
Universidad
Tecmilenio®

Universidad Tecmilenio

Consulta en Microsoft SQL Server®

Funciones de ventana en T-SQL

Semana 8

Las funciones ventana (*window functions*) son de gran apoyo en cuanto a análisis se refiere, ya que permiten tener una mejor visualización y presentación de la información que se consulta.

Dentro de estas funciones existen tres elementos principales: particionamiento de ventana, ordenación de ventana y marcos de ventana, asimismo, se deben considerar las funciones compatibles en su aplicación, por ejemplo, **SUM()** y **RANK()**.

Universidad Tecmilenio

Explicación

Descripción de las funciones de ventana

También conocidas como window functions, son aquellas que le permiten a cualquier usuario crear consultas avanzadas de modo analítico y de una manera muy operativa y eficiente.

Amazon Web Services (2022) explica que **las funciones ventana** operan en una partición o ventana de un conjunto de resultados y devuelven un valor para cada fila de esa ventana.

Explicación

Cláusula OVER

Esta cláusula determina un subconjunto de filas dentro de un conjunto de resultados de consulta. Se pueden aplicar funciones de clasificación o agregadas para establecer y calcular un valor para cada fila (Learn Tutorials, s.f.).

La **cláusula OVER** se utiliza en las funciones ventana y se basa en tres elementos:

- **Particionamiento** → **PARTITION**
- **Ordenación** → **ORDER BY**
- **Marcos**

Universidad Tecmilenio

Explicación

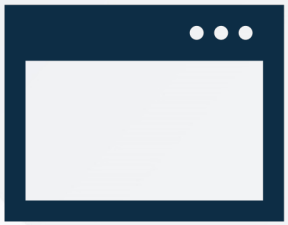
Funciones de agregación.

Funciones de clasificación.

Universidad Tecmilenio

Explicación

Uso de las funciones RANK, AGGREGATE y OFFSET



De acuerdo con Microsoft (2022), la función **RANK** devuelve el rango de cada fila dentro de la partición de un conjunto de resultados. El rango de una fila es uno más el número de rangos que preceden a la fila en cuestión.

Función AGGREGATE

Permitirá realizar operaciones sobre un conjunto de datos, en donde el resultado será un único valor agregado (Alarcón, 2021).

- COUNT.
- MIN.
- MAX.
- SUM.
- AVG.

Función OFFSET

Limitan el número de filas que serán devueltas al realizar una consulta.

- LAG.
- LEAD.
- FIRST_VALUE y LAST_VALUE.

Actividad

Con base en lo descrito en el tema, reflexiona sobre las siguientes preguntas:

01 Si se desea obtener información de los productos más vendidos en el año, ¿qué función sería ideal para aplicar y por qué?

02 Explica en qué escenario se pueden aplicar las funciones MIN y MAX al mismo tiempo.

Cierre

En este tema se aplican las distintas cláusulas de uso particular para las funciones ventana, también es importante resaltar el valor de conocer y saber emplear las funciones de agregación y de aplicación.

Utilizar este tipo de funciones nos permite tener como resultado el detalle de la información contenida en una tabla y no de manera agrupada, como se realiza en las consultas con las cláusulas tradicionales.

Bibliografía

• Alarcón, J. (2021). *Tutorial SQL #6: Agrupaciones y funciones de agregación*. Recuperado de https://www.campusmvp.es/recursos/post/Fundamentos-de-SQL-Agrupaciones-y-funciones-de-agregacion.aspx

• Amazon Web Services. (2022). *Window functions (Funciones de ventana)*. Recuperado de https://docs.aws.amazon.com/es_es/redshift/latest/dg/c_Window_functions.html

• Microsoft. (2022). *RANK (Transact-SQL)*. Recuperado de https://docs.microsoft.com/en-us/sql/t-sql/functions/rank-transact-sql

• Learn Tutorials. (s.f.). *Microsoft SQL Server Sobre la cláusula*. Recuperado de https://learntutorials.net/es/sql-server/topic/353/sobre-la-clausula

Universidad Tecmilenio

Consulta en Microsoft SQL Server®

Transformación de datos

Semana 8

Universidad Tecmilenio

Introducción

Dentro de las consultas tradicionales de T-SQL es posible agregar más cláusulas, de tal manera que la consulta sea más completa y cuando se muestren los datos de salida, estos traten de ser más completos y organizados.

Existe una similitud de estas cláusulas con la función “transponer” de Excel, que hace posible transformar las filas en columnas con la cláusula **PIVOT** y las columnas convertirlas nuevamente (o revertir) en filas con la cláusula **UNPIVOT**.

