Funções Básicas

Curso de Introdução a Oracle 11g: SQL/Avançado

Prof.: Marlon Mendes Minussi
marlonminussi@gmail.br

Funções de Conversão

Funções de Conversão

- Existem dois tipos distintos de funções:
- Funções tipo Single-Row: Estas funções operam em linhas únicas retornando um resultado para cada linha processada. Existem diferentes tipos de funções single-row. Este tópico explica os tipos listados abaixo:
- Caractere, Numérica, Data e Conversão.
- Funções do tipo Multiple-Row: Estas funções manipulam grupos de linhas para obter um resultado para cada grupo processado e foram vistas no semestre passado: Min, Max, Group By.

Funções do Tipo Single-Row

- Funções do Tipo single-row são utilizadas para manipular itens de dados. Elas recebem um ou mais argumentos e retornam um único valor para cada linha recuperada pela consulta.
- Sintaxe:
  - Function_name (column|expression,[arg1, arg2, ...])
    - Vocês pode utilizá-las nas cláusulas SELECT, WHERE e ORDER BY. Você pode também aninhar funções.
      - function_name é o nome da função.
      - column é qualquer coluna nomeada do banco de dados.
      - expression é qualquer string de caractere ou expressão calculada.
      - arg1, arg2 são quaisquer argumentos a serem utilizados pela função.

Funções de Conversão

Função	Resultado
LOWER (‘Curso SQL’)	curso sql
UPPER (‘Curso SQL’)	CURSO SQL
INITCAP(‘Curso SQL’)	Curso Sql

Funções de Conversão Maiúsculas/Minúsculas

- LOWER, UPPER e UNITCAP são as três funções de conversão entre maiúsculas e minúsculas.
  - LOWER: converte todos os caracteres de uma string para minúsculas
  - UPPER: converte todos os caracteres de uma string para maiúsculas
  - INITCAP: converte a primeira letra de cada palavra para maiúsculas e as demais para minúsculas.

    - Ex: SELECT 'O paciente '||INITCAP(nome)||' mora na '||LOWER(end)||' - '||UPPER(cidade) as "Informações sobre Paciente" FROM paciente;

Funções de Conversão Maiúsculas/Minúsculas

- Ex:
  - SELECT cod_pac, nome
    FROM paciente
    WHERE nome = ‘RENATO GAUCHO’;

  - No exemplo acima não terá retorno de nenhum registro pois o paciente foi inserido na tabela como ‘Renato Gaucho’

Funções de Conversão Maiúsculas/Minúsculas

- Ex:
  - SELECT cod_pac, nome
    FROM paciente
    WHERE LOWER(nome) = ‘renato gaucho’;
  - A cláusula where do segundo comando SQL especifica que o nome do paciente na tabela PACIENTE deve ser convertido para minúsculas e então comparado com ‘renato gaucho’.
  - Considerando que ambos os nomes estão em minúscula agora, uma correspondência é encontrada e uma linha é selecionada.

Funções de Conversão Maiúsculas/Minúsculas

- Ex:
  - SELECT cod_pac, nome

    FROM paciente

    WHERE nome= UPPER(nome);
  - A cláusula where do comando SQL especifica que o nome do paciente na tabela PACINTE seleciona todos os pacientes com o nome cadastrado em letras maiúsculas.
  - Tarefa 1: Selecione todos os clientes cadastrados com letra minúsculas.
  - Tarefa 2: Selecione todos os clientes cadastrados com a 1ª letra de cada nome maiúscula.

Funções de Manipulação de Caracteres

Função	Resultado
CONCAT (‘Str’,’String’)	StrString
SUBSTR (‘String’,1,3)	Str
LENGTH(‘String’)	6
INSTR(‘String’,’r’)	3
LPAD(renda,10,’*’)	******5000

Funções de Manipulação de Caracteres

- CONCAT, SUBSTR, LENGHT, INSTR E LPAD são as cinco funções de manipulação de caracteres apresentados neste capítulo.
  - CONCAT: concatena strings de caracteres, sendo limitado ao uso de apenas dois parâmetros.
  - SUBSTR: extrai uma string de tamanho determinado.
  - LENGHT: exibe o tamanho de uma string como um valor numérico.
  - INSTR: encontra a posição numérica de um caracter na string.
  - LPAD: retorna uma string de caracteres do tamanho especifico alinhado à direita, existe a RPAD que alinha à esquerda.

