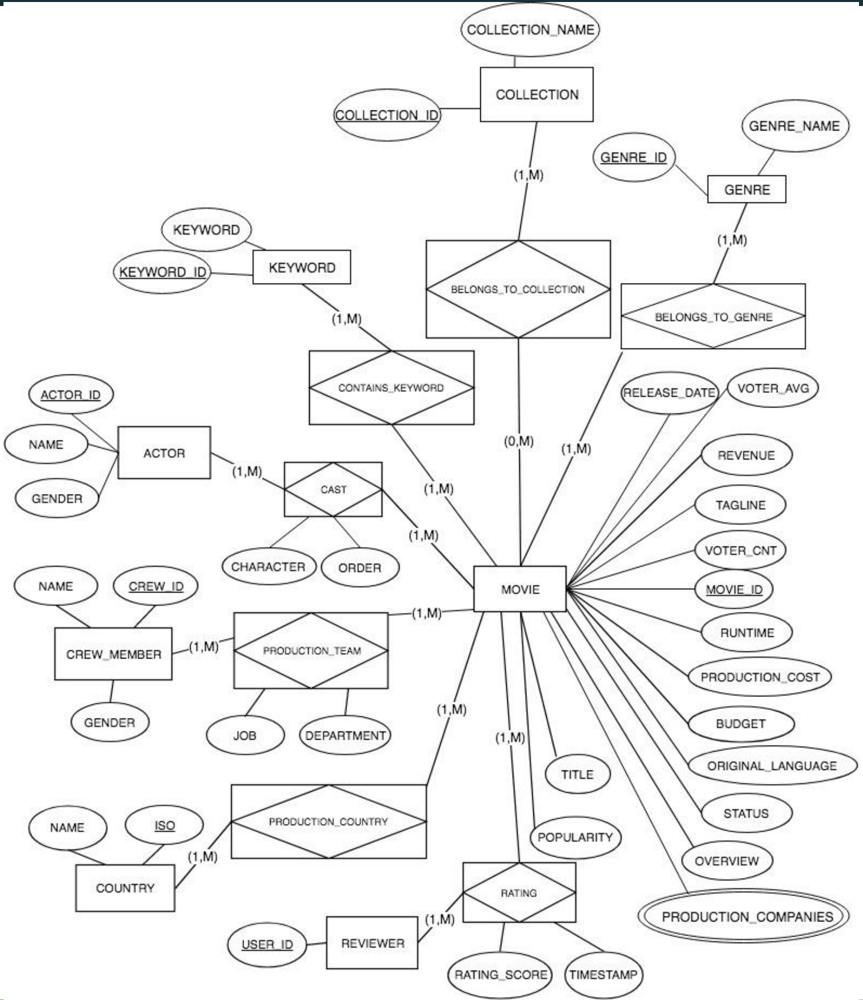
Movies Database

Isis Ramirez
Siri Manjunath
Xiaoye Zhang

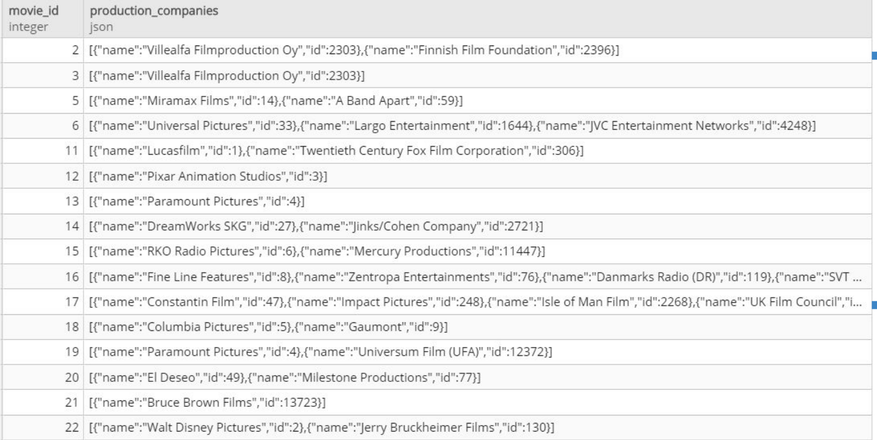
Database Introduction

- CSV format:
  - Movie
  - Cast
  - Crew
  - Ratings
- PostgreSQL 11
  - Several M:M relations

