

Actor Job

The screenshot displays the Talend Cloud Data Fabric interface for the 'Actor Job'. The main workspace shows a data flow diagram with the following components and connections:

- Actor_JSON_File** (File Input) connects to **actor_json_schema** (Schema).
- actor_json_schema** connects to **tMap_1** (Map).
- tMap_1** connects to **actor (Main order:1)** (Table Output).
- actor (Main order:1)** connects to **tLogRow_1** (Log Row).

The **Execution** tab shows the job status: 'Starting Job Actor_Job at 21:47 17-03-2024'. The **Table** view displays the data being loaded into the **Actor_DB**:

name	(ActorID)birthname	birthdate	birthplace
Akira Kurosawa	1900	1900-03-14	Atsuta, Mikasa, Japan

Director Job

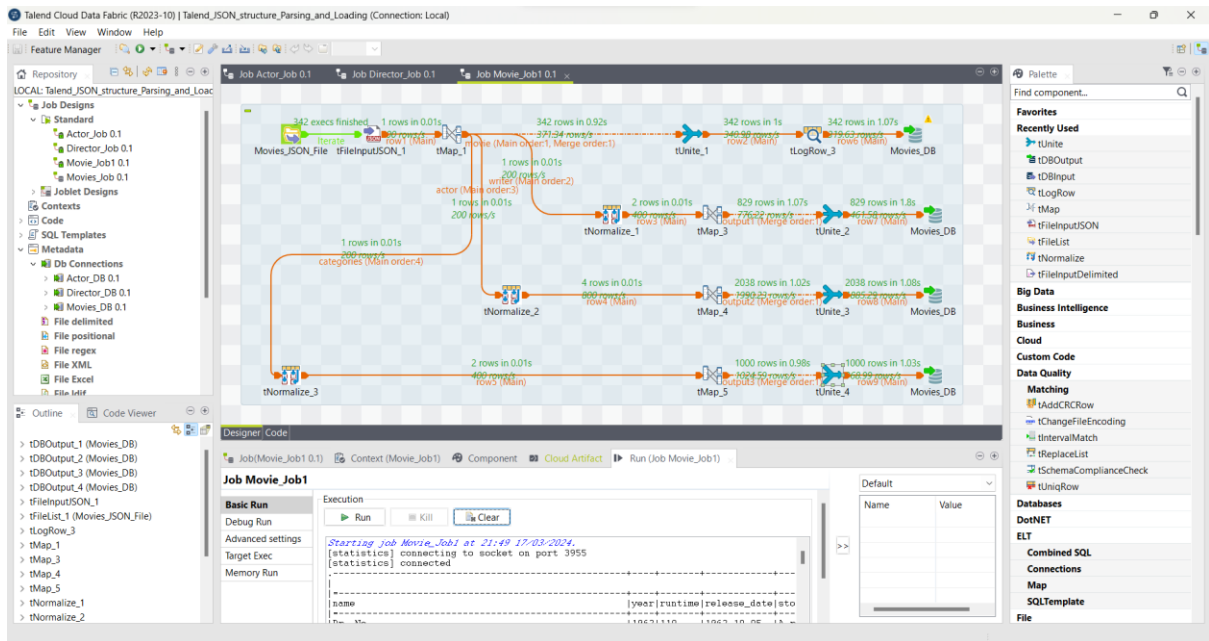
The screenshot displays the Talend Cloud Data Fabric interface for the 'Director Job'. The main workspace shows a data flow diagram with the following components and connections:

- Director_JSON_File** (File Input) connects to **director_json_schema** (Schema).
- director_json_schema** connects to **tMap_1** (Map).
- tMap_1** connects to **director (Main order:1)** (Table Output).
- director (Main order:1)** connects to **tLogRow_1** (Log Row).

The **Execution** tab shows the job status: 'Starting Job Director_Job at 21:47 17-03-2024'. The **Table** view displays the data being loaded into the **Director_DB**:

name	DirectorID	birthname	birthdate	birthplace
Akira Kurosawa	5000	Akira Kurosawa	1910-03-23	Tokyo, Japan

Movie Job



SQL screenshots:

The screenshot displays the MySQL Workbench interface. The 'Query' tab is active, showing a query that has been executed. The 'Results' tab shows the query results for the 'actor' table.

