UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

AUDITORIA EM BANCOS DE DADOS RELACIONAIS

MAURICIO THEODÓSIO MATTOS MARQUES

ORIENTADOR: HONÓRIO ASSIS FILHO CRISPIM

MONOGRAFIA DE PÓS-GRADUAÇÃO DO CURSO DE
GESTÃO DA TECNOLOGIA DA INFORMAÇÃO

PUBLICAÇÃO: XXXXX

BRASÍLIA/DF: FEVEREIRO/2002

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

AUDITORIA EM BANCOS DE DADOS RELACIONAIS

MAURÍCIO THEODÓSIO MATTOS MARQUES

Monografia de Pós-graduação submetida ao Departamento de Engenharia Elétrica da Faculdade de Tecnologia da Universidade de Brasília, como parte dos requisitos necessários para obtenção do Certificado de Conclusão do Curso de Gestão da Tecnologia da Informação.

Aprovada por:

_____

Honório Assis Filho Crispim, Msc ITA

_____

Rafael Timóteo de Souza, Prof. Dr. UnB

Data: Brasília/DF, 28 de Fevereiro de 2002.

FICHA CATALOGRÁFICA

MARQUES, MAURICIO THEODÓSIO MATTOS

Auditoria em Banco de Dados Relacionais.

x, 106p., 297 mm (DEE/FT/UnB, Pós-Graduação, Gestão de Tecnologia da Informação, 2001).

Monografia de Pós-Graduação – Universidade de Brasília. Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. Banco de Dados Relacionais 2. Auditoria

3. Segurança da Informação 4. Tecnologia da Informação

I. DEE/FT/UnB II. Título (série)

DEDICATÓRIA

Este trabalho é dedicado a minha esposa e a meus filhos. Para Vânia, Luis Mauricio e Renan, com amor e gratidão, por reconhecerem e compreenderem a importância dele, para mim, principalmente nos momentos em que, por força da dedicação, era-nos furtado o convívio familiar.

AGRADECIMENTOS

Gostaria de agradecer a algumas pessoas que foram fundamentais na realização deste trabalho. Primeiramente, a minha esposa e a meus filhos. Eles sempre estiveram comigo, dando-me aquela força indispensável, ao longo de todo o curso, durante o ano de 2001, muitas vezes, até, privando-se da companhia e dos prazeres familiares.

Ao meu orientador, Prof. Crispim, pelo seu interesse e pelas correções de rumo, sempre imprescindíveis em trabalhos dessa natureza.

Gostaria de agradecer, também àqueles que desempenharam um papel ímpar na feitura desta monografia: a cada diretor da MÚTUA – Caixa de Assistência dos Profissionais dos CREAs, Eng. Henrique Luduvice, Arq. Osni Schroeder, Eng. Luis Abílio, Eng. Ainabil Machado e Eng. Carlos Vanolli, que me apoiaram na participação deste curso; aos companheiros de trabalho da MÚTUA: o Analista de Sistemas Fabiano Scardua pelas sugestões e bate-papos sobre o tema; o Analista de Sistemas Daniel Skeff, pela formatação e editoração do texto e a Técnica Administrativa Sandra Marins, pela dedicação e presteza na digitação do mesmo.

Finalmente, gostaria de agradecer a minha querida sogra, Profª Elvira Montenegro e a minha cunhada, Bruna Montenegro, respectivamente, pelas correções semânticas e sintáticas, além da tradução do resumo desta monografia, sem as quais não conseguiria expor, com clareza, as idéias da presente proposta.

A todos, o meu eterno obrigado.

RESUMO

Esta monografia tem por objetivo abordar aspectos relevantes da segurança da informação nas organizações, com enfoque em auditoria em bancos de dados relacionais, dando uma breve introdução a definições e conceitos básicos sobre o tema, além de apresentar uma metodologia de auditoria composta de 5 etapas:

- Análise do modelo de dados;
- Identificação dos objetos e dos níveis de auditoria;
- Criação das estruturas de auditoria;
- Monitoramento;
- Auditoria.

Baseado nessas etapas, é proposto um modelo de Sistema Gestor de Auditoria (SAUDIT), através da especificação dos seus módulos e funções, a ser implementado em várias plataformas de SGBDs relacionais, que possuam recursos de triggers.

ABSTRACT

This monographic has the objective to board the main characteristics of the information security in the organizations, with a focus in the auditing relational databases, giving a brief introduction to the definitions and basic concepts about the theme, beyond to present an auditing methodology compound of 5 stages:

- Analysis of data model;
- Identification of the objects and auditing levels;
- Creation of the auditing structures;
- Monitoring;
- Auditing.

Based in this stages, it is proposed a Management System Auditing (SAUDIT), across the specifications of its modules and functions to be implemented in the several platforms of the relational SGBDs, witch has triggers means.

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

O conceito de segurança, bem como o seu tratamento, certamente tem acompanhado a humanidade desde a sua criação. Se pensarmos a respeito, verificaremos que, todos os dias, milhares de decisões e atitudes são tomadas, no sentido de preservá-la.

A noção de segurança, segundo [ZIN 97], dá margem a interpretações muito amplas:

- Segurança pode ser a garantia de integridade física a uma pessoa ou grupo de pessoas, como no caso da segurança pública;

- Segurança pode ser a garantia de integridade financeira (que pode resultar em integridade física) a uma pessoa ou grupo de pessoas como, por exemplo, um seguro de vida ou contra roubos;

- Segurança pode ser a garantia de integridade de objetos (concretos ou abstratos) que são importantes para uma pessoa ou um grupo de pessoas como, por exemplo, a proteção de um bem ou de uma informação.

Poderíamos ainda elencar uma série de outros exemplos sobre segurança, mas acreditamos que os três acima são suficientes para a compreensão da importância do tema.

O presente trabalho abordará aspectos relativos ao último exemplo, particularmente ao que se refere à segurança da informação.

Recente pesquisa do Computer Security Institute (CSI), San Francisco-CA, demonstrou o quanto os incidentes de segurança têm crescido, nos últimos anos. Com grande freqüência nos deparamos com notícias de falhas de segurança na Internet, relacionadas ao roubo de informações e sabotagens em sites. Não menos graves são os crimes digitais, nem sempre divulgados, mas causadores de grandes prejuízos às instituições afetadas. Tais crimes têm o intuito de furtar ou adulterar

informações estratégicas pertencentes às organizações e, muitas vezes, são praticadas por concorrentes de mercado. Existem também os danos causados por funcionários, devido às falhas no controle de acesso às informações, resultando na perda ou vazamento de informações importantes. Pesquisas têm apontado que 80% dos casos de ruptura de segurança de uma empresa ocorrem através dos seus funcionários.

Entretanto, grande parte dos gestores de negócios continuam atribuindo à área de Tecnologia da Informação a total responsabilidade pela condução das ações, visando a proteger a informação, tida e havida como o maior patrimônio das empresas. De acordo com [MOR 01], cabe à área de TI o "como fazer", para proteger a informação, enquanto que "o que e quando fazer" deve ser emanado da alta gerência e permeado por toda a organização. Nesse sentido, a tarefa de prover segurança às informações de uma organização torna-se mais difícil, não só pela existência das ameaças já citadas, mas também pelo fato da segurança não poder restringir-se às redes e sistemas de computadores. A informação, como se sabe, trafega por diferentes meios, e o elemento humano deve ser considerado como um fator fundamental para o sucesso da aplicação de medidas de segurança. A exemplo de outros países, aqui no Brasil a preocupação com a segurança da informação tem sido tão evidente que hoje existe legislação e normas específicas sobre o assunto.

No mundo físico, em papel, que estamos deixando, a segurança geralmente se resume a mecanismos também físicos de proteção: cofres, áreas protegidas, vigilância por pessoas especializadas etc. No mundo virtual de hoje, dos negócios eletrônicos, cresce a preocupação com a segurança dos ambientes em que eles são realizados. Agora, não é suficiente a proteção física do ambiente, ainda que extremamente necessária. São requeridos outros cuidados, para a preservação da informação que reside em nossos sistemas e trafega em nossos meios de comunicação de dados. Essa nova segurança requer planejamento, treinamento e avaliações cuidadosas.

Assim, dependendo do tipo de informação que estamos gerindo, temos que nos preocupar também com o nível de acesso a elas, fazendo com que somente

pessoas autorizadas possam manipulá-las. Esse mesmo tipo de preocupação deve estar voltado para o fato de que a segurança também deve ser flexível, no sentido de sempre se adaptar às informações e às pessoas que a utilizam.

Se levarmos em consideração as inúmeras ameaças, bem como as suas imprevisibilidades, às quais a segurança da informação é continuamente submetida, segundo [MOR 01], chegaremos à conclusão de que não existe um método ou dispositivo de segurança perfeito, seja qual for a abordagem aplicada. A realidade tem apontado que a prevenção é responsável por 1/3 das soluções de segurança, enquanto que os 2/3 restantes são constituídos pela detecção e resposta [BRI 01].

Além da falta de previsão quanto às ameaças, a insuficiência da aplicação de medidas de segurança das informações decorre, na maioria das vezes, do tratamento parcial que lhe é dado. Adotar simplesmente algumas soluções de hardware e software específicos e acreditar nisso como uma solução eficiente de segurança é um erro comum. É preciso, antes de tudo, realizar uma análise de riscos, envolvendo uma detalhada análise de vulnerabilidades, definindo ações e atribuindo valores aos eventuais impactos, no caso de concretizadas as ameaças existentes, em face de controles inadequados ou mesmo inexistentes.

Feito isso, devemos acenar com soluções envolvendo a estruturação de uma função denominada Segurança da Informação, responsável por implementar as ações propostas, bem como desenvolver e implementar uma Política de Segurança da Informação Corporativa abrangente e que corresponda às expectativas, quanto à continuidade dos negócios da organização.

De acordo com [CAM 97], essa Política Corporativa deve nortear a elaboração de padrões e diretrizes, operacionalizadas sob a forma de procedimentos. Esses procedimentos devem, então, ser detalhados em termos de hardware e software, permitindo a condução dos negócios da organização, no seu dia-a-dia, sem impactá-los, ou seja, controle que prejudica o negócio não é controle adequado. Por exemplo, às vezes, ainda na fase

se medidas para monitorar as ameaças decorrentes. Finalmente, deve-se implementar tal arquitetura de segurança com o apoio de toda a organização, inclusive da alta gerência, através de programas de conscientização e treinamento. Dessa forma, é possível demonstrar que a Segurança da Informação é uma área onde se alocam recursos, a fim de se obterem resultados, incluindo aí a economia, ao se evitar redundância de atividades de suporte, diminuição de demandas ao help desk e, sobretudo, ao minimizar o risco de manipulação indesejada de informações críticas da organização.

Diante de todas essas informações, desnecessário se faz registrar a abrangência e importância do tema Segurança da Informação. O escopo do presente trabalho se insere no assunto, na medida em que trata de uma fase da Gestão da Política de Segurança das Informações Corporativas de uma Organização, que é a manutenção da informação, propriamente dita. Em qualquer sistema de informações que demande segurança, algumas premissas básicas devem ser tomadas:

- Todos os usuários devem ser identificáveis;
- Suas ações devem ser autorizadas e monitoradas;
- Os dados manipulados devem ser recuperáveis e passíveis de auditoria.

É nesse contexto que a auditoria aparece. Apesar da auditoria variar, necessariamente, de caso para caso, existem elementos comuns a qualquer processo de auditoria. A proposta deste trabalho é a de, justamente, identificar etapas comuns da auditoria e definir uma metodologia que utilize essas etapas, no apoio à construção dos seus processos. Inicialmente, abordamos algumas definições e conceitos básicos sobre o assunto, para em seguida descrever o modelo de auditoria em bancos de dados proposto, com um estudo de caso, em aplicação desenvolvida para o SGBD Microsoft SQL Server. Finalmente, apresentamos uma proposta mais abrangente do que deve ser um sistema de auditoria (SAUDIT) a ser implementado em várias plataformas de SGBDs relacionais.

2 OBJETIVOS DO TRABALHO

Conforme ilustrado no item anterior, o presente trabalho se insere em uma fase da Gestão da Política de Segurança das Informações Corporativas de uma Organização, no que diz respeito à manutenção da informação. Dentre as medidas de segurança utilizadas em um ambiente computacional, para viabilização desta manutenção, [MOR 01] cita:

- **Medidas Preventivas:** possui como foco a prevenção da ocorrência de incidentes de segurança. Neste caso, todos os esforços estão baseados na precaução. No caso de segurança de informações, podemos citar como exemplo a atribuição de contas e senhas, bem como de direitos de acesso e níveis de acesso para quem vai utilizar ou manipular algumas informações;

- **Medidas Detectivas:** possui como foco a necessidade de se obterem auditagem, monitoramento e detecção da ocorrência de incidentes de segurança. Como exemplo dessas medidas, podemos citar os logs de bancos de dados;

- **Medidas Corretivas:** possui como foco os mecanismos utilizados para a continuidade das operações, quando da ocorrência de incidentes de segurança. Como exemplo dessas medidas, podemos citar a adoção de planos de contingência e planos de continuidade de negócios.

Este trabalho tem, como objetivo principal, apresentar uma ferramenta a ser utilizada em ambientes que utilizam banco de dados, na esteira das Medidas Detectivas, como uma das estratégias para manutenção das informações em uma organização. Trata-se de uma proposta de monitoramento de atualizações de registros, em bancos de dados relacionais, de uma forma genérica e independente de desenvolvimentos e/ou manutenções em aplicações já existentes, através da implementação de triggers de inclusão, alteração e exclusão em suas correspondentes tabelas. A idéia é muito simples: por meio de um processo automático, toda vez que houver uma alteração em qualquer tabela de um banco de

dados, a mesma será registrada numa tabela criada exclusivamente para essa finalidade (tabela de log de auditoria), registrando-se, ainda, além da informação alterada, os dados ambientais de tal atualização (responsável, local, data e hora da ocorrência).

Apresentaremos, ainda, uma proposta de especificação técnica para implementação de um Sistema de Auditoria em Bancos de Dados Relacionais (SAUDIT) e, a título de ilustração, uma implementação física de tal proposta, desenvolvida para o Sistema Gerenciador de Banco de Dados (SGBD) Microsoft SQL Server, a qual poderá ser perfeitamente implementável em outros SGBDs relacionais, desde que possuam recursos de triggers.

3. DEFINIÇÕES E CONCEITOS BÁSICOS

Para um melhor entendimento sobre os assuntos a serem tratados neste trabalho, é necessário um conhecimento mínimo das terminologias e conceitos básicos sobre o tema abordado.

3.1 SEGURANÇA DA INFORMAÇÃO

3.1.1 O que é segurança da informação

Sendo a informação um ativo importante nas empresas, se não o maior, para a realização de bons negócios, nos dias atuais, a sua preservação ganha um papel de destaque no mundo em que vivemos, pois em muitas situações pode significar a sua própria sobrevivência. Independentemente da forma como a informação está armazenada, ela deve ser protegida adequadamente. Segundo [ABN 01], a Segurança da Informação protege a informação de diversos tipos de ameaças, para garantir a continuidade dos negócios, minimizar os danos e maximizar o retorno dos investimentos e as oportunidades de novos negócios. Fundamentalmente, a segurança da informação deve buscar a preservação de 03 itens, a saber:

- **Confidencialidade:** garantia de que a informação é acessível somente a pessoas autorizadas, ou seja, manutenção do sigilo, do segredo ou da privacidade das informações, evitando que pessoas não autorizadas tenham acesso às mesmas;

- **Integridade:** salvaguarda da exatidão da informação, ou seja, qualquer tipo de alteração da informação sem a devida autorização do autor da mesma;

- **Disponibilidade:** garantia de que os usuários autorizados obtenham acesso à informação e aos ativos correspondentes, sempre que necessário, ou seja, a informação deve estar disponível para a pessoa certa no momento em que ela precisar.

Entretanto, não basta apenas um conjunto de ferramentas de softwares e implementá-los, para obtermos a tão desejada Segurança da Informação. É preciso um conjunto de controles dentro de um contexto de um Plano de Segurança, em consonância com os níveis estratégico, tático e operacional da empresa.

3.1.2 Porque a segurança da informação é necessária

Segurança da Informação é a base para proporcionar às empresas a liberdade de criação de novas oportunidades de negócio. Hoje, como sabemos, os negócios estão cada vez mais dependentes da tecnologia, onde são continuamente colocados à prova por diversos tipos de ameaças à segurança da informação, incluindo aí fraudes eletrônicas, espionagem, sabotagem, vandalismo, fogo, inundação, entre outras. Com o advento da Internet, como exemplo, assistimos a muitos casos de invasões de sistemas, novos e sofisticados vírus inundando a rede, e a clonagem de cartões de crédito se tornando uma praga mundial. A Interconexão de redes públicas e privadas e o compartilhamento de recursos de informação, aliados ao fato de que muitos sistemas de informação não foram projetados para serem seguros, conduzem-nos a uma conclusão direta de que todo cuidado é pouco na era digital, onde a informação é o maior bem a ser preservado. Segurança, porém, não é só uma questão técnica, é também de política e educação empresarial. É preciso o envolvimento de toda a organização nesse paradigma dos tempos atuais.

3.1.3 Como estabelecer requisitos de segurança

A norma ISO/IEC 17799 [ABN 01] preconiza que, para o estabelecimento dos requisitos de segurança em uma organização, existem 03 (três) pontos básicos a serem analisados.

- **Avaliação de Riscos:** onde são identificadas as ameaças aos ativos, as vulnerabilidades, a probabilidade de ocorrência e os seus impactos potenciais estimados;

- **Legislação Pertinente:** onde são analisados os estatutos, regimentos e cláusulas contratuais e que a organização, seus parceiros, contratados e prestadores de serviço estão submetidos;

- **Princípios:** Onde são analisados os objetivos e requisitos para o processamento da informação que uma organização tem que desenvolver, para apoiar suas operações.

3.1.4 Avaliando riscos de segurança

De acordo com [MOR 01], a análise de risco consiste em um processo de identificação e avaliação dos fatores de risco presentes, de forma antecipada, no ambiente organizacional, possibilitando uma visão do impacto negativo causado aos negócios. Assim, é possível determinar as prioridades de ação, em função do risco identificado, para se atingir o nível de segurança desejado. Possibilita, também, avaliar-se o tipo e o volume dos investimentos necessários, de forma antecipada, aos impactos causados pela perda ou indisponibilidades dos recursos fundamentais para o negócio. Em que pese a inexatidão deste processo, face à impossibilidade de se preverem com exatidão as possibilidades de ocorrências, sem um processo deste tipo, não é possível identificar a origem das vulnerabilidades, nem visualizar os seus riscos. As medidas de segurança não podem assegurar 100% de proteção, assim sendo, deve-se sempre avaliar a relação custo/benefício de todas elas. É preciso se achar o nível de risco de convivência aceitável. De qualquer forma, esse processo deve, no mínimo, fornecer as seguintes informações:

- Pontos vulneráveis do ambiente;

- Ameaças potenciais ao ambiente;

- Incidentes de segurança causados pela ação de cada ameaça;

- Impacto negativo no negócio, a partir da ocorrência dos incidentes prováveis de segurança;

- Riscos para o negócio, a partir de cada incidente de segurança;

- Medidas de proteção adequadas, para impedir ou diminuir o impacto de cada incidente.

