



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



UNIVERSIDAD PERUANA LOS ANDES

“FACULTAD DE INGENIERÍA”

ESCUELA PROFESIONAL “SISTEMAS Y
COMPUTACIÓN”

CÁTEDRA: Base de Datos II

CATEDRÁTICO: Ing. Fernandez Bejarano Raul Enrique

ESTUDIANTE: Vega Brañez Samuel Max

CICLO: V

SECCIÓN: B1

HUANCAYO PERÚ

2025

Manual de implementación de consultas

CUARTA PARTE DE CONSULTAS DE LA BASE DE DATOS QHATUPERU:

OPERADOR PIVOT

IV. Explica de manera clara y didáctica qué son la **OPERADOR PIVOT** en SQL y cómo se utilizan.

31. Mostrar Fecha y columnas CodTienda_1, CodTienda_2, ... con TotalEnviado por día.
32. Enunciado: Mostrar CodArtículo y columnas con cantidades por tienda 1..3.
33. Enunciado: Mostrar AñoMes y tiendas como columnas con suma de PrecioVenta*Cantidad.
34. Enunciado: Mostrar CodArtículo con columnas para cada Estado.
35. Contar artículos por presentación pivotada.
36. Generar PIVOT dinámico para todas las tiendas (ejemplo de patrón).
37. Mostrar mes y columnas por transportista con totales.
38. Contar proveedores por rango de variedad de artículos pivotado como columnas.
39. Mostrar CodArtículo y columnas por año con monto total vendido.
40. Mostrar Mes y columnas por tienda (CASE alternative).

Explicación: Operador PIVOT en SQL

El **Operador PIVOT** en SQL se utiliza para rotar filas a columnas. Convierte los valores únicos de una columna (la columna pivote) en nuevos nombres de columnas en el resultado final, y realiza una agregación sobre otra columna (la columna de valores) que se cruza con las nuevas columnas.

- **Entrada (Filas):** Generalmente es una tabla larga (muchas filas).
- **Salida (Columnas):** Una tabla ancha (menos filas, pero más columnas), donde los datos que antes se repetían en una columna ahora son encabezados.
- **Alternativa:** El mismo resultado se puede lograr usando la función de agregación **SUM()** junto con la expresión condicional **CASE WHEN** (usado en varias de las consultas a continuación).

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

Enunciado: Mostrar la fecha de salida (DiaSalida) y utilizar los códigos de tienda (e.g., 101, 102, 103) como nuevas columnas, donde el valor de cada celda es el total de artículos enviados a esa tienda en ese día.

Consulta SQL:

-- 31. Mostrar Fecha y columnas CodTienda_1, CodTienda_2, ...
con TotalEnviado por día. (PIVOT Estático)

-- 31. CORRECCIÓN USANDO SUM(CASE)

```
SELECT
    CAST(GE.FechaSalida AS DATE) AS DiaSalida,
    SUM(CASE WHEN GE.CodTienda = 101 THEN
GD.CantidadEnviada ELSE 0 END) AS Tienda_101_TotalEnviado,
    SUM(CASE WHEN GE.CodTienda = 102 THEN
GD.CantidadEnviada ELSE 0 END) AS Tienda_102_TotalEnviado,
    SUM(CASE WHEN GE.CodTienda = 103 THEN
GD.CantidadEnviada ELSE 0 END) AS Tienda_103_TotalEnviado
FROM
    GUIA_ENVIO GE
JOIN
    GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
GROUP BY
    CAST(GE.FechaSalida AS DATE)
ORDER BY
    DiaSalida;
GO
```

The screenshot shows a SQL query execution in SQL Server Enterprise Manager. The query is a static PIVOT query using SUM(CASE) to aggregate data by CodTienda (101, 102, 103) and DiaSalida. The results table shows 17 rows of data with columns DiaSalida, Tienda_101_TotalEnviado, Tienda_102_TotalEnviado, and Tienda_103_TotalEnviado.

| | DiaSalida | Tienda_101_TotalEnviado | Tienda_102_TotalEnviado | Tienda_103_TotalEnviado |
|----|------------|-------------------------|-------------------------|-------------------------|
| 1 | 2025-09-10 | 0 | 0 | 0 |
| 2 | 2025-09-11 | 0 | 0 | 0 |
| 3 | 2025-09-12 | 0 | 0 | 0 |
| 4 | 2025-09-13 | 0 | 0 | 0 |
| 5 | 2025-09-14 | 0 | 0 | 0 |
| 6 | 2025-09-15 | 0 | 0 | 0 |
| 7 | 2025-09-16 | 0 | 0 | 0 |
| 8 | 2025-09-17 | 0 | 0 | 0 |
| 9 | 2025-09-18 | 0 | 0 | 0 |
| 10 | 2025-09-19 | 0 | 0 | 0 |
| 11 | 2025-09-20 | 0 | 0 | 0 |
| 12 | 2025-09-21 | 0 | 0 | 0 |
| 13 | 2025-09-22 | 0 | 0 | 0 |
| 14 | 2025-09-23 | 0 | 0 | 0 |
| 15 | 2025-09-24 | 0 | 0 | 0 |
| 16 | 2025-09-25 | 0 | 0 | 0 |
| 17 | 2025-09-26 | 0 | 0 | 0 |

Explicación: Se usa el operador **PIVOT** de forma estática. La columna CodTienda se convierte en encabezados ([101], [102], [103]), y la función de agregación **SUM(TotalEnviado)** rellena los valores. La columna restante (DiaSalida) actúa como fila principal.

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

Enunciado: Mostrar el código y descripción del artículo con columnas separadas para la cantidad total de unidades enviadas a la Tienda 101, Tienda 102 y Tienda 103.

