

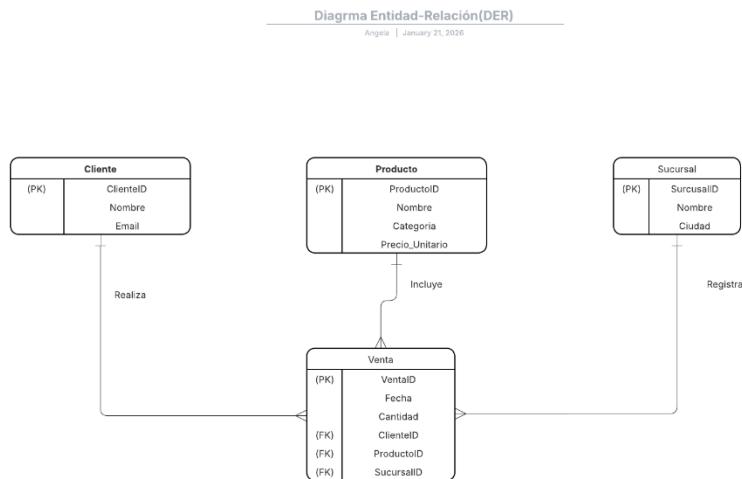
Introducción

En el presente trabajo se desarrolla un ejercicio práctico de diseño y análisis de bases de datos relacionales utilizando SQL Server, con el objetivo de aplicar conceptos de normalización, relaciones entre entidades y evaluación de desempeño de consultas SQL. El proyecto hace parte del reto propuesto, el cual busca fortalecer la comprensión del modelo relacional y el impacto que tienen las distintas estrategias de unión de tablas sobre el rendimiento del sistema.

Para ello, se diseñó un esquema relacional normalizado hasta la Tercera Forma Normal (3NF), separando la información en tablas maestras y una tabla transaccional, garantizando la integridad de los datos mediante el uso de claves primarias y foráneas. Posteriormente, se cargaron datos de prueba y se realizaron consultas SQL orientadas a comparar el uso de CROSS JOIN frente a INNER JOIN.

Finalmente, se llevó a cabo un laboratorio de desempeño en el cual se analizaron métricas como *Logical Reads* y tiempo de ejecución, permitiendo evidenciar de forma práctica cómo una mala elección de tipo de JOIN puede generar un uso ineficiente de recursos en la base de datos.

Diagrama Entidad-Relación(DER)



Captura de pantalla de los mensajes de SQL Server mostrando la reducción de *Logical Reads*.

- **CROSS JOIN**

RETO_5_6_DGD_2026

main.py | U docker-compose.yml | U USE RetoSQL:Untitled-2 | solution_tuning.sql | solution_tuning.sql U | solution_schema.sql U

EXPLORER

RETO_5_6_DGD_2026

01_data

raw

raw_sales_dump.csv

DICCIONARIO.cmd

02.sql

1. ddl diseño

solution schema.sql

template schema.sql

2. performance.lab

solution tuning.sql

template performance.sql

3. docs: teoria_y_guias

GUÍA_NORMALIZACION_3NF.md

GUÍA_PERFORMANCE_TUNING.md

app

SQL

Dockerfile

main.py

requirements.txt

data

.gitignore

docker-compose.yml

README.md

File Edit Selection View Go Run Terminal Help

RETO_5_6_dgd_2026

02.sql > 2> performance_lab > solution_tuning.sql

-- ESTADÍSTICAS TIME ON; -- Muestra tiempo de CPU

PRINT '>>> INICIO DEL BENCHMARK <<<'

-- SCENARIO 1: LA CONSULTA TÓXICA (CROSS JOIN)

PRINT '>>> EJECUTANDO CROSS JOIN (Producto Cartesiano) <<<'

PRINT '>>> Esta consulta genera un producto-cartesiano combinando todas las filas de Cliente con todas las de Producto.'

-- Esto simula el genero de desarrollador junior.

SELECT

-- C.Nombre AS Clientes,

-- P.Nombre AS Productos

FROM Cliente c

CROSS JOIN Producto p;

-- SCENARIO 2: LA CONSULTA EFICIENTE (INNER JOIN)

PRINT '>>> EJECUTANDO INNER JOIN (Ventas Reales) <<<'

PRINT '>>> Esta consulta recupera SOLO las ventas que realmente ocurrieron, usando las llaves foráneas.'

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

Results (1) Messages Open in New Tab

6:59:22 AM Started executing query at Line 5

SQL Server parse time: 0 ms, compile time: 0 ms, total time = 13 ms, elapsed time = 13 ms.

>>> INICIO DEL BENCHMARK <<<

SQL Server Execution Times:

CPU time = 4 ms, elapsed time = 4 ms.

>>> EJECUTANDO CROSS JOIN (Producto Cartesiano) <<<

SQL Server Execution Times:

CPU time = 0 ms, elapsed time = 0 ms.

(0 rows affected)

Table 'Cliente'. Scan count 1, logical reads 25, physical reads 0, page server read-ahead reads 0, lob logic 0 rows affected

Table 'Producto'. Scan count 1, logical reads 2, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logic 0 rows affected

SQL Server Execution Times:

CPU time = 27 ms, elapsed time = 26 ms.

Total execution time: 00:00:00.315

CHAT

Build with Agent

AI responses may be inaccurate.

Generate Agent instructions to onboard AI onto your codebase.

