

PROCESSO SELETIVO 05/2022

Área de Conhecimento: Tec. da Inf. apl. a Biblioteconomia

PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 1: O autor Mannino (2002) define o conceito e os recursos comuns dos Sistemas de Gerenciamento de Banco de Dados (SGBD), além de apresentar as arquiteturas dos mesmos. Assim, com base nesse autor, discorra sobre: (i) o conceito de SGBD e os seus recursos; (ii) independência de dados; (iii) arquitetura de três esquemas; (iv) processamento distribuído; e (v) arquitetura cliente–servidor.

Referência: MANNINO, Michael V. Projeto, desenvolvimento de aplicações e administração de banco de dados. São Paulo: McGraw-Hill, 2008.

Capítulo e páginas: Capítulo 1 -Página 6 até 11 e Páginas 14 até 17

Tópico da Ementa: Sistemas gerenciadores de bancos de dados (SGBD)

Resposta: (i) o conceito de SGBD e seus recursos -
Capítulo 1 - Página 6 até 11

1.2 Recursos dos Sistemas de Gerenciamento de Banco de Dados

1.2 Recursos dos Sistemas de Gerenciamento de Banco de Dados

sistema de gerenciamento de banco de dados (SGBD)

um conjunto de componentes que dão suporte à aquisição, disseminação, manutenção, recuperação e formatação de dados.

Um Sistema de Gerenciamento de Banco de Dados (SGBD) é um conjunto de componentes que dão suporte à criação, utilização e à manutenção de bancos de dados. Inicialmente, um SGBD proporcionava armazenamento e recuperação eficientes dos dados. Devido às exigências do mercado e à inovação dos produtos, os SGBDs evoluíram e hoje fornecem uma ampla gama de recursos para a aquisição, armazenamento, disseminação, manutenção, recuperação e formatação de dados. A evolução desses recursos tornou os SGBDs um tanto complexos. Pode-se levar anos de estudo e utilização para dominar um SGBD específico. Como os SGBDs continuam a evoluir, você deve sempre atualizar seu conhecimento.

Para dar uma idéia sobre os recursos que você encontrará em SGBDs comerciais, a Tabela 1.1 traz um resumo de um conjunto comum de recursos. A parte seguinte desta seção apresenta exemplos desses recursos. Alguns exemplos foram retirados do Microsoft Access, um SGBD para estações de trabalho bastante conhecido. Os capítulos seguintes se aprofundam com base na introdução aqui fornecida.

Link - Minha Biblioteca Minha Biblioteca: <https://app.minhabiblioteca.com.br/reader/books/9788580553635/pageid/29>

Resumo dos Recursos

TABELA 1.1
Resumo dos Recursos
Comuns dos SGBDs

Recurso	Descrição
Definição de banco de dados	Linguagem e ferramentas gráficas para definir entidades, relacionamentos, restrições de integridade e direitos de autorização
Acesso não-procedural	Linguagem e ferramentas gráficas para acessar dados sem codificações complexas
Desenvolvimento de aplicativo	Ferramentas gráficas para desenvolver menus, formulários de entrada de dados e relatórios; requisitos de dados para formulários e relatórios são especificados usando o acesso não-procedural
Interface de linguagem procedural	Linguagem que combina o acesso não-procedural com plenas capacidades de uma linguagem de programação
Processamento de transação	Mecanismos de controle para impedir a interferência de usuários simultâneos e recuperar dados perdidos depois de uma falha
Ajuste do banco de dados (<i>tuning</i>)	Ferramentas para controlar e melhorar o desempenho do banco de dados

(concluído!!!)

(concluído!!!)

