**CAPÍTULO 1 – Introdução**

**1. CRIANDO TABELA**

CREATE TABLE [nome\_da\_tabela] ([nome\_do\_campo] [tipo] [tamanho] NOT NULL ,PRIMARY KEY[nome\_do\_campo]);

**Exemplo:**

*CREATE TABLE TALUNO*

*(*

*COD\_ALUNO INTEGER NOT NULL,*

*NOME VARCHAR(30),*

*CIDADE VARCHAR2(30),*

*CEP VARCHAR(10),*

*PRIMARY KEY (COD\_ALUNO)*

*);*

ou:

CREATE TABLE [nome\_da\_tabela] ([nome\_do\_campo] NOT NULL PRIMARY KEY [tipo] [tamanho]);

Exemplo:

*CREATE TABLE TCURSO(*

*COD\_CURSO INTEGER NOT NULL PRIMARY KEY,*

*NOME VARCHAR(30),*

*VALOR NUMBER(8,2),*

*CARGA\_HORARIA INTEGER*

*);*

2. INSERINDO DADOS NA TABELA

INSERT INTO [NOME\_DA\_TABELA]([campo1],[campo2],[campo3]) VALUES('dado1','dado2',dado3);

Exemplo:

*INSERT INTO TALUNO(COD\_ALUNO,NOME,CIDADE,CEP)*

*VALUES (1,'EMERSON','NOVO HAMBURGO','93004900');*

ou:

INSERT INTO VALUES('dado1','dado2',dado3);

Exemplo:

*INSERT INTO TCURSO VALUES(1,'ORACLE SQL e PL/SQL',500,25);*

2.1 Inserindo valor padrão em uma coluna:

***ALTER TABLE*** *TALUNO* ***ADD*** *ESTADO* ***CHAR(2) DEFAULT*** *'RS'****;***

A coluna será inserida e preenchida com o valor específico padrão.

3. INSERINDO COLUNA, DROPANDO E RENOMEANDO COLUNA (ALTER TABLE)

3.1. Inserindo Coluna:

ALTER TABLE [nome\_da\_table] ADD [nomde\_da\_coluna][atributo1][atributo2];

Exemplo:

*ALTER TABLE TCONTRATO ADD COL\_TESTE VARCHAR(50);*

3.2. Dropando Coluna:

ALTER TABLE [nome\_da\_tabela] DROP COLUMN [nome\_da\_coluna];

Exemplo:

*ALTER TABLE TCONTRATO DROP COLUMN COL\_TESTE;*

3.3. Renomeando Coluna:

ALTER TABLE [nome\_da\_tabela] RENAME COLUMN [nome\_da\_coluna] TO [novo\_nome\_col];

Exemplo:

*ALTER TABLE TCONTRATO RENAME COLUMN COL\_TESTE TO COLUNA\_TESTE;*

4. SEQUENCE

Usado para criar uma sequencia de valores a serem inseridos automaticamente a partir do valor específico.

CREATE SEQUENCE SEQ\_[nome\_tabela] START WITH [valor]

Exemplo:

*CREATE SEQUENCE SEQ\_ALUNO START WITH 4;*

4.1. INSERINDO O VALOR COM O SEQUENCE CRIADO:

INSERT INTO [nome\_tabela] VALUE([nome\_sequence].NEXTVAL,[valores]);

Exemplo:

*INSERT INTO TALUNO(COD\_ALUNO,NOME,CIDADE,CEP)*

*VALUES (SEQ\_ALUNO.NEXTVAL,’EMILLY’,’IVOTI’,’599388388’);*

**5. SETANDO DATA ATUAL E VALOR ATUAL DO SEQUENCIA**

Data/Hora Atual

*SELECT SYSDATE FROM DUAL;*

Valor Atual da Sequencia

*SELECT SEQ\_ALUNO.CURRVAL FROM DUAL;*

6. EXCLUIR TABELAS, DESFAZER TUDO, CONFIRMAR TUDO

Excluir Tabela

*DROP TABLE [nome\_da\_tabela];*

Desfazer todas as ações ou comandos executados

*ROLLBACK;*

Confirmar todos os comandos executados

*COMMIT;*

7. EXCLUIR UM E TODOS OS REGISTROS DA TABELA

7.1. Excluir um registro:

*DELETE FROM TALUNO*

*WHERE COD\_ALUNO = 1;*

7.1. Excluir todos os registros:

*DELETE FROM TALUNO;*

8. SELECIONANDO POR VALOR ESPECÍFICO

*SELECT \* FROM TALUNO WHERE NOME = 'EMERSON';*

9. ALTERAR REGISTRO DA TABLEA

Exemplo:

Alterando um valor:

***UPDATE*** *TCONTRATO* ***SET*** *DESCONTO = 5* ***WHERE*** *COD\_CONTRATO = 1;*

ou

Alterando vários valores:

***UPDATE*** *TCONTRATO* ***SET*** *DESCONTO = 20****,*** *TOTAL = 750* ***WHERE*** *COD\_CONTRATO = 2****;***

Alterando valor com especificação maiúscula ou minúscula:

***UPDATE*** *TALUNO* ***SET***

*CIDADE = 'CANOAS'*

***WHERE Upper****(CIDADE) = 'BULGARIA'****;***

**CAPÍTULO 2 – EXECUTANDO COMANDOS SQL BÁSICOS**

SELECT [DISTINCT] {\*,column [alias], ...}

FROM table;

SELECT - é uma lista de uma ou mais colunas.

DISTINCT – suprime duplicidades.

\* - seleciona todas as colunas.

column – seleciona a coluna nomeada.

alias – fornece para as colunas selecionadas títulos diferentes.

FROM table – especifica a tabela que contém colunas.

1. Escrevendo comandos SQL

- **Comandos** SQL não fazem distinção entre maiúscula e minúscula.

- Comandos SQL podem estar em uma ou mais linhas.

- **Palavras chaves** (keywords) não podem ser abreviadas ou divididas em mais de uma linha.

- **Cláusulas** são normalmente colocadas em linhas separadas.

- **Tabulações** e identações são utilizadas para melhorar a visualização do comando.

Exemplos:

- Trazer todos os dados da tabela:

***SELECT*** *\** ***FROM*** *TALUNO****;***

- Trazer somente os dados de uma coluna:

***SELECT*** *COD\_ALUNO* ***FROM*** *TALUNO****;***

- Trazer dados de mais de uma coluna:

***SELECT*** *COD\_ALUNO, NOME* ***FROM*** *TALUNO****;***

1. Padrões de Cabeçalho de Colunas

Alinhamento padrão:

- Esquerda: Dados tipo caracteres e data.

