

## BD2\Estudos-Prova1\Exercicio-Revisao.sql

```
11 CREATE TABLE engenheiro(  
12     codeng SERIAL NOT NULL,  
13     nome VARCHAR(30),  
14     salario DECIMAL(15,2),  
15  
16     CONSTRAINT pk_eng PRIMARY KEY(codeng)  
17 );  
18  
19 CREATE TABLE projeto(  
20     codproj SERIAL NOT NULL,  
21     titulo VARCHAR(30),  
22     area VARCHAR(30),  
23  
24     CONSTRAINT pk_proj PRIMARY KEY(codproj)  
25 );  
26  
27 CREATE TABLE atuacao(  
28     codeng INT NOT NULL,  
29     codproj INT NOT NULL,  
30     funcao VARCHAR(30),  
31  
32     CONSTRAINT pk_atuacao PRIMARY KEY(codeng, codproj),  
33     CONSTRAINT fk_eng FOREIGN KEY(codeng) REFERENCES engenheiro(codeng),  
34     CONSTRAINT fk_codproj FOREIGN KEY(codproj) REFERENCES projeto(codproj)  
35 )  
36  
37 -- DROP TABLE atuacao;  
38  
39 SELECT * FROM engenheiro;  
40 SELECT * FROM projeto;  
41 SELECT * FROM atuacao;  
42  
43 -- INSERTS  
44  
45 -- Q6  
46 INSERT INTO engenheiro(codeng, nome, salario) VALUES  
47 (default, 'Marcelo Bruno', 15000.00),  
48 (default, 'Lucas Kaique', 16000.00),  
49 (default, 'Silas Leao', 17000.00),  
50 (default, 'Luiz Fernando', 18050.00)  
51  
52 UPDATE engenheiro  
53 SET salario = 10000.00  
54 WHERE codeng = 4;  
55  
56 -- Q7  
57 INSERT INTO projeto(codproj, titulo, area) VALUES  
58 (default, 'Desenvolvimento', 'dados'),  
59 (default, 'Dashboards', 'business intelligence'),  
60 (default, 'Testes', 'quality assurance')  
61  
62 -- Q8  
63 INSERT INTO atuacao(codeng, codproj, funcao) VALUES  
64 (1,1, 'analista de dados'),  
65 (2,3, 'QA pleno'),  
66 (3,2, 'analista senior')  
67
```

```
68 -- Q9
69 -- a
70 SELECT nome
71 FROM engenheiro
72 WHERE salario < 15000;
73
74 -- b
75 -- USANDO JOIN
76 SELECT e.nome
77 FROM engenheiro e
78 JOIN atuacao a
79 ON e.codeng = a.codeng
80 WHERE a.funcao LIKE '%analista%';
81
82 -- USANDO SUBQUERY
83 -- SELECT nome
84 -- FROM engenheiro
85 -- WHERE nome in (
86 --     SELECT
87 -- )
88
89 -- c. Mostre a quantidade de engenheiros por área de projeto.
90 SELECT p.area AS "Area Projeto", COUNT(a.codeng) AS "QTD FUNCIONARIOS"
91 FROM projeto p
92 JOIN atuacao a
93 ON a.codproj = p.codproj
94 GROUP BY p.area;
95
96 -- d. Verifique os nomes dos engenheiros que ganham acima da média salarial de todos
97 SELECT nome
98 FROM engenheiro
99 WHERE salario > (
100     SELECT AVG(salario)
101     FROM engenheiro
102 );
103
104 -- Q10
105 SELECT nome
106 FROM engenheiro
107 WHERE codeng IN (
108     SELECT codeng
109     FROM atuacao
110     WHERE codproj IN (
111         SELECT codproj
112         FROM projeto
113         WHERE area LIKE '%dados%'));
114
115 /* Resposta:
116 A consulta primeiramente procura pelo código do projeto cuja
117 área envolva qualquer palavra relacionada a dados na base Projeto.
118 Em seguida, busca na base Atuacao
119
120 */
121
122 -- Q11
123 SELECT codeng
124 FROM engenheiro
125 WHERE salario > 2200
126 INTERSECT
127 SELECT codeng
```

```
128         FROM atuacao;
129
130 SELECT e.codeng
131 FROM engenheiro e
132 JOIN atuacao a
133 ON e.codeng = a.codeng
134 WHERE salario > 2200;
135
136
137 /* Q12. Crie uma view mostrando os nomes dos engenheiros,
138 sua função em cada projeto e o título do projeto. Consulte-a (0,2) */
139
140 CREATE OR REPLACE VIEW EngProj(nome, funcao, titulo) AS
141 SELECT e.nome, a.funcao, p.titulo
142 FROM engenheiro e
143 JOIN atuacao a ON e.codeng = a.codeng
144 JOIN projeto p ON p.codproj = a.codproj;
145
146 SELECT * FROM EngProj;
147
148
149 SELECT * FROM engenheiro;
150 SELECT * FROM projeto;
151 SELECT * FROM atuacao;
152
153 -- Q13
154 CREATE ROLE gerente LOGIN
155 PASSWORD 'manager';
156 GRANT SELECT ON eng
157 TO gerente;
158
159 /* Q14. Mostre os engenheiros cadastrados que não possuem projetos vinculados.
160 Quais estratégias de operadores de consultas podem ser usadas nesta questão? */
161 SELECT e.nome
162 FROM engenheiro e
163 LEFT JOIN atuacao a
164 ON e.codeng = a.codeng
165 WHERE a.codproj IS NULL;
```