



**INSTITUTO
FEDERAL**

Santa Catarina

Câmpus
São José

Laboratório 3

Consulta sobre múltiplas relações

Curso: Engenharia de Telecomunicações
Disciplina: BCD29008 - Banco de Dados
Professor: Ramon Hugo de Souza

Aluna
Luiza Kuze Gomes

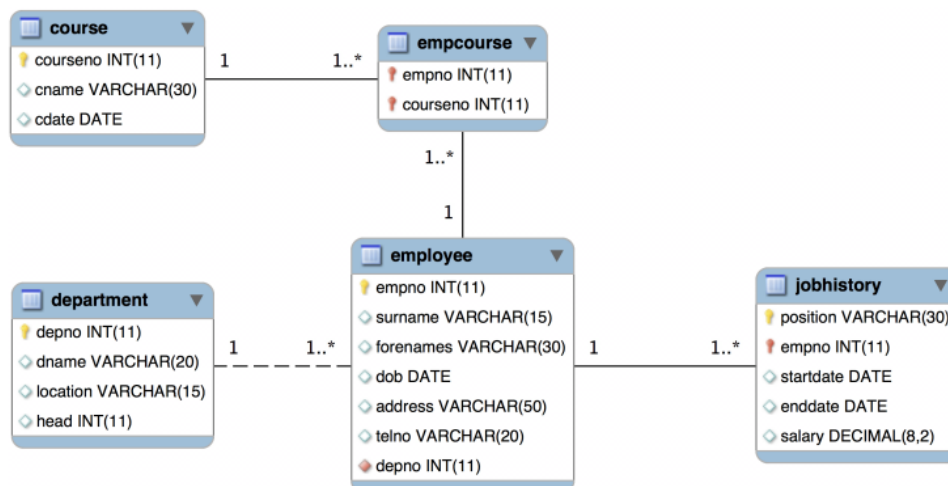
5 de junho de 2025

Sumário

1	Esquema <i>Jobs</i>	2
1.1	Questões 1 a 15	2
2	Relacionamento identificador e consultas com junções	4
2.1	Questão 16	5
2.2	Questão 17	6
2.3	Questão 18	7
2.4	Questões 19 a 21	7
2.5	Questões 22 a 24	8
3	Esquema <i>Sakila</i>	9
3.1	Questões 25 a 32	9

