

UNIVERSIDAD PERUANA LOS ANDES

FACULTAD DE INGENIERÍA

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS
Y COMPUTACIÓN**



UPLA

SEMANA 09

ASIGNATURA: Base de Datos II

ESTUDIANTE: Luis Enrique Espejo Quispe

DOCENTE: Mg. Raúl Fernández Bejarano

CICLO: V

SECCIÓN: A1

HYO-2025

SECCIÓN I: FUNCIONES DE AGREGACIÓN

DEFINICIÓN

Son funciones especiales que permiten calcular valores a partir de un conjunto de datos en una o más columnas.

UTILIDAD

Se utilizan para resumir información numérica o estadística, como totales, promedios o conteos. Muy útiles para analizar inventarios, cantidades, precios, ventas, pedidos, etc.

EJEMPLOS DE USO:

- Calcular el total de inventario de una tienda.
- Obtener el precio máximo y mínimo de productos.
- Contar cuántos proveedores existen.
- Obtener el promedio de días de entrega de pedidos.

EJERCICIOS DE APLICACIÓN:

1. Mostrar CodArticulo, DescripciónArticulo y ValorInventario

```

8  -- 1. Mostrar CodArticulo, DescripcionArticulo y ValorInventario
9  SELECT
10    A.CodArticulo,
11    A.DescripcionArticulo,
12    CAST(A.StockActual * A.PrecioProveedor AS DECIMAL(18,2)) AS ValorInventario
13  FROM ARTICULO A
14  ORDER BY A.CodArticulo;
15  GO

```

75 %

	CodArticulo	DescripcionArticulo	ValorInventario
1	1	Desktop i7 16GB	18000.00
2	2	Laptop i5 8GB 256GB	17000.00
3	3	iPad Air 64GB	7800.00
4	4	Samsung Galaxy S23	18750.00
5	5	Monitor LG 274K	7200.00
6	6	Teclado Mecánico RGB	6750.00
7	7	AirPods Pro	8400.00
8	8	Logitech C920 Webcam	3325.00
9	9	Impresora HP LaserJet	2800.00
10	10	Scanner ScanJet Pro	2400.00
11	11	Cable HDMI 2.1	2500.00
12	12	Cargador Rápido 65W	3600.00
13	13	Power Bank 25000mAh	3575.00
14	14	Disco Duro Externo 1TB	3570.00
15	15	Tarjeta MicroSD 256GB	4200.00
16	16	Monitor Digital 55	4250.00
17	17	Ventilador Noctua	3000.00
18	18	Fuente EVGA 850W	3960.00
19	19	Cable Cat6 20mts	2250.00
20	20	Funda Laptop 15.6	3150.00
21	21	Mochila Gaming	3000.00
22	22	Stand Monitor Doble	2160.00
23	23	Docking Station USB-C	6250.00
24	24	Hub USB 7 Puertos	2925.00
25	25	Base Refriger Laptop	1925.00
26	26	Mouse Pad Grande R...	3200.00
27	27	Protector Pantalla	2000.00
28	28	Lentes Gaming	2660.00
29	29	Silla Gamer Ergon	4500.00
30	30	Escritorio Gaming	3040.00
31	31	Controlador Xbox	3575.00
32	32	Headset Gaming Inal	5760.00
33	33	Ratón Gaming Inal	3600.00
34	34	Teclado Mecánico G	6000.00
35	35	Alfombrilla Gaming	3500.00
36	36	Monitor 144Hz 27	5320.00
37	37	Tarjeta RTX 4060	3840.00
38	38	Procesador Ryzen 7	3800.00
39	39	Memoria RAM 32GB	3200.00
40	40	Placa Madre ASUS R...	2240.00
41	41	Case Gaming Temper...	3240.00
42	42	Illuminación RGB LED	4500.00
43	43	Ventilador RGB Pack	4500.00
44	44	Tubos Sleeving Color	2750.00
45	45	Pasta Térmica Prem	1900.00
46	46	Aire Comprimido 420	1950.00
47	47	Pulsera Antiestática	1920.00
48	48	Kit Herramientas	2450.00
49	49	Manual Garantía Ext	10000.00
50	50	Servicio Instalación	10000.00

2. Calcular el total monetario del inventario

```
17 -- 2. Calcular el total monetario del inventario
18 SELECT
19     SUM(CAST(A.StockActual * A.PrecioProveedor AS DECIMAL(18,2))) AS TotalInventario
20     FROM ARTICULO A;
21
22
```

75 %

Results Messages

	TotalInventario
1	242185.00

3. Obtener CodLinea y PrecioProveedor promedio

```
23 -- 3. Obtener CodLinea y PrecioProveedor promedio
24 SELECT DISTINCT
25     A.CodLinea,
26     A.PrecioProveedor
27     FROM ARTICULO A
28 ORDER BY A.CodLinea;
29 GO
```

75 %

Results Messages

	CodLinea	PrecioProveedor
1	1	1200.00
2	2	850.00
3	3	650.00
4	4	750.00
5	5	400.00
6	6	150.00
7	7	280.00
8	8	95.00
9	9	350.00
10	10	200.00
11	11	25.00
12	12	45.00
13	13	65.00
14	14	85.00
15	15	35.00
16	16	850.00
17	17	50.00
18	18	180.00
19	19	30.00
20	20	35.00
21	21	75.00
22	22	120.00
23	23	250.00
24	24	45.00
25	25	55.00
26	26	40.00
27	27	20.00
28	28	95.00
29	29	450.00

30	30	380,00
31	31	65,00
32	32	180,00
33	33	75,00
34	34	200,00
35	35	50,00
36	36	380,00
37	37	320,00
38	38	380,00
39	39	160,00
40	40	280,00
41	41	180,00
42	42	60,00
43	43	90,00
44	44	25,00
45	45	20,00
46	46	15,00
47	47	12,00
48	48	35,00
49	49	50,00
50	50	100,00

4. Contar artículos descontinuados

```

50
31 -- 4. Contar artículos descontinuados
32 SELECT
33 COUNT(*) AS CantidadDescontinuados
34 FROM ARTICULO A
35 WHERE A.Descontinuado = 1;
36 GO

```

75 %

	CantidadDescontinuados
1	0

5. Mostrar PrecioMaximo y PrecioMinimo del catálogo

```

-- 5. Mostrar PrecioMaximo y PrecioMinimo del catálogo
SELECT
    MAX(A.PrecioProveedor) AS PrecioMaximo,
    MIN(A.PrecioProveedor) AS PrecioMinimo
FROM ARTICULO A;
GO

```

75 %

Results Messages

	PrecioMaximo	PrecioMinimo
1	1200.00	12.00

6. Mostrar el Valor total enviado por guía

```

-- 6. Mostrar el Valor total enviado por guía
SELECT
    GE.NumGuia,
    SUM(CAST(GD.PrecioVenta * GD.CantidadEnviada AS DECIMAL(18,2))) AS ValorTotalEnviado
FROM GUIA_ENVIO GE
INNER JOIN GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
GROUP BY GE.NumGuia
ORDER BY GE.NumGuia;
GO

```

75 %

Results Messages

	NumGuia	ValorTotalEnviado
1	1001	11600.00
2	1002	12600.00
3	1003	5100.00
4	1004	9500.00
5	1005	2750.00
6	1006	3600.00
7	1007	7600.00
8	1008	3500.00
9	1009	1920.00
10	1010	2100.00
11	1011	1575.00
12	1012	2275.00
13	1013	2850.00
14	1014	2160.00
15	1015	2750.00
16	1016	2200.00
17	1017	2100.00
18	1018	2400.00
19	1019	1710.00
20	1020	2400.00
21	1021	2200.00
22	1022	1440.00
23	1023	4160.00
24	1024	2080.00
25	1025	1280.00
26	1026	2310.00
27	1027	1740.00
28	1028	1690.00

29	1029	2900.00
30	1030	1500.00
31	1031	2520.00
32	1032	3840.00
33	1033	2300.00
34	1034	4050.00
35	1035	2660.00
36	1036	3500.00
37	1037	2520.00
38	1038	2000.00
39	1039	2420.00
40	1040	1110.00
41	1041	2160.00
42	1042	3040.00
43	1043	3120.00
44	1044	2380.00
45	1045	1440.00
46	1046	1716.00
47	1047	1584.00
48	1048	1900.00
49	1049	6860.00
50	1050	6720.00

7. Para cada CodArticulo, mostrar TotalSolicitado

```

55 -- 7. Para cada CodArticulo, mostrar TotalSolicitado
56 SELECT
57     OD.CodArticulo,
58     SUM(OD.CantidadSolicitada) AS TotalSolicitado
59     FROM ORDEN_DETALLE OD
60     GROUP BY OD.CodArticulo
61     ORDER BY OD.CodArticulo;
62 GO

```

75 %

Results Messages

	CodArticulo	TotalSolicitado
1	1	10
2	2	15
3	3	8
4	4	12
5	5	6
6	6	20
7	7	25
8	8	30
9	9	5
10	10	8
11	11	50
12	12	40
13	13	35
14	14	20
15	15	60
16	16	3
17	17	30
18	18	12
19	19	40
20	20	50
21	21	25
22	22	10
23	23	15
24	24	35
25	25	18
26	26	45
27	27	60
28	28	15

29	29	6
30	30	4
31	31	30
32	32	18
33	33	25
34	34	16
35	35	40
36	36	8
37	37	7
38	38	5
39	39	12
40	40	4
41	41	10
42	42	40
43	43	28
44	44	70
45	45	50
46	46	80
47	47	90
48	48	40
49	49	100
50	50	50

8. Contar órdenes únicas que incluyen cada artículo

```

64 -- 8. Contar órdenes únicas que incluyen cada artículo
65 SELECT
66     OD.CodArticulo,
67     COUNT(DISTINCT OD.NumOrden) AS CantidadOrdenes
68     FROM ORDEN_DETALLE OD
69     GROUP BY OD.CodArticulo
70     ORDER BY OD.CodArticulo;
71 GO
72

```

75 %

Results Messages

	CodArticulo	CantidadOrdenes
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
18	18	1
19	19	1
20	20	1
21	21	1
22	22	1
23	23	1
24	24	1
25	25	1
26	26	1
27	27	1
28	28	1

29	29	1
30	30	1
31	31	1
32	32	1
33	33	1
34	34	1
35	35	1
36	36	1
37	37	1
38	38	1
39	39	1
40	40	1
41	41	1
42	42	1
43	43	1
44	44	1
45	45	1
46	46	1
47	47	1
48	48	1
49	49	1
50	50	1

9. Calcular promedio de días para todas las órdenes con FechaSalida

```

73 -- 9. Calcular promedio de días para todas las órdenes con FechaSalida
74 SELECT
75     AVG(CAST(DATEDIFF(DAY, OC.FechaOrden, GE.FechaSalida) AS DECIMAL(10,2))) AS PromedioDias
76     FROM ORDEN_COMPRA OC
77     INNER JOIN GUIA_ENVIO GE ON 1=1
78     WHERE OC.FechaIngreso IS NOT NULL
79     AND GE.FechaSalida IS NOT NULL;
80     GO
81

```

PromedioDias
5.100000

10. Sumar CantidadEnviada por CodTransportista

```

82 -- 10. Sumar CantidadEnviada por CodTransportista
83 SELECT
84     GE.CodTransportista,
85     SUM(GD.CantidadEnviada) AS TotalEnviado
86 FROM GUIA_ENVIO GE
87 INNER JOIN GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
88 GROUP BY GE.CodTransportista
89 ORDER BY GE.CodTransportista;
90 GO

```

75 %

Results Messages

	CodTransportista	TotalEnviado
1	501	8
2	502	12
3	503	6
4	504	10
5	505	5
6	506	18
7	507	20
8	508	25
9	509	4
10	510	7
11	511	45
12	512	35
13	513	30
14	514	18
15	515	55
16	516	2
17	517	28
18	518	10
19	519	38
20	520	48
21	521	22
22	522	9
23	523	13
24	524	32
25	525	16
26	526	42
27	527	58
28	528	13
29	529	5
30	530	3
31	531	28
32	532	16
33	533	23
34	534	15
35	535	38
36	536	7
37	537	6
38	538	4
39	539	11
40	540	3
41	541	9
42	542	38
43	543	26
44	544	68
45	545	48
46	546	78
47	547	88
48	548	38
49	549	98
50	550	48

SECCIÓN II: CLÁUSULA GROUP BY

DEFINICIÓN

Es una cláusula de SQL que agrupa datos por una o varias columnas para luego aplicar funciones de agregación.

