



*Serviço Nacional de Aprendizagem Industrial*

**PELO FUTURO DO TRABALHO**

# Modelagem de banco de dados relacional: Desvendando o SQL.

Rafael C. Ventura

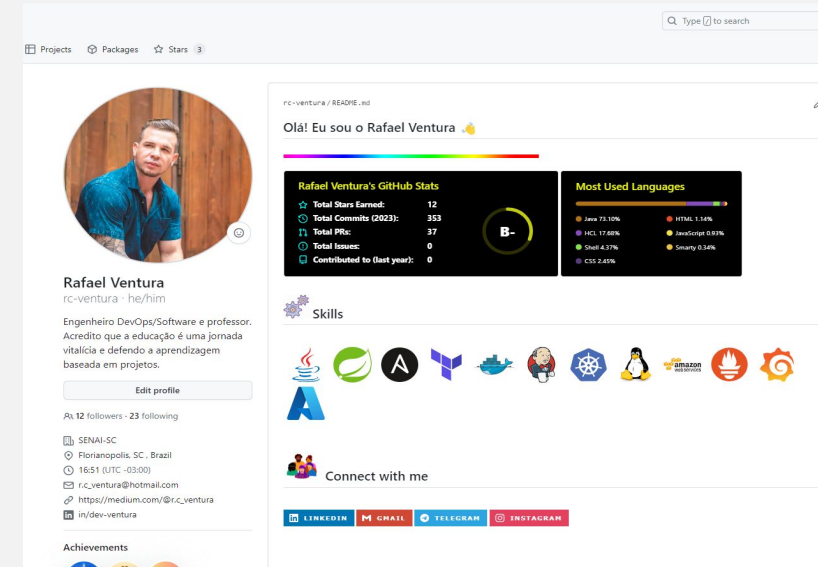
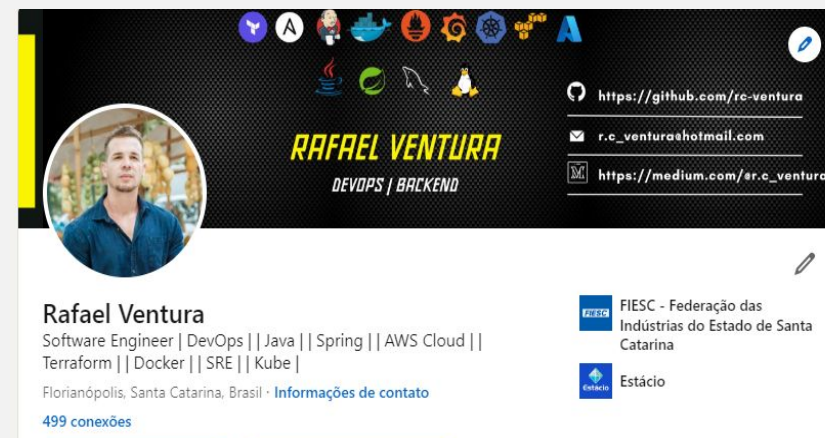
# Quem sou eu?

- Rafael C. Ventura
- Técnico em Segurança de Dados pela ANHANGUERA.
- Estudo Análise e Desenvolvimento de Sistemas na Estácio.
- Mestre em Educação pela PUC-RJ
- Leciono há mais de 10 anos.
- Experiência Cloud, IaC, DevOps
- email: rafael.ventura@edu.sc.senai.br




# Redes Sociais

- ❑ <https://www.linkedin.com/in/dev-ventura/>
- ❑ <https://github.com/rc-ventura>



# Entra 21

- ☐ Lógica com Java Script ( 4 encontros) 
- ☐ Banco de dados Relacionais (10 encontros)
- ☐ Metodologia Ágeis (1 encontro)
- ☐ Github (3 encontros)
- ☐ Programação Orientada à Objetos JS ( 10 encontros)
- ☐ React (15 encontros)
- ☐ Desenvolvimento do Trabalho de Conclusão do Curso ( 6 encontros)

# O que iremos aprender?

- Criar um banco de dados com o SGBD MySQL
- Manipular graficamente com a IDE MySQL Workbench
- Manipular em um sistema LINUX/POWERSHELL
- Compreender como é feito um banco de dados.
- Manipulação dos principais comandos SQL
- Uso do CREATE, ALTER, UPDATE, DROP, DELETE
- Apresentação do UNION
- Consultas personalizadas usando o DISTINCT
- Gerenciar chaves primárias e secundárias
- Criar registros em tabelas com INSERT INTO



# Nossas referências

□ **RelaX** (<https://dbis-uibk.github.io/relax/landing>)

□ **SCALER** (<https://www.scaler.com/topics/sql/sql-query-execution-order/>)

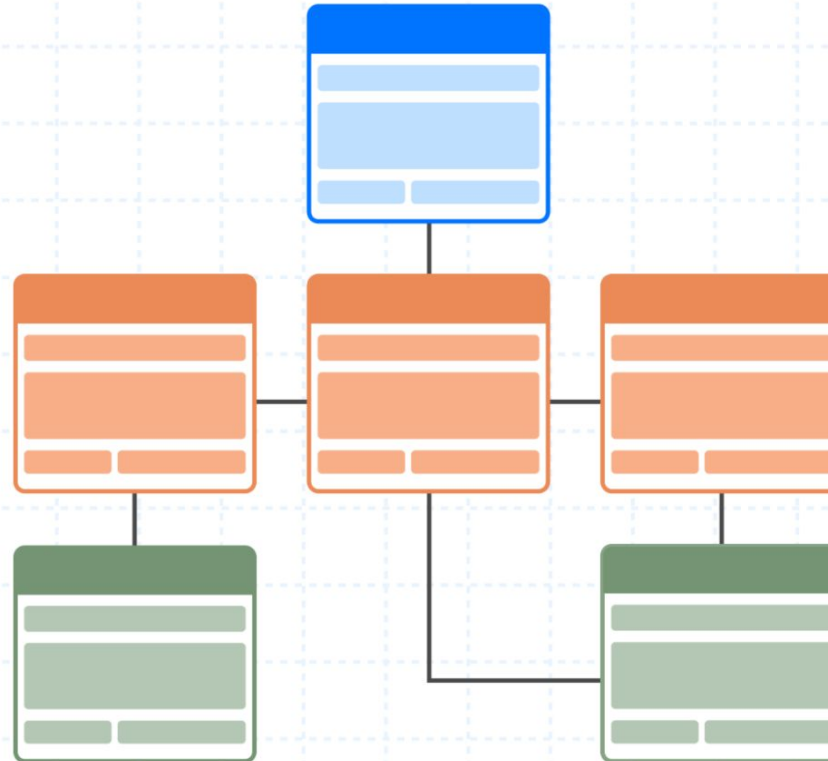
# Projeto Desafio: Banco de Dados Clube do Livro

