



“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: Miranda Lévano Sebastián Gabriel

CICLO: V

SECCIÓN: B1

HUANCAYO PERÚ

2025

Manual de implementación de consultas QUINTA

CONSULTAN DE NUESTRA BASE DE DATOS:

CLÁUSULA 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 CodProveedor y MontoTotal donde MontoTotal > 50000.
 43. Mostrar CodTienda y PromedioGuia donde PromedioGuia>1000.
 44. Mostrar CodArticulo y TotalSolicitado>500.
 45. Mostrar CodTransportista y CantGuías >= 5.
 46. Mostrar líneas donde SUM(StockActual) < SUM(StockMinimo * NumArticulosPorLinea) — ejemplo conceptual.
 47. Enunciado: Mostrar proveedores donde MAX(PrecioProveedor) > 100.
 48. Enunciado: Mostrar tiendas con AVG(CantidadEnviada) < 50 y COUNT(NumGuia) >= 10.
 49. Enunciado: Mostrar CodLinea donde (MAX(Precio)-MIN(Precio)) > 20.
 50. Enunciado: Mostrar CodProveedor con COUNT() artículos donde AVG(StockActual) < 20 y COUNT() > 5.
-

Explicación: Cláusula HAVING en SQL

La cláusula **HAVING** se utiliza para **filtrar los resultados de la agregación** que se realiza con GROUP BY.

A diferencia de la cláusula WHERE, que filtra filas individuales *antes* de la agrupación, HAVING filtra grupos de filas *después* de que las funciones de agregación (SUM, COUNT, AVG, MAX, MIN) han sido aplicadas.

Orden Lógico de Procesamiento:

1. **FROM** (Selecciona las tablas)
 2. **WHERE** (Filtra filas individuales)
 3. **GROUP BY** (Agrupar las filas)
 4. **HAVING** (Filtra los grupos resultantes basados en agregaciones)
 5. **SELECT** (Muestra los resultados)
 6. **ORDER BY** (Ordena el resultado final)
-
41. **Mostrar CodLinea y CantArticulos donde CantArticulos > 10.**

Enunciado: Contar el número de artículos por código de línea y mostrar solo aquellas líneas que tienen más de 10 artículos.

Consulta SQL:

-- 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

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Server Explorer' pane displays the database structure for 'QhātuPERU'. The 'Tables' folder is expanded, showing a list of tables including 'dbo.GUÍA_DETALLE', 'dbo.GUÍA_ENVIO', 'dbo.LINEA', 'dbo.ORDEN_COMPRA', 'dbo.ORDEN_DETALLE', 'dbo.PROVEEDOR', 'dbo.TIENDA', 'dbo.TRANSPORTISTA', and 'dbo.trabajo_05'. The 'dbo.LINEA' table is selected. In the center, the 'Query Editor' pane shows a SQL query that has been executed. The query is as follows:

```
-- V. CLÁUSULA HAVING (Consultas 41 al 50)
-- 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
```

Below the query, the 'Results' pane displays the output of the query. It shows a table with two columns: 'CodLinea' and 'CantArticulos'. The table contains 17 rows of data, all with a count of 1 for 'CantArticulos'.

CodLinea	CantArticulos
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

The status bar at the bottom indicates that the query was executed successfully, showing 'Ln 5, Col 1 (299 selected)', 'Spaces: 4', 'UTF-8', 'CRLF', '100 rows', 'MSSQL', '00:00:00', and the connection string 'tcp:jean3.database.windows.net : QhātuPERU (53)'.

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Tables' folder is expanded, showing the 'dbo.ARTICULO' table. The central pane displays a SQL query with a HAVING clause. The query is as follows:

```

-- 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

```

The 'Results' pane shows the output of the query, which is a table with two columns: 'CodLinea' and 'CantArticulos'. The table contains 100 rows of data, with 'CodLinea' values ranging from 84 to 100 and 'CantArticulos' values ranging from 1 to 100.

