

FIAP GRADUAÇÃO

# ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

**ESTRATÉGIA E IMPLEMENTAÇÃO DE ESTRUTURAS DE DADOS**

Profa. Rita de Cássia Rodrigues  
rita@fiap.com.br

AULA 31 - COMANDOS DDL/DRL (SQL)  
(DRL → LINGUAGEM PARA RECUPERAÇÃO DE DADOS)  
(DDL → LINGUAGEM PARA CONSULTA DE DADOS)  
FUNÇÕES SQL / ORACLE

---

# | Agenda

FIAP

- ✓ Objetivo
- ✓ Funções de SQL/ORACLE
- ✓ Exercícios

- ❑ Aplicar os conceitos da linguagem SQL durante a implementação do banco de dados, utilizando funções SQL/ORACLE.

## Conteúdo Programático referente a esta aula

- ❑ Linguagem para consulta/recuperação de dados
  - ❑ DRS/DQL (SELECT)
    - ❑ Funções SQL/ORACLE
  - ❑ Exercícios

# | Comandos SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

### UTILIZANDO FUNÇÕES



As funções não são necessariamente padronizadas , logo podemos ter funções que existam no SGBD SQL Server e que não existam no SGBD Oracle.

As funções devem retornar sempre UM valor, podendo eventualmente voltar mais do que um, nesse caso o seu uso em comandos SQL DML pode ter restrições.

As funções podem ter parâmetros (um ou mais) , que passam informações para que a função possa ser processada. Existem funções porém que não precisam de valores para serem processadas, neste caso existe apenas um retorno.

# | Comandos SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

### UTILIZANDO FUNÇÕES



Os parâmetros de uma função normalmente são identificados por estarem dentro de parênteses e os mesmos ficam separados por vírgula (caso sejam maiores que um).

O retorno de uma função pode ser um valor numérico, “string”, data ou eventualmente NULO.

# Comandos SQL

## FUNÇÕES SQL / ORACLE

Utilizaremos o exercício IMPLANTAÇÃO DE PROJETOS na manipulação dos dados utilizando SQL.



Scripts necessários para os exemplos desta aula:

- ✓ SCRIPT\_AULA31\_DDL\_IMPLANTACAO\_PROJETO.SQL
- ✓ SCRIPT\_AULA31\_DML\_IMPLANTACAO\_PROJETO.SQL

**Nota:** Os arquivos se encontram no portal da FIAP.

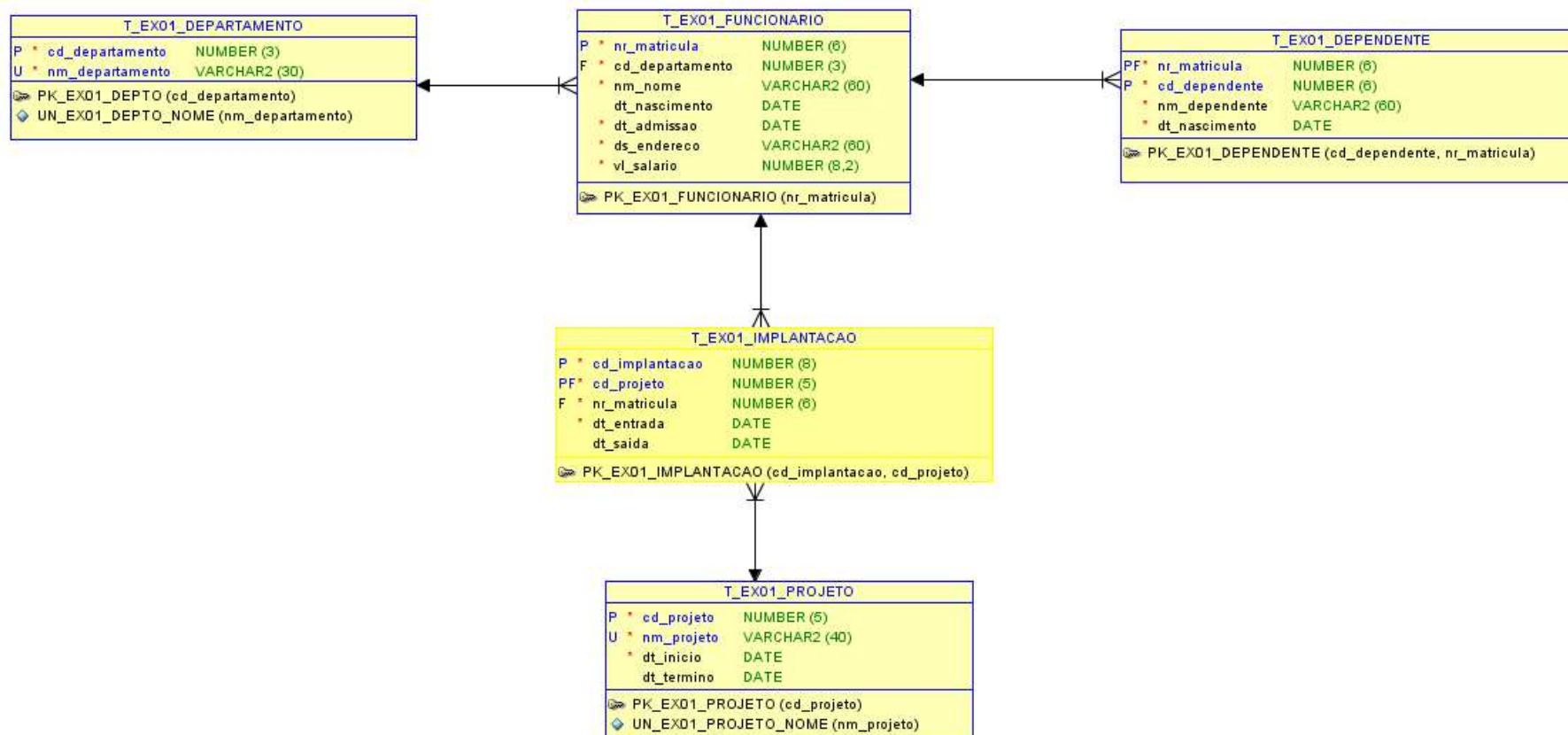




# Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

### Modelo Relacional ou Físico – IMPLANTAÇÃO DE PROJETOS



# Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

FIAP



### UTILIZANDO FUNÇÕES

**Funções de uma única linha:** Opera sobre uma linha por vez e retorna uma linha de saída para cada linha.

Existem cinco tipos de funções de uma única linha. Neste material abordaremos as 3 primeiras.

