Unidade IV

7 ADMINISTRAÇÃO DE BANCO DE DADOS

Nas unidades anteriores, apresentamos os conceitos fundamentais necessários para o entendimento e a criação de um banco de dados. Na fase de concepção e criação, o administrador de banco de dados está mais próximo do administrador de dados. Após a implementação física do projeto de banco de dados, isso não significa que o trabalho acabou, muito pelo contrário, é agora que as habilidades e experiência do DBA serão colocadas em prática, pois o trabalho de ajustes de desempenho (*tuning*) e a manutenção e a garantia de que tudo funcione da maneira desejada vão depender de ações contínuas desse profissional.

Os conceitos descritos a seguir serão apresentados de forma genérica, pois o ideal é seguir as recomendações oferecidas pelos fornecedores de banco de dados. Parece estranho, mas não é. Não adianta você comprar uma Ferrari, por exemplo, e estudar o manual do Gol. Se você quer tirar o máximo de proveito de todas as funcionalidades da Ferrari, terá que se tornar um especialista. A analogia parece absurda, mas funciona para banco de dados.

De modo geral, os SGBD costumam armazenar os dados e as estruturas de bancos de dados em locais e arquivos diferentes de onde são armazenadas as informações sobre as operações efetuadas sobre os dados e as estruturas.

Por exemplo, o SGBD *Microsoft SQL Server* armazena os dados em dois arquivos distintos:

- Um arquivo com a extensão mdf
  - Arquivo principal do banco de dados.
  - Nesse arquivo são gravados os dados e a estrutura do banco de dados.
- Um arquivo com a extenção ldf
  - Arquivo de *log* do banco de dados.

Grava todas as operações realizadas dentro do banco de dados.

Funciona como um diário de tudo o que acontece lá dentro.

Observação

Os fornecedores de banco de dados oferecem cursos específicos para capacitar os profissionais de administração de banco de dados. Nem sempre é uma tarefa trivial, em que basta ler um manual e seguir.

A escolha do banco de dados da empresa é uma decisão muito delicada, na medida em que está irá acarretar troca de aplicativos e de *hardware*. Os investimentos diretamente aplicados no banco de dados costumam ser infinitamente menores do que aqueles a serem aplicados na empresa, visando sua perfeita adequação ao novo SGBD. Essa decisão, sempre que possível, deve ser tomada por especialistas em banco de dados, com profundos conhecimentos de análise de sistemas, de banco de dados e de *software* de gerenciamento de banco de dados, de forma a evitar que a empresa escolha um banco de dados inadequado aos seus propósitos e, pouco tempo depois, seja obrigada a perder todos os investimentos realizados em *software* e *hardware*.

7.1 *Backup*

O *backup* é considerado como uma das atividades mais importantes e críticas de administração de banco de dados. Consiste em fazer uma cópia de segurança (*dump*) dos dados e da estrutura do banco de dados, e das operações realizadas na linha do tempo, entre uma cópia e outra.

O *backup* em tempo de execução é uma característica sempre disponível em todos os tipos de SGBD, porém temos aplicações que invariavelmente são comprometidas por falhas de *hardware* e outras cujo mesmo tipo de falha não causa perda alguma de dados ou de integridade.

Novamente, cada SGBD tem essa característica melhor ou pior implementada, cabendo ao administrador de banco de dados escolher aquele que lhe oferecer mais segurança.

Normalmente, as falhas podem ser de dois tipos: *soft crach* e *hard crash*. A primeira são falhas do sistema (por exemplo, queda de energia) que afetam todas as transações em curso no momento, mas não danificam fisicamente o banco de dados. A segunda são falhas da mídia (por exemplo, queda da cabeça de gravação sobre o disco) que causam danos ao banco de dados ou a uma parte dele e afetam pelo menos todas as transações que, no momento, estão usando essa parte.

7.1.1 Recuperação e concorrência

Os problemas de recuperação e concorrência de dados estão ligados a processamento de transação.

Lembrete

Transação é uma unidade de trabalho lógica, ou seja, é uma sequência de operações que transforma um estado consistente de banco de dados em outro estado consistente, sem que os pontos de consistência intermediários sejam preservados.

O sistema não deve permitir que transações sejam executadas em parte (ou seja, o programa pode terminar de forma anormal durante as atualizações), pois isso levaria o banco a um estado inconsistente. Por isso, sistemas que suportam o processamento de transação têm uma melhor forma de segurança.

Essa segurança é resumida em quatro tópicos, a saber:

- **Atomicidade**: ter a certeza de que uma transação foi totalmente executada ou totalmente cancelada. Nunca uma execução pode ficar pela metade.
- **Consistência**: não permanecer com estágio intermediário, deve ir de um estágio consistente a outro.
- **Isolamento**: uma transação, quando executada, tem que ser executada independente de outra. Não importa se elas estejam sendo processadas separadas ou de forma concorrente, o resultado terá que ser o mesmo. Existe um sistema gerenciador de transação que permite isso.
- **Durabilidade**: garantir que todas as transações não realizadas permaneçam no banco e sejam realizadas assim que o computador for reinicializado. Se, por algum problema, o computador desligar e o disco não for danificado, a durabilidade tem que estar funcionando.

Observação

A atomicidade é proporcionada por meio das operações **COMMIT** e **ROLLBACK**.

7.1.2 Pontos de sincronização

O ponto de sincronização representa a ligação entre duas transações consecutivas, mostrando onde o banco de dados está (ou deveria estar) em estado de consistência. As únicas operações que apresentam este ponto são **COMMIT** e **ROLLBACK**.

- **COMMIT:** garante que a transação foi finalizada com êxito. O banco de dados (depois do término da transação) encontra-se em estado de consistência, e todas as suas atualizações são permanentes.
- **ROLLBACK:** assinala que a transação foi malsucedida. O sistema informa que algo saiu errado e que todas as atualizações devem ser refeitas, ou seja, elimina vestígios do que foi feito.

Observação

As operações **COMMIT** e **ROLLBACK** terminam a transação e não o programa.

Todas as transações são acumuladas em um *buffer*, que, de tempo em tempo, atualiza o banco; caso ocorra uma queda de energia, as operações efetuadas até então são perdidas. Assim, para melhor segurança, é necessário que esteja presente sempre um arquivo de *log*.

A finalidade desse arquivo é registrar cronologicamente todas as transações realizadas, assim, quando houver uma falha, é possível fazer recuperação. O *log*, por sua vez, não pode ficar no *buffer*, já que assim que é desligado o computador, tudo o que está na memória é perdido.

7.2 Meios de recuperação (*restore*)

O sistema deve ser preparado para se recuperar não só das falhas locais (danos causados apenas na área em que estas ocorreram), mas também das falhas globais (danos em diversas áreas, havendo implicações expressivas no sistema). Há duas categorias de falhas, descritas a seguir.

7.2.1 Falhas do sistema

Normalmente, são causadas por queda de energia ou desligamento do computador sem finalização correta, que não danificam fisicamente o banco de dados. Assim, não se terá mais conhecimento do momento exato da transação que estava processando, consequentemente, ela será desfeita.

Ainda há as transações que foram realizadas com sucesso, mas que não tiveram tempo de ser transferidas da memória intermediária do banco de dados para o banco de dados físico.

