Banco de Dados II - MySQL

Atividade Avaliativa Individual do 2º Bimestre

Vitor Hugo Zauli Calcagnoto 2ºDS

# OBJETIVOS DE APRENDIZAGEM

Lembrar os comandos necessários para a criação de banco de dados e tabela;

Lembrar os comandos necessários para alterar a estrutura de uma tabela;

Lembrar os comandos necessários para a inserção ou atualização de dados em tabelas;

Aplicar os comandos aprendidos no bimestre para a realização de consultas simples e com mais de uma tabela, considerando as chaves (primárias e estrangeiras), bem como funções de agregação e cláusulas de ordenação e agrupamento.

# ATIVIDADE

Revisão

1. Escreva o comando SQL para **criar um banco** de dados chamado CineMarvelDC (0,5 ponto).

**R:** create database CineMarvelDC;

1. **Crie uma tabela** chamada vilao com os seguintes campos (0,5 ponto):

* idVilao (inteiro, chave primária)
* nome (texto até 100 caracteres, obrigatório)
* codinome (texto até 100 caracteres)
* origem (texto até 50 caracteres)

**R:** create table vilao (

idVilao int Not null,

nome varchar(100) not null,

codinome varchar(100),

origem varchar(50));

1. **Altere a tabela** vilao para adicionar o campo nivel\_periculosidade (inteiro) (1,0 ponto).

**R:** alter table vilao add (nivel\_periculosidade int);

1. Escreva um comando para **criar uma tabela** confronto que contenha (0,5 ponto):

* idConfronto (chave primária)
* idHeroi e idVilao (inteiros, chaves estrangeiras referenciando tabelas heroi e vilao)
* local (texto até 100 caracteres)
* resultado (vitória do herói ou vilão)

**R:** create table confronto (

idConfronto int not null primary key,

idHeroi int,

idVilao int,

local varchar(100),

resultado varchar(20),

FOREIGN KEY (idVilao) REFERENCES vilao(id),

FOREIGN KEY (idHeroi) REFERENCES heroi(id));

1. **Insira** um vilão chamado *Arthur Fleck* com codinome *Coringa*, origem *DC*, e nível de periculosidade 10 (0,5 ponto)*.*

**R:** insert into vilao ( nome, codinome, origem, nivel\_periculosidade)

values ( 'Arthur Fleck', 'coringa', 'DC', 10);

Consultas (1,0 ponto cada)

1. **Liste** todos os vilões com nível de periculosidade maior que 7.

**R:** select \* from vilao where nivel\_periculosidade > 7;

1. **Selecion**e os nomes dos heróis cujo codinome começa com a letra "S".

**R:** select \* from heroi where nome like 'S%';

1. **Liste** todos os confrontos ordenados por local em ordem alfabética decrescente.
2. **Mostre** o nome do herói, o nome do vilão e o resultado dos confrontos, unindo as tabelas heroi, vilao e confronto.
3. **Liste** os nomes dos vilões que já participaram de algum confronto com o herói chamado *Capitão América*.

**R:** SELECT vilao.nome

FROM confronto

JOIN heroi ON confronto.idHeroi = heroi.idHeroi

JOIN vilao ON confronto.idVilao = vilao.idVilao

WHERE heroi.codinome = 'Capitão America';

1. **Quantos** vilões existem no banco de dados?

**R:** select idVilao from vilao;

1. **Calcule a média** de periculosidade dos vilões agrupados por origem.

**R:** select origem, avg(nivel\_periculosidade) from vilao

group by vilao.origem;

1. Liste as origens que possuem mais de 2 vilões cadastrados.

R: SELECT origem, COUNT(\*) AS total\_vilao

FROM vilao

GROUP BY origem

HAVING COUNT(\*) > 2;

Lista de Comandos necessários

* CREATE DATABASE
* CREATE TABLE
* ALTER TABLE nome\_da\_tabela ADD …
* INSERT INTO
* UPDATE nome\_da\_tabela SET
* SELECT
  + FROM
  + JOIN
  + WHERE
  + ORDER BY
  + ON
  + GROUP BY
  + HAVING (deve ser utilizada após GROUP BY como filtro)
  + COUNT() - contagem
  + AVG() - média
  + SUM() - soma
  + LIKE - para comparações não exatas (aproximadas)