

Oracle SQL

Elementos Gerais

- [Introdução](#)
- [Elementos Básicos](#)
 - [Tipos de Dados](#)
 - [Modelos de Formato](#)
 - [Valor NULL](#)
 - [Comentários](#)
 - [Objetos de Esquema de BD](#)
 - [Nomes de Objetos](#)
- [Operadores Não-Lógicos](#)
- [Expressões](#)
- [Condições](#)
- [Funções](#)
- [Parâmetros NLS](#)
- [Bibliografia](#)

SQL

- Significado:
 - Structured Query Language -- Linguagem de consulta estruturada.
- Criada:
 - Por Edgar Codd
 - Na IBM
 - Início anos 70
- Interesse:
 - Criar
 - Alterar
 - Consultar
 - Controlar acesso
- É um *standard*
 - Exemplo
 - Servidor de BD do DEI
 - Oracle 12.1
 - Compatível
 - Standard Core SQL:2011

} BD Relacionais (BDR)

Oracle Compliance To Core SQL:2011

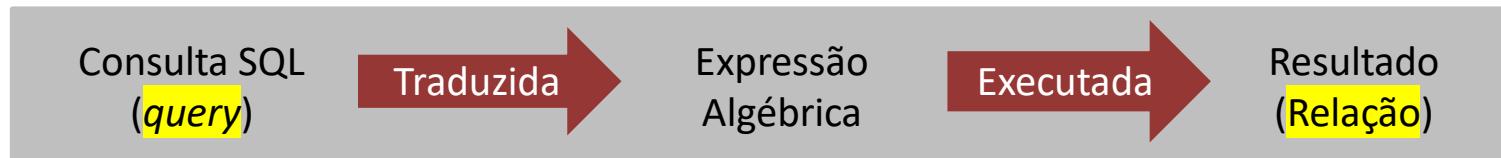
The ANSI and ISO SQL standards require conformance claims to state the type of conformance and the implemented facilities. The minimum claim of conformance is called Core SQL:2011 and is defined in Part 2, SQL/Foundation, and Part 11, SQL/Schemata, of the standard. The following products provide full or partial conformance with Core SQL:2011 as described in the tables that follow:

- Oracle Database server, release 12.1

■ SQL

- Baseada:

- Álgebra Relacional



- Linguagem declarativa

- Programador:
 - Exprime o resultado pretendido. -- não exprime procedimento para obter resultado.
 - Exemplo de *query*

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Query' tab is active, displaying the following SQL code:

```
11 SELECT nome      -- selecionar os nomes ...
12 FROM alunos     -- ... da tabela alunos ...
13 WHERE ano=1;    -- ... cujo ano é 1.
```

A callout points from the text 'Query' to the 'Query' tab. In the center, the 'Query Result' tab shows the output of the query:

NOME
1 Ana Silva
2 Rui Cunha

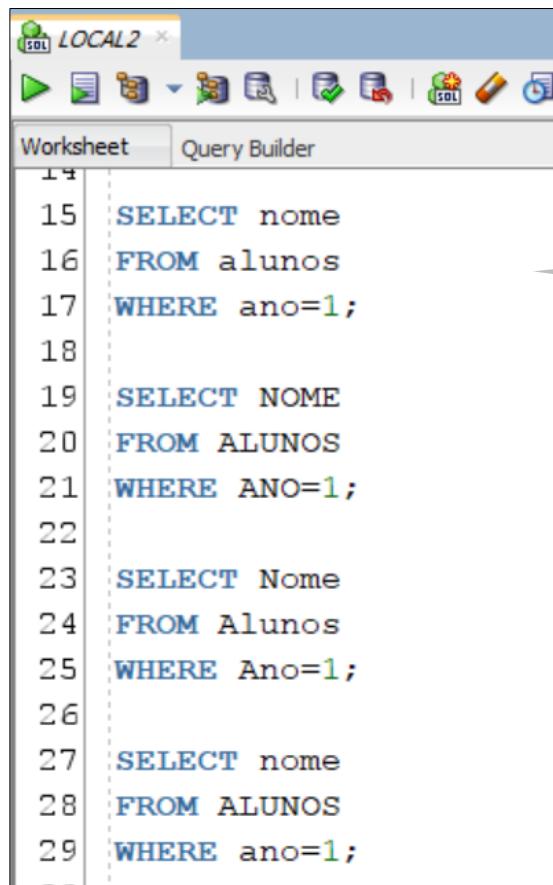
A callout points from the text 'Resultado da query' to this table. On the right, the 'ALUNOS' table is shown in the 'Data' tab:

NUMERO	NOME	ANO
1	1180123 Ana Silva	1
2	1180256 Joana Ferreira	2
3	1190189 Rui Cunha	1

A callout points from the text 'Tabela alunos' to this table. A callout also points from the text 'Comentário de linha em SQL' to the '--' comments in the query code.

- Oracle SQL

- Case-insensitive -- não distingue as letras maiúsculas das minúsculas.
 - Exemplo



The screenshot shows a SQL developer interface with a toolbar at the top and a worksheet tab selected. The code area displays five different SELECT statements, each with a line number on the left:

```
15 SELECT nome
16 FROM alunos
17 WHERE ano=1;
18
19 SELECT NOME
20 FROM ALUNOS
21 WHERE ANO=1;
22
23 SELECT Nome
24 FROM Alunos
25 WHERE Ano=1;
26
27 SELECT nome
28 FROM ALUNOS
29 WHERE ano=1;
```

Algumas formas alternativas
de um comando SELECT

■ SQL Fornece

- Elementos básicos:
 - Tipos de dados
 - Modelos de formato
 - Valor NULL
 - Comentários
 - Objetos de Esquema de BD
- Operadores
- Expressões
- Condições
- Funções
- Parâmetros NLS
- Comandos

Comandos DDL - Definição de Dados

- CREATE -- Criar
- DROP -- Eliminar
- ALTER -- Alterar

Comandos DML - Manipulação de Dados

- INSERT -- Inserir
- DELETE -- Eliminar
- UPDATE -- Atualizar
- SELECT -- Consultar

Comandos DCL - Controlo de Dados

- CREATE USER -- Definir perfis de utilizadores
- GRANT -- Fornecer acesso
- REVOKE -- Remover acesso

Elementos Básicos

Tipos de Dados

Modelos de Formato

Valor NULL

Comentários

Objetos de Esquema de BD

- [Introdução](#)
- Tipos de Dados
 - [Numéricos](#)
 - [Caráter](#)
 - [Data-Tempo e Intervalo](#)
- [Precedência](#)
- [Conversão](#)

Um Tipo de Dados

- Define dois conjuntos:
 - Conjunto de valores.
 - Conjunto de propriedades desses valores.
- Tem um nome
 - É palavra-reservada.
 - Exemplos
 - NUMBER -- valor numérico inteiro ou real.
 - CHAR -- *string* de comprimento fixo.
 - VARCHAR -- *string* de comprimento variável.
 - DATE -- data e tempo (sem frações do segundo e sem fuso horário).
 - TIMESTAMP -- data e tempo (com frações do segundo e com fuso horário).
 - INTERVAL YEAR TO MONTH -- intervalo de tempo em anos e meses.
 - INTERVAL DAY TO SECONDS -- intervalo de tempo em dias, horas, minutos e segundos.
- Interesse:
 - Permitir ao servidor de BD tratar de forma diferente ... valores de tipos de dados diferentes.
 - Exemplo
 - Permitir subtrair valores do tipo NUMBER ... mas não do tipo VARCHAR.
 - Definir o domínio das colunas (campos) das relações.

■ Um Dado duma BD

- Pertence:
 - A **algum** tipo de dados.
- Exceção:
 - **Valor NULL**
 - Representa:
 - **Valor desconhecido / inexistente.**
 - É um valor **especial**.
 - Incluído em **todos** os tipos de dados.

Daí:

- Impossível **inferir** o **tipo** de dados de valor **NULL**.

■ Tipos de Dados

- Sub dividem-se nas **categorias**:

- Numérico
- Caráter
- Data-Tempo e Intervalo

■ Cada Categoria de Tipos de Dados

- Subdivide-se nas **sub-categorias**:

- Tipos **Predefinidos** Oracle
- Tipos **Standard**
- Tipos **Definidos pelo Utilizador**

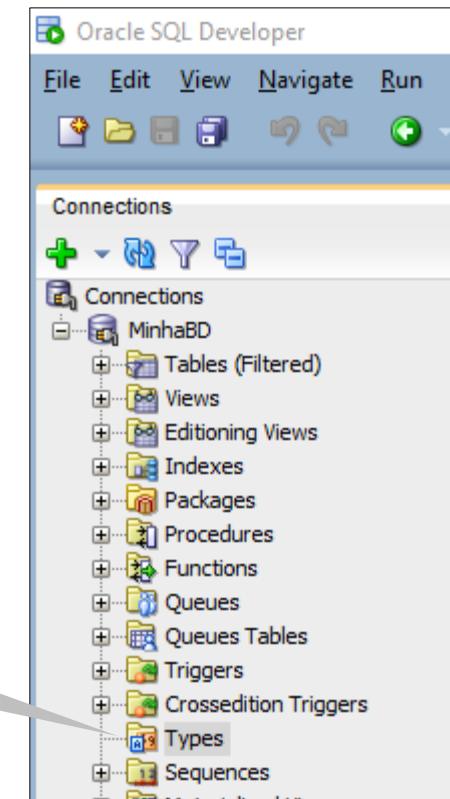
São **objetos** do esquema da “connection” (Types)

■ Cada Tipo de Dados é:

- **Escalar** -- cada valor do tipo é **atómico**.

ou

- **Não-Escalar** -- um valor é um **conjunto** de outros valores (escalares e/ou não-escalares).
-- designados coleção.



- **Tipos de Dados Oracle**

- Não existe o tipo booleano.
 - Alternativa
 - CHAR(1) -- *string* de caracteres de comprimento 1

Tipos de Dados Oracle

Tipos Numéricos

Tipos Caráter

Tipos Data-Tempo e Intervalo

Precedência

Conversão

- [Interesse](#)
- [Sub-Categorias](#)
 - [Tipos Predefinidos](#)
 - [Tipos Standard](#)
- [Valores Numéricos](#)
- [Operações Aritméticas](#)
- [Funções Numéricas](#)
- [Comparação](#)
- [Precedência](#)
- [Conversão de Tipos de Dados](#)

- Armazenar

- Números:
 - Inteiros
 - Reais:
 - Vírgula-fixa -- ex: 25.56
 - Vírgula-flutuante -- ex: 1e-25 ; 1e10
 - Positivos
 - Negativos
 - Zero
- Infinito
- NaN (“not a number”) -- resultado indefinido de uma operação.

- Sub-Categorias

- Tipos Predefinidos
- Tipos *Standard*

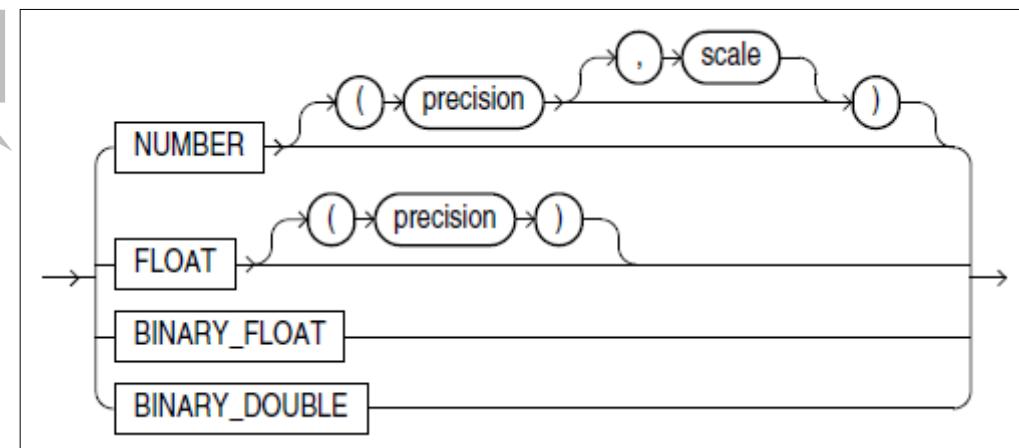
■ Tipos

- Com precisão decimal
 - NUMBER
 - representação exata de nºs decimais (dígitos ∈ [0, 9]).
 - nºs inteiros e reais, com vírgula-fixa ou vírgula-flutuante.
 - independente da plataforma de execução.
 - Com precisão binária
 - FLOAT
 - BINARY_FLOAT
 - BINARY_DOUBLE
 - nºs reais com vírgula-flutuante ; dependente da plataforma.
 - BINARY_DOUBLE tem o dobro da precisão do BINARY_FLOAT.
 - Oracle recomenda BINARY_FLOAT e BINARY_DOUBLE.

■ Subtipos NUMBER

- NUMBER(precision, scale)
- NUMBER(precision)
- NUMBER

Sintaxe



■ Subtipos FLOAT

- FLOAT(precision)
- FLOAT

- Precisão (p): máximo nº de dígitos:
 - Casas decimais incluídas.
 - Ponto decimal excluído.
- Escala (s): máximo nº de dígitos decimais.

■ Tipos de Precisão

- Precisão decimal
 - Representa de forma exata os n^{os} decimais.
 - Exemplo:
 - 0.1
- Precisão binária
 - Representa de forma aproximada os n^{os} decimais.
 - Exemplo:
 - 0.1 não pode ser representado de forma exata.
 - Vantagens:
 - Permite cálculos aritméticos mais rápidos.
 - Frequentemente reduz requisitos de armazenamento.

■ Interesses Específicos

- Para cálculos independentes da plataforma:
 - Subtipos NUMBER
- Para cálculos monetários:
 - NUMBER(p,s)
- Para cálculos científicos:
 - BINARY_FLOAT
 - BINARY_DOUBLE

■ Tipos Exatos

Tipo	Descrição	Exemplos
NUMBER(p, s)	<ul style="list-style-type: none">▪ Representação exata de nºs decimais reais:<ul style="list-style-type: none">▪ Números com vírgula-fixa.▪ NUMBER com precisão p e escala s:<ul style="list-style-type: none">▪ p: [1, 38] dígitos decimais (nºs inteiros $\in [0, 9]$).▪ s: [-84, 127] dígitos decimais.<ul style="list-style-type: none">▪ >0 = nº de casas decimais (direita da vírgula).▪ <0 = nº de dígitos à esquerda da vírgula, excluindo o das unidades.▪ Gama de valores:<ul style="list-style-type: none">▪ $[1 \times 10^{-130}, 1 \times 10^{128}]$▪ Números positivos e negativos.▪ Ocupa: [1, 22] bytes.▪ Se um valor:<ul style="list-style-type: none">▪ Exceder p: é gerado um erro.▪ Exceder s: é arredondado.	<p>Nota com 1 casa decimal [0-20]:</p> <ul style="list-style-type: none">▪ NUMBER(2,1) <p>Preço [0, 1000]:</p> <ul style="list-style-type: none">▪ NUMBER(4,2) <div data-bbox="979 619 1881 979" style="background-color: #e0e0e0; padding: 10px; border-radius: 10px; width: fit-content; margin-left: auto; margin-right: 0;"><ul style="list-style-type: none">▪ Serve para arredondar para o nº de dígitos especificados.▪ Exemplo (10,-2): arredonda para as centenas.<ul style="list-style-type: none">▪ 123.89: guardado como 100.▪ 149 \rightarrow 100 ; 150 \rightarrow 200▪ 220 \rightarrow 200 ; 250 \rightarrow 300▪ 49 \rightarrow 0 ; 50 \rightarrow 100</div>
NUMBER(p)	<ul style="list-style-type: none">▪ Representa números inteiros.▪ Equivalente a NUMBER(p, 0).	Ano: <ul style="list-style-type: none">▪ NUMBER(4)
NUMBER	<ul style="list-style-type: none">▪ Representação exata de nºs decimais reais:<ul style="list-style-type: none">▪ Números com vírgula-flutuante.▪ Gama de valores:<ul style="list-style-type: none">▪ A maior gama de NUMBER(p,s).	

Tipos Numéricos Predefinidos

■ Tipos Aproximados

Tipo	Descrição
FLOAT(p)	<ul style="list-style-type: none">▪ Subtipo de NUMBER.<ul style="list-style-type: none">▪ Representado internamente como NUMBER.▪ Precisão p:<ul style="list-style-type: none">▪ [1, 126] dígitos binários \approx [1,38] dígitos decimais.▪ Escala s:<ul style="list-style-type: none">▪ Inferido do valor.▪ Ocupa:<ul style="list-style-type: none">▪ [1, 22] bytes.
FLOAT	<ul style="list-style-type: none">▪ Precisão:<ul style="list-style-type: none">▪ 126 dígitos binários.
BINARY_FLOAT	<ul style="list-style-type: none">▪ Representa n^{os} reais de vírgula-flutuante de 32-bit.▪ Ocupa:<ul style="list-style-type: none">▪ 4 bytes.
BINARY_DOUBLE	<ul style="list-style-type: none">▪ Representa n^{os} reais de vírgula-flutuante de 64-bit.▪ Tem o dobro da precisão do BINARY_FLOAT.▪ Ocupa:<ul style="list-style-type: none">▪ 8 bytes.

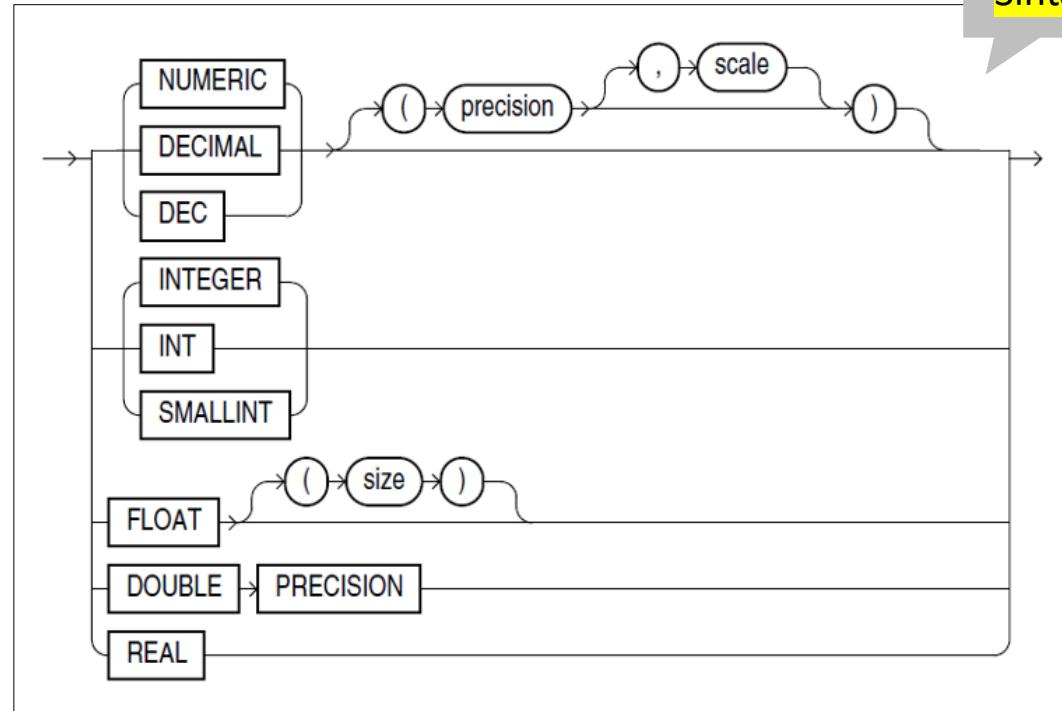
Exemplo:

- FLOAT(5)
- 5 dígitos binários

Tipos Numéricos Standard

■ Tipos

- Inteiros
 - INTEGER
 - INT
 - SMALLINT
- Reais
 - Exatos
 - NUMERIC
 - DECIMAL
 - DEC
 - Aproximados
 - FLOAT
 - DOUBLE PRECISION
 - REAL



Sintaxe

■ Subtipos NUMERIC

- NUMERIC(precision, scale)
- NUMERIC(precision)
- NUMERIC

- **Precisão (p):** nº máximo de dígitos:
 - Casas decimais **incluídas**.
 - Ponto decimal **excluído**.
- **Escala (s):** nº máximo de dígitos decimais.

■ Subtipos FLOAT

- FLOAT(size)
- FLOAT

Representação de Números Decimais	Tipo	Representa
Exata	INTEGER -- ou INT	Números inteiros: <ul style="list-style-type: none">Positivos e negativos grandes.
	SMALLINT	Números inteiros: <ul style="list-style-type: none">Positivos e negativos pequenos.
	NUMERIC NUMERIC(precisão) NUMERIC(precisão, escala)	Números reais: <ul style="list-style-type: none">Com dada precisão e escala.
	DECIMAL DECIMAL(precisão) DECIMAL(precisão, escala) -- ou DEC	Números reais: <ul style="list-style-type: none">Com dada precisão e escala.
Aproximada	REAL	
	DOUBLE PRECISION	Dobro da precisão do REAL.
	FLOAT FLOAT(size)	

Tipos Numéricos Standard

SGBD Oracle

- Converte os tipos *standard* em tipos predefinidos.

Tipo Standard	Tipo Oracle	Observação
INTEGER	NUMBER(38, 0)	Precisão = 38 Escala = 0
INT		
SMALLINT		
NUMERIC(p,s)	NUMBER(p, s)	Oracle não distingue os dois tipos <i>standard</i> : <ul style="list-style-type: none">▪ NUMERIC▪ DECIMAL p: precisão. s: escala.
DECIMAL(p,s)		
REAL	FLOAT(63)	
DOUBLE PRECISION	FLOAT(126)	Oracle não distingue os dois tipos <i>standard</i> : <ul style="list-style-type: none">▪ DOUBLE PRECISION▪ FLOAT
FLOAT		

- Sub-Categorias

- Números Inteiros
- Números Reais
 - Vírgula-Fixa
 - Vírgula-Flutuante

- Números Inteiros

- Zero, positivos e negativos
- Exemplos:
 - 0
 - +127
 - -127
 - 127 -- sem sinal \Rightarrow positivo.
- Comprimento máximo:
 - 38 dígitos decimais ($nºs$ inteiros $\in [0, 9]$).

■ Números Reais (1/3)

■ Exemplos:

- NUMBER
 - 25.6
 - +30.276
 - 0.5
 - 25e-03
- BINARY_FLOAT
 - 25.6f
 - +30.276F
- BINARY_DOUBLE
 - 0.5d
 - 25e-03D

■ Gama de valores:

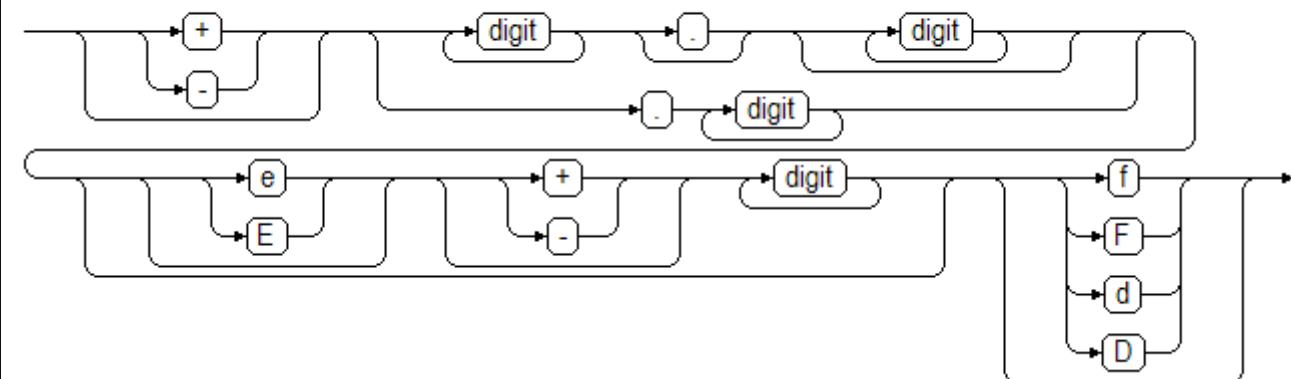
- Do respetivo tipo:
 - NUMBER
 - BINARY_FLOAT
 - BINARY_DOUBLE

■ Quando o valor excede os limites do tipo:

- Oracle lança exceção.

Sintaxe

number_value



- + or - indicates a positive or negative value. If you omit the sign, then a positive value is the default.
- digit is one of 0, 1, 2, 3, 4, 5, 6, 7, 8 or 9.
- e or E indicates that the number is specified in scientific notation. The digits after the E specify the exponent. The exponent can range from -130 to 125.
- f or F indicates that the number is a 32-bit binary floating point number of type BINARY_FLOAT.
- d or D indicates that the number is a 64-bit binary floating point number of type BINARY_DOUBLE.

If you omit f or F and d or D, then the number is of type NUMBER.

Valores Numéricos

Números Reais (2/3)

- Separadores decimais

Separador decimal da
entrada de dados.

The screenshot shows the Oracle SQL Developer interface. In the top-left, there's an 'SQL Worksheet' tab with the query:

```
INSERT INTO demo_floating_point VALUES(11.11);  
  
SELECT * FROM demo_floating_point;
```

Below it is a 'Script Output' window showing the results of the query:

NUMERO
1 11,11

The number 11,11 is highlighted in yellow, indicating the decimal separator used in the input.

Separador decimal da
saída de resultados.

The screenshot shows the Oracle SQL Developer interface with two windows open:

- Connections** window: Shows various reports and database objects.
- National Language Support Parameters** window: A table showing system parameters. One row, **13 NLS_NUMERIC_CHARACTERS**, is highlighted with a red border.

Parameter	Value
1 NLS_CALENDAR	GREGORIAN
2 NLS_CHARACTERSET	AL32UTF8
3 NLS_COMP	BINARY
4 NLS_CURRENCY	€
5 NLS_DATE_FORMAT	RR.MM.DD
6 NLS_DATE_LANGUAGE	PORTUGUESE
7 NLS_DUAL_CURRENCY	€
8 NLS_ISO_CURRENCY	PORTUGAL
9 NLS_LANGUAGE	PORTUGUESE
10 NLS_LENGTH_SEMANTICS	BYTE
11 NLS_NCHAR_CHARACTERSET	AL16UTF16
12 NLS_NCHAR_CONV_EXCP	FALSE
13 NLS_NUMERIC_CHARACTERS	, .
14 NLS_SORT	WEST EUROPEAN

Separadores decimais, por omissão:

- Saída
- Entrada

■ Números Reais (3/3)

- Valores especiais
 - Valores que não podem ser expressos por números.

Literal	Meaning	Example
binary_float_nan	A value of type BINARY_FLOAT for which the condition IS NAN is true	<pre>SELECT COUNT(*) FROM employees WHERE TO_BINARY_FLOAT(commission_pct) != BINARY_FLOAT_NAN;</pre>
binary_float_infinity	Single-precision positive infinity	<pre>SELECT COUNT(*) FROM employees WHERE salary < BINARY_FLOAT_INFINITY;</pre>
binary_double_nan	A value of type BINARY_DOUBLE for which the condition IS NAN is true	<pre>SELECT COUNT(*) FROM employees WHERE TO_BINARY_FLOAT(commission_pct) != BINARY_FLOAT_NAN;</pre>
binary_double_infinity	Double-precision positive infinity	<pre>SELECT COUNT(*) FROM employees WHERE salary < BINARY_DOUBLE_INFINITY;</pre>

■ Operações Aritméticas

- Usam
 - [Operadores Aritméticos](#)
- Operandos
 - Podem ser de diferentes tipos numéricos.
- Tipo de Resultado
 - Depende dos tipos dos operandos. -- tipo do maior valor numérico.
 - Se não representar o resultado de forma exata:
 - Resultado é truncado ou arredondado. -- depende do tipo do resultado.
- Gerada exceção:
 - Se o resultado:
 - Está fora da gama de valores do tipo do resultado;
 - Se a operação aritmética:
 - Não está definida para os operandos.

Tipos de Dados Oracle

Tipos Numéricos

Tipos Caráter

Tipos Data-Tempo e Intervalo

Precedência

Conversão

- [Introdução](#)
- [Sub-Categorias](#)
 - [Tipos Predefinidos](#)
 - [Tipos Standard](#)
- [Comparação](#)
- [Operação de Concatenação](#)
- [Funções de Carateres](#)
- [Precedência](#)
- [Conversão de Tipos de Dados](#)

Introdução

■ Interesse

- Armazenar carateres -- dados alfanuméricicos.
- Exemplos: palavras / texto livre.

■ Armazenam Caracteres

- Em **strings** de bytes
- Valores dos bytes:
 - Do **character set** ... definido na criação da BD.

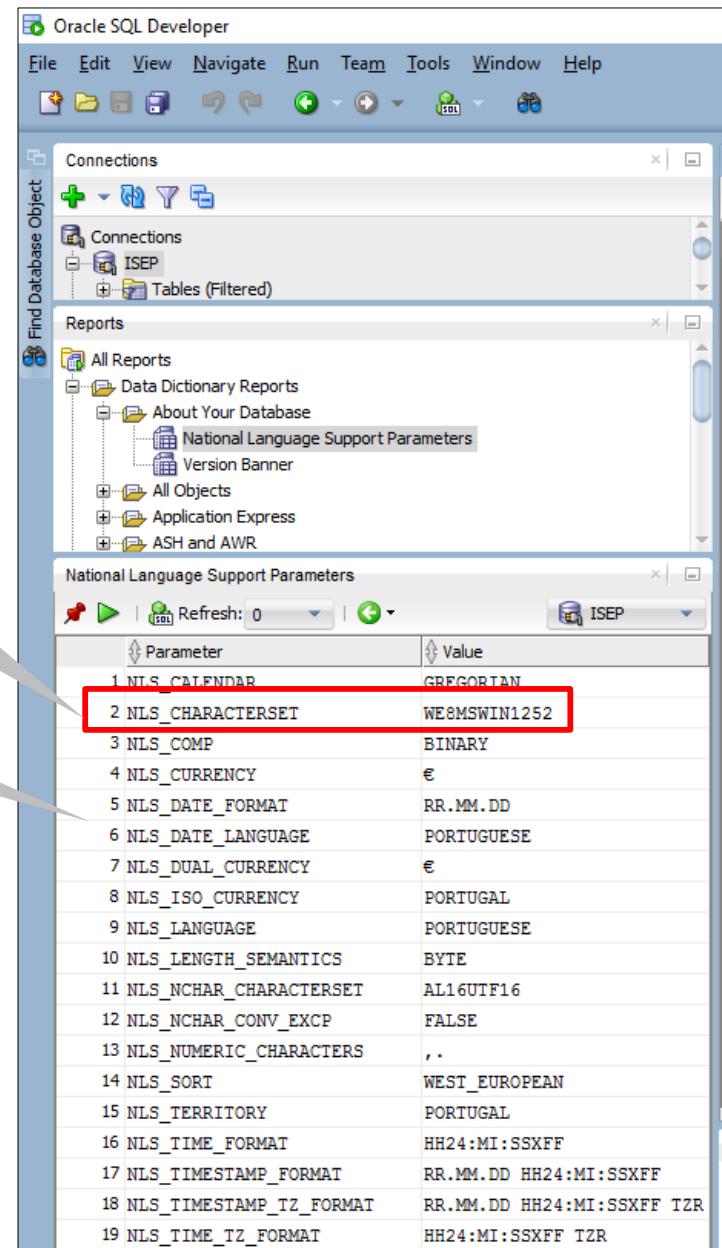
Character set por omissão.

Parâmetros de Inicialização da BD:

- Definições por omissão.
- **NLS** = National Language Support

■ Valor do Tipo

- Colocado entre **plicas**.
 - Exemplo: `'isto é uma string'`
- Plica **dentro** de *string*:
 - Duplicar plica.
 - Exemplo: `'Jackie''s car'`.
- Comprimento máximo:
 - Por omissão: 4000 bytes.



- Sub-Categorias

- Tipos Predefinidos
- Tipos *Standard*

Tipos Caráter Predefinidos

■ Tipos

- Comprimento Fixo

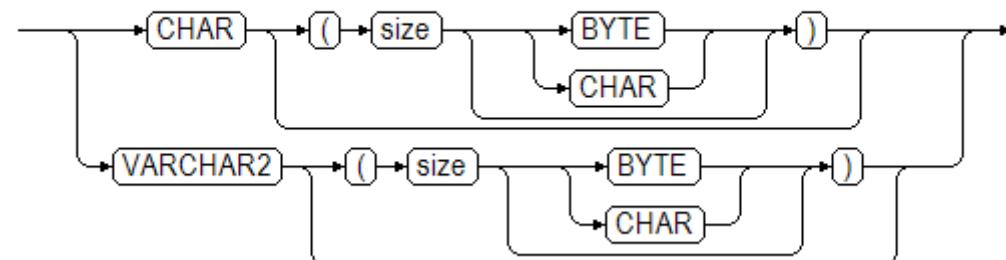
- CHAR

- Comprimento Variável

- VARCHAR2

Sintaxe

predefined_character_data_types



■ Subtipos CHAR

- CHAR -- size 1 em bytes (por omissão)
- CHAR(size) -- size em bytes (por omissão)
- CHAR(size BYTE) -- size em bytes
- CHAR(size CHAR) -- size em carateres

Size de um caráter:

- depende do character set da BD: 1, 2, 3 ou 4 bytes.

■ Subtipos VARCHAR2

- VARCHAR2(size)
- VARCHAR2(size BYTE)
- VARCHAR2(size CHAR)

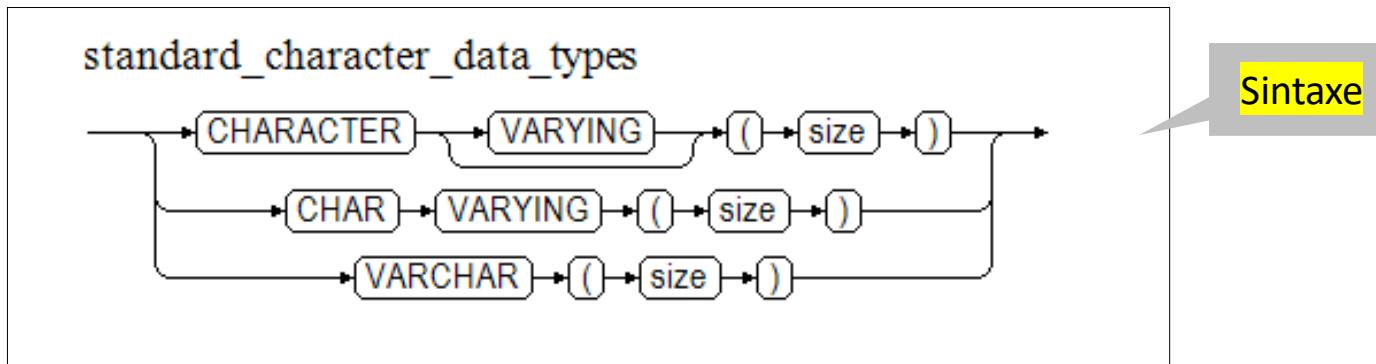
Qualifier CHAR
sobrepõe-se ao
valor por omissão.

Qualifier BYTE
por omissão.

National Language Support Parameters	
Parameter	Value
1 NLS_CALENDAR	GREGORIAN
2 NLS_CHARACTERSET	WE8MSWIN1252
3 NLS_COMP	BINARY
4 NLS_CURRENCY	€
5 NLS_DATE_FORMAT	RR.MM.DD
6 NLS_DATE_LANGUAGE	PORTUGUESE
7 NLS_DUAL_CURRENCY	€
8 NLS_ISO_CURRENCY	PORTUGAL
9 NLS_LANGUAGE	PORTUGUESE
10 NLS_LENGTH_SEMANTICS	BYTE
11 NLS_NCHAR_CHARACTERSET	AL16UTF16

Tipos Caráter Predefinidos

Comprimento	Tipo	Descrição	Exemplos
Fixo	CHAR	Armazena: <ul style="list-style-type: none">▪ 1 carácter.	S/N: <ul style="list-style-type: none">▪ CHAR(1)
	CHAR(size)	Armazena: <ul style="list-style-type: none">▪ <i>string</i> de caracteres de comprimento fixo. Comprimento: <ul style="list-style-type: none">▪ Definido em bytes, por omissão. Adicionados espaços (à direita): <ul style="list-style-type: none">▪ Se <i>string</i> tem comprimento < size.	Código Postal: <ul style="list-style-type: none">▪ CHAR(8)
Variável	VARCHAR2(size)	Armazena: <ul style="list-style-type: none">▪ <i>string</i> de caracteres de comprimento variável. Comprimento: <ul style="list-style-type: none">▪ Definido em bytes, por omissão. Vantagem: <ul style="list-style-type: none">▪ Permite poupar espaço de armazenamento. VARCHAR <ul style="list-style-type: none">▪ É sinónimo.	Nome: <ul style="list-style-type: none">▪ VARCHAR(30)



Comprimento	Tipo de Dados <i>Standard</i>	Oracle Converte no Tipo Predefinido
Fixo	CHARACTER(size)	CHAR(size)
Variável	CHAR VARYING(size)	VARCHAR2(size)
	CHARACTER VARYING(size)	VARCHAR2(size)
	VARCHAR(size)	VARCHAR2(size)

Tipos de Dados Oracle

Tipos Numéricos

Tipos Caráter

Tipos Data-Tempo e Intervalo

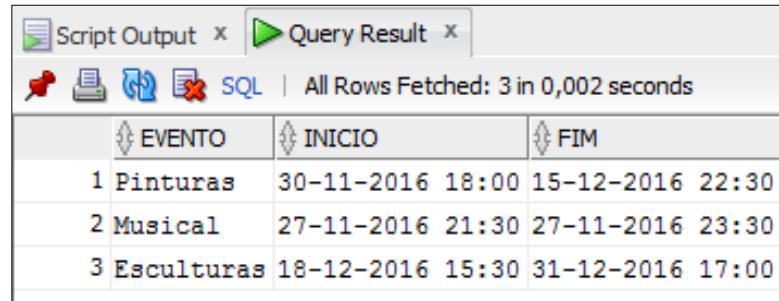
Precedência

Conversão

- [Interesse](#)
- [Tipos Data-Tempo](#)
 - [Sub-Categorias Data-Tempo](#)
 - Predefinidos
 - *Standard*
 - [Valores Data-Tempo](#)
 - [Tipos Predefinidos](#)
 - [DATE](#)
 - [TIMESTAMP](#)
- [Tipos Intervalo](#)
 - [Tipos Predefinidos](#)
 - [Valores Intervalo](#)
 - [Coluna de Tabela](#)
 - [Aritmética](#)
- [Comparação](#)
- [Funções Data-Tempo e Intervalo](#)
- [Precedência](#)
- [Conversão de Tipos de Dados](#)

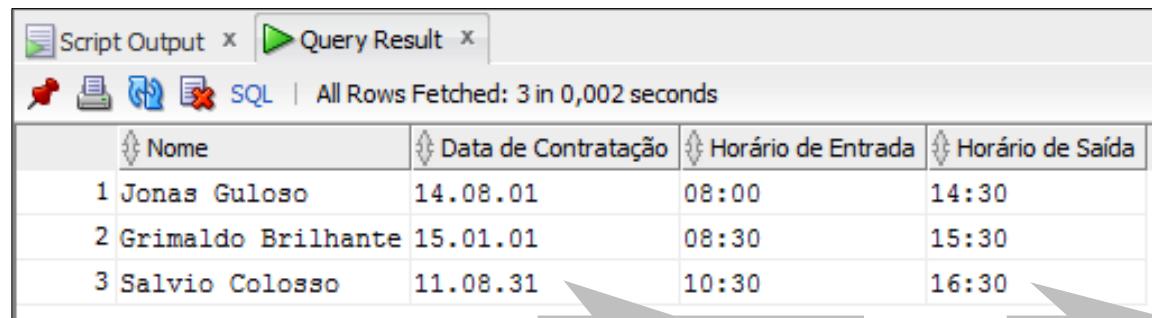
■ Representação

- Instantes de tempo: tipos Data-Tempo.



EVENTO	INICIO	FIM
1 Pinturas	30-11-2016 18:00	15-12-2016 22:30
2 Musical	27-11-2016 21:30	27-11-2016 23:30
3 Esculturas	18-12-2016 15:30	31-12-2016 17:00

Data e Tempo

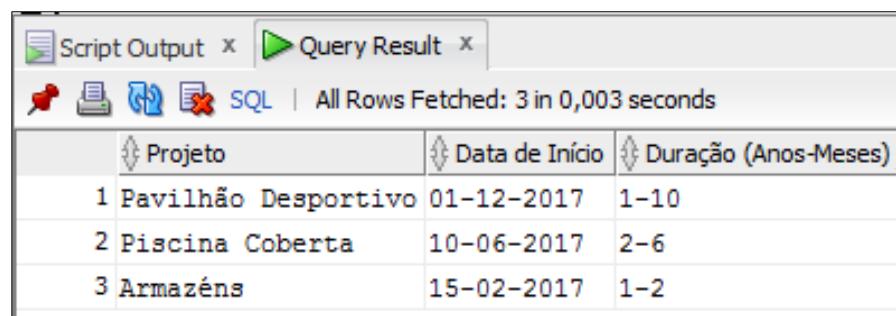


Nome	Data de Contratação	Horário de Entrada	Horário de Saída
1 Jonas Guloso	14.08.01	08:00	14:30
2 Grimaldo Brilhante	15.01.01	08:30	15:30
3 Salvio Colosso	11.08.31	10:30	16:30

Visualizada a Data

Visualizado o Tempo

- Períodos de tempo: tipos Intervalo.



Projeto	Data de Início	Duração (Anos-Meses)
1 Pavilhão Desportivo	01-12-2017	1-10
2 Piscina Coberta	10-06-2017	2-6
3 Armazéns	15-02-2017	1-2

Período de tempo.

- Categorias

- Tipos Predefinidos

- DATE
- TIMESTAMP

Em conformidade com os respetivos tipos *standard*.

- Tipos Standard

- DATE
- TIMESTAMP
- TIME -- não suportado pelo Oracle.

▪ Um Valor Data-Tempo

▪ Armazenado:

- Numa **string** de caracteres ... com **formato especial**.

- Constituída por **campos**:

- **Data**: YEAR, MONTH, DAY

- **Tempo**: HOUR, MINUTE, SECOND, TIMEZONE HOUR, TIMEZONE MINUTE

- Exemplos:

- **Data**: '2015-01-08'

- **Data-Tempo**: '2015-01-18 01:15:20.**32**'

string formada por **campos de data**:

- YEAR, MONTH e DAY

string formada por **campos de data e tempo**:

- YEAR, MONTH, DAY, HOUR, MINUTE, SECOND

Data e tempo:

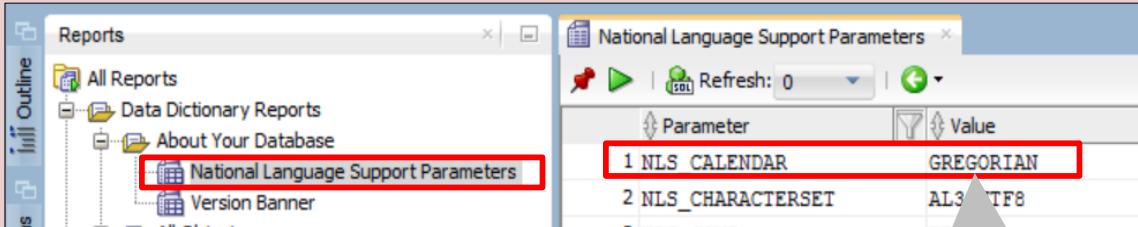
- separados por **espaço**.

Frações do segundo:

- 32 centésimas do segundo.

■ Campos de Data e Tempo

■ Valores válidos

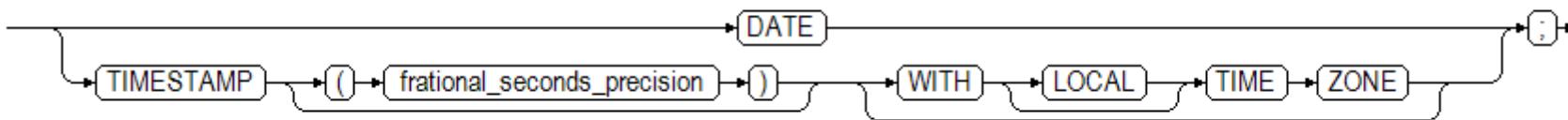
Campo	Valores Válidos
YEAR	<p>[-4712, 9999]</p> <ul style="list-style-type: none"> excluindo o ano 0.
MONTH	[01, 12]
DAY	<p>[01, 31]</p> <ul style="list-style-type: none"> Dias limitados pelos valores de MONTH e YEAR, de acordo com as regras do parâmetro NLS_CALENDAR atual. 
HOUR	[00, 23]
MINUTE	[00, 59]
SECOND	<p>[00, 59] -- Tipo DATE [00, 59.9(n)] -- Tipos TIMESTAMP</p> <ul style="list-style-type: none"> 9(n) <ul style="list-style-type: none"> Precisão da fração do segundo.

Por omissão:

- Calendário Gregoriano.

Tipos Data-Tempo Predefinidos

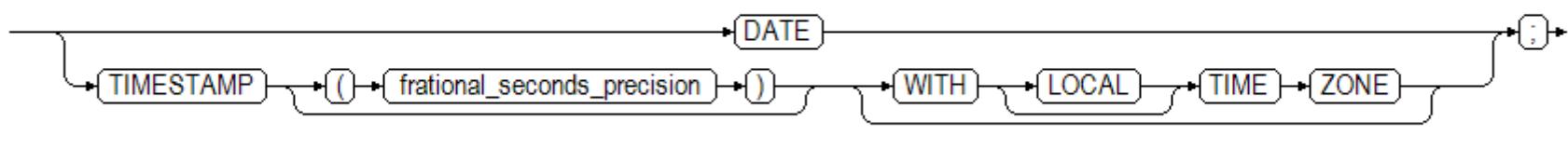
datetime_datatypes_syntax



Tipo	Armazenamento na BD	Exemplos de Valores
DATE	<p>Guarda:</p> <ul style="list-style-type: none"> Data e tempo em simultâneo. <p>Não guarda:</p> <ul style="list-style-type: none"> Frações do segundo. Fusos horários. 	'2015-11-18' DATE '2015-11-18' TO_DATE ('2015-11-18', 'YYYY-MM-DD') TO_DATE ('15-08-05 04:22','YY-MM-DD HH:MI')
TIMESTAMP	<p>Guarda:</p> <ul style="list-style-type: none"> Data e tempo em simultâneo. Frações do segundo. <p>Não guarda fusos horários.</p>	'2015-08-05' '2015-08-05 04:22:15.52' TIMESTAMP '2015-08-05 04:22:15'
TIMESTAMP WITH TIME ZONE	<p>Guarda:</p> <ul style="list-style-type: none"> Data e tempo em simultâneo. Frações do segundo. Campos do fuso horário: <ul style="list-style-type: none"> TIMEZONE_HOUR TIMEZONE_MINUTE 	'2015-08-05 12:0:0 -08:00' '2015-08-05 12:0:0 +2:00' '2015-08-05 12:0:0' TIMESTAMP '2015-08-05 12:0:0 -08:00'
TIMESTAMP WITH LOCAL TIME ZONE	<p>Guarda:</p> <ul style="list-style-type: none"> Data e tempo em simultâneo. Frações do segundo. Fuso horário local. 	Não há valor <u>particular</u> deste tipo. Representado pelos tipos TIMESTAMP e TIMESTAMP WITH TIME ZONE .

Representa	Armazenamento na BD
Valores de data e tempo aproximado .	<p>Guarda:</p> <ul style="list-style-type: none">▪ Data e tempo em simultâneo. <p>Não guarda:</p> <ul style="list-style-type: none">▪ Frações do segundo.▪ Não guarda fusos horários.

datetime_datatypes_syntax

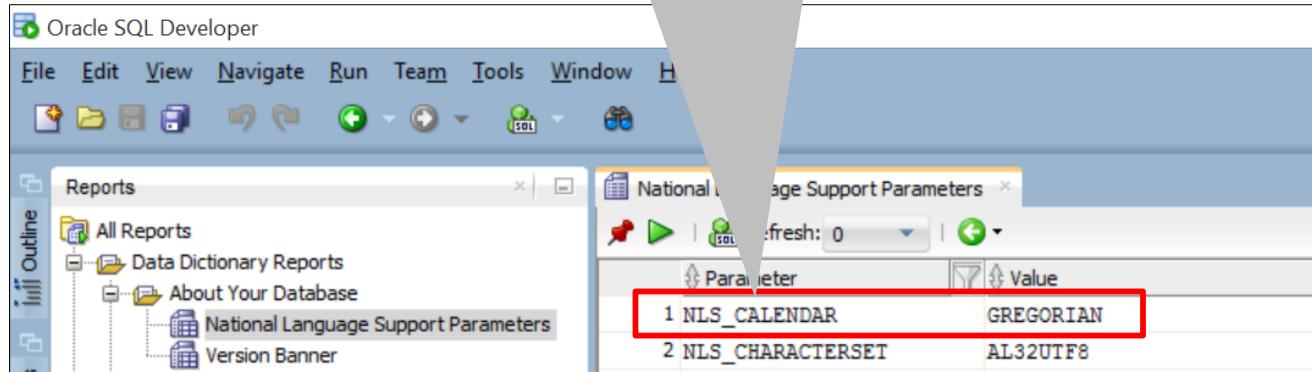


Tipo DATE

Datas dos Valores DATE

- Por omissão
 - Do calendário Gregoriano.

Especificado no parâmetro NLS_CALENDAR



Gama de Datas

- [1-1-4712 AC, 31-12-9999 DC] -- AC = Antes de Cristo ; DC = Depois de Cristo

Gama de Tempos

- Formato 24 Horas: [00:00:00, 29:59:59] -- '24-JAN-2015 15:30'
- Formato 12 Horas (AM / PM): [00:00:00, 12:00:00] -- '24-JAN-2015 3:30 PM'
-- '25-FEV-2015 1:20 AM'

Dimensão

- Fixa.
- Ocupa 7 bytes.

