

**UNIVERSIDAD PERUANA LOS ANDES FACULTAD DE
INGENIERÍA**



Asignatura: Base de Datos II

Docente: Dr. Raúl Enrique Fernández Bejarano

Alumno: SORIANO TIMOTEO Joel Kevin

Ciclo: V

huancayo-Perú-2025

Manual de implementación de consultas

Primera parte de las consultas:

Funciones de Agregación en SQL: Explicación Detallada

Las Funciones de Agregación son esenciales en el análisis de datos. Su principal característica es que **operan verticalmente** sobre un conjunto de valores (una columna) y producen un **único valor de salida** que representa el resumen de esa operación.

1. Las Cinco Funciones Fundamentales

Función	Propósito	Aplicación Típica	Notas Importantes
COUNT()	Cuenta el número de filas o valores.	Saber el tamaño de un mercado, cuántos pedidos se hicieron.	Se puede usar COUNT(*) (cuenta todas las filas) o COUNT(columna) (cuenta solo valores NO nulos).
SUM()	Devuelve la suma total de valores.	Calcular los ingresos totales, el inventario valorado total.	Solo funciona con columnas numéricas . Ignora automáticamente los valores nulos (NULL).
AVG()	Calcula el valor promedio.	Determinar el salario promedio, el tiempo promedio de respuesta.	Solo funciona con columnas numéricas . El promedio es $\frac{\text{SUM}(X)}{\text{COUNT}(X \text{ no nulos})}$.

Función	Propósito	Aplicación Típica	Notas Importantes
MIN()	Devuelve el valor más pequeño.	Encontrar el producto más barato, la fecha de registro más antigua.	Funciona con columnas numéricas, de texto (orden alfabético) y de fecha .
MAX()	Devuelve el valor más grande.	Encontrar el producto más caro, la fecha de registro más reciente.	Funciona con columnas numéricas, de texto y de fecha .

2. Uso Avanzado con GROUP BY y HAVING

El poder real de las funciones de agregación se manifiesta al agrupar datos.

Cláusula GROUP BY

- **Función:** Agrupa filas que tienen los mismos valores en una o más columnas. La función de agregación se aplica **independientemente** a cada grupo formado.
- **Regla esencial:** Si usas una función de agregación en la cláusula SELECT, cualquier columna que **no** esté dentro de una función de agregación **debe** estar listada en la cláusula GROUP BY.
 - *Incorrecto:* SELECT Departamento, AVG(Salario) FROM Empleados; (SQL no sabría qué departamento mostrar).

- Correcto: `SELECT Departamento, AVG(Salario) FROM Empleados GROUP BY Departamento;`

Cláusula HAVING

- **Función:** Es el equivalente del WHERE pero se usa para **filtrar grupos** de resultados después de que la agregación ha ocurrido. No puedes usar una función de agregación en la cláusula WHERE.
- **Ejemplo:** Mostrar solo los departamentos donde el salario promedio es superior a \$50,000.

SQL

```
SELECT Departamento, AVG(Salario) AS SalarioPromedio  
FROM Empleados  
GROUP BY Departamento  
HAVING AVG(Salario) > 50000; -- Filtra los grupos
```

3. Consideración de Valores Nulos (NULL)

Es crucial entender cómo manejan los valores nulos:

- **SUM(), AVG(), MIN(), MAX():** Estas funciones **ignoran** los valores NULL en sus cálculos. Por ejemplo, al calcular el promedio (AVG), los valores nulos no se cuentan en el denominador.
- **COUNT():**
 - COUNT(*)**:** Cuenta **todas** las filas, incluyendo aquellas con valores nulos, pues cuenta la existencia de la fila.
 - COUNT(nombre_columna)**:** Solo cuenta las filas donde nombre_columna **no es nulo**.

4. Uso de DISTINCT

Puedes combinar la palabra clave DISTINCT dentro de las funciones COUNT, SUM y AVG para que operen **solo sobre valores únicos** dentro de la columna.

Ejemplo: Contar cuántos títulos de trabajo únicos existen.

SQL

```
SELECT COUNT(DISTINCT TituloTrabajo)  
FROM Empleados;
```

Esto evita que se cuenten múltiples empleados con el mismo título de trabajo, dándote la diversidad de roles en lugar del número de empleados.

FUNCIONES DE AGREGACIÓN

- I. Explica de manera clara y didáctica qué son las **Funciones de Agregación** en SQL y cómo se utilizan.
-

1. Mostrar CodArticulo, DescripcionArticulo y ValorInventario.

Salida de cada uno de los casos:

Enunciado: Mostrar CodArticulo, DescripcionArticulo y ValorInventario.

