

# Sistema de Cinema

Felipe Luís Pinheiro - 18/0052667

João Pedro C.N. Mota - 17/0106144

Pedro Catelli - 17/0112624      Pedro Oliveira - 17/0163768

3 de julho de 2019

## Resumo

Neste relatório desenvolvemos os requisitos básicos de um sistema de banco de dados para um modelo de vendas de ingresso de um cinema.

Link para o repositório: [https://github.com/flpinheiro/banco\\_de\\_dados](https://github.com/flpinheiro/banco_de_dados).

## 1 Introdução

Requisitos gerais:

- Um cinema pode ter muitas salas, sendo necessário, por tanto, registrar informações a respeito de cada uma, como sua capacidade, ou seja, o numero de assentos disponíveis.
- O cinema apresenta muitos filmes. Um filme tem informações, titulo e duração. Assim, sempre que um filme for ser apresentado, deve-se registrá-lo também.
- Um mesmo filme pode ser apresentado em diferentes salas e em horários diferentes. Cada apresentação em uma determinada sala e horário é chamada sessão. Um filme sendo apresentado em uma sessão tem um conjunto máximo de ingressos, determinado pela capacidade da sala.
- Os clientes do cinema podem comprar ou não ingressos para assistir a uma sessão. O funcionário deve intermediar a compra do ingresso. Um ingresso deve conter informação como o tipo de ingresso (Meio ingresso ou ingresso inteiro). Além disso, um cliente só pode comprar ingressos para sessões ainda não encerradas.

## 2 Diagrama de Entidade Relacionamento

Na figura 1 mostramos a primeira versão conceitual do sistema do

## 3 Modelo Relacional

Na figura 2 mostramos o modelo relacional utilizado para implementação do programa

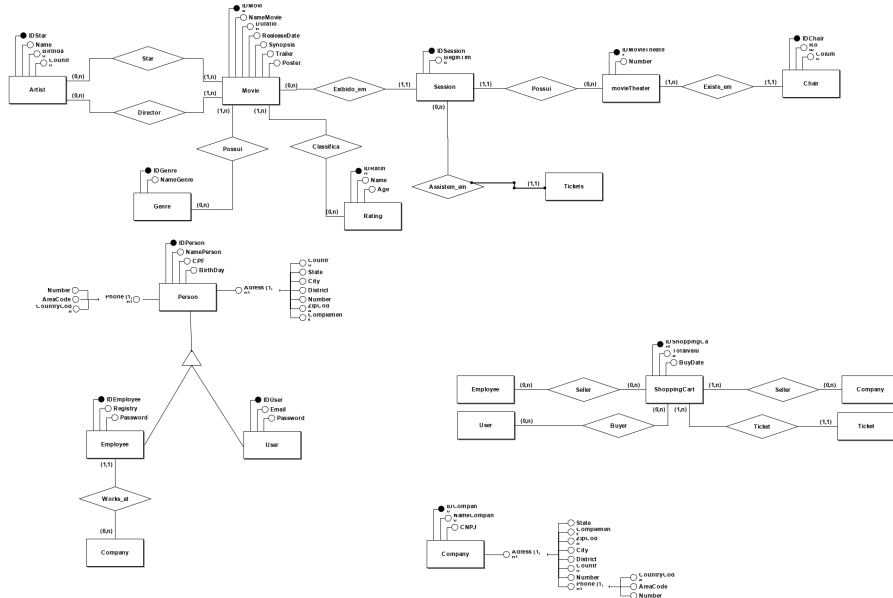


Figura 1: Modelo Entidade Relacionamento

## 4 Consultas

Nesta seção mostramos exemplo de consultas que podem ser realizadas nesse modelo relacional de banco de dados.

```

1      use unbcineflix;
2
3      select * FROM movies, ratings, genremovies, genres
where ratingid = ratings.id and movies.id = genremovies.movieid
and genremovies.genreid = genres.id;
4
5      select * from movies, artistmovies, artists where
Movies.id = artistmovies.MovieId and artistmovies.ArtistId =
artists.Id;
6
7      select * from movietheaters, addresses, companies
where addresses.Id = movietheaters.AddressCompanyId and
addresses.CompanyId = companies.Id and addresses.Discriminator
= 'AddressCompany';
8
9      select * from session, movietheaters, tickets where
session.Id = tickets.SessionId and session.AddressCompanyId =
movietheaters.AddressCompanyId and movietheaters.
MovieTheaterNumber = session.MovieTheaterNumber;
10
11     select * from people, addresses, phones where people
.id = addresses.PersonId and people.id = phones.PersonId and
addresses.Discriminator = 'AddressPerson';

