UNIPAC
UNIVERSIDADE PRESIDENTE ANTÔNIO CARLOS
FACULDADE DE CIÊNCIA DA COMPUTAÇÃO
E COMUNICAÇÃO SOCIAL

CURSO DE CIÊNCIA DA COMPUTAÇÃO

Adriana Diniz dos Santos

SEGURANÇA E INTEGRIDADE EM BANCO DE DADOS

BARBACENA
JULHO DE 2005

Adriana Diniz Dos Santos

SEGURANÇA E INTEGRIDADE EM BANCO DE DADOS

Trabalho de Conclusão de Curso apresentado à Universidade Presidente Antônio Carlos – UNIPAC – Barbacena, como requisito para obtenção do título de bacharel em Ciência da Computação.

ORIENTADOR: Eduardo Macedo Bhering

BARBACENA
JULHO DE 2005

Adriana Diniz Dos Santos

SEGURANÇA E INTEGRIDADE EM BANCO DE DADOS

Trabalho de Conclusão de Curso apresentado à Universidade Presidente Antônio Carlos – UNIPAC – Barbacena, como requisito para obtenção do título de bacharel em Ciência da Computação.

Aprovada em _____ /_____ /_____

BANCA EXAMINADORA

Prof.: Eduardo Macedo Bhering (Orientador)

Prof. M.: Luís Augusto Mattos Mendes

Prof. M.: Gustavo Campos Menezes

Dedico este trabalho aos meus pais José Diniz e Neuza porque as palavras nunca serão suficientes para expressar a gratidão e o respeito que tenho para com aqueles que não só me deram a vida como também orientaram meus passos. E às minhas irmãs Júlia, Lílian e Liliane que me ajudaram a chegar até o fim. Foi por vocês que cheguei até aqui e é por vocês que seguirei sempre em frente!

Agradeço a Deus, ao meu orientador Eduardo Bhering e aos professores Gustavo e Luís Augusto. Agradeço principalmente a Emanuella Maciel, minha grande amiga que me ajudou e tornou tudo isso possível, e que por mais que lhe agradeça será pouco por sua dedicação compreensão nos momentos em que mais precisei. A você grande amiga que ouviu os meus desabafos, que presenciou o meu silêncio, que me acompanhou, chorou, riu, sentiu, participou, aconselhou... Que sofreu, hoje a você ofereço um sorriso, pois na validade de toda a minha luta, nos méritos de minha conquista, há muito da presença de você.

RESUMO

O trabalho apresenta uma visão geral sobre as técnicas e aplicações usadas para garantir a Segurança e Integridade de um Banco de Dados Relacional, utilizando a Linguagem SQL.

A Segurança é a proteção contra a divulgação, alteração ou destruição não autorizada dos dados e a Integridade é a tentativa de garantir que os dados armazenados sejam precisos e corretos, preservando-os assim contra atualizações não válidas.

Palavras Chave: Segurança, Integridade, Banco de Dados Relacional, SQL.

LISTA DE SIGLAS

DBA – Database Administrator ou Administrador de Banco de Dados

DBMS – Database Management Sytem

DDL – Data Definition Language ou Linguagem de Definição de Dados

DML – Data Manipulation Language ou Linguagem de Manipulação de Dados

Linguagem de Consulta Estruturada ou Linguagem de Consulta Padrão

SGBD – Sistema Gerenciador de Banco de Dados

SQL – Structured Query Language ou Standard Query Language

SQL-92 – Standard Database Language (1992) ou Versão Padrão SQL (1992)

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

A Segurança e Integridade tornaram-se uma preocupação imprescindível para qualquer indivíduo encarregado pelo gerenciamento de bases de dados, que armazenam informações de grande importância e que suportam vários usuários.

Garantir que um ambiente de banco de dados seja totalmente seguro não é tarefa fácil, porém com a aplicação de certas técnicas, pode-se chegar a excelentes níveis de segurança e integridade.

A Segurança refere-se a proteção contra a divulgação, alteração ou destruição não autorizadas dos dados. Já a Integridade tenta garantir que os dados armazenados sejam precisos e corretos, preservando-os contra atualizações não válidas.

Com o propósito de abordar tais aspectos, este trabalho de conclusão de curso apresenta uma visão geral sobre as técnicas e aplicações usadas para garantir a Integridade e Segurança em um Banco de Dados Relacional, utilizando a linguagem SQL.

O primeiro capítulo traz conceitos e definições de Sistema de Banco de Dados, Administrador de Banco de Dados, Sistema Gerenciador de Banco de Dados, Modelo Relacional e Linguagem SQL. O segundo e terceiro capítulos descrevem os conceitos, técnicas e aplicações de Segurança e Integridade. No quinto capítulo são apresentados os Mecanismos de Implementação, seguidos de conclusão e referência bibliográfica.

2 GERENCIAMENTO DE BANCO DE DADOS

2.1 – Conceitos e Definições

Um Sistema de Banco de Dados é basicamente um sistema computadorizado cujo propósito é manter as informações e torná-las disponíveis quando solicitadas (DATE, 2000).

Oferecem um grande número de benefícios, onde um dos mais importantes é a Independência Física dos Dados (imunidade de programas aplicativos a alterações no modo de armazenar fisicamente os dados e obter acesso a eles).

Envolve quatro componentes principais: Dados, *Hardware*, *Software* e Usuários (DATE, 1991).

- Dados: são informações armazenadas no banco de dados;
- *Hardware*: são discos magnéticos (para guardar os dados armazenados) associados a outros dispositivos de Entrada/Saída e também a processadores e memória principal que fornecem suporte à execução do *software* do sistema de banco de dados;
- *Software*: é o gerenciador do banco de dados, conhecido como SGBD - Sistema Gerenciador de Banco de Dados ou DBMS – *DataBase Management System*. É ele quem trata todas as solicitações de acesso ao banco de dados por parte dos usuários, como: busca, atualização, remoção e inserção, possibilitando assim, uma visão geral do banco de dados;
- Usuários: os usuários podem ser divididos em programadores de aplicações, usuários finais e Administradores de Banco de Dados - DBA's (responsável pela administração do banco de dados e do sistema de banco de dados de acordo com as normas estabelecidas).

