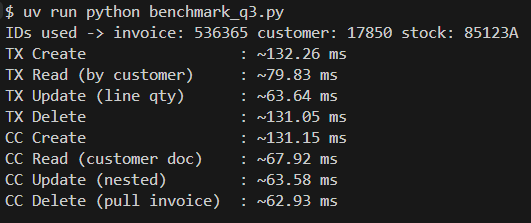
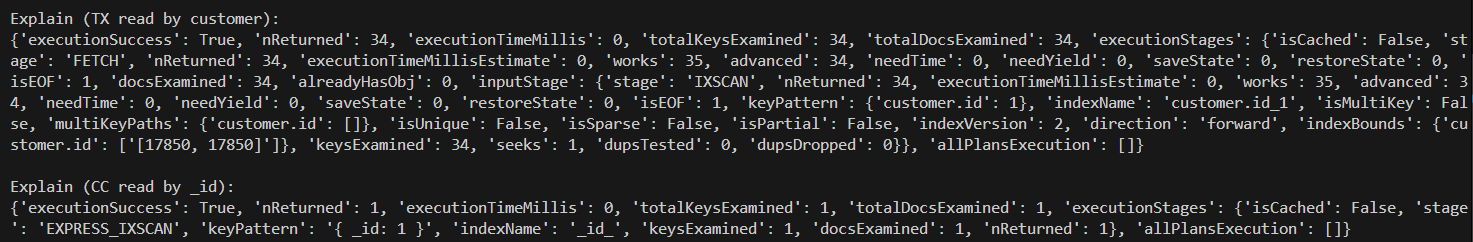
Assignment 4

3. Perform CRUD operations on both approaches and write the differences you have observed regarding the performance and provide proof for your claims. (5 Marks)



We compared **Transaction-Centric (TX)** and **Customer-Centric (CC)** MongoDB data models by performing CRUD operations and analyzing execution statistics.

* In the **TX model**, each invoice is stored as a separate document.
  + To fetch all invoices of a customer, MongoDB must scan multiple documents (e.g., 34 documents for customer 17850).
  + This makes TX better for **invoice-specific operations** (e.g., retrieving or updating one invoice quickly).
* In the **CC model**, each customer is stored as one document with all invoices embedded inside.
  + To fetch all invoices of a customer, MongoDB retrieves just **one document** (the customer record).
  + This makes CC better for **customer history queries** (e.g., analyzing all purchases by a customer).

**Performance observation:**

* **Reads and deletes** are faster in CC (fewer documents to scan).
* **Invoice-level operations** (like working with one invoice) are faster in TX.
* **Updates and creates** show similar performance.
* On **local MongoDB**, all operations run in <1 ms. On **Atlas cluster**, network latency increases times (~60–130 ms), but the relative trend remains the same.

Therefore, **TX is optimized for invoice-centric workloads, while CC is optimized for customer-centric analytics**. Both approaches have trade-offs, and the choice depends on the type of queries required.

Github Repo Link :- <https://github.com/Roshan-21k/mlops2025w_142502024>