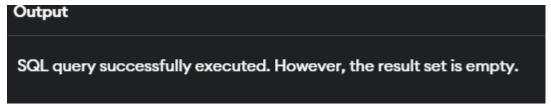
## Avaliação 3 de Banco de dados:

Crie um banco de dados SCA\_251.bd. Nele defina as Tabela TB\_ALUNO, TB\_PROFESSOR, TB\_CURSO e TB\_DISCIPLINA:

```
CREATE DATABASE SCA 242.bd;
CREATE TABLE TB ALUNOS (
   Matricula INT;
   Email VARCHAR (255) NOT NULL;
   PRIMARY KEY (Matricula)
CREATE TABLE TB PROFESSORES (
   SIAP INT;
   Email VARCHAR(255) NOT NULL;
   PRIMARY KEY (SIAP);
CREATE TABLE TB CURSO (
   Período INT;
   Id curso INT;
   PRIMARY KEY (Id curso);
CREATE TABLE TB DISCIPLINA (
   Período INT;
```

```
Id_disciplina INT;
    PRIMARY KEY (Id_disciplina);
);
```





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

```
Relacionamentos e Cardinalidade:

TB_ALUNO – Cursa – TB_CURSO (N : 1)

TB_ALUNO – Matriculados – TB_DISCIPLINA (N : N)
```

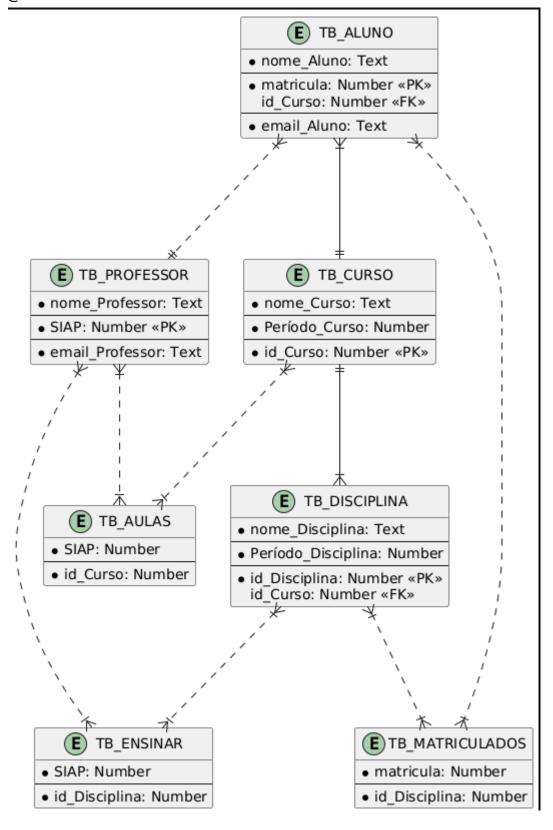
```
TB_PROFESSOR – Dão aulas – TB_CURSO (N : N)
TB_PROFESSOR – Ensinam – TB_DISCIPLINA (N : N)
TB_PROFESSOR – Orientam –TB_ALUNO (N : 1)

TB_CURSO – Matriz curricular – TB_DISCIPLINA (1 : N)
```

2) Crie um Diagrama de um Modelo Lógico no qual são definidas chaves primárias (id) e chaves estrangeiras(tabelald) 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;

```
@startuml
entity TB_ALUNO {
    * nome_Aluno: Text
    --
    * matricula: Number <<PK>>
    id_Curso: Number <<FK>>
    --
    *email_Aluno: Text
}
entity TB_PROFESSOR {
    * nome_Professor: Text
    --
    * SIAP: Number <<PK>>
    --
    * email_Professor: Text
}
entity TB_CURSO {
    * nome_Curso: Text
    --
    * Período_Curso: Number
    --
    * Período_Curso: Number
    --
```

```
* id_Curso: Number << PK>>
}
entity TB_DISCIPLINA {
 * nome_Disciplina: Text
 * Período_Disciplina: Number
 * id_Disciplina: Number << PK>>
 id Curso: Number << FK>>
}
entity TB_MATRICULADOS {
 * matricula: Number
 * id_Disciplina: Number
entity TB_AULAS {
 * SIAP: Number
 * id_Curso: Number
entity TB_ENSINAR {
* SIAP: Number
 * id_Disciplina: Number
}
TB_ALUNO }|...|| TB_PROFESSOR
TB_ALUNO } |--- || TB_CURSO
TB_ALUNO }|...|{ TB_MATRICULADOS
TB_DISCIPLINA } | ... | { TB_MATRICULADOS
TB_PROFESSOR }|...|{ TB_AULAS
TB_CURSO }|...|{ TB_AULAS
TB_PROFESSOR }|...|{ TB_ENSINAR
TB_DISCIPLINA } | ... | { TB_ENSINAR
TB_CURSO ||---|{ TB_DISCIPLINA
```