Consulta SQL:

-- 32. Mostrar CodArticulo y columnas con cantidades por tienda 1..3. (PIVOT Estático con SUM(CASE))

SELECT

A.CodArticulo,
A.DescripcionArticulo,

```
SUM(CASE WHEN GE.CodTienda = 101 THEN
GD.CantidadEnviada ELSE 0 END) AS Tienda_101_Cantidad,
SUM(CASE WHEN GE.CodTienda = 102 THEN
GD.CantidadEnviada ELSE 0 END) AS Tienda_102_Cantidad,
SUM(CASE WHEN GE.CodTienda = 103 THEN
GD.CantidadEnviada ELSE 0 END) AS Tienda_103_Cantidad
FROM
ARTICULO A
LEFT JOIN
    GUIA_DETALLE GD ON A.CodArticulo = GD.CodArticulo
LEFT JOIN
    GUIA_ENVIO GE ON GD.NumGuia = GE.NumGuia
GROUP BY
    A.CodArticulo, A.DescripcionArticulo
ORDER BY
    A.CodArticulo;
GO
```

File Edit View Help Search

CONNECTIONS: AZURE + ... connected SQLQuery_1.sql - (62) t...U (sa1) SQLQuery_1 - (53) t...U (sa1) 2

Run Cancel Disconnect Change Database: QhātuPERU Estimated Plan

Enable Actual Plan Parse

```

30  DiaSalida;
31  GO
32  ---
33
34  -- 32. Mostrar CodArticulo y columnas con cantidades por tienda 1..3. (PIVOT Estático)
35  SELECT
36      A.CodArticulo,
37      A.DescripcionArticulo,
38      SUM(CASE WHEN GE.CodTienda = 101 THEN GD.CantidadEnviada ELSE 0 END) AS Tienda_101
39      SUM(CASE WHEN GE.CodTienda = 102 THEN GD.CantidadEnviada ELSE 0 END) AS Tienda_102
40      SUM(CASE WHEN GE.CodTienda = 103 THEN GD.CantidadEnviada ELSE 0 END) AS Tienda_103
41  FROM

```

Results Messages

| | CodArticulo | DescripcionArticulo | Tienda_101_Cantidad | Tienda_1 |
|----|-------------|-------------------------------|---------------------|----------|
| 1 | 1 | Filtro Hepa Mod-2 Sku-1 | 53 | 0 |
| 2 | 2 | Rodamiento Z-20 Mod-3 Sku-2 | 0 | 56 |
| 3 | 3 | Sensor Óptico Mod-4 Sku-3 | 0 | 0 |
| 4 | 4 | Rodamiento Z-20 Mod-5 Sku-4 | 0 | 0 |
| 5 | 5 | Válvula Flujo Mod-6 Sku-5 | 0 | 0 |
| 6 | 6 | Malla Industrial Mod-7 Sku-6 | 0 | 0 |
| 7 | 7 | Válvula Flujo Mod-8 Sku-7 | 0 | 0 |
| 8 | 8 | Filtro Hepa Mod-9 Sku-8 | 0 | 0 |
| 9 | 9 | Válvula Flujo Mod-10 Sku-9 | 0 | 0 |
| 10 | 10 | Malla Industrial Mod-1 Sku-10 | 0 | 0 |
| 11 | 11 | Sensor Óptico Mod-2 Sku-11 | 0 | 0 |
| 12 | 12 | Cable Cat 6 Mod-3 Sku-12 | 0 | 0 |
| 13 | 13 | Malla Industrial Mod-4 Sku-13 | 0 | 0 |
| 14 | 14 | Rodamiento Z-20 Mod-5 Sku-14 | 0 | 0 |
| 15 | 15 | Válvula Flujo Mod-6 Sku-15 | 0 | 0 |
| 16 | 16 | Tornillo Titanio Mod-7 Sku-16 | 0 | 0 |

Ln 34, Col 1 (680 selected) Spaces: 4 UTF-8 CRLF 100 rows MSSQL 00:00:00 tcp:jean3.database.windows.net : QhātuPERU (53)

File Edit View Help Search

CONNECTIONS: AZURE + ... connected SQLQuery_1.sql - (62) t...U (sa1) SQLQuery_1 - (53) t...U (sa1) 2

Run Cancel Disconnect Change Database: QhātuPERU Estimated Plan

Enable Actual Plan Parse

```

33
34  -- 32. Mostrar CodArticulo y columnas con cantidades por tienda 1..3. (PIVOT Estático)
35  SELECT
36      A.CodArticulo,
37      A.DescripcionArticulo,
38      SUM(CASE WHEN GE.CodTienda = 101 THEN GD.CantidadEnviada ELSE 0 END) AS Tienda_101
39      SUM(CASE WHEN GE.CodTienda = 102 THEN GD.CantidadEnviada ELSE 0 END) AS Tienda_102
40      SUM(CASE WHEN GE.CodTienda = 103 THEN GD.CantidadEnviada ELSE 0 END) AS Tienda_103
41  FROM
42      ARTICULO A
43  LEFT JOIN
44      GUIA_DETALLE GD ON A.CodArticulo = GD.CodArticulo
45  LEFT JOIN

