

CURSO MYSQL

Curso em Vídeo (Gustavo Guanabara)

https://www.youtube.com/watch?v=Ofktsne-utM&list=PLHz_AreHm4dkBs-795Dsgvau_ekxg8g1r

Resumo do curso feito por Roberto Pinheiro

COMANDO DDL - DATA DEFINITION LANGUAGE

Criando um banco de dados

CREATE DATABASE cadastro *Comando para a criação de um banco de dados*

DEFAULT CHARACTER SET utf8 *Parâmetro para uso de caracteres especiais brasileiros, assim como no html.*

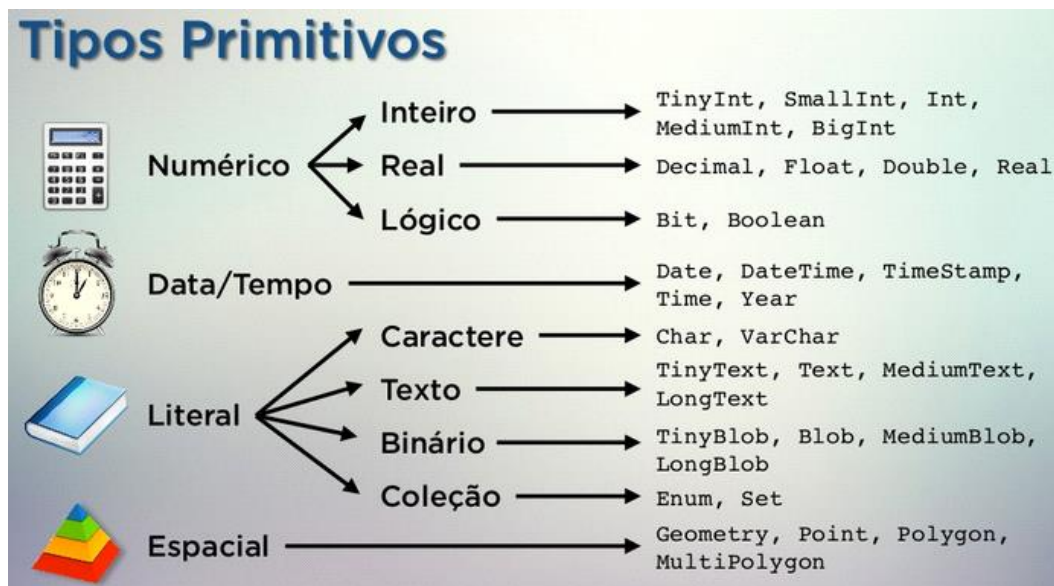
DEFAULT COLLATE utf8_general_ci; *Mesma lógica do parâmetro acima. Não esquecer do ; para fechar um comando.*

Abrindo um banco de dados

USE cadastro; *Caso um banco de dados esteja em outro arquivo.*

Tipos Primitivos de Dados

Especificação do tipo de dado a ser inserido, se é número, palavras, datas, etc.



Criando tabela

CREATE TABLE pessoas (*Comando para criação de uma tabela.*

nome VARCHAR(30) NOT NULL, *Criação de um dado dentro da tabela. Varchar(30) indica palavras de 30 caracteres no máximo. Not null indica que o campo não pode ser deixado em branco.*

sexo ENUM('M', 'F'), *Enum indica que conterà o armazenamento de um único caracter, nesse caso "M" ou "F".*

peso DECIMAL(5,2), *Decimal indica que será um número com cinco casas decimais, e com duas após a vírgula.*

altura DECIMAL(3,2), *Mesma lógica usada acima. (Ex: 2,20)*

nacionalidade VARCHAR(20) DEFAULT 'Brasil' *O uso de default "Brasil" indica que caso não seja digitado nada, será considerado Brasil a nacionalidade.*

) DEFAULT CHARSET=UTF8; *Fim do comando indicando o uso dos caracteres especiais UTF-8.*

Apagando uma tabela

DROP TABLE pessoas; *Comando para apagar tabelas.*

Inserindo chave primária

CREATE TABLE pessoas (

id int NOT NULL AUTO_INCREMENT, *Esse comando sinaliza que a criação de dados é única, não se repetindo, considerando que várias pessoas vão ser registradas. Assim será incrementado a inserção de novas pessoas. Sendo o primeiro 1, o segundo 2, e assim sucessivamente.*

nome VARCHAR(30) NOT NULL,

nascimento DATE,

sexo ENUM('M', 'F'),

peso DECIMAL(5,2),

altura DECIMAL(3,2),

nacionalidade VARCHAR(20) DEFAULT 'Brasil',

PRIMARY KEY (id) *Indicando que o campo id criado é uma primary key, ou seja, não se repete.*

) DEFAULT CHARSET=UTF8;

Inserindo dados na tabela

insert into pessoas *Comando para inserir os dados na tabela pessoas...*

(id, nome, nascimento, sexo, peso, altura, nacionalidade) *Colocar o nome dos campos para dizer onde os dados na tabela serão inseridos.*

values *...continuação do comando.*

(default, 'Godofredo', '1984-01-02', 'M', '78.5', '1.83', 'Brasil');

