The background features a large, abstract graphic composed of overlapping green triangles of varying shades, creating a sense of depth and motion.

# DATA EXPLORATION OF PUBLIC FILMS

By

## Ayomide Akinrotimi

## Servers (2)

local\_db

## Databases (3)

coffee

film

Casts

Catalogs

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 (4)

films

people

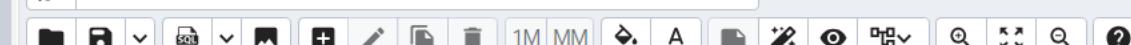
reviews

roles

Trigger Functions

Types

## film/postgres@local\_db



	public
	films
	id integer
	title character varying
	release_year integer
	country character varying
	duration integer
	language character varying
	certification character varying
	gross bigint
	budget bigint

	public
	people
	id integer
	name character varying
	birthdate date
	deathdate date

	public
	reviews
	id integer
	film_id integer
	num_user integer
	num_critic integer
	imdb_score real
	num_votes integer
	facebook_likes integer

Untitled\*

Servers (2)

local\_db

Databases (3)

- > coffee
- > film
- > Casts
- > Catalogs
- > 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 (4)

- > films
- > people
- > reviews

film/postgres@PostgreSQL 16

No limit

Query History

```
1 --Connect to the "film" database using PostgreSQL and PgAdmin
2 select *
3 from films
4
5
```

Scratch Pad X

Data Output Messages Notifications



	<b>id</b> [PK] integer	<b>title</b> character varying	<b>release_year</b> integer	<b>country</b> character varying	<b>duration</b> integer	<b>language</b> character varying	<b>certification</b> character varying	<b>gross</b> bigint
1	1	Intolerance: Love's Struggle Throughout the Ages	1916	USA	123	[null]	Not Rated	[null]
2	2	Over the Hill to the Poorhouse	1920	USA	110	[null]	[null]	3000000
3	3	The Big Parade	1925	USA	151	[null]	Not Rated	[null]
4	4	Metropolis	1927	Germany	145	German	Not Rated	26435
5	5	Pandora's Box	1929	Germany	110	German	Not Rated	9950
6	6	The Broadway Melody	1929	USA	100	English	Passed	2808000
7	7	Hell's Angels	1930	USA	96	English	Passed	[null]
8	8	A Farewell to Arms	1932	USA	79	English	Unrated	[null]

Total rows: 1000 of 4968 Query complete 00:00:00.207

Ln 3, Col 11

## Servers

local\_db

## Databases (3)

coffee

film

Casts

Catalogs

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 (4)

films

people

reviews

roles

Trigger Functions

Types

film/postgres@PostgreSQL 16



No limit

Query

```
1 --Question 2
2 --What are the top 10 highest-grossing films in the database, and when were they released?
3
4 select title,gross-budget as profit,release_year
5 from films
6 where gross-budget is not null
7 order by gross-budget DESC
8 limit 10
9
```

Scratch Pad x

Data Output

Messages

Notifications



	title character varying	profit bigint	release_year integer
1	Star Wars: Episode VII - The Force Awakens	691627416	2015
2	Avatar	523505847	2009
3	Jurassic World	502177271	2015
4	Titanic	458672302	1997
5	Star Wars: Episode IV - A New Hope	449935665	1977
6	E.T. the Extra-Terrestrial	424449459	1982
7	The Avengers	403279547	2012
8	The Avengers	403279547	2012
9	The Lion King	377783777	1994
10	Star Wars: Episode I - The Phantom Menace	359544677	1999

Total rows: 10 of 10

Query complete 00:00:00.099

Ln 8, Col 9

Servers (2)

local\_db

Databases (3)

- coffee
- film
  - Casts
  - Catalogs
  - 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 (4)
    - films
    - people
    - reviews

film/postgres@PostgreSQL 16

No limit

## Query

Query History

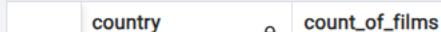
```
1 --Question 3
2 --How many films in the database were released in each country,
3 --and what are the top five countries?
4
5 select country, count(title) as count_of_films
6 from films
7 where country is not null
8 group by country
9 order by count(title) desc
10 limit 5
```

