

FIAP GRADUAÇÃO

BUILDING RELATIONAL DATABASE

Prof. Diogo Alves
profdiogo.alves@fiap.com.br

SELECT: CONCEITO DE FUNÇÃO DE LINHA E FUNÇÃO DE GRUPO DE LINHA

Agenda

- ✓ Objetivo
- ✓ Executando funções de linha e de grupo no projeto Sakspildap
- ✓ Exercícios

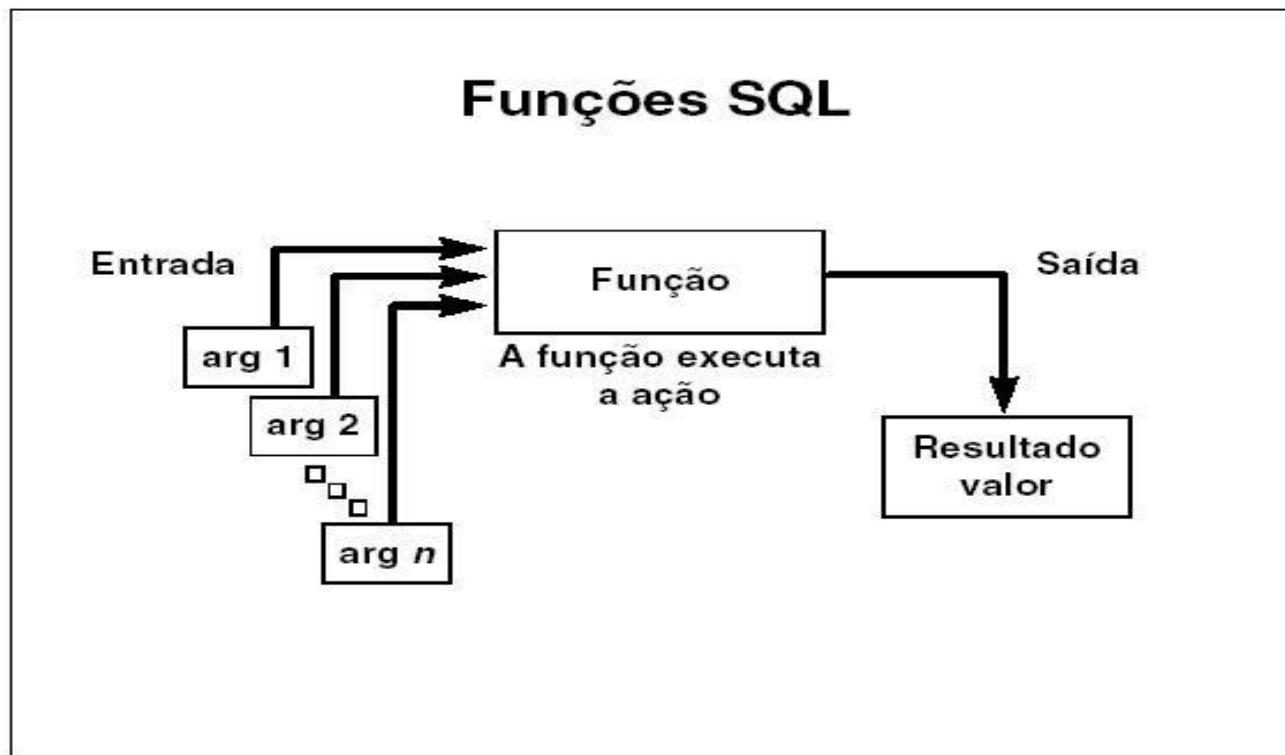
Objetivos

- ❑ Aplicar os conceitos da linguagem SQL durante a implementação do banco de dados, utilizando funções de linha e funções de grupo de linhas.

Conteúdo Programático referente a esta aula

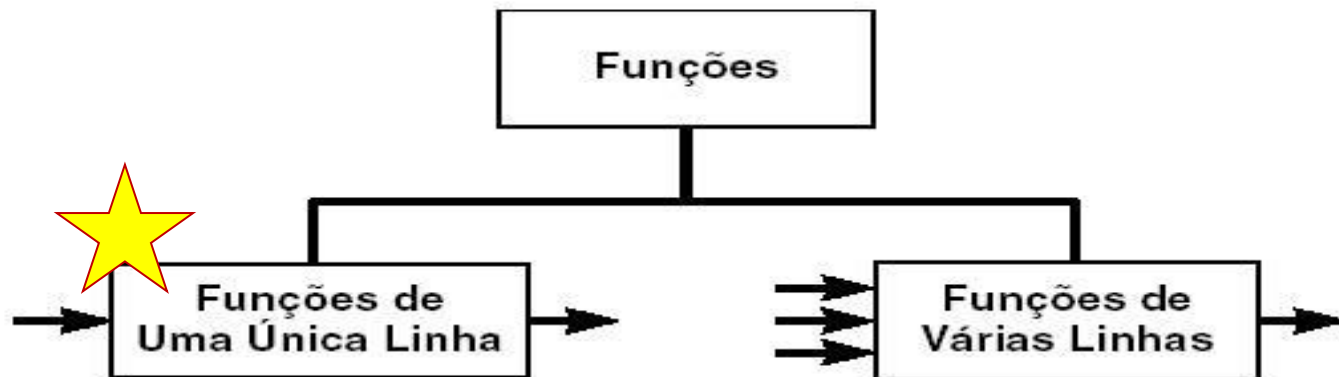
- ☐ Linguagem para visualizar os dados
 - ☐ FUNÇÕES SIMPLES
 - ☐ FUNÇÕES DE GRUPO DE LINHAS
 - ☐ Exercícios durante a apresentação da aula

- As funções são utilizadas para manipular valores de dados
- Operam sob um conjunto de linhas, retornando o resultado baseado neste conjunto
- Têm como objetivo disponibilizar recursos aos usuários, facilitando o manuseio das informações



- Executar cálculos usando dados
- Modificar itens individuais
- Manipular saída para grupos de linhas
- Formatar datas e números para exibição
- Converter tipos de dados de coluna