```

Results Messages

| | CodArticulo | DescripcionArticulo | Tienda_101_Cantidad | Tienda_1 |
|-----|-------------|----------------------------------|---------------------|----------|
| 84 | 84 | Cable Cat 6 Mod-5 Sku-84 | 0 | 0 |
| 85 | 85 | Filtro Hepa Mod-6 Sku-85 | 0 | 0 |
| 86 | 86 | Filtro Hepa Mod-7 Sku-86 | 0 | 0 |
| 87 | 87 | Interruptor Termico Mod-8 Sku-87 | 0 | 0 |
| 88 | 88 | Filtro Hepa Mod-9 Sku-88 | 0 | 0 |
| 89 | 89 | Rodamiento Z-20 Mod-10 Sku-89 | 0 | 0 |
| 90 | 90 | Tornillo Titanio Mod-1 Sku-90 | 0 | 0 |
| 91 | 91 | Rodamiento Z-20 Mod-2 Sku-91 | 0 | 0 |
| 92 | 92 | Tornillo Titanio Mod-3 Sku-92 | 0 | 0 |
| 93 | 93 | Interruptor Termico Mod-4 Sku-93 | 0 | 0 |
| 94 | 94 | Interruptor Termico Mod-5 Sku-94 | 0 | 0 |
| 95 | 95 | Cable Cat 6 Mod-6 Sku-95 | 0 | 0 |
| 96 | 96 | Interruptor Termico Mod-7 Sku-96 | 0 | 0 |
| 97 | 97 | Sensor Óptico Mod-8 Sku-97 | 0 | 0 |
| 98 | 98 | Malla Industrial Mod-9 Sku-98 | 0 | 0 |
| 99 | 99 | Tornillo Titanio Mod-10 Sku-99 | 0 | 0 |
| 100 | 100 | Válvula Flujo Mod-1 Sku-100 | 0 | 0 |

Ln 34, Col 1 (680 selected) Spaces: 4 UTF-8 CRLF 100 rows MSSQL 00:00:00 tcp:jean3.database.windows.net : QhātuPERU (53)

Explicación: Se utiliza la técnica de **Agregación Condicional** (SUM(CASE WHEN...)) como alternativa al operador PIVOT. Se agrupa por artículo y se suman las cantidades solo cuando la tienda cumple la condición específica dentro del CASE.

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

Enunciado: Mostrar los resultados de venta agrupados por mes y año (AñoMes), utilizando los códigos de tienda como columnas, donde el valor es el monto total vendido (PrecioVenta * Cantidad).

Consulta SQL:

```
-- 33. CORRECCIÓN USANDO CTE (Para evitar el error de alias)
WITH VentasMensuales AS (
    SELECT
        FORMAT(GE.FechaSalida, 'yyyy-MM') AS AnoMes, -- El alias
        se define primero en la CTE
        GE.CodTienda,
        (GD.PrecioVenta * GD.CantidadEnviada) AS Monto
    FROM
        GUIA_ENVIO GE
    JOIN
        GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
)
SELECT
    AnoMes,
    SUM(CASE WHEN CodTienda = 101 THEN Monto ELSE 0 END)
AS Monto_Tienda_101,
    SUM(CASE WHEN CodTienda = 102 THEN Monto ELSE 0 END)
AS Monto_Tienda_102,
    SUM(CASE WHEN CodTienda = 103 THEN Monto ELSE 0 END)
AS Monto_Tienda_103
FROM
    VentasMensuales
GROUP BY
    AnoMes -- ¡Ahora sí existe en el GROUP BY!
ORDER BY
    AnoMes;
GO
```

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Connections' pane shows the 'QhātuPERU' database selected. The 'Tables' folder is expanded, showing 'dbo.ARTICULO'. The main pane displays a SQL query with a CTE named 'VentasMensuales'. The query calculates monthly sales by date and store code. The results table shows data for September 2025 and October 2025.

| | AnoMes | Monto_Tienda_101 | Monto_Tienda_102 | Monto_Tienda_103 |
|---|---------|------------------|------------------|------------------|
| 1 | 2025-09 | 0.00 | 0.00 | 0.00 |
| 2 | 2025-10 | 196.10 | 246.40 | 300.90 |

Explicación: Se usa la **Agregación Condicional** (SUM(CASE WHEN...)). Se agrupa el resultado por la expresión de fecha FORMAT(GE.FechaSalida, 'yyyy-MM') y se suma el valor monetario (PrecioVenta * CantidadEnviada) solo si el registro pertenece al CodTienda especificado.

34. Mostrar CodArticulo con columnas para cada

Estado. Enunciado: Mostrar el código de artículo y pivotar la columna

Estado de las órdenes de detalle ('Recibido', 'Pendiente') para ver la cantidad solicitada total en cada estado.

Consulta SQL:

-- 34. CORREGIDO: Mostrar CodArticulo con columnas para cada Estado (SUM(CASE) para evitar NULL)

SELECT

CodArticulo,


```
-- Si el estado es 'Recibido', suma la cantidad solicitada, si no,
suma 0.
SUM(CASE WHEN Estado = 'Recibido' THEN CantidadSolicitada
ELSE 0 END) AS TotalRecibido,
-- Si el estado es 'Pendiente', suma la cantidad solicitada, si no,
suma 0.
SUM(CASE WHEN Estado = 'Pendiente' THEN
CantidadSolicitada ELSE 0 END) AS TotalPendiente
FROM
ORDEN_DETALLE
GROUP BY
CodArticulo
ORDER BY
CodArticulo;
GO
```

File Edit View Help

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

Database: QhatuPERU

Run Cancel Disconnect Change Estimated Plan

Enable Actual Plan Parse

```

83      CodArticulo,
84      -- Si el estado es 'Recibido', suma la cantidad solicitada, si no, suma 0.
85      SUM(CASE WHEN Estado = 'Recibido' THEN CantidadSolicitada ELSE 0 END) AS TotalReci
86      -- Si el estado es 'Pendiente', suma la cantidad solicitada, si no, suma 0.
87      SUM(CASE WHEN Estado = 'Pendiente' THEN CantidadSolicitada ELSE 0 END) AS TotalPen
88  FROM
89      ORDEN_DETALLE
90  GROUP BY
91      CodArticulo
92  ORDER BY
93      CodArticulo;
94  GO
95  
```

