-- las variables

set serveroutput on

declare

indetificador integer:= 50;

nombre varchar2(25):= 'jose';

apodo char(10):='joselo';

sueldo number(5):= 30000;

comision decimal(4,2):= 50.20;

fecha date:=to\_date('2020/07/09','YYYY/MM/DD');

fecha\_atual date :=(sysdate);

mensaje CONSTANT varchar2(50):='te escribo ahora';

--pi CONSTANT decimal(1,2):=3.14;

saludo varchar2(50) default 'buenos dias a todos';

begin

dbms\_output.put\_line('su indentificador es: '||indetificador);

dbms\_output.put\_line('su nombre es: '||nombre);

dbms\_output.put\_line('su apodo es: '||apodo);

dbms\_output.put\_line('su sueldo es: '||sueldo);

dbms\_output.put\_line('su comision es: '||comision);

dbms\_output.put\_line('su fecha es: '||fecha);

dbms\_output.put\_line('su fecha actual es: '||fecha\_atual);

dbms\_output.put\_line('su saludo es: '||saludo);

dbms\_output.put\_line('su mensaje es: '||mensaje);

end;

-- condicionales if

declare

a number(2):=40;

b number(2):=30;

begin

IF a<b THEN

dbms\_output.put\_line(a||' es mayor que '|| b);

ELSE

dbms\_output.put\_line(a||'es mayor que'|| b);

END IF;

end;

-- otros ejemplo de condicionale if

set serveroutput on

declare

numero number (3):=100;

begin

if(numero = 10) then

dbms\_output.put\_line('valor es 10');

elsif (numero = 20) then

dbms\_output.put\_line('valor es 20');

elsif(numero = 30) then

dbms\_output.put\_line('valor de numero 30');

else

dbms\_output.put\_line('ninguno de los valores fue encontrado');

end if;

DBMS\_OUTPUT.PUT\_LINE('el valor exacto de la variable es; '|| numero);

end;

-- bucle tipo de bucle

-- loop bucle basico

declare

valor number:= 10;

begin

loop

DBMS\_OUTPUT.PUT\_LINE(valor);

valor := valor +10;

if valor > 50 then

exit;

end if;

end loop;

DBMS\_OUTPUT.PUT\_LINE('valor final' ||valor);

end;

-- los string

set serveroutput on

declare

nombre varchar2(20):= 'jose perez';

direccion varchar2(30):= 'calle primera n1';

detalles clob:='este detalle de la variable clob que iniciamos en la seccion declarativa es una variables de gran almacenamiento';

eleccion char(1):='y';

begin

if eleccion = 'y' then

DBMS\_OUTPUT.PUT\_LINE(nombre);

DBMS\_OUTPUT.PUT\_LINE(direccion);

DBMS\_OUTPUT.PUT\_LINE(detalles);

end if;

end;

--funciones para variable minuscula y mayuscula

declare

saludo varchar2(12):= 'hola a todos';

begin

DBMS\_OUTPUT.PUT\_LINE(upper(saludo));

DBMS\_OUTPUT.PUT\_LINE(lower(saludo));

DBMS\_OUTPUT.PUT\_LINE(initcap(saludo));

DBMS\_OUTPUT.PUT\_LINE(substr(saludo,2));

DBMS\_OUTPUT.PUT\_LINE(instr(saludo,'t'));

end;

-- quitar caracter

declare

saludo varchar2(30):= '#### hola a todos ####';

begin

DBMS\_OUTPUT.PUT\_LINE(Rtrim(saludo,'#'));

DBMS\_OUTPUT.PUT\_LINE(Ltrim(saludo,'#'));

DBMS\_OUTPUT.PUT\_LINE(trim('#'from saludo));

end;

-- while

declare

valor number(3):=10;

begin

while valor<50 loop

DBMS\_OUTPUT.PUT\_LINE(valor);

valor:= valor + 10;

end loop;

end;

-- bucle while tabla de multiplicar

declare

valor number(3):=1;

begin

while valor<6 loop

DBMS\_OUTPUT.PUT\_LINE('5 x '||valor||' = '|| 5\*valor);

valor:= valor + 1;

end loop;

end;

-- bucle for

set serveroutput on

begin

for a in 1..6 loop

DBMS\_OUTPUT.PUT\_LINE(a);

end loop;

end;