FUNÇÃO	DESCRIÇÃO
CARACTERE	Manipulam strings de caracteres.
NUMÉRICAS	Efetuem cálculos.
CONVERSÃO	Convertem um valor de um tipo de banco de dados para outro.
DATA	Processam datas e horas.
EXPRESSÃO REGULAR	Utilizam expressões regulares para procurar dados. Introduzidas no Oracle 10g e ampliadas no 11g.

---

# **| Comandos SQL**

## **Consulta dos Dados utilizando a linguagem SQL**

FIAP



# **FUNÇÕES DE CARACTERE**

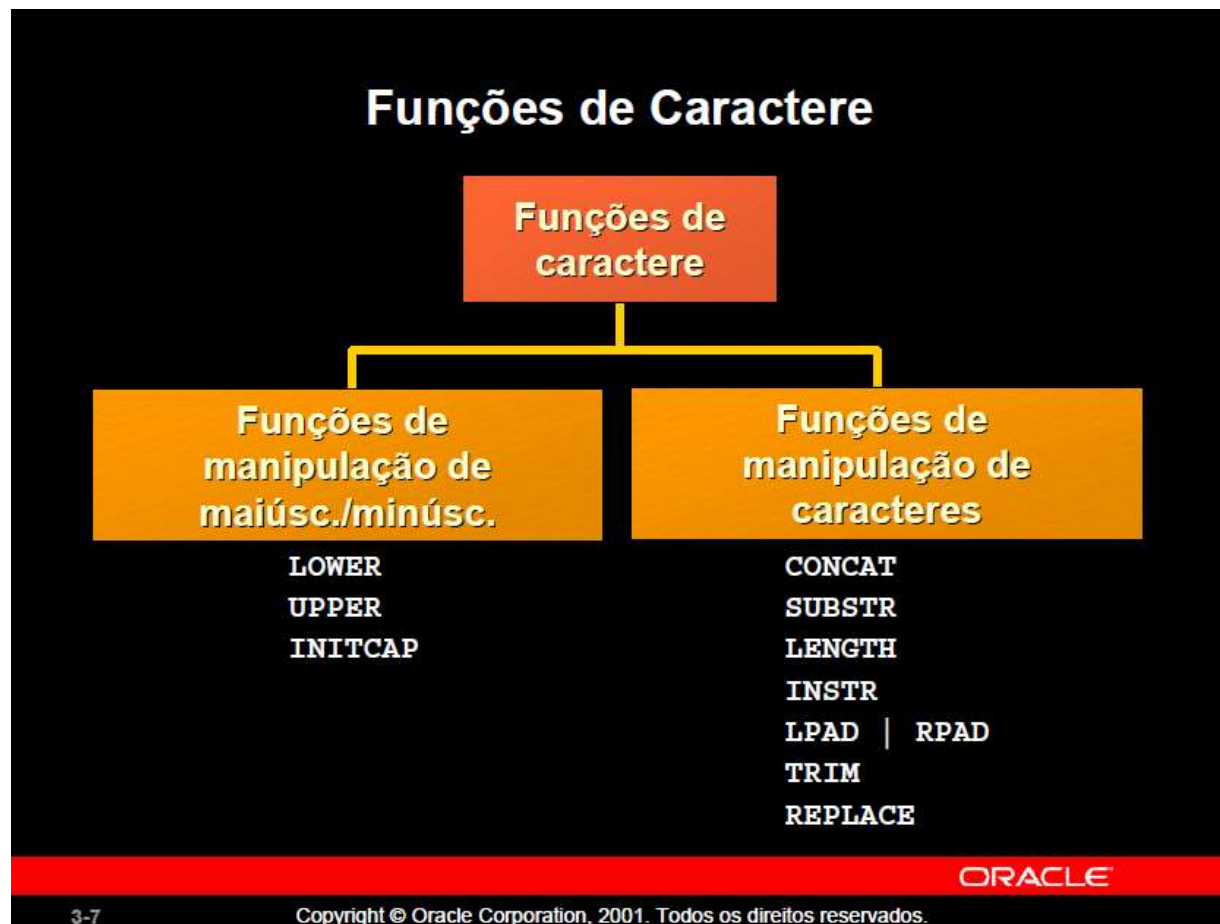


# | Comandos SQL

FIAP

Consulta dos Dados utilizando a linguagem SQL

## FUNÇÕES DE CARACTERE



# | Comandos SQL

FIAP

Consulta dos Dados utilizando a linguagem SQL

**FUNÇÕES DE CARACTERE – CONVERSÃO DE ESCRITA**



FUNÇÃO	DESCRIÇÃO
<b>LOWER(x)</b>	Converte as letras de x para maiúsculas e retorna a nova string.
<b>UPPER(x)</b>	Converte as letras de x para minúsculas e retorna a nova string.
<b>INITCAP(x)</b>	Converte a letra inicial de cada palavra da string x em maiúsculas e retorna a nova string.

# | Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

### FUNÇÕES DE CARACTERE – CONVERSÃO DE ESCRITA



```
SELECT LOWER(F.NM_NOME)      "MINUSCULAS (LOWER) " ,  
       UPPER(F.NM_NOME)      "MAIUSCULAS (UPPER) " ,  
       INITCAP(F.NM_NOME)     "MAIUSCULA/MINUSCULA (INITCAP) "  
  
FROM T_EX01_FUNCIONARIO F;
```

	MINUSCULAS (LOWER)	MAIUSCULAS (UPPER)	MAIUSCULA/MINUSCULA (INITCAP)
1	ana maria	ANA MARIA	Ana Maria
2	rosa maria	ROSA MARIA	Rosa Maria
3	antonia camargo	ANTONIA CAMARGO	Antonia Camargo
4	joão da silva	JOÃO DA SILVA	João Da Silva
5	joaquim xavier	JOAQUIM XAVIER	Joaquim Xavier
6	josé maria	JOSÉ MARIA	José Maria
7	josé lisboa	JOSÉ LISBOA	José Lisboa

# | Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

## FUNÇÕES DE CARACTERE – MANIPULAÇÃO DE CARACTERES



FUNÇÃO	DESCRIÇÃO
ASCII(x)	Retorna o código ASCII do caractere x.
CHR(x)	Retorna o caractere referente ao código ASCII informado em x.

```
SELECT ASCII('R') , CHR(82) FROM DUAL;
```

