

# TUIA - BASES DE DATOS I

## FUNCIONES DE AGREGADO

### Documentación (SQL Server)

#### [Funciones de agregado \(Transact-SQL\)](#)

- Una función de agregado realiza un cálculo sobre un conjunto de valores y devuelve un solo valor.
- Las funciones de agregado se suelen usar con la cláusula GROUP BY de la instrucción SELECT.

# TUIA - BASES DE DATOS I

## FUNCIONES DE AGREGADO

- Todas las funciones de agregado son deterministas (devuelven el mismo valor cuando se las llama con el mismo conjunto de datos)
- Las funciones de agregado solo se pueden usar como expresiones en las situaciones siguientes:
  - La lista de selección de una instrucción SELECT (una subconsulta o una consulta externa).
  - La cláusula HAVING.

# TUIA - BASES DE DATOS I

## FUNCIONES DE AGREGADO

- APPROX\_COUNT\_DISTINCT
- AVG
- CHECKSUM\_AGG
- COUNT
- COUNT\_BIG
- GROUPING
- GROUPING\_ID
- MAX
- MIN
- STDEV
- STDEVP
- STRING\_AGG
- SUM
- VAR
- VARP

# TUIA - BASES DE DATOS I

## FUNCIONES DE AGREGADO

Ejemplos:

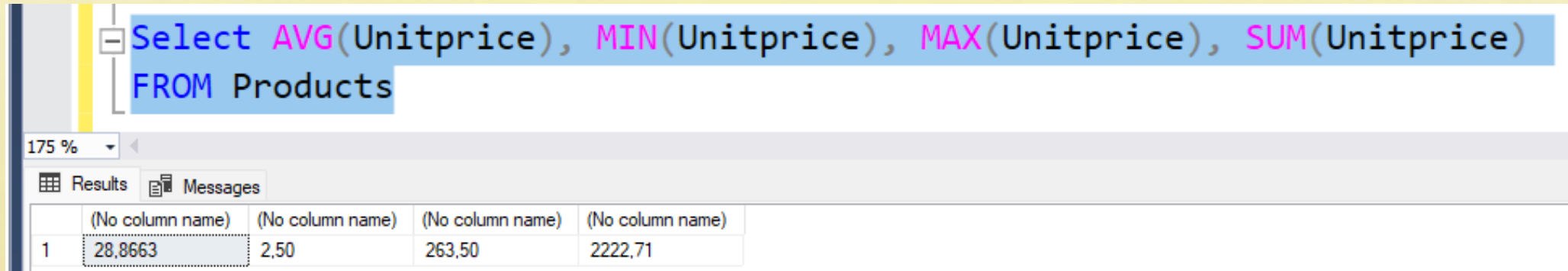
```
Select AVG(Unitprice) FROM Products
Select MIN(Unitprice) FROM Products
Select MAX(Unitprice) FROM Products
Select SUM(Unitprice) FROM Products
```

(No column name)	
1	28,8663
(No column name)	
1	2,50
(No column name)	
1	263,50
(No column name)	
1	2222,71

# TUIA - BASES DE DATOS I

## FUNCIONES DE AGREGADO

Ejemplos:



```
Select AVG(Unitprice), MIN(Unitprice), MAX(Unitprice), SUM(Unitprice)
FROM Products
```

175 %

Results Messages

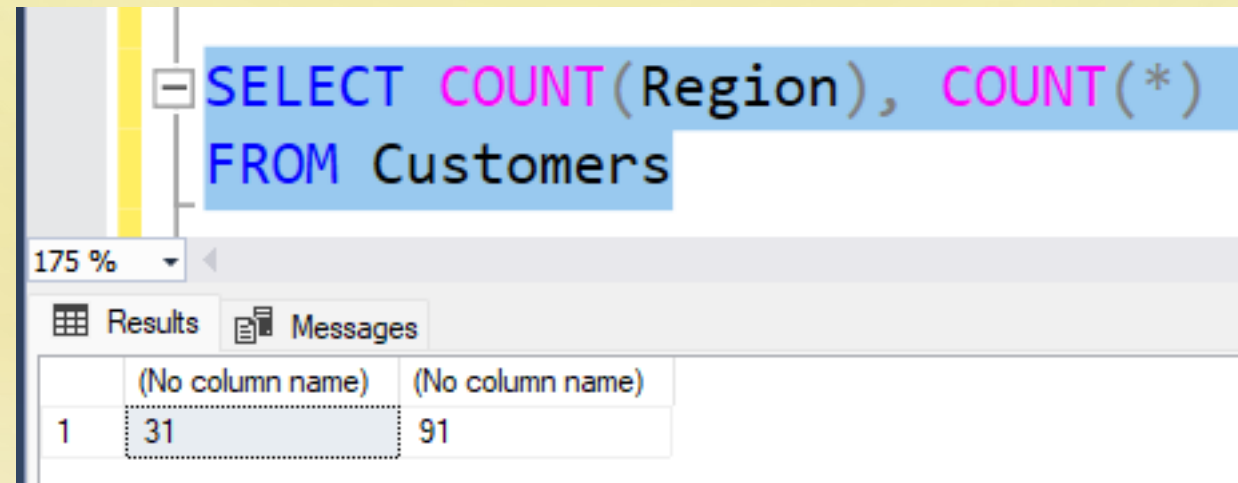
	(No column name)	(No column name)	(No column name)	(No column name)
1	28,8663	2,50	263,50	2222,71

# TUIA - BASES DE DATOS I

## FUNCIONES DE AGREGADO

### COUNT()

- COUNT(NombreCampo) – Cuenta todos los registros de donde NombreCampo es distinto de NULL.
- COUNT(\*) – Cuenta todos los **registros**.



