

# PROJETO BANCO DE DADOS

*CODER*

## LIVRARIA

GUILHERME SALGADO  
JOÃO VITOR OLIVEIRA  
THIAGO ALMEIDA



# INTRODUÇÃO

ESTE TRABALHO DEMONSTRARÁ A APLICAÇÃO PRÁTICA DOS CONHECIMENTOS ADQUIRIDOS NO CURSO DE BANCO DE DADOS, MINISTRADO PELO PROFESSOR WILLIAN FRANÇA.

FIZEMOS O DESENVOLVIMENTO DE UM BANCO DE DADOS COMPLETO PARA UMA LIVRARIA, ABORDANDO TODAS AS FASES DO PROCESSO DE MODELAGEM E IMPLEMENTAÇÃO.

O PROJETO ABRANGERÁ DESDE A CRIAÇÃO DO DIAGRAMA ENTIDADE-RELACIONAMENTO (ER) ATÉ A UTILIZAÇÃO DE FERRAMENTAS SQL PARA MANIPULAÇÃO E GERAÇÃO DE RELATÓRIOS.

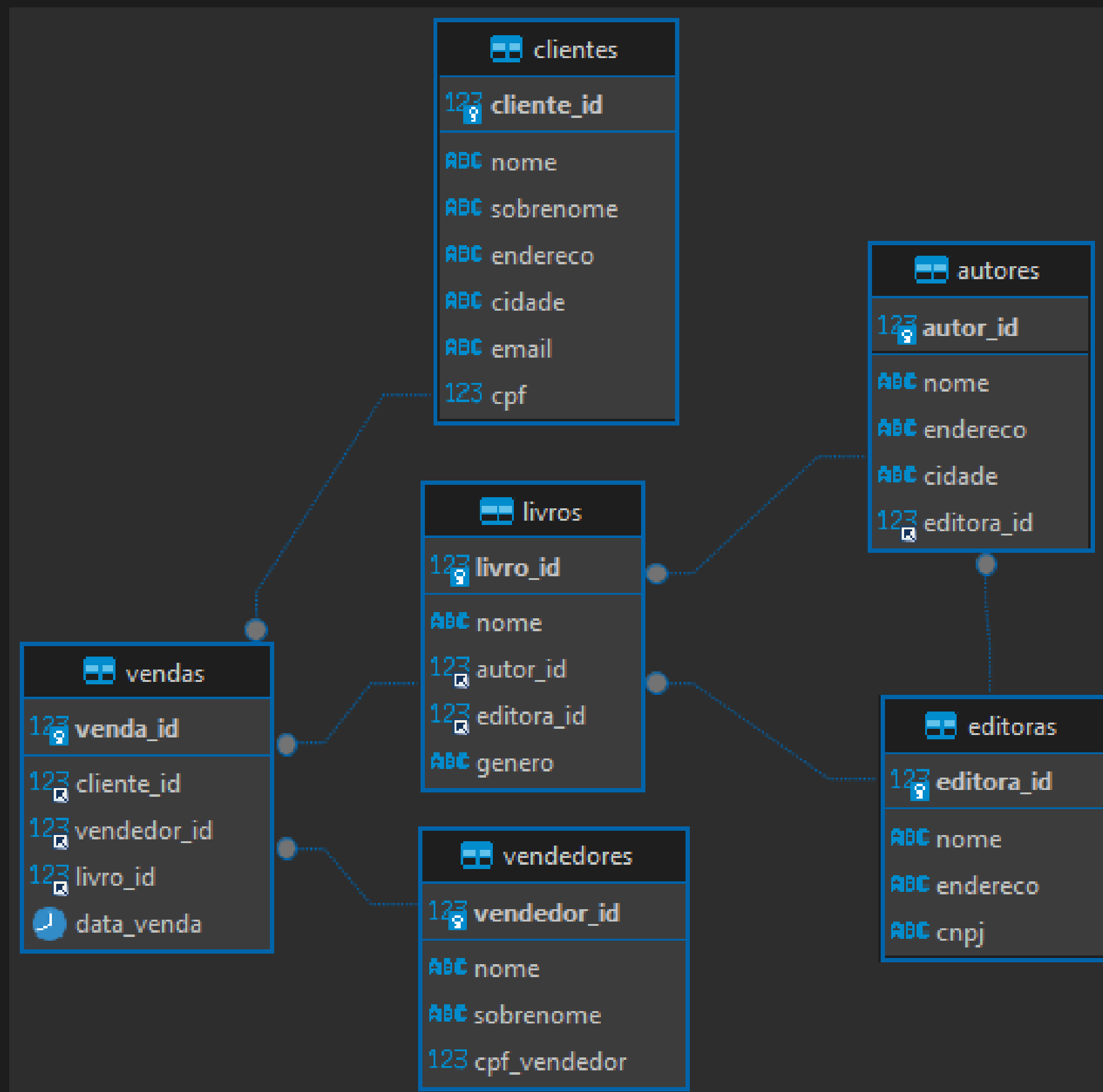
# OBJETIVO

- DESENVOLVER UM BANCO DE DADOS PARA UMA LIVRARIA.
- **MODELAGEM**: CRIAR UM DIAGRAMA ER QUE REFLITA FIELMENTE AS OPERAÇÕES DA LIVRARIA.
- **IMPLEMENTAÇÃO**: ESTRUTURAR TABELAS E RELACIONAMENTOS NO BANCO DE DADOS.
- **MANIPULAÇÃO**: UTILIZAR SQL PARA INSERÇÃO, CONSULTA E ATUALIZAÇÃO DE DADOS.
- **RELATÓRIOS**: GERAR RELATÓRIOS QUE OFEREÇAM INSIGHTS ESTRATÉGICOS SOBRE AS VENDAS, ESTOQUE E CLIENTES DA LIVRARIA.

# SITUAÇÃO-PROBLEMA

DESENVOLVER UM BANCO DE DADOS QUE CENTRALIZE AS INFORMAÇÕES DE ESTOQUE, VENDAS E CLIENTES, PERMITINDO UM CONTROLE MAIS EFICIENTE E A GERAÇÃO DE RELATÓRIOS DETALHADOS. ISSO AJUDARÁ A LIVRARIA A OTIMIZAR SUAS OPERAÇÕES, MELHORAR A PRECISÃO DOS DADOS E APOIAR A TOMADA DE DECISÕES ESTRATÉGICAS BASEADAS EM INFORMAÇÕES CONFIÁVEIS.

# DIAGRAMA ER



# TABELAS LIVRARIA

**CODER**

## 1. CLIENTES

Mostra os dados cadastrais dos clientes.

