

## PROJETO BANCO DE DADOS: MODELO ER/DDL/DML e exemplos de DQL

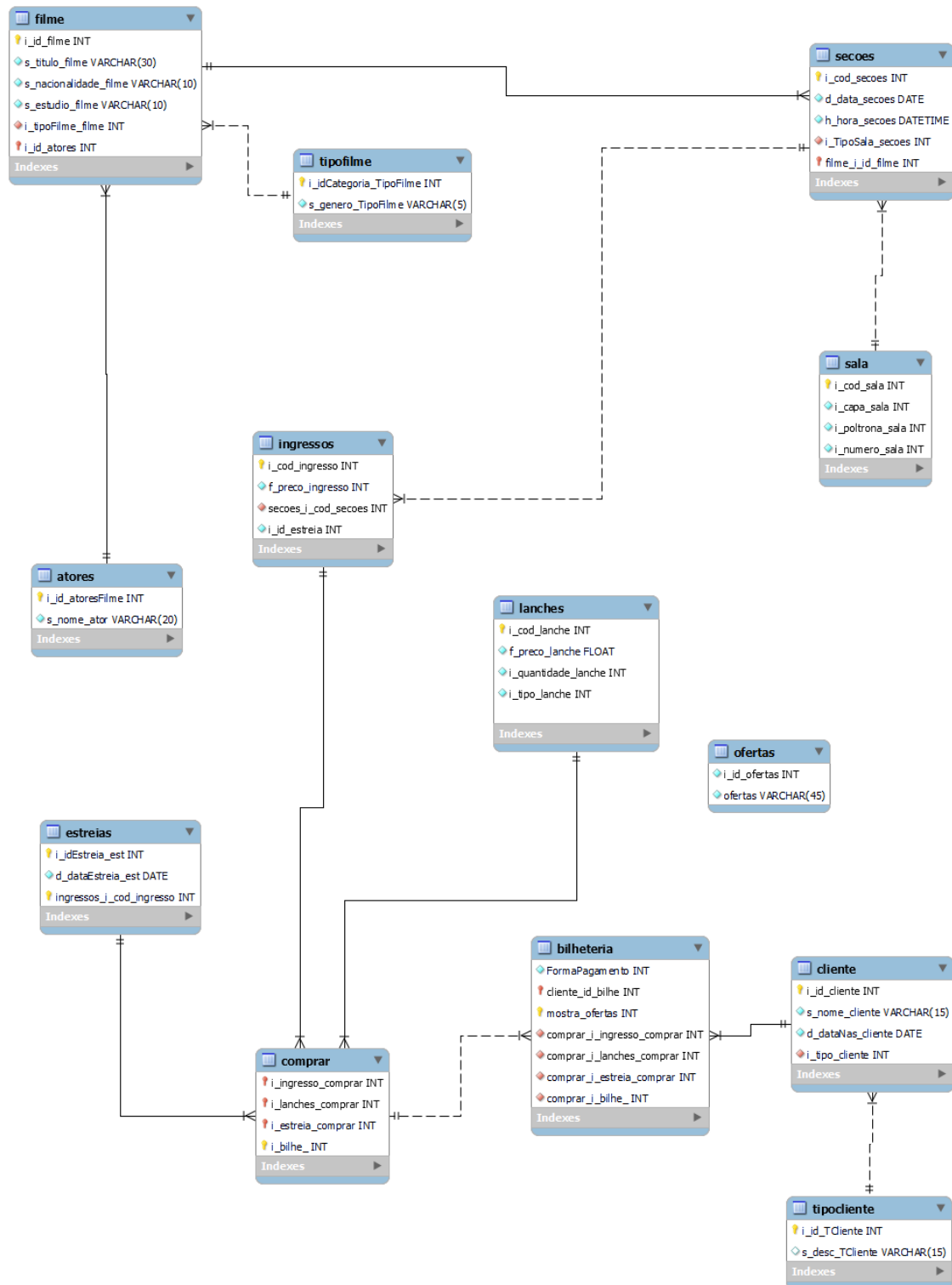
Professor: Marcelo Iury de Sousa Oliveira

CI/UFPB

Período: 2022.1

Equipe: MIGUEL DE PAULA DA SILVA CRUZ (20190024690) e JACKSON LEANDRO DO NASCIMENTO (20180019614)

### MODELO Entidade Relacionamento



### Observações gerais:

Utilizamos o MySQL como SGBD em consonância com a linguagem Python para a manipulação de tais dados, para modelagem gráfica e conexão com o banco de dados foi utilizado ferramentas que o framework Django disponibiliza. O código do banco de dados foi escrito totalmente em SQL, e a modelo diagrama entidade, antes feito a mão/programas, foi gerado pelo próprio SGBD MySQL.