--tabla de multiplicar por for

set serveroutput on

begin

for a in 1..6 loop

DBMS\_OUTPUT.PUT\_LINE('5 x '||a ||' = '|| 5\*a);

end loop;

end;

-- funciones de case

create or replace function semana(a number) return varchar2

is

dia varchar2(250);

begin

dia := ' ';

case a

when 1 then dia:='hoy es lunes';

when 2 then dia:='hoy es martes';

when 3 then dia:='hoy es miercoles';

when 4 then dia:='hoy es jueves';

when 5 then dia:='hoy es viernes';

when 6 then dia:='hoy es sabado';

when 7 then dia:='hoy es domingo';

else dia :='este dia no existe';

end case;

return dia;

end;

select semana(1) as como from dual;

create or replace function probar(a char) RETURN varchar2

is

opction varchar2(70);

begin

opction:= '';

case a

WHEN '+' then opction:='es una suma';

when '-' then opction:='es una reta';

when '\*' then opction :='es una multiplicacion';

when '/' then opction:='es una divicion';

else opction:='no es ninguna de las opciones';

end case;

RETURN opction;

end;

select probar('/') AS OPERADOR FROM DUAL;

select\*from empleado;

SELECT\*FROM control;

INSERT into empleado values(3,'juan','martin');

create or REPLACE trigger insertempleado

before insert on empleado

for each row

begin

insert into control VALUES(user,sysdate);

end;

create table controles(

usuario varchar2(70),

fecha date,

name varchar(50))

select\*from empleados;

-- los diparadore de varias funciones update,delete y insert

create or replace trigger para\_empleados

before insert or update or delete

on empleados

for each row

begin

if inserting then

INSERT into controles values(user,sysdate,'ingreso');

end if;

if updating then

INSERT into controles values(user,sysdate,'actualizar');

end if;

if deleting then

INSERT into controles values(user,sysdate,'eliminando');

end if;

end para\_empleados;

create table libros(

codigo number(6),

titulo varchar2(40),

autor varchar2(30),

auditoria varchar2(30),

precio number(6,2));

create table ofertas(

codigo number(6),

precio number(6,2),

usuario varchar2(20),

fecha date);

insert into libros values(100,'Uno','Richard Bach','Planeta',25);

insert into libros values(103,'El aleph','Borges','Emece',28);

insert into libros values(105,'Matematica estas ahi','Paenza','Nuevo siglo',12);

insert into libros values(120,'Aprenda PHP','Molina Mario','Nuevo siglo',55);

insert into libros values(145,'Alicia en el pais de las maravillas','Carroll','Planeta',35);

select \* from libros;

select \* from ofertas;

create or replace trigger ingresar\_libros

before insert

on libros

for each row

begin

if(:new.precio <= 30) then

insert into ofertas values(:new.codigo, :new.precio, user, sysdate);

end if;

end ingresar\_libros;

insert into libros values(21,'chapiro','gaskin','planeta',29);

create or replace trigger disparador\_empleados

before insert or update or delete on empleados

for each row

begin

if(:new.sueldo > 5000) then

raise\_aplication\_error(-20000, 'sueldo no puede los $5000.00');

end if;

insert into controles values(user,sysdate,'inserctado');

if(:old.seccion = 'gerente') then raise\_aplication\_error(-20000,'no se puede eliminar el puesto de gerencia');

end if;

insert into controles values(user,sysdate,'borrado');

if updating('documento') then raise\_application\_error(-20000,'no se puede actualizar numero documentos');

end if;

end disparador\_empleados;

set serveroutput on

declare

c\_codi empleado.codigo%type;

cursor c\_emple is

select\*from empledo where codigo = 3;

begin

open c\_emple;

dbms\_output.put\_line(c\_codi);

close c\_emple;

-- funciones

create or replace FUNCTION eq\_f(a VARCHAR2, b VARCHAR2) RETURN NUMBER AS

BEGIN

IF a = b THEN RETURN 1;

ELSE RETURN 0;

END IF;

END;

set serveroutput on

begin

select algo (1,3) from dual;

create or replace function algo (a number, b number) return varchar2 AS

begin

if a < b then return 'es mayor que a';

else return 'es lo contrario';

end if;

end;

set SERVEROUT ON;

DECLARE

A INTEGER ;

BEGIN

FOR A IN 1..6 LOOP

DBMS\_OUTPUT.PUT\_LINE (A);

