

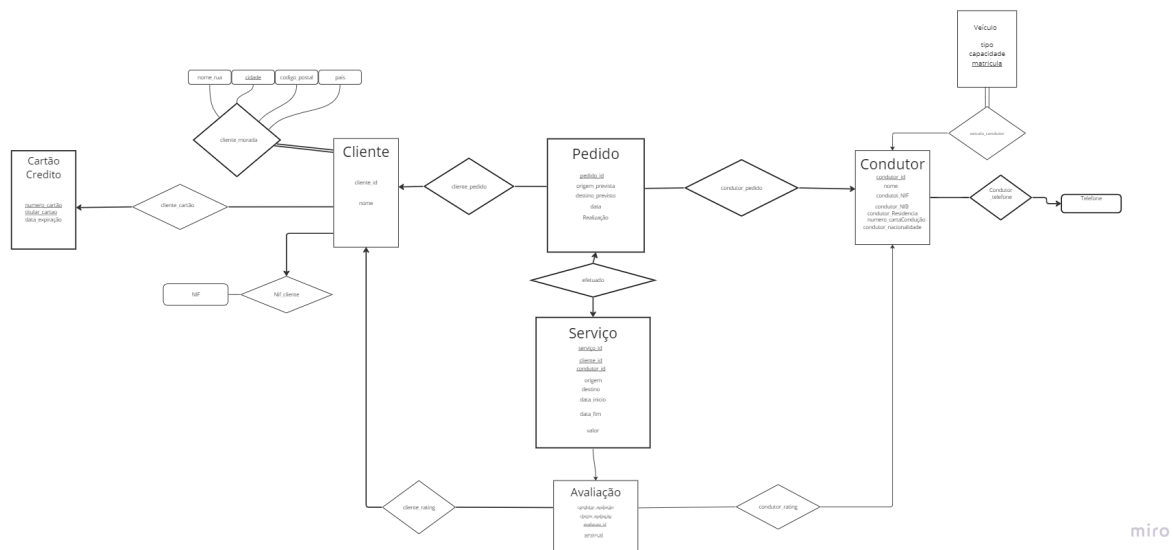
**Relatorio do Trabalho II de Base de dados**



UNIVERSIDADE  
DE ÉVORA

Trabalho Realizado por:  
Miguel Aleixo, 51653  
Antonio Carvalho, 51483  
João Trindade, 52241

## Pergunta 1:



## Pergunta 2:

### Entidades:

-->Cliente(cliente\_id,nome)

PRIMARY KEY -->cliente\_id

-->Condutor(condutor\_id,condutor\_NIF,numero\_cartaCondução,condutor\_NIB,Residencia,Nacionalidade,Nome)

PRIMARY KEY --> condutor\_id

-->CartãoCredito(numero\_cartão,titular\_cartão,data\_expiração)

PRIMARY KEY --> Titular\_cartão, numero\_cartão

-->Veiculo(capacidade,tipo,matricula)

PRIMARY KEY -->matricula

-->Pedido(cliente\_id,pedido\_id,origem\_prevista,destino\_previsto,data\_prevista,Realização)

PRIMARY KEY -->pedido\_id

FOREIGN KEY --> cliente\_id da entidade Cliente

-->Serviço(serviço\_id,cliente\_id,condutor\_id,origem,destino,data\_inicio,data\_fim,valor)

PRIMARY KEY --> serviço\_id

FOREIGN KEY --> cliente\_id da entidade Cliente

FOREIGN KEY -->condutor:\_id da entidade Condutor

-->Avaliação(condutor\_avaliação,cliente\_avaliação,serviço\_id,avaliação\_id)

PRIMARY KEY --> avaliação\_id

FOREIGN KEY --> serviço\_id da entidade Serviço

## Relações:

-->Cliente\_morada(cliente\_id,nome\_ rua,cidade,codigo\_postal,pais)

PRIMARY KEY --> cliente\_id

FOREIGN KEY --> cliente\_id da entidade Cliente

FOREING KEY --> nome\_ rua da entidade Morada

-->Cliente\_cartão(cliente\_id,numero\_cartão,titular\_cartão)