	ASCII('R')	CHR(82)
1	82	R



# | Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

## FUNÇÕES DE CARACTERE – MANIPULAÇÃO DE CARACTERES



FUNÇÃO	DESCRIÇÃO
CONCAT(x,y)	Anexa o conteúdo de y a x e retorna a nova string.

```
SELECT CONCAT(CONCAT(F.NR_MATRICULA, '- '), F.NM_NOME)  
FROM T_EX01_FUNCIONARIO F;
```

	CONCAT(CONCAT(F.NR_MATRICULA, '- '), F.NM_NOME)
1	1234-ANA MARIA
2	1235-ROSA MARIA
3	3245-ANTONIA CAMARGO
4	2233-JOÃO DA SILVA
5	7866-JOAQUIM XAVIER
6	9876-JOSÉ MARIA
7	9899-JOSÉ LISBOA



# | Comandos SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

### FUNÇÕES DE CARACTERE – MANIPULAÇÃO DE CARACTERES



FUNÇÃO	DESCRIÇÃO
<b>INSTR(x, LOCALIZAR_STRING [, INÍCIO] [,OCORRÊNCIA])</b>	<p>Procura “localizar_String” em x e retorna a posição em que ocorre.</p> <p>Pode ser informada a posição início (opcional) para iniciar a busca.</p> <p>Pode ser fornecida a ocorrência (opcional), que indica qual ocorrência de “localizar_String” deve ser retornada.</p>

# Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

## FUNÇÕES DE CARACTERE – MANIPULAÇÃO DE CARACTERES



```
SELECT F.NM_NOME  
       INSTR(F.NM_NOME, 'A')  
       INSTR(F.NM_NOME, 'A', 3)  
       INSTR(F.NM_NOME, 'A', 3, 2)  
FROM T_EX01_FUNCIONARIO F;
```

	NM_NOME	INSTR(F.NM_NOME,'A')	INSTR(F.NM_NOME,'A',3)	INSTR(F.NM_NOME,'A',3,2)
1	ANA MARIA	1	3	6
2	ROSA MARIA	4	4	7
3	ANTONIA CAMARGO	1	7	10
4	JOÃO DA SILVA	7	7	13
5	JOAQUIM XAVIER	3	3	10
6	JOSÉ MARIA	7	7	10
7	JOSÉ LISBOA	11	11	0

# Comandos SQL

FIAP

Consulta dos Dados utilizando a linguagem SQL

FUNÇÕES DE CARACTERE – MANIPULAÇÃO DE CARACTERES



FUNÇÃO	DESCRIÇÃO
LENGTH(x)	Retorna o tamanho em caracteres de x.

```
SELECT F.NM_NOME , LENGTH(F.NM_NOME)
FROM T_EX01_FUNCIONARIO F;
```

	NM_NOME	LENGTH(F.NM_NOME)
1	ANA MARIA	9
2	ROSA MARIA	10
3	ANTONIA CAMARGO	15
4	JOÃO DA SILVA	13
5	JOAQUIM XAVIER	14
6	JOSÉ MARIA	10
7	JOSÉ LISBOA	11



## Consulta dos Dados utilizando a linguagem SQL

### FUNÇÕES DE CARACTERE – MANIPULAÇÃO DE CARACTERES



FUNÇÃO	DESCRIÇÃO
<b>LPAD (x, largura [, string_preenchimento])</b>	<p>Preenche x com espaços a esquerda para que o comprimento total da string tenha até n caracteres largura. Pode ser fornecida uma string para o preenchimento a esquerda de x.</p> <p>Quando não fornecida a string de preenchimento é utilizado o padrão espaços em branco.</p>
<b>RPAD (x, largura [, string_preenchimento])</b>	<p>Preenche x com espaços a direita para que o comprimento total da string tenha até n caracteres largura. Pode ser fornecida uma string para o preenchimento a direita de x.</p> <p>Quando não fornecida a string de preenchimento é utilizado o padrão espaços em branco.</p>

# | Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

## FUNÇÕES DE CARACTERE – MANIPULAÇÃO DE CARACTERES



```
SELECT F.NM_NOME  
       F.VL_SALARIO  
       RPAD (F.NM_NOME, 60)  
       RPAD (F.NM_NOME, 60, ' . ' )  
       LPAD (VL_SALARIO, 10, '*' )  
  
FROM T_EX01_FUNCIONARIO F;
```

N	NM_NOME	N	VL_SALARIO	N	RPAD(F.NM_NOME,60)	N	RPAD(F.NM_NOME,60,' . ')	N	LPAD(VL_SALARIO,10,'*')
1	ANA MARIA		1234,56		ANA MARIA		ANA MARIA.....		***1234,56
2	ROSA MARIA		2345		ROSA MARIA		ROSA MARIA.....		*****2345
3	ANTONIA CAMARGO		7654,99		ANTONIA CAMARGO		ANTONIA CAMARGO.....		***7654,99
4	JOÃO DA SILVA		3452,12		JOÃO DA SILVA		JOÃO DA SILVA.....		***3452,12
5	JOAQUIM XAVIER		980		JOAQUIM XAVIER		JOAQUIM XAVIER.....		*****980
6	JOSÉ MARIA		1789		JOSÉ MARIA		JOSÉ MARIA.....		*****1789
7	JOSÉ LISBOA		1780		JOSÉ LISBOA		JOSÉ LISBOA.....		*****1780

# | Comandos SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

## FUNÇÕES DE CARACTERE – MANIPULAÇÃO DE CARACTERES



FUNÇÃO	DESCRIÇÃO
<b>LTRIM (x [, string corte] )</b>	Corta caracteres a esquerda de x. String_corte (opcional) especifica os caracteres a serem cortados. Se nenhuma string for fornecida, por padrão são cortados os espaços em branco.
<b>RTRIM (x [, string corte] )</b>	Corta caracteres a direita de x. String_corte (opcional) especifica os caracteres a serem cortados. Se nenhuma string for fornecida, por padrão são cortados os espaços em branco.
<b>TRIM ( [ car_corte FROM] x)</b>	Corta caracteres a direita e a esquerda de x. String_corte (opcional) especifica os caracteres a serem cortados. Se nenhuma string for fornecida, por padrão são cortados os espaços em branco.

# Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

## FUNÇÕES DE CARACTERE – MANIPULAÇÃO DE CARACTERES



```
SELECT F.NM_NOME  
      ,  
      LTRIM(F.NM_NOME) ,  
      LTRIM(F.NM_NOME, 'A') ,  
      RTRIM(F.NM_NOME) ,  
      RTRIM(F.NM_NOME, 'O') ,  
      TRIM(F.NM_NOME) ,  
      TRIM('A' FROM F.NM_NOME)  
  
FROM T_EX01_FUNCIONARIO F;
```

ID	NM_NOME	LTRIM(F.NM_NOME)	LTRIM(F.NM_NOME,'A')	RTRIM(F.NM_NOME)	RTRIM(F.NM_NOME,'O')	TRIM(F.NM_NOME)	TRIM('A'FROMF.NM_NOME)
1	ANA MARIA	ANA MARIA	NA MARIA	ANA MARIA	ANA MARIA	ANA MARIA	NA MARI
2	ROSA MARIA	ROSA MARIA	ROSA MARIA	ROSA MARIA	ROSA MARIA	ROSA MARIA	ROSA MARI
3	ANTONIA CAMARGO	ANTONIA CAMARGO	NTONIA CAMARGO	ANTONIA CAMARGO	ANTONIA CAMARG	ANTONIA CAMARGO	NTONIA CAMARGO
4	JOÃO DA SILVA	JOÃO DA SILVA	JOÃO DA SILVA	JOÃO DA SILVA	JOÃO DA SILVA	JOÃO DA SILVA	JOÃO DA SILV
5	JOAQUIM XAVIER	JOAQUIM XAVIER	JOAQUIM XAVIER	JOAQUIM XAVIER	JOAQUIM XAVIER	JOAQUIM XAVIER	JOAQUIM XAVIER
6	JOSÉ MARIA	JOSÉ MARIA	JOSÉ MARIA	JOSÉ MARIA	JOSÉ MARIA	JOSÉ MARIA	JOSÉ MARI
7	JOSÉ LISBOA	JOSÉ LISBOA	JOSÉ LISBOA	JOSÉ LISBOA	JOSÉ LISBOA	JOSÉ LISBOA	JOSÉ LISBO

# Comandos SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

## FUNÇÕES DE CARACTERE – MANIPULAÇÃO DE CARACTERES



FUNÇÃO	DESCRIÇÃO
NVL (x, valor)	Retorna valor, caso x seja nulo; caso contrário, x será retornado.
NVL2 (x, valor1, valor2)	Retorna valor 1 se x não é nulo; caso contrário, valor 2 é retornado.

```
SELECT F.DT_NASCIMENTO ,  
       NVL (F.DT_NASCIMENTO , '01/01/1900' ) ,  
       NVL2 (F.DT_NASCIMENTO , SYSDATE , '01/01/1900' )  
FROM T_EX01_FUNCIONARIO F;
```



	DT_NASCIMENTO	NVL(F.DT_NASCIMENTO,'01/01/1900')	NVL2(F.DT_NASCIMENTO,SYSDATE,'01/01/1900')
1	10/02/1986 00:00:00	10/02/1986 00:00:00	29/09/2013 20:30:57
2	15/08/1970 00:00:00	15/08/1970 00:00:00	29/09/2013 20:30:57
3	18/07/1986 00:00:00	18/07/1986 00:00:00	29/09/2013 20:30:57
4	(null)	01/01/1900 00:00:00	01/01/1900 00:00:00
5	21/04/1987 00:00:00	21/04/1987 00:00:00	29/09/2013 20:30:57
6	08/09/1998 00:00:00	08/09/1998 00:00:00	29/09/2013 20:30:57
7	07/10/1996 00:00:00	07/10/1996 00:00:00	29/09/2013 20:30:57



# Comandos SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

### FUNÇÕES DE CARACTERE – MANIPULAÇÃO DE CARACTERES



FUNÇÃO	DESCRIÇÃO
REPLACE(x , string_busca, string_substituta)	Procura “string_busca” em x e substitui por “string_substituta”.

```
SELECT F.NM_NOME  
  
      REPLACE (F.NM_NOME , 'A' , '*' )  
  
FROM T_EX01_FUNCIONARIO F;
```



Nº	NM_NOME	Nº	REPLACE(F.NM_NOME,'A','*')
1	ANA MARIA		*N* M*RI*
2	ROSA MARIA		ROS* M*RI*
3	ANTONIA CAMARGO		*NTONI* C*M*RG
4	JOÃO DA SILVA		JOÃO D* SILV*
5	JOAQUIM XAVIER		JO*QUIM X*VIER
6	JOSÉ MARIA		JOSÉ M*RI*
7	JOSÉ LISBOA		JOSÉ LISBO*

# Comandos SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

### FUNÇÕES DE CARACTERE – MANIPULAÇÃO DE CARACTERES



FUNÇÃO	DESCRIÇÃO
<b>SUBSTR(x, início [, comprimento] )</b>	Retorna uma substring de x que começa na posição especificada por início. O comprimento pode ser fornecido opcionalmente.

```
SELECT F.NM_NOME  
       ,  
       SUBSTR (F.NM_NOME , 3)  
       ,  
       SUBSTR (F.NM_NOME , 3 , 5)
```



```
FROM T_EX01_FUNCIONARIO F;
```

	NM_NOME	SUBSTR(F.NM_NOME,3)	SUBSTR(F.NM_NOME,3,5)
1	ANA MARIA	A MARIA	A MAR
2	ROSA MARIA	SA MARIA	SA MA
3	ANTONIA CAMARGO	TONIA CAMARGO	TONIA
4	JOÃO DA SILVA	ÃO DA SILVA	ÃO DA
5	JOAQUIM XAVIER	AQUIM XAVIER	AQUIM
6	JOSÉ MARIA	SÉ MARIA	SÉ MA
7	JOSÉ LISBOA	SÉ LISBOA	SÉ LI

---

# **| Comandos SQL**

## **Consulta dos Dados utilizando a linguagem SQL**

FIAP



# **FUNÇÕES NUMÉRICAS**



# | Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

### FUNÇÕES NUMÉRICAS



Utilizadas para efetuar cálculos. Alguns exemplos.

FUNÇÃO	DESCRIÇÃO
<b>ROUND (x , [y])</b>	Retorna o resultado do arredondamento de x com y casas decimais (opcional). Se y omitido, arredonda em 0 casa decimal. Se y for negativo, x será arredondado à esquerda do ponto decimal.
<b>MOD ( x, y)</b>	Retorna o resto, quando x é dividido por y.
<b>TRUNC (x, [ y])</b>	Retorna o resultado do truncamento de x com y casas decimais opcionais. Se y omitido, arredonda em 0 casa decimal. Se y for negativo, x será arredondado à esquerda do ponto decimal.
<b>ABS (x)</b>	Retorna o valor absoluto de x. Ignora o sinal (+ ou - ).
<b>SQRT (x)</b>	Retorna a raiz quadrada de x.

