



INSERT INTO

INSERINDO DADOS NA TABELA CLIENTE

```
insert into CLIENTE (ID_CLI,NOME_CLI,ENDERECO_CLI,CIDADE_CLI,CEP_CLI,UF_CLI,CGC_CLI,IE_CLI)
values (01,'FLAVIO MOTA','RUA X','SAO PAULO','02812260','SP','000055555','66663333');
```

```
insert into CLIENTE (ID_CLI,NOME_CLI,ENDERECO_CLI,CIDADE_CLI,CEP_CLI,UF_CLI,CGC_CLI,IE_CLI)
values (02,'ROCHELE MOTA','RUA X','SAO PAULO','02812260','SP','000044444','666622222');
```

```
insert into CLIENTE (ID_CLI,NOME_CLI,ENDERECO_CLI,CIDADE_CLI,CEP_CLI,UF_CLI,CGC_CLI,IE_CLI)
values (03,'MARIA DE JESUS','RUA Y','RIO DE JANEIRO','01512260','RJ','222255555','77773333');
```



INSERT INTO

INSERINDO DADOS NA TABELA VENDEDOR

```
insert into VENDEDOR (ID_VENDEDOR,NOME_VEND,SALARIO_VEND,COMISSAO)
values (01,'JOAO',1000,10);
```

```
insert into VENDEDOR (ID_VENDEDOR,NOME_VEND,SALARIO_VEND,COMISSAO)
values (02,'PEDRO',1500,15);
```

```
insert into VENDEDOR (ID_VENDEDOR,NOME_VEND,SALARIO_VEND,COMISSAO)
values (03,'LUIZ',800,5);
```



INSERT INTO

INSERINDO DADOS NA TABELA PRODUTO

```
insert into PRODUTO(ID_PRODUTO,UNIDADE,DESCRICAO,VAL_UNIT)
values (01,'PC','TABLET',180);
```

```
insert into PRODUTO(ID_PRODUTO,UNIDADE,DESCRICAO,VAL_UNIT)
values (02,'UN','NOTEBOOK',1800);
```

```
insert into PRODUTO(ID_PRODUTO,UNIDADE,DESCRICAO,VAL_UNIT)
values (03,'UN','MOUSE',15);
```



INSERT INTO

INSERINDO DADOS NA TABELA PEDIDO

```
insert into PEDIDO (NUM_PEDIDO, ID_CLI, ID_VENDEDOR, PRAZO_ENTREGA)
values (001, 01, 03, '2014/03/30');
```

```
insert into PEDIDO (NUM_PEDIDO, ID_CLI, ID_VENDEDOR, PRAZO_ENTREGA)
values (002, 02, 01, '2014/03/05');
```

```
insert into PEDIDO (NUM_PEDIDO, ID_CLI, ID_VENDEDOR, PRAZO_ENTREGA)
values (003, 01, 01, '2014/10/01');
```



INSERT INTO

INSERINDO DADOS NA TABELA ITENS_PEDIDO

```
insert into ITENS_PEDIDO (ID_ITEM_PEDIDO,NUM_PEDIDO,ID_PRODUTO,QUANTIDADE)  
values (001,01,03,5);
```

```
insert into ITENS_PEDIDO (ID_ITEM_PEDIDO,NUM_PEDIDO,ID_PRODUTO,QUANTIDADE)  
values (002,01,02,2);
```

```
insert into ITENS_PEDIDO (ID_ITEM_PEDIDO,NUM_PEDIDO,ID_PRODUTO,QUANTIDADE)  
values (003,02,01,2);
```

