

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

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

@startuml

hide circle

skinparam linetype polyline

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

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

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

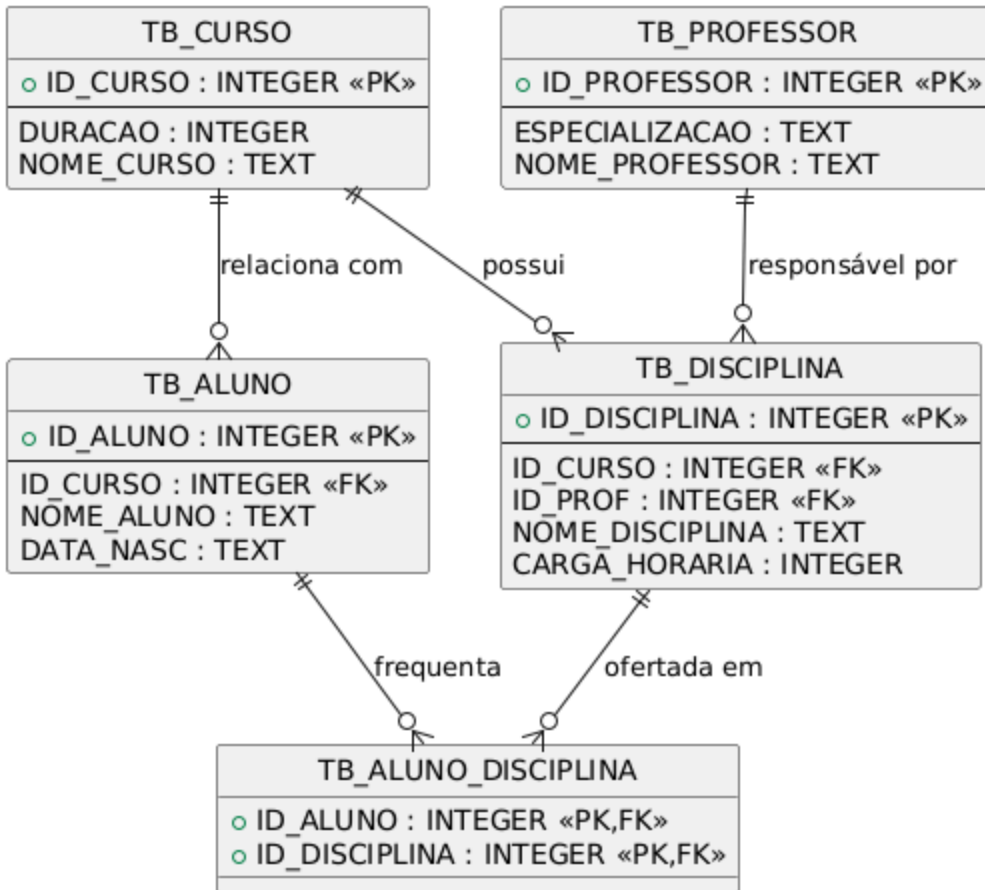
```
entity TB_DISCIPLINA {
  + ID_DISCIPLINA : INTEGER <<PK>>
  --
  ID_CURSO : INTEGER <<FK>>
  ID_PROF : INTEGER <<FK>>
  NOME_DISCIPLINA : TEXT
  CARGA_HORARIA : INTEGER
}
```

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

' Relacionamentos ajustados

```
TB_CURSO ||--o{ TB_ALUNO : "relaciona com"
TB_CURSO ||--o{ TB_DISCIPLINA : "possui"
TB_PROFESSOR ||--o{ TB_DISCIPLINA : "responsável por"
TB_ALUNO ||--o{ TB_ALUNO_DISCIPLINA : "frequenta"
TB_DISCIPLINA ||--o{ TB_ALUNO_DISCIPLINA : "ofertada em"
```

