MO410 – Banco de Dados I

1º Semestre - 2005

Prof. Responsável: Geovane Cayres Magalhães

Resumo do Capítulo 21

Segurança

Grupo 13

Aluno	RA
Fabiana Bellette Gil	028671
Marclei Santos Neves	039095

Índice

Índice de Figuras

1. Introdução

Para garantir segurança na modelagem de aplicações de banco de dados, devemos lembrar de considerar três objetivos principais:

- **Sigilo** → Informação não deve ser acessada por usuários não autorizados
- **Integridade** → Apenas usuários autorizados podem modificar dados
- **Disponibilidade** → Usuários autorizados não podem ter acesso bloqueado

Para cumprimento desses objetivos, devemos adotar uma política de segurança que determine quais medidas de segurança devem ser garantidas. Isso implica em estabelecer quais dados serão protegidos e quais usuários terão acesso a eles.
Em seguida, um mecanismo de segurança deve ser utilizado para garantir a política.

Visões também podem ser usadas como recurso de segurança. Uma vez que torna visível apenas os dados apropriados a um grupo de usuários.

Com a utilização de aplicações cujas requisições são provenientes da Internet, a autenticação de usuários tornou-se importante em sistemas de banco de dados.

2. Controle de Acesso

Um Sistema de Gerenciamento de Banco de Dados (SGBD) provê dois mecanismos para controle de acesso.
Discretionary Access Control (DAC) é baseado no conceito de direito de acesso ou privilégio. Um privilégio permite a um usuário acessar algum objeto de uma determinada maneira, por exemplo, leitura ou escrita. O usuário que cria um objeto automaticamente possui todos os privilégios sobre esse objeto. Para conceder ou revogar permissões a outros usuários, o criador do objeto pode utilizar os comandos GRANT e REVOKE respectivamente.
DAC tem algumas fraquezas, entre elas, a possibilidade de um usuário mal intencionado ter acesso a dados sigilosos trapaceando um usuário autorizado.

Mandatory Access Control (MAC) é baseado em políticas globais do sistema que não podem ser alteradas por usuários específicos. Cada objeto do banco de dados é associado a uma classe de segurança, cada usuário é associado a um passe para uma classe de segurança e regras são verificadas na leitura e escrita dos objetos do banco de dados pelos usuários.

2.1. Discretionary Access Control - DAC

SQL suporta Discretionary Access Control através dos comandos GRANT e REVOKE.
GRANT dá privilégios aos usuários.
Sintaxe:

GRANT privilégios ON objeto TO usuários [WITH GRANT OPTION]

Onde:

Objeto – tabela ou visão

Privilégios podem ser:
SELECT – permissão para ler todas as colunas da tabela ou visão especificada em objeto incluindo aquelas adicionadas posteriormente através do comando ALTER TABLE.

INSERT(coluna) – permissão de inserir linhas na coluna citada no parâmetro da tabela especificada em objeto. Se não for restringida a coluna, a permissão é válida para todas as colunas da tabela. Os privilégios UPDATE(coluna) e UPDATE são similares.

DELETE – permissão de apagar linhas da tabela especificada em objeto

REFERENCES(coluna) – permissão de definir chaves estrangeiras que façam referência a coluna especificada em objeto. Quando não determinada coluna, a permissão é válida para todas as colunas, inclusive as criadas posteriormente.

WITH GRANT OPTION – permite que o usuário possa repassar a permissão a outros usuários. Quando um usuário cria uma tabela, ele possui automaticamente todos os possíveis privilégios sobre ela e a permissão de repassá-lo a outros usuários. Se o objeto criado for uma visão, o usuário que a criou terá sobre ela os mesmos privilégios que tem sobre as visões e tabelas utilizadas na visão. Para criar uma visão, é necessário ter, no mínimo, privilégio de seleção em todos os objetos utilizados.

