

## CONCEITOS BÁSICOS DE BANCO DE DADOS

1. (ENADE 2005) Considere um sistema bancário simplificado e uma transação T1, que transfira R\$ 100,00 da conta X para a conta Y e é definida pelas operações listadas acima. Considere ainda que uma transação T2 esteja sendo executada simultaneamente com T1. Caso a transação T2 realize a operação Escrita(Y) depois da execução da operação 4 e antes da execução da operação 6 por T1, qual propriedade de transações será violada no banco de dados do referido sistema bancário?

T1	
1	Leitura(X);
2	$X = X - 100;$
3	Escrita(X);
4	Leitura(Y);
5	$Y = Y + 100;$
6	Escrita(Y);

- a) Atomicidade.
- b) Isolamento.
- c) Distributividade.
- d) Consistência.
- e) Durabilidade.

2. (POSCOMP 2002) Dentre as características do modelo relacional e do modelo de objetos em bancos de dados, qual afirmação é INCORRETA?

- a) O relacionamento de herança é diretamente representado no modelo relacional.
- b) O relacionamento binário N x M é representado de modo semelhante nos dois modelos.
- c) O modelo de objetos possui mais recursos estruturais para a representação de dados que o relacional.
- d) O modelo de objetos provê uma representação bem próxima de linguagens de programação.
- e) O modelo de objetos é mais adequado para a representação de tipos abstratos de dados.

3.(POSCOMP 2004) Transações em SGBD relacionais normalmente preenchem os requisitos ACID (atomicidade, consistência, isolamento e durabilidade). Considere as seguintes afirmações:

- I) *Isolamento* significa que o efeito das operações de alteração efetuadas por uma transação T não são vistas por outras transações, até que a transação T encerre.
- II) *Isolamento* significa que os dados protegidos para alteração por uma transação não podem ser protegidos para alteração por outra transação.
- III) *Durabilidade* significa que o efeito de uma operação (INSERT, DELETE ou UPDATE) sobre a base de dados não pode ser desfeito.

IV) *Durabilidade* significa que o efeito das operações de alteração executadas por uma transação não pode ser desfeito, após do final bem sucedido (COMMIT) da transação.

Quanto a estas afirmativas vale que:

- a) Somente as afirmativas I) e III) são corretas.
- b) Somente as afirmativas I) e IV) são corretas.
- c) Somente as afirmativas II) e III) são corretas.
- d) Somente as afirmativas II) e IV) são corretas.
- e) Somente a afirmativa I) é correta.

4. (POSCOMP 2006) Assinalar a opção correta acerca das sentenças abaixo:

- I. Os níveis de isolamento de uma transação SQL são: Read Uncommitted, Read Committed, Repeatable Read e Serializable.
- II. Atomicidade e Durabilidade são garantidas pelo Gerenciador de Recuperação do SGBD.
- III. São propriedades de uma transação: Atomicidade, Consistência, Integridade e Durabilidade.

- (a) Apenas I é verdadeira.
- (b) Apenas I e II são verdadeiras.
- (c) Apenas II e III são verdadeiras.
- (d) Apenas I e III são verdadeiras.
- (e) Todas são verdadeiras

5. (POSCOMP 2008) Considere que as transações **T1** e **T2** abaixo possam ocorrer simultaneamente.

<b>T1</b>	<b>T2</b>
Leitura(A); A = A + 100; Escrita(A); Leitura(B); B = B - 100; Escrita(B);	Leitura(B); Leitura(A); Print (A+B);

Analisar as seguintes situações.

I. A operação Leitura(A) de **T2** é executada após a operação Escrita(A) e antes da operação Leitura(B) de **T1**. Entretanto, a operação Escrita(B) de **T1** causa uma violação de integridade, e a transação **T1** é abortada, sendo suas operações desfeitas.

II. Após as operações da transação **T1** terem sido executadas, é enviada uma mensagem ao usuário informando que a transação foi completada com êxito. Entretanto, antes que os buffers relativos a **T1** sejam descarregados para o meio físico, ocorre uma falha, e os dados não são efetivamente gravados.

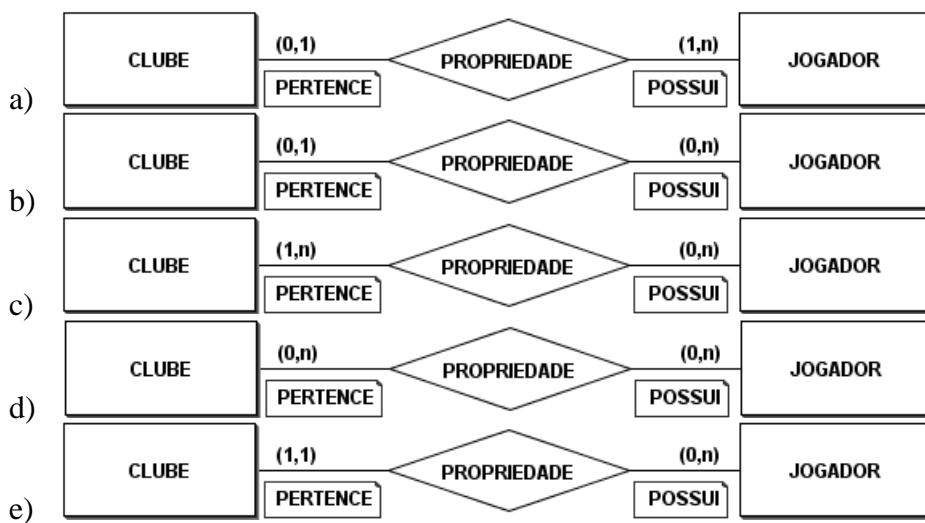
As propriedades das transações que foram violadas nessas duas situações são, *respectivamente*:

- A) Atomicidade e Consistência.
- B) Durabilidade e Atomicidade.
- C) Atomicidade e Durabilidade.
- D) Durabilidade e Isolamento.
- E) Isolamento e Durabilidade.

### MODELO ENTIDADE RELACIONAL

1. (ENADE 2005) Todo jogador deve pertencer a um único clube.

Assinale a opção que representa corretamente, no modelo entidade-relacionamento, a especificação apresentada acima.



### MODELO RELACIONAL

1. (ENADE 2008) Considere o esquema de banco de dados relacional apresentado a seguir, formado por 4 relações, que representa o conjunto de estudantes de uma universidade que podem, ou não, morar em repúblicas (moradias compartilhadas por estudantes). A relação Estudante foi modelada como um subconjunto da relação Pessoa. Considere que os atributos grifados correspondam à chave primária da respectiva relação e os atributos que são seguidos da palavra referencia sejam chaves estrangeiras.

Pessoa(IdPessoa:integer, Nome:varchar(40), Endereco:varchar(40))

FonePessoa(IdPessoa:integer referencia Pessoa, DDD:varchar(3), Prefixo:char(4), Nro:char(4))

Republica(IdRep:integer, Nome:varchar(30), Endereco:varchar(40))

Estudante(RA:integer, Email:varchar(30), IdPessoa:integer referencia Pessoa, IdRep:integer referencia Republica)

Suponha que existam as seguintes tuplas no banco de dados:

Pessoa (1, 'José Silva', 'Rua 1, 20');

Republica (20, 'Várzea', 'Rua Chaves, 2001')

Qual opção apresenta apenas tuplas válidas para esse esquema de banco de dados relacional?

- a) Estudante(10, 'jsilva@ig.com.br', null, 20);  
FonePessoa(10, '019', '3761', '1370')
- b) Estudante(10, 'jsilva@ig.com.br', 1, null);  
FonePessoa(10, '019', '3761', '1370')
- c) Estudante(10, 'jsilva@ig.com.br', 1, 20);  
FonePessoa(1, null, '3761', '1370')
- d) Estudante(10, 'jsilva@ig.com.br', 1, 50);  
FonePessoa(1, '019', '3761', '1370')
- e) Estudante(10, 'jsilva@ig.com.br', 1, null);  
FonePessoa(1, '019', '3761', '1370')

2. (POSCOMP 2003) Considere a seguinte tabela para uma base de dados relacional:

Empregado (CodEmp, NomeEmp, CodDepto)

Considere que esta tabela tem um índice na forma de uma árvore B sobre as colunas (CodEmp, CodDepto), nesta ordem.

Quanto a este índice, considere as seguintes afirmativas:

- 1) Este índice pode ser usado pelo SGBD relacional para acelerar uma consulta na qual são fornecidos os valores de CodEmp e CodDepto.
- 2) Este índice pode ser usado pelo SGBD relacional para acelerar uma consulta na qual é fornecido um valor de CodEmp.
- 3) Este índice não é adequado para ser usado pelo SGBD relacional para acelerar uma consulta na qual é fornecido um valor de CodDepto.
- 4) O algoritmo que faz inserções e remoções de entradas do índice tem por objetivo garantir que o índice fique organizado de tal forma que o acesso a cada nodo da árvore implique em número de acessos semelhantes.
- 5) O índice por árvore-B não é adequado para tabelas que sofrem grande número de inclusões e exclusões, pois exige reorganizações freqüentes.