@enduml



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

```

CREATE TABLE TB_ALUNO (
    ID_ALUNO INTEGER PRIMARY KEY AUTOINCREMENT,
    CPF TEXT NOT NULL UNIQUE,
    NOME_ALUNO TEXT NOT NULL,
    EMAIL TEXT UNIQUE,
    TELEFONE TEXT,
    NASCIMENTO TEXT
);

CREATE TABLE TB_PROFESSOR (
    ID_PROFESSOR INTEGER PRIMARY KEY AUTOINCREMENT,
    CPF TEXT NOT NULL UNIQUE,
    NOME_PROFESSOR TEXT NOT NULL,
    TELEFONE TEXT,
    EMAIL TEXT UNIQUE
);
  
```

```
CREATE TABLE TB_CURSO (  
    ID_CURSO INTEGER PRIMARY KEY AUTOINCREMENT,  
    NOME_CURSO TEXT NOT NULL,  
    CARGA_HORARIA INTEGER,  
    DESCRICAO TEXT  
);
```

```
CREATE TABLE TB_DISCIPLINA (  
    ID_DISCIPLINA INTEGER PRIMARY KEY AUTOINCREMENT,  
    NOME_DISCIPLINA TEXT NOT NULL,  
    CARGA_HORARIA INTEGER,  
    PROFESSOR_ID INTEGER,  
    CURSO_ID INTEGER,  
    FOREIGN KEY (PROFESSOR_ID) REFERENCES TB_PROFESSOR(ID_PROFESSOR),  
    FOREIGN KEY (CURSO_ID) 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.**

DB Browser for SQLite - C:\Users\PC DE CASA\BD-251\SQLITE\SCA\_251.bd.db

Arquivo Editar Exibir Ferramentas Ajuda

New Database Open Database Escrever modificações Reverter modificações Undo Open Project Save Project

Database Structure Browse Data Edit Pragmas Execute SQL



SQL 1\*

```
1 INSERT INTO TB_DISCIPLINA (NOME_DISCIPLINA, ID_CURSO, ID_PROFESSOR, CARGA_HORARIA)
2 VALUES ('PORTUGUES', 1, 1, 60);
```

Execução finalizada sem erros.

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

Na linha 1:

```
INSERT INTO TB_DISCIPLINA (NOME_DISCIPLINA, ID_CURSO, ID_PROFESSOR, CARGA_HORARIA)
VALUES ('PORTUGUES', 1, 1, 60);
```

New DatabaseOpen DatabaseEscrever modificaçõesReverter modificaçõesUndoOpen ProjectSave ProjectAn

Database StructureBrowse DataEdit PragmasExecute SQL

SQL 1\*

1SELECT \* FROM TB\_CURSO;

2

	ID_CURSO	NOME_CURSO	DESCRICAO	CARGA_HORARIA
1	1	Letras	Curso de Letras com habilitação em ...	3200
2	2	Letras	Curso de Letras com habilitação em ...	3200

Execução finalizada sem erros.  
Resultado: 2 linhas retornadas em 8 ms  
Na linha 1:  
SELECT \* FROM TB\_CURSO;

SQL 1\*

```
1 UPDATE TB_DISCIPLINA
2 SET ID_PROFESSOR = 2
3 WHERE ID_DISCIPLINA = 1;
4
```

Execução finalizada sem erros.  
Resultado: consulta executada com sucesso. Levou 1ms, 1 linhas afetadas  
Na linha 1:  
UPDATE TB\_DISCIPLINA  
SET ID\_PROFESSOR = 2  
WHERE ID\_DISCIPLINA = 1;

DB Browser for SQLite - C:\Users\PC DE CASA\BD-251\SQLITE\SCA\_251.bd.db

Arquivo Editar Exibir Ferramentas Ajuda

New Database Open Database Escrever modificações Reverter modificações Undo Open Project Save Project

Database Structure Browse Data Edit Pragma Execute SQL



SQL 1\*

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

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