

Relacionamentos e Cardinalidades

TB_ALUNO ↔ TB_CURSO

Cada aluno está associado a apenas um curso, enquanto um curso pode ter vários alunos matriculados. → **Cardinalidade:** (1,N) ↔ (1,1)

TB_CURSO ↔ TB_DISCIPLINA

Um curso é composto por múltiplas disciplinas, e cada disciplina está vinculada a um único curso. → **Cardinalidade:** (1,N) ↔ (1,1)

TB_ALUNO ↔ TB_DISCIPLINA

Um aluno pode estar inscrito em várias disciplinas, e uma disciplina pode ser frequentada por diversos alunos. → **Cardinalidade:** (N:N)

→ É necessário criar a tabela auxiliar **TB_ALUNO_DISCIPLINA** para representar essa relação.

TB_PROFESSOR ↔ TB_DISCIPLINA

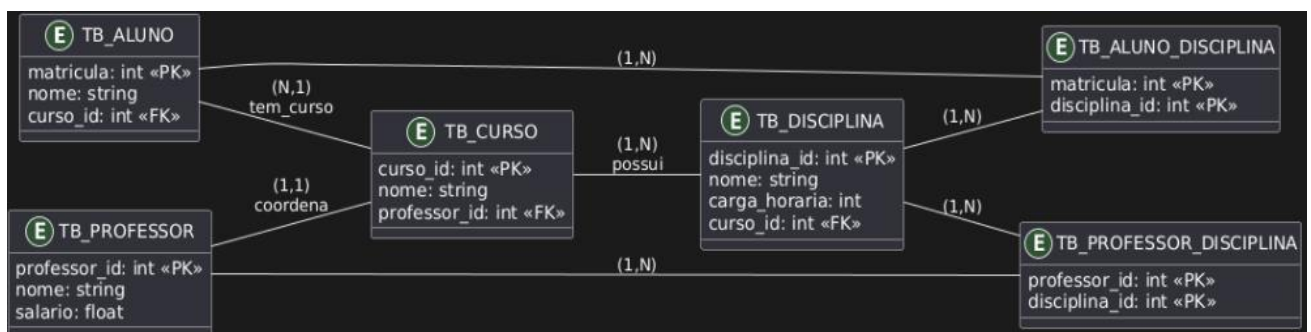
Um professor pode ensinar diversas disciplinas, e uma disciplina pode ser lecionada por diferentes professores. → **Cardinalidade:** (N:N)

→ É preciso criar a tabela de ligação **TB_PROFESSOR_DISCIPLINA** para esse relacionamento.

TB_PROFESSOR ↔ TB_CURSO

Cada curso é coordenado por um único professor, e um professor pode coordenar apenas um curso. → **Cardinalidade:** (1,1) ↔ (1,1)

DIAGRAMA:



Código UML:

@startuml

Left to right direction

entity TB_ALUNO {

```

    matricula: int <<PK>>
    nome: string
    curso_id: int <<FK>>
}
entity TB_CURSO {
    curso_id: int <<PK>>
    nome: string
    professor_id: int <<FK>>
}
entity TB_DISCIPLINA {
    disciplina_id: int <<PK>>
    nome: string
    carga_horaria: int
    curso_id: int <<FK>>
}
entity TB_PROFESSOR {
    professor_id: int <<PK>>
    nome: string
    salario: float
}
entity TB_ALUNO_DISCIPLINA {
    matricula: int <<PK>>
    disciplina: int <<PK>>
}
entity TB_PROFESSOR_DISCIPLINA {
    professor id: int <<PK>>
    disciplina_id: <<PK>>
}
TB_ALUNO -- TB_CURSO : (N, 1) \ntem curso
TB_CURSO -- TB_DISCIPLINA : (1, N) \npossui
TB_ALUNO -- TB_ALUNO_DISCIPLINA : (1, N) \
TB_DISCIPLINA -- TB_ALUNO_DISCIPLINA : (1, N) \
TB_PROFESSOR -- TB_PROFESSOR_DISCIPLINA : (1, N) \
TB_DISCIPLINA -- TB_PROFESSOR_DISCIPLINA : (1, N) \
TB_PROFESSOR -- TB_CURSO : (1, 1) \ncoordena

```

Código SQL:

```

1 CREATE TABLE TB_PROFESSOR (
2     professor_id INTEGER PRIMARY KEY,
3     nome TEXT NOT NULL,
4     salario REAL NOT NULL
5 );
6 CREATE TABLE TB_CURSO (
7     curso_id INTEGER PRIMARY KEY,
8     nome TEXT NOT NULL,
9     professor_id INTEGER UNIQUE,
10    FOREIGN KEY(professor_id) REFERENCES TB_PROFESSOR(professor_id)
11 );
12 CREATE TABLE TB_ALUNO (
13     matricula INTEGER PRIMARY KEY,
14     nome TEXT NOT NULL,
15     curso_id INTEGER NOT NULL,
16    FOREIGN KEY(curso_id) REFERENCES TB_CURSO(curso_id)
17 );
18 CREATE TABLE TB_DISCIPLINA (
19     disciplina_id INTEGER PRIMARY KEY,
20     nome TEXT NOT NULL,
21     carga_horaria INTEGER NOT NULL,
22     curso_id INTEGER NOT NULL,
23    FOREIGN KEY(curso_id) REFERENCES TB_CURSO(curso_id)
24 );
25 CREATE TABLE TB_ALUNO_DISCIPLINA (
26     matricula INTEGER NOT NULL,
27     disciplina_id INTEGER NOT NULL,
28     PRIMARY KEY (matricula, disciplina_id),
29     FOREIGN KEY(matricula) REFERENCES TB_ALUNO(matricula),
30     FOREIGN KEY(disciplina_id) REFERENCES TB_DISCIPLINA(disciplina_id)
31 );
32 CREATE TABLE TB_PROFESSOR_DISCIPLINA (
33     professor_id INTEGER NOT NULL,
34     disciplina_id INTEGER NOT NULL,
35     PRIMARY KEY (professor_id, disciplina_id),
36     FOREIGN KEY(professor_id) REFERENCES TB_PROFESSOR(professor_id),
37     FOREIGN KEY(disciplina_id) REFERENCES TB_DISCIPLINA(disciplina_id)
38 );