Uma maneira de se controlar é utilizando um método conhecido como *check point* (ponto de controle), de tempo em tempo o sistema faz as anotações cronológicas. Tem como objetivo:

- passar fisicamente o conteúdo das memórias intermediárias do banco de dados para o banco de dados físico;
- passar fisicamente o registro especial do ponto de controle para outra parte que não a anotação cronológica física.

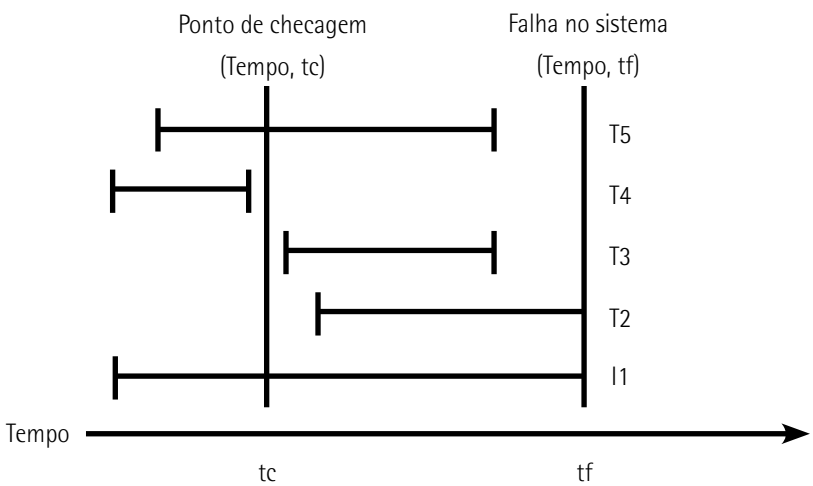


Figura 36 – Linha do tempo de uma transação RED

Nesse exemplo da figura 36, o ponto de checagem foi feito quando as transações T1 e T5 já tinham começado e a transação T4 já tinha acabado. A T3 começou e terminou depois do ponto de checagem, e a T2 começou depois e não terminou a sua execução. Neste caso, algumas terão que ser **refeitas** e outras **desfeitas**.

Tabela 1

Refeitas	Desfeitas
T3	T1
T5	T2

As transações T3 e T5 terão que ser refeitas, pois, antes de acontecer a falha do sistema, elas já tinham sido finalizadas. Então, o conteúdo concluído até o ponto de checagem precisa ser confirmado e, assim, finalizar de forma correta a transação.

Já T1 e T2 precisam ser desfeitas, pois, quando a falha ocorreu, a sua execução não havia terminado. É necessário que apague tudo o que foi feito, ou seja, desfazer, para que seja reiniciada a transação.

7.2.2 Falhas dos meios físicos

São as falhas que ocorrem danificando o disco rígido, ou seja, certa parte dos dados é destruída. Exemplo: colapso de uma cabeça de disco ou falha no controlador de disco. É importante sempre ter cópia do banco (*backup*), pois, em uma falha dos meios, se podem perder inúmeras informações, o que irá trazer prejuízo ou danos ao usuário.

Nesse caso, o sistema precisa ser recriado em outra máquina. Para que o impacto dessas ações de reconstrução do ambiente seja o menor possível, na visão do usuário, existem diversas soluções que envolvem não só a substituição do disco danificado, mas todo um projeto de contingência e alta disponibilidade do seu ambiente como um todo.

A arquitetura do seu ambiente deve ser projetada em conjunto com a equipe de analistas de suporte e de redes. Projetos desse tipo costumam envolver a alta administração da organização, pois o investimento é alto e às vezes significa a contração de serviços de *data center* especializados. Tudo vai depender do valor dos dados da empresa e de quanto vale o risco de perder os dados.

7.3 Concorrência

Em um sistema em que existe a possibilidade de se ter inúmeras transações ocorrendo ao mesmo tempo, há a necessidade de um mecanismo de controle de concorrência, que assegure que uma transação não interfira em outra. Alguns problemas que podem surgir com a ausência deste mecanismo são: perda de atualização, dependência de uma transação não confirmada, análise inconsistente. Para evitar tamanho transtorno, é utilizado um mecanismo chamado bloqueio (*lock*), que permite segurança e consistência no banco de dados.

Para entender melhor o que significa a perda de atualizações, vamos considerar a seguinte situação: duas pessoas editam o mesmo registro, uma altera o telefone e outra, o endereço; ao confirmarem, será gravada a última atualização. Isso ocorre por não haver nenhum tipo de controle no banco, mas essa situação pode ser contornada configurando-se o sistema para que não libere o registro quando ele estiver sendo usado por outro usuário.

O problema da dependência de uma transação não confirmada é que isso pode retornar para o usuário informações erradas, comprometendo muito a utilização do banco. Suponha que ocorra a seguinte situação: uma transação A faz um *update*, logo em seguida, uma transação B faz uma leitura no mesmo dado. Se, por algum motivo, houver um *rollback* na transação A, a transação B terá visto um dado que não mais existe, ou seja, o *rollback* desfez tudo da transação A, e a transação B fica desatualizada, oferecendo ao usuário informações inconsistentes.

No exemplo a seguir, fica clara essa situação: a transação A teve sua alteração e, antes mesmo de ela ser confirmada, houve um *rollback*, resultando na desatualização dos dados fornecidos na transação B.

Tabela 2

Transação A	Tempo	Transação B
UPDATE	T1	-
-	T2	*READ*
ROLLBACK	T3	-

O problema da análise inconsistente: consideremos a seguinte situação em uma conta bancária:

Tabela 3

Transação A	Tempo	Transação B
Soma 50	T1	-
Soma 20	T2	-
-	T3	*READ*
-	T4	Transferência (20)
-	T5	*COMMIT*
Total 70	T6	-

Como pode ser observado, as transações A e B estão utilizando a mesma conta bancária. A primeira faz a soma, e a segunda realiza uma transferência de dinheiro. O total da transação A retornará um resultado inconsistente, pois os dados alterados pela transação B não foram atualizados, e como ocorreu o *commit* (confirmação da transferência), a conta não teve o devido débito.

7.4 Bloqueio (*Lock*)

Havendo duas transações, com ambas precisando do mesmo dado, elas deverão ser executadas uma após a outra; isso é possível quando se utiliza a técnica do bloqueio.

Existem dois tipos de bloqueios:

- **Bloqueio compartilhado:** pode haver inúmeras pessoas utilizando o mesmo dado. Quando uma transação B solicita um dado que está sendo utilizado pela transação A, B fica em estado de espera até que A libere o bloqueio.

- **Bloqueio exclusivo:** ninguém mais tem acesso àquele dado.

Todas as operações de banco de dados geram bloqueios, até mesma a leitura.

Observação

Não pode haver dois bloqueios exclusivos nem um compartilhado e exclusivo, senão não é exclusivo.

A técnica do bloqueio pode resolver os problemas encontrados na concorrência. Como foi visto, se um mesmo registro estiver sendo utilizado ao mesmo tempo por usuários diferentes, e estes, por sua vez, fizerem alterações nele, permanecerá a última mudança feita. Com o bloqueio, as transações A e B podem utilizar o mesmo registro, sendo que o último a solicitar os dados permanece em estado de espera, embora haja desta forma um conflito entre os bloqueios gerado pelas transações.

Por razões análogas, a transação que primeiro utilizava o registro entra em estado de espera, e a outra transação é executada. E assim temos um problema que o bloqueio traz: o impasse, em que as duas transações são incapazes de prosseguir, eliminando desta forma a perda de atualização.

