

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS (INPE)

Concurso Público - NÍVEL SUPERIOR

CARGO: Tecnologista da Carreira de Desenvolvimento Tecnológico

Classe: Tecnologista Junior Padrão I

(TS09)

CADERNO DE PROVAS

PROVA DISCURSIVA

TEMA:

Discorra sobre a utilização de métodos ágeis para desenvolvimento de software. Em seu texto, inclua, necessariamente: características do desenvolvimento ágil, aplicabilidade dos métodos ágeis, vantagens e desvantagens em relação aos métodos tradicionais.

PROVA OBJETIVA DE CONHECIMENTOS ESPECÍFICOS

Questão 1: Como é possível na linguagem Java forçar a execução imediata do coletor de lixo (garbage collector) para remover um objeto da memória?

- a) () Setar todas as referências do objeto com novos valores, como por exemplo *null*.
- b) () Chamar `System.gc()`
- c) () Chamar `System.gc()` passando como parâmetro a referência do objeto a ser coletado.
- d) () Chamar `Runtime.gc()`
- e) () O coletor de lixo não pode ser forçado a ser executado imediatamente

Questão 2: Dado o seguinte código Java:

```
import java.text.SimpleDateFormat;
import java.util.GregorianCalendar;
```

```
public class Data extends GregorianCalendar{
    private static final long serialVersionUID =
5848695528815211835L;
    final SimpleDateFormat df = new
SimpleDateFormat("dd-MM-yyyy");

    public Data(int i, int j, int k) { super(i,j-1,k); }

    public String toString(){ return df.format(getTime()); }
    public static void call(Data data1, Data data2) {
        data2.set(Data.YEAR, 2013);
        data1 = data2;
    }
    public static void print(Data data1, Data data2){
        System.out.println("data1 é " + data1 + "\ndata2 é " +
data2);
    }
    public static void main(String args[]) {
        Data data1 = new Data(2011, 11, 11);
        Data data2 = new Data(2011, 12, 11);
        call (data1, data2);
        print(data1, data2);
    }
}
```

É correto dizer que:

- a) () Quando compilado não existe erro e quando executado temos como saída:
data1 é 11-12-2011
data2 é 11-01-2014
- b) () Quando compilado não existe erro e quando executado temos como saída:
data1 é 11-01-2014
data2 é 11-12-2011
- c) () Quando compilado não existe erro e quando executado temos como saída:
data1 é 11-12-2013
data2 é 11-11-2011
- d) () Quando compilado não existe erro e quando executado temos como saída:
data1 é 11-11-2011
data2 é 11-12-2013
- e) () Quando compilado ocorre erro pois a expressão `System.out.println("data1 é " + data1 + "\ndata2 é " + data2);` é rejeitada pelo compilador porque a classe `Data` não pode ser concatenada com a classe `String`.

Questão 3: Dado o seguinte código Java, no qual a numeração das linhas é mostrada nos comentários:

```
import java.io.IOException;
public class Math {
    public float sum(float a, float b) throws IOException {
        return a + b;
    }
}
class SimpleMath extends Math {
    /*I*/ float sum(float a, float b) {return a + b;}
    /*II*/public int sum(int a, int b) throws Exception { return
a + b;}
    /*III*/public float sum(float a, float b) throws Exception {
return a + b; }
    /*
```

```

IV public float sum(float p, float q) { return p + q;
*/
}

```

Teremos erro de compilação na classe SimpleMath na(s) linha(s):

- a) I e II
- b) I, II e III
- c) II e IV
- d) II, III e IV
- e) Nenhuma das alternativas acima

Questão 4: Dado o seguinte trecho de código Java:

```

boolean b = 10 % 2 == 5;
System.out.print(new Boolean("false"));
System.out.print(new Boolean("T"));
System.out.print(new Boolean(b));
System.out.print(new Boolean("True"));
System.out.print(new Boolean("no"));
System.out.print(new Boolean("yes"));

```

Ao executá-lo teremos como saída :

- a) falsefalsefalsefalsefalse
- b) falsefalsefalsefalsefalse
- c) falsefalsetruefalsefalse
- d) falsefalsefalsefalsefalse
- e) Nenhuma das alternativas acima

