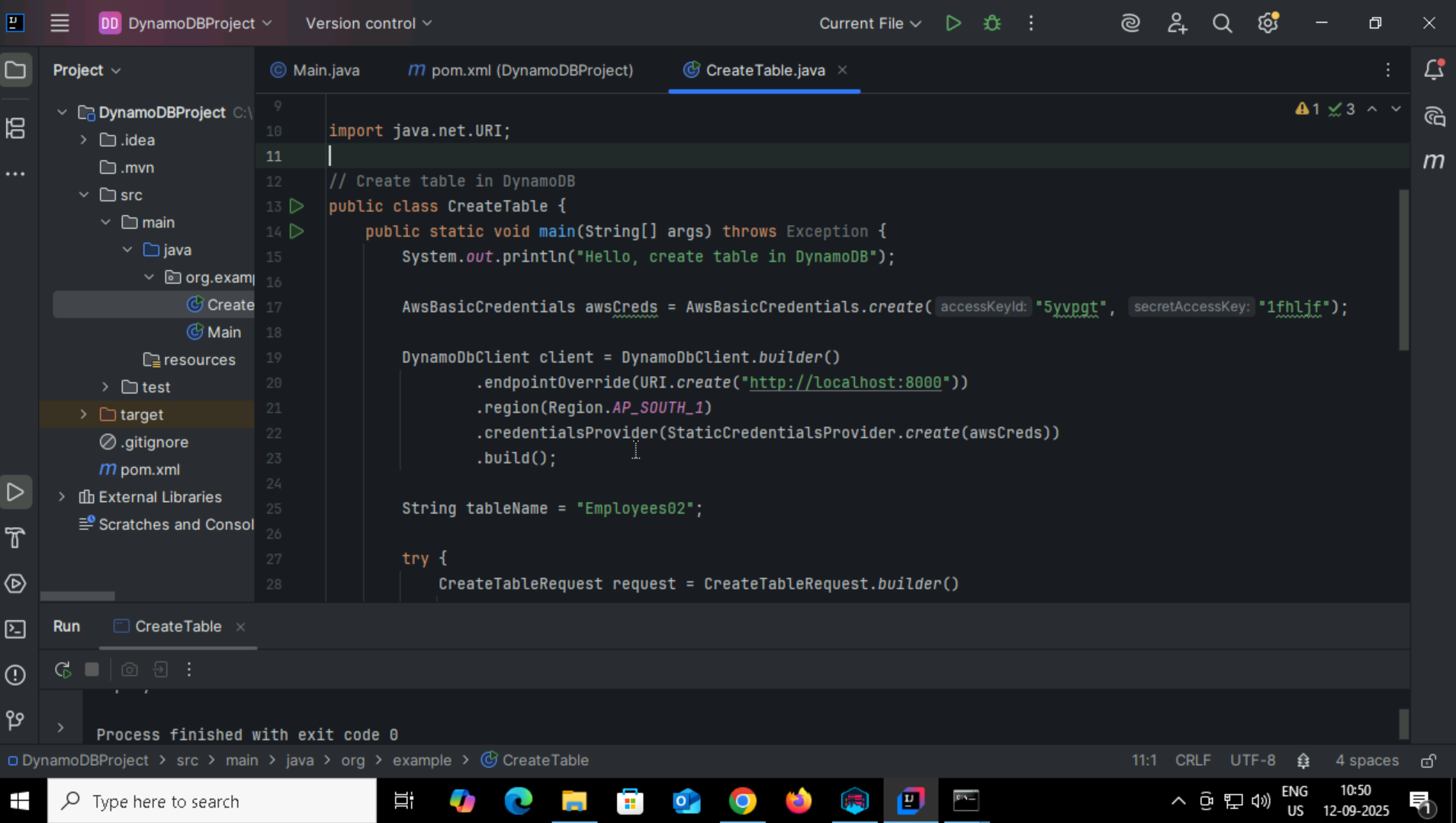
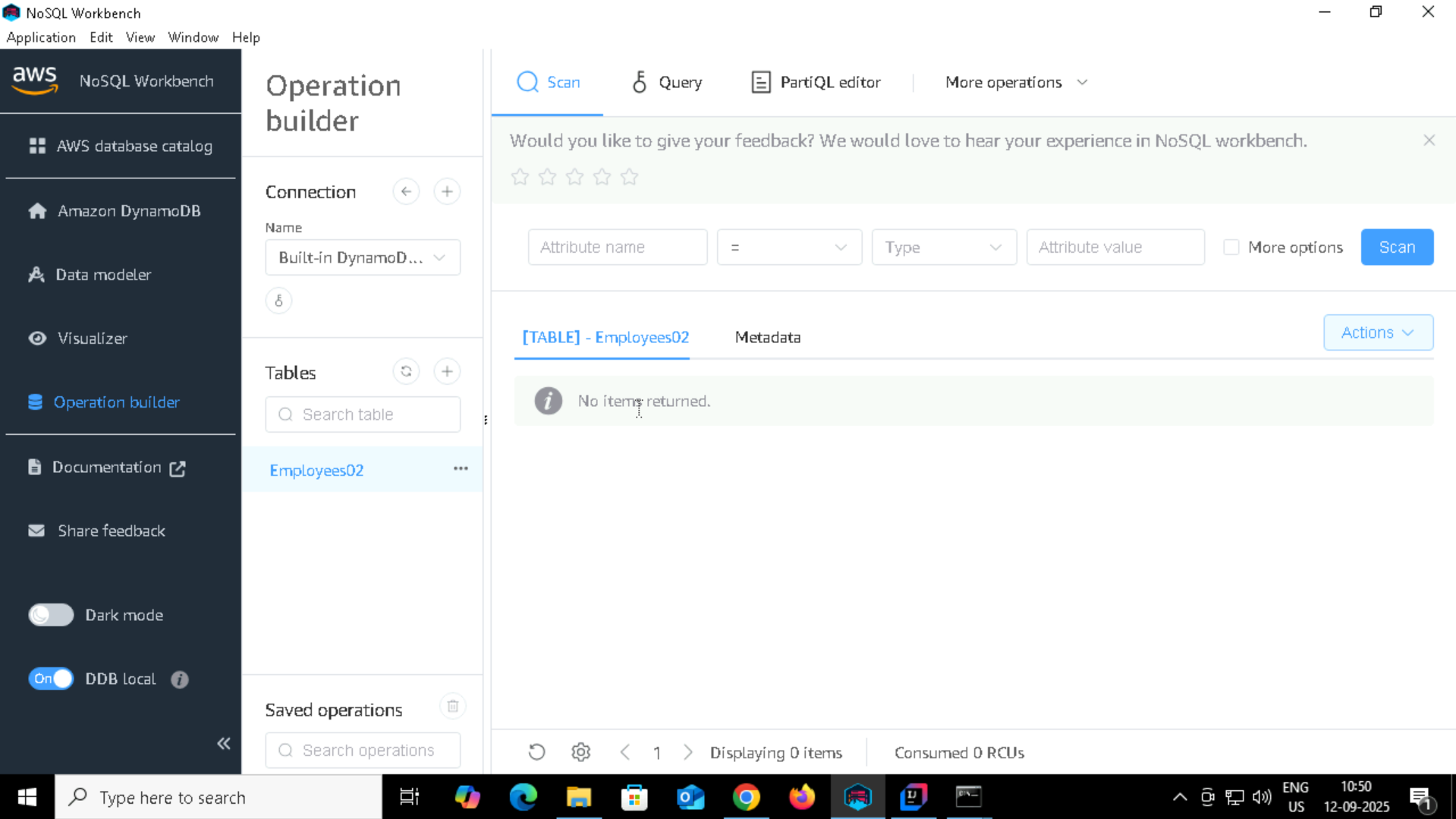
Task32:  
Task 01:

