

---

# **Explorando o Poder do MySQL**

um Guia Abrangente para Gerenciamento de Bancos de  
Dados Relacionais

V1.0.0 (OCT/2023)



## Sumário

<b>1</b>	<b>introdução</b>	<b>3</b>
<b>2</b>	<b>Endereçamento</b>	<b>4</b>
2.1	Dica de curso . . . . .	4
2.2	comandos: . . . . .	4
2.3	Tipos primitivos: . . . . .	4
2.4	Operadores Literário . . . . .	5
2.5	Sintaxe de Operadores: . . . . .	6
2.6	Operador select e seus complementos . . . . .	9
2.7	Funções de agregação . . . . .	11
2.8	Modelo Relacional . . . . .	12
2.9	Chave estrangeira . . . . .	15
2.10	Baixar Banco de dados para estudo . . . . .	20
<b>3</b>	<b>Change log</b>	<b>23</b>
3.1	version 1.0.0 . . . . .	23

## 1 introdução

O **MySQL** é amplamente apreciado por sua eficiência e facilidade de uso, permitindo que desenvolvedores e administradores de banco de dados gerenciem grandes quantidades de informações com facilidade. Ele oferece suporte para recursos avançados, como replicação, clusterização e particionamento, que são cruciais para aplicativos que exigem alta disponibilidade e escalabilidade. Isso o torna uma escolha popular para empresas que precisam de um banco de dados robusto e seguro para suas operações diárias.

Uma das características distintivas do **MySQL** é sua comunidade ativa e vibrante. Há uma vasta quantidade de recursos online, incluindo documentação, fóruns e comunidades de desenvolvedores que ajudam a resolver problemas e compartilhar conhecimento. Além disso, o **MySQL** oferece suporte comercial por meio da Oracle, garantindo que as empresas tenham acesso a serviços de suporte confiáveis.

O **MySQL** é usado em uma ampla variedade de cenários, desde pequenos sites pessoais até sistemas corporativos de missão crítica. Ele é particularmente adequado para aplicativos da web, com muitos dos maiores sites da internet usando **MySQL** como seu sistema de gerenciamento de banco de dados, graças à sua capacidade de lidar com cargas de trabalho de alto tráfego e sua capacidade de escalabilidade.

Além disso, o **MySQL** desempenhou um papel crucial no crescimento da economia digital e da revolução da informação. À medida que a era da internet evoluiu, o **MySQL** se destacou como a espinha dorsal de muitos dos serviços online que usamos diariamente. Desde redes sociais e lojas online até serviços de streaming de vídeo e música, inúmeras plataformas e aplicativos confiam no **MySQL** para armazenar e recuperar dados de maneira rápida e eficiente.

A versatilidade do **MySQL** também o torna uma escolha popular para organizações de todos os tamanhos. Seja para pequenas empresas que desejam gerenciar informações de clientes ou para grandes corporações que precisam consolidar dados de várias fontes, o **MySQL** oferece uma solução flexível que pode ser adaptada a uma ampla gama de necessidades.

Em última análise, o **MySQL** é muito mais do que apenas um sistema de gerenciamento de banco de dados; ele é um habilitador da inovação, permitindo que aplicativos e serviços inovadores prosperem na era digital. Sua influência e importância no mundo da tecnologia o tornam uma ferramenta essencial para qualquer pessoa envolvida no desenvolvimento de software e na gestão de dados.

Em resumo, o **MySQL** é um pilar na paisagem dos sistemas de gerenciamento de banco de dados, graças à sua combinação de desempenho, confiabilidade e acessibilidade. Seja para desenvolvedores individuais que desejam criar aplicativos da web, ou para grandes empresas que precisam de um sistema robusto de banco de dados, o **MySQL** continua a ser uma escolha confiável e popular em todo o mundo.

## 2 Endereçamento

### 2.1 Dica de curso

#### 2.1.1 Click Aqui para direcionar ao curso em vídeo

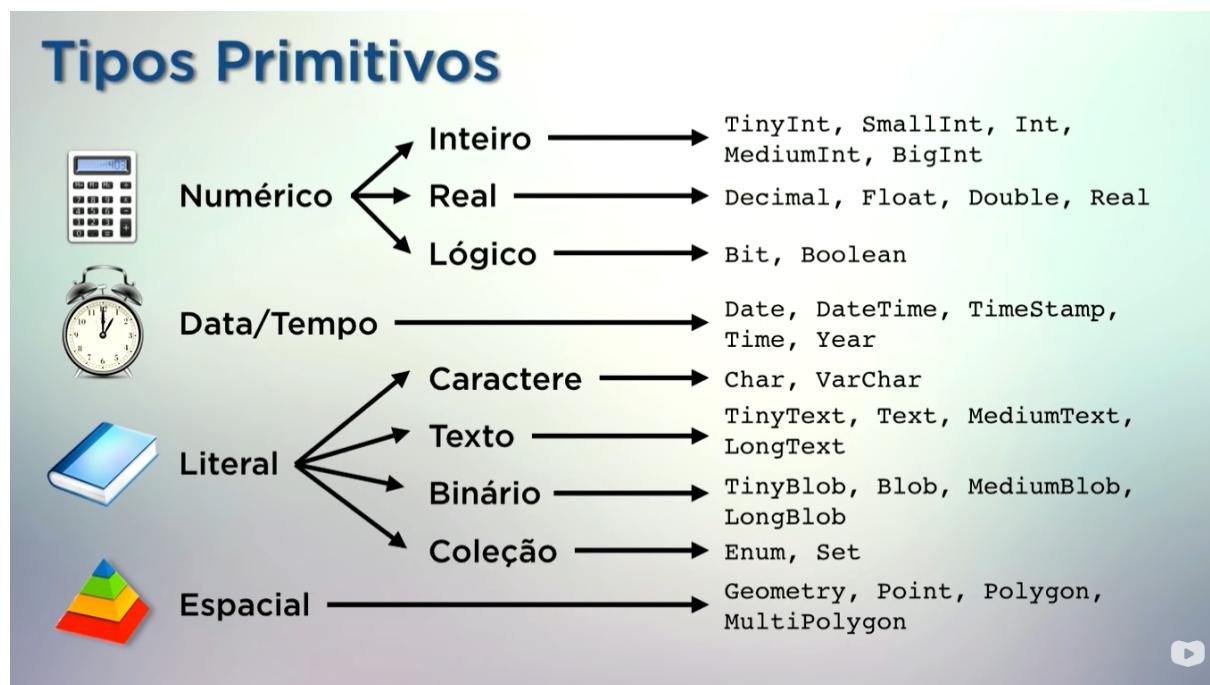
### 2.2 comandos:

```

1 create database ("Nome do banco de dados"); `Criar banco de dados`
2 use 'Nome do banco de dados' `Use esse comando para usar o banco de
   dados`
3 create table "Nome da tabela"( `Criar tabela para adicionar as
   variaveis`
4 id int auto_increment not null,
5 nome varchar(20) not null,
6 sexo enum("M", "F"),
7 altura decimal(5,2),
8 primary key (id);

```

### 2.3 Tipos primitivos:



**Figura 1:** Tipos primitivos

O **MySQL** oferece diversos tipos de dados primitivos que podem ser usados para armazenar informações em um banco de dados. Alguns dos tipos de dados primitivos mais comuns no **MySQL** incluem:

### 1. Números:

- Inteiros (INT, INTEGER): Representa números inteiros, positivos ou negativos, sem casas decimais.
- Números de ponto flutuante (FLOAT, DOUBLE): Representa números com casas decimais.

