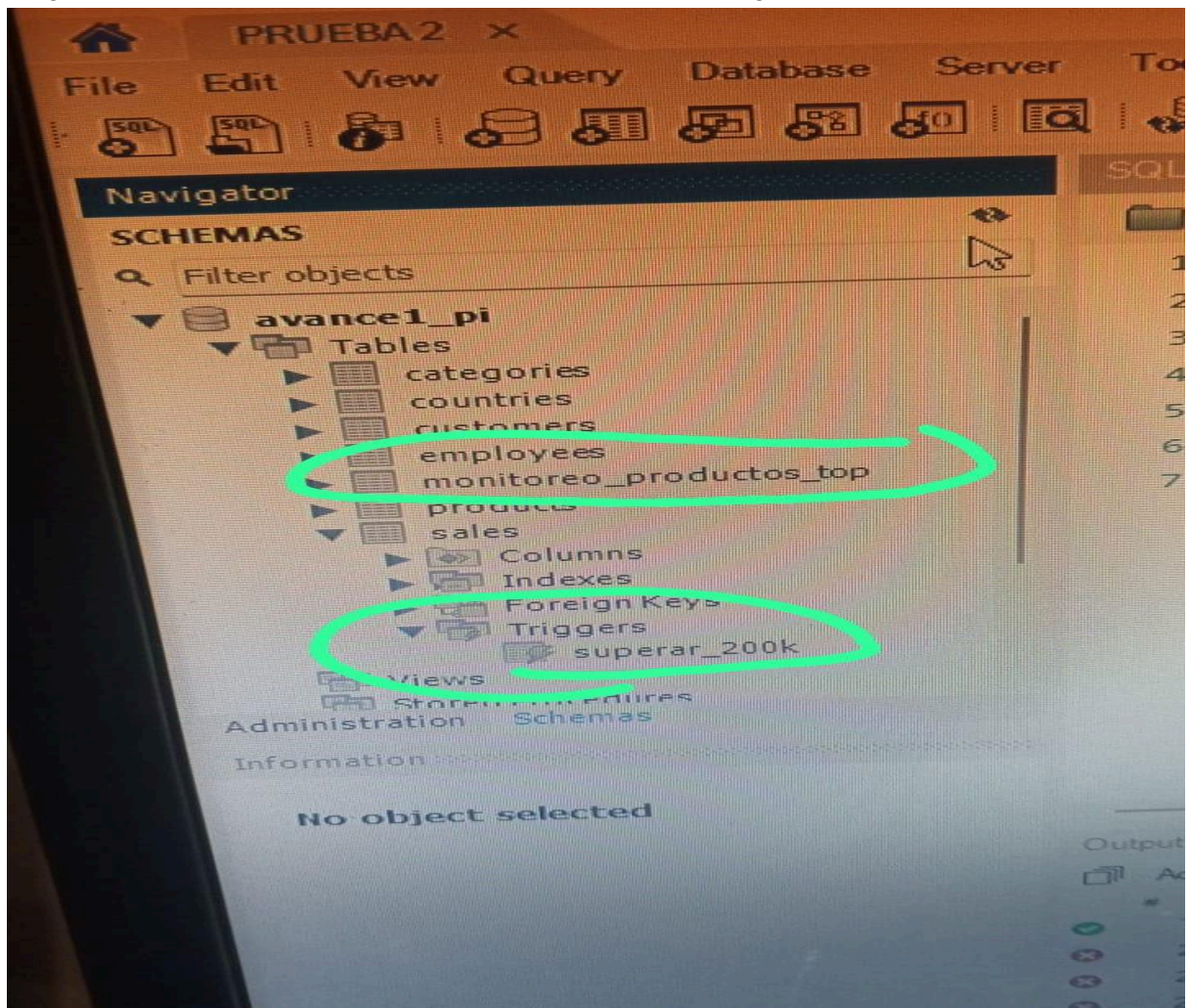


Monitoreo de Ventas y Optimización de Consultas

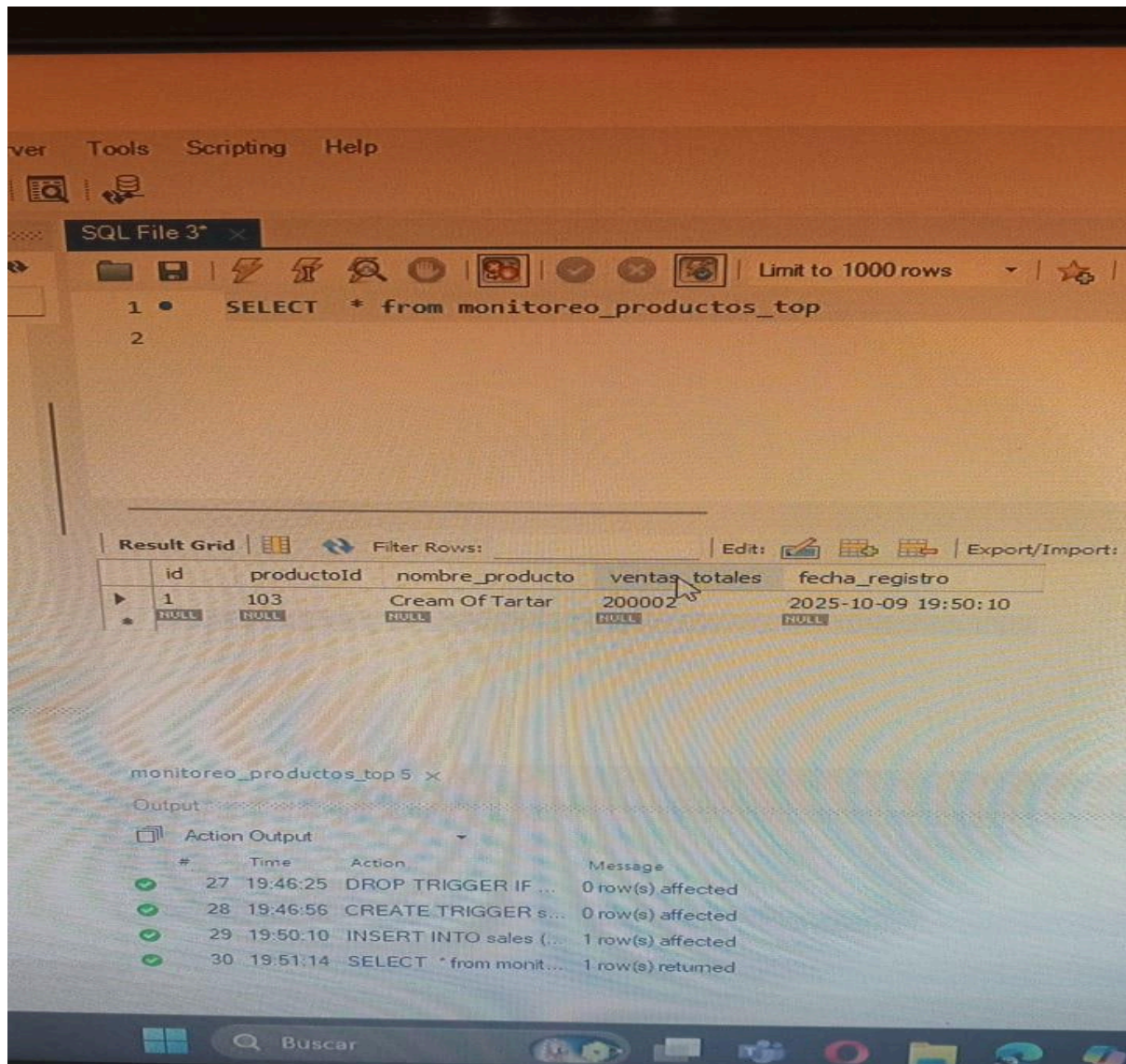
1. Creación de la tabla de monitoreo

- Se creó la tabla **monitoreo_productos_top** para registrar productos que superen las 200.000 unidades vendidas acumuladas.
- La tabla almacena: ID del producto, nombre del producto, total de ventas y fecha de registro.
- Se generó el TRIGGER para detectar los productos que lleguen a los 200mil ventas.