□ Iremos criar um banco de dados do clube do livro do zero no MySQL.



# Desvendando o SQL(STRUCTURED QUERY LANGUAGE)

## Database schema



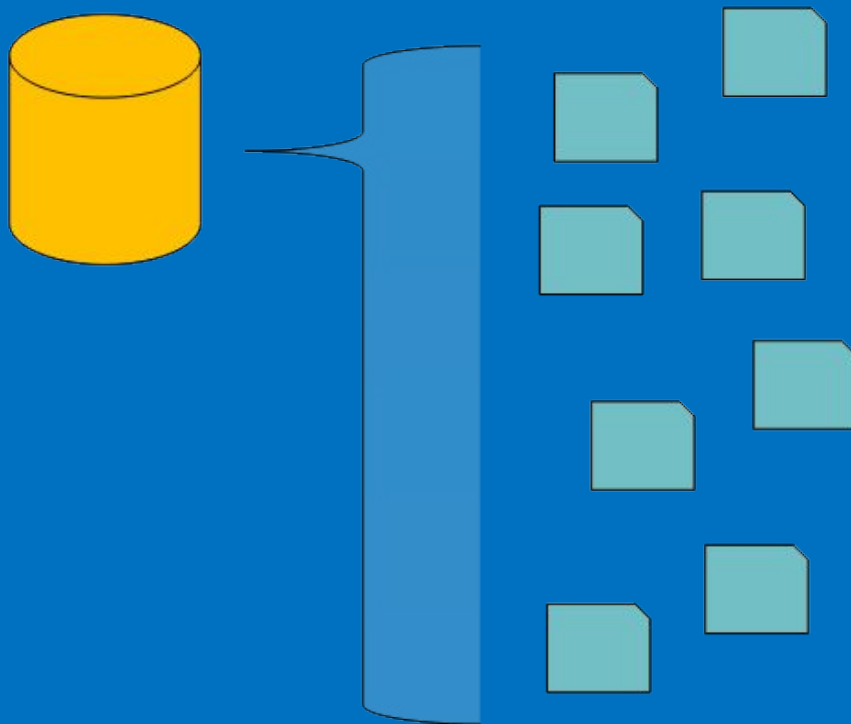
(STRUCTURED QUERY LANGUAGE)

(LINGUAGEM DE CONSULTA ESTRUTURADA)

# ÍNDICES (Index)



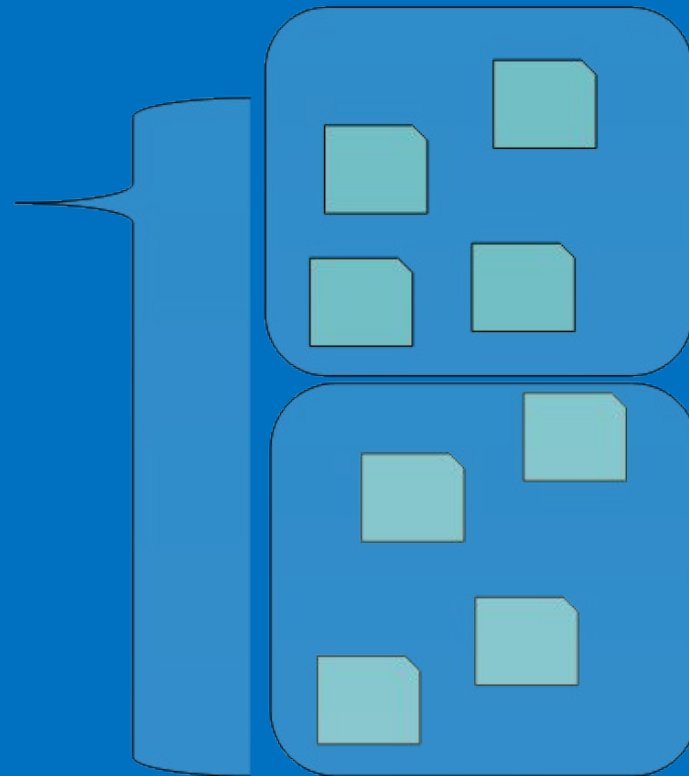
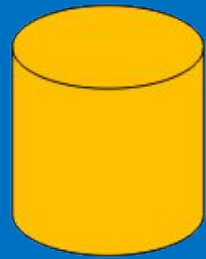
Introdução ao MYSQL