UTILIDAD

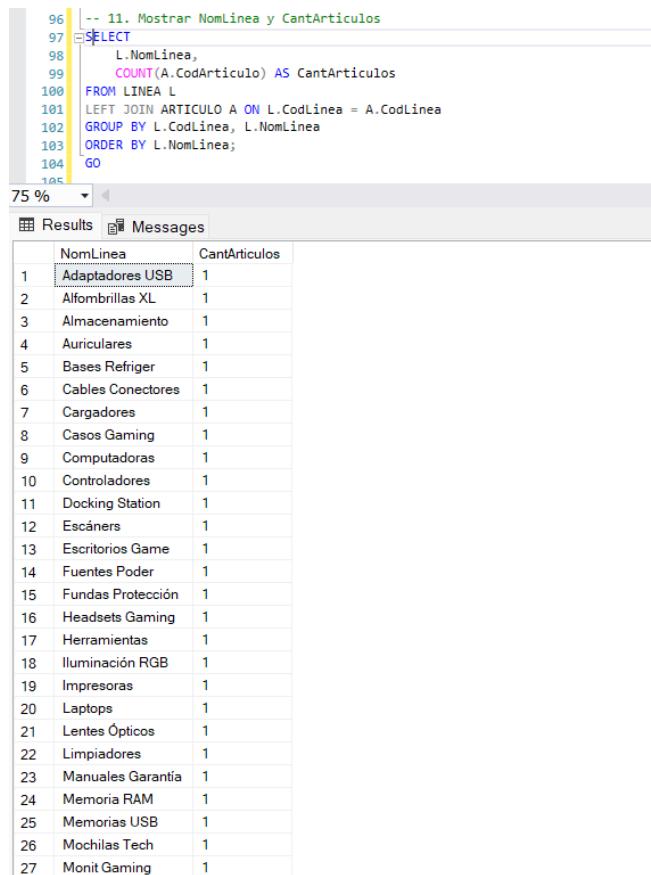
Permite obtener resultados resumidos por categorías, como por tienda, proveedor, línea de producto, etc.

EJEMPLOS DE USO:

- Mostrar totales por categoría de producto.
- Calcular ventas por tienda o proveedor.
- Obtener el promedio de inventario por línea de artículos.

EJERCICIOS DE APLICACIÓN

11. Mostrar NomLinea y CantArticulos



The screenshot shows a SQL query window with the following code:

```
96 -- 11. Mostrar NomLinea y CantArticulos
97 SELECT
98     L.NomLinea,
99     COUNT(A.CodArticulo) AS CantArticulos
100    FROM LINEA L
101   LEFT JOIN ARTICULO A ON L.CodLinea = A.CodLinea
102   GROUP BY L.CodLinea, L.NomLinea
103   ORDER BY L.NomLinea;
104
105 GO
```

The results table displays the following data:

	NomLinea	CantArticulos
1	Adaptadores USB	1
2	Alfombrillas XL	1
3	Almacenamiento	1
4	Auriculares	1
5	Bases Refriger	1
6	Cables Conectores	1
7	Cargadores	1
8	Casos Gaming	1
9	Computadoras	1
10	Controladores	1
11	Docking Station	1
12	Escáneres	1
13	Escritorios Game	1
14	Fuentes Poder	1
15	Fundas Protección	1
16	Headsets Gaming	1
17	Herramientas	1
18	Illuminación RGB	1
19	Impresoras	1
20	Laptops	1
21	Lentes Ópticos	1
22	Limpiaadores	1
23	Manuales Garantía	1
24	Memoria RAM	1
25	Memorias USB	1
26	Mochilas Tech	1
27	Monit Gaming	1

28	Monitores	1
29	Monitores Dig	1
30	Mouse Pads	1
31	Pasta Térmica	1
32	Placas Madre	1
33	Power Banks	1
34	Procesadores	1
35	Protec Antiestát	1
36	Protect Pantalla	1
37	Ratones Gaming	1
38	Refrigeración	1
39	Servicios Técnicos	1
40	Sillas Gamer	1
41	Smartphones	1
42	Stands Soportes	1
43	Tablets	1
44	Tarjet Gráficas	1
45	Tarjetas Mem	1
46	Teclados	1
47	Teclados Mecán	1
48	Tubos Sleeving	1
49	Ventiladores RGB	1
50	Webcams	1

12. Mostrar CodLinea y StockTotal

```

106 -- 12. Mostrar CodLinea y StockTotal
107 SELECT
108   L.CodLinea,
109   SUM(A.StockActual) AS StockTotal
110 FROM LINEA L
111 INNER JOIN ARTICULO A ON L.CodLinea = A.CodLinea
112 GROUP BY L.CodLinea
113 ORDER BY L.CodLinea;
114 GO
115

```

75 % ▾

	CodLinea	StockTotal
1	1	15
2	2	20
3	3	12
4	4	25
5	5	18
6	6	45
7	7	30
8	8	35
9	9	8
10	10	12
11	11	100
12	12	80
13	13	55
14	14	42
15	15	120
16	16	5
17	17	60
18	18	22
19	19	75
20	20	90
21	21	40
22	22	18
23	23	25
24	24	65
25	25	35
26	26	80
27	27	100

28	28	28
29	29	10
30	30	8
31	31	55
32	32	32
33	33	48
34	34	30
35	35	70
36	36	14
37	37	12
38	38	10
39	39	20
40	40	8
41	41	18
42	42	75
43	43	50
44	44	110
45	45	95
46	46	130
47	47	160
48	48	70
49	49	200
50	50	100

13. Para cada NumOrden, calcular CostoTotal = SUM(PrecioCompra*Cantidad)

```

116 -- 13. Para cada NumOrden, calcular CostoTotal = SUM(PrecioCompra*Cantidad)
117 SELECT
118     OD.NumOrden,
119     SUM(CAST(OD.PrecioCompra * OD.CantidadSolicitada AS DECIMAL(18,2))) AS CostoTotal
120     FROM ORDEN_DETALLE OD
121     GROUP BY OD.NumOrden
122     ORDER BY OD.NumOrden;
123 GO
124
125 -- 14. Mostrar NumGuia v. PromedioEnviado

```

75 %

Results Messages

	NumOrden	CostoTotal
1	2025001	12000.00
2	2025002	12750.00
3	2025003	5200.00
4	2025004	9000.00
5	2025005	2400.00
6	2025006	3000.00
7	2025007	7000.00
8	2025008	2850.00
9	2025009	1750.00
10	2025010	1600.00
11	2025011	1250.00
12	2025012	1800.00
13	2025013	2275.00
14	2025014	1700.00
15	2025015	2100.00
16	2025016	2550.00
17	2025017	1500.00
18	2025018	2160.00
19	2025019	1200.00
20	2025020	1750.00
21	2025021	1875.00
22	2025022	1200.00
23	2025023	3750.00
24	2025024	1575.00
25	2025025	990.00
26	2025026	1800.00
27	2025027	1200.00

28	2025028	1425.00
29	2025029	2700.00
30	2025030	1520.00
31	2025031	1950.00
32	2025032	3240.00
33	2025033	1875.00
34	2025034	3200.00
35	2025035	2000.00
36	2025036	3040.00
37	2025037	2240.00
38	2025038	1900.00
39	2025039	1920.00
40	2025040	1120.00
41	2025041	1800.00
42	2025042	2400.00
43	2025043	2520.00
44	2025044	1750.00
45	2025045	1000.00
46	2025046	1200.00
47	2025047	1080.00
48	2025048	1400.00
49	2025049	5000.00
50	2025050	5000.00

14. Mostrar NumGuia y PromedioEnviado

```

125 -- 14. Mostrar NumGuia y PromedioEnviado
126 SELECT
127     GE.NumGuia,
128     AVG(CAST(GD.CantidadEnviada AS DECIMAL(10,2))) AS PromedioEnviado
129     FROM GUIA_ENVIO GE
130     INNER JOIN GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
131     GROUP BY GE.NumGuia
132     ORDER BY GE.NumGuia;
133 GO
134

```

75 %

Results Messages

	NumGuia	PromedioEnviado
1	1001	8.000000
2	1002	12.000000
3	1003	6.000000
4	1004	10.000000
5	1005	5.000000
6	1006	18.000000
7	1007	20.000000
8	1008	25.000000
9	1009	4.000000
10	1010	7.000000
11	1011	45.000000
12	1012	35.000000
13	1013	30.000000
14	1014	18.000000
15	1015	55.000000
16	1016	2.000000
17	1017	28.000000
18	1018	10.000000
19	1019	38.000000
20	1020	48.000000
21	1021	22.000000
22	1022	9.000000
23	1023	13.000000
24	1024	32.000000
25	1025	16.000000
26	1026	42.000000
27	1027	58.000000

28	1028	13.000000
29	1029	5.000000
30	1030	3.000000
31	1031	28.000000
32	1032	16.000000
33	1033	23.000000
34	1034	15.000000
35	1035	38.000000
36	1036	7.000000
37	1037	6.000000
38	1038	4.000000
39	1039	11.000000
40	1040	3.000000
41	1041	9.000000
42	1042	38.000000
43	1043	26.000000
44	1044	68.000000
45	1045	48.000000
46	1046	78.000000
47	1047	88.000000
48	1048	38.000000
49	1049	98.000000
50	1050	48.000000

15. Contar proveedores agrupados por Ciudad

```

135 -- 15. Contar proveedores agrupados por Departamento
136 SELECT
137     P.Departamento,
138     COUNT(P.CodProveedor) AS CantProveedores
139 FROM PROVEEDOR P
140 GROUP BY P.Departamento
141 ORDER BY P.Departamento;
142 GO
143

```

Results

	Departamento	CantProveedores
1	Arequipa	10
2	Cusco	7
3	Ica	7
4	La Lib	7
5	Lima	18
6	Puno	1

16. Mostrar el número de órdenes por día (sin hora)

```
-- 16. Mostrar el número de órdenes por día (sin hora)
145 SELECT
146     CAST(OC.FechaOrden AS DATE) AS Fecha,
147     COUNT(OC.NumOrden) AS CantidadOrdenes
148     FROM ORDEN_COMPRA OC
149     GROUP BY CAST(OC.FechaOrden AS DATE)
150     ORDER BY CAST(OC.FechaOrden AS DATE);
151 GO
152
```

