

```
$shop = $request->user()->shop()->where('id', $id)->get()->first();

if($request->hasFile( key: 'image')) {
    $file = $request->file( key: 'image');

    $fileName = $request->file( key: 'image')->getClientOriginalName();

    $extensionArr = explode( delimiter: '.', $fileName);

    $fileName = $id . '.' . $extensionArr[1];

    Storage::delete([
        $id . '.jpg',
        $id . '.png',
        $id . '.gif',
        $id . '.svg',
        $id . '.jpeg',
    ]);
}
```



My SQL

Todo banco de dados falando de maneira fácil de entender é um navio que dentro desse navio apresenta containers, que dentro desse containers apresenta alguma características específicas, e tentando usar ao máximo características de cada tipo primitivos

Tabelas

Para criar uma banco de dados usamos o seguinte código `Create database cadastro;`, normalmente sempre iremos trabalhar com elas, e para ter uma tabela e preciso esses passos **banco de dados—>tabela—>registro—>campos.**

Caso eu queria mostrar a tabela eu preciso do seguinte código `select * from (nome da tabela)` esse `*` fala que é para puxar a tabela toda, mas caso eu queria só uma parte dela eu vou colocar o nome da parte que eu quero.

Além disso usamos os seguinte código na hora de criar a tabela para deixar na forma padrão da língua portuguesa. Com acentuações, vírgulas e coisas que apresenta apenas na língua portuguesa.

```
1      create database cadastro
2      default character set utf8
3      default collate utf8_general_ci;
```

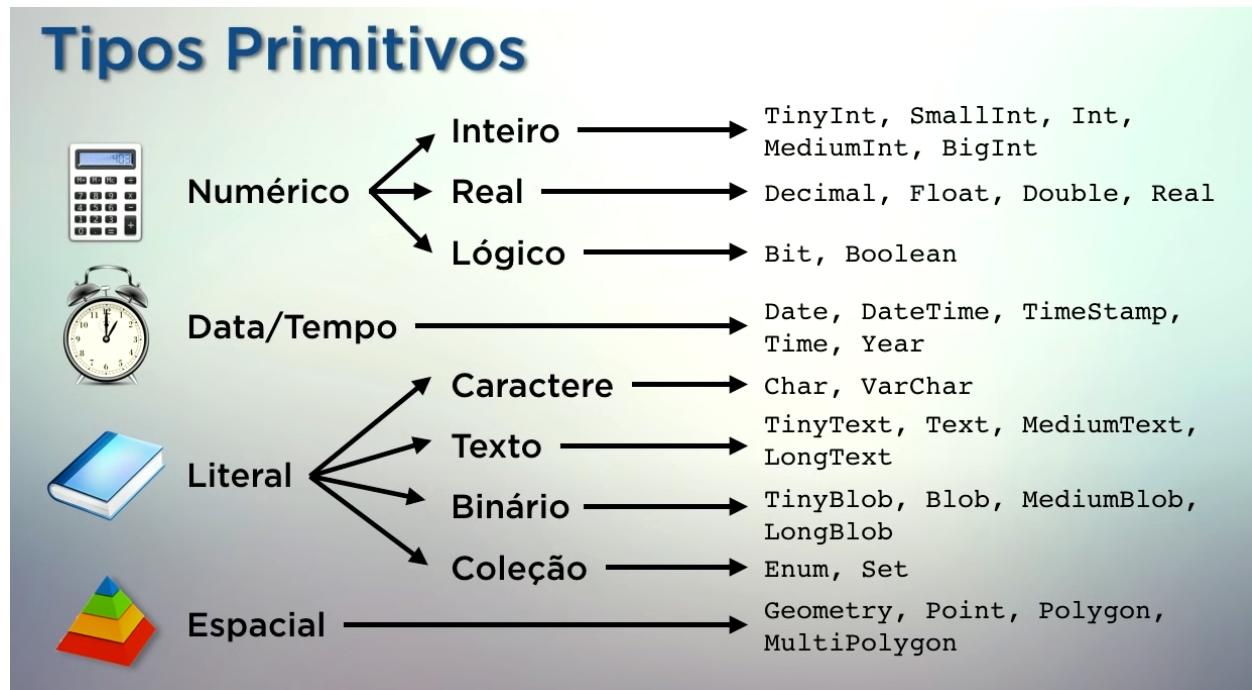
Para apagar um **tabela** ou **banco de dados** usamos o seguinte código.

```
Drop database (nome_do_banco_de_dados)
Drop table (nome_da_Tabela)
```

Caso eu queria inserir algo dentro da minha tabela vou usar os seguinte código, e `Insert Into (tabela) (nome das colunas)` determinei a onde vai ser inserido depois vou colocar os valores `valeus(valores)` segue o exemplo abaixo.

```
INSERT INTO departamento (codigo, telefone)
VALUES(103, '(14) 3432 8070');
```

Para criar uma tabela usamos o seguinte código `create table (titulo da table){};`, dentro da table e preciso determinados registro, para esses registro usamos as seguinte tipos primitivos.



Quando vamos falamos sobre colum estamos falando dos campos no meu MySQL, então sempre que falarem para adicionar uma nova colum é na verdade um novo campo.

Tipo primitivos Mais Usados

Todos são importantes mais tem alguns que utilizamos mais, primeiro vamos falar sobre os Tipo Primitivos Numérico e suas Ramificações.

Numéricos Inteiros

Dos inteiros temos alguns mais usados como `TinyInt`, `Int` e o `BigInt` que pelo o nomes deles já é auto explicativo, o `TinyInt` é usado quando sabemos que os número que o usuário irá digitar será pequeno, assim usamos o `TinyInt`.

Agora para número inteiro usamos o `Int` normal.

E o `BigInt` é usado para número grande, assim irá armazenar forma melhor o Bits dentro do banco de dados.

Numéricos Real

Um dos principais é `Decimal` na qual logo após colocar o código, inserimos dois valores numéricos na qual o primeiro sera o número de casas que o usuário irá poder inserir e na outra quanto números para ter um vírgula, segue o exemplo abaixo.

`Decimal(5,2)`

1	0	2	,	3	5
---	---	---	---	---	---

Data/Tempo

Esse usamos normalmente quando vamos se referir a tempo, todos os códigos são referente ao tempo, mas cada um com o seu diferencial, para saber o que cada um faz basta traduzir ele do inglês para português.

A data na maioria dos bancos de dados vem no formato do EUA então primeiro vem o ano e logo em seguida vem as outras partes

Literal Caractere

O `VARCHAR` guarda somente as quantidades de caracteres apresentado pelo o usuário, exemplo, caso o usuário chame-se de Zé, o nome dele apresenta apenas dois caractere assim salva apenas esse dois bits, mas se caso eu usasse o `CHAR` normal ele iria armazenar o nome mais o espaços em brancos assim ocupando mais bits.

Literal Texto

Leva o mesmo conceito do `INT` mas agora com texto, mas tem uma diferença do caractere e do texto. Normalmente Texto e para texto longos como, descrição de pessoas, guarda um texto longo no meu banco de dados, agora os `CHAR` são para Nome, endereço, numero, uma coisa caractere.