# ÍNDICES (Index)

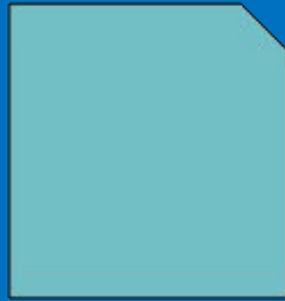


Introdução ao MYSQL

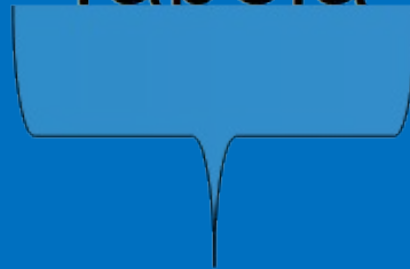


Esquemas

# ÍNDICES (Index)



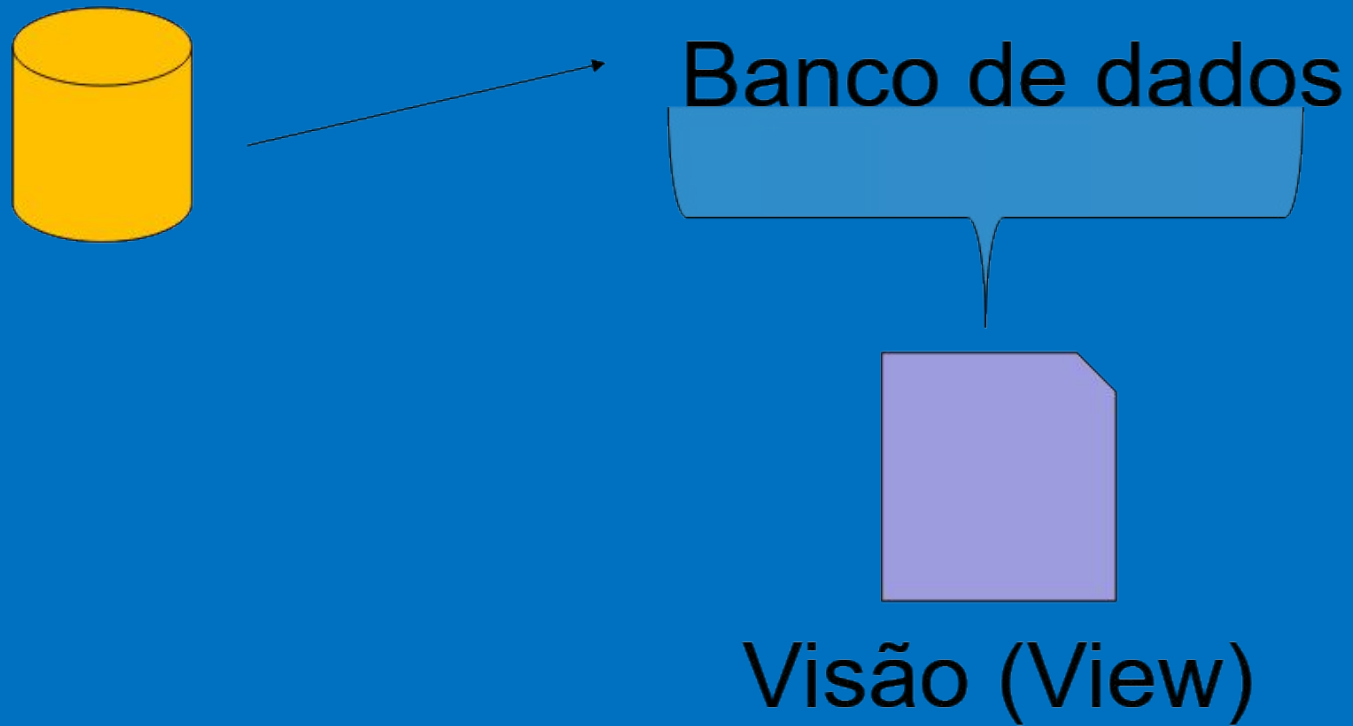
Tabela



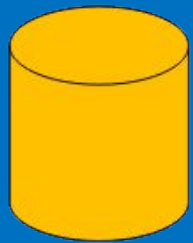
Índice

Índice					

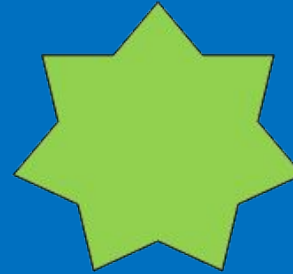
# VIEW (Vista)