PRIMARY KEY --> cliente\_id,numero\_cartão,titular\_cartão

FOREIGN KEY --> cliente\_id da entidade Cliente

FOREING KEY --> numero\_cartão da entidade CartãoCredito

-->Cliente\_NIF(cliente\_id,NIF)

PRIMARY KEY --> cliente\_id

FOREIGN KEY --> cliente\_id da entidade Cliente

-->Veiculo\_condutor(condutor\_id,matricula)

PRIMARY KEY -->cliente\_id,matricula

FOREIGN KEY --> condutor\_id da entidade Condutor

FOREIGN KEY --> matricula da entidade Veiculo

-->Cliente\_Pedido(cliente\_id, pedido\_id)

PRIMARY KEY --> cliente\_id, pedido\_id

FOREIGN KEY --> cliente\_id da entidade Cliente

FOREIGN KEY --> pedido\_id da entidade Pedido

-->Condutor\_pedido(condutor\_id,pedido\_id)

PRIMARY KEY --> condutor\_id, pedido\_id

FOREIGN KEY --> condutor\_id da entidade Condutor

FOREIGN KEY --> pedido\_id da entidade Pedido

-->Condutor\_Telefone(condutor\_id,Telefone)

PRIMARY KEY --> condutor\_id,Telefone

FOREIGN KEY --> condutor\_id da entidade Condutor

-->Condutor\_avaliação(condutor\_id,avaliação\_id)

PRIMARY KEY --> condutor\_id, avaliação\_id

FOREIGN KEY --> condutor\_id da entidade Condutor

FOREIGN KEY --> avaliação\_id da entidade Avaliação

-->Cliente\_avaliação(cliente\_id, avaliação\_id)

PRIMARY KEY --> cliente\_id, avaliação\_id

FOREIGN KEY --> cliente\_id da entidade Condutor

FOREIGN KEY --> avaliação\_id da entidade Avaliação

-->Avaliação\_serviço(serviço\_id,avaliação\_id)  
PRIMARY KEY -->serviço\_id, avaliação\_id  
FOREIGN KEY --> serviço\_id da entidade Serviço  
FOREIGN KEY --> avaliação\_id da entidade Avaliação