Literal Coleção

Apresenta o `enum` e `set`, que vai da limite na hora do cadastro assim só aceitando determinadas coisas, exemplo.

```
create database banco_de_dados;
default character set utf8
default collate utf8_gebreal_ci;

create table tabela(
    sexo null('H','F')
);
```

Auxiliares Dos Tipos Primitivos

Normalmente toda a table irá ter uma `primary key` que nada mais é que uma chave primaria para cada usuário, para ter duas pessoas iguais no banco de dados, pois nome parecido pode haver, mas nunca vai haver duas `primary key` um bom exemplo disso é o CPF só existe um por pessoa e faz a pessoa ser única.

```
create table teste (
    cpf int(9) not null,
    nome varchar(30),
    nacionalidade varchar(30) default 'Brasil',
    primary key(cpf)
);
```

Not Null

Já o `Not Null` apresenta que está parte não pode ficar sem ser inserido algo nela, então ela não pode ficar vazio.

Default na parte da column

Na parte `nacionalidade varchar(20) DEFAULT 'Brasil'` caso se ele não coloque nada, automaticamente ele vai ficar brasil.

Unique and Unsigned

Nada mas é que ele não vai deixar ser criado outra coisa com o mesmo nome, caso eu coloque o nome “curso em java para iniciantes” não terá outro, pois o `unique` não vai deixar

Caso eu coloque o `UNSIGNED` a pessoa não poderá inserir um número negativo.

O código `constraint` que acompanha alguma `key` serve para declarar ela como nome da key, caso alguma vez eu quiser puxar alguma coisa de um determinada tabela para outra, já irá ficar outras 500.

Auto_increment o que é

O `auto_increment` é uma propriedade de coluna no MySQL que permite que um valor único seja gerado automaticamente para cada nova linha inserida em uma tabela. Esse valor é incrementado automaticamente a cada nova linha, geralmente começando em 1 e aumentando em 1 para cada nova linha adicionada.

Por exemplo, se você tiver uma tabela de funcionários e quiser atribuir um ID exclusivo para cada novo funcionário adicionado, você pode definir a coluna `ID` como `auto_increment`. Dessa forma, cada vez que você adicionar um novo funcionário à tabela, o MySQL irá automaticamente atribuir um novo ID exclusivo para essa linha.

A propriedade `auto_increment` é frequentemente usada para criar chaves primárias únicas em uma tabela, o que é útil para garantir que não haja duplicação de dados ou conflitos entre as linhas da tabela.

Default charset

No final de uma table precisamos colocar a seguinte informações para demonstrar que ela é feita em pt-br, usamos o seguinte código no final, `default charset = utf8` que ficara assim.

```
)DEFAULT CHARSET = utf8;
```

Inserir Coisas Na Table

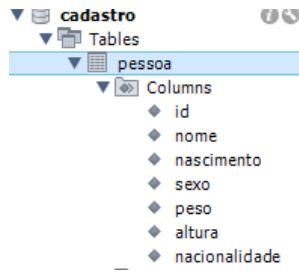
Para inserir é muito fácil apenas precisamos dar o comando `Insert Into nome da table` e depois dar o nome dos nossos itens que serão utilizados e dar um `values` com o nome dos dados, segue o exemplo abaixo.

```
insert into pessoa
(nome,nascimento,sexo,peso,altura,nacionalidade)
values
('maria','2004-06-12','F','68.5','1.53', 'Portugal');
```

Caso eu veja que minha ordem da table é o mesmo que na qual eu quero inserir, não precisamos dar o nome dos itens da nossa table, só precisamos juntar o `insert Into nome da table values` segue o exemplo.

```
insert into pessoa values
(default,'mario','2002-08-28','M','99.0','2.15','Brasil');
```

Mas só se essa parte estiver igual a ordem que eu desejo inserir, que nada mais é que a ordem que eu criar na table. E vemos também que eu defini um `default` logo no inicio, pois eu preciso dar todos as informações para ser inserido na table pelo o método simplificado.



Posso adicionar várias pessoa de uma vez, mas para cada final vou colocar uma virgula e no ultimo vou colocar um ponto e virgula.

Alterações De Table

Para mudar algo na table usamos o seguinte código `alter table (nome da table)`, agora se eu quiser adicionar algo na minha table irá ficar com o seguinte código `add column (o que eu vou add e tipo primitvo dele)` segue o exemplo abaixo.

```
ALTER TABLE pessoas
ADD COLUMN profissao varchar(10);
```

Mostrar a Desc da Table

Caso eu queria ver como está a nossa table basta eu dar o seguinte comando `desc (nome da table)` assim ele mostrará tudo o que está dentro dela. Como o exemplo abaixo.

	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	auto_increment
	nome	varchar(30)	NO		NULL	
	nascimento	date	YES		NULL	
	sexo	enum('M','F')	YES		NULL	
	peso	decimal(5,2)	YES		NULL	
	altura	decimal(3,2)	YES		NULL	
	nacionalidade	varchar(30)	YES		Brasil	
	profissao	varchar(10)	YES		NULL	

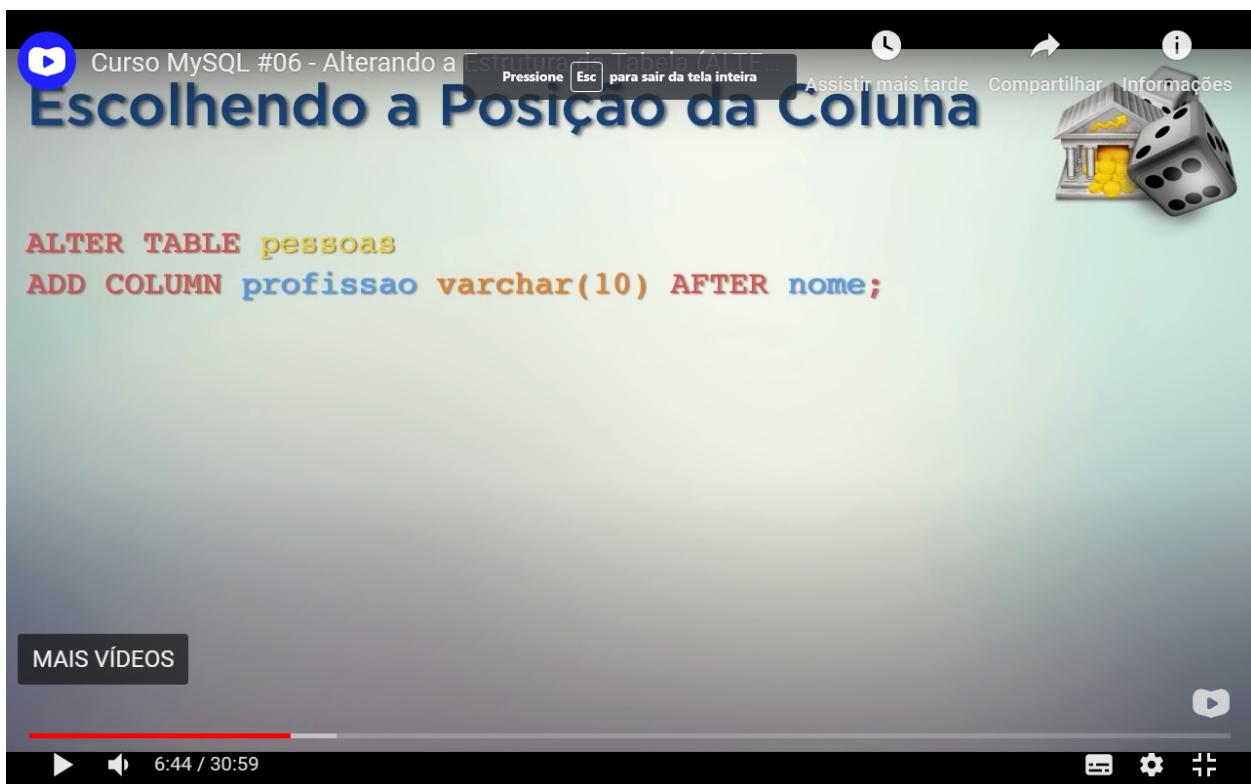
Remover coluna

Caso eu queria remover alguma column basta eu fazer o seguinte o código `alter table(nome da tabela) drop column (e o nome da column)`

```
alter table pessoa  
drop column profissao;
```

