EE4202 - Database Systems

Mini Project – Semester 4: 09 April 2024

GROUP NUMBER – 87

Table of Contents

1. Chapter 1 - Requirement Analysis

i. Functional Requirements

i. **Manage Movies and TV Series**
The system should allow users to add, update, and delete information about movies and TV series.

ii. **Track Director Information**
Users should be able to associate directors with movies and TV series and track their details such as name, age, nationality, and awards.

iii. **Studio and Website Association**
The system should facilitate linking studios and websites with TV series for streaming.

iv. **Episode Management**
Users should be able to manage episodes of TV series, including their titles, ratings, length, and cost per episode.

v. **Actor Details**
The system should store information about main actors, including their names, nationality, and awards.

vi. **Song Information**
Users should be able to track songs featured in movies, including details about singers, writers, and awards.

vii. **Review Management**
The system should allow users to add reviews for both movies and TV series, including comments, ratings, and review dates.

ii. Data Requirements

i. **Movie Data**
Information about movies such as ID, name, release year, genre, duration, language, and budget.

ii. **Director Data**
Details about directors including license number, name, age, active period, country, nationality, awards, and associations with movies/TV series.

iii. **TV Series Data**
Data related to TV series such as ID, name, release year, genre, language, number of seasons, and country of origin.

iv. **Country Data**
Information about countries including name, rank, and region.

v. **Episode Data**
Details about episodes including title, ratings, length, cost per episode, and association with TV series.

vi. **Studio Data**
Information about studios such as license number, established date, revenue, country, and association with TV series.

vii. **Website Data**
Data related to websites for streaming including name, license number, and association with TV series.

viii. **Main Actor Data**
Information about main actors including license number, name, nationality, date of birth, active period, awards, and associations with movies.

ix. **Song Data**
Details about songs featured in movies including license ID, name, singer, writer, language, awards, and associations with movies.

x. **Review Data**
Data related to reviews including comments, ratings, review date, user experience, and associations with movies/TV series.

2. Chapter 2 - Conceptual Design

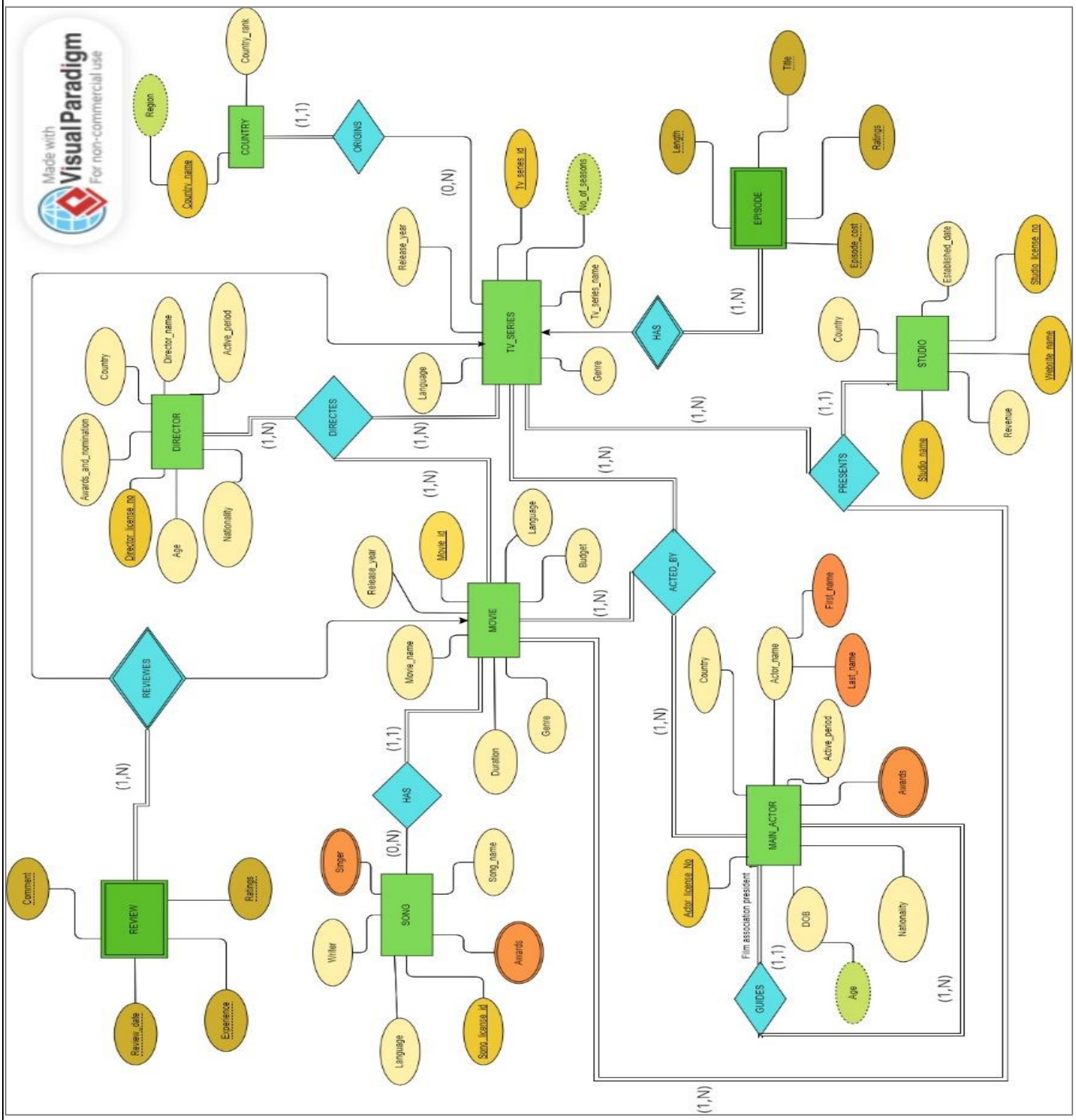


FIGURE 1 : ER DIAGRAM

1). **Recursive relationship**

- GUIDES

2). **Multivalued attribute**

- Awards in MAIN_ACTOR relation
- Awards in SONG relation
- Singer in SONG relation