Funções de Manipulação de Caracteres

Ex:

```
SELECT nome, CONCAT('COD-',cod_pac),
LENGTH(nome), INSTR(nome, 'A')
FROM paciente
WHERE SUBSTR(end,1,5) = ‘Porto’;
```

- O exemplo acima exibe o nome do paciente, a string ‘COD-’ concatenado com o código do paciente, o tamanho do nome do paciente e a posição numérica da letra A no nome, para todos os pacientes que possuem a string ‘Porto’ nas 5 primeiras posições do campo END da tabela PACIENTE.

Funções de Manipulação de Caracteres

- Ex.:
- Modificando o comando SQL acima para exibir os dados para os paciente cujo nome termine com a letra ‘o’.
  - SELECT nome, CONCAT('COD-',cod_pac), LENGTH(nome), INSTR(nome, 'A')
    FROM paciente
    WHERE LOWER(SUBSTR(nome,-1,1))= 'o’;
  - OBS: Pode usar o comando like, mas é redundante. Ex like ‘%o’;
- Ex.:
  - SELECT cod_atendimento, LPAD(valor,10,'*')
    FROM atendimento;

Funções Numéricas

Função	Resultado
ROUND	Arredonda o valor para a decimal esperada
ROUND (45.926,2)	45.93
TRUNC	Trunca o valor para a decimal especificada
TRUNC(45.926,2)	45.92
MOD	Retorna o resto da divisão
MOD(1600,300)	100

Funções Numéricas

- Funções numéricas recebem parâmetros numéricos e retornam valores numéricos. Esta seção descreve algumas das funções numéricas.

Função	Resultado
ROUND (column|expression, n)	Arredonda a coluna, expressão ou valor para n casas decimais. Se n for omitido, será considerado 0, ou seja, sem casas decimais. Se n for negativo, os números à esquerda do ponto decimal serão arredondados.
TRUNC (column|espression, n)	Trunca a coluna, expressão ou valor para n casas decimal . Se n for omitido, será considerado 0. Se n for negativo, os números à esquerda do ponto decimal serão truncados para zero.
MOD (m,n)	Retorna o resto da divisão de m por n.

Utilizando a função ROUND

- Ex.:

```
SELECT ROUND(45.923,2), ROUND(45.923,0), ROUND(45.923,-1)
FROM DUAL;
```

- Resposta:

```
ROUND(45.923,2) ROUND(45.923,0) ROUND(45.923,-1)
--------------- --------------- ---------------
          45,92              46              50
```

- A função ROUND arredonda uma coluna, expressão ou valor para n casas decimais. Se o segundo argumento é 0 ou não for informado, o valor é arredondado para zero casas decimais. Se o segundo argumento é 2, o valor é arredondado para duas casas decimais. Reciprocamente, se o segundo argumento é -1, o valor é arredondado as casas decimais a esquerda.

Utilizando a função TRUNC

- Ex.:

```
SELECT TRUNC(45.923,2), TRUNC (45.923,0), TRUNC(45.923,-1)
FROM DUAL;
```

- Resposta:

```
TRUNC (45.923,2) TRUNC(45.923,0) TRUNC(45.923,-1)
---------------- ---------------- ----------------
           45,92               45               40
```

- A função TRUNC trunca a coluna, expressão ou valor para n casas decimais.

Utilizando a função TRUNC

- A função TRUNC opera com argumentos semelhantes aos da função ROUND. Se o segundo argumento é 0 ou não for informado, o valor é truncado para zero casas decimais. Se o segundo argumento é 2, o valor é truncado para duas casas decimais. Reciprocamente, se o segundo argumento é -2, o valor é truncado duas casas decimais para esquerda.

- Com a função ROUND, a função TRUNC também pode ser utilizada com funções de data.

Utilizando a Função MOD

- Ex.:

  SELECT cod_atendimento, cod_pac, valor, MOD(valor,2)

  FROM atendimento;

- Resposta:

COD_ATENDIMENTO	COD_PAC	VALOR	MOD(VALOR,2)
1	1	1000	0
2	2	100000	0
3	4	50000	0
4	1	11550,78	10,78

- A função MOD encontra o resto da divisão do valor 1 pelo valor 2. O exemplo acima calcula o resto da divisão do total pelo desconto dos contratos que possuam desconto válido.

Trabalhando com Datas

- Formato de Data Oracle
- O Oracle armazena datas em um formato numérico interno, representando o século, ano, mês, dia, horas, minutos e segundo.
- O formato padrão para exibição e entrada de datas é DD-MON-YY.
- Datas válidas do Oracle estão entre 1 de janeiro de 4712 A.C. e 31 de dezembro de 9999 D.C.

