

Goiânia, 29 de novembro de 2019
Aluna: Mylena Mariana Pires Barbosa 201804692
Prof.: Edmundo Spoto
Disciplina: Tópicos Especiais em Banco de Dados

MYLENA:>ed
Gravou file afiedt.buf

```
1 CREATE OR REPLACE PACKAGE comm_package IS
2     g_comm_rate NUMBER := 0.1;
3     PROCEDURE reset_comm_rate
4         (v_comm_rate IN NUMBER);
5* END comm_package;
MYLENA:>/
```

Pacote criado.

MYLENA:>ed
Gravou file afiedt.buf

```
1 CREATE OR REPLACE PACKAGE BODY comm_package IS
2     FUNCTION validade_comm_rate (v_comm_rate IN NUMBER)
3         RETURN BOOLEAN
4     IS
5         v_max_comm_rate NUMBER;
6     BEGIN
7         SELECT MAX (comm/sal)
8         INTO v_max_comm_rate
9         FROM emp;
10        IF v_comm_rate > v_max_comm_rate THEN
11            RETURN (FALSE);
12        ELSE
13            RETURN (TRUE);
14        END IF;
15    END validade_comm_rate;
16    PROCEDURE reset_comm_rate (v_comm_rate IN NUMBER)
17    IS
18        v_valid BOOLEAN;
19    BEGIN
20        v_valid := validade_comm_rate (v_comm_rate);
21        IF v_valid = TRUE THEN
22            g_comm_rate := v_comm_rate;
23        ELSE
24            RAISE_APPLICATION_ERROR (-20210, 'Taxa de comissao invalida');
25        END IF;
26    END reset_comm_rate;
27* END comm_package;
MYLENA:>/
```

Corpo de Pacote criado.

MYLENA:>EXECUTE comm_package.reset_comm_rate (.15);

Procedimento PL/SQL concluído com sucesso.

//

```
SQL> select object_name, object_type
2   from user_objects
3   where object_type in ('PROCEDURE','FUNCTION')
4   ORDER BY object_name;
```

OBJECT_NAME

OBJECT_TYPE

AVERAGE_SAL
FUNCTION

DIAS
PROCEDURE

EXCLUI_FUNCIONARIO
PROCEDURE

OBJECT_NAME

OBJECT_TYPE

INCLUI_FUNCIONARIO
PROCEDURE

LOGMNR\$COL_GG_TABF_PUBLIC
FUNCTION

LOGMNR\$GSBA_GG_TABF_PUBLIC
FUNCTION

OBJECT_NAME

OBJECT_TYPE

LOGMNR\$KEY_GG_TABF_PUBLIC
FUNCTION

LOGMNR\$SEQ_GG_TABF_PUBLIC
FUNCTION

LOGMNR\$TAB_GG_TABF_PUBLIC
FUNCTION

OBJECT_NAME

OBJECT_TYPE

LOGMNR\$USER_GG_TABF_PUBLIC
FUNCTION

NEW_EMP
PROCEDURE

NEWEMP
PROCEDURE

OBJECT_NAME

OBJECT_TYPE

NOVOS_EMPREGADOS
PROCEDURE

PESQUISA_EMPREGADO
PROCEDURE

PESQUISA_SALARIO
FUNCTION

OBJECT_NAME

OBJECT_TYPE

QTDDIAS
PROCEDURE

READEMP
PROCEDURE

SHOWEMP
PROCEDURE

OBJECT_NAME

OBJECT_TYPE

SHOWEMPNAME
PROCEDURE

SHOWSALCOMM
PROCEDURE

UPDATESAL
PROCEDURE

OBJECT_NAME

OBJECT_TYPE

UPDATESALDEPT
PROCEDURE

22 linhas selecionadas.

SQL>

////////////////////////////////////

MYLENA:>ED

Gravou file afiedt.buf

```
1  SELECT TEXT
2  FROM USER_SOURCE
3  WHERE TYPE = 'PROCEDURE'
4  AND NAME='EXCLUI_FUNCIONARIO'
5* ORDER BY LINE
6  /
```

TEXT

PROCEDURE exclui_funcionario
(v_emp_no IN emp.empno%TYPE) IS
BEGIN
 DELETE FROM EMP
 WHERE empno = v_emp_no;
 IF SQL%NOTFOUND THEN
 RAISE_APPLICATION_ERROR(-20200, 'Funcionario nao existe');
 END IF;
COMMIT WORK;
END exclui_funcionario;

10 linhas selecionadas.

