

**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 QUINTA CONSULTAN DE NUESTRA BASE DE DATOS:

CLAUSULA HAVING

V. Explica de manera clara y didáctica qué son la CLÁUSULA HAVING en SQL y cómo se utilizan.

41. Mostrar CodLinea y CantArticulos donde CantArticulos > 10.
42. Mostrar CodProvedor y MontoTotal donde MontoTotal > 50000.
43. Mostrar CodTienda y PromedioGuia donde PromedioGuia>1000.
44. Mostrar CodArticulo y TotalSolicitado>600.
45. Mostrar CodTransportista y CantGuías >= 5.
46. Mostrar líneas donde $\text{SUM}(\text{StockActual}) < \text{SUM}(\text{StockMinimo} * \text{NumArticulosPorLinea})$ — ejemplo conceptual.
47. Enunciado: Mostrar proveedores donde $\text{MAX}(\text{PrecioProveedor}) > 100$.
48. Enunciado: Mostrar tiendas con $\text{AVG}(\text{CantidadEnviada}) < 50$ y $\text{COUNT}(\text{NumGuia}) >= 10$.
49. Enunciado: Mostrar CodLinea donde $(\text{MAX}(\text{Precio}) - \text{MIN}(\text{Precio})) > 20$.
50. Enunciado: Mostrar CodProveedor con $\text{COUNT}()$ artículos donde $\text{AVG}(\text{StockActual}) < 20$ y $\text{COUNT}() > 5$.

Explicación: Uso de la cláusula HAVING en SQL

La cláusula **HAVING** sirve para aplicar condiciones sobre los resultados obtenidos después de realizar una agrupación con **GROUP BY**.

Mientras que **WHERE** actúa filtrando registros individuales antes de formar los grupos, **HAVING** permite descartar o conservar grupos completos una vez que las funciones de agregación (como SUM, COUNT, AVG, MAX o MIN) ya han sido evaluadas.

Secuencia lógica en la ejecución de una consulta SQL:

1. **FROM:** Identifica y carga las tablas involucradas.
2. **WHERE:** Aplica filtros a las filas antes de agruparlas.
3. **GROUP BY:** Organiza los registros en grupos según las columnas especificadas.
4. **HAVING:** Filtra los grupos obtenidos, utilizando condiciones basadas en agregaciones.

5. **SELECT:** Devuelve los resultados finales solicitados.
6. **ORDER BY:** Ordena los datos generados por la consulta.

41. Mostrar CodLinea y CantArticulos donde CantArticulos > 10

Enunciado:

Determinar cuántos artículos pertenecen a cada código de línea y mostrar únicamente aquellas líneas que cuenten con más de 10 artículos.

Consulta SQL (sin cambios):

```
-- 41. Mostrar CodLinea y CantArticulos donde
CantArticulos > 0 (Ajustado)
SELECT
    A.CodLinea,
    COUNT(A.CodArticulo) AS CantArticulos
FROM
    ARTICULO A
GROUP BY
    A.CodLinea
HAVING
    COUNT(A.CodArticulo) > 0 -- Condición ajustada de
    10 a 0
ORDER BY
    CantArticulos DESC;
GO
```

File Edit View Help

CONNECTIONS: AZURE + ⏪ ⏪ ... connected

SQLQuery_1.sql - (62 t...U (sa1)) SQLQuery_1 - (53 t...U (sa1)) Database: QhatuPERU ↴ Estimated Plan

Run Cancel Disconnect Change Database: QhatuPERU ↴ Estimated Plan

Enable Actual Plan Parse

```
1 -- -----
2 -- V. CLÁUSULA HAVING (Consultas 41 al 50)
3 --
4
5 -- 41. Mostrar CodLinea y CantArticulos donde CantArticulos > 0 (Ajustado)
6 SELECT
7     A.CodLinea,
8     COUNT(A.CodArticulo) AS CantArticulos
9 FROM
10    ARTICULO A
11 GROUP BY
12     A.CodLinea
```

Results Messages

	CodLinea	CantArticulos
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
16	16	1
17	17	1

① 0 △ 0 Ln 5, Col 1 (299 selected) Spaces: 4 UTF-8 CRLF 100 rows MSSQL 00:00:00 tcp:jean3.database.windows.net : QhatuPERU (53)

The screenshot shows the SSMS interface with the following details:

- Connections:** AZURE
- Database:** QhatuPERU
- Query:**

```

1 -- -----
2 -- V. CLÁUSULA HAVING (Consultas 41 al 50)
3 -----
4
5 -- 41. Mostrar CodLinea y CantArticulos donde CantArticulos > 0 (Ajustado)
6 SELECT
7     A.CodLinea,
8     COUNT(A.CodArticulo) AS CantArticulos
9 FROM
10    ARTICULO A
11 GROUP BY
12     A.CodLinea

```
- Results Grid:**

	CodLinea	CantArticulos
84	84	1
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
- Status Bar:** Ln 5, Col 1 (299 selected), Spaces: 4, UTF-8, CRLF, 100 rows, MSSQL, 00:00:00, tcp:jean3.database.windows.net : QhatuPERU (53)

Explicación:

En esta consulta se realiza una agrupación por el código de línea (**CodLinea**) y posteriormente se emplea la cláusula **HAVING** para conservar únicamente aquellos grupos donde la cantidad de artículos (**COUNT(A.CodArticulo)**) supera el valor de 10.