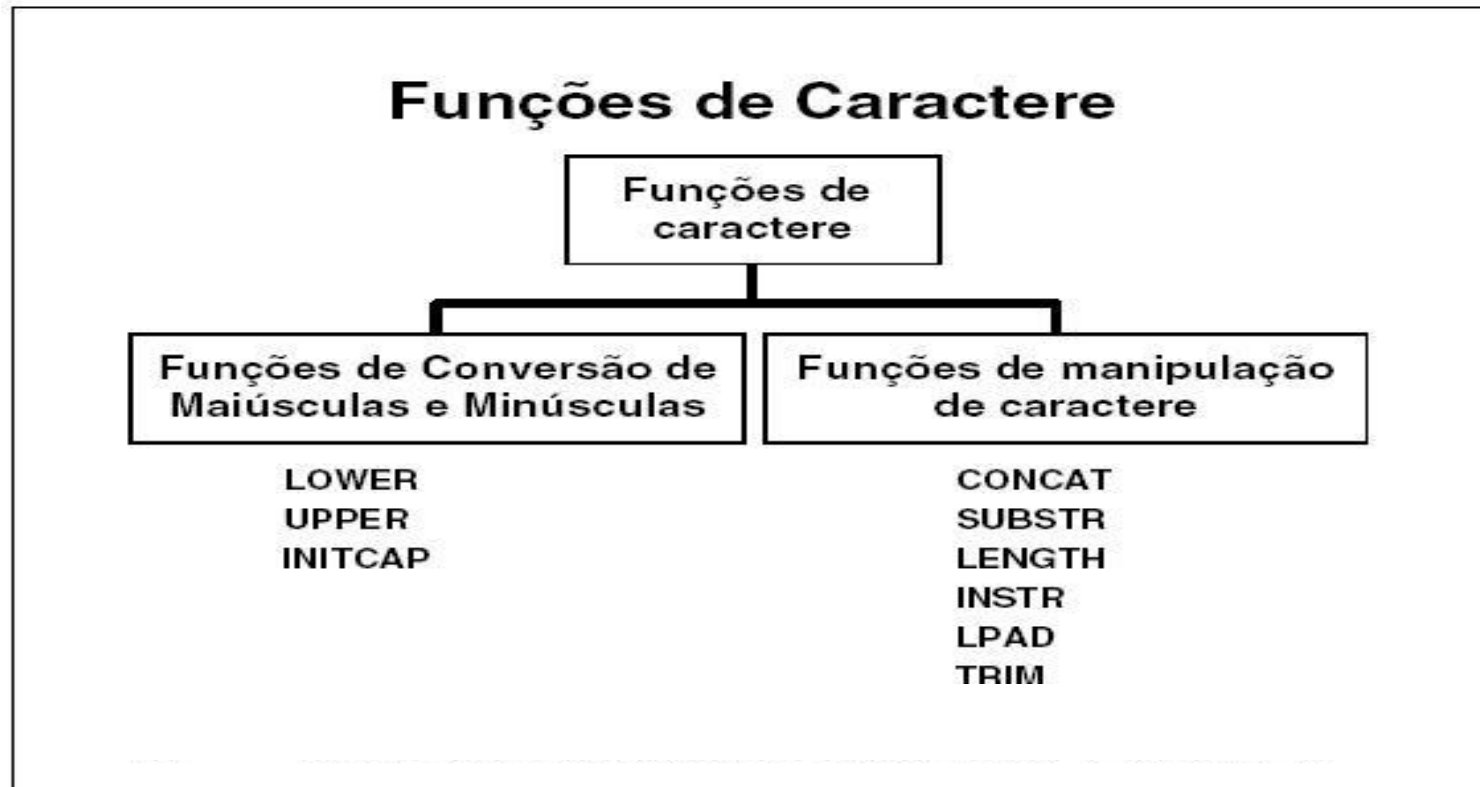
Dois Tipos de Funções SQL



- Operam somente linhas únicas e retornam um resultado por linhas
- Principais tipos :
 - Caractere
 - Número
 - Data
 - Conversão

Funções de Uma Única Linha





Linguagem SQL: Funções de Linha tipo Caractere

| Função | Objetivo |
|---|--|
| LOWER(<i>coluna expressão</i>) | Converte valores de caractere alfabético para letras minúsculas |
| UPPER(<i>coluna expressão</i>) | Converte valores de caractere alfabético para letras maiúsculas |
| INITCAP(<i>coluna expressão</i>) | Converte valores de caractere alfabético para usar maiúscula na primeira letra de cada palavra e todas as outras letras em minúsculas |
| CONCAT(<i>coluna1 expressão1, coluna2 expressão2</i>) | Concatena o primeiro valor do caractere ao segundo valor do caractere, equivalente ao operador de concatenação () |
| SUBSTR(<i>coluna expressão, m[,n]</i>) | Retorna caracteres específicos a partir do valor de caractere começando na posição <i>m</i> , até <i>n</i> caracteres depois (Se <i>m</i> for negativo, a conta inicia no final do valor de caractere. Se <i>n</i> for omitido, são retornados todos os caracteres até o final da string.) |

| Função | Objetivo |
|--|---|
| LENGTH(<i>coluna expressão</i>) | Retorna o número de caracteres do valor |
| INSTR(<i>coluna expressão, m</i>) | Retorna a posição numérica do caractere nomeado |
| LPAD(<i>coluna expressão, n, 'string'</i>) | Preenche o valor de caracter justificado à direita a uma largura total de <i>n</i> posições de caractere |
| TRIM(<i>anterior posterior ambos, trim_character FROM trim_source</i>) | Permite a você organizar cabeçalhos ou caracteres de fim de linha (ou os dois) a partir de uma string de caractere. Se <i>trim_character</i> ou <i>trim_source</i> for um caractere literal, você deve incluí-los entre aspas simples. Este é um recurso disponível a partir Oracle8i em diante. |

Funções de Manipulação de Caractere

Manipular strings de caractere

| Função | Resultado |
|------------------------------|------------|
| CONCAT(' Good ', ' String ') | GoodString |
| SUBSTR(' String ',1,3) | Str |
| LENGTH(' String ') | 6 |
| INSTR(' String ', 'r') | 3 |
| LPAD(sal,10,'*') | *****5000 |
| TRIM('S' FROM 'SSMITH') | MITH |

Usando as Funções de Manipulação de Caractere

```
SQL> SELECT ename, CONCAT (ename, job), LENGTH(ename),
2          INSTR(ename, 'A')
3 FROM      emp
4 WHERE     SUBSTR(job,1,5) = 'SALES';
```

| ENAME | CONCAT (ENAME, JOB) | LENGTH (ENAME) | INSTR (ENAME, 'A') |
|--------|---------------------|----------------|--------------------|
| MARTIN | MARTINSALESMAN | 6 | 2 |
| ALLEN | ALLENSALESMAN | 5 | 1 |
| TURNER | TURNERSALESMAN | 6 | 0 |
| WARD | WARDSALESMAN | 4 | 2 |

EXEMPLO DE FUNÇÕES DE LINHA: INITCAP, UPPER e LOWER

Planilha

Query Builder

1

2

3

4

5

6

7

8

9

SELECT

F.CD_FUNC,

F.NM_FUNCIONARIO,

INITCAP(F.NM_FUNCIONARIO) NOME_FUNC_INITCAP,

UPPER(F.NM_FUNCIONARIO) NOME_FUNC_UPPER,

LOWER(F.NM_FUNCIONARIO) NOME_FUNC_LOWER,

LENGTH(F.NM_FUNCIONARIO) QTDE_BYTES_NOME_FUNC,

SUBSTR(F.NM_FUNCIONARIO,1, INSTR(F.NM_FUNCIONARIO, ' ')) PRIMEIRO_NOME_FUNC

FROM

T_SAK_FUNCIONARIO F

ORDER BY F.NM_FUNCIONARIO;

Saída do Script

Resultado da Consulta

Resultado da Consulta 1

Resultado da Consulta 2

Resultado da Consulta 3

Resultado da C...

x

x

x

SQL

Todas as Linhas Extraídas: 10 em 0,031 segundos

| | CD_FUNC | NM_FUNCIONARIO | NOME_FUNC_INITCAP | NOME_FUNC_UPPER | NOME_FUNC_LOWER | QTDE_BYTES_NOME_FUNC | PRIMEIRO_NOME_FUNC |
|----|---------|--------------------|--------------------|--------------------|--------------------|----------------------|--------------------|
| 1 | 303 | JULIA JULIO | Julia Julio | JULIA JULIO | julia julio | 11 | JULIA |
| 2 | 102 | JUSCILENE PEREIRA | Juscilene Pereira | JUSCILENE PEREIRA | juscilene pereira | 17 | JUSCILENE |
| 3 | 101 | MARCIA KUTIZTICKS | Marcia Kutizticks | MARCIA KUTIZTICKS | marcia kutizticks | 17 | MARCIA |
| 4 | 302 | MARCIO SALMERON | Marcio Salmeron | MARCIO SALMERON | marcio salmeron | 15 | MARCIO |
| 5 | 1001 | MARIANA SOUZA | Mariana Souza | MARIANA SOUZA | mariana souza | 13 | MARIANA |
| 6 | 301 | OIVLAS SAKSPILDAP | Oivlas Sakspildap | OIVLAS SAKSPILDAP | oivlas sakspildap | 17 | OIVLAS |
| 7 | 304 | OTAVIO MILIONARIO | Otavio Milionario | OTAVIO MILIONARIO | otavio milionario | 17 | OTAVIO |
| 8 | 1000 | PAULO SILVA | Paulo Silva | PAULO SILVA | paulo silva | 11 | PAULO |
| 9 | 1002 | RAFAELA DIAS | Rafaela Dias | RAFAELA DIAS | rafaela dias | 12 | RAFAELA |
| 10 | 1003 | TESTE DA APLICAÇÃO | Teste Da Aplicação | TESTE DA APLICAÇÃO | teste da aplicação | 18 | TESTE |

Linguagem SQL: Funções de Linha tipo Caractere

```

1 SELECT F.CD_FUNC,
2        F.NM_FUNCIONARIO,
3        INITCAP(F.NM_FUNCIONARIO) NOME_FUNC_INITCAP,
4        UPPER(F.NM_FUNCIONARIO) NOME_FUNC_UPPER,
5        LOWER(F.NM_FUNCIONARIO) NOME_FUNC_LOWER
6 FROM   T_SAK_FUNCIONARIO F
7 WHERE  LOWER(F.NM_FUNCIONARIO) LIKE 'MA%'
8 ORDER BY F.NM_FUNCIONARIO;
9

```

Resultado da Consulta x | Resultado da Consulta 1 x | Resultado da Consulta 2

Tarefa concluída em 0,078 segundos

nenhuma linha selecionada

Veja a cláusula WHERE aqui. O que aconteceu ? Por que não funcionou ?

