MAC0439 – Laboratório de Bancos de Dados

Aula 22

Autorização em SQL

Privilégios

GRANT e REVOKE

Grafo de Autorização

13 de novembro de 2015

Profa. Kelly Rosa Braghetto

(Adaptação dos slides do prof. Jeffrey Ullman, da *Stanford University*)

Autorização em BDs– Objetivos

- Garantir que usuários vejam somente os dados que lhe são permitidos
- Proteger o banco de dados contra modificações feitas por usuários maliciosos

Autorização

- Um sistema de arquivos associa certos privilégios aos objetos (arquivos) que ele gerencia
  - Geralmente: leitura, escrita e execução
- Um sistema de arquivos identifica certos usuários aos quais privilégios são garantidos:
  - Geralmente: o proprietário, um grupo, todos os usuários

Privilégios – (1)

- A SQL mantém um conjunto mais detalhado de privilégios sobre objetos (relações) que os mantidos por um sistema de arquivos tradicional
- São 9 privilégios no total, sendo que alguns deles podem ser restringidos a uma coluna de uma relação
  - **Ex.: SELECT, INSERT, DELETE, UPDATE,** REFERENCES, TRIGGER, EXECUTE, USAGE, ...

Privilégios – (2)

- Alguns privilégios importantes em uma **relação** (tabela ou visão):
- SELECT = direito de acessar (ler) todas as colunas da relação
  - Pode ser aplicado também só a alguns atributos
- INSERT = direito de inserir tuplas
  - Pode ser aplicado também só a alguns atributos
- DELETE = direito de remover tuplas
- UPDATE = direito de modificar tuplas
  - Pode ser aplicado também só a alguns atributos

Exemplo: Privilégios

- Para o comando abaixo:

```
INSERT INTO Refrigerantes(nome)
    SELECT nome_refri FROM Vendas
    WHERE NOT EXISTS
        (SELECT * FROM Refrigerantes
         WHERE nome = nome_refri);
```

Refris que não Aparecem em Refrigerantes. Adicionamo-os em Refrigerantes com fabricante valendo NULL.

- Precisamos dos privilégios SELECT sobre Vendas e Refrigerantes, e INSERT sobre Refrigerantes ou Refrigerantes.nome

Objetos do Banco de Dados

- Os objetos sobre os quais privilégios geralmente se aplicam são as tabelas armazenadas e as visões
- Outros privilégios se referem ao direito de criar objetos de um determinado tipo, por exemplo, *triggers*
- **Visões constituem uma ferramenta importante para o controle de acesso**

Exemplo: Visões como Controle de Acesso

- Podemos não querer conceder o privilégio de SELECT sobre Empregado(nome, endereço, salário)
- Mas é mais seguro conceder a permissão de SELECT sobre:

```
CREATE VIEW EmpregadosSeguros AS
  SELECT nome, endereço FROM Empregados;
```

- Consultas sobre `EmpregadosSeguros` não requerem o privilégio de SELECT sobre `Empregados`, somente sobre `EmpregadosSeguros`

IDs de Autorização

- Um usuário é referenciado por meio de seu *ID de autorização*, geralmente seu login
- Existe um ID de autorização chamado **PUBLIC**
  - Conceder um privilégio ao PUBLIC torna-o disponível a qualquer ID de autorização

Concessão de Privilégios

- Um usuário tem todos os privilégios possíveis sobre um objeto que foi criado por ele mesmo
- Um usuário pode conceder privilégios a outros usuários (IDs de autorização), incluindo ao PUBLIC
- Um usuário pode conceder privilégios **WITH GRANT OPTION**, que permitem que o usuário que recebeu a “concessão” possa, por sua vez, conceder privilégios a terceiros

O Comando GRANT

- Para conceder privilégios, use:

  GRANT <lista de privilégios>

  ON <relação ou outro objeto>

  TO <lista de IDs de autorização>;

- Se quiser que o receptor da concessão possa passar os privilégios para terceiros, adicione ao final do comando:

  WITH GRANT OPTION

Exemplo: GRANT

- Suponha que você é o proprietário de Vendas. Você pode fazer:

```
GRANT SELECT, UPDATE(preço)
ON Vendas TO kelly;
```

- Agora, a Kelly tem o direito de executar qualquer consulta sobre Vendas, mas pode apenas modificar a coluna preço de uma tupla.

Exemplo: Opção “Grant”

- Suponha que concedemos também:

```
GRANT UPDATE ON Vendas TO kelly
WITH GRANT OPTION;
```

- Agora, a Kelly pode, além de modificar qualquer atributo em Vendas, conceder a outros usuários o privilégio UPDATE sobre Vendas.
  - Isso significa também que ela pode conceder privilégios mais específicos, como: `UPDATE(preço)ON Vendas`.

