



“Año De La Recuperación Y
Consolidación De La Economía Peruana”

UNIVERSIDAD PERUANA LOS ANDES

“FACULTAD DE INGENIERÍA”

ESCUELA PROFESIONAL “SISTEMAS Y
COMPUTACIÓN”

CÁTEDRA: Base de Datos II

CATEDRÁTICO: Ing. Fernandez Bejarano Raul Enrique

ESTUDIANTE: Vega Brañez Samuel Max

CICLO: V

SECCIÓN: B1

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

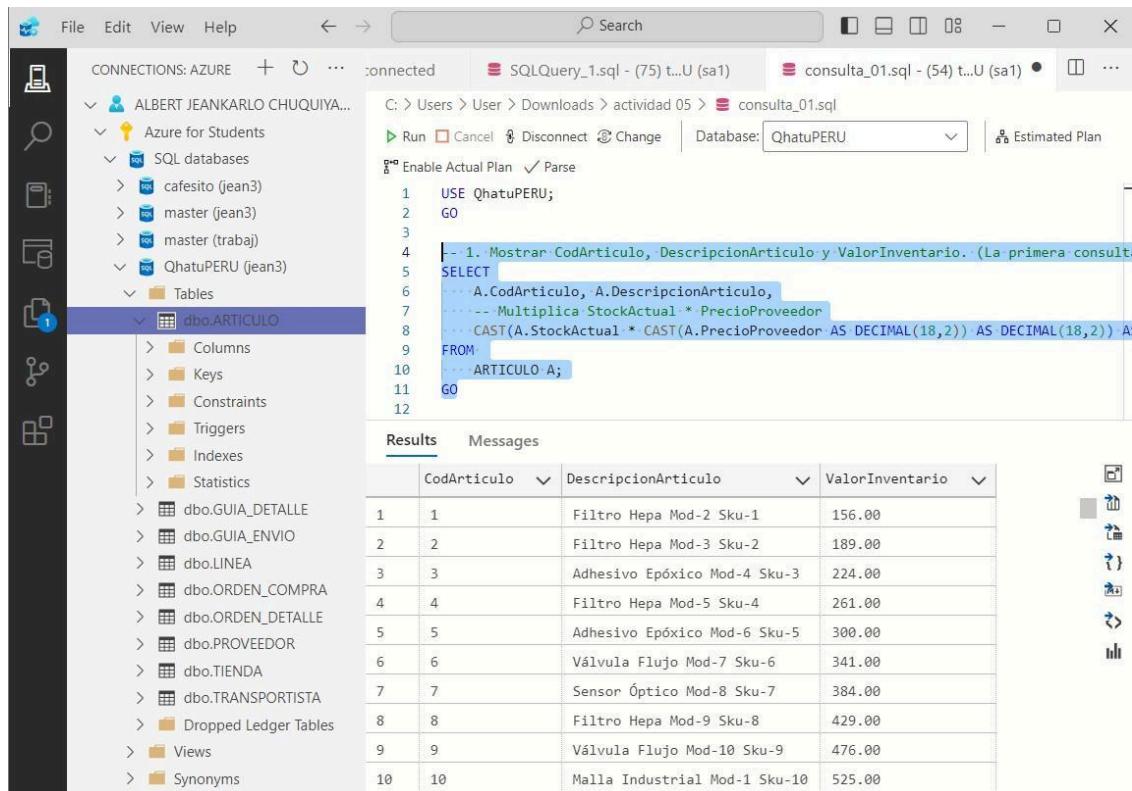
A.CodArticulo, A.DescripcionArticulo,

-- 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 CHUQUIA...". The "dbo.ARTICULO" table is selected.
- Object Explorer (Left):** Shows the table structure for "dbo.ARTICULO" with items like Columns, Keys, Constraints, Triggers, Indexes, Statistics, and Views.
- SQL Editor (Center):** Displays the SQL query with line numbers 1 through 12. Lines 4-11 are highlighted in blue, indicating the main query.
- Results Grid (Bottom):** Shows the output of the query with 10 rows of data.