A FIGURA 1 representa uma visão simplificada de um sistema de banco de dados e seus quatro componentes principais.

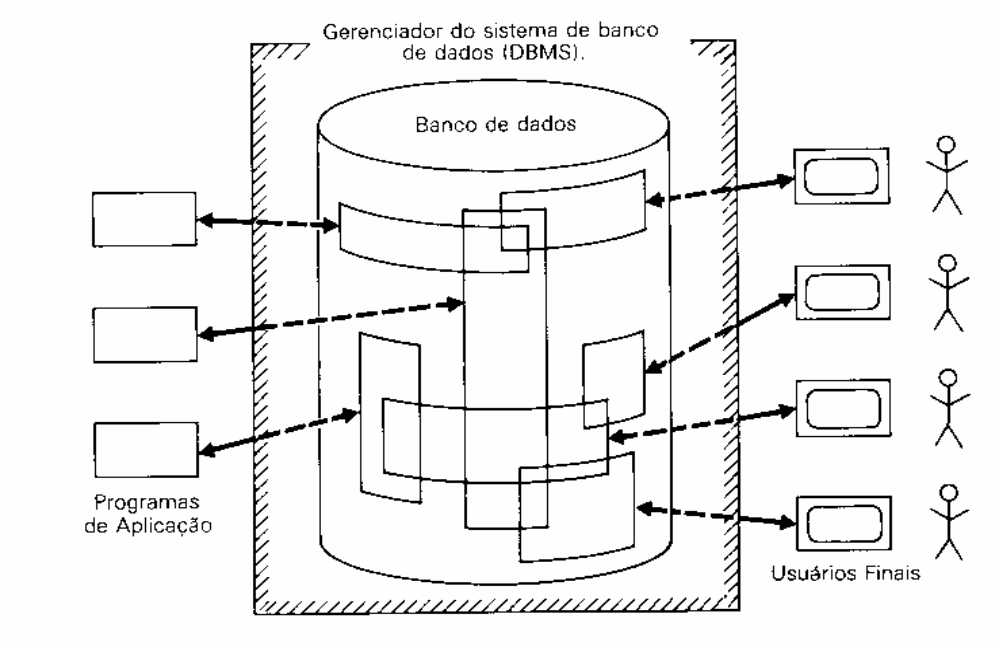

FIGURA 1: Representação Simplificada de um Sistema de Banco de Dados.
FONTE: (DATE, 2000)

O sistema de banco de dados possui diversas vantagens e desvantagens como (DATE, 1988):

- **Vantagens**
    - Rapidez na manipulação e no acesso à informação;
    - Redução do esforço humano;
    - Disponibilização da informação no tempo necessário;
    - Controle integrado de informações distribuídas fisicamente;
    - Redução de redundância e de inconsistência de informações;
    - Compartilhamento de dados;
    - Aplicação automática de restrições de segurança;
    - Redução de problemas de integridade.

- **Desvantagens**

    - Sem dispositivos de controle adequados, a segurança pode ficar comprometida; por exemplo, no caso de acesso não autorizado a dados;
    - A integridade das informações pode ser comprometida se não houver mecanismos de controle; por exemplo no caso de manipulação concorrente de dados;
    - A operação do sistema de banco de dados e o desenvolvimento de aplicações precisam ser feitos com muita precisão para evitar que informações não correspondam à realidade;
    - A administração do sistema de banco de dados pode se tornar muito complexa em ambientes distribuídos, com grande volume de informações manipuladas por uma grande quantidade de usuários.

Os Sistemas de Banco de Dados fundamentam-se em uma série de abordagens diferentes, em particular, os Sistemas Relacionais.

Os Sistemas Relacionais baseiam-se em uma teoria formal chamada Modelo Relacional de Dados, onde os dados são representados por meio de linhas em tabelas. Essas linhas podem ser interpretadas como proposições verdadeiras, possuindo operadores para as operações entre tabelas e linhas (DATE, 2000).

As razões pelas quais os Sistemas de Bancos de Dados Relacionais se tornaram tão dominantes são:

- Admitem a interpretação precedente de dados e banco de dados de forma direta e trivial;
- Possuem uma linguagem padrão para a manipulação de seu banco de dados, a SQL (*Structured Query Language*).

2.2 – Administrador de Banco de Dados e Sistema Gerenciador de Banco de Dados

Um Administrador de Banco de Dados (DBA) é a unidade central que organiza, garante segurança e manutenção do banco de dados. Este controle centralizado possibilita algumas vantagens como (LIMA;RAMOS,2005):

- A redundância pode ser reduzida;
- A inconsistência pode ser evitada;
- Os dados podem ser compartilhados;
- Assegurar que todos os padrões aplicáveis seja observados na representação dos dados;
- Aplicar restrições de segurança;
- Manter a integridade;
- Equilibrar as necessidades conflitantes.

Date (1991), fala que o DBA deve ser o elo de ligação com os usuários do banco de dados, para garantir a disponibilidade dos dados que estes necessitam e, auxiliá-los na preparação dos esquemas externos. Isso garante que a estrutura do banco de dados não seja alterada.

O DBA, detendo toda a autoridade sobre os dados operacionais pode (LIMA;RAMOS,2005):

- Assegurar que os únicos meios de acesso ao banco de dados sejam realizados através de certos canais;
- Definir os controles de segurança a adotar, sempre que for empreendido o acesso a determinados dados especiais;
- Estabelecer diferentes controles para cada tipo de acesso (recuperação, modificação) e para cada parte da informação do banco de dados.

