

- Foreign Keys
- Triggers
- movies
 - Columns
 - id
 - original_title
 - budget
 - popularity
 - release_date
 - revenue
 - title
 - vote_average
 - vote_count
 - overview
 - tagline
 - uid
 - director_id
 - Indexes
 - Foreign Keys

tion Schemas

```

3      -- Q1. Can you get all data about movies?
4
5      -- In this we use left join , the left table is movies and the right table is Directors when we join 2 tables all the Records
6      -- from the Left table remains same and the common(Matched) records from the right table, and there is no match the results contain
7      -- null from the directors table.
8
9      • select m.director_id, m.original_title,m.budget,m.popularity,m.release_date,m.revenue,m.title,m.vote_average,
10         m.vote_count,m.overview,m.tagline,d.name,d.id,d.department
11         from movies as m
12         left join directors as d
13         on m.director_id = d.id;
14

```

	director_id	original_title	budget	popularity	release_date	revenue	title	vote_average	vote_count	over
	4762	Avatar	237000000	150	2009-12-10	2787965087	Avatar	7.2	11800	In the
	4763	Pirates of the Caribbean: At World's End	300000000	139	2007-05-19	961000000	Pirates of the Caribbean: At World's End	6.9	4500	Capt
	4764	Spectre	245000000	107	2015-10-26	880674609	Spectre	6.3	4466	A cry
	4765	The Dark Knight Rises	250000000	112	2012-07-16	1084939099	The Dark Knight Rises	7.6	9106	Follow
	4766	John Carter	260000000	43	2012-03-07	284139100	John Carter	6.1	2124	John
	4767	Spider-Man 3	258000000	115	2007-05-01	890871626	Spider-Man 3	5.9	3576	The s
	4768	Tangled	260000000	48	2010-11-24	591794936	Tangled	7.4	3330	Wher
	4769	Avengers: Age of Ultron	280000000	134	2015-04-22	1405403694	Avengers: Age of Ultron	7.3	6767	Wher
	4770	Harry Potter and the Half-Blood Prince	250000000	98	2009-07-07	933850197	Harry Potter and the Half-Blood Prince	7.4	5203	As H:

Read On!

Tables

- directors
 - Columns
 - name
 - id
 - gender
 - uid
 - department
 - Indexes
 - Foreign Keys
 - Triggers
- movies
 - Columns
 - id
 - original_title
 - budget
 - popularity
 - release_date
 - revenue

Schemas

```

1  -- Q2. How do you get all data about directors?
2
3  -- We use Left join, directors table as left table and movies table as right. All the records from the directors table(Left table)
4  -- and the matching records from the movies table(Right table). If there is no matching records from the Right table
5  -- the records contain NULL values.
6
7  • select *
8    from directors as d left join movies as m
9    on d.id = m.director_id;
10

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	name	id	gender	uid	department	id	original_title	budget	popularity	release_date	revenue	title
▶	James Cameron	4762	2	2710	Directing	43597	Avatar	237000000	150	2009-12-10	2787965087	Avatar
	Gore Verbinski	4763	2	1704	Directing	43598	Pirates of the Caribbean: At World's End	300000000	139	2007-05-19	961000000	Pirates of the Caribbean: At World's End
	Sam Mendes	4764	2	39	Directing	43599	Spectre	245000000	107	2015-10-26	880674609	Spectre
	Christopher Nolan	4765	2	525	Directing	43600	The Dark Knight Rises	250000000	112	2012-07-16	1084939099	The Dark Knight Rises
	Andrew Stanton	4766	2	7	Directing	43601	John Carter	260000000	43	2012-03-07	284139100	John Carter
	Sam Raimi	4767	2	7623	Directing	43602	Spider-Man 3	258000000	115	2007-05-01	890871626	Spider-Man 3
	Byron Howard	4768	2	76595	Directing	43603	Tangled	260000000	48	2010-11-24	591794936	Tangled
	Joss Whedon	4769	2	12891	Directing	43604	Avengers: Age of Ultron	280000000	134	2015-04-22	1405403694	Avengers: Age of Ultron
	David Yates	4770	2	11343	Directing	43605	Harry Potter and the Half-Blood Prince	250000000	98	2009-07-07	933959197	Harry Potter and the Half-Blood Prince
	Bryan Singer	4772	2	9032	Directing	43607	Superman Returns	270000000	57	2006-06-28	391081192	Superman Returns

CS

- gender
- uid
- department
- Indexes
- Foreign Keys
- Triggers
- movies
- Columns
 - id
 - original_title
 - budget
 - popularity
 - release_date
 - revenue
 - title
 - vote_average
 - vote_count
 - overview
 - tacline
- Schemas

title

latin1_swedish_ci

n:

ext

Session

Limit to 50000 rows