```

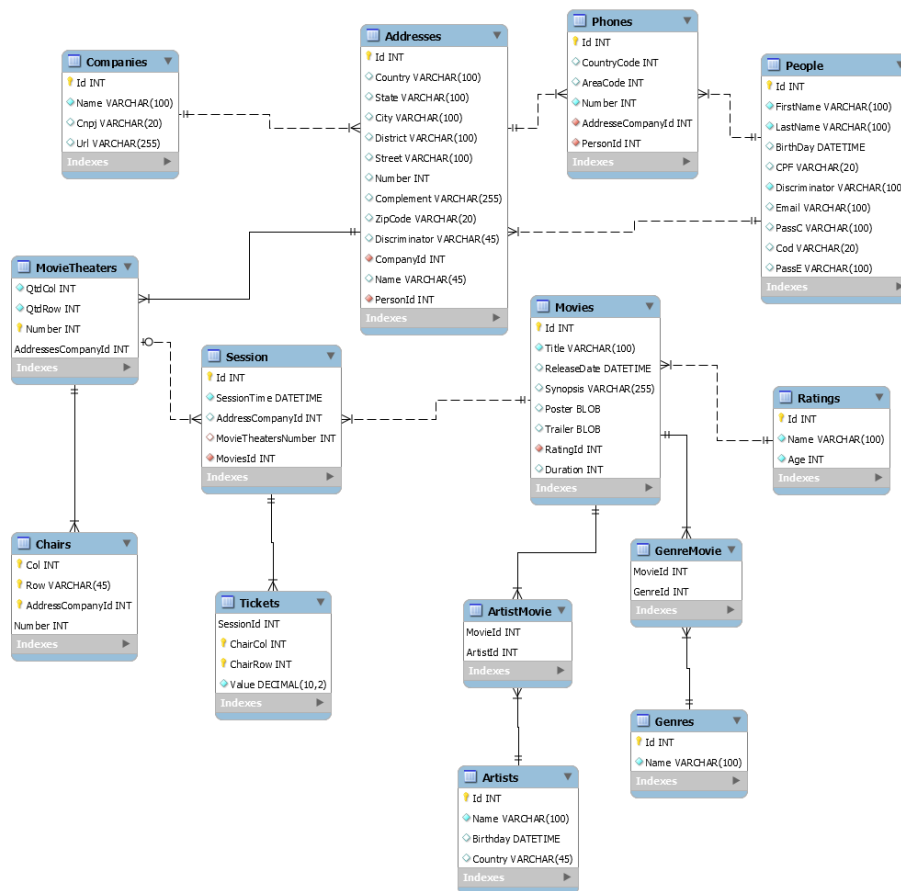


Figura 2: Modelo Relacional

## 5 Script Sql

Nesta seção mostramos o script sql para geração do banco de dados, que foi gerado utilizando o modelo acima e foi gerado automaticamente pelo MySQL.

```

1      -- MySQL Script generated by MySQL Workbench
2      -- Thu Jun 27 18:36:45 2019
3      -- Model: New Model      Version: 2.0
4      -- MySQL Workbench Forward Engineering
5
6      SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,
UNIQUE_CHECKS=0;
7      SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
8      SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='
ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,
NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION'
;
9
10

