

Universidad Tecnológica De Panamá
Facultad De Ingeniería en Sistemas Computacionales
LICENCIATURA EN DESAROLLO DE SOFTWARE

BASE DE DATOS II
LABORATORIO N.5

PROFESOR: HENRY LEZCANO

INTEGRANTES:

RODRÍGUEZ, VÍCTOR 20-70-7414

MOSQUERA, ALEJANDRO 8-972-1615

GONZÁLEZ, ELADIO 8-972-2440

DIEGO NAVARRO

PANAMÁ, 8 DE NOVIEMBRE DE 2023

Creacion de tablas

```
SQL> CREATE TABLE Sucursal (  
2   cod_sucursal NUMBER CONSTRAINT SUCURSAL_cod_sucursal_id_PK PRIMARY KEY,  
3   nombre_sucursal VARCHAR2(25) NOT NULL UNIQUE,  
4   montoprestamo NUMBER NOT NULL  
5 );
```

Table created.

```
SQL> CREATE TABLE Cliente (  
2   cliente_id NUMBER CONSTRAINT CLIENTE_cliente_id_PK PRIMARY KEY,  
3   cedula VARCHAR2(25) NOT NULL,  
4   nombre VARCHAR2(25) NOT NULL,  
5   apellido VARCHAR2(25) NOT NULL,  
6   sexo CHAR NOT NULL,  
7   fecha_nacimiento DATE NOT NULL,  
8   edad NUMBER NOT NULL,  
9   sucursal NUMBER NOT NULL,  
10  CONSTRAINT CLIENTE_SUCURSAL_FK FOREIGN KEY (sucursal) REFERENCES Sucursal(cod_sucursal)  
11 );
```

Table created.

```
SQL> CREATE TABLE Profesion (  
2   profesion_id NUMBER CONSTRAINT PREFESION_profesion_id_PK PRIMARY KEY,  
3   profesion VARCHAR2(25) NOT NULL,  
4   cliente NUMBER NOT NULL,  
5   CONSTRAINT PROFESION_cliente_FK FOREIGN KEY (cliente) REFERENCES Cliente(cliente_id)  
6 );
```

Table created.

```
SQL> CREATE TABLE Tipo_Email (  
2   tipo_email_id NUMBER CONSTRAINT TIPO_tipo_id_PK PRIMARY KEY,  
3   tipo_email VARCHAR2(25) NOT NULL  
4 );
```

Table created.

```
SQL> CREATE TABLE Email (  
2   email VARCHAR2(25) NOT NULL UNIQUE,  
3   cliente NUMBER NOT NULL,  
4   tipo_email NUMBER NOT NULL,  
5   CONSTRAINT EMAIL_cliente_fk FOREIGN KEY (cliente) REFERENCES Cliente(cliente_id),  
6   CONSTRAINT EMAIL_tipo_fk FOREIGN KEY (tipo_email) REFERENCES Tipo_Email(tipo_email_id),  
7   CONSTRAINT EMAIL_email_id_pk PRIMARY KEY (email, cliente)  
8 );
```

Table created.

```
SQL> CREATE TABLE Tipo_Telefono (  
2   tipo_telefono_id NUMBER CONSTRAINT TIPOTELEFONO_id_pk PRIMARY KEY,  
3   tipo_telefono varchar2(25) NOT NULL  
4 );
```

Table created.

```
SQL> CREATE TABLE Telefono (  
2   telefono VARCHAR2(25) NOT NULL,  
3   cliente NUMBER NOT NULL,  
4   tipo_telefono NUMBER NOT NULL,  
5   CONSTRAINT TELEFONO_telefono_id_pk PRIMARY KEY(telefono, cliente),  
6   CONSTRAINT Telefono_tipo_telefono FOREIGN KEY (tipo_telefono) REFERENCES Tipo_Telefono(tipo_telefono_id)  
7 );
```

Table created.

```
SQL> CREATE TABLE Tipo_Prestamo (  
2   tipo_prestamo_id NUMBER CONSTRAINT TIPO_PRESTAMO_id_pk PRIMARY KEY,  
3   tipo_prestamo VARCHAR2(50) NOT NULL  
4 );
```

