

# Proyecto Holaluz

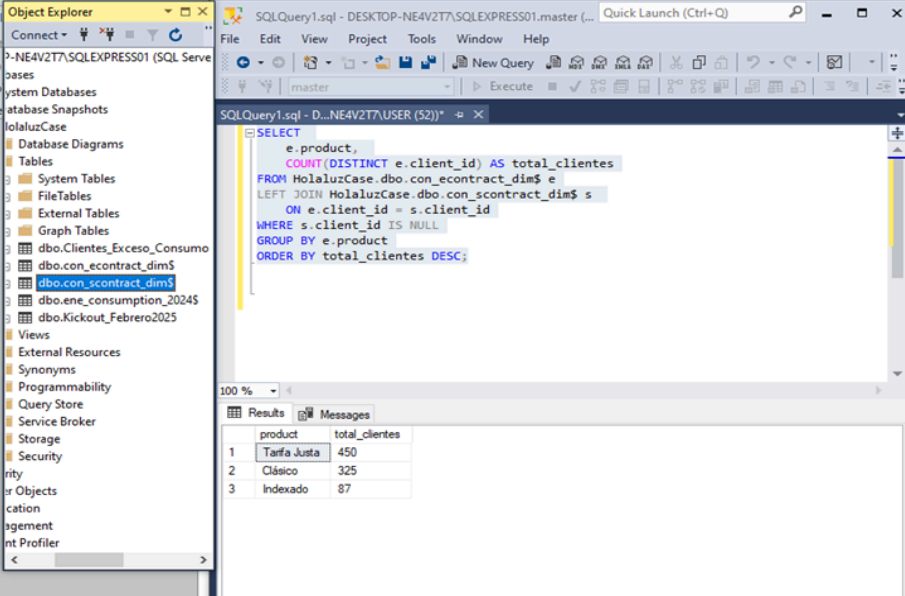
Matias Alejandro Proano Burbano

## 1. Resumen Ejecutivo

Este documento presenta el análisis técnico solicitado para evaluar el comportamiento de los clientes bajo la Tarifa Justa, segmentar clientes según su perfil de consumo, calcular pérdidas económicas y proponer mejoras de eficiencia operativa. Se utilizaron consultas SQL a través de la herramienta SQL Server Management sobre la base de datos *HolaluzCase*, análisis en Excel, y un diseño de dashboard en Power BI para visualización y control operativo.

## Ejercicio 1

a)



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure, including tables like `dbo.con_econtract_dim5` and `dbo.con_scontract_dim5`. The main window displays a SQL query in the Query Editor:

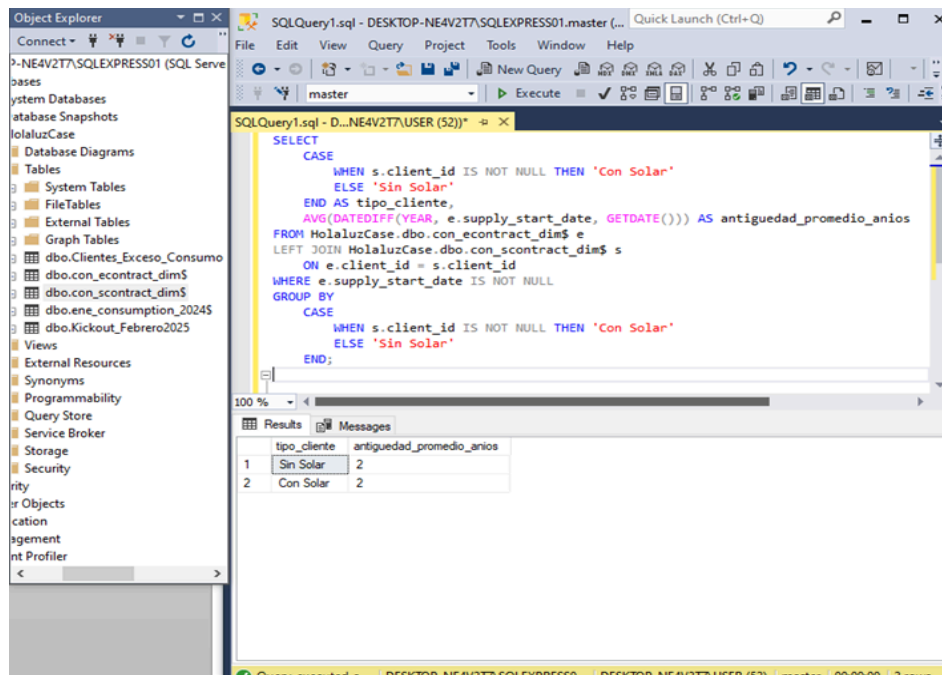
```
SELECT
    e.product,
    COUNT(DISTINCT e.client_id) AS total_clientes
FROM HolaluzCase.dbo.con_econtract_dim5 e
LEFT JOIN HolaluzCase.dbo.con_scontract_dim5 s
    ON e.client_id = s.client_id
WHERE s.client_id IS NULL
GROUP BY e.product
ORDER BY total_clientes DESC;
```

