Task 4

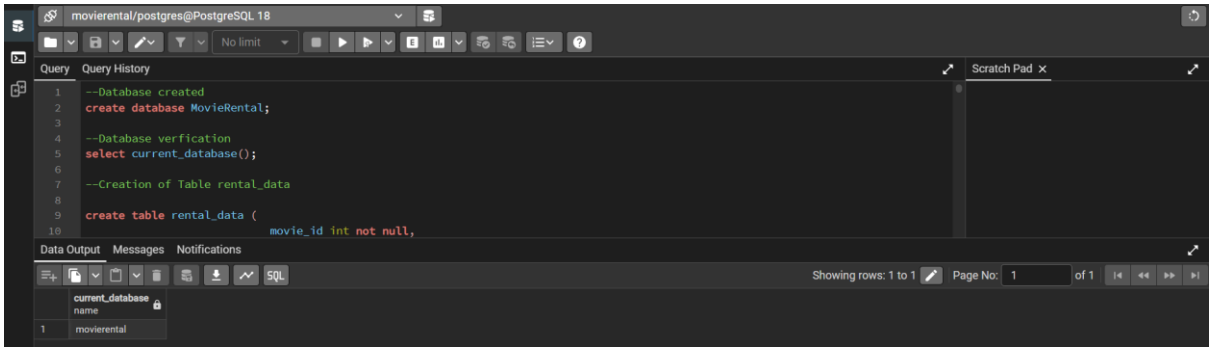
Project: Movie Rental Analysis System (using Redshift or PostgreSQL)

Objective:

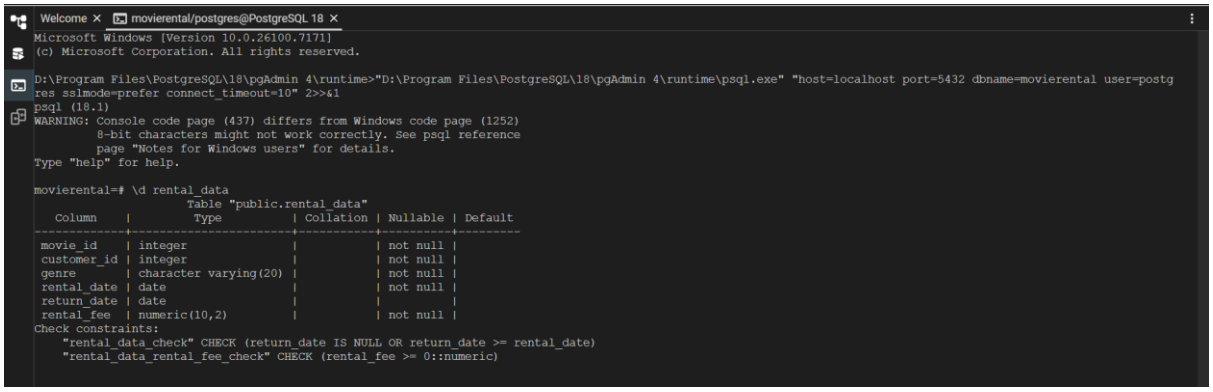
Perform advanced analysis on movie rental data using OLAP operations.

GITHUB: https://github.com/xrahulcrx/Movie_Rental_Analysis_System

Database Creation:

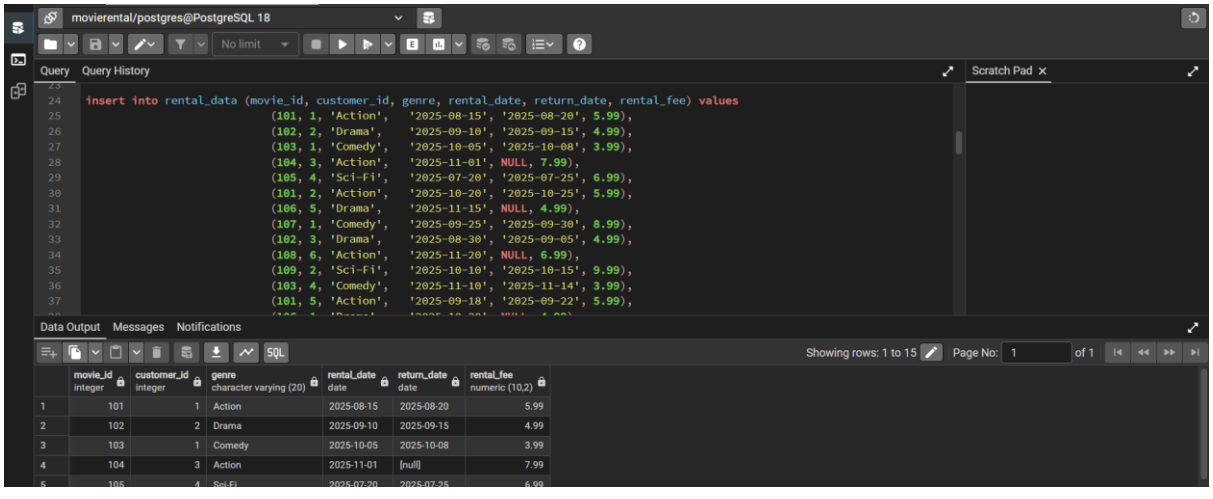


Create table rental_data with columns:



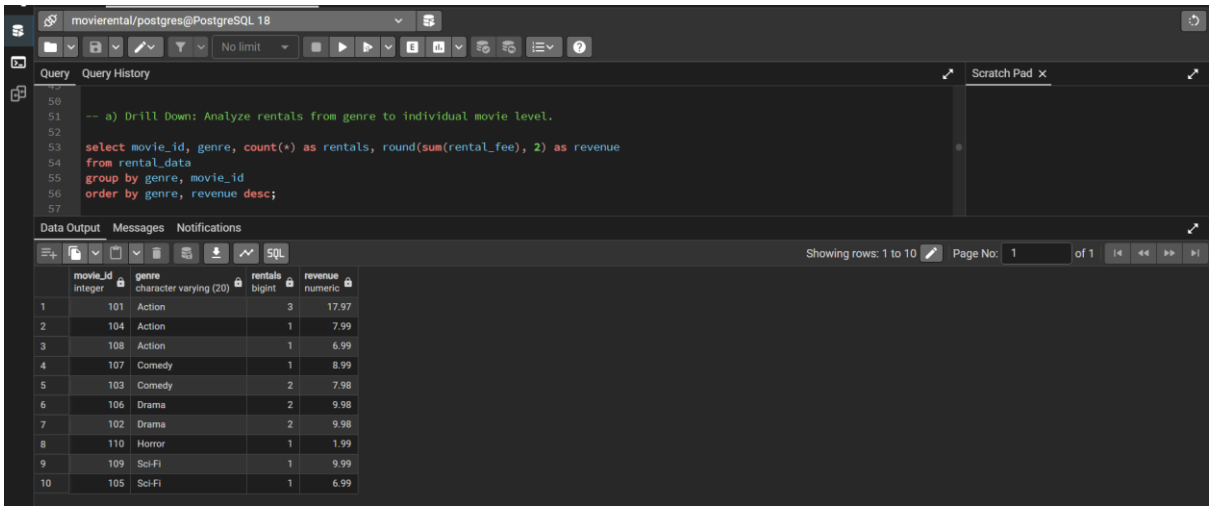
Data Creation:

Insert 10–15 sample rental records.



OLAP Operations:

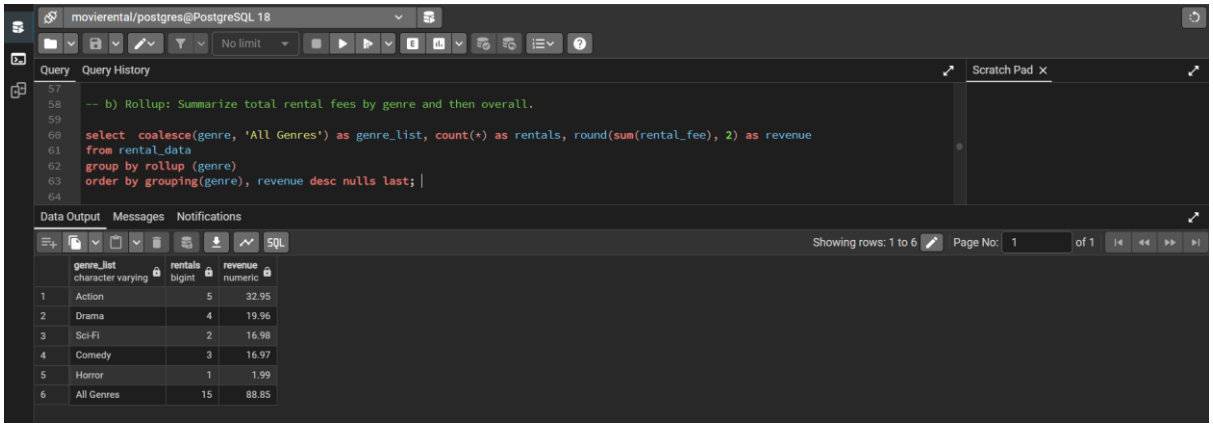
a) **Drill Down: Analyze rentals from genre to individual movie level.**



```
-- a) Drill Down: Analyze rentals from genre to individual movie level.

select movie_id, genre, count(*) as rentals, round(sum(rental_fee), 2) as revenue
from rental_data
group by genre, movie_id
order by genre, revenue desc;
```