	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

The screenshot shows the SSMS interface. In the object browser on the left, under 'CONNECTIONS: AZURE', it lists databases: ALBERT JEANKARLO CHUQUIYACU, Azure for Students, and QhatuPERU. Under QhatuPERU, it shows tables like dbo.ARTICULO, dbo.GUIA_DETALLE, etc. The main window displays a T-SQL query in the 'Query Editor' tab:

```

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

```

The results of the query are shown in a table titled 'Results' with columns: CodArticulo, DescripcionArticulo, and ValorInventario. The data is as follows:

CodArticulo	DescripcionArticulo	ValorInventario
191	Interruptor Termico Mod-2 Sku-191	42336.00
192	Malla Industrial Mod-3 Sku-192	42749.00
193	Cable Cat 6 Mod-4 Sku-193	43164.00
194	Válvula Flujo Mod-5 Sku-194	43581.00
195	Rodamiento Z-20 Mod-6 Sku-195	44000.00
196	Válvula Flujo Mod-7 Sku-196	44421.00
197	Rodamiento Z-20 Mod-8 Sku-197	44844.00
198	Rodamiento Z-20 Mod-9 Sku-198	45269.00
199	Aceite Sintético Mod-10 Sku-199	45696.00
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
```

```

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

-- 3. Obtener CodLinea y PrecioProveedor promedio.
SELECT
    ...
    CAST(Avg(A.PrecioProveedor) AS DECIMAL(18,2)) AS
    PrecioProveedorPromedio
FROM
    ...
    ARTICULO A
GROUP BY
    ...
    A.CodLinea
;
GO

```

CodArticulo	DescripcionArticulo	ValorInventario
191	Interruptor Termico Mod-2 Sku-191	42336.00
192	Malla Industrial Mod-3 Sku-192	42749.00
193	Cable Cat 6 Mod-4 Sku-193	43164.00
194	Válvula Flujo Mod-5 Sku-194	43581.00
195	Rodamiento Z-20 Mod-6 Sku-195	44000.00
196	Válvula Flujo Mod-7 Sku-196	44421.00
197	Rodamiento Z-20 Mod-8 Sku-197	44844.00
198	Rodamiento Z-20 Mod-9 Sku-198	45269.00
199	Aceite Sintético Mod-10 Sku-199	45696.00
200	Sensor Óptico Mod-1 Sku-200	46125.00
	TotalInventarioMonetario	
1	3314700.00	

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

```

23 -- 3. Obtener CodLinea y PrecioProveedor promedio.
24
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

```

	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


```

23 -- 3. Obtener CodLinea y PrecioProveedor promedio.
24
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

```

	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 productos que pertenecen a esa misma línea.

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 Microsoft SQL Server Management Studio (SSMS) interface. On the left, the Object Explorer tree view shows the database structure, including the 'ARTICULO' table under the 'dbo' schema. On the right, the 'SQLQuery_1.sql' tab is active, displaying the following SQL script:

```
-- 4. Contar artículos descontinuados.
SELECT
    COUNT(CodArticulo) AS TotalArticulosDescontinuados
FROM
    ARTICULO
WHERE
    Descontinuado = 1;
GO
```

The 'Results' tab shows the output of the query:

TotalArticulosDescontinuados
1
5

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

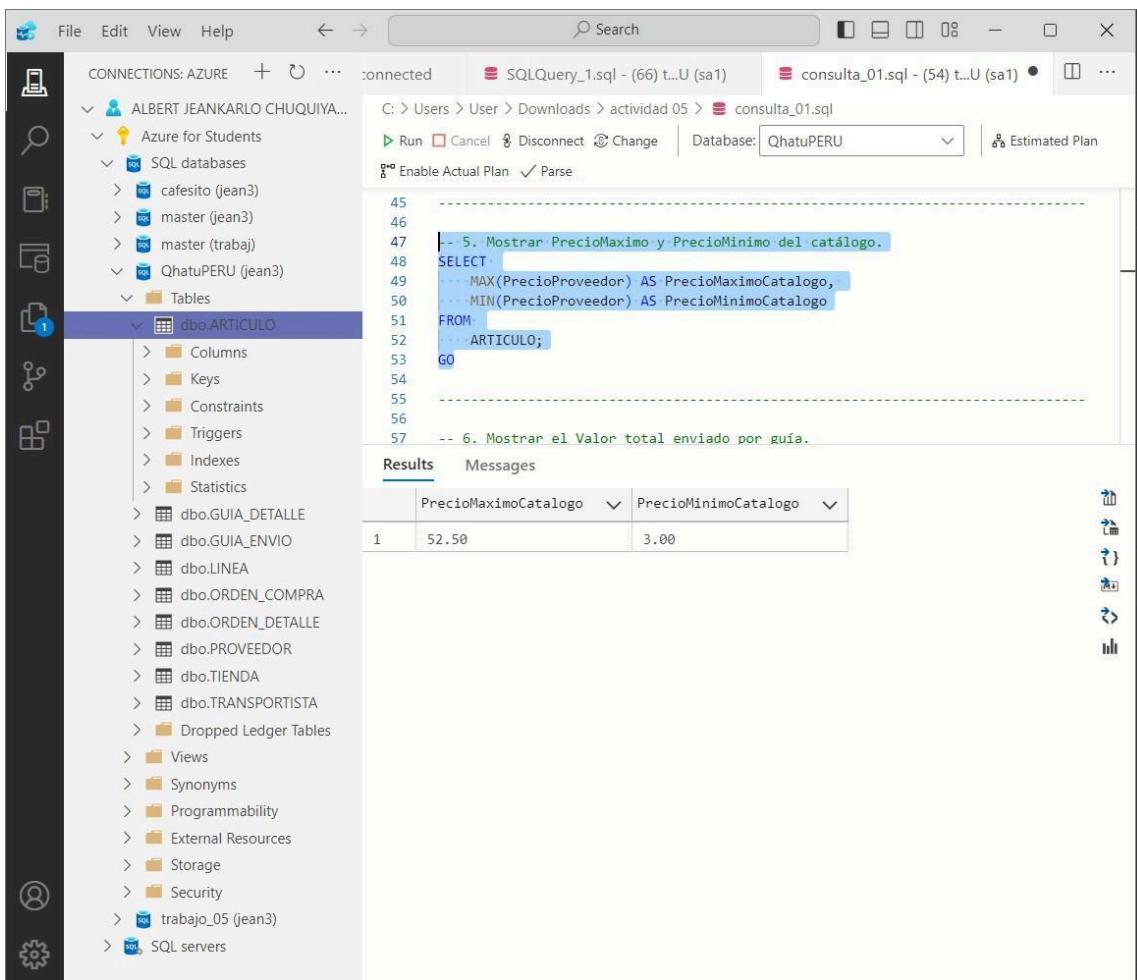
MAX(PrecioProveedor) **AS** PrecioMaximoCatalogo,

MIN(PrecioProveedor) **AS** PrecioMinimoCatalogo

FROM

ARTICULO;

GO



The screenshot shows the SSMS interface with the following details:

- File Explorer:** Shows the database structure under "ALBERT JEANKARLO CHUQUIYAPU". The "Tables" node is expanded, showing "dbo.ARTICULO" and its columns.
- SQL Query Editor:** Contains the following T-SQL code:

```
-- 5. Mostrar PrecioMaximo y PrecioMinimo del catálogo.  
SELECT  
    MAX(PrecioProveedor) AS PrecioMaximoCatalogo,  
    MIN(PrecioProveedor) AS PrecioMinimoCatalogo  
FROM  
    ARTICULO;  
GO
```
- Results Grid:** Displays the output of the query:

	PrecioMaximoCatalogo	PrecioMinimoCatalogo
1	52.50	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...'. It includes 'Azure for Students', 'SQL databases', and 'Tables'. Under 'Tables', 'dbo.ARTICULO' is selected, and its sub-items like 'Columns', 'Keys', etc., are visible.
- Top bar:** File, Edit, View, Help, Search, Database: QhatuPERU, Estimated Plan.
- Main pane (Query Editor):** Displays the T-SQL code for query 6. The code is:

```
54  
55  
56  
57 -- 6. Mostrar el Valor total enviado por guía.  
58 SELECT  
59     GD.NumGuia,  
60     CAST(SUM(GD.CantidadEnviada * GD.PrecioVenta) AS MONEY) AS ValorTotalEnviado  
61 FROM  
62     GUIA_DETALLE GD  
63 GROUP BY  
64     GD.NumGuia;  
65 GO  
66
```
- Results pane:** Shows the execution results in a table format. The table has two columns: 'NumGuia' and 'ValorTotalEnviado'. The data is as follows:

	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	1391.60

The screenshot shows the SSMS interface. In the top right, there are two tabs: 'SQLQuery_1.sql - (66) t...U (sa1)' and 'consulta_01.sql - (54) t...U (sa1)'. The database dropdown shows 'QhatuPERU'. The main pane displays a T-SQL query:

```

54
55
56
57 -- 6. Mostrar el Valor total enviado por guía.
58 SELECT
59     ...
60     ...
61 FROM ...
62     ...
63 GROUP BY ...
64     ...
65 GO
66

```

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

	NumGuia	ValorTotalEnviado
85	4085	19062.50
86	4086	19465.60
87	4087	19872.90
88	4088	20284.40
89	4089	20700.10
90	4090	21120.00
91	4091	21544.10
92	4092	21972.40
93	4093	22404.90
94	4094	22841.60
95	4095	23282.50
96	4096	23727.60
97	4097	24176.90
98	4098	24630.40
99	4099	25088.10
100	4100	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) Database: QhatuPERU Estimated Plan

ALBERT JEANKARLO CHUQUIYA...
Azure for Students
SQL databases
cafesito (jean3)
master (jean3)
master (trabaj)
QhatuPERU (jean3)
Tables
dbo.ARTICULO
Columns
Keys
Constraints
Triggers
Indexes
Statistics
dbo.GUIA_DETALLE
dbo.GUIA_ENVIO
dbo.LINEA
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)
SQL servers

7. Para cada CodArticulo, mostrar TotalSolicitado.

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

Spaces: 4 UTE-8 CR LF SQL 100 rows Choose SQL Language 00:00:00 tcnjean3.database.windows.net : QhatuPERU

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left, there's a sidebar with various icons for file operations, search, and database management. The main area has tabs for 'SQLQuery_1.sql' and 'consulta_01.sql'. The database dropdown shows 'QhatuPERU'. The results pane shows a query for calculating total requested items per article:

```

-- 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

```

The results table shows 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

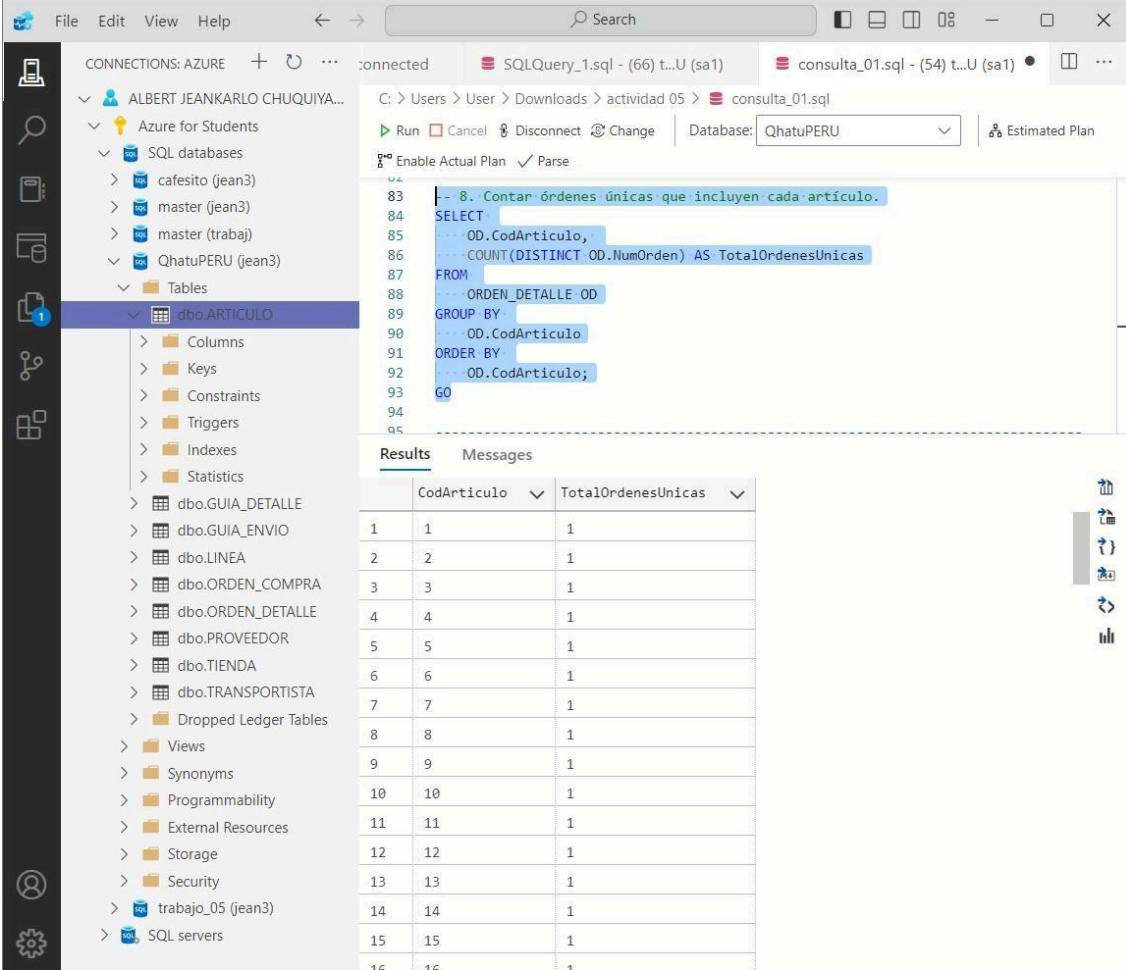
 ORDEN_DETALLE OD

GROUP BY

 OD.CodArticulo

ORDER BY

OD.CodArticulo;
GO



The screenshot shows the SSMS interface with the following details:

- File Bar:** File, Edit, View, Help.
- Toolbar:** Standard toolbar with icons for New Query, Save, Print, etc.
- Connections:** AZURE (connected), ALBERT JEANKARLO CHUQUIYA...
- Object Explorer:** Shows the database structure under QhatuPERU (jean3). The **dbo.ARTICULO** table is selected.
- Query Editor:** Contains a T-SQL script:

```
-- 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
```
- Results Grid:** Displays the results of the query:

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

The screenshot shows the SSMS interface. On the left is the Object Explorer pane, which lists connections (AZURE), databases (Azure for Students, master, trabaj), and tables (dbo.ARTICULO, dbo.GUIA_DETALLE, etc.). The main area shows a query window with the following SQL code:

```

-- 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 results pane displays a table with two columns: CodArticulo and TotalOrdenesUnicas, containing 100 rows where all values are 1.

	CodArticulo	TotalOrdenesUnicas
85	85	1
86	86	1
87	87	1
88	88	1
89	89	1
90	90	1
91	91	1
92	92	1
93	93	1
94	94	1
95	95	1
96	96	1
97	97	1
98	98	1
99	99	1
100	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 includes connections to 'AZURE' and 'ALBERT JEANKARLO CHUQUIYA...', and databases like 'cafesito', 'master', 'trabaj', and 'QhatuPERU'. Under 'QhatuPERU', the 'dbo' schema is expanded, showing tables such as 'ARTICULO', 'GUIA_DETALLE', 'GUIA_ENVIO', 'LINEA', 'ORDEN_COMPRA', 'ORDEN_DETALLE', 'PROVEEDOR', 'TIENDA', 'TRANSPORTISTA', and others. The right side of the screen contains a query editor window with the following SQL code:

```

-- 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

```

The results pane shows a single row with the value '3.00' under the column 'PromedioDiasRepcion'.

