



ECEC – ESCOLA DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO

CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

FUNDAMENTOS DE BANCO DE DADOS

01.Introdução

Para usar um computador é necessário um software. As empresas utilizam softwares específicos para agilizar o processamento de suas atividades e também para facilitar a manipulação das informações.

Para manipular informações em um programa é necessária a utilização de bancos de dados. Sendo que um dado é um símbolo.

A informação, por outro lado, é a interpretação de um conjunto de dados.

Assim, por exemplo, a data 01/01/2000 é um dado e podemos considerar que esse dado nos remete à informação **13 anos**, que é uma forma de interpretação sobre essa data em relação à data atual.

O tratamento de informações necessita que os dados sejam armazenados em meios magnéticos para que possam ser atualizados e recuperados para se efetuar consultas e algum tipo de processamento.

Os armazenamentos automatizados eram, inicialmente, efetuados apenas em arquivos do tipo texto, por exemplo.

As linguagens menos atuais processavam apenas arquivos semelhantes a arquivos do tipo texto. Posteriormente, foram criados os bancos de dados. As linguagens atuais, além de possuírem recursos para processarem arquivos (de texto e outros), oferecem recursos para manipular dados de bancos de dados.

Mas o que é um arquivo?

É um conjunto de linhas, onde cada linha possui um conjunto de caracteres, separados ou não.

Em um arquivo texto tem-se as PALAVRAS que são formadas pelo agrupamento de caracteres.

Existem arquivos, cujas linhas possuem dados específicos como nomes, endereços, cpf e outros.

Esses dados são denominados de CAMPO.

Considere uma folha de pagamento, cujas filiais enviam seus dados para uma central de processamento. Esses dados são enviados via arquivos de texto.

Exemplo do formato dos arquivos de texto:

```
00112345601202000100000150000001
002123456012020002000000000000003
00381999901202000100001200000001
00489999901202000400000100000001
```

Posição: 1 4 10 12 16 19 30

Posição de 01 a 03: número sequencial



Posição de 04 a 09: matrícula do funcionário

Posição de 10 a 11: mês do processamento

Posição de 12 a 15: ano do processamento

Posição de 16 a 18: código do movimento (salário, hora extra, gratificação, etc.)

Posição de 19 a 29: valor do movimento

Posição de 30 a 32: quantidade do movimento (ex. 03 horas extras)

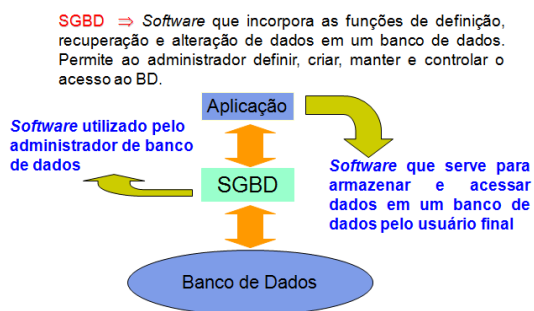
***CADA LINHA NO ARQUIVO REPRESENTA UM REGISTRO NO MOVIMENTO FINANCEIRO DA FOLHA DE PAGAMENTO**

02.Banco de dados

Um banco de dados é um conjunto de arquivos que são controlados por um programa denominado de **Sistema Gerenciador de Banco de Dados (SGDB)**.

Assim, um **banco de dados é uma coleção integrada de dados que objetivam atender uma comunidade de usuários.**

Um banco de dados pode ser acessado por diversos aplicativos (programas específicos como folha de pagamento, sistema de vendas e outros) que podem ser desenvolvidos por linguagens de programação distintas.



02.01.Algumas vantagens de um banco de dados

- Permite acesso por múltiplos programas
- Permite o acesso por diversos usuários ao mesmo tempo
- Permite restrições de acesso
- Fornecem controle de segurança a falhas

02.02.Tipos de Banco de dados

Desde a criação do primeiro banco de dados, surgiram diversas estruturas que originaram bancos de dados com funcionamentos diferenciados.

São os seguintes tipos de bancos de dados criados até o momento:

Hierárquicos

Em rede

Relacionais

Baseados em objetos.

