

Trabalho Experimental

Enunciado F

Licenciatura em Engenharia Informática

Bases de Dados

Paulo Nogueira Martins

Autores:

Francisco Ruano – al78474

Maria Inês Cardoso – al78222

Miguel Teixeira – al78321

Rui Madureira – al78282

Vila Real, 2024

Índice

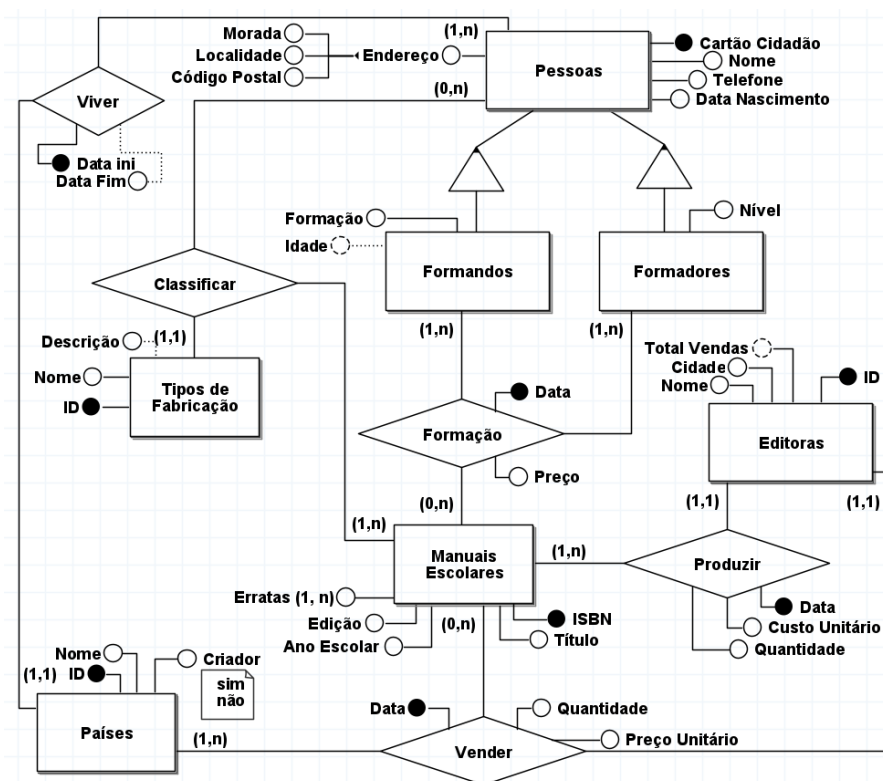
1.	Introdução	1
2.	Enquadramento Teórico	2
I.	Modelo Entidade-Relacionamento	2
II.	Modelo Relacional	2
III.	Normalização até à Terceira Forma Normal	2
IV.	Modelo Físico da base de dados	3
V.	Implementação do Modelo Físico e criação do diagrama da base de dados ...	3
VI.	Definição e Manipulação de dados	3
3.	Desenvolvimento	4
I.	Mapeamento do Diagrama E-R para o Modelo Relacional	4
II.	Normalização do Modelo Relacional até à Terceira Forma Normal	5
	Primeira Forma Normal (1FN)	5
	Segunda Forma Normal (2FN)	6
	Terceira Forma Normal (3FN)	7
III.	Implementação do Modelo Físico da Base de Dados	9
IV.	Criação do diagrama da base de dados	12
V.	Inserção de dados nas tabelas	12
VI.	Pesquisa de informação na base de dados	15
VII.	Stored Procedures	17
VIII.	Triggers	18
4.	Conclusão	19

1. Introdução

No âmbito da unidade curricular de Base de Dados foi-nos proposta a elaboração de um projeto experimental que consiste na criação de uma base de dados relativa a manuais escolares.

Para este processo começamos por fazer o mapeamento do diagrama Entidade-Relacionamento para o modelo relacional. De seguida, procedemos à normalização do modelo relacional até à sua terceira forma normal (3FN), e implementamos o modelo físico da base de dados recorrendo à linguagem SQL. Por fim, utilizando a implementação no SQL, criamos o diagrama final da base de dados.

O enunciado que nos foi atribuído fornece uma visão abrangente do processo de produção e venda de manuais escolares, detalhando as entidades e relações envolvidas no mesmo. O seguinte modelo Entidade-Relacionamento traduz este processo:



O sistema de gestão de vendas de livros é uma plataforma abrangente que armazena informações sobre Pessoas, Manuais Escolares, Países, entre outros.... Além disso, oferece recursos como a análise dos países mais ativos no mercado de livros, tornando-se uma ferramenta essencial e eficiente nas operações relacionadas à venda de livros e formação.

A interligação entre Pessoas, Livros e Formações, bem como a capacidade de identificação precisa por meio de atributos únicos, como cartão de cidadão e ISBN, garante uma gestão integrada e eficaz da base de dados.

2. Enquadramento Teórico

I. Modelo Entidade-Relacionamento

Um modelo E-R, ou Modelo Entidade-Relacionamento, é uma representação visual dos dados e das relações entre eles, num sistema de informação. Utiliza entidades para representar objetos do mundo real, e relacionamentos para descrever como essas entidades interagem entre si. Os modelos E-R são fundamentais na fase de conceção de bases de dados, uma vez que ajudam a entender e comunicar a estrutura do sistema de forma clara e concisa.

II. Modelo Relacional

Um modelo relacional é uma representação estruturada dos dados em forma de tabelas. Nesse modelo, os dados são organizados em conjuntos de linhas e colunas, onde cada linha representa uma instância de uma entidade e cada coluna representa um atributo dessa entidade.