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

	product	total_clientes
1	Tarifa Justa	450
2	Clásico	325
3	Indexado	87

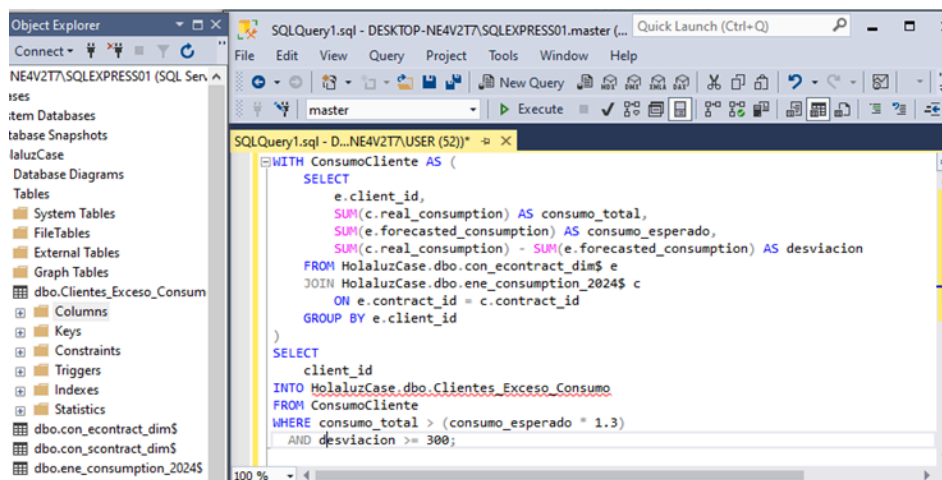
Este query identifica cuántos clientes únicos hay por tipo de producto, por otro lado se están excluyendo a los clientes con instalaciones solares. Así tenemos una visión clara de cuántos clientes activos hay para cada producto de luz que lo llamaremos “puro”.

b)

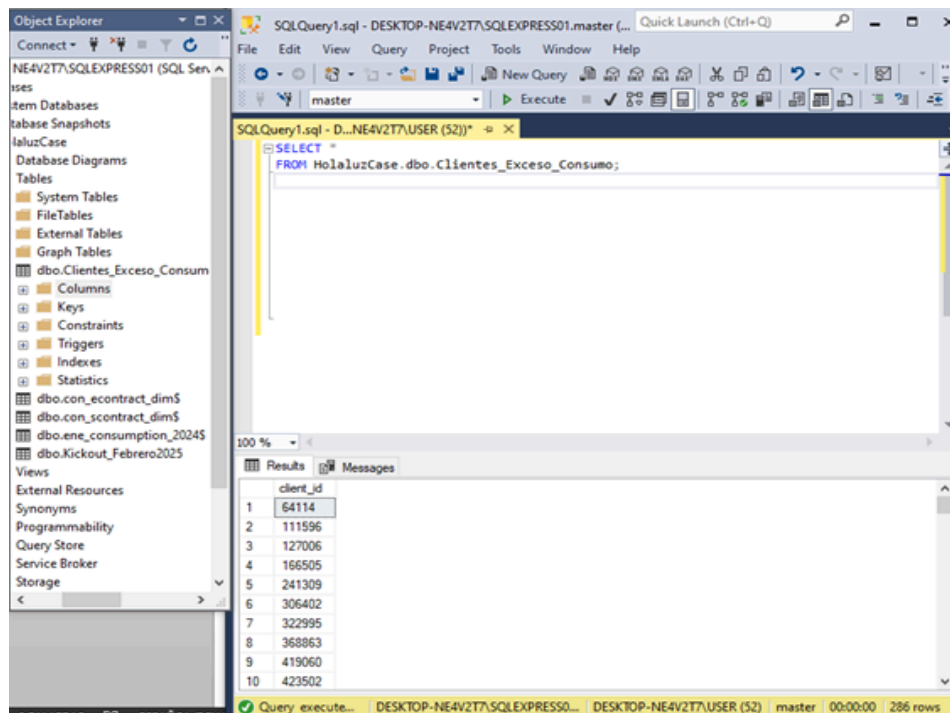


Se calcula la edad promedio en años de los contratos activos, diferenciando entre clientes con y sin instalación solar, este análisis lo sacamos en la columna *supply\_start\_date*.

c)