Create a table using Java code and check if the table is created.



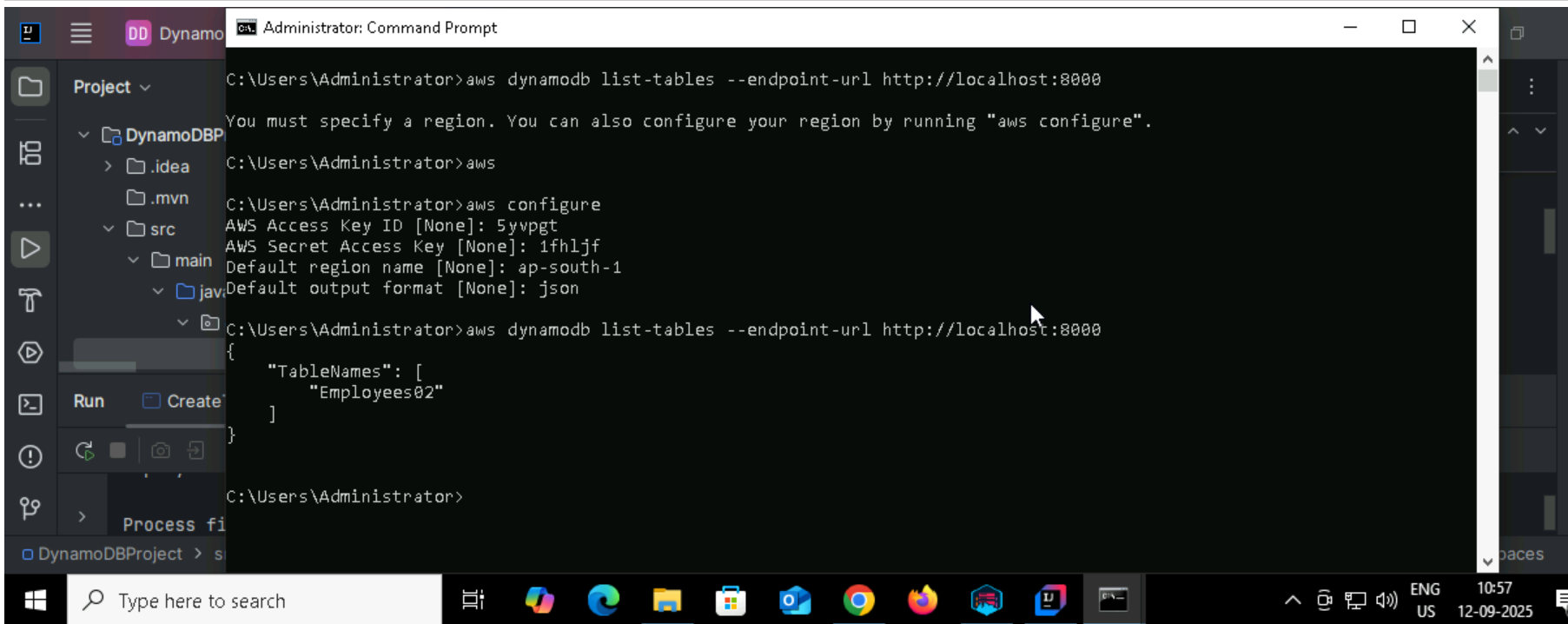


Task 02:

Using the same above java code

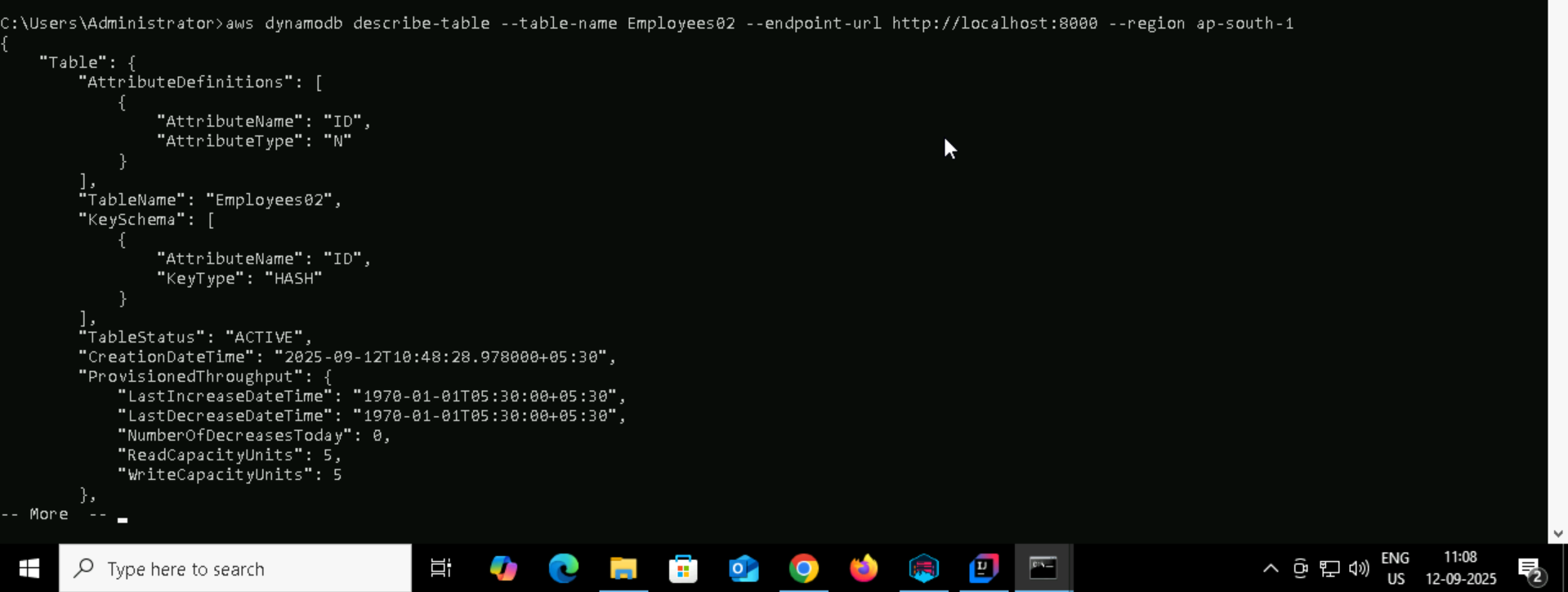
Change the post no and table name

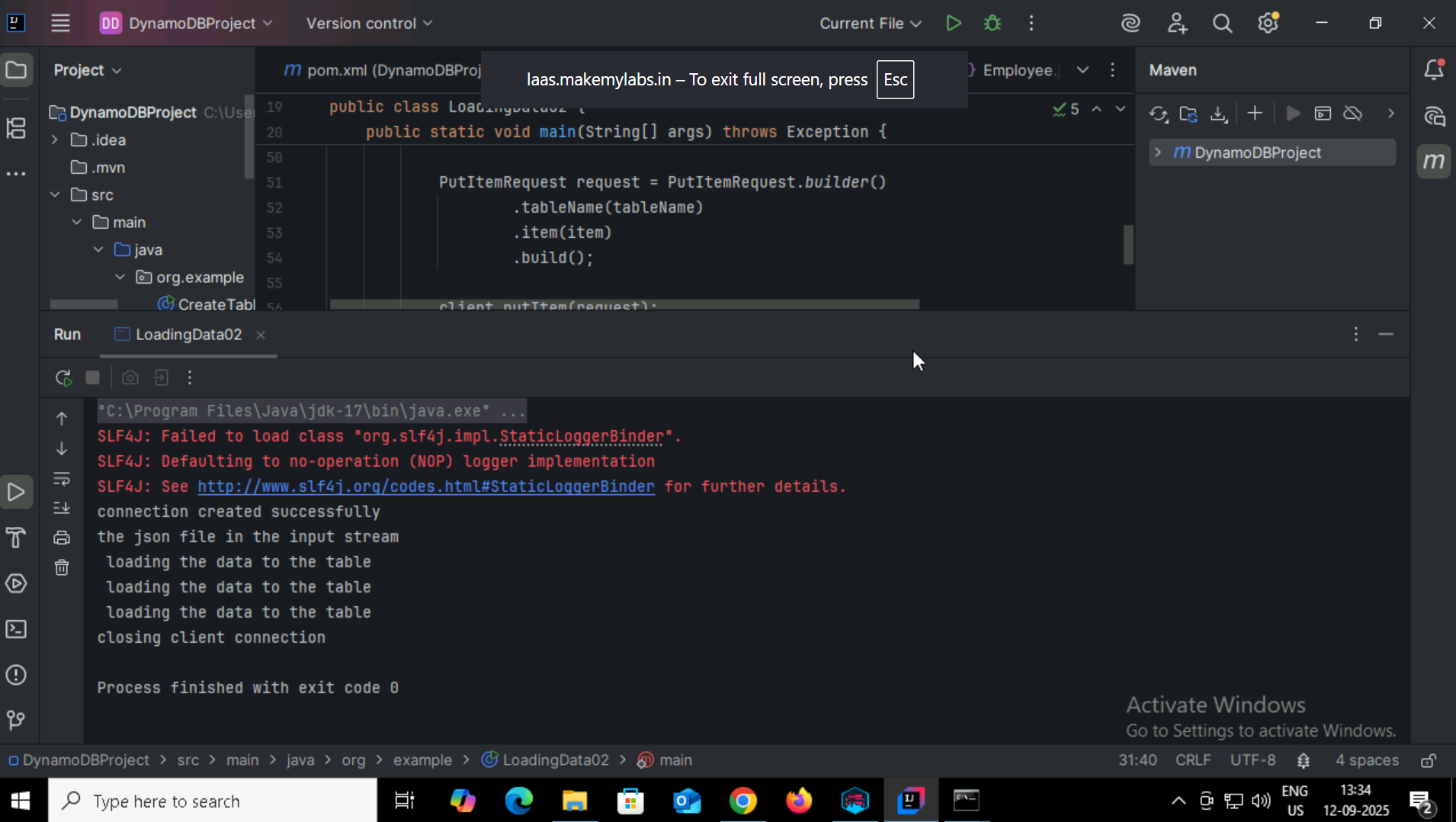
To see the table reflecting in your cli prompt:

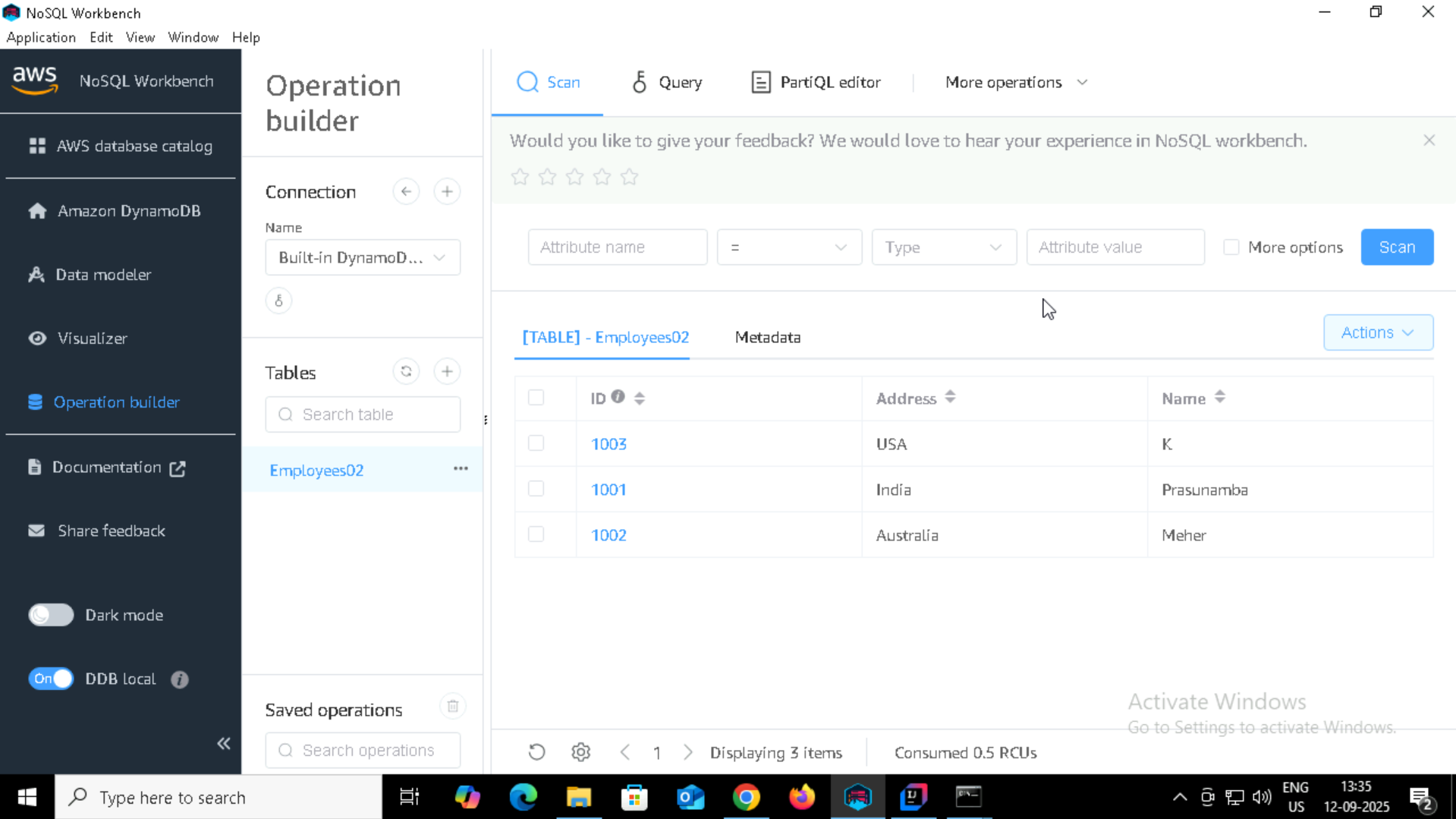
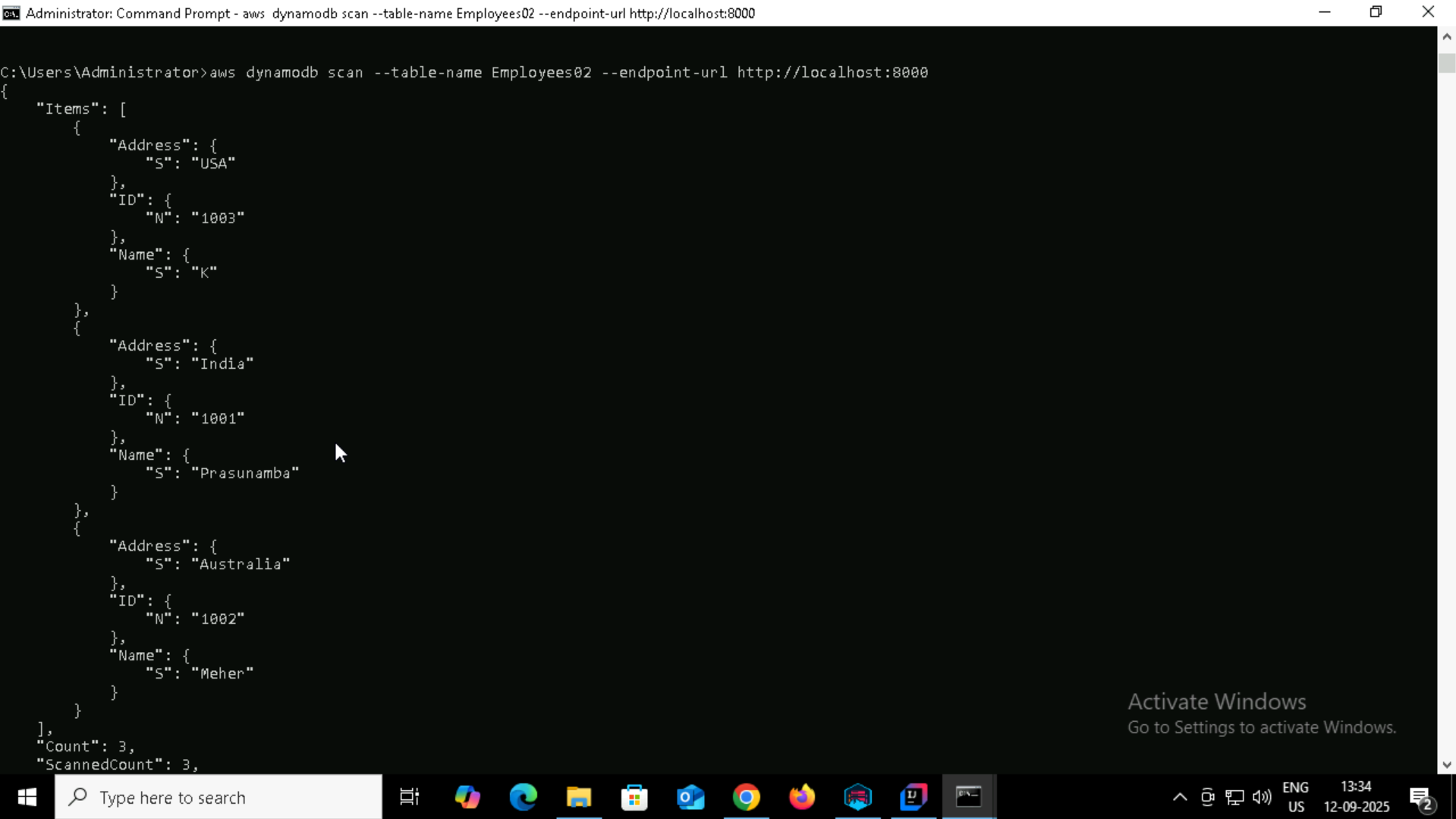