### **Pergunta 3:**

**Cada cliente tem um cliente\_id único:**

cliente\_id → nome

**Cada cliente tem um NIF único:**

NIF → cliente\_id

**Cada condutor tem um condutor\_id único:**

nome → condutor\_id

**Cada condutor tem uma só carta de condução:**

numero\_cartaConducao → condutor\_id

**Cada condutor tem um NIF e um NIB únicos:**

condutor\_NIF, condutor\_NIB → cliente\_id

**Cada titular de um cartão tem um numero de cartão único:**

titular\_cartao → numero\_cartao

**A cada pedido está associado um cliente:**

pedido\_id → cliente\_id

**A cada serviço está associado um cliente e um condutor:**

serviço\_id → cliente\_id, condutor\_id

**Cada avaliação está associada a um serviço:**

avaliacao\_id → serviço\_id

**A cada condutor está associada uma matrícula do seu veículo:**

condutor\_id → matricula

**Cada avaliação de um condutor está associada a um cliente:**

condutor\_rating → cliente\_id

**Cada avaliação de um cliente está associada a um condutor:**

cliente\_rating → condutor\_id

**Cada cliente tem apenas uma morada de faturação:**

cliente\_id → nome\_ rua

**Várias ruas estão associadas a um código postal:**

nome\_ rua → codigo\_postal

#### **Pergunta 4**

{cliente\_id} → {nome}

{NIF} → {cliente\_id}

{cliente\_id} → {nome\_ rua}

{nome} → {condutor\_id}

{numero\_cartaConducao} → {condutor\_id}

{condutor\_NIF, condutor\_NIB} → {condutor\_id}

{condutor\_id} → {matricula}

{titular\_cartao} → {numero\_cartao}

{pedido\_id} → {cliente\_id}

{servico\_id} → {cliente\_id, condutor\_id}

{avaliacao\_id} → {servico\_id}

{condutor\_avaliacao} → {cliente\_id}

{cliente\_avaliacao} → {condutor\_id}

{nome\_ rua} → {codigo\_postal}

#### **Pergunta 5**

A base de dados já se encontra na forma normal Boyce-Codd

#### **Pergunta 7**

**1. Cliente:**

- Chave Primária: cliente\_id
- Chave Candidata: cliente\_id

- Chave Estrangeira: N/A
- 2. **Condutor:**
  - Chave Primária: condutor\_id
  - Chave Candidata: condutor\_id
  - Chave Estrangeira: N/A
- 3. **CartaoCredito:**
  - Chave Primária: Titular\_Cartao, Numero\_Cartao
  - Chave Candidata: Titular\_Cartao, Numero\_Cartao
  - Chave Estrangeira: N/A
- 4. **Residencia:**
  - Chave Primária: Nome\_Rua, Cidade
  - Chave Candidata: Nome\_Rua, Cidade
  - Chave Estrangeira: N/A
- 5. **Veiculo:**
  - Chave Primária: Matricula
  - Chave Candidata: Matricula
  - Chave Estrangeira: N/A
- 6. **Pedido:**
  - Chave Primária: Pedido\_id
  - Chave Candidata: Pedido\_id
  - Chave Estrangeira: cliente\_id (referenciando Cliente)
- 7. **Servico:**
  - Chave Primária: Servico\_id
  - Chave Candidata: Servico\_id
  - Chaves Estrangeiras: cliente\_id (referenciando Cliente), condutor\_id (referenciando Condutor)
- 8. **Avaliacao:**
  - Chave Primária: Avaliacao\_id
  - Chave Candidata: Avaliacao\_id
  - Chave Estrangeira: Servico\_id (referenciando Servico)

## **Relações:**

### **Cliente\_Morada:**

Chave Primária: cliente\_id

Chave Candidata: cliente\_id

Chave Estrangeira: cliente\_id (referenciando Cliente), nome\_rua, cidade (referenciando Residencia)

### **Cliente\_Cartao:**

Chave Primária: cliente\_id, numero\_cartao, titular\_cartao

Chave Candidata: cliente\_id, numero\_cartao, titular\_cartao

Chave Estrangeira: cliente\_id (referenciando Cliente), numero\_cartao, titular\_cartao (referenciando CartaoCredito)

**Cliente\_NIF:**

Chave Primária: cliente\_id

Chave Candidata: cliente\_id

Chave Estrangeira: cliente\_id (referenciando Cliente)

**Veiculo\_Condutor:**

Chave Primária: condutor\_id, matricula

Chave Candidata: condutor\_id, matricula

Chaves Estrangeiras: condutor\_id (referenciando Condutor), matricula (referenciando Veiculo)

**Cliente\_Pedido:**

Chave Primária: cliente\_id, pedido\_id

Chave Candidata: cliente\_id, pedido\_id

Chave Estrangeira: cliente\_id (referenciando Cliente), pedido\_id (referenciando Pedido)

**Condutor\_Pedido:**

Chave Primária: condutor\_id, pedido\_id

Chave Candidata: condutor\_id, pedido\_id

Chave Estrangeira: condutor\_id (referenciando Condutor), pedido\_id (referenciando Pedido)

**Condutor\_Telefone:**

Chave Primária: condutor\_id, telefone

Chave Candidata: condutor\_id, telefone

Chave Estrangeira: condutor\_id (referenciando Condutor)

**Condutor\_Avaliacao:**

Chave Primária: condutor\_id, avaliacao\_id

Chave Candidata: condutor\_id, avaliacao\_id

Chaves Estrangeiras: condutor\_id (referenciando Condutor), avaliacao\_id (referenciando Avaliacao)

#### Cliente\_Rating:

Chave Primária: cliente\_id, avaliacao\_id

Chave Candidata: cliente\_id, avaliacao\_id

Chaves Estrangeiras: cliente\_id (referenciando Cliente), avaliacao\_id (referenciando Avaliacao)

#### Avaliacao\_Rating:

Chave Primária: servico\_id, avaliacao\_id

Chave Candidata: servico\_id, avaliacao\_id

Chaves Estrangeiras: servico\_id (referenciando Servico), avaliacao\_id (referenciando Avaliacao)