O comando SELECT com todos os campos de uma tabela



Selecione Campos Origem Tabela
SELECT * FROM CLIENTE

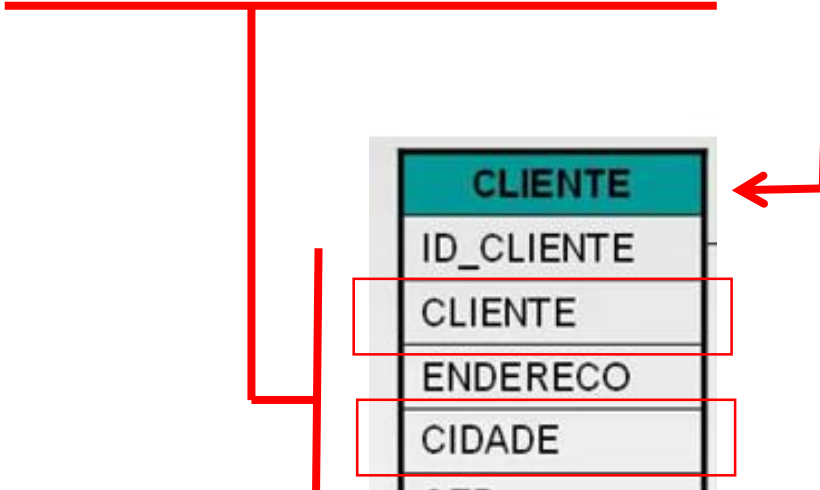
CLIENTE
ID_CLIENTE
CLIENTE
ENDereco
CIDADE
CEP
UF
CGC
IE

Resultado da pesquisa

Results		Messages						
	ID_CLI	NOME_CLI	ENDereco_CLI	CIDADE_CLI	CEP_CLI	UF_CLI	CGC_CLI	IE_CLI
1	1	FLAVIO MOTA	RUA X	SAO PAULO	02812260	SP	0000555555	66663333
2	2	ROCHELE MOTA	RUA X	SAO PAULO	02812260	SP	000044444	666622222
3	3	MARIA DE JESUS	RUA Y	RIO DE JANEIRO	01512260	RJ	2222255555	77773333

O comando SELECT especificando os campos

Selecione Campos Origem Tabela
SELECT NOME_CLI, CIDADE_CLI FROM CLIENTE



CLIENTE
ID_CLIENTE
CLIENTE
ENDERECO
CIDADE
CEP
UF
CGC
IE

Resultado da pesquisa

	NOME_CLI	CIDADE_CLI
1	FLAVIO MOTA	SAO PAULO
2	ROCHELE MOTA	SAO PAULO
3	MARIA DE JESUS	RIO DE JANEIRO

O comando SELECT com ORDER BY ASC E DESC

Selecione Campos Origem Tabela Campo a ser ordenado
SELECT * FROM VENDEDOR ORDER BY NOME_VEND ASC

Resultado da pesquisa

	ID_VENDEDOR	NOME_VEND	SALARIO_VEND	COMISSAO
1	1	JOAO	1000,00	10,00
2	3	LUIZ	800,00	5,00
3	2	PEDRO	1500,00	15,00

Ordem crescente (não é obrigatório)

A consulta será ordenada por essa coluna em ordem crescente.

Selecione Campos Origem Tabela Campo a ser ordenado
SELECT * FROM VENDEDOR ORDER BY NOME_VEND DESC

Resultado da pesquisa

	ID_VENDEDOR	NOME_VEND	SALARIO_VEND	COMISSAO
1	2	PEDRO	1500,00	15,00
2	3	LUIZ	800,00	5,00
3	1	JOAO	1000,00	10,00

Ordem decrescente (é obrigatório)

A consulta será ordenada por essa coluna em ordem decrescente.

O comando SELECT e a cláusula WHERE

Selecione Campos Origem Tabela Onde

SELECT * FROM CLIENTE WHERE ID_CLI = 1

Condição para a busca

Campo Parâmetro

CLIENTE
ID_CLIENTE
CLIENTE
ENDereco
CIDADE
CEP
UF
CGC
IE

Resultado da pesquisa

Results		Messages						
	ID_CLI	NOME_CLI	ENDereco_CLI	CIDADE_CLI	CEP_CLI	UF_CLI	CGC_CLI	IE_CLI
1	1	FLAVIO MOTA	RUA X	SAO PAULO	02812260	SP	0000555555	66663333

O comando SELECT e a cláusula WHERE

Condição para a busca

```
SELECT * FROM CLIENTE WHERE ID_CLI >= 2 AND CIDADE_CLI = 'SAO PAULO'
```

Resultado da pesquisa

	ID_CLI	NOME_CLI	ENDereco_CLI	CIDADE_CLI	CEP_CLI	UF_CLI	CGC_CLI	IE_CLI
1	2	ROCHELE MOTA	RUA X	SAO PAULO	02812260	SP	000044444	666622222

↑
Parâmetro 1

↑
Parâmetro 2

O comando SELECT usando BETWEEN

Campo para o intervalo

```
SELECT * FROM PEDIDO WHERE PRAZO_ENTREGA  
BETWEEN '2014-05-01' AND '2014-05-31'
```