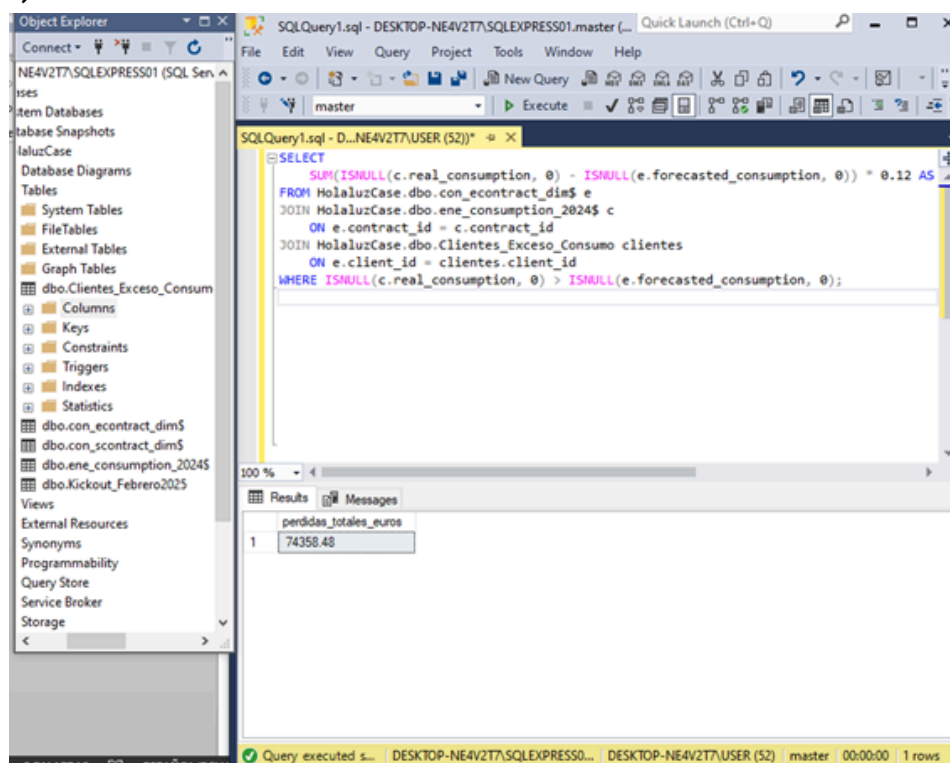


Este fue el query que usé para crear la tabla donde podemos identificar un conjunto de clientes que sobrepasan en más de 30% su consumo previsto y además superan los 300kWh de desviación, incumpliendo las condiciones de la Tarifa Justa.



Aquí se puede observar la tabla que se creó a través del query.

d)



Estamos calculando las pérdidas económicas totales que tiene Holaluz por clientes que tienen un sobreconsumo, considerando un coste de 0,12 €/kWh.

e)

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure, including tables like `dbo.con_econtract_dim$` and `dbo.ene_consumption_2024$`. The main window displays a SQL query titled "ITH ConsumoDesviacion AS" which calculates the average consumption deviation (in kWh) for clients with and without solar installation. The query uses `ISNULL` to handle null values and `AVG` to calculate the average deviation.

```

SELECT
    e.client_id,
    SUM(ISNULL(c.real_consumption, 0)) AS consumo_real,
    SUM(ISNULL(e.forecasted_consumption, 0)) AS consumo_esperado,
    SUM(ISNULL(c.real_consumption, 0)) - SUM(ISNULL(e.forecasted_consumption, 0)) AS
FROM HolaluzCase.dbo.con_econtract_dim$ e
JOIN HolaluzCase.dbo.ene_consumption_2024$ c
    ON e.contract_id = c.contract_id
GROUP BY e.client_id

SELECT
    CASE
        WHEN s.client_id IS NOT NULL THEN 'Con Solar'
        ELSE 'Sin Solar'
    END AS tipo_cliente,
    AVG(desviacion) AS desviacion_promedio_kwh
FROM ConsumoDesviacion d
LEFT JOIN HolaluzCase.dbo.con_econtract_dim$ s
    ON d.client_id = s.client_id
GROUP BY
    CASE
        WHEN s.client_id IS NOT NULL THEN 'Con Solar'
        ELSE 'Sin Solar'
    END;

```

The Results pane shows the output of the query:

tipo_cliente	desviacion_promedio_kWh
Sin Solar	2051.24361948956
Con Solar	1794.67391304348

Query executed successfully. 2 rows.

Estamos calculando la desviación media de consumo (**en kWh**) para los clientes con y sin instalación solar que lo nombramos como “Sin Solar” y “Con Solar”.

f)

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure, including tables like `dbo.con_econtract_dim$` and `dbo.ene_consumption_2024$`. The main window displays a SQL query titled "SELECT TOP 5" which identifies the top 5 contracts with the highest negative impact on the margin, based on the difference between real and forecasted consumption.

```

SELECT TOP 5
    e.contract_id,
    e.client_id,
    e.forecasted_consumption,
    c.real_consumption,
    (c.real_consumption - e.forecasted_consumption) * 0.12 AS impacto_margen_euros
FROM HolaluzCase.dbo.con_econtract_dim$ e
JOIN HolaluzCase.dbo.ene_consumption_2024$ c
    ON e.contract_id = c.contract_id
WHERE e.forecasted_consumption IS NOT NULL
    AND c.real_consumption > e.forecasted_consumption
ORDER BY
    (c.real_consumption - e.forecasted_consumption) * 0.12 DESC;

```

The Results pane shows the output of the query:

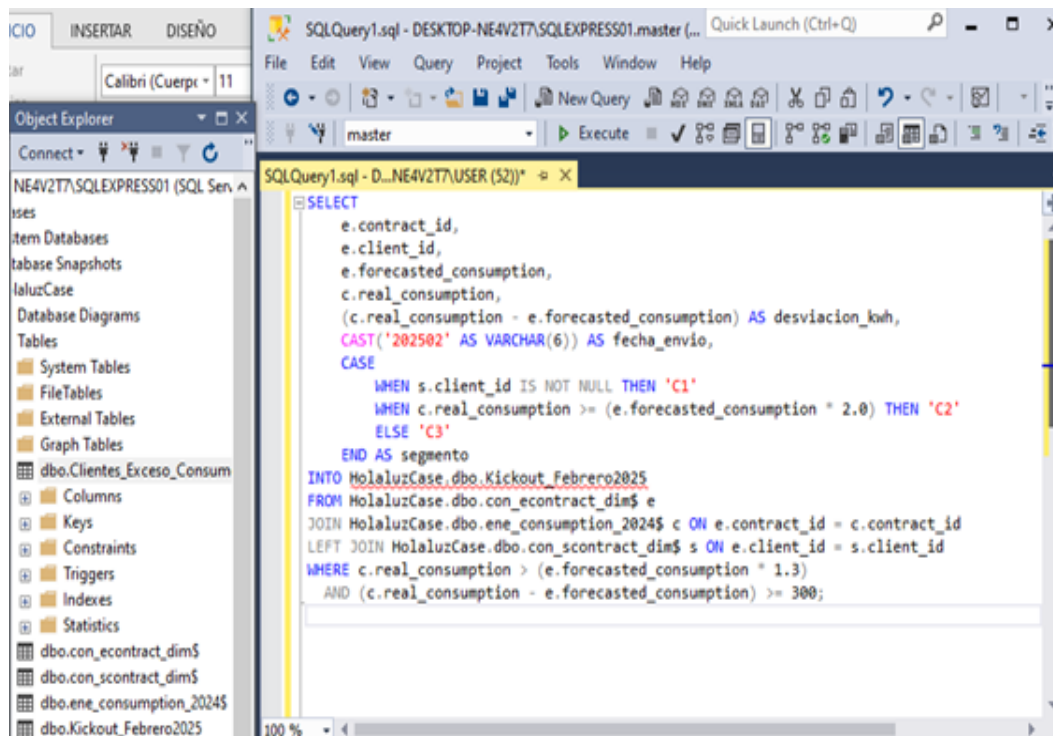
contract_id	client_id	forecasted_consumption	real_consumption	impacto_margen_euros
962	2619489	1000	5969	596.28
460	4214682	1037	5963	591.12
844	3971141	1010	5775	571.8
785	2989821	1187	5929	569.04
806	7463643	1191	5827	556.32

Query executed successfully. 5 rows.

Identificamos los 5 contratos que tienen mayor impacto negativo en el margen, ordenados por el exceso de consumo económico.

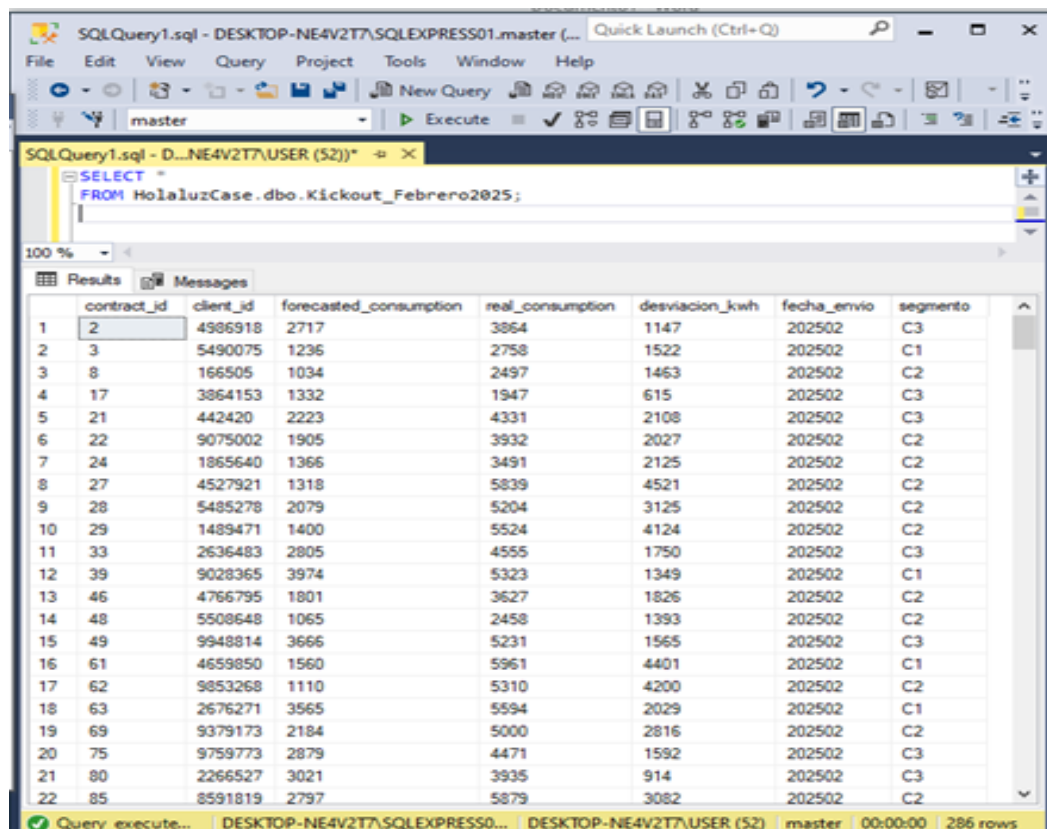
## Ejercicio 2 - Kick Out (Febrero, 2025)

a)



The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure, including tables like `dbo.con_econtract_dim$`, `dbo.con_scontract_dim$`, `dbo.ene_consumption_2024$`, and `dbo.Kickout_Febrero2025`. The main window displays the following SQL query:

```
SELECT
    e.contract_id,
    e.client_id,
    e.forecasted_consumption,
    c.real_consumption,
    (c.real_consumption - e.forecasted_consumption) AS desviacion_kwh,
    CAST('202502' AS VARCHAR(6)) AS fecha_envio,
    CASE
        WHEN s.client_id IS NOT NULL THEN 'C1'
        WHEN c.real_consumption >= (e.forecasted_consumption * 2.0) THEN 'C2'
        ELSE 'C3'
    END AS segmento
INTO HolaluzCase.dbo.Kickout_Febrero2025
FROM HolaluzCase.dbo.con_econtract_dim$ e
JOIN HolaluzCase.dbo.ene_consumption_2024$ c ON e.contract_id = c.contract_id
LEFT JOIN HolaluzCase.dbo.con_scontract_dim$ s ON e.client_id = s.client_id
WHERE c.real_consumption >= (e.forecasted_consumption * 1.3)
AND (c.real_consumption - e.forecasted_consumption) >= 300;
```



