

Exercícios de Revisão para Grau B

1. Sobre a utilização da sintaxe SQL, informe se é falso (F) ou verdadeiro (V) o que se afirma abaixo. A seguir, indique a opção com a sequência correta.

() A instrução SELECT permite definir quais colunas de quais tabelas desejamos obter as fontes (tabelas) dos dados e os ordenamentos, entre outros.

() A instrução DELETE não permite definir uma condição, enquanto que o comando SELECT permite.

() A instrução SELECT é utilizada para retornar um conjunto de linhas a partir de uma ou mais tabelas do banco de dados.

() A instrução INSERT permite definir quais colunas terão valores a serem preenchidos, mas não permite inserir um conjunto de registros a partir de um comando SELECT.

() A instrução UPDATE permite definir uma condição nos mesmos moldes do comando SELECT, envolvendo, inclusive, mais de uma tabela.

- a) V - V - F - F - F
- b) V - F - V - F - V
- c) F - F - F - V - V
- d) F - V - V - V - F
- e) V - F - V - V - F

-
2. Considere o comando UPDATE da linguagem SQL, para alterar dados em um banco de dados. Considere que a tabela e colunas são denominadas da seguinte forma:

Nome da tabela: produtos;	Valor do custo: custo;
Valor de venda: venda;	Código do fornecedor: fornecedor.

Utilizando a seguinte linha de código:

```
UPDATE produtos SET custo = custo * 1.2, venda = venda * 1.25 WHERE fornecedor = 10;
```

Podemos então afirmar que

- A. a tabela produtos terá todos os valores de custo reajustados em 1.2; e os de venda em 1.25.
 - B. a sintaxe do comando UPDATE está incorreta.
 - C. o comando UPDATE, da forma como está, permite alterar os valores das duas colunas (custo e venda) ao mesmo tempo.
 - D. o fornecedor cujo código é "10" será excluído da tabela.
 - E. o fornecedor cujo código é "10" terá os valores de custo e venda dos seus produtos incrementados em 25% e 20% respectivamente.
-

3. As consultas em SQL podem ser muito complexas. A forma básica do comando select, às vezes denominada mapeamento ou bloco select-from-where, é composta pelas três cláusulas SELECT, FROM e WHERE, e tem a seguinte forma: SELECT FROM WHERE ; sendo que o WHERE pode ser condicional.

Analise a situação a seguir.

“Para cada projeto localizado em ‘Mauá’, liste o número do projeto, o número do departamento que o controla e o sobrenome, endereço e data de nascimento do gerente do departamento.”

Assinale a alternativa que exprime de forma correta essa solicitação

- a) SELECT Projnumero, Dnum, Unome, Endereco, Datanasc
FROM PROJETO, DEPARTAMENTO, FUNCIONARIO
WHERE Dnum=Dnumero AND
Cpf_gerente=Cpf AND
Projlocal='Mauá';
 - b) SELECT Projnumero, Dnum, Unome, Endereco, Datanasc;
FROM PROJETO, DEPARTAMENTO, FUNCIONARIO;
WHERE Dnum=Dnumero AND;
Cpf_gerente=Cpf AND;
Projlocal='Mauá';
 - c) SELECT Projnumero, Dnum, Unome, Endereco, Datanasc
FROM PROJETO, DEPARTAMENTO, FUNCIONARIO
WHERE Dnum=Dnumero AND;
Cpf_gerente=Cpf AND;
Projlocal='Mauá';
 - d) SELECT Projnumero, Dnum, Unome, Endereco, Datanasc
FROM PROJETO, DEPARTAMENTO, FUNCIONARIO
WHERE Dnum=Dnumero,
Cpf_gerente=Cpf;
Projlocal='Mauá';
 - e) SELECT Projnumero, Dnum, Unome, Endereco, Datanasc
FROM PROJETO, DEPARTAMENTO, FUNCIONARIO
WHERE Dnum=Dnumero;
Cpf_gerente=Cpf AND
Projlocal='Mauá';
-

4. Considere que o *script* SQL abaixo tenha sido executado com sucesso.

```
create view xxx as
select c.id_concurso, concurso,
       cg.id_cargo, cargo,
       sum(pontos) y
from concurso c
       inner join inscrição i
       on c.id_concurso=i.id_concurso
       inner join cargo cg
       on cg.id_cargo=i.id_cargo
group by c.id_concurso,
         concurso,
         cg.id_cargo,
         cargo
```

