

# SQL – LINGUAGEM DE MANIPULAÇÃO DE DADOS

## 1. FACULDADE

Considere a seguinte base de dados das classificações obtidas nas várias provas realizadas pelos alunos nas cadeiras de um ou mais cursos, com as tabelas e instâncias de seguida apresentadas:

ALUNO	
<u>nr</u>	Nome
100	João
110	Manuel
120	Rui
130	Abel
140	Fernando
150	Ismael

PROF	
<u>sigla</u>	Nome
ECO	Eugénio
FNF	Fernando
JLS	João

CADEIRA			
<u>cod</u>	Design	curso	regente
TS1	Teoria dos Sistemas 1	IS	FNF
BD	Bases de Dados	IS	ECO
EIA	Estruturas de Informação e Algoritmos	IS	ECO
EP	Electrónica de Potência	AC	JLS
IE	Instalações Eléctricas	AC	JLS

**PROVA**

<b>nr</b>	<b>cod</b>	<b>data</b>	<b>nota</b>
100	TS1	92-02-11	8
100	TS1	93-02-02	11
100	BD	93-02-04	17
100	EIA	92-01-29	16
100	EIA	93-02-02	13
110	EP	92-01-30	12
110	IE	92-02-05	10
110	IE	93-02-01	14
120	TS1	93-01-31	15
120	EP	93-02-04	13
130	BD	93-02-04	12
130	EIA	93-02-02	7
130	TS1	92-02-11	8
140	TS1	93-01-31	10
140	TS1	92-02-11	13
140	EIA	93-02-02	11
150	TS1	92-02-11	10
150	EP	93-02-02	11
150	BD	93-02-04	17
150	EIA	92-01-29	16
150	IE	93-02-02	13

A chave da tabela PROVA é constituída pelos atributos nr, cod e data, permitindo guardar o resultado de mais do que uma prova por cadeira. Admita que todos os alunos inscritos a uma cadeira fizeram pelo menos uma prova a essa cadeira. Responda em SQL às perguntas indicadas abaixo e obtenha a resposta no sistema de gestão de base de dados.

- Quais os números dos alunos?  
**SELECT nr FROM Aluno;**
- Qual o código e designação das cadeiras do curso 'AC'?  
**SELECT cod, design FROM Cadeira WHERE curso = 'AC';**
- Existem nomes comuns a alunos e profs? Quais?  
**SELECT nome FROM Aluno INTERSECT SELECT nome FROM Prof;**
- Quais os nomes específicos dos alunos, i.e., que nenhum professor tem?  
**SELECT nome FROM Aluno EXCEPT SELECT nome FROM Prof;**
- Quais os nomes das pessoas relacionadas com a faculdade?  
**SELECT nome FROM Aluno UNION SELECT nome FROM Prof;**
- Quais os nomes dos alunos que fizeram alguma prova de 'ts1'?  
**SELECT distinct nome FROM Aluno, Prova  
WHERE Aluno.nr=Prova.nr AND Prova.cod = 'TS1';**
- Quais os nomes dos alunos com inscrição no curso 'IS'?  
**SELECT distinct nome FROM Aluno, Cadeira, Prova  
WHERE Aluno.nr=Prova.nr AND Prova.cod=Cadeira.cod  
AND Cadeira.curso = 'IS';**
- Qual a relação dos nomes dos alunos que concluíram o curso 'IS'.

```
SELECT DISTINCT nome FROM Aluno
WHERE nr NOT IN
  (SELECT nr AS alunonr
   FROM Aluno, Cadeira
   WHERE curso='IS' AND NOT (cod IN
    (SELECT cod
     FROM Prova
     WHERE nota>=10 AND nr=alunonr)))
);
```

