Assignment 4

142201019
G.Sai Rohith

Q3)
Transaction-Centric CRUD Timings (seconds):
Insert: 0.000950
Read: 0.000728
Update: 0.000621
Delete: 0.000569

Customer-Centric CRUD Timings (seconds):
Insert: 0.000855
Read: 0.000615
Update: 0.000689
Delete: 0.000566

CRUD Performance Comparison: Transaction-Centric vs Customer-Centric

1. **Insert Operation**
    - Customer-Centric inserts are generally faster for small datasets because all invoices for a customer are embedded in a single document.
2. **Read Operation**
    - Both approaches provide fast read access. Customer-Centric reads can be slightly faster when fetching all invoices for a customer.
3. **Update Operation**
    - Transaction-Centric updates are simpler and faster because each invoice is a separate document.
    - Customer-Centric updates require navigating nested arrays, which adds slight overhead.
4. **Delete Operation**
    - Deleting a single invoice in Transaction-Centric is straightforward.
    - Deleting an invoice from a nested array in Customer-Centric is slightly more complex but comparable for small datasets.
5. **General Observations**
    - **Transaction-Centric:** Best suited for invoice-level operations; CRUD operations are isolated and simple.

- **Customer-Centric:** Best suited for customer-level aggregation or reporting; operations on individual invoices are more complex due to nesting.

6. **Scalability Insight**

   - Transaction-Centric scales better for large numbers of invoices.
   - Customer-Centric may slow down for updates and deletes as the number of invoices per customer grows.

7. **Conclusion**

   - Transaction-Centric is optimal for invoice-focused workflows.
   - Customer-Centric is optimal for customer-focused workflows and reporting.

Q4)