

Resultados Pretendidos

2.

<> -> theta join, **P** -> project, ***** -> natural join, **F** -> função de agregação, **/** → division

a)

```
A <- PESSOA <> CLIENTEVIAGEM
```

```
B <- A <> VIAGEM
```

```
C <-  $\sigma$ (nproprio = António AND apelido = Silva)B
```

```
RESULT <- P(noident, nif, idsistema, data, inicio, hfim, valfinal) C
```

OU

```
AntónioSilva <-  $\sigma$ (Pessoa.nproprio = António AND  
Pessoa.apelido = Silva)PESSOA
```

```
B <- AntónioSilva <> (AntónioSilva.id =  
CLIENTEVIAGEM.idpessoa) CLIENTEVIAGEM
```

```
C <- B <> VIAGEM
```

```
RESULT <- P(noident, nif, idsistema, data, inicio, hfim,  
valfinal) C
```

b) CondutoresHabilitados&Carros ← CONDUTORHABILITADO <>
(CONDUTORHABILITADO.veículo = VEÍCULO.id)VEÍCULO

```
ProprietárioBrunoFerreira ←  $\sigma$ (PESSOA.npróprio=Bruno AND  
PESSOA.apelido=Ferreira)
```

```
ProprietárioCarlosBrito ←  $\sigma$ (PESSOA.npróprio=Carlos AND  
PESSOA.apelido=Brito)
```

```
CondutoresHabilitadosFiltrados ←  
CondutoresHabilitados&Carros.proprietário =  
ProprietárioBrunoFerreira.id AND  
CondutoresHabilitados&Carros.proprietário !=  
ProprietárioCarlosBrito.id)
```

```
Result ← P(CondutoresHabilitadosFiltrados.npróprio,  
CondutoresHabilitadosFiltrados.nif)  
CondutoresHabilitadosFiltrados <>  
(CondutoresHabilitadosFiltrados.condutor = PESSOA.id)PESSOA
```

OU

```
CondutoresHabilitados ← CONDUTORHABILITADO <>
(CONDUTORHABILITADO.condutor = PESSOA.id) PESSOA

ProprietárioBrunoFerreira ← σ(PESSOA.npróprio=Bruno AND
PESSOA.apelido=Ferreira)

ProprietárioCarlosBrito ← σ(PESSOA.npróprio=Carlos AND
PESSOA.apelido=Brito)

Result ← P(CondutoresHabilitados.proprietário,
CondutoresHabilitados.npróprio,
CondutoresHabilitados.nif))) ∩
P(ProprietárioBrunoFerreira.id,
ProprietárioBrunoFerreira.apelido,
ProprietárioBrunoFerreira.nif) -
P(ProprietárioCarlosBrito.id,
ProprietárioCarlosBrito.npróprio,
ProprietárioCarlosBrito.nif)

c) Viagens2021 <- P(VIAGEM.data)σ(VIAGEM.data=2021)
CondutoresComViagens2021 <- VIAGEM / Viagens2021
NViagensCondutor2021 <- condutor F COUNT(condutor)
CondutoresComViagens2021
NMaxViagensCondutor2021 ← NViagensCondutor2021 <> (count=F
max(count) and NViagensCondutor2021.condutor =
PESSOA.id) PESSOA

d) CondutoresComViagens <- P(VIAGEM.condutor)VIAGEM
Condutores <- p(condutor)P(CONDUTOR.idpessoa)CONDUTOR
CondutoresSemViagens <- Condutores - CondutoresComViagens
DadosCondutoresSemViagens <- PESSOA <> (PESSOA.id =
CondutoresSemViagens.condutor)CondutoresSemViagens
Result <- P(PESSOA.id, PESSOA.npróprio, PESSOA.nif)
DadosCondutoresSemViagens

e) Viagens2021Condutor <- condutor F
COUNT(YEAR(VIAGEM.data)=2021) VIAGEM
```

```
Result <- P(nome, totalViagens)(PESSOA <> (PESSOA.id =  
Viagens2021Condutor.condutor) Viagens2021Condutor
```

OU

```
NotViagens2021 <- σ(YEAR(VIAGEM.data) != 2021) VIAGEM  
Viagens2021 <- VIAGEM - NotViagens2021  
Viagens2021condutor <- condutor F COUNT(data)  
Result <- P(nome, totalViagens)(PESSOA <> (PESSOA.id =  
Viagens2021Condutor.condutor) Viagens2021Condutor
```