

Aula 00

*SEFAZ-SP (Auditor Fiscal - Agente Fiscal
de Rendas - Gestão Tributária) Passo
Estratégico de Tecnologia da Informação*

Autor:
Fernando Pedrosa Lopes

26 de Junho de 2024

BANCO DE DADOS E SGBD

Sumário

Conteúdo	1
Glossário de termos	3
Roteiro de revisão	5
Introdução e Contextualização	5
Bancos de Dados	7
Tipos de SGBD (Histórico)	13
Aposta estratégica	19
Questões Estratégicas	20
Questionário de revisão e aperfeiçoamento	25
Perguntas	25
Perguntas e Respostas	26
Lista de Questões Estratégicas	30
Gabaritos	32

CONTEÚDO

Bancos de Dados. Histórico. Tipos de BD's. Sistema Gerenciador de Bancos de Dados (SGBD). Propriedades. Aplicações. Funções.

ANÁLISE ESTATÍSTICA

Inicialmente, convém destacar o percentual de incidência do assunto, dentro da disciplina Banco de Dados e Business Intelligence em concursos/cargos similares. Quanto maior o percentual de cobrança de um dado assunto, maior sua importância.

Obs.: *um mesmo assunto pode ser classificado em mais de um tópico devido à multidisciplinaridade de conteúdo.*

Assunto	Relevância na disciplina em concursos similares
SQL	21.6 %



BI (Business Intelligence)	9.0 %
DW - Data Warehouse	7.2 %
SQL Server	7.2 %
Oracle	6.3 %
Banco de Dados Multidimensionais	5.4 %
Data Mining	5.4 %
Administração de banco de dados	3.6 %
Banco de Dados	2.7 %
Formas normais	2.7 %
ETL (Extract Transform Load)	2.7 %
Banco de Dados Relacionais	2.7 %
Arquitetura de Banco de Dados	1.8 %
SGBD - Sistema de Gerenciamento de Banco de Dados	1.8 %
OLAP (On-line Analytical Processing)	1.8 %
Segurança	1.8 %
MS-Access	1.8 %
Modelo relacional	1.8 %
Metadados e Metainformação	1.8 %
Álgebra relacional	0.9 %
Banco de Dados Paralelos e Distribuídos	0.9 %
Gerência de Transações	0.9 %
Modelagem de dados	0.9 %
Gatilhos (Triggers)	0.9 %
DER - Diagrama de Entidade e Relacionamento	0.9 %
Visão (View)	0.9 %
Banco de Dados Textuais	0.9 %
Índices	0.9 %
PostgreSQL	0.9 %
MySQL	0.9 %
Big Data	0.9 %



GLOSSÁRIO DE TERMOS

Faremos uma lista de termos que são relevantes ao entendimento do assunto desta aula. Caso tenha alguma dúvida durante a leitura, esta seção pode lhe ajudar a esclarecer.

Banco de Dados: é uma coleção organizada de informações que podem ser armazenadas, gerenciadas e acessadas eletronicamente.

Tabela: é uma estrutura de dados usada em um banco de dados relacional para armazenar informações em linhas e colunas.

Linha: também conhecida como registro ou tupla, é uma entrada individual na tabela que contém informações específicas sobre um determinado objeto ou evento.

Coluna: também conhecida como campo, é uma estrutura vertical na tabela que contém um tipo de informação específico.

SGBD: Sistema de Gerenciamento de Banco de Dados é um software que permite aos usuários criar, manter e gerenciar bancos de dados.

Confiabilidade: é a capacidade de um banco de dados de operar com precisão, consistência e sem interrupções.

Disponibilidade: é a capacidade do banco de dados de estar sempre disponível para os usuários.

Integridade: é a garantia de que os dados armazenados em um banco de dados são precisos e consistentes.

Segurança: é a proteção dos dados do banco de dados contra acesso não autorizado e uso indevido.

Desempenho: é a capacidade do banco de dados de processar e retornar consultas rapidamente.

Escalabilidade: é a capacidade de aumentar a capacidade do banco de dados à medida que a demanda por informações cresce.



Flexibilidade: é a capacidade do banco de dados de se adaptar facilmente a novos requisitos de negócios e mudanças no ambiente de negócios.

SGBD Hierárquico: um Sistema de Gerenciamento de Banco de Dados (SGBD) Hierárquico é um tipo de banco de dados que organiza os dados em uma estrutura hierárquica de árvore. Nesse modelo, os dados são armazenados em registros que estão conectados por meio de relações pai-filho.

SGBD em Rede: um Sistema de Gerenciamento de Banco de Dados (SGBD) em Rede é um tipo de banco de dados que organiza os dados em uma estrutura de rede, na qual os registros podem ter múltiplos relacionamentos com outros registros. Diferente do modelo hierárquico, onde a navegação é restrita a um caminho hierárquico fixo, o modelo em rede permite que os registros sejam conectados por meio de relacionamentos complexos.

SGBD Relacional: um tipo de banco de dados que organiza os dados em tabelas relacionadas. O modelo relacional é baseado na teoria dos conjuntos e utiliza chaves primárias e chaves estrangeiras para estabelecer as relações entre as tabelas. Nesse modelo, os dados são armazenados em linhas e colunas, e as operações são realizadas por meio de consultas SQL (Structured Query Language).

SGBD Orientado a Objetos: um tipo de banco de dados que permite o armazenamento e a manipulação direta de objetos complexos, como objetos do mundo real, classes e heranças. Esse modelo estende os conceitos do modelo relacional, permitindo que objetos sejam armazenados como entidades persistentes.

SGBD NoSQL: um tipo de banco de dados que difere dos modelos tradicionais (relacional, hierárquico, em rede) em sua abordagem de armazenamento e consulta de dados. O termo "NoSQL" significa "Not Only SQL" e se refere a uma variedade de modelos de banco de dados que não seguem o esquema rígido do modelo relacional.

ROTEIRO DE REVISÃO

A ideia desta seção é apresentar um roteiro para que você realize uma revisão completa do assunto e, ao mesmo tempo, destacar aspectos do conteúdo que merecem atenção.



Introdução e Contextualização

Antes de estudarmos bancos de dados, é interessante você entender o que existia *antes* dos bancos de dados e os problemas que isso gerava.

Antes do surgimento dos sistemas gerenciadores de bancos de dados, os sistemas de arquivo eram amplamente utilizados para armazenar informações e dados em computadores. Esses sistemas eram basicamente coleções de arquivos individuais mantidos em discos rígidos, fitas magnéticas ou outros dispositivos de armazenamento. Os dados eram organizados de acordo com as necessidades do aplicativo e geralmente tinham uma estrutura hierárquica de pastas e subpastas.

