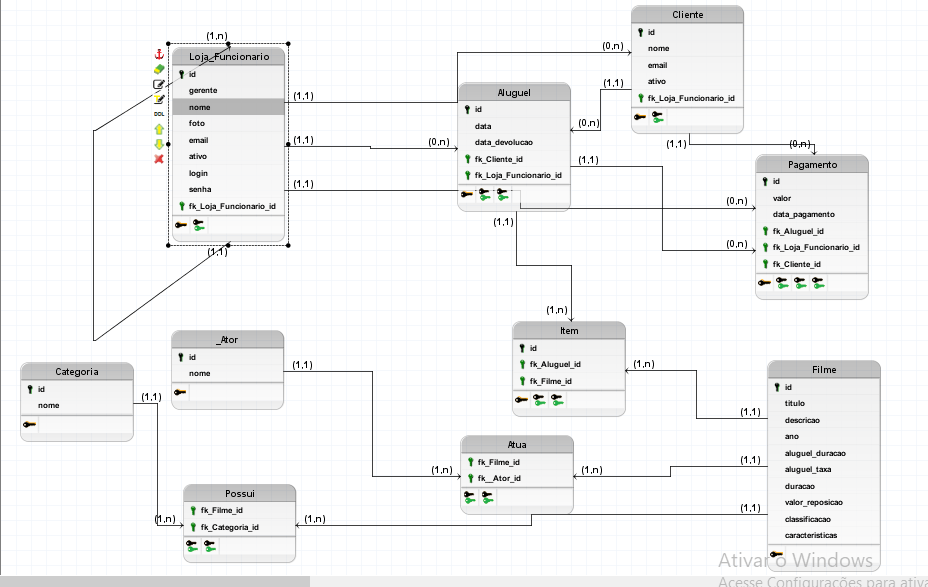
Projeto Banco de Dados Relacional: Locadora

Guilherme Augusto da Silva Afonso

**Modelo lógico BRMODELO:**



**Modelo Físico:**

CREATE TABLE Loja\_Funcionario (

id INTEGER PRIMARY KEY,

gerente VARCHAR(30),

nome VARCHAR(30),

foto BLOB,

email VARCHAR(30),

ativo BOOLEAN,

login VARCHAR(30),

senha VARCHAR(30),

fk\_Loja\_Funcionario\_id INTEGER

);

CREATE TABLE Cliente (

id INTEGER PRIMARY KEY,

nome VARCHAR(30),

email VARCHAR(30),

ativo BOOLEAN,

fk\_Loja\_Funcionario\_id INTEGER

);

CREATE TABLE \_Ator (

id INTEGER PRIMARY KEY,

nome VARCHAR(30)

);

CREATE TABLE Categoria (

id INTEGER PRIMARY KEY,

nome VARCHAR(30)

);

CREATE TABLE Filme (

id INTEGER PRIMARY KEY,

titulo VARCHAR(30),

descricao VARCHAR(30),

ano INTEGER,

aluguel\_duracao TIMESTAMP,

aluguel\_taxa FLOAT,

duracao TIME,

valor\_reposicao FLOAT,

classificacao VARCHAR(30),

caracteristicas VARCHAR(30)

);

CREATE TABLE Aluguel (

id INTEGER PRIMARY KEY,

data DATE,

data\_devolucao DATE,

fk\_Cliente\_id INTEGER,

fk\_Loja\_Funcionario\_id INTEGER

);

CREATE TABLE Item (

id INTEGER PRIMARY KEY,

fk\_Aluguel\_id INTEGER,

fk\_Filme\_id INTEGER

);

CREATE TABLE Pagamento (

id INTEGER PRIMARY KEY,

valor FLOAT,

data\_pagamento DATE,

fk\_Aluguel\_id INTEGER,

fk\_Loja\_Funcionario\_id INTEGER,

fk\_Cliente\_id INTEGER

);

CREATE TABLE Possui (

fk\_Filme\_id INTEGER,

fk\_Categoria\_id INTEGER

);

CREATE TABLE Atua (

fk\_Filme\_id INTEGER,

fk\_\_Ator\_id INTEGER

);

ALTER TABLE Loja\_Funcionario ADD CONSTRAINT FK\_Loja\_Funcionario\_2

FOREIGN KEY (fk\_Loja\_Funcionario\_id)

