# **CS 262 – DBMS Lab (A.Y. 2021 – 2022)**

# **Lab Cycle – 3**

1. Write a PL/SQL block to read a number and check whether it is a palindrome or not.

declare

n number;

m number;

rev number:=0;

r number;

begin

n:=&n;

m:=n;

while n>0

loop

r:=mod(n,10);

rev:=(rev\*10)+r;

n:=trunc(n/10);

end loop;

if m=rev

then

dbms\_output.put\_line('number is palindrome');

else

dbms\_output.put\_line('number is not palindrome');

end if;

end;

/

SQL> @Z:\plsql\palindrome.sql

23 /

Enter value for n: 54695

old 7: n:=&n;

new 7: n:=54695;

number is not palindrome

PL/SQL procedure successfully completed.

SQL> @Z:\plsql\palindrome.sql

23 /

Enter value for n: 12321

old 7: n:=&n;

new 7: n:=12321;

number is palindrome

PL/SQL procedure successfully completed.

SQL> @Z:\plsql\palindrome.sql

Input truncated to 1 characters

Enter value for n: 5445

old 7: n:=&n;

new 7: n:=5445;

number is palindrome

PL/SQL procedure successfully completed.

1. Write a PL/SQL Program to print the first n fibonacci numbers.

declare

first number:=0;

second number:=1;

third number;

n number:=&n;

i number;

begin

dbms\_output.put\_line('Fibonacci series is:');

dbms\_output.put\_line(first);

dbms\_output.put\_line(second);

for i in 2..n

loop

third:=first+second;

first:=second;

second:=third;

dbms\_output.put\_line(third);

end loop;

end;

/

output:

Enter value for n: 4

old 2: n number:=&n;

new 2: n number:=4;

0

1

1

2

3

PL/SQL procedure successfully completed.

1. Write a PL/SQL block which will accept a number and checks if it is a prime number or not.

declare

n number;

i number;

flag number;

begin

i:=2;

flag:=1;

n:=&n;

for i in 2..n/2

loop

if mod(n,i)=0 then

flag:=0;

exit;

end if;

end loop;

if flag=1 then

dbms\_output.put\_line('prime';

else

dbms\_output.put\_line('not prime');

end if;

end;

/

SQL> @C:\Users\y20cs127\Desktop\primenumber.sql

Input truncated to 1 characters

Enter value for n: 27

old 8: n:=&n;

new 8: n:=27;

not prime

PL/SQL procedure successfully completed.

SQL> @C:\Users\y20cs127\Desktop\primenumber.sql

Input truncated to 1 characters

Enter value for n: 3

old 8: n:=&n;

new 8: n:=3;

prime

PL/SQL procedure successfully completed.

1. Write a PL/SQL block which will accept two numbers and find out their LCM and HCF. The output should be stored in a table called **DEMO\_TAB**.

DEMO\_TAB

|  |  |  |  |
| --- | --- | --- | --- |
| Number1 | Number2 | LCM | HCF |

DECLARE

num1 NUMBER;

num2 NUMBER;

t NUMBER;

s NUMBER;

LCM NUMBER;

BEGIN

num1 :=&num1;

num2 :=&num2;

s :=num1\*num2;

WHILE MOD(num2, num1) != 0

LOOP

t := MOD(num2, num1);

num2 := num1;

num1 := t;

END LOOP;

LCM:=s/num1;

dbms\_output.Put\_line('HCF of '||num1||' and '||num2||' is '||num1);

dbms\_output.Put\_line('LCM of '||num1||' and '||num2||' is '||LCM);

insert into DEMO\_TAB values(num1,LCM);

END;

/

SQL> create table DEMO\_TAB(

2 HCF NUMERIC(5,2),

3 LCM NUMERIC(5,2));

Table created.

SQL> @Z:\plsql\HCF.sql

Input truncated to 1 characters

Enter value for num1: 9

old 8: num1 :=&num1;

new 8: num1 :=9;

Enter value for num2: 18

old 9: num2 :=&num2;

new 9: num2 :=18;

HCF of 9 and 18 is 9

LCM of 9 and 18 is 18

PL/SQL procedure successfully completed.