No entanto, os sistemas de arquivo tinham várias limitações que os tornavam inadequados para gerenciar grandes quantidades de dados. Por exemplo, a falta de consistência e integridade de dados era um problema comum, pois não havia mecanismos para garantir que os dados fossem atualizados corretamente ou para impedir que dados duplicados fossem inseridos. Além disso, a manipulação de dados em sistemas de arquivo era muitas vezes difícil e requeriam programação personalizada, o que limitava a capacidade dos usuários em acessar e gerenciar informações.

Assim, surgiu a motivação por trás da criação de Sistemas Gerenciadores de Bancos de Dados (SGBD), a partir da necessidade de superar essas limitações e fornecer uma solução mais eficiente e flexível para gerenciar grandes quantidades de dados.



O primeiro sistema de gerenciamento de banco de dados foi o sistema de gerenciamento de arquivos (IMS), desenvolvido pela IBM na década de 1960. Esse sistema introduziu a ideia de armazenar dados em um formato estruturado e permitiu que os usuários acessassem os dados por meio de uma linguagem de consulta padronizada.

Desde então, vários sistemas gerenciadores de bancos de dados foram desenvolvidos, como o Oracle, SQL Server, MySQL, PostgreSQL, MongoDB, entre outros. Esses sistemas oferecem recursos avançados, como transações, segurança de acesso, replicação de dados, escalabilidade e suporte a vários usuários simultâneos. Eles se tornaram uma parte essencial da infraestrutura de tecnologia da informação em empresas, governos e outras organizações que precisam gerenciar grandes quantidades de dados com eficiência e segurança.

Bancos de Dados x Sistema de Arquivos

Segundo *Navathe*, são quatro, as principais características da abordagem de banco de dados que a fazem sobressair em relação às abordagens de processamento de arquivo.

Natureza auto descritiva: um banco de dados possui metadados que descrevem a estrutura e os relacionamentos dos dados armazenados, permitindo que a aplicação possa compreender como os dados estão organizados e como podem ser manipulados.

Isolamento entre programa e dados (abstração de dados): isso significa que o banco de dados é capaz de abstrair as complexidades do armazenamento e recuperação de dados do usuário, permitindo que ele possa focar na lógica da aplicação. O usuário não precisa saber como os dados estão armazenados fisicamente, apenas como acessá-los e manipulá-los.

Suporte a múltiplas visões de dados: um banco de dados pode apresentar diferentes visões dos dados para diferentes usuários ou grupos de usuários. Cada visão pode ser personalizada para atender às necessidades específicas de cada usuário ou grupo, fornecendo acesso apenas aos dados relevantes e de interesse para eles.

Compartilhamento de dados e processamento de transação multiusuário: um banco de dados permite que vários usuários acessem e manipulem os mesmos dados simultaneamente. Isso é possível por meio do controle de concorrência, que garante



que as transações sejam processadas de forma segura e consistente, mesmo em ambientes de acesso concorrente aos dados.

Bancos de Dados

Um banco de dados é um conjunto organizado de informações relacionadas que são armazenadas eletronicamente em um sistema de computador. Essas informações podem incluir dados estruturados (como números, datas e nomes) e dados não estruturados (como texto, imagens e vídeos), e são organizadas de maneira que possam ser facilmente acessadas, manipuladas e atualizadas pelos usuários.

Uma analogia comum para explicar o conceito de um banco de dados é imaginar uma biblioteca. Em uma biblioteca, os livros são organizados em prateleiras de acordo com sua categoria e gênero, como ficção, não ficção, ciência, história, etc. Cada livro possui informações, como título, autor, editora, ano de publicação, entre outras, e essas informações são armazenadas em um catálogo. Os usuários da biblioteca podem acessar o catálogo para encontrar o livro que desejam, verificar se ele está disponível e onde está localizado na biblioteca.

Da mesma forma, em um banco de dados, as informações são organizadas em tabelas que representam entidades (como clientes, produtos, pedidos, etc.), e cada tabela contém colunas que representam os atributos (como nome, endereço, telefone, preço, etc.). As tabelas são relacionadas entre si por meio de chaves estrangeiras, criando um modelo de dados que reflete a estrutura e os relacionamentos dos dados no mundo real. Os usuários do banco de dados podem acessar essas informações por meio de consultas SQL (Structured Query Language), que permitem selecionar, inserir, atualizar ou excluir dados de acordo com suas necessidades.

Por exemplo, imagine uma loja que vende produtos online. O banco de dados da loja pode incluir uma tabela de clientes, uma tabela de produtos e uma tabela de pedidos. A tabela de clientes pode conter informações como nome, endereço e e-mail dos clientes. A tabela de produtos pode incluir informações como nome, descrição e preço de cada produto. A tabela de pedidos pode relacionar os clientes aos produtos que eles compraram, incluindo informações como a quantidade de cada produto, a data do pedido e o status do pagamento e entrega.

Usando consultas SQL, os usuários da loja podem recuperar informações específicas do banco de dados, como o histórico de compras de um cliente, o estoque de um



produto ou o número de pedidos em um determinado período. Essas informações podem ser usadas para melhorar a eficiência do negócio, tomar decisões estratégicas e fornecer um melhor atendimento ao cliente.

Propriedades Implícitas

Navathe cita três propriedades implícitas que contribuem para o entendimento do banco de dados.

Primeiramente, o BD representa algum aspecto do mundo real, às vezes chamado de minimundo ou de universo de discurso (*UoD – Universe of Discourse*). As mudanças no minimundo devem ser refletidas no banco de dados.

A segunda característica implícita diz que a coleção de dados é logicamente coerente com algum significado inerente. Uma variedade aleatória de dados não pode ser chamada de banco de dados. Um banco de dados pode armazenar as informações de uma empresa, uma faculdade ou um órgão do setor público. Veja que essas informações estão dentro de um contexto, sendo, logicamente coerente.

A terceira propriedade afirma que um banco de dados é construído e populado com dados para uma finalidade específica. Ele possui um grupo de usuários bem definido e algumas aplicações, previamente concebidas, sobre as quais esses usuários interessados fazem acesso aos dados. O repositório deve ter um propósito, ele tem que atender alguma necessidade de acesso ao conjunto dos dados.

Em resumo

- ✓ Um banco de dados é um conjunto organizado de informações relacionadas que são armazenadas eletronicamente em um sistema de computador.
- ✓ As informações incluem dados estruturados (como números, datas e nomes) e dados não estruturados (como texto, imagens e vídeos).
- ✓ As informações são organizadas em tabelas que representam entidades (como clientes, produtos, pedidos, etc.).
- ✓ Cada tabela contém colunas que representam os atributos (como nome, endereço, telefone, preço, etc.).
- ✓ As tabelas são relacionadas entre si por meio de chaves estrangeiras, criando um modelo de dados que reflete a estrutura e os relacionamentos dos dados no mundo real.



