

Introdução a



Banco de dados

Sumário

Introdução.....	2
Bancos de Dados Relacionais (RDBMS).....	3
Como modelar um banco de dados relacional.....	4
Modelo de Entidade e Relacionamento (MER).....	5
SQL.....	7
Comandos SQL.....	8
Recuperando informações - O Comando SELECT.....	9
Exercícios Resolvidos.....	10
Exercícios propostos.....	14

Introdução

Um banco de dados de computadores é uma coleção organizada de dados eletrônicos, geralmente armazenados e gerenciados por sistemas de gerenciamento de banco de dados (SGBDs). Esses dados podem ser estruturados de várias maneiras, como tabelas, documentos, gráficos, etc., e são projetados para serem facilmente acessados, gerenciados e atualizados. Os bancos de dados de computadores são amplamente utilizados em uma variedade de aplicações, desde sistemas de gerenciamento de empresas até aplicativos da web e dispositivos móveis, para armazenar e manipular grandes volumes de informações de forma eficiente e segura.

SGBD

Existem vários tipos de Sistemas de Gerenciamento de Banco de Dados (SGBDs):

- **Bancos de Dados Relacionais (RDBMS):** Esses SGBDs armazenam dados em tabelas relacionais, com linhas representando registros e colunas representando atributos. Exemplos incluem MySQL, PostgreSQL, Oracle Database e SQL Server.
- **Bancos de Dados NoSQL:** Projetados para lidar com tipos de dados não relacionais e semiestruturados, como documentos, gráficos, colunas largas, etc. Exemplos incluem MongoDB, Cassandra, Couchbase e Redis.
- **Bancos de Dados de Grafos:** Especializados em armazenar e consultar dados em forma de grafos, com nós representando entidades e arestas representando conexões entre elas. Exemplos incluem Neo4j, Amazon Neptune e ArangoDB.
- **Bancos de Dados de Séries Temporais:** Otimizados para armazenar e analisar dados de séries temporais, como dados de sensores, logs de servidor, dados financeiros, etc. Exemplos incluem InfluxDB, TimescaleDB e Prometheus.
- **Bancos de Dados de Objetos:** Projetados para armazenar objetos complexos, como documentos XML, arquivos de mídia, objetos Java, etc. Exemplos incluem db4o e ObjectDB.
- **Bancos de Dados em Memória:** Armazenam dados na memória principal do computador para acesso rápido e eficiente. Exemplos incluem Redis, Memcached e SAP HANA.
- **Sistemas de Gerenciamento de Banco de Dados Distribuídos (Distributed DBMS):** Distribuem dados em vários servidores para melhor escalabilidade e disponibilidade. Exemplos incluem Google Bigtable, Amazon DynamoDB e Apache Cassandra.

Bancos de Dados Relacionais (RDBMS)

Um RDBMS (Sistema de Gerenciamento de Banco de Dados Relacional) é um tipo de sistema de gerenciamento de banco de dados que organiza e gerencia dados de acordo com o modelo relacional. Nesse modelo, os dados são armazenados em tabelas que consistem em linhas e colunas. Cada linha representa uma entrada de dados única e é referida como um "registro" ou "tupla", enquanto as colunas representam atributos ou campos específicos dos dados.

As principais características de um RDBMS incluem:

- **Estrutura Tabular:** Os dados são organizados em tabelas relacionais, com linhas representando registros e colunas representando atributos.
- **Chaves Primárias e Estrangeiras:** Relacionamentos entre tabelas são estabelecidos usando chaves primárias e estrangeiras para garantir a integridade referencial dos dados.
- **Suporte a Consultas SQL:** RDBMS oferecem suporte à linguagem SQL (Structured Query Language) para manipular e consultar dados de forma eficiente.
- **Integridade de Dados:** Garantem a integridade dos dados por meio de restrições de integridade, como chaves primárias, chaves estrangeiras e restrições de unicidade.
- **Recursos de Transação e ACID:** Suportam transações ACID (Atomicidade, Consistência, Isolamento, Durabilidade) para garantir a consistência e a confiabilidade dos dados.
- **Desempenho e Otimização:** Oferecem recursos de otimização de consultas para melhorar o desempenho das operações de consulta e manipulação de dados.