■ Valores DATE (1/4)

Formas de Representação	Exemplos
String	<p>'2016-01-08' '2016/01/08' '16-01-08' '16-1-8' '2016 janeiro 08' '16 jan 08' -- nome do mês abreviado '2015-08-05 04:22:15' '2016-09-02 03:11' '2015-12-19 04'</p>
Standard <ul style="list-style-type: none"> ▪ Formato 'YYYY-MM-DD'. ▪ Usa prefixo DATE. ▪ Não contém o tempo. 	<p>DATE '2016-01-08'</p> <div data-bbox="1224 807 1614 1048" style="background-color: #e0e0e0; padding: 10px; width: fit-content; margin-left: 20px;"> <p>Modelo de formato.</p> <ul style="list-style-type: none"> ▪ Y : dígito do ano. ▪ M : dígito do mês. ▪ D : dígito do dia. </div>
Expressão <u>TO_DATE</u> <ul style="list-style-type: none"> ▪ <u>TO_DATE</u> é uma função. ▪ Pode substituir uma <i>string</i> DATE. 	<p>TO_DATE('2016-01-08', 'YYYY-MM-DD') TO_DATE('16-01-08', 'YY-MM-DD') TO_DATE('2016 jan 08 17:10:05', 'YYYY MON DD HH24:MI:SS')</p>

Tipo DATE

■ Valores DATE (2/4)

- Formato por omissão (1/2)
 - Especificado no parâmetro [NLS_DATE_FORMAT](#).

Formato aplicado na:

- Visualização dos valores armazenados.
- Introdução de valores na BD.

Parameter	Value
1 NLS_CALENDAR	GREGORIAN
2 NLS_CHARACTERSET	AL32UTF8
3 NLS_COMP	BINARY
4 NLS_CURRENCY	€
5 NLS_DATE_FORMAT	RR.MM.DD
6 NLS_DATE_LANGUAGE	PORTUGUESE

- Alteração do parâmetro NLS_DATE_FORMAT ... válida apenas na sessão atual.

```
Worksheet      Query Builder
ALTER SESSION
SET NLS_DATE_FORMAT = 'DD-MON-YYYY HH24:MI:SS';

Script Output      Query Result      Query Result 1      Query
SQL | All Rows Fetched: 1 in 0,001 seconds

DATE
1 31-OUT-2016 15:58:59
```

Parameter	Value
1 NLS_CALENDAR	GREGORIAN
2 NLS_CHARACTERSET	AL32UTF8
3 NLS_COMP	BINARY
4 NLS_CURRENCY	€
5 NLS_DATE_FORMAT	DD-MON-YYYY HH24:MI:SS

Tipo DATE

■ Valores DATE (3/4)

■ Formato por omissão (2/2)

■ RR . MM . DD

The screenshot shows a Worksheet window with the following code:

```

1 | INSERT INTO demo_date VALUES('2016/01/01');
2 | INSERT INTO demo_date VALUES('2016/1/2');
3 | INSERT INTO demo_date VALUES('16/01/03');
4 | INSERT INTO demo_date VALUES('2016-01-04');
5 | INSERT INTO demo_date VALUES('2016.01.05');
6 | INSERT INTO demo_date VALUES('2016 01 06');
7 | INSERT INTO demo_date VALUES('16 janeiro 07');
8 | INSERT INTO demo_date VALUES('16 jan 8');
9 |
10| SELECT * FROM demo_date;
  
```

A callout box points to the first four lines with the text: "Exemplos de alguns formatos de entrada válidos."

Below the code, the Script Output window shows the output:

```

DATA
-----
16.01.02
16.01.03
16.01.04
16.01.05
16.01.06
16.01.07
16.01.08
16.01.01

8 rows selected
  
```

A callout box points to the first few lines with the text: "Assumidos diferentes separadores de campos de data."

At the bottom, a callout box points to the date separator ".": "Formato de saída:
RR.MM.DD"

The screenshot shows a Worksheet window with the following code:

```

13 |
14 | INSERT INTO demo_date VALUES('2016/10/10 12:05:19');
  
```

An error message is displayed in the Script Output window:

```

Error starting at line : 14 in command -
INSERT INTO demo_date VALUES('2016/10/10 12:05:19')
Error report -
SQL Error: ORA-01861: literal does not match format string
01861. 00000 - "literal does not match format string"
*Cause: Literals in the input must be the same length as literals in
the format string (with the exception of leading whitespace). If the
"FX" modifier has been toggled on, the literal must match exactly,
with no extra whitespace.
*Action: Correct the format string to match the literal.
  
```

A callout box points to the error message with the text: "Valor inválido."

Below, another Worksheet window shows a successful insertion using TO_DATE:

```

1 | INSERT INTO demo_date VALUES(TO_DATE('2016-10-10 12:05:19','YYYY-MM-DD HH:MI:SS'));
  
```

The Script Output window shows the result:

```

Task completed in 0,002 seconds
1 row inserted.
  
```

A callout box points to the TO_DATE expression with the text: "Expressão TO_DATE válida."

■ Valores DATE (4/4)

■ Exemplos

Formas de Representação	Valor DATE Inserido na BD	Valor DATE Armazenado na BD	Valor DATE Visualizado na Saída (Script Output)
String	<p>'2015-11-18'</p> <p>Tem de usar o formato por omissão.</p> <p>Os separadores dos campos podem ser diversos caracteres: /, espaço, ponto, ...</p>	Data guardada juntamente com o <u>tempo por omissão</u> .	<p>2015.11.18</p> <ul style="list-style-type: none"> Usado o formato por omissão.
Standard	<p>DATE '2015-11-18'</p> <ul style="list-style-type: none"> Formato 'YYYY-MM-DD'. Usa prefixo DATE. Não contém o tempo. 	Data guardada juntamente com o <u>tempo por omissão</u> .	<p>2015.11.18</p> <ul style="list-style-type: none"> Usado o formato por omissão.
Expressão <u>TO_DATE</u>	<p>TO_DATE('12:08:35', 'HH:MI:SS')</p> <p>Formato especificado pode ser diferente do formato por omissão.</p> <p>Por exemplo, permite introduzir apenas tempos.</p>	Tempo inserido juntamente com a <u>data por omissão</u> .	<ul style="list-style-type: none"> Visualizada apenas a <u>data por omissão</u>.

Tipo DATE

Valores DATE Armazenados (1/2)

- Tempo por omissão

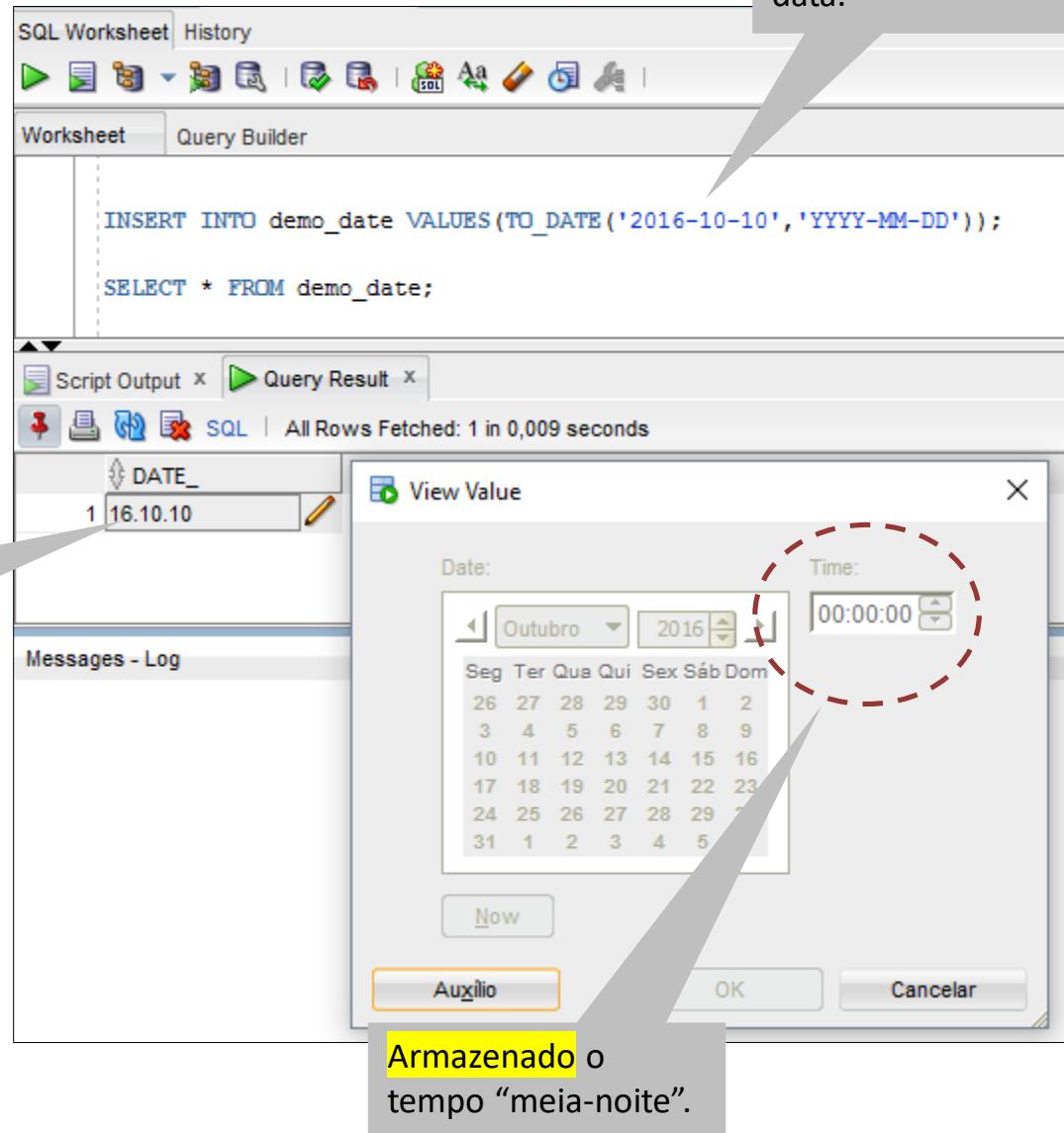
- Meia-noite:

- Formato 24 horas
 - '00:00:00'

- Formato 12 horas
 - AM / PM
 - '12:00:00'

Visualizado apenas a data , de acordo com o parâmetro NLS_DATE_FORMAT.

National Language Support Parameters	
Parameter	Value
1 NLS_CALENDAR	GREGORIAN
2 NLS_CHARACTERSET	WE8MSWIN1252
3 NLS_COMP	BINARY
4 NLS_CURRENCY	€
5 NLS_DATE_FORMAT	RR.MM.DD
6 NLS_DATE_LANGUAGE	PORTUGUESE



■ Valores DATE Armazenados (2/2)

■ Data por omissão

- 1º dia do mês atual.

■ Exemplo

The screenshot shows the Oracle SQL Worksheet interface. In the 'Worksheet' tab, the following SQL code is entered:

```
19  
20: INSERT INTO demo_date VALUES (TO_DATE('12:12:12', 'HH:MI:SS'));  
21  
22: SELECT * FROM demo_date;  
23
```

The output window shows the result of the query:

DATE_
1 16.10.01

A callout box highlights the date '1 16.10.01' and states: '1º dia do mês 10.'

A modal dialog titled 'View Value' is open, showing a calendar and a time picker. The calendar displays October 2016. The time picker shows '12:12:12'. A dashed red circle highlights the time value. A callout box next to the dialog states: 'Armazenado o tempo inserido.'

A callout box in the top right corner states: 'Tempo registado em 25-10-2016.'

■ Colunas DATE (1/4)

- Contém sempre:
 - Ambos os campos de data e tempo.
- Para processar apenas datas:
 - Guardar todas as datas:
 - Com o tempo por omissão “meia-noite”.
 - Guardar apenas:
 - Valor DATE só com campos de data. -- tempos todos iguais a “meia-noite”.
 - Exemplo: '16-10-26'.
 - Resultado da função TO_DATE sem especificar o tempo.
 - Exemplo: TO_DATE('16-10-26','YY-MM-DD').
 - Resultado da expressão TRUNC(SYSDATE).
 - Função SYSDATE
 - Retorna a data e o tempo atual do sistema; resultado é do tipo DATE.
 - Função TRUNC
 - Trunca a parte data de um valor DATE; resultado é do tipo DATE.
 - Justificação
 - Próximos slides.

■ Colunas DATE (2/4)

■ Função SYSDATE (1/3)

- Retorna a **data** e o **tempo** atual do sistema num valor do tipo DATE: **data e tempo**.
- Exemplo:

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, there are tabs for 'Start Page', 'BDDAD 2.sql' (which is the active tab), and 'DEMO_DATE'. Below the tabs is a toolbar with various icons. The main area is divided into two panes: 'Worksheet' on the left and 'Query Builder' on the right. The 'Worksheet' pane contains the following SQL code:

```
INSERT INTO demo_date VALUES(SYSDATE); -- insere a data atual do sistema na tabela "demo_date".
SELECT * FROM demo_date; -- seleciona todas linhas da tabela "demo_date".
```

A gray callout box points from the text 'SYSDATE inserida na BD.' to the first line of the code.

In the 'Query Result' tab, a table named 'DATE_' is displayed with one row:

	DATE_
1	16.10.26

Three gray callout boxes point to specific parts of the result:

- 'Data armazenada.' points to the date value '16.10.26'.
- 'Tempo armazenado.' points to the time value '15:38:49'.
- 'Data inserida na BD.' points to the explanatory comment in the code.

A 'View Value' dialog box is open over the table, showing a calendar and a time picker. The calendar displays October 2016, and the time picker shows 15:38:49. Buttons at the bottom of the dialog include 'Now', 'Auxílio' (Help), 'OK', and 'Cancelar' (Cancel).

Tipo DATE

Colunas DATE (3/4)

- Função SYSDATE (2/3)

- Exemplo:

Consultar as linhas da tabela com a data inserida anteriormente.

Problema: resultado = vazio

- Partes dos tempos são diferentes.
- Tempo da SYSDATE = '15:38:49'.
- Tempo da data comparada é o tempo por omissão: "meia-noite"

Solução:

- Comparar a data pretendida com as datas armazenadas truncadas.
- Requer truncagem de todas as datas armazenadas.

Slide seguinte apresenta uma solução com melhor desempenho.

The screenshot shows the Oracle SQL Worksheet interface. In the top pane, there is a code editor with the following SQL query:

```
-- seleciona todas as linhas da tabela "demo_date" com a data especificada
SELECT * FROM demo_date WHERE date_ = '16.10.26';
```

In the bottom pane, there are several tabs labeled "Script Output", "Query Result", "Query Result 1", "Query Result 2", and "Query Result 3". The "Query Result" tab is active, showing the message "All Rows Fetched: 0". Below the tabs, there is a toolbar with icons for "DATE_" and other functions.

Comparação inclui os tempos.

The screenshot shows the Oracle SQL Worksheet interface. In the top pane, there is a code editor with the following SQL query:

```
-- seleciona todas as linhas da tabela "demo_date" com a data especificada
SELECT * FROM demo_date WHERE TRUNC(date_) = '16.10.26';
```

In the bottom pane, there are several tabs labeled "Script Output", "Query Result", "Query Result 1", "Query Result 2", and "Query Result 3". The "Query Result 1" tab is active, showing the message "All Rows Fetched: 1 in 0,002 seconds". Below the tabs, there is a toolbar with icons for "DATE_" and other functions.

Resultado esperado.

■ Colunas DATE (4/4)

■ Função SYSDATE (3/3)

- Exemplo:

The screenshot shows an Oracle SQL Worksheet interface. In the top-left pane, there is a code editor with two queries:

```
SQL Worksheet History
Worksheet Query Builder
INSERT INTO demo_date VALUES (TRUNC(SYSDATE));
SELECT * FROM demo_date WHERE date_ = '16.10.26';
```

In the bottom-right pane, there is a "Query Result" tab showing the output of the second query:

DATE_
1 16.10.26

Annotations on the slide highlight specific parts of the code and results:

- A callout points to the first query in the code editor with the text: "Mais rápida que a da solução do slide anterior."
- A callout points to the result table with the text: "SYSDATE truncada antes de inserir."
- A callout points to the second query in the code editor with the text: "Na comparação de datas:"
- A callout points to the result table with the text: "Daí:"
- A callout points to the result table with the text: "Esta consulta é mais rápida do que a da solução do slide anterior."

- Operações Aritméticas Suportadas (1/2)

Operando	Operação	DATE	TIMESTAMP	INTERVAL	Numérico (Nº de Dias)
DATE	+	illegal	illegal	DATE + INTERVAL = DATE	DATE + Nº DIAS = DATE
	-	DATE – DATE = nº de dias	DATE – TIMESTAMP = INTERVAL	DATE – INTERVAL = DATE	DATE – Nº DIAS = DATE
	*			Illegal	
	/			Illegal	

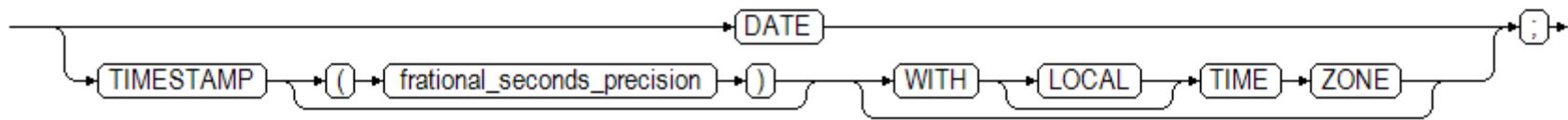
■ Operações Aritméticas Suportadas (2/2)

- Exemplos
 - SYSDATE – TO_DATE('2015-11-17')
 - Nº de dias entre datas:
 - Atual do sistema e especificada.
 - SYSDATE
 - Inclui um tempo.
 - Característica do tipo DATE.
 - Resultado
 - Pode incluir uma fração de um dia.
 - Exemplo

The screenshot shows the Oracle SQL Developer interface. In the top-left window (Worksheet), there is a query:SELECT SYSDATE - TO_DATE ('2015-12-12')
FROM DUAL;The bottom-right window (Query Result) displays the output of the query:SYSDATE-TO_DATE('2015-12-12')
1 317,191377314814814814814814814814814814815

The screenshot shows the Oracle SQL Developer interface with two queries in the top-left worksheet:SELECT * FROM DUAL;The bottom-right window (Query Result) displays the output of the second query:DUMMY
1 X

datetime_datatypes_syntax



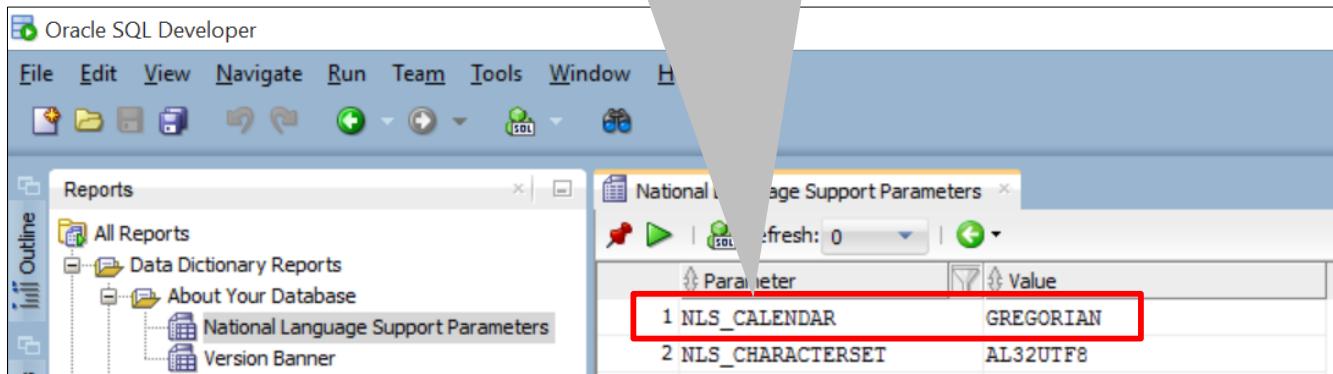
Variantes	Armazenamento	Precisão de Frações do Segundo	Exemplos de Valores
TIMESTAMP(p) <ul style="list-style-type: none"> Representação de datas e tempos mais precisa do que <code>DATE</code>. Permite especificar a precisão de frações do segundo (p). 	Guarda: <ul style="list-style-type: none"> Data e tempo em simultâneo. Frações do segundo. Não guarda: <ul style="list-style-type: none"> Fusos horários. Ocupa: <ul style="list-style-type: none"> 7 ou 11 bytes. Depende da precisão. 	Nº de dígitos da parte fracional do campo SECOND. Valor: [0,9]	'2015-08-05' '2015-08-05 04:22:15' '2015-08-05 04:22:15.52' <code>TIMESTAMP '2015-08-05 04:22:15'</code> <code>TIMESTAMP '2015-08-05 04:22:15.17'</code> <code>TO_TIMESTAMP('2015-08-05', 'YYYY-MM-DD')</code> <code>TO_TIMESTAMP('2015-08-05 04:22:15.00', 'YYYY-MM-DD HH24:MI:SS.FF')</code>
TIMESTAMP		=TIMESTAMP(6)	

Frações do segundo.

Datas dos Valores

- Por omissão
 - Do calendário Gregoriano.
 - Igual ao tipo DATE.

Especificado no parâmetro NLS_CALENDAR



Gama de Datas

- [1-1-4712 AC, 31-12-9999 DC] -- AC = Antes de Cristo ; DC = Depois de Cristo
- Igual ao tipo DATE.

Gama de Tempos

- Formato 24 Horas: [00:00:00, 23:59:59,999999999] -- '24-JAN-2015 15:30:08,32'
- Formato 12 Horas (AM / PM): [00:00:00, 12:00:00,999999999] -- '24-JAN-2015 3:30:08,32 PM'
-- '25-FEV-2015 1:20 A'
- Diferente do tipo DATE.
 - Frações do segundo.

■ Valores TIMESTAMP (1/5)

Formas de Representação	Exemplos
String	<p>'2016-01-08' -- tempo opcional.</p> <p>'2016/01/08' -- separadores de datas diferentes.</p> <p>'16-01-08' -- ano = dois últimos dígitos</p> <p>'16-1-8' -- dia e ano com 1 dígito</p> <p>'2016 janeiro 08' -- mês por extenso; ling. por omissão</p> <p>'16 jan 08' -- nome do mês abreviado</p> <p>'2015-08-05 04:22:15.23' -- tempo com frações do seg.</p> <p>'2016-09-02 03:11' -- tempo sem seg</p> <p>'2015-12-19 04' -- tempo só tem horas.</p>
Standard	<p>TIMESTAMP '2016-10-27 01:15:25'</p> <p>TIMESTAMP '2016-10-27 01:15:25.5'</p> <p>TIMESTAMP '2016-10-27 01:15:25.123456789'</p>
Expressão TO_TIMESTAMP	<p>TO_TIMESTAMP('2016-01-08', 'YYYY-MM-DD')</p> <p>TO_TIMESTAMP('16-01-08', 'YY-MM-DD')</p> <p>TO_TIMESTAMP('2016 jan 08 17:10:05', 'YYYY MON DD HH24:MI:SS')</p>

Tipo TIMESTAMP

■ Valores TIMESTAMP (2/5)

■ Formato por omissão (1/2)

- Especificado no parâmetro NLS_TIMESTAMP_FORMAT.

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Reports' tree view is open, with 'National Language Support Parameters' selected and highlighted with a red box. In the main pane, the 'Parameter' table shows various NLS parameters. The row for 'NLS_TIMESTAMP_FORMAT' is also highlighted with a red box. The table data is as follows:

Parameter	Value
1 NLS_CALENDAR	POR
2 NLS_CHARACTERSET	PORTUGAL
3 NLS_COMP	BYTE
4 NLS_CURRENCY	AL16UTF1
5 NLS_DATE_FORMAT	FALSE
6 NLS_DATE_LANGUAGE	WEST_EUROPEAN
7 NLS_DUAL_CURRENCY	PORTUGAL
8 NLS_ISO_CURRENCY	HH24:MI:SSXFF
9 NLS_LANGUAGE	RR.MM.DD HH24:MI:SSXFF
10 NLS_LENGTH_SEMANTICS	17 NLS_TIMESTAMP_FORMAT
11 NLS_NCHAR_CHARACTERSET	RR.MM.DD HH24:MI:SSXFF TZR
12 NLS_NCHAR_CONV_EXCP	18 NLS_TIMESTAMP_TZ_FORMAT
13 NLS_NUMERIC_CHARACTERS	19 NLS_TIME_TZ_FORMAT
14 NLS_SORT	RR.MM.DD HH24:MI:SSXFF TZR
15 NLS_TERRITORY	RR.MM.DD HH24:MI:SSXFF TZR
16 NLS_TIME_FORMAT	RR.MM.DD HH24:MI:SSXFF TZR
17 NLS_TIMESTAMP_FORMAT	RR.MM.DD HH24:MI:SSXFF TZR
18 NLS_TIMESTAMP_TZ_FORMAT	RR.MM.DD HH24:MI:SSXFF TZR
19 NLS_TIME_TZ_FORMAT	RR.MM.DD HH24:MI:SSXFF TZR

RR: 2 últimos dígitos do ano.

MM: dígitos do mês ([01-12]; Janeiro 01).

DD: dígitos do dia do mês([01-31]).

HH24: dígitos da hora (formato 24 horas: [0-23])

MI: dígitos dos minutos ([0-59]).

SS: dígitos dos segundos ([0-59]).

X: separador decimal local.

FF: frações do segundo (precisão = 9 dígitos)

Formato aplicado na:

- Visualização dos valores armazenados.
- Introdução de valores na BD.

■ Valores TIMESTAMP (3/5)

■ Formato por omissão (2/2)

■ Exemplo

```

Worksheet Query Builder
CREATE TABLE demo_timestamp (
    tempo TIMESTAMP
);

1 INSERT INTO demo_timestamp VALUES ('2016-10-27');
2 INSERT INTO demo_timestamp VALUES ('2016-10-27 12');
3 INSERT INTO demo_timestamp VALUES ('2016/10/27 12:12');
4 INSERT INTO demo_timestamp VALUES ('2016.10.27 12:12:12,00');
5 INSERT INTO demo_timestamp VALUES ('2016 10 27 12:12:12,123');
6 INSERT INTO demo_timestamp VALUES ('2016-10-27 12:12:12,1234');
7 INSERT INTO demo_timestamp VALUES ('2016-10-27 12:12:12,123456');
8 INSERT INTO demo_timestamp VALUES ('2016-10-27 12:12:12,1234567');
9 INSERT INTO demo_timestamp VALUES ('2016-out-27');
10 INSERT INTO demo_timestamp VALUES ('16-out-27 12:12:12,123');

Script Output x | Query Result x | Query Result 1 x
SQL | All Rows Fetched: 10 in 0,005 seconds

TEMPO
1 16.10.27 00:00:00,000000000
2 16.10.27 12:00:00,000000000
3 16.10.27 12:12:00,000000000
4 16.10.27 12:12:12,000000000
5 16.10.27 12:12:12,123000000
6 16.10.27 12:12:12,123400000
7 16.10.27 12:12:12,123456000
8 16.10.27 12:12:12,123457000
9 16.10.27 00:00:00,000000000
10 16.10.27 12:12:12,123000000

```

Valores de entrada válidos;
satisfazem o formato por
omissão.

Frações do segundo
arredondadas
porque TIMESTAMP
tem precisão = 6.

Formato de saída,
por omissão.

■ Valores TIMESTAMP Armazenados (1/2)

- Tempo por omissão
 - Meia-noite:
 - Formato 24 horas
 - '00:00:00'
 - Exemplo

The screenshot shows the Oracle SQL Developer interface. At the top, there is a 'Worksheet' tab with the following SQL code:

```
CREATE TABLE demo_timestamp (
    tempo TIMESTAMP
);
```

A callout bubble points to this code with the text: "Inserida apenas a data."

Below it is another 'Worksheet' tab with the following SQL code:

```
21
22 INSERT INTO demo_timestamp VALUES ('2016-out-27');
23
```

At the bottom, the 'Script Output' tab displays the result of the insert query:

TEMPO
1 16.10.27 00:00:00,000000000

A callout bubble points to the output with the text: "Armazenado o tempo ‘meia-noite’".

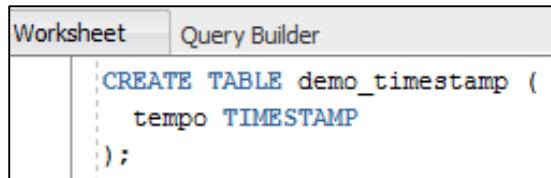
At the very bottom, a callout bubble points to the timestamp value in the output with the text: "Visualização com o formato por omissão:".

Below the output, the system variable `NLS_TIMESTAMP_FORMAT` is shown with its value: `RR.MM.DD HH24:MI:SSXFF`.

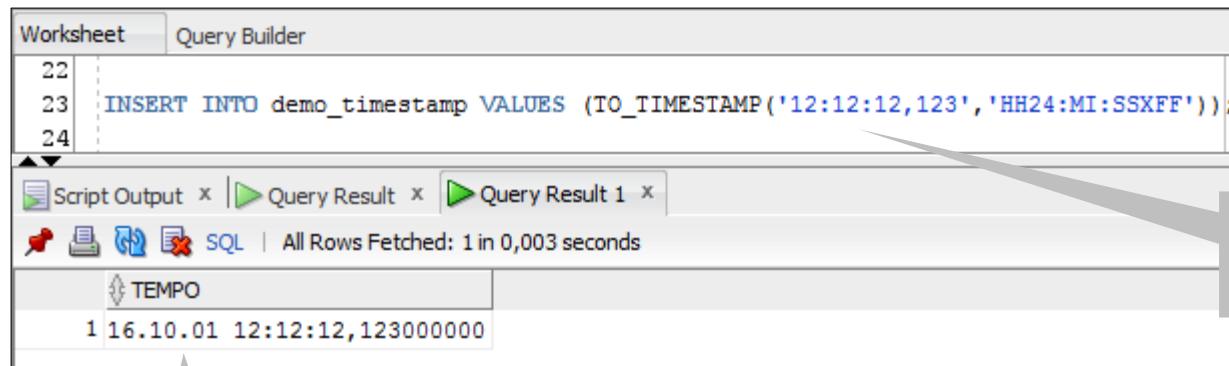
■ Valores TIMESTAMP Armazenados (2/2)

- Data por omissão

- 1º dia do mês atual.
- Exemplo



```
CREATE TABLE demo_timestamp (
    tempo TIMESTAMP
);
```



```
22 |
23 | INSERT INTO demo_timestamp VALUES (TO_TIMESTAMP('12:12:12,123','HH24:MI:SSXFF'));
24 |
```

Script Output | Query Result | Query Result 1 |
SQL | All Rows Fetched: 1 in 0,003 seconds

TEMPO
1 16.10.01 12:12:12,123000000

Tempo registrado
em 29-10-2016.

1º dia do mês 10.

- Operações Aritméticas Suportadas

Operando	Operação	DATE	TIMESTAMP	INTERVAL	Numérico (Representa Nº de Dias)
TIMESTAMP	+	Illegal TIMESTAMP + DATE	Illegal TIMESTAMP + TIMESTAMP	$\text{TIMESTAMP} + \text{INTERVAL} =$ TIMESTAMP	$\text{TIMESTAMP} + \text{NºDIAS} =$ DATE -- frações do segundo perdidas
	-	$\text{TIMESTAMP} - \text{DATE} =$ INTERVAL	$\text{TIMESTAMP} - \text{TIMESTAMP} =$ INTERVAL	$\text{TIMESTAMP} - \text{INTERVAL} =$ TIMESTAMP	$\text{TIMESTAMP} - \text{NºDIAS} =$ DATE -- frações do segundo perdidas
	*			Illegal	
	/			Illegal	

■ Subtração de Valores TIMESTAMP

The screenshot shows an Oracle SQL Worksheet interface. The code entered is:

```

1 SELECT  TIMESTAMP'2016-12-11 14:23:12' "Tempo 1",
2        TIMESTAMP'2016-11-10 13:10:05' "Tempo 2",
3        TIMESTAMP'2016-12-11 14:23:12' - TIMESTAMP'2016-11-10 13:10:05' "Tempo 1 - Tempo 2"
4 FROM DUAL;

```

The output window shows the results:

Tempo 1	Tempo 2	Tempo 1 - Tempo 2
16.12.11 14:23:12,0000000000	16.11.10 13:10:05,0000000000	+31 01:13:07.000000

A yellow box highlights the result: **31 dias, 1 hora, 13 minutos e 7 segundos**.

■ Resultado em Nº de Horas

The screenshot shows an Oracle SQL Worksheet interface. The code entered is:

```

6 SELECT  ROUND(EXTRACT(DAY FROM TIMESTAMP'2016-12-11 14:23:12'-TIMESTAMP'2016-11-10 13:10:05')) * 24 +
7      EXTRACT(HOUR FROM TIMESTAMP'2016-12-11 14:23:12'-TIMESTAMP'2016-11-10 13:10:05') +
8      (EXTRACT(MINUTE FROM TIMESTAMP'2016-12-11 14:23:12'-TIMESTAMP'2016-11-10 13:10:05') / 60) +
9      (EXTRACT(SECOND FROM TIMESTAMP'2016-12-11 14:23:12'-TIMESTAMP'2016-11-10 13:10:05') / 3600),1) "Nº de Horas"
10 FROM DUAL;

```

The output window shows the result:

Nº de Horas
745,2

A yellow box highlights the result: **745,2**.

- Combinar Data DATE com Tempo TIMESTAMP

The diagram illustrates the process of combining a DATE and a TIMESTAMP into a single timestamp. It consists of three main components: 'Worksheet' (containing the SQL code), 'Tipo DATE' (highlighted in yellow), 'Concatenação' (highlighted in yellow), and 'Tipo TIMESTAMP' (highlighted in yellow). The 'Worksheet' shows the following SQL code:

```
12 -- Combinar data e tempo
13
14 SELECT TO_TIMESTAMP(TO_CHAR((SELECT MAX(data_) FROM data_tempo_demo), 'YYYY-MM-DD') || ' ' || TO_CHAR((SELECT MAX(tempo)
15 FROM data_tempo_demo), 'HH24:MI:SS'), 'YYYY-MM-DD HH24:MI:SS') FROM DUAL;
16
```

The 'Query Result' tab shows the output of the query:

1	12.12.12 01:04:05,000000000
1	12.12.12 01:04:05,000000000

- Converter Data de Tipo TIMESTAMP em Tipo DATE

- Interesse

- Representação de períodos de tempo.

- Exemplos

Projeto	Data de Início	Duração (Anos-Meses)
1 Pavilhão Desportivo	01-12-2017	1-10
2 Piscina Coberta	10-06-2017	2-6
3 Armazéns	15-02-2017	1-2

Período de Tempo.

Espetáculo	Data	Horário	Duração (H:M)
1 Magia	01-12-2017	21:30	1:30
2 Concerto	10-12-2016	21:00	2:15
3 Folclore	15-12-2016	21:15	1:45

Período de Tempo.

■ Tipos Predefinidos (1/2)

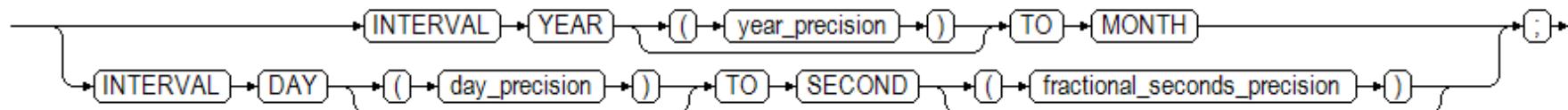
- INTERVAL YEAR TO MONTH -- pode representar intervalo de tempo em anos e meses.
 -- Tamanho fixo: 5 bytes.
- INTERVAL DAY TO SECOND -- pode representar intervalo em dias, horas, minutos e segundos.
 -- Tamanho fixo: 11 bytes.

Tipos Predefinidos = Tipos *Standard*

■ Tipos Predefinidos (2/2)

- Sintaxe da declaração

interval_datatypes_syntax



- Precisão: -- ano, dia ou frações de segundo
 - Nº de dígitos $\in [0,9]$
 - Por omissão:
 - ano = 2
 - dia = 2
 - Frações do segundo = 6
- Exemplos
 - INTERVAL YEAR TO MONTH -- precisão do ano: 2.
 - INTERVAL YEAR(3) TO MONTH -- precisão do ano: 3.
 - INTERVAL DAY TO SECOND -- precisão do dia 2 e das frações do segundo 6.
 - INTERVAL DAY(4) TO SECOND(9) -- precisão do dia 4 e das frações do segundo 9.

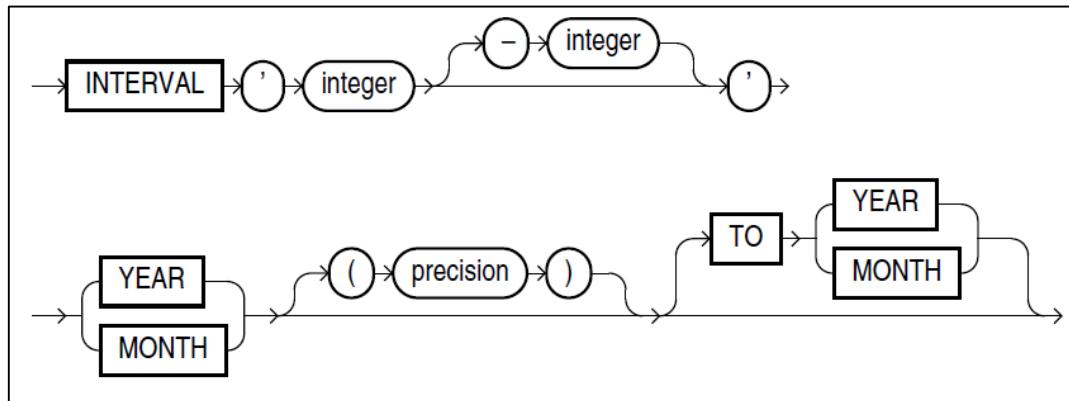
- Valores

- Constituídos por campos data-tempo:
 - YEAR e MONTH -- valores INTERVAL YEAR TO MONTH
 - DAY, HOUR, MINUTE e SECOND -- valores INTERVAL DAY TO SECOND
- Válidos

Campo Data-Tempo	Valores Válidos
YEAR	Qualquer valor inteiro positivo ou negativo.
MONTH	[0, 11]
DAY	Qualquer valor inteiro positivo ou negativo.
HOUR	[0, 23]
MINUTE	[0,59]
SECOND	[0, 59.9(n)] -- n = precisão das frações do segundo

■ Valores INTERVAL YEAR TO MONTH

■ Sintaxe

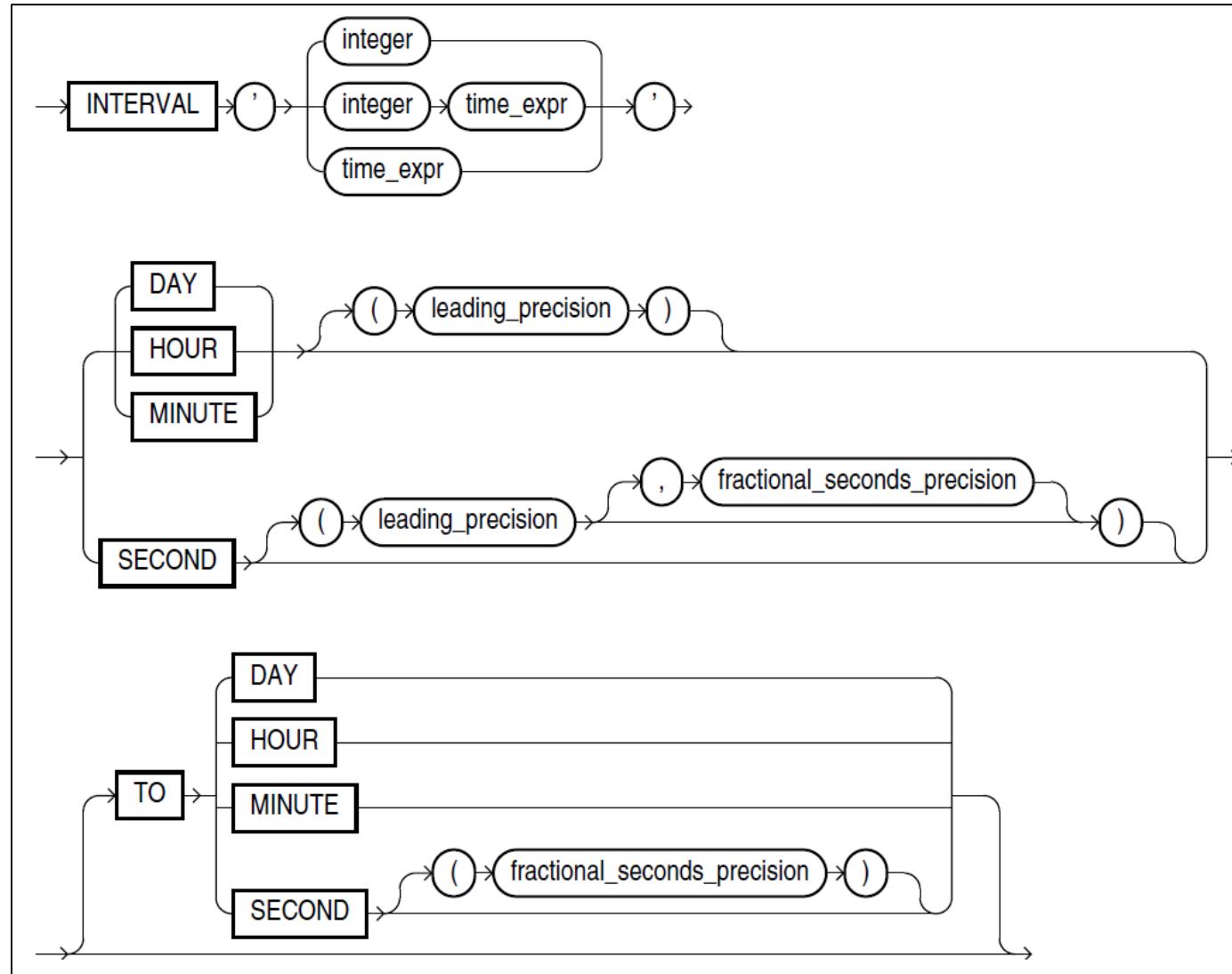


■ Exemplos

Form of Interval Literal	Interpretation
INTERVAL '123-2' YEAR(3) TO MONTH	An interval of 123 years, 2 months. You must specify the leading field precision if it is greater than the default of 2 digits.
INTERVAL '123' YEAR(3)	An interval of 123 years 0 months.
INTERVAL '300' MONTH(3)	An interval of 300 months.
INTERVAL '4' YEAR	Maps to INTERVAL '4-0' YEAR TO MONTH and indicates 4 years.
INTERVAL '50' MONTH	Maps to INTERVAL '4-2' YEAR TO MONTH and indicates 50 months or 4 years 2 months.
INTERVAL '123' YEAR	Returns an error, because the default precision is 2, and '123' has 3 digits.

▪ Valores INTERVAL DAY TO SECOND (1/2)

▪ Sintaxe



■ Valores INTERVAL DAY TO SECOND (2/2)

■ Exemplos

Form of Interval Literal	Interpretation
INTERVAL '4 5:12:10.222' DAY TO SECOND(3)	4 days, 5 hours, 12 minutes, 10 seconds, and 222 thousandths of a second.
INTERVAL '4 5:12' DAY TO MINUTE	4 days, 5 hours and 12 minutes.
INTERVAL '400 5' DAY(3) TO HOUR	400 days 5 hours.
INTERVAL '400' DAY(3)	400 days.
INTERVAL '11:12:10.2222222' HOUR TO SECOND(7)	11 hours, 12 minutes, and 10.2222222 seconds.
INTERVAL '11:20' HOUR TO MINUTE	11 hours and 20 minutes.
INTERVAL '10' HOUR	10 hours.
INTERVAL '10:22' MINUTE TO SECOND	10 minutes 22 seconds.
INTERVAL '10' MINUTE	10 minutes.
INTERVAL '4' DAY	4 days.
INTERVAL '25' HOUR	25 hours.
INTERVAL '40' MINUTE	40 minutes.
INTERVAL '120' HOUR(3)	120 hours.
INTERVAL '30.12345' SECOND(2,4)	30.1235 seconds. The fractional second '12345' is rounded to '1235' because the precision is 4.

■ Apresentar Campos do Intervalo (1/2)

- Concatenar combinações de funções de caracteres.
 - Exemplo:
 - EXTRACT -- extrai campos de um valor de tipo intervalo data-tempo.
- Exemplo 1
 - INTERVAL DAY TO SECOND

```
SELECT order_id, EXTRACT(DAY FROM (SYSDATE - order_date) DAY TO SECOND)
      || ' days '
      || EXTRACT(HOUR FROM (SYSDATE - order_date) DAY TO SECOND)
      || ' hours' "Interval"
FROM orders;

ORDER_ID Interval
-----
 2458 780 days 23 hours
 2397 685 days 22 hours
 2454 733 days 21 hours
 2354 447 days 20 hours
 2358 635 days 20 hours
 2381 508 days 18 hours
 2440 765 days 17 hours
 2357 1365 days 16 hours
 2394 602 days 15 hours
 2435 763 days 15 hours
 . . .
```

- Apresentar Campos do Intervalo (2/2)

- Exemplo 2 - INTERVAL YEAR TO MONTH

```
SELECT last_name, EXTRACT(YEAR FROM (SYSDATE - hire_date) YEAR TO MONTH)
    || ' years '
    || EXTRACT(MONTH FROM (SYSDATE - hire_date) YEAR TO MONTH)
    || ' months' "Interval"
FROM employees;
```

LAST_NAME	Interval
OConnell	2 years 3 months
Grant	1 years 9 months
Whalen	6 years 1 months
Hartstein	5 years 8 months
Fay	4 years 2 months
Mavris	7 years 4 months
Baer	7 years 4 months
Higgins	7 years 4 months
Gietz	7 years 4 months
...	

- Operações Aritméticas Suportadas (1/7)

Operando	Operação	DATE	TIMESTAMP	INTERVAL	Numérico
INTERVAL	+	$DATE + INTERVAL = DATE$	$TIMESTAMP + INTERVAL = TIMESTAMP$	$INTERVAL + INTERVAL = INTERVAL$	illegal
	-	illegal	illegal	$INTERVAL - INTERVAL = INTERVAL$	illegal
	*	illegal	illegal	illegal	$INTERVAL * NÚMERO = INTERVAL$
	/	illegal	illegal	illegal	$INTERVAL / NÚMERO = INTERVAL$

■ Operações Aritméticas Suportadas (2/7)

■ Exemplo 1

The screenshot shows the Oracle SQL Worksheet interface. The top tab bar has 'Worksheet' and 'Query Builder' tabs, with 'Worksheet' selected. The main area contains the following SQL code:

```
SELECT DATE '2016-11-10' + INTERVAL '20' DAY
FROM DUAL;
```

Below the code, there are two tabs: 'Script Output' and 'Query Result'. The 'Query Result' tab is selected, showing the output of the query:

	DATE'2016-11-10'+INTERVAL'20'DAY
1	16.11.30

Below the table, a status message reads: 'All Rows Fetched: 1 in 0,001 seconds'.

■ Operações Aritméticas Suportadas (3/7)

■ Exemplo 2

The screenshot shows the Oracle SQL Developer interface. The top tab bar has 'Worksheet' selected. In the worksheet pane, the following SQL code is entered:

```
SELECT TO_TIMESTAMP('17:15','HH24:MI') + INTERVAL '5:10' HOUR TO MINUTE
FROM DUAL;
```

Below the worksheet, the 'Query Result' tab is active, showing the output of the query:

	TO_TIMESTAMP('17:15','HH24:MI')+INTERVAL'5:10'HOURTOMINUTE
1	16.11.01 22:25:00,000000000

The status bar at the bottom indicates "All Rows Fetched: 1 in 0,001 seconds".

- Operações Aritméticas Suportadas (4/7)

- Exemplo 3

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there's a small red box highlighting the 'Worksheet' tab. The main area displays a SQL query:

```
SELECT INTERVAL '5-3' YEAR TO MONTH + INTERVAL '20' MONTH  
FROM DUAL;
```

In the bottom-right corner of the screenshot, there's a yellow box containing the text "6 anos e 11 meses".

Below the query, the results are shown in the 'Query Result' tab:

INTERVAL'5-3'YEARTOMONTH+INTERVAL'20'MONTH
1 +06-11

The result row contains the value "1 +06-11".

6 anos e 11 meses

■ Operações Aritméticas Suportadas (5/7)

■ Exemplo 4

The screenshot shows the Oracle SQL Developer interface. In the top-left tab, 'Worksheet' is selected. Below it, a SQL query is entered:

```
SELECT INTERVAL '20' DAY - INTERVAL '240' HOUR  
FROM DUAL;
```

In the bottom panel, the 'Query Result' tab is active. The output shows the result of the subtraction:

INTERVAL'20'DAY-INTERVAL'240'HOUR
1 +10 00:00:00.000000

A callout box with a grey arrow points from the text "10 dias" at the bottom left to the "1 +10" part of the result table.

10 dias

▪ Operações Aritméticas Suportadas (6/7)

▪ Exemplo 5

The screenshot shows the Oracle SQL Developer interface. In the Worksheet tab, a query is entered:

```
SELECT INTERVAL '20 5' DAY TO HOUR * 8  
FROM DUAL;
```

In the Query Result tab, the output is displayed:

	INTERVAL'205'DAYTOHOUR*
1	+161 16:00:00.000000

The result is annotated with a yellow box at the bottom left containing the text "161 dias e 16 horas".

161 dias e 16 horas

■ Operações Aritméticas Suportadas (7/7)

■ Exemplo 6

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a 'Worksheet' tab with the following SQL code:

```
SELECT INTERVAL '20' DAY / 5
FROM DUAL;
```

In the bottom-right pane, there is a 'Query Result' tab showing the output of the query. The output is a single row with the value: `INTERVAL'20'DAY/5`. The value is displayed as `1 +04 00:00:00.000000`.

