

PROYECTO ANALISIS DE DATOS

“Respondiendo a requisitos de negocio”

HERRAMIENTAS UTILIZADAS



SQL Server



SSMS Management Studio

Autor: Daniel Villar Rodríguez

-- La empresa comercial "xx" me ha contratado como analista de datos T-SQL y necesita
-- respuestas a sus requerimientos de negocio:

-- 1. Se necesitan obtener una concatenación de Mes y Año sobre la tabla Ventas.Ordenes, que
cuenta con las siguientes columnas:

OBJETIVOS

- • id_orden, con alias codOrden.
- • id_cliente, sin alias.
- • ciudadtransporte, con alias ciudad de transporte.
- • paistransporte con alias país.
- • Agregar una columna llamada MesAño_Orden, que concatene el mes y el año de orden

SCRIPT

```
USE BD_SQL
```

```
SELECT
```

```
id_orden AS codOrden,
```

```
id_cliente,
```

```
ciudadtransporte AS "Ciudad de transporte",
```

```
paistransporte AS País,
```



```
CONCAT (MONTH (fecha_orden),'-', YEAR (fecha_orden)) AS MesAñoOrden
```

```
FROM
```

```
Ventas.Ordenes
```

```
;
```

RESULTADO

 Results  Messages					
	codOrden	id_cliente	Ciudad de transporte	País	MesAñoOrden
1	10248	85	Reims	Francia	7-2020
2	10249	79	Münster	Alemania	7-2020
3	10250	34	Rio de Janeiro	Brasil	7-2020
4	10251	84	Lyon	Francia	7-2020
5	10252	76	Charleroi	Bélgica	7-2020
6	10253	34	Rio de Janeiro	Brasil	7-2020
7	10254	14	Bern	Suiza	7-2020
8	10255	68	Genève	Suiza	7-2020
9	10256	88	Resende	Brasil	7-2020
10	10257	35	San Cristóbal	Venezuela	7-2020
11	10258	20	Graz	Austria	7-2020
12	10259	13	Mexico D.F.	Mexico	7-2020

-- 2. Se requiere obtener la edad exacta de cada empleado de la tabla RRHH.Empleados con las siguientes columnas:

OBJETIVOS

- • id_empleado.
- • nombre y apellido, que sea la concatenación de las columnas nombre y columna apellido.
- • edad, sea que la diferencia entre el año de la fecha de nacimiento y el año de contrato del empleado.

SCRIPT

```
SELECT
  id_empleado,
  CONCAT (nombre, ' ', apellido) AS NomApellido,
  DATEDIFF (YEAR, fecha_nac, GETDATE()) AS Edad
FROM
  RRHH.Empleados
;
```

RESULTADO

Results		Messages	
	id_empleado	NomApellido	Edad
1	1	María García	56
2	2	Juan Martínez	52
3	3	Ana López	41
4	4	José Fernández	67
5	5	Luis Díaz	49
6	6	Laura González	41
7	7	Carlos Ruíz	44
8	8	María Hernández	46
9	9	Patricia Pérez	38

-- 3. Se requiere obtener un resultado de fechas concretas para determinados clientes y ordenes de pedido con los siguientes objetivos:

OBJETIVOS

- • Que cuente con las columnas id_orden, fecha_orden, fecha_transporte e id_cliente.
- • Que filtre solo los registros de abril 2022 (por fecha_orden).

SCRIPT

```
SELECT
  id_orden,
  fecha_orden,
  fecha_transporte,
  id_cliente
FROM
  Ventas.Ordenes
WHERE
  fecha_orden BETWEEN '20220401' AND '20220430'
;
```

RESULTADO

	id_orden	fecha_orden	fecha_transporte	id_cliente
1	10990	2022-04-01	2022-04-07	20
2	10991	2022-04-01	2022-04-07	63
3	10992	2022-04-01	2022-04-03	77
4	10993	2022-04-01	2022-04-10	24
5	10994	2022-04-02	2022-04-09	83
6	10995	2022-04-02	2022-04-06	58
7	10996	2022-04-02	2022-04-10	63
8	10997	2022-04-03	2022-04-13	46
9	10998	2022-04-03	2022-04-17	91
10	10999	2022-04-03	2022-04-10	56
11	11000	2022-04-06	2022-04-14	65
12	11001	2022-04-06	2022-04-14	24

-- 4. Se quiere conocer los datos de los importes totales con unos requisitos determinados:

OBJETIVOS

- • Que cuente con todas las columnas, y agregando el importe total.
- • Que filtre solo los registros con un importe mayor a 10000, o que tengan un precio unitario mayor a 250.

SCRIPT

```
SELECT
  *,
  (preciounitario * cantidad) AS ImporteTotal
FROM
  Ventas.DetalleOrden
WHERE
  (preciounitario * cantidad) > 10000
OR
  preciounitario > 250
;
```

RESULTADO

Results Messages							
	id_orden	id_producto	preciounitario	cantidad	descuento	flete_proporcional	ImporteTotal
1	10353	38	210,80	50	0.200	180.32	10540,00
2	10417	38	210,80	50	0.000	17.57	10540,00
3	10424	38	210,80	49	0.200	123.54	10329,20
4	10518	38	263,50	15	0.000	72.72	3952,50
5	10540	38	263,50	30	0.000	251.91	7905,00
6	10541	38	263,50	4	0.100	17.16	1054,00
7	10616	38	263,50	15	0.050	29.13	3952,50
8	10672	38	263,50	15	0.100	47.88	3952,50
9	10783	38	263,50	5	0.000	62.49	1317,50
10	10805	38	263,50	10	0.000	118.67	2635,00
11	10816	38	263,50	30	0.050	359.89	7905,00

-- 5. Se quieren obtener datos concretos a una determinada fecha:

OBJETIVOS

- • Que cuente con las columnas fecha_orden, fecha_transporte, paistransporte, flete e id_transportista.
- • La fecha_orden debe estar entre el 01 de Enero de 2021 y el 06 de Mayo de 2021.

SCRIPT

```
SELECT
    fecha_orden,
    fecha_transporte,
    paistransporte,
    flete,
    id_transportista
FROM
    Ventas.Ordenes
WHERE
    fecha_orden BETWEEN '20210101' AND '20210506'
;
```

RESULTADO

	fecha_orden	fecha_transporte	paistransporte	flete	id_transportista
1	2021-01-01	2021-01-16	Reino Unido	83,93	3
2	2021-01-01	2021-01-10	USA	12,51	1
3	2021-01-02	2021-01-10	Austria	67,88	2
4	2021-01-03	2021-01-09	Austria	73,79	3
5	2021-01-03	2021-01-08	Italia	155,97	1
6	2021-01-06	2021-01-22	Venezuela	34,82	1
7	2021-01-07	2021-01-13	Brasil	108,04	1
8	2021-01-07	2021-01-30	Alemania	91,48	2
9	2021-01-08	2021-01-14	Francia	11,26	1
10	2021-01-09	2021-01-14	Argentina	29,83	1
11	2021-01-10	2021-01-15	Canadá	2,40	3
12	2021-01-10	2021-01-21	Canadá	23,65	3

-- 6. Se requieren datos concretos entre fechas:

OBJETIVOS

- • Que cuente con las columnas fecha_requerida, fecha_transporte, paistransporte, flete e id_transportista.
- • La fecha requerida debe ser menor a la fecha de transporte (incorporar registros que aún no se transportan).