E agora, porque funcionou ?

```

1 SELECT F.CD_FUNC,
2        F.NM_FUNCIONARIO,
3        INITCAP(F.NM_FUNCIONARIO) NOME_FUNC_INITCAP,
4        UPPER(F.NM_FUNCIONARIO) NOME_FUNC_UPPER,
5        LOWER(F.NM_FUNCIONARIO) NOME_FUNC_LOWER
6 FROM   T_SAK_FUNCIONARIO F
7 WHERE  UPPER(F.NM_FUNCIONARIO) LIKE 'MA%'
8 ORDER BY F.NM_FUNCIONARIO;
9

```

Resultado da Consulta x | Resultado da Consulta 1 x | Resultado da Consulta 2 x | Saída do Script x | Pl

Todas as Linhas Extraídas: 4 em 0,017 segundos

| | CD_FUNC | NM_FUNCIONARIO | NOME_FUNC_INITCAP | NOME_FUNC_UPPER | NOME_FUNC_LOWER |
|---|---------|-------------------|-------------------|-------------------|-------------------|
| 1 | 1010 | MARCELA MATOS | Marcela Matos | MARCELA MATOS | marcela matos |
| 2 | 101 | MARCIA KUTIZTICKS | Marcia Kutizticks | MARCIA KUTIZTICKS | marcia kutizticks |
| 3 | 302 | MARCIO SALMERON | Marcio Salmeron | MARCIO SALMERON | marcio salmeron |
| 4 | 1001 | MARIANA SOUZA | Mariana Souza | MARIANA SOUZA | mariana souza |

| Função | Objetivo |
|--|---|
| ROUND(<i>coluna</i> <i>expressão</i> , <i>n</i>) | Arredonda a coluna, expressão ou valor para <i>n</i> casas decimais ou se <i>n</i> for omitido, nenhuma casa decimal (Se <i>n</i> for negativo, os números à esquerda do ponto decimal serão arredondados.) |
| TRUNC(<i>coluna</i> <i>expressão</i> , <i>n</i>) | Trunca a coluna, expressão ou valor para <i>n</i> casas decimais ou se <i>n</i> for omitido, nenhuma casa decimal (Se <i>n</i> for negativo, os números à esquerda do ponto decimal serão truncados para zero.) |
| MOD(<i>m</i> , <i>n</i>) | Retorna o resto de <i>m</i> dividido por <i>n</i> . |

The screenshot shows the Oracle SQL*Plus interface with two SQL queries and their results. The first query uses the ROUND function with different precision and scale values. The second query uses the TRUNC function with different precision and scale values.

```

SQL> SELECT ROUND(45.923,2) CALC1, ROUND(45.923,0) CALC2,
2         ROUND(45.923,-1) CALC3, ROUND(44.923,-1) CALC4
3 FROM DUAL;

  CALC1    CALC2    CALC3    CALC4
-----
  45,92      46      50      40

SQL> SELECT TRUNC(45.923,2) TRUNC1, TRUNC(45.923,0) TRUNC2,
2         TRUNC(45.923,-1) TRUNC3, TRUNC(45.923,-2) TRUNC4
3 FROM DUAL;

 TRUNC1    TRUNC2    TRUNC3    TRUNC4
-----
  45,92      45      40      0
    
```

Linguagem SQL: Funções de Linha tipo Numérico

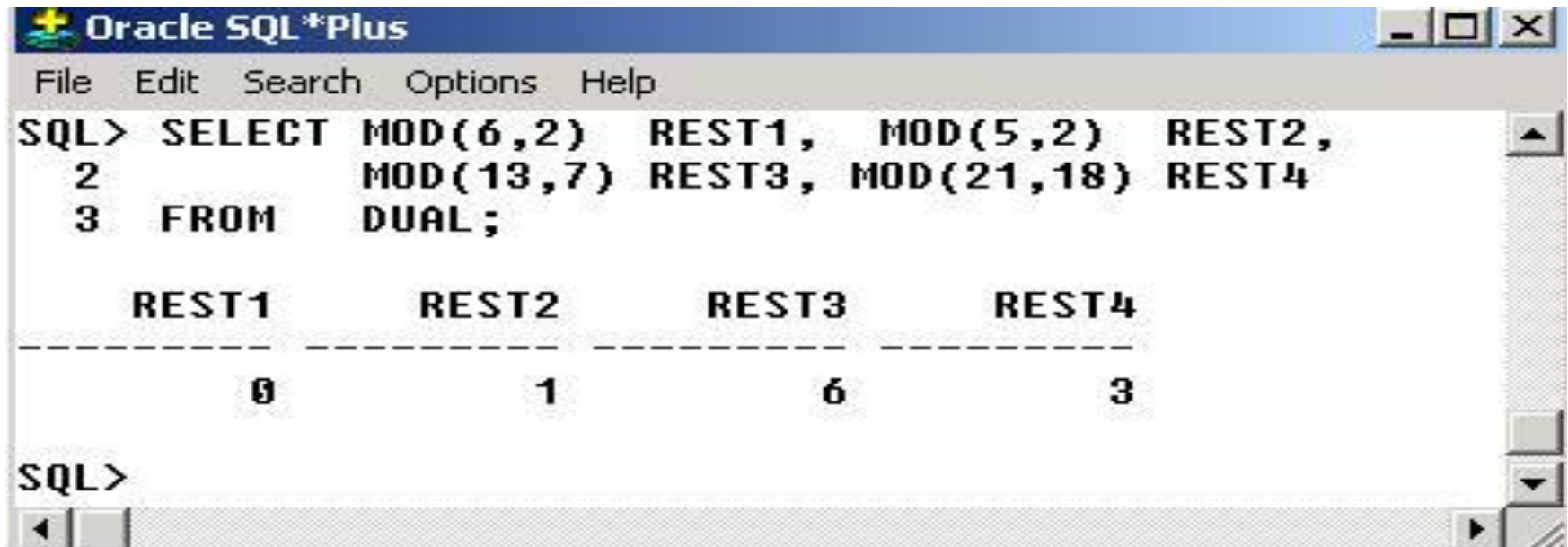
| Planilha | | Query Builder | | | | | | |
|----------|--------|--|--|--|--|--|--|--|
| 1 | SELECT | D.CD_DEPTO, | | | | | | |
| 2 | | D.NM_DEPTO, | | | | | | |
| 3 | | F.CD_FUNC, | | | | | | |
| 4 | | F.NM_FUNCIONARIO, | | | | | | |
| 5 | | F.VL_SALARIO, | | | | | | |
| 6 | | ROUND(F.VL_SALARIO,1) Salario_round_1, | | | | | | |
| 7 | | ROUND(F.VL_SALARIO,0) Salario_round_0, | | | | | | |
| 8 | | TRUNC(F.VL_SALARIO,1) Salario_trunc_1 | | | | | | |
| 9 | FROM | T_SAK_FUNCIONARIO F INNER JOIN T_SAK_DEPTO D | | | | | | |
| 10 | ON | (F.CD_DEPTO = D.CD_DEPTO); | | | | | | |

| | | | | | |
|-----------------------|-------------------------|-------------------------|-----------------|---------------------|-------------------------|
| Resultado da Consulta | Resultado da Consulta 1 | Resultado da Consulta 2 | Saída do Script | Plano de Explicação | Resultado da Consulta 3 |
|-----------------------|-------------------------|-------------------------|-----------------|---------------------|-------------------------|

