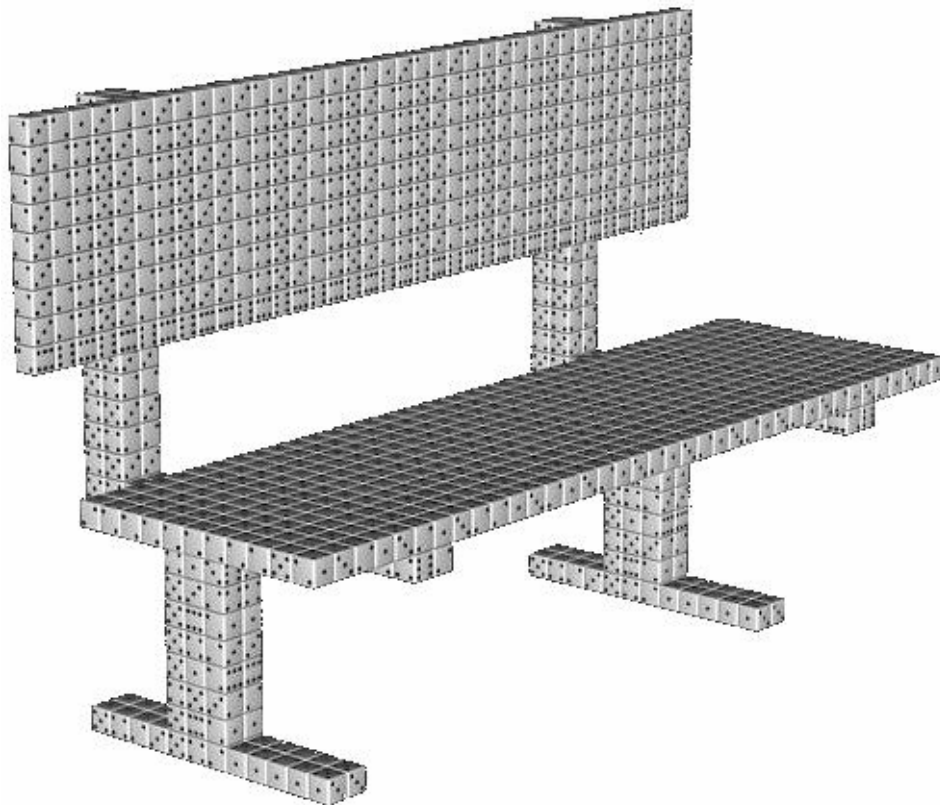


Banco de Dados



Índice

CONCEITOS DE BANCO DE DADOS	3
1 – O QUE É UM BANCO DE DADOS.....	3
2 – DIVISÕES DE UM BANCO DE DADOS	4
3 – TIPOS DE DADOS	5
4 – CHAVES PRIMÁRIAS E SECUNDÁRIAS	7
4.1 – Chave Primária	8
4.2 – Chave Secundária	9
5 – RELACIONAMENTOS ENTRE TABELAS	10
5.1 – Tipos de Relacionamentos.....	12
6 – SISTEMAS DE GERENCIAMENTO DE BANCO DE DADOS (SGDB)	14
7 – FUNÇÕES DE UM ADMINISTRADOR DE BANCO DE DADOS	14
DATA BASE UTILITY (DBU)	16
1 – O QUE É DATA BASE UTILITY (DBU)?	16
2 – APRESENTAÇÃO DO DBU	16

Esta apostila é uma compilação de várias apostilas e informações retiradas da internet, devendo ser utilizado apenas como suporte às aulas aplicadas em Sala no 1º módulo do Ensino Técnico em Informática.

Prof. Décio Malta

Prof. Décio Malta
E-mail: deciomalta@terra.com.br
MSN: décio_malta@hotmail.com
ICQ: 26385580

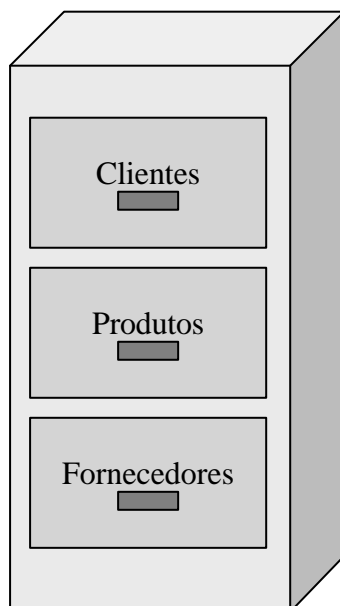
Conceitos de Banco de Dados

1 – O que é um Banco de Dados

A informática tem como finalidade automatizar tarefas, antes realizadas manualmente pelo ser humano, disponibilizando de forma rápida e eficiente às ferramentas necessárias para a execução de uma tarefa. Com o passar do tempo a informática vem se adequando aos novos hábitos e estilos, sendo de fundamental importância em nosso cotidiano.

Um exemplo de como a informática nos auxilia em nosso dia-a-dia, são os aplicativos utilizados para executar determinadas tarefas que antes eram executadas de forma manual e trabalhosa. Exemplo: Antigamente caso fosse necessário escrever um memorando seria necessário recorrermos a uma caneta e papel ou na melhor das hipóteses a uma máquina de escrever, hoje em dia torna-se difícil imaginar a execução de tal tarefa sem um editor de texto (ex. MS-Word, Writer).