2. Registro de venta que supera las 200.000 unidades

- Se insertó una venta de prueba para un producto específico.
- El trigger se activó automáticamente y registró el producto en la tabla de monitoreo al superar el umbral.



Descripción:

Se confirma que el trigger funciona correctamente y que la información registrada es precisa: ID del producto, nombre, total de ventas y fecha.

3. Análisis de consulta antes de índices

- Se ejecutó la consulta para obtener los 5 productos más vendidos y el vendedor top por producto.
- Se utilizó EXPLAIN para analizar el plan de ejecución antes de aplicar índices.

The screenshot shows the Oracle Workbench interface. The SQL editor contains the following query:

```
34 r.total_sold
35 FROM ranked_sales r
36 JOIN top_products t ON r.product_id = t.product_id
37 JOIN employees e ON e.employee_id = r.employee_id
38 JOIN products p ON p.product_id = r.product_id
39 WHERE r.rn = 1
40 ORDER BY t.total_quantity DESC;
```

The 'Result Grid' tab is selected, displaying the EXPLAIN plan for the query. The plan consists of five steps:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	PRIMARY \$>	ALL	ALL	ALL	ALL	ALL	ALL	5	100.00	Using where; Using filesort
1	PRIMARY	p	ALL	eq_ref	PRIMARY	PRIMARY	4	t.product_id	1	100.00	Using where
1	PRIMARY	<derived2>	ALL	ref	<auto_key0>	<auto_key0>	5	t.product_id	6462	10.00	Using where
1	PRIMARY	e	ALL	eq_ref	PRIMARY	PRIMARY	4	r.employee_id	1	100.00	Using where
2	DERIVED	<derived3>	ALL	ALL	ALL	ALL	ALL	ALL	646146	100.00	Using filesort
3	DERIVED	<subquery4>	ALL	ALL	ALL	ALL	ALL	ALL	100.00	100.00	Using temporary
4	DERIVED	s	ALL	ALL	ALL	ALL	ALL	ALL	6461460	10.00	Using where; Using join buffer (hash join)
5	MATERIALIZED	<derived5>	ALL	ALL	ALL	ALL	ALL	ALL	5	100.00	Using where
5	DERIVED	sales	ALL	ALL	ALL	ALL	ALL	ALL	6461460	100.00	Using temporary; Using filesort

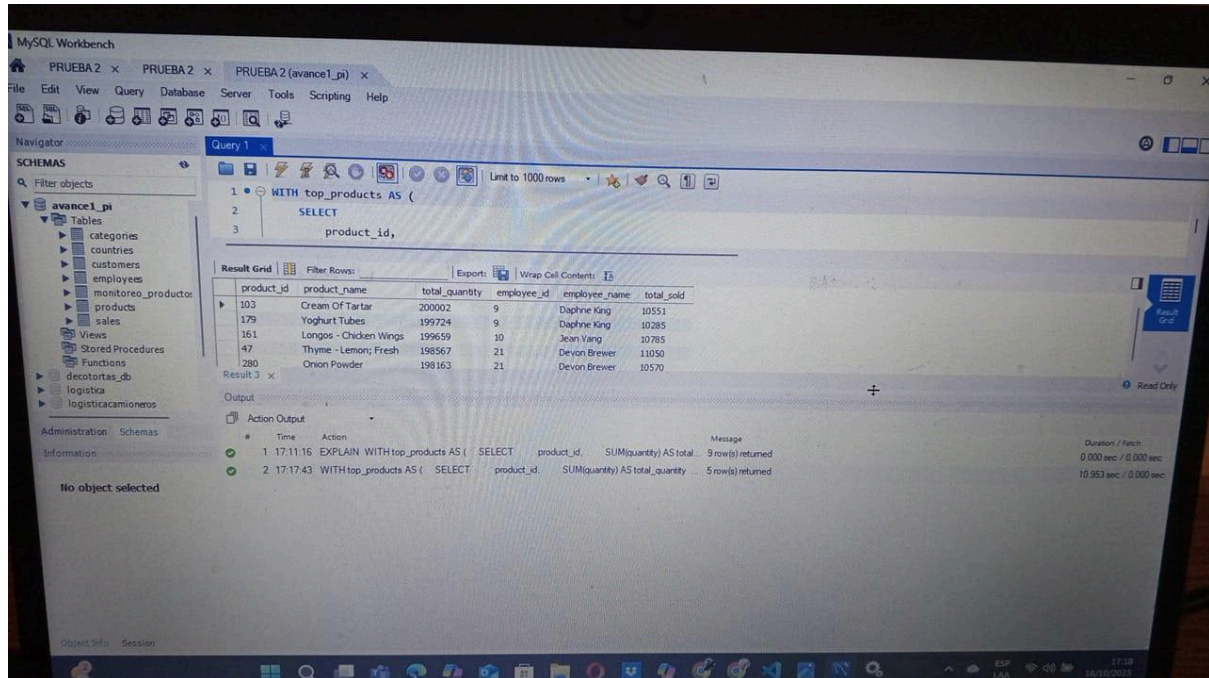
The 'Output' tab shows the execution of the EXPLAIN statement, indicating that the query was executed successfully at 17:11:16.

