

Q1 – FGV – TJSC – Analista de Sistemas – 2015

João escreveu os dois comandos SQL abaixo.

I. `select * from T where x>5`

II. `select * from T where not x>5`

Curiosamente, os dois comandos produziram resultados com zero linha. Intrigado, pois sabia que a instância de T não estava vazia, João escreveu outros dois comandos:

III. `select count(*) from T`

IV. `select count(x) from T`

Quando executados, os comandos III e IV, necessariamente, produziram resultados r1 e r2, tal que:

- (A) $r1 = 0$ e $r2 = 0$;
- (B) r1 é um número ≥ 1 e $r2 = 0$;
- (C) r1 é um número ≥ 0 e r2 é NULL;
- (D) r1 é NULL e r2 é NULL;
- (E) $r1 = 0$ e r2 é NULL.

Q2 - FGV – PGE-RO – Micro - 2015

Nos comandos SQL que empregam funções de agregação em conjunto com a cláusula `group by`, é possível filtrar linhas do resultado depois da aplicação das funções de agregação por meio da cláusula:

- (A) `Case`;
- (B) `Distinctrow`;
- (C) `Full outer join`;
- (D) `Having`;
- (E) `Where`.

Q3 – FGV – COMPESA – 2014

Considere as tabelas T1 e T2, com as respectivas instâncias, a seguir.

T1		T2	
a	B	a	c
2	4	2	8
3	2	2	4
1	2	3	1
		3	5
		1	7

Sabe-se que o comando

```
select x.a, count(*) x from T1 x, T2 y
where x.a=y.a
group by x.a
order by . . .
```

produziu

a	x
3	2
2	2
1	1

Nesse caso, é correto afirmar que a cláusula *order by* usada no comando foi

- (A) **order by 2**
- (B) **order by avg(y.c)**
- (C) **order by x.b**
- (D) **order by sum(y.c)**
- (E) **order by count(y.c)**

Q4 – FGV – TCMSP –2015

Considere a tabela relacional criada pelo comando

```
create table xx
(a int null, b int null, c int null)
```

Depois de instanciada com um conjunto de registros, os seguintes comandos foram executados:

```

select count(*) from XX
select count(distinct A) from XX
select count(distinct B) from XX
select count(*) from XX where C>10
select count(*) from XX where not C>10

```

Sabendo-se que esses comandos produziram como resultado, respectivamente, os números 10, 10, 0, 0 e 5, analise as quatro alternativas para a definição da tabela XX:

A lista com todos os comandos que são válidos e compatíveis com a instância corrente da tabela é:

I.

```

CREATE TABLE XX(
    A int NULL,
    B int NULL,
    C int NULL )

```

II.

```

CREATE TABLE XX(
    A int primary key,
    B int NULL,
    C int NULL )

```

III.

```

CREATE TABLE XX(
    A int NULL,
    B int NULL,
    C int )

```

IV.

```

CREATE TABLE XX(
    A int,
    B int primary key,
    C int NULL )

```

- (A) I, II;
- (B) I, II, III;
- (C) II, IV;
- (D) I, III;
- (E) IV.

Q5 – FGV – DPEMT – 2015

Analise o comando SQL a seguir.

```

select x,
       sum (nota) as soma,
       count (nota) as numero
from inscricao
group by x
having ???

```

Assinale a opção que indica a expressão que, ao ser utilizada para substituir o trecho “???”, invalida o comando SQL acima.

- (A) count (nota) > 1
- (B) x=2
- (C) max (nota) = 10
- (D) nota > 7
- (E) (select max(nota) from inscricao) > 5

Q6 – FGV – DPERO – 2015

Analise o comando SQL a seguir.

```

select max(A1) X, count(*) Y, sum(A1) Z
from T

```

Executado quando a instância da tabela T estiver vazia (com zero registros), esse comando produz como resultado:

(A)

X	Y	Z
NULL	0	NULL

(B)

X	Y	Z
NULL	NULL	NULL

(D)

X	Y	Z
0	NULL	0

(C)

X	Y	Z
0	0	0

(E)

