BASE DE DADOS

Oracle
PL/SQL –
Triggers

TRIGGERS

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

- Semelhante aos Procedimentos e às Funções.
- Associados a Tabelas e a Views
- Executados automaticamente quando:
  - Há modificação de dados (DML Trigger)
    - INSERT, UPDATE, coluna UPDATE ou DELETE
  - modificação de esquema (DDL Trigger)
  - eventos do sistema, login / logoff do utilizador (Trigger de sistema)
- A utilização de triggers deve ser muito cuidadosa (apenas quando necessário) o uso excessivo de triggers pode resultar em interdependências complexas (Cascadind Triggers) que dificultam a manutenção de grandes aplicações.

TRIGGERS

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

CREATE [OR REPLACE] TRIGGER *trigger_name*

AFTER | BEFORE | INSTEAD OF *a_trigger_event* → ***instrução de triggering***

ON *table_name (or view_name)*

[**FOR EACH ROW[WHEN trigger_condition**]] → ***restrição***

DECLARE (opcional)

BEGIN (obrigatório)

Executado unicamente quando a condição de Trigger é TRUE

EXCEPTION (opcional)

Exception Section

END; (obrigatório)

ação ou corpo do trigger

NOTA: Um *trigger_event* pode ser qualquer combinação de um INSERT, DELETE e/ou UPDATE numa tabela ou view

TRIGGERS

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

- Tipos

- Quando se define um Trigger é possível especificar se este deve ser executado:
  - para cada linha afectada pela instrução de triggering, tal como um Update statement que actualiza 'n' linhas. (triggers de linha)
  - para cada instrução de triggering, independentemente do numero de linhas que afecte (triggers de instrução)
  - antes da instrução de triggering
  - depois da instrução de triggering

- É possível ter vários triggers do mesmo tipo para a mesma tabela

TRIGGERS - LINHA

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Exemplo de Trigger de linha

- **Criando um trigger de Linha**

```
/* Validando o domínio de um salário */
create or replace trigger testa_salario
   before insert or update of salario on funcionario
   for each row
   begin
     if :new.salario > 8000 then
       raise_application_error(-20000,'VALOR INCORRETO');
     end if;
   end;
/
```

Obs: RAISE_APPLICATION_ERROR (número do erro, mensagem do erro);

-> *número do erro compreendido entre* -20000 e –20999

TRIGGERS

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

– for each row

- Esta opção, quando especificada, "dispara" o trigger em cada registo afectado pela instrução de triggering.
- A ausência desta opção indica que o trigger só é executado uma única vez para cada instrução e não separadamente para cada registo afectado
- Quando se especifica com uma condição (cláusula WHEN)
  - a condição será avaliada para todos os registos afectados pelo trigger. Se a avaliação resultar em TRUE para o registo, então a ação do trigger é executada em relação a esse registo, caso contrário não será executada.
  - A expressão na cláusula WHEN deve ser uma expressão SQL e não pode incluir subqueries;

TRIGGERS - INSTRUÇÃO

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

- Tem a finalidade de tratar a execução de ações sobre tabelas independentemente de quantas linhas forem afetadas.
- Através deste tipo de Trigger podemos registar a execução de comandos INSERT, UPDATE e DELETE contra tabelas que tenham Triggers contemplando essas ações.
- Caso um comando UPDATE atualize 1000 linhas, um Trigger deste tipo apenas dispararia 1 única vez. Este tipo de Trigger não pode referenciar qualquer valor contido em uma coluna da tabela. Isso ocorre porque o mesmo dispara uma única vez.

TRIGGERS - INSTRUÇÃO

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

```
CREATE OR REPLACE TRIGGER trg_aud_trn
BEFORE INSERT OR DELETE OR UPDATE  ON transportador
BEGIN
    IF TO_NUMBER (TO_CHAR (SYSDATE, "hh24")) NOT BETWEEN 9 AND 18
    THEN
    raise_application_error(-20001,"Operação não pode ser executada fora do horário de
    expediente.");
    END IF;
END;
/
```