////////////////////////////////////

SQL> ED

Gravou file afiedt.buf

```
1  CREATE OR REPLACE TRIGGER derive_commission_pct
2  BEFORE INSERT OR UPDATE OF sal ON emp
3  FOR EACH ROW
4  BEGIN
5      IF NOT (:NEW.JOB IN ('MANAGER','PRESIDENT'))
6          and   :NEW.SAL > 5000
7      THEN
8          RAISE_APPLICATION_ERROR
9          (-20202,'Employee cannot earn this amount');
10     END IF;
11* END;
SQL> /
```

Gatilho criado.

```
SQL> UPDATE EMP SET SAL=6500 WHERE ENAME='MILLER';
UPDATE EMP SET SAL=6500 WHERE ENAME='MILLER'
```

*

ERRO na linha 1:

ORA-20202: Employee cannot earn this amount

ORA-06512: em "SYSTEM.DERIVE_COMMISSION_PCT", line 5

ORA-04088: erro durante a execução do gatilho 'SYSTEM.DERIVE_COMMISSION_PCT'

```
MYLENA:>UPDATE EMP SET SAL=6500 WHERE ENAME='KING';
```

1 linha atualizada.

////////////////////////////////////

```
MYLENA:>ED
```

Gravou file afiedt.buf

```
1 CREATE OR REPLACE TRIGGER secure_emp
2 BEFORE INSERT ON emp
3 BEGIN
4     IF (TO_CHAR (sysdate,'DY') IN ('SAT','SUN'))
5         OR (TO_NUMBER (TO_CHAR(sysdate,'HH24')) NOT BETWEEN 8 AND 18) THEN
6         RAISE_APPLICATION_ERROR (-20500, 'Nao e possivel incluir funcionario fora
do horario normal de trabalho');
7     END IF;
8* END;
MYLENA:>/
```

Gatilho criado.

```
MYLENA:>ED
```

Gravou file afiedt.buf

```
1 CREATE OR REPLACE TRIGGER secure_emp
2 BEFORE DELETE OR INSERT OR UPDATE ON emp
3 BEGIN
4     IF (TO_CHAR (sysdate,'DY') IN ('FRY','SUN'))
5         OR (TO_NUMBER (TO_CHAR(sysdate,'HH24')) NOT BETWEEN 17 AND 18)
THEN
6         IF DELETING THEN
7             RAISE_APPLICATION_ERROR (-20502, 'Nao e possivel remover
funcionario fora do horario normal de trabalho');
8         ELSIF INSERTING THEN
9             RAISE_APPLICATION_ERROR (-20503, 'Nao e possivel inserir
funcionario fora do horario normal de trabalho');
10        ELSIF UPDATING ('SAL') THEN
11            RAISE_APPLICATION_ERROR (-20504, 'Nao e possivel alterar a
coluna salario do funcionario fora do horario normal de trabalho');
12        END IF;
13    END IF;
14* END;
```

MYLENA:>/

Gatilho criado.

MYLENA:>INSERT INTO EMP VALUES(7935, 'LIAM', 'MANAGER', 7934, SYSDATE, 2000, 0, 20);

INSERT INTO EMP VALUES(7935, 'LIAM', 'MANAGER', 7934, SYSDATE, 2000, 0, 20)
*

ERRO na linha 1:

ORA-20503: Nao e possivel inserir funcionario fora do horario normal de trabalho

ORA-06512: em "SYSTEM.SECURE_EMP", line 7

ORA-04088: erro durante a execucao do gatilho 'SYSTEM.SECURE_EMP'

MYLENA:>INSERT INTO EMP VALUES(7936, 'EMILY', 'CLERK', 7934, SYSDATE, 1000, 500, 20);

INSERT INTO EMP VALUES(7936, 'EMILY', 'CLERK', 7934, SYSDATE, 1000, 500, 20)
*

ERRO na linha 1:

ORA-20503: Nao e possivel inserir funcionario fora do horario normal de trabalho

ORA-06512: em "SYSTEM.SECURE_EMP", line 7

ORA-04088: erro durante a execucao do gatilho 'SYSTEM.SECURE_EMP'

////////////////////////////////////

MYLENA:>create table auditoria

```
2      (cd_emp number(4),
3      dt_ht_operacao date,
4      tp_operacao varchar2(1),
5      nm_usuario varchar2(30),
6      vl_sal_antigo number(10,2),
7      vl_sal_atual number(10,2))
8  /
```

Tabela criada.

