

**C.03/09
TÉCNICO EM COMPUTAÇÃO
COM ÊNFASE NA
TECNOLOGIA ORACLE PL/SQL**

CADERNO	1
GABARITO	1
APLICAÇÃO	MANHÃ

Aplicação: 07/março

LEIA COM ATENÇÃO AS INSTRUÇÕES

- 1 - A duração da prova é de 4 horas, já incluído o tempo de preenchimento do cartão de respostas.
- 2 - O candidato que, na primeira hora de prova, se ausentar da sala e a ela não retornar, será eliminado do concurso público.
- 3 - Os três últimos candidatos a terminar a prova deverão permanecer na sala e somente poderão sair juntos do recinto, após aposição em ata de suas respectivas assinaturas.
- 4 - Você poderá levar o seu caderno de questões faltando 1 hora para o término da Prova.

INSTRUÇÕES - PROVA OBJETIVA

- 1 - Confira atentamente se este caderno de perguntas, que contém **60** questões objetivas, está completo.
- 2 - Confira se seus dados e o **cargo** escolhido, indicados no **cartão de respostas**, estão corretos. Se notar qualquer divergência, notifique imediatamente o Fiscal/Chefe Local. Terminada a conferência, você deve assinar o cartão de respostas no espaço apropriado.
- 3 - Verifique se o número do Gabarito e do Caderno de Perguntas é o mesmo.
- 4 - Cuide de seu **cartão de respostas**. Ele não pode ser rasurado, amassado, dobrado nem manchado.
- 5 - Para cada questão objetiva são apresentadas cinco alternativas de respostas, apenas uma das quais está correta. Você deve assinalar essa alternativa de modo contínuo e denso.
- 6 - Se você marcar mais de uma alternativa, sua resposta será considerada errada mesmo que uma das alternativas indicadas seja a correta.

AGENDA

- 16/03/2010, divulgação do gabarito da Prova objetiva:
<http://concursos.biorio.org.br>
- 17 a 19/03/2010, recursos contra o gabarito da Prova Objetiva na Internet:
<http://concursos.biorio.org.br> até as 17h
- Após a transmissão imprimir e entregar no Protocolo da PROCERGS de 9h às 17h
- 07/04/2010, divulgação do resultado da análise dos recursos da Prova Objetiva.
- 08/04/2010, divulgação do Resultado Final da Prova Objetiva.
- Informações:
Tel: 21 3525-2480 das 9 às 17h;
Internet:
<http://concursos.biorio.org.br>
E-mail:
procergs2009@biorio.org.br

Posto de Atendimento:
Av. Ipiranga nº 1.090 (Colégio Estadual Protásio Alves)
9h às 12h e das 13h30min às 17h

**Concursos**

CONHECIMENTOS ESPECÍFICOS

01 - Na lógica de programação, existem estruturas condicionais e de repetição. Nesse sentido, observe as afirmativas a seguir, em relação à estrutura de controle conhecida como *enquanto ... <condição> ... faça ...*, utilizada na programação de computadores.

- I - A condição é testada no final da execução.
- II - Se o resultado do teste é VERDADEIRO, permanece a execução das instruções contidas no bloco de repetição.
- III - Se o resultado do teste é FALSO, a execução é finalizada com saída da estrutura de repetição.

Assinale a alternativa correta:

- (A) apenas a afirmativa I está correta;
- (B) apenas a afirmativa II está correta;
- (C) apenas as afirmativas I e II estão corretas;
- (D) apenas as afirmativas II e III estão corretas;
- (E) todas as afirmativas estão corretas.

02 - No contexto da PL/SQL, a estrutura de controle *para ... faça ...* também pode ser utilizada, permitindo que uma seqüência de comandos seja executada um número fixo de vezes estabelecido no comando. No início do comando a quantidade de vezes que o mesmo será executado já é conhecida, uma vez que não se pode alterar o valor de <contador> durante a iteração. A sintaxe dessa estrutura em PL/SQL está exemplificada na seguinte opção:

- (A)

<pre>FOR k TO 1..3 LOOP <seqüência de comandos> END LOOP;</pre>

- (B)

<pre>FOR k IN 1..3 DO <seqüência de comandos> NEXT ;</pre>
--
- (C)

<pre>FOR k OF 1 TO 3 DO <seqüência de comandos> ENDFOR ;</pre>
--
- (D)

<pre>FOR DO k LOOP 1..3 <seqüência de comandos> NEXT ;</pre>
--
- (E)

<pre>FOR k IN 1..3 LOOP <seqüência de comandos> END LOOP;</pre>

03 - Para recompilar e executar *procedures*, devem ser utilizados comandos PL/SQL, respectivamente, com as seguintes sintaxes:

- (A) RUN PROCEDURE [schema.]procedure COMPILE e SUBMIT NomeProc(Param);
- (B) RUN PROCEDURE [schema.]procedure COMPILE e CALL NomeProc(Param);
- (C) RUN PROCEDURE [schema.]procedure COMPILE e EXEC NomeProc(Param);
- (D) ALTER PROCEDURE [schema.]procedure COMPILE e SUBMIT NomeProc(Param);
- (E) ALTER PROCEDURE [schema.]procedure COMPILE e EXEC NomeProc(Param).

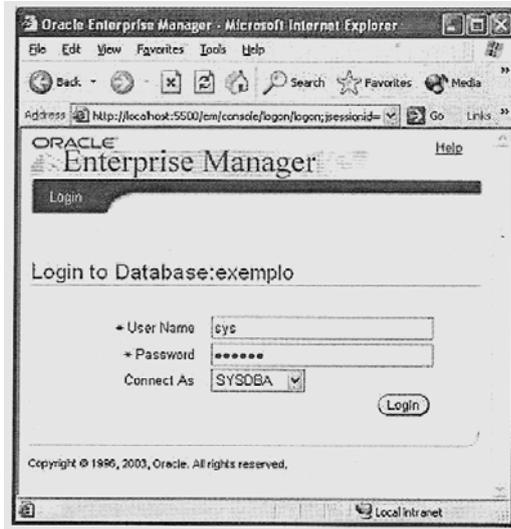
04 - Durante o processo de instalação do *Oracle 10g* são criados usuários automaticamente, sendo solicitada apenas uma senha de acesso, que posteriormente podem ser trocadas. Dois deles são caracterizados a seguir.

- I - Possui o privilégio de SYSDBA e assim pode executar todas as tarefas administrativas sem exceção.
- II - Podem-se executar as tarefas de gerenciamento do banco de dados, mas não se pode iniciar ou parar uma instância.

Esses usuários são, respectivamente:

- (A) SYS e SYSTEM;
- (B) ADMIN e USER;
- (C) MASTER e SLAVE;
- (D) TOTAL e MANAGER;
- (E) ADMINISTRATOR e GUEST.