Apenas o dono de um schema pode executar CREATE, ALTER e DROP. Esse tipo de privilégio não pode ser repassado ou revogado.

Assim como os comandos GRANT e REVOKE, visões também são importantes componentes que permitem garantir segurança em um SGBD. Através delas, é possível apresentar informações a usuários, escondendo outras informações que não devem ser acessadas.

Privilégios são concedidos a identificadores de autorização, que podem representar um usuário ou um grupo. Assim, para executar comandos, o usuário primeiramente deve identificar-se e informar sua senha.

Alguns exemplos de GRANT e REVOKE:

Considerando que Joe criou as tabelas Boats, Reserves e Sailors.

GRANT INSERT, DELETE ON Reserves TO Yuppy WITH GRANT OPTION → Yuppy pode inserir e remover dados da tabela Reserves e repassar o privilégio a outros usuários

GRANT SELECT ON Reserves TO Michael
GRANT SELECT ON Sailors TO Michael WITH GRANT OPTION → Michael pode selecionar dados em Reserves e Sailors e repassar o privilégio em Sailors, não em Reserves. Ele também poderia criar uma visão que referenciasse as duas tabelas, mas não poderia repassar o privilégio de seleção sobre a visão a outros usuários.
Michael também poderia criar uma restrição baseada nas informações de Sailors ou Reserves.

```
CREATE TABLE Sneaky (maxrating INTEGER
                     CHECK (maxrating >=
                            (SELECT MAX(S.rating)
                             FROM Sailors)))
```

A cada inserção de novos registros em Sneaky, o maxrating é comparado com o maior valor existente em Sailors. Por esse motivo, Michael precisa ter privilégio de seleção em Sailors.

GRANT UPDATE(rating) ON Sailors TO Leah → Leah pode atualizar a coluna rating em Sailors. Contudo, ela não poderia executar o comando:

```
UPDATE Sailors S
SET S.rating = S.rating -1
```

Pois ela não tem privilégio de seleção na coluna rating.

GRANT REFERENCES(bid) ON Boats TO Bill → Bill pode referenciar bid como chave estrangeira em outra tabela.

Quando não especificada nenhuma coluna ou grupo de colunas no GRANT, o privilégio é válido para todas as colunas do objeto, inclusive as adicionadas posteriormente.

REVOKE é o comando complementar ao GRANT. Com ele pode tanto ser retirado um privilégio ou o GRANT OPTION de um privilégio de um usuário. Sintaxe:

REVOKE [GRANT OPTION FOR] privilégio ON objeto FROM usuários {RESTRICT | CASCADE}|

Nem sempre que um usuário revogar o privilégio de outro fará com que este perca realmente a permissão sobre o objeto, pois ele pode ter recebido o mesmo privilégio múltiplas vezes de outros usuários.

A cláusula CASCADE faz com que um privilégio ou GRANT OPTION concedido a um usuário seja revogado dele e de todos os usuários que receberam o privilégio

exclusivamente dele. Se esses usuários que estão perdendo o privilégio tiverem concedido o privilégio a outros, todos perdem também, desde que não tenham recebido GRANT de outro usuário. O privilégio é considerado abandonado se foi revogado como conseqüência do usuário que concedeu o privilégio tê-lo perdido.

A cláusula RESTRICT faz com que o comando seja recusado se ao retirar o privilégio dos usuários escolhidos em usuários outros privilégios tornem-se abandonados.

Se um usuário receber privilégio múltiplas vezes de um mesmo usuário, basta um único REVOKE para que ele perca o privilégio concedido.

