Visões e Permissões de Acesso

Carina Friedrich Dorneles
dorneles@inf.ufsc.br

Banco de dados I

Permissões de Acesso

Carina Friedrich Dorneles
dorneles@inf.ufsc.br

Banco de dados I

Autorização de Acesso

- Objetivo
  - proteção contra acessos não autorizados
- Subsistema de Autorização de Acesso (SAA)
  - controla **quais dados** um usuário/grupo de usuários pode ter acesso
  - controla **quais operações** um usuário/grupo de usuários pode realizar sobre estes dados
- Funções do subsistema de autorização
  - **especificação** de autorizações
  - **verificação** de autorizações

Autorização de Acesso

- Cadastro de usuários/grupos
  - *login + password*
- Especificação de autorizações
  - envolve três pontos
    - Agente (**usuário** ou **grupo**)
    - Grânulo (**BD, tabela, atributos, tuplas**, ...)
    - Operação (**select, update**, ...)

Agentes

- DBA
  - superusuário (pode tudo)
  - alguns privilégios são exclusivos dele
    - *recovery* BD, configuração parâmetros do SGBD, ...
  - concede/retira (revoga) privilégios de acesso
- outros agentes
  - todos os privilégios de acesso aos grânulos (BDs e tabelas) que criou
  - concede/revoga privilégios para estes grânulos a outros agentes

Classificação de Autorização de Acesso

- Baseadas no grânulo + operação
  - é ou não válido para todos os usuários
    - permissões públicas ou secretas
- Baseadas nas três dimensões
  - grânulo + operação + agente
  - utiliza matrizes de autorização de acesso
- Baseadas em restrições
  - utiliza visões

Considerações sobre Autorização de Acesso

- Premissa básica
  - ***"quem não consulta não pode atualizar"***
- O que fazer na ocorrência de violações
  - podem ser configuradas pelo DBA
    - mensagens de advertência
    - registro de tentativas
    - bloqueio de acesso
- Administrar corretamente permissões sobre tabelas e visões
  - exemplo
    - não faz sentido uma mesma permissão sobre uma tabela base e uma visão derivada dela

GRANT

GRANT privilegios ON nomeTabela *ou* nomeVisao
TO listaUsuarios *ou* grupoUsuarioLinux ou PUBLIC
[WITH GRANT OPTION]

Ou

GRANT EXECUTE ON PROCEDURE nomeProced
TO listaUsuarios

GRANT

- Privilégios
  - ALL (SELECT, DELETE, INSERT, UPDATE e REFERENCES)
  - SELECT, DELETE, INSERT, UPDATE
  - EXECUTE (GRANT EXECUTE ON PROCEDURE)
  - REFERENCES (para colunas que são chaves estrangeiras)
- Objetos
  - procedure, trigger, tabela ou visão;
- ListaUsuarios, grupoUsuarioLinux
  - Nome de um usuário, ou nome de um grupo (definidos através de roles)
- PUBLIC
  - Fornece acesso a todos os usuários
- [WITH GRANT OPTION]
  - Opcional. Os usuários que aparecem em um comando GRANT com "WITH GRANT OPTION" poderão configurar privilégios a outros usuários.

Usuário vs. ROLES

- Usuário
  - Cria usuário no BD

```
CREATE USER beto WITH PASSWORD ‘abc#02';
```

  - No Postgresql: *Login Roles*

- ROLE

```
CREATE ROLE gerente
```

  - Cria o nome de um papel
  - Podem ser associados a usuários
  - São especificados quando o usuário se conecta

  - No Postgresql: *Group Roles*

Exemplos

GRANT select, insert **ON** medico **TO** User1, User2 **WITH GRANT OPTION;**

GRANT update (email, nome) **ON** paciente **TO** User2;

GRANT select **ON** consulta **TO** PUBLIC;

GRANT ALL ON laboratorio **TO** User1,User2, User4;

GRANT EXECUTE ON PROCEDURE mediaPonderada **TO** User1;

GRANT update **ON** CONSULTA **TO** admCons; /*(admCons é um role)*/
GRANT admCons **TO** pedro, maria;

REVOKE

REVOKE privilegios
ON nomeTabela *ou* nomeVisao
FROM objeto *ou* listaUsuarios *ou*
grupoUsuarioLinux *ou* PUBLIC

Ou
REVOKE EXECUTE ON PROCEDURE nomeProced
FROM objeto *ou* listaUsuarios

Exemplos

revoke all on Ambulatorio from U1, U4

revoke delete on Paciente from U3

revoke select on Medico from PUBLIC

revoke all from U5

Visões

Carina Friedrich Dorneles
dorneles@inf.ufsc.br

Banco de dados I

Visões

- Consultas pré-definidas sobre uma ou mais tabelas
  - Tabela derivada de outra(s) tabela(s), ou de outra(s) visão(s)
  - Construída dinamicamente, previamente analisada e otimizada
    - Portanto, ela está sempre atualizada
    - Alterações nas tabelas base são refletidas automaticamente nas visões
- Também chamadas de **tabelas virtuais**
  - Não existe fisicamente no BD
- Alguns SGBDs suportam **visões materializadas**

Visões

- Objetivos
  - **Segurança**
    - Permissões de acesso limitadas à visão
  - **Simplificar consulta**

CREATE VIEW

```
CREATE VIEW nomeVisao (col1, ..., coln)
AS
consulta
```

- **Parâmetros**
  - *nomeVisao*
    - O nome da visão a ser criada
  - ***consulta***
    - Uma consulta SQL (SELECT FROM WHERE)

Exemplo

- Supondo a tabela:

filme

codigo	***titulo***	***genero***	***anoPublic***
1	titanic	drama	2000
2	X Men	ficcao	2000
3	Spider Man	ficcao	2004
4	Super Man	ficcao	1998
5	Bad Boys	acao	2000
6	Menina de ouro	drama	

Exemplo

- Supondo a tabela:

filme

codigo	***titulo***	***genero***	***anoPublic***
1	titanic	drama	2000
2	X Men	ficcao	2000
3	Spider Man	ficcao	2004
4	Super Man	ficcao	1998
5	Bad Boys	acao	2000
6	Menina de ouro	drama	

- E a seguinte visão:

```
CREATE VIEW ficcao (cod, titulo)
AS
SELECT codigo, titulo
FROM filme
WHERE genero = ‘ficcao’
```

Visão

- Visão gerada

ficcao

2	X Men
3	Spider Man
4	Super Man

- Apenas com os filmes sobre ficção

- Todas as novas consultas que são a respeito de filmes de ficção são feitas na visão
  - a tabela é menor e portanto acesso mais rápido

Deleção

- Deletar uma visão

```
DROP VIEW nomeVisao
```

- Alguns SGBDs aceitam a deleção de várias visões ao mesmo tempo

```
DROP VIEW nome1, nome2,..., nomen
```

Exercícios