05 - Observe a figura abaixo, que mostra a janela referente ao *Oracle 10g Enterprise Manager Database Control*.



A janela refere-se a uma ferramenta gráfica que é:

- (A) totalmente Web, devendo ser acessada via browser;
- (B) exclusiva para uso não Web, em ambientes Windows;
- (C) o browser do Oracle, permitindo a navegação em sites na internet;
- (D) o gerenciador de comandos SQL, exclusivamente para ambientes Linux;
- (E) executada a partir do Shell, via comando \$ORACLE_HOME/network/admin.

06 - No que diz respeito aos SGBDs, chave estrangeira é definida como uma chave de uma tabela que apresenta a seguinte característica:

- (A) é utilizada na atribuição dinâmica de chaves primárias à tabela de índices do banco;
- (B) é utilizada quando se deseja quebrar os relacionamentos estabelecidos entre tabelas;
- (C) é utilizada como argumento de pesquisa em uma consulta à tabela de chaves primárias;
- (D) possui associação com a chave primária de uma outra tabela;
- (E) possui relacionamento comutativo com as demais tabelas do banco.

07 - No *Oracle 8i*, as “views” de dicionário de dados dividem-se em três categorias : DBA, ALL e USER. A “view” através da qual é possível saber qual o número de “extents” de uma tabela, é:

- (A) DBA_SEGMENTS;
- (B) DBA_DATA_FILES;
- (C) USER_FREE_SPACE;
- (D) ALL_TAB_EXTENTS;
- (E) ALL SEGMENTS TABLE.

08 - O *Oracle 10g* disponibiliza para os usuários um amplo conjunto de tipos de dados nativos. Os usuários podem adicionar novos tipos utilizando o comando *CREATE TYPE*. Um deles é utilizado para representar ponto flutuante e precisão simples, enquanto que outros é usado para armazenar dados não estruturados e grande quantidade de bytes como imagens, fotos, vídeo, áudio, mapas e gráficos. Esses tipos são, respectivamente:

- (A) REAL e BINARY SMALL OBJECT;
- (B) REAL e BINARY LARGE OBJECT;
- (C) FLOAT e BINARY SMALL OBJECT;
- (D) DOUBLE e BINARY LARGE OBJECT;
- (E) DOUBLE e BINARY SMALL OBJECT.

09 - Na versão *Oracle 10g*, além do desempenho, acrescentaram-se dois novos tipos de dados *BINARY_FLOAT* e *BINARY_DOUBLE*, fazendo com que as operações sejam executadas mais rapidamente do que usando o tipo NUMBER. As únicas instruções SQL permitidas diretamente em um programa *PL/SQL* são *DMLs* como *SELECT*, *INSERT*, *UPDATE* e *DELETE* e instruções de controle de transação, como:

- (A) COMMIT, RECOVERY e CHECKPOINT;
- (B) BACKUP, RECOVERY e CHECKPOINT;
- (C) COMMIT, RECOVERY e SAVEPOINT;
- (D) BACKUP, ROLLBACK e SAVEPOINT;
- (E) COMMIT, ROLLBACK e SAVEPOINT.

10 - No *Oracle 10g* é possível a utilização das chamadas *constraints*, que servem para validar a entrada de dados nas tabelas. Existem vários tipos de *constraints*, como NOT NULL que não permite valor nulo e UNIQUE, que exige valores exclusivos nos registros das tabelas e PRIMARY KEY, para definir uma chave primária em uma tabela. Para definir uma condição a ser satisfeita e para criar um relacionamento entre tabelas, existem as seguintes *constraints*:

- (A) CHECK e TABLE KEY;
- (B) VARIANT e TABLE KEY;
- (C) CHECK e FOREIGN KEY;
- (D) CHECK e RELATION KEY;
- (E) VARIANT e FOREIGN KEY.

11 - No **Oracle 10g** uma instância é uma área de memória à qual os usuários se conectam. Serve como forma de acesso ao banco de dados, sendo composta de duas partes, a *SGA (System Global Area)* e os processos *background*. A *SGA* é composta, principalmente, de três subáreas, uma área de comandos, outra de dados e uma última de armazenamento de histórico. Essas áreas são conhecidas, respectivamente, como:

- (A) *index pool, large pool e history pool;*
- (B) *index area, main area e backup area;*
- (C) *command area, database area e history area;*
- (D) *shared pool, database buffer e redo log buffer;*
- (E) *command pool, database area e backup buffer.*

12 - O **Oracle 10g** é um banco de dados que utiliza um tipo de gerenciamento dos recursos da rede que permite a realocação do processamento e do armazenamento de dados de forma rápida e automática dentro de uma malha corporativa. O conceito da computação em malha é o compartilhamento de diversos recursos entre vários usuários com muitos aplicativos. Corresponde a um cenário em que todos os sistemas computacionais sejam compartilhados, de modo a criar um grupo de recursos dinâmicos. É uma tecnologia que implica em um método flexível e com ótima relação custo/benefício para gerenciamento da informação, conhecida por:

- (A) *multitask computing;*
- (B) *relational computing;*
- (C) *pipeline computing;*
- (D) *shared computing;*
- (E) *grid computing.*

13 - O **Oracle 10g** disponibiliza funções para conversão entre tipos de dados. Assim, para converter seu argumento em um tipo **VARCHAR2** deve ser utilizada a seguinte função:

- (A) **CONVERT_TO_VARCHAR2;**
- (B) **CONVERT_TO_CHAR;**
- (C) **CONVERT_TO_TEXT;**
- (D) **TO_CHAR;**
- (E) **TO_TEXT.**

14 - O **Oracle 11g** está disponível em várias versões, sendo uma delas indicada para corporações de qualquer tamanho que necessitem suportar grandes volumes de transações e aplicações de *Business Intelligence*. Provê escalabilidade para todas as configurações de hardware e pode ser utilizado para gerenciar grandes volumes de informações com alto nível de segurança. Essas características apontam para o **Oracle 11g** na seguinte versão:

- (A) **PROFESSIONAL EDITION;**
- (B) **MIDDLEWARE EDITION;**
- (C) **ENTERPRISE EDITION;**
- (D) **EXPRESSION EDITION;**
- (E) **STANDARD EDITION.**

15 - São *tablespaces* do **Oracle 10g**:

- (A) **INDEX, SYSTEM e DATA;**
- (B) **UNDO, SYSOUT e SYSIN;**
- (C) **SYSTEM, SYSAUX e UNDO;**
- (D) **SYSOUT, SYSIN e SYSAUX;**
- (E) **DATAMAIN, DATARECOVER e INDEX.**