Ao executarmos um GRANT, o SGBD armazena em uma tabela o descritor do privilégio, que contém o concessor do privilégio, a quem foi concedido, qual privilégio foi concedido e se foi com opção de GRANT OPTION.
Ao criar uma tabela ou visão, o usuário criador recebe automaticamente as permissões sobre os objetos. Nesse caso, o descritor de privilégio tem como concessor o System.
Podemos desenhar um gráfico de autorizações para representar o efeito de uma série de GRANT´s, no qual os nós são os usuários e os arcos indicam os privilégios concedidos. Por exemplo:

GRANT SELECT ON Sailors TO Art WITH GRANT OPTION (executed by Joe)
GRANT SELECT ON Sailors TO Bob WITH GRANT OPTION (executed by Art)
GRANT SELECT ON Sailors TO Art WITH GRANT OPTION (executed by Bob)
GRANT SELECT ON Sailors TO Cal WITH GRANT OPTION (executed by Joe)
GRANT SELECT ON Sailors TO Bob WITH GRANT OPTION (executed by Cal)
REVOKE SELECT ON Sailors FROM Art CASCADE (executed by Joe)

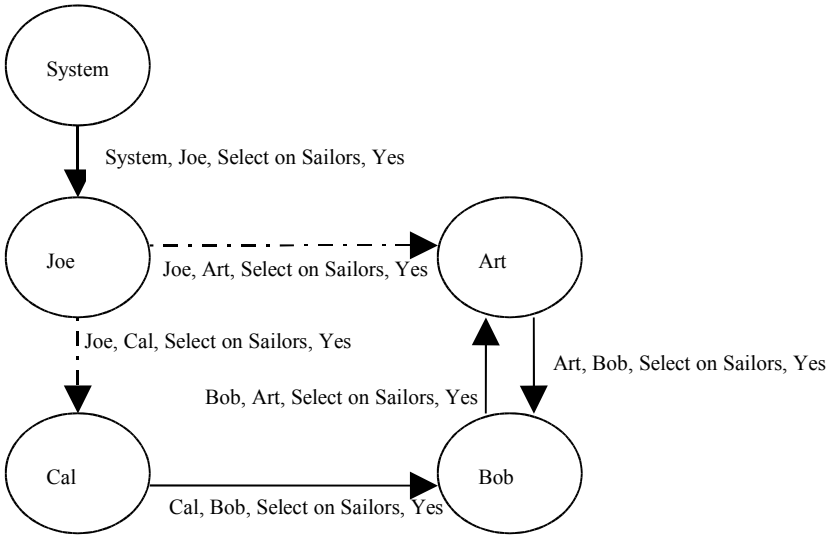

Figura 1 - Exemplo de gráfico de autorizações

O que acontece quando Joe tira a permissão de Art com opção CASCADE?

Os nós restantes seguem a regra: "Se o nó N ainda tem um arco de entrada rotulado com algum privilégio, há um caminho de System até o nó N no qual cada rótulo de arco contém o mesmo privilégio mais o GRANT OPTION".

Art continuará a ter o privilégio pois ainda existe um caminho de System até Art já que ele recebeu o privilégio de Bob. Bob perde o privilégio dado por Art, mas continua com aquele concedido por Cal. Nesse caso, todos os usuários permanecem com privilégio de Select em Sailors.

E se Joe resolvesse revogar o privilégio de Cal?

O arco de Joe até Cal é removido e conseqüentemente de Cal até Bob também. Como não existe caminho de System até os nós Art e Bob, ele são considerados abandonados. Joe é o único usuário que permanece com privilégio.

2.1.1. GRANT E REVOKE em visões e Restrições de Integridade

O privilégio adquirido pelo criador de uma visão varia conforme ele ganha ou perde privilégios nas tabelas utilizadas na visão. Alguns aspectos importantes sobre os comandos GRANT e REVOKE quando eles envolvem visões e restrições de integridade.

1. Uma visão pode ser removida caso usuário que a criou perca o privilégio de SELECT. Remover a visão significa remover sua definição do catálogo do sistema. Se o usuário receber de volta o privilégio sobre a tabela, a visão precisa ser recriada.