END LOOP;

END;

create or replace procedure saludo As

begin

dbms\_output.put\_line('hola saludo');

end;

execute saludo;

declare

a number(2):=40;

b number(2):=30;

begin

IF a<b THEN

dbms\_output.put\_line(a||' es mayor que '|| b);

ELSE

dbms\_output.put\_line(a||'es mayor que'|| b);

END IF;

end;

declare

a number(2):=40;

begin

dbms\_output.put\_line(a||' es mayor que '|| b);

ELSE

dbms\_output.put\_line(a||'es mayor que'|| b);

END IF;

end;

select \* from empleado;

set serveroutput on

declare

a number(2):=1;

-- b number(2):=1;

begin

while a < 10 loop

dbms\_output.put\_line('5 x '||a||' = '||5\*a);

a := a+1;

end loop;

end;

--crear packeke

create or replace package producto\_paquetes as

procedure saludo;

function sumar return number;

end producto\_paquetes;

-- cuerpo de paquete--

create or replace package body producto\_paquetes as

procedure saludo

as begin

null;

end ;

function sumar return number as

begin

return 2+4;

end sumar;

end;

select producto\_paquetes.sumar from dual;

begin

producto\_paquetes.saludo;

end;

-- programacion orientada a objeto

drop type producto;

drop type body producto;

create or replace type productos as object(

--atributo

codigo number,

nombre varchar2(100),

precio number,

-- metodo

member function ver\_producto return varchar2,

member function ver\_precio return number,

member procedure cambiar\_precio (pvp number),

static procedure auditoria,

constructor function productos(n1 varchar) return self as result

);

create or replace type body productos as

member function ver\_producto return varchar2 as

begin

return 'codigo : '||codigo||'nombre: '||nombre||'precio: '||precio;

end ver\_producto;

member function ver\_precio return number as

begin

return precio;

end ver\_precio;

member procedure cambiar\_precio(pvp number)

as

begin

precio:=pvp;

end cambiar\_precio;

static procedure auditoria as

begin

insert into auditoria values(user,sysdate);

end;

constructor function productos(n1 varchar) return self as result

is

begin

self.nombre:= n1;

self.precio:= null;

self.codigo:= SQ1.nextval;

end;

end;

set serveroutput on

declare

valor1 productos;

begin

valor1:= productos(100,'manzana',10);

dbms\_output.put\_line(valor1.ver\_precio());

dbms\_output.put\_line(valor1.ver\_producto());

valor1.cambiar\_precio(20);

dbms\_output.put\_line(valor1.ver\_precio());

end;

create sequence sq1;

create table auditoria(

usuario varchar2(100),

fecha date);

select\*from auditoria

set serveroutput on

declare

c\_emp empleado%rowtype;

begin

for c\_emp in(select\*from empleado)loop

dbms\_output.put\_line(c\_emp.nombre);

end loop;

end;

--cursor rowtype

set serveroutput on

declare

c\_emp empleado%rowtype;

begin

select \* into c\_emp from empleado where codigo =3;

dbms\_output.put\_line(c\_emp.nombre);

end;

set serveroutput on

declare

emp empleado%rowtype;

begin

select\*into emp from empleado where codigo=2;

DBMS\_OUTPUT.PUT\_LINE(emp.nombre);

end;

create or replace function sums return number is

c number(2);

begin

select count(\*) into c from empleado;

return c;

end ;

select sums from dual

drop table casa;

create table casa(

numero int,

nombre varchar2(30),

datos varchar2(2000)

constraint c1 check (datos is json));

insert into casa VALUES(33,'EJEMPLO','{

"PAIS":"R.D",

"CUIDAD":"STO.DGO",

"POBLACION":"45000"

}');

SELECT\*FROM CASA:

select c.datos.pais from casa c;

select\*from all\_tables;

set serveroutput on

declare

filas number(2);

begin

update empleados

set sueldo = sueldo + 100

where sueldo >= 4000;

if sql%notfound then

dbms\_output.put\_line('no hay ninguna fila para actualizar');

elsif sql%found then

filas := sql%rowcount;

dbms\_output.put\_line(filas||' fueron actualizada');

end if;

end;

select\*from empleados

select\*from empleado

set serveroutput on

declare

emp\_cod empleado.codigo%type;

emp\_nom empleado.nombre%type;

emp\_ape empleado.apellido%type;

cursor cu\_empleado is

select codigo,nombre,apellido

from empleado

where codigo = 2;

begin

open cu\_empleado;

