



**UNIVERSIDADE
DE ÉVORA**

1º Trabalho de Bases de Dados

**Companhia de Táxis
2019/2020**



Trabalho realizado por:

- Miguel Menúria, nº43566**
- Gonçalo Correia, nº43735**

Exercício 1: Chaves Candidatas, Primárias e Estrangeiras

motorista(Nome, NCartaCond, DataNasc, Nbi):

Chaves Candidatas: {NCartaCond}, {Nbi}

Chave Primária: {Nbi}

Chaves Estrangeiras: {}

telefone(Nbi, Telefone):

Chaves Candidatas: {Nbi}, {Telefone}

Chave Primária: {Nbi}

Chaves Estrangeiras: {Nbi}

modelo(Marca, Modelo, Nlugares, Consumo):

Chaves Candidatas: { Marca, Modelo}

Chave Primária: {Marca, Modelo}

Chaves Estrangeiras: {}

taxi(Modelo, Ano, Kms, Matricula):

Chaves Candidatas: {Matricula}

Chave Primária: {Matricula}

Chaves Estrangeiras: {}

serviço(DataInicio, DataFim, Kms, Valor, Matricula, CoordGPSInic, CoordGPSfim):

Chaves Candidatas: {DataInicio, Matricula}

Chave Primária: {}

Chaves Estrangeiras: {Matricula}

turno(DataInicio, DataFim, KmInicio, KmFim, Matricula, Nbi):

Chaves Candidatas: {DataInicio, Nbi}

Chave Primária: {DataInicio, Nbi}

Chaves Estrangeiras: {Matricula}, {Nbi}

cliente(Nome, Morada, CódigoPostal, Nif):

Chaves Candidatas: {Nif}

Chave Primária: {Nif}

Chaves Estrangeiras: {}

pedido(Nif, MoradaInicio, CódigoPostalInicio, DataPedido, matricula, DataInicio):

Chaves Candidatas: {Nif, DataPedido}

Chave Primária: {Nif, DataPedido}

Chaves Estrangeiras: {Matricula}, {Nif}

Exercício 2: Criação das tabelas de cada relação na base de dados

```
Create table motorista (  
    Nome varchar (25),  
    NcartaCond char (5),  
    DataNasc varchar (9),  
    Nbi integer primary key  
);
```

```
Create table telefone (  
    Nbi integer primary key,  
    Telefone varchar (50),  
    Foreign key (Nbi) references motorista on delete restrict  
);
```

```
Create table modelo (  
    Marca varchar (30),  
    Modelo varchar (30),  
    Nlugares integer,  
    Consumo varchar (30),  
    Primary key (Marca, Modelo)  
);
```

```
Create table taxi (  
    Modelo varchar (35),  
    Ano integer,  
    Kms integer,  
    Matricula char (8) primary key  
);
```

```
Create table serviço (  
    DataInicio TIMESTAMP NULL,  
    DataFim TIMESTAMP NULL,  
    Kms integer,  
    Valor float,  
    Matricula char (8),  
    CoordGPSInic varchar (30),  
    CoordGPSFim varchar (30),  
    Foreign key (Matricula) references taxi on delete restrict  
);
```

```
Create table turno (  
    DataInicio TIMESTAMP,  
    DataFim TIMESTAMP,  
    KmInicio integer,  
    KmFim integer,  
    Matricula char (8),  
    Nbi integer,  
    Primary key (DataInicio, Nbi),  
    Foreign key (Matricula) references taxi on delete restrict,  
    Foreign key (Nbi) references motorista on delete restrict  
);
```

```
Create table cliente1 (  
    Nome varchar (25),  
    Morada varchar (35),  
    CódigoPostal varchar (30),  
    Nif BIGINT primary key  
);
```

```
Create table pedido (  
    Nif BIGINT,  
    MoradaInicio varchar (35),  
    CódigoPostalInicio varchar (30),  
    DataPedido TIMESTAMP,  
    Matricula char (8),  
    DataInicio TIMESTAMP,  
    Primary key (Nif, DataPedido),  
    Foreign key (Matricula) references taxi on delete restrict,  
    Foreign key (Nif) references cliente1 on delete restrict  
);
```