Scratch Pad

## Data Output

Messages

Notifications



	country character varying	count_of_films bigint
1	USA	3750
2	UK	443
3	France	153
4	Canada	123
5	Germany	97

Total rows: 5 of 5

Query complete 00:00:00.133

Ln 5, Col 42

Servers (2)

local\_db

Databases (3)

- coffee
- film
  - Casts
  - Catalogs
  - 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 (4)
    - films
    - people
    - reviews

film/postgres@PostgreSQL 16

 No limit 

Query Query History

```
1 --Question 4
2 --How many films are available in each language,
3 --and what are the top three languages represented?
4
5 select distinct(language),count(title)
6 from films
7 where language is not null
8 group by distinct(language)
9 order by count(title) desc
10 limit 3
```

Scratch Pad X

Data Output Messages Notifications



	language character varying	count bigint
1	English	4635
2	French	72
3	Spanish	40

Total rows: 3 of 3 | Query complete 00:00:00.123

Ln 10, Col 8

Servers (2)

local\_db

Databases (3)

- > coffee
- > film
  - > Casts
  - > Catalogs
  - > 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 (4)
    - > films
    - > people
    - > reviews

film/postgres@PostgreSQL 16

No limit

Query Query History

```
1 --Question 5
2 --What is the average IMdb score for films in the database?
3
4 select avg(imdb_score) as average_imdb_score
5 from reviews
```

Scratch Pad X

Data Output Messages Notifications

	average_imdb_score	double precision
1	6.42648953616907	

Total rows: 1 of 1 Query complete 00:00:00.070

Ln 4, Col 45

Servers (2)

- local\_db
- Databases (3)
  - coffee
  - film
  - Scratch Pad

Query History

- Question 6
- Which country has made the highest profit from movies?
- [Tip: To avoid a biased result, use average instead of sum.]
- select country , sum(gross-budget) as profit
- from films
- where "gross" is not null and "budget" is not null
- group by country
- order by sum(gross-budget) desc
- limit 10

Schemas (1)

- public
  - Aggregates
  - Collations
  - Domains
  - FTS Configurations
  - FTS Dictionaries
  - FTS Parsers
  - FTS Templates
  - Foreign Tables
  - Functions
  - Materialized Views
  - Operators
  - Procedures
  - Sequences
  - Tables (4)
    - films
    - people
    - reviews
    - roles
  - Trigger Functions
  - Types

film/postgres@PostgreSQL 16

## Query

```
1 --Question 6
2 --Which country has made the highest profit from movies?
3 --[Tip: To avoid a biased result, use average instead of sum.]
4
5 select country , sum(gross-budget) as profit
6 from films
7 where "gross" is not null and "budget" is not null
8 group by country
9 order by sum(gross-budget) desc
10 limit 10
```

## Scratch Pad

## Data Output



	country character varying	profit numeric
1	USA	53748646579
2	UK	1555276582
3	New Zealand	245159794
4	Canada	206987067
5	Australia	205335929
6	South Africa	103740241
7	Taiwan	98681364
8	Mexico	22115665
9	Argentina	16792809
10	Peru	12362581

Total rows: 10 of 10 | Query complete 00:00:00.101

Ln 10, Col 9

Servers (2)  
local\_db  
Databases (3)  
coffee  
film (selected)  
Casts  
Catalogs  
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 (4)  
films  
people  
reviews  
roles  
Trigger Functions  
Types

film/postgres@PostgreSQL 16

Query History

```
1 --Question 7
2 --Which movie made the highest profit in the 21st century
3
4
5 SELECT title,(gross - budget) AS profit, release_year
6 FROM films
7 WHERE (gross - budget) IS NOT NULL AND release_year >=2001
8 ORDER BY profit DESC
9 limit 10
```