FETCH cu\_empleado INTO emp\_cod,emp\_nom,emp\_ape;

dbms\_output.put\_line(emp\_cod||' '||emp\_nom||' '||emp\_ape);

close cu\_empleado;

end;

set serveroutput on

declare

emp empleado%rowtype;

begin

for emp in (select\*from empleado)loop

dbms\_output.put\_line(emp.nombre);

end loop;

end;

set serveroutput on

declare

begin

update empleado

set nombre = 'juan'

where codigo = 1;

if sql%notfound then

dbms\_output.put\_line('no existe ');

end if;

end;

create or replace trigger gatos

before or delete on empleado

for each row

begin

inserte into control values (user,sysdate)

end;

set SERVEROUT ON;

DECLARE

A INTEGER ;

BEGIN

FOR A IN 1..6 LOOP

DBMS\_OUTPUT.PUT\_LINE (A);

END LOOP;

END;

create or replace procedure saludo As

begin

dbms\_output.put\_line('hola saludo');

end;

execute saludo;

declare

a number(2):=40;

b number(2):=30;

begin

IF a<b THEN

dbms\_output.put\_line(a||' es mayor que '|| b);

ELSE

dbms\_output.put\_line(a||'es mayor que'|| b);

END IF;

end;

declare

a number(2):=40;

begin

dbms\_output.put\_line(a||' es mayor que '|| b);

ELSE

dbms\_output.put\_line(a||'es mayor que'|| b);

END IF;

end;

select \* from empleado;

set serveroutput on

declare

a number(2):=1;

-- b number(2):=1;

begin

while a < 10 loop

dbms\_output.put\_line('5 x '||a||' = '||5\*a);

a := a+1;

end loop;

end;

--crear packeke

create or replace package producto\_paquetes as

procedure saludo;

function sumar return number;

end producto\_paquetes;

-- cuerpo de paquete--

create or replace package body producto\_paquetes as

procedure saludo

as begin

null;

end ;

function sumar return number as

begin

return 2+4;

end sumar;

end;

select producto\_paquetes.sumar from dual;

begin

producto\_paquetes.saludo;

end;

-- programacion orientada a objeto

drop type producto;

drop type body producto;

create or replace type productos as object(

--atributo

codigo number,

nombre varchar2(100),

precio number,

-- metodo

member function ver\_producto return varchar2,

member function ver\_precio return number,

member procedure cambiar\_precio (pvp number),

static procedure auditoria,

constructor function productos(n1 varchar) return self as result

);

create or replace type body productos as

member function ver\_producto return varchar2 as

begin

return 'codigo : '||codigo||'nombre: '||nombre||'precio: '||precio;

end ver\_producto;

member function ver\_precio return number as

begin

return precio;

end ver\_precio;

member procedure cambiar\_precio(pvp number)

as

begin

precio:=pvp;

end cambiar\_precio;

static procedure auditoria as

begin

insert into auditoria values(user,sysdate);

end;

constructor function productos(n1 varchar) return self as result

is

begin

self.nombre:= n1;

self.precio:= null;

self.codigo:= SQ1.nextval;

end;

end;

set serveroutput on

declare

valor1 productos;

begin

valor1:= productos(100,'manzana',10);

dbms\_output.put\_line(valor1.ver\_precio());

dbms\_output.put\_line(valor1.ver\_producto());

valor1.cambiar\_precio(20);

dbms\_output.put\_line(valor1.ver\_precio());

end;

create sequence sq1;

create table auditoria(

usuario varchar2(100),

fecha date);

select\*from auditoria

set serveroutput on

declare

c\_emp empleado%rowtype;

begin

for c\_emp in(select\*from empleado)loop

dbms\_output.put\_line(c\_emp.nombre);

end loop;

end;

--cursor rowtype

set serveroutput on

declare

c\_emp empleado%rowtype;

begin

select \* into c\_emp from empleado where codigo =3;

dbms\_output.put\_line(c\_emp.nombre);

end;

set serveroutput on

declare

emp empleado%rowtype;

begin

select\*into emp from empleado where codigo=2;