Com o sistema de banco de dados não foi diferente, este surgiu da necessidade de obter informações de uma forma mais ágil do que a utilizada corriqueiramente, a melhor forma de compreender um banco de dados e compará-lo a um arquivo de documentos utilizado geralmente em empresas, vide próxima figura:



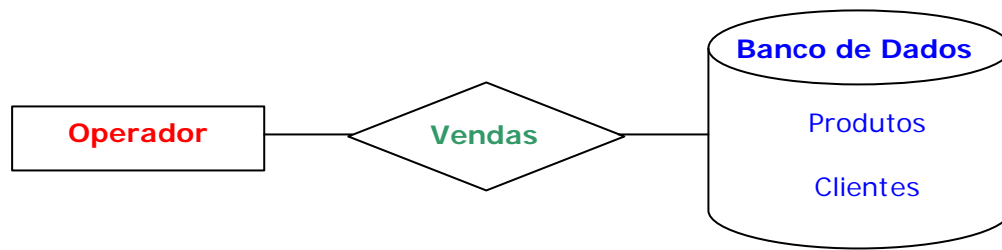
(Figura 1)

Na **Figura 1** vemos um arquivo comum utilizado em empresas para arquivamento de documentos e informações necessárias ao seu cotidiano, porém, estes arquivos eram de difícil manuseio, toda vez que fosse necessário utilizar alguma informação que constasse neste, era necessário abrir gavetas, pegar pastas, documentos etc., sendo que esta operação acabava por demorar muito tempo.

Uma ação que deveria ser efetuada de forma rápida acabava por demorar muito, como também, onerava a mão de obra que era exclusiva para execução desta atividade.

O banco de dados é basicamente um sistema de manutenção de informações pelo computador, ou seja, um sistema cujo objetivo global é manter as informações e torná-las disponíveis de forma rápida e eficiente ao operador. Trata-se de qualquer informação considerada significativa ao indivíduo ou à organização servida pelo sistema.

Abaixo vemos um exemplo simplificado de como um banco de dados pode vir a contribuir na organização de um departamento de Vendas:



A figura acima demonstra que o banco de dados é apenas uma parte de um processo, ou seja, neste exemplo temos um sistema de **Vendas** que é responsável por administrar às vendas de um estabelecimento qualquer, neste processo existem três figuras importantes:

- 1) **Operador** – É responsável por operar o aplicativo (usuário);
- 2) **Vendas** – Aplicativo responsável pelo processo, sendo que este interage entre operador e Banco de Dados.
- 3) **Banco de Dados** – Local onde ficam armazenadas as informações, neste caso específico armazena informações dos **Produtos e Clientes**, variando de acordo com o segmento e sistema a ser desenvolvido.

Os sistemas de banco de dados estão disponíveis em máquinas que de pequeno porte até os computadores de grande porte. Os recursos proporcionados por um determinado sistema são, até certo ponto, definidos pelo tamanho e pela potência da máquina utilizada. Os sistemas de grande porte, em particular, tendem a ter múltiplos usuários e os de pequeno porte a ter menor número de usuários, devido a sua limitação técnica.

2 – Divisões de um Banco de Dados

Podemos dividir um Banco de dados como se fosse um arquivo (Figura 1), pois, este nada mais é que uma forma organizada de armazenamento, porém de forma digital.

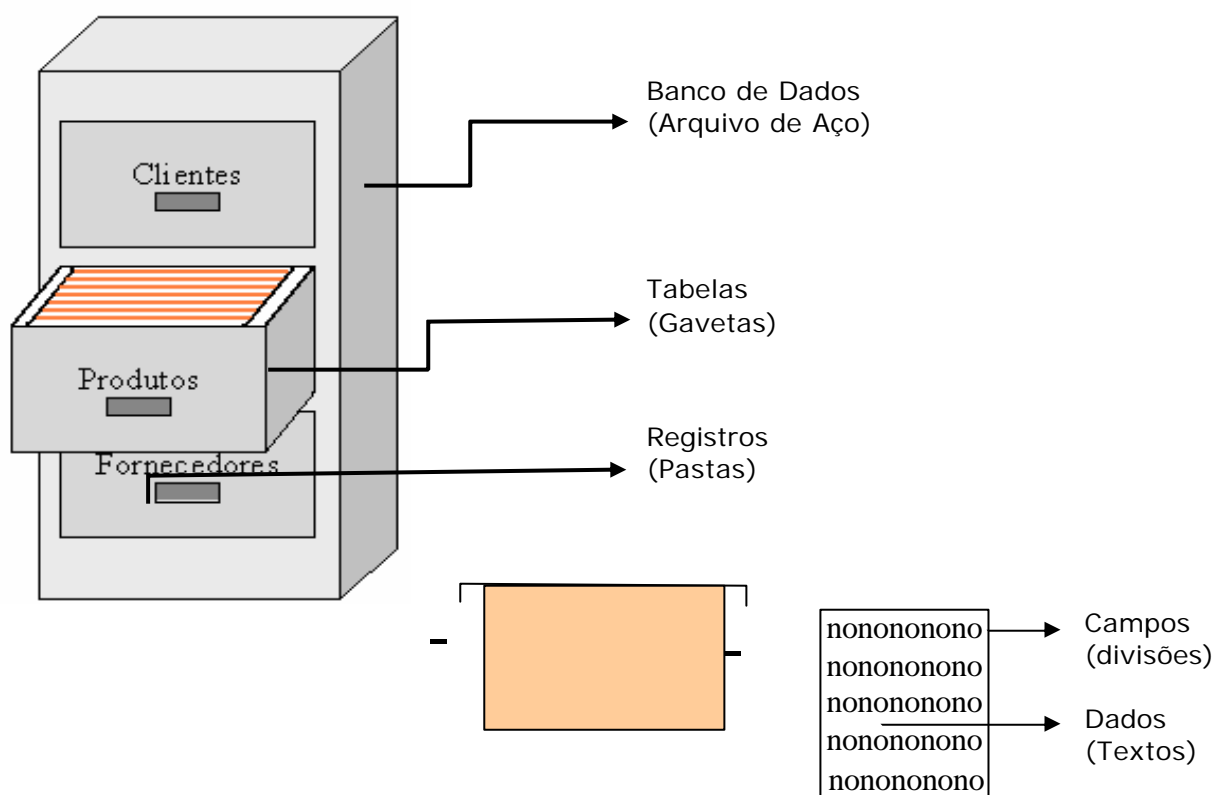
O Banco de dados é dividido em 5 partes:

- 1) Banco de Dados;
- 2) Tabelas;
- 3) Registros;
- 4) Campos;
- 5) Dados.

Para auxiliar a compreensão, vamos fazer um comparativo com o arquivo físico (Figura 1) para que sejam evidenciadas suas semelhanças.

Banco de Dados	?	Arquivo de Aço completo;
Tabelas	?	Gavetas do Arquivo;
Registros	?	Pastas Suspensas que estão dentro da Gaveta;
Campos	?	Informações que estão dentro das pastas;
Dados	?	Cada texto que compõe os Campos.

Vide próxima ilustração:



(Figura 2)

3 – Tipos de Dados

Ao ser criada a tabela, é necessário criar a estrutura da mesma, ou seja, definir quais os campos existirão, para executar tal tarefa faz-se necessário uma análise minuciosa dos detalhes para qual será utilizado o banco de dados, pois estes deverão ser os dados preenchidos pelo usuário ao utilizá-lo. Deve ser considerado nesta análise o tamanho que será utilizado para cada campo, para que no momento de criação seja discriminado o tamanho que cada campo irá ocupar.

Por exemplo, ao preencher um currículo são solicitadas diversas informações, sendo que estas informações são baseadas nas necessidades da empresa e quais detalhes desejam obter.

Ao se definir um campo é necessário informar qual o tipo de dado será digitado, existem basicamente cinco tipos de dados, conforme segue:

- a) Tipo Numérico – Neste campo são aceitos apenas números em seu preenchimento, sendo estes de zero a nove (0 a 9). Geralmente este tipo de dado **é utilizado apenas para campos que exijam cálculos**.

Exemplo:

Campo	Tamanho	Preenchimento
Código	04	1245
Salário	07	1.536,23
Numero de Filhos	02	05
Etc	99	99999999999

- b) Tipo Texto – Neste campo são aceitos todos os tipos de caracteres incluindo letras do alfabeto, números e caracteres especiais. Em algumas situações este campo também pode ser nomeado como Alfanumérico, Literal ou Character.

Exemplo:

Campo	Tamanho	Preenchimento
Nome	60	Pedro H. da Silva Xavier Neto
E-mail	60	zuzu_beleza@toligado.com.br
Rua	60	Senador José Filho Neto Jr.
Compl	20	Prédio 1º de Janeiro

- c) Tipo Data – Neste campo são aceitos apenas valores tipo data, podendo variar o formato da mesma * (DDMMAA, DDMMAAAA, MMDDAAAA etc.)

* → DD = Dia / MM = Mês / AA = Ano

Exemplo:

Campo	Tamanho	Preenchimento
Data de Nascimento	06	25/03/79
Data de Inclusão	06	300104
Data de Casamento	08	25-12-2002
Data de Visita	08	19072004

- d) Tipo Lógico – Neste campo são aceitas apenas duas situações “S” ou “N”, este tipo de campo é útil quando o preenchimento do campo deverá ser “Sim” ou “Não”. Em algumas situações este campo também pode ser nomeado como Verdadeiro / Falso, Sim / Não, Booleano etc.

Exemplo:

Campo	Tamanho	Preenchimento
Casado	01	S
Visitado	01	S
Filhos	01	N

- e) Tipo Memorando – Neste campo é aceito textos, ou seja, segue a mesma configuração do campo tipo texto, porém tem capacidade para um número maior de caracteres, sendo fixado em MegaBytes, podendo variar pelo tipo de banco de dados.

Exemplo:

Campo	Tamanho	Preenchimento
Situação	XX	Imóvel encontra-se vago, morador informou que o mesmo não vem sendo utilizado, pois se trata de um imóvel para aluguel, deverá ser efetuada uma supressão da ligação para que não sejam emitidas novas contas.

