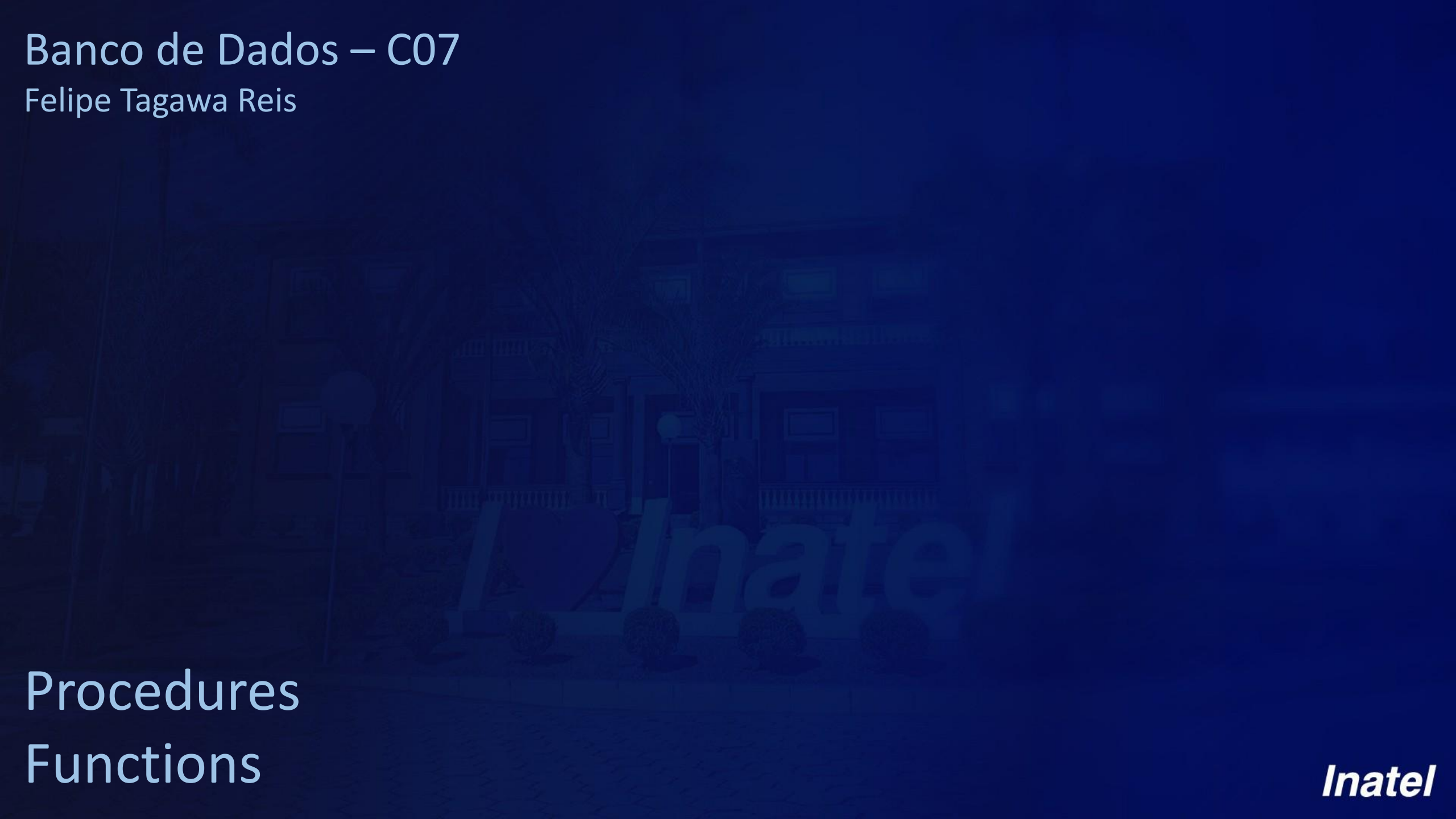
Banco de Dados – C07

Felipe Tagawa Reis



Procedures
Functions

Inatel

Procedures

Procedures são blocos de código SQL que podem ser armazenados no banco de dados e executados quando necessário. De forma geral, executam comandos, modificam dados, mas não retornam valores diretamente. Prioriza consistência e segurança, na medida em que uma funcionalidade seja utilizada em diferentes lugares, a manutenção é facilitada e sua centralização evita a chance de erros.