Table created.

```

SQL> CREATE TABLE Prestamo (
2     numero_prestamo NUMBER NOT NULL,
3     cliente NUMBER NOT NULL,
4     tipo_prestamo NUMBER NOT NULL,
5     fecha_aprobado DATE NOT NULL,
6     monto_aprobado NUMBER NOT NULL,
7     tasa_interes NUMBER NOT NULL,
8     letra_mensual NUMBER NOT NULL,
9     monto_pagado NUMBER NOT NULL,
10    fecha_pago DATE NOT NULL,
11    sucursal NUMBER NOT NULL,
12    saldo_actual NUMBER NOT NULL,
13    interes_pagado NUMBER NOT NULL,
14    fecha_modificacion TIMESTAMP(3),
15    usuario varchar2(25) NOT NULL,
16    CONSTRAINT PRESTAMO_numero_prestamo_pk PRIMARY KEY (numero_prestamo),
17    CONSTRAINT PRESTAMO_cliente_id_fk FOREIGN KEY (cliente) REFERENCES Cliente(cliente_id),
18    CONSTRAINT PRESTAMO_tipo_prestamo_fk FOREIGN KEY (tipo_prestamo) REFERENCES Tipo_Prestamo(tipo_prestamo_id),
19    CONSTRAINT PRESTAMO_SUCURSAL_FK FOREIGN KEY (sucursal) REFERENCES Sucursal(cod_sucursal)
20 );

```

Table created.

```

SQL> CREATE TABLE transacpagos (
2     transaccion_id NUMBER CONSTRAINT TRANSACPAGOS_transaccion_id_PK PRIMARY KEY,
3     fecha_transaccion DATE NOT NULL,
4     monto_pago NUMBER NOT NULL,
5     estado char NOT NULL,
6     fecha_insercion TIMESTAMP(3) NOT NULL,
7     usuario varchar2(25) NOT NULL,
8     sucursal NUMBER NOT NULL,
9     cliente NUMBER NOT NULL,
10    prestamo NUMBER NOT NULL,
11    CONSTRAINT TRANSACPAGOS_sucursal_FK FOREIGN KEY (sucursal) REFERENCES Sucursal(cod_sucursal),
12    CONSTRAINT TRANSACPAGOS_cliente_FK FOREIGN KEY (cliente) REFERENCES Cliente(cliente_id),
13    CONSTRAINT TRANSACPAGOS_tipo_prestamo_FK FOREIGN KEY (prestamo) REFERENCES Tipo_Prestamo(tipo_prestamo_id)
14 );

```

Table created.

■ Creacion de las secuencias

```

SQL>
SQL> CREATE SEQUENCE profesion_id_seq START WITH 1 INCREMENT BY 1;

```

Sequence created.

```

SQL>
SQL>

```

```
SQL>
SQL> CREATE SEQUENCE cliente_id_seq START WITH 1 INCREMENT BY 1;

Sequence created.

SQL> █
```

```
SQL> CREATE SEQUENCE prestamo_id_seq START WITH 1 INCREMENT BY 1;

Sequence created.

SQL> █
```

```
SQL> CREATE SEQUENCE transaccion_id_seq START WITH 1 INCREMENT BY 1;

Sequence created.

SQL> █
```

- Creacion de los procedimientos almacenados y las funciones

```
SQL> CREATE OR REPLACE PROCEDURE InsertarTipoTelefono (
  2   p_tipo_telefono IN Tipo_Telefono.tipo_telefono%TYPE,
  3   p_tipo_telefono_id IN Tipo_Telefono.tipo_telefono_id%TYPE
  4 ) AS
  5 BEGIN
  6   INSERT INTO Tipo_Telefono (tipo_telefono_id, tipo_telefono)
  7     VALUES (p_tipo_telefono_id, p_tipo_telefono);
  8   COMMIT;
  9   END InsertarTipoTelefono;
 10  /

Procedure created.
```

```

SQL> CREATE OR REPLACE PROCEDURE InsertarTipoCorreo (
  2     p_tipo_correo IN Tipo_Email.tipo_email%TYPE,
  3     p_tipo_correo_id IN Tipo_Email.tipo_email_id%TYPE
  4 ) AS
  5 BEGIN
  6     INSERT INTO Tipo_Email (tipo_email_id, tipo_email)
  7         VALUES (p_tipo_correo_id, p_tipo_correo);
  8 COMMIT;
  9 END InsertarTipoCorreo;
10 /