### 2. Texto:

- Caractere (CHAR, CHARACTER): Armazena uma sequência fixa de caracteres.
- Varchar (VARCHAR, VARIABLE CHARACTER): Armazena uma sequência variável de caracteres.
- Texto (TEXT): Usado para armazenar grandes quantidades de texto.

### 3. Datas e Horas:

- Data (DATE): Armazena datas (ano, mês e dia).
- Hora (TIME): Armazena horários do dia.
- Data e Hora (DATETIME, TIMESTAMP): Armazena data e horário juntos.

### 4. Booleano:

- Booleano (BOOL, BOOLEAN): Armazena valores verdadeiro (TRUE) ou falso (FALSE).

### 5. Imagens e Arquivos:

- Blob (BINARY LARGE OBJECT): Usado para armazenar dados binários, como imagens, áudio ou outros tipos de arquivos.

### 6. Outros Tipos Especiais:

- Enumeração (ENUM): Define um conjunto de valores possíveis.
- Conjunto (SET): Similar a ENUM, mas pode armazenar múltiplos valores selecionados.

### 7. Tipos Geoespaciais (para Sistemas de Informações Geográficas):

- Pontos, linhas, polígonos, etc., utilizados para representar informações geográficas

## 2.4 Operadores Literário

1. **ALL** - VERDADEIRO se todos os valores da subconsulta atenderem à condição.
2. **AND** - VERDADEIRO se todas as condições separadas por E for VERDADEIRO

3. **ANY** - VERDADEIRO se qualquer um dos valores de subconsulta atender à condição
4. **BETWEEN** - VERDADEIRO se o operando estiver dentro do intervalo das comparações
5. **EXISTS** - VERDADEIRO se a subconsulta retornar um ou mais registros
6. **IN** - VERDADEIRO se o operando for igual a um de uma lista de expressões
7. **LIKE** - VERDADEIRO se o operando corresponder a um padrão
8. **NOT** - Exibe um registro se a(s) condição(ões) NÃO FOR VERDADEIRA
9. **OR** - VERDADEIRO se qualquer uma das condições se separar
10. **SOME** - VERDADEIRO se qualquer um dos valores de subconsulta atender à condição
11. **AS** - ELE faz apelido de coluna (Exemplo: cursos, usando o as, cursos as c) Ele faz o nome cursos fica so c

## 2.5 Sintaxe de Operadores:

```

1 - use 'banco de dados';
2 -----
3 - describe 'tabela'; (desc)
4 - insert into tabela
5 - (variaveis)
6 - values
7 - ('O que você quer adicionar nessa tabela');
8 -----
9 - Select * from 'tabela'; `Seleciona`
```

*Adicionando e removendo colunas com o operador alter*

```

1 alter table 'Nome da tabela'
2 add column profissao varchar(20); `Nome da coluna que você quer
   adicionar`
3
4 alter table 'Nome da tabela'
5 drop column 'Nome da coluna que você quer remover'; `Nome da coluna que
   você quer remover`
```

*Adicionando colunar onde quiser*

```

1 alter table 'Nome da tabela'
2 add column profissao varchar(20) after 'Nome da coluna que você quer
   depois';
3
4 alter table 'Nome da tabela'
5 add column profissao varchar(20) first; `Adicionar a coluna em primeiro
```

*Renomear o nome da coluna*

```

1 alter table 'Nome da tabela'
2 change profissao prof varchar(20); `Renomear o nome da coluna que você
quer`
```

*Adicionando tipos primitivos em variáveis já existente*

```

1 alter table 'Nome da tabela'
2 add primary key ('Nome da coluna que você quer');
```

*Renomear a tabela*

```

1 alter table 'Nome da table antiga'
2 rename to 'Nome da table nova';
```

*Comandos completares ao se criar uma nova tabela ou um banco de dados*

```

1 if not exists(se não existir) ele serve para criar uma tabela que não
exista no seu banco de dados, exemplo(se existir uma tabela chamada
pessoas, e se você não coloca esse comando e caso você queira cria
uma tabela chamada pessoas ele apaga essa tua tabela pessoas e cria
uma nova, e você perde os dados) isso acontece para a mesma coisa no
banco de dados o create database.
2
3 create table if not exists cursos(
4 nome varchar(20) not null,
5 profissao varchar(20),
6 nascimento date
7 );
8
9 drop database if exists 'Nome da tabela'
10 if exists(se existir) ele só remove a tabela se realmente ela exista,
    se caso ela não exista, ela manda uma mensagem que ela não existi.
```

*Modificando valores da tabela*

```

1 update 'Nome da tabela'
2 set 'Nome da coluna' = ' ' `O que deseja renomear o valor`
3 where 'Onde vc quer alterar' = ' ' `Valor ou nome da onde quer alterar
`;
4
5 update 'Nome da tabela'
6 set 'Nome da coluna' = ' ' `O que deseja renomear o valor`, 'Nome da
    coluna' = 'Valor a ser alterado' `Isso é uma forma de renomear duas
    coisas ao mesmo tempo na mesma linha`
7 where 'Onde vc quer alterar' = ' ' `Valor ou nome da onde quer alterar
`;
```