SCRIPT

```
SELECT
    fecha_requerida,
    fecha_transporte,
    paistransporte,
    flete,
    id_transportista
FROM
    Ventas.Ordenes
WHERE
    fecha_requerida < fecha_transporte
    OR fecha_transporte IS NULL
;
```

RESULTADO

	fecha_requerida	fecha_transporte	paistransporte	flete	id_transportista
31	2022-01-26	2022-02-06	Francia	63,54	2
32	2022-01-27	2022-02-04	Argentina	90,85	1
33	2022-02-05	2022-02-10	USA	487,57	3
34	2022-04-01	2022-04-08	Suecia	151,52	2
35	2022-04-02	2022-04-08	Francia	19,79	1
36	2022-04-02	2022-04-08	Venezuela	2,08	1
37	2022-04-07	2022-04-24	España	16,16	1
38	2022-05-06	NULL	Austria	79,46	3
39	2022-05-11	NULL	Argentina	3,17	3
40	2022-05-19	NULL	Venezuela	65,00	2
41	2022-05-20	NULL	USA	18,84	3
42	2022-05-21	NULL	Canadá	70,58	2

-- 7. Se quieren conocer datos muy precisos sobre los clientes:

OBJETIVOS

- • Que cuente con las columnas id_cliente, nombrecliente, direccion, país y ciudad.
- • La dirección debe empezar por consonante (no vocal) y debe terminar en un número.

SCRIPT

```
SELECT
  id_cliente,
  nombrecliente,
  direccion,
  pais,
  ciudad
FROM
  Ventas.Clientes
WHERE
  direccion LIKE '[^aeiou]%' AND direccion LIKE '%[0-9]'
;
```

RESULTADO

	id_cliente	nombrecliente	direccion	pais	ciudad
1	3	Cliente KBUDE	Mataderos 7890	Mexico	Mexico D.F.
2	5	Cliente HGVLZ	Berguvsvägen 5678	Suecia	Luleå
3	6	Cliente XHXJV	Forsterstr. 7890	Alemania	Mannheim
4	8	Cliente QUHWH	C/ Araquil, 0123	España	Madrid
5	11	Cliente UBHAU	Fauntleroy Circus 4567	Reino Unido	Londres
6	12	Cliente PSNMQ	Cerrito 3456	Argentina	Buenos Aires
7	13	Cliente VMLOG	Sierras de Granada 7890	Mexico	Mexico D.F.
8	14	Cliente WNMAF	Hauptstr. 0123	Suiza	Berna
9	17	Cliente FEVNN	Walserweg 4567	Alemania	Aquisgrán
10	21	Cliente KIDPX	Rua Orós, 3456	Brasil	Sao Paulo
11	22	Cliente DTD MN	C/ Moralarzal, 5678	España	Madrid

-- 8. Se necesitan saber datos concretos sobre los teléfonos de los clientes:

OBJETIVOS

- • Que cuente con las columnas id_clientes, nombrecliente, nombrecontacto, ciudad, país, teléfono.
- • El valor de la columna país debe terminar con la letra “a”.
- • El teléfono debe tener el siguiente patrón: Los primeros números deben estar formados por 2 o 3 caracteres, seguidos de un guion.

SCRIPT

```
SELECT
id_cliente,
nombrecliente,
nombrecontacto,
ciudad,
pais,
telefono
FROM
Ventas.Clientes
WHERE
pais LIKE '%a'
AND telefono LIKE '__-%' OR telefono LIKE '____-%'
;
```

RESULTADO

	id_cliente	nombrecliente	nombrecontacto	ciudad	pais	telefono
1	1	Cliente NRZBB	Allen, Michael	Berlín	Alemania	030-3456789
2	25	Cliente AZJED	Carlson, Jason	München	Alemania	089-8901234
3	27	Cliente WMFEA	Schmöllerl, Martin	Torino	Italia	011-2345678
4	44	Cliente OXFRU	Louverdis, George	Frankfurt a.M.	Alemania	069-7890123
5	49	Cliente CQRAA	Duerr, Bernard	Bergamo	Italia	035-345678
6	70	Cliente TMXGN	Makovac, Zrinka	Stavern	Noruega	07-89 01 23
7	87	Cliente ZHYOS	Ludwig, Michael	Oulu	Finlandia	981-123456
8	90	Cliente XBBVR	Larsson, Katarina	Helsinki	Finlandia	90-012 3456

-- 9. Se necesita filtrar la tabla Produccion.Categorias donde la descripción tenga exactamente 2 comas en su valor:

SCRIPT

```
SELECT
*
FROM
    Produccion.Categorias
WHERE
    descripcion LIKE '%,%,%'
;
```

RESULTADO

	id_categoria	nombrecategoria	descripcion
1	1	Bebidas	Bebidas suaves, cafés, té, cervezas y ales
2	2	Condimentos	Salsas dulces y saladas, aderezos, spreads y con...
3	5	Cereales/Granos	Pan, galletas, pasta y cereales

-- 10. ¿Cuántos empleados se han contratado cada año?

SCRIPT

```
SELECT
    DATEPART (YEAR, fecha_contrato) AS AñoContrato,
    COUNT (DISTINCT id_empleado) AS Nro_Empleados
FROM
    RRHH.Empleados
GROUP BY
    DATEPART (YEAR, fecha_contrato)
;
```

RESULTADO

	AñoContrato	Nro_Empleados
1	2020	3
2	2021	3
3	2022	3

-- 11. ¿De dónde son nuestros clientes?; se necesita saber el número de las ciudades distintas de cada país al que vendemos productos.

SCRIPT

```
SELECT
    paistransporte,
    COUNT (DISTINCT ciudadtransporte) AS Nro_Ciudades
FROM
    Ventas.Ordenes
GROUP BY
    paistransporte
;
```

RESULTADO

 Results  Messages		
	paistransporte	NroCiudades
1	Alemania	11
2	Argentina	1
3	Austria	2
4	Bélgica	2
5	Brasil	4
6	Canadá	3
7	Dinamarca	2
8	España	3
9	Finland	2
10	Francia	9
11	Irlanda	1

-- 12. Necesitamos enumerar las órdenes y saber el descuento aplicado y el precio máximo unitario al que se le vende.

SCRIPT

```
SELECT
    descuento,
    COUNT (DISTINCT id_orden) AS Nro_Ordenes,
    MAX (preciounitario) AS Precio_Máximo
FROM
    Ventas.DetalleOrden
GROUP BY
    descuento
;
```

RESULTADO

Results		Messages	
	descuento	Nro_Ordenes	Precio_Máximo
1	0.000	613	263,50
2	0.010	1	15,00
3	0.020	1	25,00
4	0.030	1	33,25
5	0.040	1	81,00
6	0.050	87	263,50
7	0.060	1	34,00
8	0.100	75	263,50
9	0.150	75	99,00
10	0.200	73	210,80
11	0.250	72	210,80

-- 13. Del requerimiento anterior, los grupos deben estar multiplicados por 100 para que sea más legible el descuento aplicado y también se necesitan los siguientes requerimientos adicionales:

OBJETIVOS

- • Se requieren sólo los grupos de descuento que tienen más de 1 orden.
- • Saber de aquellos registros que tienen algún descuento.