Além disso, permite ainda definir a chave de cada tabela que pode ser de diferentes tipos, sendo que as que utilizamos no contexto do nosso enunciado são as seguintes:

- Chave primária: É a chave que efetivamente identifica;
- Chave estrangeira: É o atributo ou conjunto de atributos de uma relação que é chave primária noutra relação.

III. Normalização até à Terceira Forma Normal

A normalização do modelo relacional até à Terceira Forma Normal (3FN) é um processo de refinamento do modelo de dados com o objetivo de eliminar redundâncias e anomalias de atualização, garantindo assim a consistência e integridade dos dados. Esta normalização consiste na aplicação de uma série de regras para decompor as tabelas em estruturas mais simples e coesas, reduzindo a redundância e evitando dependências funcionais indesejadas.

Até se atingir a Terceira Forma Normal, o modelo relacional tem de passar pelas primeira e segunda normalizações. A Primeira Forma Normal (1FN) não contém atributos multivalor, nem grupos repetidos, uma vez que é obtida a partir do Mapeamento do Diagrama E-R para o Modelo Relacional.

Já na Segunda Forma Normal (2FN), desenvolvida a partir da 1FN, todos os atributos não chave têm de depender funcionalmente da totalidade da chave.

Finalmente, na 3FN, obtida a partir da 2FN, todos os atributos não chave não podem depender funcionalmente uns dos outros, e, na sua verificação, deverão ser eliminados atributos derivados, uma vez que estes implicariam uma dependência funcional entre elementos não chave. A 3FN é um dos níveis de normalização mais comuns e visa garantir que cada atributo de

uma tabela seja diretamente dependente da chave primária, eliminando assim dependências transitivas.

IV. Modelo Físico da base de dados

A implementação do modelo físico da base de dados consiste na tradução do modelo conceptual (E-R ou relacional) em estruturas concretas de armazenamento de dados num sistema de gestão de bases de dados específico. Desta forma, envolve a definição de tipos de dados, índices, restrições de integridade, entre outros elementos, de acordo com as capacidades e requisitos do sistema de gestão de bases de dados escolhido.

V. Implementação do Modelo Físico e criação do diagrama da base de dados

Por fim, através da aplicação *SQL Server Management Studio*, é possível a criação do diagrama da base de dados. Este diagrama é uma representação visual do modelo de dados, que mostra as tabelas, as suas relações e os atributos.

Assim, é uma ferramenta útil para comunicar a estrutura da base de dados de forma clara e acessível para todos os envolvidos no projeto, incluindo desenvolvedores, gestores de projeto e utilizadores finais.

VI. Definição e Manipulação de dados

Aquando da criação da base de dados, é possível criar tabelas nas quais são definidos atributos/ valores, bem como as regras a que estes devem obedecer. Assim, utilizamos os seguintes comandos disponibilizados pelo SQL para esse mesmo fim:

- CREATE DATABASE – Permite criar uma base de dados
- CREATE TABLE – Permite criar uma tabela
- NOT NULL – Impede a introdução de valores nulos numa coluna
- DEFAULT – Define um valor por defeito, caso não sejam introduzidos valores numa coluna
- CHECK() – Permite definir critérios de validação de dados introduzidos numa coluna
- UNIQUE – Garante que todos os valores de uma coluna sejam distintos
- PRIMARY KEY – Indica a chave primária de uma tabela
- FOREIGN KEY – Indica a chave estrangeira de uma tabela
- REFERENCES – Indica a que tabela pertence a chave estrangeira de uma determinada tabela

Após a criação da base de dados, podemos manipular as tabelas criadas, adicionando, alterando ou removendo atributos das mesmas. Isto é possível através de comandos disponibilizados pelo SQL, tais como:

- INSERT – Permite inserir novos registos numa determinada tabela
- SELECT – Permite consultar uma base de dados relacional
- SubQueries – Referem-se a qualquer SELECT ou conjunto de SELECTs contidos dentro de outro SELECT
- Stored Procedures – Armazena um conjunto de procedimentos que podem ser executados de forma repetida
- CREATE PROCEDURE – Permite criar um procedimento
- EXECUTE – Permite executar procedimentos
- Triggers - Stored Procedures que são executados automaticamente, em resposta a modificações dos dados de uma tabela
- CREATE TRIGGER – Permite criar um Trigger

3. Desenvolvimento

I. Mapeamento do Diagrama E-R para o Modelo Relacional

Pessoas (CC, Nome, Data_Nascimento, Endereco_Morada, Endereco_CodigoPostal, Endereco_Localidade, Telefone)

Formandos (CC, Idade, Formacao)

- CC referencia Pessoas

Formadores (CC, Nivel)

- CC referencia Pessoas

ManuaisEscolares (ISBN, Titulo, AnoEscolar, Edicao, Erratas)

Editoras (Editora_ID, Nome_Editora, Cidade_Editora, TotVendas)

TiposdeFabricacao (Fabricacao_ID, NomeFabricacao, Descricao)

Países (Pais_ID, NomePais, Criador)

Viver (DataInicio, DataFim, Pais_ID, CC)

- Pais_ID referencia Países
- CC referencia Pessoas

Classificar (CC, ISBN, Fabricacao_ID)

- CC referencia Pessoas
- ISBN referencia ManuaisEscolares
- Fabricacao_ID referencia TiposdeFabricacao

Formacao (DataFormacao, PrecoFormacao, ISBN, CC)