Outro cenário possível é quando a transação B solicita um registro que está sendo utilizado pela transação A, nesse caso, a transação B entra em estado de espera. Enquanto A não alcançar o ponto de sincronismo (ROLLBACK ou COMMIT), B não é liberado para prosseguir. Assim que A terminar sua execução, B é liberado e, neste ponto, ele já pode obter um valor confirmado pela transação anterior, seja por ROLLBACK ou por COMMIT, mas, de qualquer maneira, B não dependerá de uma atualização não confirmada.

Se uma transação A está em bloqueio compartilhado, e B solicita implicitamente um bloqueio exclusivo no mesmo registro, B fica em estado de espera até que A libere. Assim que liberado, a transação A não consegue mais ser executada, pois a solicitação implícita de bloqueio compartilhado entra em conflito com o bloqueio exclusivo feito por B, forçando um impasse no qual ambos ficam na espera.

Observação

Impasse é uma situação em que as transações ficam paradas à espera da outra para liberar bloqueio. Isso pode acontecer com várias transações, não necessariamente com duas. Quando houver um impasse, é necessário que o sistema detecte, liberando os bloqueios, permitindo assim que continue a transação. A maneira como será feito isso, o programador resolverá, mas é importante que o usuário não tenha conhecimento disso.

7.5 Arquitetura básica do ambiente de banco de dados

Você pode ter um ambiente de alta disponibilidade e desempenho por meio da arquitetura de *hardware* combinada com os recursos de *software*. As técnicas básicas envolvem dois conceitos: *cluster* e replicação.

Cluster é nome dado a um conjunto de computadores ligados em rede que funcionam como se fossem um único computador, compartilhando recursos de memória, armazenamento e processamento. Essa técnica faz parte do plano de contingência e alta disponibilidade, pois você pode desenvolver o seu projeto de banco de dados distribuídos.

7.5.1 *Cluster* de alto desempenho

Também conhecido como *cluster* de alta *performance*, ele funciona permitindo que ocorra uma grande carga de processamento com um volume alto de *gigaflops* em computadores comuns.

7.5.2 *Cluster* de alta disponibilidade

São *clusters* cujos sistemas conseguem permanecer ativos por um longo período de tempo e em plena condição de uso. Pode-se dizer que eles nunca param seu funcionamento, além disso, conseguem detectar erros, protegendo-se de possíveis falhas.

7.5.3 *Cluster* para balanceamento de carga

Esse tipo de *cluster* tem como função controlar a distribuição equilibrada do processamento. Requer um monitoramento constante na sua comunicação e em seus mecanismos de redundância, pois, se ocorrer alguma falha, haverá uma interrupção no seu funcionamento.

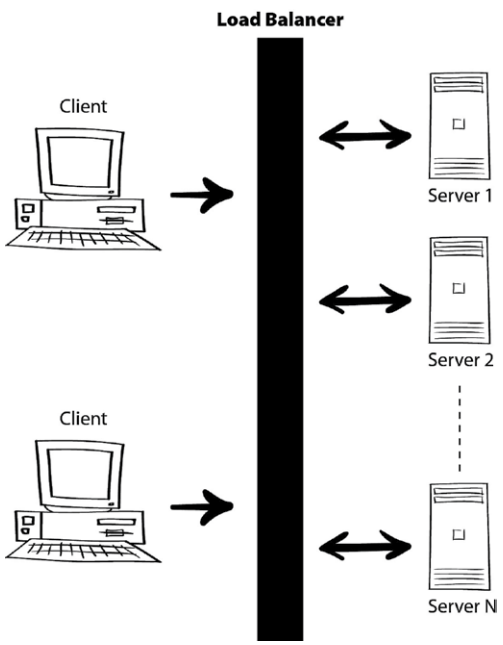


Figura 37 – Exemplo de esquema de um cluster

7.6 Replicação

Replicação de banco de dados é a cópia dos dados do banco de dados original para outro banco de dados. São vantagens da replicação em banco de dados:

- sistema menos sensível a falhas, por causa da redundância;
- trabalho com balanceamento da carga;
- *backup on-line* dos dados.

A replicação de banco de dados também pode ser usada para garantir a integração e a integridade entre os bancos de dados de sistemas com modelo de dados relacional distribuídos. Nesse caso, o sistema gerenciador de replicação de dados precisa ser capaz de replicar as transações em tempo real. Geralmente, os servidores de replicação só conseguem replicar uma imagem do banco (*snapshot*), portanto, não dá para implementar uma arquitetura desse tipo.

Saiba mais

Para conhecer essa solução, recomendamos a leitura do *case* de sucesso da USP, que está disponível no *site* da Sybase por meio do *link* <http://www.sybase.com.br/files/Success_Stories/USP_Replication_Server.pdf>. Acesso em: 14 jun. 2012.

7.6.1 Replicação síncrona (*eager*)

A transação só é concluída após todos os servidores fazerem *commit*. É conhecida como *phase 2 commit*, ou seja, o *commit* é realizado em duas fases. Apesar de garantir consistência de transação entre servidores, é de baixa escalabilidade. Outra característica dessa arquiteta é a indisponibilidade em caso de queda de rede. Foi muito pesquisada nos últimos dez anos, várias implementações foram realizadas, mas é considerada impraticável para a maioria dos ambientes de produção por ser de altíssimo custo e de difícil manutenção.

7.6.2 Replicação assíncrona (*lazy*)

A transação é concluída localmente e depois replicada, portanto, com alta escalabilidade. Não garante consistência de transação direta entre os servidores. Para ter essa consistência, a implementação deve ser modelada e implementada de tal forma que garanta essa consistência. É de baixo custo e resistente a quedas de rede. Esse tipo de replicação foi implementado na USP e é usado até hoje.

7.6.3 Replicação unidirecional (*master-slave*)

A replicação é realizada sempre no mesmo e único sentido, de um banco de dados A para um banco de dados B. É usado normalmente para *hot-backup* de servidores de banco de dados e também utilizado para melhoria de desempenho de consultas em *sites* remotos. Apenas a base *master* recebe atualizações. Pouco sujeito a inconsistências, mesmo no modelo *lazy*.

7.6.4 Replicação multidirecional (*multi-master*)

A replicação é realizada em vários sentidos. Usada para garantir alta disponibilidade e garantir melhor desempenho tanto em consultas quanto em atualizações. Todas as bases podem receber atualizações. Sujeito a inconsistências no modelo *lazy*.

7.6.5 Exemplo de funcionamento de um *cluster* com replicação

Na figura a seguir, temos uma estrutura em *cluster* de alta disponibilidade, em que os servidores de banco de dados estão em uma estrutura de replicação *master-slave*.