SCRIPT

```
SELECT
  (descuento * 100) AS descuento,
  COUNT (DISTINCT id_orden) AS Nro_Ordenes,
  MAX (preciounitario) AS Precio_Máximo
FROM
  Ventas.DetalleOrden
WHERE
  descuento IS DISTINCT FROM '0'
GROUP BY
  descuento
HAVING
  COUNT (DISTINCT id_orden) >1
;
```

RESULTADO

	descuento	Nro_Ordenes	Precio_Máximo
1	5.000	87	263,50
2	10.000	75	263,50
3	15.000	75	99,00
4	20.000	73	210,80
5	25.000	72	210,80

-- 14. Necesitamos un detalle de los precios máximos que nos cobra cada proveedor, con las siguientes características adicionales:

OBJETIVOS

- • Conocer los proveedores que tengan más de 2 productos asociados.
- • Únicamente los registros que no están descontinuados (activos).

SCRIPT

```
SELECT
    id_proveedor,
    MAX (preciounitario) AS Precio_Máximo
FROM
    Produccion.Productos
WHERE
    descontinuado = 0
GROUP BY
    id_proveedor
HAVING
    COUNT (*) > 2
;
```

RESULTADO

Results		Messages
	id_proveedor	Precio_Máximo
1	1	19,00
2	2	22,00
3	3	40,00
4	6	23,25
5	7	62,50
6	8	81,00
7	11	43,90
8	12	33,25
9	14	34,80
10	15	36,00
11	16	18,00

-- 15. Necesitamos saber de cada producto vendido, el descuento aplicado y el número de pedidos por cada descuento aplicado:

SCRIPT

```
SELECT
  id_producto,
  descuento,
  COUNT (DISTINCT id_orden) AS Nro_Ordenes
FROM
  Ventas.DetalleOrden
GROUP BY
  id_producto,
  descuento
ORDER BY
  id_producto ASC,
  descuento ASC
;
```

RESULTADO

 Results  Messages

	id_producto	descuento	Nro_Ordenes
1	1	0.000	22
2	1	0.050	1
3	1	0.100	1
4	1	0.150	5
5	1	0.200	4
6	1	0.250	5
7	2	0.000	19
8	2	0.050	3
9	2	0.100	1
10	2	0.150	5
11	2	0.200	10

-- 16. ¿Cuáles son las ciudades de entrega de nuestros productos donde el flete es inferior a 200?:

SCRIPT

```
SELECT
    ciudadtransporte,
    SUM (flete) AS Total_Flete
FROM
    Ventas.Ordenes
GROUP BY
    ciudadtransporte
HAVING
    SUM (flete) < 200
;
```

RESULTADO

 Results  Messages		
	ciudadtransporte	Total_Flete
1	Barcelona	37,98
2	Butte	129,96
3	Caracas	67,80
4	Helsinki	88,41
5	Kirkland	70,01
6	Mannheim	168,26
7	Münster	125,97
8	Paris	108,28
9	Reims	58,41
10	Resende	194,71
11	Torino	75,13

-- 17. Necesitamos conocer el flete que se le aplica al cliente cada año y cada mes de venta , que cumpla los siguientes requisitos adicionales:

OBJETIVOS

- • Se requieren datos desde el 01 de Enero de 2021 en adelante.
- • Para una mejor visualización de los datos, se requiere una ordenación de los resultados de forma ASCENDENTE, por Año y Mes.

SCRIPT

```
SELECT
    DATEPART (YEAR, fecha_orden) AS Año_Orden,
    DATEPART (MONTH, fecha_orden) AS Mes_Orden,
    (SUM (flete) / COUNT (DISTINCT id_cliente)) AS Flete_por_cliente
FROM
    Ventas.Ordenes
WHERE
    fecha_orden > '20210101'
GROUP BY
    DATEPART (YEAR, fecha_orden),
    DATEPART (MONTH, fecha_orden)
ORDER BY
    DATEPART (YEAR, fecha_orden) ASC,
    DATEPART (MONTH, fecha_orden) ASC
;
```

RESULTADO

 Results  Messages

	Año_Orden	Mes_Orden	Flete_por_cliente
4	2021	4	108,8555
5	2021	5	133,1307
6	2021	6	74,106
7	2021	7	98,3488
8	2021	8	102,609
9	2021	9	119,8907
10	2021	10	140,9117
11	2021	11	74,4018
12	2021	12	101,5664
13	2022	1	133,2546
14	2022	2	122,084

-- 18. Se necesitan conocer unos requisitos en concreto sobre la tabla Ventas.Clientes:

OBJETIVOS

- • Que contenga las columnas: id_cliente, nombrecliente, ciudad, pais y región.
- • Que el resultado muestre las regiones ordenadas de forma ASCENDENTE, y debajo aparezcan todos los valores NULLs.

SCRIPT

```
SELECT
id_cliente,
nombrecliente,
ciudad,
pais,
region
FROM
Ventas.Clientes
ORDER BY
CASE WHEN region IS NULL THEN 1 ELSE 0 END,
region ASC
;
```

RESULTADO

	id_cliente	nombrecliente	ciudad	pais	region
25	62	Cliente WFIZJ	Sao Paulo	Brasil	SP
26	81	Cliente YQQ...	Sao Paulo	Brasil	SP
27	88	Cliente SRQV...	Resende	Brasil	SP
28	35	Cliente UMTL...	San Cristóbal	Venezuela	Táchira
29	43	Cliente UISOJ	Walla Walla	USA	WA
30	89	Cliente YBQTI	Seattle	USA	WA
31	82	Cliente EYHKM	Kirkland	USA	WA
32	75	Cliente XOJYP	Lander	USA	WY
33	76	Cliente SFOG...	Charleroi	Bélgica	NULL
34	72	Cliente AHPOP	Londres	Reino Unido	NULL
35	73	Cliente JMIKW	Kobenhavn	Dinamarca	NULL
36	74	Cliente YSHYL	Paris	Francia	NULL

-- 19. ¿Cuántos productos tenemos en activo y cuantos discontinuados? Sólo se requieren datos por categorías.

SCRIPT

```
SELECT
  id_categoria,
  SUM (CASE WHEN discontinuado = 0 THEN 1 ELSE 0 END) as NoDescontinuado,
  SUM (CASE WHEN discontinuado = 1 THEN 1 ELSE 0 END) as Descontinuado
FROM
  Produccion.Productos
GROUP BY
  id_categoria
;
```

RESULTADO

Results		Messages	
	id_categoria	NoDescontinuado	Descontinuado
1	1	11	1
2	2	11	1
3	3	13	0
4	4	10	0
5	5	6	1
6	6	2	4
7	7	4	1
8	8	12	0

-- 20. Queremos un cuadro de doble entrada para saber las veces que damos un determinado descuento al cliente según un baremo de agrupación de precios unitarios, con las siguientes características:

OBJETIVOS

- • Agrupar la tabla por el descuento aplicado.
- • Mostrar el número de registros por grupos de preciounitario ([0 a 20>, [20 a 50>, [50 a 80>, [80 a más>).
- • Agregar una columna extra con el Total.