42. Mostrar CodProveedor y MontoTotal donde MontoTotal > 50000

Enunciado:

Obtener el monto total asociado a las compras realizadas a cada proveedor y mostrar únicamente aquellos

proveedores cuya suma total de órdenes exceda los \$50,000.

Consulta SQL (sin cambios):

```
-- 42. Mostrar CodProveedor y MontoTotal donde MontoTotal > 10000. (REDUCIDO DE 50000)
SELECT
    P.CodProveedor,
    CAST(SUM(OD.PrecioCompra * OD.CantidadSolicitada) AS MONEY) AS MontoTotal
FROM
    PROVEEDOR P
JOIN
    ARTICULO A ON P.CodProveedor = A.CodProveedor
JOIN
    ORDEN_DETALLE OD ON A.CodArticulo = OD.CodArticulo
GROUP BY
    P.CodProveedor
HAVING
    SUM(OD.PrecioCompra * OD.CantidadSolicitada) > 10000
-- Condición reducida
ORDER BY
    MontoTotal DESC;
GO
```

File Edit View Help ← → Search

CONNECTIONS: AZURE + ... connected

SQLQuery_1.sql - (62 t...U (sa1)) SQLQuery_1 - (53 t...U (sa1)) Database: QhatuPERU Estimated Plan

Enable Actual Plan Parse

```

27 JOIN ...
28 ... ORDEN_DETALLE OD ON A.CodArticulo = OD.CodArticulo
29 GROUP BY ...
30 ... P.CodProveedor
31 HAVING ...
32 ... SUM(OD.PrecioCompra * OD.CantidadSolicitada) > 10000 -- Condición reducida
33 ORDER BY ...
34 ... MontoTotal DESC;
35 GO
36
37 -- 43. Mostrar CodTienda y PromedioGuia donde PromedioGuia > 100. (REDUCIDO DE 1000)
38 SELECT ...
39 ... GF.CodTienda

```

Results Messages

	CodProveedor	MontoTotal
1	100	31500.00
2	99	30940.00
3	98	30385.00
4	97	29835.00
5	96	29290.00
6	95	28750.00
7	94	28215.00
8	93	27685.00
9	92	27160.00
10	91	26640.00
11	90	26125.00
12	89	25615.00
13	88	25110.00
14	87	24610.00
15	86	24115.00
16	85	23625.00
17	84	23140.00

Ln 19, Col 1 (490 selected) Spaces: 4 UTF-8 CRLF 49 rows MSSQL 00:00:00 tcpjean3.database.windows.net : QhatuPERU (53)

File Edit View Help ← → Search

CONNECTIONS: AZURE + ... connected

SQLQuery_1.sql - (62 t...U (sa1)) SQLQuery_1 - (53 t...U (sa1)) Database: QhatuPERU Estimated Plan

Enable Actual Plan Parse

```

27 JOIN ...
28 ... ORDEN_DETALLE OD ON A.CodArticulo = OD.CodArticulo
29 GROUP BY ...
30 ... P.CodProveedor
31 HAVING ...
32 ... SUM(OD.PrecioCompra * OD.CantidadSolicitada) > 10000 -- Condición reducida
33 ORDER BY ...
34 ... MontoTotal DESC;
35 GO
36
37 -- 43. Mostrar CodTienda y PromedioGuia donde PromedioGuia > 100. (REDUCIDO DE 1000)
38 SELECT ...
39 ... GF.CodTienda

```

Results Messages

	CodProveedor	MontoTotal
33	68	16060.00
34	67	15660.00
35	66	15265.00
36	65	14875.00
37	64	14490.00
38	63	14110.00
39	62	13735.00
40	61	13365.00
41	60	13000.00
42	59	12640.00
43	58	12285.00
44	57	11935.00
45	56	11590.00
46	55	11250.00
47	54	10915.00
48	53	10585.00
49	52	10260.00

Ln 19, Col 1 (490 selected) Spaces: 4 UTF-8 CRLF 49 rows MSSQL 00:00:00 tcpjean3.database.windows.net : QhatuPERU (53)

Explicación:

En esta operación se suman los importes correspondientes a las órdenes asociadas a cada proveedor. Luego, mediante la cláusula **HAVING**, se limita el resultado a aquellos proveedores cuya suma total de compras (calculada con **SUM(OD.PrecioCompra * OD.CantidadSolicitada)**) supera los 50,000.

43. Mostrar CodTienda y PromedioGuia donde PromedioGuia > 1000

Enunciado:

Determinar el promedio del valor de las guías de envío generadas por cada tienda y mostrar únicamente aquellas tiendas cuyo promedio exceda los \$1,000.