Quanto a estas afirmativas pode se dizer que:

- (a) Nenhuma das afirmativas está correta
- (b) Apenas as afirmativas 1), 2), 3) e 4) estão corretas
- (c) Todas afirmativas estão corretas
- (d) Apenas as afirmativas 1), 2) e 4) estão corretas
- (e) Apenas as afirmativas 1), 2) e 5) estão corretas

3. (POSCOMP 2004) Considere um modelo entidade-relacionamento de uma indústria.

Este modelo representa peças com suas propriedades, bem como a composição de peças (peças podem ser usadas na composição de outras peças).

Este modelo entidade relacionamento contém:

- Uma entidade *Peça*, com atributos código e nome e peso.
- Um auto-relacionamento de *Peça*, chamado *Composição*. Neste auto-relacionamento uma peça tem papel de *componente* e outra peça papel de *composto*. O auto-relacionamento tem cardinalidade n:n e tem um atributo *quantidade*, que informa quantas unidades da peça componente são usadas na peça composto.

Uma base de dados relacional que implementa corretamente este modelo entidade relacionamento e está na terceira forma normal é (chaves primárias estão sublinhadas):

a) Peca (CodPeca, NomePeca, PesoPeca)  
 Composicao(CodPecaComposto,CodPecaComponente,Quantidade)  
 Composição.CodPecaComposto referencia Peca  
 Composição.CodPecaComponente referencia Peca

b) Peca (CodPeca, NomePeca, PesoPeca, CodPecaComposto, Quantidade)  
 Peca.CodPecaComposto referencia Peca

c) Peca (CodPeca, NomePeca, PesoPeca)  
 Composicao(CodPecaComposto,CodPecaComponente,Quantidade)  
 Composição.CodPecaComposto referencia Peca  
 Composição.CodPecaComponente referencia Peca

d) Peca (CodPeca, NomePeca, PesoPeca, CodPecaComposto, CodPecaComponente,Quantidade)  
 Peca.CodPecaComposto referencia Peca  
 Peca.CodPecaComponente referencia Peca

e) Composto (CodPeca, NomePeca, PesoPeca)  
 Componente (CodPeca, NomePeca, PesoPeca, CodPecaComposto, Quantidade)  
 Componente.CodPecaComposto referencia Composto

4. (POSCOMP 2005) Em um banco de dados relacional, considere os esquemas de relação:

Pessoa (CPF, Pro\_ssao)  
 Trabalha (CPF, CGC, Período)  
 Firma (CGC, nome, endereço)

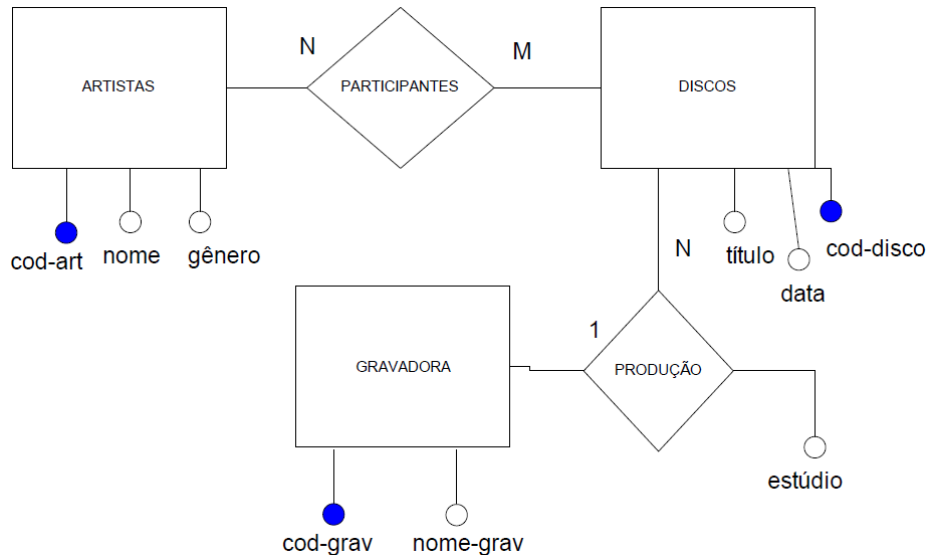
e considere as operações de álgebra relacional União, Interseção, Diferença, Junção Natural, Projeção e Seleção.

A consulta “Qual a profissão das pessoas que trabalham em alguma firma de nome X” exige ao menos a seguinte operação para ser processada:

- Interseção de Pessoa, Trabalha e Firma.
- Junção Natural de Pessoa, Trabalha e Firma.

- (c) União de Pessoa, Trabalha e Firma.  
 (d) Seleção de Pessoa, Trabalha e Firma.  
 (e) Nada pode ser afirmando porque os dados não foram fornecidos.

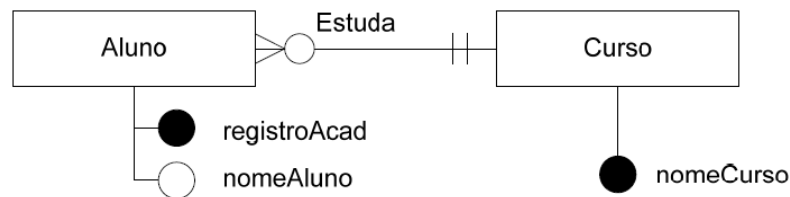
5. (POSCOMP 2008) Considere o projeto lógico do banco de dados representado pelo modelo E-R abaixo.



Diga de que tabelas e campos ele resulta (obs: campos chave estão sublinhados):

- A) ARTISTAS (cod-art, nome, gênero)  
 PARTICIPANTES (cod-art, cod-disco)  
 DISCOS (cod-disco, título, data)  
 PRODUÇÃO (cod-grav, cod-disco, estúdio)  
 GRAVADORA (cod-grav, nome-grav)
- B) ARTISTAS (cod-art, nome, gênero)  
 PARTICIPANTES (cod-art, cod-disco)  
 DISCOS (cod-disco, título, data, cod-grav, estúdio)  
 GRAVADORA (cod-grav, nome-grav)
- C) ARTISTAS (cod-art, nome, gênero)  
 PARTICIPANTES (cod-art, cod-disco)  
 DISCOS (cod-disco, título, data, cod-grav, nome-grav, estúdio)
- D) ARTISTAS (cod-art, nome, gênero)  
 DISCOS (cod-disco, título, data, cod-art)  
 PRODUÇÃO (cod-grav, cod-disco, estúdio)  
 GRAVADORA (cod-grav, nome-grav)
- E) ARTISTAS (cod-art, nome, gênero)  
 DISCOS (cod-disco, título, data, cod-grav, estúdio, cod-art)  
 GRAVADORA (cod-grav, nome-grav)

6. (POSCOMP 2009) Dado o diagrama de entidades e relacionamentos abaixo, qual o conjunto de relações que representam as tabelas estritamente necessárias para implementá-lo, onde as chaves primárias aparecem sublinhadas:



A) Aluno (registroAcad, nomeAluno)  
Curso (nomeCurso, registroAcad)

B) Aluno (codAluno, registroAcad, nomeAluno, codCurso)  
Curso (codCurso, nomeDept)

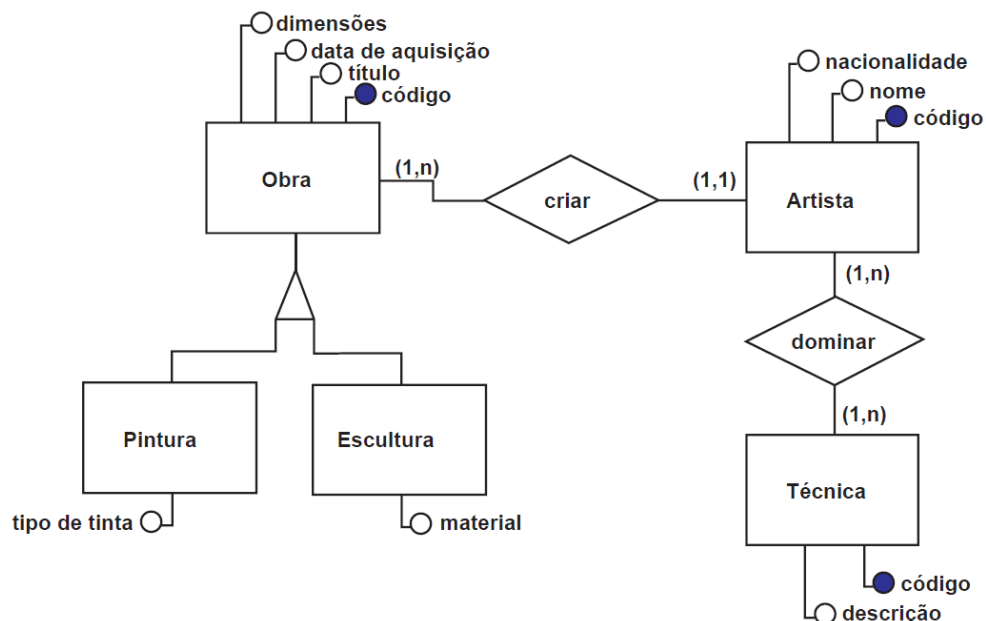
C) Aluno (codAluno, registroAcad, nomeAluno)  
Curso (codCurso, nomeCurso)  
Estuda(codAluno, codCurso)

D) Aluno (registroAcad, nomeAluno)  
Curso (nomeCurso)  
Estuda (registroAcad, nomeCurso)

E) Aluno (registroAcad, nomeAluno, nomeCurso)  
Curso (nomeCurso)

7. (ENADE 2017)

O diagrama Entidade-Relacionamento (DER) a seguir apresenta a modelagem conceitual de dados de um sistema de informação para um museu. A partir dessa modelagem observa-se o seguinte: uma Obra é criada por um único Artista e um Artista pode criar no mínimo uma Obra e no máximo várias Obras; as entidades Pintura e Escultura são especializações da entidade Obra; um Artista tem o domínio de várias Técnicas, assim como uma Técnica é dominada por diversos Artistas.



