



LABORATÓRIO PRÁTICO • 120 MINUTOS

Administração Avançada: Views, Procedures e Redes

Aprofundamento técnico em abstração SQL, rotinas no servidor, controle granular de usuários (DCL) e roteamento de conexões externas no XAMPP.



Abstração Lógica



Privilégios DCL



Rede e Firewall

Roteiro da Sessão (120 min)

Estrutura progressiva: da base de dados local até a exposição segura na rede.

00 – 20 min

Views & Esquema

Apresentação da base de testes e criação de tabelas virtuais.

20 – 50 min

Stored Procedures

Delimitadores, parâmetros IN/OUT e lógicas de atualização.

50 – 80 min

Gestão DCL

Tipos de host, criação de contas e matriz de privilégios.

80 – 120 min

Rede & Laboratório

Configuração do my.ini e desafios práticos integrados.

Base de Dados: Loja Digital

Esquema relacional utilizado em todos os exemplos e exercícios da aula.

clientes

id INT (PK)
nome VARCHAR(100)
email VARCHAR(100)



pedidos

id INT (PK)
id_cliente INT (FK)
id_produto INT (FK)
quantidade INT



produtos

id INT (PK)
nome VARCHAR(100)
preco DECIMAL(10,2)
estoque INT

Views: Tabelas Virtuais Lógicas



Ocultação Estrutural

Protege colunas sensíveis expondo apenas dados necessários para cada setor.



Simplificação de JOINS

Encapsula consultas complexas em um único objeto de fácil leitura.



Armazenamento Zero

O banco armazena apenas a instrução SQL, não duplicando registros no disco.

Exemplo: Consolidar vendas por cliente

```
CREATE VIEW vw_resumo_vendas AS
SELECT
    c.nome,
    p.nome AS produto,
    pd.quantidade,
    (pd.quantidade * p.preco) AS total
FROM pedidos pd
JOIN clientes c ON pd.id_cliente = c.id
JOIN produtos p ON pd.id_produto = p.id;

-- Uso simplificado:
SELECT * FROM vw_resumo_vendas;
```

Stored Procedures: Lógica no Servidor

Rotinas pré-compiladas para regras de negócio pesadas.



O Delimitador (\$\$ ou //)

No MySQL, altera-se temporariamente o caractere finalizador padrão (;). Isso evita que o bloco BEGIN/END seja executado prematuramente durante a criação.



Fluxo de Parâmetros

- **IN:** Recebe valores do cliente.
- **OUT:** Retorna variáveis calculadas.
- **INOUT:** Modifica valores existentes.



Vantagem de Rede

Múltiplos comandos SQL rodam internamente no servidor. O cliente trafega apenas a chamada (CALL).

Sintaxe: Procedure com Transação

Exemplo: Registrar pedido e baixar estoque automaticamente

```
DELIMITER //
```

```
CREATE PROCEDURE sp_novo_pedido(  
    IN p_cliente INT,  
    IN p_produto INT,  
    IN p_qtd INT  
)  
BEGIN  
    -- Insere o registro de venda  
    INSERT INTO pedidos (id_cliente, id_produto, quantidade)  
    VALUES (p_cliente, p_produto, p_qtd);  
  
    -- Atualiza o estoque do produto logicamente  
    UPDATE produtos  
    SET estoque = estoque - p_qtd  
    WHERE id = p_produto;  
END //
```

```
DELIMITER ;
```

```
-- Chamada da rotina via aplicação:
```

```
CALL sp_novo_pedido(1, 45, 2);
```

Identidade e Restrição de Origem

No MySQL, um usuário é definido pela combinação `'nome'@'host'`.



'user'@'localhost'

Acesso estritamente interno. Aplicável a APIs hospedadas na mesma máquina do banco.



'user'@'%'

O curinga % permite conexões de qualquer IP externo. Requer controle rigoroso de senha.



'user'@'192.168.1.5'

Restrição fixa. Útil para permitir que apenas um servidor Web ou parceiro específico acesse.

DCL: Segurança Granular (GRANT / REVOKE)

Privilégio	Escopo de Ação
SELECT	Apenas leitura de registros.
INSERT, UPDATE	Modificação de dados existentes.
EXECUTE	Invocação de Stored Procedures.
ALL PRIVILEGES	Administração total da base.



Após DCL, execute **FLUSH PRIVILEGES** para forçar a recarga em memória.

```
-- Criando usuário externo:
CREATE USER 'vendedor'@'%' IDENTIFIED BY 'Senha123';

-- Isolando acesso apenas à View:
GRANT SELECT ON loja_db.vw_resumo_vendas
TO 'vendedor'@'%';

-- Permitindo execução da Procedure:
GRANT EXECUTE ON PROCEDURE loja_db.sp_novo_pedido
TO 'vendedor'@'%';

-- Removendo privilégios acidentais:
REVOKE DELETE ON loja_db.* FROM 'vendedor'@'%';

FLUSH PRIVILEGES;
```

Acesso Remoto Físico: XAMPP e Firewall

Por padrão, o MariaDB do XAMPP recusa conexões externas.

1

Arquivo my.ini

Abra a configuração via Painel XAMPP e altere a interface de escuta.

```
bind-address=0.0.0.0
```

2

Reiniciar Serviço

Pare e inicie o módulo MySQL para aplicar a nova rede.

Stop → Start

3

Regra de Firewall

No Windows, crie uma regra de entrada para liberar tráfego.

Porta Local TCP 3306

Arquitetura e Boas Práticas



Isolamento do Root

O usuário root nunca deve ter acesso externo. Mantenha-o sempre como `root@localhost`.



Menor Privilégio via Views

Em vez de conceder SELECT na tabela bruta, crie uma View restrita e conceda acesso apenas à View.



Sanitização Nativa

Forçar aplicações a usar Stored Procedures previne ataques diretos de SQL Injection estrutural.

Laboratório Prático • Parte 1

Duração: 20 min

1. Setup da Base

- Crie as tabelas clientes, produtos e pedidos.
- Insira 3 registros falsos em cada tabela.

2. Criação de View

- Crie a view vw_estoque_critico.
- Retorne produtos com estoque < 10.

3. Stored Procedure

- Crie sp_atualiza_preco.
- Receba ID do produto e novo preço (IN).
- Execute o UPDATE internamente.

Laboratório Prático • Parte 2 e Fim

Duração: 30 min



4. Rede & Acesso

- Libere o `bind-address` no `my.ini`.
- Crie o usuário `'dev'@'%'`.
- Conceda `SELECT` na view criada no Ex. 2.



5. Teste Individual

- Descubra seu IP na rede (`ipconfig`).
- Abra um novo terminal simulando acesso externo à sua máquina.

```
mysql -u dev -h SEU_IP_AQUI -p
```



6. Auditoria DCL

- Consulte a view e confirme o sucesso.
- Tente executar `DELETE` em produtos (deve falhar).
- No terminal root, aplique `REVOKE` e reteste a leitura.

✓ Laboratório concluído. Commit das soluções no repositório.