

# Trabalho Prático de Introdução a Banco de Dados - 2023.2 (2024)

Thiago Pereira Lopes Chaves

Instituto de Computação – Universidade Federal do Amazonas (UFAM)  
69080-900 – Manaus – AM – Brasil

`thiago.chaves@icomp.ufam.edu.br`

**Abstract.** *This report describes the process taken by Thiago Chaves while making a relational database solution for a fictional library. Here the following topics are detailed: the used tools and its purposes, and the developed models/queries/views/indexes.*

**Resumo.** *Este relatório descreve o processo tomado por Thiago Chaves ao criar uma solução de banco de dados relacional para um livraria fictícia. Nele são detalhados: as ferramentas utilizadas e as razões de serem usadas, e os modelos/consulta/views/índices desenvolvidos.*

## 1. Introdução e Objetivos

O objetivo deste relatório é retratar como foi feita uma das soluções de banco de dados relacional para o Problema da Livraria (apresentado anteriormente na disciplina de Introdução a Banco de Dados do curso de Engenharia de Software da Universidade Federal do Amazonas), apontando as ferramentas utilizadas, escolhas feitas e conclusões tomadas durante todo o processo. A solução foi desenvolvida para um trabalho chamado de “Trabalho Prático - (Terceira Nota Parcial IBD - 2023-2 (2024))”.

## 2. Ferramentas, Linguagens e SGBD

O banco de dados criado para solucionar o Problema da Livraria foi desenvolvido ao redor das seguintes ferramentas: um caderno, um *notebook*, a linguagem Python (para população automática do banco de dados), o software Docker e o SGBD MySQL – cujos papéis serão descritos a seguir.

Antes do início da criação do banco, um planejamento intuitivo foi elaborado: tendo em vista o documento de descrição do trabalho na tela do *notebook*, Thiago começou a fazer anotações no seu caderno para visualizar quais seriam as melhores alternativas quanto ao SGBD e à linguagem de população de dados de maneira ampla.

A linguagem MySQL foi selecionada pelo motivo de ser muito boa para iniciantes, por possuir suporte a Python e C++ (as principais linguagens de Thiago Chaves) e por possuir uma grande comunidade (fontes de informação) – segundo websites de tecnologia disponibilizados no final deste documento. Além disso, a outra opção disponibilizada pelo professor responsável era PostgreSQL, conhecido por ser mais complexo e por ter menos informações na Internet, o que levou MySQL a se tornar a escolha mais plausível.

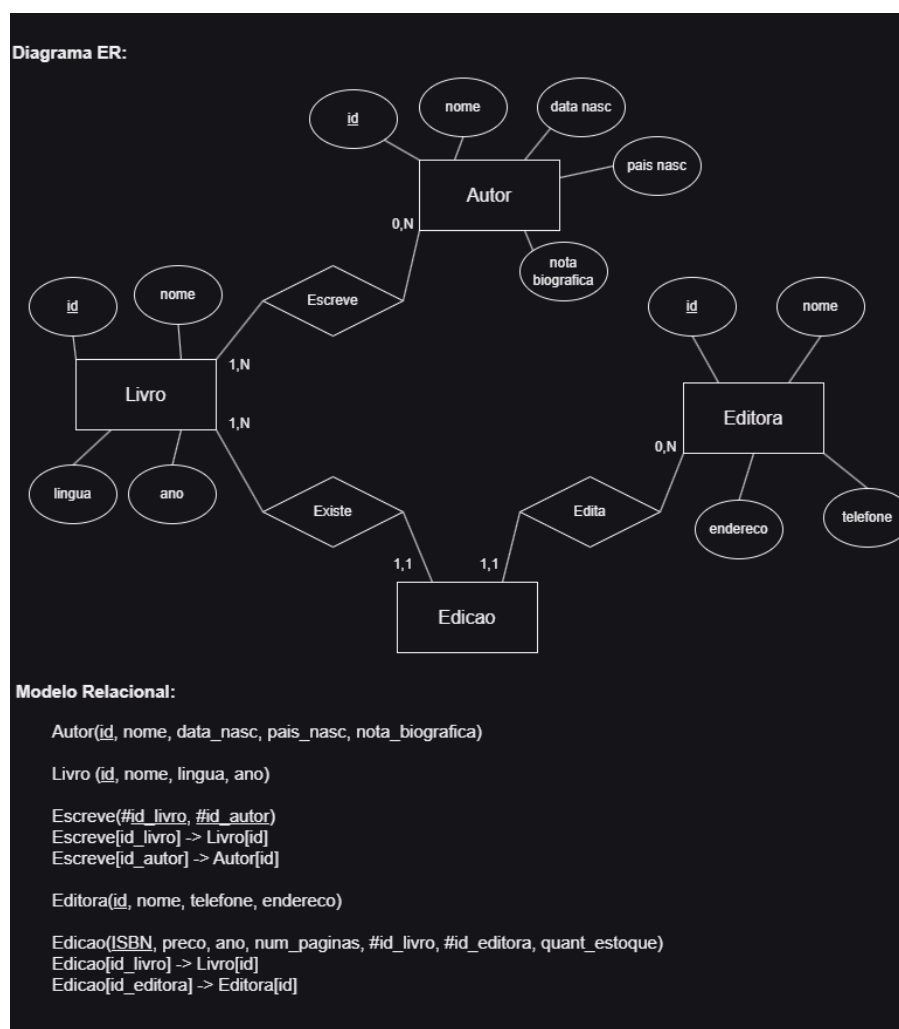
A linguagem Python foi escolhida para a população do banco de dados porque, além de também poder se conectar ao MySQL como C++, possui um tutorial bem