# PROCEDURES (funções)



Banco de dados



Procedures

# PROCEDURES (funções)



Introdução ao MYSQL



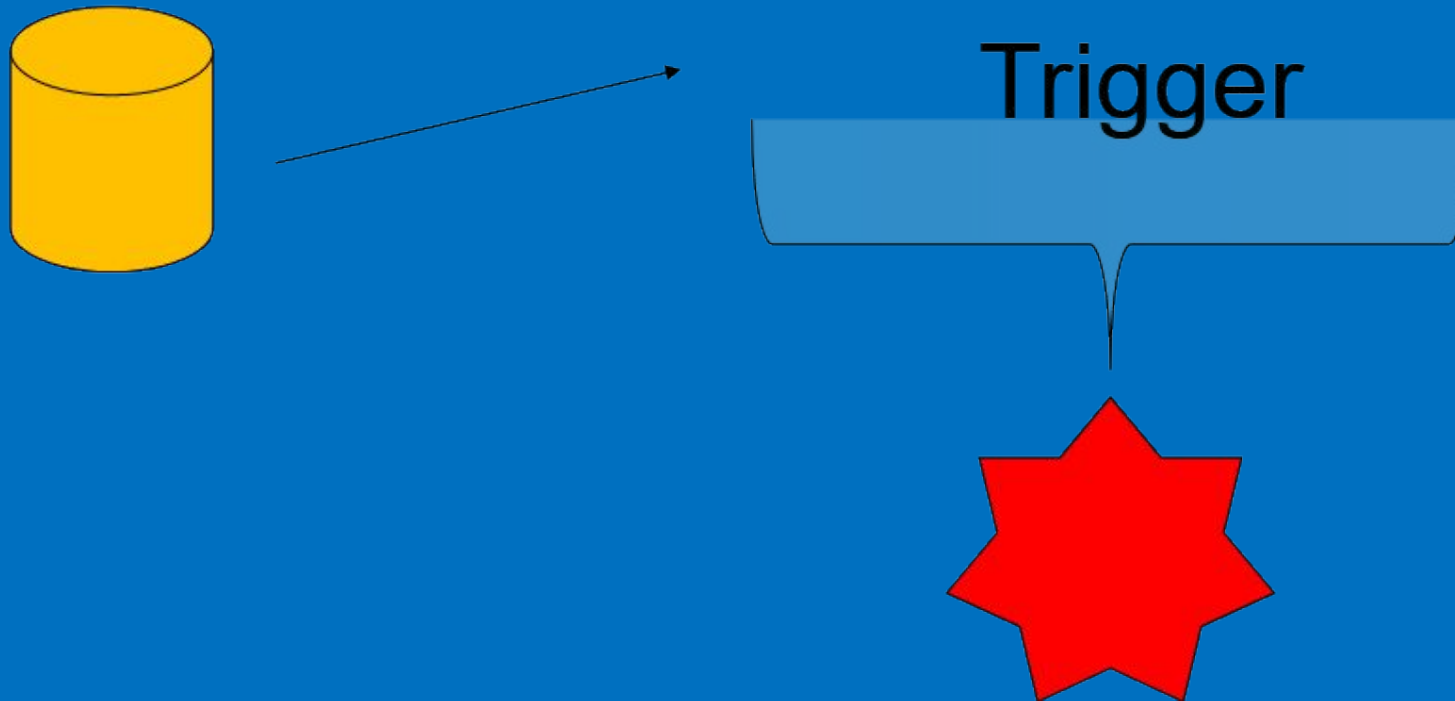
## Procedure e Funções



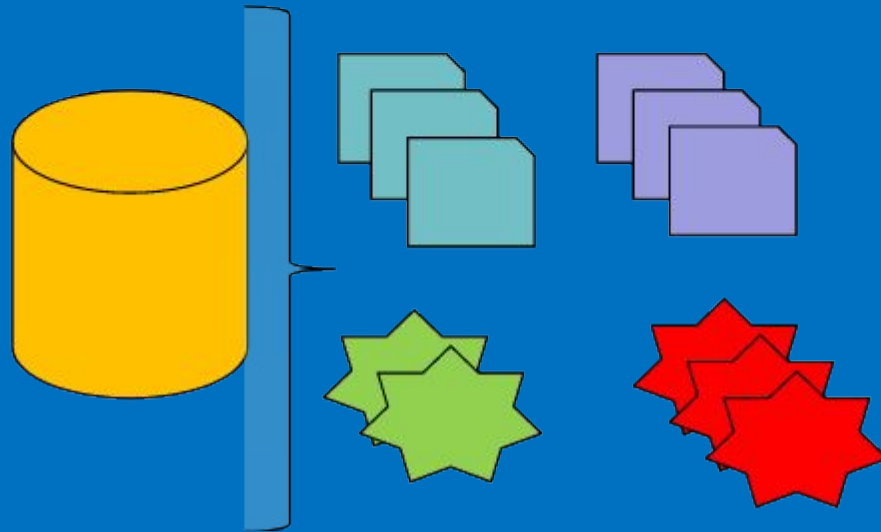
```
If a > 0 then  
X = x + y  
Z = instr(z+1)  
Select * from apha  
Put alpha in tableX
```