Exercício 3: Inserção dos dados na BD

a) insert into modelo values ('Renault', 'Espace', 7, '71 aos 100');

insert into modelo values ('Mercedes', 'CLK', 7, '91 aos 100');

insert into modelo values ('Honda', 'Civic', 5, '51 aos 100');

insert into modelo values ('Mercedes', 'Classe S', 5, '6.51 aos 100');

insert into taxi values ('Renault Espace', 2015, 123098, '22-AA-22');

insert into taxi values ('Mercedes CLK', 2014, 234554, '21-AA-22');

insert into taxi values ('Honda Civic', 2012, 89764, '20-AA-22');

insert into taxi values ('Mercedes Classe S', 2015, 79744, '19-AA-22');

b) insert into motorista values ('Manuel Duarte', 'L-123', '14/1/76', 1234);

insert into motorista values ('Fernando Nobre', 'L-124', '14/1/77', 1235);

insert into motorista values ('Anibal Silva', 'L-125', '14/1/78', 1236);

insert into motorista values ('Francisco Lopes', 'L-126', '14/1/79', 1237);

c) insert into cliente1 values ('José Silva', 'Rua Antonio Silva 23', '7100-434 Évora', 600700800900);

insert into cliente1 values ('Francisco Passos', 'Rua Manuel Passos 12', '7000-131 Évora', 600700800901);

insert into cliente1 values ('Pedro Sousa', 'Rua Joaquim Sousa 21', '7500-313 Évora', 600700800902);

d) insert into pedido values (600700800900, 'Rua Silva Pais 33', '7120-212 Évora', '2/1/2016 9:00', '19-AA-22', '2/1/2016 8:43');

insert into pedido values (600700800901, 'Rua de Portel 18', 7005-397 Évora', '2/1/2016 9:15', '20-AA-22', '2/1/2016 9:37');

insert into pedido values (600700800902, 'Rua Pais Silva 35', '7005-747 Évora', '2/1/2016 11:30', '21-AA-22', '2/1/2016 12:03');

insert into pedido values (600700800901, 'Rua das Nogueiras 12', '7120-314 Évora', '3/1/2016 10:02', '21-AA-22', '3/1/2016 10:11');

insert into pedido values (600700800902, 'Rua de Évora 31', '7545-124 Évora', '3/1/2016 11:21', '19-AA-22', '3/1/2016 12:00');

insert into pedido values (600700800900, 'Rua Manuel Abreu 27', '7654-456 Évora', '3/1/2016 15:15', '20-AA-22', '3/1/2016 15:30');

e) insert into turno values ('2/1/2016 8:00', '2/1/2016 17:00', 79744, 79944, '19-AA-22', 1234);

insert into turno values ('2/1/2016 8:00', '2/1/2016 17:00', 89764, 89964, '20-AA-22', 1235);

insert into turno values ('3/1/2016 8:00', '3/1/2016 17:00', 234554, 234954, '21-AA-22', 1236);

insert into turno values ('3/1/2016 8:00', '3/1/2016 17:00', 123098, 123498, '22-AA-22', 1237);

f) insert into serviço values (NULL,NULL,NULL,NULL,NULL,NULL,NULL);

insert into serviço values ('2/1/2016 8:12', '2/1/2016 8:32', 12, 5.25, '19-AA-22', '0,75', '0,76');

insert into serviço values ('2/1/2016 8:43', '2/1/2016 8:52', 7, 3.25, '19-AA-22', '1,77', '1,77');

insert into serviço values ('2/1/2016 8:53', '2/1/2016 9:59', 98, 53.25, '19-AA-22', '2,78', '150,21');

insert into serviço values ('2/1/2016 10:13', '2/1/2016 10:29', 18, 19.25, '19-AA-22', '3,89', '33,50');

insert into serviço values ('2/1/2016 11:10', '2/1/2016 11:39', 23, 22.25, '19-AA-22', '5,47', '48,02');

insert into serviço values ('2/1/2016 12:00', '2/1/2016 13:39', 21, 42.25, '19-AA-22', '3,22', '103,44');

insert into serviço values ('2/1/2016 15:20', '2/1/2016 15:39', 9, 12.25, '19-AA-22', '6,55', '7,89');

Exercício 5: Expressões em SQL

a) select Matricula
from taxi
where modelo like '%Mercedes%'

b) Select Nome
From motorista, turno, taxi
where motorista.Nbi = turno.Nbi and modelo like '%Mercedes%'
and turno.Matricula = taxi.Matricula

c) Select Telefone
From telefone, motorista, pedido, turno
where pedido.Nif = 600700800900 and motorista.Nbi =
telefone.Nbi and pedido.Matricula = turno.Matricula and
motorista.Nbi = turno.Nbi

d)select Modelo
from motorista, taxi, turno
where motorista.Nome = 'Anibal Silva' and taxi.Matricula =
turno.Matricula and motorista.Nbi = turno.Nbi

e)select motorista.Nome
from motorista, pedido, cliente1, turno
where cliente1.Nome not in ('José Silva') and pedido.Nif =
cliente1.Nif and pedido.Matricula = turno.Matricula and
motorista.Nbi = turno.Nbi

f)select Nome
from motorista, taxi, turno
where taxi.Modelo not like '%Mercedes%' and taxi.Matricula =
turno.Matricula and turno.Nbi = motorista.Nbi