TRIGGERS

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

- Síntese

	Linha	**Instrução**	**Before**	**After**
Execução	Executado sempre que uma tabela é afectada pela instrução de triggering	Executado tendo em consideração a instrução de triggering, independentemente do número de registos afectados	Executa a ação do trigger antes da instrução de triggering;	Executa a ação do trigger depois de executada a instrução de triggering
Utilidade	Se o código contido na ação do trigger depender dos dados resultantes da instrução de triggering ou dos registos afectados	Se o código na ação do trigger não depender dos dados resultantes da instrução de triggering ou dos registos afectados	Permite eliminar processamento desnecessário da instrução de triggering e o seu eventual rollback (casos em que se geram excepções na ação do trigger)	Controlar o timming dum trigger;
Aplicação		Questões de segurança relacionadas com o utilizador; Registos de Auditoria;	Cálculos de valores de colunas específicas antes da instrução de triggering (INSERT ou DELETE) estar completa	

TRIGGERS

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

– Síntese

Tipo Trigger	Caracteristicas
BEFORE instrução	A ação do trigger é executada antes da instrução de triggering;
AFTER instrução	A ação do trigger é executada depois de executada a instrução de triggering
BEFORE linha	A ação do trigger é executada: 1. de acordo com a restrição do trigger; 2. antes de cada linha ser afectada pela instrução de triggering; 3. antes da verificação das restrições de integridade.
AFTER linha	A ação do trigger é executada para cada registo de acordo com a restrição do trigger e depois de modificados os registos pela instrução de triggering. É feito o lock dos registos afectados.

TRIGGERS

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

-Aceder aos valores de atributos

- No corpo dum trigger é possível aceder aos valores antigos e novos dos atributos do registo afectado pela instrução de triggering.
- Existem dois nomes de correlação para cada coluna da tabela a ser modificada:
  - um para o valor antigo (:OLD)
    - **:OLD.nome_atributo** - indica o valor anterior de um campo que está a ser alterado por um comando DELETE ou UPDATE
  - outro para o valor novo (:NEW):
    - **:NEW.nome_atributo** . Indica um novo valor para um campo que está a ser alterado por um comando INSERT ou UPDATE