Recursos

1.2.1 Definição de Banco de Dados

1.2.1 Definição de Banco de Dados

Para definir um banco de dados, as entidades e relacionamentos devem estar especificados. Na maioria dos SGBDs comerciais, as tabelas armazenam coleções de entidades. Uma tabela (Figura 1.4) tem uma linha de cabeçalho (primeira linha) mostrando os nomes das colunas e um corpo (outras linhas) mostrando o conteúdo da tabela. Os relacionamentos indicam as ligações entre as tabelas. Por exemplo, o relacionamento ligando a tabela de alunos à tabela de matrícula mostra os cursos oferecidos feitos por aluno.

A maioria dos SGBDs fornece várias ferramentas para definir bancos de dados. A SQL (Structured Query Language – Linguagem de Consulta Estruturada) é uma linguagem padrão da indústria suportada pela maioria dos SGBDs. A SQL pode ser usada para definir tabelas, relacionamentos entre tabelas, restrições de integridade (regras que definem os dados permitidos) e direitos de autorização (regras que restringem o acesso aos dados). O Capítulo 3 descreve as instruções de SQL para definir tabelas e relacionamentos.

Além da SQL, muitos SGBDs fornecem ferramentas gráficas, orientadas por janelas. As figuras 1.5 e 1.6 mostram ferramentas gráficas para definir tabelas e relacionamentos. Usando a janela de Definição de Tabela, da Figura 1.5, o usuário pode definir propriedades

Tabela

um arranjo de dados bidimensionais identificado. Uma tabela consiste em um cabeçalho e um corpo.

SQL

uma linguagem de banco de dados padrão da indústria que inclui instruções para a definição de banco de dados, a manipulação da base de dados e o controle do banco de dados.

Link - Minha Biblioteca Minha Biblioteca:

<https://app.minhabiblioteca.com.br/reader/books/9788580553635/pageid/29>

<https://app.minhabiblioteca.com.br/reader/books/9788580553635/pageid/30>

1.2.2 Acesso Não-procedural

linguagem de banco de dados não-procedural

uma linguagem como a SQL que permite que você especifique qual parte do banco de dados acessar em vez de codificar um procedimento complexo. Linguagens não-procedurais não incluem instruções de repetição.

1.2.2 Acesso Não-procedural

O recurso mais importante de um SGBD é a capacidade de responder a consultas. Uma consulta é uma requisição aos dados para responder a uma pergunta. Por exemplo, o usuário pode querer saber quais são os clientes com contas altas ou os produtos com boas vendas em uma determinada região. O acesso **não-procedural** permite que usuários com habilidades computacionais limitadas submetam consultas. O usuário especifica as partes de um banco de dados a serem recuperadas, não os detalhes de implementação de como uma recuperação ocorre. Os detalhes de implementação envolvem procedimentos de codificação complexos com laços. As linguagens não-procedurais não têm instruções de repetição (*for*, *while* e assim por diante) porque somente as partes de um banco de dados a recuperar são especificadas.

O acesso não-procedural pode reduzir o número de linhas de código por um fator de 100 em comparação com o acesso procedural. Como uma grande parte dos softwares de negócio envolve o acesso aos dados, o acesso não-procedural pode proporcionar uma melhora substancial na produtividade do software.

Para entender a importância do acesso não-procedural, pense na seguinte analogia: planejar suas férias. Você define o seu destino, o orçamento da viagem, a duração da estada e a data de partida. Esses fatos indicam "o que" da sua viagem. Para especificar o "como" de sua viagem, você necessita indicar muito mais detalhes, tais como a melhor rota para seu destino, a primeira opção de hotel, transporte em terra e assim por diante. Seu processo de planejamento é muito mais fácil se você tiver um profissional para ajudá-lo com estes detalhes adicionais. Como um profissional de planejamento, um SGBD executa o planejamento detalhado para responder a consultas expressas em uma linguagem não-procedural.

A maioria dos SGBDs fornece mais de uma ferramenta para o acesso não-procedural. A instrução SELECT da SQL, descrita no Capítulo 4, é um meio não-procedural para acessar um banco de dados. Muitos SGBDs também fornecem ferramentas gráficas para acessar bancos de dados. A Figura 1.7 mostra uma ferramenta gráfica disponível no Microsoft Access. Para propor uma consulta ao banco de dados, um usuário apenas tem que indicar as tabelas necessárias, os relacionamentos e as colunas. O Access se responsabiliza por gerar o plano para recuperar os dados solicitados. A Figura 1.8 mostra o resultado da execução da consulta na Figura 1.7.