Atualmente, os bancos de dados relacionais são os mais utilizados na atualidade, cujo funcionamento é baseado no relacionamento entre seus elementos.



Os Sistemas Gerenciadores de Banco de dados (SGBD) mais utilizados na atualidade são:

Oracle, SQLServer, MySQL, Postgres, Sybase e diversos outros. MySQL e Postgres são softwares open source.

02.02. Funcionamento básico de um banco de dados relacional

Um banco de dados é denominado de relacional porque permite o relacionamento entre seus elementos, denominados de **tabelas**.

Uma tabela é como se fosse um arquivo dentro da estrutura de um SGBD específico.

Uma tabela possui **colunas** e **linhas**.

Uma coluna representa um campo em uma tabela.

Uma linha representa um registro em uma determinada tabela.

Para melhor compreensão, imagine uma planilha eletrônica contendo o **código**, o **nome** e o **telefone** de pessoas, conforme ilustrado a seguir.

	A	B	C
1	código	nome	fone
2	1	José Bonifácio	11111111
3	2	Olavo Bilac	22222222
4	3	Machado de Assis	99999999
5			

Observe que essa planilha possui linhas e colunas. As linhas são indicadas por números à esquerda e as colunas são indicadas por letras, na parte superior. Considere que os dados dessa planilha comecem na linha 2, sendo que na linha 1 estão os nomes das colunas da planilha (**código**, **nome** e **fone**).

A linha 2 até a linha 4 possuem informações de cada pessoa. **SÃO OS REGISTROS DA TABELA.**

Cada uma dessas linhas, portanto, representam um registro.

Dessa forma, temos o 1º registro com os dados de **José Bonifácio**. O 2º registro tem os dados de **Olavo Bilac**. O 3º registro possui os dados de **Machado de Assis**.

As tabelas de um banco de dados possuem estrutura semelhante, sendo que um banco de dados permite cadastrar milhões de linhas (ou registros) em uma única tabela.

CADA TABELA EM UM BANCO DE DADOS POSSUI UM NOME ÚNICO. UM BANCO NÃO POSSUI DUAS TABELAS COM O MESMO NOME.

UMA TABELA NÃO POSSUI DOIS CAMPOS COM O MESMO NOME.

* Costuma-se denominar de **t-uplas (ou tuplas)** uma certa quantidade de linhas de uma determinada tabela.

02.03. Resumo

Em resumo, para uma tabela denominada de **Pessoa EM UM BANCO DE DADOS**, temos:



cpf	nome	telefone
123456789-00	José Bonifácio	89898989
234567890-12	Dom Pedro II	98989898
345678901-23	Machado de Assis	78787878

Diagram illustrating a table structure with columns and rows. The columns are labeled 'COLUNAS' and the rows are labeled 'LINHAS'.

Sendo que:

- Uma coluna possui um IDENTIFICADOR ÚNICO (nome) na tabela, um TIPO e TAMANHO (quantidade de caracteres que a coluna deve suportar)

• Exemplo:

TABELA Pessoa:

Nome da Coluna	Tipo	tamanho
cpf	varchar	11
nome	varchar	40
fone	varchar	8
data	date	*
sexo	int	*

* = tamanho estabelecido pelo SGBD

02.04.Relacionamento entre tabelas

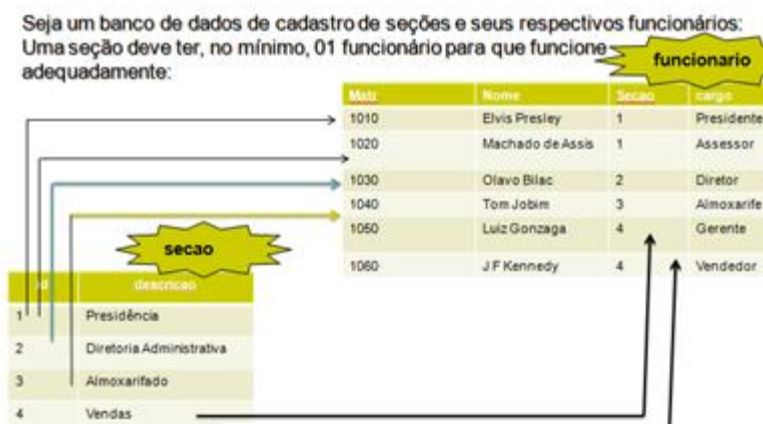
Um banco de dados permite o relacionamento entre duas ou mais tabelas.