- ISBN referencia ManuaisEscolares
- CC referencia Pessoas

Produzir (DataProduzir, CustoUnidade, Quantidade_Produzir, Editora_ID, ISBN)

- Editora_ID referencia Editoras
- ISBN referencia ManuaisEscolares

Vender (DataVender, Quantidade_Vender, Preco_Unidade, Pais_ID, ISBN, Editora_ID)

- Pais_ID referencia Países
- ISBN referencia ManuaisEscolares
- Editora_ID referencia Editoras

II. Normalização do Modelo Relacional até à Terceira Forma Normal

Primeira Forma Normal (1FN)

Pessoas (CC, Nome, Data_Nascimento, Endereco_Morada, Endereco_CodigoPostal, Endereco_Localidade, Telefone)

Formandos (CC, Idade, Formacao)

- CC referencia Pessoas

Formadores (CC, Nivel)

- CC referencia Pessoas

ManuaisEscolares (ISBN, Titulo, AnoEscolar, Edicao, Erratas)

Editoras (Editora_ID, Nome_Editora, Cidade_Editora, TotVendas)

TiposdeFabricacao (Fabricacao_ID, NomeFabricacao, Descricao)

Países (Pais_ID, NomePaís, Criador)

Viver (DataInicio, DataFim, Pais_ID, CC)

- Pais_ID referencia Países
- CC referencia Pessoas

Classificar (CC, ISBN, Fabricacao_ID)

- CC referencia Pessoas
- ISBN referencia ManuaisEscolares
- Fabricacao_ID referencia TiposdeFabricacao

Formacao (DataFormacao, PrecoFormacao, ISBN, CC)

- ISBN referencia ManuaisEscolares
- CC referencia Pessoas

Produzir (DataProduzir, CustoUnidade, Quantidade_Produzir, Editora_ID, ISBN)

- Editora_ID referencia Editoras
- ISBN referencia ManuaisEscolares

Vender (DataVender, Quantidade_Vender, Preco_Unidade, Pais_ID, ISBN, Editora_ID)

- Pais_ID referencia Países
- ISBN referencia ManuaisEscolares
- Editora_ID referencia Editoras

Segunda Forma Normal (2FN)

Pessoas (CC, Nome, DataNascimento, Telefone, ID_Endereco)

- ID_Endereco referencia Endereco

Endereco (ID_Endereco, EnderecoMorada, EnderecoLocalidade, EnderecoCodigoPostal)

Formandos (CC, Idade, Formacao)

- CC referencia Pessoas

Formadores (CC, Nivel)

- CC referencia Pessoas

ManuaisEscolares (ISBN, Titulo, AnoEscolar, Edicao)

Erratas (Errata_ID, Texto)

ManuaisEscolares_Erratas (Errata_ID, ISBN)

- Errata_ID referencia Erratas
- ISBN referencia ManuaisEscolares

Editoras (Editora_ID, Nome_Editora, Cidade_Editora, TotVendas)

TiposdeFabricacao (Fabricacao_ID, NomeFabricacao, Descricao)

Países (Pais_ID, NomePais, Criador)

Viver (DataInicio, DataFim, Pais_ID, CC)

- Pais_ID referencia Países
- CC referencia Pessoas

Classificar (CC, ISBN, Fabricacao_ID)

- CC referencia Pessoas
- ISBN referencia ManuaisEscolares
- Fabricacao_ID referencia TiposdeFabricacao

Formacao (DataFormacao, PrecoFormacao, ISBN, CC)

- ISBN referencia ManuaisEscolares
- CC referencia Pessoas

Produzir (DataProduzir, CustoUnidade, Quantidade_Produzir, Editora_ID, ISBN)

- Editora_ID referencia Editoras
- ISBN referencia ManuaisEscolares

Vender (DataVender, Quantidade_Vender, Preco_Unidade, Pais_ID, ISBN, Editora_ID)

- Pais_ID referencia Países
- ISBN referencia ManuaisEscolares
- Editora_ID referencia Editoras

Terceira Forma Normal (3FN)

Pessoas (CC, Nome, Data_Nascimento, Telefone, ID_Endereco)

- ID_Endereco referencia Endereco

Endereco (ID_Endereco, Endereco_Morada, Endereco_Localidade, Endereco_CodigoPostal)

Formandos (CC, Idade, Formacao)

- CC referencia Pessoas

Formadores (CC, Nivel)

- CC referencia Pessoas

ManuaisEscolares (ISBN, Titulo, AnoEscolar, Edicao, Fabricacao_ID)

- Fabricacao_ID referencia TiposdeFabricacao

Erratas (Errata_ID, Texto)

ManuaisEscolares_Erratas (~~Errata_ID~~, ISBN)

- ISBN referencia ManuaisEscolares
- Errata_ID referencia Erratas

Editoras (Editora_ID, Nome_Editora, Cidade_Editora, TotVendas)

TiposdeFabricacao (Fabricacao_ID, NomeFabricacao, Descricao)

Países (Pais_ID, NomePais, Criador)

Viver (DataInicio, DataFim, ~~Pais_ID~~, CC)

- Pais_ID referencia Países
- CC referencia Pessoas

Classificar (CC, ISBN, ~~Fabricacao_ID~~)

- CC referencia Pessoas
- ISBN referencia ManuaisEscolares
- Fabricacao_ID referencia TiposdeFabricacao

Formacao (DataFormacao, PrecoFormacao, ISBN, CC)