Link - Minha Biblioteca Minha Biblioteca:

<https://app.minhabiblioteca.com.br/reader/books/9788580553635/pageid/31>

1.2.3 Desenvolvimento de Aplicativos e Interface de Linguagem Procedural

1.2.3 Desenvolvimento de Aplicativos e Interface de Linguagem Procedural

A maioria dos SGBDs vai muito além de simplesmente acessar dados. Ferramentas gráficas são fornecidas para construir aplicativos completos usando formulários e relatórios. Os **formulários** de entrada de dados fornecem uma ferramenta conveniente para entrar e editar dados, enquanto os **relatórios** melhoram a aparência dos dados que são exibidos ou impressos. O formulário na Figura 1.9 pode ser usado para adicionar novas atribuições de cursos para um professor e mudar as atribuições existentes. O relatório na Figura 1.10 usa a indentação para mostrar os cursos lecionados pelos professores em vários departamentos. O estilo com indentação pode ser mais fácil de visualizar do que o estilo tabular mostrado na Figura 1.8. Muitos formulários e relatórios podem ser desenvolvidos com uma ferramenta gráfica sem codificação detalhada. Por exemplo, as figuras 1.9 e 1.10 foram desenvolvidas sem codificação. O Capítulo 10 descreve os conceitos fundamentais para o desenvolvimento de formulários e relatórios.

O acesso não-procedural possibilita a criação de formulários e relatórios sem codificação extensa. Como parte da criação de um formulário ou relatório, o usuário indica os requisitos de dados usando uma linguagem não-procedural (SQL) ou ferramenta gráfica. Para completar uma definição de formulário ou relatório, o usuário indica a formatação dos dados, a interação do usuário e outros detalhes.

Além das ferramentas de desenvolvimento de aplicativos, uma interface de **linguagem procedural** adiciona as capacidades completas de uma linguagem de programação de computador. O acesso não-procedural e as ferramentas de desenvolvimento de aplicativos, embora convenientes e poderosos, às vezes não são eficientes o bastante ou não fornecem o nível de controle necessário para o desenvolvimento de aplicativos. Quando tais ferramentas não são adequadas, os SGBDs disponibilizam as capacidades completas de uma linguagem de

interface de linguagem procedural

um método para combinar uma linguagem não-procedural como a SQL com uma linguagem de programação como COBOL ou Visual Basic.

Link - Minha Biblioteca Minha Biblioteca: <https://app.minhabiblioteca.com.br/reader/books/9788580553635/pageid/32>

1.2.4 Recursos de Suporte às Operações de Banco de Dados

processamento de transações

processamento eficiente e confiável de grandes volumes de trabalho repetitivo. Os SGBDs asseguram que usuários simultâneos não interferem uns com os outros e que falhas não causam trabalho perdido.

1.2.4 Recursos de Suporte às Operações de Banco de Dados

O processamento de transações possibilita que um SGBD processe grandes volumes de trabalho repetitivo. Uma transação é uma unidade de trabalho que deve ser processada de modo confiável sem interferência de outros usuários e sem perda de dados devido a falhas. Exemplos de transações: retirar dinheiro em um caixa eletrônico, fazer uma reserva em uma companhia aérea e matricular-se em um curso. Um SGBD assegura que as transações sejam livres de interferência de outros usuários, que partes de uma transação não sejam perdidas devido a falhas e que as transações não tornem o banco de dados inconsistente. O processamento de transações é em grande parte um caso de "bastidores". O usuário não sabe os detalhes sobre o processamento de transações além das garantias em relação à confiabilidade.