```

```

11      -- Schema UnBCineFlix
12      --
13      DROP SCHEMA IF EXISTS `UnBCineFlix` ;
14
15      --
16      -- Schema UnBCineFlix
17      --
18      CREATE SCHEMA IF NOT EXISTS `UnBCineFlix` DEFAULT
19      CHARACTER SET utf8 ;
20      USE `UnBCineFlix` ;
21
22      -- Table `UnBCineFlix`.`Addresses`
23      --
24      CREATE TABLE IF NOT EXISTS `UnBCineFlix`.`Addresses`
25      ` (
26          `Id` INT NOT NULL AUTO_INCREMENT,
27          `Country` VARCHAR(100) NULL,
28          `State` VARCHAR(100) NULL,
29          `City` VARCHAR(100) NULL,
30          `District` VARCHAR(100) NULL,
31          `Street` VARCHAR(100) NULL,
32          `Number` INT NULL,
33          `Complement` VARCHAR(255) NULL,
34          `ZipCode` VARCHAR(20) NULL,
35          `Discriminator` VARCHAR(45) NULL,
36          `CompanyId` INT NOT NULL,
37          `Name` VARCHAR(45) NULL,
38          `PersonId` INT NOT NULL,
39          PRIMARY KEY (`Id`),
40          INDEX `fk_Addresses_People1_idx` (`PersonId`
41          ` ASC) VISIBLE,
42          INDEX `fk_Addresses_Companies1_idx` (`
43          CompanyId` ASC) VISIBLE,
44          CONSTRAINT `fk_Addresses_People1`
45          FOREIGN KEY (`PersonId`)
46          REFERENCES `UnBCineFlix`.`People`
47          (`Id`)
48          ON DELETE NO ACTION
49          ON UPDATE NO ACTION,
50          CONSTRAINT `fk_Addresses_Companies1`
51          FOREIGN KEY (`CompanyId`)
52          REFERENCES `UnBCineFlix`.`Companies`
53          (`Id`)
54          ON DELETE NO ACTION
55          ON UPDATE NO ACTION)
56      ENGINE = InnoDB;
57
58      -- Table `UnBCineFlix`.`ArtistMovie`
59      --
60      CREATE TABLE IF NOT EXISTS `UnBCineFlix`.`
61      ArtistMovie` (
62          `MovieId` INT NOT NULL,

```

```

59         `ArtistId` INT NOT NULL,
60         PRIMARY KEY (`MovieId`, `ArtistId`),
61         INDEX `fk_Movie_has_Artist_Artist1_idx` (`
ArtistId` ASC) VISIBLE,
62         INDEX `fk_Movie_has_Artist_Movie1_idx` (`
MovieId` ASC) VISIBLE,
63         CONSTRAINT `fk_Movie_has_Artist_Movie1`
64         FOREIGN KEY (`MovieId`)
65         REFERENCES `UnBCineFlix`.`Movies`
(`Id`)
66         ON DELETE NO ACTION
67         ON UPDATE NO ACTION,
68         CONSTRAINT `fk_Movie_has_Artist_Artist1`
69         FOREIGN KEY (`ArtistId`)
70         REFERENCES `UnBCineFlix`.`Artists`
(`Id`)
71         ON DELETE NO ACTION
72         ON UPDATE NO ACTION)
73     ENGINE = InnoDB;
74
75     ---
76     --- Table `UnBCineFlix`.`Artists`
77     ---
78
79     CREATE TABLE IF NOT EXISTS `UnBCineFlix`.`Artists`
(
80         `Id` INT NOT NULL,
81         `Name` VARCHAR(100) NOT NULL,
82         `Birthday` DATETIME NULL,
83         `Country` VARCHAR(45) NULL,
84         PRIMARY KEY (`Id`))
85     ENGINE = InnoDB;
86
87     ---
88     --- Table `UnBCineFlix`.`Chairs`
89     ---
90
91     CREATE TABLE IF NOT EXISTS `UnBCineFlix`.`Chairs` (
92         `Col` INT NOT NULL,
93         `Row` VARCHAR(45) NOT NULL,
94         `AddressCompanyId` INT NOT NULL,
95         `Number` INT NOT NULL,
96         PRIMARY KEY (`Col`, `Row`, `
AddressCompanyId`, `Number`),
97         INDEX `fk_Chairs_MovieTheaters1_idx` (`
AddressCompanyId` ASC, `Number` ASC) VISIBLE,
98         CONSTRAINT `fk_Chairs_MovieTheaters1`
99         FOREIGN KEY (`Number`)
100         REFERENCES `UnBCineFlix`.`
MovieTheaters` (`Number`)
101         ON DELETE NO ACTION
102         ON UPDATE NO ACTION)
103     ENGINE = InnoDB;
104
105     ---
106     --- Table `UnBCineFlix`.`Companies`
107