1 Esquema Jobs

Figura 1: Esquema Jobs



1.1 Questões 1 a 15

```
1  -- 1. Nome, sobrenome e dia dos funcionários aniversariantes em março
2  SELECT forenames, surname, dob
3  FROM employee
4  WHERE MONTH(dob) = 3;
5
6  -- 2. Funcionários que assumiram o cargo de "Accountant" em 1980
7  SELECT forenames, surname
8  FROM employee e
9  JOIN jobhistory j ON e.empno = j.empno
10 WHERE position = 'Accountant'
11     AND YEAR(startdate) = 1980;
12
13 -- 3. Funcionários que ainda trabalham na empresa (cargo atual, sem enddate)
14 SELECT empno, forenames
15 FROM employee
16 WHERE empno IN (
17     SELECT empno
18     FROM jobhistory
19     WHERE enddate IS NULL
20 );
21
22 -- 4. Nome e cargo atual de todos os funcionários
23 SELECT e.forenames, e.surname, j.position
24 FROM employee e
25 JOIN jobhistory j ON e.empno = j.empno
26 WHERE j.enddate IS NULL;
27
28 -- 5. Total de salários por departamento
29 SELECT d.dname, SUM(j.salary) AS total_salario
30 FROM employee e
31 JOIN department d ON e.depno = d.depno
32 JOIN jobhistory j ON e.empno = j.empno
33 GROUP BY d.dname;
34
35 -- 6. Total da folha de pagamento
36 SELECT SUM(salary) AS total_folha
37 FROM jobhistory;
```

```

38
39 -- 7. Funcionários que nunca fizeram um curso
40 SELECT empno, forenames, surname
41 FROM employee
42 WHERE empno NOT IN (
43     SELECT DISTINCT empno FROM empcourse
44 );
45
46 -- 8. Pessoas que fizeram curso em 1988
47 SELECT DISTINCT e.forenames, e.surname
48 FROM employee e
49 JOIN empcourse ec ON e.empno = ec.empno
50 JOIN course c ON ec.courseno = c.courseno
51 WHERE YEAR(c.cdate) = 1988;
52
53 -- 9. Nome do departamento e total de funcionários por departamento
54 SELECT d.dname, COUNT(e.empno) AS total_funcionarios
55 FROM department d
56 LEFT JOIN employee e ON d.depno = e.depno
57 GROUP BY d.dname;
58
59 -- 10. Nome do departamento, nome do curso, total de pessoas por curso
60 SELECT d.dname, c.cname, COUNT(DISTINCT ec.empno) AS total_pessoas
61 FROM empcourse ec
62 JOIN employee e ON ec.empno = e.empno
63 JOIN department d ON e.depno = d.depno
64 JOIN course c ON ec.courseno = c.courseno
65 GROUP BY d.dname, c.cname;
66
67 -- 11. Maior salário por departamento
68 SELECT d.dname, MAX(j.salary) AS maior_salario
69 FROM jobhistory j
70 JOIN employee e ON j.empno = e.empno
71 JOIN department d ON e.depno = d.depno
72 GROUP BY d.dname;
73
74 -- 12. Menor entre os maiores salários de cada departamento
75 SELECT MIN(maior_salario) AS menor_dos_maiores
76 FROM (
77     SELECT MAX(j.salary) AS maior_salario
78     FROM jobhistory j
79     JOIN employee e ON j.empno = e.empno
80     JOIN department d ON e.depno = d.depno
81     GROUP BY d.dname
82 ) AS maiores;
83
84 -- 13. Migrar todos os funcionários de "accounts" para "engineering"
85 UPDATE employee
86 SET depno = (
87     SELECT depno FROM department WHERE dname = 'engineering'
88 )
89 WHERE depno = (
90     SELECT depno FROM department WHERE dname = 'accounts'
91 );
92
93 -- 14. Excluir todos os funcionários do departamento "accounts"
94 DELETE FROM employee
95 WHERE depno = (
96     SELECT depno FROM department WHERE dname = 'accounts'
97 );
98
99 -- 15. Excluir todos os funcionários do departamento "engineering"
100 --- Excluir primeiro da tabela empcourse

```

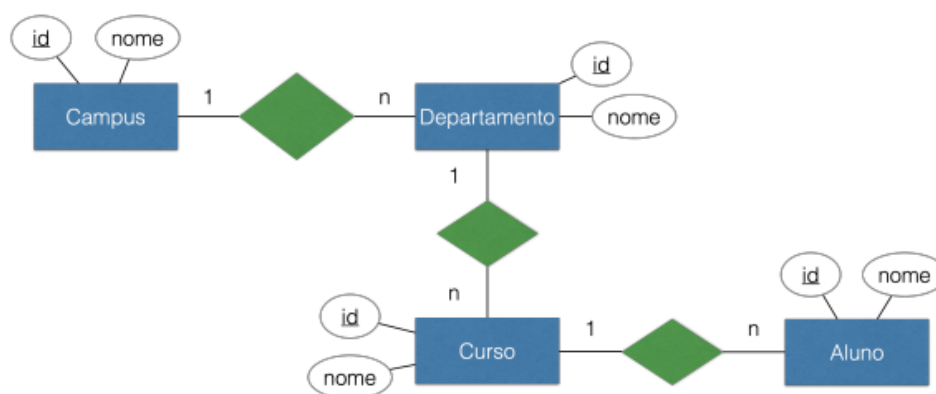
```

101 DELETE FROM empcourse
102 WHERE empno IN (
103     SELECT empno FROM employee
104     WHERE depno = (
105         SELECT depno FROM department WHERE dname = 'engineering'
106     )
107 );
108 -- Depois excluir efetivamente
109 DELETE FROM employee
110 WHERE depno = (
111     SELECT depno FROM department WHERE dname = 'engineering'
112 );

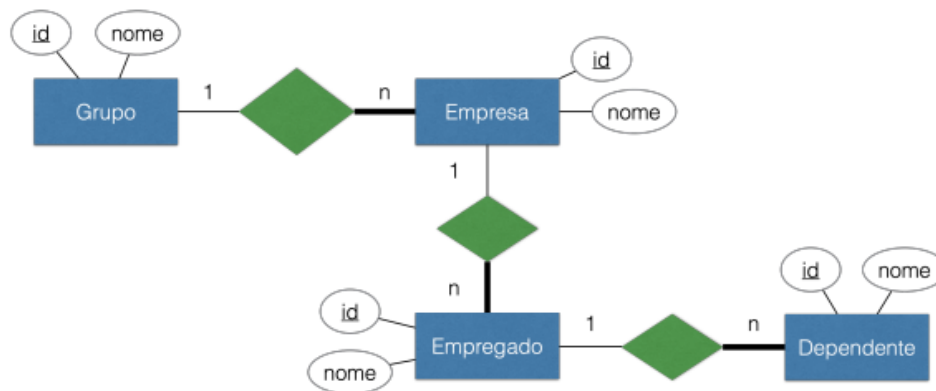
```

