

UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES
LICENCIATURA EN INGENIERÍA DE SISTEMAS DE INFORMACIÓN
SISTEMAS DE BASES DE DATOS II

LABORATORIO No. 6
FUNDAMENTOS DEL LENGUAJE-CURSORES-PROCEDIMIENTOS Y
FUNCIONES.
APLICAR PROGRAMACION ALMACENADA DE BASE DE DATOS-PL/SQL
ORACLE.

Integrantes:

ANDREINA GÓMEZ 8-939-1682

STEFANIE AROSEMENA 8-885-1747

EMANOL GONZALEZ 3-745-1637

JOSÉ QUINTERO 8-952-698

DOCENTE: ING. HENRY J. LEZCANO P.

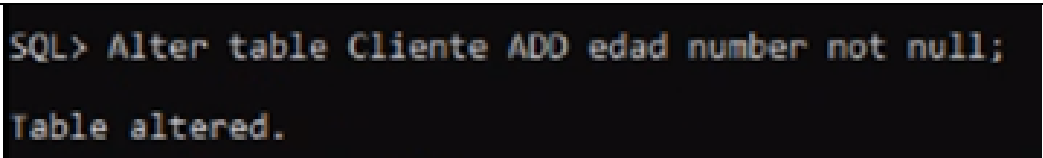
GRUPO: 1IF121

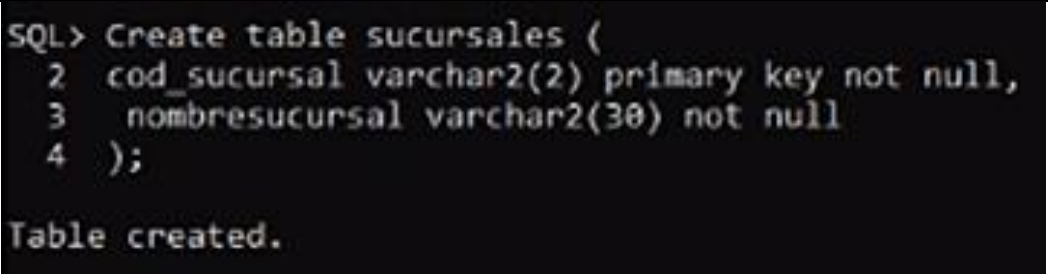
FECHA: 19 OCTUBRE.

AÑO 2020.

Tomando en cuenta lo proporcionado por la compañía Financiera Márquez-Cedeño, S.A. del Laboratorio No.5.

‘Una Compañía financiera necesita levantar un modelo físico de datos a partir de un modelo lógico relacional para una sección de su proceso de negocio (otorgamiento de préstamos), en lo corresponde a la entrada de la información de los clientes y los préstamos e información relacionada a esta sección del proceso a dicha base de datos’

1.	Alter table Cliente Add edad number not null; 
----	--

2.	Create table Sucursales(cod_sucursal varchar2(2) primary key not null, nombresucursal varchar(30) not null); 
----	---

	<p>Create table sucursal_Tprestamo(cd_sucursal varchar2(2) not null, Constraint fk_sucursal foreign key (cd_sucursal) references sucursales(cod_sucursal), cd_prestamo varchar2(5) not null, Constraint fk_tprestamo foreign key (cd_prestamo) references Tipo_prestamo (cod_tipoprestamo), Constraint pk_sucursal_tprestamo primary key (cd_sucursal,cd_prestamo), Montoprestamos number not null);</p>
	<pre>SQL> Create table sucursal_Tprestamo(2 cd_sucursal varchar2(2) not null, 3 Constraint fk_sucursal foreign key (cd_sucursal) references sucursales(cod_sucursal), 4 cd_prestamo varchar2(5) not null, 5 Constraint fk_tprestamo foreign key (cd_prestamo) references Tipo_prestamo (cod_tipoprestamo), 6 Constraint pk_sucursal_tprestamo primary key (cd_sucursal,cd_prestamo), 7 Montoprestamos number not null 8); Table created.</pre>