Descripción:

Se observa el tiempo de ejecución y la tabla detallada de EXPLAIN.

4. Ejecución de la consulta sin índices

- La consulta se ejecutó sin índices para registrar el tiempo de ejecución real.



Descripción:

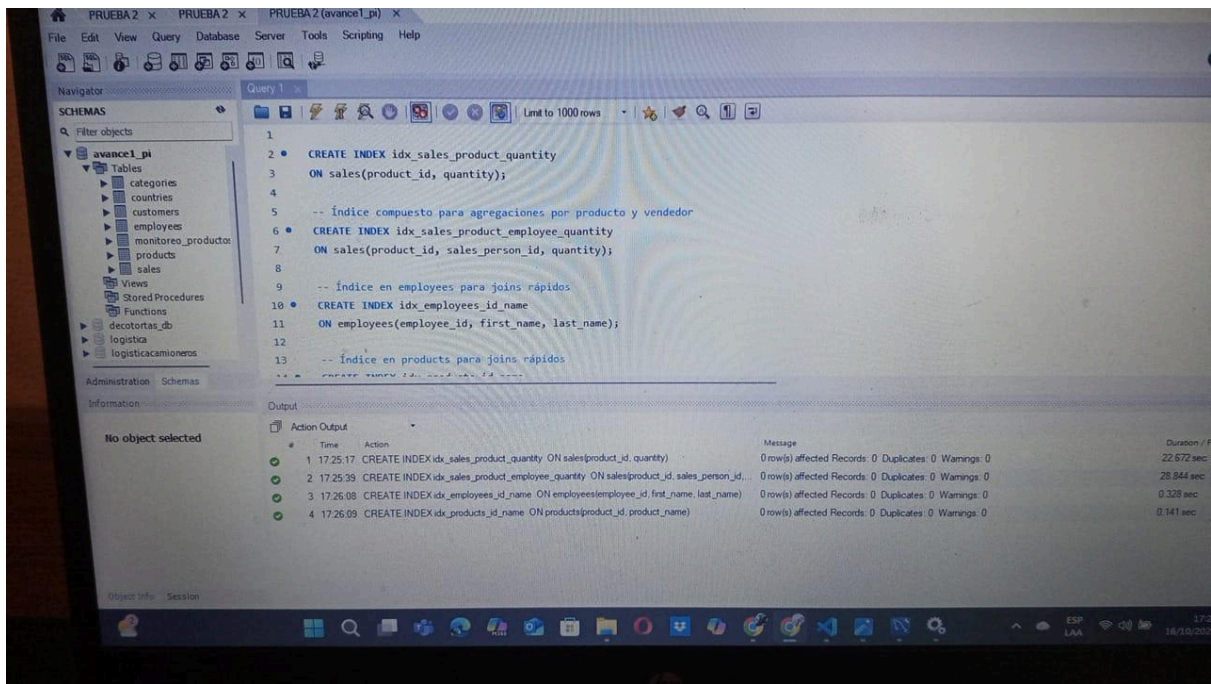
Se obtiene el tiempo que tarda la base de datos en procesar la consulta antes de aplicar optimizaciones y el resultado de la consulta.

