



# Introdução a Bancos de Dados

Prof. Nazareno de Oliveira Pacheco  
[nazareno.pacheco@prof.sc.senac.br](mailto:nazareno.pacheco@prof.sc.senac.br)

# Objetivo da disciplina

- Ensinar os alunos a modelar e manipular informações em bancos de dados relacionais utilizando a linguagem SQL, reforçando sempre a utilização de boas práticas.

# Agenda

- O que é Banco de Dados
- Como ele funciona
- Desafio
- Exercícios

Senac

# O que são Banco de Dados?

- Uma base de dados é um conjunto de arquivos relacionados entre si. (CHU 1983)
- Uma base de dados é uma coleção de dados operacionais armazenados, usados pelos sistemas de aplicação de uma determinada organização. (Date, 1985)
- Uma base de dados é uma coleção de dados relacionados. (Elmasri e Navathe 1989)

# Armazenando informações

- Atualmente, os programas que fizemos perdiam todas as informações quando eram finalizados
- Porém, em muitos casos precisamos guardar estas informações para utilizar mais tarde
- Para utilizar o termo técnico, podemos dizer que precisamos “***persistir***” estas informações

# Armazenamento próprio?

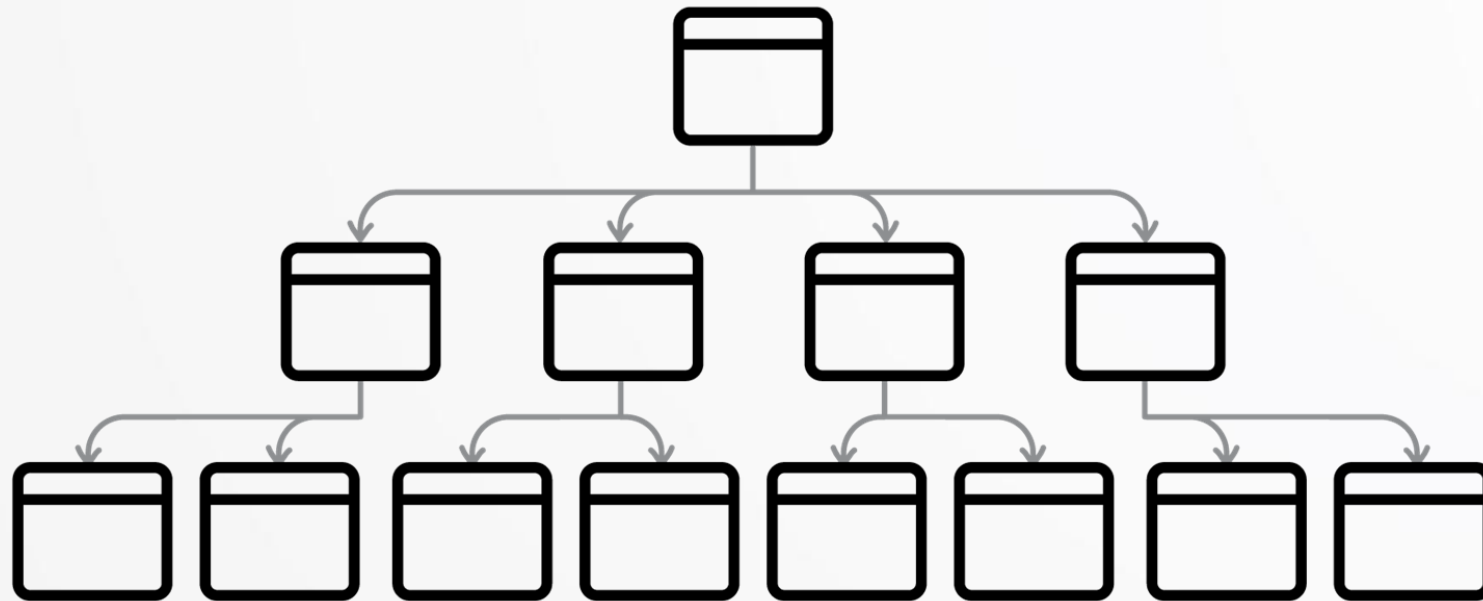
- Uma solução possível é gravar todas estas informações em arquivos próprios
- Porém, isto pode trazer problemas:
  - **Desempenho**: à medida em que a quantidade de informações cresce, nosso programa vai ficando mais lento
  - **Complexidade**: à medida em que nosso sistema evolui, será necessário armazenar mais informações, e isto torna o armazenamento complexo