- ISBN referencia ManuaisEscolares
- CC referencia Pessoas

Produzir (DataProduzir, CustoUnidade, Quantidade_Produzir, ~~Editora_ID~~, ISBN)

- Editora_ID referencia Editoras
- ISBN referencia ManuaisEscolares

Vender (DataVender, Quantidade_Vender, Preco_Unidade, ~~Pais_ID~~, ISBN, ~~Editora_ID~~)

- Pais_ID referencia Países
- ISBN referencia ManuaisEscolares
- Editora_ID referencia Editoras

III. Implementação do Modelo Físico da Base de Dados

```

USE master
GO
CREATE DATABASE Trabalho_BD
GO
USE Trabalho_BD
GO
CREATE TABLE Endereco(
    ID_Endereco          CHAR(5)          NOT NULL,
    Endereco_Morada       VARCHAR(50)      NOT NULL,
    Endereco_CodigoPostal CHAR(8)          NOT NULL,
    Endereco_Localidade   VARCHAR(50)      NOT NULL,

    CHECK (ID_Endereco LIKE '[0-9][0-9][0-9][0-9][0-9]'),
    CHECK (Endereco_CodigoPostal LIKE '[0-9][0-9][0-9][0-9]-[0-9][0-9][0-9]'),
    PRIMARY KEY (ID_Endereco)
)

CREATE TABLE Pessoas(
    CC                   CHAR(13)          NOT NULL,
    Nome                 VARCHAR(50)       NOT NULL,
    Data_Nascimento      SMALLDATETIME    NOT NULL,
    ID_Endereco          CHAR(5)          NOT NULL,
    Telefone             BIGINT           NOT NULL,

    CHECK (Telefone LIKE '[2-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
    CHECK (CC LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]-[0-9][A-Z][A-Z][0-9]'),
    PRIMARY KEY (CC),
    FOREIGN KEY (ID_Endereco) REFERENCES Endereco(ID_Endereco)
)

CREATE TABLE Formadores(
    CC                   CHAR(13)          NOT NULL,
    Nivel               INT              NOT NULL,

    CHECK (Nivel>=1 AND Nivel<=10),
    PRIMARY KEY (CC),
    FOREIGN KEY (CC) REFERENCES Pessoas(CC)
)

CREATE TABLE TiposdeFabricacao(
    Fabricacao_ID        CHAR(5)          NOT NULL,
    NomeFabricacao       VARCHAR(50)      NOT NULL,
    Descricao            VARCHAR(255),

    CHECK (Fabricacao_ID LIKE '[0-9][0-9][0-9][0-9][0-9]'),
    PRIMARY KEY (Fabricacao_ID)
)

CREATE TABLE ManuaisEscolares(
    ISBN                 BIGINT           NOT NULL,
    Titulo               VARCHAR(50)      NOT NULL,
    AnoEscolar           INT              NOT NULL,
    Edicao                INT              NOT NULL,
    Fabricacao_ID        CHAR(5)          NOT NULL,

    CHECK (Edicao>0),
    CHECK (AnoEscolar BETWEEN 1 AND 12),
    CHECK (ISBN LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
    PRIMARY KEY (ISBN),
    FOREIGN KEY (Fabricacao_ID) REFERENCES TiposdeFabricacao (Fabricacao_ID)
)

```

```

CREATE TABLE Erratas(
    Errata_ID          CHAR(5)          NOT NULL,
    Texto              VARCHAR(255)     NOT NULL,

    CHECK (Errata_ID LIKE '[0-9][0-9][0-9][0-9][0-9]'),
    PRIMARY KEY (Errata_ID)
)

CREATE TABLE ManuaisEscolares_Erratas(
    ISBN              BIGINT            NOT NULL,
    Errata_ID         CHAR(5)          NOT NULL,

    PRIMARY KEY (Errata_ID),
    FOREIGN KEY (Errata_ID) REFERENCES Erratas(Errata_ID),
    FOREIGN KEY (ISBN) REFERENCES ManuaisEscolares(ISBN)
)

CREATE TABLE Editoras(
    Editora_ID        CHAR(5)          NOT NULL,
    Nome_Editora      VARCHAR(50)      NOT NULL,
    Cidade_Editora    VARCHAR(50)      NOT NULL,
    TotVendas         INT              NOT NULL,

    CHECK (Editora_ID LIKE '[0-9][0-9][0-9][0-9][0-9]'),
    CHECK (TotVendas >= 0),
    PRIMARY KEY(Editora_ID)
)

CREATE TABLE Pais(
    Pais_ID           CHAR(5)          NOT NULL,
    NomePais          VARCHAR(50)      NOT NULL,
    Criador           BIT              NOT NULL DEFAULT 0,

    CHECK (Pais_ID LIKE '[0-9][0-9][0-9][0-9][0-9]'),
    PRIMARY KEY (Pais_ID)
)

CREATE TABLE Viver(
    DataInicio        SMALLDATETIME    NOT NULL DEFAULT GETDATE(),
    DataFim           SMALLDATETIME,
    Pais_ID           CHAR(5)          NOT NULL,
    CC                CHAR(13)         NOT NULL,

    CHECK (DataFim > DataInicio),
    PRIMARY KEY (DataInicio, Pais_ID, CC),
    FOREIGN KEY (Pais_ID) REFERENCES Pais(Pais_ID),
    FOREIGN KEY (CC) REFERENCES Pessoas(CC)
)

CREATE TABLE Classificar(
    CC                CHAR(13)          NOT NULL,
    ISBN              BIGINT            NOT NULL,
    Fabricacao_ID     CHAR(5)          NOT NULL,

    PRIMARY KEY (CC, ISBN, Fabricacao_ID),
    FOREIGN KEY (CC) REFERENCES Pessoas(CC),
    FOREIGN KEY (ISBN) REFERENCES ManuaisEscolares(ISBN),
    FOREIGN KEY (Fabricacao_ID) REFERENCES TiposdeFabricacao(Fabricacao_ID),
)

