

PREFEITURA MUNICIPAL DE CAMPINAS
CONCURSO PÚBLICO PARA PROVIMENTO DE CARGOS

Cód. BB23

PROGRAMADOR DE MICROCOMPUTADOR JÚNIOR

CONHECIMENTOS BÁSICOS

1. O número decimal 2174 representado na base hexadecimal corresponde ao valor
 - (A) 88A
 - (B) 92A
 - (C) 87D
 - (D) 82F
 - (E) 81C
2. Um microcomputador possui memória principal com capacidade para armazenar palavras de 16 bits em cada uma de suas M células e o seu barramento de endereços tem 10 bits de tamanho. Sabendo-se que em cada célula pode-se armazenar o valor exato de uma palavra, poderemos armazenar nesta memória
 - (A) 1024 bytes
 - (B) 4098 bytes
 - (C) 8096 bytes
 - (D) 512 bytes
 - (E) 2048 bytes
3. Um microcomputador possui uma capacidade máxima de memória principal (RAM) com 32K células, cada uma capaz de armazenar uma palavra de 8 bits. Então, o máximo endereço em hexadecimal desta memória será
 - (A) 7FFF
 - (B) 8F00
 - (C) 700A
 - (D) FFFF
 - (E) FF78
4. Um microcomputador possui uma capacidade máxima de memória principal (RAM) com 16K células, cada uma capaz de armazenar uma palavra de 8 bits. Então, o tamanho do barramento de endereços do sistema é igual a
 - (A) 16 bits
 - (B) 14 bits
 - (C) 12 bits
 - (D) 10 bits
 - (E) 18 bits
5. Memórias consistem de uma certa quantidade de células ou localizações onde cada qual pode armazenar uma determinada peça de informação. Cada célula tem um número, chamado endereço, através do qual os programas podem fazer referência. Todas as células na memória têm o mesmo número de bits. Se uma memória é formada por endereços de 10 bits e células de 16 bits, então o número máximo de células diretamente endereçáveis pelos programas é
 - (A) 256
 - (B) 2048
 - (C) 512
 - (D) 1024
 - (E) 4096

6. O número binário 1111 0000 0111 0000 1111 1111 0001 0010 representado na base hexadecimal corresponde ao valor
- (A) 12 0F 70 F0
 - (B) 0F 07 F0 21
 - (C) F0 8F F1 10
 - (D) F0 70 FF 12
 - (E) F1 10 8F 11
7. No modelo relacional de dados, o operador relacional que corresponde a um subconjunto do produto cartesiano entre duas tabelas é
- (A) projeção
 - (B) união
 - (C) junção
 - (D) seleção
 - (E) subtração
8. O modelo lógico de dados de um banco de dados relacional pode ser feito com o uso do diagrama de
- (A) Transações
 - (B) Entidade/Relacionamento
 - (C) Integridade Referencial
 - (D) Integridade de Domínio
 - (E) Eventos
9. A consistência de um banco de dados é assegurada pelo sistema gerenciador de banco de dados através do suporte
- (A) à réplica de dados
 - (B) à regras de integridade
 - (C) à clusterização dos dados
 - (D) à redundância de dados
 - (E) a transações
10. No projeto de um banco de dados relacional, as chaves candidatas que não foram escolhidas como chaves primárias são chamadas de chaves
- (A) alternadas
 - (B) estrangeiras
 - (C) desnormalizadas
 - (D) derivadas
 - (E) calculadas
11. Um programador necessita desenvolver uma aplicação com o uso de um driver ODBC. O programa em tempo de execução irá construir o comando SQL que será tratado pelo servidor de banco de dados. Podemos, então, afirmar que o programa emprega o conceito de
- (A) SQL ESTÁTICO
 - (B) SQL PARAMETRIZADO
 - (C) SQL INTERMEDIÁRIO
 - (D) SQL DINÂMICO
 - (E) SQL DISTRIBUÍDO
12. Os sistemas gerenciadores de banco de dados usualmente disponibilizam um componente chamado catálogo, que pode ser usado para auxiliar a execução do módulo
- (A) analisador sintático
 - (B) otimizador
 - (C) analisador semântico
 - (D) analisador léxico
 - (E) gerador de logs

