

```

1  -- =====
2  -- COLA COMPLETA DE SQL PARA ESTUDOS
3  -- =====
4
5  -- =====
6  -- 1. SELEÇÃO DE DADOS
7  -- =====
8  SELECT * FROM tabela; -- Seleciona todas as colunas
9  SELECT coluna1, coluna2 FROM tabela; -- Seleciona colunas específicas
10 SELECT coluna1 AS novo_nome FROM tabela; -- Renomeia coluna com alias
11
12 -- =====
13 -- 2. FILTRAGEM
14 -- =====
15 SELECT * FROM tabela WHERE coluna = 'valor'; -- Filtra registros
16 -- Operadores: =, <>, >, <, >=, <=
17 SELECT * FROM tabela WHERE coluna1 = 'x' AND coluna2 > 10; -- Combina condições
18 SELECT * FROM tabela WHERE coluna BETWEEN 10 AND 20; -- Intervalo
19 SELECT * FROM tabela WHERE coluna IN ('A', 'B', 'C'); -- Lista de valores
20 SELECT * FROM tabela WHERE coluna LIKE 'Jo%'; -- Começa com Jo
21 SELECT * FROM tabela WHERE coluna LIKE '%ão'; -- Termina com ão
22
23 -- =====
24 -- 3. ORDENAÇÃO E LIMITE
25 -- =====
26 SELECT * FROM tabela ORDER BY coluna ASC; -- Ordena crescente
27 SELECT * FROM tabela ORDER BY coluna DESC; -- Ordena decrescente
28 SELECT * FROM tabela LIMIT 10; -- Limita a 10 registros
29
30 -- =====
31 -- 4. FUNÇÕES DE AGREGAÇÃO
32 -- =====
33 SELECT COUNT(*) FROM tabela; -- Conta registros
34 SELECT AVG(coluna) FROM tabela; -- Média
35 SELECT SUM(coluna) FROM tabela; -- Soma
36 SELECT MIN(coluna) FROM tabela; -- Menor valor
37 SELECT MAX(coluna) FROM tabela; -- Maior valor
38
39 -- =====
40 -- 5. AGRUPAMENTO
41 -- =====
42 SELECT coluna, COUNT(*)
43 FROM tabela
44 GROUP BY coluna; -- Agrupa por coluna
45
46 SELECT coluna, COUNT(*)
47 FROM tabela
48 GROUP BY coluna
49 HAVING COUNT(*) > 5; -- Filtra grupos
50
51 -- =====
52 -- 6. JUNÇÕES (JOINS)
53 -- =====
54 SELECT a.coluna, b.coluna
55 FROM tabelaA a
56 INNER JOIN tabelaB b ON a.id = b.id; -- INNER JOIN
57
58 SELECT a.coluna, b.coluna
59 FROM tabelaA a
60 LEFT JOIN tabelaB b ON a.id = b.id; -- LEFT JOIN
61
62 SELECT a.coluna, b.coluna
63 FROM tabelaA a
64 RIGHT JOIN tabelaB b ON a.id = b.id; -- RIGHT JOIN
65
66 -- =====
67 -- 7. MANIPULAÇÃO DE DADOS
68 -- =====
69 INSERT INTO tabela (coluna1, coluna2) VALUES ('valor1', 'valor2'); -- Inserir
70 UPDATE tabela SET coluna1 = 'novo_valor' WHERE id = 1; -- Atualizar
71 DELETE FROM tabela WHERE id = 1; -- Deletar
72
73 -- =====

```

```

74 -- 8. FUNÇÕES DE JANELA (AVANÇADO)
75 -- =====
76 SELECT coluna,
77         ROW_NUMBER() OVER (ORDER BY coluna) AS num_linha,      -- Numeração
78         RANK() OVER (ORDER BY coluna) AS ranking,              -- Ranking
79         SUM(coluna) OVER (PARTITION BY grupo) AS soma_por_grupo -- Soma por grupo
80 FROM tabela;
81
82 -- =====
83 -- COLA DE CRIAÇÃO DE TABELAS
84 -- =====
85
86 -- =====
87 -- 1. TABELA SIMPLES
88 -- =====
89 CREATE TABLE clientes (
90     id INT PRIMARY KEY,      -- chave primária
91     nome VARCHAR(100) NOT NULL, -- texto até 100 caracteres, obrigatório
92     email VARCHAR(150) UNIQUE -- não pode repetir
93 );
94
95 -- =====
96 -- 2. TIPOS DE DADOS COMUNS
97 -- =====
98 CREATE TABLE produtos (
99     id INT PRIMARY KEY AUTO_INCREMENT, -- numeração automática
100    nome VARCHAR(200) NOT NULL,          -- texto
101    preco DECIMAL(10,2) NOT NULL,        -- número com 2 casas decimais
102    estoque INT DEFAULT 0,               -- valor padrão
103    criado_em DATE,                      -- apenas data
104    atualizado_em TIMESTAMP               -- data e hora
105 );
106
107 -- =====
108 -- 3. CHAVES ESTRANGEIRAS
109 -- =====
110 CREATE TABLE pedidos (
111     id INT PRIMARY KEY AUTO_INCREMENT,
112     cliente_id INT NOT NULL,
113     data_pedido DATE NOT NULL,
114     valor_total DECIMAL(10,2),
115     FOREIGN KEY (cliente_id) REFERENCES clientes(id) -- chave estrangeira
116 );
117
118 -- =====
119 -- 4. RESTRIÇÕES
120 -- =====
121 CREATE TABLE funcionarios (
122     id INT PRIMARY KEY AUTO_INCREMENT,
123     nome VARCHAR(100) NOT NULL,
124     salario DECIMAL(10,2) CHECK (salario > 0), -- restrição: salário positivo
125     cargo VARCHAR(50) NOT NULL,
126     departamento VARCHAR(50) DEFAULT 'Geral'
127 );
128
129 -- =====
130 -- 5. ÍNDICES
131 -- =====
132 CREATE TABLE vendas (
133     id INT PRIMARY KEY AUTO_INCREMENT,
134     produto_id INT NOT NULL,
135     quantidade INT NOT NULL,
136     data_venda DATE NOT NULL,
137     FOREIGN KEY (produto_id) REFERENCES produtos(id)
138 );
139
140 CREATE INDEX idx_venda_data ON vendas(data_venda); -- índice para buscas rápidas
141
142 -- =====
143 -- 6. CHAVE COMPOSTA
144 -- =====
145 CREATE TABLE curso_aluno (
146     curso_id INT NOT NULL,

```

```
147     aluno_id INT NOT NULL,
148     PRIMARY KEY (curso_id, aluno_id), -- chave composta
149     FOREIGN KEY (curso_id) REFERENCES cursos(id),
150     FOREIGN KEY (aluno_id) REFERENCES alunos(id)
151 );
152
153 -- =====
154 -- 7. TABELA TEMPORÁRIA
155 -- =====
156 CREATE TEMPORARY TABLE temp_relatorio (
157     id INT,
158     descricao VARCHAR(200)
159 );
160
161 -- =====
162 -- 8. ENUM (MySQL)
163 -- =====
164 CREATE TABLE status_pedido (
165     id INT PRIMARY KEY AUTO_INCREMENT,
166     status ENUM('Pendente', 'Pago', 'Cancelado') NOT NULL
167 );
168
```