ERD


COLLECTION_NAME
COLLECTION
COLLECTION_ID
GENRE_NAME
GENRE_ID
GENRE
(1,M)
(1,M)
KEYWORD
KEYWORD_ID
KEYWORD
BELONGS_TO_COLLECTION
BELONGS_TO_GENRE
(1,M)
CONTAINS_KEYWORD
ACTOR_ID
NAME
ACTOR
GENDER
(1,M)
CAST
(1,M)
(0,M)
(1,M)
(1,M)
CHARACTER
ORDER
RELEASE_DATE
VOTER_AVG
REVENUE
TAGLINE
VOTER_CNT
MOVIE_ID
MOVIE
RUNTIME
PRODUCTION_COST
BUDGET
ORIGINAL_LANGUAGE
STATUS
OVERVIEW
PRODUCTION_COMPANIES
NAME
CREW_ID
CREW_MEMBER
(1,M)
PRODUCTION_TEAM
(1,M)
GENDER
JOB
DEPARTMENT
(1,M)
(1,M)
TITLE
POPULARITY
NAME
ISO
COUNTRY
(1,M)
PRODUCTION_COUNTRY
RATING
(1,M)
USER_ID
REVIEWER
RATING_SCORE
TIMESTAMP

Multi-valued JSON Data



movie_id	production_companies
integer	json
2	[{"name":"Villealfa Filmproduction Oy","id":2303},{"name":"Finnish Film Foundation","id":2396}]
3	[{"name":"Villealfa Filmproduction Oy","id":2303}]
5	[{"name":"Miramax Films","id":14},{"name":"A Band Apart","id":59}]
6	[{"name":"Universal Pictures","id":33},{"name":"Largo Entertainment","id":1644},{"name":"JVC Entertainment Networks","id":4248}]
11	[{"name":"Lucasfilm","id":1},{"name":"Twentieth Century Fox Film Corporation","id":306}]
12	[{"name":"Pixar Animation Studios","id":3}]
13	[{"name":"Paramount Pictures","id":4}]
14	[{"name":"DreamWorks SKG","id":27},{"name":"Jinks/Cohen Company","id":2721}]
15	[{"name":"RKO Radio Pictures","id":6},{"name":"Mercury Productions","id":11447}]
16	[{"name":"Fine Line Features","id":8},{"name":"Zentropa Entertainments","id":76},{"name":"Danmarks Radio (DR)","id":119},{"name":"SVT ...
17	[{"name":"Constantin Film","id":47},{"name":"Impact Pictures","id":248},{"name":"Isle of Man Film","id":2268},{"name":"UK Film Council","i...
18	[{"name":"Columbia Pictures","id":5},{"name":"Gaumont","id":9}]
19	[{"name":"Paramount Pictures","id":4},{"name":"Universum Film (UFA)","id":12372}]
20	[{"name":"El Deseo","id":49},{"name":"Milestone Productions","id":77}]
21	[{"name":"Bruce Brown Films","id":13723}]
22	[{"name":"Walt Disney Pictures","id":2},{"name":"Jerry Bruckheimer Films","id":130}]

```
[{"name": "Villealfa Filmproduction Oy", "id":
2303}, {"name": "Finnish Film Foundation", "id":
2396}]
```