Results Messages

| | CodArticulo | TotalRecibido | TotalPendiente |
|----|-------------|---------------|----------------|
| 1 | 1 | 105 | 0 |
| 2 | 2 | 110 | 0 |
| 3 | 3 | 115 | 0 |
| 4 | 4 | 120 | 0 |
| 5 | 5 | 0 | 125 |
| 6 | 6 | 130 | 0 |
| 7 | 7 | 135 | 0 |
| 8 | 8 | 140 | 0 |
| 9 | 9 | 145 | 0 |
| 10 | 10 | 0 | 150 |
| 11 | 11 | 155 | 0 |
| 12 | 12 | 160 | 0 |
| 13 | 13 | 165 | 0 |
| 14 | 14 | 170 | 0 |
| 15 | 15 | 0 | 175 |
| 16 | 16 | 180 | 0 |
| 17 | 17 | 185 | 0 |

Ln 81, Col 1 (554 selected) Spaces: 4 UTF-8 CRLF 100 rows MSSQL 00:00:00 tcp:jean3.database.windows.net : QhatuPERU (53)

File Edit View Help

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

Database: QhatuPERU

Run Cancel Disconnect Change Estimated Plan

Enable Actual Plan Parse

```

83      CodArticulo,
84      -- Si el estado es 'Recibido', suma la cantidad solicitada, si no, suma 0.
85      SUM(CASE WHEN Estado = 'Recibido' THEN CantidadSolicitada ELSE 0 END) AS TotalReci
86      -- Si el estado es 'Pendiente', suma la cantidad solicitada, si no, suma 0.
87      SUM(CASE WHEN Estado = 'Pendiente' THEN CantidadSolicitada ELSE 0 END) AS TotalPen
88  FROM
89      ORDEN_DETALLE
90  GROUP BY
91      CodArticulo
92  ORDER BY
93      CodArticulo;
94  GO
95  
```

Results Messages

| | CodArticulo | TotalRecibido | TotalPendiente |
|-----|-------------|---------------|----------------|
| 84 | 84 | 520 | 0 |
| 85 | 85 | 0 | 525 |
| 86 | 86 | 530 | 0 |
| 87 | 87 | 535 | 0 |
| 88 | 88 | 540 | 0 |
| 89 | 89 | 545 | 0 |
| 90 | 90 | 0 | 550 |
| 91 | 91 | 555 | 0 |
| 92 | 92 | 560 | 0 |
| 93 | 93 | 565 | 0 |
| 94 | 94 | 570 | 0 |
| 95 | 95 | 0 | 575 |
| 96 | 96 | 580 | 0 |
| 97 | 97 | 585 | 0 |
| 98 | 98 | 590 | 0 |
| 99 | 99 | 595 | 0 |
| 100 | 100 | 0 | 600 |

Ln 81, Col 1 (554 selected) Spaces: 4 UTF-8 CRLF 100 rows MSSQL 00:00:00 tcp:jean3.database.windows.net : QhatuPERU (53)

Explicación: Se utiliza el operador **PIVOT** para rotar los valores de la columna Estado (Recibido, Pendiente) y convertirlos en columnas. La agregación aplicada es **SUM(CantidadSolicitada)**.

35. Contar artículos por presentación pivotada.

Enunciado: Contar cuántos artículos existen por cada línea de producto, pivotando la columna Presentacion (e.g., 'Paq.1', 'Paq.2') para que cada presentación sea una columna.

Consulta SQL:

-- 35. Contar artículos por presentación pivotada. (PIVOT Estático con Agregación Condicional)

```
SELECT
    L.NomLinea,
    COUNT(CASE WHEN A.Presentacion = 'Paq.1' THEN
A.CodArticulo END) AS Presentacion_Paq1,
    COUNT(CASE WHEN A.Presentacion = 'Paq.2' THEN
A.CodArticulo END) AS Presentacion_Paq2,
    COUNT(CASE WHEN A.Presentacion = 'Paq.3' THEN
A.CodArticulo END) AS Presentacion_Paq3
FROM
    ARTICULO A
JOIN
    LINEA L ON A.CodLinea = L.CodLinea
GROUP BY
    L.NomLinea
ORDER BY
    L.NomLinea;
```

GO

File Edit View Help Search

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

Run Cancel Disconnect Change Estimated Plan

Enable Actual Plan Parse

```
86 -- Si el estado es 'Pendiente', suma la cantidad solicitada, si no, suma 0.
87 SUM(CASE WHEN Estado = 'Pendiente' THEN CantidadSolicitada ELSE 0 END) AS TotalPer
88 FROM
89 ORDEN_DETALLE
90 GROUP BY
91 CodArticulo
92 ORDER BY
93 CodArticulo;
94 GO
95 ---
96
97 -- 35. Contar articulos por presentación pivotada. (PIVOT Estático con Agregación Cond
```

Results Messages

| | NomLinea | Presentacion_Paq1 | Presentacion_Paq2 | Presentacion_Paq3 |
|----|-----------|-------------------|-------------------|-------------------|
| 1 | LN-ACAB13 | 0 | 1 | 0 |
| 2 | LN-ACAB18 | 0 | 0 | 1 |
| 3 | LN-ACAB23 | 0 | 0 | 0 |
| 4 | LN-ACAB28 | 1 | 0 | 0 |
| 5 | LN-ACAB3 | 0 | 0 | 0 |
| 6 | LN-ACAB33 | 0 | 1 | 0 |
| 7 | LN-ACAB38 | 0 | 0 | 1 |
| 8 | LN-ACAB43 | 0 | 0 | 0 |
| 9 | LN-ACAB48 | 1 | 0 | 0 |
| 10 | LN-ACAB53 | 0 | 1 | 0 |
| 11 | LN-ACAB58 | 0 | 0 | 1 |
| 12 | LN-ACAB63 | 0 | 0 | 0 |
| 13 | LN-ACAB68 | 1 | 0 | 0 |
| 14 | LN-ACAB73 | 0 | 1 | 0 |
| 15 | LN-ACAB78 | 0 | 0 | 1 |
| 16 | LN-ACAB8 | 1 | 0 | 0 |
| 17 | LN-ACAB83 | 0 | 0 | 0 |