Nota:

- SYSDATE
- SYSDATE é uma função de data que retorna a data e hora atual.
- Você pode utilizar SYSDATE da maneira que você utiliza qualquer outro nome de coluna.
- Por exemplo, você pode exibir a data atual selecionando SYSDATE a partir de uma tabela.
- É habitual selecionar SYSDATE a partir de uma tabela dummy chamada DUAL.

Nota:

- DUAL
- A tabela DUAL pertence ao usuário SYS e pode ser acessada por todos os usuários do banco de dados.
- A tabela DUAL contém apenas uma coluna e uma linha.
- Esta tabela é útil quando se deseja consultar valores, como por exemplo, o valor de uma constante, pseudocoluna ou expressão que não é originada de uma tabela com dados do usuário.

- Exemplo: Mostre a data corrente utilizando a tabela DUAL.

```
SELECT SYSDATE
FROM DUAL;
```

Cálculos com Datas

- Uma vez que o banco de dados armazena as como números, você pode executar cálculos utilizando os operadores aritméticos como a adição e subtração.
- Você pode adicionar e subtrai constantes numéricas e datas.

Cálculos com datas

Operação	Resultado	Descrição
Data + número	Data	Adiciona um número de dias para a data
Data - número	Data	Subtrai um numero de dias a partir de uma data
Data - data	Numero de dias	Subtrai uma data a partir de outra
Data +número/24	Data	Adiciona um numero de horas para uma data

Cálculos com datas

- Ex.:
- ALTER TABLE paciente add dt_nasc date;

- UPDATE paciente SET dt_nasc= ‘10/03/1964’ WHERE cod_pac=1;

- SELECT nome, (SYSDATE-dt_nasc)/7 SEMANAS FROM paciente;

- O exemplo acima exibe o nome e o número de semanas de vida dos pacientes. Ele subtrai a data atual(SYSDATE) a partir da data na qual o paciente nasceu e divide o resultado por 7 para calcular numero de semanas.

Funções de Data

Função	Descrição
MONTHS_BETWEEN (date1,date2)	Número de meses entre duas datas
ADD_MONTHS (date,n)	Adiciona meses do calendário para uma data
NEXT_DAY (date, ‘char’)	Próximo dia da data especificada
LAST_DAY (date)	Ultimo dia do mês
ROUND (date,[‘fmt’])	Data arredondada
TRUNC (date,[‘fmt’])	Data truncada

Funções de Data

Função	Resultado
MONTHS_BETWEEN (’01-SEP-96,’11-JAN-95’)	19.6774194
ADD_MONTHS (‘11-JAN-94’,6)	‘11-JAN-94’
NEXT_DAY (‘01-SEP-95’, ‘FRIDAY’)	’08-SEP-95’
LAST_DAY (‘01-SEP-95’)	’30-SEP-95’

Funções de Data

- Exemplo:
- Para todos os pacientes que nasceram a menos de 500 meses, selecione, o código do cliente, a data de nascimento, o número de meses vividos, a data precisa após 6 meses do nascimento deste paciente, a primeira sexta-feira após a data de nascimento e o último dia do mês do paciente.

```
SELECT cod_pac, dt_nasc,
            MONTHS_BETWEEN(SYSDATE, dt_nasc),
            ADD_MONTHS(dt_nasc, 6),
            NEXT_DAY(dt_nasc, 'QUINTA'),
            LAST_DAY(dt_nasc)
FROM paciente
WHERE MONTHS_BETWEEN(SYSDATE, dt_nasc) < 500;
```

Funções de Data

- Exemplo:
- Compare as datas de nascimento para todos os paciente que nasceram em 1973. Mostre o código do paciente, a data de nascimento e o mês com as funções ROUND e TRUNC.

```
SELECT cod_pac, dt_nasc,
            ROUND(dt_nasc, 'MONTH'),
            TRUNC(dt_nasc, 'MONTH')
FROM paciente
WHERE dt_nasc LIKE '%73';
```

Funções de Conversão

Função	Propósito
TO_CHAR (number|date,[‘fmt’])	Converte um valor numérico ou data para uma string de caracteres do tipo VARCAR2 no formato fmt.
TO_NUMBER (char)	Converte uma string de caracteres contendo apenas dígitos para um numero.
TO_DATE (char,[‘fmt’])	Converte uma string de caracteres representando uma data para um valor de data de acordo com o fmt especificado (se o fmt for omitido, o padrão é DD-MON-YY).
TO_TIMESTAMP(char,[‘fmt’])	Converte uma string de caracteres representando uma data para um valor de data de maior precisão de acordo com o fmt especificado.

TO_CHAR(Datas e Horas)

- Converte a data ou registro de data/hora (timestamp) em uma string de caractere VARCHER2.
- Se o formato for especificado, ele é utilizado para controlar como resultado é estruturado.

