CSDS 341 HW 1

1

CK	SK	K	$\rightarrow$
		A	A
		B	A,B
✓	✓	C	A,B,C,D,E
		D	D
		E	D,E
		AB	A,B
	✓	AC	A,B,C,D,E
		AD	A,D
	✓	AE	A,B,C,D,E
	✓	BD	A,B,C,D,E
	✓	BE	A,B,C,D,E
		DE	D,E
	✓	ABD	A,B,C,D,E
	✓	ABE	A,B,C,D,E
	✓	BDE	A,B,C,D,E

r

A table may only have one Primary Key, which will be a selection from (sometimes) multiple Candidate Keys.

2

a

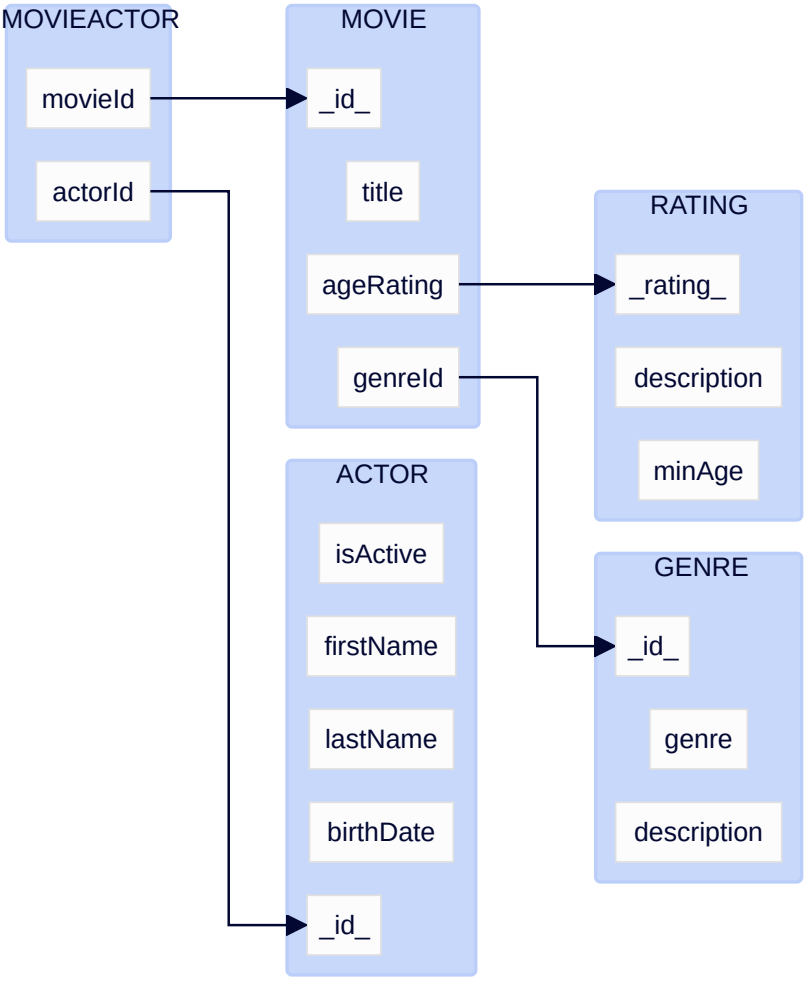
i

$\text{MOVIE}(\underline{\text{id}}, \text{title}, \text{ageRating}, \text{genreId})$
$\text{RATING}(\underline{\text{rating}}, \text{description}, \text{minAge})$
$\text{GENRE}(\underline{\text{id}}, \text{genre}, \text{description})$

