National University
of computer and emerging sciences

DATA BASE assignment 03:

SE-Q

Submitted to:

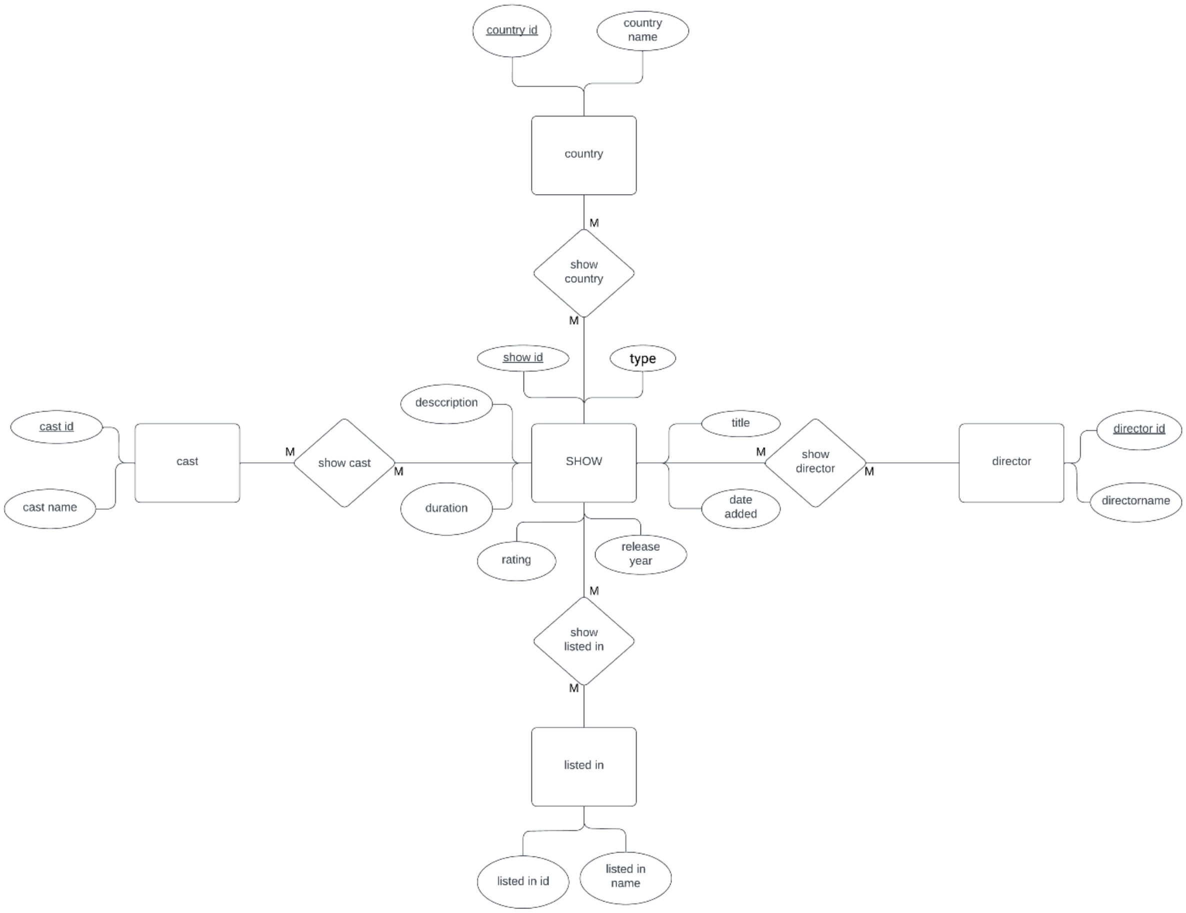
MAAM JAVARIA IMTIAZ.

GROUP MEMBERS:

Taimoor Anwar i211232.

Ghulam Mohaiuddin i211130.

ERD


country id
country name
country
M
show country
M
show id
type
desccription
duration
SHOW
title
date added
rating
release year
cast id
cast name
cast
M
show cast
M
M
show director
M
director
director id
directorname
M
show listed in
M
listed in
listed in id
listed in name

TABLES and ATTRIBUTES:

1. Show

show_id (VARCHAR(50), PRIMARY KEY)

type (VARCHAR(50))

title (VARCHAR(100))

date_added (VARCHAR(50))

release_year (INT)

rating (VARCHAR(50))

duration (VARCHAR(50))

description (VARCHAR(MAX))

2. Director

director_id (INT, PRIMARY KEY)

director_name (VARCHAR(MAX))

3. Show_Director (junction table for Show and Director)

show_id (VARCHAR(50), FOREIGN KEY REFERENCES Show(show_id))

director_id (INT, FOREIGN KEY REFERENCES Director(director_id))

PRIMARY KEY (show_id, director_id)

4. Cast

cast_id (INT, PRIMARY KEY)

cast_name (VARCHAR(MAX))

5. Show_Cast (junction table for Show and Cast)

show_id (VARCHAR(50), FOREIGN KEY REFERENCES Show(show_id))

cast_id (INT, FOREIGN KEY REFERENCES Cast(cast_id))

PRIMARY KEY (show_id, cast_id)

6. Country

country_id (INT, PRIMARY KEY)

country_name (VARCHAR(MAX))

7. Show_Country (junction table for Show and Country)

show_id (VARCHAR(50), FOREIGN KEY REFERENCES Show(show_id))

country_id (INT, FOREIGN KEY REFERENCES Country(country_id))

PRIMARY KEY (show_id, country_id)

8. Listed_In

listed_in_id (INT, PRIMARY KEY)

listed_in_name (VARCHAR(MAX))

9. Show_Listed_In (junction table for Show and Listed_In)

show_id (VARCHAR(50), FOREIGN KEY REFERENCES Show(show_id))

listed_in_id (INT, FOREIGN KEY REFERENCES Listed_In(listed_in_id))

PRIMARY KEY (show_id, listed_in_id)

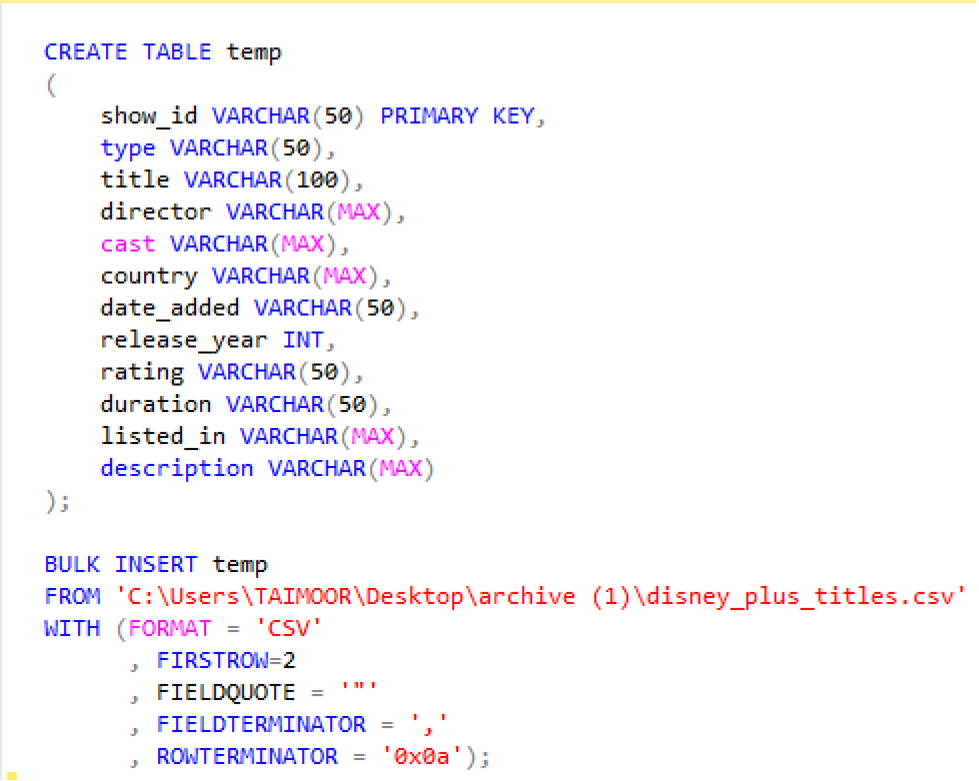
EXPLANATION:

- The Show table is the central table in this database schema.
- It is connected to the Director table via the Show_Director table, indicating a many-to-many relationship since a show can have multiple directors, and a director can direct multiple shows.
- It is connected to the Cast table via the Show_Cast table, which also indicates a many-to-many relationship since a show can have multiple cast members, and a cast member can be in multiple shows.
- It is connected to the Country table via the Show_Country table, indicating a many-to-many relationship as a show can be available in multiple countries, and a country can have multiple shows.
- It is connected to the Listed_In table via the Show_Listed_In table, which indicates a many-to-many relationship since a show can be listed in multiple categories, and a category can have multiple shows.

TABLE CREATION:

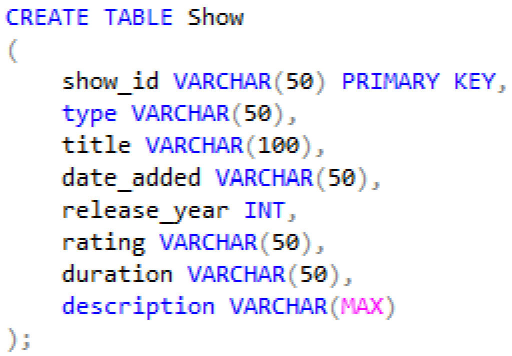
TEMP:

WE CREATE TEMPORARY TABLE SO STORE WHOLE DATA OF CSV AND THEN WE WILL STORE DATA INTO DEIFFERENT TABLE FROM TEMPORARY TABLE.



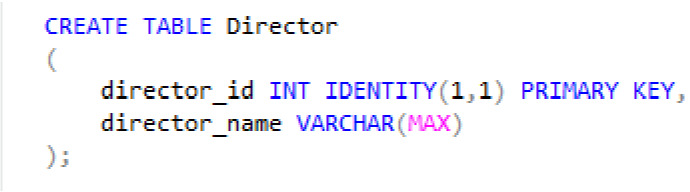
```sql
CREATE TABLE temp
(
    show_id VARCHAR(50) PRIMARY KEY,
    type VARCHAR(50),
    title VARCHAR(100),
    director VARCHAR(MAX),
    cast VARCHAR(MAX),
    country VARCHAR(MAX),
    date_added VARCHAR(50),
    release_year INT,
    rating VARCHAR(50),
    duration VARCHAR(50),
    listed_in VARCHAR(MAX),
    description VARCHAR(MAX)
);

BULK INSERT temp
FROM 'C:\Users\TAIMOOR\Desktop\archive (1)\disney_plus_titles.csv'
WITH (FORMAT = 'CSV'
      , FIRSTROW=2
      , FIELDQUOTE = '"'
      , FIELDTERMINATOR = ','
      , ROWTERMINATOR = '0x0a');
```

SHOW:



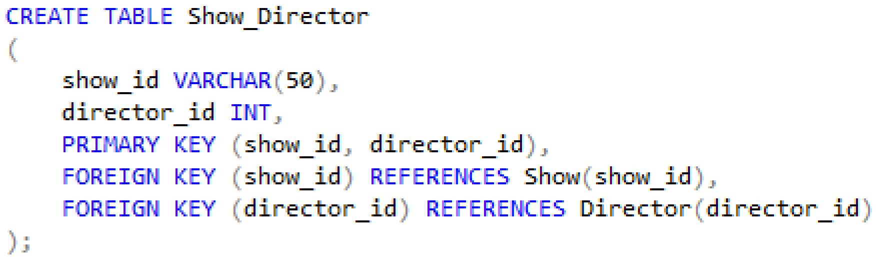
```
CREATE TABLE Show
(
    show_id VARCHAR(50) PRIMARY KEY,
    type VARCHAR(50),
    title VARCHAR(100),
    date_added VARCHAR(50),
    release_year INT,
    rating VARCHAR(50),
    duration VARCHAR(50),
    description VARCHAR(MAX)
);
```

DIRECTOR:



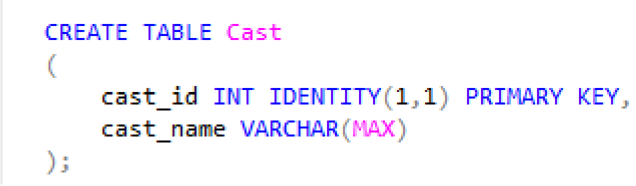
```
CREATE TABLE Director
(
    director_id INT IDENTITY(1,1) PRIMARY KEY,
    director_name VARCHAR(MAX)
);
```

SHOW_DIRECTOR:



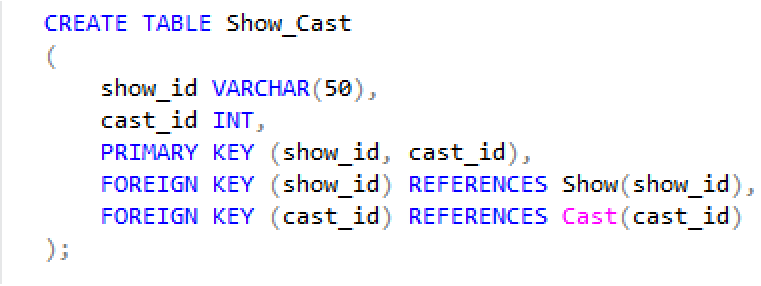
```
CREATE TABLE Show_Director
(
    show_id VARCHAR(50),
    director_id INT,
    PRIMARY KEY (show_id, director_id),
    FOREIGN KEY (show_id) REFERENCES Show(show_id),
    FOREIGN KEY (director_id) REFERENCES Director(director_id)
);
```

CAST:



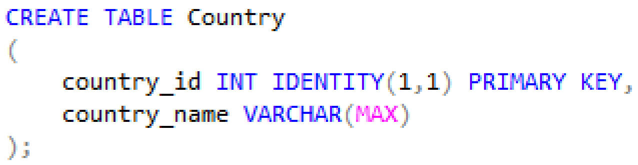
```
CREATE TABLE Cast
(
    cast_id INT IDENTITY(1,1) PRIMARY KEY,
    cast_name VARCHAR(MAX)
);
```

SHOW_CAST:



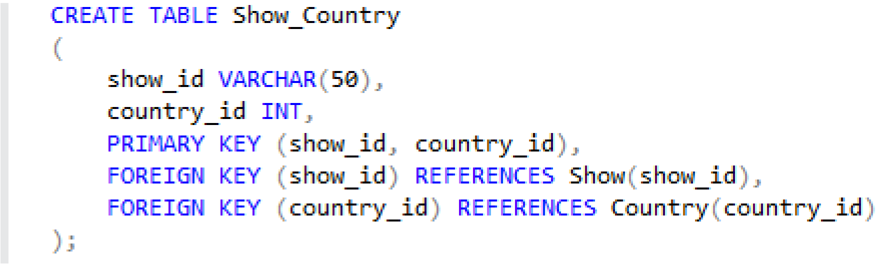
```
CREATE TABLE Show_Cast
(
    show_id VARCHAR(50),
    cast_id INT,
    PRIMARY KEY (show_id, cast_id),
    FOREIGN KEY (show_id) REFERENCES Show(show_id),
    FOREIGN KEY (cast_id) REFERENCES Cast(cast_id)
);
```

COUNTRY:



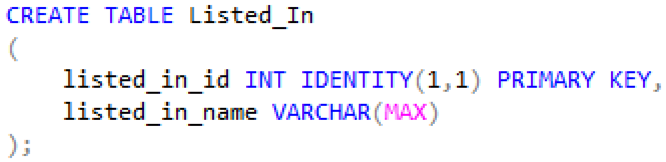
```
CREATE TABLE Country
(
    country_id INT IDENTITY(1,1) PRIMARY KEY,
    country_name VARCHAR(MAX)
);
```

SHOW_COUNTRY:



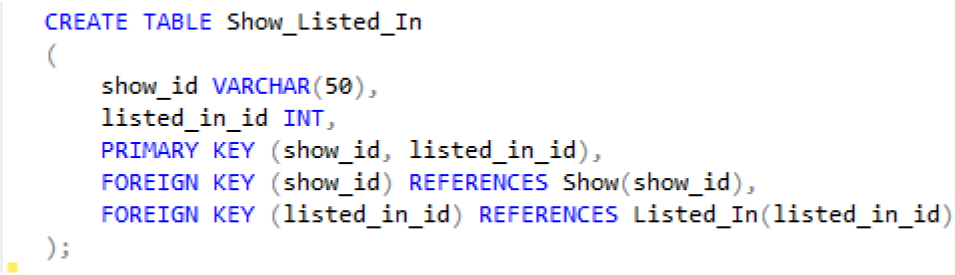
```
CREATE TABLE Show_Country
(
    show_id VARCHAR(50),
    country_id INT,
    PRIMARY KEY (show_id, country_id),
    FOREIGN KEY (show_id) REFERENCES Show(show_id),
    FOREIGN KEY (country_id) REFERENCES Country(country_id)
);
```

LISTED_IN:



```
CREATE TABLE Listed_In
(
    listed_in_id INT IDENTITY(1,1) PRIMARY KEY,
    listed_in_name VARCHAR(MAX)
);
```

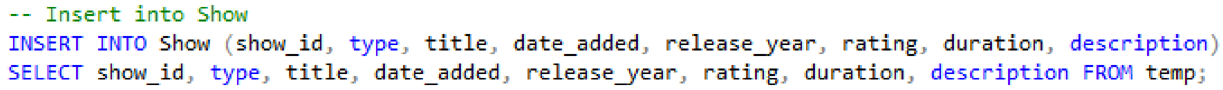
SHOW_LISTED_IN:



```
CREATE TABLE Show_Listed_In
(
    show_id VARCHAR(50),
    listed_in_id INT,
    PRIMARY KEY (show_id, listed_in_id),
    FOREIGN KEY (show_id) REFERENCES Show(show_id),
    FOREIGN KEY (listed_in_id) REFERENCES Listed_In(listed_in_id)
);
```

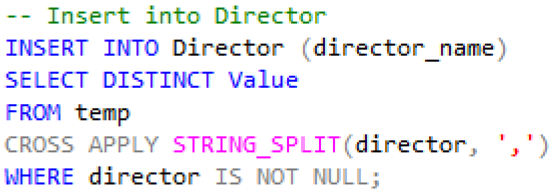
INSERTION:

SHOW:



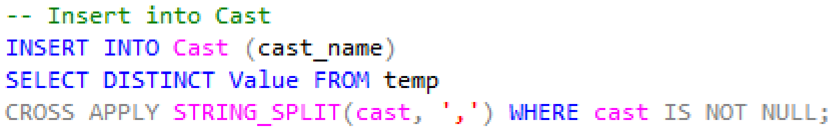
```
-- Insert into Show
INSERT INTO Show (show_id, type, title, date_added, release_year, rating, duration, description)
SELECT show_id, type, title, date_added, release_year, rating, duration, description FROM temp;
```

DIRECTOR:



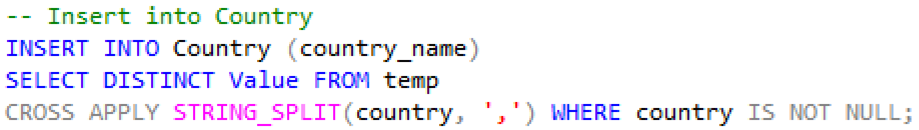
```
-- Insert into Director
INSERT INTO Director (director_name)
SELECT DISTINCT Value
FROM temp
CROSS APPLY STRING_SPLIT(director, ',')
WHERE director IS NOT NULL;
```

CAST:



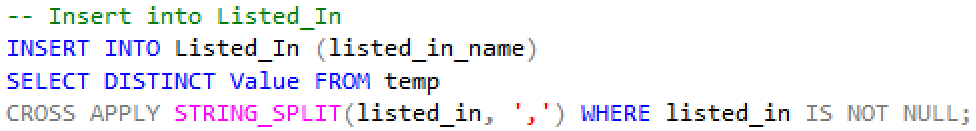
```
-- Insert into Cast
INSERT INTO Cast (cast_name)
SELECT DISTINCT Value FROM temp
CROSS APPLY STRING_SPLIT(cast, ',') WHERE cast IS NOT NULL;
```