É necessário, também, realizar análises periódicas dos riscos de segurança e dos controles implementados, para:

- Considerar as mudanças nos requisitos dos negócios e suas prioridades;

- Considerar novas ameaças e vulnerabilidades;

- Confirmar que os controles permanecem eficientes e adequados.

As avaliações de risco são sempre realizadas primeiro em nível mais geral, como forma de priorizar recursos em áreas de alto risco e, então, em um nível mais detalhado, para solucionar riscos específicos.

3.1.5 Seleção de controles

Uma vez identificados os requisitos de segurança, convém que os controles sejam selecionados e implementados, para assegurar que os riscos sejam reduzidos a um nível aceitável. Convém que os controles sejam selecionados com base nos custos de implementação, em relação aos riscos que serão reduzidos e às perdas potenciais, se as falhas na segurança ocorrerem. Convém também considerar fatores não financeiros como, por exemplo, queda da imagem da organização. Abaixo, relacionamos alguns controles que julgamos essenciais para a maioria das organizações:

- Proteção de dados e privacidade de informações pessoais;

- Salvaguarda de registros organizacionais;

- Direitos de propriedade intelectual;
- Documento da Política de Segurança da Informação;
- Definição de responsabilidades na segurança da Informação;
- Educação e treinamento em segurança da Informação;
- Relatório dos incidentes de segurança;
- Gestão da continuidade do negócio.

3.1.6 Fatores críticos de sucesso

Em que pese não existir um modelo ideal de segurança preestabelecido, pois os mesmos dependem do tipo de negócio, estratégia, ambiente operacional, cultura da empresa, entre outros dados, a experiência tem mostrado que os seguintes fatores são geralmente críticos, para o sucesso da implementação da segurança da informação de uma organização:

- Política de Segurança, objetivos e atividades, que reflitam os objetivos do negócio;
- Um enfoque na implementação da segurança, que seja consistente com a cultura organizacional;
- Comprometimento e apoio visível da direção;
- Um bom entendimento dos requisitos de segurança, avaliação de risco e gerenciamento de risco;
- Divulgação eficiente da segurança para todos os gestores e funcionários;

- Distribuição das diretrizes sobre as normas e política de segurança da Informação para todos os funcionários e fornecedores;

- Proporcionar educação e treinamento adequados;

- Um abrangente e balanceado sistema de medição, que é usado para avaliar o desempenho da gestão da segurança da informação e obtenção de sugestões para a sua melhoria.

3.2 FRAUDES

Os imensos sistemas de informações interligados, hoje em dia, principalmente através da Internet, são território relativamente fértil ao cometimento de fraudes. Enquanto os agentes da lei procuram combater esse novo tipo de crime, eles sempre parecem estar um passo atrás do seu autor. A tecnologia de programas legais avança constantemente, mas o mesmo acontece com ferramentas de espionagem e invasão. Parte do que torna a fraude com computador tão singular e difícil de combater deve-se à sua dupla natureza: ela pode ser a ferramenta usada para cometer a fraude, como também pode ser o objeto da mesma.

3.2.1 O computador como ferramenta para o cometimento de fraudes

Segundo [MOR 01], a fraude pode ser entendida como qualquer tipo de exploração em um sistema, com o objetivo de enganar terceiros, através de alguma forma ou recurso. De um modo geral, são necessárias duas habilidades para se cometer a maioria das fraudes em computador. Inicialmente, o fraudador precisa acessar o sistema, e isso normalmente exige o conhecimento prévio de uma identificação e de uma senha de acesso. Em seguida, ele tem de saber como manipular o sistema, para produzir o resultado desejado. Apesar de ser possível, na maioria dos casos, detectar as fraudes, o sistema computacional da empresa precisa estar preparado a ponto de poder rastrear todas as operações efetuadas pelo fraudador. A seguir, elencamos alguns exemplos de fraudes, citados por [JUN 01], possíveis de ocorrer, tanto em redes corporativas quanto na Internet.

- **Redes Corporativas**
  - Roubo de senhas;
  - Uso indevido de conta e senha com alto nível de acesso;
  - Decodificação, seguida de alteração de programas executáveis;
  - Acesso e alteração de dados direto em banco de dados.

- **Internet**
  - **Roubo de Identidade:** segundo o FBI, é o crime de colarinho branco que mais cresce nos EUA. Em torno de 500 mil americanos têm suas identidades usurpadas, por ano. Quatro companhias americanas de seguros já oferecem apólices para cobrir furto de identidade e os seus prejuízos conseqüentes;

  - **Sites estão de olho em nossa navegação:** "surfar" na Internet transmite uma sensação de anonimato, mas a maioria dos sites usa cookies, para coletar dados de sua navegação. Muitos não sabem, mas os bowsers incluem nomes, endereços eletrônicos e outros dados que podem ser capturados e arquivados. O próprio protocolo TCP/IP pode "dedurar" o usuário, pois cada micro, ao se plugar na rede, tem um endereço IP (Internet Protocol) próprio;

  - **Sites de e-commerce repassam dados de clientes:** as informações pessoais podem ser facilmente vendidas ou furtadas, principalmente em sites de e-commerce, que coletam grande quantidade de dados. Lojistas virtuais coletam informações de compradores e já se tornou prática comum, na grande rede, a venda desses dados;

  - **Casos de falsos sites de comércio eletrônico:** o FBI recentemente estourou uma rede russa de fraude, na qual os hackers estavam envolvidos no que foi cunhado de "Website spoofing" (clonar um site e se beneficiar de seus serviços e usuários). É fácil conseguir nomes de domínios praticamente iguais a outros, já existentes. Um exemplo disso

foi a recente clonagem do site do Bank of América, um dos maiores dos EUA, que teve seu domínio imitado com a simples supressão do ponto, após o www (wwwbankofamerica.com).

- **Espionagem é outra ameaça do mundo virtual:** estranhos podem usar seu PC, para espioná-lo. Eles penetram no micro e olham tudo dentro dele, se suas defesas estiverem desativadas. Outro método utilizado é enganar o PC e fazê-lo baixar programas de bisbilhotagem. Daí o nome "Cavalo de Tróia", programa que se esconde dentro de outro, aparentemente inofensivo. A maioria dos vírus é criada para danificar micros, mas alguns são desenvolvidos para furtar informações. Um exemplo é o Back Oriffice, criado pelo grupo Cult of the Dead Cow (culto da vaca morta), que invade e domina o micro, espiona senhas e até a seqüência de teclas pressionadas.

3.2.2 O computador como objeto de fraudes

A cada vez que um acesso a um sistema é obtido ilegalmente, que dados ou equipamentos são roubados ou danificados intencionalmente, ou mesmo que um software seja copiado ilegalmente, o computador se torna objeto de uma fraude. A seguir, citamos alguns exemplos encontrados em [STA 98].

- **Acesso e uso ilegal:** desde o surgimento da Tecnologia da Informação, os computadores têm sido vítimas dos hackers. Um hacker é uma pessoa aficionada pela tecnologia de computadores, que gasta seu tempo aprendendo e usando sistemas computacionais. Ao contrário do cracker (muito confundido com hacker), o hacker, depois de entrar em um sistema, não altera a informação, pois isto vai de encontro aos seus princípios. Existem ainda os Phreakers, que são os harckers da telefonia. De uma forma ou de outra, com a crescente utilização das redes, os riscos de problemas, causados por esses indivíduos, têm aumentado. Os principais problemas são o acesso não-autorizado a sistemas computacionais e redes, introduzindo e desativando sistemas de comunicações, penetrando

em redes de voz e de fax, obtendo acesso remoto a PBX e perpetrando fraudes em correios eletrônicos de voz. Como alternativas de soluções estão as senhas de redes, a criação de trilhas de auditoria e procedimentos de auditorias de redes, a limitação do acesso físico às redes, o treinamento de segurança e uma variedade de dispositivos de codificação para fax, linhas telefônicas privadas e outros sistemas de redes e comunicações. Apesar desses esforços, muitos especialistas ainda crêem que os procedimentos de prevenção contra essa natureza de fraude ainda são inadequados. Para se ter uma idéia, a Riptech (http://www.riptech.com), empresa americana especializada em segurança da informação, divulgou recentemente o Internet Security Threat Report, que é um estudo que mostra que a atividade hacker cresceu bastante nos últimos meses. Entre julho e dezembro de 2001, o índice de ataques por empresa aumentou quase 80%. Os Estados Unidos lideram o ranking dos países de onde partem mais ataques cibernéticos. Em segundo lugar ficou a Coréia do Sul;

- **Alteração e destruição de dados:** o uso intencional de programas ilegais e destrutivos, para adulterar dados, é um ato tão criminoso quanto a destruição de bens tangíveis. Os programas mais comuns desse tipo são os vírus e os vermes. Um vírus é um programa que se oculta dentro de outro programa, já um verme age independentemente, instalando cópias dele mesmo em outros sistemas, destruindo programas e interrompendo o funcionamento de redes e de sistemas computacionais. Alguns vírus e vermes atacam computadores pessoais, enquanto outros atacam sistemas de redes e de cliente-servidor. Um vírus ou verme que ataca sistemas de redes e de cliente-servidor normalmente tem efeitos mais graves, pois ele pode afetar centenas ou milhares de computadores pessoais e outros equipamentos ligados à rede. Na verdade, eles não têm a ver diretamente com a Internet. O crescimento do número de sites do mundo inteiro, porém, tem contribuído e muito para a disseminação dos vírus e vermes, pois facilitou enormemente a troca de arquivos entre computadores, o que antes era feito basicamente por meio de disquetes e BBS (Bulletim Board

System), que foi um sistema de comunicação digital através de linha telefônica, em que um banco de dados central era acessado, via modem, por vários computadores, permitindo a troca de arquivos e mensagens. Hoje em dia, um vírus que se encontra no Japão, por exemplo, pode rapidamente infectar um microcomputador ou uma rede aqui, no Brasil, bastando, para tal, um simples e-mail com um arquivo anexado, contaminado. A seguir, listamos as formas mais comuns usadas, atualmente, para se infectar um sistema com um vírus:

- Anexos de mensagens recebidas via e-mail;
- Arquivos infectados, armazenados em servidores FTP (File Transfer Protocol);
- Arquivos recebidos via ICQ (I Seek You);
- Disquetes;
- BBS (Bulletim Board System);
- Newsgroups.

Um estudo da Sophos, empresa que comercializa softwares, revela que foram identificadas mais de 11.000 novas espécies de vírus, em 2001. A maior parte desses vírus infecta as máquinas através de e-mails. Para se ter uma idéia de como os equipamentos podem estar vulneráveis, foram detectadas mais de 1,6 milhão de mensagens contaminadas, circulando na web. Esse número é da Messagelabs, que monitora o fluxo para empresas, em vários países, e significa que circula um e-mail contaminado a cada 18 segundos. De cada 370 mensagens verificadas pela Messagelabs, pelo menos uma estava infectada com algum tipo de vírus. Em 2000, a proporção era de um para 700 mensagens.

Existem basicamente três tipos de vírus:

- **Vírus de Programas:** os vírus deste tipo atacam os arquivos executáveis (extensão .com, .exe etc). Quando o programa é executado, o vírus infecta o computador. Como estes tipos de vírus normalmente se ocultam nos arquivos executáveis, os mesmos podem ser detectados pelo exame do comprimento ou tamanho do arquivo supostamente infectado;

- **Vírus de Macro:** os vírus deste tipo atacam os arquivos .doc ou .xls, que são arquivos feitos em Word e Excel, pois carregam junto de si macros que são lidos por estes programas;

- **Vírus de Sistema:** os vírus deste tipo se escondem em qualquer disquete e contaminam justamente o setor de "boot" do disco, comprometendo o funcionamento do sistema operacional. Todas as vezes em que o microcomputador infectado pelo vírus é ligado, este fica residente na memória, contaminando outros arquivos;

- **Cavalos de Tróia:** os "Cavalos de Tróia" não são considerados vírus nativos, pois não se duplicam e tampouco contaminam outros programas. Eles são programas autônomos, muitas vezes camuflados em aplicativos do tipo jogos. Podem ficar inativos no computador por muito tempo e só trabalham quando uma determinada data chegar. A diferença de um vírus para um "Cavalo de Tróia" é que o vírus se auto-reproduz, e o "Cavalo de Tróia" não, além de precisar ser executado, para se instalar. Ao serem instalados em um computador, criam uma maneira de alguém entrar no computador conectado à Internet, sem que o dono perceba.

  - **Roubo de dados e de Informações:** as pessoas que acessam sistemas ilegalmente, muitas vezes fazem isso para roubar dados e informações. Embora os casos de roubo de dados mais divulgados envolvam pessoas de fora da organização lesada, os dados são geralmente roubados por pessoas da organização, que conhecem os sistemas corporativos da

empresa. Com a crescente miniaturização e densidade de armazenamento dos discos flexíveis, está muito mais fácil, hoje, para os funcionários descontentes, roubarem software e dados corporativos do que antes;

- **Roubo de Equipamentos:** a miniaturização dos computadores teve um efeito semelhante nos roubos de equipamentos computacionais. Os computadores portáteis (e os dados e informações neles armazenados) são alvos fáceis para os ladrões;

- **Pirataria de Software: a** pirataria de software é uma prática ilícita, caracterizada pela reprodução e/ou uso indevido de programas de computadores (software) legalmente protegidos, sem a autorização expressa do titular da obra e, conseqüentemente, sem a devida licença de uso.

3.2.3 A prevenção de fraudes relacionadas a computadores

Devido ao crescente uso atual de computadores, está sendo maior a ênfase à prevenção e detecção de fraudes relacionadas a computadores. Qualquer empresa que use um sistema de informação deve se envolver ativamente com a prevenção e a detecção de fraudes. Uma das melhores maneiras de se envolver é projetar sistemas de segurança como parte do seu sistema de informação, antes que o mesmo seja colocado em regime de produção. Outra estratégia é atribuir as diversas tarefas de processamento a várias pessoas. A separação de tarefas e responsabilidades auxilia a prevenir a fraude, porque mais pessoas estarão envolvidas e poderão atuar como verificadoras umas das outras. Em linhas gerais, antes de se implementarem controles, as organizações devem considerar os tipos de fraudes relacionadas ao computador que podem ocorrer, as conseqüências dessas fraudes e o custo e a complexidade dos controles necessários, para tentar evitá-los.

3.3 PRIVACIDADE

A questão da privacidade trata da coleta de dados de uma determinada pessoa e o uso ou mau uso desses dados por terceiros, sem a devida permissão. Dados sobre cada um de nós estão constantemente sendo coletados e armazenados. Esses dados freqüentemente são distribuídos por redes facilmente acessíveis, sem o nosso conhecimento ou autorização. Poucas pessoas têm ciência de que, uma vez conectado à internet, informações podem não se tornar tão confidenciais quanto se imagina ou gostaria que estivessem. Por esta razão, muitas empresas estão proibindo o armazenamento de informações confidenciais e de extremo valor, em áreas públicas acessíveis pela internet.

Os defensores da privacidade lutam, nos EUA, para aprovar legislação federal que obrigue os sites a permitirem aos internautas exercer o direito de exclusão de seus dados nas transações de e-commerce, mas estamos longe de um denominador comum. Ainda falta muito para a tecnologia alcançar um controle completo sobre as informações que trafegam na Internet, de forma a oferecer segurança a seus usuários.

3.3.1 O direito básico à privacidade

Sendo a privacidade um direito constitucional, com o advento das redes de computadores, ela tornou-se um desafio. Hoje são produzidos e utilizados mais dados e informações do que nunca. Ficam, então, algumas indagações, a saber: quem é o dono dessas informações e desse conhecimento? Se uma determinada organização investe em recursos para obter dados sobre uma pessoa, ela é a proprietária desses dados? E, como tal, pode utilizá-los da maneira que lhe convier?

A legislação governamental responde a essas perguntas, até certo ponto, pelas repartições de governo, mas elas permanecem sem resposta, no caso de organizações privadas.

3.3.2 Questões de privacidade

A questão da privacidade é um assunto que assume uma importância especial na medida em que, como já mencionado, os dados pessoais podem ser coletados, armazenados e utilizados muitas vezes sem o conhecimento e/ou autorização do seu titular. Se imaginarmos que, ao longo de nossa vida, registramos dados a respeito de fatos e acontecimentos vivenciados (nascimento, registros escolares, empregos, identidade, CPF, serviço militar, imposto de renda, IPVA, IPTU, seguros, cartões de crédito, empréstimos, compras diversas, exames médicos etc), teremos a idéia da dimensão do volume de informações registradas, de cunho pessoal, que de alguma forma estão armazenadas em alguns lugares e em arquivos de computador.

Os governos federal, estadual e municipal, sem dúvida nenhuma, são os maiores coletores de dados. Pela combinação de informações sobre pessoas, obtidas desses diversos registros, obtém-se o perfil de um potencial cliente, para determinada empresa. É aí que a informação ganha um valor comercial. Até que ponto o comércio dessas informações infringe a privacidade? Violações de privacidade mais evidentes são os casos em que pessoas que se conectam a serviços de informações, na grande rede, têm seus discos rígidos examinados ou vasculhados. É aí que os direitos individuais podem ser desrespeitados.

O direito à privacidade no trabalho também é uma questão muito importante. Alguns especialistas acreditam que haverá um choque entre os trabalhadores que desejam privacidade e as empresas que pretendem saber mais sobre seus empregados. Se, de um lado, as empresas têm o direito legal de monitorar e-mails de seus empregados, trilhas de auditoria, ICQ e outras formas de instant messaging, para evitar abusos e desvios de conduta, desnecessário dizer que muitos trabalhadores consideram esse tipo de supervisão desumano. No fundo, essas questões devem ser devidamente pactuadas no contrato de trabalho entre as partes envolvidas, o quem nem sempre ocorre.