16 - Deseja-se criar no **Oracle 10g** uma *tablespace* que aloca um espaço em que tabelas ficam armazenadas. O *datafile* associado a ela deve ter tamanho de 60 MB, chamado **SPACE** que ter expandir seu tamanho automaticamente de 20 em 20 MB. A sintaxe para o comando é:

- (A)

```
CREATE TABLESPACE SPACE
DATAFILE 'C:\ORACLEXE\ORADATA\SPACE.DBF'
SIZE 60 M AUTOMATIC ON NEXT 20 M
MAXSIZE UNLIMITED;
```
- (B)

```
CREATE TABLESPACE SPACE
DATAFILE 'C:\ORACLEXE\ORADATA\SPACE.DBF'
SIZE UNLIMITED AUTOEXTENDED ON NEXT 20 M
MAXSIZE 60 M;
```
- (C)

```
CREATE TABLESPACE SPACE
DATAFILE 'C:\ORACLEXE\ORADATA\SPACE.ORA'
AUFOSIZE 60 M EXTENDED ON NEXT 20 M
MAXSIZE UNLIMITED;
```
- (D)

```
CREATE TABLESPACE SPACE
DATAFILE 'C:\ORACLEXE\ORADATA\SPACE.ORA'
AUFOSIZE UNLIMITED AUTOMATIC ON NEXT 20 M
MAXSIZE 60 M;
```
- (E)

```
CREATE TABLESPACE SPACE
DATAFILE 'C:\ORACLEXE\ORADATA\SPACE.DBF'
SIZE 60 M AUTOEXTENDED ON NEXT 20 M
MAXSIZE UNLIMITED;
```

17 - Uma base de dados **Oracle 10g** é constituída por um conjunto de três tipos de arquivos, que são:

- (A) de índice, de controle e de *log*;
- (B) de índice, de dados e de tabelas;
- (C) de controle, de dados e de *redo log*;
- (D) de índice, de dados primários e de *backup*;
- (E) de controle, de dados primários e de *recover*.

18 - A **Data Definition Language – DDL** descreve como as tabelas e outros objetos **Oracle** podem ser definidos, alterados e removidos. São comandos da DDL:

- (A) CREATE TABLE, ALTER TABLE e DROP TABLE;
- (B) UPDATE TABLE, DROP TABLE e SELECT TABLE;
- (C) ALTER TABLE, DELETE TABLE e CREATE TABLE;
- (D) SELECT TABLE, INSERT TABLE e DELETE TABLE;
- (E) ALTER TABLE, e UPDATE TABLE e INSERT TABLE.

19 - O **Oracle 10g** possui um extenso conjunto de tipos de dados nativos para muitas aplicações. Se *integer* é utilizado para números inteiros, uma função retorna a data atual, enquanto que outra retorna a data e hora atuais do sistema. Para esses casos, o **Oracle** disponibiliza, respectivamente, as seguintes funções:

- (A) DATE e DATE_HOUR;
- (B) STAMPDATE e STAMPTIME;
- (C) SYSDATE e SYSTIMESTAMP;
- (D) ACTUAL_DATE e ACTUAL_TIME;
- (E) CURRENCY_DATE e CURRENCY_TIME.

20 - O administrador do banco de dados – **DBA** precisa executar comandos diferenciados para analisar e gerenciar um banco de dados **Oracle 10g**. Para parar a execução de um banco de dados, o **DBA** utiliza o comando *shutdown*, empregado com parâmetros de acordo com a situação. Um parâmetro é utilizado numa situação em que as instruções não são concluídas, sendo que o **Oracle** fecha e desmonta o **BD** antes de desativar. Nessas condições, o comando a ser dado pelo **DBA** será o:

- (A) *shutdown* TRANSACIONAL;
- (B) *shutdown* IMMEDIATE;
- (C) *shutdown* NORMAL;
- (D) *shutdown* ABORT;
- (E) *shutdown* NOW.

21 - No **Oracle 10g**, o **Oracle Listener** é o processo no servidor que “escuta” as solicitações de requisições na porta **TCP/IP** determinada para o **Oracle**. Quando o **Listener** recebe uma requisição válida de conexão de uma aplicação cliente, ele direciona a requisição para o servidor de banco de dados. A partir daí, cliente e o servidor comunicam-se diretamente. Para um tipo de requisição *database*, usada para conexão do **BD** utilizando o **Oracle Net** sobre o **TCP/IP**, como no caso da conexão remota do **SQL Command Line** ou de aplicações **Java** conectadas com **JDBC**, a porta **TCP** padrão é:

- (A) 1024;
- (B) 1521;
- (C) 3121;
- (D) 4096;
- (E) 8080.

22 - No **Oracle 10g**, inserido nos comandos **SQL** existem dois tipos de funções, uma função de linha, que retorna um resultado por linha da tabela e, uma função de grupo, que retorna um resultado por grupo de linhas da tabela. Nesse sentido, pode-se afirmar que, com relação à execução da função **SUBSTR** (*string*, *m*[,*n*]), são válidas as afirmações contidas nas opções abaixo, EXCETO:

- (A) se *n* for menor que 1, retorna **NULL**;
- (B) se *n* não está presente, retorna a *string* inteira como padrão;
- (C) se *m* for positivo, os caracteres que retornam são contados da esquerda;
- (D) resulta no retorno da *string* que inicia no caractere *n*, com comprimento dos caracteres *m*;
- (E) se *m* for negativo, os caracteres retornam iniciando do final da *string* e são contados da direita.

23 - No **Oracle 10g**, para acessar o gerador de consultas, deve-se selecionar a opção **SQL** → **COMANDO** → **Criar**. Nesse caso, **COMANDO** é denominado:

- (A) *Query Resource*;
- (B) *Query Config*;
- (C) *Query Builder*;
- (D) *Query View*;
- (E) *Query Tool*.

24 - No ambiente *Oracle*, a *Data Control Language* – *DCL* controla os aspectos de autorização de dados e licenças de usuários para controlar quem tem acesso para ver ou manipular dados dentro do banco de dados. Um dos comandos remove ou restringe a capacidade de um usuário de executar operações, enquanto que outro autoriza o usuário a executar ou setar operações. Esses comandos são denominados, respectivamente:

- (A) REMOVE e RELEASE;
- (B) REMOVE e GRANT;
- (C) LOCK e GRANT;
- (D) REVOKE e GRANT;
- (E) REVOKE e RELEASE.

