**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’

|  |  |
| --- | --- |
|  | Alter table Cliente  Add edad number not null; |
|  |

|  |  |
| --- | --- |
|  | Create table Sucursales(  cod\_sucursal varchar2(2) primary key not null,  nombresucursal varchar(30) not null); |
|  |
| 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  ); |
|  |

|  |  |
| --- | --- |
|  | ALTER TABLE Cliente ADD (  codi\_sucursal varchar2(2) not null,  constraint fk\_codi\_sucursal foreign key(codi\_sucursal) references sucursales (cod\_sucursal)  ); |
|  |
| ALTER TABLE Prestamo ADD (  codig\_sucursal varchar2(2) not null,  constraint fk\_codig\_sucursal foreign key(codig\_sucursal) references sucursales (cod\_sucursal)  ); |
|  |

|  |  |
| --- | --- |
|  | 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)  ); |
| Texto  Descripción generada automáticamente |

|  |  |
| --- | --- |
| 5. | Create table Transacpagos (  id\_transaccion varchar2(10) primary key not null,  cod\_sucursal varchar2(2) not null,  id\_cliente varchar2(10) not null,  tipoprestamo varchar2(5),  fechatransaccion date not null,  monto\_pago number ,  fechainsercion date not null,  iden\_usuario varchar2(5) not null,  constraint iden\_usuario foreign key(iden\_usuario) references usuarios(id\_usuario),  constraint fk\_c\_sucursal foreign key(cod\_sucursal) references sucursales(cod\_sucursal),  constraint fk\_c\_cliente foreign key(id\_cliente) references cliente(id\_cliente),  constraint fk\_c\_transaccion foreign key(tipoprestamo) references Tipo\_prestamo(Cod\_tipoprestamo),  constraint uk\_transaccion unique(id\_cliente, tipoprestamo)  ); |
|  |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 6parte \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**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

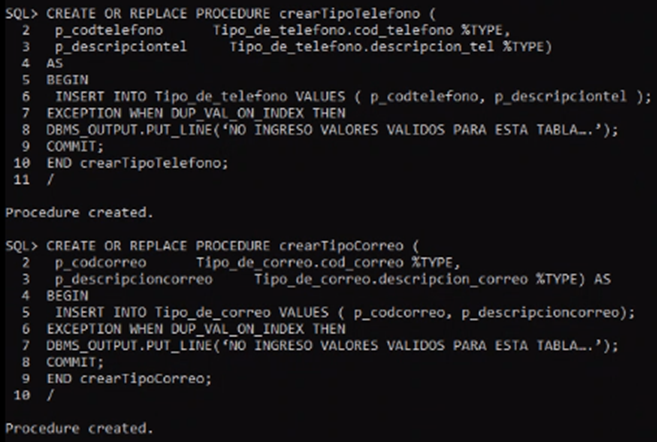
INSERT INTO Tipo\_de\_telefono VALUES ( p\_codtelefono, p\_descripciontel );

EXCEPTION WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE(‘NO INGRESO VALORES VALIDOS PARA ESTA TABLA….’);

COMMIT;

END crearTipoTelefono;



* 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

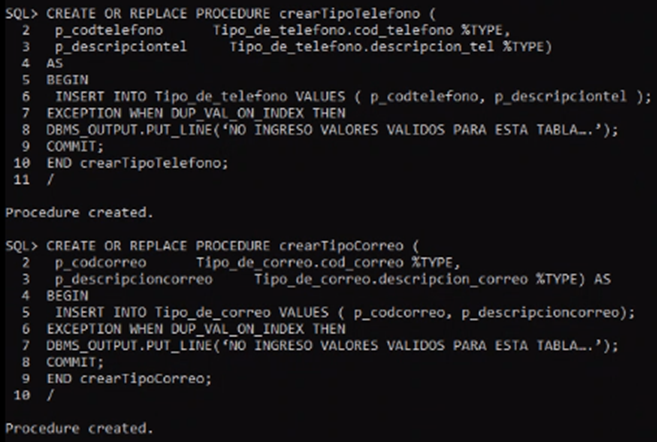
INSERT INTO Tipo\_de\_correo VALUES ( p\_codcorreo, p\_descripcioncorreo);

EXCEPTION WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE(‘NO INGRESO VALORES VALIDOS PARA ESTA TABLA….’);

COMMIT;

END crearTipoCorreo;



* Profesiones

CREATE OR REPLACE PROCEDURE crearProfesion (

p\_codprofesion Tipo\_profesion.cod\_profesion%TYPE,

p\_descripcion Tipo\_profesion.profesion%TYPE) AS

BEGIN

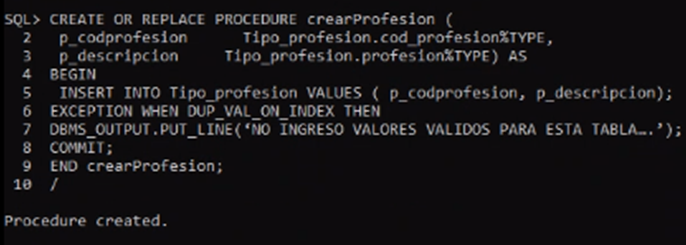
INSERT INTO Tipo\_profesion VALUES ( p\_codprofesion, p\_descripcion);

EXCEPTION WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE(‘NO INGRESO VALORES VALIDOS PARA ESTA TABLA….’);

COMMIT;

END crearProfesion;