```

```

CREATE TABLE Formacao(
    DataFormacao          SMALLDATETIME          NOT NULL,
    PrecoFormacao          MONEY                  NOT NULL,
    ISBN                   BIGINT                 NOT NULL,
    CC                      CHAR(13)              NOT NULL,
    ID_Formacao            CHAR(5)                NOT NULL,

    CHECK (PrecoFormacao >= 0),
    CHECK (ID_Formacao LIKE '[0-9][0-9][0-9][0-9][0-9]'),
    PRIMARY KEY (ID_Formacao),
    UNIQUE (ISBN, CC),
    FOREIGN KEY (ISBN) REFERENCES ManuaisEscolares(ISBN),
    FOREIGN KEY (CC) REFERENCES Pessoas(CC)
)

CREATE TABLE Formandos(
    CC                      CHAR(13)              NOT NULL,
    Idade                   INT,
    ID_Formacao            CHAR(5)                NOT NULL,

    CHECK(Idade > 0),
    PRIMARY KEY (CC, ID_Formacao),
    FOREIGN KEY (CC) REFERENCES Pessoas(CC),
    FOREIGN KEY (ID_Formacao) REFERENCES Formacao(ID_Formacao)
)

CREATE TABLE Produzir(
    DataProduzir           SMALLDATETIME          NOT NULL,
    CustoUnidade           MONEY                  NOT NULL,
    Quantidade_Produzir     INT                   NOT NULL,
    Editora_ID             CHAR(5)                NOT NULL,
    ISBN                   BIGINT                 NOT NULL,

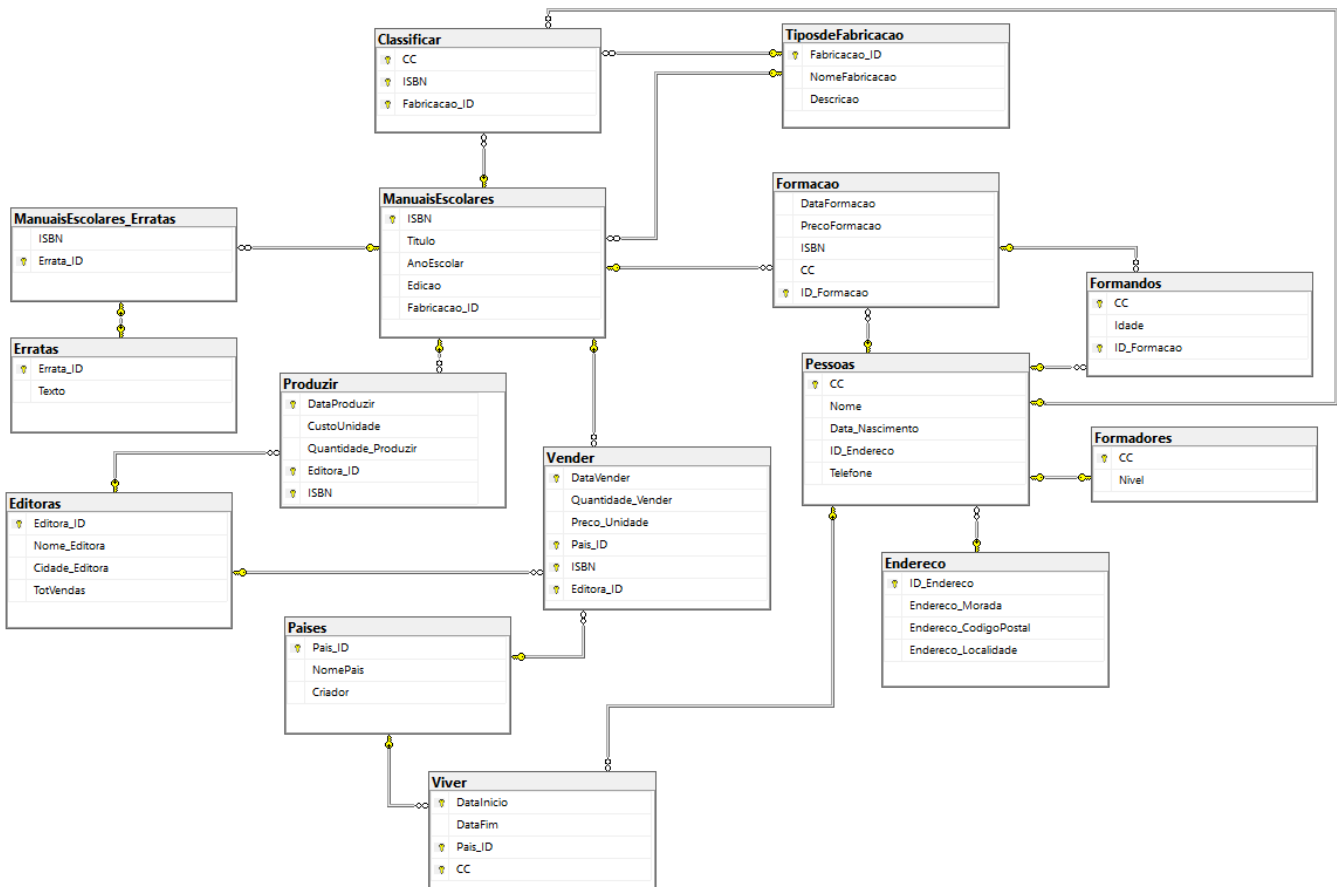
    CHECK (CustoUnidade>0),
    CHECK (Quantidade_Produzir>0),
    PRIMARY KEY (DataProduzir, Editora_ID, ISBN),
    FOREIGN KEY (Editora_ID) REFERENCES Editoras(Editora_ID),
    FOREIGN KEY (ISBN) REFERENCES ManuaisEscolares(ISBN),
)

CREATE TABLE Vender(
    DataVender             SMALLDATETIME          NOT NULL,
    Quantidade_Vender       INT                   NOT NULL,
    Preco_Unidade          MONEY                  NOT NULL,
    Pais_ID                CHAR(5)                NOT NULL,
    ISBN                   BIGINT                 NOT NULL,
    Editora_ID             CHAR(5)                NOT NULL,

    CHECK (Preco_Unidade>0),
    CHECK (Quantidade_Vender>=0),
    PRIMARY KEY (Pais_ID,ISBN,Editora_ID,DataVender),
    FOREIGN KEY (Pais_ID) REFERENCES Países(Pais_ID),
    FOREIGN KEY (ISBN) REFERENCES ManuaisEscolares(ISBN),
    FOREIGN KEY (Editora_ID) REFERENCES Editoras(Editora_ID),
)

