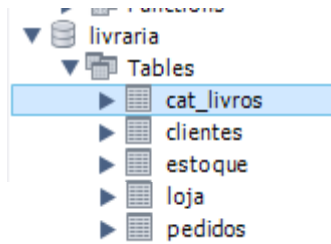


TEMÁTICA DO BANCO DE DADOS

O banco se trata de um sistema de vendas de uma rede de livrarias, registrando o cadastro dos clientes, o estoque, o cadastro das lojas da rede, cadastro dos livros bem como registrando todos os pedidos realizados.

TABELAS QUE COMPREENDEM O BANCO



cat_livros – cadastro contendo os livros classificados por gênero, nome, editora, ano de publicação e escritor;

Table: cat_livros						
Columns:						
Cod_Prod	int PK					
Gênero	varchar(50)					
Nome	varchar(200)					
Editora	varchar(50)					
Ano_Publi	decimal(4,0)					
Escritor	varchar(100)					

	Cod_Prod	Gênero	Nome	Editora	Ano_Publi	Escritor
▶	1	Fantasia	O Senhor dos Anéis - a Sociedade do Anel	Martins Fontes	2002	J. R. R. Tolkien
	2	Fantasia	O Senhor dos Anéis - as Duas Torres	Martins Fontes	2002	J. R. R. Tolkien
	3	Fantasia	O Senhor dos Anéis: o Retorno do Rei	Martins Fontes	2002	J. R. R. Tolkien
	4	Ficção Científica	Fundação	Aleph	2009	Isaac Asimov
	5	Bibliografia	Chatô o Rei do Brasil	Companhia das letras	1994	Fernando Morais
	6	Bibliografia	Steve Jobs	Companhia das letras	2011	Walter Isaacson
	7	História	Sapiens: uma Breve Historia da Humanidade	L&pm	2017	Yuval Noah Harari
	8	História	Homo Deus	L&pm	2018	Yuval Noah Harari
	9	História	21 lições para o século 21	L&pm	2019	Yuval Noah Harari

clientes – cadastro com informações pessoais e de endereço, dos clientes registrados na livraria;

Table: clientes										
Columns:										
Cod_CPF	varchar(11) PK									
Nome	varchar(100)									
Sobrenome	varchar(100)									
Telefone	decimal(11,0)									
End_Rua	varchar(100)									
End_Numero	decimal(5,0)									
End_Complet	varchar(100)									
End_Bairro	varchar(100)									
End_Cidade	varchar(100)									
End_UF	varchar(2)									

	Cod_CPF	Nome	Sobrenome	Telefone	End_Rua	End_Numero	End_Complet	End_Bairro	End_Cidade	End_UF
▶	00013254774	Bernardo	Edson Bernardes	83989183137	Rua Alzira Luciana de Oliveira	281	58309846	Comercial Norte	Bayeux	PB
	00057538131	Erick	Sócrates	67992084467	Rua Antenor Navarro	723	58400976	Prata	Campina Grande	PB
	03796551874	Clara	Ayla Julia de Paula	66999610565	Avenida Guanarapes	622	78730314	Jardim Igassu	Rondonópolis	MT
	06027205830	Sophie	Sarah Martins	83998682459	Rua Antenor Navarro	723	58400976	Prata	Campina Grande	PB
	08607937707	Sueli	Flávia Tatiane Araújo	82995703367	Rua Antenor Gomes de Oliveira	864	57055905	Farol	Maceió	AL
	11552598829	Kaique	Theo da Mota	77989994321	Rua T	781	45070450	São Pedro	Vitória da Conquista	BA
	18707312555	Raul	César Pinto	66996935708	Avenida das Águas	627	78553240	Residencial das Acácias	Sinop	MT
	22197478168	Pietro	Ricardo Breno Assis	61983713594	Quadra QE 17 Conjunto I	100	71050092	Guará II	Brasília	DF
	26880615422	Giovanni	Matheus Vieira	48985573280	Rua Antônio Silveira	131	88810436	Mina do Mato	Criciúma	SC

estoque – registra a quantidade em estoque, valor unitário e valor total dos livros;

Table: estoque					
Columns:					
Cod_Prod	int PK				
Qtd	int				
Vlr_Unid	decimal(10,2)				
Vlr_Total	decimal(10,2)				

