- Configurar acesso ao Oracle:

- TNSNames.ora

- Caminho padrão: \app\orcl\product\11.2.0\client\_1\Network\Admin

- Criar um alias diferente dos já existentes;

- Informar o IP do servidor;

- Informar a porta do servidor (padrão 1521);

- Informar o nome do serviço.

- Exemplo:

HINOERP =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = **192.168.1.150**)(PORT = **1521**))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = **ORCL**)

)

)

- Testar a conexão: TNSPING NOME\_HOST. Exemplo: TNSPING HINOERP.

- Acesso via texto ao banco de dados:

- Programa: sqlplus;

- Caminho padrão: \app\orcl\product\11.2.0\client\_1\bin

- Acessar: sqlplus USUARIO/SENHA@HOST.

Exemplo: sqlplus HN\_DESENV/hino@HINOERP

SET LINESIZE 255;

SET PAGESIZE 200;

- Exemplos de selects:

-- TODAS AS TABELAS DO USUÁRIO

SELECT \*

FROM USER\_TABLES

-- TODAS AS COLUNAS DE UMA TABELA

SELECT \*

FROM USER\_TAB\_COLUMNS

WHERE TABLE\_NAME = 'GEESTAB'

-- TODOS OS COMENTÁRIOS DE UMA TABELA

SELECT \*

FROM USER\_TAB\_COMMENTS

WHERE TABLE\_NAME = 'GEESTAB'

-- TODOS OS COMENTÁRIOS DE UMA TABELA/COLUNA

SELECT \*

FROM USER\_COL\_COMMENTS

WHERE TABLE\_NAME = 'GEESTAB'

- Teclas de atalho do TOAD:

F9: executa o comando selecionado;

Shift+F9: executa o comando da linha do cursor;

F5: executa todos os comandos da tela (finalizar os comandos com “;”);

F4: exibe formulário com o detalhe do objeto em que o cursor esteja posicionado;

Debug:

Shift+F5: breakpoint;

F11: Rodar

Shift+F8: passo a passo;

Shift+F7: acessa procedimento chamado;

Shift+F12: roda até o cursor.

Exemplos:

**Observação**: no from colocar as tabelas maiores (com mais registros/colunas em primeiro)

-- IDENTAÇÃO DE SELECT

SELECT COLUNAS -- ATÉ 4 POR LINHA

FROM TABELAS -- ATÉ 4 POR LINHA

WHERE

AND

GROUP BY

ORDER BY

-- SEMELHANTE AO ISNULL DO SQLSERVER

SELECT CODPAIS, NVL(CODPAIS, 0)

FROM GEMOEDA

-- LEFT JOIN

SELECT \*

FROM GEMOEDA, GEPAIS

WHERE GEPAIS.CODPAIS (+)= GEMOEDA.CODPAIS

-- DECODE E CASE

SELECT ATIVO,

DECODE(ATIVO, 1, 'SIM', 'NAO')

FROM GEESTAB

SELECT STATUS,

DECODE(STATUS, 'P', 'Pendente',

'A', 'Aprovada',

'E', 'Encerrada',

'R', 'Reprovada') DESCSTATUS,

CASE

WHEN (STATUS = 'P') THEN 'Pendente'

WHEN (STATUS = 'A') THEN 'Aprovada'

WHEN (STATUS = 'E') THEN 'Encerrada'

WHEN (STATUS = 'R') THEN 'Reprovada'

END CASESTATUS

FROM MNOS

-- CONCATENAR TEXTO ( O + DO SQLSERVER)

SELECT SIGLA || ' - ' || NOME

FROM GEPAIS

-- CRIAR UMA TABELA A PARTIR DE OUTRA

CREATE TABLE GEESTAB\_BKP AS SELECT \* FROM GEESTAB;

SELECT \*

FROM GEESTAB\_BKP

-- CONNECT BY E START WITH

CREATE TABLE ESTRUTURA

(

PAI NUMBER,

FILHO NUMBER

);