COUNTRY:



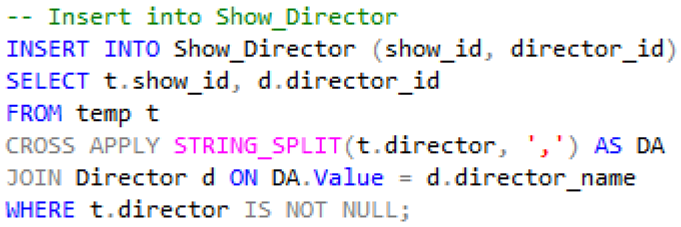
```
-- Insert into Country
INSERT INTO Country (country_name)
SELECT DISTINCT Value FROM temp
CROSS APPLY STRING_SPLIT(country, ',') WHERE country IS NOT NULL;
```

LISTED_IN:



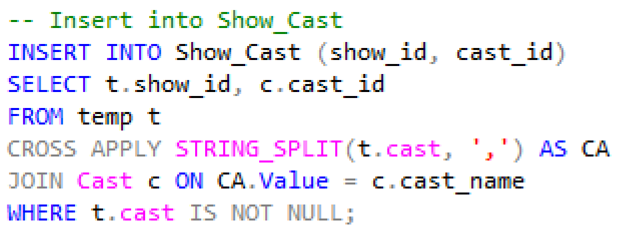
```
-- Insert into Listed_In
INSERT INTO Listed_In (listed_in_name)
SELECT DISTINCT Value FROM temp
CROSS APPLY STRING_SPLIT(listed_in, ',') WHERE listed_in IS NOT NULL;
```

SHOW_DIRECTOR:



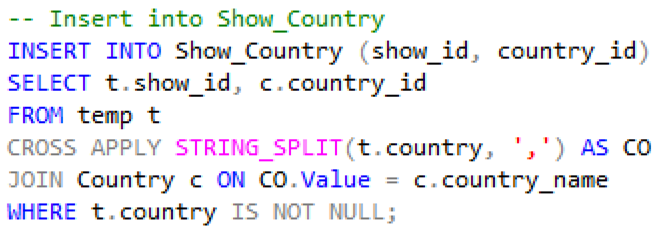
```
-- Insert into Show_Director
INSERT INTO Show_Director (show_id, director_id)
SELECT t.show_id, d.director_id
FROM temp t
CROSS APPLY STRING_SPLIT(t.director, ',') AS DA
JOIN Director d ON DA.Value = d.director_name
WHERE t.director IS NOT NULL;
```

SHOW_CAST:



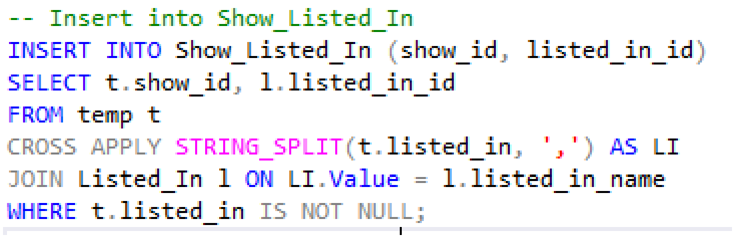
```sql
-- Insert into Show_Cast
INSERT INTO Show_Cast (show_id, cast_id)
SELECT t.show_id, c.cast_id
FROM temp t
CROSS APPLY STRING_SPLIT(t.cast, ',') AS CA
JOIN Cast c ON CA.Value = c.cast_name
WHERE t.cast IS NOT NULL;
```

SHOW_COUNTRY:



```sql
-- Insert into Show_Country
INSERT INTO Show_Country (show_id, country_id)
SELECT t.show_id, c.country_id
FROM temp t
CROSS APPLY STRING_SPLIT(t.country, ',') AS CO
JOIN Country c ON CO.Value = c.country_name
WHERE t.country IS NOT NULL;
```

SHOW_LISTED_IN:



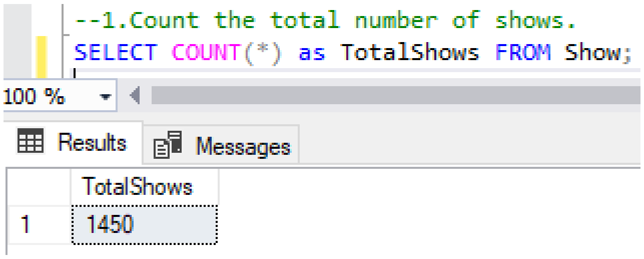
```sql
-- Insert into Show_Listed_In
INSERT INTO Show_Listed_In (show_id, listed_in_id)
SELECT t.show_id, l.listed_in_id
FROM temp t
CROSS APPLY STRING_SPLIT(t.listed_in, ',') AS LI
JOIN Listed_In l ON LI.Value = l.listed_in_name
WHERE t.listed_in IS NOT NULL;
```

QUERIES:

"As the results are very lengthy, we will only capture a snapshot of the first 10 rows."

i. 3 Queries which use various aggregation functions

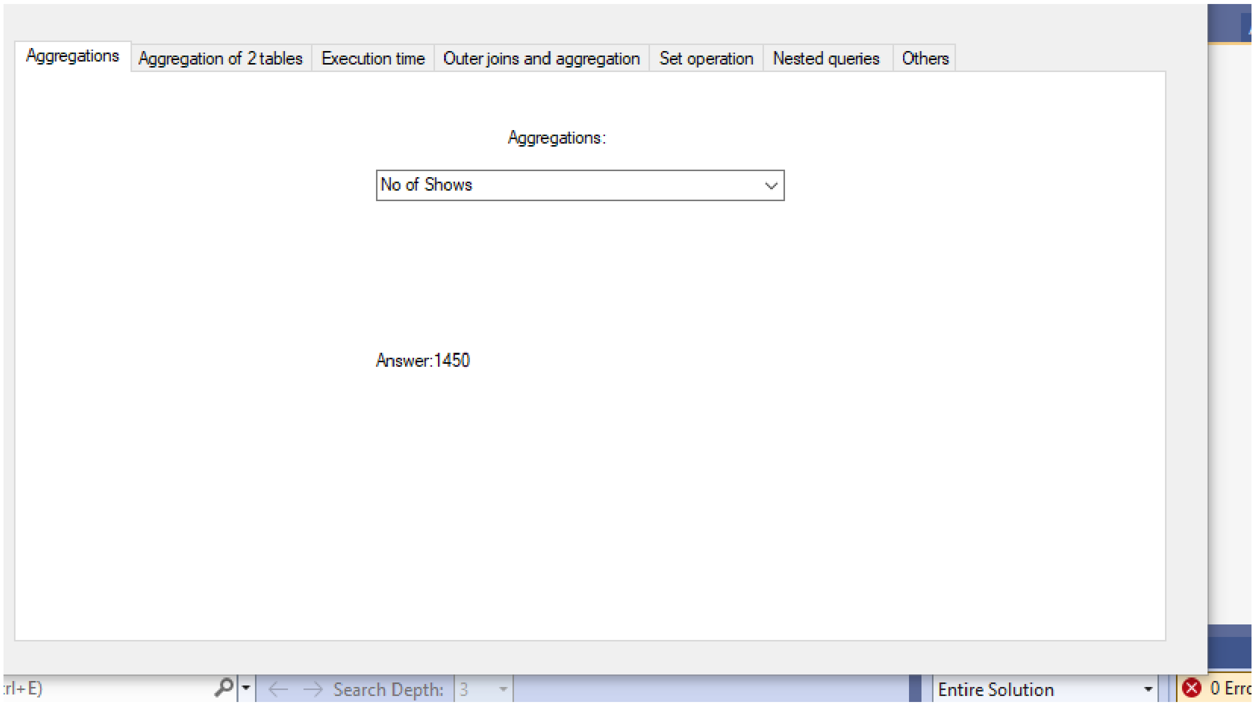
1. Count the total number of shows.



```sql
--1.Count the total number of shows.
SELECT COUNT(*) as TotalShows FROM Show;
```

	TotalShows
1	1450

UI:

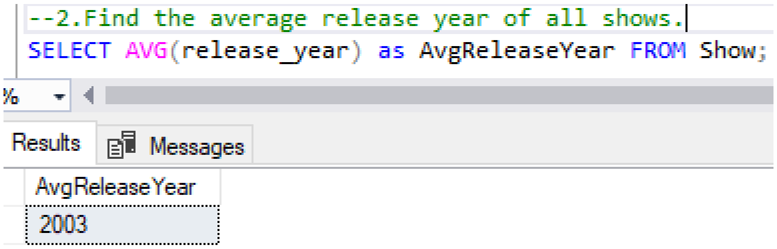


Aggregations:

No of Shows

Answer:1450

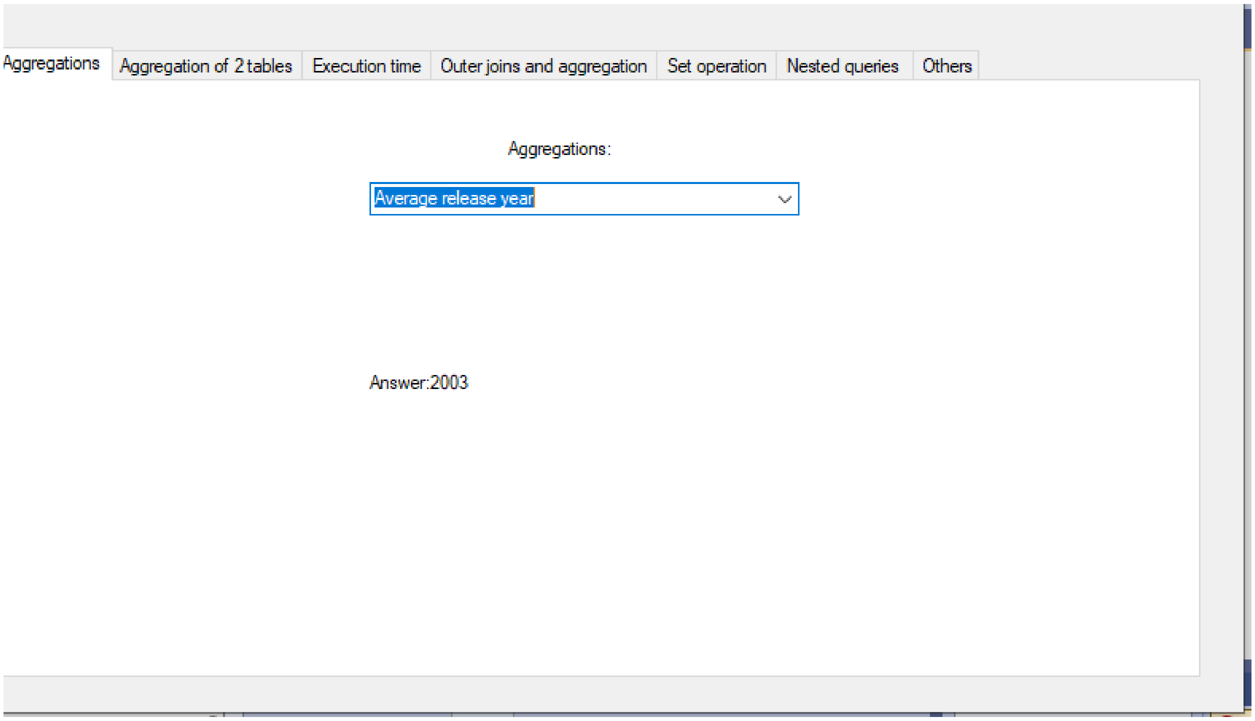
2. Find the average release year of all shows.



```
--2.Find the average release year of all shows.
SELECT AVG(release_year) as AvgReleaseYear FROM Show;
```

AvgReleaseYear
2003

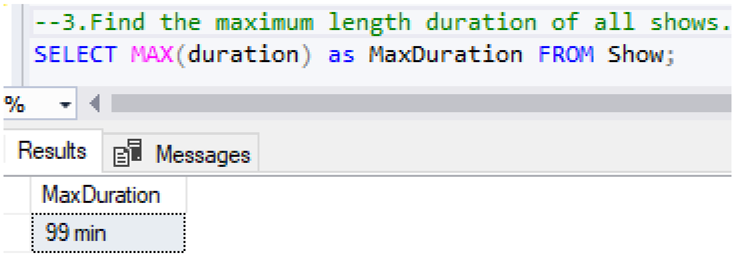
UI:



Aggregations: Average release year

Answer:2003

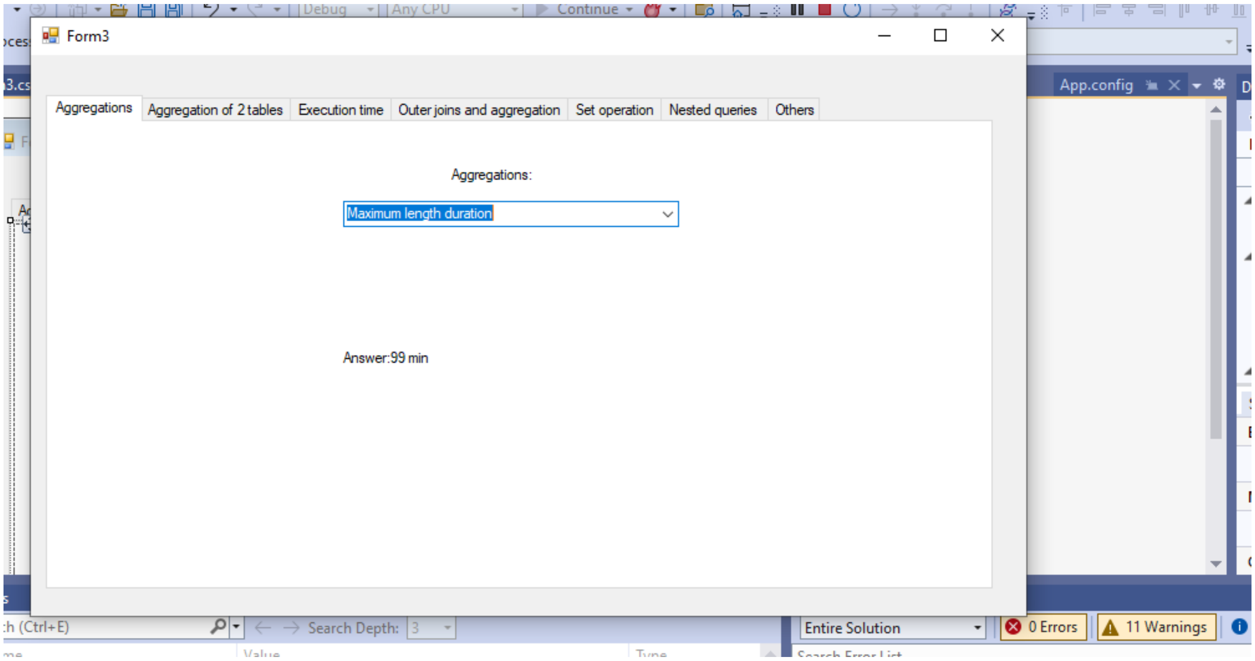
3. **Find the maximum length duration of all shows.**



```
--3.Find the maximum length duration of all shows.
SELECT MAX(duration) as MaxDuration FROM Show;
```

MaxDuration
99 min

UI:



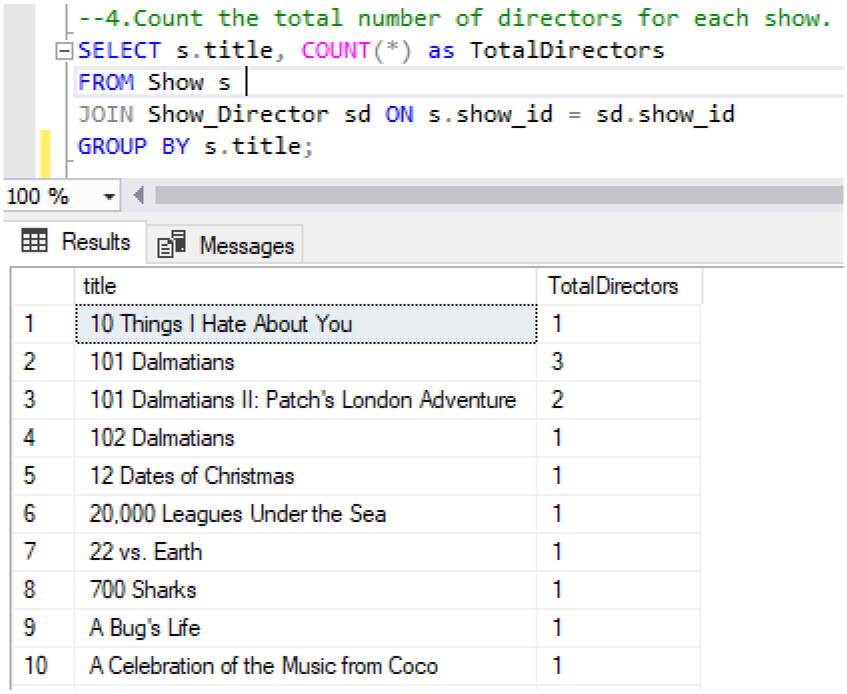
Aggregations:

Maximum length duration

Answer:99 min

ii. **3 Queries which use various aggregation functions and involve more than 2 tables (database join).**

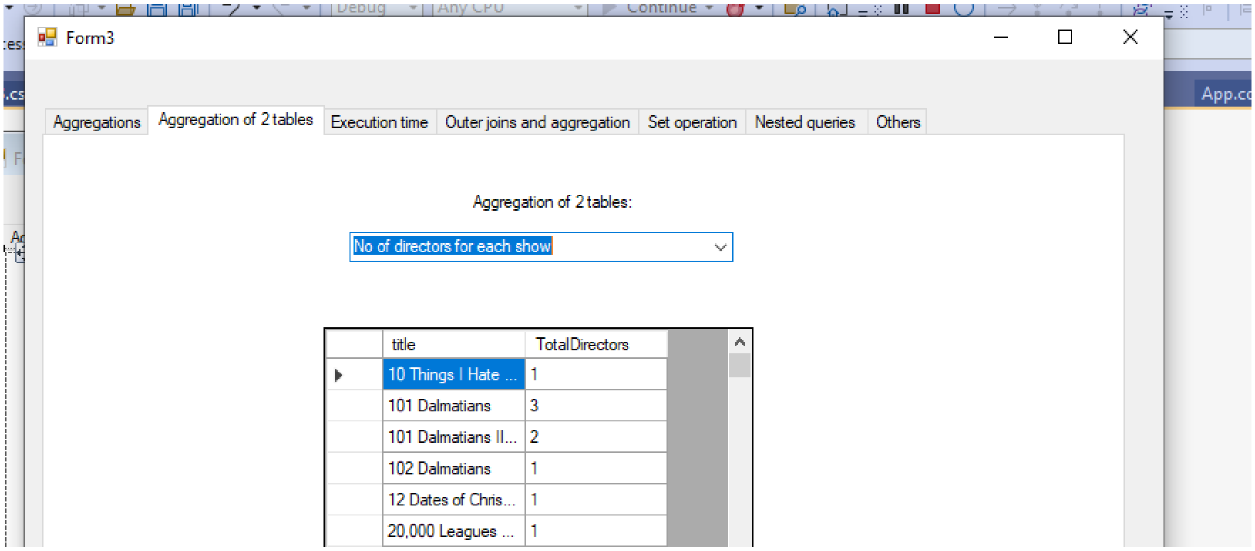
4. Count the total number of directors for each show.



```
--4.Count the total number of directors for each show.
SELECT s.title, COUNT(*) as TotalDirectors
FROM Show s
JOIN Show_Director sd ON s.show_id = sd.show_id
GROUP BY s.title;
```

	title	TotalDirectors
1	10 Things I Hate About You	1
2	101 Dalmatians	3
3	101 Dalmatians II: Patch's London Adventure	2
4	102 Dalmatians	1
5	12 Dates of Christmas	1
6	20,000 Leagues Under the Sea	1
7	22 vs. Earth	1
8	700 Sharks	1
9	A Bug's Life	1
10	A Celebration of the Music from Coco	1