Cod_Prod	Qtd	Vlr_Unid	Vlr_Total
1	52	40.99	2131.48
2	100	45.20	4520.00
3	70	50.10	3507.00
4	10	149.99	1499.90
5	3	6.30	18.90
6	5	20.15	100.75
7	15	50.00	750.00
8	27	55.00	1485.00
9	29	60.00	1740.00

-- Reference: FK_ESTOQUE (table: ESTOQUE)

ALTER TABLE ESTOQUE ADD CONSTRAINT FK_PEDIDOS_CAT_LIVROS FOREIGN KEY
FK_PEDIDOS_CAT_LIVROS (Cod_Prod)

REFERENCES CAT_LIVROS (Cod_Prod);

Foreign Key: FK_PEDIDOS_CAT_LIVROS	
Definition:	
Target	cat_livros (Cod_Prod → Cod_Prod)
On Update	RESTRICT
On Delete	RESTRICT

loja – cadastro com informações das lojas que fazem parte da rede;

Table: loja											
Columns:											
Cod_Loja	int PK										
Nome	varchar(50)										
CNPJ	decimal(14,0)										
Telefone	decimal(11,0)										
End_Rua	varchar(50)										
End_Numero	decimal(5,0)										
End_Complet	varchar(50)										
End_Bairro	varchar(50)										
End_Cidade	varchar(50)										
End_UF	varchar(2)										

Cod_Loja	Nome	CNPJ	Telefone	End_Rua	End_Numero	End_Complet	End_Bairro	End_Cidade	End_UF
125	Loja_MG	94706905000125	3135130931	Beco Seis	135	Pôr do Sol	São Francisco	Coronel Fabriciano	MG
143	Loja_SP	38466789000143	11345678765	Travessa Maestro Cardim	75	Shopping center	Jardins	São Paulo	SP
146	Loja_PR	22418392000146	4126635714	Travessa Murtede	117	Pôr do Sol	Santa Terezinha	Colombo	PR
170	Loja_GO	76370580000170	6125153408	Quadra QB 31	588	Mansões Pôr do Sol	Águas Lindas de Goiás	Goiás	GO

pedidos – registra as transações de venda, contendo entre outros, o código do cliente, do livro, quantidade, valor unitário e total, loja e etc;

Table: pedidos										
Columns:										
Cod_CPF	varchar(11)									
Cod_Pedido	int PK									
Cod_Prod	int									
Qtd	int									
Data_Hora	datetime									
Nota_Fiscal	int									
Cod_loja	int									
Vlr_Unid	decimal(10,2)									
Vlr_Total	decimal(10,2)									

Cod_CPF	Cod_Pedido	Cod_Prod	Qtd	Data_Hora	Nota_Fiscal	Cod_loja	Vlr_Unid	Vlr_Total
06027205830	1	3	2	2022-08-14 09:17:00	123132	125	50.10	100.20
74499106400	2	5	10	2022-08-14 13:00:00	1232510	143	6.30	63.00
22197478168	3	1	3	2022-08-14 13:15:00	123313	146	40.99	122.97
11552598829	4	1	4	2022-08-14 19:59:00	123414	170	40.99	163.96
11552598829	5	2	2	2022-08-14 22:22:00	123522	170	45.20	90.40
18707312555	6	20	5	2022-08-15 07:10:00	1236205	146	37.00	185.00
51669309401	7	19	1	2022-08-15 09:30:00	1237191	125	36.00	36.00
91632221756	8	17	2	2022-08-15 18:35:00	1238172	143	25.00	50.00
26880615422	9	15	1	2022-08-16 09:17:00	1239151	146	10.99	10.99

-- Reference: FK_PEDIDOS (table: PEDIDOS)

```
ALTER TABLE PEDIDOS ADD CONSTRAINT FK_PEDIDOS_CLIENTES FOREIGN KEY  
FK_PEDIDOS_CLIENTES (Cod_CPF)
```

```
REFERENCES CLIENTES (Cod_CPF);
```

Foreign Key: FK_PEDIDOS_CLIENTES

Definition:

Target	clientes (Cod_CPF → Cod_CPF)
On Update	RESTRICT
On Delete	RESTRICT

-- Reference: FK_PEDIDOS (table: PEDIDOS)

```
ALTER TABLE PEDIDOS ADD CONSTRAINT FK_PEDIDOS_ESTOQUE FOREIGN KEY  
FK_PEDIDOS_ESTOQUE (Cod_Prod)
```

```
REFERENCES ESTOQUE (Cod_Prod);
```