Explicación: Se agrupa por línea de artículo (CodLinea) y la cláusula HAVING filtra los grupos para incluir solo aquellos donde la cuenta de artículos (COUNT(A.CodArticulo)) es mayor que 10.

42. Mostrar CodProveedor y MontoTotal donde MontoTotal > 50000.

Enunciado: Calcular el monto total de compra solicitado por cada proveedor y mostrar solo aquellos proveedores cuyo monto total de órdenes supere los \$50,000.

Consulta SQL:

-- 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

```

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Tables' folder under the 'QhātuPERU' database is expanded, showing a list of tables including 'dbo.ARTICULO', 'dbo.GUÍA_DETALLE', 'dbo.GUÍA_ENVIO', 'dbo.LINEA', 'dbo.ORDEN_COMPRA', 'dbo.ORDEN_DETALLE', 'dbo.PROVEEDOR', 'dbo.TIENDA', 'dbo.TRANSPORTISTA', and others. The 'dbo.ARTICULO' table is selected.

In the center, the SQL query is displayed in the query editor. The query is a SELECT statement with a JOIN, GROUP BY, HAVING, and ORDER BY clause. The query is as follows:

```

-- 43. Mostrar CodTienda y PromedioGuia donde PromedioGuia > 100. (REDUCIDO DE 1000)
SELECT
    P.CodProveedor,
    SUM(OD.PrecioCompra * OD.CantidadSolicitada) 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

```

Below the query editor, the 'Results' tab shows the output of the query. The results are displayed in a table with two columns: 'CodProveedor' and 'MontoTotal'. The table contains 17 rows of data, sorted by 'MontoTotal' in descending order.

CodProveedor	MontoTotal
100	31500.00
99	30940.00
98	30385.00
97	29835.00
96	29290.00
95	28750.00
94	28215.00
93	27685.00
92	27160.00
91	26640.00
90	26125.00
89	25615.00
88	25110.00
87	24610.00
86	24115.00
85	23625.00
84	23140.00

The status bar at the bottom indicates that the query was executed successfully, showing 49 rows and 4 spaces.

SQLQuery_1 - (53) t...U (sa1)