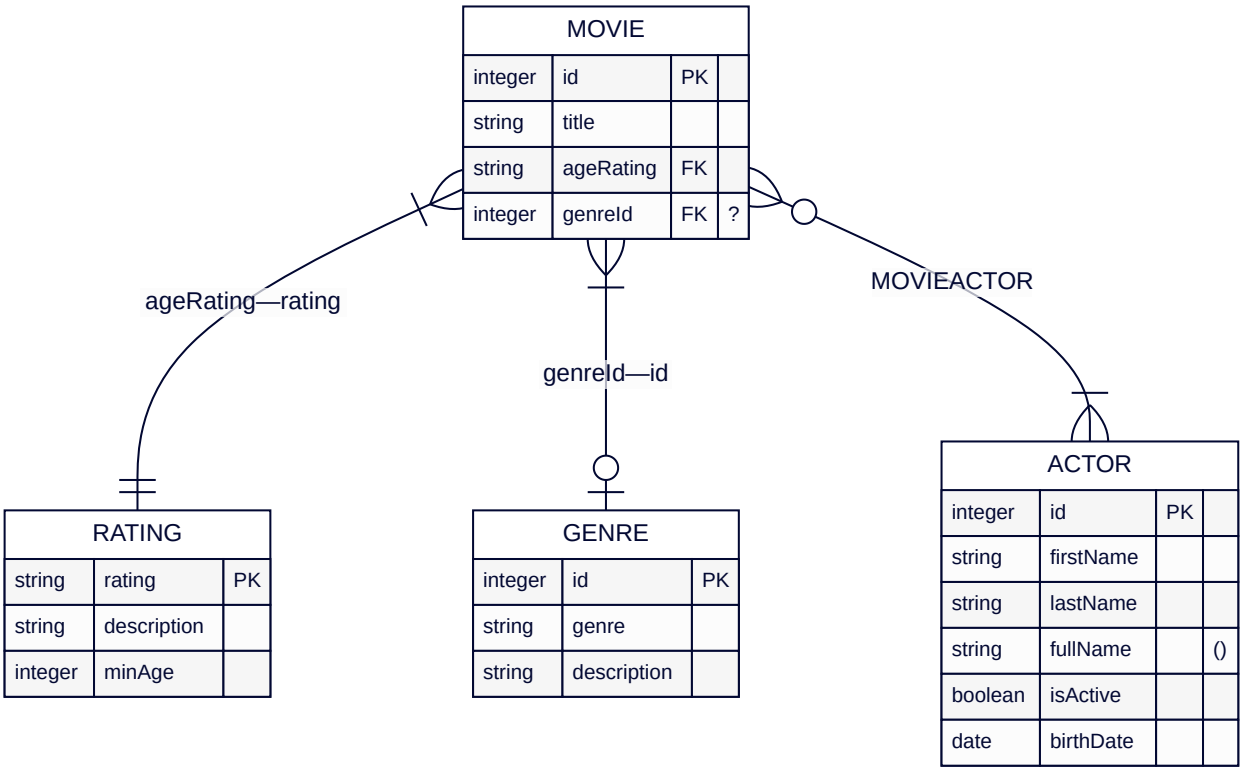
ACTOR(id, firstName, lastName, isActive, birthDate)
MOVIEACTOR(movieId, actorId)

1. I've renamed a few of the attributes, especially the id's in order to have a more consistent and clear description of the model
2. I changed the type of status from string to boolean as there is only two possible values and no point of storing a string
3. I split the name attribute into first and last name as it would be useful for generating the checks for payment, with a derived field for fullName
4. I added an actors attribute to the movie table in order to connect actors to movies. This attribute does not need to exist on the actor table as that would be redundant
5. I've also added a relational table, MOVIEACTOR for the many-to-many relation