SQL> select \* from DEMO\_TAB;

HCF LCM

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

9 18

1. Write a Pl/SQL program using FOR loop to insert ten rows into a database table.

SQL> create table demo(

2 i number,

3 i2 number,

4 i3 number

5 );

Table created.

SQL> @C:\Users\y20cs127\Desktop\for\_loop.sql

8 /

PL/SQL procedure successfully completed.

SQL> select \* from demo;

I I2 I3

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

1 1 1

2 4 8

3 9 27

4 16 64

5 25 125

6 36 216

7 49 343

8 64 512

9 81 729

10 100 1000

10 rows selected.

1. Consider the following relation schemas

Inventory

|  |  |  |
| --- | --- | --- |
| Product\_ID | Product\_name | Quantity |

Purchase\_Record

|  |  |  |
| --- | --- | --- |
| Product\_ID | Status | Date |

Write a PL/SQL block to read the quantity of a product from inventory and if it is > 0 reduce the quantity by 1 and record the status of purchase of that product as ‘PURCHASED’. Otherwise record the status of purchase of that product as ‘OUT OF STOCK’. While recording the status of a purchase, record the date of transaction.

SQL> create table inventory(product\_id number(3),

2 product\_name varchar(20),

3 quantity number(5));

Table created.

SQL> insert into inventory values(1,'apple',2);

1 row created.

SQL> insert into inventory values(2,'samsung',0);

1 row created.

SQL> insert into inventory values(3,'oneplus',1);

1 row created.

SQL> create table purchase\_Record(product\_id number(3),

2 status varchar(30),

3 dt varchar(10));

Table created.

declare

q inventory.quantity%type;

id inventory.product\_id%type:=&id;

s varchar(30);

cursor data is select quantity from inventory where product\_id=id;

begin

open data;

fetch data into q;

loop

if(q>0) then

update inventory

set quantity=quantity-1

where product\_id=id;

s:='PURCHASED';

insert into purchase\_Record values(id,s,sysdate);

fetch data into q;

else

s:='OUT OF STOCK';

insert into purchase\_Record values(id,s,sysdate);

fetch data into q;

end if;

exit when data%notfound;

end loop;

end;

output:

SQL> @C:\Users\Admin\Desktop\sql\purchase.sql

25 /

Enter value for id: 1

old 3: id inventory.product\_id%type:=&id;

new 3: id inventory.product\_id%type:=1;

PL/SQL procedure successfully completed.

SQL> select \* from inventory;

PRODUCT\_ID PRODUCT\_NAME QUANTITY

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

1 apple 1

2 samsung 0

3 oneplus 1

SQL> select \* from purchase\_Record;

PRODUCT\_ID STATUS DT

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

1 PURCHASED 08-SEP-22

1. Create a table employee with eno, ename, and basic\_pay attributes, insert 3 to 4 records and write a PL/SQL block to calculate the Gross salary & Net salary for an employee for the following conditions:
   * + HRA is 15% of basic.
     + DA is 62% of basic.
     + PF is 780/- if gross salary exceeds 8000, otherwise 600/-.
     + Professional tax is 2% of basic.

and then print the employee no, name, hra, da, pf, ptax, gross salary & net salary for that employee.

SQL> create table employee(eno number(2),

2 ename varchar(20),

3 basicpay number(10));

SQL> insert into employee values(1,'sita',80000);

1 row created.

SQL> insert into employee values(2,'yamuna',1500);

1 row created.

SQL> insert into employee values(3,'satya',1330);

1 row created.

SQL> insert into employee values(4,'keerthy',2000);

1 row created.

declare

name employee.ename%type;

id employee.eno%type;

sal employee.basicpay%type;

hra number(5);

da number(5);

pf number(5);

tax number(5);

gs number(10);

ns number(10);

cursor data is select eno,ename,basicpay from employee;

begin

open data;

loop

fetch data into id,name,sal;

if(data%found) then

hra:=0.15\*sal;

da:=0.62\*sal;

gs:=hra+da+sal;

if(gs>8000)then

pf:=780;

else

pf:=600;

end if;

tax:=0.02\*sal;

ns:=gs-pf-tax;

dbms\_output.put\_line(id||' '||name||' '||hra||' '||da||' '||pf||' '||tax||' '||gs||' '||ns);

end if;

exit when data%notfound;

end loop;

end;

output:

SQL> @C:\Users\Admin\Desktop\sql\employee\_salary.sql

32 /

1 sita 12000 49600 780 1600 141600 139220

2 yamuna 225 930 600 30 2655 2025

3 satya 200 825 600 27 2355 1728

4 keerthy 300 1240 600 40 3540 2900

PL/SQL procedure successfully completed.