SCRIPT

```
SELECT
    descuento,
    SUM(CASE WHEN preciounitario <=20 THEN 1 ELSE 0 END) AS '<20',
    SUM(CASE WHEN preciounitario >20 AND preciounitario <=50 THEN 1 ELSE 0 END) AS '21-50',
    SUM(CASE WHEN preciounitario >50 AND preciounitario <=80 THEN 1 ELSE 0 END) AS '51-80',
    SUM(CASE WHEN preciounitario >80 THEN 1 ELSE 0 END) AS '>80',
    COUNT (*) AS Total
FROM
    Ventas.DetalleOrden
GROUP BY
    descuento
;
```

RESULTADO

Results Messages						
	descuento	<20	21-50	51-80	>80	Total
1	0.000	732	488	56	41	1317
2	0.100	101	60	7	5	173
3	0.050	98	65	11	11	185
4	0.040	0	0	0	1	1
5	0.060	0	1	0	0	1
6	0.150	93	57	5	2	157
7	0.030	1	2	0	0	3
8	0.010	1	0	0	0	1
9	0.020	1	1	0	0	2
10	0.250	83	55	8	8	154
11	0.200	100	53	5	3	161

-- 21. ¿Cuál es el flete máximo que nos aplican?:

OBJETIVOS

- • Agrupar el resultado en base al mes de la fecha_orden.
- • Agregar tres columnas: 2020, 2021 y 2022, que muestre el flete máximo de cada uno de estos años.
- • Que el resultado esté ordenado por mes.

SCRIPT

```
SELECT
    DATEPART (MONTH, fecha_orden) AS Mes,
    MAX (CASE WHEN YEAR (fecha_orden) = 2020 THEN flete ELSE 0 END) AS "2020",
    MAX (CASE WHEN YEAR (fecha_orden) = 2021 THEN flete ELSE 0 END) AS "2021",
    MAX (CASE WHEN YEAR (fecha_orden) = 2022 THEN flete ELSE 0 END) AS "2022"
FROM
    Ventas.Ordenes
GROUP BY
    DATEPART (MONTH, fecha_orden)
ORDER BY
    DATEPART (MONTH, fecha_orden) ASC
;
```

RESULTADO

Results		Messages		
	Mes	2020	2021	2022
1	1	0,00	458,78	719,78
2	2	0,00	189,09	603,54
3	3	0,00	708,95	657,54
4	4	0,00	789,95	830,75
5	5	0,00	1007,64	258,64
6	6	0,00	252,49	0,00
7	7	208,58	544,08	0,00
8	8	229,24	487,38	0,00
9	9	257,62	388,98	0,00
10	10	214,27	810,05	0,00
11	11	360,63	243,73	0,00
12	12	890,78	351,53	0,00

-- 22. Queremos encriptar el teléfono de los clientes para que solo se vean los dos últimos dígitos, con las siguientes características:

OBJETIVOS

- • Que contenga las columnas: id_cliente, nombrecliente, ciudad, pais, telefono.
- • Agregar una columna calculada llamada teléfono_encriptado, que solo muestre los últimos 2 dígitos, al resto ponerle '*'.

SCRIPT

```
SELECT
  id_cliente,
  nombrecliente,
  ciudad,
  pais,
  telefono,
  -- busco la longitud de los caracteres a encriptar -> LEN (telefono)-2,
  -- los encripto -> REPLICATE ('*', LEN (telefono)-2),
  -- busco los caracteres visibles -> RIGHT ( telefono, 2),
  -- se concatena para llegar al resultado:
  REPLICATE ('*', LEN (telefono)-2)+RIGHT ( telefono, 2) AS Teléfono_encriptado
FROM
  Ventas.Clientes
;
```

RESULTADO

Results Messages						
	id_cliente	nombrecliente	ciudad	pais	telefono	Teléfono_encriptado
1	1	Cliente NRZBB	Berlín	Alemania	030-3456789	*****89
2	2	Cliente MLTDN	Mexico D.F.	Mexico	(5) 789-0123	*****23
3	3	Cliente KBUDE	Mexico D.F.	Mexico	(5) 123-4567	*****67
4	4	Cliente HFBZG	Londres	Reino Unido	(171) 456-7890	*****90
5	5	Cliente HGVLZ	Luleå	Suecia	0921-67 89 01	*****01
6	6	Cliente XHXJV	Mannheim	Alemania	0621-67890	*****90
7	7	Cliente QXVLA	Estrasburgo	Francia	67.89.01.23	*****23
8	8	Cliente QUHWH	Madrid	España	(91) 345 67 89	*****89
9	9	Cliente RTXGC	Marsella	Francia	23.45.67.89	*****89
10	10	Cliente EEALV	Tsawassen	Canadá	(604) 901-2345	*****45
11	11	Cliente UBHAU	Londres	Reino Unido	(171) 789-0123	*****23
12	12	Cliente PSNMO	Buenos Aires	Argentina	(11) 800-1234	*****34

-- 23. Queremos codificar los clientes según su nombre y apellido, con los siguientes requerimientos:

OBJETIVOS

- • Que contenga las columnas: id_cliente, nombrecliente, ciudad, pais y nombrecontacto.
- • Agregar una columna calculada llamada Código basada en el campo nombrecontacto, que concatene los primeros 2 caracteres del apellido y los 2 primeros 2 caracteres del nombre (TODO EN MAYUSCULAS)

SCRIPT

```
SELECT
id_cliente,
nombrecliente,
ciudad,
pais,
nombrecontacto,
-- busco los dos primeros caracteres del apellido -> LEFT (nombrecontacto, 2),
-- busco la posición donde empieza los caracteres del nombre -> CHARINDEX(',',
nombrecontacto) + 2
-- extraigo los dos caracteres del nombre -> SUBSTRING (nombrecontacto, CHARINDEX(',',
nombrecontacto) + 2, 2)
-- concatenamos el resultado obtenido y lo ponemos en mayúsculas:
UPPER (LEFT (nombrecontacto, 2)+SUBSTRING (nombrecontacto, CHARINDEX(',',
nombrecontacto) + 2, 2)) AS Código
FROM
Ventas.Clientes
;
```

RESULTADO

	id_cliente	nombrecliente	ciudad	pais	nombrecontacto	Código
1	1	Cliente NRZBB	Berlín	Alemania	Allen, Michael	ALMI
2	2	Cliente MLTDN	Mexico D.F.	Mexico	Hassall, Mark	HAMA
3	3	Cliente KBUDE	Mexico D.F.	Mexico	Strome, David	STDA
4	4	Cliente HFBZG	Londres	Reino Unido	Cunningham, Conor	CUCO
5	5	Cliente HGVLZ	Luleå	Suecia	Higginbotham, Tom	HITO
6	6	Cliente XHXJV	Mannheim	Alemania	Polonia, Carole	POCA
7	7	Cliente QXVLA	Estrasburgo	Francia	Bansal, Dushyant	BADU
8	8	Cliente QUHWH	Madrid	España	Ilyina, Julia	ILJU
9	9	Cliente RTXGC	Marsella	Francia	Raghav, Amritansh	RAAM
10	10	Cliente EEALV	Tsawassen	Canadá	Culp, Scott	CUSC
11	11	Cliente UBHAU	Londres	Reino Unido	Jaffe, David	JADA

-- 24. Necesitamos codificar los pedidos con los siguientes requisitos:

OBJETIVO

- • Que contenga las columnas: fecha_orden, id_cliente, paistransporte e id_orden.
- • Agregar una columna calculada llamada cod_orden que rellene ceros a la izquierda, de tal manera que el resultado siempre tenga 8 caracteres.