O comando

```
select
  concurso,
  cargo,
  y/
  (select count(*) from inscrição i
   where i.id_concurso=xxx.id_concurso
   and i.id_cargo=xxx.id_cargo) z
from xxx
```

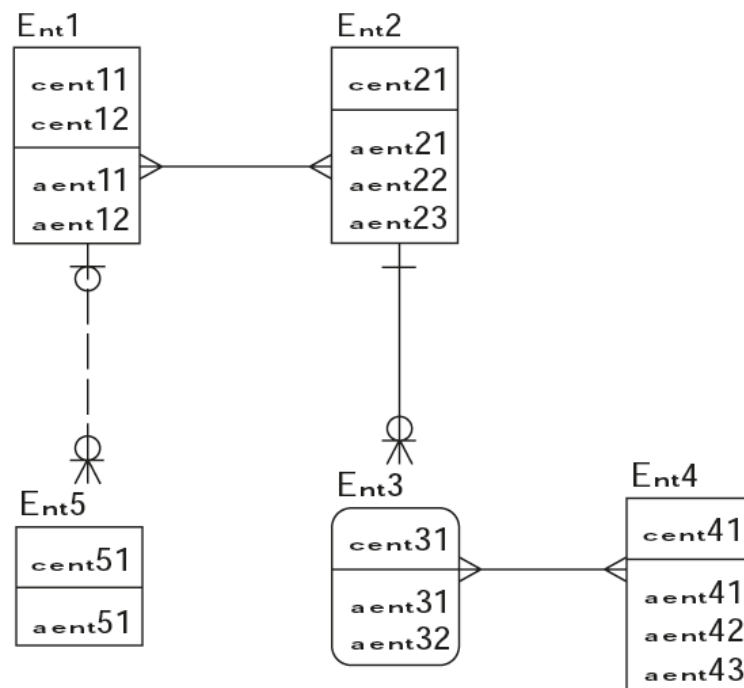
produz:

- a) a média de pontos para cada concurso;
 - b) o número de inscritos para cada cargo de cada concurso;
 - c) a média de pontos para cada cargo de cada concurso;
 - d) o número de pontos para cada cargo de cada concurso;
 - e) o número de cargos e o número de inscritos em cada concurso.
-

5. Considere o diagrama de entidades e relacionamentos a seguir, onde as chaves primárias de cada entidade se encontram na parte superior dos retângulos. As entidades fortes são representadas por retângulos e as entidades fracas são representadas por retângulos com cantos arredondados.

O diagrama atende as seguintes restrições:

- entre Ent1 e Ent2, tem-se um relacionamento muitos para muitos;
- entre as Entidades Ent2 e Ent3, tem-se um relacionamento de um para nenhum, um ou muitos;
- entre Ent1 e Ent5, tem-se um relacionamento de zero ou um para zero, um ou muitos; e
- entre Ent3 e Ent4, tem-se um relacionamento de muitos para muitos.



Aplicando a terceira forma normal ao modelo, qual será o total de colunas que deve ser criado para representar as chaves estrangeiras?

- 3.
- 5.
- 7.
- 8.
- 9.

6. Em um modelo de dados que descreve a publicação acadêmica de pesquisadores de diferentes instituições em eventos acadêmicos, considere as tabelas abaixo.

DEPARTAMENTO (#CodDepartamento, NomeDepartamento)

EMPREGADO (#CodEmpregado, NomeEmpregado, CodDepartamento, Salario)

Na linguagem SQL, o comando mais simples para recuperar os códigos dos departamentos cuja média salarial seja maior que 2000 é:

- a)

```
SELECT CodDepartamento
FROM EMPREGADO
GROUP BY CodDepartamento
HAVING AVG(Salario) > 2000
```
- b)

```
SELECT CodDepartamento
FROM EMPREGADO
WHERE AVG(Salario) > 2000
GROUP BY CodDepartamento
```
- c)

```
SELECT CodDepartamento
FROM EMPREGADO
WHERE AVG(Salario) > 2000
```
- d)

```
SELECT CodDepartamento, AVG(Salario) > 2000
FROM EMPREGADO
GROUP BY CodDepartamento
```
- e)

```
SELECT CodDepartamento
FROM EMPREGADO
GROUP BY CodDepartamento
ORDER BY AVG(Salario) > 2000
```