Intervalor

Resultado da pesquisa

	NUM_PEDIDO	ID_CLI	ID_VENDEDOR	PRAZO_ENTREGA
1	2	2	1	2014-05-03
2	3	1	1	2014-05-01

Campo para o intervalo

O comando SELECT usando NOT BETWEEN

Campo para o intervalo

```
SELECT * FROM PEDIDO WHERE PRAZO_ENTREGA  
NOT BETWEEN '2014-05-01' AND '2014-05-31'
```

Intervalor

Resultado da pesquisa

	NUM_PEDIDO	ID_CLI	ID_VENDEDOR	PRAZO_ENTREGA
1	1	1	3	2014-10-03



Campo para o intervalo

O comando SELECT usando o comparador LIKE

Campo para a consulta

```
SELECT * FROM CLIENTE WHERE NOME_CLI LIKE 'F%'
```

Serão filtrados todos os clientes que o nome inicia com a letra 'F'

	ID_CLI	NOME_CLI	ENDERECO_CLI	CIDADE_CLI	CEP_CLI	UF_CLI	CGC_CLI	IE_CLI
1	1	FLAVIO MOTA	RUA X	SAO PAULO	02812260	SP	0000555555	66663333

Resultado da pesquisa

Campo para a consulta

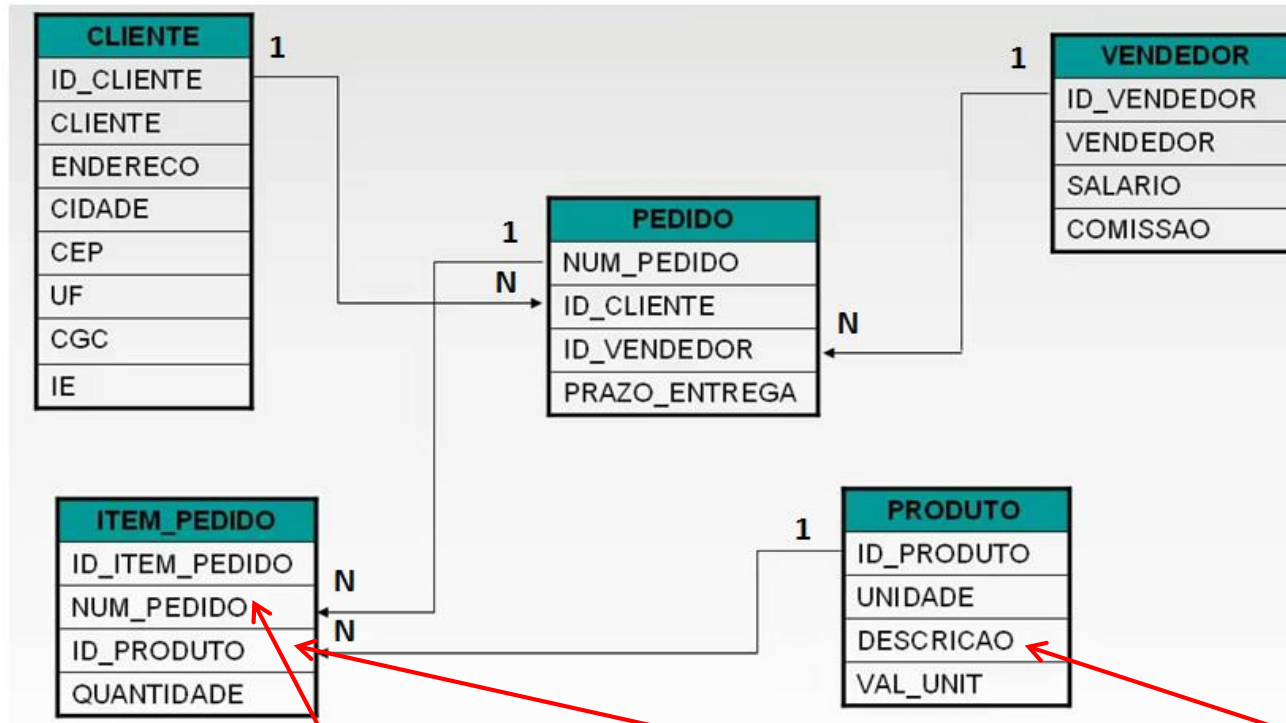
```
SELECT * FROM CLIENTE WHERE NOME_CLI LIKE '%MOTA%'
```

Serão filtrados todos os clientes que tenha em qualquer lugar do seu nome a palavra 'MOTA'