13. Na linguagem SQL, o comando empregado para se eliminar uma tabela é

- (A) DELETE
- (B) REMOVE
- (C) ERASE
- (D) FLUSH
- (E) DROP

14. Uma preocupação usualmente encontrada nas metodologias de desenvolvimento de sistemas se refere à complexidade de um programa. Podemos associar a complexidade de um programa à medida encontrada para se compreendê-lo. Quanto mais complexo o programa, mais difícil de entendê-lo. A complexidade do programa usualmente está associada ao número de caminhos possíveis de execução e da dificuldade em se trilhar estes caminhos. Evitar complexidade desnecessária aumenta a confiabilidade do programa e reduz o esforço necessário de desenvolvimento e manutenção do programa.

A métrica de complexidade Ciclômática de McCabe é uma das medidas mais usadas e tem sido automatizada por muitas ferramentas CASE e ferramentas de reengenharia de sistemas.

Dado o trecho de programa abaixo:

```

INÍCIO
IF Condição-1 THEN
    IF Condição-2 THEN
        DO FUNCA
        DO FUNCB
    ELSE
        DO FUNCC
        DO FUNCD
    ENDIF
ELSE
    DO FUNCE
ENDIF
FIM

```

Podemos afirmar que a métrica de complexidade McCabe será

- (A) 3
- (B) 4
- (C) 5
- (D) 6
- (E) 7

15. Um programador, empregando a análise estruturada de sistemas como metodologia de desenvolvimento de sistemas, freqüentemente representa graficamente o fluxo de informações e transformações aplicadas aos dados através de um diagrama chamado diagrama de fluxo de dados. Este diagrama, dentre outras funções, auxilia o programador a

- (A) implementar o modelo de dados da aplicação
- (B) desenvolver o programa de testes da aplicação
- (C) implementar o código das funções a serem desenvolvidas na aplicação
- (D) checar o balanceamento das informações em cada nível do diagrama
- (E) implementar as estruturas de dados da aplicação

16. Usualmente encontramos, nas linguagens de programação, métodos de passagem de parâmetros os quais permitem que parâmetros sejam transmitidos entre o programa principal e os subprogramas. Dentre estes métodos, destaca-se um no qual são passadas cópias dos parâmetros envolvidos na transmissão. Esta técnica é conhecida por

- (A) passagem por referência
- (B) passagem por valor
- (C) passagem por endereço
- (D) passagem por atribuição
- (E) passagem por empilhamento

17. Na operação de um disco magnético, o tempo necessário para um cabeçote de leitura efetuar o correto posicionamento do cilindro é chamado
- (A) search
 - (B) delay
 - (C) seek
 - (D) latency
 - (E) rotational
18. Um dos algoritmos empregados nos sistemas operacionais modernos no qual o gerenciamento de processos é feito através da definição de um time-slice, é o
- (A) round-robin
 - (B) concurrent-job
 - (C) thrash-time
 - (D) pool-process
 - (E) ready-job
19. O conceito de dependência funcional é usualmente empregado num projeto de banco de dados relacional na etapa de
- (A) otimização
 - (B) definição das transações
 - (C) replicação de dados
 - (D) definição das regras de integridade
 - (E) normalização
20. Um programador necessita acessar um arquivo por meio de um índice secundário. Se o índice criado pelo programador tiver uma entrada para cada registro do arquivo de dados, então o índice será chamado
- (A) socket
 - (B) hashing
 - (C) heap
 - (D) denso
 - (E) B-tree
21. As anomalias de atualização de áreas compartilhadas em um sistema operacional podem ser resolvidas através da técnica conhecida por
- (A) thrashing
 - (B) semaphore
 - (C) spooling
 - (D) deadlock
 - (E) throughput
22. A recuperação de transações em bancos de dados distribuídos pode ser feita através de um protocolo conhecido por:
- (A) Recovery-Replic
 - (B) Rollback-Replic
 - (C) Logic-Distributed
 - (D) Two-Phase Commit
 - (E) Distributed-Fased
23. A estrutura de dados que corresponde a um array de caracteres é conhecida por
- (A) string
 - (B) pointer
 - (C) struct
 - (D) object
 - (E) construct