25 - Utilizando *SQL* no *Oracle 10g*, deseja-se selecionar todos os funcionários com departamento igual ao conjunto de todos os departamentos de todos os funcionários. Para isso, deve-se utilizar o seguinte comando *SQL*:

- (A)

```
SELECT * FROM FUNCIONARIO  
WHERE DEPARTAMENTO IN (SELECT DEPARTAMENTO FROM FUNCIONARIO);
```
- (B)

```
SELECT * FROM FUNCIONARIO  
WHERE DEPARTAMENTO GROUP BY (SELECT DEPARTAMENTO FROM FUNCIONARIO);
```
- (C)

```
SELECT * FROM FUNCIONARIO  
WHERE DEPARTAMENTO HAVING (SELECT DEPARTAMENTO FROM FUNCIONARIO);
```
- (D)

```
SELECT * FROM FUNCIONARIO  
WHERE DEPARTAMENTO ON (SELECT DEPARTAMENTO FROM FUNCIONARIO);
```
- (E)

```
SELECT * FROM FUNCIONARIO  
WHERE DEPARTAMENTO BY (SELECT DEPARTAMENTO FROM FUNCIONARIO);
```

26 - Um recurso no *ORACLE 10g* define as codificações em *PL/SQL* executadas automaticamente quando ocorre algum evento vinculado a uma tabela ou visão, e que ficam armazenadas no banco de dados e são executadas em uma operação *INSERT*, *UPDATE* ou *DELETE* de uma tabela do banco de dados. Esse recurso é acionado automaticamente em decorrência de determinadas ações sobre os dados de uma tabela, sendo conhecido como:

- (A) trigger;
- (B) package;
- (C) flashback;
- (D) foreign function;
- (E) stored procedure.

27 - Analise o comando *SQL* no *Oracle 10g*.

```
SELECT *  
FROM ( SELECT *  
        FROM CADASTRO  
        ORDER BY MATRICULA DESC )  
WHERE ROWNUM <= 2  
ORDER BY MATRICULA
```

Esse comando tem por objetivo:

- (A) Mostrar os registros da tabela CADASTRO, em ordem crescente por MATRICULA;
- (B) Criar uma tabela ROWNUM por meio da técnica DML, em ordem decrescente de MATRICULA;
- (C) Criar uma nova tabela por meio da técnica *join*, com dados ordenados em ordem ascendente;
- (D) Mostrar as duas primeiras linhas do CADASTRO, ordenados por MATRICULA de forma decrescente;
- (E) Gerar um relatório com dados da tabela CADASTRO, ordenados em ordem decrescente por ROWNUM.

28 - Uma visão é uma forma particular de visualizar dados de uma ou mais tabelas. No *Oracle*, uma visão pode ser criada somente para consulta de dados ou leitura, como também atualizações ou eliminações de dados. Uma *view* que permite visualizar o nome de todos os funcionários cadastrados é exemplificada na opção:

- (A) REPLACE VIEW lista_funcionarios AS SELECT nome ON FUNCIONARIOS;
- (B) REPLACE VIEW lista_funcionarios WITH SELECT nome FROM FUNCIONARIOS;
- (C) CREATE VIEW lista_funcionarios BY SELECT nome FROM FUNCIONARIOS;
- (D) CREATE VIEW lista_funcionarios WITH SELECT nome ON FUNCIONARIOS;
- (E) CREATE VIEW lista_funcionarios AS SELECT nome FROM FUNCIONARIOS.

29 - Utilizando *SQL* no *Oracle 10g*, deseja-se obter o nome dos clientes e dos funcionários, mas mostrando elementos duplicados no resultado. Para isso, deve-se utilizar o seguinte comando *SQL*:

- (A)

```
SELECT NOME FROM CLIENTE
UNION ALL
SELECT NOME_FUNCIONARIO FROM FUNCIONARIO
```
- (B)

```
SELECT NOME FROM CLIENTE
UNION KEEP
SELECT NOME_FUNCIONARIO FROM FUNCIONARIO
```
- (C)

```
SELECT NOME FROM CLIENTE
UNION
SELECT NOME_FUNCIONARIO FROM FUNCIONARIO
```
- (D)

```
SELECT NOME FROM CLIENTE
UNION NOT REMOVE
SELECT NOME_FUNCIONARIO FROM FUNCIONARIO
```
- (E)

```
SELECT NOME FROM CLIENTE
UNION INCLUDE
SELECT NOME_FUNCIONARIO FROM FUNCIONARIO
```

30 - Analise o código abaixo, que se refere a um exemplo prático de criação de usuário no *Oracle 10g*.

```
CREATE USER Fabiano
IDENTIFIED BY fabuloso
PADRÃO TABLESPACE data
TEMPORARY TABLESPACE temp
QUOTA 10M ON data
QUOTA 35M ON users
PASSWORD EXPIRE;
```

Os parâmetros *QUOTA* e *PASSWORD EXPIRE* são definidos, respectivamente, com os seguintes objetivos:

- (A) definir o espaço máximo permitido para os objetos pertencentes ao usuário no *tablespace* / forçar o usuário a redefinir a senha quando ele efetua o primeiro *login* no banco de dados usando o *SQL*Plus*.
- (B) definir o espaço máximo permitido no disco rígido / forçar o usuário a redefinir a senha quando ele efetua o primeiro *login* no banco de dados usando o *SQL*Plus*.
- (C) definir o espaço máximo permitido no disco rígido / forçar o usuário a redefinir a senha quando ele efetua o *login* no banco de dados usando o *SQL*Plus*.
- (D) definir o espaço máximo permitido para os objetos pertencentes ao usuário no *tablespace* / forçar o usuário a redefinir a senha toda a vez que ele efetua o *login* no banco de dados usando o *SQL*Plus*.
- (E) definir o espaço máximo permitido para os objetos pertencentes ao usuário no *tablespace* / forçar o usuário a redefinir a senha quando ele efetua o terceiro *login* no banco de dados usando o *SQL*Plus*.

31 - No *Oracle10g*, para gerar um subtotal da soma dos salários em cada departamento e um subtotal de todos os salários dos empregados dentro da empresa, deve-se utilizar o seguinte comando:

- (A)

```
SELECT departamento_id, empregado_id, SUM(salario)
ON empregado GROUP BY CUBE (departamento_id,
empregado_id);
```
- (B)

```
SELECT departamento_id, empregado_id, SUM(salario)
FROM empregado GROUP BY CUBE (departamento_id,
empregado_id);
```
- (C)

```
SELECT departamento_id, empregado_id, SUM(salario)
FROM empregado GROUP BY CUBE departamento_id,
empregado_id, SUM(salario);
```
- (D)

```
SELECT departamento_id, empregado_id, SUM(salario)
ON empregado GROUP BY ROLLUP (departamento_id,
empregado_id);
```
- (E)

```
SELECT departamento_id, empregado_id, SUM(salario)
FROM empregado GROUP BY ROLLUP
(departamento_id, empregado_id).
```

32 - Em uma consulta em um Banco de Dados Oracle, um parâmetro deve ser utilizado na sintaxe do comando *SELECT*, para eliminar linhas em duplicata na tabela resultante. Esse parâmetro é:

- (A) *Different*;
- (B) *Discrete*;
- (C) *Distinct*;
- (D) *Unique*;
- (E) *Unlike*.