4 dias

Tipos de Dados Oracle

Tipos Numéricos

Tipos Caráter

Tipos Data-Tempo e Intervalo

Precedência

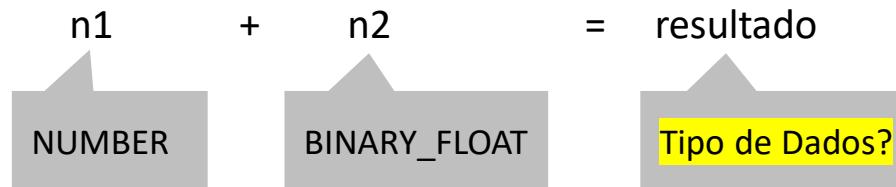
Conversão

- [Interesse](#)
- [Precedência](#)
 - [Geral](#)
 - [Tipos Numéricos](#)

■ Para Oracle

- Avaliar expressões com diferentes tipos de dados.

- Exemplo



■ Solução Oracle

1. Determina o tipo de dados dos operandos com maior precedência. -- precedência = preferência.

- Exemplo:

- BINARY_FLOAT

2. Converte implicitamente os outros tipos de dados para esse tipo de dados (maior precedência).

- Exemplo:

- NUMBER convertido para BINARY_FLOAT

3. Avalia a expressão.

- Resultado é do tipo com maior precedência.

- Exemplo:

- Resultado do tipo BINARY_FLOAT

Precedência	Tipo de Dados
Maior	Data-Tempo e Intervalo
	Numéricos
	Caráter
Menor	Outros Tipos

Precedência	Tipo de Dados
Maior	BINARY_DOUBLE
	BINARY_FLOAT
Menor	NUMBER

■ Expressão sobre Múltiplos Valores Numéricos

- Com operando(s) BINARY_DOUBLE:
 - Oracle tenta:
 - Conversão implícita de todos os operandos ... para BINARY_DOUBLE.
 - Antes da execução da operação.
- Sem operando BINARY_DOUBLE:
 - Com operando(s) BINARY_FLOAT:
 - Oracle tenta:
 - Conversão implícita de todos os operandos ... para BINARY_FLOAT.
 - Antes da execução da operação.
 - Sem operando(s) BINARY_FLOAT:
 - Oracle tenta:
 - Conversão implícita de todos os operandos ... para NUMBER.
 - Antes da execução da operação.

Tipos de Dados Oracle

Tipos Numéricos

Tipos Caráter

Tipos Data-Tempo e Intervalo

Precedências

Conversão

- [Introdução](#)
- [Conversão Implícita](#)
- [Conversão Explicita](#)

- Oracle Suporta

- Conversão Implicita -- conversão automática de valor de um tipo de dados para outro tipo.
- Conversão Explícita -- conversão especificada através de funções de conversão SQL.

- Oracle Recomenda

- Conversão Explícita

- Razões:

- Código SQL mais comprehensível.

- Melhor desempenho.

- Conversão implícita:

- Depende do contexto de execução.

- Pode não funcionar sempre da mesma forma.

- Exemplo:

- Conversão de valor data_tempo → VARCHAR2

- Retorno pode ser inesperado ... porque depende do valor do parâmetro NLS_DATE_FORMAT.

- Algoritmos da conversão implícita:

- Podem ser diferentes entre produtos Oracle.

- Pode haver diferenças entre versões do mesmo produto Oracle.

- Feita Automaticamente pelo Oracle

- Quando faz sentido. -- p.ex., conversão de valor VARCHAR para NUMBER se for um número.

- Conversões Possíveis:

	CHAR	VARCHAR2	DATE	TIMESTAMP / INTERVAL	NUMBER	BINARY_FLOAT	BINARY_DOUBLE
CHAR	✓	✓	✓	✓	✓	✓	✓
VARCHAR2	✓		✓	✓	✓	✓	✓
DATE	✓	✓					
TIMESTAMP / INTERVAL	✓	✓					
NUMBER	✓	✓				✓	✓
BINARY_FLOAT	✓	✓			✓		✓
BINARY_DOUBLE	✓	✓			✓	✓	

- Regras de Conversão:

- Argumento de função/operador de tipo inválido:
 - Convertido para tipo válido.
- Em operações de INSERT e UPDATE:
 - Valor é convertido para o tipo de dados da coluna alterada.
- Na comparação de valor caráter com:
 - Valor numérico: valor caráter é convertido para numérico.
 - Valor DATE: valor caráter é convertido para DATE.
- Conversões:
 - Entre valores caráter/NUMBER e números *floating-point*:
 - Podem ser imprecisas ... porque:
 - Tipos caráter e NUMBER usam precisão decimal para representar o valor numérico.
 - Tipos *floating-point* usam precisão binária.
 - BINARY_FLOAT → BINARY_DOUBLE
 - São exatas.
 - BINARY_DOUBLE → BINARY_FLOAT:
 - Imprecisa apenas se valor BINARY_DOUBLE usar mais bits de precisão do que os suportados pelo BINARY_FLOAT.

- Especificada

- Usando funções de conversão SQL

de	CHAR VARCHAR2	NUMBER	Data-Tempo Intervalo	BINARY_FLOAT	BINARY_DOUBLE
CHAR VARCHAR2	TO_CHAR (char)	TO_NUMBER	TO_DATE TO_TIMESTAMP TO_TIMESTAMP_TZ TO_YMINTERVAL TO_DSINTERVAL	TO_BINARY_FLOAT	TO_BINARY_DOUBLE
NUMBER	TO_CHAR (number)		TO_DATE NUMTOYMINTEGER NUMTODSINTERVAL	TO_BINARY_FLOAT	TO_BINARY_DOUBLE
Data-Tempo Intervalo	TO_CHAR (date)				
BINARY_FLOAT	TO_CHAR (number)	TO_NUMBER		TO_BINARY_FLOAT	TO_BINARY_DOUBLE
BINARY_DOUBLE	TO_CHAR (number)	TO_NUMBER		TO_BINARY_FLOAT	TO_BINARY_DOUBLE

Elementos Básicos

Tipos de Dados

Modelos de Formato

Valor NULL

Comentários

Objetos de Esquema de BD

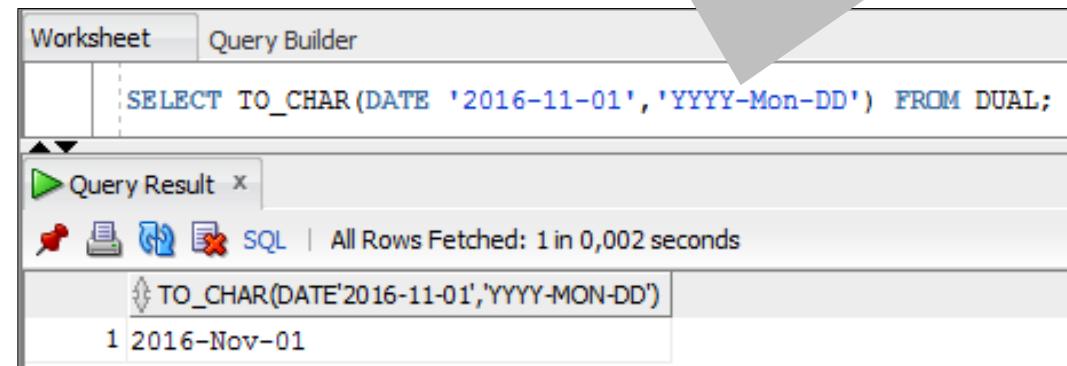
- [Introdução](#)
- [Modelos de Formato](#)
 - [Numérico](#)
 - [Data-Tempo](#)
- [Modificadores de Modelos de Formato](#)
- [Regras de Conversão String-para-Data](#)

Introdução

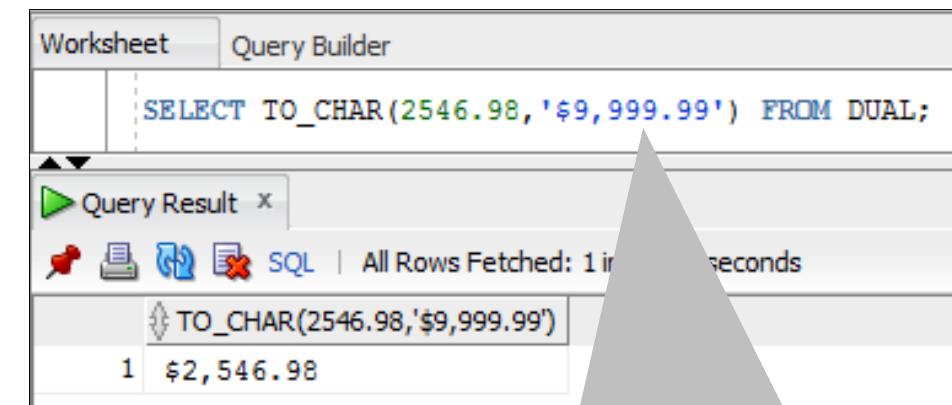
Um Modelo de Formato

- É um valor ... tipo caráter.
 - Exemplo: 'YYYY-MM-DD'
- Constituído por:
 - 1/+ elementos de formato.
 - Exemplos: YYYY, MM e DD
- Descreve o formato de um valor:
 - Data-tempo / numérico.
 - Armazenado numa string.
 - Exemplo: '2016-11-01'
- Interesse ... é para o Oracle:
 - Interpretar o valor na string.
 - Formatar um valor retornado pela BD.
 - Não modifica a representação interna do valor guardado na BD.

Modelo de formato **data-tempo** que descreve o formato do valor **DATE** especificado através dos elementos: YYYY, Mon, DD e hifen(-).



```
Worksheet Query Builder
SELECT TO_CHAR(DATE '2016-11-01', 'YYYY-Mon-DD') FROM DUAL;
Query Result
SQL | All Rows Fetched: 1 in 0,002 seconds
TO_CHAR(DATE'2016-11-01','YYYY-MON-DD')
1 2016-Nov-01
```



```
Worksheet Query Builder
SELECT TO_CHAR(2546.98, '$9,999.99') FROM DUAL;
Query Result
SQL | All Rows Fetched: 1 in 0,002 seconds
TO_CHAR(2546.98,'$9,999.99')
1 $2,546.98
```

Modelo de formato **numérico** que descreve o formato do valor **numérico** especificado através dos elementos: \$, 9, vírgula e ponto.

- Usados nas Funções ... de Conversão

Função	Tipo Convertido	Tipo Retornado
TO_CHAR	NUMBER BINARY_FLOAT BINARY_DOUBLE	VARCHAR2
TO_NUMBER	CHAR VARCHAR2 BINARY_DOUBLE	NUMBER
TO_BINARY_FLOAT	CHAR VARCHAR2	BINARY_FLOAT
TO_BINARY_DOUBLE	CHAR VARCHAR2	BINARY_DOUBLE

Worksheet Query Builder

```
SELECT TO_NUMBER('123.45','999.99') FROM DUAL;
```

Query Result

TO_NUMBER('123.45','999.99')
123,45

Modelo de formato descreve o formato do número especificado numa string.

Worksheet Query Builder

```
SELECT TO_CHAR(123.45,'999.9') FROM DUAL;
```

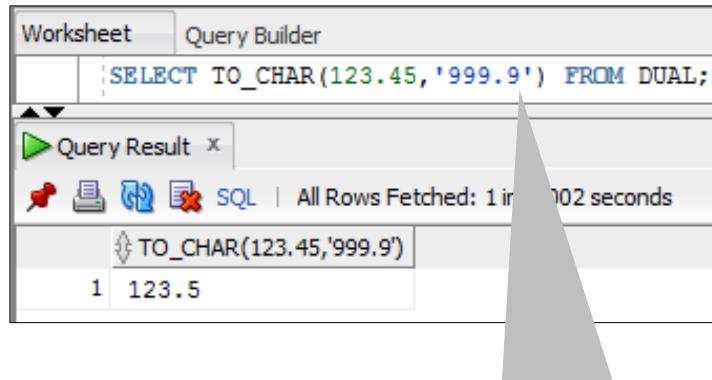
Query Result

TO_CHAR(123.45,'999.9')
123.5

Modelo de formato arredonda o número especificado.

▪ Arredondam Número Especificado

- Para o nº de dígitos significativos especificado.



Worksheet Query Builder

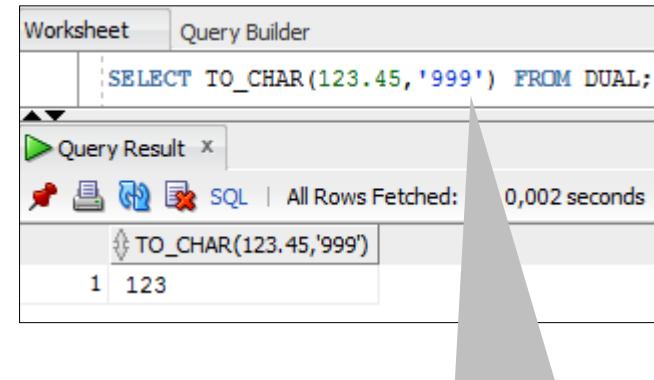
```
SELECT TO_CHAR(123.45, '999.9') FROM DUAL;
```

Query Result x

SQL | All Rows Fetched: 1 in 0,002 seconds

	TO_CHAR(123.45,'999.9')
1	123,5

Arredondamento para
uma casa decimal.



Worksheet Query Builder

```
SELECT TO_CHAR(123.45, '999') FROM DUAL;
```

Query Result x

SQL | All Rows Fetched: 1 in 0,002 seconds

	TO_CHAR(123.45,'999')
1	123

Arredondamento
sem casas decimais.

Número Especificado

- Substituído por sinais # ... quando:
 - NDSE do nº especificado > NDSE definido no modelo de formato.

The screenshot shows the Oracle SQL Worksheet interface. In the 'Worksheet' tab, a query is entered: `SELECT TO_CHAR(123.45, '99') FROM DUAL;`. In the 'Query Result' tab, the output is displayed: `TO_CHAR(123.45,'99')` and the result row shows `1 ###`. Two red arrows point from the text 'Substituído por sinais # ... quando:' to the '99' placeholder in the query and the three '#' characters in the result.

NDSE

- nº de dígitos significativos à esquerda do separador decimal.

■ Valor NUMBER

- Positivo e **extremamente** grande: valor substituído pelo sinal de infinito **~**
- Negativo e **extremamente** pequeno: valor substituído pelo sinal de infinito negativo **-~**

■ Valor BINARY_FLOAT / BINARY_DOUBLE

- É convertido para CHAR.
- Valores infinitos / NaN:
 - Com modelo de formato:
 - Especificado: valor substituído por sinais #.
 - Não-especificado: string Inf / NaN.

Modelo de formato especificado.

```
Worksheet Query Builder
SELECT TO_CHAR(2f/0, '999') FROM DUAL;

Query Result
SQL | All Rows Fetched: 1 in 0,001 seconds
TO_CHAR(2F/0,'999')
1 #####
```

Modelo de formato não-especificado.

```
Worksheet Query Builder
SELECT TO_CHAR(2f/0) FROM DUAL;

Query Result
SQL | All Rows Fetched: 1 in 0,001
TO_CHAR(2F/0)
1 Inf
```

Modelo de formato especificado.

```
Worksheet Query Builder
SELECT TO_CHAR(REMAINDER(2f,0), '999999') FROM DUAL;

Query Result
SQL | All Rows Fetched: 1 in 0,003 seconds
TO_CHAR(REMAINDER(2F,0),'999999')
1 #####
```

Modelo de formato não-especificado.

```
Worksheet Query Builder
SELECT TO_CHAR(REMAINDER(2f,0)) FROM DUAL;

Query Result
SQL | All Rows Fetched: 1 in 0 seconds
TO_CHAR(REMAINDER(2F,0))
1 Nan
```

■ Elementos de Formato Numérico (1/4)

Element	Example	Description	
,	,999	Returns a comma in the specified position. You can specify multiple commas in a number format model. Restrictions: <ul style="list-style-type: none">■ A comma element cannot begin a number format model.■ A comma cannot appear to the right of a decimal character or period in a number format model.	
.	.99	Returns a decimal point, which is a period (.) in the specified position. Restriction: You can specify only one period in a number format model.	
\$	\$9999	Returns value with a leading dollar sign.	
0	0999	Returns leading zeros. 9990	Returns trailing zeros.
9	9999	Returns value with the specified number of digits with a leading space if positive or with a leading minus if negative. Leading zeros are blank, except for a zero value, which returns a zero for the integer part of the fixed-point number.	
B	B9999	Returns blanks for the integer part of a fixed-point number when the integer part is zero (regardless of zeros in the format model).	
C	C999	Returns in the specified position the ISO currency symbol (the current value of the NLS_ISO_CURRENCY parameter).	
D	99D99	Returns in the specified position the decimal character, which is the current value of the NLS_NUMERIC_CHARACTER parameter. The default is a period (.). Restriction: You can specify only one decimal character in a number format model.	

■ Elementos de Formato Numérico (2/4)

Element	Example	Description
EEEE	9.9EEEE	Returns a value using in scientific notation.
G	9G999	Returns in the specified position the group separator (the current value of the NLS_NUMERIC_CHARACTER parameter). You can specify multiple group separators in a number format model. Restriction: A group separator cannot appear to the right of a decimal character or period in a number format model.
L	L999	Returns in the specified position the local currency symbol (the current value of the NLS_CURRENCY parameter).
MI	9999MI	Returns negative value with a trailing minus sign (-). Returns positive value with a trailing blank. Restriction: The MI format element can appear only in the last position of a number format model.
PR	9999PR	Returns negative value in <angle brackets>. Returns positive value with a leading and trailing blank. Restriction: The PR format element can appear only in the last position of a number format model.
RN	RN	Returns a value as Roman numerals in uppercase.
rn	rn	Returns a value as Roman numerals in lowercase. Value can be an integer between 1 and 3999.

■ Elementos de Formato Numérico (3/4)

Element	Example	Description
S	s9999 9999s	Returns negative value with a leading minus sign (-). Returns positive value with a leading plus sign (+). Returns negative value with a trailing minus sign (-). Returns positive value with a trailing plus sign (+). Restriction: The S format element can appear only in the first or last position of a number format model.
TM	TM	The text minimum number format model returns (in decimal output) the smallest number of characters possible. This element is case insensitive. The default is TM9, which returns the number in fixed notation unless the output exceeds 64 characters. If the output exceeds 64 characters, then Oracle Database automatically returns the number in scientific notation. Restrictions: <ul style="list-style-type: none">■ You cannot precede this element with any other element.■ You can follow this element only with one 9 or one E (or e), but not with any combination of these. The following statement returns an error: <code>SELECT TO_CHAR(1234, 'TM9e') FROM DUAL;</code>

■ Elementos de Formato Numérico (4/4)

Element	Example	Description
U	U9999	Returns in the specified position the Euro (or other) dual currency symbol, determined by the current value of the NLS_DUAL_CURRENCY parameter.
V	999V99	Returns a value multiplied by 10^n (and if necessary, round it up), where n is the number of 9's after the V.
X	xxxx xxxx	Returns the hexadecimal value of the specified number of digits. If the specified number is not an integer, then Oracle Database rounds it to an integer. Restrictions: <ul style="list-style-type: none">■ This element accepts only positive values or 0. Negative values return an error.■ You can precede this element only with 0 (which returns leading zeroes) or FM. Any other elements return an error. If you specify neither 0 nor FM with X, then the return always has one leading blank. Refer to the format model modifier FM on page 2-68 for more information.

■ Resultados de Conversões

modelo de formato

number	'fmt'	Result
-1234567890	9999999999S	'1234567890-'
0	99.99	' .00'
+0.1	99.99	' .10'
-0.2	99.99	' -.20'
0	90.99	' 0.00'
+0.1	90.99	' 0.10'
-0.2	90.99	' -0.20'
0	9999	' 0'
1	9999	' 1'
0	B9999	' '
1	B9999	' 1'
0	B90.99	' '
+123.456	999.999	' 123.456'
-123.456	999.999	' -123.456'
+123.456	FM999.009	'123.456'
+123.456	9.9EEEE	' 1.2E+02'
+1E+123	9.9EEEE	' 1.0E+123'

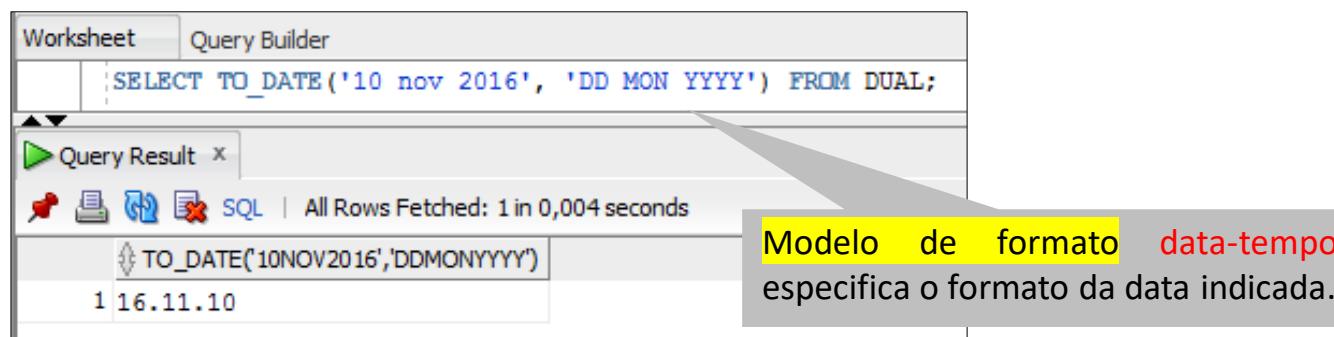
■ Resultados de Conversões

modelo de formato

number	'fmt'	Result
+123.456	FM9.9EEEE	'1.2E+02'
+123.45	FM999.009	'123.45'
+123.0	FM999.009	'123.00'
+123.45	L999.99	'\$123.45'
+123.45	FML999.99	'\$123.45'
+1234567890	9999999999S	'1234567890+'

■ Usados nas Funções de Conversão

Função de Conversão	Tipo Retornado	Formato por Omissão Especificado em	Converte
TO_DATE	DATE	NLS_DATE_FORMAT	Valor do tipo caráter, num formato diferente do formato por omissão, num tipo data-tempo.
TO_TIMESTAMP	TIMESTAMP	NLS_TIMESTAMP_FORMAT	
TO_TIMESTAMP_TZ	TIMESTAMP WITH TIME ZONE	NLS_TIMESTAMP_TZ_FORMAT	
TO_CHAR	VARCHAR2		Valor data-tempo num valor caráter que está num formato diferente do formato por omissão.



■ Comprimento do modelo de formato

- Nº máximo de carateres <= 22

■ Elementos de Formato Data-Tempo (1/18)

- Letras Maiúsculas e Minúsculas:
 - Refletidas no resultado.
 - Exemplos:
 - MONTH ⇒ NOVEMBRO
 - Month ⇒ Novembro
 - month ⇒ novembro
- Abreviatura:
 - Refletida no resultado.
 - Exemplos:
 - MON ⇒ NOV
 - Mon ⇒ Nov
 - Mon ⇒ nov

Worksheet Query Builder

```
SELECT TO_CHAR(DATE '2016-11-05','YYYY-MONTH-DD') FROM DUAL;
```

Query Result x

SQL | All Rows Fetched: 1 in 0,002 seconds

TO_CHAR(DATE'2016-11-05','YYYY-MONTH-DD')
1 2016-NOVEMBRO -05

Worksheet Query Builder

```
SELECT TO_CHAR(DATE '2016-11-05','YYYY-Month-DD') FROM DUAL;
```

Query Result x

SQL | All Rows Fetched: 1 in 0,004 seconds

TO_CHAR(DATE'2016-11-05','YYYY-Month-DD')
1 2016-Novembro -05

Worksheet Query Builder

```
SELECT TO_CHAR(DATE '2016-11-05','YYYY-month-DD') FROM DUAL;
```

Query Result x

SQL | All Rows Fetched: 1 in 0,002 seconds

TO_CHAR(DATE'2016-11-05','YYYY-month-DD')
1 2016-novembro -05

Worksheet Query Builder

```
SELECT TO_CHAR(DATE '2016-11-05','YYYY-mon-DD') FROM DUAL;
```

Query Result x

SQL | All Rows Fetched: 1 in 0,003 seconds

TO_CHAR(DATE'2016-11-05','YYYY-mon-DD')
1 2016-nov-05

■ Elementos de Formato Data-Tempo (2/18)

- Podem ser:

- Caracteres de pontuação:

- - -- hífen.
 - /
 - ,
 - .
 - ;
 - :

- Valores de caracteres:

- Entre aspas.

Carateres de pontuação diferentes

The screenshot shows the Oracle SQL Developer interface. In the Worksheet tab, six queries are listed, each using a different separator in the TO_DATE function's format string. All queries return the same result in the Query Result tab: TO_DATE('2016:11:05', 'RR.MM.DD') which outputs 16.11.05.

```

SELECT TO_DATE('2016-11-05','YYYY-MM-DD') FROM DUAL;
SELECT TO_DATE('2016/11/05','YYYY/MM/DD') FROM DUAL;
SELECT TO_DATE('2016,11,05','YYYY,MM,DD') FROM DUAL;
SELECT TO_DATE('2016.11.05','YYYY.MM.DD') FROM DUAL;
SELECT TO_DATE('2016;11;05','YYYY;MM;DD') FROM DUAL;
SELECT TO_DATE('2016:11:05','YYYY:MM:DD') FROM DUAL;
    
```

Resultado:

- Igual para todas as queries.

Formato por omissão:

- RR.MM.DD

The screenshot shows the Oracle SQL Developer interface. A query uses the TO_CHAR function with a format model 'DD "de" Month"de" YYYY'. The result in the Query Result tab is '05 de Novembro de 2016'.

```

SELECT TO_CHAR(TO_DATE('2016-11-05','YYYY-MM-DD'), 'DD "de" Month"de" YYYY') FROM DUAL;
    
```

string "de" como elemento de formato.

Modelos de Formato Data-Tempo

Elementos de Formato Data-Tempo (3/18)

Caracteres de pontuação

- Na conversão de *strings* para datas:
 - São flexíveis
 - Exemplo: Função **TO_DATE**
 - Modelo de formato ... com carateres de pontuação (Ex: **YYYY-MM-DD**):
 - Corresponde:
 - A várias *strings* de datas.
 - Ex: **2016-11-05, 2016.11.5**
 - Os carateres de pontuação (Ex: **-**):
 - Podem ser substituídos por ... carateres não-alfanuméricos.
 - Exemplos: espaços, #, %, &, \$, ...
 - Os elementos numéricos da *string* data (Ex: **2016**):
 - Nº de dígitos <= nº de dígitos especificados no modelo de formato.
 - Exemplos:
 - **YYYY** especifica 4 dígitos no máximo.
 - **MM** especifica 2 dígitos no máximo.

```

Worksheet
SELECT TO_DATE('2016-11-05','YYYY-MM-DD') FROM DUAL;
SELECT TO_DATE('2016 11 05','YYYY-MM-DD') FROM DUAL;
SELECT TO_DATE('2016.11.05','YYYY-MM-DD') FROM DUAL;
SELECT TO_DATE('2016.11.5','YYYY-MM-DD') FROM DUAL;
SELECT TO_DATE('16.11.5','YYYY-MM-DD') FROM DUAL;

Query Result
SQL | All Results Retrieved: 1 in 0,002 seconds
TO_DATE('2016-11-05','YYYY-MM-DD')
1 16.11.05
  
```

Expressões TO_DATE equivalentes.

- Modelo de formato: **YYYY-MM-DD**
- *Strings* data com vários formatos.

■ Elementos de Formato Data-Tempo (4/18)

- Alguns elementos:

- Dependem do:

- País
- Linguagem

} usada pela BD.

- Exemplo:

- Elementos de formato data-tempo:

- MONTH
- MON
- DAY
- DY
- BC / AD / B.C. / A.D.
- AM / PM / A.M. / P.M.

} Retornam valores por extenso.

Nome do mês:

- Em português.
- É a linguagem por omissão.
- Especificada em NLS_DATE_LANGUAGE.

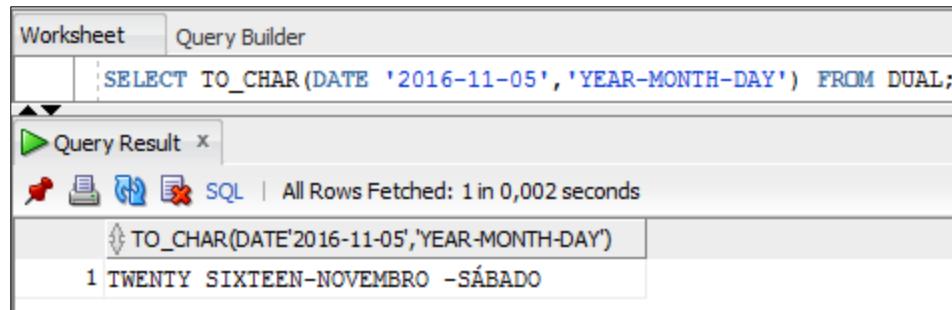
```
Worksheet Query Builder
SELECT TO_CHAR(DATE '2016-11-05', 'YYYY-MONTH-DD') FROM DUAL;
Query Result x
SQL | All Rows Fetched: 1 in 0,002 seconds
TO_CHAR(DATE'2016-11-05','YYYY-MONTH-DD')
1 2016-NOVEMBRO -05
```

6	NLS_DATE_LANGUAGE	PORTUGUESE
---	-------------------	------------

■ Elementos de Formato Data-Tempo (5/18)

■ Elementos:

- YEAR
 - SYEAR
- } retornam valores em inglês (sempre).



The screenshot shows the Oracle SQL Worksheet interface. In the 'Worksheet' tab, a query is entered:

```
SELECT TO_CHAR(DATE '2016-11-05','YEAR-MONTH-DAY') FROM DUAL;
```

In the 'Query Result' tab, the output is displayed:

TO_CHAR(DATE'2016-11-05','YEAR-MONTH-DAY')
1 TWENTY SIXTEEN-NOVEMBER -SÁBADO

The result shows the date '2016-11-05' formatted as 'TWENTY SIXTEEN-NOVEMBER -SÁBADO'.

- Elementos de Formato Data-Tempo (6/18)

Oracle Recomenda:

- Usar o elemento do ano de 4 dígitos YYYY.
 - Em vez dos elementos do ano mais curtos (YY, Y, ...).
- Razão:
 - Elemento YYYY elimina ambiguidades.
 - Elementos do ano mais curtos:
 - Afetam a otimização de *queries* ... porque o ano:
 - É desconhecido em tempo de compilação da *query*.
 - Só pode ser determinado em *runtime*.

■ Elementos de Formato Data-Tempo (7/18)

Element	TO_* datetime functions?	Description
-	Yes	Punctuation and quoted text is reproduced in the result.
/		
,		
.		
;		
:		
"text"		
AD	Yes	AD indicator with or without periods.
A.D.		
AM	Yes	Meridian indicator with or without periods.
A.M.		
BC	Yes	BC indicator with or without periods.
B.C.		

■ Elementos de Formato Data-Tempo (8/18)

Element	TO_ * datetime functions?	Description
CC		Century.
SCC		<ul style="list-style-type: none">■ If the last 2 digits of a 4-digit year are between 01 and 99 (inclusive), then the century is one greater than the first 2 digits of that year.■ If the last 2 digits of a 4-digit year are 00, then the century is the same as the first 2 digits of that year. <p>For example, 2002 returns 21; 2000 returns 20.</p>
D	Yes	Day of week (1-7). This element depends on the NLS territory of the session.
DAY	Yes	Name of day.
DD	Yes	Day of month (1-31).
DDD	Yes	Day of year (1-366).

■ Elementos de Formato Data-Tempo (9/18)

Element	TO_ * datetime functions?	Description
DL	Yes	<p>Returns a value in the long date format, which is an extension of the Oracle Database DATE format, determined by the current value of the NLS_DATE_FORMAT parameter. Makes the appearance of the date components (day name, month number, and so forth) depend on the NLS_TERRITORY and NLS_LANGUAGE parameters. For example, in the AMERICAN_AMERICA locale, this is equivalent to specifying the format 'fmDay, Month dd, yyyy'. In the GERMAN_GERMANY locale, it is equivalent to specifying the format 'fmDay, dd. Month yyyy'.</p> <p>Restriction: You can specify this format only with the TS element, separated by white space.</p>
DS	Yes	<p>Returns a value in the short date format. Makes the appearance of the date components (day name, month number, and so forth) depend on the NLS_TERRITORY and NLS_LANGUAGE parameters. For example, in the AMERICAN_AMERICA locale, this is equivalent to specifying the format 'MM/DD/YYYY'. In the ENGLISH_UNITED KINGDOM locale, it is equivalent to specifying the format 'DD/MM/YYYY'.</p> <p>Restriction: You can specify this format only with the TS element, separated by white space.</p>
DY	Yes	Abbreviated name of day.
E	Yes	Abbreviated era name (Japanese Imperial, ROC Official, and Thai Buddha calendars).
EE	Yes	Full era name (Japanese Imperial, ROC Official, and Thai Buddha calendars).

- Elementos de Formato Data-Tempo (10/18)

Element	TO_* datetime functions?	Description
FF [1..9]	Yes	<p>Fractional seconds; no radix character is printed. Use the X format element to add the radix character. Use the numbers 1 to 9 after FF to specify the number of digits in the fractional second portion of the datetime value returned. If you do not specify a digit, then Oracle Database uses the precision specified for the datetime data type or the data type's default precision. Valid in timestamp and interval formats, but not in DATE formats.</p> <p>Examples: 'HH:MI:SS.FF'</p> <pre>SELECT TO_CHAR(SYSTIMESTAMP, 'SS.FF3') from DUAL;</pre>
FM	Yes	<p>Returns a value with no leading or trailing blanks.</p> <p>See Also: FM on page 2-68</p>
FX	Yes	<p>Requires exact matching between the character data and the format model.</p> <p>See Also: FX on page 2-68</p>
HH	Yes	Hour of day (1-12).
HH12		
HH24	Yes	Hour of day (0-23).

■ Elementos de Formato Data-Tempo (11/18)

Element	TO_* datetime functions?	Description
IW		Calendar week of year (1-52 or 1-53), as defined by the ISO 8601 standard. <ul style="list-style-type: none">■ A calendar week starts on Monday.■ The first calendar week of the year includes January 4.■ The first calendar week of the year may include December 29, 30 and 31.■ The last calendar week of the year may include January 1, 2, and 3.
IYYY		4-digit year of the year containing the calendar week, as defined by the ISO 8601 standard.
IYY		Last 3, 2, or 1 digit(s) of the year containing the calendar week, as defined by the ISO 8601 standard.
IY		
I		
J	Yes	Julian day; the number of days since January 1, 4712 BC. Number specified with J must be integers.
MI	Yes	Minute (0-59).
MM	Yes	Month (01-12; January = 01).
MON	Yes	Abbreviated name of month.
MONTH	Yes	Name of month.
PM	Yes	Meridian indicator with or without periods.
P.M.		
Q		Quarter of year (1, 2, 3, 4; January - March = 1).

■ Elementos de Formato Data-Tempo (12/18)

Element	TO_* datetime functions?	Description
RM	Yes	Roman numeral month (I-XII; January = I).
RR	Yes	Lets you store 20th century dates in the 21st century using only two digits. See Also: " The RR Datetime Format Element " on page 2-66
RRRR	Yes	Round year. Accepts either 4-digit or 2-digit input. If 2-digit, provides the same return as RR. If you do not want this functionality, then enter the 4-digit year.
SS	Yes	Second (0-59).
SSSSS	Yes	Seconds past midnight (0-86399).
TS	Yes	Returns a value in the short time format. Makes the appearance of the time components (hour, minutes, and so forth) depend on the NLS_TERRITORY and NLS_LANGUAGE initialization parameters. Restriction: You can specify this format only with the DL or DS element, separated by white space.
TZD	Yes	Daylight saving information. The TZD value is an abbreviated time zone string with daylight saving information. It must correspond with the region specified in TZR. Valid in timestamp and interval formats, but not in DATE formats. Example: PST (for US/Pacific standard time); PDT (for US/Pacific daylight time).
TZH	Yes	Time zone hour. (See TZM format element.) Valid in timestamp and interval formats, but not in DATE formats. Example: 'HH:MI:SS.FFTZH:TZM'.
TZM	Yes	Time zone minute. (See TZH format element.) Valid in timestamp and interval formats, but not in DATE formats. Example: 'HH:MI:SS.FFTZH:TZM'.

■ Elementos de Formato Data-Tempo (13/18)

Element	TO_* datetime functions?	Description
TZR	Yes	Time zone region information. The value must be one of the time zone region names supported in the database. Valid in timestamp and interval formats, but not in DATE formats. Example: US/Pacific
WW		Week of year (1-53) where week 1 starts on the first day of the year and continues to the seventh day of the year.
W		Week of month (1-5) where week 1 starts on the first day of the month and ends on the seventh.
X	Yes	Local radix character. Example: 'HH:MI:SSXFF'.
Y, YYY	Yes	Year with comma in this position.
YEAR		Year, spelled out; \$ prefixes BC dates with a minus sign (-).
SYEAR		
YYYY	Yes	4-digit year; \$ prefixes BC dates with a minus sign.
SYYYY		
YYY	Yes	Last 3, 2, or 1 digit(s) of year.
YY		
Y		

- Elementos de Formato Data-Tempo (14/18)

- Elemento RR (1/4)
 - Semelhante ao elemento de ano YY
 - Mas, mais flexível.
 - Interesse:
 - Armazenar ... datas de outros séculos.
 - Exemplo:
 - Armazenar datas do século 20 no século 21 ...
... especificando apenas os 2 últimos dígitos do ano.

- Elementos de Formato Data-Tempo (15/18)

- Elemento RR (2/4)

- Funcionamento:

- Ano retornado depende simultaneamente:
 - 2 dígitos do ano especificado.
 - 2 últimos dígitos do ano atual.

2 dígitos do ano especificado	2 últimos dígitos do ano atual	Ano retornado com elemento RR	Ano retornado com elemento YY
[00, 49] (10)	[00, 49]	2 primeiros dígitos = do ano atual	2 primeiros dígitos = do ano atual
	(2016)	(2010)	(2010)
[50, 99] (98)	[50, 99]	2 primeiros dígitos = do ano atual + 1	2 primeiros dígitos = do ano atual
	(2060)	(2110)	(2010)
[00, 49] (2016)	[00, 49]	2 primeiros dígitos = do ano atual - 1	2 primeiros dígitos = do ano atual
	(2016)	(1998)	(2098)
[50, 99] (2060)	[50, 99]	2 primeiros dígitos = do ano atual	2 primeiros dígitos = do ano atual
	(2060)	(2098)	(2098)

(Exemplo)

■ Elementos de Formato Data-Tempo (16/18)

■ Elemento RR (3/4)

■ Exemplo 1:

The screenshot shows the Oracle SQL Worksheet interface. In the 'Worksheet' tab, the following SQL query is entered:

```
SELECT TO_CHAR(TO_DATE('06-11-98', 'DD-MM-YY'), 'DD-MM-YYYY') FROM DUAL;
```

The 'Query Result' tab displays the output:

TO_CHAR(TO_DATE('06-11-98','DD-MM-YY'),'DD-MM-YYYY')
1 06-11-2098

A callout box points to the 'YY' placeholder in the format model with the following text:

Elemento de formato YY:
Acrescentados sempre: 2 primeiros dígitos do ano atual do sistema (2016).

The screenshot shows the Oracle SQL Worksheet interface. In the 'Worksheet' tab, the following SQL query is entered:

```
SELECT TO_CHAR(TO_DATE('06-11-98', 'DD-MM-RR'), 'DD-MM-YYYY') FROM DUAL;
```

The 'Query Result' tab displays the output:

TO_CHAR(TO_DATE('06-11-98','DD-MM-RR'),'DD-MM-YYYY')
1 06-11-1998

A callout box points to the 'RR' placeholder in the format model with the following text:

Elemento de formato RR:
Para ano atual entre [00, 49] e ano especificado entre [50, 99], acrescentados: 2 primeiros dígitos do ano atual - 1.

■ Elementos de Formato Data-Tempo (17/18)

■ Elemento RR (4/4)

■ Exemplo 2:

The screenshot shows the Oracle SQL Worksheet interface. In the 'Query Builder' tab, the following SQL code is entered:

```
SELECT TO_CHAR(TO_DATE('06-11-16', 'DD-MM-YY'), 'DD-MM-YYYY') FROM DUAL;
```

The results are displayed in the 'Query Result' tab:

	TO_CHAR(TO_DATE('06-11-16','DD-MM-YY'),'DD-MM-YYYY')
1	06-11-2016

A yellow box highlights the 'YY' in the format string 'DD-MM-YY'. A grey arrow points from this highlighted area to the explanatory text on the right.

Elemento de formato YY:

Acrescentados **sempre**: 2 primeiros dígitos do ano atual do sistema.

The screenshot shows the Oracle SQL Worksheet interface. In the 'Query Builder' tab, the following SQL code is entered:

```
SELECT TO_CHAR(TO_DATE('06-11-16', 'DD-MM-RR'), 'DD-MM-YYYY') FROM DUAL;
```

The results are displayed in the 'Query Result' tab:

	TO_CHAR(TO_DATE('06-11-16','DD-MM-RR'),'DD-MM-YYYY')
1	06-11-2016

A yellow box highlights the 'RR' in the format string 'DD-MM-RR'. A grey arrow points from this highlighted area to the explanatory text on the right.

Elemento de formato RR:

Para **ano atual** entre [00, 49] e **ano especificado** entre [00, 49], acrescentados: 2 primeiros dígitos do ano atual.

■ Elementos de Formato Data-Tempo (18/18)

■ Sufixos

Suffix	Meaning	Example Element	Example Value
TH	Ordinal Number	DDTH	4TH
SP	Spelled Number	DDSP	FOUR
SPTH or THSP	Spelled, ordinal number	DDSPTH	FOURTH

- Valores retornados:
 - Em inglês. -- sempre
- Válidos:
 - Para formatar saída.
- Inválidos:
 - Para inserir datas na BD.
- Exemplo:

The screenshot shows the Oracle SQL Worksheet interface. The 'Worksheet' tab is active, displaying the following SQL query:

```
SELECT TO_CHAR(TO_DATE('06-11-16', 'DD-MM-YYYY'), 'DDTHSP') FROM DUAL;
```

The 'Query Result' pane shows the output of the query:

TO_CHAR(TO_DATE('06-11-16','DD-MM-YYYY'),'DDTHSP')
1 SIXTH

Below the table, it says "All Rows Fetched: 1 in 0,002 seconds".

- Usados
 - Nos modelos de formato.

- Interesse
 - Controlar:
 - Funcionamento dos elementos de formato.
 - Nomeadamente:
 - Adicionar zeros à esquerda ... em elementos numéricos.
 - Adicionar espaços à direita ... em elementos de caracteres.
 - Verificar o formato exato.
 - Exemplo

Worksheet Query Builder

```
SELECT TO_CHAR(TO_DATE('2016.6.5','YYYY-MM-DD'),'DD-MM-YYYY') FROM DUAL;
```

Query Result

All Rows Fetched: 1 in 0,002 seconds

	TO_CHAR(TO_DATE('2016.6.5','YYYY-MM-DD'),'DD-MM-YYYY')
1	05-06-2016

Elementos de formato DD e MM:
Adicionam zeros à esquerda do dia e do mês, especificados.

Modificadores de Modelos de Formato

Elementos Modificadores de Formato

- FM (fill mode) -- controla a adição de **espaços/zeros**.
-- feita por **elementos de formato**.
- FX (format exact) -- controla o **formato exato** dos **elementos de formato**.

The screenshot shows the Oracle SQL Worksheet interface. In the 'Worksheet' tab, a query is entered:

```
SELECT TO_CHAR(TO_DATE('2016.6.5','YYYY-MM-DD'), 'FMDD-MM-YYYY') FROM DUAL;
```

The result is displayed in the 'Query Result' tab:

	TO_CHAR(TO_DATE('2016.6.5','YYYY-MM-DD'), 'FMDD-MM-YYYY')
1	5-6-2016

A callout box points to the 'FMDD-MM-YYYY' part of the query with the text: "Modificador FM: Cancela adição de zeros ao dia, ao mês e ao ano."

The screenshot shows the Oracle SQL Worksheet interface. In the 'Worksheet' tab, a query is entered:

```
SELECT TO_CHAR(TO_DATE('2016-06-05','FXYYYY-MM-DD'), 'YYYY-MM-DD') FROM DUAL;
```

The result is displayed in the 'Query Result' tab:

	TO_CHAR(TO_DATE('2016-06-05','FXYYYY-MM-DD'), 'YYYY-MM-DD')
1	2016-06-05

A callout box points to the 'FXYYYY-MM-DD' part of the query with the text: "Modificador FX: Obriga a especificar uma data com o número máximo de dígitos permitido pelo elemento de formato correspondente, i.e., ano com 4 dígitos, mês com 2 dígitos e dia com 2 dígitos."

Modificadores de Modelos de Formato

Um Modificador

- Pode ser usado:
 - Múltiplas vezes no mesmo modelo de formato.
 - Cada ocorrência subsequente:
 - Cancela o efeito do modificador anterior.
 - Efeito do modificador:
 - Permitido ... a partir da 1º ocorrência.
 - Inibido a partir da 2ª ocorrência.
 - Permitido ... a partir da 3ª ocorrência.
 - Inibido a partir da 4ª ocorrência.
 - E assim sucessivamente.

2º modificador cancela efeito do 1º modificador, permitindo aos elementos de formato seguintes adicionar zeros.

The screenshot shows the Oracle SQL Worksheet interface. The top tab bar has 'Worksheet' and 'Query Builder'. The main area contains the following SQL code:

```
SELECT TO_CHAR(TO_DATE('2016.6.5','YYYY-MM-DD'), 'FMDD-FMMM-YYYY') FROM DUAL;
```

The 'Query Result' tab is selected, showing the output:

	TO_CHAR(TO_DATE('2016.6.5','YYYY-MM-DD'), 'FMDD-FMMM-YYYY')
1	5-06-2016

Below the table, status information says 'All Rows Fetched: 1 in 0,002 seconds'.

1º modificador inibe adição de zeros aos elementos de formato seguintes (DD neste caso).

Modificadores de Modelos de Formato

▪ Modificador FM (Fill Mode)

- Controla (inibe/permite) a adição de:
 - Espaços ... à direita de elementos em *strings* (valores alfanuméricicos).
 - Zeros à esquerda de elementos numéricos.
- Interesse
 - Mostrar os elementos especificados com a mesma largura.
- Adição de zeros
 - Elementos especificado ... com a largura máxima permitida pelo elemento de formato.
 - Exemplos de elementos de formato:
 - YYYY adiciona zeros à esquerda para a largura de 4 dígitos.
 - MM/DD ... adiciona zero à esquerda para a largura de 2 dígitos.
- Adição de espaços
 - Elemento especificado ... com a largura do elemento especificado mais comprido.
 - Exemplos de elementos de formato:
 - MONTH ... adiciona espaços para a largura do nome do mês completo mais comprido.
 - MON adiciona espaços para a largura do nome do mês abreviado mais comprido.
 - DAY adiciona espaços para a largura do nome do dia completo mais comprido.
 - DY Adiciona espaços para a largura do nome do dia abreviado mais comprido.

Modificadores de Modelos de Formato

▪ Modificador FX (Format Exact) (1/2)

- Especifica:
 - Formato exato do elemento especificado.
- Num elemento especificado na função TO_DATE:
 - Caracteres de pontuação / “texto”:
 - Devem corresponder exatamente ao elemento de formato.
 - Espaços extra:
 - Não devem existir. -- sem FX, os espaços extra são ignorados.
 - Elementos numéricos:
 - Devem ter o mesmo número de dígitos dos correspondentes elementos de formato.
 - Sem FX, os números podem ignorar zeros à esquerda.
- Retorna erro:
 - Se qualquer destas condições for violada.

The screenshot shows the Oracle SQL Developer interface. In the top tab bar, 'Worksheet' is selected. Below it, a query is written in the worksheet:

```
SELECT TO_CHAR(TO_DATE('2016-06-05','FXYYYY-MM-DD'),'YYYY-MM-DD') FROM DUAL;
```

In the bottom tab bar, 'Query Result' is selected. The result of the query is displayed in a table:

TO_CHAR(TO_DATE('2016-06-05','FXYYYY-MM-DD'),'YYYY-MM-DD')
1 2016-06-05

Modificador FX:

Obriga a especificar uma data com o formato exato do modelo de formato: o ano, o mês e o dia devem ter 4 dígitos, 2 dígitos e 2 dígitos, respectivamente. Também têm de estar separados por -.

Modificadores de Modelos de Formato

■ Modificador FX (Format Exact) (2/2)

■ Exemplos

```
UPDATE table  
SET date_column = TO_DATE(char, 'fmt');
```

char	'fmt'	Match or Error?
'15/ JAN /1998'	'DD-MON-YYYY'	Match
' 15! JAN % /1998'	'DD-MON-YYYY'	Error
'15/JAN/1998'	'FXDD-MON-YYYY'	Error
'15-JAN-1998'	'FXDD-MON-YYYY'	Match
'1-JAN-1998'	'FXDD-MON-YYYY'	Error
'01-JAN-1998'	'FXDD-MON-YYYY'	Match
'1-JAN-1998'	'FXFMDD-MON-YYYY'	Match

'fmt' = **modelo de formato**

- Regras Adicionais (não usando o modificador FX) (1/2)

- Caracteres de pontuação na string da data:
 - Podem ser omitidos ... se forem especificados todos os dígitos, incluindo zeros.

The screenshot shows the Oracle SQL Worksheet interface. In the 'Worksheet' tab, the query `SELECT TO_DATE('20161105', 'YYYY-MM-DD') FROM DUAL;` is entered. In the 'Query Result' tab, the output is displayed in a table:

	TO_DATE('20161105','YYYY-MM-DD')
1	16.11.05

- Podem ser substituídos por caracteres não-alfanuméricos.

- Exemplos:

- #, %, ...

The screenshot shows the Oracle SQL Worksheet interface. In the 'Worksheet' tab, the query `SELECT TO_DATE('2016#11#05', 'YYYY-MM-DD') FROM DUAL;` is entered. In the 'Query Result' tab, the output is displayed in a table:

	TO_DATE('2016#11#05','YYYY-MM-DD')
1	16.11.05

- **Regras Adicionais (não usando o modificador FX) (2/2)**

- Campos de tempo na *string* da data:
 - Podem ser ignorados.