*Cuidado !!!*

1 caso queira executar o código abaixo sem restrições, siga a dica abaixo

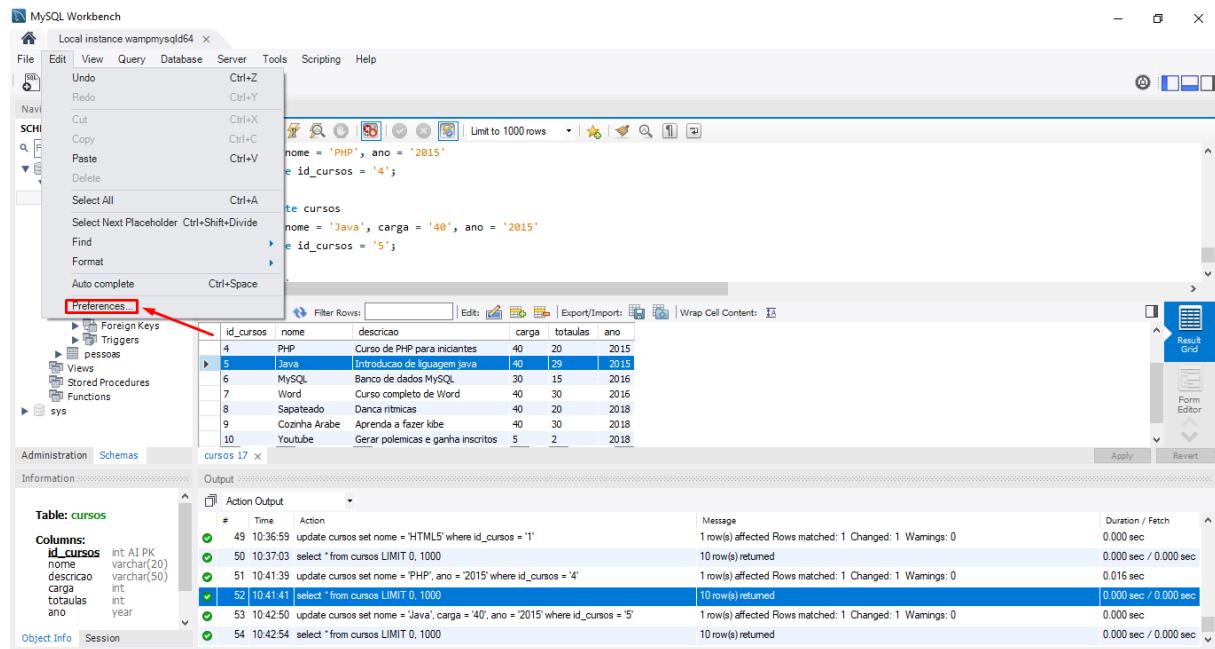


Figura 2: orientação

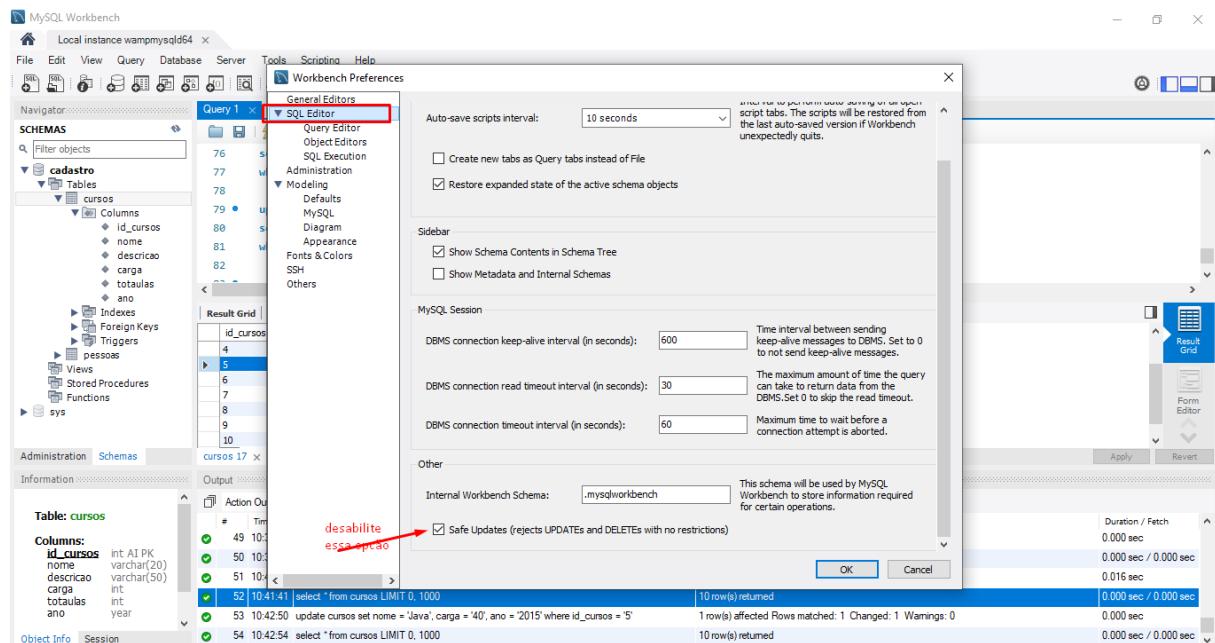


Figura 3: desabilitar

mas a plataforma MySQL Workbench ele tem tipo uma segurança nas configurações que ela evita que voce altere varios linhas de uma vez. caso você execute esse codigo ele altera tudo que esta com ano de 2015;

```
1 update cursos
2 set ano = '2050', carga = '100'
3 where ano = '2015';
4
5 update cursos
6 set ano = '2050', carga = '100'
7 where ano = '2015'
8 limit 1; `Ele limitar so uma alteração`
```

*deletar linha*

```
1 delete from 'Nome da tabela'
2 where 'Nome da variavel quer voce quer que delete as informações' = ' '
   `Onde voce quer deletar`;
3
4 delete from 'Nome da tabela'
5 where 'Nome da variavel quer voce quer que delete as informações' = ' '
   `Onde voce quer deletar`
6 limit 3; `Sempre e bom limitar pois caso teja coisas relacionada ao
mesmo nome ele apaga tudo`
```

*Apagar tudo da tabela*

```
1 truncate table cursos;
2 outra forma e mais simple:
3 truncate cursos;
```

## 2.6 Operador select e seus complementos

```
1 select * from 'Nome da tabela';
2
3 select * from 'Nome da tabela'
4 order by 'Nome da coluna que voce quer ordenar';
5
6 select * from 'Nome da tabela'
7 order by 'Nome da coluna' desc;
```

Esse desc seria para ser ordem decrescente nos nos ao inves de começa no a, ele começaria no w tem o asc e a mesma coisa de deixar vazio.

```

1 select 'Nomes da colunas que quer que apareça','mais colunas'(nao
    precisa coloca as '') from 'Nome da tabela';
2
3 select * from 'Nome da coluna'
4 where 'coluna' = 'variavel' 'Ele ordena onde você colocou a condição,
    se caso queira que ano seja ordenado adicione ele ao where'
5 order by;
6
7 select * from 'Nome da tabela'
8 where 'coluna' >= 10
9 order by 'coluna';

```

**order by Ele ordena onde você colocou a condição, se caso queira que ano seja ordenado adicione ele ao where**

#### OBS: Da para fazer com todos os operadores lógicos

```

1 select * from 'Nome da tabela'
2 where 'coluna' between 'valor' and 'valor'; (between seria entre)
3
4 select * from 'Nome da tabela'
5 where 'coluna' in ('valor','valor','valor'); (ele mostra os valores
    selecionados)

```

*Usando o operador Like*

```

1 select * from 'Nome da tabela' '% significa nenhum ou varios caractere'
2 where 'coluna' like 'a%';

```

**% significa nenhum ou varios caractere LIKE em mysql significa parecido ou semelhante a% ele vai busca os valores que começa com a letra a %a ele vai busca os valores que termina com a letra a %a% ele vai busca em qualquer lugar o valor em sua tabela, não importa onde seja a localização.**

```

1 select * from 'Nome da tabela'
2 where 'coluna' not like '%a%';

```

*Exemplo(Ele procura algo na tabela que não tenha a)*

```

1 select * from 'Nome da tabela'
2 where 'coluna' like 'Ph%p_';

```

**\*\* \_ \*\* ele exige que tenha algo nesse caractere não importa o que seja um numero ou uma letra**

*Distinguindo*

```

1 select distinct 'coluna' from 'tabela'
2 order by 'coluna';

```

**Distinct** ele busca sem repetir as palavras ou numero(exemplo: você tem quatro pessoas que mora no brasil e uma pessoa que mora em portugal, ao usar esse comando ele so vai busca 1 brasil e 1 portugal pois ele filtra)

## 2.7 Funções de agregação

### Count

```
1 select count(*) from 'Tabela';
2 select count('Coluna') from 'Tabela' where 'coluna' < 30;
```

Exemplo1 (Ele conta quantos valores cadastrado na sua tabela) Exemplo2 (Ele conta quantos valores tem menor que 30)

### Max

```
1 select max('Coluna') from 'Tabela';
2 select max('Coluna') from 'Tabela' where ano = '2016';
```

Exemplo1 (Ele retorna o maior valor dessa sua coluna) Exemplo2 (Ele retorna o maior valor do ano de 2016)

### Min

```
1 select min(*) from 'Tabela';
2 select min('Coluna') from tabela where ano = '2016';
```

Exemplo1 (Ele retorna o menor valor da tabela) Exemplo2 (Ele retorna o menor valor de 2016)

### Sum

```
1 select sum('Coluna') from 'Tabela';
2
3 select sum('Coluna') from 'tabela' where ano = '2016'; `Ele somaria o
total se aulas em 2016`
```

Exemplo1 (sua coluna seria de aula, então com esse comando ele somaria o total de aula que você realizou) Exemplo2 (Ele somaria o total se aulas em 2016)