- Ex:

  Select to_char(SYSDATE, 'DD-MON-YY HH24:MI:SS') "Data e Hora" from dual;

Data e Hora

19-MAI-09 09:28:52

Exemplos:

```
select    (TO_CHAR(DATA, 'DD-MON-YYYY HH24:MI:SS') )
      from FRETE
/

select    (TO_CHAR(DATA, 'DD-MM-YYYY HH12:MI:SS') )
      from FRETE
/

select    (TO_CHAR(DATA, 'Day / Mon / YYYY HH24:MI:SS') )
      from FRETE
/
select    (TO_CHAR(DATA, 'Day / Month / YYYY HH24:MI:SS') )
      from FRETE
/
```

TO_CHAR(Número)

- Converte o argumento NUMBER de num em um VARCHAR2.
- Se especificado, formato governa a conversão.
- Se o formato não for especificado, a string resultante terá exatamente tantos caracteres quantos necessários para conter os dígitos significativos de num.

Exemplos:

- select TO_CHAR(123456, '99G99G99') "Resultado" from dual;
- select replace(TO_CHAR(123456, '99G99G99'),'.',',') "Resultado" from dual
- select TO_CHAR(123456.34, '$999g999g990d00') "Resultado" from dual
- select TO_CHAR(123456, '$9999.99') "Resultado" from dual;
- select TO_CHAR(123456.34, 'fm999g999g990d00') "Resultado" from dual
- A letra G coloca uma vírgula na saída da consulta.
- FM serve para tirar casas em branco.

TO_DATE

- Converte a string CHAR ou VARCHAR2 em uma DATE.
- Ex: select TO_DATE('10 agosto, 2010', 'DD Month, YYYY') "Resultado" from dual;
- Ex:

```
DECLARE
V_CurrentDate DATE;
BEGIN
V_CurrentDate:= TO_DATE('10 Janeiro, 1982', 'DD Month,
   YYYY');
DBMS_OUTPUT.PUT_LINE('V_CurrentDate');
DBMS_OUTPUT.PUT_LINE(V_CurrentDate);
END;
```

TO_NUMBER

- Converte a string CHAR ou VARCHAR2 em um valor de NUMBER. Ex:
  - select TO_NUMBER(’12.45.98’, ‘9999999’) “Resultado” from dual;
- SET SERVEROUTPUT ON (para visualizar o resultado execute o comando).
- Ex:

```
DECLARE
V_Numero NUMBER;
BEGIN
V_Numero := TO_NUMBER ('12.45.12','9999999');
DBMS_OUTPUT.PUT_LINE('V_Numero');
DBMS_OUTPUT.PUT_LINE(V_Numero);
END;
```

TO_TIMESTAMP

- Converte a string CHAR ou VARCHAR2 em uma string TIMESTAMP (é uma data de maior precisão).
- Ex:
  - select TO_TIMESTAMP('10 Janeiro, 1982', 'DD Month, YYYY') "Resultado" from dual;
- Ex:

```
DECLARE
V_CurrentDate TIMESTAMP;
BEGIN
V_CurrentDate:= TO_TIMESTAMP('10 Janeiro, 1982', 'DD Month, YYYY');
DBMS_OUTPUT.PUT_LINE('V_CurrentDate');
DBMS_OUTPUT.PUT_LINE(V_CurrentDate);
END;
```

Função NVL, NVL 2

- Utilizado para converter um valor nulo para outro valor do mesmo tipo.

Exemplos:

NVL(desconto,0])

NVL (data_nasc,'01-JAN-71')

NVL (nome_cliente,'Nome não cadastrado')

NVL2(cidade,estado,'Localidade não definida')

Função NVL, NVL 2

- Sintaxe:

  NVL(expr1, expre2)

  Onde:

  expr1 é o valor ou expressão de origem que pode conter nulo.

  expr2 é o valor de destino utilizado quando o valor de origem é nulo.

  – Você pode utilizar a função NVL para converter qualquer tipo de dado, porém, o valor de retorno deve ser sempre do mesmo tipo de dado do parâmetro expre1.

Função NVL, NVL 2

- Para converter um valor nulo para outro valor do mesmo tipo, utilize a função NVL2.
- Sintaxe:

  NVL(expr1, expr2, expr3)

  Onde:

  expr1 é o valor ou expressão de origem que pode conter nulo.

  expr2 é o valor caso expr1 não seja nula.

  expr3 é o valor de destino utilizado quando o valor de origem é nulo.

