



Avaliação 03 - Banco de Dados

André de Almeida Dantas - 20232011060131

Banco de dados SCA_251:

Etapa 1 - Identificando Relacionamentos e Cardinalidades

TB_ALUNO ↔ TB_CURSO: Muitos alunos pertencem a um curso, e um curso pode ter muitos alunos.

→ Cardinalidade: (1,N) ↔ (1,1)

TB_CURSO ↔ TB_DISCIPLINA: Um curso pode ter várias disciplinas e uma disciplina pertence a um curso.

→ Cardinalidade: (1,N) ↔ (1,1)

TB_ALUNO ↔ TB_DISCIPLINA: Um aluno pode cursar várias disciplinas e uma disciplina pode ser cursada por vários alunos.

→ Cardinalidade: (N:N)

→ Necessário criar a tabela **TB_ALUNO_DISCIPLINA**

TB_PROFESSOR ↔ TB_DISCIPLINA: Um professor pode ministrar várias disciplinas e uma disciplina pode ser ministrada por vários professores.

→ Cardinalidade: (N:N)

→ Necessário criar a tabela **TB_PROFESSOR_DISCIPLINA**

TB_PROFESSOR ↔ TB_CURSO: Um professor coordena apenas um curso, mas um curso é coordenado por um único professor.

→ Cardinalidade: (1,1) ↔ (1,N)

Etapa 2 - Diagrama Lógico do Banco de Dados

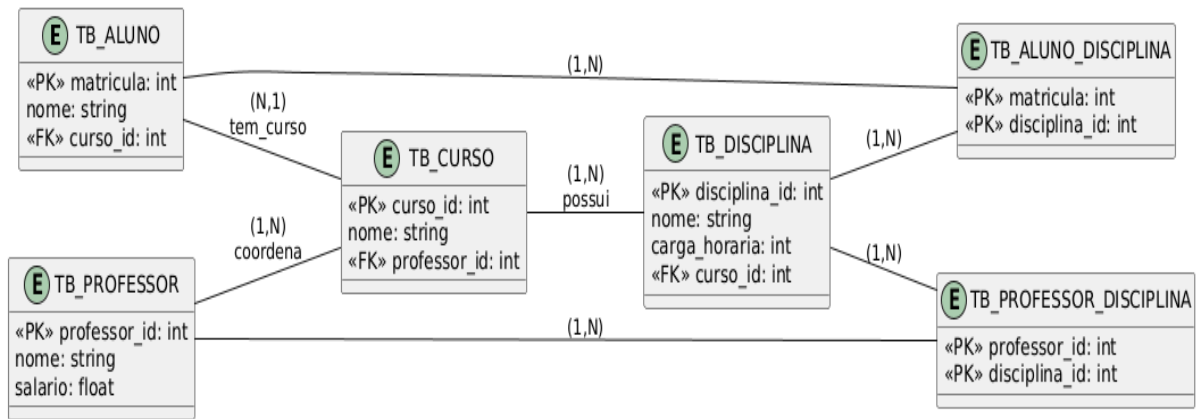


Diagrama:

Código UML:

```
@startuml
left to right direction
' Entidade Aluno
entity TB_ALUNO {
    <<PK>> matricula: int
    nome: string
    <<FK>> curso_id: int
}

' Entidade Curso
entity TB_CURSO {
    <<PK>> curso_id: int
    nome: string
    <<FK>> professor_id: int
}

' Entidade Disciplina
entity TB_DISCIPLINA {
    <<PK>> disciplina_id: int
    nome: string
    carga_horaria: int
    <<FK>> curso_id: int
}

' Entidade Professor
entity TB_PROFESSOR {
    <<PK>> professor_id: int
    nome: string
    salario: float
}

' Tabela intermediária Aluno x Disciplina
entity TB_ALUNO_DISCIPLINA {
    <<PK>> matricula: int
    <<PK>> disciplina_id: int
}

' Tabela intermediária Professor x Disciplina
entity TB_PROFESSOR_DISCIPLINA {
    <<PK>> professor_id: int
    <<PK>> disciplina_id: int
}

' Relacionamentos com cardinalidades escritas
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,N)\ncoordena
@enduml
```

Etapa 3 - Códigos SQL para a criação das tabelas e do BD

Como estava trabalhando no DB BROWSER, não precisei usar o:

```
CREATE DATABASE SCA_251;  
USE SCA_251;
```