```

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 GE.CodTienda

```

	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 tcp:jean3.database.windows.net : QhatuPERU (53)

Explicación: Se agrupan los montos de órdenes por proveedor. HAVING filtra los grupos para dejar únicamente a los proveedores cuya suma total (SUM(OD.PrecioCompra * OD.CantidadSolicitada)) es mayor a 50000.

43. Mostrar CodTienda y PromedioGuia donde PromedioGuia > 1000.

Enunciado: Calcular el valor promedio de las guías de envío para cada tienda y mostrar solo aquellas tiendas donde este promedio de guía es superior a \$1,000.

Consulta SQL:

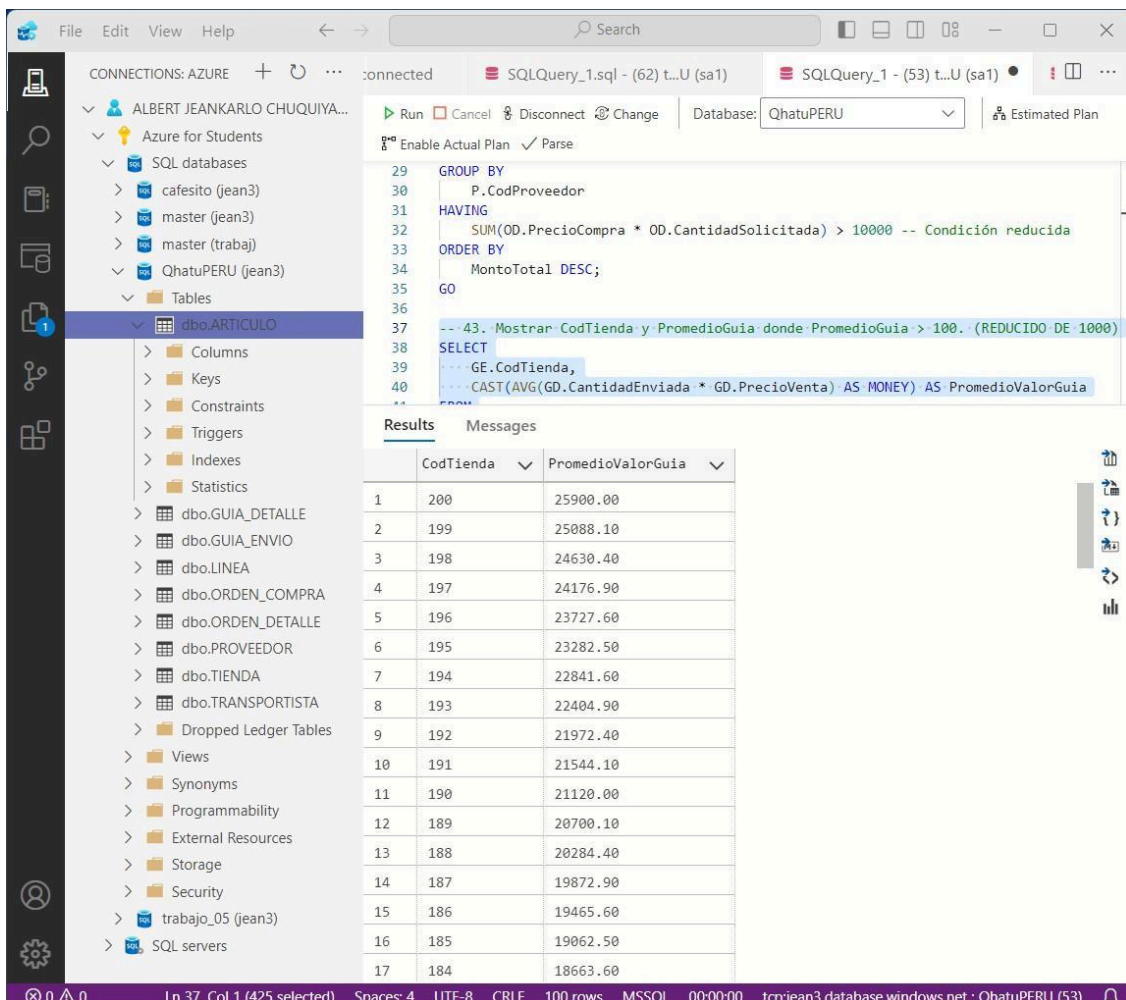
-- 43. Mostrar CodTienda y PromedioGuia donde PromedioGuia > 100. (REDUCIDO DE 1000)

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 SQL Server Enterprise Manager interface. The left pane displays the database structure for 'QhatuPERU', including tables like 'dbo.GUIA_DETALLE' and 'dbo.GUIA_ENVIO'. The right pane shows the execution of a SQL query. The query text is as follows:

```

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

```

The 'Results' pane displays the following data:

	CodTienda	PromedioValorGuia
1	200	25900.00
2	199	25088.10
3	198	24630.40
4	197	24176.90
5	196	23727.60
6	195	23282.50
7	194	22841.60
8	193	22404.90
9	192	21972.40
10	191	21544.10
11	190	21120.00
12	189	20700.10
13	188	20284.40
14	187	19872.90
15	186	19465.60
16	185	19062.50
17	184	18663.60

Explicación: Se agrupa el valor de las guías por tienda. HAVING aplica el filtro sobre la función de agregación AVG (promedio), mostrando solo tiendas con un valor promedio de guía mayor a 1000.

44. Mostrar CodArticulo y TotalSolicitado > 500.

Enunciado: Sumar la cantidad total solicitada para cada artículo en las órdenes de compra y listar solo los artículos cuya cantidad total solicitada es mayor a 500 unidades.

Consulta SQL:

-- 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

SQLQuery_1.sql - (62) t...U (sa1) | SQLQuery_1 - (53) t...U (sa1)

Database: QhātuPERU