^ O default está relacionado ao auto increment na criação da tabela, não preciso dizer quem é a nova pessoa cadastrada, o sistema já vai fazer isso pra mim.

select * from pessoas; *Comando para selecionar todas as entradas de pessoas.*

insert into pessoas values *Comando para inserir dados na tabela sem a utilização dos nomes dos campos, considerando que os dados a serem inseridos estão na mesma ordem dos campos.*

(default, 'Godofredo', '1984-01-02', 'M', '78.5', '1.83', 'Brasil');

insert into pessoas *Forma de utilizar o comando para inserir dados de várias pessoas*

(id, nome, nascimento, sexo, peso, altura, nacionalidade)

values

(default, 'Ana', '1975-12-22', 'F', '52.3', '1.45', 'EUA'), *atenção para a vírgula separando os dados a serem inseridos.*

(default, 'Pedro', '2000-07-15', 'M', '52.3', '1.45', 'Brasil'),

(default, 'Maria', '1999-05-30', 'F', '75.9', '1.70', 'Portugal'); *apenas o último recebe o ; já que é o fim do comando.*

select * from pessoas;

Modificando a estrutura da tabela

COMANDO DDL - DATA DEFINITION LANGUAGE

Adicionando uma nova coluna

alter table pessoas *Comando para alterar os dados da tabela que foi criada*

add column profissao varchar(10); *Adicionando uma nova coluna, ou um novo campo dentro da tabela pessoas.*

describe pessoas; *Comando para visualizar a tabela pessoas.*

*Digitar: select * from pessoas; -- e executar para ele mostrar em formato convencional de tabela, mostrando os campos como colunas. E por enquanto, a profissão de cada pessoa estará nula.*

Removendo uma coluna

alter table pessoas *Novamente o comando para alterar dados da tabela*

drop column profissao; *Excluindo a coluna profissao.*

describe pessoas;

Escolhendo posição para adicionar uma nova coluna

alter table pessoas

add column profissao varchar(10) after nome; *Com esse parâmetro eu indico a posição específica onde quero colocar essa nova coluna, nesse caso irá ficar depois de nome*

describe pessoas;

alter table pessoas

add column codigo int first; *Como não existe um parâmetro para colocar antes de uma determinada coluna, caso eu queira colocar uma coluna nova antes da primeira coluna, usar: first. Caso eu não digite nada, será considerado como a última posição.*

describe pessoas;

Modificando definições

alter table pessoas

modify column profissao varchar(20); *Posso modificar o tipo primitivo da coluna, ou, como nesse caso, estou modificando a quantidade de caracteres. Deve ser colocado no fim do comando: not null default ""; -- Nesse caso será considerado como texto vazio, mas não*

describe pessoas; *campo vazio, não causando conflito interno, para o caso de eu delimitar que os campos não podem ser vazios.*

Renomeando coluna

alter table pessoas

change column profissao prof varchar(20); *Uso esse comando para mudar o nome da coluna, mas além disso tbm posso fazer as outras modificações de modify.*

describe pessoas;

Renomeando tabela

alter table pessoas

rename to gafanhotos; *Comando para modificar o nome de toda a tabela. Dar refresh para aparecer a tabela com o novo nome.*

describe gafanhotos;

Mais uma tabela

CREATE TABLE IF NOT EXISTS cursos (*if not exists -- Indicação de que só irá ser criado essa nova tabela se já não existir outra tabela com o mesmo nome.*
nome VARCHAR(30) NOT NULL UNIQUE, *Não confundir com o varchar, Não estou dizendo que é uma chave primária. Só não vai deixar já que o text é para textos longos. criar duas tabelas com o mesmo nome.*
descricao TEXT,
carga int UNSIGNED, *Unsigned significa sem sinal, ou seja, não é pra considerar sinais negativos ou qualquer outro.*
totaulas int,
ano year DEFAULT '2016' *Caso não tenha ano, o padrão será 2016.*
) DEFAULT CHARSET=UTF8;

describe cursos;

Adicionando chave primária após a tabela ter sido criada

alter table cursos

add column idcurso int first; *Usando a mesma lógica de modificação de tabela, estou adicionando uma coluna de chave primária na primeira posição de colunas.*

alter table cursos

add primary key(idcurso); *Adicionando a primary key em idcurso, agora cada inserção será desvinculada de outras, como visto anteriormente. Esse processo só foi feito pq já tínhamos criado a tabela.*

describe cursos;

COMANDO DDL - DATA DEFINITION LANGUAGE

Apagando a tabela

drop table cursos;

Tipos de comandos



```
select * from gafanhotos;
```

```
select * from cursos;
```

