

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

Entidade 1	Entidade 2	Relacionamento	Cardinalidade
CURSO	ALUNO	Um curso possui vários alunos	1:N
CURSO	DISCIPLINA	Um curso possui várias disciplinas	1:N
PROFESSOR	DISCIPLINA	Um professor pode lecionar disciplinas	1:N
ALUNO	DISCIPLINA	Aluno cursa várias disciplinas e uma disciplina pode ter vários alunos	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;

```
@startuml
hide circle
skinparam linetype ortho
```

```
entity TB_CURSO {
+ ID_CURSO : INTEGER <<PK>>
--
NOME_CURSO : TEXT
DURACAO : INTEGER
}
```

```
entity TB_PROFESSOR {
+ ID_PROFESSOR : INTEGER <<PK>>
--
NOME_PROFESSOR : TEXT
ESPECIALIZACAO : TEXT
}
```

```
entity TB_ALUNO {
+ ID_ALUNO : INTEGER <<PK>>
--
NOME_ALUNO : TEXT
DATA_NASCIMENTO : TEXT
ID_CURSO : INTEGER <<FK>>
}
```

```
entity TB_DISCIPLINA {
+ ID_DISCIPLINA : INTEGER <<PK>>
--
```

```

    NOME_DISCIPLINA : TEXT
    CARGA_HORARIA : INTEGER
    ID_CURSO : INTEGER <<FK>>
    ID_PROFESSOR : INTEGER <<FK>>
}

entity TB_ALUNO_DISCIPLINA {
    + ID_ALUNO : INTEGER <<PK,FK>>
    + ID_DISCIPLINA : INTEGER <<PK,FK>>
}

```

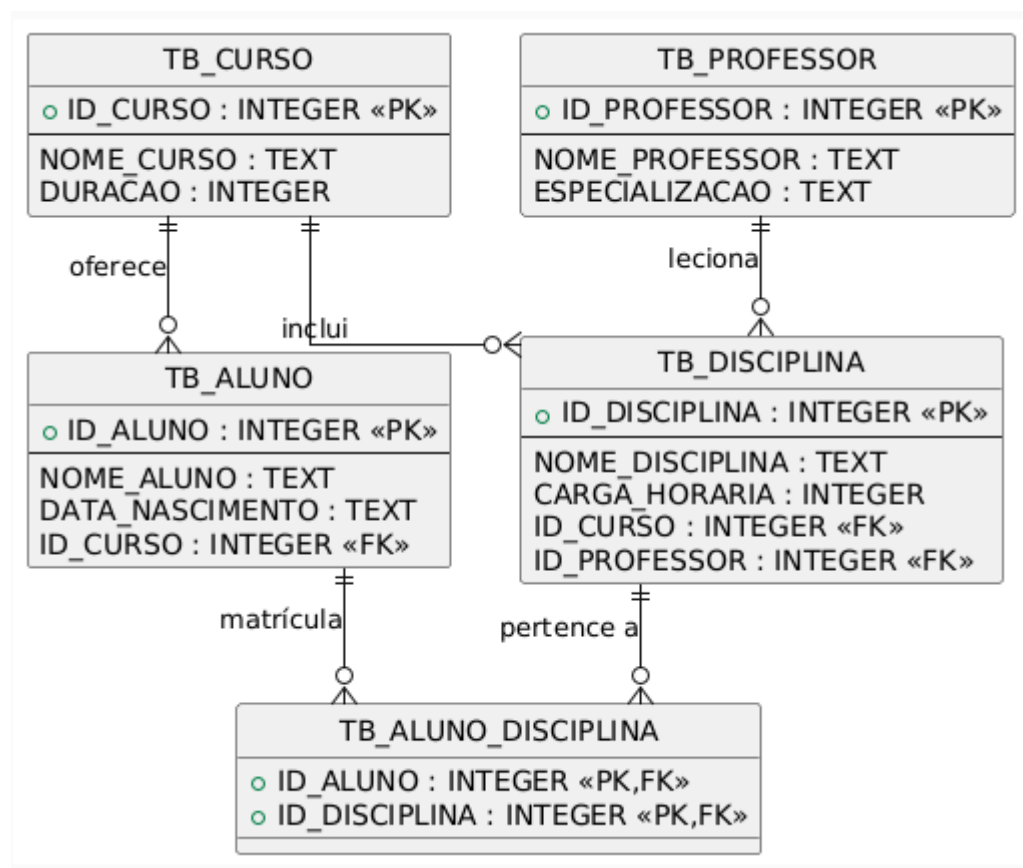
' Relacionamentos

```

TB_CURSO ||--o{ TB_ALUNO : "oferece"
TB_CURSO ||--o{ TB_DISCIPLINA : "inclui"
TB_PROFESSOR ||--o{ TB_DISCIPLINA : "leciona"
TB_ALUNO ||--o{ TB_ALUNO_DISCIPLINA : "matricula"
TB_DISCIPLINA ||--o{ TB_ALUNO_DISCIPLINA : "pertence a"