Consulta SQL:

```
SELECT  
    A.CodArticulo,  
    A.DescripcionArticulo,  
    CAST(A.StockActual * CAST(A.PrecioProveedor AS DECIMAL(18,2)) AS DECIMAL(18,2)) AS  
    ValorInventario  
FROM ARTICULO A;
```

Explicación: Multiplica stock por precio por fila; castea para precisión.

2. Calcular el total monetario del inventario.
3. Obtener CodLinea y PrecioProveedor promedio.
4. Contar artículos descontinuados.
5. Mostrar PrecioMaximo y PrecioMinimo del catálogo.
6. Mostrar el Valor total enviado por guía.
7. Para cada CodArticulo, mostrar TotalSolicitado.
8. Contar órdenes únicas que incluyen cada artículo.
9. Calcular promedio de días por todas las órdenes con FechaIngreso.
10. Sumar CantidadEnviada por CodTransportista.

Consultas SQL y Explicaciones (Texto Continuo)

1. Mostrar CodArticulo, DescripcionArticulo y ValorInventario.

Enunciado: Mostrar CodArticulo, DescripcionArticulo y ValorInventario.

Código SQL:

-- 1. Mostrar CodArticulo, DescripcionArticulo y ValorInventario. (La primera consulta solicitada)

SELECT

A.CodArticulo, A.DescripcionArticulo,

-- Multiplica StockActual * PrecioProveedor

CAST(A.StockActual * CAST(A.PrecioProveedor AS

DECIMAL(18,2)) AS DECIMAL(18,2)) AS ValorInventario

FROM

ARTICULO A;

GO

The screenshot shows the SSMS interface with the following details:

- File Explorer (Left):** Shows the database structure under "ALBERT JEANKARLO CHUQUIYAHUA...". The "Tables" node is expanded, and "dbo.ARTICULO" is selected.
- Object Explorer (Left):** Shows the same database structure, with "dbo.ARTICULO" also selected.
- SQL Editor (Center):** Displays the SQL query from the code block above. The results of the query are shown in the "Results" tab below.
- Results Tab (Bottom):** Contains a table with 10 rows of data. The columns are: CodArticulo, DescripcionArticulo, and ValorInventario.

	CodArticulo	DescripcionArticulo	ValorInventario
1	1	Filtro Hepa Mod-2 Sku-1	156.00
2	2	Filtro Hepa Mod-3 Sku-2	189.00
3	3	Adhesivo Epóxico Mod-4 Sku-3	224.00
4	4	Filtro Hepa Mod-5 Sku-4	261.00
5	5	Adhesivo Epóxico Mod-6 Sku-5	300.00
6	6	Válvula Flujo Mod-7 Sku-6	341.00
7	7	Sensor Óptico Mod-8 Sku-7	384.00
8	8	Filtro Hepa Mod-9 Sku-8	429.00
9	9	Válvula Flujo Mod-10 Sku-9	476.00
10	10	Malla Industrial Mod-1 Sku-10	525.00

```

USE QhatuPERU;
GO
-- 1. Mostrar CodArticulo, DescripcionArticulo y ValorInventario. (La primera consulta)
SELECT
    A.CodArticulo, A.DescripcionArticulo,
    -- Multiplica StockActual * PrecioProveedor
    CAST(A.StockActual * CAST(A.PrecioProveedor AS DECIMAL(18,2)) AS DECIMAL(18,2))
FROM
    ARTICULO A;
GO

```

	CodArticulo	DescripcionArticulo	ValorInventario
191	191	Interruptor Termico Mod-2 Sku-191	42336.00
192	192	Malla Industrial Mod-3 Sku-192	42749.00
193	193	Cable Cat 6 Mod-4 Sku-193	43164.00
194	194	Válvula Flujo Mod-5 Sku-194	43581.00
195	195	Rodamiento Z-20 Mod-6 Sku-195	44800.00
196	196	Válvula Flujo Mod-7 Sku-196	44423.00
197	197	Rodamiento Z-20 Mod-8 Sku-197	44844.00
198	198	Rodamiento Z-20 Mod-9 Sku-198	45269.00
199	199	Aceite Sintético Mod-10 Sku-199	45696.00
200	200	Sensor Óptico Mod-1 Sku-200	46125.00

Explicación: Esta consulta multiplica el StockActual por el PrecioProveedor para obtener el valor de inventario de cada artículo. Se utiliza CAST para asegurar la precisión decimal adecuada en el resultado monetario.

