

UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES DEPARTAMENTO DE
INGENIERÍA DE SOFTWARE

IMPLEMENTACION DE BASE DE
DATOS II

TEMA:
Proyecto Semestral

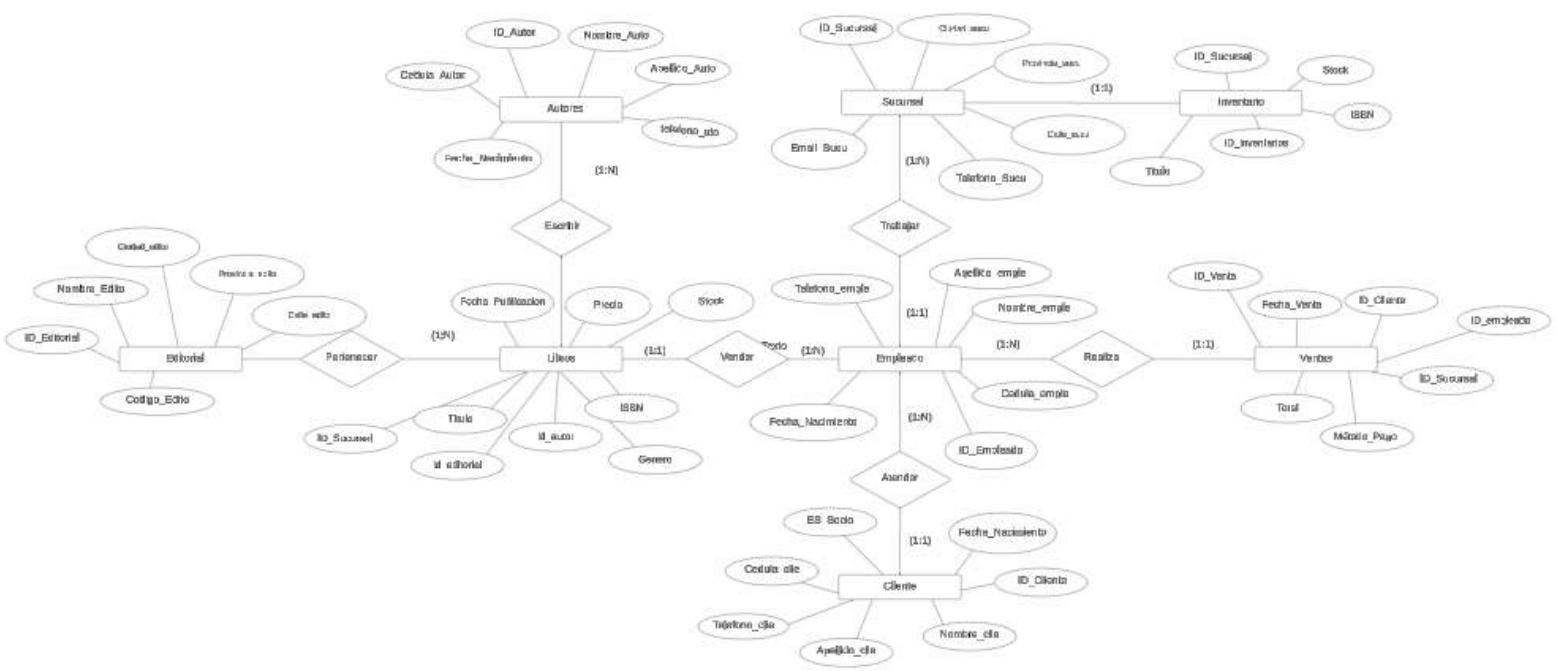
GRUPO 8:
JOSE GOMEZ 20-14-7317
ENRIQUE BATISTA 8-1011-1855
ANTONIO CARMONA 8-1002-766
SEBASTIAN MEJIA 4-816-1428
MARIA QUIÑONES 8-988-1507

PROFESOR:
HENRY LEZCANO

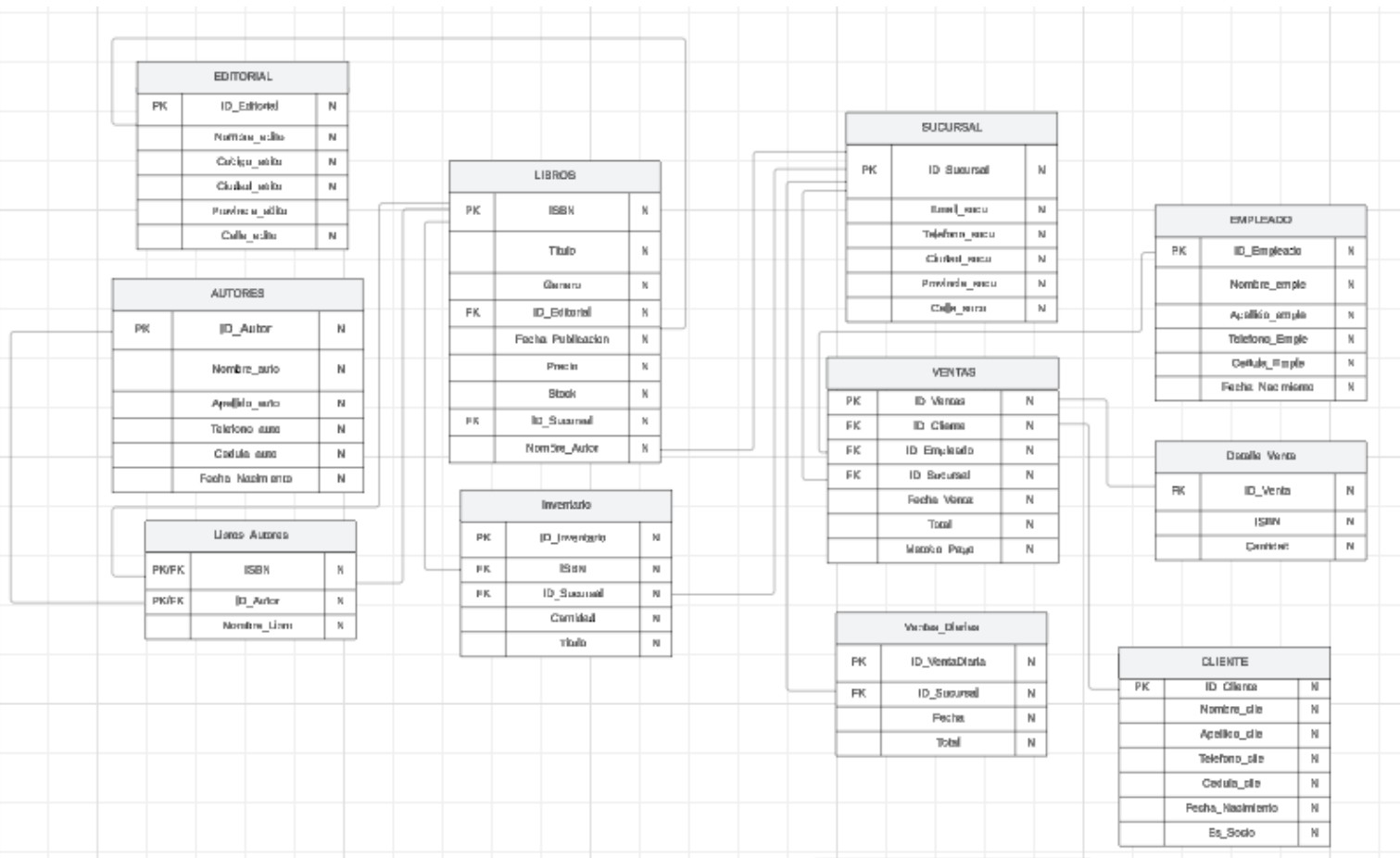
GRUPO:
1SF133

FECHA DE ENTREGA:
03/07/2024

Entidad relación



Lógico relacional normalizado



Análisis de requerimientos

| Requisito | Descripción |
|---|--|
| Ingreso y Gestión de Autores en la Base de Datos | Crear un procedimiento para agregar autores a la tabla Autores, garantizando unicidad en la cédula y generando IDs únicos. Incluir una vista para facilitar la consulta. |
| Ingreso y Gestión de Sucursales en la Base de Datos | Desarrollar un procedimiento para añadir sucursales a la tabla Sucursal, asegurando unicidad en teléfono y correo electrónico. Incluir una vista para consulta de sucursales. |
| Ingreso y Gestión de Libros en Sucursales y stock en el Inventario | Crear un procedimiento para añadir libros a sucursales, actualizar o insertar en las tablas Libros e Inventario. Manejar errores y mostrar mensajes. |
| Búsqueda de Libros en una Sucursal | Desarrollar un procedimiento para buscar libros en una sucursal específica y mostrar disponibilidad, precio, stock y autor. |
| Búsqueda de Libros por Autor | Crear un procedimiento para listar libros de un autor específico, mostrando detalles como ISBN, título, género y stock. |
| Aplicar Descuentos en Facturas | Implementar una función para aplicar descuentos en las facturas: 20% para miembros, 10% para mayores de 55 años, o ninguno si no se cumplen estas condiciones. |
| Generar Factura de la Compra | Desarrollar un procedimiento para registrar compras, calcular totales con descuentos y actualizar inventario. Incluir manejo de errores y mensajes de estado. |
| Registrar Ventas Diarias | Crear un procedimiento para registrar automáticamente el total de ventas diarias por sucursal en la tabla VentasDiarias. |
| Vista para Consultar Ventas Diarias | Crear una vista para consultar un resumen de ventas diarias por sucursal y fecha, agrupando y ordenando la información. |
| Generar Informe de Ventas por Sucursal | Desarrollar un procedimiento para generar un informe detallado de ventas por sucursal, incluyendo detalles de cada venta y libros vendidos. |
| Actualizar Información de Clientes | Crear un procedimiento para actualizar la información de los clientes en la tabla Clientes, incluyendo nombre, teléfono, cédula, y estatus de membresía. |
| Ingreso y Gestión de Clientes En la Base de Datos | Desarrollar un procedimiento almacenado denominado IngresarCliente para añadir nuevos registros a la tabla Clientes, asegurando que la cédula sea única y generando automáticamente IDs únicos. También, crear una vista llamada |

| | |
|--|---|
| | Vista_Clientes para simplificar la consulta de los clientes añadidos. |
|--|---|

Documentación de procesos

1. Ingresar Autor: Crear un procedimiento para agregar autores a la tabla Autores, garantizando unicidad en la cédula y generando IDs únicos. Incluir una vista para facilitar la consulta.

