Project 4

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

Task 1 Database Creation:

Question:

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).

Query:

```
CREATE DATABASE MovieRental;

USE MovieRental;

CREATE TABLE rental_data (

MOVIE_ID integer,

CUSTOMER_ID integer,

GENRE varchar(50),

RENTAL_DATE date,
RETURN_DATE date,

RENTAL_FEE numeric(5,2)

);
```

Task 2 Data Entry:

Question:

Insert 10–15 sample rental records.

Query:

```
INSERT INTO rental_data (MOVIE_ID, CUSTOMER_ID, GENRE, RENTAL_DATE,
RETURN_DATE, RENTAL_FEE)

VALUES

(1, 101, 'Action', '2025-06-01', '2025-06-03', 4.99),

(2, 102, 'Comedy', '2025-06-02', '2025-06-04', 3.99),

(3, 103, 'Drama', '2025-06-03', '2025-06-06', 4.99),

(4, 104, 'Action', '2025-06-04', '2025-06-07', 5.99),

(5, 105, 'Sci-Fi', '2025-06-05', '2025-06-08', 5.99),

(1, 106, 'Action', '2025-06-06', '2025-06-09', 4.99),

(6, 107, 'Comedy', '2025-06-07', '2025-06-10', 3.99),

(7, 108, 'Drama', '2025-06-08', '2025-06-11', 4.99),

(8, 109, 'Sci-Fi', '2025-06-09', '2025-06-12', 5.99),

(9, 110, 'Action', '2025-06-10', '2025-06-13', 5.99),

(10, 111, 'Comedy', '2025-06-11', '2025-06-14', 3.99),

(11, 112, 'Drama', '2025-06-12', '2025-06-15', 4.99),

(12, 113, 'Action', '2025-06-13', '2025-06-16', 5.99),

(13, 114, 'Sci-Fi', '2025-06-14', '2025-06-17', 5.99),

(14, 115, 'Comedy', '2025-06-15', '2025-06-18', 3.99);
```

Task 3 OLAP Operations:

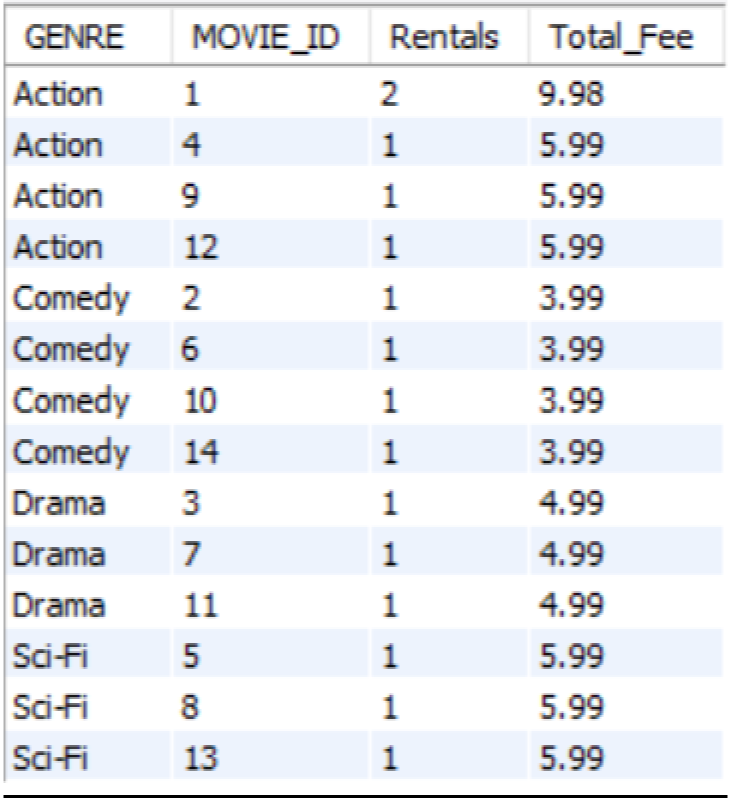
Question:

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

Query:

```
SELECT GENRE, MOVIE_ID, COUNT(*) as Rentals, SUM(RENTAL_FEE) as Total_Fee

FROM rental_data

GROUP BY GENRE, MOVIE_ID

ORDER BY GENRE, MOVIE_ID;
```

OUTPUT:



GENRE	MOVIE_ID	Rentals	Total_Fee
Action	1	2	9.98
Action	4	1	5.99
Action	9	1	5.99
Action	12	1	5.99
Comedy	2	1	3.99
Comedy	6	1	3.99
Comedy	10	1	3.99
Comedy	14	1	3.99
Drama	3	1	4.99
Drama	7	1	4.99
Drama	11	1	4.99
Sci-Fi	5	1	5.99
Sci-Fi	8	1	5.99
Sci-Fi	13	1	5.99