Para desempenhar todas essas funções, o DBA irá precisar de diversos programas utilitários como auxilio às tarefas precedentes. Estes programas estão em sua maioria associados aos softwares conhecidos como SGBD. Abaixo encontramos alguns exemplos dos tipos de utilitários necessários (LIMA;RAMOS,2005):

- Rotinas de Cargas para criar a versão inicial do Banco de Dados;
- Rotinas de Reorganização (para reorganizar o Banco de Dados e reutilizar espaço ocupado por dados obsoletos);
- Rotinas de Controle de Uso (para anotar cada operação feita no Banco de Dados, juntamente com a informação sobre o usuário que realizam e os registros dos estados anteriores e posteriores);
- Rotinas de Recuperação (para restaurar o Banco de Dados a um estado anterior depois de uma falha de hardware ou de programação);
- Rotinas de Análise Estatística (para ajudá-lo na monitoração do desempenho).

O DBA, detendo controle central do banco de dados, pode assegurar que todos os padrões aplicáveis sejam observados na representação dos dados. Esta padronização é importante para facilitar o intercâmbio dos dados ou a migração dos dados entre sistemas.

E para o problema de manutenção da integridade o DBA tenta assegurar que os dados do banco de dados estejam corretos, pois a inconsistência entre duas entradas que pretendem representar um mesmo “dado” pode gerar uma falta de integridade. Este problema pode ocorrer se houver redundância nos dados armazenados. Entretanto, mesmo que ela não exista, o banco de dados ainda pode conter uma informação incorreta (LIMA;RAMOS, 2005).

A grande ferramenta do DBA são os Sistemas Gerenciadores de Banco de Dados (SGBD) e seus utilitários, que oferecem a seus usuários acesso aos dados, ajudando a transformá-los em informações e, permitem criar, atualizar e extrair informações de suas

bases de dados. Estes estão cada vez mais completos e com diversas funcionalidades para tratar as múltiplas necessidades de requisitos dos sistemas (SILBERSCHATZ, 1994).

O SGBD proporciona a interface de usuário ao sistema de banco de dados, pois é o software que manipula todos os acessos ao banco de dados como:

- O usuário emite uma solicitação de acesso, usando uma sublinguagem específica de dados;
- Intercepta a solicitação e analisa-a;
- Inspeciona os esquemas externos para aquele usuário, o mapeamento externo/conceitual correspondente, o esquema conceitual, o mapeamento conceitual/interno e a definição da estrutura de armazenamento;
- Executa as operações necessárias no banco de dados armazenado.

2.3 - Modelo de Dados Relacional

Os Sistemas Relacionais são baseados em uma fundamentação formal ou teórica, chamada Modelo Relacional de Dados, isso significa que é um sistema que possui (DATE, 2000):

- Aspecto Estrutural: os dados são percebidos pelos usuários como tabelas;
- Aspecto Manipulativo: são os operadores disponíveis para manipular tabelas;
- Aspecto de Integridade: tabelas satisfazem a certas restrições de integridade. Está relacionado entre outras coisas com Chaves Primárias e Estrangeiras.

As tabelas são a estrutura lógica em um sistema relacional. Representam uma abstração do modo como os dados estão armazenados fisicamente, ou seja, uma abstração na qual numerosos detalhes do nível de armazenamento, como posicionamento e seqüência de

registro, representações de valores armazenados, entre outros, estão todos ocultos do usuário (SILBERSCHATZ, 1994).

O Modelo de Dados relacional representa os dados contidos em um Banco de Dados através de relações. Estas relações contêm informações sobre as entidades representadas e seus relacionamentos. É claramente baseado no conceito de matrizes, onde as chamadas linhas (das matrizes) seriam os registros e as colunas (das matrizes) seriam os campos. Os nomes das tabelas e dos campos são de fundamental importância para compreender o que está sendo armazenado, onde está sendo armazenado e qual a relação existente entre os dados armazenados (DATE, 1988).

Cada linha de uma relação é chamada de *Tupla* e cada coluna, de *Atributo.* O conjunto de valores possíveis de serem assumidos por um atributo, é intitulado de *Domínio*. O *domínio* consiste de um grupo de valores atômicos a partir dos quais um ou mais atributos retiram seus valores reais.

O esquema de uma relação são os campos (colunas) existentes em uma tabela. Já a instância da relação consiste no conjunto de valores que cada atributo assume em um determinado instante. Portanto, os dados armazenados no Banco de Dados, são formados pelas instâncias das relações (DATE, 1988).

As relações não podem ser duplicadas e a ordem de entrada de dados no Banco de Dados não deverá ter qualquer importância para as relações, no seu tratamento. Os atributos deverão ser atômicos, isto é, não são níveis de novas divisões.

A Chave Primária é o atributo que define um registro, dentre uma coleção de registros; A Chave Secundária possibilitam pesquisas ou ordenações alternativas, ou seja, diferentes da ordem criada a partir da chave primária ou da ordenação natural (física) da tabela; A Chave

Composta é aquela que contém mais de um atributo e a Chave Estrangeira é aquela que permitir a ligação lógica entre uma tabela (onde ela se encontra) com outra na qual ele é chave primária.

Uma característica principal em um banco de dados modelo relacional é que satisfazem o principio da informação, onde todo o conteúdo de informação do banco de dados é representado somente de um modo, ou seja, com valores explícitos em posições de colunas em linhas de tabelas (DATE, 2000).

2.4 - SQL (*Structured Query Language*)

A linguagem SQL é uma linguagem destinada inicialmente à manipulação de banco de dados relacionais, devido a sua simplicidade e facilidade de uso (SILBERSCHATZ, 1994).

Foi desenvolvida originalmente pela IBM *Research* no início da década de 70. Ela foi implementada pela primeira vez em um protótipo chamado *System R*, onde pretendia ser especificamente uma sublinguagem de dados, e reimplementada em numerosos produtos da IBM e de muitos outros fornecedores. A versão em uso do padrão é o SQL-92, pois versões anteriores não possuem alguns recursos de suporte a implementação (DATE, 2000).