2. Calcular el total monetario del inventario.

Enunciado: Calcular el total monetario del inventario.

Código SQL:

-- 2. Calcular el total monetario del inventario.

SELECT

```

CAST(SUM(A.StockActual * CAST(A.PrecioProveedor AS
DECIMAL(18,2))) AS MONEY) AS TotalInventarioMonetario
FROM
    ARTICULO A;
GO

```

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left, the Object Explorer pane displays a tree view of database connections, databases, and tables. The 'Tables' node for the 'dbo' schema is selected, showing the 'ARTICULO' table. The main window contains a query editor with the following SQL code:

```

-- 2. Calcular el total monetario del inventario.
SELECT
    CAST(SUM(A.StockActual * CAST(A.PrecioProveedor AS DECIMAL(18,2)) AS MONEY) AS TotalInventarioMonetario
FROM
    ARTICULO A;
GO

```

Below the query editor is a results grid titled 'Results'. It lists various inventory items with their codes, descriptions, and current stock values. The last row of the grid is a summary row labeled 'TotalInventarioMonetario' with a value of 3314700.00.

	CodArticulo	DescripcionArticulo	StockActual	PrecioProveedor	TotalInventarioMonetario
191	191	Interruptor Térmico Mod-2 Sku-191	100	42336.00	42336.00
192	192	Malla Industrial Mod-3 Sku-192	100	42749.00	42749.00
193	193	Cable Cat 6 Mod-4 Sku-193	100	43164.00	43164.00
194	194	Válvula Flujo Mod-5 Sku-194	100	43581.00	43581.00
195	195	Rodamiento Z-20 Mod-6 Sku-195	100	44800.00	44800.00
196	196	Válvula Flujo Mod-7 Sku-196	100	44421.00	44421.00
197	197	Rodamiento Z-20 Mod-8 Sku-197	100	44844.00	44844.00
198	198	Rodamiento Z-20 Mod-9 Sku-198	100	45269.00	45269.00
199	199	Aceite Sintético Mod-10 Sku-199	100	45696.00	45696.00
200	200	Sensor Óptico Mod-1 Sku-200	100	46125.00	46125.00
		TotalInventarioMonetario			3314700.00
1					

Explicación: Se utiliza la función de agregación SUM() para sumar el valor de inventario calculado de *todos* los artículos, obteniendo el valor total del inventario en todo el catálogo.

3. Obtener CodLinea y PrecioProveedor promedio.

Enunciado: Obtener CodLinea y PrecioProveedor promedio.

Código SQL:

-- 3. Obtener CodLinea y PrecioProveedor promedio.

SELECT

 A.CodLinea,

 CAST(AVG(A.PrecioProveedor) AS DECIMAL(18,2)) AS

 PrecioProveedorPromedio

FROM

 ARTICULO A

GROUP BY

 A.CodLinea;

GO

File Edit View Help ← → ⌂ Search

CONNECTIONS: AZURE + ⌂ ... connected SQLQuery_1.sql - (75) t...U (sa1) consulta_01.sql

C:\> Users > User > Downloads > actividad 05 > consulta_01.sql

Run Cancel Disconnect Change Database: QhatuPERU Estimated Plan

Enable Actual Plan Parse

```

23
24  -- 3. Obtener CodLinea y PrecioProveedor promedio.
25  SELECT
26      A.CodLinea,
27      CAST(AVG(A.PrecioProveedor) AS DECIMAL(18,2)) AS PrecioProveedorPromedio
28  FROM
29      ARTICULO A
30  GROUP BY
31      A.CodLinea;
32  GO
33
34

```

Results Messages

	CodLinea	PrecioProveedorPromedio
1	1	3.00
2	2	3.50
3	3	4.00
4	4	4.50
5	5	5.00
6	6	5.50
7	7	6.00
8	8	6.50
9	9	7.00
10	10	7.50

File Edit View Help ← → ⌂ Search

CONNECTIONS: AZURE + ⌂ ... connected SQLQuery_1.sql - (75) t...U (sa1) consulta_01.sql

C:\> Users > User > Downloads > actividad 05 > consulta_01.sql

Run Cancel Disconnect Change Database: QhatuPERU Estimated Plan

Enable Actual Plan Parse

```

23
24  -- 3. Obtener CodLinea y PrecioProveedor promedio.
25  SELECT
26      A.CodLinea,
27      CAST(AVG(A.PrecioProveedor) AS DECIMAL(18,2)) AS PrecioProveedorPromedio
28  FROM
29      ARTICULO A
30  GROUP BY
31      A.CodLinea;
32  GO
33
34