Inserindo dados na tabela cursos

```
use cadastro;
```

```
insert into cursos values Adicionando os dados na tabela cursos.
```

```
('1','HTML4','Curso de HTML5','40','37','2014'), Importante sempre deixar os dados em aspas simples ou duplas, indicar que são dados.
('2','Algoritmos','Lógica de Programação','20','15','2014'),
('3','Photoshop','Dicas de Photoshop CC','10','8','2014'),
('4','PGP','Curso de PHP para iniciantes','40','20','2010'),
('5','Jarva','Introdução à Linguagem Java','10','29','2000'),
('6','MySQL','Banco de Dados MySQL','30','15','2016'),
('7','Word','Curso completo de Word','40','30','2016'),
('8','Sapateado','Danças rítmicas','40','30','2018'),
('9','Cozinha Árabe','Aprenda a fazer kibe','40','30','2018'),
('10','YouTuber','Gerar polêmica e ganhar inscritos','5','2','2018');
```

```
select * from cursos;
```

Cada linha é o equivalente a um registro, os campos são as colunas.

Modificando 1 linha incorreta

COMANDO DML - DATA MANIPULATION LANGUAGE

update cursos *O comando update são para modificar as linhas, o comando alter table são pra modificar as colunas.*

set nome='HTML 5' *Estou modificando o nome para "Html5" onde o idcurso é 1.*

where idcurso='1';

select * from cursos;

update cursos

set nome='PHP', ano='2015' *Estou modificando o nome para "PHP" e o ano para "2015" onde o idcurso é 4.*

where idcurso='4';

select * from cursos;

update cursos

set nome='Java', ano='2015', carga='40' *Estou modificando o nome para "Java", o ano para "2015" e a carga para 40 onde o idcurso é 5.*

where idcurso='5'

limit 1; *Ele está limitando a modificação para uma única linha.*

select * from cursos;

Modificando mais de 1 registro

No menu "Edit" do MySQL Workbench acessar "Preferences" e na aba "SQL

Editor" desabilitar a caixa "Safe Updates" *Proteção por padrão está ativada, desativar apenas para entender o funcionamento do que vem a seguir.*

update cursos

set ano='2050', carga='800'

where ano='2018'; *Aqui ele vai modificar os anos e cargas de diversas linhas, o que é extremamente perigoso e não aconselhável.*

select * from cursos;

update cursos

set ano='2050'

where ano='2018';

select * from cursos;

Modificando apenas 1 registro de vários existentes

```
update cursos  
set ano = '2018', carga = '0'  
where ano = '2050'  
limit 1; Agora com a limitação, ele vai alterar a primeira linha apenas.
```

```
select * from cursos;
```

Deletando 1 registro

COMANDO DML - DATA MANIPULATION LANGUAGE

```
delete from cursos Comando para apagar uma linha de cursos, onde o idcursos é 8.  
where idcurso='8';
```

```
select * from cursos;
```

Deletando mais de 1 registro

```
delete from cursos  
where ano='2050' Aqui estou dizendo pra apagar todas as linhas onde o ano é 2050.  
limit 2; Porém como eu limitei, só vai apagar as duas primeiras linhas.
```

```
select * from cursos;
```

Apagando todas as linhas de uma tabela

COMANDO DML - DATA MANIPULATION LANGUAGE

```
truncate table cursos; Remove todas as linhas, mas não remove a tabela em si.
```

ou

```
truncate cursos; Mesma lógica do que foi digitado acima.
```


Tipos de dados



Novamente, DDL modifica a estrutura e DML modifica os dados.

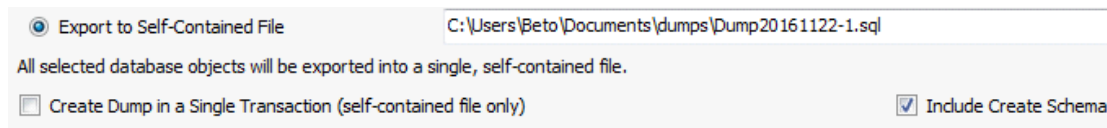
Gerando cópias de segurança

- No menu "Server" do MySQL Workbench acessar "Data export"

O backup de um banco de dados chama-se dump.

- Para gerar o backup (Server --> Data Export --> Dump Structure and Data)

- Selecionar a opção "Export to Self-Contained File" e marcar a caixa "Include Create Schema", como indicado abaixo:



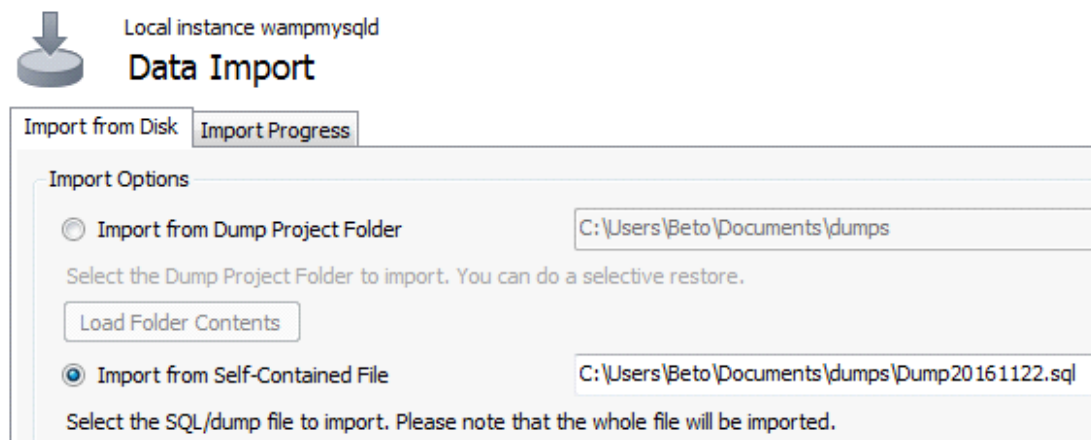
- Clicar no botão "Start export"