O ajuste do banco de dados inclui um número de monitores e programas utilitários para melhorar o desempenho. Alguns SGBDs podem controlar como um banco de dados é usado, a

distribuição de várias partes de um banco de dados e o crescimento do banco de dados. Os programas utilitários podem ser disponibilizados para reorganizar um banco de dados, selecionar estruturas físicas para melhor desempenho e reparar partes danificadas de um banco de dados.

O processamento de transações e o ajuste do banco de dados são mais proeminentes em SGBDs que dão suporte a grandes bancos de dados com muitos usuários simultâneos. Esses SGBDs são conhecidos como SGBDs corporativos porque os bancos de dados suportados freqüentemente são fundamentais para o funcionamento de uma organização. SGBDs corporativos normalmente rodam em servidores poderosos e têm um alto custo. Por sua vez, SGBDs para estações de trabalho rodando em computadores pessoais e servidores pequenos têm recursos limitados de processamento de transações, todavia tem um custo muito mais baixo. Os SGBDs para estações de trabalho dão suporte a bancos de dados utilizados por equipes de trabalhos e pequenas empresas. SGBDs embutidos são uma categoria emergente de software de banco de dados. Como seu nome já diz, um SGBD embutido reside em um sistema maior, ou um aplicativo ou um aparelho como um assistente digital pessoal (PDA) ou um *smart card*. Os SGBDs embutidos têm recursos limitados de processamento de transações, mas têm exigências mínimas de memória, processamento e armazenamento.

Link - Minha Biblioteca Minha Biblioteca:<https://app.minhabiblioteca.com.br/reader/books/9788580553635/pageid/33>
<https://app.minhabiblioteca.com.br/reader/books/9788580553635/pageid/34>

1.2.5 Recursos de Terceiros

1.2.5 Recursos de Terceiros

Além dos recursos oferecidos diretamente pelos fornecedores de SGBDs, softwares independentes também estão disponíveis para muitos SGBDs. Na maioria dos casos, softwares independentes estendem os recursos já disponíveis no software de banco de dados. Por exemplo, muitos fornecedores independentes proporcionam ferramentas avançadas de projeto de banco de dados que estendem as capacidades de definição e ajuste do banco de dados fornecidas pelos SGBDs. A Figura 1.11 mostra um diagrama de banco de dados (um diagrama entidade-relacionamento) criado com o Visio Professional, uma ferramenta para projeto de banco de dados. O DER na Figura 1.11 pode ser transformado em tabelas com suporte da maioria dos SGBDs comerciais. Em alguns casos, softwares independentes competem diretamente com os produtos de banco de dados. Por exemplo, fornecedores de software independentes têm disponíveis ferramentas de desenvolvimento de aplicativos que podem ser usadas no lugar das fornecidas com o produto de banco de dados.

Link - Minha Biblioteca Minha Biblioteca:<https://app.minhabiblioteca.com.br/reader/books/9788580553635/pageid/34>

Resposta :(ii) independência de dados; (iii) arquitetura de três esquemas

Capítulo 1 - Página 14 até 17

1.4.1 A Independência de Dados e a Arquitetura de Três Esquemas

1.4.1 A Independência de Dados e a Arquitetura de Três Esquemas

Nos SGBDs iniciais, havia uma conexão próxima entre um banco de dados e os programas de computador que acessavam o banco de dados. Essencialmente, o SGBD era considerado parte de uma linguagem de programação. Como resultado, a definição do banco de dados era parte dos programas de computação que acessavam o banco de dados. Além disso, o significado conceitual de um banco de dados não estava separado de sua implementação física em disco magnético. As definições sobre a estrutura de um banco de dados e sua implementação física estavam misturadas dentro dos programas de computação.