- ✓ Os usuários do banco de dados podem acessar essas informações por meio de consultas SQL (Structured Query Language).
- ✓ As consultas SQL permitem selecionar, inserir, atualizar ou excluir dados de acordo com as necessidades dos usuários.
- ✓ Um banco de dados é fundamental para o gerenciamento de informações em empresas, organizações e outras áreas em que a coleta, armazenamento e acesso a dados é fundamental para o sucesso do negócio.

Propriedades

A literatura sobre o assunto nos traz várias propriedades desejáveis em Bancos de Dados. Veja a seguir alguns exemplos dessas propriedades com suas respectivas descrições.

Propriedade	Descrição
Confiabilidade	Os dados armazenados no banco de dados devem ser precisos e consistentes. Os dados devem ser protegidos contra falhas de hardware e software, bem como contra acesso não autorizado.
Disponibilidade	O banco de dados deve estar disponível para uso quando necessário. Isso significa que o sistema deve estar sempre funcionando e acessível para os usuários autorizados.
Integridade	Os dados armazenados no banco de dados devem estar protegidos contra alterações não autorizadas ou indevidas. As restrições de integridade devem ser aplicadas para garantir que os dados permaneçam precisos e consistentes.
Segurança	O acesso ao banco de dados deve ser controlado e limitado a usuários autorizados. Os dados devem ser protegidos contra ameaças internas e externas, como hackers, vírus e ataques cibernéticos.
Desempenho	O banco de dados deve ser capaz de processar grandes volumes de dados rapidamente e sem atrasos. O desempenho do sistema deve ser otimizado para garantir que as consultas possam ser processadas em tempo hábil.
Escalabilidade	O banco de dados deve ser capaz de lidar com um aumento na quantidade de dados armazenados ou no número de usuários que acessam o sistema. A escalabilidade do sistema deve ser planejada e implementada para garantir que o sistema possa crescer à medida que as necessidades de negócios mudam.



Flexibilidade	O banco de dados deve ser flexível o suficiente para permitir que os dados sejam modelados de acordo com as necessidades de negócios em constante evolução. Isso significa que o modelo de dados deve ser capaz de se adaptar às mudanças nas necessidades de negócios sem comprometer a integridade ou a segurança dos dados.
----------------------	---

Bancos de Dados x SGBD

Banco de dados e Sistema de Gerenciamento de Banco de Dados (SGBD) são conceitos diferentes, embora estejam intimamente relacionados.

Um banco de dados, como já mencionamos, é um conjunto organizado de informações relacionadas que são armazenadas eletronicamente em um sistema de computador. Ele pode ser simplesmente uma coleção de arquivos ou tabelas em um servidor de banco de dados.

- Por outro lado, um Sistema de Gerenciamento de Banco de Dados (SGBD) é um software que permite aos usuários definir, criar, manipular e gerenciar o banco de dados. O SGBD fornece uma interface para os usuários interagirem com o banco de dados, bem como um conjunto de ferramentas para criar, manter e consultar o banco de dados.

O SGBD é responsável por gerenciar o acesso concorrente aos dados, garantir a integridade dos dados, realizar backups e restaurações, bem como garantir a segurança e a privacidade dos dados. O SGBD também fornece suporte para transações e controle de acesso, permitindo que os usuários controlem quem pode acessar quais dados.

Ou seja, um banco de dados é um repositório de informações organizado, enquanto um SGBD é um software que gerencia e controla o acesso a esse banco de dados.

Existem diversos Sistemas de Gerenciamento de Banco de Dados (SGBD) disponíveis no mercado, com diferentes recursos e funcionalidades. Alguns exemplos de SGBD são:

- Oracle: um SGBD relacional que é amplamente utilizado em grandes empresas e organizações.
- MySQL: um SGBD de código aberto, muito popular para aplicações web e sistemas de gerenciamento de conteúdo.



- PostgreSQL: um SGBD de código aberto que oferece recursos avançados para aplicações empresariais.
- Microsoft SQL Server: um SGBD relacional desenvolvido pela Microsoft, utilizado principalmente em ambientes Windows.
- MongoDB: um SGBD NoSQL que permite armazenar dados de forma flexível e escalável, muito utilizado em aplicações web modernas.
- Cassandra: um SGBD NoSQL que oferece alta escalabilidade e disponibilidade para aplicações distribuídas e em nuvem.

Cada SGBD tem suas próprias vantagens e desvantagens, e a escolha do sistema ideal dependerá das necessidades específicas de cada projeto.

Funções do SGBD

As principais funções de um SGBD incluem:

Função	Descrição
Definição do esquema	Gerencia a definição do esquema de dados, incluindo a estrutura dos dados, as restrições de integridade, as relações entre as tabelas e outras informações do banco.
Controle de redundância	Gerencia a redundância de dados para garantir que os dados sejam armazenados de forma eficiente e consistente, evitando a duplicação desnecessária de dados.
Controle de acesso	Gerencia os níveis de acesso e permissões dos usuários, garantindo a segurança e privacidade dos dados.
Integridade dos dados	Garante que os dados estejam corretos e coerentes, através de regras de integridade definidas pelo usuário ou pelo sistema.
Backup e recuperação	Permite a criação de cópias de segurança (backups) do banco de dados, bem como a recuperação do sistema em caso de falhas ou problemas.
Controle de concorrência	Gerencia o acesso simultâneo de múltiplos usuários aos dados, garantindo a consistência e integridade do banco de dados.
Gerenciamento de transações	Gerencia transações, que são operações compostas de várias ações que devem ser executadas em conjunto ou abortadas, garantindo a atomicidade, consistência, isolamento e durabilidade (ACID) das transações.
Linguagem de consulta	Oferece uma linguagem de consulta para permitir que os usuários interajam com o banco de dados e realizem operações.

Sobre propriedades ACID:



As propriedades ACID são um conjunto de características fundamentais que garantem a integridade e consistência dos dados em um sistema de banco de dados. A sigla ACID representa as seguintes propriedades:

1. **Atomicidade (Atomicity):** A atomicidade garante que uma transação seja tratada como uma unidade indivisível de trabalho. Isso significa que todas as operações em uma transação devem ser executadas com sucesso ou, caso contrário, todas as operações devem ser desfeitas (rollback), deixando o banco de dados em seu estado original. Em outras palavras, uma transação é realizada por completo ou não é realizada de forma alguma.
2. **Consistência (Consistency):** A consistência assegura que somente transações válidas e conformes às regras do banco de dados sejam aceitas. Após o término de uma transação, o banco de dados deve estar em um estado consistente, seguindo todas as restrições de integridade, regras de negócio e definições de esquema.
3. **Isolamento (Isolation):** O isolamento garante que cada transação seja executada de forma isolada, sem interferir nas outras transações em execução simultaneamente. Cada transação deve operar como se estivesse ocorrendo de forma exclusiva, sem acesso ou conhecimento das alterações realizadas por outras transações em andamento. Isso evita problemas como leituras sujas, leituras não repetíveis e escritas fantasmas.
4. **Durabilidade (Durability):** A durabilidade garante que, uma vez que uma transação tenha sido confirmada (committed), seus resultados permaneçam permanentemente armazenados e não sejam perdidos, mesmo no caso de falhas de hardware, software ou reinicializações do sistema. Após o commit de uma transação, seus efeitos devem ser persistentes e sobreviverem a qualquer evento de falha.