--Restricción de unicidad para el atributo cedula

ALTER TABLE Autores

ADD CONSTRAINT uq_autores_cedula UNIQUE (Cedula_auto);

```
SQL>
SQL> ALTER TABLE Autores
  2  ADD CONSTRAINT uq_autores_cedula UNIQUE (Cedula_auto);
Table altered.
```

--Ingresar Autor

CREATE SEQUENCE Autores_SEQ

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

SET SERVEROUTPUT ON;

-- Crear el procedimiento almacenado

CREATE OR REPLACE PROCEDURE IngresarAutor (

p_Nombre_auto IN VARCHAR2,

p_Apellido_auto IN VARCHAR2,

p_Telefono_auto IN VARCHAR2,

p_Cedula_auto IN VARCHAR2,

p_Fecha_nacimiento IN DATE

)

AS

v_Inserted BOOLEAN := FALSE; -- Variable para controlar si se realizó la inserción

BEGIN

BEGIN

-- Insertar nuevo autor

INSERT INTO Autores (ID_Autor, Nombre_auto, Apellido_auto, Telefono_auto, Cedula_auto, Fecha_Nacimiento)

VALUES (Autores_SEQ.NEXTVAL, p_Nombre_auto, p_Apellido_auto, p_Telefono_auto, p_Cedula_auto, p_Fecha_nacimiento);

v_Inserted := TRUE; -- Indicar que se realizó la inserción

EXCEPTION

WHEN DUP_VAL_ON_INDEX THEN

-- Generar error si el autor con la misma cédula ya existe

RAISE_APPLICATION_ERROR(-20001, 'Ya existe un autor con la misma cédula.');

WHEN OTHERS THEN

-- Manejar otros errores

RAISE_APPLICATION_ERROR(-20002, 'Error inesperado: ' || SQLERRM);

END;

-- Mostrar mensaje de éxito solo si la inserción fue realizada

IF v_Inserted THEN

DBMS_OUTPUT.PUT_LINE('Autor ingresado correctamente.');

END IF;

END;

/

-- Solicitar datos al usuario y llamar al procedimiento

DECLARE

v_Nombre_auto VARCHAR2(100);

v_Apellido_auto VARCHAR2(100);

v_Telefono_auto VARCHAR2(20);

v_Cedula_auto VARCHAR2(20);

v_Fecha_nacimiento DATE;

BEGIN

-- Solicitar datos al usuario

v_Nombre_auto := '&Ingrese_nombre_auto';

v_Apellido_auto := '&Ingrese_apellido_auto';

v_Telefono_auto := '&Ingrese_telefono_auto';

v_Cedula_auto := '&Ingrese_cedula_auto';

v_Fecha_nacimiento := TO_DATE('&Ingrese_fecha_nacimiento', 'YYYY-MM-DD');

-- Llamar al procedimiento para agregar el autor

IngresarAutor(v_Nombre_auto, v_Apellido_auto, v_Telefono_auto, v_Cedula_auto,
v_Fecha_nacimiento);

END;

/

```

SQL> SET SERVEROUTPUT ON;
SQL>
SQL> -- Crear el procedimiento almacenado
SQL> CREATE OR REPLACE PROCEDURE IngresarAutor (
2   p_Nombre_auto IN VARCHAR2,
3   p_Apellido_auto IN VARCHAR2,
4   p_Telefono_auto IN VARCHAR2,
5   p_Cedula_auto IN VARCHAR2,
6   p_Fecha_nacimiento IN DATE
7 )
8 AS
9   v_Inserted BOOLEAN := FALSE; -- Variable para controlar si se realizó la inserción
10 BEGIN
11   BEGIN
12     -- Insertar nuevo autor
13     INSERT INTO Autores (ID_Autor, Nombre_auto, Apellido_auto, Telefono_auto, Cedula_auto, Fecha_Nacimiento)
14     VALUES (Autores_SEQ.NEXTVAL, p_Nombre_auto, p_Apellido_auto, p_Telefono_auto, p_Cedula_auto, p_Fecha_nacimiento);
15
16     v_Inserted := TRUE; -- Indicar que se realizó la inserción
17   EXCEPTION
18     WHEN DUP_VAL_ON_INDEX THEN
19       -- Generar error si el autor con la misma cédula ya existe
20       RAISE_APPLICATION_ERROR(-20001, 'Ya existe un autor con la misma cédula.');
```

```

21     WHEN OTHERS THEN
22       -- Manejar otros errores
23       RAISE_APPLICATION_ERROR(-20002, 'Error inesperado: ' || SQLERRM);
24   END;
25
26   -- Mostrar mensaje de éxito solo si la inserción fue realizada
27   IF v_Inserted THEN
28     DBMS_OUTPUT.PUT_LINE('Autor ingresado correctamente.');
```

```

29   END IF;
30 END;
31 /
Procedure created.

SQL>
SQL> -- Solicitar datos al usuario y llamar al procedimiento
SQL> DECLARE
2   v_Nombre_auto VARCHAR2(100);
3   v_Apellido_auto VARCHAR2(100);
4   v_Telefono_auto VARCHAR2(20);
5   v_Cedula_auto VARCHAR2(20);
6   v_Fecha_nacimiento DATE;
7 BEGIN
8   -- Solicitar datos al usuario
9   v_Nombre_auto := '&Ingreso_nombre_auto';
10  v_Apellido_auto := '&Ingreso_apellido_auto';
11  v_Telefono_auto := '&Ingreso_telefono_auto';
12  v_Cedula_auto := '&Ingreso_cedula_auto';
13  v_Fecha_nacimiento := TO_DATE('&Ingreso_fecha_nacimiento', 'YYYY-MM-DD');
```

```

14
15  -- Llamar al procedimiento para agregar el autor
16  IngresarAutor(v_Nombre_auto, v_Apellido_auto, v_Telefono_auto, v_Cedula_auto, v_Fecha_nacimiento);
17 END;
18 /
Enter value for ingreso_nombre_auto: Enrique
old 9:  v_Nombre_auto := '&Ingreso_nombre_auto';
new 9:  v_Nombre_auto := 'Enrique';
Enter value for ingreso_apellido_auto: Batista
old 10: v_Apellido_auto := '&Ingreso_apellido_auto';
new 10: v_Apellido_auto := 'Batista';
Enter value for ingreso_telefono_auto: 1234
old 11: v_Telefono_auto := '&Ingreso_telefono_auto';
new 11: v_Telefono_auto := '1234';
Enter value for ingreso_cedula_auto: 12345678
old 12: v_Cedula_auto := '&Ingreso_cedula_auto';
new 12: v_Cedula_auto := '12345678';
Enter value for ingreso_fecha_nacimiento: 2024/12/12
old 13: v_Fecha_nacimiento := TO_DATE('&Ingreso_fecha_nacimiento', 'YYYY-MM-DD');
```

```

new 13: v_Fecha_nacimiento := TO_DATE('2024/12/12', 'YYYY-MM-DD');
Autor ingresado correctamente.

PL/SQL procedure successfully completed.

```

--Crear vista de los autores

CREATE OR REPLACE VIEW Vista_Autores AS

SELECT

ID_Autor AS "ID de Autor",

Nombre_auto AS "Nombre",

Apellido_auto AS "Apellido",

Telefono_auto AS "Teléfono",

Cedula_auto AS "Cédula",

Fecha_Nacimiento AS "Fecha de Nacimiento"

FROM Autores;

SELECT * FROM Vista_Autores;



| ID de Autor | Nombre | Apellido | Teléfono | Fecha de Nacimiento |
|-------------|--------|----------|----------|---------------------|
| 1 | Carlos | Alvarado | 3200 | 22/08/1978 |

-
2. **Ingresar nueva sucursal:** Desarrollar un procedimiento para añadir sucursales a la tabla Sucursal, asegurando unicidad en teléfono y correo electrónico. Incluir una vista para consulta de sucursales.

--crear secuencia para generación de ID

CREATE SEQUENCE seq_sucursal_id

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

```
SQL> CREATE SEQUENCE seq_sucursal_id
  2   START WITH 1
  3   INCREMENT BY 1
  4   NOCACHE
  5   NOCYCLE;

Sequence created.
```

-- Agregar una nueva restricción de unicidad

```
ALTER TABLE Sucursal
```

```
ADD CONSTRAINT uq_sucursal_telefono_email UNIQUE (Telefono_sucu, Email_sucu);
```

```
SQL> -- Modificar la tabla Sucursal para agregar una restricción de unicidad
SQL> ALTER TABLE Sucursal
  2   ADD CONSTRAINT uq_sucursal_direccion_telefono UNIQUE (Direccion_sucu, Telefono_sucu);

Table altered.
```

```
SET SERVEROUTPUT ON;
```

-- Crear el procedimiento almacenado

```
CREATE OR REPLACE PROCEDURE sp_AgregarSucursal (
```

```
  p_Ciudad IN VARCHAR2,
```

```
  p_Provincia IN VARCHAR2,
```

```
  p_Calle IN VARCHAR2,
```

```
  p_Telefono IN VARCHAR2,
```

```
  p_Email IN VARCHAR2
```

```
)
```

```
AS
```

```
  v_count INTEGER;
```

```
  v_Inserted BOOLEAN := FALSE; -- Variable para controlar si se realizó la inserción
```

```
BEGIN
```

```
  -- Verificar si ya existe una sucursal con la misma combinación de dirección (ciudad, provincia,
  calle) y correo electrónico
```

```
  SELECT COUNT(*)
```

```
  INTO v_count
```

```
  FROM Sucursal
```

```
WHERE Ciudad_sucu = p_Ciudad  
  
AND Provincia_sucu = p_Provincia  
  
AND Calle_sucu = p_Calle  
  
AND Email_sucu = p_Email;
```

```
IF v_count > 0 THEN
```

```
-- Generar error si la sucursal ya existe
```

```
RAISE_APPLICATION_ERROR(-20000, 'Ya existe una sucursal con la misma dirección y correo electrónico.');
```

```
ELSE
```

```
-- Insertar nueva sucursal si no existe
```

```
INSERT INTO Sucursal (ID_Sucursal, Ciudad_sucu, Provincia_sucu, Calle_sucu, Telefono_sucu, Email_sucu)
```

```
VALUES (seq_sucursal_id.NEXTVAL, p_Ciudad, p_Provincia, p_Calle, p_Telefono, p_Email);
```

```
v_Inserted := TRUE; -- Indicar que se realizó la inserción
```

```
END IF;
```

```
-- Mostrar mensaje de éxito solo si la inserción fue realizada
```

```
IF v_Inserted THEN
```

```
DBMS_OUTPUT.PUT_LINE('Sucursal agregada correctamente.');
```

```
END IF;
```

```
EXCEPTION
```

```
WHEN OTHERS THEN
```

```
DBMS_OUTPUT.PUT_LINE('Error inesperado: ' || SQLERRM);
```

```
END;
```

```
/
```

```

PL/SQL procedure successfully completed.

SQL> SET SERVEROUTPUT ON;
SQL>
SQL> -- Crear el procedimiento almacenado
SQL> CREATE OR REPLACE PROCEDURE sp_AgregarSucursal (
  2   p_Ciudad IN VARCHAR2,
  3   p_Provincia IN VARCHAR2,
  4   p_Calle IN VARCHAR2,
  5   p_Telefono IN VARCHAR2,
  6   p_Email IN VARCHAR2
  7 )
  8 AS
  9   v_count INTEGER;
10   v_Inserted BOOLEAN := FALSE; -- Variable para controlar si se realizó la inserción
11 BEGIN
12   -- Verificar si ya existe una sucursal con la misma combinación de dirección (ciudad, provincia, calle) y correo electrónico
13   SELECT COUNT(*)
14   INTO v_count
15   FROM Sucursal
16   WHERE Ciudad_sucu = p_Ciudad
17   AND Provincia_sucu = p_Provincia
18   AND Calle_sucu = p_Calle
19   AND Email_sucu = p_Email;
20
21   IF v_count > 0 THEN
22     -- Generar error si la sucursal ya existe
23     RAISE_APPLICATION_ERROR(-20000, 'Ya existe una sucursal con la misma dirección y correo electrónico.');
```

-- Solicitar datos al usuario y llamar al procedimiento

DECLARE

v_Ciudad VARCHAR2(100);

v_Provincia VARCHAR2(100);

v_Calle VARCHAR2(255);

v_Telefono VARCHAR2(20);

v_Email VARCHAR2(100);

BEGIN

-- Solicitar datos al usuario

v_Ciudad := '&Ingrese_ciudad_sucursal';

v_Provincia := '&Ingrese_provincia_sucursal';

v_Calle := '&Ingrese_calle_sucursal';

v_Telefono := '&Ingrese_telefono_sucursal';

v_Email := '&Ingrese_email_sucursal';