UI:

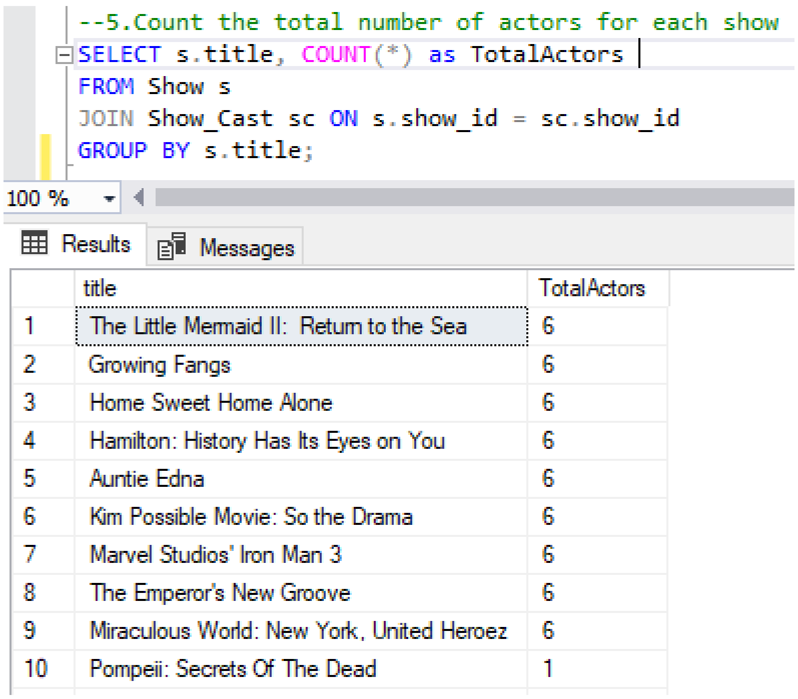


Aggregation of 2 tables:

No of directors for each show

title	TotalDirectors
10 Things I Hate ...	1
101 Dalmatians	3
101 Dalmatians II...	2
102 Dalmatians	1
12 Dates of Chris...	1
20,000 Leagues ...	1

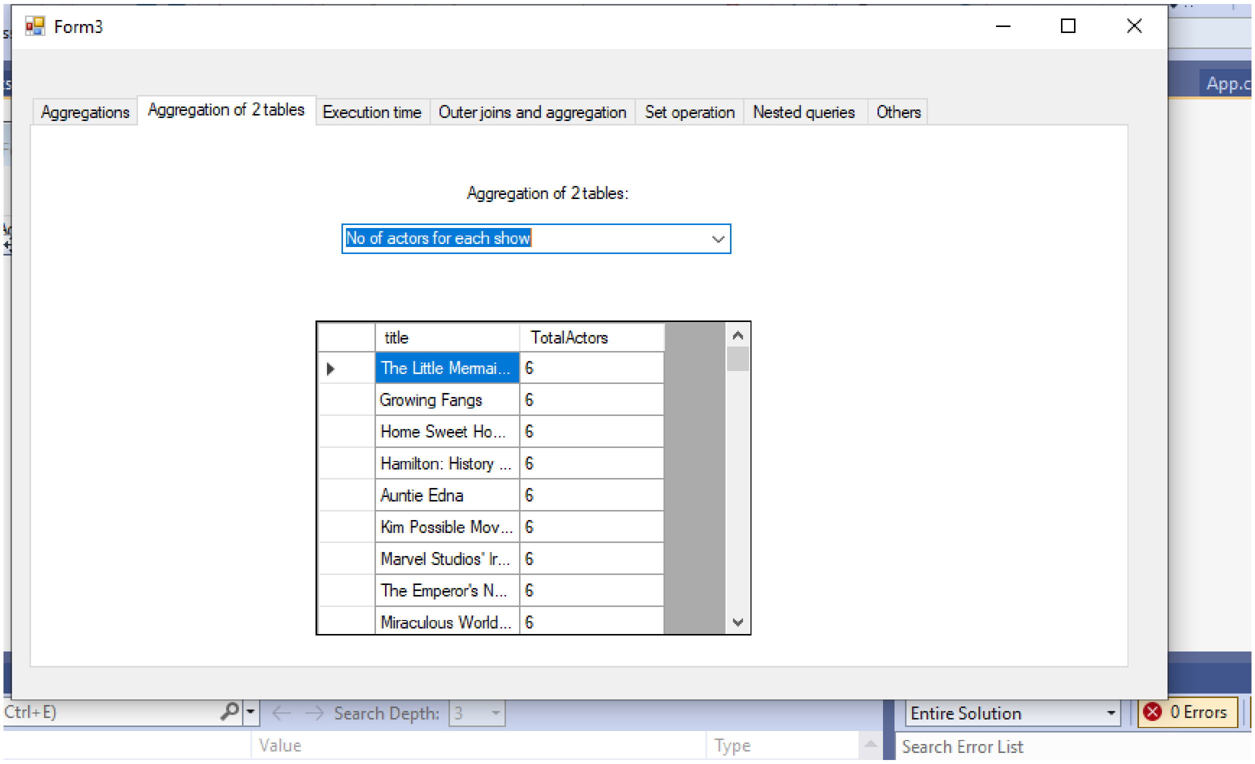
5. Count the total number of actors for each show.



```
--5.Count the total number of actors for each show
SELECT s.title, COUNT(*) as TotalActors
FROM Show s
JOIN Show_Cast sc ON s.show_id = sc.show_id
GROUP BY s.title;
```

	title	TotalActors
1	The Little Mermaid II: Return to the Sea	6
2	Growing Fangs	6
3	Home Sweet Home Alone	6
4	Hamilton: History Has Its Eyes on You	6
5	Auntie Edna	6
6	Kim Possible Movie: So the Drama	6
7	Marvel Studios' Iron Man 3	6
8	The Emperor's New Groove	6
9	Miraculous World: New York, United Heroez	6
10	Pompeii: Secrets Of The Dead	1

UI:

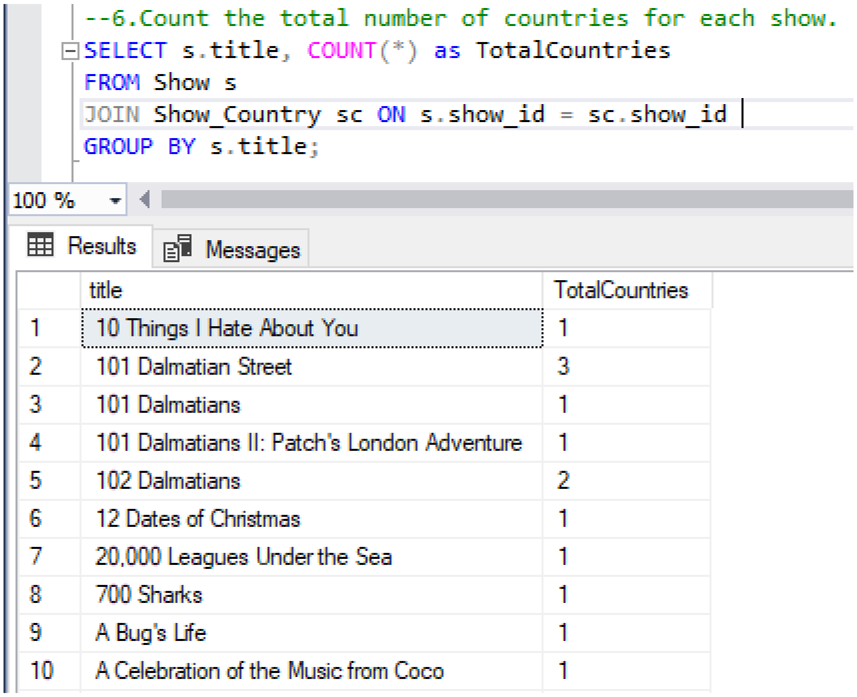


Aggregation of 2 tables:

No of actors for each show

title	TotalActors
The Little Mermai...	6
Growing Fangs	6
Home Sweet Ho...	6
Hamilton: History ...	6
Auntie Edna	6
Kim Possible Mov...	6
Marvel Studios' Ir...	6
The Emperor's N...	6
Miraculous World...	6

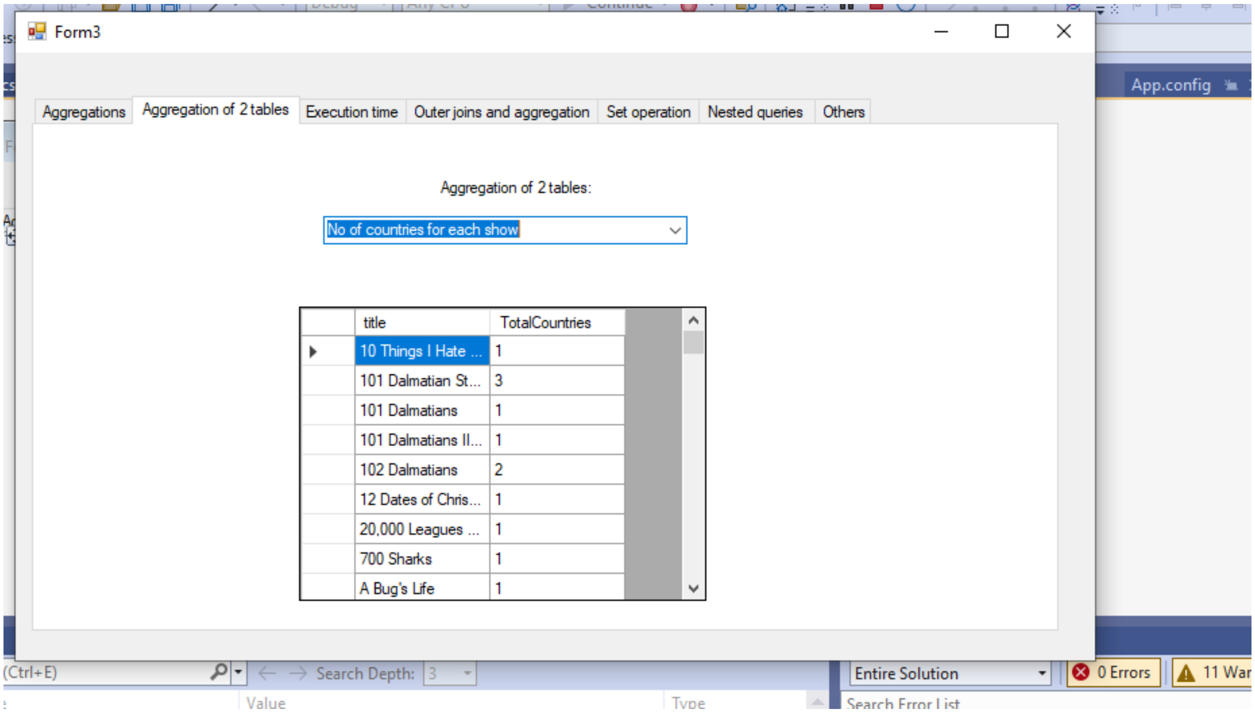
6. Count the total number of countries for each show.



```
--6.Count the total number of countries for each show.
SELECT s.title, COUNT(*) as TotalCountries
FROM Show s
JOIN Show_Country sc ON s.show_id = sc.show_id
GROUP BY s.title;
```

	title	TotalCountries
1	10 Things I Hate About You	1
2	101 Dalmatian Street	3
3	101 Dalmatians	1
4	101 Dalmatians II: Patch's London Adventure	1
5	102 Dalmatians	2
6	12 Dates of Christmas	1
7	20,000 Leagues Under the Sea	1
8	700 Sharks	1
9	A Bug's Life	1
10	A Celebration of the Music from Coco	1

UI:

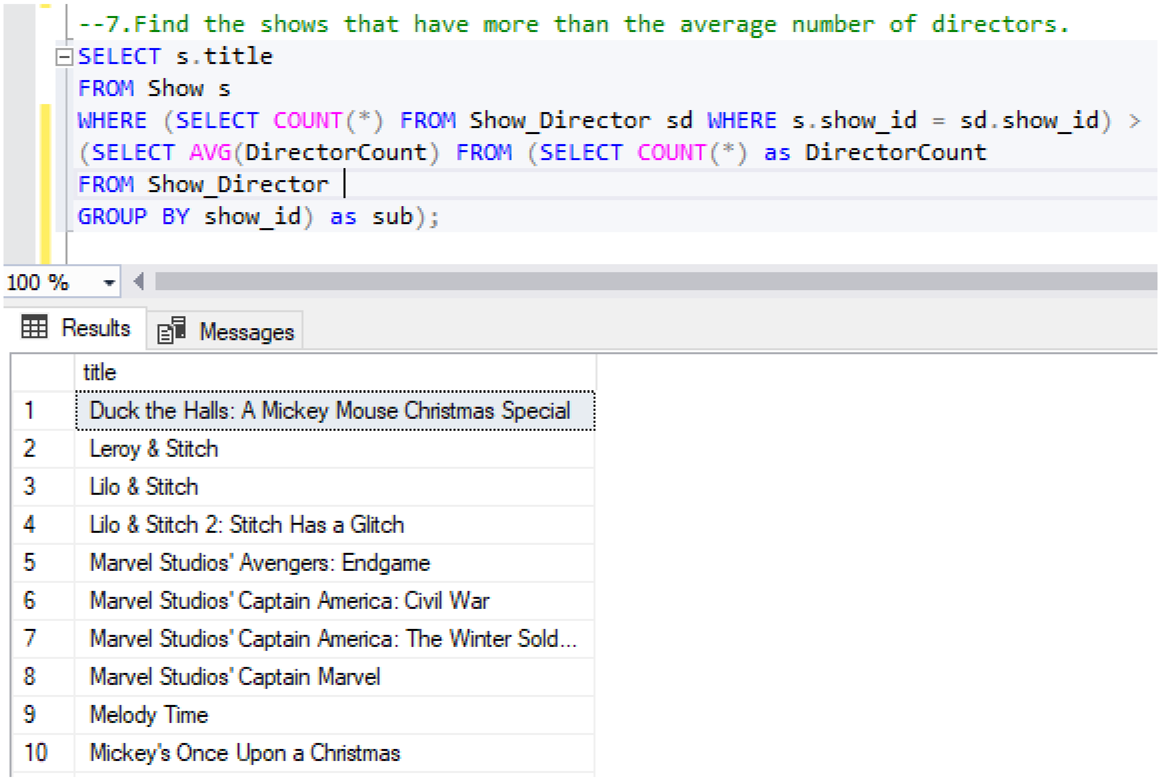


Aggregation of 2 tables: No of countries for each show

title	TotalCountries
10 Things I Hate ...	1
101 Dalmatian St...	3
101 Dalmatians	1
101 Dalmatians II...	1
102 Dalmatians	2
12 Dates of Chris...	1
20,000 Leagues ...	1
700 Sharks	1
A Bug's Life	1

iii. 3 Nested queries

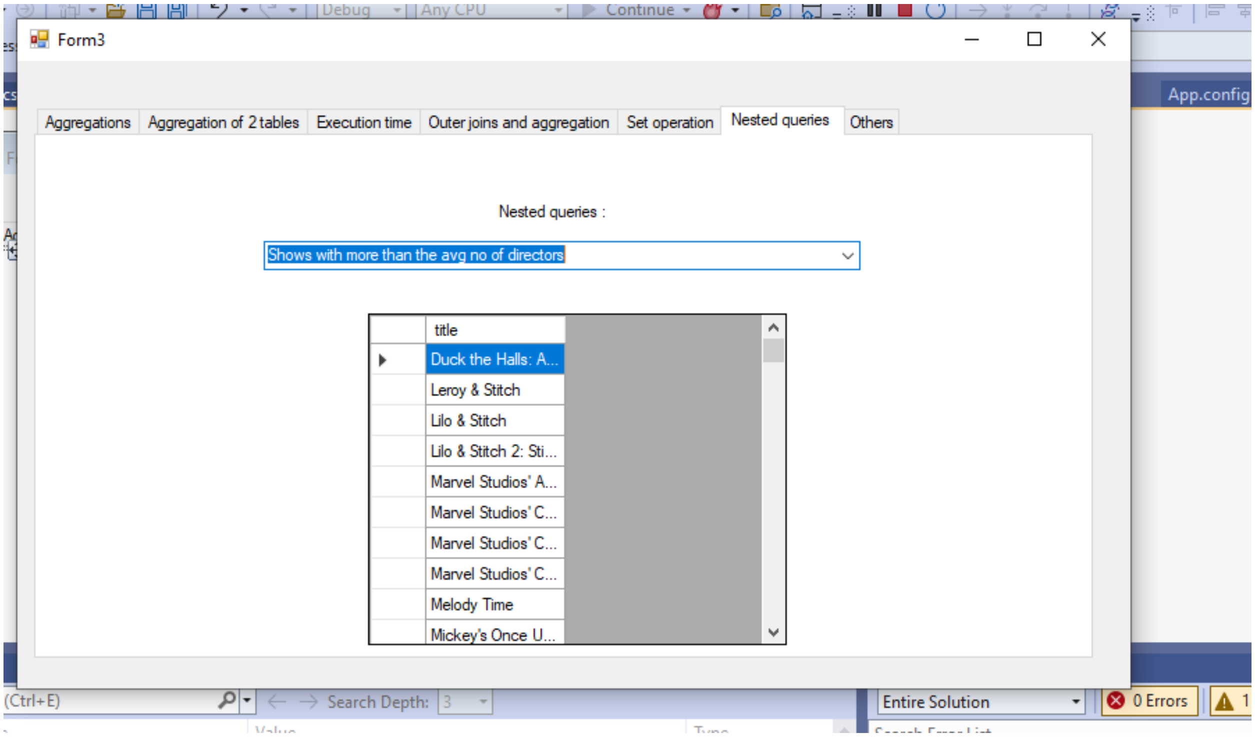
7. Find the shows that have more than the average number of directors.



```
--7.Find the shows that have more than the average number of directors.
SELECT s.title
FROM Show s
WHERE (SELECT COUNT(*) FROM Show_Director sd WHERE s.show_id = sd.show_id) >
(SELECT AVG(DirectorCount) FROM (SELECT COUNT(*) as DirectorCount
FROM Show_Director
GROUP BY show_id) as sub);
```

	title
1	Duck the Halls: A Mickey Mouse Christmas Special
2	Leroy & Stitch
3	Lilo & Stitch
4	Lilo & Stitch 2: Stitch Has a Glitch
5	Marvel Studios' Avengers: Endgame
6	Marvel Studios' Captain America: Civil War
7	Marvel Studios' Captain America: The Winter Sold...
8	Marvel Studios' Captain Marvel
9	Melody Time
10	Mickey's Once Upon a Christmas

UI:

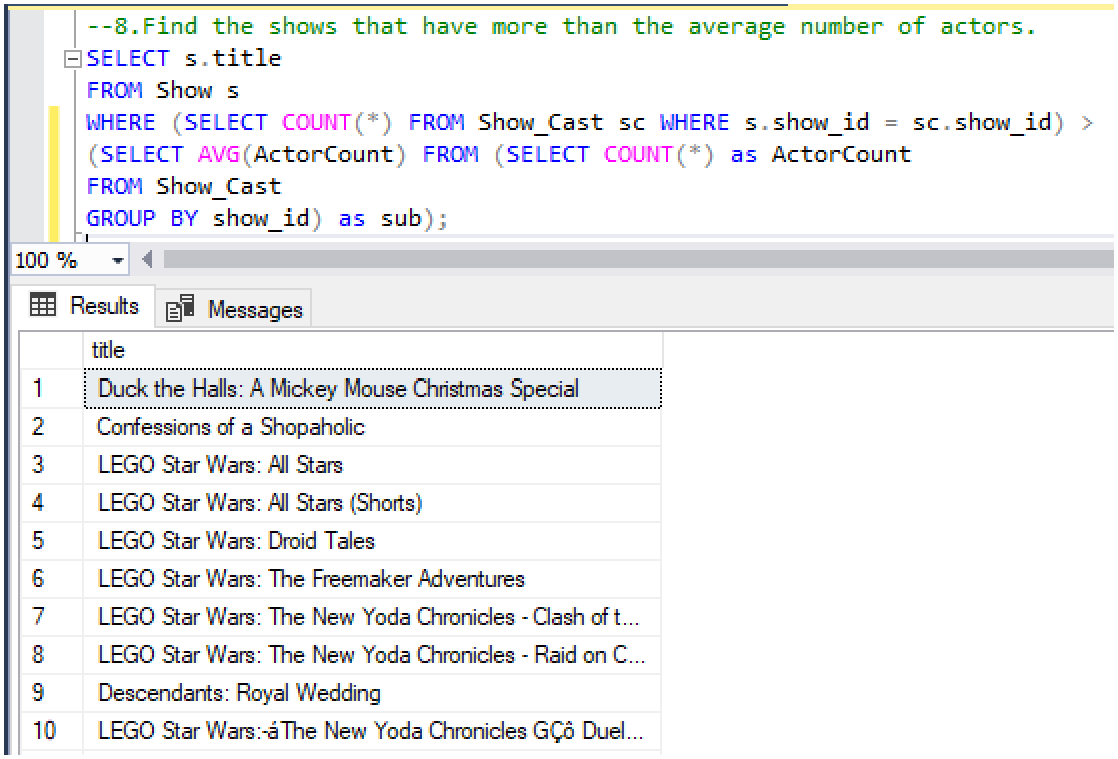


Nested queries :

Shows with more than the avg no of directors

	title
▶	Duck the Halls: A...
	Leroy & Stitch
	Lilo & Stitch
	Lilo & Stitch 2: Sti...
	Marvel Studios' A...
	Marvel Studios' C...
	Marvel Studios' C...
	Marvel Studios' C...
	Melody Time
	Mickey's Once U...

