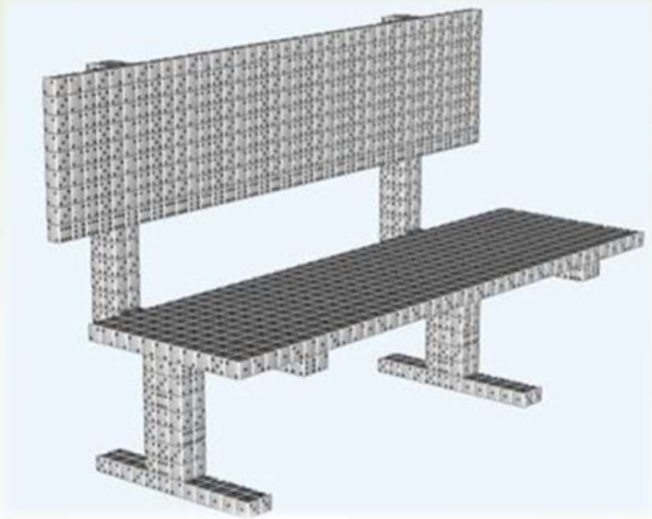

Banco de Dados II
- SQL Avançado - *Triggers*

Prof. Angelo Augusto Frozza, Dr.

http://about.me/TilFrozza

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Roteiro

- *Triggers* (Gatilhos)

Triggers

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

- ***Triggers*** (ou **gatilhos**) são objetos acessórios a tabelas e visões, que funcionam como *listeners* (“ouvidores”) de eventos

- O objetivo das ***triggers*** é observar a ocorrência de eventos (inserção, atualização ou exclusão) em registros ou execução de comandos SQL sobre objetos aos quais os gatilhos estão vinculados, **ANTES** ou **DEPOIS** da sua ocorrência.
  - É comum que sejam implementados para executar operações que são derivadas, diretamente, de outras operações.
  - Exemplo: gravar *logs* de manutenção de dados.

Triggers

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

- **Vantagens**
  - Mais uma alternativa que permite checar a integridade de dados.
  - Podem capturar erros na lógica de negócios na camada de BD.
  - É uma alternativa para executar tarefas no próprio BD.
  - Útil para auditar modificações no BD.

Triggers

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Limitações

- *Triggers* não podem utilizar certas declarações
  - LOAD DATA, LOAD TABLE, BACKUP DATABASE, RESTORE, FLUSH e RETURN
  - COMMIT, ROOLBACK, START TRANSACTION, LOCK/UNLOCK TABLES, ALTER, CREATE, DROP, RENAME, PREPARE, EXECUTE
  - SQL dinâmica
  - Chamar uma *Stored Procedure* ou *Function*

Triggers

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Desvantagens

- Serve como validador de apenas uma parte das informações.
- Por ser executada na camada de BD fica difícil para o usuário saber o que está acontecendo no BD.
- Pode aumentar o atraso no processamento no servidor de BD.

Triggers

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

- No ***PostgreSQL*** as ***Triggers*** são criadas a partir de dois objetos do banco de dados:
  - ***Trigger Function*** – define como uma tarefa vai ser executada (**CREATE FUNCTION**)
  - ***Trigger*** – define a tarefa a ser executada (**CREATE TRIGGER**)

Triggers

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

- Sintaxe básica: ***Trigger Function***

```
CREATE OR REPLACE FUNCTION
                trigger_function_name()
RETURNS trigger AS $ExemploFuncao$
BEGIN
    /* Aqui definem-se os códigos que
serão executados */
RETURN NEW;
END;
$ExemploFuncao$ LANGUAGE 'plpgsql';
```

Triggers

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

- Sintaxe básica: ***Trigger***

```
CREATE [ CONSTRAINT ] TRIGGER name
    { BEFORE AFTER INSTEAD OF } { event [ OR ... ] }
    ON tableName
    [ FROM referenced_table_NAME ]
    [ NOT DEFERRABLE | [ DEFERRABLE ]
     { INITIALLY IMMEDIATE | INITIALLY DEFERRED } ]
    [ FOR [ EACH ] { ROW | STATEMENT } ]
    [ WHEN ( condition ) ]
    EXECUTE PROCEDURE functionName ( arguments )
```

Triggers

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

- Exemplo prático:
  - **Criação de um *log* no BD da Clínica Médica** para registro de alterações na tabela ***funcionarios***.

```
CREATE TABLE log (
    id       SERIAL PRIMARY KEY,
    date     TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    table    TEXT NOT NULL,
    oldData  TEXT DEFAULT '',
    newData  TEXT DEFAULT ''
);
```

