Segurança e Controle de Acesso em Banco de Dados

Dominando GRANT e REVOKE para Proteger Seus Dados

UC3 - Aula 10

Ciência de Dados

Outubro de 2025

Tópicos Principais

Usuários e Papéis

Privilégios

Controle de Acesso

Aprenda a:

✓ Criar usuários e papéis

✓ Conceder privilégios com GRANT

✓ Revogar privilégios com REVOKE

✓ Implementar segurança profissional

Abertura e Contexto: Por que Segurança Importa

🔒 "No mundo dos dados, a segurança é a prioridade zero."

Um banco de dados não apenas armazena informações, mas também as protege.

O que é Controle de Acesso?

Controle de Acesso é o mecanismo que garante que **apenas pessoas autorizadas** possam realizar ações específicas em um banco de dados. Ele responde à pergunta: **"Quem pode fazer o quê, onde e quando?"**

Sem controle de acesso, qualquer pessoa com acesso ao banco poderia visualizar, modificar ou deletar dados críticos. Isso é um risco inaceitável para qualquer empresa.

Quando Usar GRANT e REVOKE?

GRANT (Concessão)

Use quando um novo funcionário entra na empresa ou muda de função. Você concede os privilégios necessários para que ele execute seu trabalho.

REVOKE (Revogação)

Use quando um funcionário sai da empresa, muda de função ou quando uma falha de segurança é detectada. Você remove privilégios imediatamente.

Conformidade Legal

Leis como LGPD (Lei Geral de Proteção de Dados) exigem que você controle quem acessa dados pessoais. Sem GRANT/REVOKE, você não consegue cumprir.

Proteção de Dados

O ativo mais valioso de uma empresa é seus dados. Controle de acesso é a primeira linha de defesa contra roubo, vazamento ou corrupção de dados.

Fundamentos: Usuários, Papéis e Privilégios

1. Usuários (Users)

Usuários são **contas individuais** que acessam o banco de dados. Cada pessoa tem um usuário único com uma senha.

Exemplo: `joao_analista`, `maria_gerente`, `carlos_dev`

Criação: `CREATE USER joao WITH PASSWORD 'senha123';`

Características: Identidade única, Autenticação (senha), Rastreabilidade de ações.

2. Papéis / Roles (Grupos de Permissões)

Papéis são **grupos de permissões** que podem ser atribuídos a múltiplos usuários. Simplificam a gestão de acesso.

Exemplo: `leitor`, `editor`, `administrador`

Criação: `CREATE ROLE leitor;`

Vantagem: Em vez de conceder privilégios a 50 analistas individualmente, você concede uma vez ao papel `analista` e atribui o papel a todos.

3. Privilégios (Permissions)

Privilégios são **ações específicas** que um usuário ou papel pode executar. Exemplos: SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, EXECUTE.

DML (Manipulação): `SELECT` (ler), `INSERT` (inserir), `UPDATE` (atualizar), `DELETE` (deletar)

DDL (Definição): `CREATE` (criar), `DROP` (deletar), `ALTER` (modificar)

Outros: `EXECUTE` (executar Stored Procedures)

Tipos de Privilégios em SQL

1. DML (Data Manipulation Language) - Manipulação de Dados

Privilégios para **manipular dados** (ler, inserir, atualizar, deletar) sem alterar a estrutura das tabelas.

`SELECT`	Ler/visualizar dados de uma tabela
`INSERT`	Inserir novas linhas em uma tabela
`UPDATE`	Atualizar dados existentes em uma tabela
`DELETE`	Deletar linhas de uma tabela

Caso de Uso: *Analistas de dados precisam SELECT para ler dados, mas não precisam de INSERT/UPDATE/DELETE. Vendedores precisam de SELECT e INSERT para registrar vendas.*

2. DDL (Data Definition Language) - Definição de Estrutura

Privilégios para **alterar a estrutura** do banco de dados (criar, deletar, modificar tabelas e objetos).