33 - O *Oracle Server* usa *constraints* para impedir a entrada de dados inválidos nas tabelas. Uma dessas *constraints*, também chamada de integridade referencial, designa uma coluna ou combinação de colunas como uma chave estrangeira e estabelece um relacionamento entre uma chave primária ou uma chave exclusiva na mesma tabela ou em uma tabela diferente. Este tipo é denominado:

- (A) Constraint *CHECK*;
- (B) Constraint *UNIQUE*;
- (C) Constraint *NOT NULL*;
- (D) Constraint *PRIMARY KEY*;
- (E) Constraint *FOREIGN KEY*.

34 - Analise a estrutura a seguir, referente a um código em PL/SQL

```
DECLARE
  v_TestVar NUMBER :=1;
BEGIN
  ESTRUTURA
END;
```

A sintaxe correta para a codificação da estrutura CASE, que deve substituir a palavra ESTRUTURA, é exemplificada na opção:

- (A)

```
CASE v_TestVar
  2 : DBMS_OUTPUT_LINE(' Dois!');
  3 : DBMS_OUTPUT_LINE(' Três!');
ELSE DBMS_OUTPUT_LINE(' Hum!');
END ;
```
- (B)

```
SWITCH v_TestVar
  2 : DBMS_OUTPUT_LINE(' Dois!');
  3 : DBMS_OUTPUT_LINE(' Três!');
OTHERWISE DBMS_OUTPUT_LINE(' Hum!');
END ;
```
- (C)

```
CASE v_TestVar
  2 = DBMS_OUTPUT_LINE(' Dois!');
  3 = DBMS_OUTPUT_LINE(' Três!');
OTHERWISE DBMS_OUTPUT_LINE(' Hum!');
END CASE;
```
- (D)

```
SWITCH v_TestVar
  = 2 : DBMS_OUTPUT_LINE(' Dois!');
  = 3 : DBMS_OUTPUT_LINE(' Três!');
ELSE DBMS_OUTPUT_LINE(' Hum!');
END SWITCH;
```
- (E)

```
CASE v_TestVar
  WHEN 2 THEN DBMS_OUTPUT_LINE(' Dois!');
  WHEN 3 THEN DBMS_OUTPUT_LINE(' Três!');
ELSE DBMS_OUTPUT_LINE(' Hum!');
END CASE;
```

35 - Na linguagem *PL/SQL*, para exponenciação, conjunção lógica e diferente de na comparação lógica, são utilizados, respectivamente, os seguintes operadores:

- (A) **, OR e !=;
(B) **, OR e <>;
(C) ^^, AND e !=;
(D) ^^, AND e <>;
(E) **, AND e !=.

36 - Os índices são utilizados para agilizar o processamento das consultas, fazendo com que elas sejam executadas com maior rapidez. Para alterar um índice, um exemplo da sintaxe é:

- (A) ALTER INDEX joaquim.funcionario.idx ROOLBACK COMPUTE STATISTICS;
(B) ALTER INDEX joaquim.funcionario.idx RECOVER COMPUTE STATISTICS;
(C) ALTER INDEX joaquim.funcionario.idx COMMIT COMPUTE STATISTICS;
(D) ALTER INDEX joaquim.funcionario.idx REBUILD COMPUTE STATISTICS;
(E) ALTER INDEX joaquim.funcionario.idx REINDEX COMPUTE STATISTICS.

37 - *PL/SQL* é uma linguagem padrão do **ORACLE** que apresenta a característica de permitir a construção de programas compostos por blocos, inclusive aninhados em outros. Os tipos de blocos *PL/SQL* podem ser blocos anônimos rotulados ou não, subprogramas conhecidos por procedures ou funções e triggers. Para começar a programação com *PL/SQL*, um comando de configuração que permite a ativação da emissão de mensagens, é:

- (A) SET MESSAGE ENABLE;
(B) SET SERVEROUTPUT ON;
(C) SET MESSAGE DISABLE;
(D) SET ENABLE CONFIG ON;
(E) SET SERVEROUTPUT OFF.

38 - No que diz respeito à segurança e, em consequência com o uso de passwords, o **Oracle** disponibilizava uma rotina de verificação de password no *script \$ORACLE_HOME/rdbms/admin/utlpwdmg.sql*. O **Oracle Database 11g** melhorou essa funcionalidade. Na nova versão do *utlpwdmg.sql* encontra-se duas cópias diferentes da rotina de verificação de password. A primeira é uma nova versão denominada *verify_function_11G*, enquanto que a versão antiga, continua disponível no *utlpwdmg.sql*. Uma nova função de verificação oferecida pela nova versão *verify_function_11G* preconiza que uma password:

- (A) deve conter um dígito e uma letra;
(B) deve ter um mínimo de treze caracteres;
(C) pode ser o mesmo que o nome de um usuário;
(D) pode ser do mesmo comprimento que o nome do usuário;
(E) pode ser o nome do usuário soletrado de traz para a frente.

39 - No que diz respeito à segurança o *Oracle 11g* adicionou novos parâmetros para ajudar a proteger contra vários ataques de internet, incluindo os ataques DOS e de força bruta. Dentre esses parâmetros, um define uma quantidade de tentativas de autenticação, que uma conexão do cliente pode fazer em um servidor, antes que o processo do cliente seja derrubado. O valor padrão é 10. Esse parâmetro é denominado:

- (A) Sec_case_sensitive_login;
- (B) Sec_max_failed_login_attempts;
- (C) Sec_protocol_error_trace_action;
- (D) Sec_return_server_release_banner;
- (E) Sec_protocol_error_further_action.

40 - Dentre os novos recursos do *Oracle 11g* está um tipo de colunas, que contém dados derivados, dentro do banco de dados. Para usar estas colunas pode-se usar o comando *alter table*. O mais importante é que esse novo recurso não consome qualquer espaço de armazenamento, visto que são computadas na execução. No entanto, existem algumas restrições para o seu uso.

- Não se pode escrever nesse tipo de coluna;
- Não há suporte para as tabelas de índice organizado, externa, objeto, grupo ou temporárias;
- Não há suporte para os tipos de dados atendidos pelo *Oracle* e tipos definidos pelo usuário.

Esse tipo é denominada coluna:

- (A) particionada;
- (B) segmentada;
- (C) matricial;
- (D) vetorial.
- (E) virtual.