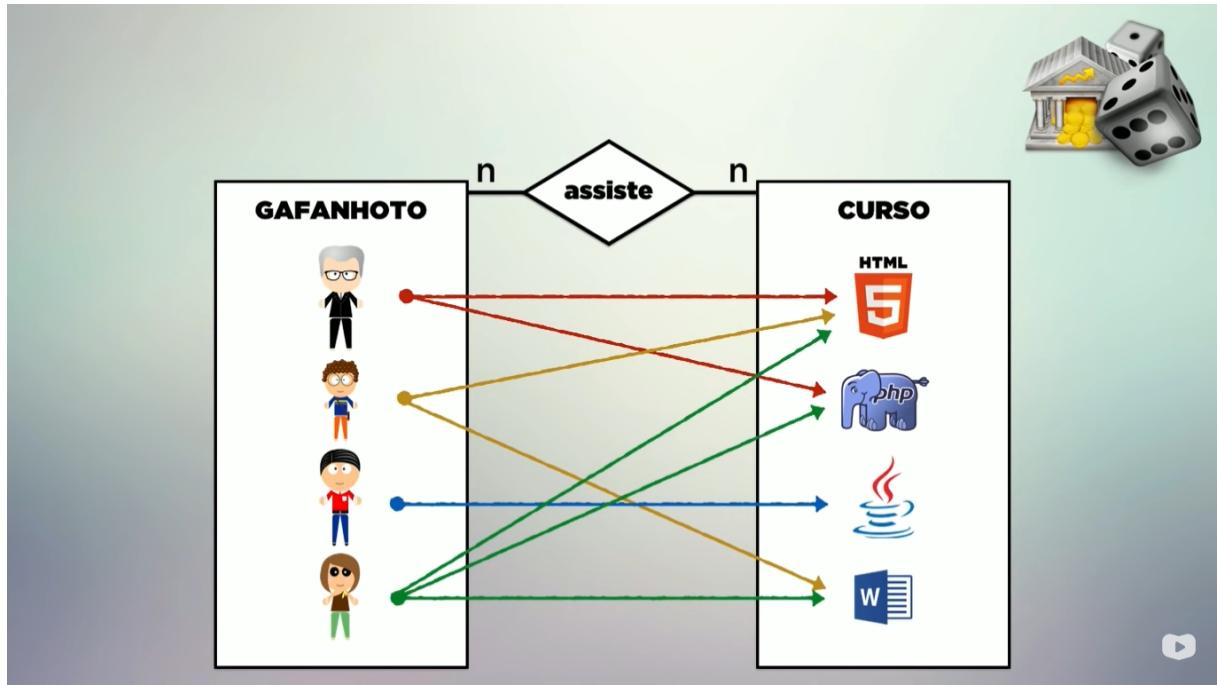
### AVG

```
1 select avg('Coluna') from 'tabela' where 'coluna' = '2016';
```

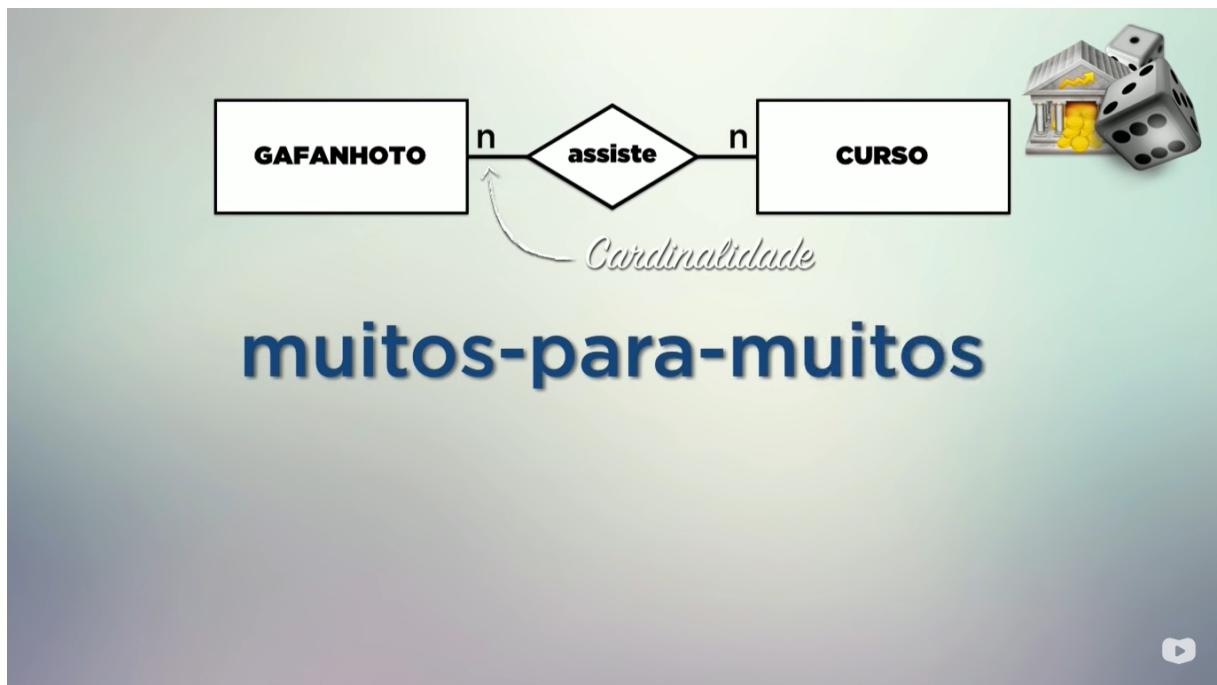
Ele vai soma e dividir pelo tanto de valores que apareceu

