

## 5.2 - Entidades e Relacionamentos

00:00:00:00 - 00:00:00:44

Olá, pessoal. Bom, na última aula nós vimos o que são bancos de dados e a diferença de bancos relacionais e bancos não relacionais. Nessa aula nós vamos entender, vamos explorar mais essa questão de relacionamento das tabelas e na parte prática nós vamos ver como fica uma tabela em um gerenciador de banco de dados. Bora lá. Bom, voltamos pra nossa loja de música. Vimos que fica mais organizado assim, né? Porém, como é que cada tabela se relaciona mesmo? Uma forma de entender melhor esse relacionamento é a partir de um diagrama de entidade e relacionamento, um DER, ou diagrama ER, que é uma ferramenta de modelagem de dados que descreve as entidades, objetos ou conceitos em um sistema e os relacionamentos entre essas entidades.

00:00:00:44 - 00:00:01:29

No diagrama, as entidades, tabelas, são representadas por retângulos. E os relacionamentos são representados por linhas que conectam as entidades envolvidas. Aqui, por exemplo, a gente tem os retângulos e as linhas que conectam. O objetivo do diagrama é fornecer uma visão clara e concisa da estrutura do banco de dados, facilitando o entendimento das relações entre as entidades, as tabelas. Para nós que consultamos os dados, é muito importante ter documentações desse tipo para que a gente entenda os dados que estamos lidando e evite pegar informações erradas. Por isso, é importante entender como ler esse tipo de diagrama. Então, vamos entender primeiro os tipos de relacionamento que podem ter entre tabelas, ou seja, as entidades de bancos de dados relacionais.

00:00:01:29 - 00:00:02:15

Temos o relacionamento de tipo 1 para 1, onde cada registro de uma tabela está relacionado a exatamente um registro em outra tabela. Esses números, né, que costumam aparecer assim, ó, 1 para 1, representam a cardinalidade do relacionamento. Por exemplo, a gente tem uma tabela chamada pessoa e uma outra tabela chamada carteira, carteira de identidade. Cada pessoa tem uma única carteira de identidade, e cada carteira de identidade pertence a uma única pessoa, por isso, de um para um. Normalmente, nos diagramas, esse relacionamento também vai ser indicado por um símbolo, como, por exemplo, uma linha, ou uma linha com um traço, ou com dois traços. No caso do relacionamento um para um, ele é representado por uma linha contínua, assim.

00:00:02:15 - 00:00:03:12

Então vai ter uma tabela, uma linha e outra tabela. e nas extremidades vai estar os números que vão representar essa cardinalidade, ou seja, de 1 pra 1. Temos também o relacionamento 1 pra muitos, que é representado 1 pra N, né? 1 tracinho N. Aqui um registro em uma tabela pode ser relacionado a muitos registros em outra tabela, mas cada registro na segunda tabela está relacionado a exatamente um registro na primeira tabela. Por exemplo, a gente tem aqui uma tabela chamada álbum e uma tabela chamada música. Cada álbum pode ter várias músicas, mas cada música pertence a apenas um álbum. Então, 1 para N. E aí, se a gente for olhar, 1 para N é 1 para muitos, o símbolozinho é assim, 1 para muitos, ok? E a gente pode ter o contrário também, né, o relacionamento N para 1. N para 1.

00:00:03:12 - 00:00:04:13

Onde vários registros de uma tabela são relacionados a exatamente um registro em outra tabela. Nesse mesmo exemplo aí do álbum e da música, várias músicas podem pertencer a um único álbum. Então eu tenho várias músicas, múltiplas músicas pertencem a um álbum, ok? E a gente também tem o relacionamento muitos para muitos, N para N, onde vários registros em uma tabela podem estar relacionados a vários registros em outra tabela e vice-versa. Por exemplo, uma tabela clientes e uma tabela álbuns. Cada cliente pode comprar vários álbuns e cada álbum pode ser comprado por vários clientes. Então, N para N. E aí eu coloquei aqui do lado também para vocês já associarem esses símbolos que podem estar realmente ligando as tabelas. Uma linha reta de 1 para 1, 1 para N, N, 1 para N também, 1 para muitos, 1 para 1. Esse obrigatório significa que obrigatoriamente vai ter que ter um representante nessa tabela. Então, não pode ser 1 para nada, porque pode ter esse relacionamento também.