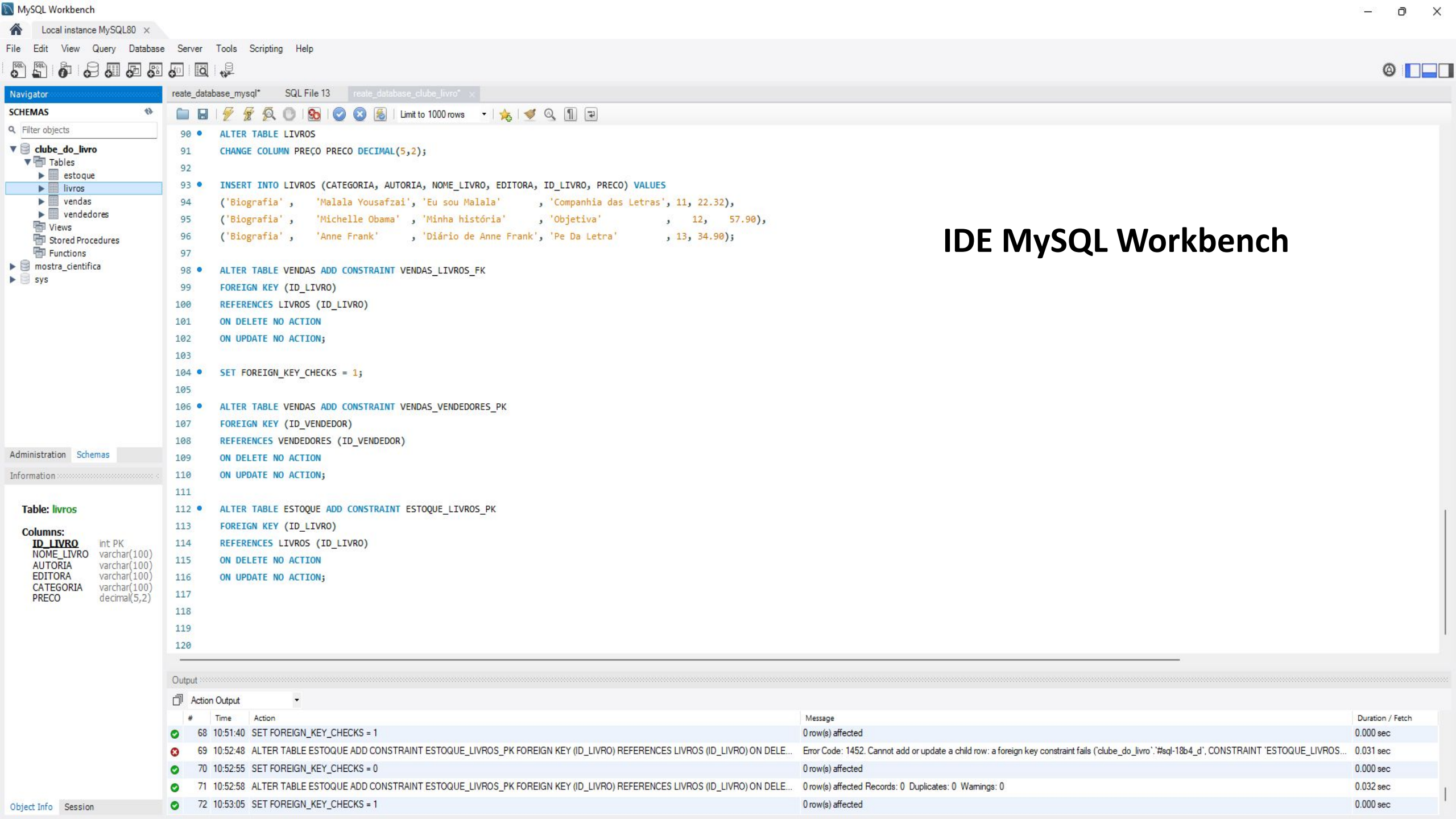


# TRIGGER (Gatilhos)

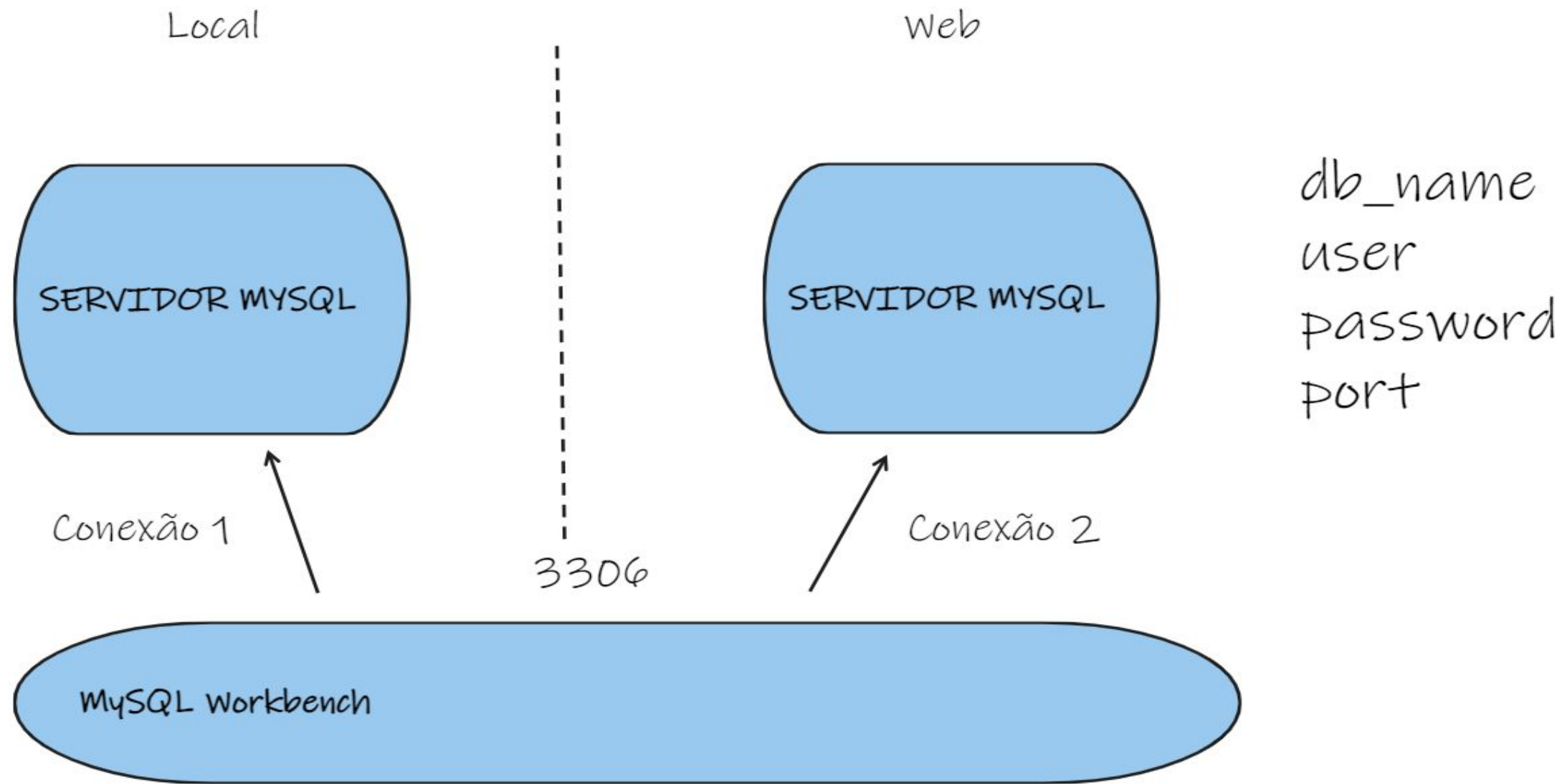


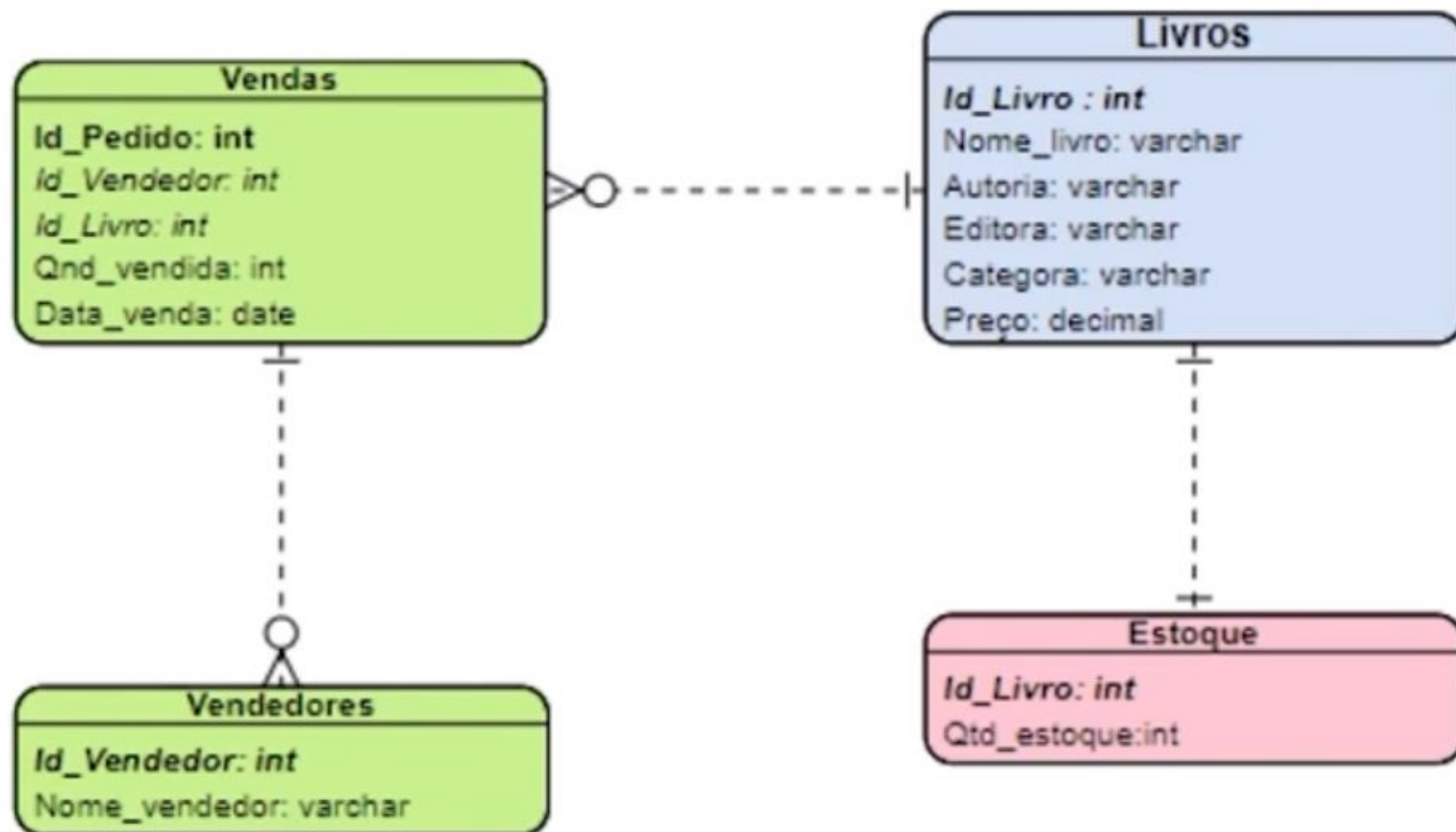
# Banco de Dados











# CRIANDO UM SCHEMA OU DATABASE?

- **DATABASE**= Entidade principal que contém várias coleções de dados, chamadas de esquemas.
  - Base de dados clube do livro ( possui tabelas, índices etc)
- **SCHEMA**= Organiza e estrutura os objetos dentro de um banco de dados.
  - Dentro da base de dados clube do livro podemos ter um schema que agrupa tabelas relacionadas a clientes, um schema que agrupa dados relacionados a editora.

# CRIANDO UM SCHEMA OU DATABASE?

- **Procedures** = Permite criar uma lógica baseada em funções, estruturas de repetição e condicionais.
  - Funções para conversão de dados, cálculos matemáticos etc
  
- **Triggers** = É um aviso programado que dispara a partir de um evento criado.
  - Sempre que alguém alterar uma tabela dispara um trigger de alerta.



