

# Programação e Desenvolvimento de Banco de Dados

## Repositório de Dados

Prof. Dr. Gilberto Fernandes Jr.

1

- Unidade de Ensino: 1
- Competência da Unidade: Conhecer e compreender a linguagem de consulta estruturada e a criação de um banco de dados.
- Resumo: Saber elaborar script SQL para criação de um banco de dados e suas tabelas
- Palavras-chave: SQL, banco de dados, tabela
- Título da Teleaula: Repositório de dados
- Teleaula nº: 1

2

## Contextualização

- Modelagem física
- Linguagem de consulta estruturada (SQL)
- Criação de banco de dados
- Criação de tabelas

3

## Linguagem de consulta estruturada (SQL)

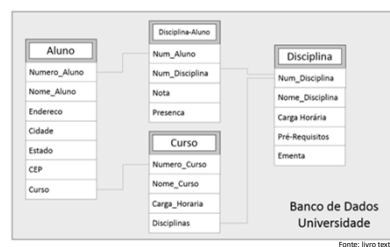
4

## Introdução

- Assuntos importantes em programação em Bancos de Dados
  - **Programação**
  - **Bancos de Dados Relacionais** (manipulação por meio do **SQL**).

5

## Relembrando a representação de um banco



6

## A Linguagem SQL

- Linguagem padrão dos SGBD.
- SQL, do inglês *Structured Query Language*
- É a forma de tratarmos os vários aspectos de um banco de dados:
  - Definir esquemas de relacionamento;
  - Criar restrições em relacionamentos;
  - Realizar consultas interativas;
  - Segurança e controle de atualizações.

7

## A Linguagem SQL

- Nos seus estudos, você utilizará o SGBD **MySQL**.
- Trabalha com várias plataformas, tendo sido escrito em C e C++
- Definir instruções SQL incorporadas de maneira embutida ou dinâmica em linguagens, como por exemplo, Node.js, PHP, C#, C++, Java, Android, Swift, etc.

8

## A Linguagem SQL

- Na linguagem SQL, destacam-se cinco subconjuntos de instruções:
- **Linguagem de Definição de Dados (DDL)**  
CREATE, DROP, ALTER, TRUNCATE
- **Linguagem de Manipulação dos Dados (DML)**  
INSERT, UPDATE, DELETE



Fonte: livro texto

9

## A Linguagem SQL

- **Linguagem de Consulta a Dados (DQL)**  
SELECT, SHOW, HELP
- **Linguagem de Controle de Dados (DCL)**  
GRANT, REVOKE
- **Linguagem de Transação de Dados (DTL)**  
START TRANSACTION, COMMIT, SAVEPOINT, ROLLBACK



Fonte: livro texto

10

## Estrutura básicas de consulta em SQL

"Em um SGBD relacional, teremos repositórios de dados armazenados, mas com um conjunto de tabelas que podem ser identificadas unicamente" (CARDOSO, 2013)

- Consultas são resultados de um produto cartesiano das tabelas especificadas
- Estabelecer **condições precisas** para evitar apresentar resultados enormes e imprecisos!

11

## As Cláusulas SELECT e FROM

- **SELECT** – identificação dos campos desejados.
- **FROM** – lista as tabelas que deverão ser lidas.

```
SELECT Nome
FROM cidade;
```

```
SELECT *
FROM cidade;
```

```
SELECT Nome, Populacao / 2
FROM cidade;
```

```
SELECT DISTINCT Nome
FROM cidade;
```

12

## A Cláusula WHERE

- **WHERE** - Consiste em expressões lógicas envolvendo os campos das tabelas da cláusula FROM.

```
SELECT Nome, Populacao
FROM cidade
WHERE Populacao < 100000;
```

```
SELECT Nome, Populacao
FROM cidade
WHERE Populacao BETWEEN 90000 AND 100000;
```

13

## Operação de renomeação e variáveis do registro

- Os campos podem ser renomeados com a cláusula **AS**
  - nome-antigo **AS** nome-novo

```
SELECT Nome, Populacao AS PopulacaoDaCidade
FROM cidade;
```

```
SELECT C.Nome, C.Populacao, P.população
FROM cidade as C, pais as P
```