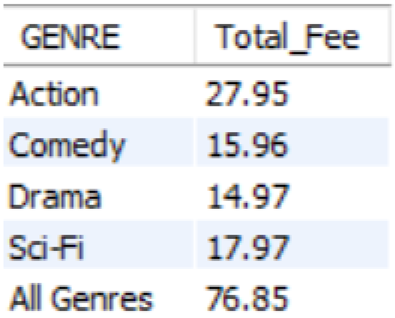
Question:

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

Query:

```
SELECT IFNULL(GENRE, 'All Genres') as GENRE, SUM(RENTAL_FEE) as Total_Fee

FROM rental_data

GROUP BY GENRE WITH ROLLUP;
```

OUTPUT:



GENRE	Total_Fee
Action	27.95
Comedy	15.96
Drama	14.97
Sci-Fi	17.97
All Genres	76.85

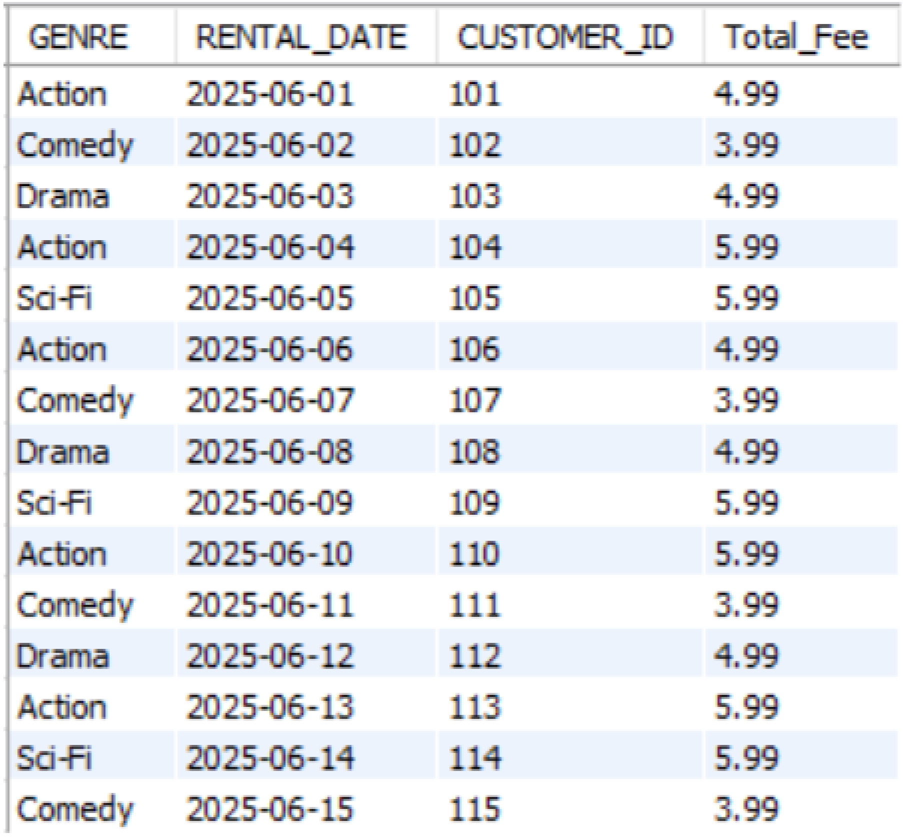
Question:

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

Query:

```
(SELECT GENRE, RENTAL_DATE, CUSTOMER_ID, SUM(RENTAL_FEE) as Total_Fee

FROM rental_data

GROUP BY GENRE, RENTAL_DATE, CUSTOMER_ID

UNION ALL

SELECT GENRE, RENTAL_DATE, NULL, SUM(RENTAL_FEE) as Total_Fee

FROM rental_data

GROUP BY GENRE, RENTAL_DATE

UNION ALL

SELECT GENRE, NULL, NULL, SUM(RENTAL_FEE) as Total_Fee

FROM rental_data
GROUP BY GENRE
UNION ALL
SELECT NULL, NULL, NULL, SUM(RENTAL_FEE) as Total_Fee
FROM rental_data)
LIMIT 15;
```

OUTPUT:



GENRE	RENTAL_DATE	CUSTOMER_ID	Total_Fee
Action	2025-06-01	101	4.99
Comedy	2025-06-02	102	3.99
Drama	2025-06-03	103	4.99
Action	2025-06-04	104	5.99
Sci-Fi	2025-06-05	105	5.99
Action	2025-06-06	106	4.99
Comedy	2025-06-07	107	3.99
Drama	2025-06-08	108	4.99
Sci-Fi	2025-06-09	109	5.99
Action	2025-06-10	110	5.99
Comedy	2025-06-11	111	3.99
Drama	2025-06-12	112	4.99
Action	2025-06-13	113	5.99
Sci-Fi	2025-06-14	114	5.99
Comedy	2025-06-15	115	3.99