Ln 97, Col 1 (519 selected) Spaces: 4 UTF-8 CRLF 100 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

Run Cancel Disconnect Change Estimated Plan

Enable Actual Plan Parse

```
86 -- Si el estado es 'Pendiente', suma la cantidad solicitada, si no, suma 0.
87 SUM(CASE WHEN Estado = 'Pendiente' THEN CantidadSolicitada ELSE 0 END) AS TotalPer
88 FROM
89 ORDEN_DETALLE
90 GROUP BY
91 CodArticulo
92 ORDER BY
93 CodArticulo;
94 GO
95 ---
96
97 -- 35. Contar articulos por presentación pivotada. (PIVOT Estático con Agregación Cond
```

Results Messages

| | NomLinea | Presentacion_Paq1 | Presentacion_Paq2 | Presentacion_Paq3 |
|-----|------------|-------------------|-------------------|-------------------|
| 84 | LN-HERRKZZ | 0 | 0 | 1 |
| 85 | LN-HERR27 | 0 | 0 | 0 |
| 86 | LN-HERR32 | 1 | 0 | 0 |
| 87 | LN-HERR37 | 0 | 1 | 0 |
| 88 | LN-HERR42 | 0 | 0 | 1 |
| 89 | LN-HERR47 | 0 | 0 | 0 |
| 90 | LN-HERR52 | 1 | 0 | 0 |
| 91 | LN-HERR57 | 0 | 1 | 0 |
| 92 | LN-HERR62 | 0 | 0 | 1 |
| 93 | LN-HERR67 | 0 | 0 | 0 |
| 94 | LN-HERR7 | 0 | 0 | 0 |
| 95 | LN-HERR72 | 1 | 0 | 0 |
| 96 | LN-HERR77 | 0 | 1 | 0 |
| 97 | LN-HERR82 | 0 | 0 | 1 |
| 98 | LN-HERR87 | 0 | 0 | 0 |
| 99 | LN-HERR92 | 1 | 0 | 0 |
| 100 | LN-HERR97 | 0 | 1 | 0 |

Ln 97, Col 1 (519 selected) Spaces: 4 UTF-8 CRLF 100 rows MSSQL 00:00:00 tcpjean3.database.windows.net : QhatuPERU (53)

Explicación: Se utiliza la **Agregación Condicional** (COUNT(CASE WHEN...)) para el conteo. La función COUNT solo cuenta los valores no nulos; al colocar A.CodArticulo dentro del CASE y no tener un ELSE, solo se cuenta el artículo si su presentación coincide con la condición, logrando el efecto de pivoteo.

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

Enunciado: Generar la consulta del Punto 31 de forma dinámica, para que la cantidad de columnas de tienda se ajuste automáticamente si se añaden nuevas tiendas a la base de datos, sin tener que modificar el código SQL.

Consulta SQL:

-- 36. Generar PIVOT dinámico para todas las tiendas (ejemplo de patrón) - CORREGIDO PARA MOSTRAR CERO EN LUGAR DE NULL

```
DECLARE @ColumnasPivotadas NVARCHAR(MAX),
        @ColumnasISNULL NVARCHAR(MAX), @SQL NVARCHAR(MAX);
```

-- 1. Construir la lista de columnas pivotadas (e.g., [101], [102], [103], ...)

```
SELECT
```

```
    @ColumnasPivotadas = COALESCE(@ColumnasPivotadas + ', ',
    ") + QUOTENAME(CodTienda)
```

```
FROM
```

```
    TIENDA
```

```
ORDER BY
```

```
    CodTienda;
```

-- 2. Construir la lista de columnas con ISNULL (e.g., ISNULL([101], 0) AS [101], ISNULL([102], 0) AS [102], ...)

```
SELECT
```

```
    @ColumnasISNULL = COALESCE(@ColumnasISNULL + ', ', '') +
    'ISNULL(' + QUOTENAME(CodTienda) + ', 0) AS ' +
    QUOTENAME(CodTienda)
```

```
FROM
```

```
    TIENDA
```

ORDER BY

CodTienda;

-- 3. Construir la consulta SQL dinámica

SET @SQL =

N'

SELECT

DiaSalida,

' + @ColumnasIsNull + N' -- Sustituye NULL por 0

FROM

(

SELECT

CAST(GE.FechaSalida AS DATE) AS DiaSalida,

GE.CodTienda,

SUM(GD.CantidadEnviada) AS TotalEnviado

FROM

GUIA_ENVIO GE

JOIN

GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia

GROUP BY

CAST(GE.FechaSalida AS DATE), GE.CodTienda

) AS Fuente

PIVOT

(

SUM(TotalEnviado)

FOR CodTienda IN (' + @ColumnasPivotadas + N')

) AS TablaPivot

ORDER BY

DiaSalida;';

-- 4. Ejecutar la consulta dinámica

EXEC sp_executesql @SQL;

