

## BANCO DE DADOS

### Linguagem de Definição de Dados (DDL)

A manutenção do banco de dados é feita em dois grupos claros: Objetos do BD e Tuplas das Entidades. Usaremos como exemplo um clube de tratamento da forma física, selecionado, conceituado e imaginário chamado Visual Spa.

A tabela HOSPEDE contém as informações sobre os sócios (tabela 1):

NOME	GENERO	BIOTIPO	ALTURA
MIGUEL	M	M	1,67
JOSIEL	M	M	1,72
RAQUEL	F	G	1,65
LUCIANA	F	G	1,80
JOANA	F	M	1,65
EMANUEL	M	M	1,78

*Tabela 1*

E a tabela QUARTO a seguir é um exemplo da lista de todos os hóspedes que chegaram ou saíram no período entre 01 e 15/01/2010.

NOME	QUARTO	CHEGADA	SAIDA	DESCONTO
MIGUEL	4	01-01-2010	08-01-2010	0,20
JOSIEL	2	01-01-2010	15-01-2010	0,10
RAQUEL	1	01-01-2010	15-01-2010	
LUCIANA	3	01-01-2010	08-01-2010	0,10
JOANA	5	01-01-2010	15-01-2010	
EMANUEL	6	01-01-2010	15-01-2010	0,12
MIGUEL	3	09-01-2010	15-01-2010	
LUCIANA	4	09-01-2010	15-01-2010	

*Tabela 2*

### Base de Dados: CREATE, ALTER, DROP, TRUNCATE, COMMENT E RENAME

Use a instrução **CREATE** para definir um novo objeto (tabela, índice, Banco de Dados ou outros), seus campos e restrições de seu campo.

**Sintaxe:** CREATE [tipo] <nome>

A seguir o código para a criação do Banco de Dados:

```
CREATE DATABASE visualspa
```

A seguir o código necessário a criação da tabela HOSPEDE e QUARTO:

```
CREATE TABLE hospede (  
  nome VARCHAR(25) NOT NULL,  
  genero VARCHAR(1),  
  biotipo VARCHAR(1),  
  altura NUMERIC(5,2),  
  PRIMARY KEY (nome)  
)  
CREATE TABLE quarto (  
  nome VARCHAR(25) NOT NULL,  
  quarto INT NOT NULL,  
  chegada DATE NOT NULL,  
  saida DATE,  
  desconto NUMERIC(5,2),  
  PRIMARY KEY (nome, quarto)  
)
```

Utilizamos a instrução **ALTER** para modificar a estrutura de um objeto depois de ter sido criado, ou seja, este comando permite modificar objetos no banco de dados.

**Sintaxe:** ALTER <tipo> <nome> (ADD/DROP nome\_atributo1 <tipo> [NOT NULL], nome\_atributoN <tipo> [NOT NULL])

A seguir o código para a modificação da tabela QUARTO, adicionando a chave de ligação entre esta tabela e HOSPEDE:

```
ALTER TABLE quarto ADD CONSTRAINT fkQuarto FOREIGN KEY (nome) REFERENCES hospede  
(nome) ON UPDATE CASCADE ON DELETE CASCADE
```

Use a instrução **DROP** para remover um objeto depois de criado, ou seja, este comando permite a eliminação de objetos no banco de dados.

**Sintaxe:** DROP <tipo> <nome>

A seguir o código para a eliminação da tabela QUARTO (*não o execute pois necessitaremos dessas entidades para os exercícios posteriores*):

```
DROP TABLE quarto
```

O comando **TRUNCATE TABLE** permite remover todas as linhas de uma tabela em uma única operação, sem registrar as exclusões de linhas individuais.

O **TRUNCATE TABLE** é como executar a instrução **DELETE**, porém, sem usar a cláusula **WHERE**. Portanto, é usada para apagar completamente o conteúdo de uma tabela no MySQL.

Entretanto, a cláusula **TRUNCATE TABLE** é mais rápida e utiliza menos recursos de sistema e log de transações durante sua execução.

Exemplo: Vamos excluir todos os registros presentes na tabela hospede: (Não executar o comando, pois utilizaremos a frente na apostila).

```
TRUNCATE TABLE hospede
```

Outra dica para comentar as colunas é usar "**COMMENT**"

```
CREATE TABLE quarto (  
  nome VARCHAR(25) NOT NULL COMMENT,  
  quarto INT NOT NULL COMMENT,  
  PRIMARY KEY (nome, quarto)  
)
```

Outra dica que podemos renomear uma usando o comando **RENAME TABLE**.

```
RENAME TABLE hospede TO hospedes
```