

# BANCO DE DADOS

## INTRODUÇÃO

Banco de dados é um conjunto de informações relacionadas entre si, sobre um determinado assunto ou entidade que é armazenado por meio magnético.

A principal finalidade da criação dos bancos é o armazenamento organizado das informações de uma organização, visando a otimização dos sistemas facilitando entrada de dados, alterações, consultas, formulários, e o uso adequado destas informações para que a organização tenha em banco de dados adequado.

No decorrer do processo de aperfeiçoamento foi surgindo a necessidade de novos tipos de bancos para suprimento das necessidades das empresas e maior eficácia no processo de armazenamento de informações. Foram surgindo assim modelos de banco de dados como: corporativo, distribuído, rede, hierárquico, relacional e datawarehouse.

Cada um desses modelos de banco de dados que veremos a seguir possuem suas finalidades e características próprias. Os modelos de banco de dados foram criados para que houvesse maior interação das organizações com os sistemas informatizados, trazendo mais segurança, rapidez, confiabilidade nas informações obtidas, e maior qualidade nos projetos desenvolvidos.

## **MODELO DE BANCO DE DADOS CORPORATIVO**

---

Definimos como banco de dados corporativo o armazenamento de todos os arquivos e informações de uma empresa agrupados de uma forma que seja possível consulta-los com uma maior exatidão e rapidez de qualquer parte da empresa.

O banco de dados corporativos não é uma mudança nas características fundamentais de um banco de dados, mas sim a disponibilidade de novas tecnologias para o seu acesso e armazenamento, ele vai além das fronteiras de um banco de dados, aonde é aperfeiçoado o acesso aos dados através de uma melhoria nas tecnologias de processamento com componentes mais rápidos, terminais remotos, além da aceitação cada vez maior de operações compartilhadas, ou seja, os dados podem ser gerenciados como um recurso.

Esses dados podem ser considerados recursos porque uma mesma informação ou dado possuem uma duração e sua importância e pode ser requisitado ou necessário para várias partes da empresa, com isso se torna importante gerencia-los, sendo o único recurso que mantém a pista dos demais.

O banco de dados corporativo é gerenciado por um departamento cuja atribuições, em geral tem a responsabilidade pelo banco de dados que inclui segurança, projeto, desempenho e publicidade. Além de proporcionar muito mais que arquivos simples, comuns, do passado. Organizado de modo adequado ele pode oferecer :

- 📖 Acesso rápido e aleatório aos dados;
- 📖 Acesso dos executivos à gama inteira de informações sobre a empresa;
- 📖 Referências cruzada e compatibilização de informações;
- 📖 Funções interdepartamentais automatizadas;
- 📖 Prognósticos e estimativas de projeções, baseados na história e em fatos esperados.

Além de proporcionar a uma empresa vantagens como:

- 📖 Redução de custo;
- 📖 Boa vontade do cliente;
- 📖 Menos erros;
- 📖 Maior vigor no controle executivo;
- 📖 Melhor planejamento.

## **BANCO DE DADOS CORPORATIVO COMO FERRAMENTA**

### **Lotus Notes**

#### **Conceitos Básicos**

Todas as aplicações da ferramenta Notes são desenvolvidas através de um banco de dados do tipo corporativo, no Notes, um banco de dados é qualquer estrutura que contenha informações, que pode se localizar em seu micro particular ou em qualquer outro ponto da rede ou estando amplamente disponível para todos que estejam usando o servidor. Quando criado um novo banco de dados podemos ajustá-lo e disponibilizá-lo para utilizar qualquer tipo de informações nele contida.

No Notes podemos consultar e visualizar documentos ou informações com um formato fácil e próprio, além de ajustar e personalizar essas visualizações fazendo com que ele utilize todas as informações desejadas. Facilitar a manutenção e atualização dos arquivos nele existente, listando todos os documentos dando a alternativa de atualizar, incluir ou remover qualquer tipo de informação.

O Notes armazena informações em documentos e esses documentos são formados por campos específicos em um formulário, sendo este uma tela personalizável e a base de todo o documento, alguns destes formulários - como em qualquer outro banco de dados - possui campos textos, datas, números e itens em formato RTF (gráficos e outros), que necessitam palavras específicas ou nomes de usuários. Esses podem conter quaisquer combinações de gráficos, textos e informações digitadas pelo usuário.

### **QUE SE PODE FAZER COM LOTUS NOTES**

Ele é um produto muito flexível pode suportar plataformas múltiplas, representar um verdadeiro sistema de gerenciamento de documentos.

Podemos utilizar o Notes para reduzir o manuseio de papéis no escritório, automatizar projetos de fluxos de trabalho, acompanhar e discutir assuntos. Além de incluir sistemas de controle de estoque, arquivos de bibliotecas e aplicações gerenciais

### **CARACTERÍSTICAS**

#### **Multiplataforma**

Notes pode ser executado em diversas plataformas, como exemplo, em microcomputadores Macintosh, ambiente Windows e sistemas operacionais UNIS e OS/2 ao mesmo tempo. Quando é acessado as informações o Notes tem a necessidade de ter uma máquina servidora para manter e

organizar as informações, acompanhar a segurança e o privilégios de acesso, compartilhar informações entre todos os usuários e encaminhar documentos, arquivos e informações.

### Bancos de Dados

Usando o Notes podemos armazenar informações, tópicos de discussões, arquivos e mensagens em formato típico de banco de dados. O Notes também permite importatar vários tipos de dados e criar visualizações totalmente personalizadas para resumir e relatar todos os documentos no seu banco de dados, podendo assim importar qualquer tipo de planilha para o banco de dados além de ter a possibilidade de enviá-la pelo correio eletrônico para qualquer colega.

Podemos também disponibilizar de informações alocadas em diversos servidores espalhados pela empresa, ou seja se eu tenho um dado em minha máquina posso disponibilizar para toda a rede e toda e qualquer mudança que eu ou outra pessoa venha a fazer será atualizado em todos os pontos da rede.