3	<p>ALTER TABLE Cliente ADD (codi_sucursal varchar2(2) not null, constraint fk_codi_sucursal foreign key(codi_sucursal) references sucursales (cod_sucursal));</p>
	<pre>SQL> ALTER TABLE Cliente ADD (2 codi_sucursal varchar2(2) not null, 3 constraint fk_codi_sucursal foreign key(codi_sucursal) references sucursales (cod_sucursal) 4); Table altered.</pre>

	<p>ALTER TABLE Prestamo ADD (codig_sucursal varchar2(2) not null, constraint fk_codig_sucursal foreign key(codig_sucursal) references sucursales (cod_sucursal));</p>
	<pre>SQL> ALTER TABLE Prestamo ADD (2 codig_sucursal varchar2(2) not null, 3 constraint fk_codig_sucursal foreign key(codig_sucursal) references sucursales (cod_sucursal) 4); Table altered.</pre>

4.	<p>ALTER TABLE Prestamo ADD (Saldoactual number, interespagado number, fechamodificacion date not null , ide_usuario varchar2(5) not null, constraint prestamo_usuario_fk foreign key(ide_usuario) references usuarios(id_usuario));</p>
	<pre>SQL> alter table prestamo 2 add(saldo_actual number not null, 3 interes_pagado number (15,2), 4 fecha_modificacion date not null, 5 usuario varchar2(30) not null); Table altered.</pre>

5.	<p>Create table Transacpagos (</p> <p>id_transaccion varchar2(10) primary key not null,</p> <p>cod_sucursal varchar2(2) not null,</p> <p>id_cliente varchar2(10) not null,</p> <p>tipoprestamo varchar2(5),</p> <p>fechatransaccion date not null,</p> <p>monto_pago number ,</p> <p>fechainsercion date not null,</p> <p>iden_usuario varchar2(5) not null,</p> <p>constraint iden_usuario foreign key(iden_usuario) references usuarios(id_usuario),</p> <p>constraint fk_c_sucursal foreign key(cod_sucursal) references sucursales(cod_sucursal),</p> <p>constraint fk_c_cliente foreign key(id_cliente) references cliente(id_cliente),</p> <p>constraint fk_c_transaccion foreign key(tipoprestamo) references Tipo_prestamo(Cod_tipoprestamo),</p> <p>constraint uk_transaccion unique(id_cliente, tipoprestamo)</p> <p>);</p>
	<pre> SQL> Create table Transacpagos (2 id_transaccion varchar2(10) primary key not null, 3 cod_sucursal varchar2(2) not null, 4 id_cliente varchar2(10) not null, 5 tipoprestamo varchar2(5), 6 fechatransaccion date not null, 7 monto_pago number , 8 fechainsercion date not null, 9 iden_usuario varchar2(5) not null, 10 constraint iden_usuario foreign key(iden_usuario) references usuarios(id_usuario), 11 constraint fk_c_sucursal foreign key(cod_sucursal) references sucursales(cod_sucursal), 12 constraint fk_c_cliente foreign key(id_cliente) references cliente(id_cliente), 13 constraint fk_c_transaccion foreign key(tipoprestamo) references Tipo_prestamo(Cod_tipoprestamo), 14 constraint uk_transaccion unique(id_cliente, tipoprestamo) 15); Table created. </pre>