-
7. Um Analista de TI da Copergás deseja excluir de um banco de dados aberto e em condições ideais todos os produtos da tabela produto que possuem no campo nome valores terminados em 'ão'. Para isso, deverá utilizar a instrução SQL

- a)

```
DELETE * FROM produto WHERE nome LIKE '_ão';
```
- b)

```
DELETE FROM produto WHERE nome CONTAINS '%ão';
```
- c)

```
DELETE FROM produto WHERE nome LIKE '%ão';
```
- d)

```
DELETE * FROM produto WHERE nome LIKE '*ão';
```
- e)

```
DELETE FROM produto WHERE nome = '%ão';
```

* Adaptado de FCC 2016 Copergás PE Analista TI

8. Considere que um BD relacional foi criado através dos comandos SQL a seguir:

```
CREATE TABLE Empregado (  
    matricula char(6) primary key,  
    nome varchar(25),  
    deptoId int  
);  
  
CREATE TABLE Departamento (  
    id int UNIQUE,  
    nome varchar(25)  
);  
  
ALTER TABLE Empregado ADD CONSTRAINT fk_empregado_depto  
FOREIGN KEY (deptoId)  
REFERENCES Departamento(id);  
  
INSERT INTO Departamento VALUES (1, 'Vendas');  
INSERT INTO Departamento VALUES (2, 'Marketing');  
INSERT INTO Departamento VALUES (3, 'Financeiro');  
INSERT INTO Empregado VALUES ('100000', 'Julio', 1);  
INSERT INTO Empregado VALUES ('100001', 'Fernanda', 3);  
INSERT INTO Empregado VALUES ('100002', 'Adriano', 3);  
INSERT INTO Empregado VALUES ('100003', 'Paulo', 2);  
INSERT INTO Empregado VALUES ('100004', 'Marcos', 3);  
INSERT INTO Empregado VALUES ('100005', 'Luis', null);
```

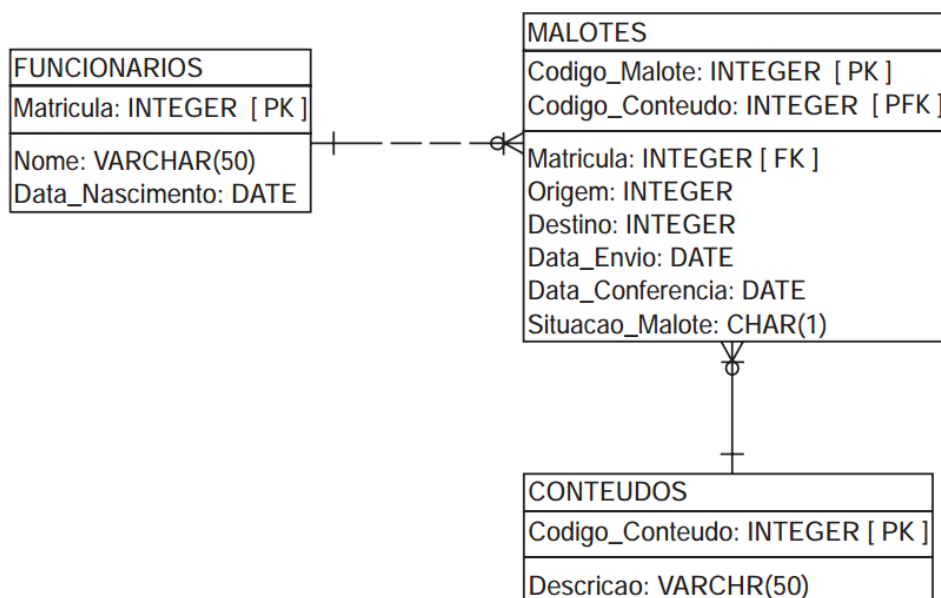
Considere a consulta de seleção a seguir:

```
SELECT * FROM Empregado LEFT OUTER JOIN Departamento ON  
    Empregado.deptoId = Departamento.id;
```

Qual a quantidade de registros retornados?

- A. 3
 - B. 4
 - C. 5
 - D. 6
 - E. 7
-

9. Pedro foi contratado como desenvolvedor de software de uma empresa. Em seu primeiro dia de trabalho ele se deparou com o DER (Diagrama Entidade-Relacionamento), que representa os dados de um sistema de controle de malotes. Foi solicitado a Pedro relatório para o sistema contendo os seguintes dados: o nome de todos os funcionários que enviaram os malotes, o código dos malotes enviados, a descrição de seus conteúdos e a situação dos malotes. Para a geração do relatório, Pedro tem que fazer uma consulta utilizando o comando SELECT da linguagem SQL.