24. Durante a etapa de compilação de um programa, os nomes de variáveis definidas no programa são convertidos em endereços de montagem numa tabela conhecida por
- (A) tabela de montagem
 - (B) tabela de sintaxe
 - (C) tabela léxica
 - (D) tabela semântica
 - (E) tabela de símbolos
25. Os sistemas operacionais modernos empregam algoritmos de gerenciamento de memória com o objetivo de manter em execução o maior número de processos possível. A “anomalia de Belady” se torna mais presente no algoritmo
- (A) FIFO
 - (B) LRU
 - (C) FRU
 - (D) ROUND-ROBIN
 - (E) MMU
26. Assinale a alternativa incorreta:
- (A) A análise de requisitos habilita o projetista de software a especificar funções e performance do software, indicando a interface com outros elementos do software e estabelecendo regras de projeto que o software deve atender.
 - (B) A análise de requisitos é a primeira atividade técnica dentro do processo de engenharia de software e permite que o desenvolvedor construa modelos dos dados e processos de forma a representar a essência dos requisitos do software.
 - (C) Em muitos casos, não é possível na fase de análise dos requisitos do software especificar-se completamente um problema. Nesse caso, pode-se aplicar a técnica de prototipação que resulta num modelo executável do software a partir do qual os requisitos podem ser refinados.
 - (D) A fase de planejamento de software é a atividade de gerenciamento de software que combina métodos de estimativa com análise de riscos, scheduling e outras atividades de tomadas de decisão.
 - (E) documento de especificação de requisitos de software apresenta uma descrição precisa do software, uma representação do fluxo de dados e de controle da informação, critérios de validação do software, especificação das estruturas de dados e uma especificação dos processos do software através de primitivas funcionais.
27. Assinale a alternativa incorreta:
- (A) O modelo espiral, empregado no desenvolvimento de software, incorpora as melhores características das metodologias clássicas de ciclo de vida e prototipação, incluindo um novo elemento – a análise de risco, que não está presente nestes paradigmas de construção de software.
 - (B) O modelo espiral, empregado em desenvolvimento de software, é atualmente uma das mais realistas abordagens de desenvolvimento para projetos de sistema e software em larga escala. Esta metodologia emprega uma abordagem evolucionária para a engenharia de software.
 - (C) A grande vantagem do modelo espiral empregado no desenvolvimento de software se refere à facilidade em convencer grandes clientes, particularmente em situações de contrato, que a abordagem evolutiva do modelo é controlável. O modelo, ao invés de empregar um protótipo, enfatiza a capacidade técnica dos programadores alocados para a construção do software.
 - (D) O modelo espiral, empregado em desenvolvimento de software, mantém os mesmos passos sugeridos no ciclo de vida clássico, mas incorpora um ambiente interativo que mais realisticamente reflete o mundo real, demandando uma consideração direta de todos os aspectos técnicos em todos os estágios do projeto, com vistas a reduzir riscos antes de se tornarem problemáticos.
 - (E) As técnicas de “quarta-geração”, empregadas em desenvolvimento de software, correspondem a um conjunto de ferramentas que têm uma coisa em comum: Cada ferramenta habilita o projetista de software a especificar alguma característica do software a um nível mais alto.

28. Um programador foi alocado para o desenvolvimento de uma aplicação e, para tanto, estará empregando uma metodologia de desenvolvimento de sistemas conhecida por “ciclo de vida clássico”. Assinale a alternativa incorreta.