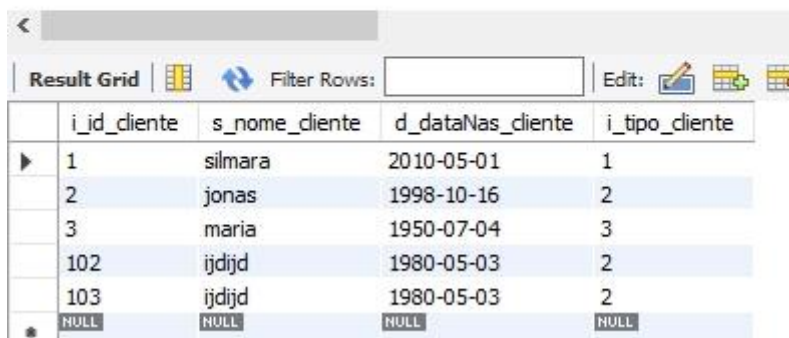
### Apresentações iniciais:

Antes de analisar o código do banco de dados, vamos analisar poucos exemplos de DML para as tabelas TipoCliente e Cliente:

```
1
2 • insert into tipocliente values (1, 'crianca');
3 • insert into tipocliente values (2, 'adulto');
4 • insert into tipocliente values (3, 'idoso');
5
6 • insert into cliente values (100, 'silmara', '2010-05-01', 1);
7 • insert into cliente values (101, 'jonas', '1998-10-16', 2);
8 • insert into cliente values (102, 'maria', '1950-07-04', 3);
```

Começamos inserindo os registros na tabela TipoCliente, onde temos o id 1 para criança, 2 para adulto e 3 para idoso. (Posteriormente adicionamos o id 666 para flamenguista).

```
1 • select *from cliente
```



	i_id_cliente	s_nome_cliente	d_dataNas_cliente	i_tipo_cliente
▶	1	silmara	2010-05-01	1
	2	jonas	1998-10-16	2
	3	maria	1950-07-04	3
	102	ijdiyd	1980-05-03	2
	103	ijdiyd	1980-05-03	2
•	NULL	NULL	NULL	NULL

(Print na tabela) Posteriormente adicionamos dois clientes de nomes inexistentes, somente para alterar...

```

1 • update cliente set s_nome_cliente = 'jefferson' where i_id_cliente = 102;
2 • update cliente set s_nome_cliente = 'rafaela' where i_id_cliente = 103;
3 • update cliente set d_dataNas_cliente = '1990-08-05' where i_id_cliente = 103;
4 • select *from cliente

```

Result Grid				
Filter Rows:				
Edit:				
Export/Import:				
Wrap				
i_id_cliente	s_nome_cliente	d_dataNas_cliente	i_tipo_cliente	
1	silmara	2010-05-01	1	
2	jonas	1998-10-16	2	
3	maria	1950-07-04	3	
102	jefferson	1980-05-03	2	
103	rafaela	1990-08-05	2	
NULL	NULL	NULL	NULL	

Esta parte da tabela é de extrema importância, já que daqui podemos definir se uma pessoa consegue meia ou entrada gratuita pelo seu tipo e data de nascimento.

Todo o código do banco de dados tem seus relacionamentos indicados em suas próprias tabelas, mas obviamente em algum momento utilizamos conceitos DDL como:

```
alter table <tabela_origem> add constraint <nome_restricao>
```

```
foreign key(<campo_tabelaOrigem>)
```

```
references <tabela_destino> (<campo_tabela_destino>);
```

para fazer/mudar relacionamentos (1:n) entre tabelas.

### Sintaxe de Leitura:

A maior parte do código, para efeitos de fácil leitura entre programadores e sua implementação, principalmente para chaves estrangeiras, foi utilizada a seguinte norma de ordem para a declaração de um atributo de uma tabela:

<tipo\_de\_dados>\_<nome\_atributo>\_<tabela\_origem>.

