Psql to find power of number

declare

power number:=2;

num1 number:=3;

res number:=1;

begin

while power!=0 loop

res :=res\*num1;

power:=power-1;

end loop;

DBMS\_OUTPUT.PUT\_LINE('RESult='||res);

end;

prime number in psql

declare

start1:=1;

end1:=100;

count1:=0;

begin

for i in start1..end1 loop

for j in 2..start1 loop

if mod(start1,2)==0 then

count1:=count1+1;

exit when count1=1;

end loop;

if count1=0;

DBMS\_OUTPUT.PUT\_LINE(i);

end if;

end loop;

arrays in plsql

DECLARE

TYPE department IS TABLE OF NUMBER

INDEX BY VARCHAR2(64);

Department\_name department;

i VARCHAR2(64);

BEGIN

Department\_name('electrical'):=5000;

Department\_name('computer'):=70000;

Department\_name('mechanical;'):=9000;

i:=department\_name.first;

loop

DBMS\_OUTPUT.PUT\_LINE(department\_name(i));

i:=department\_name.next(i);

exit when i is null;

end loop;

end;

bulk collect

declare

type emp\_salaries is table of number;

e\_sal emp\_salaries;

begin

select salary bulk collect into e\_sal from hr.employees;

for i in e\_sal.first..e\_sal.last

loop

DBMS\_OUTPUT.PUT\_LINE (e\_sal(i));

end loop;

end;

VARRAY

declare

type e\_salaries is VARRAY(5) OF INTEGER;

e\_sal e\_salaries;

begin

e\_sal:=e\_salaries(1000,2000,3000,4000);

for i in e\_sal.first..e\_sal.last

loop

DBMS\_OUTPUT.PUT\_LINE (e\_sal(i));

end loop;

end;

max salary per departments using nested table array

DECLARE

type emp\_salaries is table of number;

e\_sal emp\_salaries;

BEGIN

select max(salary) bulk collect into e\_sal from hr.employees group by department\_id ;

for i in e\_sal.first..e\_sal.last

loop

Dbms\_output.put\_line(e\_sal(i));

end loop;

end;

DECLARE

MSG VARCHAR(100);

BEGIN

MSG:=PRINT\_HELLO();

DBMS\_OUTPUT.PUT\_LINE(MSG);

END;

create or replace function findavg(dept\_id number)return number

as

avgsal number;

begin

select avg(salary) into avgsal from hr.employees where salary =( SELECT max(salary) froM hr.departments );

return avgsal;

end findavg;

declare

avgsal number;

begin

avgsal:=findavg(90);

dbms\_output.put\_line(avgsal);

end;

procedure

create or replace procedure incsal AS

type salariesarray is varray(500) of integer;

salarray salariesarray;

avgsal integer;

temp integer;

newsal number;

BEGIN

select avg(salary) into avgsal from hr.employees;

select salary bulk collect into salarray from hr.employees where salary>avgsal;

for i in salarray.first..salarray.last

loop

temp:=0;

Dbms\_output.put\_line(salarray(i));

temp:=salarray(i)\*salarray(i)\*0.03;

Dbms\_output.put\_line('new salary:'||temp);

end loop;

end;

--invoke procedure

begin

incsal();

end;

cash back

create or replace procedure cashback as

price integer:=5000;

cashb integer;

accid integer:=3;

begin

cashb:=price\*0.01;

update account\_details set balance=balance-price where account\_id=accid;

dbms\_output.putline('balance debited');

update account\_details set balance=balance+cashb where account\_id=accid;

dbms\_output.putline('cash back added');

end cashback;

cursor

declare cursor c1 is select last\_name,job\_id from hr.employees;

begin

for item in c1

loop

dbms\_output.put\_line('name='|| item.last\_name||',job='||item.job\_id);

end loop;

end;

declare

v\_jobid hr.employees.job\_id%type;

v\_lastname hr.employees.last\_name%type;

cursor c1 is select last\_name,job\_id from hr.employees;

begin

open c1;

loop

fetch c1 into v\_lastname,v\_jobid;

exit when c1%notfound;

dbms\_output.put\_line(v\_lastname||','||v\_jobid);

end loop;

close c1;

end;

TRIGGER

CREATE OR REPLACE TRIGGER BALANCE\_TRIGGER\_ACCOUNT\_DETAILS

BEFORE INSERT OR UPDATE ON ACCOUNT\_DETAILS

FOR EACH ROW

BEGIN

IF UPDATING THEN

IF:NEW.ACC\_BALANCE<1000 THEN

RAISE\_APPLICATION\_ERROR(-20100,'REACHED MINIMM BALANCE');

END IF;

END IF;

# END;