Scratch Pad

Data Output Messages Notifications



	title character varying	profit bigint	release_year integer
1	Star Wars: Episode VII - The Force Awakens	691627416	2015
2	Avatar	523505847	2009
3	Jurassic World	502177271	2015
4	The Avengers	403279547	2012
5	The Avengers	403279547	2012
6	The Dark Knight	348316061	2008
7	The Hunger Games	329999255	2012
8	Deadpool	305024263	2016
9	The Hunger Games: Catching Fire	294645577	2013
10	Despicable Me 2	292049635	2013

Total rows: 10 of 10 Query complete 00:00:00.157

Ln 9, Col 9



Servers (2)

local\_db

Databases (3)

coffee

film



Casts



Catalogs



Event Triggers



Extensions



Foreign Data Wrappers



Languages



Publications



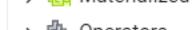
Schemas (1)



Aggregates



Collations



Domains



FTS Configurations



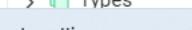
FTS Dictionaries



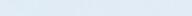
FTS Parsers



FTS Templates



Foreign Tables



Functions



Materialized Views



Operators



Procedures



Sequences



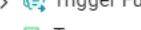
Tables (4)



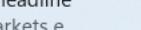
films



people



reviews



roles



Trigger Functions



Types

film/postgres@PostgreSQL 16



Query History

Scratch Pad x

```
1 --Question 8
2 --How many people in the database are still alive (based on birthdate and death date)?
3
4 select count(birthdate)-count(deathdate) as living_cast
5 from people
```

Data Output Messages Notifications



living\_cast

bigint

1

5365

Total rows: 1 of 1 Query complete 00:00:00.108

Ln 4, Col 51





Servers (2)

local\_db

Databases (3)

coffee

film



Catalogs



Extensions

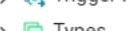
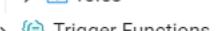
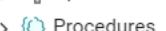
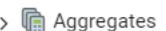


Languages



Schemas (1)

public



No limit



Query

Query History

Scratch Pad x

```
1 --Question 9
2 --Which year has the highest number of movie releases?
3
4 select release_year, count(title) as number_of_movies_released
5 from films
6 where release_year is not null
7 group by release_year
8 order by count(title) desc
```

Data Output

Messages

Notifications



	release_year integer	number_of_movies_released bigint
1	2009	260
2	2014	252
3	2006	238
4	2013	236
5	2010	227
6	2015	226
7	2008	225
8	2011	224
9	2005	221
10	2012	220

Total rows: 91 of 91 Query complete 00:00:00.089

Ln 8, Col 27

Servers (2)

local\_db

Databases (3)

coffee

film

- > Casts
- > Catalogs
- > 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 (4)
  - > films
  - > people
  - > reviews
  - > roles
- > Trigger Functions
- > Types

film/postgres@PostgreSQL 16

v

No limit

Query

```
1 --Question 10
2 --Determine the top 10 people with the most roles in the database.
3
4
5 SELECT COUNT(role) as number_of_roles, person_id
6 FROM people AS p
7 inner JOIN roles AS r
8 ON p.id = r.person_id
9 GROUP BY person_id
10 ORDER BY COUNT(role) DESC
11 LIMIT 10
12
```

Scratch Pad x

c

Data Output Messages Notifications

	number_of_roles bigint	person_id integer
1	54	6764
2	47	5777
3	41	4022
4	40	1053
5	39	7492
6	38	5317
7	36	1500
8	34	4832
9	34	910
10	34	2025

Total rows: 10 of 10 Query complete 00:00:00.108

Ln 10, Col 20

Servers (2)

local\_db

Databases (3)

coffee

film

- > Casts
- > Catalogs
- > 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 (4)
  - > films
  - > people
  - > reviews
  - > roles

- > Trigger Functions
- > Types

film/postgres@PostgreSQL 16

v

No limit

Query

Query History

Scratch Pad x

```
1 --Question 11
2 --Who are the top 10 actors or directors with the most roles in the database
3
4
5 SELECT COUNT(role) as number_of_role, name
6 FROM people AS p
7 inner JOIN roles AS r
8 ON p.id = r.person_id
9 GROUP BY name
10 ORDER BY COUNT(role) DESC
11 LIMIT 10
12
13
```