INSERT INTO ESTRUTURA (PAI,FILHO) VALUES (1, 2);

INSERT INTO ESTRUTURA (PAI,FILHO) VALUES (2, 3);

INSERT INTO ESTRUTURA (PAI,FILHO) VALUES (3, 4);

INSERT INTO ESTRUTURA (PAI,FILHO) VALUES (4, 5.1);

INSERT INTO ESTRUTURA (PAI,FILHO) VALUES (4, 5.2);

INSERT INTO ESTRUTURA (PAI,FILHO) VALUES (1, 2.1);

SELECT PAI, FILHO, NIVEL, DECODE(NIVEL, 1, '', LPAD(' ', (NIVEL - 1) \* 3, ' ')) || PAI ARVORE

FROM (

SELECT PAI, FILHO, LEVEL NIVEL

FROM ESTRUTURA

CONNECT BY PAI = PRIOR FILHO

START WITH PAI = 1)

SELECT PAI, FILHO, PAI + FILHO, PAI || FILHO

FROM ESTRUTURA

-- TABELA DUAL

SELECT \*

FROM DUAL

-- DATA E CONVERSOES

SELECT SYSDATE FROM DUAL -- DATA E HORA ATUAL

SELECT TRUNC(SYSDATE) FROM DUAL -- SOMENTE A DATA

-- TRUNC SERVE PARA "CORTAR" TANTO DATA COMO CASAS DECIMAIS

-- CONVERSOES

SELECT TO\_CHAR(SYSDATE, 'HH24:MI:SS') FROM DUAL

SELECT TO\_DATE(TO\_CHAR(SYSDATE, 'DD/MM/YYYY'), 'DD/MM/YYYY') FROM DUAL

SELECT TO\_DATE('15/12/2013', 'DD/MM/YYYY') FROM DUAL

SELECT TRUNC(15.987654321) FROM DUAL -- NAO ARREDONDA

SELECT ROUND(15.987654321, 0) FROM DUAL -- ARREDONDA SEM CASAS DECIMAIS

SELECT ROUND(15.987654321, 2) FROM DUAL -- ARREDONTA COM CASAS DECIMAIS

SELECT CEIL(15.123456789) FROM DUAL -- ARREDONDA SEMPRE PARA CIMA (RETORNANDO INTEIRO)

SELECT FLOOR(15.987654321) FROM DUAL -- ARREDONDA SEMPRE PARA BAIXO (RETORNANDO INTEIRO)

SELECT TO\_CHAR(SYSDATE, 'd') FROM DUAL -- DIA DA SEMANA (INICIANDO NO DOMINGO)

SELECT TO\_CHAR(SYSDATE, 'day') FROM DUAL -- DIA POR EXTENSO

SELECT TO\_CHAR(SYSDATE, 'MONTH'), TO\_CHAR(SYSDATE, 'month') FROM DUAL -- DIA POR EXTENSO

SELECT TRUNC(LAST\_DAY(SYSDATE)) FROM DUAL -- ULTIMA DIA

SELECT TO\_CHAR(LAST\_DAY(SYSDATE), 'd') FROM DUAL -- ULTIMA DIA

SELECT TO\_CHAR(TO\_DATE('01/' || TO\_CHAR(SYSDATE, 'MM/YYYY'), 'DD/MM/YYYY'), 'D') FROM DUAL

SELECT TO\_CHAR(LAST\_DAY(ADD\_MONTHS(SYSDATE, -1)) + 1, 'D') FROM DUAL

SELECT TO\_NUMBER('1,2') FROM DUAL -- CONVERTER PARA NUMERO

SELECT TO\_CHAR(1.2, '9D00') FROM DUAL -- CONVERTER PARA NUMERO

-- MERGE (INSERT E UPDATE EM UM MESMO COMANDO)