-- Llamar al procedimiento para agregar la sucursal

sp_AgregarSucursal(v_Ciudad, v_Provincia, v_Calle, v_Telefono, v_Email);

END;

```

SQL>
SQL> -- Solicitar datos al usuario y llamar al procedimiento
SQL> DECLARE
2   v_Ciudad VARCHAR2(100);
3   v_Provincia VARCHAR2(100);
4   v_Calle VARCHAR2(255);
5   v_Telefono VARCHAR2(20);
6   v_Email VARCHAR2(100);
7 BEGIN
8   -- Solicitar datos al usuario
9   v_Ciudad := '&Ingreso_ciudad_sucursal';
10  v_Provincia := '&Ingreso_provincia_sucursal';
11  v_Calle := '&Ingreso_calle_sucursal';
12  v_Telefono := '&Ingreso_telefono_sucursal';
13  v_Email := '&Ingreso_email_sucursal';
14
15  -- llamar al procedimiento para agregar la sucursal
16  sp_AgregarSucursal(v_Ciudad, v_Provincia, v_Calle, v_Telefono, v_Email);
17 END;
18 /

```

-- Crear una vista para mostrar los detalles de las sucursales de manera clara

```
CREATE OR REPLACE VIEW vista_Sucursales AS
```

```
SELECT
```

```
    ID_Sucursal AS "ID de Sucursal",
```

```
    Ciudad_sucu AS "Ciudad",
```

```
    Provincia_sucu AS "Provincia",
```

```
    Calle_sucu AS "Calle",
```

```
    Telefono_sucu AS "Teléfono",
```

```
    Email_sucu AS "Correo Electrónico"
```

```
FROM Sucursal;
```

```
/;
```

```

SQL>
SQL>
SQL> -- Crear una vista para mostrar todos los atributos de la tabla Sucursal
SQL> CREATE OR REPLACE VIEW vw_Sucursal AS
2  SELECT
3     ID_Sucursal,
4     Ciudad_sucu,
5     Provincia_sucu,
6     Calle_sucu,
7     Telefono_sucu,
8     Email_sucu
9  FROM
10     Sucursal;
View created.

```

-- Consultar la vista para ver la información de las sucursales

```
SELECT * FROM vista_Sucursales;
```

```

Enter value for Ingrese_ciudad_sucursal: panama
old 5: v_Ciudad := 'Alingrese_ciudad_sucursal';
new 5: v_Ciudad := 'panama';
Enter value for Ingrese_provincia_sucursal: panama
old 10: v_Provincia := 'Alingrese_provincia_sucursal';
new 10: v_Provincia := 'panama';
Enter value for Ingrese_calle_sucursal: 15
old 11: v_Calle := 'Alingrese_calle_sucursal';
new 11: v_Calle := '15';
Enter value for Ingrese_telefono_sucursal: 1234
old 12: v_Telefono := 'Alingrese_telefono_sucursal';
new 12: v_Telefono := '1234';
Enter value for Ingrese_email_sucursal: panama@gmail.com
old 13: v_Email := 'Alingrese_email_sucursal';
new 13: v_Email := 'panama@gmail.com';
Sucursal agregada correctamente.

PL/SQL procedure successfully completed.

```

```

Enter value for Ingrese_ciudad_sucursal: panama
old 5: v_Ciudad := 'Alingrese_ciudad_sucursal';
new 5: v_Ciudad := 'panama';
Enter value for Ingrese_provincia_sucursal: panama
old 10: v_Provincia := 'Alingrese_provincia_sucursal';
new 10: v_Provincia := 'panama';
Enter value for Ingrese_calle_sucursal: 15
old 11: v_Calle := 'Alingrese_calle_sucursal';
new 11: v_Calle := '15';
Enter value for Ingrese_telefono_sucursal: 1234
old 12: v_Telefono := 'Alingrese_telefono_sucursal';
new 12: v_Telefono := '1234';
Enter value for Ingrese_email_sucursal: panama@gmail.com
old 13: v_Email := 'Alingrese_email_sucursal';
new 13: v_Email := 'panama@gmail.com';
Error Inesperado: ORA-20000: Ya existe una sucursal con la misma direcci3n y correo electr3nico.

PL/SQL procedure successfully completed.

```

| ID_SUCURSAL | CIUDAD_SUCU | PROVINCIA_SUCU | TELEFONO_SUCU | EMAIL_SUCU |
|-------------|-------------|----------------|---------------|------------------|
| 1 | | | 1234 | panama@gmail.com |
| 2 | | | 1234 | panama@gmail.com |
| 3 | | | 1234 | panama@gmail.com |
| 4 | panama | panama | 1234 | panama@gmail.com |

3. **Ingresar libro en sucursal** : Crear un procedimiento para a adir libros a sucursales, actualizar o insertar en las tablas Libros e Inventario. Manejar errores y mostrar mensajes.

SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE InsertarLibroEnSucursal (

p_ISBN IN VARCHAR2,

p_Titulo IN VARCHAR2,

p_Genero IN VARCHAR2,

p_ID_Editorial IN NUMBER,

p_Fecha_Publicacion IN DATE,

p_Precio IN NUMBER,

p_Stock IN NUMBER,

```

    p_ID_Sucursal IN NUMBER,

    p_Nombre_Autor IN VARCHAR2
)

AS

    v_Count NUMBER;

BEGIN

    -- Verificar si el libro ya existe en la tabla Libros

    BEGIN

        SELECT COUNT(*)

        INTO v_Count

        FROM Libros

        WHERE ISBN = p_ISBN;

    IF v_Count > 0 THEN

        -- Actualizar libro existente en la tabla Libros

        UPDATE Libros

        SET Titulo = p_Titulo,

            Genero = p_Genero,

            ID_Editorial = p_ID_Editorial,

            Fecha_Publicacion = p_Fecha_Publicacion,

            Precio = p_Precio,

            Stock = Stock + p_Stock, -- Incrementar el stock existente

            Nombre_Autor = p_Nombre_Autor

        WHERE ISBN = p_ISBN;

    ELSE

        -- Insertar nuevo libro si no existe en Libros

```

```
INSERT INTO Libros (ISBN, Titulo, Genero, ID_Editorial, Fecha_Publicacion, Precio, Stock, ID_Sucursal, Nombre_Autor)
```

```
VALUES (p_ISBN, p_Titulo, p_Genero, p_ID_Editorial, p_Fecha_Publicacion, p_Precio, p_Stock, p_ID_Sucursal, p_Nombre_Autor);
```

```
END IF;
```

```
EXCEPTION
```

```
WHEN OTHERS THEN
```

```
RAISE_APPLICATION_ERROR(-20001, 'Error al verificar o insertar en Libros: ' || SQLERRM);
```

```
END;
```

```
-- Verificar si el libro ya existe en la tabla Inventario
```

```
BEGIN
```

```
SELECT COUNT(*)
```

```
INTO v_Count
```

```
FROM Inventario
```

```
WHERE ISBN = p_ISBN AND ID_Sucursal = p_ID_Sucursal;
```

```
IF v_Count > 0 THEN
```

```
-- Si el libro ya existe en Inventario para la misma sucursal, actualizar el stock
```

```
UPDATE Inventario
```

```
SET Stock = Stock + p_Stock
```

```
WHERE ISBN = p_ISBN AND ID_Sucursal = p_ID_Sucursal;
```

```
ELSE
```

```
-- Si el libro no existe en Inventario para esta sucursal, insertar un nuevo registro
```

```
INSERT INTO Inventario (ID_Inventario, ISBN, ID_Sucursal, Stock, Titulo)
```

```
VALUES (Inventario_SEQ.NEXTVAL, p_ISBN, p_ID_Sucursal, p_Stock, p_Titulo);
```


END IF;

EXCEPTION

WHEN OTHERS THEN

RAISE_APPLICATION_ERROR(-20002, 'Error al actualizar o insertar en Inventario: ' ||
SQLERRM);

END;

DBMS_OUTPUT.PUT_LINE('Libro agregado correctamente en la sucursal.');

END;

/

/

```
SQL> CREATE OR REPLACE PROCEDURE InsertarLibroEnSucursal (  
2   p_ISBN IN VARCHAR2,  
3   p_Titulo IN VARCHAR2,  
4   p_Genero IN VARCHAR2,  
5   p_ID_Editorial IN NUMBER,  
6   p_Fecha_Publicacion IN DATE,  
7   p_Precio IN NUMBER,  
8   p_Stock IN NUMBER,  
9   p_ID_Sucursal IN NUMBER  
10 )  
11 AS  
12   v_Count NUMBER;  
13 BEGIN  
14   -- Verificar si el libro ya existe  
15   BEGIN  
16     SELECT COUNT(*)  
17     INTO v_Count  
18     FROM Libros  
19     WHERE ISBN = p_ISBN;  
20  
21     IF v_Count > 0 THEN  
22       RAISE_APPLICATION_ERROR(-20001, 'El libro con ISBN: ' || p_ISBN || ' ya existe.');23     END IF;  
24   EXCEPTION  
25     WHEN OTHERS THEN  
26       RAISE_APPLICATION_ERROR(-20002, 'Error al verificar el libro: ' || SQLERRM);  
27   END;  
28  
29   -- Verificar si la sucursal existe  
30   BEGIN  
31     SELECT COUNT(*)  
32     INTO v_Count  
33     FROM Sucursal  
34     WHERE ID_Sucursal = p_ID_Sucursal;  
35  
36     IF v_Count = 0 THEN  
37       RAISE_APPLICATION_ERROR(-20003, 'La sucursal con ID: ' || p_ID_Sucursal || ' no existe.');38     END IF;  
39   EXCEPTION  
40     WHEN OTHERS THEN  
41       RAISE_APPLICATION_ERROR(-20004, 'Error al verificar la sucursal: ' || SQLERRM);  
42   END;  
43  
44   -- Insertar el libro  
45   BEGIN  
46     INSERT INTO Libros (ISBN, Titulo, Genero, ID_Editorial, Fecha_Publicacion, Precio, Stock, ID_Sucursal)  
47     VALUES (p_ISBN, p_Titulo, p_Genero, p_ID_Editorial, p_Fecha_Publicacion, p_Precio, p_Stock, p_ID_Sucursal);  
48   EXCEPTION  
49     WHEN DUP_VAL_ON_INDEX THEN  
50       RAISE_APPLICATION_ERROR(-20005, 'El libro con ISBN: ' || p_ISBN || ' ya existe.');51     WHEN OTHERS THEN  
52       RAISE_APPLICATION_ERROR(-20006, 'Error inesperado: ' || SQLERRM);  
53   END;  
54  
55   DBMS_OUTPUT.PUT_LINE('Libro agregado correctamente en la sucursal.');56 END;  
57 /  
  
Procedure created.
```

-- Bloque PL/SQL para solicitar datos e invocar el procedimiento

DECLARE

v_ISBN VARCHAR2(20);

v_Titulo VARCHAR2(255);

v_Genero VARCHAR2(100);

v_ID_Editorial NUMBER;

v_Fecha_Publicacion DATE;

v_Precio NUMBER;

v_Stock NUMBER;

v_ID_Sucursal NUMBER;

v_Nombre_Autor VARCHAR2(100);

BEGIN

-- Solicitar datos al usuario

v_ISBN := '&Ingreso_ISBN';

v_Titulo := '&Ingreso_Titulo';

v_Genero := '&Ingreso_Genero';

v_ID_Editorial := &Ingreso_ID_Editorial;

v_Fecha_Publicacion := TO_DATE('&Ingreso_Fecha_Publicacion', 'YYYY/MM/DD');

v_Precio := &Ingreso_Precio;

v_Stock := &Ingreso_Stock;

v_ID_Sucursal := &Ingreso_ID_Sucursal;

v_Nombre_Autor := '&Ingreso_Nombre_Autor';

