Bases de Datos Estructuradas

2024

- Segunda Unidad

2 . Consultas avanzadas sobre una o más tablas | Horas de la Unidad: 24 | Horas Presenciales: 18 | Horas Online: 6

APRENDIZAJES ESPERADOS	CRITERIOS DE EVALUACIÓN	CONTENIDOS MÍNIMOS OBLIGATORIOS
2.1 Genera consultas y subconsultas con el objetivo de mostrar resúmenes de datos, en base a requerimientos y procedimientos establecidos.	2.1.1 Formula consultas empleando funciones de una sola fila aplicables a cadenas, números y fechas, a partir de requerimientos de visualización. 2.1.2 Elabora consultas empleando funciones de conversión y expresiones condicionales, en base a procedimientos establecidos. 2.1.3 Formula consultas de resumen empleando funciones de grupo, a partir de requerimientos de visualización. 2.1.4 Genera consultas aplicables a múltiples tablas, a partir de requerimientos de visualización. 2.1.5 Genera subconsultas aplicables a una o más tablas, a partir de requerimientos de visualización.	• Funciones de una sola fila. • Funciones de conversión y expresiones condicionales. • Funciones de grupo y agrupamiento de filas usando las cláusulas GROUP BY y HAVING. • Consultas sobre múltiples tablas usando uniones y autouniones INNER JOIN o OUTER JOIN. • Subconsultas de una sola fila y de varias filas con los operadores ALL o ANY.

- Segunda Evaluación

2	Sentencias de consulta de datos (DQL) usando SQL.	2.1.1 Formula consultas empleando funciones de una sola fila aplicables a cadenas, números y fechas, a partir de requerimientos de visualización. 2.1.2 Elabora consultas empleando funciones de conversión y expresiones condicionales en base a procedimientos establecidos. 2.1.3 Produce consultas de resumen empleando funciones de grupo, a partir de requerimientos de visualización. 2.1.4. Genera consultas aplicables a múltiples tablas, a partir de requerimientos de visualización. 2.1.5 Genera subconsultas aplicables una o más tablas, a partir de requerimientos de visualización.	Los estudiantes, de manera individual, deben diseñar y ejecutar consultas sobre una o más tablas creadas previamente, a partir de los requerimientos planteados por el docente.	Escala de apreciación	30%

Script Base de datos

https://github.com/ebravo930/Inacap_2024_Bases-de-Datos-Estructuradas

Ejemplos de Consultas Complejas

Con esta base de datos, puedes realizar las siguientes consultas:

1. Subconsultas y Funciones de Agregación:

Encuentra los clientes que han gastado más que el promedio de todos los clientes.

```
SELECT FirstName, LastName, TotalSpent
FROM (
    SELECT c.CustomerID, c.FirstName, c.LastName, SUM(o.TotalAmount) AS
TotalSpent
    FROM Customers c
    JOIN Orders o ON c.CustomerID = o.CustomerID
    GROUP BY c.CustomerID, c.FirstName, c.LastName
)
WHERE TotalSpent > (SELECT AVG(TotalAmount) FROM Orders);
```

2. Consulta con JOIN y Condiciones Complejas:

Encuentra los nombres de los productos que han sido ordenados más de 3 veces.

```
SELECT p.ProductName, COUNT(od.ProductID) AS OrderCount
FROM Products p
JOIN OrderDetails od ON p.ProductID = od.ProductID
GROUP BY p.ProductName
HAVING COUNT(od.ProductID) > 3;
```

3. Vista de Resumen de Ventas:

Crea una vista que muestre el total de ventas por cliente.

```
CREATE OR REPLACE VIEW CustomerSalesSummary AS
SELECT c.CustomerID, c.FirstName, c.LastName, SUM(o.TotalAmount) AS
TotalSales
FROM Customers c
JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.CustomerID, c.FirstName, c.LastName;
```

Procedimiento Almacenado para Agregar un Nuevo Producto

El procedimiento almacenado **sp_AgregarProducto** se utiliza para insertar un nuevo producto en la tabla **Products**.

```
CREATE OR REPLACE PROCEDURE sp_AgregarProducto (
    p_ProductName NVARCHAR2,
    p_Category NVARCHAR2,
    p_Price NUMBER
) AS
BEGIN
    INSERT INTO Products (ProductName, Category, Price)
    VALUES (p_ProductName, p_Category, p_Price);

    DBMS_OUTPUT.PUT_LINE('Producto agregado: ' ||
p_ProductName || ', Categoría: ' || p_Category || ', Precio: ' ||
p_Price);
END;
/
```

Trigger para Auditar Eliminaciones de Productos

El trigger trg_AuditProductDelete se activa cada vez que se elimina un producto de la tabla Products y registra la información en una tabla de auditoría llamada ProductAudit.

