INSTITUTO SUPERIOR DO LITORAL DO PARANÁ

Prof. Luiz Efigênio

Exame – 2º Bimestre
Disciplina: Administração de Banco de Dados II
Professor: Luiz Carlos Efigênio

- Compreender a importância da segurança em bancos de dados.
- Identificar como funcionam usuários, permissões e roles.
- Aplicar boas práticas de controle de acesso e privilégios mínimos.
- Entender o papel do DBA e da auditoria na segurança de dados.

1. Introdução à Segurança em Bancos de Dados

A segurança em bancos de dados é um **conjunto de práticas e políticas** voltadas para proteger as informações armazenadas contra acessos não autorizados, modificações indevidas ou perdas.
Ela é essencial em qualquer ambiente corporativo, pois o banco de dados contém o **ativo mais valioso da empresa: a informação**.

Os principais objetivos da segurança de dados são:

- **Confidencialidade:** garantir que apenas pessoas autorizadas acessem os dados.
- **Integridade:** assegurar que os dados não sejam alterados de forma indevida.
- **Disponibilidade:** manter o banco acessível e funcional quando necessário.

---

2. Usuários no Banco de Dados

Cada usuário no banco de dados representa uma **entidade que pode executar ações** (consultar, inserir, atualizar, deletar).

Em sistemas como **MySQL, PostgreSQL ou SQL Server**, o usuário possui:

- Um **nome de login** (credencial);
- Uma **senha**;
- E um **conjunto de permissões** que define o que ele pode ou não fazer.

Exemplo em **MySQL**:

```sql
CREATE USER 'analista'@'localhost' IDENTIFIED BY 'senhaSegura123';
```

Esse comando cria um usuário local chamado **`analista`**.
Por padrão, ele **não tem permissão** para fazer nada até que o administrador conceda privilégios.

---

3. Permissões e Privilégios

Permissões (ou privilégios) definem **quais ações o usuário pode executar** sobre um banco ou tabela.

Exemplo:

```sql
GRANT SELECT, INSERT ON logistica.pedidos TO 'analista'@'localhost';
```

- `SELECT` → permite consultar dados.
- `INSERT` → permite inserir novos registros.
- `ON logistica.pedidos` → indica a tabela e o banco onde as permissões se aplicam.

Para revogar:

```
REVOKE INSERT ON logistica.pedidos FROM 'analista'@'localhost';
```

Essas permissões podem ser atribuídas em diferentes níveis:

- **Global:** afetam todos os bancos;
- **Banco específico:** apenas um banco;
- **Tabela ou coluna:** granularidade máxima.

4. Roles (Papéis)

Um **role** é um grupo de permissões que pode ser atribuído a vários usuários.
Ele simplifica o gerenciamento, pois evita a necessidade de configurar permissões manualmente para cada usuário.

Exemplo em **PostgreSQL**:

```sql
CREATE ROLE gestor;
GRANT SELECT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO gestor;
GRANT gestor TO luiz, maria;
```

➡️ O role `gestor` tem permissões de leitura, atualização e exclusão.
➡️ Os usuários `luiz` e `maria` herdam automaticamente essas permissões.

Essa abordagem é essencial em empresas grandes, pois facilita o controle de acesso baseado em **função (RBAC – Role-Based Access Control)**.

---

5. Boas Práticas de Segurança em Bancos de Dados

1. **Princípio do menor privilégio:** conceder apenas o necessário para cada usuário.
2. **Separar funções:** administradores, desenvolvedores e usuários comuns devem ter perfis distintos.
3. **Auditar acessos:** manter logs de quem acessou, quando e o que foi alterado.
4. **Usar senhas fortes:** combinar letras maiúsculas, minúsculas, números e caracteres especiais.
5. **Evitar acesso direto ao banco:** preferir APIs ou sistemas intermediários.
6. **Aplicar criptografia:** tanto em repouso (dados armazenados) quanto em trânsito (dados transmitidos).
7. **Atualizar e revisar permissões regularmente:** evitar que ex-funcionários mantenham acesso indevido.
8. **Backup seguro e verificado:** garantir restauração rápida em caso de ataque ou falha.