2. Se o usuário que criou a visão receber privilégios adicionais sobre as tabelas, automaticamente os ganhará na visão.

3. A distinção entre privilégios de SELECT e REFERENCES é importante. O privilégio de SELECT não dá permissão ao usuário criar chaves estrangeiras para determinado campo. Se um usuário possui privilégio de REFERENCES a um determinado campo e cria uma chave estrangeira em uma tabela, ao perder o privilégio, a chave estrangeira é removida, mas a tabela não.

2.2. Mandatory Access Control - MAC

Discretionary Access Control enquanto geralmente efetivo, possui certas vulnerabilidades, tais como o cavalo de Tróia. Um usuário mal intencionado pode criar uma tabela, dar acesso a um outro usuário (que nem fica sabendo disso) e alterar alguma aplicação deste mesmo usuário para copiar dados sigilosos para esta nova tabela. A modificação de aplicações para tal fim está fora do controle do SGBD. O Mandatory Access Control é projetado para tratar tais falhas no Discretionary Access Control.
O modelo popular para o MAC, chamado Bell-LaPadula, é composto de objetos (tabelas, visões, linhas, colunas), sujeitos (usuários, programas), classes de segurança e passes. Cada objeto do banco de dados é associado a uma classe de segurança e cada sujeito é associado a um passe para uma classe de segurança. Para simplificação serão usadas quatro classes top secret (TS), secret (s), confidential (C) e unclassified (U). Nesse sistema, TS > S > C > U.

O modelo Bell-LaPadula impõe duas restrições em todas as leituras e escritas em objetos:

- **Propriedade Simples de Segurança**: Sujeito S pode ler um objeto O apenas se class (S) >= class (O).
- **Propriedade ***: Sujeito S pode escrever em um objeto O apenas se class (S) <= class (O).

Se o DAC também é especificado, estas regras representam restrições adicionais. Assim, para ler ou escrever em objetos é preciso ter o privilégio necessário (obtido via comandos GRANT) e as classes de segurança dos objetos e dos usuários deve respeitar as restrições de precedência.

2.2.1. Relações Multiníveis e Poli-Instanciação

Implementando-se MAC em um SGBD, cada objeto deve estar associado a uma Classe de Segurança. Os objetos podem estar na granularidade de tabelas, linhas ou colunas. Assumindo que cada linha está associada a uma Classe de Segurança em uma tabela, tem-se então uma tabela Multinível onde cada usuário com diferentes passes vêem um conjunto de linhas diferentes quando acessam a tabela.
Para ilustrar esta propriedade, considere a instância da tabela Boats a seguir:

bid	bname	color	class
101	Salsa	Red	S
102	Pinto	Brown	C

Usuários com passe S e TS conseguem ver ambas as linhas. Um usuário com passe C vê somente a 2ª. linha e um usuário com passe U, não vê nenhuma.
Se um usuário com passe C tenta inserir a linha <101, Pasta, Blue, C>, tem-se um dilema:

- Se a inserção é permitida → violaria a restrição de chave
- Se não for permitida → o usuário conclui que há outro objeto com chave 101 que tem classe de segurança > C.

O problema é resolvido considerando a classe parte da chave. Desta forma a instância da tabela é modificada como mostrado abaixo:

bid	**bname**	**color**	**class**
101	Salsa	Red	S
101	Pasta	Blue	C
102	Pinto	Brown	C

Usuários com passe C vêem apenas as linhas para "Pasta" e "Pinto" mas, usuários com passe S ou TS, vêem todas as linhas. A presença de objetos que aparentam ter valores diferentes para usuários com diferentes passes é chamada de Poli-Instanciação.