Add uma column

Caso eu queria add de um column em uma determinada posição vou fazer seguinte, `alter table pessoa` na linha debaixo `add column (nome da nova column) (tipo primitivo) after (nome da column que vai add depois ou antes dela)` como no exemplo abaixo

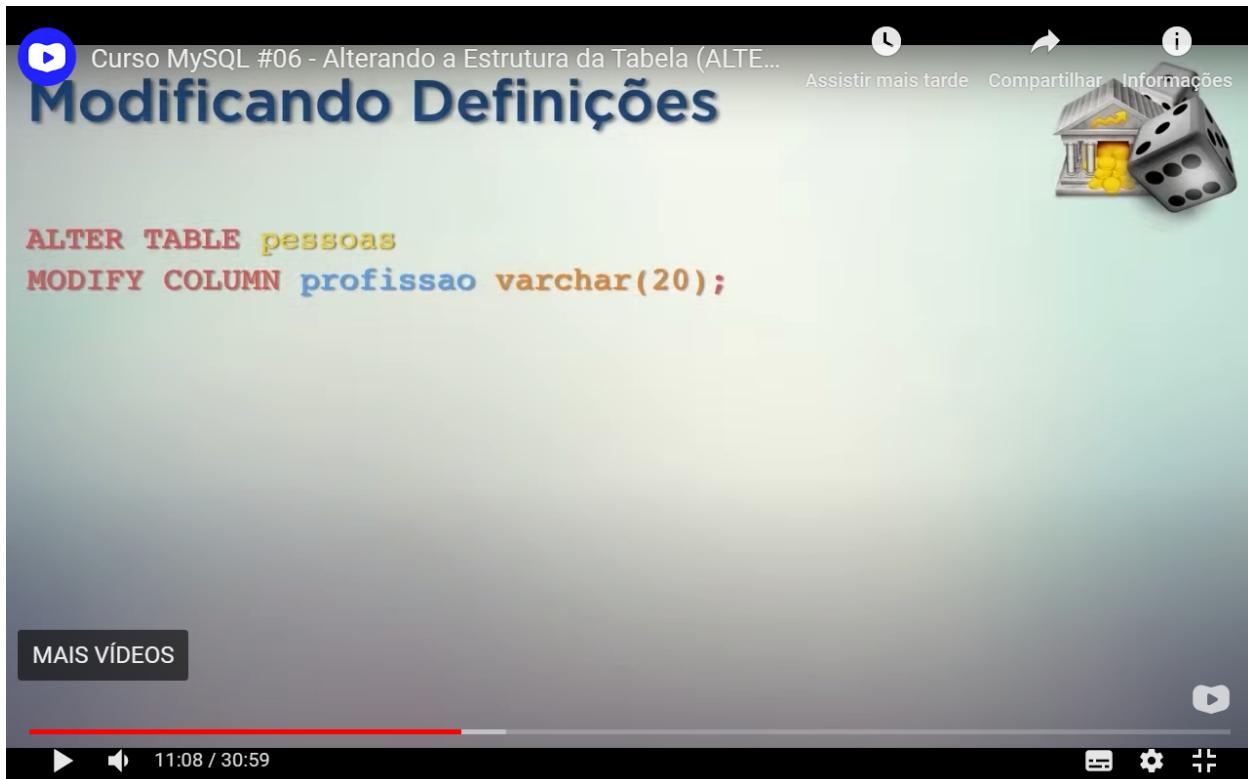


Posição Dos Column

Caso eu queria adicionar ele em primeiro lugar, vamos usar o seguinte código `add column pessoa varchar(30) first;` assim ele irá inserir no começo, agora caso eu queria colocar ele depois de alguma coisa, vou com o `after (nome da column)` , agora caso eu não coloque nada, ele sempre vai ser inserido no último.

Mudanças no tipo primitivo

Caso eu queria mudar o tipo primitivo de uma column vou o seguinte código, `modify column (nome da column) (e tipo primitivo que eu desejo mudar)`



Mudar o nome da Column

Caso eu queria mudar o nome da minha column vou usar a seguinte código `change column (o nome antigo da column) (nome novo da column)` segue exemplo, caso eu não coloque os tipos primitivos ele irá zerar e perderá todas as configurações feitas, então caso eu queria que não perca nada, eu preciso especificar tudo, mas isso só no `change`

Renomeando Coluna



```
ALTER TABLE pessoas  
CHANGE COLUMN profissao prof varchar(20);
```



Modificar o nome da table

Caso eu queria mudar o nome da table toda, vou usar o seguinte código. `Rename to (nome novo);`

Renomeando Tabela



```
ALTER TABLE pessoas  
RENAME TO gafanhotos;
```



Parametros no create

Na hora de criar uma table eu posso dar uma verificação, se caso existe essa table ela não irá ser criado ou ao contrário.

se eu colocar `create table if not exist (nome da table)` ele irá ser criado se ela não existir, mas se caso ela existir vai ficar com `create table if exist (nome da table)` que ele irá criar outra table.

Manipulação de Linhas

Caso eu precise mudar algo dentro da minha planilhas, vou usar os seguintes códigos para fazer isso.

```
Update(nome do table)  
set (nome da coluna) = (o que eu vou querer mudar)  
where (nome da coluna) = (Uma caracteristicas especifica na qual ele pode achar isso)  
limit 1;
```

Na parte de `Update` e o que vamos atualizar, se vamos atualizar a table irá ficar `update(nome da planilha)`

Além disso é necessário usar a seguinte código quando fomos fazer o `upadte` e esse código é o `limit` pois pode haver algum momento que pode modificar várias linhas, mas com o `limit` isso não acontece.

E o `Set` é usado para especificar a coluna que ele deseja mexer.