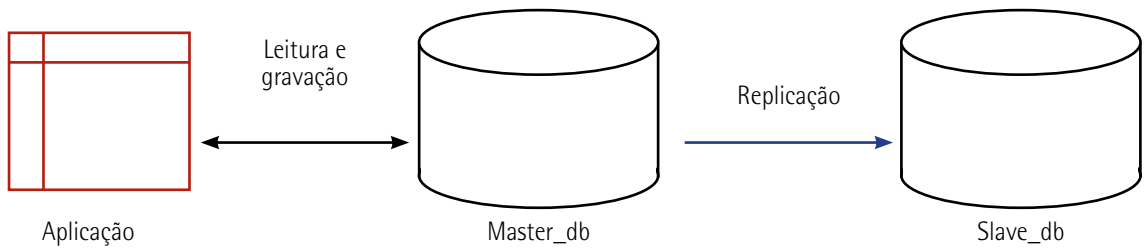


Figura 38 – Estrutura de *cluster*

Suponha que o servidor principal (*master*) de banco de dados apresente alguma falha e o torne indisponível. Existem diversos *softwares* por meio dos quais você consegue configurar o monitoramento da comunicação entre os servidores. Assim que for detectada a falha de comunicação, ela deve ser redirecionada automaticamente para conectar com o servidor replicado; em alguns minutos, a aplicação começará a funcionar a partir do servidor (*slave*).

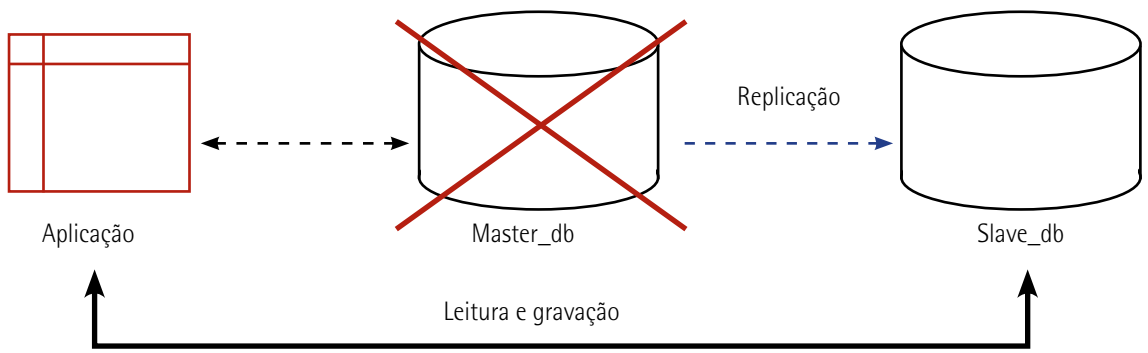


Figura 39

8 MODELO DIMENSIONAL

A modelagem dimensional é uma técnica voltada especialmente para a implementação de um modelo que permita a visualização de dados de forma intuitiva e com altos índices de *performance* na extração de dados. É a técnica de projeto mais utilizada para a construção de *data warehouses* (DW), que busca um padrão de apresentação dos dados de fácil visualização pelo usuário final e bom desempenho para consultas.

O modelo dimensional, ou multidimensional, detecta os relacionamentos inerentes aos dados para armazená-los em matrizes multidimensionais conhecidas como cubo de dados ou hipercubo, caso o cubo possua mais de três dimensões, conforme a figura a seguir:

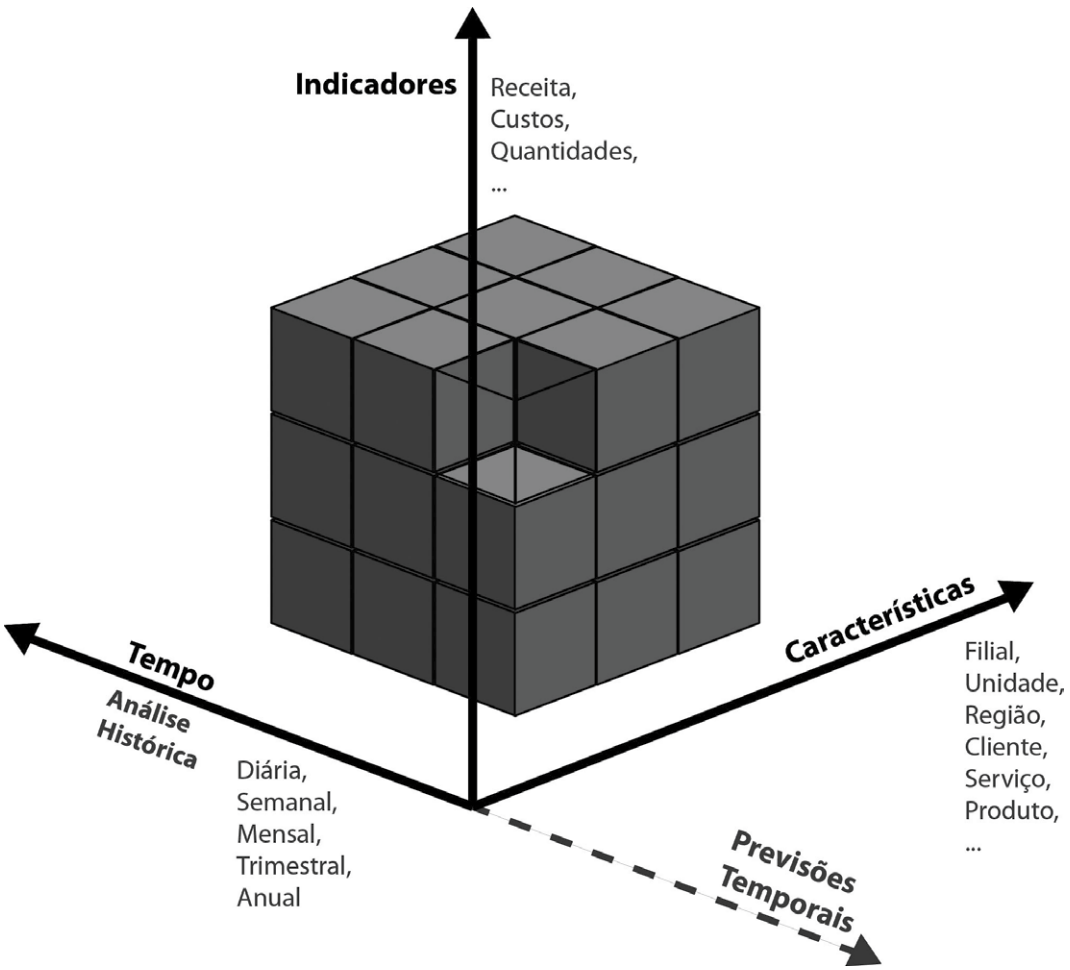


Figura 40 – Representação gráfica de um cubo de dados

Permite que os dados sejam consultados diretamente em qualquer combinação de dimensões, sem a necessidade de consultas complexas no banco de dados.

Ferramentas próprias proporcionam a visualização dos dados de acordo com as dimensões escolhidas pelo usuário.

O modelo dimensional é composto por uma tabela dominante, com múltiplas junções, chamada de tabela de fatos, e de um conjunto de tabelas conhecidas como tabelas de dimensão.

Tipos de esquema do modelo dimendisional:

- **Esquema estrela (*star schema*):** é um dos mais usados na modelagem dimensional e consiste em uma tabela de fatos central e uma tabela para cada dimensão.

- **Esquema floco de neve (*snowflake*):** é uma variação do esquema estrela em que algumas tabelas de dimensão são normalizadas, criando-se uma tabela de dimensão secundária cuja chave se torna estrangeira na tabela de dimensão primária. Essa técnica interfere no entendimento dos dados pelo usuário e na *performance* de algumas consultas, portanto, não deve ser usada como regra geral, embora às vezes possa ser interessante ou até necessária.