```

```

108      --
109      CREATE TABLE IF NOT EXISTS `UnBCineFlix`.`Companies`
110      (
111          `Id` INT NOT NULL AUTO_INCREMENT,
112          `Name` VARCHAR(100) NOT NULL,
113          `Cnpj` VARCHAR(20) NULL,
114          `Url` VARCHAR(255) NULL,
115          PRIMARY KEY (`Id`))
116      ENGINE = InnoDB;
117
118      --
119      -- Table `UnBCineFlix`.`GenreMovie`
120      --
121      CREATE TABLE IF NOT EXISTS `UnBCineFlix`.`GenreMovie` (
122          `MovieId` INT NOT NULL,
123          `GenreId` INT ZEROFILL NOT NULL,
124          PRIMARY KEY (`MovieId`, `GenreId`),
125          INDEX `fk_Movie_has_Genre_Genre1_idx` (`GenreId` ASC) VISIBLE,
126          INDEX `fk_Movie_has_Genre_Movie1_idx` (`MovieId` ASC) VISIBLE,
127          CONSTRAINT `fk_Movie_has_Genre_Movie1`
128              FOREIGN KEY (`MovieId`)
129              REFERENCES `UnBCineFlix`.`Movies`
130              (`Id`)
131              ON DELETE NO ACTION
132              ON UPDATE NO ACTION,
133          CONSTRAINT `fk_Movie_has_Genre_Genre1`
134              FOREIGN KEY (`GenreId`)
135              REFERENCES `UnBCineFlix`.`Genres`
136              (`Id`)
137              ON DELETE NO ACTION
138              ON UPDATE NO ACTION)
139      ENGINE = InnoDB;
140
141      --
142      -- Table `UnBCineFlix`.`Genres`
143      --
144      CREATE TABLE IF NOT EXISTS `UnBCineFlix`.`Genres` (
145          `Id` INT ZEROFILL NOT NULL,
146          `Name` VARCHAR(100) NOT NULL,
147          PRIMARY KEY (`Id`))
148      ENGINE = InnoDB;
149
150      --
151      -- Table `UnBCineFlix`.`MovieTheaters`
152      --
153      CREATE TABLE IF NOT EXISTS `UnBCineFlix`.`MovieTheaters` (
154          `QtdCol` INT NOT NULL,
155          `QtdRow` INT NOT NULL,

```

```

156         `Number` INT NOT NULL,
157         `AddressesCompanyId` INT NOT NULL,
158         PRIMARY KEY (`Number`, `AddressesCompanyId`
159     `),
160     INDEX `fk_MovieTheaters_Addresses1_idx` (`
161     AddressesCompanyId`
162     ASC) VISIBLE,
163     CONSTRAINT `fk_MovieTheaters_Addresses1`
164     FOREIGN KEY (`AddressesCompanyId`)
165     REFERENCES `UnBCineFlix`.`Addresses`
166     (`Id`)
167     ON DELETE NO ACTION
168     ON UPDATE NO ACTION)
169     ENGINE = InnoDB;
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208

```

---

```

169     -- Table `UnBCineFlix`.`Movies`
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208

```

---

```

171     CREATE TABLE IF NOT EXISTS `UnBCineFlix`.`Movies` (
172         `Id` INT NOT NULL AUTO_INCREMENT,
173         `Title` VARCHAR(100) NOT NULL,
174         `ReleaseDate` DATETIME NULL,
175         `Synopsis` VARCHAR(255) NULL,
176         `Poster` BLOB NULL,
177         `Trailer` BLOB NULL,
178         `RatingId` INT NOT NULL,
179         `Duration` INT NULL,
180         PRIMARY KEY (`Id`),
181         INDEX `fk_Movie_Rating1_idx` (`RatingId`
182     ASC) VISIBLE,
183         CONSTRAINT `fk_Movie_Rating1`
184         FOREIGN KEY (`RatingId`)
185         REFERENCES `UnBCineFlix`.`Ratings`
186         (`Id`)
187         ON DELETE NO ACTION
188         ON UPDATE NO ACTION)
189         ENGINE = InnoDB;
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208

```

---

```

191     -- Table `UnBCineFlix`.`People`
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208

```

---

```

193     CREATE TABLE IF NOT EXISTS `UnBCineFlix`.`People` (
194         `Id` INT NOT NULL AUTO_INCREMENT,
195         `FirstName` VARCHAR(100) NOT NULL,
196         `LastName` VARCHAR(100) NOT NULL,
197         `BirthDay` DATETIME NULL,
198         `CPF` VARCHAR(20) NULL,
199         `Discriminator` VARCHAR(100) NOT NULL,
200         `Email` VARCHAR(100) NULL,
201         `PassC` VARCHAR(100) NULL,
202         `Cod` VARCHAR(20) NULL,
203         `PassE` VARCHAR(100) NULL,
204         PRIMARY KEY (`Id`))
205         ENGINE = InnoDB;
206
207
208

