

TASK 6: SALES TREND ANALYSIS USING AGGREGATIONS

1.Monthly Revenue and Rental Volume:

Query Query History

Scratch Pad x

```
1 SELECT
2   EXTRACT(YEAR FROM p.payment_date) AS payment_year,
3   EXTRACT(MONTH FROM p.payment_date) AS payment_month,
4   SUM(p.amount) AS total_revenue,
5   COUNT(DISTINCT r.rental_id) AS total_rentals
6 FROM
7   payment p
8 JOIN
9   rental r ON p.rental_id = r.rental_id
10 GROUP BY
11   payment_year, payment_month
12 ORDER BY
13   payment_year, payment_month;
```

rental_id date ,return ,last_updatee
renta_id amount ,payment_date

Data Output Messages Notifications

Showing rows: 1 to 4 Page No: 1 of 1

	payment_year numeric	payment_month numeric	total_revenue numeric	total_rentals bigint
1	2007	2	8351.84	2016
2	2007	3	23886.56	5644
3	2007	4	28559.46	6752
4	2007	5	514.18	182

2.Results for a Specific Month and Year (Result from second, third ,fourth months)

Query

```
1 SELECT
2     EXTRACT(YEAR FROM p.payment_date) AS payment_year,
3     EXTRACT(MONTH FROM p.payment_date) AS payment_month,
4     SUM(p.amount) AS total_revenue,
5     COUNT(DISTINCT r.rental_id) AS total_rentals
6 FROM
7     payment p
8 JOIN
9     rental r ON p.rental_id = r.rental_id
10 WHERE
11     EXTRACT(YEAR FROM p.payment_date) = 2007 AND EXTRACT(MONTH
12 GROUP BY
13     payment_year, payment_month
14 ORDER BY
15     payment_year, payment_month;
```

Scratch Pad x

payment_date
"2007-04-09 17:21:21.996577"
rental_id date ,return ,last_updatee
renta_id amount ,payment_date

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1

	payment_year numeric	payment_month numeric	total_revenue numeric	total_rentals bigint
1	2007	4	28559.46	6752

Query

```
1 SELECT
2     EXTRACT(YEAR FROM p.payment_date) AS payment_year,
3     EXTRACT(MONTH FROM p.payment_date) AS payment_month,
4     SUM(p.amount) AS total_revenue,
5     COUNT(DISTINCT r.rental_id) AS total_rentals
6 FROM
7     payment p
8 JOIN
9     rental r ON p.rental_id = r.rental_id
10 WHERE
11     EXTRACT(YEAR FROM p.payment_date) = 2007 AND EXTRACT(MONTH
12 GROUP BY
13     payment_year, payment_month
14 ORDER BY
15     payment_year, payment_month;
```

Scratch Pad x

payment_date
"2007-04-09 17:21:21.996577"
rental_id date ,return ,last_updatee
renta_id amount ,payment_date

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of

	payment_year numeric	payment_month numeric	total_revenue numeric	total_rentals bigint
1	2007	3	23886.56	5644

dvdrental/postgres@PostgreSQL 17

Query Query History Scratch Pad x

```

1 SELECT
2   EXTRACT(YEAR FROM p.payment_date) AS payment_year,
3   EXTRACT(MONTH FROM p.payment_date) AS payment_month,
4   SUM(p.amount) AS total_revenue,
5   COUNT(DISTINCT r.rental_id) AS total_rentals
6 FROM
7   payment p
8 JOIN
9   rental r ON p.rental_id = r.rental_id
10 WHERE
11   EXTRACT(YEAR FROM p.payment_date) = 2007 AND EXTRACT(MONTH
12 GROUP BY
13   payment_year, payment_month
14 ORDER BY
15   payment_year, payment_month;

```

Scratch Pad x

```

"payment_date"
"2007-04-09 17:21:21.996577"
rental_id date,return_last_updatee
renta_id amount ,payment_date

```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1

	payment_year numeric	payment_month numeric	total_revenue numeric	total_rentals bigint
1	2007	2	8351.84	2016

3.Results for Specific Time Periods

dvdrental/postgres@PostgreSQL 17

Query Query History Scratch Pad x

```

1 SELECT
2   EXTRACT(YEAR FROM p.payment_date) AS payment_year,
3   EXTRACT(MONTH FROM p.payment_date) AS payment_month,
4   SUM(p.amount) AS total_revenue,
5   COUNT(DISTINCT r.rental_id) AS total_rentals
6 FROM
7   payment p
8 JOIN
9   rental r ON p.rental_id = r.rental_id
10 WHERE
11   EXTRACT(YEAR FROM p.payment_date) = 2007 -- Limiting to the
12 GROUP BY
13   payment_year, payment_month
14 ORDER BY
15   payment_year, payment_month;
16
17

```

Scratch Pad x

```

"payment_date"
"2007-04-09 17:21:21.996577"
rental_id date,return_last_updatee
renta_id amount ,payment_date

```

Data Output Messages Notifications

Showing rows: 1 to 4 Page No:

	payment_year numeric	payment_month numeric	total_revenue numeric	total_rentals bigint
1	2007	2	8351.84	2016
2	2007	3	23886.56	5644
3	2007	4	28559.46	6752
4	2007	5	514.18	182

4.Count for Volume

The screenshot shows a PostgreSQL query editor interface. The top bar indicates the connection is to 'dvdrental/postgres@PostgreSQL 17'. The main editor area contains the following SQL query:

```
1 SELECT
2   COUNT(DISTINCT r.rental_id) AS rental_volume
3 FROM
4   rental r;
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with the following structure:

	rental_volume bigint
1	16044

The table shows a single row with the value 16044 for the rental_volume column. The interface also includes a 'Query History' tab, a 'Scratch Pad' tab, and a 'Messages' tab. The 'Data Output' tab has a toolbar with icons for adding, deleting, and refreshing data, as well as a 'SQL' button. The status bar at the bottom indicates 'Showing rows: 1 to 1' and 'Page No: 1 of 1'.