MY SQL - EXPLORANDO RELACIONAMENTOS COM WORKBENCH

PARTE 01 - INTRODUÇÃO AO COMMAND LINE:

**1.OPERAÇÕES BÁSICAS NO COMMAND LINE 1:**

\*command line: (operação do database via terminal, pois é como será feita a primeira conexão com o banco de dados)

-windows: no prompt de comando, mysql -u root

-para criar tabelas via terminal: CREATE TABLES cursos(id curso INT NOT NULL PRIMARY KEY AUTO\_INCREMENT, nome VARCHAR(10));

-para inserir dados na table via terminal:

INSERT INTO cursos (nome) VALUES (‘MySQL’);

INSERT INTO cursos (nome) VALUES (‘HTML’);

INSERT INTO cursos (nome) VALUES (‘CSS’);

(atalho de código, seta para cima)

-update html para html5:

UPDATE cursos SET nome=’HTML5’ WHERE id\_curso=2;

**2.OPERAÇÕES BÁSICAS NO COMMAND LINE 2:**

\*exemplo executando insert, select e delete.

\*exemplo usando alter table:

ALTER TABLE cursos ADD carga\_horaria INT(2)

\*usando update:

UPDATE cursos SET carga\_horaria=20;

(vai aplicar para todos pois não foi colocado WHERE).

\*usando drop table para deletar tabela

DROP TABLE usuarios;

\*usando drop database para deletar database

DROP DATABASE teste;

PARTE 02 - MODELO RELACIONAL:

**1.INTRODUÇÃO AO TEMA:**

-relacionar informações entre várias tabelas

-é possível pelo princípio dos dados atômicos, informações atomizadas.

-as chaves estrangeiras permitem o relacionamento das tabelas (foreign key).

\*inner join (é usado para ler a relação das tabelas).

SELECT \* FROM pessoas JOIN cursos ON pessoas.fk\_cursos = cursos.id\_cursos



**2.CRIANDO AS TABELAS VIDEOS E AUTHOR:**

-retirada do campo data.

-criação da tabela pelo php (não esquecer o primary e autoincrement)

-povoamento da tabela

INSERT INTO videos (author, titulo, likes, dislikes) VALUES (‘Maria’, ‘MySQL’, 10, 2);...

…

-criando tabela author pelo php.

-no sql, inserir informação por informação:

INSERT INTO author (name, born) VALUES (‘Maria’, ‘1992-04-19’);...

…

**3.PREPARANDO A TABELA VIDEOS PARA RECEBER DADOS DE OUTRA TABELA:**

\*criando o relacionamento na tabela video:

-é necessário mudar a estrutura da tabela.

-alterar o tipo da coluna author de text para inteiro

-deletar os nomes para ficarem em branco

UPDATE videos SET author=’’;

(como não foi colocado o WHERE todos serão alterados)

-preencher o author de uma tabela na outra pela referência ao id:

UPDATE videos SET author=1 WHERE id video=1;...

…

**4.REALIZANDO CONSULTA COM O JOIN 1:**

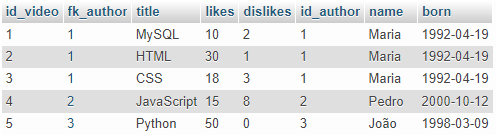
\*definição do local onde a foreign key vai entrar (fk\_).

-em ‘visão de relações’: digitar o nome da coluna que a relação está sendo trabalhada fk\_author, marcar cascade tanto em on delete quanto em on update e marcar os outros campos (coluna, banco de dados, tabela e coluna)(no php).

-para trazer o nome do autor:

SELECT \* FROM videos JOIN author ON videos.fk\_author = author.id\_author

-juntou as tabelas apenas onde a fk\_author é igual a id\_author



\*para mostrar apenas o título e o nome do autor:

SELECT videos.title, author.name FROM videos JOIN author ON videos.fk\_author=author.id\_author;

**5.CRIANDO A TABELA SEO E RELACIONANDO COM A TABELA VIDEOS:**

\*criando tabela SEO (php).

-povoando tabela de categorias:

INSERT INTO videos (fk\_author, title, likes, dislikes) VALUES (2, 'PHP', 20, 8);

INSERT INTO seo (category) VALUES (‘frontend’);

INSERT INTO seo (category) VALUES (‘backend’);

**6.REALIZANDO CONSULTA COM O JOIN 2:**

**7.CRIANDO A TABELA PLAYLIST:**

**8.CRIANDO A TABELA VIDEOS\_PLAYLIST RELACIONANDO-A COM OUTRAS TABELAS**