

Name – Swaraj Jadhav

Branch – Computer Science

Email – jswaraj48@gmail.com

1. Database Query Optimization

Scenario: You have a PostgreSQL database with a `users` table and an `orders` table. The `orders` table has a foreign key reference to the `users` table. Write a SQL query to fetch the top 5 users with the highest total order amount in the last month. Assume that the `orders` table has columns `user_id`, `amount`, and `order_date`.

Requirements:

- Write the SQL query.
Explain how you would index the tables to optimise the query.

---> Query

```
mysql> SELECT u.user_id, SUM(o.amount) AS all_amount
-> FROM users u
-> JOIN orders o ON u.user_id =o.user_id
-> WHERE o.order_date >= (CURRENT_DATE - INTERVAL '1 month')
-> GROUP BY u.user_id
-> ORDER BY all_amount DESC
-> LIMIT 5;|
```

Explanation :

JOIN: Using the foreign key connection (`user_id`), this joins the `users` and `orders` tables.

WHERE: Filters records to include only orders made in the last month (`order_date >= (CURRENT_DATE - INTERVAL '1 month')`).

GROUP BY: Using this we can add up the total for every user, the results are grouped by `user_id`.

ORDER BY: Orders the results by the total amount in descending order to get the top users.

LIMIT 5: Restricts the result to the top 5 users.

Indexing the table to optimize the query :

- To implement the query it is required to join the tables `users` and `orders`. The unique and common in both table is `user_id`. So if we index on `orders.user_id` it will speed up the join process by reducing no of rows that needs to be scanned.
- As we need to find relevant records within last month, we can index on the column of `order_date`. It will help to find relevant records.