Universidad Tecmilenio

Explicación

Introducción a la transformación de datos mediante la implementación de PIVOT y UNPIVOT

Microsoft (2022) explica que **PIVOT** realiza una operación de agrupamiento en la tabla de entrada con respecto a las columnas de agrupamiento y devuelve una fila para cada grupo.

```
SELECT (ColumnNames)
FROM (TableName)
PIVOT
 (
   AggregateFunction(ColumnToBeAggregated)
   FOR PivotColumn IN (PivotColumnValues)
 ) AS (Alias) //Alias is a temporary name for a table
```

Uso de la cláusula PIVOT.

Universidad Tecmilenio

Explicación

Results | Messages | Execution plan | Client Statistics

	Num_Eje	Nombre	Departamento	Norte	Sur	Oriente
1	100	Rosas Gonzalez Alberto	Cajas	12000.000000	NULL	NULL
2	101	Carrillo López Tania	Ejecutivo	16000.000000	NULL	NULL
3	102	Rodríguez Esparza Lucía	Cajas	NULL	12000.000000	NULL
4	103	Cepeda Frias Carlos	Cajas	12000.000000	NULL	NULL
5	104	Vives Albarran Olga	Promotor	NULL	NULL	8000.000000
6	105	Esguerra Salinas Camila	Promotor	NULL	NULL	8000.000000
7	106	Ríos Alanis Esther	Ejecutivo	12000.000000	NULL	NULL
8	107	Prado Sosa Ivan	Promotor	NULL	NULL	8000.000000

Resultado del uso de la cláusula PIVOT.

La cláusula **UNPIVOT** permite transformar una tabla de entrada cambiando las columnas de los resultados por filas, facilitando al lector la interpretación de los datos (Amazon Web Services, 2022).

Se logra la combinación de los datos de sus columnas de entrada en dos columnas de resultados: una **columna de nombres** y la **columna de valores** contiene valores.

```
SELECT *
FROM
(
  SELECT column1, column2, ... columnN
  FROM tables
  WHERE conditions
)
UNPIVOT [INCLUDE|EXCLUDE NULLS]
(
  unpivot_clause
  FOR unpivot_for_clause
  IN (column1, column2, ... columnN)
)
ORDER BY expression [ ASC | DESC ];
```

Utilización de la cláusula UNPIVOT.

Explicación

Uso de PIVOT para dinamizar un conjunto de resultados

Las **tablas dinámicas** de T-SQL se consideran una manera de presentar un resumen de información, la cual se genera a partir de un conjunto de datos subyacente.

Algunas de las tablas dinamizadas se realizan para permitir que se analice la información contenida en esta, "principalmente para cortar los datos y generar consultas analíticas" (Das, 2020).

Universidad Tecmilenio

Explicación

Considerar lo siguiente para poder hacer un PIVOT dinámico:

- Realizar la consulta en forma de PIVOT.
- Declarar variables que te permitan almacenar los valores y que posteriormente serán ejecutados con la cláusula **Exec**.
- Al declarar una variable que permita almacenar la cadena generada a través de PIVOT, estás realizando una consulta (**query**) dentro de un *string*.
- Utilizar la cláusula **Exec** para realizar la ejecución de este tipo de consultas.

La cláusula Execute (Exec) se emplea de manera dinámica con la finalidad de ejecutar algún comando de SQL y asociarlo a cualquier variable, campo o arreglo.

Explicación

Uso de UNPIVOT para anular la dinamización de un conjunto de resultados

El uso de **UNPIVOT** es muy simple, debido a que esta cláusula realiza una rotación inversa a la cláusula PIVOT, lo que permite transponer columnas a filas.

Actividad

Con base en lo descrito en el tema, reflexiona sobre las siguientes preguntas:

01 ¿Cuál es la ventaja de usar el comando PIVOT?

02 La información muy pocas veces es estática, por lo tanto, siempre se debe pensar en el dinamismo. ¿En qué escenario vendría bien utilizar un PIVOT dinámico?

Cierre

Las cláusulas **PIVOT** y **UNPIVOT** son de gran apoyo en T-SQL, ya que muestran los resultados de una consulta de tal manera que sea más accesible para el usuario comprender cómo los datos pueden transformarse.

Ambas cláusulas realizan esta transformación de datos y esto permite presentar la información en un reporte de una forma más clara y detallada.