3). **Composite attribute**

- Actor_name in MAIN_ACTOR relation

4). **Derived attribute**

- Age in the MAIN_ACTOR

5). **Multiple candidate key**

- In the relation STUDIO

6). Strong entities

- DIRECTOR
- COUNTRY
- TV_SERIES
- MOVIE
- SONG
- STUDIO

7). Derived attribute

- Age in the MAIN_ACTOR

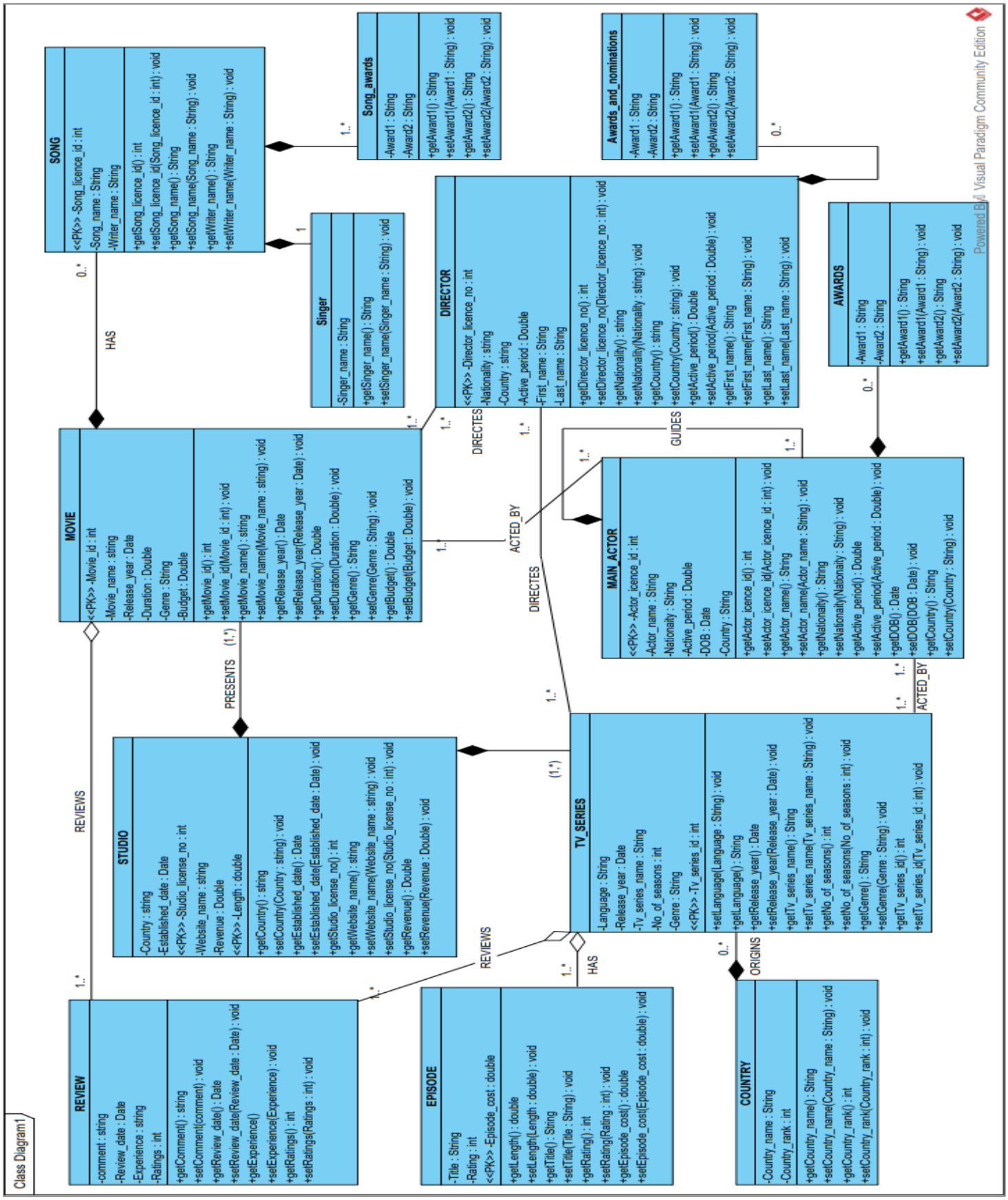


FIGURE 2 : UML DIAGRAM

3. Chapter 3 – Implementation

3.1 Creation of the schema

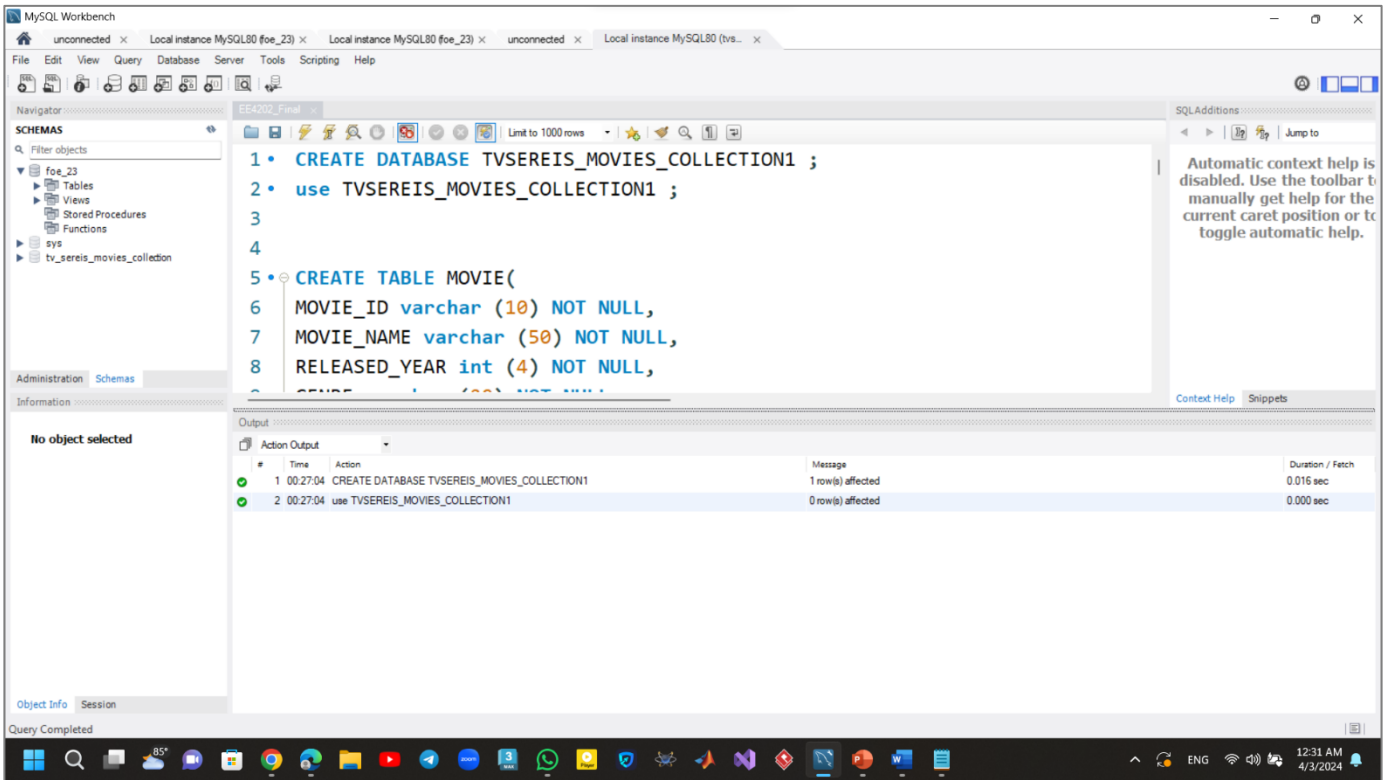


FIGURE 3 : CREATION OF SCHEMA

3.2 Table Definitions

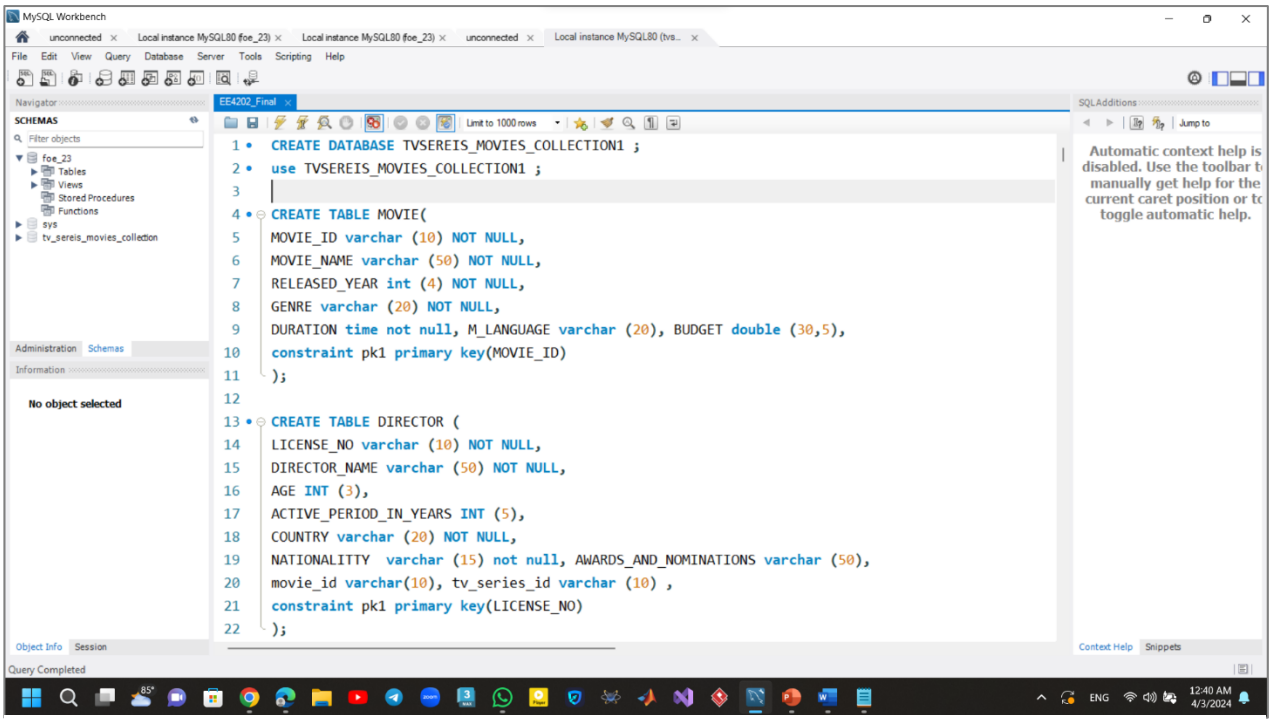


FIGURE 4 : CREATING MOVIE AND DIRECTOR TABLES

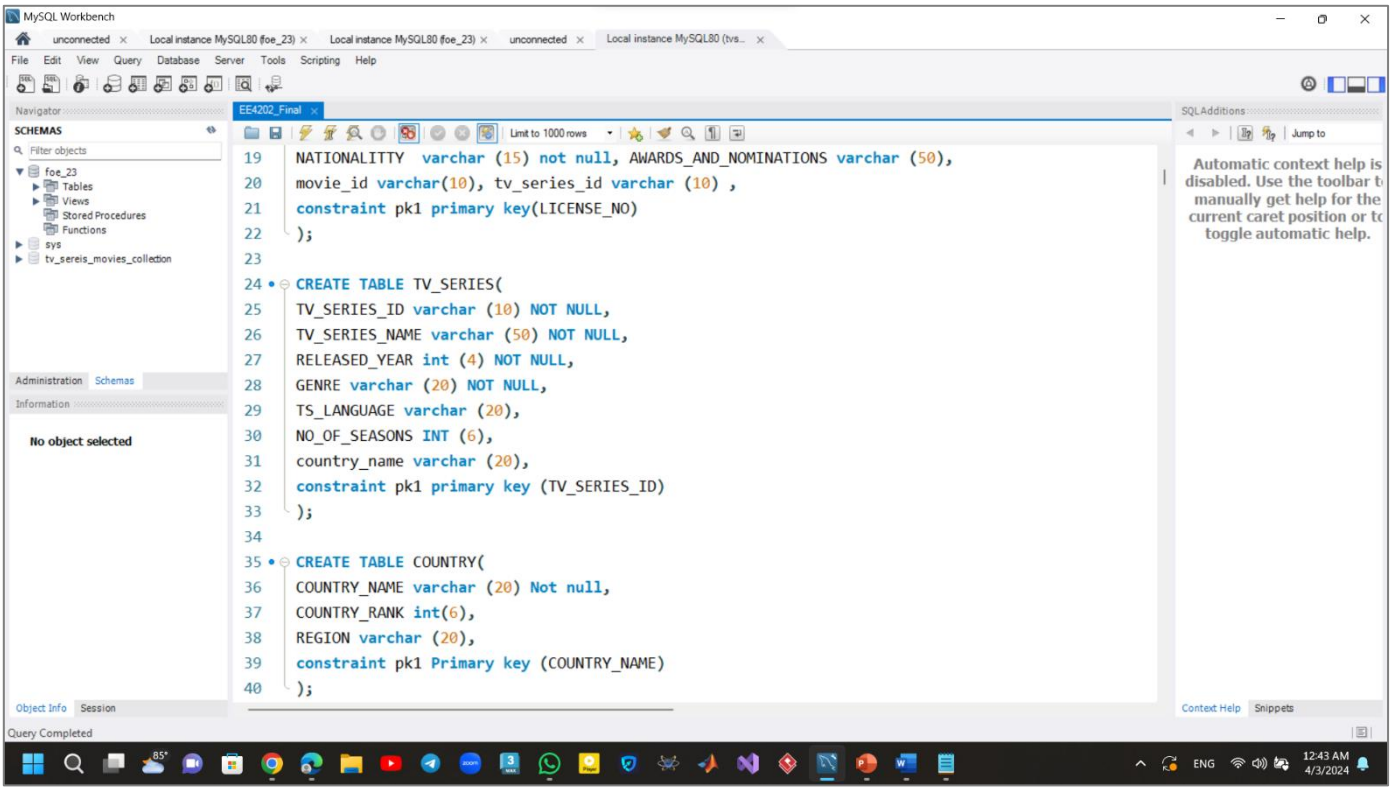


FIGURE 6 : CREATING TV SERIES AND COUNTRY TABLES

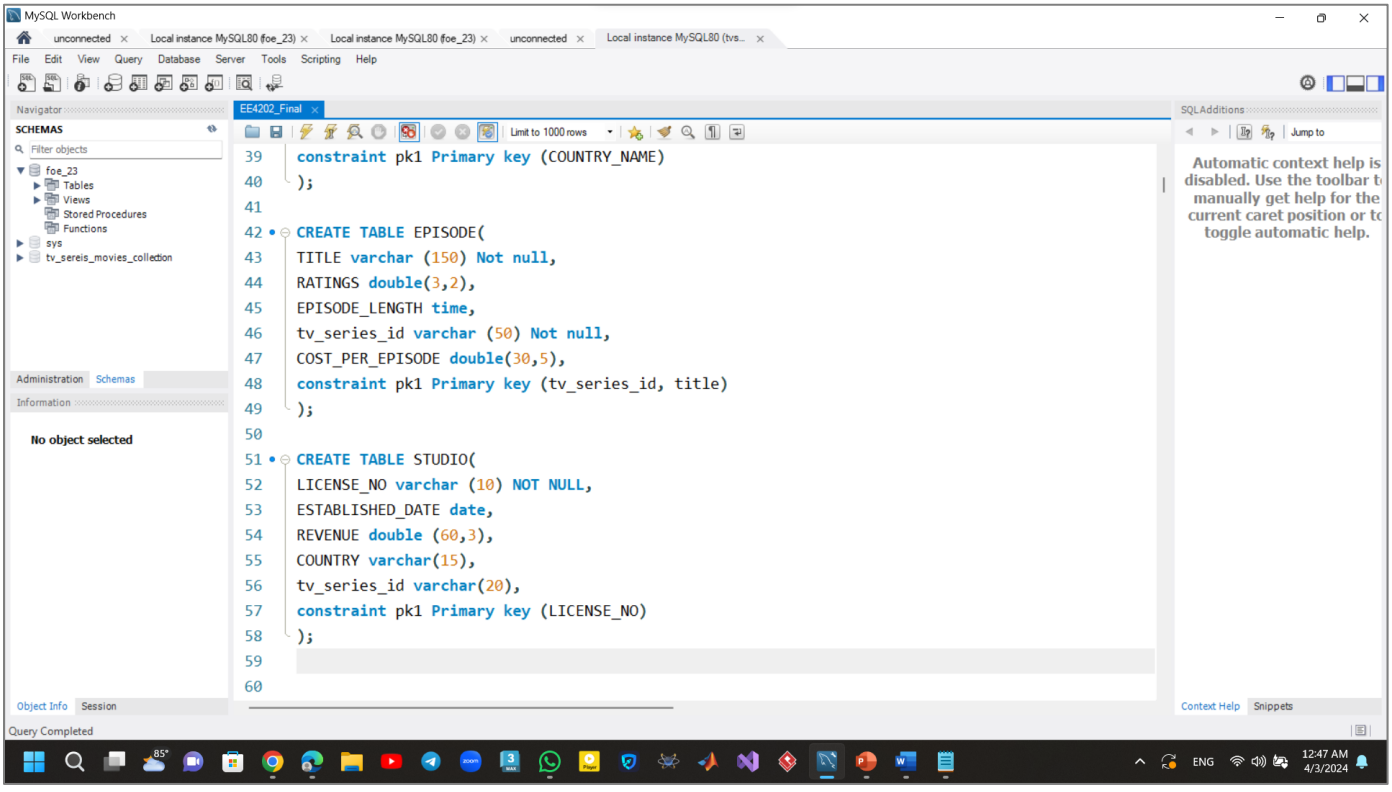


FIGURE 5 : CREATING EPISODE AND STUDIO TABLES

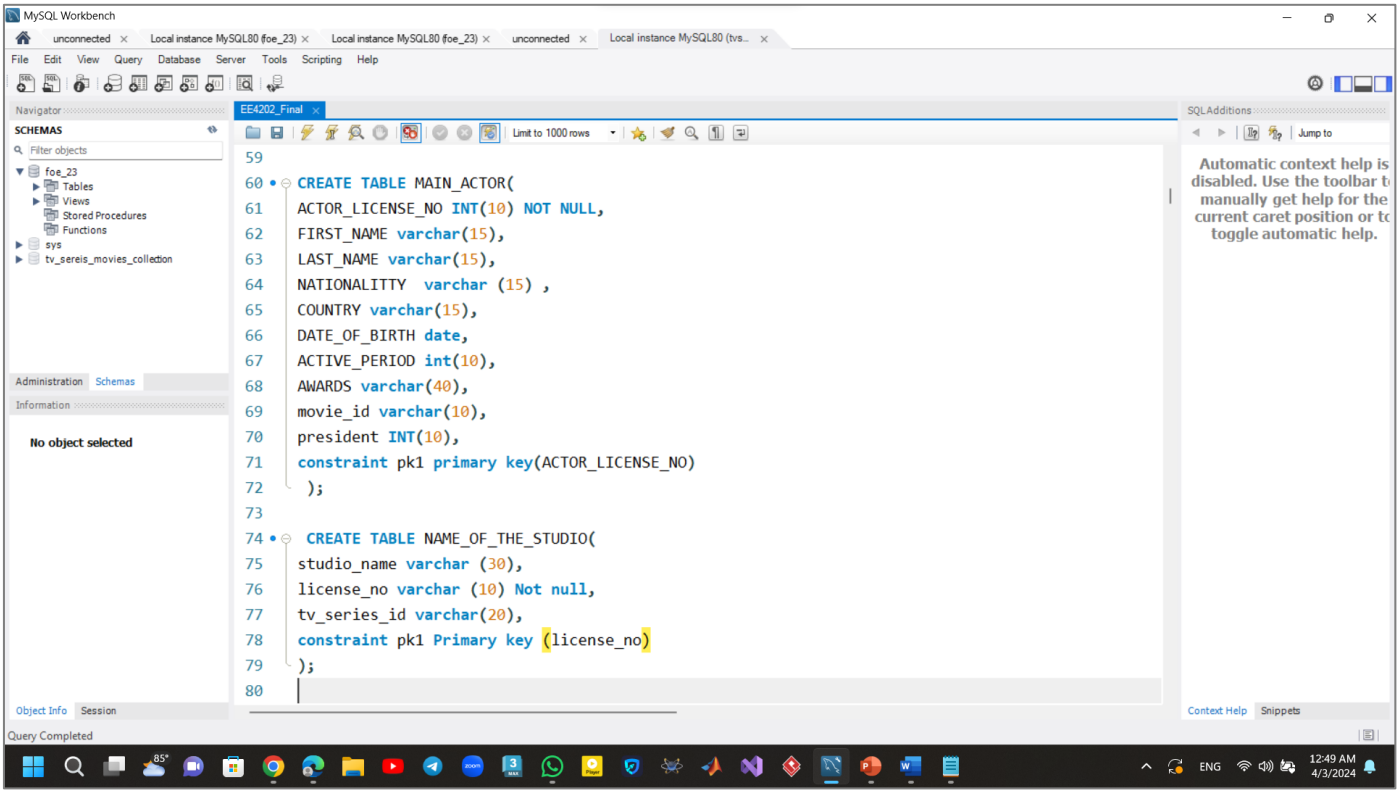


FIGURE 8 : CREATING MAIN ACTOR AND NAME OF THE STUDIO TABLES

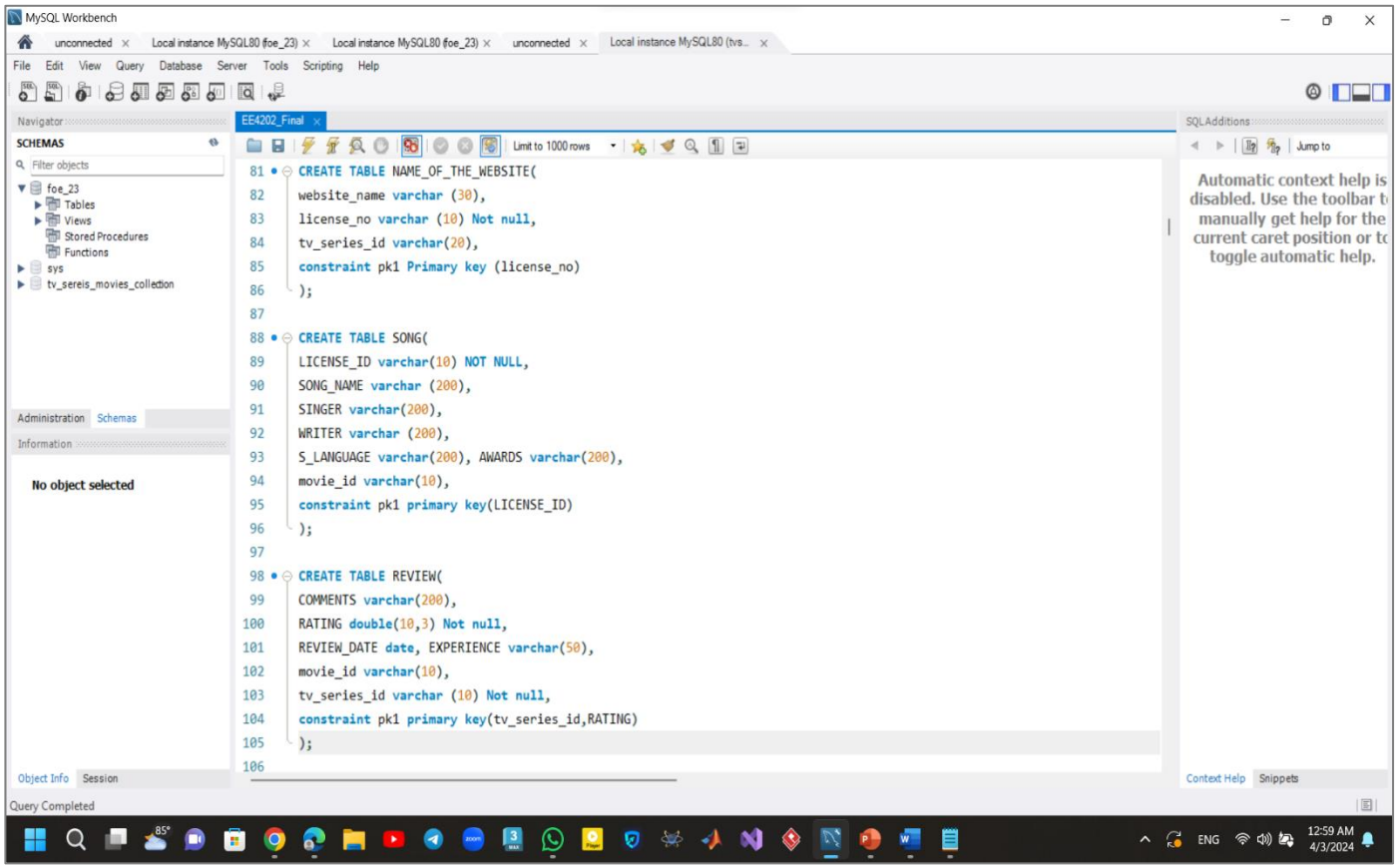


FIGURE 7 : CREATING NAME OF THE WEBSITE, SONGS, REVIEW TABLES

3.3 Data Insertion

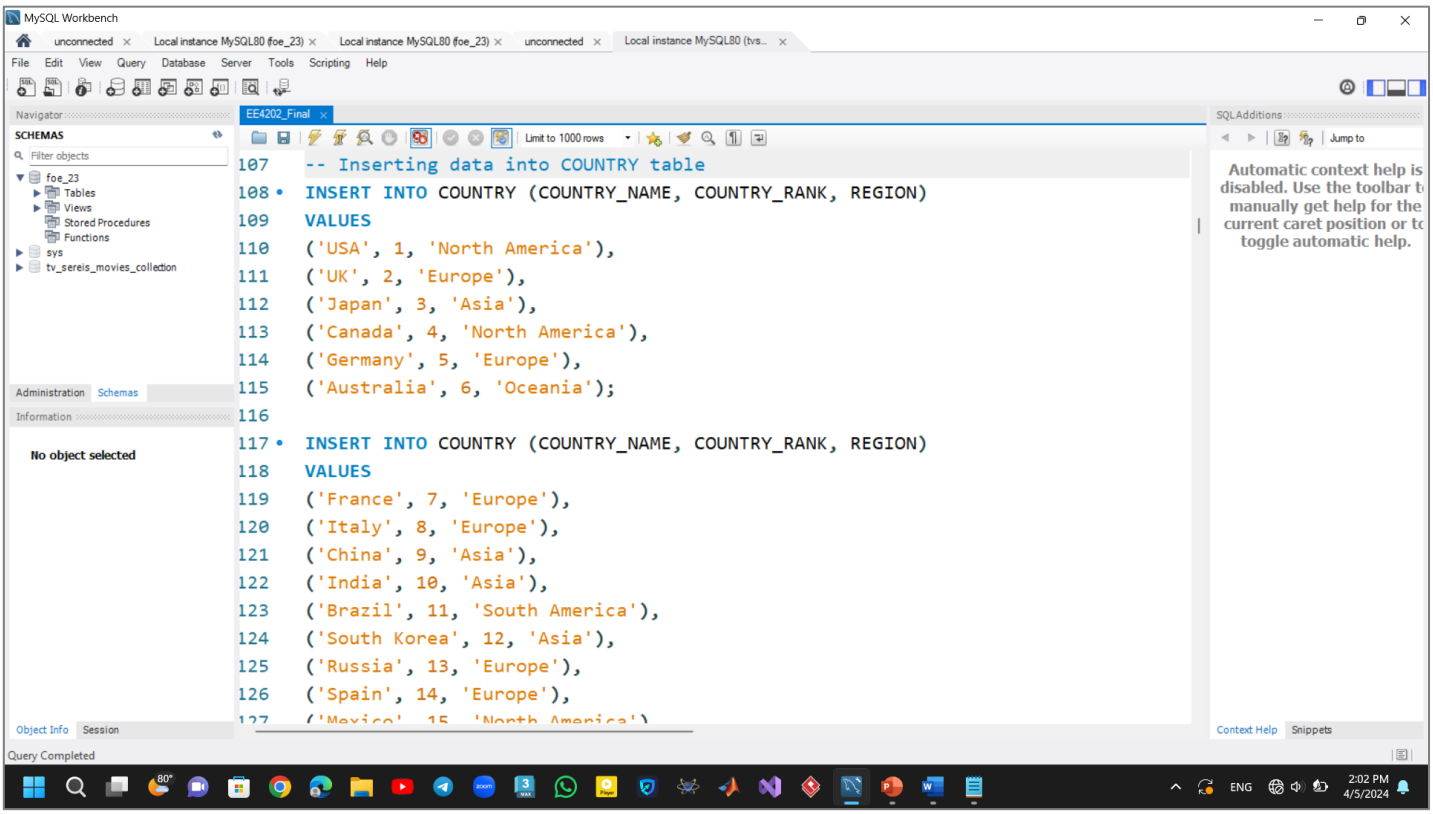


FIGURE 10 : INSERTING TUPLES INTO COUNTRY TABLE

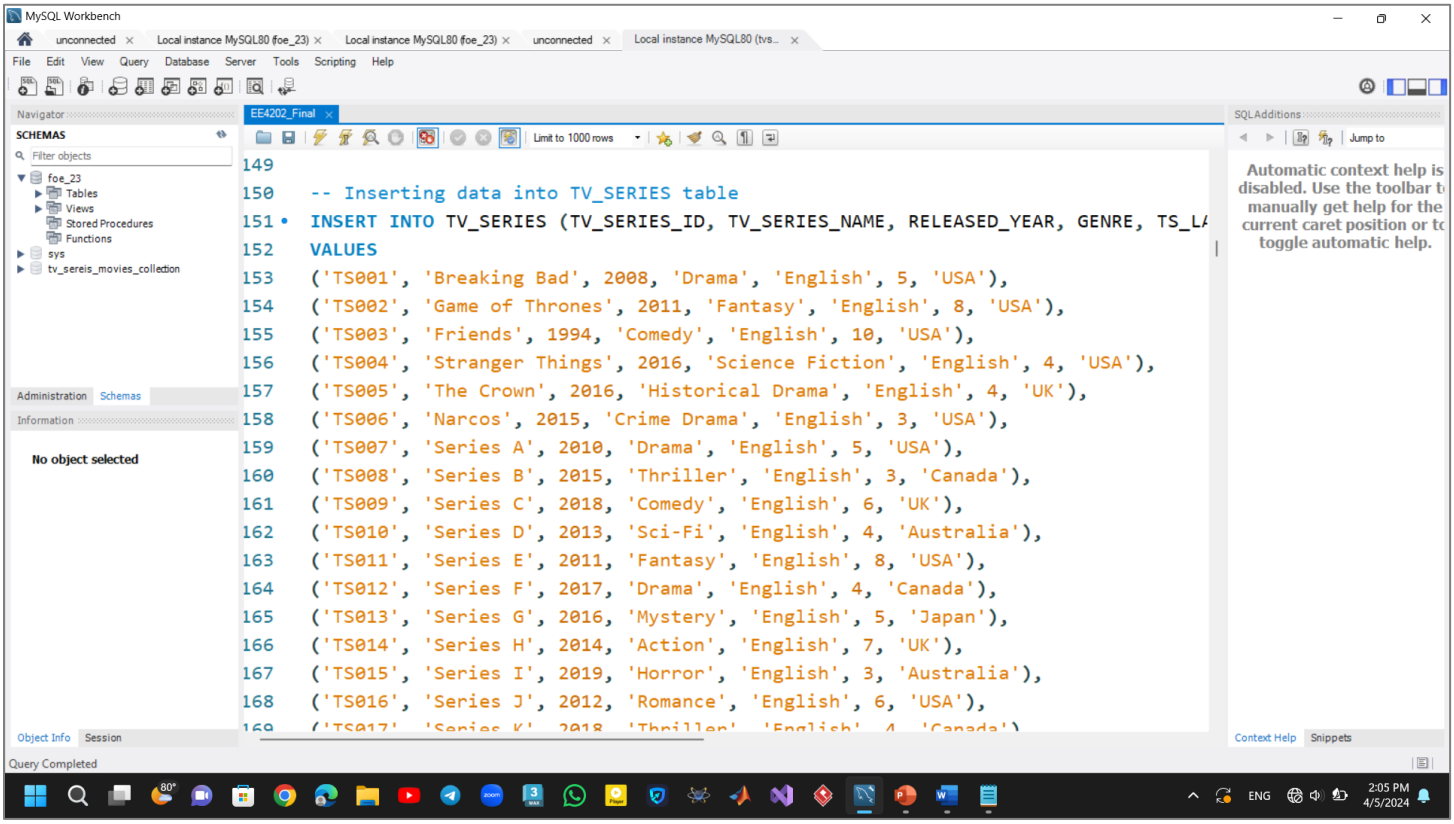


FIGURE 9 : INSERT TUPLES INTO TV SERIES TABLE

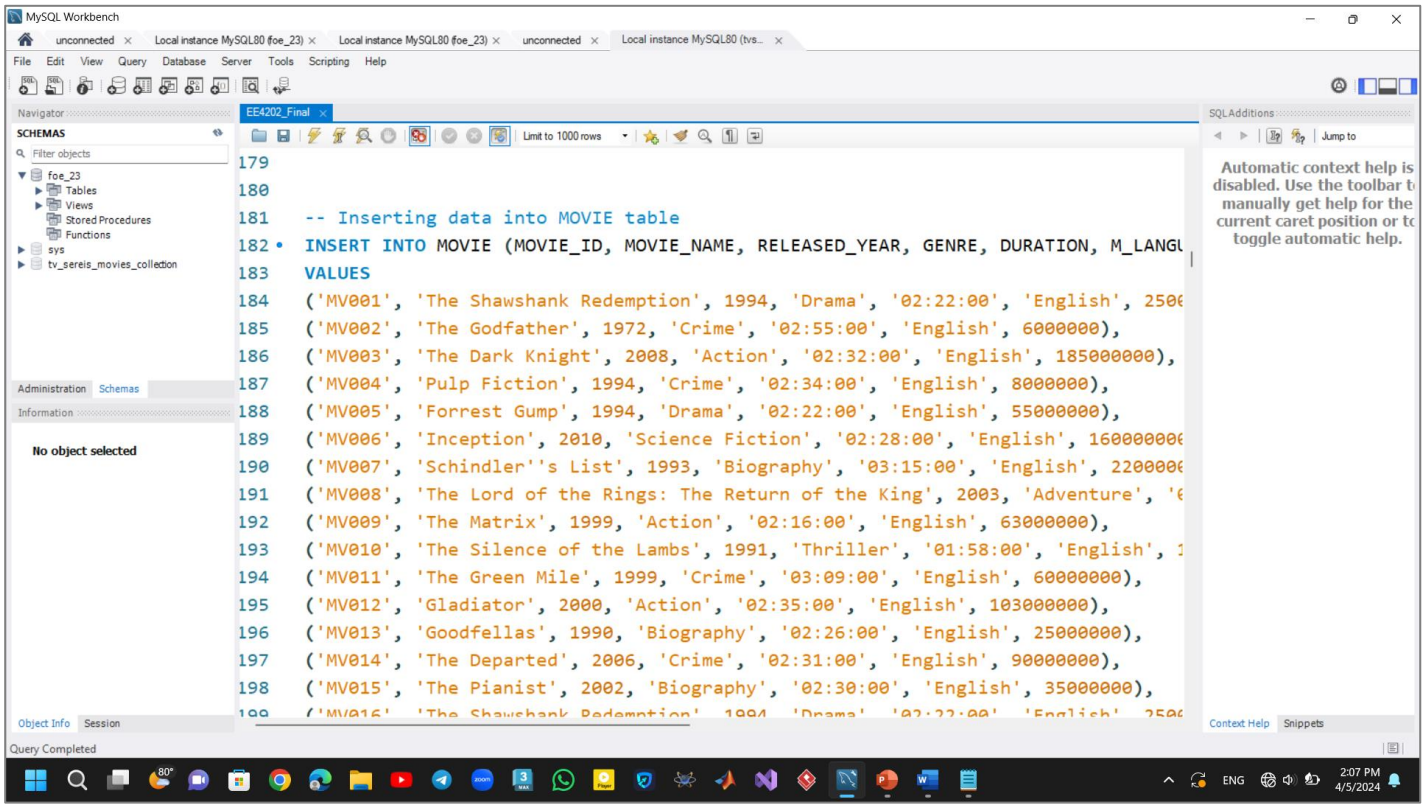


FIGURE 12 : INSERTING TUPLES INTO MOVIE TABLE

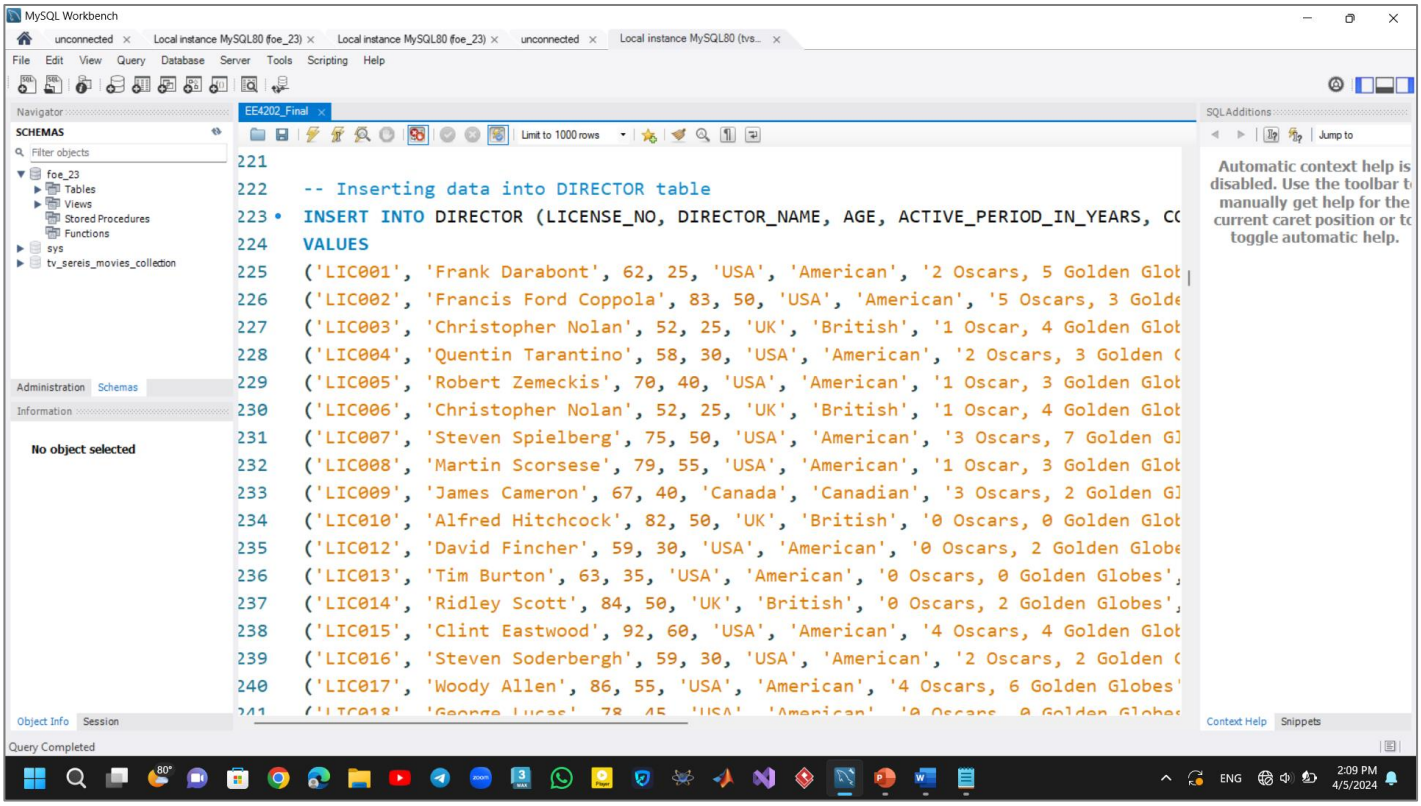


FIGURE 11: INSERTING TUPLES INTO DIRECTOR TABLE

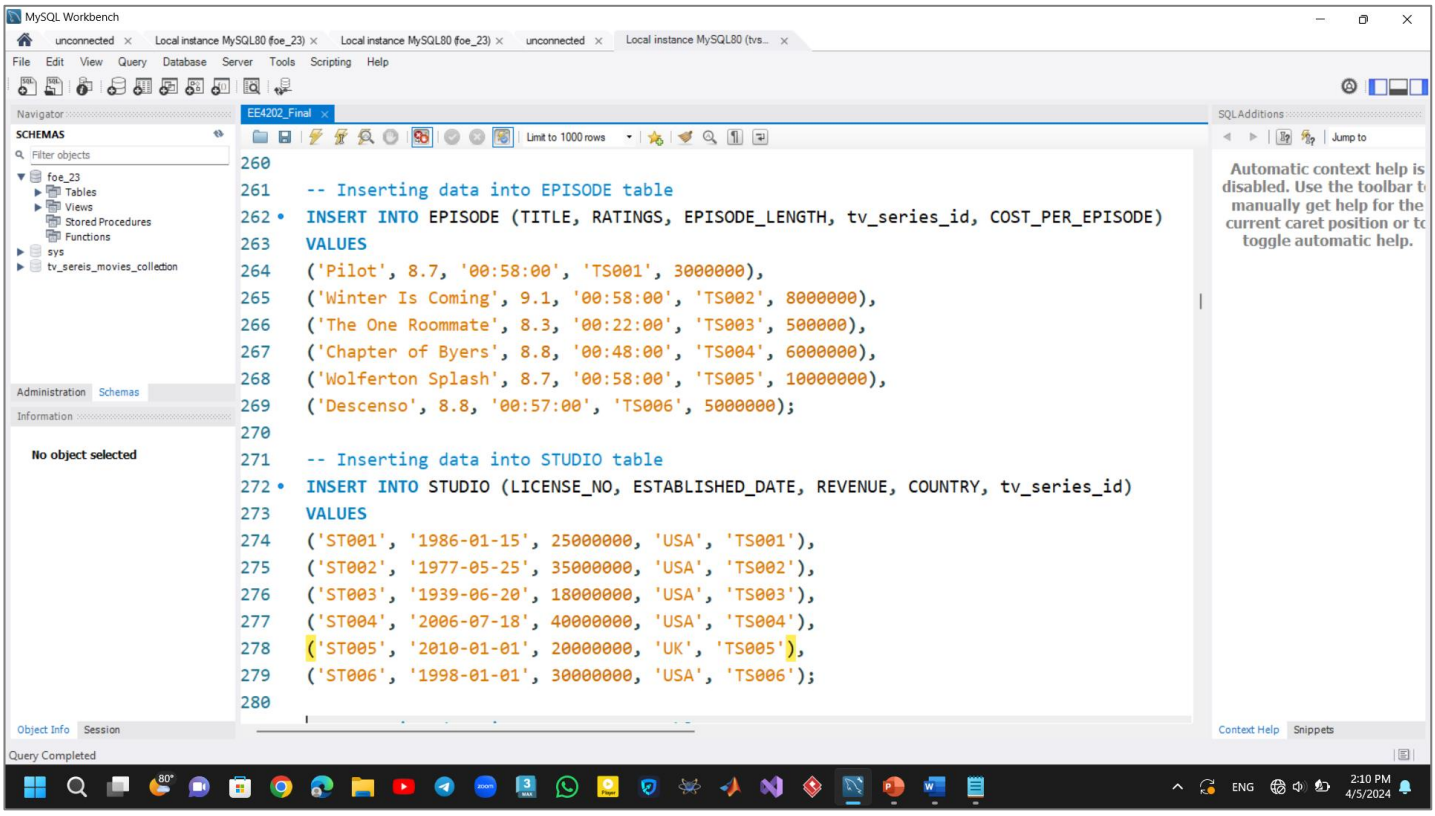


FIGURE 14 : INSERTING TUPLES INTO EPISODE AND STUDIO TABLES

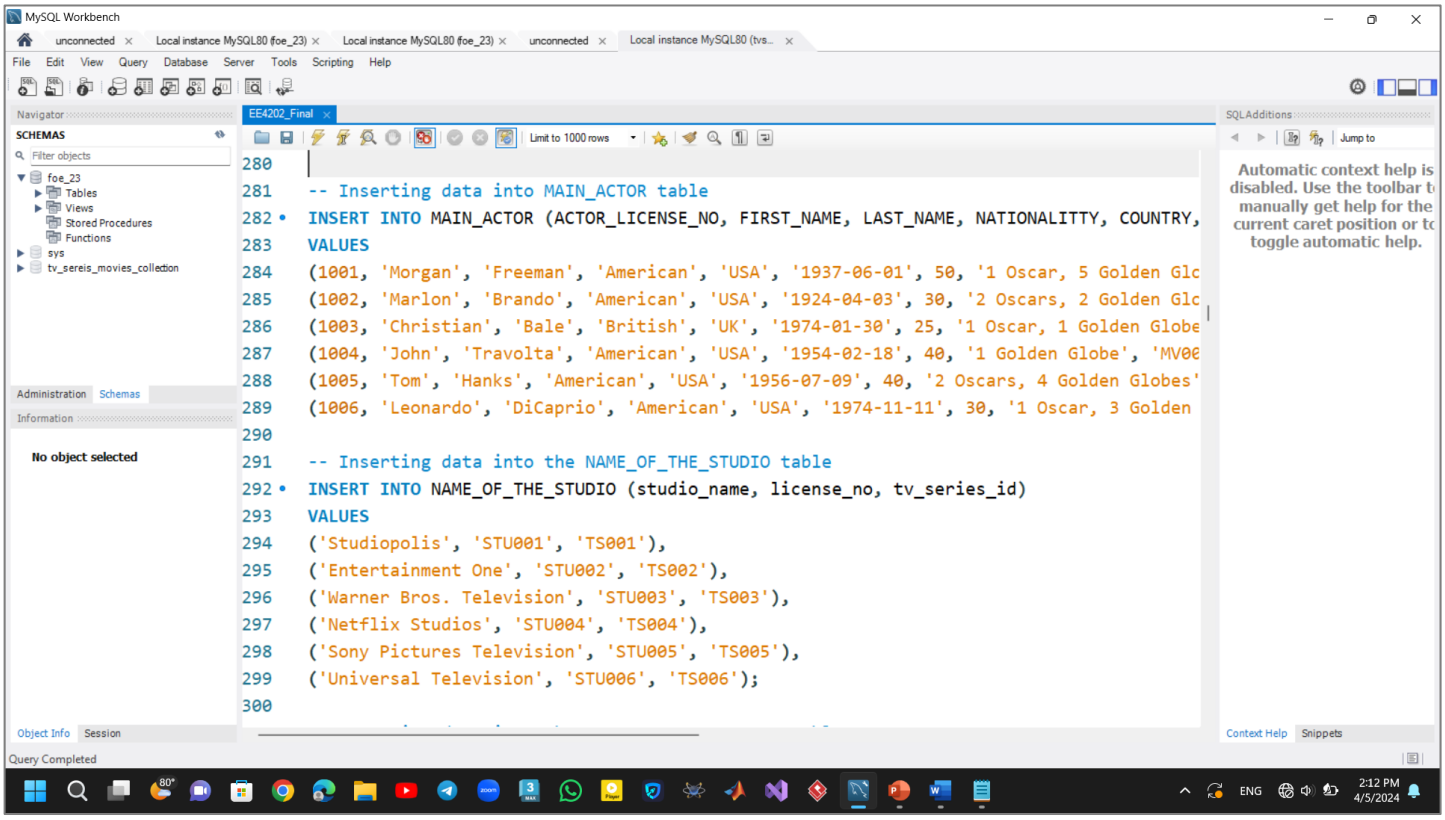


FIGURE 13 : INSERTING TUPLES INTO MAIN_ACTOR AND NAME_OF_THE_STUDIO TABLES

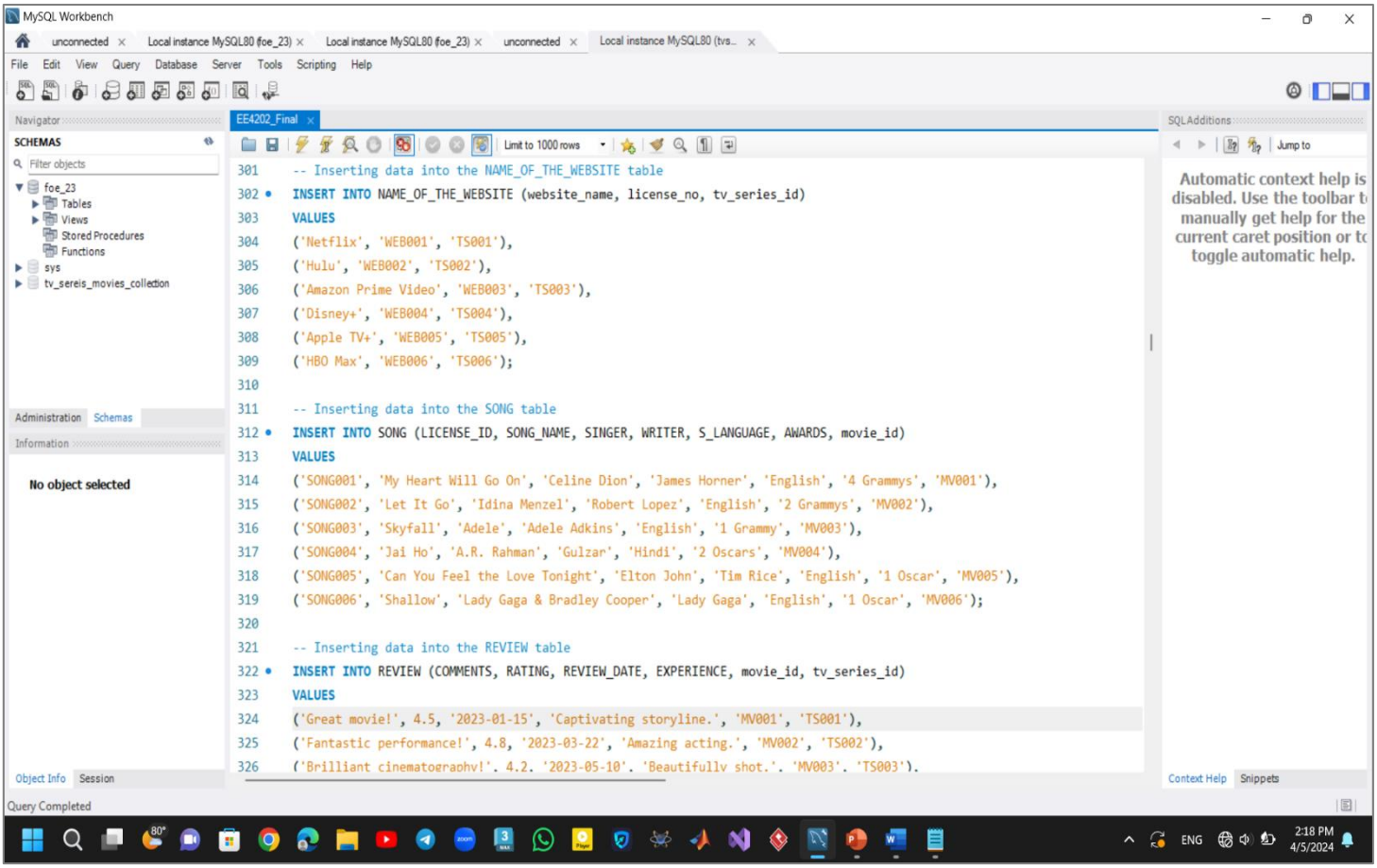
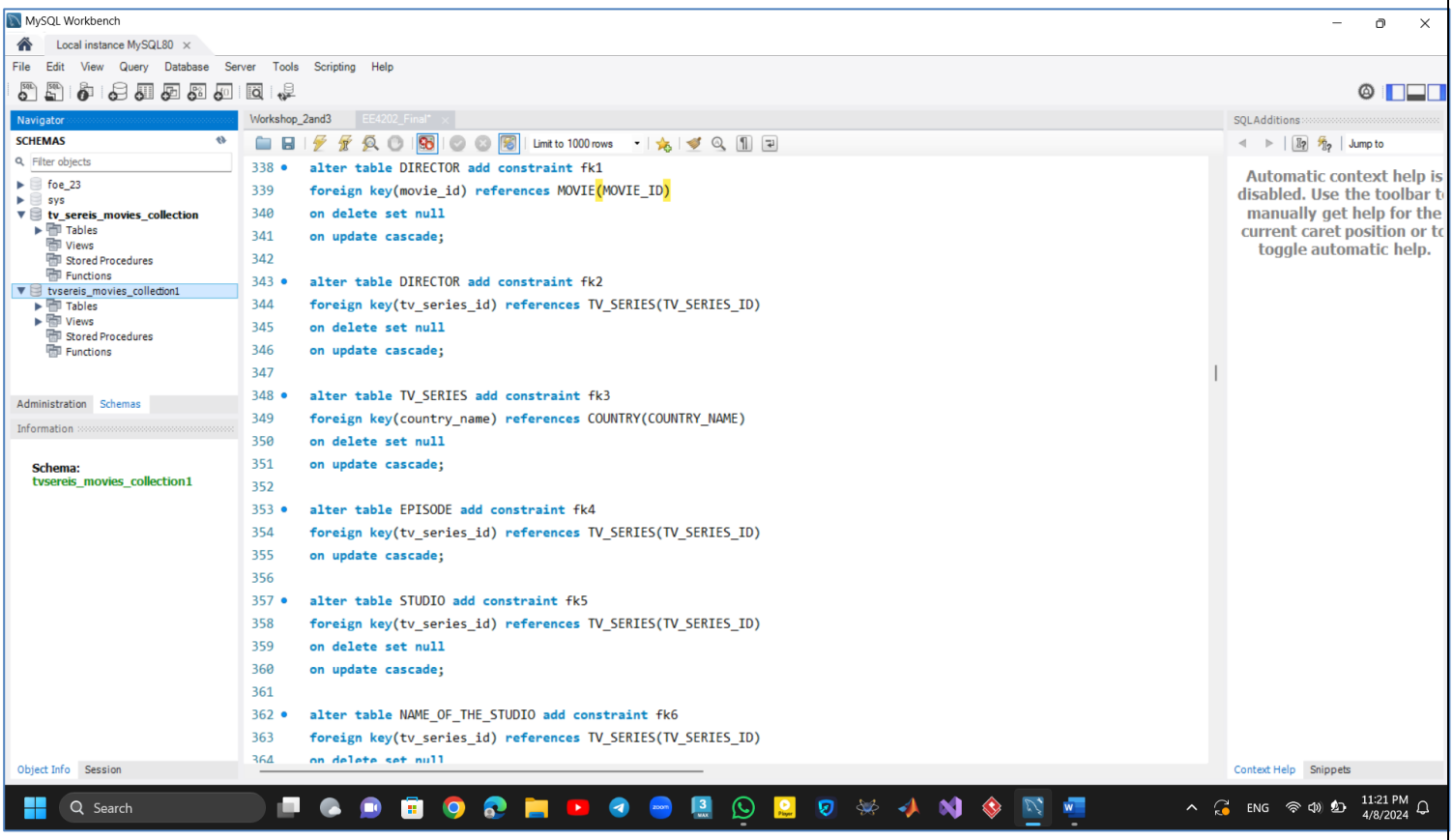
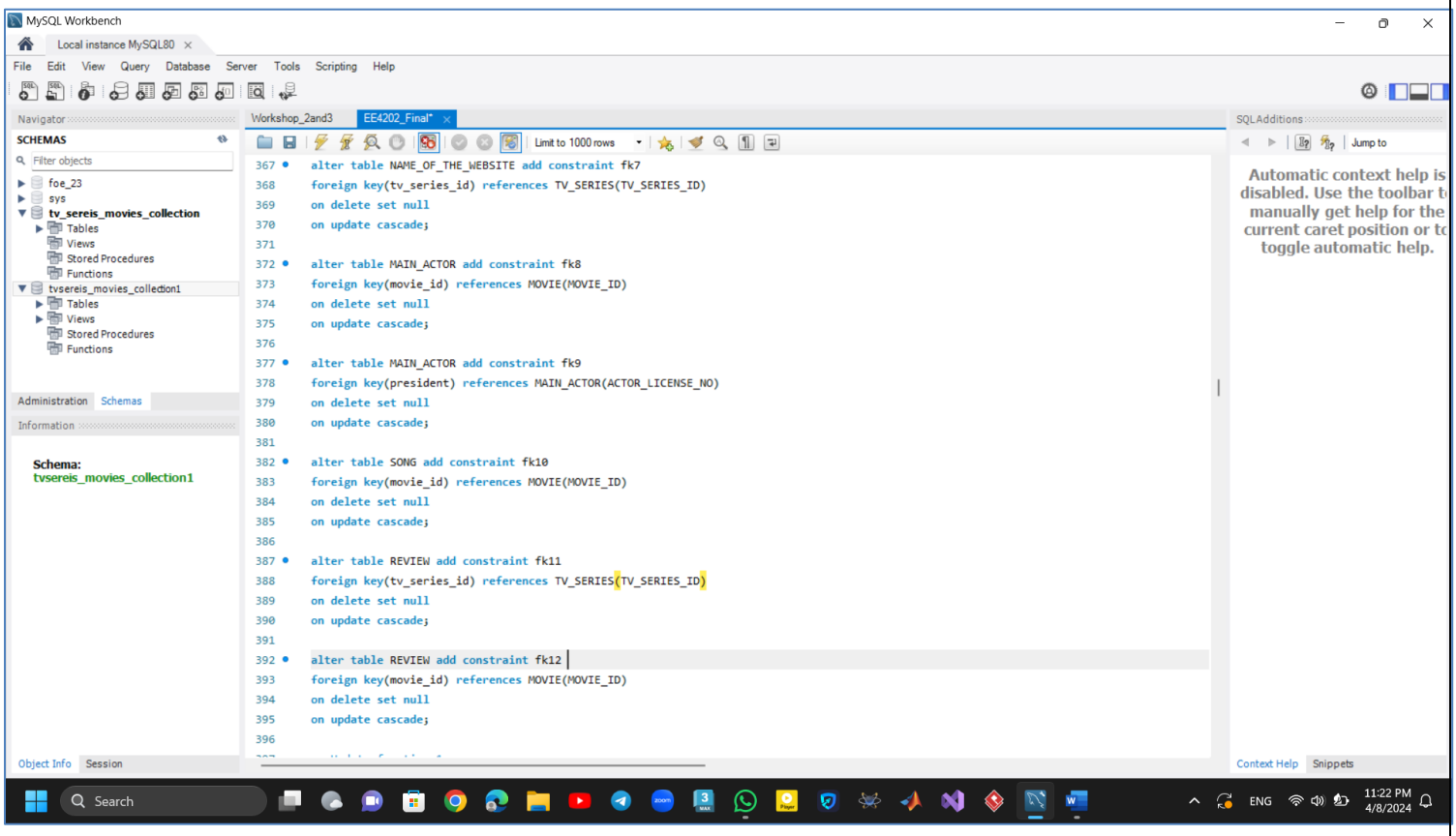


FIGURE 15 : INSERTING TUPLES INTO NAME_OF THE_WEBSITE, SONG AND REVIEW TABLES

3.4 Creating foreign key constraints





```
224
225 •   alter table DIRECTOR add constraint fk1
226     foreign key(movie_id) references MOVIE(MOVIE_ID)
227     on delete set null
228     on update cascade;
```

FIGURE 17 : INSERTION OF 1ST FOREIGN KEY IN DIRECTOR RELATION

```
229
230 •   alter table DIRECTOR add constraint fk2
231     foreign key(tv_series_id) references TV_SERIES(TV_SERIES_ID)
232     on delete set null
233     on update cascade;
```

FIGURE 16 : INSERTION OF 2ND FOREIGN KEY IN DIRECTOR RELATION

```
235 •   alter table TV_SERIES add constraint fk3
236     foreign key(country_name) references COUNTRY(COUNTRY_NAME)
237     on delete set null
238     on update cascade;
239
```

```
239
240 •   alter table EPISODE add constraint fk4
241     foreign key(tv_series_id) references TV_SERIES(TV_SERIES_ID)
242     on update cascade;
243
```

```
248
249 •   alter table NAME_OF_THE_STUDIO add constraint fk6
250     foreign key(tv_series_id) references TV_SERIES(TV_SERIES_ID)
251     on delete set null
252     on update cascade;
```

FIGURE 18 : INSERTION OF 1ST FOREIGN KEY IN EPISODE RELATION

```
254 •   alter table NAME_OF_THE_WEBSITE add constraint fk7
255     foreign key(tv_series_id) references TV_SERIES(TV_SERIES_ID)
256     on delete set null
257     on update cascade;
```

FIGURE 19 : INSERTION OF 1ST FOREIGN KEY IN NAME_OF_THE_WEBSITE RELATION

```
243
244 •   alter table STUDIO add constraint fk5
245     foreign key(tv_series_id) references TV_SERIES(TV_SERIES_ID)
246     on delete set null
247     on update cascade;
```

FIGURE 20 : INSERTION OF 1ST FOREIGN KEY IN NAME_OF_THE_STUDIO RELATION

```
259 •   alter table MAIN_ACTOR add constraint fk8
260     foreign key(movie_id) references MOVIE(MOVIE_ID)
261     on delete set null
262     on update cascade;
263
```

FIGURE 21 : INSERTION OF 1ST FOREIGN KEY IN MAIN_ACTOR RELATION

```
264 •   alter table MAIN_ACTOR add constraint fk9
265     foreign key(president) references MAIN_ACTOR(ACTOR_LICENSE_NO)
266     on delete set null
267     on update cascade;
268
```

FIGURE 22: INSERTION OF 2ND FOREIGN KEY IN MAIN_ACTOR_RELATION

```
269 •   alter table SONG add constraint fk10
270     foreign key(movie_id) references MOVIE(MOVIE_ID)
271     on delete set null
272     on update cascade;
```

FIGURE 25: INSERTION OF 1ST FOREIGN KEY IN SONG RELATION

```
274 •   alter table REVIEW add constraint fk11
275     foreign key(tv_series_id) references TV_SERIES(TV_SERIES_ID)
276     on delete set null
277     on update cascade;
```

FIGURE 24: INSERTION OF 1ST FOREIGN KEY IN REVEW RELATION

```
279 •   alter table REVIEW add constraint fk12
280     foreign key(movie_id) references MOVIE(MOVIE_ID)
281     on delete set null
282     on update cascade;
```

FIGURE 23: INSERTION OF 2ND FOREIGN KEY IN REVIEW RELATION

3.5 Updating and Deleting

COUNTRY_NAME	COUNTRY_RANK	REGION
Australia	6	Oceania
Canada	4	North America
Germany	5	Europe
Japan	3	Asia
UK	2	Europe
USA	1	North America
NULL	NULL	NULL

FIGURE 29 : TABLE OF COUNTRY RELATION BEFORE UPDATE

```
298     -- Update function 1
299 •   UPDATE COUNTRY
300     SET COUNTRY_RANK = 10
301     WHERE COUNTRY_NAME = 'Japan';
302
303     -- Update function 2
304 •   UPDATE COUNTRY
305     SET REGION = 'America-1'
306     WHERE COUNTRY_NAME = 'Canada';
```

FIGURE 28 : QUERY FOR THE ABOVE UPDATING COUNTRY RELATION

COUNTRY_NAME	COUNTRY_RANK	REGION
Australia	6	Oceania
Canada	4	America-1
Germany	5	Europe
Japan	10	Asia
UK	2	Europe
USA	1	North America
NULL	NULL	NULL

FIGURE 27 : TABLE OF COUNTRY RELATION AFTER THE 1ST UPDATE

```
307
308     -- Delete function
309 •   DELETE FROM COUNTRY
310     WHERE COUNTRY_NAME = 'Germany';
```

FIGURE 26 : : QUERY FOR THE ABOVE DELETING IN COUNTRY RELATION

	COUNTRY_NAME	COUNTRY_RANK	REGION
▶	Australia	6	Oceania
	Canada	4	America-1
	Japan	10	Asia
	UK	2	Europe
	USA	1	North America
*	NULL	NULL	NULL

FIGURE 31 : TABLE OF COUNTRY RELATION AFTER THE DELETE

```
312    -- Update function 1
313 •  UPDATE DIRECTOR
314    SET ACTIVE_PERIOD_IN_YEARS = 35
315    WHERE LICENSE_NO = 'LIC001';
316
317    -- Delete function
318 •  DELETE FROM DIRECTOR
319    WHERE LICENSE_NO = 'LIC006';
```

FIGURE 30 : QUERY FOR THE ABOVE UPDATING AND DELETING IN DIRECTOR RELATION

	LICENSE_NO	DIRECTOR_NAME	AGE	ACTIVE_PERIOD_IN_YEARS	COUNTRY	NATIONALITTY	AWARDS_AND_NOMINATIONS	movie_id	tv_series_id
▶	LIC001	Frank Darabont	62	25	USA	American	2 Oscars, 5 Golden Globes	MV001	NULL
	LIC002	Francis Ford Coppola	83	50	USA	American	5 Oscars, 3 Golden Globes	MV002	NULL
	LIC003	Christopher Nolan	52	25	UK	British	1 Oscar, 4 Golden Globes	MV003	NULL
	LIC004	Quentin Tarantino	58	30	USA	American	2 Oscars, 3 Golden Globes	MV004	NULL
	LIC005	Robert Zemeckis	70	40	USA	American	1 Oscar, 3 Golden Globes	MV005	NULL
	LIC006	Christopher Nolan	52	25	UK	British	1 Oscar, 4 Golden Globes	NULL	TS003
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

FIGURE 32 : TABLE OF DIRECTOR RELATION BEFORE UPDATE

	LICENSE_NO	DIRECTOR_NAME	AGE	ACTIVE_PERIOD_IN_YEARS	COUNTRY	NATIONALITTY	AWARDS_AND_NOMINATIONS	movie_id	tv_series_id
▶	LIC001	Frank Darabont	62	35	USA	American	2 Oscars, 5 Golden Globes	MV001	NULL
	LIC002	Francis Ford Coppola	83	50	USA	American	5 Oscars, 3 Golden Globes	MV002	NULL
	LIC003	Christopher Nolan	52	25	UK	British	1 Oscar, 4 Golden Globes	MV003	NULL
	LIC004	Quentin Tarantino	58	30	USA	American	2 Oscars, 3 Golden Globes	MV004	NULL
	LIC005	Robert Zemeckis	70	40	USA	American	1 Oscar, 3 Golden Globes	MV005	NULL
	LIC006	Christopher Nolan	52	25	UK	British	1 Oscar, 4 Golden Globes	NULL	TS003
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

FIGURE 33 : TABLE OF DIRECTOR RELATION AFTER 1ST UPDATE

LICENSE_NO	DIRECTOR_NAME	AGE	ACTIVE_PERIOD_IN_YEARS	COUNTRY	NATIONALITTY	AWARDS_AND_NOMINATIONS	movie_id	tv_series_id
LIC001	Frank Darabont	62	35	USA	American	2 Oscars, 5 Golden Globes	MV001	NULL