Creación de la Tabla de AuditoríaPrimero, crea la tabla ProductAudit para almacenar los registros de auditoría:

```
CREATE TABLE ProductAudit (
        AuditID NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
        ProductID NUMBER,
        ProductName NVARCHAR2(100),
        Category NVARCHAR2(50),
        Price NUMBER(10, 2),
        DeletionDate DATE
);
```

Creación del TriggerAhora, crea el trigger trg_AuditProductDelete para auditar las eliminaciones:

```
CREATE OR REPLACE TRIGGER trg_AuditProductDelete
AFTER DELETE ON Products
FOR EACH ROW
BEGIN
    INSERT INTO ProductAudit (ProductID, ProductName, Category, Price,
DeletionDate)
    VALUES (:OLD.ProductID, :OLD.ProductName, :OLD.Category, :OLD.Price,
SYSDATE);

    DBMS_OUTPUT.PUT_LINE('Producto eliminado: ' || :OLD.ProductName || ',
Categoría: ' || :OLD.Category || ', Precio: ' || :OLD.Price);
END;
/
```

Ejercicios

Generar uno cuando se INSERTE

Generar uno cuando se ACTUALIZAR

Ejercicios

Generar uno cuando se INSERTE

```
CREATE TABLE ProductInsertAudit (
    AuditID NUMBER GENERATED BY DEFAULT AS
IDENTITY PRIMARY KEY,
    ProductID NUMBER,
    ProductName NVARCHAR2(100),
    Category NVARCHAR2(50),
    Price NUMBER(10, 2),
    InsertionDate DATE
);
```

```
CREATE OR REPLACE TRIGGER trg_AuditProductInsert
AFTER INSERT ON Products
FOR EACH ROW
BEGIN
    INSERT INTO ProductInsertAudit (ProductID, ProductName, Category, Price,
InsertionDate)
    VALUES (:NEW.ProductID, :NEW.ProductName, :NEW.Category, :NEW.Price,
SYSDATE);

    DBMS_OUTPUT.PUT_LINE('Producto insertado: ' || :NEW.ProductName || ',
Categoría: ' || :NEW.Category || ', Precio: ' || :NEW.Price);
END;
/
```

Ejercicios

Generar uno cuando se ACTUALICE

```
CREATE TABLE ProductUpdateAudit (
    AuditID NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    ProductID NUMBER,
    OldProductName NVARCHAR2(100),
    NewProductName NVARCHAR2(100),
    OldCategory NVARCHAR2(50),
    NewCategory NVARCHAR2(50),
    OldPrice NUMBER(10, 2),
    NewPrice NUMBER(10, 2),
    UpdateDate DATE
);
```

```
CREATE OR REPLACE TRIGGER trg_AuditProductUpdate
AFTER UPDATE ON Products
FOR EACH ROW
BEGIN
    INSERT INTO ProductUpdateAudit (ProductID, OldProductName,
NewProductName, OldCategory, NewCategory, OldPrice, NewPrice,
UpdateDate)
    VALUES (:OLD.ProductID, :OLD.ProductName, :NEW.ProductName,
:OLD.Category, :NEW.Category, :OLD.Price, :NEW.Price, SYSDATE);

    DBMS_OUTPUT.PUT_LINE('Producto actualizado: ID ' || :NEW.ProductID ||
                ', Nombre: ' || :OLD.ProductName || ' a ' || :NEW.ProductName
||
                ', Categoría: ' || :OLD.Category || ' a ' || :NEW.Category ||
                ', Precio: ' || :OLD.Price || ' a ' || :NEW.Price);
END;
/
```

Funciones de Una Sola Fila

Descripción:
Utiliza las funciones de una sola fila UPPER, LOWER, y INITCAP para formatear los nombres de los clientes en diferentes formatos.

Funciones de Una Sola Fila

Descripción:
Utiliza las funciones de una sola fila UPPER, LOWER, y INITCAP para formatear los nombres de los clientes en diferentes formatos.

```
-- Consulta para formatear nombres de clientes utilizando funciones de una sola fila
SELECT
    CustomerID,
    UPPER(FirstName) AS FirstName_Upper,
    LOWER(LastName) AS LastName_Lower,
    INITCAP(FirstName || ' ' || LastName) AS NombreCompleto_Capitalizado
FROM Customers;
```

Funciones de Conversión y Expresiones Condicionales
Descripción:
Utiliza funciones de conversión (TO_CHAR, TO_DATE) y una expresión condicional (CASE) para mostrar la fecha de registro de los clientes en diferentes formatos y categorizarlos en función de su antigüedad.