75 %

Results Messages

	Fecha	CantidadOrdenes
1	2025-09-01	1
2	2025-09-02	1
3	2025-09-03	1
4	2025-09-10	1
5	2025-09-12	1
6	2025-09-13	1
7	2025-09-15	1
8	2025-09-17	1
9	2025-09-18	1
10	2025-09-20	1
11	2025-09-22	1
12	2025-09-23	1
13	2025-10-01	1
14	2025-10-02	1
15	2025-10-03	1
16	2025-10-05	1
17	2025-10-07	1
18	2025-10-08	1
19	2025-10-10	1
20	2025-10-12	1
21	2025-10-13	1
22	2025-10-15	1
23	2025-10-17	1
24	2025-10-18	1
25	2025-10-20	1
26	2025-10-22	1
27	2025-10-23	1
28	2025-10-25	1
29	2025-10-27	1
30	2025-10-28	1
31	2025-11-01	1
32	2025-11-02	1
33	2025-11-05	1
34	2025-11-06	1
35	2025-11-10	1
36	2025-11-11	1
37	2025-11-15	1
38	2025-11-16	1
39	2025-11-20	1
40	2025-11-21	1
41	2025-11-25	1
42	2025-11-26	1
43	2025-12-01	1
44	2025-12-02	1
45	2025-12-05	1
46	2025-12-06	1
47	2025-12-08	1
48	2025-12-09	1
49	2025-12-10	1
50	2025-12-11	1

17. Sumar (CantidadEnviada*PrecioVenta) por CodTienda

```
--> 153 -- 17. Sumar (CantidadEnviada*PrecioVenta) por CodTienda
154 SELECT
155     GE.CodTienda,
156     SUM(CAST(GD.CantidadEnviada * GD.PrecioVenta AS DECIMAL(18,2))) AS TotalVentas
157     FROM GUIA_ENVIO GE
158     INNER JOIN GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
159     GROUP BY GE.CodTienda
160     ORDER BY GE.CodTienda;
161 GO
162
```

75 %

Results Messages

	CodTienda	TotalVentas
1	101	11600.00
2	102	12600.00
3	103	5100.00
4	104	9500.00
5	105	2750.00
6	106	3600.00
7	107	7600.00
8	108	3500.00
9	109	1920.00
10	110	2100.00
11	111	1575.00
12	112	2275.00
13	113	2850.00
14	114	2160.00
15	115	2750.00
16	116	2200.00
17	117	2100.00
18	118	2400.00
19	119	1710.00
20	120	2400.00
21	121	2200.00
22	122	1440.00
23	123	4160.00
24	124	2080.00
25	125	1280.00
26	126	2310.00
27	127	1740.00
28	128	1690.00
29	129	2900.00
30	130	1500.00
31	131	2520.00
32	132	3840.00
33	133	2300.00
34	134	4050.00
35	135	2660.00
36	136	3500.00
37	137	2520.00
38	138	2000.00
39	139	2420.00
40	140	1110.00
41	141	2160.00
42	142	3040.00
43	143	3120.00
44	144	2380.00
45	145	1440.00
46	146	1716.00
47	147	1584.00
48	148	1900.00
49	149	6860.00
50	150	6720.00

18. Mostrar artículos cuyo StockActual <= StockMinimo

```
163  -- 18. Mostrar artículos cuyo StockActual <= StockMinimo
164  SELECT
165      A.CodArticulo,
166      A.DescripcionArticulo,
167      A.StockActual,
168      A.StockMinimo
169  FROM ARTICULO A
170  WHERE A.StockActual <= A.StockMinimo
171  ORDER BY A.CodArticulo;
172  GO
```

75 %

Results Messages

CodArticulo	DescripcionArticulo	StockActual	StockMinimo
-------------	---------------------	-------------	-------------

19. Mostrar CodProveedor, NomProveedor y CantArticulos

```
-- 19. Mostrar CodProveedor, NomProveedor y CantArticulos
175  SELECT
176      P.CodProveedor,
177      P.NomProveedor,
178      COUNT(A.CodArticulo) AS CantArticulos
179  FROM PROVEEDOR P
180  LEFT JOIN ARTICULO A ON P.CodProveedor = A.CodProveedor
181  GROUP BY P.CodProveedor, P.NomProveedor
182  ORDER BY P.CodProveedor;
183  GO
```

75 %

Results Messages

	CodProveedor	NomProveedor	CantArticulos
1	1	Intel LatAm	5
2	2	AMD Distribuidora	5
3	3	NVIDIA Perú	5
4	4	HP Inc Perú	5
5	5	Dell Technologies	5
6	6	Lenovo Distribuidora	5
7	7	Apple Distribuidor	5
8	8	Asus LatAm	5
9	9	MSI Distribution	5
10	10	Gigabyte Perú	5
11	11	Samsung Electronics	0
12	12	LG Electronics	0
13	13	Sony LatAm	0
14	14	Panasonic Perú	0
15	15	Canon Distribuidora	0
16	16	Epson Perú	0
17	17	Corsair Gaming	0
18	18	Logitech LatAm	0
19	19	Razer Inc Dist	0
20	20	SteelSeries Perú	0
21	21	Kingston Technol	0
22	22	Crucial Distribuidor	0
23	23	Western Digital	0
24	24	Seagate Technol	0
25	25	Toshiba Electronics	0
26	26	Transcend Memory	0
27	27	SanDisk Distribuidor	0

28	28	Qualcomm Perú	0
29	29	Broadcom Distrib	0
30	30	Realtek LatAm	0
31	31	Mediatek Distribui	0
32	32	SK Hynix Perú	0
33	33	Micron Technology	0
34	34	Netgear Distribuidor	0
35	35	TP-Link Perú	0
36	36	Ubiquiti Networks	0
37	37	D-Link Distribuidor	0
38	38	Linksys LatAm	0
39	39	ASUS ROG Gaming	0
40	40	Acer Distribuidor	0
41	41	TUF Gaming Dist	0
42	42	Alienware Distrib	0
43	43	ROG Strix Perú	0
44	44	Predator Gaming	0
45	45	IdeaPad Pro	0
46	46	ThinkPad Distrib	0
47	47	XPS Dell Perú	0
48	48	Pavilion HP	0
49	49	Spectre HP Distrib	0
50	50	Envy Collection	0

20. Mostrar para cada Estado sumar CantidadSolicitada

```

185 -- 20. Mostrar para cada Estado sumar CantidadSolicitada
186 SELECT
187     OD.Estado,
188     SUM(OD.CantidadSolicitada) AS TotalSolicitado
189 FROM ORDEN_DETALLE OD
190 GROUP BY OD.Estado
191 ORDER BY OD.Estado;
192 GO
193

```

75 %

	Estado	TotalSolicitado
1	Parcial	137
2	Pendiente	625
3	Recibido	673

Sección III: Cláusula OVER (Funciones de Ventana)

DEFINICIÓN

OVER permite utilizar funciones de agregación sin perder el detalle de cada fila, es decir, muestra el valor calculado y las filas originales.

UTILIDAD

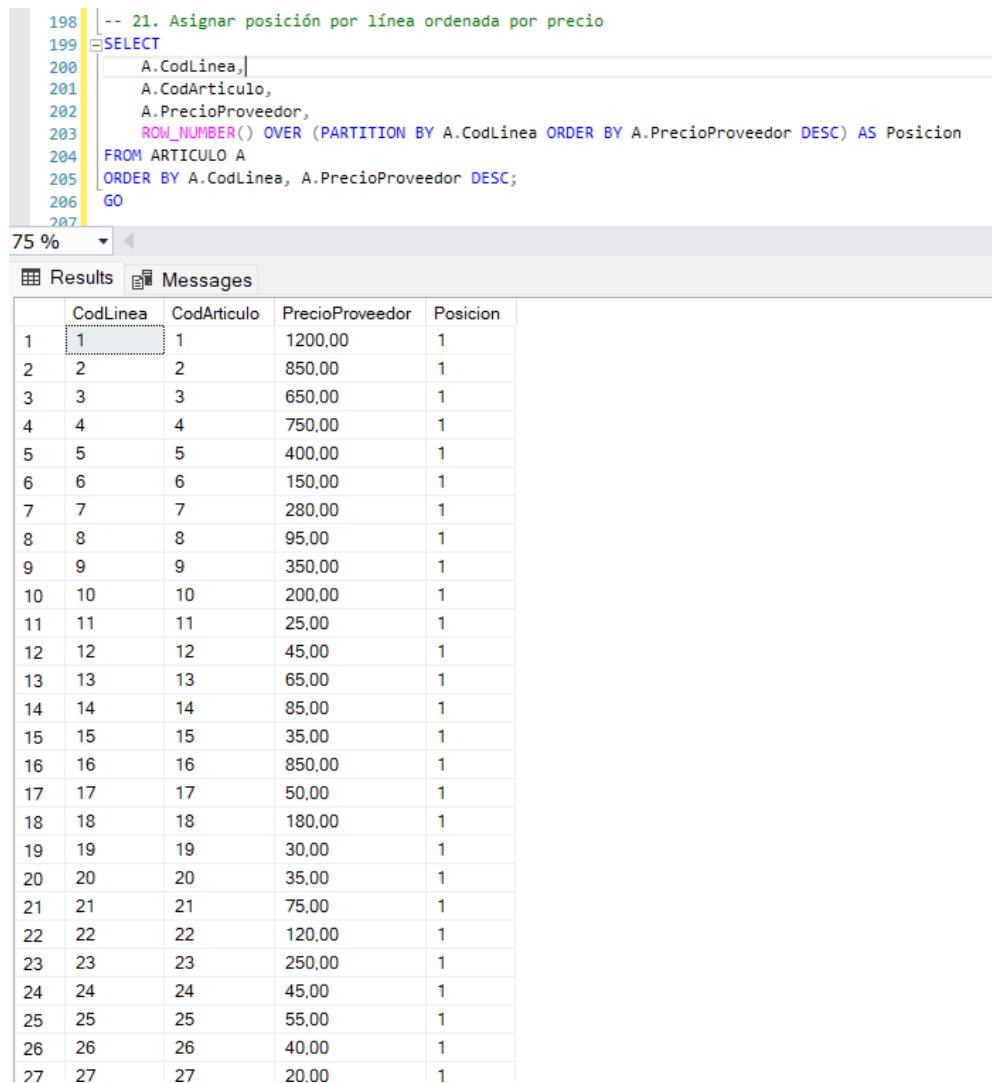
Es útil para estadísticas avanzadas por fila, numeraciones, rankings, acumulados y comparaciones.

APLICACIONES COMUNES:

- ROW_NUMBER, RANK, DENSE_RANK: Numerar filas ordenadas.
- Comparar cada fila con el promedio por categoría.
- Calcular acumulados de ventas por fecha o por tienda.
- Mostrar totales parciales sin agrupar los datos.

EJERCICIOS DE APLICACIÓN

21. Asignar posición por línea ordenada por precio



The screenshot shows a SQL query window and its results. The query is as follows:

```
198  -- 21. Asignar posición por línea ordenada por precio
199  SELECT
200      A.CodLinea,
201      A.CodArticulo,
202      A.PrecioProveedor,
203      ROW_NUMBER() OVER (PARTITION BY A.CodLinea ORDER BY A.PrecioProveedor DESC) AS Posicion
204  FROM ARTICULO A
205  ORDER BY A.CodLinea, A.PrecioProveedor DESC;
206  GO
207
```

The results grid has columns: CodLinea, CodArticulo, PrecioProveedor, and Posicion. The data shows 27 rows, all with a Posicion value of 1, indicating that each line item has a unique position when ordered by price in descending order.