Ela se opõe a outras linguagens, sendo uma linguagem Declarativa (Não Procedural), onde o usuário descreve a informação desejada, sem fornecer um procedimento específico para obtê-la.

A SQL representa um conjunto de comandos responsáveis pela definição das tabelas, comandos e atualização dos dados em um SGBD.

Os comandos existentes nesta linguagem são subdivididos em dois grupos (DATE, 2000):

- Comandos DDL (*Data Definition Language*) – fornece uma sintaxe específica para criar, declarar, alterar variáveis de base de dados e variáveis de relação das tabelas e índices de um sistema, fornecendo uma realização concreta dos operadores declarativos do modelo de dados;
- Comandos DML (*Data Manipulation Language*) – Conjunto de comandos responsáveis pela consulta e atualização dos dados armazenados em um banco de dados. Fornece uma realização concreta dos operadores manipulativos do modelo de dados. É responsável pela consulta e atualização dos dados armazenados em um banco de dados.

A Linguagem SQL tem como grande virtude a capacidade de gerenciar índices, sem a necessidade de controle individualizado do índice corrente, algo muito comum nas linguagens de manipulação de dados do tipo registro.

Entretanto, está longe de ser a linguagem relacional perfeita, pois apresenta falhas em um número muito grande de aspectos. Porém não existe nenhum produto no mercado atual que ofereça suporte a todos os detalhes do modelo relacional. E, por ser admitida por quase todos os produtos existentes no mercado, a linguagem SQL tornou-se a linguagem padrão do modelo relacional (DATE, 2000).

3 SEGURANÇA

A segurança consiste em proteger os dados do banco de dados contra a exposição, alteração ou destruição desautorizadas. Para proteger um banco de dados medidas de segurança devem ser tomadas em vários níveis como (DATE, 2000):

- Aspectos legais;
- Controles físicos;
- Questões de política;
- Problemas operacionais;
- Controles de *Hardware*;
- Segurança do Sistema Operacional;
- Questões que são de preocupação específica do próprio Sistema de Bando de Dados.

Existem duas abordagens gerais para a segurança de dados, conhecidas como Controle Discricionário e Controle Mandatário (DATE, 2000).

No Controle Discricionário um dado usuário, terá em geral direitos ou privilégios de acesso diferentes sobre objetos diferentes, havendo poucas limitações, sendo esses esquemas muito flexíveis.

Já no controle mandatário cada objeto de dados é assinalado com certo nível de classificação, e cada usuário recebe um certo nível de liberação. O acesso a um determinado objeto de dados só pode ser feito por usuários com a liberação apropriada. Esses esquemas mandatários tendem a ser assim hierárquicos por natureza e comparativamente rígidos.

3.1 – Identificação e Autenticação

O sistema não pode permitir que qualquer operação seja validada no banco de dados, a menos que o usuário esteja autorizado para realizar a operação em questão.

Para cada usuário, o sistema terá de manter um registro (perfil do usuário), especificando os objetos que o usuário está autorizado a acessar, e as operações que ele está autorizado a realizar sobre esses objetos. Mas antes de acessar o banco de dados, os usuários terão de identificar-se, dirigindo assim, o sistema para o perfil apropriado do usuário (DATE, 1988).

Em geral, o procedimento de identificação/autenticação pode ser repetido quantas vezes se desejar, mas só pode ser repetido até uma vez por operação individual, se o banco de dados contiver informações particularmente sensíveis.

O processo de identificação poderá envolver o fornecimento de um número de operações. Já o processo de autenticação envolve o fornecimento de informações conhecidas apenas para a pessoa que esteja no procedimento de identificação.

3.2 – Regras de Autorização

A segurança possui semelhanças com a Integridade, pois o sistema permiti que regras de autorização (análogas a regras de integridade) sejam expressas em uma linguagem de alto nível, como SQL.

Assim como as regras de integridade, as desautorizações serão compiladas e armazenadas no dicionário do sistema, e uma vez lançadas dentro do sistema, serão cumpridas a partir daquela ocasião.

O compilador das regras de autorização e o mecanismo de cumprimento correspondente

compõem em conjunto o Subsistema de Segurança (DATE, 1991).

Um usuário pode ter várias formas de autorização sobre partes do banco de dados, dentre elas as seguintes (DATE, 1988):

- **Autorização *Read*:** permite leitura, mas não modificações de dados;
- **Autorização *Insert*:** permite inserção de novos dados, mas não modificação de dados existentes;
- **Autorização *Update*:** permite modificação, mas não remoção de dados;
- **Autorização *Delete*:** permite remoção de dados.

Um usuário pode receber todos, nenhum ou uma combinação desses tipos de autorização, mas além das formas de autorização de acesso aos dados, pode ser concedida autorização a um usuário para modificar esquemas do banco de dados, como (DATE, 1988):

- **Autorização *Index*:** permite criação e remoção de índices;
- **Autorização *Resource*:** permite criação de novas relações (tabelas);
- **Autorização *Alteration*:** permite adição ou remoção de atributos em uma relação (tabela);
- **Autorização *Drop*:** permite remoção de relações (tabelas).

A última forma de autorização é dada ao administrador do banco de dados (DBA), porém o nível de autorização dado ao DBA é análoga àquele fornecido a um superusuário de um sistema operacional.

3.3 – Recursos para garantir Segurança

Dois recursos mais ou menos independentes estão envolvidos nesse processo (DATE, 2000):

- Mecanismo de Visão (*View*);
- Subsistema de autorização.

O mecanismo de visão pode ser utilizado para esconder dados que não podem ser vistos, de usuários não autorizados e/ou permitir a visualização de dados resultantes de uma determinada consulta. É uma importante medida de segurança, entretanto sofre certas desvantagens como:

- Pode ser incômodo, no caso em que ao usuário devem ser concedidos diferentes níveis de acesso para subconjuntos diferentes da mesma tabela;
- Quando um registro for inserido ou atualizado não é exigido que eles satisfaçam os tipos de dados já definidos;
- Quaisquer campos da tabela base que não apareçam na visão estarão nulos no novo registro.