Bibliografía

• Amazon Web Services. (2022). *Cláusula FROM*. Recuperado de https://docs.aws.amazon.com/es_es/redshift/latest/dg/r_FROM_clause30.html

• Das, A. (2020). *Dynamic Pivot Tables in SQL Server*. Recuperado de https://www.sqlshack.com/dynamic-pivot-tables-in-sql-server/

• Microsoft. (2022). *Cláusula FROM más JOIN, APPLY, PIVOT (Transact-SQL)*. Recuperado de https://docs.microsoft.com/es-es/sql/t-sql/queries/from-transact-sql?view=sql-server-ver16

Universidad Tecmilenio

Consulta en Microsoft SQL Server®

Conjuntos de agrupación

Semana 8

Los operadores **GROUPING SETS, CUBE** y **ROLLUP** nos permiten, en algunos casos, optimizar y simplificar determinadas consultas.

GROUPING SETS: este operador es de gran ayuda con múltiples agrupaciones.

ROLLUP se considera muy útil para la generación de reportes en donde sea posible mostrar totales y subtotales.

El operador **CUBE** es capaz de generar un conjunto de resultados para todas las combinaciones posibles de los subtotales.

Universidad Tecmilenio

Explicación

Introducción a los conjuntos de agrupación mediante el uso de GROUPING SETS, CUBE y ROLLUP

GROUPING SETS es un operador que combina distintas consultas en una sola y se le considera una extensión de la cláusula GROUP BY (Bone, 2021).

```
SELECT
   aggregate_function(column_1)
   column_2,
   column_3,
FROM
   table_name
GROUP BY
   GROUPING SETS (
      (column_2, column_3),
      (column_2),
      (column_3),
      ()
);
```

Sintaxis de la cláusula GROUPING SETS.

Universidad Tecmilenio

Explicación

Uso de GROUPING SETS

```
Select
Sum(Sueldo)AS Sueldo_Eje, Num_Eje, Nombre, Sucursal, Departamento
FROM T_Eje
Group By
GROUPING SETS (
(Num_Eje, Nombre, Sucursal, Departamento), --SUMA DEL SUELDO POR LOS CAMPOS DESCRITOS
(Sucursal), --HACE LA AGRUPACIÓN POR LA COLUMNA ELEGIDA EN LA SALIDA
() --MUESTRA LA SUMA TOTAL
)
```

Uso del operador GROUPING SETS en la tabla T Eje.

Results | Messages

	Sueldo_Eje	Num_Eje	Nombre	Sucursal	Departamento	
1	12000.00	100	Rosas Gonzalez Alberto	Norte	Cajas	
2	16000.00	101	Camillo López Tania	Norte	Ejecutivo	
3	12000.00	103	Cepeda Frias Carlos	Norte	Cajas	
4	12000.00	106	Ríos Alanis Esther	Norte	Ejecutivo	
5	52000.00	NULL	NULL	Norte	NULL	Subtotal x Sucursal
6	8000.00	104	Vives Albarran Olga	Oriente	Promotor	
7	8000.00	105	Esguerra Salinas Camila	Oriente	Promotor	
8	8000.00	107	Prado Sosa Ivan	Oriente	Promotor	
9	24000.00	NULL	NULL	Oriente	NULL	Subtotal x Sucursal
10	12000.00	102	Rodríguez Esparza Lucía	Sur	Cajas	
11	12000.00	NULL	NULL	Sur	NULL	Subtotal x Sucursal
12	88000.00	NULL	NULL	NULL	NULL	Fila de Totales

Resultado del uso de GROUPING SETS.

Universidad Tecmilenio

Uso de CUBE y ROLLUP

```
Select
Sum(Sueldo)AS Sueldo_Eje, Num_Eje, Nombre, Departamento,
GROUPING (Departamento) AS Identificador
FROM T_Eje
Group By Num_Eje, Nombre, Departamento
With Rollup;
```

Uso del operador ROLLUP.

Explicación

Considera el ejemplo anterior para ver cómo **ROLLUP** y **CUBE** se emplean para la visualización de la información (SQLSERVER TUTORIAL, 2022).