The screenshot shows the Oracle SQL Worksheet interface. In the 'Worksheet' tab, the following SQL code is entered:

```
SELECT TO_DATE('2016-11-05','YYYY-MM-DD HH:MI:SS') FROM DUAL;
```

In the 'Query Result' tab, the output is displayed in a table:

	TO_DATE('2016-11-05','YYYY-MM-DD HH:MI:SS')
1	16.11.05

The status bar at the bottom indicates "All Rows Fetched: 1 in 0,003 seconds".

- Quando **falta a correspondência** entre elementos do formato e da *string* da data:
 - Oracle tenta os seguintes elementos de formato alternativos:

Original Format Element	Additional Format Elements to Try in Place of the Original
'MM'	'MON' and 'MONTH'
'MON'	'MONTH'
'MONTH'	'MON'
'YY'	'YYYY'
'RR'	'RRRR'

Elementos Básicos

Tipos de Dados

Modelos de Formato

Valor NULL

Comentários

Objetos de Esquema de BD

- Representa

- Valor **inexistente / desconhecido.**

- Armazenável numa Coluna

- De **qualquer tipo** de dados.
- **Sem restrições** de integridade:
 - NOT NULL
 - PRIMARY KEY

The screenshot shows the Oracle SQL Developer interface with the 'ALUNO' table selected. The table has three columns: 'NUMERO', 'NOME', and 'TELEFONE'. There are two rows of data:

	NUMERO	NOME	TELEFONE
1	1000000	Toto	2100000000
2	2000000	Chuck	(null)

Coluna **sem restrições**:

- NOT NULL
- PRIMARY KEY

■ Resultado da Operação

- NULL
- Exceção:
 - Operação de concatenação.

Operando NULL Concatenação

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there are two yellow boxes with arrows pointing to specific parts of the interface: one labeled "Operando NULL" pointing to the word "NULL" in the query, and another labeled "Concatenação" pointing to the concatenation operator "||". The main area shows a "Worksheet" tab with the following SQL code:

```
SELECT 10 + NULL, NULL || 'Toto'  
FROM DUAL;
```

Below the worksheet is a "Query Result" window. It displays the results of the executed query. The first column contains the expression "10+NULL" and the second column contains "NULL||'Toto'". The result row shows a value of "(null)" in the first column and "Toto" in the second column. A yellow box at the bottom is labeled "Resultado".

Tabela DUAL.

The screenshot shows the Oracle SQL Developer interface. In the top-right corner, there is a yellow box with an arrow pointing to the "Tabela DUAL." label. The main area shows a "Worksheet" tab with the following SQL code:

```
SELECT * FROM DUAL;
```

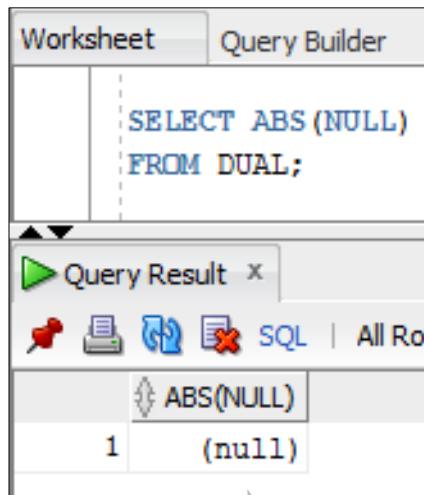
Below the worksheet is a "Query Result" window. It displays the results of the executed query. The result row shows a value of "X" in the column. A yellow box at the bottom is labeled "Resultado".

▪ Maioria das Funções Escalares

- Retornam NULL.
- Exemplo

Função escalar:

- Retorna um valor simples. -- não-estruturado.



The screenshot shows the Oracle SQL Developer interface. In the 'Worksheet' tab, the following SQL code is entered:

```
SELECT ABS(NULL)
FROM DUAL;
```

In the 'Query Result' tab, the output is displayed in a table:

	ABS(NULL)
1	(null)

Retorno

■ Resultado NULL

- Pode ser substituído por ... valor não NULL:
 - Usando funções apropriadas.
- Exemplo:
 - Função NVL

Substitui NULL por string

A screenshot of the Oracle SQL Developer interface. The query window shows the following SQL statement:

```
SELECT NVL(NULL, 'Não aplicável') FROM DUAL;
```

The results window displays the output of the query:

	NVL(NULL,'NÃOAPLICÁVEL')
1	Não aplicável

The output row contains the value 'Não aplicável'.

Substitui NULL por número

A screenshot of the Oracle SQL Developer interface. The query window shows the following SQL statement:

```
SELECT NVL(NULL, 20) FROM DUAL;
```

The results window displays the output of the query:

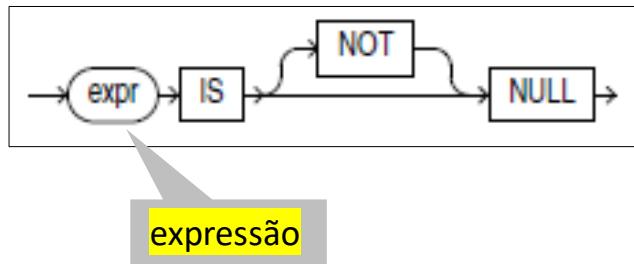
	NVL(NULL,20)
1	20

The output row contains the value '20'.

■ Interesse

- Testar o valor NULL. -- única condição para testar NULL.

■ Sintaxe



■ Formas

Forma	Propósito	Exemplo
expr IS NULL	expr = NULL \Rightarrow Resultado TRUE expr \neq NULL \Rightarrow Resultado FALSE	<pre>SELECT last_name FROM employees WHERE commission_pct IS NULL ORDER BY last_name;</pre>
expr IS NOT NULL	expr = NULL \Rightarrow Resultado FALSE expr \neq NULL \Rightarrow Resultado TRUE	

- Operadores de Comparaçāo
 - =, >, >=, <, <=, !=, <>

- Condição com Operando NULL

- Exemplo:

- a > b -- a=10 e b=NULL

- Resultado:

- UNKNOWN

- Razão:

- Resultado depende do NULL.
 - NULL não pode ser igual / diferente de:
 - Qualquer valor.
 - Outro NULL.

- Teste de Comparaçāo

- Resultados possíveis:

- TRUE
 - FALSE
 - UNKNOWN -- com operando NULL.

- Operadores Lógicos

- AND, OR e NOT
- Resultados possíveis:
 - TRUE
 - FALSE
 - UNKNOWN -- com operando NULL

- Tabelas de Verdade

- AND

AND	TRUE	FALSE	UNKNOWN
TRUE	TRUE	FALSE	UNKNOWN
FALSE	FALSE	FALSE	FALSE
UNKNOWN	UNKNOWN	FALSE	UNKNOWN

- OR

OR	TRUE	FALSE	UNKNOWN
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	UNKNOWN
UNKNOWN	TRUE	UNKNOWN	UNKNOWN

- NOT

--	TRUE	FALSE	UNKNOWN
NOT	FALSE	TRUE	UNKNOWN

Resultado Lógico UNKNOWN

■ Comportamento

- Quase como FALSE
 - UNKNOWN ≠ FALSE
 - NOT FALSE = TRUE
 - mas
 - NOT UNKNOWN = UNKNOWN
- Exemplos

Table 2–20 Conditions Containing Nulls

Condition	Value of A	Evaluation
a IS NULL	10	FALSE
a IS NOT NULL	10	TRUE
a IS NULL	NULL	TRUE
a IS NOT NULL	NULL	FALSE
a = NULL	10	UNKNOWN
a != NULL	10	UNKNOWN
a = NULL	NULL	UNKNOWN
a != NULL	NULL	UNKNOWN
a = 10	NULL	UNKNOWN
a != 10	NULL	UNKNOWN

```
14 | SELECT *
15 | FROM DUAL
16 | WHERE DUMMY=NULL;
```

The screenshot shows the Oracle SQL Developer interface. In the 'Query Result' tab, there is no data returned, indicated by the message 'All Rows Fetched: 0'. A yellow box labeled 'UNKNOWN' is placed over the word 'UNKNOWN' in the WHERE clause of the SQL code.

Resultado Vazio

```
14 | SELECT *
15 | FROM DUAL
16 | WHERE NOT (DUMMY=NULL);
```

The screenshot shows the Oracle SQL Developer interface. In the 'Query Result' tab, there is no data returned, indicated by the message 'All Rows Fetched: 0 in 0,001 se'. A yellow box labeled 'UNKNOWN' is placed over the word 'UNKNOWN' in the WHERE clause of the SQL code.

Resultado Vazio

■ Interesse

- Facilitar o processamento do valor NULL.

■ Funções

Nome	Funcionalidade
NVL	Recebe, por parâmetro, 2 expressões. Avalia a 1ª expressão e se for NULL retorna a 2ª expressão, senão retorna a 1ª expressão (expressão avaliada). Útil para substituir o valor NULL (retornado como um espaço) por uma string no resultado de uma query.
NVL2	Recebe, por parâmetro, 3 expressões. Avalia a 1ª expressão e se for NULL retorna a 2ª expressão, senão retorna a 3ª expressão.
COALESCE	É a generalização da função NVL. Recebe, por parâmetro, uma lista de expressões e retorna o valor da 1ª expressão não NULL.
LNVL	Recebe por parâmetro uma condição e, retorna TRUE, se a condição for FALSE / UNKNOWN. Retorna FALSE se a condição for TRUE. Útil quando uma condição pode ser avaliada para UNKNOWN (se qualquer operando da condição for NULL).
NANVL	Útil para números de vírgula-flutuante do tipo BINARY_FLOAT / BINARY_DOUBLE. Substitui o valor NaN (not a number) pelo valor de uma expressão especificada.
NULLIF	Recebe, por parâmetro, 2 expressões. Se forem iguais retorna NULL, senão, retorna a 1ª expressão.

Função NVL (1/3)

Sintaxe



NVL(expr1, expr2)

Semântica

Retorna:

- expr1 ... se expr1 ≠ NULL.
- expr2 ... se expr1 = NULL.

Interesse:

- Substituir NULL por um valor string/numérico:
 - No resultado de uma query.

Tipo de retorno:

- VARCHAR2 / tipo de dados numérico.

NVL(NULL,'NÃOAPLICÁVEL')
1 Não aplicável

NVL(NULL,20)
1 20

Função NVL (2/3)

Semântica

Argumentos expr1 e expr2:

- Tipo de dados:
 - Qualquer.
- Com tipos diferentes (tipo expr1 ≠ tipo expr2):
 - Tipos incompatíveis:
 - Retornado um erro.
 - Tipos compatíveis:
 - Convertida implicitamente uma das expressões.
 - Tipo expr1 = *string* de caracteres ⇒ tipo expr2 convertido para tipo expr1.
 - Retornado valor VARCHAR2:
 - No character set de expr1.
 - Tipo expr1 = numérico
 - Determinada a expr com maior precedência numérica.
 - Outro argumento convertido implicitamente para esse tipo de dados.
 - Retornado valor desse tipo de dados.

Função NVL (3/3)

Semântica

Exemplo

```
SELECT last_name, NVL(TO_CHAR(commission_pct), 'Not Applicable') commission
  FROM employees
 WHERE last_name LIKE 'B%'
 ORDER BY last_name;
```

LAST_NAME	COMMISSION
Baer	Not Applicable
Baida	Not Applicable
Banda	.1
Bates	.15
Bell	Not Applicable
Bernstein	.25
Bissot	Not Applicable
Bloom	.2
Bull	Not Applicable

Função NVL2 (1/3)

Sintaxe



NVL2(expr1, expr2, expr3)

Semântica

Retorna:

- expr2 ... se expr1 ≠ NULL.
- expr3 ... se expr1 = NULL.

Interesse:

- Substituir NULL por um valor string/numérico:
 - No resultado de uma *query*.

Tipo de retorno:

- VARCHAR2 / tipo de dados numérico.

```
Worksheet | Query Builder
SELECT NVL2(NULL, 'exp2', 'exp3')
FROM DUAL;
```

Query Result | All Rows Fetched: 1 in 0 sec

NVL2(NULL,'EXP2','EXP3')
1 exp3

```
Worksheet | Query Builder
SELECT NVL2(20, 'exp2', 'exp3')
FROM DUAL;
```

Query Result | All Rows Fetched: 1 in 0,0 sec

NVL2(20,'EXP2','EXP3')
1 exp2

Função NVL2 (2/3)

Semântica

Argumentos expr1, expr2 e expr3:

- Tipo de dados:
 - Qualquer.
- Com tipos diferentes (tipo expr2 ≠ tipo expr3):
 - Tipos incompatíveis:
 - Retornado um erro.
 - Tipos compatíveis:
 - Convertida implicitamente uma das expressões.
 - Tipo expr2 = *string* de caracteres ⇒ tipo expr3 convertido para tipo expr2.
 - Retornado valor VARCHAR2:
 - No *character set* de expr2.
 - Tipo expr2 = numérico
 - Determinada a expr com maior precedência numérica.
 - Outro argumento convertido implicitamente para esse tipo de dados.
 - Retornado valor desse tipo de dados.

Função NVL2 (3/3)

Semântica

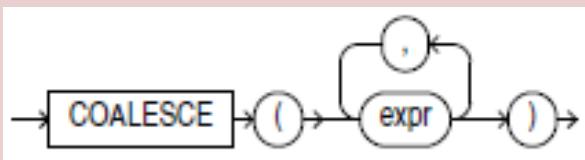
Exemplo

```
SELECT last_name, salary,
       NVL2(commission_pct, salary + (salary * commission_pct), salary) income
  FROM employees
 WHERE last_name like 'B%'
 ORDER BY last_name;
```

LAST_NAME	SALARY	INCOME
Baer	10000	10000
Baida	2900	2900
Banda	6200	6820
Bates	7300	8395
Bell	4000	4000
Bernstein	9500	11875
Bissot	3300	3300
Bloom	10000	12000
Bull	4100	4100

Função COALESCE (1/2)

Sintaxe



COALESCE(expr, expr [, expr [...]]) -- [] = opcional ; obrigatório pelo menos duas expr.

Semântica

Retorna um dos seguintes resultados:

- Primeira expressão expr não-NUL da lista de expressões.
- NULL
 - Se todas as expressões forem avaliadas para NULL.

Tipo de retorno:

- Tipo numérico ... com a maior precedência numérica.
 - Se todas as expressões forem numéricas / convertíveis implicitamente para tipo numérico.

Função COALESCE (2/2)

Semântica

Exemplo

```
SELECT product_id, list_price, min_price,  
       COALESCE(0.9*list_price, min_price, 5) "Sale"  
  FROM product_information  
 WHERE supplier_id = 102050  
 ORDER BY product_id;
```

PRODUCT_ID	LIST_PRICE	MIN_PRICE	Sale
1769	48		43.2
1770		73	73
2378	305	247	274.5
2382	850	731	765
3355			5

Preço dos produtos:

SE (produto tem list_price) ENTÃO
 preço = 0.9 * list_price

SENÃO SE (produto tem min_price) ENTÃO
 preço = min_price

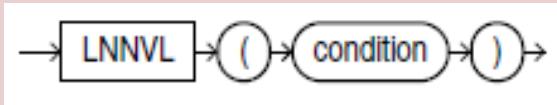
SENÃO

preço = 5

FIMSE

Função LNNVL (1/2)

Sintaxe



LNNVL(condition)

condition:

- Condição simples.
- Ilegal operadores AND, OR ou BETWEEN.

Semântica

Retorna um dos seguintes resultados:

- TRUE: condição = FALSE / UNKNOWN
- FALSE: condição = TRUE

a = 2 e b = NULL		
Condição	Valor da Condição	Retorno de LNNVL
a = 1	FALSE	TRUE
a = 2	TRUE	FALSE
a IS NULL	FALSE	TRUE
b = 1	UNKNOWN	TRUE
b IS NULL	TRUE	FALSE
a = b	UNKNOWN	TRUE

Função LNNVL (2/2)

Semântica

Exemplo

```
SELECT COUNT(*)  
  FROM employees  
 WHERE commission_pct < .2;
```

```
COUNT(*)  
-----  
11
```

Empregados que recebem comissão inferior a 20%.

- Excluindo:
 - Empregados que não recebem comissão.

```
SELECT COUNT(*)  
  FROM employees  
 WHERE LNNVL(commission_pct >= .2);
```

```
COUNT(*)  
-----  
83
```

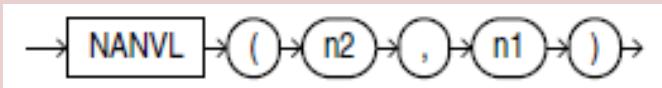
Empregados que recebem comissão inferior a 20%.

- Incluindo:
 - Empregados que não recebem comissão.

Condição oposta à anterior.

Função NANVL (1/2)

Sintaxe



NANVL(n2, n1)

Semântica

Retorna:

- n1 ... se n2 = NaN. -- n1 alternativa a n2 = NaN
- n2 ... se n2 ≠ NaN.

Interesse:

- Substituir NaN por um valor numérico.
- Útil apenas para valores do tipo BINARY_FLOAT / BINARY_DOUBLE.

Tipo de retorno:

- Tipo numérico ... com a maior precedência numérica.
 - Se todas as expressões forem numéricas / convertíveis implicitamente para tipo numérico

Função NANVL (2/2)

Semântica

Exemplo

```
INSERT INTO float_point_demo
VALUES (0, 'NaN', 'NaN');

SELECT *
FROM float_point_demo;

DEC_NUM BIN_DOUBLE  BIN_FLOAT
-----
1234.56 1.235E+003 1.235E+003
0          Nan        Nan
```

```
SELECT bin_float, NANVL(bin_float,0)
FROM float_point_demo;
```

```
BIN_FLOAT NANVL(BIN_FLOAT,0)
-----
1.235E+003      1.235E+003
Nan            0
```

Nan substituído por 0.

Função NULLIF (1/2)

Sintaxe



NULLIF(expr1, expr2)

Semântica

Retorna:

- NULL ... se expr1 = expr2. -- comparadas as expressões especificadas.
- expr1 ... se expr1 ≠ expr2.

Equivalente logicamente à expressão CASE:

- CASE WHEN expr1=expr2 THEN **NULL** ELSE expr1 END.

Retorno:

- Ambas as expressões de **tipos numéricos**:
 - Determinada a expr com maior precedência numérica.
 - Outro argumento convertido implicitamente para esse tipo de dados.
 - Retornado valor desse tipo de dados.
- Ambas as expressões **não numéricas**:
 - Tipo expr1 tipo ≠ expr2 ⇒ retornado um erro.

Função NULLIF (2/2)

Semântica

Exemplo

```
SELECT e.last_name, NULLIF(j.job_id, e.job_id) "Old Job ID"  
  FROM employees e, job_history j  
 WHERE e.employee_id = j.employee_id  
 ORDER BY last_name, "Old Job ID";
```

LAST_NAME	Old Job ID
De Haan	IT_PROG
Hartstein	MK_REP
Kaufling	ST_CLERK
Kochhar	AC_ACCOUNT
Kochhar	AC_MGR
Raphaely	ST_CLERK
Taylor	SA_MAN
Taylor	
Whalen	AC_ACCOUNT
Whalen	

Empregados que mudaram de tarefa desde a sua contratação.

Elementos Básicos

Tipos de Dados

Modelos de Formato

Valor NULL

Comentários

Objetos de Esquema de BD

■ Interesse

- Facilitar:
 - Leitura
 - Manutenção
- } do código da aplicação.

■ Tipos de Comentário

- Comentários em comandos SQL
 - Armazenados:
 - Juntamente com o código da aplicação.
 - Tipos
 - Linha --
 - Multi-linha /* ... */
 - Ignorados
 - Na execução da aplicação.
- Comentários de esquema de BD
 - Armazenados:
 - No dicionário de dados.
 - Juntamente com metadados sobre os próprios objetos.
 - Criação
 - Comando COMMENT.

```
-- isto é um comentário de linha
/*
   Isto é um
   comentário multi-linha
*/
```

Elementos Básicos

Tipos de Dados

Modelos de Formato

Valor NULL

Comentários

Objetos de Esquema de BD

Objetos de Esquema de BD

■ BD Oracle

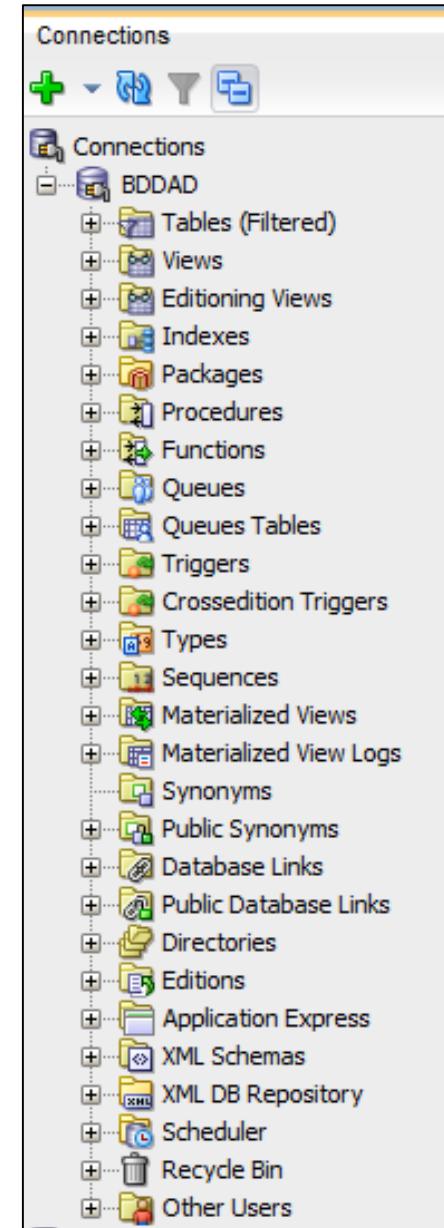
- Reconhece dois tipos de objetos:
 - **Associados** a um esquema particular de BD.
 - **Não-associados** ... a qualquer esquema particular de BD.

Objetos de um Esquema de BD

- Clusters
- **Constraints**
- Database links
- **Database triggers**
- Dimensions
- External procedure libraries
- Index-organized tables
- Indexes
- Indextypes
- Java: classes, resources, sources
- Materialized views
- Materialized view logs
- Mining models
- Object tables
- Object types
- Object views
- Operators
- **Packages**
- Sequences
- **Stored functions, stored procedures**
- Synonyms
- Tables
- Views

Objetos **não-associados** a qualquer esquema particular de BD:

- Contexts
- Directories
- Editions
- Restore points
- Roles
- Rollback segments
- Tablespaces
- **Users**



■ Alguns Objetos

- Constituídos por sub-objetos -- partes de objetos.
 - Exemplo:
 - Objeto Tabela.
 - Sub-objetos de Tabela:
 - Colunas.
 - Restrições de integridade (*constraints*).

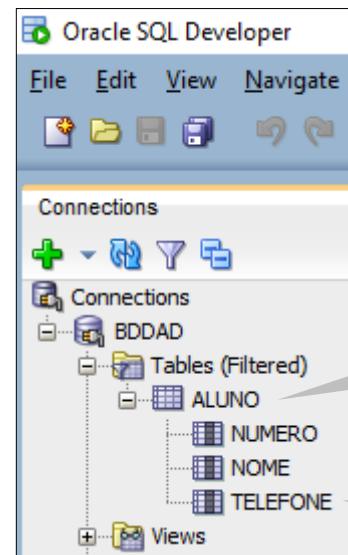


Tabela ALUNO:

- Objeto.

Coluna TELEFONE:

- Sub-objeto de tabela.

- **Objetos e Sub-Objetos**

- Têm nomes.

- **Nomes de Objetos Oracle**

- Têm prefixo:
 - SYS_ -- objetos do sistema.
 - ORA_ -- alguns objetos fornecidos.

■ Regras de Criação de Nomes (1/6)

- **Tipos** de nomes:

- Não-citado -- nome **simples**;
 -- exemplo: NOTA.
 - Citado -- nome **composto**;
 -- exemplo: "NOTA de ALUNO".

- **Comprimento:**

- [1, 30] **bytes**.

- **Exceção:**

- Nome da BD: [1, 8] **bytes**.

Nome delimitado
por aspas.

■ Regras de Criação de Nomes (2/6)

- Nomes **não-citados**:
 - Não podem ser palavras-reservadas do Oracle SQL.
- Carateres:
 - Inicial:
 - Carater **alfabético** ... do *character-set* da BD.
 - Restantes:
 - Carateres **alfanuméricos**
 - **_ (underscore)**
 - **\$ (dólar)** -- Oracle não recomenda.
 - **# (cardinal)** -- Oracle não recomenda.
- Case-Insensitive:
 - Oracle insensível às letras minúsculas e maiúsculas.
 - Exemplo:
 - nome = NOME = Nome
 - Oracle **interpreta** as letras como **maiúsculas**.
 - **Armazena** apenas letras **maiúsculas**.

■ Regras de Criação de Nomes (3/6)

- Nomes **citados**:
 - Podem ser palavras-reservadas do Oracle SQL.
 - Oracle não recomenda.
- Carater **inicial**:
 - Qualquer.
 - Exceto aspas.
- Case-Sensitive:
 - Exemplo:
 - “nome” ≠ “NOME” ≠ “Nome”

Oracle Não Recomenda:

- **Nomes-citados**.
 - Razão:
 - Aceite pelo SQL *PLUS (usado pelo SQL Developer).
 - Pode não ser aceite noutras ferramentas.

■ Regras de Criação de Nomes (4/6)

- Usar **caracteres**:
 - Do *character-set* ASCII.
 - Fornecem ótima compatibilidade ... entre diferentes:
 - Plataformas
 - Sistemas Operativos
 - Não usar (para **evitar conflitos** de nomes):
 - **Prefixos**:
 - SYS_ -- em objetos do sistema Oracle.
 - ORA_ -- em alguns objetos fornecidos pelo Oracle.
 - **Nomes de funções SQL predefinidas**.

■ Regras de Criação de Nomes (5/6)

■ *Namespace*

- Objetos que partilham o mesmo *namespace*:
 - Têm nomes únicos. -- não há nomes duplicados.

▪ Exemplo:

- Tables
- Views
- Sequences

Num esquema de BD (ex: BDDAD):

- Nomes Tabelas ≠ Nomes Views ≠ Nomes Sequências

▪ Objetos que têm o seu próprio *namespace*:

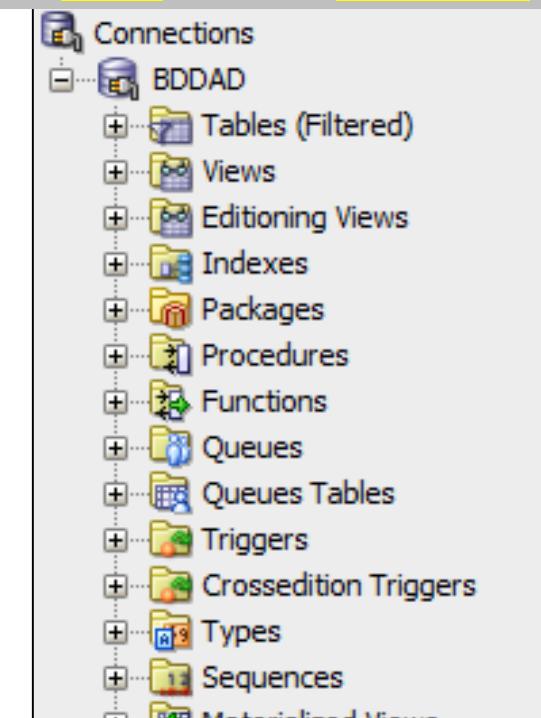
- Indexes
- Constraints

▪ Cada esquema de BD:

- Tem o seu próprio *namespace*.

▪ *Colunas*:

- Da mesma tabela/vista:
 - Não podem ter o mesmo nome.
- De diferentes tabelas/vistas:
 - Podem ter o mesmo nome.



■ Regras de Criação de Nomes (6/6)

- **Nomes:**
 - Devem ser úteis, descritivos e pronunciáveis.
 - Equilibrar os objetivos:
 - Nome curto para facilitar o uso.
 - Nome o mais descritivo possível.
- **Usar regras consistentes.**
 - Ajuda os utilizadores:
 - Perceberem o papel de cada tabela na aplicação.
 - Exemplo de regra:
 - Aplicação FINANÇAS
 - Nomes de **todas** as tabelas ... iniciados com fin_
- Descrição da **mesma** entidade/atributo em **várias** tabelas.
 - Usar o **mesmo nome**.
 - Exemplo:
 - Tabelas Empregado e Departamento.
 - As colunas do número do departamento:
 - Em ambas as tabelas.
 - Devem ter o mesmo nome nr_departamento.

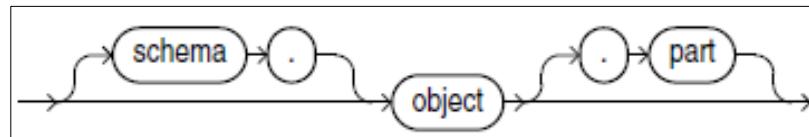
Tabela Departamento	
nr_departamento	...
...	...

Tabela Empregado	
...	nr_departamento
...	...

- Comandos SQL

- Referenciar Objetos/Sub-Objetos de um Esquema de BD

- Sintaxe



- object -- nome de um objeto.
 - schema -- esquema que contém o object.
 - opcional.
 - Útil para referenciar um objeto de outro esquema (preciso permissão).
 - Desnecessário para referenciar um objeto do próprio esquema.
 - part -- é uma parte do objeto (sub-objeto); p.ex., coluna de uma tabela.

- Exemplo

- Esquema EMPRESA
 - Tabela EMPREGADO(id_empregado, nome)
 - Referências da coluna nome
 - EMPREGADO.nome -- comando SQL do esquema EMPRESA.
 - EMPRESA.EMPREGADO.nome -- comando SQL de esquema ≠ de EMPRESA.

Operadores Não-Lógicos

- [Introdução](#)
- [Categorias de Operadores](#)
- [Operando NULL](#)
- [Precedência de Operadores](#)
- Operadores Não-Lógicos
 - [Aritméticos](#)
 - [Concatenação](#)
 - [Conjuntos](#)

■ Interesse

- Processar dados ... e retornar um resultado.
- Exemplo
 - $2 + 3$ -- resultado = 5

■ Dados Processados

- Designados operandos / argumentos
- Exemplo
 - $2 + 3$ -- operando_2 + operando_3

■ Representação

- Caracteres especiais -- ex: * representa a multiplicação.
- Palavras-Chave -- ex: IN, UNION, AND

■ Categorias

- Pelo Nº de Operandos
 - Unários -- operam sobre um único operando.
-- formato: operador operando -- ex: -2
 - Binários -- operam sobre dois operandos.
-- formato: operando1 operador operando2 -- ex: 2 + 3
- Pela Funcionalidade
 - [Aritméticos](#) +, -, *, /
 - [Concatenação](#) ||
 - [Relacionais](#) =, !=, <>, >, <, >=, <=
 - [Lógicos](#) AND, OR, NOT
 - [Conjuntos](#) UNION, UNION ALL, INTERSECT, MINUS
 - [Outros](#) BETWEEN, IS, IN, LIKE, REGEXP_LIKE

■ Resultado da Operação

- NULL
- Exceção:
 - Operação de concatenação.

Operando NULL Concatenação

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there are tabs for 'Worksheet' and 'Query Builder'. Below them, a code editor window contains the following SQL query:

```
SELECT 10 + NULL, NULL || 'Toto'  
FROM DUAL;
```

Below the code editor is a 'Query Result' window. It displays the results of the executed query. The first column is labeled '(10+NULL)' and contains '(null)'. The second column is labeled '(NULL||'Toto')' and contains 'Toto'. A status bar at the bottom of the window indicates 'All Rows Fetched: 1 in 0,001s'.

Resultado

Tabela DUAL.

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there are tabs for 'SQL Worksheet' and 'History'. Below them, a code editor window contains the following SQL query:

```
SELECT * FROM DUAL;
```

Below the code editor is a 'Script Output' window. It displays the results of the executed query. The first column is labeled '(DUMMY)' and contains 'X'. A status bar at the bottom of the window indicates 'All Rows Fetched: 1 in 0,001s'.

■ Definição

- É a ordem pela qual são executados os operadores numa expressão.

Precedência	Operador	Observação
Maior	+, -	Sinais
	* /	Multiplicação e Divisão
	+, -,	Soma, Subtração e Concatenação
Menor	Condições SQL	Avaliadas depois destes operadores SQL

■ Expressão com Múltiplos Operadores

- Operadores com maior precedência:
 - Executados antes dos operadores com menor precedência.
- Operadores com igual precedência:
 - Executados da esquerda para a direita ... dentro da mesma expressão.

■ Parêntesis Curvos

- Sobreponem-se às precedências dos operadores.

■ Interesse

- Negar, adicionar, subtrair, multiplicar e dividir:
 - Valores numéricos.
- Usar também:
 - Na aritmética data-tempo e intervalo.

■ Operadores

- +
 - Unário -- é sinal.
 - Binário -- é adição.
- -
 - Unário -- é sinal.
 - Binário -- é subtração.
- * -- multiplicação.
- / -- divisão.

■ Tipo de Retorno

- Operadores unários:
 - Igual ao do operando.
- Operadores binários:
 - Igual ao do operando com maior precedência numérica (o maior valor).

- Divisão de Valor por Zero

- Valor do tipo NUMBER

- Gera um erro

Worksheet Query Builder

```
SELECT 2/0 FROM DUAL;
```

Query Result x

SQL | Executing:SELECT 2/0 FROM DUAL;

ORA-01476: divisor is equal to zero
01476. 00000 - "divisor is equal to zero"
*Cause:
*Action:

Worksheet Query Builder

```
SELECT TO_CHAR(remainder(2,0)) FROM DUAL;
```

Query Result x

SQL | Executing:SELECT TO_CHAR(remainder(2,0)) FROM DUAL;

ORA-01476: divisor is equal to zero
01476. 00000 - "divisor is equal to zero"
*Cause:
*Action:

- Valor do tipo BINARY_FLOAT / BINARY_DOUBLE

- Gera infinity / NaN

Worksheet Query Builder

```
SELECT 2f/0 FROM DUAL;
```

Query Result x

SQL | All Rows Fetch

2f/0
1 Infinity

Worksheet Query Builder

```
SELECT 2d/0 FROM DUAL;
```

Query Result x

SQL | All Rows Fetch

2d/0
1 Infinity

Worksheet Query Builder

```
SELECT TO_CHAR(remainder(2f,0)) FROM DUAL;
```

Query Result x

SQL | All Rows Fetched: 1 in 0,001 seconds

TO_CHAR(remainder(2f,0))
1 Nan

■ Exemplos

Operator	Purpose	Example
+ -	When these denote a positive or negative expression, they are unary operators.	<pre>SELECT * FROM order_items WHERE quantity = -1 ORDER BY order_id, line_item_id, product_id;</pre> <pre>SELECT * FROM employees WHERE -salary < 0 ORDER BY employee_id;</pre>
+ -	When they add or subtract, they are binary operators.	<pre>SELECT hire_date FROM employees WHERE SYSDATE - hire_date > 365 ORDER BY hire_date;</pre>
* /	Multiply, divide. These are binary operators.	<pre>UPDATE employees SET salary = salary * 1.1;</pre>

■ Interesse

- Obter a junção de duas *strings* de caracteres.

Table 4–3 Concatenation Operator

Operator	Purpose	Example
<code> </code>	Concatenates character strings and CLOB data.	<code>SELECT 'Name is ' last_name FROM employees ORDER BY last_name;</code>

■ Resultado

- Outra *string* que combina as duas *strings* concatenadas.
- Tipo
 - CHAR -- ambos os operandos forem do tipo CHAR.
 - VARCHAR2 -- um dos operandos for do tipo VARCHAR2.

■ Recomendações

- Concatenar expressão que pode ser NULL:
 - Usar a função NVL
 - Para converter explicitamente a expressão para *string* de comprimento zero.
- Função CONCAT
 - Alternativa para:
 - Portabilidade de aplicações entre ambientes com diferentes *character sets*.
 - Exemplo:
 - ASCII e EBCDIC codificam a barra vertical de forma diferente.

- **Interesse**

- Combinar resultados de 2 *queries* num simples resultado. -- resultado = conjunto de linhas.

- **Queries Compostas**

- Queries com operadores de conjuntos.

- **Operadores**

- Binários
- Tipos

Operador	Retorno
UNION	Todas as linhas distintas selecionadas por qualquer das queries.
UNION ALL	Todas as linhas selecionadas por qualquer das queries, incluindo as linhas duplicadas.
INTERSECT	Todas as linhas distintas selecionadas por ambas as queries.
MINUS	Todas as linhas distintas selecionadas pela 1ª query mas não pela 2ª query.

- **Precedência**

- Todos os operadores têm igual precedência.

Expressões

- Introdução
- Expressões
 - Simples
 - Compostas
- Tipos de Expressões
 - Coluna
 - Data-Tempo
 - Função
 - Acesso a Objetos
 - Scalar Subquery
 - Listas

■ Definição

- Combinação de um ou mais:
 - Valores
 - Operadores
 - Funções SQL

■ Exemplos

- `2 * 2` -- expressão do tipo NUMBER = tipo dos números 2
- `TO_CHAR(TRUNC(SYSDATE+7))` -- expressão que combina valor, operador e funções

■ Objetivo

- Obter um valor.

■ Geralmente

- Assume o tipo de dados dos seus componentes.

■ Nos Comandos

■ **SELECT**

- Na lista selecionada
- Na condição das cláusulas
 - WHERE
 - HAVING
 - ORDER BY

■ **INSERT**

- Cláusula VALUES

■ **UPDATE**

- Cláusula SET

Condições

- Condição

- [Definição](#)
- [Uso](#)
- [Precedências](#)
- [Formas](#)
 - [Comparação](#)
 - [Vírgula-Flutuante](#)
 - [Lógica](#)
 - [Pattern-Matching](#)
 - [NULL](#)
 - [BETWEEN](#)
 - [EXISTS](#)
 - [IN](#)

- **Uma Condição**

- **Especifica:**
 - Combinação de:
 - Uma ou mais expressões.
 - Operadores lógicos.
- **Retorna:**
 - Um dos valores:
 - TRUE
 - FALSE
 - UNKNOWN -- devido a NULL.

Uso de Condição

Cláusula	Comandos			
	SELECT	DELETE	UPDATE	CREATE TABLE
WHERE	√	√	√	
HAVING	√			
CHECK				√

	Tipo de Condição	Propósito
Maior Prioridade	Operadores SQL avaliados antes das condições SQL	
	=, !=, <, >, <=, >=	Comparação
	IS [NOT] NULL	
	LIKE	
	[NOT] BETWEEN	Comparação
	[NOT] IN	
	EXISTS	
Menor Prioridade	NOT	Negação
	AND	Conjunção
	OR	Disjunção

- **Formas**

- Comparação
- Vírgula-Flutuante
- Lógicas
- *Pattern-Matching*
- NULL
- BETWEEN
- EXISTS
- IN

- **Interesse**

- Comparar **duas** expressões.

- **Resultado**

- TRUE
 - FALSE
 - UNKNOWN

- **Tipos de Comparação**

- Numérica
- Carateres
- Data-Tempo
- Intervalo

- **Comparação Numérica**

- **Tipos** de comparação:
 - NUMBER
 - BINARY_FLOAT
 - BINARY_DOUBLE
- **Determinação do tipo** de comparação:
 - Baseada na precedência numérica.
- Valor **NaN**:
 - É valor de vírgula-flutuante.
 - NaN > qualquer outro valor numérico.

■ Comparação de Carateres (1/4)

- Baseada em 2 medidas:
 - Ordenação:
 - Binária -- usada por omissão.
-- segundo ordem numérica dos códigos do *character set* da BD.
ou
 - Linguística -- segundo ordem alfabética dos carateres.
-- usada se ordem de códigos numéricos ≠ ordem linguística dos carateres.
 - Semântica de comparação de espaços (brancos):
 - Espaços adicionados -- uso em comparações de duas expressões de tipo CHAR.
ou
 - Espaços não-adicionados -- uso em comparações com pelo menos 1 expressão VARCHAR.

Comparação de Caracteres (2/4)

- Ordem binária:
 - Usada:
 - Por omissão.
 - Compara:
 - Concatenações de códigos numéricos, de caracteres, ... definidos no character set da BD.
 - Caráter A > Caráter B:
 - Se código numérico de A > código numérico de B
 - Espaços (brancos):
 - Código < código de qq outro caráter.
 - Na maioria dos character sets.
- Ordem linguística:
 - Interesse:
 - Se ordem de códigos numéricos ≠ ordem linguística, dos carateres comparados.
 - Para usar:
 - Parâmetro NLS_SORT ≠ BINARY
 - Parâmetro NLS_COMP = LINGUISTIC

Parâmetros
NLS.

Valores por
omissão.

Parameter	Value
1 NLS_CALENDAR	GREGORIAN
2 NLS_CHARACTERSET	WE8MSWIN1252
3 NLS_COMP	BINARY
4 NLS_CURRENCY	€
5 NLS_DATE_FORMAT	RR.MM.DD
6 NLS_DATE_LANGUAGE	PORTUGUESE
7 NLS_DUAL_CURRENCY	€
8 NLS_ISO_CURRENCY	PORTUGAL
9 NLS_LANGUAGE	PORTUGUESE
10 NLS_LENGTH_SEMANTICS	BYTE
11 NLS_NCHAR_CHARACTERSET	AL16UTF16
12 NLS_NCHAR_CONV_EXCP	FALSE
13 NLS_NUMERIC_CHARACTERS	,
14 NLS_SORT	WEST EUROPEAN

■ Comparação de Caracteres (3/4)

- Semântica de comparação de espaços (brancos) (1/2)
 - Espaços adicionados
 - Usada ... quando ambos os valores comparados são do tipo CHAR / valores textuais.
 - Procedimento de comparação:
 1. Se dois valores comparados tiverem comprimentos diferentes:
 - Adicionados espaços no fim do valor mais curto ⇒ comprimentos iguais.
 2. Comparação caráter a caráter ... até ao primeiro caráter diferente e:
 - Se ... forem encontrados caracteres diferentes:
 - O valor que tiver o maior caráter é considerado o maior valor.
 - Senão ... os valores são considerados iguais.
 - Assim considerados iguais:
 - Valores diferentes apenas no número de espaços finais.
 - Espaços não-adicionados
 - Usada ... quando ambos os valores comparados são do tipo VARCHAR2.
 - Procedimento semelhante ao anterior, exceto quando:
 - Os valores têm diferentes comprimentos mas são iguais até ao final do mais pequeno:
 - Considerado o maior valor, o que tem o maior comprimento.

Comparação de Caracteres (4/4)

- Semântica de comparação de espaços (brancos) (2/2)
 - Exemplos

Espaços Adicionados	Espaços Não-Adicionados	Obs
'ac' > 'ab'	'ac' > 'ab'	
'ab' > 'a '	'ab' > 'a '	
'ab' > 'a'	'ab' > 'a'	
'ab' = 'ab'	'ab' = 'ab'	
'a ' = 'a'	'a ' > 'a'	Diferentes

- Frequentemente:
 - Resultados são iguais ... nos dois tipos de comparação.

- **Comparação Data-Tempo**

- **Data Mais Recente > Data Mais Antiga**
 - '2014-12-24' > '2012-12-24'
- **Tempo Mais Recente > Tempo Mais Antigo**
 - '2015-11-19 16:30:05' > '2015-11-19 8:15:00'

▪ Comparação Intervalo

- Valores de tipo interval:
 - Têm de ser do mesmo tipo.

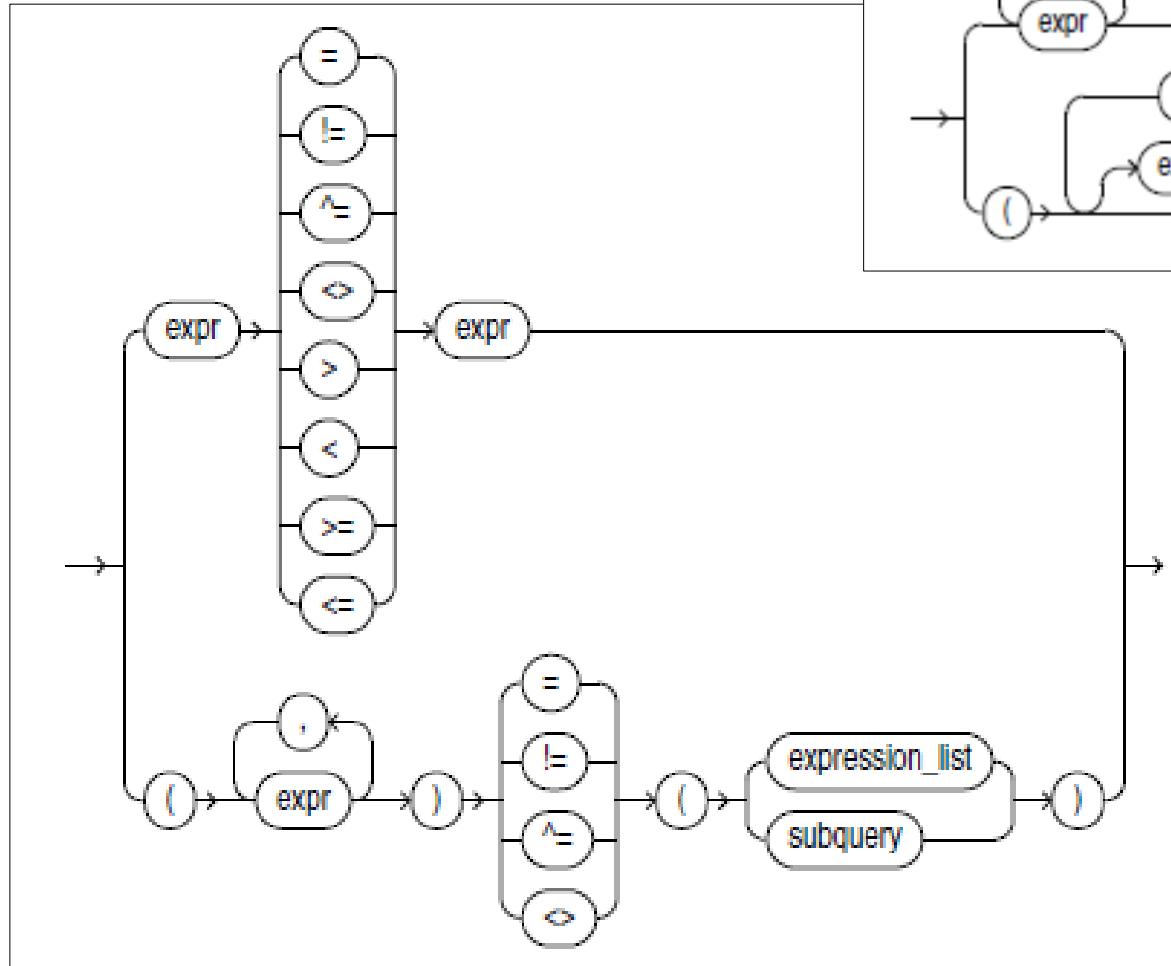
■ Tipos de Condição

Tipo de Condição	Propósito	Exemplo
=	Teste de igualdade.	
!= <>	Teste de desigualdade.	
> <	Testes “maior que” e “menor-que”.	
>= <=	Testes “maior ou igual que” e “menor ou igual que”	
ANY SOME	<p>Compara um valor com:</p> <ul style="list-style-type: none">▪ Cada um dos valores de uma lista▪ Ou o resultado de uma subquery. <p>Têm de ser precedidos pelos operadores: =, !=, >, <, >=, <=</p> <p>Podem ser seguidos por qualquer expressão / subquery que retorna um ou mais valores.</p> <p>Se o resultado da subquery for vazio:</p> <ul style="list-style-type: none">▪ Retorna FALSE.	<pre>SELECT * FROM employees WHERE salary = ANY (SELECT salary FROM employees WHERE department_id = 30) ORDER BY employee_id;</pre>
ALL	<p>Semelhante ao ANY/SOME.</p> <p>Se o resultado da subquery for vazio:</p> <ul style="list-style-type: none">▪ Retorna TRUE.	<pre>SELECT * FROM employees WHERE salary >= ALL (1400, 3000) ORDER BY employee_id;</pre>

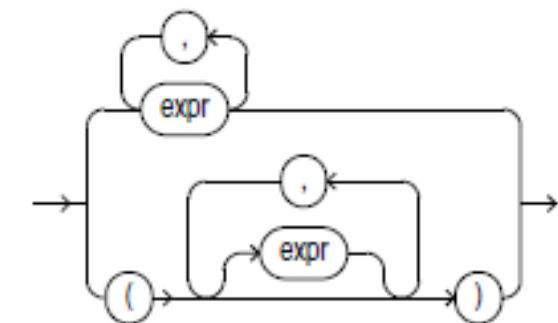
- Condição Simples

- Especifica comparação com resultados de:
 - Expressões
 - Subquery

- Sintaxe

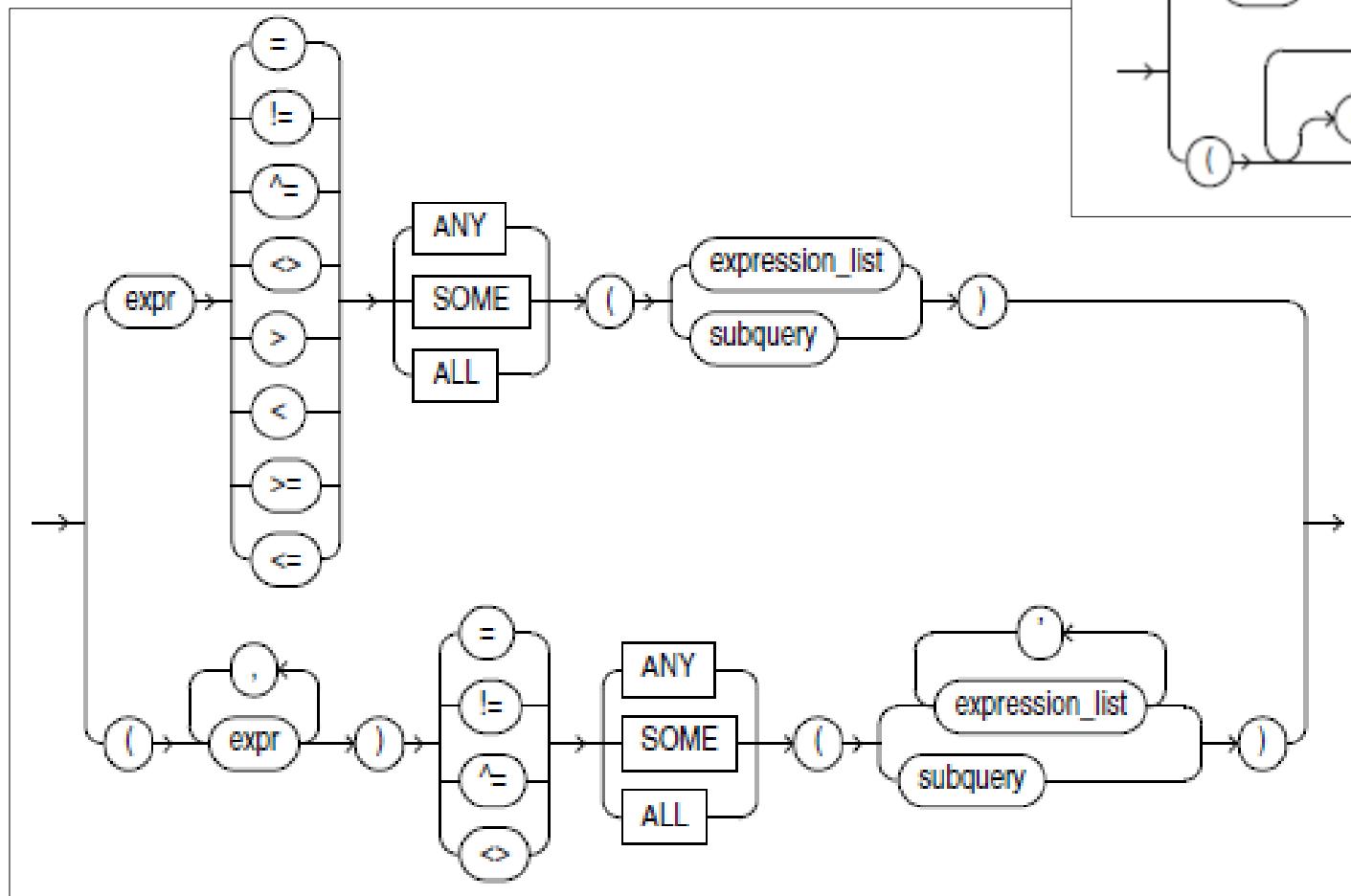


expression_list ::=

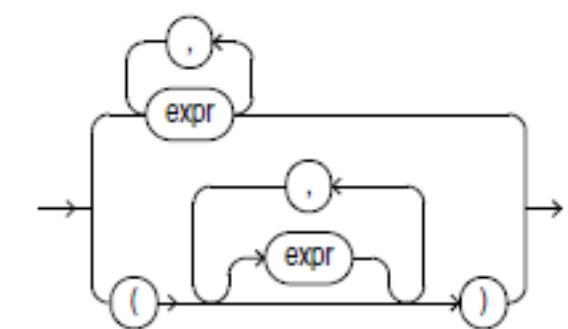


Condição de Grupos

- Especifica comparação com todos os elementos de:
 - Uma lista
 - Subquery
- Sintaxe



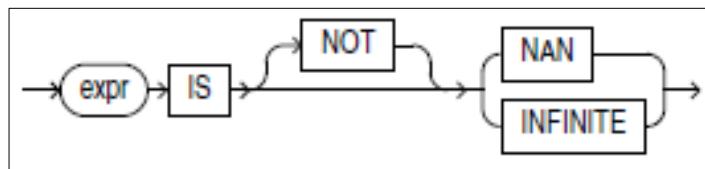
expression_list::=



■ Interesse

- Testar se resultado de expressão é:
 - Infinito
 - Indefinido -- NAN="not a number"; valor especial.

