

Questão 1/10 - Banco de Dados ➔

Considerando a tabela a seguir, denominada de employees:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	24000.00	0.00	0	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1987-06-18	AD_VP	17000.00	0.00	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	1987-06-19	AD_VP	17000.00	0.00	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	1987-06-20	IT_PROG	9000.00	0.00	102	60
104	Bruce	Ernst	BERNST	590.423.4568	1987-06-21	IT_PROG	6000.00	0.00	103	60
105	David	Austin	DAUSTIN	590.423.4569	1987-06-22	IT_PROG	4800.00	0.00	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	1987-06-23	IT_PROG	4800.00	0.00	103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	1987-06-24	IT_PROG	4200.00	0.00	103	60

Escreva um *query SQL*, que escreva a expressão *CANCELADO*, atualizando todos os campos EMAIL de todos os funcionários do departamento 60.

Para completar esta tarefa você deverá escrever o *query SQL* desejado.

OBS: Por gentileza, anexar a resposta da questão caso não consiga prosseguir ou o sistema apresente uma mensagem "sem conexão com internet". Anexar documento .doc ou .pdf .

RESOLUÇÃO:

```
UPDATE employees SET email = 'CANCELADO' WHERE department_id = 60;
```

← Questão 2/10 - Banco de Dados →

Considerando a tabela a seguir, denominada de employees:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	24000.00	0.00	0	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1987-06-18	AD_VP	17000.00	0.00	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	1987-06-19	AD_VP	17000.00	0.00	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	1987-06-20	IT_PROG	9000.00	0.00	102	60
104	Bruce	Ernst	BERNST	590.423.4568	1987-06-21	IT_PROG	6000.00	0.00	103	60
105	David	Austin	DAUSTIN	590.423.4569	1987-06-22	IT_PROG	4800.00	0.00	103	60

Escreva um *query SQL* que devolva o total de salário, campo SALARY, por departamento, campo DEPARTMENT_ID.

Para completar esta tarefa você deverá escrever o *query SQL* especificado.

RESOLUÇÃO:

```
SELECT SUM(salary) AS total_salario, department_id FROM employees GROUP BY department_id;
```

← Questão 3/10 - Banco de Dados →

Considerando a tabela a seguir, denominada de employees, escreva um *query* SQL que calcule a soma dos salários, campo SALARY, que deverão ser pagos aos funcionários.

Employee_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT
100	Steven	teste	teste@teste.com	99999-9999	25/11/2014	teste	24000	teste	teste	90
101	Neena	teste	teste@teste.com	99999-9999	26/11/2014	teste	3500	teste	teste	90
102	Lex	teste	teste@teste.com	99999-9999	27/11/2014	teste	1600	teste	teste	90
103	Alexander	teste	teste@teste.com	99999-9999	28/11/2014	teste	1000	teste	teste	90
104	Bruce	teste	teste@teste.com	99999-9999	29/11/2014	teste	1000	teste	teste	60
105	David	teste	teste@teste.com	99999-9999	30/11/2014	teste	5700	teste	teste	60
106	Valli	teste	teste@teste.com	99999-9999	01/12/2014	teste	2000	teste	teste	60
107	Diana	teste	teste@teste.com	99999-9999	02/12/2014	teste	4400	teste	teste	60