SQL | Todas as Linhas Extraídas: 13 em 0,024 segundos

| CD_DEPTO | NM_DEPTO | CD_FUNC | NM_FUNCIONARIO | VL_SALARIO | SALARIO_ROUND_1 | SALARIO_ROUND_0 | SALARIO_TRUNC_1 |
|----------|---|---------|--------------------|------------|-----------------|-----------------|-----------------|
| 1 | 1234 TECNOLOGIA DA INFORMAÇÃO CORPORATIVA | 1000 | PAULO SILVA | 18176,69 | 18176,7 | 18177 | 18176,6 |
| 2 | 1234 TECNOLOGIA DA INFORMAÇÃO CORPORATIVA | 1001 | MARIANA SOUZA | 7362,96 | 7363 | 7363 | 7362,9 |
| 3 | 1234 TECNOLOGIA DA INFORMAÇÃO CORPORATIVA | 1002 | RAFAELA DIAS | 20957,84 | 20957,8 | 20958 | 20957,8 |
| 4 | 1234 TECNOLOGIA DA INFORMAÇÃO CORPORATIVA | 1003 | TESTE DA APLICAÇÃO | 13499,37 | 13499,4 | 13499 | 13499,3 |
| 5 | 10 FINANCEIRO | 101 | MARCIA KUTIZTICKS | 10000,54 | 10000,5 | 10001 | 10000,5 |
| 6 | 10 FINANCEIRO | 102 | JUSCILENE PEREIRA | 5000,67 | 5000,7 | 5001 | 5000,6 |
| 7 | 30 DIRETORIA | 301 | OIVLAS SAKSPILDAP | 20000,13 | 20000,1 | 20000 | 20000,1 |
| 8 | 60 CONTAS A PAGAR | 302 | MARCIO SALMERON | 5679,54 | 5679,5 | 5680 | 5679,5 |
| 9 | 60 CONTAS A PAGAR | 303 | JULIA JULIO | 2256,67 | 2256,7 | 2257 | 2256,6 |
| 10 | 70 CONTAS A RECEBER | 304 | OTAVIO MILIONARIO | 8766,13 | 8766,1 | 8766 | 8766,1 |
| 11 | 500 CONTABILIDADE | 1010 | MARCELA MATOS | 4234,76 | 4234,8 | 4235 | 4234,7 |
| 12 | 500 CONTABILIDADE | 1011 | RODRIGO RODRIGUES | 7654,22 | 7654,2 | 7654 | 7654,2 |
| 13 | 500 CONTABILIDADE | 1012 | WILLIAM WILL | 45654,22 | 45654,2 | 45654 | 45654,2 |

- Função MOD(num1, num2) retorna o “resto” da divisão entre 2 números



The screenshot shows the Oracle SQL*Plus interface. The title bar reads "Oracle SQL*Plus". The menu bar includes "File", "Edit", "Search", "Options", and "Help". The command window contains the following SQL query:

```
SQL> SELECT MOD(6,2) REST1, MOD(5,2) REST2,  
2          MOD(13,7) REST3, MOD(21,18) REST4  
3 FROM DUAL;
```

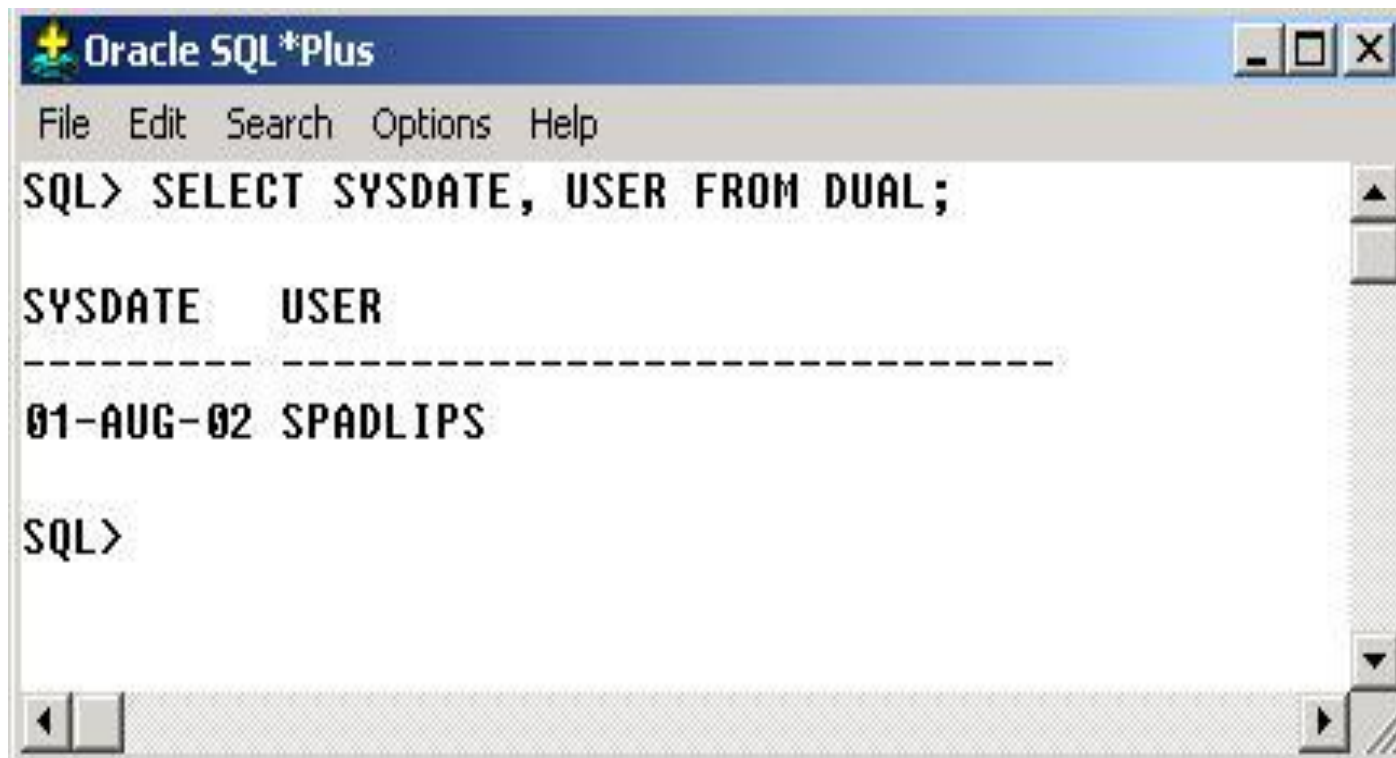
The query results are displayed in a table format:

| REST1 | REST2 | REST3 | REST4 |
|-------|-------|-------|-------|
| 0 | 1 | 6 | 3 |

The prompt "SQL>" is visible at the bottom left of the window.

Trabalhando com Datas

- O Oracle armazena datas em um formato numérico interno: século, ano, mês, dia, horas, minutos, segundos.
- O formato de data default é DD-MON-YY.
- **SYSDATE** é uma função de retorno de data e hora.
- **DUAL** é uma tabela fictícia usada para visualizar **SYSDATE**.



The screenshot shows the Oracle SQL*Plus application window. The title bar reads "Oracle SQL*Plus". The menu bar includes "File", "Edit", "Search", "Options", and "Help". The command prompt shows the user has entered the command "SQL> SELECT SYSDATE, USER FROM DUAL;". The output is displayed in a table format with two columns: "SYSDATE" and "USER". The first row of data shows the date "01-AUG-02" and the username "SPADLIPS". The prompt "SQL>" is visible at the bottom of the window.

```

Oracle SQL*Plus
File Edit Search Options Help
SQL> SELECT SYSDATE, USER FROM DUAL;

SYSDATE    USER
-----
01-AUG-02  SPADLIPS

SQL>
  
```

Aritmética com Datas

- Adicionar ou subtrair um número de, ou para, uma data para um valor de *data* resultante.
- Subtrair duas datas a fim de localizar o *número* de dias entre estas datas.
- Adicionar *horas* para uma data dividindo o número de horas por 24.