2 Relacionamento identificador e consultas com junções

Figura 2: Esquemas sistema acadêmico e empresa



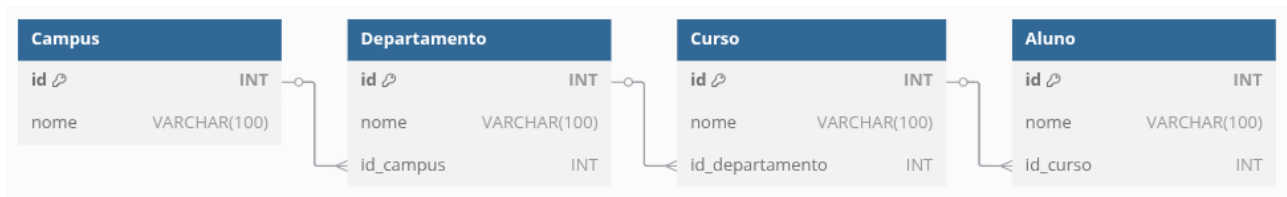
(a) Esquema sistema acadêmico



(b) Esquema empresa

2.1 Questão 16

Figura 3: Modelagem do sistema acadêmico

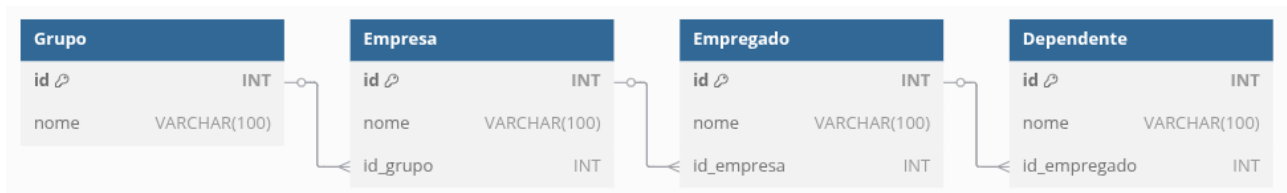


```

1  -- Esquema Sistema Acadêmico
2  CREATE DATABASE sistema_academico;
3  USE sistema_academico;
4
5  CREATE TABLE Campus (
6      id INT PRIMARY KEY,
7      nome VARCHAR(100) UNIQUE
8  );
9
10 CREATE TABLE Departamento (
11     id INT PRIMARY KEY,
12     nome VARCHAR(100) UNIQUE,
13     id_campus INT,
14     FOREIGN KEY (id_campus) REFERENCES Campus(id)
15 );
16
17 CREATE TABLE Curso (
18     id INT PRIMARY KEY,
19     nome VARCHAR(100),
20     id_departamento INT,
21     FOREIGN KEY (id_departamento) REFERENCES Departamento(id)
22 );
23
24 CREATE TABLE Aluno (
25     id INT PRIMARY KEY,
26     nome VARCHAR(100),
27     id_curso INT,
28     FOREIGN KEY (id_curso) REFERENCES Curso(id)
29 );

```

Figura 4: Modelagem da empresa



```

1  -- Esquema Empresa
2  CREATE DATABASE empresa;
3  USE empresa;
4
5  CREATE TABLE Grupo (
6      id INT PRIMARY KEY,