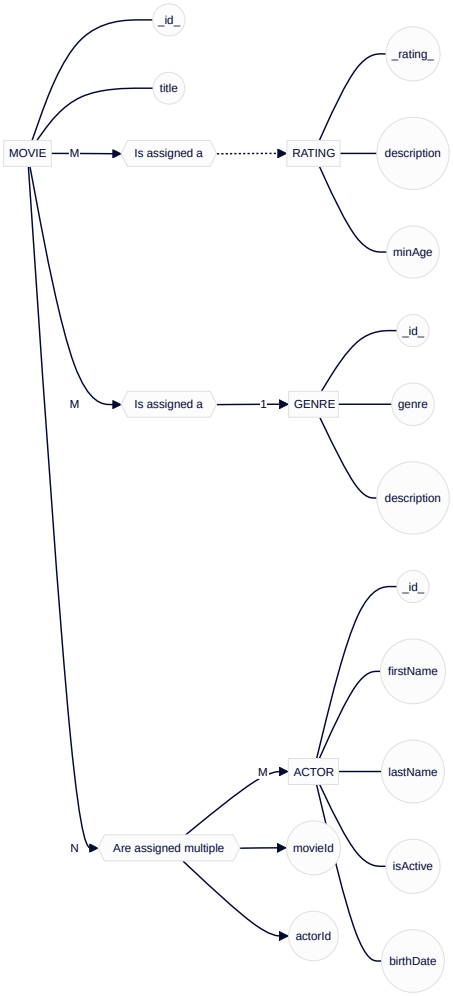
ii



iii


MOVIE
integer id PK
string title
string ageRating FK
integer genreId FK ?
ageRating—rating
genreId—id
MOVIEACTOR
RATING
string rating PK
string description
integer minAge
GENRE
integer id PK
string genre
string description
ACTOR
integer id PK
string firstName
string lastName
string fullName ()
boolean isActive
date birthDate

iv



V

MOVIES			
id	title	ageRating	genreId
1	Lost	G	1
2	Found	R	2

RATING		
rating	description	minAge
G	General	0
R	Restricted	18

GENRE		
id	genre	description
1	rom	Romance
2	sci-fi	Science Fiction

ACTOR				
id	firstName	lastName	isActive	birthDate
1	John	Doe	true	1950-01-01

ACTOR				
2	Johnny	Dohe	true	1953-03-01
3	Jon	Do	true	1959-02-01
4	Jhon	Dhoe	false	1955-08-01
5	Joe	Dohn	true	1952-04-01
6	Joey	Doey	true	1957-01-01