# CRIANDO UM SCHEMA OU DATABASE

- **CREATE SCHEMA (nome\_do\_schema);**

- **CREATE DATABASE (nome\_da\_database);**

# SELECIONAR A DATABASE OU SCHEMA

- `USE (nome_da_database);`
- `USE (nome_do_schema);`

# CRIANDO UMA TABELA

- Referência a modelagem física do banco de dados (SQL POWER ARCHITECT)
- `CREATE TABLE (nome_da_tabela) (  
coluna1 Tipo_do_dado (numero ) AUTO-INCREMENT, NOT NULL  
coluna2 Tipo_do_dado ( numero) ,  
coluna3 Tipo_do_dado (numero ) NOT NULL,  
primary key (nome_da_coluna)  
);`

# TABELA LIVROS

```
CREATE TABLE LIVROS (  
    ID_LIVRO INT NOT NULL,  
    NOME_LIVRO VARCHAR(100) NOT NULL,  
    AUTORIA VARCHAR(100) NOT NULL,  
    EDITORA VARCHAR(100) NOT NULL,  
    CATEGORIA VARCHAR(100) NOT NULL,  
    PRECO DECIMAL(5,2) NOT NULL,  
    PRIMARY KEY (ID_LIVRO)  
);
```

# TABELA ESTOQUE

```
CREATE TABLE ESTOQUE (  
    ID_LIVRO INT NOT NULL,  
    QTD_ESTOQUE INT NOT NULL,  
    PRIMARY KEY (ID_LIVRO)  
);
```

