



# Atividade Revisão Teórica

**Observação:** As fontes foram deixadas também nas referências, final da página

## Parte 1

1. O que é uma entidade em um banco de dados e qual é o seu propósito principal?

Basicamente, podemos considerar que é aquele com que capturamos os dados a partir de uma... entidade! Um objeto pode ser gerado a partir desta tabela para se gerar uma pessoa com informações completas. Dentro do banco de dados, isso pode ser representado por uma tabela.

**Fonte:**

<https://www.bing.com/ck/a?!&&p=72046e9c468510a2JmltdHM9MTY4NTY2NDAwMCZpZ3VpZD0yNWNhMGYzNC02YjliLTZkMzktMWRmYi0wMDY4NmE0ZTZjYjAmaW5zaWQ9NTQzMg&ptn=3&hsh=3&fclid=25ca0f34-6b9b-6d39-1dfb-00686a4e6cb0&psq=O+que+é+uma+entidade+em+um+banco+de+dados+e+qual+é+o+seu+propósito+principal?&u=a1aHR0cHM6Ly90ZWNoZW50ZXluY29tLmJyL28tcXVILWUtby1tb2RlbG8tZW50aWRhZGUTcmVsYWNPb25hbWVudG8tbW9kZWxvLWVyLWludHJvZHVjYW8vIzp-OnRleHQ9RW50aWRhZGUIMjAIQzMIQTkIMjB1bWEIMjBjb2lzYSUyMG91JTIwb2JqZXRvJTIwbm8IMjBtdW5kbyxjb25zaWRlcmFkb3MIMjBlnRpZGFkZXMuJTIwRXNzZXMIMjBkYWRvcyUyMHMIQzMIQTnVJTIwY2hhbWFkb3MIMjBkZSUyMGF0cmliidXRvcy4&ntb=1>

1

2. Quais são as características que definem uma entidade em um modelo de banco de dados relacional?

Exemplo, em uma tabela “pessoa”, iremos capturar as informações de nome, cpf, idade, etc. Nesta tabela, pessoa é uma entidade e estas informações em si, são atributos da entidade. E assim essa tabela possui, chave primária, (aquela que é primordial e não pode faltar na identificação do objeto de cada entidade), relacionamentos com outras tabelas através de chaves estrangeiras, (informações que são exportadas e importadas de outras tabelas, que podem ser atributos de uma entidade), cardinalidade, (a quantidade de ocorrências que podem ocorrer em um relacionamento entre duas entidades.)

**Fonte:**

<https://www.bing.com/ck/a?!&&p=72046e9c468510a2JmldtHM9MTY4NTY2NDAwMCZpZ3VpZD0yNWNhMGYzNC02YjliLTZkMzktMWRmYi0wMDY4NmE0ZTZjYjAmaW5zaWQ9NTQzMg&pptn=3&hsh=3&fclid=25ca0f34-6b9b-6d39-1dfb-00686a4e6cb0&psq=O+que+é+uma+entidade+em+um+banco+de+dados+e+qual+é+o+seu+propósito+principal?&u=a1aHR0cHM6Ly90ZWNoZW50ZXluY29tLmJyL28tcXVILWUtby1tb2RlbG8tZW50aWRhZGUCmVsYWNpb25hbWVudG8tbW9kZWxvLWVvLWludHJvZHVjYW8vIzp-OnRleHQ9RW50aWRhZGUiMjAIQzMIQTklMjB1bWEIMjBjb2lzYSUyMG91JTIwb2JqZXRvJTIwbm8IMjBtdW5kbyxjb25zaWRlcmFkb3MIMjBibnRpZGFkZXMuJTIwRXNzZXMIMjBkYWRvcyUyMHMIQzMIQTNvJTIwY2hhbWFKb3MIMjBkZSUyMGF0cmliidXRvcy4&ntb=1>

3. Explique a diferença entre uma entidade forte e uma entidade fraca.

Uma entidade forte pode ser considerada aquela que pode coexistir independentemente, ou seja, sem relacionamentos dentro do banco de dados, não dependendo de nenhuma informação. A entidade fraca é aquela que depende.

**Fonte:**

<https://www.bing.com/ck/a?!&&p=72046e9c468510a2JmldtHM9MTY4NTY2NDAwMCZpZ3VpZD0yNWNhMGYzNC02YjliLTZkMzktMWRmYi0wMDY4NmE0ZTZjYjAmaW5zaWQ9NTQzMg&pptn=3&hsh=3&fclid=25ca0f34-6b9b-6d39-1dfb-00686a4e6cb0&psq=O+que+é+uma+entidade+em+um+banco+de+dados+e+qual+é+o+seu+propósito+principal?&u=a1aHR0cHM6Ly90ZWNoZW50ZXluY29tLmJyL28tcXVILWUtby1tb2RlbG8tZW50aWRhZGUCmVsYWNpb25hbWVudG8tbW9kZWxvLWVvLWludHJvZHVjYW8vIzp-OnRleHQ9RW50aWRhZGUiMjAIQzMIQTklMjB1bWEIMjBjb2lzYSUyMG91JTIwb2JqZXRvJTIwbm8IMjBtdW5kbyxjb25zaWRlcmFkb3MIMjBibnRpZGFkZXMuJTIwRXNzZXMIMjBkYWRvcyUyMHMIQzMIQTNvJTIwY2hhbWFKb3MIMjBkZSUyMGF0cmliidXRvcy4&ntb=1>