DBMS\_OUTPUT.PUT\_LINE(emp.nombre);

end;

create or replace function sums return number is

c number(2);

begin

select count(\*) into c from empleado;

return c;

end ;

select sums from dual

drop table casa;

create table casa(

numero int,

nombre varchar2(30),

datos varchar2(2000)

constraint c1 check (datos is json));

insert into casa VALUES(33,'EJEMPLO','{

"PAIS":"R.D",

"CUIDAD":"STO.DGO",

"POBLACION":"45000"

}');

SELECT\*FROM CASA:

select c.datos.pais from casa c;

select\*from all\_tables;

set serveroutput on

declare

filas number(2);

begin

update empleados

set sueldo = sueldo + 100

where sueldo >= 4000;

if sql%notfound then

dbms\_output.put\_line('no hay ninguna fila para actualizar');

elsif sql%found then

filas := sql%rowcount;

dbms\_output.put\_line(filas||' fueron actualizada');

end if;

end;

select\*from empleados

select\*from empleado

set serveroutput on

declare

emp\_cod empleado.codigo%type;

emp\_nom empleado.nombre%type;

emp\_ape empleado.apellido%type;

cursor cu\_empleado is

select codigo,nombre,apellido

from empleado

where codigo = 2;

begin

open cu\_empleado;

FETCH cu\_empleado INTO emp\_cod,emp\_nom,emp\_ape;

dbms\_output.put\_line(emp\_cod||' '||emp\_nom||' '||emp\_ape);

close cu\_empleado;

end;

set serveroutput on

declare

emp empleado%rowtype;

begin

for emp in (select\*from empleado)loop

dbms\_output.put\_line(emp.nombre);

end loop;

end;

set serveroutput on

declare

begin

update empleado

set nombre = 'juan'

where codigo = 1;

if sql%notfound then

dbms\_output.put\_line('no existe ');

end if;

end;

create or replace trigger gatos

before or delete on empleado

for each row

begin

inserte into control values (user,sysdate)

end;

--variables compuesta con for

set serverout on;

declare

v\_empleados empleados%rowtype;

begin

for v\_empleados in (select\*from empleados) loop

dbms\_output.put\_line(v\_empleados.nombre || ' '||v\_empleados.sueldo);

end loop;

end;

--para actualizar

set serverout on;

begin

update empleados set sueldo = 2500 where documento = '3333';

if sql%notfound then

dbms\_output.put\_line('no existe para modificar');

end if;

end;

-- cursores con parametro

set serveroutput on

declare

nom empleado.nombre%type;

ape empleado.apellido%type;

cursor c\_empleado (cod empleado.codigo%type)

is

select nombre,apellido from empleado where codigo = cod;

begin

open c\_empleado(2);

loop

fetch c\_empleado into nom,ape;

exit when c\_empleado%notfound;

dbms\_output.put\_line(nom||' '||ape);

end loop;

close c\_empleado;

end;

--pratica he ejercicio

create table agenda(

apellido varchar2(30),

nombre varchar2(20),

domicilio varchar2(30),

telefono varchar2(11));

insert into agenda(apellido,nombre,domicilio,telefono)values('arias','jose','calle numero tres','8099996059');

select \* from agenda where nombre = 'jose';

update agenda set nombre ='juan' where apellido = 'arias';

delete agenda where apellido = 'arias';

create or replace PACKAGE juan as

procedure actualiza\_emp(a VARCHAR2, b number);

FUNCTION restar(a number,b number) return number;

end;

--paquetes

create or replace PACKAGE body juan as

procedure actualiza\_emp(a VARCHAR2, b number)as

begin

update empleado set nombre = a where codigo = b;

end;

function restar(a number,b number) return number is

c number(3);

begin

c:=a-b;

return c;

end;

end;

BEGIN

juan.actualiza\_emp('pablo',3);

end;

execute procedure juan.actualiza\_emp('francis',2);

select\*from empleados

set serveroutput on

declare

filas number(3);

begin

update empleados set sueldo = sueldo-500 where sueldo >= 4000;

if sql%notfound then

DBMS\_OUTPUT.PUT\_LINE('no existe la cantidad');

elsif sql%found then

filas:= sql%rowcount;

DBMS\_OUTPUT.PUT\_LINE(filas);

end if;

end;

set serveroutput on

declare

create or replace trigger jose

before delete on empleado

for each row

begin

INSERT INTO control VALUES (user,sysdate);

end;