Data Output

Messages

Notifications

	number_of_role	name
	bigint	character varying
1	54	Robert De Niro
2	47	Morgan Freeman
3	41	Johnny Depp
4	40	Bruce Willis
5	39	Steve Buscemi
6	38	Matt Damon
7	36	Clint Eastwood
8	34	Bill Murray
9	34	Liam Neeson
10	34	Nicolas Cage

Total rows: 10 of 10 Query complete 00:00:00.142

Ln 10, Col 20

## Servers

local\_db

## Databases (3)

- > coffee
- > film
  - > Casts
  - > Catalogs
  - > 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 (4)
  - > films
  - > people
  - > reviews
  - > roles
- > Trigger Functions
- > Types

## film/postgres@PostgreSQL 16

▼



No limit

▼



## Query Query History

## Scratch Pad x

```
1 --Question 12
2 --Identify how many people in the database are still alive.
3
4 select count(birthdate)- count(deathdate) as living_cast
5 from people
6 where "birthdate" is not null
7
8
```

## Data Output Messages Notifications



living_cast	
	bigint
1	5371

Total rows: 1 of 1 Query complete 00:00:00.089

Ln 1, Col 14

## Servers

local\_db

## Databases (3)

coffee

film

Casts

Catalogs

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 (4)

films

people

reviews

roles

Trigger Functions

Types

film/postgres@PostgreSQL 16



No limit

Query Query History

Scratch Pad x

```
1 --Question 13
2 --Calculate the average number of user and critic reviews for films
3
4 select avg(num_user) as average_number_of_users,avg(num_critic) as average_number_of_critic
5 from reviews
```

Data Output Messages Notifications

	average_number_of_users	average_number_of_critic
	numeric	numeric
1	275.6999393816932714	141.9138561560341325

Total rows: 1 of 1 Query complete 00:00:00.129

Ln 5, Col 13

Servers (2)

- local\_db
- Databases (3)
  - coffee
  - film
  - Scratch Pad
- 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 (4)
      - films
      - people
      - reviews
      - roles
    - Trigger Functions
    - Types

film/postgres@PostgreSQL 16

Query Query History

```
1 --Question 14a|  
2 --Identify films with the highest number of user and critic reviews  
3  
4  
5 SELECT title,film_id, num_user,num_critic  
6 FROM reviews  
7 inner JOIN films  
8 ON reviews.film_id=films.id  
9 WHERE num_user IS NOT NULL  
10 ORDER BY num_user DESC  
11  
12
```

Scratch Pad x

Data Output Messages Notifications



	title	film_id	num_user	num_critic
	character varying	integer	integer	integer
1	The Lord of the Rings: The Fellowship of the Ring	1667	5060	297
2	The Dark Knight	3110	4667	645
3	The Shawshank Redemption	742	4144	199
4	Star Wars: Episode VII - The Force Awakens	4746	4080	828
5	The Matrix	1310	3646	313
6	Star Wars: Episode I - The Phantom Menace	1281	3597	320
7	Star Wars: Episode II - Attack of the Clones	1847	3516	284
8	The Blair Witch Project	1296	3400	360
9	Star Wars: Episode III - Revenge of the Sith	2424	3286	359
10	The Lord of the Rings: The Return of the King	2045	3189	328

Total rows: 1000 of 4949 Query complete 00:00:00.158

Ln 1, Col 15

Servers (2)  
local\_db  
Databases (3)  
coffee  
film (selected)  
Casts  
Catalogs  
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 (4)  
films  
people  
reviews  
roles  
Trigger Functions  
Types

film/postgres@PostgreSQL 16

No limit

Query History

Scratch Pad

```
1 --Question 14b
2 --Identify films with the highest number of user and critic reviews
3
4
5 SELECT title,film_id, num_user,num_critic
6 FROM reviews
7 LEFT JOIN films
8 ON reviews.film_id=films.id
9 WHERE num_critic IS NOT NULL
10 ORDER BY num_critic DESC
11 limit 10
12
13
14
```