**Fonte:**

1

<https://www.bing.com/ck/a?!&&p=72046e9c468510a2JmldtHM9MTY4NTY2NDAwMCZpZ3VpZD0yNWNhMGYzNC02YjliLTZkMzktMWRmYi0wMDY4NmE0ZTJyYjAmaW5zaWQ9NTQzMg&ptn=3&hsh=3&fclid=25ca0f34-6b9b-6d39-1dfb-00686a4e6cb0&psq=O+que+é+uma+entidade+em+um+banco+de+dados+e+qual+é+o+seu+propó+sito+principal?&u=a1aHR0cHM6Ly90ZWNoZW50ZXluY29tLmJyL28tcXVILWUtby1tb2RibG8tZW50aWRhZGUtcmVsYWNpb25hbWVudG8tbW9kZWxvLWVvLWludHJvZHVjYW8vIzp-OnRleHQ9RW50aWRhZGUIMjAlQzMlQ-TkIMjB1bWEIMjBjb2lzYSUyMG91JTlwb2JqZXRvJTlwbm8IMjBtdW5kbyxjb25zaWRlcmFkb3MIMjBlbnRpZGFkZXMmJTlwbRXNzZXMMjBkYWRvcyUyMHMlQzMlQTNvJTlwy2hhbWFkb3MIMjBkZSUyMGF0cmliidXRvcy4&ntb=1>

#### 4. Como você identificaria e nomearia uma entidade em um projeto de banco de dados?

Definição clara do nome, ex: Tabela de cliente, deve possuir informações com chave primária, dependendo estrangeiras, atributos em si, sobre o cliente.

**Fonte:**

<https://www.bing.com/ck/a?!&&p=72046e9c468510a2JmldtHM9MTY4NTY2NDAwMCZpZ3VpZD0yNWNhMGYzNC02YjliLTZkMzktMWRmYi0wMDY4NmE0ZTJyYjAmaW5zaWQ9NTQzMg&ptn=3&hsh=3&fclid=25ca0f34-6b9b-6d39-1dfb-00686a4e6cb0&psq=O+que+é+uma+entidade+em+um+banco+de+dados+e+qual+é+o+seu+propó+sito+principal?&u=a1aHR0cHM6Ly90ZWNoZW50ZXluY29tLmJyL28tcXVILWUtby1tb2RibG8tZW50aWRhZGUtcmVsYWNpb25hbWVudG8tbW9kZWxvLWVvLWludHJvZHVjYW8vIzp-OnRleHQ9RW50aWRhZGUIMjAlQzMlQ-TkIMjB1bWEIMjBjb2lzYSUyMG91JTlwb2JqZXRvJTlwbm8IMjBtdW5kbyxjb25zaWRlcmFkb3MIMjBlbnRpZGFkZXMmJTlwbRXNzZXMMjBkYWRvcyUyMHMlQzMlQTNvJTlwy2hhbWFkb3MIMjBkZSUyMGF0cmliidXRvcy4&ntb=1>

#### 5. Quais são os tipos de relacionamentos que uma entidade pode ter com outras entidades em um modelo de banco de dados relacional?

Relacionamento Um-para-Um (1:1): Uma entidade está associada a no máximo uma instância de outra entidade, e vice-versa. Exemplo: "Funcionário" tem um único "Endereço".

Relacionamento Um-para-Muitos (1:N): Uma entidade está associada a várias instâncias de outra entidade, mas cada instância desta última está associada a no máximo uma instância da primeira entidade. Exemplo: "Turma" tem vários "Alunos", mas cada "Aluno" está associado a apenas uma "Turma".

Relacionamento Muitos-para-Muitos (N:N): Várias instâncias de uma entidade estão associadas a várias instâncias de outra entidade. Uma tabela intermediária é usada para mapear as associações entre as entidades. Exemplo: "Livro" pode ter vários "Autores" e um "Autor" pode ter escrito vários "Livros".

**Fonte:**

<https://www.bing.com/ck/a?!&&p=5e0d50ae0e0bd481JmltdHM9MTY4NTY2NDAwMCZpZ3VpZD0yNWNhMGYzNC02YjliLTZkMzktMWRmYi0wMDY4NmE0ZTZjYjAmaW5zaWQ9NTQyOA&p=3&hsh=3&fclid=25ca0f34-6b9b-6d39-1dfb-00686a4e6cb0&psq=Quais+são+os+tipos+de+relacionamentos+que+uma+entidade+pode+ter+com+outras+entidades+em+um+modelo+de+banco+de+dados+relacional?&u=a1aHR0cHM6Ly93d3cuZGV2bWVkaWEuY29tLmJyL21lci1lLWRlci1tb2RlbGFnZW0tZGUtYmFuY29zLWRILWRhZG9zLzE0MzMylzp-OnRleHQ9RGUIMjBhY29yZG8IMjBjb20IMjBhJTlwcXVhbnRpZGFkZSUyMGRlJTlwb2JqZXRvcyUyMGVudm9sdmlkb3MsdW1hJTlwdW5pZGFkZSUyMGRhJTlwb3V0cmElMjBlbnRpZGFkZS4IMjAuLi4IMjBNYWlzMjIwaXRlbnM&ntb=1>

## Parte 2

1. O que são atributos em um banco de dados e qual é a sua função?

São as informações, as colunas de cada tabela, e cada atributo é uma informação.

