

MANUAL SQL SERVER MANAGEMENT STUDIO – CARLOS JOSÉ TORRE GARCÍA

MANUAL, MODULO 5, CONSULTAS AVANZADAS EN SQL SERVER:

1. CONSULTAS DE AGREACION

Las consultas de agregación en SQL Server permiten resumir o analizar conjuntos de datos, por ejemplo, calcular totales, promedios, máximos, etc.

A. Funciones de Agregacion (SUM,COUNT,AVG,MIN,MAX):

Estas funciones operan sobre un conjunto de filas y devuelven un solo valor.

```
-- Total de productos
SELECT COUNT(*) AS TotalProductos FROM Productos;

-- Suma del stock disponible
SELECT SUM(Stock) AS StockTotal FROM Productos;

-- Precio promedio de los productos
SELECT AVG(Precio) AS PrecioPromedio FROM Productos;

-- Precio más alto
SELECT MAX(Precio) AS PrecioMaximo FROM Productos;

-- Precio más bajo
SELECT MIN(Precio) AS PrecioMinimo FROM Productos;
```

B. Agrupacion de datos

GROUP BY permite agrupar registros por un campo y aplicar funciones de agregación sobre cada grupo.

Ejemplo:

Supongamos que tienes una columna Categoria:

```
-- Mostrar stock total por categoría
SELECT Categoria, SUM(Stock) AS TotalStock
FROM Productos
GROUP BY Categoria;
```

Puedes usar HAVING para filtrar resultados después de agrupar:

```
-- Solo categorías con más de 100 unidades en total
SELECT Categoria, SUM(Stock) AS TotalStock
FROM Productos
GROUP BY Categoria
HAVING SUM(Stock) > 100;
```

C. Subconsultas

Una subconsulta es una consulta dentro de otra consulta. Puede estar en el SELECT, FROM, o WHERE.

Ejemplos:

1. Subconsulta en WHERE

```
-- Productos cuyo precio es mayor al promedio
SELECT Nombre, Precio
FROM Productos
WHERE Precio > (SELECT AVG(Precio) FROM Productos);
```

2. Subconsulta en FROM:

```
-- Calcular promedio de stock por categoría y filtrar el resultado final
SELECT Categoria, PromedioStock
FROM (
    SELECT Categoria, AVG(Stock) AS PromedioStock
    FROM Productos
    GROUP BY Categoria
) AS Subconsulta
WHERE PromedioStock > 50;
```

2. CONSULTAS DE UNION

Las consultas de unión permiten combinar datos de dos o más tablas usando relaciones entre ellas.

A. Tipos de uniones (INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN):

1. INNER JOIN:

Devuelve solo las filas que tienen coincidencia en ambas tablas.

```
SELECT P.Nombre, C.Nombre AS Categoria
FROM Productos P
INNER JOIN Categorías C ON P.IdCategoria = C.IdCategoria;
```

2. LEFT JOIN:

Devuelve todas las filas de la tabla izquierda y las coincidencias de la derecha (si no hay coincidencia, muestra NULL).

```
SELECT P.Nombre, C.Nombre AS Categoria
FROM Productos P
LEFT JOIN Categorías C ON P.IdCategoria = C.IdCategoria;
```

3. RIGHT JOIN:

Devuelve todas las filas de la tabla derecha y las coincidencias de la izquierda.

```
SELECT P.Nombre, C.Nombre AS Categoria
FROM Productos P
RIGHT JOIN Categorías C ON P.IdCategoria = C.IdCategoria;
```

FULL JOIN:

Devuelve todas las filas cuando hay coincidencias en cualquiera de las tablas.

```
SELECT P.Nombre, C.Nombre AS Categoria
FROM Productos P
FULL JOIN Categorías C ON P.IdCategoria = C.IdCategoria;
```

B. Múltiples Uniones

Puedes unir más de dos tablas encadenando varias sentencias JOIN.

```
SELECT V.IdVenta, C.Nombre AS Cliente, P.Nombre AS Producto
FROM Ventas V
INNER JOIN Clientes C ON V.IdCliente = C.IdCliente
INNER JOIN Productos P ON V.IdProducto = P.IdProducto;
```

C. Subconsultas En Uniones

Puedes usar subconsultas dentro de una unión, especialmente para combinar resultados temporales o agrupados.

```
-- Subconsulta para obtener productos más caros y unirlos con su categoría
SELECT P.Nombre, C.Nombre AS Categoria
FROM (
    SELECT * FROM Productos WHERE Precio > 100
) AS P
INNER JOIN Categorías C ON P.IdCategoria = C.IdCategoria;
```

También puedes unir subconsultas entre sí:

```
-- Unión entre dos subconsultas
SELECT *
FROM
    (SELECT IdProducto, Nombre FROM Productos WHERE Stock > 0) AS A
INNER JOIN
    (SELECT IdProducto FROM Ventas WHERE Fecha >= '2024-01-01') AS B
ON A.IdProducto = B.IdProducto;
```

3. CONSULTAS DE FUNCIONES DE VENTANA

Las funciones de ventana permiten realizar cálculos sobre un conjunto de filas relacionadas (una "ventana") sin agrupar los datos como GROUP BY.

A. Funciones OVER

El cláusula OVER define la ventana sobre la que se aplican funciones como:

- ROW_NUMBER() – número de fila por grupo.
- RANK() – ranking con empates.
- DENSE_RANK() – ranking sin huecos.
- SUM(), AVG(), COUNT() – funciones agregadas sobre la ventana.

```
-- Numerar los productos por categoría
SELECT
    Nombre, Categoria,
    ROW_NUMBER() OVER (PARTITION BY Categoria ORDER BY Precio DESC) AS Numero
FROM Productos;
```

B. Participaciones De Ventana

PARTITION BY divide los datos en grupos (ventanas independientes) sin agrupar el resultado final, permitiendo ver totales por grupo en cada fila.

```
-- Calcular el total de stock por categoría, mostrado en cada fila
SELECT
    Nombre, Categoria, Stock,
    SUM(Stock) OVER (PARTITION BY Categoria) AS StockPorCategoria
FROM Productos;
```

C. Marcos De Ventana

ROWS BETWEEN y RANGE permiten definir el marco exacto de filas que se incluirán en el cálculo dentro de cada partición.

```
-- Calcular promedio móvil del precio actual y los 2 anteriores
SELECT
    Nombre, Precio,
    AVG(Precio) OVER (ORDER BY IdProducto
                     ROWS BETWEEN 2 PRECEDING AND CURRENT ROW) AS PromedioMovil
FROM Productos;
```