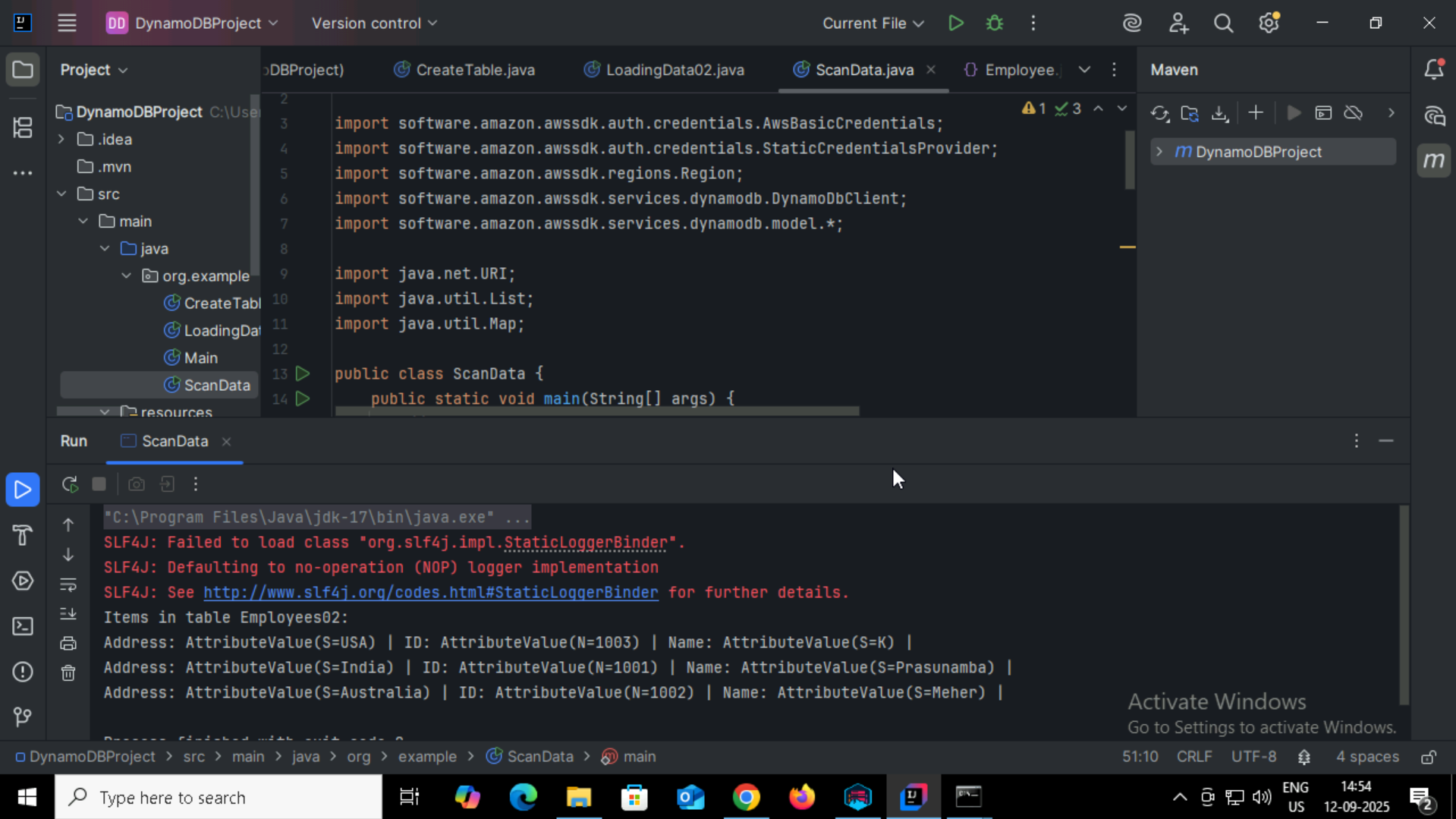
Task 03:

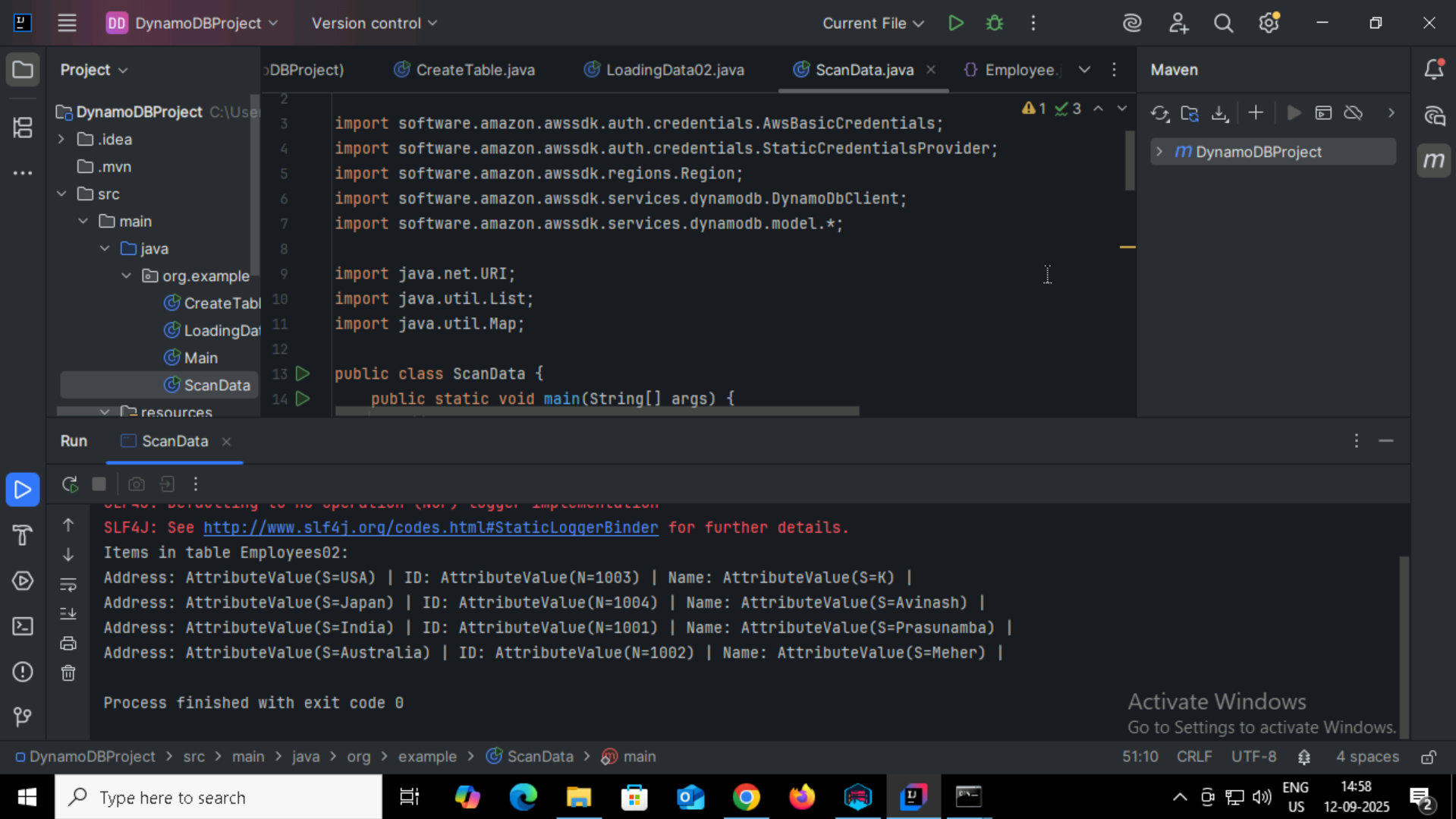
For the same above tables see the description of the tables..