`CREATE`	Criar novas tabelas, índices, views, etc.
`DROP`	Deletar tabelas, índices, views, etc.
`ALTER`	Modificar estrutura de tabelas (adicionar colunas, etc.)

Caso de Uso: *Apenas **administradores e DBAs** devem ter DDL. Funcionários normais NUNCA devem ter CREATE/DROP/ALTER para evitar acidentes.*

3. Outros - Privilégios Especiais

Privilégios para **executar objetos especiais** como Stored Procedures e Functions.

`EXECUTE`	Executar Stored Procedures e Functions

Caso de Uso: *Uma Stored Procedure pode fazer INSERT/UPDATE/DELETE internamente, mas o usuário não precisa ter esses privilégios diretos. Você concede apenas EXECUTE na SP. Exemplo: **SP_RegistrarVenda** pode inserir em Pedidos, mas o vendedor só precisa de EXECUTE.*

GRANT: Conceito e Sintaxe

GRANT é o comando SQL que **concede privilégios** a usuários ou papéis. Ele responde à pergunta: "**Que ações este usuário/papel pode executar?**"

Você usa GRANT quando um novo funcionário entra na empresa ou muda de função. Você especifica exatamente quais privilégios ele precisa para fazer seu trabalho (e nada mais).

Sintaxe Básica do GRANT

```
GRANT privilégio(s) ON objeto TO usuário/papel;
```

privilégio(s): Ação(ões) que será(ão) permitida(s) (SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, EXECUTE, ALL)
ON objeto: Tabela, sequência ou banco de dados onde o privilégio se aplica
TO usuário/papel: Quem recebe o privilégio (usuário individual ou papel/role)

Exemplos Práticos de GRANT

1. Concessão Simples

```
GRANT SELECT ON Clientes
TO leitor;
```

Concede apenas SELECT (leitura) na tabela Clientes ao papel leitor. Leitor pode ler dados, mas não pode modificar.

2. Múltiplos Privilégios

```
GRANT SELECT, INSERT, UPDATE
ON Clientes TO editor;
```

Concede SELECT, INSERT e UPDATE (sem DELETE) ao papel editor. Editor pode ler, inserir e atualizar dados.

3. WITH GRANT OPTION

```
GRANT SELECT ON Clientes
TO editor
WITH GRANT OPTION;
```

Concede SELECT e permite que editor conceda SELECT a outros usuários. Use com cuidado (delegação de permissões).

Atividade Prática 1: Criação de Estrutura de Segurança - Parte 1

Tarefa Principal

Você é o DBA (Administrador de Banco de Dados) de uma empresa. Precisa criar uma **estrutura de segurança profissional** com três papéis (roles) e dois usuários (users) que serão atribuídos a esses papéis.

🎯 Objetivo: Criar 3 Papéis (leitor, editor, administrador) e 2 Usuários (joao, maria)

Passo a Passo Detalhado: Primeiros 3 Passos

PASSO 1

Criar os 3 Papéis (Roles)

Crie três papéis que representam diferentes níveis de acesso:

```
CREATE ROLE leitor;
CREATE ROLE editor;
CREATE ROLE administrador;
```

Resultado esperado: Três papéis criados com sucesso.

PASSO 2

Criar os 2 Usuários (Users)

Crie dois usuários que representam funcionários da empresa:

```
CREATE USER joao
WITH PASSWORD 'senha_joao_123';

CREATE USER maria
WITH PASSWORD 'senha_maria_456';
```

Nota: Use senhas seguras em produção!

PASSO 3

Atribuir Papéis aos Usuários

Atribua cada usuário a um papel:

```
-- João recebe papel LEITOR
GRANT leitor TO joao;

-- Maria recebe papel EDITOR
GRANT editor TO maria;
```

Resultado: João tem papel leitor, Maria tem papel editor.

Próximo slide: Veja a continuação com os Passos 4-6 (Conceder Privilégios aos Papéis, Teste de Acesso, Documentação).