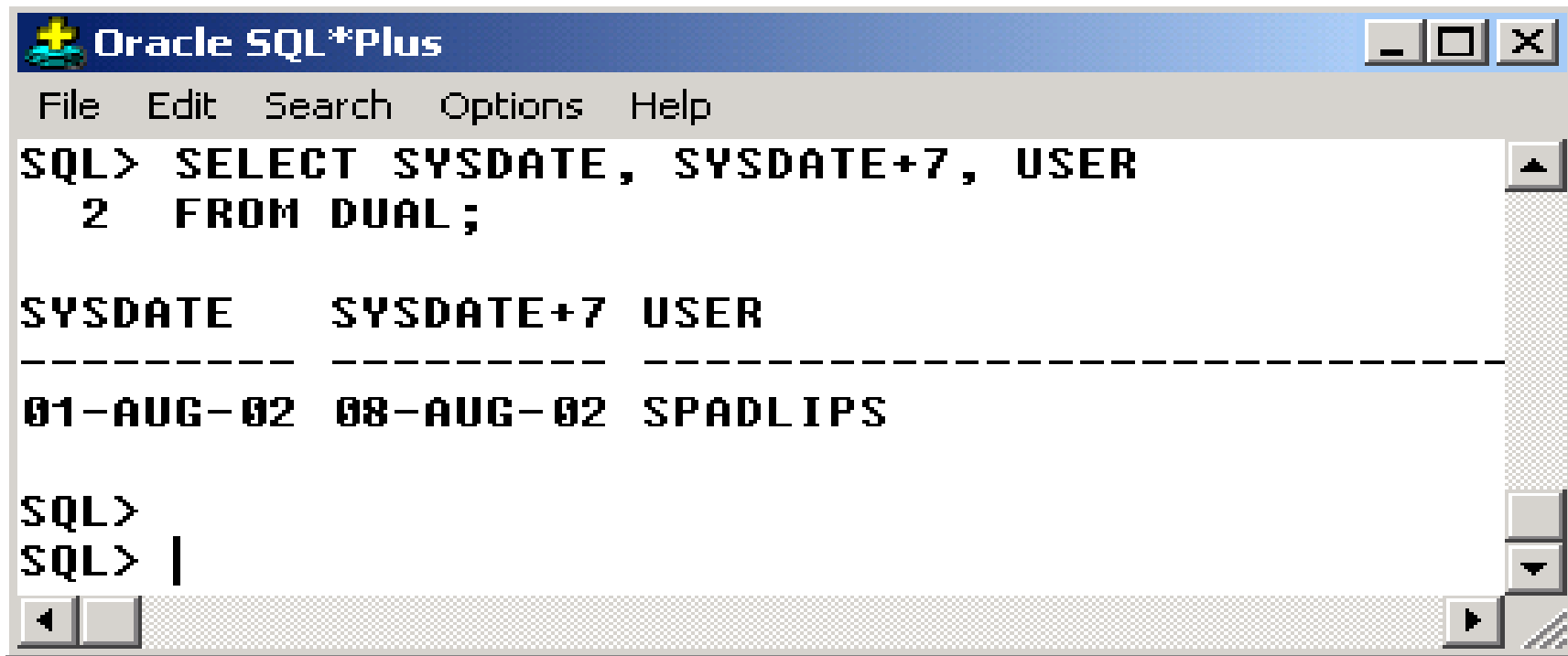
Aritmética com Datas

Já que os bancos de dados armazenam datas como números, você pode executar cálculos usando operadores aritméticos como adição e subtração. É possível adicionar e subtrair constantes de número bem como datas.

Você pode executar as seguintes operações:

| Operação | Resultado | Descrição |
|------------------|----------------|---|
| data + número | Data | Adiciona um número de dias para uma data |
| data - número | Data | Subtrai um número de dias de uma data |
| data - data | Número de dias | Subtrai uma data de outra |
| data + número/24 | Data | Adiciona um número de horas para uma data |

- Exemplo de cálculo utilizando datas



The screenshot shows the Oracle SQL*Plus window. The title bar is blue with the Oracle logo and the text "Oracle SQL*Plus". The menu bar includes "File", "Edit", "Search", "Options", and "Help". The command prompt shows the following SQL query:

```
SQL> SELECT SYSDATE, SYSDATE+7, USER  
2 FROM DUAL;
```

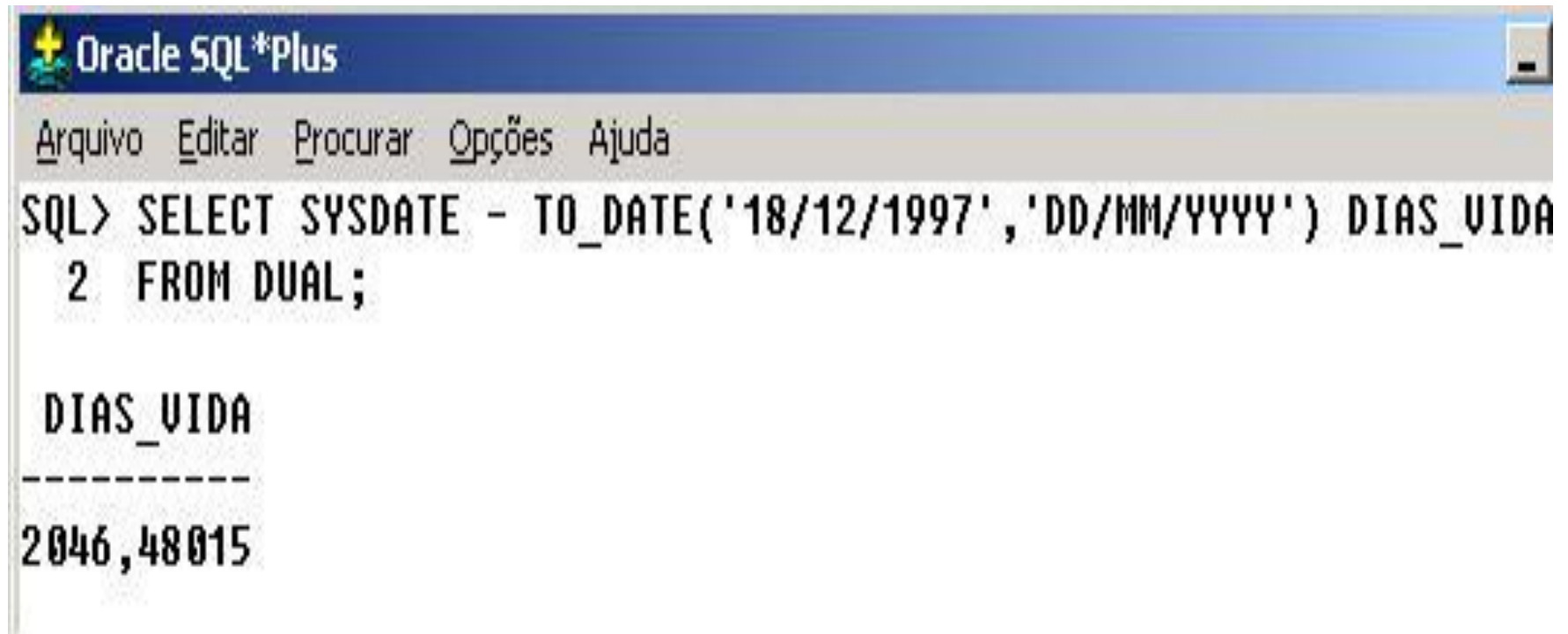
The query result is displayed in a table format with three columns: SYSDATE, SYSDATE+7, and USER. The data row shows the current date as 01-AUG-02, the date 7 days later as 08-AUG-02, and the user as SPADLIPS.

| SYSDATE | SYSDATE+7 | USER |
|-----------|-----------|----------|
| 01-AUG-02 | 08-AUG-02 | SPADLIPS |

The command prompt shows the user entering the next command:

```
SQL>  
SQL> |
```

- Exemplo de cálculo mostrando quantos dias de vida você tem no momento da execução do comando.