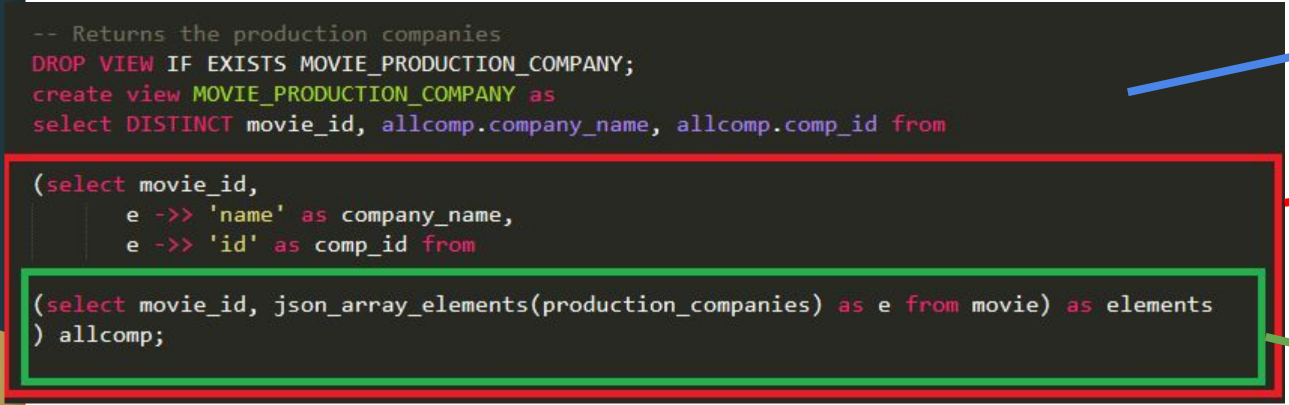
□

```
[{"name": "Constantin Film", "id": 47}, {"name":
"Impact Pictures", "id": 248}, {"name": "Isle of
Man Film", "id": 2268}, ...]
```

□

MOVIE_PRODUCTION_COMPANY

Production Company View



```
-- Returns the production companies
DROP VIEW IF EXISTS MOVIE_PRODUCTION_COMPANY;
create view MOVIE_PRODUCTION_COMPANY as
select DISTINCT movie_id, allcomp.company_name, allcomp.comp_id from

(select movie_id,
        e ->> 'name' as company_name,
        e ->> 'id' as comp_id from

(select movie_id, json_array_elements(production_companies) as e from movie) as elements
) allcomp;
```

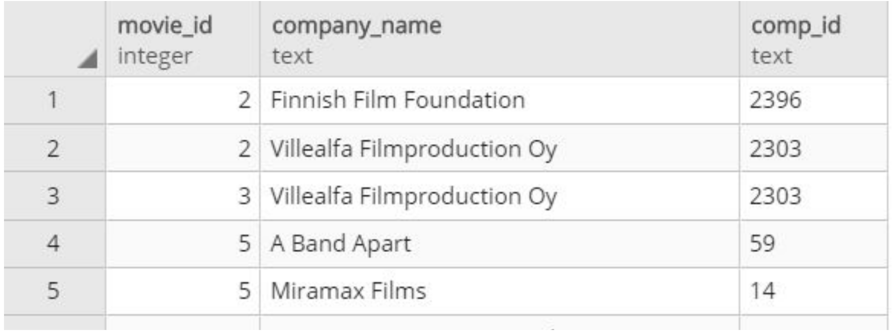
SELECT DISTINCT MOVIE - COMPANY PAIRS

SELECT DESIRED JSON FIELDS

->>

EXTRACT JSON OBJECTS FROM ARRAYS

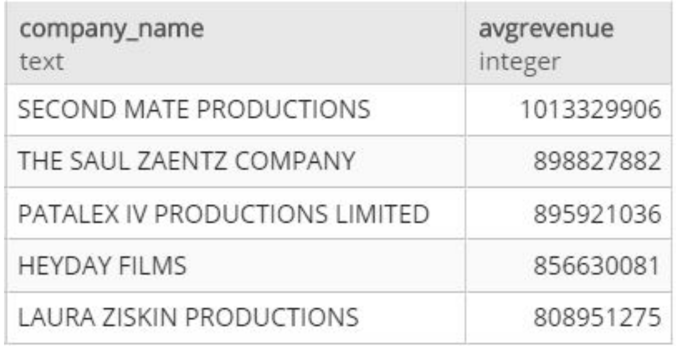
JSON_ARRAY_ELEMENTS()



	movie_id integer	company_name text	comp_id text
1	2	Finnish Film Foundation	2396
2	2	Villealfa Filmproduction Oy	2303
3	3	Villealfa Filmproduction Oy	2303
4	5	A Band Apart	59
5	5	Miramax Films	14