Foreign Key: FK_PEDIDOS_ESTOQUE

Definition:

Target	estoque (Cod_Prod → Cod_Prod)
On Update	RESTRICT
On Delete	RESTRICT

-- Reference: FK_PEDIDOS (table: PEDIDOS)

```
ALTER TABLE PEDIDOS ADD CONSTRAINT FK_PEDIDOS_LOJA FOREIGN KEY FK_PEDIDOS_LOJA  
(Cod_Loja)
```

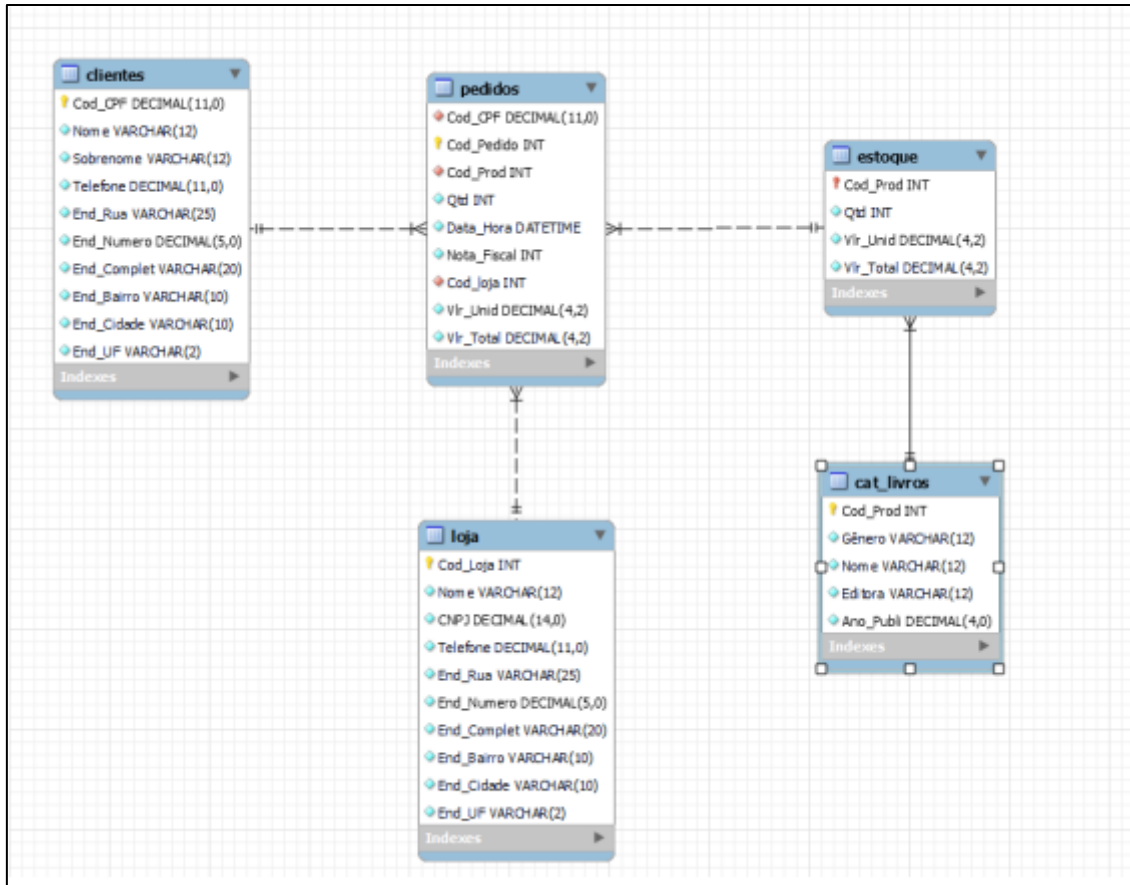
```
REFERENCES LOJA (Cod_Loja);
```

Foreign Key: FK_PEDIDOS_LOJA

Definition:

Target	loja (Cod_loja → Cod_Loja)
On Update	RESTRICT
On Delete	RESTRICT

DIAGRAMA DE ENTIDADE RELACIONAMENTO



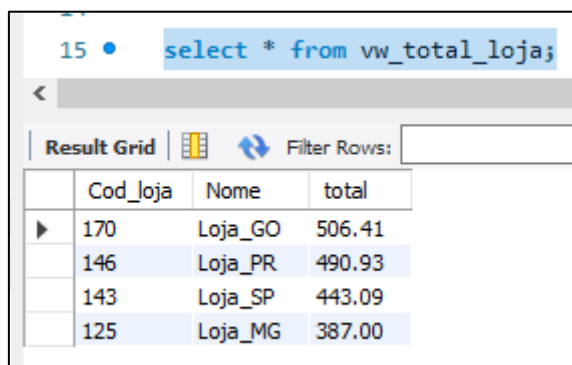
LISTAGEM DE VIEWS

1ª) Valor total por loja: view que soma o total vendido por loja, com o objetivo de rankiamento das lojas da rede. Seleciona o código da loja e o valor total da tabela *pedidos*, trazendo o nome da loja via JOIN da tabela *loja*.

```
USE livraria;

-- SOMAR VALOR TOTAL POR LOJA
-- TABELAS: pedidos, loja
-- OBJETIVO: Rankiamento das lojas

CREATE OR REPLACE VIEW vw_total_loja AS
SELECT p.Cod_loja, l.Nome, SUM(Vlr_Total) AS total
FROM pedidos as p
LEFT JOIN loja as l on (p.Cod_loja = l.Cod_loja)
GROUP BY p.Cod_loja;
```



	Cod_loja	Nome	total
▶	170	Loja_GO	506.41
	146	Loja_PR	490.93
	143	Loja_SP	443.09
	125	Loja_MG	387.00

2ª) Lojas que já fizeram uma ou mais vendas: view que retorna o código da loja e o nome da loja da tabela *loja*, que tiveram uma ou mais vendas registradas na tabela *pedidos*.



```
-- TRAZER APENAS AS LOJAS QUE JÁ VENDERAM
-- TABELAS: loja, pedidos
-- OBJETIVO: Verificar andamento da campanha de mkt

CREATE OR REPLACE VIEW vw_loja_vendeu AS
SELECT DISTINCT l.Cod_loja, l.Nome
FROM loja as l
INNER JOIN pedidos as p on (p.Cod_loja = l.Cod_loja)
WHERE l.Cod_Loja IN (SELECT p.cod_loja FROM pedidos);
```

26

27 • `select * from vw_loja_vendeu;`

<

Result Grid   Filter Rows:

	Cod_loja	Nome
▶	125	Loja_MG
	143	Loja_SP
	146	Loja_PR
	170	Loja_GO

3ª) Clientes que compraram na loja com UF diferente da do seu endereço: view que retorna o CPF, nome e UF de residência do cliente da tabela *cliente*, bem como a UF da loja da tabela *loja*, dos clientes que apresentam UF do endereço diferente do UF da loja registrados na tabela *pedidos*;

```
-- TRAZER CLIENTES QUE COMPRARAM DA LOJA DO ESTADO DIFERENTE DO DA RESIDENCIA
-- TABELAS: clientes, pedidos, loja
-- OBJETIVO: Entender por que os clientes não compraram na loja da residência;




• CREATE OR REPLACE VIEW vw_cliente_dif_uf AS
SELECT DISTINCT c.Cod_CPF, c.Nome, c.End_UF as Cliente_UF, l.End_UF as Loja_UF
FROM clientes as c
INNER JOIN pedidos as p on (c.Cod_CPF = p.Cod_CPF)
INNER JOIN loja as l on (l.Cod_Loja = p.Cod_Loja)
WHERE c.End_UF <> l.End_UF
ORDER BY Nome;
```

40

41 • `select * from vw_cliente_dif_uf;`

42

<

Result Grid   Filter Rows: Export: 

	Cod_CPF	Nome	Cliente_UF	Loja_UF
▶	99072294661	Andrea	PR	SP
	00013254774	Bernardo	PB	PR
	91632221756	Bianca	PE	SP
	03796551874	Clara	MT	GO
	26880615422	Giovanni	SC	PR
	45168890563	Ian	RJ	PR
	11552598829	Kaique	BA	GO
	43014459275	Lucca	MG	SP
	51669309401	Murilo	PR	MG
	53895487600	Pietra	AP	MG
	22197478168	Pietro	DF	PR
	18707312555	Raul	MT	PR
	06027205830	Sophie	PB	MG
	08607937707	Sueli	AL	GO
	74499106400	Tereza	PB	SP