- (A) Ao utilizarmos um ambiente de desenvolvimento de software que emprega técnicas de “quarta-geração”, usualmente temos à nossa disposição algumas ou todas as seguintes ferramentas: linguagens não-procedurais para consultas a bancos de dados, geração de relatórios, manipulação de dados, interação e definição de telas, geração de código, geração de gráficos e planilhas.
- (B) O modelo de desenvolvimento baseado no ciclo de vida clássico corresponde a uma abordagem seqüencial de desenvolvimento que se inicia com uma atividade de análise e engenharia do sistema, para em seguida uma análise dos requisitos do software, projeto, codificação, teste e manutenção. Nesta metodologia, o cliente facilmente estabelece todos os requisitos de forma explícita, sendo, por esta razão, a metodologia clássica e mais empregada no processo de desenvolvimento de software.
- (C) Tendo em vista que usualmente o software é parte de um grande sistema, o trabalho de especificação se inicia com o estabelecimento de requisitos para todos os elementos do sistema para em seguida alocar uma parte destes requisitos para o software. Esta visão de sistema é essencial quando o software deve fazer interface com outros elementos, tais como: hardware, pessoas e bancos de dados.
- (D) Para entender a natureza dos programas a serem desenvolvidos, o projetista de software deve entender os requisitos do software e suas funções. Esta atividade deve ser documentada através de um documento de especificação e revisada com o cliente.
- (E) O projeto do software é um processo multi-passos que está centrado em quatro distintos atributos: estrutura de dados, arquitetura do software, detalhamento dos procedimentos e interface. O processo de projeto do software corresponde a mapear requisitos de software através da aplicação de técnicas que permitam a implementação física do sistema. Deve também ser documentado e torna-se parte da configuração do software.

29. Assinale a alternativa incorreta:

- (A) Com muito poucas exceções, o domínio corrente de aplicações que empregam técnicas de “quarta-geração” em engenharia de software é limitado a sistemas de informações orientadas a negócios, especificamente análise de informações e geração de relatórios extraídos de grandes bases de dados. Entretanto, novas ferramentas CASE estão sendo disponibilizadas para a geração automática de código para aplicações de engenharia e real-time.
- (B) O processo de desenvolvimento de software, independentemente do paradigma metodológico empregado, contém uma fase de definição do sistema onde se focaliza “o que” será construído. Esta fase é composta por três etapas: a análise do sistema, o planejamento do projeto de software e análise dos requisitos.
- (C) O processo de desenvolvimento de software, independentemente do paradigma metodológico empregado, contém uma fase de desenvolvimento do sistema onde se focaliza “como” o software será construído. Esta fase é composta por três etapas: o projeto do software, a codificação e teste do software.
- (D) O processo de desenvolvimento de software, independentemente do paradigma metodológico empregado, contém uma fase de manutenção do sistema onde se focaliza “a mudança” a ser processada no software construído. Esta fase é composta por três etapas: a correção, a adaptação e melhorias.
- (E) Ao empregarmos técnicas de “quarta-geração” em grandes projetos de software, podemos mover diretamente da fase de definição de requisitos para a fase de testes, uma vez que temos à nossa disposição uma linguagem de quarta geração. Nesse caso, não iremos necessitar de uma estratégia de projeto para o software, uma vez que um gerador de código estará presente para a atividade.

30. Num projeto de banco de dados relacional, uma coluna de uma determinada tabela só pode conter valores atômicos. Esta propriedade caracteriza a

- (A) segunda forma normal
- (B) primeira forma normal
- (C) terceira forma normal
- (D) quarta forma normal
- (E) quinta forma normal