# Bancos de Dados

- Uma das soluções mais comuns para este tipo de problema é a utilização de um **Sistema de Gerenciamento de Bancos de Dados**, ou SGBD (ou, em inglês, *Database Management System*, DBMS)
- Um SGBD é uma aplicação que oferece mecanismos para armazenamento, gerenciamento e recuperação de informações
- Estas aplicações são projetadas para garantir o gerenciamento de grandes quantidades de informação de forma eficiente e organizada

# Tipos de Bancos de Dados

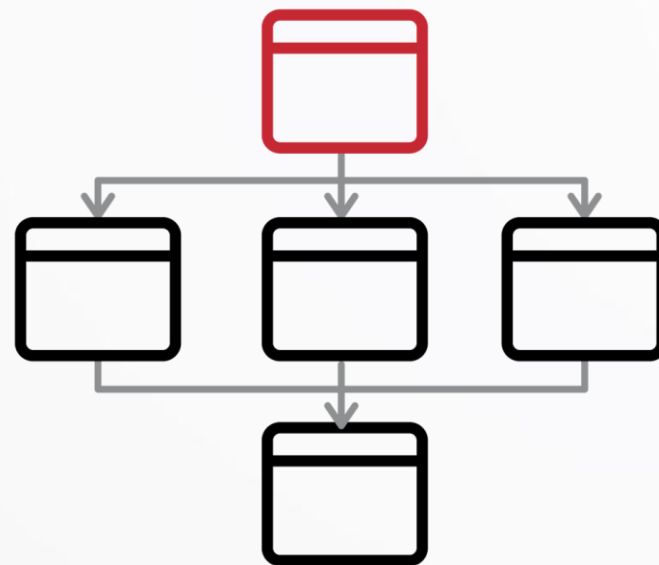
- **Hierárquicos:** O modelo hierárquico organiza os dados em uma estrutura do tipo árvore e estabelece relações de “um para muitos”, pois cada registro tem um único “pai” ou raiz. Por conta disso, costuma-se dizer que o banco de dados hierárquico possui uma relação de campos “pais” para campos “filhos”. Foram muito utilizados antigamente, mas são raros nos dias de hoje. São muito eficientes, mas sua estrutura é bastante rígida.