## 2.8 Modelo Relacional

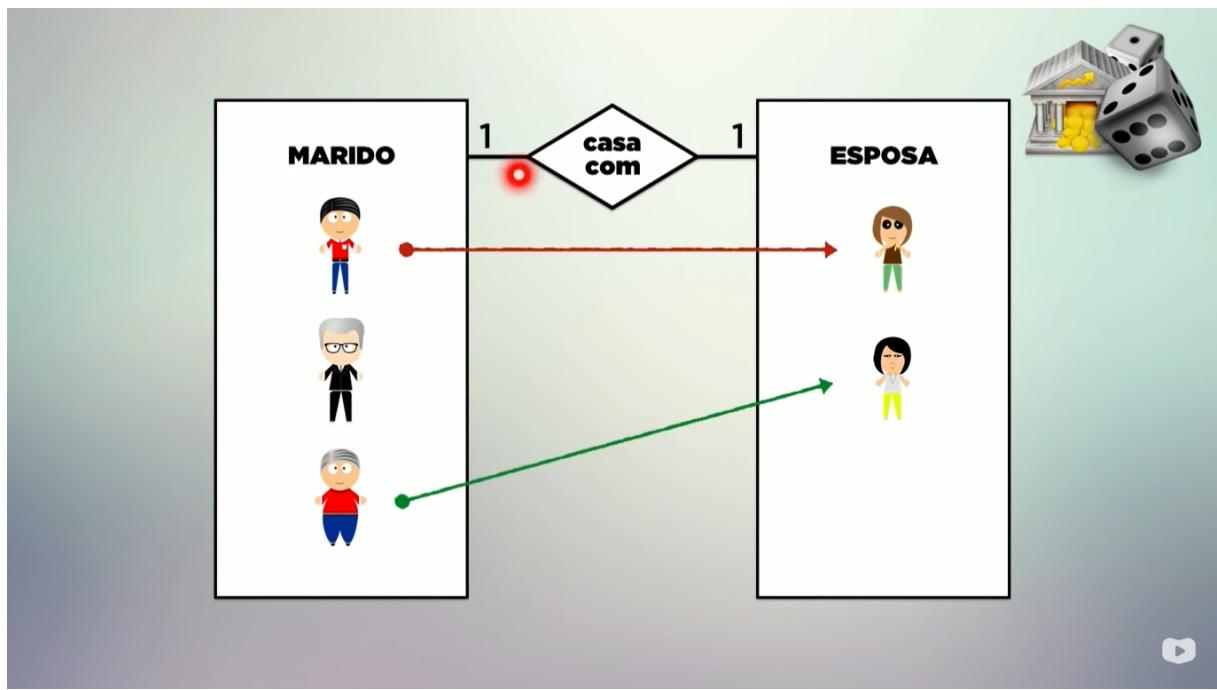
- Cardinalidade
- Relacionamento



**Figura 4:** exemplo1



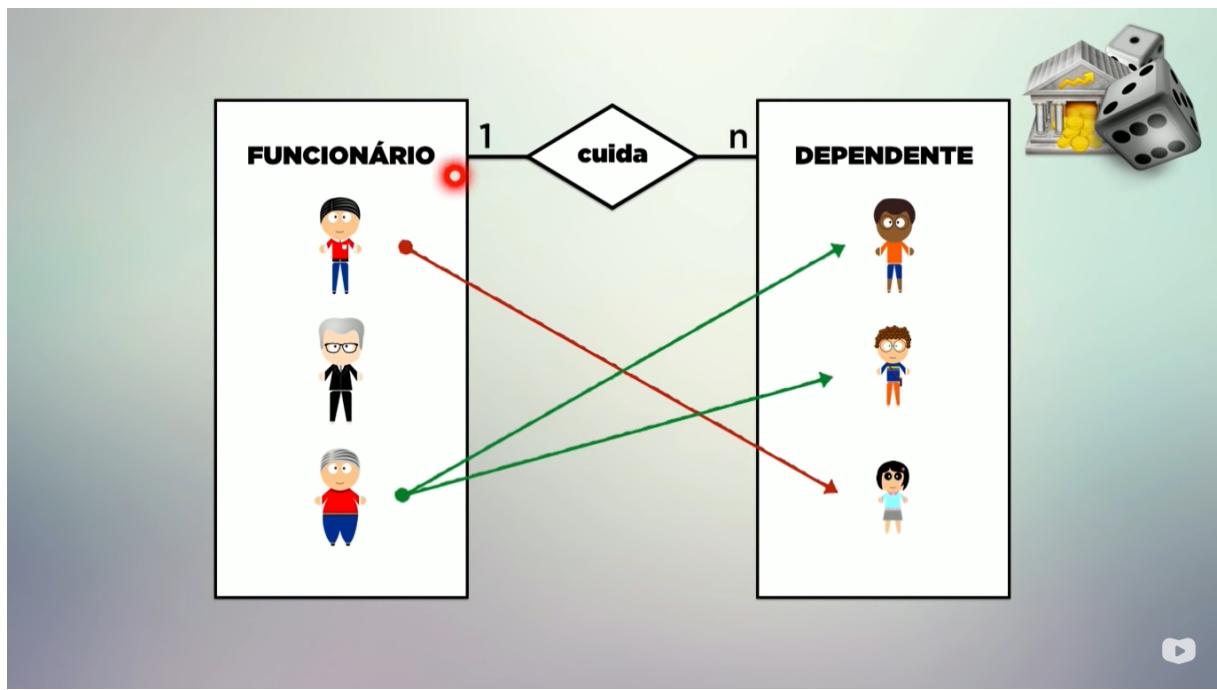
**Figura 5:** Exemplo2



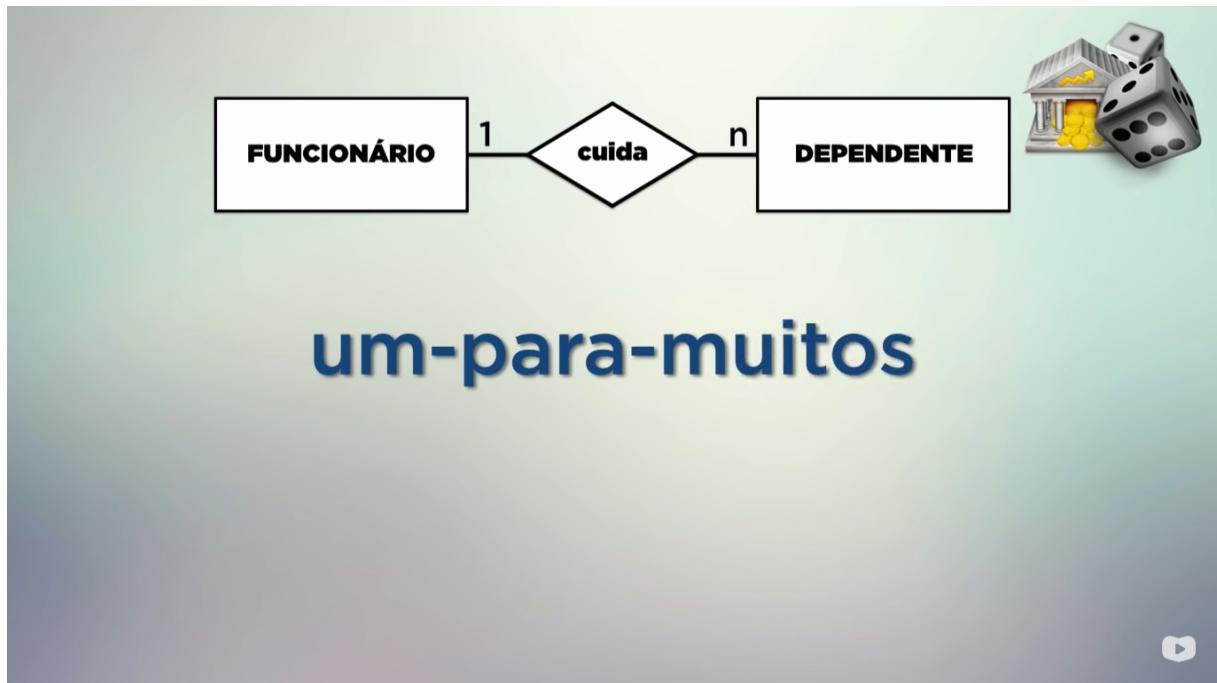
**Figura 6:** Exemplo3



**Figura 7:** Exemplo4



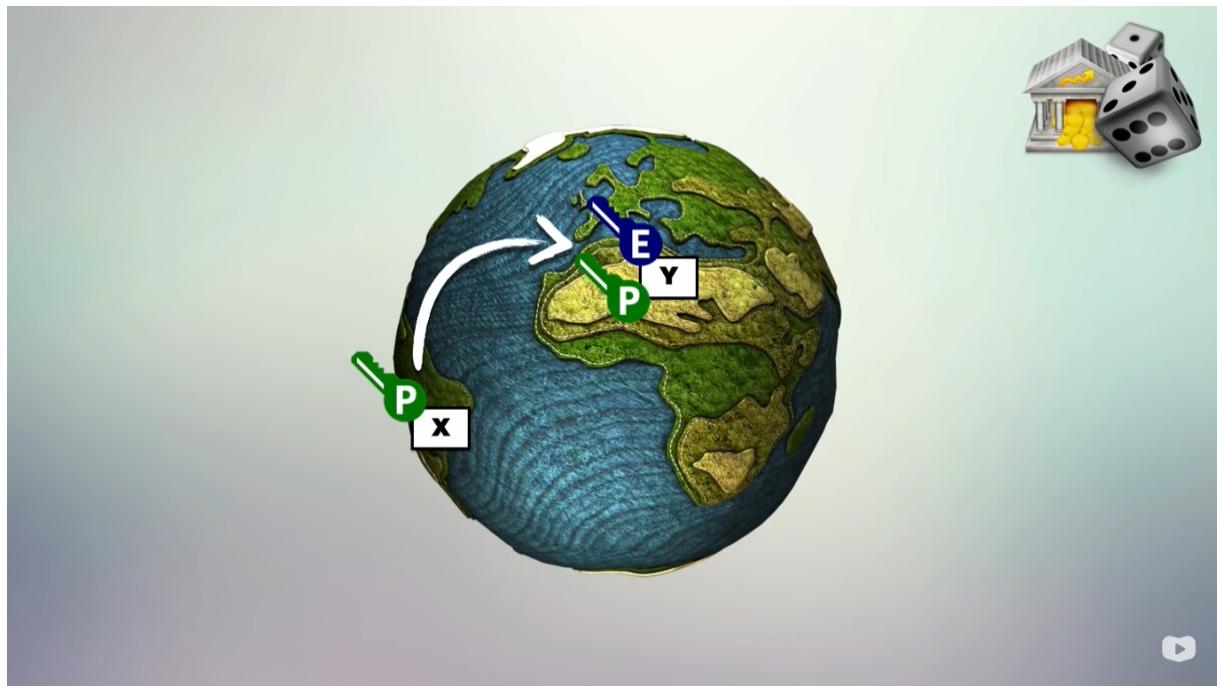
**Figura 8:** Exemplo5



**Figura 9:** Exemplo6

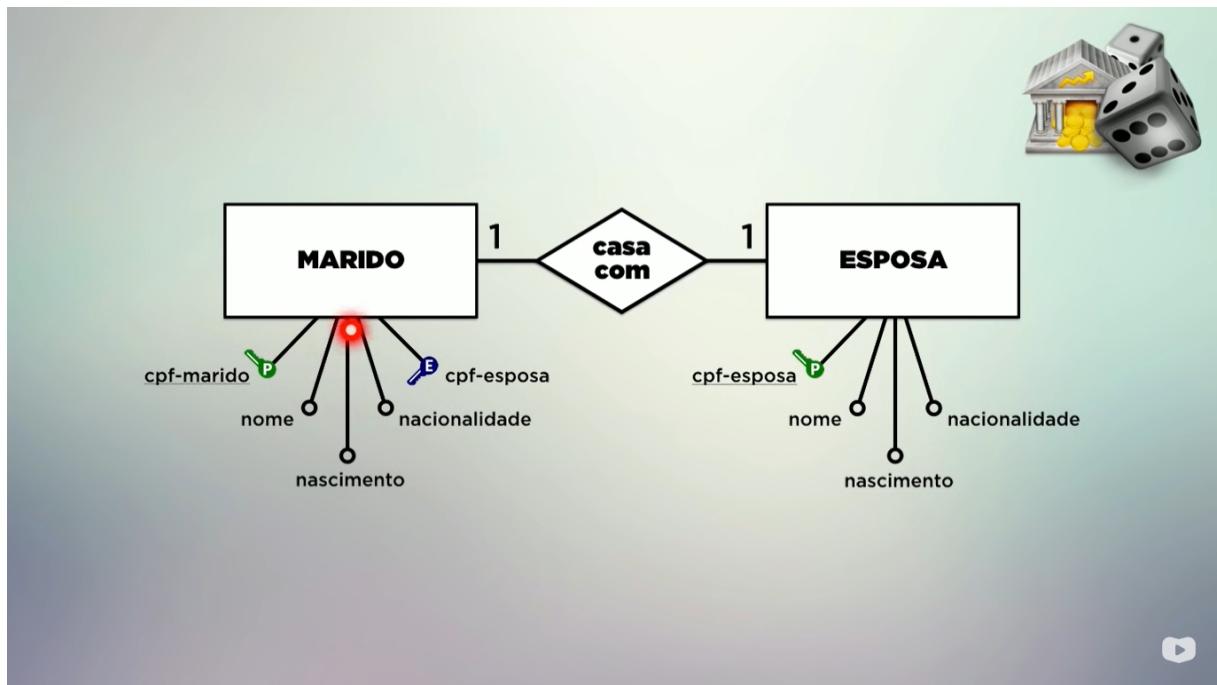
## 2.9 Chave estrangeira

- seria uma chave de outro container(exemplo: Brasil tem uma chave primaria e estados unidos também tem uma chave primaria mas eu quero levar a chave primaria do brasil para os estados unidos ao invés de ela chega com chave primaria ela chega com chave estrangeira)