# TABELA VENDAS

```
CREATE TABLE VENDAS (  
    ID_PEDIDO INT NOT NULL,  
    ID_VENDEDOR INT NOT NULL,  
    ID_LIVRO INT NOT NULL,  
    QTD_VENDIDA INT NOT NULL,  
    DATA_VENDA DATE NOT NULL,  
    PRIMARY KEY (ID_VENDEDOR, ID_PEDIDO)  
);
```

# TABELA VENDEDORES

```
CREATE TABLE VENDEDORES (  
    ID_VENDEDOR INT NOT NULL,  
    NOME_VENDEDOR VARCHAR(255) NOT NULL,  
    PRIMARY KEY (ID_VENDEDOR)  
);
```

# Criando os relacionamentos (chave estrangeira)

1) Relação entre as tabelas Vendas e Livros

```
ALTER TABLE VENDAS ADD CONSTRAINT VENDAS_LIVROS_FK  
FOREIGN KEY (ID_LIVRO)  
REFERENCES LIVROS (ID_LIVRO)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION;
```

- Altera a tabela adicionando uma restrição (nome\_chave\_estrangeira)
- Declara a coluna da chave estrangeira
- Faz referência a tabela origem da chave (coluna)
- Opcionalmente criando restrições para melhorar a integridade do meus dados, um erro será criado se eu atualizar ou deletar um registro na tabela filha.



# Criando relacionamentos (chave estrangeira)

2) Relação entre as tabelas Livros e Estoque

```
ALTER TABLE ESTOQUE ADD CONSTRAINT ESTOQUE_LIVROS_FK  
FOREIGN KEY (ID_LIVRO)  
REFERENCES LIVROS (ID_LIVRO)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION;
```

# CHAVE ESTRANGEIRA (Relacionamentos)

3) Relação entre as tabelas Vendedores e Vendas

```
ALTER TABLE VENDAS ADD CONSTRAINT VENDAS_VENDEDORES_FK  
FOREIGN KEY (ID_VENDEDOR)  
REFERENCES VENDEDORES (ID_VENDEDOR)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION;
```

# DESATIVANDO A CHAVE ESTRANGEIRA

- Para começarmos a inserir registros precisamos desativar a chave estrangeira.
- Como temos uma coluna na tabela que faz referência a outra tabela (chave estrangeira), teremos dificuldade em inserir os dados livremente pois agora temos colunas associadas em duas tabelas diferentes ( princípio da integridade de dados)
- `SET FOREIGN_KEY_CHECKS = 0;`

# INSERINDO REGISTROS EM UM TABELA

```
INSERT INTO (nome_data_tabela) VALUES (  
coluna1,  
coluna2,  
coluna3  
);
```

```
INSERT INTO (nome_data_tabela) (coluna1, coluna2, coluna3)  
VALUES  
(coluna1, coluna2, coluna3);
```

# INSERINDO REGISTROS EM UM TABELA

```
INSERT INTO LIVROS VALUES (  
    1,  
    'Percy Jackson e o Ladrão de Raios',  
    'Rick Riordan',  
    'Intrínseca',  
    'Aventura',  
    34.45  
);
```

# INSERINDO REGISTROS EM MÚLTIPLAS LINHAS

## LIVROS

### INSERT INTO LIVROS VALUES

```
(2, 'A Volta ao Mundo em 80 Dias', 'Júlio Verne', 'Principis', 'Aventura', 21.99),
(3, 'O Cortiço', 'Aluísio de Azevedo', 'Panda Books', 'Romance', 47.8),
(4, 'Dom Casmurro', 'Machado de Assis', 'Via Leitura', 'Romance', 19.90),
(5, 'Memórias Póstumas de Brás Cubas', 'Machado de Assis', 'Antofágica', 'Romance',
45),
(6, 'Quincas Borba', 'Machado de Assis', 'L&PM Editores', 'Romance', 48.5),
(7, 'Ícaro', 'Gabriel Pedrosa', 'Ateliê', 'Poesia', 36),
(8, 'Os Lusíadas', 'Luís Vaz de Camões', 'Montecristo', 'Poesia', 18.79),
(9, 'Outros Jeitos de Usar a Boca', 'Rupi Kaur', 'Planeta', 'Poesia', 34.8);
```

# INSERINDO REGISTROS EM MÚLTIPLAS LINHAS

## VENDEDORES

INSERT INTO VENDEDORES VALUES

```
(1,'Paula Rabelo'),  
(2,'Juliana Macedo'),  
(3,'Roberto Barros'),  
(4,'Barbara Jales');
```

# INSERINDO REGISTROS EM MÚLTIPLAS LINHAS

## VENDAS

INSERT INTO VENDAS VALUES

```
(1, 3, 7, 1, '2020-11-02'),  
(2, 4, 8, 2, '2020-11-02'),  
(3, 4, 4, 3, '2020-11-02'),  
(4, 1, 7, 1, '2020-11-03'),  
(5, 1, 6, 3, '2020-11-03'),  
(6, 1, 9, 2, '2020-11-04'),  
(7, 4, 1, 3, '2020-11-04'),  
(8, 1, 5, 2, '2020-11-05'),  
(9, 1, 2, 1, '2020-11-05'),  
(10, 3, 8, 2, '2020-11-11'),  
(11, 1, 1, 4, '2020-11-11'),  
(12, 2, 10, 10, '2020-11-11'),  
(13, 1, 12, 5, '2020-11-18'),  
(14, 2, 4, 1, '2020-11-25'),  
(15, 3, 13, 2, '2021-01-05'),  
(16, 4, 13, 1, '2021-01-05'),  
(17, 4, 4, 3, '2021-01-06'),  
(18, 2, 12, 2, '2021-01-06');
```