LIC002	Francis Ford Coppola	83	50	USA	American	5 Oscars, 3 Golden Globes	MV002	NULL
LIC003	Christopher Nolan	52	25	UK	British	1 Oscar, 4 Golden Globes	MV003	NULL
LIC004	Quentin Tarantino	58	30	USA	American	2 Oscars, 3 Golden Globes	MV004	NULL
LIC005	Robert Zemeckis	70	40	USA	American	1 Oscar, 3 Golden Globes	MV005	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

FIGURE 34 : TABLE OF DIRECTOR RELATION AFTER DELETING

TITLE	RATINGS	EPISODE_LENGTH	tv_series_id	COST_PER_EPISODE
Pilot	8.70	00:58:00	TS001	3000000.00000
Winter Is Coming	9.00	00:58:00	TS002	8000000.00000
Chapter of Byers	8.80	00:48:00	TS004	6000000.00000
Wolferton Splash	8.70	00:58:00	TS005	10000000.00000
NULL	NULL	NULL	NULL	NULL

FIGURE 36 : TABLE OF EPISODE RELATION BEFORE UPDATING

```
435      -- Update function 1
436      UPDATE EPISODE
437      SET COST_PER_EPISODE = 7000000
438      WHERE TITLE = 'Pilot';
439
440      -- Update function 2
441      UPDATE EPISODE
442      SET RATINGS = 8.0
443      WHERE TITLE = 'Winter Is Coming';
444
445      -- Delete function
446      DELETE FROM EPISODE
447      WHERE TITLE = 'Pilot';
```

FIGURE 35 : QUERY FOR THE ABOVE UPDATING AND DELETING IN EPISODE RELATION

TITLE	RATINGS	EPISODE_LENGTH	tv_series_id	COST_PER_EPISODE
Pilot	8.70	00:58:00	TS001	7000000.00000
Winter Is Coming	9.00	00:58:00	TS002	8000000.00000
Chapter of Byers	8.80	00:48:00	TS004	6000000.00000
Wolferton Splash	8.70	00:58:00	TS005	10000000.00000
NULL	NULL	NULL	NULL	NULL

FIGURE 37 : TABLE OF EPISODE RELATION AFTER 1ST AND 2ND UPDATE

TITLE	RATINGS	EPISODE_LENGTH	tv_series_id	COST_PER_EPISODE
Winter Is Coming	8.00	00:58:00	TS002	8000000.00000
Chapter of Byers	8.80	00:48:00	TS004	6000000.00000
Wolferton Splash	8.70	00:58:00	TS005	10000000.00000
NULL	NULL	NULL	NULL	NULL

FIGURE 38 : TABLE OF EPISODE RELATION AFTER DELETING

ACTOR_LICENSE_NO	FIRST_NAME	LAST_NAME	NATIONALITY	COUNTRY	DATE_OF_BIRTH	ACTIVE_PERIOD	AWARDS	movie_id	president
1001	Resets all sorted columns	reeman	American	USA	1937-06-01	50	1 Oscar, 5 Golden Globes	MV001	NULL
1002	Marlon	Brando	American	USA	1924-04-03	30	2 Oscars, 2 Golden Globes	MV002	NULL
1003	Christian	Bale	British	UK	1974-01-30	25	1 Oscar, 1 Golden Globe	MV003	1003
1004	John	Travolta	American	USA	1954-02-18	40	1 Golden Globe	MV004	1004
1005	Tom	Hanks	American	USA	1956-07-09	40	2 Oscars, 4 Golden Globes	MV005	1005
1006	Leonardo	DiCaprio	American	USA	1974-11-11	30	1 Oscar, 3 Golden Globes	MV006	1006
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

FIGURE 40 : TABLE OF MAIN_ACTOR RELATION BEFORE UPDATING

```
335     -- Update function 1
336 •   UPDATE MAIN_ACTOR
337     SET AWARDS = '3 Oscars, 5 Golden Glo
338     WHERE ACTOR_LICENSE_NO = 1001;
339
340     -- Update function 2
341 •   UPDATE MAIN_ACTOR
342     SET DATE_OF_BIRTH = '1976-11-11'
343     WHERE ACTOR_LICENSE_NO = 1006;
344
345     -- Delete function
346 •   DELETE FROM MAIN_ACTOR
347     WHERE ACTOR_LICENSE_NO = 1004;
```

FIGURE 39 : QUERY FOR THE ABOVE UPDATING AND DELETING IN MAIN_ACTOR RELATION

ACTOR_LICENSE_NO	FIRST_NAME	LAST_NAME	NATIONALITY	COUNTRY	DATE_OF_BIRTH	ACTIVE_PERIOD	AWARDS	movie_id	president
1001	Morgan	Freeman	American	USA	1937-06-01	50	3 Oscars, 5 Golden Globes	MV001	NULL
1002	Marlon	Brando	American	USA	1924-04-03	30	2 Oscars, 2 Golden Globes	MV002	NULL
1003	Christian	Bale	British	UK	1974-01-30	25	1 Oscar, 1 Golden Globe	MV003	1003
1004	John	Travolta	American	USA	1954-02-18	40	1 Golden Globe	MV004	1004
1005	Tom	Hanks	American	USA	1956-07-09	40	2 Oscars, 4 Golden Globes	MV005	1005
1006	Leonardo	DiCaprio	American	USA	1976-11-11	30	1 Oscar, 3 Golden Globes	MV006	1006
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

1004

FIGURE 41 : TABLE OF MAIN_ACTOR RELATION AFTER 1ST AND 2ND UPDATE

ACTOR_LICENSE_NO	FIRST_NAME	LAST_NAME	NATIONALITTY	COUNTRY	DATE_OF_BIRTH	ACTIVE_PERIOD	AWARDS	movie_id	president
1001	Morgan	Freeman	American	USA	1937-06-01	50	3 Oscars, 5 Golden Globes	MV001	NULL
1002	Marlon	Brando	American	USA	1924-04-03	30	2 Oscars, 2 Golden Globes	MV002	NULL
1003	Christian	Bale	British	UK	1974-01-30	25	1 Oscar, 1 Golden Globe	MV003	1003
1005	Tom	Hanks	American	USA	1956-07-09	40	2 Oscars, 4 Golden Globes	MV005	1005
1006	Leonardo	DiCaprio	American	USA	1976-11-11	30	1 Oscar, 3 Golden Globes	MV006	1006
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

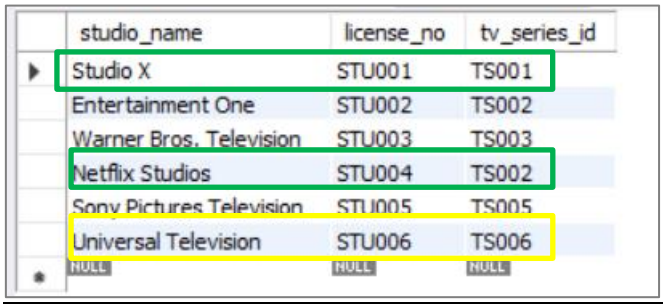
FIGURE 42 : TABLE OF MAIN_ACTOR RELATION AFTER DELETING

studio_name	license_no	tv_series_id
Studioopolis	STU001	TS001
Entertainment One	STU002	TS002
Warner Bros. Television	STU003	TS003
Netflix Studios	STU004	TS004
Sony Pictures Television	STU005	TS005
Universal Television	STU006	TS006
NULL	NULL	NULL

FIGURE 43 : TABLE OF NAME_OF_THE_STUDIO RELATION BEFORE UPDATING

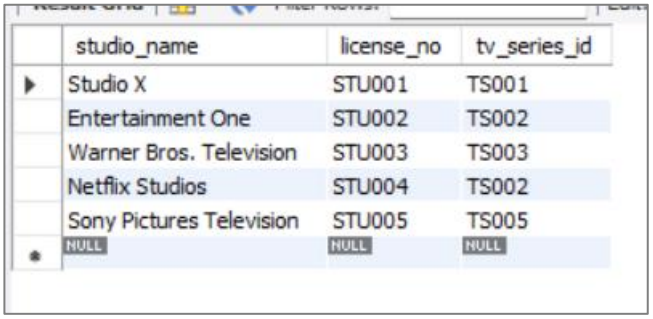
```
378    -- Update function 1
379 •  UPDATE NAME_OF_THE_STUDIO
380    SET studio_name = 'Studio X'
381    WHERE license_no = 'STU001';
382
383    -- Update function 2
384 •  UPDATE NAME_OF_THE_STUDIO
385    SET tv_series_id = 'TS002'
386    WHERE license_no = 'STU004';
387
388    -- Delete function
389 •  DELETE FROM NAME_OF_THE_STUDIO
390    WHERE license_no = 'STU006';
```

FIGURE 44 : QUERY FOR THE ABOVE UPDATING AND DELETING IN NAME_OF_THE_STUDIO RELATION



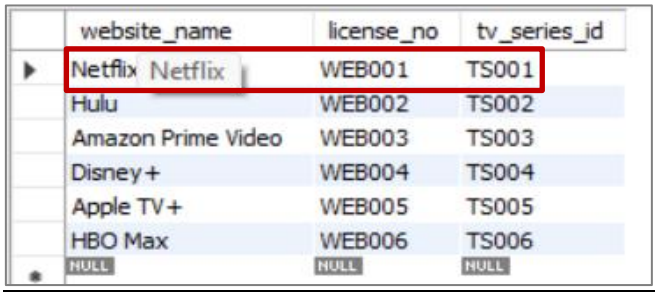
studio_name	license_no	tv_series_id
Studio X	STU001	TS001
Entertainment One	STU002	TS002
Warner Bros. Television	STU003	TS003
Netflix Studios	STU004	TS002
Sony Pictures Television	STU005	TS005
Universal Television	STU006	TS006
NULL	NULL	NULL

FIGURE 46 : TABLE OF NAME_OF_THE_STUDIO RELATION AFTER 1ST AND 2ND UPDATING



studio_name	license_no	tv_series_id
Studio X	STU001	TS001
Entertainment One	STU002	TS002
Warner Bros. Television	STU003	TS003
Netflix Studios	STU004	TS002
Sony Pictures Television	STU005	TS005
NULL	NULL	NULL

FIGURE 45 : TABLE OF NAME_OF_THE_STUDIO RELATION AFTER DELETING



website_name	license_no	tv_series_id
Netflix	WEB001	TS001
Hulu	WEB002	TS002
Amazon Prime Video	WEB003	TS003
Disney+	WEB004	TS004
Apple TV+	WEB005	TS005
HBO Max	WEB006	TS006
NULL	NULL	NULL

FIGURE 48 : TABLE OF NAME_OF_THE_WEBSITE RELATION BEFORE UPDATING

```
392     -- Update function 1
393 •   UPDATE NAME_OF_THE_WEBSITE
394     SET website_name = 'Netflix Prime'
395     WHERE license_no = 'WEB001';
396
397     -- Update function 2
398 •   UPDATE NAME_OF_THE_WEBSITE
399     SET tv_series_id = 'TS005'
400     WHERE license_no = 'WEB003';
401
402     -- Delete function
403 •   DELETE FROM NAME_OF_THE_WEBSITE
404     WHERE license_no = 'WEB006';
```

FIGURE 47: QUERY FOR THE ABOVE UPDATING AND DELETING IN NAME_OF_THE_WEBSITE RELATION

website_name	license_no	tv_series_id
Netflix Prime	WEB001	TS001
Hulu	WEB002	TS002
Amazon Prime Video	WEB003	TS005
Disney+	WEB004	TS004
Apple TV+	WEB005	TS005
HBO Max	WEB006	TS006
NULL	NULL	NULL

FIGURE 49 : : TABLE OF NAME_OF_THE_WEBSITE RELATION AFTER 1ST AND 2ND UPDATING

website_name	license_no	tv_series_id
Netflix Prime	WEB001	TS001
Hulu	WEB002	TS002
Amazon Prime Video	WEB003	TS005
Disney+	WEB004	TS004
Apple TV+	WEB005	TS005
NULL	NULL	NULL

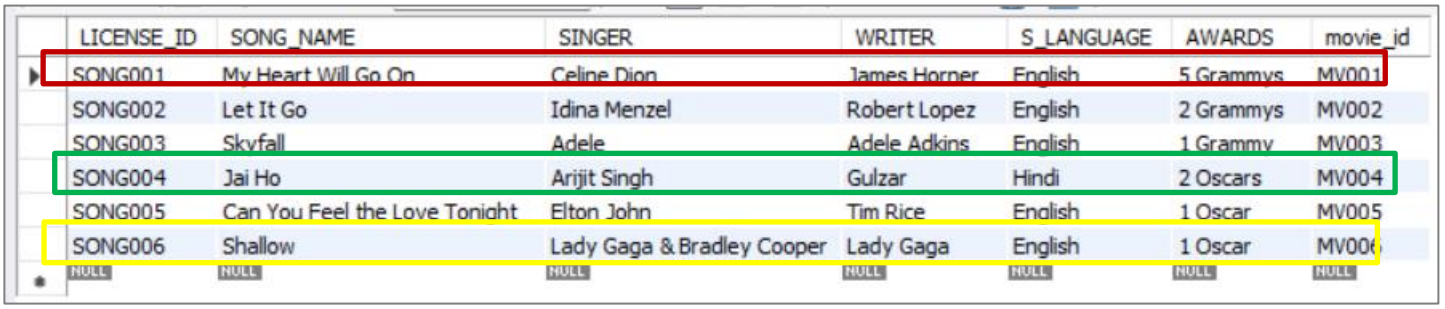
FIGURE 50 : : TABLE OF NAME_OF_THE_WEBSITE RELATION AFTER DELETING

LICENSE_ID	SONG_NAME	SINGER	WRITER	S_LANGUAGE	AWARDS	movie_id
SONG001	My Heart Will Go On	Celine Dion	James Horner	English	4 Grammys	MV001
SONG002	Let It Go	Idina Menzel	Robert Lopez	English	2 Grammys	MV002
SONG003	Skyfall	Adele	Adele Adkins	English	1 Grammy	MV003
SONG004	Jai Ho	A.R. Rahman	Gulzar	Hindi	2 Oscars	MV004
SONG005	Can You Feel the Love Tonight	Elton John	Tim Rice	English	1 Oscar	MV005
SONG006	Shallow	Lady Gaga & Bradley Cooper	Lady Gaga	English	1 Oscar	MV006
NULL	NULL	NULL	NULL	NULL	NULL	NULL