```

Results Messages

	CodLinea	PrecioProveedorPromedio
191	191	98.00
192	192	98.50
193	193	99.00
194	194	99.50
195	195	100.00
196	196	100.50
197	197	101.00
198	198	101.50
199	199	102.00
200	200	102.50

Explicación: Esta consulta agrupa los artículos por CodLinea y calcula el precio promedio del proveedor (AVG()) para todos los

4. Contar artículos descontinuados.

Enunciado: Contar artículos descontinuados.

Código SQL:

-- 4. Contar artículos descontinuados.

```
SELECT  
    COUNT(CodArticulo) AS TotalArticulosDescontinuados  
FROM  
    ARTICULO  
WHERE  
    Descontinuado = 1;  
GO
```

The screenshot shows the SSMS interface with the following details:

- File Edit View Help** menu at the top.
- CONNECTIONS: AZURE** in the left sidebar.
- ALBERT JEANKARLO CHUQUINA...** under Azure for Students.
- SQL databases** node expanded, showing databases: **cafesito (jean3)**, **master (jean3)**, **master (trabajo)**, **QhatuPERU (jean3)**.
- Tables** node expanded under **QhatuPERU (jean3)**, showing tables: **dbo.ARTICULO**, **dbo.GUIA_DETALLE**, **dbo.GUIA_ENVIO**, **dbo.UNEA**, **dbo.ORDEN_COMPRA**, **dbo.ORDEN_DETALLE**, **dbo.PROVEEDOR**, **dbo.TIENDA**, **dbo.TRANSPORTISTA**, **Dropped Ledger Tables**, **Views**, **Synonyms**, **Programmability**, **External Resources**, **Storage**, **Security**, **trabajo_05 (jean3)**, and **SQL servers**.
- SQLQuery_1.sql - (66 t...U (sa1))** and **consulta_01.sql - (54 t...U (sa1))** in the top right.
- Run**, **Cancel**, **Disconnect**, **Change**, **Database: QhatuPERU**, **Estimated Plan** buttons in the top right.
- Enable Actual Plan** and **Parse** checkboxes.
- Code Editor:**

```
35  
36 -- 4. Contar articulos descontinuados.  
37 SELECT  
38     COUNT(CodArticulo) AS TotalArticulosDescontinuados  
39 FROM  
40     ARTICULO  
41 WHERE  
42     Descontinuado = 1;  
43 GO  
44  
45  
46  
47 -- 5. Mostrar PrecioMaximo y PrecioMinimo del catalogo.  
48
```
- Results** tab selected, showing the result of the query:

TotalArticulosDescontinuados
1
5
- Messages** tab.

Explicación: Se filtra la tabla ARTICULO usando WHERE Descontinuado = 1 y luego se usa COUNT() para contar el número total de artículos marcados como descontinuados.

5. Mostrar PrecioMaximo y PrecioMinimo del catálogo.

Enunciado: Mostrar PrecioMaximo y PrecioMinimo del catálogo.

Código SQL:

-- 5. Mostrar PrecioMaximo y PrecioMinimo del catálogo.

