

O MySQL

O MySQL é um sistema de gerenciamento de banco de dados (SGBD) relacional de código aberto utilizado na maioria das aplicações para gerir suas bases de dados, que utiliza a linguagem SQL (Structured Query Language ou Linguagem de Consulta Estruturada) como interface. É linguagem mais popular para inserir, acessar e gerenciar o conteúdo armazenado num banco de dados.

Para utilizar o MySQL, é necessário instalar um servidor e uma aplicação cliente. O servidor é o responsável por armazenar os dados, responder às requisições, controlar a consistência dos dados, bem como a execução de transações concomitantes entre outras. O cliente se comunica com o servidor através da SQL.

Este SGBD inicialmente foi desenvolvido para trabalhar com projetos de pequeno e médio porte, com a capacidade de suportar por volta de cem milhões de registros em cada tabela, podendo chegar ao tamanho médio de aproximadamente cem megabytes por tabela, entretanto, esses eram os tamanhos recomendados nas primeiras versões. Porém, hoje em dia o MySQL ultrapassa extraordinariamente esses limites e capacidades das versões anteriores.

MySQL é conhecido por ser de fácil utilização, e usado por empresas que trabalham com grandes volumes de dados, tais como, NASA, Bradesco, HP, Sony entre outras grandes empresas de renome. Possui uma interface extremamente simples e é compatível com grande parte dos sistemas operacionais. Podemos dizer que essas são duas das grandes características que fazem o MySQL ser tão utilizado atualmente e estar em constante crescimento.

1. Trabalhando com o MySQL

Aqui demonstraremos queries básicas para se trabalhar com o MySQL.

1.1 Criar uma tabela

Para se criar um banco de dados é bastante simples:

```
CREATE DATABASE escola;
```

1.2 Fazer uso do Banco de dados criado

Depois criar o bando de dados, é necessário avisar ao mysql que vamos utilizá-lo:

```
USE escola;
```

1.3 Criar a tabela que será utilizada:

```
CREATE TABLE alunos(  
    codigo int AUTO_INCREMENT PRIMARY KEY,  
    nome varchar(30) NOT NULL,  
    responsavel varchar(30),  
    cidade varchar(20) NOT NULL,  
    turma varchar(10) NOT NULL  
);
```

- AUTO_INCREMENT pode ser utilizado para automatizar um código que sirva de chave primária de uma tabela.
- PRIMARY KEY define a chave primária da tabela, isto é, o campo que serve como chave da tabela e que não pode ser repetido.
- NOT NULL define que um determinado campo seja de preenchimento obrigatório.

1.4 Inserir dados na tabela:

```
insert into alunos (nome, responsavel, cidade, turma) values ("João", "João Pai", "São Paulo", "Robótica");
```

1.5 Listar os dados inseridos na tabela

```
SELECT * FROM alunos;
```

O comando SELECT permite diversas variações e combinações nele, podemos buscar exatamente o que queremos e do jeito que queremos, por exemplo.

```
SELECT nome FROM alunos;
```

Aqui o select traz só o nome dos alunos.

1.6 Ordenação

Podemos ordenar a listagem dos dados:

```
SELECT * FROM alunos ORDER BY nome asc;
```

Aqui ordenamos a listagem pelo nome em ordem alfabética

1.7 Alteração dos dados

Para alterar um registro, usamos o comando UPDATE, com ele é possível editar os campos de sua tabela e colocar outro valor neles.

```
UPDATE alunos SET nome="Ricardo" WHERE codigo=2;
```

1.8 Deleção

```
DELETE FROM alunos WHERE codigo=3;
```

Uma vez executado, esse comando não é reversível, portanto tome bastante cuidado ao deletar algum registro de seu banco de dados.

Existem inúmeras sintaxes de comandos SQL que podem ser criadas para os mais diversos fins. Durante o decorrer do curso, construiremos outras mais queries.