8.1 Fatos

Os fatos são os atributos do comportamento. São medidas de ações que ocorrem entre diferentes objetos ou dimensões. Em geral, são numéricos ou aditivos (podem ser somados) e raramente textuais.

A tabela de fatos é a tabela principal de um modelo dimensional. É formada por uma chave, composta pelas chaves das dimensões relacionadas e de um ou mais fatos mensuráveis, ou medidas, em geral numéricas, para cada conjunto das dimensões. Observe-se que muitas vezes não existe um valor associado para um cruzamento de dimensões.

O fato representa uma medição do negócio, isto é, fato é tudo aquilo que pode ser medido. A lista de dimensões define a granularidade da tabela de fatos (qual é o escopo da medição).

As características básicas de um fato são:

- variam ao longo do tempo;

- tem valores numéricos;

- seu histórico cresce com o passar do tempo.

Uma linha da tabela de fato corresponde a uma medição. Uma medição é uma linha da tabela de fatos. Todas as tabelas de fatos estão alinhadas com a mesma granularidade. As medições dos fatos devem ser numéricas e aditivas. As tabelas de fato representam relações N:N. As medidas podem ser classificadas em:

- **Valores aditivos**: medidas em que podem ser aplicados os operadores, tais como soma, porcentagem etc. Faz sentido adicioná-los continuamente e sobre todas as dimensões, por exemplo, vendas em R$ e vendas em unidades.

- **Valores não aditivos**: medidas que não podem ser manipuladas livremente, como porcentagtem ou valores relativos, tais como temperatura e condição do tempo.

8.2 Dimensão

As dimensões descrevem pessoas, lugares e coisas relacionadas a um negócio. As tabelas de dimensão são formadas por uma chave primária e por atributos que a descrevem. Quanto maior o tempo dedicado à descrição dos atributos, ao preenchimento dos seus valores e à garantia da qualidade dos dados, melhor será o DW.

Cada dimensão tem sua própria granularidade, que não pode ser menor que da tabela de fatos, mas pode ser maior.

Dimensões determinam o contexto em que ocorreram os fatos. No modelo dimensional, cada dimensão está associada a um ou mais fatos, sendo estas usualmente mapeadas em entidades não numéricas e informativas. As dimensões contêm descritores textuais da empresa.

As tabelas de dimensão são pontos de entrada para a tabela de fatos. Atributos de dimensões eficazes produzem recursos analíticos eficientes. As dimensões implementam a interface de usuário para o DW.

- maioria dos fatos envolve pelo menos quatro dimensões básicas: onde, quando, quem e o quê:
- dimensão **onde** determina o local em que o fato ocorreu (local geográfico, filial);
- dimensão **quando** é a própria dimensão tempo;
- dimensão **quem** determina quais entidades participaram do fato (cliente, fornecedor, funcionário);
- dimensão **o quê** determina qual é o objeto do fato (produto, serviço).

Observação

Existem casos em que algumas dimensões podem conter milhões de entradas, por exemplo, a dimensão cliente em uma companhia telefônica; neste caso, a navegação por essa dimensão pode tornar-se demorada. Pode-se, neste caso, utilizar índices nos atributos que sejam objetos de navegação.

Frequentemente, os campos mais utilizados em uma dimensão grande possuem um domínio pequeno, ou seja, assumem uma pequena quantidade de valores. Em uma dimensão cliente, esses atributos podem ser atributos demográficos, como sexo, faixa etária e classe social. Neste caso, pode-se optar pela criação de uma minidimensão separada da dimensão cliente para aumentar a eficiência da navegação.

Atributos como o número da nota fiscal de venda, aparentemente, deveriam fazer parte da tabela de fatos. Em um banco de dados relacional, ele seria o atributo determinante do cabeçalho da nota fiscal. Em um banco de dados dimensional, normalmente todos os atributos determinados pelo número da nota

foram armazenados em dimensões próprias e fariam parte da chave primária dos itens da nota. Ainda assim, pode-se utilizar esse atributo para agrupar os fatos pelo documento original. Atributos desse tipo são representados como dimensões degeneradas (descaracterizadas), isto é, chaves de dimensão sem uma dimensão correspondente.

8.2.1 Chave artificial (*surrogate key*)

Utilizar uma chave candidata como chave primária de uma dimensão pode causar problemas caso esta chave não seja absolutamente estável ao longo do tempo. Uma modificação em uma chave de uma dimensão pode ocasionar um grande volume de mudanças nas tabelas de fato relacionadas a essa dimensão.

A solução para esse problema é utilizar chaves artificiais absolutamente estáveis ao longo do tempo. Uma chave artificial é um campo inteiro e autoincremental, que aumenta a cada novo registro incluído na tabela de dimensão.

8.3 Medidas

Medidas são atributos que quantificam um determinado fato, representando o desempenho de um indicador em relação às dimensões que fazem parte do fato. O contexto de uma medida é determinado em função das dimensões do fato (MACHADO, 2000).

8.4 Granularidade

A granularidade é o nível de detalhe de um banco de dados dimensional. É um dos pontos mais importantes no projeto de um DW porque:

- refere-se ao nível de detalhe em que serão armazenados os dados no DW, quanto maior o detalhamento, mais baixo o nível de granularidade;
- afeta o volume de dados do DW e, portanto, a *performance* na extração de informações.

Um DW pode ser implementado em níveis duais de granularidade ao longo do tempo. É possível manter as informações mais recentes em um baixo nível de granularidade, aumentando assim as possibilidades de extração de informações. À medida que os dados vão ficando obsoletos, é possível resumi-los em um alto nível de granularidade de forma a manter a *performance*.

Observação

A granularidade define o nível de detalhe dos dados existentes no DW. Quanto maior o nível de detalhes, mais baixo o nível de granularidade e vice-versa. Ela determina o volume de dados do DW e o tipo de consulta que pode ser atendida.

Granularidade alta:

- Economia de espaço em disco.
- Redução na capacidade de atender consultas.

Granularidade baixa:

- Grande quantidade de espaço em disco.
- Aumento na capacidade de responder a qualquer questão.

Quadro 1

Fatos	Dimensões	Medidas
Representam um item, transação ou evento de negócio.	Determinam o contexto de um assunto de negócios, como uma análise de vendas de produtos.	São os atributos numéricos que representam um fato e são determinadas pela combinação das dimensões que participam dele.
Refletem a evolução dos negócios.	São os balizadores de análise de dados.	Representam o desempenho de um indicador de negócios relativo às dimensões que participam de um fato.
São representados por conjuntos de valores numéricos (medidas) que variam ao longo do tempo.	Normalmente não possuem atributos numéricos, pois são somente descritivas e classificatórias dos elementos que participam de um fato.	Podem possuir uma hierarquia de composição de seu valor.

8.5 Agregados

Agregados são resumos construídos a partir de fatos individuais, inicialmente por questões de *performance*, ou quando o ambiente dos fatos é inexpressivo na menor granularidade. Agregados permitem que as aplicações antecipem os resultados a serem pesquisados pelo usuário, eliminando a necessidade de se repetirem cálculos comumente realizados.

Múltiplos agregados podem ser construídos, representando os agrupamentos mais comuns dentro das dimensões do DW a fim de aumentar a *performance*. A contrapartida ao aumento da *performance* é a elevação do consumo de espaço de armazenamento.