A estreita associação entre um banco de dados e programas relacionados levaram a problemas de manutenção de software. A manutenção de software, incluindo mudanças de requisitos, correções e aperfeiçoamentos, pode consumir uma grande porcentagem dos orçamentos de desenvolvimento de software. Nos SGBDs iniciais, a maioria das mudanças na definição do banco de dados causava alterações nos programas de computador. Em muitos casos, as mudanças nos programas envolviam uma detalhada inspeção do código-fonte, um processo intensivo de mão-de-obra. Este trabalho de inspeção de código é semelhante ao trabalho de adequação ao ano 2000, em que todos os formatos de data foram mudados para quatro algarismos. O ajuste de desempenho de um banco de dados era difícil porque às vezes centenas de programas de computador tinham que ser recompilados a cada mudança. Como as mudanças de definição de banco de dados são comuns, uma grande parte dos recursos para manutenção de software era dedicada às mudanças nos bancos de dados. Alguns estudos calcularam que a porcentagem dispendida chegou a 50% dos recursos de manutenção de software.

independência dos dados

um banco de dados deve ter uma identidade separada dos aplicativos (programas de computador, formulários e relatórios) que o usam. A identidade separada permite que a definição de banco de dados seja alterada sem afetar os aplicativos relacionados.

arquitetura de três esquemas

uma arquitetura para compartimentalizar as descrições de banco de dados. A Arquitetura de Três Esquemas foi proposta como uma maneira de alcançar independência de dados.

O conceito de independência de dados surgiu para aliviar problemas com a manutenção de programas. Independência de dados significa que um banco de dados deve ter uma identidade separada dos aplicativos (programas de computador, formulários e relatórios) que o utilizam. A identidade separada permite que a definição de banco de dados seja mudada sem afetar os aplicativos relacionados. Por exemplo, se uma nova coluna é adicionada a uma tabela, os aplicativos que não usam a nova coluna não devem ser afetados. Da mesma maneira, se uma nova tabela é adicionada, somente os aplicativos que necessitam da nova tabela devem ser afetados. Esta separação deve ser ainda mais pronunciada se uma mudança só afeta a implementação física de um banco de dados. Os especialistas em bancos de dados devem ser livres para fazer experiência com o ajuste do desempenho sem se preocupar com as mudanças nos programas de computador.

Em meados da década de 1970, o conceito de independência de dados levou à proposta da Arquitetura de Três Esquemas representada na Figura 1.12. A palavra esquema conforme aplicada a bancos de dados significa descrição de banco de dados. A Arquitetura de Três Esquemas inclui três níveis de descrição de banco de dados. O nível externo é o nível do usuário. Cada grupo de usuários pode ter uma visão externa separada (ou visão, para resumir) de um banco de dados customizado para as necessidades específicas do grupo.

Por sua vez, os esquemas interno e conceitual representam o banco de dados inteiro. O esquema conceitual define as entidades e os relacionamentos. Para um banco de dados de uma empresa, o esquema conceitual pode ser bastante grande, talvez centenas de tipos de entidade e relacionamentos. Como o esquema conceitual, o esquema interno representa o banco de dados inteiro. No entanto, o esquema interno representa a visão do armazenamento do banco de dados, ao passo que o esquema conceitual representa o significado lógico do banco de dados. O esquema interno define arquivos, grupos de dados em um dispositivo de armazenamento como um disco rígido. Um arquivo pode armazenar uma ou mais entidades descritas no esquema conceitual.

Para tornar os três níveis do esquema mais claros, a Tabela 1.4 mostra as diferenças entre definição de banco de dados nos níveis dos três esquemas usando exemplos dos recursos descritos na Seção 1.2. Mesmo em um banco de dados simplificado de uma universidade, as diferenças entre os níveis dos esquemas são claras. Com um banco de dados mais complexo, as diferenças seriam ainda mais pronunciadas com muitas outras visões, um esquema conceitual muito maior e um esquema interno mais complexo.