MERGE INTO GEESTAB

USING (SELECT 5 CODESTAB,

'ANGELO MARCOLIN' RAZAOSOCIAL,

'ANGELO' NOMEFANTASIA,

1 CODENDERECO,

1 CODMOEDA,

1 ATIVO

FROM DUAL) DADOS

ON (DADOS.CODESTAB = GEESTAB.CODESTAB)

WHEN MATCHED THEN

UPDATE SET GEESTAB.NOMEFANTASIA = DADOS.NOMEFANTASIA

WHEN NOT MATCHED THEN

INSERT

(GEESTAB.CODESTAB,

GEESTAB.RAZAOSOCIAL,

GEESTAB.NOMEFANTASIA,

GEESTAB.CODENDERECO,

GEESTAB.CODMOEDA,

GEESTAB.ATIVO

)

VALUES

(DADOS.CODESTAB,

DADOS.RAZAOSOCIAL,

DADOS.NOMEFANTASIA,

DADOS.CODENDERECO,

DADOS.CODMOEDA,

DADOS.ATIVO);

-- PROCEDURES, FUNCTIONS, PACKAGES

CREATE TABLE TESTE

(

TABELA VARCHAR2(25),

TOTLIN NUMBER

);

DECLARE

nTOTLIN NUMBER;

BEGIN

SELECT COUNT(\*)

INTO nTOTLIN

FROM GEESTAB;

INSERT INTO TESTE

(TABELA, TOTLIN)

VALUES

('GEESTAB', nTOTLIN);

COMMIT;

END;

DECLARE

nTOTLIN NUMBER;

BEGIN

FOR R IN (SELECT TABLE\_NAME

FROM USER\_TABLES) LOOP

EXECUTE IMMEDIATE 'SELECT COUNT(\*) FROM ' || R.TABLE\_NAME INTO nTOTLIN;

INSERT INTO TESTE

(TABELA, TOTLIN)

VALUES

(R.TABLE\_NAME, nTOTLIN);

END LOOP;

COMMIT;

END;

--EXEMPLO:

CREATE OR REPLACE PROCEDURE PROC\_TESTE

(

pNOMETAB IN VARCHAR2

)

IS

BEGIN

IF (pNOMETAB = 'GEESTAB') THEN

INSERT INTO TESTE

(TABELA, TOTLIN)

VALUES

(pNOMETAB, (SELECT COUNT(\*) FROM GEESTAB));

ELSIF (pNOMETAB = 'GEMOEDA') THEN

INSERT INTO TESTE

(TABELA, TOTLIN)

VALUES

(pNOMETAB, (SELECT COUNT(\*) FROM GEMOEDA));

END IF;

COMMIT;

END PROC\_TESTE;

BEGIN

PROC\_TESTE('GEMOEDA');

END;

SELECT \*

FROM TESTE

CREATE OR REPLACE FUNCTION FCT\_TESTE

(

pNOMETAB IN VARCHAR2

) RETURN NUMBER

IS

nTOTLIN NUMBER;

-- vTESTE VARCHAR2(1);

BEGIN

IF (pNOMETAB = 'GEESTAB') THEN

SELECT COUNT(\*)

INTO nTOTLIN

FROM GEESTAB;

ELSIF (pNOMETAB = 'GEMOEDA') THEN

SELECT COUNT(\*)

INTO nTOTLIN

FROM GEMOEDA;

END IF;

RETURN nTOTLIN;

END FCT\_TESTE;

SELECT FCT\_TESTE('GEMOEDA') RET\_FUNCAO FROM DUAL

CREATE OR REPLACE FUNCTION FCT\_TOTMOEDAPAIS

(

pCODPAIS IN NUMBER

) RETURN NUMBER

IS

nTOTAL NUMBER;

BEGIN

nTOTAL := 0;

FOR R IN (SELECT CODPAIS

FROM GEMOEDA) LOOP

IF (R.CODPAIS = pCODPAIS) THEN

nTOTAL := nTOTAL + 1;

END IF;

END LOOP;

RETURN nTOTAL;

END FCT\_TOTMOEDAPAIS;

