Q1 - CESGRANRIO - IBGE – Tecnologia da Informação – 2014

SQL é uma linguagem dedicada à operação de Bancos de Dados relacionais, padronizada internacionalmente, e que pode ser encontrada nos principais SGBD modernos.

Os principais comandos da sua linguagem de manipulação de dados (DML) são:

(A) ALTER, CREATE e DROP

(B) CREATE, DELETE, READ e UPDATE

(C) CREATE, DESTROY, FIND e INCLUDE

(D) SELECT, DELETE, INSERT e UPDATE

(E) SELECT, JOIN, PROJECT e RENAME

Q2 - BIO-RIO – RJ-RJ – Desenvolvimento – 2015

"A linguagem SQL é do tipo declarativa e constituída das três sublinguagens a seguir:

1. ____ - inclui os comandos SELECT, INSERT, UPDATE e DELETE;
2. ____ - inclui os comandos CREATE, ALTER e DROP;
3. ____ - inclui os comandos GRANT e REVOKE."

As siglas que completam corretamente as lacunas do fragmento acima são respectivamente:

a) DML, DCL e DDL.

b) DML, DDL e DCL.

c) DCL, DDL e DML.

d) DDL, DML e DCL.

e) DDL, DCL e DML.

Q3 - FCC – MANAUSPREV – An. Previdênciário – 2015

A linguagem SQL é dividida em subconjuntos de acordo com as operações que se deseja efetuar sobre um banco de dados. Considere os grupos de comandos:

I. CREATE, ALTER, DROP.

II. GRANT, REVOKE.

III. DELETE, UPDATE, INSERT.

Os comandos listados em

a) I correspondem à Data Control Language - DCL e II à Data Definition Language - DDL.

b) I correspondem à Data Manipulation Language - DML e III à Data Control Language - DCL.

c) II correspondem à Data Manipulation Language - DML e III à Data Control Language - DCL.

d) I correspondem à Data Definition Language - DDL e III à Data Manipulation Language - DML.

e) II correspondem à Data Control Language - DCL e III à Data Definition Language - DDL.

Q4 – FGV – TCMSP – Analista de Sistemas – 2015

Analise os três comandos a seguir e as afirmativas a respeito de seus efeitos no âmbito do MS SQL Server.

delete from x

truncate table x

drop table x

I. O comando *delete* e o comando *truncate* removem o mesmo conjunto de registros da tabela X.

II. O comando *drop*, quando usado com a opção "with no removal", produz exatamente o mesmo efeito do comando *truncate*.

III. Devido às suas características operacionais, o comando *delete* é usualmente executado muito mais rapidamente que o comando *truncate*.

Está correto o que se afirma em:

(A) somente I;

(B) somente III;

(C) somente I e II;

(D) somente II e III;

(E) I, II e III.

Q5 - FGV – PROCEMPA – 2014

Com referência ao banco BD_CERVEJA, considere que João, analista da empresa, recebeu a tarefa de fazer a engenharia reversa do *script*, e tentou escrever o que, na sua concepção, poderia ser o *script* de criação da tabela Cliente, mostrado a seguir.

```
CREATE TABLE CLIENTE(
 nomeCliente nvarchar(50) NOT NULL,
 nomeFavorita nvarchar(50) NOT NULL,
 Constraint PK_CLIENTE
 PRIMARY KEY (nomeCliente),
 Constraint FK_Cliente_Cerveja
 FOREIGN KEY (nomeFavorita)
   references CERVEJA (nomeCerveja)
   on delete set null
   on update cascade)
```

CLIENTE

nomeCliente	nomeFavorita
Ana	Stella
Mariana	Original
Pedro	Bohemia
Rafael	NULL
Thiago	Stella

CERVEJA

nomeCerveja
Bohemia
Original
Stella

Quando pediu a opinião de uma colega sobre esse *script*, João recebeu os seguintes comentários:

I. Não é possível que haja uma chave estrangeira definida como João imaginou, pois o atributo que constitui a chave estrangeira obrigatoriamente deveria ser denominado **nomeCerveja**, tal qual o atributo da tabela referenciada.