FIGURE 52: TABLE OF SONG RELATION BEFORE UPDATING

```
406     -- Update function 1
407 •   UPDATE SONG
408     SET AWARDS = '5 Grammys'
409     WHERE LICENSE_ID = 'SONG001';
410
411     -- Update function 2
412 •   UPDATE SONG
413     SET SINGER = 'Arijit Singh'
414     WHERE LICENSE_ID = 'SONG004';
415
416     -- Delete function
417 •   DELETE FROM SONG
418     WHERE LICENSE_ID = 'SONG006';
```

FIGURE 51 : QUERY FOR THE ABOVE UPDATING AND DELETING IN SONG RELATION



LICENSE_ID	SONG_NAME	SINGER	WRITER	S_LANGUAGE	AWARDS	movie_id
SONG001	My Heart Will Go On	Celine Dion	James Horner	English	5 Grammys	MV001
SONG002	Let It Go	Idina Menzel	Robert Lopez	English	2 Grammys	MV002
SONG003	Skyfall	Adele	Adele Adkins	English	1 Grammy	MV003
SONG004	Jai Ho	Arijit Singh	Gulzar	Hindi	2 Oscars	MV004
SONG005	Can You Feel the Love Tonight	Elton John	Tim Rice	English	1 Oscar	MV005
SONG006	Shallow	Lady Gaga & Bradley Cooper	Lady Gaga	English	1 Oscar	MV006
NULL	NULL	NULL	NULL	NULL	NULL	NULL

FIGURE 53 : TABLE OF SONG RELATION AFTER 1ST AND 2ND UPDATING

LICENSE_ID	SONG_NAME	SINGER	WRITER	S_LANGUAGE	AWARDS	movie_id
SONG001	My Heart Will Go On	Celine Dion	James Horner	English	5 Grammys	MV001
SONG002	Let It Go	Idina Menzel	Robert Lopez	English	2 Grammys	MV002
SONG003	Skyfall	Adele	Adele Adkins	English	1 Grammy	MV003
SONG004	Jai Ho	Arijit Singh	Gulzar	Hindi	2 Oscars	MV004
SONG005	Can You Feel the Love Tonight	Elton John	Tim Rice	English	1 Oscar	MV005
NULL	NULL	NULL	NULL	NULL	NULL	NULL

FIGURE 54: TABLE OF SONG RELATION AFTER DELETING

COMMENTS	RATING	REVIEW_DATE	EXPERIENCE	movie_id	tv_series_id
Great movie!	4.500	2023-01-15	Captivating storyline.	MV001	TS001
Fantastic performance!	4.800	2023-03-22	Amazing acting.	MV002	TS002
Brilliant cinematography!	4.200	2023-05-10	Beautifully shot.	MV003	TS003
Outstanding music!	4.700	2023-07-05	Great soundtrack.	MV004	TS004
Classic Disney!	4.600	2023-09-18	Nostalgic.	MV005	TS005
Incredible chemistry!	4.900	2023-11-30	Compelling performances.	MV006	TS006
NULL	NULL	NULL	NULL	NULL	NULL

FIGURE 56 : TABLE OF REVIEW RELATION BEFORE UPDATING

```
420     -- Update function 1
421 •   UPDATE REVIEW
422     SET RATING = 4.9
423     WHERE movie_id = 'MV001';
424
425     -- Update function 2
426 •   UPDATE REVIEW
427     SET COMMENTS = 'The best series ever
428     WHERE tv_series_id = 'TS002';
429
430     -- Delete function
431 •   DELETE FROM REVIEW
432     WHERE movie_id = 'MV006';
```

FIGURE 55 : QUERY FOR THE ABOVE UPDATING AND DELETING IN REVIEW RELATION

COMMENTS	RATING	REVIEW_DATE	EXPERIENCE	movie_id	tv_series_id
Great movie!	4.900	2023-01-15	Captivating storyline.	MV001	TS001
The best series ever!	4.800	2023-03-22	Amazing acting.	MV002	TS002
Brilliant cinematography!	4.200	2023-05-10	Beautifully shot.	MV003	TS003
Outstanding music!	4.700	2023-07-05	Great soundtrack.	MV004	TS004
Classic Disney!	4.600	2023-09-18	Nostalgic.	MV005	TS005
Incredible chemistry!	4.900	2023-11-30	Compelling performances.	MV006	TS006
NULL	NULL	NULL	NULL	NULL	NULL

FIGURE 57: TABLE OF REVIEW RELATION AFTER 1ST AND 2ND UPDATING

COMMENTS	RATING	REVIEW_DATE	EXPERIENCE	movie_id	tv_series_id
Great movie!	4.900	2023-01-15	Captivating storyline.	MV001	TS001
The best series ever!	4.800	2023-03-22	Amazing acting.	MV002	TS002
Brilliant cinematography!	4.200	2023-05-10	Beautifully shot.	MV003	TS003
Outstanding music!	4.700	2023-07-05	Great soundtrack.	MV004	TS004
Classic Disney!	4.600	2023-09-18	Nostalgic.	MV005	TS005
NULL	NULL	NULL	NULL	NULL	NULL

FIGURE 58: TABLE OF REVIEW RELATION AFTER DELETING

LICENSE_NO	ESTABLISHED_DATE	REVENUE	COUNTRY	tv_series_id
ST001	1986-01-15	25000000.000	USA	TS001
ST002	1977-05-25	35000000.000	USA	TS002
ST003	1939-06-20	18000000.000	USA	TS003
ST004	2006-07-18	40000000.000	USA	TS004
ST005	2010-01-01	20000000.000	UK	TS005
ST006	1998-01-01	30000000.000	USA	TS006
NULL	NULL	NULL	NULL	NULL

FIGURE 59 :: TABLE OF STUDIO RELATION BEFORE UPDATING

```
351     -- Update function 1
352 ●   UPDATE STUDIO
353     SET REVENUE = 30000000
354     WHERE LICENSE_NO = 'ST001';
355
356     -- Update function 2
357 ●   UPDATE STUDIO
358     SET COUNTRY = 'UK'
359     WHERE LICENSE_NO = 'ST004';
360
361     -- Delete function
362 ●   DELETE FROM STUDIO
363     WHERE LICENSE_NO = 'STU006';
```

Wrong LICENSE_NO. So that row was not removed

FIGURE 60: QUERY FOR THE ABOVE UPDATING AND DELETING IN STUDIO RELATION (WITH SOME ERROR)

LICENSE_NO	ESTABLISHED_DATE	REVENUE	COUNTRY	tv_series_id
ST001	1986-01-15	30000000.000	USA	TS001
ST002	1977-05-25	35000000.000	USA	TS002
ST003	1939-06-20	18000000.000	USA	TS003
ST004	2006-07-18	40000000.000	UK	TS004
ST005	2010-01-01	20000000.000	UK	TS005
ST006	1998-01-01	30000000.000	USA	NULL
NULL	NULL	NULL	NULL	NULL

FIGURE 61 : TABLE OF STUDIO RELATION AFTER 1ST AND 2ND UPDATING

LICENSE_NO	ESTABLISHED_DATE	REVENUE	COUNTRY	tv_series_id
ST001	1986-01-15	30000000.000	USA	TS001
ST002	1977-05-25	35000000.000	USA	TS002
ST003	1939-06-20	18000000.000	USA	TS003
ST004	2006-07-18	40000000.000	UK	TS004
ST005	2010-01-01	20000000.000	UK	TS005
ST006	1998-01-01	30000000.000	USA	NULL
NULL	NULL	NULL	NULL	NULL

FIGURE 63 : TABLE OF STUDIO RELATION AFTER TRYING TO DELETE

```
351      -- Update function 1
352      UPDATE STUDIO
353      SET REVENUE = 30000000
354      WHERE LICENSE_NO = 'ST001';
355
356      -- Update function 2
357      UPDATE STUDIO
358      SET COUNTRY = 'UK'
359      WHERE LICENSE_NO = 'ST004';
360
361      -- Delete function
362      DELETE FROM STUDIO
363      WHERE LICENSE_NO = 'ST006';
```

Now it has removed

LICENSE_NO	ESTABLISHED_DATE	REVENUE	COUNTRY	tv_series_id
ST001	1986-01-15	30000000.000	USA	TS001
ST002	1977-05-25	35000000.000	USA	TS002
ST003	1939-06-20	18000000.000	USA	TS003
ST004	2006-07-18	40000000.000	UK	TS004
ST005	2010-01-01	20000000.000	UK	TS005
NULL	NULL	NULL	NULL	NULL

FIGURE 62 : TABLE OF STUDIO RELATION AFTER DELETING

TV_SERIES_ID	TV_SERIES_NAME	RELEASED_YEAR	GENRE	TS_LANGUAGE	NO_OF_SEASONS	country_name
TS001	Breaking Bad	2008	Drama	English	5	USA
TS002	Game of Thrones	2011	Fantasy	English	8	USA
TS003	Friends	1994	Comedy	English	10	USA
TS004	Stranger Things	2016	Science Fiction	English	4	USA
TS005	The Crown	2016	Historical Drama	English	4	UK
TS006	Narcos	2015	Crime Drama	English	3	USA
NULL	NULL	NULL	NULL	NULL	NULL	NULL

FIGURE 64: TABLE OF TV_SERIES RELATION BEFORE UPDATING

TV_SERIES_ID	TV_SERIES_NAME	RELEASED_YEAR	GENRE	TS_LANGUAGE	NO_OF_SEASONS	country_name
TS001	Breaking Bad	2008	Drama	English	6	USA
TS002	Game of Thrones	2011	Fantasy	English	8	USA
TS003	Friends	1994	Comedy	English	10	USA
TS004	Stranger Things	2016	Science Fiction	English	4	USA
TS005	The Crown	2016	Historical Drama	English	4	UK
TS006	Narcos	2015	Crime Drama	English	3	USA
NULL	NULL	NULL	NULL	NULL	NULL	NULL

FIGURE 65 : TABLE OF TV_SERIES RELATION AFTER 1ST UPDATING

TV_SERIES_ID	TV_SERIES_NAME	RELEASED_YEAR	GENRE	TS_LANGUAGE	NO_OF_SEASONS	country_name
TS001	Breaking Bad	2008	Drama	English	6	USA
TS002	Game of Thrones	2013	Fantasy	English	8	USA
TS003	Friends	1994	Comedy	English	10	USA
TS004	Stranger Things	2016	Science Fiction	English	4	USA
TS005	The Crown	2016	Historical Drama	English	4	UK
TS006	Narcos	2015	Crime Drama	English	3	USA
NULL	NULL	NULL	NULL	NULL	NULL	NULL

FIGURE 66 : TABLE OF TV_SERIES RELATION AFTER 2ND UPDATING

ERROR MESSAGE! WHEN TRYING TO DELETE A TUPLUE FROM THE TV_SERIES RELATION

0	227	23:07:47	alter table EPISODE add constraint fk4 foreign key(tv_series_id) references TV_SERIES(TV_SERIES_ID) on update cascade ON DELETE CASCADE	Error Code: 1826. Duplicate foreign key constraint name 'fk4'	0.000 sec

BECAUSE OF THIS CONSTRAINT

```
240 ●    alter table EPISODE add constraint fk4
241      foreign key(tv_series_id) references TV_SERIES(TV_SERIES_ID)
242      on update cascade;
243
```

```
240 •  alter table EPISODE add constraint fk4
241    foreign key(tv_series_id) references TV_SERIES(TV_SERIES_ID)
242    on update cascade
243    on delete cascade;
```

FIGURE 68 : UPDATED CONSTRAINT

```
364    -- Update function 1
365 •  UPDATE TV_SERIES
366    SET NO_OF_SEASONS = 6
367    WHERE TV_SERIES_ID = 'TS001';
368
369    -- Update function 2
370 •  UPDATE TV_SERIES
371    SET RELEASED_YEAR = 2013
372    WHERE TV_SERIES_NAME = 'Game of Thrones';
373
374    -- Delete function
375 •  DELETE FROM TV_SERIES
376    WHERE TV_SERIES_ID = 'TS006';
```

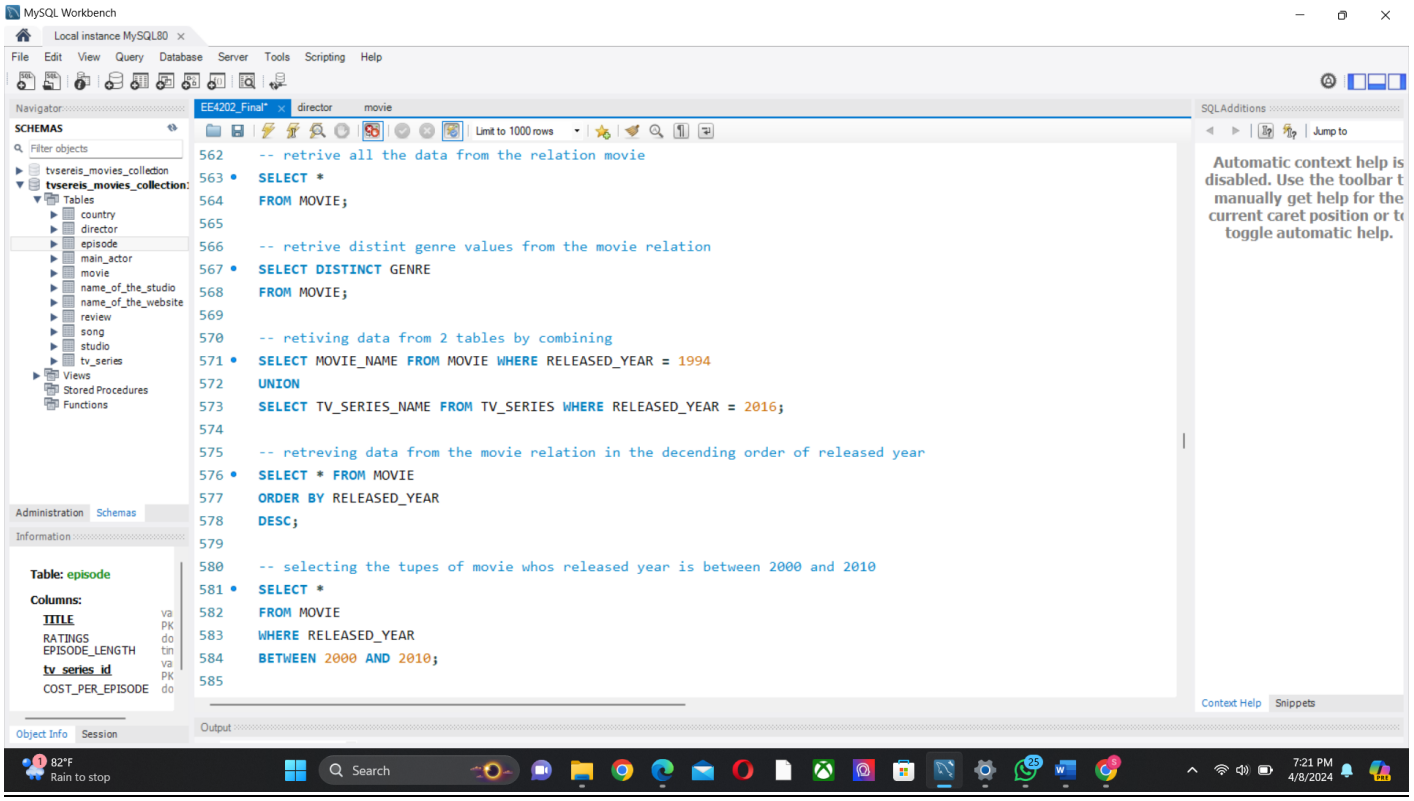
FIGURE 67: QUERY FOR THE ABOVE UPDATING AND DELETING IN TV_SERIES

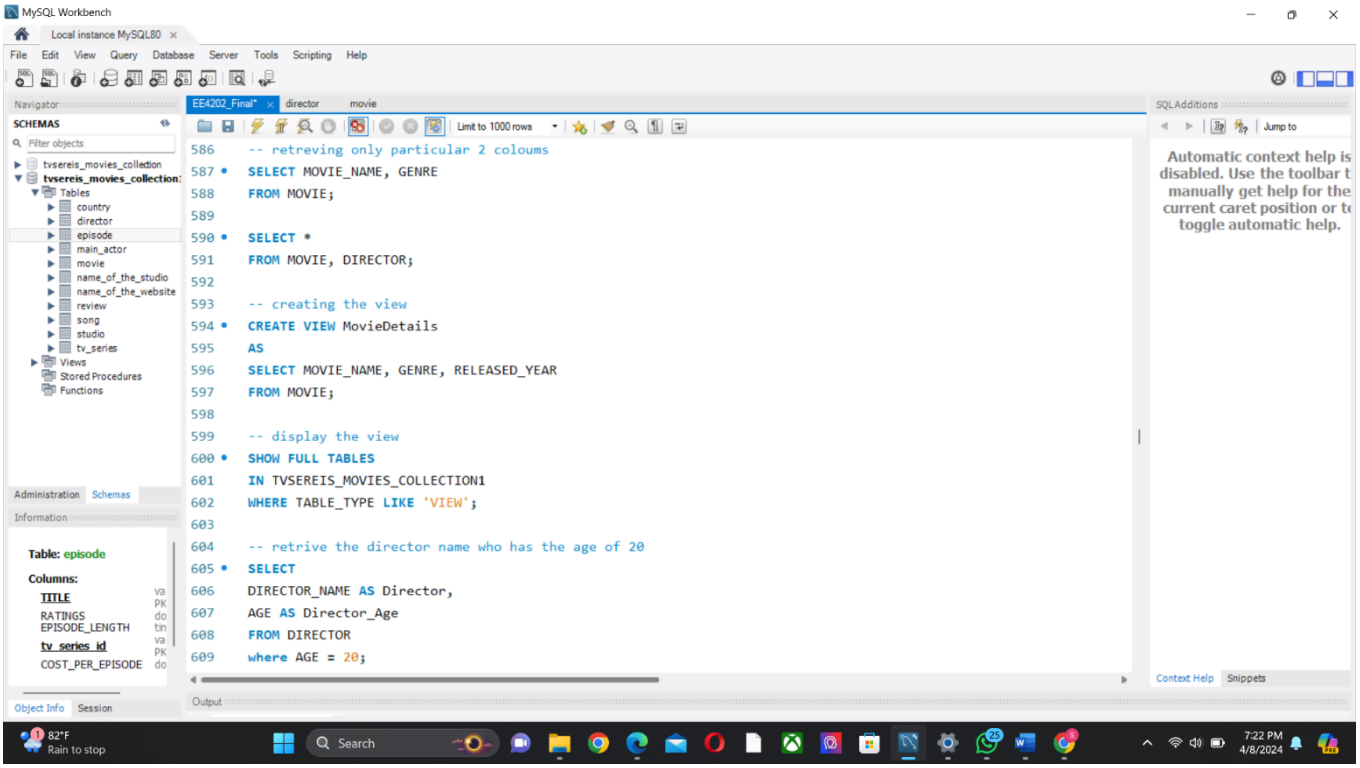
TV_SERIES_ID	TV_SERIES_NAME	RELEASED_YEAR	GENRE	TS_LANGUAGE	NO_OF_SEASONS	country_name
TS001	Breaking Bad	2008	Drama	English	6	USA
TS002	Game of Thrones	2013	Fantasy	English	8	USA
TS003	Friends	1994	Comedy	English	10	USA
TS004	Stranger Things	2016	Science Fiction	English	4	USA
TS005	The Crown	2016	Historical Drama	English	4	UK
NULL	NULL	NULL	NULL	NULL	NULL	NULL

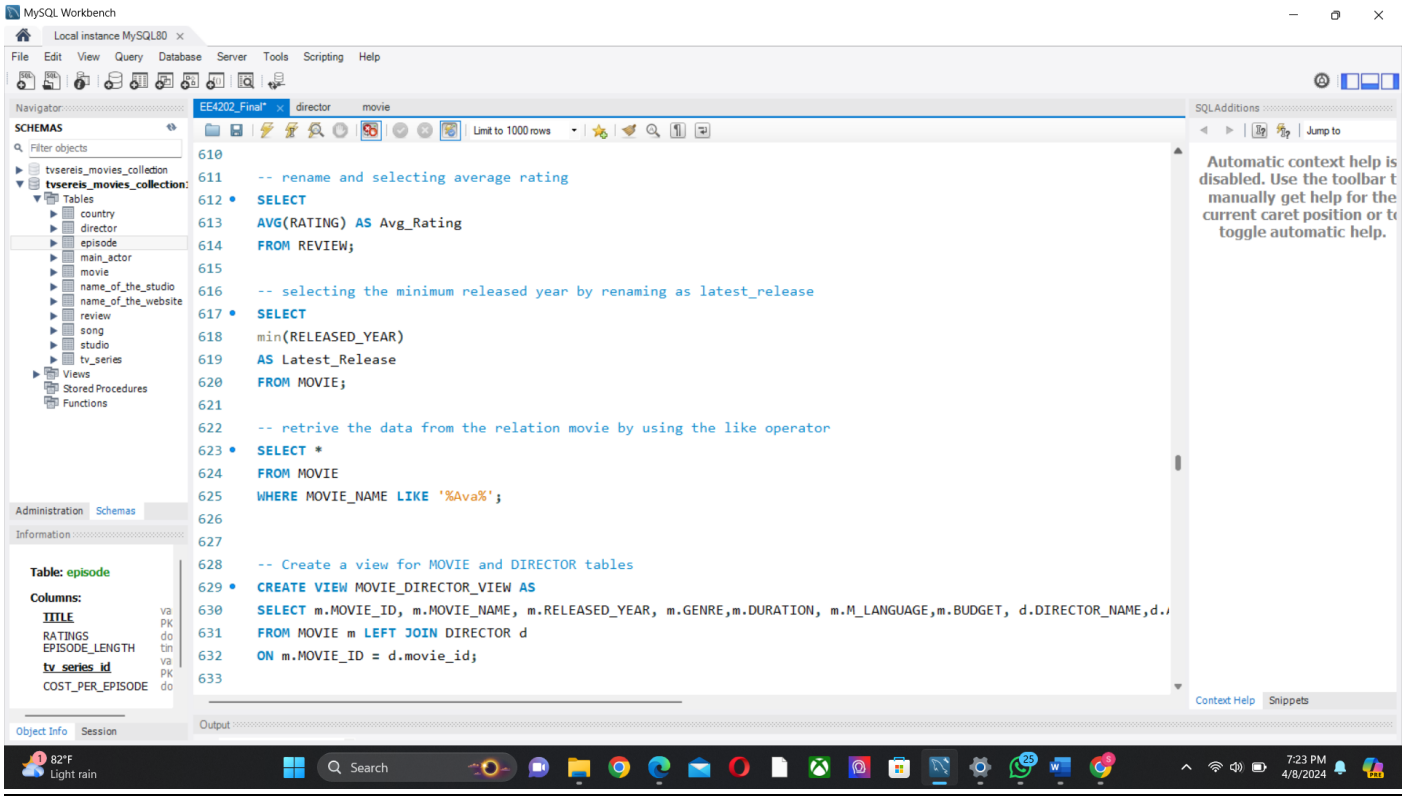
FIGURE 69 : TABLE OF TV_SERIES RELATION AFTER DELETING

4. Chapter 4 - Transactions

i. Simple queries






MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
tvsereis_movies_collection
tvsereis_movies_collection:
Tables
country
director
episode
main_actor
movie
name_of_the_studio
name_of_the_website
review
song
studio
tv_series
Views
Stored Procedures
Functions
Administration Schemas
Information
Table: episode
Columns:
TITLE
RATINGS
EPISODE_LENGTH
tv_series_id
COST_PER_EPISODE
Object Info Session
EE4202_Final* director movie
Limit to 1000 rows
610
611 -- rename and selecting average rating
612 SELECT
613 AVG(RATING) AS Avg_Rating
614 FROM REVIEW;
615
616 -- selecting the minimum released year by renaming as latest_release
617 SELECT
618 min(RELEASED_YEAR)
619 AS Latest_Release
620 FROM MOVIE;
621
622 -- retrive the data from the relation movie by using the like operator
623 SELECT *
624 FROM MOVIE
625 WHERE MOVIE_NAME LIKE '%Ava%';
626
627
628 -- Create a view for MOVIE and DIRECTOR tables
629 CREATE VIEW MOVIE_DIRECTOR_VIEW AS
630 SELECT m.MOVIE_ID, m.MOVIE_NAME, m.RELEASED_YEAR, m.GENRE,m.DURATION, m.M_LANGUAGE,m.BUDGET, d.DIRECTOR_NAME,d.
631 FROM MOVIE m LEFT JOIN DIRECTOR d
632 ON m.MOVIE_ID = d.movie_id;
633
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
Output
82°F Light rain
Search
7:23 PM 4/8/2024

```
448
449 •  SELECT *
450    FROM MOVIE;
```

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

MOVIE_ID	MOVIE_NAME	RELEASED_YEAR	GENRE	DURATION	M_LANGUAGE	BUDGET
MV001	The Shawshank Redemption	1994	Drama	02:22:00	English	25000000.00000
MV002	The Godfather	1972	Drama, Crime	02:55:00	English	6000000.00000
MV003	The Dark Knight	2008	Action	02:32:00	English	200000000.00000
MV004	Pulp Fiction	1994	Crime	02:34:00	English	8000000.00000
MV005	Forrest Gump	1994	Drama	02:22:00	English	55000000.00000
MV006	Inception	2010	Science Fiction	02:28:00	English	160000000.00000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

FIGURE 70 : QUERY AND OUTPUT FOR THE SELECT OPERATION

```
451
452 •  SELECT DISTINCT
453    GENRE FROM MOVIE;
```

Result Grid | Filter Rows:

GENRE
Drama
Drama, Crime
Action
Crime
Science Fiction

FIGURE 71 : : QUERY AND OUTPUT FOR THE DISTINCT SELECT OPERATION

```
454
455 •   SELECT MOVIE_NAME FROM MOVIE WHERE RELEASED_YEAR = 1994
456     UNION
457     SELECT TV_SERIES_NAME FROM TV_SERIES WHERE RELEASED_YEAR = 2016;
458
```

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

MOVIE_NAME
The Shawshank Redemption
Pulp Fiction
Forrest Gump
Stranger Things
The Crown

FIGURE 72 : QUERY AND OUTPUT FOR THE SET OPERATION UNION

```
458
459 •   SELECT * FROM MOVIE ORDER BY RELEASED_YEAR DESC;
460
```

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

MOVIE_ID	MOVIE_NAME	RELEASED_YEAR	GENRE	DURATION	M_LANGUAGE	BUDGET
MV006	Inception	2010	Science Fiction	02:28:00	English	160000000.00000
MV003	The Dark Knight	2008	Action	02:32:00	English	200000000.00000
MV001	The Shawshank Redemption	1994	Drama	02:22:00	English	25000000.00000
MV004	Pulp Fiction	1994	Crime	02:34:00	English	8000000.00000
MV005	Forrest Gump	1994	Drama	02:22:00	English	55000000.00000
MV002	The Godfather	1972	Drama, Crime	02:55:00	English	6000000.00000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

FIGURE 73: QUERY AND OUTPUT FOR THE CREATING ORDERED VIEW

```
460
461 •   SELECT * FROM MOVIE WHERE RELEASED_YEAR BETWEEN 2000 AND 2010;
462
```

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

MOVIE_ID	MOVIE_NAME	RELEASED_YEAR	GENRE	DURATION	M_LANGUAGE	BUDGET
MV003	The Dark Knight	2008	Action	02:32:00	English	200000000.00000
MV006	Inception	2010	Science Fiction	02:28:00	English	160000000.00000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

FIGURE 74 : QUERY AND OUTPUT FOR THE SET OPERATION BETWEEN

```
462
463 •  SELECT MOVIE_NAME, GENRE FROM MOVIE;
464
```

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

MOVIE_NAME	GENRE
The Shawshank Redemption	Drama
The Godfather	Drama, Crime
The Dark Knight	Action
Pulp Fiction	Crime
Forrest Gump	Drama
Inception	Science Fiction