- Direita: Dados numéricos

Exibição padrão nome coluna: Maiúsculas (se quiser minúscula, tem que forçar).

1. Expressões aritméticas

- Crie expressões em dados tipo NUMBER e DATE utilizando operadores aritméticos.

|  |  |
| --- | --- |
| Operador | Descrição |
| + | Soma |
| - | Subtração |
| \* | Multiplicação |
| / | Divisão |

1. Definindo um valor Nulo

- Um nulo é um valor que é indispensável, não atribuído, desconhecido ou inaplicável.

- Um nulo não é o mesmo que zero ou um espaço em branco.

1. Definindo um Alias de Coluna

- Altera o cabeçalho de uma coluna.

- É útil em cálculos.

- Segue imediatamente o nome da coluna: palavra chave AS entre o nome da coluna e o alias é opcional.

- Requer aspas duplas se ele possui espaços ou caracteres especiais ou deve diferenciar maiúsculas de minúsculas.

Exemplo:

Utilizando Alias de Coluna

***SELECT*** *cod\_aluno codigo****,*** *nome* ***AS*** *aluno* ***FROM*** *taluno;*

CODIGO ALUNO

...

***SELECT*** *nome “Aluno”* ***FROM*** *taluno;*

Aluno preferencial

...

1. Operador de concatenação

- Concatena colunas ou strings de caracteres a outras colunas

- É representado através de duas barras verticais (||)

- Cria uma coluna resultante que é uma expressão caractere.

Exemplo:

***SELECT*** *cod\_aluno* ***||*** *nome* ***AS*** *“Aluno”* ***FROM*** *taluno;*

1. Strings de caracteres Literais

- Um literal é qualquer caractere, expressão ou números incluídos na lista da cláusula SELECT.

- Valores literais do tipo data e caractere devem ser colocados entre aspas simples.

- Cada string de caractere é exibida uma vez para cada linha retornada.

Exemplo:

***SELECT*** *nome* ***|| ‘*** *nasceu em* ***‘ ||*** *nascimento* ***AS “****Nascimento do Aluno****” FROM*** *taluno****;***

1. Linhas Duplicadas

E exibição padrão de consultas, lista todas as linhas, incluindo linhas duplicadas.

***SELECT*** *cidade**CIDADE* ***FROM*** *taluno****;***

CIDADE

São Paulo

Rio de Janeiro

Novo Hamburgo

Novo Hamburgo

São Paulo

São Paulo

Novo Hamburgo

1. Eliminando linhas duplicadas

Elimine linhas duplicadas utilizando a palavra chave DISTINCT na cláusula SELECT.

***SELECT DISTINCT*** *cidade**CIDADE* ***FROM*** *taluno****;***

CIDADE

São Paulo

Rio de Janeiro

Novo Hamburgo

1. Pesquisa por Ordem

Exemplo:

***SELECT*** *NOME* ***AS***

*CURSO,*

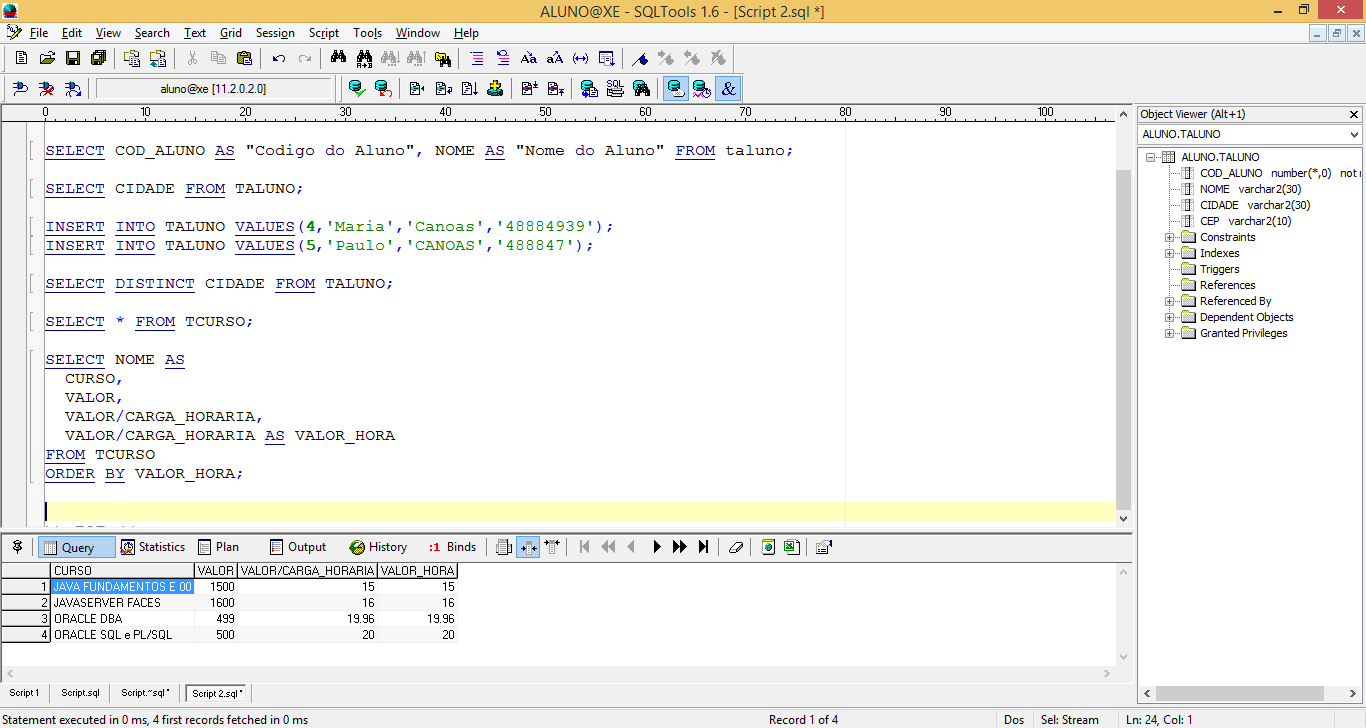
*VALOR,*

*VALOR/CARGA\_HORARIA,*

*VALOR/CARGA\_HORARIA* ***AS*** *VALOR\_HORA*

***FROM*** *TCURSO*

***ORDER BY*** *VALOR\_HORA****;***



1. Arredondado o Valor para casas decimais específicas

Duas casas decimais:

***Round(****VALOR/CARGA\_HORARIA,****2) AS*** *VALOR\_HORA*

Sem casas decimais:

***Round(****VALOR/CARGA\_HORARIA****) AS*** *VALOR\_HORA*

1. Trabalhando com valores nulos

***SELECT*** *COD\_CONTRATO****,***

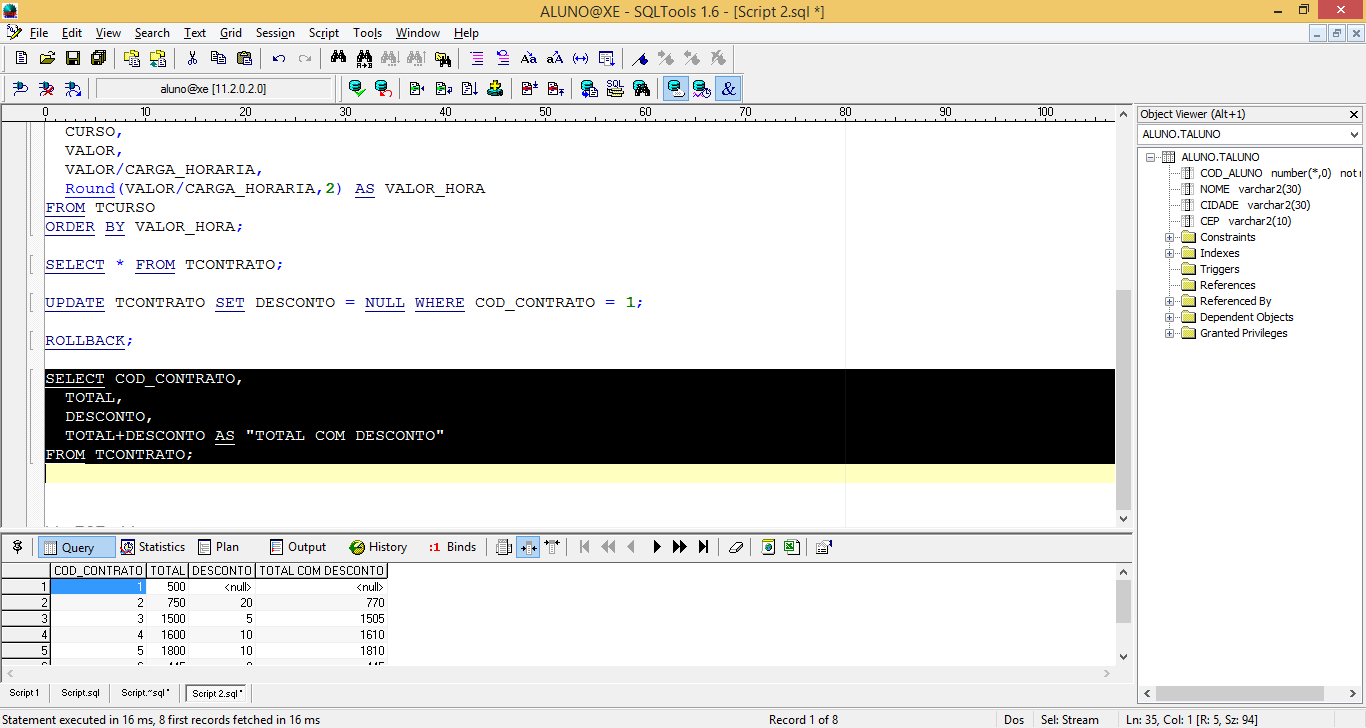
*TOTAL,*

*DESCONTO,*

*TOTAL+DESCONTO* ***AS*** *"TOTAL COM DESCONTO"*

***FROM*** *TCONTRATO****;***

O resultado abaixo mostra que a soma de valores ‘null’ retorna null.



Para tratar esse tipo de resultado, usamos o NVL no campo que desejamos que retorne o valor 0:

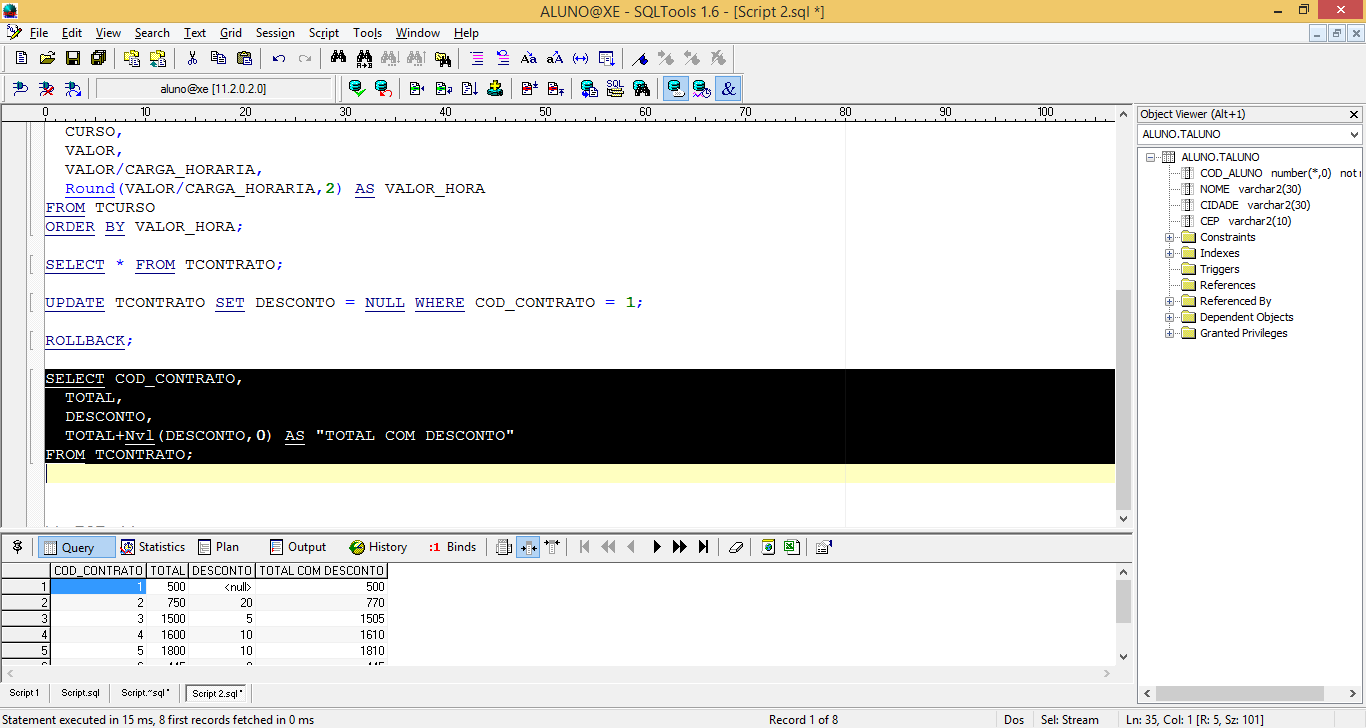
***SELECT*** *COD\_CONTRATO****,***

*TOTAL,*

*DESCONTO,*

*TOTAL****+Nvl(****DESCONTO****,0) AS*** *"TOTAL COM DESCONTO"*

***FROM*** *TCONTRATO****;***



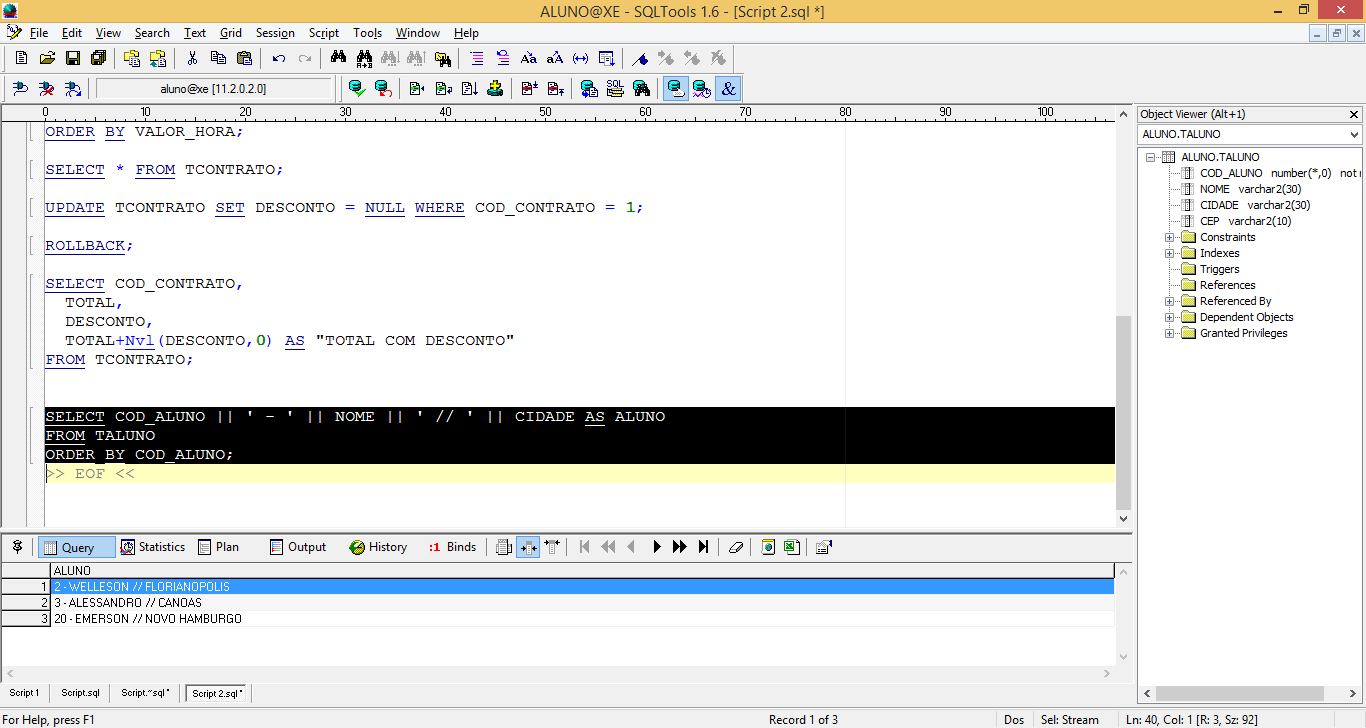
1. Exemplo de pesquisa usando Concatenação

***SELECT*** *COD\_ALUNO* ***||*** *' - '* ***||*** *NOME* ***||*** *' // '* ***||*** *CIDADE* ***AS*** *ALUNO*

***FROM*** *TALUNO*

***ORDER BY*** *COD\_ALUNO****;***

O resultado será:



**CAPÍTULO 3 – Restringindo e Ordenando dados**

- Limitar as linhas recuperadas por uma consulta  
- Ordenar as linhas recuperadas por uma consulta

3.1 Limitando as Linhas Selecionadas

***SELECT [DISTINCT]*** *{\*, column [alias], ...}*

***FROM*** *table*

***[WHERE*** *condition(s)****];***

Na sintaxe:

WHERE - restringe a consulta para as linhas que satisfazem a condição.

condition – é composta de nomes de coluna expressões, constantes, e operadores de comparação.

Exemplo:

***SELECT*** *cod\_aluno, nome*

***FROM*** *taluno*

***WHERE*** *estado = ‘RS’****;***

Strings de caractere e valores tipo data são inclusos entre aspas simples.

Valores caractere fazem distinção entre maiúsculas e minúsculas e valores tipo data são sensíveis ao formato.

3.2 Operadores de Comparação

|  |  |
| --- | --- |
| Operador | Significado |
| = | Igual a |
| > | Maior que |
| >= | Maior que ou igual a |
| < | Menor que |
| <= | Menor que ou igual a |
| <> ou != | Diferente de |
|  | |
| *...WHERE exp operador value* | |

Exemplo:

***SELEC*** *cod\_contrato, data, total, desconto + 10*

***FROM*** *tcontrato*

***WHERE*** *total <= desconto + 10****;***

3.3 Outros Operadores de Comparação

|  |  |
| --- | --- |
| Operador | Significado |
| BETWEEN  ...AND... | Entre dois valores (inclusive) |
| IN(list) | Igual a um dos valores da lista |
| LIKE | Igual a um padrão de caracteres |
| IS NULL | Possui um valor nulo |

Exemplo:

1. Selecionar todos os alunos de estados diferentes e que recebem igual ou menor o salário de 800 e ordenar em ordem decrescente:

***SELECT*** *\** ***FROM*** *TALUNO*

***WHERE*** *ESTADO* ***<>*** *'RS'*

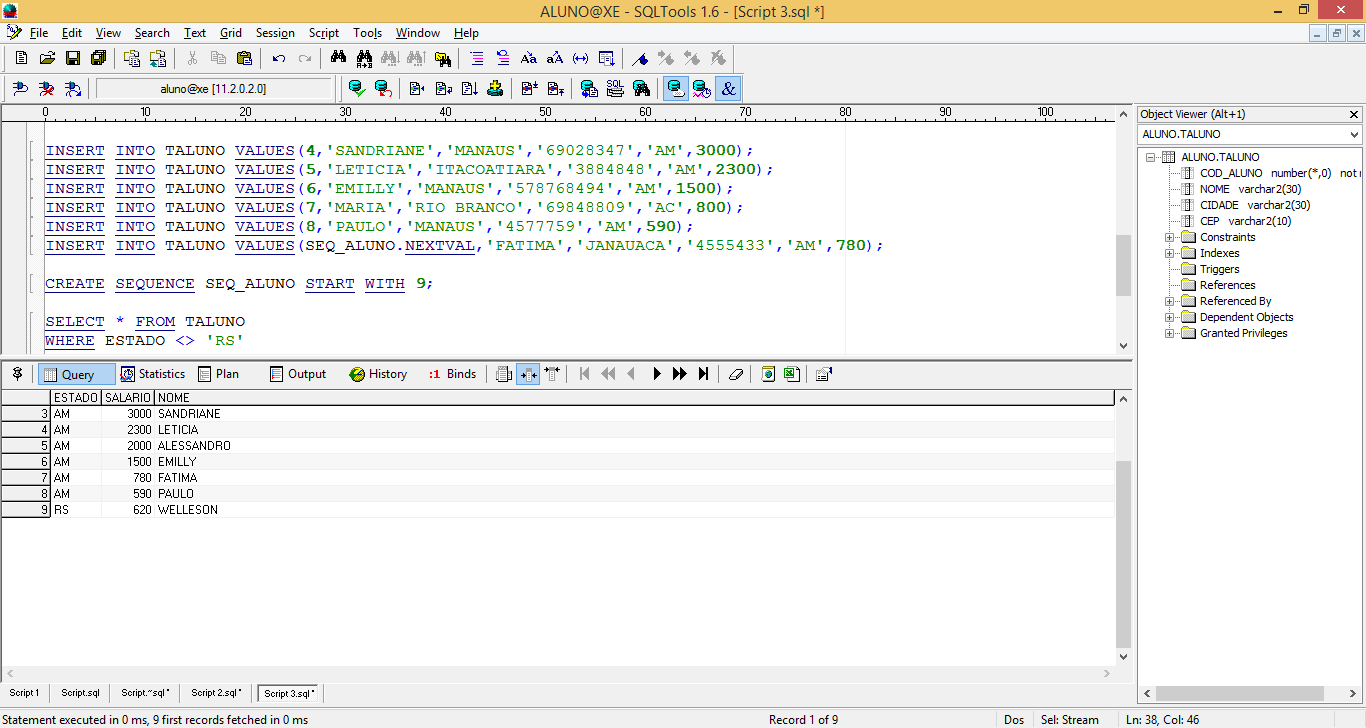
***AND*** *SALARIO* ***<=*** *800*

***ORDER BY*** *SALARIO* ***DESC;***

1. Ordem alfabética segue um critério de organização por grupo. Por exemplo, vamos listar em ordem alfabética todos os estados e em cada grupo de estado, organizar em ordem decrescente os salários:

***SELECT*** *ESTADO, SALARIO, NOME* ***FROM*** *TALUNO*

***ORDER BY*** *ESTADO, SALARIO* ***DESC;***



1. Filtrando valores com data/hora

Trunc = ignora as horas:

***SELECT*** *COD\_ALUNO, NASCIMENTO,* ***Trunc(****NASCIMENTO****),*** *NOME*

***FROM*** *TALUNO*

***WHERE Trunc(****NASCIMENTO****) =*** *'05/08/2015'****;***

1. Selecionando Datas com Between

***SELECT*** *COD\_ALUNO, NASCIMENTO,* ***Trunc(****NASCIMENTO****),*** *NOME*

***FROM*** *TALUNO*

***WHERE*** *NASCIMENTO*

***BETWEEN To\_Date(****'05/08/2015 14:33'****,'DD/MM/YYYY HH24:MI')***

***AND To\_Date(****'05/08/2015 23:33'****,'DD/MM/YYYY HH24:MI');***

1. Selecionar contratos que calcule + 1000 e listar TOTAIS que sejam menor ou igual ao desconto com + 1000;

***SELECT*** *COD\_CONTRATO, DATA, TOTAL, DESCONTO, DESCONTO* ***+ 1000 AS*** *CALCULO*

***FROM*** *TCONTRATO*

***WHERE TOTAL <=*** *DESCONTO* ***+ 1000;***

1. Listando valores NULL e diferentes de NULL e Intervalos de valores

6.1. Valores Not Null

***SELECT*** *DESCONTO*

***FROM*** *TCONTRATO*

***WHERE*** *DESCONTO* ***IS NOT NULL;***

6.2. Valores Null

***SELECT*** *DESCONTO*

***FROM*** *TCONTRATO*

***WHERE*** *DESCONTO* ***IS NULL;***

6.3. Intervalo de Valores

***SELECT*** *DESCONTO*

***FROM*** *TCONTRATO*

***WHERE*** *DESCONTO* ***BETWEEN 0 AND 10;***

1. Considerando o valor Null

***SELECT*** *DESCONTO****, Nvl(****DESCONTO****,0)***

***FROM*** *TCONTRATO*

***WHERE Nvl(****DESCONTO****,0) BETWEEN*** *0* ***AND*** *10****;***

1. Between e equivalente:

8.1 Selecionar descontos entre 0 e 10:

***SELECT*** *\** ***FROM*** *TCONTRATO*

***WHERE*** *DESCONTO* ***>=*** *0*

***AND*** *DESCONTO* ***<=*** *10****;***

Ou:

***SELECT*** *\** ***FROM*** *TCONTRATO*

***WHERE*** *DESCONTO* ***BETWEEN*** *0* ***AND*** *10****;***

8.2 Selecionar descontos entre 0 e 10 e Null

***SELECT*** *COD\_CONTRATO, TOTAL, DESCONTO,* ***Nvl(****DESCONTO****,0)***

***FROM*** *TCONTRATO*

***WHERE Nvl(****DESCONTO****,0) BETWEEN*** *0* ***AND*** *10****;***

Ou:

***SELECT*** *\** ***FROM*** *TCONTRATO*

***WHERE*** *DESCONTO* ***>=*** *0*

***AND*** *DESCONTO* ***<=*** *10*

***OR*** *DESCONTO* ***=*** *NULL****;***

* 1. Selecionar registros que contenham apenas os valores específicos

***SELECT*** *\** ***FROM*** *TCURSO*

***WHERE*** *COD\_CURSO* ***IN (****1,2,4****);***

* 1. Selecionar apenas os cursos não vendidos

***SELECT*** *\** ***FROM*** *TCURSO*

***WHERE*** *COD\_CURSO* ***NOT IN (SELECT*** *COD\_CURSO* ***FROM*** *TITEM****);***

8.5 Selecionar apenas os cursos vendidos

***SELECT*** *\** ***FROM*** *TCURSO*

***WHERE*** *COD\_CURSO* ***IN (SELECT*** *COD\_CURSO* ***FROM*** *TITEM****);***

* 1. Pesquisa por string

- Selecionar os cursos que começam com O:

***SELECT*** *\** ***FROM*** *TCURSO* ***WHERE*** *NOME* ***LIKE '****O%****';***

- Selecionar os cursos que terminam com O:

***SELECT*** *\** ***FROM*** *TCURSO* ***WHERE*** *NOME* ***LIKE '%****O****';***

- Selecionar os cursos que contenham O em qualquer parte do nome:

***SELECT*** *\** ***FROM*** *TCURSO* ***WHERE*** *NOME* ***LIKE '%****O****';***

* 1. Precedência de Operadores

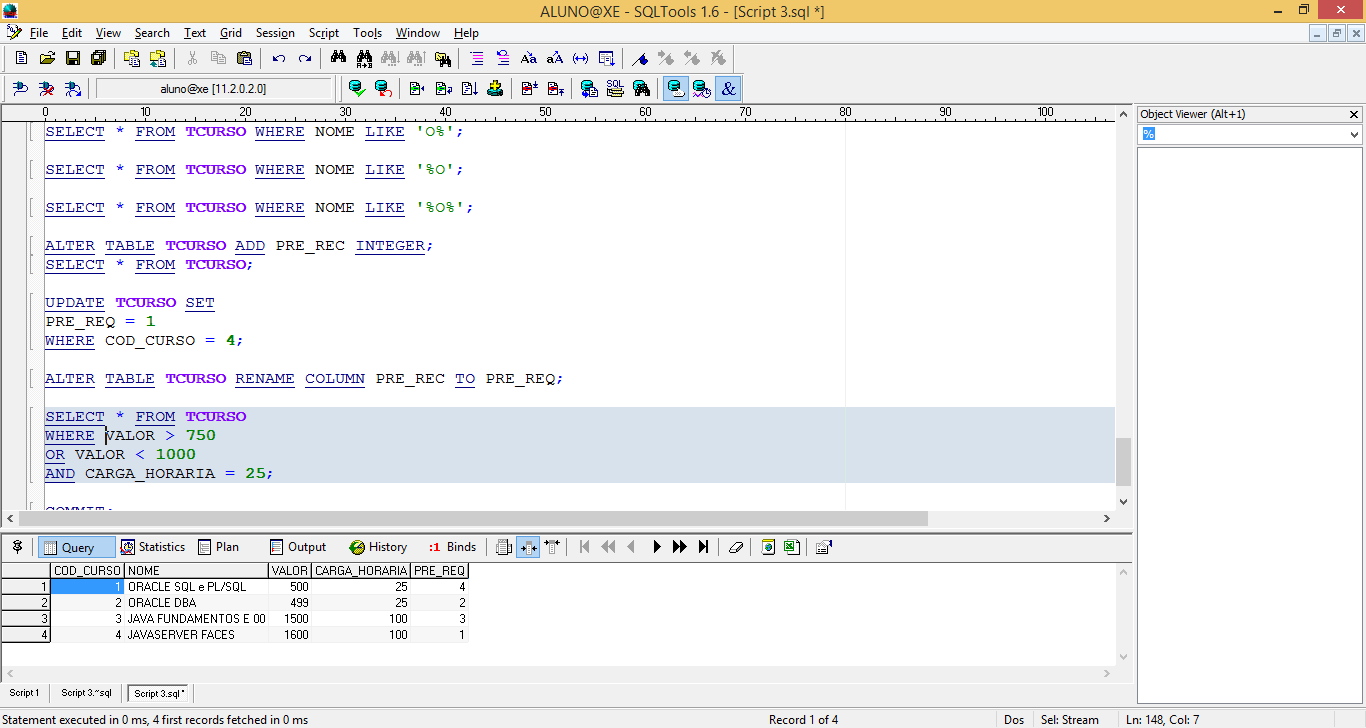
Primeiro será executado o AND, em seguida o OR, antes os parênteses:

***SELECT*** *\** ***FROM*** *TCURSO*

***WHERE*** *VALOR* ***>*** *750*

***OR*** *VALOR* ***<*** *1000*

***AND*** *CARGA\_HORARIA* ***=*** *25****;***

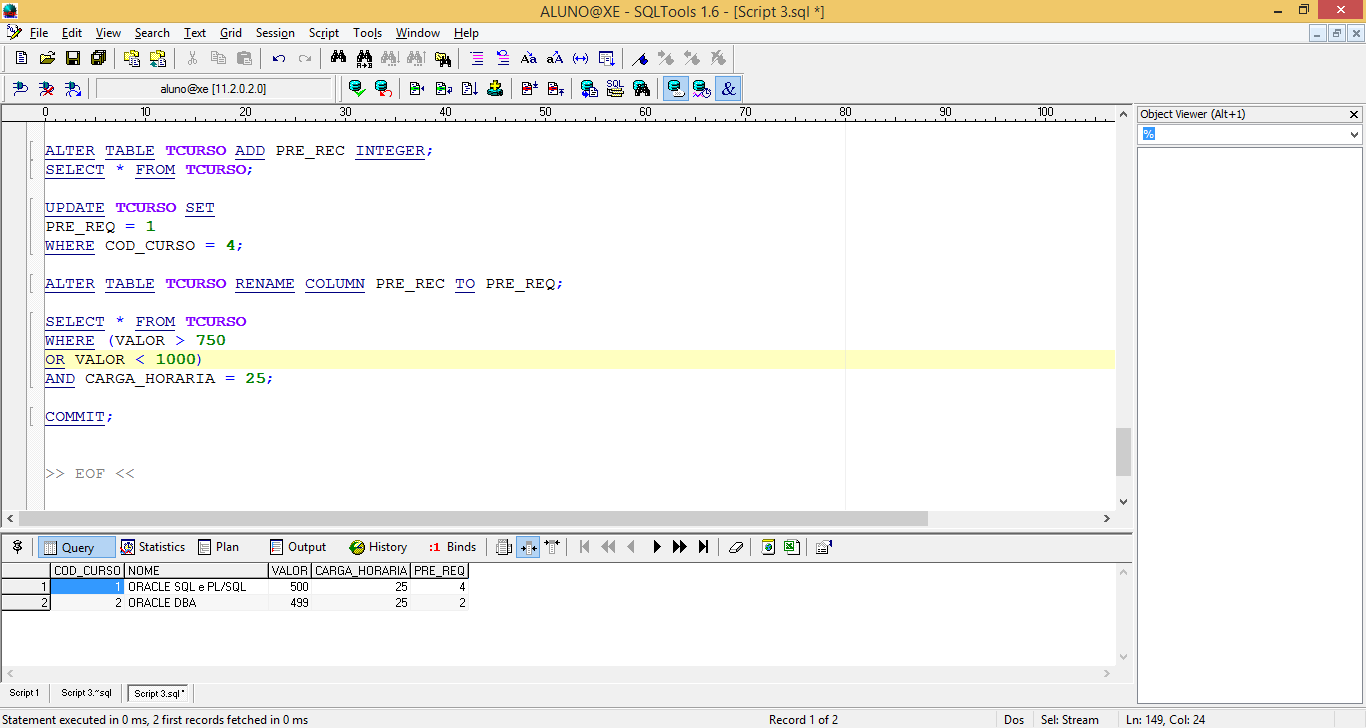


***SELECT*** *\** ***FROM*** *TCURSO*

***WHERE*** *(VALOR* ***>*** *750*

***OR*** *VALOR* ***<*** *1000)*

***AND*** *CARGA\_HORARIA* ***=*** *25****;***



CAPÍTULO 4 – Introdução as Funcões Básicas

- Descrever os vários tipos de funções disponíveis em SQL.

- Utilizar funções do tipo caractere, numéricas e de datas em comandos SELECT.

- Descrever o uso de funções de conversão.

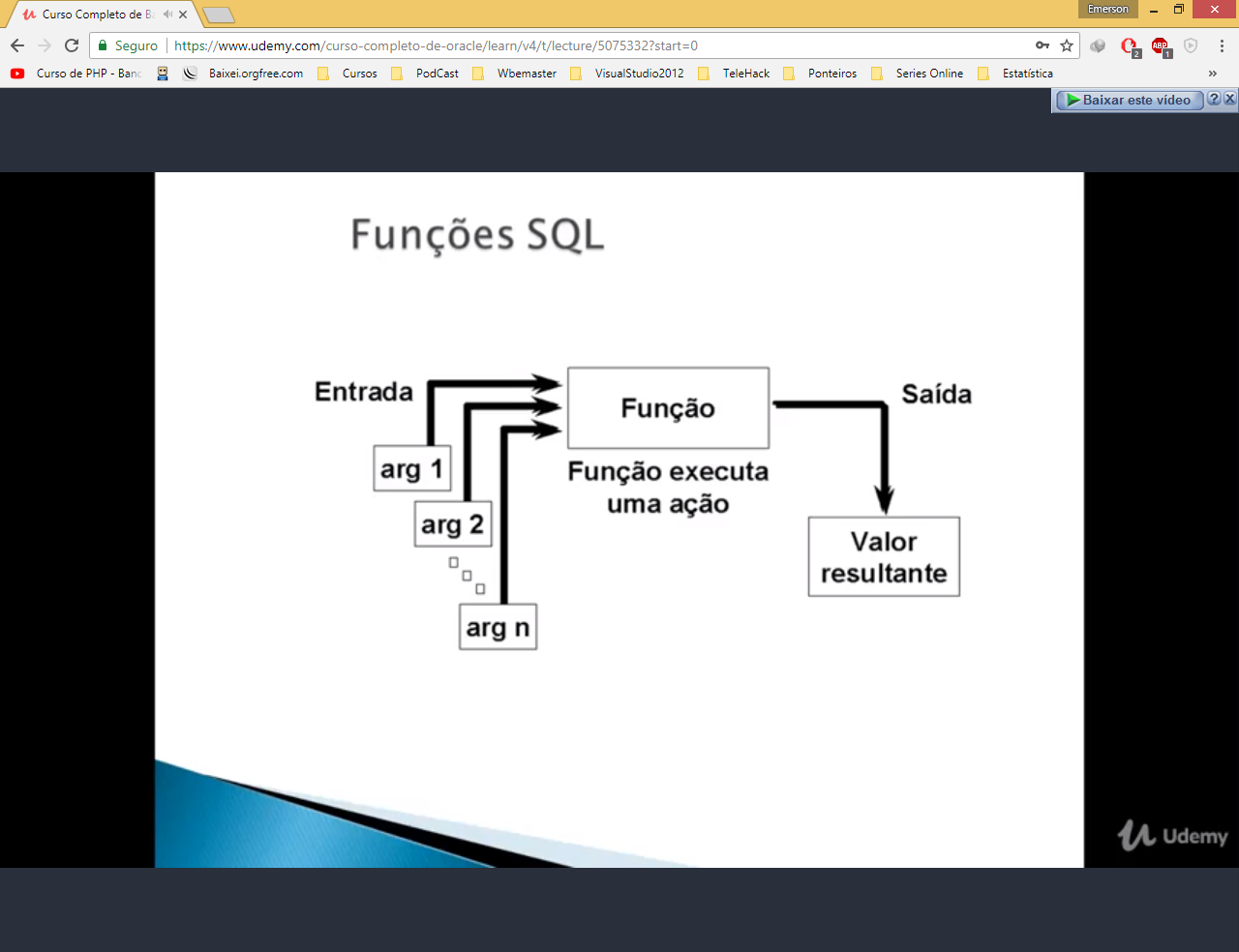


Figura - Função de entrada do tipo single row.