Os bancos de dados podem ser indexados, o que possibilita uma fácil consulta e dá a alternativa de pesquisar itens específicos através de grandes quantidades de dados de maneira rápida e de formato fácil. Você tem a alternativa de deixar o Notes reindexar automaticamente os documentos sempre que novas informações sejam registradas mantendo atualizados.

### NotesMail

Ele possui um completo software de correio eletrônico embutido em seu sistema possibilitando ao seus usuários a emissão e o recebimentos de mensagens, arquivos, com características de multimídias, vídeo e de áudio, e normais para qualquer outro usuário do NotesMail. Além de possibilitar o encaminhamento de documentos e programas automaticamente, encontrando o caminho independente do tamanho ou porte da sua rede, tendo assim uma maior eficiência e exatidão no encaminhamento das informações.

### Segurança

Seu sistema de segurança rígido é altamente integrado que limita o acesso de um número de pessoas que possam utilizar a rede ou determinados bancos de dados nele existente ou até mesmo parte dele. Com isso ele dá uma maior proteção aos seus dados evitando que qualquer pessoa não autorizada acesse seus dados sem autorização.

## **CRIAÇÃO DE BANCO DE DADOS USANDO LOTUS NOTES**

O Lotus Notes é constituído principalmente de diversos bancos de dados. Se você usar um banco de dados de caixa de correio ou de um sistema de desenvolvimento de projeto, todo banco de dados é dependente de estrutura consistente de bancos de dados do Notes. Certos comandos funcionam em todos os bancos de dados do Notes, podendo assim abrir, fechar, remover e acrescentar todos eles

na sua área de trabalho da mesma maneira. Essa consistência torna fácil compartilhar bancos e aplicações do Notes entre grupos e departamentos porque todos podem acessá-los da mesma maneira. No entanto, cada banco de dados tem também suas próprias características.

É característico de cada banco de dados possuir seu próprio conjunto de documentos, que você pode compor, e visualizações para ver essas informações. Por exemplo, um banco de dados de endereços pode ter diferentes documentos para empresa, órgãos governamentais e pessoas. Esse banco de dados pode ter uma configuração de visualização distinta para cada tipo de documento e uma visualização combinada que lista todos os endereços independentemente de tipo. Outro exemplo poderia ser um banco de dados de gerenciamento de projeto.

A remoção de banco de dados é fácil, pois a combinação de teclas para executar essa função é simples.

Para se copiar bancos de dados é importante que se faça uma cópia reserva, por segurança e com o objetivo de facilitar o uso, porque assim se por um acaso acontecer algo de errado no seu trabalho, já tem-se uma cópia de segurança para evitar transtornos.

Todas as aplicações Notes consistem em um ou mais bancos de dados com funções específicas. Esses bancos de dados são desenvolvidos, um de cada vez e com muita cautela. O desenvolvimento de novos bancos de dados permite melhor adequação a interface Notes e seus diversos recursos. A criação de um banco de dados faz com que se veja o Notes sob a perspectiva do projeto.

Uma etapa necessária na criação de um banco de dados é o desenvolvimento de formulários. Um *formulário* é o que você usa para compor documentos para preencher o banco de dados. Quando você cria o formulário, o Notes lhe dá uma tela totalmente em branco para que você a personalize. Você poderá adicionar todos os tipos de campos para armazenar nesse formulário. Você poderá adicionar todos os tipos de campos para armazenar diferentes tipos de informações. Os campos podem conter textos, datas, desenhos, nomes, tabelas, e gráficos.

Para criar uma visualização, você adiciona colunas que podem apresentar quase todos os tipos de campos de formulário. Após gravar sua visualização, você terá um banco de dados em condições de trabalho.

O Notes tem uma linguagem de macros que lhe permite tirar proveito dos seus muitos recursos e funções nos formulários e visualizações do banco de dados. O uso da linguagem de macros é normalmente um trabalho mais avançado de projeto, mas você deverá estar ciente de suas possibilidades.

O último passo para criação do banco de dados é a verificação dos níveis de segurança do usuário. Pode-se incluir usuários, um de cada vez para dar certos tipos de acesso e determinados usuários, ou poderá estabelecer um nível de segurança padrão para todos os usuários ao acessarem aquele banco de dados. A segurança do Notes é outro tópico complexo que somente começa com o banco de dados.

## **MODELO DE BANCO DE DADOS DISTRIBUÍDO**

---

A ligação de computadores situados em locais diferentes deu origem à disciplina de redes de computadores . Uma rede é representada como um grafo em que os computadores estão nos modos e as arestas denotam as ligações entre eles . A rede é dita homogênea se os computadores são do mesmo tipo e heterogênea no caso contrário . Protocolos ( *Regras que disciplinam a interconexão dos computadores* ) adequados devem ser desenvolvidos .

As redes se tornam economicamente mais conveniente à medida que cai o custo das máquinas , embora o custo e as dificuldades de transmissão possam ser fatores limitares . Pode ser mais vantajoso para a empresa Ter uma rede de minicomputadores , localizando cada um junto a um grupo de departamento ou junto a cada filial , disso, várias empresas podem ligar suas máquinas em rede e compartilhar o uso da rede , obtendo uma maior capacidade de processamento.

Isso representa a possibilidade de atingir um grau maior de descentralização das atividades . No caso de banco de dados , é comum que certas informações:

📖 sejam necessária apenas a um departamento.

📖 devam ser rapidamente disponíveis a esse departamento.

E assim seria razoável que essas informações fossem guardadas junto ao departamento . Da descentralização da guarda da informação decorre a noção de banco de dados distribuídos . Com bancos de dados distribuídos , os arquivos ou partições de arquivos são localizados de modo a otimizar sua disponibilidade .