Com base nas regras de mapeamento que transformam o Modelo Conceitual em um Modelo Lógico Relacional, avalie as afirmações a seguir, a respeito do Esquema Lógico Relacional gerado a partir do DER apresentado.

- I. No Esquema Lógico Relacional, haverá uma tabela associativa, criada em função do relacionamento muitos para muitos entre as entidades Artista e Técnica, que terá uma chave primária composta pelo código do artista e o código da técnica.
- II. No Esquema Lógico Relacional, haverá uma tabela Artista na qual o atributo código do artista será a chave primária da tabela, e o código da obra será uma chave estrangeira que fará referência a uma obra existente na tabela Obra.
- III. No Esquema Lógico Relacional, haverá, em função da generalização/especialização, uma tabela Obra com apenas os atributos código da obra, título, data de aquisição e dimensões, e duas outras tabelas: a tabela Pintura, com apenas o atributo tipo de tinta, e a tabela Escultura com apenas o atributo material.

É correto o que se afirma em

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

8. (POSCOMP 2016) Considere um banco de dados para apoiar a correção das provas do POSCOMP. Sabe-se que há as relações CANDIDATO, QUESTAO e RESPOSTA. O atributo X da relação QUESTAO é uma chave estrangeira. Com base apenas nessas informações, analise as assertivas abaixo sobre a definição de restrições para essa chave estrangeira:

- I. Pode ter valores repetidos nas tuplas de QUESTAO.
- II. Pode ter valor nulo em algumas das tuplas de QUESTAO.
- III. Pode referenciar, ao mesmo tempo, as relações CANDIDATO e RESPOSTA.
- IV. Pode referenciar outra chave estrangeira.
- V. Pode ter valor não nulo distinto de todos os valores presentes na chave primária da relação que referencia.

Quais estão corretas?

- A) Apenas I e II.
- B) Apenas I e III.
- C) Apenas II e V.
- D) Apenas III e IV.
- E) Apenas IV e V.

## NORMALIZAÇÃO

1. (ENADE 2008) Considere o esquema de relação:

Cliente (CPF, nome, RGemissor, RGnro, endereco, loginemail, dominioemail)



e as seguintes dependências funcionais (DF) válidas sobre o esquema:

DF1: CPF  $\div$  nome, RGemissor, RGnro, endereço, loginemail, dominioemail

DF2: RGemissor, RGnro  $\div$  CPF, nome, endereço, loginemail, Dominioemail

DF3: loginemail, dominioemail  $\div$  CPF

Qual é o conjunto completo de chaves candidatas de Cliente e em que forma normal mais alta essa relação está?

- a) {(RGemissor, RGnro), (CPF)}, na Forma Normal de Boyce-Codd (FNBC).
- b) {(RGemissor, RGnro), (CPF)}, na Segunda Forma Normal (2FN).
- c) {(loginemail, dominioemail)}, na Forma Normal de Boyce-Codd (FNBC).
- d) {(RGemissor, RGnro), (loginemail, dominioemail),(CPF)}, na Forma Normal de Boyce-Codd (FNBC).
- e) {(RGemissor, RGnro), (loginemail, dominioemail),(CPF)}, na Segunda Forma Normal (2FN).

2. (POSCOMP 2003) Considere a seguinte tabela em uma base de dados relacional (chave primária sublinhada):

Tabela1(CodAluno,CodDisciplina,AnoSemestre, NomeAluno, NomeDisciplina, CodNota, DescricaoNota)

Considere as seguintes dependências funcionais:

CodAluno  $\rightarrow$  NomeAluno

CodDisciplina  $\rightarrow$  NomeDisciplina

(CodAluno,CodDisciplina,AnoSemestre)  $\rightarrow$  CodNota

(CodAluno,CodDisciplina,AnoSemestre)  $\rightarrow$  DescricaoNota

CodNota  $\rightarrow$  DescricaoNota

Considerando as formas normais, qual das afirmativas abaixo se aplica:

- (a) A tabela encontra-se na primeira forma normal, mas não na segunda forma normal.
- (b) A tabela encontra-se na segunda forma normal, mas não na terceira forma normal.
- (c) A tabela encontra-se na terceira forma normal, mas não na quarta forma normal.
- (d) A tabela não está na primeira forma normal.
- (e) A tabela está na quarta forma normal.

3. (POSCOMP 2005) Qual das seguintes informações é verdadeira?

- (a) Nem toda relação que está na FNBC (Forma Normal de “Boyce-Codd”) está também na 3FN (Terceira Forma Normal).
- (b) Se a relação R possui somente uma chave candidata, ela sempre está na FNBC.
- (c) Se a relação R está na 3FN e toda chave candidata é simples, então não podemos afirmar que R está na FNBC.
- (d) Uma dependência funcional multivalorada na relação R, na forma  $X \twoheadrightarrow Y$ , é dita trivial somente se  $XY = R$ .
- (e) Uma dependência funcional multivalorada na relação R, na forma  $X \twoheadrightarrow Y$ , é dita trivial se  $Y \subseteq X$  ou  $XY = R$ .

4. (POSCOMP 2009) Considere a relação abaixo, obtida via processo de engenharia reversa em documentos de uma empresa.

Emp (CodEmp, CodDept, CodMaq, Nome, Sala, NomeDept, NomeMáquina)

Através de um processo de normalização (não necessariamente completo), chegou-se ao seguinte conjunto de relações:

R1 (CodEmp, Nome, CodDept, CodMaq)

R2 (CodDept, NomeDept, Sala)

R3 (CodMaq, NomeMáquina).

Considere que as seguintes dependências funcionais se aplicam a estas relações:

$\text{CodEmp} \rightarrow \text{Nome}$

$\text{CodDept} \rightarrow \{\text{NomeDept}, \text{Sala}\}$

$\text{CodMáquina} \rightarrow \text{NomeMáquina}$

Assinale a alternativa **CORRETA**:

- A) A relação Emp encontra-se na segunda forma normal (2FN).
- B) Todas as três relações R1, R2 e R3 encontram-se na segunda forma normal (2FN).
- C) Somente as relações R1 e R3 encontram-se na segunda forma normal (2FN).
- D) Somente a relação R3 encontra-se na terceira forma normal (3FN).
- E) Nenhuma das afirmativas anteriores é verdadeira.

5. (POSCOMP 2010) O processo de normalização baseia-se no conceito de forma normal, que é uma regra que deve ser obedecida por uma relação para que seja considerada bem projetada. Com base nos conhecimentos sobre normalização, considere as afirmativas a seguir.

I. A Primeira Forma Normal (1FN) define que a relação não deve conter atributos não atômicos ou as relações aninhadas. A ação que deve ser tomada para deixar uma relação na 1FN é formar uma nova relação para cada atributo não atômico ou para cada relação aninhada.

II. A Segunda Forma Normal (2FN) define que, além de estar na 1FN, para as relações que possuam chaves primárias com vários atributos, nenhum atributo externo à chave deve ser funcionalmente dependente de parte da chave primária. A ação que deve ser

tomada é decompor e montar uma nova relação para cada chave parcial com seu(s) atributo(s) dependente(s).

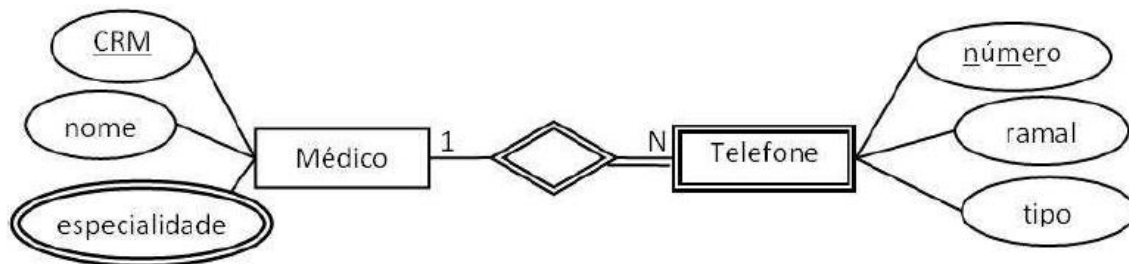
III. A Terceira Forma Normal (3FN) define que, além de estar na 2FN, as relações não devem ter atributos que não pertençam a uma chave, funcionalmente determinados por outro atributo que também não pertença a uma chave (ou por um conjunto de atributos não chave). A ação que deve ser tomada é decompor e montar uma relação que contenha o(s) atributo(s) não chave que determina(m) funcionalmente o(s) outro(s) atributo(s).