Procedimiento almacenado para la carga o inserción de las tablas paramétricas

- **Tipo de Teléfonos**

```
CREATE OR REPLACE PROCEDURE crearTipoTelefono (  
  p_codtelefono    Tipo_de_telefono.cod_telefono%TYPE,  
  p_descripciontel Tipo_de_telefono.descripcion_tel%TYPE)  
AS  
BEGIN  
  INSERT INTO Tipo_de_telefono VALUES ( p_codtelefono, p_descripciontel );  
  EXCEPTION WHEN DUP_VAL_ON_INDEX THEN  
    DBMS_OUTPUT.PUT_LINE('NO INGRESO VALORES VALIDOS PARA  
ESTA TABLA....');  
  COMMIT;  
END crearTipoTelefono;
```

```
SQL> CREATE OR REPLACE PROCEDURE crearTipoTelefono (  
  2  p_codtelefono    Tipo_de_telefono.cod_telefono %TYPE,  
  3  p_descripciontel  Tipo_de_telefono.descripcion_tel %TYPE)  
  4  AS  
  5  BEGIN  
  6    INSERT INTO Tipo_de_telefono VALUES ( p_codtelefono, p_descripciontel );  
  7  EXCEPTION WHEN DUP_VAL_ON_INDEX THEN  
  8    DBMS_OUTPUT.PUT_LINE('NO INGRESO VALORES VALIDOS PARA ESTA TABLA...');  
  9  COMMIT;  
 10  END crearTipoTelefono;  
 11  /  
  
Procedure created.
```

- Tipo de Correos

```
CREATE OR REPLACE PROCEDURE crearTipoCorreo (  
    p_codcorreo    Tipo_de_correo.cod_correo%TYPE,  
    p_descripcioncorreo    Tipo_de_correo.descripcion_correo%TYPE) AS  
BEGIN  
    INSERT INTO Tipo_de_correo VALUES ( p_codcorreo, p_descripcioncorreo);  
EXCEPTION WHEN DUP_VAL_ON_INDEX THEN  
    DBMS_OUTPUT.PUT_LINE('NO INGRESO VALORES VALIDOS PARA  
ESTA TABLA....');  
COMMIT;  
END crearTipoCorreo;
```

```
SQL> CREATE OR REPLACE PROCEDURE crearTipoCorreo (  
2   p_codcorreo    Tipo_de_correo.cod_correo %TYPE,  
3   p_descripcioncorreo    Tipo_de_correo.descripcion_correo %TYPE) AS  
4 BEGIN  
5   INSERT INTO Tipo_de_correo VALUES ( p_codcorreo, p_descripcioncorreo);  
6 EXCEPTION WHEN DUP_VAL_ON_INDEX THEN  
7   DBMS_OUTPUT.PUT_LINE('NO INGRESO VALORES VALIDOS PARA ESTA TABLA..');  
8 COMMIT;  
9 END crearTipoCorreo;  
10 /  
Procedure created.
```

- Profesiones

```
CREATE OR REPLACE PROCEDURE crearProfesion (  
  p_codprofesion    Tipo_profesion.cod_profesion%TYPE,  
  p_descripcion     Tipo_profesion.profesion%TYPE) AS  
BEGIN  
  INSERT INTO Tipo_profesion VALUES ( p_codprofesion, p_descripcion);  
EXCEPTION WHEN DUP_VAL_ON_INDEX THEN  
  DBMS_OUTPUT.PUT_LINE('NO INGRESO VALORES VALIDOS PARA  
ESTA TABLA....');  
COMMIT;  
END crearProfesion;
```

```
SQL> CREATE OR REPLACE PROCEDURE crearProfesion (  
2   p_codprofesion    Tipo_profesion.cod_profesion%TYPE,  
3   p_descripcion     Tipo_profesion.profesion%TYPE) AS  
4 BEGIN  
5   INSERT INTO Tipo_profesion VALUES ( p_codprofesion, p_descripcion);  
6 EXCEPTION WHEN DUP_VAL_ON_INDEX THEN  
7   DBMS_OUTPUT.PUT_LINE('NO INGRESO VALORES VALIDOS PARA ESTA TABLA...');  
8 COMMIT;  
9 END crearProfesion;  
10 /  
  
Procedure created.
```

