

**SISTEMA FIBRA**

**SENAI TAGUATINGA -DF**

**CURSO DE ADMINISTRADOR DE BANCO DE DADOS**

**Aluno: Leandro Sousa de Jesus**

1. Defina a ordem correta da criação de um banco de dados.

Análise de requisitos, modelagem conceitual, modelagem lógica, modelo físico e implementação.

1. Quais os principais tipos de atributos.

Simples, composto, multivalorado e derivado.

1. Quais as principais funções de um Administrador de Banco de dados.

As principais responsabilidades de um DBA incluem assegurar a eficiência, segurança, integridade e disponibilidade dos sistemas de banco de dados.

1. Como você define chave primaria?

A chave primária serve como um identificador exclusivo para cada registro em uma tabela, assegurando que cada linha seja única em relação às demais.

1. O que é um atributo composto e que tipo de informação é armazenado nesse campo?

Um atributo composto é aquele que pode ser dividido em atributos menores, sendo que cada um representa uma parte específica da informação. Exemplos incluem endereço, data e nome completo.

1. Qual a definição de TUPLA?

Uma tupla é uma estrutura que representa uma linha em uma tabela, formada por um conjunto ordenado de valores que correspondem aos atributos específicos da entidade modelada. Cada tupla é composta por elementos que armazenam informações sobre um único registro, sendo que cada elemento é um valor associado a um atributo definido no esquema da tabela.

1. Qual a função da chave estrangeira no Banco de Dados Relacionais?

A foreign key serve como um mecanismo para estabelecer e manter a integridade referencial entre tabelas, garantindo que os dados em uma tabela estejam consistentemente relacionados a dados em outra. Ela cria uma ligação entre uma coluna de uma tabela (chave estrangeira) e a chave primária de outra tabela, assegurando que os valores inseridos sejam válidos e correspondam a registros existentes. Isso impede a criação de registros órfãos e contribui para a consistência e a qualidade dos dados no banco.

1. Qual significado da sigla SGBD.

Sistema Gerenciador de Banco de Dados, são softwares que permitem a criação, manutenção e manipulação de bancos de dados.

1. Explique a diferença entre dados estruturados e não estruturados.

Dados estruturados são aqueles que possuem um formato claro e bem definido, organizados em tabelas compostas por linhas e colunas. Cada coluna representa um atributo, enquanto cada linha corresponde a um registro específico.  
Já os dados não estruturados não possuem uma estrutura pré-determinada, ou seja, não se encaixam em tabelas ou formatos rígidos. Esses dados podem incluir textos, imagens, vídeos, áudios ou qualquer outra forma de informação que não siga um modelo específico.

1. Qual a vantagem de se utilizar um SGBD.

O SGBD permite armazenar grandes volumes de dados de maneira organizada, facilitando o gerenciamento, a consulta e a manipulação dessas informações de forma eficiente. Ele oferece mecanismos para controlar o acesso aos dados, garantindo que apenas usuários autorizados possam realizar determinadas operações (como leitura, inserção ou alteração), aumentando a segurança das informações. O SGBD assegura que os dados sejam armazenados de maneira consistente, por meio de restrições como chaves primárias, chaves estrangeiras e outras regras de integridade, evitando dados duplicados ou inconsistentes. Com um SGBD, é possível realizar consultas complexas de forma rápida e eficiente, utilizando linguagens como SQL. Isso facilita a extração de informações valiosas para a tomada de decisões. A maioria dos SGBDs possui recursos para backup e recuperação de dados, o que garante que, em caso de falhas, as informações possam ser restauradas de forma rápida, minimizando o impacto de eventuais perdas de dados.

1. Quais ferramentas são necessárias junto ao MySql para desenvolver o modelo físico do banco de dados.

Para desenvolver o modelo físico de um banco de dados no MySQL, algumas ferramentas podem ser utilizadas para facilitar o processo de design, criação, gerenciamento e manutenção. O MySQL Workbench é a ferramenta oficial da Oracle para o MySQL, amplamente utilizada para modelagem de banco de dados, design de esquema, administração, criação de consultas SQL e gerenciamento de servidores MySQL. O PHPmyAdmin é uma ferramenta baseada na web usada para gerenciar bancos de dados MySQL. Embora não seja voltada para a modelagem do banco de dados como o MySQL Workbench, ela oferece funcionalidades para criar e administrar a estrutura física do banco de dados, como a criação de tabelas, índices e relacionamentos. O DBDesigner 4 é uma ferramenta de modelagem de banco de dados open-source que também suporta MySQL. Ela permite criar modelos físicos de bancos de dados, gerar diagramas ER e gerar scripts SQL automaticamente a partir do modelo.

