

# BANCO DE DADOS

## ATIVIDADE-03

por: Dylan Mesquita de Melo

Crie um banco de dados SCA\_242.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.

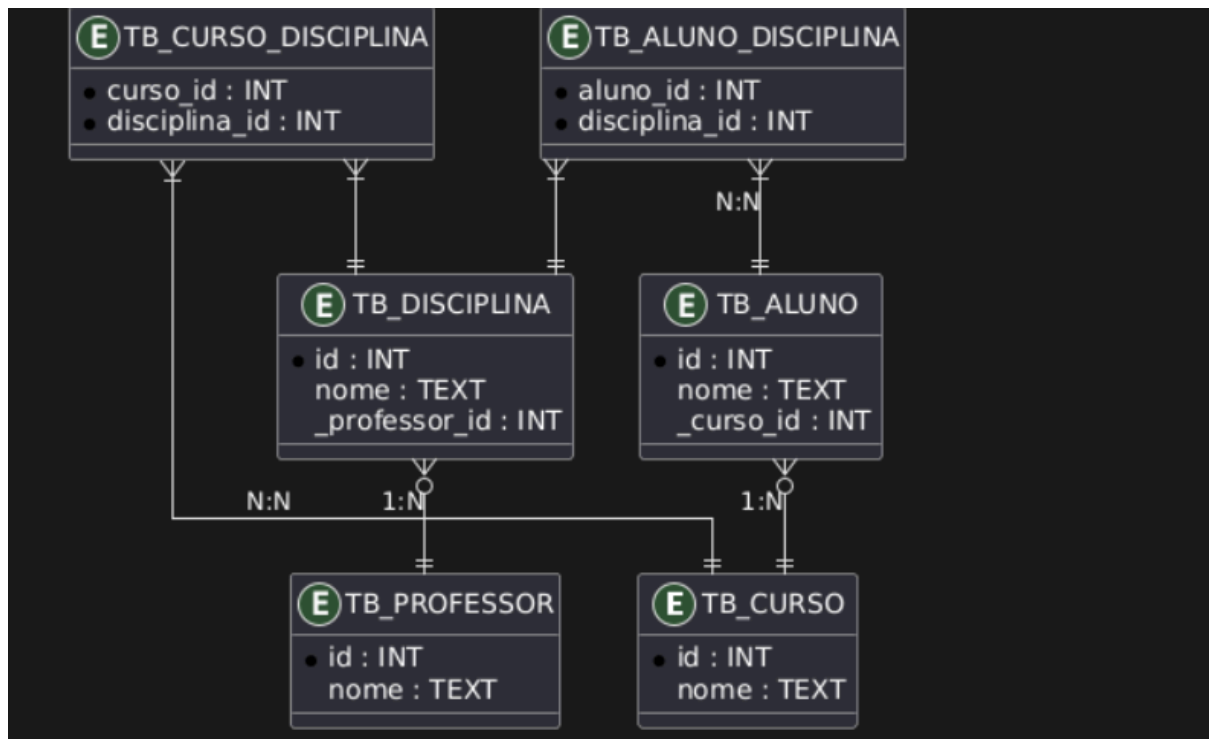
TB\_ALUNO : TB\_PROFESSOR (1:1)

TB\_ALUNO : TB\_DISCIPLINA / TB\_CURSO (1:n)

TB\_PROFESSOR : TB\_DISCIPLINA / TB\_CURSO(1:N)

TB\_DISCIPLINA : TB\_CURSO (n:n)

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;



# BANCO DE DADOS

## ATIVIDADE-03

*por: Dylan Mesquita de Melo*

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

### Tabelas (6)

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_DISCIPLINA</b>		CREATE TABLE TB_CURSO_DISCIPLINA( 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.

# BANCO DE DADOS

## ATIVIDADE-03

por: Dylan Mesquita de Melo

### Labelas (6)

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

```
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, 'info');
2 INSERT INTO TB_ALUNO (ID, nome, cursoID) VALUES (1, 'fabio', 1), (2, 'andre', 2);
3
```

Execução finalizada com erros.  
Resultado: UNIQUE constraint failed: TB\_CURSO.ID  
Na linha 1:  
INSERT INTO TB\_CURSO (ID, nome) VALUES (1, 'informática'), (2, 'info');

# BANCO DE DADOS

## ATIVIDADE-03

por: Dylan Mesquita de Melo

```
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* x SQL 2* x SQL 3* x SQL 4* x SQL 5* x SQL 6* x 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
```