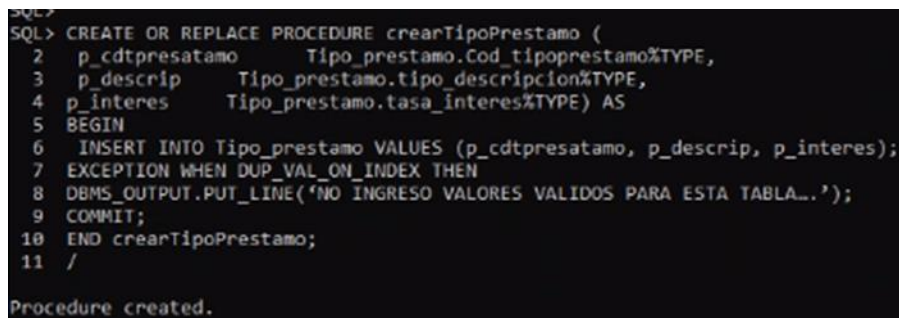

- Sucursales,

```
CREATE OR REPLACE PROCEDURE crearSucursal (  
  p_codsucursal   sucursales.cod_sucursal%TYPE,  
  p_nombre       sucursales.nombresucursal%TYPE) AS  
BEGIN  
  INSERT INTO sucursales VALUES ( p_codsucursal, p_nombre);  
EXCEPTION WHEN DUP_VAL_ON_INDEX THEN  
  DBMS_OUTPUT.PUT_LINE('NO INGRESO VALORES VALIDOS PARA  
ESTA TABLA....');  
COMMIT;  
END crearSucursal;
```

```
SQL>  
SQL> CREATE OR REPLACE PROCEDURE crearSucursal (  
2   p_codsucursal   sucursales.cod_sucursal%TYPE,  
3   p_nombre       sucursales.nombresucursal%TYPE) AS  
4 BEGIN  
5   INSERT INTO sucursales VALUES ( p_codsucursal, p_nombre);  
6 EXCEPTION WHEN DUP_VAL_ON_INDEX THEN  
7   DBMS_OUTPUT.PUT_LINE('NO INGRESO VALORES VALIDOS PARA ESTA TABLA...');  
8 COMMIT;  
9 END crearSucursal;  
10 /  
  
Procedure created.
```

- **Tipos de prestamo**

```
CREATE OR REPLACE PROCEDURE crearTipoPrestamo (  
  p_cdtpresatamo Tipo_prestamo.Cod_tipoprestamo%TYPE,  
  p_descrip Tipo_prestamo.tipo_descripcion%TYPE,  
  p_interes Tipo_prestamo.tasa_interes%TYPE) AS  
BEGIN  
  INSERT INTO Tipo_prestamo VALUES (p_cdtpresatamo, p_descrip, p_interes);  
EXCEPTION WHEN DUP_VAL_ON_INDEX THEN  
  DBMS_OUTPUT.PUT_LINE('NO INGRESO VALORES VALIDOS PARA  
ESTA TABLA....');  
COMMIT;  
END crearTipoPrestamo;  
/
```



```
SQL> CREATE OR REPLACE PROCEDURE crearTipoPrestamo (  
2  p_cdtpresatamo Tipo_prestamo.Cod_tipoprestamo%TYPE,  
3  p_descrip Tipo_prestamo.tipo_descripcion%TYPE,  
4  p_interes Tipo_prestamo.tasa_interes%TYPE) AS  
5  BEGIN  
6  INSERT INTO Tipo_prestamo VALUES (p_cdtpresatamo, p_descrip, p_interes);  
7  EXCEPTION WHEN DUP_VAL_ON_INDEX THEN  
8  DBMS_OUTPUT.PUT_LINE('NO INGRESO VALORES VALIDOS PARA ESTA TABLA...');  
9  COMMIT;  
10 END crearTipoPrestamo;  
11 /  
  
Procedure created.
```

- **Usuario**

```
CREATE OR REPLACE PROCEDURE crearUsuario(  
  p_idUsuario usuarios.id_usuario%TYPE,  
  p_nombreUsuario usuarios.nombre_usuario%TYPE)  
AS  
BEGIN  
  INSERT INTO usuarios VALUES (p_idUsuario, p_nombreUsuario);  
EXCEPTION WHEN DUP_VAL_ON_INDEX THEN  
  DBMS_OUTPUT.PUT_LINE('NO INGRESO VALORES VALIDOS PARA  
ESTA TABLA....');  
COMMIT;  
END crearUsuario;  
/
```

Procedimiento almacenado para la carga o inserción de los clientes con toda la información correspondiente. Este procedimiento debe invocar una Función que calcule la edad de los clientes.