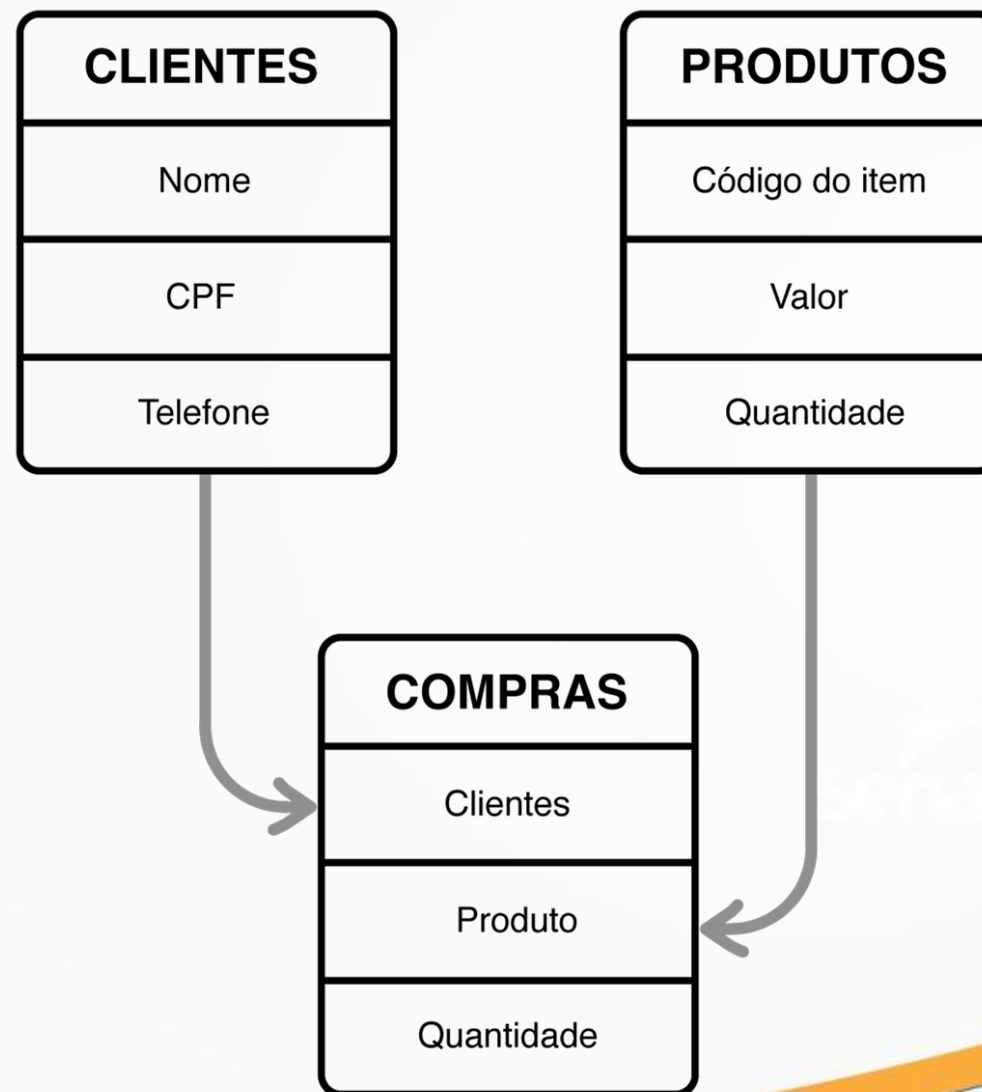
# Tipos de Bancos de Dados

- **Em Rede:** O modelo rede foi desenvolvido com base na estrutura de banco de dados hierárquica, sendo uma espécie de evolução, tendo em vista que otimizou o processo de consulta. Para isso, foi estabelecida uma relação de “muitos para muitos”, na qual um registro “filho” pode se conectar a vários registros “pais”. São uma evolução dos modelos hierárquicos. Oferecem um pouco mais de flexibilidade, mas também não são mais muito utilizados hoje