Então, aqui estão os códigos para a criação das tabelas:

```
1 CREATE TABLE TB_PROFESSOR (  
2     professor_id INT PRIMARY KEY,  
3     nome VARCHAR(100),  
4     salario FLOAT  
5 );  
6  
7 CREATE TABLE TB_CURSO (  
8     curso_id INT PRIMARY KEY,  
9     nome VARCHAR(100),  
10    professor_id INT,  
11    FOREIGN KEY (professor_id) REFERENCES TB_PROFESSOR(professor_id)  
12 );  
13  
14 CREATE TABLE TB_ALUNO (  
15     matricula INT PRIMARY KEY,  
16     nome VARCHAR(100),  
17     curso_id,  
18     FOREIGN KEY (curso_id) REFERENCES TB_CURSO(curso_id)  
19 );  
20  
21 CREATE TABLE TB_DISCIPLINA (  
22     disciplina_id INT PRIMARY KEY,  
23     nome VARCHAR(100),  
24     carga_horaria INT,  
25     curso_id INT,  
26     FOREIGN KEY (curso_id) REFERENCES TB_CURSO(curso_id)  
27 );  
28  
29 CREATE TABLE TB_ALUNO_DISCIPLINA (  
30     matricula INT,  
31     disciplina_id INT,  
32     PRIMARY KEY(matricula, disciplina_id),  
33     FOREIGN KEY (matricula) REFERENCES TB_ALUNO(matricula),  
34     FOREIGN KEY (disciplina_id) REFERENCES TB_DISCIPLINA(disciplina_id)  
35 );  
36  
37 CREATE TABLE TB_PROFESSOR_DISCIPLINA (  
38     professor_id INT,  
39     disciplina_id INT,  
40     PRIMARY KEY (professor_id, disciplina_id),  
41     FOREIGN KEY (professor_id) REFERENCES TB_PROFESSOR(professor_id),  
42     FOREIGN KEY (disciplina_id) REFERENCES TB_DISCIPLINA(disciplina_id)  
43 );
```

saída:

```
Execução finalizada sem erros.  
Resultado: consulta executada com sucesso. Levou 0ms  
Na linha 37:  
CREATE TABLE TB_PROFESSOR_DISCIPLINA (  
professor_id INT,  
disciplina_id INT,  
PRIMARY KEY (professor_id, disciplina_id),  
FOREIGN KEY (professor_id) REFERENCES TB_PROFESSOR(professor_id),  
FOREIGN KEY (disciplina_id) REFERENCES TB_DISCIPLINA(disciplina_id)  
)
```

Etapa 4 - EXEMPLOS de CRUD(Insert, Select, Update e Delete)

Insert:

```
1  INSERT INTO TB_PROFESSOR VALUES (1, "Ricardo D. Taveira", 23981.27);
2  INSERT INTO TB_CURSO VALUES (060, "Informática", 1);
3  INSERT INTO TB_ALUNO VALUES (20232011060131, "André de Almeida Dantas", 060);
4  INSERT INTO TB_DISCIPLINA VALUES (251, "Banco de Dados", 80, 060);
5
6  INSERT INTO TB_ALUNO_DISCIPLINA VALUES (20232011060131, 251);
7  INSERT INTO TB_PROFESSOR_DISCIPLINA VALUES (1, 251);
8  |
```

Execução finalizada sem erros.

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

Na linha 7:

```
INSERT INTO TB_PROFESSOR_DISCIPLINA VALUES (1, 251);
```

Read:

```
1  SELECT nome FROM TB_ALUNO WHERE curso_id = 060;
2
3  SELECT a.nome AS aluno, d.nome AS disciplina
4  FROM TB_ALUNO a
5  JOIN TB_ALUNO_DISCIPLINA ad ON a.matricula = ad.matricula
6  JOIN TB_DISCIPLINA d ON d.disciplina_id = ad.disciplina_id
7  |
```

	aluno	disciplina
1	André de Almeida Dantas	Banco de Dados

Execução finalizada sem erros.

Resultado: 1 linhas retornadas em 5 ms

Na linha 3:

```
SELECT a.nome AS aluno, d.nome AS disciplina
FROM TB_ALUNO a
JOIN TB_ALUNO_DISCIPLINA ad ON a.matricula = ad.matricula
JOIN TB_DISCIPLINA d ON d.disciplina_id = ad.disciplina_id
```

Update:

```
1 UPDATE TB_PROFESSOR SET salario = 24000.00 WHERE professor_id = 1;  
2 UPDATE TB_PROFESSOR SET nome = "Ricardo Duarte Taveira" WHERE professor_id = 1  
3
```

Execução finalizada sem erros.

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

Na linha 2:

```
UPDATE TB_PROFESSOR SET nome = "Ricardo Duarte Taveira" WHERE professor_id = 1
```

Delete:

```
1 DELETE FROM TB_ALUNO_DISCIPLINA WHERE matricula = 20232011060131 AND disciplina_id = 251;  
2  
3 DELETE FROM TB_ALUNO WHERE matricula = 20232011060131  
4
```

Execução finalizada sem erros.

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

Na linha 3:

```
DELETE FROM TB_ALUNO WHERE matricula = 20232011060131
```