**Fonte:**

<https://www.bing.com/ck/a?!&&p=72046e9c468510a2JmltdHM9MTY4NTY2NDAwMCZpZ3VpZD0yNWNhMGYzNC02YjliLTZkMzktMWRmYi0wMDY4NmE0ZTZjYjAmaW5zaWQ9NTQzMg&p=3&hsh=3&fclid=25ca0f34-6b9b-6d39-1dfb-00686a4e6cb0&psq=O+que+é+uma+entidade+em+um+banco+de+dados+e+qual+é+o+seu+propósito+principal?&u=a1aHR0cHM6Ly90ZWNoZW50ZXluY29tLmJyL28tcXVILWUtby1tb2RlbG8tZW50aWRhZGUtcmVsYWNpb25hbWVudG8tbW9kZWxvLWVvLWludHJvZHVjYW8vIzp-OnRleHQ9RW50aWRhZGUIMjAlQzMIQTklMjB1bWEIMjBjb2l2YSUyMG91JTlwb2JqZXRvJTlwbm8IMjBtdW5kb3Vudm9sdmlkb3MIMjBlbnRpZGFkZSUyMGRlJTlwb2JqZXRvcyUyMGVudm9sdmlkb3MIMjBNYWlzMjIwaXRlbnM&ntb=1>

2. Explique a diferença entre um atributo simples e um atributo composto.

Um atributo simples é um valor atômico que não pode ser dividido em partes menores. Por exemplo, o número de telefone de um aluno é um valor atômico de 10 dígitos. Já um atributo composto é feito de mais de um atributo simples. O seu conteúdo é formado por vários itens menores. Por exemplo, o endereço pode ser dividido em vários outros atributos como rua, número, bairro e cidade. Dentro de uma aplicação, isso pode gerar um objeto composto usado pelo desenvolvedor.

**Fonte:**

<https://www.bing.com/ck/a?!&&p=e53f7a6c3f2ab85dJmltdHM9MTY4NTY2NDAwMCZpZ3VpZD0yNWNhMGYzNC02YjliLTZkMzktMWRmYi0wMDY4NmE0ZTZjYjAmaW5zaWQ9NTQyOQ&p=3&hsh=3&fclid=25ca0f34-6b9b-6d39-1dfb-00686a4e6cb0&psq=Explique+a+diferença+entre+um+atributo+simples+e+um+atributo+composto.&u=a1aHR0cHM6Ly93d3cuZGV2bWVkaWEuY29tLmJyL21lci1lLWRlci1tb2RlbGFnZW0tZGUtYmFuY29zLWRILWRhZG9zLzE0MzMylzp-OnRleHQ9RGUIMjBhY29yZG8IMjBjb20IMjBhJTlwcXVhbnRpZGFkZSUyMGRlJTlwb2JqZXRvcyUyMGVudm9sdmlkb3MsdW1hJTlwdW5pZGFkZSUyMGRhJTlwb3V0cmElMjBlbnRpZGFkZS4IMjAuLi4IMjBNYWlzMjIwaXRlbnM&ntb=1>

0EIMjBub21IJTJDJTIwc2V4byUyQyUyMGRhdGEIMjBkZSUyMG5hc2NpbWVudG8IMkMIMjBkZW50cmUIMjBvdXRyb3Mu&ntb=1

### 3. Quais são os tipos de atributos que podem existir em um modelo de banco de dados relacional?

Os tipos de atributos em um modelo de banco de dados relacional incluem: numéricos, de texto, de data e hora, booleanos, chave primária, chave estrangeira e calculados/derivados. Eles representam valores numéricos, textuais, datas, valores lógicos e chaves de identificação. Os atributos calculados são derivados de outros atributos existentes no banco de dados. A escolha adequada do tipo de atributo depende da natureza dos dados e das operações a serem realizadas, podendo gerar a partir de consultas ou subconsultas.

**Fonte:**

 <https://www.bing.com/ck/a?!&&p=e53f7a6c3f2ab85dJmltdHM9MTY4NTY2NDAwMCZpZ3VpZD0yNWNhMGYzNC02YjliLTZkMzktMWRmYi0wMDY4NmE0ZTZjYjAmaW5zaWQ9NTQyOQ&ptn=3&hsh=3&fclid=25ca0f34-6b9b-6d39-1dfb-00686a4e6cb0&psq=Explique+a+diferença+entre+um+atributo+simples+e+um+atributo+composto.&u=a1aHR0cHM6Ly93d3cubHVpcy5ibG9nLmJyL2FuYWxpc2UtZGUtZW50aWRhZG9tYXRYaWJ1dG9zLXNpbXBsZXMtY29tcG9zdG9zLW11bHRpdMFSb3JhZG9zLmh0bWwJOn46dGV4dD1RdWFuZG8IMjB1bSUyMGF0cmliidXRvJTlwbiVDMYyVBM28IMjAlQzMIQTklMjBjB21wb3N0byUyQyUyMHJlY2ViZSUyMHVtLHNpbXBsZXMIM0EIMjBub21IJTJDJTIwc2V4byUyQyUyMGRhdGEIMjBkZSUyMG5hc2NpbWVudG8IMkMIMjBkZW50cmUIMjBvdXRyb3Mu&ntb=1>

### 4. O que é uma chave primária e qual é o seu propósito em um banco de dados?

É aquela chave que digamos que define o atributo principal daquela tabela, ou que define-a, e assim ele não pode faltar na construção do objeto total com seus atributos, sendo o atributo principal. Sendo assim podemos definir que ainda, é a ponte de relacionamentos desta tabela criada, com outra tabela.