SCRIPT

```
SELECT
    fecha_orden,
    id_cliente,
    paistransporte,
    id_orden,
    RIGHT (CONCAT('00000000',id_orden), 8) AS cod_orden
FROM
    Ventas.Ordenes
;
```

RESULTADO

	fecha_orden	id_cliente	paistransporte	id_orden	cod_orden
1	2020-07-04	85	Francia	10248	00010248
2	2020-07-05	79	Alemania	10249	00010249
3	2020-07-08	34	Brasil	10250	00010250
4	2020-07-08	84	Francia	10251	00010251
5	2020-07-09	76	Bélgica	10252	00010252
6	2020-07-10	34	Brasil	10253	00010253
7	2020-07-11	14	Suiza	10254	00010254
8	2020-07-12	68	Suiza	10255	00010255
9	2020-07-15	88	Brasil	10256	00010256
10	2020-07-16	35	Venezuela	10257	00010257
11	2020-07-17	20	Austria	10258	00010258

-- 25. Queremos obtener una consulta con la fecha de inicio y la fecha de cierre de la semana de los pedidos de los clientes, con los siguientes requisitos:

OBJETIVOS

- • Que tenga las columnas id_orden y fecha_orden.
- • Agregar una columna con el número del día de la semana.
- • Agregar dos columnas: inicio_semana, que muestre la fecha donde inicia la semana (las semanas deben empezar los lunes) y fin_semana, que muestre la fecha donde finaliza la semana (las semanas deben terminar los domingos).

SCRIPT

```
SELECT
  id_orden,
  fecha_orden,
  DATEPART (WEEKDAY, fecha_orden) AS Nro_dia_sem,
  DATEADD (DAY, -(DATEPART (WEEKDAY, fecha_orden)-1), fecha_orden) AS Inicio_semana,
  DATEADD (DAY, 7+(-(DATEPART (WEEKDAY, fecha_orden))), fecha_orden) AS Final_semana
FROM
  Ventas.Ordenes
;
```

RESULTADO

	id_orden	fecha_orden	Nro_dia_sem	Inicio_semana	Final_semana
7	10254	2020-07-11	7	2020-07-05	2020-07-11
8	10255	2020-07-12	1	2020-07-12	2020-07-18
9	10256	2020-07-15	4	2020-07-12	2020-07-18
10	10257	2020-07-16	5	2020-07-12	2020-07-18
11	10258	2020-07-17	6	2020-07-12	2020-07-18
12	10259	2020-07-18	7	2020-07-12	2020-07-18
13	10260	2020-07-19	1	2020-07-19	2020-07-25
14	10261	2020-07-19	1	2020-07-19	2020-07-25
15	10262	2020-07-22	4	2020-07-19	2020-07-25
16	10263	2020-07-23	5	2020-07-19	2020-07-25
17	10264	2020-07-24	6	2020-07-19	2020-07-25

-- 26. ¿Cuáles son los pedidos de los tres últimos días de cada mes?, con el siguiente requisito adicional:

OBJETIVO

-- • Que tenga las columnas id_orden, id_cliente, fecha_orden.

SCRIPT

```
SELECT
    id_orden,
    id_cliente,
    fecha_orden
FROM
    Ventas.Ordenes
WHERE
    fecha_orden BETWEEN DATEADD (DAY, -2, EOMONTH (fecha_orden, 0)) AND EOMONTH
(fecha_orden, 0)
;
```

RESULTADO

	id_orden	id_cliente	fecha_orden
1	10267	25	2020-07-29
2	10268	33	2020-07-30
3	10269	89	2020-07-31
4	10293	80	2020-08-29
5	10294	65	2020-08-30
6	10317	48	2020-09-30
7	10340	9	2020-10-29
8	10341	73	2020-10-29
9	10342	25	2020-10-30
10	10343	44	2020-10-31
11	10366	29	2020-11-28
12	10367	82	2020-11-28

-- 27. Queremos saber el flete promedio de nuestros clientes de Alemania y Brasil, con los siguientes requisitos adicionales:

OBJETIVOS

- • Solo queremos registros de 2021 en adelante.
- • Agrupar y ordenar los resultados en base al Año y Semana de la fecha de orden.
- • Solo los resultados con aquellas semanas y años donde ambos países tienen flete promedio.

SCRIPT

```
SELECT
  DATEPART (YEAR, fecha_orden) AS Año,
  DATEPART (WEEK, fecha_orden) AS Sem_del_año,
  AVG (CASE WHEN paistransporte = 'Alemania' THEN flete END) AS Alemania,
  AVG (CASE WHEN paistransporte = 'Brasil' THEN flete END) AS Brasil
FROM
  Ventas.Ordenes
WHERE
  fecha_orden > '2021'
GROUP BY
  DATEPART (YEAR, fecha_orden),
  DATEPART (WEEK, fecha_orden)
HAVING
  AVG (CASE WHEN paistransporte = 'Alemania' THEN flete END) IS NOT NULL
  AND AVG (CASE WHEN paistransporte = 'Brasil' THEN flete END) IS NOT NULL
ORDER BY
  DATEPART (YEAR, fecha_orden),
  DATEPART (WEEK, fecha_orden)
;
```

RESULTADO

Results Messages				
	Año	Sem_del_año	Alemania	Brasil
1	2021	2	91,48	108,04
2	2021	4	17,55	55,95
3	2021	8	101,885	68,66
4	2021	13	4,93	71,07
5	2021	15	22,53	46,77
6	2021	17	155,06	3,53
7	2021	21	509,295	68,65
8	2021	24	36,65	60,43
9	2021	26	101,615	3,01
10	2021	27	111,19	37,965
11	2021	30	27,79	79,40
12	2021	32	97,18	50,07

-- 28. Necesitamos información acerca el precio máximo y el precio mínimo por cada categoría de producto que compramos y vendemos.

SCRIPT

```
SELECT
  a.id_categoria,
  CONCAT ('[' , MIN (b.preciounitario),'-', MAX (b.preciounitario), '>') AS
Rangos_precios_compra_unit,
  CONCAT ('[' , MIN (a.preciounitario),'-', MAX (a.preciounitario), '>') AS Rangos_precios_vta_unit
FROM
  Produccion.Productos a
JOIN
  Ventas.DetalleOrden b
ON
  a.id_producto = b.id_producto
GROUP BY
  a.id_categoria
ORDER BY
  a.id_categoria ASC
;
```

RESULTADO

Results		Messages	
	id_categoria	Rangos_precios_compra_unit	Rangos_precios_vta_unit
1	1	[3.60-263.50>	[4.50-263.50>
2	2	[8.00-43.90>	[10.00-43.90>
3	3	[7.30-81.00>	[9.20-81.00>
4	4	[2.00-55.00>	[2.50-55.00>
5	5	[5.60-38.00>	[7.00-38.00>
6	6	[5.90-123.79>	[7.45-123.79>
7	7	[8.00-53.00>	[10.00-53.00>
8	8	[4.80-62.50>	[6.00-62.50>

-- 29. Queremos ver la cantidad de pedidos y los que nos cuesta el flete promedio en lo que se refiere a fechas límite de entrega de productos que nos indica el cliente.

OBJETIVOS

- • queremos un informe donde se agrupen las fechas de entrega en formato YYYYMM.
- • sabemos que el flete sale con muchos decimales, sólo queremos verlo a 2 decimales.