Atividade Prática 2: GRANT - Concessão de Privilégios

Tarefa: Conceder Privilégios aos Papéis

Conceda privilégios específicos aos três papéis em três tabelas: **Clientes**, **Pedidos** e **Produtos**.

LEITOR

- ✓ **SELECT**
- ✗ INSERT
- ✗ UPDATE
- ✗ DELETE

EDITOR

- ✓ **SELECT**
- ✓ **INSERT**
- ✓ **UPDATE**
- ✗ DELETE

ADMINISTRADOR

- ✓ **SELECT**
- ✓ **INSERT**
- ✓ **UPDATE**
- ✓ **DELETE**

Passo a Passo: Primeiros 3 Passos

PASSO 1: Conceder SELECT ao LEITOR

Privilégio de Leitura

Conceda **SELECT** em todas as três tabelas:

```
GRANT SELECT ON Clientes TO leitor;
GRANT SELECT ON Pedidos TO leitor;
GRANT SELECT ON Produtos TO leitor;
```

PASSO 2: Conceder SELECT, INSERT, UPDATE ao EDITOR

Privilégios de Manipulação (sem DELETE)

Conceda **SELECT, INSERT, UPDATE**:

```
GRANT SELECT, INSERT, UPDATE ON Clientes TO editor;
GRANT SELECT, INSERT, UPDATE ON Pedidos TO editor;
GRANT SELECT, INSERT, UPDATE ON Produtos TO editor;
```

PASSO 3: Conceder ALL PRIVILEGES ao ADMINISTRADOR

Privilégios Completos

Conceda **TODOS os privilégios**:

```
GRANT ALL PRIVILEGES ON Clientes TO administrador;
GRANT ALL PRIVILEGES ON Pedidos TO administrador;
GRANT ALL PRIVILEGES ON Produtos TO administrador;
```

REVOKE: Conceito e Sintaxe

O que é REVOKE?

REVOKE é o comando que **remove privilégios** de um usuário ou papel. É o oposto de GRANT. Use REVOKE quando:

- Um funcionário **muda de função** e não precisa mais de certos privilégios
- Um funcionário **sai da empresa** e deve perder todo acesso
- Uma **falha de segurança** é detectada e você precisa remover acesso imediatamente
- Uma **política de segurança muda** e privilégios precisam ser ajustados

Regra de Ouro: **REVOKE deve ser o espelho de GRANT**. Se você concedeu SELECT, INSERT, UPDATE com GRANT, revogue esses mesmos privilégios com REVOKE.

Sintaxe Básica

```
REVOKE privilégio ON objeto FROM usuário/papel;
```

privilégio: SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, EXECUTE, ou ALL
objeto: Tabela, Schema, Sequência, Procedure
FROM: Usuário ou papel que terá o privilégio removido

Revogação Simples

```
REVOKE SELECT
ON Clientes
FROM joao;
```

Remove apenas SELECT. João não consegue mais ler Clientes, mas mantém outros privilégios em outras tabelas.

Revogação Múltipla

```
REVOKE SELECT, INSERT,
UPDATE ON Pedidos
FROM maria;
```

Remove três privilégios de uma vez. Maria perde leitura, inserção e atualização em Pedidos.

Cenários Práticos de REVOKE

1. Mudança de Função

Maria era EDITOR (SELECT, INSERT, UPDATE). Agora é LEITOR (apenas SELECT). Revogue INSERT e UPDATE.

2. Saída de Funcionário

João está saindo. Revogue TODOS os privilégios, remova o papel, e delete o usuário.

3. Falha de Segurança

Detectou-se que um dev não deveria ter acesso a dados de clientes. Revogue SELECT em Clientes imediatamente.

Atividade Prática 3: Revogação de Privilégios (REVOKE)

⚠️ Cenário: Mudança de Política de Segurança

A empresa descobriu que o papel **editor** tinha privilégios excessivos. Especificamente, editores podiam modificar a tabela **TabelaPrecos**, o que é um risco financeiro. A partir de agora, **apenas administradores** podem alterar preços.