| clientes |            |
|----------|------------|
| 123      | cliente_id |
| ABC      | nome       |
| ABC      | sobrenome  |
| ABC      | endereço   |
| ABC      | cidade     |
| ABC      | email      |
| 123      | cpf        |

Identifica o cliente **INT PRIMARY KEY**  
Primeiro nome cliente **VARCHAR(50)**  
Sobrenome cliente **VARCHAR(50)**  
Endereço cliente **VARCHAR(200)**  
Cidade cliente **VARCHAR(50)**  
Email para contato **VARCHAR(100)**  
CPF **VARCHAR(14)**

## 2. AUTORES

Mostra os dados dos autores.

| autores |            |
|---------|------------|
| 123     | autor_id   |
| ABC     | nome       |
| ABC     | endereço   |
| ABC     | cidade     |
| 123     | editora_id |

Identifica o autor **INT PRIMARY KEY**  
Nome completo autor **VARCHAR(50)**  
Endereço autor **VARCHAR(200)**  
Cidade **VARCHAR(50)**  
Identifica a editora que o autor publicou o livro **INT FOREIGN KEY REFERENCES (EDITORA\_ID)**

# TABELAS LIVRARIA

**CODER**

## 3. EDITORAS

| editoras       |
|----------------|
| 123 editora_id |
| ABC nome       |
| ABC endereco   |
| ABC cnpj       |

Mostra os dados cadastrais das editoras

Identifica a editora **INT PRIMARY KEY**

Nome da editora **VARCHAR(75)**

Endereço editora **VARCHAR(200)**

CPF **VARCHAR(18)**

## 5. LIVROS

| livros         |
|----------------|
| 123 livro_id   |
| ABC nome       |
| 123 autor_id   |
| 123 editora_id |
| ABC genero     |