-- Llamar al procedimiento para insertar el libro en la sucursal

InsertarLibroEnSucursal(

p_ISBN => v_ISBN,

p_Titulo => v_Titulo,

p_Genero => v_Genero,

p_ID_Editorial => v_ID_Editorial,

p_Fecha_Publicacion => v_Fecha_Publicacion,

p_Precio => v_Precio,

p_Stock => v_Stock,

p_ID_Sucursal => v_ID_Sucursal,

p_Nombre_Autor => v_Nombre_Autor

);

COMMIT; -- Confirmar los cambios en la base de datos

DBMS_OUTPUT.PUT_LINE('Libro agregado correctamente en la sucursal.');

END;

/

```
SQL> BEGIN
2   v_ISBN VARCHAR(20);
3   v_Titulo VARCHAR(200);
4   v_Autor VARCHAR(100);
5   v_ID_Editorial NUMBER;
6   v_Fecha_Publicacion DATE;
7   v_Precio NUMBER;
8   v_Stock NUMBER;
9   v_ID_Sucursal NUMBER;
10
11 -- Solicitar datos al usuario
12 v_ISBN := <Ingrese ISBN>;
13 v_Titulo := <Ingrese Titulo>;
14 v_Autor := <Ingrese Autor>;
15 v_ID_Editorial := <Ingrese ID Editorial>;
16 v_Fecha_Publicacion := TO_DATE(<Ingrese Fecha Publicacion>, 'YYYYMMDD');
17 v_Precio := <Ingrese Precio>;
18 v_Stock := <Ingrese Stock>;
19 v_ID_Sucursal := <Ingrese ID Sucursal>;
20
21 -- Llamar al procedimiento para agregar el libro en la sucursal
22 InsertarLibroEnSucursal(v_ISBN, v_Titulo, v_Autor, v_ID_Editorial, v_Fecha_Publicacion, v_Precio, v_Stock, v_ID_Sucursal);
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512

```

4. **Buscar libros disponibles en sucursal por nombre de autor** : Desarrollar un procedimiento para buscar libros en una sucursal específica y mostrar disponibilidad, precio, stock y autor.

```
SET SERVEROUTPUT ON;
```

```
--procedimiento almacenado para la búsqueda de libros
```

```
CREATE OR REPLACE PROCEDURE BuscarLibroEnSucursal (
```

```
    p_Titulo IN VARCHAR2,
```

```
    p_Ciudad IN VARCHAR2,
```

```
    p_Provincia IN VARCHAR2,
```

```
    p_Calle IN VARCHAR2
```

```
)
```

```
AS
```

```
    v_ID_Sucursal NUMBER;
```

```
    v_Precio DECIMAL(10, 2);
```

```
    v_Stock NUMBER;
```

```
    v_Nombre_Autor VARCHAR2(100);
```

```
BEGIN
```

```
-- Verificar si la sucursal existe y obtener su ID
```

```
BEGIN
```

```
    SELECT ID_Sucursal
```

```
    INTO v_ID_Sucursal
```

```
    FROM Sucursal
```

```
    WHERE UPPER(Ciudad_sucu) = UPPER(p_Ciudad)
```

```
        AND UPPER(Provincia_sucu) = UPPER(p_Provincia)
```

```
        AND UPPER(Calle_sucu) = UPPER(p_Calle);
```

```
EXCEPTION
```

```
    WHEN NO_DATA_FOUND THEN
```

```
        RAISE_APPLICATION_ERROR(-20001, 'No se encontró la sucursal con los datos proporcionados.');
```

```
    WHEN OTHERS THEN
```

```
        RAISE_APPLICATION_ERROR(-20002, 'Error al buscar la sucursal: ' || SQLERRM);
```

```
END;
```

```
-- Buscar el libro en la sucursal
```

```
BEGIN
```

```
    SELECT Precio, Stock, Nombre_Autor
```

```
    INTO v_Precio, v_Stock, v_Nombre_Autor
```

```
    FROM Libros
```

```
    WHERE Titulo = p_Titulo
```

```
    AND ID_Sucursal = v_ID_Sucursal;
```

```
    IF v_Stock > 0 THEN
```

```
        DBMS_OUTPUT.PUT_LINE('El libro con título: ' || p_Titulo || ' está disponible en la sucursal: ' || p_Ciudad || ', ' || p_Provincia || ', ' || p_Calle);
```

```
        DBMS_OUTPUT.PUT_LINE('Autor: ' || v_Nombre_Autor);
```

```
        DBMS_OUTPUT.PUT_LINE('Precio: ' || v_Precio || ', Stock: ' || v_Stock);
```

```
    ELSE
```

```
        DBMS_OUTPUT.PUT_LINE('El libro con título: ' || p_Titulo || ' está en la sucursal, pero no hay stock disponible.');
```

```
    END IF;
```

```
EXCEPTION
```

```
    WHEN NO_DATA_FOUND THEN
```

```
        DBMS_OUTPUT.PUT_LINE('El libro con título: ' || p_Titulo || ' no está disponible en la sucursal.');
```

```
    WHEN OTHERS THEN
```

```
        RAISE_APPLICATION_ERROR(-20003, 'Error al buscar el libro en la sucursal: ' ||  
SQLERRM);
```

```
    END;
```

```
END;
```

```
/
```

```
SQL> CREATE OR REPLACE PROCEDURE BuscarLibroEnSucursal (  
2   p_Titulo IN VARCHAR2,  
3   p_Ciudad IN VARCHAR2,  
4   p_Provincia IN VARCHAR2,  
5   p_Calle IN VARCHAR2  
6 )  
7 AS  
8   v_ID_Sucursal NUMBER;  
9   v_Count NUMBER;  
10  v_Precio DECIMAL(10, 2);  
11  v_Stock NUMBER;  
12 BEGIN  
13  -- Verificar si la sucursal existe y obtener su ID  
14  BEGIN  
15    SELECT ID_Sucursal  
16    INTO v_ID_Sucursal  
17    FROM Sucursal  
18    WHERE UPPER(Ciudad_sucu) = UPPER(p_Ciudad)  
19    AND UPPER(Provincia_sucu) = UPPER(p_Provincia)  
20    AND UPPER(Calle_sucu) = UPPER(p_Calle);  
21  
22    IF v_ID_Sucursal IS NULL THEN  
23      RAISE_APPLICATION_ERROR(-20001, 'La sucursal con ciudad: ' || p_Ciudad || ', provincia: ' || p_Provincia || ' y calle: ' || p_Calle || ' no existe.');
```

--solicitud de datos al usuario

DECLARE

v_Titulo VARCHAR2(255);

v_Ciudad VARCHAR2(100);

v_Provincia VARCHAR2(100);

v_Calle VARCHAR2(255);

BEGIN

-- Solicitar datos al usuario

v_Titulo := '&Ingreso_Titulo';

v_Ciudad := '&Ingreso_Ciudad';

v_Provincia := '&Ingrese_Provincia';

v_Calle := '&Ingrese_Calle';

-- Llamar al procedimiento para buscar el libro en la sucursal

BuscarLibroEnSucursal(

p_Titulo => v_Titulo,

p_Ciudad => v_Ciudad,

p_Provincia => v_Provincia,

p_Calle => v_Calle

);

END;

/

```
SQL> DECLARE
2   v_Titulo      VARCHAR2(255);
3   v_Ciudad      VARCHAR2(100);
4   v_Provincia   VARCHAR2(100);
5   v_Calle       VARCHAR2(255);
6 BEGIN
7   -- Solicitar datos al usuario
8   v_Titulo := '&Ingrese_Titulo';
9   v_Ciudad := '&Ingrese_Ciudad';
10  v_Provincia := '&Ingrese_Provincia';
11  v_Calle := '&Ingrese_Calle';
12
13  -- Llamar al procedimiento para buscar el libro en la sucursal
14  BuscarLibroEnSucursal(
15    p_Titulo => v_Titulo,
16    p_Ciudad => v_Ciudad,
17    p_Provincia => v_Provincia,
18    p_Calle => v_Calle
19  );
20 END;
21 /
Enter value for ingrese_titulo: fisica
old 8:   v_Titulo := '&Ingrese_Titulo';
new 8:   v_Titulo := 'fisica';
Enter value for ingrese_ciudad: panama
old 9:   v_Ciudad := '&Ingrese_Ciudad';
new 9:   v_Ciudad := 'panama';
Enter value for ingrese_provincia: panama
old 10:  v_Provincia := '&Ingrese_Provincia';
new 10:  v_Provincia := 'panama';
Enter value for ingrese_calle: 15
old 11:  v_Calle := '&Ingrese_Calle';
new 11:  v_Calle := '15';
El libro con título: fisica está disponible en la sucursal: panama, panama, 15
Precio: 50, Stock: 30
PL/SQL procedure successfully completed.
```

5. Desplegar lista de libros de un autor : Crear un procedimiento para listar libros de un autor específico, mostrando detalles como ISBN, título, género y stock.

```
SET SERVEROUTPUT ON;
```

```
CREATE OR REPLACE PROCEDURE MostrarLibrosPorAutor (
```

```
    p_Nombre_Autor IN VARCHAR2
```

```
)
```

```
AS
```

```
BEGIN
```

```
    -- Consultar la lista de libros del autor especificado
```

```
    FOR r IN (
```

```
        SELECT ISBN, Titulo, Genero, Fecha_Publicacion, Precio, Stock, ID_Sucursal
```

```
        FROM Libros
```

```
        WHERE UPPER(Nombre_Autor) = UPPER(p_Nombre_Autor)
```

```
    ) LOOP
```

```
        DBMS_OUTPUT.PUT_LINE('ISBN: ' || r.ISBN ||
```

```
                                ', Titulo: ' || r.Titulo ||
```

```
                                ', Genero: ' || r.Genero ||
```

```
                                ', Fecha Publicacion: ' || TO_CHAR(r.Fecha_Publicacion, 'YYYY-MM-DD') ||
```

```
                                ', Precio: ' || r.Precio ||
```

```
                                ', Stock: ' || r.Stock ||
```

```
                                ', ID Sucursal: ' || r.ID_Sucursal);
```

```
    END LOOP;
```

```
    -- Verificar si se encontraron libros
```

```
    IF SQL%ROWCOUNT = 0 THEN
```



```
        DBMS_OUTPUT.PUT_LINE('No se encontraron libros para el autor: ' ||  
p_Nombre_Autor);
```

```
    END IF;
```

```
EXCEPTION
```

```
    WHEN OTHERS THEN
```

```
        DBMS_OUTPUT.PUT_LINE('Error inesperado: ' || SQLERRM);
```

```
END;
```

```
/
```

```
SQL> CREATE OR REPLACE PROCEDURE MostrarLibrosPorAutor (  
2     p_Nombre_Autor IN VARCHAR2  
3 )  
4 AS  
5 BEGIN  
6     -- Consultar la lista de libros del autor especificado  
7     FOR r IN (  
8         SELECT ISBN, Titulo, Genero, Fecha_Publicacion, Precio, Stock, ID_Sucursal  
9         FROM Libros  
10        WHERE UPPER(Nombre_Autor) = UPPER(p_Nombre_Autor)  
11    ) LOOP  
12        DBMS_OUTPUT.PUT_LINE('ISBN: ' || r.ISBN ||  
13                               ', Titulo: ' || r.Titulo ||  
14                               ', Genero: ' || r.Genero ||  
15                               ', Fecha Publicacion: ' || TO_CHAR(r.Fecha_Publicacion, 'YYYY-MM-DD') ||  
16                               ', Precio: ' || r.Precio ||  
17                               ', Stock: ' || r.Stock ||  
18                               ', ID Sucursal: ' || r.ID_Sucursal);  
19    END LOOP;  
20  
21    -- Verificar si se encontraron libros  
22    IF SQL%ROWCOUNT = 0 THEN  
23        DBMS_OUTPUT.PUT_LINE('No se encontraron libros para el autor: ' || p_Nombre_Autor);  
24    END IF;  
25 EXCEPTION  
26     WHEN OTHERS THEN  
27         DBMS_OUTPUT.PUT_LINE('Error inesperado: ' || SQLERRM);  
28 END;  
29 /  
  
Procedure created.
```

