

Introdução a Banco de Dados – Modelo Físico

Bruno Costa

Sumário

- Projeto Físico (3ª Fase)
- MySQL e SQL
- DDL e DML
- Tipos de Dados
- Comandos DDL (Create, Drop e Alter)
- Comando DML (Select, Insert, Update e Delete)

SQL

- Hoje é expandido para *Structured Query Language* ou seja Linguagem de Consulta Estruturada.
- Foi criada e implementada pela IBM
- É a linguagem padrão para SGBDs relacionais

SQL

- É uma linguagem de banco de dados abrangente tem instruções para definição de dados, consultas e atualizações.
- É dividida em: DDL E DML.
- **DDL – Linguagem de Definição de Dados:** fornece comandos para definições de esquemas de relação, criação/remoção de tabelas, criação de índices e modificação de esquemas
- **DML – Linguagem de Manipulação de Dados:** inclui uma linguagem de consulta baseada na álgebra relacional e cálculo relacional de tupla. Compreende comandos para inserir, consultar, remover e modificar tuplas num Banco de Dados.

SQL

- A SQL usa os termos **tabela**, **linha** e **coluna** para os termos do modelo relacional formal *relação*, *tupla* e *atributo*, respectivamente.
- Obs.: A linguagem SQL não é case sensitive ou seja não diferencia letras maiúsculas de minúsculas.

Tipos de Dados

- Alguns tipos de Dados mais comuns são:
 - CHAR,
 - VARCHAR,
 - TEXT
 - INT,
 - BIGINT
 - DECIMAL,
 - DATE,
 - DATETIME,
 - TIME.

Tipos de Dados

- Caracteres:
- **CHARACTER(X)** ou **CHAR** – Representa um String de tamanho x.
- **CHARACTER VARYING(X)** ou **VARCHAR** – representa um String de tamanho x. Pode conter dados de textos de até 255 caracteres de tamanho.
- **TEXT** – é utilizado quando se deseja armazenar uma quantidade grande de texto, pois não tem um tamanho definido.

Tipos de Dados

- Números Exatos:
- **INTEGER** ou **INT** – utilizado para armazenar números inteiros.
- **BIGINT** – é utilizado quando se deseja armazenar valor 'numéricos extensos.
- **DECIMAL**– utilizado para armazenar números decimais, tem uma precisão e uma escala (números de dígitos na parte fracionária). Muito usado para representar dinheiro.
- Sintaxe: DECIMAL(5,2).
- *os números mostram quantos dígitos o banco de dados deve esperar em frente dos decimais, e quantos depois.*
- *Exe.: 19892.23 – cinco antes e 2 depois.*

Tipos de Dados

- Datetimes:
- **DATE**: armazena ano(4 dígitos), mês (2dígitos) e dia (2 dígitos). O Tipo de dado dever ser informados no padrão internacional.
- *Exemplo: 1989-06-01*
- **TIME**: armazena hora (2 dígitos), minuto (2 dígitos) e segundo (2 dígitos);
- **DATETIME** – armazena data e hora.

Comandos DDL

SQL – COMANDOS BÁSICOS

- **Comando da Linguagem de definição de dados (DDL)**
- CREATE – criar banco de dados ou tabela
- DROP – apagar banco de dados ou tabela
- ALTER – alterar tabelas(adicionar colunas, renomear colunas ou tabelas, alterar tipos de dados de colunas)

COMANDO CREATE

- É utilizado para basicamente criar banco de dados como as tabelas que farão parte dele.
- Sintaxe para criação do banco

```
mysql> CREATE DATABASE nome_do_banco;
```

É o comando

É o nome do banco

A instrução é encerrada com um ponto de vírgula (;)

```
mysql> CREATE DATABASE nome_do_banco;  
Query OK, 1 row affected (0.00 sec)
```

Esta ultima linha será o resultado do SGBD informando que uma *linha foi afetada ou modificada, construída etc.*

CREATE DATABASE

- Depois que o banco de dados é criado é preciso informar ao SGBD que você irá utilizá-lo.
- Através do comando **USE**.

```
mysql> use nome_do_banco;  
Database changed  
mysql>
```

- O SGBD sempre dará uma resposta caso esteja correta ou não!
- Neste ele informou que o “banco de dados mudou/foi alterado” agora podemos trabalhar em cima do banco.

Comando Create Table

- CREATE TABLE nome_da_tabela
- (- nome_da_coluna1 tipo_de_dado,
- Nome_da_coluna2 tipo_de_dado
-);

Comando Create Table

- Exemplo:
- CREATE TABLE Autor
(
 cod_autor INT(12) PRIMARY KEY,
 nome varchar(30) NOT NULL,
 nascimento date NOT NULL,
);

CREATE TABLE

- Exemplo:
- CREATE TABLE LIVRO (
 cod_livro INT(12) PRIMARY KEY,
 nome VARCHAR(30) NOT NULL,
 fk_cod_autor INT(12) NOT NULL,
 FOREIGN KEY (cod_autor) REFERENCES Autor(cod_autor)
);

COMANDO ALTER

- As definições de uma tabela básica ou de outros elementos do esquema poderão ser alterados pelo comando **ALTER**.
- **Alter Table**
 - Permite alterar ou adicionar atributos de uma determinada tabela.
 - Utilizado na alteração ou exclusão das restrições da tabela.

Alter Table

- **Sintaxe para adicionar uma nova coluna**

- ALTER TABLE nome_tabela ADD nome_coluna *tipo de dado*;

- **Sintaxe para modificar uma coluna de uma tabela**

- ALTER TABLE nome_tabela MODIFY nome_coluna *tipo de dado*;

- **Sintaxe para remover uma coluna de uma tabela**

- ALTER TABLE nome_tabela DROP nome_coluna;

- **Sintaxe para alterar atributo(coluna) de uma tabela**

- ALTER TABLE nome_tabela CHANGE nome_coluna nome_coluna_nova [tipo_de_dado];

Alter Table

- **Sintaxe para adicionar uma restrição a uma tabela**
 - ALTER TABLE nome_tabela ADD CONSTRAINT nome_restrição *nome_coluna*;
- **Sintaxe para remover restrição**
 - ALTER TABLE nome_tabela DROP nome_restrição;
- **Sintaxe para remover vínculo de chave estrangeira**
 - ALTER TABLE nome_tabela DROP FOREIGN KEY <nome_coluna_fk>;

Alter Table

- ALTER TABLE livro ADD idioma varchar(15)
 - Adiciona a coluna idioma de tamanho 15 na tabela livro
- ALTER TABLE livro MODIFY idioma varchar(30)
 - Altera a coluna idioma para o tamanho 30
- ALTER TABLE livro DROP idioma
 - Exclui a coluna idioma
- ALTER TABLE livro DROP PRIMARY KEY (coluna)
 - Exclui a chave primária da tabela Livro
- ALTER TABLE livro ADD CONSTRAINT PRIMARY KEY (coluna)
 - Adiciona chave primária
- ALTER TABLE livro ADD CONSTRAINT FOREIGN KEY (coluna) REFERENCES nome_da_tabela (coluna)
 - Adiciona chave estrangeira