41 - Num banco de dados *Oracle 11g*, um índice é uma estrutura de dados que acelera o acesso a linhas numa tabela. Esse índice é composto de um ou mais níveis de ramos de blocos e um único nível de blocos terminais – “leaf nodes”. Os blocos dos ramos contêm informações sobre a faixa de valores contida no nível seguinte de blocos. O número de níveis de ramos entre a raiz e os blocos folha é chamado de profundidade do índice. Os blocos folha contêm os valores do índice em si e os ROWID das linhas associadas. O padrão usado no *Oracle* é conhecido como índice:

- (A) B*-tree;
- (B) Binary-Tree;
- (C) De Mapas de Bits;
- (D) De Lista Invertida;
- (E) De Chaves Reversas.

42 - Observe abaixo um trecho de código que ilustra a utilização de uma CLÁUSULA introduzida nos novos recursos do *Oracle 11g*, e que permite definir a ordem em que um grupo de *triggers* será executado.

```
Create or replace trigger trigger_two
before insert on test
CLÁUSULA trigger_one
begin
    null;
end;
/
```

Nessas condições, a palavra que substitui CLÁUSULA no código é:

- (A) *set*;
- (B) *define*;
- (C) *follow*;
- (D) *priority*;
- (E) *sequence*.

43 - Garantir a integridade e a recuperabilidade dos dados é função primordial do DBA e assim, a realização e a validação de *backup* são vitais. Nesse processo, o modo de arquivamento inicial é definido no comando CREATE DATABASE, sendo definido como:

- (A) POLICY BACKUP;
- (B) ARCHIVELOG;
- (C) MANDATORY;
- (D) BACKUP SET;
- (E) NOARCHIVELOG.

44 - Muitas vezes um DBA que gerencia um banco de dados Oracle 10g precisa excluir uma determinada tabela. Nesse caso, para excluir uma tabela denominada CLIENTES, a sintaxe indicada é:

- (A) DELETE TABLE pessoal PURGE;
- (B) DROP TABLE pessoal ERASE;
- (C) DROP TABLE pessoal PURGE;
- (D) DELETE TABLE pessoal ALL;
- (E) DROP TABLE pessoal ALL.

45 - No funcionamento normal de um banco de dados Oracle ocorrem atividades de manutenção, com inclusão, alteração e exclusão de dados e informações, Nessa manutenção de registros, é preciso utilizar um comando do SQL, sendo a sintaxe correta desse comando exemplificada na seguinte opção:

- (A) UPDATE CADASTRO ON DEPTO = 'PESSOAL' TO NOME = 'CLAUDIO';
- (B) UPDATE CADASTRO SET DEPTO = 'PESSOAL' WHERE NOME = 'CLAUDIO';
- (C) UPDATE CADASTRO OVER DEPTO = 'PESSOAL' WHERE NOME = 'CLAUDIO';
- (D) UPDATE CADASTRO ON DEPTO = 'PESSOAL' WHERE NOME = 'CLAUDIO';
- (E) UPDATE CADASTRO SET DEPTO = 'PESSOAL' TO NOME = 'CLAUDIO'.

46 - O Oracle 10g disponibiliza um recurso que permite ao DBA executar tarefas associadas ao armazenamento e gerenciamento de discos, dentro de uma estrutura totalmente Oracle, independente de fornecedores. Com esse recurso é possível agrupar um conjunto de discos físicos em uma entidade lógica denominada *diskgroup*, análogo a um arquivo de sistema espalhado – “*striping de filesystem*”, servindo para armazenar arquivos do banco sendo que não é “bufferizado”. Esse recurso é conhecido por:

- (A) SAN – Storage Area Network;
- (B) NAS – Network Attached Storage;
- (C) DRM – Database Resource Manager;
- (D) ODM – Optimizer Database Manager;
- (E) ASM – Automatic Storage Management.

47 - Um DBA gerencia uma base dados Oracle10i e deparou-se com o problema das tabelas grandes. Com o objetivo de melhorar o desempenho das operações de manutenção, *backup*, recuperações, transações e consultas, a solução e particioná-las. Para isso, são dois métodos de particionamento disponíveis:

- (A) *Swap Partitioning* e *List Partitioning*;
- (B) *List Partitioning* e *Swap Partitioning*;
- (C) *Hash Partitioning* e *Range Partitioning*;
- (D) *Hash Partitioning* e *Encryption Partitioning*;
- (E) *Encryption Partitioning* e *Range Partitioning*.

48 - No contexto do *Oracle 10g*, privilégio é o direito de executar um tipo de instrução SQL ou de acessar o objeto de um outro usuário. Estão incluídos os direitos de estabelecer conexão com um banco de dados, criar uma tabela, selecionar linhas da tabela de outro usuário e executar uma procedure de outro usuário. A tarefa do DBA de controlar os privilégios inclui possibilitar ao usuário o direito de realizar um tipo de operação, conceder e revogar acesso para a execução de funções do sistema, conceder privilégios diretamente aos usuários ou às *roles* e conceder privilégios a todos os usuários. Para conceder privilégios de sistema aos usuários é utilizada a seguinte instrução SQL:

- (A) ACCOUNT;
- (B) GRANT;
- (C) PROFILE;
- (D) ACCESS;
- (E) RIGHTS.

49 - Uma *view* é uma tabela lógica baseada em uma tabela ou em outra *view*, que não contém dados próprios, mas é como uma janela através da qual os dados das tabelas podem ser vistos ou alterados. Um exemplo para gerar uma *view* é:

- (A)

```
GENERATE VIEW v80
AS SELECT matricula, nome, salario
FROM pessoal WHERE depto = 80;
DESCRIBE v80
```
- (B)

```
GENERATE VIEW v80
OF SELECT matricula, nome, salario
ON pessoal WHERE depto = 80;
DESCRIBE v80
```
- (C)

```
CREATE VIEW v80
AS SELECT matricula, nome, salario
FROM pessoal WHERE depto = 80;
DESCRIBE v80
```
- (D)

```
CREATE VIEW v80
AS SELECT matricula, nome, salario
ON pessoal WHERE depto = 80;
DESCRIBE v80
```
- (E)

```
CREATE VIEW v80
OF SELECT matricula, nome, salario
FROM pessoal WHERE depto = 80;
DESCRIBE v80
```

50 - O comando Create Trigger cria e habilita *triggers*, que é um bloco de código PL/SQL associado a uma tabela. Se um comando SQL for executado sobre esta tabela as *triggers* são disparadas automaticamente pelo Oracle. Normalmente as regras de negócio são implementadas em *triggers*. A seguir é mostrada a sintaxe desse comando.

```
CREATE [OR REPLACE] TRIGGER
[schema.]trigger
  {BEFORE | AFTER}
  {DELETE | INSERT | UPDATE [OF column
  [, column] ...]}

[OR {DELETE | INSERT | UPDATE [OF
column [, column] ...]} ...
ON [schema.]table
  [ [REFERENCING { OLD [AS] old [NEW
  [AS] new
  | NEW [AS] new [OLD [AS] old } ] ]
FOR EACH ROW
[WHEN (condition)] ]
pl/sql_block
```