--solicitud de datos al usuario

```
DECLARE
```

```
    v_Nombre_Autor VARCHAR2(100);
```

```
BEGIN
```

```
    -- Solicitar el nombre del autor al usuario
```

```
    v_Nombre_Autor := '&Ingrese_Nombre_Autor';
```

```
    -- Llamar al procedimiento para mostrar los libros del autor
```

```
MostrarLibrosPorAutor(v_Nombre_Autor);
```

```
END;
```

```
/
```

```
SQL> DECLARE
2   v_Nombre_Autor VARCHAR2(100);
3 BEGIN
4   -- Solicitar el nombre del autor al usuario
5   v_Nombre_Autor := '&Ingreso_Nombre_Autor';
6
7   -- Llamar al procedimiento para mostrar los libros del autor
8   MostrarLibrosPorAutor(v_Nombre_Autor);
9 END;
10 /
Enter value for ingreso_nombre_autor: enrique
old 5:   v_Nombre_Autor := '&Ingreso_Nombre_Autor';
new 5:   v_Nombre_Autor := 'enrique';
ISBN: 6789, Titulo: database, Genero: educacion, Fecha Publicacion: 2007-03-03, Precio: 60, Stock: 12, ID Sucursal: 4
ISBN: 4567, Titulo: cafe, Genero: CULINARIO, Fecha Publicacion: 2006-08-04, Precio: 34, Stock: 40, ID Sucursal: 4
PL/SQL procedure successfully completed.
```

- 6. Aplicar Descuentos en Facturas:** Se configurará la base de datos para aplicar descuentos automáticamente en las facturas. Esto incluye ajustar el procedimiento de generación de facturas para aplicar un descuento del 20% a clientes miembros de la librería, un 10% a clientes mayores de 55 años, o no aplicar descuento si no se cumplen estas condiciones.

```
CREATE OR REPLACE FUNCTION AplicarDescuento(
```

```
    p_id_cliente IN INT,
```

```
    p_monto_original IN DECIMAL
```

```
) RETURN DECIMAL IS
```

```
    v_descuento DECIMAL(10, 2) := 0;
```

```
    v_es_socio NUMBER(1);
```

```
    v_fecha_nacimiento DATE;
```

```
    v_edad INT;
```

```
    v_fecha_actual DATE := SYSDATE;
```

```
BEGIN
```

-- Obtener si el cliente es socio y su fecha de nacimiento

```
SELECT Es_Socio, Fecha_Nacimiento INTO v_es_socio, v_fecha_nacimiento  
FROM Clientes  
WHERE ID_Cliente = p_id_cliente;
```

-- Calcular la edad del cliente

```
v_edad := TRUNC(MONTHS_BETWEEN(v_fecha_actual, v_fecha_nacimiento) /  
12);
```

-- Aplicar el descuento correspondiente (solo uno)

```
IF v_es_socio = 1 THEN  
    v_descuento := 0.20;  
ELSIF v_edad > 55 THEN  
    v_descuento := 0.10;  
ELSE  
    v_descuento := 0;  
END IF;
```

-- Calcular el monto final con el descuento aplicado

```
RETURN p_monto_original * (1 - v_descuento);  
  
END;
```

/

7. **Generar factura de la compra:** Desarrollar un procedimiento para registrar compras, calcular totales con descuentos y actualizar inventario. Incluir manejo de errores y mensajes de estado.

```
CREATE OR REPLACE TYPE VARCHAR2_LIST AS TABLE OF VARCHAR2(20);
```

/

CREATE OR REPLACE TYPE NUMBER_LIST AS TABLE OF NUMBER;

/

CREATE SEQUENCE Ventas_SEQ

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE OR REPLACE PROCEDURE generar_factura_compra (

p_id_cliente IN INT,

p_id_sucursal IN INT,

p_isbn_list IN VARCHAR2_LIST,

p_cantidad_list IN NUMBER_LIST

) IS

v_total_sin_descuento DECIMAL(10, 2) := 0;

v_descuento DECIMAL(10, 2) := 0;

v_total_con_descuento DECIMAL(10, 2) := 0;

v_venta_id INT;

BEGIN

-- Calcular el total sin descuento

FOR i IN 1 .. p_isbn_list.COUNT LOOP

DECLARE

v_precio_libro DECIMAL(10, 2);

BEGIN

SELECT Precio

INTO v_precio_libro

FROM Libros

WHERE ISBN = p_isbn_list(i);

v_total_sin_descuento := v_total_sin_descuento + (v_precio_libro *
p_cantidad_list(i));

EXCEPTION

WHEN NO_DATA_FOUND THEN

DBMS_OUTPUT.PUT_LINE('Error: El libro con ISBN ' || p_isbn_list(i) || ' no
fue encontrado.');

RETURN;

END;

END LOOP;

-- Aplicar descuento utilizando la función AplicarDescuento

v_total_con_descuento := AplicarDescuento(p_id_cliente, v_total_sin_descuento);

-- Insertar la venta en la tabla Ventas

INSERT INTO Ventas (ID_Venta, ID_Cliente, ID_Sucursal, Total)

VALUES (Ventas_SEQ.NEXTVAL, p_id_cliente, p_id_sucursal, v_total_con_descuento)

RETURNING ID_Venta INTO v_venta_id;

-- Insertar detalles de la venta en la tabla Detalle_Venta

```
FOR i IN 1 .. p_isbn_list.COUNT LOOP
    INSERT INTO Detalle_Venta (ID_Venta, ISBN, Cantidad)
    VALUES (v_venta_id, p_isbn_list(i), p_cantidad_list(i));
END LOOP;
```

-- Mostrar mensaje de éxito

```
DBMS_OUTPUT.PUT_LINE('Compra registrada exitosamente. Total con descuento: '
|| v_total_con_descuento);
```

COMMIT;

EXCEPTION

WHEN OTHERS THEN

```
DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
```

ROLLBACK;

END;

/

```
SQL> CREATE OR REPLACE PROCEDURE generar_factura_compra (
  2   p_id_cliente IN INT,
  3   p_id_sucursal IN INT,
  4   p_isbn_list IN VARCHAR2_LIST,
  5   p_cantidad_list IN NUMBER_LIST
  6 ) IS
  7   v_total_sin_descuento DECIMAL(18, 2) := 0;
  8   v_descuento DECIMAL(18, 2) := 0;
  9   v_total_con_descuento DECIMAL(18, 2) := 0;
10   v_venta_id INT;
11 BEGIN
12   -- Calcular el total sin descuento
13   FOR i IN 1 .. p_isbn_list.COUNT LOOP
14     DECLARE
15       vPrecioLibro DECIMAL(18, 2);
16     BEGIN
17       SELECT Precio
18       INTO vPrecioLibro
19       FROM Libros
20       WHERE ISBN = p_isbn_list(i);
21       v_total_sin_descuento := v_total_sin_descuento + (vPrecioLibro * p_cantidad_list(i));
22     EXCEPTION
23       WHEN NO_DATA_FOUND THEN
24         DBMS_OUTPUT.PUT_LINE('Error: El libro con ISBN ' || p_isbn_list(i) || ' no fue encontrado.');
```

CREATE OR REPLACE TRIGGER trg_upd_inv_post_venta

AFTER INSERT ON Detalle_Venta

FOR EACH ROW

BEGIN

UPDATE Inventario

SET Cantidad = Cantidad - :NEW.Cantidad

WHERE ISBN = :NEW.ISBN

AND ID_Sucursal = (SELECT ID_Sucursal FROM Ventas WHERE ID_Venta = :NEW.ID_Venta);

END;

/

```
SQL> CREATE OR REPLACE TRIGGER trg_upd_inv_post_venta
2  AFTER INSERT ON Detalle_Venta
3  FOR EACH ROW
4  BEGIN
5      UPDATE Inventario
6      SET Cantidad = Cantidad - :NEW.Cantidad
7      WHERE ISBN = :NEW.ISBN
8      AND ID_Sucursal = (SELECT ID_Sucursal FROM Ventas WHERE ID_Venta = :NEW.ID_Venta);
9  END;
10 /
Trigger created.
```

DECLARE

v_id_cliente INT := 1; -- Ejemplo de ID de cliente, puedes cambiarlo según sea necesario

v_id_sucursal INT;

v_isbn_list VARCHAR2_LIST := VARCHAR2_LIST();

v_cantidad_list NUMBER_LIST := NUMBER_LIST();

v_isbn VARCHAR2(20);

v_cantidad NUMBER;

v_mas_libros CHAR(1);

v_venta_id INT;

v_total_venta DECIMAL(10, 2);

BEGIN

-- Solicitar el ID de la sucursal al usuario

DBMS_OUTPUT.PUT_LINE('Ingrese el ID de la sucursal:');

v_id_sucursal := &v_id_sucursal;

-- Solicitar detalles de los libros

LOOP

DBMS_OUTPUT.PUT_LINE('Ingrese el ISBN del libro:');

v_isbn := '&v_isbn';

v_isbn_list.EXTEND;

v_isbn_list(v_isbn_list.COUNT) := v_isbn;

DBMS_OUTPUT.PUT_LINE('Ingrese la cantidad del libro:');

v_cantidad := '&v_cantidad';

v_cantidad_list.EXTEND;

v_cantidad_list(v_cantidad_list.COUNT) := v_cantidad;

DBMS_OUTPUT.PUT_LINE('¿Desea agregar otro libro? (S/N):');

v_mas_libros := '&v_mas_libros';

EXIT WHEN UPPER(v_mas_libros) != 'S';

END LOOP;

-- Llamar al procedimiento para generar la factura

generar_factura_compra(v_id_cliente, v_id_sucursal, v_isbn_list, v_cantidad_list);

-- Obtener el ID de la última venta insertada

SELECT MAX(ID_Venta)

INTO v_venta_id

FROM Ventas;

-- Mostrar los detalles de la venta

```
DBMS_OUTPUT.PUT_LINE('Factura de la compra:');

DBMS_OUTPUT.PUT_LINE('-----');

DBMS_OUTPUT.PUT_LINE('Fecha: ' || TO_CHAR(SYSDATE, 'DD-MON-YYYY HH24:MI:SS'));

DBMS_OUTPUT.PUT_LINE('ID Venta: ' || v_venta_id);

DBMS_OUTPUT.PUT_LINE('ID Cliente: ' || v_id_cliente);

DBMS_OUTPUT.PUT_LINE('ID Sucursal: ' || v_id_sucursal);

DBMS_OUTPUT.PUT_LINE('Detalles de la compra:');

DBMS_OUTPUT.PUT_LINE('ISBN    | Cantidad | Precio unitario | Subtotal');

DBMS_OUTPUT.PUT_LINE('-----');
```