4ª) Clientes que compraram mais de uma vez: view que retorna – se o cliente comprou mais de uma vez - a quantidade de vezes que o cliente comprou da tabela *pedidos*, nome e UF de residência do cliente da tabela *cliente*.

```
-- TRAZER CLIENTES QUE COMPRARAM MAIS DE UMA VEZ
-- TABELAS: pedidos, clientes
-- OBJETIVO: Analisar recorrência

CREATE OR REPLACE VIEW vw_cliente_recorrente AS
SELECT DISTINCT p.Cod_CPF, c.Nome, c.End_UF as Cliente_UF, COUNT(p.Cod_CPF) as QTD
FROM pedidos as p
INNER JOIN clientes as c on (c.Cod_CPF = p.Cod_CPF)
GROUP BY p.Cod_CPF
HAVING QTD <> 1;
```

54 • `select * from vw_cliente_recorrente;`

Result Grid | Filter Rows: | Export:

	Cod_CPF	Nome	Cliente_UF	QTD
▶	11552598829	Kaique	BA	2
	37062475009	Valentina	MG	2
	43014459275	Lucca	MG	2

5ª) Livros com estoque menor que 10: view que retorna o código do livro da tabela *estoque*, juntamente com o seu respectivo nome da tabela *cat_livros*, dos livros que apresentem estoque menor que 10.

```
-- TRAZER LIVROS COM ESTOQUE MENOR QUE 10
-- TABELAS: estoque, cat_livros
-- OBJETIVO: Verificar livros para repor estoque;

• CREATE OR REPLACE VIEW vw_estoque_menor10 AS
SELECT e.cod_prod, c.nome
FROM estoque as e
INNER JOIN cat_livros as c on ( e.cod_prod = c.cod_prod )
WHERE Qtd < 10;
```

65
66 • `select * from vw_estoque_menor10;`

Result Grid | Filter Rows: | Export:

	cod_prod	nome
▶	5	Chatô o Rei do Brasil
	6	Steve Jobs
	13	Dom Casmurro
	14	Iracema
	15	Macunaíma

LISTAGEM DE FUNÇÕES

1ª) Nome do Livro: função que retorna o nome do livro da tabela *cat_livros*, conforme o *cod_prod* informado.

```
1  -- CRIAR FUNÇÃO PARA TRAZER O NOME DO LIVRO
2  -- TABELAS: cat_livros
3  -- OBJETIVO: Consulta rápida ao nome do livro pelo código
4
5
6  DELIMITER $$
7  • USE `livraria`$$
8  • CREATE DEFINER=`root`@`localhost` FUNCTION `nome_livros`(codigos INT) RETURNS varchar(100) CHARSET utf8mb4
9    READS SQL DATA
10  BEGIN
11    DECLARE resultado VARCHAR(100);
12    SET resultado = (SELECT Nome FROM cat_livros where Cod_Prod = codigos);
13    RETURN resultado;
14  END$$
15
16  DELIMITER ;
17  ;
18
19  -- TESTE DA FUNÇÃO
20
21  • SELECT nome_livros(10) as NOME;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

NOME
Quem Mexeu no Meu Queijo?

2ª) Valor do estoque: função que retorna o valor total do estoque da tabela *estoque*, conforme o *cod_prod* informado.

```
23  -- FUNÇÃO COM O VALOR TOTAL DO ESTOQUE
24  -- TABELAS: estoque
25  -- OBJETIVO: Análise do estoque pelo código do livro
26
27  DELIMITER $$
28  • USE `livraria`$$
29  • CREATE DEFINER=`root`@`localhost` FUNCTION `total_estoque`(codigos INT) RETURNS FLOAT
30    READS SQL DATA
31  BEGIN
32    DECLARE resultado FLOAT;
33    SET resultado = (SELECT Qtd * Vlr_Unid from estoque where Cod_Prod = codigos);
34    RETURN resultado;
35  END$$
36
37  DELIMITER ;
38  ;
39
40  -- TESTE DA FUNÇÃO
41
42  • SELECT total_estoque(1);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

total_estoque(1)
2131.48

LISTAGEM DE STORED PROCEDURES

1ª) sp_ordena_clientes: procedure que seleciona e ordena (ASC OU DESC) a tabela *clientes* conforme o campo escolhido pelo usuário. Campo pode ser (Cod_CPF, Nome, Sobrenome, Telefone, End_Rua, End_Numero, End_Complet, End_Bairro, End_Cidade, End_UF).

```
DELIMITER $$
USE `livraria`$$
DROP PROCEDURE IF EXISTS sp_ordena_clientes;
CREATE PROCEDURE sp_ordena_clientes (IN campo CHAR(20), IN tipo CHAR(4))
BEGIN
    IF campo <> '' THEN
        SET @campoorder = concat('ORDER BY ', campo);
    ELSE
        SET @campoorder = '';
    END IF;

    SET @clausula = concat('SELECT * FROM clientes ', @campoorder, ' ', tipo);
    PREPARE runSql from @clausula;
    EXECUTE runSql;
END $$
;
```

