



**ESCOLA DE CIÊNCIAS E TECNOLOGIA**  
**CURSO: ENGENHARIA INFORMÁTICA**  
**DISCIPLINA: BASES DE DADOS**  
**DOCENTES: PROF. IRENE RODRIGUES E PROF. LUÍS RODRIGUES**



**TRABALHO REALIZADO POR:**  
MARLENE OLIVEIRA Nº 25999

**28 DE NOVEMBRO DE 2010**

## ***1. Chaves Primárias e Chaves Estrangeiras***

### **Relação Motorista:**

- Chave Primária: Nbi;
- Chave Estrangeira: Não tem.

### **Relação Telefone:**

- Chave Primária: Telefone;
- Chave Estrangeira: Nbi (da relação Motorista).

### **Relação Modelo:**

- Chave Primária: Modelo;
- Chave Estrangeira: Não tem.

### **Relação Taxi:**

- Chave Primária: Matricula;
- Chave Estrangeira: Modelo (da relação Modelo).

### **Relação Serviço:**

- Chave Primária: Matricula, DataInicio;
- Chave Estrangeira: Matricula (da relação Taxi).

### **Relação Turno:**

- Chave Primária: Nbi, DataInicio;
- Chave Estrangeira: Matricula (da relação Taxi), Nbi (da relação Motorista).

### **Relação Cliente:**

- Chave Primária: Nif;
- Chave Estrangeira: Não tem.

### **Relação Pedido:**

- Chave Primária: Nif, MoradaInicio, DataInicio;
- Chave Estrangeira: Nif (da relação Cliente); Matricula, DataInicio (da relação Serviço).

## ***2. Comandos SQL para a criação das tabelas que compõem a base de dados.***

**Nota:** Assume-se que o modelo é unico para cada marca (por exemplo: a marca Mercedes não pode ter um modelo Civic). Alguns dos nomes que continham cedilhas ou acentos tiveram de ser ligeiramente alterados de modo a serem compatíveis com a codificação do PostgreSQL.

Estas tabelas já constam no ficheiro cria\_bd.txt.

### **Relação Motorista:**

```
create table Motorista(nome char(30) not null, NCartaCond char(15) not null, DataNasc date not null, Nbi integer unique not null check(Nbi>0), primary key(Nbi));
```

### Relação Telefone:

create table Telefone(Nbi integer not null check(Nbi>0),telefone integer unique not null check(telefone>0),primary key(telefone),foreign key(Nbi) references motorista on delete restrict);

### Relação Modelo:

create table Modelo(marca char(15) not null,modelo char(30) unique not null,Nlugares integer not null check(Nlugares>0),Consumo float not null check(consumo>0.0),primary key(modelo));

### Relação Táxi:

create table Taxi(modelo char(30) not null,ano integer not null check(ano>0),kms integer not null check(kms>0),matricula char(10) unique not null,primary key(matricula),foreign key(modelo) references modelo on delete restrict);

### Relação Serviço:

create table Servico(datainicio timestamp unique not null, datafim timestamp not null, kms float not null check(kms>0.0), valor float not null check(valor>0.0),matricula char(10) not null,primary key(matricula,datainicio),foreign key(matricula) references taxi on delete restrict);

### Relação Turno:

create table Turno(datainicio timestamp not null,datafim timestamp not null,kminicio integer not null check(kminicio>0),kmfim integer not null check(kmfim>0),matricula char(10) not null,Nbi integer not null check(Nbi>0),primary key(matricula,Nbi,datainicio),foreign key(matricula) references taxi on delete restrict,foreign key(Nbi)references motorista on delete restrict);

### Relação Cliente:

create table Cliente(nome char(30) not null,morada char(40) not null, codigopostal char(30) not null,nif bigserial unique not null check(Nif>0),primary key(nif));

### Relação Pedido:

create table Pedido(nif bigserial not null check(Nif>0),moradainicio char(30) not null,codigopostalinicio char(30) not null,datapedido timestamp not null,matricula char(10) not null,datainicio timestamp not null,primary key(nif,moradainicio,Datainicio),foreign key(nif) references cliente on delete restrict,foreign key(matricula,datainicio) references servico on delete restrict);

## **3. Código SQL para inserir informação na base de dados.**

Nota: Alguns os valores a inserir na base de dados que continham cedilhas ou acentos tiveram de ser ligeiramente alterados de modo a serem compatíveis com a codificação do PostgreSQL.