Como modelar um banco de dados relacional

O processo de modelagem de banco de dados envolve uma série de etapas fundamentais para garantir que o sistema atenda aos requisitos de dados e funcione de forma eficiente. Desde a identificação dos requisitos e entidades, passando pela normalização e definição de relacionamentos, até a implementação e teste do modelo, cada fase visa assegurar a integridade, consistência e desempenho do banco de dados. Essa abordagem estruturada permite a criação de um sistema robusto, adaptável e alinhado às necessidades do projeto.

Identificar Requisitos: Entenda os requisitos do sistema e as necessidades de dados. Isso envolve identificar os diferentes tipos de dados que serão armazenados, as relações entre esses dados e os requisitos de integridade, como restrições de chave estrangeira e regras de negócio.

Identificar Entidades e Atributos: Identifique as entidades principais do sistema, que são os objetos sobre os quais os dados serão armazenados. Para cada entidade, identifique os atributos que descrevem as características dessas entidades.

Normalizar as Entidades: Aplique técnicas de normalização para garantir a integridade dos dados e reduzir a redundância. A normalização envolve dividir as entidades em tabelas relacionadas de forma que cada tabela tenha uma única responsabilidade e não haja repetição desnecessária de dados.

Definir Relacionamentos: Identifique os relacionamentos entre as diferentes entidades e defina as chaves estrangeiras para representar esses relacionamentos. Isso envolve determinar a cardinalidade dos relacionamentos (um para um, um para muitos, muitos para muitos) e garantir a consistência dos dados com restrições de integridade referencial.

Refinar o Modelo: Refine o modelo de banco de dados, revisando e ajustando conforme necessário para garantir que atenda aos requisitos do sistema de forma eficiente e eficaz. Isso pode envolver a adição de índices para melhorar o desempenho de consultas, a desnormalização se necessário para otimizar consultas comuns e a consideração de requisitos de escalabilidade e segurança.

Documentar o Modelo: Documente o modelo de banco de dados, incluindo diagramas de entidade-relacionamento (ER), esquemas de tabelas, definições de chaves primárias e estrangeiras, restrições de integridade e outras informações relevantes para ajudar na implementação e manutenção do banco de dados.

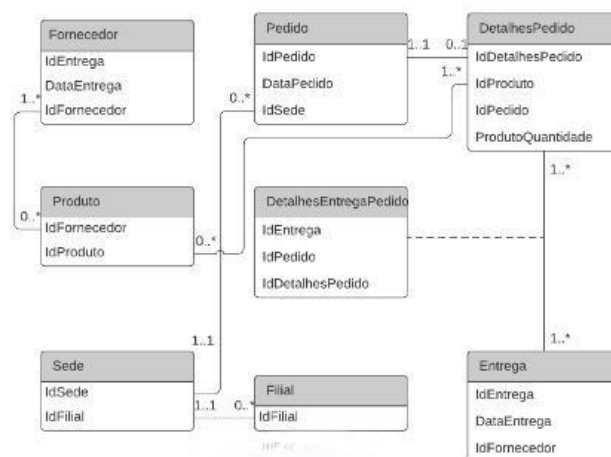
Implementar o Modelo: Implemente o modelo de banco de dados utilizando um Sistema de Gerenciamento de Banco de Dados Relacional (RDBMS) adequado, criando as tabelas, colunas, chaves primárias, chaves estrangeiras e outras estruturas conforme definido no modelo.

Testar e Iterar: Teste o banco de dados para garantir que ele atenda aos requisitos e funcione conforme esperado. Faça iterações no modelo conforme necessário para lidar com quaisquer problemas identificados durante o teste.

Modelo de Entidade e Relacionamento (MER)

Um Modelo de Entidade e Relacionamento (MER), também conhecido como Diagrama de Entidade-Relacionamento (DER), é uma representação visual dos dados e dos relacionamentos entre esses dados em um sistema de banco de dados. Ele descreve as entidades (ou objetos) dentro do sistema, juntamente com os relacionamentos entre essas entidades.