SELECT

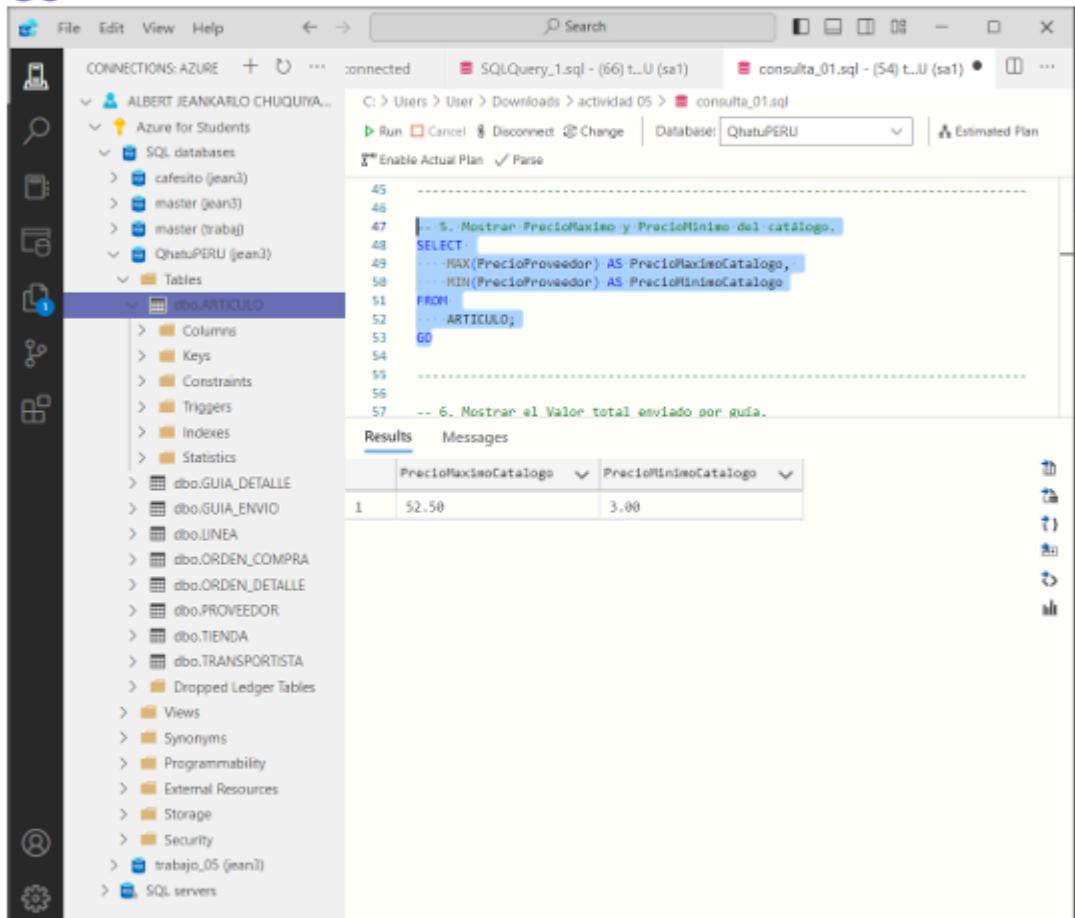
MAX(PrecioProveedor) **AS** PrecioMaximoCatalogo,

MIN(PrecioProveedor) **AS** PrecioMinimoCatalogo

FROM

ARTICULO;

GO



The screenshot shows the SSMS interface with the following details:

- Object Explorer:** Shows the database structure under "ALBERT JEANKARLO CHUQUIAYA...". The "Tables" node is expanded, showing "dbo.ARTICULO" and its columns.
- Query Editor:** Contains the SQL code for question 5. The results pane shows the output of the query.
- Results:** A table with two columns: "PrecioMaximoCatalogo" and "PrecioMinimoCatalogo". The value for "PrecioMaximoCatalogo" is 52.50 and for "PrecioMinimoCatalogo" is 3.00.

Explicación: Se utilizan las funciones de agregación MAX() y MIN() sobre la columna PrecioProveedor para identificar los precios más alto y más bajo de todos los artículos en el catálogo.

6. Mostrar el Valor total enviado por guía.

Enunciado: Mostrar el Valor total enviado por guía.

Código SQL:

Explicación: Agrupa los detalles de envío por NumGuia y suma el valor monetario de los productos enviados (CantidadEnviada * PrecioVenta) para determinar el valor total de cada guía.

-- 6. Mostrar el Valor total enviado por guía.

SELECT

```
GD.NumGuia,
CAST(SUM(GD.CantidadEnviada * GD.PrecioVenta) AS
MONEY) AS ValorTotalEnviado
FROM
    GUIA_DETALLE GD
GROUP BY
    GD.NumGuia;
```

GO

The screenshot shows the SSMS interface with the following details:

- Left pane (Object Explorer):** Shows the database structure under 'ALBERT JEANKARLO CHUQUIAYA...'. The 'Tables' node is expanded, and 'dbo.ARTICULO' is selected.
- Right pane (Query Editor):**
 - Shows the query code:

```
-- 6. Mostrar el Valor total enviado por guía.
SELECT
    GD.NumGuia,
    CAST(SUM(GD.CantidadEnviada * GD.PrecioVenta) AS MONEY) AS ValorTotalEnviado
FROM
    GUIA_DETALLE GD
GROUP BY
    GD.NumGuia;
GO
```
 - Execution status: 'connected' and 'Run' button is highlighted.
 - Database: 'QhatuPERU'
 - Results tab: Displays the output of the query as a table.