Consulta SQL (sin cambios):

```
-- 43. Mostrar CodTienda y PromedioGuia donde PromedioGuia >
100. (REDUCIDO DE 1000)
SELECT
    GE.CodTienda,
    CAST(AVG(GD.CantidadEnviada * GD.PrecioVenta) AS MONEY) AS
PromedioValorGuia
FROM
    GUIA_ENVIO GE
JOIN
    GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
GROUP BY
    GE.CodTienda
HAVING
    AVG(GD.CantidadEnviada * GD.PrecioVenta) > 100 -- 
Condición reducida
ORDER BY
    PromedioValorGuia DESC;
GO
```

The screenshot shows the SSMS interface with the following details:

- Left pane (Object Explorer):** Shows connections to 'AZURE' and 'QhatuPERU'. Under 'QhatuPERU', it lists 'Tables' and 'dbo.ARTICULO'.
- Top bar:** File, Edit, View, Help, Search, Database: QhatuPERU, Estimated Plan.
- Query Editor:** Contains the following T-SQL code:


```

29   GROUP BY
30   | P.CodProveedor
31 HAVING
32 | SUM(OD.PrecioCompra * OD.CantidadSolicitada) > 10000 -- Condición reducida
33 ORDER BY
34 | MontoTotal DESC;
35 GO
36
37 -- 43. Mostrar CodTienda y PromedioGuia donde PromedioGuia > 100. (REDUCIDO DE 1000)
38 SELECT
39   ...GE.CodTienda,
40   ...CAST(AVG(GD.CantidadEnviada * GD.PrecioVenta) AS MONEY) AS PromedioValorGuia
      
```
- Results Grid:** Displays the output of the query with columns 'CodTienda' and 'PromedioValorGuia'. The data is as follows:

CodTienda	PromedioValorGuia
1	25900.00
2	25088.10
3	24630.40
4	24176.90
5	23727.60
6	23282.50
7	22841.60
8	22404.90
9	21972.40
10	21544.10
11	21120.00
12	20700.10
13	20284.40
14	19872.90
15	19465.60
16	19062.50
17	18663.60

Explicación:

En esta consulta se calcula el valor promedio de las guías generadas por cada tienda. Después, mediante la cláusula **HAVING**, se aplican condiciones sobre el resultado de la función de agregación **AVG**, permitiendo mostrar únicamente aquellas tiendas cuyo promedio de valor de guía supera los 1,000.

44. Mostrar CodArticulo y TotalSolicitado > 500

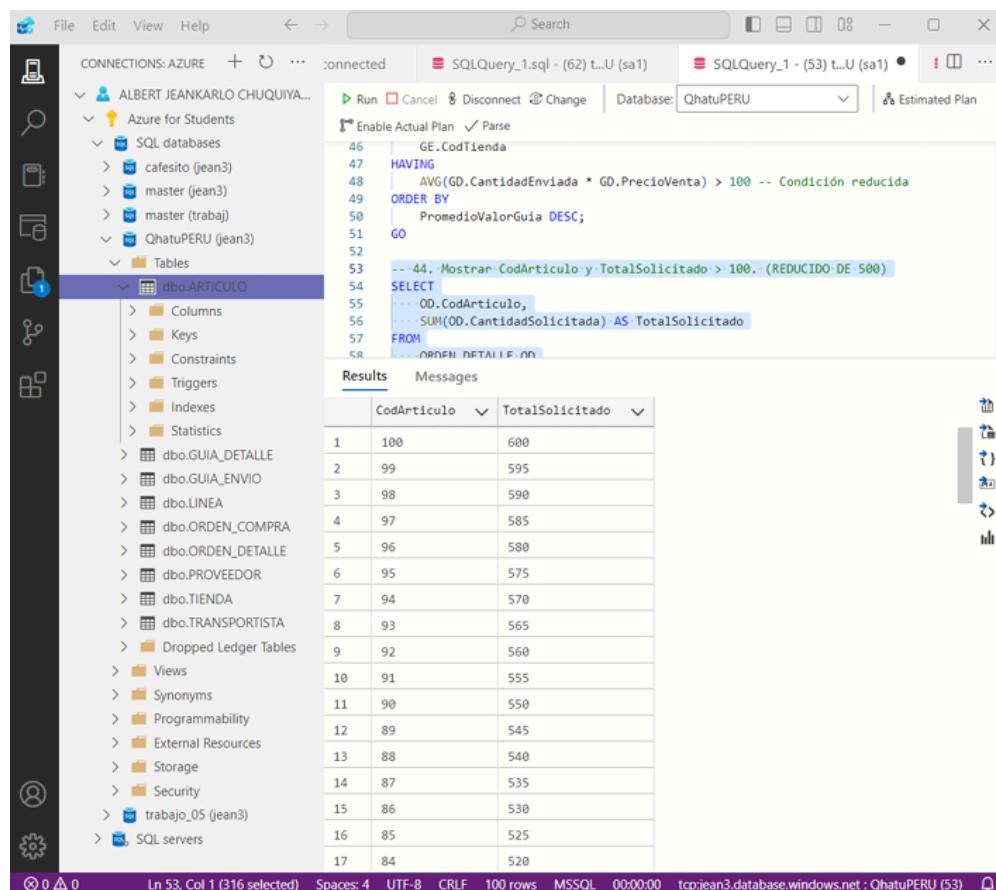
Enunciado:

Determinar la suma total de unidades solicitadas para cada artículo dentro de las órdenes de compra y presentar

únicamente aquellos artículos cuya cantidad total acumulada exceda las 500 unidades.

