

```

1  /*
2  Banco de Dados
3  Prof.ª M.ª Mariana Meirelles de Mello
4  Semestre: 2025.2
5  Data: 21/10/2025
6
7  Aula 01: Modelo Físico de Banco de Dados - Primeiros comandos SQL
8  */
9
10 /*
11 BOAS PRÁTICAS!
12     1. Usar indentação para deixar o código mais legível e claro.
13         - Uma instrução por linha
14         - Letras maiúsculas para comandos SQL e minúsculas para nomes de tabelas e
           colunas
15     2. Nomes significativos.
16         - Não utilizar acentos, espaços e caracteres especiais.
17     3. Utilize comentários em trechos importantes e mais complexos.
18         -- comentario de linha
19 */
20
21 /*
22 Iremos utilizar inicialmente o "MySQL 8.0 Command Line Client - Unicode"
23 Ao utilizar o "MySQL 8.0 Command Line Client - Unicode" devemos prestar muita atenção
24 nos comandos que estamos escrevendo, pois se tivermos algum erro, teremos que
25 reescrever todo o comando que desejamos.
26
27 Em SQL para dizer ao nosso SGBD que finalizamos um comando devemos utilizar o ;
28 (ponto-e-vírgula). Logo, quando terminar um comando não esqueça do ;.
29 */
30
31 /*
32 Quando desejarmos visualizar algum objeto que exista no nosso SGBD devemos utilizar o
33 comando SHOW, seguido do objeto desejado.
34 */
35
36 -- Visualizar as bases de Dados
37 SHOW databases;
38
39 /* Retorno do comando com as databases do meu banco
40 +-----+
41 | Database |
42 +-----+
43 | clinica  |
44 | information_schema |
45 | mysql    |
46 | performance_schema |
47 | sakila   |
48 | sys      |
49 | world    |
50 +-----+
51 */
52
53 -- Para utilizar uma base de dados, utilizamos o comando USE.
54 USE clinica;
55
56 -- Para visualizar as tabelas dessa base de dados, utilizamos o comando SHOW visto
57 anteriormente.
58 SHOW TABLES;
59
60 /* Retorno do comando:
61 +-----+
62 | Tables_in_clinica |
63 +-----+
64 | consulta          |
65 | medico            |
66 | paciente          |
67 +-----+
68 */
69
70 /*
71 Se precisarmos ver a estrutura dos campos de uma tabela podemos utilizar o comando
72 DESC de describe.

```

```

67  Sintaxe: DESC nome_tabela
68  */
69  DESC consulta;
70
71  /* Retorno do comando:
72
73  +-----+-----+-----+-----+-----+-----+
74  | Field          | Type          | Null | Key | Default | Extra          |
75  +-----+-----+-----+-----+-----+-----+
76  | idConsulta     | int           | NO   | PRI | NULL    | auto_increment |
77  | dt_consulta    | date          | NO   |     | NULL    |                |
78  | diagnostico    | varchar(200)  | YES  |     | NULL    |                |
79  | fk_idPaciente  | int           | NO   | MUL | NULL    |                |
80  | fk_idMedico    | int           | NO   | MUL | NULL    |                |
81  +-----+-----+-----+-----+-----+-----+
82  */
83
84  /*
85  Agora vamos criar a nossa primeira base de dados de exemplo para treinarmos os
86  primeiros comandos de SQL.
87  Sempre para criar algo, seja uma base de dados, uma tabela, uma view utilizamos o
88  comando: CREATE
89
90  Sintaxe: CREATE <OBJETO> <nome>.
91
92  O nome do nosso primeiro banco será, aula01_noite.
93  */
94
95  CREATE DATABASE aula01_noite;
96
97  -- Ao visualizarmos nossas databases, a que acabamos de criar já aparecerá no nosso
98  resultado:
99  SHOW DATABASES;
100
101  /* Retorno do comando:
102
103  +-----+
104  | Database          |
105  +-----+
106  | aula01_noite     |
107  | clinica           |
108  | information_schema |
109  | mysql             |
110  | performance_schema |
111  | sakila            |
112  | sys               |
113  | world             |
114  +-----+
115  */
116
117  /*
118  Para começarmos a trabalhar com essa base, criar tabelas, inserir dados etc,
119  precisamos dizer para o MySQL que queremos usar a base "aula01_noite". Para isso,
120  temos que utilizar o comando USE <base_dados>;
121  */
122
123  USE aula01_noite;
124
125  -- Vamos criar a nossa primeira tabela!
126  /*
127  Sempre para criar algum objeto utilizamos o comando CREATE. Como queremos criar uma
128  tabela, utilizaremos este comando seguido de "table" e o nome que você deseja para a
129  sua tabela.
130  Sintaxe:
131
132  Estrutura genérica para criação de uma tabela. As restrições não são obrigatórias.
133  CREATE TABLE nome_tabela(
134      atributo1      tipo      [restricoes],
135      atributo2      tipo      [restricoes],
136      atributoN      tipo      [restricoes]
137  );
138
139  A nossa primeira tabela será de 'produto' e deve ter os seguintes atributos:
140  - idProduto: chave primária e de controle do SGBD