Estes valores já constam no ficheiro cria\_bd.sql, pelo que não é necessário introduzi-los.

a)

- i. insert into modelo values('Renault','Espace',7,7);  
insert into taxi values('Espace',2005,123098,'22-AA-22');
- ii. insert into modelo values('Mercedes','CLK',7,9);

insert into taxi values('CLK',2004,234554,'21-AA-22');

- iii. insert into modelo values('Honda','Civic',5,5);  
insert into taxi values('Civic',2006,89764,'20-AA-22');
- iv. insert into modelo values('Mercedes','Mercedes-Benz Classe S',5,6.5);  
insert into taxi values('Mercedes-Benz Classe S',2005,79744,'19-AA-22');

b)

- i. insert into motorista values('Manuel Duarte','L-123','14-1-1966',1234);  
insert into telefone values(1234,266262626);  
insert into telefone values(1234,939393939);
- ii. insert into motorista values('Fernando Nobre','L-124','14-1-1967',1235);  
insert into telefone values(1235,266262627);  
insert into telefone values(1235,939393940);
- iii. insert into motorista values('Anibal Silva','L-125','14-1-1968',1236);  
insert into telefone values(1236,266262628);  
insert into telefone values(1236,939393941);
- iv. insert into motorista values('Francisco Lopes','L-126','14-1-1968',1237);  
insert into telefone values(1237,266262629);

c)

- i. insert into cliente values('Jose Silva','Rua Antonio Silva 23','7100-131 Evora',600700800900);
- ii. insert into cliente values('Francisco Passos','Rua Manuel Passos 12','7100-131 Evora',600700800901);
- iii. insert into cliente values('Pedro Sousa','Rua Joaquim Sousa 21','7500-313 Evora',600700800902);

d)

- i. insert into pedido values(600700800900,'Rua Silva Pais 33','7120-212 Evora','2-1-2010 9:00:00','19-AA-22','2-1-2010 8:43:00');

e)

- i. insert into turno values('2-1-2010 8:00','2-1-2010 17:00',79744,79944,'19-AA-22',1234);
- ii. insert into turno values('2-1-2010 8:00','2-1-2010 17:00',89764,89964,'20-AA-22',1235);
- iii. insert into turno values('3-1-2010 8:00','3-1-2010 17:00',234554,234954,'21-AA-22',1236);
- iv. insert into turno values('3-1-2010 8:00','3-1-2010 17:00',123098,123498,'22-AA-22',1237);

£

- i. insert into servico values('2-1-2010 8:12:00','2-1-2010 8:32:00',12,5.25,'19-AA-22');
- ii. insert into servico values('2-1-2010 8:43:00','2-1-2010 8:52:00',7,3.25,'19-AA-22');
- iii. insert into servico values('2-1-2010 8:53:00','2-1-2010 9:59:00',98,53.25,'19-AA-22');
- iv. insert into servico values('2-1-2010 10:13:00','2-1-2010 10:29:00',18,19.25,'19-AA-22');
- v. insert into servico values('2-1-2010 11:10:00','2-1-2010 11:39:00',23,22.25,'19-AA-22');
- vi. insert into servico values('2-1-2010 12:00:00','2-1-2010 13:39:00',21,42.25,'19-AA-22');
- vii. insert into servico values('2-1-2010 15:20:00','2-1-2010 15:39:00',9,12.25,'19-AA-22');

#### ***4. Colocar mais informação na base de dados.***

Os valores seguintes foram inseridos para facilitar o teste das queries das perguntas. Estes valores já constam no ficheiro cria\_bd.txt.

Nota: Alguns dos valores a inserir na base de dados que continham cedilhas ou acentos tiveram de ser ligeiramente alterados de modo a serem compatíveis com a codificação do PostgreSQL.