Questão 5: Dado o seguinte código Java:

```

class MyThread extends Thread {
    public MyThread(String name, int priority){
        super(name);
        this.setPriority(priority);
    }
    public void run() {
        System.out.print(getName());
    }
    public static void main(String... args) {
        Thread t0 = new MyThread("t0",0);
        Thread t1 = new MyThread("t1",5);
        t0.start();
        t1.start();
    }
}

```

Ao executá-lo espera-se a seguinte saída:

- a) t0t1
- b) t1t0
- c) Erro em tempo de compilação
- d) Erro em tempo de execução
- e) Nenhuma das alternativas acima

Questão 6: Dado o seguinte script groovy, no qual a numeração das linhas é mostrada nos comentários:

```

/*1*/def list = ['a','b','c','d','e','f','g']
/*2*/list[2..3] = 'z'
/*3*/assert list == ['a', 'b', 'z', 'e', 'f', 'g']

```

```

/*4*/list[4..4]= ['u','v']
/*5*/assert list == ['a', 'b', 'z', 'e', 'u', 'v', 'g']
/*6*/list[0..1]= []
/*7*/assert list == ['z', 'e', 'u', 'v', 'g']
/*8*/list[1..1]= []
/*9*/assert list == ['z', 'u', 'v', 'g']

```

Podemos afirmar que ao executar o script:

- a) Teremos falha no assert da linha 3
- b) Teremos falha no assert da linha 5
- c) Teremos falha no assert da linha 7
- d) Teremos falha no assert da linha 9
- e) Não teremos falhas

Questão 7: Dado o seguinte script groovy, no qual a numeração das linhas é mostrada nos comentários:

```

/*1*/def list = Collections.singletonList('a')
/*2*/list << 'a'
/*3*/assert list == ['a', 'a']
/*4*/list << 'b'
/*5*/assert list == ['a','a','b']
/*6*/list << 'c'
/*7*/assert list == ['a','a','b','c']
/*8*/list = list - 'a'
/*9*/assert list == ['a','b','c']

```

Podemos afirmar que ao executar o script:

- a) Teremos exceção na linha 1
- b) Teremos exceção na linha 2
- c) Teremos falha no assert da linha 3
- d) Teremos falha no assert da linha 9
- e) Não teremos nenhuma exceção ou falha nos asserts

Questão 8: Dadas as seguintes sentenças sobre a linguagem Groovy:

- I. Uma closure sempre retorna um valor.
- II. Uma closure é como um bloco de código regular que é executado no momento em que é encontrado.
- III. O trecho de código Groovy a seguir não é compilável

```
def x=false; if(x=true) println "$x"
```

IV. O código abaixo produz na saída padrão: foo bar

```

def foo = { bar, foo-> println "${bar} ${foo}" }
def bar = foo.curry("foo")
bar("bar")

```

É correto dizer que:

- a) Somente a I é verdadeira
- b) Somente II e IV são falsas
- c) Todas são verdadeiras
- d) Todas são falsas

e) Nenhuma das anteriores

Questão 9: Dadas as seguintes sentenças sobre o framework Grails:

- I. Como novidades na versão 2.1 do Grails temos, entre outras coisas, o plugin “cache” por padrão, a criação de alias para comandos e o uso do banco de dados H2 no lugar do HSQLDB;
- II. Na versão 2.1 do Grails, conf, filters, taglib, services, domain e controllers são nomes de artefatos padrão usados na configuração do Logging.
- III. Grails oferece suporte ao conceito de configuração por ambiente.
- IV. Grails oferece suporte à migração de banco de dados ao estilo Rails através do plugin “Database Migration”.

É correto dizer que:

- a) Somente a IV é falsa
- b) Somente a I é falsa
- c) Somente III e IV são falsas
- d) Todas são falsas
- e) Nenhuma das alternativas anteriores

Questão 10: No framework Grails as palavras chave belongsTo, hasOne, hasMany correspondem respectivamente a relacionamentos do tipo:

- a) one-to-one, many-to-one, one-to-many
- b) one-to-one, one-to-one, many-to-many
- c) one-to-many, many-to-one, many-to-one
- d) many-to-one, one-to-one, one-to-many
- e) many-to-many, one-to-many, one-to-many