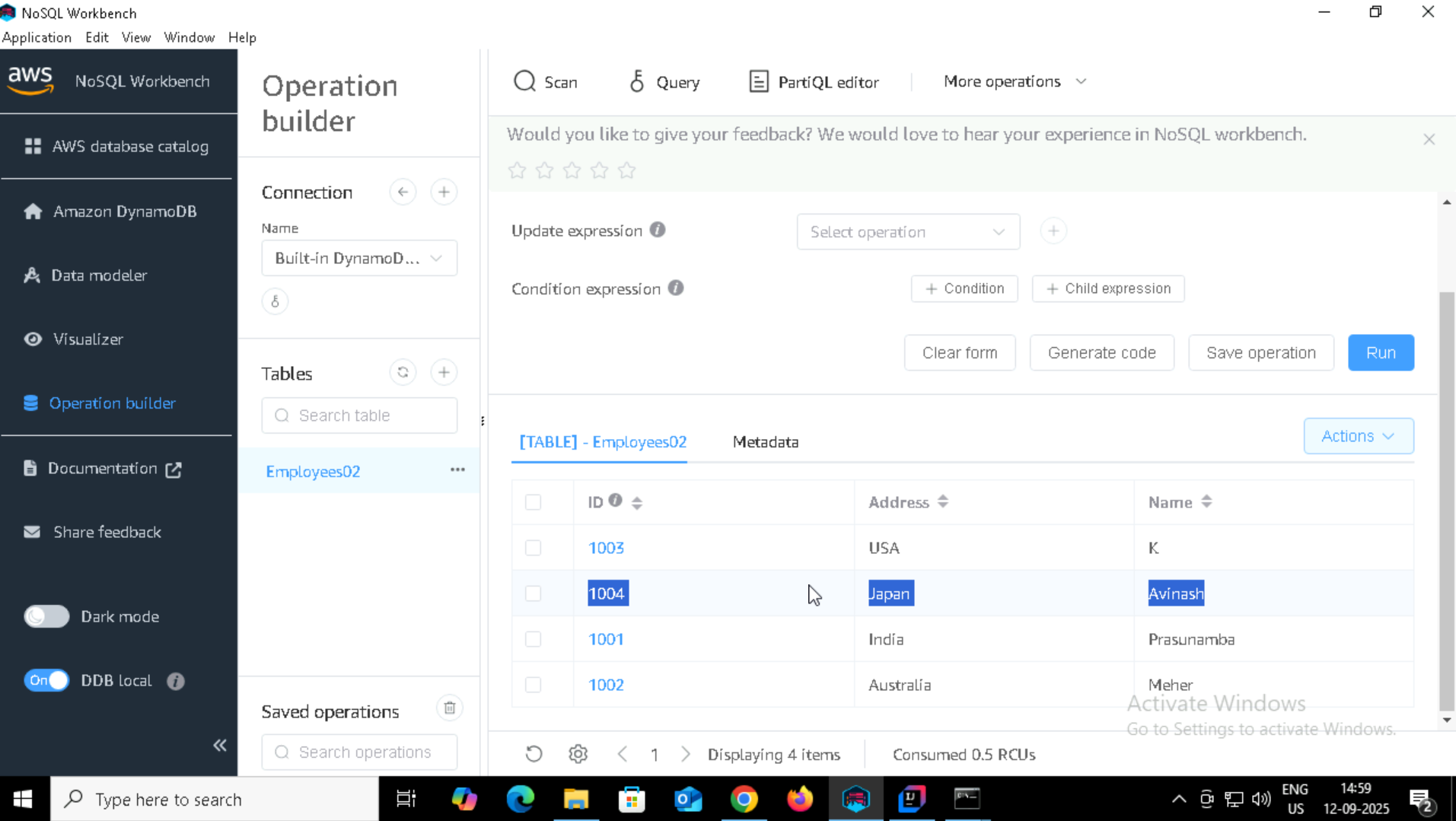


Task004:  


  
  
Task005:  


Task006:  


Task007:  




Task 07:

How do you lower the cost of DynomaDB..explain ways to do so…

1. Choose the Right Capacity Mode

DynamoDB offers two capacity modes:

On-Demand: You pay per read/write request. Good for unpredictable traffic but more expensive for high, consistent workloads.