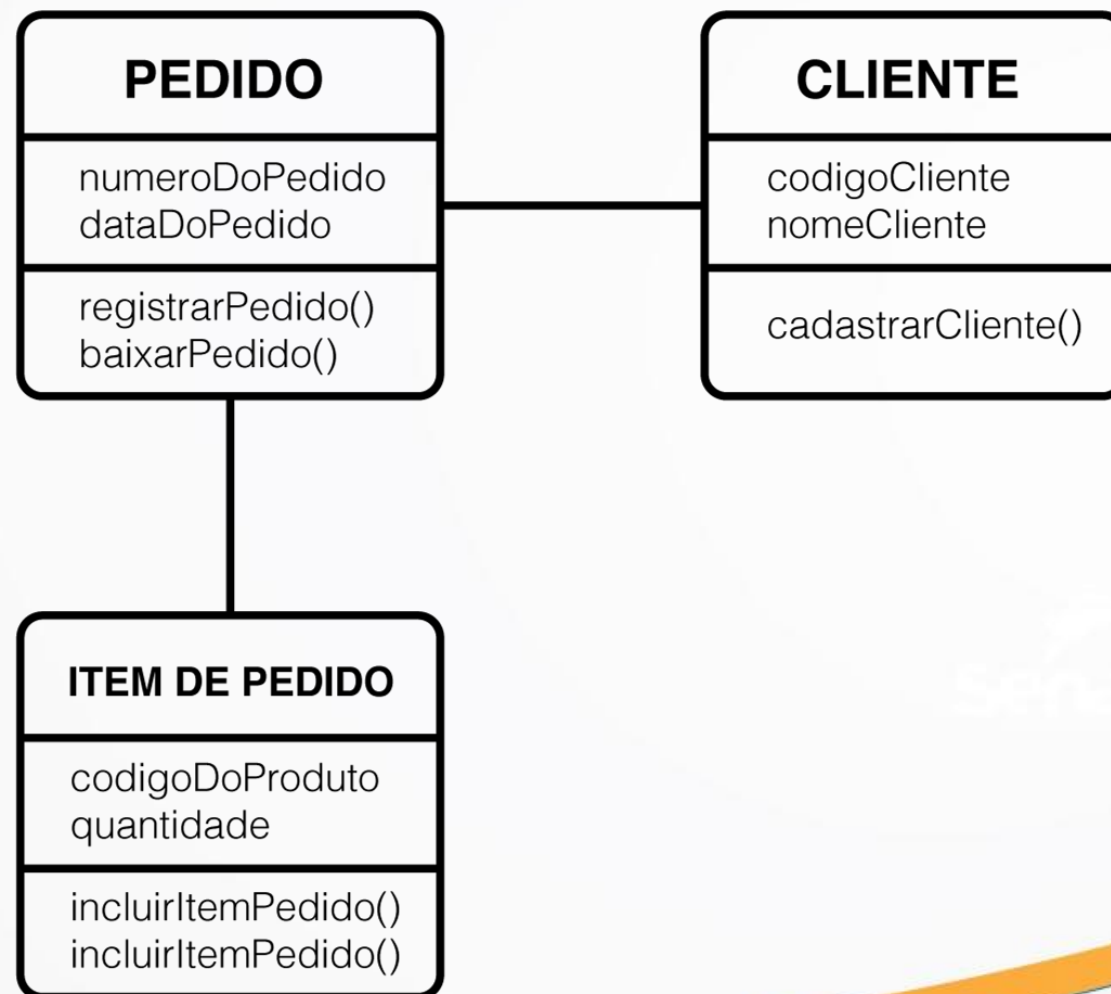
# Tipos de Bancos de Dados

- **Relacionais:** O modelo relacional é um dos mais utilizados atualmente. Sua estrutura classifica dados em tabelas – também chamadas de relações – compostas por colunas e linhas. Esse tipo de banco de dados recebe o nome de relacional, pois sua estrutura é projetada especificamente para dados que conversam entre si.
- **O foco da nossa disciplina é este tipo de banco**



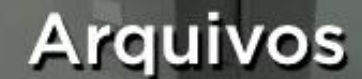
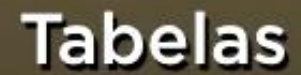
# Tipos de Bancos de Dados

- **Orientados a Objetos:** aplicam as estratégias de orientação a objetos às bases de dados. Foram considerados o “próximo passo” em bancos de dados, mas na prática seu uso simplesmente não se popularizou



# Bancos de Dados Relacionais

- No início da década de 80, houve o estouro do modelo de Bancos de Dados Relacionais
  - Este modelo é baseado nos trabalhos do Dr. E. F. Codd, que trabalhava na IBM e procurava soluções para os problemas de bancos de dados hierárquicos e em rede.
- Bancos de dados relacionais armazenam as informações em tabelas



# Tabelas

- Uma tabela de um banco de dados pode ser comparada com uma planilha Excel
- Tabelas são compostas por diversas linhas e colunas
  - Cada linha de uma tabela corresponde a um registro diferente
  - Cada coluna representa um atributo diferente de um registro que compõe nossa tabela

# Exemplo de uma tabela

NOME	TELEFONE	EMAIL
Jon Snow	9999999	lordsnow@thewall.com
Petyr Baelish	2222222	littlefinger@harrenhal.com
Ned Stark	9876543	winteriscomming@winterfell.com
Tyrion Lannister	3334444	the_imp@casterlyrock.com



# Mais que uma planilha

- Além de armazenar informações como uma planilha, um banco de dados fornece muito mais funcionalidades, como, por exemplo:
  - Seleção de registros específicos baseados em um conjunto de critérios informados
  - Realização de cálculos de informações agregadas (somatório, média...)
  - Expressar relacionamentos entre tabelas diferentes



# SQL

- Existe uma linguagem padronizada para manipulação de informações em bancos de dados, o **SQL** (Structured Query Language, ou em português, Linguagem Estruturada de Consulta)
- Foi desenvolvida inicialmente pela IBM, e logo começou a ser utilizada por outros fornecedores

# Padronização

- Em 1986, a ANSI (American National Standards Institute) padronizou o uso da linguagem. Mais tarde, em 1987, a ISO fez o mesmo
  - O objetivo da padronização do SQL é garantir maior interoperabilidade entre SGBD's distintos
- Este padrão foi revisto em 1992, 1999 e 2003, adicionando novos recursos à linguagem

# Padronização

- Nem todos os SGBDs seguem o padrão ANSI do SQL estritamente
- Por esta razão, cada SGBD possui particularidades extensões próprias
- Ainda assim, a padronização ajuda na grande maioria das tarefas

# Escolhendo um SGBD relacional

- Atualmente existe um grande número de opções de SGBD's disponíveis
- Existem diversas características a serem consideradas
  - Desktop: Um SGBD Desktop pode rodar na própria máquina pessoal de um usuário. São leves, e oferecem um facilidades

