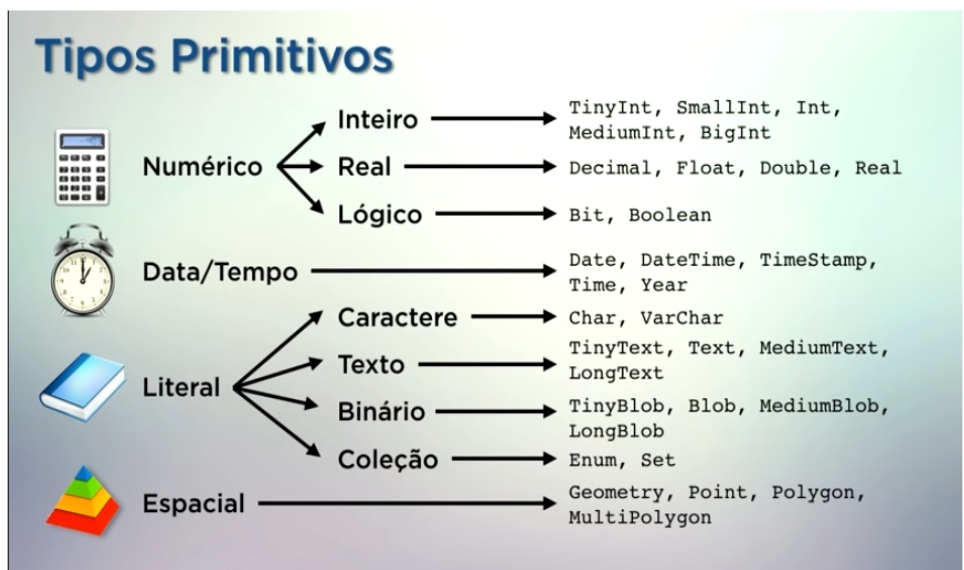
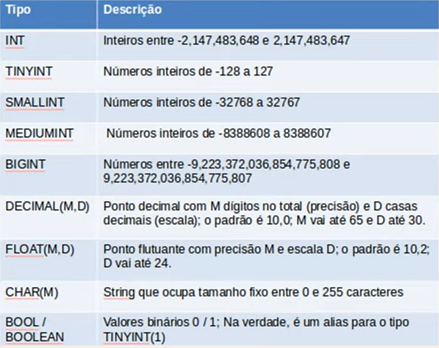
create database cadastro

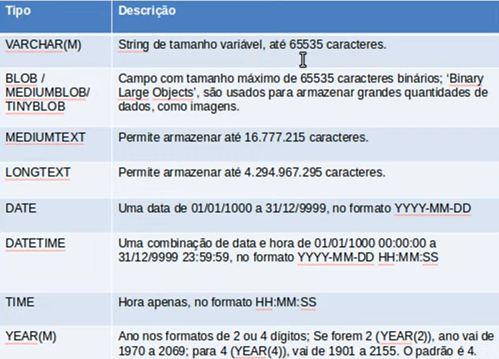
default character set utf8

default collate utf8\_general\_ci;

drop database cadastro;







create table pessoas(

id int not null auto\_increment,

nome varchar(30) not null,

nascimento date,

sexo enum('M','F'),

peso decimal(5,2),

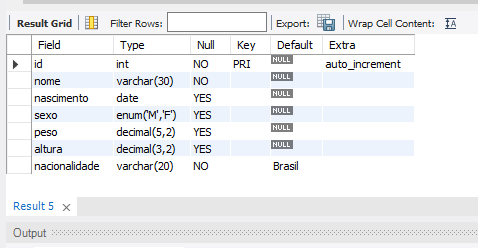
altura decimal(3,2),

nacionalidade varchar(20) default 'Brasil' not null,

primary key (id)