SCRIPT

```
SELECT
  CONVERT (varchar(6), fecha_requerida, 112) AS Fecha_entrega,
  COUNT (DISTINCT id_orden) AS Nro_Ordenes,
  CONVERT (varchar, AVG (flete), 1) AS Flete_promedio
FROM
  Ventas.Ordenes
GROUP BY
  CONVERT (varchar(6), fecha_requerida, 112)
;
```

RESULTADO

	Fecha_entrega	Nro_Ordenes	Flete_promedio
1	202007	1	58.17
2	202008	25	58.41
3	202009	22	53.10
4	202010	27	47.94
5	202011	23	59.70
6	202012	24	93.44
7	202101	34	88.52
8	202102	27	65.07
9	202103	31	51.33
10	202104	33	66.50
11	202105	32	97.18
12	202106	29	118.19

30. Queremos ver el importe de venta neto de cada orden

OBJETIVOS

-- • que en el listado nos aparezcan los datos con los separadores de miles y el símbolo de € para una mejor visualización.

SCRIPT

```
SELECT
    id_orden,
    id_producto,
    cantidad,
    CONCAT (preciounitario, ' €') AS PrecioUnitario,
    -- preciouunitario * cantidad * (1-descuento) AS Importe_venta_netto,
    -- CONVERT (money, descuento)-> se convierten los datos de la columna descuento en
    formato money,
    -- CONVERT (varchar(6),CONVERT (money, descuento *100)) AS descuento,
    (CASE WHEN (CONCAT (SUBSTRING((CONVERT (varchar(6),CONVERT (money, descuento
    *100))),
    (CHARINDEX('.', CONVERT (varchar(6),CONVERT (money, descuento *100))) -2),2),' %')) = '0 %'
    THEN 'N/A' ELSE (CONCAT (SUBSTRING((CONVERT (varchar(6),CONVERT (money, descuento
    *100))),
    (CHARINDEX('.', CONVERT (varchar(6),CONVERT (money, descuento *100))) -2),2),' %')) END)
    AS Dcto_aplicado,
    CONCAT ((CONVERT (varchar, preciouunitario * cantidad * (1-CONVERT (money, descuento))
    ,1)), ' €') AS Importe_venta_netto
FROM
    Ventas.DetalleOrden
;
```

RESULTADO

	id_orden	id_producto	cantidad	PrecioUnitario	Dcto_aplicado	Importe_venta_netto
1	10248	11	12	14.00 €	N/A	168.00 €
2	10248	42	10	9.80 €	N/A	98.00 €
3	10248	72	5	34.80 €	N/A	174.00 €
4	10249	14	9	18.60 €	N/A	167.40 €
5	10249	51	40	42.40 €	N/A	1,696.00 €
6	10250	41	10	7.70 €	N/A	77.00 €
7	10250	51	35	42.40 €	15 %	1,261.40 €
8	10250	65	15	16.80 €	15 %	214.20 €
9	10251	22	6	16.80 €	5 %	95.76 €
10	10251	57	15	15.60 €	5 %	222.30 €
11	10251	65	20	16.80 €	N/A	336.00 €
12	10252	20	40	64.80 €	5 %	2 462.40 €

31. Se requiere información sobre el número de pedidos de clientes por cada porcentaje de descuento.

OBJETIVOS

-- • que tenga un formato de porcentaje legible a simple vista.

SCRIPT

```
SELECT
    CONCAT((CONVERT (varchar, CONVERT (decimal (5,0), (descuento * 100)))),' %') AS Porc_Dcto,
    COUNT (DISTINCT id_orden) AS Nro_Pedidos
FROM
    Ventas.DetalleOrden
GROUP BY
    descuento
ORDER BY
    descuento ASC
;
```

RESULTADO

Results Messages		
	Porc_Dcto	Nro_Pedidos
1	0 %	613
2	1 %	1
3	2 %	1
4	3 %	1
5	4 %	1
6	5 %	87
7	6 %	1
8	10 %	75
9	15 %	75
10	20 %	73
11	25 %	72

32. Se requiere información acerca de las cantidades vendidas agrupadas por el nombre de categoría.

SCRIPT

```
SELECT
  T3.nombrecategoria,
  SUM (T1.cantidad) AS Total_Cantidad
FROM
  Ventas.DetalleOrden T1
  INNER JOIN
  Produccion.Productos T2
  ON
  T1.id_producto = T2.id_producto
  INNER JOIN
  Produccion.Categorias T3
  ON
  T2.id_categoria = T3.id_categoria
GROUP BY
  T3.nombrecategoria
;
```

RESULTADO

 Results  Messages		
	nombrecategoria	Total_Cantidad
1	Productos Frescos	2990
2	Mariscos	7681
3	Condimentos	5298
4	Bebidas	9532
5	Carne/Aves	4199
6	Dulces	7906
7	Cereales/Granos	4562
8	Productos Lácteos	9149

33. ¿Cuánto gastamos en flete según el destino de envío?

OBJETIVOS

-- • Necesitamos la información agrupada por país y ciudad del cliente, detallando el nombre de transportista y que refleje con un total de flete por país y ciudad.

SCRIPT

```
SELECT
  a.paistransporte,
  b.ciudad,
  SUM (CASE WHEN c.compañia = 'Transportista GVSUA' THEN flete ELSE 0 END) AS GVSUA,
  SUM (CASE WHEN c.compañia = 'Transportista ZHISH' THEN flete ELSE 0 END) AS ZHISN,
  SUM (CASE WHEN c.compañia = 'Transportista ETYRN' THEN flete ELSE 0 END) AS ETYRN,
  SUM (CASE WHEN c.compañia = 'Transportista GVSUA' THEN flete ELSE 0 END)+
  SUM (CASE WHEN c.compañia = 'Transportista ZHISH' THEN flete ELSE 0 END)+
  SUM (CASE WHEN c.compañia = 'Transportista ETYRN' THEN flete ELSE 0 END) AS Total
FROM
  Ventas.Ordenes a
  INNER JOIN
  Ventas.Clientes b
  ON
  a.id_cliente = b.id_cliente
  INNER JOIN
  ventas.Transportistas c
  ON
  a.id_transportista = c.id_transportista
GROUP BY
  a.paistransporte,
  b.ciudad
ORDER BY
  Total DESC
;
```

RESULTADO

Results Messages						
	paistransporte	ciudad	GVSUA	ZHISN	ETYRN	Total
1	Austria	Rio de Janeiro	2004,74	0,00	2611,76	4616,50
2	USA	Boise	1389,62	0,00	3200,80	4590,42
3	Alemania	Cunewalde	1746,33	0,00	1672,57	3418,90
4	Brasil	Sao Paulo	708,32	0,00	1598,69	2307,01
5	Irlanda	Cork	262,71	0,00	1890,41	2153,12
6	Reino Unido	Londres	578,83	0,00	1097,60	1676,43
7	Brasil	Rio de Janeiro	680,40	0,00	836,37	1516,77
8	Suecia	Bräcke	779,26	0,00	672,40	1451,66
9	Suecia	Luleå	189,44	0,00	1074,51	1263,95
10	Venezuela	San Cristóbal	454,59	0,00	612,89	1067,48
11	Canadá	Montréal	154,68	0,00	903,53	1058,21
12	Alemania	Frankfurt a M	136,57	0,00	518,15	654,72

34. Se precisa información de todos los clientes y el detalle de sus pedidos.



OBJETIVOS

-- • se requiere que se vean primeros los clientes que no han comprado nada.