REFERENCES Loja\_Funcionario (id);

ALTER TABLE Cliente ADD CONSTRAINT FK\_Cliente\_2

FOREIGN KEY (fk\_Loja\_Funcionario\_id)

REFERENCES Loja\_Funcionario (id)

ON DELETE CASCADE;

ALTER TABLE Aluguel ADD CONSTRAINT FK\_Aluguel\_2

FOREIGN KEY (fk\_Cliente\_id)

REFERENCES Cliente (id)

ON DELETE CASCADE;

ALTER TABLE Aluguel ADD CONSTRAINT FK\_Aluguel\_3

FOREIGN KEY (fk\_Loja\_Funcionario\_id)

REFERENCES Loja\_Funcionario (id)

ON DELETE CASCADE;

ALTER TABLE Item ADD CONSTRAINT FK\_Item\_2

FOREIGN KEY (fk\_Aluguel\_id)

REFERENCES Aluguel (id)

ON DELETE RESTRICT;

ALTER TABLE Item ADD CONSTRAINT FK\_Item\_3

FOREIGN KEY (fk\_Filme\_id)

REFERENCES Filme (id)

ON DELETE RESTRICT;

ALTER TABLE Pagamento ADD CONSTRAINT FK\_Pagamento\_2

FOREIGN KEY (fk\_Aluguel\_id)

REFERENCES Aluguel (id)

ON DELETE CASCADE;

ALTER TABLE Pagamento ADD CONSTRAINT FK\_Pagamento\_3

FOREIGN KEY (fk\_Loja\_Funcionario\_id)

REFERENCES Loja\_Funcionario (id)

ON DELETE CASCADE;

ALTER TABLE Pagamento ADD CONSTRAINT FK\_Pagamento\_4

FOREIGN KEY (fk\_Cliente\_id)

REFERENCES Cliente (id)

ON DELETE CASCADE;

ALTER TABLE Possui ADD CONSTRAINT FK\_Possui\_1

FOREIGN KEY (fk\_Filme\_id)

REFERENCES Filme (id)

ON DELETE RESTRICT;

ALTER TABLE Possui ADD CONSTRAINT FK\_Possui\_2

FOREIGN KEY (fk\_Categoria\_id)

REFERENCES Categoria (id)

ON DELETE RESTRICT;

ALTER TABLE Atua ADD CONSTRAINT FK\_Atua\_1

FOREIGN KEY (fk\_Filme\_id)

REFERENCES Filme (id)

ON DELETE RESTRICT;

ALTER TABLE Atua ADD CONSTRAINT FK\_Atua\_2

FOREIGN KEY (fk\_\_Ator\_id)

REFERENCES \_Ator (id)

ON DELETE RESTRICT;

**Views criadas:**

CREATE VIEW AtoresFilmes AS

SELECT A.nome AS Ator, F.titulo AS Filme

FROM \_Ator A

JOIN Atua AT ON A.id = AT.fk\_\_Ator\_id

JOIN Filme F ON AT.fk\_Filme\_id = F.id;

CREATE VIEW FilmesCategoriasAtores AS

SELECT F.titulo AS Filme, C.nome AS Categoria, A.nome AS Ator

FROM Filme F

JOIN Possui P ON F.id = P.fk\_Filme\_id

JOIN Categoria C ON P.fk\_Categoria\_id = C.id

JOIN Atua AT ON F.id = AT.fk\_Filme\_id

JOIN \_Ator A ON AT.fk\_\_Ator\_id = A.id;

CREATE VIEW FilmesMaisAlugados AS

SELECT F.titulo AS Filme, COUNT(\*) AS TotalAlugueis

FROM Filme F

JOIN Item I ON F.id = I.fk\_Filme\_id

GROUP BY F.titulo

ORDER BY TotalAlugueis DESC

LIMIT 5;

CREATE VIEW PagamentosEmAberto AS

SELECT P.id AS PagamentoID, C.nome AS Cliente, P.valor AS Valor

FROM Pagamento P

JOIN Cliente C ON P.fk\_Cliente\_id = C.id

WHERE P.data\_pagamento IS NULL;

CREATE VIEW LojasGerentes AS

SELECT L.nome AS Loja, LF.gerente AS Gerente

FROM Loja\_Funcionario LF

JOIN Loja\_Funcionario L ON LF.fk\_Loja\_Funcionario\_id = L.id;