TRIGGERS - Exemplo

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

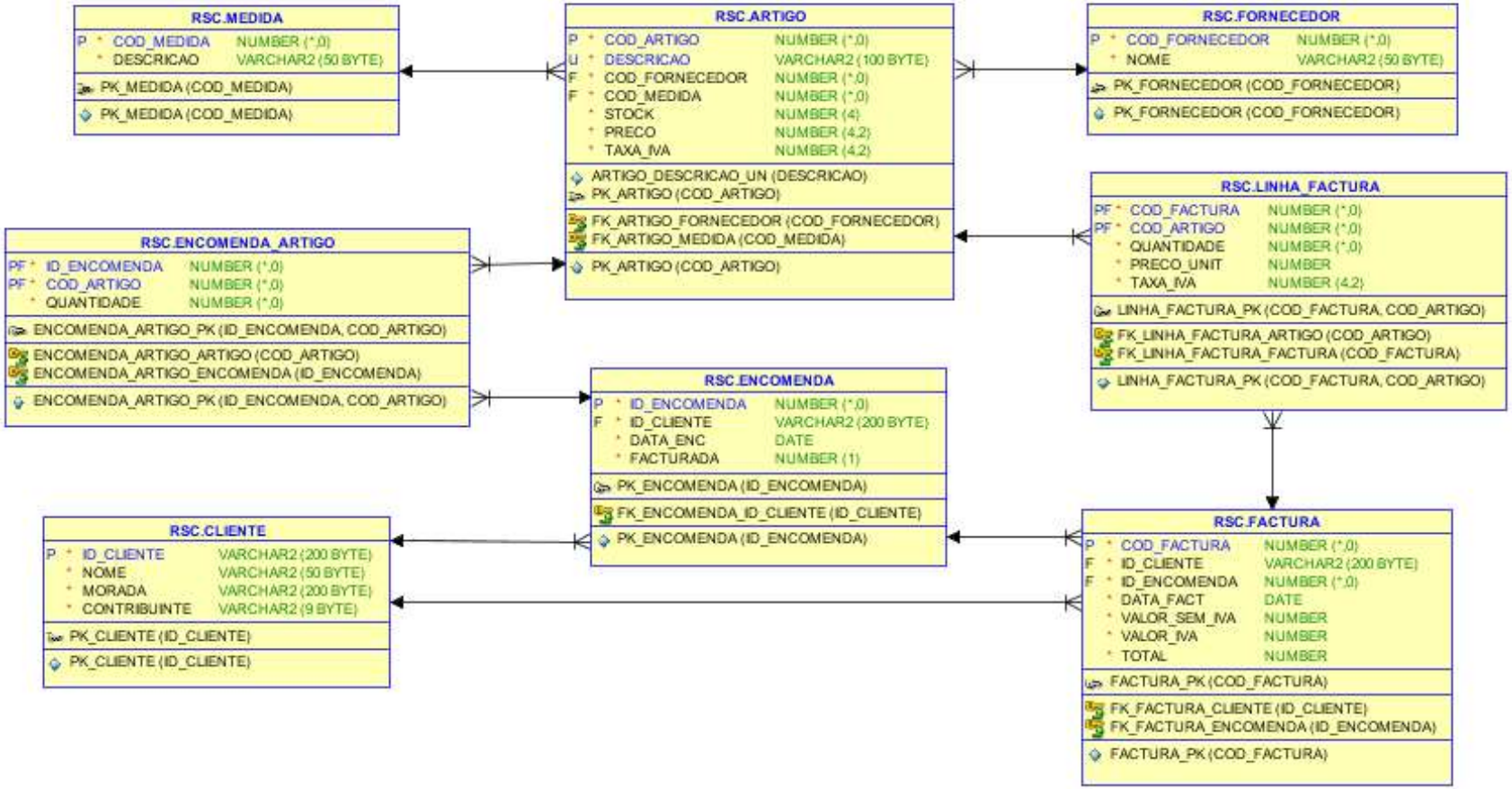
```
Set serveroutput on;  // Necessário para visualizar a saída
/* Imprimindo o valor antigo e o novo do salário */
create or replace trigger saldif
before delete or insert or update on funcionario
for each row
  declare
  sal_diff funcionario.salario%type;
  begin
                if (:new.cod_pessoa > 0)  then
                  sal_diff := :new.salario - :old.salario;
                  dbms_output.put(' antigo: '||:old.salario);
                  dbms_output.put(' novo: '||:new.salario);
                  dbms_output.put_line(' Diferença:'||sal_diff);
          end if;
       end;
/
```

Operação	:OLD	:NEW
INSERT	x	√
UPDATE	√	√
DELETE	√	x

TRIGGERS -EXERCÍCIO 1

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Criar um trigger na tabela encomenda_artigo para controloar que a quantidade de encomenda de cada artigo não ultrapasse o stock disponivel e actualiza o stock desse artigo



TRIGGERS EXERCÍCIO 1 – Resolução….

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

```
create or replace trigger VALIDA_STOCK
  before insert or update on encomenda_artigo
  for each row

  declare var_stock number;
    EX_ERRO EXCEPTION;
  BEGIN

     select stock
     into var_stock
     from Artigo
     where Artigo.cod_Artigo = :new.cod_Artigo;
  -- valida quantidade disponivel em stock
     if  (var_stock - :new.quantidade) <= 0 then
        RAISE EX_ERRO;
     end if;

   -- atualiza stock--
   update artigo set stock=(var_stock - :new.quantidade)
   where artigo.cod_artigo=:new.cod_artigo;

  EXCEPTION
    WHEN EX_ERRO THEN
      RAISE_APPLICATION_ERROR(-20006,'não existe quantidade suficiente em stock');

  END VALIDA_STOCK;
```

TRIGGERS EXERCÍCIO 1 – Teste 1 - nok

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

```
-- Testar trigger com nova encomenda 7000
INSERT INTO ENCOMENDA (ID_ENCOMENDA, ID_CLIENTE, DATA_ENC, FACTURADA)
  VALUES ('7000', '101', sysdate, '0');

select * from encomenda;
```

	ID_ENCOMENDA	ID_CLIENTE	DATA_ENC	FACTURADA
1	1000	100	18.11.18	1
2	2000	101	18.11.18	1
3	3000	102	18.11.18	1
4	4000	100	18.11.18	0
5	5000	103	18.11.18	1
6	6000	103	18.11.18	1
7	7000	101	18.11.18	0

```
select * from artigo;
-- adicionar artigo 1002 na enomenda 7000
-- com quantidade 200 e stock 100  - erro de falta de stock
INSERT INTO ENCOMENDA_ARTIGO (ID_ENCOMENDA, COD_ARTIGO, QUANTIDADE)
  VALUES ('7000', '1002', '200');
```