The screenshot shows the Oracle SQL*Plus interface. The title bar reads "Oracle SQL*Plus". The menu bar includes "Arquivo", "Editar", "Procurar", "Opções", and "Ajuda". The command prompt shows the following SQL query:

```
SQL> SELECT SYSDATE - TO_DATE('18/12/1997','DD/MM/YYYY') DIAS_VIDA  
2 FROM DUAL;
```

The result of the query is displayed below the command:

| DIAS_VIDA |
|------------|
| 2046,48015 |

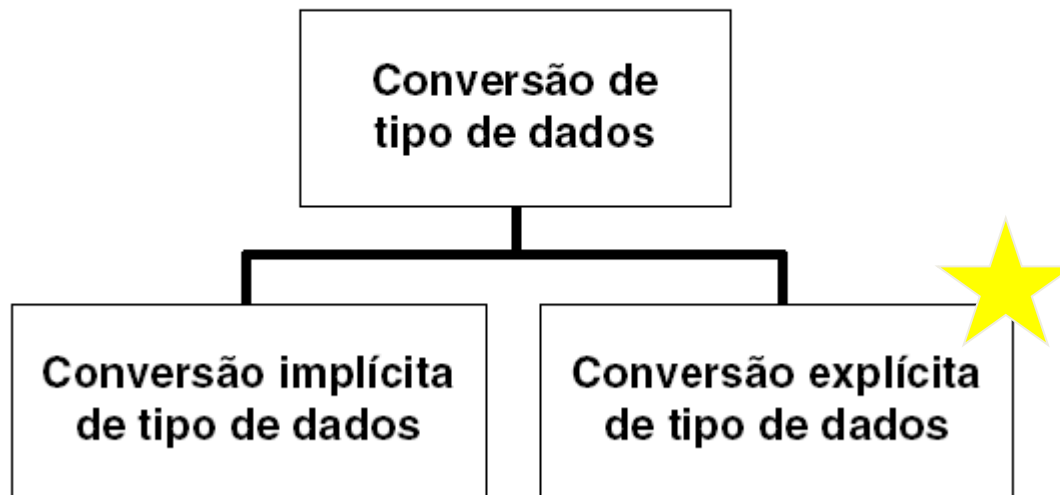
Funções de Data

| Função | Descrição |
|-----------------------|--|
| MONTHS_BETWEEN | Número de meses entre duas datas |
| ADD_MONTHS | Adiciona meses de calendário para a data |
| NEXT_DAY | Dia seguinte da data especificada |
| LAST_DAY | Último dia do mês |
| ROUND | Data de arredondamento |
| TRUNC | Data para truncada |

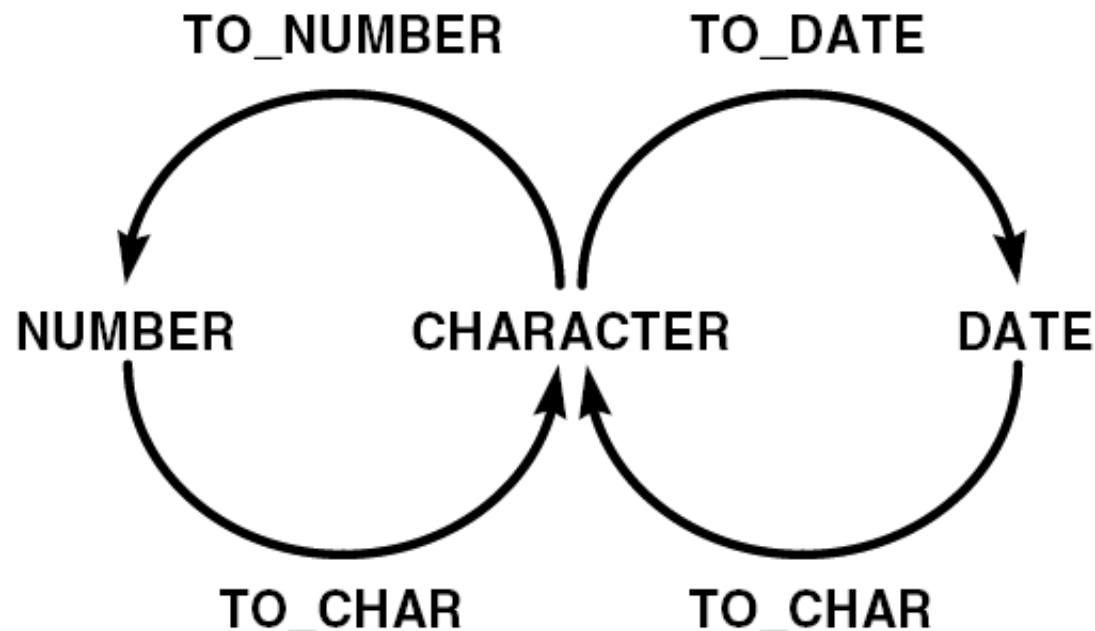
Usando Funções de Data

- MONTHS_BETWEEN ('01-SEP-95','11-JAN-94')
→ 19.6774194
- ADD_MONTHS ('11-JAN-94',6) → '11-JUL-94'
- NEXT_DAY ('01-SEP-95','FRIDAY') → '08-SEP-95'
- LAST_DAY('01-SEP-95') → '30-SEP-95'

Funções de Conversão



Conversão Explícita de Tipo de Dados



Usando a Função TO_CHAR com Datas

```
SQL> SELECT ename,
2      TO_CHAR(hiredate, 'fmDD Month YYYY') HIREDATE
3  FROM emp;
```

| ENAME | HIREDATE |
|--------|-------------------|
| ----- | ----- |
| KING | 17 November 1981 |
| BLAKE | 1 May 1981 |
| CLARK | 9 June 1981 |
| JONES | 2 April 1981 |
| MARTIN | 28 September 1981 |
| ALLEN | 20 February 1981 |
| ... | |

14 rows selected.

Função TO_CHAR com Números

Ao trabalhar com valores de número tais como strings de caractere, você deve converter esse números para o tipo de dados de caractere usando a função TO_CHAR , que traduz o tipo de dados de NUMBER para tipo de dados VARCHAR2. Essa técnica é particularmente útil com concatenação.

Elementos de Formato de Número

Se estiver convertendo um número para tipo de dados de caractere, você pode usar os seguintes elementos:

| Elemento | Descrição | Exemplo | Resultado |
|----------|--|------------|-----------|
| 9 | Posição numérica (número de 9s determinam o tamanho da exibição) | 999999 | 1234 |
| 0 | Exibe zeros à esquerda | 099999 | 001234 |
| \$ | Sinal de dólar flutuante | \$999999 | \$1234 |
| L | Símbolo da moeda local flutuante | L999999 | FF1234 |
| . | Ponto decimal na posição especificada | 999999.99 | 1234.00 |
| , | Vírgula na posição especificada | 999,999 | 1,234 |
| MI | Sinais de menos à direita (valores negativos) | 999999MI | 1234- |
| PR | Coloca números negativos entre parênteses | 999999PR | <1234> |
| EEEE | Notação científica (formato deve especificar quatro Es) | 99.999EEEE | 1.234E+03 |
| V | Multiplica por 10 <i>n</i> vezes (<i>n</i> = número de 9s após o V) | 9999V99 | 123400 |
| B | Exibe valores de zero como espaço, não 0 | B9999.99 | 1234.00 |

Funções TO_NUMBER e TO_DATE

- Converter uma string de caractere para um formato de número usando a função TO_NUMBER

```
TO_NUMBER(carac[, 'fmt'])
```

- Converter uma string de caractere para um formato de data usando a função TO_DATE

```
TO_DATE(carac[, 'fmt'])
```

Função DECODE

Facilita pesquisas condicionais realizando o trabalho de uma instrução CASE ou IF-THEN-ELSE

```
DECODE(col/express, pesquisa1, resultado1  
        [, pesquisa2, resultado2, ..., ]  
        [, default])
```

Usando a Função DECODE

```
SQL> SELECT job, sal,
2          DECODE(job, 'ANALYST', SAL*1.1,
3                    'CLERK',   SAL*1.15,
4                    'MANAGER', SAL*1.20,
5                              SAL)
6          REVISED_SALARY
7 FROM emp;
```

| JOB | SAL | REVISED_SALARY |
|-----------|------|----------------|
| PRESIDENT | 5000 | 5000 |
| MANAGER | 2850 | 3420 |
| MANAGER | 2450 | 2940 |
| ... | | |

14 rows selected.

