



PROGRAMAÇÃO SQL

Views e Stored Procedures

OBJECTIVOS

- Transactions
- Views
- Built-In Functions
- Stored Procedures

BUILD-IN FUNCTIONS

- Funções pré-definidas fornecidas pelo MySQL para realizar operações específicas em dados
- Têm o objetivo de simplificar consultas, manipulação e análise de dados

STRING FUNCTIONS

- CONCAT(), LENGTH(), SUBSTRING(), UPPER(), LOWER()

```
SELECT    CONCAT(nome, ' ', morada)
FROM      aluno;
```

DATE/HOUR FUNCTIONS

- NOW(), DATE(), TIME(), YEAR(), MONTH(), DAY()

```
SELECT    YEAR(data_inscrição),  
MONTH(data_inscrição), DAY(data_inscrição),  
NOW() AS 'Data de Hoje'  
FROM      aluno;
```

NUMBER FUNCTIONS

- ABS(), ROUND(), CEIL(), FLOOR(), RAND()

```
SELECT ROUND(preco*stock, 2)
FROM produto;
```

ADVANCED FUNCTIONS

- IF(), CASE WHEN

```
SELECT nome, IF(idade>=18, 'Adulto', 'Menor') AS  
  'Faixa Etária'  
FROM aluno;
```

VANTAGENS

- Reduz a complexidade do código SQL
- Agiliza a manipulação de dados ao fornecer operações prontas para uso
- Facilita a criação de consultas poderosas com menos esforço
- Seguem os mesmos padrões entre SGBDs
- São otimizadas internamente pelos SGBDs para proporcionar melhor desempenho.

MAIS FUNÇÕES

- Para mais funções prontas a usar no MySQL podem consultar:

https://www.w3schools.com/mysql/mysql_ref_functions.asp

OBJECTIVOS

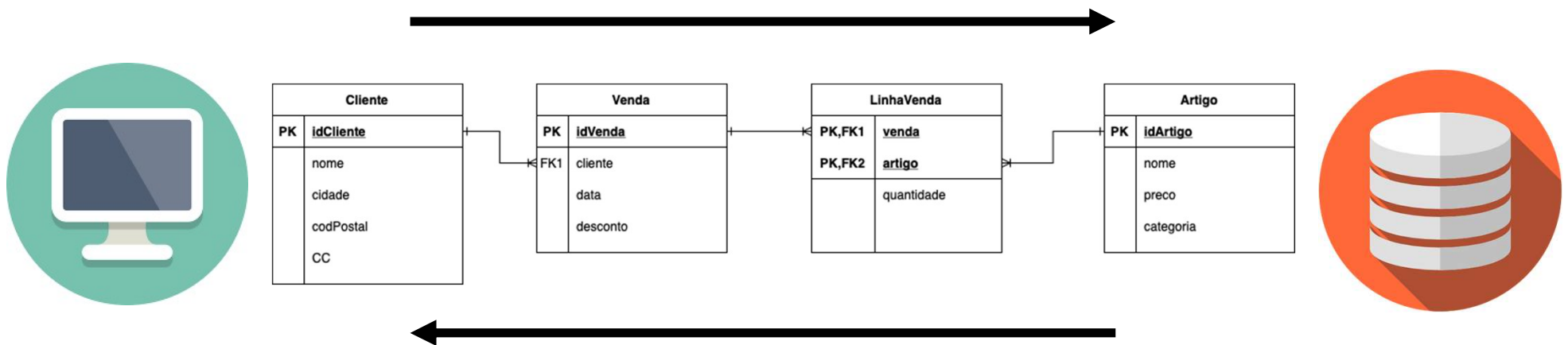
- Transactions
- Views
- Built-In Functions
- Stored Procedures