SCRIPT

```
SELECT
  T1.id_cliente,
  T2.id_orden
FROM
  Ventas.Clientes T1
FULL OUTER JOIN
  ventas.Ordenes T2
ON T1.id_cliente = T2.id_cliente
ORDER BY
  CASE WHEN T2.id_orden IS NULL THEN 0 ELSE 1 END,
  T1.id_cliente ASC
;
```

RESULTADO

 Results  Messages

	id_cliente	id_orden
1	22	NULL
2	57	NULL
3	1	10643
4	1	10692
5	1	10702
6	1	10835
7	1	10952
8	1	11011
9	2	10308
10	2	10625
11	2	10759
12	2	10026

35. Se requiere, de todos los clientes,

OBJETIVOS

- • aquellos que compraron sólo en Marzo 2021.
- • del resto, que se represente por un valor NULL.

SCRIPT

```
SELECT
  T1.id_cliente,
  T2.fecha_orden,
  T2.id_orden
FROM
  Ventas.Clientes T1
  LEFT OUTER JOIN
  Ventas Ordenes T2
  ON T1.id_cliente = T2.id_cliente
  AND fecha_orden BETWEEN '20210301' AND '20210331'
ORDER BY
  T1.id_cliente ASC
;
```

RESULTADO

Results		Messages	
	id_cliente	fecha_orden	id_orden
7	7	NULL	NULL
8	8	NULL	NULL
9	9	2021-03-11	10470
10	10	NULL	NULL
11	11	2021-03-11	10471
12	11	2021-03-24	10484
13	12	NULL	NULL
14	13	NULL	NULL
15	14	NULL	NULL
16	15	2021-03-06	10466
17	16	2021-03-03	10462
18	17	NULL	NULL

36. De los clientes que compraron en Marzo 2021, se requiere:

OBJETIVOS

- • visualizar por separado los que sí compraron (y la cantidad de pedidos) de los que no compraron.
- • que el listado ordenado por cliente.

SCRIPT

```
SELECT
  T1.id_cliente,
  (CASE WHEN COUNT (DISTINCT T2.id_orden) = 0
  THEN 'No' ELSE 'Sí' END) AS Pide_en_Mar_21,
  COUNT (T2.id_orden) AS Nro_pedidos
FROM
  Ventas.Clientes T1
LEFT OUTER JOIN
  Ventas.Ordenes T2
ON T1.id_cliente = T2.id_cliente
AND fecha_orden BETWEEN '20210301' AND '20210331'
GROUP BY
  T1.id_cliente
ORDER BY
  (CASE WHEN COUNT (DISTINCT T2.id_orden) = 0
  THEN 0 ELSE 1 END) DESC,
  T1.id_cliente ASC
;
```

RESULTADO

	id_cliente	Pide_en_Mar_21	Nro_pedidos
16	60	Sí	1
17	62	Sí	1
18	65	Sí	1
19	67	Sí	1
20	72	Sí	1
21	76	Sí	2
22	83	Sí	1
23	84	Sí	1
24	89	Sí	2
25	1	No	0
26	2	No	0
27	3	No	0

37. Se requiere información sobre el inventario de la empresa:

OBJETIVOS

- • stock mínimo y máximo y stock de seguridad de cada producto.
- • que estén ordenados por nombre de categoría.

SCRIPT

```
SELECT
  T1.nombrecategoria,
  T2.nombreproducto,
  CONCAT ('[' ,AVG (T3.cantidad),'-', AVG (T3.cantidad)*2,'>') AS Stock_min_max,
  (AVG (T3.cantidad)/2)+(AVG (T3.cantidad)) AS Stock_seguridad
FROM
  Produccion.Categorias T1
  INNER JOIN
  Produccion.Productos T2
  ON
  T1.id_categoria = T2.id_categoria
  INNER JOIN
  Ventas.DetalleOrden T3
  ON
  T2.id_producto = T3.id_producto
  INNER JOIN
  Ventas.Ordenes T4
  ON
  T3.id_orden = T4.id_orden
GROUP BY
  T1.nombrecategoria,
  T2.nombreproducto
ORDER BY
  T1.nombrecategoria ASC
;
```

RESULTADO

Results Messages				
	nombrecategoria	nombreproducto	Stock_min_max	Stock_seguridad
4	Bebidas	Producto NEVTJ	[24-48>	36
5	Bebidas	Producto HHYDP	[21-42>	31
6	Bebidas	Producto ZZZHR	[20-40>	30
7	Bebidas	Producto XLXQF	[18-36>	27
8	Bebidas	Producto RECZE	[24-48>	36
9	Bebidas	Producto JYGFE	[25-50>	37
10	Bebidas	Producto QOGNU	[22-44>	33
11	Bebidas	Producto SWNJY	[26-52>	39
12	Bebidas	Producto LSOFL	[26-52>	39
13	Carne/Aves	Producto BLCAX	[26-52>	39
14	Carne/Aves	Producto AOZBW	[19-38>	28
15	Carne/Aves	Producto VJXVH	[23-46>	34

38. ¿Cuánto representa en porcentaje sobre el total el gasto en fletes por los envíos de productos al país de los clientes?

SCRIPT

```
SELECT
    T1.paistransporte,
    CONVERT (varchar, (SUM (T1.flete)), 1) AS Flete_total,
    CONCAT((SUM (T1.flete) / (SELECT SUM (A.flete) FROM Ventas.Ordenes A))*100, ' %') AS
Porc_del_Total
FROM
    Ventas.Ordenes T1
GROUP BY
    T1.paistransporte
ORDER BY
    SUM (T1.flete) DESC
;
```

RESULTADO

Results		Messages	
	paistransporte	Flete_total	Porc_del_Total
1	USA	13,771.29	21.20 %
2	Alemania	11,283.28	17.37 %
3	Austria	7,391.50	11.38 %
4	Brasil	4,880.19	7.51 %
5	Francia	4,237.84	6.52 %
6	Suecia	3,237.60	4.98 %
7	Reino Unido	2,954.27	4.54 %
8	Irlanda	2,755.24	4.24 %
9	Venezuela	2,735.18	4.21 %
10	Canadá	2,198.09	3.38 %
11	Dinamarca	1,396.19	2.14 %
12	Suiza	1,368.53	2.10 %

39. ¿Cuánto representan las ventas en cada país en porcentaje sobre el total de ventas?

SCRIPT

```
SELECT
    T2.paistransporte,
    CONVERT (varchar,(SUM (T1.cantidad * T1.preciounitario)),1) AS Importe_total,
    (SUM (T1.cantidad * T1.preciounitario) /
    (
    SELECT
        SUM (A.cantidad * A.preciounitario)
    FROM
        Ventas.DetalleOrden A
    INNER JOIN Ventas Ordenes B
        ON A.id_orden = B.id_orden
    ))*100 AS Porc_del_total
FROM
    Ventas.DetalleOrden T1
    INNER JOIN Ventas Ordenes T2
        ON T1.id_orden = T2.id_orden
GROUP BY
    T2.paistransporte
ORDER BY
    Porc_del_total DESC
;
```

RESULTADO

Results		Messages	
	paistransporte	Importe_total	Porc_del_total
1	USA	263,566.98	19,45
2	Alemania	244,640.63	18,06
3	Austria	139,496.63	10,29
4	Brasil	114,968.48	8,48
5	Francia	85,498.76	6,31
6	Venezuela	60,814.89	4,48
7	Reino Unido	60,616.51	4,47
8	Suecia	59,523.70	4,39
9	Irlanda	57,317.39	4,23
10	Canadá	55,334.10	4,08
11	Bélgica	35,134.98	2,59
12	Dinamarca	34,782.25	2,56

40. Se requiere información general sobre las unidades totales vendidas por cada año, pero sólo de aquellos 20 clientes con más ventas.