FIGURE 75 : QUERY AND OUTPUT FOR THE PROJECT OPERATION

```
464
465 •  SELECT * FROM MOVIE, DIRECTOR;
466
467
```

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

MOVIE_ID	MOVIE_NAME	RELEASED_YEAR	GENRE	DURATION	M_LANGUAGE	BUDGET	LICENSE_NO	DIRECTOR_NAME	AGE	ACTIVE_PERIO
MV001	The Shawshank Redemption	1994	Drama	02:22:00	English	25000000.00000	LIC005	Robert Zemeckis	70	40
MV001	The Shawshank Redemption	1994	Drama	02:22:00	English	25000000.00000	LIC004	Quentin Tarantino	58	30
MV001	The Shawshank Redemption	1994	Drama	02:22:00	English	25000000.00000	LIC003	Christopher Nolan	52	25
MV001	The Shawshank Redemption	1994	Drama	02:22:00	English	25000000.00000	LIC002	Francis Ford Coppola	83	50
MV001	The Shawshank Redemption	1994	Drama	02:22:00	English	25000000.00000	LIC001	Frank Darabont	62	35
MV002	The Godfather	1972	Drama, Crime	02:55:00	English	6000000.00000	LIC005	Robert Zemeckis	70	40
MV002	The Godfather	1972	Drama, Crime	02:55:00	English	6000000.00000	LIC004	Quentin Tarantino	58	30
MV002	The Godfather	1972	Drama, Crime	02:55:00	English	6000000.00000	LIC003	Christopher Nolan	52	25
MV002	The Godfather	1972	Drama, Crime	02:55:00	English	6000000.00000	LIC002	Francis Ford Coppola	83	50
MV002	The Godfather	1972	Drama, Crime	02:55:00	English	6000000.00000	LIC001	Frank Darabont	62	35
MV003	The Dark Knight	2008	Action	02:32:00	English	200000000.00000	LIC005	Robert Zemeckis	70	40
MV003	The Dark Knight	2008	Action	02:32:00	English	200000000.00000	LIC004	Quentin Tarantino	58	30
MV003	The Dark Knight	2008	Action	02:32:00	English	200000000.00000	LIC003	Christopher Nolan	52	25
MV003	The Dark Knight	2008	Action	02:32:00	English	200000000.00000	LIC002	Francis Ford Coppola	83	50
MV003	The Dark Knight	2008	Action	02:32:00	English	200000000.00000	LIC001	Frank Darabont	62	35
MV004	Pulp Fiction	1994	Crime	02:34:00	English	8000000.00000	LIC005	Robert Zemeckis	70	40
MV004	Pulp Fiction	1994	Crime	02:34:00	English	8000000.00000	LIC004	Quentin Tarantino	58	30
MV004	Pulp Fiction	1994	Crime	02:34:00	English	8000000.00000	LIC003	Christopher Nolan	52	25
MV004	Pulp Fiction	1994	Crime	02:34:00	English	8000000.00000	LIC002	Francis Ford Coppola	83	50
MV004	Pulp Fiction	1994	Crime	02:34:00	English	8000000.00000	LIC001	Frank Darabont	62	35
MV005	Forrest Gump	1994	Drama	02:22:00	English	55000000.00000	LIC005	Robert Zemeckis	70	40
MV005	Forrest Gump	1994	Drama	02:22:00	English	55000000.00000	LIC004	Quentin Tarantino	58	30
MV005	Forrest Gump	1994	Drama	02:22:00	English	55000000.00000	LIC003	Christopher Nolan	52	25
MV005	Forrest Gump	1994	Drama	02:22:00	English	55000000.00000	LIC002	Francis Ford Coppola	83	50
MV006	Inception	2010	Science Fiction	02:28:00	English	160000000.00000	LIC005	Robert Zemeckis	70	40
MV006	Inception	2010	Science Fiction	02:28:00	English	160000000.00000	LIC004	Quentin Tarantino	58	30
MV006	Inception	2010	Science Fiction	02:28:00	English	160000000.00000	LIC003	Christopher Nolan	52	25
MV006	Inception	2010	Science Fiction	02:28:00	English	160000000.00000	LIC002	Francis Ford Coppola	83	50
MV006	Inception	2010	Science Fiction	02:28:00	English	160000000.00000	LIC001	Frank Darabont	62	35

FIGURE 76 : QUERY AND OUTPUT FOR THE CARTESIAN PRODUCT OPERATION

```
466
467 •   CREATE VIEW MovieDetails AS
468     SELECT MOVIE_NAME, GENRE, RELEASED_YEAR FROM MOVIE;
469
470 •   SHOW FULL TABLES IN TVSEREIS_MOVIES_COLLECTION1 WHERE TABLE_TYPE LIKE 'VIEW';
471
472
```

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

Tables_in_tvsereis_movies_collection1	Table_type
moviedetails	VIEW

FIGURE 77 : QUERY, OUTPUT AND CHECK THE VIEW CREATION FOR THE CREATE VIEW OPERATION

```
472 •   SELECT
473     DIRECTOR_NAME AS Director,
474     AGE AS Director_Age
475     FROM DIRECTOR;
476
```

Result Grid | Filter Rows:

Director	Director_Age
Frank Darabont	62
Francis Ford Coppola	83
Christopher Nolan	52
Quentin Tarantino	58
Robert Zemeckis	70

FIGURE 78 : QUERY AND OUTPUT FOR THE RENAMING OPERATION

```
477 •   SELECT
478     AVG(RATING) AS Avg_Rating
479     FROM REVIEW;
480
```

Result Grid | Filter Rows: | Export

Avg_Rating
4.6400000

FIGURE 79 : QUERY AND OUTPUT FOR THE AVERAGE AGGREGATION FUNCTION

```
481 •  SELECT
482    MAX(RELEASED_YEAR)
483    AS Latest_Release
484    FROM MOVIE;
485
```

Result Grid | Filter Rows:

Latest_Release
2010

FIGURE 80 : QUERY AND OUTPUT FOR THE MAXIMUM AGGREGATION FUNCTION

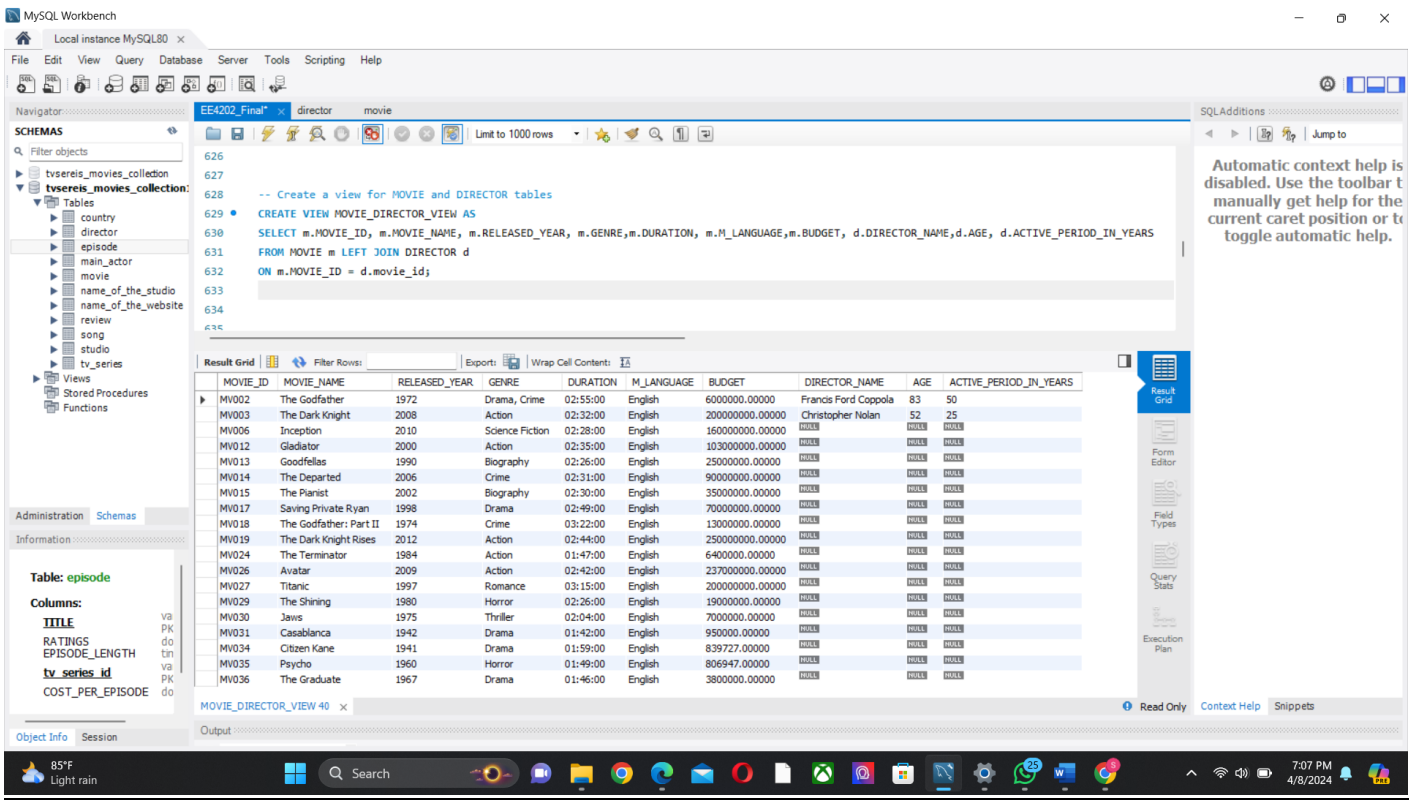
```
485
486 •  SELECT *
487    FROM MOVIE
488    WHERE MOVIE_NAME LIKE '%Godfather%';
489
```

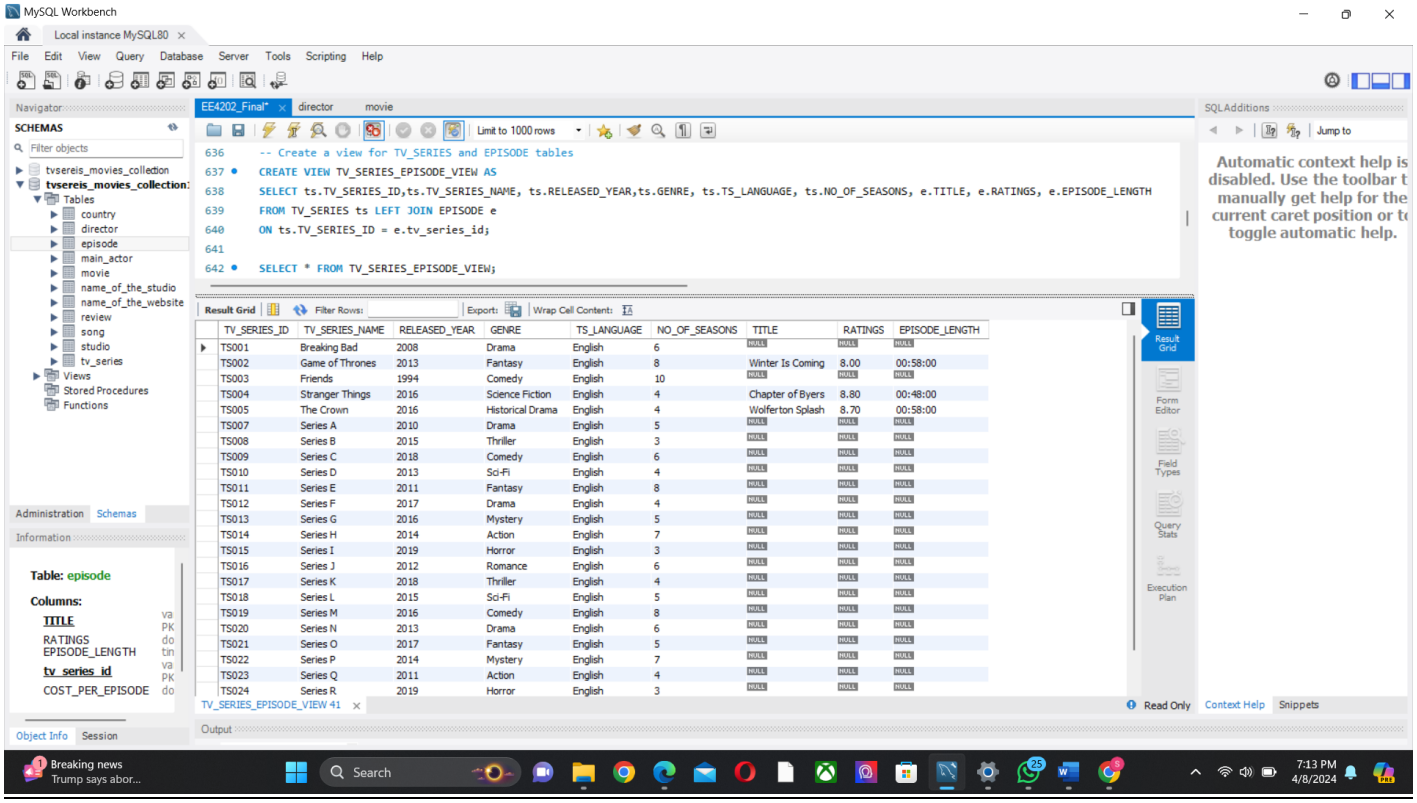
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Conten

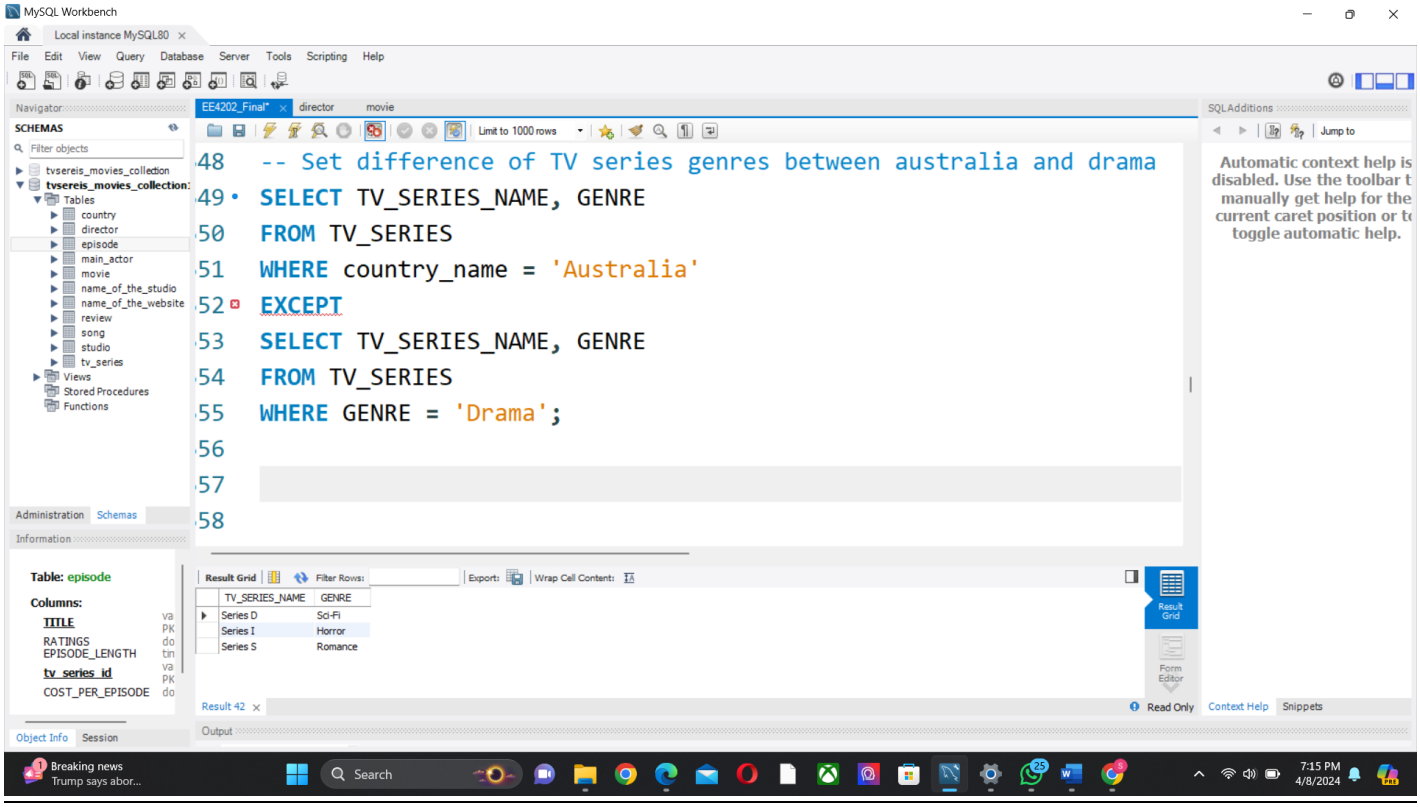
MOVIE_ID	MOVIE_NAME	RELEASED_YEAR	GENRE	DURATION	M_LANGUAGE	BUDGET
MV002	The Godfather	1972	Drama, Crime	02:55:00	English	6000000.00000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

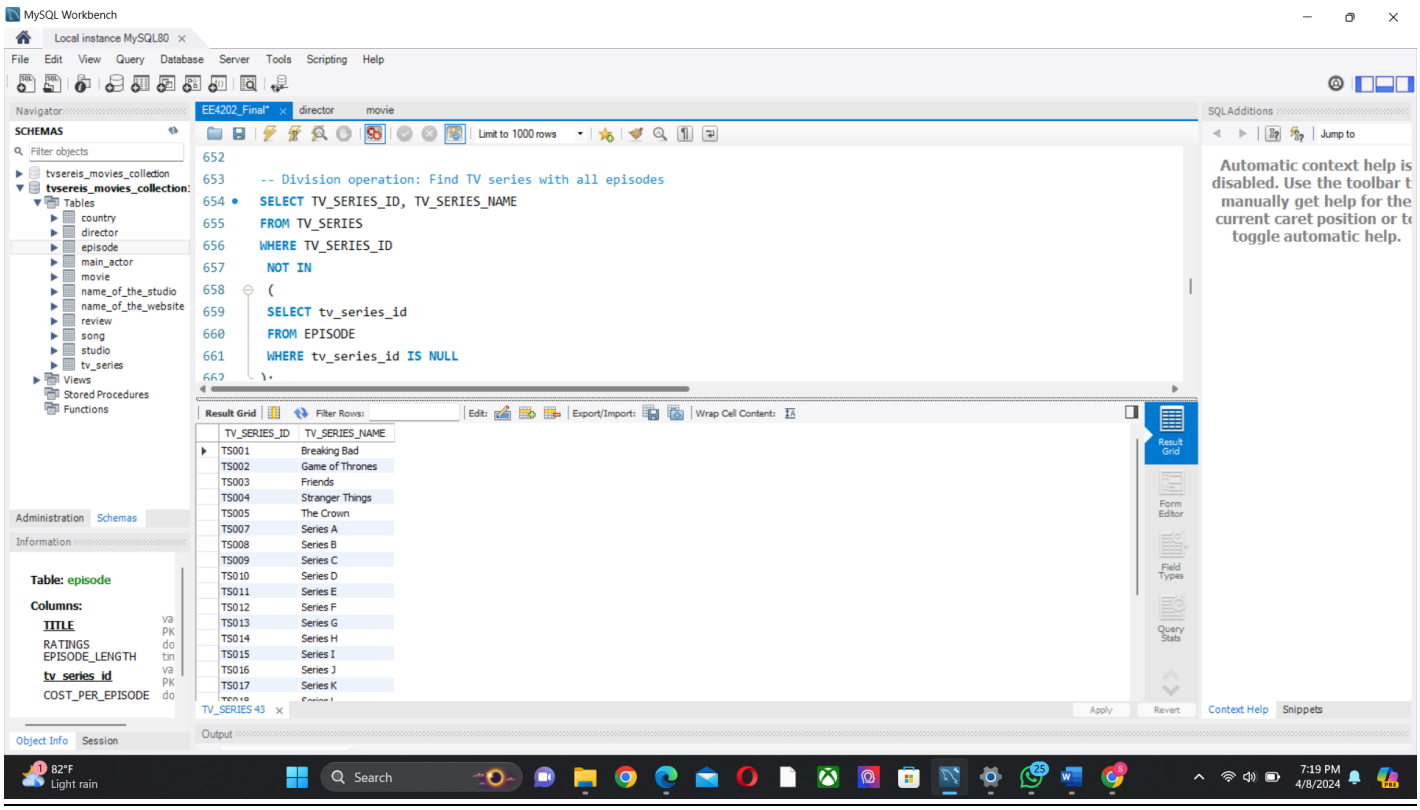
FIGURE 81 : QUERY AND OUTPUT FOR THE LIKE KEYWORD