The screenshot shows the results of the query execution. The Results pane displays a table with 22 rows and 8 columns. The status bar at the bottom indicates that the query executed successfully, returning 286 rows.

	contract_id	client_id	forecasted_consumption	real_consumption	desviacion_kwh	fecha_envio	segmento
1	2	4986918	2717	3864	1147	202502	C3
2	3	5490075	1236	2758	1522	202502	C1
3	8	166505	1034	2497	1463	202502	C2
4	17	3864153	1332	1947	615	202502	C3
5	21	442420	2223	4331	2108	202502	C3
6	22	9075002	1905	3932	2027	202502	C2
7	24	1865640	1366	3491	2125	202502	C2
8	27	4527921	1318	5839	4521	202502	C2
9	28	5485278	2079	5204	3125	202502	C2
10	29	1489471	1400	5524	4124	202502	C2
11	33	2636483	2805	4555	1750	202502	C3
12	39	9028365	3974	5323	1349	202502	C1
13	46	4766795	1801	3627	1826	202502	C2
14	48	5508648	1065	2458	1393	202502	C2
15	49	9948814	3666	5231	1565	202502	C3
16	61	4659850	1560	5961	4401	202502	C1
17	62	9853268	1110	5310	4200	202502	C2
18	63	2676271	3565	5594	2029	202502	C1
19	69	9379173	2184	5000	2816	202502	C2
20	75	9759773	2879	4471	1592	202502	C3
21	80	2266527	3021	3935	914	202502	C3
22	85	8591819	2797	5879	3082	202502	C2

En esta etapa se construyó una tabla específica con los contratos que incumplen las condiciones de la Tarifa Justa de Holaluz. La lógica de exclusión se basó en dos criterios simultáneos:

1. Desviación superior al 30% respecto al consumo estimado (*forecasted\_consumption*).
2. Un exceso absoluto de consumo mínimo de 300 kWh.

La tabla contiene:

- Identificadores de contrato y cliente.
- El consumo esperado y el consumo real.
- La desviación en kWh.
- Una marca de envío (*fecha\_envio = '202502'*) que permite trazar el lote de Kick out.
- La segmentación en tres categorías:

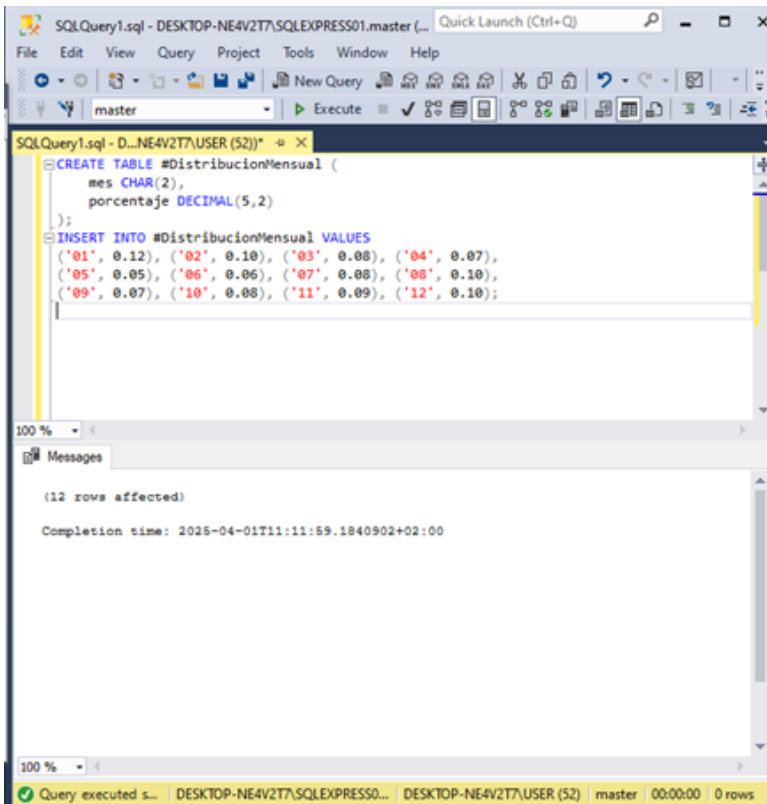
C1: Clientes con instalación solar.

C2: Sin solar y con desviación superior al 100%.

C3: Sin solar y desviación entre 30% y 99%.

