Para construir consultas em um banco de dados, é fundamental conhecer o banco com que se vai trabalhar. Já sabemos que toda base é composta por tabelas, que cada uma delas possui uma chave primária, e que se relacionam por meio de um mecanismo de chaves estrangeiras. Portanto, ao trabalhar com um banco, é importante identificar as tabelas que estão à disposição, o que significa cada campo e qual o seu tipo, quais são as chaves primárias e como as tabelas se relacionam entre si (ou seja, entender quais são as chaves estrangeiras).

Para obter todas essas informações, uma opção é consultar a **documentação** do banco de dados. Em empresas pequenas, é comum que a pessoa que irá consultar o banco seja a mesma que o projetou (e, por isso, esse documento talvez não tenha sido elaborado). Independentemente do tamanho da empresa, nem sempre teremos à disposição uma documentação satisfatória.

Por meio do MySQL Workbench também temos a possibilidade de conhecer tudo que está incluso no nosso banco. Então, nesse vídeo, vamos abrir o MySQL Workbench, entrar na nossa conexão e verificar o que temos na nossa base "sucos\_vendas".

Começaremos clicando na seta à esquerda do nome "sucos\_vendas" para abrir um menu com mais informações sobre a base. Veremos quatro grupos de componentes: *tables*, *views*, *stored procedures* e *functions* - nosso foco agora serão as tabelas (*tables*).

Ao abrir o menu *tables*, notaremos que há cinco tabelas: "itens\_notas\_fiscais", "notas\_fiscais", "tabela\_de\_clientes", "tabela\_de\_produtos" e "tabela\_de\_vendedores".

Pelos próprios nomes, já temos uma boa noção do que elas tratam: "tabelas\_de\_vendedores" é a lista de vendodores, "tabela\_de\_produtos" são os produtos vendidos pela empresa de sucos, "tabela\_de\_clientes" é a lista de clientes, "notas\_fiscais" e "itens\_notas\_fiscais" são referentes às notas fiscais.

Normalmente uma nota fiscal é composta por um cabeçalho seguido de uma lista dos itens comprados pelo cliente, então é comum representar uma nota fiscal com duas tabelas - uma com os cabeçalhos e outra com os itens.

Quando eu nomeei essas tabelas, me preocupei mais com a didática do que com a nomenclatura, então ficou fácil deduzir o que significa cada tabela. Em projeto reais, no entanto, você não vai encontrar esses nomes amigáveis. Você pode se deparar com nomenclaturas como "tb018665" ou "tabCli", nomes que não te darão muitas pistas, principalmente porque, dentro das empresas, existem regras de nomenclaturas para as tabelas.

Assim sendo, é interessante usar o MySQL Workbench para descobrir o que cada tabela contém e como ela se relaciona com outras tabelas. A seguir, vamos descobrir de que trata "tabela\_de\_clientes".

Abrindo o menu de "tabelas\_de\_clientes", veremos quatro subgrupos: *Columns* (colunas, que também podemos chamar de "campos"), *Indexes* (índices), *Foreign Keys* (chaves estrangeiras) e *Triggers*.

Nesse curso, não vamos nos aprofundar em *triggers*, pois não têm uma relação estreita com consultas SQL, e também não focaremos em índices. Como eu já havia explicado sobre organização de banco de dados em um curso anterior, vale lembrar que índices são estruturas que nos auxiliam a encontrar elementos mais rapidamente. Isso é uma regra no MySQL: para toda chave primária ou estrangeira, o MySQL automaticamente criará um índice. Isso ocorre porque a chave primária é uma coluna que não pode se repetir, assim, toda vez que eu insiro um item novo em uma tabela, o banco de dados faz uma busca nessa tabela para verificar se esse item já existe, e o índice facilitará muito nessa busca. O mesmo vale para a chave estrangeira: se há duas tabelas que se relacionam e eu quero incluir um novo registro na tabela-filha cujo campo tem uma relação de chave estrangeira com a tabela-pai, essa inclusão só é possível se esse campo existir na tabela-pai. Então, o banco de dados fará uma busca e só permitirá o novo registro se essa condição for atendida. Também nesse caso o índice funciona como um facilitador.