SELECT FCT\_TOTMOEDAPAIS(1) FROM DUAL

CREATE OR REPLACE FUNCTION FCT\_TOTMOEDAPAIS2

(

pCODPAIS IN NUMBER

) RETURN NUMBER

IS

nTOTAL NUMBER;

CURSOR CUR\_MOEDA IS

SELECT CODPAIS

FROM GEMOEDA;

BEGIN

nTOTAL := 0;

FOR R IN CUR\_MOEDA LOOP

IF (R.CODPAIS = pCODPAIS) THEN

nTOTAL := nTOTAL + 1;

END IF;

END LOOP;

RETURN nTOTAL;

END FCT\_TOTMOEDAPAIS2;

SELECT FCT\_TOTMOEDAPAIS2(1) FROM DUAL

SELECT COUNT(\*)

FROM GEMOEDA

WHERE CODPAIS = 1

-- EXEMPLO DE TRIGGER

CREATE OR REPLACE TRIGGER TRG\_GEESTAB\_IUD

BEFORE INSERT OR UPDATE OR DELETE ON GEESTAB

FOR EACH ROW

BEGIN

-- :OLD.CODESTAB -> VALOR ANTIGO

-- :NEW.CODESTAB -> VALOR NOVO

IF INSERTING THEN

NULL;

ELSIF UPDATING THEN

NULL;

ELSIF DELETING THEN

NULL;

END IF;

END;

/

-- FUNÇÕES ANALÍTICAS

CREATE TABLE TAB\_TESTE

(

CODIGO NUMBER(2) NOT NULL,

SUBGRUPO NUMBER(2),

DESCRICAO VARCHAR2(50),

VALOR NUMBER,

CONSTRAINT PK\_TAB\_TESTE PRIMARY KEY (CODIGO)

);

INSERT INTO TAB\_TESTE (CODIGO, SUBGRUPO, DESCRICAO, VALOR) VALUES (1, 1, 'VALOR 1', 10);

INSERT INTO TAB\_TESTE (CODIGO, SUBGRUPO, DESCRICAO, VALOR) VALUES (2, 1, 'VALOR 2', 20);

INSERT INTO TAB\_TESTE (CODIGO, SUBGRUPO, DESCRICAO, VALOR) VALUES (3, 1, 'VALOR 3', 30);

INSERT INTO TAB\_TESTE (CODIGO, SUBGRUPO, DESCRICAO, VALOR) VALUES (4, 2, 'VALOR 4', 40);

INSERT INTO TAB\_TESTE (CODIGO, SUBGRUPO, DESCRICAO, VALOR) VALUES (5, 2, 'VALOR 5', 50);

COMMIT;

SELECT CODIGO, SUBGRUPO, DESCRICAO, VALOR,

SUM(VALOR) OVER (PARTITION BY SUBGRUPO ORDER BY CODIGO) VALOR\_ACUMULADO\_SUBGRUPO,

SUM(VALOR) OVER (ORDER BY CODIGO) VALOR\_ACUMULADO\_GERAL,

LEAD(VALOR) OVER (ORDER BY CODIGO) PROXIMO\_VALOR

FROM TAB\_TESTE

ORDER BY SUBGRUPO, CODIGO

Mais funcoes em: http://www.oracle-base.com/articles/misc/analytic-functions.php

-- Sequências

CREATE TABLE TESTESEQ

(

CODIGO NUMBER,

DESCRICAO VARCHAR2(50),

CONSTRAINT PKTESTESEQ PRIMARY KEY (CODIGO)

)

CREATE SEQUENCE SEQ\_TESTESEQ

START WITH 1

MAXVALUE 99999

MINVALUE 1

NOCYCLE

NOCACHE

NOORDER;

INSERT INTO TESTESEQ

(CODIGO, DESCRICAO)

VALUES

(SEQ\_TESTESEQ.NEXTVAL, 'UM');

SELECT SEQ\_TESTESEQ.NEXTVAL

FROM DUAL

INSERT INTO TESTESEQ

(CODIGO, DESCRICAO)