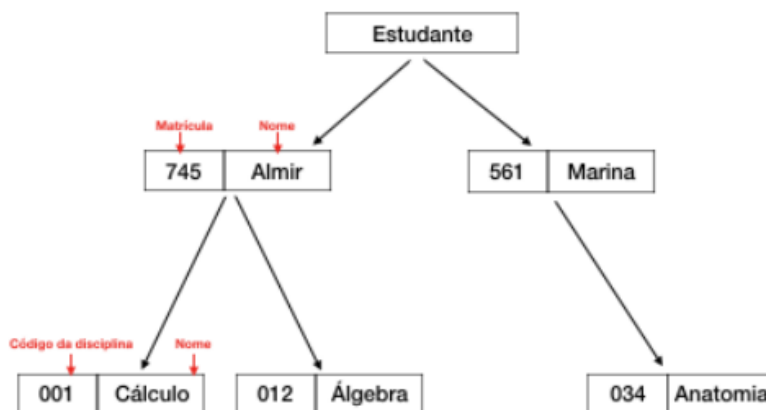
Tipos de SGBD (Histórico)

O desenvolvimento dos Sistemas de Gerenciamento de Bancos de Dados (SGBD) evoluiu ao longo do tempo, desde sistemas hierárquicos e em rede até sistemas relacionais, orientados a objetos, objeto-relacionais e NoSQL. A seguir, apresentamos um resumo cronológico das principais características e desenvolvimentos de cada tipo de SGBD:

SGBD Hierárquico (1960s)



O primeiro SGBD foi o Sistema de Gerenciamento de Banco de Dados Hierárquico (DBMS), desenvolvido na década de 1960. Este sistema usava um modelo hierárquico, onde os dados eram organizados em uma estrutura em forma de árvore, com um único nó raiz. As informações eram acessadas a partir do nó raiz, e só era possível acessar os dados filhos de um nó por meio do nó pai. Veja um exemplo:



O modelo Hierárquico foi bastante útil no início dos SGBD's, mas tinha uma série de desvantagens:

- **Rigidez:** o modelo hierárquico impõe uma estrutura rígida de organização dos dados, onde os relacionamentos são estabelecidos em uma hierarquia de pai-filho. Isso pode tornar o projeto e a manutenção do banco de dados complexos e difíceis de serem modificados.
- **Falta de flexibilidade:** devido à natureza hierárquica, consultas que envolvem relacionamentos complexos ou navegação entre diferentes caminhos da hierarquia podem ser difíceis de serem expressas e executadas no modelo hierárquico.
- **Dificuldade de escalabilidade:** a adição de novos tipos de relacionamentos ou a expansão do banco de dados pode ser complicada no modelo hierárquico, pois requer alterações na estrutura hierárquica existente.

SGBD em Rede (1970s)

Em seguida, foi desenvolvido o Sistema de Gerenciamento de Banco de Dados em Rede (DBMS) na década de 1970. Este sistema usava um modelo em rede, onde os dados eram organizados em uma estrutura mais complexa, permitindo que um registro tivesse vários pais e filhos. Isso tornava a recuperação de dados mais flexível do que no modelo hierárquico. Porém, o modelo em Rede ainda apresentava uma série de desvantagens, quais sejam:



- **Complexidade e dificuldade de compreensão:** o modelo em rede permite a definição de relacionamentos complexos entre os registros, o que pode resultar em uma estrutura de banco de dados difícil de ser compreendida e gerenciada.
- **Dependência de programação complexa:** para acessar e manipular os dados em um banco de dados em rede, é necessário o uso de uma linguagem de programação de baixo nível, como a CODASYL, o que aumenta a complexidade do desenvolvimento e da manutenção das aplicações.
- **Falta de flexibilidade e extensibilidade:** a adição ou modificação de relacionamentos requer alterações no esquema do banco de dados e na lógica de acesso aos dados, o que pode ser difícil de ser implementado e mantido.

SGBD Relacional (1980s)

O modelo relacional surgiu na década de 1980, com o desenvolvimento dos primeiros sistemas de gerenciamento de bancos de dados relacionais (RDBMS - *Relational Database Management Systems*). Esses sistemas usavam o modelo relacional, que permitia que os dados fossem organizados em tabelas. Os dados eram acessados por meio de comandos SQL (Structured Query Language), o que tornou o acesso e a manipulação dos dados mais simples e intuitivos. Este é o principal tipo de SGBD utilizado até hoje!

O sucesso do banco de dados relacional se deve a várias razões, incluindo:

Modelo Simples: o modelo relacional é baseado em tabelas, onde os dados são organizados em linhas e colunas. Essa estrutura é fácil de entender e representa uma abstração natural dos dados do mundo real. Isso torna a modelagem e o design de banco de dados mais simples e intuitivos.

Flexibilidade: o modelo permite a definição de relacionamentos entre tabelas por meio de chaves estrangeiras. Isso permite a criação de estruturas complexas de dados, representando de forma eficiente as relações entre entidades. A flexibilidade do modelo permite a modificação e expansão dos esquemas de banco de dados de forma relativamente fácil.

Integridade dos Dados: o modelo possui mecanismos embutidos para garantir a integridade dos dados. Restrições de integridade, como chaves primárias, chaves estrangeiras e restrições de unicidade, ajudam a manter a consistência dos dados armazenados no banco de dados. Isso contribui para a confiabilidade e qualidade dos dados.



Linguagem de Consulta Poderosa: a linguagem SQL (Structured Query Language) é amplamente utilizada para consultar e manipular dados em bancos de dados relacionais. O SQL oferece uma ampla gama de recursos e comandos para recuperar, inserir, atualizar e excluir dados de maneira eficiente. A simplicidade e a flexibilidade da linguagem SQL contribuíram para a popularidade do modelo relacional.

Suporte de SGBDs: o modelo relacional é suportado por uma ampla variedade de Sistemas de Gerenciamento de Banco de Dados (SGBDs) comerciais e de código aberto. Esses SGBDs fornecem recursos avançados de gerenciamento de dados, como otimização de consultas, transações, controle de acesso e segurança. Isso facilita o desenvolvimento, implantação e manutenção de sistemas baseados em banco de dados relacional.

Compatibilidade e Padronização: o modelo, hoje em dia, é um padrão estabelecido na indústria de bancos de dados. Isso significa que as aplicações e os bancos de dados desenvolvidos usando o modelo relacional são altamente compatíveis entre diferentes SGBDs. Isso permite a portabilidade dos sistemas e a interoperabilidade entre diferentes ambientes.