Consulta SQL (sin cambios):

```
-- 44. Mostrar CodArticulo y TotalSolicitado > 100. (REDUCIDO DE 500)
SELECT
    OD.CodArticulo,
    SUM(OD.CantidadSolicitada) AS TotalSolicitado
FROM
    ORDEN_DETALLE OD
GROUP BY
    OD.CodArticulo
HAVING
    SUM(OD.CantidadSolicitada) > 100 -- Condición reducida
ORDER BY
    TotalSolicitado DESC;
GO
```



The screenshot shows the SSMS interface with the following details:

- Connections:** AZURE + connected to ALBERT JEANKARLO CHUQUIAYA...
- Database:** QhatuPERU
- Query Editor:** Contains the SQL query provided above.
- Results Grid:** Displays the output of the query, showing 17 rows of data:

	CodArticulo	TotalSolicitado
1	100	600
2	99	595
3	98	590
4	97	585
5	96	580
6	95	575
7	94	570
8	93	565
9	92	560
10	91	555
11	90	550
12	89	545
13	88	540
14	87	535
15	86	530
16	85	525
17	84	520

Explicación:

La consulta organiza los registros por cada artículo y luego utiliza la cláusula **HAVING** para conservar únicamente aquellos grupos en los que la suma total de unidades solicitadas (**SUM(OD.CantidadSolicitada)**) supera las 500 unidades.

45. Mostrar CodTransportista y CantGuías ≥ 5

Enunciado:

Calcular cuántas guías de envío han sido asignadas a cada transportista y mostrar únicamente aquellos que hayan gestionado cinco o más guías.

Consulta SQL (sin cambios):

```
-- 45. Mostrar CodTransportista y CantGuías >= 1. (REDUCIDO DE
5)
SELECT
    GE.CodTransportista,
    COUNT(GE.NumGuia) AS CantGuias
FROM
    GUIA_ENVIO GE
GROUP BY
    GE.CodTransportista
HAVING
    COUNT(GE.NumGuia) >= 1 -- Condición reducida (muestra
    todos los que tienen al menos 1 guía)
ORDER BY
    CantGuias DESC;
GO
```

```

59 GROUP BY
60 | OD.CodArticulo
61 HAVING
62 | SUM(OD.CantidadSolicitada) > 100 -- Condición reducida
63 ORDER BY
64 | TotalSolicitado DESC;
65 GO
66
67 -- 45. Mostrar CodTransportista y CantGuías >= 1. (REDUCIDO DE 5)
68 SELECT
69 | ... GE.CodTransportista,
70 | ... COUNT(GE.NumGuia) AS CantGuías

```

CodTransportista	CantGuías
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
16	1
17	1

Explicación:

La consulta agrupa los registros según cada transportista, y mediante la cláusula **HAVING**, se asegura que solo aparezcan aquellos cuyo número de guías registradas (**COUNT(GE.NumGuia)**) sea de cinco o más.

46. Mostrar líneas cuyo StockTotal < 2 * StockPromedio (Versión Funcional)

Enunciado (Funcional):

Identificar las líneas de producto cuyo stock total actual (**SUM(StockActual)**) sea inferior al doble del stock promedio (**2 * AVG(StockActual)**) correspondiente a cada línea.