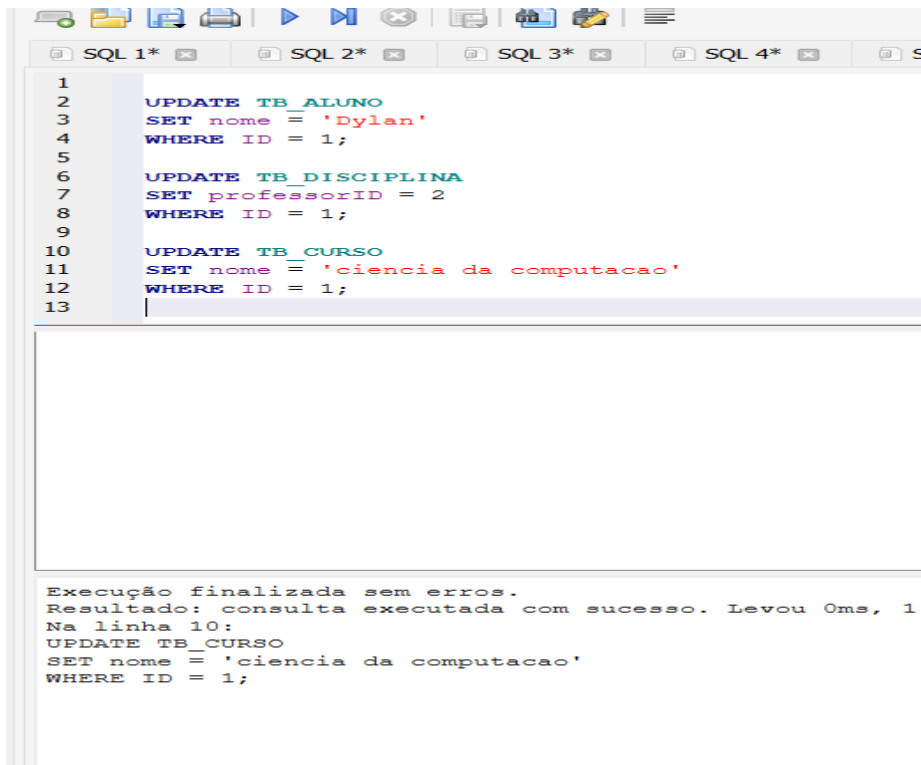
	Aluno	Disciplina
1	fabio	Banco de Dados
2	andre	MFDS

Execução finalizada sem erros.  
Resultado: 2 linhas retornadas em 7 ms  
Na linha 14:  
SELECT A.nome AS Aluno, D.nome AS Disciplina  
FROM TB\_ALUNO\_DISCIPLINA AD  
JOIN TB\_ALUNO A ON AD.alunoID = A.ID  
JOIN TB\_DISCIPLINA D ON AD.disciplinaID = D.ID;

# BANCO DE DADOS

## ATIVIDADE-03

por: Dylan Mesquita de Melo

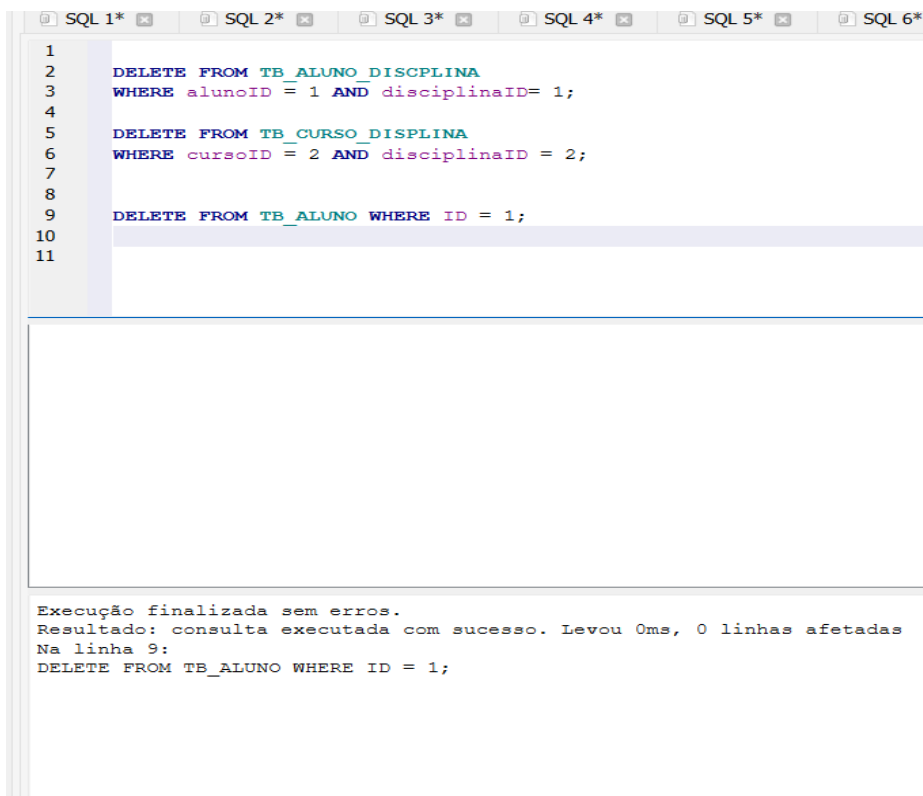


The screenshot shows a SQL IDE with a toolbar at the top and a tab bar with tabs labeled SQL 1\*, SQL 2\*, SQL 3\*, SQL 4\*, and SQL 5\*. The main editor area contains the following SQL code:

```
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
```

Below the code editor, the execution results are displayed:

Execução finalizada sem erros.  
Resultado: consulta executada com sucesso. Levou 0ms, 1  
Na linha 10:  
UPDATE TB\_CURSO  
SET nome = 'ciencia da computacao'  
WHERE ID = 1;



The screenshot shows a SQL IDE with a toolbar at the top and a tab bar with tabs labeled SQL 1\*, SQL 2\*, SQL 3\*, SQL 4\*, SQL 5\*, and SQL 6\*. The main editor area contains the following SQL code:

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

Below the code editor, the execution results are displayed:

Execução finalizada sem erros.  
Resultado: consulta executada com sucesso. Levou 0ms, 0 linhas afetadas  
Na linha 9:  
DELETE FROM TB\_ALUNO WHERE ID = 1;