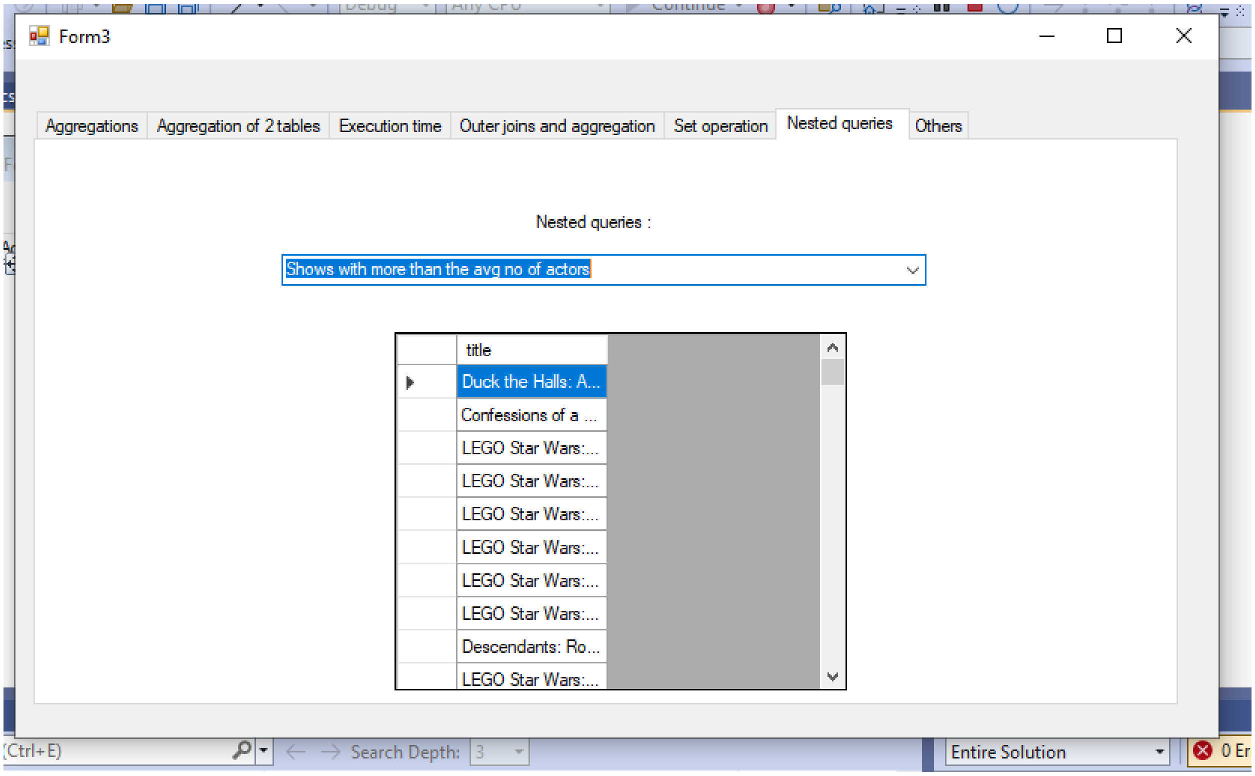
8. Find the shows that have more than the average number of actors.



```
--8.Find the shows that have more than the average number of actors.
SELECT s.title
FROM Show s
WHERE (SELECT COUNT(*) FROM Show_Cast sc WHERE s.show_id = sc.show_id) >
(SELECT AVG(ActorCount) FROM (SELECT COUNT(*) as ActorCount
FROM Show_Cast
GROUP BY show_id) as sub);
```

	title
1	Duck the Halls: A Mickey Mouse Christmas Special
2	Confessions of a Shopaholic
3	LEGO Star Wars: All Stars
4	LEGO Star Wars: All Stars (Shorts)
5	LEGO Star Wars: Droid Tales
6	LEGO Star Wars: The Freemaker Adventures
7	LEGO Star Wars: The New Yoda Chronicles - Clash of t...
8	LEGO Star Wars: The New Yoda Chronicles - Raid on C...
9	Descendants: Royal Wedding
10	LEGO Star Wars:-áThe New Yoda Chronicles GÇô Duel...

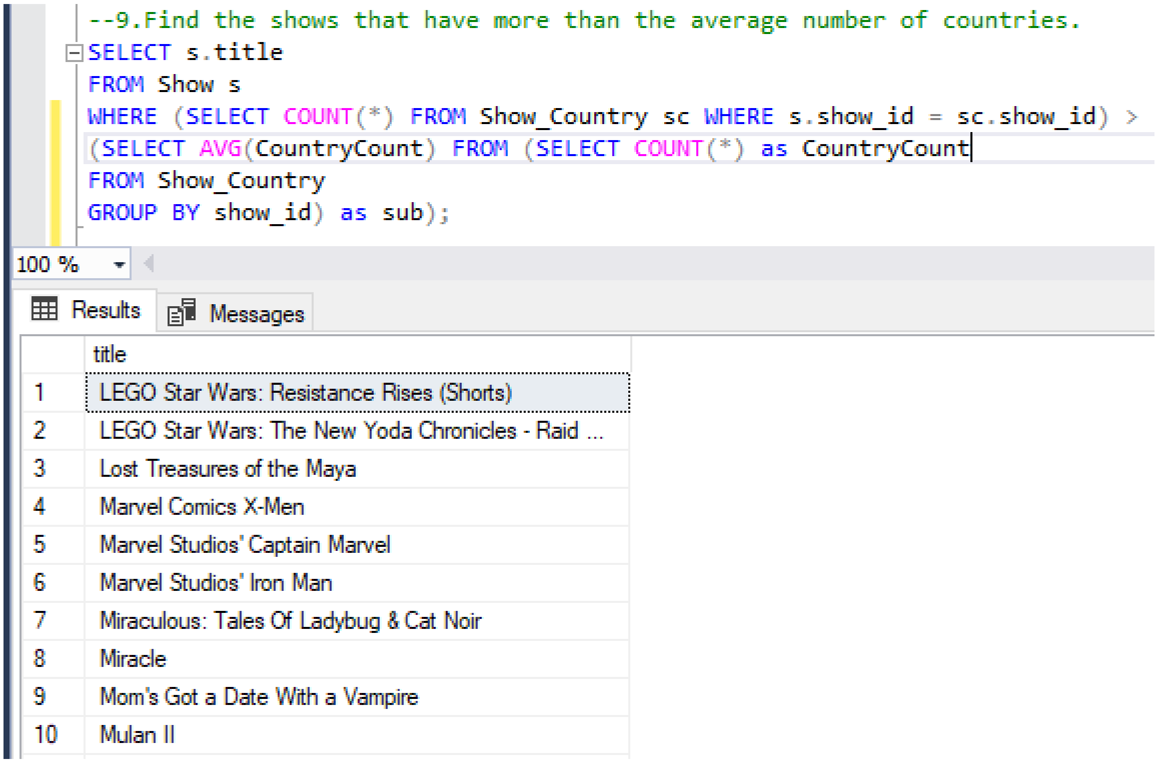
UI:



Nested queries : Shows with more than the avg no of actors

title
Duck the Halls: A...
Confessions of a ...
LEGO Star Wars:...
LEGO Star Wars:...
LEGO Star Wars:...
LEGO Star Wars:...
LEGO Star Wars:...
LEGO Star Wars:...
Descendants: Ro...
LEGO Star Wars:...

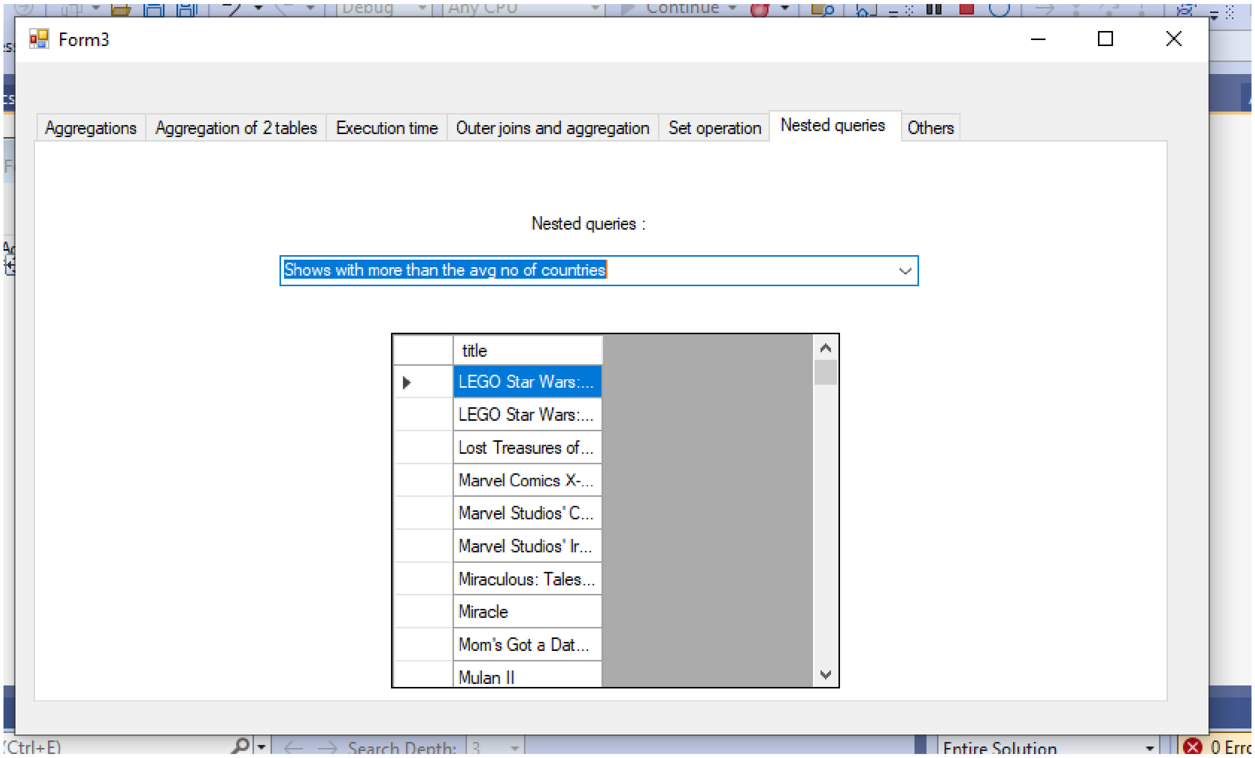
9. Find the shows that have more than the average number of countries.



```
--9.Find the shows that have more than the average number of countries.
SELECT s.title
FROM Show s
WHERE (SELECT COUNT(*) FROM Show_Country sc WHERE s.show_id = sc.show_id) >
(SELECT AVG(CountryCount) FROM (SELECT COUNT(*) as CountryCount
FROM Show_Country
GROUP BY show_id) as sub);
```

	title
1	LEGO Star Wars: Resistance Rises (Shorts)
2	LEGO Star Wars: The New Yoda Chronicles - Raid ...
3	Lost Treasures of the Maya
4	Marvel Comics X-Men
5	Marvel Studios' Captain Marvel
6	Marvel Studios' Iron Man
7	Miraculous: Tales Of Ladybug & Cat Noir
8	Miracle
9	Mom's Got a Date With a Vampire
10	Mulan II

UI:

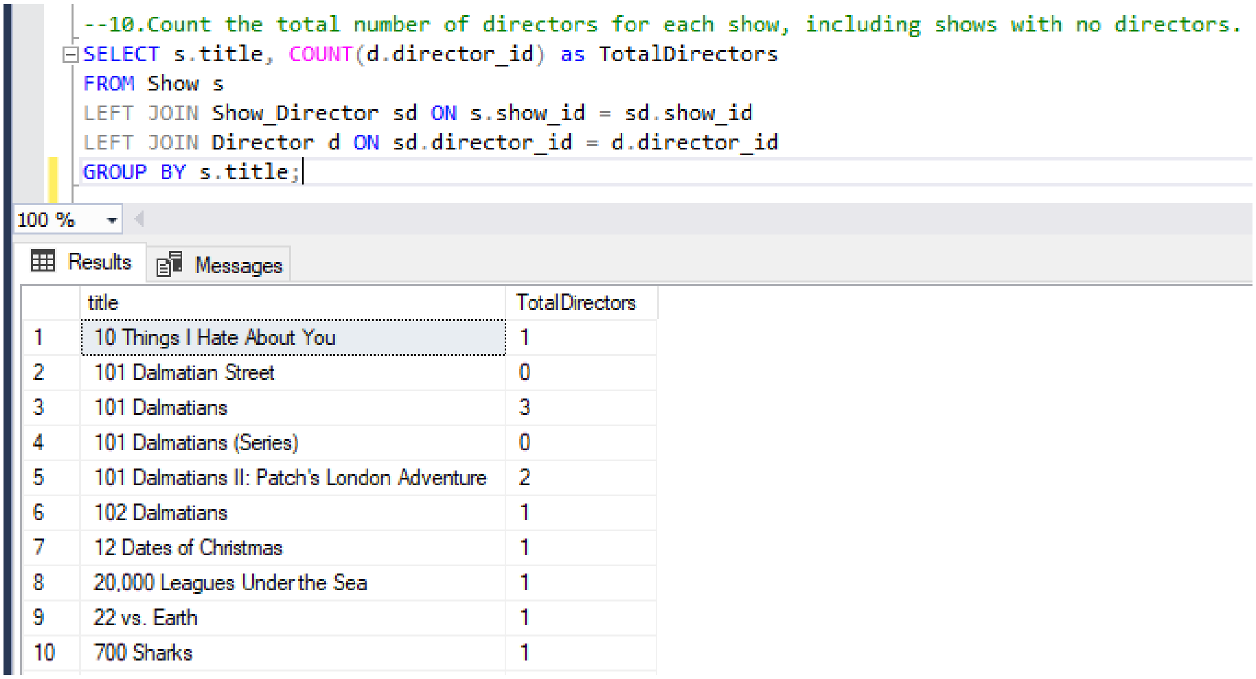


Nested queries : Shows with more than the avg no of countries

title
LEGO Star Wars:...
LEGO Star Wars:...
Lost Treasures of...
Marvel Comics X-...
Marvel Studios' C...
Marvel Studios' Ir...
Miraculous: Tales...
Miracle
Mom's Got a Dat...
Mulan II

iv. 3 queries involving outer joins and aggregation functions

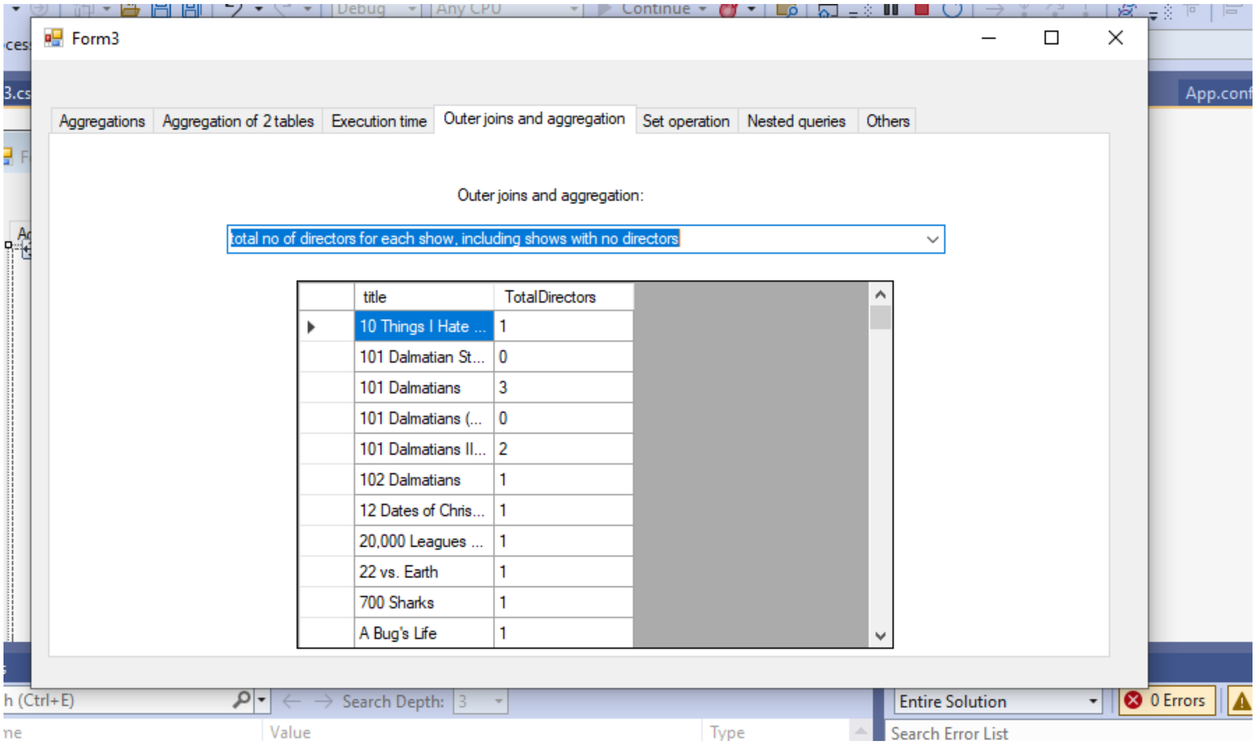
10. Count the total number of directors for each show, including shows with no directors.



```sql
--10.Count the total number of directors for each show, including shows with no directors.
SELECT s.title, COUNT(d.director_id) as TotalDirectors
FROM Show s
LEFT JOIN Show_Director sd ON s.show_id = sd.show_id
LEFT JOIN Director d ON sd.director_id = d.director_id
GROUP BY s.title;
```

	title	TotalDirectors
1	10 Things I Hate About You	1
2	101 Dalmatian Street	0
3	101 Dalmatians	3
4	101 Dalmatians (Series)	0
5	101 Dalmatians II: Patch's London Adventure	2
6	102 Dalmatians	1
7	12 Dates of Christmas	1
8	20,000 Leagues Under the Sea	1
9	22 vs. Earth	1
10	700 Sharks	1