	ID_CLI	NOME_CLI	ENDERECO_CLI	CIDADE_CLI	CEP_CLI	UF_CLI	CGC_CLI	IE_CLI
1	1	FLAVIO MOTA	RUA X	SAO PAULO	02812260	SP	0000555555	66663333
2	2	ROCHELE MOTA	RUA X	SAO PAULO	02812260	SP	0000444444	666622222

Resultado da pesquisa

O comando SELECT usando DUAS TABELAS



```

SELECT ITENS_PEDIDO.NUM_PEDIDO, ITENS_PEDIDO.ID_PRODUTO, PRODUTO.DESCRICAO
FROM ITENS_PEDIDO, PRODUTO
WHERE ITENS_PEDIDO.NUM_PEDIDO = 1 AND ITENS_PEDIDO.ID_PRODUTO = PRODUTO.ID_PRODUTO
  
```

Condição 01

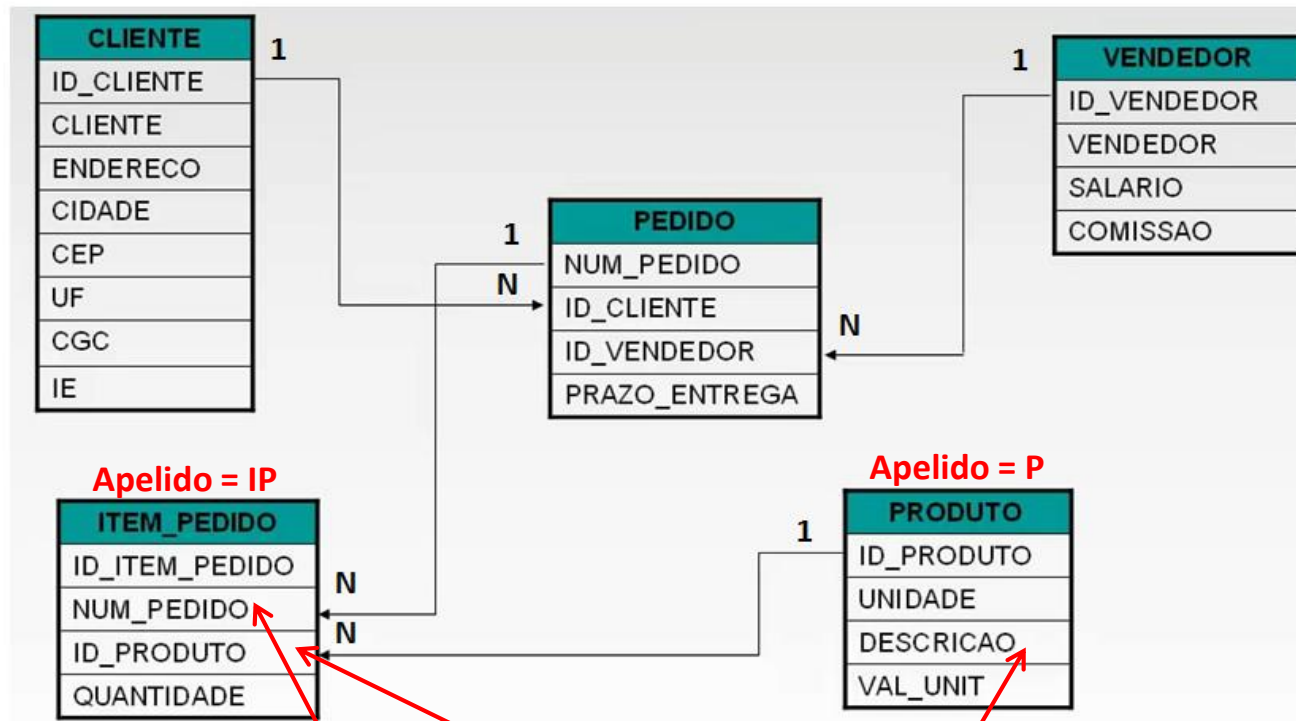
Condição 02 (Junção das tabelas)