```

@enduml



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

```
CREATE TABLE TB_ALUNO (
```

```

ID_ALUNO INTEGER PRIMARY KEY AUTOINCREMENT,
NOME_ALUNO TEXT NOT NULL,
CPF TEXT UNIQUE NOT NULL,
DATA_NASCIMENTO DATE,
EMAIL TEXT UNIQUE,
TELEFONE TEXT
);
CREATE TABLE TB_PROFESSOR (
ID_PROFESSOR INTEGER PRIMARY KEY AUTOINCREMENT,
NOME_PROFESSOR TEXT NOT NULL,
CPF TEXT UNIQUE NOT NULL,
EMAIL TEXT UNIQUE,
TELEFONE TEXT
);
CREATE TABLE TB_CURSO (
ID_CURSO INTEGER PRIMARY KEY AUTOINCREMENT,
NOME_CURSO TEXT NOT NULL,
DESCRICAO TEXT,
CARGA_HORARIA INTEGER
);
CREATE TABLE TB_DISCIPLINA (
ID_DISCIPLINA INTEGER PRIMARY KEY AUTOINCREMENT,
NOME_DISCIPLINA TEXT NOT NULL,
ID_PROFESSOR INTEGER,
ID_CURSO INTEGER,
CARGA_HORARIA INTEGER,
FOREIGN KEY (ID_PROFESSOR) REFERENCES TB_PROFESSOR(ID_PROFESSOR),
FOREIGN KEY (ID_CURSO) REFERENCES TB_CURSO(ID_CURSO)
);

```

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

SQL 1*

```
1  INSERT INTO TB_DISCIPLINA (NOME_DISCIPLINA, ID_PROFESSOR, ID_CURSO, CARGA_HORARIA)
2  VALUES ('Banco de Dados I', 1, 1, 60);
3
4  INSERT INTO TB_DISCIPLINA (NOME_DISCIPLINA, ID_PROFESSOR, ID_CURSO, CARGA_HORARIA)
5  VALUES ('Programação Orientada a Objetos', 1, 1, 80);
6
7  INSERT INTO TB_DISCIPLINA (NOME_DISCIPLINA, ID_PROFESSOR, ID_CURSO, CARGA_HORARIA)
8  VALUES ('Estrutura de Dados', NULL, 1, 70); -- Exemplo sem professor atribuído
```

Execução finalizada sem erros.

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

Na linha 7:

```
INSERT INTO TB_DISCIPLINA (NOME_DISCIPLINA, ID_PROFESSOR, ID_CURSO, CARGA_HORARIA)
VALUES ('Estrutura de Dados', NULL, 1, 70);
```

```
1 SELECT * FROM TB_CURSO;  
2 SELECT NOME_CURSO, CARGA_HORARIA FROM TB_CURSO WHERE CARGA_HORARIA > 3000;|
```

Resultado: 0 linhas retornadas em 13 ms

Na linha 2:

```
SELECT NOME_CURSO, CARGA_HORARIA FROM TB_CURSO WHERE CARGA_HORARIA > 3000;
```

```
1 UPDATE TB_PROFESSOR
2 SET TELEFONE = '(31) 99999-0000'
3 WHERE ID_PROFESSOR = 1;
```

Execução finalizada sem erros.
Resultado: consulta executada com sucesso. Levou 0ms, 1 linhas afetadas
Na linha 1:
UPDATE TB_PROFESSOR
SET TELEFONE = '(31) 99999-0000'
WHERE ID_PROFESSOR = 1;

```
1 DELETE FROM TB_ALUNO WHERE ID_ALUNO = 2;
```

Execução finalizada sem erros.

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

Na linha 1:

```
DELETE FROM TB_ALUNO WHERE ID_ALUNO = 2;
```