```

IV. Criação do diagrama da base de dados



V. Inserção de dados nas tabelas

```

INSERT INTO Endereco (ID_Endereco, Endereco_Morada, Endereco_CodigoPostal, Endereco_Localidade)
VALUES
('10001', 'Rua A', '1234-567', 'Vila Real'),
('10002', 'Rua B', '2345-678', 'Porto'),
('10003', 'Rua C', '3456-789', 'Lisboa'),
('10004', 'Rua D', '4567-890', 'Braga'),
('10005', 'Rua E', '5678-901', 'Bragança'),
('10006', 'Rua F', '6789-012', 'Coimbra');
    
```

```

INSERT INTO Pessoas (CC, Nome, Data_Nascimento, ID_Endereco, Telefone)
VALUES
('12345678-1AB2', 'Luis Martelo', '1985-01-07', '10001', 912345678),
('23456789-2BC3', 'Sergio Pereira', '1990-02-02', '10002', 923456789),
('34567890-3CD4', 'Carlos Sinais', '2000-03-03', '10003', 934567890),
('45678901-4DC5', 'Jorge Silva', '1998-04-10', '10004', 945678901),
('56789012-5EF6', 'Cristiano Corneta', '1973-11-16', '10005', 967362945),
('67890123-7FG8', 'Frederico Vassoura', '1968-05-28', '10006', 927302028);
    
```

```

INSERT INTO TiposdeFabricacao (Fabricacao_ID, NomeFabricacao, Descricao)
VALUES
('20001', 'Tipo A', 'Descricao A'),
('20002', 'Tipo B', 'Descricao B'),
('20003', 'Tipo C', 'Descricao C');
    
```

```
INSERT INTO ManuaisEscolares (ISBN, Titulo, AnoEscolar, Edicao, Fabricacao_ID)
VALUES
```

```
(9781234567890, 'Matemática A', 5, 3, '20001'),
(9782345678901, 'História', 8, 2, '20002'),
(9783456789012, 'Inglês', 3, 1, '20003'),
(9784567890123, 'Português', 2, 4, '20003'),
(9785678901234, 'Ciências', 7, 1, '20001');
```

```
INSERT INTO Erratas (Errata_ID, Texto)
VALUES
```

```
('30001', 'Erro na página 10, corrigido no capítulo 2.'),
('30002', 'Erro na página 25, corrigido no capítulo 5.'),
('30003', 'Erro na página 30'),
('30004', 'Erro na página 40');
```

```
INSERT INTO ManuaisEscolares_Erratas (ISBN, Errata_ID)
VALUES
```

```
(9781234567890, '30001'),
(9782345678901, '30002'),
(9783456789012, '30003'),
(9784567890123, '30004');
```

```
INSERT INTO Editoras (Editora_ID, Nome_Editora, Cidade_Editora, TotVendas)
VALUES
```

```
('40001', 'Arial', 'Porto', 1000),
('40002', 'Porto Editora', 'Lisboa', 2000),
('40003', 'LeYa', 'Faro', 1500),
('40004', 'Planeta', 'Lisboa', 0);
```

```
INSERT INTO Países (Pais_ID, NomePais, Criador)
VALUES
```

```
('50001', 'Portugal', 1),
('50002', 'Espanha', 1),
('50003', 'França', 0),
('50004', 'Alemanha', 0),
('50005', 'Cuba', 1),
('50006', 'Inglaterra', 0);
```

```
INSERT INTO Viver (DataInicio, DataFim, Pais_ID, CC)
VALUES
```

```
('2020-01-01', '2021-01-01', '50001', '12345678-1AB2'),
('2021-02-01', '2022-02-01', '50002', '23456789-2BC3'),
('2021-01-01', '2021-12-31', '50003', '34567890-3CD4'),
('2021-01-02', '2021-12-30', '50004', '45678901-4DC5'),
('2020-03-27', '2022-12-31', '50005', '56789012-5EF6'),
('2020-12-01', '2021-10-20', '50006', '67890123-7FG8');
```

```
INSERT INTO Classificar (CC, ISBN, Fabricacao_ID)
VALUES
```

```
('12345678-1AB2', 9781234567890, '20001'),
('23456789-2BC3', 9782345678901, '20002'),
('34567890-3CD4', 9783456789012, '20003'),
('45678901-4DC5', 9784567890123, '20003');
```

```
INSERT INTO Formacao (DataFormacao, PrecoFormacao, ISBN, CC, ID_Formacao)
VALUES
```

```
('2022-05-01', 150.00, 9781234567890, '56789012-5EF6', '60001'),
('2022-06-01', 200.00, 9782345678901, '56789012-5EF6', '60002'),
('2023-01-01', 100.00, 9783456789012, '56789012-5EF6', '60003'),
('2023-01-02', 300.00, 9784567890123, '56789012-5EF6', '60004'),
('2022-07-01', 250.00, 9781234567890, '67890123-7FG8', '60005'),
('2023-02-01', 300.00, 9782345678901, '67890123-7FG8', '60006'),
('2022-08-01', 200.00, 9783456789012, '67890123-7FG8', '60007'),
('2022-08-02', 200.00, 9784567890123, '67890123-7FG8', '60008');
```



```
INSERT INTO Formandos (CC, Idade, ID_Formacao)
VALUES
```

```
( '12345678-1AB2', 44, '60001' ),
( '12345678-1AB2', 44, '60004' ),
( '12345678-1AB2', 44, '60003' ),
( '23456789-2BC3', 34, '60002' ),
( '23456789-2BC3', 34, '60006' ),
( '23456789-2BC3', 34, '60007' ),
( '34567890-3CD4', 24, '60001' ),
( '34567890-3CD4', 24, '60006' ),
( '34567890-3CD4', 24, '60002' ),
( '45678901-4DC5', 22, '60003' ),
( '45678901-4DC5', 22, '60005' ),
( '45678901-4DC5', 22, '60006' );
```

```
INSERT INTO Formadores (CC, Nivel)
VALUES
```

```
( '56789012-5EF6', 6 ),
( '67890123-7FG8', 8 );
```

```
INSERT INTO Produzir (DataProduzir, CustoUnidade, Quantidade_Produzir, Editora_ID, ISBN)
VALUES
```

```
( '2023-01-01', 10.00, 100, '40001', 9781234567890 ),
( '2023-02-01', 15.00, 150, '40002', 9782345678901 ),
( '2023-03-01', 30.00, 3000, '40003', 9783456789012 ),
( '2023-04-01', 30.00, 3000, '40004', 9784567890123 ),
( '2023-05-01', 25.00, 500, '40001', 9785678901234 );
```

```
INSERT INTO Vender (DataVender, Quantidade_Vender, Preco_Unidade, Pais_ID, ISBN, Editora_ID)
VALUES
```

```
( '2023-03-01', 50, 20.00, '50001', 9781234567890, '40001' ),
( '2023-04-01', 75, 25.00, '50002', 9782345678901, '40002' ),
( '2023-07-10', 1500, 35.00, '50003', 9783456789012, '40003' ),
( '2023-05-15', 2000, 40.00, '50004', 9784567890123, '40004' );
```