Observação

Resumos armazenados com o objetivo de melhorar o desempenho de consultas, o agregado é composto por registros na tabela de fatos que representam o resumo do registro de nível básico da tabela de fatos, associados a registros agregados das dimensões.

8.6 Passos do projeto de DW

1) Decidir qual(is) processo(s) do negócio devemos modelar, por meio da combinação do conhecimento do negócio com o conhecimento dos dados que estão disponíveis.

2) Definir o grão do processo do negócio. O grão é o nível fundamental atômico de dados que representará o processo na tabela de fatos.

3) Escolher as dimensões que serão aplicadas a cada registro da tabela de fatos. Para cada dimensão escolhida, descrever todos os diferentes atributos de dimensão (campos) que preencham cada tabela dimensional.

4) Escolher os fatos mensuráveis que irão popular cada registro da tabela de fatos. Fatos mensuráveis são quantidades numéricas aditivas, como quantidade vendida e vendas (em espécie).

Funcionalmente, o processo de *data warehousing* engloba três etapas:

- extração dos dados operacionais de diversas fontes (banco de dados, planilhas, arquivos convencionais etc.);
- organização e integração dos dados em um banco de dados, o DW;
- acesso aos dados integrados de forma eficiente e flexível pelo usuário final.

Fontes de dados: dados extraídos, em geral, do ambiente operacional da empresa para o ambiente de tomada de decisões. Podem estar armazenados em bancos de dados de vários modelos, planilhas eletrônicas (*Excel*, por exemplo), arquivos convencionais ou até em documentos e textos. Podem ser inclusive externos à organização, por exemplo, indicadores econômicos.

Área de transporte de dados (*data staging area*): representa a área intermediária de armazenamento entre as fontes de dados e o DW, onde é implementado o processo de ETL (*Extract, Transform and Load* – extração, transformação e carga/atualização).

Os dados são extraídos do ambiente operacional para a área de transporte. São transformados quando necessário, para uniformizar formatos e conteúdos como: dados iguais com nomes diferentes, duplicação de dados, codificações distintas para o mesmo dado etc. A transformação inclui o processo de limpeza, que elimina erros nos dados. Em seguida, são carregados no DW. Existem ferramentas ETL (*back-end*) para implementar esses processos, tanto gratuitas quanto pagas.

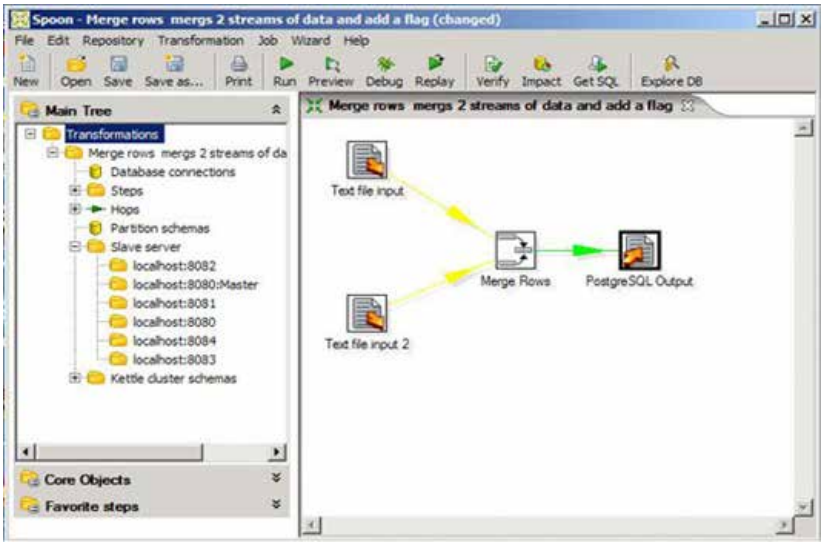


Figura 41 – Ferramenta de ETL Pentaho

Saiba mais

Para conhecer mais sobre a ferramenta *Open Source Pentaho*, recomendamos os *sites*:

<www.pentaho.com>

<kettle.pentaho.com>

<community.pentaho.com/projects/bi_platform>

<*software*livre.org/pentaho>

<pentahobrasil.org>

Servidor de apresentação: máquina física de destino onde os dados do DW são organizados e armazenados para acesso direto pelos usuários finais, geradores de relatório e outras aplicações.

Três sistemas bem distintos são necessários para o funcionamento de um DW: o sistema fonte, a área de transporte de dados e o servidor de apresentação.

Se esse servidor for um banco de dados relacional, as tabelas serão organizadas em esquemas estrela. Como a maioria das grandes implementações tem sido em bancos relacionais, a maior parte das discussões específicas do servidor de apresentação é direcionada nesse sentido.

Processo de negócio: conjunto coerente de atividades de negócio que fazem sentido aos usuários dos nossos DW. Essa definição é propositalmente um pouco vaga. Neste contexto, assumimos que um

processo de negócio é um agrupamento útil de recursos de informação com um tema coerente. Em muitos casos, implementaremos um ou mais *datamarts* para cada processo de negócio.

Armazenamento de dados operacionais – ODS (*Operational Data Store*): esse conceito tem tido muitas definições para ser útil. Essencialmente, pode representar um sistema operacional separado servindo como ponto de integração entre diversos sistemas operacionais ou contendo dados integrados em nível detalhado, que seria na verdade parte do DW.

Olap (*On-line Analytic Processing*): processamento analítico *on-line* é um conjunto de princípios vagamente definidos que fornecem uma estrutura dimensional de suporte à tomada de decisões (consulta e apresentação de textos e dados numéricos). O termo OLAP também é usado para definir um grupo de fornecedores que oferecem produtos de bancos de dados multidimensionais e não relacionais orientados ao suporte à tomada de decisões.

Saiba mais

Rolap: *Relational-Olap* (implementada em banco de dados relacional).

Molap: *Multidimensional-Olap* (implementada sem depender de banco de dados relacional).

Dolap: *Desktop-Olap* (utiliza os recursos da máquina cliente, portanto, é extremamente limitada).

8.7 *Data mart* (DM)

O *data mart* é uma parte do DW. Dentro de uma empresa, enquanto o DW armazena os dados da empresa inteira, o DM armazena os dados de um único departamento, por exemplo.

É um subconjunto lógico de um DW. Tem escopo limitado, para atender a uma determinada área ou processo do negócio. Representa um projeto que pode vir a ser completado e não um empreendimento galático. Um DW é composto pela união de todos os seus DM.

Estabelecemos certos requisitos específicos para cada DM: deve ser representado por um modelo dimensional e dentro de um mesmo DW. Todos esses DM devem ser construídos a partir de dimensões e fatos conformes, que têm o mesmo significado em todas as tabelas.

8.8 Ferramentas de acesso aos dados

Possibilitam a exploração do DW pelo usuário final. Variam de simples ferramentas de consultas *ad hoc* e geradores de relatórios, aplicações para usuários finais que acessem os dados do DW, a ferramentas de análises sofisticadas para Olap e mineração de dados.

Ferramentas de consultas *ad hoc* permitem ao usuário formular suas próprias consultas, manipulando diretamente os dados cadastrados.

EIS – *Executive Information Systems*: sistemas que examinam uma perspectiva ampla dos dados e oferecem informações comparativas e claras para tomada de decisões.

Mineração de dados (*data mining*): técnicas e ferramentas de análise de dados com a função de descobrir e entender tendências, comportamentos, anomalias e outras relações entre os dados.

