

Banco de Dados

SQL Avançada

SELECT Avançado

- Introdução:

- *SELECT*:

- O comando *SELECT* tem poder superior ao uso da cláusula *WHERE*
 - Sub-comandos e funções podem ser usadas em conjunto com o comando *SELECT*

SELECT Avançado

- Cláusula *WHERE* com Condições Complexas:

- Sintaxe:

- select Atributo [{, Atributo}] from Tabela
[where Condição [{Operador_Lógico Condição}]];

- Operador Lógico “AND”

- Exemplos:

- select ID, Nome from Funcion
where ID_Setor is not null and ID > 1;***

SELECT Avançado

- Cláusula *WHERE* com Condições Complexas: (cont.)

- Operador Lógico “OR”

- Exemplos:

***select ID, Nome from Funcion
where Nome = 'Tadeu' or ID_Setor = 3;***

- Regras de Precedência

Ordem	Regra
1	Expressões entre parênteses “(...)”
2	Todos os operadores de comparação “=,<>,>,>=,<,<=,IN...”
3	Operador lógico “AND”
4	Operador lógico “OR”

SELECT Avançado

- Cláusula *ORDER BY*:

- Sintaxe:

```
select Atributo [ {, Atributo} ] from Tabela  
[ where Condição [ {Operador_Lógico Condição} ] ]  
[ order by Atributo [Modo_Ordenação] [ {, Atributo  
[Modo_Ordenação]} ] ];
```

- Exemplos:

```
select ID, Nome from Funcion  
order by Nome asc; >>> ou <<< order by Nome desc;
```

SELECT Avançado

- Cláusula *JOIN*:

- Sintaxe:

```
select Atributo [ {, Atributo} ] from Tabela  
[ [Cláusula_Junção] join Tabela_Junção on Condição_Junção]  
[ where Condição [ {Operador_Lógico Condição} ] ]  
[ order by Atributo [Modo_Ordenação] [ {, Atributo  
[Modo_Ordenação]} ] ];
```

SELECT Avançado

- Cláusula *JOIN*: (cont.)

- Cláusula *INNER*:

- Exemplos:

***select Funcion.ID, Funcion.Nome, Setor.Setor
from Funcion***

***inner join Setor on (Setor.ID =
Funcion.ID_Setor);***

***select fnc.ID, fnc.Nome, st.Setor from Funcion
fnc***

join Setor st on (st.ID = fnc.ID_Setor);

SELECT Avançado

- Cláusula *JOIN*: (cont.)

- Cláusula *OUTER*:

- Exemplo *LEFT*:

- select a.ID, a.Nome, b.Setor from Funcion a
left outer join Setor b on (b.ID = a.ID_Setor);*

- Exemplo *RIGHT*:

- select a.ID, a.Nome, b.Setor from Funcion a
right outer join Setor b on (b.ID = a.ID_Setor);*

- Exemplo *FULL*: (MySQL não tem)

- select a.ID, a.Nome, b.Setor from Funcion a
full outer join Setor b on (b.ID = a.ID_Setor);*

SELECT Avançado

- Cláusula *UNION*:

- Sintaxe:

```
select Atributo [ {, Atributo} ] from Tabela
[ where Condição [ {Operador_Lógico Condição} ] ]
[ [Cláusula_Junção] join Tabela_Junção on Condição_Junção]
[ union [all]
select Atributo [ {, Atributo} ] from Tabela
[ where Condição [ {Operador_Lógico Condição} ] ]
[ [Cláusula_Junção] join Tabela_Junção on Condição_Junção] ]
[ order by Atributo [Modo_Ordenação] [ {, Atributo
[Modo_Ordenação]} ] ];
```

SELECT Avançado

- Cláusula *UNION*: (cont.)

- Exemplos:

- select Nome from Funcion
 - union
 - select Nome from Cliente;

- select Nome from Funcion
 - union all
 - select Nome from Cliente;

SELECT Avançado

- Funções Básicas:

- Exemplos:

- ```
select max(Salario) as Maior_Salario from Funcion;
```

- ```
select min(Salario) as Menor_Salario from Funcion;
```

- ```
select count(Salario) as Quant_Salario from Funcion;
```

- ```
select sum(Salario) as Soma_Salario from Funcion;
```

- ```
select avg(Salario) as Media from Funcion;
```

# *SELECT* Avançado

- Cláusula *GROUP BY*:

- Sintaxe:

```
select Função(Parâmetro) [{, Função(Parâmetro)}] [{, Atributo}]
 from Tabela
 [where Condição [{Operador_Lógico Condição}]]
 [group by Atributo [{, Atributo}]]
 [order by Atributo [Modo_Ordenação] [{, Atributo
 [Modo_Ordenação]}]];
```

- Exemplo:

```
select sum(salario) as Soma_Salario, ID_Setor from Funcion
group by ID_Setor;
```

# *SELECT* Avançado

- Cláusula *HAVING*:

- Sintaxe:

```
select Função(Parâmetro) [{, Função(Parâmetro)}] [{, Atributo}]
 from Tabela
 [where Condição [{Operador_Lógico Condição}]]
 [group by Atributo [{, Atributo}]]
 [having Condição_Função]
 [order by Atributo [Modo_Ordenação] [{, Atributo
 [Modo_Ordenação]}]];
```

- Exemplo:

```
select sum(salario) as Soma_Salario, ID_Setor from Funcion
group by ID_Setor
having sum(salario) > 1500;
```

# *SELECT* Avançado

- Cláusula *DISTINCT*:

- Sintaxe:

- ```
select distinct(Atributo) [ {, Atributo} ] from Tabela  
[ where Condição [ {Operador_Lógico Condição} ] ]  
[ order by Atributo [Modo_Ordenação] [ {, Atributo  
[Modo_Ordenação]} ] ];
```

- Exemplo:

- ```
select distinct(ID_Setor) from Funcion;
```

# *SELECT* Avançado

- Operadores de Manipulação:

- Sintaxe:

```
select Operação_Manipulação [{, Atributo}] from Tabela
[where Condição [{Operador_Lógico Condição}]]
[order by Atributo [Modo_Ordenação] [{, Atributo
[Modo_Ordenação]}]];
```

- Operador de Manipulação “||”

- Exemplo:

***select Nome || ' ' || Sobrenome as  
Nome\_Completo from Cliente;***

***MySQL: select concat(Nome, ' ', Sobrenome) as  
Nome\_Completo from Cliente;***

# *SELECT* Avançado

- Operadores de Manipulação: (cont.)

- Operador de Manipulação “+”

- Exemplo:

***select Nome, (Salario + 200) as Salario\_Atual  
from Funcion;***

- Operador de Manipulação “-”

- Exemplo:

***select Nome, (Salario - 200) as Salario\_Atual  
from Funcion;***



# *SELECT* Avançado

- Operadores de Manipulação: *(cont.)*

- Operador de Manipulação “\*”

- Exemplo:

***select Nome, (Salario \* 1.5) as Salario\_Atual  
from Funcion;***

- Operador de Manipulação “/”

- Exemplo:

***select Nome, (Salario / 2) as Salario\_Quinzena  
from Funcion;***

# *SELECT* Avançado

- *Nested Queries:*

- Sintaxe:

- ```
select Atributo [ {, Atributo} ] from Tabela  
[ where Atributo Operador_Comparativo  
  [ select Atributo_Ninho from Tabela_Ninho [where  
    Condição_Ninho] ]  
[ order by Atributo [Modo_Ordenação] [ {, Atributo  
  [Modo_Ordenação]} ] ];
```

SELECT Avançado

- *Nested Queries: (cont.)*

- Exemplos:

- select ID, Nome from Cliente
where ID in (select ID_Cliente from Pedido where Valor >= 2000);

- select ID, Nome from Funcion
where ID_Setor = (select max(ID) - 1 from Setor);

- select f.id, f.nome from funcion f
join pedido p on (p.id_funcion = f.id)
where p.id_cliente in (
 select c.id from cliente c
 where c.id = p.id_cliente
 and p.dt_pedido > '2011-02-01'
);

Exercício #1

- Baseado nas tabelas povoadas no Exercício #3 dos slides sobre SQL Básica:
 - a) Mostrar os seguintes dados do pedido:
 - Numero do pedido, nome do cliente, nome do funcionário.
 - b) Mostrar em um único comando o valor total que cada cliente comprou.
 - c) Mostrar em um único comando o total de cliente cadastrado no mesmo estado.
 - d) Mostrar em um único comando o total de funcionário cadastrado no mesmo setor.
 - e) Mostrar os seguintes dados do funcionário:
 - Nome do funcionário, nome do setor.
 - f) Mostrar o nome dos funcionários e valor total das vendas do funcionário que o total da venda tenha sido superior a R\$ 1.500,00.