A Visão permite que o banco de dados seja dividido em subconjuntos de várias formas, de modo que a informação sensível possa ser oculta a usuários não autorizados. Mas ela não possibilita a especificação das operações que os usuários autorizados possam executar contra tais subconjuntos e não fornecem autorização adicional além daquelas que o usuário que cria a visão já possui.

O subsistema de autorização é de suma importância, pois controla as atualizações no banco de dados, pois caso ocorra adulteração, pode descobrir qual usuário a realizou.

Havendo qualquer suspeita de adulteração, uma auditoria no banco de dados é realizada, essa, tem por finalidade rever as entradas de usuários, para examinar todos os acessos e operações aplicadas no banco de dados durante um certo período. Tendo feito a alteração o

subsistema pode determinar o número da conta utilizada para realizar a adulteração. A auditoria é de grande importância especialmente para instituições bancárias onde há um grande número de atualizações e transações.

O subsistema de autorização permite também que usuários possuidores de direito de acesso específico, seletiva e dinamicamente concedam esses direitos a terceiros (e posteriormente revogue tais direitos, se assim desejar).

Possui dois tipos de mecanismos de segurança que são (DATE, 2000):

- Mecanismo flexível: utilizado para conceder privilégios a usuários como capacidade de acessar arquivo, registros ou campos de dados específicos em modo especificado (tal como ler, inserir, excluir ou autorizar);
- Mecanismo obrigatório: utilizado para impor segurança multinível através de classificação de dados e usuários em várias classes (ou níveis) de segurança implementando a seguir uma política apropriada para a segurança

3.4 – Criptografia

As diversas medidas que um sistema de banco de dados utiliza para impedir o acesso de intrusos às informações podem não ser suficientes para solucionar problemas, o ideal é criptografar os dados (PILLATI, 2005).

A FIGURA 2 mostra o funcionamento da criptografia, onde um algoritmo de criptografia substitui cada caracter de uma palavra original a ser criptografada, pelo próximo caracter do alfabeto correspondente ao caracter da palavra original.

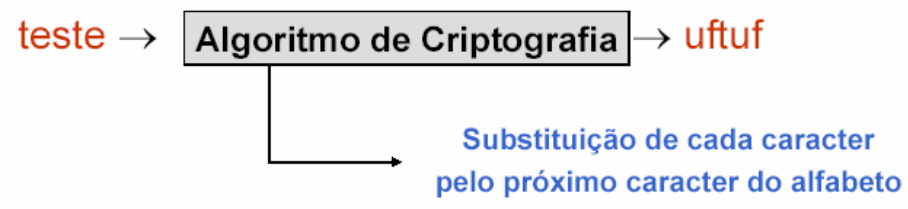

FIGURA 2: Funcionamento da Criptografia
FONTE: (PILLATI, 2005)

Uma boa técnica de criptografia tem as seguintes propriedades:

- É relativamente simples para os usuários autorizados codificar e decifrar os dados;
- O esquema de criptografia não depende do segredo do algoritmo, mas de um parâmetro do algoritmo chamado chave de criptografia;
- É extremamente difícil para um intruso determinar a chave de criptografia.

Existem três tipos de esquemas criptográficos.

- Simétricos
- Assimétricos
- Híbridos

3.4.1 - Criptografia Simétrica

Na criptografia simétrica a chave de codificação é igual a chave de decodificação**.** Ao passar a chave para o receptor deve-se ter cuidado, pois intrusos não devem interceptá-la.

Esse esquema tem como principal vantagem a performance como mostra a FIGURA 3.

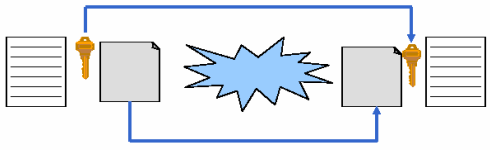

FIGURA 3: Criptografia Simétrica**.**
FONTE: (PILLATI, 2005)

3.4.2 - Criptografia Assimétrica

Na criptografia assimétrica a chave de codificação é diferente da chave de decodificação. Cada participante possui um par de chaves (uma pública e uma privada), onde todos disponibilizam suas chaves públicas e guardam suas chaves privadas.

Quando um usuário quer mandar uma mensagem para outro, a mensagem é codificada com a chave pública do outro. Desta forma, só o outro usuário terá a chave privada para decifrar a mensagem.

Isso garante maior segurança e menor desempenho.

A FIGURA 4 representa perfeitamente a troca de informações entre dois usuários no esquema de criptografia assimétrica.

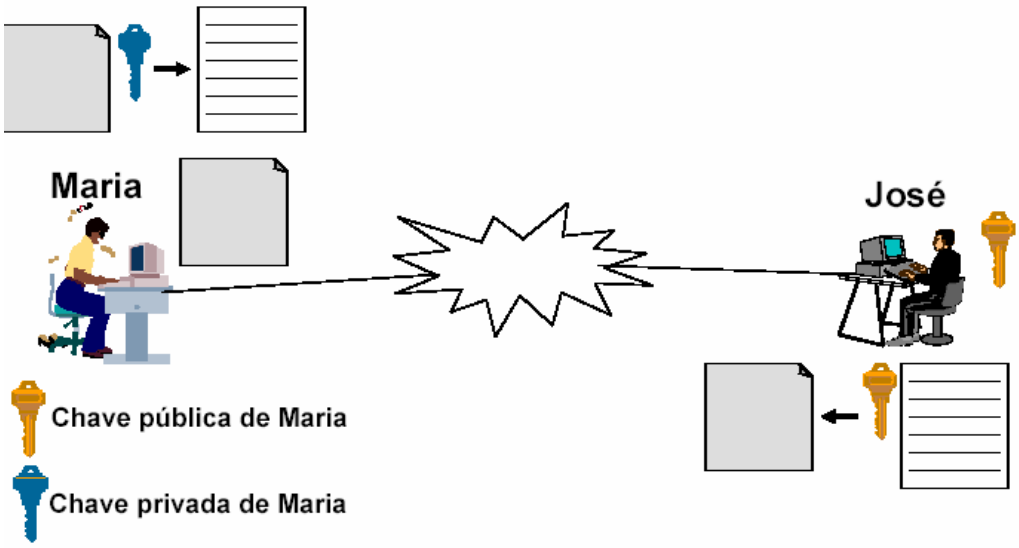

FIGURA 4: Criptografia Assimétrica
FONTE: (PILLATI, 2005)