Procedures

Procedures são blocos de código SQL que podem ser armazenados no banco de dados e executados quando necessário. De forma geral, executam comandos, modificam dados, mas não retornam valores diretamente. Prioriza consistência e segurança, na medida em que uma funcionalidade seja utilizada em diferentes lugares, a manutenção é facilitada e sua centralização evita a chance de erros.

Desvantagem: Requer maior processamento por parte do servidor.

Delimita início e fim de blocos de Código (‘$$’ no lugar de ‘;’)

Procedures

```
1  DELIMITER $$
2
3  DROP PROCEDURE IF EXISTS AtualizarSalario$$
4
5  CREATE PROCEDURE AtualizarSalario(IN funcionario_id INT, IN
percentual_aumento DECIMAL(5,2))$$
6
7  BEGIN
8      UPDATE Funcionarios
9      SET salario = salario * (1 + percentual_aumento / 100)
10     WHERE id = funcionario_id;
11
12     SELECT 'Salário atualizado com sucesso!' AS Resultado;
13 END $$
14
15 DELIMITER ;
```

Inatel

Procedures

Delimita início e fim de blocos de Código (‘$$’ no lugar de ‘;’)

Criação da Procedure

```
1  DELIMITER $$
2
3  DROP PROCEDURE IF EXISTS AtualizarSalario$$
4
5  CREATE PROCEDURE AtualizarSalario(IN funcionario_id INT, IN
percentual_aumento DECIMAL(5,2))$$
6
7  BEGIN
8      UPDATE Funcionarios
9      SET salario = salario * (1 + percentual_aumento / 100)
10     WHERE id = funcionario_id;
11
12     SELECT 'Salário atualizado com sucesso!' AS Resultado;
13 END $$
14
15 DELIMITER ;
```

Inatel

Procedures

Delimita início e fim de blocos de Código (‘$$’ no lugar de ‘;’)

Criação da Procedure

Parâmetros

```
1  DELIMITER $$
2
3  DROP PROCEDURE IF EXISTS AtualizarSalario$$
4
5  CREATE PROCEDURE AtualizarSalario(IN funcionario_id INT, IN
percentual_aumento DECIMAL(5,2))$$
6
7  BEGIN
8      UPDATE Funcionarios
9      SET salario = salario * (1 + percentual_aumento / 100)
10     WHERE id = funcionario_id;
11
12     SELECT 'Salário atualizado com sucesso!' AS Resultado;
13 END $$
14
15 DELIMITER ;
```

Inatel

Procedures

Delimita início e fim de blocos de Código (‘$$’ no lugar de ‘;’)

Criação da Procedure

Início do Corpo

Parâmetros

```
1  DELIMITER $$
2
3  DROP PROCEDURE IF EXISTS AtualizarSalario$$
4
5  CREATE PROCEDURE AtualizarSalario(IN funcionario_id INT, IN
percentual_aumento DECIMAL(5,2))$$
6
7  BEGIN
8      UPDATE Funcionarios
9      SET salario = salario * (1 + percentual_aumento / 100)
10     WHERE id = funcionario_id;
11
12     SELECT 'Salário atualizado com sucesso!' AS Resultado;
13 END $$
14
15 DELIMITER ;
```

Inatel

Procedures

Delimita início e fim de blocos de Código (‘$$’ no lugar de ‘;’)