-- Calcular subtotal por libro y mostrar detalles

```
FOR r IN (

    SELECT dv.ISBN, dv.Cantidad, l.Precio, (dv.Cantidad * l.Precio) AS Subtotal

    FROM Detalle_Venta dv

    JOIN Libros l ON dv.ISBN = l.ISBN

    WHERE dv.ID_Venta = v_venta_id

) LOOP

    DBMS_OUTPUT.PUT_LINE(r.ISBN || ' ' || r.Cantidad || ' ' || r.Precio || ' ' || r.Subtotal);

    v_total_venta := v_total_venta + r.Subtotal;

END LOOP;

DBMS_OUTPUT.PUT_LINE('-----');

DBMS_OUTPUT.PUT_LINE('Total con descuento: ' || v_total_venta);

END;
```


/

```
16 -- Solicitar detalles de los libros
17 LOOP
18   DEMS_OUTPUT.PUT_LINE('Ingrese el ISBN del libro:');
19   v_isbn := 'dv_isbn';
20   v_isbn_list.EXTEND;
21   v_isbn_list(v_isbn_list.COUNT) := v_isbn;
22
23   DEMS_OUTPUT.PUT_LINE('Ingrese la cantidad del libro:');
24   v_cantidad := 'dv_cantidad';
25   v_cantidad_list.EXTEND;
26   v_cantidad_list(v_cantidad_list.COUNT) := v_cantidad;
27
28   DEMS_OUTPUT.PUT_LINE('¿Desea agregar otro libro? (S/N):');
29   v_mas_libros := 'dv_mas_libros';
30
31   EXIT WHEN UPPER(v_mas_libros) != 'S';
32 END LOOP;
33
34 -- Llamar al procedimiento para generar la factura
35 generar_factura_compra(v_id_cliente, v_id_sucursal, v_isbn_list, v_cantidad_list);
36
37 -- Obtener el ID de la última venta insertada
38 SELECT MAX(ID_Venta)
39 INTO v_venta_id
40 FROM Ventas;
41
42 -- Mostrar los detalles de la venta
43 DEMS_OUTPUT.PUT_LINE('Factura de la compra:');
44 DEMS_OUTPUT.PUT_LINE('-----');
45 DEMS_OUTPUT.PUT_LINE('Fecha: ' || TO_CHAR(SYSDATE, 'DD-MON-YYYY HH24:MI:SS'));
46 DEMS_OUTPUT.PUT_LINE('ID Venta: ' || v_venta_id);
47 DEMS_OUTPUT.PUT_LINE('ID Cliente: ' || v_id_cliente);
48 DEMS_OUTPUT.PUT_LINE('ID Sucursal: ' || v_id_sucursal);
49 DEMS_OUTPUT.PUT_LINE('Detalles de la compra:');
50 DEMS_OUTPUT.PUT_LINE('ISBN | Cantidad | Precio unitario | Subtotal');
51 DEMS_OUTPUT.PUT_LINE('-----');
52
53 -- Calcular subtotal por libro y mostrar detalles
54 FOR r IN (
55   SELECT dv.ISBN, dv.Cantidad, l.Precio, (dv.Cantidad * l.Precio) AS Subtotal
56   FROM Detalle_Venta dv
57   JOIN Libros l ON dv.ISBN = l.ISBN
58   WHERE dv.ID_Venta = v_venta_id
59 ) LOOP
60   DEMS_OUTPUT.PUT_LINE(r.ISBN || ' | ' || r.Cantidad || ' | ' || r.Precio || ' | ' || r.Subtotal);
61   v_total_venta := v_total_venta + r.Subtotal;
62 END LOOP;
63
64 DEMS_OUTPUT.PUT_LINE('-----');
65 DEMS_OUTPUT.PUT_LINE('Total con descuento: ' || v_total_venta);
66
67 END;
68 /
Enter value for v_id_sucursal: 4567
old 14: v_id_sucursal := 'dv_id_sucursal';
new 14: v_id_sucursal := 4567;
Enter value for v_isbn: 654
old 19: v_isbn := 'dv_isbn';
new 19: v_isbn := '654';
Enter value for v_cantidad: 5
old 24: v_cantidad := 'dv_cantidad';
new 24: v_cantidad := '5';
Enter value for v_mas_libros: n
old 29: v_mas_libros := 'dv_mas_libros';
new 29: v_mas_libros := 'n';
Ingreso el ID de la sucursal:
Ingreso el ISBN del libro:
Ingreso la cantidad del libro:
¿Desea agregar otro libro? (S/N):
Error: El libro con ISBN 654 no fue encontrado.
Factura de la compra:
-----
Fecha: 02-JUL-2024 18:29:24
ID Venta: 1
ID Cliente: 1
ID Sucursal: 4567
Detalles de la compra:
ISBN | Cantidad | Precio unitario | Subtotal
-----
Total con descuento:
PL/SQL procedure successfully completed.
```

8. **Registrar ventas diarias:** Crear un procedimiento para registrar automáticamente el total de ventas diarias por sucursal en la tabla VentasDiarias.

CREATE SEQUENCE VENTASDIARIAS_SEQ

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE OR REPLACE PROCEDURE registrar_ventas_diarias (

p_id_sucursal IN INT,

p_fecha IN DATE,

```

    p_id_ventadiaria OUT INT
) IS
    v_total_ventas_diarias DECIMAL(10, 2) := 0;

BEGIN

    -- Calcular el total de ventas para la sucursal y fecha dadas

    SELECT NVL(SUM(Total), 0)

    INTO v_total_ventas_diarias

    FROM Ventas

    WHERE ID_Sucursal = p_id_sucursal

        AND TRUNC(Fecha_Venta) = TRUNC(p_fecha);

    -- Insertar el registro en VentasDiarias

    INSERT INTO VentasDiarias (ID_VentaDiaria, ID_Sucursal, Fecha, Total)

    VALUES      (VENTASDIARIAS_SEQ.NEXTVAL,      p_id_sucursal,      p_fecha,
v_total_ventas_diarias)

    RETURNING ID_VentaDiaria INTO p_id_ventadiaria;

    -- Mostrar mensaje de éxito

    DBMS_OUTPUT.PUT_LINE('Ventas diarias registradas correctamente para la
sucursal ' || p_id_sucursal ||

        ' en la fecha ' || TO_CHAR(p_fecha, 'DD-MON-YYYY'));

    COMMIT;

EXCEPTION

    WHEN OTHERS THEN

        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);

        ROLLBACK;

```

END registrar_ventas_diarias;

/

```
11> CREATE OR REPLACE PROCEDURE registrar_ventas_diarias (  
2  p_id_sucursal IN INT,  
3  p_fecha IN DATE  
4  ) IS  
5  v_total_ventas_diarias DECIMAL(10, 2) := 0;  
6  BEGIN  
7      -- Calcular el total de ventas para la sucursal y fecha dadas  
8      SELECT NVL(SUM(Total), 0)  
9          INTO v_total_ventas_diarias  
10         FROM Ventas  
11        WHERE ID_Sucursal = p_id_sucursal  
12              AND TRUNC(Fecha_Venta) = TRUNC(p_fecha);  
13  
14      -- Insertar el registro en VentasDiarias  
15      INSERT INTO VentasDiarias (ID_Sucursal, Fecha, Total)  
16      VALUES (p_id_sucursal, p_fecha, v_total_ventas_diarias);  
17  
18      -- Mostrar mensaje de éxito  
19      DBMS_OUTPUT.PUT_LINE('Ventas diarias registradas correctamente para la sucursal ' || p_id_sucursal ||  
20                          ' en la fecha ' || TO_CHAR(p_fecha, 'DD-MON-YYYY'));  
21  
22      COMMIT;  
23  EXCEPTION  
24      WHEN OTHERS THEN  
25          DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);  
26          ROLLBACK;  
27  END registrar_ventas_diarias;  
28  /  
  
procedure created.
```

---Bloque Anonimo para ingresar sucursal para saber sus ventas diarias

SET SERVEROUTPUT ON;

DECLARE

v_id_sucursal INT;

v_fecha DATE;

v_id_ventadiaria INT;

BEGIN

-- Solicitar el ID de la sucursal

DBMS_OUTPUT.PUT_LINE('Ingrese el ID de la sucursal:');

v_id_sucursal := &v_id_sucursal;

-- Solicitar la fecha

DBMS_OUTPUT.PUT_LINE('Ingrese la fecha (DD/MM/YYYY):');

v_fecha := TO_DATE('&v_fecha', 'DD/MM/YYYY');

-- Ejecutar el procedimiento para registrar las ventas diarias

```
    registrar_ventas_diarias(p_id_sucursal => v_id_sucursal, p_fecha => v_fecha,  
p_id_ventadiaria => v_id_ventadiaria);
```

-- Mostrar mensaje de éxito con el ID de la venta diaria registrada

```
    DBMS_OUTPUT.PUT_LINE('Proceso completado exitosamente. ID de venta diaria:  
' || v_id_ventadiaria);
```

```
EXCEPTION
```

```
    WHEN OTHERS THEN
```

```
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
```

```
        ROLLBACK;
```

```
END;
```

```
/
```

```
SQL> ---Bloque Anonimo para ingresar sucursal para saber sus ventas diarias  
SQL>  
SQL> SET SERVEROUTPUT ON;  
SQL>  
SQL>  
SQL>  
SQL> DECLARE  
2  
3     v_id_sucursal INT;  
4  
5     v_fecha DATE;  
6  
7 BEGIN  
8  
9     -- Solicitar el ID de la sucursal  
10    DBMS_OUTPUT.PUT_LINE('Ingrese el ID de la sucursal:');  
11  
12    v_id_sucursal := &v_id_sucursal;  
13  
14    -- Solicitar la fecha  
15    DBMS_OUTPUT.PUT_LINE('Ingrese la fecha (DD-MON-YYYY):');  
16  
17    v_fecha := TO_DATE('&v_fecha', 'DD-MON-YYYY');  
18  
19  
20  
21  
22  
23    -- Ejecutar el procedimiento  
24    registrar_ventas_diarias(p_id_sucursal => v_id_sucursal, p_fecha => v_fecha);  
25  
26  
27 END;  
28  
29 /  
Enter value for v_id_sucursal: 76543  
old 13:     v_id_sucursal := &v_id_sucursal;  
new 13:     v_id_sucursal := 76543;  
Enter value for v_fecha: 2/5/1992
```

9. **Vistas para consultar ventas diarias:** Crear una vista para consultar un resumen de ventas diarias por sucursal y fecha, agrupando y ordenando la información

--Se crea la vista Ventas Diarias

CREATE OR REPLACE VIEW Resumen_Ventas_Diarias AS

SELECT

ID_Sucursal AS "ID de Sucursal",

Fecha AS "Fecha de Venta",

SUM(Total) AS "Total de Ventas Diarias"

FROM

VentasDiarias

GROUP BY

ID_Sucursal, Fecha

ORDER BY

ID_Sucursal, Fecha;

/

--Se consulta la vista SELECT * FROM Resumen_Ventas_Diarias;

```
SQL>
SQL> --Se crea la vista Ventas Diarias
SQL> CREATE OR REPLACE VIEW Resumen_Ventas_Diarias AS
2  SELECT
3      ID_Sucursal AS "ID de Sucursal",
4      Fecha AS "Fecha de Venta",
5      SUM(Total) AS "Total de Ventas Diarias"
6  FROM
7      VentasDiarias
8  GROUP BY
9      ID_Sucursal, Fecha
10 ORDER BY
11     ID_Sucursal, Fecha;

View created.

SQL> /

View created.

SQL>
SQL>
SQL> --Se consulta la vista
SQL> SELECT * FROM Resumen_Ventas_Diarias;

ID de Sucursal Fecha de Total de Ventas Diarias
-----
1 03-JUL-24 2800
2 02-JUL-24 1750
```

10. Generar Informe de Ventas por Sucursal: Desarrollar un procedimiento para generar un informe detallado de ventas por sucursal, incluyendo detalles de cada venta y libros vendidos.