Um banco de dados distribuído é um conjunto de banco de dados armazenados em diferentes nodos de uma rede de computadores , sendo esses banco de dados correlacionados logicamente , seja por relações funcionais , seja porque são no todo ou em parte cópias múltiplas das mesmas informações , de modo a constituir em qualquer caso uma única coleção de dados .

O critério de localização dos dados é influenciado pela estrutura de controle adotada na rede . Na estrutura hierárquica ( *sistema hierarquicamente distribuído* ) , os computadores executam tarefas que interagem de modo mais ou menos estruturado e controlado pelos membros de mais alto nível na hierarquia . Na estrutura simétrica ( *sistema simetricamente distribuído* ) , todos os computadores cooperam a um mesmo nível lógico , embora relações patrão – servo possa ser criadas dinamicamente na execução de alguma tarefa.

Com uma estrutura simétrica , a tendência é colocar junto a cada nodo os dados mais utilizados no modo . Para dados utilizados com frequência por dois ou mais nodos , pode –se criar cópias residentes em cada um desses nodos .

É um problema difícil manter a consistência entre as cópias quando ocorre uma atualização no nodo . Para dados utilizados como frequência por dois ou mais nodos , pode –se criar cópias residentes em cada um desses nodos .

Com uma estrutura simétrica , a tendência é colocar junto a cada nodo os dados mais utilizados no nodo. Para dados utilizados com frequência por dois ou mais nodos , pode –se criar cópias residentes em cada um desses nodos .

E um problema difícil manter a consistência entre as cópias quando ocorre uma atualização em um nodo contendo uma das cópias . A atualização deve atuar corretamente , independentemente da velocidade de transmissão dos dados ; o nodo que vai empreender a atualização deve notificar a todos os demais que contenham cópias sua intenção de atualizar o dado , e, se houver algum conflito , este deve ser resolvido nesse momento ( *antes portanto das fases de bloqueio e da atualização propriamente dita* ) ; com isso se procura inclusive evitar impasses .

Uma utilidade especial das cópias surge na seguinte situação em que um nodo **A** precisa de um dado que existe em um nodo **B** . Se o computador em **B** não estiver ativo no momento , **A** pode recorrer a algum outro nodo **C** que possua cópia do mesmo dado.

De um modo geral , deve haver para os programas um tipo adicional de independência quanto aos dados : o programa não deve precisar especificar o local em que estão os dados , dirigindo – se ao sistema como um todo , e deixando que a localização dos dados seja feita automaticamente . Assim também a escolha da cópia em **B** ou **C** seria automática.

Para a execução de uma transição envolvendo dados em diferentes nodos há dois métodos básicos :

Os dados requeridos pelo modo solicitante são localizados e enviados a ele como parte da execução da transação . Pode ocorrer que os dados estejam armazenados em formas diferentes em cada nodo e nesse caso deve ser feita sua conversão antes de serem transmitido .

A transmissão é decomposta em subtransações , cada uma das quais será executada em um só nodo. O envio de dados de um nodo a outro ocorrerá apenas quando se tiver de confrontar dados de nodos diversos e quanto aos dados que sejam resposta a uma consulta .

Os diretórios de dados tem uma importância vital em qualquer modalidade de bancos de dados distribuídos . Deverão incluir a localização dos arquivos , com o que inclusive se poderá garantir a independência dos programas quanto a localização sendo está feita pelo sistema por consulta ao diretório de dados.

O diretório de dados pode ser mantidos em um único nodo central ou então em cada nodo , criando –se neste caso o problema de manter atualizadas as cópias do diretório de dados . Uma solução interessante é manter em cada nodo um diretório de dados com todas as informações sobre os dados locais ( *e talvez dos não locais de uso freqüente* ) e sobre os demais dados apenas a informação de onde se encontram *denominado-se está parte do diretório de dados catálogo de localização – switching catalog* ) .

A área de bancos de dados distribuídos ainda necessita de muita pesquisa em virtude de sua novidade e complexidade . Problemas como os de manter integridade , consistência e segurança se

agravam ; a recuperação após um erro se complica se atinge diversos nodos . Alem disso surgem problemas novos como os envolvidos na execução distribuída de transações , conversão de dados ,

sincronização de eventos nos nodos , e escolha de estratégias “globais “ que otimizam a manipulação levando em conta maquinas e sistemas heterogêneos.

## VANTAGENS DA DISTRIBUIÇÃO DE DADOS

A principal vantagem de um sistema de banco de dados distribuído é compartilhar e acessar dados de uma maneira segura e eficiente .

## DESVANTAGENS DA DISTRIBUIÇÃO DE DADOS

A principal desvantagem do sistema de banco de dados distribuídos é o acréscimo de complexidade exigida para assegurar coordenação própria entre os locais . Este aumento de complexidade toma a forma de :

### **Custo de desenvolvimento de software**

É mais difícil implementar um sistema de banco de dados distribuído e, conseqüentemente , muito mais caro.

### **Maior Potencial para erro**

Uma vez que os locais que compõem o sistema distribuído operam em paralelo , é difícil manter a precisão de algoritmos . O potencial existe para erros extremamente sutis;

### **Aumento de overhead de processamento**

A troca de mensagens e a computação adicional exigida para se conseguir coordenação interlocal é uma forma de overhead que não aparece nos sistemas centralizados .

**Obs:** Ao escolher um projeto para um sistema do banco de dados , o projetista deve analisar as vantagens e desvantagens de distribuição de dados .



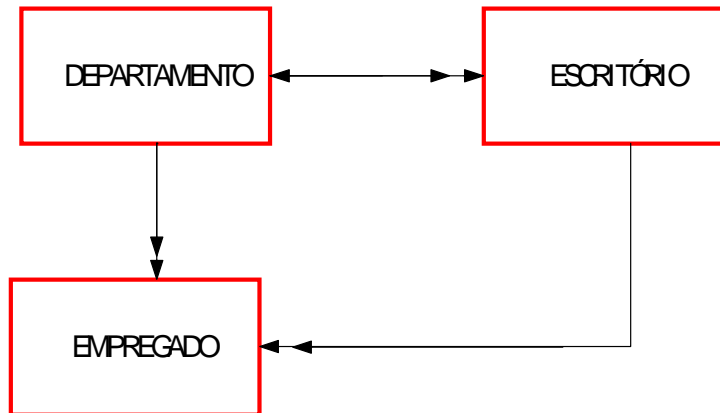
## MODELO DE BANCO DE DADOS REDE

Vários sistemas de gerenciamento de banco de dados se baseiam num modelo de rede comumente conhecido como modelo “CODASYL”.