Localhost

RetosSQL

Results (1) Messages		
	Cliente ↕	Producto ↕
1	Iuisa fernanda	Laptop Gamer X
2	maria gomez	Laptop Gamer X
3	j balvin	Laptop Gamer X
4	Shakira Mebarak	Laptop Gamer X
5	Falcao Garcia	Laptop Gamer X
6	CLAUDIA BAHAMON	Laptop Gamer X
7	LAURA ACUÑA	Laptop Gamer X
8	james rodriguez	Laptop Gamer X
9	JORGE RODRIGUEZ	Laptop Gamer X
10	karol g	Laptop Gamer X
11	miguel angel	Laptop Gamer X
12	Carlos Ruiz	Laptop Gamer X
13	Sofia Vergara	Laptop Gamer X
14	feid	Laptop Gamer X
15	EGAN BERNAL	Laptop Gamer X
16	Pedro Pascal	Laptop Gamer X

- INNER JOIN

The screenshot shows the Reto 5.6_dgb_2026 environment in VS Code. The Explorer sidebar displays various files and folders related to the project, including `02.sql`, `02_ddl_demo`, `03_docs_teorica_y_guia`, and `app`. The `02.sql` file contains a SQL script for an INNER JOIN query:

```

21
22 -- ESCENARIO 2: LA CONSULTA EFICIENTE (INNER JOIN)
23
24 PRINT '--- EXECUTANDO INNER JOIN (Ventas Reales) ---';
25
26 -- Esta consulta recupera SOLO las ventas que realmente ocurrieron, usando las llaves foráneas.
27
28 SELECT
29     c.Nombre AS Cliente,
30     p.Nombre AS Producto,
31     v.Fecha,
32     v.Cantidad
33
34 FROM Venta v
35     INNER JOIN Cliente c ON v.ClienteID = c.ClienteID
36     INNER JOIN Producto p ON v.ProductoID = p.ProductoID;
37
38 -- COMPARACIÓN:
39 -- los logical Reads serán mínimos porque SQL Server usa los índices (PKs) para buscar directamente las filas relacionadas.
40 -- El número de filas devuelto es exacto a la realidad del negocio, no inflado artificialmente.
41
42 SET STATISTICS IO OFF;
43 SET STATISTICS TIME OFF;

```

The terminal window shows the execution results:

```

Started executing query at Line 25
SQL Server parse and compile time:
   CPU time = 2 ms, elapsed time = 2 ms.
--- EXECUTANDO INNER JOIN (Ventas Reales) ---

CPU time = 4 ms, elapsed time = 4 ms.
(200 rows affected)
Table 'Ventas'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, lob logic
Table 'Venta'. Scan count 1, logical reads 2, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logic
Table 'Producto'. Scan count 1, logical reads 2, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logic
Table 'Cliente'. Scan count 1, logical reads 2, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logic

CPU Server Execution Times:
   CPU time = 4 ms, elapsed time = 4 ms.
(200 rows affected)
Table 'Ventas'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logic
Table 'Venta'. Scan count 1, logical reads 2, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logic
Table 'Producto'. Scan count 1, logical reads 2, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logic
Table 'Cliente'. Scan count 1, logical reads 2, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logic

CPU Server Execution Times:
   CPU time = 162 ms, elapsed time = 185 ms.
Total execution time: 00:00:00.305

```

The status bar at the bottom indicates: In 35, Col 54 (64 selected) | Spaces:4 | UTF-8 | CRLF | SQL | MSSQL | SQLCMD Off | 200 rows affected | 00:00:00.305 | Count: 1 | Distinct Count: 1 | Null Count: 0 | localhost | RetoSQL | 702 x 462 | 21/01/2028.

	Cliente	Producto	Fecha	Cantidad
1	I Luisa fernanda	Laptop Gamer X	2024-02-27 00:00:00.000	2
2	I Luisa fernanda	Tarjeta de Video	2024-05-31 00:00:00.000	4
3	M maria gomez	Laptop Gamer X	2024-02-29 00:00:00.000	5
4	J j balvin	Laptop Gamer X	2024-06-15 00:00:00.000	5
5	S Shakira Mebarak	Monitor 24	2024-01-02 00:00:00.000	2
6	S Shakira Mebarak	Webcam HD	2024-02-25 00:00:00.000	3
7	I Luisa fernanda	Mouse Wireless	2024-04-01 00:00:00.000	3
8	J j balvin	Headset Pro	2024-05-17 00:00:00.000	1
9	F Falcao Garcia	Mouse Wireless	2024-06-09 00:00:00.000	5
10	C CLAUDIA BAHAMON	Memoria RAM 16GB	2024-01-12 00:00:00.000	2
11	L LAURA ACUÑA	Mouse Wireless	2024-04-07 00:00:00.000	3
12	J James Rodriguez	Tarjeta de Video	2024-04-04 00:00:00.000	3
13	M MARIO BORRIGUET	Monitor 24	2024-05-01 00:00:00.000	2

El uso de CROSS JOIN genera un producto cartesiano entre las tablas involucradas, combinando todos los registros de una tabla con todos los registros de la otra, independientemente de si existe o no una relación lógica entre ellos. Esto provoca un crecimiento innecesario del número de filas resultantes y un aumento significativo en las lecturas lógicas (*Logical Reads*), afectando negativamente el rendimiento del sistema.

Por el contrario, el INNER JOIN permite unir únicamente los registros que cumplen una condición de relación, normalmente basada en claves primarias y foráneas. Como se observó en los resultados del experimento, esta estrategia reduce drásticamente las lecturas lógicas y el tiempo de ejecución, ya que el motor de base de datos procesa solo la información relevante.

En conclusión, el INNER JOIN es la opción adecuada para trabajar con datos relacionales reales, mientras que el CROSS JOIN debe usarse únicamente en casos muy específicos y controlados, debido a su alto costo computacional.