

# Atividade de Banco de Dados - Avaliação 3

Nome: Miguel Angelo da Silva Costa

Crie um banco de dados SCA\_251.bd.

Nele defina as Tabela TB\_ALUNO, TB\_PROFESSOR, TB\_CURSO e TB\_DISCIPLINA.

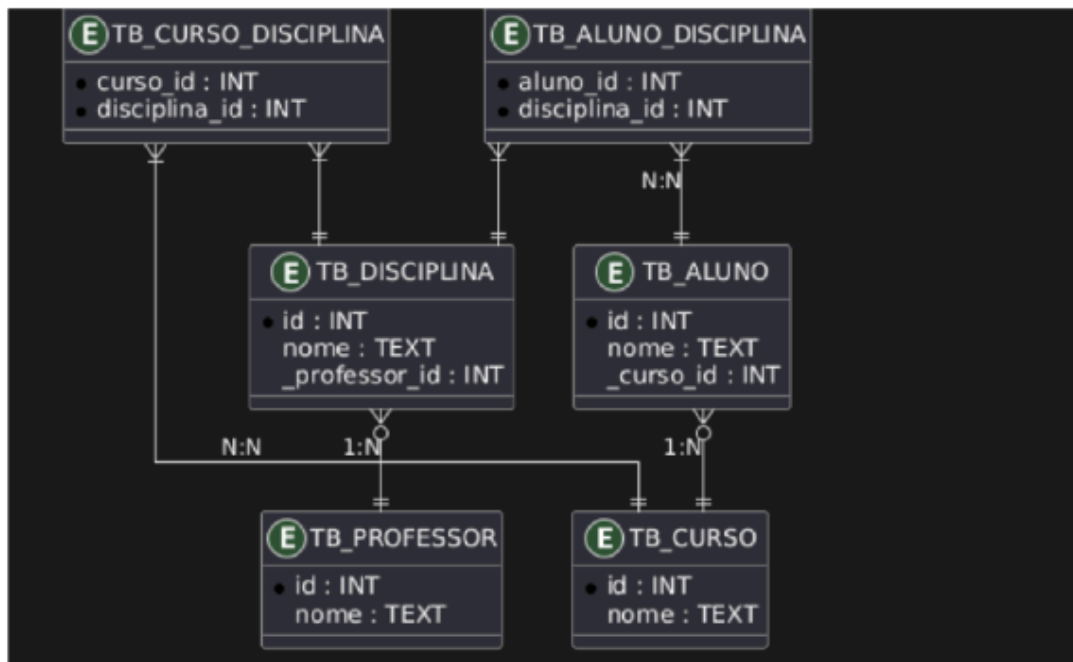
Pede-se:

1) Identifique os possíveis relacionamentos com as suas cardinalidades envolvendo as tabelas.

## Relacionamentos:

- **TB\_ALUNO** está matriculado em **TB\_CURSO**:
  - Cardinalidade: Muitos para Um (N:1).
  - Cada aluno pertence a um curso; um curso pode ter vários alunos.
- **TB\_DISCIPLINA** pertence a **TB\_CURSO**:
  - Cardinalidade: Muitos para Um (N:1).
  - Cada disciplina pertence a um curso; um curso pode ter várias disciplinas.
- **TB\_PROFESSOR** ministra **TB\_DISCIPLINA**:
  - Cardinalidade: Muitos para Muitos (N:N).
  - Um professor pode ministrar várias disciplinas e uma disciplina pode ser ministrada por vários professores.
- **TB\_ALUNO** cursa **TB\_DISCIPLINA**:
  - Cardinalidade: Muitos para Muitos (N:N).
  - Um aluno pode cursar várias disciplinas, e uma disciplina pode ser cursada por vários alunos.

2) Crie um Diagrama de um Modelo Lógico no qual são definidas chaves primárias (id) e chaves estrangeiras(tabelaid) seus relacionamentos e suas cardinalidades. Onde existir relacionamentos N:N criar uma tabela TB\_E1\_TB\_E2 com chave primária concatenada envolvendo as chaves primárias de cada tabela;



3) Escreva o comando SQL cria esse banco de dados.