Os parâmetros **schema**, **BEFORE** e **WHEN** são utilizados, respectivamente, com os seguintes significados:

- (A) nome da *trigger* que está sendo criada / indica ao ORACLE para disparar a *trigger* ANTES de executar o comando / contém uma condição SQL que deve ser FALSA para permitir o disparo da *trigger*;
- (B) nome do usuário dono da *trigger* / indica ao ORACLE para disparar a *trigger* ANTES de executar o comando / contém uma condição SQL que deve ser VERDADEIRA para permitir o disparo da *trigger*;
- (C) nome da *trigger* que está sendo criada / indica ao ORACLE para disparar a *trigger* DEPOIS de executar o comando / contém uma condição SQL que deve ser VERDADEIRA para permitir o disparo da *trigger*;
- (D) nome do usuário dono da *trigger* / indica ao ORACLE para disparar a *trigger* ANTES de executar o comando / contém uma condição SQL que deve ser FALSA para permitir o disparo da *trigger*;
- (E) nome do usuário dono da *trigger* / indica ao ORACLE para disparar a *trigger* DEPOIS de executar o comando / contém uma condição SQL que deve ser FALSA para permitir o disparo da *trigger*.

51 - Observe e analise o código abaixo, que utiliza sintaxe PL/SQL.

```
CREATE OR REPLACE TRIGGER

```

O objetivo e o significado desse código estão indicado na seguinte opção:

- (A) gerar chave primária para a tabela FILIAL, dependendo da primeira fatura da filial na tabela FATURA;
- (B) gerar chave primária para a tabela FATURA, dependendo da primeira fatura da filial na tabela FILIAL;
- (C) gerar chave primária para a tabela FILIAL, dependendo da última fatura da filial na tabela FATURA;
- (D) gerar chave primária para a tabela FATURA, dependendo da última fatura da filial na tabela FILIAL;
- (E) gerar chave primária para as tabelas FATURA e FILIAL, independente da fatura.

52 - O comando *Create Function* possibilita a criação de uma função de usuário. Uma função constitui um conjunto de comandos PL/SQL normalmente utilizados para efetuar um cálculo ou manipular um valor. Podem ser utilizadas dentro de comandos SQL. Nas functions não é aconselhável fazer alterações no status da base, por exemplo, incluir, alterar, deletar registros. Ao alterar o status do banco, esta função não poderá ser utilizada dentro de comandos SQL. Assim a utilidade da function é descaracterizada, assumindo a função da procedure que é própria para alterar dados. Nesse contexto, analise o código abaixo, que utiliza sintaxe PL/SQL.

```
CREATE FUNCTION cal_media(p_deptno
IN NUMBER)
RETURN NUMBER IS v_media number;
BEGIN
    SELECT avg(sal)
    INTO v_media
    FROM emp
    WHERE dept_no = p_deptno;
    RETURN(v_media);
END;
```

Após analisar o código acima, verifica-se que refere-se a uma função que tem por objetivo determinar:

- (A) o maior salário em um determinado departamento;
- (B) o menor salário em um determinado departamento;
- (C) a média dos salários de um determinado departamento;
- (D) os salários com valores acima de um valor em um determinado departamento;
- (E) um aumento de 10% nos salários no âmbito de um determinado departamento.

53 - Para recompilar e executar procedures, devem ser utilizados comandos PL/SQL, respectivamente, com as seguintes sintaxes:

- (A) RUN PROCEDURE [schema.]procedure COMPILE e SUBMIT NomeProc(Param);
- (B) RUN PROCEDURE [schema.]procedure COMPILE e CALL NomeProc(Param);
- (C) RUN PROCEDURE [schema.]procedure COMPILE e EXEC NomeProc(Param);
- (D) ALTER PROCEDURE [schema.]procedure COMPILE e SUBMIT NomeProc(Param);
- (E) ALTER PROCEDURE [schema.]procedure COMPILE e EXEC NomeProc(Param).

54 - Em PL/SQL uma *warning* ou *error condition* é chamada uma **exception**. Existem algumas exceções já definidas pelo Oracle com MNEUMÔNICOS para referência e, para as demais, podem ser dados nome pelo usuário. Quando um erro ocorre, uma **exception** é setada, isto é, a seqüência de execução do programa é interrompida e o controle é transferido para a área de tratamento de execução do programa. As exceções pré-definidas pelo Oracle são setadas quando a condição de erro ocorre. As exceções criadas pelo programa deverão ser setadas explicitamente pelo verbo RAISE. O nome da exceção *DUP_VAL_ON_INDEX* corresponde ao Oracle Error ORA-00001 e expressa uma condição de erro que é setada se:

- (A) houver ocorrido uma divisão por zero;
- (B) ocorrer *timeout* enquanto o ORACLE estiver aguardando por um recurso;
- (C) acontecer conversão, *constraint error*, *truncation* em uma operação aritmética;
- (D) for tentada uma inclusão de uma coluna com valor duplicado em uma tabela que possui um índice *unique* nessa coluna;
- (E) algum comando SQL tentar uma conversão de uma string para número e esta conversão falhar porque a *string* não representa um número.

55 - A **PROCERGS** emprega um sistema computadorizado que possui as seguintes características:

- Utiliza um banco de dados **Oracle 11g** operando segundo a filosofia cliente/servidor.
- O sistema prioriza um requisito onde o tempo de resposta é de suma importância e altamente relevante para a credibilidade da instituição.
- Todas as transações geram a atualização imediata no banco de dados Oracle, no momento em que ocorrem.

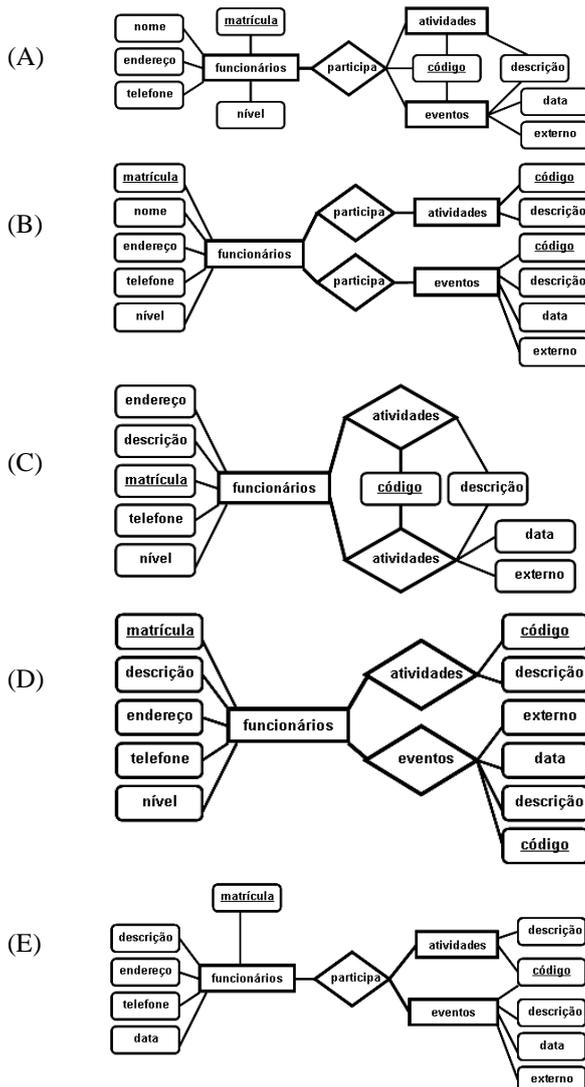
Conclui-se que é um sistema do tipo:

- (A) online/batch;
- (B) offline/batch;
- (C) batch/real time;
- (D) online/real time;
- (E) offline/real time.

56 - Um funcionário da **PROCERGS** está modelando o banco de dados **Oracle 11g** da empresa e nessa tarefa verificou, no seu estudo preliminar, que é conveniente manter informações sobre os funcionários, e estabeleceu os seguintes requisitos:

- I - cada funcionário possui um único número de matrícula, nome, endereço e telefone.
- II - os funcionários participam de atividades, como cursos de auto-aperfeiçoamento e palestras e dos eventos realizados e patrocinados pela PROCERGS.
- III - as atividades são identificadas por um código, sendo desejável que se armazene também o nome da atividade.
- IV - os eventos são externos ou internos, são identificados por um código individual, possuem uma descrição específica e data de realização.
- V - conforme o nível funcional e apreciação pelo órgão responsável, todos os funcionários podem participar dos eventos.
- VI - é importante ainda que a PROCERGS conheça a participação de cada funcionário em cada atividade ou evento.

Nas condições acima descritas, o *Diagrama Entidade-Relacionamento* está indicado na opção:



57 - No banco de dados **Oracle 11g** do **PROCERGS**, que contém as tabelas **FUNCIONARIOS**, **ANALISTAS** e **PROGRAMADORES**, ao cadastrar um *Programador* é feita a verificação se a *matrícula do Funcionário* que foi digitado, já existe na tabela **FUNCIONARIOS**. Se não existir, o cadastro do *Programador* não será aceito.

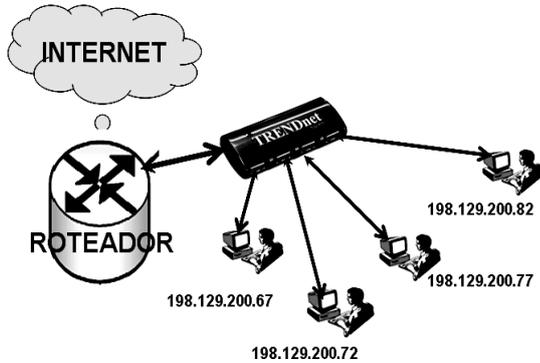
Entretanto, existe um recurso que garante a execução das seguintes situações:

- I - Quando uma *matrícula do Funcionário* for excluída da tabela **FUNCIONARIOS**, pode-se configurar para que o banco de dados exclua, automaticamente, na tabela **PROGRAMADORES**, toda tupla de *Programador* para o Funcionário que está sendo excluído. Esse procedimento é conhecido como "**Propagar exclusão dos registros relacionados**".
- II - Quando a *matrícula do Funcionário* for alterada na tabela **FUNCIONARIOS**, pode-se configurar para o banco de dados atualizar, automaticamente, todos as tuplas de *matrículas dos Funcionários* na tabela **PROGRAMADORES**, de tal forma que não fiquem registros órfãos, isto é, registros de **PROGRAMADORES** com uma *matrícula do funcionário* para o qual não existe mais um correspondente na tabela **FUNCIONARIOS**. Esse procedimento é conhecido como "**Propagar atualização dos campos relacionados**".

O recurso acima descrito é denominado:

- (A) integridade referencial;
- (B) integridade globalizada;
- (C) análise de inconsistências;
- (D) encapsulamento particionado;
- (E) encapsulamento segmentado.

58 - A figura abaixo refere-se à infra-estrutura de uma sub-rede que possui quatro microcomputadores utilizando os IP's indicados e que dá suporte a uma Base de Dados **Oracle 11g**.



Sendo utilizado um esquema de máscara de tamanho fixo e o CIDR 198.129.200.64/26, pode-se afirmar que a máscara que essa sub-rede está utilizando é:

- (A) 255.255.255.0;
- (B) 255.255.255.192;
- (C) 255.255.255.224;
- (D) 255.255.255.240;
- (E) 255.255.255.248.

59 - Em uma empresa de informática, um Analista exerce suas funções de **DBA** na administração de um BD **Oracle**. Num determinado dia, ele se deparou com a seguinte situação:

- I - Um programa bloqueou vários registros para evitar a ocorrência de corrida.
- II - Um processo PROCERGS_1 bloqueou o acesso a um registro RG1
- III - O processo PROCERGS_2 bloqueou o acesso a um registro RG2
- IV - O processo PROCERGS_1 tentou acessar o registro RG2
- V - O processo PROCERGS_2 tentou acessar o registro RG1
- VI - Neste ponto, foi criada uma situação de impasse, e os processos PROCERGS_1 e PROCERGS_2 ficaram bloqueados e assim permaneceram.

Essa situação é caracterizada e conhecida pelo termo:

- (A) *Hashing*;
- (B) *Crashing*;
- (C) *Overlock*;
- (D) *Deadlock*;
- (E) *Quicklock*.

60 - A **PL/SQL** permite a utilização de diversas funções pré-definidas para manipulação dos dados. Pode-se usá-las onde expressões do mesmo tipo são permitidas. Assim, duas situações são descritas a seguir.

- I - FNC1 - retira, da esquerda para direita, os caracteres <set> até que seja encontrado um caracter diferente de <set>
- II - FNC2 - retorna a data e hora correntes, do sistema.

As funções aqui definidas como FNC1 e FNC2 são identificadas, respectivamente, por:

- (A) RTRIM e SYSDATE;
- (B) LTRIM e SYSDATE;
- (C) SUBSTRING e SYSDATE;
- (D) LTRIM e DATENOW;
- (E) RTRIM e DATENOW.



BIO-RIO Concursos

Av. Carlos Chagas Filho, 791 - Cidade Universitária - Ilha do Fundão – RJ

Central de Atendimento: (21) 3525-2480

Internet: <http://concursos.biorio.org.br>

E-mail: concursos@biorio.org.br

procergs2009@biorio.org.br