9. Qual a nota máxima existente nas provas?  
**SELECT max(nota) FROM Prova;**
10. Qual a nota média nas provas de BD?  
**SELECT avg(nota) FROM Prova WHERE cod='BD';**
11. Qual o número de alunos?  
**SELECT count(\*) FROM Aluno;**
12. Qual o número de cadeiras de cada curso?  
**SELECT count(\*) FROM Cadeira GROUP BY curso;**
13. Qual o número de provas de cada aluno?  
**SELECT count(\*) FROM Prova GROUP BY nr;**
14. Qual a média do número de provas por aluno?  
**SELECT avg(sum) FROM (SELECT count(\*) as sum FROM Prova GROUP BY nr);**
15. Qual o nome e respetiva média atual (cadeiras feitas, em qualquer curso) de cada aluno?  
**SELECT nome, AVG(maxNota)**  
**FROM (SELECT nome, cod, MAX(nota) maxNota**  
**FROM Prova, Aluno**  
**WHERE Prova.nr = Aluno.nr**  
**AND nota >= 10**  
**GROUP BY nome, cod)**  
**GROUP BY nome;**
16. Qual a nota máxima de cada cadeira e qual o aluno que a obteve?  
**SELECT A.cod, nome, maxNota**  
**FROM (SELECT cod, MAX(nota) maxNota**  
**FROM Prova**  
**GROUP BY cod) A, Prova, Aluno**  
**WHERE A.cod = Prova.cod**  
**AND nota=maxNota**  
**AND Prova.nr=Aluno.nr;**
17. Obtenha a relação ordenada por curso dos nomes dos alunos formados.

```
SELECT DISTINCT nome, curso
FROM Aluno, Prova, Cadeira C
WHERE Aluno.nr = Prova.nr AND Prova.cod = C.cod AND Aluno.nr NOT
IN
  (SELECT nr AS alunonr
   FROM Aluno, Cadeira
   WHERE Cadeira.curso = C.curso AND NOT (cod IN
     (SELECT cod
      FROM Prova
      WHERE nota >= 10 AND nr=alunonr)))
ORDER BY curso, nome;
```

[Baseado num exercício de Gabriel David]

## 2. OFICINA

Considere a seguinte base de dados.

Marca (idMarca, nome)

Modelo (idModelo, nome, idMarca -> Marca)

CodPostal (codPostal1, localidade)

Cliente (idCliente, nome, morada, codPostal1 -> CodPostal, codPostal2, telefone)

Carro (idCarro, matricula, idModelo -> Modelo, idCliente -> Cliente)

Reparacao (idReparacao, dataInicio, dataFim, idCliente -> Cliente, idCarro -> Carro)

Peca (idPeca, codigo, designacao, custoUnitario, quantidade)

ReparacaoPeca (idReparacao -> Reparacao, idPeca -> Peca, quantidade)

PecaModelo (idPeca -> Peca, idModelo -> Modelo)

Especialidade (idEspecialidade, nome, custoHorario)

Funcionario (idFuncionario, nome, morada, codPostal1 -> CodPostal, codPostal2, telefone, idEspecialidade -> Especialidade);

FuncionarioReparacao (idFuncionario -> Funcionario, idReparacao -> Reparacao, numHoras)

Responda em SQL às perguntas indicadas abaixo e obtenha a resposta no sistema de gestão de base de dados.

- a. Quais as peças com custo unitário inferior a 10€ e cujo código contém '98'?

```
SELECT * FROM Peca WHERE custoUnitario<10 AND codigo LIKE  
'%98%';
```

- b. Quais as matrículas dos carros que foram reparados no mês de Setembro de 2010, i.e., cuja reparação terminou nesse mês?

```
SELECT matricula  
FROM Carro, Reparacao  
WHERE Carro.idCarro=Reparacao.idCarro AND strftime('%m',  
dataFim) = '09' AND strftime('%Y', dataFim) = '2010';
```

- c. Quais os nomes dos clientes proprietários de carros que utilizaram peças com custo unitário superior a 10€? Apresente o resultado ordenado por ordem descendente do custo unitário.

```
SELECT nome FROM Cliente, Carro, Reparacao, ReparacaoPeca, Peca  
WHERE Cliente.idCliente=Carro.idCliente  
AND Carro.idCarro=Reparacao.idCarro  
AND Reparacao.idReparacao=ReparacaoPeca.idReparacao  
AND ReparacaoPeca.idPeca=Peca.idPeca  
AND custoUnitario>10  
ORDER BY custoUnitario DESC;
```

- d. Quais os nomes dos clientes que não têm (tanto quanto se saiba) carro?

```
SELECT nome FROM Cliente  
WHERE idCliente NOT IN  
(SELECT idCliente From Carro);
```

- e. Qual o número de reparações feitas a cada carro?

```
SELECT matricula, COUNT(*) "Num Reparacoes" FROM Carro,  
Reparacao  
WHERE Carro.idCarro = Reparacao.idCarro  
GROUP BY matricula;
```