```

```

7     nome VARCHAR(100) UNIQUE
8 );
9
10 CREATE TABLE Empresa (
11     id INT PRIMARY KEY,
12     nome VARCHAR(100) UNIQUE,
13     id_grupo INT,
14     FOREIGN KEY (id_grupo) REFERENCES Grupo(id)
15 );
16
17 CREATE TABLE Empregado (
18     id INT PRIMARY KEY,
19     nome VARCHAR(100),
20     id_empresa INT,
21     FOREIGN KEY (id_empresa) REFERENCES Empresa(id)
22 );
23
24 CREATE TABLE Dependente (
25     id INT PRIMARY KEY,
26     nome VARCHAR(100),
27     id_empregado INT,
28     FOREIGN KEY (id_empregado) REFERENCES Empregado(id)
29 );

```

2.2 Questão 17

```

1  -- Inserção de dados no esquema sistema acadêmico
2  -- Inserir Campi
3  INSERT INTO Campus (id, nome) VALUES (1, 'São José'), (2, 'Florianópolis'), (3, 'Lages');
4
5  -- Inserir Departamentos
6  INSERT INTO Departamento (id, nome, id_campus) VALUES
7      (1, 'Telecomunicações', 1),
8      (2, 'Refrigeração', 1),
9      (3, 'Engenharia', 2),
10     (4, 'Computação', 3);
11
12 -- Inserir Cursos
13 INSERT INTO Curso (id, nome, id_departamento) VALUES
14     (1, 'Engenharia de Telecomunicações', 1),
15     (2, 'Análise e Desenvolvimento de Sistemas', 1),
16     (3, 'Engenharia Elétrica', 3),
17     (4, 'Ciências da Computação', 4);
18
19 -- Inserir Alunos
20 -- Eng. Telecomunicações
21 INSERT INTO Aluno (id, nome, id_curso) VALUES
22     (1, 'Aluno A', 1),
23     (2, 'Aluno B', 1),
24     (3, 'Aluno C', 1),
25     (4, 'Aluno D', 1),
26     (5, 'Aluno E', 1);
27
28 -- ADS
29 INSERT INTO Aluno (id, nome, id_curso) VALUES
30     (6, 'Aluno F', 2),
31     (7, 'Aluno G', 2),
32     (8, 'Aluno H', 2),
33     (9, 'Aluno I', 2);
34
35 -- Eng. Elétrica

```

```

36 INSERT INTO Aluno (id, nome, id_curso) VALUES
37     (10, 'Aluno J', 3),
38     (11, 'Aluno K', 3);
39
40 -- Ciências da Computação
41 INSERT INTO Aluno (id, nome, id_curso) VALUES
42     (12, 'Aluno L', 4);

```

2.3 Questão 18

```

1  -- inserção de dados sobre o esquema empresa
2  -- Inserir Grupos
3  INSERT INTO Grupo (id, nome) VALUES (1, 'ABC'), (2, 'DEF'), (3, 'GHI');
4
5  -- Inserir Empresas
6  INSERT INTO Empresa (id, nome, id_grupo) VALUES
7      (1, 'O baratão', 1),
8      (2, 'A moda', 1),
9      (3, 'Popular', 2),
10     (4, 'Exclusiva', 3);
11
12 -- Inserir Empregados
13 INSERT INTO Empregado (id, nome, id_empresa) VALUES
14     (1, 'Empregado A', 1),
15     (2, 'Empregado B', 1),
16     (3, 'Empregado C', 2),
17     (4, 'Empregado D', 3),
18     (5, 'Empregado E', 4);
19
20 -- Inserir Dependentes
21 -- 5 dependentes do Empregado A
22 INSERT INTO Dependente (id, nome, id_empregado) VALUES
23     (1, 'Dep A1', 1),
24     (2, 'Dep A2', 1),
25     (3, 'Dep A3', 1),
26     (4, 'Dep A4', 1),
27     (5, 'Dep A5', 1);
28
29 -- 4 dependentes do Empregado C
30 INSERT INTO Dependente (id, nome, id_empregado) VALUES
31     (6, 'Dep C1', 3),
32     (7, 'Dep C2', 3),
33     (8, 'Dep C3', 3),
34     (9, 'Dep C4', 3);
35
36 -- 2 dependentes do Empregado D
37 INSERT INTO Dependente (id, nome, id_empregado) VALUES
38     (10, 'Dep D1', 4),
39     (11, 'Dep D2', 4);
40
41 -- 1 dependente do Empregado E
42 INSERT INTO Dependente (id, nome, id_empregado) VALUES
43     (12, 'Dep E1', 5);