■ Sintaxe



Tipo da expr:

- Tipo numérico.
- Tipo convertível implicitamente em tipo numérico.

■ Formas

Forma	Propósito	Exemplo
expr IS NAN	expr = NAN \Rightarrow Resultado TRUE	<pre>SELECT COUNT(*) FROM employees WHERE commission_pct IS NOT NAN;</pre>
expr IS NOT NAN	expr \neq NAN \Rightarrow Resultado TRUE	
expr IS INFINITE	expr = + INF \Rightarrow Resultado TRUE expr = - INF \Rightarrow Resultado TRUE	
expr IS NOT INFINITE	expr \neq + INF \Rightarrow Resultado TRUE expr \neq - INF \Rightarrow Resultado TRUE	<pre>SELECT last_name FROM employees WHERE salary IS NOT INFINITE;</pre>

■ Interesse

- Combinar condições simples (dois operandos) para produzir um resultado.
- Inverte o resultado de uma simples condição.

■ Tipo de Condição

Tipo de Condição	Propósito				Exemplo
NOT expr (negação)	--	TRUE	FALSE	UNKNOWN	
	NOT	FALSE	TRUE	UNKNOWN	
expr1 AND expr2 (conjunção)	AND	TRUE	FALSE	UNKNOWN	
	TRUE	TRUE	FALSE	UNKNOWN	
	FALSE	FALSE	FALSE	FALSE	
	UNKNOWN	UNKNOWN	FALSE	UNKNOWN	
expr1 OR expr2 (disjunção)	OR	TRUE	FALSE	UNKNOWN	
	TRUE	TRUE	TRUE	TRUE	
	FALSE	TRUE	FALSE	UNKNOWN	
	UNKNOWN	TRUE	UNKNOWN	UNKNOWN	

```
SELECT *
FROM employees
WHERE NOT (job_id IS NULL)
ORDER BY employee_id;
```

```
SELECT *
FROM employees
WHERE job_id = 'PU_CLERK'
AND department_id = 30
ORDER BY employee_id;
```

```
SELECT *
FROM employees
WHERE job_id = 'PU_CLERK'
OR department_id = 10
ORDER BY employee_id;
```

- **Interesse**

- Comparar valores de tipos caráter.

- **Formas**

- Condição **LIKE**
 - Condição **REGEXP_LIKE**

- Condição LIKE (1/2)

- Especifica
 - Teste baseado em “pattern-matching”. -- procura de correspondência de um padrão.
- Sintaxe
 - 'valor de procura' [NOT] LIKE 'padrão de pesquisa' [ESCAPE 'caráter']
 - Padrão de pesquisa
 - Inclui caracteres especiais (“wildcards”)
 - Exemplos
 -  -- representa qualquer sequência de 0 ou mais caracteres.
 -  -- underscore - representa um só caráter qualquer.
 - ESCAPE
 - Anula contexto especial de caráter “wildcard”
 - Caráter (comprimento 1)
 - Exemplos
 - nome **LIKE 'T%**' -- nome começado por “T”
 - nome **NOT LIKE 'T%**' -- nome não começado por “T”
 - nome **LIKE '_a%**' -- nome cujo 2º caráter é “a”
 - mensagem **LIKE '%as'** -- mensagem terminada em “as”
 - mensagem **LIKE '%vendas%**' -- mensagem que contém a palavra “vendas”
 - mensagem **LIKE '%@%%' ESCAPE '@'** -- mensagem que contém uma “%”

Condições “Pattern-Matching”

▪ Condição LIKE (2/2)

▪ Padrões sem %

- Condição só pode ser TRUE:

- Se ambos os operandos tiverem o mesmo comprimento.

```
CREATE TABLE demo_like (
    local_CHAR    CHAR(10),
    local_VARCHAR VARCHAR(10)
);

INSERT INTO demo_like VALUES ('SEIXAL','SEIXAL');
INSERT INTO demo_like VALUES ('OLIVAL','OLIVAL');
INSERT INTO demo_like VALUES ('ALCOCHETE','ALCOCHETE');
```

Script Output | Query Result | All Rows Fetched: 3 in 0,004 seconds

LOCAL_CHAR	LOCAL_VARCHAR
1 SEIXAL	SEIXAL
2 OLIVAL	OLIVAL
3 ALCOCHETE	ALCOCHETE

Locais guardados em strings de comprimento 10.

Adicionados espaços para preencher as strings.

Exemplo: 'SEIXAL '

- Adicionados 4 espaços.

Cada local guardado em string de comprimento ajustado à sua dimensão.

Ex: 'SEIXAL'

```
SELECT *
FROM demo_like
WHERE local_char LIKE 'SEIXAL';
```

Script Output | Query Result | All Rows Fetched: 0 in 0,004 seconds

LOCAL_C...	LOCAL_V...
------------	------------

Resultado da query vazio porque:
'SEIXAL' ≠ 'SEIXAL'

Padrão 'SEIXAL%' resolve o problema.

```
SELECT *
FROM demo_like
WHERE local_varchar LIKE 'SEIXAL';
```

Script Output | Query Result | All Rows Fetched: 1 in 0,004 seconds

LOCAL_CHAR	LOCAL_VARCHAR
1 SEIXAL	SEIXAL

Problema não surge em tipos VARCHAR.

Condições “Pattern-Matching”

■ Condição REGEXP_LIKE (1/6)

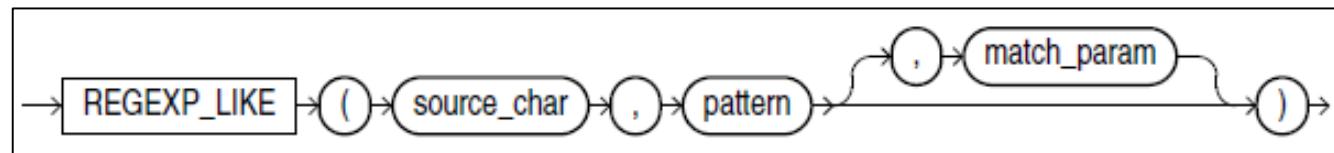
■ Interesse

- Semelhante ao operador LIKE.
- Exceto
 - Procura correspondências de uma expressão regular.
 - Procura de padrões mais complexos.

■ Expressão Regular

- Conforme a norma “POSIX regular expression”.

■ Sintaxe



- source_char: valor onde pesquisar correspondências da expressão regular indicada.
- pattern: expressão regular.
- match_parameter: string que permite alterar o funcionamento por omissão do operador.
Pode-se especificar um ou mais dos seguintes valores:

- 'i' case-insensitivity
- 'c' case-sensitivity
- 'n' ponto-final corresponde ao caráter “newline”.
- 'm' tratar source_char com múltiplas linhas.
- 'x' ignorar espaços.

■ Condição REGEXP_LIKE (2/6)

■ Funcionamento por Omissão

- source_char tratado como uma simples linha.
- O ponto final (.) não corresponde ao carácter “newline”.
- “Case sensitivity” determinado pelo valor do parâmetro NLS_SORT.

■ Condição REGEXP_LIKE (3/6)

■ Operadores de Expressões Regulares (1/2)

Operator	Description
\	The backslash character can have four different meanings depending on the context. It can: <ul style="list-style-type: none">▪ Stand for itself▪ Quote the next character▪ Introduce an operator▪ Do nothing
*	Matches zero or more occurrences
+	Matches one or more occurrences
?	Matches zero or one occurrence
	Alternation operator for specifying alternative matches
^	Matches the beginning of a string by default. In multiline mode, it matches the beginning of any line anywhere within the source string.
\$	Matches the end of a string by default. In multiline mode, it matches the end of any line anywhere within the source string.
.	Matches any character in the supported character set except NULL

Condições “Pattern-Matching”

Condição REGEXP_LIKE (4/6)

Operadores de Expressões Regulares (2/2)

Operator	Description
\$	Matches the end of a string by default. In multiline mode, it matches the end of any line anywhere within the source string.
.	Matches any character in the supported character set except NULL
[]	Bracket expression for specifying a matching list that should match any one of the expressions represented in the list. A non-matching list expression begins with a circumflex (^) and specifies a list that matches any character except for the expressions represented in the list. To specify a right bracket (]) in the bracket expression, place it first in the list (after the initial circumflex (^), if any). To specify a hyphen in the bracket expression, place it first in the list (after the initial circumflex (^), if any), last in the list, or as an ending range point in a range expression.
()	Grouping expression, treated as a single subexpression
{m}	Matches exactly m times
{m,}	Matches at least m times
{m,n}	Matches at least m times but no more than n times
\n	The backreference expression (n is a digit between 1 and 9) matches the n^{th} subexpression enclosed between '(' and ')' preceding the \n
[..]	Specifies one collation element, and can be a multicharacter element (for example, [.ch.] in Spanish)
[::]	Specifies character classes (for example, [:alpha:]). It matches any character within the character class.
[==]	Specifies equivalence classes. For example, [=a=] matches all characters having base letter 'a'.

■ Condição REGEXP_LIKE (5/6)

■ Exemplo 1

The following query returns the first and last names for those employees with a first name of Steven or Stephen (where `first_name` begins with `Ste` and ends with `en` and in between is either `v` or `ph`):

```
SELECT first_name, last_name
FROM employees
WHERE REGEXP_LIKE (first_name, '^Ste(v|ph)en$')
ORDER BY first_name, last_name;
```

FIRST_NAME	LAST_NAME

Steven	King
Steven	Markle
Stephen	Stiles

■ Condição REGEXP_LIKE (6/6)

■ Exemplo 2

The following query returns the last name for those employees with a double vowel in their last name (where last_name contains two adjacent occurrences of either a, e, i, o, or u, regardless of case):

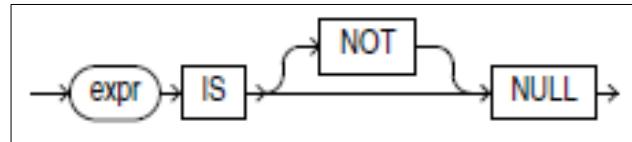
```
SELECT last_name
FROM employees
WHERE REGEXP_LIKE (last_name, '([aeiou])\1', 'i')
ORDER BY last_name;
```

LAST_NAME

```
-----  
De Haan  
Greenberg  
Khoo  
Gee  
Greene  
Lee
```

■ Interesse

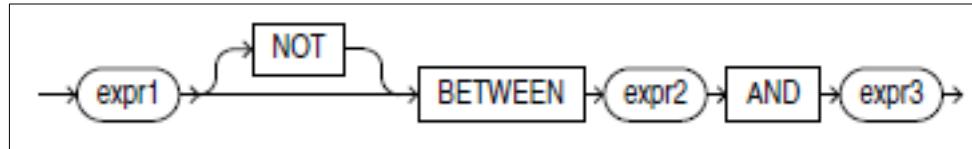
- Testar o valor NULL. -- única condição para testar NULL.

■ Sintaxe**■ Formas**

Forma	Propósito	Exemplo
expr IS NULL	expr = NULL \Rightarrow Resultado TRUE expr \neq NULL \Rightarrow Resultado FALSE	<pre>SELECT last_name FROM employees WHERE commission_pct IS NULL ORDER BY last_name;</pre>
expr IS NOT NULL	expr = NULL \Rightarrow Resultado FALSE expr \neq NULL \Rightarrow Resultado TRUE	

■ Interesse

- Testar se o valor de uma expressão ... está num intervalo definido por outras duas expressões.

■ Sintaxe

- Todas as expressões (expr1, expr2 e expr3):

- Devem ser do mesmo tipo de dados:
 - Tipo numérico.
 - Tipo caráter.
 - Tipo data-tempo.
- Não são todas do mesmo tipo:
 - Convertidas implicitamente para um tipo comum.
 - Se não for possível \Rightarrow gerada exceção.

expr1 BETWEEN expr2 AND expr3:

- $expr1 \geq expr2 \text{ AND } expr1 \leq expr3$.

Example

```
SELECT * FROM employees
WHERE salary
BETWEEN 2000 AND 3000
ORDER BY employee_id;
```

■ Semântica

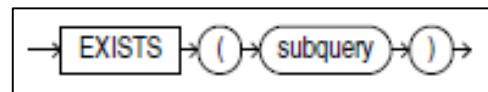
- $expr3 < expr2 \Rightarrow$ resultado vazio.
- $expr1 = \text{NULL} \Rightarrow$ resultado NULL.
- $expr1 \neq \text{NULL} \Rightarrow$ resultado FALSE / TRUE.

■ Exemplos

Exemplo	Alternativas
idade BETWEEN 30 AND 40	idade \geq 30 AND idade \leq 40
idade NOT BETWEEN 30 AND 40	NOT (idade BETWEEN 30 AND 40) NOT (idade \geq 30 AND idade \leq 40)

■ Interesse

- Testar a existência de linhas numa *subquery*. -- conjunto [não] vazio.

■ Sintaxe**■ Semântica**

- Retorna TRUE ... quando *subquery* retorna pelo menos uma linha.

■ Exemplos

- Testar a existência ...
- Testar conjunto vazio ...

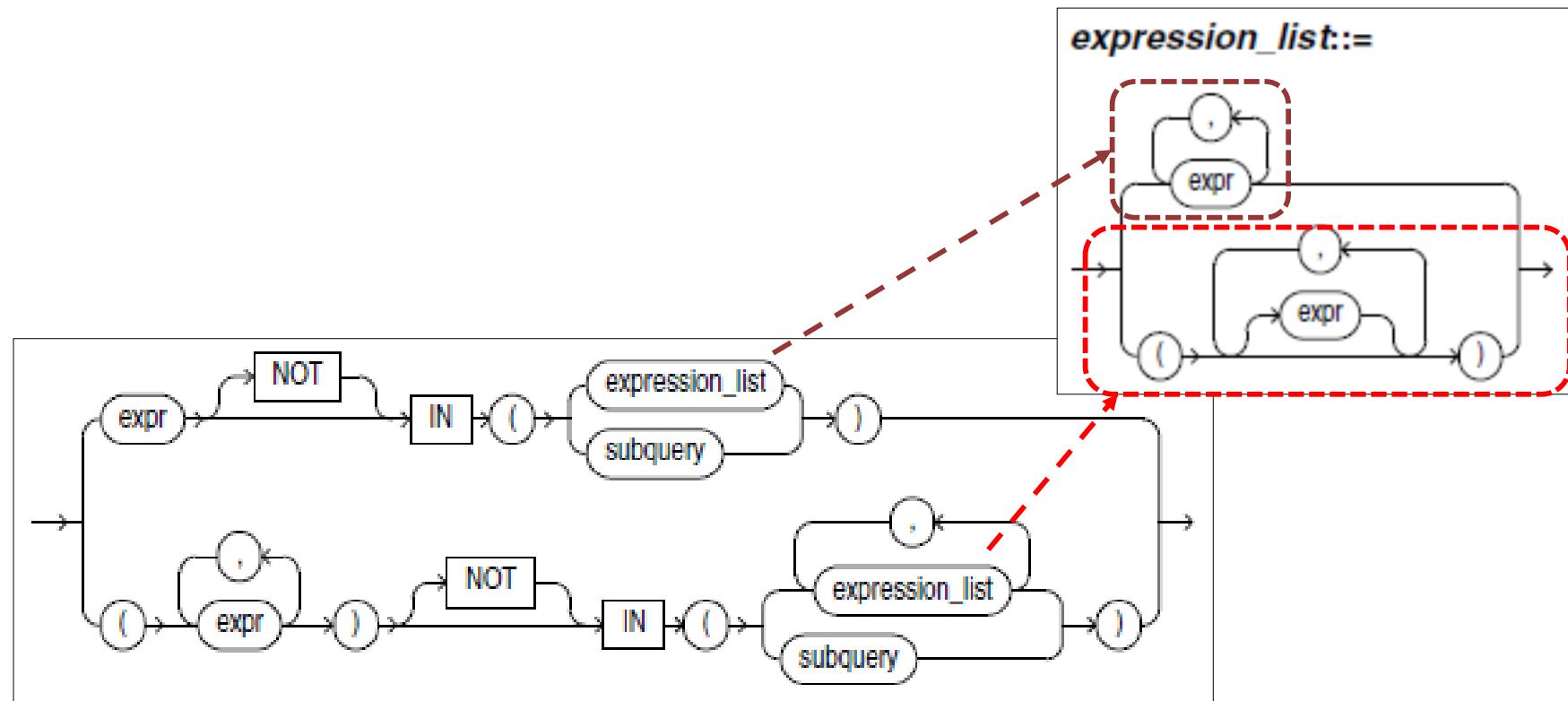
Example

```
SELECT department_id
  FROM departments d
 WHERE EXISTS
   (SELECT * FROM employees e
    WHERE d.department_id
      = e.department_id)
 ORDER BY department_id;
```

■ Interesse

- Testar se um valor é membro de conjunto.
- Conjunto é definido de uma das seguintes formas:
 - Lista de valores.
 - Resultado de *subquery*.

■ Sintaxe

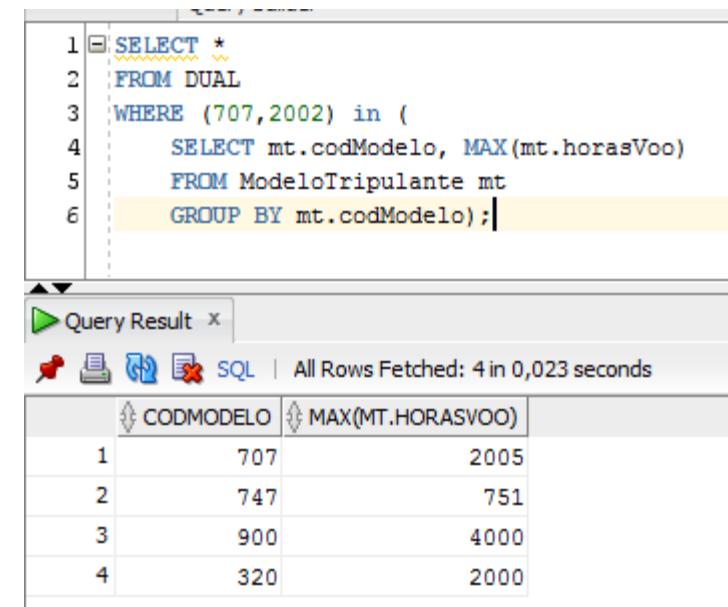


■ Semântica

Tipo de Condição	Funcionamento	Exemplo
IN	<p>Equivalente:</p> <ul style="list-style-type: none">▪ = ANY▪ Exemplo:<ul style="list-style-type: none">▪ $\text{expr IN (e1, e2, e3)} \Leftrightarrow \text{expr=e1 OR expr=e2 OR expr=e3}$	<ul style="list-style-type: none">▪ Lista de valores:<pre>SELECT * FROM employees WHERE job_id IN ('PU_CLERK', 'SH_CLERK') ORDER BY employee_id;</pre>▪ Subquery:<pre>SELECT * FROM employees WHERE salary IN (SELECT salary FROM employees WHERE department_id =30) ORDER BY employee_id;</pre>
NOT IN	<p>Equivalente:</p> <ul style="list-style-type: none">▪ ≠ ALL▪ Exemplo:<ul style="list-style-type: none">▪ $\text{expr NOT IN (e1, e2, e3)} \Leftrightarrow \text{NOT(expr=e1 OR expr=e2 OR expr=e3)} \Leftrightarrow \text{expr!=e1 AND expr!=e2 AND expr!=e3}$ <p>FALSE:</p> <ul style="list-style-type: none">▪ Se conjunto tiver qualquer membro NULL.▪ Exemplo no slide seguinte.	

■ Exemplos

Exemplo	Alternativas
localidade IN ('Porto', 'Braga','Matosinhos')	localidade='Porto' OR localidade='Braga' OR localidade='Matosinhos'
localidade NOT IN ('Porto', 'Braga', 'Matosinhos')	NOT(localidade='Porto' OR localidade='Braga' OR localidade='Matosinhos') localidade!=‘Porto’ AND localidade!=‘Braga’ AND localidade=‘Matosinhos’
codigoPostal IN (1000, 1100, 1200)	codigoPostal=1000 OR codigoPostal=1100 OR codigoPostal=1200



The screenshot shows an Oracle SQL developer interface. In the SQL editor, there is a query:

```
1 | SELECT *
2 | FROM DUAL
3 | WHERE (707,2002) in (
4 |     SELECT mt.codModelo, MAX(mt.horasVoo)
5 |     FROM ModeloTripulante mt
6 |     GROUP BY mt.codModelo);
```

In the Query Result window, the output is:

CODMODELO	MAX(MT.HORASVOO)
1	2005
2	751
3	4000
4	2000

Condição IN

NOT IN

Worksheet Query Builder

```
SELECT * FROM demo_in;
```

Script Output Query Result

SQL | All Rows Fetched

NUMERO	
1	1
2	2
3	3
4	4
5	5

Resultado da query é vazio.

- Condição NOT IN é equivalente a:
 - numero!=3 AND numero!=4 AND numero!=10 AND numero!=NULL \Rightarrow resultado UNKNOWN
- Assim, para os números existentes na tabela (3 e 4):
 - Resultado global é FALSE \Rightarrow resultado vazio.

Resultado da query não é vazio.

Worksheet Query Builder

```
SELECT *  
FROM demo_in  
WHERE numero NOT IN(3,4,10);
```

Script Output Query Result

SQL | All Rows Fetched: 3 in 0

NUMERO	
1	1
2	2
3	5

Lista de valores com sem valores NULL.

Worksheet Query Builder

```
SELECT *  
FROM demo_in  
WHERE numero NOT IN(3,4,10,NULL);
```

Script Output Query Result

SQL | All Rows Fetched: 0 in 0,003 se

NUMERO	
--------	--

Lista de valores com um elemento NULL.

■ NOT IN

Worksheet Query Builder

```
SELECT * FROM demo_in;
```

Script Output Query Result

SQL | All Rows Fetched

	NUMERO
1	1
2	2
3	3
4	4
5	5

Worksheet Query Builder

```
SELECT *  
FROM demo_in  
WHERE numero NOT IN(SELECT * FROM demo_in WHERE numero=0);
```

Script Output Query Result

SQL | All Rows Fetched: 5 in 0,009 seconds

	NUMERO
1	5
2	4
3	3
4	1
5	2

Subquery com resultado vazio.

Retornadas todas as linhas.

Funções SQL

- [Introdução](#)
- [Categorias](#)
- [Funções Escalares](#)
 - [Numéricas](#)
 - [Carateres](#)
 - [Retorno Carateres](#)
 - [Retorno Numérico](#)
 - [Data-Tempo](#)
 - [Intervalo](#)
 - [Comparação Geral](#)
 - [Conversão](#)
 - [Relacionadas com NULL](#)
- [Funções de Agregação](#)
- [Bibliografia](#)

■ Função

- Semelhante aos operadores.
 - Diferença no formato dos seus argumentos.
 - Formato permite operar sobre zero, um, dois ou mais argumentos.
- Manipula dados e retorna um resultado.

função(argumento1, argumento2, ...)

■ Argumentos de Tipos Diferentes dos Declarados

- Oracle tenta converter:
 - Para o tipo esperado antes de executar a função.

■ Argumentos NULL

- Maioria das funções escalares -- funções cujo resultado é baseado num valor simples.
 - Retornam NULL
- Função NVL
 - Permite retornar um valor quando ocorre um NULL.
 - Exemplo:
 - NVL(comissao, 0)
 - Retorna 0 (expr2) quando a comissao (expr1) é NULL.
 - Caso contrário, retorna comissao (expr1).

Syntax



■ Sem Argumentos

- Especificadas sem parêntesis; Especificado apenas o nome da função.

- **Categorias**

- Escalares
 - Agregação

- **Funções Escalares**

- Retornam:
 - Um valor baseado num valor simples.
 - Uma simples linha por cada linha da tabela/vista consultada.
 - Podem ser usadas no comando SELECT
 - Cláusulas
 - SELECT
 - WHERE
 - HAVING
 - Exemplo:
 - UPPER – converte um valor *string* para letras maiúscula.

- **Funções de Agregação**

- Retornam:
 - Um valor sobre um conjunto de linhas especificadas na cláusula WHERE.
 - Um valor sobre cada grupo de linhas indicados na cláusula GROUP BY.
 - Exemplo:
 - SUM – retorna soma dos valores dumha coluna.

- **Categorias de Funções Escalares**

- Numéricas
- Carateres
- Data-Tempo
- Comparação Geral
- Conversão
- Relacionadas com NULL

Funções SQL

Numéricas

Carateres

Data-Tempo

Comparação Geral

Conversão

- **Características Gerais**

- Recebem e retornam:
 - Valores numéricos.
- Maioria:
 - Retorna valores NUMBER -- precisão de 38 dígitos decimais (nºs inteiros $\in [0, 9]$).

- **Categorias**

- [Trigonometrícias](#)
- [Logaritmicas](#)
- [Arredondamento](#)
- [Resto da Divisão](#)
- [Outras](#)

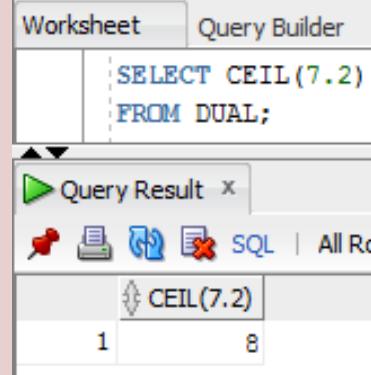
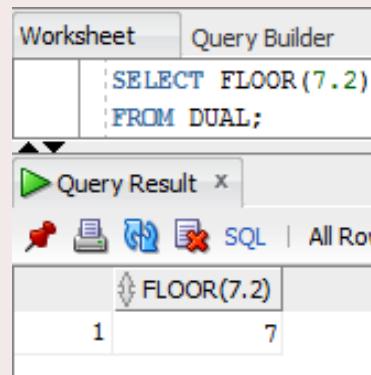
▪ Funções Trigonométricas

Função	Funcionalidade
COS(n)	Retorna o coseno de <u>n</u> . n: ângulo expresso em radianos.
SIN(n)	Retorna o seno de <u>n</u> .
TAN(n)	Retorna a tangente de <u>n</u> .
ACOS(n)	Retorna o arco do coseno de <u>n</u> .
ASIN(n)	Retorna o arco do seno de <u>n</u> .
ATAN(n)	Retorna o arco da tangente de <u>n</u> .
COSH(n)	Retorna o arco do coseno hiperbólico de <u>n</u> .
SINH(n)	Retorna o arco do seno hiperbólico de <u>n</u> .
TANH(n)	Retorna o arco da tangente hiperbólica de <u>n</u> .
ATAN2(n1, n2)	Retorna o arco da tangente de <u>n1</u> e <u>n2</u> .

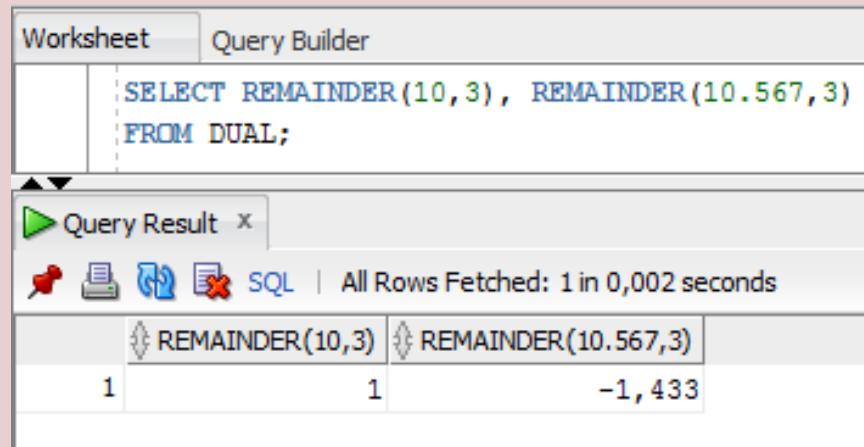
- Funções Logaritmicas

Função	Funcionalidade
LOG(n2, n1)	Retorna o logaritmo, na base <u>n2</u> , de <u>n1</u> .
LN(n)	Retorna o logaritmo natural de <u>n</u> (n>0).

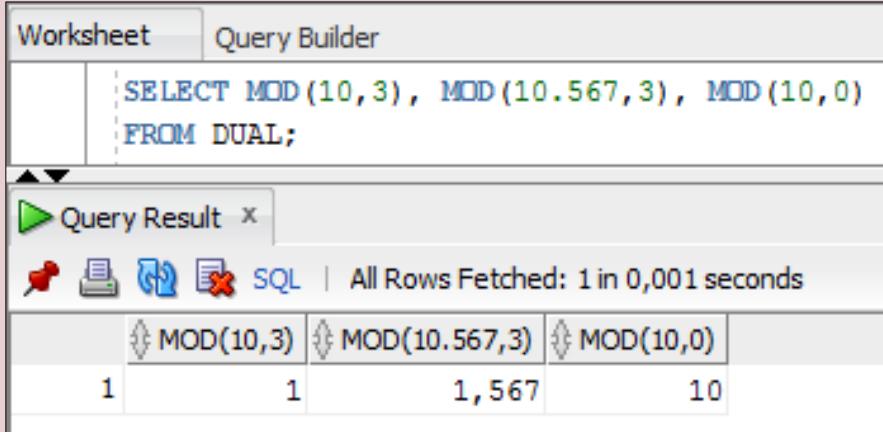
- Funções de Arredondamento

Função	Funcionalidade
CEIL(n)	<p>Retorna o menor <u>inteiro</u> $\geq n$. Arredonda por excesso.</p> 
FLOOR(n)	<p>Retorna o maior <u>inteiro</u> $\leq n$. Arredonda por defeito.</p> 
ROUND(n)	Arredonda o número <u>n</u> para zero casas decimais ($n \geq 0$).
ROUND(n, integer)	Arredonda <u>n</u> para <u>integer</u> casas decimais.
TRUNC(n)	Retorna <u>n</u> truncado das casas decimais (sem casas decimais).
TRUNC(n1, n2)	Retorna <u>n1</u> truncado com <u>n2</u> casas decimais.

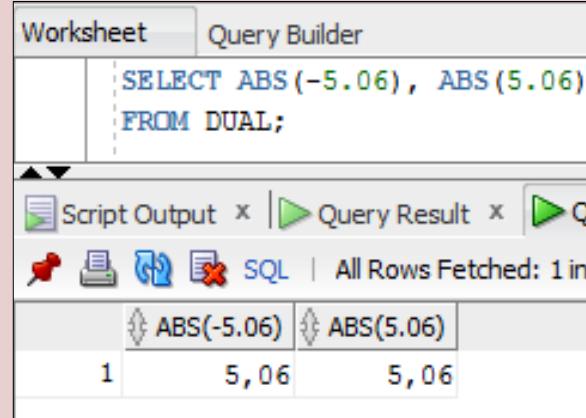
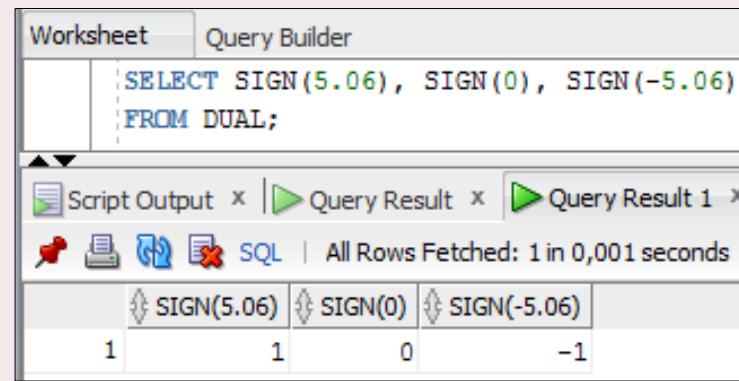
■ Funções de Resto da Divisão (1/2)

Função	Funcionalidade				
REMAINDER(n2, n1)	<p>Retorna o resto da divisão de <u>n2</u> por <u>n1</u>. Usa a função ROUND() para arredondar o resultado. Se <u>n1</u>=0 ou <u>n2</u>=infinito:</p> <ul style="list-style-type: none">▪ Gerada exceção se argumentos são do tipo NUMBER.▪ Retornado o valor NaN se argumentos são do tipo BINARY_FLOAT / BINARY_DOUBLE.  <p>The screenshot shows the Oracle SQL Developer interface. In the Worksheet tab, the following SQL query is entered:</p> <pre>SELECT REMAINDER(10,3), REMAINDER(10.567,3) FROM DUAL;</pre> <p>In the Query Result tab, the output is displayed in a grid:</p> <table border="1"><thead><tr><th>REMAINDER(10,3)</th><th>REMAINDER(10.567,3)</th></tr></thead><tbody><tr><td>1</td><td>-1,433</td></tr></tbody></table> <p>The status bar at the bottom indicates "All Rows Fetched: 1 in 0,002 seconds".</p>	REMAINDER(10,3)	REMAINDER(10.567,3)	1	-1,433
REMAINDER(10,3)	REMAINDER(10.567,3)				
1	-1,433				

- Funções de Resto da Divisão (2/2)

Função	Funcionalidade								
MOD(n2, n1)	<p>Retorna o resto da divisão de n2 por n1.</p> <p>Retorna n2 se n1 = 0.</p> <p>Usa a função FLOOR() para arredondar o resultado. -- arredonda por defeito.</p>  <p>The screenshot shows the Oracle SQL Developer interface. In the 'Worksheet' tab, the following SQL code is entered:</p> <pre>SELECT MOD(10,3), MOD(10.567,3), MOD(10,0) FROM DUAL;</pre> <p>In the 'Query Result' tab, the output is displayed in a grid:</p> <table border="1"><thead><tr><th></th><th>MOD(10,3)</th><th>MOD(10.567,3)</th><th>MOD(10,0)</th></tr></thead><tbody><tr><td></td><td>1</td><td>1,567</td><td>10</td></tr></tbody></table>		MOD(10,3)	MOD(10.567,3)	MOD(10,0)		1	1,567	10
	MOD(10,3)	MOD(10.567,3)	MOD(10,0)						
	1	1,567	10						

- Outras Funções (1/3)

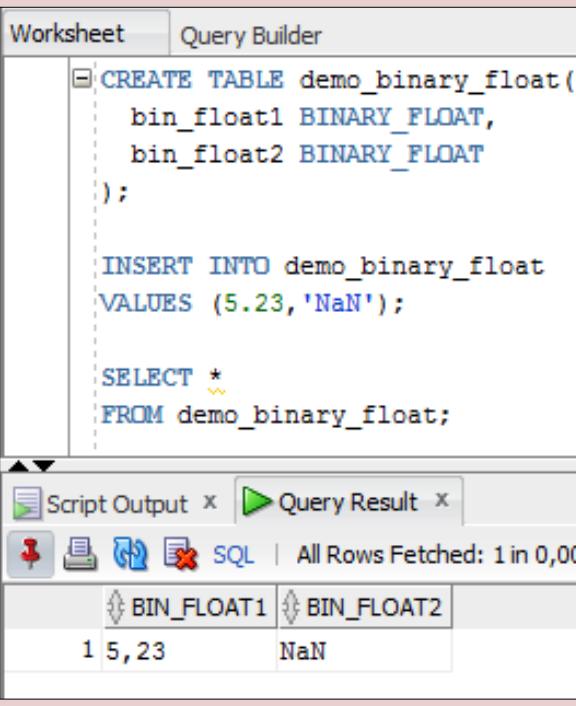
Função	Funcionalidade
ABS(n)	<p>Retorna o valor absoluto de <u>n</u>.</p>  <pre>Worksheet Query Builder SELECT ABS(-5.06), ABS(5.06) FROM DUAL; Script Output x Query Result x Query Result 1 x SQL All Rows Fetched: 1 in 0,001 seconds ABS(-5.06) ABS(5.06) 5,06 5,06</pre>
SIGN(n)	<p>Retorna 1, 0 ou -1, para <u>n</u>>0, <u>n</u>=0 ou <u>n</u><0, respectivamente.</p>  <pre>Worksheet Query Builder SELECT SIGN(5.06), SIGN(0), SIGN(-5.06) FROM DUAL; Script Output x Query Result x Query Result 1 x SQL All Rows Fetched: 1 in 0,001 seconds SIGN(5.06) SIGN(0) SIGN(-5.06) 1 0 -1</pre>

■ Outras Funções (2/3)

Função	Funcionalidade															
SQRT(n)	<p>Retorna a raiz quadrada de <u>n</u>.</p> <p><u>n</u><0:</p> <ul style="list-style-type: none">▪ BINARY_FLOAT / BINARY_DOUBLE \Rightarrow NaN (“not a number”)▪ NUMBER \Rightarrow lançada exceção. <p>The screenshot shows two Oracle SQL Developer windows. The top window displays a script for creating a table 'demo_sqrt' with columns 'num' (NUMBER) and 'bin_float' (BINARY_FLOAT). It also contains INSERT statements for values (5.23, 5.23) and (0, -5.23), followed by a SELECT query. The bottom window shows the execution results in two tabs: 'Script Output' and 'Query Result'. The 'Script Output' tab shows the table creation and insert statements. The 'Query Result' tab shows the output of the SELECT query, which returns two rows: (5.23, 5.23) and (0, -5.23). The 'Query Result' tab also includes a 'SQL' tab showing the executed SQL statement.</p> <p>Worksheet Query Builder</p> <pre>CREATE TABLE demo_sqrt(num NUMBER, bin_float BINARY_FLOAT); INSERT INTO demo_sqrt VALUES (5.23,5.23); INSERT INTO demo_sqrt VALUES (0,-5.23); SELECT * FROM demo_sqrt;</pre> <p>Script Output Query Result SQL All Rows Fetched</p> <table border="1"><thead><tr><th></th><th>NUM</th><th>BIN_FLOAT</th></tr></thead><tbody><tr><td>1</td><td>5,23</td><td>5,23</td></tr><tr><td>2</td><td>0</td><td>-5,23</td></tr></tbody></table> <p>Worksheet Query Builder</p> <pre>SELECT SQRT(num), SQRT(bin_float) FROM demo_sqrt;</pre> <p>Script Output Query Result Query Result 1 Query Result 2</p> <pre>SELECT SQRT(num), SQRT(bin_float) FROM demo_sqrt;</pre> <p>All Rows Fetched: 2 in 0,002 seconds</p> <table border="1"><thead><tr><th>SQRT(NUM)</th><th>SQRT(BIN_FLOAT)</th></tr></thead><tbody><tr><td>2,2869193252058543046676174301097857356</td><td>2,2869194</td></tr><tr><td>2</td><td>0 NaN</td></tr></tbody></table>		NUM	BIN_FLOAT	1	5,23	5,23	2	0	-5,23	SQRT(NUM)	SQRT(BIN_FLOAT)	2,2869193252058543046676174301097857356	2,2869194	2	0 NaN
	NUM	BIN_FLOAT														
1	5,23	5,23														
2	0	-5,23														
SQRT(NUM)	SQRT(BIN_FLOAT)															
2,2869193252058543046676174301097857356	2,2869194															
2	0 NaN															

- Outras Funções (3/3)

Função	Funcionalidade
EXP(n)	Retorna e elevado a <u>n</u> . -- e = 2.7182...
POWER(n2, n1)	Retorna <u>n2</u> elevado a <u>n1</u> . Se n2<0 então n1 tem de ser inteiro.
NANVL(n2,n1)	Retorna o valor alternativo <u>n1</u> quando <u>n2</u> é NaN (“not a number”). Útil para números BINARY_FLOAT e BINARY_DOUBLE.



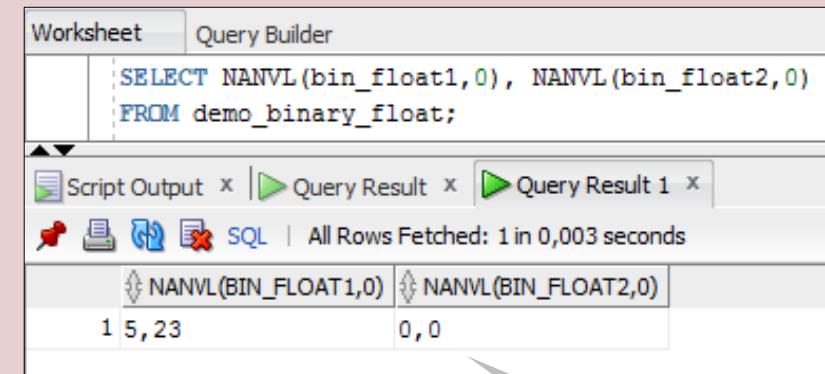
```

CREATE TABLE demo_binary_float(
    bin_float1 BINARY_FLOAT,
    bin_float2 BINARY_FLOAT
);

INSERT INTO demo_binary_float
VALUES (5.23, 'NaN');

SELECT *
FROM demo_binary_float;
  
```

	BIN_FLOAT1	BIN_FLOAT2
1	5,23	NaN



```

SELECT NANVL(bin_float1,0), NANVL(bin_float2,0)
FROM demo_binary_float;
  
```

NANVL(BIN_FLOAT1,0)	NANVL(BIN_FLOAT2,0)
5,23	0,0

Valor NaN substituído por zero.

Funções SQL

Numéricas

Carateres

Data-Tempo

Comparação Geral

Conversão

Categorias

- Retornam valores de carateres.
- Retornam valores numéricos.

- **Características Gerais**

- Retornam valores do tipo VARCHAR2.
 - Comprimento máximo = comprimento máximo do tipo.
 - Valor pode ser truncado -- sem mensagem de erro.

- **Categorias**

- Junção de carateres.
- Conversão de letras para maiúsculas ou minúsculas.
- Retorna parte de *string*.
- Adição de carateres.
- Substituição de carateres.
- Conversão de código para caráter.

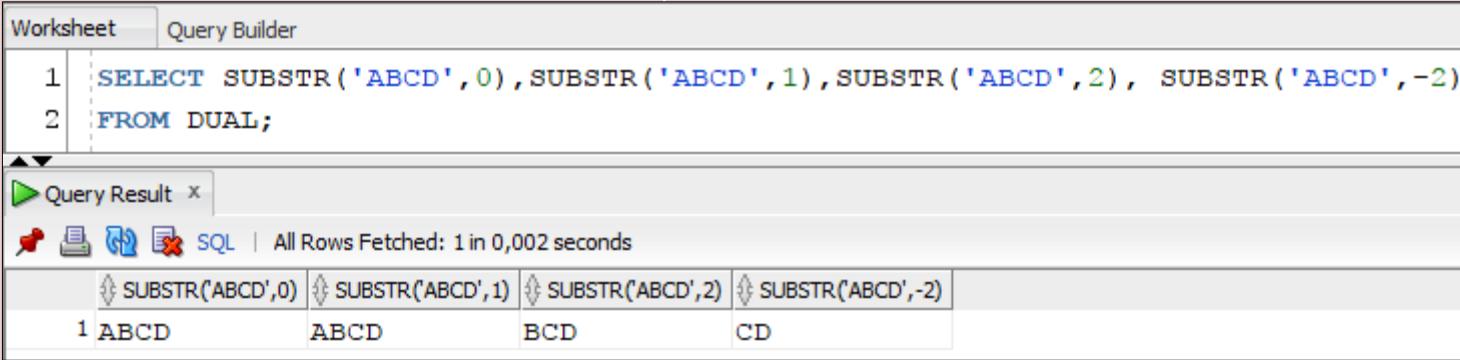
- Junção de Carateres

Função	Funcionalidade
CONCAT(char1, char2)	<p>Retorna a concatenação das <i>strings</i> <u>char1</u> e <u>char2</u>.</p> <p>Alternativa ao operador :</p> <ul style="list-style-type: none">Independente do <i>character set</i> da BD.Útil para a <u>portabilidade</u> de aplicações.

- Conversão de Letras para Maiúsculas ou Minúsculas

Função	Funcionalidade
INITCAP(char)	Retorna a <i>string char</i> com a primeira letra de cada palavra em maiúscula e as restantes letras minúsculas.
LOWER(char)	Retorna a <i>string char</i> com todas as letras minúsculas.
UPPER(char)	Retorna a <i>string char</i> com todas as letras maiúsculas.

- Retorna Parte de *String*

Função	Funcionalidade										
SUBSTR(char, position)	<p>Retorna uma parte de <u>char</u>, desde a <u>position</u> (até ao fim).</p> <p><u>position</u>=0 \Leftrightarrow <u>position</u>=1</p> <p><u>position</u>>0 \Rightarrow contagem desde o início de <u>char</u></p> <p><u>position</u><0 \Rightarrow contagem desde o fim de <u>char</u></p>										
 <pre>Worksheet Query Builder 1 SELECT SUBSTR('ABCD', 0), SUBSTR('ABCD', 1), SUBSTR('ABCD', 2), SUBSTR('ABCD', -2) 2 FROM DUAL;</pre> <p>Query Result All Rows Fetched: 1 in 0,002 seconds</p> <table border="1"> <thead> <tr> <th></th> <th>SUBSTR('ABCD',0)</th> <th>SUBSTR('ABCD',1)</th> <th>SUBSTR('ABCD',2)</th> <th>SUBSTR('ABCD',-2)</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>ABCD</td> <td>ABCD</td> <td>BCD</td> <td>CD</td> </tr> </tbody> </table>		SUBSTR('ABCD',0)	SUBSTR('ABCD',1)	SUBSTR('ABCD',2)	SUBSTR('ABCD',-2)	1	ABCD	ABCD	BCD	CD	
	SUBSTR('ABCD',0)	SUBSTR('ABCD',1)	SUBSTR('ABCD',2)	SUBSTR('ABCD',-2)							
1	ABCD	ABCD	BCD	CD							
SUBSTR(char, position, substring_length)	<p>Retorna uma parte de <u>char</u>, desde a <u>position</u> e com <u>substring_length</u> de caracteres.</p> <p><u>substring_length</u><1 \Rightarrow retorno NULL.</p>										
REGEXP_SUBSTR(source_char, pattern)	Estende a funcionalidade da função SUBSTR, permitindo pesquisar numa <i>string</i> correspondências de uma expressão regular.										

- Eliminação de Carateres

Função	Funcionalidade
TRIM(trim_source)	Elimina espaços no início e no fim da <i>string trim_source</i> .
TRIM(trim_character, trim_source)	Elimina o caráter <u>trim_character</u> no início e no fim da <i>string trim_source</i> .
TRIM(LEADING FROM trim_source)	Elimina espaços no início de <u>trim_source</u> .
TRIM(LEADING trim_character FROM trim_source)	Elimina <u>trim_character</u> no início de <u>trim_source</u> .
TRIM(TRAILING FROM trim_source)	Elimina espaços no fim de <u>trim_source</u> .
TRIM(TRAILING trim_character FROM trim_source)	Elimina <u>trim_character</u> no fim de <u>trim_source</u> .
TRIM(BOTH FROM trim_source)	Equivalente a TRIM(trim_source).
TRIM(BOTH trim_character FROM trim_source)	Elimina <u>trim_source</u> no início e no fim da <i>string trim_source</i> .
LTRIM(char, set)	Elimina do início da <i>string char</i> os carateres contidos na <i>string set</i> . -- L=LEFT
LTRIM(char)	Elimina espaços no início da <i>string char</i> .
RTRIM(char, set)	Semelhante a LTRIM. -- R=Right.
RTRIM(char)	Elimina espaços no fim da <i>string char</i> .