FUNCIÓN PARA SACAR EDAD	
<pre>CREATE OR REPLACE FUNCTION sacarEdad (p_fecfin IN DATE) RETURN NUMBER IS v_anos NUMBER := 0; BEGIN v_anos := FLOOR(MONTHS_BETWEEN(TO_DATE(sysdate) ,p_fecfin) / 12); RETURN v_anos; EXCEPTION WHEN OTHERS THEN RETURN 0; END sacarEdad;</pre>	
<pre>SQL> CREATE OR REPLACE FUNCTION sacarEdad (p_fecfin IN DATE) 2 RETURN NUMBER 3 IS 4 v_anos NUMBER := 0; 5 BEGIN 6 v_anos := FLOOR(MONTHS_BETWEEN(TO_DATE(sysdate) ,p_fecfin) / 12); 7 RETURN v_anos; 8 EXCEPTION 9 WHEN OTHERS THEN 10 RETURN 0; 11 END sacarEdad; 12 / Function created.</pre>	

Probando función:

```
SQL> SET SERVEROUTPUT ON
SQL> Declare
2 fechanaci date := TO_DATE('03/07/2000','DD/MM/YYYY');
3 edad number;
4 Begin
5 edad := sacarEdad(Fechanaci);
6 DBMS_OUTPUT.PUT_LINE(edad);
7 End;
8 /
20

PL/SQL procedure successfully completed.
```

Procedimiento para crear cliente

```
CREATE OR REPLACE PROCEDURE crearCliente(  
  p_idCliente Cliente.Id_cliente%TYPE,  
  p_nombreCliente Cliente.nombre%TYPE,  
  p_apellidoCliente Cliente.apellido%TYPE,  
  p_cedulaCliente Cliente.cedula%TYPE,  
  p_sexoCliente Cliente.sexo%TYPE,  
  p_fnacimientoCliente Cliente.fecha_nac%TYPE,  
  p_cdProfesionCliente Cliente.cod_profesion%TYPE,  
  p_cdSucursal Cliente.codi_sucursal%TYPE  
)AS  
  v_edad number:=0;  
BEGIN  
  dbms_output.put_line(p_fnacimientoCliente);  
  v_edad := sacarEdad (p_fnacimientoCliente);  
  dbms_output.put_line(v_edad);  
  Insert into Cliente values(  
    p_idCliente, p_nombreCliente,  
    p_apellidoCliente, p_cedulaCliente,  
    p_sexoCliente, p_fnacimientoCliente,  
    p_cdProfesionCliente, v_edad,  
    p_cdSucursal  
  );  
EXCEPTION WHEN DUP_VAL_ON_INDEX THEN  
  DBMS_OUTPUT.PUT_LINE(' NO PUEDE DUPLICCAR CLIENTES...');  
COMMIT;  
END crearCliente;
```

```

SQL> CREATE OR REPLACE PROCEDURE crearCliente(
  2 p_idCliente Cliente.Id_cliente%TYPE,
  3 p_nombreCliente Cliente.nombre%TYPE,
  4 p_apellidoCliente Cliente.apellido%TYPE,
  5 p_cedulaCliente Cliente.cedula%TYPE,
  6 p_sexoCliente Cliente.sexo%TYPE,
  7 p_fnacimientoCliente Cliente.fecha_nac%TYPE,
  8 p_cdProfesionCliente Cliente.cod_profesion%TYPE,
  9 p_cdSucursal Cliente.cod_sucursal%TYPE
10 )AS
11 v_edad number:=0;
12 BEGIN
13 dbms_output.put_line(p_fnacimientoCliente);
14 v_edad := sacarEdad (p_fnacimientoCliente);
15 dbms_output.put_line(v_edad);
16 /*Insert into Cliente values(
17 p_idCliente, p_nombreCliente,
18 p_apellidoCliente, p_cedulaCliente,
19 p_sexoCliente, p_fnacimientoCliente,
20 p_cdProfesionCliente, v_edad,
21 p_cdSucursal
22 );*/
23 EXCEPTION WHEN DUP_VAL_ON_INDEX THEN
24 DBMS_OUTPUT.PUT_LINE(' NO PUEDE DUPLICAR CLIENTES..');
25 COMMIT;
26 END crearCliente;
27 /

Procedure created.

SQL> SET SERVEROUTPUT ON
SQL> Begin
  2 crearCliente('CL01','Andreina','Gomez','8-939-1022','Femenino', to_date('19/01/1999', 'DD/MM/YYYY'),'P1','S1');
  3 End;
  4 /
19/01/99
21