VI. Pesquisa de informação na base de dados

Qual foi o 1º manual vendido? [ISBN, Título, Data, País (nome)]

```
SELECT ME.ISBN, ME.Titulo, CONVERT(DATE, VE.DataVender) AS DataVender, P.NomePais
FROM ManuaisEscolares ME, Vender VE, Países P, Viver VI
WHERE ME.ISBN = VE.ISBN AND
      VE.Pais_ID = P.Pais_ID AND
      P.Pais_ID = VI.Pais_ID AND
      VE.DataVender = (SELECT MIN(DataVender)
                      FROM Vender)
```

Resultado:

	ISBN	Título	DataVender	NomePais
1	9781234567890	Matemática A	2023-03-01	Portugal

Quantos manuais já foram produzidos por cada editora? [Editora (nome, cidade), N_Manuais]

```
SELECT E.Nome_Editora, E.Cidade_Editora,
       (SELECT SUM(Quantidade_Produzir)
        FROM Produzir P
        WHERE P.Editora_ID = E.Editora_ID) AS N_Manuais
FROM Editoras E;
```

Resultado:

	Nome_Editora	Cidade_Editora	N_Manuais
1	Arial	Porto	600
2	Porto Editora	Lisboa	150
3	LeYa	Faro	3000
4	Planeta	Lisboa	3000

Que formadores vivem em países criadores? [Pessoas (nome, telefone)]

```
SELECT DISTINCT P.Nome, P.Telefone
FROM Pessoas P, Formadores F, Viver V, Países PA
WHERE P.CC = F.CC AND
      P.CC = V.CC AND
      V.Pais_ID = PA.Pais_ID AND
      PA.Criador = 1;
```

Resultado:

	Nome	Telefone
1	Cristiano Corneta	967362945

Que formandos participaram em mais formações nos últimos 2 anos? [nome, Total Formações]

```
SELECT TOP 1
    P.Nome, (SELECT COUNT(DISTINCT F.ID_Formacao)
             FROM Formandos F
             WHERE F.CC = P.CC AND F.ID_Formacao IN
                 (SELECT ID_Formacao
                  FROM Formacao
                  WHERE DataFormacao >= DATEADD(YEAR, -2, GETDATE()))) AS TotalFormacoes
FROM
    Pessoas P
WHERE
    EXISTS (SELECT 1
            FROM Formandos F
            WHERE F.CC = P.CC)
ORDER BY
    TotalFormacoes DESC;
```