- Para testar, após gerar o backup:

drop database cadastro;

drop database test;

- Para restaurar o backup (Server --> Data Import)



- Clicar em "Start Import"

Console MySQL

Usuário: root

Senha:

Comandos

show databases;

Apresenta os bancos de dados

status;

Verifica qual é o banco de dados ativo no momento.

use cadastro;

Abre o banco de dados "cadastro".

show tables;

Apresenta as tabelas do banco de dados ativo.

describe cursos;

Apresenta a estrutura da tabela "cursos".

select * from gafanhotos;

Lista os registros da tabela "gafanhotos".

update cursos set nome = 'Ph' where idcurso = 4;

Na tabela "cursos", altera o conteúdo do campo nome para "Ph", no registro 4.

alter table 'gafanhotos' drop 'codigo';

Exclui o campo "codigo" da tabela "ganhotos".

show create table amigos;

Mostra os comandos usados para criar a tabela "amigos".

show create database exemplo;

Mostra os comandos usados para criar o banco de dados "exemplo".

Usando o comando select

Selecionando colunas (campos)

select * from cursos;

Seleciona todos os campos (colunas) da tabela "cursos" por ordem do campo que contém a chave primária.

**select * from cursos
order by nome;**

ou

**select * from cursos
order by nome asc;**

Seleciona todos os campos (colunas) da tabela "cursos" por ordem crescente do campo "nome".

**select * from cursos
order by nome desc;**

Seleciona todos os campos (colunas) da tabela "cursos" por ordem decrescente do campo "nome".

**select nome, carga, ano from cursos
order by nome;**

Seleciona os campos nome, carga e ano (nessa ordem) da tabela "cursos" por ordem crescente do campo "nome".

**select ano, nome, carga from cursos
order by ano, nome;**

Seleciona os campos ano, nome e carga (nessa ordem) da tabela "cursos" por ordem crescente do campo "ano" e depois por ordem de "nome".

Selecionando linhas (registros)

Clausula where

```
select * from cursos  
where ano='2016'  
order by nome;
```

Seleciona todos os campos (colunas) da tabela "cursos", cujo campo ano contenha '2016' e por ordem crescente do campo "nome".

```
select nome, descricao, ano from cursos  
where ano<=2015  
order by ano, nome;
```

Seleciona os campos "nome" e "descricao" da tabela "cursos", com "ano" menor ou igual a 2015, por ordem crescente de "ano" e depois de "nome".

Operador between

```
select nome, ano from cursos  
where ano between 2014 and 2016  
order by ano desc, nome;
```

Seleciona os campos "nome" e "ano" da tabela "cursos", com "ano" entre 2014 e 2016, por ordem decrescente de ano e depois por "nome".

Operador in

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

Seleciona os campos "nome", "descricao" e "ano", com "ano" 2014 ou 2016, por ordem de "ano".

Clausula and

**select nome, carga, totaulas from cursos
where carga > 35 and totaulas < 35;**

Seleciona os campos "nome", "carga" e "totaulas" com carga maior que 35 e total de aulas menor que 30.

Clausula or

**select nome, carga, totaulas from cursos
where carga > 35 or totaulas < 35;**

Seleciona os campos "nome", "carga" e "totaulas" com carga maior que 35 ou total de aulas menor que 30.

Tipos de comandos



Operador like

O operador "like" não é case sensitive.

```
select * from cursos  
where nome like 'P%';
```

Seleciona todos os registros cujo nome começa com a letra P.

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

Seleciona todos os campos da tabela "cursos" cujo nome termina com "a".

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

Seleciona todos os campos da tabela "cursos" cujo nome contenha a letra "a" em qualquer posição do nome.

```
select * from cursos  
where nome not like '%a%';
```

Seleciona todos os campos da tabela "cursos" cujo nome não contenha a letra "a" em nenhuma posição do nome.

```
select * from cursos
```

where nome like 'ph%p';

Seleciona todos os campos da tabela "cursos" cujo nome comece com "ph" e termine com "p".

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

Seleciona todos os campos da tabela "cursos" cujo nome comece com "ph" e termine com "p" seguido de qualquer outro caractere.