Criação da Procedure

Início do Corpo

Corpo

Parâmetros

```
1  DELIMITER $$
2
3  DROP PROCEDURE IF EXISTS AtualizarSalario$$
4
5  CREATE PROCEDURE AtualizarSalario(IN funcionario_id INT, IN
percentual_aumento DECIMAL(5,2))$$
6
7  BEGIN
8      UPDATE Funcionarios
9      SET salario = salario * (1 + percentual_aumento / 100)
10     WHERE id = funcionario_id;
11
12     SELECT 'Salário atualizado com sucesso!' AS Resultado;
13 END $$
14
15 DELIMITER ;
```

Inatel

Procedures

Delimita início e fim de blocos de Código (‘$$’ no lugar de ‘;’)

Criação da Procedure

Início do Corpo

Corpo

Finaliza toda a Procedure

Parâmetros

```
1  DELIMITER $$
2
3  DROP PROCEDURE IF EXISTS AtualizarSalario$$
4
5  CREATE PROCEDURE AtualizarSalario(IN funcionario_id INT, IN
percentual_aumento DECIMAL(5,2))$$
6
7  BEGIN
8      UPDATE Funcionarios
9      SET salario = salario * (1 + percentual_aumento / 100)
10     WHERE id = funcionario_id;
11
12     SELECT 'Salário atualizado com sucesso!' AS Resultado;
13 END $$
14
15 DELIMITER ;
```

Inatel

Procedures

Delimita início e fim de blocos de Código (‘$$’ no lugar de ‘;’)

Criação da Procedure

Início do Corpo

Corpo

Finaliza toda a Procedure

Parâmetros

Restaura o delimitador padrão para ‘;’

```
1  DELIMITER $$
2
3  DROP PROCEDURE IF EXISTS AtualizarSalario$$
4
5  CREATE PROCEDURE AtualizarSalario(IN funcionario_id INT, IN
percentual_aumento DECIMAL(5,2))$$
6
7  BEGIN
8      UPDATE Funcionarios
9      SET salario = salario * (1 + percentual_aumento / 100)
10     WHERE id = funcionario_id;
11
12     SELECT 'Salário atualizado com sucesso!' AS Resultado;
13 END $$
14
15 DELIMITER ;
```

Inatel

Procedures - Delimiter

Forma Incorreta

```
-- ✗ PROBLEMA: Sem delimiter personalizado
CREATE PROCEDURE Teste()
BEGIN
    SELECT 'Oi'; -- MySQL pensa que a procedure termina aqui!
    SELECT 'Tchau';
END; -- Isso nunca seria executado
```

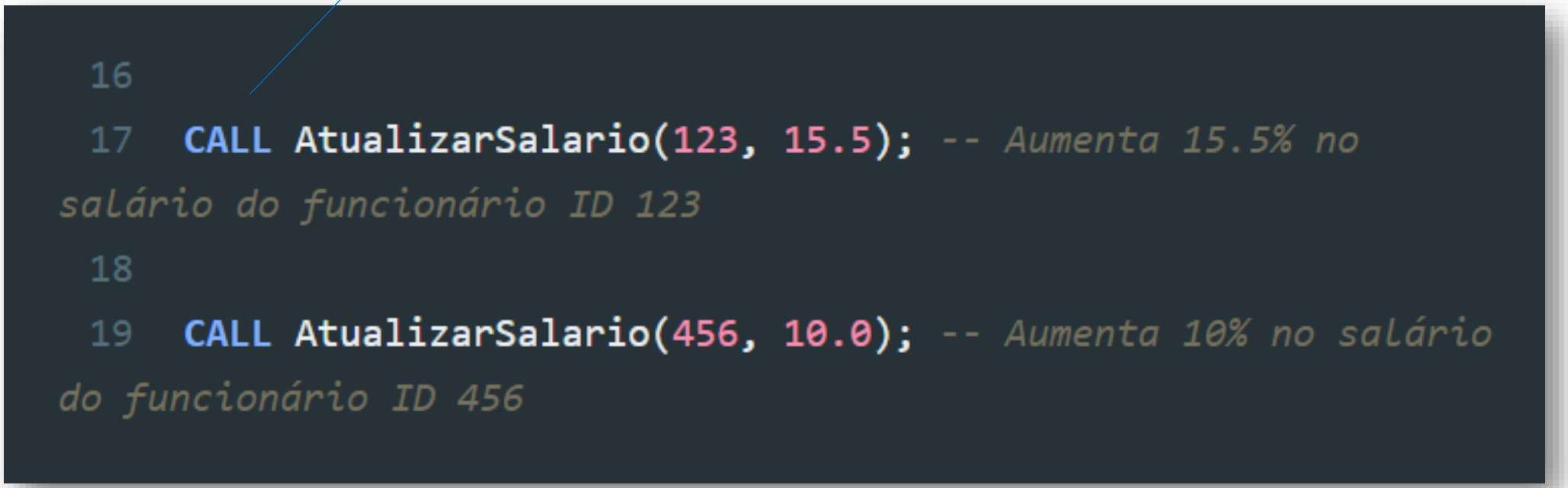
Forma Correta

```
DELIMITER $$
CREATE PROCEDURE Teste()
BEGIN
    SELECT 'Oi';      -- MySQL ignora este ;
    SELECT 'Tchau';   -- E este também
END $$                -- Só termina aqui!
DELIMITER ;
```