Questão 11: Dadas as seguintes classes de domínio no framework Grails, ambas definidas no arquivo “Person.groovy” é correto afirmar que:

```
class Person {
    Address homeAddress
    Address workAddress
    static embedded = ['homeAddress', 'workAddress']
}
class Address{
    String number
    String code
}
```

- a) Este código terá como resultado duas tabelas no banco de dados, uma para a classe Person e outra para a classe Address.
- b) A palavra "embedded" permite a implementação da noção de composição, isto é, faz com que este código resulte em uma única tabela no banco de dados.

c) Um dos resultados deste código é uma tabela no banco de dados chamada 'Person', da qual uma das colunas terá o nome de 'homeAdress'.

d) Através do GORM não é possível escolher um nome diferente para a tabela que persistirá a classe Person, embora isso seja possível através do Hibernate.

e) Há erro de compilação neste código, pois as linhas não foram finalizadas com ponto e vírgula (;).

Questão 12: Dadas as seguintes sentenças sobre a implementação da persistência das classes no Grails, podemos afirmar que a sentença correta é:

- a) Caso não seja especificado como deve ser implementado no banco de dados o relacionamento de extensão, ou herança, entre duas classes, o Grails utilizará uma tabela para cada classe.
- b) "Table-per-hierarchy" e "table-per-subclass" são formas distintas de mapear relacionamentos do tipo "one-to-many" e "many-to-one" entre classes.
- c) Por default, quando "hasMany" é utilizado para definir um relacionamento, o Grails cria uma propriedade do tipo "java.util.set", o que garante a exclusividade, mas não a ordem dos valores.
- d) Um dos problemas que impedem a utilização do Grails quando já existe um banco de dados criado é que o GORM não permite customizar o nome de tabelas e campos para que correspondam aos já existentes no banco de dados.
- e) Uma questão a ser melhorada nas próximas versões do Grails é possibilitar que operações de "update" e "delete" em cascata sejam possíveis através do GORM.

Questão 13: Dadas as seguintes sentenças sobre a abordagem da Orientação a Objeto:

- I. Uma classe pode conter um conjunto de operações e itens de dados combinados em uma simples entidade.
- II. Um objeto pode ser definido como uma instância de uma classe.
- III. Superclasse é uma especialização de um conjunto de classes relacionadas a ela.
- IV. Herança é a propriedade que uma classe tem de se basear em operações e atributos de uma outra classe existente e nunca nela mesma.

É correto dizer que:

- a) Somente I, II e IV são verdadeiras
- b) Somente I e II são verdadeiras
- c) Somente III e IV são verdadeiras
- d) Todas são verdadeiras
- e) Nenhuma das alternativas anteriores

Questão 14: Dadas as seguintes sentenças sobre a abordagem da Orientação a Objeto:

- I. O reuso do código não pode ser considerado como um benefício oriundo da herança.
- II. Como exemplos de herança podemos citar: classe APARTAMENTO e classe MORADIA, como também classe FOLHA e classe ÁRVORE
- III. Podemos dizer que acontece sobrecarga de operação quando são definidas duas ou mais operações com o mesmo nome, mas com diferentes listas de parâmetros.
- IV. Membros públicos de uma dada classe são vistos por todas as operações desta classe e também de outras, diferente dos membros protegidos que são vistos somente pelos métodos desta classe.

É correto dizer que:

- a) Somente I, II e IV são falsas
- b) Somente I e III são falsas
- c) Somente II e IV são falsas
- d) Todas são falsas
- e) Nenhuma das anteriores

Questão 15: Dadas as seguintes sentenças sobre a abordagem da Orientação a Objeto:

- I. Herança simples ocorre quando a classe herda somente de uma outra classe, nunca de duas ou mais, ou seja, a classe tem somente um pai. Isso é possível na linguagem Java.
- II. Herança múltipla ocorre quando a classe herda de mais de uma classe, ou seja, a classe possui mais de um pai. Isso é possível na linguagem Java.
- III. Classes abstratas não podem ser instanciadas.
- IV. Na linguagem Java as classes abstratas só podem possuir métodos do tipo abstratos.