	movie_id integer	genre character varying (20)	rentals bigint	revenue numeric
1	101	Action	3	17.97
2	104	Action	1	7.99
3	108	Action	1	6.99
4	107	Comedy	1	8.99
5	103	Comedy	2	7.98
6	106	Drama	2	9.98
7	102	Drama	2	9.98
8	110	Horror	1	1.99
9	109	Sci-Fi	1	9.99
10	105	Sci-Fi	1	6.99

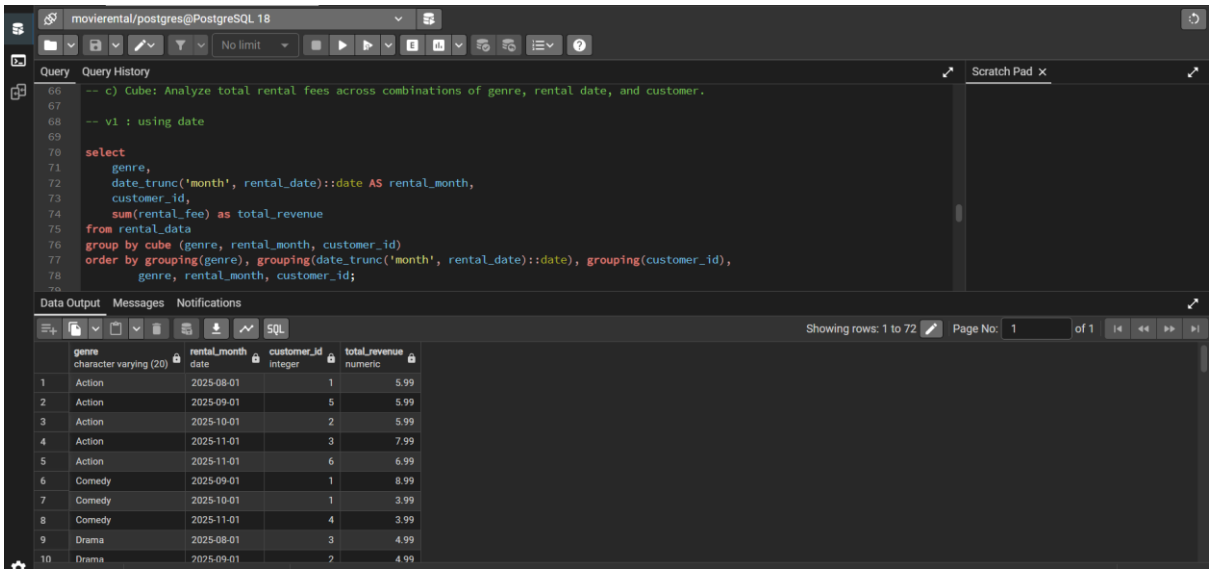
b) **Rollup: Summarize total rental fees by genre and then overall.**



```
-- b) Rollup: Summarize total rental fees by genre and then overall.

select  coalesce(genre, 'All Genres') as genre_list, count(*) as rentals, round(sum(rental_fee), 2) as revenue
from rental_data
group by rollup (genre)
order by grouping(genre), revenue desc nulls last;
```

