

Anexos técnicos

A partir de las consultas SQL que construimos durante el desarrollo del proyecto, fue posible consolidar la información necesaria para evaluar la calidad de los datos, realizar los procesos de limpieza pertinentes y avanzar hacia un entendimiento inicial coherente con el problema de negocio. Estas consultas permiten observar de manera directa la completitud, consistencia y estructura de las distintas fuentes, y facilitan la identificación de valores faltantes, anomalías o patrones que justifican las decisiones de depuración. Del mismo modo, los resultados obtenidos con cada query sirven como evidencia para sustentar el análisis realizado en el informe, ya que muestran cómo se integran las tablas, cómo se derivan las variables necesarias para el modelado y cómo se comportan los datos frente a los requerimientos descritos en la guía. Gracias a estas salidas es posible respaldar cada sección del entregable y disponer de los anexos técnicos que validan el proceso.

```
WITH ventas_cliente AS (
    SELECT
        ss_customer_sk          AS id_cliente,
        COUNT(DISTINCT ss_ticket_number)  AS num_pedidos,
        COUNT(ss_item_sk)          AS num_items,
        SUM(ss_net_paid)           AS monto_comprado
    FROM store_sales
    GROUP BY ss_customer_sk
),
devoluciones_cliente AS (
    SELECT
        sr_customer_sk          AS id_cliente,
        COUNT(DISTINCT sr_ticket_number)  AS num_pedidos_dev,
        COUNT(sr_item_sk)          AS num_items_dev,
        SUM(sr_return_amt)          AS monto_devuelto
    FROM store_returns
    GROUP BY sr_customer_sk
)
SELECT
    v.id_cliente AS cliente,
    -- relación de pedidos devueltos vs pedidos totales
    ROUND(
        COALESCE(
            1.0 * d.num_pedidos_dev / NULLIF(v.num_pedidos, 0),
            0
        ),
        7
    ) AS ratio_de_pedidos_devueltos,
    -- proporción de artículos devueltos vs artículos comprados
    ROUND(
        COALESCE(
            1.0 * d.num_items_dev / NULLIF(v.num_items, 0),
            0
        ),
        7
    ) AS ratio_de_articulos_devueltos,
    -- relación de importe devuelto vs importe comprado
    ROUND(
        COALESCE(
            d.monto_devuelto / NULLIF(v.monto_comprado, 0),
            0
        ),
        7
    ) AS ratio_monetario_de_devolución,
    -- frecuencia de devolución (número de pedidos devueltos)
    COALESCE(d.num_pedidos_dev, 0) AS frecuencia_de_retorno
FROM ventas_cliente v
LEFT JOIN devoluciones_cliente d
    ON v.id_cliente = d.id_cliente;
```

File Edit View Query Git Project Tools Extensions Window Help Search Solution

Object Explorer

```

1  WITH ventas_cliente AS (
2      SELECT
3          ss.customer_sk AS id_cliente,
4          COUNT(DISTINCT ss.ticket_number) AS num_pedidos,
5          SUM(ss.item_pk) AS num_itens,
6          SUM(ss.net_paid) AS monto_comprado
7      FROM store_sales
8      GROUP BY ss.customer_sk
9  ),
10  devoluciones_cliente AS (
11      SELECT
12          sr.customer_sk AS id_cliente,
13          COUNT(DISTINCT sr.ticket_number) AS num_pedidos_dev,
14          COUNT(sr.item_sk) AS num_itens_dev,
15          SUM(sr.return_amt) AS monto_devuelto
16      FROM store_returns
17      GROUP BY sr.customer_sk
18  )
19
20  SELECT
21      v.id_cliente AS cliente,
22      -- relación de pedidos devueltos vs pedidos totales
23      ROUNDED(
24          COALESCE(
25              1.0 * d.num_pedidos_dev / NULLIF(v.num_pedidos, 0),
26              0
27          ), 7
28      ) AS ratio_de_pedidos_devueltos,
29
30      -- proporción de artículos devueltos vs artículos comprados
31      ROUNDED(
32          COALESCE(
33              1.0 * d.num_itens_dev / NULLIF(v.num_itens, 0),
34              0
35          ), 7
36      ) AS ratio_de_itens_devueltos
37
38  FROM
39      ventas_cliente v
40      JOIN devoluciones_cliente d
41      ON v.id_cliente = d.id_cliente
42
43  ORDER BY
44      cliente
45
46  
```