É correto dizer que:

- a) Somente I, II e III são falsas
- b) Somente a II é falsa
- c) Somente a IV é falsa
- d) Somente II e IV são falsas
- e) Nenhuma das alternativas acima

Questão 16: Dadas as seguintes sentenças:

- I. Sobrecarga de operador é a técnica utilizada na orientação a objetos para se alterar o comportamento de operadores como '+', '-', '*' e '/'
- II. A linguagem Java não permite sobrecarga de operadores.
- III. A palavra reservada 'final' na declaração de uma classe Java impede que ela possua classes que a herdam.
- IV. Na linguagem Java uma classe filha pode alterar o comportamento dos métodos herdados

de sua classe pai através da sobrescrita de métodos.

É correto dizer que:

- a) Somente I e IV são verdadeiras
- b) Somente II e III são verdadeiras
- c) Somente I, II e III são verdadeiras
- d) Todas são verdadeiras
- e) Nenhuma das alternativas anteriores

Questão 17: Dadas as seguintes sentenças sobre programação Orientada a Objeto:

- I. Encapsulamento é uma técnica para esconder o detalhamento de implementação das operações de uma classe.
- II. Uma vez a classe encapsulada, podemos tratá-la com uma caixa preta.
- III. Na implementação de uma operação de classe são definidos os seus detalhes internos.
- IV. Quando usada corretamente, a técnica de encapsulamento nos permite a criação de componentes plugáveis – pluggable.

É correto dizer que:

- a) Somente I, II e IV são verdadeiras
- b) Somente I e II são verdadeiras
- c) Somente III e IV são verdadeiras
- d) Todas são verdadeiras
- e) Nenhuma das alternativas anteriores

Questão 18: Dadas as seguintes sentenças sobre a linguagem Groovy:

- I. Arquivos “.groovy” compilados tornam-se “.class”.
- II. Métodos e classes são *public* por default, assim é opcional o uso dos modificadores.
- III. Pode ser omitido o tipo das variáveis e dos atributos, assim como o tipo de retorno dos métodos.
- IV. Comando de return é opcional nos métodos, assim como parênteses em chamadas de métodos com um parâmetro.

É correto dizer que:

- a) Somente I, II e III são verdadeiras
- b) Somente I e II são verdadeiras
- c) Todas são verdadeiras
- d) Somente III e IV são verdadeiras
- e) Nenhuma das alternativas anteriores

Questão 19: Com relação à UML é correto afirmar:

- a) É um método para especificação, visualização, documentação de modelos de sistemas de software.
- b) O diagrama de objetos é um de seus diagramas estruturais e representa a instância de um diagrama de classes num determinado ponto do tempo.

- c) () O diagrama de seqüências é um de seus diagramas estruturais e mostra um conjunto de mensagens trocadas entre objetos e a ordem em que essas mensagens acontecem.
- d) () Valores de atributos não podem ser colocados em um diagrama de objetos.
- e) () O diagrama de fluxo de dados é um de seus diagramas de comportamento e mostra objetos, seus relacionamentos e o fluxo de dados entre eles.

Questão 20: Considere as seguintes sentenças sobre o diagrama de classes da UML:

- I. Atributos e operações podem ter sua visibilidade anotada na representação da classe.
- II. Em um dos compartimentos da classe é possível especificar o nome, o tipo e o valor inicial dos atributos no formato: nome: tipo = valor
- III. O relacionamento de dependência não se aplica a este diagrama.
- IV. O relacionamento de composição não se aplica a este diagrama.
- V. O relacionamento, do tipo agregação, de uma classe com outra classe contida em um outro pacote (*package*) pode ser representada neste tipo de diagrama.

É correto afirmar que:

- a) () Todas são verdadeiras.
- b) () Somente II e III são verdadeiras.
- c) () Nenhuma é verdadeira.
- d) () Somente I e II são verdadeiras.
- e) () Somente I, II e V são verdadeiras.