```

Saída:

```

Execução finalizada sem erros.
Resultado: consulta executada com sucesso. Levou 0ms
Na linha 32:
CREATE TABLE TB_PROFESSOR_DISCIPLINA (
    professor_id INTEGER NOT NULL,
    disciplina_id INTEGER NOT NULL,
    PRIMARY KEY (professor_id, disciplina_id),
    FOREIGN KEY(professor_id) REFERENCES TB_PROFESSOR(professor_id),
    FOREIGN KEY(disciplina_id) REFERENCES TB_DISCIPLINA(disciplina_id)
);

```

Insert:

```

INSERT INTO TB_PROFESSOR (professor_id, nome, salario) VALUES (1, 'Ricardo Taveira', 4500.00);

INSERT INTO TB_CURSO (curso_id, nome, professor_id) VALUES (10, 'informática', 1);

INSERT INTO TB_ALUNO (matricula, nome, curso_id) VALUES (100, 'Iuri da Costa Aragão', 10);

INSERT INTO TB_DISCIPLINA (disciplina_id, nome, carga_horaria, curso_id) VALUES (200, 'Banco de Dados', 60, 10);

INSERT INTO TB_ALUNO_DISCIPLINA (matricula, disciplina_id) VALUES (100, 200);

INSERT INTO TB_PROFESSOR_DISCIPLINA (professor_id, disciplina_id) VALUES (1, 200);

```

Saída:

Execução finalizada sem erros.

Resultado: consulta executada com sucesso. Levou 0ms, 1 linhas afetadas

Na linha 11:

```
INSERT INTO TB_PROFESSOR_DISCIPLINA (professor_id, disciplina_id) VALUES (1, 200);
```

Select:

```

1  SELECT A.matricula, A.nome AS aluno_nome, C.nome AS curso_nome
2  FROM TB_ALUNO A
3  JOIN TB_CURSO C ON A.curso_id = C.curso_id;
4  SELECT D.nome AS disciplina
5  FROM TB_DISCIPLINA D
6  JOIN TB_ALUNO_DISCIPLINA AD ON D.disciplina_id = AD.disciplina_id
7  WHERE AD.matricula = 100;
8  SELECT P.nome AS coordenador, C.nome AS curso
9  FROM TB_PROFESSOR P
10 JOIN TB_CURSO C ON P.professor_id = C.professor_id;
11 SELECT P.nome AS professor, D.nome AS disciplina
12 FROM TB_PROFESSOR P
13 JOIN TB_PROFESSOR_DISCIPLINA PD ON P.professor_id = PD.professor_id
14 JOIN TB_DISCIPLINA D ON PD.disciplina_id = D.disciplina_id
15 WHERE D.nome = 'Banco de Dados';

```

Saída:

	professor	disciplina
1	Ricardo Taveira	Banco de Dados

Execução finalizada sem erros.

Resultado: 1 linhas retornadas em 5 ms

Na linha 11:

```

SELECT P.nome AS professor, D.nome AS disciplina
FROM TB_PROFESSOR P
JOIN TB_PROFESSOR_DISCIPLINA PD ON P.professor_id = PD.professor_id
JOIN TB_DISCIPLINA D ON PD.disciplina_id = D.disciplina_id
WHERE D.nome = 'Banco de Dados';

```

Update:

```

1  UPDATE TB_PROFESSOR
2  SET salario = 5000.00
3  WHERE professor_id = 1;
4
5  UPDATE TB_CURSO
6  SET nome = 'Engenharia de Computação'
7  WHERE curso_id = 10;
8
9  UPDATE TB_ALUNO
10 SET nome = 'Nome Novo'
11 WHERE matricula = 100;
12
13 UPDATE TB_DISCIPLINA
14 SET carga_horaria = 80
15 WHERE disciplina_id = 200;

```

Saída:

Execução finalizada sem erros.

Resultado: consulta executada com sucesso. Levou 0ms, 1 linhas afetadas

Na linha 13:

```

UPDATE TB_DISCIPLINA
SET carga_horaria = 80
WHERE disciplina_id = 200;

```

Delete:

```

1  DELETE FROM TB_ALUNO_DISCIPLINA
2  WHERE matricula = 100 AND disciplina_id = 200;
3  DELETE FROM TB_PROFESSOR_DISCIPLINA
4  WHERE professor_id = 1 AND disciplina_id = 200;
5  DELETE FROM TB_DISCIPLINA
6  WHERE disciplina_id = 200;
7  DELETE FROM TB_ALUNO
8  WHERE matricula = 100;
9  DELETE FROM TB_CURSO
10 WHERE curso_id = 10;
11 DELETE FROM TB_PROFESSOR
12 WHERE professor_id = 1;

```

Saída:

Execução finalizada sem erros.

Resultado: consulta executada com sucesso. Levou 0ms, 1 linhas afetadas

Na linha 11:

```

DELETE FROM TB_PROFESSOR
WHERE professor_id = 1;

```