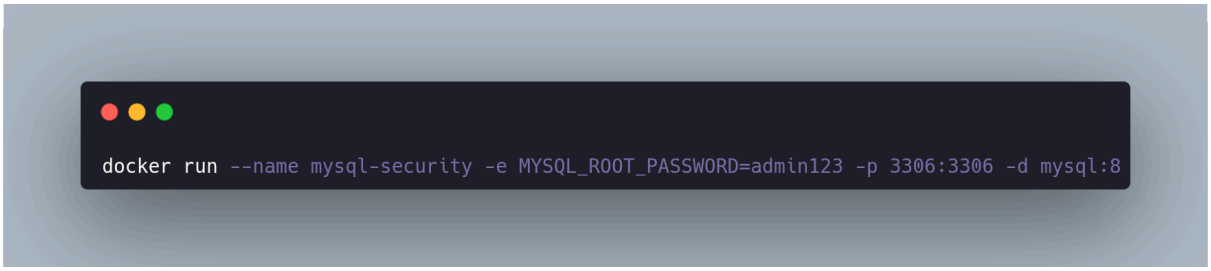
6. Atividade Prática

Objetivo:

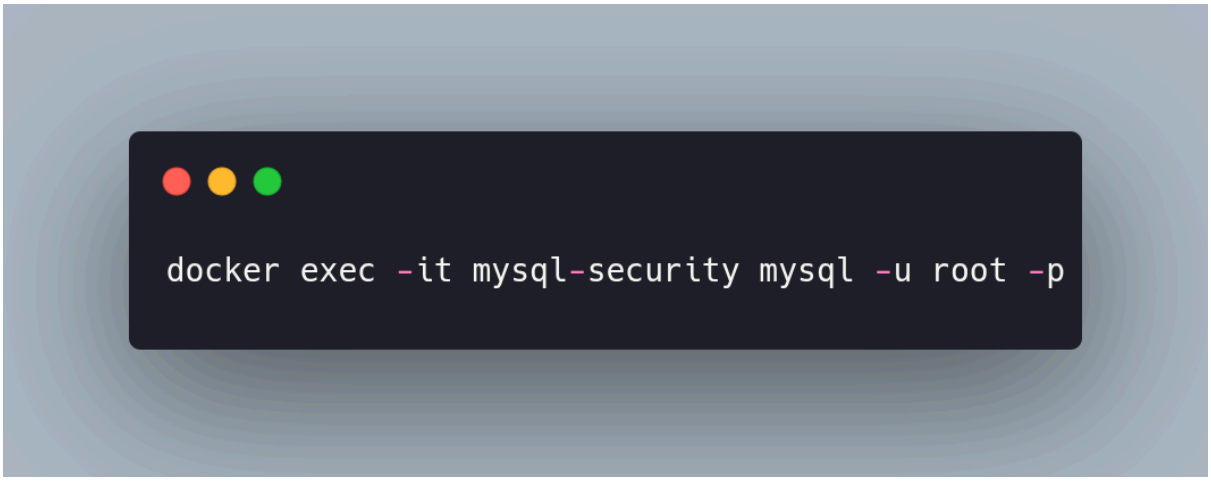
Configurar usuários, permissões e roles em um ambiente Docker com MySQL.

Passos:

1. **Criar um container MySQL:**



2. **Acessar o container:**



3. **Criar banco e usuários:**

```
CREATE DATABASE empresa;
CREATE USER 'analista'@'%' IDENTIFIED BY 'senha123';
CREATE ROLE 'leitor';
GRANT SELECT ON empresa.* TO 'leitor';
GRANT 'leitor' TO 'analista';
```

4. **Testar as permissões:**

```
mysql -u analista -p
USE empresa;
SELECT * FROM tabela_teste; -- deve funcionar
INSERT INTO tabela_teste VALUES (...); -- deve falhar
```

7. Discussão em Sala

Após a prática, discutir:

- Quais problemas surgem quando permissões são mal configuradas?
- Como roles ajudam a reduzir erros administrativos?
- Por que o princípio do menor privilégio é tão importante?

---

8. Conclusão

A segurança em bancos de dados não é apenas uma questão técnica — é **uma responsabilidade compartilhada** entre administradores, desenvolvedores e gestores. Entender como funcionam **usuários, permissões e roles** é o primeiro passo para garantir **integridade, confidencialidade e disponibilidade** dos dados, pilares da segurança da informação.

Referências

BARRETO, Jeanine dos Santos *et al.* **Fundamentos de segurança da informação.** Porto Alegre: SAGAH, 2018.

MORAES, Alexandre Fernandes de. **Cibersegurança e a nova geração de firewalls.** São Paulo: Saraiva Educação, 2021.

WANDERLEY, Alex Rodrigo Moisés Costa; PONTUAL, Ricardo de Almeida. **Gerenciamento de servidores.** São Paulo: Saraiva Educação, 2019.

DATE, C. J. **Introdução a sistemas de bancos de dados.** 8. ed. Rio de Janeiro: Elsevier, 2004.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistemas de banco de dados.** 6. ed. São Paulo: AMGH, 2012.

STALLINGS, William. **Segurança de redes e de computadores: princípios e prática.** 6. ed. São Paulo: Pearson, 2019.

ELMASRI, Ramez; NAVATHE, Shamkant. **Sistemas de banco de dados.** 7. ed. São Paulo: Pearson, 2017.