Inatel

Procedures - Utilização

Chamando a Procedure



```
16
17  CALL AtualizarSalario(123, 15.5); -- Aumenta 15.5% no
salário do funcionário ID 123
18
19  CALL AtualizarSalario(456, 10.0); -- Aumenta 10% no salário
do funcionário ID 456
```

Inatel

Procedures - Utilização

Chamando a Procedure

```
16
17  CALL AtualizarSalario(123, 15.5); -- Aumenta 15.5% no
salário do funcionário ID 123
18
19  CALL AtualizarSalario(456, 10.0); -- Aumenta 10% no salário
do funcionário ID 456
```

```
DROP PROCEDURE AtualizarSalario;
```

Apagando a Procedure

Inatel

Procedures – While + IF

```sql
DELIMITER $$

DROP PROCEDURE IF EXISTS ContarNumeros$$

CREATE PROCEDURE ContarNumeros(
    IN limite INT
)$$

BEGIN
    -- Declaração de variáveis
    DECLARE contador INT DEFAULT 1;
    DECLARE pares INT DEFAULT 0;
    DECLARE impares INT DEFAULT 0;

    -- Loop WHILE para contar de 1 até o limite
    WHILE contador <= limite DO

        -- Verifica se o número é par ou ímpar
        IF contador MOD 2 = 0 THEN
            -- É par
            SET pares = pares + 1;
        ELSE
            -- É ímpar
            SET impares = impares + 1;
        END IF;

        -- Incrementa contador
        SET contador = contador + 1;

    END WHILE;

    -- Retorna resultado
    SELECT
        limite AS 'Limite',
        pares AS 'Números Pares',
        impares AS 'Números Ímpares';

END $$

DELIMITER ;
```

Inatel

Functions

Functions são blocos de código SQL que podem ser armazenados no banco de dados e executados quando necessário. De forma geral, modificam dados, retornando valores diretamente.
Prioriza consistência e segurança, na medida em que uma funcionalidade seja utilizada em diferentes lugares, a manutenção é facilitada e sua centralização evita a chance de erros;

Não podem executar comandos como INSERT, UPDATE e DELETE diretamente (mais simples que Procedures)

Functions

Aleatoriedade?

```
DELIMITER $$

DROP FUNCTION IF EXISTS soma $$

CREATE FUNCTION soma(a FLOAT, b FLOAT) RETURNS FLOAT
DETERMINISTIC
BEGIN
    RETURN a + b;
END $$

DELIMITER ;

SELECT soma(10.5, 4.25) AS Resultado;

DROP FUNCTION soma;
```

Tipo do Retorno

Chamando a Function

Inatel

AULA CONCLUÍDA!

Inatel