1. Qual a diferença entre SGBD e SQL.

O SGBD é o software que gerencia o banco de dados, sendo responsável por armazenar, organizar, recuperar, proteger e manter a integridade dos dados. Ele fornece uma interface para os usuários e aplicações interagirem com o banco de dados, controlando o acesso, segurança, backups e transações. Já a SQL (Structured Query Language) é a linguagem utilizada para interagir com esse software, permitindo a criação, consulta, atualização e exclusão de dados. A SQL é essencial para a comunicação com o SGBD, possibilitando a execução de operações como inserção de novos registros, alteração de dados existentes, recuperação de informações específicas e definição de estruturas de banco de dados, como tabelas e índices. Assim, o SGBD e a SQL trabalham juntos para garantir que os dados sejam manipulados de maneira eficiente, segura e consistente.

1. Como era chamada a Linguagem SQL, por quem foi concebida. Cite o autor pioneiro.

A Linguagem SQL (Structured Query Language) foi inicialmente chamada de **SEQUEL** (Structured English Query Language), que foi desenvolvida na década de 1970 pela **IBM**. A SEQUEL foi criada para ser uma linguagem de consulta para o sistema de gerenciamento de banco de dados relacional (RDBMS). Os principais pesquisadores responsáveis pelo desenvolvimento da SEQUEL foram **Donald D. Chamberlin** e **Raymond F. Boyce**, ambos da IBM. O nome SEQUEL foi posteriormente alterado para SQL devido a questões de marca registrada. Chamberlin e Boyce desempenharam papéis fundamentais na concepção da linguagem, que viria a se tornar o padrão para a interação com bancos de dados relacionais. A SQL foi padronizada pela **ANSI** (American National Standards Institute) em 1986, consolidando-se como a principal linguagem utilizada em SGBDs.

1. Liste os principais tipos de dados.

Os principais tipos de dados em um banco de dados podem variar dependendo do SGBD (Sistema de Gerenciamento de Banco de Dados) utilizado, mas em geral, os mais comuns incluem:

**Números Inteiros (INTEGER)**: Usados para armazenar números inteiros, ou seja, números sem casas decimais. Exemplo: 1, -42, 1000.

**Números Reais (FLOAT, DOUBLE, DECIMAL)**: Usados para armazenar números com casas decimais. O tipo **FLOAT** é utilizado para valores de ponto flutuante de precisão simples, **DOUBLE** para precisão dupla, e **DECIMAL** ou **NUMERIC** para valores decimais com precisão exata, geralmente usados em valores financeiros.

**Caractere (CHAR)**: Usado para armazenar cadeias de caracteres de tamanho fixo. Por exemplo, **CHAR(10)** armazenaria uma string de exatamente 10 caracteres, preenchendo com espaços em branco, se necessário.

**String (VARCHAR)**: Semelhante ao tipo **CHAR**, mas a diferença é que o **VARCHAR** armazena cadeias de caracteres de tamanho variável, o que economiza espaço de armazenamento, pois não preenche com espaços em branco.

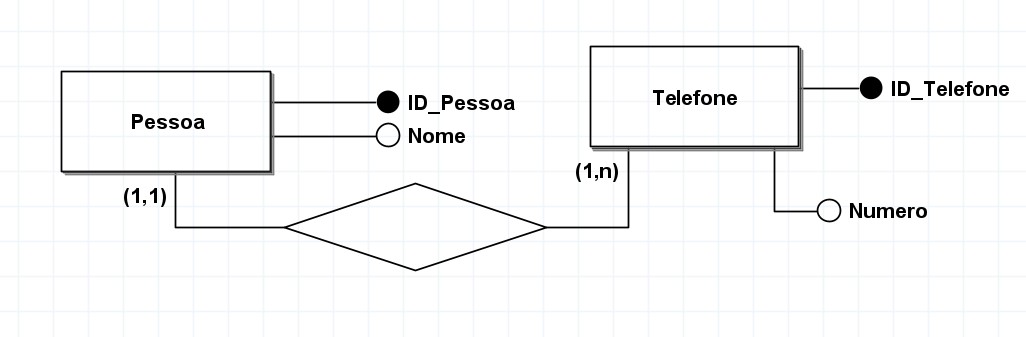
**Boolean (BOOLEAN ou BOOL)**: Usado para armazenar valores lógicos, geralmente representados como **TRUE** ou **FALSE**, para indicar condições binárias.