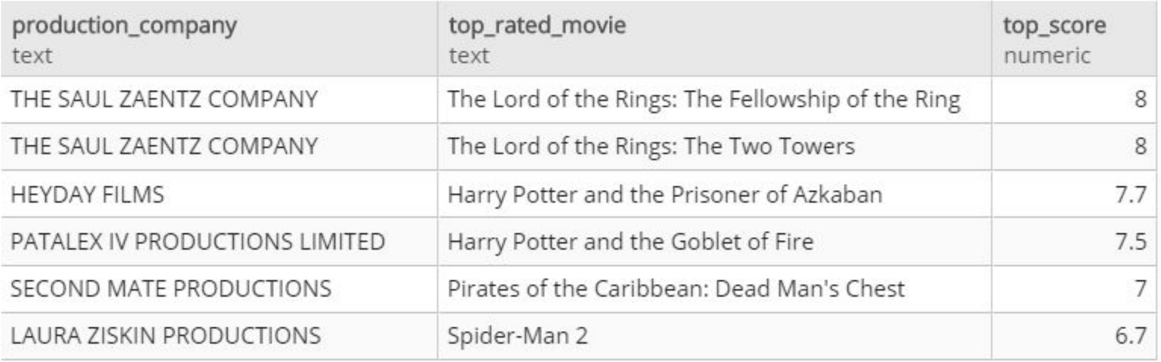
Production Company Analysis

HIGHEST AVG REVENUE (TOP 5)



company_name	avgrevenue
text	integer
SECOND MATE PRODUCTIONS	1013329906
THE SAUL ZAENTZ COMPANY	898827882
PATALEX IV PRODUCTIONS LIMITED	895921036
HEYDAY FILMS	856630081
LAURA ZISKIN PRODUCTIONS	808951275

TOP RATED MOVIES FOR PREVIOUS COMPANIES



production_company	top_rated_movie	top_score
text	text	numeric
THE SAUL ZAENTZ COMPANY	The Lord of the Rings: The Fellowship of the Ring	8
THE SAUL ZAENTZ COMPANY	The Lord of the Rings: The Two Towers	8
HEYDAY FILMS	Harry Potter and the Prisoner of Azkaban	7.7
PATALEX IV PRODUCTIONS LIMITED	Harry Potter and the Goblet of Fire	7.5
SECOND MATE PRODUCTIONS	Pirates of the Caribbean: Dead Man's Chest	7
LAURA ZISKIN PRODUCTIONS	Spider-Man 2	6.7

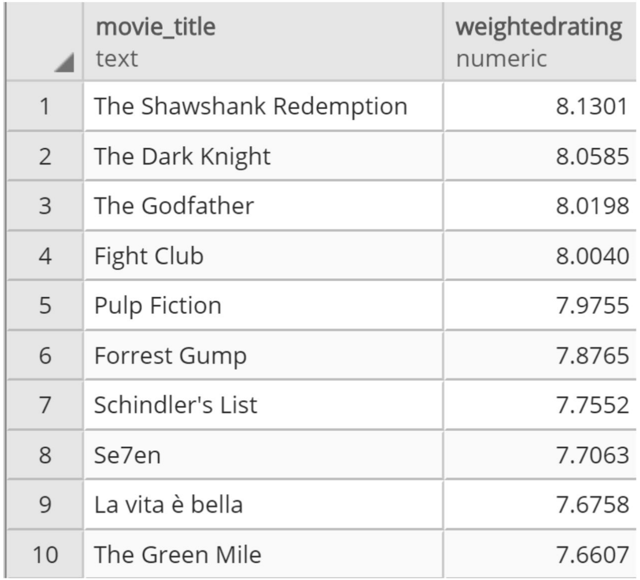
Weighted Rating Recommender

$$\text{Weighted Rating} = \left(\frac{v}{v+m} * \mathrm{R}\right) + \left(\frac{m}{v+m} * C\right)$$