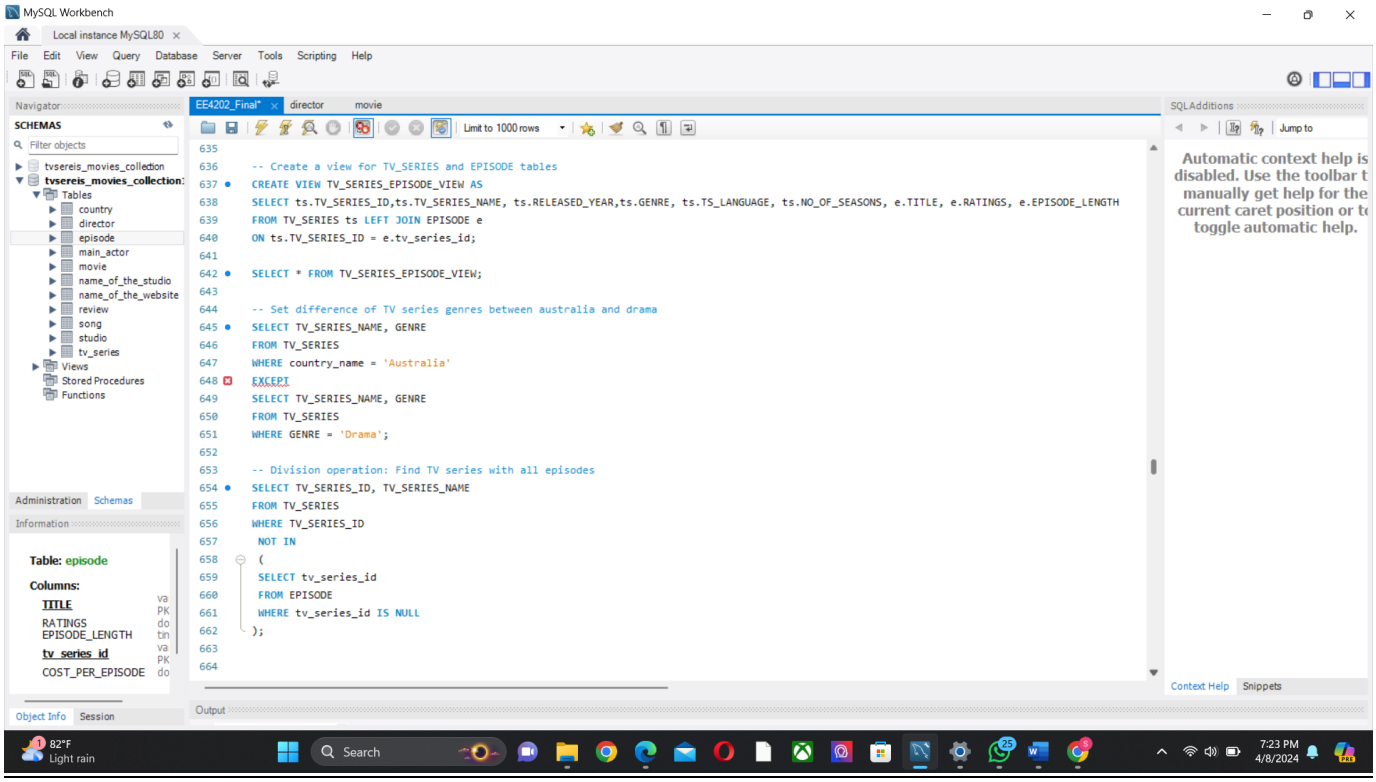
ii. Complex queries

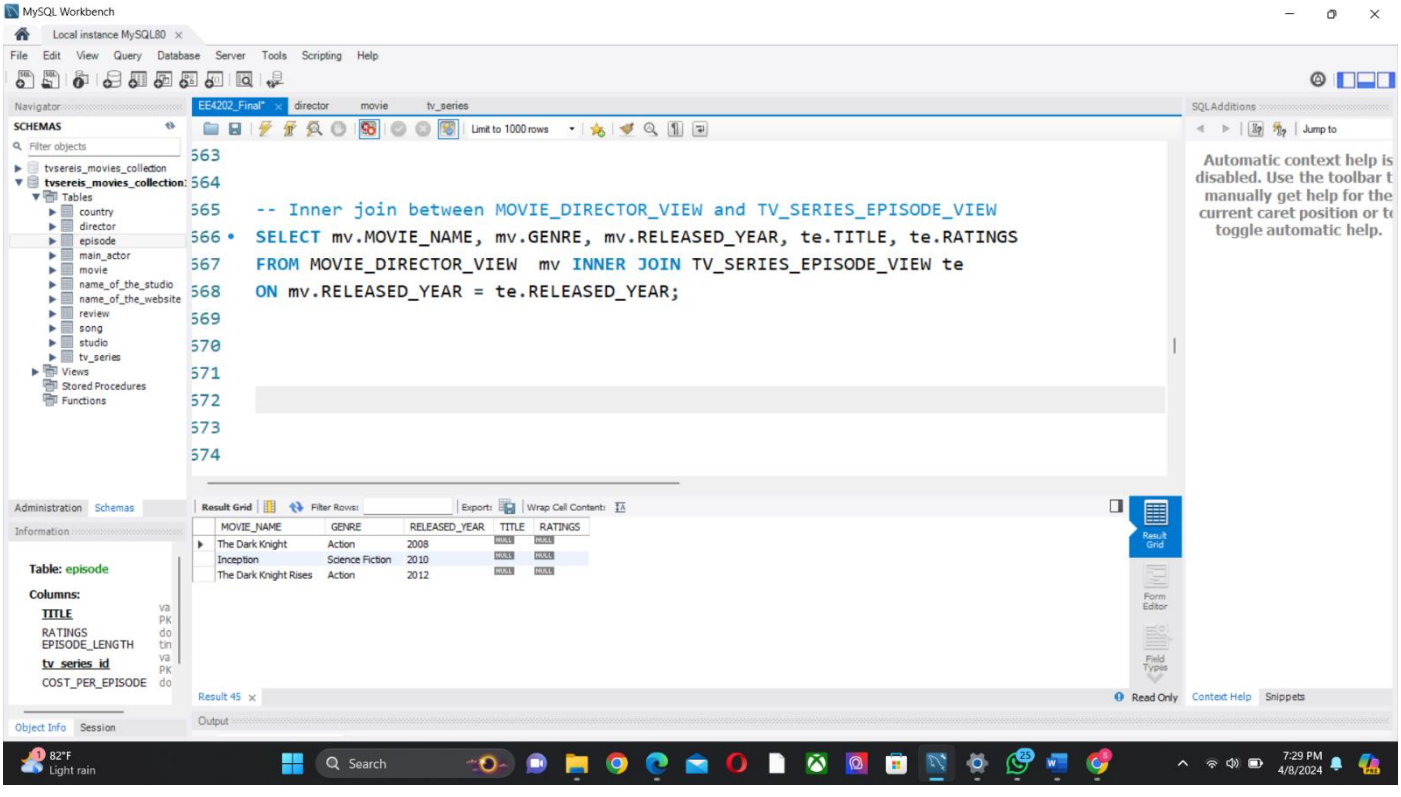


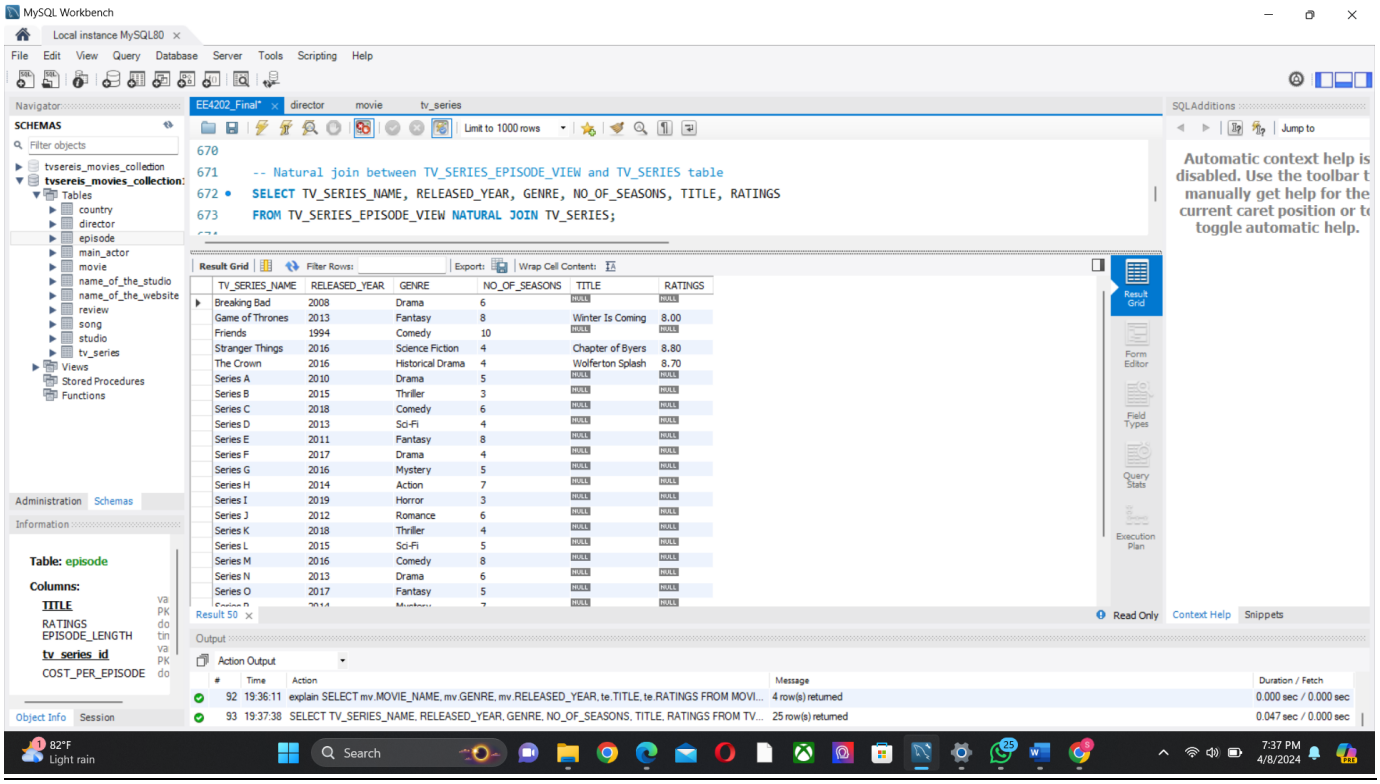


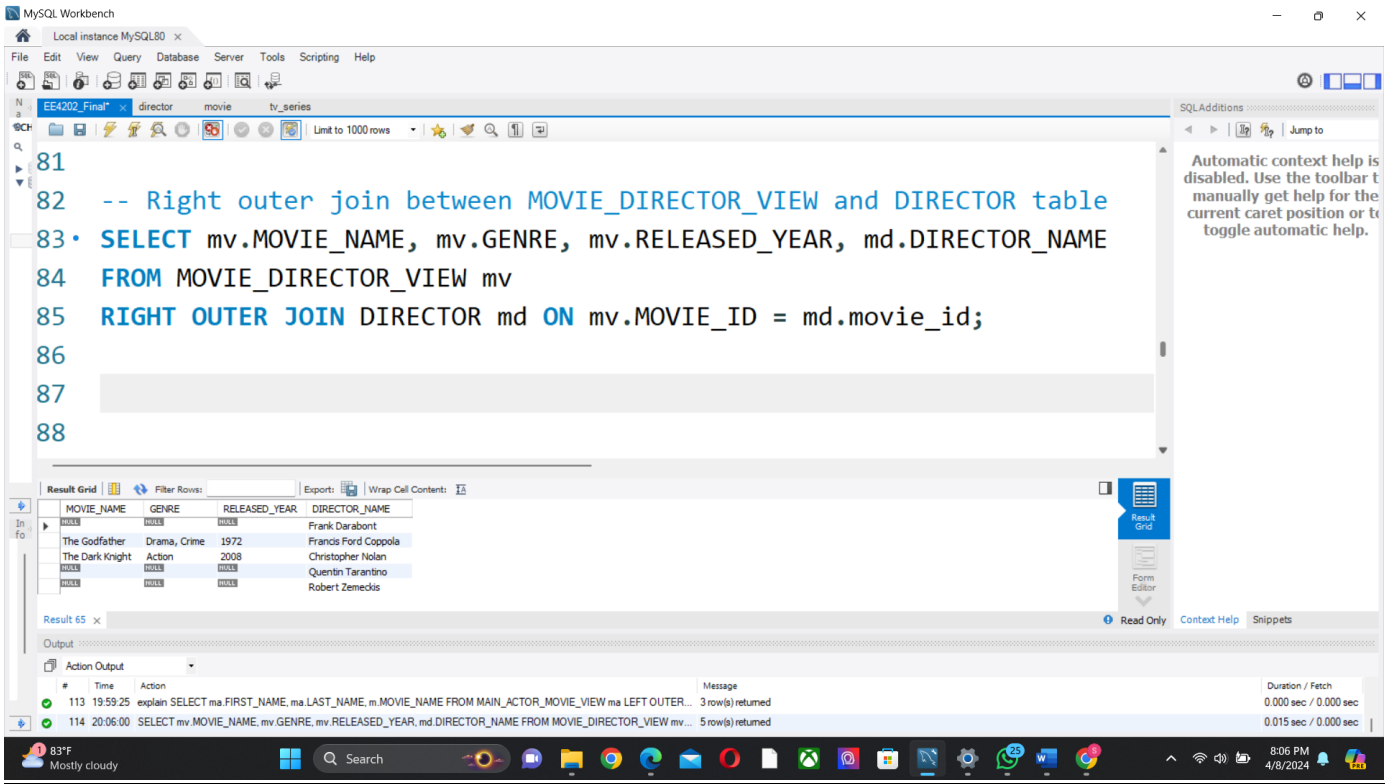

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
tvsereis_movies_collection
Tables
country
director
episode
main_actor
movie
name_of_the_studio
name_of_the_website
review
song
studio
tv_series
Views
Stored Procedures
Functions
EE4202_Final*
director
movie
Limit to 1000 rows
48 -- Set difference of TV series genres between australia and drama
49 SELECT TV_SERIES_NAME, GENRE
50 FROM TV_SERIES
51 WHERE country_name = 'Australia'
52 EXCEPT
53 SELECT TV_SERIES_NAME, GENRE
54 FROM TV_SERIES
55 WHERE GENRE = 'Drama';
56
57
58
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
TV_SERIES_NAME GENRE
Series D Sci-Fi
Series I Horror
Series S Romance
Result 42
Read Only
Context Help
Snippets
Output
Table: episode
Columns:
TITLE
RATINGS
EPISODE_LENGTH
tv_series_id
COST_PER_EPISODE
Breaking news
Trump says abor...
Search
7:15 PM
4/8/2024

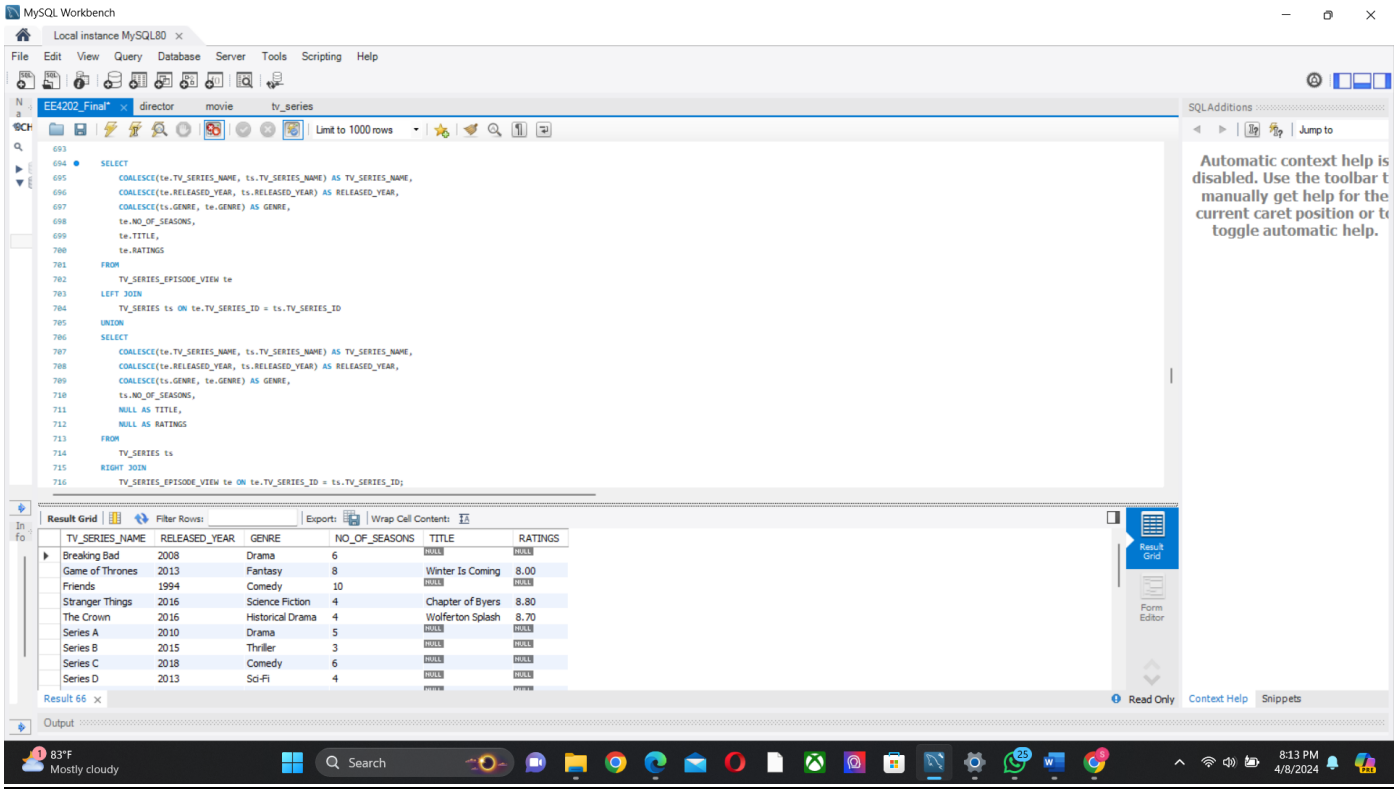

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
tvsereis_movies_collection
Tables
country
director
episode
main_actor
movie
name_of_the_studio
name_of_the_website
review
song
studio
tv_series
Views
Stored Procedures
Functions
EE4202_Final*
director
movie
Limit to 1000 rows
652
653 -- Division operation: Find TV series with all episodes
654 SELECT TV_SERIES_ID, TV_SERIES_NAME
655 FROM TV_SERIES
656 WHERE TV_SERIES_ID
657 NOT IN
658 (
659 SELECT tv_series_id
660 FROM EPISODE
661 WHERE tv_series_id IS NULL
662);
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
TV_SERIES_ID TV_SERIES_NAME
TS001 Breaking Bad
TS002 Game of Thrones
TS003 Friends
TS004 Stranger Things
TS005 The Crown
TS007 Series A
TS008 Series B
TS009 Series C
TS010 Series D
TS011 Series E
TS012 Series F
TS013 Series G
TS014 Series H
TS015 Series I
TS016 Series J
TS017 Series K
TV_SERIES 43
Apply
Revert
Context Help
Snippets
Output
Table: episode
Columns:
TITLE
RATINGS
EPISODE_LENGTH
tv_series_id
COST_PER_EPISODE
82°F
Light rain
Search
7:19 PM
4/8/2024


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
tvsereis_movies_collection
tvsereis_movies_collection:
Tables
country
director
episode
main_actor
movie
name_of_the_studio
name_of_the_website
review
song
studio
tv_series
Views
Stored Procedures
Functions
Administration Schemas
Information
Table: episode
Columns:
TITLE
RATINGS
EPISODE_LENGTH
tv_series_id
COST_PER_EPISODE
Object Info Session
EE4202_Final* director movie
Limit to 1000 rows
635
636 -- Create a view for TV_SERIES and EPISODE tables
637 CREATE VIEW TV_SERIES_EPISODE_VIEW AS
638 SELECT ts.TV_SERIES_ID,ts.TV_SERIES_NAME, ts.RELEASED_YEAR,ts.GENRE, ts.TS_LANGUAGE, ts.NO_OF_SEASONS, e.TITLE, e.RATINGS, e.EPISODE_LENGTH
639 FROM TV_SERIES ts LEFT JOIN EPISODE e
640 ON ts.TV_SERIES_ID = e.tv_series_id;
641
642 SELECT * FROM TV_SERIES_EPISODE_VIEW;
643
644 -- Set difference of TV series genres between australia and drama
645 SELECT TV_SERIES_NAME, GENRE
646 FROM TV_SERIES
647 WHERE country_name = 'Australia'
648 EXCEPT
649 SELECT TV_SERIES_NAME, GENRE
650 FROM TV_SERIES
651 WHERE GENRE = 'Drama';
652
653 -- Division operation: Find TV series with all episodes
654 SELECT TV_SERIES_ID, TV_SERIES_NAME
655 FROM TV_SERIES
656 WHERE TV_SERIES_ID
657 NOT IN
658 (
659 SELECT tv_series_id
660 FROM EPISODE
661 WHERE tv_series_id IS NULL
662);
663
664
Output
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
82°F
Light rain
Search
7:23 PM
4/8/2024


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
tvsereis_movies_collection
tvsereis_movies_collection:
Tables
country
director
episode
main_actor
movie
name_of_the_studio
name_of_the_website
review
song
studio
tv_series
Views
Stored Procedures
Functions
Administration Schemas
Information
Table: episode
Columns:
TITLE
RATINGS
EPISODE_LENGTH
tv_series_id
COST_PER_EPISODE
Object Info Session
EE4202_Final* director movie tv_series
Limit to 1000 rows
563
564
565 -- Inner join between MOVIE_DIRECTOR_VIEW and TV_SERIES_EPISODE_VIEW
566 SELECT mv.MOVIE_NAME, mv.GENRE, mv.RELEASED_YEAR, te.TITLE, te.RATINGS
567 FROM MOVIE_DIRECTOR_VIEW mv INNER JOIN TV_SERIES_EPISODE_VIEW te
568 ON mv.RELEASED_YEAR = te.RELEASED_YEAR;
569
570
571
572
573
574
Result Grid Filter Rows: Export: Wrap Cell Content:
MOVIE_NAME GENRE RELEASED_YEAR TITLE RATINGS
The Dark Knight Action 2008 NULL NULL
Inception Science Fiction 2010 NULL NULL
The Dark Knight Rises Action 2012 NULL NULL
Result 45
Read Only
Output
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.
Context Help Snippets
82°F
Light rain
Search
7:29 PM
4/8/2024


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
EE4202_Final* director movie tv_series
670
671 -- Natural join between TV_SERIES_EPISODE_VIEW and TV_SERIES table
672 • SELECT TV_SERIES_NAME, RELEASED_YEAR, GENRE, NO_OF_SEASONS, TITLE, RATINGS
673 FROM TV_SERIES_EPISODE_VIEW NATURAL JOIN TV_SERIES;
TV_SERIES_NAME RELEASED_YEAR GENRE NO_OF_SEASONS TITLE RATINGS
Breaking Bad 2008 Drama 6
Game of Thrones 2013 Fantasy 8 Winter Is Coming 8.00
Friends 1994 Comedy 10
Stranger Things 2016 Science Fiction 4 Chapter of Byers 8.80
The Crown 2016 Historical Drama 4 Wolferton Splash 8.70
Series A 2010 Drama 5
Series B 2015 Thriller 3
Series C 2018 Comedy 6
Series D 2013 Sci-Fi 4
Series E 2011 Fantasy 8
Series F 2017 Drama 4
Series G 2016 Mystery 5
Series H 2014 Action 7
Series I 2019 Horror 3
Series J 2012 Romance 6
Series K 2018 Thriller 4
Series L 2015 Sci-Fi 5
Series M 2016 Comedy 8
Series N 2013 Drama 6
Series O 2017 Fantasy 5
Table: episode
Columns:
TITLE
RATINGS
EPISODE_LENGTH
tv_series_id
COST_PER_EPISODE
Output
Action Output
92 19:36:11 explain SELECT mv.MOVIE_NAME, mv.GENRE, mv.RELEASED_YEAR, te.TITLE, te.RATINGS FROM MOVI... 4 row(s) returned 0.000 sec / 0.000 sec
93 19:37:38 SELECT TV_SERIES_NAME, RELEASED_YEAR, GENRE, NO_OF_SEASONS, TITLE, RATINGS FROM TV... 25 row(s) returned 0.047 sec / 0.000 sec
82°F Light rain
Search
7:37 PM 4/8/2024


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
EE4202_Final* director movie tv_series
81
82 -- Right outer join between MOVIE_DIRECTOR_VIEW and DIRECTOR table
83 • SELECT mv.MOVIE_NAME, mv.GENRE, mv.RELEASED_YEAR, md.DIRECTOR_NAME
84 FROM MOVIE_DIRECTOR_VIEW mv
85 RIGHT OUTER JOIN DIRECTOR md ON mv.MOVIE_ID = md.movie_id;
86
87
88
MOVIE_NAME GENRE RELEASED_YEAR DIRECTOR_NAME
Frank Darabont
The Godfather Drama, Crime 1972 Francis Ford Coppola
The Dark Knight Action 2008 Christopher Nolan
Quentin Tarantino
Robert Zemeckis
Output
Action Output
113 19:59:25 explain SELECT ma.FIRST_NAME, ma.LAST_NAME, m.MOVIE_NAME FROM MAIN_ACTOR_MOVIE_VIEW ma LEFT OUTER... 3 row(s) returned 0.000 sec / 0.000 sec
114 20:06:00 SELECT mv.MOVIE_NAME, mv.GENRE, mv.RELEASED_YEAR, md.DIRECTOR_NAME FROM MOVIE_DIRECTOR_VIEW mv... 5 row(s) returned 0.015 sec / 0.000 sec
83°F Mostly cloudy
Search
8:06 PM 4/8/2024




```sql
-- Create a view for MOVIE and DIRECTOR tables
CREATE VIEW MOVIE_DIRECTOR_VIEW AS
SELECT m.MOVIE_ID, m.MOVIE_NAME, m.RELEASED_YEAR, m.GENRE, m.DURATION, m.M_LANGUAGE, m.BUDGET, d.DIRECTOR_NAME, d.AGE, d.ACTIVE_PERIOD_IN_YEARS
FROM MOVIE m
LEFT JOIN DIRECTOR d ON m.MOVIE_ID = d.movie_id;

-- Create a view for TV_SERIES and EPISODE tables
CREATE VIEW TV_SERIES_EPISODE_VIEW AS
SELECT ts.TV_SERIES_ID, ts.TV_SERIES_NAME, ts.RELEASED_YEAR, ts.GENRE, ts.TS_LANGUAGE, ts.NO_OF_SEASONS, e.TITLE, e.RATINGS, e.EPISODE_LENGTH
FROM TV_SERIES ts
LEFT JOIN EPISODE e ON ts.TV_SERIES_ID = e.tv_series_id;

-- Create a view for MAIN_ACTOR and MOVIE tables
CREATE VIEW MAIN_ACTOR_MOVIE_VIEW AS
SELECT ma.ACTOR_LICENSE_NO, ma.FIRST_NAME, ma.LAST_NAME, ma.NATIONALITTY, ma.COUNTRY, ma.DATE_OF_BIRTH, ma.ACTIVE_PERIOD, ma.AWARDS, m.MOVIE_NAME
FROM MAIN_ACTOR ma
LEFT JOIN MOVIE m ON ma.movie_id = m.MOVIE_ID;

SHOW FULL TABLES IN TVSEREIS_MOVIES_COLLECTION1 WHERE TABLE_TYPE LIKE 'VIEW';
```

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

Tables_in_tvsereis_movies_collection1	Table_type
main_actor_movie_view	VIEW
movie_director_view	VIEW
moviedetails	VIEW
tv_series_episode_view	VIEW

FIGURE 82 : COMPLEX QUERY AND OUTPUT FOR THE CREATING VIEW AND CHECK THE VIEWS

```
509
510     -- Union of two sets of TV series from different countries
511     SELECT TV_SERIES_NAME, RELEASED_YEAR, GENRE, TS_LANGUAGE
512     FROM TV_SERIES
513     WHERE country_name = 'USA'
514     UNION
515     SELECT TV_SERIES_NAME, RELEASED_YEAR, GENRE, TS_LANGUAGE
516     FROM TV_SERIES
517     WHERE country_name = 'UK';
518
```

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

TV_SERIES_NAME	RELEASED_YEAR	GENRE	TS_LANGUAGE
Breaking Bad	2008	Drama	English
Game of Thrones	2013	Fantasy	English
Friends	1994	Comedy	English
Stranger Things	2016	Science Fiction	English
The Crown	2016	Historical Drama	English

FIGURE 83 : COMPLEX QUERY AND OUTPUT FOR SET OPERATION UNION

```
519     -- Intersection of TV series genres between two different countries
520     SELECT TV_SERIES_NAME, GENRE
521     FROM TV_SERIES
522     WHERE country_name = 'USA'
523     INTERSECT
524     SELECT TV_SERIES_NAME, GENRE
525     FROM TV_SERIES
526     WHERE country_name = 'UK';
527
```

It is showing that INTERSECT is not valid for this point

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

TV_SERIES_NAME	GENRE

FIGURE 84 : COMPLEX QUERY AND OUTPUT FOR SET OPERATION INTERSECT

```
528     -- Set difference of TV series genres between USA and UK
529     SELECT TV_SERIES_NAME, GENRE
530     FROM TV_SERIES
531     WHERE country_name = 'USA'
532     EXCEPT
533     SELECT TV_SERIES_NAME, GENRE
534     FROM TV_SERIES
535     WHERE country_name = 'UK';
536
```

It is showing that EXCEPT is not valid for this point

Result Grid | Filter Rows: | Export: | Wrap Cel

TV_SERIES_NAME	GENRE
Breaking Bad	Drama
Game of Thrones	Fantasy
Friends	Comedy
Stranger Things	Science Fiction

FIGURE 85 : COMPLEX QUERY AND OUTPUT FOR SET OPERATION EXCEPT

```
536
537     -- Division operation: Find TV series with all episodes
538     SELECT TV_SERIES_ID, TV_SERIES_NAME
539     FROM TV_SERIES
540     WHERE TV_SERIES_ID NOT IN (
541         SELECT tv_series_id
542         FROM EPISODE
543         WHERE tv_series_id IS NULL
544     );
545
546
547
```

Result Grid | Filter Rows: | Edit:

TV_SERIES_ID	TV_SERIES_NAME
TS001	Breaking Bad
TS002	Game of Thrones
TS003	Friends
TS004	Stranger Things
TS005	The Crown
NULL	NULL

FIGURE 86: COMPLEX QUERY AND OUTPUT FOR DIVISION OPERATION

```
546      -- Nested query with set difference
547      SELECT TV_SERIES_NAME
548      FROM TV_SERIES
549      WHERE TV_SERIES_ID IN (
550          SELECT TV_SERIES_ID
551          FROM EPISODE
552          WHERE RATINGS > 9.0
553      );
554
```

Result Grid | Filter Rows: | Export:

TV_SERIES_NAME

FIGURE 87: COMPLEX QUERY AND OUTPUT FOR NESTED QUERY WITH SET DIFFERENT

```
555      -- Multiset comparison with correlated nested query
556      SELECT MOVIE_NAME
557      FROM MOVIE m
558      WHERE EXISTS (
559          SELECT 1
560          FROM DIRECTOR d
561          WHERE d.movie_id = m.MOVIE_ID AND d.ACTIVE_PERIOD_IN_YEARS > 30
562      );
563
```

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

MOVIE_NAME
The Shawshank Redemption
The Godfather
Forrest Gump

FIGURE 88 : COMPLEX QUERY AND OUTPUT FOR MULTISET COMPARISON WITH CORRELATED NESTED QUERY

```
564     -- Correlated nested query with set difference
565     SELECT TV_SERIES_NAME
566     FROM TV_SERIES ts
567     WHERE ts.TV_SERIES_ID NOT IN (
568         SELECT tv_series_id
569         FROM EPISODE e
570         WHERE e.RATINGS > (
571             SELECT AVG(RATINGS)
572             FROM EPISODE
573         )
574     );
575
```

Result Grid | Filter Rows: | Export:

TV_SERIES_NAME
Breaking Bad
Game of Thrones
Friends

FIGURE 89 : COMPLEX QUERY AND OUTPUT FOR CORRELATED NESTED QUERY WITH SET DIFFERENCE

```
576     -- Recursive query to find all TV series and their episodes
577     WITH RECURSIVE TVSeries_CTE AS (
578         SELECT TV_SERIES_ID, TV_SERIES_NAME
579         FROM TV_SERIES
580         UNION ALL
581         SELECT e.tv_series_id, e.TITLE
582         FROM TVSeries_CTE t
583         JOIN EPISODE e ON t.TV_SERIES_ID = e.tv_series_id
584     )
585     SELECT *
586     FROM TVSeries_CTE;
```

FIGURE 90 : COMPLEX QUERY FOR RECURSION

```
546     -- Inner join between MOVIE_DIRECTOR_VIEW and TV_SERIES_EPISODE_VIEW
547     SELECT mv.MOVIE_NAME, mv.GENRE, mv.RELEASED_YEAR, te.TITLE, te.RATINGS
548     FROM MOVIE_DIRECTOR_VIEW  mv
549     INNER JOIN TV_SERIES_EPISODE_VIEW te ON mv.MOVIE_ID = te.TV_SERIES_ID;
550
```

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

MOVIE_NAME	GENRE	RELEASED_YEAR	TITLE	RATINGS

FIGURE 91 : COMPLEX QUERY FOR INNER JOIN

```
550
551     -- Natural join between TV_SERIES_EPISODE_VIEW and TV_SERIES table
552     SELECT TV_SERIES_NAME, RELEASED_YEAR, GENRE, NO_OF_SEASONS, TITLE, RATINGS
553     FROM TV_SERIES_EPISODE_VIEW
554     NATURAL JOIN TV_SERIES;
```

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

TV_SERIES_NAME	RELEASED_YEAR	GENRE	NO_OF_SEASONS	TITLE	RATINGS
Breaking Bad	2008	Drama	6	NULL	NULL
Game of Thrones	2013	Fantasy	8	Winter Is Coming	8.00
Friends	1994	Comedy	10	NULL	NULL
Stranger Things	2016	Science Fiction	4	Chapter of Byers	8.80
The Crown	2016	Historical Drama	4	Wolferton Splash	8.70

Result 25

FIGURE 92 : COMPLEX QUERY FOR NATURAL JOIN

```
555
556     -- Left outer join between MAIN_ACTOR_MOVIE_VIEW and MOVIE table
557     SELECT ma.FIRST_NAME, ma.LAST_NAME, m.MOVIE_NAME
558     FROM MAIN_ACTOR_MOVIE_VIEW ma
559     LEFT OUTER JOIN MOVIE m ON ma.MOVIE_NAME = m.MOVIE_NAME;
```

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

FIRST_NAME	LAST_NAME	MOVIE_NAME
Morgan	Freeman	The Shawshank Redemption
Marlon	Brando	The Godfather
Christian	Bale	The Dark Knight
Tom	Hanks	Forrest Gump
Leonardo	DiCaprio	Inception

FIGURE 93 : COMPLEX QUERY FOR LEFT OUTER JOIN

```
560
561     -- Right outer join between MOVIE_DIRECTOR_VIEW and DIRECTOR table
562     SELECT mv.MOVIE_NAME, mv.GENRE, mv.RELEASED_YEAR, md.DIRECTOR_NAME
563     FROM MOVIE_DIRECTOR_VIEW mv
564     RIGHT OUTER JOIN DIRECTOR md ON mv.MOVIE_ID = md.movie_id;
```

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

MOVIE_NAME	GENRE	RELEASED_YEAR	DIRECTOR_NAME
The Shawshank Redemption	Drama	1994	Frank Darabont
The Godfather	Drama, Crime	1972	Francis Ford Coppola
The Dark Knight	Action	2008	Christopher Nolan
Pulp Fiction	Crime	1994	Quentin Tarantino
Forrest Gump	Drama	1994	Robert Zemeckis

FIGURE 94 : COMPLEX QUERY FOR RIGHT OUTER JOIN

```
566     -- Full outer join between TV_SERIES_EPISODE_VIEW and TV_SERIES table
567     SELECT COALESCE(te.TV_SERIES_NAME, ts.TV_SERIES_NAME) AS TV_SERIES_NAME, te.RELEASED_YEAR, ts.GENRE, te.NO_OF_SEASONS, te.TITLE, te.RATINGS
568     FROM TV_SERIES_EPISODE_VIEW te
569     FULL OUTER JOIN TV_SERIES ts ON te.TV_SERIES_ID = ts.TV_SERIES_ID;
```