MOVIEACTOR	
movieId	actorId
1	1
1	2
1	3
2	2
2	4
2	5

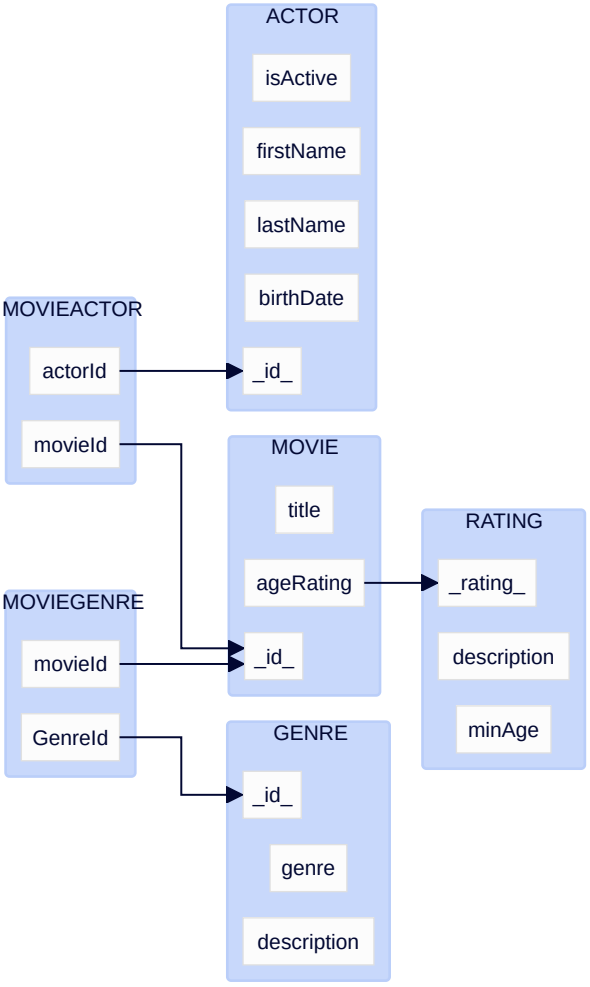
b

i

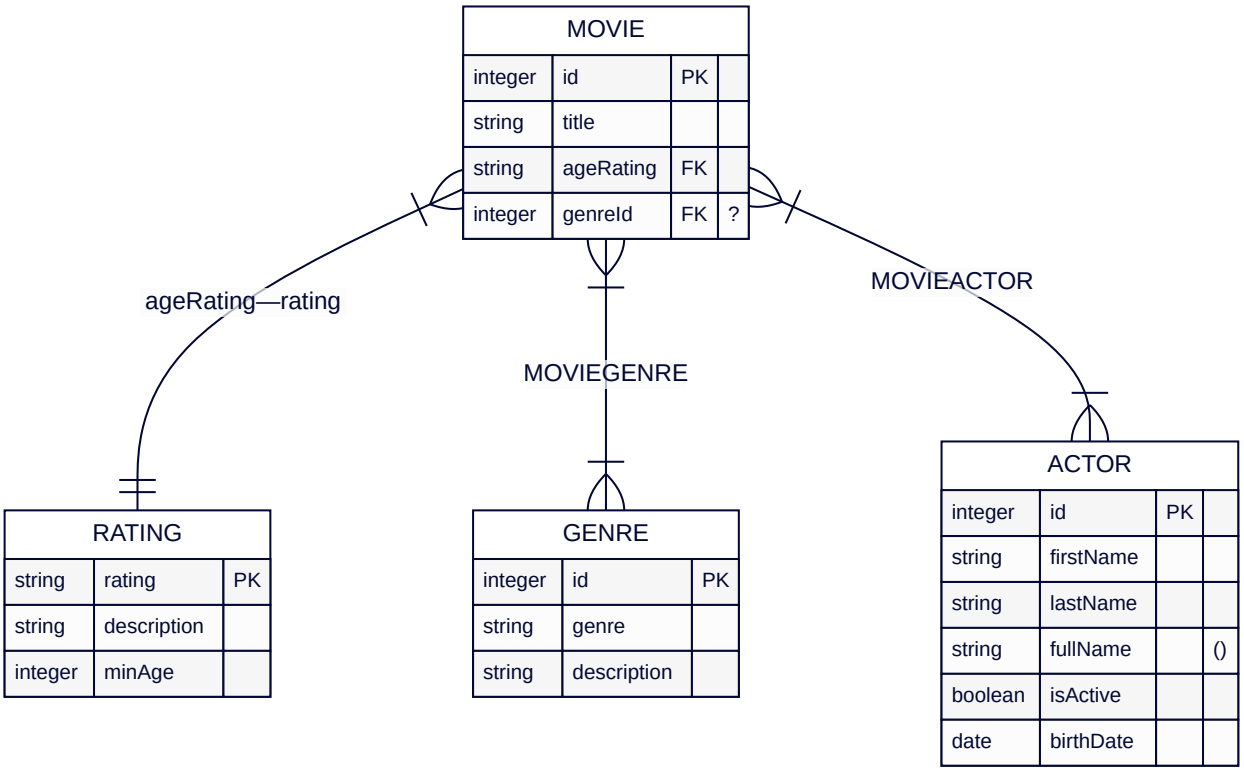
MOVIE(id, title, ageRating)
RATING(rating, description, minAge)
GENRE(id, genre, description)
ACTOR(id, firstName, lastName, isActive, birthDate)
MOVIEACTOR(movieId, actorId)
MOVIEGENRE(movieId, genreId)

1. I've also another relational table, MOVIEGENRE for the many-to-many relation

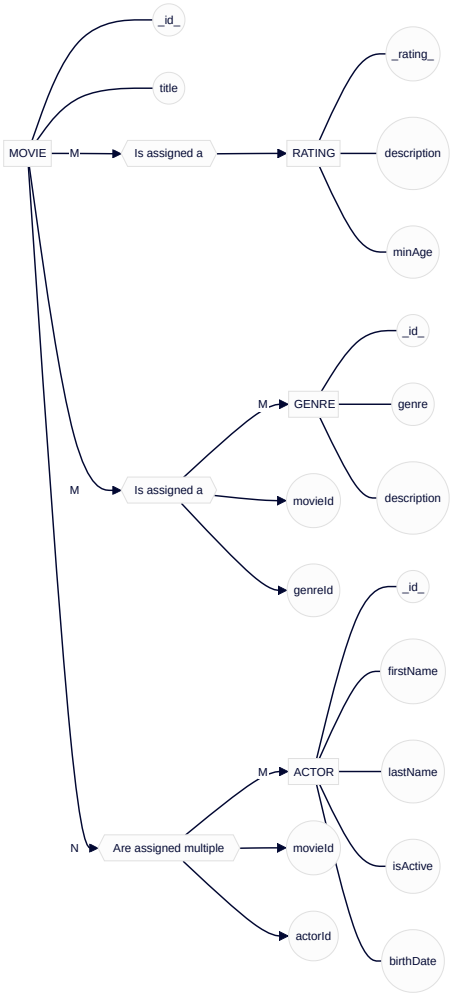
ii


ACTOR
isActive
firstName
lastName
birthDate
_id_
MOVIEACTOR
actorId
movieId
MOVIE
title
ageRating
_id_
RATING
rating
description
minAge
MOVIEGENRE
movieId
GenreId
GENRE
_id_
genre
description