The screenshot shows a SQL query editor with a query window and a results window. The query is: `SELECT COUNT(Region), COUNT(*) FROM Customers`. The results window shows a single row with two columns, both labeled "(No column name)". The first column contains the value 31, and the second column contains the value 91.

	(No column name)	(No column name)
1	31	91

# TUIA - BASES DE DATOS I

## FUNCIONES DE AGREGADO

- Las funciones de agregado devuelven UN valor tomando varias filas como entrada (son como un “resumen”).
- NO puedo mezclarlas en el SELECT con otras columnas (a no ser que agrupe por las mismas...)

```
SELECT Country, COUNT(*)  
FROM Customers
```

Msg 8120, Level 16, State 1, Line 18

Column 'Customers.Country' is invalid in the select list because it is not contained in either an aggregate function or a table reference.

# TUIA - BASES DE DATOS I

## CLÁUSULA GROUP BY

- Cláusula de la instrucción SELECT que divide el resultado de la consulta en grupos de filas, normalmente con el fin de realizar una o varias agregaciones en cada grupo.
- La instrucción SELECT devuelve una fila por grupo.

Documentación:

[GROUP BY \(Transact-SQL\) - SQL Server](#)



# TUIA - BASES DE DATOS I

## CLÁUSULA GROUP BY

Sintaxis:

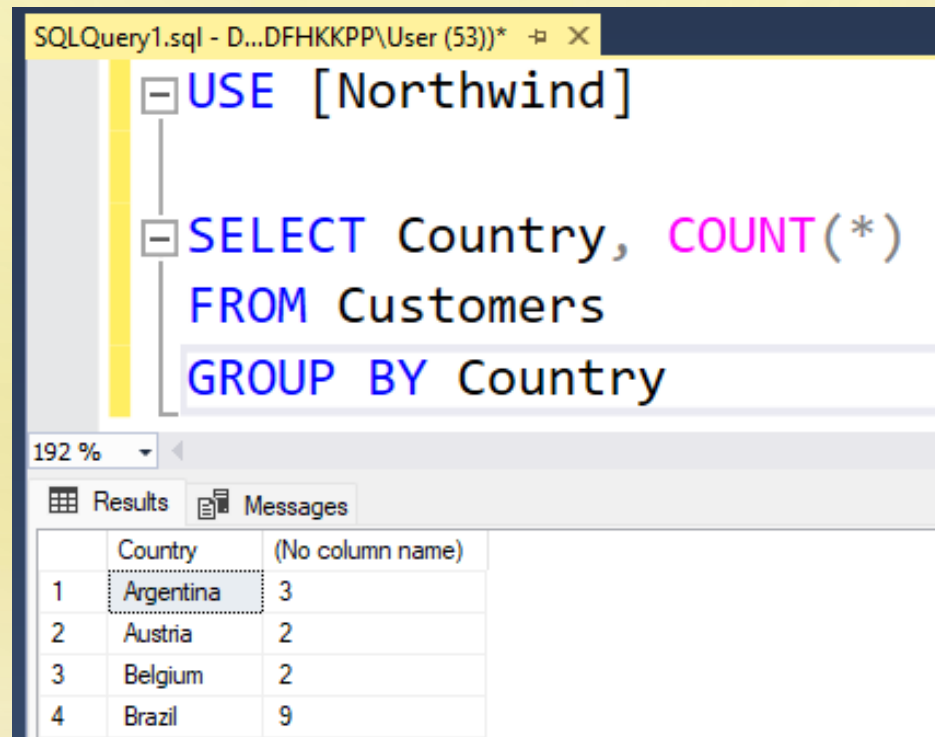
```
-- Syntax for SQL Server and Azure SQL Database
-- ISO-Compliant Syntax

GROUP BY {
    column-expression
    | ROLLUP ( <group_by_expression> [ ,...n ] )
    | CUBE ( <group_by_expression> [ ,...n ] )
    | GROUPING SETS ( <grouping_set> [ ,...n ] )
    | () --calculates the grand total
} [ ,...n ]
```

# TUIA - BASES DE DATOS I

## CLÁUSULA GROUP BY

En el select\_list podemos utilizar los campos agrupados y funciones de agregado:



The screenshot shows a SQL query window titled 'SQLQuery1.sql - D...DFHKKPP\User (53))' with the following SQL code:

```
USE [Northwind]

SELECT Country, COUNT(*)
FROM Customers
GROUP BY Country
```