Query:

```
SELECT * FROM actor_table;
```

Results:

name	birthdate	birthplace	ActorID
Aamir Khan	1965-03-14 00:00:00	Mumbai, Maharashtra, India	501
Adam Sandler	1966-09-09 00:00:00	New York, United States	503
Adrianne Palicki	1983-05-06 00:00:00	Toledo, Ohio, USA	505
Aisha N. Tyler	1970-09-18 00:00:00	San Francisco, California, United States	507
Aishwarya Rai Bachchan	1973-11-01 00:00:00	Mangalore, Karnataka, India	509
Alkesh Kumar	1967-09-09 00:00:00	Aamrtaar, Punjab, India	511
Alexandra Daddario	1986-03-16 00:00:00	Manhattan, New York City, New York, USA	513
Amber Heard	1986-04-22 00:00:00	Austin, Texas, U.S	515
Amithabh Bachchan	1942-10-11 00:00:00	Allahabad, India	517
Amy Adams	1974-10-20 00:00:00	Vicenza, Italy	519
Anatole Taubman	1970-12-23 00:00:00	Zurich, Switzerland	521
Angelina Jolie	1975-06-04 00:00:00	Los Angeles, California, United States	523
Anne Hathaway	1982-11-12 00:00:00	Brooklyn, New York, USA	525
Anushka Shetty	1981-11-07 00:00:00	Mangalore, India	527
Anya Taylor-Joy	1996-04-16 00:00:00	Miami, Florida, United States	529
Arnold Schwarzenegger	1947-07-30 00:00:00	Thal, Styria, Austria	531
Ashley Scott	1977-07-13 00:00:00	Metairie, Louisiana	533
Ashton Kutcher	1978-02-07 00:00:00	Cedar Rapids, Iowa, United States	535
Ben Affleck	1972-08-15 00:00:00	Berkeley, California, USA	537
Benedict Cumberbatch	1976-07-19 00:00:00	Hammersmith, London, England	539
Bette Davis	1908-04-05 00:00:00	Lowell, Massachusetts, USA	541
Bingbing Li	1973-02-27 00:00:00	Harbin, China	543
Bob Gunton	1945-11-15 00:00:00	Santa Monica, California, USA	545
Bonnie Aarons	1979-06-03 00:00:00	Los Angeles, California, United States	547
Brad Beyer	1973-09-20 00:00:00	Waukegan, Wisconsin	549

The interface includes a left sidebar with a 'Schemas' view showing the database structure, an 'Information' view showing the table details, and an 'Output' view showing the query results. The 'Execution' tab at the bottom shows the job status and logs.

MySQL Workbench

Talend_JSON (json) x Talend_JSON (json) x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHEMAS

actor

air_traffic

json

Tables

actor_table

categories_table

director_table

movie

new_actor_table

writer_table

Views

Stored Procedures

Functions

quiz

seattle_pet_license

seattle_pet_license1

seattle_pet_license2

Administration Schemas

Information

Table: actor_table

Columns:

name varchar(100)

birthname varchar(100)

birthdate datetime

birthplace varchar(100)

ActorID int PK

Query 1 categories_table SQL File 1*

Limit to 1000 rows

1 SELECT * FROM actor_table;

2

3 SELECT COUNT(*) FROM actor_table;

4

5 SELECT * FROM director_table;

6

7 SELECT * FROM movie;

8

9 SELECT * FROM writer_table;

10

11 SELECT * FROM new_actor_table;

12

13 SELECT * FROM categories_table;

14

15 select distinct(name) from (select a.name, categories, count(distinct actors), count(distinct writer) from new_actor_table a

Result Grid

Filter Rows:

Exports: | Wrap Cell Contents: |

Result Grid

Form Editor

Field Types

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Read Only Context Help Snippets

Object Info Session

Output

MySQL Workbench

Talend_JSON (json) x Talend_JSON (json) x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHEMAS

actor

air_traffic

json

Tables

actor_table

categories_table

director_table

movie

new_actor_table

writer_table

Views

Stored Procedures

Functions

quiz

seattle_pet_license

seattle_pet_license1

seattle_pet_license2

Administration Schemas

Information

Table: actor_table

Columns:

name varchar(100)

birthname varchar(100)

birthdate datetime

birthplace varchar(100)