Os tipos descritos acima seguem uma linha dos que aparecem em grande parte dos Bancos de Dados, porém estes aspectos podem variar, em algumas situações pode haver mais ou menos tipos de dados, dependendo dos recursos oferecidos pelo Sistema Gerenciador de Banco de dados (ver capítulo referente à SGDB), como também pode variar a nomenclatura utilizada para denominar o seu tipo.

4 – Chaves Primárias e Secundárias

As chaves de um banco de dados são utilizadas para organizar as informações dentro do arquivo geral, ou seja, na figura anterior (Figura 2) existe uma forma de organização das informações, ou seja, o arquivo está organizado da seguinte forma:

Arquivo → Contém informações referentes a Produtos, Cliente e Fornecedores;

Gavetas → As gavetas estão divididas entre Produtos, Cliente e Fornecedores, ou seja, foi separada uma gaveta para cada item de forma a agilizar a procura;

Pastas → Cada pasta é referente a um Produto, Cliente ou Fornecedor variando da gaveta em que esteja guardada, sendo que estas pastas encontram-se organizada de forma alfabética crescente auxiliando assim a procura;

Dados → Os dados estão impressos nos documentos de forma a auxiliar a procura, por exemplo, o nome no referenciado no título do documento seguido das informações principais do Fornecedor ou Ciente, no caso dos produtos os mesmo estão organizados por nome do produto seguido de código, descrição, estoque etc..

As chaves poderiam ser comparadas ao índice de um livro, ou seja, é um método fácil e rápido de localizar uma informação em um volume de informações.

4.1 – Chave Primária

Chave primária é a forma principal de organização do Banco de Dados, aquela na qual o Banco de Dados utilizará como ordem no momento de exibir as informações. **Toda tabela deve possuir uma chave primária.**

Na figura anterior (Figura 2), por exemplo, a tabela (gaveta) de clientes está organizada em ordem alfabética pelo Código do cliente, sendo esta a sua chave primária.

Veja tabela de clientes abaixo organizada pela chave primária (Código).

Código	Nome	Estado
001	José Liveiras Bueno Fagundes	SP
002	Maria Joaquina da Silva	SP
003	Joaquim da Silva Manuel	MG
004	Antonio Roberto	RS
005	Felisberto Gonçalves	MS

Atenção:

Ao se definir a chave primária para uma tabela é necessário que se atendam alguns critérios, sendo estes de grande importância para garantir a integridade e eficiência do banco de dados, sendo assim, vamos apontar abaixo alguns itens referente à Chave Primária.

- Existência: Toda tabela deve possuir apenas um campo definido como chave primária;
- Redundância: O campo definido como chave primária não deve conter redundância em seu conteúdo, como também, não deve conter dado nulo ou vazio.

4.2 – Chave Secundária

Chave secundária é uma forma auxiliar de organizar as informações, o banco de dados não utilizará esta chave para visualizar as informações a princípio, para utilizá-la será necessário informar qual a chave desejada. **Para cada tabela podem haver varias chaves secundárias.**

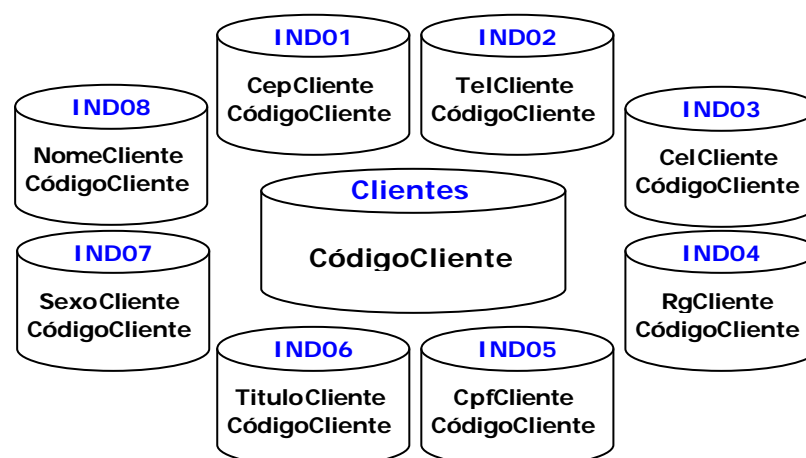
Na figura anterior (Figura 2), por exemplo, a tabela (gaveta) de clientes está organizada em ordem sequencial pelo “Código” do cliente, porém, pode ser interessante organizar esta tabela também pelo “Nome” do cliente para uma consulta, caso o usuário não lembre seu código etc., sendo assim, a organização pelo “Nome” seria uma chave secundária para esta tabela.

Veja tabela de clientes a seguir organizada pela chave secundária (Nome).

Código	Nome	Estado
004	Antonio Roberto	RS
005	Felisberto Gonçalves	MS
003	Joaquim da Silva Manuel	MG
001	José Liveiras Bueno Fagundes	SP
002	Maria Joaquina da Silva	SP

Atenção:

Ao se definir a chave secundária para uma tabela é necessário que seja avaliado criteriosamente se há necessidade de sua existência, pois, a criação de forma excessiva pode inviabilizar o sistema devido a dificuldade para atualização do banco de dados e seus índices, conforme pode ser visto no exemplo abaixo:



(Figura 3)

Para cada chave secundária definida, é criado um arquivo paralelo que tem por finalidade otimizar a consulta, sendo que, este arquivo é composto pelo campo definido e a chave primária.

