

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)

The screenshot displays the MongoDB Atlas mlops interface. The top navigation bar includes the Atlas logo, a dropdown menu for 'GALAM's Or...', and links for 'Access Manager' and 'Billing'. On the right, there are links for 'All Clusters', 'Get Help', and a 'GALAM' button. The main interface is divided into a left sidebar and a central content area. The sidebar contains a 'Project 0' dropdown, a 'Data Services' tab, and a 'Charts' tab. Below this is a navigation menu with sections: 'Overview', 'DATABASE', 'Clusters', 'SERVICES', 'Atlas Search', 'Stream Processing', 'Triggers', 'Migration', 'Data Federation', 'SECURITY', 'Quickstart', 'Backup', 'Database Access', 'Network Access', 'Advanced', and 'Goto'. The central content area shows the 'online_retail.transactions' collection. It includes a '+ Create Database' button, a search bar for namespaces, and a list of collections: 'online_retail', 'customers', 'transactions', 'retail_db', and 'sample_mflix'. The 'transactions' collection is selected. The main view for this collection shows its metadata: 'STORAGE SIZE: 133MB', 'LOGICAL DATA SIZE: 2MB', 'TOTAL DOCUMENTS: 1000', and 'INDEXES TOTAL SIZE: 68KB'. It also has tabs for 'Find', 'Indexes', 'Schema Anti-Patterns', 'Aggregation', and 'Search Indexes'. A 'Generate queries from natural language in Compass' link is present. Below this is a 'Filter' section with a query input field. A sample document is displayed in a JSON format:

```
{
  "_id": "536365",
  "invoiceDate": "2010-12-01T08:26:00.000+00:00",
  "customer": {
    "id": 17850,
    "country": "United Kingdom"
  },
  "items": [
    {
      "stockCode": "85123A",
      "description": "WHITE HANGING HEART T-LIGHT HOLDER",
      "quantity": 6
    }
  ]
}
```

 On the right side of the interface, there are buttons for 'PREVIEW', 'New Data Explorer', 'VISUALIZE YOUR DATA', and 'REFRESH'. A chatbot window for 'MBot' is also visible, offering help with the next project.