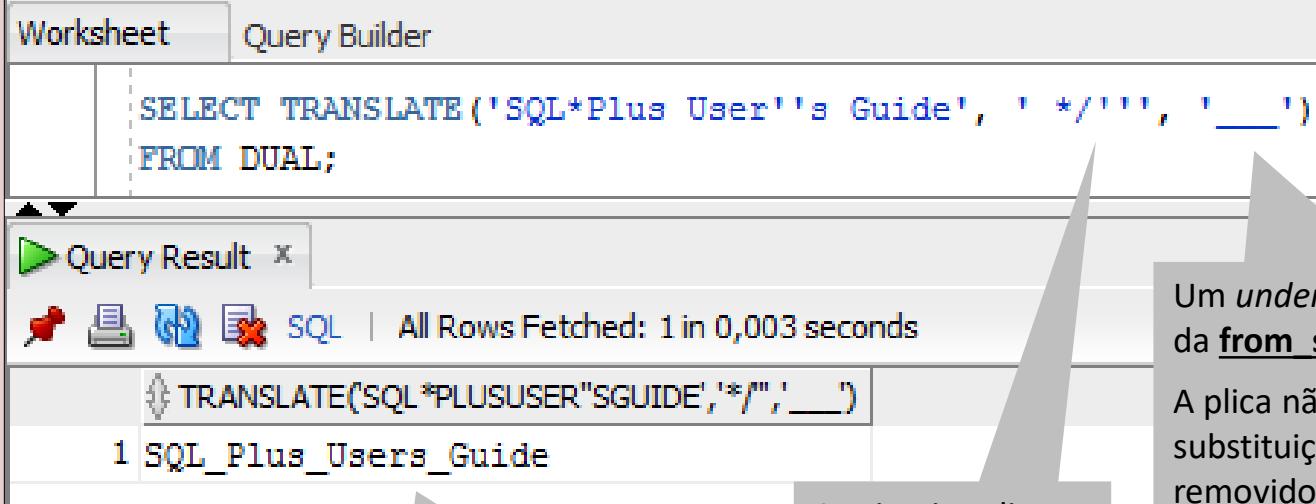
- Adição de Carateres

Função	Funcionalidade
LPAD(expr1, n, expr2)	<p>Retorna <u>expr1</u> numa <i>string</i> de <u>n</u> carateres, sendo adicionados à <u>esquerda</u> uma sequência da <i>string</i> <u>expr2</u>.</p> <pre data-bbox="725 400 1628 688">SELECT LPAD('Page 1',15,'*.*') "LPAD example" FROM DUAL;</pre> <p>LPAD example ----- *.*.*.*Page 1</p> <p>Se n < comprimento(expr1)</p> <ul style="list-style-type: none"> ▪ Retornada expr1 que se encaixa em n carateres.
LPAD(expr1, n)	<p>Semelhante à função anterior, sendo <u>expr2</u>=espaço.</p> <p>Funções <u>úteis</u> para formatação da saída de uma <i>query</i>.</p>
RPAD(expr1, n, expr2)	<p>Semelhante a LPAD(expr1, n, expr2).</p> <p>Adição de carateres é feita à direita.</p>
RPAD(expr1, n)	<p>Semelhante a LPAD(expr1, n).</p> <p>Adição de carateres é feita à direita.</p>

- Substituição de Caracteres (1/2)

Função	Funcionalidade
REPLACE(char, search_string)	<p>Remove de <u>char</u> todas as ocorrências de <u>search_string</u>.</p> <ul style="list-style-type: none"> ▪ <u>search_string</u>=NULL ⇒ retornado <u>char</u>.
REPLACE(char, search_string, replacement_string)	<p>Substitui em <u>char</u> todas as ocorrências de <u>search_string</u> por <u>replacement_string</u>.</p> <ul style="list-style-type: none"> ▪ <u>replacement_string</u>=NULL ⇒ (= anterior).
REGEXP_REPLACE(source_char, pattern) REGEXP_REPLACE(source_char, pattern, replace_string) REGEXP_REPLACE(source_char, pattern, replace_string, position)	<p>Estendem as funcionalidades de REPLACE permitindo pesquisar numa <i>string</i> correspondências de uma expressão regular.</p>

▪ Substituição de Caracteres (2/2)

Função	Funcionalidade
TRANSLATE(expr, from_string, to_string)	<p>Cada carácter de <u>from_string</u>, existente em <u>expr</u>, é substituído pelo carácter correspondente de <u>to_string</u>.</p> <p>Um carácter de <u>from_string</u> que não tenha carácter correspondente em <u>to_string</u> é removido de <u>expr</u>.</p>  <p>String pode ser usada como nome de ficheiro.</p> <p>A primeira plica serve de carácter de escape da segunda plica.</p> <p>Um underscore para cada carácter da <u>from_string</u>: espaço, * e /.</p> <p>A plica não tem um carácter de substituição associado. Assim é removido em vez de substituído.</p>

- Conversão de Código para Caráter

Função	Funcionalidade
CHR(n)	<p>Retorna o caráter com o código binário <u>n</u> no <i>character set</i> da BD.</p> <pre>SELECT CHR(67) CHR(65) CHR(84) "Dog" FROM DUAL;</pre> <p>Dog --- CAT</p> <p><i>Character-set:</i> WE8ISO8859P1</p>

■ Características Gerais

- Argumentos
 - Podem ser de qualquer tipo de caracteres.

Função	Funcionalidade
ASCII(char)	Retorna representação decimal, do primeiro caráter de <u>char</u> , no <i>character set</i> da BD.
INSTR(string, substring)	Retorna posição da <u>substring</u> na <u>string</u> (índice do primeiro caráter da <u>substring</u>).
INSTR(string, substring, position)	Retorna posição da <u>substring</u> na <u>string</u> (índice do primeiro caráter da <u>substring</u>) procurada a partir da <u>position</u> . <u>position<0:</u> <ul style="list-style-type: none">▪ Contagem da <u>position</u> feita para trás, a partir do fim da <u>string</u>.▪ Procura da <u>substring</u> é feita para trás.
INSTR(string, substring, position, occurrence)	Retorna a posição da n-ésima <u>occurrence</u> da <u>substring</u> procurada na <u>string</u> a partir da <u>position</u> (índice do primeiro caráter da <u>substring</u>).
REGEXP_INSTR	Estende a função INSTR, permitindo pesquisar numa <i>string</i> correspondências de uma expressão regular.
LENGTH(char)	Retorna o comprimento de <u>char</u> em caracteres definidos pelo <i>character set</i> da BD.
LENGTHB(char)	Retorna o comprimento de <u>char</u> em bytes.
REGEXP_COUNT	Completa a função REGEXP_INSTR, retornando o número de ocorrências do padrão especificado.

Funções SQL

Numéricas

Carateres

Data-Tempo

Comparação Geral

Conversão

■ Características Gerais

- Operam sobre valores:
 - DATE
 - TIMESTAMP
 - INTERVAL
- Algumas funções criadas para o tipo DATE:
 - Com argumentos e retorno do tipo DATE:
 - ADD_MONTHS
 - CURRENT_DATE
 - LAST_DAY
 - NEW_TIME
 - NEXT_DAY
 - ROUND
 - TRUNC
 - Com argumentos do tipo DATE e retorno do tipo NUMBER
 - MONTHS_BETWEEN

Argumentos TIMESTAMP:

- Convertidos implicitamente para DATE.

Argumentos TIMESTAMP:

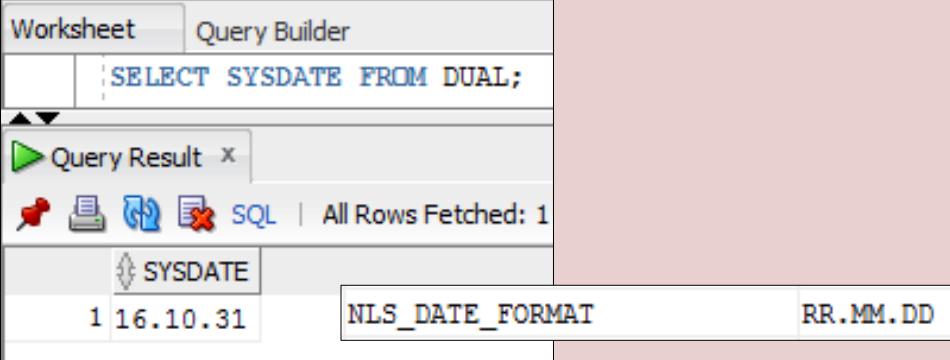
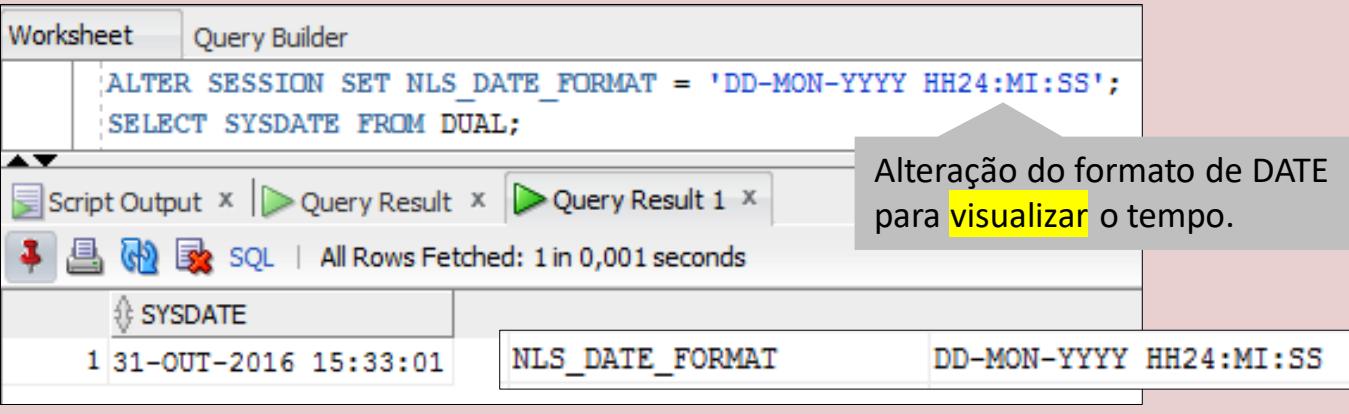
- Não permitidos.

Argumentos TIMESTAMP:

- Não permitidos.

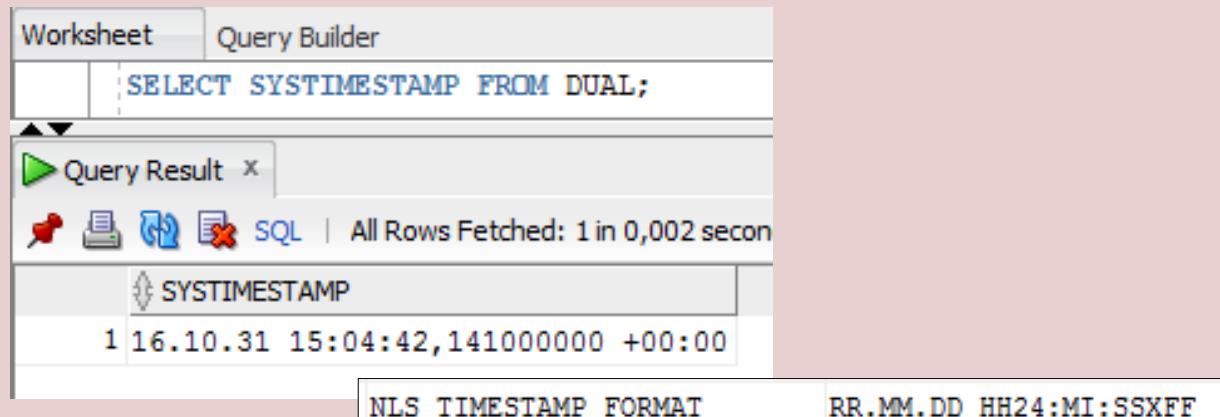
Categoria	Função	Funcionalidade
Data e Tempo Atuais	<u>SYSDATE</u>	Data e tempo atuais do sistema do servidor BD, tipo DATE.
	<u>SYSTIMESTAMP</u>	Semelhante a SYSDATE, exceto retorno tipo TIMESTAMP.
	<u>CURRENT_DATE</u>	Data e tempo atuais do fuso horário da sessão, tipo DATE.
	<u>CURRENT_TIMESTAMP</u>	Semelhante a CURRENT_DATE com retorno do tipo TIMESTAMP WITH TIME ZONE.
	<u>LOCALTIMESTAMP</u>	Semelhante a CURRENT_TIMESTAMP com retorno tipo TIMESTAMP.
Fuso Horário	<u>DBTIMEZONE</u>	Retorna o valor do fuso horário da BD.
	<u>TZ_OFFSET</u>	Retorna o <i>offset</i> do fuso horário especificado.
Data Especifica	<u>ADD_MONTHS</u>	Data = data +/- o número de meses indicado.
	<u>LAST_DAY</u>	Data do último dia do mês da data indicada.
	<u>NEXT_DAY</u>	Data do dia da semana seguinte à da data indicada.
	<u>ROUND</u>	Data mais próxima da data indicada para a unidade especificada.
	<u>TRUNC</u>	Data com a parte do tempo truncada para a unidade indicada.
Parte de Valor Data-Tempo	<u>EXTRACT</u>	Valor de um só campo data/tempo de uma data-tempo e intervalo.
Número de Meses	<u>MONTHS_BETWEEN</u>	Número de meses entre duas datas indicadas.
Conversão de Fuso Horário	<u>NEW_TIME</u>	Converte datas e tempos de fusos horários diferentes.

- Data e Tempo Atuais (1/5)

Função	Funcionalidade												
SYSDATE Função sem argumentos.	Retorna a data e o tempo atuais do sistema operativo que aloja o servidor da BD. Tipo de retorno = DATE. Formato definido no parâmetro NLS_DATE_FORMAT. Não pode ser usado numa restrição CHECK.  <pre data-bbox="451 540 988 655">Worksheet Query Builder : SELECT SYSDATE FROM DUAL;</pre> <p data-bbox="451 669 988 741">Query Result x</p> <p data-bbox="451 741 988 813">SQL All Rows Fetched: 1</p> <table border="1" data-bbox="451 813 1401 900"> <thead> <tr> <th data-bbox="451 813 700 871">SYSDATE</th> <th data-bbox="700 813 1180 871">NLS_DATE_FORMAT</th> <th data-bbox="1180 813 1401 871">RR.MM.DD</th> </tr> </thead> <tbody> <tr> <td data-bbox="451 871 700 900">1 16.10.31</td> <td data-bbox="700 871 1180 900"></td> <td data-bbox="1180 871 1401 900"></td> </tr> </tbody> </table>  <pre data-bbox="451 943 1564 1087">Worksheet Query Builder : ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MON-YYYY HH24:MI:SS'; : SELECT SYSDATE FROM DUAL;</pre> <p data-bbox="451 1101 1180 1173">Query Result x</p> <p data-bbox="451 1173 1180 1245">SQL All Rows Fetched: 1 in 0,001 seconds</p> <table border="1" data-bbox="451 1245 1800 1350"> <thead> <tr> <th data-bbox="451 1245 892 1303">SYSDATE</th> <th data-bbox="892 1245 1180 1303">NLS_DATE_FORMAT</th> <th data-bbox="1180 1245 1800 1303">DD-MON-YYYY HH24:MI:SS</th> </tr> </thead> <tbody> <tr> <td data-bbox="451 1303 892 1350">1 31-OCT-2016 15:33:01</td> <td data-bbox="892 1303 1180 1350"></td> <td data-bbox="1180 1303 1800 1350"></td> </tr> </tbody> </table>	SYSDATE	NLS_DATE_FORMAT	RR.MM.DD	1 16.10.31			SYSDATE	NLS_DATE_FORMAT	DD-MON-YYYY HH24:MI:SS	1 31-OCT-2016 15:33:01		
SYSDATE	NLS_DATE_FORMAT	RR.MM.DD											
1 16.10.31													
SYSDATE	NLS_DATE_FORMAT	DD-MON-YYYY HH24:MI:SS											
1 31-OCT-2016 15:33:01													

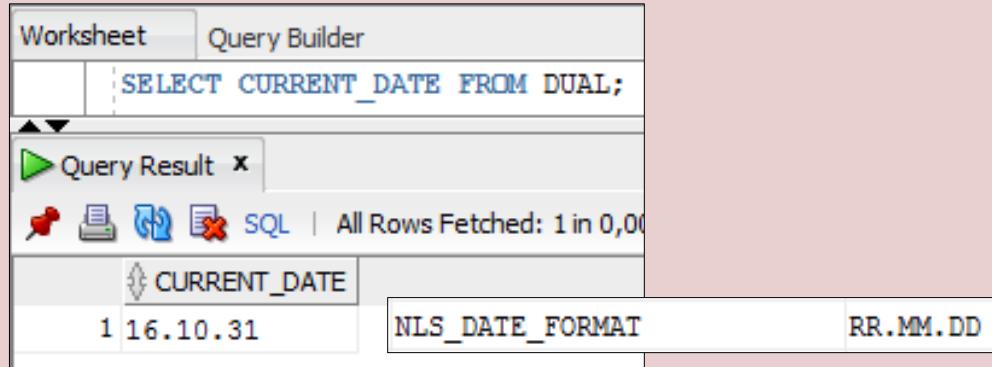
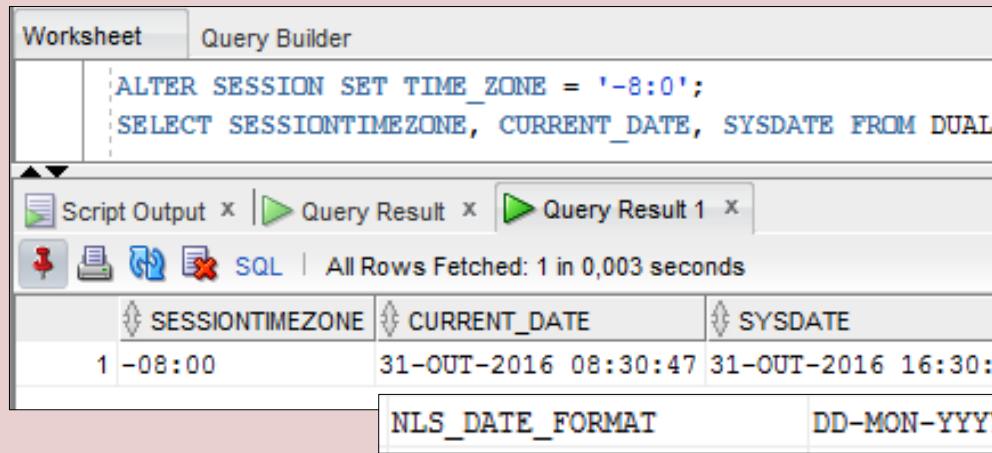
Funções Data-Tempo

■ Data e Tempo Atuais (2/5)

Função	Funcionalidade
SYSTIMESTAMP Função sem argumentos.	Retorna a data e o tempo atuais do sistema operativo que aloja o servidor da BD, incluindo: <ul style="list-style-type: none">▪ Frações do segundo.▪ Fuso horário. Tipo de retorno = TIMESTAMP WITH TIME ZONE.  <p>The screenshot shows the Oracle SQL Developer interface. In the Worksheet tab, the query <code>SELECT SYSTIMESTAMP FROM DUAL;</code> is entered. In the Query Result tab, the output is displayed in a table with one row. The column is labeled <code>SYSTIMESTAMP</code> and the value is <code>16.10.31 15:04:42,141000000 +00:00</code>. Below the result table, the system variable <code>NLS_TIMESTAMP_FORMAT</code> is set to <code>RR.MM.DD HH24:MI:SSXFF</code>.</p>

Funções Data-Tempo

■ Data e Tempo Atuais (3/5)

Função	Funcionalidade
CURRENT_DATE Função sem argumentos.	<p>Retorna a data e o tempo atuais do fuso horário da sessão. É sensível ao fuso horário da sessão.</p> <ul style="list-style-type: none"> ▪ Ao contrário da função SYSDATE. <p>Tipo de retorno = DATE.</p>  <pre>Worksheet Query Builder SELECT CURRENT_DATE FROM DUAL; Query Result x SQL All Rows Fetched: 1 in 0,00 CURRENT_DATE 1 16.10.31 NLS_DATE_FORMAT RR.MM.DD</pre>  <pre>Worksheet Query Builder ALTER SESSION SET TIME_ZONE = '-8:0'; SELECT SESSIONTIMEZONE, CURRENT_DATE, SYSDATE FROM DUAL; Script Output x Query Result x Query Result 1 x SQL All Rows Fetched: 1 in 0,003 seconds SESSIONTIMEZONE CURRENT_DATE SYSDATE -08:00 31-OCT-2016 08:30:47 31-OCT-2016 16:30:47 NLS_DATE_FORMAT DD-MON-YYYY HH24:MI:SS</pre>

- Data e Tempo Atuais (4/5)

Função	Funcionalidade
CURRENT_TIMESTAMP Função sem argumentos.	Retorna a data e o tempo atuais do fuso horário da sessão. Tipo de retorno = TIMESTAMP WITH TIME ZONE. Precisão: <ul style="list-style-type: none"> ▪ 6 dígitos decimais das frações do segundo.
CURRENT_TIMESTAMP(precision)	Semelhante a CURRENT_TIMESTAMP. <ul style="list-style-type: none"> ▪ Diferença:<ul style="list-style-type: none"> ▪ Permite definir a precisão das frações do segundo. ▪ [0, 9] dígitos decimais.

The screenshot shows the Oracle SQL Developer interface. The top navigation bar has tabs for 'Worksheet' and 'Query Builder', with 'Worksheet' selected. Below the tabs, a query is entered in the main pane:

```
SELECT CURRENT_TIMESTAMP, CURRENT_TIMESTAMP(1) FROM DUAL;
```

The results are displayed in the 'Script Output' tab, which shows the output of the query:

CURRENT_TIMESTAMP	CURRENT_TIMESTAMP(1)
16.10.31 17:38:37,294000000 EUROPE/LONDON	16.10.31 17:38:37,300000000 EUROPE/LONDON

Below the results, status information is shown: 'All Rows Fetched: 1 in 0,001 seconds'.

- Data e Tempo Atuais (5/5)

Função	Funcionalidade
LOCALTIMESTAMP Função sem argumentos.	Semelhante a CURRENT_TIMESTAMP. <ul style="list-style-type: none"> ▪ Diferença: <ul style="list-style-type: none"> ▪ Não retorna o fuso horário. <p>Tipo de retorno = TIMESTAMP.</p>
LOCALTIMESTAMP(precision)	Semelhante a LOCALTIMESTAMP. <ul style="list-style-type: none"> ▪ Diferença: <ul style="list-style-type: none"> ▪ Permite definir a precisão das frações do segundo <ul style="list-style-type: none"> ▪ [0, 9] dígitos decimais.

The screenshot shows the Oracle SQL Developer interface. In the 'Worksheet' tab, a query is run:

```
SELECT CURRENT_TIMESTAMP, LOCALTIMESTAMP, LOCALTIMESTAMP(1) FROM DUAL;
```

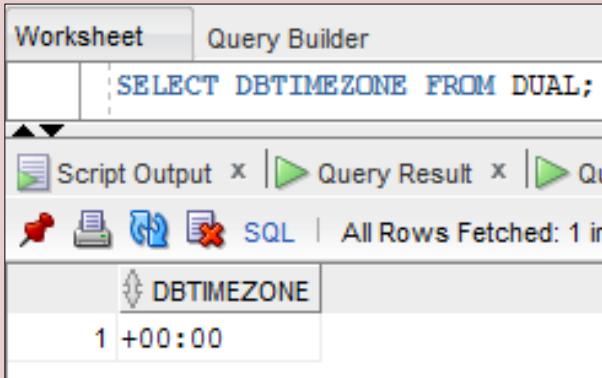
The results are displayed in the 'Query Result' tab:

CURRENT_TIMESTAMP	LOCALTIMESTAMP	LOCALTIMESTAMP(1)
16.10.31 17:53:20,033000000 EUROPE/LONDON	16.10.31 17:53:20,033000000	16.10.31 17:53:20,000000000

Below the tabs, status icons indicate the session state: Script Output, Query Result, Query Result 1, Query Result 2, and SQL. The message 'All Rows Fetched: 1 in 0,003 seconds' is shown.

▪ Fuso Horário (1/3)

Função	Funcionalidade
<p>DBTIMEZONE</p> <p>Função sem argumentos.</p>	<p>Retorna o valor do fuso horário da BD.</p> <p>É um desvio do fuso horário:</p> <ul style="list-style-type: none"> ▪ Formato: <ul style="list-style-type: none"> ▪ Alternativas: <ul style="list-style-type: none"> ▪ [+ -] TZH:TZM ▪ Nome da região do fuso horário. ▪ Depende: <ul style="list-style-type: none"> ▪ Forma como foi especificado o fuso horário da BD, através do mais recente comando CREATE DATABASE ou ALTER DATABASE.

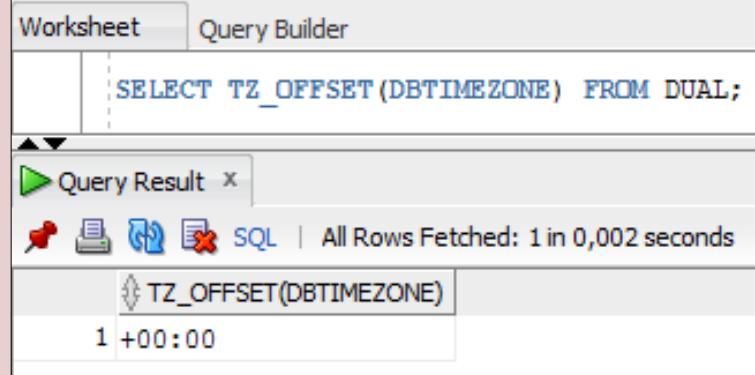
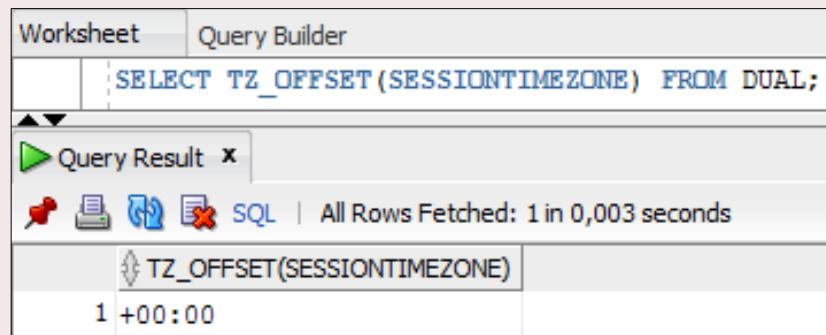


The screenshot shows the Oracle SQL Developer interface. In the top tab bar, 'Worksheet' is selected. Below it, the 'Query Builder' tab is visible. The main workspace contains a single line of SQL code: 'SELECT DBTIMEZONE FROM DUAL;'. At the bottom of the interface, there are several tabs: 'Script Output', 'Query Result', and 'SQL'. The 'Query Result' tab is active, displaying the output of the query. The result table has one row with one column, labeled 'DBTIMEZONE'. The value in the cell is '+00:00'. There are also icons for refresh, print, and other database operations.

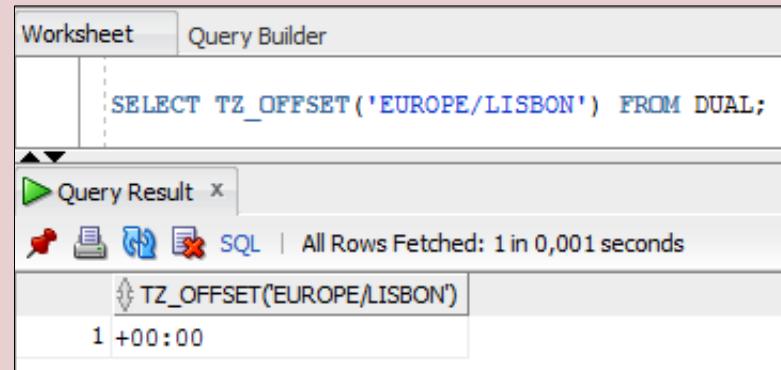
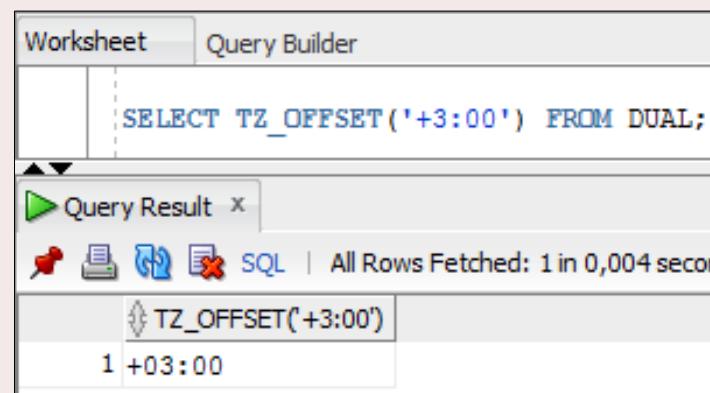
```
Worksheet Query Builder
SELECT DBTIMEZONE FROM DUAL;
Script Output x | Query Result x | SQL | All Rows Fetched: 1 in
DBTIMEZONE
1 +00:00
```

Funções Data-Tempo

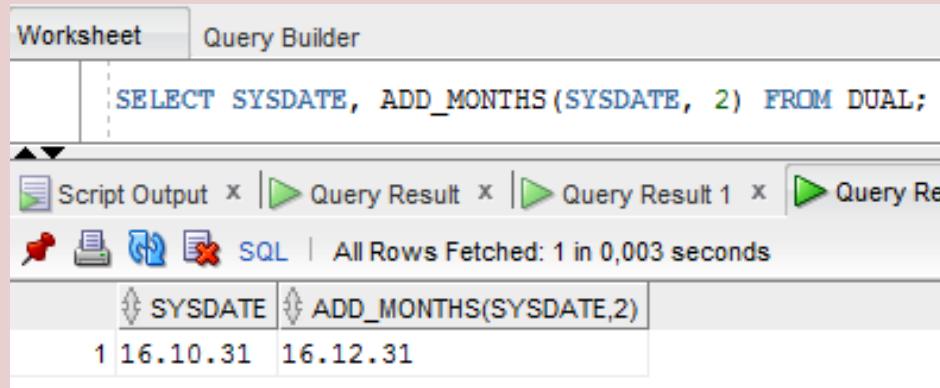
▪ Fuso Horário (2/3)

Função	Funcionalidade
TZ_OFFSET(DBTIMEZONE)	<p>Retorna o <i>offset</i> do fuso horário da BD.</p>  <p>The screenshot shows the Oracle SQL Developer interface. In the Worksheet tab, the query <code>SELECT TZ_OFFSET(DBTIMEZONE) FROM DUAL;</code> is entered. In the Query Result tab, the output is displayed in a table with one row labeled '1' and the value '+00:00'. The interface includes standard SQL navigation icons (New, Open, Save, Print, Refresh, Stop, Run).</p>
TZ_OFFSET(SESSIONTIMEZONE)	<p>Retorna o <i>offset</i> do fuso horário da sessão da BD.</p>  <p>The screenshot shows the Oracle SQL Developer interface. In the Worksheet tab, the query <code>SELECT TZ_OFFSET(SESSIONTIMEZONE) FROM DUAL;</code> is entered. In the Query Result tab, the output is displayed in a table with one row labeled '1' and the value '+00:00'. The interface includes standard SQL navigation icons (New, Open, Save, Print, Refresh, Stop, Run).</p>

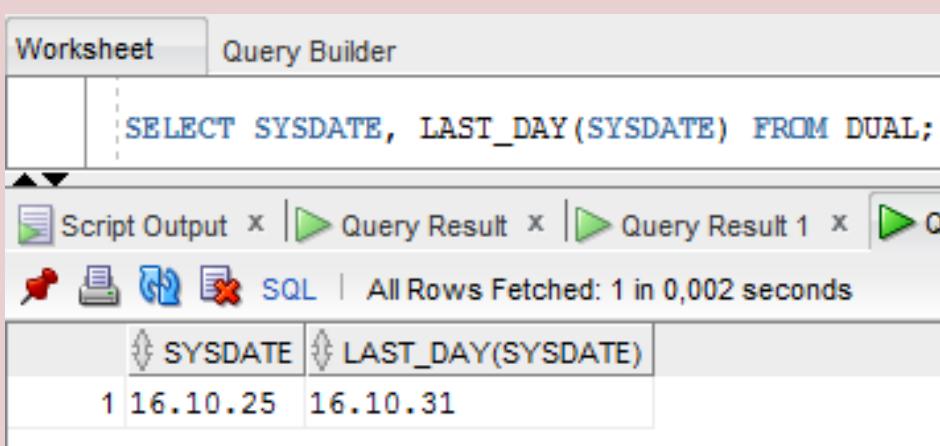
- Fuso Horário (3/3)

Função	Funcionalidade
TZ_OFFSET('time_zone_name') <pre>-- Lista de nomes de fusos horários SELECT * FROM V\$TIMEZONE_NAMES ;</pre>	<p>Retorna o <i>offset</i> do fuso horário de nome time_zone_name.</p>  <pre>Worksheet Query Builder SELECT TZ_OFFSET('EUROPE/LISBON') FROM DUAL; Query Result SQL All Rows Fetched: 1 in 0,001 seconds TZ_OFFSET('EUROPE/LISBON') 1 +00:00</pre>
TZ_OFFSET('+hh:mi') TZ_OFFSET('-hh:mi')	<p>Retorna o próprio <i>offset</i> de fuso horário especificado.</p>  <pre>Worksheet Query Builder SELECT TZ_OFFSET('+3:00') FROM DUAL; Query Result SQL All Rows Fetched: 1 in 0,004 seconds TZ_OFFSET('+3:00') 1 +03:00</pre>

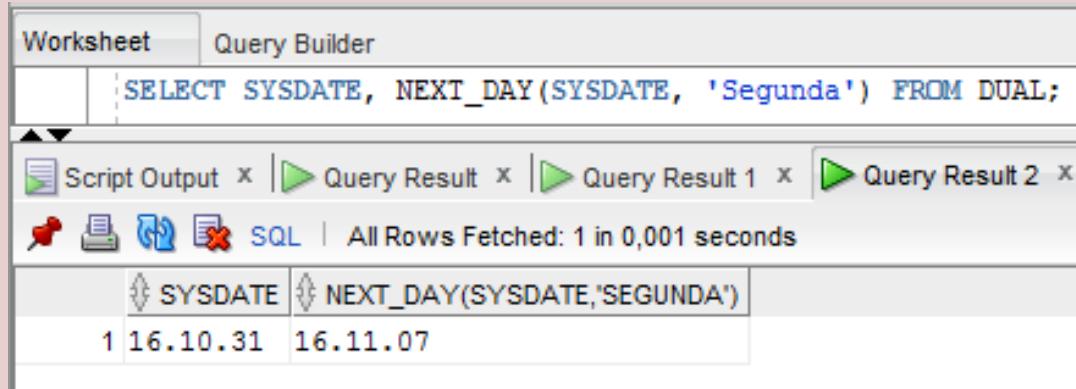
- Data Específica (1/21)

Função	Funcionalidade
ADD_MONTHS(date, integer)	<p>Retorna a data <u>date</u> mais o número <u>integer</u> de meses.</p> <p>Tipo do retorno = DATE.</p> <p><u>date</u> = último dia de um mês:</p> <ul style="list-style-type: none"> ▪ Retornado sempre o último dia do mês resultante.  <pre data-bbox="768 504 1708 892">Worksheet Query Builder SELECT SYSDATE, ADD_MONTHS(SYSDATE, 2) FROM DUAL; Script Output Query Result Query Result 1 Query Res SYSDATE ADD_MONTHS(SYSDATE,2) 1 16.10.31 16.12.31 Worksheet Query Builder SELECT SYSDATE, ADD_MONTHS(SYSDATE, 1) FROM DUAL; Script Output Query Result Query Result 1 Query Res SYSDATE ADD_MONTHS(SYSDATE,1) 1 16.10.31 16.11.30</pre>

- Data Específica (2/21)

Função	Funcionalidade						
LAST_DAY(date)	<p>Retorna a data do último dia do mês que contém <u>date</u>.</p> <p>Tipo do retorno = DATE.</p>  <p>The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab is active, displaying the SQL query:</p> <pre>SELECT SYSDATE, LAST_DAY(SYSDATE) FROM DUAL;</pre> <p>Below the worksheet, the 'Script Output' tab is selected, showing the results:</p> <table border="1"><thead><tr><th></th><th>SYSDATE</th><th>LAST_DAY(SYSDATE)</th></tr></thead><tbody><tr><td>1</td><td>16.10.25</td><td>16.10.31</td></tr></tbody></table> <p>The results show that the current system date is 16.10.25, and the last day of the month is 16.10.31.</p>		SYSDATE	LAST_DAY(SYSDATE)	1	16.10.25	16.10.31
	SYSDATE	LAST_DAY(SYSDATE)					
1	16.10.25	16.10.31					

- Data Específica (3/21)

Função	Funcionalidade				
NEXT_DAY(date, char)	<p>Retorna a data do primeiro dia da semana designado <u>char</u>, posterior uma semana a <u>date</u>.</p> <p>Tipo do retorno = DATE.</p>  <p>The screenshot shows the Oracle SQL Developer interface. In the top tab bar, 'Worksheet' is selected. Below it, a query window contains the SQL statement: 'SELECT SYSDATE, NEXT_DAY(SYSDATE, 'Segunda') FROM DUAL;'. Below the query window, there are four tabs: 'Script Output', 'Query Result', 'Query Result 1', and 'Query Result 2'. The 'Query Result' tab is active, displaying the output of the query:</p> <table border="1"><thead><tr><th>SYSDATE</th><th>NEXT_DAY(SYSDATE,'SEGUNDA')</th></tr></thead><tbody><tr><td>16.10.31</td><td>16.11.07</td></tr></tbody></table> <p>Below the table, a status message says 'All Rows Fetched: 1 in 0,001 seconds'.</p>	SYSDATE	NEXT_DAY(SYSDATE,'SEGUNDA')	16.10.31	16.11.07
SYSDATE	NEXT_DAY(SYSDATE,'SEGUNDA')				
16.10.31	16.11.07				

■ Data Específica (5/21)

- Modelos de Formato (1/11)

Modelos de Formato	Unidade de Arredondamento/Truncagem
CC	Junta 01 aos 2 dígitos iniciais do ano de 4 dígitos.
SCC	<ul style="list-style-type: none">■ Arredondamento ... dos 2 dígitos iniciais do ano:<ul style="list-style-type: none">■ Para cima:<ul style="list-style-type: none">■ A partir dos 2 últimos dígitos ≥ 51.■ Exemplos:<ul style="list-style-type: none">■ 2016-10-31 arredondada para baixo: 2001-01-01.■ 1951-08-11 arredondada para cima: 2001-01-01.■ 1950-08-11 arredondada para baixo: 1901-01-01.■ Truncagem ... dos 2 dígitos iniciais do ano:<ul style="list-style-type: none">■ Exemplo:<ul style="list-style-type: none">■ 2016-10-31 truncada para 2001-01-01 -- ... do ano da data.■ 1951-08-01 truncada para 1901-01-01

■ Data Específica (6/21)

- Modelos de Formato (2/11)

Modelos de Formato	Unidade de Arredondamento/Truncagem
SYYY	Ano.
YYYY	<ul style="list-style-type: none"> ■ Arredondamento ... do ano:
YEAR	<ul style="list-style-type: none"> ■ Para cima:
SYEAR	<ul style="list-style-type: none"> ■ A partir de 1 julho.
YYY	<ul style="list-style-type: none"> ■ Exemplos:
YY	<ul style="list-style-type: none"> ■ 2016-10-31 arredondada para cima: 2017-01-01
Y	<ul style="list-style-type: none"> ■ 2016-06-30 arredondada para baixo: 2016-01-01
	<ul style="list-style-type: none"> ■ Truncagem ... do ano: <p>■ Exemplo:</p> <ul style="list-style-type: none"> ■ 2016-10-31 truncada para 2016-01-01 -- ... do ano da data. ■ 2016-06-30 truncada para 2016-01-01

- **Data Específica (7/21)**

- Modelos de Formato (3/11)

Modelos de Formato	Unidade de Arredondamento/Truncagem
IYYY	Primeira segunda-feira (primeiro dia de semana) do ano.
IYY	<ul style="list-style-type: none"> ▪ Definido no <i>standard ISO 8610</i>.
IY	<ul style="list-style-type: none"> ▪ Arredondamento ... para a primeira segunda-feira do ano.
I	<ul style="list-style-type: none"> ▪ Para cima: ▪ A partir de 1 de julho. ▪ Exemplo: ▪ 2016-10-31 arredondada para cima: 2017-01-02 ▪ 2016-06-31 arredondada para baixo: 2016-01-04 ▪ Truncagem ... para a primeira segunda-feira do ano: ▪ Exemplo: ▪ 2016-10-31 truncada para 2016-01-04 -- ... do ano da data. ▪ 2016-06-30 truncada para 2016-01-04

■ Data Específica (8/21)

- Modelos de Formato (4/11)

Modelos De Formato	Unidade de Arredondamento/Truncagem
Q	<p><i>Quarter</i> (Trimestres: T1, T2, T3 e T4).</p> <ul style="list-style-type: none">■ Arredondamento ... do trimestre:<ul style="list-style-type: none">■ Para cima:<ul style="list-style-type: none">■ A partir do 16º dia do 2º mês do trimestre.■ Exemplos:<ul style="list-style-type: none">■ 2016-08-15 arredondada para baixo: 2016-07-01 -- início T3■ 2016-08-16 arredondada para cima: 2016-10-01 -- início T4■ Truncagem ... do trimestre:<ul style="list-style-type: none">■ Exemplo:<ul style="list-style-type: none">■ 2016-08-15 truncada para 2016-07-01 -- início do semestre■ 2016-08-16 truncada para 2016-07-01

■ Data Específica (9/21)

- Modelos de Formato (5/11)

Modelos de Formato	Unidade de Arredondamento/Truncagem
MONTH	Mês.
MON	<ul style="list-style-type: none">■ Arredondamento ... do mês.
MM	<ul style="list-style-type: none">■ Para cima:
RM	<ul style="list-style-type: none">■ A partir do 16º dia do mês.■ Exemplos:<ul style="list-style-type: none">■ 2016-10-15 arredondada para baixo: 2016-10-01■ 2016-10-16 arredondada para cima: 2016-11-01■ Truncagem ... do mês:<ul style="list-style-type: none">■ Exemplo:<ul style="list-style-type: none">■ 2016-10-15 truncada para 2016-10-01 -- início do mês da data.■ 2016-10-16 truncada para 2016-10-01

■ Data Específica (10/21)

- Modelos de Formato (6/11)

Modelos de Formato	Unidade de Arredondamento/Truncagem
WW	<p>Dia da semana do ano (1-53):</p> <ul style="list-style-type: none"> ■ 1^a semana: [1, 7] de janeiro. ■ Arredondamento ... do dia da semana do ano. <ul style="list-style-type: none"> ■ Para cima: <ul style="list-style-type: none"> ■ A partir do 5º dia da semana. ■ Exemplos: <ul style="list-style-type: none"> ■ 2016-01-04 arredondada para 2016-01-01 -- início S1 ■ 2016-01-05 arredondada para 2016-01-08 -- início S2 ■ Truncagem ... do dia da semana do ano: <ul style="list-style-type: none"> ■ Exemplo: <ul style="list-style-type: none"> ■ 2016-01-04 truncada para 2016-01-01 -- início da semana da data. ■ 2016-01-05 truncada para 2016-01-01

■ Data Específica (11/21)

- Modelos de Formato (7/11)

Modelos de Formato	Unidade de Arredondamento/Truncagem
IW	<p>Segunda-feira.</p> <ul style="list-style-type: none">■ Definido no <i>standard ISO 8610</i>.■ Arredondamento ... da segunda-feira.<ul style="list-style-type: none">■ Para cima:<ul style="list-style-type: none">■ A partir de sexta-feira.■ Exemplos:<ul style="list-style-type: none">■ 2016-10-27 arredondada para baixo: 2016-10-24■ 2016-10-28 arredondada para cima: 2016-10-31■ Truncagem ... do dia da semana do ano:<ul style="list-style-type: none">■ Exemplo:<ul style="list-style-type: none">■ 2016-10-27 truncada para 2016-10-24 -- 2ªF da semana da data.■ 2016-10-28 truncada para 2016-10-24

■ Data Específica (12/21)

- Modelos de Formato (8/11)

Modelos de Formato	Unidade de Arredondamento ou Truncagem
W	<p>Dia da semana do mês (1-5):</p> <ul style="list-style-type: none">■ 1^a semana: [1, 7]■ Arredondamento ... do dia da semana do mês.<ul style="list-style-type: none">■ Para cima:<ul style="list-style-type: none">■ A partir do 5º dia da semana.■ Exemplos:<ul style="list-style-type: none">■ 2016-10-04 arredondada para baixo: 2016-10-01 -- início S1■ 2016-10-05 arredondada para cima: 2016-10-08 -- início S2■ Truncagem ... do dia da semana do mês:<ul style="list-style-type: none">■ Exemplo:<ul style="list-style-type: none">■ 2016-10-04 truncada para 2016-10-01 -- início da semana da data.■ 2016-10-05 truncada para 2016-10-01

■ Data Específica (13/21)

- Modelos de Formato (9/11)

Modelos de Formato	Unidade de Arredondamento ou Truncagem
DDD, DD, J	Dia.
DAY	Dia inicial da semana.
DY	<ul style="list-style-type: none"> ■ Definido implicitamente no parâmetro NLS_TERRITORY.
D	<ul style="list-style-type: none"> ■ Arredondamento ... dia inicial da semana. <ul style="list-style-type: none"> ■ Para cima: ■ A partir do 5º dia da semana. ■ Exemplo: <ul style="list-style-type: none"> ■ 2016-10-31 arredondada para baixo: 2016-10-30 ■ 2016-11-03 arredondada para cima: 2016-11-06 ■ Truncagem ... do dia inicial da semana: <ul style="list-style-type: none"> ■ Exemplo: <ul style="list-style-type: none"> ■ 2016-10-31 truncada para 2016-10-30 -- início da semana da data. ■ 2016-11-03 truncada para 2016-10-30

- **Data Específica (14/21)**

- Modelos de Formato (10/11)

Modelos de Formato	Unidade de Arredondamento ou Truncagem
HH HH12 HH24	<p>Hora.</p> <ul style="list-style-type: none"> ▪ HH = HH12 = [0, 12] ▪ HH24 = [0,23] ▪ Arredondamento ... da hora. <ul style="list-style-type: none"> ▪ Para cima: ▪ A partir do minuto 30. ▪ Exemplos: <ul style="list-style-type: none"> ▪ 2016-11-01 15:29 arredondada para baixo: 2016-11-01 15:00 ▪ 2016-11-01 15:31 arredondada para cima: 2016-11-01 16:00 ▪ Truncagem ... da hora: <ul style="list-style-type: none"> ▪ Exemplo: <ul style="list-style-type: none"> ▪ 2016-11-01 15:29 truncada para 2016-11-01 15:00 ▪ 2016-11-01 15:31 truncada para 2016-11-01 15:00

■ Data Específica (15/21)

■ Modelos de Formato (11/11)

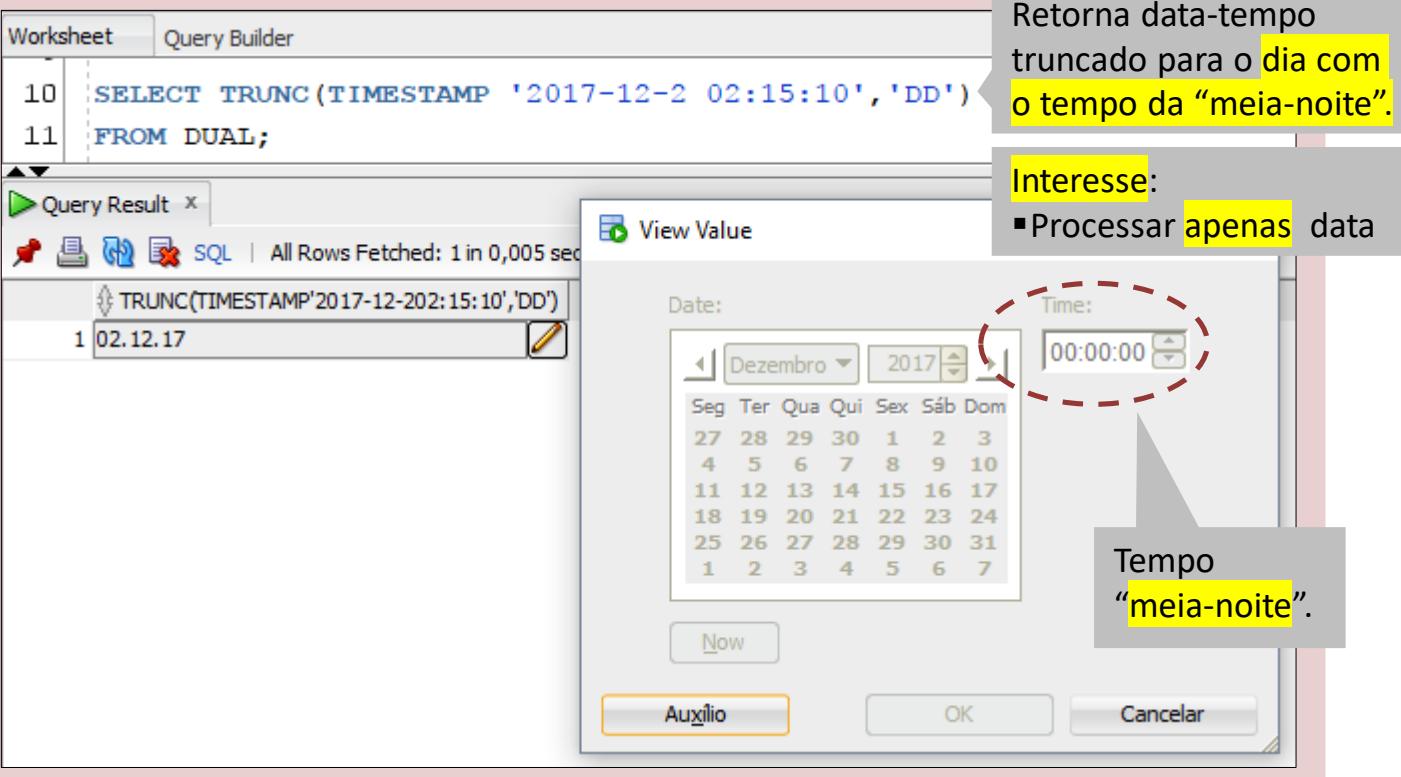
Modelos de Formato	Unidade de Arredondamento ou Truncagem
MI	<p>Minuto.</p> <ul style="list-style-type: none">▪ Arredondamento ... do minuto.<ul style="list-style-type: none">▪ Para cima:<ul style="list-style-type: none">▪ A partir do segundo 30.▪ Exemplos:<ul style="list-style-type: none">▪ 2016-11-01 15:00:29 arredondada para baixo: 2016-11-01 15:00:00▪ 2016-11-01 15:00:31 arredondada para cima: 2016-11-01 15:01:00▪ Truncagem ... do minuto:<ul style="list-style-type: none">▪ Exemplo:<ul style="list-style-type: none">▪ 2016-11-01 15:00:29 arredondada para cima: 2016-11-01 15:00:00.▪ 2016-11-01 15:00:31 arredondada para cima: 2016-11-01 15:00:00