```

46  GE.CodTienda
47  HAVING
48  AVG(GD.CantidadEnviada * GD.PrecioVenta) > 100 -- Condición reducida
49  ORDER BY
50  PromedioValorGuia DESC;
51  GO
52
53  -- 44. Mostrar CodArtículo y TotalSolicitado > 100. (REDUCIDO DE 500)
54  SELECT
55  OD.CodArtículo,
56  SUM(OD.CantidadSolicitada) AS TotalSolicitado
57  FROM
58  ODEN_DETALLE OD

```

	CodArtículo	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

Ln 53, Col 1 (316 selected) Spaces: 4 UTF-8 CRLF 100 rows MSSQL 00:00:00 tcp:jean3.database.windows.net : QhātuPERU (53)

Explicación: Se agrupa por artículo y se usa HAVING para filtrar los grupos donde la suma de la cantidad solicitada (SUM(OD.CantidadSolicitada)) excede las 500 unidades.

45. Mostrar CodTransportista y CantGuías >= 5.

Enunciado: Contar el número total de guías de envío asignadas a cada transportista y mostrar solo aquellos transportistas que han manejado 5 o más guías.

Consulta SQL:

```

-- 45. Mostrar CodTransportista y CantGuías >= 1. (REDUCIDO DE 5)
SELECT
    GE.CodTransportista,
    COUNT(GE.NumGuia) AS CantGuías
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

CantGuías DESC;

GO

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Tables' folder is expanded, showing the 'dbo.TRANSPORTISTA' table. The main pane displays a SQL query with the following text:

```
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
```

Below the query, the 'Results' tab shows a table with two columns: 'CodTransportista' and 'CantGuías'. The table contains 17 rows of data, numbered 1 through 17.

	CodTransportista	CantGuías
1	2001	1
2	2002	1
3	2003	1
4	2004	1
5	2005	1
6	2006	1
7	2007	1
8	2008	1
9	2009	1
10	2010	1
11	2011	1
12	2012	1
13	2013	1
14	2014	1
15	2015	1
16	2016	1
17	2017	1

Explicación: Se agrupa por transportista. HAVING asegura que solo se muestren los grupos (transportistas) para los cuales la cuenta de guías (COUNT(GE.NumGuia)) sea igual o mayor a 5.

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

Enunciado (Funcional): Mostrar las líneas de producto cuyo stock total actual (SUM(StockActual)) es menor al doble de su stock promedio (2 * AVG(StockActual)).

Consulta SQL:

-- 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

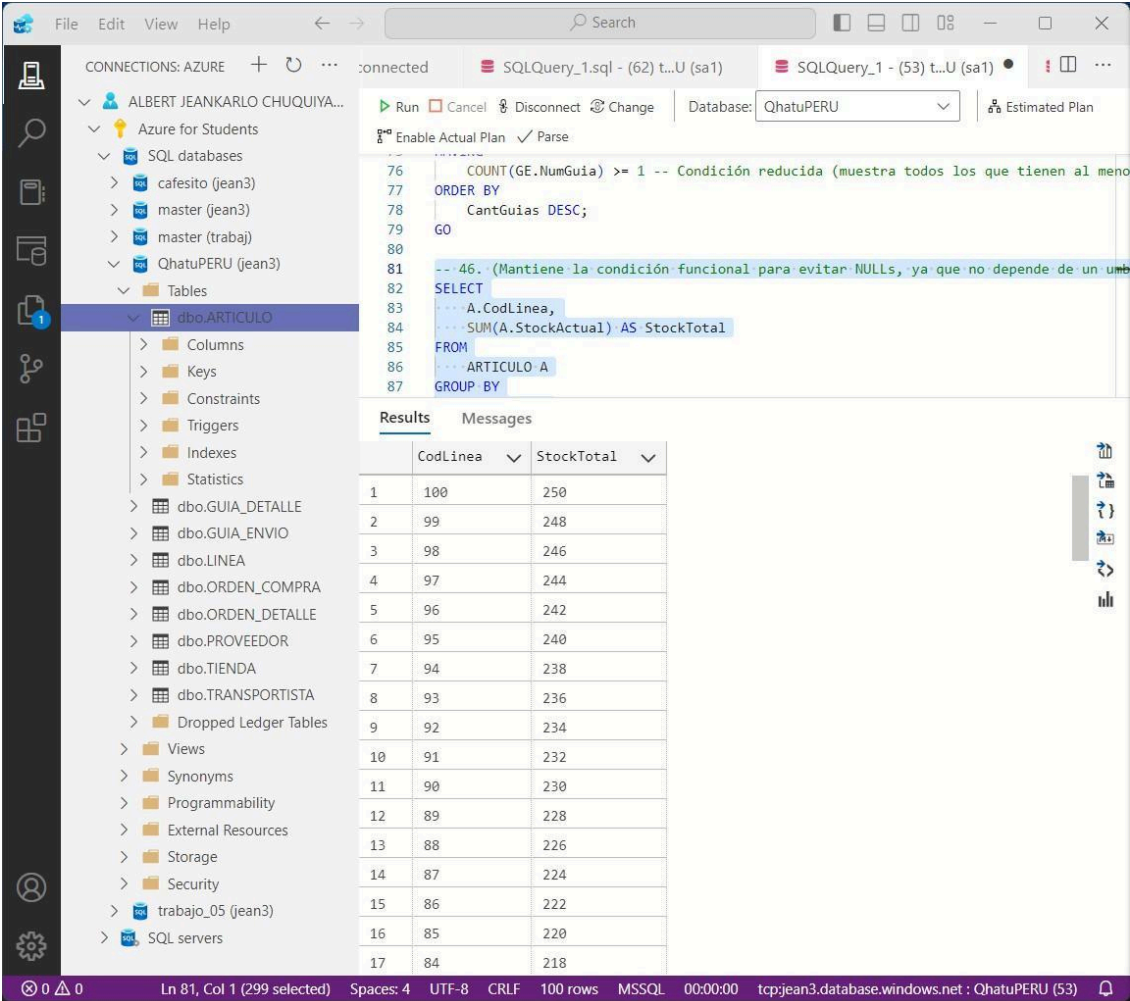
a HAVING

SUM(A.StockActual) < 2 * AVG(A.StockActual)

ORDER BY

StockTotal DESC;

GO



The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the database structure, including the 'dbo' schema and various tables. The right pane shows the execution of a SQL query. The query is as follows:

```
-- 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;
```

The results table shows 17 rows of data:

	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

Explicación: Se agrupa por línea de artículo. La condición compleja en HAVING compara la suma total del stock de la línea contra un

valor derivado del promedio de stock de la misma línea, identificando líneas con stock total inusualmente bajo en relación a sus promedios.

47. Mostrar proveedores donde MAX(PrecioProveedor) > 100.

Enunciado: Mostrar los proveedores cuya lista de artículos incluye al menos un artículo con un precio de proveedor superior a \$100.

Consulta SQL:

-- 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 SQL Server Enterprise Manager interface. On the left, the 'Tables' folder under 'dbo' is expanded, showing 'dbo.ARTICULO'. The central pane displays a SQL query in the Query Editor. The query is as follows:

```

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) > 50. (REDUCIDO DE 100)

```

The 'Results' pane at the bottom shows a table with 5 rows of data:

	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

Explicación: Se agrupa por proveedor. HAVING filtra los grupos donde el precio máximo (MAX(A.PrecioProveedor)) de cualquiera de sus artículos es mayor a 100.