- insert into pedido values(600700800901,'Rua Nova 44','7050-123 Montemor','2-1-2010 9:30:00','19-AA-22','2-1-2010 10:00:00');
- insert into servico values('2-1-2010 9:00','2-1-2010 10:30:00',15,13.50,'19-AA-22');
- insert into servico values('2-1-2010 11:00:00','2-1-2010 14:00:00',90,65.90,'20-AA-22');
- insert into turno values('3-2-2010 08:00:00','3-2-2010 17:00:00',99764, 100050,'20-AA-22',1235);
- insert into servico values('3-1-2010 9:00:00','3-1-2010 11:00:00',80,60,'22-AA-22');
- insert into servico values('2-1-2010 14:00:00','2-1-2010 15:00:00',40,30,'22-AA-22');
- insert into turno values('2-1-2010 8:00:00','2-1-2010 17:00:00',122050,123498,'22-AA-22',1237);
- insert into servico values('2-1-2010 10:00:00','2-1-2010 11:00:00',15,14,'19-AA-22');
- insert into pedido values(600700800901,'Rua Vasco da Gama 15','7050-345 Montemor','2-1-2010 10:50:00','20-AA-22','2-1-2010 11:00:00');
- insert into pedido values(600700800902,'Rua Alvares Pereira 7','7100-131 Evora','3-1-2010 8:45:00','22-AA-22','3-1-2010 9:00:00');
- insert into pedido values(600700800900,'Rua Serpa Pinto 10','7100-132 Evora','2-1-2010 13:45:00','22-AA-22','2-1-2010 14:00:00');
- insert into pedido values(600700800900,'Rua Alfredo Dinis 8','7050-500 Silveiras','2-1-2010 8:00:00','19-AA-22','2-1-2010 08:12:00');
- insert into pedido values(600700800902,'Rua Machado dos Santos 9','7000-456 Reguengos','2-1-2010 08:00:00','19-AA-22','2-1-2010 8:43:00');
- insert into pedido values(600700800901,'Rua Almirante Reis 45','7000-212 Evora','2-1-2010 08:40:00','19-AA-22','2-1-2010 08:53:00');
- insert into pedido values(600700800902,'Rua da Moagem 35','7200-123 Vendas Novas','2-1-2010 10:00:00','19-AA-22','2-1-2010 10:13:00');
- insert into pedido values(600700800900,'Rua Fernao de Magalhaes 78','7000-321 Evora','2-1-2010 10:50:00','19-AA-22','2-1-2010 11:10:00');
- insert into pedido values(600700800900,'Rua Marechal Gomes da Costa 23','7000-456 Evora','2-1-2010 11:40:00','19-AA-22','2-1-2010 12:00:00');
- insert into pedido values(600700800901,'Rua Fontes Pereira de Melo 6','7600-678 Beja','2-1-2010 15:00:00','19-AA-22','2-1-2010 15:20:00');

## 5. Expressões SQL que permitem responder às perguntas

Estas expressões constam no ficheiro perguntas.txt.

a)

```
select matricula
from taxi natural inner join modelo
where marca='Mercedes';
```

b)

```
select nome
from motorista natural inner join turno natural inner join taxi natural inner join modelo
where marca = 'Mercedes';
```

c)

```
select distinct telefone
from telefone natural inner join turno natural inner join (select matricula from pedido natural
inner join servico where nif=600700800900) as M
where turno.nbi = telefone.nbi;
```

d)

```
select matricula
from motorista natural inner join turno
where nome = 'Anibal Silva';
```

e)

```
(select nome from motorista)
except
(select nome from motorista natural inner join turno natural inner join (select matricula from
cliente natural inner join pedido natural inner join servico where nome ='Jose Silva' and pedido.nif
= cliente.nif) as M where turno.nbi = motorista.nbi);
```

f)

```
(select nome from motorista natural inner join turno natural inner join taxi natural inner join
modelo where turno.matricula = taxi.matricula and motorista.nbi = turno.nbi)
except
(select nome from motorista natural inner join turno natural inner join taxi natural inner join
modelo where marca = 'Mercedes' and turno.matricula = taxi.matricula and motorista.nbi =
turno.nbi);
```

g)

```
select nome, count(nbi) as num_servicos
from motorista natural inner join turno natural inner join (select matricula from servico)as
SM
group by nome;
```

h)

```
select nome,sum(valor) as lucro
from motorista natural inner join turno natural inner join (select matricula,valor from
```

servico) as SVM  
group by nome;