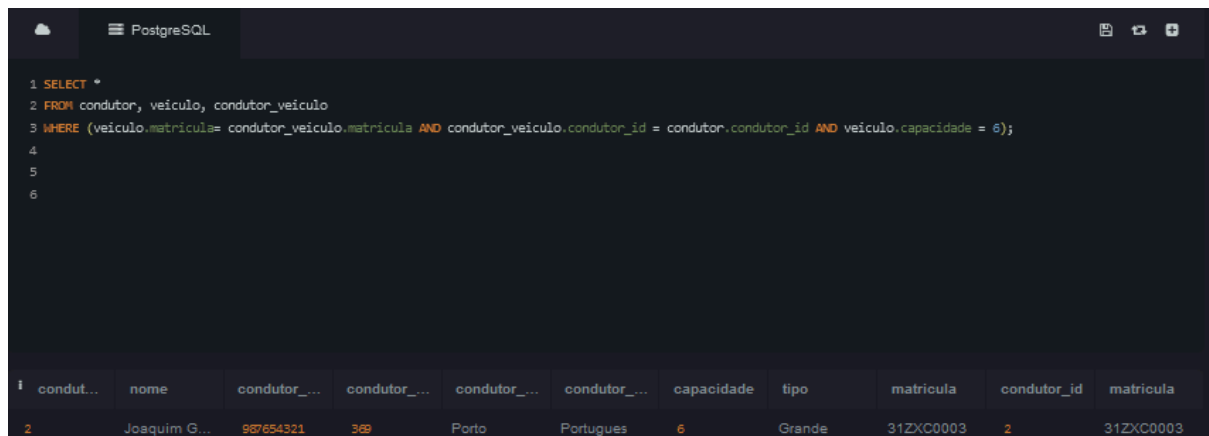
### Pergunta 10

a)

select \*

from condutor, veiculo, condutor\_veiculo

where (veiculo.matricula= condutor\_veiculo.matricula AND condutor\_veiculo.condutor\_id = condutor.condutor\_id AND veiculo.capacidade = 6);



The screenshot shows a PostgreSQL query editor with a dark theme. The query is as follows:

```
1 SELECT *
2 FROM condutor, veiculo, condutor_veiculo
3 WHERE (veiculo.matricula= condutor_veiculo.matricula AND condutor_veiculo.condutor_id = condutor.condutor_id AND veiculo.capacidade = 6);
4
5
6
```

Below the query, the results are displayed in a table with 11 columns: condut..., nome, condutor\_..., condutor\_..., condutor\_..., condutor\_..., capacidade, tipo, matricula, condutor\_id, matricula. The first row of data shows a driver named Joaquim G. with a license number 987654321, driving a vehicle with license 31ZXC0003.

condut...	nome	condutor_...	condutor_...	condutor_...	condutor_...	capacidade	tipo	matricula	condutor_id	matricula
2	Joaquim G...	987654321	369	Porto	Portugues	6	Grande	31ZXC0003	2	31ZXC0003

b)

select cliente\_id, condutor\_id

from servico

where servico.origem = 'Evora' AND servico.destino= 'Montemor' AND servico.data\_inicio > '2023/10/31 23:59:59' AND servico.data\_fim < '2023/12/01 00:00:00'



```

PostgreSQL

1 SELECT cliente_id, condutor_id
2 FROM servico
3 WHERE servico.origem = 'Evora' AND servico.destino = 'Montemor' AND servico.data_inicio > '2023/10/31 23:59:59' AND servico.data_fim < '2023/12/01 00:00:00'
4
5
6
7
8

```

cliente_id	condutor_id
2	1
2	2

c)

```

select cliente.nome
from cliente , (SELECT cliente.cliente_id as id, AVG(avaliacao.cliente_avaliacao) AS
                media_avaliacao
                FROM avaliacao
                JOIN cliente_rating ON avaliacao.avaliacao_id =
                cliente_rating.avaliacao_id
                JOIN cliente ON cliente_rating.cliente_id = cliente.cliente_id
                GROUP BY cliente.cliente_id) as avg5(id,media)
where avg5.media = 5 and avg5.id = cliente.cliente_id;