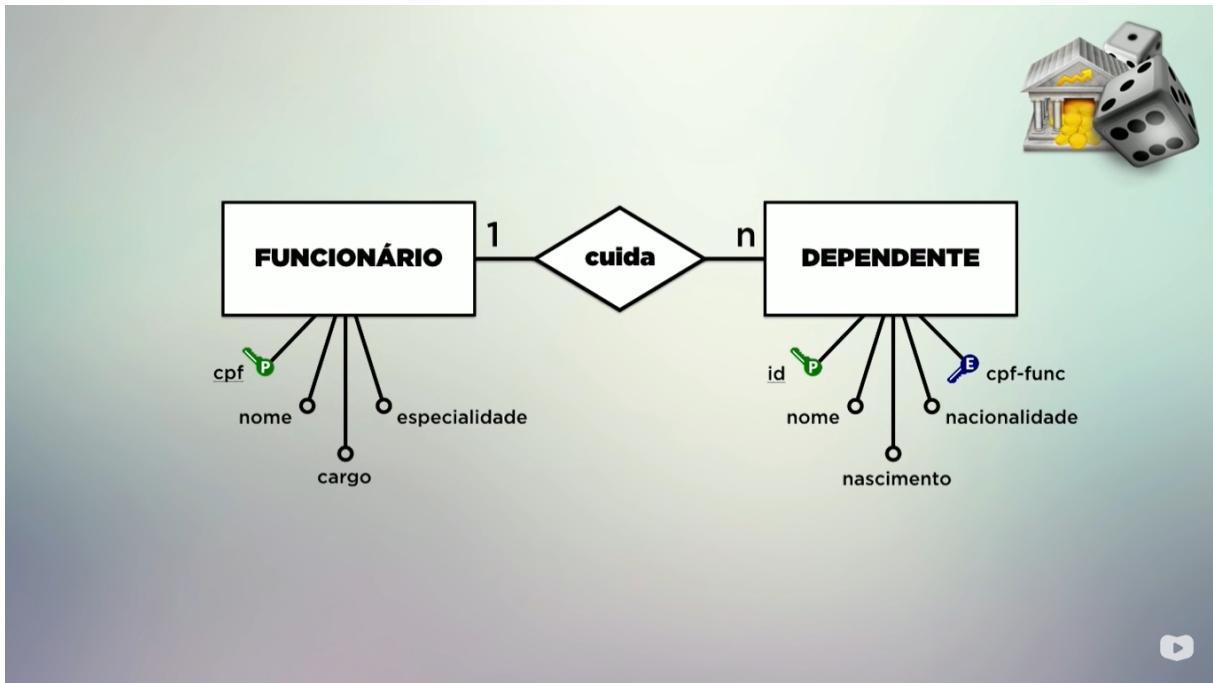
**Figura 10:** chave estrangeira

**Relacionamento de um-para-um** utilizando chave estrangeira



**Figura 11:** um-para-um

**Relacionamento de um-para-muitos** utilizando chave estrangeira obs: quando é um para muitos, sempre coloca a chave estrangeira no muitos



**Figura 12:** um-para-muitos

### Comandos Workbench

```

1 alter table gafanhotos
2 add column 'Nome da coluna estrangeira' int;
3
4 alter table gafanhotos
5 add foreign key (Nome da coluna estrangeira)
6 references 'nome da tabela que você quer exporta a chave' ('Nome da
    coluna');