	CodLinea	CodArticulo	PrecioProveedor	Posicion
1	1	1	1200,00	1
2	2	2	850,00	1
3	3	3	650,00	1
4	4	4	750,00	1
5	5	5	400,00	1
6	6	6	150,00	1
7	7	7	280,00	1
8	8	8	95,00	1
9	9	9	350,00	1
10	10	10	200,00	1
11	11	11	25,00	1
12	12	12	45,00	1
13	13	13	65,00	1
14	14	14	85,00	1
15	15	15	35,00	1
16	16	16	850,00	1
17	17	17	50,00	1
18	18	18	180,00	1
19	19	19	30,00	1
20	20	20	35,00	1
21	21	21	75,00	1
22	22	22	120,00	1
23	23	23	250,00	1
24	24	24	45,00	1
25	25	25	55,00	1
26	26	26	40,00	1
27	27	27	20,00	1

28	28	28	95,00	1
29	29	29	450,00	1
30	30	30	380,00	1
31	31	31	65,00	1
32	32	32	180,00	1
33	33	33	75,00	1
34	34	34	200,00	1
35	35	35	50,00	1
36	36	36	380,00	1
37	37	37	320,00	1
38	38	38	380,00	1
39	39	39	160,00	1
40	40	40	280,00	1
41	41	41	180,00	1
42	42	42	60,00	1
43	43	43	90,00	1
44	44	44	25,00	1
45	45	45	20,00	1
46	46	46	15,00	1
47	47	47	12,00	1
48	48	48	35,00	1
49	49	49	50,00	1
50	50	50	100,00	1

22. Calcular costo por orden y su RANK(RankCosto)

```

208 -- 22. Calcular costo por orden y su RANK(SaldoCalidad)
209 SELECT
210     OD.NumOrden,
211     SUM(CAST(OD.PrecioCompra * OD.CantidadSolicitada AS DECIMAL(18,2))) AS CostoTotal,
212     RANK() OVER (ORDER BY SUM(CAST(OD.PrecioCompra * OD.CantidadSolicitada AS DECIMAL(18,2))) DESC) AS RankCosto
213 FROM ORDEN_DETALLE OD
214 GROUP BY OD.NumOrden
215 ORDER BY RankCosto;
216 GO
217

```

75 %

Results Messages

	NumOrden	CostoTotal	RankCosto
1	2025002	12750,00	1
2	2025001	12000,00	2
3	2025004	9000,00	3
4	2025007	7000,00	4
5	2025003	5200,00	5
6	2025049	5000,00	6
7	2025050	5000,00	6
8	2025023	3750,00	8
9	2025032	3240,00	9
10	2025034	3200,00	10
11	2025036	3040,00	11
12	2025006	3000,00	12
13	2025008	2850,00	13
14	2025029	2700,00	14
15	2025016	2550,00	15
16	2025043	2520,00	16
17	2025042	2400,00	17
18	2025005	2400,00	17
19	2025013	2275,00	19
20	2025037	2240,00	20
21	2025018	2160,00	21
22	2025015	2100,00	22
23	2025035	2000,00	23
24	2025031	1950,00	24
25	2025039	1920,00	25
26	2025038	1900,00	26
27	2025033	1875,00	27

28	2025021	1875.00	27
29	2025026	1800.00	29
30	2025012	1800.00	29
31	2025041	1800.00	29
32	2025044	1750.00	32
33	2025009	1750.00	32
34	2025020	1750.00	32
35	2025014	1700.00	35
36	2025010	1600.00	36
37	2025024	1575.00	37
38	2025030	1520.00	38
39	2025017	1500.00	39
40	2025028	1425.00	40
41	2025048	1400.00	41
42	2025011	1250.00	42
43	2025027	1200.00	43
44	2025022	1200.00	43
45	2025019	1200.00	43
46	2025046	1200.00	43
47	2025040	1120.00	47
48	2025047	1080.00	48
49	2025045	1000.00	49
50	2025025	990.00	50

23. Mostrar TotalDia y AcumuladoVentas ordenado por fecha

```
218 -- 23. Mostrar TotalDia y AcumuladoVentas ordenado por fecha
219 SELECT |
SQLQuery1.sql - HP123.QhatuPeru (HP123\USUARIO (67))*
220     SUM(CAST(GD.CantidadEnviada * GD.PrecioVenta AS DECIMAL(18,2))) AS TotalDia,
221     OVER (ORDER BY CAST(GE.FechaSalida AS DATE)) ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS AcumuladoVentas
222 FROM GUIA_ENVIO GE
223 INNER JOIN GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
224 GROUP BY CAST(GE.FechaSalida AS DATE)
225 ORDER BY CAST(GE.FechaSalida AS DATE);
```

	Fecha	TotalDia	AcumuladoVentas
1	2025-10-01	11600.00	11600.00
2	2025-10-02	12600.00	24200.00
3	2025-10-03	5100.00	29300.00
4	2025-10-04	9500.00	38800.00
5	2025-10-05	2750.00	41550.00
6	2025-10-06	3600.00	45150.00
7	2025-10-07	7600.00	52750.00
8	2025-10-08	3500.00	56250.00
9	2025-10-09	1920.00	58170.00
10	2025-10-10	2100.00	60270.00
11	2025-10-11	1575.00	61845.00
12	2025-10-12	2275.00	64120.00
13	2025-10-13	2850.00	66970.00
14	2025-10-14	2160.00	69130.00
15	2025-10-15	2750.00	71880.00
16	2025-10-16	2200.00	74080.00
17	2025-10-17	2100.00	76180.00
18	2025-10-18	2400.00	78580.00
19	2025-10-19	1710.00	80290.00
20	2025-10-20	2400.00	82690.00
21	2025-10-21	2200.00	84890.00
22	2025-10-22	1440.00	86330.00
23	2025-10-23	4160.00	90490.00
24	2025-10-24	2080.00	92570.00
25	2025-10-25	1820.00	93850.00
26	2025-10-26	2310.00	96160.00
27	2025-10-27	1740.00	97900.00

28	2025-10-28	1690.00	99590.00
29	2025-10-29	2900.00	102490.00
30	2025-10-30	1500.00	103990.00
31	2025-10-31	2520.00	106510.00
32	2025-11-01	3840.00	110350.00
33	2025-11-02	2300.00	112650.00
34	2025-11-03	4050.00	116700.00
35	2025-11-04	2660.00	119360.00
36	2025-11-05	3500.00	122860.00
37	2025-11-06	2520.00	125380.00
38	2025-11-07	2000.00	127380.00
39	2025-11-08	2420.00	129800.00
40	2025-11-09	1110.00	130910.00
41	2025-11-10	2160.00	133070.00
42	2025-11-11	3040.00	136110.00
43	2025-11-12	3120.00	139230.00
44	2025-11-13	2380.00	141610.00
45	2025-11-14	1440.00	143050.00
46	2025-11-15	1716.00	144766.00
47	2025-11-16	1584.00	146350.00
48	2025-11-17	1900.00	148250.00
49	2025-11-18	6860.00	155110.00
50	2025-11-19	6720.00	161830.00

24. Calcular precio de orden y su RankSaldoCalidad

```

230 -- 24. Calcular precio de orden y su RankSaldoCalidad
231 SELECT |
232   OD.NumOrden,
233   OD.CodArticulo,
234   CAST(OD.PrecioCompra * OD.CantidadSolicitada AS DECIMAL(18,2)) AS SubTotal,
235   RANK() OVER (PARTITION BY OD.NumOrden ORDER BY OD.PrecioCompra DESC) AS RankPrecio
236   FROM ORDEN_DETALLE OD
237   ORDER BY OD.NumOrden, RankPrecio;
238 GO
239

```

75 %

	NumOrden	CodArticulo	SubTotal	RankPrecio
1	2025001	1	12000.00	1
2	2025002	2	12750.00	1
3	2025003	3	5200.00	1
4	2025004	4	9000.00	1
5	2025005	5	2400.00	1
6	2025006	6	3000.00	1
7	2025007	7	7000.00	1
8	2025008	8	2850.00	1
9	2025009	9	1750.00	1
10	2025010	10	1600.00	1
11	2025011	11	1250.00	1
12	2025012	12	1800.00	1
13	2025013	13	2275.00	1
14	2025014	14	1700.00	1
15	2025015	15	2100.00	1
16	2025016	16	2550.00	1
17	2025017	17	1500.00	1
18	2025018	18	2160.00	1
19	2025019	19	1200.00	1
20	2025020	20	1750.00	1
21	2025021	21	1875.00	1
22	2025022	22	1200.00	1
23	2025023	23	3750.00	1
24	2025024	24	1575.00	1
25	2025025	25	990.00	1
26	2025026	26	1800.00	1
27	2025027	27	1200.00	1

28	2025028	28	1425.00	1
29	2025029	29	2700.00	1
30	2025030	30	1520.00	1
31	2025031	31	1950.00	1
32	2025032	32	3240.00	1
33	2025033	33	1875.00	1
34	2025034	34	3200.00	1
35	2025035	35	2000.00	1
36	2025036	36	3040.00	1
37	2025037	37	2240.00	1
38	2025038	38	1900.00	1
39	2025039	39	1920.00	1
40	2025040	40	1120.00	1
41	2025041	41	1800.00	1
42	2025042	42	2400.00	1
43	2025043	43	2520.00	1
44	2025044	44	1750.00	1
45	2025045	45	1000.00	1
46	2025046	46	1200.00	1
47	2025047	47	1080.00	1
48	2025048	48	1400.00	1
49	2025049	49	5000.00	1
50	2025050	50	5000.00	1

25. Mostrar PrecioAnteriorMismoProveedor usando LAG

```

240 -- 25. Mostrar PrecioAnteriorMismoProveedor usando LAG
241 SELECT
242     A.CodProveedor,
243     A.CodArticulo,
244     A.PrecioProveedor,
245     LAG(A.PrecioProveedor) OVER (PARTITION BY A.CodProveedor ORDER BY A.CodArticulo) AS PrecioAnterior
246     FROM ARTICULO A
247     ORDER BY A.CodProveedor, A.CodArticulo;
248 GO

```

Results Messages

	CodProveedor	CodArticulo	PrecioProveedor	PrecioAnterior
1	1	1	1200.00	NULL
2	1	11	25.00	1200.00
3	1	21	75.00	25.00
4	1	31	65.00	75.00
5	1	41	180.00	65.00
6	2	2	850.00	NULL
7	2	12	45.00	850.00
8	2	22	120.00	45.00
9	2	32	180.00	120.00
10	2	42	60.00	180.00
11	3	3	650.00	NULL
12	3	13	65.00	650.00
13	3	23	250.00	65.00
14	3	33	75.00	250.00
15	3	43	90.00	75.00
16	4	4	750.00	NULL
17	4	14	85.00	750.00
18	4	24	45.00	85.00
19	4	34	200.00	45.00
20	4	44	25.00	200.00
21	5	5	400.00	NULL
22	5	15	35.00	400.00
23	5	25	55.00	35.00
24	5	35	50.00	55.00
25	5	45	20.00	50.00
26	6	6	150.00	NULL
27	6	16	850.00	150.00

28	6	26	40,00	850,00
29	6	36	380,00	40,00
30	6	46	15,00	380,00
31	7	7	280,00	NULL
32	7	17	50,00	280,00
33	7	27	20,00	50,00
34	7	37	320,00	20,00
35	7	47	12,00	320,00
36	8	8	95,00	NULL
37	8	18	180,00	95,00
38	8	28	95,00	180,00
39	8	38	380,00	95,00
40	8	48	35,00	380,00
41	9	9	350,00	NULL
42	9	19	30,00	350,00
43	9	29	450,00	30,00
44	9	39	160,00	450,00
45	9	49	50,00	160,00
46	10	10	200,00	NULL
47	10	20	35,00	200,00
48	10	30	380,00	35,00
49	10	40	280,00	380,00
50	10	50	100,00	280,00

