

# Programação e Desenvolvimento de Banco de Dados

## Manipulação de dados e estruturas

Prof. Dr. Gilberto Fernandes Jr.

1

- Unidade de Ensino: 2
- Competência da Unidade: Conhecer e compreender a criação e manipulação de tabelas.
- Resumo: Saber elaborar script SQL inserção e manutenção de dados, alteração e exclusão de tabelas e constraints.
- Palavras-chave: SQL, manipulação de dados, constraints
- Título da Teleaula: Manipulação de dados e estruturas
- Teleaula nº: 2

2

## Contextualização

- Comandos utilizados na manipulação de bancos de dados
- Alteração de tabelas e *constraints*
- Exclusão de tabelas em banco de dados

3

## Comandos para manipulação de bancos de dados: SELECT, INSERT, UPDATE e DELETE

4

## Introdução

- Linguagem de manipulação de dados (DML)
  - inserir, atualizar, excluir ou modificar dados
  - Principais instruções
  - Cláusulas condicionais

5

## A cláusula SELECT

- Consulta SQL e o resultado é uma tabela
- Cláusula **WHERE**
  - AND, OR, NOT, <, <=, >, >=, = e <>
- Cláusula **FROM**
  - Especifica uma ou mais tabelas

```
SELECT * FROM convidado;
```

```
SELECT * FROM convidado WHERE nome LIKE 'A%';
```

6

### Instrução de Inserção (INSERT)

- Permite adicionar novas linhas ou registro numa tabela existente.

```
INSERT INTO nome_tabela (col1, col2) VALUES(15, 2);
```

```
INSERT INTO nome_tabela (col1, col2) VALUES(15, 2),(1,3);
```

- Inserir múltiplas linhas

```
INSERT INTO nome_tabela (a,b,c)
VALUES(1,2,3),(4,5,6),(7,8,9);
```

7

### Instrução de Inserção (INSERT)

Exemplo: considere a tabela a seguir

```
CREATE TABLE IF NOT EXISTS convidado (
  id INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
  nome VARCHAR(50) NOT NULL DEFAULT "",
  nascimento DATE,
  estudante ENUM('Não', 'Sim') NOT NULL DEFAULT 'Não'
);
```

8

```
INSERT INTO CONVIDADO (nome, nascimento, estudante)
VALUES ('Dani Moura', '1979-03-28', 'Sim');
```

Resultado:

id	nome	nascimento	estudante
1	Dani Moura	1979-03-28	Sim

Fonte: livro texto

```
INSERT INTO CONVIDADO (nome, nascimento, estudante)
VALUES ('Rui Albuquerque', null, 'Sim');
```

Resultado:

id	nome	nascimento	estudante
1	Dani Moura	1979-03-28	Sim
2	Rui Albuquerque		Sim

Fonte: livro texto

9

### Instrução de Atualização (UPDATE)

- UPDATE:** atualiza colunas de linhas existentes na tabela nomeada com novos valores.
- SET:** indica quais colunas modificar e os valores que devem ser fornecidos.

```
UPDATE convidado
SET estudante = 'Sim'
WHERE nome = 'Lebrencio Grulher'
AND nascimento = '08-Jul-1990';
```

10

### Instrução de Atualização (UPDATE)

- WHERE:** condições que identificam quais linhas devem ser atualizadas

```
UPDATE convidado SET estudante = 'Não';
```

- ORDER BY e LIMIT:**

```
UPDATE convidado
SET estudante = 'Sim'
WHERE nascimento < '08-Jul-1990'
LIMIT 10
ORDER BY nome;
```

11

### Instrução de Atualização (UPDATE)

- UPDATE pode ser usado para várias tabelas ao mesmo tempo
  - Não pode utilizar ORDER BY e LIMIT, neste caso

```
UPDATE lista, produto SET lista.preco = produto.preco
WHERE lista.id = produto.id;
```

12

### Instrução de Exclusão (DELETE)

- Instrução DML que exclui linhas de uma tabela.
- Uso semelhante à UPDATE.
- Sua sintaxe geral é dada por:

```
DELETE FROM nome_tabela
[WHERE condição]
[ORDER BY ...]
[LIMIT quantidade_linhas]
```

13

### Instrução de Exclusão (DELETE)

- Exemplos:

```
DELETE FROM convidados
WHERE estudante = 'Sim'
ORDER BY nome
LIMIT 10;
```

```
DELETE FROM log_usuario
WHERE usuario = 'rm'
ORDER BY datahora_acao LIMIT 1;
```

14

## Alteração de tabelas

15

### Alteração de tabelas (ALTER TABLE)

- Comando que altera a estrutura de uma tabela
  - Adicionar ou excluir colunas
  - Criar ou destruir índices
  - Renomear colunas, ou a própria tabela
  - Alterar o mecanismo de armazenamento

```
ALTER TABLE nome_tabela
[especificação_alteração
[,especificação_alteração]
...]
```