```

Procedure created.

```

SQL> CREATE OR REPLACE PROCEDURE InsertarProfesion (
  2     p_nombre_profesion IN Profesion.profesion%TYPE,
  3     p_cliente IN Cliente.cliente_id%TYPE
  4 ) AS
  5 BEGIN
  6     INSERT INTO Profesion (profesion_id, profesion, cliente)
  7         VALUES (profesion_id_seq.NEXTVAL, p_nombre_profesion, p_cliente);
  8 COMMIT;
  9 END InsertarProfesion;
10 /

```

Procedure created.

```

SQL> CREATE OR REPLACE PROCEDURE InsertarSucursal (
  2     p_cod_sucursal IN Sucursal.cod_sucursal%TYPE,
  3     p_nombre_sucursal IN Sucursal.nombre_sucursal%TYPE,
  4     p_montoprestamo IN Sucursal.montoprestamo%TYPE
  5 ) AS
  6 BEGIN
  7     INSERT INTO Sucursal (cod_sucursal, nombre_sucursal, montoprestamo)
  8         VALUES (p_cod_sucursal, p_nombre_sucursal, p_montoprestamo);
  9 COMMIT;
10 END InsertarSucursal;
11 /

```

Procedure created.

Procedure created.

```
SQL> CREATE OR REPLACE PROCEDURE InsertarTipoPrestamo (  
2     p_tipo_prestamo IN Tipo_Prestamo.tipo_prestamo%TYPE,  
3     p_tipo_prestamo_id IN Tipo_Prestamo.tipo_prestamo_id%TYPE  
4 ) AS  
5 BEGIN  
6     INSERT INTO Tipo_Prestamo (tipo_prestamo_id, tipo_prestamo)  
7     VALUES (p_tipo_prestamo_id, p_tipo_prestamo);  
8     COMMIT;  
9     END InsertarTipoPrestamo;  
10  /
```

Procedure created.

SQL> █

```
SQL> CREATE OR REPLACE FUNCTION CalcularEdad(  
2     p_fecha_nacimiento Cliente.fecha_nacimiento%TYPE  
3 ) RETURN NUMBER IS  
4     v_edad Cliente.edad%TYPE;  
5 BEGIN  
6     v_edad := FLOOR(MONTHS_BETWEEN(SYSDATE, p_fecha_nacimiento) / 12);  
7     RETURN v_edad;  
8 END CalcularEdad;  
9  /
```

Function created.