Os mapeamentos de esquema descrevem como um esquema em um nível superior é derivado de um esquema em um nível inferior. Por exemplo, as visões externas na Tabela 1.4 são derivadas das tabelas no esquema conceitual. O mapeamento dá o conhecimento para converter uma requisição usando uma visão externa (por exemplo, HighGPAView) em uma

TABELA 1.4

Exemplo do Banco de Dados de uma Universidade Demonstrando as Diferenças entre os Níveis dos Esquemas

Nível do Esquema	Descrição
Externo	VisãoMédiaAlta: dados necessários para a consulta na Figura 1.7 VisãoFormulárioAtribuiçãoProfessor: dados necessários para o formulário na Figura 1.9 VisãoRelatórioCargaTrabalhoProfessor: dados necessários para o relatório na Figura 1.10
Conceitual	Tabelas e relacionamentos de Alunos, Matrículas, Cursos, Professores e Oferecimentos de Cursos (Figura 1.6)
Interno	Arquivos necessários para armazenar as tabelas; arquivos extras (propriedade indexada na Figura 1.5) para melhorar o desempenho

requisição usando as tabelas no esquema conceitual. O mapeamento entre os níveis interno e conceitual mostra como as entidades são armazenadas em arquivos.

Os SGBDs, usando esquemas e mapeamentos, asseguram a independência dos dados. Em geral, os aplicativos acessam um banco de dados usando uma visão. O SGBD converte uma requisição do aplicativo em uma requisição usando o esquema conceitual em vez da visão. O SGBD então transforma a requisição do esquema conceitual em uma requisição usando o esquema interno. A maioria das alterações no esquema conceitual ou interno não afeta os aplicativos porque os aplicativos não usam os níveis de esquema mais baixos. O SGBD, não o usuário, é responsável por usar os mapeamentos para fazer as transformações. Para mais detalhes sobre mapeamentos e transformações, o Capítulo 10 descreve visões e transformações entre os níveis conceitual e externo. O Capítulo 8 descreve a otimização de consulta, o processo de converter uma consulta em um nível conceitual em uma representação no nível interno.

A Arquitetura de Três Esquemas é um padrão oficial do American National Standards Institute (ANSI). Contudo, os detalhes específicos do padrão nunca foram amplamente adotados; o padrão serve como uma diretriz sobre como a independência dos dados pode ser alcançada. O espírito da Arquitetura de Três Esquemas é largamente implementada em SGBDs de terceira e quarta gerações.

Link - Minha Biblioteca Minha

Biblioteca: <https://app.minhabiblioteca.com.br/reader/books/9788580553635/pageid/37>

<https://app.minhabiblioteca.com.br/reader/books/9788580553635/pageid/38>

Resposta: (iv) processamento distribuído e (v) arquitetura cliente-servidor

Capítulo 1 - Página 14 até 17

1.4.2 O Processamento Distribuído e a Arquitetura Cliente-Servidor

1.4.2 O Processamento Distribuído e a Arquitetura Cliente-Servidor

Com a crescente importância da computação em rede e da Internet, o processamento distribuído está se tornando uma função crucial dos SGBDs. O processamento distribuído permite que computadores geograficamente dispersos cooperem entre si ao fornecer acesso aos dados. Uma grande parte do comércio eletrônico na Internet envolve acessar e atualizar bancos de dados remotos. Muitos bancos de dados em operações de varejo, bancárias e do mercado financeiro estão agora disponíveis pela Internet. Os SGBDs usam a capacidade disponível da rede e as capacidades de processamento local para proporcionar um acesso remoto eficiente aos bancos de dados.