- f. Qual o número de dias em que cada carro esteve em reparação?

```
SELECT matricula, SUM(dataFim-dataInicio) "No de dias"  
FROM Carro, Reparacao  
WHERE Carro.idCarro=reparacao.idCarro  
GROUP BY matricula;
```

- g. Qual o custo unitário médio, o valor total e o número de unidades das peças, bem como o valor da peça mais cara e da mais barata?

```
SELECT AVG(custoUnitario) "Média", SUM(custoUnitario*quantidade)  
"Val total", COUNT(*) "No de peças", MIN(custoUnitario) "preço  
menor", MAX(custoUnitario) "preço maior"  
FROM Peca;
```

- h. Qual a especialidade que foi utilizada mais vezes nas reparações dos carros de cada marca?

**Com utilização de vistas**

```
CREATE VIEW EspecialidadesMarca AS  
SELECT Marca.nome AS nomeMarca, Especialidade.nome AS  
nomeEspec, COUNT(*) AS numEspecMarca  
FROM Especialidade, Funcionario, FuncionarioReparacao, Reparacao,  
Carro, Modelo, Marca
```

```

WHERE Especialidade.idEspecialidade=Funcionario.idEspecialidade
AND Funcionario.idFuncionario=FuncionarioReparacao.idFuncionario
AND FuncionarioReparacao.idReparacao=Reparacao.idReparacao AND
Reparacao.idCarro=Carro.idCarro AND
Carro.idModelo=Modelo.idModelo AND
Modelo.idMarca=Marca.idMarca
GROUP BY Marca.nome, Especialidade.nome;

```

```

SELECT nomeMarca AS nomeMarca1, nomeEspec
FROM EspecialidadesMarca
WHERE numEspecMarca IN (
    SELECT MAX(numEspecMarca)
    FROM EspecialidadesMarca
    GROUP BY nomeMarca
    HAVING nomeMarca=nomeMarca1);

```

- i. Qual o preço total de cada reparação?

Com utilização de vistas

```

CREATE VIEW PrecoReparacao1 AS
SELECT FuncionarioReparacao.idReparacao AS idReparacao,
ifnull(SUM(Especialidade.custoHorario*FuncionarioReparacao.numHo
ras),0) AS precoFuncionario
FROM Especialidade, Funcionario, FuncionarioReparacao
WHERE Especialidade.idEspecialidade=Funcionario.idEspecialidade
AND Funcionario.idFuncionario=FuncionarioReparacao.idFuncionario
GROUP BY FuncionarioReparacao.idReparacao;

```

```

CREATE VIEW PrecoReparacao2 AS
SELECT ReparacaoPeca.idReparacao AS idReparacao,
ifnull(SUM(Peca.custoUnitario*ReparacaoPeca.quantidade),0) AS
precoPeca
FROM ReparacaoPeca, Peca
WHERE ReparacaoPeca.idPeca=Peca.idpeca
GROUP BY ReparacaoPeca.idReparacao;

```

```

CREATE VIEW precoReparacao AS
SELECT ifnull(idReparacao1,idReparacao2) AS idReparacao,
ifnull(precoFuncionario,0) + ifnull(precoPeca,0) AS preco
FROM
(SELECT PrecoReparacao1.idReparacao AS idReparacao1,
PrecoReparacao1.precoFuncionario, PrecoReparacao2.idReparacao AS
idReparacao2, PrecoReparacao2.precoPeca
FROM PrecoReparacao1
LEFT JOIN PrecoReparacao2
ON PrecoReparacao1.idReparacao = PrecoReparacao2.idReparacao
UNION ALL
SELECT PrecoReparacao1.idReparacao AS idReparacao1,
PrecoReparacao1.precoFuncionario, PrecoReparacao2.idReparacao AS
idReparacao2, PrecoReparacao2.precoPeca
FROM PrecoReparacao2
LEFT JOIN PrecoReparacao1
ON PrecoReparacao1.idReparacao = PrecoReparacao2.idReparacao
WHERE PrecoReparacao1.idReparacao IS NULL);

```

```
SELECT * FROM precoReparacao;
```

- j. Qual o preço total das reparações com custo total superior a 60€?

