

CSDS 341 HW 1

1

<i>CK</i>	<i>SK</i>	<i>K</i>	\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

MOVIE(id, title, ageRating, genreId, actors)

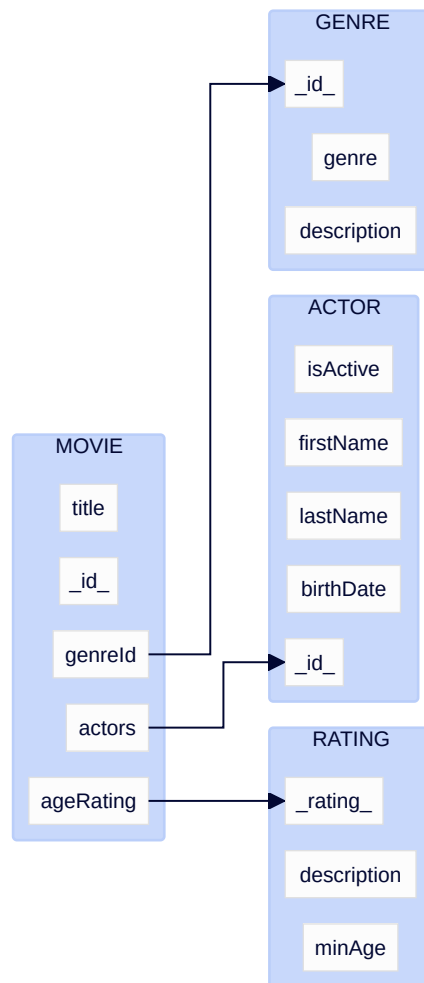
RATING(rating, description, minAge)

GENRE(id, genre, description)

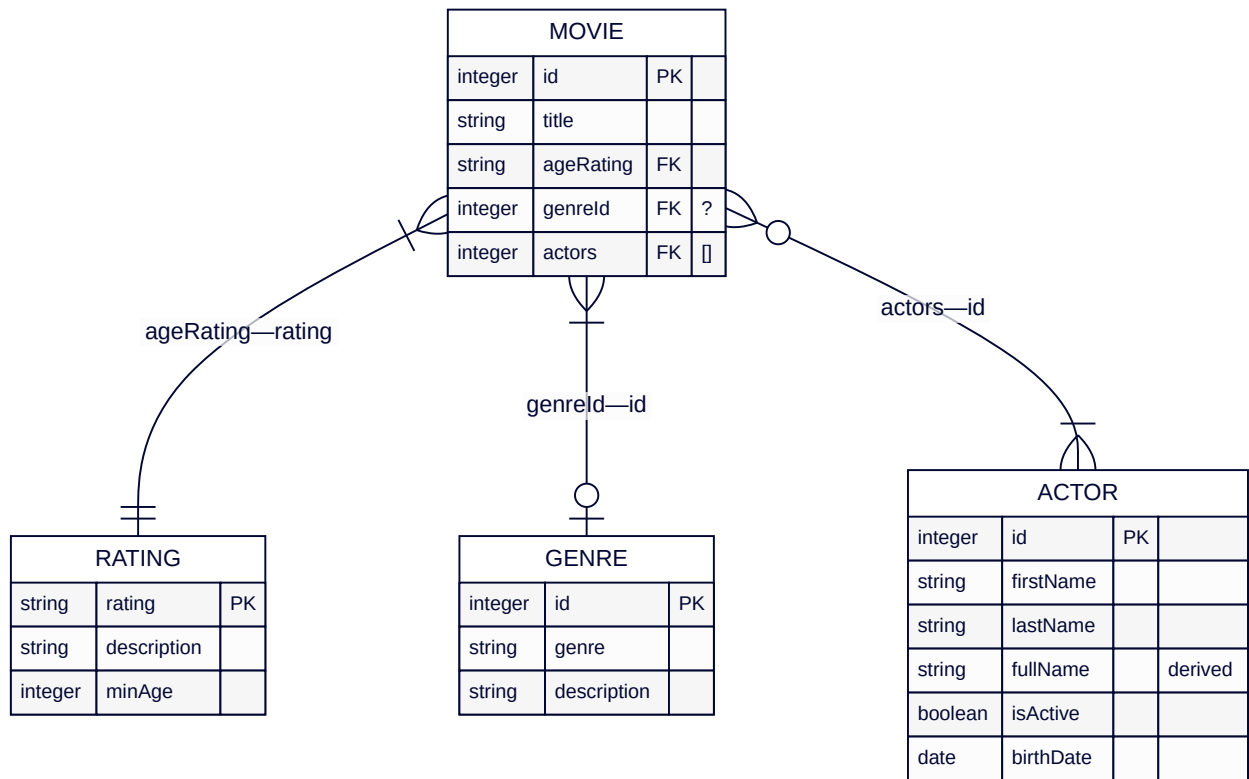
ACTOR(id, firstName, lastName, isActive, birthDate)