Revogando Privilégios

REVOKE <lista de privilégios>

ON <relação ou outro objeto>

FROM <lista de IDs de autorização>;

- Depois desse comando, a sua concessão desses privilégios não pode mais ser usada por esses usuários para justificar o uso deles do privilégio
  - Mas eles podem continuar tendo o privilégio caso eles também o tenham obtido de algum outro usuário, de maneira independente

Opções do REVOKE

- É possível “complementar” o comando REVOKE com:
  1. CASCADE. Nesse caso, quaisquer concessões feitas pelo revogador não serão mais válidas, não importa o quão longe o privilégio tenha sido repassado
  2. RESTRICT. Nesse caso, se o privilégio tiver sido repassado a outros, o comando REVOKE falhará com uma mensagem avisando que algo mais tem que ser feito para “dar cabo do privilégio propagado.”

Revogando a GRANT OPTION

REVOKE GRANT OPTION

FOR <listas de privilégios>

ON <relação ou outro objeto>

FROM <lista de IDs de autorização>

[CASCADE | RESTRICT];

- Mantém os privilégios atribuídos, mas revoga o direito de conceder esses privilégios a terceiros
- O uso do CASCADE faz com que os privilégios concedidos a terceiros por meio do GRANT OPTION sejam revogados

Grafo de Autorização

- Nós = usuário / privilégio / “grant option”? / é proprietário?
  - UPDATE ON R, UPDATE(a) ON R e UPDATE(b) ON R ficam em nós diferentes
  - SELECT ON R e SELECT ON R WITH GRANT OPTION ficam em nós diferentes
- Aresta $X \rightarrow Y$ significa que o nó *X foi usado para conceder Y*

Notação para Nós

- Use *AP* para o nó representando o ID de autorização *A* e que tem privilégio *P*
  - *P* * = privilégio *P* “with grant option”
  - *P* ** = a fonte do privilégio *P*
    - Ou seja, *A* é o proprietário do objeto no qual *P* é um privilégio
    - Observe que ** implica em “grant option”

Manipulação das Arestas – (1)

- Quando *A concede P* a *B*, desenhamos uma aresta de *AP* * ou *AP* ** para *BP*.
  - Ou para *BP* * se a concessão é “with grant option”
- Mas se *A concede um* subprivilégio *Q* de *P* [como UPDATE(a) on R quando *P é* UPDATE ON R] então a aresta vai para *BQ* or *BQ* *

Manipulação de Arestas – (2)

- Regra fundamental: Um usuário *C* possui o privilégio *Q* somente enquanto existe um caminho entre *XP* ** e *CQ*, *CQ* *, ou *CQ* **, e *P* é um superprivilégio de *Q*.
  - Lembre-se de que *P* poderia ser *Q*, e *X* poderia ser *C*.

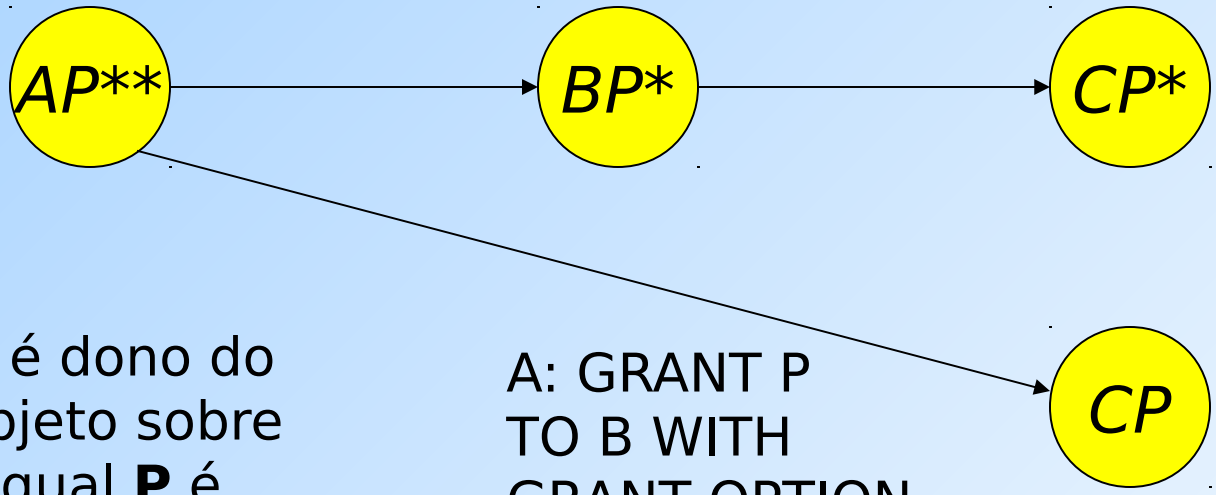
Manipulação de Arestas – (3)

- Se *A* revoga *P* de *B* com a opção CASCADE, remova a aresta de *AP* para *BP*.
- Mas se *A* usa RESTRICT, e existe uma aresta de *BP* para qualquer lugar, então rejeite a revogação e não faça nenhuma mudança no grafo.