GO

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Tables' folder under 'dbo' is expanded, showing 'dbo.ARTICULO'. The main pane displays a SQL query (SQLQuery_1) with a dynamic PIVOT statement. The query uses a variable '@ColumnasPivotadas' to dynamically generate the PIVOT clause based on the 'CodTienda' values in the 'TIENDA' table. The results pane shows a table with columns: DiaSalida, 101, 102, 103, 104, 105, 106, 107. The data shows dates from 2025-09-10 to 2025-09-25, with values 0 for all columns except 'DiaSalida'.

```

115 -- 36. Generar PIVOT dinámico para todas las tiendas (ejemplo de patrón) -- CORREGIDO
116 DECLARE @ColumnasPivotadas NVARCHAR(MAX), @ColumnasISNULL NVARCHAR(MAX), @SQL NVARCHAR(MAX)
117
118 -- 1. Construir la lista de columnas pivotadas (e.g., [101], [102], [103], ...)
119 SELECT
120     @ColumnasPivotadas = COALESCE(@ColumnasPivotadas + ', ', '') + QUOTENAME(CodTienda)
121 FROM
122     TIENDA
123 ORDER BY
124     CodTienda;
125
126 -- 2. Construir la lista de columnas con ISNULL (e.g., ISNULL([101], 0) AS [101], ISNULL([102], 0) AS [102], ...)
127

```

| | DiaSalida | 101 | 102 | 103 | 104 | 105 | 106 | 107 |
|----|------------|-----|-----|-----|-----|-----|-----|-----|
| 1 | 2025-09-10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 2025-09-11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 2025-09-12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 2025-09-13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 2025-09-14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 2025-09-15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 2025-09-16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 2025-09-17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 2025-09-18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 2025-09-19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | 2025-09-20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 2025-09-21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 13 | 2025-09-22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 2025-09-23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 2025-09-24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | 2025-09-25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Explicación: Se utiliza SQL dinámico (EXEC sp_executesql). Primero, una variable (@ColumnasPivotadas) construye una lista de todos los CodTienda existentes, separados por coma y entre corchetes ([]). Luego, esta variable se concatena en la cadena de la consulta PIVOT, que se ejecuta al final. Esto hace que el PIVOT se adapte a cualquier cambio en la tabla TIENDA.

37. Mostrar mes y columnas por transportista con totales.

Enunciado: Mostrar el monto total vendido (PrecioVenta * Cantidad) agrupado por mes, utilizando los nombres de los transportistas (o una parte de ellos) como columnas.

Consulta SQL:

-- 37. Mostrar mes y columnas por transportista con totales. (PIVOT Estático con Agregación Condicional)

SELECT

DATENAME(MONTH, GE.FechaSalida) AS Mes,

```

SUM(CASE WHEN T.NomTransportista LIKE 'Transp. Rápido%'
THEN (GD.CantidadEnviada * GD.PrecioVenta) ELSE 0 END) AS
Monto_Transp_Rapido,
SUM(CASE WHEN T.NomTransportista LIKE 'Courier Veloz%'
THEN (GD.CantidadEnviada * GD.PrecioVenta) ELSE 0 END) AS
Monto_Courier_Veloz,
SUM(CASE WHEN T.NomTransportista LIKE 'LogiExpress%'
THEN (GD.CantidadEnviada * GD.PrecioVenta) ELSE 0 END) AS
Monto_LogiExpress
FROM
    GUIA_ENVIO GE
JOIN
    GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
JOIN
    TRANSPORTISTA T ON GE.CodTransportista =
T.CodTransportista
GROUP BY
    DATENAME(MONTH, GE.FechaSalida),
    MONTH(GE.FechaSalida) -- Agrupa por nombre y número de mes
para ordenar
ORDER BY
    MONTH(GE.FechaSalida);
GO

```

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the database structure for 'QhātuPERU', including tables like 'dbo.GUÍA_DETALLE', 'dbo.GUÍA_ENVIO', and 'dbo.TRANSPORTISTA'. The right pane shows the execution of a query in 'SQLQuery_1 - (53) t...U (sa1)'. The query is a PIVOT query that calculates the total amount for different transport companies (Rápido, Veloz, LogiExpress) grouped by month. The results are displayed in a table with columns: Mes, Monto_Transp_Rapido, Monto_Courier_Veloz, and Monto_LogiExpress. The results show data for September and October.

| | Mes | Monto_Transp_Rapido | Monto_Courier_Veloz | Monto_LogiExpress |
|---|-----------|---------------------|---------------------|-------------------|
| 1 | September | 43159.30 | 76380.10 | 43322.90 |
| 2 | October | 57731.40 | 43904.50 | 26783.70 |

Explicación: Se usa la **Agregación Condicional** (SUM(CASE WHEN...)) para pivotar los montos de venta. Se agrupa por el nombre del mes (DATENAME) y se suma el valor de venta solo para los transportistas que cumplen con la condición LIKE. El MONTH(GE.FechaSalida) se usa en el GROUP BY y ORDER BY para asegurar el orden cronológico.

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

Enunciado: Clasificar a los proveedores según la cantidad de artículos diferentes que suministran (Variedad Baja, Media o Alta) y pivotar estos rangos para contar cuántos proveedores caen en cada categoría.

Consulta SQL:

```
-- 38. Contar proveedores por rango de variedad de artículos
pivotado como columnas. (PIVOT Estático usando CTE y CASE)
WITH ProveedorVariedad AS (
  -- 1. Contar cuántos artículos suministra cada proveedor
  SELECT
    CodProveedor,
    COUNT(CodArticulo) AS
    CantArticulos FROM
    ARTICULO
  GROUP BY
    CodProveedor
),
RangoProveedor AS (
  -- 2. Clasificar cada proveedor en un rango de variedad
  SELECT
    CodProveedor,
    CASE
      WHEN CantArticulos = 1 THEN 'Baja (1 art.)'
      WHEN CantArticulos BETWEEN 2 AND 5 THEN 'Media (2-5
arts.)'
      ELSE 'Alta (>5 arts.)'
    END AS Rango
  FROM
    ProveedorVariedad
)
```

-- 3. Pivotar los rangos (contar cuántos proveedores caen en cada rango)

SELECT

'Total Proveedores' AS Categoria,

SUM(CASE WHEN Rango = 'Baja (1 art.)' THEN 1 ELSE 0 END)