O relacionamento entre tabelas é uma grande característica dos bancos de dados relacionais.

Um relacionamento permite a uma tabela se comunicar com outra tabela do banco de dados.

Conforme a situação, poderá existir tabelas em um banco de dados que não possuem relacionamentos com outras tabelas.

A figura a seguir ilustra duas tabelas, **funcionario** e **secao**, e um relacionamento entre elas.





Na figura acima, a tabela funcionário (sem acento) possui o código da seção (também sem acento).

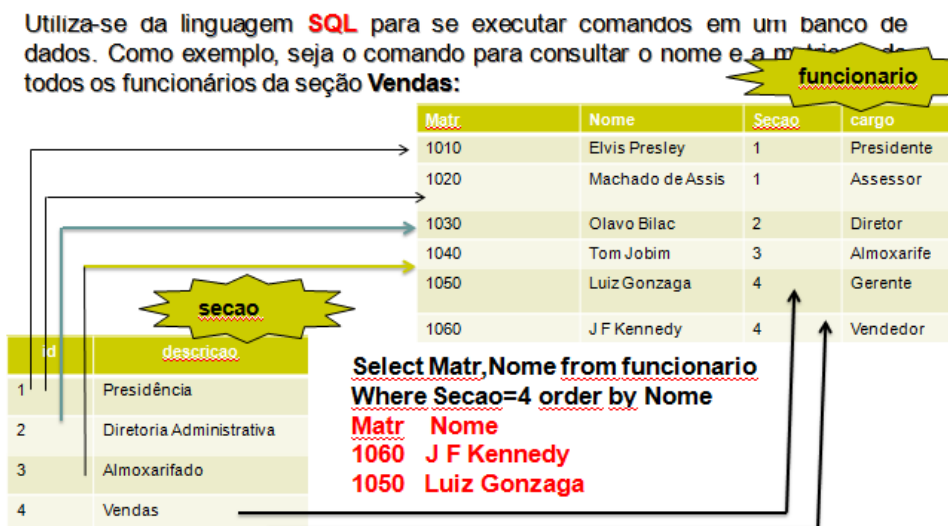
Neste caso, como exemplo, o funcionário Elvis Presley possui o código de seção igual a 1. Esse código é o correspondente ao mesmo **id** da tabela **secao**.

Portanto, o relacionamento entre tabelas em um banco de dados é muito importante, pois permite estabelecer a comunicação entre duas tabelas.

Todos os SGBD (por exemplo, MySQL e Oracle) da atualidade possuem uma linguagem de script, denominada de SQL, que permite ao programador e administrador do banco de dados executar comandos de consulta, inclusão, alteração e exclusão de dados.

Assim, com o relacionamento entre as tabelas **funcionario** e **secao** é possível, por exemplo, acessarmos os dados do funcionário **Elvis Presley** e, com o código da seção existente no registro desse funcionário (na tabela **funcionario**), podemos acessar a tabela **secao** e buscar o nome da seção através desse código. Neste caso, conforme a figura anterior, será acessada a descrição **Presidência** como nome da seção do respectivo funcionário.

A figura a seguir, ilustra um exemplo de uma instrução SQL com o comando SELECT:



02.05. Chave Primária

Denomina-se de **CHAVE PRIMÁRIA** uma informação que deve ser única para todos os registros existentes em uma tabela.

Essa informação pode ser constituída de um único campo ou mais de um campo.

Portanto, uma tabela pode ter como chave primária apenas um campo ou dois ou mais campos que, combinados, se tornam únicos em uma tabela.

Como exemplo, considere a base de dados da Receita Federal.



Neste caso, de forma hipotética e para melhor compreensão, podemos considerar que a chave primária da tabela de contribuintes seja o **cpf**, visto que duas pessoas não possuem o mesmo cpf.

Portanto, como o cpf é único para cada pessoa, este campo poderá ser a **chave primária** para a tabela de contribuintes da receita federal.