MYLENA:>CREATE TABLE AUDIT_EMP_VALUES (USER_NAME VARCHAR2(10),

```
2      TIMESTAMP      DATE,
3      EMPNO           NUMBER(5),
4      OLD_NAME        VARCHAR2(20),
5      NEW_NAME        VARCHAR2(20),
6      OLD_JOB         VARCHAR2(10),
7      NEW_JOB         VARCHAR2(10),
8      OLD_MGR         NUMBER(5),
9      NEW_MGR         NUMBER(5),
10     OLD_SAL         NUMBER(12,2),
11     NEW_SAL         NUMBER(12,2));
```

Tabela criada.

MYLENA:>CREATE OR REPLACE TRIGGER audit_emp_values

```
2  AFTER DELETE OR INSERT OR UPDATE ON emp
3  FOR EACH ROW
4  BEGIN
5  INSERT INTO audit_emp_values ( user_name, timestamp, empno, old_name,
```

```

new_name, old_job, new_job, old_mgr, new_mgr, old_sal, new_sal)
  6  VALUES ( user, sysdate, :old.empno, :old.ename, :new.ename, :old.job,
:new.job, :old.mgr, :new.mgr, :old.sal, :new.sal);
  7  END;
  8  /

```

Gatilho criado.

```

MYLENA:>INSERT INTO EMP VALUES(7935, 'LIAM', 'MANAGER', 7782, SYSDATE, 2000, 0,
20);

```

1 linha criada.

```

MYLENA:>UPDATE EMP SET SAL=3000 WHERE EMPNO=7935;

```

1 linha atualizada.

```

MYLENA:>DELETE FROM EMP WHERE ENAME='LIAM';

```

1 linha excluída.

```

MYLENA:>SELECT * FROM AUDIT_EMP_VALUES;

```

USER_NAME	TIMESTAMP	EMPNO	OLD_NAME	NEW_NAME	OLD_JOB	NEW_JOB	OLD_MGR
NEW_MGR	OLD_SAL	NEW_SAL					
-----	-----	----	-----		-----		
-----	-----	-----	-----				
SYSTEM	29/11/19			LIAM		MANAGER	
7782		2000					
SYSTEM	29/11/19	7935	LIAM	LIAM	MANAGER	MANAGER	7782
7782	2000	3000					
SYSTEM	29/11/19	7935	LIAM		MANAGER		7782
	3000						

```

////////////////////////////////////
trigger_0

```

```

MYLENA:>ED
Gravou file afiedt.buf

```

```

 1  CREATE OR REPLACE TRIGGER INS_AUDI
 2  AFTER INSERT OR DELETE OR UPDATE ON EMP
 3  FOR EACH ROW
 4  BEGIN
 5      IF INSERTING THEN
 6          INSERT INTO AUDITORIA
 7          VALUES(:NEW.EMPNO,SYSDATE,'I',USER,0,:NEW.SAL);
 8      ELSIF DELETING THEN
 9          INSERT INTO AUDITORIA
10          VALUES(:NEW.EMPNO,SYSDATE,'D',USER,:OLD.SAL,0);
11      ELSIF UPDATING THEN
12          INSERT INTO AUDITORIA

```

```

13          VALUES(:NEW.EMPNO,SYSDATE,'D',USER,:OLD.SAL,:NEW.SAL);
14      END IF;
15*  END;
MYLENA:>/

```

Gatilho criado.

```

MYLENA:>INSERT INTO EMP VALUES(7935, 'LIAM', 'MANAGER', 7782, SYSDATE, 2000, 0,
20);

```

1 linha criada.

```

MYLENA:>UPDATE EMP SET SAL=3000 WHERE EMPNO=7935;

```

1 linha atualizada.

```

MYLENA:>DELETE FROM EMP WHERE ENAME='LIAM';

```

1 linha excluída.

```

MYLENA:>SELECT * FROM AUDITORIA;