No issues found

Results Messages

cliente	ratio_de_pedidos_devueltos	ratio_de_itens_devueltos	ratio_monto_de_devolucion	frecuencia_de_retorno
1 29727	0.000000000	0.000000000	0.0000000	0
2 97643	0.061818000	0.078179000	0.03704	3
3 57247	0.000000000	0.000000000	0.0000000	0
4 2040	0.068000000	0.068000000	0.017281	4
5 2040	0.000000000	0.000000000	0.0000000	0
6 81474	0.000000000	0.000000000	0.0000000	0
7 60253	0.200000000	0.136363000	0.065762	3
8 48778	0.000000000	0.000000000	0.0000000	0
9 80439	0.000000000	0.000000000	0.0000000	0
10 55118	0.000000000	0.000000000	0.0000000	0

Ln 53 Ch 36 SPC CRLF

DESKTOP-SDA345T\SQLEXPRESS... DESKTOP-SDA345T\b_o_x... tpcobb_1gb 00:00:00 37,336 rows

Ready

```

SELECT
    c.c_customer_sk AS id_cliente,
    LTRIM(RTRIM(c.c_first_name + ' ' + c.c_last_name)) AS nombre_completo,
    c.c_birth_year AS anio_nacimiento,
    c.c_birth_month AS mes_nacimiento,
    c.c_birth_day AS dia_nacimiento,
    c.c_birth_country AS pais_origen,
    c.c_last_review_date AS revision
FROM customer AS c;
```

File Edit View Query Git Project Tools Extensions Window Help Search Solution

Object Explorer

```

1  SELECT
2      c.c_customer_sk AS id_cliente,
3      LTRIM(RTRIM(c.c_first_name + ' ' + c.c_last_name)) AS nombre_completo,
4      c.c_birth_year AS anio_nacimiento,
5      c.c_birth_month AS mes_nacimiento,
6      c.c_birth_day AS dia_nacimiento,
7      c.c_birth_country AS pais_origen,
8      c.c_last_review_date AS fecha_ultima_revision
9  FROM customer AS c;
10 
```

No issues found

Results Messages

id_cliente	nombre_completo	anio_nacimiento	mes_nacimiento	dia_nacimiento	pais_origen	fecha_ultima_revision
1 6089	Todd Reynolds	1926	6	25	GUYANA	mmXOHNE
2 6168	Timothy Hill	1941	6	12	BOTSWANA	bZS
3 6298	Chalte Austin	1904	6	16	TURKMENISTAN	Qm64fA1g
4 6335	Michael Byrne	1942	6	14	SRI LANKA	goEabbb
5 6340	John Ogden	1931	6	8	ARMENIA REPUBLIC OF	u
6 6432	Jennifer Shannon	1962	6	4	BENIN	adjeB
7 6546	Nancy Alvarez	1947	6	28	LESOTHO	DGR5R
8 6608	Shannon Wright	1961	6	20	ESTONIA	reAOBT8p
9 6644	Gilbert Allen	1964	6	15	ANDORRA	nEXk6P8h
10 6671	Anna Ayon	1946	6	17	UZBEKISTAN	NTXY
11 6941	Christine Morgan	1973	6	15	JAMAICA	1
12 6985	Raymond Wheeler	1990	6	1	AMERICAN SAMOA	Nhgufy
13 7000	Stephanie Barnes	1948	6	5	DENMARK	D
14 7100	Oleg Morozov	1987	6	24	TOGO	aDn
15 7163	Manuel Drake	1959	6	11	GUINEA	YoNC2
16 7175	Michael Nuter	1988	6	22	MARSHALL ISLANDS	19HrhvO
17 7216	John Munoz	1958	6	11	CAYMAN ISLANDS	wkUJlDe8
18 7318	Kevin Jacobs	1945	6	28	SOMALIA	X
19 7441	Wanda Pimentel	1967	6	11	ITALY	MK6s