Delete and Truncate

Quando a gente for deletar algo vamos usar os seguinte códigos, `delete from (nome da table)` e agora algo que pode especificar ela, `where (alguma coluna que pode especificar)` caso eu queria eu também posso usar o `limit` para limitar algo como a modificação de linha. Agora veja o exemplo abaixo

```
delete from curso
where idcurso ='10'
limit 1;
```

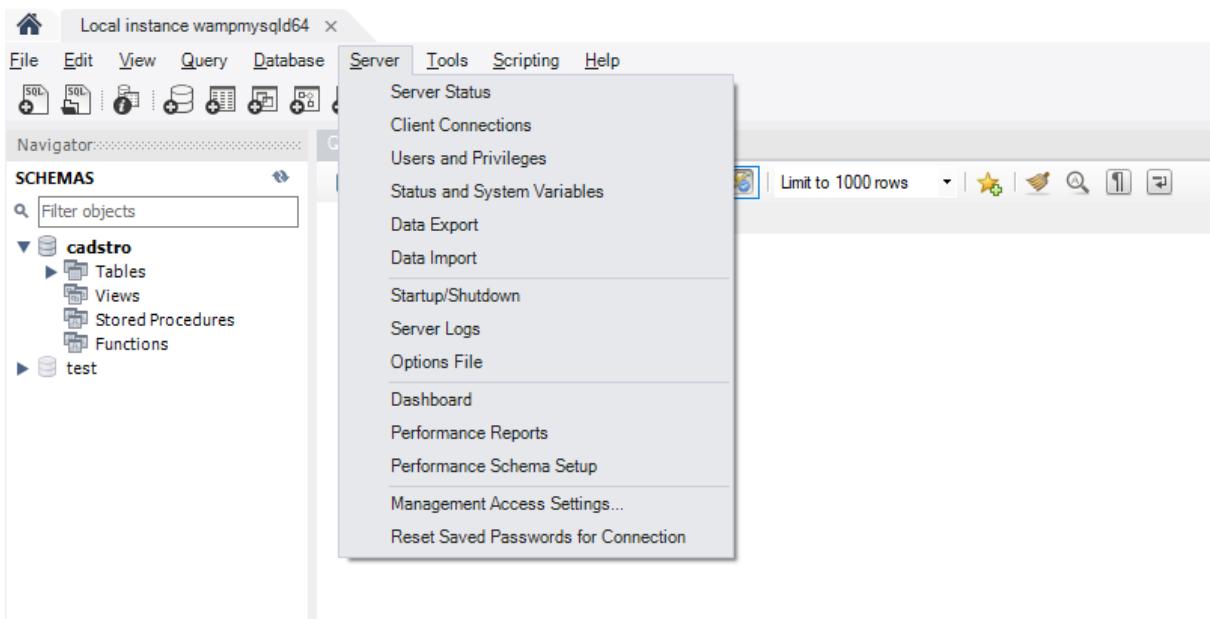
Agora caso eu queria apagar todos os dados da nossa table, vamos usar o seguinte código `Truncate table (nome da table)` assim vai apagar tudo que está dentro dela. E podemos usar para delimitar a informação que irá chegar para o usuário assim usando quando chegar a parte de Select

Backup Importação De Bancos De Dados

Um termo muito utilizado na hora que eu vou fazer backup de algum banco de dados é do Dump, que é um termo usado na hora de falar que vamos fazer um Backup.

Para fazer um backup de um projetos precisamos ir na parte de **serve** e logo em seguida na parte **Data Export**. Lá irá abrir uma parte com todos os nossos bancos de dados, assim selecionando qual baco a gente deseja e vamos o backup deles.

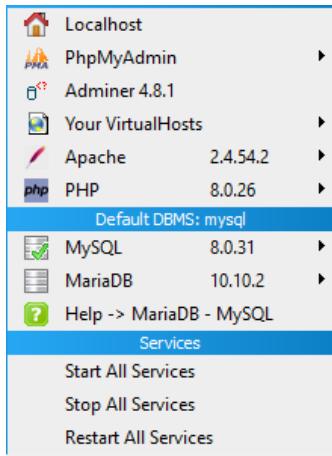
Agora para importar basta escolher a opção abaixo do **Data Export**. Para assim importar o banco de dados, de um lugar para outro.



Phpmyadmin

Nada mas é um aplicativo web na qual podemos modificar tudo o que desejamos, além de apresentar algumas funções idênticas a MySQL workbech, na minha opinião é uma maneira mais fácil de visualizar o que está sendo feito no Banco de Dados, além de conseguir sem fazer um código modificar tudo.

Para acessa-lo basta abrir o Wamp e quando aberto irá dar um interface que irá ter a opção do meu Phpmyadmin.



Quando fomos usar isso normalmente ele irá perdi uma senha e o nome do servidor, como estamos fazendo pelo o Wamp basta colocar como nome **Root** e a senha vazia, depois e só conectar.

Como usar

Posso ver os comando de como foi criado uma table no meu sql, basta eu abrir o Wamp e abrir ele pelo terminar, após isso basta eu fazer abrir a pasta a onde ele se localiza e dar p seguinte código no terminar. `Show create table (nome da tabela);` e serve também para o DataBase

Nomes

Você pode modificar os nomes da table indo em **operação** lá você pode mudar tudo, tanto o nome das table quanto o nome das colunas, além de conseguir mudar a database

Dump

Para eu gerar um dump no PhpMyAdmin preciso clicar na casinha, e logo em seguida em exportar, caso eu faço isso para exportar uma base de dados, ai isso iria funcionar só se caso eu queira importar o `database`.

Select

Distinct

Posso faze conta como várias outras coisas, além de apresentar vários outros comandos, como o `distinct` que não deixa nada se repetir, então quando possivelmente tem uma duplidade eu posso usar o `distinct` para retirar essa

duplicidade.

Length

Ele é um comando usado para conta a quantidade de caracteres que existe dentro de um `varchar`, melhor dizendo ele é o `Len` do Python.

Order By

Na parte do select podemos ter um comando chamado `order by` que esse comando nada mais é que um auxiliar do select, apenas podemos usar ele usando o select junto, mas o que ele faz??? Ele apresenta para o usuário de maneira que você quiser algo, exemplo.

```
Select * from cursos  
order by name;
```

Isso apresenta uma lista de select mas em forma alfabética, assim como eu posso fazer outras maneiras de apresentar o select.

Caso eu queria que seja apresentada uma lista de forma descendente basta na eu colocar o `desc` que na parte do select significa descendente.

```
Select * from (table escolhida)  
order by (alguma coluna da table) desc;
```

Diferença do Order By para o Where

`where` nada mais é mais um auxiliar do select, mas agora eu sei a diferença do `ordem by` e o `where`, o `ordem by` ele mexe com a coluna da table, então se eu precisar fazer alguma modificação da table e eu quero mudar alguma coluna, vou usar o `ordem by` mas agora se eu eu precisar alguma linha, vou usar o comando `where`.

E eu posso usar os dois juntos para dar um resultado ainda mais filtrado e preciso. Segue o exemplo abaixo.

```

4 • select * from cursos
5 where ano = '2016'
6 order by nome;
7
8 • describe cursos;


```

	idcurso	nome	descricao	carca	totaulas	ano
6	MySQL	Bancos de Dados MySQL	30	15	2016	
9	POO	Curso de Programação Orientada a O...	60	35	2016	
19	Redes	Curso de Redes para Iniciantes	40	15	2016	
► 7	Word	Curso completo de Word	40	30	2016	
*	HULL	HULL	HULL	HULL	HULL	

Diferença entre Between e In

Agora o `IN` vou colocar valores específicos agora no `Between` eu posso especificar faixas de valores, então essa é a maior diferença entre eles, segue o exemplo abaixo.