- Data Específica (16/21)

Função	Funcionalidade
ROUND(date, fmt)	<p><u>date</u> arredondada para a unidade definida pelo <u>modelo de formato fmt</u>.</p> <p>Tipo do retorno = DATE.</p> <p>Insensível ao parâmetro de sessão NLS_CALENDAR.</p> <ul style="list-style-type: none"> ■ Funcionamento baseado nas regras do calendário Gregoriano. <p>Exemplo:</p> <ul style="list-style-type: none"> ■ <u>fmt</u> = '<u>YEAR</u>' -- arredondamento baseado no ano. ■ 2016-10-31 arredondada para cima: 2017-01-01 ■ 2016-06-30 arredondada para baixo: 2016-01-01
ROUND(date)	<p>Retorna <u>date</u>.</p> <p><u>Modelo de formato</u> por omissão = 'DD'</p> <p>Tipo do retorno é DATE.</p>

Funções Data-Tempo

- Data Específica (17/21)**

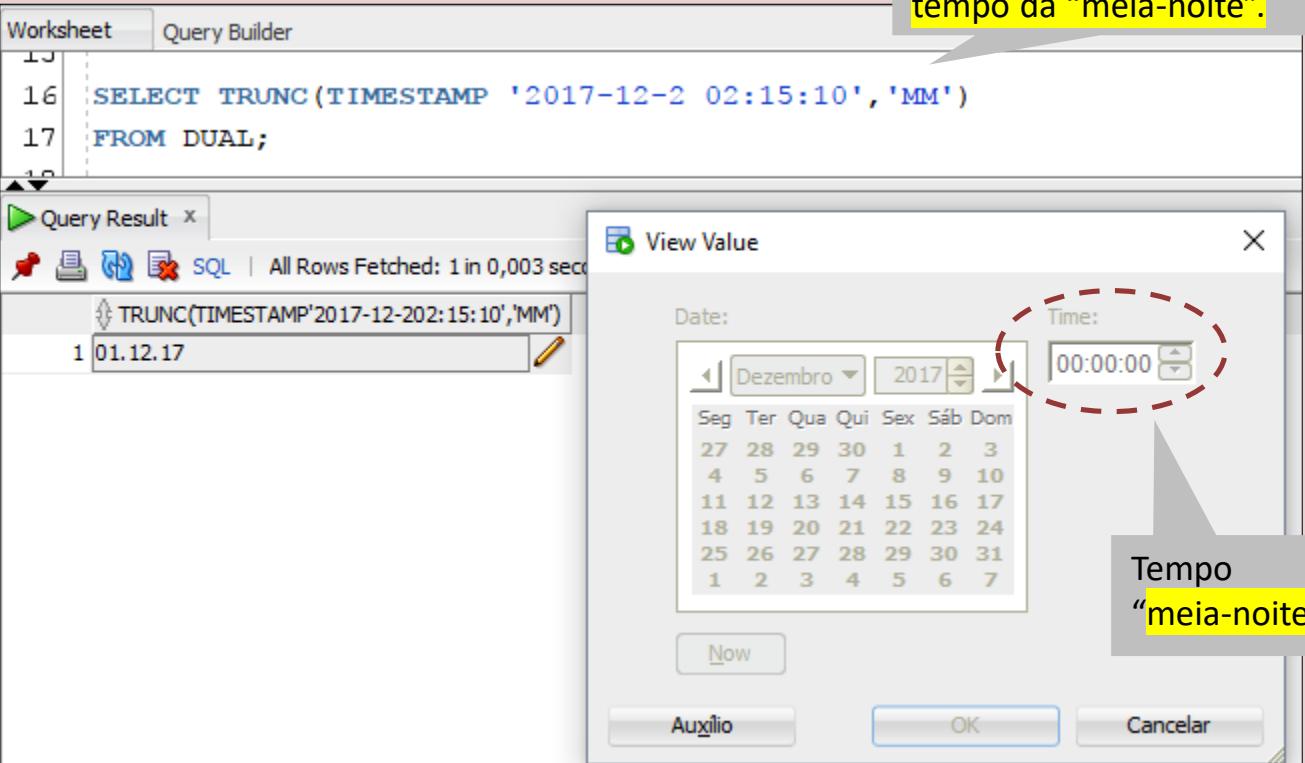
Função	Funcionalidade		
TRUNC(date, fmt)	<p>Retorna <u>date</u> truncada para a unidade especificada no <u>modelo de formato fmt</u>.</p> <p>Tipo do retorno = DATE.</p> <p>Insensível ao parâmetro de sessão NLS_CALENDAR.</p> <ul style="list-style-type: none"> ■ Funcionamento baseado nas regras do calendário Gregoriano.  <p>The screenshot shows the Oracle SQL Developer interface. In the Worksheet tab, a query is run:</p> <pre> 10 SELECT TRUNC(TIMESTAMP '2017-12-2 02:15:10', 'DD') 11 FROM DUAL; </pre> <p>The result is displayed in the Query Result window:</p> <table border="1"> <thead> <tr> <th>TRUNC(TIMESTAMP'2017-12-20 02:15:10','DD')</th> </tr> </thead> <tbody> <tr> <td>1 02.12.17</td> </tr> </tbody> </table> <p>A modal dialog titled "View Value" is open, showing a calendar and time picker. The calendar displays December 2017. The time picker shows "00:00:00". A dashed red circle highlights the time picker, and a callout box points to it with the text: "Retorna data-tempo truncado para o dia com o tempo da ‘meia-noite’". Another callout box points to the time picker with the text: "Interesse: Processar apenas data". A third callout box points to the time picker with the text: "Tempo ‘meia-noite’".</p>	TRUNC(TIMESTAMP'2017-12-20 02:15:10','DD')	1 02.12.17
TRUNC(TIMESTAMP'2017-12-20 02:15:10','DD')			
1 02.12.17			

Funções Data-Tempo

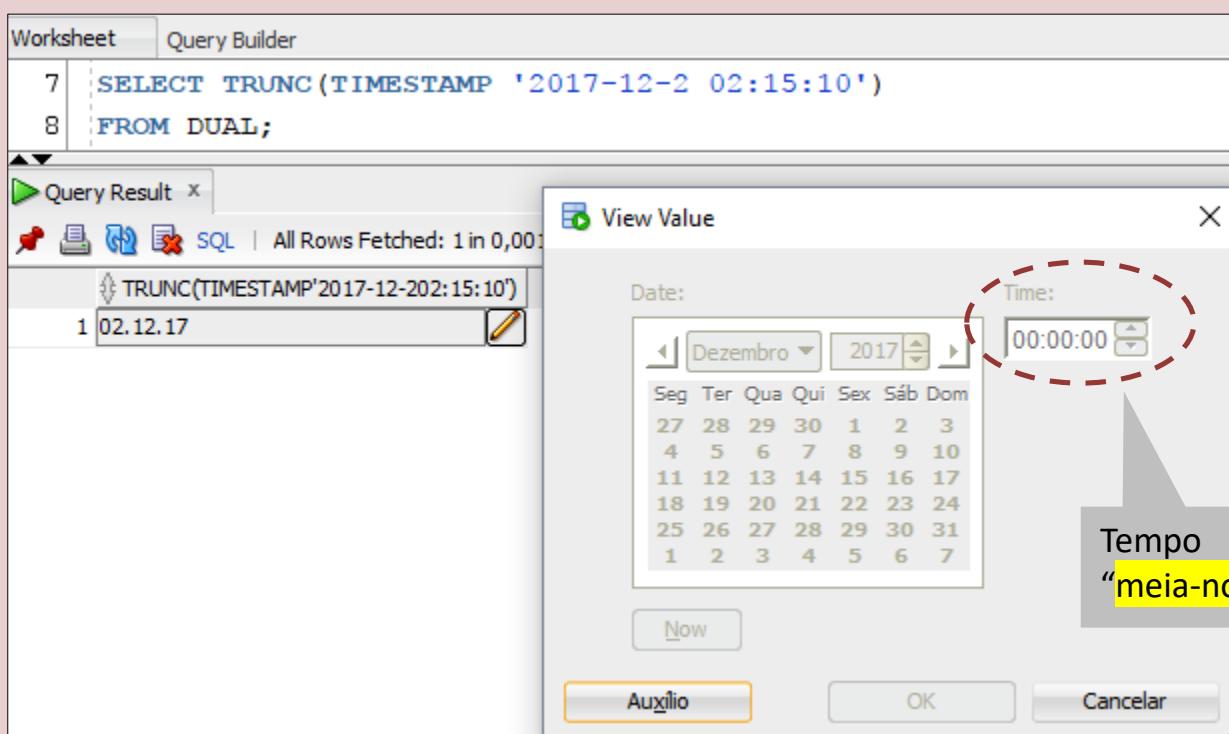
- Data Específica (18/21)

Função	Funcionalidade
TRUNC(date, fmt)	<p>Retorna data-tempo truncado para o início do ano e com o tempo da “meia-noite”.</p> <pre> Worksheet Query Builder 13 SELECT TRUNC(TIMESTAMP '2017-12-2 02:15:10', 'YYYY') 14 FROM DUAL; Query Result SQL All Rows Fetched: 1 in 0,001 seconds 1 01.01.17 View Value Date: Janeiro 2017 Time: 00:00:00 Buttons: Now, Auxílio, OK, Cancelar </pre>

- Data Específica (19/21)

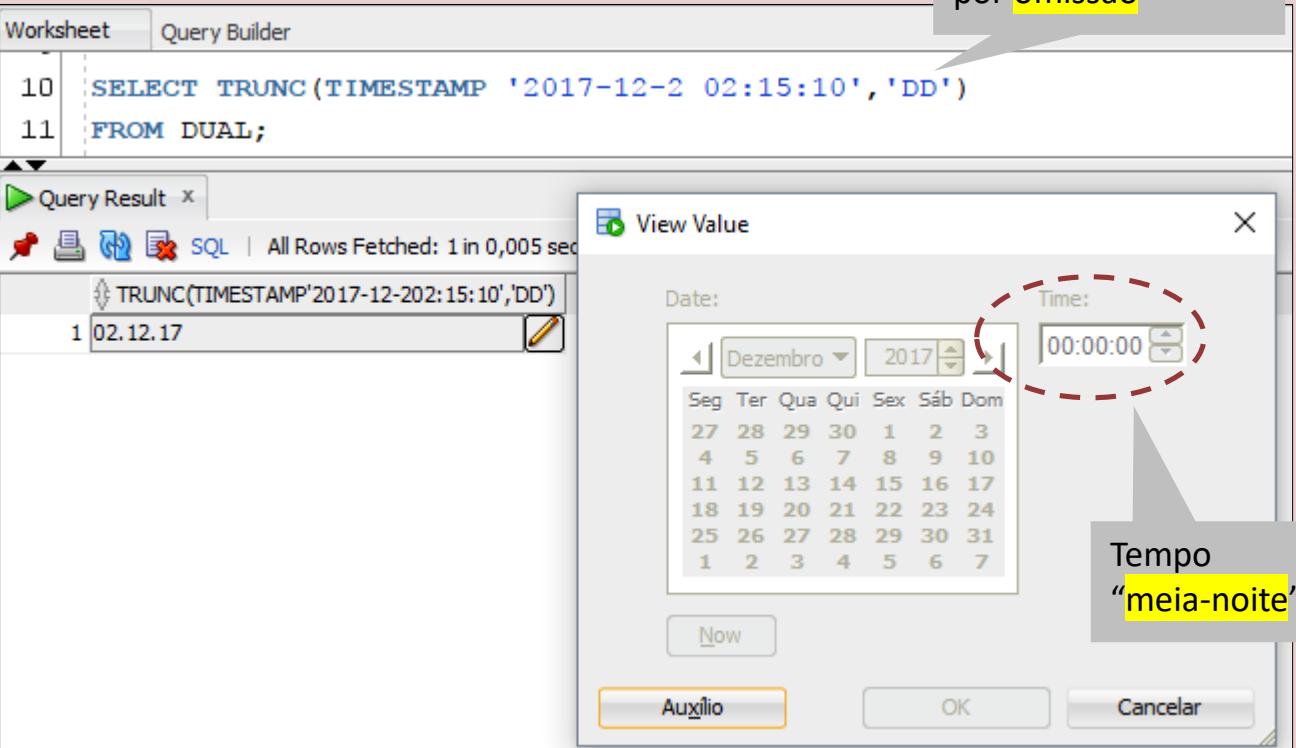
Função	Funcionalidade		
TRUNC(date, fmt)	<p>Retorna data-tempo truncado para o início do mês e com o tempo da “meia-noite”.</p>  <pre> Worksheet Query Builder 16 SELECT TRUNC(TIMESTAMP '2017-12-2 02:15:10', 'MM') 17 FROM DUAL; </pre> <p>Query Result</p> <table border="1"> <thead> <tr> <th>TRUNC(TIMESTAMP'2017-12-202:15:10','MM')</th> </tr> </thead> <tbody> <tr> <td>01.12.17</td> </tr> </tbody> </table> <p>View Value</p> <p>Date: <input type="text" value="01.12.17"/></p> <p>Time: <input type="text" value="00:00:00"/></p> <p>Buttons: Now, Auxílio, OK, Cancelar</p>	TRUNC(TIMESTAMP'2017-12-202:15:10','MM')	01.12.17
TRUNC(TIMESTAMP'2017-12-202:15:10','MM')			
01.12.17			

- Data Específica (20/21)

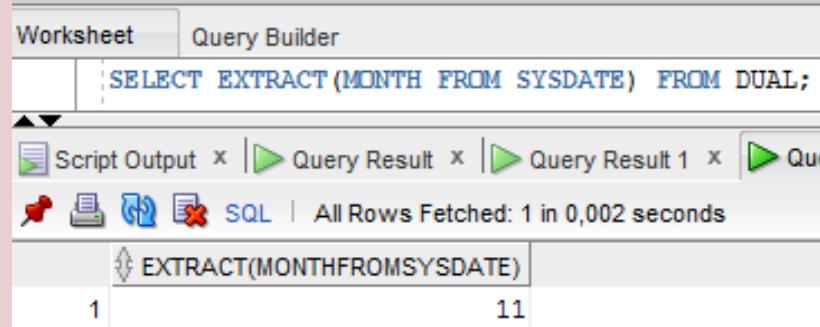
Função	Funcionalidade		
TRUNC(date)	<p>Retorna <u>date</u> truncada para o dia com o tempo da “meia-noite”.</p> <p><u>Modelo de formato</u> por omissão = 'DD'</p> <p>Tipo do retorno = DATE.</p>  <pre> Worksheet Query Builder 7 SELECT TRUNC(TIMESTAMP '2017-12-2 02:15:10') 8 FROM DUAL; </pre> <p>Query Result</p> <table border="1"> <thead> <tr> <th>TRUNC(TIMESTAMP'2017-12-202:15:10')</th> </tr> </thead> <tbody> <tr> <td>1 02.12.17</td> </tr> </tbody> </table> <p>View Value</p> <p>Date: Dezembro 2017</p> <p>Time: 00:00:00</p> <p>Tempo “meia-noite”.</p>	TRUNC(TIMESTAMP'2017-12-202:15:10')	1 02.12.17
TRUNC(TIMESTAMP'2017-12-202:15:10')			
1 02.12.17			

Funções Data-Tempo

- Data Específica (21/21)

Função	Funcionalidade		
TRUNC(date)	<p>Retorna <u>date</u> truncada para o dia com o tempo da “meia-noite”.</p> <p><u>Modelo de formato</u> por omissão = 'DD'</p> <p>Tipo do retorno = DATE.</p>  <p>The screenshot shows the Oracle SQL Developer interface. In the Worksheet tab, the following SQL code is entered:</p> <pre>10 SELECT TRUNC(TIMESTAMP '2017-12-2 02:15:10', 'DD') 11 FROM DUAL;</pre> <p>The Query Result shows the output of the query:</p> <table border="1"> <thead> <tr> <th>TRUNC(TIMESTAMP'2017-12-202:15:10','DD')</th> </tr> </thead> <tbody> <tr> <td>1 02.12.17</td> </tr> </tbody> </table> <p>A modal dialog titled "View Value" is open, showing the date and time selection interface. The date is set to December 2, 2017, and the time is set to 00:00:00. A callout bubble points to the time selection area with the text "Modelo de formato por omissão". Another callout bubble points to the time value with the text "Tempo ‘meia-noite’".</p>	TRUNC(TIMESTAMP'2017-12-202:15:10','DD')	1 02.12.17
TRUNC(TIMESTAMP'2017-12-202:15:10','DD')			
1 02.12.17			

Campo de Valor Data-Tempo (1/2)

Função	Funcionalidade										
<p>EXTRACT(field FROM expr)</p> <p>field:</p> <ul style="list-style-type: none"> 1. YEAR, MONTH 2. DAY 3. HOUR, MINUTE, SECOND 4. TIMEZONE_HOUR, TIMEZONE_MINUTE, TIMEZONE_REGION, TIMEZONE_ABBR 	<p>Extrai e retorna o valor do field especificado, contido na expressão expr (expressão data-tempo ou intervalo data-tempo).</p>  <table border="1" data-bbox="668 688 1877 1350"> <thead> <tr> <th>Fields</th><th>Tipo de <u>expr</u></th></tr> </thead> <tbody> <tr> <td>1</td><td>DATE, Qualquer tipo TIMESTAMP, INTERVAL YEAR TO MONTH</td></tr> <tr> <td>2</td><td>DATE, Qualquer tipo TIMESTAMP, INTERVAL DAY TO SECOND</td></tr> <tr> <td>3</td><td>Qualquer tipo TIMESTAMP, INTERVAL DAY TO SECOND <ul style="list-style-type: none"> ▪ Tipo DATE é inválido ... porque, neste caso, a expr é assumida do tipo DATE standard ANSI, o qual não tem campos de tempo. </td></tr> <tr> <td>4</td><td>TIMESTAMP WITH TIME ZONE TIMESTAMP WITH LOCAL TIME ZONE</td></tr> </tbody> </table>	Fields	Tipo de <u>expr</u>	1	DATE, Qualquer tipo TIMESTAMP, INTERVAL YEAR TO MONTH	2	DATE, Qualquer tipo TIMESTAMP, INTERVAL DAY TO SECOND	3	Qualquer tipo TIMESTAMP, INTERVAL DAY TO SECOND <ul style="list-style-type: none"> ▪ Tipo DATE é inválido ... porque, neste caso, a expr é assumida do tipo DATE standard ANSI, o qual não tem campos de tempo. 	4	TIMESTAMP WITH TIME ZONE TIMESTAMP WITH LOCAL TIME ZONE
Fields	Tipo de <u>expr</u>										
1	DATE, Qualquer tipo TIMESTAMP, INTERVAL YEAR TO MONTH										
2	DATE, Qualquer tipo TIMESTAMP, INTERVAL DAY TO SECOND										
3	Qualquer tipo TIMESTAMP, INTERVAL DAY TO SECOND <ul style="list-style-type: none"> ▪ Tipo DATE é inválido ... porque, neste caso, a expr é assumida do tipo DATE standard ANSI, o qual não tem campos de tempo. 										
4	TIMESTAMP WITH TIME ZONE TIMESTAMP WITH LOCAL TIME ZONE										

Parte de Valor Data-Tempo (2/2)

Função	Funcionalidade
<p>EXTRACT(field FROM expr)</p> <p><u>field</u>:</p> <ul style="list-style-type: none"> 1. YEAR, MONTH, DAY 2. HOUR, MINUTE, SECOND 3. TIMEZONE_HOUR, TIMEZONE_MINUTE, TIMEZONE_REGION, TIMEZONE_ABBR 	<p>Tipo de retorno:</p> <ul style="list-style-type: none"> ▪ VARCHAR2 -- <u>field</u>: TIMEZONE_REGION e TIMEZONE_ABBR. ▪ NUMBER -- <u>field</u>: restantes. <p>Retorno UNKNOWN:</p> <ul style="list-style-type: none"> ▪ Em algumas combinações ambíguas de <u>field</u> e <u>expr</u>. <pre>SELECT EXTRACT(TIMEZONE_REGION FROM TIMESTAMP '1999-01-01 10:00:00 -08:00') FROM DUAL;</pre> <pre>EXTRACT(TIMEZONE_REGIONFROMTIMESTAMP'1999-01-0110:00:00-08:00')</pre> <hr/> <pre>UNKNOWN</pre> <div style="background-color: #d3d3d3; padding: 10px; margin-top: 10px;"> <p>Ambiguidade</p> <ul style="list-style-type: none"> ▪ Devido a existirem vários nomes de regiões do fuso horário especificado. </div>

Funções Data-Tempo

- Parte de Tempo

Tipo DATE (Oracle) - Exemplo

Tabela datas:

	Script Output	Query Result
	SQL All Rows Fetched: 1 in 0\$0	
1	DATA_HORA	04-11-2018 22:30:00

```
Worksheet Query Builder
445 | SELECT EXTRACT(HOUR FROM data_hora)
446 | FROM datas;

Script Output Query Result
SQL | Executing:SELECT EXTRACT(HOUR FROM data_hora)FROM datas
ORA-30076: invalid extract field for extract source
30076. 00000 - "invalid extract field for extract source"
*Cause: The extract source does not contain the specified extract field.
*Action:
Error at Line: 445 Column: 26
```

```
Worksheet Query Builder
448 | SELECT EXTRACT(HOUR FROM CAST(data_hora AS TIMESTAMP))
449 | FROM datas;

Script Output Query Result
SQL | All Rows Fetched: 1 in 0$013 seconds
EXTRACT(HOURFROMCAST(DATA_HORAASTIMESTAMP))
1 22
```

```
Worksheet Query Builder
1 | SELECT TO_NUMBER(TO_CHAR(SYSDATE, 'HH24'))
2 | FROM DUAL;

Query Result
SQL | All Rows Fetched: 1 in 0$031 seconds
TO_NUMBER(TO_CHAR(SYSDATE,'HH24'))
1 5
```

■ Número de Meses (1/3)

Função	Funcionalidade
MONTHS_BETWEEN(date1, date2)	<p>Retorna o número de meses entre as datas <u>date1</u> e <u>date2</u>.</p> <p>Resultado:</p> <ul style="list-style-type: none">▪ $\text{date1} > \text{date2} \Rightarrow \text{resultado} > 0$▪ $\text{date1} < \text{date2} \Rightarrow \text{resultado} < 0$▪ Em geral<ul style="list-style-type: none">▪ Número decimal.▪ Número inteiro apenas se ambas as datas forem:<ul style="list-style-type: none">▪ O último dia do mês.▪ O mesmo dia do mês.

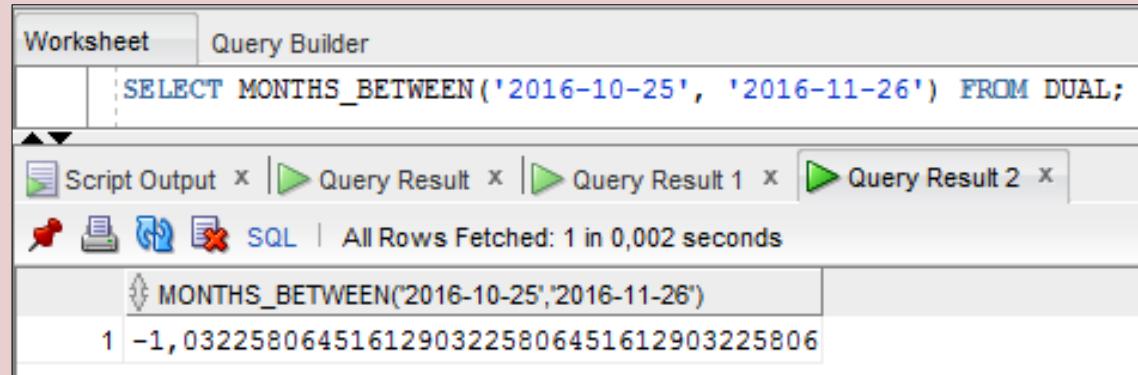
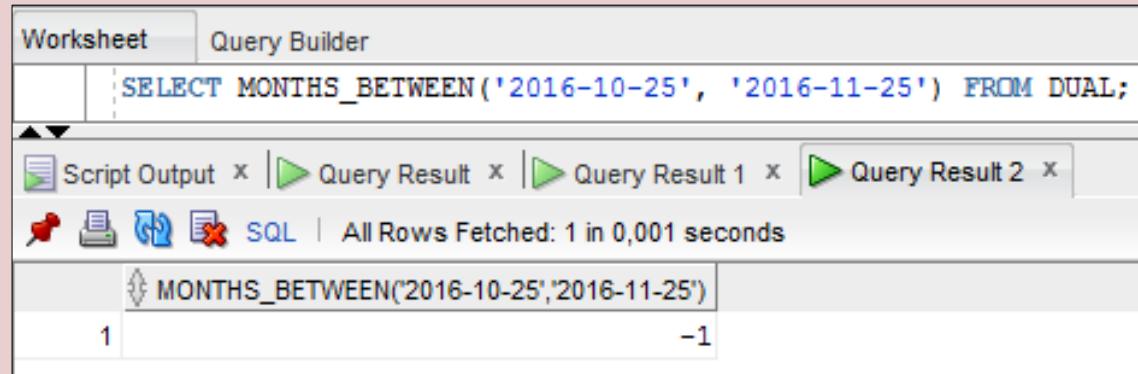
The screenshot shows the Oracle SQL Worksheet interface. The query window contains the following code:

```
Worksheet      Query Builder
SELECT MONTHS_BETWEEN('2016-10-25', '2016-11-26') FROM DUAL;
```

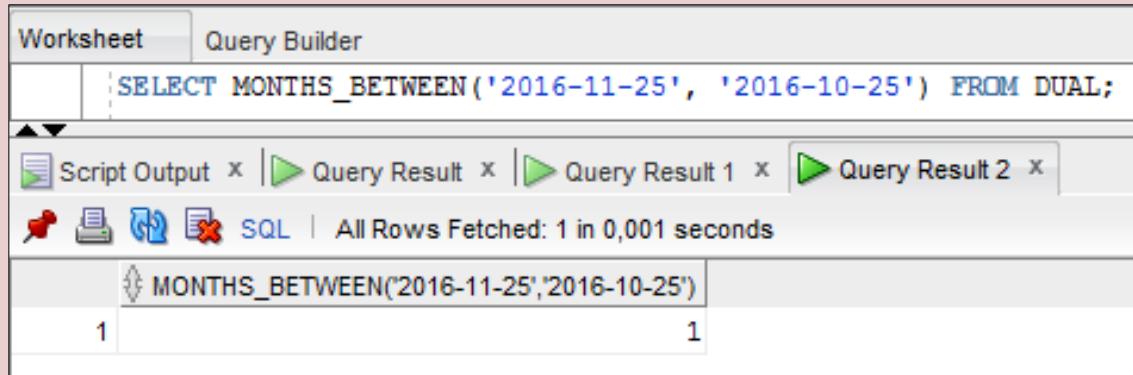
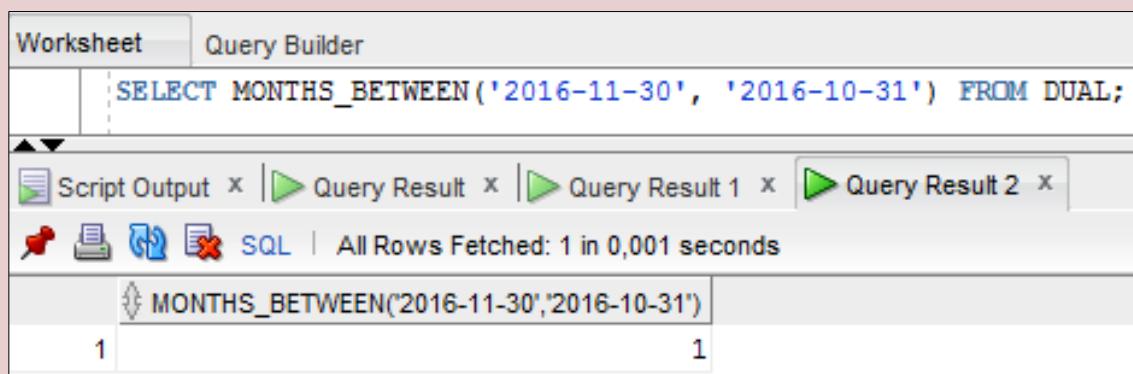
The results window shows the output of the query:

Script Output	Query Result	Query Result 1	Query Result 2
SQL	All Rows Fetched: 1 in 0,002 seconds		
	MONTHS_BETWEEN('2016-10-25','2016-11-26')		
1	-1,03225806451612903225806451612903225806		

■ Número de Meses (2/3)

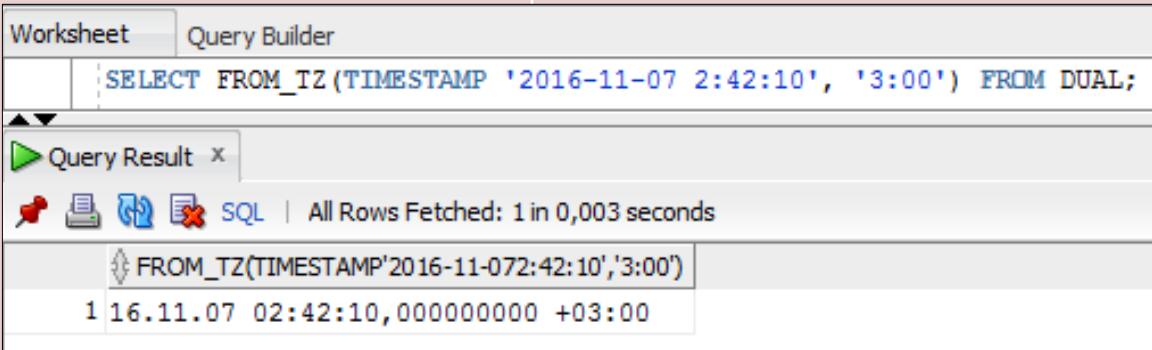
Função	Funcionalidade
MONTHS_BETWEEN(date1, date2)	 <p>The screenshot shows the Oracle SQL Developer interface. In the Worksheet tab, the query <code>SELECT MONTHS_BETWEEN('2016-10-25', '2016-11-26') FROM DUAL;</code> is run. The result is displayed in the Query Result tab, showing a single row with the value <code>-1,03225806451612903225806451612903225806</code>.</p>
	 <p>The screenshot shows the Oracle SQL Developer interface. In the Worksheet tab, the query <code>SELECT MONTHS_BETWEEN('2016-10-25', '2016-11-25') FROM DUAL;</code> is run. The result is displayed in the Query Result tab, showing a single row with the value <code>-1</code>.</p>

■ Número de Meses (3/3)

Função	Funcionalidade		
MONTHS_BETWEEN(date1, date2)	 <p>The screenshot shows the Oracle SQL Developer interface. In the Worksheet tab, the query <code>SELECT MONTHS_BETWEEN('2016-11-25', '2016-10-25') FROM DUAL;</code> is run. The results are displayed in the Query Result tab, showing a single row with the value 1.</p> <table border="1"><thead><tr><th>MONTHS_BETWEEN('2016-11-25','2016-10-25')</th></tr></thead><tbody><tr><td>1</td></tr></tbody></table>	MONTHS_BETWEEN('2016-11-25','2016-10-25')	1
MONTHS_BETWEEN('2016-11-25','2016-10-25')			
1			
	 <p>The screenshot shows the Oracle SQL Developer interface. In the Worksheet tab, the query <code>SELECT MONTHS_BETWEEN('2016-11-30', '2016-10-31') FROM DUAL;</code> is run. The results are displayed in the Query Result tab, showing a single row with the value 1.</p> <table border="1"><thead><tr><th>MONTHS_BETWEEN('2016-11-30','2016-10-31')</th></tr></thead><tbody><tr><td>1</td></tr></tbody></table>	MONTHS_BETWEEN('2016-11-30','2016-10-31')	1
MONTHS_BETWEEN('2016-11-30','2016-10-31')			
1			

Conversão de Tipos de Dados

Função	Funcionalidade
<code>FROM_TZ(timestamp_value, time_zone_value)</code>	<p>Converte o valor TIMESTAMP <u>timestamp_value</u> e o fuso horário <u>zone_value</u> num valor TIMESTAMP WITH TIME ZONE.</p> <p>Valor <u>zone_value</u>:</p> <ul style="list-style-type: none"> ▪ String no formato 'TZH:TZM'. ▪ Expressão string que retorna string no formato TZR com opção TZD.



The screenshot shows the Oracle SQL Developer interface. In the 'Worksheet' tab, a query is written:

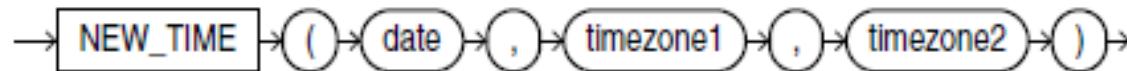
```
SELECT FROM_TZ(TIMESTAMP '2016-11-07 2:42:10', '3:00') FROM DUAL;
```

In the 'Query Result' tab, the output is displayed:

	FROM_TZ(TIMESTAMP'2016-11-07 2:42:10','3:00')
1	16.11.07 02:42:10,000000000 +03:00

Função NEW_TIME (1/2)

Sintaxe



`NEW_TIME(date, timezone1, timezone2)`

Semântica

Retorna a data e o tempo, no fuso horário **timezone2**, da data-tempo **date** no fuso horário **timezone1**.

Tipo de retorno:

- DATE

Para visualizar a parte do tempo.

Tipo do argumento **date**:

Formato por omissão:

NLS_DATE_FORMAT

RR.MM.DD

- Qualquer tipo data-tempo.

Worksheet Query Builder

```

ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MON-YYYY HH24:MI:SS';
SELECT NEW_TIME(TO_DATE('2016-11-07 23:21:10','YYYY-MM-DD HH24:MI:SS'), 'GMT', 'AST') FROM DUAL;
  
```

Script Output Query Result

SQL | All Rows Fetched: 1 in 0,002 seconds

1	07-NOV-2016 19:21:10
---	----------------------

Função NEW_TIME (2/2)

Semântica

Nº de Fusos Horários:

- Limitado.
- Siglas que podem ser usadas:

- AST, ADT: Atlantic Standard or Daylight Time
- BST, BDT: Bering Standard or Daylight Time
- CST, CDT: Central Standard or Daylight Time
- EST, EDT: Eastern Standard or Daylight Time
- GMT: Greenwich Mean Time
- HST, HDT: Alaska-Hawaii Standard Time or Daylight Time.
- MST, MDT: Mountain Standard or Daylight Time
- NST: Newfoundland Standard Time
- PST, PDT: Pacific Standard or Daylight Time
- YST, YDT: Yukon Standard or Daylight Time

Funções SQL

Numéricas

Carateres

Data-Tempo

Comparação Geral

Conversão

- **Funções**

- **GREATEST** -- Retorna o **maior** valor de lista de uma/mais expressões.
- **LEAST** -- Retorna o **menor** valor de lista de uma/mais expressões.

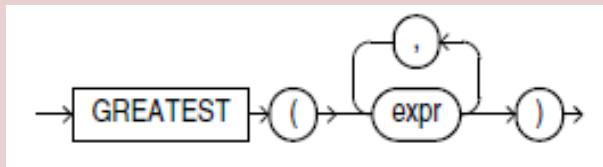
- **Tipo de Retorno**

- **Determinado** por:
 - Tipo da 1^a expressão da lista.

Função GREATEST (1/4)

Sintaxe

GREATEST(expr [, expr [, ...]]) -- [] = opcional



Semântica

Retorna o maior de lista de uma ou mais expressões.

Tipo de retorno:

- Determinado pelo
 - Tipo da 1^a expressão.

Função GREATEST (2/4)

Semântica (continuação)

Funcionamento:

1. Determina o tipo da 1º argumento (expressão).
2. Se esse tipo for:
 - a) Numérico:
 - i. Determina o argumento com a maior precedência numérica.
 - ii. Converte implicitamente os restantes argumentos para esse tipo.
 - iii. Compara os argumentos.
 - iv. Retorna o maior valor dos argumentos, do tipo determinado em i.
 - b) Não numérico:
 - i. Restantes argumentos são convertidos implicitamente para o tipo do 1º argumento.
 - ii. Compara os argumentos.
 - iii. Retorna o maior valor dos argumentos, do tipo do 1º argumento.
 - Tipo VARCHAR2

Função GREATEST (3/4)

Semântica (continuação)

Comparação de *strings* (por omissão):

- Binária

3 NLS_COMP

BINARY

- Baseada:

- Códigos numéricos dos caracteres.
 - Definidos no *character set* da BD.

- *Strings* tratadas:

- Como sequências de bytes.

The screenshot shows a SQL developer interface. In the 'Worksheet' tab, the query `SELECT GREATEST('PORTO', 'PORTALEGRE') FROM DUAL;` is entered. In the 'Query Result' tab, the output is shown in a table:

	GREATEST('PORTO','PORTALEGRE')
1	PORTO

Below the table, it says 'All Rows Fetched: 1 in 0,001 seconds'.

Comparação Linguística de *strings*:

- NLS_COMP = LINGUISTIC
- NLS_SORT ≠ BINARY.

Função GREATEST (4/4)

Semântica (continuação)

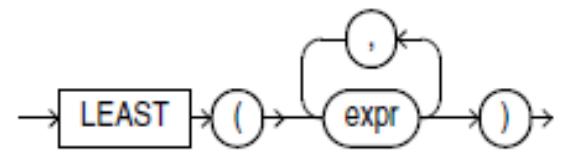
The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there are tabs for 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, displaying the SQL query: 'SELECT GREATEST(1, '3,192', '2,4') FROM DUAL;'. Below the query is the 'Query Result' window, which contains a single row of data. The result table has one column labeled 'GREATEST' with the value '3,192'. The status bar at the bottom of the window indicates 'All Rows Fetched: 1 in 0,001 seconds'.

	GREATEST
1	3,192

Função LEAST

Sintaxe

`LEAST(expr [, expr [, ...]])` -- [] = opcional



Semântica

Semelhante à função GREATEST.

- Diferença:
 - Retorna o menor valor da lista.

Worksheet Query Builder

```
SELECT LEAST('PORTO', 'PORTALEGRE') FROM DUAL;
```

Query Result x

SQL | All Rows Fetched: 1 in 0,002 seconds

	LEAST('PORTO','PORTALEGRE')
1	PORTALEGRE

Worksheet Query Builder

```
SELECT LEAST(1, '3,192', '2,4') FROM DUAL;
```

Query Result x

SQL | All Rows Fetched: 1 in 0 seconds

	LEAST(1,'3,192','2,4')
1	1

Funções SQL

Numéricas

Carateres

Data-Tempo

Comparação Geral

Conversão

■ Características Gerais

- Convertem explicitamente ... tipos de dados.
 - Nomeadamente:
 - Convertem valor de um tipo de dados num valor de outro tipo.
 - Exemplo:
 - Convertem data do tipo *string* numa data do tipo **DATE**.
- Nomes de funções (genericamente)
 - Iniciados com **TO_**
 - Exemplos:
 - TO_DATE
 - TO_CHAR
 - TO_TIMESTAMP
 - TO_YMINTERVAL
 - Com formato: tipo_de_dados_entrada TO tipo_de_dados_retornado
 - Exemplos:
 - NUMTODSINTERVAL
 - NUMTOYMIINTERVAL
 - BIN_TO_NUM

Categoria	Função	Funcionalidade
Para Número	BIN TO NUM	Converte um vetor de bits no número decimal equivalente.
	TO NUMBER	Converte um valor numérico/ <i>string</i> numérica num valor do tipo NUMBER.
	TO BINARY FLOAT	Converte valor num número de vírgula-flutuante de precisão simples.
	TO BINARY DOUBLE	Converte valor num número de vírgula-flutuante de precisão dupla.
Para <i>String</i> de Carateres	TO CHAR(number)	Converte número numa <i>string</i> com o formato indicado.
	TO CHAR(datetime)	Converte valor data-tempo numa <i>string</i> com o formato indicado.
Para Data-Tempo	TO DATE	Converte uma <i>string</i> data-tempo num valor do tipo DATE.
	TO TIMESTAMP	Converte uma <i>string</i> data-tempo num valor do tipo TIMESTAMP.
	TO TIMESTAMP TZ	Converte uma <i>string</i> data-tempo num valor TIMESTAMP WITH TIME ZONE.
Para Intervalo	TO YMINTERVAL	Converte uma <i>string</i> num valor INTERVAL YEAR TO MONTH.
	TO DSINTERVAL	Converte uma <i>string</i> num valor INTERVAL DAY TO SECOND.
	NUMTOYMINTERVAL	Converte um número num valor INTERVAL YEAR TO MONTH.
	NUMTODSINTERVAL	Converte um número num valor INTERVAL DAY TO SECOND.
Para Outro Tipo Compatível	CAST	Converte um valor de um tipo de dados predefinido num valor de outro tipo de dados predefinido compatível.

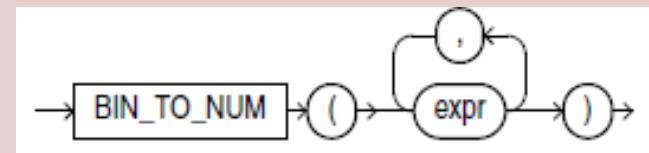
Conversão Explícita de Tipos de Dados

de	para	CHAR VARCHAR2	NUMBER	Data-Tempo Intervalo	BINARY_FLOAT	BINARY_DOUBLE
CHAR VARCHAR2	TO_CHAR (char)	TO_NUMBER		TO_DATE TO_TIMESTAMP TO_TIMESTAMP_TZ TO_YMINTERVAL TO_DSINTERVAL	TO_BINARY_FLOAT	TO_BINARY_DOUBLE
NUMBER	TO_CHAR (number)			TO_DATE NUMTOYMINTEGERVAL NUMTODSINTERVAL	TO_BINARY_FLOAT	TO_BINARY_DOUBLE
Data-Tempo Intervalo	TO_CHAR (date)					
BINARY_FLOAT	TO_CHAR (number)	TO_NUMBER			TO_BINARY_FLOAT	TO_BINARY_DOUBLE
BINARY_DOUBLE	TO_CHAR (number)	TO_NUMBER			TO_BINARY_FLOAT	TO_BINARY_DOUBLE

Função BIN_TO_NUM

Sintaxe

`BIN_TO_NUM(expr [, expr [, ...]])` -- [] = opcional



Semântica

Converte um vetor de bits no número decimal equivalente.

Tipo de retorno:

- NUMBER.

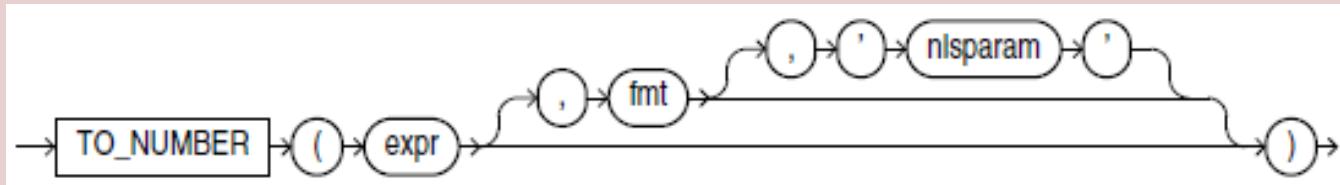
Cada argumento:

- Representa:
 - Um bit do vetor.
- Tipo:
 - Numérico qualquer.
 - Alfanuméricico:
 - Se valor convertível para o tipo NUMBER.

Worksheet	Query Builder				
	<code>SELECT BIN_TO_NUM(1,0,1,0) FROM DUAL;</code>				
Query Result x X Y Z SQL All Rows Fetched: 1 in 0,002 seconds					
	<table border="1"> <thead> <tr> <th></th> <th>BIN_TO_NUM(1,0,1,0)</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>10</td> </tr> </tbody> </table>		BIN_TO_NUM(1,0,1,0)	1	10
	BIN_TO_NUM(1,0,1,0)				
1	10				

Função TO_NUMBER (1/3)

Sintaxe



TO_NUMBER(expr)

TO_NUMBER(expr, fmt)

TO_NUMBER(expr, fmt, 'nlsparam')

Semântica

Converte o valor da expressão expr do tipo BINARY_DOUBLE ou CHAR, VARCHAR2 ou NVARCHAR2 contendo um número com o formato especificado no modelo de formato fmt, num valor do tipo NUMBER.

Tipo de retorno:

- NUMBER

1	123,45

Função TO_NUMBER (2/3)

Semântica

Modelo de formato fmt:

- Opcional.
- Aplicado a expr:
 - Tipo *string* de caracteres.

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there are tabs for 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, containing the SQL query: `SELECT TO_NUMBER('1234.45','9999.99') FROM DUAL;`. Below the worksheet, the 'Query Result' tab is open, showing the output of the query. The output table has two columns: 'TO_NUMBER('1234.45','9999.99')' and '1'. The value '1234,45' is displayed in the first row of the second column. The status bar at the bottom of the window indicates 'All Rows Fetched: 1 in 0,003 seconds'.

- Quando é omitido:
 - Argumento expr:
 - Deve ter o formato de números por omissão:

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there are tabs for 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, containing the SQL query: `SELECT TO_NUMBER('1234,45') FROM DUAL;`. Below the worksheet, the 'Query Result' tab is open, showing the output of the query. The output table has two columns: 'TO_NUMBER('1234,45')' and '1'. The value '1234,45' is displayed in the first row of the second column. The status bar at the bottom of the window indicates 'All Rows Fetched: 1 in 0,001 seconds'.

Função TO_NUMBER (3/3)

Semântica

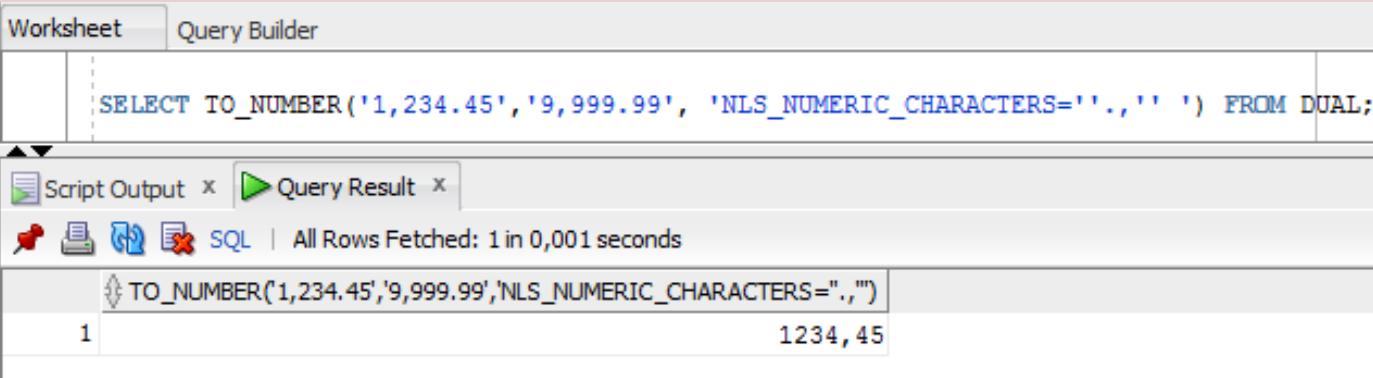
Parâmetro NLS 'nlsparam' :

- Opcional.
- Especifica os seguintes caracteres da expr:
 - Separador decimal (d).
 - Separador de grupos (g).
 - Símbolo da moeda local.
 - Símbolo de moeda internacional.
- Formato:
 - 'NLS_NUMERIC_CHARACTERS = "dg"'
NLS_CURRENCY=' 'texto'
NLS_ISO_CURRENCY= território '

Separadores por omissão:

- Decimal (,)
- Grupos (.)

13 NLS_NUMERIC_CHARACTERS , ..



The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there are tabs for 'Worksheet' and 'Query Builder'. Below them, a code editor window contains the following SQL query:

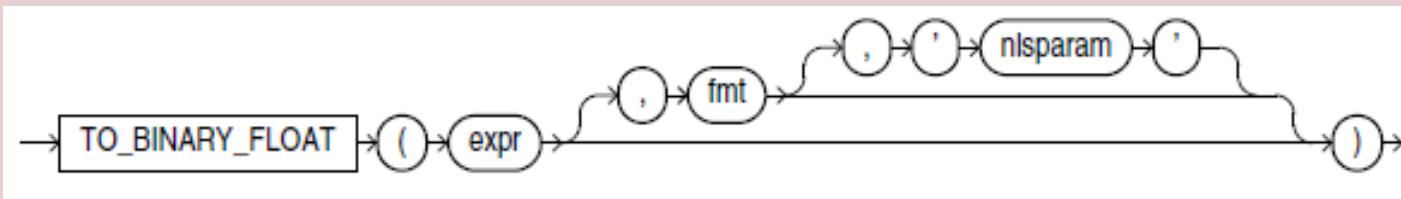
```
SELECT TO_NUMBER('1,234.45','9,999.99', 'NLS_NUMERIC_CHARACTERS=''.,''') FROM DUAL;
```

Below the code editor, there are two tabs: 'Script Output' and 'Query Result'. Under 'Query Result', there is a toolbar with icons for refresh, save, and cancel, followed by the text 'SQL | All Rows Fetched: 1 in 0,001 seconds'. The results table shows one row with the value '1234,45'.