3. Segurança para Aplicações de Internet

Quando bancos de dados são acessados de uma localização segura, autenticação baseada em mecanismo de senhas é normalmente adequada. Para acessos de uma rede externa (a compra de um livro pela internet), o estabelecimento de confiança é difícil. Como exemplo, se alguém com o cartão de crédito registrado para Sam quer comprar um livro pela internet, como a Amazon pode ter certeza que ninguém roubou o cartão dele? Como Sam pode ter certeza que o formulário pedindo o número do cartão é realmente uma parte legítima do site da Amazon, e não uma aplicação fraudulenta com o objetivo de obter o número de seu cartão de crédito?
Este exemplo ilustra a necessidade de uma implementação mais sofisticada para a autenticação do que um simples mecanismos de senha. Criptografia é a técnica utilizada para tratar estes problemas.

3.1. Criptografia

A idéia básica por trás da criptografia é mascarar os dados para transmissão. Para isto, aplica-se um algoritmo de criptografia nos dados usando uma chave de criptografia. Tem-se como resultado, uma versão criptografada dos dados. Existe também um algoritmo de descriptografia que usa os dados criptografados e uma chave de descriptografia como entrada e retorna os dados originais. Sem a chave de descriptografia correta, o algoritmo de descriptografia gera dados sem sentido. Em resumo, tem-se:

- Encrypt(dado, chave de criptografia) = dado criptografado
- Decrypt(dado criptografado, chave de descriptografia) = dado original

A figura abaixo ilustra esta idéia.

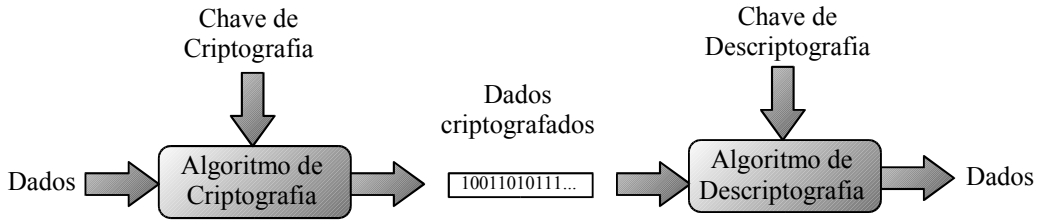

Figura 2 - Criptografia / Descriptografia

Existem dois modos de criptografia: **criptografia simétrica** e **criptografia por chave pública**.
Na criptografia simétrica, a chave de criptografia também é usada para a descriptografia e, neste caso, todos os usuários autorizados sabem a chave (isto é um problema).
Um exemplo de criptografia simétrica é o DES (Data Encryption Standard) que é usado desde 1977 e tem chave de 56 bits. Existe também o AES que tem chaves de 128 bits (opcionalmente, 192 ou 256).
Outro modo de criptografia é a criptografia por chave pública onde cada usuário tem duas chaves sendo uma chave pública para criptografia (conhecida por todos) e uma chave privada para descriptografia (conhecida somente pelo usuário, o que evita o problema do DES). Um exemplo de criptografia por chave pública é o algoritmo RSA que é baseado na observação de que, embora verificar se um número é primo ou não é simples, determinar os fatores primos de um número não-primo é extremamente difícil.
Uma propriedade muito importante dos algoritmos de criptografia e descriptografia é que a função das chaves de criptografia de descriptografia podem ser trocadas, ou seja:

decrypt(d,(encrypt(e,I))) = I = decrypt(e,(encrypt(d,I)))

onde:
e = chave de criptografia
d = chave de descriptografia
I = dado

Uma vez que a função das chaves de criptografia e descriptografia podem ser trocadas, elas são simplesmente chamadas de chave pública e chave privada.

3.2. Certificando Servidores: O Protocolo SSL

Baseado no que foi visto até aqui, voltemos ao exemplo de compra de um livro pela internet. Uma chave e uma chave privada são associada com a Amazon. Qualquer pessoa, digamos Sam, pode enviar um pedido de compra para a Amazon criptografando o pedido usando a chave pública da Amazon. Apenas a Amazon pode descriptografar o pedido de compra pois o algoritmo de descriptografia requer a chave privada da Amazon, conhecida somente pela própria Amazon.