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

```

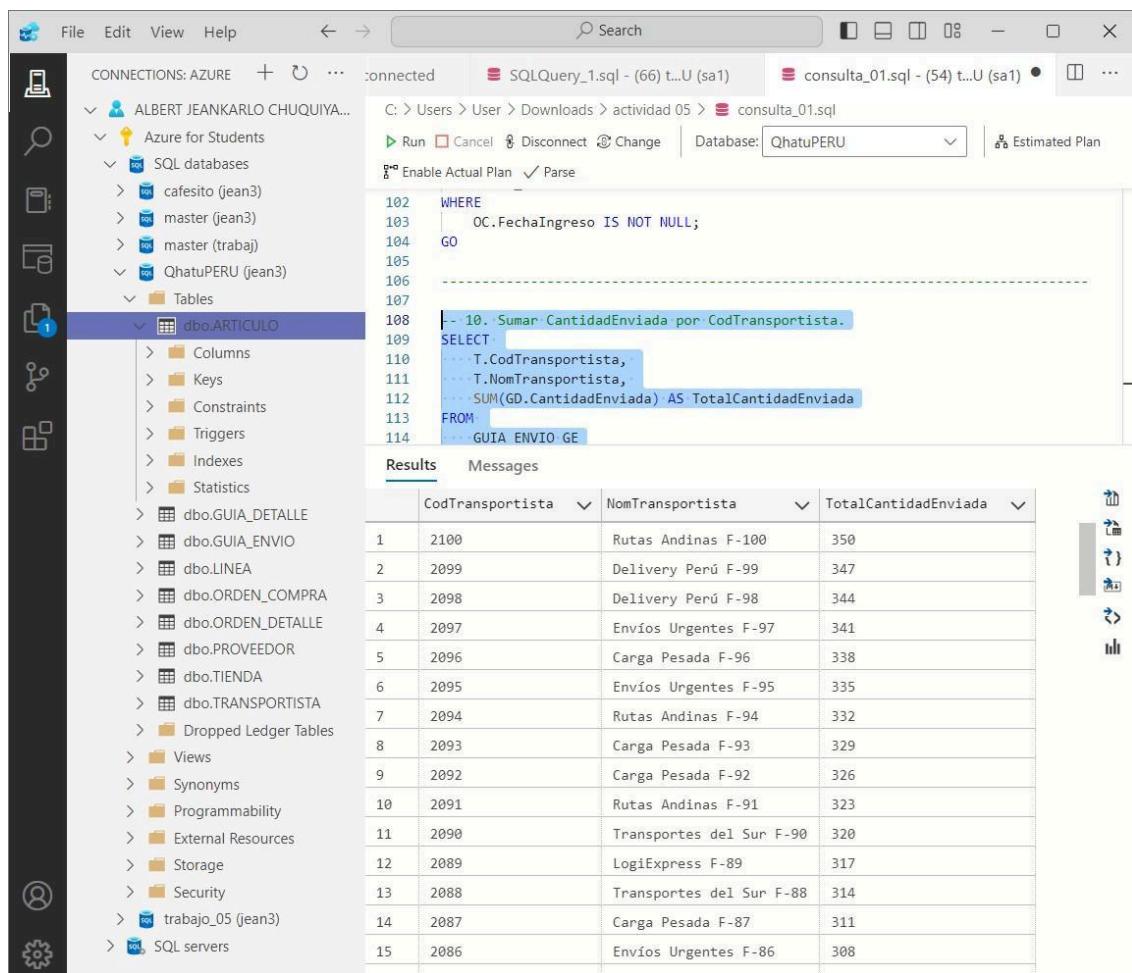
T.CodTransportista,
T.NomTransportista,

```

```

        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 SSMS interface with the following details:

- File Bar:** File, Edit, View, Help.
- Toolbar:** Standard toolbar icons.
- Connections:** AZURE, connected.
- Query Editor:**
 - Script: SQLQuery_1.sql - (66) t...U (sa1)
 - Results: consulta_01.sql - (54) t...U (sa1)
 - Database: QhatuPERU
 - Buttons: Run, Cancel, Disconnect, Change, Estimated Plan.
 - Text Area:


```

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

```
- Object Explorer:** Shows the database structure under ALBERT JEANKARLO CHUQUIAYA...:
 - Azure for Students
 - SQL databases:
 - cafesito (jean3)
 - master (jean3)
 - master (trabaj)
 - QhatuPERU (jean3):
 - Tables (selected): db.ARTICULO
 - Columns
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
 - dbo.GUIA_DETALLE
 - dbo.GUIA_ENVIO
 - dbo.LINEA
 - 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)
 - SQL servers

	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 Microsoft SQL Server Management Studio (SSMS) interface. On the left is the Object Explorer pane, which lists connections, databases, and tables. A specific table named 'dbo.ARTICULO' is selected. The main area contains a query window with the following SQL code:

```

102 WHERE
103 OC.FechaIngreso 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
114 GUIA_ENVIO GE

```

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

	CodTransportista	NomTransportista	TotalCantidadEnviada
85	2016	Envíos Urgentes F-16	98
86	2015	Fletes Seguros F-15	95
87	2014	Carga Pesada F-14	92
88	2013	Rutas Andinas F-13	89
89	2012	Courier Veloz F-12	86
90	2011	Delivery Perú F-11	83
91	2010	Delivery Perú F-10	80
92	2009	LogiExpress F-9	77
93	2008	Delivery Perú F-8	74
94	2007	Fletes Seguros F-7	71
95	2006	LogiExpress F-6	68
96	2005	Rutas Andinas F-5	65
97	2004	Envíos Urgentes F-4	62
98	2003	Carga Pesada F-3	59
99	2002	Transp. Rápido F-2	56
100	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.