No exemplo da figura 3, foram definidas 8 chaves secundárias, este exemplo demonstra quanto pode ser crítico à criação de chaves secundárias de forma desordenada, pois, para cada registro inserido na tabela **Clientes**, faz-se necessário inserir também um registro nas tabelas de Índices (IND01 à IND08), inviabilizando assim o armazenamento físico.

Como também, a cada vez que fosse Inserido / Alterado / Apagado um registro o Banco de Dados será necessário reordenar as chaves secundárias o que exigirá grande uso de recursos da máquina, podendo comprometer a performance.

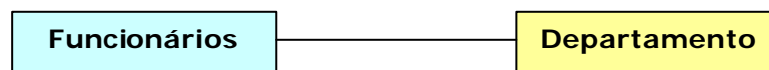
5 – Relacionamentos entre tabelas

Em algumas situações será necessário efetuar o relacionamento entre tabelas, este recurso é muito utilizado dentro dos bancos de dados relacionais, o surgimento desta tecnologia foi de grande utilidade, pois conseguir melhorar a qualidade dos sistemas de informação.

Um relacionamento deve ser criado com objetivo de atender algumas situações referente ao banco de dados, são elas:

- Reduzir espaço de armazenamento em disco;
- Garantir integridade da informação;
- Otimizar o processo de cadastro / consulta;
- Melhorar a performance no processamento das informações.

Veja a seguir um exemplo de relacionamento:



(Figura 4)

O relacionamento demonstrado na figura 4 indica que a tabela de "Funcionários" se relaciona com a tabela de "Departamentos", ou seja, cada funcionário está vinculado a um departamento.

Vejamos um exemplo a seguir que demonstra a economia de espaço com o uso de relacionamento entre tabelas e o uso sem relacionamentos:

Exemplo 1:

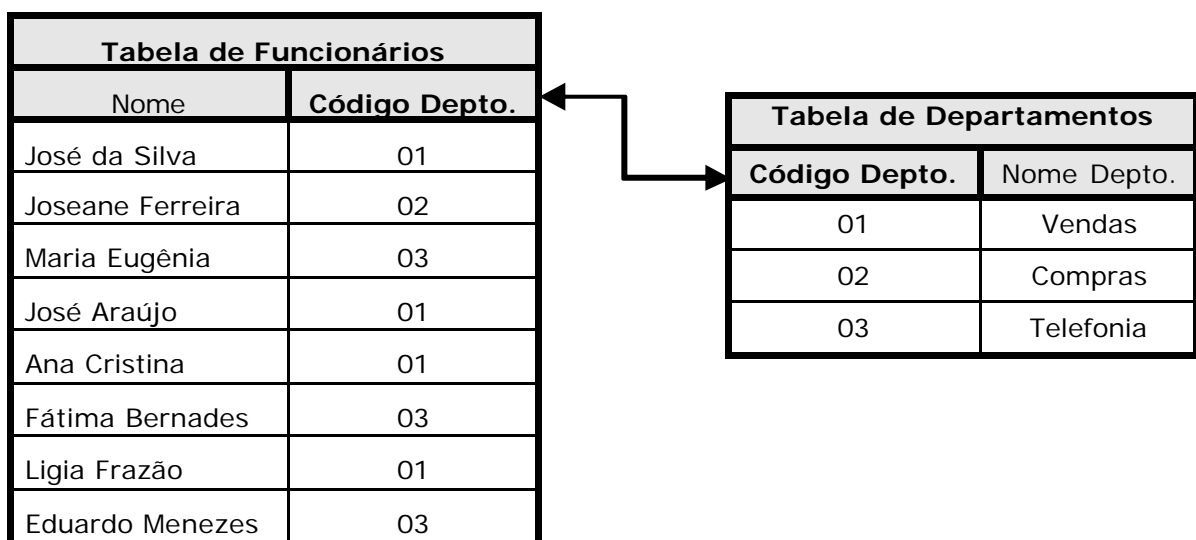
Banco de dados de Funcionários onde Nome e Departamento são campos da mesma tabela, sendo que nesta situação não existem relacionamentos.

Tabela de Funcionários	
Nome	Departamento
José da Silva	Vendas
Joseane Ferreira	Compras
Maria Eugênia	Telefonia
José Araújo	Vendas
Ana Cristina	Vendas
Fátima Bernades	Telefonia
Ligia Frazão	Vendas
Eduardo Menezes	Almoxarife

Neste exemplo nota-se que no campo "Departamento" optou-se em descrever o nome do departamento completo.

Exemplo 2:

Banco de dados de Funcionários onde existem duas Tabelas (Funcionários e Departamentos) onde Nome e Código de Depto. são campos da tabela de Funcionários e Código Depto e Nome Depto são campos da tabela de Departamentos, para esta situação existe relacionamento entre as duas tabelas, conforme pode ser verificado a seguir.