- v - the number of votes for the movie
- m - the minimum votes required to be considered
- R - the average rating score of the movie
- C - the mean vote rating across whole dataset

For m, we want to filter 10% movies that have insufficient vote counts and keep the remaining 90% movies as recommendation pool. According to our dataset, we set m to 1800.

```
select * from WR_Recommender('Léon',
1800, 'genre') limit 10;
```

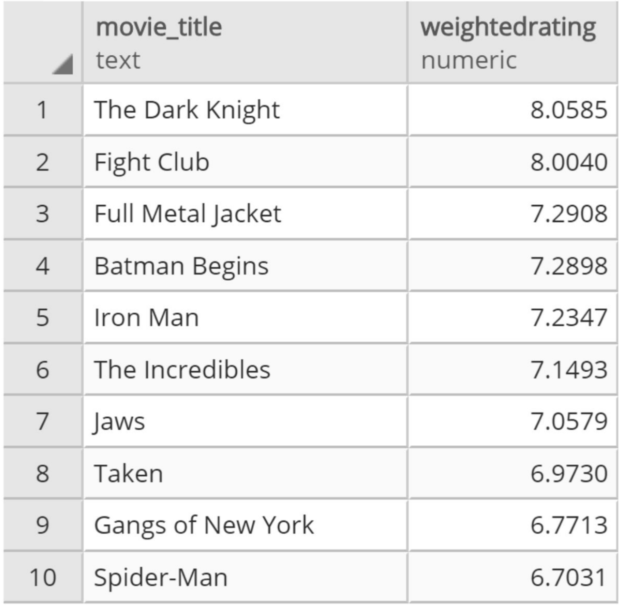


	movie_title text	weightedrating numeric
1	The Shawshank Redemption	8.1301
2	The Dark Knight	8.0585
3	The Godfather	8.0198
4	Fight Club	8.0040
5	Pulp Fiction	7.9755
6	Forrest Gump	7.8765
7	Schindler's List	7.7552
8	Se7en	7.7063
9	La vita è bella	7.6758
10	The Green Mile	7.6607

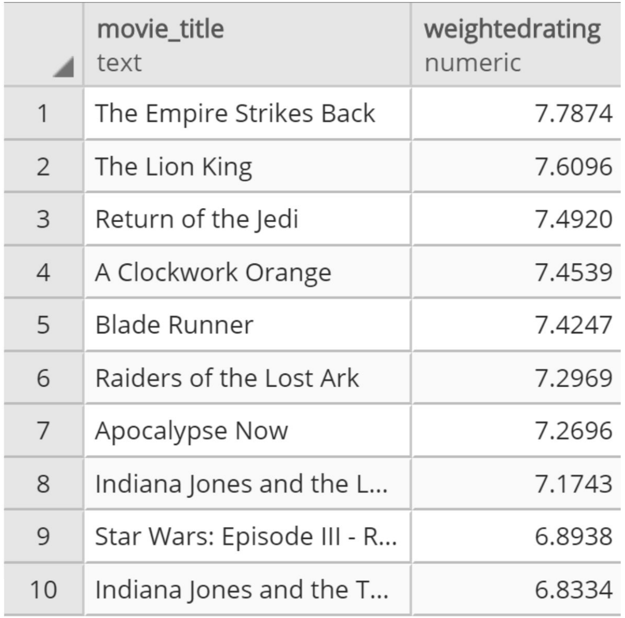
Weighted Rating Recommender

```
select * from WR_Recommender('Batman',
1800, 'keyword') limit 10;
```

```
select * from WR_Recommender('Star
Wars', 1800, 'cast') limit 10;
```



