Laboratório Guiado: GRANT na Prática

Validação de Permissões RBAC e Princípio do Mínimo Privilégio

Transformando a Política de Segurança em Código SQL Funcional

Abertura: Revisão do PoLP e Cenário

Revisão do Princípio do Mínimo Privilégio (PoLP)

Princípio do Mínimo Privilégio (PoLP)

Conceder apenas os privilégios **exatos** necessários para cada função. Ninguém deve ter mais acesso do que o estritamente necessário para realizar seu trabalho.

Revisão do RBAC

- **RBAC**: Gerenciar **Papéis (Roles)** em vez de usuários individuais para eficiência e consistência.
- **Objetivo do Lab**: Traduzir a política de segurança (RBAC + PoLP) em código SQL funcional.

Cenário Prático: Loja Virtual (Assumido da Aula 03)

Tabelas Críticas

- ✓ **Clientes**
- ✓ **Vendas**
- ✓ **Produtos**

Papéis e Usuários de Teste

Papel (Role)	Usuário de Teste	Risco
leitor_relatorios	João	Baixo
operador_vendas	Maria	Médio/Alto
administrador_bd	Pedro	Crítico

Bloco 2: Laboratório Guiado - Concessão de Privilégios (GRANT)

</> Instruções: Execução pelo Administrador

Neste bloco, as duplas devem executar os comandos **GRANT** na sessão de um usuário administrador (Pedro) para traduzir a política de segurança em código funcional.

1. Leitor de Relatórios

Conceder apenas **SELECT** nas tabelas principais (Baixo Risco).

2. Operador de Vendas

Conceder **SELECT, INSERT, UPDATE** em Vendas e **SELECT** em Produtos (Médio/Alto Risco).

3. Administrador BD

Conceder **ALL PRIVILEGES** e discutir o risco inerente (Risco Crítico).

GRANT: Papel Leitor de Relatórios (Baixo Risco)

Papel: leitor_relatorios (Usuário: João)

Objetivo: Consultar dados para relatórios e análises. Risco: Baixo

Roteiro de Concessão (Executar como Administrador)

- Conceder permissão de **SELECT** na tabela **Clientes**.
- Conceder permissão de **SELECT** na tabela **Vendas**.
- Conceder permissão de **SELECT** na tabela **Produtos**.

```
-- Conceder SELECT nas tabelas principais
GRANT SELECT ON nome_do_banco.Clientes TO 'leitor_relatorios'@'%';
GRANT SELECT ON nome_do_banco.Vendas TO 'leitor_relatorios'@'%';
GRANT SELECT ON nome_do_banco.Produtos TO 'leitor_relatorios'@'%';
```

Justificativa do PoLP

Por que apenas SELECT?

O leitor de relatórios não precisa alterar, inserir ou deletar dados. Conceder qualquer outra permissão violaria o Princípio do Mínimo Privilégio (PoLP) e aumentaria o risco de segurança sem necessidade.

GRANT: Papel Operador de Vendas (Médio/Alto Risco)

Papel: operador_vendas (Maria)

Médio/Alto Risco

Tabela Vendas

```
GRANT SELECT, INSERT, UPDATE
ON nome_do_banco.Vendas
TO 'operador_vendas'@'%';
```

Justificativa (PoLP)

Permite registrar novas vendas (INSERT) e corrigir erros de lançamento (UPDATE). O SELECT é necessário para visualizar o histórico de vendas. **DELETE é negado** para proteger a integridade dos dados.

Tabela Produtos

```
GRANT SELECT
ON nome_do_banco.Produtos
TO 'operador_vendas'@'%';
```

Justificativa (PoLP)

Permite consultar preço e estoque antes de registrar uma venda (SELECT). **INSERT/UPDATE/DELETE são negados** para evitar que o operador altere o catálogo ou o estoque diretamente.

GRANT: Papel Administrador BD (Risco Crítico)

⚠ Objetivo: Conceder Controle Total (DDL e DML)

Este papel possui o **Risco Crítico** mais alto. Deve ser atribuído apenas a DBAs confiáveis. Ele tem permissão para **destruir** o banco de dados (DROP TABLE, DELETE FROM sem WHERE).

COMANDO GRANT

```
GRANT ALL PRIVILEGES ON nome_do_banco.* TO 'administrador_bd'@'%' WITH GRANT OPTION;
```

Risco do GRANT ALL

O `GRANT ALL PRIVILEGES` concede todas as permissões DDL (CREATE, ALTER, DROP) e DML (SELECT, INSERT, UPDATE, DELETE). **Regra de Ouro**: Nunca conceda este privilégio a usuários que não sejam DBAs ou administradores de segurança.

Cláusula WITH GRANT OPTION

Esta cláusula permite que o `administrador_bd` conceda as permissões que ele possui a outros usuários. **Risco**: Um administrador mal-intencionado pode criar um novo usuário com controle total. Use com extrema cautela.

Bloco 3: Validação com Queries de Teste

Instruções: Testando a Política de Segurança

Neste bloco, as duplas devem testar a eficácia da política de segurança implementada no Bloco 2. O foco é garantir que o **Princípio do Mínimo Privilégio (PoLP)** foi respeitado.

1. Logar como Usuário de Teste

Logar separadamente como João (Leitor), Maria (Operador) e Pedro (Admin).

2. Executar Roteiro de Teste

Executar as queries de teste para verificar se as permissões foram concedidas e negadas corretamente.