```

```

PostgreSQL

1 SELECT cliente.nome
2 FROM cliente , (SELECT cliente.cliente_id AS ID, AVG(avaliacao.cliente_avaliacao) AS
3 media_avaliacao
4 FROM avaliacao
5 JOIN cliente_rating ON avaliacao.avaliacao_id =
6 cliente_rating.avaliacao_id
7 JOIN cliente ON cliente_rating.cliente_id = cliente.cliente_id
8 GROUP BY cliente.cliente_id) AS avg5(ID,media)
9 WHERE avg5.media = 5 AND avg5.id = cliente.cliente_id;
10

```

nome
Nina Simone

d)

```

SELECT condutor_id
FROM condutor
WHERE condutor_id NOT IN (
    SELECT condutor_id
    FROM servico
    WHERE servico.destino ='Beja'
);

```

```
PostgreSQL

1 SELECT condutor_id
2 FROM condutor
3 WHERE condutor_id NOT IN (
4   SELECT condutor_id
5   FROM servico
6   WHERE servico.destino = 'Beja'
7 );
8

condutor_id
2
```

e)

select telefone

from cliente, condutor\_telefone, condutor, servico

where condutor\_telefone.condutor\_id = condutor.condutor\_id AND condutor.condutor\_id =  
servico.condutor\_id AND servico.destino = 'Viana do Castelo' AND servico.data\_inicio  
>'2023/12/20 00:00:00'AND servico.data\_inicio <'2023/12/21 00:00:00' AND  
servico.cliente\_id = cliente.cliente\_id AND cliente.nome = 'Manuel Santos'

```
PostgreSQL

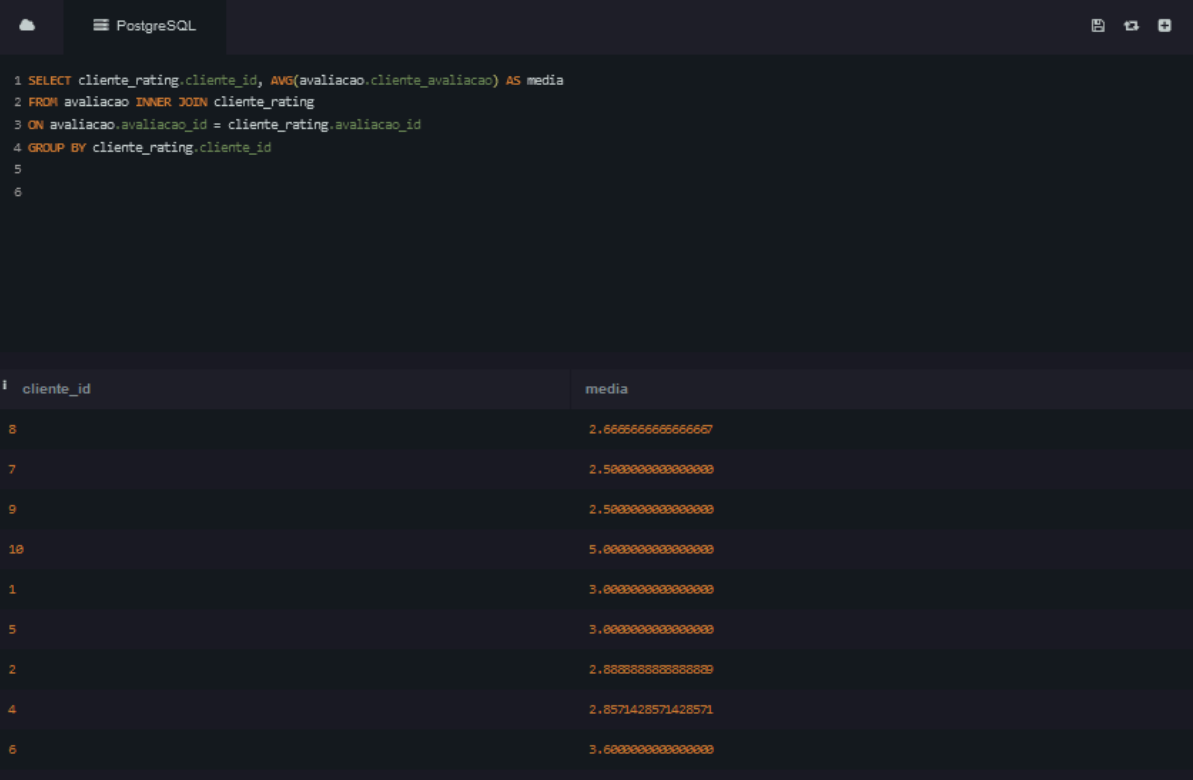
1 SELECT telefone
2 FROM cliente, condutor_telefone, condutor, servico
3 WHERE condutor_telefone.condutor_id = condutor.condutor_id AND condutor.condutor_id = servico.condutor_id AND servico.destino = 'Viana do Castelo' AND servico.data_
4
5
6
7

telefone

9123456
```