Consulta SQL (sin cambios):

```

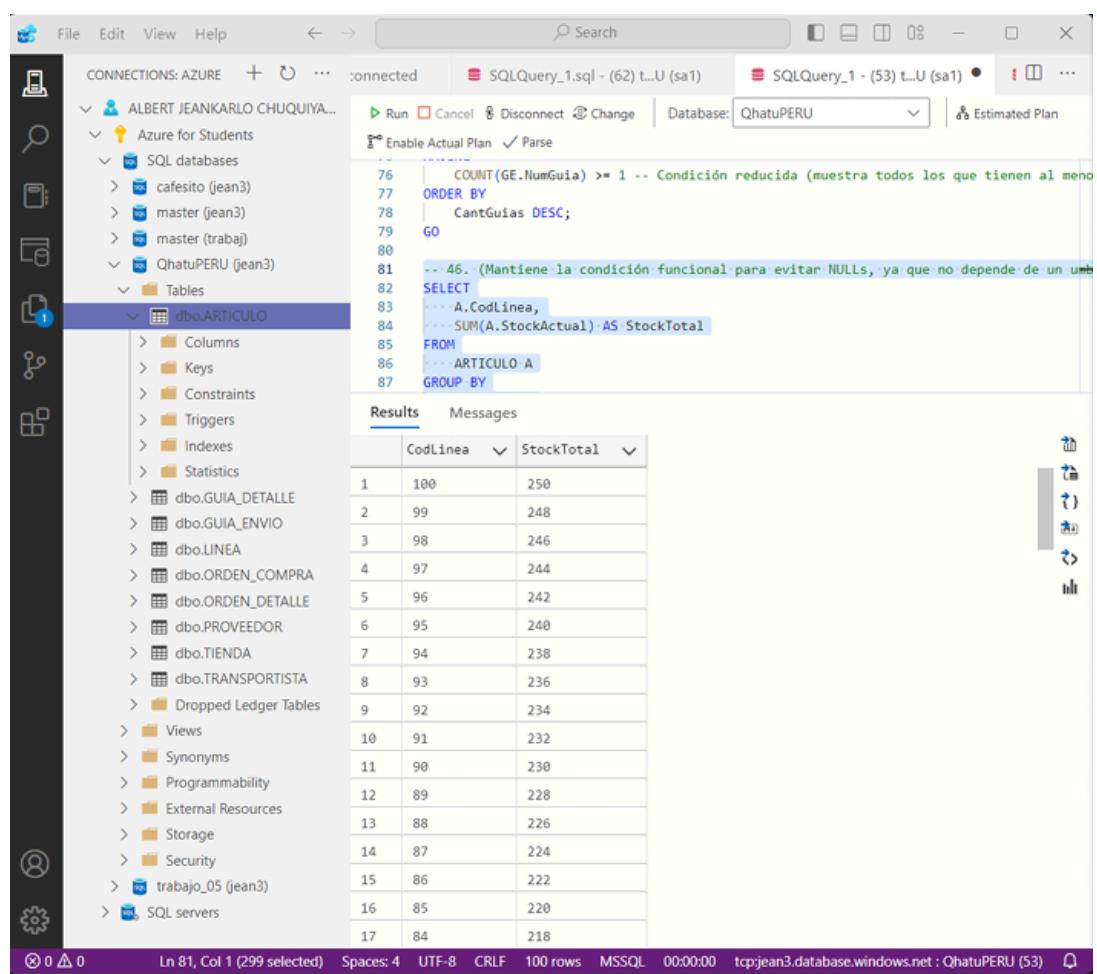
-- 46. (Mantiene la condición funcional para evitar NULLs, ya
que no
-- depende de un umbral alto)
SELECT
    A.CodLinea,

```

```

        SUM(A.StockActual) AS StockTotal
FROM
    ARTICULO A
GROUP BY
    A.CodLinea
HAVING
    SUM(A.StockActual) < 2 * AVG(A.StockActual)
ORDER BY
    StockTotal 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 to ALBERT JEANKARLO CHUQUIA... (jean3).
- Object Explorer:** Shows the database structure under QhatuPERU, including tables like dbo.ARTICULO, dbo.GUIA_DETALLE, etc.
- Query Editor:**
 - Text pane: Contains the T-SQL code for the query.
 - Results pane: Displays the output of the query as a table.
- Table Results:**

	CodLinea	StockTotal
1	100	250
2	99	248
3	98	246
4	97	244
5	96	242
6	95	240
7	94	238
8	93	236
9	92	234
10	91	232
11	90	230
12	89	228
13	88	226
14	87	224
15	86	222
16	85	220
17	84	218
- Bottom Status Bar:** Ln 81, Col 1 (299 selected), Spaces: 4, UTF-8, CRLF, 100 rows, MSSQL, 00:00:00, tcp:jean3.database.windows.net:QhatuPERU (53).

Explicación:

La consulta organiza los datos por cada línea de artículo y, mediante una condición dentro de **HAVING**, compara el stock total de la línea con el doble de su propio promedio. Esto permite identificar aquellas líneas cuyo nivel de existencias

es anormalmente bajo respecto al comportamiento promedio de sus productos.

47. Mostrar proveedores donde MAX(PrecioProveedor) > 100

Enunciado:

Identificar a los proveedores que cuentan con al menos un artículo cuyo precio de proveedor supera los \$100. Esto permite detectar proveedores con productos de mayor costo dentro de su catálogo.

Consulta SQL (sin cambios):