Con esta estructura nos va a permitir automatizar el proceso de selección de clientes a quienes se enviará la notificación para el cambio de tarifa, y sienta las bases para análisis posteriores en dashboards y procesos operativos.

b)



```

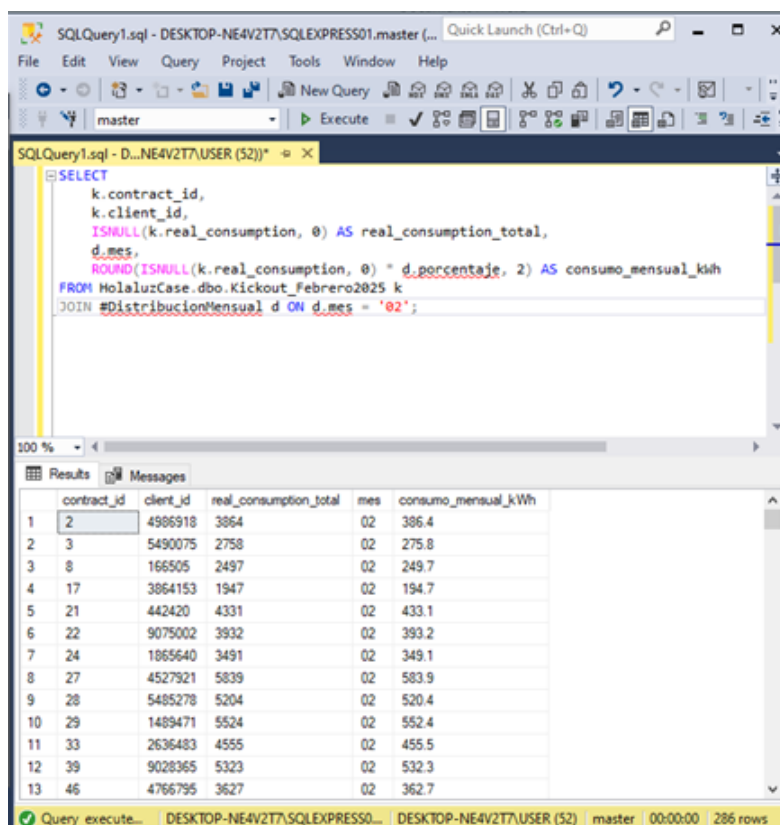
SQLQuery1.sql - DESKTOP-NE4V2T7\SQLEXPRESS01.master (... Quick Launch (Ctrl+Q))
File Edit View Query Project Tools Window Help
master Execute
SQLQuery1.sql - D:\NE4V2T7\USER (52))
CREATE TABLE #DistribucionMensual (
    mes CHAR(2),
    porcentaje DECIMAL(5,2)
);
INSERT INTO #DistribucionMensual VALUES
('01', 0.12), ('02', 0.10), ('03', 0.08), ('04', 0.07),
('05', 0.05), ('06', 0.06), ('07', 0.08), ('08', 0.10),
('09', 0.07), ('10', 0.08), ('11', 0.09), ('12', 0.10);

Messages
(12 rows affected)
Completion time: 2025-04-01T11:11:59.1840902+02:00

Query executed successfully. DESKTOP-NE4V2T7\SQLEXPRESS01.master DESKTOP-NE4V2T7\USER (52) master 00:00:00 0 rows

```

Con este query creamos la tabla con los meses y sus debidos porcentajes.



```

SQLQuery1.sql - DESKTOP-NE4V2T7\SQLEXPRESS01.master (... Quick Launch (Ctrl+Q))
File Edit View Query Project Tools Window Help
master Execute
SQLQuery1.sql - D:\NE4V2T7\USER (52))
SELECT
    k.contract_id,
    k.client_id,
    ISNULL(k.real_consumption, 0) AS real_consumption_total,
    d.mes,
    ROUND(ISNULL(k.real_consumption, 0) * d.porcentaje, 2) AS consumo_mensual_kWh
FROM HolaluzCase.dbo.Kickout_Febrero2025 k
JOIN #DistribucionMensual d ON d.mes = '02';

Results
contract_id client_id real_consumption_total mes consumo_mensual_kWh
1 2 4986918 3864 02 386.4
2 3 5490075 2758 02 275.8
3 8 166505 2497 02 249.7
4 17 3864153 1947 02 194.7
5 21 442420 4331 02 433.1
6 22 9075002 3932 02 393.2
7 24 1865640 3491 02 349.1
8 27 4527921 5839 02 583.9
9 28 5485278 5204 02 520.4
10 29 1489471 5524 02 552.4
11 33 2636483 4555 02 455.5
12 39 9028365 5323 02 532.3
13 46 4766795 3627 02 362.7

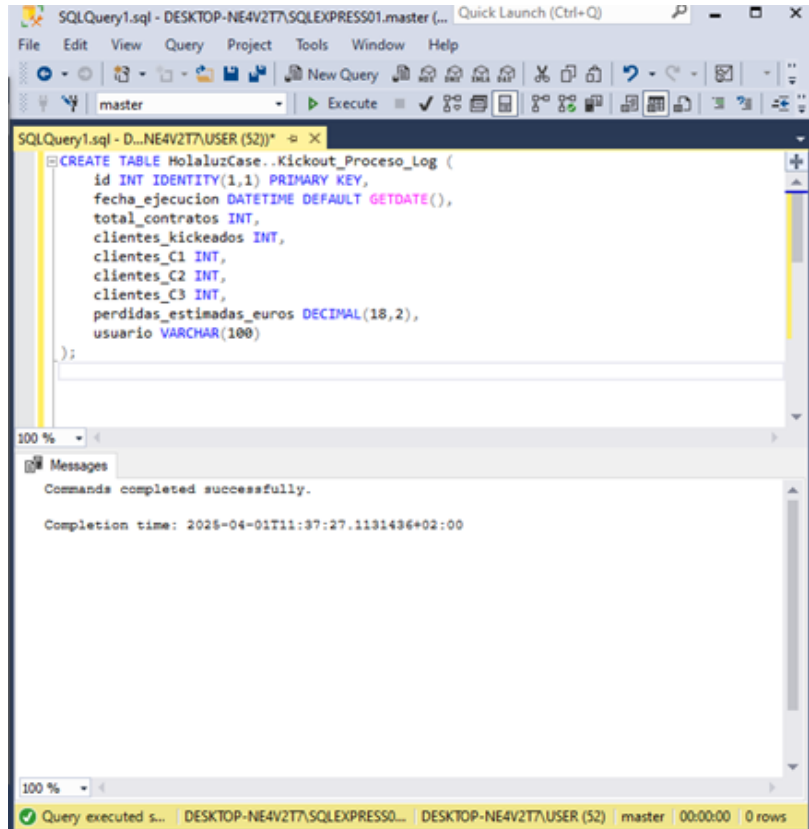
Query executed successfully. DESKTOP-NE4V2T7\SQLEXPRESS01.master DESKTOP-NE4V2T7\USER (52) master 00:00:00 286 rows