# | Comandos SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

### FUNÇÕES NUMÉRICAS



```
SELECT F.VL_SALARIO ,  
       ROUND (F.VL_SALARIO) ,  
       ROUND (F.VL_SALARIO,1) ,  
       TRUNC (F.VL_SALARIO) ,  
       TRUNC (F.VL_SALARIO,1) ,  
       MOD (F.VL_SALARIO,3) ,  
       ABS (F.VL_SALARIO) ,  
       SQRT (F.VL_SALARIO)  
FROM T_EX01_FUNCIONARIO F;
```

# Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

### FUNÇÕES NUMÉRICAS



	VL_SALARIO	ROUND(F.VL_SALARIO)	ROUND(F.VL_SALARIO,1)	TRUNC(F.VL_SALARIO)	TRUNC(F.VL_SALARIO,1)	MOD(F.VL_SALARIO,3)
1	1234,56	1235	1234,6	1234	1234,5	1,56
2	2345	2345	2345	2345	2345	2
3	7654,99	7655	7655	7654	7654,9	1,99
4	3452,12	3452	3452,1	3452	3452,1	2,12
5	980	980	980	980	980	2
6	1789	1789	1789	1789	1789	1
7	1780	1780	1780	1780	1780	1

ABS(F.VL_SALARIO)	SQRT(F.VL_SALARIO)
1234,56	35,13630600959639866393338464041805575975
2345	48,42520005121300477207350865061554758954
7654,99	87,49279970374705068436597081010308845903
3452,12	58,75474448927507675473744951255150979614
980	31,30495168499705574972843136223786729616
1789	42,29657196511320112162885521810233030698
1780	42,19004621945797299958144258618469535728

---

# **| Comandos SQL**

## **Consulta dos Dados utilizando a linguagem SQL**

FIAP



# **FUNÇÕES DE CONVERSÃO**



# | Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

### FUNÇÕES DE CONVERSÃO



Utilizadas para converter um valor de um tipo de dado para outro.

Alguns exemplos.

FUNÇÃO	DESCRIÇÃO
<b>TO_CHAR (x, [, formato])</b>	Converte x em uma string VARCHAR2. Pode ser fornecido um formato (opcional), indicando o formato de x.
<b>TO_NUMBER(x, [, formato])</b>	Converte x em um valor NUMBER.
<b>TO_DATE(x, [, formato])</b>	Converte x em um valor DATE.



# Comandos SQL



FIAP

Consulta dos Dados utilizando a linguagem SQL

FUNÇÕES DE CONVERSÃO (TO\_CHAR) - EXEMPLOS



```
SELECT TO_CHAR(F.VL_SALARIO, 'L9999999.99') "SIMBOLO MOEDA LOCAL" ,  
       TO_CHAR(F.NR_MATRICULA, '0099999')  
FROM T_EX01_FUNCIONARIO F;
```

	 SIMBOLO MOEDA LOCAL	 TO_CHAR(F.NR_MATRICULA,'0099999')
1	R\$1234.56	0001234
2	R\$2345.00	0001235
3	R\$7654.99	0003245
4	R\$3452.12	0002233
5	R\$980.00	0007866
6	R\$1789.00	0009876
7	R\$1780.00	0009899



# Comandos SQL

FIAP

Consulta dos Dados utilizando a linguagem SQL

FUNÇÕES DE CONVERSÃO (TO\_CHAR) - EXEMPLOS



```
SELECT TO_CHAR(12345678, '0099999999') ,  
       TO_CHAR(1234.56, 'C99,999.99') "SIMBOLO MOEDA ISO" ,  
       TO_CHAR(-120.30, '99,999.99')  
  
FROM DUAL;
```



	TO_CHAR(12345678,'0099999999')	SIMBOLO MOEDA ISO	TO_CHAR(-120.30,'99,999.99')
1	0012345678	BRL1,234.56	-120.30

# | Comandos SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

## FUNÇÕES DE CONVERSÃO (TO\_CHAR) - FORMATAÇÃO



Tabela 4-4 Parâmetros de formatação numérica

Parâmetro	Exemplos de formato	Descrição
9	999	Retorna dígitos nas posições especificadas, com um sinal negativo à esquerda se o número é negativo.
0	0999 9990	0999: Retorna um número com zeros à esquerda. 9990: Retorna um número com zeros à direita.
.	999.99	Retorna um ponto decimal na posição especificada.
,	9,999	Retorna uma vírgula na posição especificada.
\$	\$999	Retorna um cifrão à esquerda.
B	B9.99	Se a parte inteira de um número de ponto fixo é zero, retorna espaços para os zeros.
C	C999	Retorna o símbolo de moeda ISO na posição especificada. O símbolo vem do parâmetro de banco de dados NLS_ISO_CURRENCY definido pelo DBA.

# Comandos SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

## FUNÇÕES DE CONVERSÃO (TO\_CHAR) - FORMATAÇÃO



Tabela 4-4 Parâmetros de formatação numérica (continuação)

Parâmetro	Exemplos de formato	Descrição
D	9D99	Retorna o símbolo de ponto decimal na posição especificada. O símbolo vem do parâmetro de banco de dados NLS_NUMERIC_CHARACTER (o padrão é um caractere de ponto-final).
EEEE	9.99EEEE	Retorna o número usando a notação científica.
FM	FM90.9	Remove os espaços à esquerda e à direita do número.
G	9G999	Retorna o símbolo de separador de grupo na posição especificada. O símbolo vem do parâmetro de banco de dados NLS_NUMERIC_CHARACTER.
L	L999	Retorna o símbolo de moeda local na posição especificada. O símbolo vem do parâmetro de banco de dados NLS_CURRENCY.

# | Comandos SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

## FUNÇÕES DE CONVERSÃO (TO\_CHAR) - FORMATAÇÃO



Tabela 4-4 *Parâmetros de formatação numérica* (continuação)

Parâmetro	Exemplos de formato	Descrição
MI	999MI	Retorna um número negativo com um sinal de menos à direita. Retorna um número positivo com um espaço à direita.
PR	999PR	Retorna um número negativo entre sinais de menor e maior (< >). Retorna um número positivo com espaços à esquerda e à direita.
RN rn	RN rn	Retorna o número como algarismos romanos. RN retorna numerais maiúsculos; rn retorna numerais minúsculos. O número deve ser um valor inteiro entre 1 e 3999.
S	S999 999S	S999: Retorna um número negativo com um sinal de negativo à esquerda; retorna um número positivo com um sinal de positivo à esquerda. 999S: Retorna um número negativo com um sinal de negativo à direita; retorna um número positivo com um sinal de positivo à direita.

# | Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

## FUNÇÕES DE CONVERSÃO (TO\_CHAR) - FORMATAÇÃO



Tabela 4-4 Parâmetros de formatação numérica (continuação)

Parâmetro	Exemplos de formato	Descrição
TM	TM	Retorna o número usando a quantidade mínima de caracteres. O padrão é TM9, que retorna o número usando notação fixa, a não ser que o número de caracteres seja maior do que 64. Se for maior do que 64, o número será retornado usando notação científica.
U	U999	Retorna o símbolo de moeda duplo (o Euro, por exemplo) na posição especificada. O símbolo vem do parâmetro de banco de dados NLS_DUAL_CURRENCY.
V	99V99	Retorna o número multiplicado por $10^x$ , onde $x$ é o número de caracteres 9 após a letra V. Se necessário, o número é arredondado.
X	XXXX	Retorna o número em hexadecimal. Se o número não é um valor inteiro, ele é arredondado para um inteiro.

# | Comandos SQL

FIAP




Consulta dos Dados utilizando a linguagem SQL

FUNÇÕES DE CONVERSÃO (TO\_NUMBER) - EXEMPLOS



```
SELECT TO_NUMBER('103') ,  
       TO_NUMBER('103') + 30 ,  
       TO_NUMBER('- $12,345.99' , '$99,999.99')  
FROM DUAL;
```



	 TO_NUMBER('103')	 TO_NUMBER('103')+30	 TO_NUMBER('- \$12,345.99' , '\$99,999.99')
1	103	133	-12345,99

# Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

### FUNÇÕES DE CONVERSÃO – CAST( x as TIPO)

Utilizada para converter x em um tipo de banco de dados compatível especificado por tipo. Conversões válidas assinaladas com X.

De	Para						
	BINARY_FLOAT BINARY_DOUBLE	CHAR VARCHAR2z	NUMBER	DATE TIMESTAMP INTERVAL	RAW	ROWID UROWID	NCHAR NVARCHAR2
BINARY_FLOAT BINARY_DOUBLE	X	X	X				X
CHAR VARCHAR2	X	X	X	X	X	X	
NUMBER	X	X	X				X
DATE TIMESTAMP INTERVAL		X		X			
RAW		X			X		
ROWID UROWID		X				X	
NCHAR NVARCHAR2	X		X	X	X	X	X

Nota: RAW – dados binários de comprimento variável, de 1 byte a 2 KB.

FIAP





# | Comandos SQL

FIAP

Consulta dos Dados utilizando a linguagem SQL

## FUNÇÕES DE CONVERSÃO (CAST) - EXEMPLOS



-- CONVERTER VALORES LITERAIS EM TIPOS ESPECÍFICOS

SELECT

CAST(12345.67 AS VARCHAR2(10)) ,

CAST('29/09/2013' AS DATE) ,

CAST(12345.678 AS NUMBER(10,2))

FROM DUAL;



	CAST(12345.67ASVARCHAR2(10))	CAST('29/09/2013'ASDATE)	CAST(12345.678ASNUMBER(10,2))
1	12345,67	29/09/2013 00:00:00	12345,68

# | Comandos SQL

FIAP

Consulta dos Dados utilizando a linguagem SQL

## FUNÇÕES DE CONVERSÃO (CAST) - EXEMPLOS



-- CONVERTER VALORES DE COLUNA DE UM TIPO PARA OUTRO

SELECT

CAST (F.VL\_SALARIO AS VARCHAR2(10) ) ,

CAST (F.VL\_SALARIO + 1000 AS NUMBER (8,2))

FROM T\_EX01\_FUNCIONARIO F;



R	2	CAST(F.VL_SALARIOASVARCHAR2(10))	R	2	CAST(F.VL_SALARIO+1000ASNUMBER(8,2))
1		1234,56			2234,56
2		2345			3345
3		7654,99			8654,99
4		3452,12			4452,12
5		980			1980
6		1789			2789
7		1780			2780

---

# **| Comandos SQL**

## **Consulta dos Dados utilizando a linguagem SQL**

FIAP



# **FUNÇÕES DE DATA**



# Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

FIAP



### FUNÇÕES DE DATA

As funções abordadas seguem a sintaxe padronizada pela Oracle.

As funções de data, permitem manipular data e hora no Oracle.

São algumas das funções que permitem manipular datas:

FUNÇÃO	DESCRIÇÃO
<b>SYSDATE</b>	Permite recuperar a data do sistema.
<b>ADD_MONTHS()</b>	Permite recuperar uma data, incrementada de <u>n</u> meses. Assim podemos subtraís ou somar meses a uma data.
<b>LASTDAY()</b>	Permite recuperar o último dia do mês, da data informada.
<b>MONTHS_BETWEEN()</b>	Permite recuperar o número de meses entre duas datas.
<b>NEXT_DAY()</b>	Permite recuperar o próximo dia da semana, a partir de da data informada.

## Comandos SQL

Consulta dos Dados utilizando a linguagem SQL

FUNÇÕES DE DATA - SYSDATE

FIAP



-- EXEMPLO - SYSDATE

```
SELECT SYSDATE FROM DUAL;
```

SYSDATE
17/09/2017 07:36:24



# Comandos SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

### FUNÇÕES DE DATA - ADD\_MONTHS()



No exemplo **ADD\_MONTHS(SYSDATE, 1)**, será recuperada a data do sistema, incrementada de um mês, ou seja, será adicionado um mês a data informada.

No exemplo **ADD\_MONTHS(SYSDATE, -11)**, será recuperada a data do sistema, incrementada de um mês negativo, ou seja, será subtraído um mês a data informada.

-- EXEMPLO - FUNÇÕES DE DATA - ADD\_MONTHS()

```
SELECT  SYSDATE                "DATA ATUAL",  
        ADD_MONTHS(SYSDATE, 1) "PRÓXIMO MÊS",  
        ADD_MONTHS(SYSDATE, -1) "MÊS ANTERIOR"  
FROM    DUAL;
```

DATA ATUAL	PRÓXIMO MÊS	MÊS ANTERIOR
17/09/2017 07:38:26	17/10/2017 07:38:26	17/08/2017 07:38:26



# | Comandos SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

### FUNÇÕES DE DATA - ADD\_MONTHS()



No exemplo **ADD\_MONTHS(DT\_ADMISSAO, 1)**, será recuperada a data do de  
admissão do funcionário, incrementada de um mês, ou seja, será adicionado um  
mês a data informada.