- **Entidade:** Uma entidade é um objeto ou conceito no mundo real que é representado dentro do sistema de banco de dados. Por exemplo, em um sistema de biblioteca, as entidades podem incluir "Livro", "Autor", "Cliente" e "Empréstimo". Cada entidade é representada como um retângulo no diagrama.
- **Atributo:** Um atributo é uma propriedade ou característica de uma entidade que descreve seus detalhes. Por exemplo, um livro pode ter atributos como "Título", "ISBN" e "Ano de Publicação".
- **Relacionamento:** Um relacionamento é uma associação entre duas ou mais entidades que descreve como elas estão conectadas ou interagem entre si. Por exemplo, um relacionamento entre as entidades "Livro" e "Autor" indica que um livro pode ter um ou mais autores.
- **Cardinalidade:** A cardinalidade de um relacionamento descreve o número de instâncias de uma entidade que podem estar associadas a uma instância da outra entidade. Pode ser "um para um", "um para muitos" ou "muitos para muitos".
- **Chave Primária:** Uma chave primária é um atributo (ou conjunto de atributos) que identifica exclusivamente cada instância de uma entidade em um banco de dados. Ele garante que não haja duplicatas e é usado para indexação e referência de registros.
- **Chave Estrangeira:** Uma chave estrangeira é um atributo em uma entidade que faz referência à chave primária de outra entidade. É usado para estabelecer relacionamentos entre entidades em um banco de dados.



SQL

SQL (Structured Query Language) é uma linguagem de programação utilizada para gerenciar dados em bancos de dados relacionais. É uma linguagem padronizada pela ANSI (American National Standards Institute) e pela ISO (International Organization for Standardization) e é amplamente utilizada em sistemas de gerenciamento de banco de dados relacionais (RDBMS), como MySQL, PostgreSQL, Oracle Database, SQL Server, SQLite, entre outros.

As principais funcionalidades do SQL incluem:

- **Consulta de Dados:** Permite realizar consultas para recuperar dados de uma ou mais tabelas em um banco de dados.
- **Inserção de Dados:** Permite adicionar novos registros a uma tabela em um banco de dados.
- **Atualização de Dados:** Permite modificar os dados existentes em uma tabela em um banco de dados.
- **Exclusão de Dados:** Permite remover registros de uma tabela em um banco de dados.
- **Definição de Dados:** Permite definir a estrutura de um banco de dados, incluindo a criação e alteração de tabelas, índices, restrições de integridade e outros objetos de banco de dados.
- **Controle de Acesso:** Permite controlar o acesso aos dados por meio de privilégios de usuário, como SELECT, INSERT, UPDATE, DELETE, entre outros.

Comandos SQL

Os comandos em SQL podem ser categorizados em várias classes, cada uma com seu propósito específico para interagir com um banco de dados relacional. Aqui estão os principais tipos de comandos em SQL:

DDL (Data Definition Language):

CREATE: Usado para criar objetos de banco de dados, como tabelas, índices, visões e procedimentos armazenados.

ALTER: Permite modificar a estrutura de objetos existentes no banco de dados, como adicionar ou remover colunas de uma tabela.

DROP: Remove objetos de banco de dados, como tabelas, índices ou visões.

TRUNCATE: Remove todos os registros de uma tabela, mantendo sua estrutura intacta.

DML (Data Manipulation Language):

SELECT: Recupera dados de uma ou mais tabelas.

INSERT: Adiciona novos registros a uma tabela.

UPDATE: Modifica os dados existentes em uma tabela.

DELETE: Remove registros de uma tabela.

DCL (Data Control Language):

GRANT: Concede permissões específicas a usuários ou grupos de usuários para acessar objetos de banco de dados.

REVOKE: Revoga ou remove as permissões previamente concedidas a usuários ou grupos de usuários.

TCL (Transaction Control Language):

COMMIT: Confirma as transações realizadas desde o último COMMIT ou ROLLBACK, tornando as alterações permanentes no banco de dados.

ROLLBACK: Desfaz as alterações feitas durante uma transação que ainda não foi confirmada (COMMIT).