VALUES

(SEQ\_TESTESEQ.NEXTVAL, 'TRES');

-- Exemplos

-- PEGAR PARTE DA STRING

SELECT SUBSTR('ABCDEF', 1, 3) FROM DUAL

SELECT SUBSTR('ABCDEF', 3) FROM DUAL

-- LOCALIZAR UM TEXTO NA STRING

SELECT INSTR('123\*ABCDEFA\*', '\*') FROM DUAL

-- LOCALIZAR A SEGUNDA REPETIÇAO DE UM TEXTO NA STRING

SELECT INSTR('123\*ABCDEFA\*', '\*', 1, 2) FROM DUAL

SELECT SUBSTR('123\*ABCDEFA\*', INSTR('123\*ABCDEFA\*', '\*') + 1, INSTR('123\*ABCDEFA\*', '\*', 1, 2) - INSTR('123\*ABCDEFA\*', '\*') - 1) FROM DUAL

-- TAMANHO DA STRING

SELECT LENGTH('123\*ABCDEFA\*') FROM DUAL

-- COMPLETAR A STRING

SELECT LPAD('1234', 10) FROM DUAL

SELECT LPAD('1234', 10, '0') FROM DUAL

SELECT RPAD('ABCD', 10) || 'Z' FROM DUAL

-- TROCAR CARACTER

SELECT REPLACE('123\*ABCDEFA\*', 'A', 'Z') FROM DUAL

SELECT REPLACE('123\*ABCDEFA\*', 'A') FROM DUAL

-- COMPRAR QUAL O MAIOR VALOR

SELECT GREATEST(10, 20, 30, 40) FROM DUAL

SELECT GREATEST('A', 'B') FROM DUAL

- **Dump ou DMP (backup):**

Exportar:

EXP USUARIO/SENHA@HOST FILE=NOME\_ARQ.DMP LOG=NOME\_ARQ.LOG

Importar (criar o usuário antes):

IMP SYSTEM/SENHA@HOST FROMUSER=NOME\_USUARIO\_EXP TOUSER=NOVOUSUARIO FILE=NOME\_ARQ.DMP LOG=NOME\_ARQ\_IMP.LOG

-- Exceção

BEGIN

BEGIN

INSERT INTO GEESTAB

(CODESTAB, RAZAOSOCIAL, NOMEFANTASIA, CODENDERECO,

CODMOEDA, ATIVO)

VALUES

(100, 'A', 'A', 200,

1, 1);

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

RAISE\_APPLICATION\_ERROR(-20001, 'REGISTRO JA EXISTENTE');

WHEN OTHERS THEN

RAISE;

END;

END;

-- Visão

CREATE OR REPLACE VIEW GEESTADOPAIS

(

CODESTADO,

NOMEESTADO,

CODPAIS,

NOMEPAIS

)

AS

SELECT GEESTADO.CODESTADO, GEESTADO.NOME, GEESTADO.CODPAIS, GEPAIS.NOME NOMEPAIS

FROM GEESTADO, GEPAIS

WHERE GEPAIS.CODPAIS = GEESTADO.CODPAIS;

SELECT \*

FROM GEESTADOPAIS

WHERE CODESTADO = 1

-- Global temporary table

CREATE GLOBAL TEMPORARY TABLE GTT

(

CODIGO NUMBER,

DESCRICAO VARCHAR2(60)

) ON COMMIT PRESERVE ROWS;

INSERT INTO GTT

(CODIGO, DESCRICAO)

VALUES

(1, 'UM');

COMMIT;

SELECT \*

FROM GTT;

CREATE GLOBAL TEMPORARY TABLE GTTD

(

CODIGO NUMBER,

DESCRICAO VARCHAR2(60)

) ON COMMIT DELETE ROWS;

INSERT INTO GTTD

(CODIGO, DESCRICAO)

VALUES

(1, 'UM');

SELECT \*

FROM GTTD

COMMIT;

SELECT \*

FROM GTTD;