f)

select cliente\_rating.cliente\_id, AVG(avaliacao.cliente\_avaliacao) as media  
from avaliacao INNER JOIN cliente\_rating  
ON avaliacao.avaliacao\_id = cliente\_rating.avaliacao\_id  
group by cliente\_rating.cliente\_id



The screenshot shows a PostgreSQL query editor with a dark theme. The query is as follows:

```

1 SELECT cliente_rating.cliente_id, AVG(avaliacao.cliente_avaliacao) AS media
2 FROM avaliacao INNER JOIN cliente_rating
3 ON avaliacao.avaliacao_id = cliente_rating.avaliacao_id
4 GROUP BY cliente_rating.cliente_id
5
6

```

Below the query, the results are displayed in a table with two columns: `cliente_id` and `media`. The results are as follows:

cliente_id	media
8	2.666666666666667
7	2.500000000000000
9	2.500000000000000
10	5.000000000000000
1	3.000000000000000
5	3.000000000000000
2	2.888888888888889
4	2.8571428571428571
6	3.600000000000000
3	3.000000000000000

g)  
 SELECT servico.condutor\_id, COUNT(\*) as total\_services  
 FROM servico  
 GROUP BY condutor\_id;

PostgreSQL	
<pre> 1 SELECT servico.condutor_id, COUNT(*) AS total_services 2 FROM servico 3 GROUP BY condutor_id; </pre>	
condutor_id	total_services
1	14
3	11
5	13
4	10
2	7

h)

with avgc(avgc,cliente) as(select avg(avaliacao.condutor\_avaliacao) as avgc,  
 cliente\_rating.cliente\_id as cliente  
 from cliente\_rating, avaliacao, condutor\_avaliacao,condutor  
 where avaliacao.avaliacao\_id=cliente\_rating.avaliacao\_id and  
 condutor\_avaliacao.avaliacao\_id = avaliacao.avaliacao\_id AND  
 condutor\_avaliacao.condutor\_id=condutor.condutor\_id AND  
 condutor.nome = 'Joaquim Gomes'  
 group by cliente\_rating.cliente\_id)

select cliente.nome  
 from cliente, (SELECT max(avgc) as avgmax  
                   from avgc) as maxavg(avgmax), avgc as a2  
 where maxavg.avgmax = a2.avgc and a2.cliente= cliente.cliente\_id

```
PostgreSQL

1 with avgc(avgc,cliente) AS(SELECT AVG(avaliacao.condutor_avaliacao) AS avgc, cliente_rating.cliente_id AS cliente
2 FROM cliente_rating, avaliacao, condutor_rating,condutor
3 WHERE avaliacao.avaliacao_id=cliente_rating.avaliacao_id AND
4 condutor_rating.avaliacao_id = avaliacao.avaliacao_id AND
5 condutor_rating.condutor_id=condutor.condutor_id AND
6 condutor.nome = 'Joaoquin Gomes'
7 GROUP BY cliente_rating.cliente_id)
8
9 SELECT cliente.nome
10 FROM cliente, (SELECT MAX(avgc) AS avgmax
11 FROM avgc) AS maxavg(avgmax), avgc AS a2
12 WHERE maxavg.avgmax = a2.avgc AND a2.cliente= cliente.cliente_id
13
14 nome
Maria Silva
```

