**VIEWS**

|  |
| --- |
| CREATE VIEW <nome da view> AS SELECT...; |

Cria uma tabela temporária ;

Ex:

CREATE VIEW tabelaAnorexica AS

SELECT e.cpf, e.salario, g.codLaboratorio FROM empregado e, engenheiro g

WHERE e.cpf = g.cpf;

CREATE VIEW tabelaAvuadora AS

SELECT ....;

**TRIGGERS**

|  |
| --- |
| CREATE [OR REPLACE] TRIGGER <nome da tabela> [BEFORE, AFTER or INSTEAD OF] [INSERT, DELETE or UPDATE] ON <nome da tabela>REFERENCING NEW AS <novo\_nome> OLD AS <antigo\_nome> FOR EACH ROW or STATEMENT[DECLARE]...BEGINEND |

Se usa gatilhos para garantir que quando uma operação é processada, ações relacionadas são executadas.

Ex:

CREATE TABLE logDemitidos

( nome varchar2(30);

 Salário NUMBER;

 Data TIMESTAMP;

);

CREATE TRIGGER trigao AFTER DELETE ON empregado

FOR EACH ROW

BEGIN

INSERT INTO logDemitidos (nome, salario, data) VALUES (:OLD.nome, :OLD.salario,

SYSTIMESTAMP);

END;

/

CREATE TRIGGER log\_lab AFTER INSERT ON laboratorio

FOR EACH STATEMENT

BEGIN

INSERT INTO log\_lab(nome, codigo,data) VALUES (:OLD…..

**BLOCOS**

|  |
| --- |
| DECLARE -- declaraçõesBEGIN -- instruçõesEXCEPTION --manipulação de errosEND; |

DECLARE

v\_local varchar2(30) = 'olaenfermeira';

v\_custo NUMBER(10) = 99999;

BEGIN

UPDATE local SET custo=0 WHERE local='tupi';

EXCEPTION

 IF SQL%NOTFOUND THEN

 INSERT INTO local(local,custo) VALUES(v\_local,v\_custo);

 END IF;

END;

**LAÇOS**

|  |  |  |
| --- | --- | --- |
| LOOP<instruções>END LOOP; | FOR <contador> IN <inferior><superior> LOOP<instruções>END LOOP; | WHILE <condição>LOOP<instruções>END LOOP |

**FUNCTIONS**

|  |
| --- |
| CREATE OR REPLACE FUNCTION <nome da função> (<nome da variável> IN <nome da tabela.atributo>%TYPE)RETURN <tipo do retorno> IS <tipo instanciado do retorno>BEGIN...INTO <variável de retorno>...RETURN <variável de retorno>END <nome da função>;/ |

CREATE OR REPLACE FUNCTION localiza\_empregado(v\_nome IN empregado.nome%TYPE)

RETURN VARCHAR2 IS retorno VARCHAR(30);

BEGIN

SELECT nome INTO retorno FROM empregado WHERE nome=v\_nome;

RETURN retorno;

END localiza\_empregado;

/

CREATE OR REPLACE FUNCTION pega\_cpf(v\_cpf IN engenheiro.cpf%TYPE)

RETURN NUMBER IS retorno NUMBER(10);

BEGIN

SELECT cpf INTO retorno FROM engenheiro WHERE v\_cpf=cpf;

RETURN retorno;

END pega\_cpf;

/

CREATE OR REPLACE FUNCTION pega\_codigo(v\_codigo IN produto.codigo%TYPE)

RETURN NUMBER IS retorno NUMBER(10);

BEGIN

SELECT codigo INTO retorno FROM produto WHERE v\_codigo=codigo;

RETURN retorno;

END pega\_codigo;

/