3.4.3 – Criptografia Híbrida

Na criptografia híbrida os esquemas simétricos e assimétricos são utilizados em conjunto, onde a chave simétrica é usada para codificar a mensagem e a chave pública do receptor da mensagem é usada para codificar a chave simétrica.

Possibilita assim melhor performance e maior segurança.

4 INTEGRIDADE

O termo integridade é utilizado em contextos de banco de dados com o significado de precisão, correção ou validade dos dados. Sua função é assegurar e preservar os dados no banco de dados contra atualizações não-válidas (DATE, 1985).

Atualizações não-válidas podem ser causadas por erros na entrada dos dados; por erros da parte do operador ou do programador da aplicação; por falhas do sistema, e até por falsificação deliberada (entretanto essa não é uma questão de integridade, mas de segurança).

O termo integridade também é usado para referir-se apenas a situação especial que surge num sistema multiusuário. Neste sistema é possível que duas transações sendo executadas concomitantemente, cada uma delas corretas por si mesma, passam a interferir uma na outra de modo a produzir resultados incorretos.

Entretanto, os sistemas que fornecem integridade, tipicamente garantem apenas que tal interferência não possa ocorrer; eles não se preocupam com a questão de as transações individuais estarem ou não corretas, quando consideradas isoladamente, eles só garantem que os erros não serão introduzidos pelo fato de as transações serem processadas ao mesmo tempo (DATE, 1985).

Embora este nível de proteção seja obviamente importante, e de difícil obtenção de confiabilidade, parece conveniente reconhecer que o problema da integridade é mais amplo do que apenas o problema da interferência.

A aplicação da integridade é baseada em regras, onde existe a monitoração de transações (especialmente, operações de atualização) e detecção de violações da integridade e caso ocorra uma violação, tomar a ação apropriada (por exemplo, rejeitar a operação, informar a violação, talvez até corrigir o erro) (DATE, 1988).

As regras são compiladas e armazenadas no dicionário do sistema por um componente

especial do subsistema de integridade, (um compilador das regras de integridade).

Uma vez lançadas no sistema, as regras são então utilizadas a partir daquele ponto. Caso uma nova regra possa ser definida em qualquer ocasião, a definição deve ser rejeitada, se a nova regra for violada pelo estado atual do banco de dados.

Uma abordagem geral á implementação é interrogar as regras de integridade em tempo de compilação e assim incorporar verificações adequadas diretamente dentro do próprio código-objeto das aplicações ao invés de prover um componente de sistema distinto para realizar a verificação em tempo de execução (DATE. 1988).

A principal vantagem é que a validação é tratada pelo sistema, ao invés de ser deixada para as aplicações individuais e as vantagens conseqüentes são que as regras estão concentradas em uma única localização central (dicionário) ao invés de estarem espalhadas pelas aplicações, sendo assim mais fáceis de entender e modificar em sua totalidade. Se necessário também há uma chance melhor de detectar contradições ou redundância dentro dessas regras, com a possibilidade de fazer com que o processo de validação global desempenhe com maior eficiência.

As regras de integridade podem ser divididas em seis categorias amplas, que são descritas abaixo (DATE, 2000).

4.1 – Regras de Integridade

- Regras de Integridade de Tipo ou Domínio: especifica os valores válidos para um determinado tipo (domínio);
- Regras de Integridade de Atributo: especifica os valores válidos para um determinado atributo e nunca deve ser violada;

- Regras de Integridade de Variável de Relação: especifica os valores válidos para uma determinada variável de relação e, é verificada quando essa é atualizada;
- Regras de Integridade do Banco de Dados: especifica os valores válidos para um determinado banco de dados, onde relaciona entre si duas ou mais variáveis de relações distintas;
- Regras de Integridade Referencial: garante que o banco de dados não deva conter quaisquer valores de chaves estrangeiras não associadas;
- Regras de Integridade dos usuários: são regras definidas por usuários de acordo com suas necessidades e especificações do banco de dados.

5 MECANISMOS DE IMPLEMENTAÇÃO

5.1 – Especificação e Regras de Segurança em SQL

O padrão SQL possui comandos para conceder e revogar privilégios como: *delete*, *insert*, *select* (*read*), *update* e *references* (que restringe a declaração de chaves estrangeiras pelo usuário quando cria relações). Permite que usuários criem chaves estrangeiras que se refiram a outras relações (DATE, 2000).

A declaração *grant* é usada para conferir autorização, pois concede aos usuários a autorização *select* sobre a relação. A autorização *update* pode tanto ser conferida a todos os atributos da relação como a apenas um deles.

Por *default*, não é permitido a um usuário que recebeu um privilégio em SQL conceder esse privilégio a outro usuário. Se desejar conceder um privilégio e permitir ao receptor passar esse privilégio a outros usuários, é necessário anexar a condição *with grant option* ao comando *grant* apropriado. Pode ser utilizado o privilégio *all privileges*, como forma abreviada para todos os privilégios e *public*, da mesma forma, para todos os usuários (DATE, 2000).

A declaração *revoke* é usada para revogar uma autorização. A revogação de um privilégio concedido a um usuário pode fazer com que outros usuários também percam esse privilégio. Para impedir a revogação em cascata utiliza-se a opção *restrict*. Neste caso, haverá a emissão de uma mensagem de erro caso haja qualquer revogação em cascata, e a ação de revogar não será implementada.