CODASYL significa: Conferência Sobre Linguagem de Sistemas de Dados (Conference on Data Systems Languages).

Uma rede é, essencialmente um conjunto ilimitado de nós (tipos de registros neste caso) e de ramais de ligação, ou bordas. Na verdade, uma hierarquia é apenas um tipo particular de rede. Uma rede não apresenta o conceito de nó “raiz” e os registros podem ter diversos tipos de registros - pais, assim como diversos tipos de registros - filhos.

Abaixo temos um modelo de um banco de dados em rede mostrando detalhadamente.



Observe que utiliza-se uma seta simples para representar uma associação unívoca e logicamente utiliza-se uma seta dupla para representar uma associação múltipla.

Verifica-se que na figura acima um departamento possui diversos empregados e diversos escritórios e que, em cada escritório, há vários empregados lotados. Veja, não há sentido de “raiz” embora se possa argumentar que o registro EMPREGADO tem dois “pais”.

Um modelo de rede para um sistema de gerenciamento de dados compartilham funcionalmente de muitas semelhanças com o modelo hierárquico. As ocorrências de registros são interligados por ponteiros; Os ramais apresentam relacionamentos de um-para-muitos, os registros especiais chamados de registros de “ligação”, são usados com relacionamento um-para-muitos, para se implantar relacionamentos de muitos-para-muitos e existe um sentido de estrutura física versus estrutura lógica (chamados “esquemas” e “subesquemas”). Os registros na rede são “interligados” do mesmo modo que o são os registros hierárquicos.

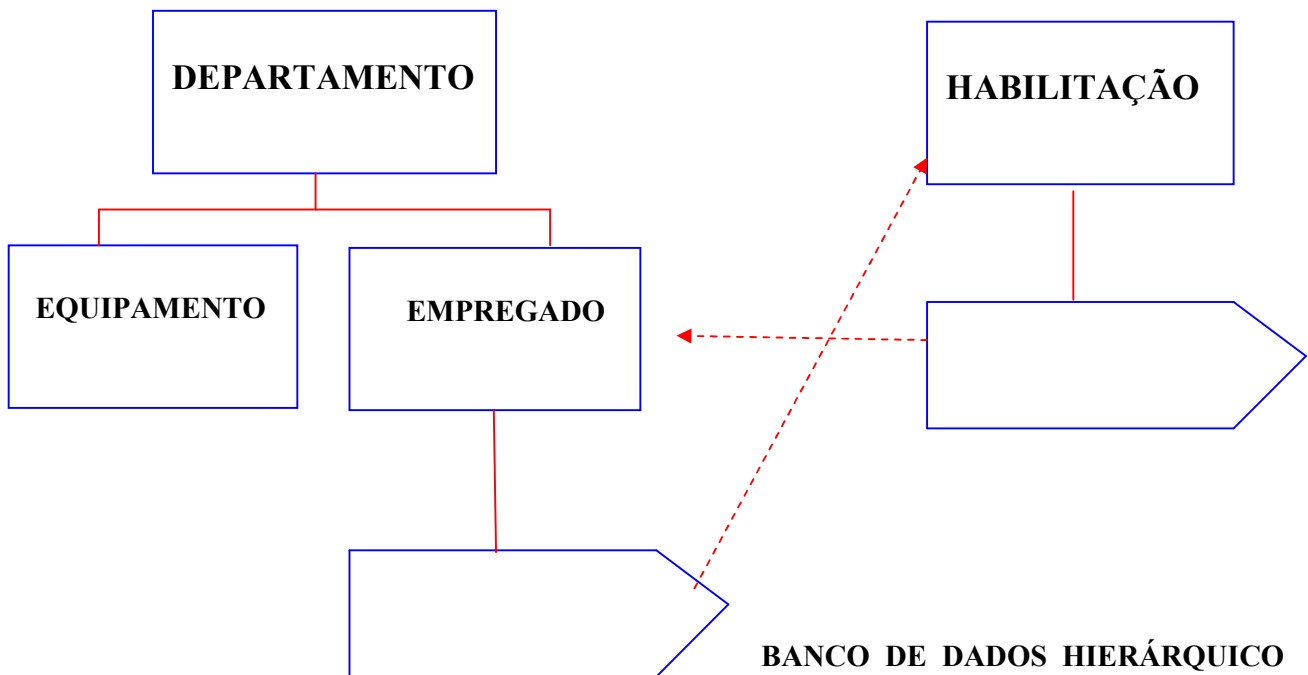
É importante lembrar que ambos os sistemas-hierárquicos e de rede são chamados de “sistemas de navegação” já que os dois casos, os programadores precisam atravessar um conjunto de registros pré-articulados (pré-interligados). Sabe-se que esta pré-articulação representa um benefício em termos de desempenho, embora seja restrita em termos da complexibilidade do processo do projeto e das modificações posteriores ao projeto.

## **MODELO DE BANCO DE DADOS HIERÁRQUICO**

A linguagem de dados da IBM CD/I-(DATA LANGUAGE I) constitui um relevante exemplo de gerenciamento de banco de dados, cuja o método utilizado o modelo Hierárquico. Onde esta linguagem faz parte em banco de dados de seu sistema de gerenciamento de informações (IMS-INFORMATION MANAGEMENT SYSTEM) e sistema de controle de informações sobre o cliente (CICS-CUSTOMER INFORMATION CONTROL SYSTEM).

