

Agrupación y particionamiento

Presenta: Juliho Castillo Colmenares

¡Bienvenidos a la actividad práctica del módulo!

Como actividad de este módulo, deberás aplicar las consultas que has aprendido en el módulo.

¡Buena suerte!

Paso a paso:

Usa la base de datos `AdventureWorks2022` para completar los siguientes ejercicios. Cada ejercicio se basa en las lecciones que has visto en el curso. Asegúrate de seguir las instrucciones cuidadosamente y aplicar el tipo de unión adecuado para cada caso.

1. Consulta con GROUP BY:

- Utiliza la tabla `Sales.SalesOrderDetail` para crear una consulta que obtenga el `ProductID`, el número total de ventas (`OrderQty`) y el valor total de ventas (`LineTotal`) por cada producto usando la cláusula `GROUP BY`.

```
SELECT
    ProductID,
    SUM(OrderQty) AS TotalSalesQuantity,
    SUM(LineTotal) AS TotalSalesValue
FROM
    [AdventureWorks2022].[Sales].[SalesOrderDetail]
GROUP BY
    ProductID;
```

RESULTS			
	ProductID	TotalSalesQuan...	TotalSalesValue
1	925	625	93584.422996
2	902	36	7201.872000
3	710	90	513.000000
4	879	249	39591.000000
5	733	90	32120.820000

2. Filtrado con HAVING:

- Utiliza la tabla `Sales.SalesOrderDetail` para crear una consulta que agrupe los datos por `ProductID` y obtenga el número total de ventas (`OrderQty`) por producto. Filtra los grupos para mostrar solo aquellos productos que han tenido más de 50 ventas usando la cláusula `HAVING`.

```
SELECT
    ProductID,
    SUM(OrderQty) AS TotalQuantity
FROM
    [AdventureWorks2022].[Sales].[SalesOrderDetail]
GROUP BY
    ProductID
HAVING
    SUM(OrderQty) > 50;
```

RESULTS		
	ProductID	TotalQuantity
1	925	625
2	710	90
3	879	249
4	733	90
5	856	1616

3. Uso de `OVER` con `PARTITION BY`:

- Utiliza la tabla `Sales.SalesOrderDetail` para crear una consulta que muestre el `SalesOrderID`, `ProductID`, `LineTotal` y el valor total de ventas (`LineTotal`) para cada `SalesOrderID` usando la cláusula `OVER` con `PARTITION BY`.

```
SELECT
    SalesOrderID,
    ProductID,
    LineTotal,
    SUM(LineTotal) OVER (PARTITION BY SalesOrderID) AS TotalSalesValuePerOrder
FROM
    [AdventureWorks2022].[Sales].[SalesOrderDetail];
```

RESULTS				
	SalesOrderID	ProductID	LineTotal	TotalSalesValue...
1	43659	776	2024.994000	20565.620600
2	43659	777	6074.982000	20565.620600
3	43659	778	2024.994000	20565.620600
4	43659	771	2039.994000	20565.620600
5	43659	772	2039.994000	20565.620600

4. Numeración con ROW_NUMBER():

- Utiliza la tabla `Sales.SalesOrderDetail` para crear una consulta que muestre el `SalesOrderID`, `SalesOrderDetailID`, `ProductID` y `LineTotal`. Usa la función `ROW_NUMBER()` para numerar cada línea de pedido dentro de cada `SalesOrderID`.

```
SELECT
    SalesOrderID,
    SalesOrderDetailID,
    ProductID,
    LineTotal,
    ROW_NUMBER() OVER (PARTITION BY SalesOrderID ORDER BY SalesOrderDetailID) AS
    RowNumber
FROM
    [AdventureWorks2022].[Sales].[SalesOrderDetail];
```

RESULTS					
	SalesOrderID	SalesOrderDeta...	ProductID	LineTotal	RowNumber
1	43659	1	776	2024.994000	1
2	43659	2	777	6074.982000	2
3	43659	3	778	2024.994000	3
4	43659	4	771	2039.994000	4
5	43659	5	772	2039.994000	5

5. Asignación de rangos con RANK():

- Utiliza la tabla `Sales.SalesOrderDetail` para crear una consulta que muestre el `SalesOrderID`, `ProductID` y `LineTotal`. Asigna un rango a cada producto dentro de cada `SalesOrderID` basado en el `LineTotal` usando la función `RANK()`.

-- Utiliza la tabla Sales.SalesOrderDetail para crear una consulta que muestre el SalesOrderID, ProductID y LineTotal. Asigna un rango a cada producto dentro de cada SalesOrderID basado en el LineTotal usando la función RANK().

```
SELECT
    SalesOrderID,
    ProductID,
    LineTotal,
    RANK() OVER (PARTITION BY SalesOrderID ORDER BY LineTotal DESC) AS Rank
FROM
    [AdventureWorks2022].[Sales].[SalesOrderDetail];
```

▲ RESULTS				
	SalesOrderID	ProductID	LineTotal	Rank
1	43659	777	6074.982000	1
2	43659	773	4079.988000	2
3	43659	774	2039.994000	3
4	43659	771	2039.994000	3
5	43659	772	2039.994000	3

6. Asignación de rangos con DENSE_RANK():

- Utiliza la tabla Sales.SalesOrderDetail para crear una consulta que muestre el SalesOrderID, ProductID y LineTotal. Asigna un rango denso a cada producto dentro de cada SalesOrderID basado en el LineTotal usando la función DENSE_RANK().

-- Utiliza la tabla Sales.SalesOrderDetail para crear una consulta que muestre el SalesOrderID, ProductID y LineTotal. Asigna un rango denso a cada producto dentro de cada SalesOrderID basado en el LineTotal usando la función DENSE_RANK().

```
SELECT
    SalesOrderID,
    ProductID,
    LineTotal,
    DENSE_RANK() OVER (PARTITION BY SalesOrderID ORDER BY LineTotal DESC) AS
DenseRank
FROM
    [AdventureWorks2022].[Sales].[SalesOrderDetail];
```

RESULTS				
	SalesOrderID	ProductID	LineTotal	DenseRank
1	43659	777	6074.982000	1
2	43659	773	4079.988000	2
3	43659	774	2039.994000	3
4	43659	771	2039.994000	3
5	43659	772	2039.994000	3

7. Filtrado con HAVING y ordenación:

- Utiliza la tabla `Sales.SalesOrderDetail` para crear una consulta que agrupe los datos por `ProductID` y obtenga el valor total de ventas (`LineTotal`). Filtra para mostrar solo aquellos productos con un valor total de ventas superior a \$5000 usando la cláusula `HAVING`. Ordena los resultados por el valor total de ventas en orden descendente.

```

SELECT
    ProductID,
    SUM(LineTotal) AS TotalSalesValue
FROM
    [AdventureWorks2022].[Sales].[SalesOrderDetail]
GROUP BY
    ProductID
HAVING
    SUM(LineTotal) > 5000
ORDER BY
    TotalSalesValue DESC;

```

RESULTS		
	ProductID	TotalSalesValue
1	782	4400592.800400
2	783	4009494.761841
3	779	3693678.025272
4	780	3438478.860423
5	781	3434256.941928