Data Output Messages Notifications

	title character varying	film_id integer	num_user integer	num_critic integer
1	Star Wars: Episode VII - The Force Awakens	4746	4080	828
2	The Dark Knight Rises	4049	2701	813
3	Prometheus	3999	2326	775
4	Django Unchained	3932	1193	765
5	Skyfall	4023	1498	750
6	Skyfall	4022	1498	750
7	Mad Max: Fury Road	4699	1588	739
8	Gravity	4185	1885	738
9	Man of Steel	4218	2536	733
10	Avatar	3203	3054	723

Total rows: 10 of 10 Query complete 00:00:00.138

Ln 11, Col 9

## Servers

## local\_db

## Databases (3)

&gt; coffee

## film

&gt; Casts

&gt; Catalogs

&gt; Event Triggers

&gt; Extensions

&gt; Foreign Data Wrappers

&gt; Languages

&gt; Publications

## Schemas (1)

## &gt; public

&gt; Aggregates

&gt; Collations

&gt; Domains

&gt; FTS Configurations

&gt; FTS Dictionaries

&gt; FTS Parsers

&gt; FTS Templates

&gt; Foreign Tables

&gt; Functions

&gt; Materialized Views

&gt; Operators

&gt; Procedures

&gt; Sequences

## &gt; Tables (4)

&gt; films

&gt; people

&gt; reviews

&gt; roles

&gt; Trigger Functions

&gt; Types

## film/postgres@PostgreSQL 16

## :

## :

## Query Query History

## Scratch Pad x

```
1 --Question 15a
2 --Which films have the most Facebook likes, and is there a correlation with their IMDb scores?
3 --this query shows the film with the highest number of facebook likes
4
5 select title,facebook_likes, imdb_score
6 from films
7 inner join reviews
8 on films.id=reviews.id
9 order by facebook_likes desc
10
```

## Data Output Messages Notifications



	title character varying	facebook_likes integer	imdb_score real
1	The Conformist	1364147	8.1
2	The Godfather	349000	8.6
3	Ernest & Celestine	199000	8.5
4	Train	197000	6.9
5	Casper	191000	8.1
6	Star Trek: Insurrection	190000	8.1
7	Goodfellas	175000	8.8
8	Any Given Sunday	166000	7.9
9	Fortress	165000	8.1
10	Waking Ned Devine	164000	8.5

Total rows: 1000 of 4968

Query complete 00:00:00.136

Ln 1, Col 15

Servers  
local\_db  
Databases (3)  
coffee  
film  
Casts  
Catalogs  
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 (4)  
films  
people  
reviews  
roles  
Trigger Functions  
Types

film/postgres@PostgreSQL 16

No limit

Query History

Scratch Pad

--Question 15b  
--Which films have the most Facebook likes, and is there a correlation with their IMDb scores?  
--this query shows the film with the highest number of imdb score

```
1 --Question 15b
2 --Which films have the most Facebook likes, and is there a correlation with their IMDb scores?
3 --this query shows the film with the highest number of imdb score
4
5 select title,facebook_likes,imdb_score
6 from films
7 inner join reviews
8 on films.id=reviews.id
9 order by imdb_score desc
10
```

Data Output

	title	facebook_likes	imdb_score
1	Goal! The Dream Begins	0	9.5
2	Lethal Weapon 3	108000	9.3
3	Bobby	43000	9.2
4	American Wedding	0	9.1
5	American Ninja 2: The Confrontation	37000	9
6	Bubble Boy	14000	9
7	An Inconvenient Truth	41000	8.9
8	After	40000	8.9
9	House of Sand and Fog	20000	8.9
10	Twilight	45000	8.9

Total rows: 1000 of 4968 Query complete 00:00:00.250 Ln 9, Col 20

Servers (2)