O `Between` um comando usado para dar valores que estão em um determinado momento até outro. Segue o exemplo abaixo.

```

1 • select nome, ano from cursos
2 where ano between 2014 and 2016;

```

Agora o `In` vai apenas mostrar valores específicos que colocou para ser mostrado, igual o exemplo abaixo e não se esqueça de colocar as coisas entre parênteses, como no exemplo abaixo.

```

select nome,descricao,ano from cursos
where ano in (2014, 2016)
order by ano;

```

Diferença And e Or

A diferença é simples caso eu coloque `and` vai apresentar o valor que eu quero se as duas coisas forem corretas, agora para o `or` vai me apresentar o resultado se uma das duas forem corretas, então se uma já estiver certo, ele me dará o resultado.

p	q	p AND q
V	V	V
V	F	F
F	V	F
F	F	F
p	q	p OR q
V	V	V
V	F	V
F	V	V
F	F	F

Like

Ele é bem parecido com igual, mas a diferença que o `=` é um caractere, já o `LIKE` e tudo que possui aquele caractere específico que eu quero, para usar isso também é preciso colocar um `%`

Segue o exemplo.

```
select * from cursos
where nome like 'p%'
```

Código

3	Photoshop5	Dicas de Photoshop CC	10	8	2014
4	PHP	Curso de PHP para iniciantes	40	20	2015
30	PHP4	Curso de PHP, versão 4.0	30	11	2010
29	PHP7	Curso de PHP, versão 7.0	40	20	2020
9	POO	Curso de Programação Orientada a Objetos	60	35	2016
16	PowerPoint	Curso completo de PowerPoint	30	12	2018
22	Premiere	Curso de Edição de Vídeos com Premiere	20	10	2017
8	Python	Curso de Python	40	18	2017

Resultado

White Card

Nada mais é códigos que são auxiliadores para fazer determinada função, um bom exemplo disso é o `%` que foi usado no exemplo acima.

Caso eu coloque no final o por cento, ele vai me apresentar todas as informações que tem a determinada coisa que eu queria no começo, agora caso se eu coloque no final, vai apresentar apenas aqueles dados que terminam com ele.

Com o % no final

```
select * from cursos  
where nome like '%a'
```

5	Java	Introdução à Linguagem Java	40	29	2015
20	Segurança	Curso de Segurança	15	8	2018
25	Joomla	Curso de Criação de Sites com Joomla	60	30	2019

Outra **white card** é o _ que serve para, precisa ter outro caractere depois ou antes das coisas que você está procurando, exemplo

```
select * from cursos  
where nome like 'ph%p_'
```

3	Photoshop5	Dicas de Photoshop CC	10	8	2014
30	PHP4	Curso de PHP, versão 4.0	30	11	2010
29	PHP7	Curso de PHP, versão 7.0	40	20	2020

Vamos ver esse exemplo, vemos que ter que começar por ph, logo em segunda usamos o % isso é o parâmetro para definir que tem começar com ph e que vai terminar com p também, agora o _ serve para falar que precisar ter alguma coisa no final.

Distinct

Serve para retirar as repetições do select, exemplo, eu preciso pegar um país de origem de várias pessoas e eu sei que a maioria é Brasileira, então, para não ficar repetida vou usar o seguinte comando, **Distinct** para que na hora que for apresentado seja uma vez só.

Brasil	Brasil
Portugal	Portugal
Moçambique	Moçambique
Irlanda	Irlanda
Brasil	EUA
Brasil	Japão
EUA	Canadá
Brasil	Angola
Portugal	Alemanha
EUA	Itália
Irlanda	
Brasil	
Brasil	
Portugal	
EUA	
Brasil	
França	
Brasil	
Portugal	
Brasil	

Sem o `Distinct`

Com o `Distinct`

Função De Agregação ou Funções de grupo de valores

- `AVG(n)`= Calcula o valor médio de "n" ignorando os valores nulos.
- `COUNT (*) | Expressão`= Conta o número de vezes que a expressão avalia algum dado com valor não nulo. A opção "*" conta todas as linhas selecionadas.
- `MAX (expressão)`= Calcula o máximo.
- `MIN (expressão)`= Calcula o mínimo.
- `SUM (expressão)`= Obtém a soma dos valores da expressão.

Exemplo do ChatGpt sobre a imagem acima.

notas	
aluno_id	nota
1	7.5
2	6.8
3	8.2
4	9.1
5	7.0
6	6.5

O primeiro que apresenta é um exemplo com `AVG(n)`

Para calcular a média das notas, podemos usar a seguinte QL:

```
sql
SELECT AVG(nota) as media_notas FROM notas;
```

clientes		
id	nome	idade
1	João	35
2	Maria	28
3	Pedro	42
4	Ana	28
5	Luiz	50
6	Fernanda	31

O segundo apresenta com o `COUNT`

para contar o número de clientes na tabela "clientes", podemos usar a seguinte QL:

```
sql
SELECT COUNT(*) as num_clientes FROM clientes;
```

produtos		
id	nome	preco
1	Camiseta	29.90
2	Calça jeans	99.90
3	Tênis	159.90
4	Blusa	49.90
5	Jaqueta	249.90
6	Meia	9.90

produto_mais_caro		
		249.90

Veja que ele apresentou o item mais caro da apresenta na loja. já para pegar o menor iremos ver no exemplo abaixo.

Para encontrar o produto mais caro da tabela "produtos", podemos!

SQL:

```
sql
SELECT MAX(preco) as produto_mais_caro FROM produtos;
```

vendas		
id	data	valor_total
1	2022-01-01	150.00
2	2022-01-03	80.00
3	2022-01-05	200.00
4	2022-01-07	120.00
5	2022-01-09	50.00
6	2022-01-11	90.00

valor_minimo		
		50.00

Apresentou o menor valor de todos que havia dentro da lista.

Para encontrar o valor mínimo na coluna "valor_total" da tabela "ve

gueinte consulta SQL:

```
sql
SELECT MIN(valor_total) as valor_minimo FROM vendas;
```

```
-- criação da tabela "vendas"
CREATE TABLE vendas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    data_venda DATE NOT NULL,
    valor_venda DECIMAL(10,2) NOT NULL
);

-- inserção de dados na tabela
INSERT INTO vendas (data_venda, valor_venda) VALUES
('2022-01-01', 100.50),
('2022-01-02', 200.75),
('2022-01-03', 150.00),
('2022-01-04', 75.25),
('2022-01-05', 300.00);

-- soma dos valores de venda
SELECT SUM(valor_venda) AS total_vendas FROM vendas;
```

total_vendas

826.50

Ele vai pegar os valores de todos e vai somar e dar o resultado dele.

Grupos

Além de separar eles podemos também juntá-los com alguns comandos como `group by` e o próprio `where`, normalmente eles usam bastante o agrupamento com as funções de agregações, então segue o exemplo abaixo.

```
select ano,count(*) from cursos //Nessa parte eu vou apresentar o ano e a quantidade de vezes que aparece esse ano na table
group by ano //Agrupei todos os anos iguais. então com o count eu consigo contar quantos anos iguais tem na table
order by count(*) desc; //eu vou order eles por contador e do maior para o menor
```

Além disso, a outro comando que é uma auxiliar do `group by` que é o `having`, ele apenas funciona se eu apresentar o `group by` e ele é usado como se fosse um `where` mas com diferença que ele vai filtrar grupos, segue o exemplo abaixo.

```
select ano, count(*) from cursos //vai me apresentar o ano e quantidade de cursos nesse ano
group by ano //agrupados pro anos
having ano > 2013 //parecido com where, mas de grupos então vai em apresentar grupos acima de 2013
order by ano desc; //irá ser apresentado por ano do maior para menor
```

Dois Select

Podemos dar dois `select` para filtrar uma coisa mais específica, caso eu queira a média de alunos que tiraram mais da metade da nota vou fazer o seguinte.