3.3.3 O impacto da invasão da privacidade

Quem de nós já não recebeu malas diretas, e-mails, fax ou ligações telefônicas oriundas de fornecedores, oferecendo supostas ofertas e propagandas de produtos e serviços não solicitados? Todos os dias geramos informações, sem saber como as mesmas serão usadas. Esses são alguns exemplos em que o uso de informações gera um aborrecimento maior que propriamente uma ameaça à privacidade individual. Recentemente, a publicação de um livro intitulado "Endereços Eletrônicos dos Ricos e Famosos" provocou um aborrecimento eletrônico para algumas celebridades. O rápido crescimento da correspondência eletrônica inútil, como anúncios não solicitados, enviados por fax ou pela internet, é sem dúvida nenhuma, um subproduto do progresso das telecomunicações. Outro exemplo típico é o de espionagem, onde o software espião "fotografa" secretamente sites, chats groups, e-mails que o usuário visita ou envia e salva-os num arquivo secreto. Esse tipo de programa funciona do modo camuflado, de forma que o espionado jamais saberá o que acontece. O que se conclui de tudo isso é que, embora os dados normalmente sejam usados corretamente, as oportunidades para o seu mau uso são abundantes, e o impacto de suas conseqüências, imprevisíveis.

3.3.4 A correção no uso da informação

O mercado de informações é cada vez mais lucrativo, o que torna provável a continuidade desse tipo de negócio. A correção no uso dessas informações nos aponta para quatro questões relativas à privacidade: conhecimento, controle, notificação e consentimento. As pessoas devem ter o conhecimento de dados armazenados sobre elas e o direito de corrigir erros em sistemas de bancos de dados corporativos. Se as informações sobre pessoas forem utilizadas para outras finalidades, essas pessoas devem ser solicitadas a dar seu consentimento, previamente. Todas as pessoas têm o direito de saber e a capacidade de decidir sobre a utilização de seus dados por terceiros.

3.3.5 Legislação sobre privacidade

A privacidade é, em nossos dias, uma preocupação que transcende os limites nacionais. Nesse sentido, é importante apoiar-se em experiências e orientações fornecidas pelos órgãos e entidades de padronização em segurança de informação.

- **NIST** – (National Institute of Standard and Tecnology) dos Estados Unidos - possui publicações disponíveis na Internet – http://csrc.nist.gov/

- **BSI** – (British Standards Institute) da Inglaterra – possui publicações na Internet – http://www.bsi-global.com. ABS779, a qual se tornou o padrão ISO 17799, provê um conjunto de controle e boas práticas em segurança da informação;

- **ABNT** – (Associação Brasileira de Normas Técnicas), http://www.abnt.org.br – norma traduzida do padrão ISO 17799 o qual passou a ser o padrão brasileiro.

Da mesma forma, a legislação brasileira vem tentando acompanhar a evolução da tecnologia, e vários decretos já foram publicados, regendo a segurança digital.

- **Decreto nº 2.134, de 24 de janeiro de 1977:** regula a classificação, a reprodução e o acesso a documentos públicos de natureza sigilosa, que digam respeito à segurança da sociedade e do Estado e à intimidade do indivíduo;

- **Decreto nº 2.910, de 29 de dezembro de 1998:** estabelece normas para a salvaguarda de documentos, materiais, áreas, comunicações e sistemas de informação de natureza sigilosa;

- **Decreto nº 3.505, de 13 de junho de 2000:** institui a política de Segurança da Informação nos órgãos e entidades da Administração Pública Federal;

- **Decreto nº 3.587, de 05 de setembro de 2000:** estabelece normas para infra-estrutura das Chaves Públicas do Poder Executivo Federal.

- Existem, ainda, diversos Projetos de Lei em tramitação na Câmara dos Deputados e no Senado Federal, que tratam de assuntos relacionados a assinaturas digitais e crimes na Internet.

- **PL Nº 84/99 – Deputado Luiz Piauhylino:** trata de crimes de Informática. Aborda invasões de sistemas, danos a dados armazenados em computador, obtenção indevida de informações e difusão de pornografia;

- **PL nº 1.589/99 – Deputado Luciano Pizzato:** dispõe sobre comércio eletrônico, validade jurídica de documentos eletrônicos e assinatura digital. Dispõe condições para oferta de produtos e serviços por meio eletrônico e normas de proteção ao consumidor. Define regras para emissão de certificados;

- Existe, também, uma lei que trata de crimes contra a Previdência Social.

- **Lei nº 9.983/00:** Dentre os crimes por computador, trata da inserção de dados falsos e modificação ou alteração de dados, não autorizadas.

3.3.6 Normas corporativas sobre privacidade

Algumas empresas adotam códigos de privacidade próprios, que especificam como a privacidade de empregados, clientes e consumidores deve ser protegida. Nesses casos, elas compreendem que a invasão da privacidade pode prejudicar os negócios, afastar os clientes e reduzir drasticamente a receita e, conseqüentemente, os seus lucros. Essas normas devem abordar o conhecimento, o controle, a

notificação e o consentimento quanto ao armazenamento e utilização das informações. Podem tratar, também, de quem tem direito ao acesso a dados privados e quando eles podem ser usados. Os problemas para uma organização envolvida com a invasão da privacidade de alguém, podem ir muito além do aumento de tempo e de custos e resultarem na perda de confiança pública e de clientes.

3.4 AUDITORIA

Segundo [STA 98], auditoria é a análise dos registros de uma organização, para verificar sua exatidão. Mais especificamente, é a auditoria de Sistemas de Informações que tenta indicar quais controles e procedimentos devem ser estabelecidos, em uma determinada organização, e se os mesmos estão sendo utilizados corretamente. A auditoria é uma preocupação que deveria ser tomada em qualquer sistema de segurança que se preocupe com violações. É através dela que muitas fraudes eletrônicas são detectadas, já que, na maioria dos casos, a fraude eletrônica é camuflada como um acesso autorizado.

Entretanto, segundo [ZIN 97], a auditoria é um dos processos de segurança que é mais desconsiderado na hora de se estabelecerem medidas corriqueiras de proteção. Os motivos para tal desinteresse são, principalmente, de ordem econômica e de razões inerentes ao próprio processo de auditoria. Os objetivos da auditoria não são de impedir o acesso indevido, mas sim, de detectar possíveis violações que resistiram aos processos de identificação e de autorização e, se possível, determinar o caminho do violador e sua identidade.

Para a eficácia de uma auditoria, um dos requisitos fundamentais diz respeito à capacitação dos seus recursos humanos. Não basta estabelecer técnicas rígidas de auditoria, se os profissionais envolvidos na sua execução não são competentes o suficiente. Conseqüentemente, esses profissionais devem ser altamente qualificados, o que, sem dúvida nenhuma, torna os seus processos mais caros, limitando, assim, o universo de empresas que adotam este tipo de controle.

Devido aos altos custos envolvidos nesse tipo de tecnologia, os avanços das medidas de segurança mais recentes não têm sido uma alternativa viável economicamente para qualquer tipo de organização. Outro motivo para que a auditoria seja preterida, em favor de outras metodologias, é o de que não existem muitas ferramentas de auditoria disponíveis no mercado. As grandes empresas sempre podem desenvolver soluções próprias, mas, via de regra, não as colocam no mercado. É neste contexto que este trabalho se insere. Apesar de existir uma grande variedade de tipos de auditoria, em função da natureza das atividades a serem auditadas, existem elementos comuns em cada processo, e a proposta aqui é a de identificá-los e definir uma metodologia de apoio aos seus processos de auditoria.

3.4.1 Auditoria de sistemas de informação

De acordo com [STA 98], a auditoria de Sistemas de Informação busca responder a duas perguntas básicas:

- Que procedimentos e controles foram estabelecidos?
- Esses procedimentos e controles estão sendo usados adequadamente?

Além dessas questões básicas, alguns outros aspectos costumam ser examinados durante uma auditoria. Entre eles, estão o fluxo de documentos e relatórios, dentro de uma organização, o treinamento de usuários nos sistemas existentes e o tempo necessário para executar determinadas tarefas e resolver problemas e gargalos dos mesmos.

Existem, basicamente, dois tipos de auditoria:

- **Auditoria Interna:** conduzida por funcionários da organização;
- **Auditoria Externa:** conduzida por empresas especializadas ou por profissionais sem vínculo empregatício com a organização.

Em qualquer dos casos, seguindo [ZIN 97], existem algumas etapas que devem ser cumpridas, para se executar a função de auditoria, que consiste em um processo de verificação de registros de transações efetuadas, através da realização de trilhas de auditoria, com a identificação do "o quê", "quando", "quem" e "onde" as mesmas foram realizadas. Normalmente, essas transações estão todas devidamente autorizadas e são consideradas válidas e corretas. Porém, na auditoria, são considerados outros padrões para análise dessas transações. Parâmetros estatísticos são um bom exemplo: se um determinado usuário, devidamente autorizado a inserir registros de contas novas em um sistema, tem uma média de 100 inserções por mês, a auditoria pode verificar, que em determinado mês, esse usuário fez 150 inserções. Esta variação pode apontar para duas possibilidades:

- As inserções podem ser perfeitamente normais, decorrente de um mês movimentado e atípico, até então;

- As inserções podem ser fraudulentas, em que pese estarem devidamente autorizadas.

Assim, a auditoria de sistemas de informação também pode ser utilizada no processo de busca de indícios e comprovação de denúncias, buscando a comprovação de fraudes, violações e de seus autores.

Em [ABN 01], são preconizados alguns controles de auditoria de sistemas, com o intuito de minimizar os riscos de interrupção dos processos do negócio:

- Convém que os requisitos de auditoria sejam acordados com o nível apropriado da direção;

- Convém que o escopo da verificação seja acordado e controlado;

- Convém que a verificação esteja limitada ao acesso somente para leitura de software e dados;

- Convém que outros acessos diferentes dos mencionados, anteriormente, sejam permitidos somente através de cópias isoladas dos arquivos do sistema, que devem ser apagados ao final da auditoria;

- Convém que recursos de tecnologia, para execução da verificação, sejam identificados explicitamente e tornados disponíveis;

- Convém que requisitos para processamento adicional ou especial sejam identificados e acordados;

- Convém que todo acesso seja monitorado e registrado, de forma a produzir uma trilha de referência;

- Convém que todos os procedimentos, requerimentos e responsabilidades sejam documentados;

- Convém que acessos às ferramentas de auditoria de sistemas, isto é, software ou arquivos de dados sejam protegidos, para prevenir contra qualquer possibilidade de uso impróprio ou comprometimento;

- Convém que tais ferramentas sejam separadas de sistemas em desenvolvimento e em operação e não sejam mantidas em mídias de biblioteca ou áreas de usuários, a menos que forneçam um nível apropriado de proteção adicional.

3.4.2 O perfil do auditor

Segundo [ZIN 97], auditores são profissionais altamente qualificados, treinados para detectar indícios de fraudes que eventualmente possam ter sido cuidadosamente acobertados. Recebem treinamento especial para tanto e têm de possuir qualidades pessoais inatacáveis, pois o trabalho de auditoria pode ser completamente arruinado por posturas antiéticas. Uma fraude conduzida por uma quadrilha ou por uma chefia pode não ser descoberta nunca, bastando, para tanto,

que a equipe de auditoria esteja envolvida. Assim, o maior custo da auditoria é justamente o das qualidades morais dos profissionais que a conduzem.

Uma metodologia de auditoria, por melhor que seja, não é suficiente para garantir a eficácia da mesma. Ela é somente uma ferramenta que, como qualquer outra, depende e muito do seu operador, o auditor.

No caso presente, para uma metodologia de auditoria em banco de dados relacionais, o papel do auditor deve ser definido de acordo com a equipe responsável pela implantação e administração de banco de dados da organização. Essa equipe, cujo responsável é denominado de DBA (Database Administration) possui as seguintes atribuições:

- Instalação do SGBD (Sistema Gerenciador de Banco de Dados);
- Criação de banco de dados;
- Administração e monitoramento de espaço em disco, memória e conexões;
- Autorização de contas e determinação de permissões para os usuários;
- Realização de cópias de segurança (Backup) e a restauração (Restore) dessas cópias;
- Diagnóstico de problemas do sistema;
- "Fine-Tuning" do SGBD, para obter a melhor performance.
- Dependendo do tamanho dessa equipe, nem sempre caberá ao DBA a responsabilidade pela condução do processo de auditoria. Esta tarefa poderá ser perfeitamente delegada a algum profissional da equipe, desde que tenha as devidas permissões.

Este conceito é particularmente importante, neste trabalho, na medida em que o auditor, no caso de auditoria de bancos de dados relacionais, deverá demonstrar estar apto a desempenhar, entre outras tarefas:

- Analisar os objetos dos bancos de dados, quanto à necessidade de serem auditados ou não;

- Determinar o tipo de auditoria e o nível da mesma para cada objeto a ser auditado;

- Determinar os períodos em que a auditoria deve ser realizada;

- Determinar os processos de auditoria que devem ser conduzidos nos registros;

- Analisar os resultados dos processos e determinar se existe qualquer tipo de fraude ou violação.

Estas tarefas, como veremos a seguir, nortearam a definição das etapas que compõem a metodologia proposta, no presente trabalho.

3.5 BANCO DE DADOS

Como já mencionado em tópicos anteriores, uma das fontes mais importantes para qualquer organização, nos dias de hoje, é a sua coleção de dados. Pela importância crítica dos dados, muitas organizações têm-se voltado para as tecnologias de bancos de dados, no intuito de ajudá-las a administrarem melhor as imensas quantidades de dados com que elas têm que lidar.

3.5.1 Gerenciamento de dados

Sem os dados e a capacidade de processá-los, uma organização não teria condições de realizar, com eficiência e eficácia, a maioria de suas atividades. Para que esses dados sejam transformados em informação útil, devem ser,

primeiramente, organizados de forma significativa, para que possam ser devidamente gerenciados. Hoje em dia, com a velocidade das transformações que acontecem no mundo globalizado, em algumas situações, o gerenciamento desses dados pode e tornar-se bastante complexo.

Segundo [BEA 01], a maioria das organizações enfrenta, hoje, sérios problemas, causados pelo armazenamento de mais e mais dados históricos e pelo acesso a novas fontes de informação que se multiplicam rapidamente. Um dos meios mais básicos de gerenciamento de dados é através de arquivos individualizados por assunto. Normalmente, as empresas apresentam centenas ou mesmo milhares de "ilhas de informação", cujos dados não foram projetados com compatibilidades entre si. Para que decisões ou consultas estratégicas possam ser obtidas, a partir desses dados, é necessário que eles possam ser integrados, transformando-se, assim, em um recurso valioso de informações e conhecimentos. Com essa integração, vieram os bancos de dados. Segundo [STA 98], a abordagem de banco de dados é aquela em que um conjunto de dados relacionados é compartilhado por múltiplos programas aplicativos. Em vez de utilizarem arquivos de dados separados, cada aplicativo usa uma coleção de arquivos de dados que se inter-relacionam no banco de dados. Para que esses dados possam ser usados, de forma estratégica, as empresas dependem, cada vez mais, de novas tecnologias de banco de dados que possam ajudá-los nas tarefas de armazenar, acessar e usar

toda essa informação.

Para a implantação de uma abordagem de banco de dados, no gerenciamento de dados dentro de uma organização, é necessário um software chamado SGBD (Sistema de Gerenciamento de Banco de Dados). O SGBD consiste em um grupo de programas que pode ser usado com uma interface entre um banco de dados e um usuário ou um banco de dados e um programa aplicativo.

Ainda segundo [STA 98], a abordagem de banco de dados oferece vantagens e desvantagens, em relação ao enfoque tradicional, baseado em arquivos (tabelas 3.1 e 3.2).

Tabela 3.1 - Vantagens da abordagem de banco de dados

Campo	Explicação
Redundância reduzida de dados	A abordagem de banco de dados pode reduzir ou eliminar a redundância de dados. Os dados são organizados por um SGBD e armazenados em apenas um local. Disto resulta uma utilização do espaço do sistema de armazenamento com maior eficiência.
Integridade aperfeiçoada dos dados	Com a abordagem tradicional, algumas modificações dos dados não eram refletidas em todas as cópias dos dados mantidos em arquivos separados. Isto é evitado com a abordagem de banco de dados, porque não existem arquivos separados que contenham cópias dos mesmos dados.
Modificação e atualização mais fáceis	Com a abordagem de bancos de dados, o SGBD coordena, atualiza e faz modificações dos dados. Os programadores e usuários não têm que saber onde os dados estão fisicamente armazenados. Os dados são armazenados e modificados, ao mesmo tempo. A modificação e a atualização são também mais fáceis, porque os dados são armazenados em apenas um local, na maioria dos casos.

Independência de dados e programas	O SGBD organiza os dados, independentemente do programa aplicativo. Com a abordagem de banco de dados, o programa aplicativo não é afetado pela localização ou tipo de dado. A introdução de novos tipos de dados irrelevantes, para uma aplicação em particular, não requer uma nova redação daquele aplicativo, para manter a compatibilidade com o arquivo de dados.
Melhor acesso aos dados e à informação	A maior parte dos SGBD tem softwares que facilitam o acesso e a recuperação de dados do banco de dados. Na maioria dos casos, comandos simples podem ser dados, para obterem-se informações importantes. Relações importantes entre registros podem ser mais facilmente investigadas e exploradas, e os aplicativos podem ser combinados com maior facilidade.
Padronização do acesso aos dados	Uma característica básica da abordagem do banco de dados é a padronização, uma abordagem uniforme do acesso aos bancos de dados. Isto significa que os mesmos procedimentos globais são usados por todos os programas aplicativos, para recuperar os dados e a informação.

Uma estrutura para desenvolvimento de programa	Os procedimentos padronizados de acesso aos bancos de dados podem significar maior padronização do desenvolvimento de programas. Como os programas passam pelo SGBD, para ter acesso aos dados, no banco de dados, o acesso padronizado ao banco de dados pode fornecer um molde consistente para o desenvolvimento de programas. Além disso, cada programa aplicativo precisa apenas se referir ao SGBD e não, ao arquivo de dados propriamente dito, reduzindo o tempo de desenvolvimento do aplicativo.
Melhor proteção global dos dados	O uso e acesso dos dados localizados de forma centralizada são mais fáceis de serem monitorados e controlados. Códigos de segurança e senhas podem assegurar que apenas pessoas autorizadas tenham acesso a dados e informações particulares, no banco de dados, o que assegura a privacidade.
Fontes de dados e de informação compartilhadas	O custo do hardware, software e de pessoal pode ser estendido a um grande número de aplicações e usuários. Esta é uma característica básica de um SGBD.

Tabela 3.2 - Desvantagens da Abordagem de Banco de Dados

Desvantagens	Explicação
Custo relativamente alto de compra e operação de SGBD em um ambiente de operação de grande porte	Alguns SGBDs de grande porte podem custar centenas de dólares.
Equipe especializada	Funcionários especializados adicionais e pessoal de operação podem ser necessários, para implementarem e coordenarem o uso do banco de dados. Deve-se notar, contudo, que algumas organizações têm conseguido implementar uma abordagem de banco de dados sem pessoal adicional.
Vulnerabilidade aumentada	Embora os bancos de dados ofereçam melhor segurança, devido às medidas de segurança que podem ser concentradas em um sistema, eles também tornam um número maior de dados acessível ao invasor, se a segurança falhar. Além disso, se, por alguma razão, houver uma falha no SGBD, os programas de aplicações múltiplas serão afetados.