```

2.4 Questões 19 a 21

```

1  -- Consultas sobre o esquema sistema acadêmico
2  -- 19. Nome dos cursos e o nome do campus onde é ofertado

```



```

3 SELECT c.nome AS curso, ca.nome AS campus
4 FROM Curso c
5 JOIN Departamento d ON c.id_departamento = d.id
6 JOIN Campus ca ON d.id_campus = ca.id;
7
8 -- 20. Nome dos alunos do campus "São José"
9 SELECT a.nome
10 FROM Aluno a
11 JOIN Curso c ON a.id_curso = c.id
12 JOIN Departamento d ON c.id_departamento = d.id
13 JOIN Campus ca ON d.id_campus = ca.id
14 WHERE ca.nome = 'São José';
15
16 -- 21. Campus, departamento, curso e total de alunos por curso
17 SELECT ca.nome AS campus, d.nome AS departamento, c.nome AS curso, COUNT(a.id) AS total_alunos
18 FROM Aluno a
19 JOIN Curso c ON a.id_curso = c.id
20 JOIN Departamento d ON c.id_departamento = d.id
21 JOIN Campus ca ON d.id_campus = ca.id
22 GROUP BY ca.nome, d.nome, c.nome;

```

2.5 Questões 22 a 24

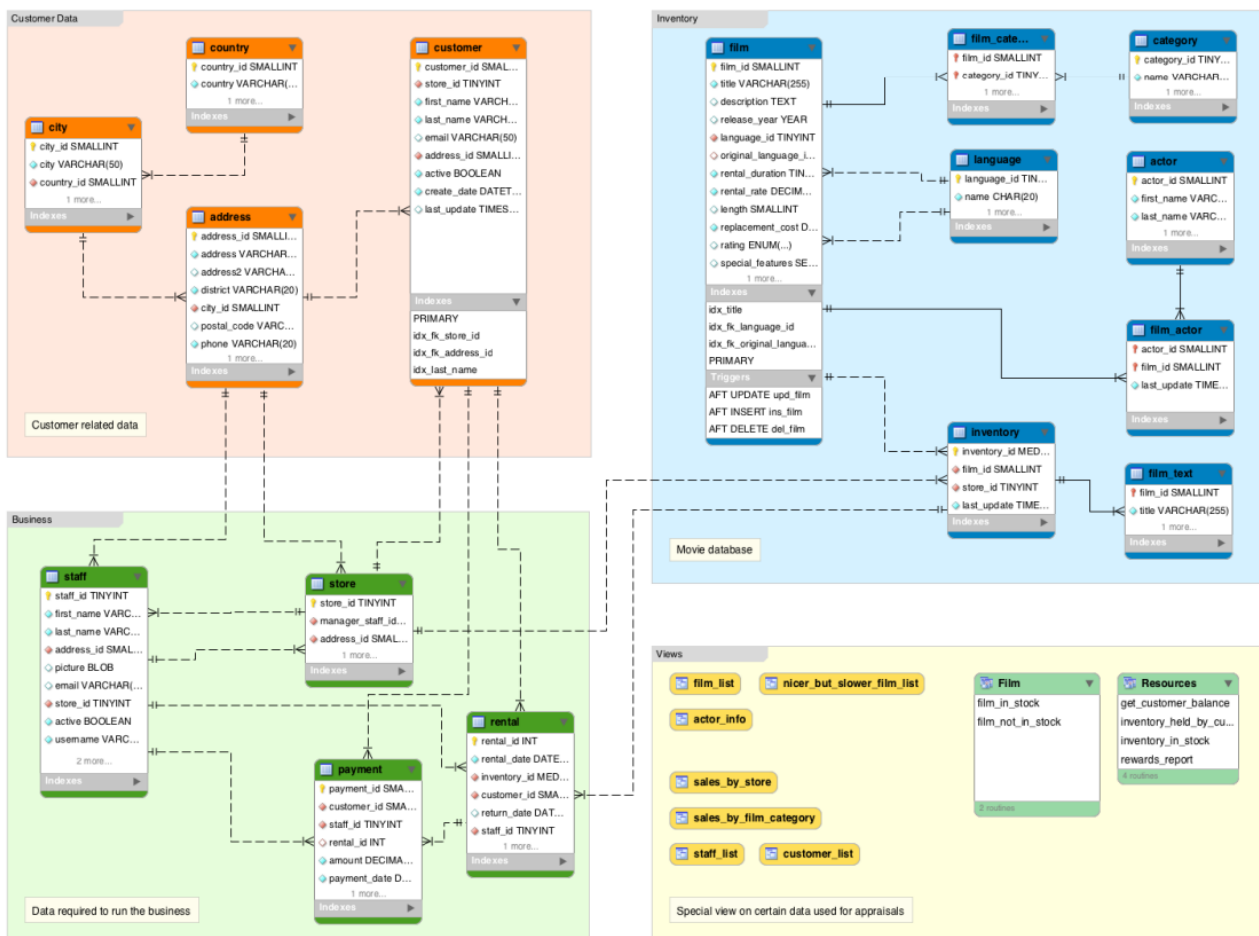
```

1 -- Consultas sobre o esquema empresa
2 -- 22. Nome dos empregados e nome do grupo
3 SELECT e.nome AS empregado, g.nome AS grupo
4 FROM Empregado e
5 JOIN Empresa em ON e.id_empresa = em.id
6 JOIN Grupo g ON em.id_grupo = g.id;
7
8 -- 23. Dependentes dos funcionários do grupo "ABC"
9 SELECT d.nome AS dependente
10 FROM Dependente d
11 JOIN Empregado e ON d.id_empregado = e.id
12 JOIN Empresa em ON e.id_empresa = em.id
13 JOIN Grupo g ON em.id_grupo = g.id
14 WHERE g.nome = 'ABC';
15
16 -- 24. Grupo, empresa, empregado e total de dependentes
17 SELECT g.nome AS grupo, em.nome AS empresa, e.nome AS empregado, COUNT(d.id) AS total_dependentes
18 FROM Empregado e
19 LEFT JOIN Dependente d ON e.id = d.id_empregado
20 JOIN Empresa em ON e.id_empresa = em.id
21 JOIN Grupo g ON em.id_grupo = g.id
22 GROUP BY g.nome, em.nome, e.nome;

