

RESUMO CURSO MYSQL - CURSO EM VÍDEO

DDL - Data Definition Language

- Create Database
- Create Table
- Alter Table
- Drop Table (apaga toda as estruturas)

DDL (Data Definition Language) (Suporte)

- CREATE : definição de um objeto (tabela, índice)
- ALTER : alteração de um objeto
- DROP : eliminação de um objeto

DML - Data Manipulation Language

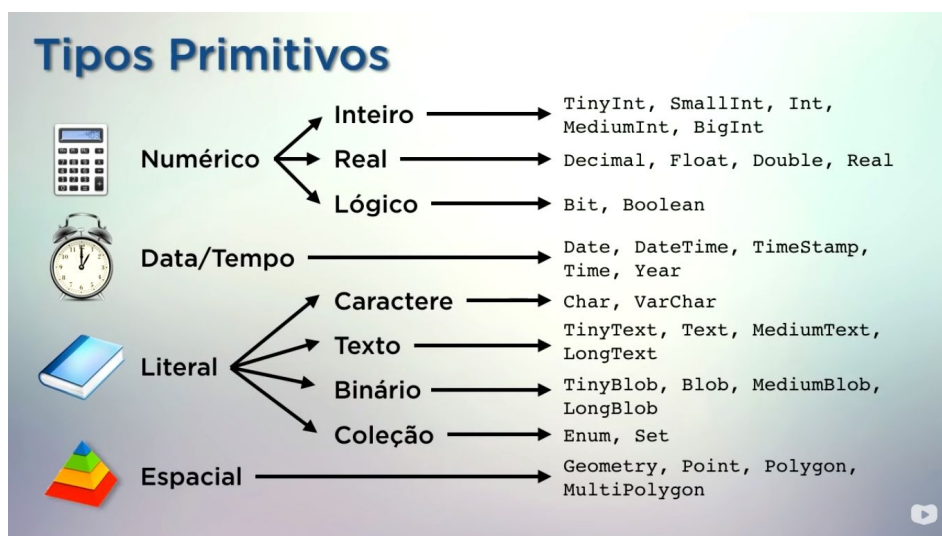
- Insert Into
- Update
- Delete
- Truncate (apaga os dados, mas mantém a estrutura)

DML (Data Manipulation Language) (Desenvolvimento)

- SELECT : pesquisa de dados
- UPDATE : atualização de dados
- DELETE : eliminação de dados
- INSERT : inserção de dados

DQL - Data Query Language

- Select



DCL (Data Control Language) (Suporte/Desenvolvimento)

- GRANT : fornecer um privilégio
- REVOKE : tirar um privilégio

```
create database cadastro
default character set utf8
default collate utf8_general_ci;
```

```
use cadastro; (Abrir banco de dados cadastro)
```

```
desc gafanhotos;
```

```
alter table pessoas
add column profissao varchar(10);
```

```
alter table pessoas
drop column profissao;
```

```
alter table pessoas
add column profissao varchar(10) after nome;
```

```
alter table pessoas
add column codigo int first;
```

```
alter table pessoas
modify column profissao varchar(20) not null default '';
```

```
alter table pessoas
change column profissao prof varchar(20);
```

```
alter table pessoas
rename to gafanhotos;
```

```
select * from gafanhotos;
```

```
create table if not exists cursos(
nome varchar(30) not null unique,
descricao text,
carga int unsigned,
totaulas int,
ano year default '2016'
)default charset = utf8;
```

```
alter table cursos
add primary key (idcursos); (definir chave primaria)
```

drop table if exists pessoas; (excluir tabelas inteira)

insert into pessoas value

('1', 'Pedro', '22'),

('2', 'Miguel', '34')

>>> Manipular linhas <<<

create table cursos(

idcursos int,

nome varchar (30) not null unique,

descricao text,

carga int unsigned,

totaulas int unsigned,

ano year default '2016'

)default charset=utf8;

insert into cursos values

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

('2', 'Algoritmos', 'Lgica 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', 'Bancos de Dados MySQL', '30', '15', '2016'),

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

('8', 'Sapateado', 'Danças Ritmicas', '40', '30', '2018'),

('9', 'Cozinha Arabe', 'Aprenda a fazer Kibe', '40', '30', '2018'),

('10', 'YouTuber', 'Gerar polêmica e ganhar inscritos', '5', '2', '2018');

update cursos (fazer modificações/atualização nas linhas)

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

where idcursos = '4';

update cursos

set nome = 'Java', carga = '40', ano = '2015'

where idcursos = '5'

limit 1;

update cursos

set carga = '0', ano = '2018'

where ano = '2050'

limit 1; (limita apenas a uma modificação, não realiza todas)

delete from cursos

```
where idcursos = '8';
```

```
delete from cursos  
where ano = '2050'  
limit 2;
```

```
truncate table cursos; (deleta todos os dados da tabela)
```

>>> Gerenciar cópias de segurança <<<

Server > Export date

Server > Import date

Aula 8

```
show tables; (mostrar tabelas)
```

Os dados estão armazenados em um servidor MySQL

Console é a tela preta, a forma de trabalhar por ele é o mesmo do workbench, pois os comandos usados são os mesmo.