3.5.2 Organizando os dados em um banco de dados

A construção de um banco de dados requer algumas considerações, a saber:

- **Conteúdo:** que dados devem ser coletados e a que custo?
- **Acesso:** que dados podem ser acessados e por quem?
- **Estrutura lógica:** como os dados devem ser organizados?

- **Estrutura física:** onde os dados devem estar fisicamente armazenados?

A construção de um banco de dados exige, ainda, dois diferentes tipos de projetos:

- **Projeto lógico:** envolve a identificação e o detalhamento de relações entre os diferentes itens de dados e o seu agrupamento, em uma forma ordenada;

- **Projeto físico:** é o modelo de como os dados serão organizados e localizados, dentro do banco de dados.

Uma das ferramentas que os projetistas de bancos de dados usam para mostrar as relações entre dados é o modelo de dados. Um modelo de dados consiste em um mapa ou diagrama de entidades e suas inter relações. A modelagem de dados de uma empresa envolve a análise da necessidade de dados e informações de toda a organização.

Os diagramas de Entidade-Relacionamento (ER) podem ser empregados para mostrar as relações entre entidades, bem como as suas cardinalidades (um para um, um para muitos e muitos para muitos). Esta estratégia facilita o entendimento de modelo por pessoas não especializadas.

Os bancos de dados usam, basicamente, quatro modelos de organização:

- **Hierárquico:** tem um tipo de registro no alto, chamado pai, com registros subordinados, chamados filhos. Cada pai pode ter vários filhos, mas cada filho pode ter, apenas, um pai. Sua principal característica é a eficiência no acesso aos dados. Os gerenciadores mais comuns baseados neste modelo são: IMS da IBM e o SYSTEM 2000 da Honeywell;

- **Rede:** expansão da estrutura hierárquica, é uma relação proprietário – membro, na qual cada membro pode ter mais de um proprietário. Oferece

maior flexibilidade de organização de dados, em relação ao hierárquico. Os gerenciadores mais comuns baseados neste modelo são: IDMS, IDMS/R, DMS 1100, TOTAL e IMF;

- **Relacional:** em lugar de uma hierarquia de relações predefinidas, os dados são arranjados em tabelas bidimensionais. As tabelas podem estar ligadas por elementos de dados comuns, que são usados para acessar os dados, quando o banco de dados é solicitado. Cada linha é chamada tupla e representa um registro. As colunas da tabela são chamadas atributos, e os valores admissíveis para esses atributos são chamados domínio. As manipulações de dados básicos em uma tabela incluem a sua seleção (linhas), projeção (colunas) e junção (união de duas ou mais tabelas). É o modelo mais fácil de controlar, mais flexível e mais intuitivo, em relação aos anteriores. Os gerenciadores mais comuns baseados neste modelo são: Oracle, DB2, Microsoft SQL Server, Sybase, Interbase, Informix e Ingres;

- **Orientado a Objeto:** é basicamente um sistema em que a unidade de armazenamento é o objeto, com o mesmo conceito das linguagens de programação orientadas a objetos, a diferença fundamental é a persistência dos objetos, ou seja, os objetos continuam a existir mesmo após o encerramento do programa, através das construções orientadas a objeto, os programadores podem esconder os detalhes da implementação de seus módulos, compartilhar a referência a objetos e expandir seus sistemas através de módulos existentes. O banco de dados orientado a objeto combina os benefícios e conceitos da orientação a objetos com a funcionalidade dos bancos de dados. Os gerenciadores mais comuns baseados neste modelo são: Caché, Jasmine e Postgres.

A finalidade da análise de dados é avaliar os dados, para descobrir anomalias com o banco de dados. O processo de correção dessas anomalias é chamado de normalização. A normalização envolve o desmembramento de um arquivo em dois

ou mais, a fim de se corrigirem anomalias, quando feita em bancos de dados relacionais.

Por fim, são as necessidades da organização que determinarão o tipo e o modelo de banco de dados a usar. Conhecendo-se essas necessidades, outras características adicionais ainda devem ser consideradas, em termos de custo, controle e complexidade para a escolha de um banco de dados:

- **Tamanho:** refere-se ao número de registros ou à necessidade total de armazenamento;

- **Volatilidade:** refere-se à freqüência com que esses registros sofrem alterações;

- **Imediação:** refere-se à medida da rapidez com que essas alterações devem ser feitas.

Alguns autores acreditam que os SGBDs orientados a objetos serão os sucessores dos SGBDs relacionais. Isto se deve, principalmente ao fato que os primeiros preservam as características relacionais e possibilitam a utilização de características orientadas a objetos.

3.5.3 Conceitos fundamentais sobre SGBD

Conforme já abordado, um Sistema Gerenciador de Banco de Dados é um grupo de programas usado como uma interface, entre um banco de dados e os programas aplicativos ou entre um banco de dados e o usuário. De uma forma geral, um SGBD deve fornecer os seguintes recursos:

- **Variedade de interfaces com o usuário:** em função da diversidade de nível de conhecimento dos usuários que vão utilizar o mesmo SGBD, ele deve oferecer várias interfaces, incluindo consultas através de exemplos,

linguagem natural e formulários para usuários finais, linguagem de consulta interativa para usuários experientes entre outros;

- **Independência física dos dados:** programas aplicativos de bancos de dados devem ter total independência da estrutura física dos dados do banco. Este importante recurso torna possíveis alterações nos dados armazenados no banco, sem ter de fazer quaisquer alterações nos programas aplicativos de banco de dados. Como exemplo, podemos citar a alteração do critério de ordenação dos dados de uma determinada tabela, no banco de dados;

- **Independência lógica dos dados:** da mesma forma, o SGBD deve possibilitar que sejam feitas alterações na estrutura lógica do banco, sem afetar os programas aplicativos de banco de dados. Como exemplo, podemos citar a inclusão de um novo atributo na estrutura de uma determinada tabela no banco de dados;

- **Otimização de consulta:** o SGBD deve possuir um otimizador de consultas, que considere uma variedade de possíveis estratégias de execução para consultar os dados e, então, selecionar a mais eficiente;

- **Integridade de dados:** o SGBD deve possuir recursos de identificação de dados logicamente incoerentes e rejeitar o seu armazenamento;

- **Controle de concorrência:** o SGBD deve possuir mecanismos de controle que garantam que vários aplicativos podem acessar o mesmo dado, simultaneamente, inclusive para atualizações;

- **Backup e Recuperação:** o SGBD deve ter um subsistema que seja responsável pela gravação e recuperação de dados, quando houver necessidade, em decorrência de alguma falha de hardware, software ou mesmo de operação;

- **Segurança e autorização:** o SGBD deve ter controles de segurança internos, a fim de garantir que os seu dados estejam protegidos contra qualquer tipo de usuário não-autorizado ou mesmo contra mau uso.

Na década de 70, D.D. Chamberlain e outros do laboratório de pesquisa da IBM, em San Jose, Califórnia, desenvolveram uma linguagem de manipulação de dados padronizada, chamada linguagem de consulta estruturada (SQL – Structured Query Language). Em 1986, o American National Standard Institute (ANSI) e a International Standard Organizacion (ISO) adotaram o SQL como linguagem de consulta-padrão para bancos de dados relacionais. Após uma revisão para um padrão intermediário, em 1989, um padrão muito mais volumoso, chamado SQL 92, foi desenvolvido e finalmente lançado em dezembro de 1992. Atualmente, essas duas organizações estão desenvolvendo um novo padrão, o SQL 3, que abrange vários novos conceitos de bancos de dados, incluindo triggers, procedures armazenadas e numerosos conceitos orientados a objetos.

Sendo [PET 99], em comparação com algumas linguagens mais tradicionais, como C, C++ e Java, a SQL é uma linguagem orientada a conjuntos (enquanto que as primeiras são chamadas linguagens orientadas a registro). Isto significa que a SQL pode consultar muitas linhas de uma ou mais tabelas, usando apenas uma declaração. Este recurso é uma das vantagens mais importantes da SQL, permitindo o uso dessa linguagem em um nível logicamente mais alto do que as linguagens procedurais.

Ainda seguindo [PET 99], outra propriedade importante da SQL é sua característica não-procedural. Todo programa escrito em uma linguagem procedural (C,C++, Java) descreve como uma tarefa é realizada, passo a passo. Em comparação, a SQL, assim como qualquer outra linguagem não-procedural, descreve o que o usuário deseja. Desta forma, cabe ao SGBD a responsabilidade de encontrar o modo apropriado de resolver os pedidos dos usuários.

A SQL, assim como todas as linguagens de banco de dados, contém duas sub-linguagens, a saber:

linhas de telecomunicações, algumas empresas estão construindo uma réplica dos seus bancos de dados, que são periodicamente atualizados;

- **Bancos de dados orientados a objetos:** armazenam dados, como objetos que contêm tanto os dados quanto as instruções de processamento, necessários para completar a transação do banco de dados. Os objetos podem ser recuperados e relacionados por um Sistema de Gerenciamento de Banco de Dados Orientado para Objeto (SGBDO). Também oferecem a capacidade de reutilizar e modificar objetos existentes, para desenvolverem novos aplicativos de bancos de dados;

- **Bancos de dados de imagem:** além de armazenarem dados numéricos e alfanuméricos, armazenam imagens scaneadas como, por exemplo: documentos, mapas, raios-X, entre outros. Apresenta, como principal desvantagem, o alto consumo de espaço de armazenamento requerido. Um exemplo desse banco de dados são os sistemas de gerência eletrônica de documentos (SGED);

- **Bancos de dados de hipertexto:** armazenam dados alfanuméricos de uma forma não estruturada, em pedaços chamados nós. O usuário, então, estabelece ligações entre esses nós. As relações entre os dados, neste caso, podem ser criadas para atender as especificações do usuário, em vez de seguir os modelos anteriormente descritos;

- **Bancos de dados de hipermídia:** são uma extensão dos bancos de dados de hipertexto. Permitem armazenar e manipular formas de dados de hipermídia: dados gráficos, sonoros, de vídeos e alfanuméricos.

3.5.5 Gerenciando banco de dados

Segundo [STA 98], o gerenciamento de um banco de dados é parte da administração do banco de dados, onde os DBAs, são responsáveis por todos os aspectos do banco de dados da organização, a saber:

- Projeto e coordenação global do banco de dados;
- Desenvolvimento e manutenção de esquemas e subesquemas;
- Desenvolvimento e manutenção do dicionário de dados;
- Implementação do SGBD;
- Documentação do sistema e do usuário;
- Operações globais do SGBD;
- Testes e manutenções do SGBD;
- Estabelecimento de procedimentos de emergência e de recuperação, em caso de falhas.

Não cabe ao DBA a responsabilidade pela exatidão desses dados, mas cabe a ele agir como um guardião e monitorar o uso do banco de dados pelos seus usuários.

Muitos usuários querem ter acesso à maior parte possível dos dados armazenados em um banco de dados. Como os dados de uma organização podem ser confidenciais, as políticas de segurança devem ser desenvolvidas, para especificar quais dados podem ser recuperados e por quem. Como, certamente, haverá sempre muitos usuários para banco de dados, o potencial de segurança desses dados e os problemas de invasão de privacidade devem ser tratados, o quanto antes, quando da análise, projeto e implementação dos bancos de dados e

SGBDs. Muitos SGBDs possuem excelentes procedimentos e técnicas para proteção da privacidade individual e a manutenção da segurança dos dados; um exemplo disso é o SGBD Microsoft SQL Server, que será abordado no tópico a seguir.

3.5.6 O SGBD Microsoft SQL Server

Como parte do modelo aqui proposto, será utilizado o SGBD Microsoft SQL Server 2000, que passaremos a chamar de SQL Server. A ferramenta SQL - Server é um sistema gerenciador de banco de dados, amplamente conhecido, muito utilizado, atualmente, tanto em larga escala quanto para pequenos sistemas. Será através dele que faremos a implementação das triggers de auditoria, de forma totalmente parametrizada e independente de aplicações de terceiros.

3.5.6.1 SQL Server - definição

O Microsoft SQL Server 2000 é um Sistema de Gerenciamento de Bancos de Dados com solução de Data Warehousing, e-commerce, computação móvel e automação de força de venda [RAM 00]. Oferece uma alta escalabilidade, podendo ser utilizado tanto em notebooks como em grandes servidores. Possui novos recursos de processamento paralelo, otimizando consultas e row-level locking dinâmico. Outro recurso é o servidor OLAP integrado com o Data Tranformation Service, que permite a leitura de qualquer Banco de Dados.

O SQL Server oferece uma plataforma eficiente e flexível, suportando banco de dados de terabytes de informações: ele se adapta perfeitamente a aplicativos existentes e fornece um ambiente de baixo custo para personalizar e desenvolver novos aplicativos, criados exclusivamente para atender às necessidades de uma corporação.

O SQL Server pode ser usado tanto em laptops com o Windows 95 ou Windows 98 quanto em clusters multiprocessados com o Windows NT® Server Enterprise Edition ou Windows 2000 - com uma base única de códigos, fornecendo

100 por cento de compatibilidade do aplicativo com uma variedade de opções de distribuição.

O SQL Server foi desenvolvido para reduzir o custo total da empresa, facilitando a criação, o gerenciamento e a distribuição de aplicativos baseados no processamento de transações on-line (OLTP, On-line Transaction Processing). O SQL Server fornece ajuste e administração automatizados ao banco de dados, com excelente desempenho, bem como ferramentas sofisticadas para operações complexas. Inovações na facilidade de uso, escalabilidade, confiabilidade e desempenho, um modelo de programação rápido e simples para desenvolvedores, novo bloqueio dinâmico no nível de linha, backup ativo e gerenciamento multilocal. A versão Desktop para Windows 95, Windows 98, Windows NT Workstation ou Windows 2000 permite ao usuário acessar dados e aplicativos a partir de qualquer lugar. O SQL Server oferece muitas opções de replicação, para assegurar que alterações efetuadas em dados sejam automaticamente sincronizadas, incluindo alterações realizadas com o sistema operando off-line.

O SQL Server é perfeitamente compatível com o Windows NT 2000 e com sua tecnologia Internet Information Server (IIS), bem com o Microsoft Site Server, para fornecer a plataforma ideal de banco de dados para o comércio eletrônico. O SQL Server oferece facilidades de distribuição, excelente capacidade de gerenciamento, consulta de texto inovadora, consulta em Inglês e fácil publicação na Web, além da confiabilidade, escalabilidade e segurança necessárias para manter um site de comércio dinâmico.

3.5.6.2 Inovações e novos recursos

Esta nova versão apresenta novos recursos e melhorias em relação às versões anteriores, dentre as quais:

- **Simplificação da Configuração:** ajuste automático de memória, uso de discos e realização de diversas tarefas sem a necessidade de configuração específica, por exemplo, a atualização de estatísticas.

- **Novos Comandos SQL:**

  - **ALTER TABLE** - Modifica a definição de uma tabela, adicionando ou eliminando colunas e restrições ou desabilitando ou habilitando restrições.

  - **ALTER PROCEDURE** - Altera uma procedure previamente criada através da instrução CREATE PROCEDURE, sem alterar as permissões nem afetar quaisquer triggers ou procedures armazenadas dependentes.

  - **ALTER TRIGGER** - Altera a definição de um trigger criado previamente através da instrução CREATE TRIGGER. Se esta opção especificar informações de trigger diferentes da instrução original (CREATE), anulará o comportamento originalmente especificado.

  - **ALTER VIEW** - Altera uma view previamente criada através da instrução CREATE VIEW, sem alterar triggers ou procedures dependentes armazenadas e sem alterar permissões.

  - **BULK INSERT** - Copia um arquivo de dados em uma tabela de banco de dados em formato especificado pelo usuário

  - **COMMIT WORK** - Marca o final de uma transação. Esta instrução funciona de forma idêntica à instrução COMMIT TRANSACTION, exceto pelo fato desta aceitar um nome de transação definido pelo usuário. Especificando ou não a palavra-chave opcional WORK, esta sintaxe de COMMIT é compatível com o SQL-92.

  - **DENY** - Cria uma entrada no Sistema de Segurança que nega a permissão proveniente de uma conta de segurança no banco de dados atual, e impede que a conta de segurança herde a permissão por meio de seus membros de grupo ou funcionais.

- **ROLLBACK WORK** - Retrocede uma transação especificada pelo usuário para o início de uma transação. Esta instrução funciona de forma idêntica à instrução ROLLBACK TRANSACTION, exceto pelo fato desta aceitar um nome de transação definido pelo usuário. Especificando ou não a palavra-chave opcional WORK, esta sintaxe de ROLLBACK é compatível com o SQL-92. Ao aninhar transações, ROLLBACK WORK sempre retrocede à instrução BEGIN TRANSACTION mais externa e decrementa a variável global ***@@trancount*** até zero.

- **RESTORE** - restaura um log e um banco de dados inteiro, arquivo(s) de banco(s) de dados ou um log.

- **RESTORE FILELISTONLY** - retorna um conjunto de resultados com uma lista dos arquivos de bancos de dados e de logs contidos no conjunto de backups.

- **RESTORE HEADERONLY** - Restaura todas as informações de cabeçalho de backup para todos os conjuntos de backups, em um determinado dispositivo de backup. O resultado da execução da instrução é um conjunto de resultados.

- **RESTORE LABELONLY** - Retorna o conjunto de resultados contendo informações sobre a mídia de backup, identificado pelo argumento < backup_device > dado.

- **RESTORE VERIFYONLY** - Verifica o backup, mas não o restaura. A verificação checa se o conjunto de backups é completo, e se todos os volumes podem ser lidos, entretanto, não verifica a estrutura de dados contidas nos volumes de backup. Retorna uma mensagem afirmativa, caso o backup seja válido (The backup set is valid.').

- **Melhorias no gerenciamento de dados:** o tamanho das páginas do banco de dados passa a ser de 8k (anterior=2k). O número máximo de bytes em um registro passa a ser de 8060 bytes. As tabelas podem ter agora 1024 campos, ao invés de 250. O SQL também passa a suportar tipo de dados unicode.

- SQL Server Agent: **monitora os eventos no log do Windows NT. Quando o evento acontece, o SQL Server Agent compara os detalhes do evento com os alertas definidos no ambiente, e implementa a resposta se necessário.**

- **Serviços OLAP Integrados:** o processamento analítico on-line (OLAP, On-line Analytical Processing) fornece desempenho e eficiência otimizados para criação de relatórios, análise, suporte à decisão e modelagem de dados da empresa.

- **DTS Integrado:** os serviços de transformação de dados (DTS, Data Transformation Services) facilitam a criação e automatização da manutenção de data warehouses, permitindo importar, exportar e transformar dados de origens diferentes, graficamente.