Mostra os dados cadastrais dos livros

Identifica o livro **INT PRIMARY KEY**

Nome do livro **VARCHAR(150)**

‘autor\_id’ : Identica o autor do livro  
**INT FOREIGN KEY REFERENCES (AUTHOR\_ID)**

‘editora\_id’ : Identica a editora do livro  
**INT FOREIGN KEY REFERENCES (EDITOR\_ID)**

‘genero’ : Gênero do livro  
**VARCHAR(75)**

## 4. VENDEDORES

| vendedores       |
|------------------|
| 123 vendedor_id  |
| ABC nome         |
| ABC sobrenome    |
| 123 cpf_vendedor |

Mostra os dados cadastrais dos vendedores

Identifica o vendedor **INT PRIMARY KEY**

Primeiro nome vendedor **VARCHAR(75)**

Sobrenome Vendedor **VARCHAR(200)**

CPF Vendedor **VARCHAR(18)**

# TABELAS LIVRARIA

## 6. VENDAS

| vendas          |
|-----------------|
| 123 venda_id    |
| 123 cliente_id  |
| 123 vendedor_id |
| 123 livro_id    |
| data_venda      |

Mostra os dados das vendas

Identifica a venda **INT PRIMARY KEY**

‘cliente\_id’ : Identica o cliente que comprou o livro **INT FOREIGN KEY REFERENCES (AUTOR\_ID)**

‘vendedor\_id’ : Identica o vendedor que fez a venda **INT FOREIGN KEY REFERENCES (VENDEDOR\_ID)**

‘livro\_id’ : Identica o livro da venda **INT FOREIGN KEY REFERENCES (LIVRO\_ID)**

‘Data venda’ : Mostra a data e a hora da venda **DATETIME**



# VIEWS

**CODER**

## 1. VIEW DOS AUTORES MAIS VENDIDOS

```
-- VIEW 1 -- RETORNA AUTORES MAIS VENDIDOS  
  
create or replace VIEW autor_mais_vendido AS  
SELECT a.nome, COUNT(v.venda_id) AS livros_vendidos  
FROM vendas v  
JOIN livros l ON v.livro_id = l.livro_id  
JOIN autores a ON l.autor_id = a.autor_id  
GROUP BY a.nome  
ORDER BY 2 desc;  
  
SELECT * FROM autor_mais_vendido;
```

## RESULTADO

|   | ABC nome ▼        | 123 livros_vendidos ▼ |
|---|-------------------|-----------------------|
| 1 | Bruno Carvalho    | 3                     |
| 2 | Antonio Gomes     | 3                     |
| 3 | Ricardo Fernandes | 2                     |
| 4 | Mariana Vieira    | 1                     |
| 5 | Luana Pereira     | 1                     |
| 6 | Camila Ribeiro    | 1                     |

# VIEWS

**CODER**

## 2. VIEW DAS CIDADES DOS CLIENTES

```
-- VIEW 2 -- RETORNA QUANTOS CLIENTES TEMOS POR CIDADE  
  
CREATE VIEW cidade_cliente AS  
SELECT cidade, COUNT(*) AS numero_clientes  
FROM clientes  
GROUP BY cidade;
```

## RESULTADO

|   | ABC cidade ▼   | 123 numero_clientes ▼ |
|---|----------------|-----------------------|
| 1 | Sao Paulo      | 1                     |
| 2 | Rio de Janeiro | 1                     |
| 3 | Belo Horizonte | 1                     |
| 4 | Porto Alegre   | 1                     |
| 5 | Manaus         | 2                     |
| 6 | Curitiba       | 2                     |


# VIEWS

**CODER**

## 3. VIEW EDITORA DOS AUTORES

```
-- VIEW 3 -- RETORNA QUAL A EDITORA DE CADA AUTOR  
  
CREATE VIEW editora_autor AS  
SELECT a.nome AS nome_autor, e.nome AS nome_editora  
FROM autores AS a  
JOIN editoras AS e ON a.editora_id = e.editora_id  
ORDER BY a.nome;
```