ActorID int PK

Query 1 categories_table

Limit to 1000 rows

1 SELECT * FROM actor_table;

2

3 SELECT * FROM director_table;

Result Grid

Filter Rows:

Exports/Imports: | Wrap Cell Contents: |

name	birthname	birthdate	birthplace	DirectorID
Akira Kurosawa	Akira Kurosawa	1910-03-23 00:00:00	Tokyo, Japan	5001
Alfred Hitchcock	Alfred Joseph Hitchcock	1899-08-13 00:00:00	Leytonstone, Essex, England	5003
Ang Lee	Ang Lee	1954-10-23 00:00:00	Pingtung, Taiwan	5005
Anurag Kashyap	Anurag Singh Kashyap	1972-09-10 00:00:00	Gorakhpur, Uttar Pradesh, India	5007
Billy Wilder	Samuel Wilder	1906-06-22 00:00:00	Sucha, Galicia, Austria-Hungary	5009
Brad Bird	Brad Bird	1957-09-24 00:00:00	Kalispell, Montana, USA	5011
Chris Columbus	Chris Joseph Columbus	1958-09-10 00:00:00	Spangler, Pennsylvania, USA	5013
Christopher Nolan	Christopher Edward Nolan	1970-07-30 00:00:00	Westminster, London, England	5015
Clint Eastwood	Clinton Eastwood Jr.	1930-05-31 00:00:00	San Francisco, California, U.S.	5017
Damien Chazelle	Imtiaz Ali	1985-01-19 00:00:00	Providence, Rhode Island, United States	5019
David Fincher	David Andrew Leo Fincher	1962-08-28 00:00:00	Denver, Colorado, United States	5021
David Lynch	David Keith Lynch	1946-01-20 00:00:00	Massula, Montana, USA	5023
Dibakar Banerjee	Dibakar Banerjee	1969-06-21 00:00:00	New Delhi, India	5025
Francis Ford Coppola	Francis Ford Coppola	1937-04-07 00:00:00	Detroit, Michigan, United States	5027
Fritz Lang	Friedrich Christian Anton ...	1890-12-05 00:00:00	Venna, Austria-Hungary	5029
George Lucas	George Walton Lucas Jr.	1944-05-14 00:00:00	Modesto, California, U.S.	5031
Greg Robbins	Greg Robbins	1959-07-06 00:00:00	Los Angeles, California, USA	5033
Guy Ritchie	Guy Ritchie	1968-09-10 00:00:00	Hatfield, Hertfordshire, England	5035
Harmony Korine	Harmony Korine	1973-01-04 00:00:00	Bolinas, California, USA	5037
Imtiaz Ali	Imtiaz Ali	1971-06-16 00:00:00	Jamshedpur, Jharkhand, India	5039
J. J. Abrams	Jeffrey Jacob Abrams	1966-06-27 00:00:00	New York, U.S.	5041
James Cameron	James Francis Cameron	1954-08-16 00:00:00	Kapuskasing, Ontario, Canada	5043
Jason Reitman	Jason Reitmans	1977-10-09 00:00:00	Montreal, Quebec, Canada	5045
John Ford	John Martin Feeney	1894-02-01 00:00:00	Cape Elizabeth, Maine, United States	5047
Jon Turteltaub	Jonathan Charles Turteltaub	1963-08-08 00:00:00	New York City, New York, USA	5049

director_table 21 x

Apply Revert Context Help Snippets

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Object Info Session

Output

MySQL Workbench

Talend_JSON (json) x Talend_JSON (json) x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHEMAS

- actor
- air_traffic
- json
 - Tables
 - actor_table
 - categories_table
 - director_table
 - movie
 - new_actor_table
 - writer_table
 - Views
 - Stored Procedures
 - Functions
- quiz
- seattle_pet_license
- seattle_pet_license1
- seattle_pet_license2
- Administration
- Schemas

Information

Table: actor_table

Columns:

- name varchar(100)
- birthname varchar(100)
- birthdate datetime
- birthplace varchar(100)
- ActorID int PK