█

```

SQL>
SQL> CREATE OR REPLACE PROCEDURE InsertarCliente(
2     p_cedula IN Cliente.cedula%TYPE,
3     p_nombre IN Cliente.nombre%TYPE,
4     p_apellido IN Cliente.apellido%TYPE,
5     psexo IN Cliente.sexo%TYPE,
6     p_fecha_nacimiento IN Cliente.fecha_nacimiento%TYPE,
7     p_email IN Email.email%TYPE,
8     p_tipo_correo IN Tipo_Email.tipo_email%TYPE,
9     p_telefono IN Telefono.telefono%TYPE,
10    p_tipo_telefono IN Tipo_Telefono.tipo_telefono%TYPE,
11    p_profesion IN Profesion.profesion%TYPE,
12    p_sucursal IN Sucursal.nombre_sucursal%TYPE
13 ) AS
14    v_tipo_correo Tipo_Email.tipo_email_id%TYPE;
15    v_tipo_telefono Tipo_Telefono.tipo_telefono_id%TYPE;
16    v_cliente_id Cliente.cliente_id%TYPE;
17    v_edad Cliente.edad%TYPE;
18    v_sucursal Sucursal.cod_sucursal%TYPE;
19    BEGIN
20
21
22    SELECT cod_sucursal INTO v_sucursal
23    FROM sucursal WHERE nombre_sucursal = p_sucursal;
24    SELECT tipo_telefono_id INTO v_tipo_telefono
25    FROM Tipo_Telefono WHERE tipo_telefono = p_tipo_telefono;
26    SELECT tipo_email_id INTO v_tipo_correo
27    FROM Tipo_Email WHERE tipo_email = p_tipo_correo;
28
29    v_edad := CalcularEdad(p_fecha_nacimiento);
30
31    INSERT INTO Cliente (cliente_id, cedula, nombre, apellido, sexo, fecha_nacimiento, edad, sucursal)
32    VALUES (cliente_id_seq.NEXTVAL, p_cedula, p_nombre, p_apellido, psexo, p_fecha_nacimiento, v_edad, v_sucursal);
33    SELECT cliente_id INTO v_cliente_id
34    FROM Cliente WHERE cedula = p_cedula;
35    InsertarProfesion(p_profesion, v_cliente_id);
36    INSERT INTO Email (email, cliente, tipo_email)
37    VALUES (p_email, v_cliente_id, v_tipo_correo);
38
39    INSERT INTO Telefono (telefono, cliente, tipo_telefono)
40    VALUES (p_telefono, v_cliente_id, v_tipo_telefono);
41    COMMIT;
42    EXCEPTION
43    WHEN NO_DATA_FOUND THEN
44        -- cuando no se encuentra el valor en la tabla
45        DBMS_OUTPUT.PUT_LINE('Valor no encontrado.');
```

Procedure created.

```
SQL> █
```



```

SQL> CREATE OR REPLACE PROCEDURE InsertarPrestamo(
2     p_cliente_cedula IN Cliente.cedula%TYPE,
3     p_tipo_prestamo IN Tipo_Prestamo.tipo_prestamo%TYPE,
4     p_fecha_aprobado IN Prestamo.fecha_aprobado%TYPE,
5     p_monto_aprobado IN Prestamo.monto_aprobado%TYPE,
6     p_tasa_interes IN Prestamo.tasa_interes%TYPE,
7     p_letra_mensual IN Prestamo.letra_mensual%TYPE,
8     p_fecha_pago IN Prestamo.fecha_pago%TYPE,
9     p_nombre_sucursal IN Sucursal.nombre_sucursal%TYPE,
10    p_usuario IN Prestamo.usuario%TYPE
11 ) AS
12 v_cliente_id Cliente.cliente_id%TYPE;
13 v_sucursal Sucursal.cod_sucursal%TYPE;
14 v_tipo_prestamo Tipo_Prestamo.tipo_prestamo_id%TYPE;
15 BEGIN
16
17 SELECT cliente_id INTO v_cliente_id
18 FROM Cliente WHERE cedula = p_cliente_cedula;
19 SELECT cod_sucursal INTO v_sucursal
20 FROM Sucursal WHERE nombre_sucursal = p_nombre_sucursal;
21 SELECT tipo_prestamo_id INTO v_tipo_prestamo
22 FROM Tipo_Prestamo WHERE tipo_prestamo = p_tipo_prestamo;
23

```

```

23
24
25     INSERT INTO Prestamo (
26         numero_prestamo,
27         cliente,
28         tipo_prestamo,
29         fecha_aprobado,
30         monto_aprobado,
31         tasa_interes,
32         letra_mensual,
33         monto_pagado,
34         fecha_pago,
35         sucursal,
36         saldo_actual,
37         interes_pagado,
38         usuario
39     )
40     VALUES (
41         prestamo_id_seq.NEXTVAL,
42         v_cliente_id,
43         v_tipo_prestamo,
44         p_fecha_aprobado,
45         p_monto_aprobado,
46         p_tasa_interes,
47         p_letra_mensual,
48         0,
49         p_fecha_pago,
50         v_sucursal,
51         p_monto_aprobado,
52         0,
53         p_usuario
54     );
55

