

Cognitive Data Science

PROF. DR. FERNANDO T. FERNANDES

FIAP

Vimos

- Normalização

FIAP

Agenda

- Stored Procedures
- Functions
- Triggers
- Views

FIAP

Stored Procedures

- Comando SQL que pode ser salvo e reutilizado.
- Pode receber parâmetros

```
delimiter //
```

```
CREATE PROCEDURE <nome_proc> (<parametros>)  
BEGIN  
    <comando sql>  
END //
```

```
delimiter ;
```

```
CALL <nome_proc>;
```

Stored Procedures

○ Exemplo:

```
mysql> create procedure exibirProgramadores ()  
-> BEGIN  
-> SELECT * FROM programador;  
-> END //  
Query OK, 0 rows affected (0.02 sec)  
  
mysql> delimiter ;
```

```
mysql> CALL exibirProgramadores();  
+-----+-----+-----+-----+-----+  
| idProgramador | CPF          | nome  | email          | habilidades |  
+-----+-----+-----+-----+-----+  
|          1   | 12345678911 | Miguel | miguel@gmail.com | c#, sql, java, ruby, python |  
|          2   | 98765432110 | Luana  | luana@gmail.com  | sql, python, aws, nodejs, k8s, docker, php |  
|          3   | 55555555555 | Ana    | ana@gmail.com    | java, javascript, python |  
+-----+-----+-----+-----+-----+
```

Stored Procedures com parâmetros

○ Exemplo:

```
mysql> delimiter //
mysql> create procedure exhibirProgramador (id INT)
-> BEGIN
->   SELECT * FROM programador WHERE idProgramador = id;
-> END //
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
```

```
mysql> CALL exhibirProgramador(1);
+-----+-----+-----+-----+-----+
| idProgramador | CPF          | nome   | email                | habilidades                |
+-----+-----+-----+-----+-----+
|              | 12345678911 | Miguel | miguel@gmail.com     | c#, sql, java, ruby, python |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Functions

- Trecho de código encapsulado.
- Pode ser usado para armazenar fórmulas ou regras de negócio reutilizáveis.

```
DELIMITER //
```

```
SELECT <nome_func>(parametros);
```

```
CREATE FUNCTION <nomefuncao> (<lista_parametros>)
```

```
RETURNS <tipodado>
```

```
DETERMINISTIC
```

```
  BEGIN
```

```
    RETURN <valor_retornado>;
```

```
  END//
```

```
DELIMITER ;
```

Functions - Exemplo

○ Cálculo de IMC

```
mysql> DELIMITER //
```

```
mysql> CREATE FUNCTION IMC (peso FLOAT, altura FLOAT)
```

```
    -> RETURNS FLOAT
```

```
    -> DETERMINISTIC
```

```
    -> BEGIN
```

```
    -> RETURN peso/(altura*altura);
```

```
    -> END//
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> DELIMITER ;
```

```
mysql> SELECT IMC(70,1.70);
```

IMC(70,1.70)
24.2215

TRIGGERS

- Trecho de código invocado automaticamente quando ocorre um evento (INSERT, UPDATE ou DELETE) em uma tabela.
- Podem auxiliar a garantir a integridade dos dados
- Podem auxiliar a executar tarefas agendadas
- Podem ser usadas para auditoria
 - Ex: quem exclui o registro XYX ?

```
CREATE TRIGGER <nometrigger>  
<BEFORE|AFTER> <INSERT|UPDATE|DELETE>  
ON <NOMETABELA> FOR EACH ROW  
COMANDO TRIGGER;
```

TRIGGERS - auditoria

```
mysql> CREATE TABLE programador_auditoria (  
-> id INT AUTO_INCREMENT PRIMARY KEY,  
-> habilidades VARCHAR(100) NOT NULL,  
-> usuario VARCHAR(50) NOT NULL,  
-> data DATE NULL,  
-> acao VARCHAR(50) NULL  
-> );
```

1) Crie a tabela de auditoria

```
mysql> CREATE TRIGGER antes_atualizar_programador  
-> BEFORE UPDATE ON programador  
-> FOR EACH ROW  
-> INSERT INTO programador_auditoria  
-> SET acao = 'update',  
->     habilidades = OLD.habilidades,  
->     data = NOW(),  
->     usuario = CURRENT_USER();
```

2) Crie a TRIGGER para a tabela

```
mysql> UPDATE programador  
-> SET habilidades = CONCAT(habilidades, ' ', 'javascript')  
-> WHERE idProgramador = 1;
```

3) Realizar uma alteração

TRIGGERS - auditoria

4) Valor auditado de habilidades