Linguagem SQL: Funções de Linha: DECODE

Planilha
Query Builder

```

1 SELECT F.CD_FUNC,
2         F.NM_FUNCIONARIO,
3         F.DS_ESTADO_CIVIL,
4         F.DT_NASCIMENTO,
5         T.CD_TELEFONE SEQUENCIA_TELEFONE,
6         --CONCAT(T.NR_DDD, ' ', CONCAT(T.NR_TELEFONE, ' ')) NUMERO_TELEFONE,
7         CONCAT( CONCAT( CONCAT( CONCAT( CONCAT( '(', T.NR_DDD ), ') ' ), SUBSTR( T.NR_TELEFONE,1,5) ), '-'), SUBSTR(T.NR_TELEFONE,5,4)) NUMERO_TELEFONE,
8         DECODE(T.TP_TELEFONE, 1, 'RESIDENCIAL', 2, 'CELULAR PROPRIO', 3, 'COMERCIAL', 'OUTROS') TIPO_TELEFONE
9 FROM   T_SAK_FUNCIONARIO F INNER JOIN T_SAK_TELEFONE T
10 ON     (F.CD_FUNC = T.CD_FUNC);

```

Saída do Script x
Resultado da Consulta x
Resultado da Consulta 1 x
Resultado da Consulta 2 x
Resultado da Consulta 3 x
Resultado da Consulta 4 x
Resultado da Consulta 5 x
Resu

SQL
Todas as Linhas Extraídas: 6 em 0,033 segundos

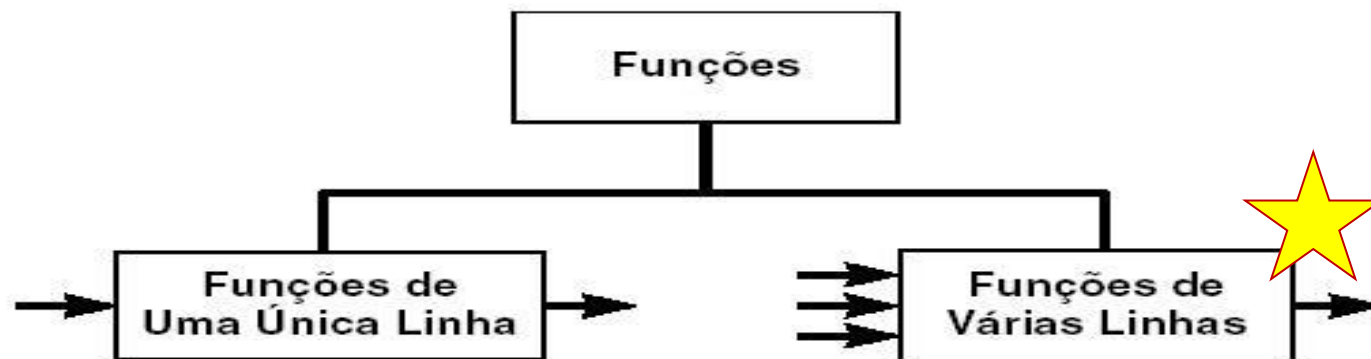
| | CD_FUNC | NM_FUNCIONARIO | DS_ESTADO_CIVIL | DT_NASCIMENTO | SEQUENCIA_TELEFONE | NUMERO_TELEFONE | TIPO_TELEFONE |
|---|---------|--------------------|-----------------|---------------|--------------------|-------------------|-----------------|
| 1 | 1002 | RAFAELA DIAS | CASADA | 18/12/74 | | 1 (13) 33540-0421 | CELULAR PROPRIO |
| 2 | 1003 | TESTE DA APLICAÇÃO | DIVORCIADO | 12/12/80 | | 1 (11) 99157-7666 | CELULAR PROPRIO |
| 3 | 101 | MARCIA KUTIZTICKS | SOLTEIRA | 30/05/67 | | 1 (11) 99157-7666 | CELULAR PROPRIO |
| 4 | 102 | JUSCILENE PEREIRA | SOLTEIRA | 25/07/03 | | 1 (11) 24567-7788 | RESIDENCIAL |
| 5 | 102 | JUSCILENE PEREIRA | SOLTEIRA | 25/07/03 | | 2 (21) 98065-5442 | CELULAR PROPRIO |
| 6 | 102 | JUSCILENE PEREIRA | SOLTEIRA | 25/07/03 | | 3 (15) 76541-1122 | COMERCIAL |

Sumário

Use as funções para realizar o seguinte:

- **Executar cálculos usando dados**
- **Modificar itens de dados individuais**
- **Manipular saída para grupos de linhas**
- **Alterar formatos de data para exibição**
- **Converter tipos de dados de coluna**

Dois Tipos de Funções SQL



Objetivos

Depois de completar esta lição, você poderá fazer o seguinte:

- Identificar as funções de grupo disponíveis
- Descrever o uso de funções de grupo
- Agrupar dados usando a cláusula **GROUP BY**
- Incluir ou excluir linhas agrupadas usando a cláusula **HAVING**

- **Diferente das funções que manipulam uma linha** de cada vez, as **funções de grupo operam em conjuntos de linhas** para fornecer um resultado por grupo;
- As operações das funções de grupo **podem envolver todas as linhas de uma tabela ou um conjuntos de linhas** definidos por critérios preestabelecidos;
- Assim como as funções de diversas outras linguagens, as funções disponíveis, na linguagem SQL, requerem argumentos para realizar operações e retornar valores.

O Que São Funções de Grupo?

As funções de grupo operam em conjuntos de linhas para fornecer um resultado por grupo.

EMP

| DEPTNO | SAL |
|--------|------|
| 10 | 2450 |
| 10 | 5000 |
| 10 | 1300 |
| 20 | 800 |
| 20 | 1100 |
| 20 | 3000 |
| 20 | 3000 |
| 20 | 2975 |
| 30 | 1600 |
| 30 | 2850 |
| 30 | 1250 |
| 30 | 950 |
| 30 | 1500 |
| 30 | 1250 |

"salário
máximo na
tabela EMP"

MAX (SAL)

5000

Usando Funções de Grupo

```
SELECT      [coluna,] group_function(coluna)
FROM        tabela
[WHERE      condição]
[GROUP BY   coluna]
[ORDER BY   coluna];
```

| FUNÇÃO | DESCRIÇÃO |
|--------------|---|
| AVG () | Retorna a média obtida entre os valores de um conjunto. |
| COUNT () | Retorna a quantidade de ocorrências. |
| MAX () | Retorna o maior valor de um conjunto. |
| MIN () | Retorna o menor valor de um conjunto. |
| SUM () | Retorna a somatória dos valores de um conjunto. |
| STDDEV () | Retorna o desvio padrão do conjunto. |
| VARIANCE () | Retorna a variância do conjunto. |

Usando Funções AVG e SUM

Você pode usar AVG e SUM para dados numéricos.

```
SQL> SELECT  AVG(sal), MAX(sal),
2           MIN(sal), SUM(sal)
3 FROM      emp
4 WHERE     job LIKE 'SALES%';
```

| AVG (SAL) | MAX (SAL) | MIN (SAL) | SUM (SAL) |
|-----------|-----------|-----------|-----------|
| ----- | ----- | ----- | ----- |
| 1400 | 1600 | 1250 | 5600 |

Usando Funções MIN e MAX

Você pode usar MIN e MAX para qualquer tipo de dados.

```
SQL> SELECT MIN(hiredate), MAX(hiredate)
         2 FROM emp;
```

| MIN (HIRED | MAX (HIRED |
|------------|------------|
| ----- | ----- |
| 17-DEC-80 | 12-JAN-83 |

Qual é a diferença entre os 2 comandos abaixo?