Descreve uma Hierarquia como um conjunto de registros interligados, isto é , em cada um há um tipo de registro "PAIS", exceto a Raiz.

Numa Hierarquia, o acesso aos dados é utilizados através de segmento "Raiz", quando um campo é escolhido como campo - chave e os outros são baseados em índice ou acesso aleatório.



## **VALOR DOS DADOS NORMALIZADOS**

Há duas formas importantes de dados normalizados.

1. livre de redundância , entre campos não chave
2. sujeitos a possíveis modificações estruturais devidas a razões de desempenho, onde é constitua o projeto final de um banco de dados.

## **ESTRUTURAÇÃO DOS DADOS BANCO DE DADOS HIERÁRQUICO**

Em geral , ao se converter um conjunto de tabelas normalizadas em uma estrutura Hierárquica de dados, se transformam em nós, onde a apresentação de vários ponteiros para articularem as ocorrências de tipos iguais e registros diferentes.

## **MODIFICAÇÃO NO ACESSO E NO DESEMPENHO**

O projetista estuda uma alternativa para que os campos acessam aleatoriamente. Geralmente este acontecimento é mais inviável em segmentos da Raiz , onde o acesso é aleatório , portanto a presença de seus campos-chave (até os preparativos para esta propriedade constituem uma decisão a ser tomada).Suponha-se , porém , que uma aplicação requeira acesso aleatório a um campo que não seja um segmento raiz.

1)Na presença caso haja necessidade de se encontrar mecânicos, aleatoriamente, através do seu número. Ou suponha-se que uma aplicação requeira acesso aleatório a um campo não-chave, num segmento origem .

2)Que se queira encontrar lojas, aleatoriamente, por meio de cidade. Neste caso, devem-se empregar índices secundários ou reorganizar a estrutura de modo que certos segmentos não-raízes passem a ser raízes de novas hierarquias.

Em termos de desempenho, é preciso considerar-se , na DL/I, uma série de fatores, muitos dos quais se inter-relacionam: a alteração em um deles pode requerer balanceamento, ou alteração com efeito compensador , em outro.

De longas cadeias gêmeas - por hipótese , um número muito grande de mecânicos trabalhando em uma única loja - que podem exercer efeito nocivos sobre o desempenho, já que o sistema não permite acesso aleatório ao meio de uma tal cadeia , portanto esta cadeia deve ter acesso de seqüências e segmentos ligados por ponteiros.

## OUTRA CONSIDERAÇÃO SOBRE O DESEMPENHO

Refere-se ao número de ocorrências de um segmento raiz de uma hierarquia. Usando-se novamente a estrutura da figura 10.10, suponhamos que exista um número razoavelmente pequeno de lojas, mas que, em cada loja, haja um número substancial de mecânicos ou também que haja uma referência de programa que sejam para os mecânicos e que elas tenham o conhecimento de nome ou número, bem como sua loja.

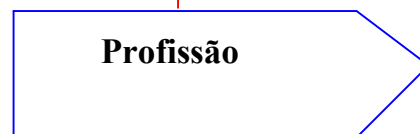
### REPRESENTAÇÃO DL/I, INCLUINDO UM PARENTESCO LÓGICO

#### *Segmento relativo á loja*

Nº da Loja	Cidade Da Loja	Suhev.
------------	----------------	--------

Nº do Mecânico	Nome do Mecânico	Idade do Mecânico
----------------	------------------	-------------------

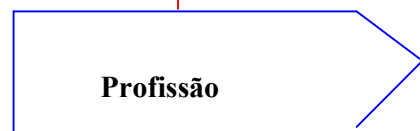
#### *Segmento relativo ao Mecânico*



#### *Segmento "filho lógico" Da proficiência*

Nº da Habilitação	Categoria Da Habilitação
-------------------	--------------------------

#### *Segmento relativo á habilitação*



#### *Segmento "filho lógico"*

Segmento relativo ao Mecânico

Segmento relativo ao Mecânico

## **MODELO DE BANCO DE DADOS RELACIONAL**

---

Um banco de dados relacional representa um afastamento significativo entre o modelo hierárquico e o modelo de rede. Arquivos muito simples e lineares não são articulados por meios artificiais, como ponteiros. A integração é executada através do software e não diante da estrutura dos dados, realizada nos sistemas de navegação.

Um banco de dados relacional visa:

- 📖 Manter os dados de forma não redundante ( repetição de vários campos em várias tabelas );
- 📖 Executar processamento integrado;
- 📖 Lidar com relações múltiplas ( relacionamentos);
- 📖 Fornecer certo grau de independência dos dados.

O sistema relacional compara todos os registros de um arquivo com os registros de outro arquivo, buscando correspondências entre valores do campo união em cada par de registros. Ao encontrar um par, os campos de ambos registros vão para um arquivo recém criado, selecionando e destacando a informação necessária através de outros operadores.

Diante de um processamento integrado, o banco de dados torna-se simplificado causando melhoria na independência dos dados com o modelo relacional, obtendo maior desempenho. Em arquivos grandes não podemos executar uma união completa, devido ao tempo requerido. Na procura de solução para o tempo requerido, organizações investem estudos na utilização de hardware e software, buscando técnicas de uso em larga escala de índices possibilitando:

- 📖 Acesso mais rápido, a um número maior de campos;
- 📖 Interarticulação de arquivos a baixo nível;
- 📖 Pré processador para responder a uma consulta com eficiência, antes do acesso ao banco de dados.

## **MODELO DE BANCO DE DADOS OBJETO**

---

Os bancos de dados orientados a objeto integram a orientação a objetos com aptidões de bancos de dados. Através de construções orientadas a objeto, os usuários podem esconder os detalhes da

implementação de seus módulos, compartilhar a referência a objetos e expandir seus sistemas através de módulos existentes. A funcionalidade de banco de dados é necessária para assegurar o compartilhamento simultâneo e a continuidade das informações nas aplicações. Através dos bancos de dados, os usuários podem obter o estado em que os objetos se encontram, e estar atualizados entre as várias solicitações de programa, e diversos usuários podem ao mesmo tempo compartilhar a mesma informação. Os mesmos combinam os benefícios e conceitos da orientação a objeto com a funcionalidade dos bancos de dados.

