

## 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;
```

The screenshot shows a SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with a tree view of the database structure. The main pane shows a query window with the following SQL code:

```

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
),
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
FROM customer AS c

```

The results pane shows a table with the following columns: cliente, ratio\_de\_pedidos\_devueltos, ratio\_de\_articulos\_devueltos, frecuencia\_de\_devolucion, frecuencia\_de\_retorno. The table contains 10 rows of data.

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;

The screenshot shows a SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with a tree view of the database structure. The main pane shows a query window with the following SQL code:

```

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 fecha_ultima_revision
FROM customer AS c

```

The results pane shows a table with the following columns: id\_cliente, nombre\_completo, anio\_nacimiento, mes\_nacimiento, dia\_nacimiento, pais\_origen, fecha\_ultima\_revision. The table contains 18 rows of data.

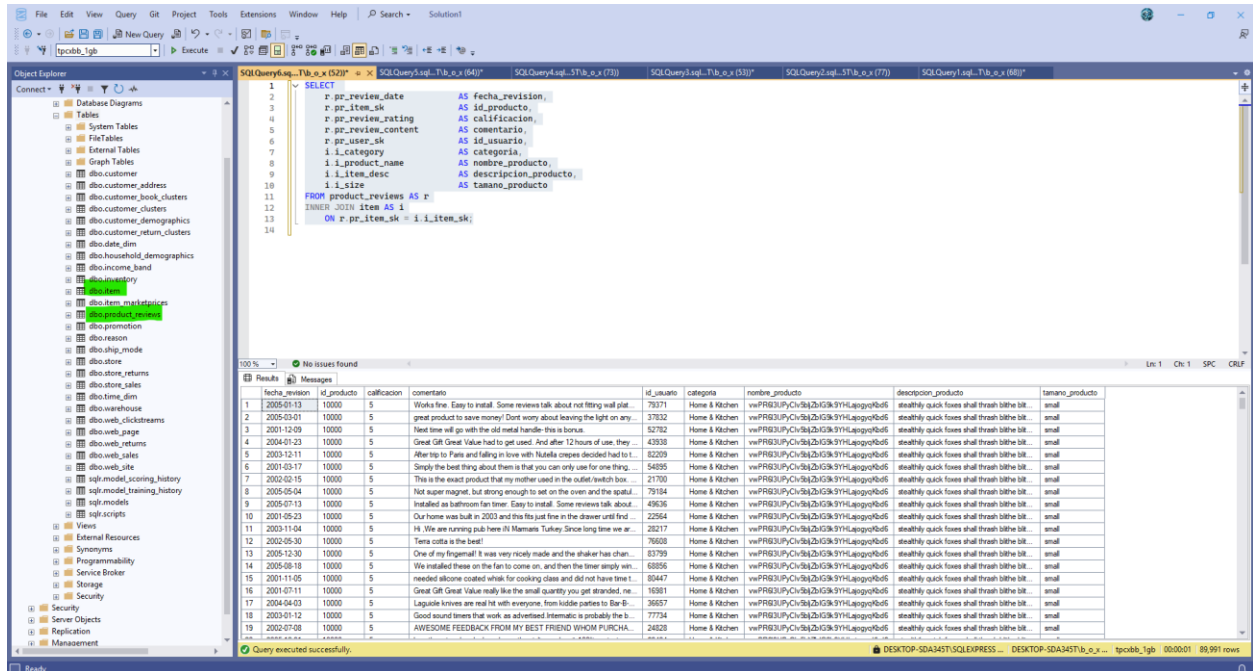
SELECT

```
r.pr_review_date      AS fecha_revisión,
r.pr_item_sk          AS id_producto,
r.pr_review_rating    AS calificación,
r.pr_review_content   AS comentario,
r.pr_user_sk          AS id_usuario,
i.i_category          AS categoría,
i.i_product_name      AS nombre_producto,
i.i_item_desc         AS descripción_producto,
i.i_size              AS tamaño_producto
```

FROM product\_reviews AS r

INNER JOIN item AS i

```
ON r.pr_item_sk = i.i_item_sk;
```



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);
```

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'AdventureWorks2019' database selected. The central pane shows a T-SQL query window with the following query:

```

SELECT
  CAST(YEAR(d.d_date) AS VARCHAR(4)) AS anio_mes,
  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);

```

The right pane shows the 'Results' tab with the following data:

anio_mes	total_ventas
2001-01	4203419.19
2001-02	40164080.07
2001-03	43394863.31
2001-04	42219200.64
2001-05	42943448.44
2001-06	42947957.52
2001-07	43384851.84
2001-08	43659186.85
2001-09	43179069.29
2001-10	4404143.17
2001-11	42947705.56
2001-12	42872617.95
2002-01	4389874.89
2002-02	3873283.12
2002-03	44138005.25
2002-04	43411961.22
2002-05	44040406.61
2002-06	44533384.82

The status bar at the bottom indicates 'Query executed successfully' and '59 rows'.

```

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;

```

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'AdventureWorks2019' database selected. The central pane shows a T-SQL query window with the following query:

```

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;

```

The right pane shows the 'Results' tab with the following data:

factura	fecha	id_producto	nombre_producto	cantidad	precio_unitario	
1	75117	30-06-2014	860	Half Finger Gloves, L	1	24.49
1	75118	30-06-2014	323	Trouser Tire Tube	1	4.99
1	75118	30-06-2014	873	Patch Kit/8 Patches	1	2.29
1	75118	30-06-2014	877	Blue Wash - Detergent	1	7.99
1	75118	30-06-2014	876	Hotch Rock - 48Bike	1	120.00
1	75119	30-06-2014	921	Mountain Tire Tube	1	4.99
1	75119	30-06-2014	930	HL Mountain Tire	1	35.00
1	75119	30-06-2014	873	Patch Kit/8 Patches	1	2.29
1	75120	30-06-2014	878	Fender Set - Mountain	1	21.98
1	75120	30-06-2014	884	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	HL 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	Hi-Airpase Bike Stand	1	109.00
1	75123	30-06-2014	712	AWC Logo Cap	1	8.99

The status bar at the bottom indicates 'Query executed successfully' and '121,317 rows'.

# 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](#)