AS Baja,

SUM(CASE WHEN Rango = 'Media (2-5 arts.)' THEN 1 ELSE 0
END) AS Media,

SUM(CASE WHEN Rango = 'Alta (>5 arts.)' THEN 1 ELSE 0
END) AS Alta

FROM

RangoProveedor;

GO

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the database structure for 'QhatsuPERU', including tables like 'dbo.ARTICULO'. The right pane shows the execution plan for a query, with the 'Results' tab active. The results table has the following data:

| | Categoria | Baja | Media | Alta |
|---|-------------------|------|-------|------|
| 1 | Total Proveedores | 100 | 0 | 0 |

Explicación: Se utiliza un patrón de CTE anidadas y **Agregación Condicional**. La primera CTE calcula la variedad, la segunda CTE clasifica al proveedor en un Rango usando CASE, y la consulta final cuenta (SUM(CASE WHEN...)) cuántos proveedores caen en cada uno de los rangos definidos.

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

Enunciado: Mostrar el código de artículo y pivotar el año de la venta para que cada año se convierta en una columna, mostrando el monto total vendido de ese artículo en dicho año.

Consulta SQL:

-- 39. CORREGIDO: Mostrar CodArticulo y columnas por año con monto total vendido (SUM(CASE) para evitar NULL)

SELECT

GD.CodArticulo,

A.DescripcionArticulo,

-- Si el año es 2024, suma el monto, si no, suma 0.

SUM(CASE WHEN YEAR(GE.FechaSalida) = 2024 THEN
(GD.CantidadEnviada * GD.PrecioVenta) ELSE 0 END) AS

Ventas_2024,

-- Si el año es 2025, suma el monto, si no, suma 0.

SUM(CASE WHEN YEAR(GE.FechaSalida) = 2025 THEN
(GD.CantidadEnviada * GD.PrecioVenta) ELSE 0 END) AS

Ventas_2025

FROM

GUIA_DETALLE GD

JOIN

GUIA_ENVIO GE ON GD.NumGuia = GE.NumGuia

JOIN

ARTICULO A ON GD.CodArticulo = A.CodArticulo

GROUP BY

GD.CodArticulo, A.DescripcionArticulo

ORDER BY

GD.CodArticulo;

GO

File Edit View Help

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

Run Cancel Disconnect Change Database: QhatuPERU Estimated Plan

Enable Actual Plan Parse

```
220 ---
221 -- 39. CORREGIDO: Mostrar CodArticulo y columnas por año con monto total vendido (SUM
222 SELECT
223     GD.CodArticulo,
224     A.DescripcionArticulo,
225     -- Si el año es 2024, suma el monto, si no, suma 0.
226     SUM(CASE WHEN YEAR(GE.FechaSalida) = 2024 THEN (GD.CantidadEnviada * GD.PrecioVent
227     -- Si el año es 2025, suma el monto, si no, suma 0.
228     SUM(CASE WHEN YEAR(GE.FechaSalida) = 2025 THEN (GD.CantidadEnviada * GD.PrecioVent
229 FROM
230     GUIDA_DETALLE GD
231 )
```

Results Messages

| | CodArticulo | DescripcionArticulo | Ventas_2024 | Ventas_2025 |
|----|-------------|-------------------------------|-------------|-------------|
| 1 | 1 | Filtro Hepa Mod-2 Sku-1 | 0.00 | 196.10 |
| 2 | 2 | Rodamiento Z-20 Mod-3 Sku-2 | 0.00 | 246.40 |
| 3 | 3 | Sensor Óptico Mod-4 Sku-3 | 0.00 | 300.90 |
| 4 | 4 | Rodamiento Z-20 Mod-5 Sku-4 | 0.00 | 359.60 |
| 5 | 5 | Válvula Flujo Mod-6 Sku-5 | 0.00 | 422.50 |
| 6 | 6 | Malla Industrial Mod-7 Sku-6 | 0.00 | 489.60 |
| 7 | 7 | Válvula Flujo Mod-8 Sku-7 | 0.00 | 560.90 |
| 8 | 8 | Filtro Hepa Mod-9 Sku-8 | 0.00 | 636.40 |
| 9 | 9 | Válvula Flujo Mod-10 Sku-9 | 0.00 | 716.10 |
| 10 | 10 | Malla Industrial Mod-1 Sku-10 | 0.00 | 800.00 |
| 11 | 11 | Sensor Óptico Mod-2 Sku-11 | 0.00 | 888.10 |
| 12 | 12 | Cable Cat 6 Mod-3 Sku-12 | 0.00 | 980.40 |
| 13 | 13 | Malla Industrial Mod-4 Sku-13 | 0.00 | 1076.90 |
| 14 | 14 | Rodamiento Z-20 Mod-5 Sku-14 | 0.00 | 1177.60 |
| 15 | 15 | Válvula Flujo Mod-6 Sku-15 | 0.00 | 1282.50 |
| 16 | 16 | Tornillo Titanio Mod-7 Sku-16 | 0.00 | 1391.60 |

Ln 222, Col 1 (738 selected) Spaces: 4 UTF-8 CRLF 100 rows MSSQL 00:00:00 tcp:jean3.database.windows.net : QhatuPERU (53)

File Edit View Help

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

Run Cancel Disconnect Change Database: QhatuPERU Estimated Plan

Enable Actual Plan Parse

```
220 ---
221 -- 39. CORREGIDO: Mostrar CodArticulo y columnas por año con monto total vendido (SUM
222 SELECT
223     GD.CodArticulo,
224     A.DescripcionArticulo,
225     -- Si el año es 2024, suma el monto, si no, suma 0.
226     SUM(CASE WHEN YEAR(GE.FechaSalida) = 2024 THEN (GD.CantidadEnviada * GD.PrecioVent
227     -- Si el año es 2025, suma el monto, si no, suma 0.
228     SUM(CASE WHEN YEAR(GE.FechaSalida) = 2025 THEN (GD.CantidadEnviada * GD.PrecioVent
229 FROM
230     GUIDA_DETALLE GD
231 )
```