Below the query, the 'Results' tab is active, displaying a table with 4 rows and 3 columns. The columns are 'Country', '(No column name)', and an unlabeled column. The data is as follows:

	Country	(No column name)	
1	Argentina	3	
2	Austria	2	
3	Belgium	2	
4	Brazil	9	

# TUIA - BASES DE DATOS I

## CLÁUSULA GROUP BY

- Es posible agrupar por más de un campo.
- Podemos ordenar por cualquier campo del select\_list, incluyendo las funciones de agregado.

```
SELECT Country, Region , COUNT(*) AS Cant  
FROM Customers  
GROUP BY Country, Region  
ORDER BY Country, Region
```

192 %

Results Messages

	Country	Region	Cant
1	Argentina	NULL	3
2	Austria	NULL	2
3	Belgium	NULL	2
4	Brazil	RJ	3
5	Brazil	SP	6

# TUIA - BASES DE DATOS I

## CLÁUSULA HAVING

- Especifica una condición de búsqueda para un grupo o agregado. Sólo se puede utilizar con la instrucción SELECT.
- Normalmente HAVING se usa con una cláusula GROUP BY.
- Cuando no se usa GROUP BY, hay un solo grupo implícito agregado.

### **Documentación:**

[HAVING \(Transact-SQL\) - SQL Server](#)

# TUIA - BASES DE DATOS I

## CLÁUSULA HAVING

- Así como la cláusula WHERE establece un filtro o condición para las filas, HAVING lo hace para los agrupamientos

```
SELECT Country, Region, COUNT(*) AS Cant
FROM Customers
WHERE Region IS NOT NULL
GROUP BY Country, Region
HAVING COUNT(*) > 3
ORDER BY Country
```

192 %

Results Messages

	Country	Region	Cant
1	Brazil	SP	6
2	USA	OR	4

# TUIA - BASES DE DATOS I

## SUBCONSULTAS

- Una subconsulta es una consulta anidada en una instrucción SELECT, INSERT, UPDATE o DELETE, o bien en otra subconsulta.
- Las subconsultas se pueden utilizar en cualquier parte en la que se permita una expresión.
- Podemos encontrarla en el **select\_list**, el **FROM** o el **WHERE** por ejemplo.

# TUIA - BASES DE DATOS I

## SUBCONSULTAS

En este caso la utilizamos en el **select\_list** para traer información almacenada en otra tabla:

```
SELECT ProductID, ProductName, CategoryID,  
    (SELECT CategoryName  
     FROM Categories  
     WHERE CategoryID = Products.CategoryID ) AS Categoria  
FROM Products
```

	ProductID	ProductName	CategoryID	Categoria
1	1	Chai	1	Beverages
2	2	Chang	1	Beverages
3	3	Aniseed Syrup	2	Condiments
4	4	Chef Anton's Cajun Seasoning	2	Condiments

# TUIA - BASES DE DATOS I

## FUNCIÓN EXISTS

Esta función devuelve verdadero o falso dependiendo si la subconsulta incluida en ella devuelve filas o no.

Sintaxis:

**EXISTS** ( subquery )



# TUIA - BASES DE DATOS I

## FUNCIÓN EXISTS

El siguiente ejemplo devuelve 'Verdadero' ya que la subconsulta devuelve filas

```
if ( EXISTS(  
    SELECT CategoryName  
    FROM Categories  
    WHERE 1 = 1  
    )  
) SELECT 'Verdadero'
```

192 %

Results Messages

	(No column name)
1	Verdadero

# TUIA - BASES DE DATOS I

## FUNCIÓN EXISTS

### Ejemplo

- Mostrar los clientes de la tabla Customers que NO tienen órdenes en la tabla Orders:

```
SELECT *  
FROM Customers  
WHERE NOT EXISTS ( SELECT *  
                    FROM Orders  
                    WHERE CustomerID = Customers.CustomerID )
```

192 %

Results Messages

	CustomerID	CompanyName	ContactName	ContactTitle	Address	City	Region	PostalCode	Country	Phone	Fax
1	FISSA	FISSA Fabrica Inter. Salchichas S.A.	Diego Roel	Accounting Manager	C/ Morazarzal, 86	Madrid	NULL	28034	Spain	(91) 555 94 44	(91) 555 55 93
2	PARIS	Paris spécialités	Marie Bertrand	Owner	265, boulevard Charonne	Paris	NULL	75012	France	(1) 42.34.22.66	(1) 42.34.22.77

# TUIA - BASES DE DATOS I

## RECURSOS ADICIONALES

- Base de Datos ADVENTUREWORKS

Es una base de datos de ejemplo con numerosas tablas y un volumen de datos interesante. Utilizar la base de tipo OLTP según la versión de SQL Server que tengan instalado, se pueden descargar desde:

<https://learn.microsoft.com/es-es/sql/samples/adventureworks-install-configure?view=sql-server-ver16&tabs=ssms>

- Tutorial de SQL Server Management Studio

Permite aprender a usar la herramienta desde cero. Disponible en:

<https://learn.microsoft.com/es-es/sql/ssms/quickstarts/ssms-connect-query-sql-server?view=sql-server-ver16>