CREATE VIEW ClientesLojas AS

SELECT L.nome AS Loja, C.nome AS Cliente

FROM Loja\_Funcionario LF

JOIN Loja\_Funcionario L ON LF.fk\_Loja\_Funcionario\_id = L.id

JOIN Cliente C ON LF.id = C.fk\_Loja\_Funcionario\_id

ORDER BY L.nome, C.nome;

**Criação da tabela Estoque:**

CREATE TABLE Estoque (

id INTEGER PRIMARY KEY,

fk\_Loja\_Funcionario\_id INTEGER,

fk\_Filme\_id INTEGER,

quantidade INTEGER,

FOREIGN KEY (fk\_Loja\_Funcionario\_id) REFERENCES Loja\_Funcionario (id),

FOREIGN KEY (fk\_Filme\_id) REFERENCES Filme (id)

);

ALTER TABLE Item ADD fk\_Estoque\_id INT;

ALTER TABLE Item ADD CONSTRAINT FK\_Item\_4

FOREIGN KEY (fk\_Estoque\_id)

REFERENCES Estoque (id)

ON DELETE RESTRICT;

**Function Inventario\_em\_estoque:**

DELIMITER @

CREATE FUNCTION inventario\_em\_Estoque(estoque\_id INT) RETURNS TINYINT(1)

BEGIN

DECLARE qtde\_alugados INT;

DECLARE qtde\_devolvidos INT;

DECLARE qtde\_total INT;

SELECT COUNT(\*) INTO qtde\_alugados

FROM Item I

JOIN Aluguel A ON I.fk\_Aluguel\_id = A.id

WHERE A.data\_devolucao IS NULL AND I.fk\_Estoque\_id = estoque\_id;

SELECT COUNT(\*) INTO qtde\_devolvidos

FROM Item I

JOIN Aluguel A ON I.fk\_Aluguel\_id = A.id

WHERE A.data\_devolucao IS NOT NULL AND I.fk\_Estoque\_id = estoque\_id;

SELECT quantidade INTO qtde\_total

FROM Estoque

WHERE id = estoque\_id;

IF qtde\_alugados + qtde\_devolvidos < qtde\_total THEN

RETURN 1;

ELSE

RETURN 0;

END IF;

END @

DELIMITER ;

**Stored Procedure Ator:**

DELIMITER @

CREATE PROCEDURE Ator(IN quantidade INT)

BEGIN

SELECT nome

FROM \_Ator

LIMIT quantidade;

END @

DELIMITER ;

**Stored Procedure Filmes\_em\_Estoque:**

DELIMITER @

CREATE PROCEDURE Filmes\_em\_Estoque(IN filme\_id INT, IN loja\_id INT, OUT filme\_qtde INT)

BEGIN

SELECT quantidade INTO filme\_qtde

FROM Estoque

WHERE fk\_Filme\_id = filme\_id AND fk\_Loja\_Funcionario\_id = loja\_id;

END @

DELIMITER ;

**Stored Procedure Filmes\_alugados:**

DELIMITER @

CREATE PROCEDURE Filmes\_alugados(IN filme\_id INT, IN loja\_id INT, OUT filme\_qtde INT)

BEGIN

SELECT COUNT(\*) INTO filme\_qtde

FROM Item I

JOIN Aluguel A ON I.fk\_Aluguel\_id = A.id

JOIN Estoque E ON I.fk\_Estoque\_id = E.id

WHERE A.data\_devolucao IS NULL AND E.fk\_Filme\_id = filme\_id AND E.fk\_Loja\_Funcionario\_id = loja\_id;

END @

DELIMITER ;

**Alterando tabela cliente:**

**ALTER TABLE Cliente ADD criado\_em DATETIME;**

**Criando tabela log:**

CREATE TABLE log (

reg VARCHAR(255),

msg VARCHAR(255)

);

**Criando trigger AFTER UPDATE:**

DELIMITER @

CREATE TRIGGER cliente\_AFTER\_UPDATE

AFTER UPDATE ON Cliente

FOR EACH ROW

BEGIN

INSERT INTO log (reg, msg) VALUES ('Cliente', CONCAT('ID: ', NEW.id, ', Nome: ', NEW.nome, ', E-mail: ', NEW.email, ', Ativo: ', NEW.ativo));

END @

delimiter ;