);

begin

delete emp where eno=2;

if sql%found then

dbms\_output.put\_line('Record deleted!');

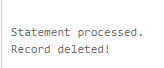
else

dbms\_output.put\_line('Record not deleted!');

end if;

end;

select \* from emp;



3. Write a PL/SQL code to display a message to provide the information about the number of records deleted by the delete statement issued in a PL/SQL block.

create table emp(eno number(3) primary key,ename char(20),sal number(20));

insert into emp values(1,'a',2);

insert into emp values(2,'b',3);

insert into emp values(3,'c',2);

begin

delete emp where eno>=2;

dbms\_output.put\_line(sql%rowcount||' rows have been deleted!');

end;

C:\Users\User\Desktop\3_3.PNG

4. Write a PL/SQL code to display the empno, ename, job of employees of department number 10 for EMP table. EMP (empno, ename, job, sal, deptno).

create table emp(empnno number(3) primary key,ename char(20),job varchar2(20),sal number(6),deptno number(2));

insert into emp values(1,'a','ass prof',20000,1);

insert into emp values(2,'b','ass prof',30000,3);

insert into emp values(3,'c','prof',200000,10);

declare

cursor c1 is select empnno,ename,job from emp where deptno=10;

rec c1%rowtype;

begin

open c1;

loop

fetch c1 into rec;

exit when c1%notfound;

dbms\_output.put\_line('Empno: '||rec.empnno);

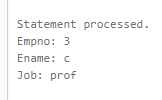
dbms\_output.put\_line('Ename: '||rec.ename);

dbms\_output.put\_line('Job: '||rec.job);

end loop;

close c1;

end;



5.)a.) Write a PL/SQL code to display the employee number and name of top 5 highest paid employees.

CREATE TABLE emp(empno NUMBER(3), ename VARCHAR(15), job VARCHAR(15), sal

NUMBER(8), deptno NUMBER(4));

INSERT INTO emp VALUES(100, 'Anirudh', 'SDE', 1000000, 11);

INSERT INTO emp VALUES(101, 'Parth', 'Cloud Expert', 150000, 10);

INSERT INTO emp VALUES(102, 'Sparsh', 'Data Scientist', 5000000, 10);

INSERT INTO emp VALUES(103, 'Aryan', 'Web Developer', 600000, 11);

INSERT INTO emp VALUES(104, 'Bhagyesh', 'App Developer', 500000, 12);

INSERT INTO emp VALUES(105, 'Mayank', 'AWS Architect', 230000, 12);

DECLARE

CURSOR C1 IS SELECT empno, ename, job FROM emp ORDER BY sal DESC;

data C1%ROWTYPE;

BEGIN

OPEN C1;

LOOP

FETCH C1 INTO data;

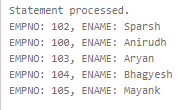
EXIT WHEN C1%ROWCOUNT=6 OR C1%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('EMPNO: '|| data.EMPNO||', ENAME: '||data.ENAME);

END LOOP;

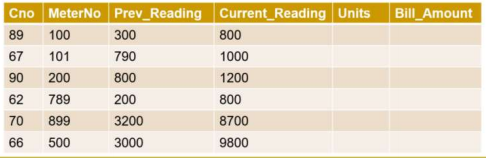
CLOSE C1;

END;



5)b). Let us consider CUST table with columns Cno PK, MeterNo Unique, Prev\_Reading, Current\_Reading, Units, Bill\_Amount Units=Current\_Reading-Prev\_Reading.

• Bill will be charged according to following rate: • First 100 Units 0.5 Rs per unit and after 0.75 Rs for every unit consume.



create table cust(cno number(5), meterno number(5), prev\_reading number(10),

current\_reading number(10), units number(7), bill\_amount number(10));

insert into cust values(89,100,300,800,null,null);

insert into cust values(67,101,790,1000,null,null);

insert into cust values(90,200,800,1200,null,null);

insert into cust values(62,789,200,800,null,null);

insert into cust values(70,899,3200,8700,null,null);

insert into cust values(66,500,3000,9800,null,null);

create or replace procedure bill\_all is

cursor a is select \* from cust;

b number;

u number;

extra number;

begin

for rec in a loop

u:=rec.current\_reading-rec.prev\_reading;

if u<=100 then

b:=u\*0.5;

else

extra:=u-100;

b:=(0.5 \* 100) +extra\*0.75;

end if;

update cust set units=u, bill\_amount=b where cno=rec.cno;

end loop;

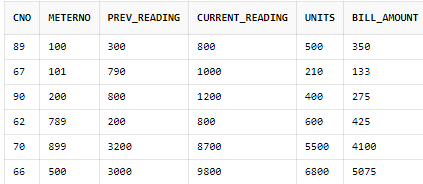
end;

begin

bill\_all;

end;

select \* from cust;



6. Write a PL/SQL code to update units and bill amount of customer number 89 in CUST table.

create table cust(cno number(3), meterno number(5), prev\_reading number(5),

current\_reading number(5), units number(5), bill\_amount number(10));

insert into cust values(89,100,300,800,null,null);

insert into cust values(67,101,790,1000,null,null);

insert into cust values(90,200,800,1200,null,null);

insert into cust values(62,789,200,800,null,null);

insert into cust values(70,899,3200,8700,null,null);

insert into cust values(66,500,3000,9800,null,null);

create or replace procedure get\_bill(c\_no in number) is

data cust%rowtype;

begin

select \* into data from cust where cno=c\_no;

update cust

set units=data.current\_reading-data.prev\_reading,

bill\_amount=units\*10

where cno=data.cno;

end;

declare

cno number;

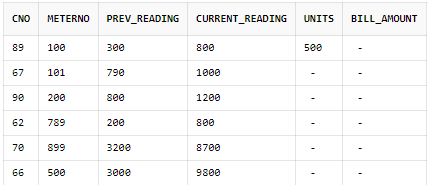
begin

cno := 89;

get\_bill(cno);

end;

select \* from cust;



7. Write a PL/SQL code to update units and bill amount of all the customers in CUST table.

create table cust(cno number(5), meterno number(5), prev\_reading number(10),

current\_reading number(10), units number(7), bill\_amount number(10));

insert into cust values(89,100,300,800,null,null);

insert into cust values(67,101,790,1000,null,null);

insert into cust values(90,200,800,1200,null,null);

insert into cust values(62,789,200,800,null,null);

insert into cust values(70,899,3200,8700,null,null);

insert into cust values(66,500,3000,9800,null,null);

create or replace procedure bill\_all is

cursor a is select \* from cust;

b number;

u number;

extra number;

begin

for rec in a loop

u:=rec.current\_reading-rec.prev\_reading;

if u<=100 then

b:=u\*0.5;

else

extra:=u-100;

b:=(0.5 \* 100) +extra\*0.75;

end if;

update cust set units=u, bill\_amount=b where cno=rec.cno;

end loop;

end;

begin

bill\_all;

end;

select \* from cust;