PHPMYAdmin - é uma outra plataforma e que é mais fácil de utilizar e é uma ótima fonte de aprendizado, pois ao executar alguma ação ele te mostra os comandos que seria utilizado no terminal SQL.

No console:

>'show create table amigos' ou 'show create database amigos'; (ele mostra os comandos utilizados para criar a tabela, caso esta tabela tenha sido feita no PHPmysqldimin pro exemplo.

>>**Select:**

- select * from nome da tabela; (Seleciona todas as colunas da tabela)
- select * from cursos
order by nome; (ordena a tabela pela coluna nome)
- select * from cursos
order by nome desc; (ordena a tabela pela coluna nome em ordem decrescente - este desc não é de 'describe')
- select nome, carga, ano from cursos
order by nome; (mostra apenas as colunas específicas - especificadas no código)
- select ano, carga, nome from cursos
order by ano, nome; (mostra as colunas específicas e ordena a tabela na ordem selecionada)
- select * from cursos
where ano = '2016' (pode usar também sinais: >, <, <=, >=, !=, <>)
order by nome; (seleciona apenas as linhas com ano 2016)
- select nome, ano from cursos
where ano between 2014 and 2016
order by nome desc, ano asc;

- `select nome, descricao, ano from cursos`
`where ano in ('2015', '2016')`
`order by nome;`
- `select * from cursos`
`where carga > 35 and totaulas < 30;` (condições em duas colunas distintas)
- `select * from cursos`
`where carga > 35 or totaulas < 30;` (condições em duas colunas distintas)
- Link1: <http://wiki.icmc.usp.br/images/9/9b/SCC0241211DMLP02.pdf>
- Link2: <https://www.devmedia.com.br/guia/guia-completo-de-sql/38314>
- Link3: <http://www.benejsan.com.br/wp-content/uploads/2014/09/apostila1sql.pdf>
- `Select * from cursos`
`where nome like 'P%';` (Aparece todos os cursos que começam com a letra P);
- `Select * from cursos`
`where nome like '%A';` (Aparece todos os cursos que terminam com a letra A).
- `Select * from cursos`
`where nome like '%A%';` (Aparece todos os cursos que tem a letra A em qualquer posição da palavra);
- `Select * from cursos`
`where nome not like '%A%';` (Aparece apenas nomes onde não aparece a letra A);
- `Select * from cursos`
`where nome like 'ph%p';` (Busca os cursos que começam com ph e termina com p);
- `Select * from cursos`
`where nome like 'ph%p_';` (Busca os cursos que começam com ph e termina com p e tem algum caracter depois). Se colocar `__` (2) ele trará 2 coisas após;
- `Select distinct nacionalidade from gafanhotos;` (Traz os países sem repetição - países distintos);
- `Select count(*) from cursos;` (vai contar todos os cursos que tem cadastrado - numero de linhas totais);
- `Select count(*) from cursos where carga > 40;` (vai contar todos os cursos com carga horária maior que 40 horas);
- `Select max(carga) from cursos;` (vai trazer a maior carga horária de todos os cursos);
- `Select min(carga) from cursos;` (vai trazer a menor carga horária de todos os cursos);
- `Select sum(carga) from cursos;` (vai trazer a soma da carga horária de todos os cursos);
- `Select avg(carga) from cursos;` (vai trazer a média da carga horária de todos os cursos);
- `Select totaulas, count(*) from cursos`
`group by totaulas` (Conta o total de cursos que tenho dentro de cada total de aulas)
`order by totaulas;`
- `Select carga, count(nome) from cursos where totaulas = 30`
`group by carga;` (Conta o total de cursos que tem dentro de cada carga horária - considerando apenas cursos com total de 30 aulas);
- `Select ano, count(*) from cursos`
`group by ano` (Conta o total de cursos que teve em cada ano)
`having ano > 2016;` (Apenas cursos que ocorreram apos 2016- o having funciona apenas dentro do mesmo parâmetro que foi feito no group by);

- *Select carga, count(*) from cursos
where ano > 2015
group by carga* (vai contar os cursos agrupador pelas cargas acima de 2015)
having carga > (Select carga, count() from cursos);* (Vai trazer apenas as cargas que estiverem acima da média de todas as cargas);

Join/Primary Key/ Foreign Key/References

- Muitos p/ muitos:
 - Criando tabela:

```
create table gafanhotos_assiste_curso (
  id int not null auto_increment,
  data date,
  idgafanhotos int,
  idcurso int,
  primary key (id),
  foreign key (idgafanhotos) references gafanhotos(id),
  foreign key (idcurso) references cursos(idcursos)
) default charset = utf8;
```
 - Inserindo dados:

```
insert into gafanhotos_assiste_curso values
(default, '2014-03-01', '1','2');
```
 - Trazendo informações para a tabela central - junção:

```
select g.nome, c.nome from gafanhotos g
join gafanhoto_assiste_curso a
on g.id = a.idgafanhotos
join cursos c
on c.idcurso = a.idcurso;
```