Manipulação de Arestas – (4)

- Após a revisão das arestas, precisamos verificar se cada nó possui um caminho de algum nó do tipo ** (representando propriedade) até ele
- Qualquer nó que não possua esse caminho representa um privilégio revogado e deve ser removido do grafo

Exemplo: Grafo de Autorização



B: GRANT P
TO C WITH
GRANT OPTION

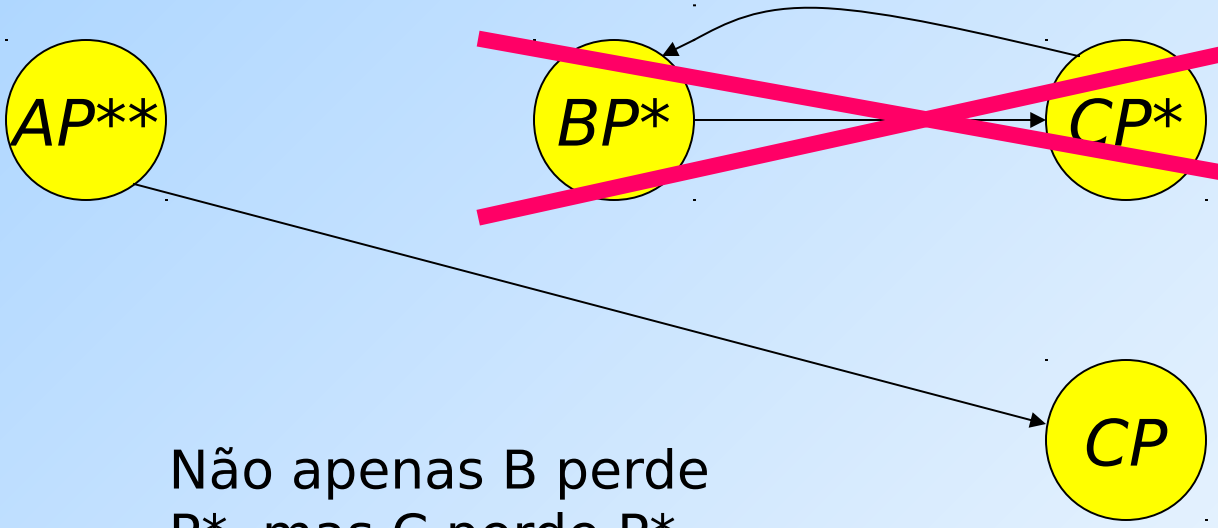
A é dono do objeto sobre o qual **P** é um privilégio

A: GRANT P
TO B WITH
GRANT OPTION

A: GRANT P
TO C

Exemplo: Grant Diagram

A executa
REVOKE P FROM B CASCADE;



Ainda que C tenha passado P para B, ambos os nós serão cortados fora.

Não apenas B perde P*, mas C perde P*. Remova BP* e CP*.

Entretanto, C continua tendo P sem “grant option” por causa da concessão direta.

Para os exercícios desta aula...

- Para que seja possível resolver e testar os exercícios da lista 13, foram criados no PostgreSQL da rede Linux três novos usuários que podem se conectar a qualquer um dos BDs mantidos lá.
- Apesar de poderem se conectar com os BDs, esses usuários ainda não possuem permissão de acesso sobre nenhum objeto dos BDs.
- As identificações desses usuários são:
  - “userglobal1”, “userglobal2” e “userglobal3”
  - A senha de cada um deles é igual ao ID.

Para os exercícios desta aula...

- Se os objetos aos quais você quer conceder acesso a um outro usuário estão dentro de um *schema*, é preciso conceder também ao usuário o privilégio de uso (USAGE) desse *schema*:

GRANT USAGE ON SCHEMA [nome esquema] TO [ID de autorização]

Referências Bibliográficas

- Capítulo 8 do livro “Database Systems – The Complete Book” (1ª edição), Garcia-Molina, Ullman e Widom