

## BANCO DE DADOS I - QUESTÕES DE REVISÃO – CONTEÚDO LINGUAGEM SQL

(ENADE 2005 - 73) Considere o seguinte script SQL de criação de um banco de dados.

```
CREATE TABLE PECAS (CODIGO NUMERIC(5) NOT NULL,  
DESCRICAO VARCHAR(20) NOT NULL,  
ESTOQUE NUMERIC(5) NOT NULL,  
PRIMARY KEY(CODIGO));
```

```
CREATE TABLE FORNECEDORES  
(COD_FORN NUMERIC(3) NOT NULL,  
NOME VARCHAR(30) NOT NULL,  
PRIMARY KEY(COD_FORN));
```

```
CREATE TABLE FORNECIMENTOS  
(COD_Peca NUMERIC(5) NOT NULL,  
COD_FORN NUMERIC(3) NOT NULL,  
QUANTIDADE NUMERIC(4) NOT NULL,  
PRIMARY KEY(COD_Peca, COD_FORN),  
FOREIGN KEY (COD_Peca) REFERENCES PECAS,  
FOREIGN KEY (COD_FORN) REFERENCES  
FORNECEDORES);
```

A partir desse script, assinale a opção que apresenta comando SQL que permite obter uma lista que contenha o nome de cada fornecedor que tenha fornecido alguma peça, o código da peça fornecida, a descrição dessa peça e a quantidade fornecida da referida peça.

A) SELECT \* FROM PECAS, FORNECEDORES, FORNECIMENTOS;

B) SELECT \* FROM PECAS, FORNECEDORES,  
FORNECIMENTOS WHERE PECAS.CODIGO = FORNECIMENTOS.COD\_Peca AND  
FORNECEDORES.COD\_FORN = FORNECIMENTOS.COD\_FORN;

C) SELECT NOME, CODIGO, DESCRICAO, QUANTIDADE  
FROM PECAS, FORNECEDORES, FORNECIMENTOS;

D) SELECT NOME, CODIGO, DESCRICAO, QUANTIDADE  
FROM PECAS, FORNECEDORES, FORNECIMENTOS  
WHERE PECAS.CODIGO = FORNECIMENTOS.COD\_Peca  
AND FORNECEDORES.COD\_FORN = FORNECIMENTOS.COD\_FORN;

E) SELECT DISTINCT NOME, CODIGO, DESCRICAO, QUANTIDADE  
FROM PECAS, FORNECEDORES, FORNECIMENTOS  
WHERE CODIGO = COD\_Peca;

(ENADE 2008 – 44) Considere um banco de dados relacional que contém as seguintes tabelas, em que o grifo representa a chave primeira da tabela.

**Produtos (idProduto, descricao, valorUnitario)**

**Estoque (idFilial, idProduto, quantidade)**

A tabela Produtos é populada com aproximadamente 10.000 registros, enquanto a tabela Estoque é populada com aproximadamente 100.000 registros. Para escrever uma consulta em SQL que determine o valor total das mercadorias em estoque de uma filial cujo identificador é igual a 132, pode-se usar uma das duas codificações apresentadas a seguir.

Consulta 1

```
SELECT
SUM(valorUnitario * quantidade)
FROM
Estoque, Produtos
WHERE
Estoque.idFilial = 132 AND
Produtos.idProduto = Estoque.idProduto
```

Consulta 2

```
SELECT
SUM(valorUnitario * quantidade)
FROM
Produtos, Estoque
WHERE
Estoque.idFilial = 132 AND
Produtos.idProduto = Estoque.idProduto
```

A partir dessas informações e considerando que a tabela Estoque possua um índice sobre a coluna idFilial, analise as seguintes asserções.

O processamento da Consulta 1 tem melhor desempenho que o da Consulta 2

**porque**

a quantidade de registros processados na consulta 1 é, no pior caso, igual a 10.000, enquanto o pior caso na consulta 2 terá  $(10.000)^2$  registros.

Assinale a opção correta a respeito dessas asserções.

- A) As duas asserções são proposições verdadeiras, e a segunda é uma justificativa correta da primeira.
- B) As duas asserções são proposições verdadeiras, mas a segunda não é justificativa correta da primeira.
- C) A primeira asserção é uma proposição verdadeira, e a segunda, uma proposição falsa.
- D) A primeira asserção é uma proposição falsa, e a segunda, uma proposição verdadeira.
- E) Tanto a primeira quanto a segunda asserções são proposições falsas.

Considere as seguintes tabelas:

```
CREATE TABLE Departamento
(IdDep int NOT NULL,
NomeDep varchar(15),
CONSTRAINT Departamentopkey PRIMARY KEY (IdDep));
```

```
(IdEmpregado int NOT NULL,
IdDep int,
salario float,
CONSTRAINT Empregadopkey PRIMARY KEY
(IdEmpregado),
CONSTRAINT EmpregadoIdDepfkey FOREIGN KEY (IdDep)
REFERENCES Departamento(IdDep)
ON UPDATE RESTRICT ON DELETE RESTRICT )
```

```
CREATE TABLE Empregado
```

Considere as seguintes consultas SQL.

```
I - SELECT NomeDep, count(*)
FROM Departamento D, Empregado E
WHERE D.IdDep=E.IdDep and E.salario > 10000
GROUP BY NomeDep
HAVING count(*) > 5;
```

```
II - SELECT NomeDep, count(*)
FROM Departamento D, Empregado E
WHERE D.IdDep=E.IdDep and E.salario >10000 and
E.IdDep IN (SELECT IdDep
FROM Empregado
GROUP BY IdDep
HAVING count(*) > 5)
GROUP BY NomeDep;
```

Quando as consultas acima são realizadas, o que é recuperado em cada uma delas?

A) I: os nomes dos departamentos que possuem mais de 5 empregados que ganham mais de 10.000 reais e o número de empregados nessa condição.

II: os nomes dos departamentos que possuem mais de 5 empregados e o número de empregados que ganham mais de 10.000 reais.

B) I: os nomes dos departamentos que possuem mais de 5 empregados e o número de empregados que ganham mais de 10.000 reais.

II: os nomes dos departamentos que possuem mais de 5 empregados que ganham mais de 10.000 reais e o número de empregados nessa condição.

C) I: os nomes dos departamentos que possuem mais de 5 empregados que ganham mais de 10.000 reais e o número total de funcionários do departamento.

II: os nomes dos departamentos que possuem mais de 5 empregados que ganham mais de 10.000 reais e o número de empregados nessa condição.

D) I: os nomes dos departamentos que possuem mais de 5 empregados que ganham mais de 10.000 reais e o número de empregados nessa condição.

II: os nomes dos departamentos que possuem mais de 5 empregados que ganham mais de 10.000 reais e o número total de funcionários do departamento.

E) I: os nomes dos departamentos que possuem mais de 5 empregados que ganham mais de 10.000 reais e o número de empregados nessa condição.

II: os nomes dos departamentos que possuem mais de 5 empregados que ganham mais de 10.000 reais e o número de empregados nessa condição.

(ENADE 2011 – 46) 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**