A orientação a objeto é definida como: = tipagem de dados abstratos  
+ herança  
+ identidade do projeto

Aptidões de bancos de dados são definidas assim: = continuidade  
+ concomitância  
+ transações  
+ recuperação  
+ filtragem  
+ atualização  
+ integridade  
+ segurança  
+ desempenho

Bancos de dados orientados a objeto = orientação a objeto + aptidões de banco de dados. O potencial de bancos de dados orientados a objetos reside na íntima integração destas duas tecnologias: **orientação a objetos e bancos de dados**

## **VERDADES E MENTIRAS SOBRE BANCOS DE DADOS OBJETOS**

Os **Sistemas de Gerenciamento de Bancos de Dados Orientados a Objetos (ODBMS)** foram uma das grandes idéias do início dos anos 80. Os objetos eram a próxima onda, pois o modelo de objetos permite a modelagem complexa de objetos da forma como eles existem, ao invés de tentar espremê-los em uma estrutura bidimensional, embora que os ODBMS não tenha uma firme base teórica nem ferramentas de desenho bem desenvolvidas, todo mundo estava interessado neles. Cientistas da computação trabalhavam em protótipos ( **produção de um software que se assemelha superficialmente a um sistema proposto, mais rápido e barato do que uma versão do sistema desenvolvida na linguagem alvo** ).

O único problema era que os primeiros ODBMS não eram sistemas de bancos de dados completos, pois careciam de funções de backup e recuperação e os modelos de dados eram conflitantes e as linguagens proprietárias. Devido a sua estrutura, era impossível fazer buscas de verdade e os softwares não eram escaláveis e exigiam muita memória. Muitos desistiram dos primeiros produtos,

segundo Jeff Jones, gerente de programa da IBM para o grupo de marketing de gerenciamento de dados, sua empresa tentou usar um ODBMS puro como banco de dados embutido no Visual Warehouse.

Ela obteve a licença do ObjectStore, da Object Design. Porém a performance foi tão fraca que a IBM o substituiu pelo DB2 nas versões seguintes e trabalhando para acrescentar recursos e reduzir a quantidade de código. Experiências como essas fizeram do ODBMS por mais que uma curiosidade de laboratório, exceto por alguns nichos de mercado, como telecomunicações, os programas permaneceram como uma tecnologia em busca de um problema. Foi nesse ponto que parou o conhecimento popular, enquanto a tecnologia de ODBMS evoluía e as necessidades dos usuários mudavam a sabedoria popular dizia que os softwares eram problemáticos, mas não são. O crescente interesse em aplicações multimídia e na Internet criou novos mercados para ODBMS, dessa perspectiva, a tecnologia de ODBMS é ideal para as aplicações mais populares. O enorme crescimento da Internet, videogames, aplicações multimídia e o desenvolvimento de bancos de dados distribuídos que não se prestam ao modelo relacional está trazendo atenção renovada para o ODBMS. Como Java é uma linguagem orientada a objetos, aplicações na Internet são particularmente adequada para bancos de dados de objetos. Como agora há padrões de fato para a tecnologia de objetos, é possível disponibilizar uma aplicação para mundo inteiro, as telecomunicações representam um bom mercado. Muitos produtos, hoje fazem a modelagem e o controle de redes de comunicação em tempo real.