```

Procedimiento almacenado para la carga o inserción de los préstamos aprobados con toda la información correspondiente e igualmente este proceso deberá actualizar la información de préstamo en la tabla de sucursales.

```
SET SERVEROUTPUT ON
```

```
Begin
```

```
crearTipoPrestamo ('PT1','Personal',0.12);
```

```
End;
```

```
CREATE OR REPLACE PROCEDURE prestamoAprobado(
```

```
p_idCliente Prestamo.id_cliente%TYPE,
```

```
p_cdSucursal Prestamo.codig_sucursal%TYPE,
```

```
p_codTipoPrestamo Prestamo.cod_tipoprestamo%TYPE,
```

```
p_montoPrestamos sucursal_Tprestamo.Montoprestamos%TYPE,
```

```
P_montoAprobado Prestamo.monto_aprobado%TYPE,
```

```
p_fechaAprobado Prestamo.fecha_aprobado%TYPE,
```

```
p_noPrestamo Prestamo.no_prestamo%TYPE,
```

```
P_letraMensual Prestamo.letra_mensual%TYPE,
```

```

p_montoPagado Prestamo.monto_pagado%TYPE,
p_fechaPago Prestamo.fecha_pago%TYPE,
p_saldoActual Prestamo.Saldoactual%TYPE,
p_interesPagado Prestamo.interespagado%TYPE,
p_fechaModificacion Prestamo.fechamodificacion%TYPE,
p_idUsuario usuarios.id_usuario%TYPE
)
AS
BEGIN
Insert into Prestamo values(
p_idCliente, p_codTipoPrestamo, p_noPrestamo, p_fechaAprobado,
P_montoAprobado, P_letraMensual, p_montoPagado, p_fechaPago,
p_cdSucursal, p_saldoActual, p_interesPagado,
p_fechaModificacion,p_idUsuario
);
Insert into sucursal_Tprestamo values(
p_cdSucursal, p_codTipoPrestamo, p_montoPrestamos
);

EXCEPTION WHEN DUP_VAL_ON_INDEX THEN
    DBMS_OUTPUT.PUT_LINE(' NO PUEDE DUPLICCAR CLIENTES...');
COMMIT;
END prestamoAprobado;

Begin
crearUsuario('US01','Angelica Abadia');
End;
Begin
prestamoAprobado('CL01','S1','PT1',1000,1000,to_date('19/10/2020',
'DD/MM/YYYY'),1,10,0, to_date('19/10/2020', 'DD/MM/YYYY'),1000,0,
TO_DATE('19/10/2020 10:30:25', 'DD/MM/YYYY HH:MI:SS'),'US01');

```

End;