Nome	Tipo	Esquema
<b>TB_ALUNO</b>		CREATE TABLE TB_ALUNO ( ID INT PRIMARY KEY, nome TEXT NOT NULL, cursoID INT, FOREIGN KEY (cursoID) REFERENCES TB_CURSO(ID) )
ID	INT	"ID" INT
nome	TEXT	"nome" TEXT NOT NULL
cursoID	INT	"cursoID" INT
<b>TB_ALUNO_DISCIPLINA</b>		CREATE TABLE TB_ALUNO_DISCIPLINA ( alunoID INT, disciplinaID INT, PRIMARY KEY (alunoID, disciplinaID), FOREIGN KEY (alunoID) REFERENCES TB_ALUNO(ID), FOREIGN KEY (disciplinaID) REFERENCES TB_DISCIPLINA(ID) )
alunoID	INT	"alunoID" INT
disciplinaID	INT	"disciplinaID" INT
<b>TB_CURSO</b>		CREATE TABLE TB_CURSO( ID INT PRIMARY KEY, nome TEXT NOT NULL )
ID	INT	"ID" INT
nome	TEXT	"nome" TEXT NOT NULL
<b>TB_CURSO_DISPLINA</b>		CREATE TABLE TB_CURSO DISPLINA( cursoID INT, disciplinaID INT, PRIMARY KEY (cursoID, disciplinaID), FOREIGN KEY (cursoID) REFERENCES TB_CURSO(ID), FOREIGN KEY (disciplinaID) REFERENCES TB_DISCIPLINA(ID) )
cursoID	INT	"cursoID" INT
disciplinaID	INT	"disciplinaID" INT
<b>TB_DISCIPLINA</b>		CREATE TABLE TB DISCIPLINA ( ID INT PRIMARY KEY, nome TEXT NOT NULL, professorID INT, FOREIGN KEY (professorID) REFERENCES TB_PROFESSOR(ID) )
ID	INT	"ID" INT
nome	TEXT	"nome" TEXT NOT NULL
professorID	INT	"professorID" INT
<b>TB_PROFESSOR</b>		CREATE TABLE TB_PROFESSOR ( ID INT PRIMARY KEY, nome TEXT NOT NULL )
ID	INT	"ID" INT
nome	TEXT	"nome" TEXT NOT NULL

4) Exemplifique o uso de comando SQL que executam operações CRUD(Insert, Select, Update e Delete) envolvendo todas as tabelas.

 **INSERT**

sql

CopiarEditar

```
-- Inserir cursos
INSERT INTO TB_CURSO (nome, cargaHoraria) VALUES ('Engenharia',
3600);
INSERT INTO TB_CURSO (nome, cargaHoraria) VALUES ('Administração',
3000);

-- Inserir alunos
INSERT INTO TB_ALUNO (nome, email, cursoId) VALUES ('Ana Silva',
'ana.silva@email.com', 1);
INSERT INTO TB_ALUNO (nome, email, cursoId) VALUES ('Carlos Souza',
'carlos.souza@email.com', 2);

-- Inserir professores
INSERT INTO TB_PROFESSOR (nome, email) VALUES ('Dr. João',
'joao.prof@email.com');
INSERT INTO TB_PROFESSOR (nome, email) VALUES ('Profa. Maria',
'maria.prof@email.com');

-- Inserir disciplinas
INSERT INTO TB_DISCIPLINA (nome, cargaHoraria, cursoId) VALUES
('Matemática', 60, 1);
INSERT INTO TB_DISCIPLINA (nome, cargaHoraria, cursoId) VALUES
('Administração Financeira', 80, 2);

-- Associar aluno à disciplina
INSERT INTO TB_ALUNO_TB_DISCIPLINA (alunoId, disciplinaId) VALUES
(1, 1);
INSERT INTO TB_ALUNO_TB_DISCIPLINA (alunoId, disciplinaId) VALUES
(2, 2);

-- Associar professor à disciplina
INSERT INTO TB_PROFESSOR_TB_DISCIPLINA (professorId, disciplinaId)
VALUES (1, 1);
INSERT INTO TB_PROFESSOR_TB_DISCIPLINA (professorId, disciplinaId)
VALUES (2, 2);
```

---

 **SELECT**

sql

CopiarEditar

```
-- Selecionar todos os alunos com seus cursos
SELECT A.nome AS Aluno, A.email, C.nome AS Curso
FROM TB_ALUNO A
JOIN TB_CURSO C ON A.cursoId = C.idCurso;

-- Selecionar disciplinas e professores
SELECT D.nome AS Disciplina, P.nome AS Professor
FROM TB_DISCIPLINA D
JOIN TB_PROFESSOR_TB_DISCIPLINA PD ON D.idDisciplina =
PD.disciplinaId
JOIN TB_PROFESSOR P ON PD.professorId = P.idProfessor;
```