Função NVL, NVL 2

- Para calcular o desconto de todos os atendimentos, você precisa do valor no campo desconto. Mas e se este valor for nulo? Para mostrar os valores de desconto iguais a 0 nos contratos com descontos nulos utilizamos a função NVL, mas caso este atendimento não tenha desconto utilizamos a função NVL2 para assumir que 10% do valor do atendimento é o desconto aplicado.
- Exemplo:
  - SELECT cod_atendimento, valor, desconto, NVL(desconto,0), NVL2(desconto,desconto*.05,valor*.1) FROM atendimento;
  - SELECT cod_atendimento, diagnostico, NVL(diagnostico,'Sem Diagnostico') FROM atendimento;
  - SELECT cod_pac, cidade, estado, NVL(cidade,'Não Informada'), NVL2(cidade, estado, 'Cidade não Informada') FROM paciente;

Função DECODE

- Sintaxe:

  DECODE(col/expression, search1, result1
  [, search 2, result2,...,]
  [,default])

- A função de DECODE decodifica uma expressão de modo semelhante a lógica IF-THEN-ELSE utilizada em diversas linguagens. A função DECODE decodifica a expressão após compara-lá com cada valor de pesquisa. Se a expressão for igual ao valor de pesquisa, o resultado correspondente é retornado.
- Se o valor default for omitido, um valor nulo retorna quando o parâmetro col/expression for igual a nenhum dos valores de pesquisa.

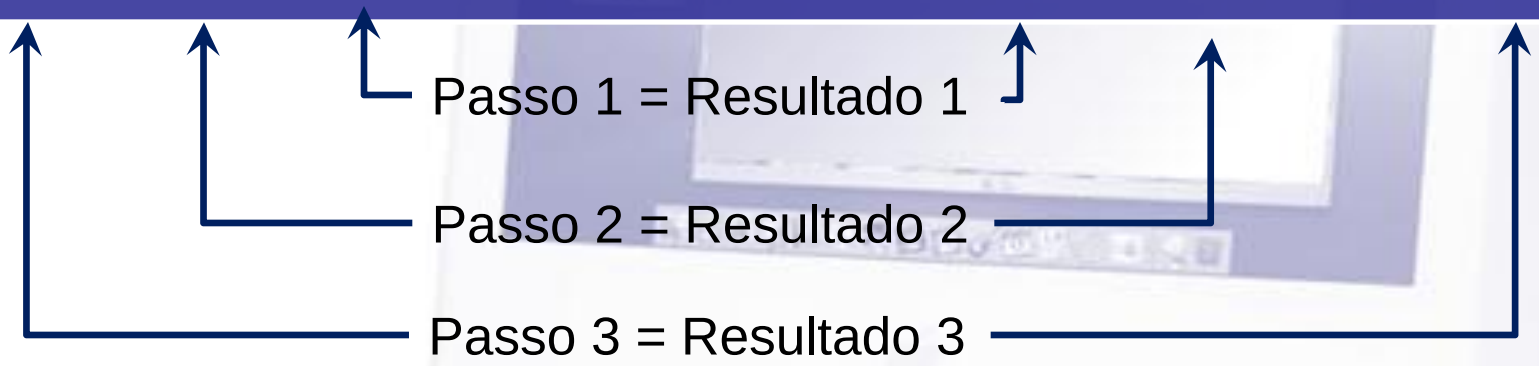
Função DECODE

- Exemplo:
  - SELECT cidade, estado, DECODE(estado,'RS', 'Gaúcho',‘RJ',‘Carioca','Brasileiro’) ESTADOS FROM paciente;

- No comando SQL acima, o valor do ESTADO é decodificado. Se ESTADO for RS, o retorno é ‘Gaúcho’; ESTADO for SP, o retorno é ‘Paulista’. Para todos os outros estados, o retorno é ‘Brasileiro.’

Aninhando Funções

F3 (F2 (F1 (col, arg1), arg2), arg3)



- Funções básicas (single-row) podem ser aninhadas em qualquer nível. Funções aninhadas são avaliadas do nível mais interno para o nível mais externo. Abaixo seguem alguns exemplos para mostrar a flexibilidades das Funções.

Aninhando Funções

- Exemplo:
  - Mostre a data da próxima sexta-feira seis meses após a data de nascimento do paciente. A data resultante deve ser formatada como por exemplo, ‘Friday, March 12th, 1982’ .Ordene o resultado pela data de nascimento.

  - SELECT dt_nasc NASCIMENTO,

    TO_CHAR(NEXT_DAY(ADD_MONTHS(dt_nasc,6),'Sexta'),'fmDay, Month ddth, YYYY') "6 Meses de Idade"

    FROM paciente order by dt_nasc;