5. Creación de índices

- Se crearon índices en las columnas más utilizadas en agregaciones y joins:

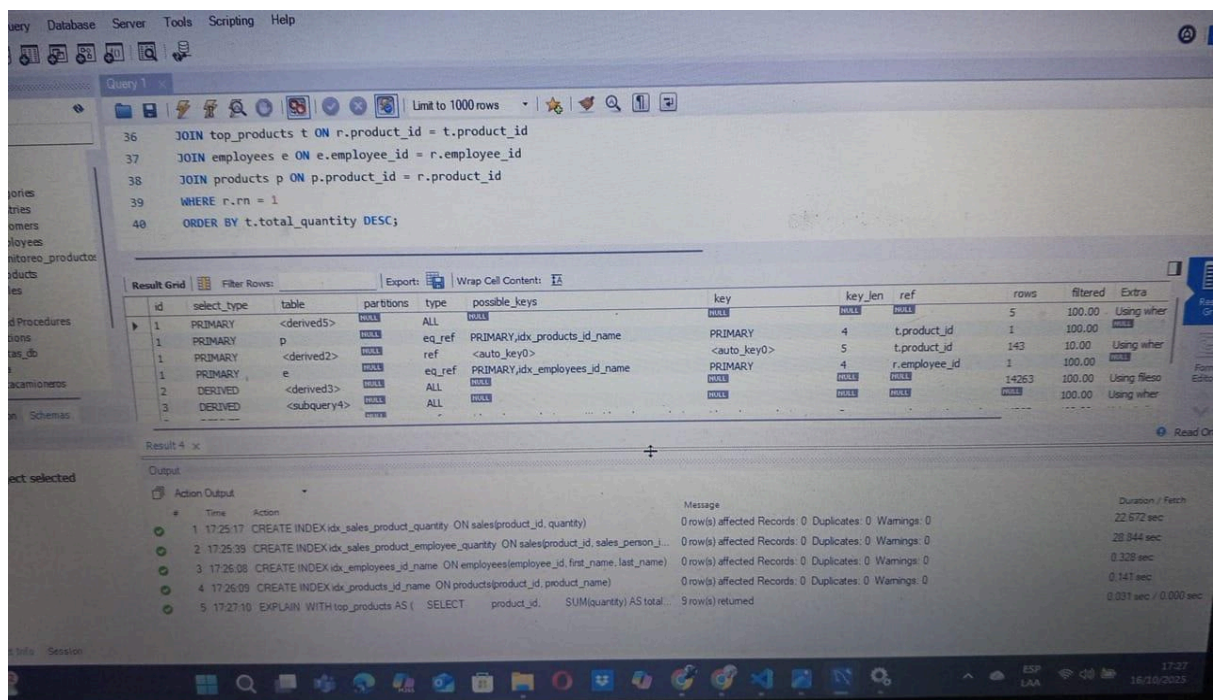
_Índices compuestos en **sales** para producto y vendedor.

_Índices en **employees** y **products** para optimizar joins.



6. Análisis de consulta después de índices

- Se volvió a ejecutar la consulta con EXPLAIN para analizar cómo los índices mejoran el plan de ejecución.