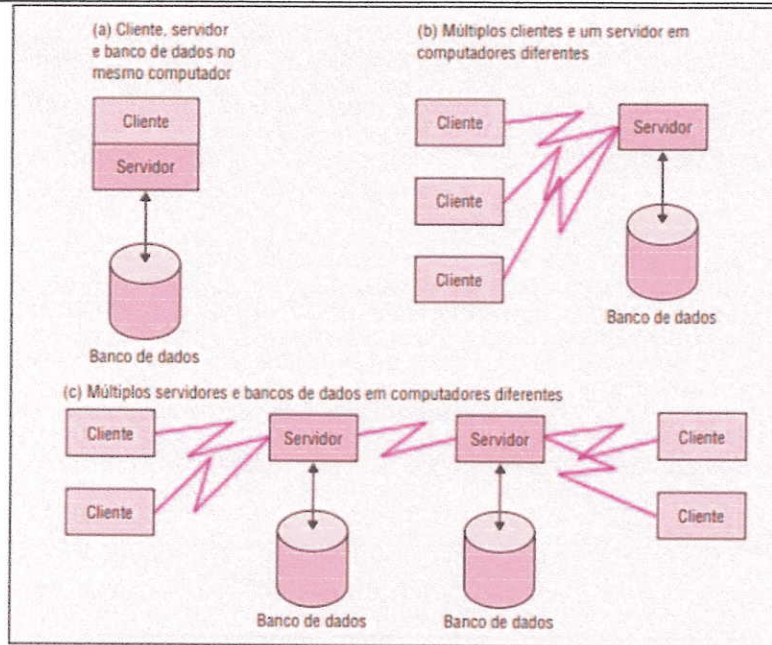
Muitos SGBDs dão suporte ao processamento distribuído utilizando uma arquitetura cliente-servidor. Um cliente é um programa que envia requisições a um servidor. Um servidor processa as requisições em nome de um cliente. Por exemplo, um cliente pode requisitar a um servidor que recupere dados de produto. O servidor localiza os dados e os envia de volta ao cliente. O cliente pode executar um processamento adicional nos dados antes de exibir os resultados ao usuário. Como outro exemplo, um cliente envia um pedido completo a um servidor. O servidor valida o pedido, atualiza um banco de dados e envia uma confirmação ao cliente. O cliente informa o usuário que a ordem foi processada.

Para melhorar o desempenho e a disponibilidade dos dados, a arquitetura cliente-servidor dá suporte a muitas maneiras de distribuir software e dados em uma rede de computadores. O esquema mais simples é colocar apenas tanto software quanto os dados no mesmo computador [Figura 1.13(a)]. Para tirar proveito de uma rede, tanto o software quanto os dados podem ser distribuídos. Na Figura 1.13(b), o software servidor e o banco de dados estão localizados em um computador remoto. Na Figura 1.13(c), o software servidor e o banco de dados estão localizados em múltiplos computadores remotos.

**arquitetura
cliente-servidor**
uma estruturação de
componentes (clientes e
servidores) e dados entre
computadores conectados
por uma rede. A arquitetura
cliente-servidor suporta
um eficiente processamento
de mensagens (requisições
de serviço) entre clientes e
servidores.

FIGURA 1.13

**Estruturas Típicas
Cliente-Servidor de
Banco de Dados e
Software**



O SGBD tem uma série de responsabilidades em uma arquitetura cliente-servidor. O SGBD fornece software que pode ser executado tanto no cliente como no servidor. O software cliente é em geral responsável por aceitar o *input* (entrada) do usuário, exibir resultados e executar algum processamento de dados. O software servidor valida as requisições do cliente, localiza bancos de dados remotos, atualiza bancos de dados remotos (se necessário) e envia os dados em um formato que o cliente entenda.

As arquiteturas cliente-servidor são uma forma flexível para os SGBDs interagirem com redes de computador. A distribuição de trabalho entre clientes e servidores e as possíveis escolhas para definir a localização de dados e software são muito mais complexas do que descrito aqui. Você aprenderá mais detalhes sobre arquiteturas cliente-servidor no Capítulo 17.

Link - Minha Biblioteca:

<https://app.minhabiblioteca.com.br/reader/books/9788580553635/pageid/39>

<https://app.minhabiblioteca.com.br/reader/books/9788580553635/pageid/40>

*O padrão de resposta deve estar fundamentado nas bibliografias exigidas pelo Edital, para evitar problemas o professor deverá citar o capítulo/página do livro utilizado.