26. Añadir columna CantidadPorLinea a cada artículo

```

250 -- 26. Añadir columna CantidadPorLinea a cada artículo
251 SELECT |
252     A.CodLinea,
253     A.CodArticulo,
254     A.DescripcionArticulo,
255     SUM(A.StockActual) OVER (PARTITION BY A.CodLinea) AS CantidadPorLinea
256     FROM ARTICULO A
257     ORDER BY A.CodLinea, A.CodArticulo;
258     GO

```

75 %

Results Messages

	CodLinea	CodArticulo	DescripcionArticulo	CantidadPorLinea
1	1	1	Desktop i7 16GB	15
2	2	2	Laptop i5 8GB 256GB	20
3	3	3	iPad Air 64GB	12
4	4	4	Samsung Galaxy S23	25
5	5	5	Monitor LG 27 4K	18
6	6	6	Teclado Mecánico RGB	45
7	7	7	AirPods Pro	30
8	8	8	Logitech C920 Webcam	35
9	9	9	Impresora HP LaserJet	8
10	10	10	Scanner ScanJet Pro	12
11	11	11	Cable HDMI 2.1	100
12	12	12	Cargador Rápido 65W	80
13	13	13	Power Bank 25000mAh	55
14	14	14	Disco Duro Externo 1TB	42
15	15	15	Tarjeta MicroSD 256GB	120
16	16	16	Monitor Digital 55	5
17	17	17	Ventilador Noctua	60
18	18	18	Fuente EVGA 850W	22
19	19	19	Cable Cat6 20mts	75
20	20	20	Funda Laptop 15.6	90
21	21	21	Mochila Gaming	40
22	22	22	Stand Monitor Doble	18
23	23	23	Docking Station USB-C	25
24	24	24	Hub USB 7 Puertos	65
25	25	25	Base Refriger Laptop	35
26	26	26	Mouse Pad Grande RGB	80
27	27	27	Protector Pantalla	100

28	28	28	Lentes Gaming	28
29	29	29	Silla Gamer Ergon	10
30	30	30	Escritorio Gaming	8
31	31	31	Controlador Xbox	55
32	32	32	Headset Gaming Inal	32
33	33	33	Ratón Gaming Inal	48
34	34	34	Teclado Mecánico G	30
35	35	35	Alfombrilla Gaming	70
36	36	36	Monitor 144Hz 27	14
37	37	37	Tarjeta RTX 4060	12
38	38	38	Procesador Ryzen 7	10
39	39	39	Memoria RAM 32GB	20
40	40	40	Placa Madre ASUS ROG	8
41	41	41	Case Gaming Tempered	18
42	42	42	Iluminación RGB LED	75
43	43	43	Ventilador RGB Pack	50
44	44	44	Tubos Sleeving Color	110
45	45	45	Pasta Térmica Prem	95
46	46	46	Aire Comprimido 420	130
47	47	47	Pulsera Antiestática	160
48	48	48	Kit Herramientas	70
49	49	49	Manual Garantía Ext	200
50	50	50	Servicio Instalación	100

27. Mostrar MontoProveedor y PorcentajeDelTotal

```

260 -- 27. Mostrar MontoProveedor y PorcentajeDelTotal
261 SELECT
262     P.CodProveedor,
263     P.NomProveedor,
264     SUM(CAST(A.StockActual * A.PrecioProveedor AS DECIMAL(18,2))) AS MontoProveedor,
265     CAST(SUM(CAST(A.StockActual * A.PrecioProveedor AS DECIMAL(18,2))) * 100.0 /
266         SUM(SUM(CAST(A.StockActual * A.PrecioProveedor AS DECIMAL(18,2)))) OVER () AS DECIMAL(10,2)) AS PorcentajeDelTotal
267 FROM PROVEEDOR P
268 LEFT JOIN ARTICULO A ON P.CodProveedor = A.CodProveedor
269 GROUP BY P.CodProveedor, P.NomProveedor
270 ORDER BY MontoProveedor DESC;
271 GO

```

75 %

	CodProveedor	NomProveedor	MontoProveedor	PorcentajeDelTotal
1	4	HP Inc Perú	33995.00	14.04
2	2	AMD Distribuidora	33020.00	13.63
3	1	Intel LatAm	30315.00	12.52
4	3	NVIDIA Perú	25725.00	10.62
5	9	MSI Distribution	22750.00	9.39
6	6	Lenovo Distribuidora	21470.00	8.87
7	10	Gigabyte Perú	20830.00	8.60
8	7	Apple Distribuidor	19160.00	7.91
9	5	Dell Technologies	18725.00	7.73
10	8	Asus LatAm	16195.00	6.69
11	11	Samsung Electronics	NULL	NULL
12	12	LG Electronics	NULL	NULL
13	13	Sony LatAm	NULL	NULL
14	14	Panasonic Perú	NULL	NULL
15	15	Canon Distribuidora	NULL	NULL
16	16	Epson Perú	NULL	NULL
17	17	Corsair Gaming	NULL	NULL
18	18	Logitech LatAm	NULL	NULL
19	19	Razer Inc Dist	NULL	NULL
20	20	SteelSeries Perú	NULL	NULL
21	21	Kingston Technol	NULL	NULL
22	22	Crucial Distribuidor	NULL	NULL
23	23	Western Digital	NULL	NULL
24	24	Seagate Technol	NULL	NULL
25	25	Toshiba Electronics	NULL	NULL

26	26	Transcend Memory	NULL	NULL
27	27	SanDisk Distribuidor	NULL	NULL
28	28	Qualcomm Perú	NULL	NULL
29	29	Broadcom Distrib	NULL	NULL
30	30	Realtek LatAm	NULL	NULL
31	31	Mediatek Distribui	NULL	NULL
32	32	SK Hynix Perú	NULL	NULL
33	33	Micron Technology	NULL	NULL
34	34	Netgear Distribuidor	NULL	NULL
35	35	TP-Link Perú	NULL	NULL
36	36	Ubiquiti Networks	NULL	NULL
37	37	D-Link Distribuidor	NULL	NULL
38	38	Linksys LatAm	NULL	NULL
39	39	ASUS ROG Gaming	NULL	NULL
40	40	Acer Distribuidor	NULL	NULL
41	41	TUF Gaming Dist	NULL	NULL
42	42	Alienware Distrib	NULL	NULL
43	43	ROG Strix Perú	NULL	NULL
44	44	Predator Gaming	NULL	NULL
45	45	IdeaPad Pro	NULL	NULL
46	46	ThinkPad Distrib	NULL	NULL
47	47	XPS Dell Perú	NULL	NULL
48	48	Pavilion HP	NULL	NULL
49	49	Spectre HP Distrib	NULL	NULL
50	50	Envy Collection	NULL	NULL

28. Mostrar solo los 3 artículos más caros por línea

```

273 -- 28. Mostrar solo los 3 artículos más caros por línea
274 SELECT *
275 FROM (
276   SELECT
277     A.CodLinea,
278     A.CodArticulo,
279     A.DescripcionArticulo,
280     A.PrecioProveedor,
281     ROW_NUMBER() OVER (PARTITION BY A.CodLinea ORDER BY A.PrecioProveedor DESC) AS RangoLinea
282   FROM ARTICULO A
283 ) AS X
284 WHERE X.RangoLinea <= 3
285 ORDER BY X.CodLinea, X.PrecioProveedor DESC;
286

```

75 %

Results Messages

	CodLinea	CodArticulo	DescripcionArticulo	PrecioProveedor	RangoLinea
1	1	1	Desktop i7 16GB	1200,00	1
2	2	2	Laptop i5 8GB 256GB	850,00	1
3	3	3	iPad Air 64GB	650,00	1
4	4	4	Samsung Galaxy S23	750,00	1
5	5	5	Monitor LG 27 4K	400,00	1
6	6	6	Teclado Mecánico RGB	150,00	1
7	7	7	AirPods Pro	280,00	1
8	8	8	Logitech C920 Webcam	95,00	1
9	9	9	Impresora HP LaserJet	350,00	1
10	10	10	Scanner ScanJet Pro	200,00	1
11	11	11	Cable HDMI 2.1	25,00	1
12	12	12	Cargador Rápido 65W	45,00	1
13	13	13	Power Bank 25000mAh	65,00	1
14	14	14	Disco Duro Externo 1TB	85,00	1
15	15	15	Tarjeta MicroSD 256GB	35,00	1
16	16	16	Monitor Digital 55	850,00	1
17	17	17	Ventilador Noctua	50,00	1
18	18	18	Fuente EVGA 850W	180,00	1
19	19	19	Cable Cat6 20mts	30,00	1
20	20	20	Funda Laptop 15.6	35,00	1
21	21	21	Mochila Gaming	75,00	1
22	22	22	Stand Monitor Doble	120,00	1
23	23	23	Docking Station USB-C	250,00	1
24	24	24	Hub USB 7 Puertos	45,00	1
25	25	25	Base Refriger Laptop	55,00	1

26	26	26	Mouse Pad Grande RGB	40,00	1
27	27	27	Protector Pantalla	20,00	1
28	28	28	Lentes Gaming	95,00	1
29	29	29	Silla Gamer Ergon	450,00	1
30	30	30	Escritorio Gaming	380,00	1
31	31	31	Controlador Xbox	65,00	1
32	32	32	Headset Gaming Inal	180,00	1
33	33	33	Ratón Gaming Inal	75,00	1
34	34	34	Teclado Mecánico G	200,00	1
35	35	35	Alfombrilla Gaming	50,00	1
36	36	36	Monitor 144Hz 27	380,00	1
37	37	37	Tarjeta RTX 4060	320,00	1
38	38	38	Procesador Ryzen 7	380,00	1
39	39	39	Memoria RAM 32GB	160,00	1
40	40	40	Placa Madre ASUS ROG	280,00	1
41	41	41	Case Gaming Tempered	180,00	1
42	42	42	Iluminación RGB LED	60,00	1
43	43	43	Ventilador RGB Pack	90,00	1
44	44	44	Tubos Sleeving Color	25,00	1
45	45	45	Pasta Térmica Prem	20,00	1
46	46	46	Aire Comprimido 420	15,00	1
47	47	47	Pulsera Antiestática	12,00	1
48	48	48	Kit Herramientas	35,00	1
49	49	49	Manual Garantía Ext	50,00	1
50	50	50	Servicio Instalación	100,00	1

29. Mostrar transportista y su DenseRank por TotalEnviado

```

287 -- 29. Mostrar transportista y su DenseRank por TotalEnviado
288 SELECT
289   T.CodTransportista,
290   T.NomTransportista,
291   SUM(GD.CantidadEnviada) AS TotalEnviado,
292   DENSE_RANK() OVER (ORDER BY SUM(GD.CantidadEnviada) DESC) AS DenseRankEnvios
293 FROM TRANSPORTISTA T
294 INNER JOIN GUIA_ENVIO GE ON T.CodTransportista = GE.CodTransportista
295 INNER JOIN GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
296 GROUP BY T.CodTransportista, T.NomTransportista
297 ORDER BY DenseRankEnvios;
298 GO
299

```

75 %

Results Messages

	CodTransportista	NomTransportista	TotalEnviado	DenseRankEnvios
1	549	ReliableLog	98	1
2	547	OnTimeDeliv	88	2
3	546	RealTime T	78	3
4	544	QuickDelivery	68	4
5	527	WarehousingT	58	5
6	515	CyberTransport	55	6
7	520	TechFreight S	48	7
8	545	InstantXpress	48	7
9	550	TrustTransp	48	7
10	511	ComputerShip	45	8
11	526	TechDistrib	42	9
12	535	RetailXpress	38	10
13	548	PunctualXpr	38	10
14	542	FastTrack D	38	10
15	519	DeviceLogistics	38	10
16	512	DigitalTransit	35	11
17	524	ComponentsXpr	32	12
18	513	ServerXpress	30	13
19	517	ElectroCart	28	14
20	531	TradeRoute E	28	14
21	543	SpeedWay Log	26	15
22	508	GadgetRutas SRL	25	16
23	533	MarketTransp	23	17
24	521	RapidPC Del	22	18
25	507	Hardware Express	20	19