Conhecidos o modelo conceitual de dados e os dados necessários para a tarefa de Pedro, o comando SELECT que ele deve executar para realizar a consulta e produzir o relatório corretamente é

- SELECT NOME, CODIGO_MALOTE, DESCRICAO, SITUACAO_MALOTE FROM MALOTES INNER JOIN CONTEUDOS ON (CODIGO_CONTEUDO = CODIGO_CONTEUDO) INNER JOIN FUNCIONARIOS ON (MATRICULA = MATRICULA);
 - SELECT NOME, CODIGO_MALOTE, DESCRICAO, SITUACAO_MALOTE FROM MALOTES, CONTEUDOS, FUNCIONARIOS WHERE (CODIGO_CONTEUDO = CODIGO_CONTEUDO) AND (MATRICULA = MATRICULA);
 - SELECT NOME, CODIGO_MALOTE, DESCRICAO, SITUACAO_MALOTE FROM MALOTES INNER JOIN CONTEUDOS INNER JOIN FUNCIONARIOS ON (MALOTES.CODIGO_CONTEUDO = CONTEUDOS.CODIGO_CONTEUDO) ON (MALOTES.MATRICULA = FUNCIONARIOS.MATRICULA);
 - SELECT NOME, CODIGO_MALOTE, DESCRICAO, SITUACAO_MALOTE FROM MALOTES INNER JOIN CONTEUDOS ON (MALOTES.CODIGO_CONTEUDO = CONTEUDOS.CODIGO_CONTEUDO) INNER JOIN FUNCIONARIOS ON (MALOTES.MATRICULA = FUNCIONARIOS.MATRICULA);
 - SELECT NOME, CODIGO_MALOTE, DESCRICAO, SITUACAO_MALOTE FROM MALOTES, CONTEUDOS, FUNCIONARIOS INNER JOIN WHERE (MALOTES.CODIGO_CONTEUDO = CONTEUDOS.CODIGO_CONTEUDO) AND (MALOTES.MATRICULA = FUNCIONARIOS.MATRICULA);
-

10. Seja a seguinte tabela de um banco de dados relacional:

Objeto (ID, Nome, Peso, Descrição).

A consulta SQL para obter o Nome e o Peso dos objetos com a restrição do Peso estar entre os valores 2 e 18 é:

- a) `SELECT Nome AND Peso
OF Objeto
HAVING Peso IN (2; 18);`
- b) `SELECT Nome, Peso
FROM Objeto
WHERE Peso IS BETWEEN 2 TO 18;`
- c) `SELECT Objeto (Nome, Peso)
HAVING Peso INSIDE (2; 18);`
- d) `SELECT Objeto (Nome, Peso)
FOR Peso INSIDE (2;18);`
- e) `SELECT Nome, Peso
FROM Objeto
WHERE Peso BETWEEN 2 AND 18;`

*Adaptado de FCC 2017 DPE-RS Analista:

11. Baseando-se no comando SQL abaixo, assinale a única alternativa correta, no contexto da Álgebra Relacional.

`SELECT nomeFiliado FROM Filiado WHERE idPartido = 1`

- a) $\sigma \text{ nomeFiliado } (\pi \text{ idPartido} = 1 (Filiado))$
- b) $\sigma \text{ nomeFiliado } (\pi \text{ idPartido} > 1 (Filiado))$
- c) $\pi \text{ nomeFiliado } (\sigma \text{ idPartido} = 1 (Filiado))$
- d) $\pi \text{ nomeFiliado } (\sigma \text{ idPartido} < 1 (Filiado))$
- e) $\pi \text{ idFiliado } (\sigma \text{ nomeFiliado} = 'Joao' (Filiado))$
-

12. Baseando-se no comando SQL abaixo, assinale a única alternativa correta, no contexto da Álgebra Relacional.

DELETE FROM Produto WHERE nome = 'barra';

- a) $\sigma_{nome} (\pi_{produto = 'barra'} (Produto))$
 - b) $Produto \leftarrow Produto - (\sigma_{nome = 'barra'} (Produto))$
 - c) $\pi_{nome} (\pi_{produto = 'barra'} (Produto))$
 - d) $\pi_{nome} (Produto)$
 - e) $Produto \leftarrow Produto - (\pi_{nome = 'barra'} (Produto))$
-

13. Sobre o conceito de banco de dados orientado a objetos, considere:

- I. Os dados são armazenados como objetos que podem ser manipulados por métodos de acesso de SGBDs relacionais ou hierárquicos, apenas.
- II. Os objetos são organizados em uma hierarquia de tipos e subtipos que recebem as características de seus supertipos.
- III. Os objetos não podem conter referências para outros objetos.