i)

```
with canc(id,total_canceled) as (SELECT cliente_pedido.cliente_id as
id,COUNT(pedido.realizacao = FALSE) as total_canceled
FROM pedido, cliente_pedido
WHERE cliente_pedido.pedido_id = pedido.pedido_id
GROUP BY 1)
```

```
select cliente.nome
FROM cliente, (select max(total_canceled)
from canc) as maxc, canc as c2
where maxc=c2.total_canceled and c2.id=cliente.cliente_id
```

```
PostgreSQL

1 SELECT
2     cliente.nome,
3     COUNT(pedido.realizacao = FALSE) AS total_canceled
4 FROM
5     cliente, pedido, cliente_pedido
6 WHERE
7     cliente.cliente_id = cliente_pedido.cliente_id AND
8     cliente_pedido.pedido_id = pedido.pedido_id
9 GROUP BY
10     cliente.nome
11 ORDER BY
12     total_canceled DESC
13 LIMIT 1;
14 |
```

nome	total_canceled
Maria Silva	12

j) select distinct condutor.condutor\_id  
from condutor inner join servico on condutor.condutor\_id = servico.condutor\_id  
where servico.condutor\_id not in (  
select servico.condutor\_id  
from pedido inner join efetuado on pedido.pedido\_id = efetuado.pedido\_id  
inner join servico on efetuado.servico\_id = servico.servico\_id  
where servico.data\_inicio < '2023/12/20 09:00:00' AND servico.data\_fim > '2023/12/20  
10:00:00'  
);

```
PostgreSQL

1 SELECT DISTINCT condutor.condutor_id
2 FROM condutor INNER JOIN servico ON condutor.condutor_id = servico.condutor_id
3 WHERE servico.condutor_id NOT IN (
4 SELECT servico.condutor_id
5 FROM pedido INNER JOIN efetuado ON pedido.pedido_id = efetuado.pedido_id
6 INNER JOIN servico ON efetuado.servico_id = servico.servico_id
7 WHERE servico.data_inicio < '2023/12/20 09:00:00' AND servico.data_fim > '2023/12/20 10:00:00'
8 );
9

i condutor_id
4
3
1
5
```

k)

```
select servico.origem, servico.destino
from servico inner join (
    SELECT MAX(servico.valor) AS vmax
    FROM servico
    WHERE EXTRACT(MONTH FROM servico.data_inicio) = 12 AND
    EXTRACT(YEAR FROM servico.data_inicio) = 2023)
as maxvalor(vmax) on servico.valor = maxvalor.vmax
```

```

1 SELECT servico.origem, servico.destino
2 FROM servico INNER JOIN (
3 SELECT MAX(servico.valor) AS vmax
4 FROM servico
5 WHERE EXTRACT(MONTH FROM servico.data_inicio) = 12 AND
6 EXTRACT(YEAR FROM servico.data_inicio) = 2023)
7 AS maxvalor(vmax) ON servico.valor = maxvalor.vmax
8
9
--

```

origem	destino
Lisboa	Viana do Castelo

l)  
 with somavalor(idc,soma)as(SELECT servico.condutor\_id, SUM(servico.valor) AS soma  
 FROM servico  
 WHERE EXTRACT(MONTH FROM servico.data\_inicio) = 11 AND EXTRACT(YEAR FROM  
 servico.data\_inicio) = 2023  
 GROUP BY servico.condutor\_id)

