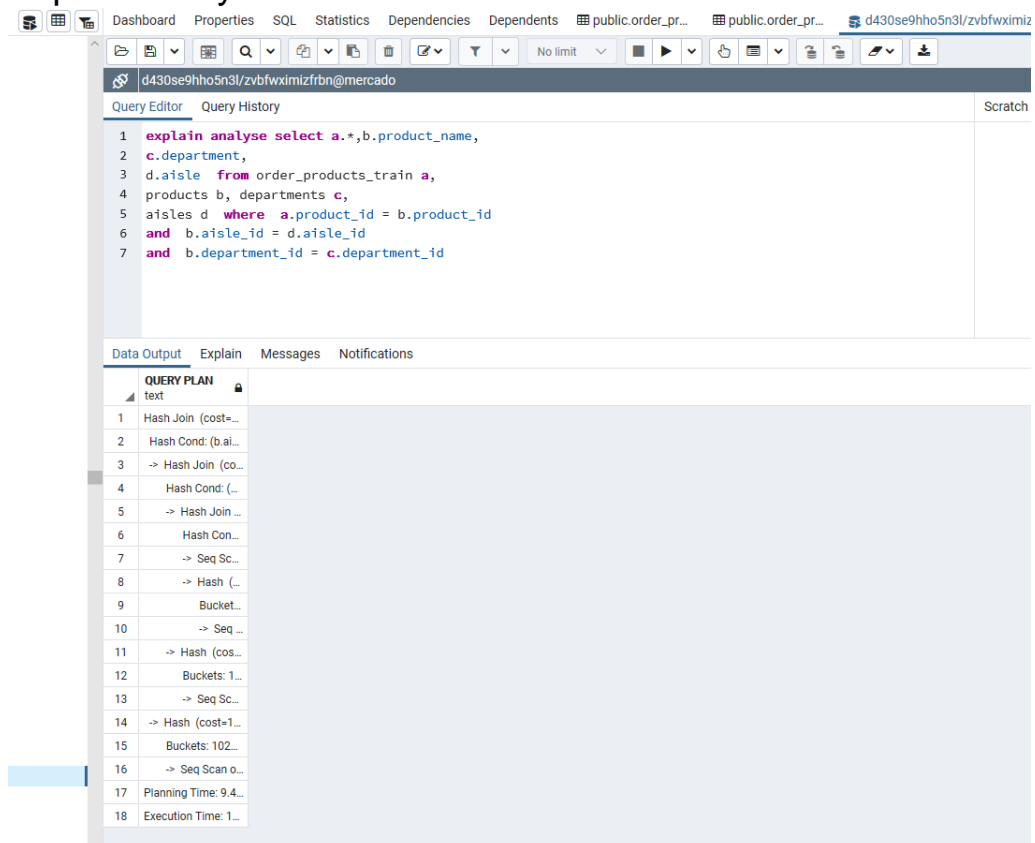


Fatec Carapicuíba – ADS 4

Programação para Banco de Dados

Evelyn Juliane Da Hora Sousa

- Explain Analyse com SEQ SCANS



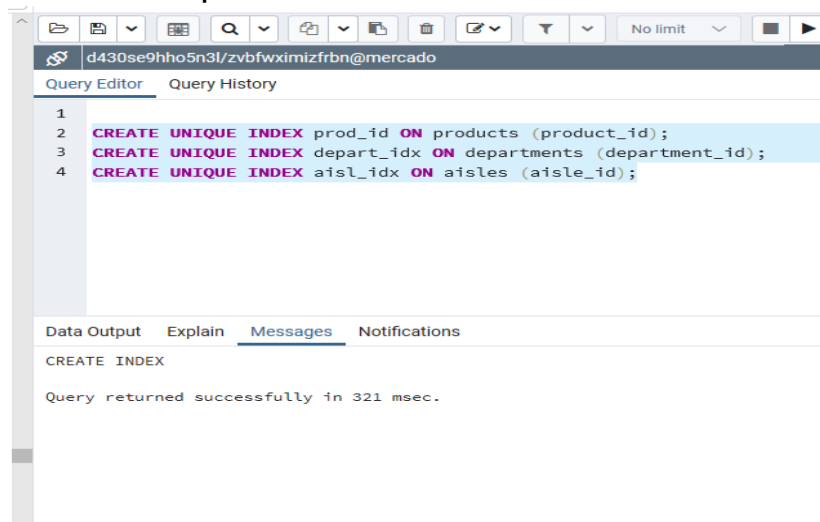
The screenshot shows a database query editor interface. The top menu bar includes Dashboard, Properties, SQL, Statistics, Dependencies, and Dependents. The main window displays an SQL query in the Query Editor:

```
1 explain analyse select a.*,b.product_name,  
2 c.department,  
3 d.aisle from order_products_train a,  
4 products b, departments c,  
5 aisles d where a.product_id = b.product_id  
6 and b.aisle_id = d.aisle_id  
7 and b.department_id = c.department_id
```

Below the query editor, the 'Data Output' tab is active, showing the 'QUERY PLAN' section. The plan consists of 18 steps:

- 1 Hash Join (cost=...
- 2 Hash Cond: (b.ai...
- 3 -> Hash Join (co...
- 4 Hash Cond: (...)
- 5 -> Hash Join ...
- 6 Hash Con...
- 7 -> Seq Sc...
- 8 -> Hash (...)
- 9 Bucket...
- 10 -> Seq ...
- 11 -> Hash (cos...
- 12 Buckets: 1...
- 13 -> Seq Sc...
- 14 -> Hash (cost=1...
- 15 Buckets: 102...
- 16 -> Seq Scan o...
- 17 Planning Time: 9.4...
- 18 Execution Time: 1...