IV. Uma dependência parcial ocorre quando um atributo, além de depender da chave primária, depende de outro atributo ou conjunto de atributos da relação. Uma dependência transitiva ocorre quando um atributo depende apenas de parte de uma chave primária composta.

Assinale a alternativa correta.

- a) Somente as afirmativas I e IV são corretas.
- b) Somente as afirmativas II e III são corretas.
- c) Somente as afirmativas III e IV são corretas.
- d) Somente as afirmativas I, II e III são corretas.
- e) Somente as afirmativas I, II e IV são corretas.

6. (POSCOMP 2013) Considere o diagrama ER (Entidade-Relacionamento), a seguir, sobre Médicos, suas especialidades e vários telefones.



Nesse diagrama, as entidades são retângulos, os relacionamentos são losangos, os atributos são ovais, os atributos multivalorados são ovais com linhas duplas, as entidades fracas são retângulos com linhas duplas e os relacionamentos identificadores são losangos com linhas duplas. Esse diagrama precisa ser mapeado a fim de armazenar dados em um Sistema de Gerenciamento de Bancos de Dados Relacional.

Com base nas regras de mapeamento e da Terceira Forma Normal (3FN), atribua V (verdadeiro) ou F (falso)

às afirmativas a seguir.

- ( ) A chave primária da tabela Telefone será composta por CRM e número.
- ( ) Uma tabela específica será criada para Médico e outra para Telefone, com uma chave estrangeira.
- ( ) Uma tabela específica será criada para Médico e outra para Telefone, sem qualquer chave estrangeira.
- ( ) Uma tabela específica será criada para o atributo Especialidade com uma chave estrangeira para a tabela Médico.

( ) Uma tabela específica será criada para o relacionamento entre Médico e Telefone, com as respectivas chaves estrangeiras.

Assinale a alternativa que contém, de cima para baixo, a sequência correta.

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

## SQL

1. (ENADE 2005) 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;`

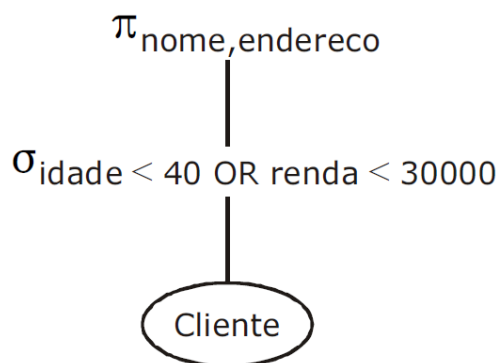
2. (ENADA 2008) O banco de dados de um sistema de controle bancário implementado por meio de um SGBD relacional possui a relação Cliente, com as informações apresentadas a seguir, em que a chave primária da relação é grifada.

Cliente (nroCliente, nome, endereco, data\_nascimento, renda, idade)

Para essa relação, foram criados dois índices secundários: ÍndiceIdade, para o atributo idade, e ÍndiceRenda, para o atributo renda. Existe um tipo de serviço nesse banco cujo alvo são tanto os clientes que possuem menos de 40 anos de idade quanto aqueles que possuem renda mensal superior a 30.000 reais. Para recuperar esses clientes, a seguinte expressão de consulta em SQL foi utilizada:

```
SELECT nome, endereco  
FROM Cliente  
WHERE idade < 40 OR renda > 30000;
```

Com o aumento do número de clientes desse banco, essa consulta passou a apresentar problemas de desempenho. Verificou-se, então, que o otimizador de consultas não considerava os índices existentes para idade e renda, e a consulta era realizada mediante varredura seqüencial na relação Cliente, tornando essa consulta onerosa. O plano de execução da consulta, usado pelo otimizador, é apresentado na árvore de consulta abaixo, na qual B e F representam as operações de projeção e de seleção, respectivamente.



Para que o otimizador de consultas passasse a utilizar os índices, a solução encontrada foi elaborar a consulta em dois blocos separados — um que recupera os clientes com idade inferior a 40 anos, e outro que recupera os clientes com renda mensal superior a 30.000 reais — para, então, juntar as tuplas das duas relações geradas.

Considerando a situação apresentada, faça o que se pede a seguir.

- Escreva o código de uma consulta em SQL que corresponda à solução proposta.
- Desenhe a árvore de consulta para essa solução.

3. (ENADE 2008) Considere as seguintes tabelas:

```
CREATE TABLE Departamento
```

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

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.

4. (ENADE 2011) 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

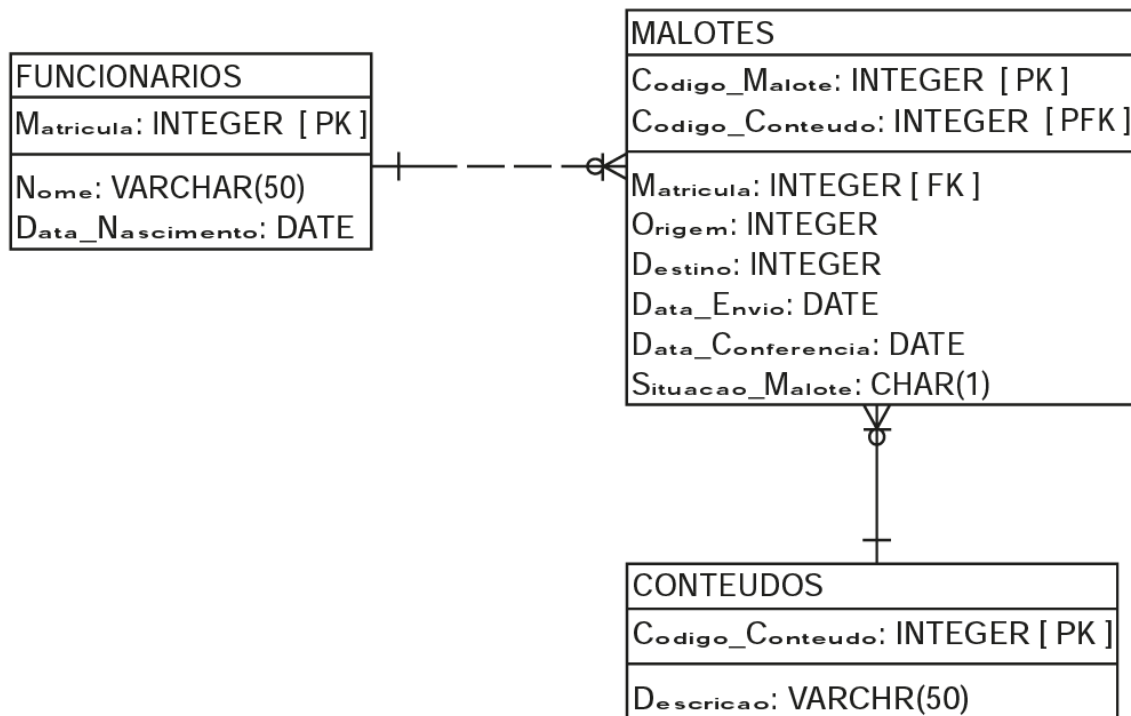
D) SELECT CodDepartamento, AVG (Salario) > 2000  
FROM EMPREGADO  
GROUP BY CodDepartamento

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

5. (ENADE 2011) 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:

- A) SELECT NOME,CODIGO\_MALOTE,DESCRICAO,SITUACAO\_MALOTE  
FROM MALOTES INNER JOIN CONTEUDOS ON (CODIGO\_CONTEUDO =  
CODIGO\_CONTEUDO) INNER JOIN FUNCIONARIOS ON (MATRICULA =  
MATRICULA);
- B) SELECT NOME, CODIGO\_MALOTE, DESCRICAO, SITUACAO\_MALOTE  
FROM MALOTES, CONTEUDOS, FUNCIONARIOS WHERE  
(CODIGO\_CONTEUDO = CODIGO\_CONTEUDO) AND (MATRICULA =  
MATRICULA);
- C) 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);
- D) 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);
- E) 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);

6. (ENADE 2014) O modelo de entidade relacionamento apresentado, representa de forma sucinta uma solução para persistência de dados de uma biblioteca. Considerando que um livro está emprestando quando possuir um registro vinculado a ele na tabela "Emprestimo", e essa tupla não possuir valor na coluna "data\_dev", o comando SQL que deve ser utilizado para listar os títulos dos livros disponíveis para empréstimo é:

A) select titulo from livro

except

select l.titulo from emprestimo e inner join livro l  
on e.livro\_cod = l.liv\_cod where e.data\_dev is null