SAVEPOINT: Define um ponto dentro de uma transação a partir do qual você pode rolar de volta as alterações.

Recuperando informações - O Comando SELECT

O comando SELECT é uma instrução fundamental em SQL usada para recuperar dados de um banco de dados. Ele permite selecionar quais colunas de uma tabela você deseja visualizar e pode ser combinado com vários modificadores para personalizar os resultados.

- **FROM:** O modificador FROM especifica a tabela da qual você deseja selecionar os dados. Por exemplo, SELECT * FROM tabela selecionará todas as colunas da tabela especificada.
- **JOIN:** O modificador JOIN é usado para combinar dados de duas ou mais tabelas com base em uma condição relacionada. Por exemplo, SELECT * FROM tabela1 JOIN tabela2 ON tabela1.coluna = tabela2.coluna combinará linhas de tabela1 com linhas correspondentes em tabela2 onde a coluna especificada é igual.
- **GROUP BY:** O modificador GROUP BY é usado para agrupar linhas que têm o mesmo valor em uma ou mais colunas e aplicar funções de agregação, como COUNT, SUM, AVG, etc., às colunas agrupadas. Por exemplo, SELECT coluna1, COUNT(coluna2) FROM tabela GROUP BY coluna1 contará o número de ocorrências únicas da coluna2 para cada valor distinto na coluna1.
- **ORDER BY:** O modificador ORDER BY é usado para classificar os resultados em ordem ascendente ou decrescente com base em uma ou mais colunas. Por exemplo, SELECT * FROM tabela ORDER BY coluna ASC ordenará os resultados da consulta em ordem crescente com base nos valores da coluna especificada.
- **COUNT():** A função COUNT() é usada para contar o número de linhas que atendem a uma determinada condição. Por exemplo, SELECT COUNT(*) FROM tabela contará o número total de linhas na tabela.

```
-- - Seleciona o nome de todos |autores brasileiros.
SELECT nome
FROM autor
WHERE nacionalidade = 'Brasil';
```

```
-- - Seleciona o título do livro e o nome de todos autores | brasileiro.
SELECT titulo, nome
FROM livro
JOIN autor ON livro.id_autor = autor.id
WHERE autor.nacionalidade = 'Brasil';
```

```
-- - Seleciona o nome do usuário e o número de empréstimos que ele fez,
-- |ordenados pelo número de empréstimos em ordem decrescente.
SELECT nome, COUNT() AS 'Número de empréstimos'
FROM usuario
JOIN emprestimo ON usuario.id = emprestimo.id_usuario
GROUP BY usuario.id
ORDER BY COUNT() DESC;
```

Exercícios Resolvidos

Modelagem

Exercício - Modele o sistema descrito abaixo:

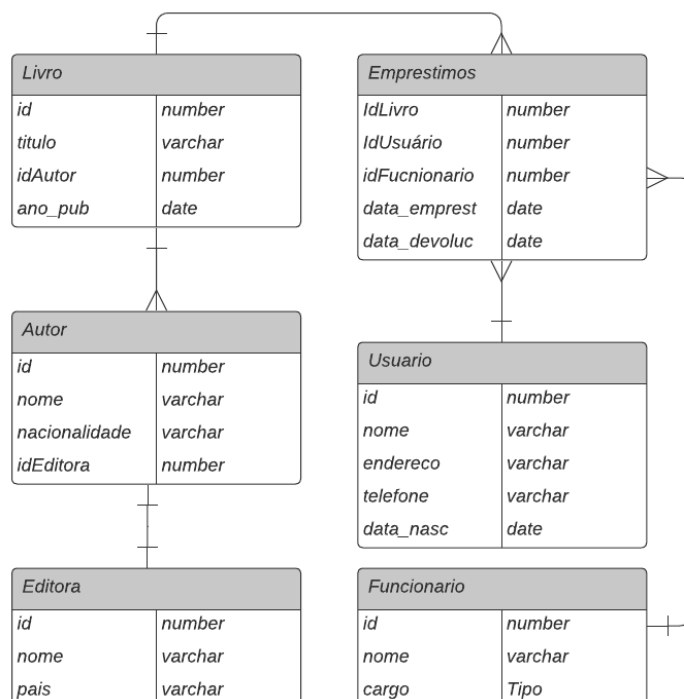
Sistema de Gerenciamento de Biblioteca: (Exercício Resolvido)