Em conjunto, esses fatores contribuíram para o sucesso do modelo relacional, tornando-o amplamente adotado na indústria de TI e sendo a base para a maioria dos sistemas de banco de dados utilizados atualmente.

SGBD Semânticos Orientado a Objetos e Objeto-Relacional (1990s)

Na década de 1990, surgiram os SGBDs semânticos orientados a objetos, que permitiam que objetos complexos e estruturas de dados fossem armazenados e manipulados diretamente no banco de dados. Isso tornou a manipulação de dados mais eficiente, permitindo a criação de aplicações complexas, como sistemas de gerenciamento de projetos, sistemas de informação geográfica e sistemas de gerenciamento de documentos.

Já os SGBDs semânticos objeto-relacional combinaram os modelos orientado a objetos e relacional, fornecendo recursos adicionais para armazenar e manipular objetos complexos e hierarquias de objetos dentro de um banco de dados relacional.

Estes SGBD's não são amplamente adotados quando comparados a bancos relacionais ou NoSQL.



SGBD NoSQL (2000s)

O NoSQL (Not only SQL) surgiu na década de 2000 como uma alternativa aos SGBDs relacionais tradicionais. Ele é usado para lidar com grandes volumes de dados não estruturados ou semiestruturados e, portanto, é particularmente adequado para aplicações de Big Data. Os bancos de dados NoSQL geralmente têm uma arquitetura distribuída, escalável e tolerante a falhas, e não requerem esquema rígido. Eles têm diferentes categorias, incluindo bancos de dados baseados em chave-valor, documentos, colunas e grafos. Este é o segundo tipo mais utilizado de SGBD hoje em dia.

O sucesso dos bancos de dados NoSQL pode ser atribuído a várias razões, incluindo:

Escalabilidade Horizontal: bancos NoSQL foram projetados para lidar com grandes volumes de dados e oferecem uma escalabilidade horizontal eficiente. Eles distribuem os dados em clusters de servidores, permitindo que novos servidores sejam adicionados facilmente conforme a demanda cresce. Isso possibilita lidar com cargas de trabalho intensivas e suportar aplicativos de alta escala.

Flexibilidade de Modelagem de Dados: bancos NoSQL oferecem uma estrutura flexível de dados que permite armazenar diferentes tipos de informações, como documentos, grafos, colunas amplamente distribuídas, entre outros. Isso permite que os desenvolvedores escolham o modelo de dados que melhor se adapte às necessidades específicas de suas aplicações, em vez de se restringirem a uma estrutura de tabelas fixa.

Desempenho e Velocidade: bancos NoSQL são otimizados para consultas rápidas e alto desempenho. Eles são projetados para atender às demandas de aplicativos em tempo real, lidando com grandes volumes de leitura e gravação de dados com latências mínimas. Essa capacidade de resposta rápida é especialmente benéfica em casos de uso que exigem baixa latência, como aplicativos da web, análise de dados em tempo real e processamento de transações em alta velocidade.

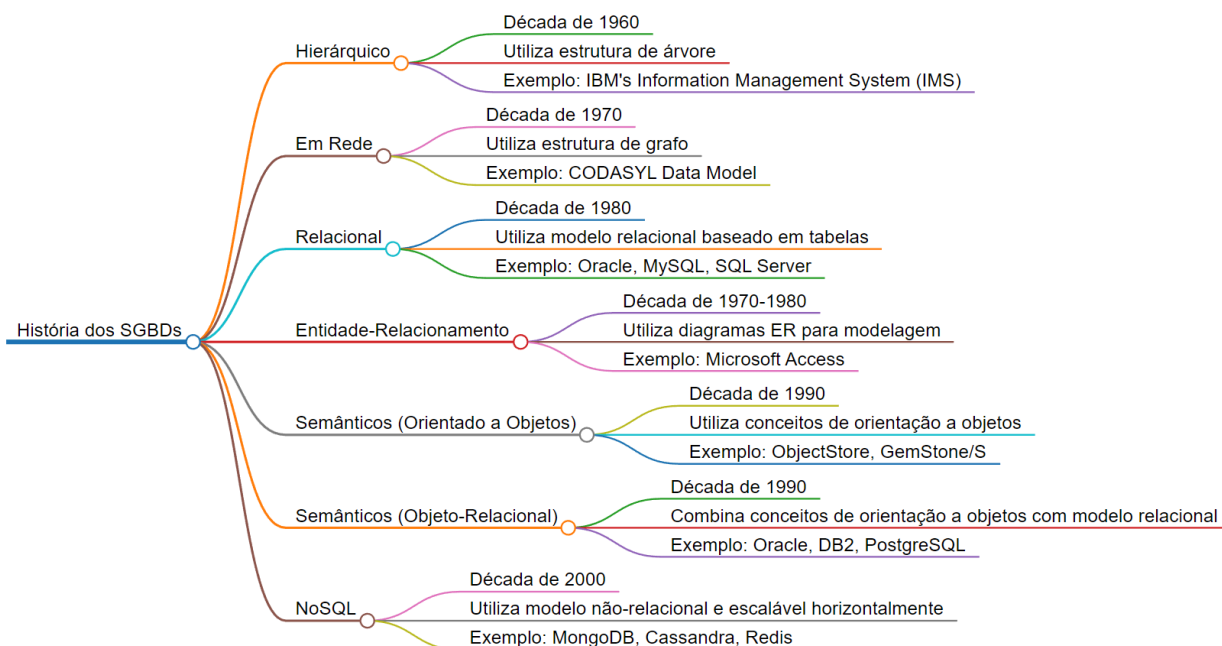
Tolerância a Falhas: bancos NoSQL são projetados para serem tolerantes a falhas, o que significa que são capazes de lidar com interrupções de hardware ou falhas de rede sem comprometer a disponibilidade e a integridade dos dados. Eles implementam técnicas como replicação de dados e distribuição geográfica para garantir a resiliência e a disponibilidade contínua dos sistemas.



Esquemas Flexíveis: bancos NoSQL permitem esquemas flexíveis, o que significa que não é necessário definir uma estrutura rígida de tabelas com antecedência. Isso permite que os dados sejam armazenados sem a necessidade de um esquema pré-determinado, facilitando a adaptação e evolução dos modelos de dados à medida que os requisitos da aplicação mudam ao longo do tempo.

Suporte para Dados Semiestruturados e Não Estruturados: Os bancos de dados NoSQL são particularmente adequados para armazenar e consultar dados semiestruturados e não estruturados, como documentos JSON, XML, dados de sensores, registros de log, mídia, entre outros. Eles fornecem recursos avançados para manipulação e consulta de dados nesses formatos, permitindo uma maior flexibilidade e capacidade de lidar com informações complexas.

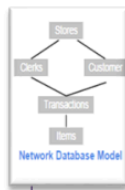
Essas características tornaram os bancos de dados NoSQL a escolha preferida para muitos aplicativos modernos, especialmente aqueles que lidam com grandes volumes de dados, necessidades de escalabilidade e requisitos de flexibilidade.