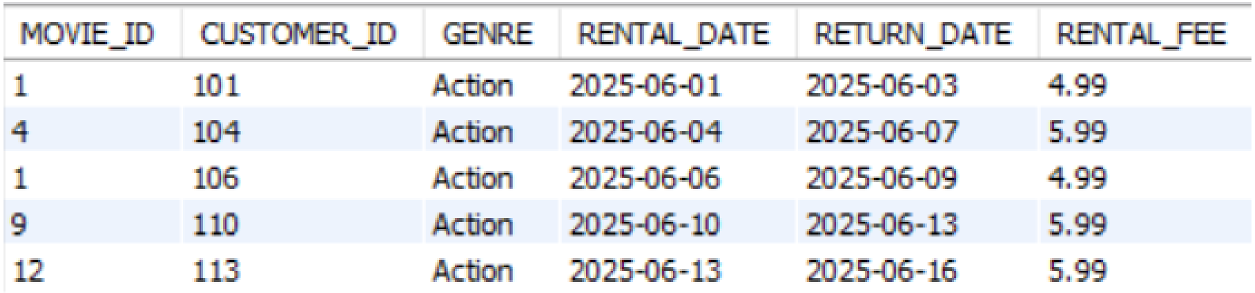
Question:

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

Query:

```
SELECT *
FROM rental_data
WHERE GENRE = 'Action';
```

OUTPUT:



MOVIE_ID	CUSTOMER_ID	GENRE	RENTAL_DATE	RETURN_DATE	RENTAL_FEE
1	101	Action	2025-06-01	2025-06-03	4.99
4	104	Action	2025-06-04	2025-06-07	5.99
1	106	Action	2025-06-06	2025-06-09	4.99
9	110	Action	2025-06-10	2025-06-13	5.99
12	113	Action	2025-06-13	2025-06-16	5.99

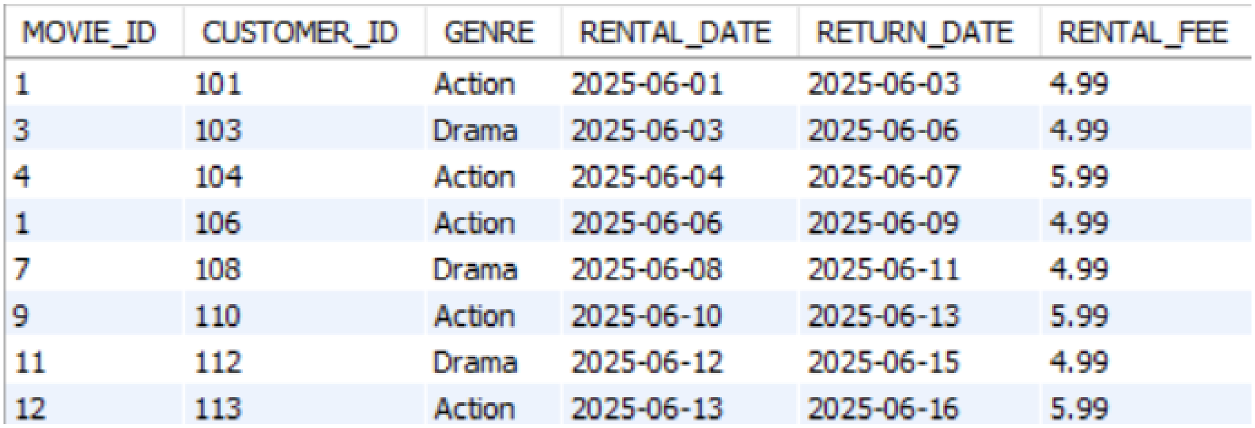
Question:

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

Query:

```
SELECT *

FROM rental_data

WHERE GENRE IN ('Action', 'Drama')

AND RENTAL_DATE >= DATE_SUB(CURDATE(), INTERVAL 3 MONTH);
```

OUTPUT:



MOVIE_ID	CUSTOMER_ID	GENRE	RENTAL_DATE	RETURN_DATE	RENTAL_FEE
1	101	Action	2025-06-01	2025-06-03	4.99
3	103	Drama	2025-06-03	2025-06-06	4.99
4	104	Action	2025-06-04	2025-06-07	5.99
1	106	Action	2025-06-06	2025-06-09	4.99
7	108	Drama	2025-06-08	2025-06-11	4.99
9	110	Action	2025-06-10	2025-06-13	5.99
11	112	Drama	2025-06-12	2025-06-15	4.99
12	113	Action	2025-06-13	2025-06-16	5.99