**Data e Hora (DATE, TIME, DATETIME, TIMESTAMP)**: Tipos usados para armazenar valores relacionados a data e hora. **DATE** armazena apenas a data (ano, mês e dia), **TIME** armazena apenas a hora, **DATETIME** armazena ambos (data e hora), e **TIMESTAMP** é usado para armazenar a data e hora com precisão de frações de segundo.

**BLOB (Binary Large Object)**: Usado para armazenar grandes volumes de dados binários, como imagens, áudios e vídeos.

**TEXT**: Usado para armazenar grandes quantidades de texto, sem uma quantidade fixa de caracteres, ideal para descrições longas, comentários e outros conteúdos textuais extensos.

1. No seguinte relacionamento, quem é entidade Fraca e quem é entidade Forte



Entidade Forte: Pessoa

Entidade Fraca: Telefone

1. Qual a diferença entre entidade Fraca e entidade Forte.

Uma **entidade forte** é aquela que possui uma existência independente, ou seja, ela pode ser identificada de forma única e não depende de nenhuma outra entidade para sua existência ou identificação. Essas entidades possuem uma **chave primária própria**, que é suficiente para distingui-las das demais no banco de dados. Exemplos típicos de entidades fortes incluem **clientes**, **produtos** e **funcionários**, que têm identificadores próprios, como um número de identificação ou código único.

Por outro lado, uma **entidade fraca** não possui uma chave primária própria e depende de uma **entidade forte** para sua identificação. Ela tem uma identificação parcial, que é composta pela chave primária da entidade forte e, frequentemente, por um atributo adicional. A dependência entre a entidade fraca e a forte é estabelecida por meio de um relacionamento de **dependência identificadora**. As entidades fracas são frequentemente usadas para representar coisas que não podem existir sem um objeto principal. Um exemplo típico de uma entidade fraca é **item de pedido**, onde a identificação do item depende do **pedido** ao qual ele pertence. Ou seja, o item não tem um identificador único sem estar associado ao pedido.

Além disso, no diagrama entidade-relacionamento (ER), a entidade fraca geralmente é representada por um **duplo retângulo**, e o relacionamento que a liga à entidade forte é mostrado com uma linha dupla. A chave primária de uma entidade fraca é composta pela chave primária da entidade forte mais um atributo parcial, chamado de **atributo discriminante**.

17. Dado as informações da estrutura das entidades, como você resolveria a seguinte situação:

Descreva:

1. O que ocorre quando tenho um Relacionamento 1:N?

Em um relacionamento 1:N (um para muitos) entre entidades em um banco de dados, uma instância de uma entidade pode estar relacionada a várias instâncias de outra entidade, mas cada instância da segunda entidade está associada a no máximo uma instância da primeira.

1. Explique os tipos de cardinalidades, e o que cada número representa:

1:1; 0:1; 1:N e 0:N.

**Um para Um (1:1)**

1:1 significa que, para cada ocorrência da entidade A, existe uma e somente uma ocorrência da entidade B. Esse tipo de relacionamento é utilizado quando a associação entre as entidades é restrita, ou seja, cada instância de A corresponde a uma única instância de B e vice-versa. Um exemplo comum seria um relacionamento entre **pessoa** e **passaporte**, onde uma pessoa tem exatamente um passaporte e cada passaporte pertence a uma única pessoa.

**Zero ou Um para Um (0:1)**

0:1 indica que a associação é opcional; a entidade A pode não ter uma correspondência na entidade B. Isso significa que, enquanto a entidade A pode ter uma instância relacionada na entidade B, ela também pode não ter nenhuma. Um exemplo seria o relacionamento entre **funcionários** e **carros corporativos**, onde um funcionário pode ou não ser associado a um carro da empresa.

**Um para Muitos (1:N)**

1:N significa que, para cada ocorrência da entidade A, podem existir várias ocorrências da entidade B. Esse é um dos relacionamentos mais comuns em bancos de dados, onde uma instância de uma entidade pode ser associada a muitas instâncias de outra. Por exemplo, em um banco de dados de **autores** e **livros**, um autor pode escrever vários livros, mas cada livro tem apenas um autor.

**Zero ou Muitos (0:N)**

0:N indica que a associação é opcional e que podem existir múltiplas ocorrências da entidade B associadas à entidade A. Ou seja, uma entidade A pode estar associada a nenhuma ou várias instâncias da entidade B. Um exemplo seria o relacionamento entre **clientes** e **pedidos**, onde um cliente pode não ter feito nenhum pedido ou pode ter feito vários pedidos ao longo do tempo.