# INSERINDO REGISTROS EM MÚLTIPLAS LINHAS

## ESTOQUE

INSERT INTO ESTOQUE VALUES

(1, 7),  
(2, 10),  
(3, 2),  
(8, 4),  
(10, 5),  
(11, 3),  
(12, 3);

# INSERINDO REGISTROS FORA DE ORDEM

```
INSERT INTO LIVROS (CATEGORIA, AUTORIA, NOME_LIVRO, EDITORA, ID_LIVRO,
PRECO) VALUES
('Biografia' , 'Malala Yousafzai', 'Eu sou Malala' , 'Companhia das Letras', 11, 22.32),
('Biografia' , 'Michelle Obama' , 'Minha história' , 'Objetiva' , 12, 57.90),
('Biografia' , 'Anne Frank' , 'Diário de Anne Frank', 'Pe Da Letra' , 13, 34.90);
```

# EXCLUINDO UM BANCO DE DADOS

**DROP DATABASE (nome\_database);**

=> Crie uma database e depois exclua;

# EXCLUINDO UMA TABELA

**DROP TABLE (nome\_tabela);**

Lembrete: desativar a restrição da chave estrangeira

=> Crie uma tabela chamada vendedores2, uma cópia da tabela vendedores

=> Exclua a tabela vendedores2

# ADICIONANDO UMA COLUNA

```
ALTER TABLE (nome_tabela) ADD COLUMN (coluna_nova, Tipo_dado);
```

**=> Adicione a coluna ano de publicação como um INT na tabela livro.**

# SELEÇÃO DISTINTA (DISTINCT)

- ❖ Recupera registros únicos.
- ❖ O SGBD remove registros duplicados.

Quero fazer uma pesquisa para buscar vendas no site feita pelos vendedores de um livro específico (Id=4) pois esse livro tem uma promoção associada e queremos dar um bônus para os vendedores?

```
SELECT DISTINCT ID_VENDEDOR FROM VENDAS  
WHERE ID_LIVRO =4
```

# DELETAR REGISTROS

**DELETE FROM (nome\_tabela) WHERE (condição);**

**DELETE FROM (nome Tabela) WHERE (condição) between (condição);**

**O livro lusíadas não é mais vendido pelo clube do livro, a editora resolveu fechar um contrato com uma outra empresa de exclusividade.**

**DELETE o registro referente ao caso.**

# ATUALIZAR REGISTROS

**UPDATE (nome\_tabela) SET (coluna\_a\_ser\_alterada);**

**Este mês é black friday no clube do livro. Então todos os livros possuem um desconto de 10%. Então vamos agora fazer um update da coluna preços na tabela livro.**

**UPDATE LIVROS SET PREÇO = 0.9 \* PREÇO;**



# GROUP BY (SUM, COUNT, MAX, MIN, MEDIA)

Para funções de agregação podemos utilizar os comandos:

- **MAX:** a partir de um conjunto de valores é retornado o maior entre eles;
- **MIN:** analisa um grupo de valores e retorna o menor entre eles;
- **SUM:** calcula o somatório dos valores de um campo específico;
- **AVG:** realiza a média aritmética dos valores de uma determinada coluna;
- **COUNT:** contabiliza a quantidade de linhas selecionadas.

Nem sempre será necessário agrupar com Group By para usar essas cláusulas. Por exemplo Max e Min pode ser usado sem Group By

# GROUP BY (SUM, MAX, MIN, MEDIA)

O Clube do livro quer ranquear os seus vendedores. Criou uma política para dar benefícios àqueles que mais vendem livros. Então os 3 primeiros vendedores que acumularem mais vendas vão receber um acréscimo de 10% do salário.

```
SELECT VENDAS.ID_VENDEDOR, VENDEDORES.NOME_VENDEDOR,  
SUM(VENDAS.QTD_VENDIDA)  
FROM VENDAS INNER JOIN VENDEDORES  
ON VENDAS.ID_VENDEDOR = VENDEDORES.ID_VENDEDOR  
GROUP BY VENDAS.ID_VENDEDOR;  
ORDER BY SUM (VENDAS.QTD_VENDIDA) DESC  
LIMIT 3;
```

# GROUP BY (SUM, MAX, MIN, MEDIA)

- 1- Qual a maior venda do clube do livro ?
- 2- Qual vendedor vendeu mais ?
- 3- A quantidade de livros que cada vendedor vendeu?
- 4- Que livro foi mais vendido?
- 5- Quais livros não foram vendidos ?
- 6- Existe algum livro que teve vendas e não faz mais parte dos livros disponíveis para comercialização? (Null em id do livro)
- 7 - Qual a média de vendas entre os vendedores?
- 8 - Qual foi o total de todas as vendas feitas ?
- 9 - Qual o maior pedido em termos de valor monetário feito?
- 10 - Quais os livros estão com um exemplar no estoque?

# DIFERENÇA

```
SELECT VENDAS.ID_VENDEDOR, VENDEDORES.NOME_VENDEDOR, VENDAS.QTD_VENDIDA  
FROM VENDAS, VENDEDORES  
WHERE VENDAS.ID_VENDEDOR = VENDEDORES.ID_VENDEDOR
```

```
SELECT VENDAS.ID_VENDEDOR, VENDEDORES.NOME_VENDEDOR, VENDAS.QTD_VENDIDA  
FROM VENDAS INNER JOIN VENDEDORES  
ON VENDAS.ID_VENDEDOR = VENDEDORES.ID_VENDEDOR
```

# EXPLICAÇÕES EXTRAS (UNION e SUBCONSULTA

<https://drive.google.com/file/d/12MUEhE0Ba8BgeWix1JNVzMApby5stgO0/view?usp=sharing>



*Serviço Nacional de Aprendizagem Industrial*

**PELO FUTURO DO TRABALHO**

**0800 048 1212**     **sc.senai.br**

Rodovia Admar Gonzaga, 2765 - Itacorubi - 88034-001 - Florianópolis, SC