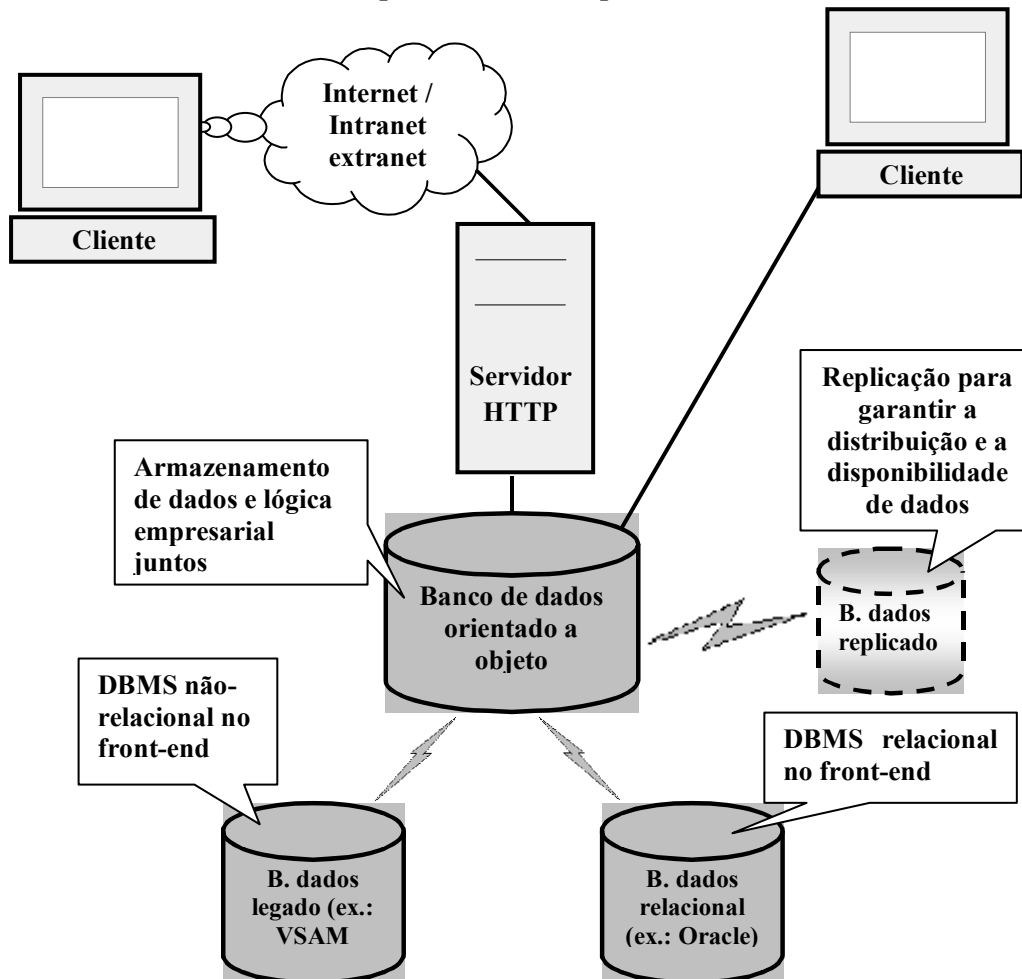
## COMO ESCOLHER UM BANCO DE DADOS

Há oito áreas-chaves a examinar antes de comprar um banco de dados de objetos:

- 📖 **Suporte a linguagem** - quais as linguagens necessárias: Java C++, OQL? Algumas linguagens proprietárias são mais velozes que OQL, mas um produto que utilize uma linguagem padrão será mais flexível e portátil.
- 📖 **Escalabilidade** - qual o maior banco de dados que o produto suporta? Qual o maior banco de dados já pronto e em funcionamento usando o produto? Quantos usuários acessam o banco de dados de uma só vez?
- 📖 **Segurança** - como é determinada a segurança - por usuário, grupo, ou ambos?
- 📖 **Backup e recuperação** - como o produto lida com backup e recuperação?
- 📖 **Transações** - como o banco de dados lida com registro, recuperação e estorno de transações?
- 📖 **Métodos** - como os ODBMS armazenam métodos? Para ser um verdadeiro ODBMS, ele deverá armazená-los no próprio banco de dados.

- 📖 **Classes de coleções** - com quais classes de coleções o banco de dados lida? ODMG, Java e diversas bibliotecas de classes de objetos, notavelmente STL definiriam certas coleções de classes comuns. O uso de classes de coleções padronizadas aumenta a portabilidade e a flexibilidade.
- 📖 **Suporte e treinamento** - que tipo de suporte e treinamento o fabricante oferece? Por quanto tempo ele treinará a equipe do comprador?

### O que um ODBMS pode fazer



Um projetista de banco de dados relacional pode usar uma ferramenta de diagramação entidade/relacionamento para verificar matematicamente se o projeto está na forma terceira normal. Um projetista de **ODBMS nem sequer tem um conceito similar a formas normais para seus objetos**. O problema das ferramentas está sumindo, o Jasmine, da Computer Associados, é um exemplo de ambiente de desenvolvimento muito bom.

Hoje, todo mundo se confunde com os bancos de dados relacionais estendidos, objeto-relacionais e os bancos de dados de objetos (puros). O maior problema do ODBMS puro é seu nome, o termo deveria ser banco de objetos (objectbase), ao invés de banco de dados de objetos, porque o



objetivo não é armazenar, manipular e recuperar dados dentro de um objeto, mas sim armazenar, manipular e recuperar os próprios objetos. Os bancos de dados de objetos puros permitem consultas mais simples sobre dados complexos.

Um banco de dados de objetos puros tem métodos, classes e outras coisas que caracterizam o modelo orientado a objetos no núcleo do banco de dados. Os objetos são ativos, diferente dos

relacionais que são passivos, e é preciso um programa hospedeiro para fazer alguma coisa com eles. Não se deve confundir ODBMS com produtos relacionais estendidos e objeto-relacionais. Eles são

projetados para solucionar um conjunto de problemas diferente. Usar um banco de dados objetos puros para armazenar dados relacionais é como manter as peças dos automóveis na forma de carros inteiramente montados e desmontar toda a frota quando é preciso contar os parafusos que existem no estoque. O usuário acaba perguntando se essa é a forma mais eficiente de se fazer um inventário.

Os ODBMS não têm uma linguagem padrão, portanto eles não são bons para desenvolvimento sério. Essa sabedoria é uma verdade convencional, mas não de fato. Nos próximos anos, ela pode se tornar falsa, enquanto o processo de padronização ISO estava em andamento, o Object Database Management Group (ODBMG), um grupo de fabricantes de ODBMS, começou a tentar padrões para bancos de dados de objetos fora da estrutura ISO. O ODMG produziu um padrão para uma linguagem de consultas para ODBMS em 1.993, sob o nome OQL. Vários fabricantes já concordaram em dar suporte ao OQL.

Os bancos de dados de objetos estão em amadurecimento, são mal entendidos e difíceis de usar, mas estão ganhando aceitação, graças ao explosivo, e um tanto especulativo, crescimento da Internet e da multimídia (voz, texto, gráficos).

Uma advertência: a passagem para a tecnologia de objetos pode ser difícil, por causa dos pesados investimentos financeiros e humanos em tecnologia relacional. Projetos envolvendo um banco de dados de objetos demorarão mais tempo com uma equipe sem experiência. Mesmo para quem estiver lidando com equipe experiente, o projeto demorará mais será mais dispendioso por causa da inerente complexidade da tecnologia de objetos. Quem estiver preparado para isso, estará pronto para um banco de dados objetos.

## **MODELO DE BANCO DE DADOS DATAWAREHOUSE**

---

Datawarehousing ( sistema e bancos de dados projetados para análise de comportamento e tendências de negócio).

Esqueça tudo o que se diz sobre Datawarehousing. Essa é melhor maneira de preparar-se para entender como construir este sistema de W.H.Innon voltado, principalmente para gerentes e analistas de sistemas. Os armazéns de dados não são feitos de programas e máquinas, mas de um extenso trabalho de análise e depuração de dados.