select condutor.nome  
 from condutor, (select max(soma) as maxv  
                   from somavalor) as maxvalor(maxv), somavalor as s2  
 where maxvalor.maxv= s2.soma and s2.idc = condutor.condutor\_id

```

1
2 with somavalor(idc,soma)AS(SELECT servico.condutor_id, SUM(servico.valor) AS soma
3 FROM servico
4 WHERE EXTRACT(MONTH FROM servico.data_inicio) = 11 AND EXTRACT(YEAR FROM servico.data_inicio) = 2023
5 GROUP BY servico.condutor_id)
6
7 SELECT condutor.nome
8 FROM condutor, (SELECT MAX(soma) AS maxv
9                 FROM somavalor) AS maxvalor(maxv), somavalor AS s2
10 WHERE maxvalor.maxv= s2.soma AND s2.idc = condutor.condutor_id
11

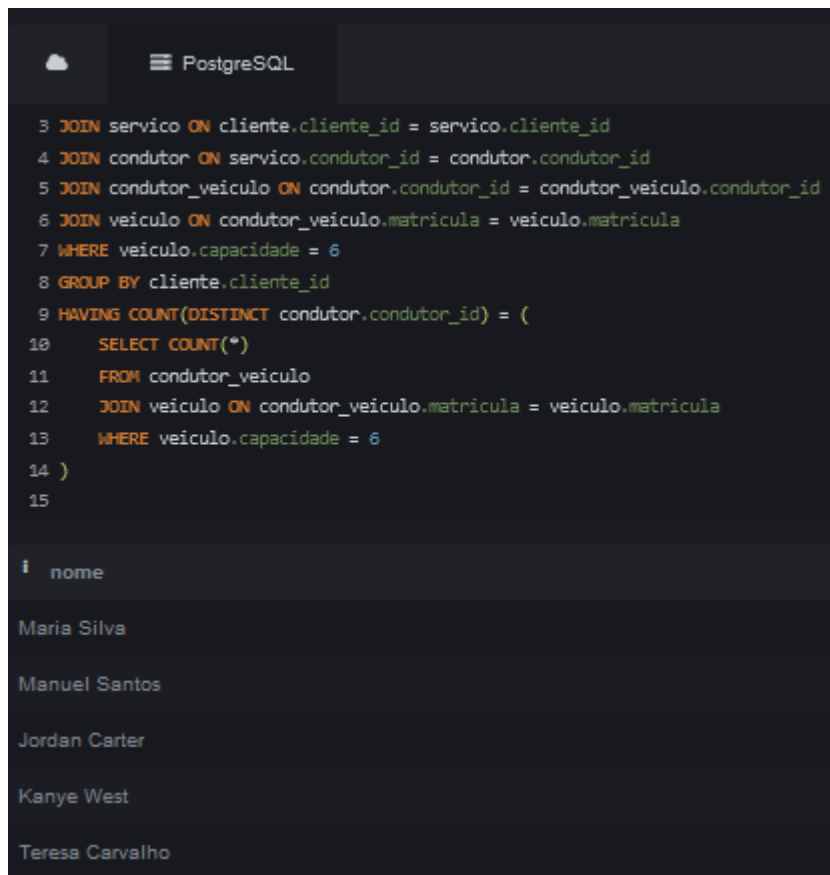
```

nome
João Carlos
Juan Cortez
José Dias



m)

```
SELECT cliente.nome
FROM cliente
JOIN servico ON cliente.cliente_id = servico.cliente_id
JOIN condutor ON servico.condutor_id = condutor.condutor_id
JOIN condutor_veiculo ON condutor.condutor_id = condutor_veiculo.condutor_id
JOIN veiculo ON condutor_veiculo.matricula = veiculo.matricula
WHERE veiculo.capacidade = 6
GROUP BY cliente.cliente_id
HAVING COUNT(DISTINCT condutor.condutor_id) = (
    SELECT COUNT(*)
    FROM condutor_veiculo
    JOIN veiculo ON condutor_veiculo.matricula = veiculo.matricula
    WHERE veiculo.capacidade = 6
)
```



The screenshot shows a PostgreSQL query editor with a dark theme. The query is a complex JOIN statement with a subquery in the HAVING clause. Below the query, the results are displayed in a table with one column named 'nome'. The results list five names: Maria Silva, Manuel Santos, Jordan Carter, Kanye West, and Teresa Carvalho.

```
3 JOIN servico ON cliente.cliente_id = servico.cliente_id
4 JOIN condutor ON servico.condutor_id = condutor.condutor_id
5 JOIN condutor_veiculo ON condutor.condutor_id = condutor_veiculo.condutor_id
6 JOIN veiculo ON condutor_veiculo.matricula = veiculo.matricula
7 WHERE veiculo.capacidade = 6
8 GROUP BY cliente.cliente_id
9 HAVING COUNT(DISTINCT condutor.condutor_id) = (
10     SELECT COUNT(*)
11     FROM condutor_veiculo
12     JOIN veiculo ON condutor_veiculo.matricula = veiculo.matricula
13     WHERE veiculo.capacidade = 6
14 )
15
```

nome
Maria Silva
Manuel Santos
Jordan Carter
Kanye West
Teresa Carvalho