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

```
CREATE TABLE TB ALUNO (
    Nome VARCHAR (255) NOT NULL,
    Matricula INT PRIMARY KEY,
    Email VARCHAR (255) NOT NULL
CREATE TABLE TB PROFESSORES (
    Nome VARCHAR (255) NOT NULL,
    SIAP INT PRIMARY KEY,
    Email VARCHAR (255) NOT NULL
);
CREATE TABLE TB CURSO (
    Nome VARCHAR (255) NOT NULL,
   Periodo INT,
    Id curso INT PRIMARY KEY
);
CREATE TABLE TB DISCIPLINA (
    Nome VARCHAR (255) NOT NULL,
    Periodo INT,
    Id disciplina INT PRIMARY KEY
);
CREATE TABLE TB MATRICULADOS (
    Matricula INT,
    Id disciplina INT,
    PRIMARY KEY (Matricula, Id disciplina),
```

```
FOREIGN KEY (Matricula) REFERENCES
TB ALUNOS (Matricula),
    FOREIGN KEY (Id disciplina) REFERENCES
TB DISCIPLINA(Id disciplina)
);
CREATE TABLE TB AULAS (
    SIAP INT,
    Id curso INT,
    PRIMARY KEY (SIAP, Id curso),
    FOREIGN KEY (SIAP) REFERENCES
TB PROFESSORES (SIAP),
    FOREIGN KEY (Id curso) REFERENCES
TB CURSO(Id curso)
);
CREATE TABLE TB ENSINAR (
    SIAP INT,
    Id disciplina INT,
    PRIMARY KEY (SIAP, Id disciplina),
    FOREIGN KEY (SIAP) REFERENCES
TB PROFESSORES(SIAP),
   FOREIGN KEY (Id disciplina) REFERENCES
TB DISCIPLINA(Id disciplina)
);
```

TB_ALUNO				
Nome	Matricula			Email
empty				
TB_AULAS				
SIAP	lc	_curso		
empty				
TB_CURSO				
Nome	Periodo		ld (	curso
empty	Toriodo			Jul 30
TR DISCIPLINA				
TB_DISCIPLINA				
Nome	Periodo		Id_discipli	ina
empty				
TB_ENSINAR				
SIAP	ld_disci	plina		
empty				
TB_MATRICULADOS				
Matricula		Id_disciplin	a	
empty				
TB_PROFESSORES				
Nome	SIAP		E	Email
empty				

## Output

SQL query successfully executed. However, the result set is empty.