```

3 Esquema Sakila

Figura 5: Esquema Sakila



3.1 Questões 25 a 32

```

1  -- 25. Liste o nome de todos idiomas em ordem alfabética
2  SELECT name FROM language ORDER BY name;
3
4  -- 26. Nome e endereço de clientes com district vazio
5  SELECT first_name, last_name, address
6  FROM customer cu
7  JOIN address a ON cu.address_id = a.address_id
8  WHERE district = '';
9
10 -- 27. Nome de todas as categorias de filmes e total de filmes por categoria
11 SELECT c.name AS categoria, COUNT(f.film_id) AS total_filmes
12 FROM category c
13 JOIN film_category fc ON c.category_id = fc.category_id
14 JOIN film f ON f.film_id = fc.film_id
15 GROUP BY c.name
16 ORDER BY total_filmes DESC;
17
18 -- 28. Título do filme e nome da categoria
19 SELECT f.title, c.name AS categoria
20 FROM film f
21 JOIN film_category fc ON f.film_id = fc.film_id

```

```

22 JOIN category c ON c.category_id = fc.category_id
23 ORDER BY c.name;
24
25 -- 29. Clientes que ainda não alugaram filme
26 SELECT cu.first_name, cu.last_name
27 FROM customer cu
28 LEFT JOIN rental r ON cu.customer_id = r.customer_id
29 WHERE r.rental_id IS NULL;
30
31 -- 30. Total de cópias por filme na loja 1
32 SELECT f.title, COUNT(i.inventory_id) AS total_copias
33 FROM film f
34 JOIN inventory i ON f.film_id = i.film_id
35 WHERE i.store_id = 1
36 GROUP BY f.title;
37
38 -- 31. Dados das locações da loja 1 que ainda não foram devolvidas
39 SELECT r.*
40 FROM rental r
41 JOIN inventory i ON r.inventory_id = i.inventory_id
42 WHERE i.store_id = 1 AND r.return_date IS NULL;
43
44 -- 32. Cliente 168 devolveu o DVD 4416
45 UPDATE rental
46 SET return_date = NOW()
47 WHERE customer_id = 168 AND inventory_id = 4416;

```