**Fonte:**

<https://www.bing.com/ck/a?!&&p=6519aaf676af461bJmltdHM9MTY4NTY2NDAwMCZpZ3VpZD0yNWNhMGYzNC02YjliLTZkMzktMWRmYi0wMDY4NmE0ZTZjYjAmaW5zaWQ9NTQyOQ&ptn=3&hsh=3&fclid=25ca0f34-6b9b-6d39-1dfb-00686a4e6cb0&psq=O+que+é+uma+chave+primária+e+qual+é+o+seu+propósito+em+um+banco+de+dados?&u=a1aHR0cHM6Ly9wdC53aWtpcGVkaWEub3JnL3dpa2kvQ2hhdmVfcHJpbSVDMYyVBMXJpYSM6fj0ZXh0PUNoYXZlcyUyMHByaW0lQzMIQTlFyaWFzJTlwJTl4ZW0IMjBpbmdsJUMzJUFBCyUyQyUyMFByaW1hcnklMjBkZSUyMGRhZG9zJTlwcmVsYWNpb25hbCUyOS4&ntb=1>

5. Descreva o conceito de chave estrangeira e como ela é usada para estabelecer relacionamentos entre tabelas.

A chave estrangeira é um atributo de uma tabela que é a chave primária de outra tabela, usada de fora, e por isso estrangeira.

**Fonte:**

<https://www.bing.com/ck/a?!&&p=6519aaf676af461bJmldtHM9MTY4NTY2NDAwMCZpZ3VpZD0yNWNhMGYzNC02YjliLTZkMzktMWRmYi0wMDY4NmE0ZTZjYjAmaW5zaWQ9NTQzMA&ptn=3&hsh=3&fclid=25ca0f34-6b9b-6d39-1dfb-00686a4e6cb0&psq=O+que+é+uma+chave+primária+e+qual+é+o+seu+propósito+em+um+banco+de+dados?&u=a1aHR0cHM6Ly9wdC53aWtpcGVkaWEub3JnL3dpa2kvQ2hhdmVfcHJpbSVDMYyVBMXJpYSM6fjp0ZXh0PUNoYXZlcyUyMHByaW0lQzMIQTfyaWFzJTlWJTl4ZW0lMjBpbmJzJUMzJUFBCyUyQyUyMFByaW1hcnklMjBrcXl3JTlwb3UIMjAlMjJQSyUyMiUyOSUyQywlMjhkYSVDMYyVBRCUyMHZlbSUyMG8lMjBub21lJTlWYmFuY28lMjBkZSUyMGRhZG9zJTlwcmlwYWNp25hbCUyOS4&ntb=1>

## Parte 3

1. O que é um relacionamento em um banco de dados e qual é o seu propósito?

O relacionamento é onde a tabela se relaciona com atributo de outra tabela, com chave estrangeira, puxando aquela informação, a completando ou completando outra tabela, ou ainda coexistindo juntos, sendo juntamente uma informação necessária as duas.

**Fonte:**

<https://www.bing.com/ck/a?!&&p=13cb561d6c09a7d8JmldtHM9MTY4NTY2NDAwMCZpZ3VpZD0yNWNhMGYzNC02YjliLTZkMzktMWRmYi0wMDY4NmE0ZTZjYjAmaW5zaWQ9NTQzMA&ptn=3&hsh=3&fclid=25ca0f34-6b9b-6d39-1dfb-00686a4e6cb0&psq=O+que+é+um+relacionamento+em+um+banco+de+dados+e+qual+é+o+seu+propósito?&u=a1aHR0cHM6Ly93d3cuaWJtLmNvbS9kb2NzL3B0LWJyL2NvbnRyb2wtZGVzay83LjYuMS54P3RvcGljPjBkZSUyMGJhbmNvJTlWZGUIMjBkYWRvcyUyMHMIQzMIQTNvJTlWYXNzb2NpYSVDMYyVBNyVDMYyVCNWVzLHNlZ3VpciUyMGRlc2NyZXZlJTlwb3MIMjBkYWRvcyUyMGJhbmNvJTlWZGUIMjBkYWRvcy4&ntb=1>

2. Explique o conceito de cardinalidade em um relacionamento e descreva os principais tipos de cardinalidade.

Cardinalidade é o tipo de relacionamento que se faz entre as tabelas a partir de sua dependência.

Relacionamento Um-para-Um (1:1): Uma entidade está associada a no máximo uma instância de outra entidade, e vice-versa. Exemplo: "Funcionário" tem um único "Endereço".