# Desktop X Servidor

- Desktop:
  - Um SGBD Desktop pode rodar na própria máquina pessoal de um usuário.
  - Oferecem um facilidades para usuários menos experientes
  - Mesmo quando não são livres, costumam ser baratos
  - Não comportam bem quantidades de dados maiores
  - Exemplos: MS Access

# Desktop X Servidor

- Servidor
  - Rodam em uma aplicação à parte, muitas vezes em um computador diferente
  - São soluções bem mais robustas, suportam uma carga de trabalho bem maior
  - Proporcionam melhor escalabilidade, ou seja, é possível melhor ainda mais o desempenho utilizando mais máquinas
  - São mais caros
  - São mais difíceis de se configurar

# Proprietários X Livres

- Proprietários
  - São desenvolvidos por empresas específicas
  - Possuem um custo de licença para a utilização (normalmente alto)
  - Costumam ser bastante robustos
  - Exemplos: Oracle, MS SQL Server, IBM DB2

Senac

# Proprietários X Livres

- Livres
  - Normalmente são desenvolvidos por uma comunidade de pessoas independente
  - Não possuem custo de utilização
  - Podem não ser tão robustos quanto alguns SGBD's proprietários (mas isto é contestável)
  - Exemplos: PostgreSQL, MySQL



# Escolhendo um SGBD relacional

- Em nossa disciplina, vamos utilizar o MySQL
- Razões:
  - Robusto
  - Bastante utilizado
  - Sem custos
- Outra opção que atenderia os mesmos requisitos seria o PostgreSQL

# Instalando o MySQL

- O download MySQL pode ser realizado no seguinte endereço
  - <https://dev.mysql.com/downloads/installer/>
  - Tome cuidado para utilizar a versão correta do seu sistema operacional (principalmente com a questão 32bit x 64bit)
  - Após o download ter sido realizado, execute o instalador  
**COMO ADMINISTRADOR**

Senac

# MÃO NA MASSA



# Criando tabelas

- Para criar uma tabela, é preciso definir quais são as colunas que desejamos utilizar
- Por exemplo: crie uma tabela para armazenar os dados de funcionários. Cada funcionário tem um número de matrícula, nome e salário

Senac

# Criando uma base de dados

- Para criar uma base de dados utilize o comando CREATE DATABASE
- A sintaxe deste comando é bastante simples

```
CREATE DATABASE nome_da_base;
```

# Criando uma base de dados

- Vamos criar uma base de dados chamada teste1
- Digite o comando

```
CREATE DATABASE TESTE1;
```

- Para executar este comando, clique no botão Execute Query
- Este comando vai criar uma nova base de dados neste servidor MySQL



# Criando tabelas

- Uma vez que nossa base de dados foi criada, podemos selecioná-la para adicionar novas tabelas
- Primeiro, feche a janela SQL da base MySQL
- Em seguida, selecione o item Databases e clique no botão Refresh

Senac

# Criando tabelas

- Por exemplo:
  - Crie uma tabela para armazenar os dados de funcionários. Cada funcionário tem um número de matrícula, nome e salário
- Neste caso, precisamos de uma tabela com três colunas:
  - Número de matrícula
  - Nome
  - Salário

Senac



# Criando tabelas

- É necessário definir um nome e um tipo de dados para cada coluna de nossa tabela:
- **NUMERO\_MATRICULA**: é um valor numérico com 4 números. Vamos utilizar o tipo **DECIMAL (4)**
- **NOME**: é um valor textual com no máximo 100 caracteres. Vamos utilizar **VARCHAR(100)**
- **SALARIO**: é um valor numérico com no máximo 8 dígitos, sendo 2 após a vírgula. Vamos utilizar **DECIMAL (8,2)**
- Observação: Vamos falar mais sobre os tipos de dados nas aulas a seguir.



# Criando tabelas

- O comando para criação de tabelas é o CREATE TABLE
- A sintaxe básica deste comando é a seguinte:

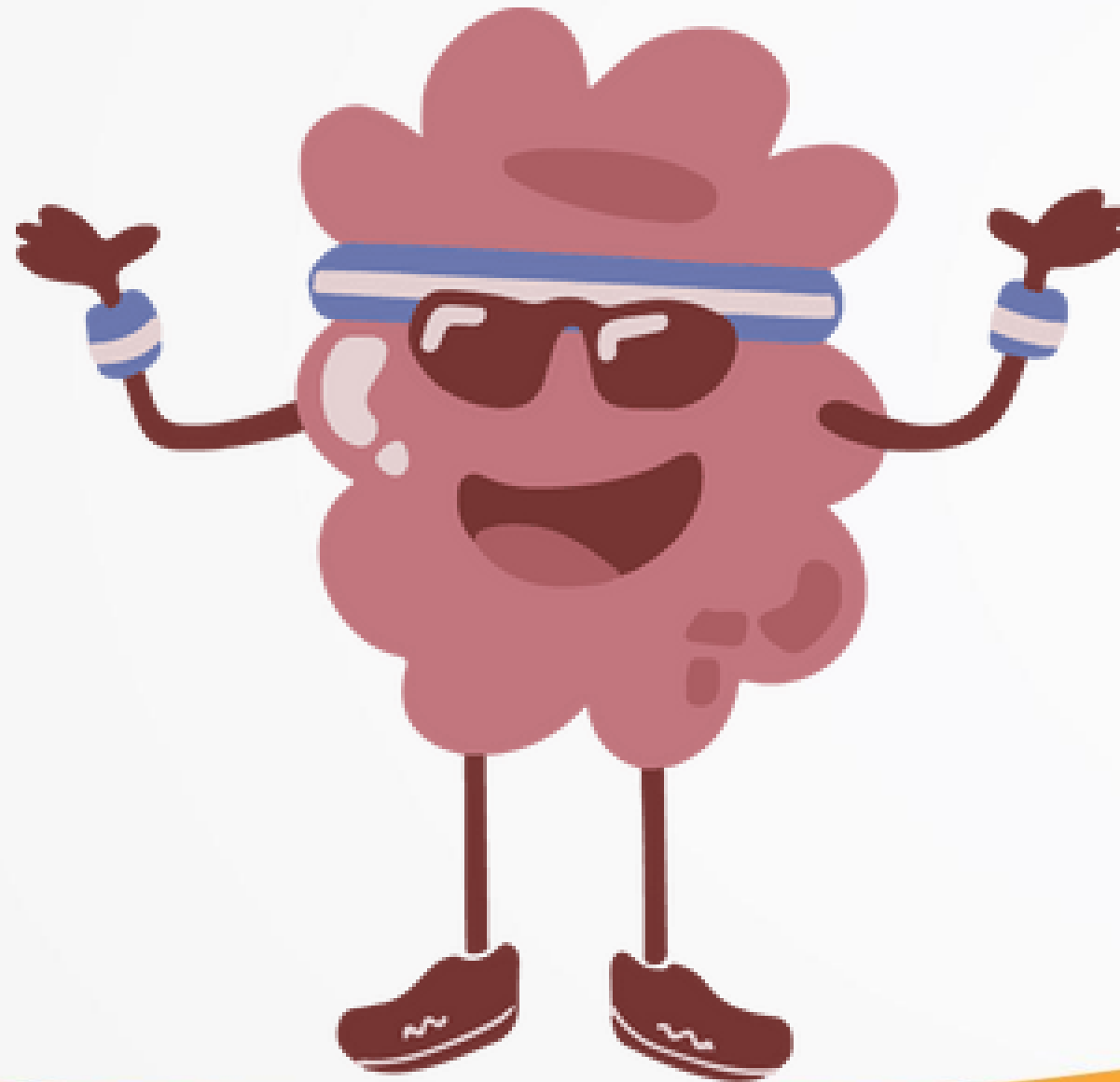
```
CREATE TABLE (  
    nome_da_coluna1        tipo_coluna1,  
    nome_da_coluna2        tipo_coluna2,  
    ...  
);
```

# Criando tabelas

- Para criar nossa tabela, utilize o seguinte comando

```
CREATE TABLE FUNCIONARIO (  
    NUMERO_MATRICULA DECIMAL(4),  
    NOME VARCHAR(100),  
    SALARIO DECIMAL(8,2)  
);
```

# Exercícios



# Exercícios (1)

1. Crie uma tabela que armazene os dados de alunos. Cada aluno tem
  - CPF,
  - número de matrícula com 8 dígitos,
  - Nome,
  - E-mail
2. Crie uma tabela que armazene os dados de produtos de uma loja. Cada produto tem:
  - Código (8 dígitos)
  - Nome
  - Descrição completa
  - Valor

# Referências

- ELMASRI, Ramez; NAVATHE, Shamkant B.. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson, 2011.
- FRANÇA, Cícero Tadeu Pereira Lima; JÚNIOR, Joaquim Celestino. **Banco de dados**. 2. ed. Fortaleza, CE : EdUECE, 2015.