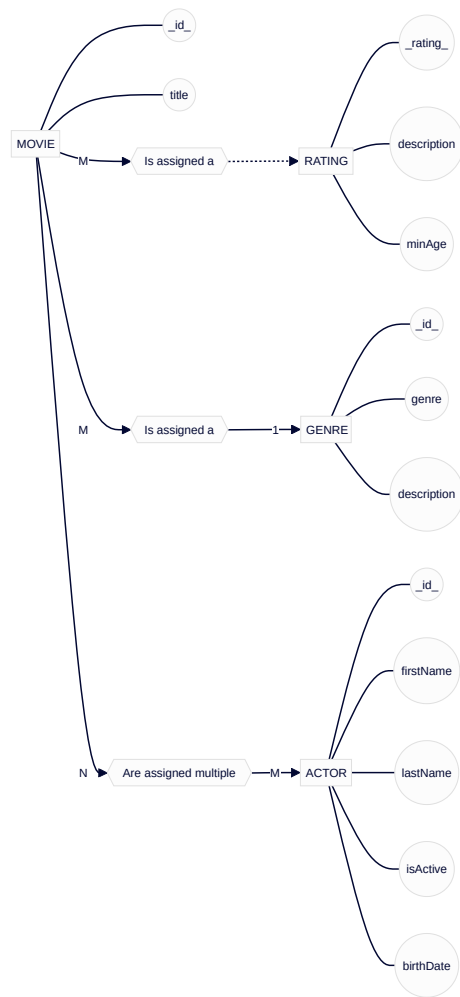
1. I've renamed a few of the attributes, especially the id's in order to have a more consistent and clear description of the model
2. I changed the type of status from string to boolean as there is only two possible values and no point of storing a string
3. I split the name attribute into first and 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

ii



iii





V

MOVIES				
id	title	ageRating	genreId	actors
1	Lost	G	1	[1,2,3]
2	Found	R	2	[2,4,5]

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

b

i

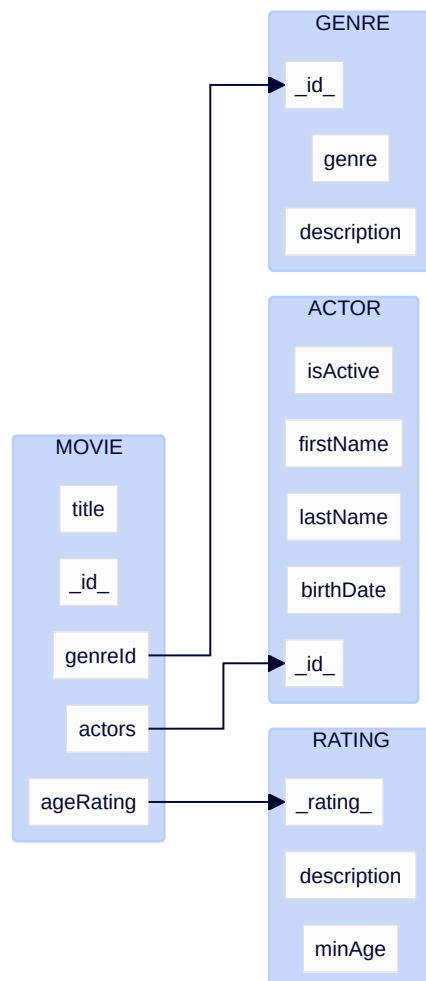
MOVIE(id, title, ageRating, genreId, actors)