Relacionamento Um-para-Muitos (1:N): Uma entidade está associada a várias instâncias de outra entidade, mas cada instância desta última está associada a no máximo uma instância da primeira entidade. Exemplo: "Turma" tem vários "Alunos", mas cada "Aluno" está associado a apenas uma "Turma".

Relacionamento Muitos-para-Muitos (N:N): Várias instâncias de uma entidade estão associadas a várias instâncias de outra entidade. Uma tabela intermediária é usada para mapear as associações entre as entidades. Exemplo: "Livro" pode ter vários "Autores" e um "Autor" pode ter escrito vários "Livros".

**Fonte:**

<https://www.bing.com/ck/a?!&&p=5e0d50ae0e0bd481JmldtHM9MTY4NTY2NDAwMCZpZ3VpZD0yNWNhMGYzNC02YjliLTZkMzktMWRmYi0wMDY4NmE0ZTZjYjAmaW5zaWQ9NTQyOA&ptn=3&hsh=3&fclid=25ca0f34-6b9b-6d39-1dfb-00686a4e6cb0&psq=Quais+são+os+tipos+de+relacionamentos+que+uma+entidade+pode+ter+com+outras+entidades+em+um+modelo+de+banco+de+dados+relacional?&u=a1aHR0cHM6Ly93d3cuZGV2bWVkaWEuY29tLmJyL21lci1lLWRlci1tb2RlbgFnZW0tZGUtYmFuY29zLWRILWRhZG9zLzE0MzMylzp-OnRleHQ9RGUIMjBhY29yZG8IMjBjb20IMjBhJTlwcXVhbnRpZGFkZSUyMGRlJTlwb2JqZXRvcyUyMGVudm9sdmlkb3MsdW1hJTlwdW5pZGFkZSUyMGRhJTlwb3V0cmElMjBlbnRpZGFkZS4IMjAuLi4lMjBNYWl3JTlwaXRlbnM&ntb=1>

3. Qual é a diferença entre cardinalidade mínima e cardinalidade máxima em um relacionamento?

A Cardinalidade indica quantas ocorrências de uma Entidade participam no mínimo e no máxima do relacionamento. Cardinalidade Mínima - define se o relacionamento entre duas entidades é obrigatório ou não. ... Cardinalidade Máxima - define a quantidade máxima de ocorrências da Entidade que pode participar do Relacionamento.

**Fonte:**

<https://euquerosabertudo.com/social/o-que-cardinalidade-mnima-e-mxima.html#:~:text=A Cardinalidade indica quantas ocorrências de uma Entidade,ocorrências da Entidade que pode participar do Relacionamento.>





<https://www.bing.com/ck/a?!&&p=5e0d50ae0e0bd481JmltdHM9MTY4NTY2NDAwMCZpZ3VpZD0yNWNhMGYzNC02YjliLTZkMzktMWRmYi0wMDY4NmE0ZTZjYjAmaW5zaWQ9NTQyOA&ptn=3&hsh=3&fclid=25ca0f34-6b9b-6d39-1dfb-00686a4e6cb0&psq=Quais+são+os+tipos+de+relacionamentos+que+uma+entidade+pode+ter+com+outras+entidades+em+um+modelo+de+banco+de+dados+relacional?&u=a1aHR0cHM6Ly93d3cuZGV2bWVkaWEuY29tLmJyL21lci1lLWRlci1tb2RlbGFnZW0tZGUtYmFuY29zLWRILWRhZG9zLzE0MzMylzp-OnRleHQ9RGUIMjBhY29yZG8IMjBjb20IMjBhJTlwcXVhbnRpZGFkZSUyMGRlJTlwb2JqZXRvcyUyMGVudm9sdmlkb3MsdW1hJTlwdW5pZGFkZSUyMGRhJTlwb3V0cmElMjBlbnRpZGFkZS4IMjAuLi4IMjBNYWlzMjIwXRIbnM&ntb=1>

## 7. Explique como implementar um relacionamento um-para-muitos em um banco de dados.

Para implementar um relacionamento um-para-muitos em um banco de dados, você precisa seguir alguns passos:

1. Identificação das entidades: Identifique as duas entidades envolvidas no relacionamento um-para-muitos. Por exemplo, considere as entidades "Turma" e "Aluno", onde uma turma pode ter vários alunos.
2. Chave primária e chave estrangeira: Certifique-se de que a entidade "Turma" tenha uma chave primária única para identificar cada turma. Essa chave primária será referenciada na entidade "Aluno" como uma chave estrangeira. Por exemplo, a chave primária da entidade "Turma" pode ser o atributo "id\_turma", e esse atributo será uma chave estrangeira na entidade "Aluno" chamada "id\_turma".
3. Criação de tabelas: Crie uma tabela para cada entidade, onde cada tabela terá os atributos relevantes para essa entidade. A tabela da entidade "Turma" terá a chave primária "id\_turma" e outros atributos relacionados à turma, como "nome\_turma" e "horario\_turma". A tabela da entidade "Aluno" terá atributos como "id\_aluno", "nome\_aluno", "idade\_aluno" e a chave estrangeira "id\_turma".
4. Relacionamento no esquema: No esquema do banco de dados, defina o relacionamento um-para-muitos entre as tabelas "Turma" e "Aluno". Isso é feito estabelecendo a chave estrangeira "id\_turma" na tabela "Aluno" que referencia a chave primária "id\_turma" na tabela "Turma".
5. Restrições de integridade referencial: Defina as restrições de integridade referencial para garantir a consistência dos dados. Isso envolve configurar a restrição de chave estrangeira na coluna "id\_turma" da tabela "Aluno", referenciando a chave primária "id\_turma" na tabela "Turma". Isso garantirá que

apenas valores existentes na tabela "Turma" possam ser referenciados pela tabela "Aluno".

