



Database Schemas e Tables

Msc Denival Araújo dos Santos

DDL – Linguagem de definição de dados

Bases de
Dados
(Databases)

- **Databases** é o conjunto de dados armazenados em um sistema de gerenciamento de banco de dados.

Criar nova database

```
create database nome_database;
```

Remover database

```
drop database nome_database
```

DDL – Linguagem de definição de dados

Esquemas Schemas

- Na prática **SCHEMA** e **DATABASE** são praticamente a mesma coisa.
- Quando utilizamos o comando **Create Database**, cria-se um banco de dados propriamente dito, quando no Schema você está criando uma subdivisão dentro de um Banco de Dados.
- Alguns SGBD possuem apenas esquemas como separadores de tabelas como o Oracle e o MySQL.
- Já outros como o PostgreSQL e o Microsoft SQL Server, possuem dois níveis de separação, bancos de dados e esquemas que por sua vez ficam dentro dos bancos de dados.
- **Sintaxe: CREATE SCHEMA nome_esquema;**

DDL – Linguagem de definição de dados

Esquemas Schemas

```
-- Cria a base de dados
create database ifpi;

-- Cria o esquema empresa
create schema cliente;
create table cliente.clientes(
    id int not null,
    nome varchar(35)
);
-- Cria o esquema empresa
create schema empresa;
create table empresa.departamento(
    id int not null primary key,
    nome varchar(35)
);
create table empresa.funcionario(
    matricula int not null primary key,
    nome varchar(100) not null,
    departamento_id int not null,
    foreign key(departamento_id) references empresa.departamento(id)
);
```

DDL – Linguagem de definição de dados

Tipos de dados

- **Campos de texto**
 - **Char(n)** – é uma cadeia de caracteres de tamanho fixo, com tamanho n definido pelo usuário.
 - **Varchar(n)** – é uma cadeia de caracteres de tamanho variável, com o tamanho n máximo definido pelo usuário.

DDL – Linguagem de definição de dados

Tipos de dados

- **Campos numéricos**
 - **Smallint** – é um número inteiro pequeno.
 - **Int** – é um número inteiro.
 - **Bigint** – é um número inteiro grande.
 - **Numeric(p,d)** – é um número de ponto fixo cuja precisão é definida pelo usuário.
- **Data e Hora**
 - **Date** – é um calendário contendo um ano (dia, mês e ano);
 - **Time** – representa horário (hora, minuto e segundo).

DDL – Linguagem de definição de dados

Tabelas (Tables)

- Coleção de dados, organizada em linhas e colunas/atributos.
- No mundo RDBMS, tabela é conhecida como relação e linha como tupla.

Sintaxe

```
create table nome_da_tabela(  
    atributo1 tipo_de_dado [restrições],  
    atributo2 tipo_de_dado [restrições],  
    ...  
    atributoN tipo_de_dado [restrições],  
    primary key(atributo_PK),  
    constraint nome_constraint foreign key(atributo_FK)  
        references tabela_referenciada(atributo_PK)  
);|
```

DDL – Linguagem de definição de dados

Tabelas
(Tables)

Exemplo

```
create table estado(  
    uf varchar(2) not null,  
    nome varchar(25),  
    primary key(uf)  
);  
  
create table cidade(  
    idcidade int not null,  
    nome varchar(35) not null,  
    estado_uf varchar(2) not null,  
    primary key(idcidade),  
    constraint fk_cidade_estado foreign key(estado_uf) references estado(uf)  
);
```


DDL – Linguagem de definição de dados

Cláusulas

- Na criação de tabelas no MySQL, temos várias opções e comandos disponíveis para definir a estrutura da tabela, suas colunas e as restrições. Um comando bem comum que podem ser usados na criação de tabelas é :
- **Sintaxe: CREATE TABLE [IF NOT EXISTS] nome_da_tabela (...)**

Exemplo

```
create table if not exists cor(  
  id int not null,  
  nome varchar(45),  
  primary key(id)|  
);
```

DDL – Linguagem de definição de dados

ON DELETE

e

ON UPDATE

- As cláusula **ON DELETE/ON UPDATE** são usadas em conjunto com a criação de uma chave estrangeira (FOREIGN KEY) para especificar o que acontece com os registros na tabela filho quando um registro correspondente na tabela pai é excluído/atualizado.
- Oferece opções para controlar o comportamento de exclusão/atualizações em cascata, nulo ou restrito.

DDL – Linguagem de definição de dados

ON DELETE

e

ON UPDATE

- **CASCADE**: Quando um registro na tabela pai é substituído ou excluído, os registros correspondentes na tabela filho também são automaticamente substituídos ou excluídos.
- **SET NULL**: Quando um registro na tabela pai é substituído ou excluído, os Valores das colunas correspondentes na tabela filho são definidos como NULL.
- **SET DEFAULT**: Similar ao SET NULL, mas os valores das colunas correspondentes na tabela filho são definidos como o valor padrão especificado.
- **RESTRICT**: Impede a exclusão ou atualização na tabela pai se houver registros correspondentes na tabela filho (**método padrão**).

DDL – Linguagem de definição de dados

ON DELETE

e

ON UPDATE

Exemplo

```
create table if not exists cor(  
    id int not null,  
    nome varchar(45),  
    primary key(id)  
);  
  
create table if not exists casa(  
    id int not null,  
    descricao varchar(35) not null,  
    id_cor int not null,  
    primary key(id),  
    foreign key(id_cor) references cor(id) ON DELETE CASCADE  
);
```

DDL – Linguagem de definição de dados

Campo
Auto
incremento

- O auto incremento permite que um número único seja gerado quando um novo registro é inserido em uma tabela. Em MYSQL trata-se da palavra chave **AUTO_INCREMENT**, cujo valor inicial padrão é 1, e se incrementa de 1 em 1.

Sintaxe

```
create table cor(  
  id int not null auto_increment,  
  nome varchar(35) not null,  
  primary key(id)  
);
```

Referências bibliográficas

Básica

- SILBERSCHATZ, Abraham; KORTH, Henry F; SUDARSHAN, S. **Sistema de banco de dados**. Rio de Janeiro: Elsevier, 2012. 861 p. ISBN 978-85-352-4535-6.
- MACHADO, Felipe Nery Rodrigues. **Banco de dados: projeto e implementação**. 3. ed. São Paulo: Érica, 2014. 396 p. ISBN 978-85-365-0019-5.
- ELMASRI, Ramez; NAVATHE, Shamkant. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson, 2011. xviii, 788 p. ISBN 978-85-7936-085-5 (broch.).

Referências bibliográficas

Complementar

- BEIGHLEY, Lyn. **Use a cabeça SQL**. Rio de Janeiro: Alta Books, 2008. 454 p. ISBN 978-85-7608-210-1.
- DATE, C. J. **Introdução a sistemas de bancos de dados**. Rio de Janeiro: Campus, 2000. xxiii, 803 p. ISBN 85-352-0560-8
- HEUSER, Carlos Alberto. **Projeto de banco de dados**. 6. ed. Porto Alegre (RS): Bookman, 2009. TEOREY, Toby J. et al. **Projeto e modelagem de banco de dados**. 2. ed. Rio de Janeiro: Elsevier, 2014.