Modelos de dados hierárquicos

•1968 - Primeiro modelo de dados a ser reconhecido. Usa estruturas de árvores onde cada registro é considerado uma coleção de campos ou atributos. Exemplo: IMS.



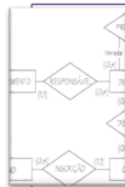
Modelo de dados em Rede

•1969 - Eliminou o conceito de hierarquia, permitindo que um mesmo registro estivesse envolvido em várias associações. Ex: IDS.

Nome do Funcionário	Preço Unitário	Comissão
Manager	120	15%
Clk	200	10%
Pusher	300	10%
Cashier	250	10%

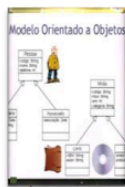
Modelo de dados relacional

•Década de 1970 - Sucessor dos modelos de dados legados. Baseia-se no conceito de relações, tuplas e atributos. Tem fundamentação na teoria dos conjuntos.
 •Década de 1980 – Os SGBDs começam a ser melhorados devido a grande aceitação do mercado.



Modelo Entidade Relacionamento.

•É um modelo conceitual de alto-nível, ou seja, é projetado para ser compreensível aos usuários comuns.
 •Formado por um conjunto de objetos chamados de **entidades** e pelo conjunto dos **relacionamentos** entre esses objetos.



Modelos de dados semântico

•Modelos de dados orientado a objetos – são mais adequados para tratamento de objetos complexos e dinâmicos.. Um objeto estará associado a um estado e um comportamento.
 •Modelo de dados objeto relacional



Modelo de dados NoSQL.

•Década de 90 – Surgem as primeiras alternativas aos modelos relacionais baseados em documentos, chave-valor e colunar.
 •Anos 2000 – As bases de dados NoSQL começam a ser reconhecidas devido ao alto poder de performance e escalabilidade

APOSTA ESTRATÉGICA

A ideia desta seção é apresentar os pontos do conteúdo que mais possuem chances de serem cobrados em prova, considerando o histórico de questões da banca em provas de nível semelhante à nossa, bem como as inovações no conteúdo, na legislação e nos entendimentos doutrinários e jurisprudenciais¹.

Em bancos de dados, a confiabilidade é uma propriedade essencial que assegura a precisão e a consistência dos dados armazenados. É vital que o sistema seja robusto o suficiente para se proteger contra falhas de hardware e software, além de se resguardar contra acessos não autorizados. A implementação de backups regulares, redundância de dados e técnicas de recuperação de desastres são práticas comuns para garantir a confiabilidade. Tais medidas protegem os dados contra perdas e garantem que, mesmo em caso de falhas, o banco de dados possa continuar operando eficientemente sem comprometer a segurança ou a integridade dos dados.

¹ Vale deixar claro que nem sempre será possível realizar uma aposta estratégica para um determinado assunto, considerando que às vezes não é viável identificar os pontos mais prováveis de serem cobrados a partir de critérios objetivos ou minimamente razoáveis.



Por outro lado, a disponibilidade refere-se à capacidade de o banco de dados estar acessível e operacional sempre que necessário. A alta disponibilidade é crucial em sistemas onde o acesso contínuo aos dados é crítico, como em aplicações financeiras ou serviços de saúde. Estratégias para alcançar alta disponibilidade incluem a utilização de clusters de servidores, balanceamento de carga e failover automático. Essas abordagens asseguram que, mesmo no caso de uma falha em um componente do sistema, o banco de dados permanece disponível sem interrupções significativas para os usuários finais.

Além disso, a integridade dos dados é fundamental para manter a confiança no banco de dados. Isso implica que os dados devem ser protegidos de alterações não autorizadas ou inadequadas. Mecanismos como restrições de integridade, controle de transações e funções de verificação de consistência são implementados para assegurar que todas as modificações nos dados sejam válidas e autorizadas. Essas medidas ajudam a manter a precisão dos dados ao longo do tempo, evitando corrupção de dados e garantindo que as operações sobre o banco de dados sejam realizadas de maneira controlada e segura.

QUESTÕES ESTRATÉGICAS

Nesta seção, apresentamos e comentamos uma amostra de questões objetivas selecionadas estrategicamente: são questões com nível de dificuldade semelhante ao que você deve esperar para a sua prova e que, em conjunto, abordam os principais pontos do assunto.

A ideia, aqui, não é que você fixe o conteúdo por meio de uma bateria extensa de questões, mas que você faça uma boa revisão global do assunto a partir de, relativamente, poucas questões.

1. (FCC / DPE-SP - 2015) As transações em bancos de dados distribuídos precisam preservar as propriedades conhecidas como ACID. Dentre estas propriedades está a:
 - a) Atomicidade.
 - b) Confidencialidade.



- c) Autenticidade.
- d) Integridade.
- e) Disponibilidade.

Comentários:

As propriedades ACID são: Atomicidade, Consistência, Isolamento e Durabilidade. A Confidencialidade, Autenticidade, Integridade e Disponibilidade são princípios fundamentais da segurança da informação e nada tem a ver com as propriedades de uma transação.

Gabarito: A

2. (FCC / DPE-RS – 2017) O dicionário de dados de um banco de dados relacional:

- a) não se aplica a tabelas com pequeno número de registros.
- b) não considera o armazenamento da lista de atributos chave das tabelas.
- c) armazena, dentre outras informações, nomes de tabelas e de seus atributos.
- d) ignora os domínios de cada atributo das tabelas.
- e) somente se aplica a tabelas com grande número de atributos.

Comentários:



A definição/descrição do banco de dados fica armazenada no próprio banco de dados na forma de um catálogo ou dicionário de dados – esses dados sobre o banco de dados são chamados de metadados. Dito isso, vamos aos itens: (a) Errado, aplica-se a tabelas de quaisquer quantidades de registros; (b) Errado, ele considera o armazenamento de todos os atributos de uma tabela – chave ou não-chave; (c) Correto, armazena diversas informações como nomes de tabelas e seus atributos; (d) Errado, ele armazena os domínios de cada atributo das tabelas; (e) Errado, aplica-se a tabelas de quaisquer quantidades de registros.

Gabarito: Letra C

3. (FCC / SABESP - 2014) Um SGBD possui a capacidade de mudar o esquema interno sem ter de alterar o esquema conceitual, conseqüentemente não havendo necessidade de alteração do esquema externo. As mudanças no esquema interno podem ser necessárias para que alguns arquivos físicos possam ser reorganizados, por exemplo, pela criação de estruturas de acesso adicionais para aperfeiçoar o desempenho da recuperação ou atualização de dados.

Essa característica de um SGBD é denominada:

- a) modelo lógico de dados.
- b) modelo físico de dados.
- c) independência modular.
- d) representação conceitual.
- e) independência física de dados.