8. O que é um relacionamento muitos-para-muitos e como ele é implementado em um banco de dados relacional?

1. Identificação das entidades: Identifique as duas entidades envolvidas no relacionamento muitos-para-muitos. Por exemplo, considere as entidades "Livro" e "Autor", onde um livro pode ter vários autores e um autor pode ter escrito vários livros.

1. Criação de tabelas: Crie uma tabela para cada entidade, onde cada tabela terá os atributos relevantes para essa entidade. A tabela "Livro" terá a chave primária "id\_livro" e outros atributos relacionados ao livro, como "titulo\_livro" e "ano\_publicacao". A tabela "Autor" terá a chave primária "id\_autor" e atributos como "nome\_autor" e "pais\_autor".

2. Criação da tabela associativa: Crie uma terceira tabela, também conhecida como tabela associativa ou tabela de junção, para representar o relacionamento muitos-para-muitos entre as entidades "Livro" e "Autor". Essa tabela terá colunas que correspondem às chaves primárias das entidades relacionadas. Por exemplo, crie uma tabela "LivroAutor" com as colunas "id\_livro" e "id\_autor".

3. Estabelecimento das chaves estrangeiras: Nas tabelas "LivroAutor", "Livro" e "Autor", estabeleça as chaves estrangeiras que fazem referência às chaves primárias correspondentes. Na tabela "LivroAutor", as colunas "id\_livro" e "id\_autor" serão chaves estrangeiras que referenciam as chaves primárias "id\_livro" na tabela "Livro" e "id\_autor" na tabela "Autor", respectivamente.

4. Relacionamento no esquema: No esquema do banco de dados, defina o relacionamento muitos-para-muitos entre as tabelas "Livro" e "Autor" por meio da tabela associativa "LivroAutor". Isso é feito estabelecendo as chaves estrangeiras nas tabelas relevantes e conectando-as corretamente.

9. Quais são as chaves estrangeiras e como elas são usadas para estabelecer relacionamentos entre tabelas?



## Parte 4

1. O que é álgebra relacional e qual é o seu propósito em um banco de dados?

A Álgebra Relacional é uma **linguagem de consulta formal**, porém procedimental, ou seja, o usuário dá as instruções ao sistema para que o mesmo realize uma sequência de operações na base de dados para calcular o resultado desejado.

**Fonte:**

```
https://www.bing.com/ck/a?!&&p=efadc5191a446b07JmltdHM9MTY4NTY2NDAwMCZpZ3VpZD0yNWNhMGYzNC02YjliLTZkMzktMWRmYi0wMDY4NmE0ZTZjYjAmaW5zaWQ9NTQzMg&pntn=3&hsh=3&fclid=25ca0f34-6b9b-6d39-1dfb-00686a4e6cb0&psq=O+que+é+álgebra+relacional+e+qual+é+o+seu+propósito+em+um+banco+de+dados?&u=a1aHR0cHM6Ly93d3cuZGV2bWVkaWEuY29tLmJyL2FsZ2VicmEtcuVsYWNpb25hbC10dXRvcmlhbnC8yNjYzIzp-OnRleHQ9QSUyMCVDMYU4MWxnZWJyYyYySUyMFJlbGFjaW9uYWwIMjAlQzMIQztkIMjB1bWEIMjBsaW5ndWFnZW0IMjBkZSUyMGNvbN1bHRhLGJhc2UIMjBkZSUyMGRhZG9zJTlwcGFyYyYyMGNhbGN1bGFyJTlwbYUyMHJlc3VsdGFkbyUyMGRlc2VqYWRvLg&ntb=1
```

2. Descreva a operação de seleção (ou restrição) na álgebra relacional e explique como ela é usada.

A operação de seleção (ou restrição) na álgebra relacional é usada para reduzir o número de tuplas em uma relação, selecionando apenas as tuplas que satisfazem uma determinada condição. Essa condição é especificada como uma expressão booleana que pode incluir operadores lógicos, operadores de comparação e valores constantes. A seleção é representada pelo símbolo sigma ( $\sigma$ ) e é usada da seguinte maneira:

$\sigma_{\text{<condição>}}(R)$

Onde R é a relação de entrada e  $\text{<condição>}$  é a expressão booleana usada para selecionar as tuplas. Por exemplo, a expressão abaixo seleciona todas as tuplas da relação "Clientes" onde o saldo é maior que 1000:

$\sigma_{\text{saldo} > 1000}(\text{Clientes})$

A operação de seleção é uma das operações fundamentais na álgebra relacional e é usada em conjunto com outras operações para realizar consultas complexas em bancos de dados relacionais.