- **Integração com o Microsoft Office 2000:** o Office 2000 ampliará, de forma significativa, a utilização do Microsoft Office como front-end de um banco de dados empresarial. O Office 2000 facilitará, ainda mais, o acesso e a análise de dados necessários para os usuários - não importando onde residam - e fornecerá mais opções de distribuição dessas informações.

- **Computação Móvel**

    - A *SQL Server Desktop* precisa de poucos recursos de máquina, é um SGBD relacional totalmente compatível para aplicativos móveis, sem recursos de administração, que facilita o uso sem intervenção de um administrador de banco de dados.

- Opções de Replicação Avançadas incluem o novo recurso Merge Replication, com resolução de conflitos e replicação anônima para sites da Internet.

- **Comércio**

  - Encriptação Dinâmica de Dados para senhas, dados, stored-procedures, views e triggers.
  - Pesquisa Completa de Texto suporta pesquisa lingüística de dados de caracteres, operando por palavras e frases, não apenas por padrões de caractere.

  - Fluxos de dados tabulares minimizam o tráfego geral e otimizam a comunicação da Internet com largura de banda restrita.

  - Novo Assistente da Web facilita a publicação de dados na Web.

- **Data Warehousing Eficiente**

  - OLAP Services integrado, para análise rápida e eficiente de informações complexas necessárias à criação de relatórios, análise de dados, suporte a decisões e modelagem de dados.

  - Novo Data Transformation Services para importar, exportar e transformar dados de origem diferente.

  - Consulta em Inglês para enviar perguntas em inglês ao banco de dados, em vez de consultas estruturadas.

- **Integração com o Microsoft Access 2000**

  - O Access 2000 pode acessar diretamente o SQL Server, permitindo uma interação cliente-servidor transparente e nativa.

  - Microsoft Data Engine (MSDE) é uma tecnologia do Access 2000 que fornece armazenamento de dados local e é totalmente compatível com o SQL Server - os usuários do Access 2000 podem utilizar o banco de dados Jet ou MSDE.

- **Integração com o Excel 2000**

  - Serviço PivotTable, junto com o OLAP Services, fornece ao Excel recursos de análise multidimensional tanto para soluções on-line quanto móveis.

- **Componentes para Web Office 2000**

  - Exibições de planilhas, gráficos e PivotTable podem ser associados diretamente ao SQL Server ou ao OLAP Services, oferecendo aos usuários uma maneira simples de visualizar e analisar dados.

- Integração com os Produtos Microsoft BackOffice®

  - Integração completa com Windows NT 2000 oferece segurança, um ambiente de aplicação Web e suporte ao Microsoft Transaction Server.

3.5.6.3 Arquitetura Cliente/Servidor

O SQL Server é um banco de dados relacional, capaz de suportar aplicações em arquitetura Cliente/Servidor, em que o banco de dados fica residente em um

computador central, chamado de servidor, e cujas informações são compartilhadas por diversos usuários que executam as aplicações em seus computadores locais, chamados de clientes. Esse tipo de arquitetura propicia uma maior integridade dos dados, pois todos os usuários utilizam as informações de uma única fonte. O tráfego de informações é consideravelmente reduzido, uma vez que apenas os resultados das solicitações dos clientes são retornados pelo servidor.

3.5.6.4 Arquitetura do SQL Server

O SQL Server é dividido em diversos componentes lógicos, como tabelas, índices, visões e outros elementos que são visíveis ao usuário. Esses elementos são fisicamente dispostos em dois ou mais arquivos em disco. O formato ou local onde os elementos lógicos são gravados, é transparente para o usuário do sistema.

Um servidor SQL Server pode conter um único banco de dados, sendo usado por diversos usuários, em diversos departamentos de uma empresa ou possuir vários bancos de dados usados por usuários específicos de cada departamento da mesma empresa de forma exclusiva.

3.5.6.5 Componentes do SQL Server

- **Banco de Dados (Databases):** contém os objetos usados para representar, armazenar e acessar os dados. É uma coleção de tabelas, visões, índices, triggers, stored procedures e outros objetos.

- **Tabelas (Tables):** são a essência do banco de dados. São as tabelas que armazenam os dados dentro do banco de dados. As tabelas agrupam os dados em forma de linhas e colunas. Cada linha representa um registro, e cada coluna representa um atributo ou campo de tabela. Cada campo da tabela mantém informações de um tipo. Os campos de uma tabela podem ter restrições quanto ao conteúdo que vão armazenar.

- **Diagramas de Banco de Dados (Database Diagrams):** o SQL Server permite a criação de Diagramas de Banco de Dados. Esses diagramas

são a representação gráfica de tabelas, índices e visões armazenadas pelo banco de dados. Todos esses elementos podem ser manipulados sem a necessidade de usar a linguagem Transact-SQL, como a alteração de características físicas do banco de dados ou suas tabelas. Os Diagramas permitem a manipulação dos elementos de um banco de dados, através dos recursos de dar um clique e arrastar (através do mouse) e da interação com caixas de diálogos, permitindo a execução de diversas tarefas.

- **Índices:** são tipos especiais de arquivos que trabalham associados a tabelas. Sua finalidade é acelerar o processo de acesso a um determinado registro ou grupo de registros. Ao realizar uma pesquisa em uma tabela indexada, o servidor detecta a coluna-chave e pesquisa no índice, que contém basicamente uma cópia da coluna a que se refere o índice e o endereço de sua linha dentro da tabela. Alguns índices são automaticamente criados, como é o caso da chave primária. Cada índice criado em uma tabela ocupa um espaço a mais no dispositivo de armazenamento.

- **Visões (Views):** é uma tabela virtual cujo conteúdo foi definido por uma consulta (query) ao banco de dados. A visão não é uma tabela física, mas um conjunto de instruções que retornam um conjunto de dados.

- **Procedimentos Armazenados (Store Procedures):** os dados armazenados somente podem ser acessados por meio da execução de comandos da linguagem Transact-SQL. Ao criar uma aplicação para servir de interface para o banco de dados, o desenvolvedor pode optar por criar um programa SQL, que seja armazenado localmente e enviado ao servidor para ser executado ou, então, criar e manter os programas no próprio servidor em *stored procedures* que podem ser acionadas por um programa na máquina cliente. Uma *stored procedure* pode aceitar parâmetros (valores que são passados para a *procedure*) para serem processados. Contudo, ao contrário de uma função (*function*), a *stored*

procedure não retorna nenhum valor. Uma vez criada uma *stored procedure*, ela pode ser usada por qualquer aplicação que acesse o banco de dados. As *stored procedures* são criadas pelo comando Transact-SQL CREATE PROCEDURE e podem ser modificadas através do comando ALTER PROCEDURE.

- **Triggers:** é uma *stored procedure* que é automaticamente executada quando um dado da tabela é alterado, em decorrência da execução de um comando SQL do tipo INSERT, UPDATE ou DELETE. Sua utilização é bastante ampla. Um *trigger* pode forçar limitações que sejam mais complexas do que as permitidas por meio da *constraint* CHECK, que limita as informações inseridas em uma coluna.

Pode-se criar um *trigger* associado ao comando INSERT, que faz a consulta a outras tabelas, retornando um valor lógico que permite limitar os dados atribuídos a uma determinada coluna. Um *trigger* pode ser criado para tratar uma replicação instantânea de um registro inserido em uma banco de dados, sendo responsável pela inserção desse mesmo registro em um outro banco de dados. Uma outra possibilidade seria a operação inversa, ou seja, ao ser excluído um registro de uma determinada tabela, o *trigger* se encarrega de eliminar outros registros associados a este, em outras tabelas. Um *trigger* é tratado como uma transação e pode ser desfeito (*rollback*) em caso de algum problema ser detectado.

- **Índices *Full-Text:*** este tipo especial de índice permite a execução de consultas baseadas em colunas, cujo conteúdo seja do tipo caracter (*Varchar* e *Text*). A tabela 3.3 mostra as principais diferenças entre um índice normal e um índice do tipo *Full-Text*.

Tabela 3.3 - Índices no SQL Server

Índices Normais	Índices Full-Text
São criados e apagados por meio de comandos SQL.	São criados e apagados por meio de *stored procedures.*
Podem existir diversos índices por tabela.	Pode existir apenas Um índice por tabela.
São automaticamente atualizados Quando os campos da tabela são alterados.	São atualizados apenas por solicitação.

- **Assistentes (Wizards):** O SQL Server usa intensivamente o recurso chamado assistentes ou *Wizards,* para realizar diversas tarefas administrativas. Sem os assistentes, tais tarefas teriam de ser efetuadas por meio de comandos Transact-SQL. O assistente, por sua vez, exibe uma série de caixas de diálogo que interagem com o usuário, solicitando informações e utilizando-as para executar as tarefas às quais se destinam. A tabela 3.4 traz uma lista dos principais assistentes disponíveis na nova versão e suas finalidades.

Tabela 3.4 - Comandos no SQL Server

Assistente	Finalidade
Create Backup	Cria um *backup* do banco de dados.
Configure Publishing and Distribution	Configura um banco de dados para replicação.
Create Alert	Executa a criação de um alerta.
Create Database	Executa a criação de um banco de dados.
Create Diagram	Executa a criação de um *Database Diagram.*
Create Index	Executa a criação de um índice.
Create Job	Executa a criação de um *job.*
Create New Data Service	Executa a instalação e configuração de um *data source* ODBC.

Create SQL Server Login	Executa a criação de um *login* de acesso aos usuários.
Create Publication	Executa a criação de uma publicação para replicação.
Create Store Procedures	Executa a criação de *stored procedures* para adição, exclusão e atualização de linhas.
Create Trace	Executa a criação de uma trace.
Create View	Executa a criação de uma visão.
Database Maintenance Plan	Executa a criação de um arquivo de manutenção.
Disable Publishing and Distribution	Executa a desabilitação de um esquema de publicação e replicação.
DTS Export	Executa a criação de um pacote DTS para exportação de dados.
DTS Import	Executa a criação de um pacote DTS para importação de dados.
Full-Text Indexing	Executa o processo de criação de um índice *Full_Text*.
Index Tuning	Executa o processo de ajuste fino de um índice.
Make Master Service	Executa o processo de criação de um *master server*.
Make Target Server	Executa o processo de criação de um *target server*.
Pull Subscription	Executa o processo de recuperação de dados de um servidor de replicação.
Push Subscription	Executa a criação do processo de envio dos dados para um servidor de replicação.
Register Servers	Executa o processo de registro de servidores SQL.

Setup	Executa o processo de instalação e configuração do SQL Server.
Transfer Data Upgrade	Atualização de transferência.
Web Assistant	Executa a criação dos passos necessários para a criação de uma página Web baseada no conteúdo de uma tabela ou importação de dados da Web.

3.5.6.5 Ferramentas

O SQL Server é acompanhado de um conjunto de ferramentas gráficas cuja finalidade é simplificar ou agilizar a execução de tarefas administrativas.

- **MICROSOFT MANAGEMENT CONSOLE (MMC):** é a interface básica para gerenciamento de um servidor *BackOffice*. O MMC está por trás de várias ferramentas, como o SQL *Enterprise Manager* ou o *Olap Manager.*

- **SQL SERVER CLIENT CONFIGURATION:** é uma ferramenta destinada a gerenciar a configuração cliente para conexões de rede definidas pelo usuário e para o *DB-Library* e *Net-Libraries.*

Podem ser configurados os seguintes itens, segundo a tabela 3.5:

Tabela 3.5 - Guia General de configuração

Opção	Função
Default Network Library	Especifica a biblioteca - padrão para comunicação com o SQL Server.
Server Alias Configuration	Indica os aliases dos computadores clientes.
Network Library	Indica a biblioteca de rede usada.
Connection parameters	Indica eventuais parâmetros associados ao endereço de conexão.
Add	Permite a adição de uma nova conexão de rede.
Remove	Permite a remoção de uma conexão.
Edit	Permite a edição de uma conexão.

- **SQL SERVER ENTERPRISE MANAGER:** é a principal ferramenta para gerenciamento de servidores SQL Server. Entre as várias tarefas realizadas pelo *Enterprise Manager*, podemos destacar:

    - Criar, visualizar e manter o conteúdo de bancos de dados por meio da criação de tabelas, *stored procedures*, índices, regras e diagramas.

- Importar e exportar dados.
- Transformar dados.
- Executar tarefas administrativas.

- **SQL SERVER NETWORK LIBRARY CONFIGURATION:** tem por finalidade configurar a conexão com o SQL Server, nos casos em que o protocolo de comunicação entre o servidor e seus clientes não esteja funcionando.

- **SQL SERVER PROFILER:** tem a finalidade de registrar continuamente as atividades do servidor. Os dados podem ser enviados para a tela por meio de gráficos ou então para um arquivo ou tabela, que poderá ser oportunamente analisado.

Através do SQL *Server Profiler,* é possível fazer a monitoração de eventos do servidor, como: tentativas de *login*, conexões e desconexões ao servidor, execução de scripts *batch* da linguagem Transact-SQL e *deadlocks*.

O SQL *Server Profiler* permite a criação de *traces,* com o propósito de coletar informações sobre o servidor e seus eventos. Essas informações podem ser armazenadas em arquivos especiais, em tabelas do banco de dados ou visualizadas diretamente pelo *Server Profile*. Uma *trace* é composta pelos dados capturados sobre os eventos que estão em monitoração. Os filtros são os critérios que determinam quais eventos serão monitorados pela *trace*. Para que o *Server Profile* possa monitorar um servidor, é necessário que esse servidor esteja devidamente registrado.

- **SQL SERVER QUERY ANALYSER:** permite executar comandos SQL, scripts e visualizar graficamente as etapas de execução de uma consulta.

Além disso, executa a análise de índices e propõe mudanças para a melhoria de desempenho.

- **SQL SERVER SERVICE MANAGER:** é uma ferramenta utilizada para iniciar ou interromper temporária ou definitivamente o SQL Server (MSSQLServer), SQL Server Agent e o Microsoft Distributed Transaction Coordinator (MSDTC). Ele deve aparecer como um ícone padrão na barra de tarefas.

- **SQL SERVER SETUP:** é usado para instalar e reconfigurar o servidor. Depois de instalado o servidor, esta ferramenta pode ser usada para ajustar opções de segurança, reconstruir o banco de dados Master, trocar opções de rede etc.

- **VERSION UPGRADE WIZARD:** facilita o trabalho de migração do banco de dados do SQL Server de versões anteriores.

- **SQL SERVER PERFORMANCE MONITOR:** oferece uma análise instantânea das atividades do banco de dados, sendo uma excelente ferramenta para diagnosticar eventuais problemas com o sistema.

O *Performance Monitor* é um raios-X das atividades do servidor, máquina local ou remota. Através dele é possível verificar o estado ou o comportamento de objetos como processadores, memória, cache, *threads,* e processos. Estes objetos podem ser associados a contadores que mostram dados estatísticos sobre os mesmos.

O *Performance Monitor* é composto por quatro janelas, chamadas *Chart, Alert, Log e Report.* Estas janelas podem ser exibidas por meio de opções de mesmo nome no menu *View.*

As estatísticas registradas serão perdidas se não forem salvas, antes do fechamento do programa. Para salvar os ajustes para uma das janelas, é usada a

opção *Save Settings* correspondente no menu *File*. Para salvar os ajustes de todas as janelas é usada a opção *Save Workspace* do menu *File*.

Para incluir a análise de novos objetos, deve ser usada a opção *Add to Chart*, onde estão disponíveis várias opções no campo *Object*.

3.5.6.7 Data Warehouse e OLAP

Um *Data Warehouse* é usado como a base de um sistema de suporte à decisão. Ele é criado para superar alguns dos problemas encontrados, quando uma organização tenta executar a análise estratégica, usando o mesmo banco de dados que é empregado para executar o *Online Transaction Processing* (OLTP).

Um sistema típico OLTP se caracteriza por grandes números de usuários simultâneos que adicionam e modificam dados. O banco de dados representa o estado de determinado negócio, tal como um sistema de reservas de companhia aérea, em um determinado momento.

Entretanto, o grande volume de dados, mantido em muitos sistemas OLTP, pode sobrecarregar uma organização. À medida que os bancos de dados ficam maiores, com dados mais complexos, o tempo de resposta pode ser rapidamente comprometido, devido à concorrência pelos recursos disponíveis.
Um sistema OLTP típico tem muitos usuários acrescentando dados ao banco de dados, enquanto um número menor de usuários gera os relatórios a partir do banco de dados. À medida que o volume dos dados aumenta, os relatórios levam mais tempo para serem gerados.

Entretanto, quando os usuários precisam analisar seus dados, inúmeros problemas podem surgir:

1. Como os usuários não entendem os relacionamentos complexos entre as tabelas, eles não podem gerar consultas *ad hoc*.

2. Como os bancos de dados de aplicativo podem ser segmentados em vários servidores, fica difícil para os usuários encontrarem as tabelas.
3. As restrições de segurança evitam que os usuários acessem os dados com o nível de detalhes de que precisam.
4. Os DBAS proíbem a consulta *ad hoc* dos sistemas OLTP, para evitar que os usuários analíticos executem consultas que podem reduzir o desempenho dos bancos de dados de produção de missão crítica.

O *Data Warehouse* e o *Online Analytical Processing* (OLAP) fornecem as chaves para solucionar esses problemas.

O *Data Warehouse* é uma abordagem para armazenar dados na qual as fontes heterogêneas de dados (que geralmente vêm de vários bancos de dados OLTP) são migradas para um armazenamento de dados homogêneo e separado.Os *Data Warehouses* fornecem estes benefícios para os usuários analíticos:

- Os dados são organizados adequadamente para as consultas analíticas e não para o processamento de transação.

- As diferenças entre as estruturas de dados em vários bancos de dados heterogêneos podem ser solucionadas.

- As regras de transformação de dados podem ser aplicadas para validar e consolidar os dados, quando eles são movidos do banco de dados OLTP para o *Data Warehouse*.

- As questões de segurança e desempenho podem ser solucionadas sem exigir alterações nos sistemas de produção.

Eventualmente, as organizações querem manter armazenamentos de dados menores e mais orientados para os tópicos, chamados *data marts*. Ao contrário de um *Data Warehouse*, que geralmente encapsula todos os dados de análise de uma

empresa, um *data mart* normalmente é um subconjunto dos dados empresariais, orientado para um conjunto menor de usuários ou funções de negócios.

- **O Que É um Data Warehouse?** de uma forma simplificada, um Data Warehouse é um banco de dados orientado para consultas, resultado de uma extensa análise e transformação de dados da empresa. Um Data Warehouse pode ajudar os aplicativos de suporte à decisão e OLAP, porque fornece dados que são consolidados e consistentes, orientados para o assunto, históricos, somente leitura (read only).

- **Dados Consolidados e Consistentes:** um *Data Warehouse* consolida os dados operacionais de uma variedade de fontes com convenções de nomeação, medidas, atributos físicos e semântica consistentes.