31. Nos sistemas gerenciadores de bancos de dados, usualmente emprega-se uma técnica para se prover acesso direto a um determinado registro armazenado num arquivo, com base num valor dado para algum campo. Esta técnica se utiliza do algoritmo de
- (A) Huffman
 - (B) Heaven
 - (C) Heath
 - (D) Hashing
 - (E) Hilbert
32. Um programador necessita incorporar ao SQL as capacidades usualmente encontradas nas linguagens de programação. Então, o programador deverá trabalhar com
- (A) stored procedures
 - (B) application program interfaces
 - (C) SQL dinâmico
 - (D) triggers
 - (E) hash tables
33. A linguagem SQL é composta de duas sub-linguagens que são chamadas
- (A) ADL e AML
 - (B) SAL e SML
 - (C) DDL e DML
 - (D) DEL e SML
 - (E) LDL e LML
34. Os sistemas operacionais modernos usualmente operam com mecanismos de compartilhamento de tempo de CPU e normalmente conseguem executar vários programas diferentes de forma "simultânea". Entretanto, cada programa isoladamente pode ser constituído por diversas rotinas no qual o sistema pode também executá-las de forma concorrente. Quando este conceito se aplica, podemos afirmar que o sistema operacional é
- (A) preemptivo
 - (B) batch
 - (C) paralelo
 - (D) de tempo real
 - (E) multithreading
35. Os diagramas de estrutura de software apresentam uma medida da interconexão entre módulos. Esta medida é conhecida por
- (A) integração
 - (B) modularidade
 - (C) coesão
 - (D) encapsulamento
 - (E) acoplamento
36. Ao empregarmos o modelo relacional durante o projeto de um banco de dados, sempre que aplicamos um operador relacional numa tabela, teremos como resultado uma nova tabela. Esta propriedade é conhecida por
- (A) integridade referencial
 - (B) fechamento
 - (C) regra de dependência
 - (D) dependência funcional
 - (E) relação de dependência

37. Num sistema operacional, o gerenciamento de processos é feito através da execução de um algoritmo que irá decidir qual processo na fila será alocado para a CPU. Dentre estes algoritmos destaca-se um no qual o processo que primeiro fizer a requisição também em primeiro lugar será servido. Este algoritmo é conhecido por
- (A) FCFS
 - (B) FPFE
 - (C) FPFR
 - (D) FCFR
 - (E) FPFQ
38. Os sistemas gerenciadores de banco de dados empregam mecanismos para recuperar as transações em processamento durante a ocorrência de uma falha local ou global do sistema. Na ocorrência destas falhas, o sistema geralmente se utiliza de arquivos ou áreas de recuperação, visando que os bancos de dados das aplicações se mantenham consistentes. O procedimento de retorno de um banco de dados a uma posição consistente é denominado
- (A) code flushing
 - (B) commit
 - (C) rollback
 - (D) log
 - (E) syncpoint
39. Um programador projetou uma aplicação que irá empregar um vetor com 2048 elementos. Considerando a situação mais desfavorável de pesquisa, para efetuarmos uma pesquisa binária neste vetor, serão necessárias
- (A) 10 comparações.
 - (B) 12 comparações.
 - (C) 11 comparações.
 - (D) 13 comparações.
 - (E) 9 comparações.
40. No projeto de um banco de dados relacional, podemos afirmar que
- (A) uma relação na terceira forma normal pode ter diversas chaves estrangeiras, mas apenas uma chave candidata.
 - (B) uma relação na quarta forma normal pode ter diversas chaves candidatas, mas apenas uma chave associativa.
 - (C) uma relação na segunda forma normal pode apresentar diversas chaves primárias, mas apenas uma chave candidata.
 - (D) Uma relação na quinta forma normal pode apresentar diversas chaves secundárias, mas apenas uma chave candidata composta.
 - (E) uma relação na primeira forma normal pode apresentar diversas chaves candidatas.