```

```

133     - nome: campo obrigatório
134     - preco: campo obrigatório
135     - dt_validade
136 */
137
138 CREATE TABLE produto(
139     idProduto INT PRIMARY KEY AUTO_INCREMENT,
140     nome VARCHAR(100) NOT NULL,
141     preco FLOAT(8,2) NOT NULL,
142     dt_validade DATE
143 );
144
145 /*
146 PRIMARY KEY - é a cláusula utilizada para definir um atributo como chave primária
147 NOT NULL - define que aquele atributo não pode receber valor nulo, ou seja, temos que
148 obrigatoriamente definir um valor ao inserir dados na tabela para aquele atributo
149 AUTO_INCREMENT - é a cláusula utilizada para passar o controle do incremento da chave
150 primária para o SGBD.
151 */
152
153 /*
154 Visualizar as tabelas que existem na nossa base de dados através do comando SHOW.
155 */
156 SHOW TABLES;
157
158 /* Retorno do comando:
159 +-----+
160 | Tables_in_aula01_noite |
161 +-----+
162 | produto                |
163 +-----+
164 */
165
166 -- Visualizar a estrutura da nossa tabela.
167 DESC produto;
168
169 /* Retorno do comando:
170 +-----+-----+-----+-----+-----+-----+
171 | Field      | Type          | Null | Key | Default | Extra          |
172 +-----+-----+-----+-----+-----+-----+
173 | idProduto  | int           | NO   | PRI | NULL    | auto_increment |
174 | nome       | varchar(100)  | NO   |     | NULL    |                 |
175 | preco      | float(8,2)    | NO   |     | NULL    |                 |
176 | dt_validade | date          | YES  |     | NULL    |                 |
177 +-----+-----+-----+-----+-----+-----+
178 */
179
180 /*
181 Após criarmos uma tabela, pode ser necessário alguma alteração na sua estrutura. Para
182 alterar campos de uma tabela, utilizamos o comando ALTER TABLE juntamente com alguma
183 das cláusulas abaixo:
184
185 CHANGE - permite alterar o nome do campo, tipo e restrições
186 MODIFY - pode fazer tudo o que o CHANGE executa, menos renomear um campo
187 ADD - adiciona um novo campo a tabela já criada
188
189 Vamos realizar algumas alterações na tabela que acabamos de criar.
190 */
191
192 -- Renomear a tabela para 'produto' para 'produtos':
193
194 ALTER TABLE produto
195 RENAME TO produtos;
196
197 SHOW TABLES;
198
199 /*Retorno do comando:
200 +-----+
201 | Tables_in_aula01_noite |
202 +-----+
203 | produtos                |
204 +-----+
205 */

```

```

202
203 -- Observe que agora o nome da nossa tabela é 'produtos'.
204
205 /*
206 Para alterar o nome de um atributo devemos usar a cláusula CHANGE junto com o ALTER
207 TABLE
208 Sintaxe:
209 ALTER TABLE <nome_da_tabela>
210 CHANGE 'nome_atual' 'novo_nome' <tipo> <restricoes>;
211
212 Alterando o nome do atributo 'preco' para 'valor':
213 */
214 ALTER TABLE produtos
215 CHANGE preco valor FLOAT(8,2) NOT NULL;
216
217 DESC produtos;
218
219 /* Retorno do comando:
220 +-----+-----+-----+-----+-----+-----+
221 | Field      | Type          | Null | Key | Default | Extra          |
222 +-----+-----+-----+-----+-----+-----+
223 | idProduto  | int           | NO   | PRI | NULL    | auto_increment |
224 | nome       | varchar(100)  | NO   |     | NULL    |                 |
225 | valor      | float(8,2)    | NO   |     | NULL    |                 |
226 | dt_validade | date          | YES  |     | NULL    |                 |
227 +-----+-----+-----+-----+-----+-----+
228 */
229
230 /*
231 Alterar o tipo de um atributo.
232 Podemos utilizar o CHANGE ou o MODIFY. Vamos ver a sintaxe com o MODIFY.
233
234 Sintaxe:
235 ALTER TABLE <nome_da_tabela>
236 MODIFY <atributo> <tipo> <restricoes>;
237
238 *Restrições, se houver.
239 Não podemos utilizar o MODIFY para renomear um atributo, apenas o CHANGE.
240
241 Vamos alterar o tipo do atributo 'valor' para 'decimal'
242 */
243
244 ALTER TABLE produtos
245 MODIFY valor DECIMAL(7,2) NOT NULL;
246
247 DESC produtos;
248
249 /* Retorno do comando:
250 +-----+-----+-----+-----+-----+-----+
251 | Field      | Type          | Null | Key | Default | Extra          |
252 +-----+-----+-----+-----+-----+-----+
253 | idProduto  | int           | NO   | PRI | NULL    | auto_increment |
254 | nome       | varchar(100)  | NO   |     | NULL    |                 |
255 | valor      | decimal(7,2)  | NO   |     | NULL    |                 |
256 | dt_validade | date          | YES  |     | NULL    |                 |
257 +-----+-----+-----+-----+-----+-----+
258 */
259
260 /*
261 Vou deixar aqui o exemplo de como fazer a alteração acima utilizando o CHANGE no
262 lugar do MODIFY.
263 (Não precisa executar, é apenas um exemplo para vocês verem a sintaxe utilizando
264 outra cláusula para executar a mesma tarefa.)
265
266 ALTER TABLE produtos
267 CHANGE valor valor DECIMAL(7,2) NOT NULL;
268
269 -- Até a próxima aula!

```