Por exemplo, em muitas organizações, as aplicações usam dados semelhantes em formatos diferentes: as datas podem ser armazenadas no formato Juliano ou gregoriano, dados verdadeiros/falsos podem ser representados como um/zero, ligado/desligado, verdadeiro/falso ou positivo/negativo. Aplicações diferentes também podem usar termos diferentes, para descreverem o mesmo tipo de dados. Uma aplicação pode usar o termo "saldo", enquanto outra usa "quantia total" para representar a quantia em dinheiro de uma conta bancária.

Os dados precisam ser armazenados no *Data Warehouse* em um único formato aceitável, decidido pelos analistas de negócios, apesar das variações nas fontes operacionais externas. Essa consolidação permite que os dados de toda a organização, tais como dados antigos dos *mainframes*, dados de planilhas ou mesmo dados da Internet, sejam consolidados no *Data Warehouse* e tenham referência cruzada eficiente, fornecendo aos analistas uma melhor compreensão do negócio.

- **Dados Orientados para o Assunto:** as fontes de dados operacionais de toda uma organização tendem a conter uma quantidade grande de dados sobre uma variedade de funções relacionadas aos negócios, tais como registros de clientes, informações sobre produtos e assim por diante. Entretanto, a maioria dessas informações está misturada a dados que não têm relevância para os relatórios de negócios ou executivos e está organizada de uma forma que torna difícil a consulta aos dados. O *Data Warehouse* organiza somente as principais informações do negócio, a partir de origens operacionais, para que estejam facilmente disponíveis para os analistas de negócios.

- **Dados Históricos:** os dados dos sistemas OLTP representam corretamente o valor atual em qualquer momento. Os sistemas OLTP quase sempre contêm somente os dados atuais. Por exemplo, uma aplicação de entrada de pedidos sempre mostra o valor atual do estoque; ela não mostra o estoque em algum momento do passado. A consulta do estoque em um momento posterior pode retornar uma resposta diferente.

Entretanto, os dados armazenados em um *Data Warehouse* são precisos, em algum ponto do passado, porque os dados armazenados representam as informações históricas. Os dados de um *Data Warehouse* geralmente representam os dados ao longo de um período de tempo, talvez até dez anos ou mais. Na verdade, os *Data Warehouses* armazenam instantâneos dos dados operacionais de um negócio, gerados ao longo de um período de tempo. Eles são precisos em um momento específico e geralmente não mudam.

- **Dados Somente Leitura:** como os dados armazenados em um *Data Warehouse* representam um ponto no tempo, as exclusões, inserções e atualizações (além daquelas envolvidas no processo de carregamento de dados - *data loads*) não se aplicam a um *Data Warehouse*. Depois que os dados foram movidos para o *Data Warehouse*, geralmente eles não mudam, a menos que os dados estejam incorretos.

Normalmente, as únicas operações que ocorrem em um *Data Warehouse*, depois que ele é configurado, são o carregamento e a consulta dos dados.

Como os dados não são modificados, depois de carregados, o projeto de um *Data Warehouse* pode ser otimizado nas consultas por meio do uso efetivo dos índices, dados pré-calculados e desnormalização do banco de dados físico.

Nos casos em que os dados do *Data Warehouse* não precisam ser modificados, as ferramentas OLAP podem ser usadas para gerenciar consultas em relação aos dados estáticos de *warehouse* e dados dinâmicos. Por exemplo, um grupo que trabalha no orçamento corporativo usaria os dados de *warehouse* para os números do ano anterior, juntamente com dados dinâmicos, tais como Orçamento e previsões do ano atual.

Data Warehouse e OLAP:

Embora sejam muito usados para significar a mesma coisa, os termos Data Warehouse e Online Analytical Processing (OLAP) são componentes diferentes de sistemas quase sempre chamados de suporte à decisão.

Um Data Warehouse é um banco de dados que contém os dados que geralmente representam o histórico comercial de uma organização. Os dados históricos de um Data Warehouse são usados para atividades de análise que aceitam as decisões de negócios em muitos níveis, desde o planejamento estratégico até a avaliação de desempenho de uma unidade organizacional separada. Os dados de um Data Warehouse são organizados para aceitar a análise, em vez de processar transações em tempo real, como nos sistemas OLTP.

O OLAP é uma tecnologia que processa os dados de um Data Warehouse em estruturas multidimensionais, para fornecer resposta rápida a consultas analíticas complexas. O propósito do OLAP é organizar e resumir grandes quantidades de dados para que eles possam ser analisados e avaliados rapidamente, usando ferramentas on-line como o Microsoft PivotTable Service e representações gráficas.

A resposta para uma consulta a dados históricos quase sempre leva a consultas subseqüentes, à medida que o analista pesquisa respostas ou explora possibilidades - os sistemas OLAP fornecem a velocidade e a flexibilidade para aceitar o analista em tempo real.

3.5.6.8 Versões disponíveis

O SQL Server está disponível nos idiomas inglês, inglês internacional, francês, alemão, espanhol e japonês nas versões:

- Enterprise Edition - O SQL Server versão Enterprise apresenta um nível mais elevado de escalabilidade e confiabilidade. Entre as principais inovações estão o *failover clustering*, recursos para particionar serviços OLAP e a capacidade de utilizar até 8 processadores e gerenciar até 32 Gigabytes de memória.

- Desktop Edition - O SQL Server versão Desktop é um SGBD relacional completo, desenvolvido para uso pessoal ou compartilhado. Pode ser executado no Windows 95/98, Windows NT Workstation ou Windows 2000. Pode utilizar até dois processadores. Esta versão foi especialmente desenvolvida para computação móvel, permitindo ao usuário acessar dados e aplicativos a partir de qualquer lugar. O SQL Server oferece muitas opções de replicação (*merge replication*), para assegurar que alterações efetuadas em dados sejam automaticamente sincronizadas, incluindo alterações realizadas com o sistema operando off-line. O SQL Server Desktop pode ser distribuído também com a licença de acesso do cliente por estação.

3.5.6.9 Requisitos do sistema e considerações finais

No geral, considera-se a seguinte configuração de equipamentos e softwares, para um bom desempenho no desenvolvimento:

Versão Servidor:

- PC com um processador Pentium III (600 MHz ou superior);
- Sistema operacional Microsoft Windows NT Server versão 4.0 ou Windows NT Server 4.0 Enterprise Edition com Service Pack 4 ou posterior (Service Pack 4 incluído);
- Microsoft Internet Explorer 4.01 com Service Pack 1 ou posterior (os dois estão incluídos);
- 128 MB de RAM;
- Espaço disponível no disco rígido:
- 65-180 MB para o servidor; cerca de 170 MB para instalação típica;
- 35-50 MB para serviços OLAP; cerca de 50 MB para instalação típica;
- 24-36 MB para a Consulta em Inglês; cerca de 36 MB para instalação típica;
- Unidade de CD-ROM;
- Monitor VGA ou de resolução mais alta; Super VGA recomendável;
- Microsoft Mouse ou dispositivo apontador compatível.

Observação: O SQL Server pode utilizar até quatro processadores. O suporte a processadores adicionais se encontra disponível no SQL Server Enterprise Edition.

Versão Desktop (semelhante aos requisitos da versão Servidor, com as seguintes exceções):

- Cada instalação do SQL Server Desktop requer uma licença de acesso do cliente por estação para o SQL Server; o SQL Server Desktop irá interagir apenas com o SQL Server no modo por estação;

- 65-180 MB de espaço disponível no disco rígido; cerca de 170 MB para instalação típica;

O SQL Server traz várias inovações em relação a sua versão anterior. As que mais chamaram a tenção foram os recursos de OLAP, Data Warehouse e OLAP.
A versão Desktop desvincula o SQL Server da necessidade de rodar apenas em servidores com o sistema operacional Windows NT.

A administração de recursos locais e remotos, bem como a otimização do desempenho do banco de dados, foi bastante simplificada através da disponibilização de diversas ferramentas, com interfaces simplificadas e a possibilidade do uso de assistentes (*Wizards*) para a maioria das tarefas a serem executadas.

A administração de backups e restaurações de banco de dados tornou-se uma tarefa bastante simplificada, podendo ser realizada, também, através de assistentes.

Por esta e outras vantagens, mencionadas no decorrer do trabalho apresentado, concluímos que esta versão é um grande avanço na tecnologia de banco de dados e que promete suprir as necessidades de grandes corporações.

4 DESCRIÇÃO DO MODELO PROPOSTO

Segundo [ZIN 97], um dos maiores obstáculos para a realização de auditorias em bancos de dados é a dificuldade de se encontrar um método, metodologia ou ferramenta que seja flexível e, ao mesmo tempo, genérica, que auxilie os desenvolvedores e administradores de bancos de dados a criarem estruturas de monitoramento e gerarem estatísticas de utilização de banco de dados adequadamente.

Via de regra, se um desses profissionais desejar a realização de auditoria de banco de dados, deverá definir e criar toda uma estrutura de apoio (tabelas para armazenamento de informações e códigos para recuperação dessas informações) e ainda definir de que forma essa estrutura deverá ser utilizada, a cada vez que uma transação ocorrer. Esse processo é todo manual, e sempre que o responsável quiser reutilizá-lo em outro banco de dados, deverá defini-lo e criá-lo novamente.

Assim, um método que se encarregue de criar todas essas estruturas e códigos, de uma forma padronizada e independente, faz-se necessário e dotaria os profissionais, ligados aos processos de auditoria de bancos de dados, de uma ferramenta valiosa na condução do monitoramento e auditoria em ambientes diversos. Assim, caberia a esses profissionais, como trabalho de criação, a identificação dos alvos no monitoramento e a personalização dos resultados da auditoria.

4.1 CONCEITUAÇÃO TÉCNICA

Para a concepção do modelo aqui proposto, conforme [ZIN 97], identificou-se a existência de 05 etapas, distintas entre si, ligadas a qualquer processo de auditoria de bancos de dados:

- **Análise do modelo de dados:** envolve o reconhecimento do esquema do banco de dados a ser auditado. Nessa fase, o responsável pelo processo de auditoria identifica todos os objetos envolvidos no banco de dados, bem como suas relações. Além dos recursos de identificação normalmente

disponibilizados nos SGBDs atuais, várias ferramentas CASE possuem recursos de identificação, através da geração de scripts, dicionário de dados e diagramas do modelo a ser auditado;

- **Identificação dos objetos e dos níveis de auditoria:** envolve a identificação e seleção, com base na análise do modelo de dados, das tabelas e campos a serem auditados. Além disso, o responsável pela auditoria deve definir, nesta fase, para cada tabela, quais os tipos de transações que deverão ser auditadas (inclusão, alteração e exclusão);

- **Criação das estruturas de auditoria:** para que a auditoria aconteça, é necessário que sejam construídas as estruturas que armazenem as informações das tabelas e campos selecionados na etapa anterior. Para construção dessas estruturas, o responsável pela auditoria deve ter em mente a capacidade de resposta para os seguintes questionamentos: quem fez o quê, onde e quando. Assim, as estruturas de auditoria deverão conter, no mínimo, as seguintes informações:

    - **Tabela:** permite a identificação do objeto auditado;

    - **Identificação do registro:** permite determinar o endereço do registro auditado;

    - **Campo:** permite identificar o campo do registro auditado;

    - **Operação:** permite identificar o tipo da transação realizada no registro auditado (inclusão, alteração ou exclusão);

    - **Conteúdo:** permite determinar o valor antigo do campo auditado;

    - **Data e hora:** permite identificar a data e hora da transação;

- **Identificação do usuário:** permite identificar o usuário responsável pela transação;

- **Identificação do local:** permite identificar o local onde foi realizada a transação;

- **Monitoramento:** esta etapa depende, não só do armazenamento eficiente das informações de auditoria, mas também da obtenção eficiente dos dados, no momento em que as transações estão sendo efetuadas. Esta etapa deverá ser implementada através de um recurso disponível, na maioria dos SGBDs relacionais, que é o TRIGGER. Conforme já abordado no item 3.5.6, triggers são conjuntos de instruções que são disparadas sempre que um evento específico (inserção, alteração ou exclusão) modifica um ou mais registros em uma tabela. A vantagem deste método é que os triggers são disparados automaticamente, sempre que é detectada uma mudança em uma tabela, não importando a origem dessa alteração. Assim sendo, o monitoramento consistirá justamente na geração dos registros adulterados, nas estruturas de auditoria mencionadas no item anterior;

- **Auditoria:** esta etapa não depende de nenhum tipo de ferramenta de apoio para ter efeito. Necessita apenas das informações obtidas na etapa de monitoramento e de alguém que seja capaz de analisá-las e de efetuar considerações sobre o resultado dessa análise. Todavia, a agilidade, nesta etapa, é fundamental para qualquer tipo de providência que seja necessária. Desta forma, a etapa de auditoria deverá ser apoiada por rotinas automatizadas que gerem estatísticas sobre as informações monitoradas e as compare com dados fornecidos pelo auditor. As estatísticas são, basicamente, totalizações obtidas por usuário e por tabela, sobre o número de transações realizadas na tabela em um determinado período. O tipo de transação (inclusão, alteração ou exclusão) também é levado em conta. O auditor pode, por exemplo, definir desvios-padrão aceitáveis para cada usuário, tabela e transação. Desta forma, ao se compararem as estatísticas geradas com os desvios-padrão estabelecidos, as rotinas podem detectar,

rapidamente, possíveis comportamentos estranhos. Esses comportamentos estranhos são, então, investigados pelo auditor, em busca de violações comprovadas. O auditor tem ainda a possibilidade de investigar individualmente os conteúdos das tabelas de auditoria.

Na identificação dessas 05 etapas, percebe-se facilmente que as três primeiras são iniciais aos processo de auditoria de banco de dados e que, se adequadamente conduzidas, são executadas apenas uma vez, a menos que o modelo de dados seja alterado ou os objetivos da auditoria sejam modificados. Já a quarta etapa (monitoramento) ocorre durante toda a utilização do banco de dados pelos usuários. Finalmente, a quinta etapa (auditoria) necessita da interferência do responsável pelo processo de auditoria de banco de dados e deverá ser conduzida pela aplicação de processos estatísticos sobre os dados que foram capturados no monitoramento. Essas duas últimas etapas são repetidas constantemente, enquanto que as três primeiras, usualmente, são executadas apenas uma vez.

4.2 CONCEPÇÃO BÁSICA

O modelo aqui proposto visa adotar os administradores de bancos de dados ou o responsável pelos processos de auditoria de uma metodologia que lhe permita selecionar quais objetos de um banco de dados devem ser monitorados.

A idéia é se implementar uma tabela, (que chamaremos de tabela de logs), no banco de dados que se quer auditar, com uma estrutura contendo todas as informações que se deseja registrar. À medida em que se faz qualquer tipo de alteração, nos registros das demais tabelas do banco de dados que se deseja auditar, através da implementação de triggers de inclusão, alteração e exclusão, a informação manipulada é registrada nessa tabela de logs. Importante aqui registrar algumas premissas relevantes, quanto à concepção de tal modelo:

- A estrutura dos campos da tabela de logs deverá contemplar genericamente qualquer campo contido no universo das tabelas do banco de dados a ser auditado;

- Como a implementação de triggers é ao nível de tabelas, só precisamos implementar os triggers (inclusão, alteração e exclusão) nas tabelas a serem auditadas e para os campos previamente selecionados;

- A estrutura da tabela de logs deverá contemplar a identificação da tabela, do registro, do campo, da operação realizada, do conteúdo do campo antes de ser manipulado, do responsável, local, data e hora da manipulação;

- Os triggers deverão ser concebidos numa codificação o mais genérica possível, sem comprometimento da performance do sistema, com o mínimo de adaptação de uma tabela para outra, facilitando assim a sua implementação, a partir de um código-padrão.

Diante de tais premissas, concebemos a estrutura básica da tabela de logs, bem como a codificação específica de cada trigger.

4.3 CARACTERÍSTICAS TÉCNICAS

Para implantação de tal modelo, em ambiente Microsoft SQL Server (versão 7.x ou superior), e em consonância com as premissas básicas já citadas, foram estabelecidas as seguintes definições:

- Tamanho máximo do nome da tabela a ser logada = 40 bytes;

- Número máximo de colunas da tabela a ser logada = 256;

- Tamanho máximo do nome do campo a ser logado = 40 bytes;

- Tamanho máximo do conteúdo do campo a ser logado = 500 bytes;

- Tamanho máximo do nome do usuário responsável pela manipulação = 20 bytes;

- Tamanho máximo do nome da estação responsável pela manipulação = 20 bytes;

- Como houve a necessidade de se armazenar um identificador do registro manipulado na tabela de logs, e como a tabela de logs é uma tabela genérica de registro de qualquer tabela, a solução encontrada foi estabelecer um tipo de identificador de registro-padrão para todas as tabelas a serem logadas. Diante disso, para que o modelo funcione, conforme estabelecido em suas premissas, impõe-se que todas as tabelas possuam um campo tipo identity, que terá a função de identificador do registro;

- Como o conteúdo da informação a ser registrada é do tipo *varchar* de 500 bytes, torna-se inviável se registrarem campos tipo *image;* sendo assim, este tipo de campo não pode ser logado;

- A necessidade de se combinarem aspas simples (') e aspas duplas (") na combinação de sentenças de comandos "Transact SQL", para implementação do modelo, exigiu a inclusão da cláusula SET QUOTED_IDENTIFIER OFF no código de criação dos triggers.

As seis primeiras limitações elencadas acima são decorrentes da calibração de valores, calcados na experiência do que ocorre com mais freqüência (podendo perfeitamente ser alteradas, conforme a necessidade). Já as demais, são decorrentes de limitações da linguagem adotada, não podendo, portanto, ser alteradas.

Isto posto, desenvolveu-se então para implementação de tal modelo, uma tabela de logs, três triggers de logs (alteração, inclusão e exclusão) a serem criadas para cada tabela que se quer auditar e uma stored procedure de consulta da tabela de logs e da tabela a ser auditada, que se inter-relacionam conforme ilustrado na figura 4.1.