```

# Adicionando a Foreign Key



```
ALTER TABLE gafanhotos
ADD COLUMN cursopreferido int;

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

Todos os direitos de reprodução e distribuição reservados ao site [CursoemVideo.com](#)

**Figura 13:** Comando

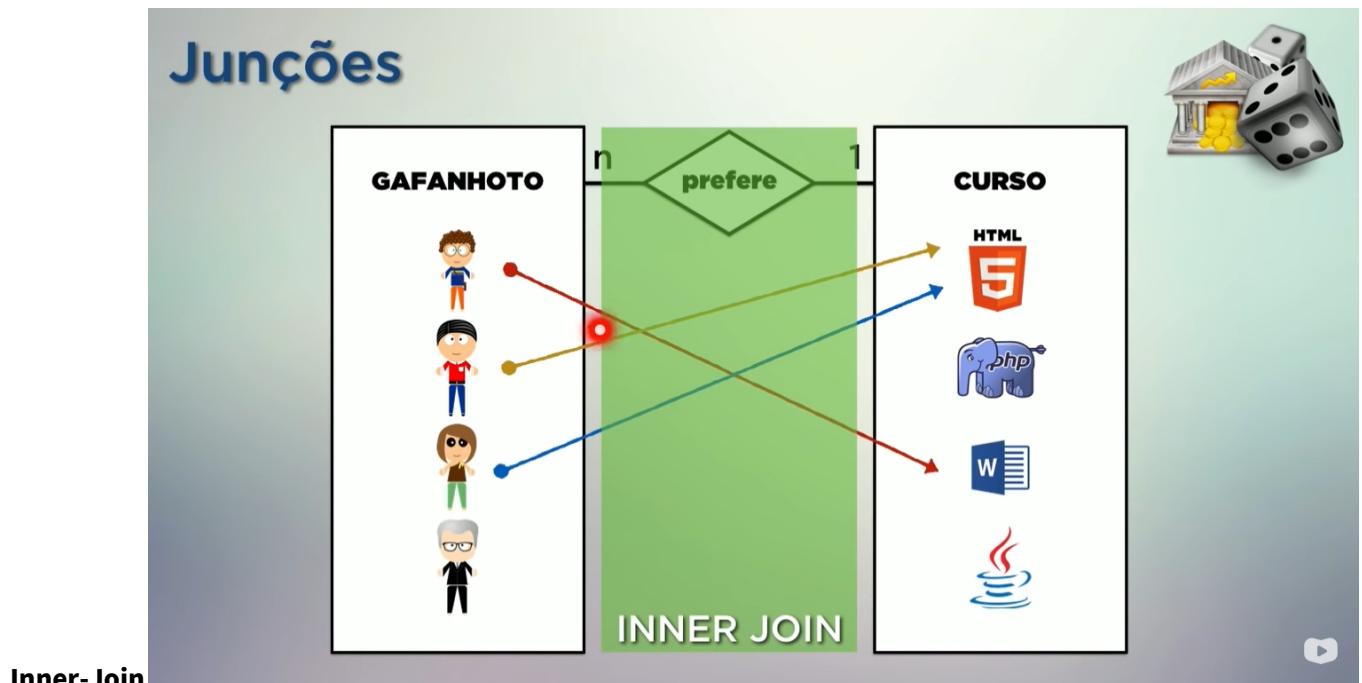
Como adiciona que tal pessoa prefere isso da primeira tabela e da outra(Exemplo: Tabela gafanhotos e tabela cursos, jose prefere o php)

#### Na pratica

```
1 update gafanhotos set 'Nome da coluna adicionada a chave estrangeira' =
   '5' where id = '2';
2
3 Comando para adicionar um valor de outra na tabela na sua atual
```

*Obs:* você não consegue deletar um valor da sua tabela estrangeira , pós ela esta relacionada a outra tabela

isso se chama integridade referencial.

**Inner-Join**

*filtrar os valores*

```

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

**Outer-Join**

*Mostrar todos os valores da tabela*

```

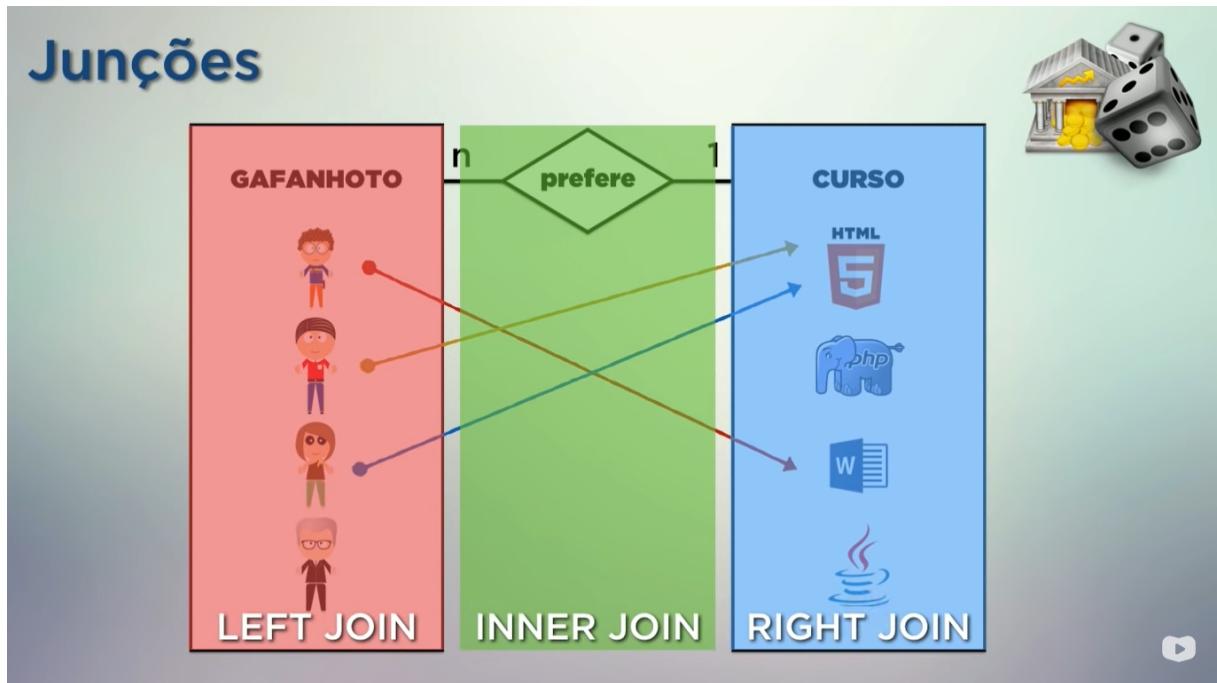
1 select g.nome, c.nome, c.ano
2 from gafanhotos as g left join cursos as c
3 on c.idcurso = g.cursopreferido;
```