Exemplo:

O ID da tabela cliente é do tipo INT e tem como origem a tabela CLIENTE:

```
I_id_cliente INT PRIMARY KEY NOT NULL;
```

O NOME da tabela cliente é do tipo String e tem como origem a tabela CLIENTE:

```
S_nome_cliente VARCHAR(20) NOT NULL;
```

Tal norma e ordem segue durante todas as tabelas.

Segue o código SQL com a criação das tabelas e seus relacionamentos:

```

1  CREATE database cinema;
2  USE cinema ;
3
4  -- Tabela: atores
5  CREATE TABLE atores (
6      i_id_atoresFilme INT NOT NULL,
7      s_nome_ator VARCHAR(20) NOT NULL,
8      PRIMARY KEY (i_id_atoresFilme));
9
10 -- Tabela: tipocliente
11 CREATE TABLE tipocliente (
12     i_id_TCliente INT NOT NULL,
13     s_desc_TCliente VARCHAR(15) NULL DEFAULT NULL,
14     PRIMARY KEY (i_id_TCliente));
15
16 -- Tabela: cliente
17 CREATE TABLE cliente (
18     i_id_cliente INT NOT NULL,
19     s_nome_cliente VARCHAR(15) NOT NULL,
20     d_dataNas_cliente DATE NOT NULL,
21     i_tipo_cliente INT NOT NULL,
22     PRIMARY KEY (i_id_cliente),
23     INDEX fk_cliente (i_tipo_cliente ASC) VISIBLE,
24     CONSTRAINT fk_cliente
25         FOREIGN KEY (i_tipo_cliente)
26         REFERENCES tipocliente (i_id_TCliente));
27
28 -- Tabela: estreias
29 CREATE TABLE estreias (
30     i_idEstreia_est INT NOT NULL,
31     d_dataEstreia_est DATE NOT NULL,
32     ingressos_i_cod_ingresso INT NOT NULL,
33     PRIMARY KEY (i_idEstreia_est, ingressos_i_cod_ingresso));
34
35 -- Tabela: tipofilme;
36 CREATE TABLE tipofilme (
37     i_idCategoria_TipoFilme INT NOT NULL,
38     s_genero_TipoFilme VARCHAR(5) NOT NULL,
39     PRIMARY KEY (i_idCategoria_TipoFilme));
40
41 -- Tabela: filme
42 CREATE TABLE filme (
43     i_id_filme INT NOT NULL,
44     s_titulo_filme VARCHAR(30) NOT NULL,
45     s_nacionalidade_filme VARCHAR(10) NOT NULL,
46     s_estudio_filme VARCHAR(10) NOT NULL,
47     i_tipoFilme_filme INT NOT NULL,
48     i_id_atores INT NOT NULL,
49     PRIMARY KEY (i_id_filme, i_id_atores),
50     INDEX fk_tipoFilme (i_tipoFilme_filme ASC) VISIBLE,
51     INDEX fk_atores_filme (i_id_atores ASC) VISIBLE,
52     CONSTRAINT fk_atores_filme
53         FOREIGN KEY (i_id_atores)
54         REFERENCES atores (i_id_atoresFilme),
55     CONSTRAINT fk_tipoFilme
56         FOREIGN KEY (i_tipoFilme_filme)
57         REFERENCES tipofilme (i_idCategoria_TipoFilme));
58
59 -- Tabela: sala
60 CREATE TABLE sala (
61     i_cod_sala INT NOT NULL,
62     i_capa_sala INT NOT NULL,
63     i_poltrona_sala INT NOT NULL,
64     i_numero_sala INT NOT NULL,
65     PRIMARY KEY (i_cod_sala));
66
67 -- Tabela: secoes
68 CREATE TABLE secoes (
69     i_cod_secoes INT NOT NULL,
70     d_data_secoes DATE NOT NULL,
71     h_hora_secoes DATETIME NOT NULL,
72     i_TipoSala_secoes INT NOT NULL,
73     filme_i_id_filme INT NOT NULL,
74     PRIMARY KEY (i_cod_secoes, filme_i_id_filme),
75     INDEX i_TipoSala_secoes (i_TipoSala_secoes ASC) VISIBLE,
76     INDEX fk_secoes_filme1_idx (filme_i_id_filme ASC) VISIBLE,
77     CONSTRAINT fk_secoes_filme1
78         FOREIGN KEY (filme_i_id_filme)
79         REFERENCES filme (i_id_filme),
80     CONSTRAINT secoes_ibfk_1
81         FOREIGN KEY (i_TipoSala_secoes)
82         REFERENCES sala (i_cod_sala));
83
84 -- Tabela: ingressos
85 CREATE TABLE ingressos (
86     i_cod_ingresso INT NOT NULL,
87     f_preco_ingresso INT NOT NULL,
88     secoes_i_cod_secoes INT NOT NULL,
89     i_id_estreia INT NOT NULL,
90     PRIMARY KEY (i_cod_ingresso),
91     INDEX fk_ingressos_secoes1_idx (secoes_i_cod_secoes ASC) VISIBLE,
92     CONSTRAINT fk_ingressos_secoes1
93         FOREIGN KEY (secoes_i_cod_secoes)
94         REFERENCES secoes (i_cod_secoes));
95
96 -- Tabela: lanches
97 CREATE TABLE lanches (
98     i_cod_lanche INT NOT NULL,
99     f_preco_lanche FLOAT NOT NULL,
100    i_quantidade_lanche INT NOT NULL,
101    i_tipo_lanche INT NOT NULL,
102    PRIMARY KEY (i_cod_lanche));
103
104 -- Tabela: comprar
105 CREATE TABLE comprar (
106     i_ingresso_comprar INT NOT NULL,
107     i_lanches_comprar INT NOT NULL,
108     i_estreia_comprar INT NOT NULL,
109     i_bilhe_ INT NOT NULL,
110     PRIMARY KEY (i_ingresso_comprar, i_lanches_comprar, i_estreia_comprar, i_bilhe_),
111     INDEX fk_comprar_estreias1_idx (i_estreia_comprar ASC) VISIBLE,
112     INDEX fk_comprar_lanches1_idx (i_lanches_comprar ASC) VISIBLE,
113     CONSTRAINT fk_comprar_estreias1
114         FOREIGN KEY (i_estreia_comprar)
115         REFERENCES estreias (i_idEstreia_est),
116     CONSTRAINT fk_comprar_ingressos1
117         FOREIGN KEY (i_ingresso_comprar)
118         REFERENCES ingressos (i_cod_ingresso),
119     CONSTRAINT fk_comprar_lanches1
120         FOREIGN KEY (i_lanches_comprar)
121         REFERENCES lanches (i_cod_lanche));

