

Consultando Dados

```
1 CREATE TABLE cliente (  
2     codigo SERIAL PRIMARY KEY,  
3     nome VARCHAR(200) NOT NULL,  
4     idade INTEGER NOT NULL,  
5     rg INTEGER NOT NULL,  
6     cpf INTEGER,  
7     Telefone VARCHAR(15),  
8     Rua VARCHAR(30),  
9     Numero INTEGER,  
10    CEP INTEGER DEFAULT 79200000,  
11    Cidade VARCHAR(20) DEFAULT 'Brasilia',  
12    Estado VARCHAR(2) DEFAULT 'DF'  
13 );  
14  
15 CREATE TABLE automovel(  
16     placa VARCHAR(8) PRIMARY KEY,  
17     renavan INTEGER,  
18     anofab INTEGER,  
19     fabricante VARCHAR(10),  
20     modelo VARCHAR(25),  
21     codCliente integer references cliente(codigo) NOT NULL  
22 );
```

```
24 CREATE TABLE ocorrencia (  
25     numero SERIAL PRIMARY KEY,  
26     local VARCHAR(50),  
27     descricao VARCHAR(200),  
28     data date,  
29     autoPlaca VARCHAR(8) references automovel(placa) NOT NULL  
30 );  
31  
32
```

Insert Seguradora

```
33 INSERT INTO cliente(nome, idade, rg) VALUES
34 ('Bruna', 25, 20700),
35 ('anderson', 19, 130000),
36 ('Caio', 32, 1200),
37 ('Andre', 30 1036000),
38 ('Rafael', 50, 6000000),
39 ('Lucas', 60, 90000);
40
41 INSERT INTO automovel(placa, renavan, anofab, fabricante,
42     modelo, codCliente) VALUES
43 ('aaa1111', 1000000, 2016, 'fiat', 'palio',1),
44 ('aaa2222', 2000000, 2019,'fiat', 'bravo', 1),
45 ('aaa3333', 5000000, 2013,'Ford', 'focus', 2),
46 ('aaa4444', 300000, 2012,'Ford', 'ka',3),
47 ('aaa5555', 3000, 2018,'fiat', 'punto',4);
48
49 INSERT INTO ocorrencia (descricao, autoPlaca) VALUES
50 ('colisão lateral na 410 Norte', 'aaa3333'),
51 ('pneu furado', 'aaa4444'),
52 ('bateria arriada, 216 norte', 'aaa2222'),
53 ('guincho para oficina, carro na liga', 'aaa3333');
```

Consultando Dados

- Com certeza a tarefa mais interessante a se fazer em um banco de dados é consultar e manipular as informações que estão armazenadas.
- O comando utilizado para a busca de informações é o mesmo para quaisquer consultas, tendo muitas opções disponíveis que podem variar dependendo do nível de detalhamento requerido.
- Consultas aos dados não os modificam

Consultando Dados

- Comando: SELECT

- Síntaxe:

SELECT [tabela1.]campo1, [tabela2.]campo2
FROM tabela1 [, tabela2] [, ...]
[WHERE...]
[GROUP BY...]
[ORDER BY...]
[ETC...]

Obs: o símbolo [] indica que a informação não é obrigatória

Comando Select

- Consultar todos os campos de todos os registros da tabela clientes (predicado *)

SELECT *

FROM Clientes

- Consultar campos específicos de todos os registros da tabela clientes (tem que listar os campos desejados)

SELECT CodCliente, NomeCliente, EndCliente

FROM Clientes

Ordenando Dados

- **ORDER BY** - Consulta de todos os campos de todos os registros da tabela clientes ordenados alfabeticamente pelo nome

SELECT *

FROM CLIENTES

ORDER BY NomeCliente

Ordenando mais de um Dados

- Consulta de campos específicos de todos os registros da tabela clientes ordenados alfabeticamente pelo nome e em seguida pela idade.

```
SELECT CodCliente, NomeCliente, Idade  
FROM CLIENTES  
ORDER BY NomeCliente, Idade
```

Exercício

1. Listar os nomes dos Clientes
2. Listar os nomes dos Clientes em ordem
3. Listar nome e Cidade dos Cliente ordenando por nome
4. Listar todos os carros (Modelo e placa) ordenando por modelo
5. Listar todas as ocorrências cadastradas

Filtrando Dados

- Consulta de todos os campos dos clientes que moram no estado de São Paulo (uso de cláusula *Where* para filtro)

SELECT *

FROM CLIENTES

WHERE Estado = 'SP'

Filtrando Dados

- Consulta de campos específicos dos clientes que moram no estado de São Paulo (uso de cláusula *Where* para filtro)

```
SELECT CodCliente, NomeCliente, EndCliente  
FROM Clientes  
WHERE Estado = 'SP'
```

Obs: além do operador = podemos usar os operadores <>, >, <, >=, <=

Filtrando e Ordenando Dados

- Consulta de todos os campos dos clientes que moram no estado de São Paulo (uso de cláusula *Where* para filtro) ordenados pelo nome do cliente

SELECT *

FROM CLIENTES

WHERE Estado = 'SP'

ORDER BY NomeCliente

Filtrando e Ordenando Dados

- Consulta de campos específicos dos clientes que moram no estado de São Paulo (uso de cláusula *Where* para filtro) ordenados pelo nome do cliente

SELECT *CodCliente, NomeCliente, EnderecoCliente*

FROM *CLIENTES*

WHERE *Estado = 'SP'*

ORDER BY *NomeCliente*

Exercício

1. Listar nome e RG dos clientes que moram Brasilia
2. Listar modelo e placa dos carros da fabricante fiat ordenando placa
3. Listar Clientes com mais de 40 anos
4. Listar Clientes com menos de 30 anos

Condições

- Usando condições para distinguir dados. Uma **condição** é parte de uma consulta que é usada para exibir informações seletivas, conforme especificadas pelo usuário.
- A cláusula **WHERE** pode conter mais de uma condição e, nesse caso as condições são unidas por operadores **AND** e **OR**.