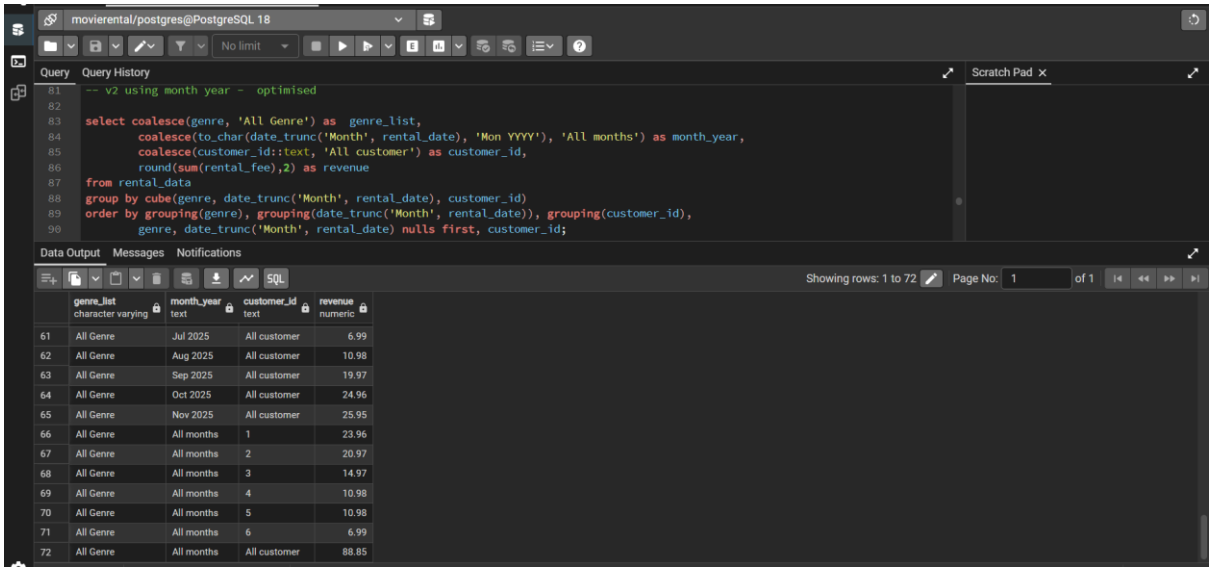
	genre_list character varying	rentals bigint	revenue numeric
1	Action	5	32.95
2	Drama	4	19.96
3	Sci-Fi	2	16.98
4	Comedy	3	16.97
5	Horror	1	1.99
6	All Genres	15	88.85

c) **Cube: Analyze total rental fees across combinations of genre, rental date, and customer.**



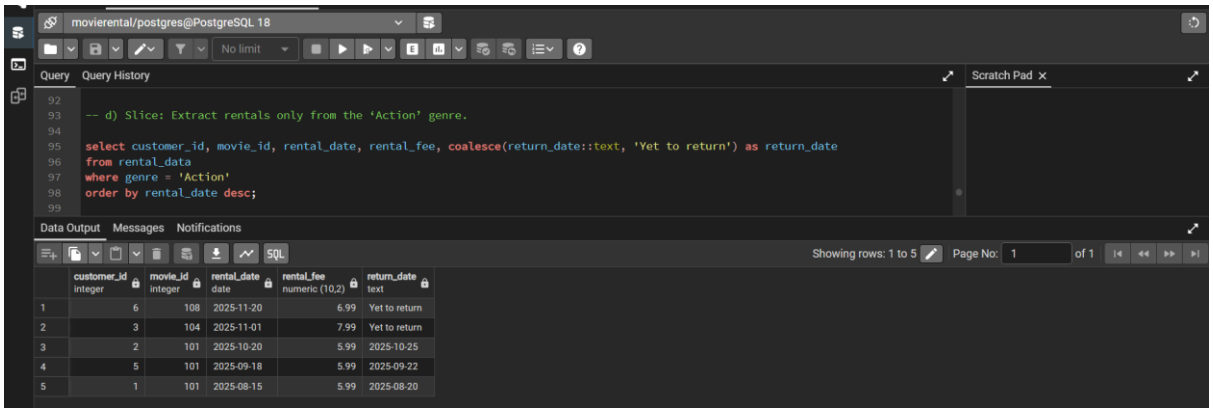
```
-- c) Cube: Analyze total rental fees across combinations of genre, rental date, and customer.

-- v1 : using date

select
    genre,
    date_trunc('month', rental_date)::date AS rental_month,
    customer_id,
    sum(rental_fee) as total_revenue
from rental_data
group by cube (genre, rental_month, customer_id)
order by grouping(genre), grouping(date_trunc('month', rental_date)::date), grouping(customer_id),
        genre, rental_month, customer_id;
```

	genre character varying (20)	rental_month date	customer_id integer	total_revenue numeric
1	Action	2025-08-01	1	5.99
2	Action	2025-09-01	5	5.99
3	Action	2025-10-01	2	5.99
4	Action	2025-11-01	3	7.99
5	Action	2025-11-01	6	6.99
6	Comedy	2025-09-01	1	8.99
7	Comedy	2025-10-01	1	3.99
8	Comedy	2025-11-01	4	3.99
9	Drama	2025-08-01	3	4.99
10	Drama	2025-09-01	2	4.99