Olap – *On-line Analytical Processing*: é uma "categoria da tecnologia de *software* que permite que os analistas, gerentes e executivos obtenham, de maneira rápida, consistente e interativa, o acesso a uma variedade de visualizações possíveis da informação" (INMOM *et al.*, 2005).

Key performance: indicador-chave de desempenho. Serve para acompanhar o desempenho de uma métrica. Para isso, necessita de outra métrica para servir de base, por exemplo, meta VS realizado. São exibidos para os usuários na forma de semáforos, conta-giros, termômetros ou figuras que sirvam para chamar bem a atenção, conforme a figura a seguir.

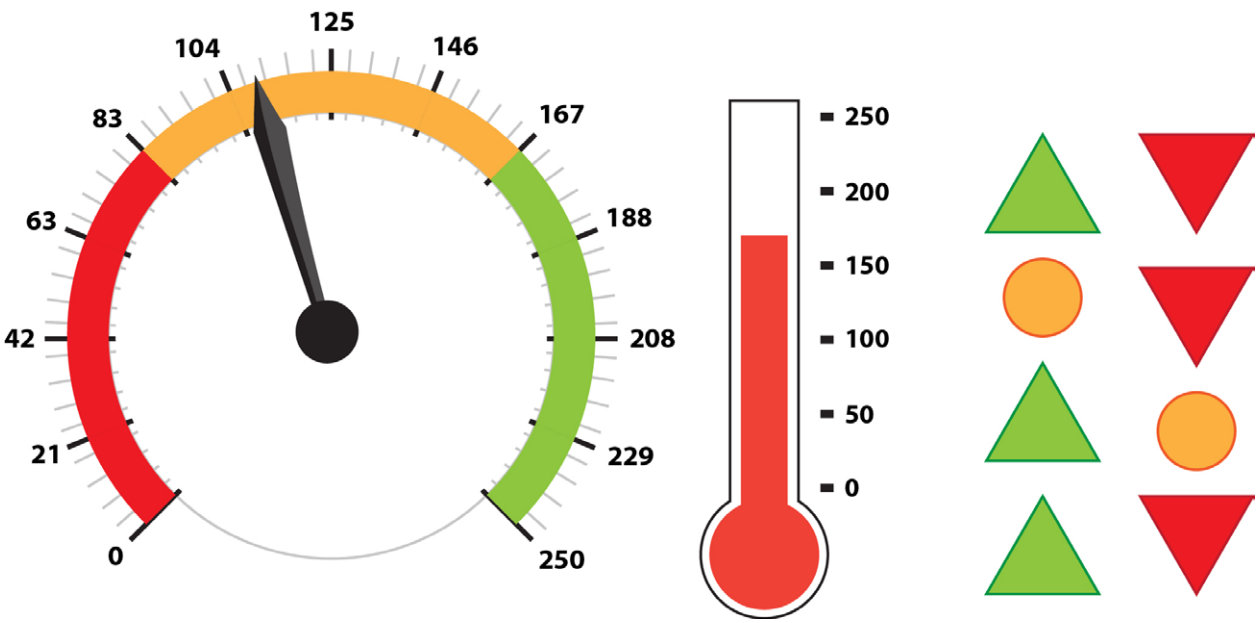


Figura 42

Dashboard é uma página que combina diversas informações a respeito de um mesmo tema. Essas informações são apresentadas de formas diferentes. Em um mesmo *dashboard*, podem coexistir relatórios, gráficos e KPis.

A seguir, um exemplo de *dashboard*.

Figura 43

Esse *dashboard*, para quem gosta de acompanhar a Champions League, o campeonato de futebol europeu, fornece, além de resultados de jogos, classificação de times e várias outras estatisticas sobre os times e seus jogadores.

Resumo

A importância dos sistemas de bancos de dados nas organizações é vista pela crescente valorização dos bancos de dados e dos sistemas gerenciadores de bancos de dados, o que gera consequentes investimentos em técnicas de gerenciamento, monitoramento, *backup* e restauração de dados e em todo o processo que envolve a importância financeira de manter a integridade dos bancos de dados. Um problema muito real refere-se ao gerenciamento de todas as contas bancárias em sistemas de arquivos permanentes de um determinado banco. Esse sistema possui uma série de programas aplicativos necessários para a manipulação por parte dos usuários, que permitem:

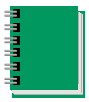
- débito e crédito em outra conta;
- um programa para adicionar uma nova conta;
- fazer pagamentos e depósitos;
- calcular aplicações;
- inserir novas alíquotas.

Esses aplicativos só foram desenvolvidos porque surgiram problemas e necessidades da organização bancária, e isso significa um processo contínuo, pois as aplicações são desenvolvidas conforme vão surgindo as necessidades.

O ambiente de *data warehouses* (DW) integra informações provenientes de uma grande variedade de bancos de dados operacionais em um único banco de dados lógico, que pode ser visto como um repositório central desenvolvido para facilitar o processo de suporte à decisão. Segundo Inmom *et al.* (2005), DW é uma coleção de dados orientados por assuntos, integrados, não voláteis e variáveis com o tempo, para suporte ao processo gerencial de tomada de decisão.

Como se trata de uma implementação estratégica, o DW deve utilizar equipamentos de tecnologia aberta, para não deixar a empresa atrelada a um único fornecedor, isso para não criar uma dependência tanto no caso de implementação tecnológica como na manutenção e suporte.

Um DW corporativo constitui a modalidade mais robusta. São soluções geralmente adotadas por grandes corporações que justifiquem o porte desse tipo de solução (maiores que 100 Gb), dois a três anos para a implementação e investimentos na ordem de seis a sete milhões de dólares. Atualmente, esse custo pode ser reduzido de forma substancial com a adoção de soluções *open source*. A escolha da modalidade de DW deve considerar outros aspectos, como: custo, tempo de desenvolvimento, ferramenta e consultoria.



Exercícios

Questão 1. A Administração de Banco de Dados é um conjunto de atividades responsáveis pela manutenção e gerenciamento de um banco de dados. O profissional que executa essas atividades é chamado de DBA (DataBase Administrator). O DBA é responsável também pela criação de *backups*, que servem para a recuperação de dados, caso ocorram problemas no *hardware* ou nos sistemas SGBDs. Todos os arquivos que contêm os bancos de dados são de responsabilidade do DBA que, além de guardar os arquivos, visa zelar pelas condições dos mesmos, pela segurança dos dados, acessibilidade, desempenho das máquinas e processos e no desenvolvimento de equipe de testes de todo o planejamento de banco de dados. Na prática, busca a melhor maneira de organizar, selecionar e armazenar todos os dados, avaliando não somente a parte técnica, mas as pessoas que irão utilizá-los.

Considerando os conceitos sobre a Administração de Banco de Dados, examine as afirmações a seguir e indique a alternativa **incorreta**:

A) O DBA altera e testa todos os procedimentos antes de abrir acesso dos dados aos seus usuários ou aplicativos, bem como gerenciar o direito de acesso de cada setor e pessoa.

B) O banco de dados deve ser simples e eficaz, para isso é necessário o trabalho de um administrador de dados que saiba projetar, instalar, atualizar, modificar, proteger e consertar erros na plataforma e nos bancos de dados.

C) Na Administração de Banco de Dados há o trabalho de personalização e suporte de todo conteúdo de um determinado banco de dados.