00:00:04:13 - 00:00:05:27

Tipo, a tabela está associando a outra, mas não tem realmente o seu associado. Então, quando a gente tem esse 1 e não está associado na outra tabela, a gente tem o 0 para 1, que está aqui, 0 para 1, ou 0 para muitos. Vamos ver como isso fica em um banco de dados real para a gente analisar. Existem muitos programas gerenciadores de bancos de dados, como MySQL, Workbench, o PgAdmin, o SQL Server, entre outros. Nem todos eles geram diagrama ER, mas aqui nesse curso nós vamos utilizar a plataforma DBeaver, que fornece uma interface amigável e permite escrever consultas em SQL, visualizar dados e gerenciar bancos de dados de forma eficiente, e que gera o diagrama para a gente. Outra coisa legal do DBeaver é que podemos conectar diferentes tipos de bancos de dados, como o MySQL, PostgreSQL, o SQL Server e etc. Bom, nos materiais complementares dessa aula vai ter um vídeo explicando como fazer para baixar e instalar o DBeaver. Então, eu aconselho que vocês pausem essa aula, vão lá nesse material, instalem o programa e depois voltem aqui para a gente continuar, beleza? Fiquem tranquilos que eu vou estar aqui aguardando vocês.

00:00:05:27 - 00:00:06:46

Vamos abrir o software. Aqui, já aberto. Assim que abrimos o DBeaver, temos essa página bem bonita. Nessa primeira barra, nós temos aqui informações básicas, como arquivo, editar, navegar, etc. Nessa barra abaixo, temos algumas coisas que usamos com mais recorrência, por isso é importante a gente dar uma maior atenção aí. Esse símbolo de tomada é para criar uma nova conexão, ou seja, criar um novo banco de dados. Se clicarmos nele, vai aparecer diferentes opções de sistemas de gerenciamento de banco de dados. Temos aqui do lado alguns símbolos de tomadas que estão em cinza, mas que representam conexão, reconexão e desconexão com o banco de dados. Ali em SQL, aqui, temos opções relacionadas a script de SQL, ou seja, a escrevermos aí um código em SQL, abrir SQL, novo script SQL, tem outras opções aqui, mas nessa barra, essas aqui que serão as importantes para a gente. Aqui do lado esquerdo, temos mais itens. Temos essa parte de navegador de banco de dados, teremos todos os bancos de dados criados ou que temos conexão. E na parte aqui de projetos, nós teremos os projetos que podem estar aqui abertos ou não.

00:00:06:46 - 00:00:07:44

E olha que legal, quando a gente instala o DBeaver vem um sample. Aqui na parte do navegador de banco de dados vem o DBeaver sample database. Esse sample é do SQLite. O SQLite que é um sistema de gerenciamento de banco de dados relacional que implementa uma versão mínima do SQL. A principal diferença entre o SQLite e outros sistemas de gerenciamento de banco de dados é que o SQLite é um banco de dados self-contained. autocontido, o que significa que toda biblioteca SQL e o banco de dados estão contidos em uma única biblioteca ou arquivo. Isso torna o SQLite fácil de usar e adequado para aplicativos que precisam de um banco de dados leve e simples de implementar. Bom, primeiro vamos à explicação desse banco, desse sample que vem aqui. Clicar aqui a gente precisa fazer o download dele, vamos fazer aqui o download. Rapidinho, só para poder aparecer as tabelas. Então, clicando aqui na aba, a gente consegue ver, clicando aqui em tabelas, a gente vê todas as tabelas que tem nele.

00:00:07:44 - 00:00:08:56

Esse banco de dados, geralmente, é chamado de Chinook, e é comumente usado de exemplo em vários softwares de gerenciamento de banco de dados. Esse banco é fictício e foi projetado para representar um sistema de gerenciamento de vendas de uma loja de música, bem parecido com o exemplo que tivemos na aula anterior, mas agora temos várias entidades, várias tabelas. O banco de dados, esse banco de dados, é útil para praticar consultas de SQL e entender como os diferentes elementos de um sistema de gerenciamento de vendas de música podem ser representados e relacionados em um banco de dados relacional. Então, se a gente clica aqui nesse item de tabelas, vai aparecer aí todas as tabelas que a gente tem nesse banco. Bom, a gente tem aqui tabelas como álbum, artista, cliente, funcionário, faixas, fatura, gênero. Vamos clicar aqui nessa tabela de álbum. Vai abrir essa aba com algumas informações da tabela. Vejam aqui, nós temos as colunas dessa tabela, que são Álbum ID, título, Artist ID e essa coluna não nomeada. Depois a gente tem o índice, a numeração de cada coluna. Depois a gente tem o tipo de dado que cada coluna aceita, tipo inteiro, que é um número, esse nVarchar, que é texto, o outro inteiro, e o blob, que é tipo imagem.