```

```

209      --- Table `UnBCineFlix`.`Phones`
210      ---
211      CREATE TABLE IF NOT EXISTS `UnBCineFlix`.`Phones` (
212          `Id` INT NOT NULL AUTO_INCREMENT,
213          `CountryCode` INT NULL,
214          `AreaCode` INT NULL,
215          `Number` INT NOT NULL,
216          `AdresseCompanyId` INT NOT NULL,
217          `PersonId` INT NOT NULL,
218          PRIMARY KEY (`Id`),
219          INDEX `fk_Phones_Addresses1_idx` (`
AdresseCompanyId` ASC) VISIBLE,
220          INDEX `fk_Phones_People1_idx` (`PersonId`
ASC) VISIBLE,
221          CONSTRAINT `fk_Phones_Addresses1`
222              FOREIGN KEY (`AdresseCompanyId`)
223              REFERENCES `UnBCineFlix`.`Addresses`
224              (`Id`)
225              ON DELETE NO ACTION
226              ON UPDATE NO ACTION,
227          CONSTRAINT `fk_Phones_People1`
228              FOREIGN KEY (`PersonId`)
229              REFERENCES `UnBCineFlix`.`People`
230              (`Id`)
231              ON DELETE NO ACTION
232              ON UPDATE NO ACTION)
233      ENGINE = InnoDB;
234
235      --- Table `UnBCineFlix`.`Ratings`
236      ---
237      CREATE TABLE IF NOT EXISTS `UnBCineFlix`.`Ratings`
238      (
239          `Id` INT NOT NULL AUTO_INCREMENT,
240          `Name` VARCHAR(100) NOT NULL,
241          `Age` INT NOT NULL,
242          PRIMARY KEY (`Id`))
243      ENGINE = InnoDB;
244
245      ---
246      --- Table `UnBCineFlix`.`Session`
247      ---
248      CREATE TABLE IF NOT EXISTS `UnBCineFlix`.`Session`
249      (
250          `Id` INT NOT NULL AUTO_INCREMENT,
251          `SessionTime` DATETIME NOT NULL,
252          `AddressCompanyId` INT NULL,
253          `MovieTheatersNumber` INT NULL,
254          `MoviesId` INT NOT NULL,
255          PRIMARY KEY (`Id`),
256          INDEX `fk_Session_MovieTheaters1_idx` (`
AddressCompanyId` ASC, `MovieTheatersNumber` ASC) VISIBLE,
257          INDEX `fk_Session_Movies1_idx` (`MoviesId`
ASC) VISIBLE,

```



```

257             CONSTRAINT `fk_Session_MovieTheaters1`
258             FOREIGN KEY (`MovieTheatersNumber`)
259             REFERENCES `UnBCineFlix`.``
MovieTheaters` (`Number`)
260             ON DELETE NO ACTION
261             ON UPDATE NO ACTION,
262             CONSTRAINT `fk_Session_Movies1`
263             FOREIGN KEY (`MoviesId`)
264             REFERENCES `UnBCineFlix`.``Movies`
(`Id`)
265             ON DELETE NO ACTION
266             ON UPDATE NO ACTION)
267         ENGINE = InnoDB;
268
269         ---
270     ---
271     --- Table `UnBCineFlix`.``Tickets`
272     ---
273     CREATE TABLE IF NOT EXISTS `UnBCineFlix`.``Tickets`
274     (
275         `SessionId` INT NOT NULL,
276         `ChairCol` INT NOT NULL,
277         `ChairRow` INT NOT NULL,
278         `Value` DECIMAL(10,2) NOT NULL,
279         PRIMARY KEY (`SessionId`, `ChairCol`, `ChairRow`),
280         INDEX `fk_Tickets_Session1_idx` (`SessionId`
281         `ASC) VISIBLE,
282         CONSTRAINT `fk_Tickets_Session1`
283         FOREIGN KEY (`SessionId`)
284         REFERENCES `UnBCineFlix`.``Session`
285         (`Id`)
286         ON DELETE NO ACTION
287         ON UPDATE NO ACTION)
288         ENGINE = InnoDB;
289
290         SET SQL_MODE=@OLD_SQL_MODE;
291         SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
292         SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

## 6 Álgebra relacional

## 7 Avaliação das formas normais