didático no website <https://www.mysqltutorial.org/> (recomendado de antemão pelo professor) sobre o assunto, o que pouparia tempo e que não afetaria qualidade da solução esperada.

Para que não houvesse a ocupação de muito espaço no *notebook* na instalação dos softwares MySQL (workbench, CLI e entre outros), foi-se escolhido o Docker como ambiente para a criação do banco de dados, através de um contêiner. Por meio do Docker, pude usar todos os comandos básicos que o MySQL oferece através de apenas um simples terminal.

Em conclusão, todas essas ferramentas – mais a seleção dos links descritos nas Referências Bibliográficas deste relatório – foram essenciais para solucionar o Problema da Livraria.

### 3. Esquemas e Consultas



**Figura 1. Modelos utilizados para a construção do banco de dados da livraria.**

Para a confecção ideal do banco de dados do Problema da Livraria, foram desenvolvidos o diagrama do modelo entidade-relacionamento com base no enunciado e, em seguida, o esquema relacional respectivo (Figura 1) no app de modelagem Diagrams Draw.io, os

quais permitem uma visão mais exata de quais deverão ser as tabelas e os atributos necessários. Não se tem muito para comentar sobre os modelos ER e relacional descritos por Thiago a não ser pelo fato de eles serem bem simples mas efetivos, levando a questão a ser resolvida.

Completos esses artefatos, Thiago logo começa a construção das tabelas e atributos descobertos via MySQL, o que não foi muito difícil por conta do apoio fornecido pelo professor responsável pelo trabalho e do site <https://www.mysqltutorial.org/>. A parte complicada de verdade estava por vir: a população do banco de dados, pois além de nunca ter feito isso antes, Thiago deveria ter cuidado ao respeitar as cardinalidades (chaves estrangeiras) de cada tabela, o que lhe dava muito medo de falhar, pois cada *while*, *insert* ou *caractere* guardava um perigo por trás (ainda bem que isso não o impediu de terminar e de entregar a qualidade pela qual preza).

Feita a população devida do banco de dados da livraria – o qual chamou de *livraria\_baudelaire* –, foram realizadas as consultas pedidas pelo professor (as quais só não foram tão complicadas de se fazer graças a um vídeo de tutorial aos comandos MySQL nomeado de *MySQL Tutorial for Beginners [Full Course]*, do canal *Programming with Mosh*). Por conta da escrita em uma única linha dos comandos MySQL via terminal, cada uma das consultas foram escritas e formatadas previamente no Bloco de Notas Windows, para só depois serem copiadas, coladas e executadas – o que foi maçante, mas também facilitador.

### 3.1. Consultas:

Seguem as consultas feitas para cada uma das respectivas questões (cada consulta foi feita levando em conta o enunciado das questões e as coisas ele ou o documento descrição do trabalho não limitavam objetivamente):

Obs 1.: Todas as entradas das questões abaixo foram consideradas como sendo os elementos que estão sendo comparados nos **WHERE**. Basta substituir os elementos que coloquei para conseguir um resultado diferente. No caso do *insert*, basta alterar a variável do *@idLivro*.

a. (SELECT) Listar os nomes de todos os autores que têm edições de seus livros publicados com uma determinada editora (id da editora dado como entrada).

R =

```
SELECT
    a.nome AS autor
FROM Autor a
LEFT JOIN Escreve e
    ON a.id = e.id_autor
INNER JOIN Livro l
    ON e.id_livro = l.id
```

```

INNER JOIN Edicao ed ON l.id = ed.id_livro
INNER JOIN Editora edt
    ON ed.id_editora = edt.id
WHERE edt.id = 4;

```

Comentário para o Professor: Segundo Thiago, o JOIN é mesmo uma ferramenta interessante, a qual pode ser confusa de início quando a utilizamos, porém isso depois passa. Poder unir tabelas com o simples chamar de um id, sem ter que criar uma tabela gigantesca, é mágico, de uma certa forma.

b. (SELECT) Dada uma palavra “XXX” dada como entrada, listar as informações das edições (número da edição, editora, título do livro e seu primeiro autor) que tenha a palavra dada no título do livro da edição.

