Agora que já sabemos os tipos, vamos criar a primeira tabela no banco de dados. Lembrando que estamos fazendo um projeto de banco de dados para uma empresa de suco de frutas.

Conversando com a empresa, nos foi passado que gostariam de armazenar no MySQL os cadastros dos clientes com as seguintes informações: o CPF do cliente, o nome completo, o endereço completo (rua, bairro, cidade, estado e CEP), a data de nascimento, a idade, o sexo, o limite de crédito para efetuar a compra dos produtos, o volume mínimo de sucos que o cliente pode comprar e se já foi realizado a primeira compra.

Analisando essas informações, percebemos que cada uma é uma coluna e será necessário definir o tipo de cada campo. Vamos, então, para o Workbench que já temos o banco de dados **sucos**, mas sem nenhuma tabela ainda.

Vamos criar a primeira tabela, então. Na parte superior esquerda da tela, clique no ícone de um papel escrito SQL com um sinal de (+) embaixo. Significa que estamos criando um script SQL e irá abrir uma nova área que digitaremos os comandos.

Para criarmos uma tabela, basta escrever CREATE TABLE, note que o Workbench nos dá algumas opções do que pode ser o comando que estamos inserindo. Seguindo o comando, é preciso inserir um nome para a tabela, que vai ser "tbcliente". Neste treinamento não vamos nos preocupar tanto com nomenclatura, mas há boas práticas de como criar nomenclatura para tabelas, campos, índices, chave primária e estrangeira, e assim por diante.

Uma imagem contendo Texto

O conteúdo gerado por IA pode estar incorreto.

Seguindo com a nossa tabela, além do nome é preciso estabelecer as colunas e seus respectivos tipos. Clicando em "Enter" e abrindo um parêntese para especificar os campos, vamos iniciar pelo CPF, o número que identifica o cliente, pode ser que tenha pontos, traços e às vezes inicia com zero. Por essa razão, por mais que o CPF sejam números, é melhor armazenar como texto. Visto que se gravar como número no momento de gravar, os zeros no início serão **truncados**. Exemplificando, se tivermos um CPF com o número 00388323102, o banco de dados guarda 3888323102. Por isso, vamos considerar o CPF com o tipo VARCHAR.

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Após a vírgula vamos inserir os próximos campos. O "NOME" também será VARCHAR e, por ter nomes compridos, vamos estabelecer 100 caracteres para reservar o nome. Continuando, temos "ENDERECO" que também devemos considerar haver nomes de ruas grandes, inclusive, às vezes é viável criar dois campos para gravar o endereço. Vamos inserir da seguinte maneira ENDERECO1 VARCHAR(150),, ENDERECO2 VARCHAR(150),, e depois o bairro BAIRRO VARCHAR(50),, a cidade CIDADE VARCHAR(50),, lembrando sempre de inserir uma vírgula em cada linha.

O próximo campo pedido pela empresa é o "ESTADO", que também será do tipo VARCHAR ESTADO VARCHAR(50), em seguida o CEP VARCHAR(8),. Na idade, como dificilmente as pessoas vão ter mais de 150 anos, não é preciso colocar um INT, um SMALLINT já atende o que precisamos IDADE SMALLINT,, o sexo, que vai ser uma letra M (masculino) e F (feminino), ficando SEXO VARCHAR(1),.

No nosso campo de limite de crédito estamos nos referindo a valores em dinheiro, valores decimais, como, limite de crédito de R$ 1.627.438,32. Então, vamos colocar LIMITE\_CREDITO FLOAT,, caso queira deixar com espaço entre a palavra limite e crédito, basta inserir crase no início e fim, assim limite credito.

Perceba que não especificamos o número de caracteres no FLOAT, mas poderíamos colocar FLOAT(10, 2),. Contudo, mesmo sabendo que dinheiro tem duas casas decimais, vamos representar o número sem fixar o número de casas decimais.

**Opinião do instrutor:** quando estamos falando de banco de dados, estamos falando sobre armazenar informação. Quando exibo a informação é que posso mostrá-la do jeito como ela deve ser representada. Ou seja, na minha opinião, você não precisa se preocupar com o formato do valor, do número, ou do texto no momento de armazená-lo no banco de dados, mas sim no momento de exibir em um relatório, em um programa. E aí você tem, não somente no SQL, mas também nas linguagens de programação com as quais vai trabalhar em conjunto com o banco de dados, de exibir a informação do jeito como ela tem que aparecer. Isto é, posso gravar um número com dez casas decimais, mas no momento em que eu for exibir, forço a exibir só com duas casas.

Já o volume de compras é expresso em litros, então, também não vamos nos preocupar se tem ou não casas decimais: VOLUME\_COMPRA FLOAT,. Finalmente, o último campo, se o cliente já realizou uma primeira compra ou não. Neste caso vamos usar o campo BIT de um espaço apenas, para eu poder ter o 1 (realizou a primeira compra) ou o 0 (ainda não realizou a primeira compra): PRIMEIRA\_COMPRA BIT(1)), esse último parêntese fecha o que inseri no início da inclusão de todas as colunas.

Código completo:

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

Ficou assim, CREATE TABLE, logo depois o nome da tabela, abro parênteses, coloco todos os campos separados por vírgula, o nome do campo e o tipo, e fecho o parêntese. Vamos rodar agora, selecionando na parte superior à área que digitamos o comando, o ícone de raio. Na parte de "Action Output" é exibida uma mensagem informando que o comando foi executado com sucesso.

Na região de "Schemas", clicando em **sucos** com o botão direito do mouse e escolhendo a opção "Refresh All", note que a tabela "tbcliente" foi criada.

Fizemos o primeiro comando de SQL para criar uma tabela. Espero que tenham entendido e gostado, até a próxima aula!