)default character set = utf8;

describe pessoas;



insert into pessoas

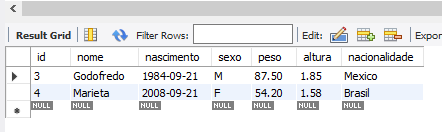
(id,nome,nascimento,sexo,peso,altura,nacionalidade)

values

(default,'Godofredo','1984/09/21','M','87.5','1.85','Mexico'),

(null,'Marieta','2008-09-21','F','54.2','1.58',default);

select \* from pessoas;



alter table pessoas

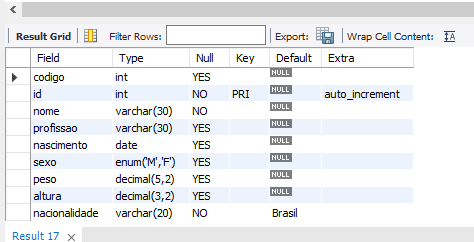
add column profissao varchar(10) after nome;

alter table pessoas

modify column profissao varchar(30) not null default ‘ ‘;

alter table pessoas

add column codigo int first;



alter table pessoas

change profissao prof varchar(20);

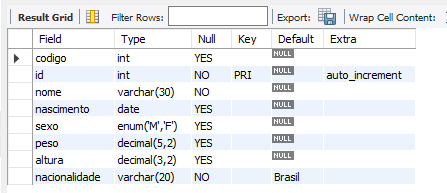
alter table pessoas

rename to gafanhotos;

alter table gafanhotos

drop column prof;

desc gafanhotos;



drop table gafanhotos;

create table if not exists cursos(

nome varchar(30) not null unique,

descricao text,

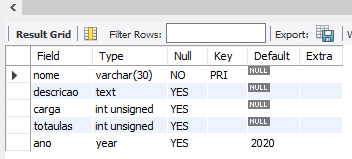
carga int unsigned,

totaulas int unsigned,

ano year default '2020'

)default charset utf8;

describe cursos;

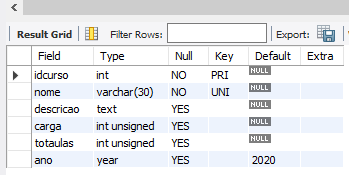


alter table cursos

add column idcurso int first;

alter table cursos

add primary key(idcurso);



insert into cursos values

('1','HTML4','Curso de HTML5','40','37','2014'),

('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'),

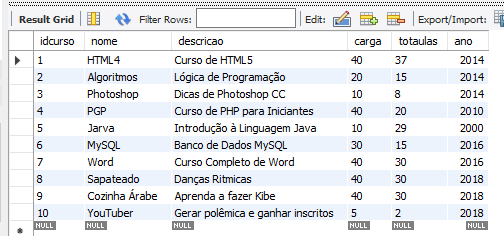
('7','Word','Curso Completo de Word','40','30','2016'),

('8','Sapateado','Danças Ritmicas','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;



update cursos

set nome = 'HTML5'

where idcurso = '1';

update cursos

set nome = 'PHP', ano ='2015'

where idcurso = '4';

UPDATE CURSOS

SET NOME='JAVA',CARGA='40',ANO='2015'

WHERE IDCURSO='5';

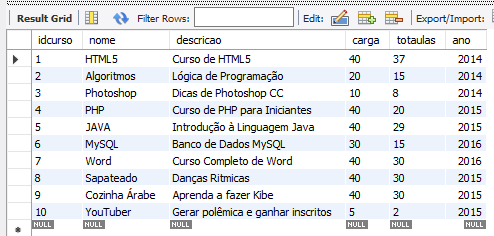
update cursos

set ano = '2015'

where ano = '2018'

limit 3;

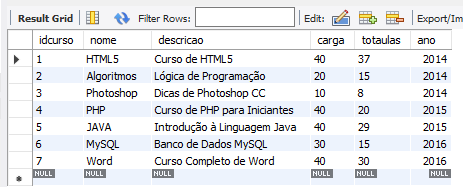
select \* from cursos;



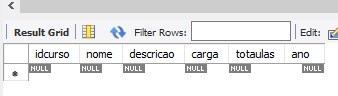
delete from cursos

where idcurso > '7'

limit 3;



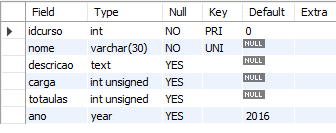
truncate cursos;





**Comando SELECT**

describe cursos;