- local\_db
- Databases (3)
  - coffee
  - film
  - Scratch Pad

Query History

- Question 16
- list the top 7 certification of movies by profit made
- select certification, sum(gross-budget) as profit
- from films
- where "certification" is not null and "gross" is not null
- group by certification
- order by sum (gross-budget) desc
- limit 7

Schemas (1)

- public
  - Aggregates
  - Collations
  - Domains
  - FTS Configurations
  - FTS Dictionaries
  - FTS Parsers
  - FTS Templates
  - Foreign Tables
  - Functions
  - Materialized Views
  - Operators
  - Procedures
  - Sequences
  - Tables (4)
    - films
    - people
    - reviews
    - roles
  - Trigger Functions
  - Types

film/postgres@PostgreSQL 16

 No limit 

## Query

Scratch Pad x

```
1 --Question 16
2 --list the top 7 certification of movies by profit made
3
4 select certification, sum(gross-budget) as profit
5 from films
6 where "certification" is not null and "gross" is not null
7 group by certification
8 order by sum (gross-budget) desc
9 limit 7
```

## Data Output

Messages Notifications



	certification character varying	profit numeric
1	PG-13	17254366580
2	PG	13043242939
3	G	3006385250
4	Approved	661034398
5	X	151388116
6	M	112108900
7	GP	36600000

Total rows: 7 of 7 | Query complete 00:00:00.135

Ln 2, Col 54

Servers (2)

local\_db

Databases (3)

coffee

film

Casts

Catalogs

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 (4)

films

people

reviews

roles

Trigger Functions

Types

film/postgres@PostgreSQL 16



No limit

E

I

C

D

H

?

Query

Query History

Scratch Pad x

```
1 --Question 17
2 --list top 7 release_year of movies by profit
3
4
5 select release_year, gross
6 from films
7 where "release_year" is not null and "gross" is not null
8 order by gross desc
9 limit 10
10
11
12
13
```

Data Output

Messages

Notifications



	release_year integer	gross bigint
1	2015	936627416
2	2009	760505847
3	1997	658672302
4	2015	652177271
5	2012	623279547
6	2012	623279547
7	2008	533316061
8	1999	474544677
9	1977	460935665
10	2015	458991599

Total rows: 10 of 10

Query complete 00:00:00.107

Ln 2, Col 46

Servers (2)

local\_db

Databases (3)

coffee

film

- > Casts
- > Catalogs
- > 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 (4)
  - > films
  - > people
  - > reviews
  - > roles
- > Trigger Functions
- > Types

film/postgres@PostgreSQL 16

v

No limit

v

v

v

v

v

v

v

v

v

v

v

v

v

v

v

v

v

v

v

v

v

v

Query

Query

Edit

```
1 --Question 18
2 --Calculate the average number of user and critic reviews for films.
3
4
5 SELECT film_id, AVG(num_user) AS average_num_user, AVG(num_critic) AS average_num_critic
6 FROM reviews
7 GROUP BY film_id
```

Scratch Pad x

c

Data Output Messages Notifications

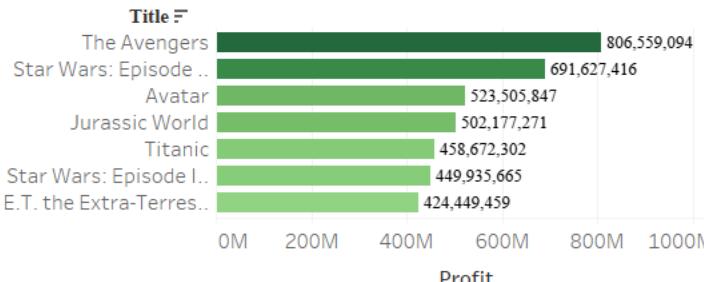
	film_id integer	average_num_user numeric	average_num_critic numeric
1	1489	3.0000000000000000	13.0000000000000000
2	4790	133.00000000000000	147.00000000000000
3	273	494.00000000000000	151.00000000000000
4	3936	10.00000000000000	56.00000000000000
5	2574	98.00000000000000	140.00000000000000
6	951	857.00000000000000	172.00000000000000
7	4326	212.00000000000000	393.00000000000000
8	2614	29.00000000000000	47.00000000000000
9	2520	31.00000000000000	71.00000000000000
10	2466	225.00000000000000	125.00000000000000