	NumGuia	ValorTotalEnviado
1	4001	196.10
2	4002	246.40
3	4003	300.90
4	4004	359.60
5	4005	422.50
6	4006	489.60
7	4007	560.90
8	4008	636.40
9	4009	716.10
10	4010	800.00
11	4011	888.10
12	4012	980.40
13	4013	1076.90
14	4014	1177.60
15	4015	1282.50
16	4016	1301.60

The screenshot shows the SSMS interface. In the object browser on the left, under 'Azure for Students' and 'QhatuPERU', the 'Tables' node is expanded, and 'dbo.ARTICULO' is selected. The main pane displays a T-SQL script:

```

54
55
56
57 -- 6. Mostrar el Valor total enviado por guías.
58 SELECT
59     GO.NumGuia,
60     CAST(SUM(GO.CantidadEnvuada * GO.PrecioVenta) AS MONEY) AS ValorTotalEnviado
61 FROM
62     GUIA_DETALLE GD
63 GROUP BY
64     GO.NumGuia;
65 GO
66

```

The 'Results' tab is selected, showing a table with two columns: 'NumGuia' and 'ValorTotalEnviado'. The data is as follows:

NumGuia	ValorTotalEnviado
85	19862.50
86	19465.60
87	19872.90
88	20284.40
89	20700.10
90	21120.00
91	21544.10
92	21972.40
93	22484.90
94	22841.60
95	23282.50
96	23727.60
97	24176.90
98	24630.40
99	25088.10
100	25900.00

7. Para cada CodArticulo, mostrar TotalSolicitado.

Enunciado: Para cada CodArticulo, mostrar TotalSolicitado.

Código SQL:

-- 7. Para cada CodArticulo, mostrar TotalSolicitado.

```

SELECT
    OD.CodArticulo,
    SUM(OD.CantidadSolicitada) AS TotalSolicitado
FROM
    ORDEN_DETALLE OD
GROUP BY
    OD.CodArticulo
ORDER BY
    OD.CodArticulo;
GO

```

File Edit View Help ← → Search

CONNECTIONS: AZURE + connected

SQLQuery_1.sql - (66 t...U (sa1)) consulta_01.sql - (54 t...U (sa1))

Run Cancel Disconnect Change Database: QhatuPERU Estimated Plan

67
68
69 ... 7. Para cada CodArticulo, mostrar TotalSolicitado.
70 SELECT
71 OD.CodArticulo,
72 ... SUM(OD.CantidadSolicitada) AS TotalSolicitado
73 FROM
74 ... ORDEN_DETALLE OD
75 GROUP BY
76 OD.CodArticulo
77 ORDER BY
78 OD.CodArticulo;
79 GO

Results Messages

	CodArticulo	TotalSolicitado
1	1	105
2	2	110
3	3	115
4	4	120
5	5	125
6	6	130
7	7	135
8	8	140
9	9	145
10	10	150
11	11	155
12	12	160
13	13	165
14	14	170
15	15	175
16	16	180

Server: A - UFT-0 CPU: SQL 100 users Choose SQL language: 00:00:00 last used database windows.net - QhatuPERU

The screenshot shows the SSMS interface. On the left is the Object Explorer tree, which is expanded to show 'AZURE' and 'QhatuPERU' database. Under 'QhatuPERU', 'Tables' is selected, and 'dbo.ARTICULO' is highlighted. The main window contains a query editor with the following SQL code:

```

-- 7. Para cada CodArticulo, mostrar TotalSolicitado.
SELECT
    OD.CodArticulo,
    SUM(OD.CantidadSolicitada) AS TotalSolicitado
FROM
    ORDEN_DETALLE OD
GROUP BY
    OD.CodArticulo
ORDER BY
    OD.CodArticulo;

```

Below the code, there are two tabs: 'Results' and 'Messages'. The 'Results' tab displays a table with the following data:

	CodArticulo	TotalSolicitado
85	85	525
86	86	530
87	87	535
88	88	540
89	89	545
90	90	550
91	91	555
92	92	560
93	93	565
94	94	570
95	95	575
96	96	580
97	97	585
98	98	590
99	99	595
100	100	600

Explicación: Agrupa los registros de órdenes por CodArticulo y utiliza SUM() sobre la columna CantidadSolicitada para obtener la cantidad total de cada artículo que ha sido solicitada.

8. Contar órdenes únicas que incluyen cada artículo.

Enunciado: Contar órdenes únicas que incluyen cada artículo.