00:00:08:56 - 00:00:10:09

Depois a gente tem o cumprimento, no caso nenhuma dessas colunas está delimitando o cumprimento. Temos informações de não nulo, isso significa que essa coluna aceita ou não ter nulos. No caso dessa tabela apenas a coluna 1 aceita nulos, essa coluna que seria com imagens. Temos essa coluna de auto-incrementar, que é quando à medida que entra um novo item na tabela, ou seja, uma nova entrada, é criado um novo número para ele, mas nenhuma coluna foi selecionada dessa forma. Enfim, temos várias informações da tabela nessa aba de propriedades. Na aba dados, aqui do lado, se a gente clicar lá, a gente consegue ver alguns dados dessa tabela. Aqui a gente tem algumas entradas. Vocês reconhecem algum título de algum artista? Vocês podem dar uma olhadinha. Olha, tem Cássia Eller. Lembrando em uma aula anterior, nós vimos que uma entidade é geralmente representada por uma tabela em um banco de dados relacional. Cada linha da tabela representa uma instância ou ocorrência dessa entidade e cada coluna da tabela representa um atributo ou característica dessa entidade. Um atributo é uma característica específica ou uma propriedade que descreve uma entidade.

00:00:10:09 - 00:00:11:27

Aqui na barra de ER diagrama, clicando lá, a gente tem um diagrama super legal de relacionamento e entidades dessa tabela com outras tabelas do banco de dados. Lembrando o que vimos sobre relacionamentos no diagrama da tabela álbum, agora podemos analisar o relacionamento dessa tabela com as outras. Então, vamos pensar o relacionamento de álbum aqui para track, para a tabela de track. Temos um losango aqui. Isso significa que é de 1 para N. Um álbum pode ter uma track, uma música mais fácil, ou várias músicas, o que é mais comum, um álbum com várias músicas. Agora vamos entender o conceito de chave primária, para a gente entender a partir do que essas tabelas se relacionam. Bom, imagine que nós estamos trabalhando no banco de dados dessa loja de música. E temos essas três tabelas, track, álbum e artista, artist. Nós precisamos de alguma forma conectar essas tabelas para a gente conseguir saber quais faixas são de qual álbum e qual álbum é de qual artista. E é aí que entra isso de chave primária. Em cada tabela teremos uma coluna especial que é chamada de chave primária.

00:00:11:27 - 00:00:12:43

Essa coluna contém valores únicos para cada registro na tabela. É como um número de identificação único para cada linha da tabela. Por exemplo, na tabela álbum, temos aí a coluna de álbum ID. Tem um símbolo de chave anexada aqui, bem pequenininho. Essa é a nossa coluna especial de chave primária, o que significa que cada item dessa coluna é único. Quando você tem uma chave primária em uma tabela, você pode utilizá-la para criar uma conexão entre essa tabela e outras tabelas. Por exemplo, na tabela de track, temos a coluna de álbum ID. que indica qual álbum é. Essa coluna de álbum de lá da tabela track é chamada de chave estrangeira, porque ela vai conectar com a outra tabela, que é a tabela de álbum. Então, ela se relaciona com a chave primária da tabela de álbum. Essa relação entre chave primária em uma tabela e a chave estrangeira em outra tabela é o que permite que você conecte as informações de diferentes tabelas em um banco de dados relacional. Isso facilita a busca e organização dos dados, permitindo que você saiba, por exemplo, qual música, faixa, pertence a qual álbum, simplesmente olhando para os IDs correspondentes das tabelas.

00:00:12:43 - 00:00:13:43

Entre a tabela Álbum e artist, temos a coluna de Artist ID, aqui. e é a chave estrangeira na tabela de álbum. Ou seja, a Artist ID é a chave primária na tabela de artista e a chave estrangeira na tabela de álbum, permitindo que essas duas tabelas se comuniquem. Legal, né? Clicando em cima do nome da coluna a gente consegue ver qual outra coluna a tabela está associada. Por exemplo, eu cliquei aqui em Artis ID, já ficou verdinho aqui e verdinho aqui em álbum. porque, né, está se conectando com essas duas colunas. Se eu clicar aqui em álbum ID, vai se conectar aqui com álbum ID de track. Bom, lá na aba de propriedades, nós conseguimos ver as chaves únicas e estrangeiras da tabela. Ó, clicando aqui em propriedades, a gente tem aqui chave única, tá falando aqui qual que é, é álbum, e aqui chave estrangeira. Ok. Bom, navegamos aí por pontos principais da tabela de álbum, peço que vocês explorem as outras tabelas, e depois de explorarem as outras tabelas, deem um clique aqui, duplo, em tabelas mesmo.

00:00:13:43 - 00:00:14:21

Aqui, onde a gente viu as informações, a gente consegue ver o nome de todas as tabelas que esse banco de dados tem. E depois vocês cliquem aqui em ER diagrama. E aqui vocês vão ter o diagrama de todo o banco, incluindo as relações entre as tabelas. Assim, podem treinar através de qual coluna cada tabela se relaciona, onde está a chave primária e onde está a chave estrangeira entre elas. Enfim, se divirtam por aí. E é isso na aula de hoje, pessoal. Nós conhecemos sobre o DBeaver, sobre o relacionamento das tabelas e vimos um exemplo de banco de dados. Até mais.