CONHECIMENTOS ESPECÍFICOS

41. Dado o pseudocódigo abaixo:

```

INICIO:
{
  INTEGER i = 0, j = 0, TRAB = 1, VAR = 2;

  para (i = 0; i <= 5; i = i + 1)
  {
    var = var + 1;
    para (j = 0; j <= 5; j = j + 1)
    {
      trab = trab + 1;
    }
  }
  IMPRIME TRAB;
  IMPRIME VAR;
}
FIM:

```

Após a execução do programa acima, as variáveis TRAB e VAR terão respectivamente os valores:

- (A) 35 e 6
- (B) 36 e 7
- (C) 38 e 9
- (D) 37 e 8
- (E) 39 e 10

42. Dado o pseudocódigo abaixo:

```

INICIO:
{
  INTEGER i = 0, j = 0, TRAB = 1, VAR = 2;
  ENQUANTO (i < 10)
  {
    var = var + 2;
    i = i + 1;
    ENQUANTO (j < 10)
    {
      trab = trab + 2;
      j = j + 1;
    }
  }
  IMPRIME TRAB;
  IMPRIME VAR;
}
FIM:

```

Após a execução do programa acima, as variáveis TRAB e VAR terão, respectivamente, os valores:

- (A) 22 e 23
- (B) 21 e 22
- (C) 20 e 21
- (D) 22 e 24
- (E) 21 e 23

43. Dado o pseudocódigo abaixo e assumindo que o operador % representa o resto de uma divisão inteira:

```

INICIO:
{
  INTEGER i=0, j=0, TRAB=1, VAR=2;
  ENQUANTO (i < 5)
  {
    var= var+4;
    i = i +1;
    ENQUANTO (j < 10)
    {
      SE (j % 2 = 0)
      {
        trab = trab + 2;
      }
      j = j + 1;
    }
  }
  IMPRIME TRAB;
  IMPRIME VAR;
}
FIM:

```

Após a execução do programa acima, as variáveis TRAB e VAR terão, respectivamente, os valores:

- (A) 11 e 12
- (B) 12 e 13
- (C) 10 e 11
- (D) 12 e 14
- (E) 11 e 13

44. Dado o pseudocódigo abaixo e assumindo que o operador % representa o resto de uma divisão inteira:

```

INICIO:
{
  INTEGER i=0, j=0, TRAB=1, VAR=2;
  FAÇA
  {
    var= var+4;
    i = i +1;
    ENQUANTO (j < 10)
    {
      SE (j % 2 = 0)
      {
        trab = trab + 2;
      }
      j = j + 1;
    }
  } ENQUANTO (i <= 10);
  IMPRIME TRAB;
  IMPRIME VAR;
}
FIM:

```

Após a execução do programa acima, as variáveis TRAB e VAR terão, respectivamente, os valores:

- (A) 12 e 47
- (B) 10 e 45
- (C) 12 e 44
- (D) 11 e 46
- (E) 11 e 43

45. Um programador irá trabalhar com um arquivo desordenado contendo 200.000 registros. O arquivo será armazenado num disco com bloco de 4.096 bytes. Os registros serão de formato fixo de tamanho 200 bytes. Considerando que para se acessar um registro do arquivo será empregada a pesquisa linear e que uma operação de I/O no disco gasta 10 ms, o tempo médio necessário para se recuperar a informação de um registro, supondo-se que o mesmo se encontre no arquivo, será igual a
- (A) 10 segundos
(B) 20 segundos
(C) 5 segundos
(D) 2 segundos
(E) 4 segundos
46. Um programador necessita armazenar valores inteiros numa estrutura de dados do tipo array bidimensional com 10 linhas e 5 colunas. Admitindo-se que cada valor inteiro ocupe 2 bytes e que o endereço de memória do array seja 5000, então o endereço de memória do elemento do array que ocupa a quinta linha e primeira coluna será
- (A) 5050
(B) 5040
(C) 5044
(D) 5054
(E) 5052