X	Y	Z
NULL	NULL	0

Q7 - CFO/QC - 2014/2015

Considere a tabela W abaixo, seus campos VALOR e DADO, e a consulta em linguagem SQL:

W	
VALOR	DADO
1	3
2	4
5	6
4	5

```
SELECT MAX (W.VALOR)
FROM W LEFT JOIN W AS Z
ON (W.VALOR = Z.DADO)
WHERE Z.DADO IS NULL;
```

Teremos como retorno da consulta SQL:

- a) 1
- b) 2
- c) 3
- d) 4
- e) 5

Q8 - FGV – TJ-GO – 2014

Considere um banco de dados relacional que possua as tabelas a seguir, e que em cada tabela os atributos sublinhados formem a chave primária:

PRODUTO (idP, descricaoP)

LOJA (idL, nomeL, enderecoL)

OFERTA (idP, idL, preçoO)

Nesse banco de dados a tabela Oferta contém informações sobre as lojas onde cada produto é vendido e o respectivo preço, e todas as tabelas possuem pelo menos um registro. Sendo assim, o comando SQL

```
select p.descricaoP
```

```
from produto p
```

```
where not exists
```

```
(select *
```

```
from loja l
```

where not exists

(select *

from oferta o

where o.idP=p.idP and o.idL=l.idL))

produz a lista de produtos que:

- a) nenhuma das lojas vende;
- b) todas as lojas vendem;
- c) são vendidos apenas em lojas que não vendem outros produtos;
- d) são vendidos por somente uma loja que vende outros produtos;
- e) não são vendidos por somente uma loja.

Q9 – FGV – TCMSP –2015

Considere as tabelas relacionais, e respectivas instâncias, mostradas a seguir. O comando SQL

```
select A from X1 where
not exists
(select * from X3 where
not exists
(select * from X2 where
X1.A=X2.C and X3.B=X2.D))
```

X1	X2		X3
A	C	D	B
1	1	1	1
2	1	3	3
3	1	5	5
4	2	1	
5	2	3	
	3	1	
	3	1	

produz um resultado com apenas uma coluna, cujo(s) valor(es) é/são:

- (A) 1
- (B) 1,2,3
- (C) 4,5
- (D) 1,2,3,4,5
- (E) NULL

Q10 - CESGRANRIO - BNDES – Desenvolvimento – 2011

As tabelas *Projeto*, *Funcionario* e *Participacao_Projeto* participam da base de dados de um sistema de controle de projetos. A estrutura dessas tabelas é a seguinte:

Projeto (IDProjeto, Nome, DataInicio, DataFim)

Funcionario (Matricula, Nome, DataNascimento)

Participacao_Projeto (IDProjeto, Matricula)

Colunas sublinhadas participam da chave primária
Colunas em negrito participam de chaves estrangeiras

O comando SQL que retorna o nome somente dos funcionários que participaram de **TODOS** os projetos é:

- (A) SELECT DISTINCT F.NOME FROM FUNCIONARIO F
INNER JOIN PARTICIPACAO_PROJETO PP
ON F.MATRICULA = PP.MATRICULA
INNER JOIN PROJETO P
ON P.IDPROJETO = PP.IDPROJETO
- (B) SELECT DISTINCT F.NOME FROM FUNCIONARIO F
INNER JOIN PARTICIPACAO_PROJETO PP
ON F.MATRICULA = PP.MATRICULA
WHERE EXISTS (SELECT 1 FROM PROJETO P
WHERE P.IDPROJETO = PP.IDPROJETO)
- (C) SELECT F.NOME FROM FUNCIONARIO F
WHERE MATRICULA IN
(SELECT MATRICULA FROM PARTICIPACAO_PROJETO)
- (D) SELECT F.NOME FROM FUNCIONARIO F
WHERE NOT EXISTS
(SELECT 1 FROM PROJETO P
WHERE NOT EXISTS (SELECT 1 FROM
PARTICIPACAO_PROJETO PP
WHERE PP.IDPROJETO = P.IDPROJETO
AND PP.MATRICULA = F.MATRICULA))
- (E) SELECT F.NOME FROM FUNCIONARIO F
WHERE NOT EXISTS
(SELECT 1 FROM PARTICIPACAO_PROJETO PP
WHERE PP.MATRICULA = F.MATRICULA
AND NOT EXISTS (SELECT 1 FROM PROJETO P
WHERE PP.IDPROJETO = P.IDPROJETO))