16

### Alteração de tabelas (ALTER TABLE)

- A sintaxe para as alterações é semelhante às cláusulas da instrução CREATE TABLE.
- Alguns comandos de **especificação\_alteração**:

```
ADD [COLUMN] (nome_coluna definição_coluna,...)
CHANGE [COLUMN] nome_antigo nome_novo definição_coluna
DROP [COLUMN] nome_coluna
RENAME COLUMN antigo_nome_coluna TO novo_nome_coluna
```

17

### Alteração de tabelas (ALTER TABLE)

- Múltiplas cláusulas ADD, ALTER, DROP e CHANGE são permitidas em uma única instrução ALTER TABLE, separadas por vírgulas

```
ALTER TABLE cliente DROP COLUMN parentesco,
DROP COLUMN telefones;
```

- Outros exemplos:

```
ALTER TABLE cliente AUTO_INCREMENT = 13;
```

```
ALTER TABLE pessoas CHARACTER SET = latin1;
```

18

### Adicionando e excluindo colunas (ADD e DROP)

- **ADD**: adicionar novas colunas a uma tabela
- **DROP**: remover colunas existentes
- **FIRST** ou **AFTER** nome\_coluna: adicionar uma coluna a uma posição específica dentro de uma linha da tabela. O padrão é na última linha.

19

### Renomeando, redefinindo e reordenando colunas

- As cláusulas **CHANGE**, **MODIFY**, **RENAME COLUMN** e **ALTER** permitem que os nomes e definições de colunas existentes sejam alterados. Exemplos:

```
ALTER TABLE pessoas CHANGE antigo novo BIGINT NOT NULL;
```

```
ALTER TABLE pessoas MODIFY nome INT NOT NULL;
```

```
ALTER TABLE pessoas RENAME COLUMN novo TO antigo;
```

20

### Usando restrições (*constraints*)

```
CREATE TABLE pessoa (
  id int NOT NULL PRIMARY KEY,
  nome varchar(255) NOT NULL,
  sobrenome varchar(255),
  idade int );
```

```
ALTER TABLE pessoa
DROP PRIMARY KEY;
```

```
ALTER TABLE pessoa
ADD CONSTRAINT PK_pessoa PRIMARY KEY (id, sobrenome);
```

21

### Usando restrições (*constraints*)

- Sintaxe para definição de restrição de chave estrangeira com **CREATE TABLE** ou **ALTER TABLE**

```
ALTER TABLE nome_tabela
ADD [CONSTRAINT [símbolo]] FOREIGN KEY
[index_nome] (index_col_nome, ...)
REFERENCES nome_tabela (index_col_nome,...)
[ON DELETE referências]
[ON UPDATE referências]
```

referências:  
RESTRICT | CASCADE | SET NULL | NO ACTION

22

### Usando restrições (*constraints*)

- Regras para nomear um índice de chave estrangeira:
  - Se definido, o valor do símbolo **CONSTRAINT** é usado. Caso contrário, o valor do **index\_nome FOREIGN KEY** é usado.
  - Se nenhum símbolo **CONSTRAINT** ou **FOREIGN KEY** **index\_nome** estiverem definidos, o nome do índice de chave estrangeira será gerado usando o nome da coluna de chave estrangeira de referência.

23

### Usando restrições (*constraints*)

- Integridade referencial:

```
CREATE TABLE pai (
  id INT NOT NULL,
  nome VARCHAR(50),
  PRIMARY KEY (id)
);
```

```
CREATE TABLE filha (
  id INT PRIMARY KEY,
  parente_id INT,
  nome VARCHAR(50)
);
```

```
ALTER TABLE filha
ADD CONSTRAINT FK_parente
FOREIGN KEY (parente_id) REFERENCES pai(id);
```

24

### Qualificadores

- São utilizados com as cláusulas ON UPDATE e ON DELETE
  - CASCADE**: qualificador que exclui ou atualiza a linha da tabela pai e exclui ou atualiza automaticamente as linhas correspondentes na tabela filha.
  - SET NULL**: exclui ou atualiza a linha da tabela pai e define como NULL a coluna ou colunas de chave estrangeira na tabela filha.
  - RESTRICT**: rejeita a operação de exclusão ou atualização da tabela pai.

25

### Qualificadores

- Para excluir uma restrição, podemos executar a instrução abaixo, em que a chave criada é mantida após esta execução, assim somente a restrição será excluída.

```
ALTER TABLE filha DROP FOREIGN KEY FK_parente;
```

26

## Testando a estrutura do banco de dados Guia Turístico

27

### Resolução de Situação-Problema

- Você está trabalhando em uma multinacional, no desenvolvimento de um sistema de Guia Turístico, e está na etapa de manipulação de banco de dados.
- Sua tarefa é inserir os dados no banco criado pela equipe anterior**

28

- O que você deve inserir:
  - Países** (com respectivos continentes): Brasil, Índia, China e Japão;
  - Estados** (com respectivas siglas): Maranhão, São Paulo, Santa Catarina, Rio de Janeiro;
  - Cidades** (com respectivas populações aproximadas): Sorocaba, Déli, Xangaim Tóquio;
  - Pontos turísticos** (com sua especificação: Quinzinho de Barros (Instituição), Parque Estadual do Jalapão (Atrativo), Torre Eiffel (Atrativo), Fogo de Chão (Restaurante).