Results Messages

| | CodArticulo | DescripcionArticulo | Ventas_2024 | Ventas_2025 |
|-----|-------------|---------------------------------|-------------|-------------|
| 84 | 84 | Cable Cat 6 Mod-5 SKU-84 | 0.00 | 18003.00 |
| 85 | 85 | Filtro Hepa Mod-6 Sku-85 | 0.00 | 19062.50 |
| 86 | 86 | Filtro Hepa Mod-7 Sku-86 | 0.00 | 19465.60 |
| 87 | 87 | Interruptor Termico Mod-8 Sk... | 0.00 | 19872.90 |
| 88 | 88 | Filtro Hepa Mod-9 Sku-88 | 0.00 | 20284.40 |
| 89 | 89 | Rodamiento Z-20 Mod-10 Sku-89 | 0.00 | 20700.10 |
| 90 | 90 | Tornillo Titanio Mod-1 Sku-90 | 0.00 | 21120.00 |
| 91 | 91 | Rodamiento Z-20 Mod-2 Sku-91 | 0.00 | 21544.10 |
| 92 | 92 | Tornillo Titanio Mod-3 Sku-92 | 0.00 | 21972.40 |
| 93 | 93 | Interruptor Termico Mod-4 Sk... | 0.00 | 22404.90 |
| 94 | 94 | Interruptor Termico Mod-5 Sk... | 0.00 | 22841.60 |
| 95 | 95 | Cable Cat 6 Mod-6 Sku-95 | 0.00 | 23282.50 |
| 96 | 96 | Interruptor Termico Mod-7 Sk... | 0.00 | 23727.60 |
| 97 | 97 | Sensor Óptico Mod-8 Sku-97 | 0.00 | 24176.90 |
| 98 | 98 | Malla Industrial Mod-9 Sku-98 | 0.00 | 24630.40 |
| 99 | 99 | Tornillo Titanio Mod-10 Sku-... | 0.00 | 25088.10 |
| 100 | 100 | Válvula Flujo Mod-1 Sku-100 | 0.00 | 25900.00 |

Ln 222, Col 1 (738 selected) Spaces: 4 UTF-8 CRLF 100 rows MSSQL 00:00:00 tcp:jean3.database.windows.net : QhatuPERU (53)

GOExplicación: Se utiliza el operador **PIVOT**. La columna Ano (obtenida de YEAR(FechaSalida)) se convierte en columnas ([2024], [2025]), y la función **SUM(MontoVendido)** rellena los valores.

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

Enunciado: Mostrar la cantidad total de artículos enviados agrupados por mes, utilizando los códigos de tienda como columnas (alternativa a PIVOT usando CASE).

Consulta SQL:

-- 40. Mostrar Mes y columnas por tienda (CASE alternative).

```
SELECT
    DATENAME(MONTH, GE.FechaSalida) AS Mes,
    SUM(CASE WHEN GE.CodTienda = 101 THEN
GD.CantidadEnviada ELSE 0 END) AS Total_Tienda_101,
    SUM(CASE WHEN GE.CodTienda = 102 THEN
GD.CantidadEnviada ELSE 0 END) AS Total_Tienda_102,
    SUM(CASE WHEN GE.CodTienda = 103 THEN
GD.CantidadEnviada ELSE 0 END) AS Total_Tienda_103
FROM
    GUIA_ENVIO GE
JOIN
    GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
GROUP BY
    DATENAME(MONTH, GE.FechaSalida),
    MONTH(GE.FechaSalida) -- Agrupa por nombre y número de mes
                             para ordenar
ORDER BY
    MONTH(GE.FechaSalida);
GO
```

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Server Explorer' pane shows the database structure for 'QhātuPERU (jean3)', with 'dbo.ARTICULO' selected. The central pane shows a SQL query being executed in the 'QhātuPERU' database. The query is as follows:

```

243 -- 40. Mostrar Mes y columnas por tienda (CASE alternative).
244 SELECT
245     DATENAME(MONTH, GE.FechaSalida) AS Mes,
246     SUM(CASE WHEN GE.CodTienda = 101 THEN GD.CantidadEnviada ELSE 0 END) AS Total_Tien
247     SUM(CASE WHEN GE.CodTienda = 102 THEN GD.CantidadEnviada ELSE 0 END) AS Total_Tien
248     SUM(CASE WHEN GE.CodTienda = 103 THEN GD.CantidadEnviada ELSE 0 END) AS Total_Tien
249 FROM
250     GUIA_ENVIO GE
251 JOIN
252     GUIA_DETALLE GD ON GE.NumGuia = GD.NumGuia
253 GROUP BY
254     DATENAME(MONTH, GE.FechaSalida)

```

Below the query editor, the 'Results' pane shows the output of the query:

| | Mes | Total_Tienda_101 | Total_Tienda_102 | Total_Tienda_103 |
|---|-----------|------------------|------------------|------------------|
| 1 | September | 0 | 0 | 0 |
| 2 | October | 53 | 56 | 59 |

The status bar at the bottom indicates the current position is at line 244, column 1, with 638 characters selected. It also shows the encoding as UTF-8, the number of rows as 2, and the connection details as 'tcp:jean3.database.windows.net : QhātuPERU (53)'.

Explicación: Se utiliza la **Agregación Condicional** (SUM(CASE WHEN...)). Se agrupa el resultado por mes y se suma la CantidadEnviada solo si el CodTienda coincide con el valor de la columna que se desea crear. Se incluye MONTH(GE.FechaSalida) en el ORDER BY para asegurar que los meses aparezcan en orden cronológico y no alfabético.