

SQL - Parte 2

- Última atualização: 15/05/2023



Coalesce

- A Função coalesce aceita uma lista de argumentos e retorna o primeiro valor não nulo.

```
Select nome, coalesce (salario,0)  
From funcionario;
```

Ana 3000
Ricardo 0
Paulo 4500

id_funcionario	nome	salario
1	Ana	3000
2	Ricardo	null
3	Paulo	4500

```
Select coalesce (3,2,null)  
From dual;  
3
```

Obs: Os argumentos devem ser do mesmo tipo. Coalesce faz parte do padrão SQL ANSI-92

NVL

- A Função NVL aceita dois argumentos. Retorna o primeiro argumento não nulo.

```
Select nome, NVL (salario,0)  
From funcionario;
```

Ana 3000
Ricardo 0
Paulo 4500

id_funcionario	nome	salario
1	Ana	3000
2	Ricardo	null
3	Paulo	4500

Obs: Não faz parte do padrão SQL ANSI-92. Esta função só tem no Oracle. Os dois argumentos serão avaliados. Os argumentos até podem ser de tipos diferentes e o Oracle pode fazer a conversão se for possível.

DECODE

```
select nome, decode(id_departamento,1,'Depto 1',2,'Depto 2, ' ' )  
from funcionario;
```



CASE

```
select nome, salario, case
    when salario > 4000 then 'Categoria 3'
    when salario > 2000 then 'Categoria 2'
    else 'Categoria 1'
end
From funcionario;
```

```
select nome, case id_departamento
    when 1 then 'Depto 1'
    when 2 then 'Depto 2'
    else ' '
end
From funcionario;
```



Funções de agregação

- Funções de Grupo: sum, count, avg, min, max, stddev, variance.
 - O count não conta valores nulos.
 - Para as outras funções os cálculos são feitos ignorando os valores nulos.
 - As funções de grupo atuam sobre um conjunto de linhas.

Ex: Apresentar a média de salários dos funcionários.

```
select avg(salario)
from funcionario;
```



Group by

- Cláusula GROUP BY: Agrupa partes do resultado de uma consulta sobre as quais é possível utilizar uma função de agregação.

Ex: Para cada departamento apresentar o código do departamento e a média de salários.

```
select id_departamento,avg(salario)
from funcionario
group by id_departamento;
```



Having

- Cláusula HAVING: Especifica condições para formação de um grupo. Só existe associada a cláusula Group By. As condições só podem envolver os atributos da cláusula group by ou uma função de agregação.
 - Ex: Apresentar o código departamento e a média de salários para os departamentos com mais de 4 funcionários.

```
select id_departamento, avg(salario)
from funcionario
group by id_departamento
having count(*) > 4;
```



Textos e Datas

- Textos e Datas devem ser colocados entre aspas simples;
- A comparação de textos diferencia maiúsculas e minúsculas;
- As datas possuem diferentes formatos



Operações com data

- Data+número: soma o número de dias a data e retorna uma data;
- Data –número: diminui o número de dias da data e retorna uma data;
- Data – data: retorna o número de dias
- Data + número/24: data com um número de horas a mais.
Select sysdate + 2
From dual;



Funções de data

- Sysdate: retorna a data e a hora atual (a data e a hora é obtida do servidor do banco de dados).
- add_months(data,meses): adiciona o número de meses a data especificada.
- Trunc(data): retorna a data com o tempo indicando zero horas.



Formatos de data

YYYY: ano;

YEAR: ano por extenso

MM: mês;

MON: nome do mês com 3 caracteres;

Month: mês por extenso;.

dd:dia,

DAY: dia por extenso;

DY: nome do dia, abreviado com 3 caracteres.



Operações com data

- To_char: converte um número ou data em caracter no formato especificado.

```
Select to_char(sysdate, 'dd/mm/yyyy hh24:mi')
```

```
From dual;
```

To_date: converte uma cadeia de caracteres para data.

```
Select id_funcionario
```

```
From funcionario
```

```
Where data_nascimento=to_date('12/07/1978','dd/mm/yyyy');
```

Obs: hh24: hora do dia no intervalo de 1 a 23. mi:minuto; ss:segundo.



NLS_DATE_FORMAT

```
select *  
from nls_database_parameters p  
where p.parameter='NLS_DATE_FORMAT';
```

```
select *  
from nls_session_parameters p  
where p.parameter='NLS_DATE_FORMAT';
```



Insert data

- Insert into funcionario valuesn(id_funcionario, nome, salario,id_departamento, data_admissao)
values(sfuncionario.nextval, 'Ana',3000,1,to_date('01/07/2021 08:20','dd/mm/yyyy hh24:mi'));
- Insert into funcionario valuesn(id_funcionario, nome, salario,id_departamento, data_admissao)
values(sfuncionario.nextval, 'Ana',3000,1,sysdate);
- Insert into funcionario valuesn(id_funcionario, nome, salario,id_departamento, data_admissao)
values(sfuncionario.nextval, 'Ana',3000,1,'01/07/21');



Comparação de Data

Ex: Recuperar o nome dos funcionários com data_admissao igual a 01/03/2020.

id_funcionario	nome	Data_admissao
1	Ana	01/03/2020 15:00
2	Ricardo	25/03/2020 11:00
3	Paulo	01/03/2021 10:00

Errado:

Select nome

From funcionario

Where data_admissao = '01/03/2020';

Comparação de Data

Ex: Selecionar o nome dos funcionários com data_admissao igual a 01/03/2020.

id_funcionario	nome	Data_admissao
1	Ana	01/03/2020 15:00
2	Ricardo	25/03/2020 11:00
3	Paulo	01/03/2021 10:00

Certo:

Select nome

From funcionario

Where data_admissao >= '01/03/2020'

And data_admissao < '02/03/2020';

Ou

Select nome

From funcionario

Where data_admissao >= '01/03/2020'

And data_admissao < '02/03/2020';