	movie_title	weightedrating
	text	numeric
1	The Dark Knight	8.0585
2	Fight Club	8.0040
3	Full Metal Jacket	7.2908
4	Batman Begins	7.2898
5	Iron Man	7.2347
6	The Incredibles	7.1493
7	Jaws	7.0579
8	Taken	6.9730
9	Gangs of New York	6.7713
10	Spider-Man	6.7031



	movie_title	weightedrating
	text	numeric
1	The Empire Strikes Back	7.7874
2	The Lion King	7.6096
3	Return of the Jedi	7.4920
4	A Clockwork Orange	7.4539
5	Blade Runner	7.4247
6	Raiders of the Lost Ark	7.2969
7	Apocalypse Now	7.2696
8	Indiana Jones and the L...	7.1743
9	Star Wars: Episode III - R...	6.8938
10	Indiana Jones and the T...	6.8334

Jaccard Movie Recommender

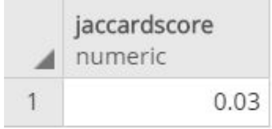
$$Jaccard\ Score = \frac{|(K, G, D, A)_{movie\ 1} \cap (K, G, D, A)_{movie\ 2}|}{|(K, G, D, A)_{movie\ 1} \cup (K, G, D, A)_{movie\ 2}|}$$

K - Keywords
G - Genres
D - Directors
A - Actors

Jaccard Movie Recommender

```
select * from JaccardScore(862, 8587);
```

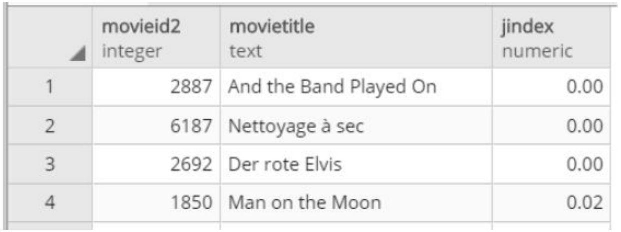
Return jaccard score for 2 specified movies



	jaccardscore numeric
1	0.03

```
select * from JaccardTable(862);
```

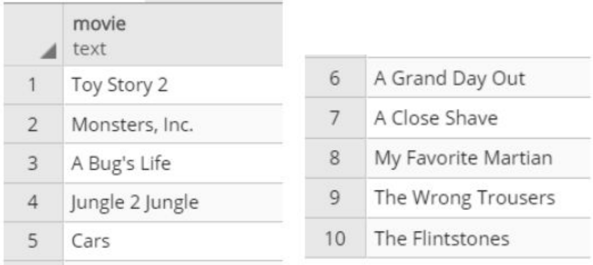
Use loop and call JaccardScore() to retrieve jaccard scores



	movieid2 integer	movietitle text	jindex numeric
1	2887	And the Band Played On	0.00
2	6187	Nettoyage à sec	0.00
3	2692	Der rote Elvis	0.00
4	1850	Man on the Moon	0.02

```
select * from JacRecommend(862, 'Toy Story');
```

Return top 10 movies from JaccardTable() result



	movie text
1	Toy Story 2
2	Monsters, Inc.
3	A Bug's Life
4	Jungle 2 Jungle
5	Cars
6	A Grand Day Out
7	A Close Shave
8	My Favorite Martian
9	The Wrong Trousers
10	The Flintstones

Comparison between 2 Recommenders

	Weighted Rating	Jaccard Score
Runtime	short	long
Scope	1	4
Accuracy	medium	high

Conclusion

PostgreSQL has pros and cons.

- Pro: **useful JSON support**, ability to handle several many-to-many relations
- Con: It was challenging to implement **complex mathematical operations** and the execution time was poor. It would be easier and more time efficient to use a tool such as Python for movie recommendation systems.

Q&A