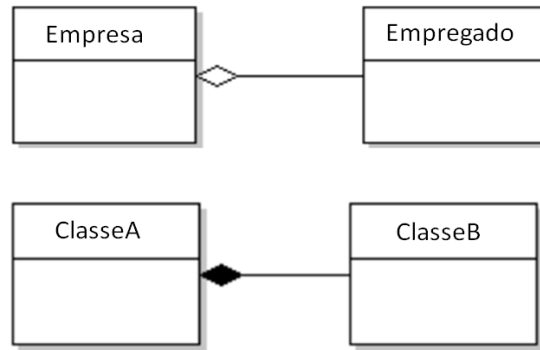
Questão 21: Dadas as seguintes sentenças sobre o diagrama de seqüências da UML:

- I. Neste diagrama é possível representar a destruição de um objeto.
- II. Este diagrama pode conter relacionamentos de agregação.
- III. Este diagrama representa a interação entre objetos de diferentes classes ao longo do tempo e pode estar associado a um caso de uso específico. Por este motivo, pode ser classificado como um diagrama estrutural e temporal.
- IV. Neste diagrama é possível representar o período de tempo durante o qual uma determinada operação de um objeto está sendo executada.

É correto dizer que:

- a) () Todas são verdadeiras.
- b) () Somente I, II e IV são verdadeiras.
- c) () Nenhuma é verdadeira.
- d) () Somente I, III e IV são verdadeiras.
- e) () Nenhuma das alternativas anteriores.

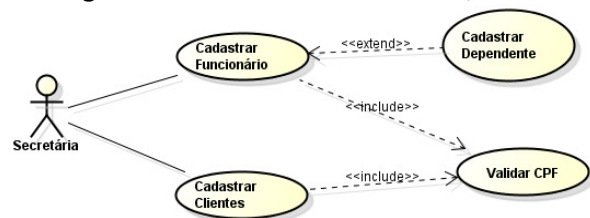
Questão 22: Dado o modelo abaixo, correspondente a um diagrama de classes da UML,



É correto dizer que:

- a) () O relacionamento entre as classes "Empresa" e "Empregado" não parece lógico, pois a leitura que se faz desta representação é: Empregado contém Empresas.
- b) () Existe um relacionamento de agregação entre as classes ClasseA e ClasseB.
- c) () O relacionamento entre as classes ClasseA e ClasseB indica que: ClasseA é parte da ClasseB.
- d) () A existência de um objeto do tipo ClasseB não faz sentido se não existir um objeto do tipo ClasseA.
- e) () Existe um relacionamento de composição entre as classes Empresa e Empregado.

Questão 23: Dado o modelo abaixo, correspondente a um diagrama de Casos de Uso da UML,



É correto dizer que:

- a) () Considerando que ao realizar o cadastro de um funcionário deve-se cadastrar também seus dependentes, quando estes existirem, a seta do relacionamento entre os Casos de Uso "Cadastrar Funcionário" e "Cadastrar Dependente" deveria estar apontando para "Cadastrar Dependente".
- b) () Considerando que ao realizar o cadastro de um funcionário deve-se cadastrar também seus dependentes, quando estes existirem, não é aconselhável a utilização do relacionamento <<extend>> entre os Casos de Uso "Cadastrar Funcionário" e "Cadastrar Dependente", pois, neste caso, faz mais sentido utilizar o relacionamento <<include>>.

- c) () Os Casos de Uso "Cadastrar Funcionário" e "Cadastrar Clientes" compartilham as mesmas funcionalidades descritas no Caso de Uso "Validar CPF".
- d) () Existe um erro nesse diagrama, pois o relacionamento <<include>> aparece duas vezes no Caso de Uso "Validar CPF".
- e) () No Caso de Uso "Cadastrar Dependente" também ocorre a validação do CPF.

Questão 24: Considere as seguintes sentenças relacionadas ao processo de desenvolvimento de software:

- I. Requisitos não podem mudar após a codificação do software.
- II. O estágio de codificação é o processo de conversão de uma especificação do sistema em um sistema executável.
- III. O modelo espiral de Boehm é um modelo de processo que explicitamente reconhece riscos.
- IV. Independentemente de qual seja o modelo de processo adotado, é recomendável que ele possa acomodar mudanças no software em desenvolvimento.
- V. Em um processo de desenvolvimento incremental, partes do sistema de software que já foram codificadas podem ser disponibilizadas ao usuário final antes de se ter o sistema totalmente desenvolvido.
- VI. Em um processo evolucionário/iterativo, há a redução de riscos relacionados ao desenvolvimento do sistema de software, entretanto, este processo requer que especificações formais do sistema estejam previamente disponíveis.