Total rows: 1000 of 4968 Query complete 00:00:00.145

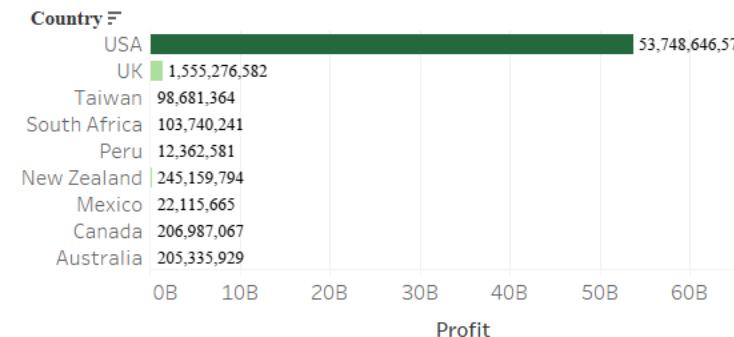
Ln 3, Col 1

# PUBLIC FILMS DASHBOARD

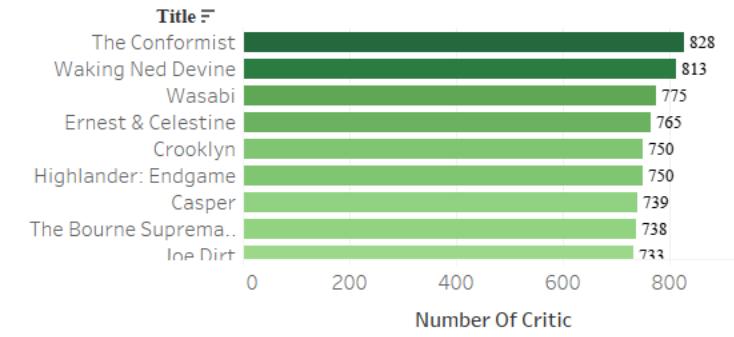
## Top 10 Movies by Profit



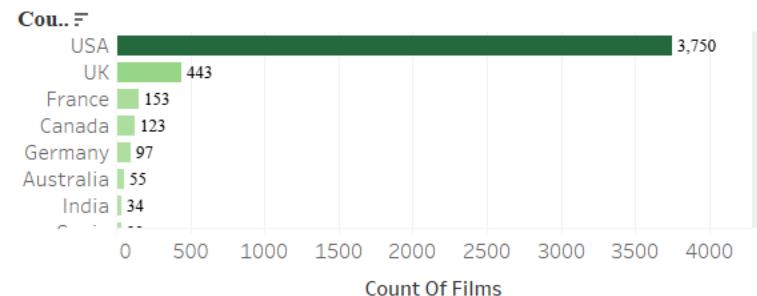
## Top 10 Country with the Highest Profit



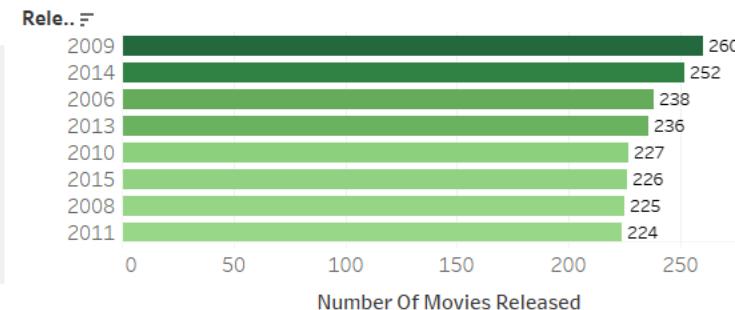
## Top 10 Movies by Critic



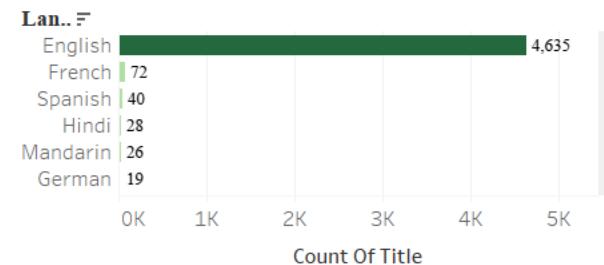
## Top 10 highest movie producing country