3. O que é a operação de projeção na álgebra relacional e quais são seus efeitos?

A operação de projeção na álgebra relacional é usada para selecionar um subconjunto dos atributos de uma relação. Essa operação é representada pelo símbolo pi ( $\pi$ ) e é usada da seguinte maneira:

$\pi \langle \text{atributos} \rangle (R)$

Onde R é a relação de entrada e  $\langle \text{atributos} \rangle$  é uma lista de atributos separados por vírgulas que especifica quais atributos devem ser incluídos na relação resultante. Por exemplo, a expressão abaixo seleciona apenas os atributos "nome" e "telefone" da relação "Clientes":

$\pi \text{ nome, telefone (Clientes)}$

Os efeitos da operação de projeção são a redução do número de atributos na relação resultante e a remoção de tuplas duplicadas com base nos atributos selecionados. A operação de projeção é frequentemente usada em conjunto com outras operações para realizar consultas complexas em bancos de dados relacionais.

4. Explique a operação de união na álgebra relacional e como ela combina os resultados de duas relações.

A operação de união na álgebra relacional é usada para combinar as tuplas de duas relações que tenham o mesmo esquema, gerando uma nova relação que contém todas as tuplas das duas relações originais, sem duplicatas. A operação de união é representada pelo símbolo "U" e é usada da seguinte forma:

$R \cup S$

Onde R e S são as relações de entrada. A relação resultante conterá todas as tuplas de R e todas as tuplas de S que não estão em R. Para que a operação de união funcione corretamente, as duas relações devem ter o mesmo esquema (ou seja, o mesmo conjunto de atributos) e o mesmo domínio de valores para cada atributo. A operação de união é uma das operações fundamentais na álgebra relacional, e é comumente usada em consultas SQL para combinar resultados de duas ou mais consultas em uma única tabela.

5. O que é a operação de produto cartesiano na álgebra relacional e em que situações ela é utilizada?

A operação de produto cartesiano na álgebra relacional é usada para combinar todas as tuplas de duas relações, independentemente de quaisquer condições de junção. Essa operação é representada pelo símbolo " $\times$ " e é usada da seguinte forma:

$R \times S$

Onde R e S são as relações de entrada. A relação resultante conterá todas as combinações possíveis de tuplas de R e S. O produto cartesiano é uma operação fundamental na álgebra relacional, mas é usada com menos frequência do que outras operações, como a união e a junção. É mais comumente usado em situações em que é necessário comparar todas as tuplas de duas relações, como em consultas que envolvem todas as combinações de produtos e clientes em um banco de dados de comércio eletrônico.

#### 6. Explique a operação de junção natural na álgebra relacional e como ela funciona.

A operação de junção natural na álgebra relacional é usada para combinar duas relações em uma relação resultante que contém apenas as tuplas que têm valores correspondentes em ambas as relações para todos os atributos com o mesmo nome. Essa operação é representada pelo símbolo " $\bowtie$ " e é usada da seguinte forma:

$R \bowtie S$

Onde R e S são as relações de entrada. A relação resultante conterá todas as tuplas que têm valores correspondentes em ambas as relações para todos os atributos com o mesmo nome. Isso significa que apenas os valores que aparecem em ambas as relações são combinados e os valores exclusivos em cada relação são eliminados. A operação de junção natural é usada com frequência em consultas SQL para combinar resultados de duas ou mais tabelas que compartilham um atributo em comum.

## Parte 5

### 1. O que é SQL e qual é o seu propósito em bancos de dados?

SQL é uma linguagem de programação usada para gerenciar bancos de dados relacionais. O objetivo principal do SQL é permitir que os usuários consultem, atualizem e gerenciem bancos de dados de forma eficiente e segura.

2. Descreva a operação SELECT em SQL e explique como ela é usada para recuperar dados de uma tabela.

A operação SELECT em SQL é usada para recuperar dados de uma tabela. É a operação mais comum em SQL e é usada para consultar dados de uma ou mais tabelas, especificando as colunas desejadas, as tabelas a serem consultadas e as condições que os dados recuperados devem satisfazer. A sintaxe básica da operação SELECT é a seguinte:

```
SELECT <colunas> FROM <tabela> WHERE <condições>
```

Onde `<colunas>` é uma lista de colunas separadas por vírgulas que serão recuperadas, `<tabela>` é a tabela ou tabelas a serem consultadas e `<condições>` é uma expressão booleana que especifica as condições que os dados recuperados devem satisfazer. Por exemplo, para recuperar todos os dados da tabela "Clientes", a consulta SQL seria a seguinte:

```
SELECT * FROM Clientes;
```

Isso recuperaria todas as colunas da tabela "Clientes". Para recuperar colunas específicas, basta listar as colunas desejadas em vez de usar o asterisco (\*). Por exemplo, para recuperar apenas o nome e o endereço dos clientes, a consulta SQL seria a seguinte:

3. Qual é o papel da cláusula WHERE em uma instrução SQL e como ela afeta os resultados da consulta?

```
SELECT * FROM Clientes WHERE saldo > 1000;
```

A cláusula WHERE permite que os usuários especifiquem critérios de seleção mais complexos do que apenas recuperar todas as linhas de uma tabela. Isso ajuda a limitar o número de linhas retornadas e a tornar as consultas SQL mais eficientes.