Comentários:



A primeira frase já mata a questão: *"Um SGBD possui a capacidade de mudar o esquema interno sem ter de alterar o esquema conceitual, conseqüentemente não havendo necessidade de alteração do esquema externo"* – trata-se da independência física de dados. Vejamos:

- Independência Física dos Dados: trata da capacidade de alterar o esquema interno sem ter de alterar o esquema conceitual. Logo, os esquemas externos também não precisam ser alterados. O mapeamento nível conceitual para o interno é a chave para a independência física de dados.

Gabarito: Letra E

4. (FCC / TJ-AP – 2014) Em um determinado momento, a coleção das informações armazenadas em um banco de dados é:

- a) o esquema desse banco de dados.
- b) uma instância desse banco de dados.
- c) um metamodelo desse banco de dados.
- d) o projeto geral desse banco de dados.
- e) uma partição desse banco de dados.

Comentários:

A coleção das informações armazenadas em um banco de dados em um determinado momento é também chamada de instância de um banco de dados. Instância (ou Estado) é o conjunto de dados armazenados no banco em um determinado momento de operação. O esquema é a descrição da estrutura dos dados; a instância são os dados em si!



Gabarito: Letra B

5. (FCC / TCE-AM – 2012) O modelo conceitual de dados:

a) é aquele em que os objetos, suas características e relacionamentos têm a representação de acordo com as regras de implementação e limitantes impostos por algum tipo de tecnologia.

b) é aquele em que os objetos, suas características e relacionamentos têm a representação fiel ao ambiente observado, independente de limitações quaisquer impostas por tecnologias, técnicas de implementação ou dispositivos físicos.

c) é aquele elaborado respeitando-se e implementando-se conceitos tais como chaves de acesso, controle de chaves duplicadas, itens de repetição (arrays), normalização, ponteiros e integridade referencial, entre outros.

d) é a fase da modelagem na qual é necessário considerar todas as particularidades de implementação, principalmente o modelo que será utilizado para a implementação futura.

e) está sempre associado às fases de projeto, contrastando com o modelo lógico, que sempre está associado à fase de análise, quando utilizado com as metodologias de desenvolvimento de sistemas e implementado por ferramentas CASE.

Comentários:



(a) Errado, o modelo conceitual é independente de tecnologia; (b) Correto; (c) Errado, o modelo conceitual não trata de conceitos como chave de acesso, controle de chaves duplicadas, arrays, normalização, entre outros; (d) Errado, o modelo conceitual é um esboço, logo não precisa considerar todas as particularidades de implementação; (e) Errado, é o inverso – nas fases de análise, busca-se representar o problema e nas fases de projeto, busca-se representar a solução do problema. Logo, a fase de análise está mais ligada ao modelo conceitual enquanto a fase de projeto está mais ligada ao modelo lógico.

Gabarito: Letra B

QUESTIONÁRIO DE REVISÃO E APERFEIÇOAMENTO

A ideia do questionário é elevar o nível da sua compreensão no assunto e, ao mesmo tempo, proporcionar uma outra forma de revisão de pontos importantes do conteúdo, a partir de perguntas que exigem respostas subjetivas.

São questões um pouco mais desafiadoras, porque a redação de seu enunciado não ajuda na sua resolução, como ocorre nas clássicas questões objetivas.

O objetivo é que você realize uma auto explicação mental de alguns pontos do conteúdo, para consolidar melhor o que aprendeu ;)

Além disso, as questões objetivas, em regra, abordam pontos isolados de um dado assunto. Assim, ao resolver várias questões objetivas, o candidato acaba memorizando pontos isolados do conteúdo, mas muitas vezes acaba não entendendo como esses pontos se conectam.

Assim, no questionário, buscaremos trazer também situações que ajudem você a conectar melhor os diversos pontos do conteúdo, na medida do possível.

É importante frisar que não estamos adentrando em um nível de profundidade maior que o exigido na sua prova, mas apenas permitindo que você compreenda melhor o assunto de modo a facilitar a resolução de questões objetivas típicas de concursos, ok?

Nosso compromisso é proporcionar a você uma revisão de alto nível!



Vamos ao nosso questionário:

Perguntas

1. O que é um banco de dados?
2. O que é uma tabela em um banco de dados?
3. O que é uma coluna em uma tabela?
4. O que é uma linha em uma tabela?
5. O que é um SGBD?
6. O que é confiabilidade em um banco de dados?
7. O que é disponibilidade em um banco de dados?
8. O que é integridade em um banco de dados?
9. O que é segurança em um banco de dados?
10. O que é desempenho em um banco de dados?
11. O que é escalabilidade em um banco de dados?
12. O que é flexibilidade em um banco de dados?
13. O que é um SGBD Hierárquico?
14. O que é um SGBD em Rede?
15. O que é um SGBD Relacional?
16. O que é um SGBD Orientado a Objetos?
17. O que é um SGBD NoSQL?
18. O que é atomicidade em bancos de dados?
19. O que é consistência em bancos de dados?



20. O que é isolamento em bancos de dados?

21. O que é durabilidade em bancos de dados?

Perguntas e Respostas

1. O que é um banco de dados?

Resposta: Um banco de dados é uma coleção organizada de dados armazenados em um computador, acessível para diferentes usuários e aplicações.

2. O que é uma tabela em um banco de dados?

Resposta: Uma tabela é uma estrutura de dados que armazena informações em um banco de dados. Ela é composta por colunas (campos) e linhas (registros).

3. O que é uma coluna em uma tabela?

Resposta: Uma coluna em uma tabela representa um campo de dados específico, como nome, endereço ou idade.

4. O que é uma linha em uma tabela?

Resposta: Uma linha em uma tabela representa um registro de dados específico, contendo valores para cada uma das colunas.

5. O que é um SGBD?

Resposta: SGBD significa Sistema de Gerenciamento de Banco de Dados, e se refere a um software responsável por gerenciar o acesso, a organização e a segurança dos dados em um banco de dados.

6. O que é confiabilidade em um banco de dados?

Resposta: Confiabilidade em um banco de dados refere-se à capacidade do sistema em garantir que os dados serão consistentes e precisos, mesmo em caso de falhas ou erros.

7. O que é disponibilidade em um banco de dados?



Resposta: Disponibilidade em um banco de dados refere-se à capacidade do sistema em estar sempre acessível para os usuários, sem interrupções ou tempo de inatividade.

8. O que é integridade em um banco de dados?

Resposta: Integridade em um banco de dados refere-se à garantia de que os dados armazenados são precisos e coerentes com as regras definidas para o banco de dados.

9. O que é segurança em um banco de dados?

Resposta: Segurança em um banco de dados refere-se à proteção dos dados contra acessos não autorizados, garantindo que somente usuários autorizados possam visualizar, alterar ou excluir informações.