```

```

55
56     UPDATE Sucursal
57     SET montoprestamo = montoprestamo - p_monto_aprobado
58     WHERE cod_sucursal = p_nombre_sucursal;
59
60     COMMIT;
61     EXCEPTION
62     WHEN NO_DATA_FOUND THEN
63         -- cuando no se encuentra el valor en la tabla
64         DBMS_OUTPUT.PUT_LINE('Valor no encontrado.');
```

Procedure created.

```

SQL>
SQL> CREATE OR REPLACE PROCEDURE InsertarPagosRecibidos(
2     p_monto_pago IN Transacpagos.monto_pago%TYPE,
3     p_usuario IN Transacpagos.usuario%TYPE,
4     p_sucursal IN Sucursal.nombre_sucursal%TYPE,
5     p_cliente_cedula IN Cliente.cedula%TYPE,
6     p_prestamo IN Prestamo.numero_prestamo%TYPE
7 ) AS
8
9     v_cliente_id Cliente.cliente_id%TYPE;
10    v_sucursal Sucursal.cod_sucursal%TYPE;
11    BEGIN
12
13    SELECT cliente_id INTO v_cliente_id
14    FROM Cliente WHERE cedula = p_cliente_cedula;
15    SELECT cod_sucursal INTO v_sucursal
16    FROM Sucursal WHERE nombre_sucursal = p_sucursal;
17
```

```

17
18 INSERT INTO transacpagos (
19     transaccion_id,
20     fecha_transaccion,
21     monto_pago,
22     estado,
23     fecha_insercion,
24     usuario,
25     sucursal,
26     cliente,
27     prestamo
28 )
29 VALUES(
30     transaccion_id_seq.NEXTVAL,
31     SYSDATE,
32     p_monto_pago,
33     'E',
34     SYSDATE,
35     p_usuario,
36     v_sucursal,
37     v_cliente_id,
38     p_prestamo
39 );
40

```

```

40
41 EXCEPTION
42     WHEN NO_DATA_FOUND THEN
43         -- cuando no se encuentra el valor en la tabla
44         DBMS_OUTPUT.PUT_LINE('Valor no encontrado. ');
45     WHEN OTHERS THEN
46         -- otras excepciones
47         DBMS_OUTPUT.PUT_LINE('Ocurrió un error en la inserción. ');
48         DBMS_OUTPUT.PUT_LINE('Ocurrió un error en la inserción. ');
49         -- para mantener la integridad de los datos
50         ROLLBACK;
51 END InsertarPagosRecibidos;
52 /

```

Procedure created.

SQL> █

```

SQL>
SQL> CREATE OR REPLACE FUNCTION calcularInteres(
  2     p_saldo_prestamo Prestamo.saldo_actual%TYPE,
  3     p_tasa_interes Prestamo.tasa_interes%TYPE
  4 ) RETURN NUMBER IS
  5 BEGIN
  6     RETURN p_saldo_prestamo * p_tasa_interes / 100;
  7 END calcularInteres;
  8 /

```

Function created.

```
SQL> █
```

```

SQL> CREATE OR REPLACE PROCEDURE ActualizarPagosRecibo AS
  2     CURSOR c_pagos IS
  3         SELECT transaccion_id, cliente, prestamo, monto_pago, sucursal
  4         FROM Transacpagos;
  5     v_interes Prestamo.interes_pagado%TYPE;
  6     v_monto_pendiente Prestamo.saldo_actual%TYPE;
  7     v_saldo Prestamo.saldo_actual%TYPE;
  8 BEGIN
  9     FOR r_pago IN c_pagos LOOP
 10         SELECT tasa_interes, saldo_actual INTO v_interes, v_saldo
 11         FROM Prestamo WHERE numero_prestamo = r_pago.prestamo AND cliente = r_pago.cliente;
 12         v_interes := calcularInteres(
 13             p_saldo_prestamo => v_saldo,
 14             p_tasa_interes => v_interes
 15         );
 16         v_monto_pendiente := r_pago.monto_pago - v_interes;
 17         IF v_monto_pendiente > 0 THEN
 18             v_saldo := v_monto_pendiente;
 19             UPDATE Prestamo
 20             SET saldo_actual = v_saldo
 21             WHERE numero_prestamo = r_pago.prestamo AND cliente = r_pago.cliente;
 22         END IF;
 23         UPDATE Prestamo
 24         SET interes_pagado = interes_pagado + v_interes
 25         WHERE numero_prestamo = r_pago.prestamo AND cliente = r_pago.cliente;
 26         UPDATE Sucursal
 27         SET montoprestamo = montoprestamo + v_saldo
 28         WHERE cod_sucursal = r_pago.sucursal;
 29     END LOOP;
 30
 31     COMMIT;
 32     EXCEPTION
 33         WHEN NO_DATA_FOUND THEN
 34             -- cuando no se encuentra el valor en la tabla
 35             DBMS_OUTPUT.PUT_LINE('Valor no encontrado.');
```