UI:

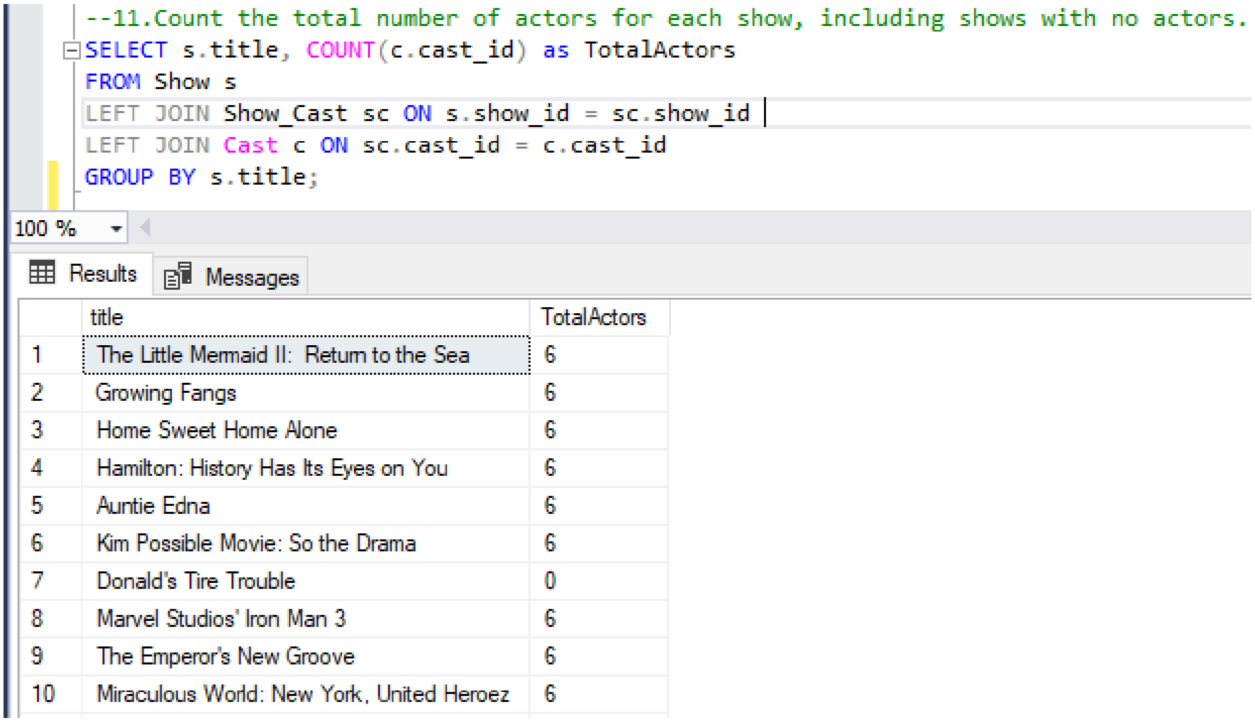


Outer joins and aggregation:

total no of directors for each show, including shows with no directors

title	TotalDirectors
10 Things I Hate ...	1
101 Dalmatian St...	0
101 Dalmatians	3
101 Dalmatians (...	0
101 Dalmatians II...	2
102 Dalmatians	1
12 Dates of Chris...	1
20,000 Leagues ...	1
22 vs. Earth	1
700 Sharks	1
A Bug's Life	1

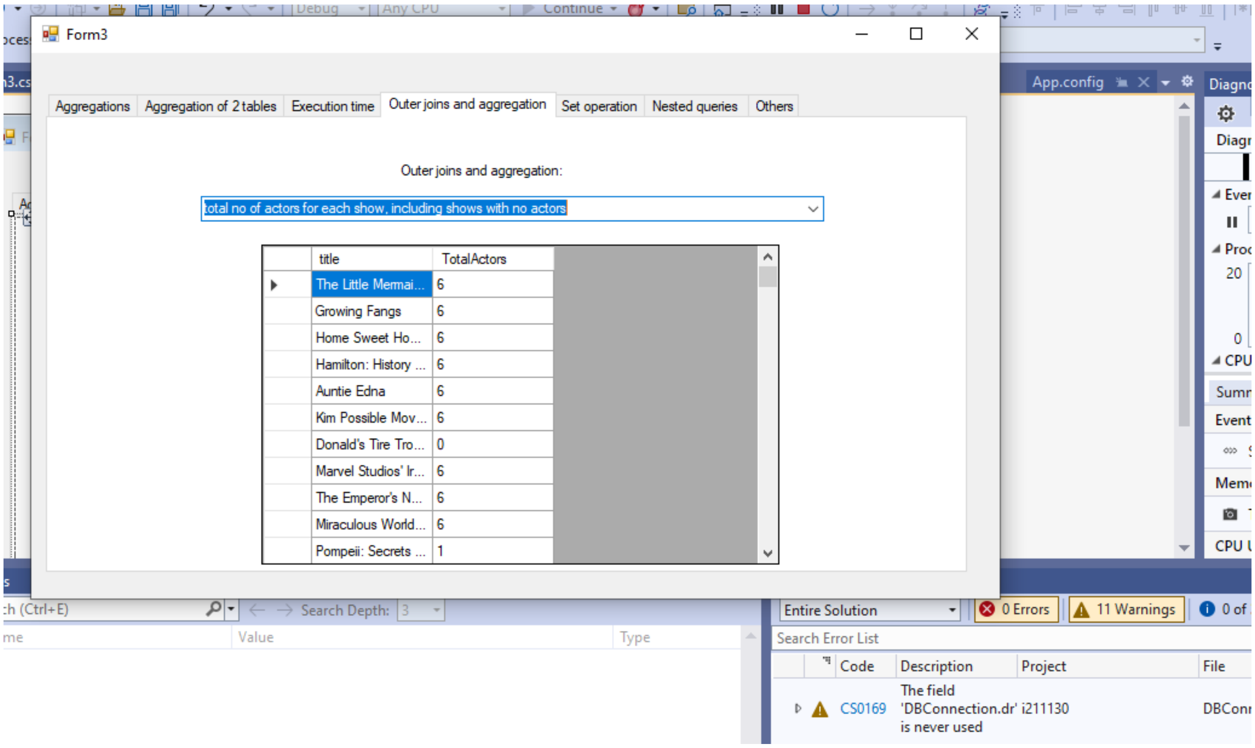
11. Count the total number of actors for each show, including shows with no actors.



```
--11.Count the total number of actors for each show, including shows with no actors.
SELECT s.title, COUNT(c.cast_id) as TotalActors
FROM Show s
LEFT JOIN Show_Cast sc ON s.show_id = sc.show_id
LEFT JOIN Cast c ON sc.cast_id = c.cast_id
GROUP BY s.title;
```

	title	TotalActors
1	The Little Mermaid II: Return to the Sea	6
2	Growing Fangs	6
3	Home Sweet Home Alone	6
4	Hamilton: History Has Its Eyes on You	6
5	Auntie Edna	6
6	Kim Possible Movie: So the Drama	6
7	Donald's Tire Trouble	0
8	Marvel Studios' Iron Man 3	6
9	The Emperor's New Groove	6
10	Miraculous World: New York, United Heroez	6

UI:



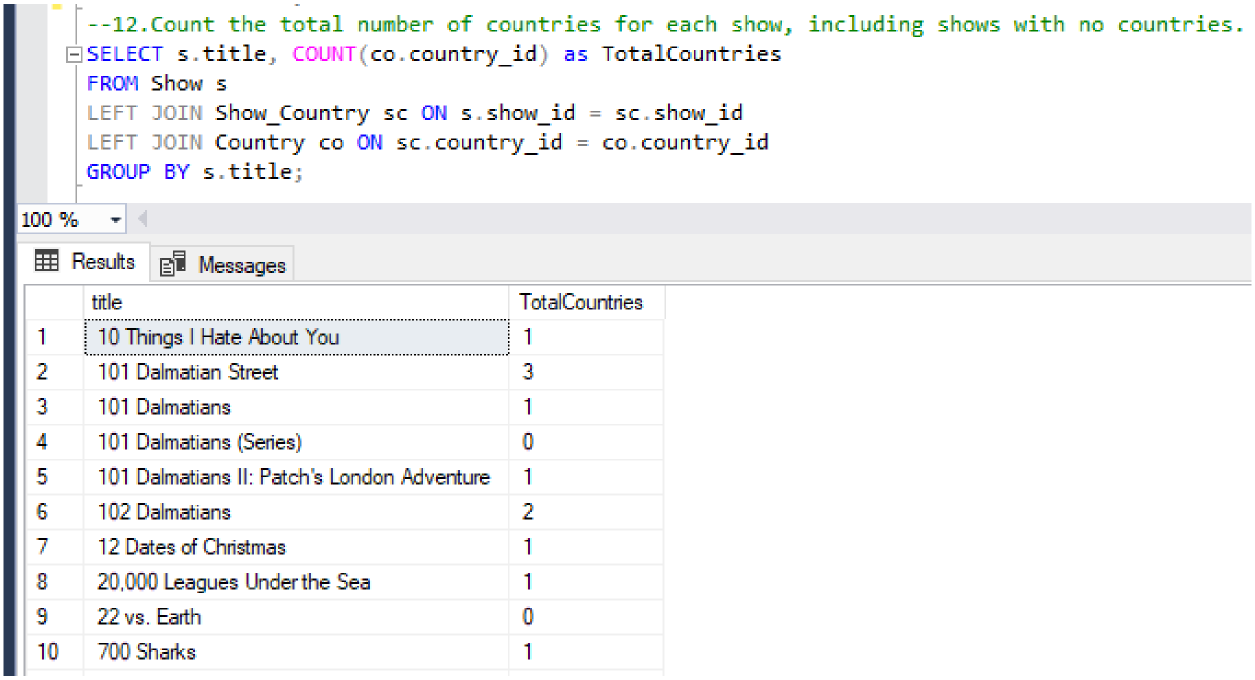
Outer joins and aggregation:

total no of actors for each show, including shows with no actors

title	TotalActors
The Little Mermai...	6
Growing Fangs	6
Home Sweet Ho...	6
Hamilton: History ...	6
Auntie Edna	6
Kim Possible Mov...	6
Donald's Tire Tro...	0
Marvel Studios' Ir...	6
The Emperor's N...	6
Miraculous World...	6
Pompeii: Secrets ...	1

Code	Description	Project	File
CS0169	The field 'DBConnection.dr' is never used	i211130	DBConn

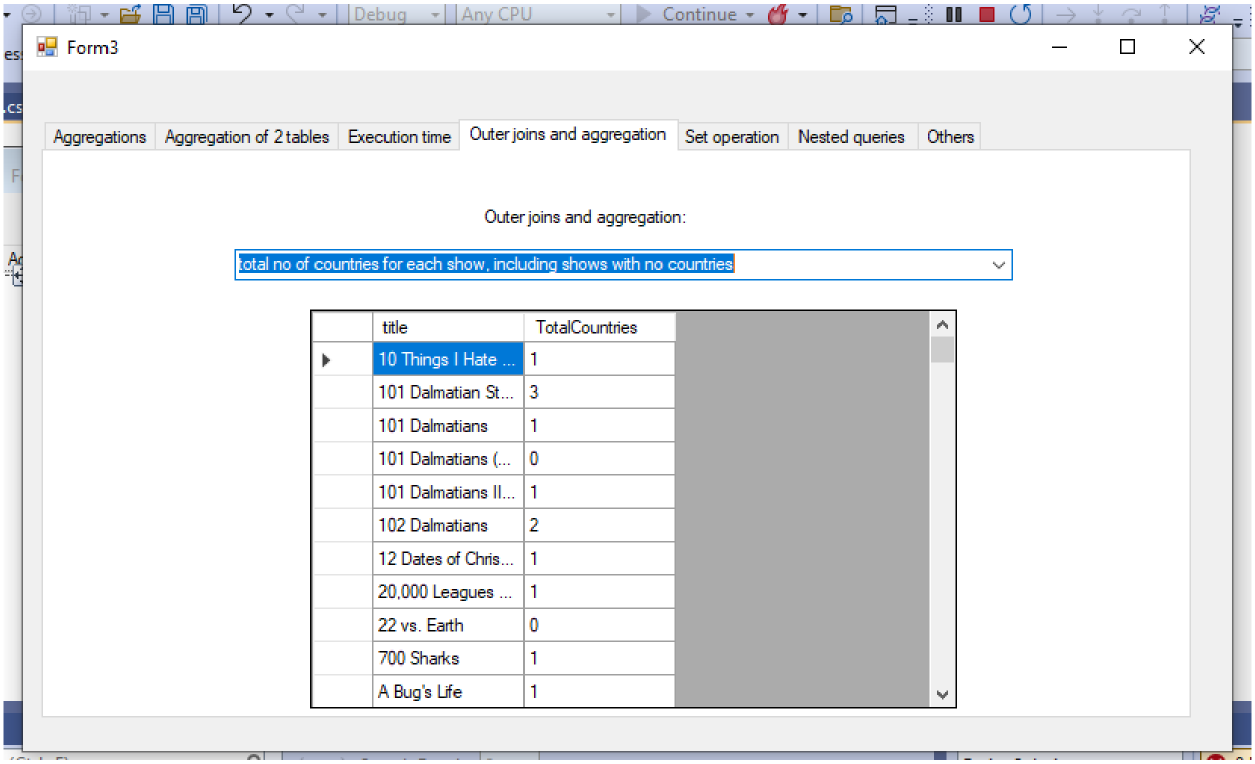
12. Count the total number of countries for each show, including shows with no countries.



```
--12.Count the total number of countries for each show, including shows with no countries.
SELECT s.title, COUNT(co.country_id) as TotalCountries
FROM Show s
LEFT JOIN Show_Country sc ON s.show_id = sc.show_id
LEFT JOIN Country co ON sc.country_id = co.country_id
GROUP BY s.title;
```

	title	TotalCountries
1	10 Things I Hate About You	1
2	101 Dalmatian Street	3
3	101 Dalmatians	1
4	101 Dalmatians (Series)	0
5	101 Dalmatians II: Patch's London Adventure	1
6	102 Dalmatians	2
7	12 Dates of Christmas	1
8	20,000 Leagues Under the Sea	1
9	22 vs. Earth	0
10	700 Sharks	1

UI:



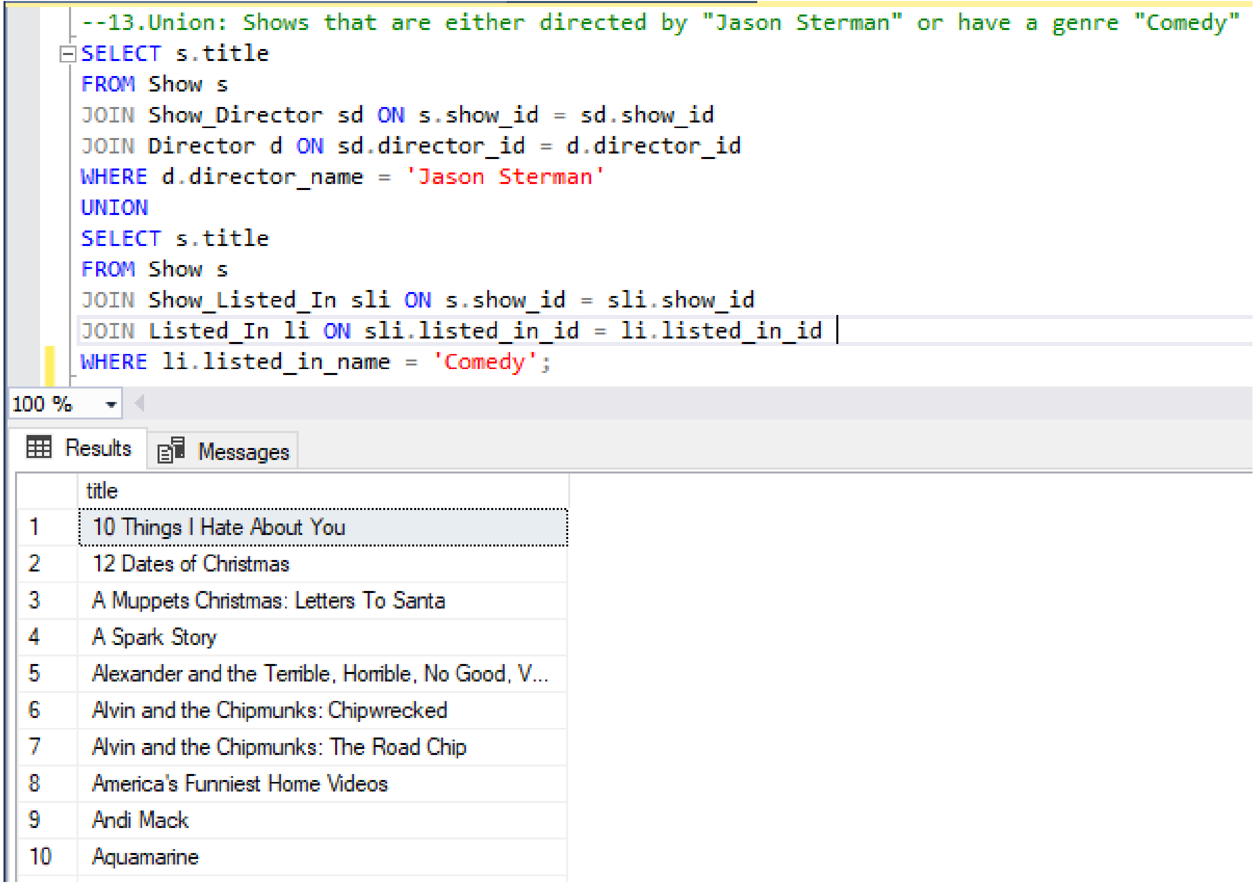
Outer joins and aggregation:

total no of countries for each show, including shows with no countries

title	TotalCountries
10 Things I Hate ...	1
101 Dalmatian St...	3
101 Dalmatians	1
101 Dalmatians (...	0
101 Dalmatians II...	1
102 Dalmatians	2
12 Dates of Chris...	1
20,000 Leagues ...	1
22 vs. Earth	0
700 Sharks	1
A Bug's Life	1

v. 3 queries involving set operators.