5.2 – Regras de Segurança em SQL

- Visões: usado para ocultar dados confidenciais de usuários não autorizados (DATE, 2000)

Ex:

Create View LF As
Select F, F#, F.Nome,F.;status F.Cidade
From F
Where F.Cidade= 'Londres';

A visão define os dados sobre os quais a autorização deve ser concedida. A própria concessão é feita por meio da instituição GRANT.

Ex:

Grant Select, Update (FNome, Status), delete
on LF
to Dan, , Misha,

Em seguida, se for concedida um privilégio, a revogação se faz por meio de da declaração *Revoke*.

Sintaxe:

Revoke [Grant option For] <lista.com. virgules de privilégios>
From <lista_com_virgulas de IDS de usuários><objeto><opção>;

5.3 – Regras de Integridade em SQL

- **Restrições de Domínio**: restrição que se aplica a cada coluna definida no domínio em questão (DATE, 2000).

Ex:

Create Domain cor char(6) default '???'
Constraint Valid_Colors
Check(value in
('Vermelho', 'Amarelo', 'Azul, 'Verde', '???));

Então, como *default*, se o usuário inserir uma linha de peça e não fornecer um valor cor para essa linha, o valor '???' será inserido nessa posição. Como alternativa, se o usuário fornecer um valor de cor, mas ele não fizer parte do conjunto válido, a operação falhará e o sistema produzirá um diagnostico apropriado que mencionará a restrição V*alid_Colors* pelo nome.

Porém, em geral o SQL, permite uma restrição de domínio envolve uma expressão booleana de complexidade arbitrária, portanto, os valores válidos para algum domínio D podem depender dos valores presentes no momento em alguma tabela T.

- **Restrições de Tabelas Básicas**

Uma restrição de tabela básica SQL é (DATE, 2000):

- **Uma definição de chave primária**

Sintaxe:

Unique (<lista_com_Virgulas de nomes de colunas>)

ou

Primary Key(<lista_com_virgulas de nomes de colunas>)

Uma <lista_com_virgulas de nomes de colunas> não deve ser vazia em nenhum dos casos. Uma determinada tabela básica pode ter no máximo uma especificação de *Primary Key*, mas qualquer número de especificações *Unique*.

No caso de *Primary Key*, cada coluna especificada é, considerada *Not Null*, mesmo que *Not Null* não seja especificada de forma explicita.

- **Uma definição de chave Estrangeiras:**

Sintaxe:

Foreign Key (< lista_com_virgulas de nomes de colunas>) References < nome da tabela básica>[(lista_com_virgulas de nomes de colunas.]

[on delete < ação referencial>]

on update<ação referencial>]

onde < ação referencia l> é No Action (o default) ou Cascade ou Set Default ou Set Null.

A segunda <lista_com_virgulas de nomes de colunas > é necessária se a chave estrangeira faz referência a uma chave primária.

- **Uma definição de restrição de verificação**

Sintaxe:

Check (< expressão condicional>)

Uma tentativa de criar uma linha r dentro de uma tabela básica T é considerada uma relação a uma restrição de verificação para T se a expressão condicional especificada dentro dessa restrição tiver valor falso para r.

Ex:

Create Table FP
(F# F# Not Null,P#P# Not null,Qde Qde Not null,
Primary Key(F#,P#),
Foreign Key (F#) references F
on delete cascade
on update cascade,
Check <Qde> 0 and Qde<5001));

onde F#,P# e Qde são os domínios definidos F e P são as tabelas.

- **Restrições Assertivas**

São definidas por meio da sintaxe de Create Assertion (DATE, 2000).

Create Assertion <nome de restrição>

Check (<expressão condicional>);

Ex: Nenhum fornecedor com estatus menor que 20 pode fornecer qualquer peça em quantidade maior que 500.

```
Create Assertion IC95 Check
(Not exists (Select* from F,FP
        Where F.Status<20
        And F.F# = FP .F#
        And FP Qde > 500));
```

6 CONCLUSÃO

A evolução das tecnologias de informação tem renovado a preocupação de organizações quanto a Segurança e Integridade de seus bancos de dados, pois permitem que dados relevantes possam ser acessados em qualquer lugar, a qualquer hora e por diversos usuários.

Devido a isso, o mercado tenta oferecer produtos que possam suprir e auxiliar tais necessidades imprescindíveis, investindo a cada dia em alta performance, segurança e integridade.

Porém, para a implementação de um Banco de Dados Relacional, o padrão mais confiável e de fácil manipulação na atualidade é a Linguagem SQL, pois possui regras e restrições que protegem dados contra divulgação, alteração ou destruição e garantem a precisão e validade.

Mas garantir total segurança e integridade em um banco de dados seja ele Relacional, Orientado a Objeto, entre outros, tem sido uma das tarefas mais difíceis desenvolvidas nos últimos tempos.

Sendo assim este trabalho de conclusão de curso descreveu todos os aspectos necessários para que haja maior Segurança e Integridade em Banco de Dados.

Como proposta para trabalhos futuros, Bancos de Dados Relacionais devem ser implementados em Linguagem SQL, onde as regras e restrições de Segurança e Integridade serão aplicadas para a proteção dos dados armazenados.

7 REFERÊNCIAS BIBLIOGRÁFICAS

DATE, C. J. **Introdução a Sistemas de Banco de Dados**. 2ª edição, Rio de Janeiro - Ed.Campus, 1985.

DATE, C. J. **Banco de Dados: Tópicos Avançados**. 2ª reimpressão, Rio de Janeiro – Ed.Campus, 1988.

DATE, C.J. **Introdução a Sistemas de Bancos de Dados**. 4ª edição,Rio de Janeiro – Ed.Campus, 1991.

DATE, C. J. **Introdução a Sistemas de Banco de Dados**. Trad. 7ª edição Americana, Rio de Janeiro - . Ed. Campus, 2000.