Este sistema gerencia o acervo de uma biblioteca, permitindo que os usuários façam empréstimos de livros. Os livros são escritos por um autor e publicados por editoras. Os funcionários são responsáveis pelo gerenciamento dos empréstimos, enquanto os usuários podem fazer empréstimos de livros disponíveis.

Entidades:	Relacionamentos:
Livro	Um livro ter um autor.
Autor	Um livro é publicado por uma editora.
Editora	Um livro pode ser emprestado a um usuário.
Empréstimo	Um funcionário pode gerenciar empréstimos.
Usuário	Um usuário pode fazer empréstimos de livros.
Funcionário	

Sistema de Gerenciamento de Biblioteca

Sebastião Gazolla Jr | January 28, 2024



Criando o Banco de dados

Exercício - Gere um script para a criação do banco de dados:

```
1 CREATE TABLE autor (  
2     id INT PRIMARY KEY,  
3     nome VARCHAR(255) NOT NULL,  
4     nacionalidade VARCHAR(255) NOT NULL,  
5     id_editora INT NOT NULL,  
6     FOREIGN KEY (id_editora) REFERENCES editora(id_editora)  
7 );  
8  
9 CREATE TABLE editora (  
10     id INT PRIMARY KEY,  
11     nome VARCHAR(255) NOT NULL,  
12     pais VARCHAR(255) NOT NULL  
13 );  
14  
15 CREATE TABLE funcionario (  
16     id INT PRIMARY KEY,  
17     nome VARCHAR(255) NOT NULL,  
18     cargo VARCHAR(255) NOT NULL  
19 );  
20  
21 CREATE TABLE livro (  
22     id INT PRIMARY KEY,  
23     titulo VARCHAR(255) NOT NULL,  
24     ano_pub DATE NOT NULL,  
25     id_autor INT NOT NULL,  
26     id_editora INT NOT NULL,  
27     FOREIGN KEY (id_autor) REFERENCES autor(id_autor),  
28     FOREIGN KEY (id_editora) REFERENCES editora(id_editora)  
29 );  
30  
31 CREATE TABLE usuario (  
32     id INT PRIMARY KEY,  
33     nome VARCHAR(255) NOT NULL,  
34     endereco VARCHAR(255) NOT NULL,  
35     telefone VARCHAR(255) NOT NULL,  
36     data_nasc DATE NOT NULL  
37 );  
38  
39 CREATE TABLE emprestimo (  
40     id INT PRIMARY KEY,  
41     id_livro INT NOT NULL,  
42     id_usuario INT NOT NULL,  
43     id_funcionario INT NOT NULL,  
44     data_emprestimo DATE NOT NULL,  
45     data_devolucao DATE NOT NULL,  
46     FOREIGN KEY (id_livro) REFERENCES livro(id_livro),  
47     FOREIGN KEY (id_usuario) REFERENCES usuario(id_usuario),  
48     FOREIGN KEY (id_funcionario) REFERENCES funcionario(id_funcionario)  
49 );  
50
```

Inserindo dados num banco de dados relacional

Exercício : Gere comandos para inserir dados nas tabelas do seu banco de dados.

Manipulando a tabela Editora:

```
INSERT INTO editora (id, nome, pais)
VALUES (1, 'Editora A', 'Brasil');

UPDATE editora
SET pais = 'Portugal'
WHERE id = 1;

DELETE FROM editora
WHERE id = 1;
```

Manipulando a tabela Autor:

```
INSERT INTO autor (id, nome, nacionalidade, id_editora)
VALUES (1, 'João Silva', 'Brasileiro', 1);

UPDATE autor
SET nacionalidade = 'Português'
WHERE id = 1;

DELETE FROM autor
WHERE id = 1;
```

