Task 4

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

Objective:

Perform advanced analysis on movie rental data using OLAP operations.

The project will include the following tasks:

Database Creation:

Create a database named MovieRental.

Create table rental_data with columns:

MOVIE_ID (integer),

CUSTOMER_ID (integer),

GENRE (varchar),

RENTAL_DATE (date),

RETURN_DATE (date),

RENTAL_FEE (numeric).

Data Creation:

Insert 10–15 sample rental records.

```
onlinestore=# CREATE DATABASE movierental;
CREATE DATABASE
onlinestore=# \c movierental
You are now connected to database "movierental" as user "postgres".
movierental=# CREATE TABLE rental_data (
movierental(#     movie_id INT,
movierental(#     customer_id INT,
movierental(#     genre VARCHAR(50),
movierental(#     rental_date DATE,
movierental(#     return_date DATE,
movierental(#     rental_fee NUMERIC(10,2)
movierental(# );
CREATE TABLE
movierental=# \dt
             List of relations
 Schema |    Name     | Type  |  Owner
--------+-------------+-------+----------
 public | rental_data | table | postgres
(1 row)


movierental=# INSERT INTO rental_data (movie_id, customer_id, genre, rental_date, return_date, rental_fee) VALUES
movierental-# (1, 101, 'Action', '2025-05-01', '2025-05-03', 150.00),
movierental-# (2, 102, 'Action', '2025-06-10', '2025-06-12', 200.00),
movierental-# (3, 103, 'Drama',  '2025-06-15', '2025-06-18', 120.00),
movierental-# (4, 104, 'Comedy', '2025-06-20', '2025-06-21', 100.00),
movierental-# (5, 105, 'Drama',  '2025-07-01', '2025-07-05', 180.00),
movierental-# (6, 101, 'Action', '2025-07-10', '2025-07-12', 170.00),
movierental-# (7, 106, 'Comedy', '2025-07-15', '2025-07-16', 90.00),
movierental-# (8, 107, 'Action', '2025-07-20', '2025-07-22', 220.00),
movierental-# (9, 102, 'Drama',  '2025-08-01', '2025-08-03', 140.00),
movierental-# (10,108, 'Horror', '2025-08-05', '2025-08-07', 160.00),
movierental-# (11,109, 'Action', '2025-08-10', '2025-08-12', 210.00),
movierental-# (12,110, 'Drama',  '2025-08-12', '2025-08-14', 190.00),
movierental-# (13,111, 'Comedy', '2025-08-15', '2025-08-16', 95.00),
movierental-# (14,112, 'Horror', '2025-08-18', '2025-08-20', 175.00),
movierental-# (15,113, 'Action', '2025-08-22', '2025-08-23', 230.00);
INSERT 0 15
movierental=# SELECT * FROM rental_data;
 movie_id | customer_id | genre  | rental_date | return_date | rental_fee
----------+-------------+--------+-------------+-------------+------------
        1 |         101 | Action | 2025-05-01  | 2025-05-03  |     150.00
        2 |         102 | Action | 2025-06-10  | 2025-06-12  |     200.00
        3 |         103 | Drama  | 2025-06-15  | 2025-06-18  |     120.00
        4 |         104 | Comedy | 2025-06-20  | 2025-06-21  |     100.00
        5 |         105 | Drama  | 2025-07-01  | 2025-07-05  |     180.00
        6 |         101 | Action | 2025-07-10  | 2025-07-12  |     170.00
        7 |         106 | Comedy | 2025-07-15  | 2025-07-16  |      90.00
        8 |         107 | Action | 2025-07-20  | 2025-07-22  |     220.00
        9 |         102 | Drama  | 2025-08-01  | 2025-08-03  |     140.00
       10 |         108 | Horror | 2025-08-05  | 2025-08-07  |     160.00
       11 |         109 | Action | 2025-08-10  | 2025-08-12  |     210.00
```

OLAP Operations:

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

```
movierental=# SELECT genre, movie_id, COUNT(*) AS total_rentals, SUM(rental_fee) AS total_
movierental-# FROM rental_data
movierental-# GROUP BY genre, movie_id
movierental-# ORDER BY genre, movie_id;
 genre  | movie_id | total_rentals | total_fees
--------+----------+---------------+------------
 Action |        1 |             1 |     150.00
 Action |        2 |             1 |     200.00
 Action |        6 |             1 |     170.00
 Action |        8 |             1 |     220.00
 Action |       11 |             1 |     210.00
 Action |       15 |             1 |     230.00
 Comedy |        4 |             1 |     100.00
 Comedy |        7 |             1 |      90.00
 Comedy |       13 |             1 |      95.00
 Drama  |        3 |             1 |     120.00
 Drama  |        5 |             1 |     180.00
 Drama  |        9 |             1 |     140.00
 Drama  |       12 |             1 |     190.00
 Horror |       10 |             1 |     160.00
 Horror |       14 |             1 |     175.00
(15 rows)
```