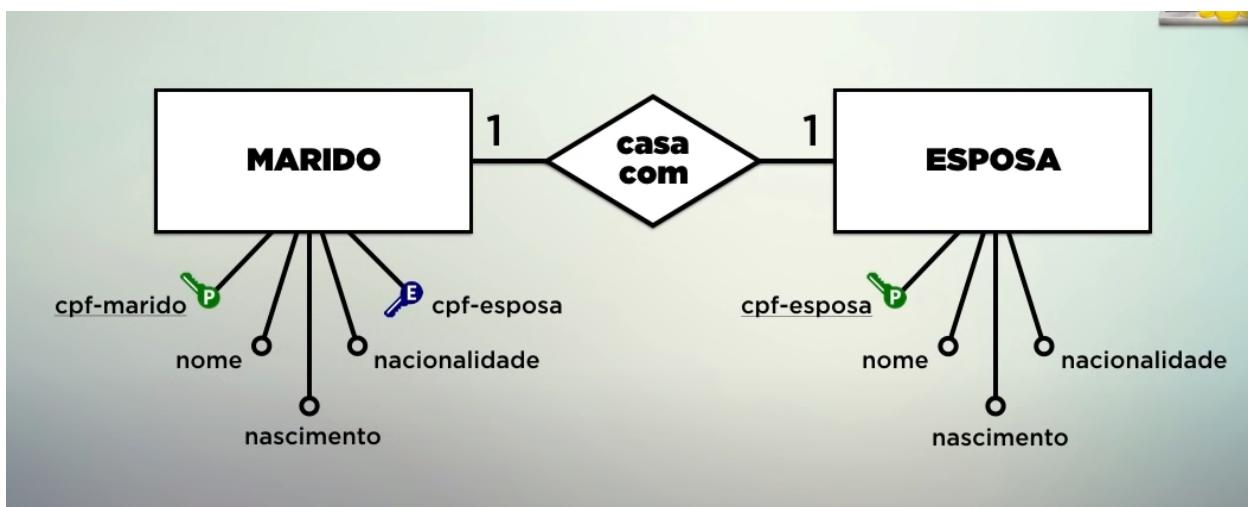
```
select nota, count(nota) from alunos //Vai puxar a nota e a quantidade de notas iguais  
group by alunos //Agrupa tudo com no alunos  
having nota > (select avg(nota) from alunos); //Vai me dar a nota e quantidade de pessoa que tiraram essa nota
```

Chave Estrangeira

Quando temos duas tabelas e precisamos importar uma chave primária para a outra tabela ao invés de chamar de chave primária x, chamamos ela de chave Estrangeira.

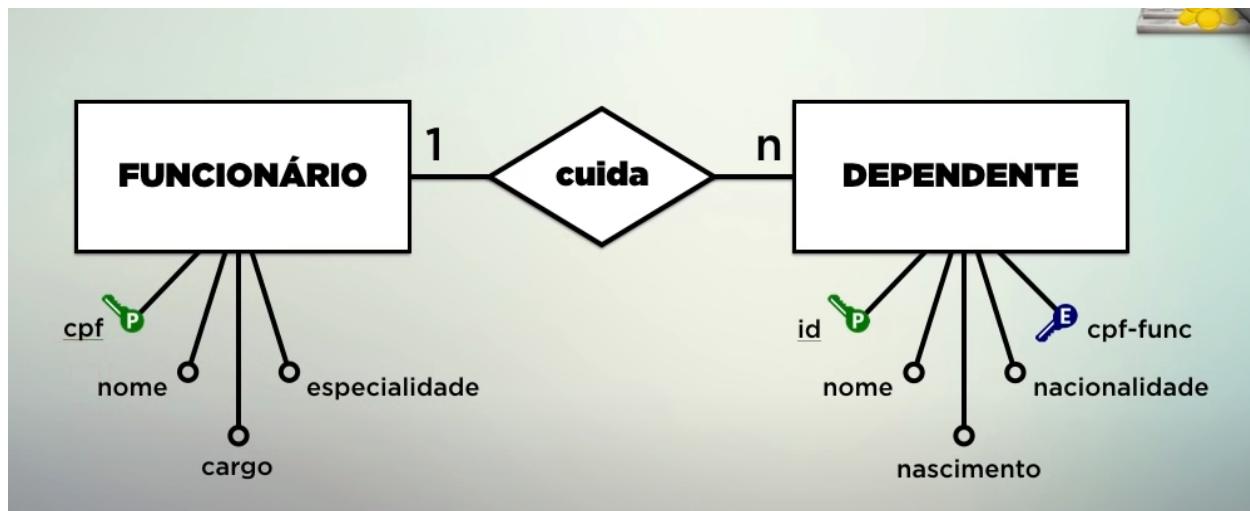
Relacionamento Um para Um

Para eu colocar uma chave estrangeira em outro local eu preciso ver se a chave estrangeira do local é compatível com a primária, exemplo, as duas precisam ser `VARCHAR` ou `INT` mas isso é para regras de relacionamento um para um. E também precisa ter sempre um dominante, para isso basta escolher um e trazer até ele a chave estrangeira.



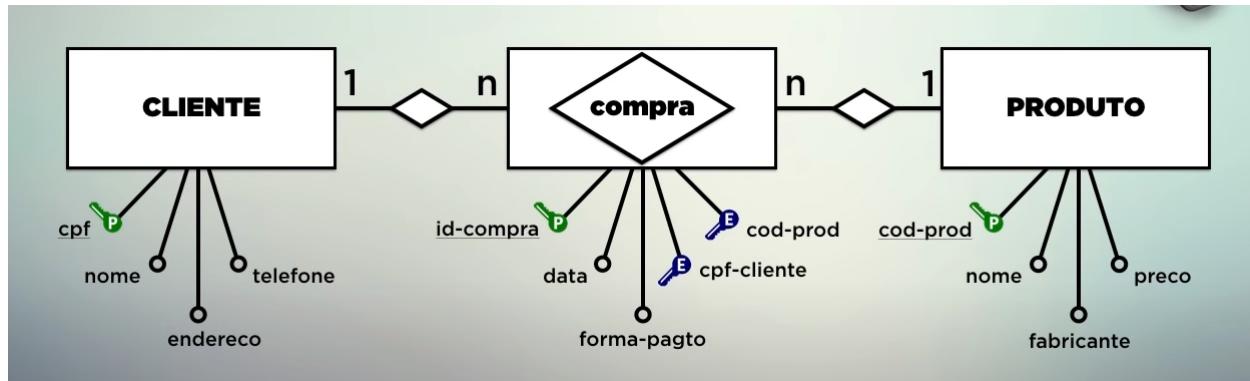
Um para Muitos

Já na de um para muitos eu preciso importar a chave primária do um para os muitos, assim fazendo ela virar uma chave estrangeira