II. Há incompatibilidade entre a semântica do *script* e a instância apresentada para a tabela.

III. Há incompatibilidade entre a semântica do *script* e a declaração dos atributos da tabela.

(A) II, apenas.

(B) II e III, apenas.

(C) I e III, apenas.

(D) I e II, apenas.

(E) I, II e III.

Q6 - BNDES – Desenvolvimento – 2007

Um analista de sistemas elabora um texto explicando um sistema de uma imobiliária. Todo departamento deve possuir um e somente um gerente. Todo empregado deve estar alocado a um e somente um departamento.

O Administrador de Dados elabora os comandos SQL para esse sistema.

CREATE TABLE empregado (

matrícula number(5) NOT NULL,

nome char(200) NOT NULL,

endereço varchar(300) NULL,

iddepto number(3) NOT NULL,

PRIMARY KEY (matricula),

FOREIGN KEY (iddepto) REFERENCES departamento [.....]

)

CREATE TABLE departamento (

iddepto number(3) NOT NULL,

nome char(200) NOT NULL,

matGerente number(5) NOT NULL,

PRIMARY KEY (iddepto),

FOREIGN KEY (matGerente) REFERENCES empregado [.....]

)

Sobre as colunas EMPREGADO.IDDEPTO e DEPARTAMENTO.MATGERENTE e suas restrições de nulidade (NULL ou NOT NULL) e de integridade referencial (chave estrangeira), é correto afirmar que

(A) não é possível ter ambas cadastradas com NOT NULL, pois ao cadastrar o primeiro departamento, um empregado deverá existir, mas não pode existir um empregado sem departamento associado.

(B) não é possível ter ambas cadastradas com NOT NULL, mesmo com a avaliação postergada das restrições, pois no momento do COMMIT o registro referenciado pela chave estrangeira já precisa estar no banco de dados, para ser validado.

(C) ambas podem ser NOT NULL, desde que o nível de isolamento das transações permita leitura suja (*read uncommitted*).

(D) ambas podem ser NOT NULL, desde que o primeiro empregado e o primeiro departamento sejam inseridos na mesma transação e que as chaves estrangeiras sejam avaliadas somente ao final dela (no momento do COMMIT), o que pode ser conseguido declarando-as como sendo de avaliação postergada (DEFERRABLE).

(E) ambas podem ser NOT NULL, mas os sistemas não poderão usar transações para cadastrar os dados nessas tabelas.

Q7 - FCC – TRE RR – 2015

Considere a instrução Oracle PL/SQL a seguir.

CREATE VIEW valores (nome, minsal, maxsal, medsal)

AS SELECT d.depnome, MIN(e.sal), MAX(e.sal), AVG(e.sal)

FROM empregado e, departamento d

WHERE e.depnro=d.depnro

GROUP BY d.depnome;

Considere a existência das tabelas departamento e empregado, relacionadas de forma que cada departamento possa ter um ou muitos empregados ligados a ele. Na tabela departamento existem os campos depnro (chave primária) e depnome e na tabela empregado existem os campos empnro (chave primária), empnome, cargo, sal e depnro (chave estrangeira). Considere que em ambas as tabelas existem registros cadastrados relacionando adequadamente departamentos a empregados.

A instrução acima

a) está incorreta, pois não é possível criar view para exibir valores a partir de duas ou mais tabelas.

b) está incorreta, pois a subconsulta que define a view não pode conter a cláusula GROUP BY

c) está correta, porém, os apelidos definidos para as colunas não serão aplicados, pois eles deveriam estar na subconsulta e não após a cláusula CREATE VIEW.

d) está incorreta, pois a função para obter a média dos valores contidos no campo sal é MED e não AVG.

e) está correta, e a view será criada com os nomes de departamento e os valores mínimo, máximo e médio dos salários por departamento.