****

Select \* from cursos;

select ano, nome, carga from cursos;

select ano, nome, carga from cursos

order by ano, nome;

select \* from cursos

where ano = '2016'

order by nome;

select nome, carga from cursos

where ano = '2016'

order by nome;

select nome, descricao from cursos

where ano <= 2015

order by nome;

select nome, descricao, ano from cursos

where ano != '2015'

order by nome;

select nome, descricao, ano from cursos

where ano <> '2015'

order by ano;

select nome, ano from cursos

where ano between 2015 and 2017

order by ano desc, nome;

select nome, ano from cursos

where ano in (2014,2016,2018)

order by ano desc, nome;

select \* from cursos

where carga > 35 and totaulas <30;

select \* from cursos

where carga > 35 or totaulas <30;

select \* from cursos

where nome like 'P%'; nome começando com P ou p

select \* from cursos

where nome like '%P'; nome terminando com P ou p

select \* from cursos

where nome like 'P%P'; nome começando e terminando com P ou p

select \* from cursos

where nome like '%PH%'; contenha PH ou ph no nome

select \* from cursos

where nome not like '%PH%'; não contenha PH ou ph no nome

select \* from cursos

where nome like 'P%P\_'; nome começando e terminando em P seguido de um caractere

select \* from cursos

where nome like 'P\_\_P%'; começa com P, dois caracteres qq, P e qualquer coisa depois

select \* from gafanhotos

where nome like '%silva%'; qualquer pessoa que tenha silva no nome

select distinct nacionalidade

from gafanhotos

order by nacionalidade; mostrar 1 de cada (não mostra repetidos)

select count(\*) from cursos; mostra quantas linhas tem

select count(\*) from cursos mostra quantas linhas tem com carga > 40

where carga > 40;

select max(carga)

from cursos; mostra maior valor de carga entre todas

select max(carga) from cursos

where ano ='2016'; mostra maior valor de carga no ano 2016

select nome, ano, min(carga) from cursos

where ano ='2016'; mostra menor valor de carga no ano 2016

select sum(totaulas)

from cursos soma a coluna total de aulas do ano 2016

where ano ='2016';

select avg(totaulas)

from cursos média da coluna total de aulas do ano 2016

where ano ='2016';

select totaulas, count(\*) from cursos

group by totaulas agrupando por total de aulas e contando quantas

order by totaulas; ocorrências tem em cada grupo

select ano, count(\*) from cursos

where totaulas > 30

group by ano

having ano > 2013

order by count(\*) desc;

select carga, count(\*) from cursos

where ano > 2015

group by carga

having carga > (select avg(carga) from cursos);

select nacionalidade, count(\*) from gafanhotos where nacionalidade != 'brasil'

group by nacionalidade

having count(nacionalidade) > 3

order by count(nacionalidade);

**Incluindo uma chave estrangeira na tabela gafanhotos:**

alter table gafanhotos

add column cursopreferido int;

alter table gafanhotos

add foreign key (cursopreferido)

references cursos(idcurso);

**Juntando dados de duas tabelas (inner join só aparecem linhas com relacionamento):**

select gafanhotos.nome, cursos.nome, cursos.ano

from gafanhotos inner join cursos

on gafanhotos.cursopreferido = cursos.idcurso

order by gafanhotos.nome;

**Para facilitar usa-se o as (A palavra inner é opcional)**

select g.nome, c.nome, c.ano

from gafanhotos as g inner join cursos as c

on g.cursopreferido = c.idcurso

order by g.nome;

**Juntando dados de duas tabelas (left outer join lista todas as linhas da tabela que está à esquerda do join, no caso gafanhotos, incluindo as que não possuem relacionamento), a palavra outer é opcional:**

select g.nome, c.nome, c.ano

from gafanhotos as g left outer join cursos as c

on g.cursopreferido = c.idcurso

order by g.nome;

**Juntando dados de duas tabelas (right outer join lista todas as linhas da tabela que está à direita do join, no caso cursos, incluindo as que não possuem relacionamento) ), a palavra outer é opcional:**

select g.nome, c.nome, c.ano

from gafanhotos as g right outer join cursos as c

on g.cursopreferido = c.idcurso

order by g.nome;

**Criando uma terceira tabela para resolver o N para N e fazendo um join das três tabelas:**

create table assiste (

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;

select g.nome,c.nome

from gafanhotos as g

join assiste as a

on g.id = a.idgafanhoto

join cursos c

on c.idcurso = a.idcurso

order by g.nome;