27
28 `call sp_ordena_clientes('Nome','Asc');`

Result Grid | Filter Rows: | Export: | Wrap Cell Content: I

Cod_CPF	Nome	Sobrenome	Telefone	End_Rua	End_Numero	End_Complet	End_Bairro	End_Cidade	End_UF
99072294661	Andrea	Joana Rezende	43998921260	Rua Gavião-de-cauda-barrada	676	86702772	Jardim Mônaco	Arapongas	PR
00013254774	Bernardo	Edson Bernardes	83989183137	Rua Alzira Luciana de Oliveira	281	58309846	Comercial Norte	Bayeux	PB
91632221756	Bianca	Sophie Bianca Teixeira	81994424766	Rua Antônio Carlos Mergulhão	809	53402705	Fragoso	Paulista	PE
03796551874	Clara	Ayla Julia de Paula	66999610565	Avenida Guararapes	622	78730314	Jardim Iguassu	Rondonópolis	MT
00057538131	Erick	Sócrates	67992084467	Rua Antenor Navarro	723	58400976	Prata	Campina Grande	PB
26880615422	Giovanni	Matheus Vieira	48985573280	Rua Antônio Silveira	131	88810436	Mina do Mato	Criciúma	SC
45168890563	Ian	Carlos Benedito da Paz	21982681994	Rua Euzébio de Farias	202	21920110	Tauá	Rio de Janeiro	RJ
11552598829	Kaique	Theo da Mota	77989994321	Rua T	781	45070450	São Pedro	Vitória da Conquista	BA
91705507948	Kauê	Tiago Jesus	45996112953	Rua Machado de Assis	314	85909350	Vila Pioneiro	Toledo	PR
43014459275	Lucca	Felipe Lucas Castro	34998583688	Praça Lisboa	213	38444330	Santiago	Araguari	MG
76679253721	Milena	Louise Ferreira	86996924361	Rua Santa Rita de Cássia	614	64212300	Alto Santa Maria	Parnaíba	PI

29
30 `call sp_ordena_clientes('Nome','Desc');`
31 --

Result Grid | Filter Rows: | Export: | Wrap Cell Content: I

Cod_CPF	Nome	Sobrenome	Telefone	End_Rua	End_Numero	End_Complet	End_Bairro	End_Cidade	End_UF
37062475009	Valentina	Olivia Beatriz da Mata	35994401354	Praça Antônio Pernambuco Chaves	621	37002620	Centro	Varginha	MG
74499106400	Tereza	Heloisa Bárbara Costa	83981655281	Travessa Marcone Coutinho	796	58700205	Centro	Patos	PB
08607937707	Sueli	Flávia Tatiane Araújo	82995703367	Rua Antenor Gomes de Oliveira	864	57055905	Farol	Maceió	AL
06027205830	Sophie	Sarah Martins	83998682459	Rua Antenor Navarro	723	58400976	Prata	Campina Grande	PB
18707312555	Raul	César Pinto	66996935708	Avenida das Águas	627	78553240	Residencial das Acácias	Sinop	MT
41013665511	Raul	Vicente Calebe Dias	47998428149	Rua São Floriano	197	89230445	Jarivatuba	Joinville	SC
22197478168	Pietro	Ricardo Breno Assis	61983713594	Quadra QE 17 Conjunto I	100	71050092	Guará II	Brasília	DF
53895487600	Pietra	Allana Beatriz Silva	96986079062	Rua Mendes de Sá	464	68908512	São Lázaro	Macapá	AP
96157068507	Pedro	Tiago Figueiredo	53996942231	Passo Um	329	96040604	Fragata	Pelotas	RS
51669309401	Murilo	Augusto Márcio Barros	41984144237	Rua Anselmo Piliati	991	82120550	Pilarzinho	Curitiba	PR
76679253721	Milena	Louise Ferreira	86996924361	Rua Santa Rita de Cássia	614	64212300	Alto Santa Maria	Parnaíba	PI
43014459275	Lucca	Felipe Lucas Castro	34998583688	Praça Lisboa	213	38444330	Santiago	Araguari	MG
91705507948	Kauê	Tiago Jesus	45996112953	Rua Machado de Assis	314	85909350	Vila Pioneiro	Toledo	PR