Em "tabela\_de\_clientes", vamos abrir o subgrupo *Columns* e conferir uma lista das colunas que compõem essa tabela ("CPF", "NOME", "ENDERECO\_1", "ENDERECO\_2", "BAIRRO", "CIDADE", "ESTADO", "CEP", "DATA\_DE\_NASCIMENTO", "IDADE", "SEXO", LIMITE\_DE\_CREDITO", "VOLUME\_DE\_COMPRA" e "PRIMEIRA\_COMPRA"). Se você clicar sobre o nome de uma das colunas, é possível ver mais detalhes no painel esquerdo inferior. Dentre essas informações, o que mais nos interessa é saber se o campo em questão é uma chave primária ou não, e com que tipo de dado estamos lidando. Clicando na coluna "CPF", por exemplo, verificamos que se trata de uma chave primária, pois vem acompanhada da sigla **PK** (em inglês, *primary key*), e que é um **varchar(11)**, ou seja, um texto de 11 caracteres.

Em seguida, exploraremos o subgrupo *Indexes* de "tabela\_de\_clientes", no qual podemos ver os índices *states*. Nesse caso, essa tabela tem apenas chave primária, então há apenas um índice associado à chave primária. Não há chaves estrangeiras, ou seja, essa tabela não tem um relacionamento com nenhuma outra - não é filha de uma tabela-pai. E também não há *triggers*.

Diferentemente disso, ao abrir *indexes* da tabela "notas\_fiscais", verifica-se que existem três índices. O de nome "*PRIMARY*" é referente à chave primária e os índices "MATRICULA" e "CPF" me dão pistas de que estão relacionados a chaves estrangeiras. Como esperado, abrindo o subgrupo *foreing keys* dessa tabela, veremos duas chaves estrangeiras cujos nomes o banco produziu automaticamente. Selecionando uma dessas chaves, são mostradas informações mais detalhadas, no painel abaixo, sobre o relacionamento entre tabelas.

Assim sendo, essa é uma das maneiras de conhecer o seu banco de dados. Mas existe também uma forma mais visual de analisar as tabelas e seus relacionamentos. Inclusive, muitas pessoas preferem ver a documentação por meio dessa representação mais visual e, no caso de bancos com poucas tabelas, até imprimir esse esquema e deixar ao alcance para consultas.

Para gerar esse esquema visual, vamos até a barra superior do programa e selecionamos **"Database > Reverse Engineer..."**. Em outras palavras, faremos uma engenharia reversa no nosso banco que já existe. Na caixa de diálogo que será aberta, selecionamos a nossa conexão usual, clicamos em "*Next*" duas vezes, selecionamos a base "sucos\_vendas", clicamos em *"Next"* mais duas vezes e, em seguida, em *"Execute"*.

Com isso, será gerado um diagrama que representa as tabelas e suas relações. Você pode dar zoom, arrastar e organizar os elementos do esquema da maneira que preferir.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Temos então várias tabelas, em cada uma delas há informações sobre campos e tipos. A tabela de produtos, a de clientes e a de vendedores são cadastrais, ou seja, nenhuma outra "chega" nelas.

Temos também a tabela "notas\_fiscais" com dados do cabeçalho das notas fiscais. Este deve conter o número identificador da nota, o cliente e o vendedor. É essencial lembrar que o cliente e o vendedor que eu coloco no cabeçalho da nota fiscal precisam necessariamente existir nas tabelas cadastrais! Então, se você clicar sobre a linha que conecta "notas\_fiscais" e "tabela\_de\_clientes", é possível ver em destaque quais são os campos que estão relacionados entre essas duas tabelas (no caso, trata-se do "CPF").

Entre "notas\_fiscais" e "tabela\_de\_vendedores", os que campos que se relacionam são os de "MATRICULA". Entre "notas\_fiscais" e "itens\_notas\_fiscais" são os campos de "NUMERO" (da nota fiscal). E entre "itens\_notas\_fiscais" e "tabela\_de\_produtos", os campos de "CODIGO\_DO\_PRODUTO".

Dessa forma, podemos utilizar esse diagrama como guia para as consultas. Normalmente, quando se tem um analista responsável por um banco de dados numa empresa, ele acaba até decorando as tabelas, os campos e os relacionamentos e sequer precisa consultar esse diagrama de entidades! Mas essa é uma ótima forma de melhor se familiarizar com seu banco de dados.