## RESULTADO

|   |  ABC nome_autor ▼ | ABC nome_editora ▼          |
|---|--|-----------------------------|
| 1 | Ana Costa  | Editora Paginas Magicas     |
| 2 | Andre Almeida  | Editora Fantasia            |
| 3 | Antonio Gomes  | Editora Encantos Literarios |
| 4 | Beatriz Nascimento   | Editora Paginas Magicas     |
| 5 | Bruno Carvalho   | Editora Fantasia            |
| 6 | Camila Ribeiro   | Editora Imaginario          |
| 7 | Carlos Silva   | Editora Iluminar            |

#### 4. VIEW EDITORA DE CADA LIVRO

```
-- VIEW 4 -- RETORNA QUAL A EDITORA DE CADA LIVRO  
  
CREATE VIEW editora_livro AS  
SELECT l.nome as nome_livro, e.nome as nome_editora  
FROM livros AS l  
JOIN editoras AS e ON l.editora_id = e.editora_id  
ORDER BY nome_livro;
```

#### RESULTADO

|   |  ABC nome_livro ▼ | ABC nome_editora ▼          |
|---|--|-----------------------------|
| 1 | 1984   | Editora Encantos Literarios |
| 2 | A Arte da Guerra   | Editora Paginas Magicas     |
| 3 | A Divina Comedia   | Editora Sussurros Poeticos  |
| 4 | A Ilha do Tesouro  | Editora Iluminar            |
| 5 | A Maquina do Tempo   | Editora Luar                |
| 6 | A Metamorfose  | Editora Paginas Magicas     |

# VIEWS

**CODER**

## 5. VIEW COMPRAS DOS CLIENTES

```
-- VIEW 5 -- RETORNA OS DADOS DE COMPRA DOS CLIENTES
CREATE VIEW compra_cliente AS
SELECT c.cliente_id, c.nome as nome_cliente, l.nome as nome_livro, v.venda_id, l.livro_id, v.data_venda
FROM vendas v
JOIN livros l ON v.livro_id = l.livro_id
JOIN clientes c using(cliente_id);
```

## RESULTADO

|   | 🔒 123 cliente_id ▼ | ABC nome_cliente ▼ | ABC nome_livro ▼             | 123 venda_id ▼ | 123 livro_id ▼ | 🕒 data_venda ▲ |
|---|--------------------|--------------------|------------------------------|----------------|----------------|----------------|
| 1 | 2                  | Carlos             | Alice no Pais das Maravilhas | 1              | 22.334         | 20%            |
| 2 | 4                  | Pedro              | As Cronicas de Narnia        | 2              | 45.678         | 20%            |
| 3 | 11                 | Carolina           | A Odisseia de Homero         | 3              | 33.446         | 20%            |
| 4 | 2                  | Carlos             | O Pequeno Principe           | 4              | 12.345         | 20%            |
| 5 | 4                  | Pedro              | Hamlet                       | 5              | 13.579         | 20% ▼          |
| < |                    |                    |                              |                |                | >              |


# FUNCTIONS

**CODER**

## RETORNA O NOME DO LIVRO PELO SEU ID

```
-- PRIMEIRA FUNÇÃO -- RETORNA O NOME DO LIVRO PELO NUMERO DO SEU ID.  
  
DELIMITER //  
  
CREATE FUNCTION retorna_nome_livro(id_livro INT)  
RETURNS VARCHAR(50)  
DETERMINISTIC  
BEGIN  
    DECLARE casca_vazia VARCHAR(50);  
    SELECT nome INTO casca_vazia FROM livros WHERE livro_id = id_livro;  
    RETURN casca_vazia;  
END  
//  
DELIMITER ;
```

## RESULTADO

|   |   |
|---|---|
|   |  retorna_nome_livro(11223) ▼ |
| 1 | Ulisses   |

# FUNCTIONS


**CODER**

## RETORNA O NÚMERO DE VENDAS POR EDITORA

```
DELIMITER //
```

```
CREATE FUNCTION vendas_por_editora(id_editora INT)  
RETURNS VARCHAR(255)  
DETERMINISTIC  
BEGIN  
    DECLARE total_vendas INT DEFAULT 0;  
    DECLARE nome_editora VARCHAR(100) DEFAULT '';  
    DECLARE resultado VARCHAR(255);  
    SELECT e.nome, COUNT(v.venda_id) INTO nome_editora, total_vendas  
    FROM vendas v  
    JOIN livros l ON v.livro_id = l.livro_id  
    JOIN editoras e ON l.editora_id = e.editora_id  
    WHERE e.editora_id = id_editora  
    GROUP BY e.nome;  
    SET resultado = CONCAT('Editora: ', nome_editora, ', Total Vendas: ', total_vendas);  
    RETURN resultado;  
END  
//  
DELIMITER ;
```