É correto afirmar que:

- a) () Todas as afirmações são verdadeiras.
- b) () Somente as afirmações I e VI são falsas.
- c) () Somente as afirmações II e IV são verdadeiras.
- d) () Somente as afirmações II, III e IV são verdadeiras.
- e) () Somente as afirmações I e V são falsas.

Questão 25: Dadas as seguintes sentenças sobre a abordagem ágil para desenvolvimento de software:

- I. Software codificado e apto a ser posto em funcionamento é uma das principais medidas de progresso.
- II. Simplicidade no desenvolvimento é uma característica recomendada.
- III. Reuniões não são desejáveis.
- IV. Os princípios de desenvolvimento priorizam a entrega mais que análise e projeto.

É correto dizer que:

- a) () Todas são verdadeiras.
- b) () Somente I, II e III são verdadeiras.
- c) () Somente I, II e IV são verdadeiras.
- d) () Somente II, III e IV são verdadeiras.
- e) () Nenhuma das anteriores.

Questão 26: Sobre a abordagem ágil para desenvolvimento de software é correto afirmar:

- a) () Evita o envolvimento do cliente no processo de desenvolvimento, pois muito *feedback* pode atrasar a entrega do software.
- b) () Gasta-se mais tempo em análises de como o sistema deve ser desenvolvido do que no desenvolvimento de programas e testes.
- c) () São exemplos de métodos ágeis para desenvolvimento de software: o Extreme Programming, o RUP e o Scrum.
- d) () Membros da equipe devem desenvolver suas próprias maneiras de trabalhar, sem processos prescritivos.
- e) () Esta abordagem é recomendada principalmente para equipes grandes de desenvolvimento de software.

Questão 27: Sobre a abordagem do RUP para desenvolvimento de software é correto afirmar:

- a) () A visão estática do RUP mostra as atividades que ocorrem durante o processo de desenvolvimento. Dentre essas atividades, chamadas de "workflows", podemos citar o "gerenciamento de configuração e mudanças", o "gerenciamento de projetos" e o "gerenciamento de pessoas".
- b) () A visão dinâmica do RUP mostra as fases do processo ao longo do tempo. Dentre essas fases, podemos citar "Requisitos", "Análise e projeto" e "Implementação".
- c) () Além da visão estática e dinâmica, o RUP ainda apresenta uma perspectiva prática, que sugere boas práticas a serem usadas durante o processo. Dentre elas, podemos citar "gerenciar requisitos" e "modelar software visualmente".
- d) () O RUP adota o paradigma da análise estruturada.
- e) () Uma das vantagens de escolher o RUP como processo de desenvolvimento de software é que ele oferece um guia completo para o desenvolvimento que pode ser usado em qualquer organização sem que seja necessário fazer adaptações.

Questão 28: Sobre a Engenharia de Requisitos para desenvolvimento de software, é correto afirmar que:

- a) () O uso de cenários na atividade de elicitação de requisitos pode ser bastante proveitoso, já que as

peças geralmente acham mais fácil se relacionar com exemplos da vida real do que com descrições abstratas.

- b) () Etnografia é uma técnica para eliciação de requisitos que tem a vantagem de não exigir que o analista de sistemas visite o ambiente de trabalho em que o sistema será utilizado.
- c) () O gerenciamento de mudança de requisitos deve começar após a entrega do software.
- d) () Um protótipo é uma versão rápida e parcial do software a ser entregue. Seu uso deve ser restrito aos desenvolvedores, não é recomendado que ele seja apresentado aos usuários finais.
- e) () A utilização da técnica de observação para levantamento de requisitos exclui a utilização da técnica de entrevistas.

Questão 29: Dadas as seguintes sentenças sobre Banco de Dados Relacional:

- I. PROJEÇÃO, SELEÇÃO e JUNÇÃO são operações da álgebra relacional.
- II. A operação de SELEÇÃO é utilizada para a união de dados de duas Tabelas diferentes em uma só Tabela.
- III. Tuplas ou registros equivalem a linhas de tabelas.
- IV. Se duas Tabelas diferentes possuem uma coluna comum definida, elas poderão ser juntadas através dessas colunas com o uso do operador de PROJEÇÃO.