4. Explique a operação INSERT em SQL e como ela é usada para inserir novos dados em uma tabela.

A operação INSERT em SQL é usada para inserir novos dados em uma tabela. É usada para adicionar uma nova linha a uma tabela existente ou para criar uma nova tabela com uma nova linha.

A sintaxe básica da operação INSERT é a seguinte:

```
INSERT INTO <tabela> (<coluna1>, <coluna2>, ...) VALUES (<valor1>, <valor2>, ...);
```

Onde `<tabela>` é o nome da tabela em que você deseja inserir os dados, `<coluna1>`, `<coluna2>`, ... são os nomes das colunas nas quais você está inserindo dados e `<valor1>`, `<valor2>`, ... são os valores que você está inserindo nessas colunas. Você pode inserir valores para todas ou algumas das colunas da tabela.

5. O que é a operação UPDATE em SQL e como ela é aplicada para modificar os dados existentes em uma tabela?

A operação UPDATE em SQL é usada para modificar os dados existentes em uma tabela. Ela é aplicada da seguinte maneira:

```
UPDATE <tabela> SET <coluna1> = <valor1>, <coluna2> = <valor2>, ... WHERE <condição>;
```

A operação UPDATE é usada para alterar os valores existentes em uma ou mais colunas de uma tabela. Ela pode ser usada em conjunto com outras operações SQL para modificar dados existentes em uma tabela e é uma das operações mais comuns em bancos de dados relacionais.

6. Descreva a operação DELETE em SQL e como ela é usada para excluir registros de uma tabela.

A operação DELETE em SQL é usada para excluir registros de uma tabela. Ela é aplicada da seguinte maneira:

```
DELETE FROM <tabela> WHERE <condição>;
```



A operação DELETE é usada para excluir registros de uma tabela com base em uma ou mais condições especificadas na cláusula WHERE. A cláusula WHERE é usada para especificar as condições que devem ser atendidas para que os registros sejam excluídos. Se a cláusula WHERE não for especificada, todos os registros da tabela serão excluídos.

7. Qual é o propósito da cláusula ORDER BY em SQL e como ela é utilizada para classificar os resultados de uma consulta?

A cláusula ORDER BY em SQL é usada para classificar os resultados de uma consulta em uma ordem específica. Ela é usada para classificar as linhas de uma tabela em uma ordem ascendente ou descendente com base em uma ou mais colunas. A cláusula ORDER BY é usada da seguinte maneira:

```
SELECT <colunas> FROM <tabela> WHERE <condições> ORDER BY <coluna1> [ASC|DESC], <coluna2> [ASC|DESC], ...;
```

Onde `<coluna1>`, `<coluna2>`, ... são as colunas pelas quais você deseja classificar os resultados. Você pode especificar várias colunas para classificação, separando-as por vírgulas. O `[ASC|DESC]` é opcional e especifica se você deseja classificar em ordem ascendente ou descendente. Se você não especificar nada, a ordem padrão é ascendente.

8. O que é a cláusula GROUP BY em SQL e qual é o seu papel na agregação de dados?

A cláusula GROUP BY em SQL é usada para agrupar linhas em uma determinada coluna ou conjunto de colunas. Ela é usada em conjunto com funções de agregação, como SUM, COUNT, AVG, MAX e MIN, para calcular estatísticas em grupos de linhas.

A sintaxe básica da cláusula GROUP BY é a seguinte:

```
SELECT <coluna1>, <coluna2>, ..., <função de agregação>(<coluna>) FROM <tabela> WHERE <condições> GROUP BY <coluna1>, <coluna2>, ...;
```

Onde `<coluna1>`, `<coluna2>`, ... são as colunas pelas quais você deseja agrupar as linhas. A `<função de agregação>` é a função que você deseja aplicar às colunas agrupadas, como SUM, COUNT, AVG, MAX ou MIN.

9. Descreva a operação de subconsulta (subquery) em SQL e como ela é aplicada para realizar consultas aninhadas.

A operação de subconsulta (subquery) em SQL é usada para realizar consultas aninhadas, ou seja, consultas dentro de outras consultas. Uma subconsulta é uma consulta que é incorporada em outra consulta e é usada para recuperar dados que são usados na consulta maior. A subconsulta é executada primeiro e seu resultado é usado na consulta maior como uma condição de seleção ou como uma fonte de dados.

A sintaxe básica de uma subconsulta é a seguinte:

```
SELECT <colunas> FROM <tabela> WHERE <coluna> <operador> (SELECT <coluna> FROM <tabela> WHERE <condição>);
```

Onde `<colunas>` é uma lista de colunas separadas por vírgulas que serão recuperadas, `<tabela>` é a tabela a ser consultada, `<coluna>` é a coluna usada como condição de seleção e `<operador>` é o operador usado na condição de seleção. A subconsulta é colocada entre parênteses e especificada como uma consulta SELECT separada.

---

## Referências:

- **"Database System Concepts"** de Abraham Silberschatz, Henry F. Korth e S. Sudarshan.
  - **"Fundamentals of Database Systems"** de Ramez Elmasri e Shamkant B. Navathe.
  - **"Database Management Systems"** de Raghu Ramakrishnan e Johannes Gehrke.
-