* Sucursales,

CREATE OR REPLACE PROCEDURE crearSucursal (

p\_codsucursal sucursales.cod\_sucursal%TYPE,

p\_nombre sucursales.nombresucursal%TYPE) AS

BEGIN

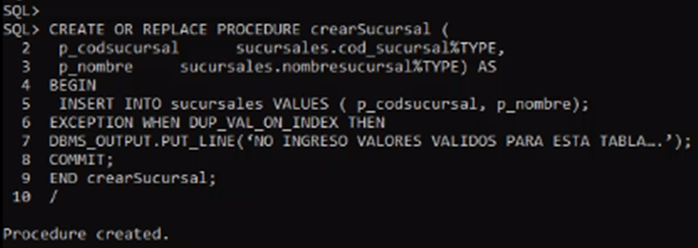
INSERT INTO sucursales VALUES ( p\_codsucursal, p\_nombre);

EXCEPTION WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE(‘NO INGRESO VALORES VALIDOS PARA ESTA TABLA….’);

COMMIT;

END crearSucursal;



* 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);

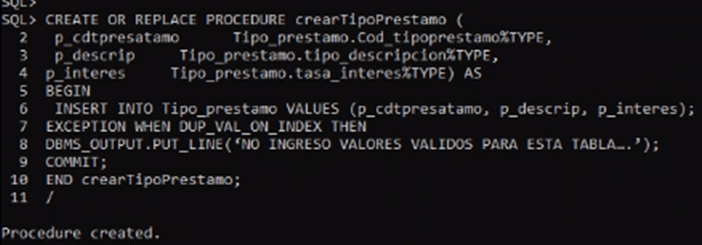
EXCEPTION WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE(‘NO INGRESO VALORES VALIDOS PARA ESTA TABLA….’);

COMMIT;

END crearTipoPrestamo;

/

****

* **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;

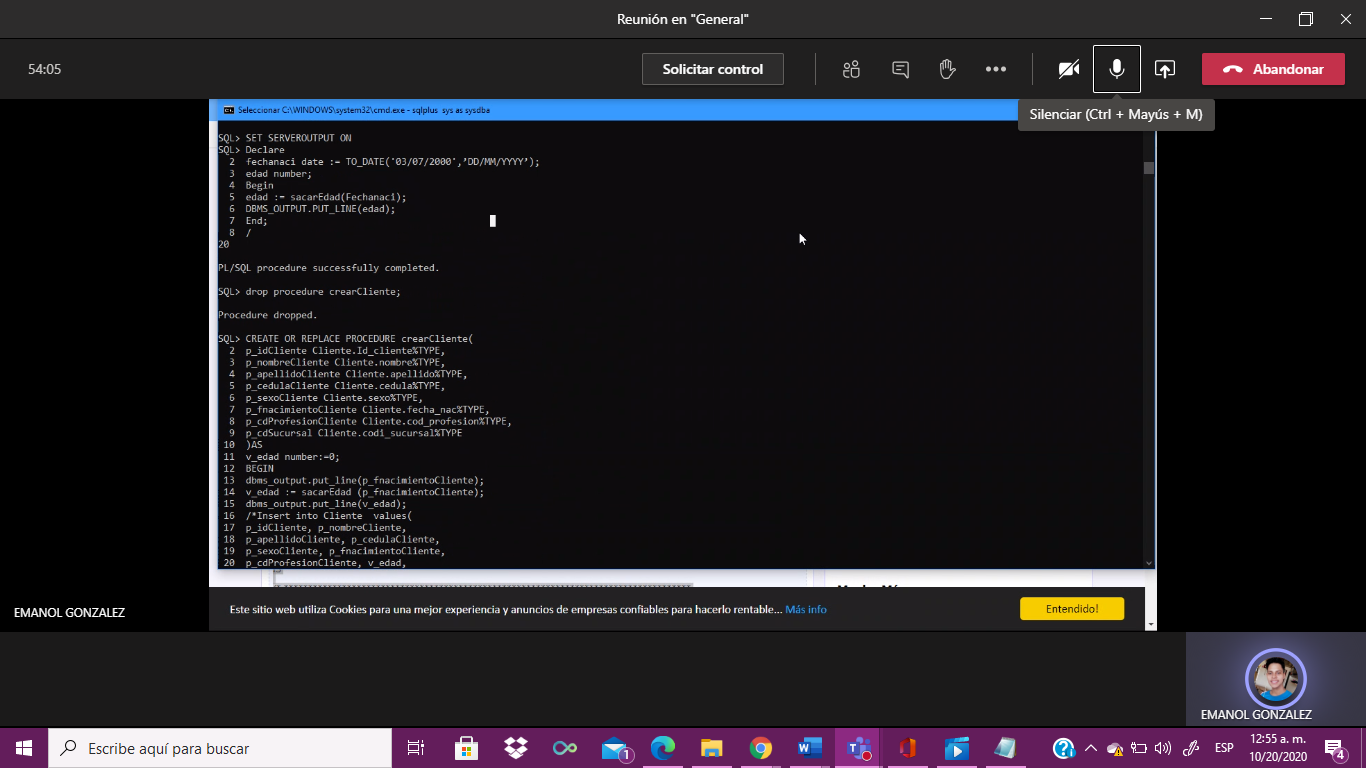
END crearUsuario;

/

**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 |
| 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; |
|  |

Probando función:



|  |
| --- |
| 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; |
|  |

**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;  / |

**Procedimiento** almacenado para la carga o inserción de los pagos recibidos de los clientes que se almacena en la tabla transacpagos. 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;  / |
|  |