É correto dizer que:

- a) () Todas são verdadeiras.
- b) () Somente I, II e III são verdadeiras.
- c) () Somente I e III são verdadeiras.
- d) () Somente a III é verdadeira.
- e) () Nenhuma das alternativas anteriores.

Questão 30: Dadas as seguintes sentenças sobre Banco de Dados Relacional:

- I. Existem 4 tipos de Relacionamentos entre tabelas: "um para um"; "muitos para um"; "um para muitos" e "muitos para muitos".
- II. Para implementar um relacionamento do tipo "muitos para muitos" entre duas tabelas é necessário criar uma terceira tabela.
- III. Numa tabela que armazena a lista de profissionais de uma empresa é aconselhável que o nome do profissional seja a Chave Primária.
- IV. Um das vantagens de utilizar uma Chave Estrangeira em um campo que referencia um

campo chave de uma outra tabela é que a Chave Estrangeira ajuda a evitar anomalias de inclusão.

É correto dizer que:

- a) () Todas são verdadeiras.
- b) () Somente I, II e IV são verdadeiras.
- c) () Somente II e IV são verdadeiras.
- d) () Somente I e III são verdadeiras.
- e) () Nenhuma das alternativas anteriores.

Questão 31: Considere que as instruções abaixo foram submetidas ao PostgreSQL e retornaram os resultados apresentados nas figuras abaixo delas (considere que a primeira linha corresponde ao nome e tipo das colunas).

SELECT * from prova.departamento;

id numeric	nome character varying	id_empresa numeric
1	Secretaria	1
3	Engenharia	1
2	RH	1

SELECT * from prova.funcionario;

id numeric	nome character varying	funcao character varying	id_departamento numeric
1	Maria	Secretária	1
2	Helena	Secretária	1
3	Marcela	Psicóloga	2

SELECT * from prova.empresa;

id numeric	cnpj character varying	nome character varying
1		INPE

Dadas as seguintes sentenças:

I. "prova" é o banco de dados no qual foram criadas as tabelas "departamento", "funcionario" e "empresa".

II. Ao submeter a instrução a seguir ao PostgreSQL:

```
SELECT d.id, d.nome, count(f.id)
FROM prova.departamento d
join prova.funcionario f on d.id=f.id_departamento
GROUP BY d.id;
```

Não ocorre erro e o seguinte resultado é retornado:

id numeric	nome character varying	count bigint
1	Secretaria	2
2	RH	1

III. Ao submeter a instrução a seguir ao PostgreSQL:

```
SELECT d.nome
FROM prova.departamento d
```

```
Join prova.funcionario f on d.id=f.id_departamento
WHERE count(f.id) > 1
GROUP BY d.nome
```

Não ocorre erro e o seguinte resultado é retornado:

nome
Secretaria

IV. Existem somente 3 registros na tabela funcionario.

É correto dizer que:

- a) () Todas são verdadeiras.
- b) () Somente II, III e IV são verdadeiras.
- c) () Somente III e IV são verdadeiras.
- d) () Somente I, III e IV são verdadeiras.
- e) () Nenhuma das alternativas anteriores.

Questão 32: Dadas as seguintes sentenças sobre a utilização de bancos de dados no PostgreSQL:

- I. Pode ser necessário inserir uma grande quantidade de dados ao se fazer a carga inicial de um banco de dados. No PostgreSQL é possível utilizar o comando COPY, em vez de usar uma série de comandos INSERT para tornar este processo mais eficiente.
- II. O comando COPY TO copia o conteúdo de uma tabela para um arquivo, enquanto o comando COPY FROM copia dados de um arquivo para uma tabela.
- III. Ao criar uma VIEW a cláusula FROM pode referenciar, além de tabelas, também outras VIEWS.
- IV. Um mesmo SEQUENCE pode ser usado em várias tabelas.

É correto dizer que:

- a) () Todas são verdadeiras.
- b) () Somente I, II e III são verdadeiras.
- c) () Somente III e IV são verdadeiras.
- d) () Somente I, III e IV são verdadeiras.
- e) () Nenhuma das alternativas anteriores.