Usando a Função COUNT

COUNT(*) retorna o número de linhas em uma tabela.

```
SQL> SELECT COUNT(*)
2 FROM emp
3 WHERE deptno = 30;
```

```
COUNT(*)
-----
6
```


Usando a Função COUNT


COUNT(expr) retorna o número de linhas não nulas.


```
SQL> SELECT COUNT(comm)
2 FROM emp
3 WHERE deptno = 30;
```


```
COUNT (COMM)
-----
4
```





| Planilha | | Query Builder | |
|----------|---|---------------|--|
| 1 | SELECT COUNT(*) QTD_LINHAS, COUNT(P.DT_TERMINO) | | |
| 2 | FROM T_SAK_PROJETO P; | | |
| 3 | | | |
| 4 | | | |

 Saída do Script x

 Resultado da Consulta x

 Resultado da Consulta 1 x

 Res



SQL

Todas as Linhas Extraídas: 1 em 0,018 segundos

| | QTD_LINHAS | COUNT(P.DT_TERMINO) |
|---|------------|---------------------|
| 1 | 4 | 1 |

Criando Grupos de Dados

EMP

| DEPTNO | SAL | | |
|--------|------|-----------|--|
| 10 | 2450 | | |
| 10 | 5000 | 2916.6667 | |
| 10 | 1300 | | |
| 20 | 800 | | |
| 20 | 1100 | 2175 | |
| 20 | 3000 | | |
| 20 | 3000 | | |
| 20 | 2975 | | |
| 30 | 1600 | | |
| 30 | 2850 | | |
| 30 | 1250 | 1566.6667 | |
| 30 | 950 | | |
| 30 | 1500 | | |
| 30 | 1250 | | |

| DEPTNO | AVG (SAL) |
|--------|-----------|
| 10 | 2916.6667 |
| 20 | 2175 |
| 30 | 1566.6667 |

Usando a Cláusula GROUP BY

Todas as colunas na lista SELECT que não estejam em funções de grupo devem estar na cláusula GROUP BY.

```
SQL> SELECT deptno, AVG(sal)
2 FROM emp
3 GROUP BY deptno;
```

| DEPTNO | AVG(SAL) |
|--------|-----------|
| 10 | 2916.6667 |
| 20 | 2175 |
| 30 | 1566.6667 |

- Exiba para o departamento de número 10, o Maior Salário, o Menor Salário, a Média Salarial, a Soma Total de Salários e a Quantidade Total de Funcionários que trabalham nesse departamento.

Planilha Query Builder

```

1 SELECT D.CD_DEPTO,
2       D.NM_DEPTO,
3       MAX( F.VL_SALARIO ) MAIOR_SALARIO_DEPTO,
4       MIN( F.VL_SALARIO ) MENOR_SALARIO_DEPTO,
5       AVG( F.VL_SALARIO ) MEDIA_SALARIAL_DEPTO,
6       SUM( F.VL_SALARIO ) SOMA_TOTAL_SALARIO_DEPTO,
7       COUNT(*)          QTDE_FUNC_DEPTO
8 FROM   T_SAK_FUNCIONARIO F INNER JOIN T_SAK_DEPTO D
9 ON (F.CD_DEPTO = D.CD_DEPTO)
10 WHERE D.CD_DEPTO = 10
11 GROUP BY D.CD_DEPTO,
12          D.NM_DEPTO;
13

```

Saída do Script x Resultado da Consulta x Resultado da Consulta 1 x Resultado da Consulta 2 x Resultado da Consulta 3 x Resultado da Consulta 4 x Res

SQL | Todas as Linhas Extraídas: 1 em 0,025 segundos

| | CD_DEPTO | NM_DEPTO | MAIOR_SALARIO_DEPTO | MENOR_SALARIO_DEPTO | MEDIA_SALARIAL_DEPTO | SOMA_TOTAL_SALARIO_DEPTO | QTDE_FUNC_DEPTO |
|---|----------|------------|---------------------|---------------------|----------------------|--------------------------|-----------------|
| 1 | 10 | FINANCEIRO | 10000,54 | 5000,67 | 7500,605 | 15001,21 | 2 |

Excluindo Resultados do Grupo

EMP

| DEPTNO | SAL | | | |
|--------|------|------|--|--|
| 10 | 2450 | | | |
| 10 | 5000 | 5000 | | |
| 10 | 1300 | | | |
| 20 | 800 | | | |
| 20 | 1100 | | | |
| 20 | 3000 | 3000 | | |
| 20 | 3000 | | | |
| 20 | 2975 | | | |
| 30 | 1600 | | | |
| 30 | 2850 | | | |
| 30 | 1250 | 2850 | | |
| 30 | 950 | | | |
| 30 | 1500 | | | |
| 30 | 1250 | | | |

"salário máximo por departamento maior do que US\$ 2.900"

| DEPTNO | MAX (SAL) |
|--------|-----------|
| 10 | 5000 |
| 20 | 3000 |

Excluindo Resultados do Grupo: Cláusula HAVING

Use a cláusula HAVING para restringir grupos

- As linhas são agrupadas.
- A função de grupo é aplicada.
- Os grupos que correspondem à cláusula HAVING são exibidos.

```
SELECT      coluna, group_function
FROM        tabela
[WHERE      condição]
[GROUP BY   group_by_expression]
[HAVING     group_condition]
[ORDER BY   coluna];
```

Navigation icons: back, forward, search, etc.

- Na utilização de funções de agrupamento, muitas vezes, é necessário estabelecer certas condições para conseguir um resultado específico;
- Entre essas **condições**, está o caso da instrução **WHERE**, que seleciona as linhas que participarão do **agrupamento** da instrução **HAVING**, que deve ser utilizada para aplicar uma condição de seleção sobre as linhas agrupadas;
- **Recomenda-se** que a **instrução HAVING** seja **incluída após** a instrução **GROUP BY**.

Linguagem SQL: Funções de Grupo de Linhas

- Exiba as seguintes informações do departamento: Seu código e nome, o Maior Salário, o Menor Salário, a Média Salarial, a Soma Total de Salários e a Quantidade Total de Funcionários que trabalham nesse departamento. **Exiba somente os departamentos que tenham mais de 1 funcionario cadastrado.**

| Planilha | | Query Builder | | | | | |
|----------|--|---|--|--|--|--|--|
| 1 | | SELECT D.CD_DEPTO, | | | | | |
| 2 | | D.NM_DEPTO, | | | | | |
| 3 | | MAX(F.VL_SALARIO) MAIOR_SALARIO_DEPTO, | | | | | |
| 4 | | MIN(F.VL_SALARIO) MENOR_SALARIO_DEPTO, | | | | | |
| 5 | | AVG(F.VL_SALARIO) MEDIA_SALARIAL_DEPTO, | | | | | |
| 6 | | SUM(F.VL_SALARIO) SOMA_TOTAL_SALARIO_DEPTO, | | | | | |
| 7 | | COUNT(*) QTDE_FUNC_DEPTO | | | | | |
| 8 | | FROM T_SAK_FUNCIONARIO F INNER JOIN T_SAK_DEPTO D | | | | | |
| 9 | | ON (F.CD_DEPTO = D.CD_DEPTO) | | | | | |
| 10 | | GROUP BY D.CD_DEPTO, | | | | | |
| 11 | | D.NM_DEPTO | | | | | |
| 12 | | HAVING COUNT(*) > 1 | | | | | |
| 13 | | ORDER BY COUNT(*) ASC; | | | | | |
| 14 | | | | | | | |
| 15 | | | | | | | |

| | | | | | | | | |
|--|---|-----------------------|-------------------------|-------------------------|--------------------------|-------------------------|-------------------------|-----|
| Saída do Script | | Resultado da Consulta | Resultado da Consulta 2 | Resultado da Consulta 3 | Resultado da Consulta 4 | Resultado da Consulta 5 | Resultado da Consulta 6 | Res |
| SQL Todas as Linhas Extraídas: 3 em 0,018 segundos | | | | | | | | |
| CD_DEPTO | NM_DEPTO | MAIOR_SALARIO_DEPTO | MENOR_SALARIO_DEPTO | MEDIA_SALARIAL_DEPTO | SOMA_TOTAL_SALARIO_DEPTO | QTDE_FUNC_DEPTO | | |
| 1 | 10 FINANCEIRO | 10000,54 | 5000,67 | 7500,605 | 15001,21 | 2 | | |
| 2 | 60 CONTAS A PAGAR | 5679,54 | 2256,67 | 3968,105 | 7936,21 | 2 | | |
| 3 | 1234 TECNOLOGIA DA INFORMAÇÃO CORPORATIVA | 20957,84 | 7362,96 | 14999,215 | 59996,86 | 4 | | |

REFERÊNCIAS



- MACHADO, Felipe Nery R. Banco de Dados - Projeto e Implementação. Érica, 2004.
- Páginas: 330, 331.
- ELMASRI, R.; NAVATHE, S.B. Sistemas de Banco de Dados: Fundamentos e Aplicações. Pearson, 2005. Páginas: 153, 154.
- PRICE, JASON, ORACLE DATABASE 11 g – SQL Domine SQL e PL-SQL no banco de Dados Oracle, Bookman, 2008. Capítulos: 2.

Outros:

- Manual Oficial Oracle – Introdução ao Oracle 9i (SQL) - Oracle Corporation, 2000, 2001.

Copyright © 2023 Prof. Diogo Alves

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).