2ª) sp_inserir_livros: procedure que insere um novo livro na tabela *cat_livros*. Para inserção, colocar os seguintes dados (ID, GÊNERO, NOME, EDITORA, ANO DE PUBLICAÇÃO E NOME DO ESCRITOR), caso um desses dados não for informado, a seguinte mensagem deve aparecer (ERRO AO CADASTRAR LIVRO, FAVOR INFORMAR TODOS OS CAMPOS: "CODIGO", "GENERO", "NOME", "EDITOR", "ANO PUBLICAÇÃO" E "ESCRITOR").

```

56 DELIMITER $$
57 • USE `livraria`$$
58 • DROP PROCEDURE IF EXISTS sp_inserir_livros;
59 CREATE PROCEDURE sp_inserir_livros (
60     IN codigo CHAR(4),
61     IN genero CHAR(50),
62     IN nome CHAR(200),
63     IN editora CHAR(50),
64     IN ano decimal(4,0),
65     IN escritor CHAR(100))
66 BEGIN
67     IF codigo = '' OR genero = '' OR nome = '' OR editora = '' OR ano = '' OR escritor = '' THEN
68         SELECT 'ERRO AO CADASTRAR LIVRO, FAVOR INFORMAR TODOS OS CAMPOS "CODIGO", "GENERO", "NOME", "EDITOR", "ANO PUBLICAÇÃO" E "ESCRITOR" AS MENSAGEM ;
69     ELSE
70         SET @nv1 = CONCAT( "INSERT INTO cat_livros(Cod_Prod, Gênero, Nome, Editora, Ano_Publi, Escritor) VALUES", "(", codigo ,",",",",genero,",",",",nome,",",",",editora,",",",",ano,",",",",escritor,",",",");");
71     PREPARE runSql FROM @nv1;
72     EXECUTE runSql;
73     END IF;
74 END $$

```

```

57
58 • call sp_inserir_livros(21,'Ficção Científica','2001: A Space Odyssey','Aleph',1968,'Arthur C. Clarke');

```

Cod_Prod	Gênero	Nome	Editora	Ano_Publi	Escritor
8	História	Homo Deus	L&pm	2018	Yuval Noah Harari
9	História	21 lições para o século 21	L&pm	2019	Yuval Noah Harari
10	Auto-Ajuda	Quem Mexeu no Meu Queijo?	Record	2020	Spencer Johnson
11	Auto-Ajuda	Casais Inteligentes Enriquecem Juntos	Gente	2004	Gustavo Cerbasi
12	Auto-Ajuda	Os 100 Segredos das Pessoas Felizes	Sextante	2001	David Niven
13	Literatura Brasileira	Dom Casmurro	Ática	1978	Machado de Assis
14	Literatura Brasileira	Iracema	Ciranda cultural	1997	José de Alencar
15	Literatura Brasileira	Macunaíma	Villa Rica	2001	Mário de Andrade
16	Ciência Política	O Príncipe	Hunter books	2011	Maquiavel
17	Ciência Política	Lava Jato	Primeira Pessoa	2016	Vladimir Netto
18	Literatura Estrangeira	O Caçador de Pipas	Nova fronteira	2005	Khaled Hosseini
19	Literatura Estrangeira	O Código da Vinci	Sextante	2004	Dan Brown
20	Literatura Estrangeira	A cabana	Sextante	2008	William P. Young
21	Ficção Científica	2001: A Space Odyssey	Aleph	1968	Arthur C. Clarke
NULL	NULL	NULL	NULL	NULL	NULL

```

57
58 • call sp_inserir_livros(21, , '2001: A Space Odyssey','Aleph',1968,'Arthur C. Clarke');

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
MENSAGEM			
ERRO AO CADASTRAR LIVRO, FAVOR INFORMAR TODOS OS CAMPOS "CODIGO", "GENERO", "NOME", "EDITOR", "ANO PUBLICAÇÃO" E "ESCRITOR"			

SCRIPTS

<https://github.com/ErickSocrates/Sql-CoderHouse>