LIMA, Eduardo Jorge Lapa; RAMOS, Joel. **Os Desafio do Administrador de Banco de Dados,** 2005. Disponível em: <http://www.miniweb.com.br/Atualidade/Tecnologia/Artigos/adm_bancodedados.html> Acesso em 15/06/2005.

PILLATI, Fábio R. **Segurança e Integridade.** Universidade De Cruz Alta, 2005. Disponível em: <http://dinf.unicruz.edu.br/~pillatt/cursos/2005_1/bd2/> Acesso em 15/06/2005.

KORTH, H.F. e SILBERSCHATZ, A.; **Sistemas de Bancos de Dados**. 2a. edição revisada, Rio de Janeiro - Makron Books, 1994.

GLOSSÁRIO

Administrador de Banco de Dados: é a pessoa que toma as decisões estratégicas e das normas com relação a dados.

All Privileges: comando abreviado para todos os privilégios.

Alteration: comando que permite a adição ou remoção de atributos.

Atributo: corresponde a coluna de uma tabela.

Banco de Dados Relacional: consiste em uma coleção de tabelas, cada uma das quais com um nome único.

Banco de Dados: é uma coleção de dados persistentes utilizados pelos sistemas de aplicações de uma determinada empresa.

Cascade: comando que realiza a exclusão em todas as tabelas filhas que possua o valor da chave que será excluída na tabela pai.

Chave Composta: contém mais de um atributo.

Chave Primária ou Primary Key: define para o banco a coluna que será a chave primária da tabela. A caso ocorra de ter mais de uma coluna deverá ser relacionada com parênteses.

Chave Privada: a chave é mantida em sigilo.

Chave Pública: a chave é aberta para que todos possam ver.

Chave Secundária: possibilita pesquisas ou ordenação alternativas.

Controle Discricionário: o usuário terá direito de acesso (também conhecido como privilégios) a diferentes objetos.

Controle Mandatário: o acesso só poderá ocorrer mediante as liberações apropriadas porque cada objeto possui um nível de classificação e cada usuário recebe um certo nível de liberação.

Create table: comando que cria a estrutura de uma tabela (arquivo) definido as colunas

(campos) e as chaves primárias e estrangeiras existentes.

Criptografia Assimétrica: a chave de codificação é diferente da decodificação aonde cada participante possui um par de chaves.

Criptografia Híbrida: os esquemas simétricos e assimétricos são utilizados em conjunto.

Criptografia Simétrica: a chave de codificação é igual à chave decodificação.

Criptografia: é a ciência de desenvolver e quebrar cifras.

Dados: dados são representados por meio de linhas em tabelas, e essas linhas podem ser interpretadas por proposições verdadeiras.

Data Definition Language: conjunto de comandos que realizam a criação, alteração e a deleção da estrutura das tabelas e índices de um sistema.

Data Manipulation Language: conjunto de comandos responsáveis pela consulta e atualização dos dados armazenados em banco de dados.

Default: comando que não permite o usuário receber privilégio em SQL e conceder privilégio a outro usuário.

Delete: comando que apaga um ou mais grupos de registros em uma tabela.

Domínio: é um conjunto de valores do qual são tomados os valores de atributo específicos de determinadas relações.

Drop: comando que realiza a retirada da coluna especificada na estrutura da tabela.

Foreign Key: define para o banco as colunas que serão chaves estrangeiras, ou campos que são chaves primárias de outras tabelas.

Grant: comando que define os dados os quais a autorização deve ser concedida.

Hardware: é parte física do computador.

Independência Física dos Dados: significa que é a maneira pelo qual os dados são representados fisicamente no meio de armazenamento secundário, bem como a técnica usada para obter acesso a eles.

Index: autoridade para criar um índice na tabela.

Insert: comando para incluir um novo registro em uma tabela.

Instância: corresponde a uma ocorrência dentro de um banco de dados.

Integridade: significa proteger os dados contra usuários autorizados.

Linguagem SQL: é a linguagem padrão para se lidar com banco de dados relacionais e é aceita por quase todos os produtos existentes no mercado.

Mecanismo de visão: pode ser utilizado para ocultar dados confidências de usuários não autorizados.

Mecanismo Flexível: são utilizados para conceder privilégios a usuários como capacidade de acessar arquivo, registros ou campos de dados.

Mecanismo Obrigatório: são utilizados para impor segurança multinível através de classificação de dados e usuários em várias classes (os níveis de segurança implementam a seguir uma política apropriada para segurança da organização).

Modelo Relacional: é uma teoria abstrata de dados que se baseia em certos aspectos matemáticos (teoria do conjunto e na lógica de predicados).

Not Null: comando que no momento da inclusão é obrigatório para que variáveis possuam um conteúdo.

Read: comando que permite leitura, mas não modificação.

References: é necessário para fazer referencia a uma tabela nomeada que especifica uma restrição de integridade.

Resourse: comando exigido para criar uma nova tabela base.

Restrict: comando que não permite exclusão na tabela pai de um registro cuja chave primária exista em alguma tabela filha.

Revoke: comando que faz a revogação de privilégios.

Segurança: significa proteger os dados contra usuários não autorizados.

Select: seleciona um conjunto de registros em uma ou mais tabelas.

Sistema de Banco de Dados: sistema computadorizado que mantém informações e as torna disponíveis quando solicitadas;

Sistemas Relacionais: são baseados em uma fundamentação formal ou teórica chamado modelo relacional de dados.

Software: é a parte lógica do computador.

Subsistema de autorização: permite que o usuário tenha privilégios específicos como conceder de forma seletiva e dinâmica privilégios a outros usuários e possibilita revogar esses privilégios se desejarem.

Tupla: corresponde a uma linha de uma tabela.

Update: comando que atualiza os dados de um ou mais grupo de registros em uma tabela.

With Grant Option: comando onde significa que usuários especificados recebam a concessão de privilégios sobre objetos com autoridade de concessão.