R =

```

SELECT
    ed.ISBN AS numero_edicao,
    edt.nome AS editora,
    l.nome AS titulo_livro,
    (SELECT
        a.nome
    FROM Autor a
    INNER JOIN Escreve e
        ON a.id = e.id_autor
    WHERE e.id_livro = l.id
    LIMIT 1
    ) AS autor
FROM Edicao ed
INNER JOIN Editora edt
    ON ed.id_editora = edt.id
INNER JOIN Livro l
    ON ed.id_livro = l.id
WHERE l.nome REGEXP 'ABISMO'
ORDER BY l.nome;

```

Comentário para o Professor: Segundo Thiago, é muito conveniente e maravilhoso o poder que o MySQL possui sobre strings, sendo o melhor deles através do comando REGEXP, que permite o encontro de qualquer linha com atributo varchar utilizando-se apenas de uma string, sem a necessidade de uma lógica absurda

envolvendo o for loop. Ele não possui um conhecimento sólido sobre expressões regulares, mas gostaria muito de aprender.

c. (SELECT) Dada uma string “XXX” dada como entrada, listar as informações das edições (id das edições, editoras, títulos dos livros) onde a string fornecida esteja presente no nome de pelo menos um dos autores dos livros.

R =

```
SELECT
    ed.ISBN AS numero_edicao,
    edt.nome AS editora,
    l.nome AS titulo_livro
FROM Edicao ed
INNER JOIN Editora edt
    ON ed.id_editora = edt.id
INNER JOIN Livro l
    ON ed.id_livro = l.id
INNER JOIN Escreve e
    ON l.id = e.id_livro
INNER JOIN Autor a
    ON e.id_autor = a.id
WHERE a.nome REGEXP 'MAR'
ORDER BY l.nome;
```

Comentário para o Professor: Segundo Thiago, essa questão é bastante similar à questão b, por isso não tem muito para comentar sobre. Talvez apenas a estranheza que causa procurar por uma coisa no nome do autor e ainda assim não colocá-lo para imprimir na tela.

d. (UPDATE) Atualizar a quantidade de estoque de todas as edições de livros de uma editora dada como entrada - aumentando em 20%.

R =

```
UPDATE
    Edicao e
INNER JOIN Editora edt
    ON e.id_editora = edt.id
SET
    e.quant_estoque = e.quant_estoque * (120/100)
WHERE edt.id = 1;
```

Comentário para o Professor: Segundo Thiago, foi muito legal descobrir que dá pra adicionar uns  $x\%$  de alguma coisa  $y$  só usando a fórmula  $(x\% + 100\%) * y = \text{novo\_y}$  (ele normalmente fazia de outro jeito). De resto, não havia nada demais na questão: pega as edições de uma certa editora e aumenta o estoque, é isso.

e. (INSERT) Inserir uma nova edição de um livro que já existe, considerando que essa edição continua associada à editora anterior.

Obs 2.: Nesta questão, interpretei que deveria inserir uma nova edição desse livro com a mesma editora da última edição já inserida.

R =

```
SET @idLivro = 1;
INSERT INTO Edicao
(
    ISBN,
    preco,
    ano,
    num_paginas,
    id_livro,
    id_editora,
    quant_estoque
)
VALUES ('XXX-X-XXXXXX-XXX-X', 10, 1400, 100, @idLivro,
(
    SELECT
        edt.id
    FROM
        Editora edt
    INNER JOIN Edicao ed
        ON edt.id = ed.id_editora
    INNER JOIN Livro l
        ON ed.id_livro = l.id
    WHERE
        l.id = @idLivro
    ORDER BY
        edt.id
```

DESC

LIMIT 1

), 1);

Comentário para o Professor: Segundo Thiago, foi muito difícil de ler e de entender o enunciado dessa questão em particular. “Quem é a última edição já inserida? E em relação a quem?”, por exemplo, são perguntas que pairavam em sua cabeça. Por isso mesmo, ele resolveu colocar uma observação logo depois do enunciado, para que o professor entenda o que foi interpretado. A solução foi criar uma certa variável como entrada pro livro existente escolhido e, a partir dele, achar o id da última editora que fez uma edição desse livro.