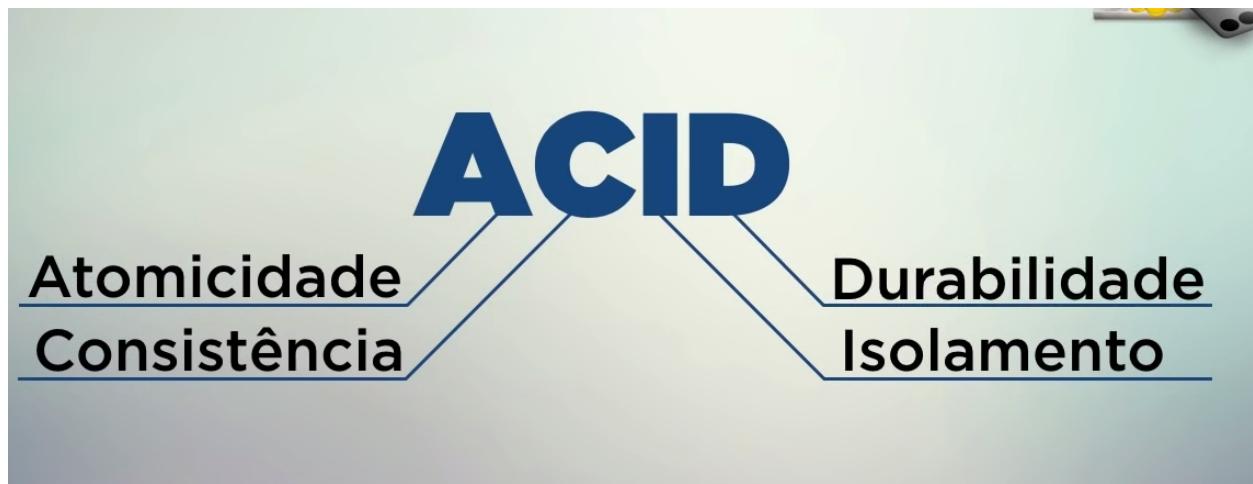


Muitos para muitos

De todos essa é o mais complicado, pois o relacionamento(Losango) vai virar uma entidade(Contêiner), logo após isso eu preciso identificar a nova conexão entre eles, no exemplo que eu peguei, ela vai ficar um para vários. Assim caso eu precise fazer algo basta eu fazer pela a conexão que eu tenho.



Para uma transferência acontecer e preciso



Como add uma chave estrangeira

Para `add` uma chave estrangeira primeiro precisamos ver qual é o tipo primitivo dela, após isso temos que `add` um `column` nova e especificar o tipo dela, logo em seguida fazemos outro `alter table` agora adicionando um valor novo para coluna como `foreign key` e o nome da coluna e damos o local que ele é localizado com o `references`. Para fazer isso segue o exemplo abaixo.

```
alter table (nome da table)
add column (nome da nova coluna) (e o tipo primitivo dela);

alter table (nome da table)
add foreign key (nome da nova coluna)
references (local a onde se localiza o que eu desejo trasferir).
```

**ALTER TABLE gafanhotos
ADD COLUMN cursopreferido int;**

**ALTER TABLE gafanhotos
ADD FOREIGN KEY (cursopreferido)
REFERENCES cursos(idcurso);**

Delete na ligação

Normalmente quando queremos apagar algo é bem fácil apenas dar o comando delete o que queremos, mas quando ele apresenta ligações entre uma tabela e outra isso já muda. Não conseguimos apagar uma coluna, pois ela apresenta uma ligação entre elas.

Join

Quando fazemos a ligação entre as duas tabelas, ele ficou apresentado apenas o id do curso, mas se caso eu queira o nome dela ao invés do id. Para isso usamos o `join`

Nada mas é que a ligação entre uma tabela e outra, assim puxando determinadas informações de um tabelada para outra. Por causa disso é preciso os dois terem o mesmo tipo primitivo, para que haja ligação entre as duas chaves.

Caso eu faça sem o `on` vai funcionar, mas o resultado apresentado vai ser outro.

```
select (nome da tabela).(nome coluna) from (nome da table)
join (nome da outra table)
on (chave primaria) = (chave secundaria);
```

```
select gafanhotos.nome, gafanhotos.cursopreferido, cursos.nome, cursos.ano
from gafanhotos join cursos
on gafanhotos.cursopreferido = cursos.idcurso;
```

Talita Nascimento	1	HTML5	2014
Emerson Gabriel	2	Algoritmos	2014
Lucas Damasceno	3	Photoshop5	2014
Leila Martins	4	PHP	2015
Letícia Neves	5	Java	2015
Daniel Morais	6	MySQL	2016
Janaína Couto	7	Word	2016
Carlisson Rosa	8	Python	2017
Jackson Telles	9	POO	2016
Danilo Araujo	10	Excel	2017

Esse é o resultado ligando as duas tabelas, vemos que deram todos cadastros que de pessoa que havia um curso preferido.

Right Join and Left Join

Temos três tipos de `join` que são o left o right e o normal, quando usamos o normal igual o exemplo acima apresenta apenas o resultado da pessoas que eu apresenta as duas chaves primaria e o que eles preferem, mas agora eu todos as pessoas não só aquelas que preferem algo, então antes do join irei dar ou `left` ou `right`, pois

com isso ele vai dar prioridade para tabela que estiver no comando, vai depender de como eu vou montar o código.
Vou tomar o código de cima como exemplo.

```
select g.nome, g.cursopreferido, c.nome, c.ano  
from gafanhotos as g right join cursos as c  
on g.cursopreferido = c.idcurso;
```

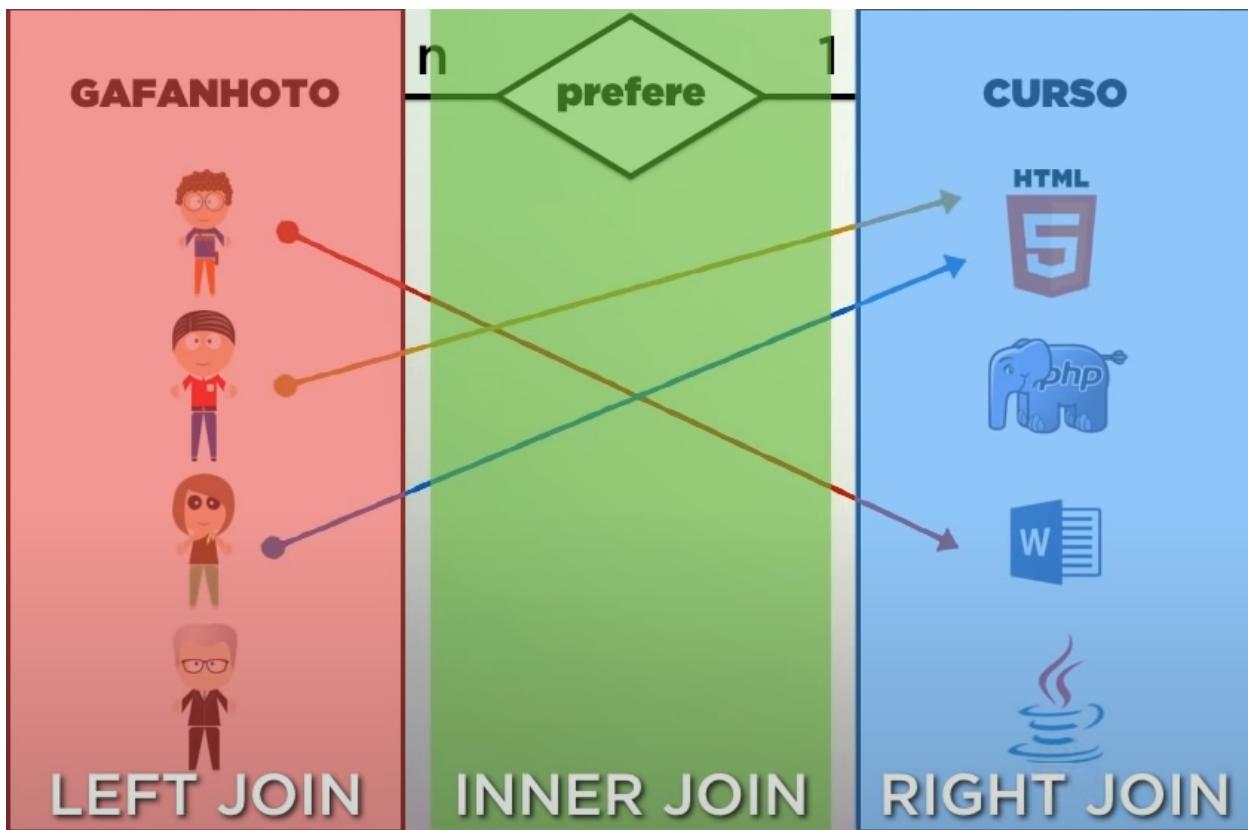
Lucas Damasceno	3	Photoshop5	2014
Leila Martins	4	PHP	2015
Letícia Neves	5	Java	2015
Daniel Moraes	6	MySQL	2016
Janaína Couto	7	Word	2016
Carlisson Rosa	8	Python	2017
Jackson Telles	9	POO	2016
Danilo Araujo	10	Excel	2017
NUL	NUL	Responsivi...	2018
NUL	NUL	C++	2017
NUL	NUL	C#	2017
NUL	NUL	Android	2018
NUL	NUL	JavaScript	2017
NUL	NUL	PowerPoint	2018
NUL	NUL	Swift	2019
NUL	NUL	Hardware	2017
NUL	NUL	Redes	2016
NUL	NUL	Segurança	2018
NUL	NUL	SEO	2017
NUL	NUL	Premiere	2017
NUL	NUL	After Effects	2018
NUL	NUL	WordPress	2019
NUL	NUL	Joomla	2019
NUL	NUL	Magento	2019
NUL	NUL	Modelagem...	2020
NUL	NUL	HTML4	2010
NUL	NUL	PHP7	2020
NUL	NUL	PHP4	2010