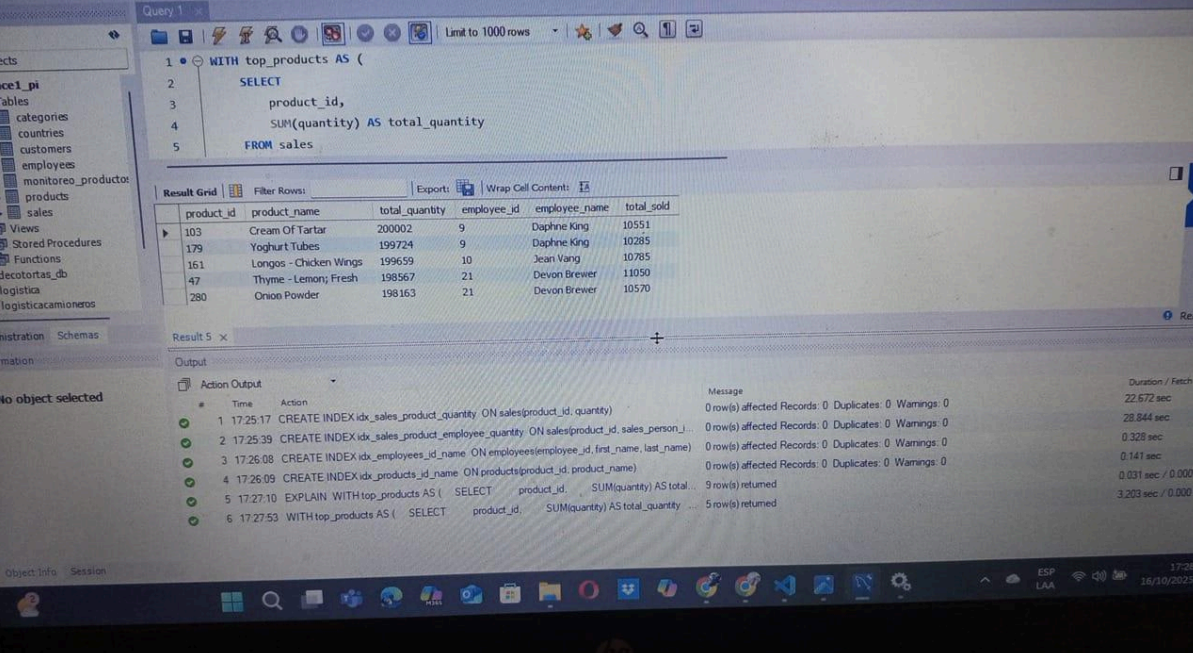


Descripción:

Se observa una reducción significativa en los tiempos de ejecución.

7. Ejecución de la consulta con índices

- Finalmente, se ejecutó la consulta sin EXPLAIN para verificar el tiempo de ejecución real después de aplicar los índices.



Query 1

```
1 WITH top_products AS (  
2     SELECT  
3         product_id,  
4         SUM(quantity) AS total_quantity  
5     FROM sales
```

product_id	product_name	total_quantity	employee_id	employee_name	total_sold
103	Cream Of Tartar	200002	9	Daphne King	10551
179	Yoghurt Tubes	199724	9	Daphne King	10285
161	Longos - Chicken Wings	199659	10	Jean Yang	10785
47	Thyme - Lemon; Fresh	198567	21	Devon Brewer	11050
280	Onion Powder	198163	21	Devon Brewer	10570

Result 5 x

Output

Time	Action	Message	Duration / Fetch
17:25:17	CREATE INDEX idx_sales_product_quantity ON sales(product_id, quantity)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	22.572 sec
17:25:39	CREATE INDEX idx_sales_product_employee_quantity ON sales(product_id, sales_person_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	28.844 sec
17:26:08	CREATE INDEX idx_employees_id_name ON employees(employee_id, first_name, last_name)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.328 sec
17:26:09	CREATE INDEX idx_products_id_name ON products(product_id, product_name)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.141 sec
17:26:09	CREATE INDEX idx_products_id_quantity ON products(product_id, SUM(quantity))	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec / 0.000 sec
17:27:10	EXPLAIN WITH top_products AS (SELECT product_id, SUM(quantity) AS total_quantity	9 row(s) returned	3.203 sec / 0.000 sec
17:27:53	WITH top_products AS (SELECT product_id, SUM(quantity) AS total_quantity	5 row(s) returned	

Descripción:

Se confirma que la optimización con índices reduce considerablemente el tiempo de procesamiento de la consulta.

8. Observaciones finales

- El trigger funciona correctamente y asegura el registro de productos que superan el umbral de ventas.
- Los índices mejoran notablemente el rendimiento de consultas complejas sobre tablas grandes.
- La combinación de análisis con EXPLAIN y medición de tiempos permite evaluar el impacto de la optimización de manera clara.