29

- Em seguida, teste a estrutura do banco realizando as seguintes alterações:
  - Alterar para "Atrativo" o primeiro ponto turístico
  - Alterar o segundo país (Índia) para ter o código "IND"
- Deletar a primeira cidade
- Avalie a estrutura do banco, verificando se tudo ocorreu corretamente!**

30

## Resolução de Situação-Problema

- Em seguida, surgiram alguns questionamentos sobre o sistema:
  - Como identificar as língua(s) nativas dos países?
  - Há alterações que devem ser feitas na estrutura do banco de dados, nas tabelas e no DER?
- Realize estas alterações:**
  - Criação de uma nova tabela (línguaPais)
  - Definição da integridade referencial entre ela e a tabela país

31

## Exclusão de tabelas em banco de dados

32

## Introdução

- Manutenção de dados (incluir, alterar e excluir)
- Respeitar estrutura (restrições)
- A restrição FOREIGN KEY é usada para impedir ações que destruam links entre tabelas

33

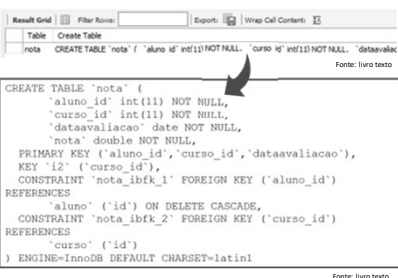
```

1 CREATE TABLE aluno (
2     id INT NOT NULL AUTO-INCREMENT PRIMARY KEY,
3     nome CHAR(50) NOT NULL
4 );
5
6 CREATE TABLE curso (
7     id INT NOT NULL AUTO-INCREMENT PRIMARY KEY,
8     nome CHAR(50) NOT NULL
9 );
10
11 CREATE TABLE nota (
12     aluno_id INT NOT NULL,
13     curso_id INT NOT NULL,
14     dataavaliacao DATE NOT NULL,
15     nota DOUBLE NOT NULL,
16     PRIMARY KEY(aluno_id, curso_id, dataavaliacao),
17     INDEX i2 (curso_id),
18     FOREIGN KEY (aluno_id) REFERENCES aluno(id) ON DELETE
19         CASCADE,
20     FOREIGN KEY (curso_id) REFERENCES curso(id) ON DELETE
21         RESTRICT
22 );

```

Fonte: livro texto

34



```

CREATE TABLE `nota` (
  `aluno_id` int(11) NOT NULL,
  `curso_id` int(11) NOT NULL,
  `dataavaliacao` date NOT NULL,
  `nota` double NOT NULL,
  PRIMARY KEY (`aluno_id`,`curso_id`,`dataavaliacao`),
  KEY `i2` (`curso_id`),
  CONSTRAINT `nota_ibfk_1` FOREIGN KEY (`aluno_id`)
REFERENCES
  `aluno` (`id`) ON DELETE CASCADE,
  CONSTRAINT `nota_ibfk_2` FOREIGN KEY (`curso_id`)
REFERENCES
  `curso` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1

```

Fonte: livro texto

35

## Outras operações

- DROP TABLE:** remover uma ou mais tabelas

```
DROP TABLE [IF EXISTS] nome_tabela [, nome_tabela] ...
```

- Exclusão com restrição de chave estrangeira:

```
ALTER TABLE nota DROP FOREIGN KEY nota_ibfk_1;
DROP TABLE IF EXISTS aluno;
```

36

### Outras operações

- **TRUNCATE TABLE:** instrução para esvaziar uma tabela completamente

```
TRUNCATE [TABLE] nome_tabela;
```

- Instruir o MySQL a "respeitar" as restrições

```
SET FOREIGN_KEY_CHECKS = 1; ← Padrão do MySQL
```

- Instruir o MySQL a "ignorar" as restrições

```
SET FOREIGN_KEY_CHECKS = 0;
```


37

### Resolução de Situação-Problema

- Você está trabalhando no desenvolvimento de um **guia turístico** em sua empresa.
- Você já tem um banco de dados com uma estrutura pronta, relacionamentos criados, chaves e restrições definidas
- No entanto, após o início de um projeto, pode haver a necessidade de revisão
  - **Incluir/excluir estruturas!**

38

### Resolução de Situação-Problema

- Você deve incluir um sistema de coordenadas! 
- **Alterar tabela de Elementos Turísticos**, adicionando campos latitude e longitude (coordenadas).
- **Alterar a tabela "Países"**, adicionando uma nota de 0 a 10 com o nível de interesse para o turista
- **Alterar tabela "Cidades"**, incluindo uma lista com os três melhores restaurantes.
- Após isso, não teremos a necessidade de manter a tabela coordenadas e você poderá excluí-la.

39

### Recapitulando

40

### Recapitulando

- Comandos utilizados na manipulação de bancos de dados
  - SELECT, INSERT, UPDATE, DELETE
- Alteração de tabelas e constraints
  - ALTER TABLE, ADD, DROP, CHANGE, MODIFY
- Exclusão de tabelas em banco de dados
  - DROP TABLE, TRUNCATE TABLE

41