Sem a vista PrecoReparacao

```
CREATE VIEW PrecoReparacao1 AS
SELECT FuncionarioReparacao.idReparacao AS idReparacao,
ifnull(SUM(Especialidade.custoHorario*FuncionarioReparacao.numHo
ras),0) AS precoFuncionario
FROM Especialidade, Funcionario, FuncionarioReparacao
WHERE Especialidade.idEspecialidade=Funcionario.idEspecialidade
AND Funcionario.idFuncionario=FuncionarioReparacao.idFuncionario
GROUP BY FuncionarioReparacao.idReparacao;
```

```
CREATE VIEW PrecoReparacao2 AS
SELECT ReparacaoPeca.idReparacao AS idReparacao,
ifnull(SUM(Peca.custoUnitario*ReparacaoPeca.quantidade),0) AS
precoPeca
FROM ReparacaoPeca, Peca
WHERE ReparacaoPeca.idPeca=Peca.idpeca
GROUP BY ReparacaoPeca.idReparacao;
```

```
CREATE VIEW precoReparacao AS
SELECT ifnull(idReparacao1,idReparacao2) AS idReparacao,
ifnull(precoFuncionario,0) + ifnull(precoPeca,0) AS preco
FROM
(SELECT PrecoReparacao1.idReparacao AS idReparacao1,
PrecoReparacao1.precoFuncionario, PrecoReparacao2.idReparacao AS
idReparacao2, PrecoReparacao2.precoPeca
FROM PrecoReparacao1
LEFT JOIN PrecoReparacao2
ON PrecoReparacao1.idReparacao = PrecoReparacao2.idReparacao
UNION ALL
SELECT PrecoReparacao1.idReparacao AS idReparacao1,
PrecoReparacao1.precoFuncionario, PrecoReparacao2.idReparacao AS
idReparacao2, PrecoReparacao2.precoPeca
FROM PrecoReparacao2
LEFT JOIN PrecoReparacao1
ON PrecoReparacao1.idReparacao = PrecoReparacao2.idReparacao
WHERE PrecoReparacao1.idReparacao IS NULL)
WHERE preco>60;
```

Com a vista PrecoReparacao

```
SELECT * FROM PrecoReparacao WHERE preco>60;
```

- k. Qual o proprietário do carro que teve a reparação mais cara?

```
SELECT nome
FROM Cliente, Carro, Reparacao, PrecoReparacao
WHERE Cliente.idCliente=Carro.idCliente
AND Carro.idCarro=Reparacao.idCarro
AND Reparacao.idReparacao=PrecoReparacao.idReparacao
AND PrecoReparacao.preco = (SELECT MAX(preco) FROM
PrecoReparacao);
```

- l. Qual a matrícula do carro com a segunda reparação mais cara?  

```
SELECT matricula
FROM Carro, Reparacao, PrecoReparacao
WHERE Carro.idCarro=Reparacao.idCarro
AND Reparacao.idReparacao=PrecoReparacao.idReparacao
AND PrecoReparacao.preco =
(SELECT MAX(preco) FROM PrecoReparacao
WHERE preco NOT IN
(SELECT MAX(preco) FROM PrecoReparacao));
```
- m. Quais são as três reparações mais caras (ordenadas por ordem decrescente de preço)?  

```
SELECT * FROM PrecoReparacao ORDER BY preco DESC LIMIT 3;
```
- n. Quais os nomes dos clientes responsáveis por reparações de carros e respetivos proprietários (só para os casos em que não são coincidentes)?  

```
SELECT C1.nome "Proprietário", C2.nome "Cliente"
FROM Cliente C1, Cliente C2, Carro, Reparacao
WHERE Reparacao.idCarro=Carro.idCarro
AND Carro.idCliente=C1.idCliente
AND Reparacao.idCliente=C2.idCliente
AND C1.idCliente<>C2.idCliente;
```
- o. Quais as localidades onde mora alguém, seja ele cliente ou funcionário?  

```
SELECT localidade FROM CodPostal, Cliente
WHERE CodPostal.codPostal1=Cliente.codPostal1
UNION
SELECT localidade FROM CodPostal, Funcionario
WHERE CodPostal.codPostal1=Funcionario.codPostal1;
```
- p. Quais as localidades onde moram clientes e funcionários?  

```
SELECT localidade FROM CodPostal, Cliente
WHERE CodPostal.codPostal1=Cliente.codPostal1
INTERSECT
SELECT localidade FROM CodPostal, Funcionario
WHERE CodPostal.codPostal1=Funcionario.codPostal1;
```
- q. Quais as peças compatíveis com modelos da Volvo cujo preço é maior do que o de qualquer peça compatível com modelos da Renault?  