j)

```
select nome, receita
from (select nome,sum(valor)as receita,datainicio from motorista natural inner join turno
natural inner join (select matricula,valor,datainicio as D from servico)as S where turno.matricula =
s.matricula and S.d between turno.datainicio and turno.datafim and motorista.nbi = turno.nbi group
by nome, datainicio)as L
where receita = (select max(receita)as mais_lucro from (select nome,sum(valor) as
receita,datainicio from motorista natural inner join turno natural inner join (select
matricula,valor,datainicio as D from servico)as S where turno.matricula = S.matricula and S.D
between turno.datainicio and turno.datafim and motorista.nbi=turno.nbi group by
nome,datainicio)as ML);
```

j)

```
select matricula,marca,modelo
from (select matricula,marca,modelo, max(kmfim-kminicio) as maxKm from modelo
natural inner join taxi natural inner join turno group by matricula,marca,modelo)as MK
where maxkm = (select max(maxkm) as themax from (select
matricula,marca,modelo,max(kmfim-kminicio) as maxKm from modelo natural inner join taxi
natural inner join turno group by matricula,marca,modelo) as maxk);
```

k)

```
select avg(tempo_medio_espera)as tempo_medio_espera_companhia
from (select all nif,pedido.datapedido,servico.datainicio as inicio_servico,
matricula,avg(servico.datainicio - pedido.datapedido)as tempo_medio_espera from servico natural
inner join pedido group by nif,pedido.datapedido,servico.datainicio,matricula)as tempo_medio;
```

l)

```
select nome, max(num_pedidos)as maxpedidos
from cliente natural inner join (select nif,count(pedido.datapedido)as num_pedidos from
pedido group by nif) as NP
where cliente.nif = NP.nif and num_pedidos = (select max(num_pedidos) as MaxP from
(select nif,count(pedido.datapedido) as num_pedidos from pedido group by nif) as NP) group by
nome;
```

m)

**Nota:** Como não nos era fornecida informação quanto ao preço do combustível, assume-se o valor 1,30 como o preço por litro de combustível. Considera-se ainda que o taxi mais lucrativo da companhia é o que possui mais ganhos depois de ser retirado o gasto ao valor total dos ganhos.

```
select marca,modelo,matricula,(total_valor - gastokm)as ganhos
from modelo natural inner join taxi natural inner join (select matricula,(total_kms *
precokm)as gastokm from (select matricula,((consumo/100)*1.3)as precokm from modelo natural
inner join taxi)as Pkm natural inner join (select matricula,sum(kms)as total_kms from servico group
by matricula)as Tkm where Pkm.matricula = Tkm.matricula) as matprecokm natural inner join
(select matricula,sum(valor)as total_valor from servico group by matricula)as K where k.matricula
= matprecokm.matricula and (total_valor - gastokm) = (select max(ganhos)as maislucrativo from
(select matricula,(total_valor - gastokm)as ganhos from (select matricula,(total_kms * precokm)as
```

gastokm from (select matricula,((consumo/100)\*1.3)as precokm from modelo natural inner join taxi)as PKm natural inner join (select matricula,sum(kms)as total\_kms from servico group by matricula)as Tkm

where Pkm.matricula = Tkm.matricula) as matprecokm natural inner join (select matricula,sum(valor)as total\_valor from servico group by matricula)as K where k.matricula = matprecokm.matricula)as ganhos\_por\_taxi) and taxi.modelo = modelo.modelo;

n)

select distinct nome,km\_fora\_serv  
from motorista natural inner join turno natural inner join (select matricula, sum(total - kmsserv) as km\_fora\_serv from (select matricula,(kmfim - kminicio) as total from turno)as kmt natural inner join (select matricula,sum(kms)as kmsserv from servico group by matricula) as kms where kmt.matricula = kms.matricula group by matricula)as foradeservico where motorista.nbi = turno.nbi and turno.matricula = foradeservico.matricula and km\_fora\_serv>0;

o)

select nome  
from motorista natural inner join (select min,max,matricula from (select matricula,max,min,td,kd from (select max(kms),matricula,datainicio as kd from servico group by matricula,kd)as K natural inner join (select min(datafim-datainicio),matricula,datainicio as td from servico group by matricula,td)as T where t.td = k.kd and k.matricula = t.matricula group by matricula,max,min,td,kd)as MaxMin natural inner join pedido where pedido.matricula = MaxMin.matricula and pedido.datainicio = MaxMin.td and pedido.datainicio = MaxMin.kd and max = (select max(kms) from servico) group by matricula,min,max) as maxkm\_mintime natural inner join turno where turno.matricula = maxkm\_mintime.matricula and motorista.nbi = turno.nbi;