B) select titulo from livro

union

select l.titulo from emprestimo e inner join livro l  
on e.livro\_cod = l.liv\_cod where e.data\_dev is null

C) select titulo from livro

except

select l.titulo from emprestimo e inner join livro l  
on e.livro\_cod = l.liv\_cod where e.data\_dev is not null

D) select titulo from livro

union select l.titulo from emprestimo e left join livro l  
on e.livro\_cod = l.liv\_cod where e.data\_dev is null

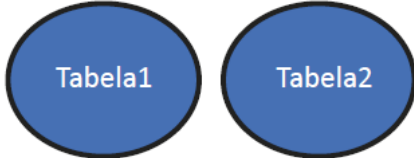
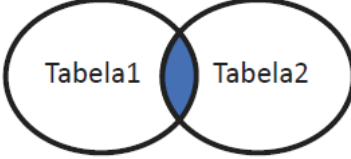
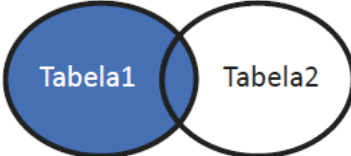
E) select titulo from livro

except

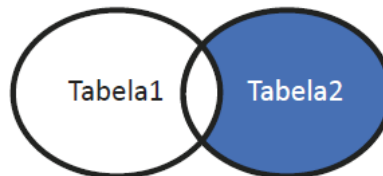
select l.titulo from emprestimo e right join livro l  
on e.livro\_cod = l.liv\_cod where e.data\_dev is not null

7. (ENADE 2017)

No modelo de dados relacional, uma junção entre tabelas cria uma pseudo-tabela derivada de duas ou mais tabelas de acordo com os critérios especificados, similar às regras da teoria dos conjuntos. Nos conjuntos a seguir, as áreas hachuradas representam os resultados das consultas SQL (padrão) em que, *id* é uma chave primária e *fk* uma chave estrangeira.

Conjuntos	Consultas SQL equivalentes
	<pre>SELECT * FROM Tabela1; SELECT * FROM Tabela2;</pre>
	<pre>SELECT * FROM Tabela1 t1; INNER JOIN Tabela2 t2 ON t1.id = t2.fk;</pre>
	<pre>SELECT * FROM Tabela1 t1; LEFT JOIN Tabela2 t2 ON t1.id = t2.fk;</pre>

Assinale a opção em que a declaração SQL é equivalente ao conjunto apresentado a seguir.



- A** `SELECT * FROM Tabela1 t1 JOIN Tabela2 t2 ON t1.id = t2.fk;`
- B** `SELECT * FROM Tabela1 t1 RIGHT JOIN Tabela2 ON t1.id = t2.fk WHERE t2.fk <> NULL;`
- C** `SELECT * FROM Tabela1 t1 WHERE EXISTS (SELECT 1 FROM Tabela2 t2 WHERE t2.id = t1.fk);`
- D** `SELECT * FROM Tabela1 t1 RIGHT JOIN Tabela2 t2 ON t1.id = t2.fk WHERE t1.id is NULL;`
- E** `SELECT * FROM Tabela1 t1 WHERE NOT EXISTS (SELECT 1 FROM Tabela2 t2 WHERE t2.id = t1.fk);`

8. (POSCOMP 2002) Supondo a Relação PROJ (PNO, Orçam), com chave primária PNO, a Relação EMP (ENO, ENome, Cargo) com chave primária ENO, e a Relação DSG (ENO, PNO, Dur, Resp), com chave primária {ENO, PNO}, chave estrangeira PNO em relação a PROJ e chave estrangeira ENO em relação a EMP. Qual das expressões da álgebra relacional abaixo NÃO corresponde à seguinte consulta SQL:

```
SELECT ENome
FROM EMP, PROJ, DSG
WHERE EMP.ENO = DSG.ENO
AND PROJ.PNO = DSG.PNO
AND Dur > 36
```

- a)  $\pi_{\text{ENome}} (\text{PROJ} \bowtie_{\text{PNO}} (\text{EMP} \bowtie_{\text{ENO}} \sigma_{\text{Dur} > 36} (\text{DSG})))$   
 b)  $\pi_{\text{ENome}} (\text{PROJ} \bowtie_{\text{PNO}} ((\pi_{\text{ENome}, \text{ENO}} (\text{EMP})) \bowtie_{\text{ENO}} (\sigma_{\text{Dur} > 36} (\text{DSG}))))$   
 c)  $\pi_{\text{ENome}} (\text{PROJ} \bowtie_{\text{PNO}} (\sigma_{\text{Dur} > 36} (\text{EMP} \bowtie_{\text{ENO}} (\text{DSG}))))$   
 d)  $\pi_{\text{ENome}} (\sigma_{\text{Dur} > 36} ((\pi_{\text{PNO}} (\text{PROJ})) \bowtie_{\text{PNO}} (\text{EMP} \bowtie_{\text{ENO}} \text{DSG})))$   
 e)  $\pi_{\text{ENome}} (\text{PROJ} \bowtie_{\text{PNO}} (\text{EMP} \bowtie_{\text{ENO}} \sigma_{\text{Dur} > 36} (\pi_{\text{Dur}} (\text{DSG}))))$

9. (POSCOMP 2003) Considere as seguintes tabelas em uma base de dados relacional:

Departamento (CodDepto, NomeDepto)

Empregado (CodEmp, NomeEmp, CodDepto)

Deseja-se obter uma tabela na qual cada linha é a concatenação de uma linha da tabela Departamento com uma linha da tabela de Empregado. Caso um departamento não possua empregados, sua linha no resultado deve conter vazio (NULL) nos campos referentes ao empregado. A operação de álgebra relacional que deve ser aplicada para combinar estas duas tabelas é:

- (a) Divisão  
 (b) Junção interna  
 (c) Junção externa  
 (d) União  
 (e) Projeção

10. (POSCOMP 2003) Considere as seguintes tabelas em uma base de dados relacional (chaves primárias sublinhadas):

Departamento (CodDepto, NomeDepto)

Empregado (CodEmp, NomeEmp, CodDepto)

Considere as seguintes restrições de integridade sobre esta base de dados relacional:

- Empregado.CodDepto é sempre diferente de NULL
- Empregado.CodDepto é chave estrangeira da tabela Departamento com cláusulas ON DELETE RESTRICT e ON UPDATE RESTRICT

Qual das seguintes validações não é especificada por estas restrições de integridade:

- (a) Sempre que uma nova linha for inserida em Empregado, deve ser garantido que o valor de Empregado.CodDepto aparece na coluna Departamento.CodDepto.  
 (b) Sempre que uma linha for excluída de Departamento, deve ser garantido que o valor de Departamento.CodDepto não aparece na coluna Empregado.CodDepto.  
 (c) Sempre que o valor de Empregado.CodDepto for alterado, deve ser garantido que o novo valor de Empregado.CodDepto aparece em Departamento.CodDepto.  
 (d) Sempre que o valor de Departamento.CodDepto for alterado, deve ser garantido que não há uma linha com o antigo valor de Departamento.CodDepto na coluna Empregado.CodDepto  
 (e) Sempre que uma nova linha for inserida em Departamento, deve ser garantido que o valor de Departamento.CodDepto aparece na coluna Empregado.CodDepto

11. (POSCOMP 2003) Considere as seguintes tabelas em uma base de dados relacional:

Departamento (CodDepto, NomeDepto)

Empregado (CodEmp, NomeEmp, CodDepto, Salario)

Considere a seguinte consulta escrita em SQL:

```
SELECT D.CodDepto, NomeDepto, SUM(E.Salario)
FROM Departamento D, Empregado E
WHERE D.CodDepto=E.CodDepto
GROUP BY D.CodDepto, NomeDepto
HAVING COUNT(*)>2 AND AVG(E.Salario)>40
```

A consulta acima obtém o seguinte resultado:

- (a) Para cada empregado que tem mais que dois departamentos, ambos com média salarial maior que 40, obter o código de departamento, seguido do nome do departamento, seguido da soma dos salários dos empregados do departamento.
- (b) Para cada departamento que tem mais que dois empregados e cuja média salarial é maior que 40, obter o código de departamento, seguido do nome do departamento, seguido da soma dos salários dos empregados do departamento.
- (c) Para cada departamento que tem mais que dois empregados e cuja média salarial, considerando todos empregados do departamento, exceto os dois primeiros, é maior que 40, obter o código de departamento, seguido do nome do departamento, seguido da soma dos salários dos empregados do departamento.
- (d) A consulta não retorna nada pois está incorreta.
- (e) Para cada departamento que tem mais que dois empregados e cuja média salarial é maior que 40 obter um grupo de linhas que contém, para cada empregado do departamento, o código de seu departamento, seguido do nome de seu departamento, seguido da soma dos salários dos empregados do departamento.