Query 1 categories_table SQL File 1*

Limit to 1000 rows

```
1 SELECT * FROM actor_table;
2
3 SELECT COUNT(*) FROM actor_table;
4
5 SELECT * FROM director_table;
6
7 SELECT COUNT(*) FROM director_table;
8
9 SELECT * FROM movie;
10
11 SELECT * FROM writer_table;
12
13 SELECT * FROM new_actor_table;
14
15 SELECT * FROM categories_table;
```

Result Grid

Filter Rows: Exports Wrap Cell Contents

Result 33

Output

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Talend_JSON (json) x Talend_JSON (json) x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHEMAS

- actor
- air_traffic
- json
 - Tables
 - actor_table
 - categories_table
 - director_table
 - movie
 - new_actor_table
 - writer_table
 - Views
 - Stored Procedures
 - Functions
- quiz
- seattle_pet_license
- seattle_pet_license1
- seattle_pet_license2
- Administration
- Schemas

Information

Table: actor_table

Columns:

- name varchar(100)
- birthname varchar(100)
- birthdate datetime
- birthplace varchar(100)
- ActorID int PK

Query 1 categories_table

Limit to 1000 rows

```
1 SELECT * FROM actor_table;
2
3 SELECT * FROM director_table;
4
5 SELECT * FROM movie;
```

Result Grid

Filter Rows: Exports Wrap Cell Contents

name	year	runtime	release_date	storyline	director
Dr. No	1962	110	1962-10-05 00:00:00	A resourceful British government agent seeks a...	Terence Young
To Kill a Mockingbird	1962	129	1962-12-25 00:00:00	Atticus Finch, a lawyer in the Depression-era So...	Robert Mulligan
Whatever Happened To Baby Jane?	1962	134	1962-10-31 00:00:00	A former child star torments her paraplegic siste...	Robert Aldrich
From Russia with Love	1963	115	1963-10-10 00:00:00	James Bond willingly falls into an assassination p...	Terence Young
Goldfinger	1964	110	1964-09-17 00:00:00	Investigating a gold magnate's smuggling, Jame...	Guy Hamilton
The Sound of Music	1965	172	1965-03-29 00:00:00	A woman leaves an Austrian convent to becom...	Robert Wise
Thunderball	1965	130	1965-12-09 00:00:00	James Bond heads to The Bahamas to recover t...	Terence Young
The Good, the Bad and the Ugly	1966	161	1966-12-29 00:00:00	A bounty hunting scam joins two men in an une...	Sergio Leone
You Only Live Twice	1967	117	1967-06-12 00:00:00	Agent 007 and the Japanese secret service ring...	Lewis Gilbert
Diamonds Are Forever	1971	120	1971-12-14 00:00:00	A diamond smuggling investigation leads James...	Guy Hamilton
The Godfather	1972	177	1972-03-15 00:00:00	When the aging head of a famous crime family ...	Francis Ford C...
La Patagonia rebelde	1974	103	1975-01-06 00:00:00	La Patagonia rebelde is an Argentinean film bas...	Hector Olivera
The Godfather: Part II	1974	200	1974-12-12 00:00:00	The continuing saga of the Corleone crime fami...	Francis Ford C...

Result 22

Output

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Talend_JSON (json) x Talend_JSON (json) x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHEMAS

- actor
- air_traffic
- json
 - Tables
 - actor_table
 - categories_table
 - director_table
 - movie
 - new_actor_table
 - writer_table
 - Views
 - Stored Procedures
 - Functions
- quiz
- seattle_pet_license
- seattle_pet_license1
- seattle_pet_license2
- Administration
- Schemas

Information

Table: actor_table

Columns:

- name varchar(100)
- birthname varchar(100)
- birthdate datetime
- birthplace varchar(100)
- ActorID int PK

Query 1 categories_table SQL File 1*

```
1 SELECT * FROM actor_table;
2
3 SELECT COUNT(*) FROM actor_table;
4
5 SELECT * FROM director_table;
6
7 SELECT COUNT(*) FROM director_table;
8
9 SELECT * FROM movie;
10
11 SELECT COUNT(*) FROM movie;
12
13 SELECT * FROM writer_table;
14
15 SELECT * FROM new_actor_table;
```

Result Grid

Filter Rows: Exports Wrap Cell Contents: 15

Result 34 x

Output

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Talend_JSON (json) x Talend_JSON (json) x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHEMAS

- actor
- air_traffic
- json
 - Tables
 - actor_table
 - categories_table
 - director_table
 - movie
 - new_actor_table
 - writer_table
 - Views
 - Stored Procedures
 - Functions
- quiz
- seattle_pet_license
- seattle_pet_license1
- seattle_pet_license2
- Administration
- Schemas

Information

Table: actor_table

Columns:

- name varchar(100)
- birthname varchar(100)
- birthdate datetime
- birthplace varchar(100)
- ActorID int PK

Query 1 categories_table

```
1 SELECT * FROM actor_table;
2
3 SELECT * FROM director_table;
4
5 SELECT * FROM movie;
6
7 SELECT * FROM writer_table;
```

Result Grid

Filter Rows: Exports Wrap Cell Contents: 15

Result 23 x

name	writer
Dr. No	Richard Maibaum
Dr. No	Johanna Harwood
Dr. No	Berkeley Mather
Dr. No	Ian Fleming
To Kill a Mockingbird	Harper Lee
To Kill a Mockingbird	Horton Foote
Whatever Happened To Baby Jane?	Henry Farrell
Whatever Happened To Baby Jane?	Lukas Heller
From Russia with Love	Richard Maibaum
From Russia with Love	Johanna Harwood
From Russia with Love	Berkeley Mather
From Russia with Love	Ian Fleming
Goldfinger	Richard Maibaum

writer_table 23 x

Output

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Talend_JSON (json) x Talend_JSON (json) x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHEMAS

- actor
- air_traffic
- json
 - Tables
 - actor_table
 - categories_table
 - director_table
 - movie
 - new_actor_table
 - writer_table
 - Views
 - Stored Procedures
 - Functions
- quiz
- seattle_pet_license
- seattle_pet_license1
- seattle_pet_license2
- Administration
- Schemas

Information

Table: actor_table

Columns:

name	varchar(100)
birthname	varchar(100)
birthdate	datetime
birthplace	varchar(100)
ActorID	int PK

Query 1 categories_table SQL File 1*

Limit to 1000 rows

```

11 SELECT COUNT(*) FROM movie;
12
13 SELECT * FROM writer_table;
14
15 SELECT COUNT(*) FROM writer_table;
16
17 SELECT * FROM new_actor_table;
18
19 SELECT * FROM categories_table;
20
21 select distinct(name) from (select a.name, categories, count(distinct actors), count(distinct writer) from new_actor_table a
22 join writer_table w
23 on w.name=a.name
24 join categories_table c
25 on c.name=a.name

```

Result Grid

COUNT(*)
829

Result 35 x

Read Only Context Help Snippets

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Talend_JSON (json) x Talend_JSON (json) x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHEMAS

- actor
- air_traffic
- json
 - Tables
 - actor_table
 - categories_table
 - director_table
 - movie
 - new_actor_table
 - writer_table
 - Views
 - Stored Procedures
 - Functions
- quiz
- seattle_pet_license
- seattle_pet_license1
- seattle_pet_license2
- Administration
- Schemas

Information

Table: actor_table

Columns:

name	varchar(100)
birthname	varchar(100)
birthdate	datetime
birthplace	varchar(100)
ActorID	int PK

Query 1 categories_table

Limit to 1000 rows

```

1 SELECT * FROM actor_table;
2
3 SELECT * FROM director_table;
4
5 SELECT * FROM movie;
6
7 SELECT * FROM writer_table;
8
9 SELECT * FROM new_actor_table;

```

Result Grid

name	actors
Dr. No	Sean Connery
Dr. No	Ursula Andress
Dr. No	Joseph Wiseman
Dr. No	Jack Lord
Dr. No	Bernard Lee
To Kill a Mockingbird	Gregory Peck
To Kill a Mockingbird	John Megna
To Kill a Mockingbird	Frank Overton
To Kill a Mockingbird	Rosemary Murphy
To Kill a Mockingbird	Ruth White
To Kill a Mockingbird	Brock Peters
To Kill a Mockingbird	Estelle Evans
To Kill a Mockingbird	Paul Fix

Result 24 x

Read Only Context Help Snippets

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Talend_JSON (json) x Talend_JSON (json) x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHMAS

actor

air_traffic

json

Tables

actor_table

categories_table

director_table

movie

new_actor_table

writer_table

Views

Stored Procedures

Functions

quiz

seattle_pet_license

seattle_pet_license1

seattle_pet_license2

Administration Schemas

Information

Table: actor_table

Columns:

name varchar(100)

birthname varchar(100)

birthdate datetime

birthplace varchar(100)