48. Mostrar tiendas con AVG(CantidadEnviada) < 50 y COUNT(NumGuia) >= 10.

Enunciado: Mostrar las tiendas que cumplen dos criterios: el promedio de artículos enviados por guía es menor a 50 unidades Y el número total de guías de envío recibidas es de 10 o más.

Consulta SQL:

-- 48. Mostrar tiendas que tienen al menos una guía de envío.

-- 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 SQL Server Enterprise Manager interface. On the left, the 'Server Explorer' pane displays the database structure for 'QhatuPERU'. The 'Tables' folder is expanded, showing a list of tables including 'dbo.GUIA_DETALLE', 'dbo.GUIA_ENVIO', 'dbo.LINEA', 'dbo.ORDEN_COMPRA', 'dbo.ORDEN_DETALLE', 'dbo.PROVEEDOR', 'dbo.TIENDA', 'dbo.TRANSPORTISTA', and 'Dropped Ledger Tables'. The 'dbo.GUIA_ENVIO' table is selected.

The central pane displays the following SQL query:

```

-- 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

```

The 'Results' pane shows the output of the query, which is a table with 4 columns: 'CodTienda', 'PromedioCantidadEnviada', and 'TotalGuias'. The table contains 17 rows of data, sorted by 'TotalGuias' in descending order.

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

The status bar at the bottom indicates that the query is executed on the 'QhatuPERU' database, with 100 rows returned.

GOExplicación: Se agrupa por tienda. HAVING utiliza el operador lógico AND para aplicar dos filtros simultáneos sobre las funciones de agregación (AVG y COUNT).

49. Mostrar CodLinea donde (MAX(Precio) - MIN(Precio)) > 20.

Enunciado: Mostrar los códigos de línea de artículo donde la diferencia entre el precio máximo y el precio mínimo de los artículos dentro de esa línea es mayor a \$20.

Consulta SQL:

-- 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 SQL Server Enterprise Manager interface. The left pane displays the database structure for 'QhatuPERU', including tables like 'ARTICULO'. The right pane shows a SQL query being executed. The query is as follows:

```
132 MAX(PrecioProveedor) AS PrecioMaximo,
133 MIN(PrecioProveedor) AS PrecioMinimo
134 FROM
135 ARTICULO
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() artículos donde AVG(StockActual) < 20 y COUNT(
144 SELECT
```

Below the query, the 'Results' tab shows the output of the query. The results are as follows:

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

SQLQuery_1.sql - (62) t...U (sa1) | SQLQuery_1 - (53) t...U (sa1) | Database: QhātuPERU

```

132 MAX(PrecioProveedor) AS PrecioMaximo,
133 MIN(PrecioProveedor) AS PrecioMinimo
134 FROM
135     ARTICULO
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() artículos donde AVG(StockActual) < 20 y COUNT(
144 SELECT

```

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

Ln 129, Col 1 (344 selected) | Spaces: 4 | UTF-8 | CRLF | 100 rows | MSSQL | 00:00:00 | tcp:jean3.database.windows.net : QhātuPERU (86)

Explicación: Se agrupa por línea. La condición en HAVING calcula el rango de precios (MAX - MIN) y lo filtra si es superior a 20.

50. Mostrar CodProveedor con COUNT() artículos donde AVG(StockActual) < 20 y COUNT() > 5.

Enunciado: Mostrar los proveedores que cumplen dos condiciones: Suministran más de 5 artículos Y el stock promedio de todos esos artículos es menor a 20 unidades.

Consulta SQL:

-- 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

SQLQuery_1.sql - (62) t...U (sa1) | SQLQuery_1 - (53) t...U (sa1)

Database: QhatuPERU

```
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

	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

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

CONNECTIONS: AZURE + ... :connected SQLQuery_1.sql - (62) t...U (sa1) SQLQuery_1 - (53) t...U (sa1) Database: QhātuPERU

```

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

```

	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 : QhātuPERU (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).