12. (POSCOMP 2004) Considere as seguintes tabelas em uma base de dados relacional, contendo informações sobre empregados, departamentos e a vinculação entre eles:

Departamento (CodDepto, NomeDepto)

Empregado (CodEmp, NomeEmp, CodDepto)

Deseja-se obter os nomes dos departamentos (NomeDepto) que não estão vinculados a nenhum Empregado. Para obter este resultado a consulta correta em SQL é a seguinte:

a) 

```
SELECT NomeDepto
FROM Departamento D, Empregado E
WHERE D.CodDepto=E.CodDepto AND
E.CodEmp IS NULL
```

b) 

```
SELECT NomeDepto
FROM Departamento D
WHERE EXISTS
(SELECT *
FROM Empregado E
WHERE CodDepto=D.CodDepto)
```

c) (SELECT NomeDeppto  
FROM Departamento D)  
EXCEPT  
(SELECT NomeDeppto  
FROM Departamento D, Empregado E  
WHERE D.CodDeppto=E.CodDeppto)

d) SELECT NomeDeppto  
FROM Departamento D, Empregado E  
WHERE D.CodDeppto<>E.CodDeppto

e) SELECT NomeDeppto  
FROM Departamento D  
WHERE CodDeppto IN  
(SELECT CodDeppto  
FROM Empregado E

13. (POSCOMP 2004) Considere as seguintes tabelas em uma base de dados relacional, contendo

informações sobre empregados, departamentos e a vinculação entre eles:

Departamento (CodDeppto, NomeDeppto)

Empregado (CodEmp, NomeEmp, CodDeppto, SalarioEmp)

Considere a seguinte consulta sobre esta base de dados:

```
SELECT D.CodDeppto, AVG(SalarioEmp)
FROM Departamento D,
Empregado E
WHERE E.CodDeppto=D.CodDeppto AND
E.SalarioEmp > 300
GROUP BY D.CodDeppto
HAVING COUNT(*) > 20
```

Esta consulta SQL tem o seguinte resultado:

a) Para departamentos com mais que 20 empregados que tenham salário maior que 300, obter o código do departamento e a média salarial dos empregados do departamento.

b) Para departamentos que tem mais que 20 empregados nos quais todos empregados tem salário maior que 300, obter o código do departamento e a média salarial dos empregados que ganham mais que 300.

c) Para departamentos que tem mais que 20 empregados nos quais todos empregados tem salário maior que 300, obter o código do departamento e a média salarial dos empregados do departamento.

d) Para departamentos que tem mais que 20 empregados, obter o código do departamento e a média salarial dos empregados que ganham mais que 300.

e) Para departamentos com mais que 20 empregados que tenham salário maior que 300, obter o código do departamento e a média salarial dos empregados do departamento que ganham mais que 300.

14. (POSCOMP 2006) Sobre os operadores da Álgebra Relacional, é correto afirmar que:

- (a) O operador de SELEÇÃO seleciona as colunas de uma tupla que satisfazem a uma determinada condição.
- (b) O numero de tuplas resultantes da aplicação do operador de PROJEÇÃO em uma dada relação R é sempre igual ao numero de tuplas de R.
- (c) O numero de tuplas resultantes da aplicação do operador de JUNÇÃO em duas relações R e S é sempre maior do que o numero de tuplas resultantes do PRODUTO CARTESIANO de R e S.
- (d) A aplicação das operações de UNIÃO e INTERSEÇÃO requerem que as relações envolvidas sejam compatíveis quanto a união.
- (e) O numero de tuplas resultantes da aplicação do operador de SELEÇÃO em uma relação R é sempre menor do que o numero de tuplas de R.

15. (POSCOMP 2006) Considere os seguintes esquemas de relação:

Departamentos (codDepto, nome, gerente)  
 Empregados (codEmp, nome, codDepto, salario)

Considere também que o atributo codDepto na relação Empregados é uma chave estrangeira que faz referência a relação Departamentos. Suponha a seguinte consulta formulada na linguagem SQL:

```
SELECT d.codDepto
FROM Empregados e, Departamentos d
WHERE e.codDepto = d.codDepto
GROUP BY d.codDepto
HAVING AVG(sal) > ALL (SELECT e.sal
FROM Empregados e, Departamentos d
WHERE e.codDepto = d.codDepto
AND d.nome = 'vendas)
```

Escolha, dentre as afirmativas abaixo, a correta:

- (a) A consulta retorna os códigos dos departamentos cujos empregados têm salário maior do que a média dos salários dos empregados que trabalham no departamento de vendas.
- (b) A consulta retorna os códigos dos departamentos cujos empregados têm salário maior do que os salários dos empregados que trabalham no departamento de vendas.
- (c) A consulta retorna os códigos dos departamentos cuja média de salário dos seus empregados é maior do que a média dos salários dos empregados que trabalham no departamento de vendas.
- (d) A consulta está formulada incorretamente.
- (e) Nenhuma das afirmativas acima está correta.

16. (POSCOMP 2007) Considere um banco de dados com as seguintes tabelas e campos:

ALUNOS (nome-aluno, código-aluno, cidade, código-curso)  
 CURSOS (nome-curso, código-curso, carga-horária)

Assinale a alternativa que apresenta a forma mais otimizada de realizar a consulta “encontrar o nome dos alunos que pertencem ao curso Computação”. (operações em ordem de execução)

- (a) Junção de cursos com alunos, seleção de linhas em que nome-curso = “Computação”, projeção do resultado sobre nome-aluno.
- (b) Junção de cursos com alunos, projeção do resultado sobre nome-aluno, seleção de linhas em que nome-curso = “Computação”.
- (c) Seleção de linhas em cursos em que nome-curso = “Computação”, projeção do resultado sobre código-curso, junção com alunos, projeção do resultado sobre nome-aluno.
- (d) Seleção de linhas em cursos em que nome-curso = “Computação”, junção com alunos, projeção do resultado sobre nome-aluno.
- (e) Seleção de linhas em cursos em que nome-curso = “Computação”, projeção do resultado sobre nome-aluno.

17. (POSCOMP 2009) Com relação às operações da álgebra relacional está **ERRADO** afirmar que o comando:

- A) SELECT extrai tuplas específicas de uma relação específica.
- B) UNION constrói uma relação consistindo em todas as tuplas que aparecem em um par de relações específicas que são compatíveis.
- C) PROJECT extrai atributos específicos de uma relação específica.
- D) JOIN constrói uma relação a partir de duas relações específicas, consistindo em todas as possibilidades de pares de tuplas, uma de cada uma das relações específicas.
- E) DIFFERENCE constrói uma relação a partir de duas relações específicas que são compatíveis, consistindo em todas as tuplas que aparecem na primeira relação e não aparecem na segunda.

18. (POSCOMP 2009) Sejam as seguintes tabelas em um banco de dados relacional:

COMPRADORES			
CID	CNOME	CIDADE	DESCONTO
C001	Lojas Cacique	Rio de Janeiro	10,00
C002	Lojas Livres	São Paulo	12,00
C003	Mercado Fácil	Curitiba	8,00
C004	Papelaria Simão	Recife	6,00
C005	Lojas da Silva	Manaus	0,00

PRODUTO				
PID	NOME	CLIENTE	QUANT	PRECO
p01	Pente	C001	11000	10
p02	Escova	C002	20000	10
p03	Barbeador	C003	15000	20
p04	Caneta	C003	20000	1
p05	Lápis	C004	10000	1
p06	Caderno	C004	14000	5
p07	Bloco	C005	5000	1,5

Qual o resultado da seguinte consulta em SQL:

```
SELECT CNAME, NOME, PRECO*(1-DESCONTO/100) AS PF
FROM COMPRADORES, PRODUTO
WHERE
DESCONTO > (SELECT AVG(DESCONTO) FROM COMPRADORES)
AND CID=CLIENTE
ORDER BY NOME,CNAME;
```

A)

CNAME	NOME	PF
Lojas Cacique	Pente	9
Lojas Livres	Escova	8,8
Mercado Fácil	Barbeador	18,4
Mercado Fácil	Caneta	0,92

B)

CNAME	NOME	PF
Lojas Cacique	Pente	9
Lojas Livres	Escova	8,8
Mercado Fácil	Barbeador	18,4
Mercado Fácil	Caneta	0,92
Papelaria Simão	Lápis	0,94
Papelaria Simão	Caderno	4,7
Lojas da Silva	Bloco	1,5

C)

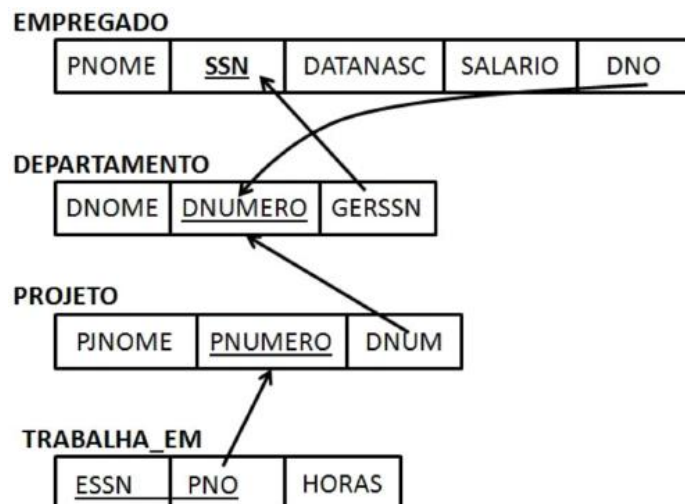
CNAME	NOME	PF
Mercado Fácil	Barbeador	18,4
Mercado Fácil	Caneta	0,92
Lojas Livres	Escova	8,8
Lojas Cacique	Pente	9

D)

CNAME	NOME	PF
Mercado Fácil	Barbeador	20
Mercado Fácil	Caneta	1
Lojas Livres	Escova	10
Lojas Cacique	Pente	10

E) Nenhuma das respostas anteriores.