Como outro exemplo, considere a carteira de identidade (conhecida como RG) de todas as secretarias de segurança do Brasil.

Como cada Unidade da Federação emite a sua RG, poderemos encontrar duas ou mais RGs com o mesmo número.

Caso fossemos criar um banco de dados único com todas as RGs do país, não poderemos utilizar apenas o número do RG como chave primária, pois este pode se repetir para cada UF.

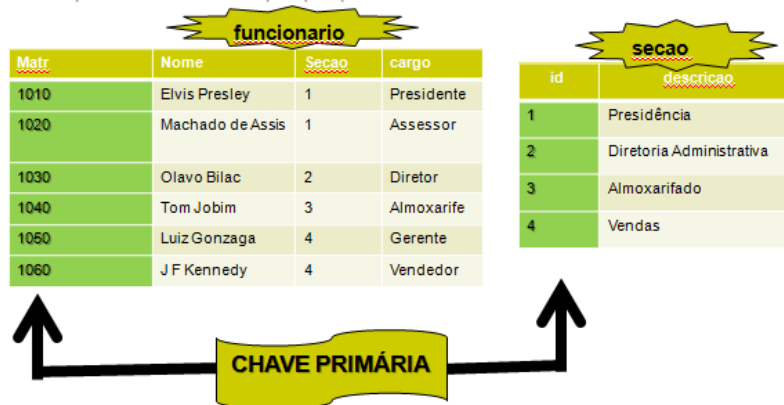
Para evitar esse problema, podemos usar dois campos: o número da RG e a UF.

Dessa forma, teremos uma chave primária com dois campos, mesmo repetindo o número da RG se tornará uma informação única devido ao UF.

Se temos dois números iguais como 234589 para as UFs GO e MG, então as respectivas chaves primárias serão distintas: 234589GO e 234589MG.

A figura a seguir ilustra detalhes sobre chave primária:

Chave primária é um campo que possui um conteúdo único em uma tabela.



Observe que a tabela **funcionario** possui o campo **matr** como uma chave primária e a tabela **secao** possui o campo **id** (de identificador) como uma chave primária sequencial.

02.06. Chave estrangeira

Se uma tabela B possui um campo (com o mesmo nome ou com outro nome) equivalente à chave primária de uma tabela A, então esse campo é denominado de **chave estrangeira**.

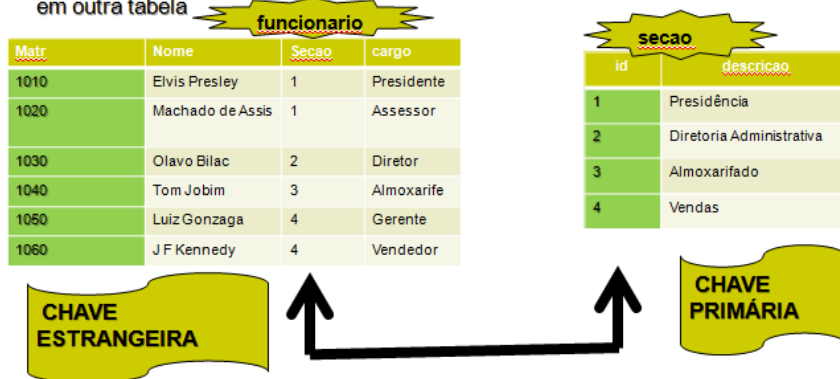
Portanto, uma chave estrangeira sempre existirá:

Quando houver um relacionamento entre duas tabelas A e B, onde a tabela B possui um campo equivalente (com mesmo nome ou não) da chave primária de A.

A figura a seguir ilustra a existência da chave uma estrangeira na tabela **funcionario**:



A chave estrangeira ocorre quando um campo de uma tabela for chave primária em outra tabela



A chave primária **id** da tabela **secao** é uma **chave estrangeira** (com outro nome, **Secao**) na tabela **funcionario**.

Exercícios:

01. Cite exemplos de, pelo menos, cinco tabelas com suas respectivas e prováveis chaves primárias.

02. Considere um banco de dados de um sistema de venda de produtos elétricos. Analise essa situação e liste os nomes das suas prováveis tabelas com suas chaves primárias e estrangeiras.