Ln 1 Ch 1 SPC CRLF

DESKTOP-SDA345T\SQLEXPRESS... DESKTOP-SDA345T\b_o_x... tpcobb_1gb 00:00:00 99,000 rows

Ready

```

SELECT
    r.pr_review_date      AS fecha_revision,
    r.pr_item_sk          AS id_producto,
    r.pr_review_rating    AS calificacion,
    r.pr_review_content   AS comentario,
    r.pr_user_sk          AS id_usuario,
    i.i_category          AS categoria,
    i.i_product_name      AS nombre_producto,
    i.i_item_desc         AS descripcion_producto,
    i.i_size               AS tamano_producto
FROM product_reviews AS r
INNER JOIN item AS i
    ON r.pr_item_sk = i.i_item_sk;

```

The screenshot shows the SSMS interface with the following details:

- Object Explorer:** Shows the database structure, including tables like `dbo.customer`, `dbo.product_reviews`, and `item`.
- Query Editor:** Contains the T-SQL code for the first query.
- Results:** Displays the output of the query, which includes columns: `fecha_revision`, `id_producto`, `calificacion`, `comentario`, `id_usuario`, `categoria`, `nombre_producto`, `descripcion_producto`, and `tamano_producto`. The results show numerous reviews from users like "Home & Kitchen" with various comments and product details.
- Status Bar:** Shows the command history: DESKTOP-SDA3451\SQLEXPRESS → DESKTOP-SDA3451\b_o_x... tpcibb_1gb 00:00:01 89,991 rows.

```

SELECT
    CAST(YEAR(d.d_date) AS VARCHAR(4))
    + '-' +
    RIGHT('0' + CAST(MONTH(d.d_date) AS VARCHAR(2)), 2) AS anio_mes,
    SUM(s.ss_quantity * s.ss_list_price) AS total_ventas
FROM store_sales AS s
INNER JOIN date_dim AS d
    ON s.ss_sold_date_sk = d.d_date_sk
WHERE NOT (YEAR(d.d_date) = 2005 AND MONTH(d.d_date) = 12)
GROUP BY
    YEAR(d.d_date),
    MONTH(d.d_date)
ORDER BY
    YEAR(d.d_date),
    MONTH(d.d_date);

```

Screenshot of SQL Server Management Studio (SSMS) showing a query results window. The query is a complex SELECT statement using CTEs and various functions like SUM, CAST, and RIGHT to calculate total sales by year and month. The results show 59 rows of data from 2001-01 to 2002-07.

```

SELECT
    CAST(YEAR(d.d_date) AS VARCHAR(4)) + '-' +
    RIGHT('0' + CAST(MONTH(d.d_date) AS VARCHAR(2)), 2) AS anio_mes,
    SUM(s.ss_quantity * s.ss_list_price) AS total_ventas
FROM stg_sales_fact s
INNER JOIN date_dim AS d
    ON s.ss_sold_date_sk = d.d_date_sk
WHERE NOT (YEAR(d.d_date) = 2005 AND MONTH(d.d_date) = 12)
GROUP BY
    CAST(YEAR(d.d_date) AS VARCHAR(4)) + '-' +
    RIGHT('0' + CAST(MONTH(d.d_date) AS VARCHAR(2)), 2),
    ORDER BY
        YEAR(d.d_date),
        MONTH(d.d_date);

```

Results	Messages
1 2001-01 4269419.39	
2 2001-02 4014400.07	
3 2001-03 4339480.31	
4 2001-04 4219200.64	
5 2001-05 4384348.44	
6 2001-06 4249757.52	
7 2001-07 4384581.84	
8 2001-08 4217800.00	
9 2001-09 4317800.29	
10 2001-10 4404114.17	
11 2001-11 42947705.56	
12 2001-12 42876517.95	
13 2002-01 4389574.89	
14 2002-02 3873280.12	
15 2002-03 4413800.25	
16 2002-04 4341196.22	
17 2002-05 4425450.41	
18 2002-06 4151334.29	
19 2002-07 4455334.82	

```

SELECT
    d.SalesOrderID AS factura,
    FORMAT(h.OrderDate, 'dd-MM-yyyy') AS fecha,
    d.ProductID AS id_producto,
    p.Name AS nombre_producto,
    d.OrderQty AS cantidad,
    d.UnitPrice AS precio_unitario
FROM Sales.SalesOrderDetail AS d
INNER JOIN Sales.SalesOrderHeader AS h
    ON d.SalesOrderID = h.SalesOrderID
INNER JOIN Production.Product AS p
    ON d.ProductID = p.ProductID;