**Não é permanente!**  
Renomeia apenas no  
**resultado** da  
consulta

14

## Operações de *String* e ordenação

- O operador **LIKE** determina a correspondência de padrões usando caracteres especiais: % ou \_

```
SELECT Nome FROM cidade WHERE Nome LIKE 'Sor%';
```

- Operador **ORDER BY** para ordenação (ASC ou DESC)

```
SELECT Nome FROM cidade ORDER BY Nome;
```

```
SELECT Nome FROM cidade ORDER BY Nome DESC;
```

15

## Conhecendo o MySQL Community Server

16

## MySQL Server

- O **MySQL Server** é um sistema gerenciador de banco de dados (SGBD) relacional de código aberto.
  - O serviço utiliza a linguagem SQL (Structure Query Language – Linguagem de Consulta Estruturada).
- **MySQL Workbench**: ferramenta gráfica de gerenciamento do banco de dados MySQL.

17

## Explorando um banco de dados

- Vamos explorar um banco de dados no software MySQL Workbench.
- Usar o banco de dados **world** (pré-instalado)
- Utilizar as instruções SQL para:
  - Entender sua estrutura de tabelas
  - Identificar suas chaves primárias e estrangeiras
  - Determinar quais os relacionamentos
  - Estabelecer instruções de consultas.

18

## Criação de um banco de dados e suas tabelas

19

## Planejamento de um banco de dados

Para planejar um banco de dados, entendemos que há cinco passos principais:

1. Coletar informações.
2. Identificar suas principais estruturas.
3. Modelar a estrutura.
4. Identificar nas estruturas os tipos de dados (brutos, de categorização, de identificação, de referência).
5. Identificar quais são seus relacionamentos.

20

## Internacionalização de um banco de dados

- Devemos prestar atenção na abrangência de utilização do banco
  - Respeitar as regras de escrita e gramática ou representação de cada país
- MySQL possui as cláusulas **CHARSET** e **COLLATION**.

21

## Internacionalização de um banco de dados

- **CHARSET**: designa um conjunto de símbolos e codificações e como eles são representados binariamente.
- **COLLATION**: conjunto de regras para comparação de caracteres em um CHARSET.



22

## Internacionalização de um banco de dados

- Como verificar CHARSET/COLLATION instalados no seu MySQL:
  - SHOW CHARACTER SET;
  - SHOW COLLATION;

Exemplos:

- CHARSET: latin1 , COLLATION: latin1\_swedish\_ci
- CHARSET: utf8 , COLLATION: utf8\_general\_ci

23

## Criar um banco de dados

- Deveremos utilizar as instruções da classe da linguagem de definição de dados (DDL).

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_nome
[create_specification] ...
create_specification:
[DEFAULT] CHARACTER SET [=] charset_name
| [DEFAULT] COLLATE [=] collation_name
```

24

## Criar um banco de dados

Exemplos:

```
CREATE DATABASE IF NOT EXISTS mundo
DEFAULT CHARSET = utf8
DEFAULT COLLATE = utf8_general_ci;
```

```
SHOW DATABASES;
```

```
ALTER DATABASE mundo CHARSET = latin1;
```

```
DROP DATABASE IF EXISTS mundo;
```

25

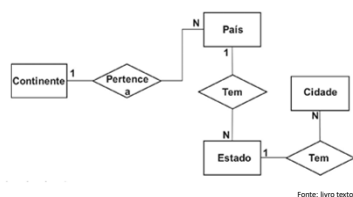
## Modelo de dados

- Diagrama Entidade-Relacionamento (DER)
  - Certeza de que todos os dados existentes em um determinado contexto estão completamente representados.
- Administrador de banco de dados (DBA) utiliza o modelo para gerar o BD físico

26

## Modelo de dados

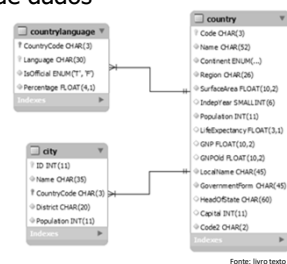
- MER



27

## Modelo de dados

- DER



28

## Criação de tabelas

Sintaxe da estrutura CREATE TABLE:

```
CREATE TABLE [IF NOT EXISTS] nome_tabela (
  Lista_campos
);
```

nome\_campo tipo\_campo[tamanho] [NOT NULL|NULL] [DEFAULT valor] [AUTO\_INCREMENT] [PRIMARY KEY]

29

## Tipos de Dados

Nomenclatura:

- M**: tamanho do dado
  - Seu valor máximo está relacionado ao tipo de dado
- D**: quantidade de dígitos "depois da vírgula".
  - Valor máximo é 30 (ou M-2)
- [ ]: partes opcionais na definição dos tipos.
- UNSIGNED**: não aceita números negativos
- ZEROFILL**: preencher com zeros à esquerda. E recebe o atributo **UNSIGNED**

30

## Tipos Numéricos

- **SMALLINT [(M)] [UNSIGNED] [ZEROFILL]**
  - inteiro no intervalo de -32768 a 32767. O intervalo sem sinal é de 0 a 65535.
- **INT [(M)] [UNSIGNED] [ZEROFILL]**
  - Inteiro no intervalo de -2147483648 a 2147483647. O intervalo sem sinal é de 0 a 4294967295.
- **BIGINT [(M)] [UNSIGNED] [ZEROFILL]**
  - intervalo de -9223372036854775808 a 9223372036854775807.

31

## Tipos numéricos

- **FLOAT [(M, D)] [UNSIGNED] [ZEROFILL]**
  - ponto flutuante, de precisão simples. Os valores admissíveis são -3,402823466E+38 a -1,175494351E-38, 0 e 1,175494351E-38 a 3,402823466E+38.
- **DOUBLE [(M, D)] [UNSIGNED] [ZEROFILL]**
  - ponto flutuante de precisão dupla.
  - -1,7976931348623157E+308 a -2,2250738585072014E-308,0
  - Sinônimo: DOUBLE PRECISION.

32

## Tipo Data e hora

- **DATE**
  - MySQL exibe no formato 'YYYY-MM-DD'
  - Permite números ou strings
- **DATETIME [(fsp)]**
  - Combinação de data e hora
  - formato 'AAAA-MM-DD HH: MM: SS [fracao]'
- **TIME [(fsp)] -> 'HH: MM: SS [fracão]'**
- **YEAR [4] -> 'YYYY'**

**fsp:** representa a quantidade de dígitos das "frações de segundo" que será armazenada.

33

## Tipo Texto

- **VARCHAR (M) [CHARACTER SET charset\_name] [COLLATE collation\_name]**
  - cadeia de comprimento variável.
- **ENUM ('valor1', 'valor2', ...) [CHARACTER SET charset\_name] [COLLATE collation\_name]**
  - objeto de string que pode ter apenas um valor, escolhido na lista de valores 'valor1', 'valor2', [...], NULO, ou vazio.
  - Armazenado como inteiro pelo banco

34

## Criando um banco de dados para um sistema de guia turístico

35

## Descrição da SP

- Você está desenvolvendo um aplicativo para o serviço de um Guia Turístico.
- Fator técnico que caracteriza nossa linguagem:
  - O banco de dados deve utilizar caracteres acentuados
  - Estar preparado para a internacionalização
- **Primeira tarefa: criação do banco de dados**
  - UTF-8 - abrangência de caracteres internacionais.
  - utf8\_general\_ci - regras comuns de comparação

36

### Descrição da SP

- Em seguida, sua responsabilidade no projeto é a de **implantar, sob forma de tabelas, a estrutura do repositório de dados do projeto**
- Estrutura básica do banco:
  - Países – armazenar os dados sobre nome, continente, área, ano de independência, população, expectativa de vida, forma de governo, capital, moeda.
  - Estados – armazenar os dados sobre nome, sigla, região, capital.
  - Cidades – nome, população, data de criação.
  - Pontos de interesses turístico

37

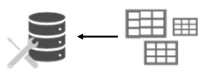
### Descrição da SP

- Os pontos de interesse turístico devem ser classificados como atrativos, serviços, equipamentos, infraestrutura de apoio e instituições ou organizações.
- Para cada elemento turístico haverá uma coordenada (latitude e longitude).
  - Tabela adicional
- O elemento turístico deve ter um campo para mostrar se está publicado ou não.
  - Seu valor padrão é falso.

38

### Resolvendo a SP

- **Vamos construir o banco de dados com o SQL!**



39

### Recapitulando

40

### Recapitulando

- Linguagem de consulta estruturada
  - SELECT, FROM, WHERE
- Criação de banco de dados
- Criação de tabelas
- Tipos de dados

41