

## CARDINALIDADE EM UM SGBD

A **cardinalidade**, em um Sistema de Gerenciamento de Banco de Dados (SGBD), refere-se à relação entre as entidades em um modelo de dados. Ela indica quantas ocorrências de uma entidade podem estar associadas a um certo número de ocorrências de outra entidade por meio de relacionamentos. A cardinalidade é uma parte essencial do projeto de banco de dados e ajuda a definir como as tabelas se relacionam entre si.

Existem dois tipos principais de cardinalidade: **cardinalidade mínima** e **cardinalidade máxima**.

### Cardinalidade Mínima:

A cardinalidade mínima indica o número mínimo de ocorrências que uma entidade pode ter em relação a outra entidade em um relacionamento. Ela é representada por símbolos ou palavras-chave, geralmente:

0: Indica que uma entidade pode não estar associada a nenhuma ocorrência da outra entidade.

1: Indica que uma entidade deve estar associada a pelo menos uma ocorrência da outra entidade.

### Cardinalidade Máxima:

A cardinalidade máxima indica o número máximo de ocorrências que uma entidade pode ter em relação a outra entidade em um relacionamento. Novamente, é representada por símbolos ou palavras-chave:

0 ou 1: Indica que uma entidade pode estar associada a no máximo uma ocorrência da outra entidade.

1: Indica que uma entidade está associada exatamente a uma ocorrência da outra entidade.

0..n: Indica que uma entidade pode estar associada a várias ocorrências da outra entidade.

1..n: Indica que uma entidade está associada a pelo menos uma ocorrência da outra entidade, mas pode estar associada a várias.

Um exemplo de como isso pode ser aplicado é considerando uma relação entre duas entidades: "Cliente" e "Pedido". A cardinalidade pode ser definida da seguinte forma:

Um cliente pode fazer zero ou mais pedidos: Cliente (0..n) - Pedido (1..n).

Um pedido deve estar associado a exatamente um cliente: Pedido (1) - Cliente (1).

A definição correta da cardinalidade ajuda a garantir a integridade dos dados e a modelar as relações entre as entidades de forma precisa dentro de um banco de dados.

No PostgreSQL, a cardinalidade é geralmente definida ao criar ou modificar tabelas por meio da linguagem SQL. Aqui está como você pode definir a cardinalidade em um relacionamento usando o PostgreSQL:

**Suponha que você tenha duas tabelas: "Cliente" e "Pedido", e deseja definir a cardinalidade entre elas.**

-- Criando a tabela Cliente

```
CREATE TABLE Cliente (  
    id_cliente SERIAL PRIMARY KEY,  
    nome VARCHAR(100)  
);
```

-- Criando a tabela Pedido

```
CREATE TABLE Pedido (  
    id_pedido SERIAL PRIMARY KEY,  
    id_cliente INT REFERENCES Cliente(id_cliente),  
    descricao TEXT  
);
```

Neste exemplo, o relacionamento entre "Cliente" e "Pedido" é definido através da coluna "id\_cliente" na tabela "Pedido" que faz referência à coluna "id\_cliente" na tabela "Cliente". Isso estabelece um relacionamento entre as tabelas.

Agora, para definir a cardinalidade mínima e máxima, você precisa entender o contexto das suas necessidades específicas. No exemplo acima, a cardinalidade é a seguinte:

**Um cliente pode fazer zero ou mais pedidos: Cliente (0..n) - Pedido (1..n).**

**Um pedido deve estar associado a exatamente um cliente: Pedido (1) - Cliente (1).**

Observe que a cardinalidade mínima é expressa pela coluna que é uma chave estrangeira (**REFERENCES Cliente(id\_cliente)**), que define a relação mínima de um pedido com um cliente. A cardinalidade máxima é determinada pelo uso da chave estrangeira e a maneira como as associações são estabelecidas.

Lembre-se de que a definição da cardinalidade pode variar com base nas regras de negócios específicas do seu aplicativo e nos requisitos do banco de dados.

A **cardinalidade máxima na chave estrangeira** é determinada pela forma como você configura as restrições de integridade referencial na tabela. No PostgreSQL, você pode usar as **palavras-chave ON DELETE e ON UPDATE** junto com a definição da chave estrangeira para controlar a cardinalidade máxima e o que acontece quando registros relacionados são excluídos ou atualizados.

Vou apresentar alguns exemplos para esclarecer como definir a cardinalidade máxima na chave estrangeira:

Um cliente pode fazer zero ou mais pedidos (Cliente 0..n - Pedido 1..n):

```
CREATE TABLE Cliente (  
    id_cliente SERIAL PRIMARY KEY,  
    nome VARCHAR(100)  
);
```

```
CREATE TABLE Pedido (  
    id_pedido SERIAL PRIMARY KEY,  
    id_cliente INT REFERENCES Cliente(id_cliente) ON DELETE CASCADE, -- DELETE  
    descricao TEXT  
);
```

CASCADE significa que se um cliente for excluído, seus pedidos também serão excluídos.

Nesse caso, estamos **usando ON DELETE CASCADE** para indicar que, se um cliente for excluído, todos os seus pedidos associados também serão excluídos. Isso está alinhado com a cardinalidade máxima de 1..n entre Cliente e Pedido.

Um pedido deve estar associado a exatamente um cliente (Pedido 1 - Cliente 1):

```
CREATE TABLE Cliente (  
    id_cliente SERIAL PRIMARY KEY,  
    nome VARCHAR(100)  
);
```

```
CREATE TABLE Pedido (  
    id_pedido SERIAL PRIMARY KEY,  
    id_cliente INT REFERENCES Cliente(id_cliente) ON DELETE RESTRICT, -- RESTRICT  
    significa que a exclusão do cliente é impedida se houver pedidos associados.  
    descricao TEXT  
);
```

Nesse exemplo, usamos **ON DELETE RESTRICT** para indicar que a exclusão de um cliente é restrita se houver pedidos associados a ele. Isso está alinhado com a cardinalidade máxima de 1 entre Pedido e Cliente.

Lembre-se de que as opções **ON DELETE** e **ON UPDATE** permitem controlar o que acontece quando há modificações nas chaves estrangeiras (quando um registro referenciado é excluído ou atualizado). As opções disponíveis incluem **CASCADE**, **RESTRICT**, **SET NULL** e **SET DEFAULT**, dependendo do comportamento desejado.