RESOLUÇÃO:

```
SELECT SUM(salary) AS soma_salarios FROM employees;
```

← Questão 4/10 - Banco de Dados →

Considere o seguinte script de criação de banco de dados. Observe que este script cria duas tabelas e insere dados em ambas:

```
CREATE TABLE Fabricantes (  
    Codigo INTEGER PRIMARY KEY NOT NULL,  
    Nome TEXT NOT NULL  
);  
  
CREATE TABLE Produtos (  
    Codigo INTEGER PRIMARY KEY NOT NULL,  
    Nome TEXT NOT NULL,  
    Preco DOUBLE NOT NULL,  
    Fabricante INTEGER NOT NULL  
);  
  
INSERT INTO Fabricantes(Codigo,Nome) VALUES(1,'Sony');  
INSERT INTO Fabricantes (Codigo,Nome) VALUES(2,'Creative Labs');  
INSERT INTO Fabricantes (Codigo,Nome) VALUES(3,'Hewlett-Packard');  
INSERT INTO Fabricantes (Codigo,Nome) VALUES(4,'Iomega');  
INSERT INTO Fabricantes (Codigo,Nome) VALUES(5,'Fujitsu');  
INSERT INTO Fabricantes (Codigo,Nome) VALUES(6,'Winchester');  
  
INSERT INTO Produtos(Codigo,Nome,Preco,Fabricante) VALUES(1,'Hard drive',240,5);  
INSERT INTO Produtos(Codigo,Nome,Preco,Fabricante) VALUES (2,'Memory',120,6);  
INSERT INTO Produtos(Codigo,Nome,Preco,Fabricante) VALUES(3,'ZIP drive',150,4);  
INSERT INTO Produtos(Codigo,Nome,Preco,Fabricante) VALUES(4,'Floppy disk',5,6);  
INSERT INTO Produtos(Codigo,Nome,Preco,Fabricante) VALUES(5,'Monitor',240,1);  
INSERT INTO Produtos(Codigo,Nome,Preco,Fabricante) VALUES(6,'DVD drive',180,2);  
INSERT INTO Produtos(Codigo,Nome,Preco,Fabricante) VALUES(7,'CD drive',90,2);  
INSERT INTO Produtos(Codigo,Nome,Preco,Fabricante) VALUES(8,'Printer',270,3);  
INSERT INTO Produtos(Codigo,Nome,Preco,Fabricante) VALUES(9,'Toner',66,3);  
INSERT INTO Produtos(Codigo,Nome,Preco, Fabricante) VALUES(10,'DVD burner',180,2);
```

Escreva um conjunto de *queries SQL* que devolvam as seguintes informações: (a) produtos cujo preço esteja entre R\$60,00 e R\$120,00; (b) calcule o preço médio de todos os produtos cujo código do fabricante seja 2 e (c) imprima uma lista dos preços médios por código de fabricante.

Escreva um conjunto de *queries SQL* que devolvam as seguintes informações: (a) produtos cujo preço esteja entre R\$60,00 e R\$120,00; (b) calcule o preço médio de todos os produtos cujo código do fabricante seja 2 e (c) imprima uma lista dos preços médios por código de fabricante.

RESOLUÇÃO:

- a) `SELECT * FROM Produtos WHERE Preco BETWEEN 61 AND 119;`
- b) `SELECT AVG(Preco) as preco_medio FROM Produtos WHERE Fabricante = 2;`
- c) `SELECT AVG(Preco) as preco_medio, Fabricante FROM Produtos GROUP BY Fabricante;`

← Questão 5/10 - Banco de Dados →

Considerando a tabela a seguir, denominada de employees:

Employee_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT
100	Steven	teste	teste@teste.com	99999-9999	25/11/2014	teste	24000	teste	teste	90
101	Neena	teste	teste@teste.com	99999-9999	26/11/2014	teste	3500	teste	teste	90
102	Lex	teste	teste@teste.com	99999-9999	27/11/2014	teste	1600	teste	teste	90
103	Alexander	teste	teste@teste.com	99999-9999	28/11/2014	teste	1000	teste	teste	90
104	Bruce	teste	teste@teste.com	99999-9999	29/11/2014	teste	1000	teste	teste	60
105	David	teste	teste@teste.com	99999-9999	30/11/2014	teste	5700	teste	teste	60
106	Valli	teste	teste@teste.com	99999-9999	01/12/2014	teste	2000	teste	teste	60
107	Diana	teste	teste@teste.com	99999-9999	02/12/2014	teste	4400	teste	teste	60

Escreva um *query SQL*, que apresente o telefone (PHONE_NUMBER) de cada empregado em ordem alfabética invertida (de Z até A) pelo primeiro nome (FIRST_NAME).

RESOLUÇÃO:

```
SELECT phone_number FROM employees ORDER BY first_name DESC;
```

ou

```
SELECT phone_number, first_name FROM employees ORDER BY first_name DESC;
```

← Questão 6/10 - Banco de Dados →

Observe a tabela *Employees* abaixo:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	24000.00	0.00	0	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1987-06-18	AD_VP	17000.00	0.00	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	1987-06-19	AD_VP	17000.00	0.00	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	1987-06-20	IT_PROG	9000.00	0.00	102	60
104	Bruce	Ernst	BERNST	590.423.4568	1987-06-21	IT_PROG	6000.00	0.00	103	60
105	David	Austin	DAUSTIN	590.423.4569	1987-06-22	IT_PROG	4800.00	0.00	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	1987-06-23	IT_PROG	4800.00	0.00	103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	1987-06-24	IT_PROG	4200.00	0.00	103	60

Escreva uma *query SQL*, que consulte o primeiro nome (*FIRST_NAME*), e o salário (*SALARY*) de todos os funcionários cujo salário seja maior que R\$5000,00 e menor que R\$10000,00. Consulte apenas os funcionários que possuem essa faixa salarial e que trabalham no departamento (*DEPARTMENT_ID*) 60.