26	514	DataFlow Log	18	20
27	506	CompuCourier	18	20
28	525	PartsDelivery	16	21
29	532	CommerceTechL	16	21
30	534	SalesDelivery	15	22
31	528	InventoryXpr	13	23
32	523	GearTransport	13	23
33	502	ElectroRutas SAC	12	24
34	539	CenterXpress	11	25
35	504	DeviceXpress	10	26
36	518	GadgetShip G	10	26
37	522	ElectroSpeed	9	27
38	541	DirectXpress	9	27
39	501	TechLog Express	8	28
40	510	TechLogistica	7	29
41	536	StoreLogistics	7	29
42	537	ShopTransport	6	30
43	503	FastTech Delivery	6	30
44	505	TechTransport A	5	31
45	529	StockDelivery	5	31
46	509	ElectroDelivery	4	32
47	538	MallDelivery	4	32
48	540	PlazaTech T	3	33
49	530	SupplychainT	3	33
50	516	TechCarrier P	2	34

30. Mostrar por guía la suma acumulada por tienda hasta esa guía

```

300 -- 30. Mostrar por guía la suma acumulada por tienda hasta esa guía
301 SELECT
302     GE.NumGuia,
303     GE.CodTienda,
304     SUM(CAST(GD.CantidadEnviada * GD.PrecioVenta AS DECIMAL(18,2))) AS VentasGuia,
305     SUM(SUM(CAST(GD.CantidadEnviada * GD.PrecioVenta AS DECIMAL(18,2))))
306         OVER (PARTITION BY GE.CodTienda ORDER BY GE.NumGuia ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS AcumuladoPorTienda
307 FROM GUIA_ENVIO GE
308 INNER JOIN GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
309 GROUP BY GE.NumGuia, GE.CodTienda
310 ORDER BY GE.CodTienda, GE.NumGuia;
311 GO
312

```

Results

	NumGuia	CodTienda	VentasGuia	AcumuladoPorTienda
1	1001	101	11600.00	11600.00
2	1002	102	12600.00	12600.00
3	1003	103	5100.00	5100.00
4	1004	104	9500.00	9500.00
5	1005	105	2750.00	2750.00
6	1006	106	3600.00	3600.00
7	1007	107	7600.00	7600.00
8	1008	108	3500.00	3500.00
9	1009	109	1920.00	1920.00
10	1010	110	2100.00	2100.00
11	1011	111	1575.00	1575.00
12	1012	112	2275.00	2275.00
13	1013	113	2850.00	2850.00
14	1014	114	2160.00	2160.00
15	1015	115	2750.00	2750.00
16	1016	116	2200.00	2200.00
17	1017	117	2100.00	2100.00
18	1018	118	2400.00	2400.00
19	1019	119	1710.00	1710.00
20	1020	120	2400.00	2400.00
21	1021	121	2200.00	2200.00
22	1022	122	1440.00	1440.00
23	1023	123	4160.00	4160.00
24	1024	124	2080.00	2080.00
25	1025	125	1280.00	1280.00

26	1026	126	2310.00	2310.00
27	1027	127	1740.00	1740.00
28	1028	128	1690.00	1690.00
29	1029	129	2900.00	2900.00
30	1030	130	1500.00	1500.00
31	1031	131	2520.00	2520.00
32	1032	132	3840.00	3840.00
33	1033	133	2300.00	2300.00
34	1034	134	4050.00	4050.00
35	1035	135	2660.00	2660.00
36	1036	136	3500.00	3500.00
37	1037	137	2520.00	2520.00
38	1038	138	2000.00	2000.00
39	1039	139	2420.00	2420.00
40	1040	140	1110.00	1110.00
41	1041	141	2160.00	2160.00
42	1042	142	3040.00	3040.00
43	1043	143	3120.00	3120.00
44	1044	144	2380.00	2380.00
45	1045	145	1440.00	1440.00
46	1046	146	1716.00	1716.00
47	1047	147	1584.00	1584.00
48	1048	148	1900.00	1900.00
49	1049	149	6860.00	6860.00
50	1050	150	6720.00	6720.00

SECCIÓN IV: OPERADOR PIVOT

DEFINICIÓN

Es una operación que convierte filas en columnas, creando una tabla cruzada o dinámica.

UTILIDAD

Se usa para reordenar datos y facilitar el análisis visual, especialmente cuando se quiere comparar valores entre categorías o fechas.

EJEMPLOS DE USO:

- Mostrar ventas por tienda en columnas.
- Comparar cantidades de artículos por transportista.
- Crear una tabla de estado de pedidos por fecha.
- Cruces en múltiples dimensiones para análisis avanzados.

EJERCICIOS DE APLICACIÓN

31. Mostrar Fechas y columnas CodTienda_1, CodTienda_2, con TotalEnviado por día

```

317 | -- 31. Mostrar Fechas y columnas CodTienda_1, CodTienda_2, con TotalEnviado por día
318 | SELECT
319 |     [1001] AS Tienda_1001,
320 |     [1002] AS Tienda_1002,
321 |     [1003] AS Tienda_1003,
322 |     [1004] AS Tienda_1004,
323 |     [1005] AS Tienda_1005
324 | FROM (
325 |     SELECT
326 |         CAST(GE.FechaSalida AS DATE) AS Fecha,
327 |         GE.NumGuia,
328 |         SUM(GD.CantidadEnviada) AS TotalEnviado
329 |     FROM GUIA_ENVIO GE
330 |     INNER JOIN GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
331 |     GROUP BY CAST(GE.FechaSalida AS DATE), GE.NumGuia
332 | ) AS datos
333 | PIVOT (
334 |     SUM(TotalEnviado) FOR NumGuia IN ([1001], [1002], [1003], [1004], [1005])
335 | ) AS tabla_pivot
336 | ORDER BY Fecha;
337 | GO

```

68 %

Results Messages

	Tienda_1001	Tienda_1002	Tienda_1003	Tienda_1004	Tienda_1005
1	8	NULL	NULL	NULL	NULL
2	NULL	12	NULL	NULL	NULL
3	NULL	NULL	6	NULL	NULL
4	NULL	NULL	NULL	10	NULL
5	NULL	NULL	NULL	NULL	5
6	NULL	NULL	NULL	NULL	NULL
7	NULL	NULL	NULL	NULL	NULL
8	NULL	NULL	NULL	NULL	NULL
9	NULL	NULL	NULL	NULL	NULL
10	NULL	NULL	NULL	NULL	NULL
11	NULL	NULL	NULL	NULL	NULL
12	NULL	NULL	NULL	NULL	NULL
13	NULL	NULL	NULL	NULL	NULL
14	NULL	NULL	NULL	NULL	NULL
15	NULL	NULL	NULL	NULL	NULL
16	NULL	NULL	NULL	NULL	NULL
17	NULL	NULL	NULL	NULL	NULL
18	NULL	NULL	NULL	NULL	NULL
19	NULL	NULL	NULL	NULL	NULL
20	NULL	NULL	NULL	NULL	NULL
21	NULL	NULL	NULL	NULL	NULL
22	NULL	NULL	NULL	NULL	NULL
23	NULL	NULL	NULL	NULL	NULL
24	NULL	NULL	NULL	NULL	NULL
25	NULL	NULL	NULL	NULL	NULL
26	NULL	NULL	NULL	NULL	NULL
27	NULL	NULL	NULL	NULL	NULL
28	NULL	NULL	NULL	NULL	NULL
29	NULL	NULL	NULL	NULL	NULL
30	NULL	NULL	NULL	NULL	NULL
31	NULL	NULL	NULL	NULL	NULL
32	NULL	NULL	NULL	NULL	NULL
33	NULL	NULL	NULL	NULL	NULL
34	NULL	NULL	NULL	NULL	NULL
35	NULL	NULL	NULL	NULL	NULL
36	NULL	NULL	NULL	NULL	NULL
37	NULL	NULL	NULL	NULL	NULL
38	NULL	NULL	NULL	NULL	NULL
39	NULL	NULL	NULL	NULL	NULL
40	NULL	NULL	NULL	NULL	NULL
41	NULL	NULL	NULL	NULL	NULL
42	NULL	NULL	NULL	NULL	NULL
43	NULL	NULL	NULL	NULL	NULL
44	NULL	NULL	NULL	NULL	NULL
45	NULL	NULL	NULL	NULL	NULL
46	NULL	NULL	NULL	NULL	NULL
47	NULL	NULL	NULL	NULL	NULL
48	NULL	NULL	NULL	NULL	NULL
49	NULL	NULL	NULL	NULL	NULL
50	NULL	NULL	NULL	NULL	NULL

32. Mostrar CodArticulo y columnas con cantidades por tienda 1..3

```
339 -- 32. Mostrar CodArticulo y columnas con cantidades por tienda 1..3
340 SELECT
341     CodArticulo,
342     [101] AS Tienda_101,
343     [102] AS Tienda_102,
344     [103] AS Tienda_103
345     FROM (
346         SELECT
347             GD.CodArticulo,
348             GE.CodTienda,
349             GD.CantidadEnviada
350             FROM GUIA_ENVIO GE
351             INNER JOIN GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
352     ) AS datos
353     PIVOT (
354         SUM(CantidadEnviada) FOR CodTienda IN ([101], [102], [103])
355     ) AS tabla_pivot
356     ORDER BY CodArticulo;
357     GO
```

68 %

Results Messages

	CodArticulo	Tienda_101	Tienda_102	Tienda_103
1	1	8	NULL	NULL
2	2	NULL	12	NULL
3	3	NULL	NULL	6
4	4	NULL	NULL	NULL
5	5	NULL	NULL	NULL
6	6	NULL	NULL	NULL
7	7	NULL	NULL	NULL
8	8	NULL	NULL	NULL
9	9	NULL	NULL	NULL
10	10	NULL	NULL	NULL
11	11	NULL	NULL	NULL
12	12	NULL	NULL	NULL
13	13	NULL	NULL	NULL
14	14	NULL	NULL	NULL
15	15	NULL	NULL	NULL
16	16	NULL	NULL	NULL
17	17	NULL	NULL	NULL
18	18	NULL	NULL	NULL
19	19	NULL	NULL	NULL
20	20	NULL	NULL	NULL
21	21	NULL	NULL	NULL
22	22	NULL	NULL	NULL
23	23	NULL	NULL	NULL
24	24	NULL	NULL	NULL
25	25	NULL	NULL	NULL
26	26	NULL	NULL	NULL
27	27	NULL	NULL	NULL
28	28	NULL	NULL	NULL
29	29	NULL	NULL	NULL
30	30	NULL	NULL	NULL
31	31	NULL	NULL	NULL
32	32	NULL	NULL	NULL
33	33	NULL	NULL	NULL
34	34	NULL	NULL	NULL
35	35	NULL	NULL	NULL
36	36	NULL	NULL	NULL
37	37	NULL	NULL	NULL
38	38	NULL	NULL	NULL
39	39	NULL	NULL	NULL
40	40	NULL	NULL	NULL
41	41	NULL	NULL	NULL
42	42	NULL	NULL	NULL
43	43	NULL	NULL	NULL
44	44	NULL	NULL	NULL
45	45	NULL	NULL	NULL
46	46	NULL	NULL	NULL
47	47	NULL	NULL	NULL
48	48	NULL	NULL	NULL
49	49	NULL	NULL	NULL
50	50	NULL	NULL	NULL