iii


MOVIE
integer id PK
string title
string ageRating FK
integer genreId FK ?
ageRating—rating
MOVIEGENRE
MOVIEACTOR
RATING
string rating PK
string description
integer minAge
GENRE
integer id PK
string genre
string description
ACTOR
integer id PK
string firstName
string lastName
string fullName ()
boolean isActive
date birthDate

iv



V

MOVIES		
id	title	ageRating
1	Lost	G
2	Found	R

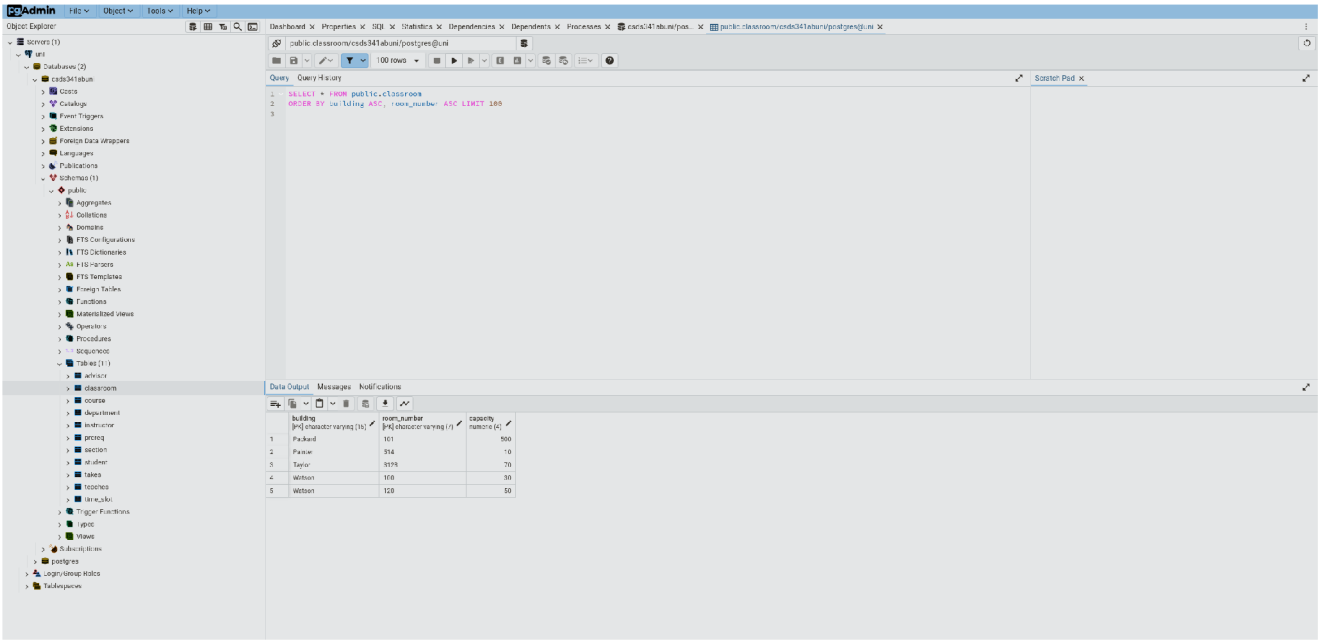
RATING		
rating	description	minAge
G	General	0
R	Restricted	18

GENRE		
id	genre	description
1	rom	Romance
2	sci-fi	Science Fiction

ACTOR				
id	firstName	lastName	isActive	birthDate
1	John	Doe	true	1950-01-01

ACTOR				
2	Johnny	Dohe	true	1953-03-01
3	Jon	Do	true	1959-02-01
4	Jhon	Dhoe	false	1955-08-01
5	Joe	Dohn	true	1952-04-01

MOVIEACTOR	
movieId	actorId
1	1
1	2
1	3
2	2
2	4
2	5

MOVIEGENRE	
movieId	actorId
1	1
1	2
2	2

4