/

```
SQL> CREATE OR REPLACE PROCEDURE prestamoAprobado(  
  2  p_idCliente Prestamo.id_cliente%TYPE,  
  3  p_cdSucursal Prestamo.codigo_sucursal%TYPE,  
  4  p_codTipoPrestamo Prestamo.cod_tipoprestamo%TYPE,  
  5  p_montoPrestamos sucursal_Tprestamo.Montoprestamos%TYPE,  
  6  p_montoAprobado Prestamo.monto_aprobado%TYPE,  
  7  p_fechaAprobado Prestamo.fecha_aprobado%TYPE,  
  8  p_noPrestamo Prestamo.no_prestamo%TYPE,  
  9  p_letraMensual Prestamo.letra_mensual%TYPE,  
10  p_montoPagado Prestamo.monto_pagado%TYPE,  
11  p_fechaPago Prestamo.fecha_pago%TYPE,  
12  p_saldoActual Prestamo.Saldoactual%TYPE,  
13  p_interesPagado Prestamo.interespagado%TYPE,  
14  p_fechaModificacion Prestamo.fechamodificacion%TYPE,  
15  p_idUsuario usuarios.id_usuario%TYPE  
16 )  
17 AS  
18 BEGIN  
19  Insert into Prestamo values(  
20  p_idCliente, p_codTipoPrestamo, p_noPrestamo, p_fechaAprobado,  
21  p_montoAprobado, p_letraMensual, p_montoPagado, p_fechaPago,  
22  p_cdSucursal, p_saldoActual, p_interesPagado,  
23  p_fechaMoficacion, p_idUsuario  
24  );  
25  EXCEPTION WHEN DUP_VAL_ON_INDEX THEN  
26  DBMS_OUTPUT.PUT_LINE('NO PUEDE DUPLICAR CLIENTES...');  
27  COMMIT;  
28  END prestamoAprobado;  
29  Begin  
30  crear Usuario('US01', 'Angelica Abadia');  
31  End;  
32  Begin  
33  prestamoAprobado('CL01', 'S1', 'PT1', 1000, 1000, to_date('19/10/2020', 'DD/MM  
YYYY'), 1,10,0, to_date('19/10/2020', 'DD/MM/YYYY'), 1000,0,TO_DATE('19/10/2020 10_  
0_25', 'DD/MM/YYYY HH:MI:SS'), 'US01');  
34  End;  
35  /
```

Procedimiento almacenado para la carga o inserción de los pagos recibidos de los clientes que se almacena en la tabla transpagos. Por lo menos uno para cada tipo de préstamos.

```
CREATE OR REPLACE PROCEDURE crearPago (  
  p_idTransPago Transacpagos.id_transaccion%TYPE,  
  p_codSucursal Transacpagos.cod_sucursal%TYPE,  
  p_idCliente Transacpagos.id_cliente%TYPE,  
  p_tipoPrestamo Transacpagos.tipoprestamo%TYPE,
```

```

p_fechaTans Transacpagos.fechatransaccion%TYPE,
p_montoPago Transacpagos.monto_pago%TYPE,
p_fechaInsercion Transacpagos.fechainsercion%TYPE,
p_idenUsuario Transacpagos.iden_usuario%TYPE
)
AS
BEGIN
Insert into Transacpagos values(
p_idTransPago,p_codSucursal,p_idCliente,p_tipoPrestamo,
p_fechaTans,p_montoPago,p_fechaInsercion, p_idenUsuario);
EXCEPTION WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE(' NO PUEDE DUPLICAR DATOS...');
COMMIT;
END crearPago;
/

```

```

SQL> CREATE OR REPLACE PROCEDURE crearPago (
2  p_idTransPago Transacpagos.id_transaccion%TYPE,
3  p_codSucursal Transacpagos.cod_sucursal%TYPE,
4  p_idCliente Transacpagos.id_cliente%TYPE,
5  p_tipoPrestamo Transacpagos.tipoprestamo%TYPE,
6  p_fechaTans Transacpagos.fechatransaccion%TYPE,
7  p_montoPago Transacpagos.monto_pago%TYPE,
8  p_fechaInsercion Transacpagos.fechainsercion%TYPE,
9  p_idenUsuario Transacpagos.iden_usuario%TYPE
10 )
11 AS
12 BEGIN
13 Insert into Transacpagos values(
14 p_idTransPago,p_codSucursal,p_idCliente,p_tipoPrestamo,
15 p_fechaTans,p_montoPago,p_fechaInsercion, p_idenUsuario);
16 EXCEPTION WHEN DUP_VAL_ON_INDEX THEN
17 DBMS_OUTPUT.PUT_LINE(' NO PUEDE DUPLICAR DATOS...');
18 COMMIT;
19 END crearPago;
20 /

```

Procedure created.