33. Mostrar AñoMes y tiendas como columnas con suma de PrecioVenta*Cantidad

```

359 -- 33. Mostrar AñoMes y tiendas como columnas con suma de PrecioVentaCantidad
360 SELECT
361     [101] AS Tienda_101,
362     [102] AS Tienda_102,
363     [103] AS Tienda_103,
364     [104] AS Tienda_104,
365     [105] AS Tienda_105
366 FROM (
367     SELECT
368         YEAR(GE.FechaSalida) * 100 + MONTH(GE.FechaSalida) AS AñoMes,
369         GE.CodTienda,
370         SUM(CAST(GD.CantidadEnviada * GD.PrecioVenta AS DECIMAL(18,2))) AS TotalVentas
371     FROM GUIA_ENVIO GE
372     INNER JOIN GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
373     GROUP BY YEAR(GE.FechaSalida) * 100 + MONTH(GE.FechaSalida), GE.CodTienda
374 ) AS datos
375 PIVOT (
376     SUM(TotalVentas) FOR CodTienda IN ([101], [102], [103], [104], [105])
377 ) AS tabla_pivot
378 ORDER BY AñoMes;
379 GO
380

```

Results Messages

	Tienda_101	Tienda_102	Tienda_103	Tienda_104	Tienda_105
1	11600.00	12600.00	5100.00	9500.00	2750.00
2	NULL	NULL	NULL	NULL	NULL

34. Mostrar CodArticulo con columnas para cada Estado

```

381 -- 34. Mostrar CodArticulo con columnas para cada Estado
382 SELECT
383     CodArticulo,
384     [Recibido] AS Recibido,
385     [Pendiente] AS Pendiente,
386     [Parcial] AS Parcial
387 FROM (
388     SELECT
389         OD.CodArticulo,
390         OD.Estado,
391         OD.CantidadRecibida
392     FROM ORDEN_DETALLE OD
393 ) AS datos
394 PIVOT (
395     SUM(CantidadRecibida) FOR Estado IN ([Recibido], [Pendiente], [Parcial])
396 ) AS tabla_pivot
397 ORDER BY CodArticulo;
398 GO
399

```

Results Messages

	CodArticulo	Recibido	Pendiente	Parcial
1	1	10	NULL	NULL
2	2	NULL	0	NULL
3	3	8	NULL	NULL
4	4	12	NULL	NULL
5	5	NULL	0	NULL
6	6	20	NULL	NULL
7	7	NULL	0	NULL
8	8	NULL	NULL	20
9	9	5	NULL	NULL
10	10	NULL	0	NULL
11	11	50	NULL	NULL
12	12	NULL	0	NULL
13	13	35	NULL	NULL
14	14	20	NULL	NULL
15	15	NULL	0	NULL
16	16	3	NULL	NULL
17	17	NULL	0	NULL
18	18	12	NULL	NULL
19	19	NULL	NULL	35
20	20	NULL	0	NULL

21	21	25	NULL	NULL
22	22	NULL	0	NULL
23	23	15	NULL	NULL
24	24	35	NULL	NULL
25	25	NULL	0	NULL
26	26	45	NULL	NULL
27	27	NULL	0	NULL
28	28	NULL	NULL	10
29	29	6	NULL	NULL
30	30	NULL	0	NULL
31	31	30	NULL	NULL
32	32	NULL	0	NULL
33	33	25	NULL	NULL
34	34	16	NULL	NULL
35	35	NULL	0	NULL
36	36	8	NULL	NULL
37	37	NULL	0	NULL
38	38	5	NULL	NULL
39	39	NULL	NULL	10
40	40	NULL	0	NULL
41	41	10	NULL	NULL
42	42	NULL	0	NULL
43	43	28	NULL	NULL
44	44	70	NULL	NULL
45	45	NULL	0	NULL
46	46	80	NULL	NULL
47	47	NULL	0	NULL
48	48	NULL	NULL	35
49	49	100	NULL	NULL
50	50	NULL	0	NULL

35. Contar artículos por presentación pivotada

```
400 -- 35. Contar artículos por presentación pivotada
401
402 SELECT
403     [Caja] AS Caja,
404     [Blister] AS Blister,
405     [Carrete] AS Carrete,
406     [Unidad] AS Unidad,
407     [Paquete] AS Paquete,
408     [Tubo] AS Tubo,
409     [Lata] AS Lata,
410     [Documento] AS Documento,
411     [Servicio] AS Servicio
412 FROM (
413     SELECT
414         CodLinea,
415         Presentacion,
416         COUNT(*) AS CantArticulos
417     FROM ARTICULO
418     GROUP BY CodLinea, Presentacion
419 ) AS datos
420 PIVOT (
421     SUM(CantArticulos) FOR Presentacion IN ([Caja], [Blister], [Carrete], [Unidad], [Paquete], [Tubo], [Lata], [Documento], [Servicio])
422 ) AS tabla_pivot;
423 GO
```

36. Generar PIVOT dinámico para todas las tiendas (ejemplo de patrón)

```
424 -- 36. Generar PIVOT dinámico para todas las tiendas (ejemplo de patrón)
425 SELECT * FROM (
426     SELECT
427         A.CodLinea,
428         GE.CodTienda,
429         SUM(GD.CantidadEnviada) AS CantidadEnviada
430     FROM ARTICULO A
431     INNER JOIN GUIA_DETALLE GD ON A.CodArticulo = GD.CodArticulo
432     INNER JOIN GUIA_ENVIO GE ON GD.NumGuia = GE.NumGuia
433     GROUP BY A.CodLinea, GE.CodTienda
434 ) AS datos
435 PIVOT (
436     SUM(CantidadEnviada) FOR CodTienda IN ([101], [102], [103], [104], [105], [106], [107], [108], [109], [110])
437 ) AS tabla_pivot
438 ORDER BY CodLinea;
439 GO
```

23	23	NULL										
24	24	NULL										
25	25	NULL										
26	26	NULL										
27	27	NULL										
28	28	NULL										
29	29	NULL										
30	30	NULL										
31	31	NULL										
32	32	NULL										
33	33	NULL										
34	34	NULL										
35	35	NULL										
36	36	NULL										
37	37	NULL										
38	38	NULL										
39	39	NULL										
40	40	NULL										
41	41	NULL										
42	42	NULL										
43	43	NULL										
44	44	NULL										
45	45	NULL										
46	46	NULL										
47	47	NULL										
48	48	NULL										
49	49	NULL										
50	50	NULL										

37. Mostrar mes y columnas por transportistas con totales

```

441 -- 37. Mostrar mes y columnas por transportistas con totales
442 SELECT
443     [501] AS Transportista_501,
444     [502] AS Transportista_502,
445     [503] AS Transportista_503
446 FROM (
447     SELECT
448         YEAR(GE.FechaSalida) * 100 + MONTH(GE.FechaSalida) AS AñoMes,
449         GE.CodTransportista,
450         SUM(GD.CantidadEnviada) AS TotalEnviado
451     FROM GUIA_ENVIO GE
452     INNER JOIN GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
453     GROUP BY YEAR(GE.FechaSalida) * 100 + MONTH(GE.FechaSalida), GE.CodTransportista
454 ) AS datos
455 PIVOT (
456     SUM(TotalEnviado) FOR CodTransportista IN ([501], [502], [503])
457 ) AS tabla_pivot
458 ORDER BY AñoMes;
459 GO

```

75 %

Results Messages

	Transportista_501	Transportista_502	Transportista_503
1	8	12	6
2	NULL	NULL	NULL

38. Contar proveedores por rango de variedad de artículos pivotado como columnas

```

461 -- 38. Contar proveedores por rango de variedad de artículos pivotado
462 SELECT
463     [1-5] AS Rango_1_5,
464     [6-10] AS Rango_6_10,
465     [11-20] AS Rango_11_20
466 FROM (
467     SELECT
468         P.CodProveedor,
469         CASE
470             WHEN COUNT(A.CodArticulo) BETWEEN 1 AND 5 THEN '1-5'
471             WHEN COUNT(A.CodArticulo) BETWEEN 6 AND 10 THEN '6-10'
472             ELSE '11-20'
473         END AS RangoArticulos,
474         COUNT(A.CodArticulo) AS CantArticulos
475     FROM PROVEEDOR P
476     LEFT JOIN ARTICULO A ON P.CodProveedor = A.CodProveedor
477     GROUP BY P.CodProveedor
478 ) AS datos
479 PIVOT (
480     COUNT(CantArticulos) FOR RangoArticulos IN ([1-5], [6-10], [11-20])
481 ) AS tabla_pivot;
482 GO
483

```

75 %

Results Messages

	Rango_1_5	Rango_6_10	Rango_11_20
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0
5	1	0	0
6	1	0	0
7	1	0	0
8	1	0	0
9	1	0	0
10	1	0	0
11	0	0	1
12	0	0	1
13	0	0	1
14	0	0	1
15	0	0	1
16	0	0	1
17	0	0	1
18	0	0	1
19	0	0	1
20	0	0	1
21	0	0	1
22	0	0	1
23	0	0	1
24	0	0	1
25	0	0	1
26	0	0	1
27	0	0	1
28	0	0	1
29	0	0	1
30	0	0	1
31	0	0	1
32	0	0	1
33	0	0	1
34	0	0	1
35	0	0	1
36	0	0	1
37	0	0	1
38	0	0	1
39	0	0	1
40	0	0	1
41	0	0	1
42	0	0	1
43	0	0	1
44	0	0	1
45	0	0	1
46	0	0	1
47	0	0	1
48	0	0	1
49	0	0	1
50	0	0	1

39. Mostrar CodArticulo y columnas por año con montos total vendido

```
484 -- 39. Mostrar CodArticulo y columnas por año con montos total vendido
485 SELECT * FROM (
486     SELECT
487         GD.CodArticulo,
488         YEAR(GE.FechaSalida) AS Año,
489         CAST(SUM(GD.CantidadEnviada * GD.PrecioVenta) AS DECIMAL(18,2)) AS MontoVendido
490     FROM GUIA_ENVIO GE
491     INNER JOIN GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
492     GROUP BY GD.CodArticulo, YEAR(GE.FechaSalida)
493 ) AS datos
494 PIVOT (
495     SUM(MontoVendido) FOR Año IN ([2025])
496 ) AS tabla_pivot
497 ORDER BY CodArticulo;
498 GO
```

75 %

Results Messages

	CodArticulo	2025
1	1	11600.00
2	2	12600.00
3	3	5100.00
4	4	9500.00
5	5	2750.00
6	6	3600.00
7	7	7600.00
8	8	3500.00
9	9	1920.00
10	10	2100.00
11	11	1575.00
12	12	2275.00
13	13	2850.00
14	14	2160.00
15	15	2750.00
16	16	2200.00
17	17	2100.00
18	18	2400.00
19	19	1710.00
20	20	2400.00
21	21	2200.00
22	22	1440.00
23	23	4160.00
24	24	2080.00
25	25	1280.00
26	26	2310.00
27	27	1740.00
28	28	1690.00
29	29	2900.00
30	30	1500.00
31	31	2520.00
32	32	3840.00
33	33	2300.00
34	34	4050.00
35	35	2660.00
36	36	3500.00
37	37	2520.00
38	38	2000.00
39	39	2420.00
40	40	1110.00
41	41	2160.00
42	42	3040.00
43	43	3120.00
44	44	2380.00
45	45	1440.00
46	46	1716.00
47	47	1584.00
48	48	1900.00
49	49	6860.00
50	50	6720.00

40. Mostrar Mes y columnas por tienda (CASE alternativo)

```
-- 40. Mostrar Mes y columnas por tienda (CASE alternativo)
SELECT
    AñoMes,
    ISNULL([101], 0) AS Tienda_101,
    ISNULL([102], 0) AS Tienda_102,
    ISNULL([103], 0) AS Tienda_103,
    ISNULL([104], 0) AS Tienda_104,
    ISNULL([105], 0) AS Tienda_105
FROM (
    SELECT
        YEAR(GE.FechaSalida) * 100 + MONTH(GE.FechaSalida) AS AñoMes,
        GE.CodTienda,
        COUNT(GE.NumGuia) AS CantidadGuias
    FROM GUIA_ENVIO GE
    GROUP BY YEAR(GE.FechaSalida) * 100 + MONTH(GE.FechaSalida), GE.CodTienda
) AS datos
PIVOT (
    SUM(CantidadGuias) FOR CodTienda IN ([101], [102], [103], [104], [105])
) AS tabla_pivot
ORDER BY AñoMes;
GO
```

The screenshot shows the results of the executed query. The output is displayed in a table titled 'Results'.