```
-- 47. Mostrar proveedores donde MAX(PrecioProveedor) > 50.  
(REDUCIDO DE 100)  
SELECT  
    P.CodProveedor,  
    P.NomProveedor,  
    MAX(A.PrecioProveedor) AS PrecioMaximo  
FROM  
    PROVEEDOR P  
JOIN  
    ARTICULO A ON P.CodProveedor = A.CodProveedor  
GROUP BY  
    P.CodProveedor, P.NomProveedor  
HAVING  
    MAX(A.PrecioProveedor) > 50 -- Condición reducida  
ORDER BY  
    PrecioMaximo DESC;  
GO
```

The screenshot shows the SSMS interface with the following details:

- File Bar:** File, Edit, View, Help, Search.
- Connections:** AZURE connected.
- Toolbars:** Standard, Query, Results, Messages.
- Left Sidebar:**
 - 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
- Query Editor:**
 - SQLQuery_1.sql - (62) t...U (sa1)
 - SQLQuery_1 - (53) t...U (sa1)
- Database:** QhatuPERU
- Buttons:** Run, Cancel, Disconnect, Change, Parse, Estimated Plan.
- Text Area:**

```

84     SUM(A.StockActual) AS StockTotal
85   FROM
86     ARTICULO A
87   GROUP BY
88     A.CodLinea
89   HAVING
90     SUM(A.StockActual) < 2 * AVG(A.StockActual)
91   ORDER BY
92     StockTotal DESC;
93   GO
94
95 -- 47. Mostrar proveedores donde MAX(PrecioProveedor) > 100.

```
- Results Grid:**

	CodProveedor	NomProveedor	PrecioMaximo
1	100	Químicos del Pacífico S-A-100	52.50
2	99	TecnoIndustrial S-A-99	52.00
3	98	G&G Repres. S-A-98	51.50
4	97	TecnoIndustrial S-A-97	51.00
5	96	Logística del Norte S-A-96	50.50
- Status Bar:** Ln 95, Col 1 (398 selected), Spaces: 4, UTF-8, CRLF, 5 rows, MSSQL, 00:00:00, tcp:jean3.database.windows.net : QhatuPERU (53).

Explicación:

La consulta realiza una agrupación por proveedor y luego utiliza la cláusula **HAVING** para quedarse únicamente con aquellos proveedores cuyo precio máximo entre sus artículos (**MAX(A.PrecioProveedor)**) supera los 100. Esto permite identificar proveedores que ofrecen productos con precios más elevados dentro de su catálogo.

48. Mostrar tiendas con $\text{AVG}(\text{CantidadEnviada}) < 50$ y $\text{COUNT}(\text{NumGuia}) \geq 10$

Enunciado:

Identificar las tiendas que satisfacen simultáneamente dos condiciones:

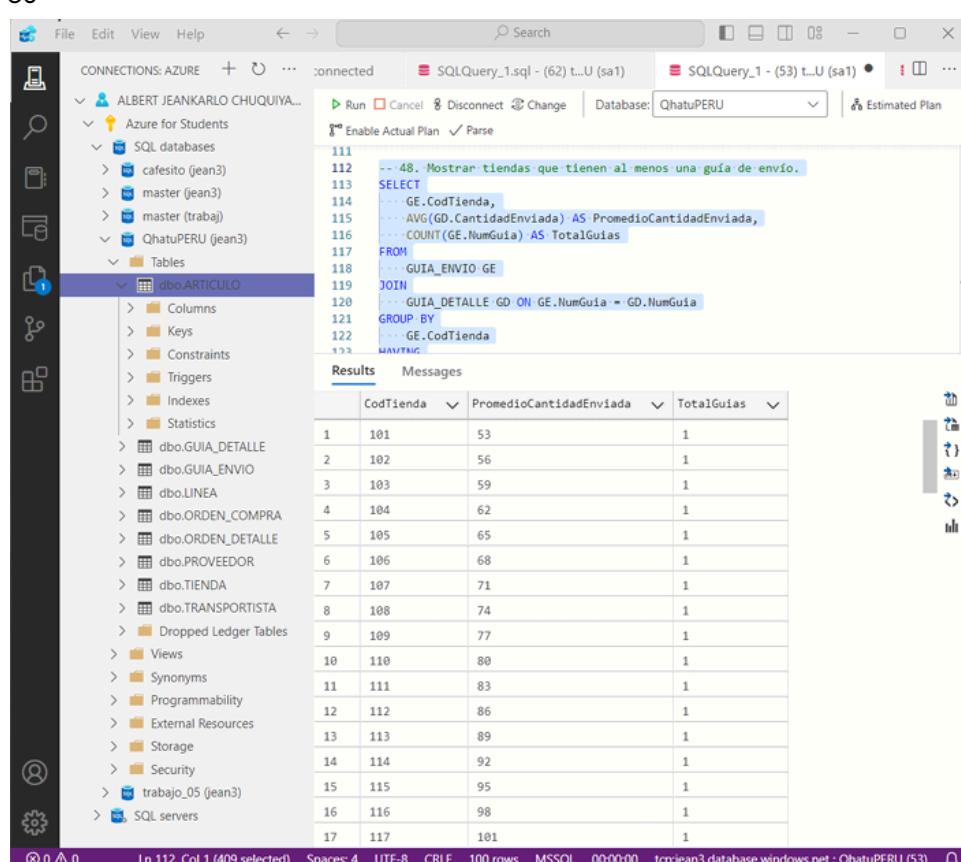
1. El promedio de unidades enviadas por guía es inferior a 50.

2. El total de guías de envío asociadas a la tienda es igual o superior a 10.

Esto permite analizar tiendas con un volumen de envíos constante pero con cantidades reducidas por guía.

Consulta SQL (sin cambios):