Resultado usando o `right` vemos que ele apresenta as pessoa que gostam do curso e apresenta logo em seguida todos os cursos, pois ele como eu coloquei um `right` ele vai priorizar a tabela cursos

Daniel Moraes	6	MySQL	2016
Talita Nascimento	1	HTML5	2014
Emerson Gabriel	2	Algoritmos	2014
Lucas Damasceno	3	Photoshop5	2014
Leila Martins	4	PHP	2015
Letícia Neves	5	Java	2015
Janaína Couto	7	Word	2016
Carlisson Rosa	8	Python	2017
Jackson Telles	9	POO	2016
Danilo Araujo	10	Excel	2017
Andreia Delfino	NULL	NULL	NULL
Valter Vilmerson	NULL	NULL	NULL
Allan Silva	NULL	NULL	NULL
Rosana Kunz	NULL	NULL	NULL
Josiane Dutra	NULL	NULL	NULL
Elvis Schwarz	NULL	NULL	NULL
Paulo Narley	NULL	NULL	NULL
Joade Assis	NULL	NULL	NULL
Nara Matos	NULL	NULL	NULL
Marcos Dissotti	NULL	NULL	NULL
Ana Carolina Me...	NULL	NULL	NULL
Guilherme de Sousa	NULL	NULL	NULL
Bruno Torres	NULL	NULL	NULL
Yuji Homa	NULL	NULL	NULL
Raiam Porto	NULL	NULL	NULL
Paulo Batista	NULL	NULL	NULL
Monique Precivalli	NULL	NULL	NULL
Herisson Silva	NULL	NULL	NULL

Aqui eu já fiz pleo o `left` então o que está com prioridade é os gafanhotos, além de apresentar todos pessoa que apresenta as duas chaves que eu queria, apresenta todo também, mas agora apresenta pessoa com prioridade os nomes.

Uma boa maneira de entender melhor o que eu acabei falando é essa imagem que na qual vai uma representação do que.



AS

O **AS** nada mas seria que a possibilidade de eu relacionar um apelido para uma tabela, exemplo, eu quero chamar a minha tabela cadastro de C, eu vou fazer na seguinte forma.

```
select g
from cadastro as C;
```

```
//Um exemplo mais pratico//

select g.nome,
c.nome,
c.ano
from gafanhotos as g join cursos as c
on c.idcurso = g.cursopreferido
order by g.nome;
```

Muitos para muitos

O modelo feito acima era do modelo um para vários, assim aplicando apenas o métodos de um para vários, agora de vários para vários?

Para isso iremos ter que fazer igual no relacionamento muitos para muitos, precisamos criar uma entidade, que nela irá trazer as duas chaves primaria das entidades e assim conseguir fazer a relação de muitos para muitos

Curiosidades

O curso em vídeo acabou, agora apenas resta as coisas que eu for aprender sozinho, então para começar bem, aqui está o primeiro conteúdo de estudo.

Length and Char_length

Quando queremos saber a quantidade de caracteres que existe dentro de `varchar` precisamos usar o `Char_length`, pois com ele achamos quantos espaços foram usado, mas agora se caso eu use o `Length` ele apenas irá me dar a quantidade

DateDiff

Ele me da a quantidade de dias de um certo período na qual eu posso escolher, um bom exemplo disso é o relatório que chefe geral do hospital pediu, ele queria os dias internados de determinados pacientes, para isso eu vou pegar a data que o usuário irá inserir como uma data final e data inicial, assim, pegando os dias internados nela. Segue o exemplo abaixo.

```
select data_ini, data_final, datediff(data_final, data_ini) as quantidade_dias  
from test_tempo_inter;
```

	data_ini	data_final	quantidade_dias
▶	2023-06-01	2023-06-21	20
	2023-06-01	2023-06-21	20

Regexp

Pode ser usado para quando queremos selecionar algo dentro de texto ou até mesmo dentro puxar alguma informação dentro do `varchar`, para usar essa função vamos usar ela logo após o `select` junto com um `where`,

dentro dessa função apresenta muitas características boas

Igual um print

Quando vamos apresentar algo para o usuário igual um print no SQL precisamos de determinadas condições e regras, vou apresentar o código

```
SELECT 'De: ' || TO_CHAR(:dt_inicial, 'dd/mm/yyyy') || ' até ' || TO_CHAR(:dt_final, 'dd/mm/yyyy')
as ds_periodo
FROM dual
```

Primeiro, o começo dele é igual a qualquer outro `Select` mas tem algumas coisas em diferentes do resto do código como.

- `||` esse código nada mais é que um +
- `TO_CHAR` e esse transforma o atributo em outro, nesse caso ele está transformando uma data em `VARCHAR`, outra coisa é que ela tem algumas regras para funcionar a primeira é que vou precisar colocar entre parentes o atributo que eu quero, logo após eu colocar uma vírgula vou inserir uma máscara em **Aspas Simples**.

Outra coisa que foi introduzida aqui foi o `From dual` que normalmente é usado para quando queremos fazer esse print no SQL, mas também pode ser usado para fazer contas ou printar algo, segue outros exemplos de como usar o `From Dual`

```
SELECT 'Hello, World!' FROM DUAL; -- Retorna a string 'Hello, World!'
```

```
SELECT 5 + 3 FROM DUAL; -- Retorna 8
```

Caso HEM From Dual

Quando vamos usar ele no HEM para fazer datas eu preciso colocar um apelido nele para eu conseguir puxar ele na parte das bandas, então preciso também configurar nas bandas.