- Create Unique Index



The screenshot shows a database query editor interface. The top menu bar includes Dashboard, Properties, SQL, Statistics, Dependencies, and Dependents. The main window displays three SQL commands in the Query Editor:

```
1  
2 CREATE UNIQUE INDEX prod_id ON products (product_id);  
3 CREATE UNIQUE INDEX depart_idx ON departments (department_id);  
4 CREATE UNIQUE INDEX aisl_idx ON aisles (aisle_id);
```

Below the query editor, the 'Messages' tab is active, showing the following output:

```
CREATE INDEX  
  
Query returned successfully in 321 msec.
```

Query Editor Query History

```

1 explain analyse select a.*,b.product_name,
2 c.department,
3 d.aisle from order_products_train a,
4 products b, departments c,
5 aisles d where a.product_id = b.product_id
6 and b.aisle_id = d.aisle_id
7 and b.department_id = c.department_id

```

Data Output Explain Messages Notifications

QUERY PLAN

text

1	Hash Join (cost=...
2	Hash Cond: (b.ai...
3	-> Hash Join (co...
4	Hash Cond: (...)
5	-> Hash Join ...
6	Hash Con...
7	-> Seq Sc...
8	-> Hash (...)
9	Bucket...
10	-> Seq ...
11	-> Hash (cos...
12	Buckets: 1...
13	-> Seq Sc...
14	-> Hash (cost=1...
15	Buckets: 102...
16	-> Seq Scan o...
17	Planning Time: 1.3...
18	Execution Time: 1...

```

1 SET enable_seqscan = ON;
2 explain analyse select a.*,b.product_name,
3 c.department,
4 d.aisle from order_products_train a,
5 products b, departments c,
6 aisles d where a.product_id = b.product_id
7 and b.aisle_id = d.aisle_id
8 and b.department_id = c.department_id

```

Data Output Explain Messages Notifications

Hash Join (cost=860.71..16082.07
rows=1384617 width=85) (actual
time=21.716..1543.925 rows=1384617
loops=1)

1 Hash Cond: (b.ai...

- SET enable_seqscan = OFF;

The screenshot shows a database query editor interface. At the top, there's a toolbar with various icons. Below it, the connection name 'd430se9hho5n3l/zvbfwximizfrbn@mercado' is displayed. The 'Query Editor' tab is active, showing a SQL query. The query is as follows:

```

1 SET enable_seqscan = OFF;
2
3 explain analyse select a.*,b.product_name,
4 c.department,
5 d.aisle from order_products_train a,
6 products b, departments c,
7 aisles d where a.product_id = b.product_id
8 and b.aisle_id = d.aisle_id
9 and b.department_id = c.department_id

```

Below the query, a tooltip for the 'Hash Join' operation is visible, showing the following details:

- Hash Join
- (cost=10000001198.89..10000016420.25
- rows=1384617 width=85) (actual
- time=394.511..1644.347 rows=1384617
- loops=1)

An 'OK' button is present in the tooltip. Below the query, the 'Data' tab is selected, showing a table with 19 rows. The first row is highlighted. The table contains the following data:

1	
2	Hash Cond: (b.ai...
3	-> Hash Join (co...
4	Hash Cond: (...
5	-> Hash Join ...
6	Hash Con...
7	-> Seq Sc...
8	-> Hash (...
9	Bucket...
10	-> Inde...
11	-> Hash (cos...
12	Buckets: 1...
13	-> Index S...
14	-> Hash (cost=6...
15	Buckets: 102...
16	-> Index Scan...
17	Planning Time: 0.5...
18	JIT:
19	Functions: 27

Podemos observar que após setarmos enable_seqscan como off o custo aumentou.

- **Create Unique Index para a maior tabela
(com set = ON)**

```

1  create INDEX order_idx on order_products_train (product_id);
2
3  explain analyse select a.*,b.product_name,
4  c.department,
5  d.aisle from order_products_train a,
6  products b, departments c,
7  aisles d where a.product_id = b.product_id
8  and b.aisle_id = d.aisle_id
9  and b.department_id = c.department_id

```

Hash Join (cost=860.71..16082.07
rows=1384617 width=85) (actual
time=25.163..1562.554 rows=1384617
loops=1)

1

2	Hash Cond: (b.ai...
3	-> Hash Join (co...
4	Hash Cond: (...)

OK

(Com SET = OFF)

```

1  SET enable_seqscan = OFF;
2  create INDEX order_idx on order_products_train
3
4  explain analyse select a.*,b.product_name,
5  c.department,
6  d.aisle from order_products_train a,
7  products b, departments c,
8  aisles d where a.product_id = b.product_id
9  and b.aisle_id = d.aisle_id
10

```

Hash Join (cost=1198.97..34234.93
rows=1384617 width=85) (actual
time=27.797..2733.021 rows=1384617
loops=1)

1

2	Hash Cond: (b.ai...
3	-> Hash Join (co...
4	Hash Cond: (...)

OK