



**Universidad Nacional Autónoma de México**

**Facultad de Ingeniería**

**Laboratorio de Bases de Datos**

**Práctica 12**

**PROGRAMACIÓN CON SQL PARTE 1**

**Semestre 2019-1**

**Integrantes:**

**Valderrama Navarro Armando**

## Introducción

En el mundo de las bases de datos es de vital importancia la velocidad y la integridad del manejo de los datos, si bien un buen diseño de base de datos es de vital importancia eso es solo un punto a cumplir dentro de este mundo, pues la tarea de los datos no termina en almacenarlos de manera correcta, sino también en la forma de consultarlos y como hacerlo.

PL/SQL es un lenguaje de programación que ayuda a la eficiencia de ciertas tareas que con un lenguaje de programación externo podrían consumir mas tiempo, durante el desarrollo de esta práctica veremos algunos ejemplos básicos del uso de este lenguaje de programación.

## Objetivos

Poner en práctica los conceptos de programación PL/SQL para crear bloques anónimos, procedimientos, disparadores (triggers) así como funciones creadas por el usuario.

### Práctica complementaria.

#### C1.

--@Autor: Armando Vaderrama

--@Fecha Creación: 24/11/2018

--@Descripción: Script encargado de validar e insertar pagos

--Habilita la salida de mensajes dbms\_output.put\_line  
set serveroutput on

create or replace procedure p\_corrige\_pagos is

```
    v_importe number;  
    v_fecha_pago date;  
    v_numpago number;  
    v_total number;  
    v_numreg number;  
    v_pago_faltante number;  
    v_indice number;
```

begin

```
    v_numreg := 0;  
    for v_indice in 1..100 loop  
        select count(*) into v_pago_faltante  
        from pago_auto  
        where auto_id = v_indice  
        and num_pago = 1;  
        if v_pago_faltante = 0 then  
            v_fecha_pago := sysdate;  
            v_importe := 2500;  
            v_numpago := 1;  
            insert into pago_auto(num_pago, auto_id, fecha_pago, importe)  
            values(v_numpago, v_indice, v_fecha_pago, v_importe);
```

```

        v_numreg := v_numreg + 1;
    end if;
end loop;

v_total := v_numreg * 2500;

dbms_output.put_line('Numero de pagos faltantes: '|| v_numreg);
dbms_output.put_line('Importe total: '|| v_total);

end;
/
show errors

```

```

AVN_P1201_AUT05-SQL> exec p_corrige_pagos
Numero de pagos faltantes: 57
Importe total: 142500
Procedimiento PL/SQL terminado correctamente.

```

## C2.

--@Autor: Armando Valderrama  
 --@Fecha Creación: 24/11/2018  
 --@Descripción: Script encargado de registrar un nuevo auto.

set serveroutput on

```

create or replace procedure p_crea_auto (
    p_auto_id in out number,
    p_marca in varchar2,
    p_modelo in varchar2,
    p_anio in varchar2,
    p_num_serie in varchar2,
    p_tipo in char,
    p_precio in NUMERIC,
    p_agencia_id in NUMERIC,
    p_num_cilindros in number,
    p_num_pasajeros in number,
    p_clase in char,
    p_peso_maximo in number,
    p_volumen in number,
    p_tipo_combustible in char)
is
    v_foto auto.foto%TYPE;
    v_fecha_status auto.fecha_status%TYPE;
    v_status_auto_id auto.status_auto_id%TYPE;
    v_cliente_id auto.cliente_id%TYPE;
    v_historico_status historico_status_auto.historico_status_id%TYPE;
    v_fecha_status_historico historico_status_auto.fecha_status%TYPE;
begin
    select auto_seq.nextval into p_auto_id from dual;

```

```

v_cliente_id:= NULL;
v_status_auto_id:= 2;
v_fecha_status:= SYSDATE;
v_foto:= empty_blob();
v_fecha_status_historico:= SYSDATE;

insert into auto (auto_id,marca,modelo,anio,num_serie,tipo,precio,descuento,
                  foto,fecha_status,status_auto_id,agencia_id,cliente_id)
values(p_auto_id,p_marca,p_modelo, p_anio,p_num_serie,p_tipo,p_precio,p_agencia_id,
      v_foto,v_fecha_status,v_status_auto_id,p_agencia_id,v_cliente_id);

if p_tipo = 'P' then
    insert into auto_particular values(p_auto_id,p_num_cilindros,p_num_pasajeros,p_clase);
elsif p_tipo ='C' then
    insert into auto_carga
values(p_auto_id,p_peso_maximo,p_volumen,p_tipo_combustible);
else
    raise_application_error(-20010,'Error tipo de auto.');
```

end if;

```

select historico_status_auto_seq.nextval INTO v_historico_status from dual;
insert into historico_status_auto
      values(v_historico_status,v_fecha_status_historico,v_status_auto_id,p_auto_id);
end p_crea_auto;
/
show errors
```