19. (POSCOMP 2010) Considere o esquema de um banco de dados relacional EMPRESA a seguir. As setas indicam as chaves estrangeiras e o sublinhado indica a chave primária.



Considere a especificação dos seguintes comandos em SQL aplicados sobre o esquema de banco de dados apresentado.

I.



```

SELECT DNUMERO,COUNT(*)
FROM DEPARTAMENTO, EMPREGADO
WHERE DNUMERO=DNO AND SALARIO>40000 AND
DNO IN (SELECT DNO
FROM EMPREGADO
GROUP BY DNO
HAVING COUNT(*)>5)
GROUP BY DNUMERO;

```

II.

```

SELECT DNO, COUNT(*), AVG(SALARIO)
FROM EMPREGADO
GROUP BY DNO;

```

Assinale a alternativa correta.

- a) Na consulta do item I, para cada cinco ou mais departamentos, a consulta retorna o número do departamento e o número dos empregados que recebem mais de 40 mil reais. Na consulta do item II, para cada empregado, a consulta retorna o número do seu departamento e a média de salários de cada departamento.
- b) Na consulta do item I, para cada departamento que tenha mais de cinco empregados, a consulta retorna o número do departamento que possui empregados que recebem mais de 40 mil reais. Na consulta do item II, para cada empregado, a consulta retorna o número do seu departamento, o número de empregados que nele trabalham e o somatório de seus salários.
- c) Na consulta do item I, para cada departamento que tenha mais de cinco empregados, a consulta retorna o número dos empregados que recebem mais de 40 mil reais. Na consulta do item II, para cada empregado, a consulta retorna o número do seu departamento, o número de empregados que trabalham com ele e o somatório de seus salários.
- d) Na consulta do item I, para cada cinco ou mais empregados, a consulta retorna o número do departamento que possui empregados que recebem mais de 40 mil reais. Na consulta do item II, para cada departamento, a consulta retorna o número do seu departamento, o número de empregados que nele trabalham e o somatório de seus salários.
- e) Na consulta do item I, para cada departamento que tenha mais de cinco empregados, a consulta retorna o número do departamento e o número dos empregados que recebem mais de 40 mil reais. Na consulta do item II, para cada departamento, a consulta retorna o número do departamento, o número de empregados que nele trabalham e a média de seus salários.

20. (POSCOMP 2010) Considere um banco de dados relacional composto pelas tabelas definidas a seguir, na linguagem SQL, em que são cadastrados alunos, disciplinas e as matrículas de alunos em disciplinas.

```

CREATE TABLE Aluno (
RA INTEGER NOT NULL,
Nome VARCHAR2(20) NOT NULL,
Curso INTEGER NOT NULL,
PRIMARY KEY (RA) );

```

```
CREATE TABLE Disciplina (  
Codigo INTEGER NOT NULL,  
Nome VARCHAR2(20) NOT NULL,  
Departamento INTEGER NOT NULL,  
PRIMARY KEY (Codigo) );
```

```
CREATE TABLE Matricula (  
RA INTEGER NOT NULL,  
Codigo INTEGER NOT NULL,  
Ano INTEGER NOT NULL,  
PRIMARY KEY (RA, Codigo),  
FOREIGN KEY (Codigo) REFERENCES Disciplina,  
FOREIGN KEY (RA) REFERENCES Aluno);
```

Considere também a consulta a seguir, expressa na linguagem SQL, em que o operador MINUS realiza a operação de subtração de relações.

```
SELECT Nome FROM Aluno A  
WHERE NOT EXISTS ( (SELECT D.Codigo FROM Disciplina D WHERE  
D.Departamento = 5)  
MINUS  
(SELECT M.Codigo FROM Matricula M WHERE M.RA = A.RA)  
)  
AND EXISTS (SELECT D.Codigo FROM Disciplina D WHERE D.Departamento = 5)
```

Considere, por fim, as afirmativas a seguir, sobre a consulta apresentada.

- I. A consulta retorna os nomes dos alunos matriculados em todas as disciplinas do departamento 5.
- II. A consulta retorna os nomes dos alunos matriculados em, pelo menos, uma disciplina do departamento 5.
- III. A consulta retorna os nomes das disciplinas do departamento 5 nas quais todos os alunos estão matriculados.
- IV. A estrutura dessa consulta permite implementar, na linguagem SQL, a operação de divisão da Álgebra Relacional.

Assinale a alternativa correta.

- a) Somente as afirmativas I e II são corretas.
- b) Somente as afirmativas I e IV são corretas.
- c) Somente as afirmativas III e IV são corretas.
- d) Somente as afirmativas I, II e III são corretas.
- e) Somente as afirmativas II, III e IV são corretas.

21. (POSCOMP 2011) Considere a relação a seguir, definida na linguagem SQL padrão.

```
CREATE TABLE EMPREGADO  
(  
CODIGO NUMBER(4) PRIMARY KEY,
```

```
NOME VARCHAR2(10),  
SALARIO NUMBER(7,2)  
)
```

Considere também as consultas (C1, C2, C3 e C4) a seguir, expressas na linguagem SQL.

```
C1:  
select NOME from EMPREGADO  
where CODIGO in ((select CODIGO from EMPREGADO)  
minus  
(select E1.CODIGO from EMPREGADO E1, EMPREGADO E2  
where E1.SALARIO < E2.SALARIO)  
)
```

Obs: o operador minus realiza a operação de subtração entre relações.

```
C2:  
select NOME from EMPREGADO  
where SALARIO = (select max(SALARIO) from EMPREGADO)
```

```
C3:  
Select NOME from EMPREGADO  
where SALARIO >= all (select SALARIO from EMPREGADO)
```

```
C4:  
select NOME from EMPREGADO  
where CODIGO in ( select E1.CODIGO from EMPREGADO E1, EMPREGADO E2  
where E1.SALARIO > E2.SALARIO  
)
```

Com relação às consultas, assinale a alternativa correta.

- a) Apenas as consultas C2 e C3 são equivalentes.
- b) Todas as consultas são equivalentes.
- c) Apenas as consultas C1 e C3 são equivalentes.
- d) Apenas as consultas C1 e C4 são equivalentes.
- e) Apenas as consultas C1, C2 e C3 são equivalentes.

22. (POSCOMP 2012) Considere as tabelas, a seguir, criadas em um banco de dados relacional através da linguagem SQL.

```
CREATE TABLE Empregado  
( ecod int PRIMARY KEY,  
  nome varchar (32),  
  salario number (7,2),  
  dcod int FOREIGN KEY REFERENCES Departamento (dcod));
```

```
CREATE TABLE Departamento  
( dcod int PRIMARY KEY,  
  dnome varchar (12),
```

chefe int FOREIGN KEY REFERENCES Empregado (ecod));

Sejam as consultas (C1, C2 e C3) também em SQL, a seguir.

C1. SELECT nome, salario FROM Empregado E, Departamento D WHERE E.dcod = D.dcod AND E.ecod = D.chefe;

C2. SELECT nome, salario FROM Empregado as E INNER JOIN Departamento as D ON E.dcod=D.dcod WHERE E.ecod = D.chefe;

C3. SELECT nome, salario FROM E.ecod = D.chefe;

Com relação às consultas, assinale a alternativa correta.

- a) Apenas a consulta C1 retorna o nome e o salário dos chefes dos departamentos.
- b) Apenas a consulta C2 retorna o nome e o salário dos chefes dos departamentos.
- c) Apenas a consulta C3 retorna o nome e o salário dos chefes dos departamentos.
- d) As consultas C1, C2 e C3 são equivalentes e retornam o nome e o salário dos chefes dos departamentos.
- e) As consultas C1 e C2 são equivalentes e retornam o nome e o salário dos chefes dos departamentos.

23.(POSCOMP 2013) Uma empresa de auditoria foi contratada para analisar o banco de dados do SUS (Sistema Único de Saúde). A primeira tarefa é encontrar os pares de médicos cadastrados que possuem o mesmo nome (homônimos) e números diferentes no CRM (Conselho Regional de Medicina) para verificar possíveis fraudes. Considere que a tabela que armazena os médicos possui o cadastro no CRM como chave primária e as seguintes colunas: nome, endereço, telefone, especialidade, data de ingresso.

Assinale a alternativa que apresenta, corretamente, a consulta em SQL que busca os resultados ordenados pelo nome dos médicos.

