JCAVI
TREINAMENTOS EM TI

Uso dos Comandos Select

O Select é a forma de trazermos ou filtramos as informações do banco de dados. Onde podemos inserir condições, comparações e funções.

Order By: Ordenar em ordem alfabética.

```
select Nome, Sobrenome from cadastro
order by Nome;
```

Where: Colocando uma condição de consulta.

```
select Nome, Sobrenome from cadastro
where Nome = "João";

select Nome, Sobrenome from cadastro
where Nome != "João";

select Nome, Sobrenome from cadastro
where id_cad > 3;

select Nome,Sobrenome from cadastro
where Nome like "Jo%";
```

Consultas Lógicas

OR,AND,NOT

```
select Nome,Sobrenome from cadastro
where Nome = "Roberto" or Sobrenome = "Silva";

select Nome,Sobrenome from cadastro
where
Nome = "Roberto" and Sobrenome = "Ferreira";

select Nome,Sobrenome from cadastro
where not Nome  = "Roberto";
```

Sub Consultas. IN, NOT IN
Sub consulta e um jeito ou uma alternativa de fazer o JOIN das informações entre tabelas.

```
select id_cad,Nome,Sobrenome from cadastro
Where id_cad  in  (1,2);

select id_cad,Nome,Sobrenome from cadastro
Where id_cad not in  (1,2);

select Nome,Sobrenome from cadastro
Where Nome not in (select nome from cadastro
                   where Nome = 'Jose');

select Nome,Sobrenome from cadastro
Where Nome  in (select nome from cadastro
                   where Nome = 'Jose');

select Nome,Sobrenome from cadastro
Where not exists (select nome from cadastro
                   where Nome = 'Roberto');
```

Distinct
Filtra registros repetidos em uma tabela e exibe somente o valor único.

```
select distinct Sobrenome from cadastro
```

JCAVI
TREINAMENTOS EM TI

Chave Estrangeira (ForeinKey).

A chave estrangeira é conhecida por criar o relacionamento entre as tabelas dos bancos de dados. Ela é chamada no modelo físico como constraints.

```
create table produtos(
  id_prod INT NOT NULL AUTO_INCREMENT,
  Nome VARCHAR(100) NOT NULL,
  Valor float(10,2) NOT NULL,
  Quantidade int(11) NOT NULL,
  id_cad int NOT NULL,
  PRIMARY KEY ( id_prod )
);

alter table produtos add constraint fk_cadastro_produtos
foreign key(id_cad)
references cadastro(id_cad)
on delete no action
on update no action;
```

Subconsultas

Utilizando sub consultas para JOIN de tabelas.

```
select Nome, Sobrenome from cadastro
where id_cad in ( select id_cad from produtos where id_cad);


select Nome, Sobrenome from cadastro
where id_cad not in ( select id_cad from produtos where id_cad);

select Nome,
(select Nome from cadastro where id_cad = produtos.id_cad)
from produtos
```

Criando Alias
Os alias geram nomes amigáveis para os selects criados através de consultas e subconsultas.

```
select Nome,
(select Nome from cadastro where id_cad = produtos.id_cad) Compra_Clientes
from produtos
```

JOINS ou Junções

Uma cláusula **join** correspondente a uma operação de junção relacional combina colunas de uma ou mais tabelas em um banco de dados relacional. Ela cria um conjunto que pode ser salvo como uma tabela ou usado da forma como está. Um JOIN é um meio de combinar colunas de uma (auto-junção) ou mais tabelas, usando valores comuns a cada uma delas.

Select sem Join

```
select Quantidade,Sobrenome  Prod_Vendidos
from produtos, cadastro
where cadastro.id_cad = produtos.id_cad
order by Quantidade;
```

Alterar nome de uma coluna

```
ALTER TABLE cadastro
Change  Nome  Nome_Cliente varchar(100);
```

Dar o Select nas tabelas com o nome dos clientes.

```
select Quantidade,Nome_Cliente  Prod_Vendidos
from produtos, cadastro
where cadastro.id_cad = produtos.id_cad
order by Quantidade;
```

Criando Join entre as tabelas sem filtros

```
select * from produtos
JOIN cadastro
```

Criando JOIN ou INNER JOIN relacionando a tabela Cadastro e Produtos

```
select Nome, Nome_Cliente from produtos
JOIN cadastro  on
produtos.id_cad = cadastro.id_cad
order by Nome_cliente ;
```

Temos também outros tipos de JOIN como LEFT JOIN, RIGHT JOIN

Mas primeiro vamos inserir mais 2 cadastros de clientes sem compras.

```
insert into cadastro( Nome_cliente,Sobrenome,CPF) Values('Juliano','silva','13131313131');
insert into cadastro( Nome_cliente,Sobrenome,CPF) Values('Mario','Cunha','14141414141');
```

LEFT JOIN

Quando executa o LEFT JOIN, o SQL irá comparar os valores da esquerda com os da direita. Seria como falar, compare os valores de “produtos” com os valores de “dados_produtos”.

Neste caso, o SQL irá analisar os valores da tabela da esquerda (produtos) e inserir na consulta. Os valores da direita serão apresentados apenas se coincidirem na consulta.

```
select  * from cadastro

 left join produtos  on

produtos.id_cad = cadastro.id_cad
```

RIGHT JOIN

RIGHT JOIN é o contrário de LEFT JOIN, ou seja, o SQL irá analisar os valores da direita primeiro e comparar com os valores da esquerda. Isso quer dizer que todos os valores da tabela da esquerda serão apresentados, e, para os valores da direita, apenas os que coincidirem com a consulta.

```
select  * from produtos
 right join cadastro  on
produtos.id_cad = cadastro.id_cad
```