But it's not supporting

FIGURE 96 : COMPLEX QUERY FOR FULL OUTER JOIN

```
571 ●   SELECT
572         COALESCE(te.TV_SERIES_NAME, ts.TV_SERIES_NAME) AS TV_SERIES_NAME,
573         COALESCE(te.RELEASED_YEAR, ts.RELEASED_YEAR) AS RELEASED_YEAR,
574         COALESCE(ts.GENRE, te.GENRE) AS GENRE,
575         te.NO_OF_SEASONS,
576         te.TITLE,
577         te.RATINGS
578     FROM
579         TV_SERIES_EPISODE_VIEW te
580     LEFT JOIN
581         TV_SERIES ts ON te.TV_SERIES_ID = ts.TV_SERIES_ID
582     UNION
583     SELECT
584         COALESCE(te.TV_SERIES_NAME, ts.TV_SERIES_NAME) AS TV_SERIES_NAME,
585         COALESCE(te.RELEASED_YEAR, ts.RELEASED_YEAR) AS RELEASED_YEAR,
586         COALESCE(ts.GENRE, te.GENRE) AS GENRE,
587         ts.NO_OF_SEASONS,
588         NULL AS TITLE,
589         NULL AS RATINGS
590     FROM
591         TV_SERIES ts
592     RIGHT JOIN
593         TV_SERIES_EPISODE_VIEW te ON te.TV_SERIES_ID = ts.TV_SERIES_ID;
594
```

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

TV_SERIES_NAME	RELEASED_YEAR	GENRE	NO_OF_SEASONS	TITLE	RATINGS
Breaking Bad	2008	Drama	6	NULL	NULL
Game of Thrones	2013	Fantasy	8	Winter Is Coming	8.00
Friends	1994	Comedy	10	NULL	NULL
Stranger Things	2016	Science Fiction	4	Chapter of Byers	8.80
The Crown	2016	Historical Drama	4	Wolferton Splash	8.70
Game of Thrones	2013	Fantasy	8	NULL	NULL
Stranger Things	2016	Science Fiction	4	NULL	NULL
The Crown	2016	Historical Drama	4	NULL	NULL

FIGURE 97 : COMPLEX QUERY FOR UPDATED FULL OUTER JOIN

```
596     -- Outer union between MOVIE_DIRECTOR_VIEW and DIRECTOR table
597     SELECT mv.MOVIE_NAME, mv.GENRE, mv.RELEASED_YEAR, md.DIRECTOR_NAME
598     FROM MOVIE_DIRECTOR_VIEW mv
599     OUTER UNION
600     SELECT NULL, NULL, NULL, DIRECTOR_NAME
601     FROM DIRECTOR;
602
603     SELECT mv.MOVIE_NAME, mv.GENRE, mv.RELEASED_YEAR, mv.DIRECTOR_NAME
604     FROM MOVIE_DIRECTOR_VIEW mv
605     UNION
606     SELECT NULL, NULL, NULL, d.DIRECTOR_NAME
607     FROM DIRECTOR d
608     LEFT JOIN MOVIE_DIRECTOR_VIEW mv ON d.DIRECTOR_NAME = mv.DIRECTOR_NAME;
609
```

THIS IS NOT SUPPORTING

UPDATED ONE

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

MOVIE_NAME	GENRE	RELEASED_YEAR	DIRECTOR_NAME
The Shawshank Redemption	Drama	1994	Frank Darabont
The Godfather	Drama, Crime	1972	Francis Ford Coppola
The Dark Knight	Action	2008	Christopher Nolan
Pulp Fiction	Crime	1994	Quentin Tarantino
Forrest Gump	Drama	1994	Robert Zemeckis
Inception	Science Fiction	2010	NULL
NULL	NULL	NULL	Frank Darabont
NULL	NULL	NULL	Francis Ford Coppola
NULL	NULL	NULL	Christopher Nolan
NULL	NULL	NULL	Quentin Tarantino
NULL	NULL	NULL	Robert Zemeckis

FIGURE 98 : COMPLEX QUERY FOR OUTER UNION (UPDATED AND NON-SUPPORTED FORMAT)

iii. Comparison of complexity of the queries before and after inserting the index

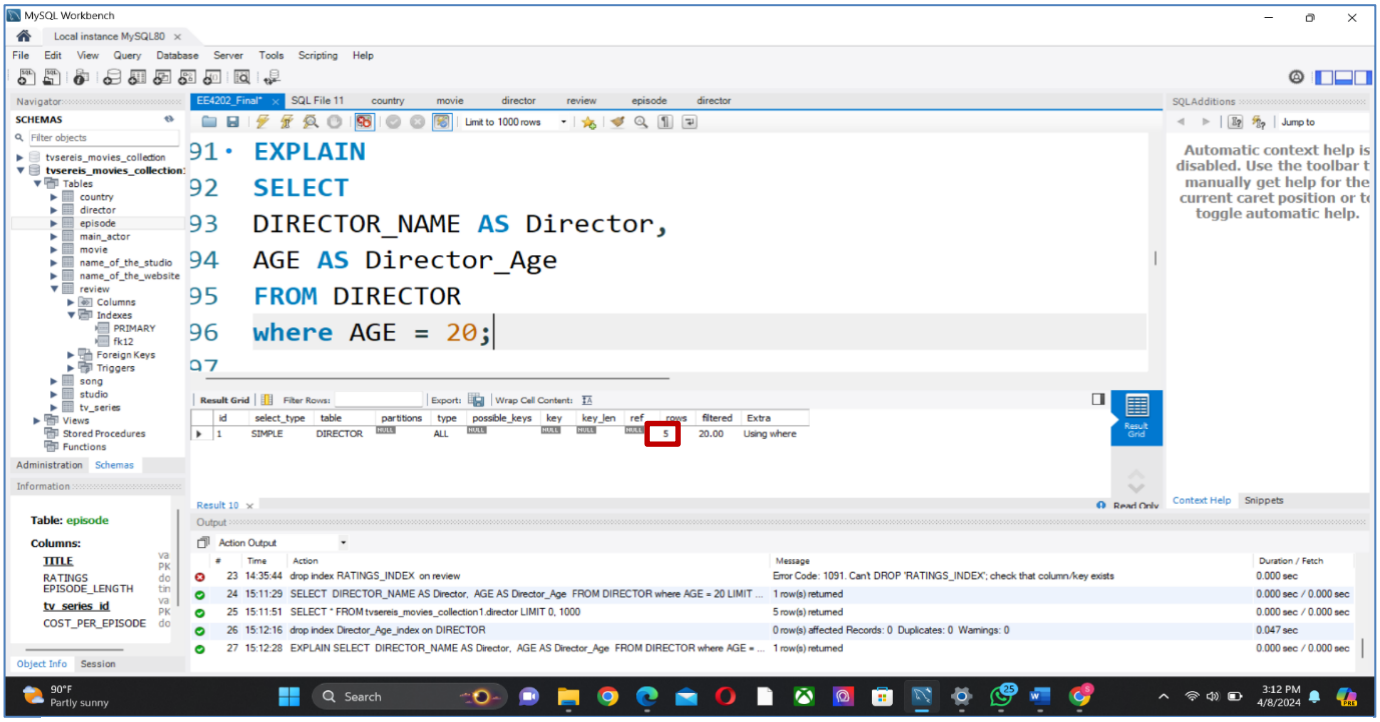


FIGURE 100 : COMPLEXITY OF THE QUERY BEFORE ADDING THE DIERCTOR_AGE_INDEX

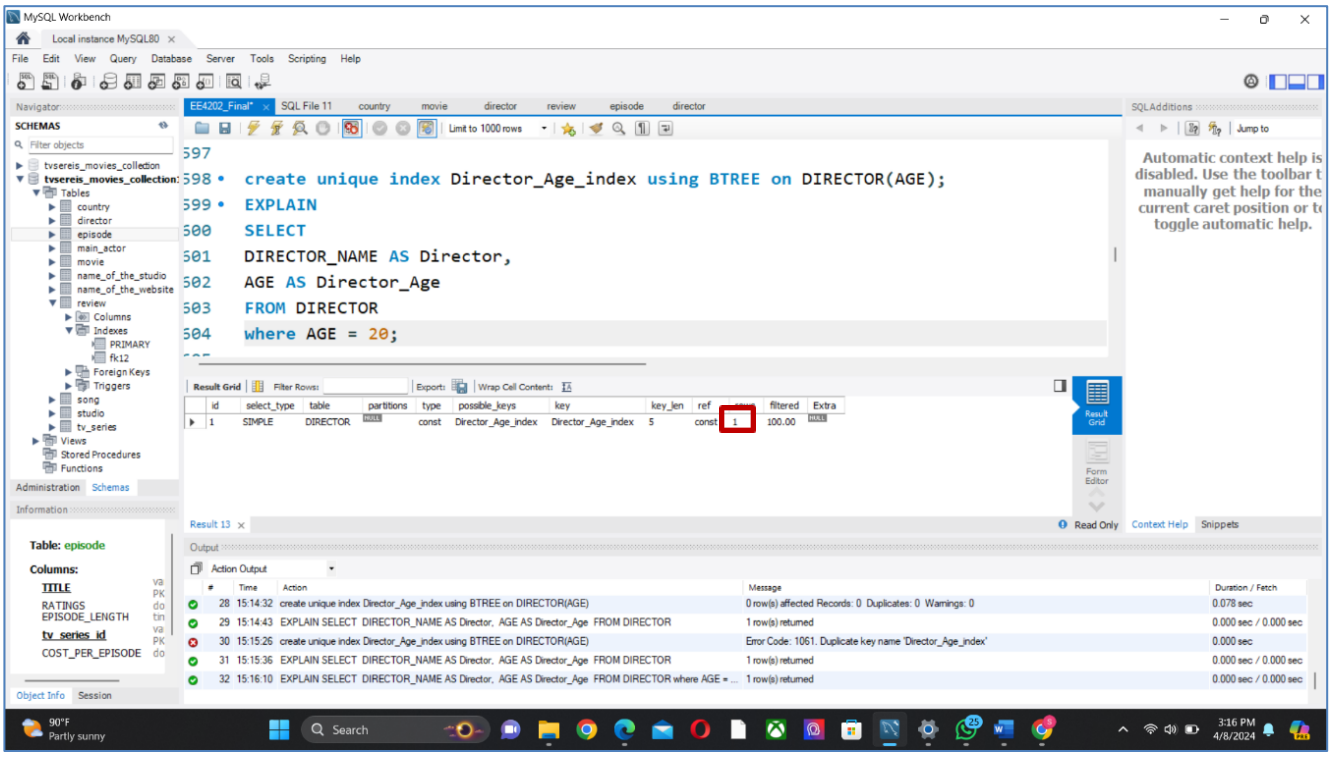


FIGURE 99 : COMPLEXITY OF THE QUERY AFTER ADDING THE DIERCTOR_AGE_INDEX

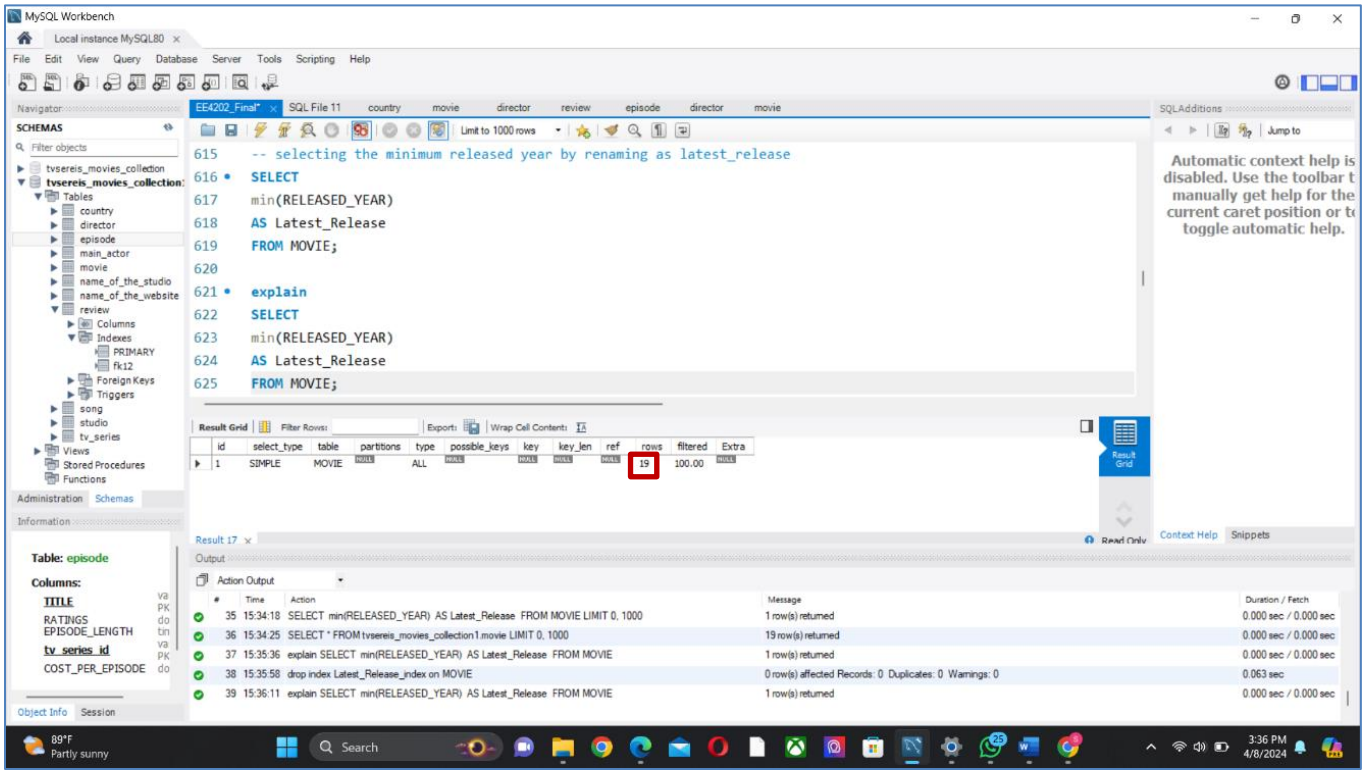


FIGURE 101 : COMPLEXITY OF THE QUERY BEFORE ADDING THE LATEST_RELEASE_INDEX

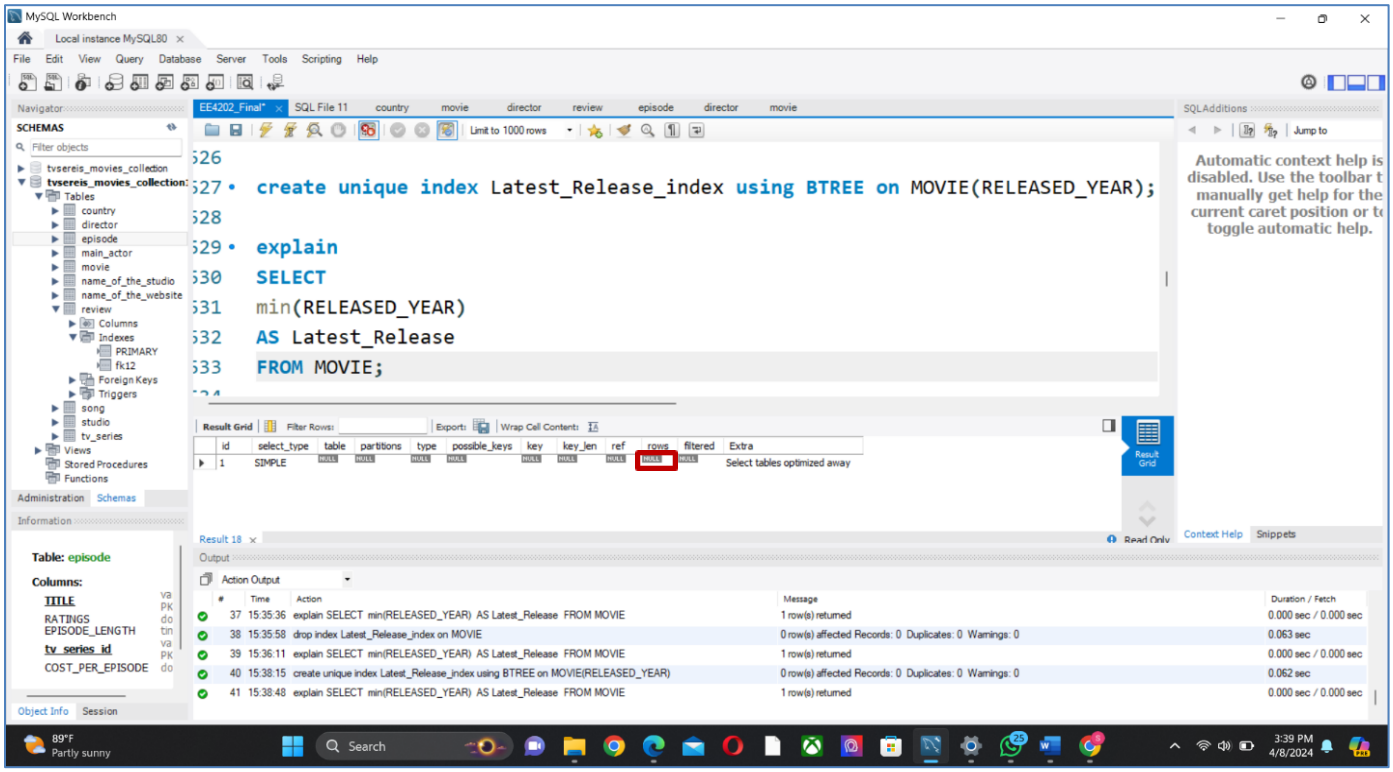


FIGURE 102 : COMPLEXITY OF THE QUERY AFTER ADDING THE LATEST_RELEASE_INDEX

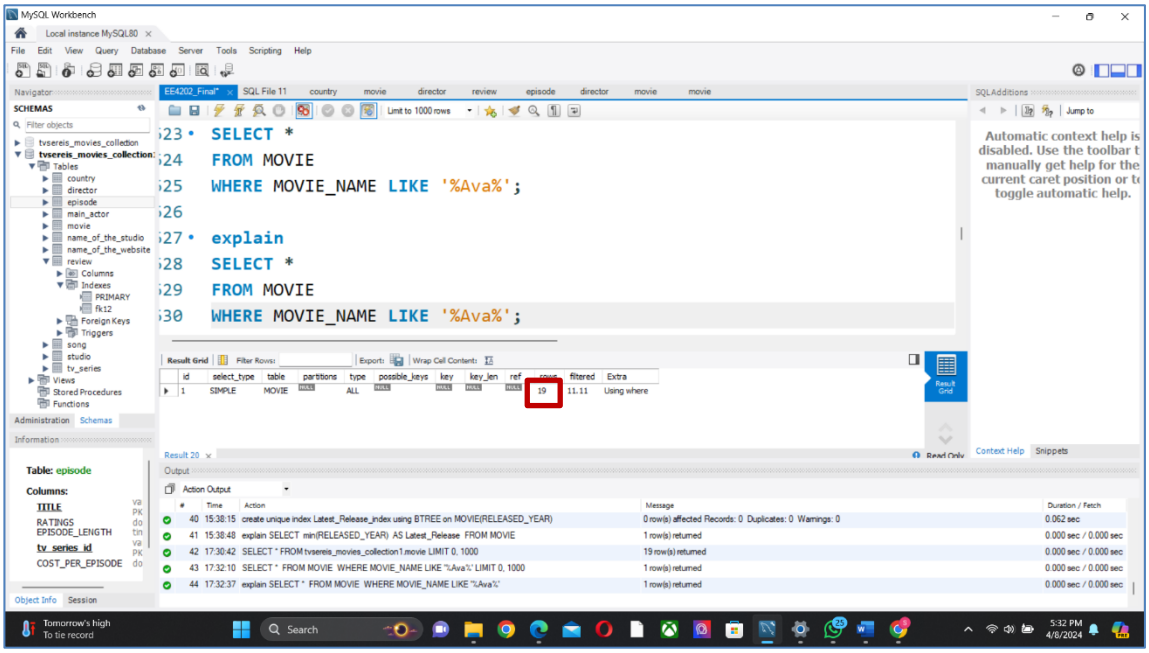


FIGURE 103 : COMPLEXITY OF THE QUERY BEFORE ADDING THE MOVIE_NAME_INDEX

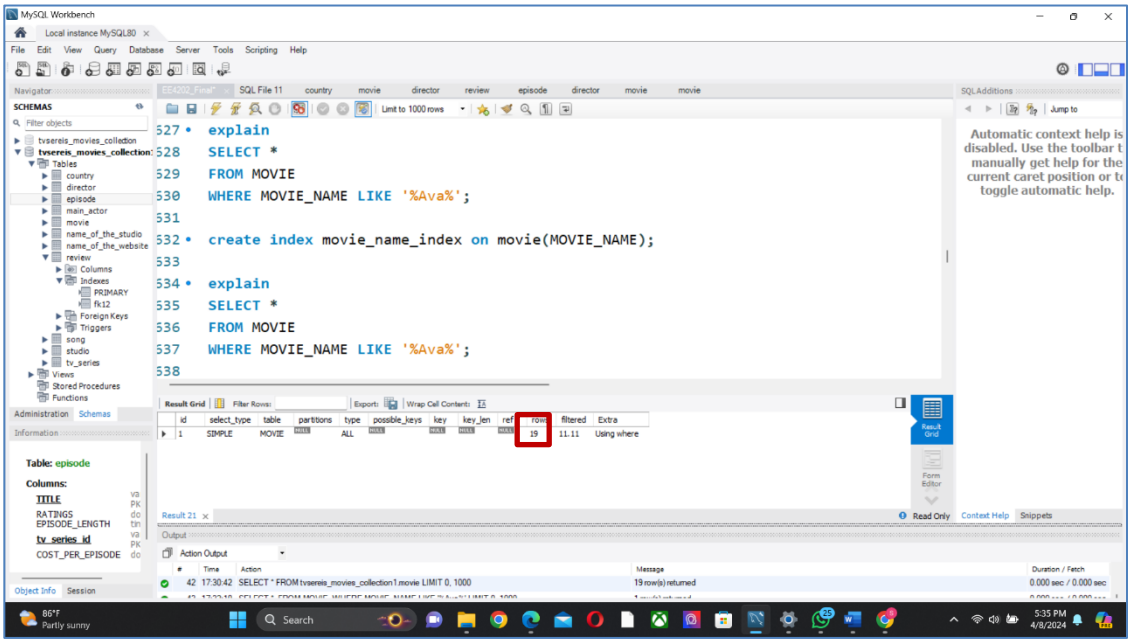


FIGURE 104 : COMPLEXITY OF THE QUERY AFTER ADDING THE MOVIE_NAME_INDEX

According the rule we cannot use indexing for the like operator because it will search the all the rows even there is a index. That's why here there is no any change in the complexity of the queries.