1. Consider the following relation schemas

Emp

|  |  |  |  |
| --- | --- | --- | --- |
| empid | name | salary | dno |

Del\_History

|  |  |  |
| --- | --- | --- |
| dno | Rows\_deleted | date |

Write a PL/SQL block to delete records of all employees who belong to a particular department and then record the dno, no of rows deleted and date on which deletion occurred in the Del\_History table using implicit cursors.

SQL> create table emp(empid number(3),

2 name varchar(20),

3 salary number(10)

4 dno number(3));

Table created.

SQL> insert into emp values(1,'sita',2000000,1);

1 row created.

SQL> insert into emp values(2,'ram',15000,2);

1 row created.

SQL> insert into emp values(3,'nani',12000,2);

1 row created.

SQL> create table del\_History(dno number(3),

2 Rowsdeleted number(3),

3 dt varchar(10));

Table created.

declare

dept number(3):=&department;

cursor data is select empid,dno from emp;

id emp.empid%type;

d emp.dno%type;

i number(2):=0;

begin

open data;

loop

fetch data into id,d;

if(data%found) then

if(d=dept) then

delete from emp where empid=id;

i:=i+1;

end if;

end if;

exit when data%notfound;

end loop;

insert into del\_History values(d,i,sysdate);

end;

output:

SQL> @C:\Users\Admin\Desktop\sql\emp\_delete.sql

22 /

Enter value for department: 2

old 2: dept number(3):=&department;

new 2: dept number(3):=2;

PL/SQL procedure successfully completed.

SQL> select \* from del\_History;

DNO ROWSDELETED DT

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

2 2 26-AUG-22

1. Given the table EMPLOYEE (EmpNo, Name, Salary, Designation, DeptID) write a cursor to select the five highest and lowest paid employees from the table.

SQL> create table employee(

2 empno number,

3 name varchar(20),

4 salary number,

5 designation varchar(10),

6 deptID number

7 );

Table created.

SQL> insert into employee values(7788,'scott',3670,'analyst',10);

1 row created.

SQL> insert into employee values(7839,'king',6050,'president',10);

1 row created.

SQL> insert into employee values(7902,'ford',3670,'analyst',10)

1 row created.

SQL> insert into employee values(7903,'miller',7000,'president',20);

1 row created.

SQL> insert into employee values(7904,'smith',8000,'professor',20);

1 row created.

SQL> insert into employee values(7905,'raja',8630,'clerk',30);

1 row created.

SQL> insert into employee values(7906,'rajesh',13000,'salesman',30);

1 row created.

SQL> insert into employee values(7907,'raju',11000,'president',30);

1 row created.

SQL> insert into employee values(7908,'allu',11050,'clerk',20);

1 row created.

SQL> insert into employee values(7909,'kyli',5040,'creater',10);

1 row created.

SQL> insert into employee values(7910,'james',11000,'clerk',20);

1 row created.

SQL> insert into employee values(1122,'sai',12500,'salesman',30);

1 row created.

SQL> insert into employee values(6789,'sam',4500,'clerk',30);

1 row created.

SQL> select \* from employee;

EMPNO NAME SALARY DESIGNATIO DEPTID

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

7788 scott 3670 analyst 10

7839 king 6050 president 10

7902 ford 3670 analyst 10

7903 miller 7000 president 20

7904 smith 8000 professor 20

7905 raja 8630 clerk 30

7906 rajesh 13000 salesman 30

7907 raju 11000 president 30

7908 allu 11050 clerk 20

7909 kyli 5040 creater 10

7910 james 11000 clerk 20

EMPNO NAME SALARY DESIGNATIO DEPTID

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

1122 sai 12500 salesman 30

6789 sam 4500 clerk 30

13 rows selected.