Tipos de Funções:

**SINGLE ROW** – Estas funções operam em linhas únicas retornando um resultado para cada linha processada.

**MULTIPLE-ROW** – Estas funções manipulam grupos de linhas para obter um resultado para cada grupo processado.

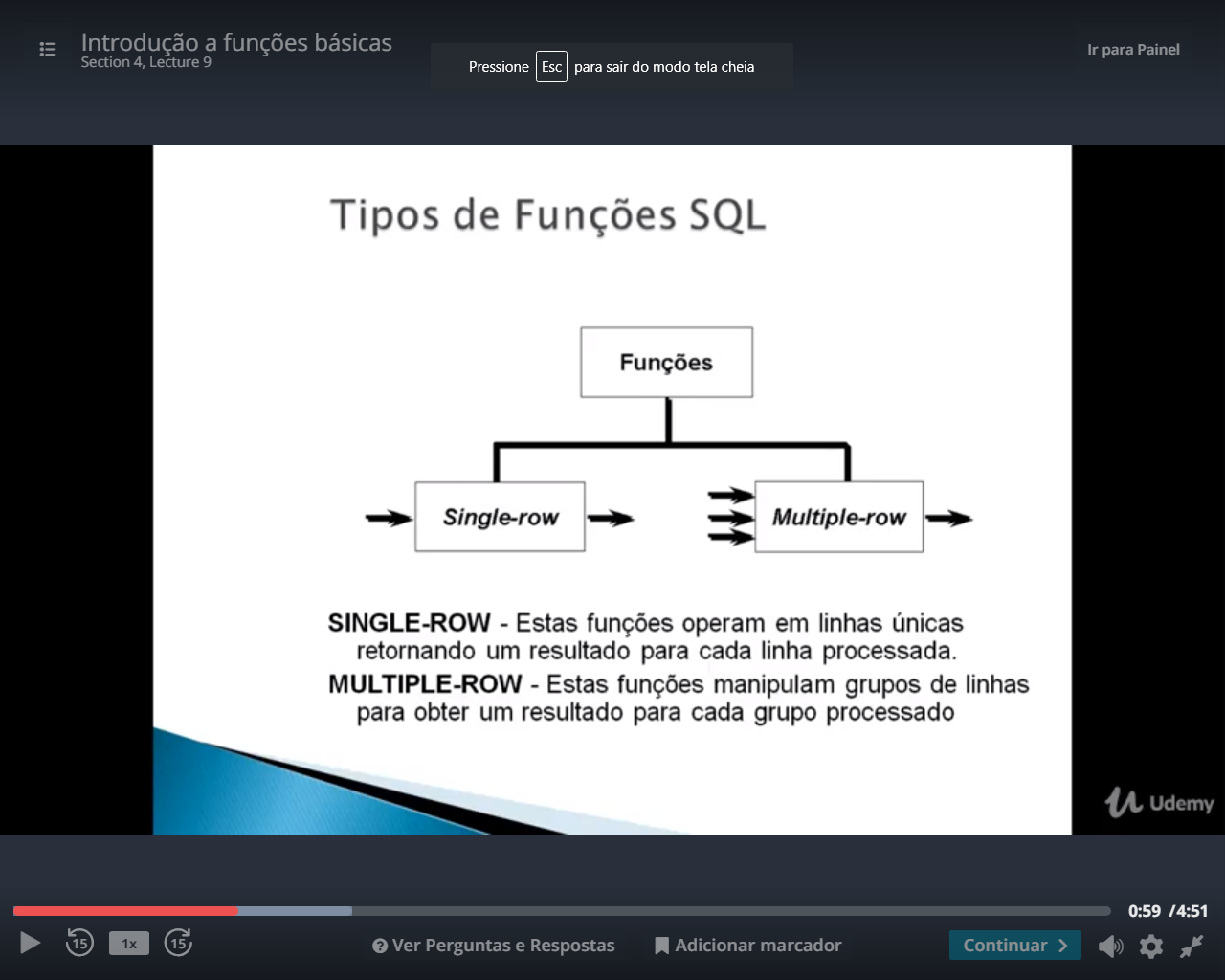


Figura - Tipos de funções.

4.1 **Funções do tipo Single-Row:**

- Manipulam itens de dados.

- Recebem argumentos e retornam um valor.

- Atuam sobre cada linha recuperada.

- Retornam um resultado por linha.

- Podem modificar o tipo de um lado.

- Podem ser aninhadas.

function\_name (column | expression, [arg1, arg2, ...])

4.1 **Funções de Caracteres**

|  |  |
| --- | --- |
| **Função** | **Propósito** |
| CONCAT(column1|expression1,column2|expression2) | Concatena a primeira string de caracateres com a segunda. Equivalente ao operador de concatenação (||). |

|  |  |
| --- | --- |
| INITCAP (column|expression) | Converte strings de caracteres deixando a primeira letra de cada palavra em maiúscula e as demais em minúsculas. |
| INSTR (column|expression,m) | Retorna a posição numérica do caracter dentro da string. |
| LENGTH (column|expression) | Retorna o número de caracteres da string. |
| LOWER (column|expression) | Converte strings de caracteres para minúscula. |
| LPAD (column|expression, n, ‘string’) | Retorna uma string com tamanho total de *n* alinhada à direita. |
| REPLACE (x,search\_string,replace\_string) | Procura a string *search\_string* em *x* e substitui por *replace\_string*. |
| RPAD (column|expression, n, ‘string’ | Retorna uma string com tamanho total de *n* alinhada à esquerda. |
| SUBTR(column|expression, m[,n]) | Retorna os caracteres especificados a partir da string de caracteres, começando na posição *m*, com tamanho de *n* caracteres. Se *m* for negativo, a contagem inicia a partir do final da string. Se *n* for omitido, são retornados todos os caracteres até o final da string. |
| UPPER(column|expression) | Converte strings de caracteres para maiúsculas. |

4.2 Funções de conversão entre Maiúsculas/Minúsculas (Exemplos)

|  |  |
| --- | --- |
| Função | Resultado |
| LOWER(‘Introdução ORACLE 11g’) | introdução oracle 11g |
| UPPER(‘Introdução ORACLE 11g’) | INTRODUÇÃO ORACLE 11G |
| INITCAP(‘Introdução ORACLE 11g’) | Introdução Oracle 11g |

SELECT ‘ O Cliente ’ || INITCAP (nome) || ‘ mora em ’ || UPPER (cidade) || ‘ - ’ || LOWER(estado) AS “Informações” FROM taluno;