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

```
CREATE TABLE TB_ALUNO (
  Nome VARCHAR(255) NOT NULL,
  Matricula INT PRIMARY KEY,
  Email VARCHAR(255) NOT NULL
);
CREATE TABLE TB_PROFESSORES (
  Nome VARCHAR(255) NOT NULL,
  SIAP INT PRIMARY KEY,
 Email VARCHAR(255) NOT NULL
);
CREATE TABLE TB_CURSO (
  Nome VARCHAR(255) NOT NULL,
  Periodo INT,
  Id_curso INT PRIMARY KEY
);
CREATE TABLE TB_DISCIPLINA (
  Nome VARCHAR(255) NOT NULL,
  Periodo INT,
 Id_disciplina INT PRIMARY KEY
);
CREATE TABLE TB_MATRICULADOS (
  Matricula INT,
  Id_disciplina INT,
  PRIMARY KEY (Matricula, Id_disciplina),
  FOREIGN KEY (Matricula) REFERENCES TB_ALUNO(Matricula),
  FOREIGN KEY (Id_disciplina) REFERENCES
TB_DISCIPLINA(Id_disciplina)
);
```

```
CREATE TABLE TB_AULAS (
  SIAP INT,
  Id_curso INT,
  PRIMARY KEY (SIAP, Id_curso),
  FOREIGN KEY (SIAP) REFERENCES TB_PROFESSORES(SIAP),
 FOREIGN KEY (Id_curso) REFERENCES TB_CURSO(Id_curso)
);
CREATE TABLE TB_ENSINAR (
  SIAP INT,
  Id_disciplina INT,
  PRIMARY KEY (SIAP, Id_disciplina),
  FOREIGN KEY (SIAP) REFERENCES TB_PROFESSORES(SIAP),
  FOREIGN KEY (Id_disciplina) REFERENCES
TB_DISCIPLINA(Id_disciplina)
);
INSERT INTO TB_ALUNO (Nome, Matricula, Email)
VALUES ('Alyne', 20232011060, 'alyne@if.com');
INSERT INTO TB_PROFESSORES (Nome, SIAP, Email)
VALUES ('Taveira', 2023060, 'taveira@if.com');
INSERT INTO TB_CURSO (Nome, Periodo, Id_curso)
VALUES ('Informática', 4, 2001);
INSERT INTO TB_DISCIPLINA (Nome, Periodo, Id_disciplina)
VALUES ('Banco de Dados', 2, 3001);
INSERT INTO TB_MATRICULADOS (Matricula, Id_disciplina)
VALUES (20232011060, 3001);
INSERT INTO TB_AULAS (SIAP, Id_curso)
VALUES (2023060, 2001);
INSERT INTO TB_ENSINAR (SIAP, Id_disciplina)
```

```
VALUES (2023060, 3001);
SELECT
 A.Nome AS Nome_Aluno,
  D.Nome AS Nome_Disciplina
FROM
 TB_ALUNO A
JOIN
  TB MATRICULADOS M ON A.Matricula = M.Matricula
JOIN
 TB_DISCIPLINA D ON M.Id_disciplina = D.Id_disciplina;
SELECT
  P.Nome AS Nome_Professor,
  D.Nome AS Nome_Disciplina
FROM
  TB PROFESSORES P
JOIN
  TB_ENSINAR E ON P.SIAP = E.SIAP
JOIN
 TB_DISCIPLINA D ON E.Id_disciplina = D.Id_disciplina;
SELECT
 P.Nome AS Nome_Professor,
 C.Nome AS Nome_Curso
FROM
 TB_PROFESSORES P
JOIN
  TB_AULAS A ON P.SIAP = A.SIAP
JOIN
  TB_CURSO C ON A.Id_curso = C.Id_curso;
UPDATE TB_ALUNO
SET Email = 'alyne@ifce.com'
```

WHERE Matricula = 20232011060;

UPDATE TB\_CURSO SET Nome = 'Informática para Internet' WHERE Id\_curso = 2001;

UPDATE TB\_DISCIPLINA SET Nome = 'Banco de Dados' WHERE Id\_disciplina = 3001;

DELETE FROM TB\_ENSINAR
WHERE SIAP = 2023060 AND Id\_disciplina = 3001;

DELETE FROM TB\_AULAS
WHERE SIAP = 2023060 AND Id\_curso = 2001;

DELETE FROM TB\_MATRICULADOS
WHERE Matricula = 20232011060 AND Id\_disciplina = 3001;

DELETE FROM TB\_DISCIPLINA WHERE Id\_disciplina = 3001;

DELETE FROM TB\_CURSO WHERE Id\_curso = 2001;

DELETE FROM TB\_PROFESSORES WHERE SIAP = 5001;

DELETE FROM TB\_ALUNO WHERE Matricula = 1001;

Matricula		Email
20232011060		alyne@ifce.com
ld_curso		0
Periodo		Id_curso
	202320110	20232011060 Id_curs

TB_DISCIPLINA			
Nome	Periodo		Id_disciplina
empty			
TB_ENSINAR			
SIAP	Id_disciplina		
empty			
TB_MATRICULAD	os		
Matricula	Matricula Id_disciplina		
empty			

TB_PROFESSORES			
Nome	SIAP	Email	
Taveira	2023060	taveira@if.com	

Output		
Nome_Aluno	Nome_Disciplina	
Alyne	Banco de Dados	
Nome_Professor	Nome_Disciplina	
Taveira	Banco de Dados	

Nome_Professor	Nome_Curso
Taveira	Informática