Q11 - CESGRANRIO - BNDES – Desenvolvimento – 2012

Considere as tabelas a seguir para responder às questões de nos 38 e 39. Essas tabelas pertencem ao esquema de um banco de dados de uma locadora de veículos.

```
CREATE TABLE VEICULO (  
  PLACA CHAR(7) NOT NULL,  
  MODELO VARCHAR2(50) NOT NULL,  
  COD_CAT CHAR(2) NOT NULL,  
  CONSTRAINT VEICULO_PK PRIMARY KEY (PLACA),  
  CONSTRAINT VEICULO_FK FOREIGN KEY (COD_CAT) REFERENCES  
    CATEGORIA (COD_CAT))
```

```
CREATE TABLE CATEGORIA (  
  COD_CAT CHAR(2) NOT NULL,  
  DESCR VARCHAR2(80) NOT NULL,  
  VAL_DIARIA NUMBER(7,2),  
  CONSTRAINT CATEGORIA_PK PRIMARY KEY (COD_CAT))
```

```
CREATE TABLE ALUGUEL (  
  PLACA CHAR(7) NOT NULL,  
  DATA_DEV NUMBER(6),  
  DATA_ALG NUMBER(6) NOT NULL,  
  CONSTRAINT ALUGUEL_PK PRIMARY KEY (PLACA, DATA_ALG),  
  CONSTRAINT ALUGUEL_FK FOREIGN KEY (PLACA) REFERENCES  
    VEICULO (PLACA))
```

Observações:

- ▶ A tabela VEICULO contém as informações sobre os veículos que a locadora dispõe para aluguel. Ela possui uma coluna chamada COD_CAT, que contém a categoria à qual um veículo pertence.
- ▶ A tabela CATEGORIA representa a tabela de preços da locadora. Ela registra o valor que será cobrado por um dia de aluguel de um veículo de uma determinada categoria.
- ▶ A tabela ALUGUEL é usada para registrar todas as operações de aluguel.
- ▶ A coluna DATA_ALG guarda a data na qual um veículo foi alugado, enquanto a coluna DATA_DEV guarda a data na qual o veículo foi devolvido. Ela é informada ao sistema quando o cliente devolve o veículo à locadora. Ambas as datas estão no formato AAMMDD. Dessa forma, a data 05/02/2011 será armazenada como 110205.

Qual comando SQL será executado com sucesso, independente do estado das tabelas que compõem a base de dados da locadora de veículos?

- (A) INSERT INTO CATEGORIA (DESCR,VAL_DIARIA,COD_CAT)
VALUES ('sedan compacto',90.00,'uc')
- (B) DELETE FROM CATEGORIA X WHERE
NOT EXISTS (SELECT COUNT(*)
FROM VEICULO V,ALUGUEL A
WHERE V.COD_CAT=X.COD_CAT AND V.PLACA=A.PLACA AND
A.DATA_DEV IS NOT NULL
GROUP BY V.COD_CAT)
- (C) INSERT INTO ALUGUEL VALUES ('LJJ2222',120618)
- (D) DELETE FROM VEICULO X WHERE
NOT EXISTS (SELECT COUNT(*)
FROM ALUGUEL A
WHERE X.PLACA=A.PLACA AND
A.DATA_DEV IS NOT NULL
GROUP BY A.PLACA)
- (E) UPDATE VEICULO X SET COD_CAT='xs' WHERE
EXISTS (SELECT COUNT(*)
FROM VEICULO V,ALUGUEL A
WHERE V.COD_CAT='xs' AND V.PLACA=A.PLACA AND
A.DATA_DEV IS NOT NULL
GROUP BY V.COD_CAT)

Gabarito

Q1 – B

Q2 – D

Q3 – B

Q4 – A

Q5 – D

Q6 – A

Q7 – B

Q8 – B

Q9 – A

Q10 – D

Q11 – E