Neste exemplo optou-se em separar o Departamento da tabela "Funcionários", esta atitude beneficia em diversos aspectos a modelagem de dados:

- Considerando que a tabela de Funcionários tende a crescer substancialmente, a opção dividir em duas tabelas relacionando-as através de um "Código de Departamento" é altamente recomendável, pois este código ocuparia menos espaço que a descrição do departamento.

- b) Se em algum momento for necessário alterar o nome de um departamento será menos complexo, tendo que alterá-lo em apenas um registro da tabela "Departamentos".

Ex.: Alterar "Vendas" para "Vendas e Licitação".

- c) O usuário do sistema não terá que cadastrar o departamento sempre que for cadastrar um novo funcionário, mas sim, vincular o código do respectivo departamento.
- d) Não haverá o risco de digitação incorreta de informação no nome do departamento, garantindo assim uniformidade da informação.
- e) Devido a qualidade da informação trabalhando de forma relacional, o sistema ficará mais eficaz e com melhor performance.

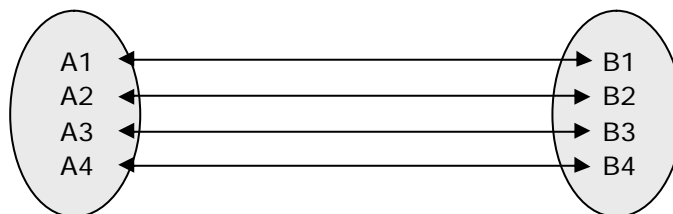
5.1 – Tipos de Relacionamentos

Como já visto anteriormente, os relacionamentos entre tabelas tendem a reduzir os espaços utilizados pelo Banco de Dados como também otimizar possíveis manutenções futuras.

Relacionaremos a seguir os tipos de relacionamentos disponíveis entre tabelas:

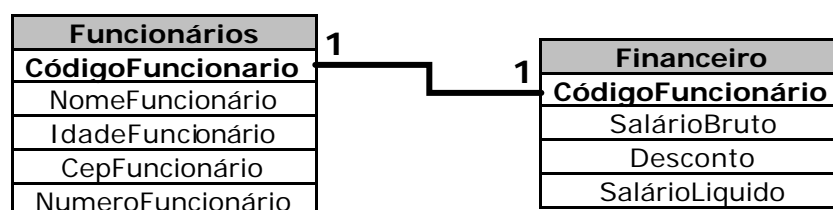
a) **Relacionamento 1:1**

No relacionamento 1:1 as coincidências entre as tabelas serão de um para um, ou seja, para cada registro em uma tabela poderá haver apenas um em outra.



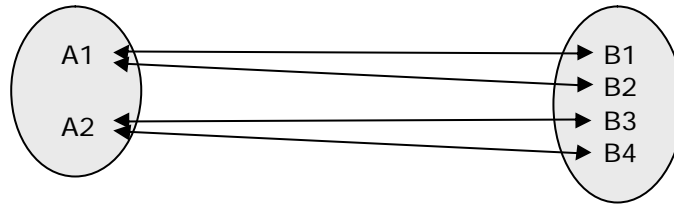
Segue exemplo de relacionamento 1:1

Exemplo: Existe uma tabela com os dados cadastrais do funcionário e outra tabela onde são armazenados os dados financeiros de cada funcionário.



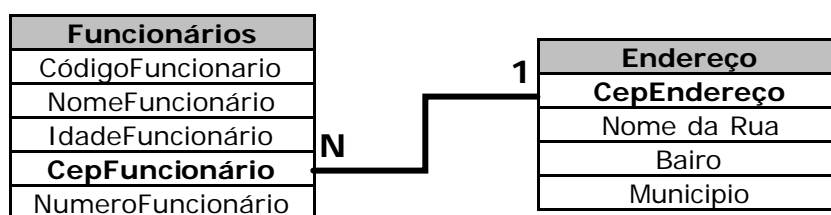
b) **Relacionamento 1:N ou N:1**

No relacionamento 1:N ou N:1 as coincidências entre as tabelas serão de um para muitos, ou seja, para cada registro em uma tabela poderão haver vários registros em outra.

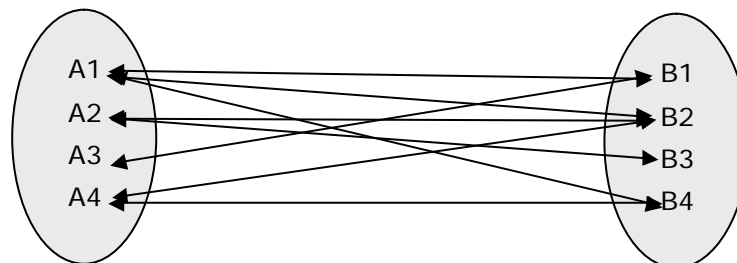


Segue exemplo de relacionamento N:1

Exemplo: Existem vários funcionários que moram na mesma rua.