-- EXEMPLO - FUNÇÕES DE DATA - ADD\_MONTHS()

```
SELECT DT_ADMISSAO ,  
       ADD_MONTHS (DT_ADMISSAO,1)  
FROM T_SIP_FUNCIONARIO;
```

DT_ADMISSAO	ADD_MONTHS(DT_ADMISSAO,1)
09/08/2010 12:00:00	09/09/2010 12:00:00
07/06/2005 12:00:00	07/07/2005 12:00:00
11/02/2003 12:00:00	11/03/2003 12:00:00
08/09/2011 12:00:00	08/10/2011 12:00:00
05/04/2010 12:00:00	05/05/2010 12:00:00
17/03/2009 12:00:00	17/04/2009 12:00:00
17/03/2009 12:00:00	17/04/2009 12:00:00



# | Comandos SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

### FUNÇÕES DE DATA – LAST\_DAY()



O comando SELECT, através das função de data LAST\_DAY(), a partir do exemplo LAST\_DAY(SYSDATE), será recuperada a data do último dia do mês, referente a data do sistema informada.

```
-- EXEMPLO - FUNÇÕES DE DATA - LAST_DAY()  
SELECT SYSDATE "DATA ATUAL",  
       LAST_DAY(SYSDATE) "DATA COM ÚLTIMO DIA DO MÊS"  
FROM DUAL;
```

DATA ATUAL	DATA COM ÚLTIMO DIA DO MÊS
17/09/2017 07:41:50	30/09/2017 07:41:50





# | Comandos SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

### FUNÇÕES DE DATA – LAST\_DAY()



No exemplo **LAST\_DAY(DT\_ADMISSAO)**, será recuperada a data do último dia do mês, referente a data de admissão do funcionário.

-- EXEMPLO - FUNÇÕES DE DATA - LAST\_DAY()

```
SELECT DT_ADMISSAO ,  
       LAST_DAY(DT_ADMISSAO)  
FROM T_SIP_FUNCIONARIO;
```

DT_ADMISSAO	LAST_DAY(DT_ADMISSAO)
09/08/2010 12:00:00	31/08/2010 12:00:00
07/06/2005 12:00:00	30/06/2005 12:00:00
11/02/2003 12:00:00	28/02/2003 12:00:00
08/09/2011 12:00:00	30/09/2011 12:00:00
05/04/2010 12:00:00	30/04/2010 12:00:00
17/03/2009 12:00:00	31/03/2009 12:00:00
17/03/2009 12:00:00	31/03/2009 12:00:00



## | Comandos SQL

FIAP

### Consulta dos Dados utilizando a linguagem SQL

#### FUNÇÕES DE DATA – MONTHS\_BETWEEN()



No exemplo **MONTHS\_BETWEEN(SYSDATE, DT\_ADMISSAO)**, será recuperada a diferença em número de meses, entre a data do sistema e a data de admissão do funcionário.

No exemplo **TRUNC(MONTHS\_BETWEEN(SYSDATE, DT\_ADMISSAO)) || ' MESES'**, será recuperada a diferença em número de meses, entre a data do sistema e a data de admissão do funcionário. A função TRUNC, irá truncar, ou seja, cortar o resultado obtido na parte inteira do número, o operador de concatenação “||” irá concatenar, unir, a palavra meses ao resultado do obtido com a execução das funções.



# | Comandos SQL

FIAP

## Consulta dos Dados utilizando a linguagem SQL

### FUNÇÕES DE DATA – MONTHS\_BETWEEN()



```
-- EXEMPLO - FUNÇÕES DE DATA - MONTHS_BETWEEN()  
  
SELECT DT_ADMISSAO ,  
       SYSDATE _____,  
       MONTHS_BETWEEN(SYSDATE, DT_ADMISSAO) ,  
       TRUNC(MONTHS_BETWEEN(SYSDATE, DT_ADMISSAO)) || ' MESES'  
  
FROM T_SIP_FUNCIONARIO;
```

DT_ADMISSAO	SYSDATE	MONTHS_BETWEEN(SYSDATE,DT_ADMISSAO)	TRUNC(MONTHS_BETWEEN(SYSDATE,DT_ADMISSAO))  'MESES'
09/08/2010 12:00:00	17/09/2017 07:44:57	85,2846090949820788530465949820788530466	85 MESES
07/06/2005 12:00:00	17/09/2017 07:44:57	147,349125224014336917562724014336917563	147 MESES
11/02/2003 12:00:00	17/09/2017 07:44:57	175,22009296594982078853046594982078853	175 MESES
08/09/2011 12:00:00	17/09/2017 07:44:57	72,31686715949820788530465949820788530466	72 MESES
05/04/2010 12:00:00	17/09/2017 07:44:57	89,41364135304659498207885304659498207885	89 MESES
17/03/2009 12:00:00	17/09/2017 07:44:57		102 102 MESES
17/03/2009 12:00:00	17/09/2017 07:44:57		102 102 MESES



## Comandos SQL

### Consulta dos Dados utilizando a linguagem SQL

#### FUNÇÕES DE DATA – NEXT\_DAY()

O comando SELECT, através das função de data NEXT\_DAY(), retornará conforme exemplo **NEXT\_DAY**(SYSDATE, 1), a próxima data, referente ao dia da semana informado.

Os valores de 1 a 7, representam respectivamente os dias da semana de domingo a sábado.

FIAP



# | Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

### FUNÇÕES DE DATA – NEXT\_DAY()

FIAP



```
-- EXEMPLO - FUNÇÕES DE DATA - NEXT_DAY()  
  
SELECT SYSDATE ,  
       NEXT_DAY(SYSDATE, 1) "PRÓXIMO DOMINGO" ,  
       NEXT_DAY(SYSDATE, 2) "PRÓXIMA SEGUNDA" ,  
       NEXT_DAY(SYSDATE, 3) "PRÓXIMA TERÇA" ,  
       NEXT_DAY(SYSDATE, 4) "PRÓXIMA QUARTA" ,  
       NEXT_DAY(SYSDATE, 5) "PRÓXIMA QUINTA" ,  
       NEXT_DAY(SYSDATE, 6) "PRÓXIMA SEXTA" ,  
       NEXT_DAY(SYSDATE, 7) "PRÓXIMO SÁBADO"  
  
FROM DUAL;
```

SYSDATE	PRÓXIMO DOMINGO	PRÓXIMA SEGUNDA	PRÓXIMA TERÇA	PRÓXIMA QUARTA	PRÓXIMA QUINTA	PRÓXIMA SEXTA	PRÓXIMO SÁBADO
17/09/2017 07:46:45	24/09/2017 07:46:45	18/09/2017 07:46:45	19/09/2017 07:46:45	20/09/2017 07:46:45	21/09/2017 07:46:45	22/09/2017 07:46:45	23/09/2017 07:46:45



---

# **| Comandos SQL**

## **Consulta dos Dados utilizando a linguagem SQL**

FIAP



# **MAIS EXEMPLOS COM DATA**



# | Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

FIAP



### EXEMPLOS COM DATA

No exemplo **TO\_CHAR(DT\_ADMISSAO, 'YYYY')**, será recuperado o ano data de admissão do funcionário.