```

1  -- Q3. Check how many movies are present in IMDB.
2
3  • select count(distinct title) as "No. of Movies"
4    from movies;
5

```

Result Grid

	No. of Movies
▶	47

Filter Rows: Export: Wrap Cell Content:

Result 3 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✗ 12	16:00:25		Error Code: 2003 Unable to connect to 18.136.157.135:3306	
✓ 13	17:05:00	select count(distinct title) from movies LIMIT 0, 50000	1 row(s) returned	0.062 sec / 0.0
✓ 14	17:05:42	select count(distinct title) as "No. of Movies" from movies LIMIT 0, 50000	1 row(s) returned	0.047 sec / 0.0
✓ 15	17:05:59	select count(distinct title) as "No. of Movies" from movies LIMIT 0, 50000	1 row(s) returned	0.047 sec / 0.0

ts

project_movie_database

roles

directors

Columns

- name
- id
- gender
- uid
- department

Indexes

Foreign Keys

Triggers

movies

Columns

- id
- original_title
- budget
- popularity
- release_date

Schemas

Limit to 50000 rows

```

1  -- Q4. Find these 3 directors: James Cameron ; Luc Besson ; John Woo
2
3  •  select * from directors
4     where name in ('James Cameron','Luc Besson','John Woo');
5

```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	name	id	gender	uid	department
▶	James Cameron	4762	2	2710	Directing
	John Woo	4893	2	11401	Directing
	Luc Besson	4949	2	59	Directing
*	NULL	NULL	NULL	NULL	NULL

directors 5 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 19	17:18:51	SELECT * FROM project_movie_database.movies LIMIT 0, 50000	47 row(s) returned	0.094 sec / 0.000
✓ 20	17:19:05	SELECT * FROM project_movie_database.directors LIMIT 0, 50000	2349 row(s) returned	0.140 sec / 0.047
✓ 21	17:22:12	select m.original_title,d.name from movies as m left join directors as d on m.director_id = d.id	2 row(s) returned	0.063 sec / 0.000
✓ 22	17:23:03	select * from directors where name in ('James Cameron','Luc Besson','John Woo') LIMIT 0, 50000	3 row(s) returned	0.047 sec / 0.000

movie_database

- es
- irectors
- Columns
 - name
 - id
 - gender
 - uid
 - department

- Indexes
- Foreign Keys
- Triggers

movies

- Columns
 - id
 - original_title
 - budget
 - popularity
 - release_date

Schemas

note_average

average double

Limit to 50000 rows

```
1  -- Q5. Find all directors with name starting with S.
2
3  • select * from directors
4    where name like 'S%' ;
```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

	name	id	gender	uid	department
▶	Sam Mendes	4764	2	39	Directing
	Sam Raimi	4767	2	7623	Directing
	Shane Black	4784	2	1108	Directing
	Steven Spielberg	4799	2	488	Directing
	Stephen Sommers	4815	2	7775	Directing
	Shawn Levy	4842	2	17825	Directing
	Steve Hickner	4852	2	44113	Directing
	Simon Wells	4855	2	21879	Directing
	Steven Soderbergh	4909	2	1884	Directing
	Simon West	4930	2	12786	Directing
	Stefen Fangmeier	4931	0	25453	Directing
	Spike Jonze	4932	2	5953	Directing
	Steve Martino	4943	2	71729	Directing
	Sergei Bodrov	4952	0	130938	Directing

directors 2 ×

Apply

Output

ects

ect_movie_database

ables

- directors
 - Columns
 - name
 - id
 - gender
 - uid
 - department
 - Indexes
 - Foreign Keys
 - Triggers
- movies
 - Columns
 - id
 - original_title
 - budget
 - popularity
 - release_date

ion Schemas

Limit to 50000 rows

```
1  -- Q6. Count female directors.
2
3  -- For female, gender = 1 is given in the Directors table.
4
5  • select count(distinct name) as No_of_Female_Director
6    from directors
7    where gender = 1;
```

Result Grid

No_of_Female_Director
150

Filter Rows: Export: Wrap Cell Content:

Result 2 x

Output

cts

directors

Columns

- name
- id
- gender
- uid
- department

Indexes

Foreign Keys

Triggers

movies

Columns

- id
- original_title
- budget
- popularity
- release_date
- revenue
- title

Schemas

release_date

latin1_swedish_ci

on:

Limit to 50000 rows

```
1  -- Q7. Find the name of the 10th first women directors?
2
3  • select * from directors
4    where gender = 1
5    order by id
6    limit 1 offset 9;
7
8  -- Angelina Jolie is the 10th First Women Director.
9
10
11
```

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content: Fetch rows:

	name	id	gender	uid	department
▶	Angelina Jolie	5086	1	11701	Directing
*	NULL	NULL	NULL	NULL	NULL

directors 8 x

Output

Action Output

Apply

Revert

AS

objects

project_movie_database

Tables

directors

movies

Views

Stored Procedures

Functions

Administration Schemas

Function

Object selected