Procedure created.

```
SQL> --tipo celular
SQL> BEGIN
  2   InsertarTipoTelefono('celular', 1);
  3   InsertarTipoTelefono('residencia', 2);
  4   InsertarTipoTelefono('familiar mas cercano', 3);
  5   InsertarTipoTelefono('conyugue', 4);
  6   END;
  7   /
```

PL/SQL procedure successfully completed.

```
SQL> --tipo correo
SQL> BEGIN
  2   InsertarTipoCorreo('academico', 1);
  3   InsertarTipoCorreo('laboral', 2);
  4   InsertarTipoCorreo('personal', 3);
  5   END;
  6   /
```

PL/SQL procedure successfully completed.

```
SQL> --tipo prestamo
SQL> BEGIN
  2   InsertarTipoPrestamo('personal', 1);
  3   InsertarTipoPrestamo('hipoteca', 2);
  4   InsertarTipoPrestamo('auto', 3);
  5   InsertarTipoPrestamo('garantizado con ahorros', 4);
  6   END;
  7   /
```

PL/SQL procedure successfully completed.

```

SQL> DECLARE
2     v_cod_sucursal Sucursal.cod_sucursal%TYPE;
3     v_nombre_sucursal Sucursal.nombre_sucursal%TYPE;
4     v_montoprestamo Sucursal.montoprestamo%TYPE;
5 BEGIN
6     DBMS_OUTPUT.PUT_LINE('ingresar codigo sucursal');
7     v_cod_sucursal := &cod_sucursal;
8     DBMS_OUTPUT.PUT_LINE('ingresar nombre sucursal');
9     v_nombre_sucursal := &nombre_sucursal;
10    DBMS_OUTPUT.PUT_LINE('ingresar monto prestamo');
11    v_montoprestamo := &montoprestamo;
12
13    BEGIN
14        InsertarSucursal(
15            p_cod_sucursal => v_cod_sucursal,
16            p_nombre_sucursal => v_nombre_sucursal,
17            p_montoprestamo => v_montoprestamo
18        );
19
20        DBMS_OUTPUT.PUT_LINE('Sucursal insertada correctamente');
21        COMMIT;
22    EXCEPTION
23        WHEN DUP_VAL_ON_INDEX THEN
24            -- de clave primaria duplicada
25            DBMS_OUTPUT.PUT_LINE('La clave primaria ya existe. No se puede insertar.');
```

26 WHEN OTHERS THEN

27 -- Maneja otras excepciones

28 DBMS_OUTPUT.PUT_LINE('Ocurrió un error al insertar la sucursal');

29 ROLLBACK;

30 END;

31 END;

32 /

Enter value for cod_sucursal: 1

old 7: v_cod_sucursal := &cod_sucursal;

new 7: v_cod_sucursal := 1;

Enter value for nombre_sucursal: 'calle 50'

old 9: v_nombre_sucursal := &nombre_sucursal;

new 9: v_nombre_sucursal := 'calle 50';

Enter value for montoprestamo: 500000

old 11: v_montoprestamo := &montoprestamo;

new 11: v_montoprestamo := 500000;

PL/SQL procedure successfully completed.

SQL> █