### 3.2. VIEW

Quanto à *VIEW* designada no documento de descrição do trabalho, esta foi feita da seguinte forma (levando em consideração, por exemplo, que não foi mencionado quantos dos livros com as maiores quantidades de edição deveriam ser adicionados, resultando na escolha do comando “LIMIT 30”):

```
CREATE VIEW LivrosComMaisEdicoes
AS
SELECT
    edt.nome AS nome_editora,
    ed.ISBN AS id_edicao,
    l.nome AS nome_livro,
    ed.quant_estoque AS qtd_edicoes_em_estoque
FROM
    Livro l
INNER JOIN
    Edicao ed
ON
    l.id = ed.id_livro
INNER JOIN
    Editora edt
ON
    ed.id_editora = edt.id
ORDER BY
    ed.quant_estoque
DESC
```

LIMIT 30;

### 3.3. Índices

Por último, divulga-se abaixo os índices constituídos com base nas consultas feitas no banco de dados e suas motivações:

1. CREATE INDEX idx\_l\_nome ON Livro(nome);

Motivo/Comentário: A consulta na questão 'b' mostrou que será frequente o uso do nome do livro nas pesquisas, o que já era esperado para um banco de dados de livraria.

2. CREATE INDEX idx\_a\_nome ON Autor(nome);

Motivo/Comentário: A consulta na questão 'c' mostrou que será frequente o uso do nome do autor nas pesquisas, o que já era esperado para um banco de dados de livraria.

3. CREATE INDEX idx\_edt\_nome ON Editora(nome);

Motivo/Comentário: Levando em conta que os nomes de autores e livros podem ser frequentemente pesquisados na hora da busca de uma certa edição em uma livraria, é conveniente que também a pesquisa pelo nome de uma editora específica seja otimizada.

Esses foram todos os esquemas, consultas, views e índices feitos no banco de dados para a resolução para o Problema da Livraria.

Link para todos os arquivos utilizados nesse trabalho:  
[https://drive.google.com/drive/folders/1\\_hg8q9Vqb0kyi6GhcREAF-OhEaiO8i2h?usp=s\\_haring](https://drive.google.com/drive/folders/1_hg8q9Vqb0kyi6GhcREAF-OhEaiO8i2h?usp=s_haring).

### 4. Conclusões

Por mais simples que pareça ser o enunciado de um problema de banco de dados, não é nada fácil quando você é apenas um iniciante e aquele é o primeiro banco de dados que você está construindo. Foram necessários vários equipamentos, anotações, diagramas e tutoriais, consultas, tentativas e erros até que Thiago chegasse ao ponto de terminar com tudo o que tinha o trabalho prático da disciplina de Introdução a Banco de Dados.

Porém, o que mais importou de verdade durante todo esse relatório é que Thiago obteve conhecimento e experiência, ao mesmo tempo em que se organizava de maneira estruturada com as suas anotações e outras ferramentas auxiliares, buscando excelência e qualidade (o que torna apenas toda essa experiência ainda mais verdadeira).

### Referências Bibliográficas

<https://learnsql.com.br/blog/as-bases-de-dados-mais-populares-em-2023>

[https://www.guru99.com/pt/free-database-software.html?gpp&gpp\\_sid](https://www.guru99.com/pt/free-database-software.html?gpp&gpp_sid)

<https://www.treinaweb.com.br/blog/os-principais-sgbds-relacionais>

<https://www.datanumen.com/pt/Blogs/11-melhores-sistemas-de-gerenciamento-de-banco-de-dados-gratuitos/>



<https://www.mysqltutorial.org/>

<https://dev.mysql.com/doc/refman/8.0/en/>

<https://stackoverflow.com/questions/14684063/mysql-source-error-2>

<https://www.w3schools.com/mysql/>

<https://ss64.com/sql/syntax-field-sizes.html>

<https://stackoverflow.com/questions/20958/list-of-standard-lengths-for-database-fields>

<https://stackoverflow.com/questions/5452760/how-to-truncate-a-foreign-key-constrained-table>

<https://faker.readthedocs.io/en/master/#>

<https://sl.bing.net/jhn7ohOvGRE>

[https://www.youtube.com/watch?v=7S\\_tz1z\\_5bA&t=1175s](https://www.youtube.com/watch?v=7S_tz1z_5bA&t=1175s)

[https://docs.google.com/presentation/d/1A0PauqPLMQwQdz9\\_-BMRnPEgmkCBJfZ4bsUAhUGick/edit#slide=id.p](https://docs.google.com/presentation/d/1A0PauqPLMQwQdz9_-BMRnPEgmkCBJfZ4bsUAhUGick/edit#slide=id.p)

<https://stackoverflow.com/questions/34773555/exporting-data-from-mysql-docker-container>

<https://sl.bing.net/lbPkhijHeS>