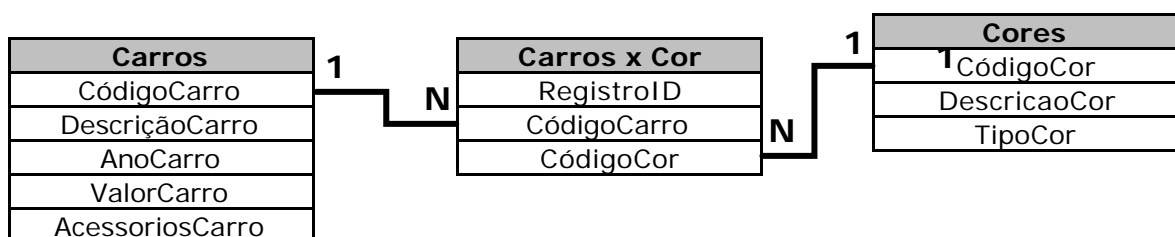
c) **Relacionamento N:N**

No relacionamento N:N as coincidências entre as tabelas serão de muitos para muitos, ou seja, para cada registro em uma tabela poderão haver vários registros em outra e vice-versa.



Segue exemplo de relacionamento N:N:

Exemplo: O Modelo de Carro Corsa pode ser fabricado em várias Cores, a Cor azul existe para Vários Modelos de Carros.



Quando tratamos de um relacionamentos N:N faz-se necessários utilizar uma tabela auxiliar para efetuar este relacionamento, para que assim os dados estejam íntegros.

6 – Sistemas de Gerenciamento de Banco de Dados (SGDB)

Um Sistema de Gerenciamento de Banco de Dados (SGBD) é uma coleção de programas que permitem ao usuário definir, construir e manipular Bases de Dados para as mais diversas finalidades.

Principais características de um SGBD:

- a) Auto-Contenção: Armazenar completamente toda a descrição dos dados, seus relacionamentos e formas de acesso.
- b) Independência dos Dados: Estar imune a mudanças na estrutura de armazenamento ou na estratégia de acesso aos dados. Sendo que os relacionamentos poderão ser feitos no próprio SGBD sem a necessidade de uma linguagem de programação para efetuar esta interligação.
- c) Abstração dos Dados: Apresentar ao usuário somente uma representação conceitual dos dados, o que não inclui maiores detalhes sobre sua forma de armazenamento real. Neste modelo, um esquema de tabelas, seus relacionamentos e suas chaves de acesso são exibidos ao usuário, porém nada é demonstrado sobre a forma interna que o SGBD administra estas informações, como também o acesso dos usuários aos registros.
- d) Visões: Permitir que cada usuário visualize os dados da forma diferente da existente previamente no Banco de Dados. Uma visão consiste de um subconjunto de dados do Banco de Dados, necessariamente derivados dos existentes no Banco de Dados.

7 – Funções de um Administrador de Banco de Dados

Um Administrador de banco de dados também é conhecido como DBA (Data Base Administrator), este profissional é responsável por toda e qualquer alteração ou criação de bancos de dados, segue atributos de um DBA:

- a) Definir o esquema conceitual: Cabe ao DBA decidir quais as informações devem ser mantidas no Banco de Dados, identificando os detalhes de interesse para a empresa.
- b) Definir o esquema interno: O DBA deve decidir como serão representados os dados no banco de dados armazenado. Esta etapa é também conhecida como Projeto Físico.

- c) Ligação com usuários: É atribuição do DBA fazer a ligação com os usuários, a fim de garantir que os dados que eles necessitam estarão disponíveis da melhor forma, como também, fornecer instruções técnicas e assistência para determinação e resolução de problemas no Banco de Dados.
- d) Definir Normas de descarga e recarga: Uma vez que a empresa esteja comprometida com o sistema de Banco de dados, ela se torna dependente de modo crítico do sucesso deste sistema. Em caso de danos a qualquer parte do banco de dados, provocados por erro humano ou por falha de hardware ou ainda do sistema operacional, é essencial que o DBA seja capaz de reparar os dados em questão com o mínimo de demora e com o menor efeito possível sobre o restante do sistema, em outras palavras o DBA é também responsável pelas rotinas de BACKUP e RESTORE do banco de dados.
- e) Monitorar o desempenho e responder a requisitos de mudança: o DBA deve acompanhar a performance do banco de dados, no que diz respeito a tempo de resposta e integridade dos dados e no caso de uma baixa performance efetuar as modificações cabíveis.

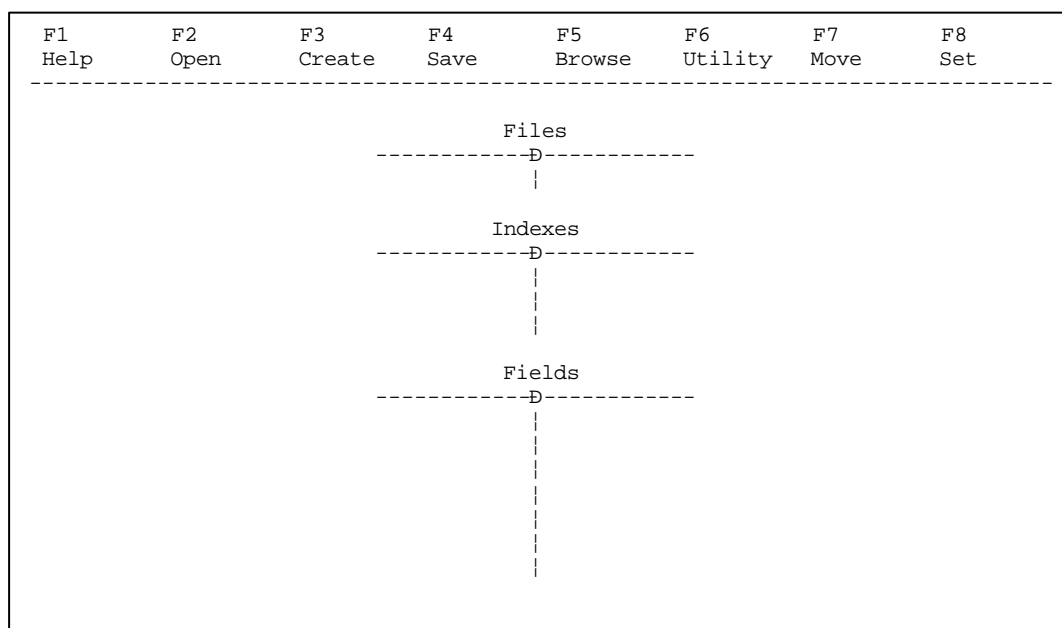
Data Base Utility (DBU)