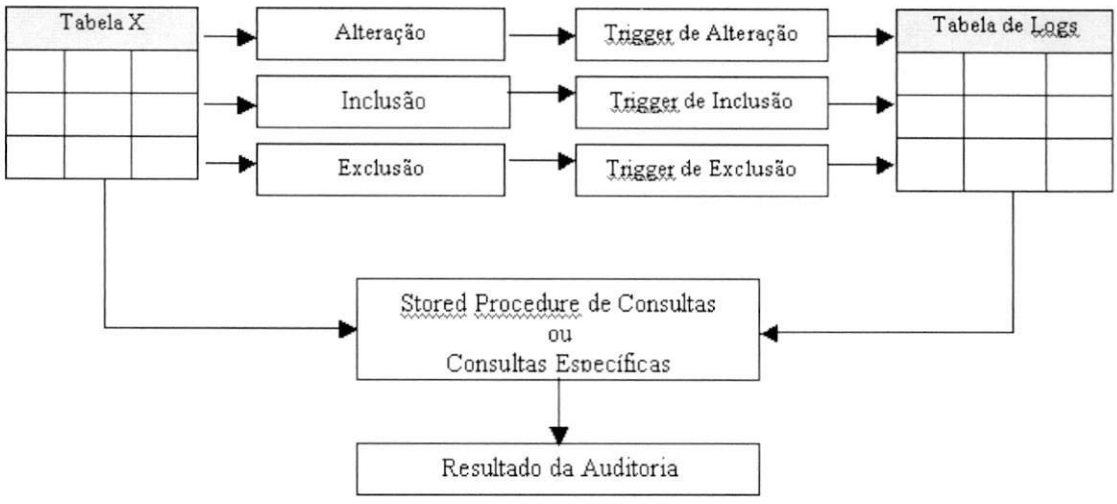

Figura 4.1 - Esquema do Modelo Proposto

A seqüência de implantação do modelo é a seguinte:

- Criação da tabela de logs no Banco de Dados;
- Criação dos triggers (inclusão, alteração e exclusão) para cada tabela a ser logada;
- Criação da Stored Procedure de consultas para realização da auditoria.

Feito isto, à medida em que as atualizações forem sendo processadas nas tabelas (independente da forma), os triggers serão disparados automaticamente e alimentarão a tabela de logs que, por sua vez, poderão ser consultadas pela procedure.

4.4 CÓDIGOS FONTES

A seguir, elencamos os scripts desenvolvidos em "Transact SQL" para implementação do modelo.

4.4.1 Criação da tabela de logs

```
CREATE TABLE [dbo.Nome da Tabela de Logs]
(
	[Log_Transacao] [int] IDENTITY (1, 1) NOT NULL ,
	[Log_Tabela] [varchar] (40) COLLATE Latin1_General_CI_AS NULL ,
	[Log_Operacao] [char] (1) COLLATE Latin1_General_CI_AS NULL ,
	[Log_identificador] [int] NULL ,
	[Log_Campo] [varchar] (40) COLLATE Latin1_General_CI_AS NULL ,
	[Log_ConteudoAntigo] [varchar] (500) COLLATE Latin1_General_CI_AS NULL ,
[Log_Usuario] [varchar] (20) COLLATE Latin1_General_CI_AS NULL CONSTRAINT [DF_
dbo.Nome da Tabela de Logs _Log_Usuario] DEFAULT (suser_sname()),
[Log_Estacao] [varchar] (20) COLLATE Latin1_General_CI_AS NULL CONSTRAINT [DF_
dbo.Nome da Tabela de Logs _Log_Estacao] DEFAULT (host_name()),
[Log_Data] [smalldatetime] NULL CONSTRAINT [DF_ dbo.Nome da Tabela de Logs
_Log_Data] DEFAULT (getdate()),
	CONSTRAINT [PK_ dbo.Nome da Tabela de Logs ] PRIMARY KEY  CLUSTERED
	(
		[Log_Transacao]
	)
ON [PRIMARY]
)
ON [PRIMARY]
GO
```

4.4.2 Criação do Trigger de alteração

```
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

CREATE TRIGGER Nome do Trigger ON dbo.Nome da Tabela
FOR UPDATE
AS
if @@rowcount = 0 return  -- Nenhum registro manipulado, portanto nada a fazer
declare @log_tabela   varchar(40),        -- Nome da tabela manipulada
	@log_operacao char(01),           -- Operação efetuada (Inclusão,
Alteração, Exclusão)
	@log_identificador int,           -- Conteúdo do identificador do registro
manipulado
	@log_campoidentity varchar(40),   -- Nome do campo identificador da tabela
manipulada
	@log_campo varchar(40),           -- Nome do campo manipulado
	@log_conteudoantigo varchar(500), -- Conteúdo do campo manipulado
	@var_auxiliar1      varchar(500), -- Variável auxiliar 1
	@var_auxiliar2      varchar(500), -- Variável auxiliar 2
	@num_col            tinyint,      -- Contador de campo
	@max_col            tinyint       -- Número de campos da tabela manipulada
set nocount on
select @log_tabela  = object_name((select parent_obj from     sysobjects where id =
@@procid))
select @log_operacao = 'A'
select @num_col = 1
```

```
select @max_col = (select max(ordinal_position)
                          from information_schema.columns
                          where table_name = @log_tabela)
select @log_campoidentity = (select syscolumns.name
                             from sysobjects,syscolumns
                             where sysobjects.type = 'U' and
                             sysobjects.name = @log_tabela and
                             sysobjects.id = syscolumns.id and
                             syscolumns.status & 128 = 128)
--
-- Criaçao das tabelas temporárias com os registros antigos e novos
--
select * into #tempo_a from deleted  -- Registro antigo
select * into #tempo_b from inserted -- Registro novo
--
-- Declaraçao, abertura e posicionamento dos cursores a / b
--
exec('
declare cursor_a cursor for select '+@log_campoidentity+' from #tempo_a
declare cursor_b cursor for select '+@log_campoidentity+' from #tempo_b')
open cursor_a
open cursor_b
fetch next from cursor_a into @var_auxiliar1
fetch next from cursor_b into @var_auxiliar2
--
-- Laço de scrool nos cursores
--
while @@fetch_status = 0
begin
   select @num_col = 1
   while (@num_col <= @max_col) -- Laço de teste dos campos da tabela
   begin
      select @log_campo = (select col_name (object_id (@log_tabela) ,@num_col) --
Definiçao do campo a ser testado
                           from information_schema.columns
                           where table_name = @log_tabela and
       column_name = col_name(object_id(@log_tabela),@num_col))
--
-- Testa se o campo foi alterado (se verdadeiro, alimenta a tabela de log)
--
      exec('
      if (select '+@log_campo+' from #tempo_a where '+@log_campoidentity+' =
'+@var_auxiliar1+') <>
         (select '+@log_campo+' from #tempo_b where '+@log_campoidentity+' =
'+@var_auxiliar2+')
      begin
         insert into dbo.Nome da Tabela de Logs(log_tabela, log_operacao,
log_campo) values ('+'"'+@log_tabela+'","'+@log_operacao+'","'+@log_campo+'")
         update dbo.Nome da Tabela de Logs set log_conteudoantigo =
convert(varchar(500),(select '+@log_campo+' from #tempo_a where
'+@log_campoidentity+' = '+@var_auxiliar1+')), log_identificador =
'+@var_auxiliar1+'
         where log_transacao = (select max(log_transacao) from dbo.Nome da Tabela
de Logs)
      end')
      select @num_col = @num_col + 1 -- Incrementa o próximo campo da tabela
      continue
   end
   fetch next from cursor_a into @var_auxiliar1 -- Posiciona próximo registro no
cursor a
   fetch next from cursor_b into @var_auxiliar2 -- Posiciona próximo registro no
cursor b
end
close cursor_a
close cursor_b
deallocate cursor_a
deallocate cursor_b
```

```
set nocount off

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

4.4.3 Criação do Trigger de inclusão

```
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

CREATE TRIGGER Nome do Trigger ON dbo.Nome da Tabela
FOR INSERT
AS
if @@rowcount = 0 return -- Nenhum registro manipulado, portanto nada a fazer
declare @log_tabela    varchar(40),       -- Nome da tabela manipulada
        @log_operacao char(01),           -- Operação efetuada (Inclusão, Alteração,
Exclusão)
        @log_identificador int,           -- Conteúdo do identificador do registro
manipulado
        @log_campoidentity varchar(40), -- Nome do campo identificador da tabela
manipulada
        @var_auxiliar2 varchar(500)       -- Variável auxiliar 2
set nocount on
select @log_tabela = object_name((select parent_obj from sysobjects where id =
@@procid))
select @log_operacao = 'I'
select @log_campoidentity = (select syscolumns.name
                             from sysobjects,syscolumns
                             where sysobjects.type = 'U' and
                                   sysobjects.name = @log_tabela and
                                   sysobjects.id = syscolumns.id and
                                   syscolumns.status & 128 = 128)
--
-- Criação das tabelas temporárias com os registros novos
--
select * into #tempo_b from inserted    -- Registro novo
--
-- Declaração, abertura e posicionamento do cursor b
--
exec('
declare cursor_b cursor for select '+@log_campoidentity+' from #tempo_b')
open cursor_b
fetch next from cursor_b into @var_auxiliar2
--
-- Laço de scrool nos cursores
--
while @@fetch_status = 0
begin
exec('
   begin
      insert into dbo.Nome da Tabela de Logs (log_tabela,log_operacao) values
('+'''+@log_tabela+''','''+@log_operacao+''')
      update dbo.Nome da Tabela de Logs set log_identificador = '+@var_auxiliar2+'
      where log_transacao = (select max(log_transacao) from dbo.Nome da Tabela de
Logs)
   end')
   fetch next from cursor_b into @var_auxiliar2-- Posiciona próximo registro no
cursor b
end
close cursor_b
deallocate cursor_b
```

```
set nocount off

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

4.4.4 Criação do Trigger de exclusão

```
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

CREATE TRIGGER Nome do Trigger ON dbo.Nome da Tabela
FOR INSERT
AS
if @@rowcount = 0 return -- Nenhum registro manipulado, portanto nada a fazer
declare @log_tabela    varchar(40), -- Nome da tabela manipulada
        @log_operacao char(01),     -- Operação efetuada (Inclusão, Alteração,
Exclusão)
        @log_identificador int,     -- Conteúdo do identificador do registro
manipulado
        @log_campoidentity varchar(40), -- Nome do campo identificador da tabela
manipulada
        @var_auxiliar2 varchar(500)  -- Variável auxiliar 2
set nocount on
select @log_tabela      = object_name((select parent_obj from sysobjects where id =
@@procid))
select @log_operacao = 'I'
select @log_campoidentity = (select syscolumns.name
                             from sysobjects,syscolumns
                             where sysobjects.type = 'U' and
                                   sysobjects.name = @log_tabela and
                                   sysobjects.id = syscolumns.id and
                                   syscolumns.status & 128 = 128)
--
-- Criação das tabelas temporárias com os registros novos
--
select * into #tempo_b from inserted  -- Registro novo
--
-- Declaraçao, abertura e posicionamento do cursor b
--
exec('
declare cursor_b cursor for select '+@log_campoidentity+' from #tempo_b')
open cursor_b
fetch next from cursor_b into @var_auxiliar2
--
-- Laço de scrool nos cursores
--
while @@fetch_status = 0
begin
exec('
   begin
      insert into dbo.Nome da Tabela de Logs (log_tabela,log_operacao) values
('+''''+@log_tabela+'',''+@log_operacao+'')
      update dbo.Nome da Tabela de Logs set log_identificador = '+@var_auxiliar2+'
      where log_transacao = (select max(log_transacao) from dbo.Nome da Tabela de
Logs)
   end')
   fetch next from cursor_b into @var_auxiliar2 -- Posiciona próximo registro no
cursor b
end
close cursor_b
deallocate cursor_b
```

```
set nocount off

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

4.4.5 Criação da Stored Procedure de consultas

```
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

CREATE PROCEDURE Nome da Stored Procedure
@log_tabela          varchar(40)    = null, -- Nome da tabela manipulada
@log_operacao     char(01)   =   null,   --   Operação   efetuada   (Inclusão,   Alteração,
Exclusão)
@log_identificador int = null, -- Conteudo do identificador do registro manipulado
@log_campo           varchar(40)   = null, -- Nome do campo manipulado
@log_conteudoantigo varchar(500) = null, -- Conteúdo do campo manipulado
@log_usuario varchar(20) = null, -- Usuário responsável pela operação efetuada
@log_estacao    varchar(20) = null, -- Estação onde a operação foi efetuada
@log_data1 smalldatetime = null, -- Primeira data em que a operação foi efetuada
@log_data2 smalldatetime = null  -- Segunda data em que a operação foi efetuada
as

if @log_tabela = '?'
begin
   print 'procedure pc_consulta
@log_tabela          varchar(40), -- Nome da tabela manipulada
@log_operacao char(01), -- Operação efetuada (Inclusão, Alteração, Exclusão)
@log_identificador int, -- Conteudo do identificador do registro manipulado
@log_campo           varchar(40),   -- Nome do campo manipulado
@log_conteudoantigo varchar(500), -- Conteúdo do campo manipulado
@log_usuario  varchar(20), -- Usuário responsável pela operação efetuada
@var_estacao  varchar(20), -- Estação onde a operação foi efetuada
@log_data1 smalldatetime,  -- Primeira data em que a operação foi efetuada
@log_data2 smalldatetime   -- Segunda data em que a operação foi efetuada'
end

declare @log_campoidentity varchar(40), -- Nome do campo identificador da tabela
manipulada
        @log_consulta varchar(500) -- Variável auxiliar para mo0ntagem da consulta
set nocount on

if @log_data2 is null
begin
   select @log_data2 = @log_data1
end

if @log_tabela is null
begin
   select @log_consulta = 'select * from dbo.Nome da Tabela de Logs'
end
else
begin
   select @log_campoidentity = (select syscolumns.name
                                from sysobjects,syscolumns
                                where sysobjects.type = 'U' and
                                      sysobjects.name = @log_tabela and
                                      sysobjects.id = syscolumns.id and
                                      syscolumns.status & 128 = 128)
   select @log_consulta = ' select * from dbo.Nome da Tabela de Logs,'+@log_tabela+
```

```
'     where     dbo.Nome     da     Tabela     de     Logs.log_identificador     =
'+@log_tabela+'.'+@log_campoidentity+' and'+
' dbo.Nome da Tabela de Logs.log_tabela = '+'"'+@log_tabela+'"'
   if @log_operacao is not null
   begin
      select @log_consulta = @log_consulta + ' and ' +
      'dbo.Nome da Tabela de Logs.log_operacao = '+'"'+@log_operacao+'"'
   end
   if @log_identificador is not null
   begin
      select @log_consulta = @log_consulta + ' and ' +
'dbo.Nome        da        Tabela        de        Logs.log_identificador        =
'+convert(varchar(10),@log_identificador)
   end
   if @log_campo is not null
   begin
      select @log_consulta = @log_consulta + ' and ' +
      'dbo.Nome da Tabela de Logs.log_campo = '+'"'+@log_campo+'"'
   end
   if @log_conteudoantigo is not null
   begin
      select @log_consulta = @log_consulta + ' and ' + 'dbo.Nome da Tabela de
Logs.log_conteudoantigo = ' + '"' + @log_conteudoantigo + '"'
   end
   if @log_usuario is not null
   begin
      select @log_consulta = @log_consulta + ' and ' + 'dbo.Nome da Tabela de
Logs.log_usuario = '+'"'+@log_usuario+'"'
   end
   if @log_estacao is not null
   begin
select @log_consulta = @log_consulta + ' and ' + 'dbo.Nome da Tabela de
Logs.log_estacao = '+'"'+@log_estacao+'"'
   end
   if @log_data1 is not null
   begin
      select @log_consulta = @log_consulta + ' and ' +
      'convert(char(10),dbo.Nome da Tabela de Logs.log_data,103) between '+
      '"'+convert(char(10),@log_data1,103)+'"'+' and '+
      '"'+convert(char(10),@log_data2,103)+'"'
   end
end

select @log_consulta = @log_consulta + ' order by log_transacao'
exec (@log_consulta)