--crear vista de auditoria

CREATE OR REPLACE VIEW VentasAuditoria AS

SELECT

v.ID_Venta,
v.ID_Sucursal,
v.Fecha_Venta,
v.Total AS Total_Venta,
d.ISBN,
d.Cantidad,
l.Titulo,
l.Precio

FROM

Ventas v

JOIN

Detalle_Venta d ON v.ID_Venta = d.ID_Venta

JOIN

Libros l ON d.ISBN = l.ISBN;

--crear procedimiento almacenado para la consulta de auditoria de ventas por sucursal

CREATE OR REPLACE PROCEDURE GenerarInformeVentas (

p_id_sucursal IN NUMBER,

p_fecha_inicio IN DATE,

p_fecha_fin IN DATE

) AS

BEGIN

DBMS_OUTPUT.PUT_LINE('Información de la Sucursal:');

DBMS_OUTPUT.PUT_LINE('ID Sucursal: ' || p_id_sucursal);

-- Información de la sucursal

FOR r IN (SELECT Ciudad_sucu, Provincia_sucu, Calle_sucu, Telefono_sucu, Email_sucu

```

FROM Sucursal

WHERE ID_Sucursal = p_id_sucursal) LOOP

DBMS_OUTPUT.PUT_LINE('Ciudad: ' || r.Ciudad_sucu);
DBMS_OUTPUT.PUT_LINE('Provincia: ' || r.Provincia_sucu);
DBMS_OUTPUT.PUT_LINE('Calle: ' || r.Calle_sucu);
DBMS_OUTPUT.PUT_LINE('Teléfono: ' || r.Telefono_sucu);
DBMS_OUTPUT.PUT_LINE('Email: ' || r.Email_sucu);

END LOOP;

```

```

DBMS_OUTPUT.PUT_LINE('Resumen de Ventas Diarias:');

```

```

-- Resumen de ventas diarias

```

```

FOR r IN (SELECT Fecha, SUM(Total) AS Total_Diario
FROM VentasDiarias
WHERE ID_Sucursal = p_id_sucursal
AND Fecha BETWEEN p_fecha_inicio AND p_fecha_fin
GROUP BY Fecha) LOOP

DBMS_OUTPUT.PUT_LINE('Fecha: ' || TO_CHAR(r.Fecha, 'DD-MON-YY') || ', Total Diario: '
|| r.Total_Diario);

END LOOP;

```

```

-- Detalle de ventas

```

```

DBMS_OUTPUT.PUT_LINE('Detalles de Ventas:');

FOR r IN (SELECT *
FROM VentasAuditoria
WHERE ID_Sucursal = p_id_sucursal
AND Fecha_Venta BETWEEN p_fecha_inicio AND p_fecha_fin) LOOP

DBMS_OUTPUT.PUT_LINE('ID Venta: ' || r.ID_Venta || ', Fecha: ' ||
TO_CHAR(r.Fecha_Venta, 'DD-MON-YY') ||

', ISBN: ' || r.ISBN || ', Título: ' || r.Titulo ||

', Cantidad: ' || r.Cantidad || ', Precio: ' || r.Precio ||

', Total: ' || r.Total_Venta);

```

END LOOP;

END;

```
SQL> CREATE OR REPLACE PROCEDURE generarInformeVentas (
2   p_id_sucursal IN NUMBER,
3   p_fecha_inicio IN DATE,
4   p_fecha_fin IN DATE
5 ) AS
6 BEGIN
7   DBMS_OUTPUT.PUT_LINE('Información de la Sucursal:');
8   DBMS_OUTPUT.PUT_LINE('ID Sucursal: ' || p_id_sucursal);
9
10  -- Información de la sucursal
11  FOR r IN (SELECT Ciudad_sucu, Provincia_sucu, Calle_sucu, Telefono_sucu, Email_sucu
12            FROM Sucursal
13            WHERE ID_Sucursal = p_id_sucursal) LOOP
14    DBMS_OUTPUT.PUT_LINE('Ciudad: ' || r.Ciudad_sucu);
15    DBMS_OUTPUT.PUT_LINE('Provincia: ' || r.Provincia_sucu);
16    DBMS_OUTPUT.PUT_LINE('Calle: ' || r.Calle_sucu);
17    DBMS_OUTPUT.PUT_LINE('Telefono: ' || r.Telefono_sucu);
18    DBMS_OUTPUT.PUT_LINE('Email: ' || r.Email_sucu);
19  END LOOP;
20
21  DBMS_OUTPUT.PUT_LINE('Resumen de Ventas Diarias:');
22
23  -- Resumen de ventas diarias
24  FOR r IN (SELECT Fecha, SUM(Total) AS Total_Diario
25            FROM VentasDiarias
26            WHERE ID_Sucursal = p_id_sucursal
27            AND Fecha BETWEEN p_fecha_inicio AND p_fecha_fin
28            GROUP BY Fecha) LOOP
29    DBMS_OUTPUT.PUT_LINE('Fecha: ' || TO_CHAR(r.Fecha, 'DD-MON-YY') || ', Total Diario: ' || r.Total_Diario);
30  END LOOP;
31
32  -- Detalle de ventas
33  DBMS_OUTPUT.PUT_LINE('Detalles de Ventas:');
34  FOR r IN (SELECT *
35            FROM VentasAuditoria
36            WHERE ID_Sucursal = p_id_sucursal
37            AND Fecha_Venta BETWEEN p_fecha_inicio AND p_fecha_fin) LOOP
38    DBMS_OUTPUT.PUT_LINE('ID Venta: ' || r.ID_Venta || ', Fecha: ' || TO_CHAR(r.Fecha_Venta, 'DD-MON-YY') ||
39                          ', ISBN: ' || r.ISBN || ', Título: ' || r.Título ||
40                          ', Cantidad: ' || r.Cantidad || ', Precio: ' || r.Precio ||
41                          ', Total: ' || r.Total_Venta);
42  END LOOP;
43 END;
44 /
Procedure created.
```

--solicitud de datos por teclado

DECLARE

v_id_sucursal NUMBER;

v_fecha_inicio DATE;

v_fecha_fin DATE;

BEGIN

-- Solicitar el ID de la sucursal

v_id_sucursal := &v_id_sucursal;

-- Solicitar la fecha de inicio

v_fecha_inicio := TO_DATE('&fecha_inicio', 'YYYY-MM-DD');

-- Solicitar la fecha de fin

v_fecha_fin := TO_DATE('&fecha_fin', 'YYYY-MM-DD');

-- Llamar al procedimiento con los datos proporcionados

```
GenerarInformeVentas(  
    p_id_sucursal => v_id_sucursal,  
    p_fecha_inicio => v_fecha_inicio,  
    p_fecha_fin => v_fecha_fin  
);  
END;  
  
/
```

```
6  -- Solicitar el ID de la sucursal  
7  v_id_sucursal := &v_id_sucursal;  
8  
9  -- Solicitar la fecha de inicio  
10 v_fecha_inicio := TO_DATE('&fecha_inicio', 'YYYY-MM-DD');  
11  
12 -- Solicitar la fecha de fin  
13 v_fecha_fin := TO_DATE('&fecha_fin', 'YYYY-MM-DD');  
14  
15 -- Llamar al procedimiento con los datos proporcionados  
16 GenerarInformeVentas(  
17     p_id_sucursal => v_id_sucursal,  
18     p_fecha_inicio => v_fecha_inicio,  
19     p_fecha_fin => v_fecha_fin  
20 );  
21 END;  
22 /  
Enter value for v_id_sucursal: 1  
old 7: v_id_sucursal := &v_id_sucursal;  
new 7: v_id_sucursal := 1;  
Enter value for fecha_inicio: 2024/7/3  
old 10: v_fecha_inicio := TO_DATE('&fecha_inicio', 'YYYY-MM-DD');  
new 10: v_fecha_inicio := TO_DATE('2024/7/3', 'YYYY-MM-DD');  
Enter value for fecha_fin: 2024/7/3  
old 13: v_fecha_fin := TO_DATE('&fecha_fin', 'YYYY-MM-DD');  
new 13: v_fecha_fin := TO_DATE('2024/7/3', 'YYYY-MM-DD');  
Información de la Sucursal:  
ID Sucursal: 1  
Ciudad: panama  
Provincia: panama  
Calle: 15  
Teléfono: 12345  
Email: panama@gmail.com  
Resumen de Ventas Diarias:  
Fecha: 03-JUL-24, Total Diario: 2800
```

11. Actualizar Información de Clientes: Crear un procedimiento para actualizar la información de los clientes en la tabla Clientes, incluyendo nombre, teléfono, cédula, y estatus de membresía.

-- Se crea el procedimiento para actualizar la información del cliente

```
CREATE OR REPLACE PROCEDURE ActualizarInformacionCliente (  
    p_id_cliente IN Clientes.ID_Cliente%TYPE,  
    p_nombre_clie IN Clientes.Nombre_clie%TYPE,
```

```

    p_apellido_clie IN Clientes.Apellido_clie%TYPE,
    p_telefono_clie IN Clientes.Telefono_clie%TYPE,
    p_cedula_clie IN Clientes.Cedula_clie%TYPE,
    p_fecha_nacimiento IN Clientes.Fecha_Nacimiento%TYPE,
    p_es_socio IN Clientes.Es_Socio%TYPE
) IS
BEGIN
    UPDATE Clientes

    SET
        Nombre_clie = p_nombre_clie,
        Apellido_clie = p_apellido_clie,
        Telefono_clie = p_telefono_clie,
        Cedula_clie = p_cedula_clie,
        Fecha_Nacimiento = p_fecha_nacimiento,
        Es_Socio = p_es_socio

    WHERE
        ID_Cliente = p_id_cliente;

    IF SQL%ROWCOUNT = 0 THEN
        DBMS_OUTPUT.PUT_LINE('No se encontró un cliente con ID ' || p_id_cliente);
    ELSE
        DBMS_OUTPUT.PUT_LINE('Información del cliente con ID ' || p_id_cliente || '
actualizada exitosamente.');
```

END IF;

END;

/

SET SERVEROUTPUT ON;

DECLARE

 v_id_cliente INT;

 v_nombre_clie VARCHAR2(100);

```

v_apellido_clie VARCHAR2(100);
v_telefono_clie VARCHAR2(20);
v_cedula_clie VARCHAR2(20);
v_fecha_nacimiento DATE;
v_es_socio INT;
BEGIN
-- Solicitar al usuario que ingrese los valores

DBMS_OUTPUT.PUT_LINE('Ingrese el ID del cliente:');

v_id_cliente := &v_id_cliente;


DBMS_OUTPUT.PUT_LINE('Ingrese el nombre del cliente:');

v_nombre_clie := '&v_nombre_clie';


DBMS_OUTPUT.PUT_LINE('Ingrese el apellido del cliente:');

v_apellido_clie := '&v_apellido_clie';


DBMS_OUTPUT.PUT_LINE('Ingrese el teléfono del cliente:');

v_telefono_clie := '&v_telefono_clie';


DBMS_OUTPUT.PUT_LINE('Ingrese la cédula del cliente:');

v_cedula_clie := '&v_cedula_clie';


DBMS_OUTPUT.PUT_LINE('Ingrese la fecha de nacimiento del cliente (YYYY-MM-DD):');

v_fecha_nacimiento := TO_DATE('&v_fecha_nacimiento', 'YYYY-MM-DD');


DBMS_OUTPUT.PUT_LINE('Ingrese el estatus de membresía (1 para socio, 0 para no
socio):');

v_es_socio := &v_es_socio;

-- Llamar al procedimiento para actualizar la información del cliente

ActualizarInformacionCliente(