ActorID int PK

Query 1 categories_table SQL File 1*

Limit to 1000 rows

5 • SELECT * FROM director_table;

6

7 • SELECT COUNT(*) FROM director_table;

8

9 • SELECT * FROM movie;

10

11 • SELECT COUNT(*) FROM movie;

12

13 • SELECT * FROM writer_table;

14

15 • SELECT COUNT(*) FROM writer_table;

16

17 • SELECT * FROM new_actor_table;

18

19 • SELECT COUNT(*) FROM new_actor_table;

Result Grid

Filter Rows:

Exports

Wrap Cell Contents

Result Grid

Form Editor

Field Types

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Read Only Context Help Snippets

Object Info Session

Output

MySQL Workbench

Talend_JSON (json) x Talend_JSON (json) x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHMAS

actor

air_traffic

json

Tables

actor_table

categories_table

director_table

movie

new_actor_table

writer_table

Views

Stored Procedures

Functions

quiz

seattle_pet_license

seattle_pet_license1

seattle_pet_license2

Administration Schemas

Information

Table: actor_table

Columns:

name varchar(100)

birthname varchar(100)

birthdate datetime

birthplace varchar(100)

ActorID int PK

Query 1 categories_table

Limit to 1000 rows

1 • SELECT * FROM actor_table;

2

3 • SELECT * FROM director_table;

4

5 • SELECT * FROM movie;

6

7 • SELECT * FROM writer_table;

8

9 • SELECT * FROM new_actor_table;

10

11 • SELECT * FROM categories_table;

Result Grid

Filter Rows:

Exports

Wrap Cell Contents

Fetch rows:

Result Grid

Form Editor

Field Types

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Read Only Context Help Snippets

Object Info Session

Output

name	categories
Dr. No	action
Dr. No	adventure
Dr. No	thriller
To Kill a Mockingbird	crime
To Kill a Mockingbird	drama
Whatever Happened To Baby Jane?	drama
Whatever Happened To Baby Jane?	horror
Whatever Happened To Baby Jane?	thriller
From Russia with Love	action
From Russia with Love	adventure
From Russia with Love	thriller
Goldfinger	action
Goldfinger	adventure

categories_table25 x

MySQL Workbench

Talend_JSON (json) x Talend_JSON (json) x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHEMAS

- actor
- air_traffic
- json
 - Tables
 - actor_table
 - categories_table
 - director_table
 - movie
 - new_actor_table
 - writer_table
 - Views
 - Stored Procedures
 - Functions
- quiz
- seattle_pet_license
- seattle_pet_licenseset
- seattle_pet_licenseset
- Administration Schemas

Information

Table: actor_table

Columns:

name	varchar(100)
birthname	varchar(100)
birthdate	datetime
birthplace	varchar(100)
ActorID	int PK

Query 1: categories_table SQL File 1*

```

9 • SELECT * FROM movie;
10
11 • SELECT COUNT(*) FROM movie;
12
13 • SELECT * FROM writer_table;
14
15 • SELECT COUNT(*) FROM writer_table;
16
17 • SELECT * FROM new_actor_table;
18
19 • SELECT COUNT(*) FROM new_actor_table;
20
21 • SELECT * FROM categories_table;
22
23 • SELECT COUNT(*) FROM categories_table;

```

Result Grid

COUNT(*)
1000

Result 37 x

Output

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Read Only Context Help Snippets

SQL Queries:

MySQL Workbench

Talend_JSON (json) x Talend_JSON (json) x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHEMAS

- actor
- air_traffic
- json
 - Tables
 - actor_table
 - categories_table
 - director_table
 - movie
 - new_actor_table
 - writer_table
 - Views
 - Stored Procedures
 - Functions
- quiz
- seattle_pet_license
- seattle_pet_licenseset
- seattle_pet_licenseset
- Administration Schemas

Information

Table: actor_table

Columns:

name	varchar(100)
birthname	varchar(100)
birthdate	datetime
birthplace	varchar(100)
ActorID	int PK