Código SQL:

-- 8. Contar órdenes únicas que incluyen cada artículo.

```

SELECT
    OD.CodArticulo,
    COUNT(DISTINCT OD.NumOrden) AS TotalOrdenesUnicas
FROM
    ORDEN_DETALLE OD
GROUP BY
    OD.CodArticulo
ORDER BY

```

OD.CodArticulo;

GO

The screenshot shows the SSMS interface with the following details:

- Object Explorer (Left):** Shows the database structure under "ALBERT JEANKARLO CHUQUIYA...". The "Tables" node is expanded, and "dbo.ARTICULO" is selected.
- Query Window (Top Right):** Displays the query:

```
-- 8. Contar órdenes únicas que incluyen cada artículo.
SELECT
    OD.CodArticulo,
    COUNT(DISTINCT OD.NumOrden) AS TotalOrdenesUnicas
FROM
    ORDEN_DETALLE OD
GROUP BY
    OD.CodArticulo
ORDER BY
    OD.CodArticulo;
```
- Results Window (Bottom Right):** Shows the results of the query in a grid format:

CodArticulo	TotalOrdenesUnicas
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1

The screenshot shows the SSMS interface with the following details:

- Connections:** AZURE (connected)
- Database:** QhatuPERU
- Query Window:**
 - Text: "8. Contar órdenes únicas que incluyen cada artículo."
 - SQL Query:

```

83: ... 8. Contar órdenes únicas que incluyen cada artículo.
84: SELECT
85:     ... OD.CodArticulo,
86:     ... COUNT(DISTINCT OD.NumOrden) AS TotalOrdenesUnicas
87: FROM
88:     ... ORDEN_DETALLE OD
89: GROUP_BY
90:     ... OD.CodArticulo
91: ORDER_BY
92:     ... OD.CodArticulo;
93: GO
94:
95: 
```
- Results Grid:**

CodArticulo	TotalOrdenesUnicas
85	1
86	1
87	1
88	1
89	1
90	1
91	1
92	1
93	1
94	1
95	1
96	1
97	1
98	1
99	1
100	1

Explicación: Agrupa por CodArticulo y utiliza COUNT(DISTINCT NumOrden) para contar cuántos números de orden *diferentes* (únicos) compraron ese artículo.

9. Calcular promedio de días por todas las órdenes con FechalIngreso.

Enunciado: Calcular promedio de días por todas las órdenes con FechalIngreso.

Código SQL:

-- 9. Calcular promedio de días por todas las órdenes con FechalIngreso.
SELECT
`CAST(AVG(CAST(DATEDIFF(day, OC.FechaOrden,
OC.FechalIngreso) AS DECIMAL(10,2))) AS DECIMAL(10,2)) AS
PromedioDiasRepcion
FROM`

```

ORDEN_COMPRA OC
WHERE
OC.FechaIngreso IS NOT NULL;
GO

```

The screenshot shows the SSMS interface. On the left is the Object Explorer tree, which is expanded to show databases like ALBERT JEANKARLO CHIQUIRÁ, Azure for Students, master, master (trabajo), QhatuPERU, and others. Under QhatuPERU, the 'Tables' node is selected, revealing tables such as dbo.ARTOCULO, dbo.GUIA_DETALLE, dbo.GUIA_ENVIO, dbo.LINEA, dbo.ORDEN_COMPRA, dbo.ORDEN_DETALLE, dbo.PROVEEDOR, dbo.TIENDA, dbo.TRANSPORTISTA, and others. On the right is the 'SQLQuery_1.sql' query editor window. It contains a T-SQL script with a comment and several lines of code. The results pane below shows a single row with a value of 3.00.

```

-- 9. Calcular promedio de días por todas las órdenes con FechaIngreso.
SELECT
    CAST(AVG(CAST(DATEDIFF(day, OC.FechaOrden, OC.FechaIngreso) AS DECIMAL(10,2))) AS
FROM
    ORDEN_COMPRA OC
WHERE
    OC.FechaIngreso IS NOT NULL;
GO

```

PromedioDiasRepcion
1 3.00

Explicación: DATEDIFF(day, ...) calcula el tiempo en días entre la orden y el ingreso. Luego, AVG() calcula el promedio de ese tiempo de recepción para todas las órdenes que ya tienen una FechaIngreso.

10. Sumar CantidadEnviada por CodTransportista.

Enunciado: Sumar CantidadEnviada por CodTransportista.