É correto o que consta em

- A. I, apenas.
 - B. II, apenas
 - C. III, apenas.
 - D. I e III, apenas.
 - E. I, II e III.
-

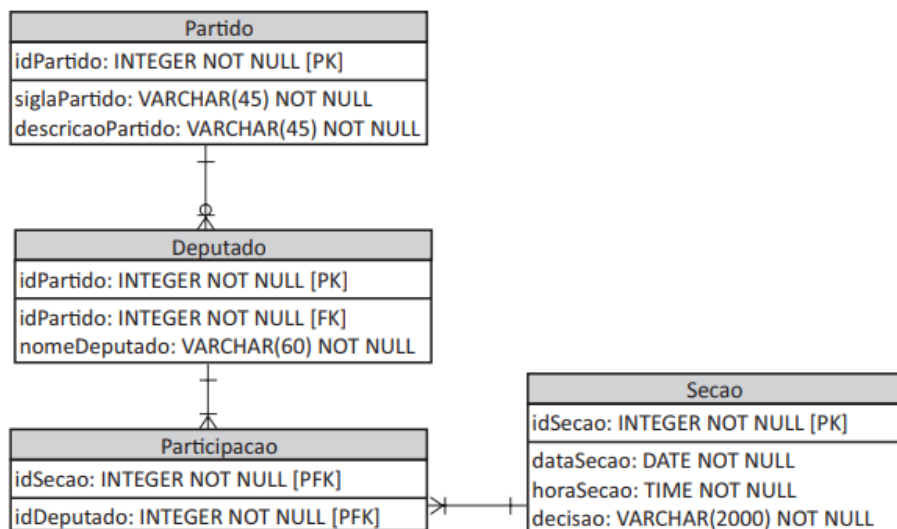
14. Sobre Bancos de Dados NoSQL, analise:

- I. Bancos de dados NoSQL do tipo orientados a documentos são apropriados para o armazenamento de dados semiestruturados.
- II. Apesar de implementarem tecnologias distintas, todos os bancos de dados NoSQL apresentam em comum apenas a implementação da tecnologia chave-valor.
- III. Além do modelo chave-valor, os banco de dados NoSQL também podem ser orientados a documentos ou baseados em grafos.

É correto o que consta em

- A. I, apenas.
 - B. II, apenas
 - C. III, apenas.
 - D. I e III, apenas.
 - E. I, II e III.
-

15. [ENADE-2017] Considere o diagrama Entidade-Relacionamento apresentado a seguir:



Qual código SQL exibe o nome de todos os deputados que compareceram a pelo menos uma seção e as datas de cada seção em que os deputados participaram?

- A) `SELECT Deputado.nomeDeputado, Secao.dataSecao FROM Deputado, Participacao, Secao WHERE Deputado.idDeputado=Participacao.idDeputado;`
- B) `SELECT Deputado.nomeDeputado, Secao.dataSecao FROM Deputado, Participacao, Secao WHERE Deputado.idDeputado = Participacao.idDeputado OR Secao.idSecao = Participacao.idSecao;`
- C) `SELECT Deputado.nomeDeputado, Secao.dataSecao FROM Deputado LEFT OUTER JOIN Participacao ON Deputado.idDeputado = Participacao.idDeputado LEFT OUTER JOIN Secao ON Secao.idSecao = Participacao.idSecao;`
- D) `SELECT Deputado.nomeDeputado, Secao.dataSecao FROM Deputado RIGHT OUTER JOIN Participacao ON Deputado.idDeputado = Participacao.idDeputado RIGHT OUTER JOIN Secao ON Secao.idSecao = Participacao.idSecao;`
- E) `SELECT Deputado.nomeDeputado, Secao.dataSecao FROM Deputado INNER JOIN Participacao ON Deputado.idDeputado = Participacao.idDeputado INNER JOIN Secao ON Participacao.idSecao=Secao.idSecao;`