### C3.

```

--@Autor: Armando Valderrama
--@Fecha Creación: 14/11/2018
--@Descripción: Script encargado de definir un trigger.
```

```
set serveroutput on
```

```

create or replace trigger tr_valida_status_auto
before insert or update of status_auto_id on auto
for each row
```

```

declare
v_id number;
v_status_id number;
v_status_id_a number;
v_pago_total number;
v_precio number;
v_num_pago number;
v_status_id number;
```

```
begin
```

```

select num_pago,status_auto_id into v_num_pago,v_status_id
from auto a,pago_auto pa
where a.auto_id = pa.auto_id
and pa.auto_id = v_id;

select sum(importe) as pago_auto, a.precio into v_pago_total, v_precio
from auto a,pago_auto pa
where a.auto_id = pa.auto_id
and pa.auto_id = v_id
group by pa.auto_id,a.precio;

if v_status_id = 4 and (v_pago_total <> v_precio) then
    raise_application_error(-20003,'Aun no se cubre el pago total');
elsif v_status_id = 3 then
    if v_num_pago <> 1 then
        raise_application_error(-20002,'Debe existir pago regitrado');
    else
        :new.descuento = 0.5*v_importe_pago;
    end if;
elsif v_status_id = 6 and (v_status_id_a <> 4) then
    raise_application_error(-20003,'El auto no ha sido vendido');
elsif v_status_id in (1,2,5) and (v_num_pago > 0 ) then
    raise_application_error(-20001,'Un auto con el estatus ingresado no puede contar con
pagos registrados.');
```

end if;

```

end tr_valida_status_auto;
/
show errors;
```

## C5.

--@Autor: Armando Valderrama

--@Fecha Creación: 14/11/2018

--@Descripción: Script encargado de generar un cadena con formato.

set serveroutput on

```

create or replace function exporta_datos_auto_csv_fx(id_auto numeric)
return varchar2
```

is

```

v_cadena varchar2(32000);
v_auto_id auto.auto_id%TYPE;
v_num_serie auto.num_serie%TYPE;
v_tipo auto.tipo%TYPE;
v_precio auto.precio%TYPE;
v_num_cilindros auto_particular.num_cilindros%TYPE;
v_peso_maximo auto_carga.peso_maximo%TYPE;
v_num_pago pago_auto.num_pago%TYPE;
v_importe pago_auto.importe%TYPE;
```

```

v_email cliente.email%TYPE;

begin
    select a.auto_id, a.num_serie, a.tipo, a.precio, pa.num_pago, pa.importe, c.email,
ac.peso_maximo, ap.num_cilindros
        into v_auto_id, v_num_serie,v_tipo, v_precio,v_num_pago, v_importe,v_email,
v_peso_maximo, v_num_cilindros
    from auto a, cliente c, pago_auto pa, auto_carga ac, auto_particular ap
    where a.auto_id = id_auto
    and a.cliente_id=c.cliente_id
    and a.auto_id = pa.auto_id
    and ac.auto_id = a.auto_id(+)
    and ap.auto_id = a.auto_id(+);

    v_cadena:= v_auto_id||','||v_num_serie||','||v_tipo||','|| v_precio||v_num_cilindros||
v_peso_maximo||','||
    v_num_pago||','||','||v_importe||','||v_email;
    return v_cadena;
end;
/
show errors

```

## Conclusiones

Durante el desarrollo de esta practica se pusieron en practica muchos de los temas vistos en teoría que están relacionados con el tema de PL/SQL y tras haber realizado los ejercicios de esta practica, podemos concluir que si bien PL/SQL conlleva cierto grado de dificultad para aprenderlo, lo mas importante es tener claras las bases del tema 9, pues en PL/SQL realmente solo se almacenan datos de consultas para poder hacer algo con esos datos, pero el éxito o fracaso de las funciones, procedimientos, etc, dependen unicamente de que las consultas sean las correctas.

## Bibliografía

Ing. Jorge A. Rodríguez Campos. Tema 10 INTRODUCCIÓN A LA PROGRAMACIÓN PL/SQL.