COD_ARTIGO	DESCRICAO	COD_FORNECEDOR	COD_MEDIDA	STOCK	PRECO	TAXA_IVA
1000	artigo 1	10	1	500	10	0,23
1001	artigo 2	11	2	1000	20	0,13
1002	artigo 3	12	3	100	30	0,06

```
INSERT INTO ENCOMENDA_ARTIGO (ID_ENCOMENDA, COD_ARTIGO, QUANTIDADE)
  VALUES ('7000', '1002', '200')
Error report -
SQL Error: ORA-20006: não existe quantidade suficiente em stock
ORA-06512: na "RSC.VALIDA_STOCK", linha 21
ORA-04088: erro durante a execução do trigger 'RSC.VALIDA_STOCK'
```

TRIGGERS EXERCÍCIO 1 – Teste 2 - ok

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

```
-- adicionar artigo 1002 com qtd 50 e stock 100  - ok e atualiza stock para 50
INSERT INTO ENCOMENDA_ARTIGO (ID_ENCOMENDA, COD_ARTIGO, QUANTIDADE)
  VALUES ('7000', '1002', '50');
```

```
select * from encomenda_artigo;
```

y Result

SQL | All Rows Fetched: 9 in 0,173 seconds

ID_ENCOMENDA	COD_ARTIGO	QUANTIDADE
1000	1000	10
1000	1001	5
1000	1002	3
2000	1000	100
2000	1001	50
3000	1000	200
5000	1000	1000
6000	1002	100
7000	1002	50

COD_ARTIGO	DESCRICAO	COD_FORNECEDOR	COD_MEDIDA	STOCK	PRECO	TAXA_IVA
1000	artigo 1	10	1	500	10	0,23
1001	artigo 2	11	2	1000	20	0,13
1002	artigo 3	12	3	100	30	0,06

COD_ARTIGO	DESCRICAO	COD_FORNECEDOR	COD_MEDIDA	STOCK	PRECO	TAXA_IVA
1000	artigo 1	10	1	500	10	0,23
1001	artigo 2	11	2	1000	20	0,13
1002	artigo 3	12	3	50	30	0,06

TRIGGERS EXERCÍCIO 1 – Teste 3 -ok

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

```
-- FACTURAR ENCOMENDA 7000 POSSUI 1 ARTIGO
UPDATE ENCOMENDA SET FACTURADA = '1' WHERE ID_ENCOMENDA='7000';
-- NOVA FACTURA COM CODIGO 6 COM 1 LINHA DE FATURA
```

COD_FACTURA	ID_CLIENTE	ID_ENCOMENDA	DATA_FACT	VALOR_SEM_IVA	VALOR_IVA	TOTAL
1	100	1000	18.11.18	290	41,4	331,4
2	101	2000	18.11.18	2000	360	2360
3	102	3000	18.11.18	2000	460	2460
4	103	5000	18.11.18	10000	2300	12300
5	103	6000	18.11.18	3000	180	3180
6	101	7000	18.11.18	1500	90	1590

COD_FACTURA	COD_ARTIGO	QUANTIDADE	PRECO_UNIT	TAXA_IVA
1	1000	10	10	0,23
1	1001	5	20	0,13
1	1002	3	30	0,06
2	1000	100	10	0,23
2	1001	50	20	0,13
3	1000	200	10	0,23
4	1000	1000	10	0,23
5	1002	100	30	0,06
6	1002	50	30	0,06

TRIGGERS EXERCÍCIO 2

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

1. Criar um Trigger na tabela Encomenda para gerar de forma automática a fatura de cada encomenda atraves da mudança de estado de encomenda não facturada (FACTURADA = 0) para encomenda facturada (FACTURADA = 1).

2. Para cada fatura criar as respectivas linhas de factura por cada artigo dessa encomenda. No caso dum cliente com várias encomendas por facturar, alterar o estado de cada encomenda para FACTURADA = 1 e criar as faturas e respectivas linhas de fatura.