Exemplo: Encontre as descrições e os valores unitários do produtos cujo valor unitário > 1.00.

```
SELECT descricao  
FROM produto  
WHERE val_unit >= 1.00;
```


Mais Exemplos

```
SELECT nome_aluno, data_nasc_aluno  
FROM alu_aluno  
WHERE data_nasc_aluno >= '30/01/1980';
```

```
SELECT *  
FROM pedido  
WHERE codigo_vendedor = 101;
```

```
SELECT cod_instrut, nome_instrut  
FROM alu_instrutor  
WHERE data_admissao_instrut >= '15/01/1997';
```

Operador AND

- O operador AND aceita várias condições em uma cláusula WHERE.
 - Para que a instrução SQL execute uma ação, todas as condições separadas por AND devem ser TRUE.
- Exemplo: Encontre os nomes dos produtos cujo valor unitário esteja entre 0.50 e 2.00

SELECT descricao

FROM produto

WHERE val_unit >= 0.50

AND val_unit <= 2.00 ;

Mais Exemplos - AND

```
SELECT nome_aluno, tel_aluno  
FROM alu_aluno  
WHERE data_nasc_aluno >= '24/02/1981'  
AND cidade_aluno = 'Campinas'
```

```
SELECT nome_cliente  
FROM cliente  
WHERE cidade = 'Campinas'  
AND cidade = 'São Paulo';
```

Exercício

1. Listar nome e RG dos clientes que moram Brasília e tem mais de 30 anos
2. Listar Clientes com idade entre 30 e 40 anos
3. Listar modelo dos carros fabricados a partir de 2017 da marca fiat
4. Monte uma consulta para o banco seguradora usando com duas condições usando o operador AND

Operador OR

- É usado para combinar várias condições em uma cláusula WHERE de uma instrução SQL.
 - Para que a instrução SQL execute uma ação, pelo menos uma das condições separadas por OR deve ser TRUE.

Exemplo: Encontre todos os dados dos vendedores cujo salário seja 2780.00 ou 4600.00.

- ***SELECT*** *codigo_vendedor, nome_vendedor, salario_fixo, faixa_comissao*
- ***FROM*** *vendedor*
- ***WHERE*** *salario_fixo = 2780*
- ***OR*** *salario_fixo = 4600;*

Exemplo

- Encontre a descrição, unidade e o valor unitário dos produtos de unidade 'M' e os valores unitários sejam 0.11 ou 1.8 ou 2.

```
SELECT unidade, descricao, val_unit  
FROM produto  
WHERE unidade = 'M'  
AND (val_unit = 0.11  
OR val_unit = 1.8  
OR val_unit = 2);
```

Exercício

1. Listar nome e idade dos clientes que moram em Brasília e tem 60 ou 19 anos
2. Monte uma consulta para o banco seguradora usando a o operador AND e o operador OR
3. Monte uma consulta para o banco seguradora usando a o operador OR

Distinguindo Dados

- Comando DISTINCT: Elimina linha repetidas

```
SELECT DISTINCT nome_cliente  
FROM cliente;
```

- **Utilização:** Quando há dados repetidos em uma base de dados
- **Observação:** Pode se distinguir uma pesquisa em mais de um campo. Ex: Nome e Endereço.

Operador Aritméticos

- São usados para executar funções matemáticas em SQL.
- Existem 4 operadores aritméticos:
 - + (adição), - (subtração), * (multiplicação), / (divisão)

Exemplo:

```
SELECT salario_fixo * 1.1  
FROM vendedor;
```

Operador Aritméticos

- *Simule um aumento de 25% nos produtos .*

```
SELECT descricao, unidade, val_unit as "Preço Atual",  
val_unit*1.25 as "Preço com Aumento"  
FROM produto;
```

- *Simule um desconto de 12% nos produtos cuja unidade seja 'M'.*

```
SELECT descricao, unidade, val_unit as "Preço Atual",  
val_unit - (val_unit*0.12) as "Preço com Desconto"  
FROM produto  
WHERE unidade = 'M';
```

Operador Between

- Propicia a pesquisa por uma determinada coluna selecionando as linhas cujo valor da coluna esteja dentro de uma faixa.

Exemplo:

```
SELECT *  
FROM vendedor  
WHERE salario_fixo BETWEEN 2000 AND 3000;
```

Exemplo:

```
SELECT *  
FROM vendedor  
WHERE salario_fixo BETWEEN 3000 AND 2000;
```

Exemplos Between

- Listar o código, descrição e o valor unitário dos produtos que tenham o valor unitário na faixa de R\$ 0.32 até R\$ 2.00.

```
SELECT codigo_produto, descricao, val_unit  
FROM produto  
WHERE val_unit BETWEEN 0.32 AND 2;
```

- Encontre o RA, nome e data de nascimento dos alunos nascidos entre 01/02/1980 a 30/10/1990

```
SELECT matricula, nome_aluno, data_nasc_aluno  
FROM alu_aluno  
WHERE data_nasc_aluno BETWEEN '01/02/1980' AND '30/10/1990';
```

Exercício

1. Criar BD (mínimo 4 tabelas)
2. Inserir dados (mínimo 6 entradas/tabela)
3. Elaborar 20 consultas com base no que já aprendemos.
4. Entregar SQL pelo NEAD