	AñoMes	Tienda_101	Tienda_102	Tienda_103	Tienda_104	Tienda_105
1	202510	1	1	1	1	1
2	202511	0	0	0	0	0

SECCIÓN V: CLÁUSULA HAVING

DEFINICIÓN

HAVING es similar a WHERE, pero se utiliza para filtrar resultados después del GROUP BY y sobre funciones de agregación.

UTILIDAD

Permite aplicar condiciones a datos agrupados, por ejemplo, mostrar solo categorías con total mayor a cierto valor.

APLICACIONES COMUNES:

- Filtrar **proveedores con inventario superior a un mínimo**.
- Mostrar **líneas con más de 10 productos**.
- Seleccionar **tiendas con promedio de entrega mayor a 5 días**.
- Analizar **ventas que superen un total específico**.

EJERCICIOS DE APLICACIÓN

41. Mostrar CodLinea y CantArticulos donde CantArticulos > 10

```
526 -- 41. Mostrar CodLinea y CantArticulos donde CantArticulos > 10
527 SELECT
528     A.CodLinea,
529     COUNT(A.CodArticulo) AS CantArticulos
530     FROM ARTICULO A
531     GROUP BY A.CodLinea
532     HAVING COUNT(A.CodArticulo) > 1
533     ORDER BY A.CodLinea;
534 GO
```

12 %

Results Messages

CodLinea CantArticulos

42. Mostrar CodProveedor y MontoTotal donde MontoTotal > 50000

```
536 -- 42. Mostrar CodProveedor y MontoTotal donde MontoTotal > 50000
537 SELECT
538     P.CodProveedor,
539     SUM(CAST(A.StockActual * A.PrecioProveedor AS DECIMAL(18,2))) AS MontoTotal
540     FROM PROVEEDOR P
541     LEFT JOIN ARTICULO A ON P.CodProveedor = A.CodProveedor
542     GROUP BY P.CodProveedor
543     HAVING SUM(CAST(A.StockActual * A.PrecioProveedor AS DECIMAL(18,2))) > 50000
544     ORDER BY MontoTotal DESC;
545 GO
546
```

82 %

Results Messages

CodProveedor MontoTotal

43. Mostrar CodTienda y PromedioGuia donde PromedioGuia > 1600

```

547 -- 43. Mostrar CodTienda y PromediGuia donde PromediGuia > 1600
548 SELECT
549     GE.CodTienda,
550     SUM(GD.CantidadEnviada * GD.PrecioVenta) AS PromedioGuia
551 FROM GUIA_ENVIO GE
552 INNER JOIN GUIA_DETALLE GD
553     ON GE.NumGuia = GD.NumGuia
554 GROUP BY GE.CodTienda
555 HAVING SUM(GD.CantidadEnviada * GD.PrecioVenta) > 1600
556 ORDER BY PromedioGuia DESC;
557 GO

```

82 %

Results Messages

	CodTienda	PromedioGuia
1	102	12600,00
2	101	11600,00
3	104	9500,00
4	107	7600,00
5	149	6860,00
6	150	6720,00
7	103	5100,00
8	123	4160,00
9	134	4050,00
10	132	3840,00
11	106	3600,00
12	108	3500,00
13	136	3500,00
14	143	3120,00
15	142	3040,00
16	129	2900,00
17	113	2850,00
18	115	2750,00
19	105	2750,00
20	135	2660,00
21	137	2520,00
22	131	2520,00
23	139	2420,00
24	120	2400,00
25	118	2400,00
26	144	2380,00
27	126	2310,00
28	133	2300,00
29	112	2275,00
30	116	2200,00
31	121	2200,00
32	141	2160,00
33	114	2160,00
34	110	2100,00
35	117	2100,00
36	124	2080,00
37	138	2000,00
38	109	1920,00
39	148	1900,00
40	127	1740,00
41	146	1716,00
42	119	1710,00
43	128	1690,00

44. Mostrar CodArticulo y TotalSolicitado > 500

```
559 -- 44. Mostrar CodArticulo y TotalSolicitado > 500
560 SELECT
561     OD.CodArticulo,
562     SUM(OD.CantidadSolicitada) AS TotalSolicitado
563 FROM ORDEN_DETALLE OD
564 GROUP BY OD.CodArticulo
565 HAVING SUM(OD.CantidadSolicitada) > 500
566 ORDER BY TotalSolicitado DESC;
567 GO
568
```

82 %

CodArticulo	TotalSolicitado
-------------	-----------------

45. Mostrar CodTransportista y CantGuias >= 5

```
569 -- 45. Mostrar CodTransportista y CantGuias >= 5
570 SELECT
571     GE.CodTransportista,
572     COUNT(GE.NumGuia) AS CantGuias
573 FROM GUIA_ENVIO GE
574 GROUP BY GE.CodTransportista
575 HAVING COUNT(GE.NumGuia) >= 5
576 ORDER BY CantGuias DESC;
577 GO
578
```

82 %

CodTransportista	CantGuias
------------------	-----------

46. Mostrar líneas donde SUM(StockActual) < SUM(StockMinimo *
NumArticulosPorLinea)

```

579 -- 46. Mostrar líneas donde SUM(StockActual) < SUM(StockMinimo * NumArticulos)
580 SELECT
581     L.CodLinea,
582     L.NomLinea,
583     SUM(A.StockActual) AS StockTotal,
584     COUNT(A.CodArticulo) AS CantArticulos,
585     SUM(A.StockMinimo) AS StockMinimoTotal
586 FROM LINEA L
587 LEFT JOIN ARTICULO A ON L.CodLinea = A.CodLinea
588 GROUP BY L.CodLinea, L.NomLinea
589 HAVING SUM(A.StockActual) < COUNT(A.CodArticulo) * 10
590 ORDER BY L.CodLinea;
591 GO
592

```

82 %

	CodLinea	NomLinea	StockTotal	CantArticulos	StockMinimoTotal
1	9	Impresoras	8	1	3
2	16	Monitores Dig	5	1	2
3	30	Escritorios Game	8	1	2
4	40	Placas Madre	8	1	2

47. Mostrar proveedores donde MAX(PrecioProveedor) > 100

```

593 -- 47. Mostrar proveedores donde MAX(PrecioProveedor) > 100
594 SELECT
595     P.CodProveedor,
596     P.NomProveedor,
597     MAX(A.PrecioProveedor) AS PrecioMaximo
598 FROM PROVEEDOR P
599 LEFT JOIN ARTICULO A ON P.CodProveedor = A.CodProveedor
600 GROUP BY P.CodProveedor, P.NomProveedor
601 HAVING MAX(A.PrecioProveedor) > 100
602 ORDER BY PrecioMaximo DESC;
603 GO
604

```

82 %

	CodProveedor	NomProveedor	PrecioMaximo
1	1	Intel LatAm	1200,00
2	2	AMD Distribuidora	850,00
3	6	Lenovo Distribuidora	850,00
4	4	HP Inc Perú	750,00
5	3	NVIDIA Perú	650,00
6	9	MSI Distribution	450,00
7	5	Dell Technologies	400,00
8	10	Gigabyte Perú	380,00
9	8	Asus LatAm	380,00
10	7	Apple Distribuidor	320,00

48. Mostrar líneas con AVG(CantidadEnviada) > 50 y COUNT(NumArticulos) >= 10

```

605 -- 48. Mostrar líneas con AVG(CantidadEnviadas) > 50 y COUNT(NumArticulos) >= 10
606 SELECT
607     L.CodLinea,
608     L.NomLinea,
609     AVG(CAST(GD.CantidadEnviada AS DECIMAL(10,2))) AS PromedioCantidad,
610     COUNT(A.CodArticulo) AS CantArticulos
611 FROM LINEA L
612 LEFT JOIN ARTICULO A ON L.CodLinea = A.CodLinea
613 LEFT JOIN GUIA_DETALLE GD ON A.CodArticulo = GD.CodArticulo
614 GROUP BY L.CodLinea, L.NomLinea
615 HAVING AVG(CAST(GD.CantidadEnviada AS DECIMAL(10,2))) > 50
616     AND COUNT(A.CodArticulo) >= 1
617 ORDER BY L.CodLinea;
618 GO
619

```

82 %

Results Messages

	CodLinea	NomLinea	PromedioCantidad	CantArticulos
1	15	Tarjetas Mem	55.000000	1
2	27	Protect Pantalla	58.000000	1
3	44	Tubos Sleeving	68.000000	1
4	46	Limpiadores	78.000000	1
5	47	Protec Antiestát	88.000000	1
6	49	Manuales Garantía	98.000000	1

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

```

620 -- 49. Mostrar CodLinea donde MAX(Precio) - MIN(Precio) > 20
621 SELECT
622     A.CodLinea,
623     L.NomLinea,
624     MAX(A.PrecioProveedor) AS PrecioMax,
625     MIN(A.PrecioProveedor) AS PrecioMin,
626     MAX(A.PrecioProveedor) - MIN(A.PrecioProveedor) AS Diferencia
627 FROM ARTICULO A
628 INNER JOIN LINEA L ON A.CodLinea = L.CodLinea
629 GROUP BY A.CodLinea, L.NomLinea
630 HAVING MAX(A.PrecioProveedor) - MIN(A.PrecioProveedor) > 20
631 ORDER BY Diferencia DESC;
632 GO
633

```

82 %

Results Messages

CodLinea	NomLinea	PrecioMax	PrecioMin	Diferencia

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

```
634 -- 50. Mostrar CodProveedor con COUNT(articulos) donde AVG(StockActual) < 20 y con COUNT > 5
635 SELECT
636     P.CodProveedor,
637     P.NomProveedor,
638     COUNT(A.CodArticulo) AS CantArticulos,
639     AVG(CAST(A.StockActual AS DECIMAL(10,2))) AS PromedioStock
640 FROM PROVEEDOR P
641 LEFT JOIN ARTICULO A
642     ON P.CodProveedor = A.CodProveedor
643 GROUP BY
644     P.CodProveedor,
645     P.NomProveedor
646 HAVING
647     COUNT(A.CodArticulo) > 5
648     AND AVG(CAST(A.StockActual AS DECIMAL(10,2))) < 20;
649 GO
```

82 % ▶

Results Messages

CodProveedor	NomProveedor	CantArticulos	PromedioStock
--------------	--------------	---------------	---------------