Resultado:

Results Messages		
	Nome	TotalFormacoes
1	Jorge Silva	3

Que manuais nunca foram vendidos? [título, editora (nome)]. Apresente-os ordenados pela editora e pelo título.

```
SELECT ME.Titulo, E.Nome_Editora
FROM ManuaisEscolares ME, Produzir P, Editoras E
WHERE ME.ISBN = P.ISBN AND
      P.Editora_ID = E.Editora_ID AND
      ME.ISBN NOT IN (SELECT ISBN FROM Vender)
ORDER BY E.Nome_Editora, ME.Titulo;
```

Resultado:

Results Messages		
	Titulo	Nome_Editora
1	Ciências	Arial

VII. Stored Procedures

Crie um procedimento que apresente quantos formandos cada formador formou no último ano. O procedimento deve retornar o valor total pago por esses clientes para essas formações.

```
CREATE PROCEDURE ContFormandosUltimoAno
AS
BEGIN
    DECLARE @primeiro_dia_ultimo_ano DATE = DATEADD(YEAR, -1, DATEADD(DAY, 1,
        CAST(FORMAT(GETDATE(), 'yyyy-01-01') AS DATE)))
    DECLARE @ultimo_dia_ultimo_ano DATE = DATEADD(DAY, -1, CAST(FORMAT(GETDATE(), 'yyyy-01-01')
        AS DATE))

    SELECT
        F.CC AS Formador_CC,
        (SELECT Nome
         FROM Pessoas
         WHERE CC = F.CC) AS NomeFormador,
        (SELECT COUNT(DISTINCT CC)
         FROM Formandos
         WHERE ID_Formacao IN (SELECT ID_Formacao
                               FROM Formacao
                               WHERE CC = F.CC AND
                                   DataFormacao BETWEEN @primeiro_dia_ultimo_ano AND
                                                           @ultimo_dia_ultimo_ano)) AS TotalFormandosFormacao,
        (SELECT SUM(PrecoFormacao)
         FROM Formacao
         WHERE CC = F.CC AND
             DataFormacao BETWEEN @primeiro_dia_ultimo_ano AND
                                   @ultimo_dia_ultimo_ano) AS ValorTotalPago
    FROM
        Formadores F
    WHERE
        EXISTS (SELECT 1
                FROM Formacao
                WHERE CC = F.CC AND
                    DataFormacao BETWEEN @primeiro_dia_ultimo_ano AND
                                          @ultimo_dia_ultimo_ano)

    ORDER BY
        TotalFormandosFormacao DESC, ValorTotalPago DESC;
END

-- comando para testar o Stored Procedure
EXECUTE ContFormandosUltimoAno
```

Resultado:


Results		Messages		
	Formador_CC	NomeFormador	TotalFormandosFormacao	ValorTotalPago
1	67890123-7FG8	Frederico Vassoura	3	300,00
2	56789012-5EF6	Cristiano Corneta	1	300,00

VIII. Triggers

Assumindo que uma formação dura 7 dias, crie um trigger que apenas deixe inserir um novo registo no relacionamento Formação se o formando não estiver noutra formação.

```
CREATE TRIGGER ImpedirFormacaoConcorrente
ON Formacao
INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS (SELECT * FROM inserted)
    BEGIN
        IF EXISTS (SELECT 1
                     FROM Formacao F, inserted I
                     WHERE F.CC = I.CC AND
                           ABS(DATEDIFF(DAY, F.DataFormacao, I.DataFormacao)) < 7)
        BEGIN
            RAISERROR ('O formando já está em outra formação nos próximos 7 dias.', 16, 1)
            ROLLBACK TRANSACTION;
        END
    ELSE
    BEGIN
        INSERT INTO Formacao (DataFormacao, PrecoFormacao, ISBN, CC, ID_Formacao)
        SELECT DataFormacao, PrecoFormacao, ISBN, CC, ID_Formacao
        FROM inserted;
    END
END
END
-- executar a inserção de dados na tabela formação para testar o trigger!
```

Resultado:

 Messages

```
Msg 50000, Level 16, State 1, Procedure ImpedirFormacaoConcorrente, Line 17 [Batch Start Line 272]
O formando já está em outra formação nos próximos 7 dias.
Msg 3609, Level 16, State 1, Line 273
The transaction ended in the trigger. The batch has been aborted.

Completion time: 2024-06-02T13:32:26.1517586+01:00
```

4. Conclusão

Após a elaboração deste trabalho experimental foi possível compreender a complexidade e a importância de um sistema de gestão de vendas de manuais escolares eficiente. Através da análise do modelo E-R e do subsequente mapeamento para um modelo relacional, bem como a normalização até à Terceira Forma Normal e a implementação do modelo físico da base de dados, tornou-se evidente a necessidade de uma abordagem estruturada e ponderada na conceção deste tipo de sistemas.

Além disso, durante o processo de desenvolvimento do projeto, foram identificados desafios significativos relacionados à integração de diferentes entidades e à garantia da consistência dos dados. A relação entre Pessoas, Livros e Formações, além de ser uma necessidade operacional, revelou-se uma componente essencial para a eficácia e eficiência do sistema. Nesse sentido, a aplicação de metodologias de modelagem de dados e a adoção de boas práticas de gestão de bases de dados desempenharam um papel fundamental na superação desses.

Com este trabalho, conseguimos aprimorar as nossas habilidades técnicas de interpretação e implementação de modelos da base de dados, bem como desenvolver uma visão mais abrangente das complexidades inerentes à gestão de informações e dados em contextos reais.