set nocount off

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
```

Como se pode verificar, esta procedure de consulta foi concebida de uma forma totalmente "parametrizada". Sendo assim, potencialmente poderemos efetuar consultas a partir da combinação de qualquer campo que compõe a tabela de logs, inclusive relacionando a consulta com o registro atual da tabela logada, como se pode verificar nos exemplos do item a seguir:

4.5 EXEMPLOS PRÁTICOS

As respostas às 10 questões abaixo procuram ilustrar a potencialidade do modelo proposto, em relação a uma determinada base de dados:

- Quantas e quais atualizações foram efetuadas na base de dados no mês X?
  Exec **nome da stored procedure**
  @log_data1 = '01/X/2002',
  @log_data2 = '31/X/2002'

- Quais foram as atualizações sofridas na tabela Y durante esse período?
  Exec **nome da stored procedure**
  @log_tabela = 'Y',
  @log_data1 = '01/X/2002',
  @log_data2 = '31/X/2002'

- Quem efetuou atualizações na tabela Z, de que tipo e quando?
  Exec **nome da stored procedure**
  @log_tabela = 'Z'

- Houve exclusões na base de dados em um determinado período? Em quais tabelas?
  Exec **nome da stored procedure**
  @log_operação = 'E',
  @log_data1 = início do período
  @log_data2 = final do período

- Como era um determinado registro X manipulado na tabela W?
  Exec **nome da stored procedure**
  @log_tabela = 'W',
  @log_identificador = X

- Quando, onde e por quem determinado registro X de uma tabela W foi manipulado no ano 2001?

Exec **nome da stored procedure**
@log_tabela = 'W',
@log_identificador = X,
@log_data1 = '01/01/2001',
@log_data2 = '31/12/2001'

- Que tipo de operação o usuário W efetuou na tabela Y, durante o primeiro semestre de 2001?

  Exec **nome da stored procedure**

  @log_tabela = 'Y',

  @log_usuário = 'W',

  @log_data1 = '01/01/2001',

  @log_data2 = '30/06/2001'

- Quem e que tipo de operações foram realizadas no campo X da tabela Y na estação Z, durante o mês de janeiro de 2002?

  Exec **nome da stored procedure**

  @log_tabela = 'Y',

  @log_campo = 'X',

  @log_estação = 'Z',

  @log_data1 = '01/01/2002',

  @log_data2 = '31/01/2002'

- Quais tabelas sofreram atualizações durante esse período?

  Select distinct (log-tabela)

  From **nome da tabela de logs**

  Where **nome da tabela de logs**.log_data between

  '01/01/2002' and '31/01/2002'

- Considerando que a média mensal de exclusões em uma tabela Y é de X exclusões, houve algum excesso em relação a essa média, no ano 2001?

  Select 'Mês'=mounth(**nome da tabela de logs**.log_data),

  'Exclusões'=count(*)

  From **nome da tabela de logs**

  Where **nome da tabela de logs**.log_operacao='E' and

  nome da tabela de logs.log_tabela=Y and

  nome da tabela de logs.log_data between

  '01/01/2001' and '31/12/2001'

  Group by month(**nome da tabela de logs**.log_data)

  Having count(*) > X

5 TRABALHOS FUTUROS

Baseada no modelo proposto, a idéia seria a concepção de um Sistema de Gestão de Auditoria, que chamaremos de SAUDIT, onde se pudesse agregar as funcionalidades do modelo proposto, abordadas no item anterior, a um número maior de SGBDs relacionais, existentes no mercado, que possuam recursos de triggers (Oracle, Sybsse, Ingres, SQL Server etc).

5.1 DESCRIÇÃO

O SAUDIT deverá ser desenvolvido de uma forma modular, onde se contemplarão 4 das 5 etapas de auditoria, descritas no item anterior (identificação dos objetos e dos níveis de auditoria, criação das estruturas de auditoria, monitoramento e auditoria), deixando a primeira etapa (análise do modelo de dados) por conta dos recursos normalmente disponibilizados nos SGBDs atuais, ou mesmo através de ferramentas CASE, já consagradas no mercado, onde através da geração de scripts, dicionário de dados e diagramas do modelo, essa análise poderá ser realizada.

A seguir, apresentamos algumas premissas básicas que deverão nortear a concepção e desenvolvimento do SAUDIT:

- Por uma questão de praticidade e robustez, o SAUDIT deverá ser desenvolvido em versões individualizadas de SGBDs, pois normalmente as organizações possuem ambientes distintos, quando da existência de mais de um SGBD instalado; sendo assim, a idéia é se ter uma versão do SAUDIT para cada SGBD em operação;

- Obedecendo a princípios básicos de segurança, o acesso ao SAUDIT deverá ser precedido por um processo de validação de usuário, tanto no banco de dados que se pretende auditar, como no próprio SAUDIT, onde só poderão acessar o sistema os usuários previamente cadastrados na aplicação e no banco de dados, bem como com as devidas permissões de manipulações e criações de objetos no banco;

- Todas as operações, numa seção conectada pelo SAUDIT, deverão ser realizadas ao nível de banco de dados, previamente selecionado pelo usuário na fase de identificação, no momento do acesso ao sistema;

- Os módulos e funções deverão estar disponibilizados de uma forma clara e sintética, correspondendo à cronologia das etapas de auditoria previamente descritas;

- A etapa de identificação dos objetos e dos níveis de auditoria deverá ser contemplada com recursos que possibilitem a seleção ao nível de campo de cada objeto a ser auditado;

- A etapa de auditoria deverá ser contemplada com recursos que possibilitem a digitação e/ou cálculo periódico das estatísticas de manipulações realizadas, por objeto auditado;

- A etapa de auditoria também deverá ser contemplada, para cada objeto auditado, com consultas individualizadas, apresentando os dados atuais e os dados anteriores à manipulação, bem como consultas estatísticas, apresentando médias de manipulações esperadas juntamente com as médias de manipulações realizadas;

- A etapa de monitoramento deverá ser contemplada com mecanismos que possibilitem ao usuário, caso queira, exportar e importar os dados auditados;

- O SAUDIT ainda deverá disponibilizar mecanismos de flexibilização das limitações apresentadas no modelo proposto, inerentes a cada SGBD e ambiente a ser auditado.

5.2 MÓDULOS E FUNÇÕES

Com base nas premissas apresentadas no item anterior, o SAUDIT deverá ser modularizado de acordo com as etapas de auditoria a contemplar, seguido de um módulo de configuração do ambiente a ser auditado. O diagrama da figura 5.1 procura ilustrar a disposição dos módulos que deverão compor o SAUDIT, bem como a hierarquia de suas funções.

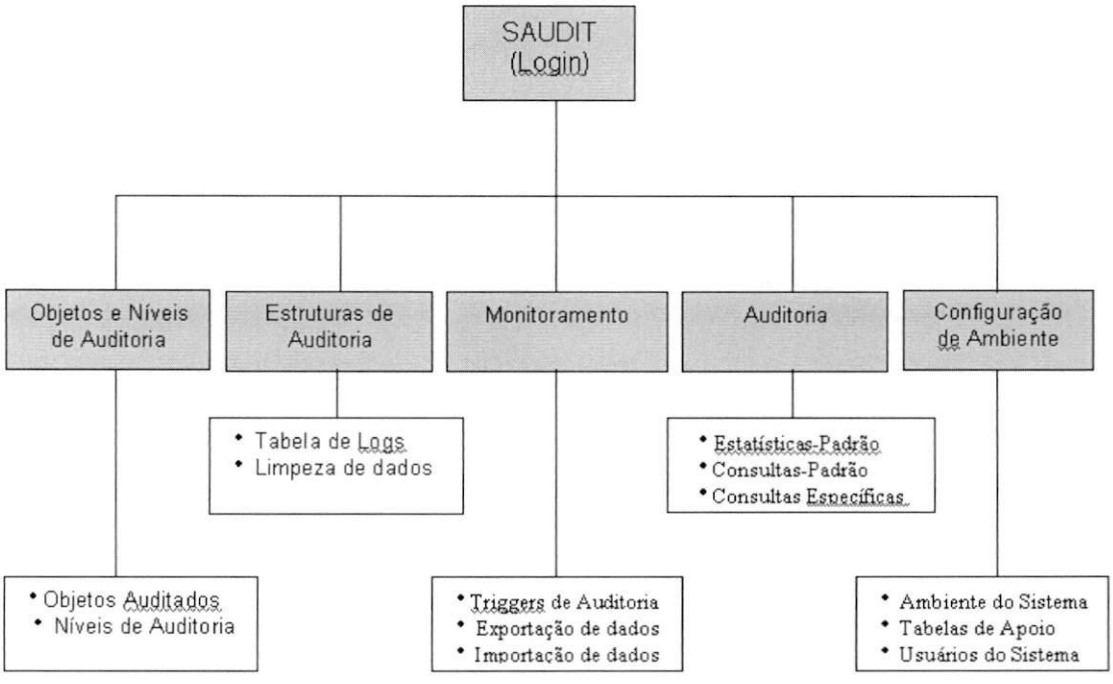

Figura 5.1 - Diagrama Hierárquico de Módulos e Funções

5.2.1 Saudit (login)

Será o módulo principal do sistema, através do qual se validará o usuário responsável pelo processo de auditoria, através do fornecimento do nome e senha de acesso, bem como o banco de dados, objeto da auditoria. Importante, aqui, registrar que a validação do usuário se dará tanto na aplicação como no banco de dados, ou seja, para se auditar um determinado banco de dados, o usuário deverá estar cadastrado no SAUDIT e no SGBD, para o banco de dados específico e com as devidas permissões de acesso/manipulações de objetos.

5.2.2 Objetos e níveis de auditoria

Será o módulo em que o usuário informará ao sistema, após a análise do modelo de dados, quais os objetos e em que nível os mesmos devem ser auditados.

- **Objetos Auditados:** função que tem como objetivo selecionar os objetos a serem auditados. Nesta fase, o SAUDIT deverá apresentar todas as tabelas que compõem o banco de dados, objeto da auditoria, permitindo que o usuário selecione, de uma forma individual ou global, as tabelas do modelo. Durante este processo de seleção, para cada tabela o SAUDIT deverá analisar a composição dos campos, em busca de possíveis chaves estrangeiras de outras tabelas, bem como a existência do campo "identity", fundamental para a formulação da auditoria. Caso seja encontrada uma ou mais chaves estrangeiras, o SAUDIT deverá apresentar essas informações ao usuário, oferecendo a opção de se incluírem, automaticamente, todas as tabelas referenciadas. Se o usuário escolher esta opção, todas as tabelas referenciadas serão incluídas na lista, passando pelo mesmo processo de verificação descrito acima. Cada tabela será incluída uma única vez na lista, sendo que as tabelas já incluídas serão ignoradas na verificação. Esta facilidade é indispensável pois, em muitos casos, uma tabela que possua várias chaves estrangeiras necessita das tabelas que são referenciadas, para que possamos entender melhor a informação auditada. Após selecionar todas as tabelas que devem ser auditadas, o usuário deverá ser conduzido a uma nova função, denominada "Níveis de Auditoria", descrita a seguir;

- **Níveis de Auditoria:** função que tem como objetivo selecionar, para cada item incluído na lista de objetos auditados, os campos a serem monitorados. De forma semelhante ao item anterior, o SAUDIT deverá disponibilizar um mecanismo de seleção, individual ou de todos os campos, pertencentes ao objeto-corrente. Importante, aqui, frisar a necessidade de verificação das limitações impostas ao modelo, abordadas no item 4.3, inerentes ao SGBD e ao ambiente a ser auditado. Normalmente, esta função é executada uma única vez por banco de dados, a menos que se queira alterar os objetos e os níveis de auditoria previamente selecionados.

5.2.3 Estruturas de auditoria

Será o módulo em que o usuário fornecerá ao sistema os dados para a criação da estrutura-base de armazenamento dos dados auditados (tabela de logs), bem como proceder a uma liberação desses dados, quando necessário.

- **Tabela de logs:** função que tem como objetivo criar a tabela que vai conter todas as informações, objeto do processo de auditoria, e que será alimentada continuamente pelos triggers de auditoria, à medida em que cada objeto, previamente selecionado, tenha um ou mais registros alterados. Normalmente esta função é invocada apenas uma vez por banco de dados auditado. A figura 5.2 procura ilustrar a estrutura da tabela de logs com os valores defaults definidos no módulo "Configuração de Ambiente":

Tabela 5.1 - Uma possível estrutura da Tabela de Logs

Tabela	Campo	Tipo	Tamanho Default	Observação
Tabela de Logs	Identificador do log	Inteiro	4	Chave primária identity
	Tabela	Varchar	40	
	Operação	Char	1	Inclusão, Alteração e Exclusão
	Identificador do Registro	Inteiro	4	
	Campo	Varchar	40	
	Conteúdo Antigo	Varchar	500	
	Usuário	Varchar	20	
	Estação	Varchar	20	
	Data	Smalldatetime	4	

- **Limpeza de dados:** função que tem como objetivo liberar (excluir) os dados auditados contidos na tabela de logs. Essa função normalmente será invocada quando o processo de auditoria se encerrar ou quando houver necessidade de liberação de espaço na estrutura montada.

5.2.4 Monitoramento

Será o módulo em que o usuário criará os triggers de auditoria que vão gerar os registros na tabela de logs, sempre que houver uma operação de inclusão, alteração ou exclusão, nos objetos previamente selecionados. Ainda neste módulo será possível importar e/ou exportar os dados auditados, caso se queira manipulá-los, através de ferramentas de terceiros, no processo de auditoria, ou mesmo armazenar os dados externamente, como uma espécie de backup.

- **Triggers de Auditoria:** função que tem como objetivo gerar os triggers de auditoria que, juntamente com a tabela de logs, irão viabilizar a geração e armazenamento dos dados auditados dos objetos selecionados. O usuário poderá optar pela geração individualizada dos triggers, para monitorar as operações de inclusão, alteração ou exclusão nos objetos e campos-alvos da auditoria;

- **Exportação de dados:** função que tem como objetivo exportar os dados da tabela de logs (em formatos diversos), para manipulação de aplicativos de terceiros ou mesmo como um backup de dados. Essa função deverá ser parametrizada pelos campos da tabela de logs e deverá ter uma opção de manutenção dos dados, na tabela de logs, ou apagá-los , quando da realização da exportação dos dados;

- **Importação de dados:** função que tem como objetivo importar os dados da tabela de logs (em formatos diversos), como uma restauração de dados. Da mesma forma anterior, o sistema deverá disponibilizar um mecanismo de inclusão ou substituição desses registros na tabela alvo de logs, quando da realização da importação desses dados.

5.2.5 Auditoria

Será o módulo em que o usuário obterá os resultados do processo de auditoria, propriamente dito. Além da pesquisa sobre os dados auditados, neste módulo, o usuário alimentará o sistema das informações estatísticas (médias e desvios – padrão) sobre as operações efetuadas em cada tabela auditada, com base nas informações coletadas no monitoramento e armazenados na tabela de logs. Essas estatísticas, então, servirão como argumentos de pesquisa na busca de comportamentos anômalos de transações realizadas por usuários, nas tabelas auditadas. É importante, aqui, frisar que possíveis extrapolações encontradas de médias e desvios-padrão, não implicam, necessariamente, uma fraude ou violação por parte do usuário, mas sim de um comportamento anômalo, que deve ser investigado pelo auditor.

- **Estatísticas-Padrão:** função que tem como objetivo alimentar o sistema das estatísticas de operações efetuadas em cada tabela auditada. O fornecimento dessas estatísticas poderá ser feito de uma forma manual (através da digitação) ou calculada pelo SAUDIT, com base nos dados contidos na tabela de logs. O sistema deverá armazenar, então, para cada tabela auditada, a média, desvio-padrão e a periodicidade por operação de inclusão, alteração ou exclusão, referente ao dado estatístico (diária, mensal ou anual);

- **Consultas Padrão:** função que tem como objetivo disponibilizar consultas pré-formatadas dos dados das operações auditadas, tanto a nível individual, quanto a nível estatístico, mediante o fornecimento de parâmetros pelo usuário. Para as consultas de natureza individual dos registros auditados, deverá ser relacionado o registro existente antes da operação de alteração, com o registro corrente na base de dados. Já para as consultas de natureza estatística, deverá ser relacionado os dados estatísticos pesquisados com os padrões médios esperados, que são previamente armazenados na lista de objetos selecionados para serem auditados;

- **Consultas Específicas:** função que tem como objetivo disponibilizar consultas específicas e não contempladas na função "Consultas Padrão". Basicamente, esta função disponibilizará ao usuário um ambiente livre, onde o mesmo poderá formular sintaxes específicas de consultas SQL na tabela de logs, juntamente com as tabelas auditadas.

5.2.6 Configuração de ambiente

Será o módulo em que o usuário calibrará as variáveis de ambiente do sistema, flexibilizando as limitações impostas pelo modelo proposto, criará as suas tabelas de apoio (objetos auditados, campos auditados e usuários do sistema), bem como cadastrará os usuários do SAUDIT ao nível de aplicação, para efeito de validação, no acesso ao mesmo.

- **Ambiente do Sistema:** função que tem como objetivo calibrar as variáveis de ambiente do sistema, bem como as limitações a serem impostas às estruturas do modelo a ser auditado, como, por exemplo:
    - Caminho de acesso ao banco de dados;
    - Nome da tabela de logs;
    - Nome da tabela de objetos auditados;
    - Nome da tabela de campos auditados;
    - Tamanho máximo do nome da tabela auditada;
    - Número máximo de campos da tabela auditada;
    - Tamanho máximo do nome do campo auditado;
    - Tamanho máximo do conteúdo do campo auditado;

- Tamanho máximo do nome do usuário;

- Tamanho máximo do nome da estação.

Essas variáveis de ambiente deverão conter valores "default", podendo a critério e necessidade do usuário, ser alteradas;

- **Tabelas de Apoio:** função que tem, como objetivo, criar as tabelas de apoio do sistema no modelo a ser auditado: objetos auditados, campos auditados e usuários do sistema. A figura 5.3 procura ilustrar a estrutura de cada tabela de apoio com seus valores defaults.

Tabela 5.2 - Uma possível estrutura das tabelas de apoio

Tabela	Campo	Tipo	Tamanho Default	Observação
Objetos Auditados	Identificador da Tabela	Inteiro	4	Chave primária identity
	Média de Inclusões	Inteiro	4	
	Desvio Padrão de Inclusões	Inteiro	4	
	Periodicidade da Média	Char	1	Diária, Mensal e Anual
	Média de Alterações	Inteiro	4	
	Desvio Padrão de Alterações	Inteiro	4	
	Periodicidade da Média	Char	1	Diária, Mensal e Anual
	Média de Exclusões	Inteiro	4	
	Desvio Padrão de Exclusões	Inteiro	4	
	Periodicidade da Média	Char	1	Diária, Mensal e Anual

Campos Auditados	Identificador da Tabela	Inteiro	4	Chave primária
	Identificador da Coluna	Inteiro	4	Chave primária
Usuários do SAUDIT	Identificador do Usuário	Inteiro	4	
	Usuário	Varchar	20	
	Senha	Varchar	15	
	Nome completo	Varchar	40	
	Endereço	Varchar	40	
	Bairro	Varchar	20	
	Cidade	Varchar	20	
	UF	Char	2	
	CEP	Char	8	
	Telefone de contato	Char	11	

- **Usuários do Sistema:** função que tem como objetivo cadastrar os usuários do SAUDIT, que serão validados, juntamente com os usuários do banco de dados a ser auditado, durante o processo de identificação, na carga do sistema. Por uma questão de segurança, a validação do usuário, no acesso ao sistema, dar-se-á através do fornecimento do usuário, senha e banco de dados que se quer auditar, onde o mesmo deverá estar cadastrado, tanto do nível de aplicativo, como ao nível de banco de dados. Todavia, esta função se limitará a cadastrar o usuário ao nível de aplicativo, ficando o seu cadastro no banco de dados por conta e responsabilidade do DBA, de acordo com os procedimentos da organização, lembrando, apenas, que o usuário a ser cadastrado deverá ter as devidas permissões de criação e alteração de tabelas no banco de dados em questão e que a sua identificação, usuário e senha, obviamente, deverá coincidir nos dois ambientes de verificação.

6 CONCLUSÕES

Ao longo deste trabalho, foi demonstrada, a importância da segurança da informação nos dias atuais, particularmente a importância da auditoria, nas organizações, para os profissionais que lidam com administração de banco de dados. Neste contexto, violações de dados podem acarretar prejuízos de ordem pessoal, material ou até moral para pessoas e/ou instituições. A prática tem demonstrado que os métodos de segurança tradicionalmente utilizados nos SGBDs (validação de usuários através de identificação e fornecimento senhas) nem sempre são suficientes para prevenir violações de dados.

O trabalho apresenta um perfil das empresas e profissionais que realizam auditoria, identificando uma carência de mercado quanto a disponibilização de um método e/ou ferramenta padronizada para realização de auditoria em banco de dados.

Desta forma, buscou-se apontar 5 etapas distintas para um processo de auditoria, que vise o desenvolvimento de uma ferramenta que supra esta demanda.

O trabalho consolidou um modelo, através do qual, pode ser implementado um código padrão para armazenamento, em tempo real, de cada transação realizada no objeto que se queira auditar (inclusão, alteração ou exclusão).

Com base nas 5 etapas diagnosticadas, bem como no modelo proposto, propôs-se um Sistema Gestor de Auditoria (SAUDIT), através da especificação dos seus módulos, onde procurou-se contemplar 4 das 5 etapas de auditoria, além de mecanismos de flexibilização das limitações inerentes ao modelo.

A implementação de um sistema como o SAUDIT, em vários SGBDs relacionais de mercado, constitui-se num importante instrumento de apoio aos profissionais que lidam com processos de auditoria, em bases de dados nas organizações.

No contexto deste trabalho, percebe-se a importância do desenvolvimento de ferramentas que auxiliam as empresas no processo de manutenção de suas informações.

Longe de esgotar o assunto sobre o tema, este trabalho possibilita um embasamento claro, conciso e preciso sobre o assunto. Esperamos que o mesmo sirva como um estímulo para novas proposições, que contribuam no desenvolvimento do estado da arte da Auditoria em Bancos de Dados Relacionais.

Do ponto de vista evolutivo, faz-se necessário uma abordagem semelhante para os bancos de dados orientados à objetos, que tem apresentado um crescimento significativo nos últimos anos.