Código SQL:

```
-- 10. Sumar CantidadEnviada por CodTransportista.
```

```
SELECT
```

```
    T.CodTransportista,
    T.NomTransportista,
```

```

SUM(GD.CantidadEnviada) AS TotalCantidadEnviada
FROM
    GUIA_ENVIO GE
JOIN
    GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
JOIN
    TRANSPORTISTA T ON GE.CodTransportista =
T.CodTransportista
GROUP BY
    T.CodTransportista, T.NomTransportista
ORDER BY
    TotalCantidadEnviada DESC;
GO

```

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left is the Object Explorer pane, which displays the database structure for 'QhatuPERU'. It includes nodes for 'ALBERT JEANCARLO CHIQUIRAS...', 'Azure for Students', 'SQL databases' (with 'cafesito (jeanl)', 'master (jeanl)', 'master (trabaj)', and 'QhatuPERU (jeanl)'), and 'Tables' (with 'dbo.ARTICULO' selected). Other tables listed include 'dbo.GUIA_DETALLE', 'dbo.GUIA_ENVIO', 'dbo.LINEA', 'dbo.ORDEN_COMPRA', 'dbo.ORDEN_DETALLE', 'dbo.PROVEEDOR', 'dbo.TIENDA', 'dbo.TRANSPORTISTA', and 'dbo.Dropped Ledger Tables'. There are also 'Views', 'Synonyms', 'Programmability', 'External Resources', 'Storage', and 'Security' sections.

The main window contains a query editor with the following SQL code:

```

182 WHERE
183 OC.FechaIngreso IS NOT NULL;
184 GO
185
186
187
188 /* 10. Sumar CantidadEnviada por CodTransportista. */
189 SELECT
190     T.CodTransportista,
191     T.NomTransportista,
192     SUM(GD.CantidadEnviada) AS TotalCantidadEnviada
193 FROM
194     GUIA_ENVIO GE

```

The 'Results' tab is selected, showing the execution results:

	CodTransportista	NomTransportista	TotalCantidadEnviada
1	2100	Rutas Andinas F-100	350
2	2099	Delivery Perú F-99	347
3	2098	Delivery Perú F-98	344
4	2097	Envíos Urgentes F-97	341
5	2096	Carga Pesada F-96	338
6	2095	Envíos Urgentes F-95	335
7	2094	Rutas Andinas F-94	332
8	2093	Carga Pesada F-93	329
9	2092	Carga Pesada F-92	326
10	2091	Rutas Andinas F-91	323
11	2090	Transportes del Sur F-90	320
12	2089	LogiExpress F-89	317
13	2088	Transportes del Sur F-88	314
14	2087	Carga Pesada F-87	311
15	2086	Envíos Urgentes F-86	308

The screenshot shows the SSMS interface. On the left is the Object Explorer pane, which lists connections, databases, and tables. A table named 'dbo.ARTICULO' is selected. The main area is the 'Query Editor' showing a T-SQL script. The script includes a WHERE clause filtering by 'OC.FecharIngreso IS NOT NULL', a GO statement, and a SELECT statement that sums 'CANTIDADENVIADA' grouped by 'CodTransportista'. The results are displayed in a grid.

```

102 WHERE
103 OC.FecharIngreso IS NOT NULL;
104 GO
105
106
107
108 --> 10. Sumar CantidadEnviada por CodTransportista.
109 SELECT
110     T.CodTransportista,
111     T.NomTransportista,
112     SUM(GD.CantidadEnviada) AS TotalCantidadEnviada
113 FROM GUIA_ENVIO GE
114

```

CodTransportista	NomTransportista	TotalCantidadEnviada
2016	Envíos Urgentes F-16	98
2015	Fletes Seguros F-15	95
2014	Carga Pesada F-14	92
2013	Rutas Andinas F-13	89
2012	Courier Veloz F-12	86
2011	Delivery Perú F-11	83
2010	Delivery Perú F-10	80
2009	LogiExpress F-9	77
2008	Delivery Perú F-8	74
2007	Fletes Seguros F-7	71
2006	LogiExpress F-6	68
2005	Rutas Andinas F-5	65
2004	Envíos Urgentes F-4	62
2003	Carga Pesada F-3	59
2002	Transp. Rápido F-2	56
2001	LogiExpress F-1	53

Explicación: Une las tres tablas (GUIA_ENVIO, GUIA_DETALLE, TRANSPORTISTA). Agrupa el resultado por transportista y suma la CantidadEnviada para determinar el volumen total transportado por cada compañía.