	NUM_PEDIDO	ID_PRODUTO	DESCRICAO
1	1	3	MOUSE
2	1	2	NOTEBOOK

Resultado da pesquisa

Flávio Mota

O comando SELECT usando DUAS TABELAS com apelido



```
SELECT IP.NUM_PEDIDO, IP.ID_PRODUTO, P.DESCRICAO
FROM ITENS_PEDIDO IP, PRODUTO P
WHERE IP.NUM_PEDIDO = 1 AND IP.ID_PRODUTO = P.ID_PRODUTO
```

Condição 01 **Condição 02 (Junção das tabelas)**

	NUM_PEDIDO	ID_PRODUTO	DESCRICAO
1	1	3	MOUSE
2	1	2	NOTEBOOK

Resultado da pesquisa

UPDATE

SELECT * FROM CLIENTE;

Result Grid

Filter Rows:

Edit:



Export/Import:




Wrap Cell Content:



ID_CLI	NOME_CLI	ENDereco_CLI	CIDADE_CLI	CEP_CLI	UF_CLI	CGC_CLI	IE_CLI
1	FLAVIO MOTA	RUA X	SAO PAULO	02812260	SP	0000555555	66663333
2	ROCHELE MOTA	RUA X	SAO PAULO	02812260	SP	000044444	666622222
3	MARIA DE JESUS	RUA Y	RIO DE JANEIRO	01512260	RJ	2222255555	77773333
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL


UPDATE CLIENTE SET NOME_CLI='FLAVIO' WHERE ID_CLI = 1;

Result Grid

  Filter Rows:

Edit:   

Export/Import:  

Wrap Cell Content: 

	ID_CLI	NOME_CLI	ENDERECO_CLI	CIDADE_CLI	CEP_CLI	UF_CLI	CGC_CLI	IE_CLI
	1	FLAVIO	RUA X	SAO PAULO	02812260	SP	0000555555	66663333
	2	ROCHELE MOTA	RUA X	SAO PAULO	02812260	SP	000044444	666622222
	3	MARIA DE JESUS	RUA Y	RIO DE JANEIRO	01512260	RJ	2222255555	77773333
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

UPDATE CLIENTE SET NOME_CLI='FLAVIO', ENDERECO_CLI ='RUA DA PAZ' WHERE ID_CLI = 1;

1	FLAVIO	RUA DA PAZ	SAO PAULO	02812260	SP	0000555555	66663333
---	--------	------------	-----------	----------	----	------------	----------

DELETE

SELECT * FROM CLIENTE;

ID_CLI	NOME_CLI	ENDereco_CLI	CIDADE_CLI	CEP_CLI	UF_CLI	CGC_CLI	IE_CLI
1	FLAVIO MOTA	RUA X	SAO PAULO	02812260	SP	0000555555	66663333
2	ROCHELE MOTA	RUA X	SAO PAULO	02812260	SP	000044444	666622222
3	MARIA DE JESUS	RUA Y	RIO DE JANEIRO	01512260	RJ	22222555555	77773333
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

DELETE FROM CLIENTE WHERE ID_CLI = 1;

ID_CLI	NOME_CLI	ENDereco_CLI	CIDADE_CLI	CEP_CLI	UF_CLI	CGC_CLI	IE_CLI
2	ROCHELE MOTA	RUA X	SAO PAULO	02812260	SP	000044444	666622222
3	MARIA DE JESUS	RUA Y	RIO DE JANEIRO	01512260	RJ	22222555555	77773333
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Observação: caso esse cliente tenha um pedido em seu nome não poderá ser excluído, o SGBD preserva a integridade referencial das informações, caso queira excluir esse registro antes deve ser excluído o pedido e os itens deste pedido, assim poderá ser excluído o cliente quando não tiver nenhuma informação relacionada a outra tabela do banco de dados.

- Funções de agregação
 - COUNT: Contagem de registros de uma consulta
 - SUM: Soma de valores
 - AVG: Média de valores
 - MAX: Valor máximo retornado pela consulta
 - MIN: Valor mínimo retornado pela consulta

COUNT

- Contagem de registros de uma consulta
`SELECT COUNT(*) FROM FUNCIONARIOS`
6



NOME	SALARIO	DEPARTAMENTO
Fernando	1250	Recursos Humanos
Guilherme	1550	Jurídico
Mário	1400	Recursos Humanos
Paulo	1230	TI
Haroldo	1330	Marketing
Ricardo	1850	TI