Exercício - Gere pesquisas para suas tabelas do banco de dados:

```
-- - Seleciona o nome de todos autores brasileiros.
SELECT nome
FROM autor
WHERE nacionalidade = 'Brasil';

-- - Seleciona o nome e o país das editoras brasileiras.
SELECT nome, pais
FROM editora
WHERE pais = 'Brasil';

-- - Seleciona o nome e o cargo dos gerentes.
SELECT nome, cargo
FROM funcionario
WHERE cargo = 'Gerente';

-- - Seleciona o título e o ano de publicação dos livros publicados entre 2000 e 2010.
SELECT titulo, ano_pub
FROM livro
WHERE ano_pub BETWEEN '2000-01-01' AND '2010-12-31';

-- - Seleciona o nome, endereço e telefone dos usuários que moram em São Paulo.
SELECT nome, endereco, telefone
FROM usuario
WHERE endereco LIKE '%São Paulo%';

-- - Seleciona o título do livro e o nome de todos autores brasileiro.
SELECT titulo, nome
FROM livro
JOIN autor ON livro.id_autor = autor.id
WHERE autor.nacionalidade = 'Brasil';

-- - Seleciona o nome do usuário e o número de empréstimos que ele fez,
-- ordenados pelo número de empréstimos em ordem decrescente.
SELECT nome, COUNT() AS 'Número de empréstimos'
FROM usuario
JOIN emprestimo ON usuario.id = emprestimo.id_usuario
GROUP BY usuario.id
ORDER BY COUNT() DESC;

-- - Seleciona o nome do funcionário e o número de empréstimos que ele fez,
-- ordenados pelo número de empréstimos em ordem decrescente.
SELECT nome, COUNT() AS 'Número de empréstimos'
FROM funcionario
JOIN emprestimo ON funcionario.id = emprestimo.id_funcionario
GROUP BY funcionario.id
ORDER BY COUNT() DESC;

-- - Seleciona o nome do autor e o número de livros que ele escreveu,
-- ordenados pelo número de livros em ordem decrescente.
SELECT nome, COUNT() AS 'Número de livros'
FROM autor
JOIN livro ON autor.id = livro.id_autor
GROUP BY autor.id
ORDER BY COUNT() DESC;
```

Exercícios propostos

Nos exercícios propostos, com base nos exercícios resolvidos, você deverá modelar o sistema descrito, identificando os requisitos de dados, entidades e atributos, aplicando técnicas de normalização, definindo os relacionamentos e refinando o modelo conforme necessário. A seguir, gere um script para a criação do banco de dados, implementando as tabelas, chaves primárias, estrangeiras e outras estruturas. Em seguida, elabore comandos SQL para a inserção de dados nas tabelas e crie consultas para realizar pesquisas no banco de dados, garantindo que todas as etapas sejam executadas de forma eficiente e que o sistema funcione conforme esperado.

Sistema de Gerenciamento de Academia:

Este sistema facilita a gestão de uma academia de ginástica, onde clientes podem se inscrever para treinos. Cada cliente é atendido por um treinador, que pode ser responsável por vários horários de treino. Equipamentos são utilizados durante os treinos, e os clientes realizam pagamentos pelo serviço.

Sistema de Gerenciamento de Restaurante:

Neste sistema, clientes fazem pedidos de produtos em um restaurante. Cada pedido pode conter vários itens do cardápio. Funcionários atendem aos pedidos dos clientes, que são associados a uma mesa específica. Os produtos são fornecidos por fornecedores externos.

Sistema de Gerenciamento de Hotel:

Este sistema administra as reservas em um hotel, onde hóspedes podem reservar quartos para estadias. Cada quarto pode ser reservado várias vezes ao longo do tempo. Funcionários lidam com as reservas e prestam serviços aos hóspedes durante sua estadia, os quais realizam pagamentos pelo serviço.

Sistema de Gerenciamento de Escola:

Neste sistema, alunos estão matriculados em disciplinas, onde cada disciplina é ministrada por um professor. As disciplinas são oferecidas em turmas específicas, e os alunos recebem notas em cada disciplina ao longo do período letivo. O horário indica quando as aulas ocorrem.