```

CD_EMP	DT_HT_OP	T	NM_USUARIO	VL_SAL_ANTIGO	VL_SAL_ATUAL
7935	29/11/19	I	SYSTEM	0	2000
7935	29/11/19	D	SYSTEM	2000	3000
	29/11/19	D	SYSTEM	3000	0

```

////////////////////////////////////
@TRIGGER_02

```

```

MYLENA:> CREATE OR REPLACE TRIGGER SAL_INF_20
2  BEFORE UPDATE ON EMP
3  FOR EACH ROW WHEN (NEW.SAL>OLD.SAL*1.2)
4  BEGIN
5      RAISE_APPLICATION_ERROR(-20000,'NAO E PERMITIDO AUMENTAR SALARIO
6      COM VALOR SUPERIOR A 20% DO SALARIO ANTERIOR');
7  END;
8  /

```

Gatilho criado.

```

MYLENA:>SELECT * FROM EMP;

```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
7839	KING	PRESIDENT		17/11/81	5000	
7566	JONES	MANAGER	7839	02/04/81	2975	
7788	SCOTT	ANALYST	7566	19/04/87	2000	

20	7876	ADAMS	CLERK	7788	23/05/87	1100	
20	7902	FORD	ANALYST	7566	03/12/81	3000	
20	7369	SMITH	CLERK	7902	17/12/80	800	
30	7698	BLAKE	MANAGER	7839	01/05/81	2850	
30	7499	ALLEN	SALESMAN	7698	20/02/81	1600	300
30	7521	WARD	SALESMAN	7698	22/02/81	1250	500
30	7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400
30	7844	TURNER	SALESMAN	7698	08/09/81	1500	0
30	7900	JAMES	CLERK	7698	03/12/81	950	
10	7782	CLARK	MANAGER	7839	09/06/81	2450	
10	7934	MILLER	CLERK	7782	23/01/82	1300	

14 linhas selecionadas.

```
MYLENA:>UPDATE EMP SET SAL=3000 WHERE EMPNO=7934;
UPDATE EMP SET SAL=3000 WHERE EMPNO=7934
```

*

ERRO na linha 1:

ORA-20000: NAO E PERMITIDO AUMENTAR SALARIO
COM VALOR SUPERIOR A 20% DO SALARIO ANTERIOR

ORA-06512: em "SYSTEM.SAL_INF_20", line 2

ORA-04088: erro durante a execucao do gatilho 'SYSTEM.SAL_INF_20'

```
////////////////////////////////////
TRIGGER_07
```

MYLENA:>ED

Gravou file afiedt.buf

```
1  create or replace trigger sal_maior_gerente
2  after update or insert of sal on emp
3  for each row
4  declare
5      v_sal    emp.sal%TYPE;
6  begin
7      SELECT sal INTO v_sal from emp where empno = :new.mgr;
8      IF (:new.sal > v_sal) THEN
9          IF UPDATING('SAL') THEN
10             RAISE_APPLICATION_ERROR (-20701, 'salario Atualizado nao
11             pode ser maior que o salario do gerente');
12          ELSIF INSERTING THEN
13             RAISE_APPLICATION_ERROR (-20702, 'salario Inserido não
14             pode ser superior ao salario do gerente');
```

```

15      end if;
16      end if;
17* end;
MYLENA:>/

```

Gatilho criado.

```
MYLENA:>select * from emp;
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM
DEPTNO	-----						

	7839	KING	PRESIDENT		17/11/81	5000	
10	7566	JONES	MANAGER	7839	02/04/81	2975	
20	7788	SCOTT	ANALYST	7566	19/04/87	2000	
20	7876	ADAMS	CLERK	7788	23/05/87	1100	
20	7902	FORD	ANALYST	7566	03/12/81	3000	
20	7369	SMITH	CLERK	7902	17/12/80	800	
20	7698	BLAKE	MANAGER	7839	01/05/81	2850	
30	7499	ALLEN	SALESMAN	7698	20/02/81	1600	300
30	7521	WARD	SALESMAN	7698	22/02/81	1250	500
30	7654	MARTIN	SALESMAN	7698	28/09/81	1250	1400
30	7844	TURNER	SALESMAN	7698	08/09/81	1500	0
30	7900	JAMES	CLERK	7698	03/12/81	950	
30	7782	CLARK	MANAGER	7839	09/06/81	2450	
10	7934	MILLER	CLERK	7782	23/01/82	1300	
10							

14 linhas selecionadas.