```
-- 48. Mostrar tiendas que tienen al menos una guía de envío.  
SELECT  
    GE.CodTienda,  
    AVG(GD.CantidadEnviada) AS PromedioCantidadEnviada,  
    COUNT(GE.NumGuia) AS TotalGuias  
FROM  
    GUIA_ENVIO GE  
JOIN  
    GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia  
GROUP BY  
    GE.CodTienda  
HAVING  
    COUNT(GE.NumGuia) >= 1 -- Solo se filtra por la existencia de la guía.  
ORDER BY  
    TotalGuias DESC;  
GO
```



The screenshot shows the SSMS interface with the following details:

- Object Explorer:** Shows the database structure under "CONNECTIONS: AZURE". It includes nodes for "ALBERT JEANKARLO CHUQUIYAHUA", "Azure for Students", "SQL databases", and "Tables". Under "Tables", "dbo.ARTICULO" is selected, showing its columns, keys, constraints, triggers, indexes, and statistics.
- Query Editor:** Contains the SQL query for question 48. The results of the query are displayed in a table titled "Results".
- Table Data:** The "Results" table has three columns: "CodTienda", "PromedioCantidadEnviada", and "TotalGuias". The data is as follows:

CodTienda	PromedioCantidadEnviada	TotalGuias
1	101	53
2	102	56
3	103	59
4	104	62
5	105	65
6	106	68
7	107	71
8	108	74
9	109	77
10	110	80
11	111	83
12	112	86
13	113	89
14	114	92
15	115	95
16	116	98
17	117	101

Explicación: Se agrupa por línea de artículo. La consulta calcula el precio máximo y el precio mínimo dentro de cada grupo mediante las funciones MAX y MIN, permitiendo observar el rango de precios por línea. En esta versión se ha eliminado la cláusula HAVING para visualizar todas las líneas, aunque el enunciado busca aquellas donde la diferencia entre MAX(PrecioProveedor) y MIN(PrecioProveedor) sea mayor a 20.

Consulta SQL (igual a la que enviaste):

```
-- 49. Mostrar CodLinea y rango de precios (Eliminando HAVING para ver todos los grupos)
SELECT
    CodLinea,
    MAX(PrecioProveedor) AS PrecioMaximo,
    MIN(PrecioProveedor) AS PrecioMinimo
FROM
    ARTICULO
GROUP BY
    CodLinea -- Se elimina la cláusula HAVING para que todos los grupos sean visibles.
ORDER BY
    CodLinea;
GO
```

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left, the Object Explorer displays the database structure under 'ALBERT JEANKARLO CHUQUIYA...'. The 'Tables' node for the 'QhatuPERU' database is expanded, and 'dbo.ARTICULO' is selected. The main window shows the 'SQLQuery_1.sql' file with the following content:

```
MAX(PrecioProveedor) AS PrecioMaximo,
MIN(PrecioProveedor) AS PrecioMinimo
FROM
    ARTICULO
GROUP BY
    CodLinea
-- Se elimina la cláusula HAVING para que todos los grupos sean visibles.
ORDER BY
    CodLinea;
GO
```

The 'Results' tab is selected, showing the output of the query:

CodLinea	PrecioMaximo	PrecioMinimo
1	3.00	3.00
2	3.50	3.50
3	4.00	4.00
4	4.50	4.50
5	5.00	5.00
6	5.50	5.50
7	6.00	6.00
8	6.50	6.50
9	7.00	7.00
10	7.50	7.50
11	8.00	8.00
12	8.50	8.50
13	9.00	9.00
14	9.50	9.50
15	10.00	10.00
16	10.50	10.50
17	11.00	11.00

The status bar at the bottom indicates: Ln 129, Col 1 (344 selected), Spaces: 4, UTF-8, CRLF, 100 rows, MSSQL, 00:00:00, tcp:jean3.database.windows.net : QhatuPERU (86).

The screenshot shows the SSMS interface with the following details:

- Left pane (Object Explorer):** Shows connections to 'AZURE' and 'QhatuPERU'. Under 'QhatuPERU', it lists 'dbo.ARTICULO' and other tables like 'dbo.GUIA_DETALLE', 'dbo.GUIA_ENVIO', etc.
- Top bar:** File, Edit, View, Help, Search, Database dropdown set to 'QhatuPERU'.
- Query Editor:** Contains the following T-SQL code:

```

152     MAX(PrecioProveedor) AS PrecioMaximo,
153     MIN(PrecioProveedor) AS PrecioMinimo
154   FROM
155     ARTICULO
156   GROUP BY
157     CodLinea
158   -- Se elimina la cláusula HAVING para que todos los grupos sean visibles.
159   ORDER BY
160     CodLinea;
161   GO
162
163   -- 50. Mostrar CodProveedor con COUNT() artículos donde AVG(StockActual) < 20 y COUNT()
164   SELECT

```
- Results Grid:** A table titled 'Results' showing data from the query. The columns are 'CodLinea', 'PrecioMaximo', and 'PrecioMinimo'. The data is as follows:

CodLinea	PrecioMaximo	PrecioMinimo
84	44.50	44.50
85	45.00	45.00
86	45.50	45.50
87	46.00	46.00
88	46.50	46.50
89	47.00	47.00
90	47.50	47.50
91	48.00	48.00
92	48.50	48.50
93	49.00	49.00
94	49.50	49.50
95	50.00	50.00
96	50.50	50.50
97	51.00	51.00
98	51.50	51.50
99	52.00	52.00
100	52.50	52.50

Explicación: Se agrupa por proveedor. La consulta calcula dos valores agregados:

- **COUNT(A.CodArticulo):** cantidad total de artículos que suministra cada proveedor.
- **AVG(A.StockActual):** stock promedio de esos artículos.

El enunciado original requería aplicar dos condiciones simultáneas mediante HAVING: proveedores con más de 5 artículos y un stock promedio menor a 20.

Sin embargo, en esta versión la cláusula HAVING fue eliminada para visualizar todos los proveedores, independientemente de si cumplen o no las condiciones.

Consulta SQL (tal cual la enviaste):

```
-- 50. Mostrar CodProveedor con COUNT() y AVG(StockActual) - Eliminando HAVING
```

```

SELECT
    A.CodProveedor,
    COUNT(A.CodArticulo) AS CantidadArticulos,
    AVG(CAST(A.StockActual AS DECIMAL(10,2))) AS PromedioStock

FROM
    ARTICULO A
GROUP BY
    A.CodProveedor -- Se elimina la cláusula HAVING. Si la tabla ARTICULO
tiene datos, esta consulta debe mostrar resultados.
ORDER BY
    CantidadArticulos DESC;
GO

```

The screenshot shows the SSMS interface with the following details:

- Object Explorer (Left):** Shows the database structure under 'ALBERT JEANKARLO CHUQUIAYA...'. The 'Tables' node is expanded, and 'dbo.ARTICULO' is selected.
- Query Window (Top Right):** Displays the T-SQL query provided in the code block.
- Results Window (Bottom Right):** Shows the execution results of the query, displaying 17 rows of data.
- Status Bar (Bottom):** Provides information about the session: 0 rows affected, Line 143, Column 1 (422 selected), and connection details: tcp://jean3.database.windows.net : QhatuPERU (86).

	CodProveedor	CantidadArticulos	PromedioStock
1	1	1	52.000000
2	2	1	54.000000
3	3	1	56.000000
4	4	1	58.000000
5	5	1	60.000000
6	6	1	62.000000
7	7	1	64.000000
8	8	1	66.000000
9	9	1	68.000000
10	10	1	70.000000
11	11	1	72.000000
12	12	1	74.000000
13	13	1	76.000000
14	14	1	78.000000
15	15	1	80.000000
16	16	1	82.000000
17	17	1	84.000000

File Edit View Help ← → Search Database: QhatuPERU Estimated Plan

CONNECTIONS: AZURE + ... connected SQLQuery_1.sql - (62) t...U (sa1) SQLQuery_1 - (53) t...U (sa1) Run Cancel Disconnect Change Database: QhatuPERU Parse

Enable Actual Plan

```

136 GROUP BY
137   | CodLinea
138   -- Se elimina la cláusula HAVING para que todos los grupos sean visibles.
139 ORDER BY
140   | CodLinea;
141 GO
142
143 -- 50. Mostrar CodProveedor con COUNT() y AVG(StockActual) - Eliminando HAVING
144 SELECT
145   ... A.CodProveedor,
146   ... COUNT(A.CodArticulo) AS CantidadArticulos,
147   ... AVG(CAST(A.StockActual AS DECIMAL(10,2))) AS PromedioStock

```

Results Messages

	CodProveedor	CantidadArticulos	PromedioStock
84	84	1	218.000000
85	85	1	220.000000
86	86	1	222.000000
87	87	1	224.000000
88	88	1	226.000000
89	89	1	228.000000
90	90	1	230.000000
91	91	1	232.000000
92	92	1	234.000000
93	93	1	236.000000
94	94	1	238.000000
95	95	1	240.000000
96	96	1	242.000000
97	97	1	244.000000
98	98	1	246.000000
99	99	1	248.000000
100	100	1	250.000000

Ln 143, Col 1 (422 selected) Spaces: 4 UTF-8 CRLF 100 rows MSSQL 00:00:00 tcp:jean3.database.windows.net:QhatuPERU (86)

Explicación: Se agrupa por proveedor. HAVING aplica la doble condición (AND) para filtrar los grupos basados en el promedio de stock actual (AVG) y la cantidad de artículos suministrados (COUNT).