Query 1: categories_table

```

3 • SELECT * FROM director_table;
4
5 • SELECT * FROM movie;
6
7 • SELECT * FROM writer_table;
8
9 • SELECT * FROM new_actor_table;
10
11 • SELECT * FROM categories_table;
12
13 • select distinct(name) from (select a.name, categories, count(distinct actors), count(distinct writer) from new_actor_table a
14 join writer_table w
15 on w.name=a.name
16 join categories_table c
17 on c.name=a.name
18 group by a.name, categories
19 having count(distinct actors)>3 and count(distinct writer)>4) a

```

Result Grid

name
Ant-Man
Ant-Man and the Wasp
Fantastic 4: Rise of the Silver Surfer
Ghost in the Shell
Moana
Snailfoot
Spiderman Homecoming
Terminator 3: Rise of the Machines
The Lost World: Jurassic Park
The Magnificent Seven

Result 28 x

Output

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Read Only Context Help Snippets

MySQL Workbench

Talend_JSON (json) x Talend_JSON (json) x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHEMAS

- actor
- air_traffic
- json
 - Tables
 - actor_table
 - categories_table
 - director_table
 - movie
 - new_actor_table
 - writer_table
 - Views
 - Stored Procedures
 - Functions
- quiz
- seattle_pet_license
- seattle_pet_licenses
- seattle_pet_licenses2

Administration Schemas

Information

Table: actor_table

Columns:

- name varchar(100)
- birthname varchar(100)
- birthdate datetime
- birthplace varchar(100)
- ActorID int PK

Query 1 categories_table

```

3 SELECT * FROM director_table;
4
5 SELECT * FROM movie;
6
7 SELECT * FROM writer_table;
8
9 SELECT * FROM new_actor_table;
10
11 SELECT * FROM categories_table;
12
13 select distinct(name) from (select a.name, categories, count(distinct actors), count(distinct writer) from new_actor_table a
14 join writer_table w
15 on w.name=a.name
16 join categories_table c
17 on c.name=a.name
18 group by a.name, categories
19 having count(distinct actors)>3 and count(distinct writer)>3) a
  
```

Result Grid

name	categories	count(distinct actors)	count(distinct writer)
Ant-Man	action	4	6
Ant-Man	adventure	4	6
Ant-Man	comedy	4	6
Ant-Man and the Wasp	action	5	5
Ant-Man and the Wasp	adventure	5	5
Ant-Man and the Wasp	sci-fi	5	5
Fantastic 4: Rise of the Silver Surfer	action	4	5
Fantastic 4: Rise of the Silver Surfer	adventure	4	5
Fantastic 4: Rise of the Silver Surfer	sci-fi	4	5
Ghost in the Shell	action	8	5

Result 29 x

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Read Only Context Help Snippets

MySQL Workbench

Talend_JSON (json) x Talend_JSON (json) x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHEMAS

- actor
- air_traffic
- json
 - Tables
 - actor_table
 - categories_table
 - director_table
 - movie
 - new_actor_table
 - writer_table
 - Views
 - Stored Procedures
 - Functions
- quiz
- seattle_pet_license
- seattle_pet_licenses
- seattle_pet_licenses2

Administration Schemas

Information

Table: actor_table

Columns:

- name varchar(100)
- birthname varchar(100)
- birthdate datetime
- birthplace varchar(100)
- ActorID int PK

Query 1 categories_table SQL File 1*

```

11 SELECT COUNT(*) FROM movie;
12
13 SELECT * FROM writer_table;
14
15 SELECT COUNT(*) FROM writer_table;
16
17 SELECT * FROM new_actor_table;
18
19 SELECT COUNT(*) FROM new_actor_table;
20
21 SELECT * FROM categories_table;
22
23 SELECT COUNT(*) FROM categories_table;
24
25 select year, name from movie order by year, name;
  
```

Result Grid

year	name
1962	Dr. No
1962	To Kill a Mockingbird
1962	Whatever Happened To Baby Jane?
1963	From Russia with Love
1964	Goldfinger
1965	The Sound of Music
1965	Thunderball
1966	The Good, the Bad and the Ugly
1967	You Only Live Twice
1971	Diamonds Are Forever
1972	The Godfather
1974	La Patagonia rebelde
1974	The Godfather: Part II

movie 38 x

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Read Only Context Help Snippets