```

```
122
123 -- Tabela: bilheteria
124 CREATE TABLE bilheteria (
125     FormaPagamento INT NOT NULL,
126     cliente_id_bilhe INT NOT NULL,
127     mostra_ofertas INT NOT NULL,
128     comprar_i_ingresso_comprar INT NOT NULL,
129     comprar_i_lanches_comprar INT NOT NULL,
130     comprar_i_estreia_comprar INT NOT NULL,
131     comprar_i_bilhe INT NOT NULL,
132     PRIMARY KEY (cliente_id_bilhe, mostra_ofertas),
133     INDEX fk_bilheteria_clientel_idx (cliente_id_bilhe ASC) VISIBLE,
134     INDEX fk_bilheteria_comprar1_idx (comprar_i_ingresso_comprar ASC, comprar_i_lanches_comprar ASC, comprar_i_estreia_comprar ASC, comprar_i_bilhe ASC) VISIBLE,
135     CONSTRAINT fk_bilheteria_clientel
136         FOREIGN KEY (cliente_id_bilhe)
137         REFERENCES cliente (i_id_cliente)
138         ON DELETE NO ACTION
139         ON UPDATE NO ACTION,
140     CONSTRAINT fk_bilheteria_comprar1
141         FOREIGN KEY (comprar_i_ingresso_comprar , comprar_i_lanches_comprar , comprar_i_estreia_comprar , comprar_i_bilhe_)
142         REFERENCES comprar (i_ingresso_comprar , i_lanches_comprar , i_estreia_comprar , i_bilhe_)
143         ON DELETE NO ACTION
144         ON UPDATE NO ACTION);
145
146 -- Tabela: ofertas
147 CREATE TABLE ofertas (
148     i_id_ofertas INT NOT NULL,
149     ofertas VARCHAR(45) NOT NULL
150 );
```