Sem utilização de vistas

```
SELECT codigo
FROM Peca, PecaModelo, Modelo, Marca
WHERE Peca.idPeca=PecaModelo.idPeca AND
PecaModelo.idModelo=Modelo.idModelo AND
Modelo.idMarca=Marca.idMarca
AND Marca.nome='Volvo' AND Peca.custoUnitario > (SELECT
MAX(Peca.custoUnitario) FROM Peca, PecaModelo, Modelo, Marca
WHERE Peca.idPeca=PecaModelo.idPeca AND
PecaModelo.idModelo=Modelo.idModelo AND
Modelo.idMarca=Marca.idMarca AND Marca.nome='Renault');
```

Com utilização de vistas

```
CREATE VIEW PecasMarca AS
```



```
SELECT Peca.idPeca as idPeca, Marca.nome as nomeMarca
FROM Peca, PecaModelo, Modelo, Marca
WHERE Peca.idPeca=PecaModelo.idPeca AND
PecaModelo.idModelo=Modelo.idModelo AND
Modelo.idMarca=Marca.idMarca;
```

```
SELECT código
FROM Peca, PecasMarca
WHERE Peca.idPeca=PecasMarca.idPeca AND
PecasMarca.nomeMarca='Volvo' AND Peca.custoUnitario > (SELECT
MAX(custoUnitario) FROM Peca, PecasMarca WHERE
Peca.idPeca=PecasMarca.idPeca AND
PecasMarca.nomeMarca='Renault');
```

- r. Quais as peças compatíveis com modelos da Volvo cujo preço é maior do que o de alguma peça compatível com modelos da Renault?

```
SELECT código
FROM Peca, PecasMarca
WHERE Peca.idPeca=PecasMarca.idPeca AND
PecasMarca.nomeMarca='Volvo' AND Peca.custoUnitario > (SELECT
MIN(custoUnitario) FROM Peca, PecasMarca WHERE
Peca.idPeca=PecasMarca.idPeca AND
PecasMarca.nomeMarca='Renault');
```

- s. Quais as matriculas dos carros que foram reparados mais do que uma vez?

```
SELECT matricula FROM Carro
WHERE idCarro IN
(SELECT idCarro FROM Reparacao
GROUP BY idCarro
HAVING COUNT(*)>1);
```

- t. Quais as datas de início e de fim e nome do proprietário das reparações feitas por carros que foram reparados mais do que uma vez?

```
SELECT dataInicio, dataFim, Cliente.nome
FROM Reparacao, Cliente, Carro
WHERE Reparacao.idCarro=Carro.idCarro AND
Carro.idCliente=Cliente.idCliente AND Carro.idCarro IN
(SELECT idCarro FROM Reparacao
GROUP BY idCarro
HAVING COUNT(*)>1);
```

- u. Quais as reparações que envolveram todas as especialidades?

```
SELECT idReparacao
FROM Reparacao
WHERE idReparacao NOT IN
  (SELECT idReparacao AS idReparacao1 FROM Reparacao,
  Especialidade
  WHERE idEspecialidade NOT IN
    (SELECT idEspecialidade
    FROM FuncionarioReparacao,Funcionario
    WHERE
    FuncionarioReparacao.idFuncionario=Funcionario.idFuncionario AND
    FuncionarioReparacao.idReparacao=idReparacao1));
```

- v. Calcule as durações de cada reparação, contabilizando até à data atual os não entregues.

```
SELECT idReparacao, ifnull(dataFim,date('now'))-dataInicio "Duração"  
FROM Reparacao;
```

- w. Substitua Renault por Top, Volvo por Down e os restantes por NoWay.

```
SELECT CASE WHEN Marca.nome='Renault' THEN 'Top'  
          WHEN Marca.nome='Volvo' THEN 'Down'  
          ELSE 'NoWay' END AS nomemarca  
FROM Carro, Modelo, Marca  
WHERE Carro.idModelo=Modelo.idmodelo AND  
Modelo.idMarca=Marca.idMarca;
```