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

```
movierental=# SELECT genre, SUM(rental_fee) AS total_fees
movierental-# FROM rental_data
movierental-# GROUP BY ROLLUP (genre);
 genre  | total_fees
--------+------------
        |    2430.00
 Comedy |     285.00
 Horror |     335.00
 Drama  |     630.00
 Action |    1180.00
(5 rows)
```

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

```
movierental=# SELECT genre, rental_date, customer_id,
movierental-#        SUM(rental_fee) AS total_fees
movierental-# FROM rental_data
movierental-# GROUP BY CUBE (genre, rental_date, customer_id)
movierental-# ORDER BY genre, rental_date, customer_id;
 genre   | rental_date | customer_id | total_fees
---------+-------------+-------------+------------
 Action  | 2025-05-01  |         101 |     150.00
 Action  | 2025-05-01  |             |     150.00
 Action  | 2025-06-10  |         102 |     200.00
 Action  | 2025-06-10  |             |     200.00
 Action  | 2025-07-10  |         101 |     170.00
 Action  | 2025-07-10  |             |     170.00
 Action  | 2025-07-20  |         107 |     220.00
 Action  | 2025-07-20  |             |     220.00
 Action  | 2025-08-10  |         109 |     210.00
 Action  | 2025-08-10  |             |     210.00
 Action  | 2025-08-22  |         113 |     230.00
 Action  | 2025-08-22  |             |     230.00
 Action  |             |         101 |     320.00
 Action  |             |         102 |     200.00
 Action  |             |         107 |     220.00
 Action  |             |         109 |     210.00
 Action  |             |         113 |     230.00
 Action  |             |             |    1180.00
 Comedy  | 2025-06-20  |         104 |     100.00
 Comedy  | 2025-06-20  |             |     100.00
 Comedy  | 2025-07-15  |         106 |      90.00
 Comedy  | 2025-07-15  |             |      90.00
 Comedy  | 2025-08-15  |         111 |      95.00
 Comedy  | 2025-08-15  |             |      95.00
 Comedy  |             |         104 |     100.00
 Comedy  |             |         106 |      90.00
 Comedy  |             |         111 |      95.00
 Comedy  |             |             |     285.00
 Drama   | 2025-06-15  |         103 |     120.00
 Drama   | 2025-06-15  |             |     120.00
 Drama   | 2025-07-01  |         105 |     180.00
 Drama   | 2025-07-01  |             |     180.00
 Drama   | 2025-08-01  |         102 |     140.00
 Drama   | 2025-08-01  |             |     140.00
 Drama   | 2025-08-12  |         110 |     190.00
 Drama   | 2025-08-12  |             |     190.00
 Drama   |             |         102 |     140.00
 Drama   |             |         103 |     120.00
 Drama   |             |         105 |     180.00
 Drama   |             |         110 |     190.00
 Drama   |             |             |     630.00
 Horror  | 2025-08-05  |         108 |     160.00
```

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

```
movierental=# SELECT *
movierental-# FROM rental_data
movierental-# WHERE genre = 'Action';
 movie_id | customer_id | genre  | rental_date | return_date | rental_fee
----------+-------------+--------+-------------+-------------+------------
        1 |         101 | Action | 2025-05-01  | 2025-05-03  |     150.00
        2 |         102 | Action | 2025-06-10  | 2025-06-12  |     200.00
        6 |         101 | Action | 2025-07-10  | 2025-07-12  |     170.00
        8 |         107 | Action | 2025-07-20  | 2025-07-22  |     220.00
       11 |         109 | Action | 2025-08-10  | 2025-08-12  |     210.00
       15 |         113 | Action | 2025-08-22  | 2025-08-23  |     230.00
(6 rows)
```

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

```
movierental=# SELECT *
movierental-# FROM rental_data
movierental-# WHERE genre IN ('Action','Drama')
movierental-#   AND rental_date >= CURRENT_DATE - INTERVAL '3 months';
 movie_id | customer_id | genre  | rental_date | return_date | rental_fee
----------+-------------+--------+-------------+-------------+------------
        2 |         102 | Action | 2025-06-10  | 2025-06-12  |     200.00
        3 |         103 | Drama  | 2025-06-15  | 2025-06-18  |     120.00
        5 |         105 | Drama  | 2025-07-01  | 2025-07-05  |     180.00
        6 |         101 | Action | 2025-07-10  | 2025-07-12  |     170.00
        8 |         107 | Action | 2025-07-20  | 2025-07-22  |     220.00
        9 |         102 | Drama  | 2025-08-01  | 2025-08-03  |     140.00
       11 |         109 | Action | 2025-08-10  | 2025-08-12  |     210.00
       12 |         110 | Drama  | 2025-08-12  | 2025-08-14  |     190.00
       15 |         113 | Action | 2025-08-22  | 2025-08-23  |     230.00
(9 rows)
```