1 – O que é Data Base Utility (DBU)?

O Aplicativo DBU é um sistema desenvolvido em Clipper que tem por finalidade manipular e administrar bancos de dados utilizados pelo Clipper (".dbf"), porém não podemos afirmar que o DBU seja um Sistema Gerenciador de Banco de Dados (SGBD), pois o mesmo não cria e/ou administra os relacionamento entre tabelas, visões etc..

2 – Apresentação do DBU

Ao entrar no DBU você verá a tela a seguir:



(Figura 5)

Na parte superior da figura anterior (Figura 5) vemos o menu de opções do DBU que pode ser acionado através das teclas "F1" até "F8" segue abaixo suas funcionalidades:

- F1 (Help) – Ajuda do sistema, utilizar quando houver dúvidas quanto a correta função do DBU, pode ser acionada a qualquer momento pelo usuário, sendo que assim que acionado dará suporte ao campo posicionado no momento.

- b) F2 (Open) – Utilizado para abrir uma tabela, índice ou visão previamente criado, segue as opções deste menu:
- File – Abre uma tabela (".dbf").
 - Index – Abre arquivo de índice gerado a partir de uma tabela (".dbf");
 - View – Abre uma visão criada anteriormente.
- c) F3 (Create) – Utilizado para criar uma nova tabela ou índice, segue as opções deste menu:
- Database – Utilizada para criar uma nova tabela, ao ser selecionado será solicitada à relação de campos irão compor a estrutura da tabela:

Deverão ser preenchidos os seguintes campos da estrutura:
- *Field Name*: Nome do campo
 - *Type*: Tipo do Campo, para alterar o tipo de campo deve ser informado a primeira letra do tipo desejado ("N", "D", "C", "M").
 - "C" – Character (texto);
 - "N" – Numérico;
 - "D" – Data;
 - "L" – Lógico (Verdadeiro ou Falso);
 - "M" – Memo.
 - *Width*: Tamanho do campo, no caso de campos tipo Data, Logical e Memo o tamanho já consta como preenchido.
 - *Dec*: Quantidade de Casas decimais do campo, aplicável apenas para campo tipo Numérico.
- Index – Cria um arquivo de índice baseado na tabela atual, para criar um índice é necessário previamente estar com uma tabela aberta, Serão solicitadas as seguintes informações para criação do índice:
 - *File*: Deve ser informado o nome do arquivo índice
 - *Key*: Deve ser informado por qual campo da tabela deve ser criado um índice.
- d) F4 (Save) – Utilizado para Salvar a estrutura de uma tabela criada ou uma Visão.
- View – Salva a visão atual da base de dados, é solicitado o nome da view.

- ii. Structure – Salva a estrutura da tabela, é necessário informar o nome da tabela ao efetuar esta ação.
- e) F5 (Browse) – Utilizado para inserir dados na tabela, faz-se necessário estar com a tabela aberta.
 - i. Database – Para inserir dados na tabela baseado na forma que foi definido na estrutura.
 - ii. View - Para inserir dados na tabela baseado na forma que foi definido na view.
- f) F6 (Utility) – Utilizado para utilizar funções para a tabela.
 - i. Copy – Copiar tabela aberta para uma nova.
 - ii. Append – Importar dados de outra tabela ou arquivo texto.
 - iii. Replace – Altera conteúdo dos campos.
 - iv. Pack – Limpa fisicamente registros deletados.
 - v. Zap – Deleta todos registros da tabela.
 - vi. Run – Executa comandos do MS-DOS.
- g) F7 (Move) – Utilizado para utilizar funções para a tabela.
 - i. Seek – Procura por uma um registro.
 - ii. Goto – Vai para um registro na tabela.
 - iii. Locate – Localiza um registro na tabela.
 - iv. Skip – Pula quantidade definida de registros
 - vii. Copy – Copiar tabela aberta para uma nova.
 - viii. Append – Importar dados de outra tabela ou arquivo texto.
 - ix. Replace – Altera conteúdo dos campos.
 - x. Pack – Limpa fisicamente registros deletados.
 - xi. Zap – Deleta todos registros da tabela.
 - xii. Run – Executa comandos do MS-DOS.
- h) Sair
 - i. Para sair basta teclar <ESC>