> **Problema:** O papel **editor** ainda tem privilégio **UPDATE** em **TabelaPrecos**. Você precisa remover esse privilégio imediatamente para cumprir a nova política de segurança.

Passo a Passo: Revogação e Teste

PASSO 1

Revogar UPDATE do Papel EDITOR

Execute este comando como administrador para remover o privilégio UPDATE:

```
REVOKE UPDATE ON TabelaPrecos
FROM editor;
```

Resultado: O papel **editor** perde a capacidade de fazer UPDATE em **TabelaPrecos**.

PASSO 2

Testar Acesso de Leitura (Deve Funcionar)

Conecte como um usuário com papel **editor** (ex: maria) e tente ler dados:

```
-- Conectar como maria (papel editor)
SELECT * FROM TabelaPrecos;
```

Resultado esperado: ✓ **Sucesso** - Maria consegue ler preços (SELECT ainda é permitido).

PASSO 3

Testar Acesso de Escrita (Deve Falhar)

Ainda como maria, tente atualizar um preço:

```
-- Conectar como maria (papel editor)
UPDATE TabelaPrecos
SET preco = 99.99
WHERE id = 1;
```

Resultado esperado: ✗ **Erro** - "permission denied for relation tabelaprecos"

Validação e Conclusão

Checklist de Sucesso

- ✓ REVOKE UPDATE foi executado sem erros
- ✓ SELECT em TabelaPrecos funciona para editor
- ✓ UPDATE em TabelaPrecos falha para editor
- ✓ Administrador ainda consegue fazer UPDATE

Documentação

Registre em um arquivo a mudança de política:

```
-- Data: 24/10/2025
-- Mudança: Remover UPDATE em TabelaPrecos do papel editor
-- Motivo: Política de segurança financeira
REVOKE UPDATE ON TabelaPrecos FROM editor;
```

Boas Práticas e Consolidação Final

Três Pilares das Boas Práticas de Segurança

1. Princípio do Mínimo Privilégio

Conceda **apenas** os privilégios necessários para que o usuário execute seu trabalho. Nada mais, nada menos.

- ✓ Analista: SELECT
- ✓ Vendedor: SELECT, INSERT
- ✓ Admin: ALL

2. Auditoria de Acesso

Mantenha **registros** de quem acessa o quê, quando e por quê. Isso ajuda a detectar atividades suspeitas.

- ✓ Logs de login
- ✓ Histórico de queries
- ✓ Rastreamento de mudanças

3. Conformidade Legal (LGPD)

Cumpra leis como LGPD (Lei Geral de Proteção de Dados). Controle quem acessa dados pessoais.

- ✓ Consentimento
- ✓ Direito ao esquecimento
- ✓ Transparência

Checklist de Validação: Sua Estrutura de Segurança Está Pronta?

- ✓ Todos os usuários têm senhas fortes e únicas?
- ✓ Ninguém tem privilégios desnecessários?
- ✓ Logs de acesso estão sendo registrados?
- ✓ Você testou a revogação de privilégios?
- ✓ Cada papel tem privilégios bem definidos?
- ✓ Há um processo para revogar acesso rapidamente?
- ✓ A documentação está atualizada?
- ✓ Há um plano de resposta a incidentes de segurança?

Conclusão: Por que Tudo Isso Importa

Segurança não é um luxo, é uma necessidade. O controle de acesso com GRANT e REVOKE é a base de qualquer sistema de banco de dados profissional. Sem ele, você não consegue:

- **Proteger dados sensíveis** (clientes, financeiro, saúde)
- **Cumprir leis** como LGPD, HIPAA, SOX
- **Detectar atividades suspeitas** (auditoria)
- **Responder a incidentes** de forma rápida e eficaz

Você agora sabe como criar uma estrutura de segurança sólida. Use esse conhecimento para proteger os dados mais valiosos da sua empresa: **as informações de seus clientes.**