Mas como Sam sabe que a chave pública é realmente da Amazon? O protocolo SSL garante isto. A figura a seguir ilustra este processo.

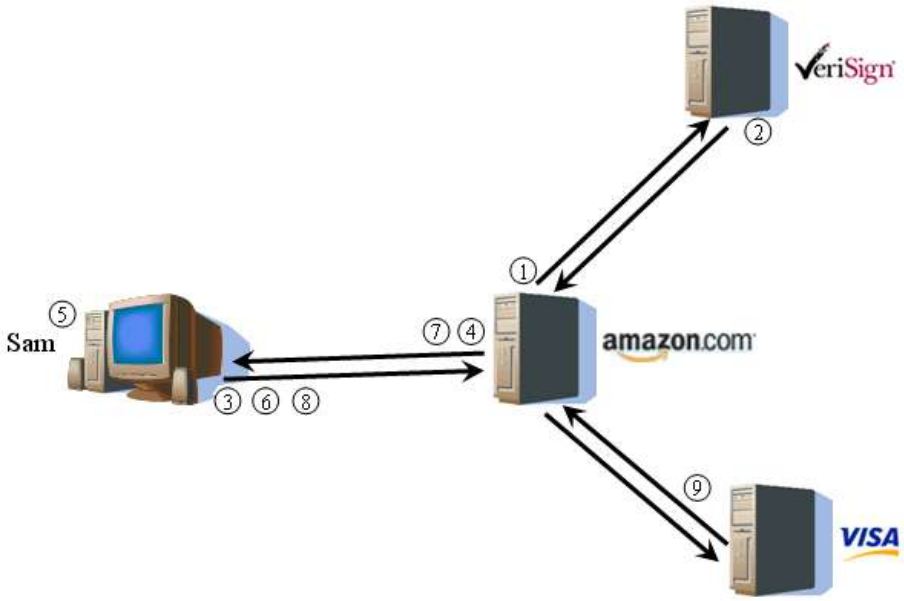

Figura 3 – Protocolo SSL

1) Existem algumas companhias que atuam como **Autoridades de Certificação (por exemplo, Verisign).** A Amazon gera uma chave publica e_A (e uma privada) e pede a Verisign que publique um certificado com a seguinte informação:

<Verisign, AMazon, https://www.amazon.com, e_A>

2) Este certificado é criptografado com a chave privada da Verisign, conhecida somente por ela. A chave pública da Verisign é conhecida por todos os navegadores

3) Sam conecta no site da Amazon. Seu browser (rodando o protocolo SSL) solicita a chave pública deste servidor.

4) O servidor da Amazon envia o certificado para criptografado pela Verisign.

5) O browser descriptografa com a chave publica da Verisign (verificando a autenticidade do certificado) e obtém a chave publica da Amazon.

6) O browser gera uma chave de sessão aleatória criptografada com a chave pública recebida. Só poderá ser descriptografada com a chave privada da Amazon (pode ser feito também um desafio contra o servidor da Amazon).

7) O servidor da Amazon recebe a chave de sessão criptografada e descriptografa com sua chave privada. Esta chave é usada então para criptografar as mensagens subseqüentes.

8) O browser criptografa o número do cartão de crédito de Sam usando a chave pública do servidor da Visa. A Amazon encaminha para o servidor visa e Somente este (que possui a chave privada) consegue descriptografar o número.

9) O servidor Visa aprova a informação e a transação continua.