Clausula distinct

**select distinct nacionalidade from gafanhotos
order by nacionalidade;**

Seleciona na tabela "gafanhoto" apenas as nacionalidades existentes, sem repetição, por ordem crescente de nacionalidade.

Funções de agregação

count()

```
select count(*) from cursos;
```

Conta o total de cursos existentes na tabela "cursos".

```
select count(*) from cursos  
where carga > 40;
```

Conta, na tabela "cursos", quantos possuem carga superior a 40 horas.

max()

```
select max(carga) from cursos;
```

Apresenta qual é a maior carga horária dentro da tabela "cursos".

```
select max(totaulas) from cursos  
where ano='2016';
```

Apresenta qual é a maior carga horária dentro da tabela "cursos" no ano de 2016.

min()

```
select min(totaulas) from cursos;
```

Apresenta o menor total de aulas existentes dentro da tabela "cursos".

```
select nome, min(totaulas) from cursos  
where ano='2016';
```

Apresenta o nome do curso e o número de aulas do curso de 2016 com o menor número de aulas.

sum()

```
select sum(totaulas) from cursos  
where ano='2016';
```

Apresenta a somatória do total de aulas, da tabela cursos, no ano de 2016.

avg()

```
select avg(totaulas) from cursos  
where ano='2016';
```

Apresenta a média de aulas, da tabela cursos, no ano de 2016.

Exercícios

1) Faça uma lista com o nome de todas as gafanhotos.

```
select nome from gafanhotos  
where sexo = 'F';
```

2) Faça uma lista com os dados de todos aqueles que nasceram entre 1/Jan/2000 e 31/Dez/2015.

```
select * from gafanhotos  
where nascimento between '2000-01-01' and '2015-12-31';
```

3) Faça uma lista com o nome de todos os homens que trabalham como programadores.

```
select nome, profissao from gafanhotos  
where profissao = 'Programador';
```

4) Faça uma lista com os dados de todas as mulheres que nasceram no Brasil e que tem seu nome iniciando com a letra "J".

```
select * from gafanhotos  
where sexo = 'F' and nacionalidade = 'Brasil' and nome like 'J%';
```

5) Faça uma lista com o nome e nacionalidade de todos os homens que tem Silva no nome, não nasceram no Brasil e pesam menos de 100 kg.

```
select nome, nacionalidade from gafanhotos  
where sexo = 'M' and nome like '%_Silva%' and nacionalidade != 'Brasil' and peso < 100;
```

6) Qual é a maior altura entre gafanhotos homens que moram no Brasil?

```
select max(altura) from gafanhotos  
where sexo='M' and nacionalidade = 'Brasil';
```

7) Qual é a média de peso dos gafanhotos cadastrados?

```
select avg(peso) from gafanhotos  
where sexo='M';
```

8) Qual é o menor peso entre as gafanhotos mulheres que nasceram fora do Brasil e entre 01/Jan/1990 e 31/Dez/2000?

```
select min(peso) from gafanhotos  
where sexo='F' and nacionalidade != 'Brasil' and nascimento between '1990-01-01'  
and '2000-12-31';
```

9) Quantas gafanhotos mulheres tem mais de 1.90m de altura?

```
select count(*) from gafanhotos  
where sexo='F' and altura > 1.90;
```

Agrupando e agregando registros

Clausula group by

```
select carga, count(nome) from cursos  
group by carga;
```

Apresenta carga e contagem por nome agrupados por carga.

Clausula having

```
select ano, count(*) from cursos  
group by ano  
having count(ano) >= 5  
order by count(*) desc;
```

Apresenta ano e total de casos agrupados por ano, dos anos com 5 ou mais casos, ordenados em ordem descendente de contagem.

Um select dentro de outro

```
select carga, count(*) from cursos  
where ano > 2015  
group by carga  
having carga > (select avg(carga) from cursos);
```

Apresenta carga e total de casos, onde ano é maior que 2015, agrupados por carga e com carga superior a média de carga dos cursos.

Exercícios

1) Faça uma lista com as profissões dos gafanhotos e seus respectivos quantitativos.

```
select profissao, count(*) from gafanhotos
where sexo = 'M'
group by profissao;
```

2) Quantos gafanhotos homens e quantas mulheres nasceram após 01/Jan/2005?

```
select sexo, count(*) from gafanhotos
where nascimento > '2005-01-01'
group by sexo;
```

3) Faça uma lista com os gafanhotos que nasceram fora do Brasil, mostrando o país de origem e o total de pessoas nascidas lá. Só nos interessam os países que tiverem mais de 3 gafanhotos com essa nacionalidade.