SQL> select \* from employee order by salary desc;

EMPNO NAME SALARY DESIGNATIO DEPTID

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

7906 rajesh 13000 salesman 30

1122 sai 12500 salesman 30

7908 allu 11050 clerk 20

7907 raju 11000 president 30

7910 james 11000 clerk 20

7905 raja 8630 clerk 30

7904 smith 8000 professor 20

7903 miller 7000 president 20

7839 king 6050 president 10

7909 kyli 5040 creater 10

6789 sam 4500 clerk 30

EMPNO NAME SALARY DESIGNATIO DEPTID

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

7788 scott 3670 analyst 10

7902 ford 3670 analyst 10

13 rows selected.

declare

cursor c1 is select name,salary from employee order by salary desc;

cursor c2 is select name,salary from employee order by salary;

v\_name varchar(20);

v\_sal number;

begin

open c1;

dbms\_output.put\_line('top 5 paid salaries are:');

loop

fetch c1 into v\_name,v\_sal;

exit when c1%rowcount>5;

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

end loop;

open c2;

dbms\_output.put\_line('Least 5 paid salaries are: ');

loop

fetch c2 into v\_name,v\_sal;

exit when c2%rowcount>5;

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

end loop;

end;

SQL> @C:\Users\y20cs127\Desktop\employee.sql

25 /

top 5 paid salaries are:

rajesh 13000

sai 12500

allu 11050

raju 11000

james 11000

Least 5 paid salaries are:

scott 3670

ford 3670

sam 4500

kyli 5040

king 6050

PL/SQL procedure successfully completed.

1. Consider the following relation schemas

Bank\_Main

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Acc\_no | Name | Address | Acc-type | Curr\_balance |

Bank\_Trans

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Acc\_no | Tr\_type | Tr\_date | Tr\_amt | Upd\_flag |

Where acc\_type is S – savings C- current

Tr\_type is D – deposit W – withdrawal

Write a PL/SQL block to update master table’s (i.e., Bank\_Main) curr\_balance field with transaction details from transaction file (i.e., Bank\_Trans) and also change the status of the Upd\_flag field of transaction table to ‘Y’ once the updation is complete.

SQL> create table bank\_main(

2 account\_no number,

3 name varchar(20),

4 address varchar(20),

5 acc\_type varchar(10),

6 current\_balance number);

Table created.

SQL> insert into bank\_main values(1001,'Ramu','Guntur','S',15000);

1 row created.

SQL> insert into bank\_main values(1002,'bindu','vizag','s',10000);

1 row created.

SQL> insert into bank\_main values(1003,'abhilash','hyderabad','c',10000);

1 row created.

SQL> insert into bank\_main values(1004,'babu','vijayawada','c',9000);

1 row created.

SQL> create table bank\_transaction(

2 account\_no number,

3 tr\_type varchar(4),

4 tr\_date number,

5 tr\_amount number,

6 upd\_flag varchar(3));

Table created.

SQL> insert into bank\_transaction values(1001,'d',020922,5000,'n');

1 row created.

SQL> insert into bank\_transaction values(1002,'w',020922,2000,'y');

1 row created.

SQL> insert into bank\_transaction values(1003,'d',020922,4000,'y');

1 row created.

SQL> insert into bank\_transaction values(1004,'w',020922,10000,'n');

1 row created.

SQL> select \* from bank\_main;

ACCOUNT\_NO NAME ADDRESS ACC\_TYPE CURRENT\_BALANCE

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

1001 Ramu Guntur S 15000

1002 bindu vizag s 10000

1003 abhilash hyderabad c 10000

1004 babu vijayawada c 9000

SQL> select \* from bank\_transaction;

ACCOUNT\_NO TR\_T TR\_DATE TR\_AMOUNT UPD

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

1001 d 20922 5000 n

1002 w 20922 2000 y

1003 d 20922 4000 y

1004 w 20922 10000 n

1. Write a PL/SQL block to handle the following built-in exceptions

***no\_data\_found***

***too\_many\_rows***

***zero\_divide***

1. Write a PL/SQL block to check whether the quantity of any product in Inventory table is < 0. If so, using an exception display relevant message and update quantity to 0.

Inventory

|  |  |  |
| --- | --- | --- |
| Product\_ID | Product\_name | Quantity |