```

```

p_id_cliente => v_id_cliente,

p_nombre_clie => v_nombre_clie,

p_apellido_clie => v_apellido_clie,

p_telefono_clie => v_telefono_clie,

p_cedula_clie => v_cedula_clie,

p_fecha_nacimiento => v_fecha_nacimiento,


p_es_socio => v_es_socio

);

END;

/

```



```

SQL> CREATE OR REPLACE PROCEDURE ActualizarInformacionCliente (
2   p_id_cliente IN Clientes.ID_Cliente%TYPE,
3   p_nombre_clie IN Clientes.Nombre_Cliente%TYPE,
4   p_apellido_clie IN Clientes.Apellido_Cliente%TYPE,
5   p_telefono_clie IN Clientes.Telefono_Cliente%TYPE,
6   p_cedula_clie IN Clientes.Cedula_Cliente%TYPE,
7   p_fecha_nacimiento IN Clientes.Fecha_Nacimiento%TYPE,
8   p_es_socio IN Clientes.Es_Socio%TYPE)
9  IS
10 BEGIN
11   UPDATE Clientes
12   SET
13     Nombre_clie = p_nombre_clie,
14     Apellido_clie = p_apellido_clie,
15     Telefono_clie = p_telefono_clie,
16     Cedula_clie = p_cedula_clie,
17     Fecha_Nacimiento = p_fecha_nacimiento,
18     Es_Socio = p_es_socio
19   WHERE
20     ID_Cliente = p_id_cliente;
21
22   IF SQL%ROWCOUNT = 0 THEN
23     DBMS_OUTPUT.PUT_LINE('No se encontró un cliente con ID ' || p_id_cliente);
24   ELSE
25     DBMS_OUTPUT.PUT_LINE('Información del cliente con ID ' || p_id_cliente || ' actualizada exitosamente.');

Procedure created.



```

SQL> SET SERVEROUTPUT ON;
SQL> BEGIN
2 ActualizarInformacionCliente(
3 p_id_cliente => 1,
4 p_nombre_clie => 'NuevoNombre',
5 p_apellido_clie => 'NuevoApellido',
6 p_telefono_clie => '323456789',
7 p_cedula_clie => '987654321',
8 p_fecha_nacimiento => TO_DATE('1980-05-15', 'YYYY-MM-DD'),
9 p_es_socio => 1 -- Estatus de membresía, 1 para socio, 0 para no socio
10);
11 END;
12 /

```



Información del cliente con ID 1 actualizada exitosamente.



PL/SQL procedure successfully completed.


```

12. Ingresar Cliente: Desarrollar un procedimiento almacenado denominado IngresarCliente para añadir nuevos registros a la tabla Clientes, asegurando que la cédula sea única y generando automáticamente IDs únicos. También, crear una vista llamada Vista_Clientes para simplificar la consulta de los clientes añadidos.

-- Restricción de unicidad para el atributo cedula

ALTER TABLE Clientes

ADD CONSTRAINT uq_clientes_cedula UNIQUE (Cedula_clie);

```
SQL> ALTER TABLE Clientes
  2  ADD CONSTRAINT uq_clientes_cedula UNIQUE (Cedula_clie);

Table altered.
```

-- Secuencia para IDs de Clientes

```
CREATE SEQUENCE Clientes_SEQ

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;
```

```
SET SERVEROUTPUT ON;
```

-- Crear el procedimiento almacenado para insertar un cliente

```
CREATE OR REPLACE PROCEDURE IngresarCliente (

  p_Nombre_clie IN VARCHAR2,

  p_Apellido_clie IN VARCHAR2,

  p_Telefono_clie IN VARCHAR2,

  p_Cedula_clie IN VARCHAR2,

  p_Fecha_nacimiento IN DATE,

  p_Es_Socio IN NUMBER

)

AS

  v_Inserted BOOLEAN := FALSE; -- Variable para controlar si se realizó la inserción

BEGIN

  BEGIN

    -- Insertar nuevo cliente

    INSERT INTO Clientes (ID_Cliente, Nombre_clie, Apellido_clie, Telefono_clie, Cedula_clie,

      Fecha_Nacimiento, Es_Socio)
```

```
VALUES (Clientes_SEQ.NEXTVAL, p_Nombre_clie, p_Apellido_clie, p_Telefono_clie,  
p_Cedula_clie, p_Fecha_nacimiento, p_Es_Socio);
```

```
v_Inserted := TRUE; -- Indicar que se realizó la inserción
```

```
EXCEPTION
```

```
WHEN DUP_VAL_ON_INDEX THEN
```

```
-- Generar error si el cliente con la misma cédula ya existe
```

```
RAISE_APPLICATION_ERROR(-20001, 'Ya existe un cliente con la misma cédula.');
```

```
WHEN OTHERS THEN
```

```
-- Manejar otros errores
```

```
RAISE_APPLICATION_ERROR(-20002, 'Error inesperado: ' || SQLERRM);
```

```
END;
```

```
-- Mostrar mensaje de éxito solo si la inserción fue realizada
```

```
IF v_Inserted THEN
```

```
DBMS_OUTPUT.PUT_LINE('Cliente ingresado correctamente.');
```

```
END IF;
```

```
END;
```

```
/
```

```
SQL> SET SERVEROUTPUT ON;  
SQL> -- Crear el procedimiento almacenado para insertar un cliente  
SQL> CREATE OR REPLACE PROCEDURE IngresarCliente (  
2   p_Nombre_clie IN VARCHAR2,  
3   p_Apellido_clie IN VARCHAR2,  
4   p_Telefono_clie IN VARCHAR2,  
5   p_Cedula_clie IN VARCHAR2,  
6   p_Fecha_nacimiento IN DATE,  
7   p_Es_Socio IN NUMBER  
8 )  
9 AS  
10  v_Inserted BOOLEAN := FALSE; -- Variable para controlar si se realizó la inserción  
11 BEGIN  
12  BEGIN  
13    -- Insertar nuevo cliente  
14    INSERT INTO Clientes (ID_Cliente, Nombre_clie, Apellido_clie, Telefono_clie, Cedula_clie, Fecha_Nacimiento, Es_Socio)  
15    VALUES (Clientes_SEQ.NEXTVAL, p_Nombre_clie, p_Apellido_clie, p_Telefono_clie, p_Cedula_clie, p_Fecha_nacimiento, p_Es_Socio);  
16  
17    v_Inserted := TRUE; -- Indicar que se realizó la inserción  
18  EXCEPTION  
19    WHEN DUP_VAL_ON_INDEX THEN  
20      -- Generar error si el cliente con la misma cédula ya existe  
21      RAISE_APPLICATION_ERROR(-20001, 'Ya existe un cliente con la misma cédula.');
```

-- Solicitar datos al usuario y llamar al procedimiento

DECLARE

v_Nombre_clie VARCHAR2(100);

v_Apellido_clie VARCHAR2(100);

v_Telefono_clie VARCHAR2(20);

v_Cedula_clie VARCHAR2(20);

v_Fecha_nacimiento DATE;

v_Es_Socio NUMBER;

BEGIN

-- Solicitar datos al usuario

v_Nombre_clie := '&Ingrese_nombre_clie';

v_Apellido_clie := '&Ingrese_apellido_clie';

v_Telefono_clie := '&Ingrese_telefono_clie';

v_Cedula_clie := '&Ingrese_cedula_clie';

v_Fecha_nacimiento := TO_DATE('&Ingrese_fecha_nacimiento', 'YYYY-MM-DD');

v_Es_Socio := '&Ingrese_es_socio'; -- 0 para no socio, 1 para socio

-- Llamar al procedimiento para agregar el cliente

IngresarCliente(v_Nombre_clie, v_Apellido_clie, v_Telefono_clie, v_Cedula_clie,
v_Fecha_nacimiento, v_Es_Socio);

END;

/

```

SQL> DECLARE
2   v_Nombre_clie VARCHAR2(100);
3   v_Apellido_clie VARCHAR2(100);
4   v_Telefono_clie VARCHAR2(20);
5   v_Cedula_clie VARCHAR2(20);
6   v_Fecha_nacimiento DATE;
7   v_Es_Socio NUMBER;
8 BEGIN
9   -- Solicitar datos al usuario
10  v_Nombre_clie := '&Ingreso_nombre_clie';
11  v_Apellido_clie := '&Ingreso_apellido_clie';
12  v_Telefono_clie := '&Ingreso_telefono_clie';
13  v_Cedula_clie := '&Ingreso_cedula_clie';
14  v_Fecha_nacimiento := TO_DATE('&Ingreso_fecha_nacimiento', 'YYYY-MM-DD');
15  v_Es_Socio := '&Ingreso_es_socio'; -- 0 para no socio, 1 para socio
16
17  -- Llamar al procedimiento para agregar el cliente
18  IngresarCliente(v_Nombre_clie, v_Apellido_clie, v_Telefono_clie, v_Cedula_clie, v_Fecha_nacimiento, v_Es_Socio);
19 END;
20 /

Enter value for ingreso_nombre_clie: Sebastian
old 10:   v_Nombre_clie := '&Ingreso_nombre_clie';
new 10:   v_Nombre_clie := 'Sebastian';
Enter value for ingreso_apellido_clie: Mejia
old 11:   v_Apellido_clie := '&Ingreso_apellido_clie';
new 11:   v_Apellido_clie := 'Mejia';
Enter value for ingreso_telefono_clie: 7777-7777
old 12:   v_Telefono_clie := '&Ingreso_telefono_clie';
new 12:   v_Telefono_clie := '7777-7777';
Enter value for ingreso_cedula_clie: 1-123-1234
old 13:   v_Cedula_clie := '&Ingreso_cedula_clie';
new 13:   v_Cedula_clie := '1-123-1234';
Enter value for ingreso_fecha_nacimiento: 2000-1-1
old 14:   v_Fecha_nacimiento := TO_DATE('&Ingreso_fecha_nacimiento', 'YYYY-MM-DD');
new 14:   v_Fecha_nacimiento := TO_DATE('2000-1-1', 'YYYY-MM-DD');
Enter value for ingreso_es_socio: 1
old 15:   v_Es_Socio := '&Ingreso_es_socio'; -- 0 para no socio, 1 para socio
new 15:   v_Es_Socio := '1'; -- 0 para no socio, 1 para socio
Cliente ingresado correctamente.

PL/SQL procedure successfully completed.

```

-- Crear vista de los clientes

```
CREATE OR REPLACE VIEW Vista_Clientes AS
```

```
SELECT
```

```
    ID_Cliente AS "ID de Cliente",
```

```
    Nombre_clie AS "Nombre",
```

```
    Apellido_clie AS "Apellido",
```

```
    Telefono_clie AS "Teléfono",
```

```
    Cedula_clie AS "Cédula",
```

```
    Fecha_Nacimiento AS "Fecha de Nacimiento",
```

```
    Es_Socio AS "Es Socio"
```

```
FROM Clientes;
```

-- Consultar la vista de los clientes

```
SELECT * FROM Vista_Clientes;
```


| ID de Cliente | | | |
|---------------|------------|-----------|----------|
| Nombre | | | |
| Apellido | | | |
| Teléfono | Cédula | Fecha de | Es Socio |
| 1 | | | |
| Sebastian | | | |
| Mejia | | | |
| 7777-7777 | 1-123-1234 | 01-JAN-00 | 1 |