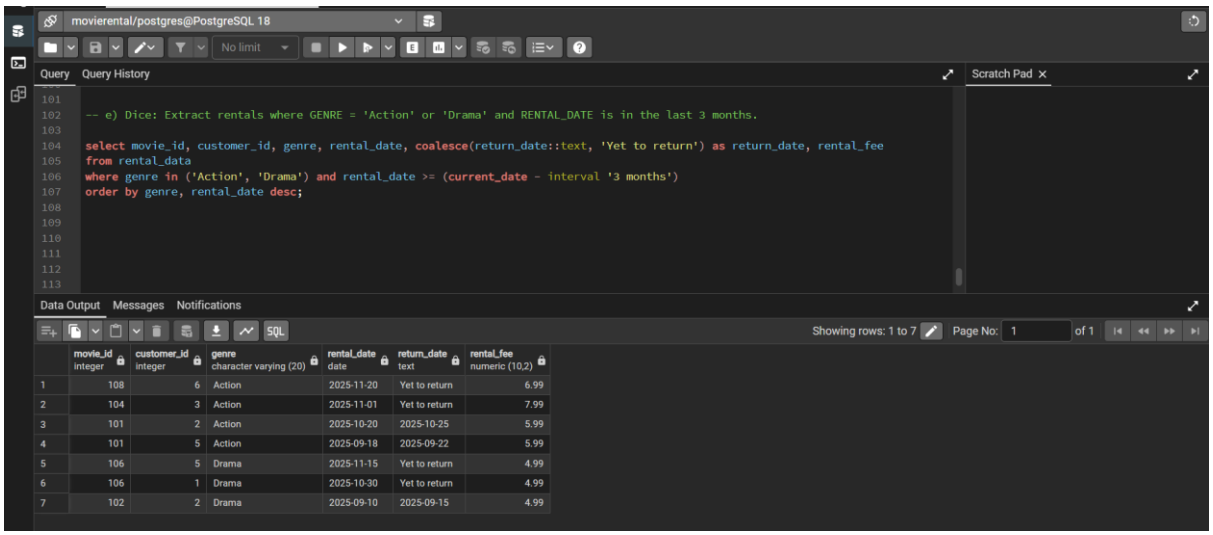
	genre_list character varying	month_year text	customer_id text	revenue numeric
61	All Genre	Jul 2025	All customer	6.99
62	All Genre	Aug 2025	All customer	10.98
63	All Genre	Sep 2025	All customer	19.97
64	All Genre	Oct 2025	All customer	24.96
65	All Genre	Nov 2025	All customer	25.95
66	All Genre	All months	1	23.96
67	All Genre	All months	2	20.97
68	All Genre	All months	3	14.97
69	All Genre	All months	4	10.98
70	All Genre	All months	5	10.98
71	All Genre	All months	6	6.99
72	All Genre	All months	All customer	88.85

d) Slice: Extract rentals only from the 'Action' genre.



	customer_id integer	movie_id integer	rental_date date	rental_fee numeric (10,2)	return_date text
1	6	108	2025-11-20	6.99	Yet to return
2	3	104	2025-11-01	7.99	Yet to return
3	2	101	2025-10-20	5.99	2025-10-25
4	5	101	2025-09-18	5.99	2025-09-22
5	1	101	2025-08-15	5.99	2025-08-20

e) Dice: Extract rentals where GENRE = 'Action' or 'Drama' and RENTAL_DATE is in the last 3 months.



	movie_id integer	customer_id integer	genre character varying (20)	rental_date date	return_date text	rental_fee numeric (10,2)
1	108	6	Action	2025-11-20	Yet to return	6.99
2	104	3	Action	2025-11-01	Yet to return	7.99
3	101	2	Action	2025-10-20	2025-10-25	5.99
4	101	5	Action	2025-09-18	2025-09-22	5.99
5	106	5	Drama	2025-11-15	Yet to return	4.99
6	106	1	Drama	2025-10-30	Yet to return	4.99
7	102	2	Drama	2025-09-10	2025-09-15	4.99