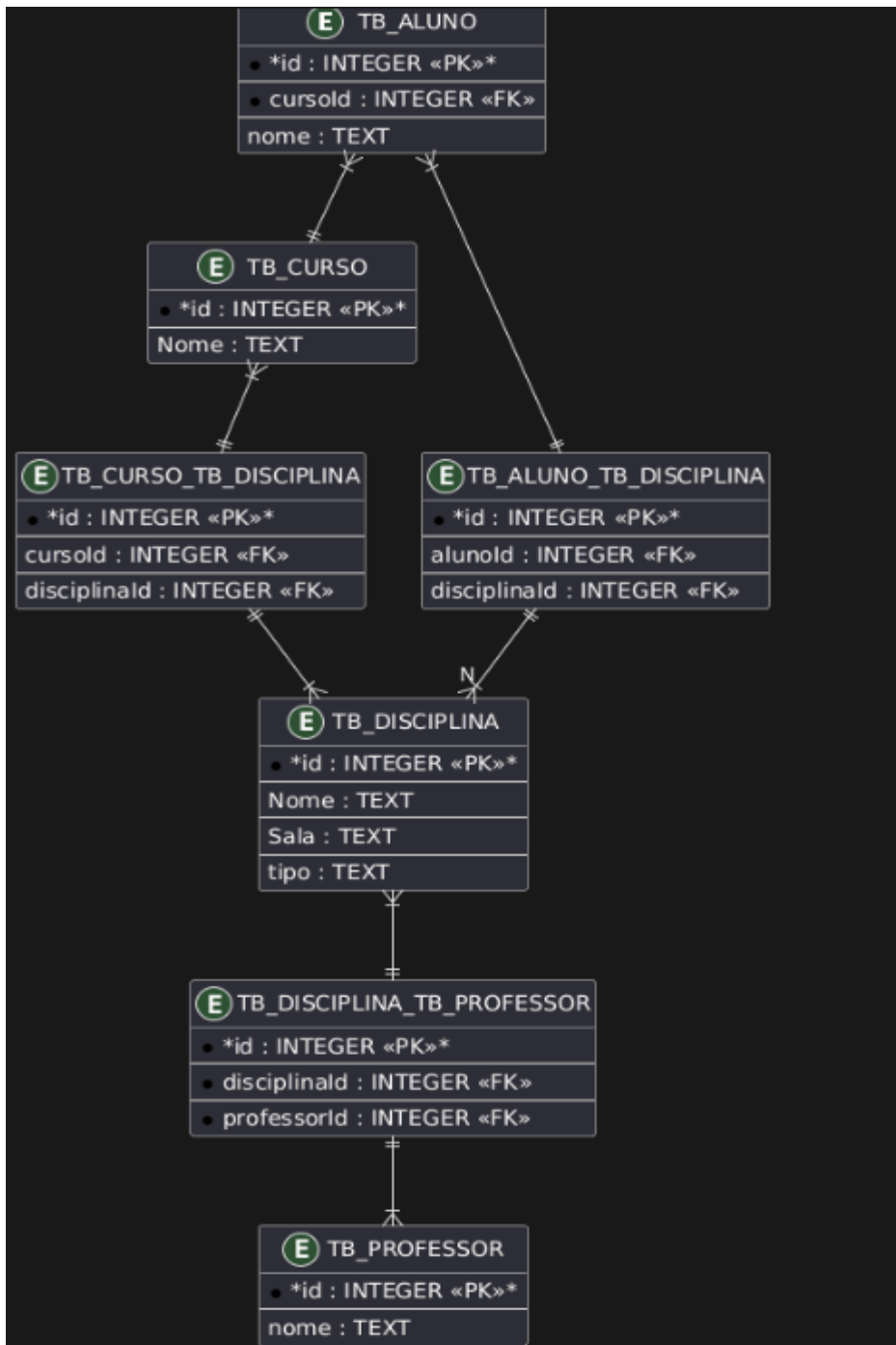


1) Identifique os possíveis relacionamentos com as suas cardinalidades envolvendo as tabelas.

