

/\*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 TotalQuantity,
        SUM(LineTotal) AS TotalSales
FROM SALES.SalesOrderDetail
--WHERE ProductID = 925
GROUP BY ProductID;
```

	ProductID	TotalQuantity	TotalSales
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
6	856	1616	86166.045931
7	756	346	302678.724000
8	779	2394	3693678.025272
9	802	190	16897.080000
10	971	322	286218.660000
11	825	850	166013.283557

/\*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 SALES.SalesOrderDetail
GROUP BY ProductID
HAVING SUM(OrderQty) > 50
```

	ProductID	TotalQuantity
1	925	625
2	710	90
3	879	249
4	733	90
5	856	1616
6	756	346
7	779	2394
8	802	190
9	971	322
10	825	850
11	948	789

✓ Query executed successfully.

/\*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 TotalOrderValue
FROM SALES.SalesOrderDetail
```

	SalesOrderID	ProductID	LineTotal	TotalOrderValue
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
6	43659	773	4079.988000	20565.620600
7	43659	774	2039.994000	20565.620600
8	43659	714	86.521200	20565.620600
9	43659	716	28.840400	20565.620600
10	43659	709	34.200000	20565.620600
11	43659	712	10.373000	20565.620600

/\*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 (ORDER BY SalesOrderID) AS RowNum
FROM SALES.SalesOrderDetail
```

SalesOrderID	SalesOrderDetailID	ProductID	LineTotal	RowNum
43659	1	776	2024.994000	1
43659	2	777	6074.982000	2
43659	3	778	2024.994000	3
43659	4	771	2039.994000	4
43659	5	772	2039.994000	5
43659	6	773	4079.988000	6
43659	7	774	2039.994000	7
43659	8	714	86.521200	8
43659	9	716	28.840400	9
43659	10	709	34.200000	10
43659	11	712	10.373000	11

```
/*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 Sales.SalesOrderDetail;
```

	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	43659	778	2024.994000	6
7	43659	776	2024.994000	6
8	43659	714	86.521200	8
9	43659	711	80.746000	9
10	43659	709	34.200000	10
11	43659	716	28.840400	11

```
/*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 SALES.SalesOrderDetail;
```

	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
6	43659	778	2024.994000	4
7	43659	776	2024.994000	4
8	43659	714	86.521200	5
9	43659	711	80.746000	6
10	43659	709	34.200000	7
11	43659	716	28.840400	8

/\*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 TotalSales
FROM SALES.SalesOrderDetail
GROUP BY ProductID
HAVING SUM(LineTotal) > 5000
ORDER BY TotalSales DESC;
```

	ProductID	TotalSales
1	782	4400592.800400
2	783	4009494.761841
3	779	3693678.025272
4	780	3438478.860423
5	781	3434256.941928
6	784	3309673.216908
7	793	2516857.314918
8	794	2347655.953454
9	795	2012447.775000
10	753	1847818.628000
11	976	1774883.557085