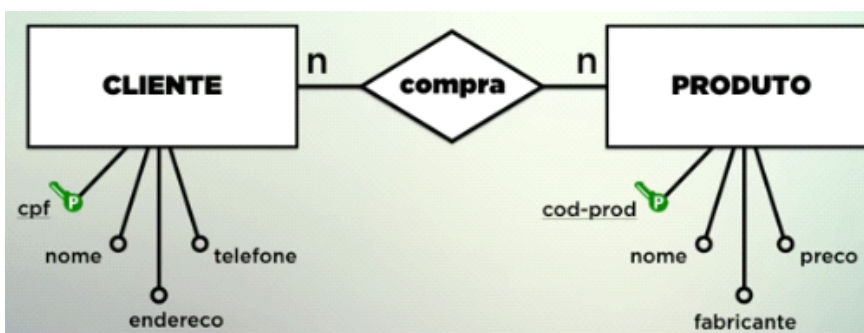
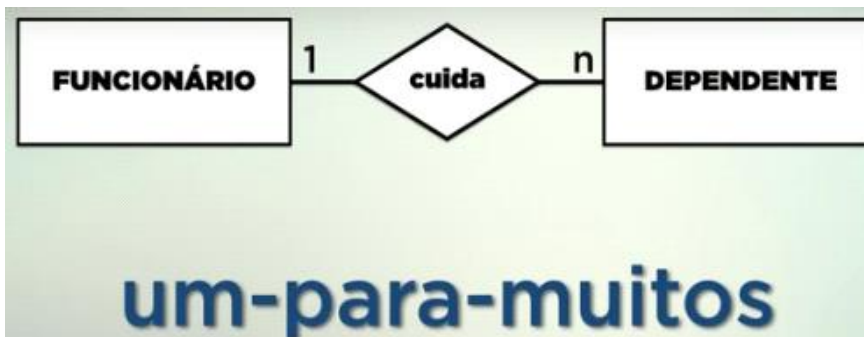
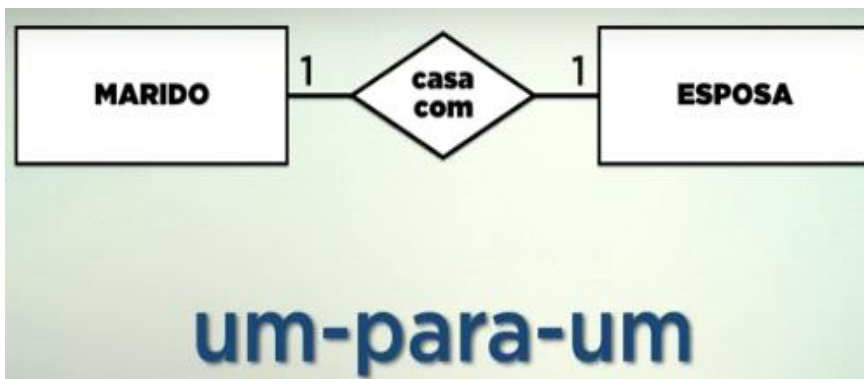
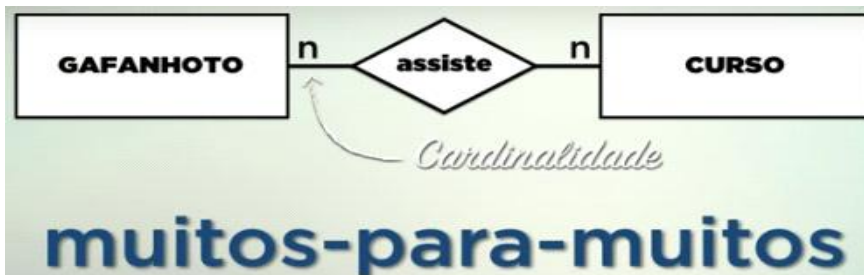
```
select nacionalidade, count(*) from gafanhotos
where nacionalidade != 'Brasil'
group by nacionalidade
having count(nacionalidade) > 3;
```

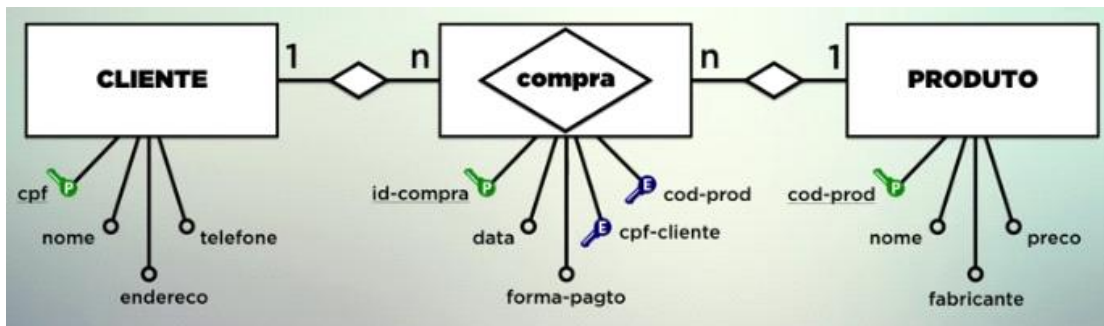
4) Faça uma lista agrupada pela altura dos gafanhotos, mostrando quantas pessoas pesam mais de 100 kg e que estão acima da média de altura de todos os cadastrados.

```
select altura, count(*) from gafanhotos
where peso > 100 and altura > (select avg(altura) from gafanhotos)
group by altura;
```

Modelo relacional (conceitos)

Cardinalidade





Chaves estrangeiras e join - Relacionando as tabelas

Engines

Engines para criar bancos de dados com tabelas compatíveis com chaves estrangeiras:

MyISAM (antigo, não é uma boa opção)

InnoDB

XtraDB

ACID

São as iniciais das quatro principais regras de uma boa transação. Toda a ação que um banco de dados possa realizar é considerada uma transação.