---

## UPDATE

sql

CopiarEditar

```
-- Atualizar email de um aluno
UPDATE TB_ALUNO
SET email = 'ana.nova@email.com'
WHERE idAluno = 1;

-- Atualizar carga horária de um curso
UPDATE TB_CURSO
SET cargaHoraria = 4000
WHERE idCurso = 1;
```

---

## DELETE

sql

CopiarEditar

```
-- Remover um aluno de uma disciplina
DELETE FROM TB_ALUNO_TB_DISCIPLINA
WHERE alunoId = 1 AND disciplinaId = 1;

-- Remover um professor
DELETE FROM TB_PROFESSOR
WHERE idProfessor = 2;
```

```
SQL 1* SQL 2* SQL 3* SQL 4* SQL 5* SQL 6* SQL 7* SQL 8* SQL 9* SQL 10* SQL 11*
1 INSERT INTO TB_CURSO (ID, nome) VALUES (1, 'Informática');
2 INSERT INTO TB_ALUNO (ID, nome, cursoID) VALUES (1, 'fabio', 1), (2, 'andre', 2);
3
```

```
1 INSERT INTO TB_PROFESSOR (ID, nome) VALUES (1, 'Taveira'), (2, 'Alisso bunitokkkk');
2 INSERT INTO TB_DISCIPLINA (ID, nome, professorID) VALUES (1, 'Banco de Dados', 1), (2, 'MFDS', 2);
3 INSERT INTO TB_ALUNO_DISCIPLINA (alunoID, disciplinaID) VALUES (1,1), (2,2);
4 INSERT INTO TB_CURSO_DISCIPLINA (cursoID, disciplinaID) VALUES (1,1), (2,2);

Execução finalizada sem erros.
Resultado: consulta executada com sucesso. Levou 0ms, 2 linhas afetadas
Na linha 4:
INSERT INTO TB_CURSO_DISCIPLINA (cursoID, disciplinaID) VALUES (1,1), (2,2);
```

```
SQL 1* SQL 2* SQL 3* SQL 4* SQL 5* SQL 6* SQL 7*
1 SELECT A.nome AS Aluno, C.nome AS Curso
2 FROM TB_ALUNO A
3 JOIN TB_CURSO C ON A.cursoID = C.ID;
4
5 SELECT D.nome AS Disciplina, P.nome AS Professor
6 FROM TB_DISCIPLINA D
7 JOIN TB_PROFESSOR P ON D.professorID = P.ID;
8
9 SELECT C.nome AS Curso, D.nome AS Disciplina
10 FROM TB_CURSO_DISCIPLINA CD
11 JOIN TB_CURSO C ON CD.cursoID = C.ID
12 JOIN TB_DISCIPLINA D ON CD.disciplinaID = D.ID;
13
14 SELECT A.nome AS Aluno, D.nome AS Disciplina
15 FROM TB_ALUNO_DISCIPLINA AD
16 JOIN TB_ALUNO A ON AD.alunoID = A.ID
17 JOIN TB_DISCIPLINA D ON AD.disciplinaID = D.ID;
18
```

Aluno	Disciplina
-------	------------

por: Dylan Mesquita de Melo

```
SQL 1* SQL 2* SQL 3* SQL 4* SQL 5
1
2 UPDATE TB_ALUNO
3 SET nome = 'Dylan'
4 WHERE ID = 1;
5
6 UPDATE TB_DISCIPLINA
7 SET professorID = 2
8 WHERE ID = 1;
9
10 UPDATE TB_CURSO
11 SET nome = 'ciencia da computacao'
12 WHERE ID = 1;
13
```

```
SQL 1* SQL 2* SQL 3* SQL 4* SQL 5* SQL 6*
1
2 DELETE FROM TB_ALUNO_DISCIPLINA
3 WHERE alunoID = 1 AND disciplinaID= 1;
4
5 DELETE FROM TB_CURSO_DISPLINA
6 WHERE cursoID = 2 AND disciplinaID = 2;
7
8
9 DELETE FROM TB_ALUNO WHERE ID = 1;
10
11
```