```



Con esta tabla podremos consultar el consumo mensual de un mes concreto, en este caso fue febrero (02), de la misma manera se podrá llenar reportes mensuales o dashboards de seguimiento de consumo y pérdidas.

c)



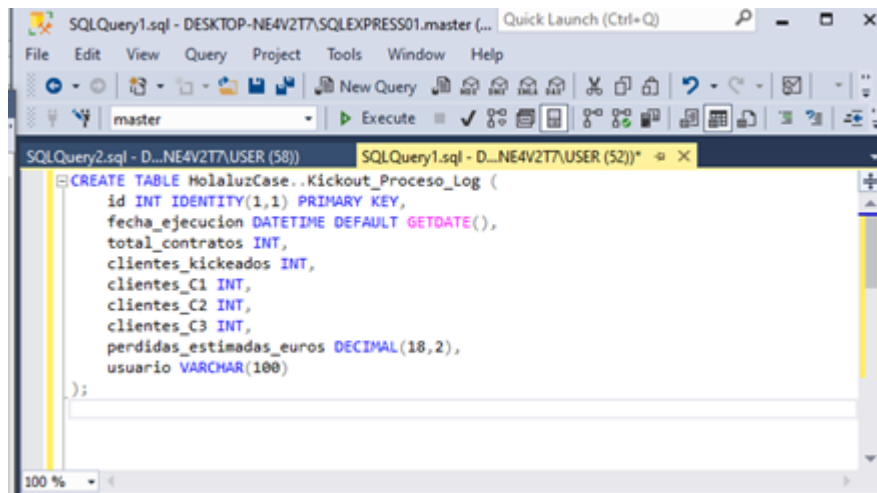
El campo `fecha_envio` se define como una columna de texto (`VARCHAR(6)`) con valor `'202502'`, indicando el año y mes y ahí será cuando se planea notificar al cliente del cambio de tarifa.

Con este valor se hará sencillo visibilizar la

- Trazabilidad histórica
- Agrupación mensual (comparar kick-outs mes a mes).
- Planificación operativa: alinear con campañas de correo electrónico por ejemplo.



d)

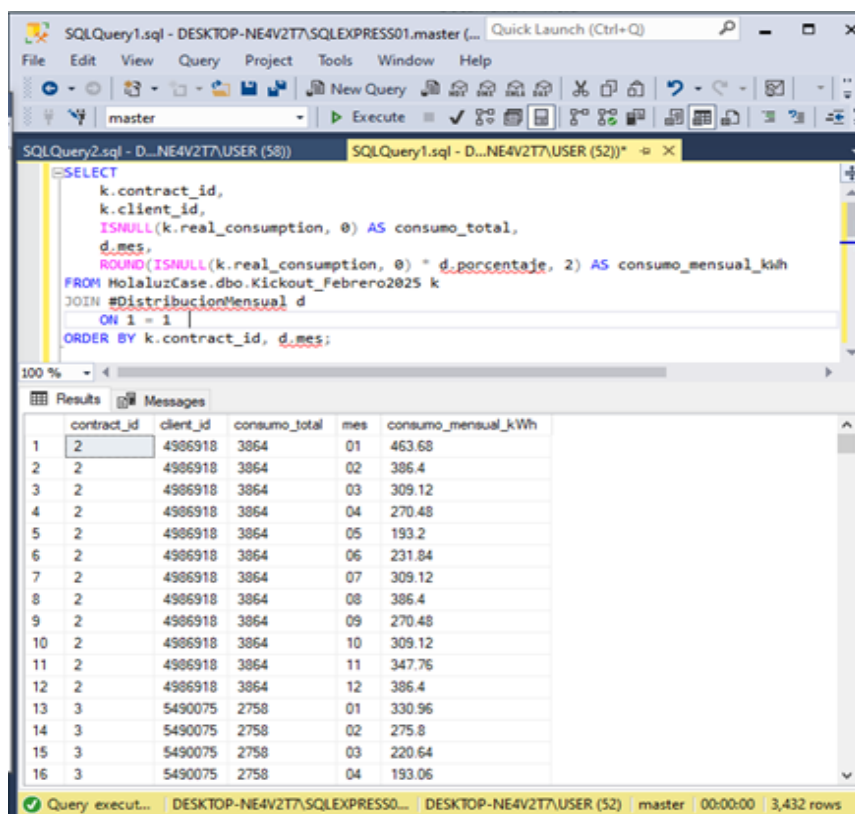


El campo **segmento** clasifica a los clientes en tres categorías:

- C1: Clientes que sí tienen instalación solar.
- C2: Clientes sin solar con una desviación superior o igual al 100%.
- C3: Clientes sin solar con desviación entre 30% y 99%.