Results | Messages

	Sueldo_Eje	Num_Eje	Nombre	Departamento	Identificador	
1	12000.00	100	Rosas Gonzalez Alberto	Cajas	0	
2	12000.00	100	Rosas Gonzalez Alberto	NULL	1	
3	12000.00	100	NULL	NULL	1	Subtotal 1
4	16000.00	101	Carrillo López Tania	Ejecutivo	0	
5	16000.00	101	Carrillo López Tania	NULL	1	
6	16000.00	101	NULL	NULL	1	Subtotal 2
7	12000.00	102	Rodríguez Esparza Lucía	Cajas	0	
8	12000.00	102	Rodríguez Esparza Lucía	NULL	1	
9	12000.00	102	NULL	NULL	1	Subtotal 3
10	12000.00	103	Cepeda Frias Carlos	Cajas	0	
11	12000.00	103	Cepeda Frias Carlos	NULL	1	
12	12000.00	103	NULL	NULL	1	Subtotal 4
13	8000.00	104	Vives Albarran Olga	Promotor	0	
14	8000.00	104	Vives Albarran Olga	NULL	1	
15	8000.00	104	NULL	NULL	1	Subtotal 5
16	8000.00	105	Esguerra Salinas Camila	Promotor	0	
17	8000.00	105	Esguerra Salinas Camila	NULL	1	
18	8000.00	105	NULL	NULL	1	Subtotal 6
19	12000.00	106	Ríos Alanis Esther	Ejecutivo	0	
20	12000.00	106	Ríos Alanis Esther	NULL	1	
21	12000.00	106	NULL	NULL	1	Subtotal 7
22	8000.00	107	Prado Sosa Ivan	Promotor	0	
23	8000.00	107	Prado Sosa Ivan	NULL	1	
24	8000.00	107	NULL	NULL	1	Subtotal 8
25	88000.00	NULL	NULL	NULL	1	Total

Resultado de uso de ROLLUP_con subtotales y total.

Universidad Tecmilenio

Explicación

```
Select
Sum(Sueldo)AS Sueldo_Eje, Num_Eje, Nombre, Departamento, Sucursal
FROM T_Eje
Group By Num_Eje, Nombre, Departamento, Sucursal
With CUBE;
```

Uso del operador CUBE.

Results | Messages

	Sueldo_Eje	Num_Eje	Nombre	Departamento	Sucursal
1	12000.00	103	Cepeda Frias Carlos	Cajas	Norte
2	12000.00	NULL	Cepeda Frias Carlos	Cajas	Norte
3	12000.00	100	Rosas Gonzalez Alberto	Cajas	Norte
4	12000.00	NULL	Rosas Gonzalez Alberto	Cajas	Norte
5	24000.00	NULL	NULL	Cajas	Norte
6	16000.00	101	Camillo López Tania	Ejecutivo	Norte
7	16000.00	NULL	Camillo López Tania	Ejecutivo	Norte
8	12000.00	106	Ríos Alanis Esther	Ejecutivo	Norte
9	12000.00	NULL	Ríos Alanis Esther	Ejecutivo	Norte
10	28000.00	NULL	NULL	Ejecutivo	Norte
11	52000.00	NULL	NULL	NULL	Norte
12	8000.00	105	Esguerra Salinas Camila	Promotor	Oriente
13	8000.00	NULL	Esguerra Salinas Camila	Promotor	Oriente
14	8000.00	107	Prado Sosa Ivan	Promotor	Oriente
15	8000.00	NULL	Prado Sosa Ivan	Promotor	Oriente
16	8000.00	104	Vives Albarran Olga	Promotor	Oriente
17	8000.00	NULL	Vives Albarran Olga	Promotor	Oriente
18	24000.00	NULL	NULL	Promotor	Oriente
19	24000.00	NULL	NULL	NULL	Oriente
20	12000.00	102	Rodríguez Esparza L...	Cajas	Sur
21	12000.00	NULL	Rodríguez Esparza L...	Cajas	Sur
22	12000.00	NULL	NULL	Cajas	Sur
23	12000.00	NULL	NULL	NULL	Sur
24	88000.00	NULL	NULL	NULL	NULL
25	12000.00	100	NULL	Cajas	Norte

Resultado de uso de CUBE.

Actividad

Con base en lo descrito en el tema, reflexiona sobre las siguientes preguntas:

01 Se desea obtener estadísticas de las ventas de todo el mes, ¿qué tipo de agrupaciones crees que le interese saber a la dirección general?

02 ¿Cómo explicarías el uso del comando CUBE en SQL Server?

Universidad Tecmilenio

Cierre

El uso de **GROUPING SETS** es de gran utilidad en consultas de tipo agrupamiento, debido a esto, se le considera una extensión de la cláusula GROUP BY, ya que la hace más eficiente y operacional por la manera en que es posible agrupar en diferentes conjuntos.

Junto con **GROUPING SETS**, está el empleo de los operadores ROLLUP, que hacen posible una salida de información en forma de reporte a manera de subtotales y totales.

CUBE genera subtotales con diversas combinaciones posibles en las columnas estipuladas con **GROUP BY**.

Bibliografía

• Bone, A. (2021). *What Is the SQL GROUPING SETS Clause, and How Do You Use it?* Recuperado de https://learnsql.com/blog/sql-grouping-sets-clause/

• SQLSERVER TUTORIAL.ORG. (2022). *SQL Server CUBE*. Recuperado de https://www.sqlservertutorial.org/sql-server-cube/