47. Dado um banco de dados relacional formado pela tabela abaixo:

| PROJETO | | | |
|-------------|--------|-------|-------------|
| Cód_Projeto | Verba | Depto | Cod_Gerente |
| 10000 | 200000 | 10 | 1001 |
| 11000 | 150000 | 30 | 1002 |
| 12000 | 300000 | 20 | 1001 |
| 13000 | 249000 | 20 | 1002 |
| 14000 | 112000 | 40 | 1003 |
| 15000 | 123000 | 60 | 1004 |
| 16500 | 45000 | 60 | 1005 |
| 16900 | 345000 | 10 | 1005 |
| 17000 | 22000 | 10 | 1004 |

O comando SQL que lista todos os projetos do departamento 10 que têm verba superior a 100000 é

- (A) SELECT PROJETO WITH depto 10 AND verba > 100000
(B) SELECT cod_projeto WITH depto = 10, verba > 100000
(C) SELECT * FROM PROJETO WHERE depto = 10 AND verba > 100000
(D) SELECT cod_projeto BETWEEN depto = 10 , verba > 100000
(E) SELECT * PROJETO WITH depto = 10 ^ verba > 100000
48. Um programador irá trabalhar com um arquivo ordenado contendo 30000 registros. O arquivo será armazenado num disco com bloco de 1024 bytes. Os registros serão de formato fixo de tamanho 100 bytes. Considerando que para se acessar um registro do arquivo será empregada a pesquisa binária e que uma operação de I/O no disco gasta 100 ms, então o tempo necessário para se recuperar a informação de um registro será igual a
- (A) 1,5 segundos
(B) 1,2 segundos
(C) 2,0 segundos
(D) 1,0 segundos
(E) 3,0 segundos

49. Dado um banco de dados relacional formado pela tabela abaixo:

| EMPREGADO | | | |
|---------------|----------------|-------|-------------|
| Cód_Empregado | Nome | Depto | Cod_Gerente |
| 10000 | José da Silva | 10 | 10010 |
| 11000 | Maria Josefa | 30 | 10020 |
| 12000 | Pedro José | 20 | 10010 |
| 13000 | Joaquim Manoel | 20 | 10020 |
| 14000 | Glória Matos | 40 | 10030 |
| 15000 | Antônio Silva | 60 | 10040 |
| 16500 | Pedro Silva | 60 | 10050 |
| 16900 | Fernando Silva | 10 | 10050 |
| 17000 | Mário Monteiro | 10 | 10040 |

O comando SQL que lista todos os empregados que têm “Silva” no nome é

- (A) SELECT EMPREGADO WITH Nome WITH “Silva”
- (B) SELECT nome WITH Nome BETWEEN “Silva”
- (C) SELECT cod_empregado BETWEEN Nome WITH “#Silva#”
- (D) SELECT * FROM EMPREGADO WHERE Nome like “%Silva%”
- (E) SELECT * EMPREGADO WITH Nome “Silva”

50. Dado um banco de dados relacional formado pela tabela abaixo:

| EMPREGADO | | | |
|---------------|----------------|-------|---------|
| Cód_Empregado | Nome | Depto | Salario |
| 10000 | José da Silva | 10 | 1200 |
| 11000 | Maria Josefa | 30 | 3400 |
| 12000 | Pedro José | 20 | 2400 |
| 13000 | Joaquim Manoel | 20 | 3400 |
| 14000 | Glória Matos | 40 | 5600 |
| 15000 | Antônio Silva | 60 | 2300 |
| 16500 | Pedro Silva | 60 | 2380 |
| 16900 | Fernando Silva | 10 | 2600 |
| 17000 | Mário Monteiro | 10 | 7800 |

O comando SQL que lista o total de empregados por departamento e a média salarial por departamento é dado por

- (A) SELECT COUNT(EMPREGADO), AVERAGY(Salario) GROUP Depto
- (B) SELECT TOTAL(EMPREGADO) AND AVERAGY(Salario) AGROUPED Depto
- (C) SELECT Depto, COUNT(*), AVG(Salario) FROM Empregado GROUP BY Depto
- (D) SELECT TOT (EMPREGADO), AVG(Salario) ORDER BY Depto
- (E) SELECT * EMPREGADO MEDIA(Salario) ORDER BY Depto