Triggers

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

```
CREATE OR REPLACE FUNCTION
            registraLogFuncionario()
RETURNS trigger AS $registraLog$
DECLARE dadosAntigos TEXT;
        dadosNovos TEXT;
        tabela TEXT;
BEGIN
    O código vai aqui!!!!
RETURN NEW;
END;
$registraLog$ LANGUAGE 'plpgsql'
```

Triggers

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

```
CREATE OR REPLACE FUNCTION registraLogFuncionario()
RETURNS trigger AS $registraLog$
DECLARE dadosAntigos TEXT; dadosNovos TEXT; tabela TEXT;
BEGIN
  tabela := 'FUNCIONARIOS';
 IF (TG_OP = 'UPDATE') THEN
    dadosAntigos := ROW(OLD.*);
    dadosNovos := ROW(NEW.*);
    INSERT INTO log (table, oldData, newData) VALUES (tabela, dadosAntigos, dadosNovos);
    RETURN NEW;
 ELSEIF (TG_OP = 'DELETE') THEN
    dadosAntigos := ROW(OLD.*);
    INSERT INTO log (table, oldData, newData) VALUES (tabela, dadosAntigos, DEFAULT);
    RETURN OLD;
 ELSEIF (TG_OP = 'INSERT') THEN
    dadosNovos := ROW(NEW.*);
    INSERT INTO log (table, oldData, newData) VALUES (tabela, DEFAULT, dadosNOvos);
    RETURN NEW;
 END IF;
RETURN NEW;
END;
$registraLog$ LANGUAGE 'plpgsql'
```

Triggers

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

```
...
  tabela := 'FUNCIONARIOS';
IF (TG_OP = 'UPDATE') THEN
   dadosAntigos := ROW(OLD.*);
   dadosNovos := ROW(NEW.*);
   INSERT INTO log (table, oldData, newData)
                VALUES (tabela, dadosAntigos, dadosNovos);
   RETURN NEW;
ELSEIF (TG_OP = 'DELETE') THEN
   dadosAntigos := ROW(OLD.*);
   INSERT INTO log (table, oldData, newData)
                VALUES (tabela, dadosAntigos, DEFAULT);
   RETURN OLD;
ELSEIF (TG_OP = 'INSERT') THEN
   dadosNovos := ROW(NEW.*);
   INSERT INTO log (table, oldData, newData)
                 VALUES (tabela, DEFAULT, dadosNOvos);
   RETURN NEW;
END IF;
...
```

Triggers

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

```
CREATE TRIGGER logFuncionario
    AFTER INSERT OR UPDATE OR DELETE
                        ON funcionarios
    FOR EACH ROW EXECUTE PROCEDURE
                registraLogFuncionario();
```

Triggers

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

- Exercícios:
  - Criar uma ***trigger*** que verifique e grave o nome de novos pacientes em MAIÚSCULO;
  - Criar uma nova tabela chamada "***log***", com os seguintes atributos: "***identificador***" (SERIAL), "***tabela***" (*varchar* com 50 posições), "***data***" (timestamp), "***operacao***" (*varchar* com 10 posições), "***dadosNovos***" (texto), "***dadosAntigos***" (texto);
  - Criar uma ***trigger*** de ***log.*** As tabelas a serem monitoradas via ***trigger*** são: "Pacientes" e 'Consultas'. A ***trigger*** deve fazer as seguintes operações:
    a) quando ocorrerem atualizações (***UPDATE***) nos registros dessas tabelas, o SGBD deve inserir registros na tabela "***log***", preenchendo seus atributos com o nome da tabela que está sendo modificada, a operação que está sendo executada ("***UPDATE***") e o conteúdo anterior e atual dos registros que estão sendo modificados;
    b) quando ocorrerem exclusões (***DELETE***) de registros dessas tabelas, o SGBD deve inserir registros na tabela "***log***", preenchendo seus atributos com o nome da tabela cujos registros estão sendo excluídos, a operação que está sendo executada ("***DELETE***") e o conteúdo dos registros que estão sendo excluídos.

Contato

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

- Prof. Angelo Augusto Frozza, Dr.

angelo.frozza@ifc.edu.br

http://www.ifc-camboriu.edu.br/~frozza

@TilFrozza

http://www.twitter.com/TilFrozza

http://about.me/TilFrozza