RESOLUÇÃO:

```
SELECT first_name, salary FROM employees WHERE salary > 5000.00 AND salary < 10000.00 AND department_id = 60;
```

← Questão 7/10 - Banco de Dados →

Considere a tabela VENDEDORES a seguir. Ela é composta pelas colunas: vendedor_id, nome, cidade e comissao.

Escreva uma query que pesquise todos os registros de vendedores cadastrados nesta tabela, exceto a coluna comissao.

Vendedor_id	nome	cidade	comissao
-----	-----	-----	-----
5001	Paulo de Tarso	New York	0.15
5002	Abdas de Susa	Paris	0.13
5005	Tomás de Aquino	Londres	0.11
5006	Brígida da Suécia	Paris	0.14
5003	Cássio de Narni	Santos	0.12
5007	Inês de Assis	Roma	0.13

RESOLUÇÃO:

```
SELECT vendedor_id, nome, cidade FROM vendedores;
```


← Questão 8/10 - Banco de Dados →

Considerando a tabela a seguir, denominada de employees:

Employee_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT
100	Steven	teste	teste@teste.com	99999-9999	25/11/2014	teste	24000	teste	teste	90
101	Neena	teste	teste@teste.com	99999-9999	26/11/2014	teste	3500	teste	teste	90
102	Lex	teste	teste@teste.com	99999-9999	27/11/2014	teste	1600	teste	teste	90
103	Alexander	teste	teste@teste.com	99999-9999	28/11/2014	teste	1000	teste	teste	90
104	Bruce	teste	teste@teste.com	99999-9999	29/11/2014	teste	1000	teste	teste	60
105	David	teste	teste@teste.com	99999-9999	30/11/2014	teste	5700	teste	teste	60
106	Valli	teste	teste@teste.com	99999-9999	01/12/2014	teste	2000	teste	teste	60
107	Diana	teste	teste@teste.com	99999-9999	02/12/2014	teste	4400	teste	teste	60

Escreva um *query SQL*, que escreva a expressão *CANCELADO*, atualizando todos os campos EMAIL de todos os funcionários do departamento 60.

RESOLUÇÃO:

```
UPDATE employees SET email = 'CANCELADO' WHERE department_id = 60;
```

← Questão 9/10 - Banco de Dados →

No quadro a seguir você pode ver a estrutura de uma tabela denominada País. Você deve escrever uma *query SQL* para criar esta tabela.

Field	Type	Null	Key	Default	Extra
PAIS_ID	varchar(2)	YES		NULL	
NOME_PAIS	varchar(40)	YES		NULL	

RESOLUÇÃO:

```
CREATE TABLE País
(
  PAIS_ID varchar(2),
  NOME_PAIS varchar(40)
);
```

← Questão 10/10 - Banco de Dados

Considerando a tabela a seguir, *Employees*, como sendo a representação de uma tabela em um banco de dados mantido em um servidor MySQL, escreva um *query SQL*, que devolva o primeiro nome, *FIRST_NAME*, e o salário, *SALARY*, de todos os funcionários cujo salário seja maior que R\$800,00.

Considerando a tabela a seguir, denominada de *employees*:

Employee_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT
100	Steven	teste	teste@teste.com	99999-9999	25/11/2014	teste	24000	teste	teste	90
101	Neena	teste	teste@teste.com	99999-9999	26/11/2014	teste	3500	teste	teste	90
102	Lex	teste	teste@teste.com	99999-9999	27/11/2014	teste	1600	teste	teste	90
103	Alexander	teste	teste@teste.com	99999-9999	28/11/2014	teste	1000	teste	teste	90
104	Bruce	teste	teste@teste.com	99999-9999	29/11/2014	teste	1000	teste	teste	60
105	David	teste	teste@teste.com	99999-9999	30/11/2014	teste	5700	teste	teste	60
106	Valli	teste	teste@teste.com	99999-9999	01/12/2014	teste	2000	teste	teste	60
107	Diana	teste	teste@teste.com	99999-9999	02/12/2014	teste	4400	teste	teste	60

RESOLUÇÃO:

```
SELECT first_name, salary FROM employees WHERE salary > 800.00;
```