A = Atomicidade (ou tudo acontece ou nada é considerado)

C = Consistência

I = Isolamento (duas ou mais transações podem ser feitas ao mesmo tempo e uma não pode interferir na outra)

D = Durabilidade

Adicionando uma chave estrangeira (foreign key)

use cadastro;

describe gafanhotos;

incluindo a coluna cursopreferido na tabela gafanhotos

```
alter table gafanhotos  
add column cursopreferido int;
```

```
describe gafanhotos;
```

tornando chave estrangeira a coluna cursopreferido

```
alter table gafanhotos  
add foreign key (cursopreferido)  
references cursos (idcurso) ;
```

```
describe gafanhotos;
```

MUL --> chave múltipla (chave estrangeira)

OBS.: a chave estrangeira tem que ser do mesmo tipo da chave primária.

inserindo dados na coluna cursopreferido

```
update gafanhotos set cursopreferido = '6' where id = '1';
```

```
select * from gafanhotos;
```

Integridade referencial

O curso com idcurso 6 não pode ser deletado porque já existe uma relação do curso com um ou mais gafanhotos.

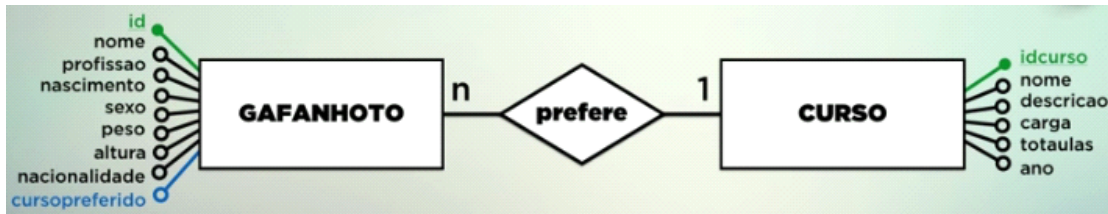
```
delete from cursos  
where idcurso = '6';
```

O curso com idcurso 28 pode ser deletado porque não existe relação do curso com nenhum gafanhoto.

```
delete from cursos  
where idcurso = '28';
```

```
select * from cursos;
```


Trabalhando com junções (inner join e on)



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

o comando inner join lista apenas os gafanhotos que possuem relacionamento com algum curso preferido.

Trabalhando com apelidos (as)

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

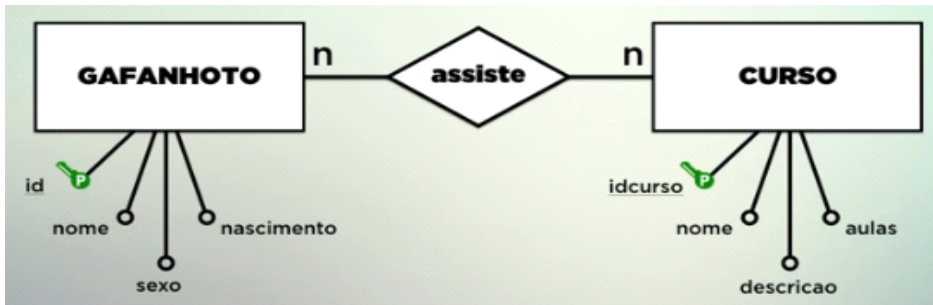
Trabalhando com junções (left outer join e right outer join)

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

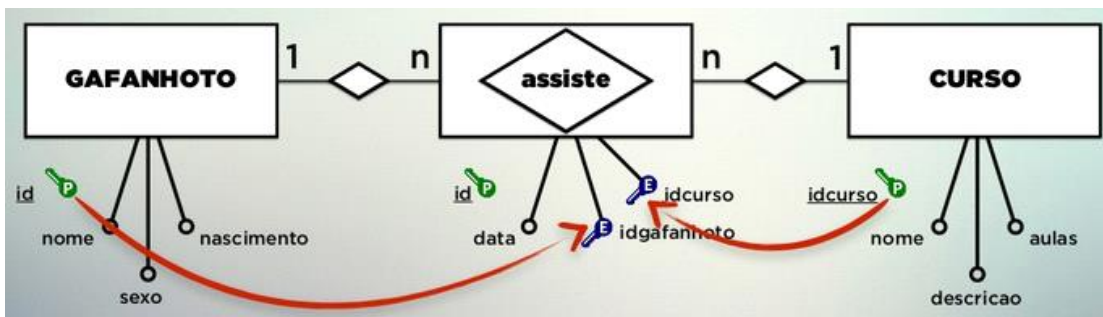
```
select g.nome, g.cursopreferido, c.nome, c.ano
from gafanhotos as g right outer join cursos as c
on c.idcurso = g.cursopreferido;
```

O comando left outer join ou right outer join lista todos os gafanhotos, inclusive os que não possuem nenhum relacionamento (curso preferido).