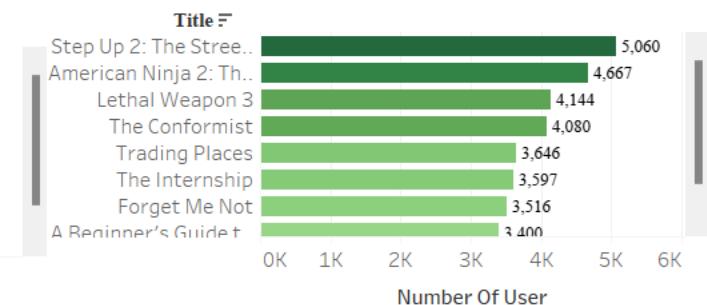
## Top 10 highest number of movies by year



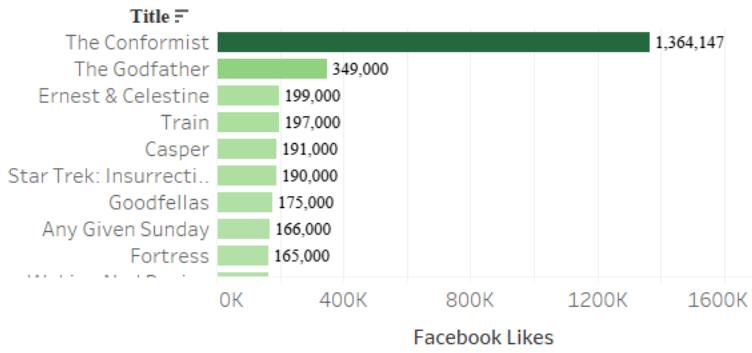
## Top 10 Movies by Language



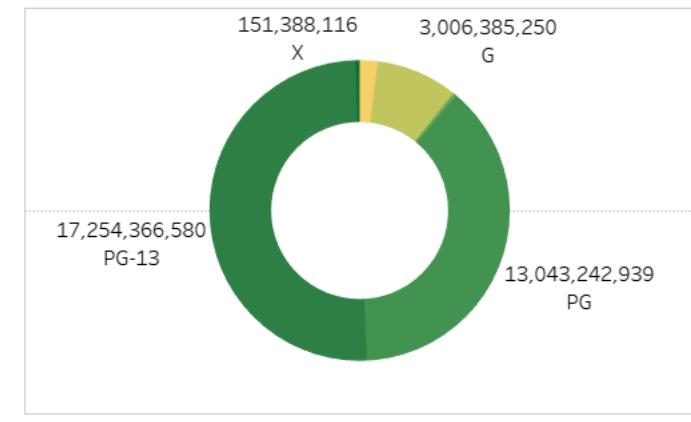
## Top 10 movies by User



## Top 10 Movies by Facebook likes



## Certification of movies by Profit made



# Findings and summary

- ❖ This movie dataset contains movie from year 1916 – 2016
- ❖ Country USA has the highest number of movie released(3,750) and highest profit on movies(53,748,646,579)
- ❖ Year 2009 has the highest number of movie released(260) and second highest profit(523505847)
- ❖ 4635 films released are in English Language which is the highest
- ❖ Movies that made the highest profit in 21<sup>st</sup> century is “star wars:Episode VII – The force Awakens”
- ❖ Number of people who are still alive is 5365
- ❖ The movie with the highest number if user is “Step up 2: The streets”
- ❖ The movies with the highest number of critic and facebook likes is “ The conformist”
- ❖ The movie with the highest number of Imdb score is “Goal! The dream begins