Funções de agregação – SQL DQL

SUM

- Soma de valores

```
SELECT SUM(SALARIO) FROM FUNCIONARIOS
```

8610



NOME	SALARIO	DEPARTAMENTO
Fernando	1250	Recursos Humanos
Guilherme	1550	Jurídico
Mário	1400	Recursos Humanos
Paulo	1230	TI
Haroldo	1330	Marketing
Ricardo	1850	TI

Funções de agregação – SQL DQL

AVG

- Média de valores

```
SELECT AVG(SALARIO) FROM FUNCIONARIOS
```

1435




NOME	SALARIO	DEPARTAMENTO
Fernando	1250	Recursos Humanos
Guilherme	1550	Jurídico
Mário	1400	Recursos Humanos
Paulo	1230	TI
Haroldo	1330	Marketing
Ricardo	1850	TI

Funções de agregação – SQL DQL

MAX

- Valor máximo da consulta
`SELECT MAX(SALARIO) FROM FUNCIONARIOS`
1850



NOME	SALARIO	DEPARTAMENTO
Fernando	1250	Recursos Humanos
Guilherme	1550	Jurídico
Mário	1400	Recursos Humanos
Paulo	1230	TI
Haroldo	1330	Marketing
Ricardo	1850	TI

Funções de agregação – SQL DQL

MIN

- Valor mínimo da consulta
SELECT MIN(SALARIO) FROM FUNCIONARIOS
1230



NOME	SALARIO	DEPARTAMENTO
Fernando	1250	Recursos Humanos
Guilherme	1550	Jurídico
Mário	1400	Recursos Humanos
Paulo	1230	TI
Haroldo	1330	Marketing
Ricardo	1850	TI

Integrando com WHERE

- `SELECT Comando FROM Tabela WHERE Condição`
- `SELECT SUM(SALARIO) FROM FUNCIONARIOS WHERE DEPARTAMENTO = 'Recursos Humanos'`

2650



NOME	SALARIO	DEPARTAMENTO
Fernando	1250	Recursos Humanos
Guilherme	1550	Jurídico
Mário	1400	Recursos Humanos
Paulo	1230	TI
Haroldo	1330	Marketing
Ricardo	1850	TI

Funções de paginação

- Funções de paginação
 - DISTINCT: Seleciona os valores únicos, sem repetição
 - ORDER BY: Ordena o resultado baseado nas colunas informadas
 - LIMIT: Limita o número de resultados retornados
 - OFFSET: Indica quantos registros devem ser avançados
- Combinações são permitidas

DISTINCT

- Seleciona os valores únicos, sem repetição
- ```
SELECT DISTINCT(DEPARTAMENTO) FROM FUNCIONARIOS
```

| Tabela FUNCIONARIOS |         |                  |
|---------------------|---------|------------------|
| NOME                | SALARIO | DEPARTAMENTO     |
| Fernando            | 1250    | Recursos Humanos |
| Guilherme           | 1550    | Jurídico         |
| Mário               | 1400    | Recursos Humanos |
| Paulo               | 1230    | TI               |
| Haroldo             | 1330    | Marketing        |
| Ricardo             | 1850    | TI               |

| Resultado        |
|------------------|
| DEPARTAMENTO     |
| Recursos Humanos |
| Jurídico         |
| TI               |
| Marketing        |

### LIMIT

- Limita o número de registros retornados
- ```
SELECT * FROM FUNCIONARIOS LIMIT 2
```

Tabela FUNCIONARIOS		
NOME	SALARIO	DEPARTAMENTO
Fernando	1250	Recursos Humanos
Guilherme	1550	Jurídico
Mário	1400	Recursos Humanos
Paulo	1230	TI
Haroldo	1330	Marketing
Ricardo	1850	TI

Resultado		
NOME	SALARIO	DEPARTAMENTO
Fernando	1250	Recursos Humanos
Guilherme	1550	Jurídico

OFFSET

- Avança (pula) registros do resultado

SELECT * FROM FUNCIONARIOS OFFSET 2

Tabela FUNCIONARIOS		
NOME	SALARIO	DEPARTAMENTO
Fernando	1250	Recursos Humanos
Guilherme	1550	Jurídico
Mário	1400	Recursos Humanos
Paulo	1230	TI
Haroldo	1330	Marketing
Ricardo	1850	TI

Resultado		
NOME	SALARIO	DEPARTAMENTO
Mário	1400	Recursos Humanos
Paulo	1230	TI
Haroldo	1330	Marketing
Ricardo	1850	TI