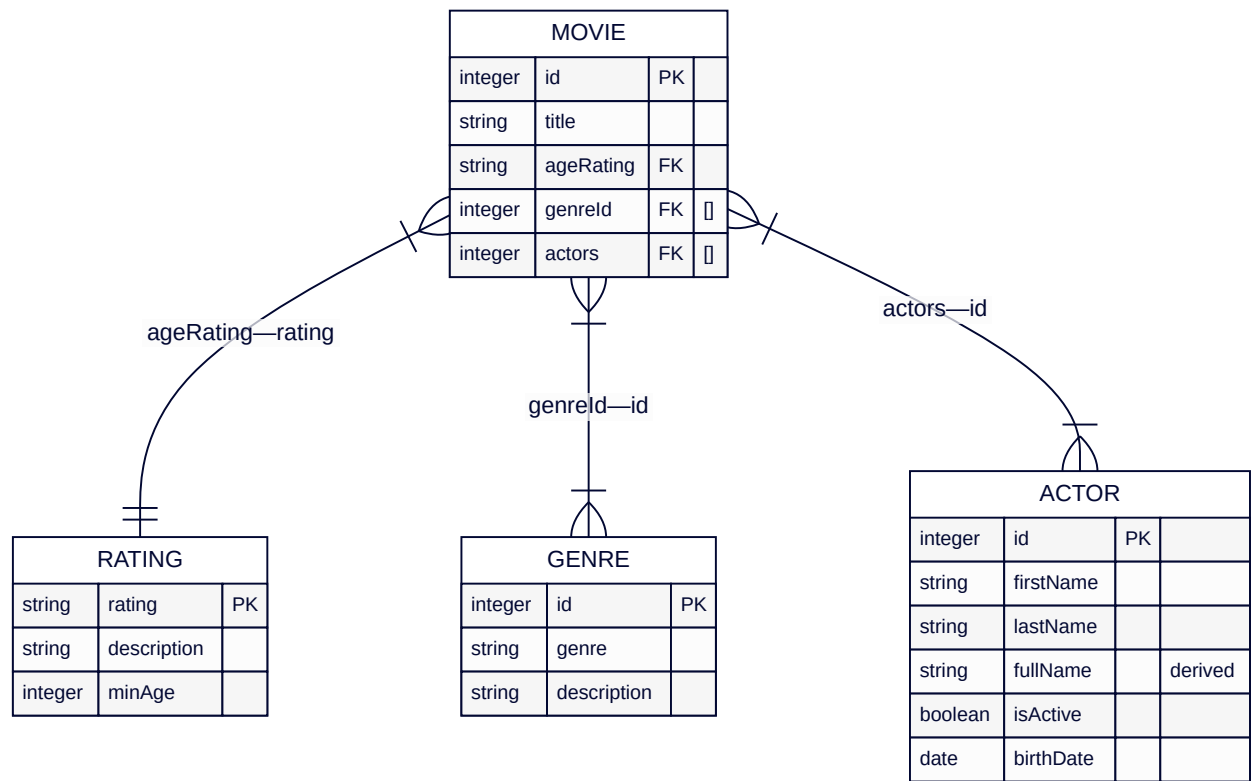
RATING(rating, description, minAge)

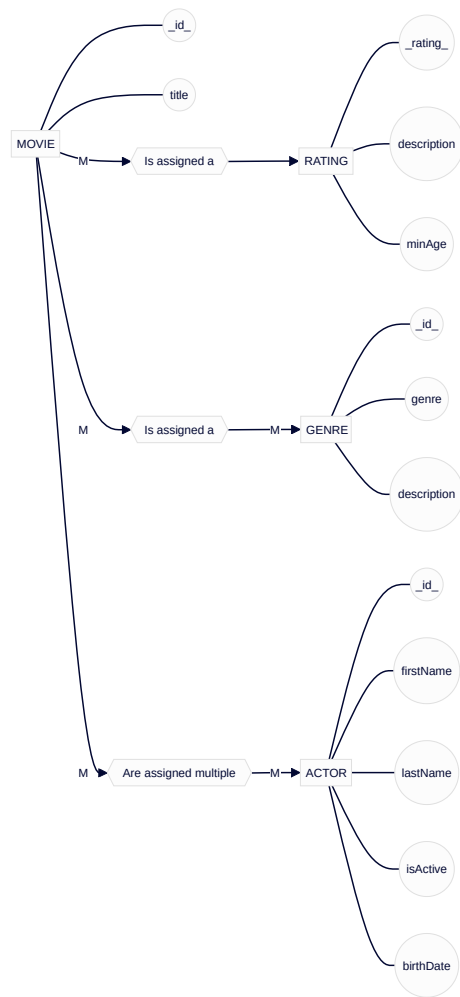
GENRE(id, genre, description)

ACTOR(id, firstName, lastName, isActive, birthDate)

ii







V

MOVIES				
id	title	ageRating	genreId	actors
1	Lost	G	[1,2]	[1,2,3]
2	Found	R	[2]	[2,4,5]

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

4

psAdmin

File

Object

Tools

Help

Object Explorer

Servers (1)

cds341abun

cds341abun

catalog

event triggers

extensions

foreign data wrappers

languages

publications

schemas (1)

public

aggregates

collations

domains

fts configurations

fts dictionaries

fts parsers

fts templates

foreign tables

functions

materialized views

operators

procedures

sequences

tables (11)

advisor

classroom

course

department

instructor

inver

section

student

timeslot

triggers

trigger functions

types

views

subscriptions

postgres

login/group roles

tablespaces

Dashboard

Properties

SQL

Statistics

Dependencies

Dependents

Processes

cds341abun/pos

public-classroom/cds341abun/postgres@uni

Query

Query History

1

2

3

SELECT

FROM public.classroom

ORDER BY building ASC, room_number ASC LIMIT 100

Data Output

Messages

Notifications

building

room_number

capacity

1

2

3

4

5

Pickard

101

500

Parmer

514

15

Taylor

3128

70

Watson

100

30

Watson

120

50