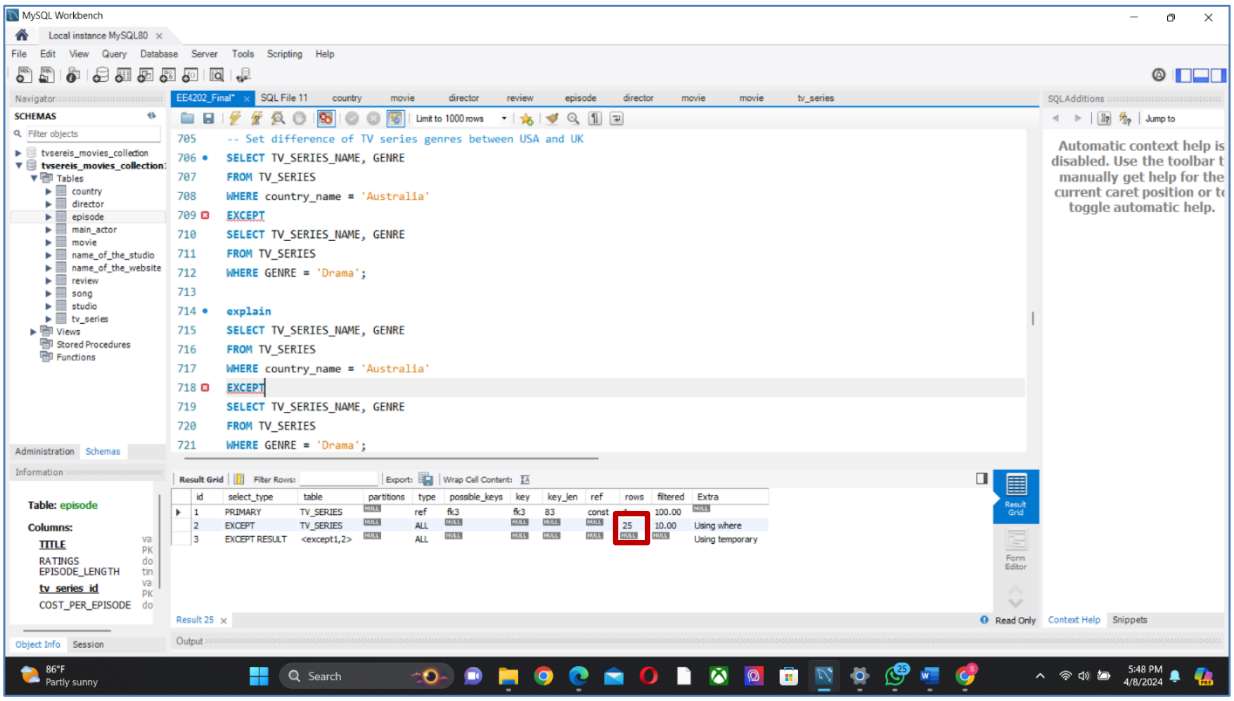


FIGURE 105 : COMPLEXITY OF THE QUERY BEFORE ADDING THE COUNTRY_NAME_INDEX

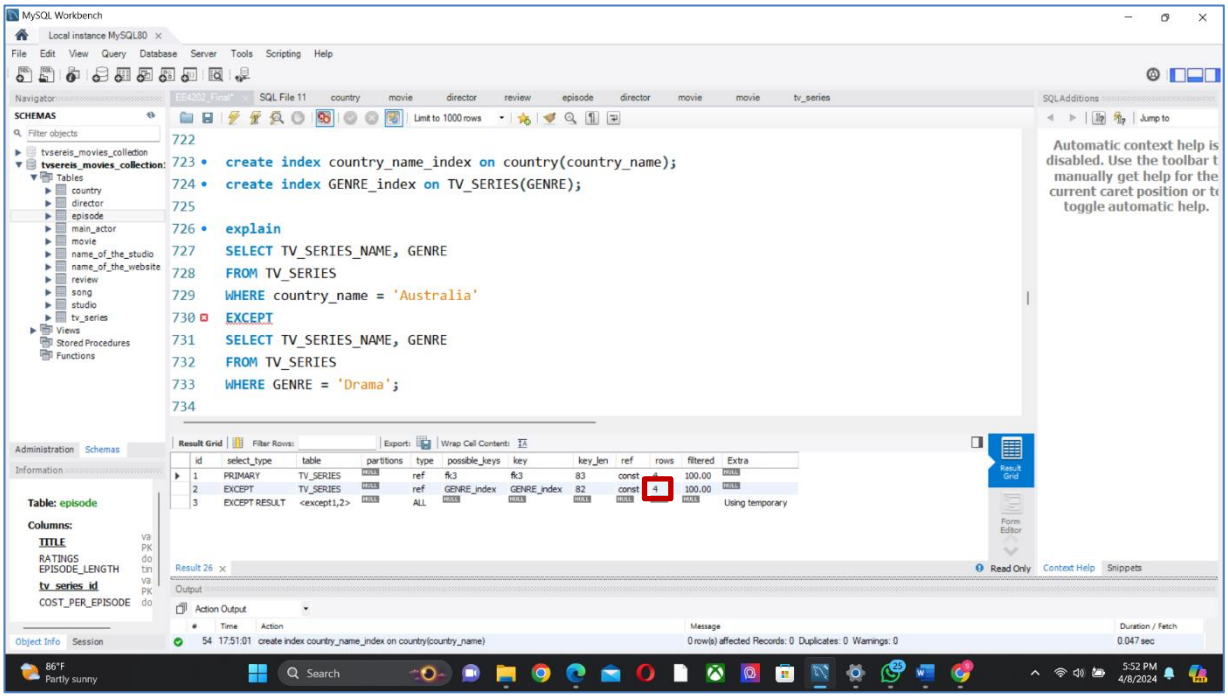


FIGURE 106 : COMPLEXITY OF THE QUERY AFTER ADDING THE COUNTRY_NAME_INDEX

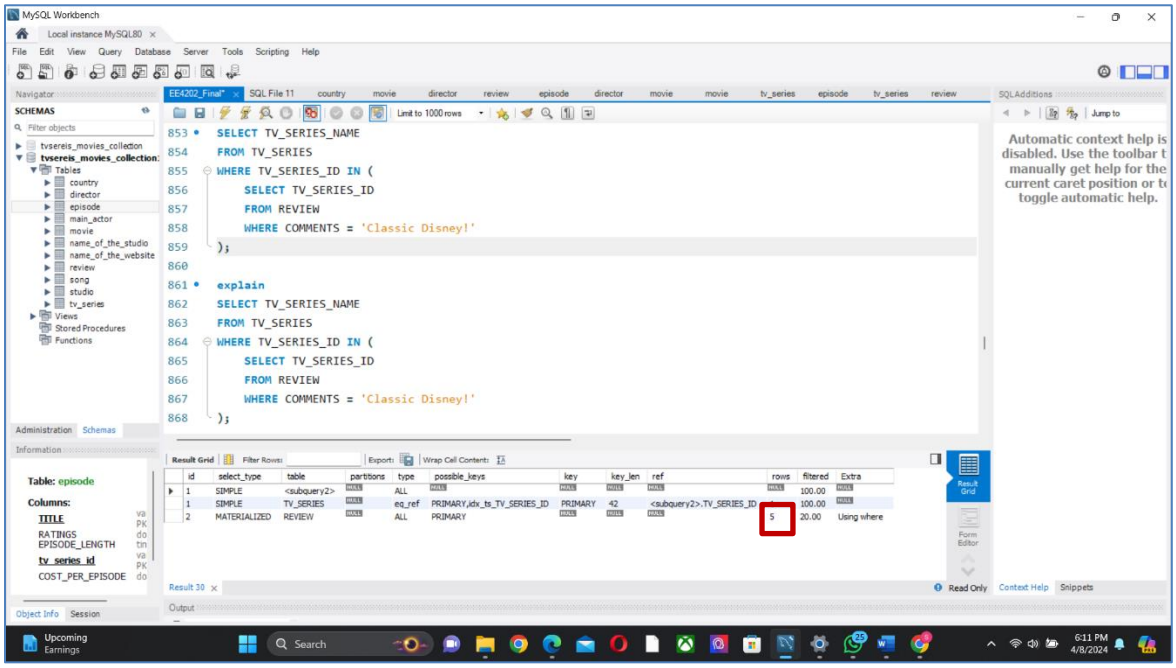


FIGURE 107 : COMPLEXITY OF THE QUERY BEFORE ADDING THE COMMENTS_INDEX

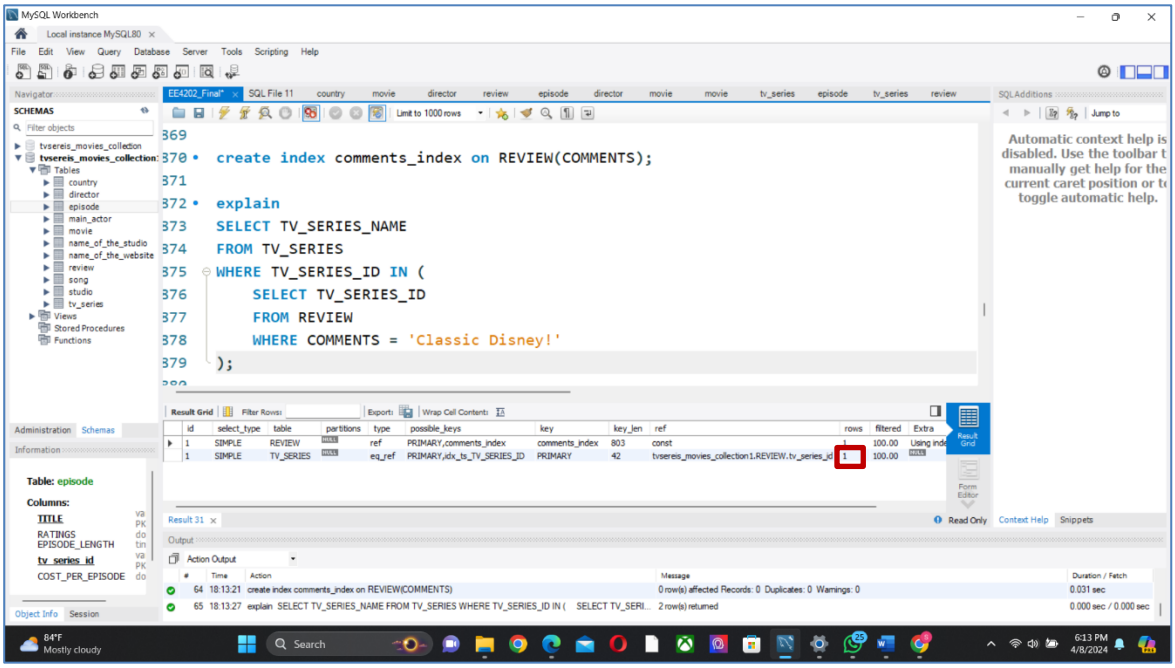


FIGURE 108 : COMPLEXITY OF THE QUERY AFTER ADDING THE COMMENTS_INDEX

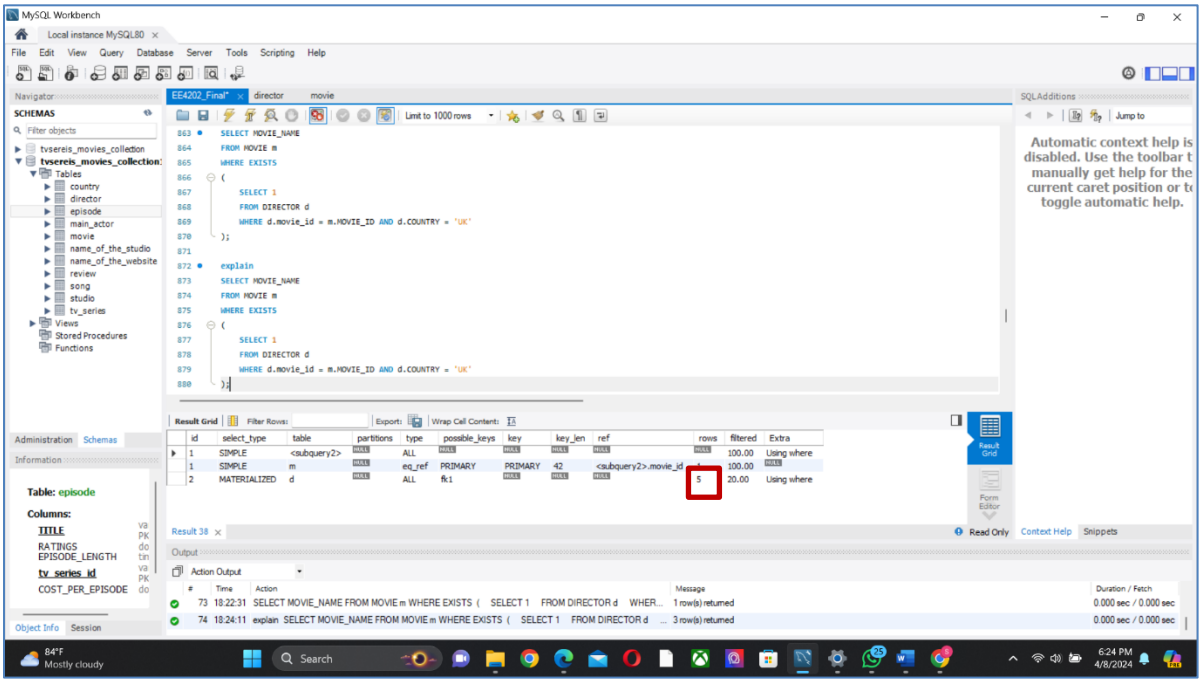


FIGURE 109 : COMPLEXITY OF THE QUERY BEFORE ADDING THE COUNTRY_NAME

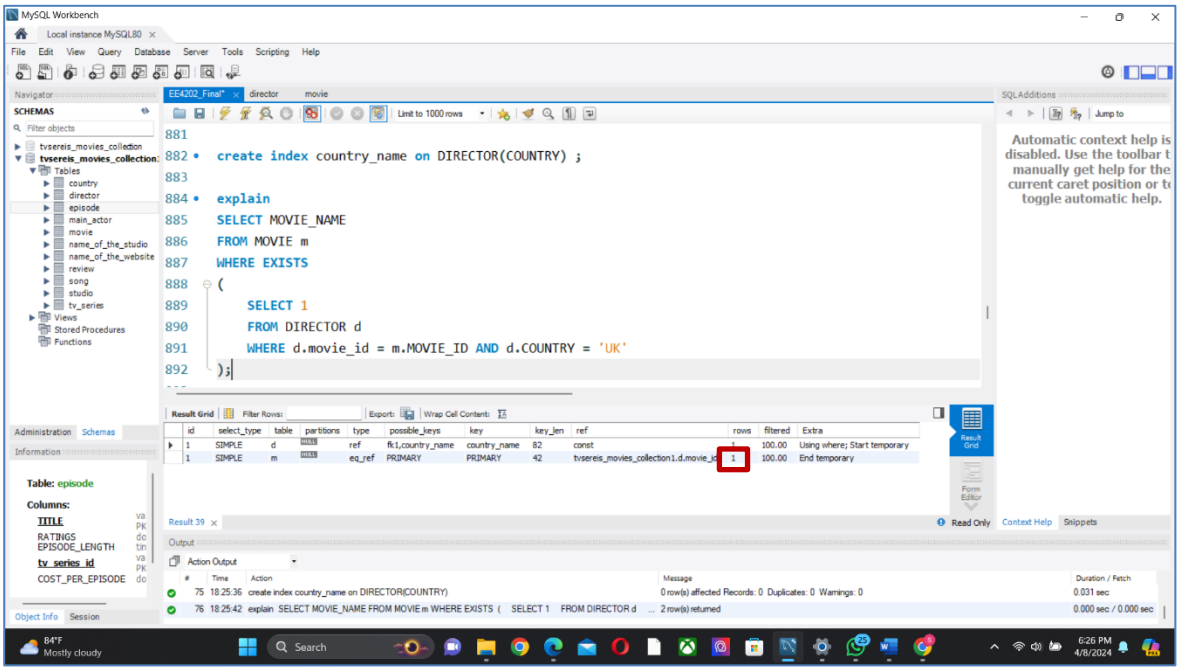


FIGURE 110 : COMPLEXITY OF THE QUERY AFTER ADDING THE COUNTRY_NAME

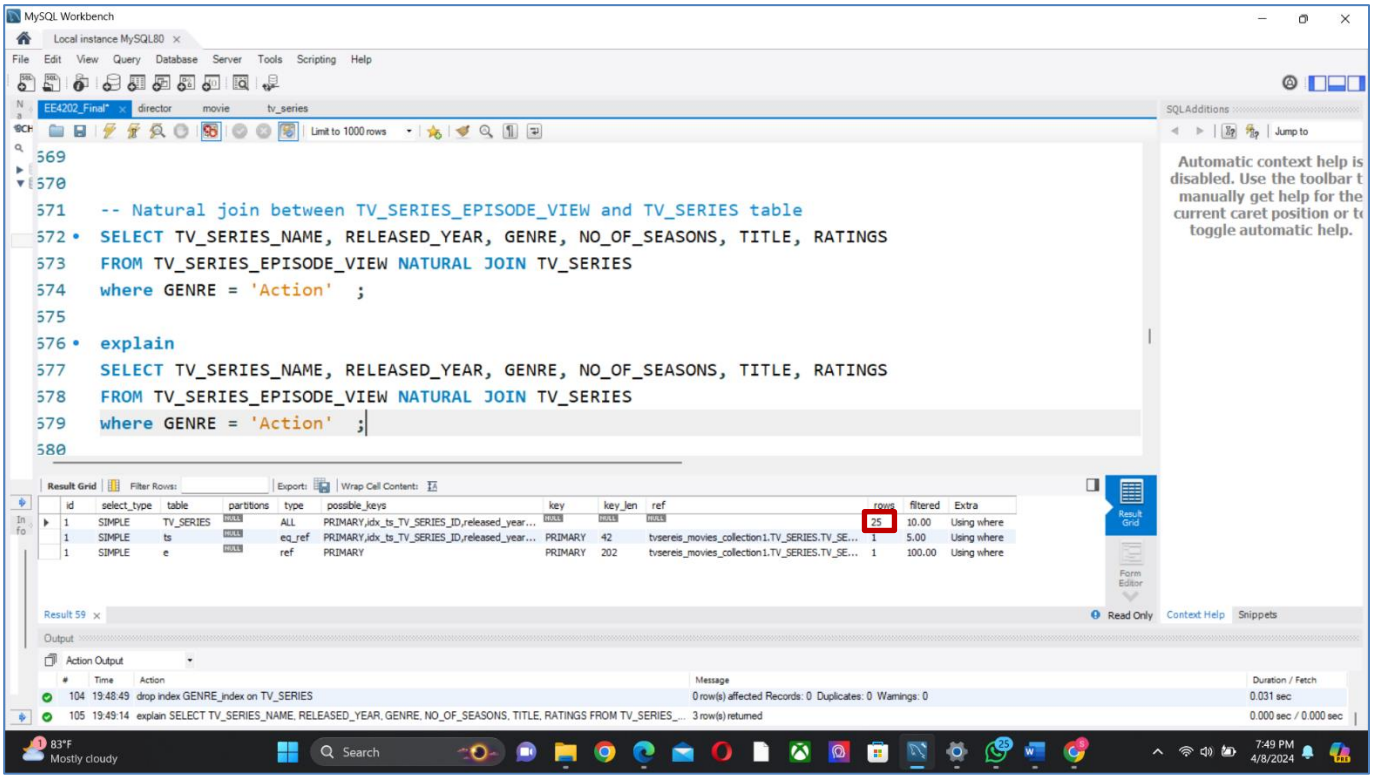


FIGURE 111 : COMPLEXITY OF THE QUERY BEFORE ADDING THE GENRE_INDEX

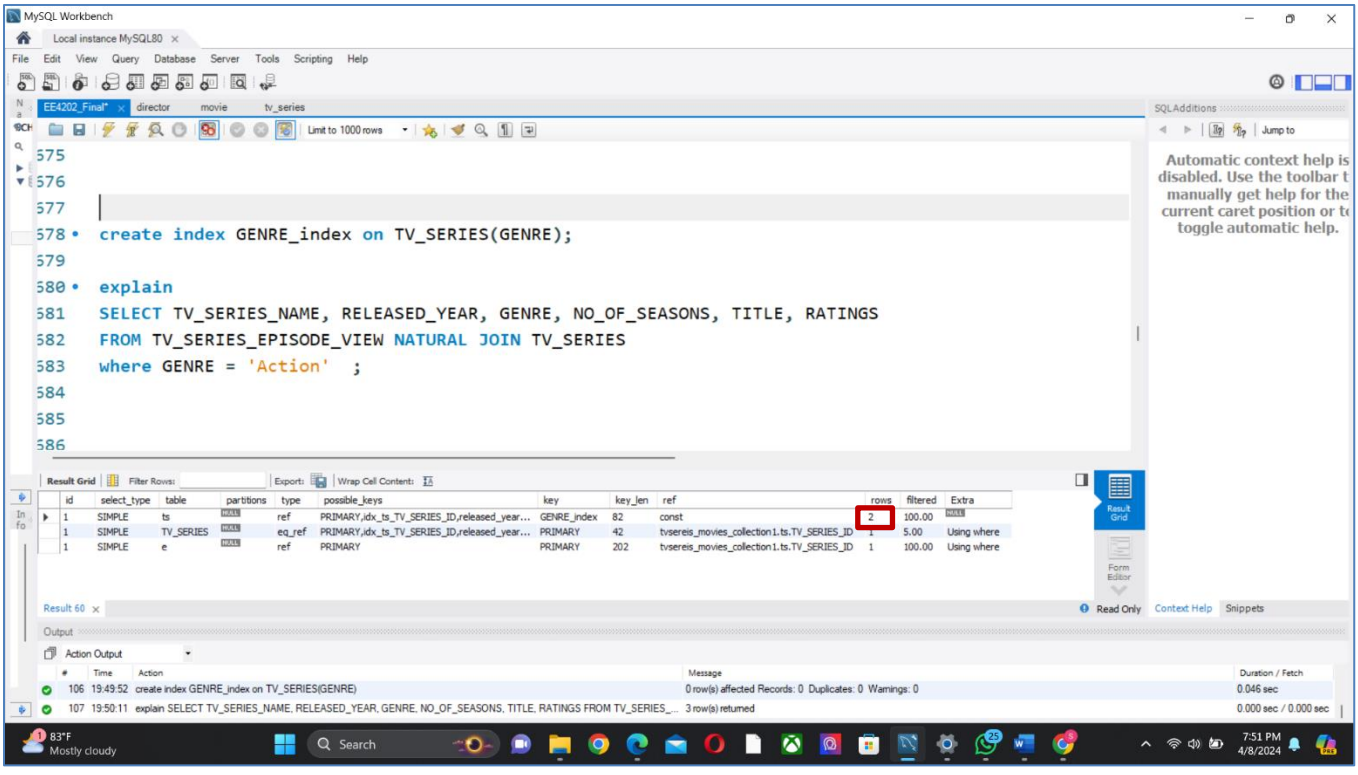


FIGURE 112 : COMPLEXITY OF THE QUERY AFTER ADDING THE GENRE_INDEX

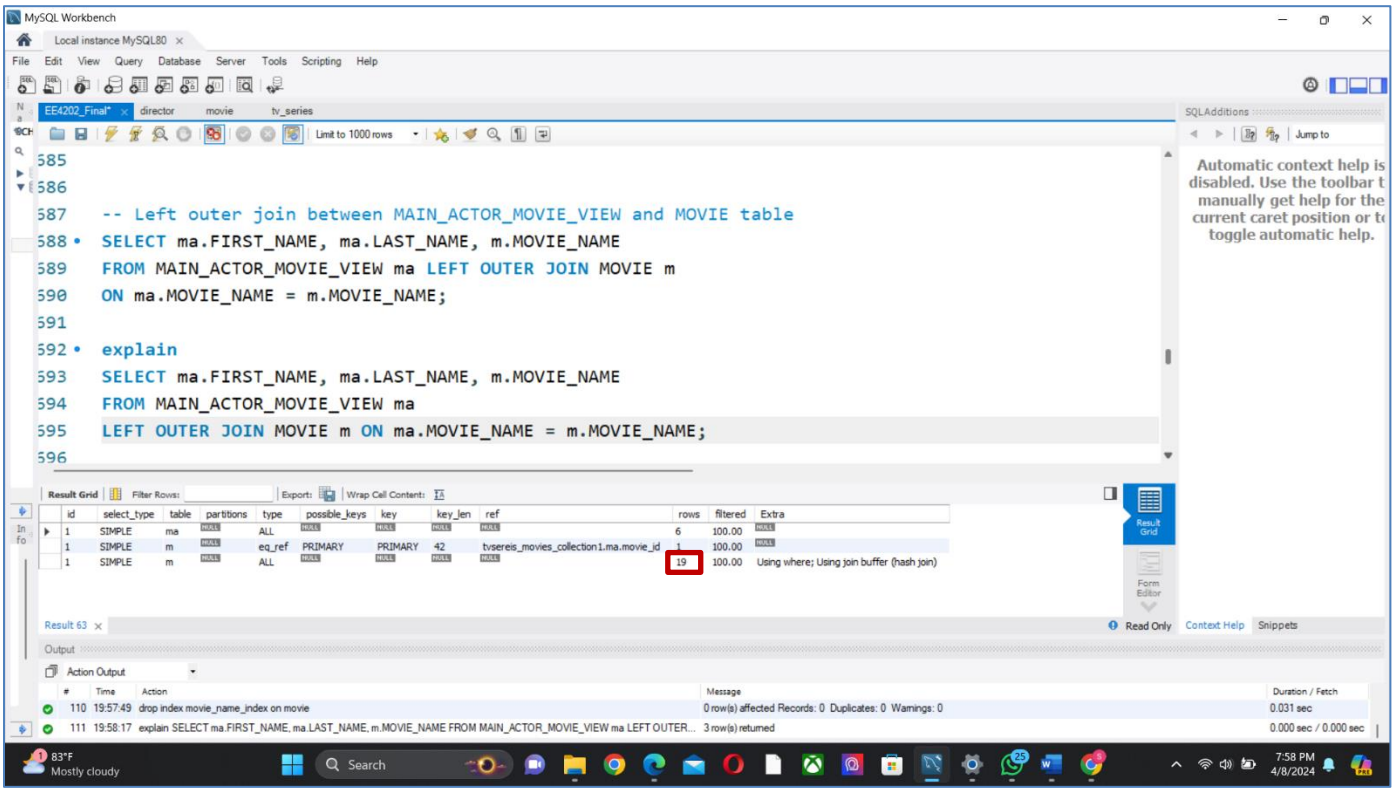


FIGURE 113 : COMPLEXITY OF THE QUERY BEFORE ADDING MOVIE_NAME_INDEX

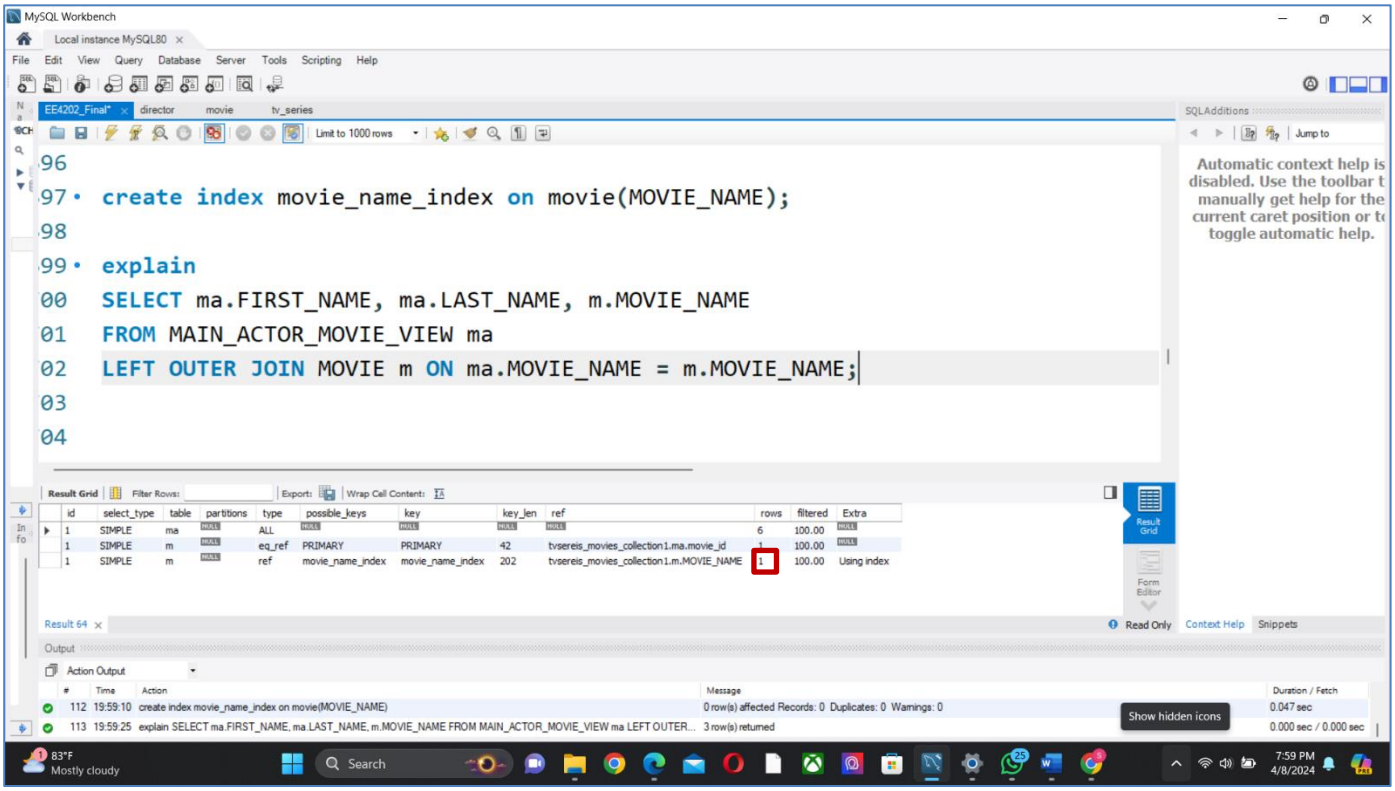


FIGURE 114 : COMPLEXITY OF THE QUERY AFTER ADDING MOVIE_NAME_INDEX

```
514 •   SELECT
515     min(RELEASED_YEAR)
516     AS Latest_Release
517     FROM MOVIE;
518
519
520 •   explain
521     SELECT
522     min(RELEASED_YEAR)
523     AS Latest_Release
524     FROM MOVIE;
525
```

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

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	MOVIE	NULL	ALL	NULL	NULL	NULL	NULL	19	100.00	NULL

FIGURE 115 : COMPLEXITY OF THE QUERY BEFORE ADDS THE INDEX

```
526 •   create unique index Latest_Release_index using BTREE on MOVIE(RELEASED_YEAR);
527
528 •   explain
529     SELECT
530     min(RELEASED_YEAR)
531     AS Latest_Release
532     FROM MOVIE;
533
```

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

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	Select tables optimized away

FIGURE 116: COMPLEXITY OF THE QUERY AFTER ADDS THE INDEX

- As the index was added according to the query tuning guide lines.
- It reduces the searching time from 19 rows to 0 rows.
- So, it reduces the complexity for the retrieval.

```
505 •   SELECT
506     DIRECTOR_NAME AS Director,
507     AGE AS Director_Age
508     FROM DIRECTOR
509     where AGE = 20;
510
511 •   EXPLAIN
512     SELECT
513     DIRECTOR_NAME AS Director,
514     AGE AS Director_Age
515     FROM DIRECTOR
516     where AGE = 20;
```

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

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	DIRECTOR	NULL	ALL	NULL	NULL	NULL	NULL	5	20.00	Using where

FIGURE 117 : COMPLEXITY OF THE QUERY BEFORE ADDS THE INDEX

```
505 •   SELECT
506     DIRECTOR_NAME AS Director,
507     AGE AS Director_Age
508     FROM DIRECTOR
509     where AGE = 20;
510
511 •   EXPLAIN
512     SELECT
513     DIRECTOR_NAME AS Director,
514     AGE AS Director_Age
515     FROM DIRECTOR
516     where AGE = 20;
```

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

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	DIRECTOR	NULL	const	Director_Age_index	Director_Age_index	5	const	1	100.00	NULL

FIGURE 118 : COMPLEXITY OF THE QUERY AFTER ADDS THE INDEX

iv. ERROR AND SOLUTION

ERROR 01

As we tried to create unique index for the non-unique attribute. So, we got this error.

0	80	20:37:15	create unique index GENRE_INDEX using BTREE on MOVIE(GENRE)	Error Code: 1062. Duplicate entry 'Drama' for key 'movie.GENRE_INDEX'	0.031 sec

3				
0	81	20:41:38	create unique index RELEASED_YEAR_INDEX using BTREE on MOVIE(RELEASED_YEAR)	Error Code: 1062. Duplicate entry '1994' for key 'movie.RELEASED_YEAR_INDEX'	0.032 sec

MOVIE_ID	MOVIE_NAME	RELEASED_YEAR	GENRE	DURATION	M_LANGUAGE	BUDGET
MV001	The Shawshank Redemption	1994	Drama	02:22:00	English	25000000.00000
MV002	The Godfather	1972	Drama, Crime	02:55:00	English	6000000.00000
MV003	The Dark Knight	2008	Action	02:32:00	English	200000000.00000
MV004	Pulp Fiction	1994	Crime	02:34:00	English	8000000.00000
MV005	Forrest Gump	1994	Drama	02:22:00	English	55000000.00000
MV006	Inception	2010	Science Fiction	02:28:00	English	160000000.00000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
691
692 • show index from MOVIE ;
```

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

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
movie	0	PRIMARY	1	MOVIE_ID	A	6	NULL	NULL		BTREE			YES	NULL
movie	0	MOVIE_NAME_INDEX	1	MOVIE_NAME	A	6	NULL	NULL		BTREE			YES	NULL

- **Because those are not unique keys. so, we cannot create the unique index. We have to create the secondary index (clustering index).**

ERROR - 02

```
explain
SELECT TV_SERIES_NAME
FROM TV_SERIES ts
WHERE ts.TV_SERIES_ID NOT IN (
    SELECT tv_series_id
    FROM EPISODE e
    WHERE e.RATINGS > (
        SELECT AVG(RATINGS)
        FROM EPISODE
    )
);
```

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

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	ts	NULL	ALL	NULL	NULL	NULL	NULL	5	100.00	NULL
1	PRIMARY	e	NULL	ref	PRIMARY	PRIMARY	202	tvsereis_movies_collection1.ts.TV_SERIES_ID	1	100.00	Using where; Not exists
3	SUBQUERY	EPISODE	NULL	ALL	NULL	NULL	NULL	NULL	4	100.00	NULL

FIGURE 119 : COMPLEXITY OF THE QUERY BEFORE ADDS THE INDEX

The error came as we try to implement the index again for the particular column.

0 74 20:25:03	create unique index TV_SERIES_NAME_INDEX using BTREE on TV_SERIES(TV_SERIES_NAME)	Error Code: 1061. Duplicate key name 'TV_SERIES_NAME_INDEX'	0.000 sec

```
create unique index TV_SERIES_NAME_INDEX
using BTREE
on TV_SERIES(TV_SERIES_NAME);

explain
SELECT TV_SERIES_NAME
FROM TV_SERIES ts
WHERE ts.TV_SERIES_ID NOT IN (
    SELECT tv_series_id
    FROM EPISODE e
    WHERE e.RATINGS > (
        SELECT AVG(RATINGS)
        FROM EPISODE
    )
);
```