**Left ele da preferencia da tabela gafanhotos**

```

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

**Right ele da preferencia da tabela cursos**

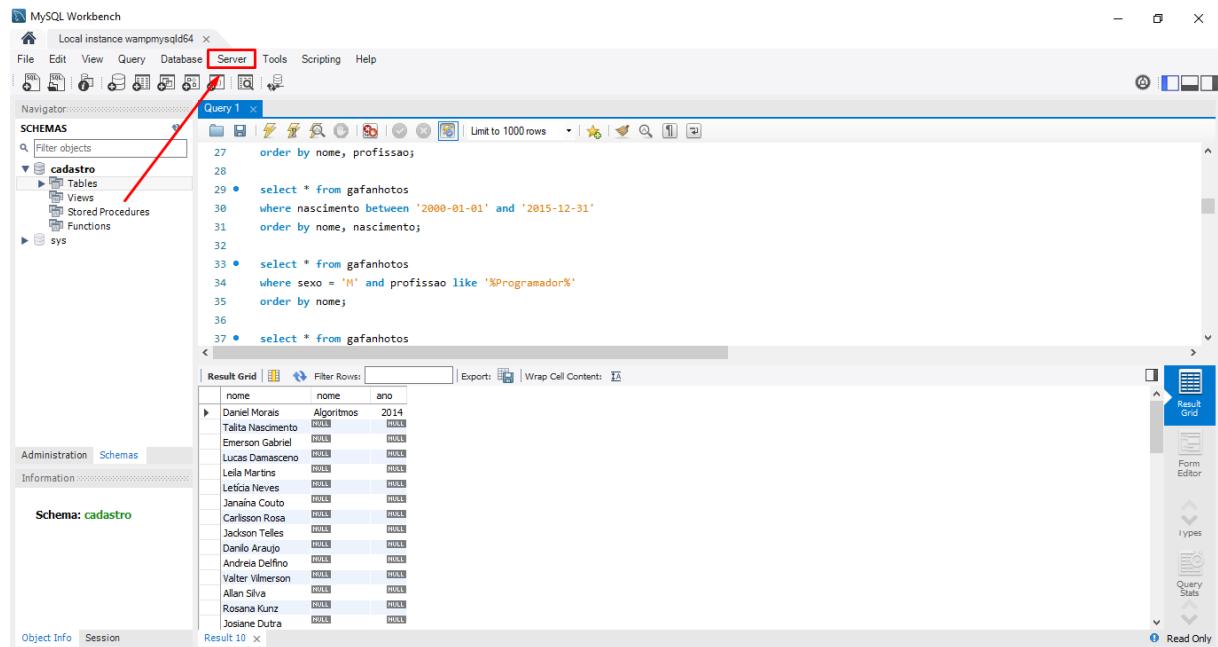
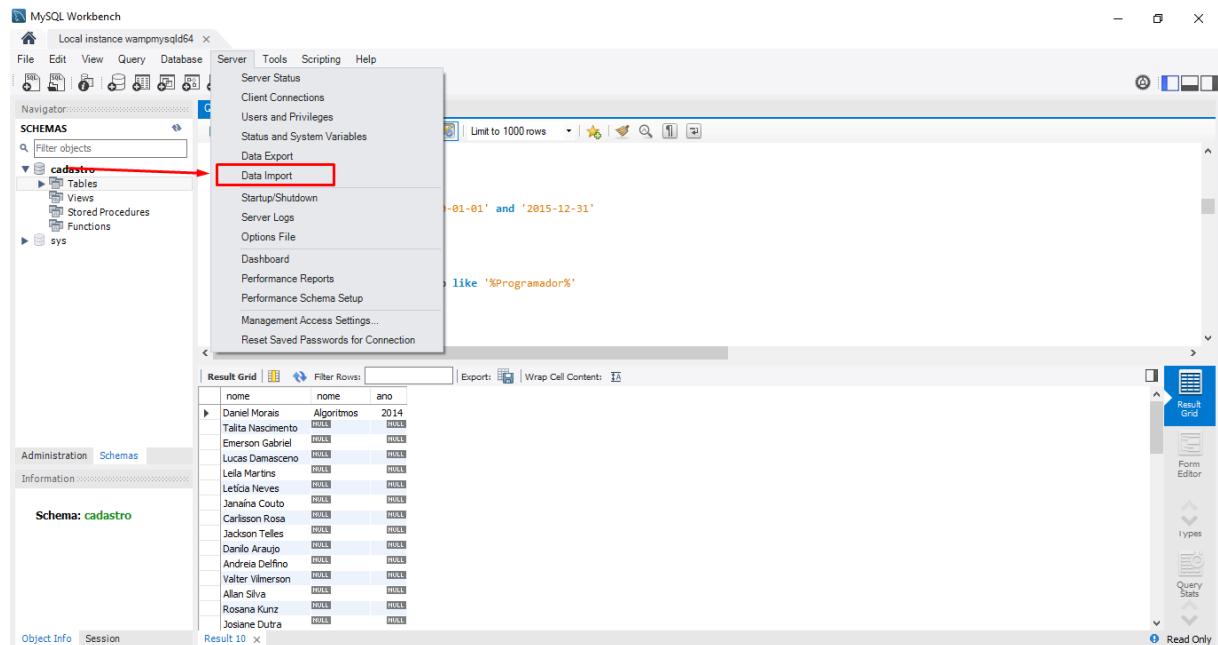


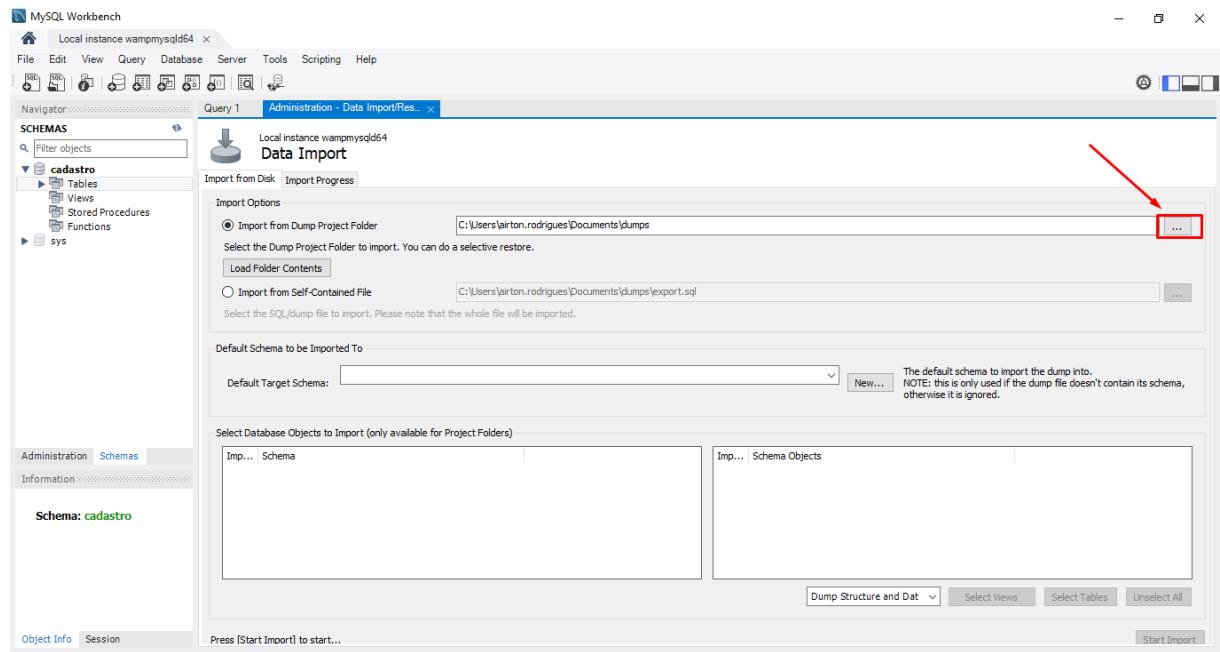
**Figura 14:** todos-os-join

## 2.10 Baixar Banco de dados para estudo

### 2.10.1 Click Aqui para baixar banco de dados

**como fazer a importa o banco de dados**

**Figura 15:** Tela\_inicial**Figura 16:** Tela\_intermediaria



**Figura 17:** Tela\_Final

### **3 Change log**

#### **3.1 version 1.0.0**

- Guia para iniciante.