Esta segmentación va a permitir personalizar la comunicación, priorizar acciones y analizar el impacto por tipo de cliente.

e)

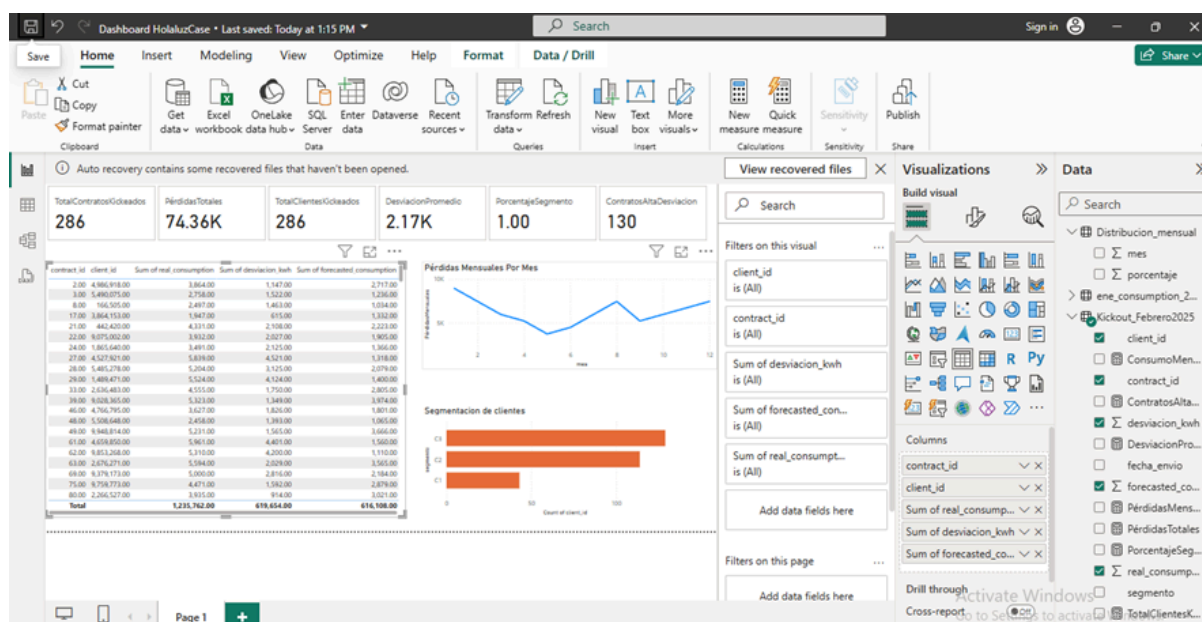


Con este query estamos multiplicando el consumo total real de cada contrato por el porcentaje de cada mes, así cada contrato aparece 12 veces. El campo **consumo\_mensual\_kWh** muestra el valor estimado de consumo para ese mes

## Conclusiones y mejoras

El proceso de Kick-Out actualmente presenta un desafío importante, ya que el análisis y la generación de la lista de clientes se realiza de forma manual. Para mejorar este flujo operativo, se propone la creación de procedimientos almacenados y tareas programadas mensualmente en SQL Server. Estos procedimientos permitirán generar automáticamente la tabla **Kickout\_<AÑO><MES>**, insertar la fecha de envío correspondiente, y registrar un log de auditoría con los detalles de la ejecución. Esta automatización incrementará la eficiencia del proceso, garantizará la consistencia mes a mes y reducirá significativamente los errores humanos.

Como complemento clave a esta automatización, se recomienda implementar un dashboard interactivo en Power BI conectado a SQL Server. Esta herramienta permitirá visualizar KPIs esenciales como el número de contratos evaluados, clientes en incumplimiento, distribución por segmento (C1, C2, C3) y las pérdidas económicas estimadas. Además, el dashboard ofrecerá análisis de consumo mensual, visualización de desviaciones por contrato, y filtros por producto, mes y segmento, mejorando así la toma de decisiones, la comunicación interdepartamental y la transparencia del proceso.



Otro desafío observado es la predicción y ajuste del consumo estimado conocido como forecast. Actualmente, muchos clientes incumplen debido a sobreestimaciones o cambios en sus patrones de uso. Se propone incorporar modelos predictivos que ajusten el **forecasted\_consumption** en base al historial de consumo, la estacionalidad y el perfil del cliente. Esta mejora reduciría la tasa de incumplimientos, aumentaría la precisión de las estimaciones y tendría un impacto positivo en el margen comercial de Holaluz.

En cuanto a la comunicación, se identificó que actualmente se trata de manera homogénea a clientes con características muy distintas. Esto puede afectar negativamente la eficacia del mensaje y la experiencia del cliente. La propuesta consiste en diseñar comunicaciones y acciones personalizadas por segmento (C1, C2, C3), lo cual permitiría una mayor adaptación a sus necesidades, reducción del churn y aumento en la conversión al cambio de tarifa.

Finalmente, se detecta la ausencia de un sistema de trazabilidad robusto. Para solucionarlo, se sugiere la implementación de una tabla de log (**Kickout\_Proceso\_Log**) que registre cada ejecución del proceso, incluyendo fecha, volumen de contratos analizados, clientes a kickear por segmento, pérdidas calculadas y el usuario que lo ejecutó. Esta solución aportará trazabilidad, cumplimiento normativo y mayor control sobre la calidad del proceso.