STORED PROCEDURE

- É uma coleção de comandos em SQL
- Agrupa tarefas repetitivas

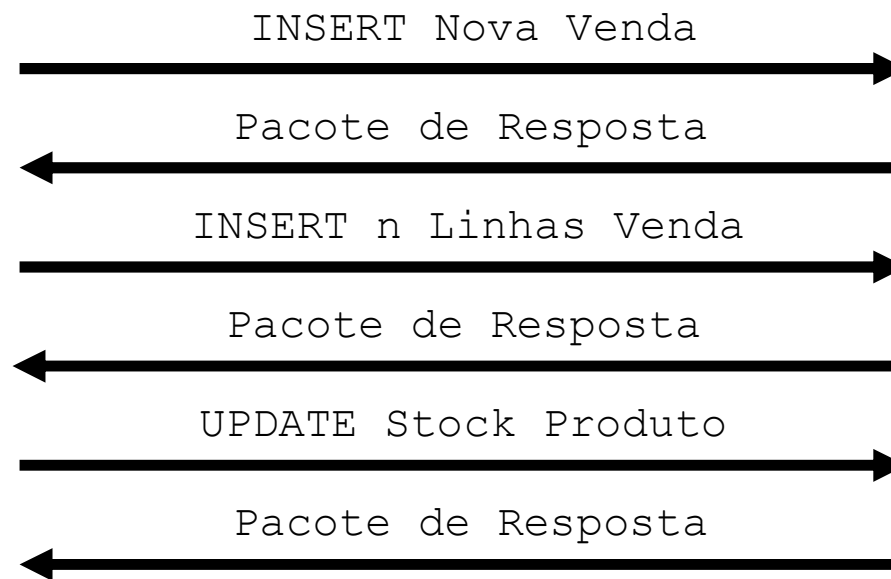
EXEMPLO

- Nova compra numa App Mobile



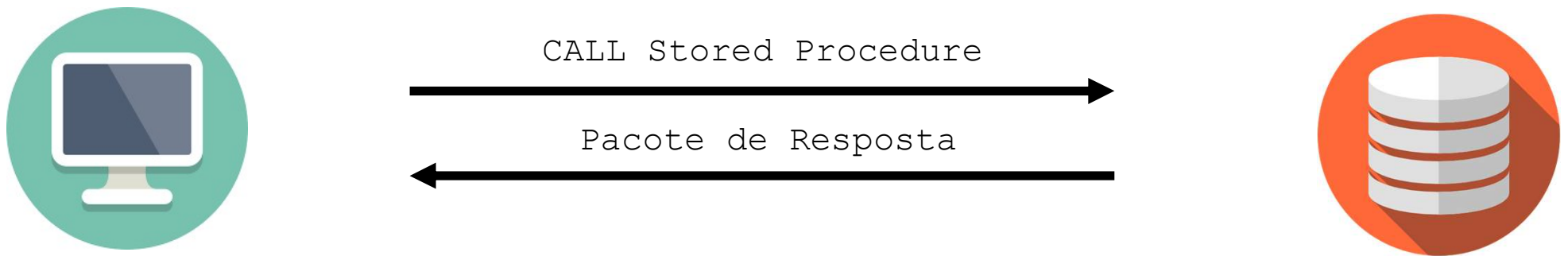
EXEMPLO

- Nova compra numa App Mobile sem Stored Procedure



EXEMPLO

- Nova compra numa App Mobile com Stored Procedure



STORED PROCEDURE

- Aceita parâmetros de entrada e retorna um valor de saída.
- Pode reduzir o tráfego na rede, visto que os comandos são executados diretamente no servidor e cria mecanismos de segurança entre a manipulação dos dados.

VANTAGENS

- Simplifica a execução de instruções SQL;
- Facilidade no desenvolvimento do lado das aplicações ou websites;
- Responsabilidade de processamento de dados é transferida para o servidor.

STORED PROCEDURE

- Estrutura em MySQL

```
DELIMITER //  
CREATE PROCEDURE nome_storedprocedures()  
BEGIN  
    comando  
END //  
DELIMITER ;
```

STORED PROCEDURE

- Estrutura em MySQL

CALL nome_storedprocedure

STORED PROCEDURE

- Estrutura em MySQL

```
DELIMITER //  
CREATE PROCEDURE nome(Entrada_Saida variável/eis tipo_de_dados)  
BEGIN  
    comando  
END //  
DELIMITER ;
```

STORED PROCEDURE

- Estrutura em MySQL

DELIMITER //

CREATE PROCEDURE nome(**IN** variável/eis tipo_de_dados)

BEGIN

comando

END //

DELIMITER ;

STORED PROCEDURE

- Estrutura em MySQL

```
DELIMITER //  
CREATE PROCEDURE nome(OUT variável/eis tipo_de_dados)  
BEGIN  
    comando  
END //  
DELIMITER ;
```

STORED PROCEDURE

- Estrutura em MySQL

```
CALL nome_storedprocedure(@nome_variável)
```

```
SELECT @nome_variável
```

STORED PROCEDURE

- Estrutura em MySQL

```
DROP PROCEDURE nome_storedprocedure
```

EXEMPLO

- Permite inserir um registo de um novo cliente

```
DELIMITER //  
CREATE PROCEDURE insereCliente  
(IN in_nomeCliente VARCHAR(200),  
  IN in_telef INT,  
  IN in_email VARCHAR(200) )  
BEGIN  
    INSERT INTO Cliente (nome, telef, mail)  
    VALUES (in_nomeCliente, in_telef, in_email);  
END //  
DELIMITER ;
```


EXEMPLO

- Executar Stored Procedure `insereCliente`

```
CALL insereCliente('Helder', '912345678', 'helder@mail.pt');
```

```
CALL insereCliente('André', '912345678', 'andre@mail.pt');
```

EXEMPLO

- Pesquisa os dados de um dado Cliente (pelo nome)

```
DELIMITER //
```

```
CREATE PROCEDURE pesqFacturasDeClientePorTelef
```

```
(IN in_telef VARCHAR(20) )
```

```
BEGIN
```

```
    SELECT c.nome, f.data FROM Cliente c, Factura f
```

```
    WHERE c.id=f.cliente and c.telef = in_telef;
```

```
    ORDER BY f.data
```

```
END //
```

```
DELIMITER ;
```

EXEMPLO

- Executar Stored Procedure pesqClientesPorNome

```
CALL pesqFacturasDeClientePorTelef('912345678');
```



www.cesae.pt