- a) SELECT M1.nome, M1.crm, M2.crm FROM Medicos as M1, Medicos as M2 WHERE M1.nome = M2.nome AND crm1 > crm2 GROUP BY nome;
  - b) SELECT M1.nome, M1.crm, M2.crm FROM Medicos M1 JOIN Medicos M2 ON M1.nome = M2.nome WHERE M1.crm > M2.crm ORDER BY M1.nome;
  - c) SELECT M1.nome, M1.crm, M2.crm FROM Medicos M1 WHERE nome IN (SELECT nome FROM Medicos M2 WHERE M1.nome = nome AND M1.crm > crm) ORDER BY nome;
  - d) SELECT nome, M1.crm, M2.crm FROM Medicos M1 NATURAL JOIN Medicos M2 WHERE M1.crm > M2.crm ORDER BY nome;
- b

e) SELECT \* FROM Medicos as M1, Medicos as M2  
WHERE M1.nome LIKE M2.nome AND crm1 > crm2  
GROUP BY M1.nome;

24. (POSCOMP 2014) Sobre SQL e seus tipos, atribua V (verdadeiro) ou F (falso) às afirmativas a seguir.

( ) No comando SQL: CREATE TABLE EMPRESA.FUNCIONÁRIO ...; , EMPRESA corresponde ao nome do esquema em que o FUNCIONÁRIO será conectado explicitamente no banco de dados.

( ) As relações declaradas por meio das instruções: CREATE VIEW ... são chamadas tabelas de base ou relações de base, nas quais a relação e suas tuplas são realmente criadas e armazenadas como um arquivo pelo SGBD.

( ) O comando: CREATE DOMAIN TIPO\_CPF AS CHAR(11); possibilita que TIPO\_CPF seja usado como uma especificação de atributo para facilitar, por exemplo, a alteração de um tipo de dado para um domínio, que seja usado por diversos atributos em um esquema.

( ) A cláusula UNIQUE especifica chaves alternativas (secundárias), mas também pode ser especificada diretamente para uma chave secundária, se esta for um único atributo, como em Dnome VARCHAR(15) UNIQUE.

( ) O tipo de dado de atributo em SQL chamado BINARY LARGE OBJECT – BLOB é um tipo de dado de cadeia de caracteres de tamanho variável, disponível para especificar colunas que possuem grandes valores de texto, como documentos.

Assinale a alternativa que contém, de cima para baixo, a sequência correta.

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

25. (POSCOMP 2015) Considere o esquema de banco de dados relacional para uma clínica médica, em que as chaves primárias estão sublinhadas:

PACIENTE (CPF, Nome, Sexo, DataDeNascimento);  
MEDICO (CRM, Nome, Sexo);  
CONSULTA (CPF, DataHora, CRM, Sala);  
MEDICAMENTO (Codigo, Nome, PrincipioAtivo);  
PRESCRICAO (CPF, DataHora, Codigo, Posologia).

Os atributos CPF em CONSULTA, CRM em CONSULTA, (CPF, DataHora) em PRESCRICAO e Codigo em PRESCRICAO são chaves estrangeiras que referenciam, respectivamente, PACIENTE, MEDICO, CONSULTA e MEDICAMENTO.

A expressão SQL pertinente à consulta “qual o nome dos medicamentos prescritos mais de uma vez, por um particular médico para um mesmo paciente, restrito às consultas em que médico e paciente possuem o mesmo nome?” é:

(A) SELECT DISTINCT X.NOME FROM MEDICAMENTO X WHERE 2 < (  
SELECT COUNT(\*) FROM PACIENTE V JOIN

MEDICO W JOIN CONSULTA Y JOIN PRESCRICAO Z ON V.CPF = Y.CPF AND  
W.CRM = Y.CRM AND Z.CPF = Y.CPF  
AND Z.DATAHORA = Y.DATAHORA WHERE Z.CODIGO = X.CODIGO AND  
V.NOME = W.NOME )

(B) SELECT DISTINCT X.NOME FROM PACIENTE V JOIN MEDICO W JOIN  
MEDICAMENTO X JOIN CONSULTA Y  
JOIN PRESCRICAO Z ON V.CPF = Y.CPF AND W.CRM = Y.CRM AND Z.CPF =  
Y.CPF AND Z.DATAHORA = Y.DATAHORA  
AND Z.CODIGO = X.CODIGO WHERE V.NOME = W.NOME GROUP BY Y.CPF,  
Y.CRM, X.CODIGO, X.NOME

(C) SELECT DISTINCT X.NOME FROM MEDICAMENTO X WHERE 2 > (  
SELECT COUNT(\*) FROM PACIENTE V JOIN  
MEDICO W JOIN CONSULTA Y JOIN PRESCRICAO Z ON V.CPF = Y.CPF AND  
W.CRM = Y.CRM AND Z.CPF = Y.CPF  
AND Z.DATAHORA = Y.DATAHORA WHERE Z.CODIGO = X.CODIGO AND  
V.NOME = W.NOME )

(D) SELECT DISTINCT X.NOME FROM PACIENTE V JOIN MEDICO W JOIN  
MEDICAMENTO X JOIN CONSULTA Y  
JOIN PRESCRICAO Z ON V.CPF = Y.CPF AND W.CRM = Y.CRM AND Z.CPF =  
Y.CPF AND Z.DATAHORA = Y.DATAHORA  
AND Z.CODIGO = X.CODIGO WHERE V.NOME = W.NOME GROUP BY Y.CPF,  
Y.CRM, X.CODIGO, X.NOME HAVING  
COUNT(\*) > 1

(E) SELECT DISTINCT X.NOME FROM PACIENTE V NATURAL JOIN MEDICO  
W NATURAL JOIN MEDICAMENTO X  
NATURAL JOIN CONSULTA Y NATURAL JOIN PRESCRICAO Z WHERE  
V.NOME = W.NOME GROUP BY X.CODIGO,  
X.NOME HAVING COUNT(\*) > 1

26. (POSCOMP 2017) Em consultas escritas em SQL, quando há pelo menos um NULL no predicado da cláusula WHERE, o resultado da avaliação é “desconhecido” (exceto quando são explicitamente empregados IS NULL ou IS NOT NULL); por exemplo, o resultado da avaliação de  $3 + \text{NULL} > 7$  é “desconhecido”. Portanto, “verdadeiro”, “falso” e “desconhecido” são os resultados possíveis na avaliação de predicados da cláusula WHERE. A regra geral é que são selecionadas apenas as combinações de *tuplas* em que o predicado é avaliado como “verdadeiro”. Seja a relação R que possui quatro *tuplas* – (12, 15, 5100), (13, NULL, 3500), (14, NULL, NULL) e (15, 12, NULL) – em que o primeiro, o segundo e o terceiro valores em cada *tupla* referem-se aos atributos at1, at2 e at3, respectivamente. Os comandos a seguir representam consultas sobre R:

(C1) select \* from R  
where (at1 >= 12) AND (at2 > 14)  
(C2) select \* from R  
where (at2 > 12) OR (at3 > 3000)  
(C3) select \* from R

where (NOT (at1<at2))

A quantidade de *tuplas* retornadas pelas execuções dos comandos (C1), (C2) e (C3), respectivamente, é:

- A) dois, um e dois.
- B) dois, dois e um.
- C) um, um e dois.
- D) um, dois e um.
- E) dois, dois, dois.

27. (POSCOMP 2018) Quando um predicado da cláusula WHERE de uma consulta SQL aninhada referencia algum atributo de uma relação presente na consulta externa, tais consultas são ditas correlacionadas (ou correlatas). Seja Q uma consulta definida por

```
SELECT DISTINCT SALARIO FROM  
EMPRESA WHERE SALARIO > ALL (SELECT Salario FROM EMPRESA) OR  
SALARIO < ANY (SELECT  
Salario FROM EMPRESA).
```

A expressão SQL que utiliza consultas correlacionadas e cujo resultado é o mesmo resultado de Q é:

- A) SELECT DISTINCT Salario FROM EMPRESA AS E WHERE EXISTS (SELECT \* FROM EMPRESA AS I WHERE I.Salario = E.Salario)
- B) SELECT DISTINCT Salario FROM EMPRESA AS E WHERE EXISTS (SELECT \* FROM EMPRESA AS I WHERE I.Salario >= E.Salario)
- C) SELECT DISTINCT Salario FROM EMPRESA AS E WHERE EXISTS (SELECT \* FROM EMPRESA AS I WHERE I.Salario <= E.Salario)
- D) SELECT DISTINCT Salario FROM EMPRESA AS E WHERE EXISTS (SELECT \* FROM EMPRESA AS I WHERE I.Salario > E.Salario)
- E) SELECT DISTINCT Salario FROM EMPRESA AS E WHERE EXISTS (SELECT \* FROM EMPRESA AS I WHERE I.Salario < E.Salario)

REFERENCIAS:

<http://portal.inep.gov.br/web/guest/educacao-superior/enade/provas-e-gabaritos>  
<https://www.sbc.org.br/documentos-da-sbc/category/153-provas-e-gabaritos-do-poscomp>