SCRIPT

```
SELECT
    YEAR (T2.fecha_orden) AS Año,
    SUM(t1.cantidad) As Unidades
FROM Ventas.DetalleOrden T1
INNER JOIN Ventas.Ordenes T2
    ON T1.id_orden = T2.id_orden
WHERE
    T2.id_cliente IN
    (
        SELECT TOP 20
            b.id_cliente
        FROM
            Ventas.DetalleOrden A
            INNER JOIN
            Ventas.Ordenes B
            ON
            a.id_orden = b.id_orden
        GROUP BY
            b.id_cliente
        ORDER BY
            SUM (a.cantidad * a.preciounitario * (1-a.descuento)) DESC
    )
GROUP BY
    YEAR (T2.fecha_orden)
ORDER BY
    YEAR (T2.fecha_orden) ASC
;
```

RESULTADO

Results		Messages
	Año	Unidades
1	2020	5697
2	2021	15216
3	2022	10510

41. Se necesita saber cuándo fue la fecha del primer pedido de cada cliente.

SCRIPT

```
SELECT
  T1.id_cliente,
  (
    SELECT
      MIN (T2.fecha_orden)
    FROM Ventas.Ordenes T2
    WHERE
      T1.id_cliente = T2.id_cliente
  ) AS Fecha_primer_pedido,
  T1.paistransporte
FROM
  Ventas.Ordenes T1
GROUP BY
  T1.id_cliente,
  T1.paistransporte
;
```

RESULTADO

Results		Messages	
	id_cliente	Fecha_primer_pedido	paistransporte
1	1	2021-08-25	Alemania
2	2	2020-09-18	Mexico
3	3	2020-11-27	Mexico
4	4	2020-11-15	Reino Unido
5	5	2020-08-12	Suecia
6	6	2021-04-09	Alemania
7	7	2020-07-25	Francia
8	8	2020-10-10	España
9	9	2020-10-16	Francia
10	10	2020-12-20	Canadá
11	11	2020-08-26	Reino Unido
12	12	2021-04-29	Argentina

42. Se requiere información de los 3 últimos pedidos de cada cliente.

SCRIPT

```
SELECT
  T1.id_cliente,
  T1.id_orden,
  T1.fecha_orden,
  T1.paistransporte
FROM
  Ventas.Ordenes T1
WHERE
  T1.fecha_orden IN
  (
    SELECT TOP 3 T2.fecha_orden
    FROM Ventas.Ordenes T2
    WHERE
      T1.id_cliente = T2.id_cliente
    GROUP BY T2.fecha_orden
    ORDER BY T2.fecha_orden DESC
  )
ORDER BY
  T1.id_cliente ASC
;
```

RESULTADO

Results		Messages		
	id_cliente	id_orden	fecha_orden	paistransporte
1	1	10835	2022-01-15	Alemania
2	1	10952	2022-03-16	Alemania
3	1	11011	2022-04-09	Alemania
4	2	10926	2022-03-04	Mexico
5	2	10625	2021-08-08	Mexico
6	2	10759	2021-11-28	Mexico
7	3	10682	2021-09-25	Mexico
8	3	10677	2021-09-22	Mexico
9	3	10856	2022-01-28	Mexico
10	4	10920	2022-03-03	Reino Unido
11	4	11016	2022-04-10	Reino Unido
12	4	10953	2022-03-16	Reino Unido

43. Se requiere información de los días transcurridos desde el primer pedido de cada cliente.

SCRIPT

```
SELECT
  T1.id_cliente,
  T1.id_orden,
  T1.fecha_orden,
  DATEDIFF (DAY,
  (
    SELECT MIN (T2.fecha_orden)
    FROM Ventas.Ordenes T2
    WHERE T1.id_cliente = T2.id_cliente
  ), T1.fecha_orden) AS Días_trans_dsd_primer_pdo
FROM
  Ventas.Ordenes T1
ORDER BY
  T1.id_cliente ASC,
  T1.id_orden ASC
;
```

RESULTADO

	id_cliente	id_orden	fecha_orden	Días_trans_dsd_primer_pdo
1	1	10643	2021-08-25	0
2	1	10692	2021-10-03	39
3	1	10702	2021-10-13	49
4	1	10835	2022-01-15	143
5	1	10952	2022-03-16	203
6	1	11011	2022-04-09	227
7	2	10308	2020-09-18	0
8	2	10625	2021-08-08	324
9	2	10759	2021-11-28	436
10	2	10926	2022-03-04	532
11	3	10365	2020-11-27	0
12	3	10507	2021-04-15	139

44. Se requiere crear una consulta con el coste acumulado y el ingreso acumulado por país y periodo.

SCRIPT

```
SELECT
T1.periodo,
T1.pais,
CONVERT (varchar, (
    SELECT SUM(T2.costo) FROM dbo.PaisCosto T2
    WHERE T1.pais = T2.pais
    AND T2.periodo <= t1.periodo
), 1) AS Coste_acum,
CONVERT (varchar, (
    SELECT SUM(T3.ingreso) FROM dbo.PaisIngresos T3
    WHERE T1.pais = T3.pais
    AND T3.periodo <= T1.periodo
),1) AS Ingreso_acum
FROM
    dbo.PaisIngresos T1
ORDER BY
    pais ASC,
    periodo ASC
;
```

RESULTADO

Results		Messages		
	periodo	pais	Coste_acum	Ingreso_acum
16	202110	Alemania	7,712.81	143,811.85
17	202111	Alemania	7,716.33	148,341.65
18	202112	Alemania	8,189.69	161,974.93
19	202201	Alemania	9,112.73	184,110.48
20	202202	Alemania	9,605.71	208,870.94
21	202203	Alemania	10,136.84	220,095.79
22	202204	Alemania	11,139.30	242,680.28
23	202205	Alemania	11,283.28	244,640.63
24	202101	Argentina	29.83	319.20
25	202102	Argentina	68.65	762.60
26	202104	Argentina	85.87	988.10
27	202105	Argentina	93.99	1,098.10

-- 45. Se requiere saber la fecha del último pedido de cada cliente.

SCRIPT

```
SELECT
  T1.id_orden,
  T1.id_cliente,
  T1.fecha_orden,
  T1.paistransporte,
  (
    SELECT MAX (T2.fecha_orden)
    FROM ventas.Ordenes T2
    WHERE T1.id_cliente = T2.id_cliente
    AND T2.fecha_orden < T1.fecha_orden
  ) AS Fecha_ult_pedido
FROM
  Ventas.Ordenes T1
ORDER BY
  T1.id_cliente ASC,
  T1.fecha_orden ASC
;
```

RESULTADO

Results		Messages			
	id_orden	id_cliente	fecha_orden	paistransporte	Fecha_ult_pedido
1	10643	1	2021-08-25	Alemania	NULL
2	10692	1	2021-10-03	Alemania	2021-08-25
3	10702	1	2021-10-13	Alemania	2021-10-03
4	10835	1	2022-01-15	Alemania	2021-10-13
5	10952	1	2022-03-16	Alemania	2022-01-15
6	11011	1	2022-04-09	Alemania	2022-03-16
7	10308	2	2020-09-18	Mexico	NULL
8	10625	2	2021-08-08	Mexico	2020-09-18
9	10759	2	2021-11-28	Mexico	2021-08-08
10	10926	2	2022-03-04	Mexico	2021-11-28
11	10365	3	2020-11-27	Mexico	NULL
12	10507	3	2021-04-15	Mexico	2020-11-27