Questão 33: Dadas as seguintes sentenças sobre a utilização de bancos de dados criados no PostgreSQL:

- I. É possível especificar, através de um Trigger, que uma determinada função seja executada sempre que ocorrer uma operação de INSERT no banco de dados.
- II. Utilizando o comando ALTER TABLE é possível alterar dados cadastrados em uma tabela.

III. É possível apagar um banco de dados estando conectado a um banco de dados diferente do que você está querendo apagar.

IV. O PostgreSQL permite que funções definidas pelo usuário possam ser escritas em outras linguagens além de SQL.

É correto dizer que:

- a) () Todas são verdadeiras.
- b) () Somente I, III e IV são verdadeiras.
- c) () Somente III e IV são verdadeiras.
- d) () Somente II, III e IV são verdadeiras.
- e) () Nenhuma das alternativas anteriores.

Questão 34: Dadas as seguintes sentenças sobre a administração e manutenção de bancos de dados no PostgreSQL:

- I. O administrador de banco de dados pode controlar a organização em disco da instalação do PostgreSQL através do uso de TABLESPACES. Por exemplo, um índice muito utilizado pode ser colocado em um disco muito rápido com alta disponibilidade, ao mesmo tempo, uma tabela armazenando dados raramente utilizados pode ser armazenada em um sistema de disco mais barato e mais lento.
- II. Um dos motivos para utilizar o comando VACUUM é atualizar as estatísticas dos dados utilizados pelo planejador de comandos do PostgreSQL.
- III. Para facilitar o gerenciamento dos objetos de um banco de dados é possível organizá-los em SCHEMAS.
- IV. O "pg_dump" faz backups consistentes do banco de dados mesmo se o banco de dados estiver sendo usado simultaneamente.

É correto dizer que:

- a) () Somente II, III e IV são verdadeiras.
- b) () Somente I, II e III são verdadeiras.
- c) () Somente II e III são verdadeiras.
- d) () Todas são verdadeiras.
- e) () Nenhuma das alternativas anteriores.

Questão 35: Considere que as seguintes instruções foram submetidas para execução no PostgreSQL na ordem em que são apresentadas:

```
CREATE TABLE notas (  
    nota decimal(4,2) CONSTRAINT chknota  
        CHECK (nota BETWEEN 0.00 AND 10.00)  
);  
INSERT INTO notas VALUES(10);  
INSERT INTO notas VALUES(9.2);  
INSERT INTO notas VALUES(9.0);  
INSERT INTO notas VALUES(8.3);
```



```

INSERT INTO notas VALUES(7.7);
INSERT INTO notas VALUES(7.4);
INSERT INTO notas VALUES(6.4);
INSERT INTO notas VALUES(5.8);
INSERT INTO notas VALUES(5.1);
INSERT INTO notas VALUES(5.0);
INSERT INTO notas VALUES(0);
SELECT CASE
    WHEN nota < 3 THEN 'E'
    WHEN nota < 5 THEN 'D'
    WHEN nota < 7 THEN 'C'
    WHEN nota < 9 THEN 'B'
    ELSE 'A'
END AS conceito,
COUNT(*) AS quantidade,
MIN(nota) AS menor,
MAX(nota) AS maior
FROM notas
GROUP BY CASE
    WHEN nota < 3 THEN 'E'
    WHEN nota < 5 THEN 'D'
    WHEN nota < 7 THEN 'C'
    WHEN nota < 9 THEN 'B'
    ELSE 'A'
END
ORDER BY conceito;

```

Dadas as seguintes sentenças:

I. A instrução SELECT gera um erro, pois não é possível utilizar a expressão CASE para agrupar valores.

II. A instrução SELECT retorna o seguinte resultado (considere que a primeira linha corresponde aos nomes e tipos das colunas):

conceito text	quantidade bigint	menor numeric	maior numeric
A	3	9.00	10.00
B	3	7.40	8.30
C	4	5.00	6.40
E	1	0.00	0.00

III. "conceito", "quantidade", "menor" e "maior" são colunas da tabela notas.

IV. Existe uma restrição na tabela notas que não permite que sejam inseridas notas entre 0 e 10.

É correto dizer que:

- a) () Todas são falsas.
- b) () Somente I e IV são verdadeiras.
- c) () Somente a II é verdadeira.
- d) () Somente a IV é verdadeira.
- e) () Nenhuma das alternativas anteriores.