	TO_NUMBER('1,234.45','9,999.99','NLS_NUMERIC_CHARACTERS=.,')
1	1234,45

Função TO_BINARY_FLOAT (1/3)

Sintaxe



TO_BINARY_FLOAT(expr)

TO_BINARY_FLOAT (expr, fmt)

TO_BINARY_FLOAT (expr, fmt, 'nlsparam')

Semântica

Converte o valor da expr num número de vírgula-flutuante com precisão-simples.

Tipo de retorno:

- BINARY_FLOAT.

Argumento expr:

- String de caracteres **compatível**.
- Valor numérico do tipo:
 - NUMBER
 - BINARY_FLOAT
 - BINARY_DOUBLE.

Conversões de *strings* / NUMBER:

- **Podem** ser imprecisas, porque:
 - BINARY_FLOAT usa precisão binária.
 - NUMBER e tipos carácter usam precisão decimal.

Conversões BINARY_DOUBLE para BINARY_FLOAT:

- **Podem** ser imprecisas, quando:
 - Valor BINARY_DOUBLE usa precisão não suportada por BINARY_FLOAT.

Função TO_BINARY_FLOAT (2/3)

Semântica

Argumentos fmt e 'nlsparam':

- Opcionais:
 - Para expr do tipo *string* de caracteres.
- fmt:
 - Especifica o modelo de formato da *string* expr.
- 'nlsparam':
 - Especifica os seguintes carateres:
 - Separador decimal (d).
 - Separador de grupos (g).
 - Símbolo da moeda local.
 - Símbolo de moeda internacional.
 - Formato:
 - 'NLS_NUMERIC_CHARACTERS = "dg"'
'NLS_CURRENCY=' 'texto' '
'NLS_ISO_CURRENCY= território '

Função TO_BINARY_FLOAT (3/3)

Semântica

Exemplos:

Worksheet Query Builder

```
SELECT TO_BINARY_FLOAT(234987.98) FROM DUAL;
```

Query Result x

SQL | All Rows Fetched: 1 in 0,008 seconds

TO_BINARY_FLOAT(234987.98)
1 234987,98

Worksheet Query Builder

```
SELECT TO_BINARY_FLOAT('234,987.26','999,999.99') FROM DUAL;
```

Query Result x

SQL | All Rows Fetched: 1 in 0,002 seconds

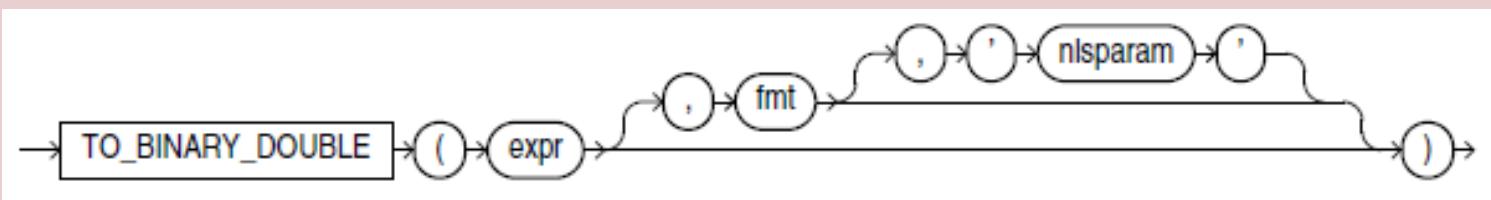
TO_BINARY_FLOAT('234,987.26','999,999.99')
1 234987,27

String de caracteres:

- **Illegal** usar elementos de formato de número de vírgula-flutuante (sufixos):
 - F, f
 - D, d

Função TO_BINARY_DOUBLE (1/3)

Sintaxe



TO_BINARY_DOUBLE(expr)

TO_BINARY_DOUBLE(expr, fmt)

TO_BINARY_DOUBLE(expr, fmt, 'nlsparam')

Semântica

Converte o valor da expr num número de vírgula-flutuante com precisão-dupla.

Tipo de retorno:

- BINARY_DOUBLE.

Argumento expr:

- String de caracteres **compatível**.
- Valor numérico do tipo:
 - NUMBER
 - BINARY_FLOAT
 - BINARY_DOUBLE.

Conversões de *strings* / NUMBER:

- Podem ser imprecisas, porque:
 - BINARY_DOUBLE usa precisão binária.
 - NUMBER e tipos caráter usam precisão decimal.

Função TO_BINARY_DOUBLE (2/3)

Semântica

Argumentos fmt e 'nlsparam':

- Opcionais:
 - Para expr do tipo *string* de caracteres.
- fmt:
 - Especifica o modelo de formato da *string* expr.
- 'nlsparam':
 - Especifica os seguintes carateres:
 - Separador decimal (d).
 - Separador de grupos (g).
 - Símbolo da moeda local.
 - Símbolo de moeda internacional.
 - Formato:
 - 'NLS_NUMERIC_CHARACTERS = "dg"'
'NLS_CURRENCY=' 'texto' '
'NLS_ISO_CURRENCY= território '

Função TO_BINARY_DOUBLE (3/3)

Semântica

Exemplos:

The screenshot shows the Oracle SQL Worksheet interface. In the 'Worksheet' tab, the query `SELECT TO_BINARY_DOUBLE(234987.98) FROM DUAL;` is entered. In the 'Query Result' tab, the output is shown in a table with one row. The column header is `TO_BINARY_DOUBLE(234987.98)` and the value is `1 234987,98`. The status bar indicates "All Rows Fetched: 1 in 0,001 seconds".

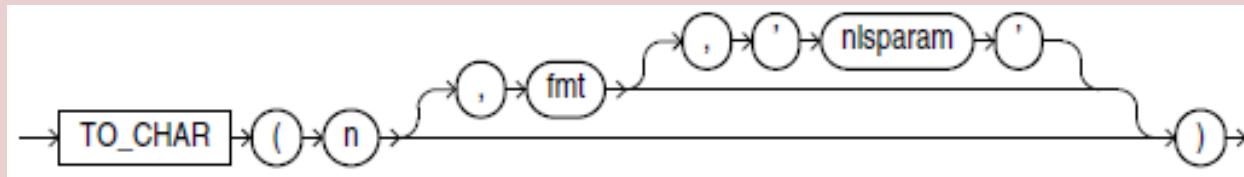
The screenshot shows the Oracle SQL Worksheet interface. In the 'Worksheet' tab, the query `SELECT TO_BINARY_DOUBLE('234,987.26','999,999.99') FROM DUAL;` is entered. In the 'Query Result' tab, the output is shown in a table with one row. The column header is `TO_BINARY_DOUBLE('234,987.26','999,999.99')` and the value is `1 234987,26`. The status bar indicates "All Rows Fetched: 1 in 0,001 seconds".

String de caracteres:

- **Illegal** usar elementos de formato de número de vírgula-flutuante (sufixos):
 - F, f
 - D, d

Função TO_CHAR(number) (1/3)

Sintaxe



TO_CHAR(n)

TO_CHAR(n, fmt)

TO_CHAR(n, fmt, 'nlsparam')

Semântica

Converte o número n num valor do tipo VARCHAR2.

Tipo de retorno:

- VARCHAR2.

Argumento n:

- Valor numérico do tipo:
 - NUMBER
 - BINARY_FLOAT
 - BINARY_DOUBLE.

Função TO_CHAR(number) (2/3)

Semântica

Argumentos fmt e 'nlsparam':

- Opcionais.
- fmt:
 - Especifica o modelo de formato numérico do valor n.
 - Quando é omitido:
 - n convertido para VARCHAR2 com comprimento exato para guardar os seus dígitos significativos.
- 'nlsparam':
 - Especifica os seguintes carateres:
 - Separador decimal (d).
 - Separador de grupos (g).
 - Símbolo da moeda local.
 - Símbolo de moeda internacional.
 - Formato:
 - 'NLS_NUMERIC_CHARACTERS = "dg"'
 - NLS_CURRENCY=' 'texto' '
 - NLS_ISO_CURRENCY= território '

Função TO_CHAR(number) (3/3)

Semântica

Exemplos:

Worksheet Query Builder

```
SELECT TO_CHAR(234987.98) FROM DUAL;
```

Query Result x

SQL | All Rows Fetched: 1 in 0,001 second

	TO_CHAR(234987.98)
1	234987,98

Worksheet Query Builder

```
SELECT TO_CHAR(234987, '999,999.99') FROM DUAL;
```

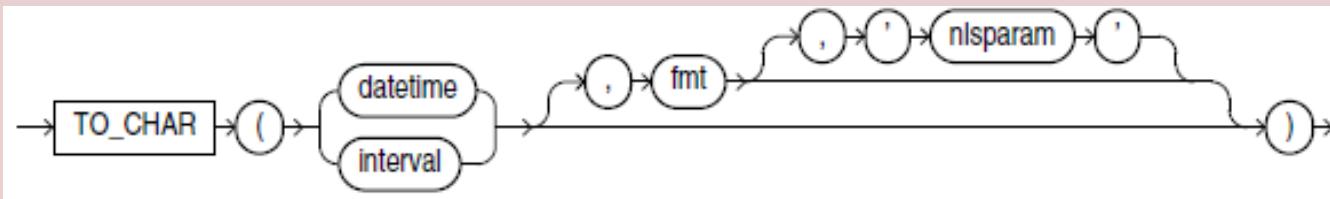
Query Result x

SQL | All Rows Fetched: 1 in 0,001 seconds

	TO_CHAR(234987,'999,999.99')
1	234,987.00

Função TO_CHAR(datetime) (1/5)

Sintaxe



TO_CHAR(datetime)

TO_CHAR(datetime, fmt)

TO_CHAR(datetime, fmt, 'nlsparam')

Semântica

Converte o valor datetime, de qualquer tipo de dados data-tempo, num valor do tipo VARCHAR2 com o formato especificado no modelo de formato fmt.

A screenshot of the Oracle SQL Worksheet interface. The top tab bar shows 'Worksheet' and 'Query Builder'. The main area contains a query: 'SELECT TO_CHAR(DATE '2016-11-07', 'DD-MM-YYYY') FROM DUAL;'. Below this is a 'Query Result' section with a green play button icon. The results table shows one row with the value '07-11-2016'.

TO_CHAR(DATE'2016-11-07','DD-MM-YYYY')
07-11-2016

Função TO_CHAR(datetime) (2/5)

Semântica

Modelo de formato fmt:

- Opcional.
- Quando é omitido:
 - datetime DATE → VARCHAR2 com NLS_DATE_FORMAT.
 - datetime TIMESTAMP → VARCHAR2 com NLS_TIMESTAMP_FORMAT.
 - datetime TIMESTAMP WITH LOCAL TIME ZONE → VARCHAR2 com NLS_TIMESTAMP_FORMAT
 - datetime TIMESTAMP WITH TIME ZONE → VARCHAR2 com NLS_TIMESTAMP_TZ_FORMAT

The screenshot shows the Oracle SQL Developer interface. In the top tab bar, 'Worksheet' is selected. Below it, the 'Query Builder' tab is visible. The worksheet area contains the SQL query:

```
SELECT TO_CHAR(DATE '2016-11-7') FROM DUAL;
```

In the 'Query Result' tab, there is a toolbar with icons for refresh, save, copy, and delete, followed by the text 'SQL | All Rows Fetched: 1 in 0,002 seconds'. The result table displays one row:

TO_CHAR(DATE'2016-11-7')
1 16.11.07

Formato por omissão:

5	NLS_DATE_FORMAT	RR.MM.DD
---	-----------------	----------

Função TO_CHAR(datetime) (3/5)

Semântica

Parâmetro NLS 'nlsparam' :

- Opcional.
- Formato:
 - 'NLS_DATE_LANGUAGE = language'
- Especifica:
 - Linguagem para descrever os nomes de meses e dias. -- nomes de anos = inglês (sempre).
- Quando é omitido:
 - Linguagem dos nomes de meses e dias ... especificada em NLS_DATE_LANGUAGE.

Worksheet Query Builder

```
SELECT TO_CHAR(TO_DATE('2016-11-6','YYYY-MM-DD'),'YEAR-MONTH-DAY') FROM DUAL;
```

Query Result

SQL | All Rows Fetched: 1 in 0,002 seconds

TO_CHAR(TO_DATE('2016-11-6','YYYY-MM-DD'),'YEAR-MONTH-DAY')
1 TWENTY SIXTEEN-NOVEMBRO -DOMINGO

Worksheet Query Builder

```
SELECT TO_CHAR(TO_DATE('2016-11-6','YYYY-MM-DD'),'YEAR-MONTH-DAY', 'NLS_DATE_LANGUAGE=ENGLISH') FROM DUAL;
```

Query Result

SQL | All Rows Fetched: 1 in 0,003 seconds

TO_CHAR(TO_DATE('2016-11-6','YYYY-MM-DD'),'YEAR-MONTH-DAY','NLS_DATE_LANGUAGE=ENGLISH')
1 TWENTY SIXTEEN-NOVEMBER -SUNDAY

Função TO_CHAR(datetime) (4/5)

Semântica

Exemplos:

Worksheet Query Builder

```
SELECT TO_CHAR(TO_DATE('2016-6-5 15:10:22','YYYY-MM-DD HH24:MI:SS')) FROM DUAL;
```

Query Result x

SQL | All Rows Fetched: 1 in 0,004 seconds

	TO_CHAR(TO_DATE('2016-6-5 15:10:22','YYYY-MM-DD HH24:MI:SS'))
1	16.06.05

Formato por omissão:

5 NLS_DATE_FORMAT	RR.MM.DD
-------------------	----------

Worksheet Query Builder

```
SELECT TO_CHAR(TO_DATE('2016-6-5 15:10:22','YYYY-MM-DD HH24:MI:SS'),
               'YYYY-MM-DD HH24:MI:SS') FROM DUAL;
```

Query Result x

SQL | All Rows Fetched: 1 in 0,002 seconds

	TO_CHAR(TO_DATE('2016-6-5 15:10:22','YYYY-MM-DD HH24:MI:SS'),'YYYY-MM-DD HH24:MI:SS')
1	2016-06-05 15:10:22

Função TO_CHAR(datetime) (5/5)

Semântica

Exemplos:

Worksheet Query Builder

```
SELECT TO_CHAR(TO_DATE('2016-11-6','YYYY-MM-DD'),'FMDay-Month-YEAR') FROM DUAL;
```

Query Result

All Rows Fetched: 1 in 0,003 seconds

TO_CHAR(TO_DATE('2016-11-6','YYYY-MM-DD'),'FMDAY-MONTH-YEAR')
1 Domingo-Novembro-TWENTY SIXTEEN

Worksheet Query Builder

```
SELECT TO_CHAR(TO_DATE('2016-11-6','YYYY-MM-DD'),'FMDay" de "Month" de "YYYY') FROM DUAL;
```

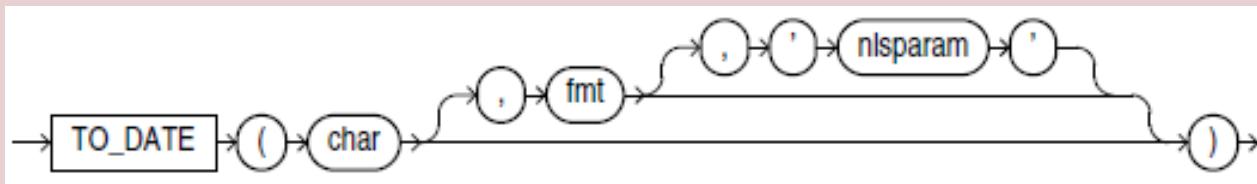
Query Result

All Rows Fetched: 1 in 0,002 seconds

TO_CHAR(TO_DATE('2016-11-6','YYYY-MM-DD'),'FMDAY"DE"MONTH"DE"YYYY')
1 Domingo de Novembro de 2016

Função TO_DATE (1/3)

Sintaxe



TO_DATE(char)

TO_DATE(char, fmt)

TO_DATE(char, fmt, 'nlsparam')

Semântica

Converte o valor char, do tipo CHAR, VARCHAR2 ou NVARCHAR2, com o formato especificado no modelo de formato fmt, num valor do tipo DATE.

Tipo de retorno:

- DATE

Converte apenas:

- Valores data-tempo
do tipo DATE.

Worksheet Query Builder

```
SELECT TO_DATE('2016-11-7 23:55:18','YYYY-MM-DD HH24:MI:SS') FROM DUAL;
```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0,001 seconds

	TO_DATE('2016-11-7 23:55:18','YYYY-MM-DD HH24:MI:SS')
1	16.11.07

Formato por omissão:

5 NLS_DATE_FORMAT	RR.MM.DD
-------------------	----------

Função TO_DATE (2/3)

Semântica

Modelo de formato fmt:

- Opcional.
- Quando é omitido:
 - Argumento char:
 - Deve ter o formato de datas por omissão:

Boa regra prática:

- Usar sempre modelos de formato fmt.
 - Código independente do formato por omissão.

The screenshot shows the Oracle SQL Developer interface. In the 'Worksheet' tab, the following SQL query is entered:

```
SELECT TO_DATE('2016-11-7') FROM DUAL;
```

In the 'Query Result' tab, the output is displayed as:

	TO_DATE('2016-11-7')
1	16.11.07

Below the table, it says "All Rows Fetched: 1 in 0,002 seconds".

- fmt = J -- J = Julian
 - Argumento char deve ser um número inteiro.

Função TO_DATE (3/3)

Semântica

Parâmetro NLS 'nlsparam' :

- Opcional.
- Formato: 'NLS_DATE_LANGUAGE = language'
- Especifica: Linguagem do texto da string char.

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there are tabs for 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, containing the following SQL code:

```
SELECT TO_DATE('2016-NOVEMBER-7', 'YYYY-Month-DD', 'NLS_DATE_LANGUAGE=ENGLISH') FROM DUAL;
```

Below the worksheet, the 'Script Output' and 'Query Result' tabs are visible. The 'Query Result' tab shows the output of the query:

	TO_DATE('2016-NOVEMBER-7','YYYY-MONTH-DD','NLS_DATE_LANGUAGE=ENGLISH')
1	16.11.07

Below the table, it says 'All Rows Fetched: 1 in 0,002 seconds'.

- Quando é omitido: Linguagem especificada em NLS_DATE_LANGUAGE.

The screenshot shows the Oracle SQL Developer interface, similar to the previous one. The 'Worksheet' tab contains the same SQL code:

```
SELECT TO_DATE('2016-NOVEMBRO-7', 'YYYY-Month-DD') FROM DUAL;
```

The 'Query Result' tab shows the output:

	TO_DATE('2016-NOVEMBRO-7','YYYY-MONTH-DD')
1	16.11.07

Below the table, it says 'All Rows Fetched: 1 in 0,005 seconds'.

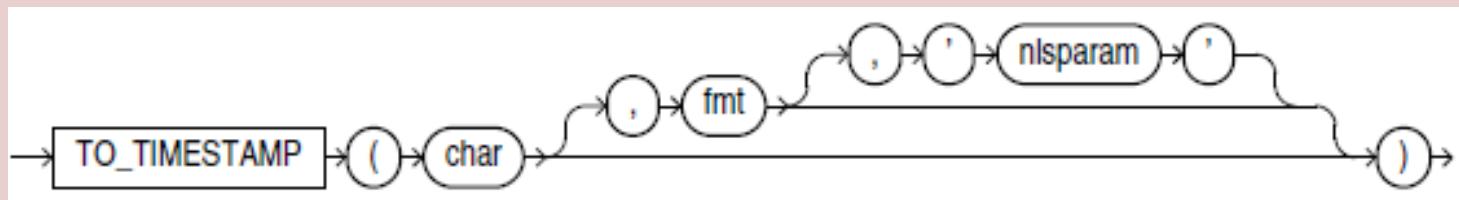
A callout box points to a table at the bottom right of the screen. The table has three columns:

	NLS_DATE_LANGUAGE	
6	PORTUGUESE	

The text 'Linguagem por omissão:' is displayed above the table.

Função TO_TIMESTAMP (1/4)

Sintaxe



TO_TIMESTAMP(char)

TO_TIMESTAMP(char, fmt)

TO_TIMESTAMP(char, fmt, 'nlsparam')

Semântica

Converte o valor char, do tipo CHAR, VARCHAR2 ou NVARCHAR2, com o formato especificado no modelo de formato fmt, num valor do tipo TIMESTAMP.

Tipo de retorno:

- TIMESTAMP

Query Result
<pre>SELECT TO_TIMESTAMP('2016-11-07 13:51:10','YYYY-MM-DD HH24:MI:SS.FF') FROM DUAL;</pre>
<pre>TO_TIMESTAMP('2016-11-0713:51:10','YYYY-MM-DDHH24:MI:SS.FF')</pre>
<pre>1 16.11.07 13:51:10,000000000</pre>

Função TO_TIMESTAMP (2/4)

Semântica

- Exemplo: tempo

The screenshot shows the Oracle SQL Worksheet interface. In the top pane, there is a code editor with the following SQL statement:

```
SELECT TO_TIMESTAMP('15:30','HH24:MI')
FROM DUAL;
```

In the bottom pane, there are two tabs: "Script Output" and "Query Result". The "Query Result" tab is active, showing the output of the query:

	TO_TIMESTAMP('15:30','HH24:MI')
1	16.11.01 15:30:00,000000000

Data por omissão:

- 1º dia do mês atual.

The screenshot shows the Oracle SQL Worksheet interface. In the top pane, there is a code editor with the following SQL statement:

```
SELECT TO_CHAR(TO_TIMESTAMP('15:30','HH24:MI'),'HH24:MI')
FROM DUAL;
```

In the bottom pane, there are two tabs: "Script Output" and "Query Result". The "Query Result" tab is active, showing the output of the query:

	TO_CHAR(TO_TIMESTAMP('15:30','HH24:MI'),'HH24:MI')
1	15:30

Função TO_TIMESTAMP (3/4)

Semântica

Modelo de formato fmt:

- Opcional.
- Quando é omitido:
 - Argumento char:
 - Deve ter o formato de datas por omissão:

17 NLS_TIMESTAMP_FORMAT RR.MM.DD HH24:MI:SSXFF

The screenshot shows the Oracle SQL Developer interface. In the 'Worksheet' tab, a query is entered: `SELECT TO_TIMESTAMP('2016-11-7 3:16:20') FROM DUAL;`. In the 'Query Result' tab, the output is displayed in a table:

	TO_TIMESTAMP('2016-11-7 3:16:20')
1	16.11.07 03:16:20,0000000000

Função TO_TIMESTAMP (4/4)

Semântica

Parâmetro NLS 'nlsparam' :

- Opcional.
- Formato: 'NLS_DATE_LANGUAGE = language'
- Especifica: Linguagem do texto da string char.

The screenshot shows the Oracle SQL Worksheet interface. In the 'Worksheet' tab, a query is run:

```
SELECT TO_TIMESTAMP('2016-NOVEMBER-07','YYYY-MONTH-DD', 'NLS_DATE_LANGUAGE=ENGLISH') FROM DUAL;
```

The 'Query Result' tab displays the output:

	TO_TIMESTAMP('2016-NOVEMBER-07','YYYY-MONTH-DD','NLS_DATE_LANGUAGE=ENGLISH')
1	16.11.07 00:00:00,000000000

- Quando é omitido: Linguagem especificada em NLS_DATE_LANGUAGE.

The screenshot shows the Oracle SQL Worksheet interface. In the 'Worksheet' tab, a query is run:

```
SELECT TO_TIMESTAMP('2016-NOVEMBRO-07','YYYY-MONTH-DD') FROM DUAL;
```

The 'Query Result' tab displays the output:

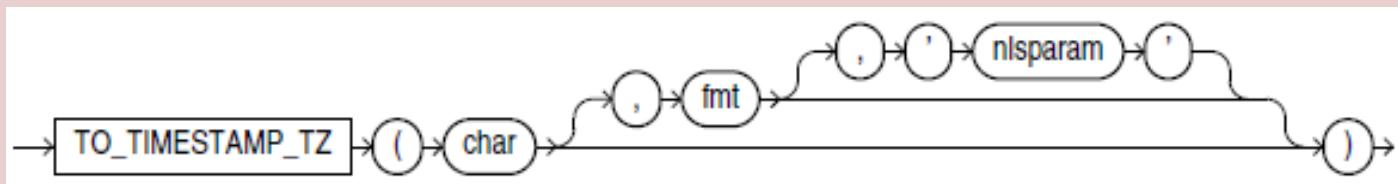
	TO_TIMESTAMP('2016-NOVEMBRO-07','YYYY-MONTH-DD')
1	16.11.07 00:00:00,000000000

A callout box points from the omitted NLS_DATE_LANGUAGE parameter in the query to the session parameter settings. The text 'Linguagem por omissão:' is highlighted in yellow.

6 NLS_DATE_LANGUAGE	PORTUGUESE
---------------------	------------

Função TO_TIMESTAMP_TZ (1/3)

Sintaxe



TO_TIMESTAMP_TZ(char)

TO_TIMESTAMP_TZ(char, fmt)

TO_TIMESTAMP_TZ(char, fmt, 'nlsparam')

Semântica

Converte o valor char, do tipo CHAR, VARCHAR2 ou NVARCHAR2, com o formato especificado no modelo de formato fmt, num valor do tipo TIMESTAMP WITH TIME ZONE.

Tipo de retorno:

- TIMESTAMP WITH TIME ZONE

Worksheet Query Builder

```
SELECT TO_TIMESTAMP_TZ('2016-11-07 13:51:10 -3:00','YYYY-MM-DD HH24:MI:SS TZH:TZM') FROM DUAL;
```

Query Result

All Rows Fetched: 1 in 0,003 seconds

TO_TIMESTAMP_TZ('2016-11-0713:51:10-3:00','YYYY-MM-DDHH24:MI:SSTZH:TZM')
1 16.11.07 13:51:10,000000000 -03:00

Função TO_TIMESTAMP_TZ (2/3)

Semântica

Modelo de formato fmt:

- Opcional.
- Quando é omitido:
 - Argumento char:
 - Deve ter o formato de datas por omissão:

18 NLS_TIMESTAMP_TZ_FORMAT RR.MM.DD HH24:MI:SSXFF TZR

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there are tabs for 'Worksheet' and 'Query Builder'. The main area contains a SQL query in the 'Worksheet' tab:

```
SELECT TO_TIMESTAMP_TZ('2016-11-07 13:51:10 -3:00') FROM DUAL;
```

Below the query, under the 'Query Result' tab, the results are displayed:

	TO_TIMESTAMP_TZ('2016-11-07 13:51:10 -3:00')
1	16.11.07 13:51:10,000000000 -03:00

The results show that the date and time '2016-11-07 13:51:10 -3:00' was converted to '16.11.07 13:51:10,000000000 -03:00' using the default format.

Função TO_TIMESTAMP_TZ (3/3)

Semântica

Parâmetro NLS 'nlsparam' :

- Opcional.
- Formato: 'NLS_DATE_LANGUAGE = language'
- Especifica: Linguagem do texto da string char.

Worksheet Query Builder

```
SELECT TO_TIMESTAMP_TZ('2016-NOVEMBER-07 13:51:10 8:00', 'YYYY-MONTH-DD HH24:MI:SS TZH:TZM',
                      'NLS_DATE_LANGUAGE=ENGLISH')
FROM DUAL;
```

Query Result x

SQL | All Rows Fetched: 1 in 0,001 seconds

◊ TO_TIMESTAMP_TZ('2016-NOVEMBER-0713:51:108:00','YYYY-MONTH-DDHH24:MI:SSTZH:TZM','NLS_DATE_LANGUAGE=ENGLISH')
1 16.11.07 13:51:10,000000000 +08:00

Worksheet Query Builder

```
SELECT TO_TIMESTAMP_TZ('2016-NOVEMBRO-07 13:51:10 8:00', 'YYYY-MONTH-DD HH24:MI:SS TZH:TZM') FROM DUAL;
```

Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0,002 seconds

◊ TO_TIMESTAMP_TZ('2016-NOVEMBRO-0713:51:108:00','YYYY-MONTH-DDHH24:MI:SSTZH:TZM')
1 16.11.07 13:51:10,000000000 +08:00

Linguagem por omissão:

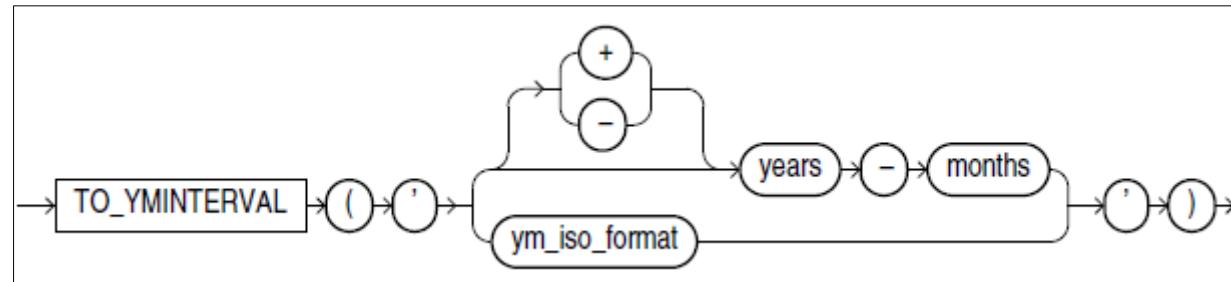
6 NLS_DATE_LANGUAGE PORTUGUESE

■ Funções Relacionadas

Conversão		Função	Exemplos	
De	Para valor do tipo			
String	INTERVAL YEAR TO MONTH	TO_YMINTERVAL	TO_YMINTERVAL('1-2')	-- 1 ano e 2 meses
			TO_YMINTERVAL('1-0')	-- 1 ano
			TO_YMINTERVAL('0-6')	-- 6 meses
	INTERVAL DAY TO SECOND	TO_DSINTERVAL	TO_DSINTERVAL('100 5:0:0')	-- 100 dias e 5 horas
			TO_DSINTERVAL('100 0:0:0')	-- 100 dias
			TO_DSINTERVAL('0 5:10:00')	-- 5 horas e 10 min.
Número (valor / expressão do tipo NUMBER)	INTERVAL YEAR TO MONTH	NUMTOYMINTERVAL	NUMTOYMINTERVAL(2, 'YEAR')	-- 2 anos
			NUMTOYMINTERVAL(2.3, 'year')	-- 2 anos e 4 meses
			NUMTOYMINTERVAL(2, 'month')	-- 2 meses
			NUMTOYMINTERVAL(2.4, 'MONTH')	-- 2 meses
			NUMTOYMINTERVAL(2.5, 'MONTH')	-- 3 meses
	INTERVAL DAY TO SECOND	NUMTODSINTERVAL	NUMTODSINTERVAL(10, 'DAY')	-- 10 dias
			NUMTODSINTERVAL(2.5, 'day')	-- 2 dias e 12 horas
			NUMTODSINTERVAL(10, 'HOUR')	-- 10 horas
			NUMTODSINTERVAL(10, 'MINUTE')	-- 10 minutos

■ Função Predefinida TO_YMINTERVAL (1/2)

■ Sintaxe

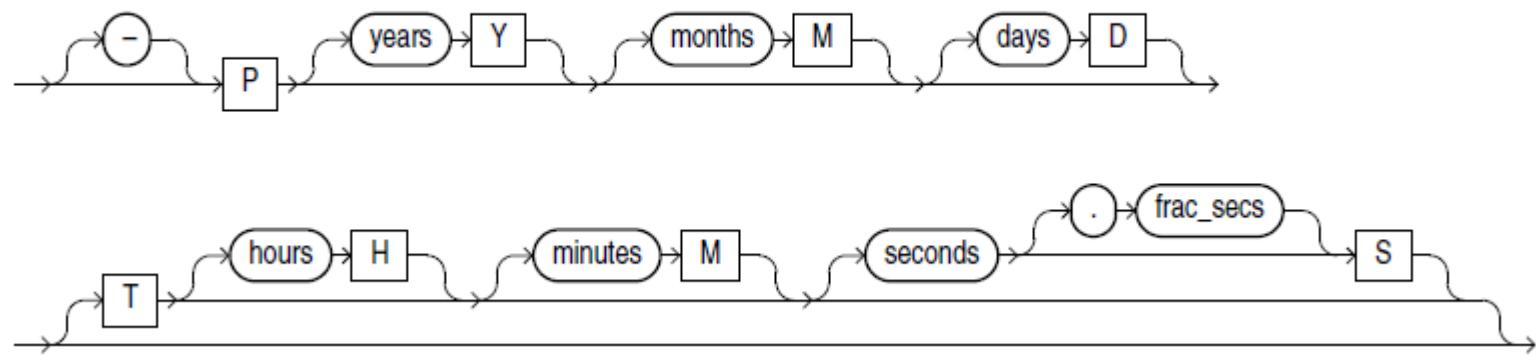


Argumentos num de dois formatos:

- Intervalo SQL.
- Duração ISO.

- Anos inteiro $\in [0, 999999999]$.
- Meses inteiro $\in [0, 11]$.
- Espaços - permitidos entre elementos.

ym_iso_format ::=



■ Função Predefinida TO_YMINTERVAL (2/2)

■ Exemplo

Worksheet Query Builder

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MON-YYYY';

SELECT "Data de Contratação", "Data de Contratação" + TO_YMINTERVAL('1-2') AS "Passados 14 Meses"
FROM empregados;
```

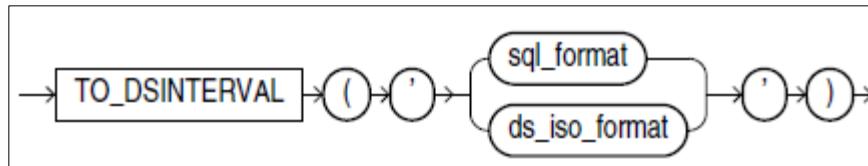
Script Output Query Result

SQL | Fetched 50 rows in 0,01 seconds

	Data de Contratação	Passados 14 Meses
1	17-JUN-2003	17-AGO-2004
2	21-SET-2005	21-NOV-2006
3	13-JAN-2001	13-MAR-2002
4	03-JAN-2006	03-MAR-2007
5	21-MAI-2007	21-JUL-2008
6	25-JUN-2005	25-AGO-2006

■ Função Predefinida TO_DSINTERVAL (1/2)

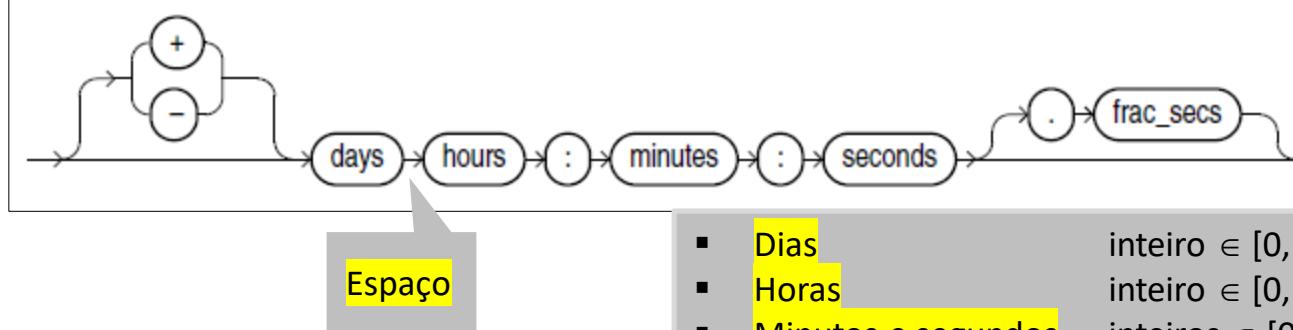
■ Sintaxe



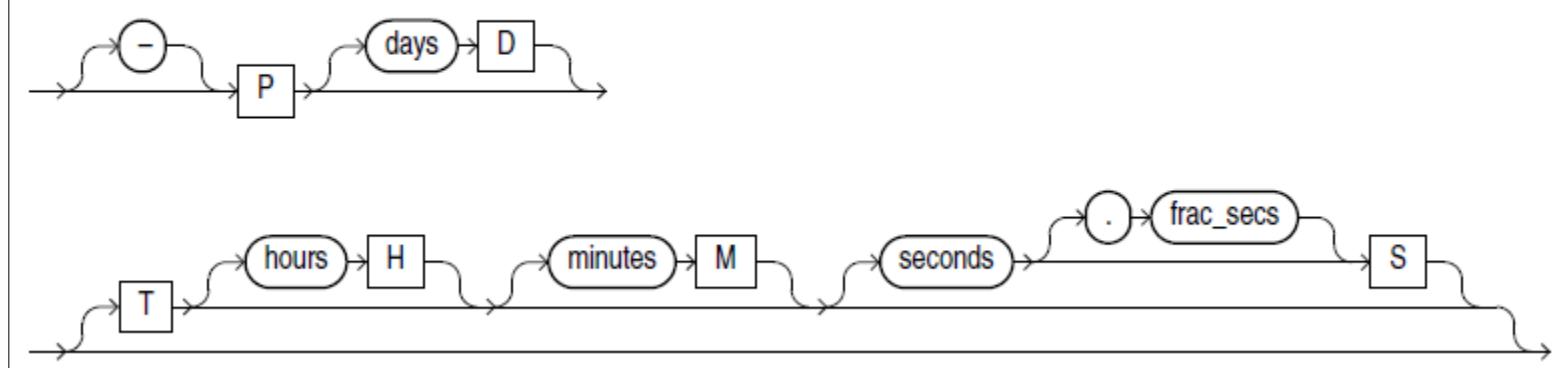
Argumentos num de dois formatos:

- Intervalo SQL
- Intervalo ISO (*standard*)

sql_format::=



ds_iso_format::=



■ Função Predefinida TO_DSINTERVAL (2/2)

■ Exemplo

The screenshot shows the Oracle SQL Developer interface. The top window is titled "Worksheet" and contains three SQL queries demonstrating different ways to add 100 days to the current date:

```
-- Data atual mais 100 dias
SELECT SYSDATE + TO_DSINTERVAL('P100D')
FROM DUAL;

-- Alternativa
SELECT SYSDATE + TO_DSINTERVAL('100 0:0:0')
FROM DUAL;

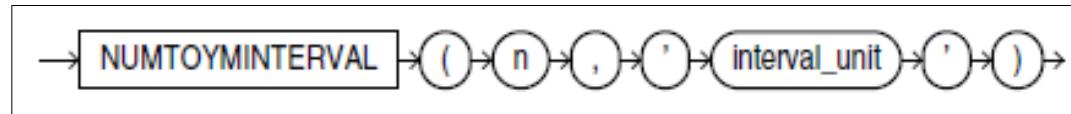
-- Alternativa
SELECT SYSDATE + INTERVAL '100' DAY(3)
FROM DUAL;
```

The bottom window is titled "Query Result" and displays the result of the first query:

	SYSDATE+TO_DSINTERVAL('P100D')
1	07-MAR-2017

- Função Predefinida NUMTOYMINTERVAL

- Sintaxe

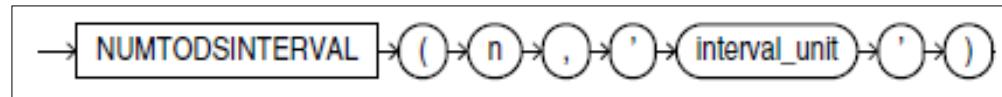


- interval_unit

- Especifica:
 - Unidade do número n:
 - 'YEAR' / 'MONTH' -- case insensitive
 - 'year' / 'month'
 - Tipo:
 - CHAR
 - VARCHAR2
 - Espaços no início/fim:
 - Ignorados.
 - Resultado
 - Por omissão:
 - Precisão = 9

■ Função Predefinida NUMTODSINTERVAL

■ Sintaxe



■ interval_unit

- Especifica:
 - Unidade do número n:
 - 'DAY' / 'HOUR' / 'MINUTE' / 'SECOND' -- case insensitive
 - 'day' / 'hour' / 'minute' / 'second'
 - Tipo:
 - CHAR
 - VARCHAR2
 - Espaços no início/fim:
 - Ignorados.
- Resultado
 - Por omissão:
 - Precisão = 9

Função CAST

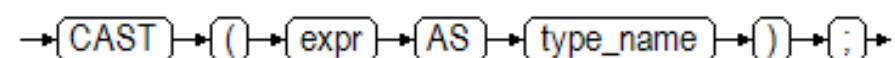
Interesse

- Conversão explícita de tipos de dados.

Sintaxe

CAST(expr AS type_name)

cast_syntax



Semântica

Converte um valor de tipo de dados predefinido num valor de outro tipo de dados predefinido.

- Tipos têm de ser compatíveis (consultar tabela no slide seguinte).

The screenshot shows the Oracle SQL Developer interface. In the top tab bar, 'Worksheet' is selected. Below it, the 'Query Builder' tab is visible. The worksheet contains the following SQL code:

```
SELECT CAST(DATE '2016-11-7' AS TIMESTAMP)  
FROM DUAL;
```

In the 'Query Result' tab, the output is displayed in a table:

	CAST(DATE'2016-11-7'ASTIMESTAMP)
1	16.11.07 00:00:00,000000000

The status bar at the bottom of the result tab indicates: All Rows Fetched: 1 in 0,003 seconds.

Função CAST

Semântica

- Tipos compatíveis

para	de	BINARY_FLOAT BINARY_DOUBLE	CHAR VARCHAR2	NUMBER	Data-Tempo Intervalo
BINARY_FLOAT BINARY_DOUBLE		√	√	√	-
CHAR, VARCHAR2		√	√	√	√
NUMBER		√	√	√	-
Data-Tempo Intervalo		-	√	-	√

Função CAST

Exemplos

The screenshot shows a SQL developer interface. The 'Worksheet' tab is selected, displaying the following SQL code:

```
572
573  SELECT CAST(TO_TIMESTAMP('2016-01-04 08:15','YYYY-MM-DD HH24:MI') AS DATE)
574  FROM DUAL;
575
```

Below the worksheet, the 'Query Result' tab is active, showing the output of the query:

	CAST(TO_TIMESTAMP('2016-01-0408:15','YYYY-MM-DDHH24:MI')ASDATE)
1	16.01.04

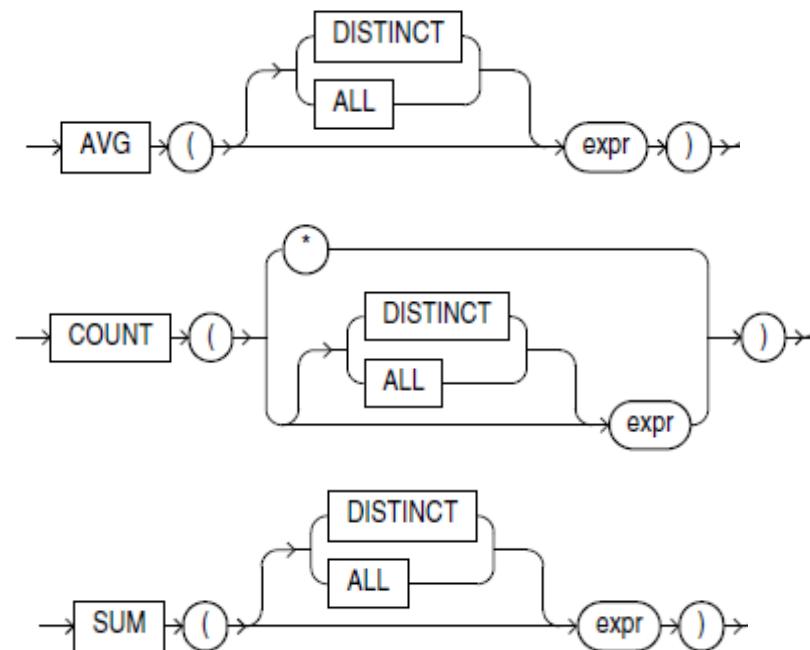
The status bar at the bottom indicates "All Rows Fetched: 1 in 0 seconds".

Funções SQL Agregação

■ Populares

Função	Retorna
AVG	Valor médio
COUNT	Nº de linhas
FIRST	1º valor
LAST	Último valor
MAX	Maior valor
MIN	Menor valor
SUM	Soma

Syntax



Retorna **soma** de valores da expressão especificada (exemplo uma coluna).

O argumento tem de ser de qualquer tipo numérico ou qualquer tipo não-numérico que possa ser convertido implicitamente para um tipo numérico.

Por exemplo, não suporta a soma de coluna de tipo INTERVAL

■ Populares

Função	Exemplo
AVG	Valor médio
COUNT	<code>COUNT(DISTINCT vr.dia_semana)</code> <code>COUNT(1)</code> <code>COUNT(vr.dia_semana)</code>
FIRST	1º valor
LAST	Último valor
MAX	Maior valor
MIN	Menor valor
SUM	Soma

Parâmetros de Inicialização da BD

- Usado para Definir

- *Character Set*
 - Tipos de dados de caracteres -- Exemplo: trocar 7-bit ASCII por Unicode UTF-8
- *Formatos*
 - Datas e tempos -- Exemplo: DD-MON-YY HH24:MM por DD-MON-YY HH:MM:SS
 - Monetários -- Exemplo: trocar € por \$
 - Linguagem natural -- Exemplo: trocar inglês por português

- Parâmetros NLS

- *Categorias:*
 - Cliente -- visualizar: SELECT * FROM **nls_session_parameters**;
 - Instância -- visualizar: SELECT * FROM **nls_instance_parameters**;
 - BD -- visualizar: SELECT * FROM **nls_database_parameters**;

■ Parâmetros NLS de Sessão

- Têm precedência sobre os parâmetros das outras categorias.
- Comando para **modificar**:
 - **ALTER SESSION**
 - Exemplo: `ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MON-YYYY HH24:MI:SS';`

PARAMETER	VALUE
NLS_LANGUAGE	PORTUGUESE
NLS_TERRITORY	PORTUGAL
NLS_CURRENCY	€
NLS_ISO_CURRENCY	PORTUGAL
NLS_NUMERIC_CHARACTERS	,.
NLS_CALENDAR	GREGORIAN
NLS_DATE_FORMAT	DD-MON-YYYY HH:MI:SS
NLS_DATE_LANGUAGE	PORTUGUESE
NLS_SORT	WEST EUROPEAN
NLS_TIME_FORMAT	HH24:MI:SSXFF
NLS_TIMESTAMP_FORMAT	RR.MM.DD HH24:MI:SSXFF
NLS_TIME_TZ_FORMAT	HH24:MI:SSXFF TZR
NLS_TIMESTAMP_TZ_FORMAT	RR.MM.DD HH24:MI:SSXFF TZR
NLS_DUAL_CURRENCY	€
NLS_COMP	BINARY
NLS_LENGTH_SEMANTICS	BYTE
NLS_NCHAR_CONV_EXCP	FALSE

■ Parâmetros NLS de Instância

- Têm precedência sobre os parâmetros da BD.
- Comando para **modificar**:
 - **ALTER SESSION**
 - Exemplo: `ALTER SYSTEM SET NLS_DATE_FORMAT = 'DD-MON-YYYY HH24:MI:SS';`

PARAMETER	VALUE
NLS_LANGUAGE	AMERICAN
NLS_TERRITORY	AMERICA
NLS_SORT	(null)
NLS_DATE_LANGUAGE	(null)
NLS_DATE_FORMAT	(null)
NLS_CURRENCY	(null)
NLS_NUMERIC_CHARACTERS	(null)
NLS_ISO_CURRENCY	(null)
NLS_CALENDAR	(null)
NLS_TIME_FORMAT	(null)
NLS_TIMESTAMP_FORMAT	(null)
NLS_TIME_TZ_FORMAT	(null)
NLS_TIMESTAMP_TZ_FORMAT	(null)
NLS_DUAL_CURRENCY	(null)
NLS_COMP	BINARY
NLS_LENGTH_SEMANTICS	BYTE
NLS_NCHAR_CONV_EXCP	FALSE

- Parâmetros NLS da BD

- Definidos durante a criação da BD.

PARAMETER	VALUE
NLS_LANGUAGE	AMERICAN
NLS_TERRITORY	AMERICA
NLS_CURRENCY	\$
NLS_ISO_CURRENCY	AMERICA
NLS_NUMERIC_CHARACTERS	.,
NLS_CHARACTERSET	AL32UTF8
NLS_CALENDAR	GREGORIAN
NLS_DATE_FORMAT	DD-MON-RR
NLS_DATE_LANGUAGE	AMERICAN
NLS_SORT	BINARY
NLS_TIME_FORMAT	HH.MI.SSXFF AM
NLS_TIMESTAMP_FORMAT	DD-MON-RR HH.MI.SSXFF AM
NLS_TIME_TZ_FORMAT	HH.MI.SSXFF AM TZR
NLS_TIMESTAMP_TZ_FORMAT	DD-MON-RR HH.MI.SSXFF AM TZR
NLS_DUAL_CURRENCY	\$
NLS_COMP	BINARY
NLS_LENGTH_SEMANTICS	BYTE
NLS_NCHAR_CONV_EXCP	FALSE
NLS_NCHAR_CHARACTERSET	AL16UTF16
NLS_RDBMS_VERSION	11.2.0.2.0

Bibliografia

- **Documentação Oracle**

- SQL Data Types
 - https://docs.oracle.com/database/121/SQLRF/sql_elements001.htm#SQLRF0021
- SQL Literals
 - https://docs.oracle.com/database/121/SQLRF/sql_elements003.htm#SQLRF00217
- Format Models
 - https://docs.oracle.com/cd/B28359_01/server.111/b28286/sql_elements004.htm

- **Oracle Tutorial**

- <http://www.techonthenet.com/oracle/index.php>

- **Norma**

- [SQL 2011](#)

- **Livro**

- Mastering Oracle SQL, 2nd Edition

Sanjay Mishra, Alan Beaulieu

O'Reilly Media

February 2009