Funciones de Conversión y Expresiones Condicionales
Descripción:
Utiliza funciones de conversión (TO_CHAR, TO_DATE) y una expresión condicional (CASE) para mostrar la fecha de registro de los clientes en diferentes formatos y categorizarlos en función de su antigüedad.

```
-- Consulta para convertir fechas y usar expresiones condicionales
SELECT
    CustomerID,
    FirstName,
    LastName,
    TO_CHAR(RegistrationDate, 'DD-MON-YYYY') AS RegistrationDate_Formatted,
    CASE
        WHEN MONTHS_BETWEEN(SYSDATE, RegistrationDate) / 12 >= 1 THEN
'Antiguo'
        ELSE 'Nuevo'
    END AS CategoriaCliente
FROM Customers;
```

Funciones de Grupo y Agrupamiento de Filas
Descripción:
Utiliza las funciones de grupo SUM, AVG, COUNT, y MAX con las cláusulas GROUP BY y HAVING para mostrar el total de ventas, el promedio, el número de órdenes y la orden más alta por cliente, filtrando solo aquellos con más de 2 órdenes.

Funciones de Grupo y Agrupamiento de Filas
Descripción:
Utiliza las funciones de grupo SUM, AVG, COUNT, y MAX con las cláusulas GROUP BY y HAVING para mostrar el total de ventas, el promedio, el número de órdenes y la orden más alta por cliente, filtrando solo aquellos con más de 2 órdenes.

```
-- Consulta para agrupar datos y usar funciones de grupo
SELECT
    CustomerID,
    SUM(TotalAmount) AS TotalVentas,
    AVG(TotalAmount) AS PromedioVentas,
    COUNT(OrderID) AS NumeroOrdenes,
    MAX(TotalAmount) AS OrdenMaxima
FROM Orders
GROUP BY CustomerID
HAVING COUNT(OrderID) > 2;
```

Consultas sobre Múltiples Tablas Usando Uniones y Autouniones
Descripción:
Realiza una consulta que use INNER JOIN para mostrar los detalles de cada orden, incluyendo el nombre del cliente, y un LEFT OUTER JOIN para mostrar todas las órdenes, incluidas aquellas sin detalles.

Consultas sobre Múltiples Tablas Usando Uniones y Autouniones

Descripción:

Realiza una consulta que use INNER JOIN para mostrar los detalles de cada orden, incluyendo el nombre del cliente, y un LEFT OUTER JOIN para mostrar todas las órdenes, incluidas aquellas sin detalles.

```
-- Consulta utilizando INNER JOIN
SELECT
    o.OrderID,
    c.FirstName || ' ' || c.LastName AS NombreCliente,
    o.OrderDate,
    o.TotalAmount
FROM Orders o
INNER JOIN Customers c ON o.CustomerID = c.CustomerID;

-- Consulta utilizando LEFT OUTER JOIN
SELECT
    o.OrderID,
    o.OrderDate,
    o.TotalAmount,
    od.ProductID,
    od.Quantity,
    od.LineTotal
FROM Orders o
LEFT OUTER JOIN OrderDetails od ON o.OrderID = od.OrderID;
```

Subconsultas de Una Sola Fila y de Varias Filas con Operadores ALL o ANY
Descripción:

Utiliza subconsultas de una sola fila y de varias filas con los operadores ALL y ANY para encontrar los productos que tienen un precio superior al de todos los productos de la categoría 'Accessories' y aquellos que tienen un precio inferior al de cualquier producto de la categoría 'Electronics'.

Subconsultas de Una Sola Fila y de Varias Filas con Operadores ALL o ANY
Descripción:

Utiliza subconsultas de una sola fila y de varias filas con los operadores ALL y ANY para encontrar los productos que tienen un precio superior al de todos los productos de la categoría 'Accessories' y aquellos que tienen un precio inferior al de cualquier producto de la categoría 'Electronics'.

```
-- Subconsulta de una sola fila con el operador ALL
SELECT ProductName, Price
FROM Products
WHERE Price > ALL (
    SELECT Price
    FROM Products
    WHERE Category = 'Accessories'
);

-- Subconsulta de varias filas con el operador ANY
SELECT ProductName, Price
FROM Products
WHERE Price < ANY (
    SELECT Price
    FROM Products
    WHERE Category = 'Electronics'
);
```

MUCHAS GRACIAS!

inacap.cl