

- 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)

```
-- IDENTIFICAÇÃO 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 REPETIÇÃO 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 CHARACTER
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 FROM USER=NOME_USUARIO_EXP TO USER=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;

```