D) Na área comercial, institucional e social há grande demanda por profissionais de Banco de Dados que atuam em sistemas de lojas, catálogos, serviços de comunicação, finanças, educacional e demais áreas.

E) As áreas de TI mais modernas utilizam ferramentas de automação em BD e vêm eliminando a necessidade de pessoas especializadas em Banco de Dados, tal como os profissionais denominados de DBAs.

Resposta correta: alternativa E.

Análise das alternativas

De acordo com os autores e especialistas em TI, nos dias de hoje, é praticamente impossível imaginar um cenário no qual uma aplicação ou sistema de aplicações não precise em ao menos uma de suas etapas consultar, manipular ou armazenar as informações resultantes dos seus processos lógicos. De nada adiantaria todo o poder de processamento dos computadores atuais, as redes de altíssima velocidade e os *softwares* desenvolvidos com tecnologia de ponta se não fosse possível armazenar os dados de forma eficiente e segura. Os dados precisam estar disponíveis, consistentes, íntegros, definidos, confiáveis, compartilhados e em segurança para que as decisões gerenciais sejam ágeis, precisas e oportunas. Dentro desse contexto é que entra o Administrador de Banco de Dados.

A) Alternativa correta.

Justificativa: o DBA é responsável por auxiliar na modelagem de dados, na validação dos modelos de dados e na integração dos modelos com modelos corporativos.

B) Alternativa correta.

Justificativa: o DBA é responsável por definir e manter padrões e normas de segurança, instalar e manter SGBDs e criar e manter instâncias de banco de dados.

C) Alternativa correta.

Justificativa: na Administração de Banco de Dados é necessário que o DBA acompanhe os tempos das consultas e administre os *backups* e espaços utilizados pelas aplicações usuárias do Banco de Dados da empresa.

D) Alternativa correta.

Justificativa: como as bases de dados corporativas estão crescendo intensamente e tornando-se cada vez mais importantes como fontes de informações necessárias à operacionalização das empresas e também como fontes de informações para o processo de tomada de decisão, torna-se necessário o papel do Administrador de Banco de Dados, que deve ser um profissional especialista, capacitado para entender e prestar suporte técnico em cada SGBD utilizado pela organização.

E) Alternativa incorreta.

Justificativa: o DBA é o responsável pela "saúde física" dos dados, isto é, pela manutenção e refinamento dos bancos de dados corporativos. Não existe, na atualidade, sistemas de Banco de Dados que prescidem de um Administrador de Banco de Dados.

Questão 2. Gerentes e executivos das organizações nacionais e internacionais necessitam de recursos computacionais que forneçam subsídios para apoio ao processo de decisão, sobretudo nos níveis tático e estratégico das organizações. A tecnologia *Data Warehouse* (DW) surgiu em meio às dificuldades que as organizações passaram a sentir pela quantidade de dados que suas aplicações estavam gerando e à dificuldade de reunir estes dados de forma organizada e consolidada para uma análise mais eficiente. A idéia do DW é reunir em um único local somente os dados considerados úteis para as tomadas de decisão. Diante desta necessidade, empresas têm utilizado os DWs para auxiliar nos processos corporativos de tomada de decisão.

Considerando os conceitos sobre DWe, examine as afirmações a seguir e indique a alternativa **incorreta**:

A) Conceitualmente, um DW é um conjunto de dados baseado em assuntos, integrado, não volátil, variável em relação ao tempo, e destinado a auxiliar em decisões de negócios.

B) Um DW, quando é orientado a um assunto, aliado ao aspecto de integração permite reunir dados corporativos em um mesmo ambiente de forma a consolidar e apresentar informações sobre um determinado tema.

C) Cada conjunto de dados, ao ser carregado em um DW, fica vinculado a um rótulo temporal que o identifica dentre os demais.

D) Na medida em que um DW vai sendo carregado com as visões sumarizadas dos dados operacionais em um determinado período, podem-se realizar análises de tendências a partir dos dados.

E) As diferenças entre um DW e bases de dados operacionais está no fato de que uma base de dados operacional é um banco de dados clássico que contém informações detalhadas e consolidadas a respeito do negócio em nível transacional.

Resolução desta questão na plataforma.

FIGURAS E ILUSTRAÇÕES

Figura 1

PETERCHEN.JPG. Disponível em: <http://www.csc.lsu.edu/~chen/>. Acesso em: 5 jul. 2012.

Figura 43

DASHBOARD.JSP. Disponível em: <http://www.tablerochampions.com/site/champions/dashboard.jsp>. Acesso em: 15 jul. 2012.

REFERÊNCIAS

Textuais

BEAULIEU, A. *Aprendendo SQL*. 1. ed. São Paulo: Novatec, 2010.

BOOCH, G.; RUMBAUGH, J. JACOBSON, I. *UML – Guia do usuário*. 13. ed. Rio de Janeiro: Elsevier, 2000.

CHU, S. Y. *Banco de dados*: organização, sistemas e administração. São Paulo: Atlas, 1983.

DATE, C. J. *Introdução a sistemas de bancos de dados*. Tradução da 7. ed. americana de Vandenberg Dantas de Souza. Rio de Janeiro: Campus, 2000.

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados*: fundamentos e aplicações. 4. ed. São Paulo: Pearson, 2005.

HERRENO, E. *Balanced scorecard e a gestão estratégica*: uma abordagem prática. Rio de Janeiro: Elsevier, 2005.

HUBBARD, D. W. *Como mensurar qualquer coisa encontrando o valor do que é intangível nos negócios*. Tradução de Ebréia de Castro Alves. Rio de Janeiro: Qualitymark, 2008.

INMON, W. *Building the DataWarehouse*, 4. ed. New Jersey, USA: John Wiley and Sons, 2005.

KIMBALL, R.; CASERTA, J. *The Datawarehouse ETL toolkit*. Boulder Creek, USA: Kimball Group, 2004.

KIMBALL, R.; ROSS, M. *The DataWarehouse Toolkit*. New Jersey, USA: John Wiley and Sons, 2002.

LEME FILHO, T. *BI – Business Intelligence no Excel*. Rio de Janeiro: Novaterra, 2012.

MACHADO, F. N. R. *Projeto de Data Warehouse*. São Paulo: Érica, 2000.

___. *Banco de dados*: projeto e implementação. São Paulo: Érica, 2004.

ROLDÁN, M. C. *Pentaho 3.2 Data Integration Beginner´s Guide*. Birmingham, UK: Packt Publishing, 2010.

SILBERSCHATZHENRY, A.; KORTHS, H. F.; SUDARSHAN, S. *Sistema de banco de dados*. 5. ed. Rio de Janeiro: Campus, 2006.

SIMCSIK, T.; POLLONI, E. G. F. *Tecnologia da informação automatizada*. São Paulo: Berkeley, 2002.

Sites

<http://certificacaobd.com.br>

<http://silasmendes.com/dba>

<http://www.mcdbabrasil.com.br>

<http://www.sybase.com/resources/blogs>

<http://dbaforums.org/oracle>

<www.kimballgroup.com>

<education.oracle.com>

<www.prometric.com>

<www.microsoft.com/learning>

<http://demo.phpmyadmin.net/master-config>

<http://nosql-database.org>

<http://*software*livre.org>

<http://www.*software*livre.gov.br>

Interativa

Informações:
www.sepi.unip.br ou 0800 010 9000