3. Registrar Erros e Soluções

Documentar qualquer erro encontrado e a solução aplicada. Isso é a documentação de segurança.

Roteiro de Teste: Leitor e Operador de Vendas

👤 Usuário: João (leitor_relatorios)

1 Teste: Tentar Inserir Venda (Escrita)

Objetivo: Testar se a permissão de escrita (INSERT) foi negada.

```
INSERT INTO Vendas (...) VALUES (...);
```

❌ ERRO - Acesso Negado (Leitor não tem INSERT)

2 Teste: Consultar Vendas (Leitura)

Objetivo: Testar se a permissão de leitura (SELECT) foi concedida.

```
SELECT * FROM Vendas;
```

✓ SUCESSO - Leitor pode consultar dados para relatórios

👔 Usuário: Maria (operador_vendas)

3 Teste: Tentar Excluir Venda (Alto Risco)

Objetivo: Testar se a permissão de exclusão (DELETE) foi negada.

```
DELETE FROM Vendas WHERE id = 10;
```

❌ ERRO - Acesso Negado (Operador não tem DELETE)

4 Teste: Corrigir Venda (Atualização)

Objetivo: Testar se a permissão de correção/atualização (UPDATE) foi concedida.

```
UPDATE Vendas SET valor = 100 WHERE id = 5;
```

✓ SUCESSO - Operador pode corrigir erros de vendas

Roteiro de Teste: Administrador e Erros Comuns

🔐 Usuário: Pedro (administrador_bd) - Teste DDL

Teste: Tentar Excluir Tabela (DROP)

```
-- Logado como Pedro (Admin)
DROP TABLE Produtos;
```

✓ SUCESSO - Administrador pode executar comandos DDL

Interpretação: O `administrador_bd` tem controle total, incluindo a capacidade de destruir dados.

⚠️ Erros Comuns no Laboratório e Soluções

Erro 1: Acesso Negado (Usuário)

`Access denied for user 'joao'@'%' to database 'nome_do_banco'`

Solução: O usuário não foi criado corretamente ou não foi atribuído ao papel. Verificar `CREATE USER` e `GRANT papel TO usuario`.

Erro 2: Erro de Sintaxe no GRANT

`You have an error in your SQL syntax...`

Solução: Revisar a sintaxe do comando `GRANT`. O MySQL é sensível a aspas, nomes de banco/tabela e a cláusula `TO 'usuario'@'%'`.

Erro 3: SELECT Falhou (Tabela Errada)

`Table 'nome_do_banco.Clientes' doesn't exist`

Solução: O `SELECT` falhou porque a permissão foi concedida apenas para uma tabela, mas o teste tentou fazer `SELECT` em outra. Verificar a tabela correta.

Debate e Consolidação: Análise da Prática

Q1 Qual foi a permissão mais difícil de restringir e por quê?

RESPOSTA CHAVE

Geralmente, restringir o **SELECT** em uma tabela específica dentro de um `GRANT ALL` em um banco de dados é complexo. O PoLP exige granularidade, e o `GRANT ALL` é o oposto.

ANÁLISE

A dificuldade reside em garantir que o usuário tenha acesso a **TODOS** os dados necessários para o trabalho, mas **NENHUM** dado a mais. A restrição deve ser cirúrgica.

Q2 Houve algum momento em que o PoLP atrapalhou a função do usuário? Como ajustar?

RESPOSTA CHAVE

Sim, quando o Operador de Vendas precisava ver o estoque (SELECT em Produtos), mas o administrador concedeu `ALL` em Vendas e esqueceu do `SELECT` em Produtos.

AJUSTE SEM VIOLAR O POLP

Ajustar a permissão concedendo o `SELECT` específico: `GRANT SELECT ON Produtos TO operador_vendas`. Isso resolve a usabilidade sem dar privilégios desnecessários (como UPDATE/DELETE em Produtos).

Q3 Qual a importância de registrar os erros e soluções encontrados durante o teste?

RESPOSTA CHAVE

O roteiro preenchido com erros e soluções é a **documentação de segurança mais valiosa**. Ele prova que a política foi testada, validada e que os riscos foram mitigados.

IMPLICAÇÕES

Em auditorias de segurança, o auditor não quer apenas ver o código `GRANT`. Ele quer ver o **teste** que prova que o `REVOKE` e o `DENY` funcionam. O registro de erros é a prova do teste.

Conclusão: Segurança Exige Testes Rigorosos

Testes Rigorosos são Inegociáveis

A política de segurança só é válida se for testada. O laboratório de hoje provou que o código SQL pode falhar ou conceder permissões indevidas se não for validado.

→ **PoLP Validado**: Garantimos que o Mínimo Privilégio foi respeitado em todos os papéis.
→ **Erros Esperados**: Acesso negado é um SUCESSO da política de segurança.

O Roteiro é sua Documentação de Segurança

O roteiro preenchido com os comandos GRANT, os testes executados e o registro de erros e soluções é a documentação de segurança mais valiosa.

→ **Prova de Conformidade**: Demonstra que a política foi implementada e testada.
→ **Base para Auditoria**: Facilita a verificação por auditores internos e externos.

Próximos Passos da Disciplina

→ **Próxima Aula**: Analisar o fluxo de dados e sua transformação (Pipeline).
→ **Foco**: Como os dados se movem do ponto A ao ponto B e como garantir a segurança nesse trânsito.

A Segurança de Dados é um Processo Contínuo de Teste e Validação

Continue aplicando o Princípio do Mínimo Privilégio em todos os seus projetos.