10. O que é desempenho em um banco de dados?

Resposta: Desempenho em um banco de dados refere-se à capacidade do sistema em processar grandes quantidades de dados em tempo hábil, sem impactar negativamente o desempenho das aplicações que acessam o banco de dados.

11. O que é escalabilidade em um banco de dados?

Resposta: Escalabilidade em um banco de dados refere-se à capacidade do sistema em lidar com um aumento no volume de dados e usuários, sem comprometer a performance e a disponibilidade.

12. O que é flexibilidade em um banco de dados?

Resposta: Flexibilidade em um banco de dados refere-se à capacidade do sistema em se adaptar a mudanças nas necessidades e requisitos de negócios, permitindo a adição ou remoção de dados e estruturas de forma simples e eficiente.

13. O que é um SGBD Hierárquico?

Resposta: Um SGBD Hierárquico é um tipo de sistema de gerenciamento de banco de dados que organiza os dados em uma estrutura hierárquica, com registros e relacionamentos pai-filho. Cada registro pode ter apenas um pai, mas pode ter vários filhos.

14. O que é um SGBD em Rede?



Resposta: Um SGBD em Rede, também conhecido como SGBD de Modelo em Rede, é um tipo de sistema de gerenciamento de banco de dados que permite relacionamentos complexos entre registros, permitindo que um registro seja conectado a vários outros registros. Ele usa uma estrutura de grafo para representar os relacionamentos entre os dados.

15. O que é um SGBD Relacional?

Resposta: Um SGBD Relacional é um tipo de sistema de gerenciamento de banco de dados que organiza os dados em tabelas com linhas e colunas, seguindo o modelo relacional. Ele usa chaves primárias e chaves estrangeiras para estabelecer relacionamentos entre as tabelas.

16. O que é um SGBD Orientado a Objetos?

Resposta: Um SGBD Orientado a Objetos é um tipo de sistema de gerenciamento de banco de dados que armazena dados como objetos, combinando dados e comportamento em uma única entidade. Ele permite o armazenamento de estruturas de dados complexas e herança entre objetos.

17. O que é um SGBD NoSQL?

Resposta: Um SGBD NoSQL, ou "Not Only SQL", é um tipo de sistema de gerenciamento de banco de dados que difere dos SGBDs tradicionais ao não seguir o modelo relacional. Os SGBDs NoSQL são projetados para lidar com grandes volumes de dados não estruturados ou semiestruturados, oferecendo alta escalabilidade e flexibilidade na modelagem de dados.

18. O que é atomicidade em bancos de dados?

Resposta: A atomicidade é uma propriedade dos bancos de dados que garante que uma transação seja tratada como uma unidade indivisível de trabalho. Isso significa que todas as operações em uma transação devem ser executadas com sucesso ou, caso contrário, todas as operações devem ser desfeitas, garantindo que o banco de dados permaneça em um estado consistente.

19. O que é consistência em bancos de dados?

Resposta: A consistência é uma propriedade dos bancos de dados que garante que apenas transações válidas e conformes às regras do banco de dados sejam aceitas. Isso significa que, após o término de uma transação, o banco de dados deve estar em um estado consistente, obedecendo às restrições de integridade definidas.



20. O que é isolamento em bancos de dados?

Resposta: O isolamento é uma propriedade dos bancos de dados que garante que cada transação seja executada de forma isolada, sem interferir nas outras transações em execução simultaneamente. Isso evita problemas como leituras sujas, leituras não repetíveis e escritas fantasmas, fornecendo uma visão consistente do banco de dados para cada transação.

21. O que é durabilidade em bancos de dados?

Resposta: A durabilidade é uma propriedade dos bancos de dados que garante que, uma vez que uma transação tenha sido confirmada, seus resultados permaneçam permanentemente armazenados e não sejam perdidos, mesmo no caso de falhas de hardware ou software. Isso é geralmente alcançado por meio da gravação dos dados em mídia persistente, como discos rígidos, garantindo que os dados sejam recuperados mesmo após uma reinicialização do sistema.

LISTA DE QUESTÕES ESTRATÉGICAS

1. (FCC / DPE-SP - 2015) As transações em bancos de dados distribuídos precisam preservar as propriedades conhecidas como ACID. Dentre estas propriedades está a:

- a) Atomicidade.
- b) Confidencialidade.
- c) Autenticidade.
- d) Integridade.
- e) Disponibilidade.

2. (FCC / DPE-RS – 2017) O dicionário de dados de um banco de dados relacional:



- a) não se aplica a tabelas com pequeno número de registros.
 - b) não considera o armazenamento da lista de atributos chave das tabelas.
 - c) armazena, dentre outras informações, nomes de tabelas e de seus atributos.
 - d) ignora os domínios de cada atributo das tabelas.
 - e) somente se aplica a tabelas com grande número de atributos.
3. (FCC / SABESP - 2014) Um SGBD possui a capacidade de mudar o esquema interno sem ter de alterar o esquema conceitual, conseqüentemente não havendo necessidade de alteração do esquema externo. As mudanças no esquema interno podem ser necessárias para que alguns arquivos físicos possam ser reorganizados, por exemplo, pela criação de estruturas de acesso adicionais para aperfeiçoar o desempenho da recuperação ou atualização de dados.

Essa característica de um SGBD é denominada:

- a) modelo lógico de dados.
 - b) modelo físico de dados.
 - c) independência modular.
 - d) representação conceitual.
 - e) independência física de dados
4. (FCC / TJ-AP – 2014) Em um determinado momento, a coleção das informações armazenadas em um banco de dados é:
- a) o esquema desse banco de dados.
 - b) uma instância desse banco de dados.



- c) um metamodelo desse banco de dados.
- d) o projeto geral desse banco de dados.
- e) uma partição desse banco de dados.

5. (FCC / TCE-AM – 2012) O modelo conceitual de dados:

a) é aquele em que os objetos, suas características e relacionamentos têm a representação de acordo com as regras de implementação e limitantes impostos por algum tipo de tecnologia.

b) é aquele em que os objetos, suas características e relacionamentos têm a representação fiel ao ambiente observado, independente de limitações quaisquer impostas por tecnologias, técnicas de implementação ou dispositivos físicos.

c) é aquele elaborado respeitando-se e implementando-se conceitos tais como chaves de acesso, controle de chaves duplicadas, itens de repetição (arrays), normalização, ponteiros e integridade referencial, entre outros.

d) é a fase da modelagem na qual é necessário considerar todas as particularidades de implementação, principalmente o modelo que será utilizado para a implementação futura.

e) está sempre associado às fases de projeto, contrastando com o modelo lógico, que sempre está associado à fase de análise, quando utilizado com as metodologias de desenvolvimento de sistemas e implementado por ferramentas CASE.



Gabaritos

1. A
2. C
3. E
4. B
5. B



ESSA LEI TODO MUNDO CONHECE: PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.



1 Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.



2 Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).



3 Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).



4 Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).



5 Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).



6 Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).



7 Concurseiro(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.



8 O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.



Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.