Relacionando tabelas com cardinalidade muitos para muitos



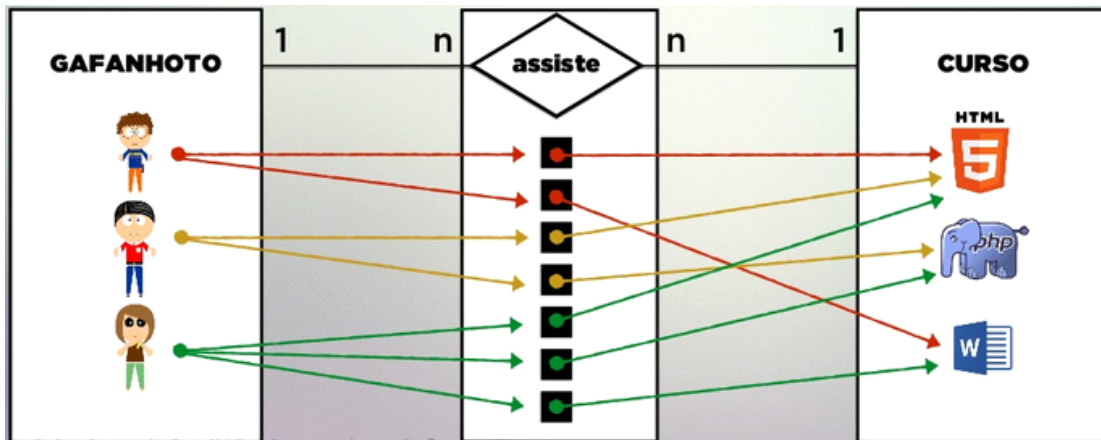
Transformando o relacionamento "assiste" em uma entidade



use cadastro;

```
CREATE TABLE gafanhoto_assiste_curso (  
  id int NOT NULL AUTO_INCREMENT,  
  data date,  
  idgafanhoto int,  
  idcurso int,  
  PRIMARY KEY (id),  
  FOREIGN KEY (idgafanhoto) REFERENCES gafanhotos (id),  
  FOREIGN KEY (idcurso) REFERENCES cursos (idcurso)  
) DEFAULT CHARSET=UTF8;
```

Entendendo a relação



Inserindo registros na tabela gafanhoto_assiste_curso

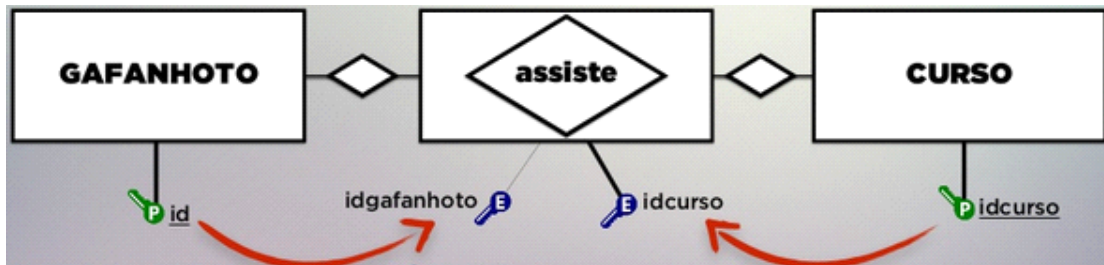
```
INSERT INTO gafanhoto_assiste_curso  
(id, data, idgafanhoto, idcurso) VALUES  
(DEFAULT, '2014-03-01', '1', '2');
```

```
select * from gafanhoto_assiste_curso;
```

```
INSERT INTO `cadastro`.`gafanhoto_assiste_curso`  
(`id`, `data`, `idgafanhoto`, `idcurso`) VALUES  
(2, '2015-12-22', 3, 6);  
INSERT INTO `cadastro`.`gafanhoto_assiste_curso`  
(`id`, `data`, `idgafanhoto`, `idcurso`) VALUES  
(3, '2014-01-01', 22, 12);  
INSERT INTO `cadastro`.`gafanhoto_assiste_curso`  
(`id`, `data`, `idgafanhoto`, `idcurso`) VALUES  
(4, '2016-05-12', 1, 19);
```

	id	data	idgafanhoto	idcurso
	1	2014-03-01	1	2
	2	2015-12-22	3	6
	3	2014-01-01	22	12
▶	4	2016-05-12	1	19
*	NULL	NULL	NULL	NULL

Select com JOIN composto em SQL



```
SELECT g.nome, c.nome FROM gafanhotos g  
JOIN gafanhoto_assiste_curso a  
ON g.id = a.idgafanhoto  
JOIN cursos c  
ON a.idcurso = c.idcurso  
order by g.nome;
```