```
mysql> SELECT * FROM programador_auditoria;
```

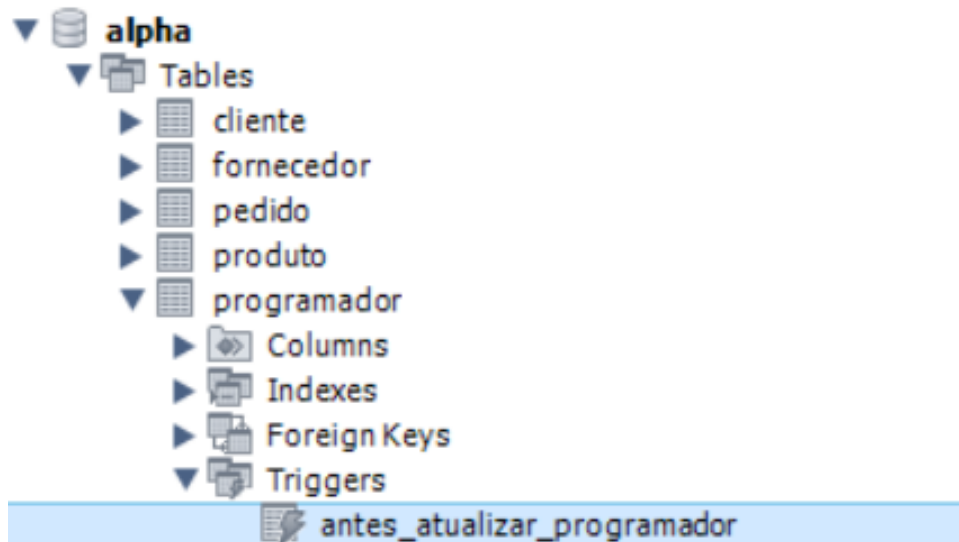
id	habilidades	usuario	data	acao
1	c#, sql, java, ruby, python	root@localhost	2022-02-04	update

5) Novo valor de habilidades do programador

```
mysql> SELECT * FROM programador;
```

idProgramador	CPF	nome	email	habilidades
1	12345678911	Miguel	miguel@gmail.com	c#, sql, java, ruby, python ,javascript
2	98765432110	Luana	luana@gmail.com	sql, python, aws, nodejs, k8s, docker, php
3	55555555555	Ana	ana@gmail.com	java, javascript, python

Listar TRIGGERS



SHOW TRIGGERS;

```
mysql> show triggers;
+-----+-----+-----+-----+-----+-----+
| Trigger          | Event | Table      | Statement                                     | Definer          | character_set_client |
+-----+-----+-----+-----+-----+-----+
| antes_atualizar_programador | UPDATE | programador | INSERT INTO programador_auditoria          | root@localhost   | utf8mb4_32bit        |
| SET acao = 'update',      |      |      |      |      |      |
|      habilidades = OLD.habilidades, |      |      |      |      |      |
|      data = NOW(),        |      |      |      |      |      |
|      usuario = CURRENT_USER() | BEFORE | 2022-02-04 10:51:13.02 | STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION | root@localhost   | utf8mb4_32bit        |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Views

- Permite criar uma tabela “virtual” para retornar uma chamada customizada
- Usada quando há múltiplos JOINS ou resultados de relatórios extensos
- Exibe somente os campos necessários
 - Pode ocultar informações de ids, etc.

```
CREATE VIEW <nomeview>  
AS  
<consulta SQL> ;
```

Views

- Listar projetos e programadores

```
mysql> CREATE VIEW  ProjetosEmAndamento AS  
-> SELECT P.sigla, PR.nome, PP.horas  
-> FROM projeto_programador PP  
-> LEFT JOIN projeto P ON PP.idProjeto = P.idProjeto  
-> LEFT JOIN programador PR ON PP.idProgramador = PR.idProgramador  
-> ORDER BY P.sigla, PR.nome;
```

```
mysql> SELECT * FROM ProjetosEmAndamento;  
+-----+-----+-----+  
| sigla  | nome   | horas |  
+-----+-----+-----+  
| alpha1 | Luana  | 8     |  
| alpha1 | Miguel | 16    |  
| alpha2 | Miguel | 2     |  
+-----+-----+-----+
```

Reutilizo uma longa *query* mais facilmente!

SGBD – Prática

- ❑ Crie uma Trigger para auditar alterações na sigla da tabela projeto
- ❑ Crie uma Proc que exibe projetos e aceita um id como parâmetro
- ❑ Crie uma Function que faça a multiplicação de 2 números
- ❑ Crie uma View que liste os projetos e programadores, exibindo:
 - ❑ sigla do projeto
 - ❑ nome do programador
 - ❑ email do programador
 - ❑ habilidades do programador

Obrigado!

Contato: proffernando.fernandes@fiap.com.br