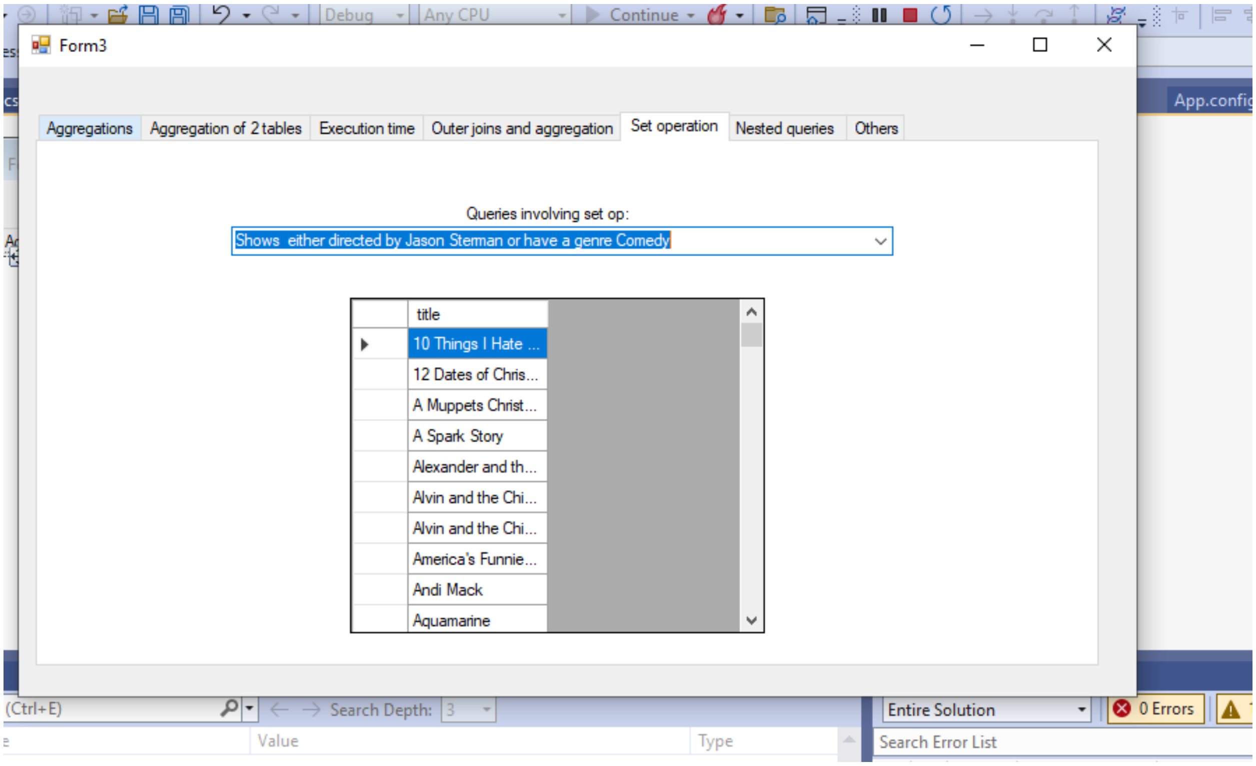
13. Union: Shows that are either directed by "Jason Sterman" or have a genre "Comedy".



```sql
--13.Union: Shows that are either directed by "Jason Sterman" or have a genre "Comedy"
SELECT s.title
FROM Show s
JOIN Show_Director sd ON s.show_id = sd.show_id
JOIN Director d ON sd.director_id = d.director_id
WHERE d.director_name = 'Jason Sterman'
UNION
SELECT s.title
FROM Show s
JOIN Show_Listed_In sli ON s.show_id = sli.show_id
JOIN Listed_In li ON sli.listed_in_id = li.listed_in_id
WHERE li.listed_in_name = 'Comedy';
```

	title
1	10 Things I Hate About You
2	12 Dates of Christmas
3	A Muppets Christmas: Letters To Santa
4	A Spark Story
5	Alexander and the Terrible, Horrible, No Good, V...
6	Alvin and the Chipmunks: Chipwrecked
7	Alvin and the Chipmunks: The Road Chip
8	America's Funniest Home Videos
9	Andi Mack
10	Aquamarine

UI:

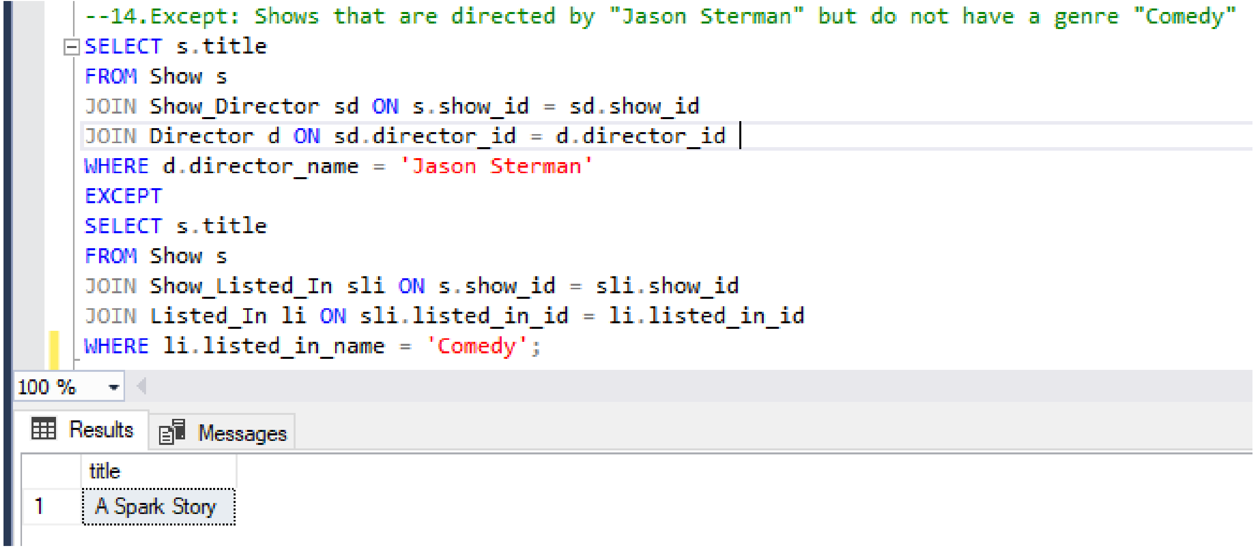


Queries involving set op:

Shows either directed by Jason Sterman or have a genre Comedy

	title
▶	10 Things I Hate ...
	12 Dates of Chris...
	A Muppets Christ...
	A Spark Story
	Alexander and th...
	Alvin and the Chi...
	Alvin and the Chi...
	America's Funnie...
	Andi Mack
	Aquamarine

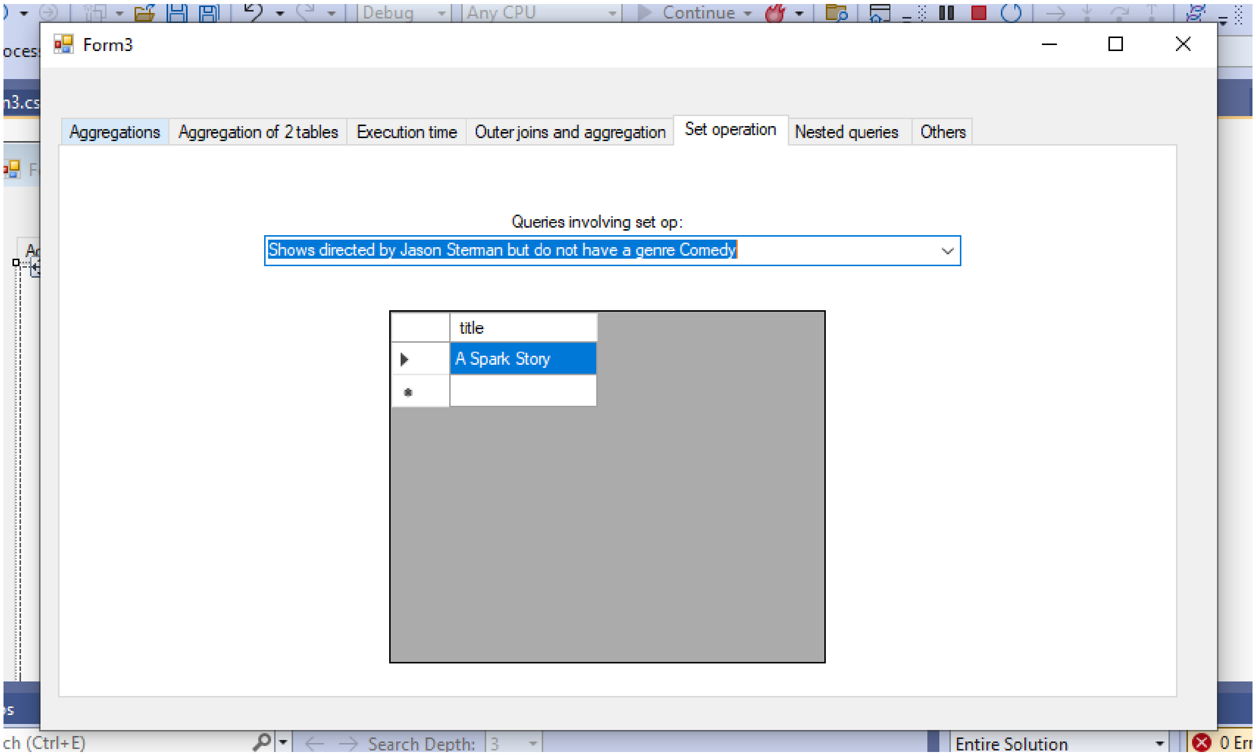
14. Except: Shows that are directed by "Jason Sterman" but do not have a genre "Comedy".



```sql
--14.Except: Shows that are directed by "Jason Sterman" but do not have a genre "Comedy"
SELECT s.title
FROM Show s
JOIN Show_Director sd ON s.show_id = sd.show_id
JOIN Director d ON sd.director_id = d.director_id
WHERE d.director_name = 'Jason Sterman'
EXCEPT
SELECT s.title
FROM Show s
JOIN Show_Listed_In sli ON s.show_id = sli.show_id
JOIN Listed_In li ON sli.listed_in_id = li.listed_in_id
WHERE li.listed_in_name = 'Comedy';
```

	title
1	A Spark Story

UI:

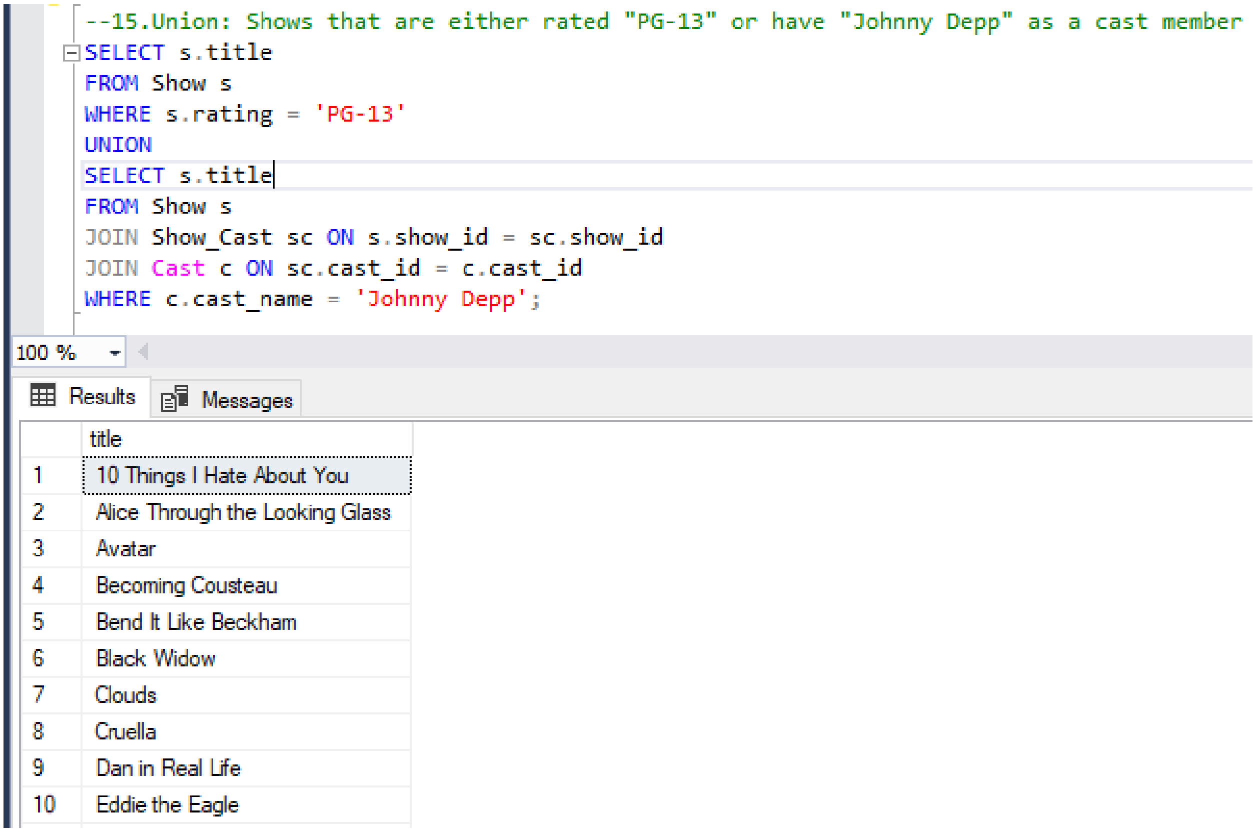


Queries involving set op:

Shows directed by Jason Sterman but do not have a genre Comedy

title
A Spark Story

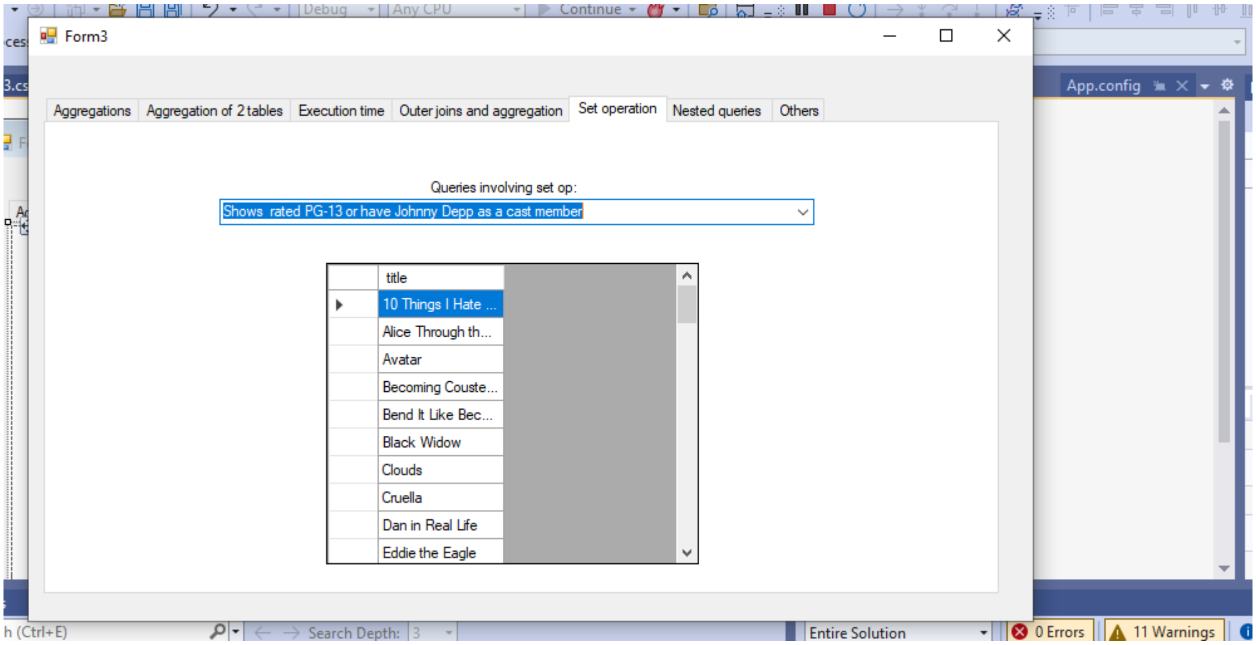
15. Union: Shows that are either rated "PG-13" or have "Johnny Depp" as a cast member.



```
--15.Union: Shows that are either rated "PG-13" or have "Johnny Depp" as a cast member
SELECT s.title
FROM Show s
WHERE s.rating = 'PG-13'
UNION
SELECT s.title
FROM Show s
JOIN Show_Cast sc ON s.show_id = sc.show_id
JOIN Cast c ON sc.cast_id = c.cast_id
WHERE c.cast_name = 'Johnny Depp';
```

	title
1	10 Things I Hate About You
2	Alice Through the Looking Glass
3	Avatar
4	Becoming Cousteau
5	Bend It Like Beckham
6	Black Widow
7	Clouds
8	Cruella
9	Dan in Real Life
10	Eddie the Eagle

UI:



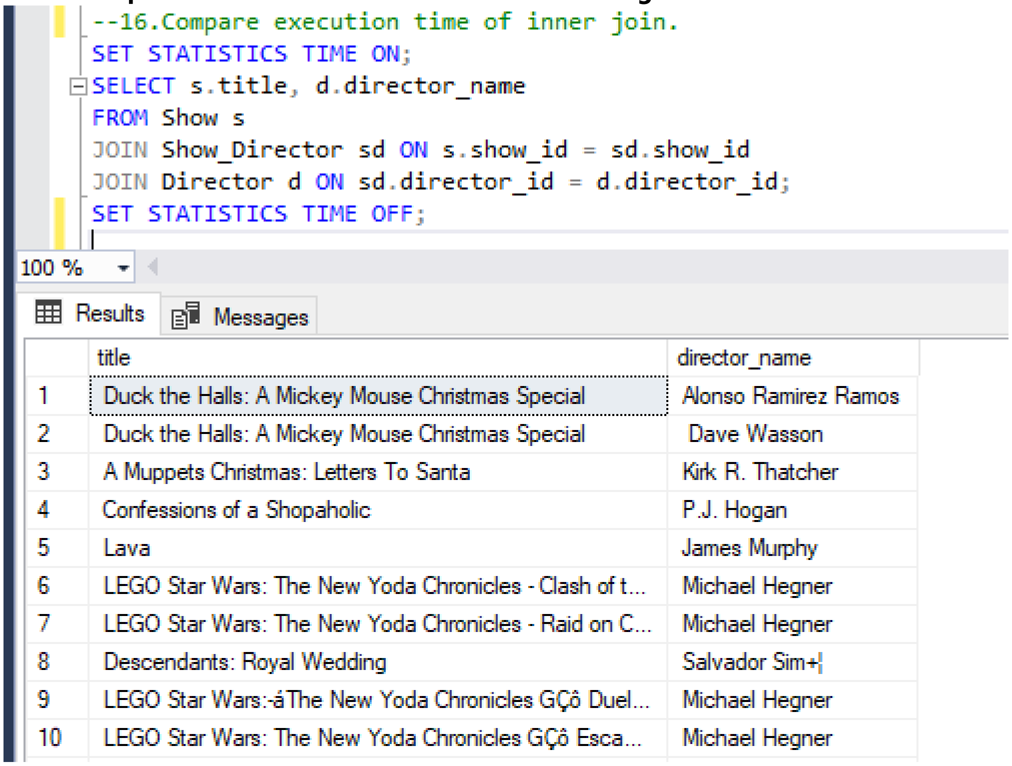
Queries involving set op:

Shows rated PG-13 or have Johnny Depp as a cast member

title
10 Things I Hate ...
Alice Through th...
Avatar
Becoming Couste...
Bend It Like Bec...
Black Widow
Clouds
Cruella
Dan in Real Life
Eddie the Eagle

vi. 3 queries to compare the difference in execution time of different types of joins like outer join vs equi join etc.