3.3. Autenticação de Usuários e Assinaturas Digitais

Voltando ao exemplo de compra de um livro pela internet, a Amazon pode utilizar autenticação baseada em senha, isto é, pedir que Sam entre com os dados de sua conta (usuário e senha). A autenticação é feita depois que SSL foi utilizado para estabelecer um id de sessão, portanto a transmissão da senha é segura. Mesmo assim, a Amazon ainda corre o risco de que o cartão de crédito seja roubado e a senha seja "hackeada".
A criptografia de chave pública é utilizada para assegurar a privacidade da informação, porém fornece outra função vital na troca segura da informação eletrônica, a autenticação.
Se a autenticação do remetente é o objetivo mais importante, pode-se reduzir o custo da criptografia usando uma assinatura da mensagem.
Autenticação neste contexto, se refere ao processo em que o receptor de uma mensagem eletrônica seguirá a fim de verificar a integridade da mensagem e também a identidade do remetente. Enquanto a criptografia é utilizada para privacidade, a **assinatura digital** é utilizada para autenticação.
Para criar uma assinatura digital, o assinante aplica uma "one-way function" (esquema hashing, por exemplo), à mensagem original e obtem uma versão resumida da mensagem (hash). Então, utiliza sua chave privada para criptografar o hash. O hash criptografado é a assinatura digital. Se a mensagem for modificada de qualquer maneira, o hash da mensagem modificada será diferente. A assinatura digital é resultado do hash e da chave privada, utilizada para criptografá-la, portanto não pode ser falsificada. A assinatura digital é então incluída na mensagem e ambas são enviadas para o receptor. O receptor recria o hash através da mensagem recebida e então utiliza a chave pública do assinante para descriptografar o hash criptografado incluído na mensagem recebida. Se os dois resultados forem iguais, pode-se concluir que:

- A mensagem realmente veio do assinante uma vez que foi assinada digitalmente por ele, garantindo a autenticidade da origem da mensagem.
- A mensagem não foi modificada, garantindo a integridade da mesma.

4. Segurança em Bancos de Dados Estatísticos

Os bancos de dados estatísticos contêm informações específicas sobre indivíduos ou eventos e são planejado apenas consultas agregadas (ex, média das idades). Por exemplo, se mantivéssemos um banco de dados estatístico com informações sobre marinheiros, seria permitido consultas a média de classificação, idade máxima, etc. Porém, não seria permitido consultas sobre marinheiros individualmente.
Pode ser possível inferir informações protegidas (como a classificação de um marinheiro em especial) a partir de buscas permitidas pelo banco. Por exemplo, se Joe é o navegador mais velho, posso perguntar: "Quantos navegadores tem idade maior que X?" para valores diferentes de X até obtermos 1 como resposta. É possível inferir a idade de Joe. Então, a idéia é permitir somente consultas que envolvam pelo menos um número N razoável de linhas. Este sistema funciona? Provavelmente não pois esta restrição é fácil de ser superada. Veja o exemplo a seguir:
Digamos que Pete deseja saber a idade de Joe. Perguntando repetidas vezes até que o sistema rejeite a consulta: "Quantos marinheiros existem com idade acima de X?" pode-se identificar um conjunto de marinheiros, incluindo Joe. (Seja X = 55 neste ponto). Na seqüência, pergunta-se: "Qual é a soma das idades maiores que X?". Seja S1 este resultado. Próxima pergunta: "Qual é a soma das idades dos marinheiros, exceto Joe, que são maiores que X, mais a idade de Pete?". Seja S2 este resultado. A partir das respostas das duas consultas, como a idade de Pete é conhecida, a idade de Joe é facilmente calculada por: S1 – S2 + (idade de Pete).
Desta forma, é possível perceber que a segurança em bancos de dados estatísticos é difícil de ser assegurada.

5. Conclusão

Projeto de segurança é uma das etapas de um projeto de banco de dados. Os três objetivos a serem considerados na fase de modelagem são sigilo, integridade e disponibilidade.
O administrador é o responsável pela segurança.
Existem dois modelos de segurança em DBMS: DAC e MAC. DAC é baseado em privilégios, disponível na maioria dos DBMS de uso comercial. MAC é baseado em classes de segurança.

6. Referência

Ramakrishnan, R. and Gerhrke, J. (2003). Database Management Systems. McGrawHill, 3rd edition.