Para muitos profissionais, a obra de Innon mostrará que seu conhecimento pode ter-se tornado obsoleto. É que ao contrário dos sistemas tradicionais, os projetos de Datawarehousing partem da implantação para chegar as necessidades do usuário.

Quem possui experiência em desenvolvimento de sistemas irá, com certeza, encontrar situações familiares. Os programas extratores são um caso típico. Já na década de 70 antes que o termo Datawarehousing ganhasse fama, gerentes pediam a programadores que enviassem relatórios e arquivos resumidos, com dados extraídos de grandes de dados corporativos.

Depois de explorar as origens do Datawarehousing, é possível montar passo a passo um projeto completo, melhor dizer mês a mês, pois o trabalho envolvido é digno de Hércules, dado a necessidade de filtragem e conciliação de todos os dados e sistemas de empresa. Mas como o próprio Innan diz, se a empresa esperar condições favoráveis para a implantação do Datawarehousing, ela nunca o fará.

Um exemplo atual do poder de um Datawarehousing:

Mais do que uma expressão, se transformou em um bordão repetido a cada duas frases por todos os funcionários da NCR Corporation, em todas as suas apresentações públicas. O exemplo partiu do próprio Chief Executive Officer (CEO) e chairman, Lars Nyberg, na abertura da Partners, a conferência anual de usuários da empresa, realizada entre os dias 5 e 9 de outubro, em San Diego, na Califórnia, e que reuniu cerca de 700 clientes vindos de todas as partes do mundo.

A NCR está direcionada a se tornar na indiscutível líder mundial em Datawarehousing, não só para os bancos e a indústria. Mais e mais os sistemas de Datawarehousing estão se tornando cruciais para todas as empresas que prestam serviços a cliente. A sua capacidade de prever one-to-one vai tornar essa tecnologia, pouco tempo, não um luxo, mas uma necessidade para todos os negócios que esperam se manter competitivos, disse Nyberg ao traçar a estratégia da empresa para os próximos três anos.

Para alcançar a meta de ganhar liderança, a NCR tomou três providências. A primeira foi tornar o seu sistema de gerenciador de banco de dados Teradata um produto mais aberto, capaz de rodar em diversas plataformas – e o primeiro passo é portá-lo para Windows NT . Isso vai acontecer, segundo Nyberg, até o fim do ano, de forma que a nova versão esteja disponível, comercialmente, no início de 98. E o NT, prometeu também o chairman, será a plataforma fundamental de todas a soluções da NCR

para ambientes que exijam alta disponibilidade e para comércio eletrônico, assim como para soluções específicas para bancos e comércio.

Outra medida da NCR foi o lançamento de duas novas plataforma de hardware, integrante a família de servidores WorldMark, voltada especificamente para soluções Datawarehousing, e que o Aberdeen Group, empresa de pesquisa de Boston, definiu como servidores mainframe.

O modelo 4700 é construído com quatro processadores Pentium Pro de 200 Mhz, por módulo, cada um com 512 cache de memória. É possível colocar até dois módulos em cada gabinete e, com oito módulos alinhados em cluster, se chega a uma plataforma capaz de suportar um sistema de Datawarehousing com 600Gb de dados.

O modelo 5150 pode ter até 128 módulos conectados, suportando, assim, acima de 600Gb e até 100Terabytes. Os dois novos servidores podem rodar o gerenciador de bancos de dados Teradata, da própria NCR, para aplicações Datawarehousing, e, ainda, o Informix XPS ou o Oracle Parallel Server para processamento de transações online. E, finalmente, a terceira providência da NCR para garantir a liderança em projetos de Datawarehousing foi celebrar uma aliança com a SAS Institute, segundo a qual as duas empresas vão combinar suas expertises em Datawarehousing e data mining para prover soluções completas e integradas para clientes de todos os portes e em todos os setores da economia.

## CONCLUSÃO

Para que uma organização cresça no mercado empresarial e saiba lidar com novas tecnologias, ela deve se preocupar especialmente, com o modo de organizar suas informações.

Um meio eficaz de gerenciamento de informações, somente é realizada através de um Banco de Dados. Além de interligar todo trabalho da organização, reduz custos, elimina duplicação de tarefas, permite uma previsão de crescimento da empresa e ajuda na elaboração de estratégias.

Fazendo uma análise dos exemplos de banco de dados, podemos perceber que é necessário uma busca de novas implementações em seu desenvolvimento, visto que para cada modelo de banco de dados existe vantagens e desvantagens.

Dando um maior enfoque ao banco de dados corporativo, foi verificada sua importância na exatidão, rapidez com que as informações devem ser conduzidas, além da segurança ao usuário. Como referência, foi estudado o banco de dados do aplicativo Lotus Notes, que mesmo possuindo uma vasta gama de recursos, verificou-se um grau de dificuldade em sua manipulação.

Concluindo a sua importância, verifica-se a disponibilidade de filtrar todas as informações de uma organização por meio de um banco de dados que realmente transmita segurança, qualidade e competitividade.

## **BIBLIOGRAFIA**

### ***Rápido Fácil***

Lotus Notes for Windows

Autor(es): Andrew Bryces Shafran

Editora: Afiliada

### ***Organização de Banco de Dados***

Autor(es): A L. Furtado

C.S. dos Santos

Editora: Campus - 7ª Edição

### ***Sistema de Banco de Dados***

Autor(es): Henry F. Korth

Abraham Silberschatz

Editora: Mc Grow Hill

### ***Planejamento Estratégico Análise de Sistemas e Projeto de Banco de Dados***

Autor(es): Mark L. Gillenson

Robert Goldberg

Editora: Livros Técnicos e Científicos Editora S/A

## **REVISTAS**

☞ Informática Hoje  
ano 12 – nº 440 - 1ª a 15 de Novembro/97

☞ Info Exame  
ano 12 – nº 142 - Janeiro/98

☞ Byte Brasil  
Ano 6 – nº 10 - Outubro/97

**OBS.:** Também formam pesquisados sites na Internet