```

Screenshot of SSMS showing a query results window. The query retrieves data from Sales.SalesOrderDetail and Sales.SalesOrderHeader tables, joining them with Production.Product. The results show 121,317 rows of data from June 2004.

```

SELECT
    d.SalesOrderID AS factura,
    FORMAT(h.OrderDate, 'dd-MM-yyyy') AS fecha,
    d.ProductID AS id_producto,
    p.Name AS nombre_producto,
    d.OrderQty AS cantidad,
    d.UnitPrice AS precio_unitario
FROM Sales.SalesOrderDetail AS d
INNER JOIN Sales.SalesOrderHeader AS h
    ON d.SalesOrderID = h.SalesOrderID
INNER JOIN Production.Product AS p
    ON d.ProductID = p.ProductID;

```

Results	Messages
1 75117 30-06-2014 866 Half-Finger Gloves, L 1 24.49	
1 75118 30-06-2014 923 Touring Tire Tube 1 4.99	
1 75118 30-06-2014 873 Patch Kit/B Patches 1 2.29	
1 75119 30-06-2014 877 Bike Wash - Dissolver 1 7.95	
1 75119 30-06-2014 876 Hitch Rack - Bike 1 100.00	
1 75119 30-06-2014 921 Mountain Tire Tube 1 4.99	
1 75119 30-06-2014 930 HU Mountain Tire 1 35.00	
1 75119 30-06-2014 873 Patch Kit/B Patches 1 2.29	
1 75120 30-06-2014 878 Fender Set - Mountain 1 21.98	
1 75120 30-06-2014 888 Short-Sleeve Classic Jersey... 1 53.99	
1 75120 30-06-2014 712 AWC Logo Cap 1 8.99	
1 75121 30-06-2014 921 Mountain Tire Tube 1 4.99	
1 75121 30-06-2014 930 HU Mountain Tire 1 35.00	
1 75121 30-06-2014 707 Sport-100 Helmet, Red 1 34.99	
1 75122 30-06-2014 878 Fender Set - Mountain 1 21.98	
1 75122 30-06-2014 712 AWC Logo Cap 1 8.99	
1 75123 30-06-2014 878 Fender Set - Mountain 1 21.98	
1 75123 30-06-2014 879 All-Purpose Bike Stand 1 159.00	
1 75123 30-06-2014 712 AWC Logo Cap 1 8.99	

Encuesta NPS

```
SELECT
    a.id          AS encuesta_id,
    a.fecha       AS fecha_respuesta,
    a.genero     AS genero_original,
    CASE
        WHEN a.genero = 'femenino' THEN 1
        WHEN a.genero = 'masculino' THEN 2
        ELSE 0
    END          AS genero_codificado,
    a.origen      AS origen_original,
    CASE
        WHEN a.origen = 'udemy'   THEN 1
        WHEN a.origen = 'frogames' THEN 2
        WHEN a.origen = 'crehana'  THEN 3
        WHEN a.origen = 'facebook' THEN 4
        WHEN a.origen = 'web'      THEN 5
        ELSE 6
    END          AS origen_codificado,
    a.edad        AS edad_encuestado,
    b.corr        AS curso_id,
    b.name        AS nombre_curso,
    a.mensaje     AS comentario,
    a.tickmarks   AS puntuacion_nps,
    CASE
        WHEN a.tickmarks <= 6 THEN 'Detractor'
        WHEN a.tickmarks >= 9 THEN 'Promotor'
        ELSE 'Pasivo'
    END          AS categoria_nps
FROM tb_nps AS a
INNER JOIN curso_nps AS b
    ON a.producto = b.id;
```

Repositorio Git Hub proyecto

[Enlace al repositorio](#)