g)select Nome
from motorista
where not exists
((select Matricula
from taxi)
except
(select Matricula
from turno
where turno.Nbi = motorista.Nbi)
)

h)select Nome, count(serviço)
from motorista, turno, serviço
where turno.Matricula = serviço.Matricula and turno.Nbi =
motorista.Nbi
group by motorista.Nome

i)select Nome, sum(valor)
from motorista, turno, serviço
where turno.Matricula = serviço.Matricula and turno.Nbi =
motorista.Nbi
group by motorista.Nome

j)select Nome, max(valor)
from motorista, turno, serviço
where turno.Matricula = serviço.Matricula and turno.Nbi =
motorista.Nbi
group by motorista.Nome

k)select taxi.Matricula, Modelo, sum((-KmInicio) + KmFim)
from taxi, turno
where turno.Matricula = taxi.Matricula
group by Modelo, taxi.Matricula

l)select avg((DataInicio - DataPedido))
from pedido

m)select Nome, count(pedido)
from cliente1, pedido
where cliente1.Nif = pedido.Nif
group by Nome

n)with x as(select taxi.Matricula, taxi.Modelo,
sum((serviço.Kms)/(valor))
from taxi, serviço
where taxi.Matricula = serviço.Matricula
group by taxi.Matricula, taxi.Modelo)

select taxi.Matricula, taxi.Modelo, max(sum)
from x, taxi, serviço
where taxi.Matricula = serviço.Matricula
group by taxi.Matricula, taxi.Modelo

o)select Nome, sum(((KmInicio) + KmFim) - serviço.Kms)
from motorista, turno, serviço
where motorista.Nbi = turno.Nbi and serviço.Matricula = turno.Matricula
group by Nome

p)select Nome, sum ((serviço.Kms)/(serviço.DataFim-serviço.DataInicio))
from motorista, serviço, turno
where motorista.Nbi = turno.Nbi and serviço.Matricula = turno.Matricula
group by Nome

Exercício 5: Expressões em Álgebra Relacional

a) π Matricula (σ Modelo = 'Mercedes' (Taxi))

b) π Nome (σ motorista.NBI = turno.NBI \wedge Modelo = 'Mercedes' \wedge turno.Matricula = taxi.Matricula (motorista x turno x táxi))

c) π Telefone (σ pedido.Nif = 600700800900 \wedge motorista.Nbi = telefone.Nbi \wedge pedido.Matricula = turno.Matricula \wedge motorista.Nbi = turno.Nbi (telefone x motorista x pedido x turno))

d) π Modelo (σ motorista.Nome = 'Anibal Silva' \wedge taxi.Matricula = turno.Matricula \wedge motorista.Nbi = turno.Nbi (motorista x táxi x turno))

e) π motorista.Nome - π motorista.Nome (σ pedido.Nif = cliente1.Nif \wedge pedido.Matricula = turno.Matricula \wedge motorista.Nbi = turno.Nbi \wedge cliente1.Nome = 'José Silva' (motorista x pedido x cliente1 x turno))

f) π Nome (motorista- π Nome (σ taxi.Modelo = 'Mercedes' \wedge taxi.Matricula = turno.Matricula \wedge turno.Nbi = motorista.Nbi (motorista x taxi x turno))

h) Nome Gcount(serviço) (σ turno.Matricula = serviço.Matricula \wedge turno.Nbi = motorista.Nbi (motorista x turno x serviço))

i) Nome Gsum(Valor)(serviço) (σ turno.Matricula = serviço.Matricula \wedge turno.Nbi = motorista.Nbi (motorista x turno x serviço))

j) Nome Gmax(Valor)(serviço) (σ turno.Matricula = serviço.Matricula \wedge turno.Nbi = motorista.Nbi (motorista x turno x serviço))

k) taxi.Matricula, Modelo Gsum((-KmInicio) + KmFim) (turno) (σ turno.Matricula = taxi.Matricula (turno x táxi))

l) Gavg(DataInicio)(pedido) - Gavg(DataPedido)

m) NomeGcount(pedido) (σ cliente1.Nif = pedido.Nif (cliente1 x pedido))

n) π táxi.Matricula, táxi.Modelo, Gsum((serviço.Kms)/(Valor))(σ taxi.Matricula = serviço.Matricula)(taxi x modelo x serviço)

o) NomeGsum(((-KmInicio) + KmFim) - serviço.Kms) (motorista.Nbi = turno.Nbi \wedge serviço.Matricula = turno.Matricula)(motorista x turno x serviço)

p) π Nome, sum ((serviço.Kms)/(serviço.DataFim - serviço.DataInicio))(motorista.Nbi = turno.Nbi \wedge serviço.Matricula = turno.Matricula)(motorista x serviço x turno)