- Os possíveis relacionamentos podem ser encontrados no diagrama abaixo
- Aluno com Disciplina -> (1,N):(1,N) | Aluno com Curso -> (1,N):(1,1) | Curso com Disciplina -> (1,N):(1,N) | Disciplina com Professor ->(1,N):(1,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;



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

CREATE DATABASE SCA_251;

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

- **INSERT**

COMANDO:

```
INSERT INTO TB_DISCIPLINA(nome)
VALUES ("Banco de Dados");
```

```
INSERT INTO TB_CURSO(nome)
VALUES ("Informática");
```

```
INSERT INTO TB_PROFESSOR(nome)
VALUES ("Ricardo");
```

```
INSERT INTO TB_ALUNO(nome, cursoId)
VALUES ("Davi", 1);
```

```
INSERT INTO TB_ALUNO_TB_DISCIPLINA(alunoId, disciplinaId)
VALUES (7, 1);
```

```
INSERT INTO TB_CURSO_TB_DISCIPLINA(cursoId, disciplinaId)
VALUES (1, 1);
```

```
INSERT INTO TB_PROFESSOR_TB_DISCIPLINA(professorId, disciplinaId)
VALUES (1, 1);
```

RESULTADO:

Execução finalizada sem erros.

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

Na linha 15:

```
INSERT INTO TB_ALUNO(nome)
VALUES ("Angelo");
```


VÁRIOS OUTROS VALORES FORAM POSTOS APÓS ESTA EXECUÇÃO...

TABELA APÓS:

Aluno:

<u>Matricula</u>	nome	<i>cursoId</i>
Filtro	Filtro	Filtro
1	Gustavo	<i>NULL</i>
2	Cayque	<i>NULL</i>
3	Arlenson	<i>NULL</i>
5	Luis	<i>NULL</i>
6	Lara	<i>NULL</i>
7	Davi	<i>NULL</i>
8	André	<i>NULL</i>
9	Alyne	1

Aluno-Disciplina:

Tabela:  TB_ALUNO_TB_DISCIPLINA

	<u>incritoId</u>	<i>alunoId</i>	<i>disciplinaId</i>
	Filtro	Filtro	Filtro
1	1	1	1
2	2	2	1
3	3	3	1
4	4	4	1
5	5	5	1
6	6	8	1
7	7	7	1
8	8	7	1

...

- UPDATE

COMANDO:

```
UPDATE TB_DISCIPLINA
SET Disciplina_id = 1
WHERE nome = "Banco de Dados";
```

```
UPDATE TB_PROFESSOR
SET Professor_id = 1
WHERE nome = "Ricardo";
```

```
UPDATE TB_ALUNO
SET Matricula = 4
WHERE nome = "Alyne";
```

```
UPDATE TB_DISCIPLINA
SET nome = "REDES 1"
WHERE nome = "Banco de Dados";
```

- SELECT

TABELA PADRÃO:

<u>Matricula</u>	nome	cursoId
Filtro	Filtro	Filtro
1	Gustavo	NULL
2	Cayque	NULL
3	Arlenson	NULL
5	Luis	NULL
6	Lara	NULL
7	Davi	NULL
8	André	NULL
9	Alyne	1

COMANDO:

```
SELECT nome FROM TB_ALUNO
WHERE Matricula IN (SELECT alunoId FROM TB_ALUNO_TB_DISCIPLINA);
```

RESULTADO:

	nome
1	Gustavo
2	Cayque
3	Arlenson
4	Alyne
5	Luis
6	Davi
7	André

Execução finalizada sem erros.
 Resultado: 7 linhas retornadas em 11 ms
 Na linha 1:
 SELECT nome FROM TB_ALUNO
 WHERE Matricula IN (SELECT alunoId FROM TB_ALUNO_TB_DISCIPLINA);

- DELETE

COMANDO:


```
DELETE FROM TB_ALUNO_TB_DISCIPLINA
WHERE alunoId > 6;
```

```
DELETE FROM TB_ALUNO
WHERE Matricula > 6;
```

Execução finalizada sem erros.
 Resultado: consulta executada com sucesso. Levou 40ms, 2 linhas afetada:
 Na linha 4:
 DELETE FROM TB_ALUNO
 WHERE Matricula > 6;

Tabela: TB_ALUNO_TB_DISCIPLINA ▼

	<u>incritoId</u>	<i>alunoId</i>	<i>disciplinaId</i>
	Filtro	Filtro	Filtro
1	1	1	1
2	2	2	1
3	3	3	1
4	4	4	1
5	5	5	1

Tabela:  TB_ALUNO

	<u>Matricula</u>	nome	<i>cursoId</i>
	Filtro	Filtro	Filtro
1	1	Gustavo	<i>NULL</i>
2	2	Cayque	<i>NULL</i>
3	3	Arlenson	<i>NULL</i>
4	4	Alyne	1
5	5	Luis	<i>NULL</i>
6	6	Lara	<i>NULL</i>

RESULTADO: Todos os alunos com número de alunold maiores que 6 foram retirados da tabela, mesmo processo os alunos com matricula maior que 6.