Q8 – CESGRANRIO - BNDES – Desenvolvimento – 2007

Um funcionário, encarregado de verificar o correto funcionamento de uma base de dados relacional, faz o seguinte teste:

```
select nome from emp where matr = 123;
```

O resultado é vazio. Então ele executa:

```
insert into emp(matr, nome, salario, ativo)

values (123, 'José da Silva', 2000, 'N');

commit;
```

O banco de dados não retorna erro e informa que inseriu uma linha. Por fim, para verificar, ele consulta novamente:

select nome from emp where matr = 123;

O resultado continua vazio.

Supondo que o sistema gerenciador de banco de dados esteja funcionando corretamente, que opção explica o ocorrido?

(A) Como o funcionário executou o primeiro SELECT momentos antes de executar o INSERT, o resultado ficou na memória cache do computador e não foi executado pelo banco de dados na segunda vez. Somente após o protocolo LRU ter retirado do *cache* o resultado do SELECT é que ele será novamente executado.

(B) Como "emp" é uma visão e uma visão é nada menos que uma consulta gravada no banco de dados, nunca é possível usá-la em operações de manipulação de dados. O COMMIT ignora a inserção anterior.

(C) "emp" é uma visão que retorna todos os empregados ativos (ativo='S'), mas foi criada sem a expressão WITH CHECK OPTION, que evitaria o problema acima.

(D) "emp" não é uma tabela, mas uma visão que retorna todos os empregados ativos (ativo='S') e foi criada com a expressão WITH CHECK OPTION. Dessa forma, como o empregado José da Silva não está ativo, o banco de dados não gravou o registro no momento do COMMIT.

(E) O funcionário executou o SELECT pouco tempo após a inserção do registro. Mesmo finalizando a transação com o COMMIT, o registro está em memória e ainda não foi gravado no disco. Somente após o CHECKPOINT é que o registro estará disponível para consulta.

Q9 - IDECAN – INMETRO - 2015

A linguagem SQL (Structured Query Language), uma linguagem padrão para utilização em banco de dados, é declarativa, ao contrário das linguagens tradicionais, que são do tipo procedimental. A linguagem SQL é constituída por três sublinguagens. Relacione adequadamente as colunas acerca das sublinguagens.

1. DML Data Manipulation Language.

2. DDL Data Definition Language.

3. DCL Data Control Language.

( ) Grant.

( ) Select.

( ) Insert.

( ) Create.

( ) Revoke.

( ) Update.

( ) Alter.

( ) Drop.

( ) Delete.

a) 3, 1, 1, 2, 3, 1, 2, 2, 1.

b) 2, 2, 1, 3, 3, 1, 2, 1, 3.

c) 1, 3, 1, 2, 2, 3, 3, 1, 2.

d) 3, 2, 1, 1, 2, 3, 2, 2, 1.

e) 2, 1, 1, 3, 1, 2, 1, 3, 2.

Q10 - FCC – TRF3 – 2014

ID	Nome	Idade
1	José Emírio	42
2	Antônio Silva	36

O comando em SQL utilizado para criar a tabela, é

a) ALTER TABLE Cadastro (ID TEXT, Nome VARCHAR, Idade INT);

b) CREATE INTO TABLE Cadastro VALUES(ID INTEGER, Nome TEXT, Idade TEXT);

c) CREATE TABLE Cadastro (ID INTEGER, Nome VARCHAR(30), Idade INTEGER);

d) CREATE Cadastro (ID INT PRIMARY KEY, Nome TEXT, Idade INT);

e) CREATE Cadastro NAMES (ID, Nome, Idade) TYPES(INT, TEXT,INT);

G**abarito**

Q1 – D

Q2 – B

Q3 – D

Q4 – A

Q5 – B

Q6 – D

Q7 – E

Q8 – C

Q9 – A

Q10 – C