Membros da Banca:

Avaliador 1 (nome e assinatura)

Avaliador 2 (nome e assinatura)

Avaliador 3 (nome e assinatura)

Presidente da Banca (nome e assinatura)

PROCESSO SELETIVO – _/ _/ _ _ _ _

Área de Conhecimento: _____

PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 3: Lancaster (1979) ao apresentar os fundamentos da Recuperação da Informação correlaciona processos de recuperação e de organização de informação, propondo intersecções teóricas basilares para área, aplicadas até os dias atuais. Fujita et. al. (2014) corroboram essa visão apresentando uma metodologia que relaciona processos de organização e recuperação da informação; disserta sobre esses processos de recuperação e organização da informação, enfatizando a relação entre eles.

Referência: O artigo da professora Mariângela Fujita que discute a avaliação da indexação através dos processos de recuperação da informação.

É fundamental que o professor que ensina RI saiba a correção entre os processos de organização da informação (catalogação e indexação) e os processos de recuperação da informação. Esse artigo é um excelente exemplo que pode ser mencionado pelos candidatos, especialmente das páginas 56 e 57.

A página 51 deste artigo apresenta com mais simplicidade a correção entre a organização da informação (no caso a indexação) e a recuperação da informação. O candidato que dissertar explicando essa correção poderá ter sua resposta considerada correta.

A versão online está publicada em diversos locais, e um deles é a BRAPCI: <https://brapci.inf.br/index.php/res/v/21846>

*O padrão de resposta deve estar fundamentado nas bibliografias exigidas pelo Edital, para evitar problemas o professor deverá citar o capítulo/página do livro utilizado.

Membros da Banca:

Avaliador 1 (nome e assinatura)

Avaliador 2 (nome e assinatura)

Avaliador 3 (nome e assinatura)

Presidente da Banca (nome e assinatura)

PROCESSO SELETIVO – _/ _/ _ _ _

Área de Conhecimento: _____

PROVA ESCRITA – PADRÃO DE RESPOSTA

QUESTÃO 2: A Plataforma DSpace (DURASPACE.ORG, 2022) é conhecida pela implantação de diversos tipos de protocolos e padrões tecnológicos, conjunto que oferece uma série de funcionalidades para integração e troca de dados entre diferentes plataformas, em especial o cliente OAI-PMH. Considerando os recursos desta Plataforma apresente um conjunto de procedimentos para realização de uma atividade de ensino que mostre aos estudantes um exemplo de realização de interoperabilidade de dados, a partir de um provedor de dados OAI-PMH da Plataforma OJS (você pode mencionar um provedor de dados OAI-PMH de outra plataforma, caso conheça) para um cliente OAI-PMH da Plataforma DSpace. (4.0 pontos)

A resposta esperada é uma listagem ou relação de procedimentos para uma operação de Harvesting entre duas plataformas, na forma de uma atividade de ensino em sala. Como se trata de um padrão, há possibilidade do candidato mencionar outros softwares, desde que atenda a explicação e o fundamento da interoperabilidade pelo protocolo OAI-PMH. A base da resposta está no capítulo do livro “Humanidades Digitais”, das páginas 166 a 172 (o livro está mencionado na bibliografia e pode ser acessado online https://nupill.ufsc.br/wp-content/uploads/2016/02/humanidades_digitaais_completo.pdf). Se o candidato tiver conhecimento prático na operação do DSpace (Item mencionado na Ementa, e os fundamentos técnicos constantes no manual do DSpace (também disponível na Bibliografia) poderá elaborar uma resposta mais completa.

*O padrão de resposta deve estar fundamentado nas bibliografias exigidas pelo Edital, para evitar problemas o professor deverá citar o capítulo/página do livro utilizado.

Membros da Banca:

Avaliador 1 (nome e assinatura)

Avaliador 2 (nome e assinatura)

Avaliador 3 (nome e assinatura)

Presidente da Banca (nome e assinatura)