No exemplo **TO\_CHAR(DT\_ADMISSAO, 'MM/YYYY')**, será recuperado o mês e ano da data de admissão do funcionário.

No exemplo **TO\_CHAR(DT\_ADMISSAO, 'DD/MM')**, será recuperado o dia e mês da data de admissão do funcionário.

No exemplo **TO\_CHAR(DT\_ADMISSAO, 'DD " de " fmMONTH " de " YYYY, 'NLS\_DATE\_LANGUAGE=PORTUGUESE')**, será recuperado a data por extenso referente a data de admissão do funcionário.

Onde fmMONTH, é um modelo de formato disponível na linguagem SQL – ORACLE e, NLS\_DATE\_FORMAT, permite especificar o formato de data padrão , pi usado com as funções TO\_CHAR() e TO\_DATE().





# | Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

### EXEMPLOS COM DATA

FIAP



-- EXEMPLO - FUNÇÕES DE DATA

```
SELECT DT_ADMISSAO ,  
       TO_CHAR(DT_ADMISSAO, 'YYYY')      "ANO CONTRATAÇÃO" ,  
       TO_CHAR(DT_ADMISSAO, 'MM/YYYY')   "MÊS E ANO CONTRATAÇÃO" ,  
       TO_CHAR(DT_ADMISSAO, 'DD/MM')     "DIA E MÊS DA CONTRATAÇÃO" ,  
       TO_CHAR(DT_ADMISSAO, 'DD " de " fmMONTH " de " YYYY',  
       'NLS_DATE_LANGUAGE=PORTUGUESE') "DATA POR EXTENSO"  
  
FROM T_SIP_FUNCIONARIO;
```

DT_ADMISSAO	ANO CONTRATAÇÃO	MÊS E ANO CONTRATAÇÃO	DIA E MÊS DA CONTRATAÇÃO	DATA POR EXTENSO
09/08/2010	2010	08/2010	09/08	09 de AGOSTO de 2010
07/06/2005	2005	06/2005	07/06	07 de JUNHO de 2005
11/02/2003	2003	02/2003	11/02	11 de FEVEREIRO de 2003
08/09/2011	2011	09/2011	08/09	08 de SETEMBRO de 2011
05/04/2010	2010	04/2010	05/04	05 de ABRIL de 2010
17/03/2009	2009	03/2009	17/03	17 de MARÇO de 2009
17/03/2009	2009	03/2009	17/03	17 de MARÇO de 2009





# | Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

### EXEMPLOS COM DATA

FIAP



No exemplo **TO\_DATE(DT\_ADMISSAO , 'DD/MM/YYYY') - 5**, será recuperada a data de admissão do funcionário, subtraída de 5 dias.

No exemplo **TO\_DATE(SYSDATE, 'DD/MM/YYYY') - TO\_DATE(DT\_ADMISSAO, 'DD/MM/YYYY')**, será recuperada a diferença entre as duas datas, o resultado é apresentado em número de dias.



# Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

### EXEMPLOS COM DATA

FIAP



#### -- EXEMPLO - FUNÇÕES DE DATA

```
SELECT DT_ADMISSAO ,  
       TO_DATE(DT_ADMISSAO , 'DD/MM/YYYY') - 5 ,  
       TO_DATE(SYSDATE, 'DD/MM/YYYY') -  
           TO_DATE(DT_ADMISSAO, 'DD/MM/YYYY')  
FROM T_SIP_FUNCIONARIO;
```

DT_ADMISSAO	TO_DATE(DT_ADMISSAO,'DD/MM/YYYY')-5	TO_DATE(SYSDATE,'DD/MM/YYYY')-TO_DATE(DT_ADMISSAO,'DD/MM/YYYY')
09/08/2010	04/08/2010	2596
07/06/2005	02/06/2005	4485
11/02/2003	06/02/2003	5332
08/09/2011	03/09/2011	2201
05/04/2010	31/03/2010	2722
17/03/2009	12/03/2009	3106
17/03/2009	12/03/2009	3106



# | Comandos SQL

## Consulta dos Dados utilizando a linguagem SQL

### EXEMPLOS COM DATA

FIAP



No exemplo `TO_DATE(DT_ADMISSAO,'DD/MM/YYYY HH:MI:SS') + 6/24`, será recuperada a data de admissão do funcionário, somadas 6 horas. O 6/24, representa 6 horas das 24 horas do dia.

#### -- EXEMPLO - FUNÇÕES DE DATA

```
SELECT DT_ADMISSAO ,
```

```
TO_DATE(DT_ADMISSAO, 'DD/MM/YYYY HH:MI:SS') + 6/24
```

```
FROM T_SIP_FUNCIONARIO;
```

DT_ADMISSAO	TO_DATE(DT_ADMISSAO,'DD/MM/YYYYHH:MI:SS')+6/24
09/08/2010 12:00:00	09/08/2010 06:00:00
07/06/2005 12:00:00	07/06/2005 06:00:00
11/02/2003 12:00:00	11/02/2003 06:00:00
08/09/2011 12:00:00	08/09/2011 06:00:00
05/04/2010 12:00:00	05/04/2010 06:00:00
17/03/2009 12:00:00	17/03/2009 06:00:00
17/03/2009 12:00:00	17/03/2009 06:00:00



## REFERÊNCIAS



- MACHADO, Felipe Nery R. Banco de Dados - Projeto e Implementação. Érica, 2004.
- ELMASRI, R.; NAVATHE, S.B. Sistemas de Banco de Dados: Fundamentos e Aplicações. Pearson, 2005.
- PRICE, JASON, ORACLE DATABASE 11 g – SQL Domine SQL e PL-SQL no banco de Dados Oracle, Bookman, 2008.
- **Outros:**
- Manual Oficial Oracle – Introdução ao Oracle 9i (SQL) - **Oracle Corporation, 2000, 2001.**

**Copyright © 2017 Profa. Rita de Cássia Rodrigues**

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