```
1  -- Q8. What are the 3 most popular movies?
2
3  • select original_title , popularity
4  from movies
5  order by popularity desc
6  limit 3;
7
8  -- Three popular movies are 'Jurassic World' , 'Captain America: Civil War' , 'Avatar'
9
10
```

< Result Grid Filter Rows: Export: Wrap Cell Content:

	original_title	popularity
▶	Jurassic World	418
	Captain America: Civil War	198
	Avatar	150

movies 1 x

Output

MAS

er objects

project_movie_database

Tables

▶ directors

▶ movies

Views

Stored Procedures

Functions

Administration Schemas

ation:.....

object selected

 Don't Limit

```
1  -- Q9. What are the 3 most bankable movies?
2
3  • select original_title , revenue, budget , (revenue - budget) as profit
4  from movies
5  order by (revenue - budget) desc
6  limit 3;
7
8  -- The three most bankable movies are 'Avatar', 'Titanic' , 'Jurassic World'.
9
```

<

Result Grid  Filter Rows: Export:  Wrap Cell Content: 

	original_title	revenue	budget	profit
▶	Avatar	2787965087	237000000	2550965087
	Titanic	1845034188	200000000	1645034188
	Jurassic World	1513528810	150000000	1363528810

Result 1 ×

Output

er objects

project_movie_database

Tables

directors

movies

Views

Stored Procedures

Functions

stration Schemas

ation

object selected

Don't Limit

```
1  -- Q10. What is the most awarded average vote since the January 1st, 2000?
2
3  •  select original_title, vote_average
4     from movies
5     where release_date > '2000-01-01'
6     order by vote_average desc
7     limit 1
```

Result Grid Filter Rows: Export: Wrap Cell Content:

	original_title	vote_average
▶	The Dark Knight Rises	7.6

movies 1 x

Output

er objects

project_movie_database

Tables

directors

movies

Views

Stored Procedures

Functions

stration Schemas

ation

object selected



Don't Limit

```

1  -- Q11. Which movie(s) were directed by Brenda Chapman?
2
3  -- Method 1 : Using Join
4  • select m.original_title, d.name
5    from movies as m inner join directors as d
6    on m.director_id = d.id
7    where d.name like 'Brenda Chapman';
8
9  -- Method 2 : Using Subquery
10 • select original_title from movies
11    where director_id in (select id from directors
12                          where name = 'Brenda Chapman');
13
14  -- No records were returned, which means Brenda Chapman has no movies associated with her.

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

original_title	name
----------------	------

Result 2 x

Output

Result Grid

Form Editor

Read Only

Navigator

HEMAS

Filter objects

project_movie_database

Tables

directors

movies

Views

Stored Procedures

Functions

Administration

Schemas

Information

No object selected

Object Info

Session

Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q13

Don't Limit

1 -- Q13. Which director is the most bankable?

2 -- ----- Method 1: Sub-Query

3 • select name from directors

4 where id in (select director_id from

5 (select director_id , sum(revenue-budget) as total_Profit from movies -- Revenue alone cannot reflect Profitability

6 group by director_id -- It might happens that Revenue is high then Budget is also high

7 order by sum(revenue-budget) desc) as c) -- So i use formula (REVENUE - BUDGET) to find Profit

8 limit 1 ;

9

10 -- ----- Method 2: Joins

11 • select d.name as Director_name , sum(m.revenue - m.budget) as Total_Profit from movies as m inner join directors as d

12 on m.director_id = d.id

13 group by Director_name

14 order by Total_Profit desc

15 limit 1 ;

16

17 -- James Cameron is the most bankable Director because hs movies generate the highest Total Revenue.

Result Grid Filter Rows: Export: Wrap Cell Content:

	Director_name	Total_Profit
▶	James Cameron	4195999275

Result 1

Read On

EMAS

filter objects

project_movie_database

- Tables
 - directors
 - movies
- Views
- Stored Procedures
- Functions

Administration Schemas

Information

No object selected

Object Info Session

Don't Limit

```
1  -- Q12. Which director made the most movies?
2  -- ----- Method 1 : Joins
3  • select d.name as Director_name , count(m.director_id) as No_of_Movies
4    from movies as m inner join directors as d on m.director_id = d.id
5    group by d.name
6    order by count(m.director_id) desc
7    limit 3;
8
9  -- ----- Mothod 2 : Subquery
10 • select name
11    from directors
12   where id in
13         (select director_id
14          from (select director_id , count(director_id) from movies
15               group by director_id
16               order by count(director_id) desc) as c)
17   limit 3;
18  -- Gore Verbinski. Sam Raimi and Peter Jackson have directed the highest number of movies. with 3 Movies Each.
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

name
Peter Jackson
Gore Verbinski
Sam Raimi

directors 1 x