## RESULTADO

|   |   |  |
|---|---|--|
|   |  | ABC vendas_por_editora(28456) ▼            |
| 1 |   | Editora: Editora Iluminar, Total Vendas: 2 |

# STORED PROCEDURES

**CODER**

## ADICIONA DADOS A TABELA CLIENTES

```
DELIMITER //
```

```
CREATE PROCEDURE adicionar_cliente2(  
    IN clienteid INT,  
    IN nomecliente VARCHAR(50),  
    IN sobrenome VARCHAR(50),  
    IN endereco_cl VARCHAR(60),  
    IN cidade_cl VARCHAR(50),  
    IN email_cl VARCHAR(50),  
    IN cpf_cl BIGINT  
)  
BEGIN  
    INSERT INTO clientes(cliente_id, nome, sobrenome, endereco, cidade, email, cpf)  
    VALUES (clienteid, nomecliente, sobrenome, endereco_cl, cidade_cl, email_cl, cpf_cl);  
END;  
//  
DELIMITER ;
```

## CHAMANDO A FUNÇÃO

```
CALL adicionar_cliente2(36, 'Thiago', 'Almeida', 'Rua Atibaia', 'Santo Andre', 'thiagotgda@gmail.com', 12978195704) ;
```

## RESULTADO

|    |    |        |         |             |             |                      |                |
|----|----|--------|---------|-------------|-------------|----------------------|----------------|
| 36 | 36 | Thiago | Almeida | Rua Atibaia | Santo Andre | thiagotgda@gmail.com | 12.978.195.704 |
|----|----|--------|---------|-------------|-------------|----------------------|----------------|



# TRIGGERS

**CODER**

## CRIAÇÃO DE TABELA DE LOGS DE LIVROS

```
CREATE TABLE livros_log (  
    log_id INT AUTO_INCREMENT PRIMARY KEY,  
    livro_id INT,  
    action VARCHAR(10),  
    user VARCHAR(50),  
    action_date DATE,  
    action_time TIME,  
    details TEXT  
);
```

### 1. TRIGGER QUE ARMAZENA OS DADOS NA TABELA DE LOGS ANTES DA INSERÇÃO DE NOVOS LIVROS

```
DELIMITER $$  
  
CREATE TRIGGER before_livro_insert  
BEFORE INSERT ON livros  
FOR EACH ROW  
BEGIN  
    INSERT INTO livros_log (livro_id, action, user, action_date, action_time, details)  
    VALUES (NEW.livro_id, 'INSERT', CURRENT_USER(), CURDATE(), CURTIME(), 'Before inserting a new book.');
```

```
END;  
$$  
DELIMITER ;
```

# TRIGGERS

**CODER**

## CRIAÇÃO DE TABELA DE LOGS DE AUTORES

```
● CREATE TABLE autores_log (  
    log_id INT AUTO_INCREMENT PRIMARY KEY,  
    autor_id INT,  
    action VARCHAR(10),  
    user VARCHAR(50),  
    action_date DATE,  
    action_time TIME,  
    details TEXT  
);
```

## 2. TRIGGER QUE ARMAZENA OS DADOS NA TABELA DE LOGS DEPOIS DA INSERÇÃO DE NOVOS AUTORES

```
DELIMITER $$  
  
● CREATE TRIGGER after_livro_update  
AFTER UPDATE ON livros  
FOR EACH ROW  
BEGIN  
    INSERT INTO livros_log (livro_id, action, user, action_date, action_time, details)  
    VALUES (NEW.livro_id, 'UPDATE', CURRENT_USER(), CURDATE(), CURTIME(), 'After updating a book.');
```

```
END;  
  
DELIMITER ;
```