16. Compare execution time of inner join.



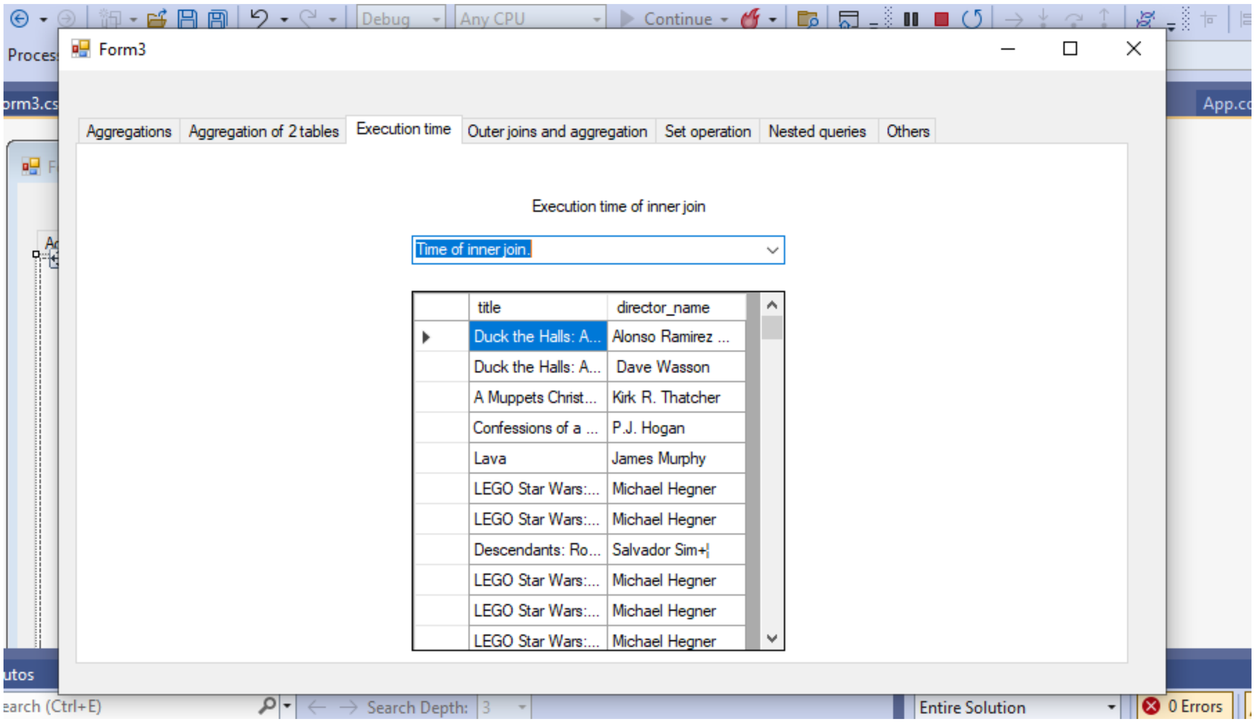
```
--16.Compare execution time of inner join.
SET STATISTICS TIME ON;
SELECT s.title, d.director_name
FROM Show s
JOIN Show_Director sd ON s.show_id = sd.show_id
JOIN Director d ON sd.director_id = d.director_id;
SET STATISTICS TIME OFF;
```

	title	director_name
1	Duck the Halls: A Mickey Mouse Christmas Special	Alonso Ramirez Ramos
2	Duck the Halls: A Mickey Mouse Christmas Special	Dave Wasson
3	A Muppets Christmas: Letters To Santa	Kirk R. Thatcher
4	Confessions of a Shopaholic	P.J. Hogan
5	Lava	James Murphy
6	LEGO Star Wars: The New Yoda Chronicles - Clash of t...	Michael Hegner
7	LEGO Star Wars: The New Yoda Chronicles - Raid on C...	Michael Hegner
8	Descendants: Royal Wedding	Salvador Sim+¦
9	LEGO Star Wars:-áThe New Yoda Chronicles GÇô Duel...	Michael Hegner
10	LEGO Star Wars: The New Yoda Chronicles GÇô Esca...	Michael Hegner

EXECUTION TIME:

```
SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 0 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

(1121 rows affected)

 SQL Server Execution Times:
   CPU time = 31 ms,  elapsed time = 83 ms.

Completion time: 2023-05-14T21:52:42.5930866+05:00
```

UI:

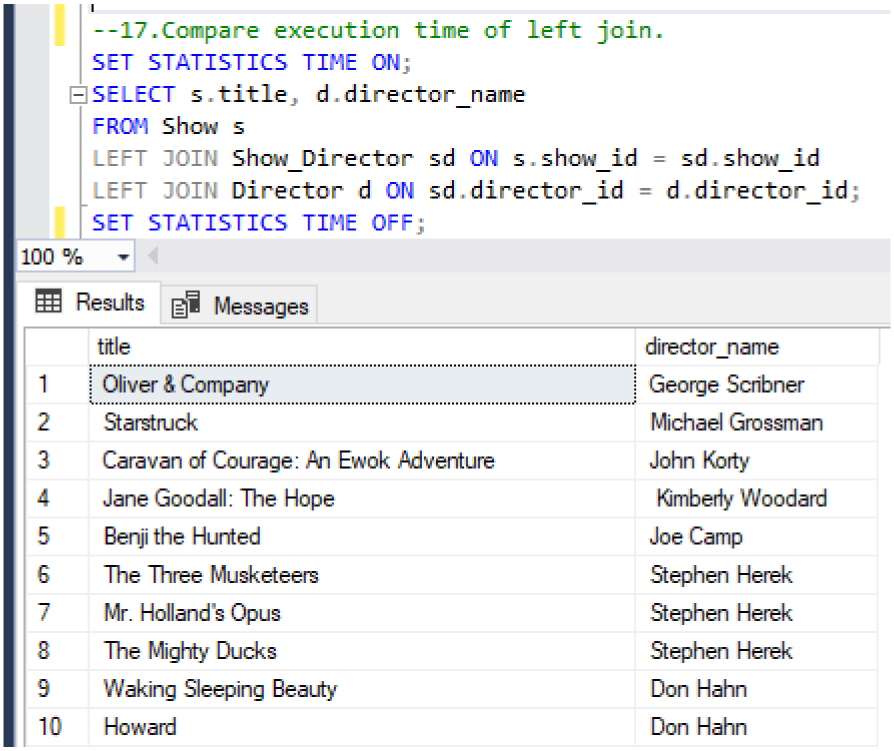


Execution time of inner join

Time of inner join.

title	director_name
Duck the Halls: A...	Alonso Ramirez ...
Duck the Halls: A...	Dave Wasson
A Muppets Christ...	Kirk R. Thatcher
Confessions of a ...	P.J. Hogan
Lava	James Murphy
LEGO Star Wars:...	Michael Hegner
LEGO Star Wars:...	Michael Hegner
Descendants: Ro...	Salvador Sim+¦
LEGO Star Wars:...	Michael Hegner
LEGO Star Wars:...	Michael Hegner
LEGO Star Wars:...	Michael Hegner

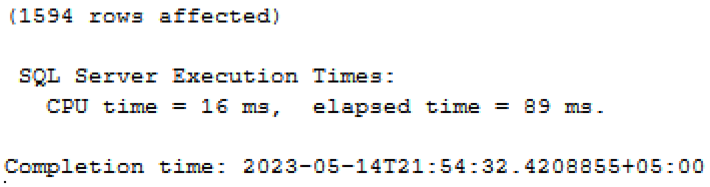
17. Compare execution time of left join.



```
--17.Compare execution time of left join.
SET STATISTICS TIME ON;
SELECT s.title, d.director_name
FROM Show s
LEFT JOIN Show_Director sd ON s.show_id = sd.show_id
LEFT JOIN Director d ON sd.director_id = d.director_id;
SET STATISTICS TIME OFF;
```

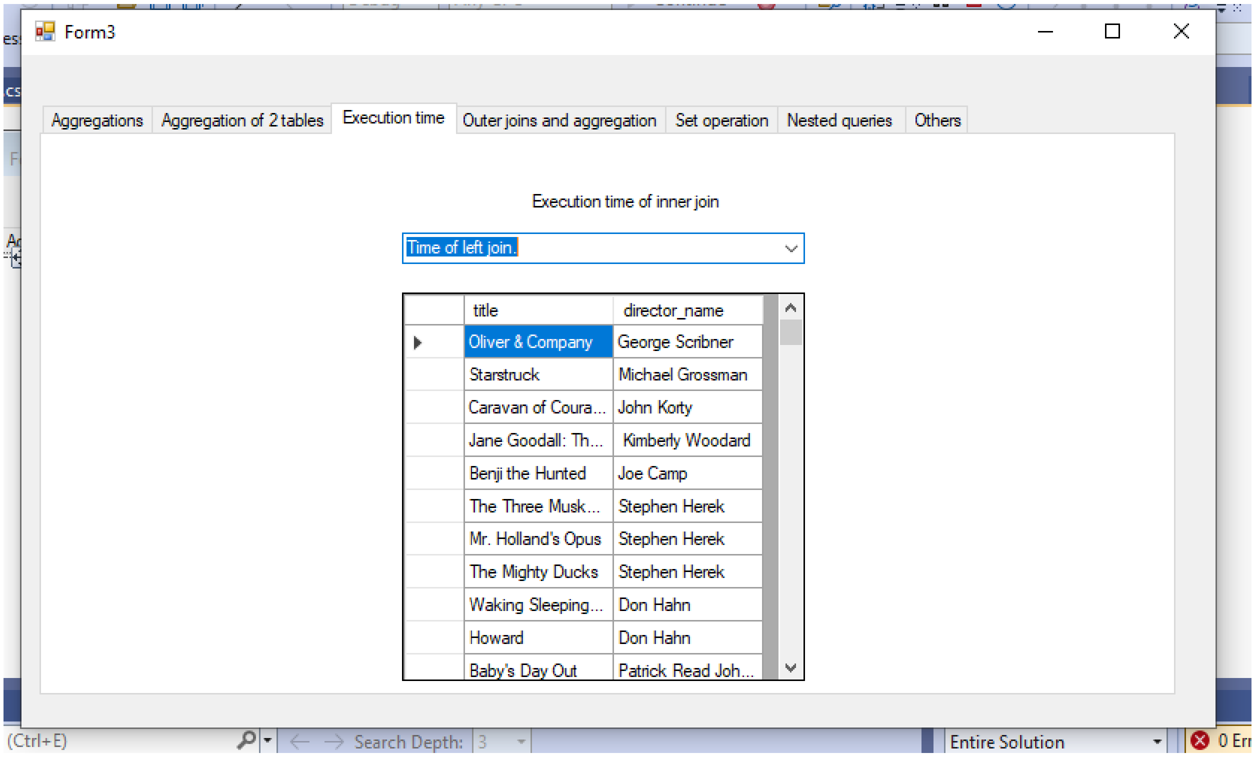
	title	director_name
1	Oliver & Company	George Scribner
2	Starstruck	Michael Grossman
3	Caravan of Courage: An Ewok Adventure	John Korty
4	Jane Goodall: The Hope	Kimberly Woodard
5	Benji the Hunted	Joe Camp
6	The Three Musketeers	Stephen Herek
7	Mr. Holland's Opus	Stephen Herek
8	The Mighty Ducks	Stephen Herek
9	Waking Sleeping Beauty	Don Hahn
10	Howard	Don Hahn

EXECUTION TIME:



```
(1594 rows affected)

 SQL Server Execution Times:
   CPU time = 16 ms,  elapsed time = 89 ms.

Completion time: 2023-05-14T21:54:32.4208855+05:00
```

UI:

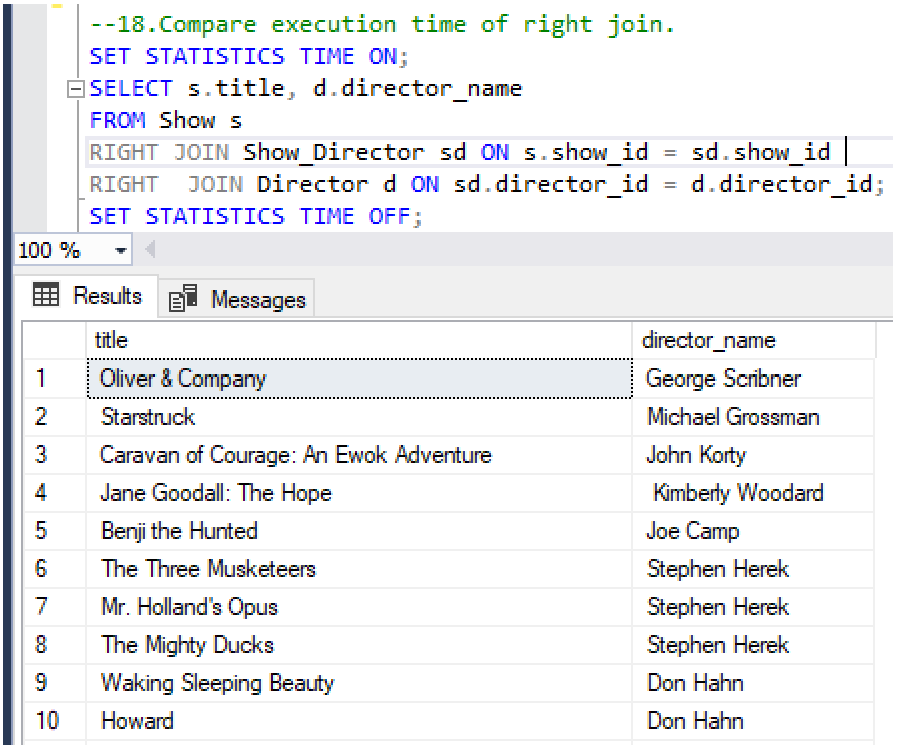


Execution time of inner join

Time of left join.

	title	director_name
▶	Oliver & Company	George Scribner
	Starstruck	Michael Grossman
	Caravan of Coura...	John Korty
	Jane Goodall: Th...	Kimberly Woodard
	Benji the Hunted	Joe Camp
	The Three Musk...	Stephen Herek
	Mr. Holland's Opus	Stephen Herek
	The Mighty Ducks	Stephen Herek
	Waking Sleeping...	Don Hahn
	Howard	Don Hahn
	Baby's Day Out	Patrick Read Joh...

18. Compare execution time of right join.



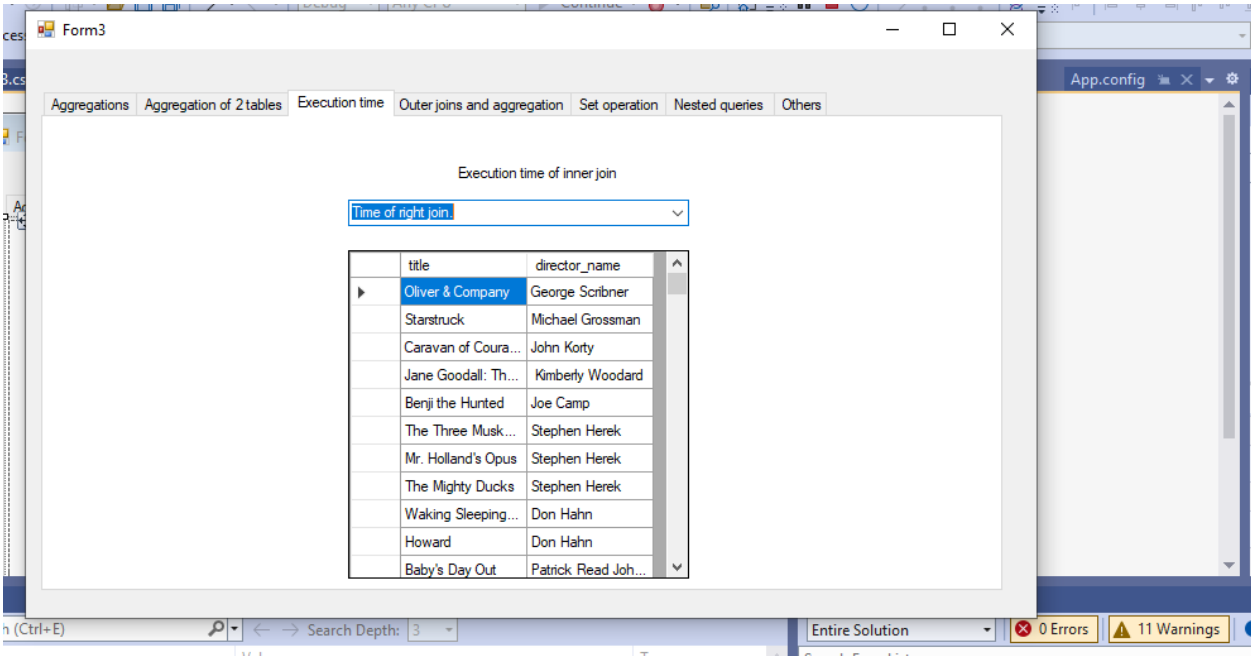
```
--18.Compare execution time of right join.
SET STATISTICS TIME ON;
SELECT s.title, d.director_name
FROM Show s
RIGHT JOIN Show_Director sd ON s.show_id = sd.show_id
RIGHT  JOIN Director d ON sd.director_id = d.director_id;
SET STATISTICS TIME OFF;
```

	title	director_name
1	Oliver & Company	George Scribner
2	Starstruck	Michael Grossman
3	Caravan of Courage: An Ewok Adventure	John Korty
4	Jane Goodall: The Hope	Kimberly Woodard
5	Benji the Hunted	Joe Camp
6	The Three Musketeers	Stephen Herek
7	Mr. Holland's Opus	Stephen Herek
8	The Mighty Ducks	Stephen Herek
9	Waking Sleeping Beauty	Don Hahn
10	Howard	Don Hahn

EXECUTION TIME:

```
(1121 rows affected)

 SQL Server Execution Times:
   CPU time = 16 ms,  elapsed time = 77 ms.

Completion time: 2023-05-14T21:55:39.5950680+05:00
```

UI:



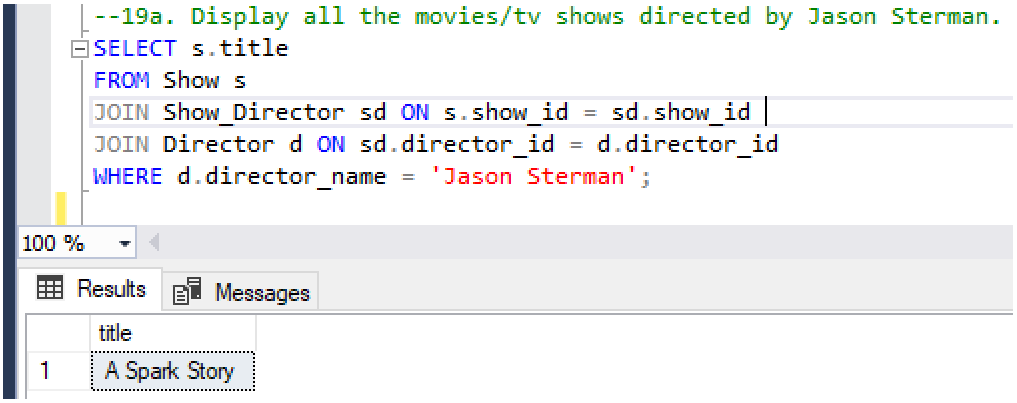
Execution time of inner join

Time of right join.

title	director_name
Oliver & Company	George Scribner
Starstruck	Michael Grossman
Caravan of Coura...	John Korty
Jane Goodall: Th...	Kimberly Woodard
Benji the Hunted	Joe Camp
The Three Musk...	Stephen Herek
Mr. Holland's Opus	Stephen Herek
The Mighty Ducks	Stephen Herek
Waking Sleeping...	Don Hahn
Howard	Don Hahn
Baby's Day Out	Patrick Read Joh...

vii. Other queries

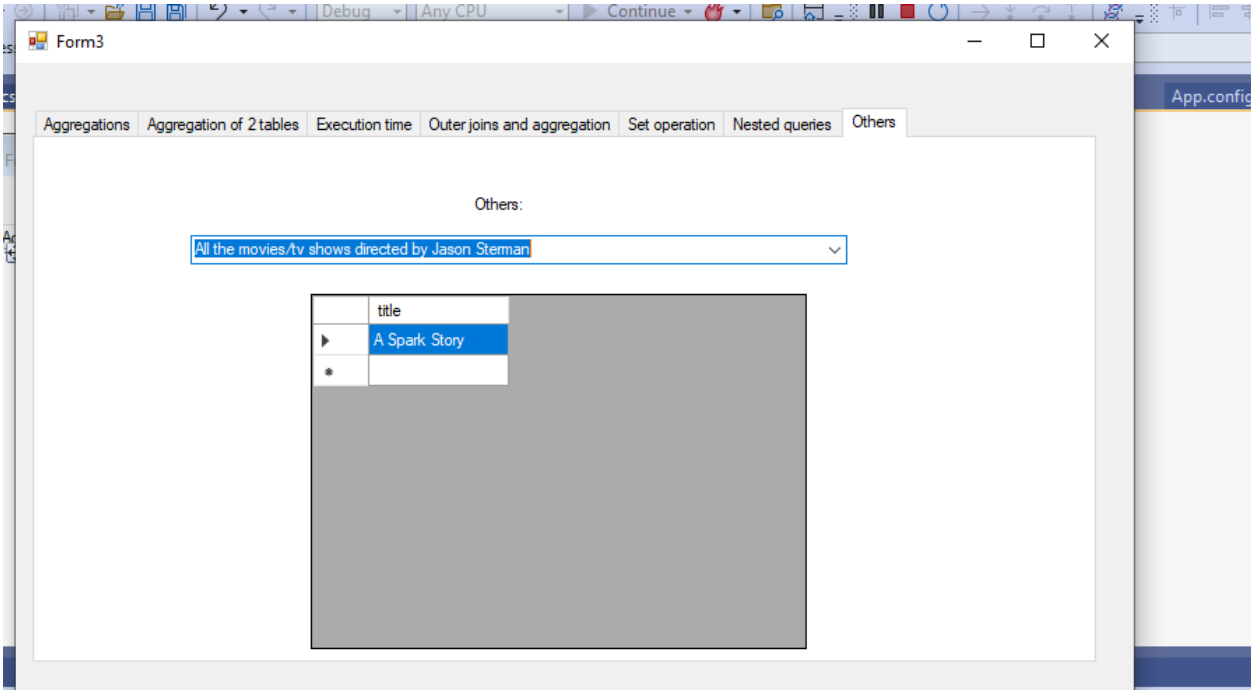
19a. Display all the movies/TV shows directed by Jason Sterman.



```sql
--19a. Display all the movies/tv shows directed by Jason Sterman.
SELECT s.title
FROM Show s
JOIN Show_Director sd ON s.show_id = sd.show_id
JOIN Director d ON sd.director_id = d.director_id
WHERE d.director_name = 'Jason Sterman';
```

	title
1	A Spark Story

UI:

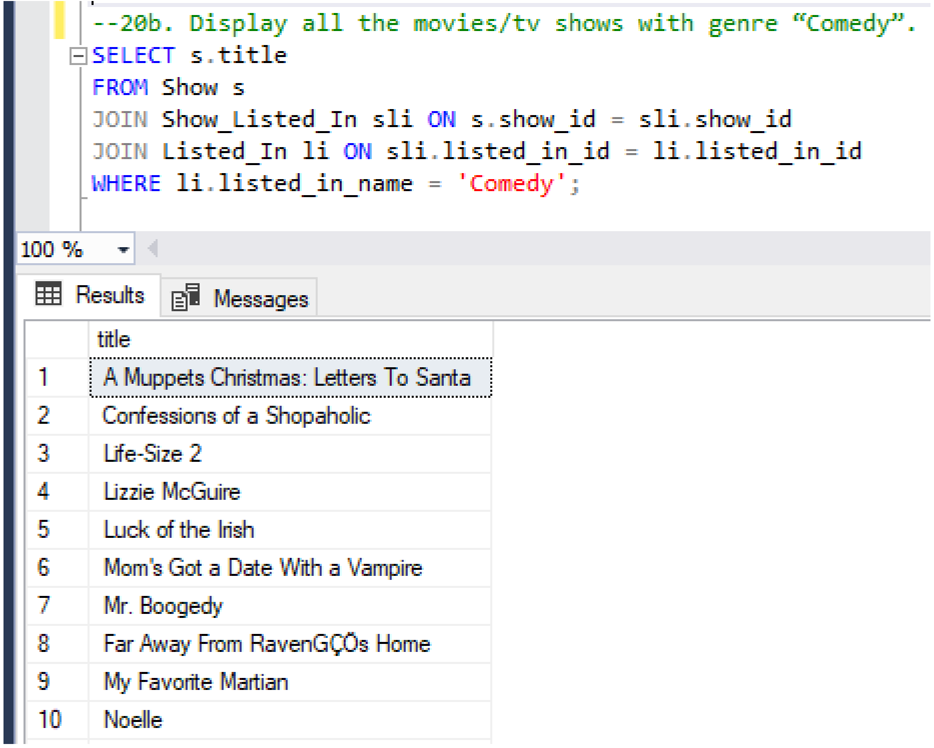


Others:

All the movies/tv shows directed by Jason Sterman

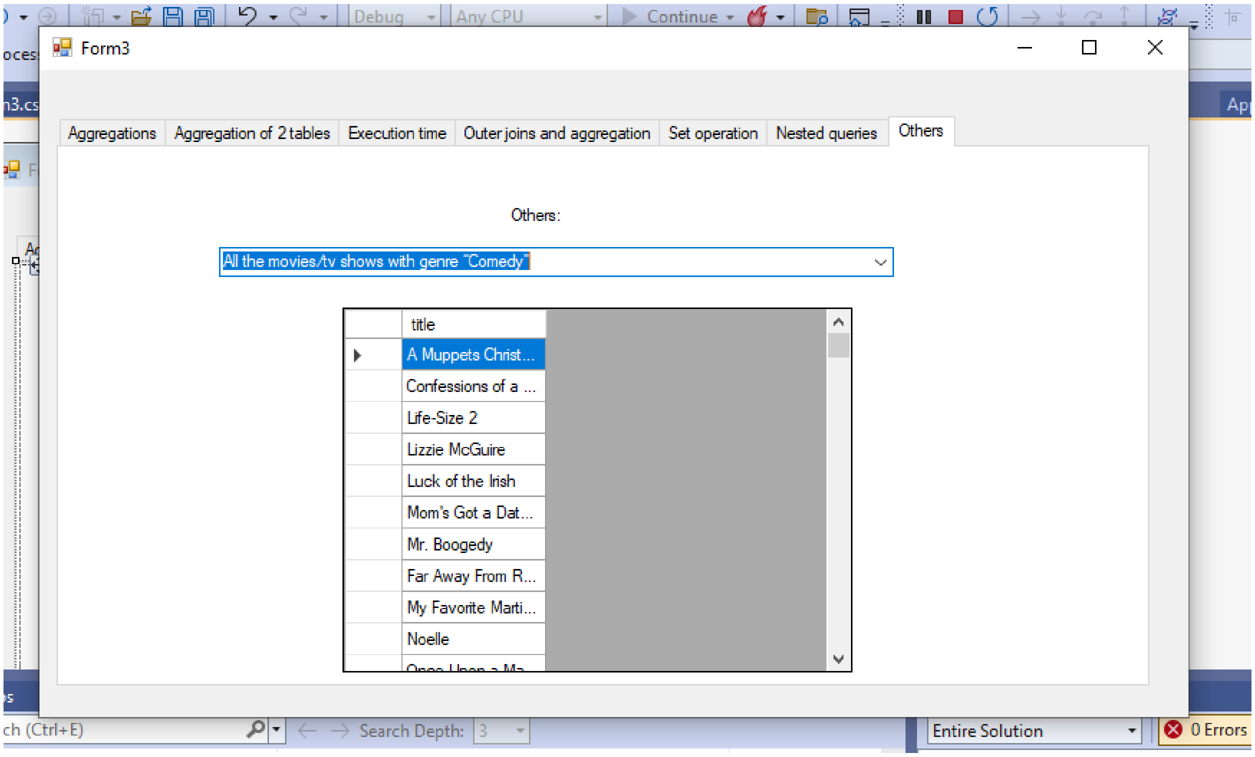
	title
▶	A Spark Story
*	

20b. Display all the movies/TV shows with genre “Comedy”.



```
--20b. Display all the movies/tv shows with genre “Comedy”.
SELECT s.title
FROM Show s
JOIN Show_Listed_In sli ON s.show_id = sli.show_id
JOIN Listed_In li ON sli.listed_in_id = li.listed_in_id
WHERE li.listed_in_name = 'Comedy';
```

	title
1	A Muppets Christmas: Letters To Santa
2	Confessions of a Shopaholic
3	Life-Size 2
4	Lizzie McGuire
5	Luck of the Irish
6	Mom's Got a Date With a Vampire
7	Mr. Boogedy
8	Far Away From RavenGÇÖs Home
9	My Favorite Martian
10	Noelle

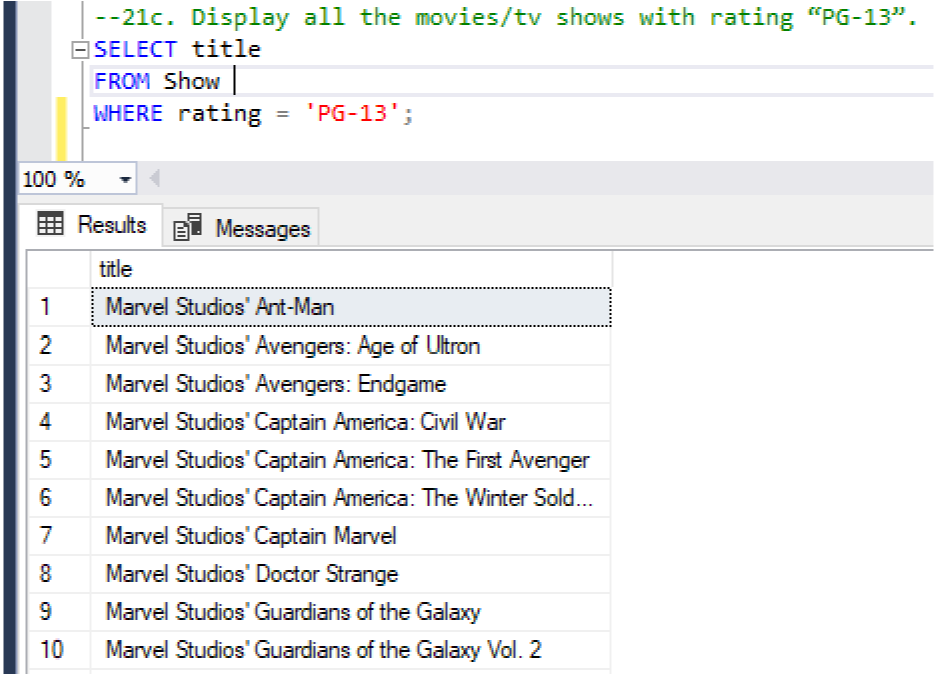


Others:

All the movies/tv shows with genre "Comedy"

title
A Muppets Christ...
Confessions of a ...
Life-Size 2
Lizzie McGuire
Luck of the Irish
Mom's Got a Dat...
Mr. Boogedy
Far Away From R...
My Favorite Marti...
Noelle
Once Upon a Ma...

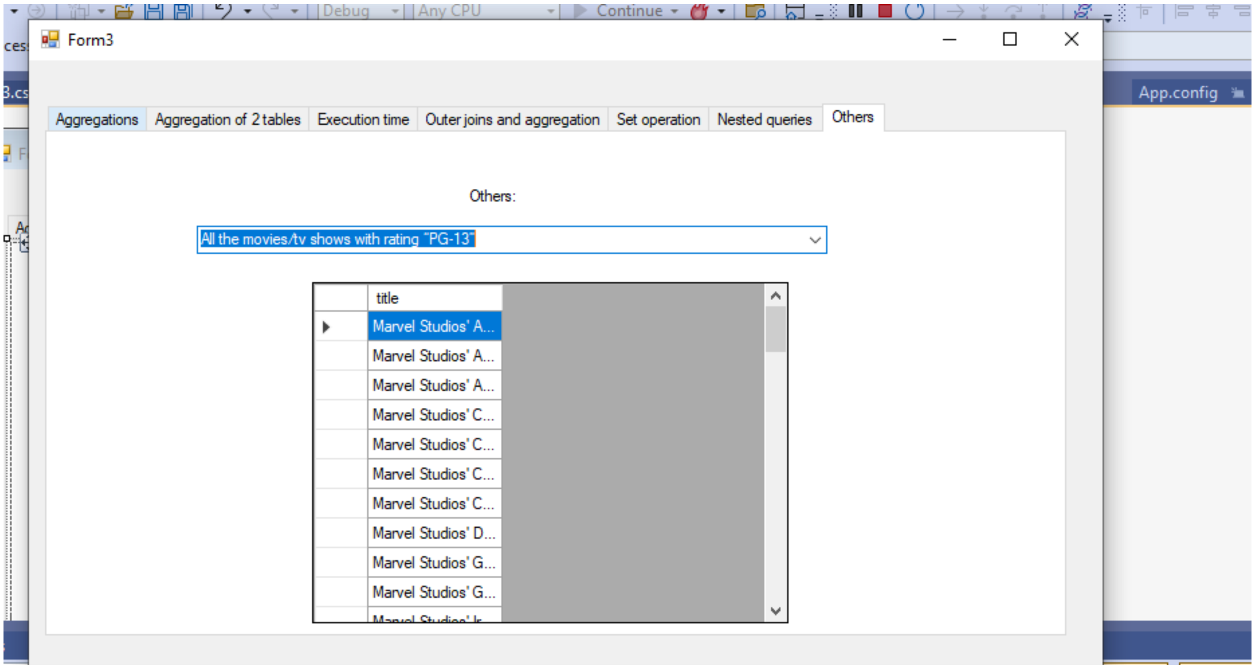
21c. Display all the movies/TV shows with rating “PG-13”.



```sql
--21c. Display all the movies/tv shows with rating “PG-13”.
SELECT title
FROM Show
WHERE rating = 'PG-13';
```

	title
1	Marvel Studios' Ant-Man
2	Marvel Studios' Avengers: Age of Ultron
3	Marvel Studios' Avengers: Endgame
4	Marvel Studios' Captain America: Civil War
5	Marvel Studios' Captain America: The First Avenger
6	Marvel Studios' Captain America: The Winter Sold...
7	Marvel Studios' Captain Marvel
8	Marvel Studios' Doctor Strange
9	Marvel Studios' Guardians of the Galaxy
10	Marvel Studios' Guardians of the Galaxy Vol. 2

UI:

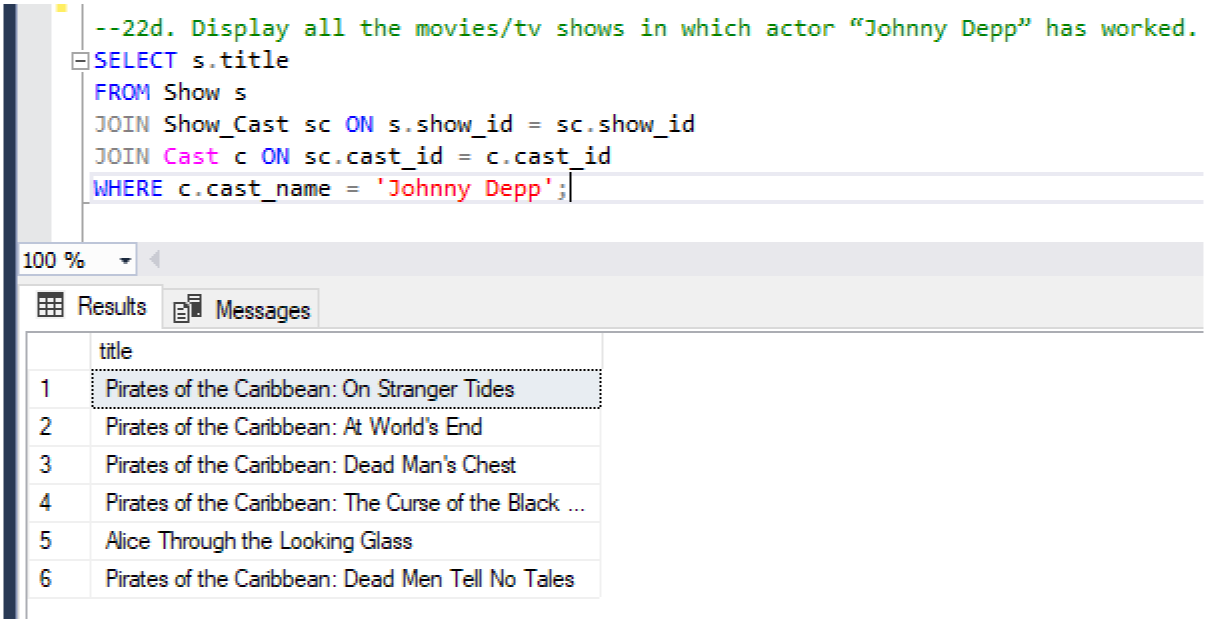


Others:

All the movies/tv shows with rating “PG-13”

title
Marvel Studios' A...
Marvel Studios' A...
Marvel Studios' A...
Marvel Studios' C...
Marvel Studios' C...
Marvel Studios' C...
Marvel Studios' C...
Marvel Studios' D...
Marvel Studios' G...
Marvel Studios' G...
Marvel Studios' Ir...

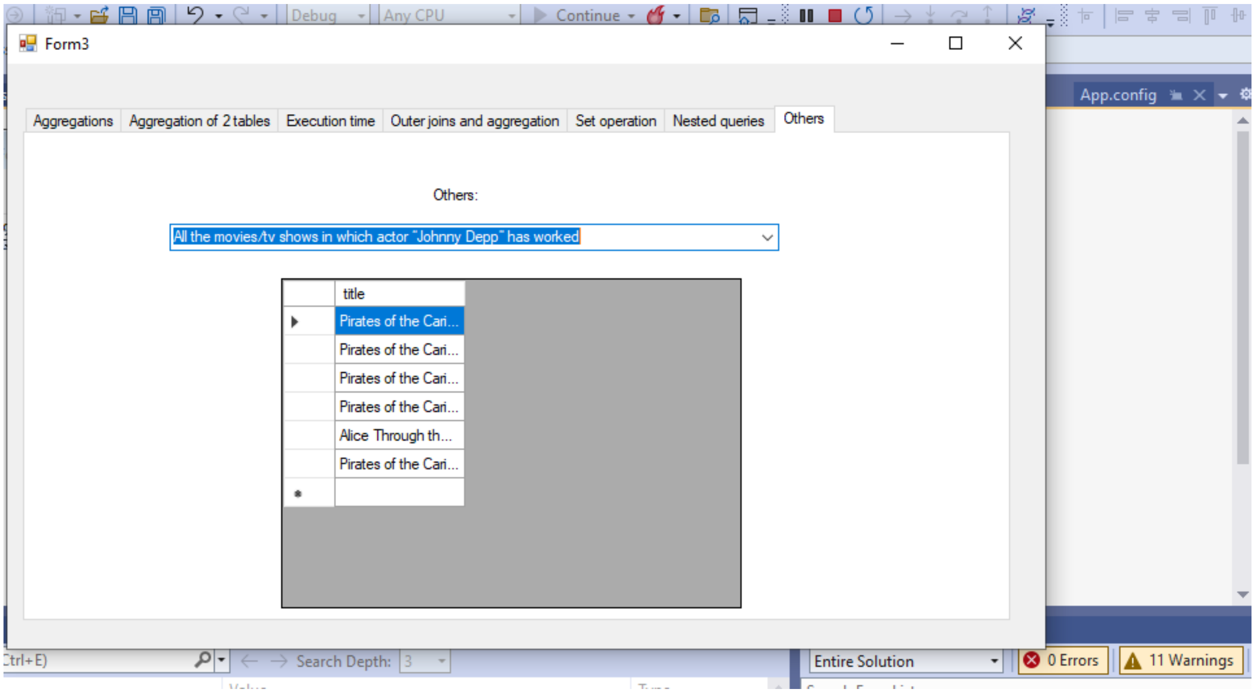
22d. Display all the movies/TV shows in which actor “Johnny Depp” has worked.



```sql
--22d. Display all the movies/tv shows in which actor “Johnny Depp” has worked.
SELECT s.title
FROM Show s
JOIN Show_Cast sc ON s.show_id = sc.show_id
JOIN Cast c ON sc.cast_id = c.cast_id
WHERE c.cast_name = 'Johnny Depp';
```

	title
1	Pirates of the Caribbean: On Stranger Tides
2	Pirates of the Caribbean: At World's End
3	Pirates of the Caribbean: Dead Man's Chest
4	Pirates of the Caribbean: The Curse of the Black ...
5	Alice Through the Looking Glass
6	Pirates of the Caribbean: Dead Men Tell No Tales

UI:



Others:

All the movies/tv shows in which actor "Johnny Depp" has worked

	title
▶	Pirates of the Cari...
	Pirates of the Cari...
	Pirates of the Cari...
	Pirates of the Cari...
	Alice Through th...
	Pirates of the Cari...
*	