Funções de agrupamento

- Funções de agrupamento
 - GROUP BY: Agrupamento de registros por categoria
 - HAVING: Seleção de agrupamento

GROUP BY

- Agrupamento de registros para operações
`SELECT DEPARTAMENTO, AVG(SALARIO) FROM
FUNCIONARIOS GROUP BY DEPARTAMENTO`

Tabela FUNCIONARIOS		
NOME	SALARIO	DEPARTAMENTO
Fernando	1250	Recursos Humanos
Guilherme	1550	Jurídico
Mário	1400	Recursos Humanos
Paulo	1230	TI
Haroldo	1330	Marketing
Ricardo	1850	TI

Resultado	
NOME	SALARIO
Jurídico	1550
Marketing	1330
Recursos Humanos	1325
TI	1540

GROUP BY

- Agrupamento de registros para operações
`SELECT DEPARTAMENTO, AVG(SALARIO) FROM
FUNCIONARIOS GROUP BY DEPARTAMENTO`

Tabela FUNCIONARIOS		
NOME	SALARIO	DEPARTAMENTO
Fernando	1250	Recursos Humanos
Guilherme	1550	Jurídico
Mário	1400	Recursos Humanos
Paulo	1230	TI
Haroldo	1330	Marketing
Ricardo	1850	TI

Resultado	
NOME	SALARIO
Jurídico	1550
Marketing	1330
Recursos Humanos	1325
TI	1540

HAVING

- Filtro de seleção para agrupamento

```
SELECT DEPARTAMENTO, AVG(SALARIO) FROM  
FUNCIONARIOS GROUP BY DEPARTAMENTO HAVING  
AVG(SALARIO) > 1500
```

Tabela FUNCIONARIOS		
NOME	SALARIO	DEPARTAMENTO
Fernando	1250	Recursos Humanos
Guilherme	1550	Jurídico
Mário	1400	Recursos Humanos
Paulo	1230	TI
Haroldo	1330	Marketing
Ricardo	1850	TI

Resultado	
NOME	SALARIO
Jurídico	1550
TI	1540

Subqueries

- Realização de consultas com filtro de seleção baseado em uma lista ou outra seleção
- IN / NOT IN

```
SELECT NOME FROM FUNCIONARIOS WHERE  
DEPARTAMENTO IN ('Marketing', 'TI')
```

Tabela FUNCIONARIOS		
NOME	SALARIO	DEPARTAMENTO
Fernando	1250	Recursos Humanos
Guilherme	1550	Jurídico
Mário	1400	Recursos Humanos
Paulo	1230	TI
Haroldo	1330	Marketing
Ricardo	1850	TI

Resultado
NOME
Paulo
Haroldo
Ricardo

Subqueries

```
SELECT NOME FROM FUNCIONARIOS WHERE  
DEPARTAMENTO IN (SELECT DEPARTAMENTO FROM  
FUNCIONARIOS GROUP BY DEPARTAMENTO HAVING  
AVG(SALARIO) > 1500)
```

Tabela FUNCIONARIOS		
NOME	SALARIO	DEPARTAMENTO
Fernando	1250	Recursos Humanos
Guilherme	1550	Juridico
Mário	1400	Recursos Humanos
Paulo	1230	TI
Haroldo	1330	Marketing
Ricardo	1850	TI

Resultado Query In	
NOME	SALARIO
Jurídico	1550
TI	1540

Resultado Final
NOME
Guilherme
Paulo
Ricardo

Exercícios

Nesse exercício vamos executar algumas consultas no banco de dados exemplo SAKILA disponível no site do MySQL. Esse banco de dados é uma modelagem de uma locadora de filmes conforme estrutura abaixo.



- Consultar todos os clientes
- Consultar todos os clientes onde o cliente_id seja maior que 100
- Consultar o nome de todas as cidades onde o pais_id seja = 10
- Consultar a quantidade de alugueis feitos
- Consultar quantas locações o cliente com o código 14 fez
- Consultar o nome do cliente que fez a locação numero 1000
- Listar, sem repetição, os anos que houveram lançamento de filme
- Alterar o ano de lançamento igual 2007 para filmes que iniciem com a Letra "B"
- Listar a quantidade de filmes que estejam classificados como "PG-13" e preço da locação maior que 2.40.
- Consultar Quais as cidades e seu pais correspondente que pertencem a um país que inicie com a Letra "E"