Provisioned: You set read/write capacity units (RCUs/WCUs). Cheaper if traffic is predictable.

Tip: Switch from on-demand to provisioned if you have predictable traffic patterns.

2. Use Auto Scaling

For provisioned mode, enable Auto Scaling:

It automatically adjusts your RCUs/WCUs based on usage.

Prevents over-provisioning (wasting money) or under-provisioning (causing throttling).

3. Optimize Read and Write Patterns

Use eventually consistent reads instead of strongly consistent reads (halves read costs).

Batch operations: BatchGetItem and BatchWriteItem reduce the number of API calls.

Avoid hot partitions: Design partition keys to distribute traffic evenly. Hot partitions can lead to throttling and higher costs.

4. Use DynamoDB Accelerator (DAX)

DAX is a caching layer for DynamoDB.

It reduces read operations to the database by serving requests from cache, lowering read costs significantly for high-read workloads.

5. Reduce Storage Costs

Use smaller attribute sizes where possible. Fewer bytes = lower storage cost.

Delete old/unneeded data regularly using TTL (Time to Live) feature.

Consider storing large objects (like images) in S3 instead of DynamoDB.

6. Use Global Tables Wisely

Multi-region replication can be expensive.

Only replicate tables across regions if needed.

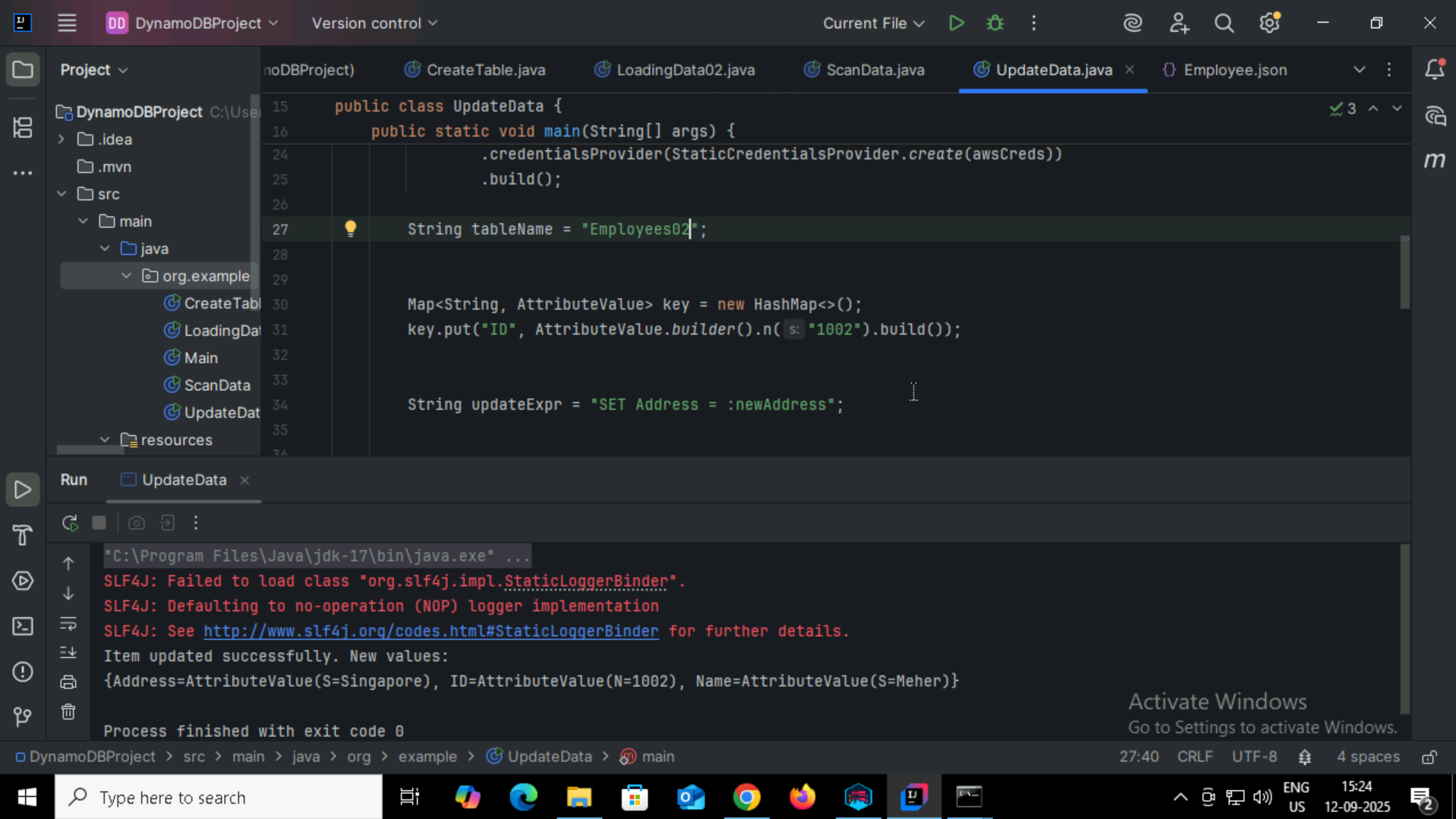
7. Monitor and Analyze Usage

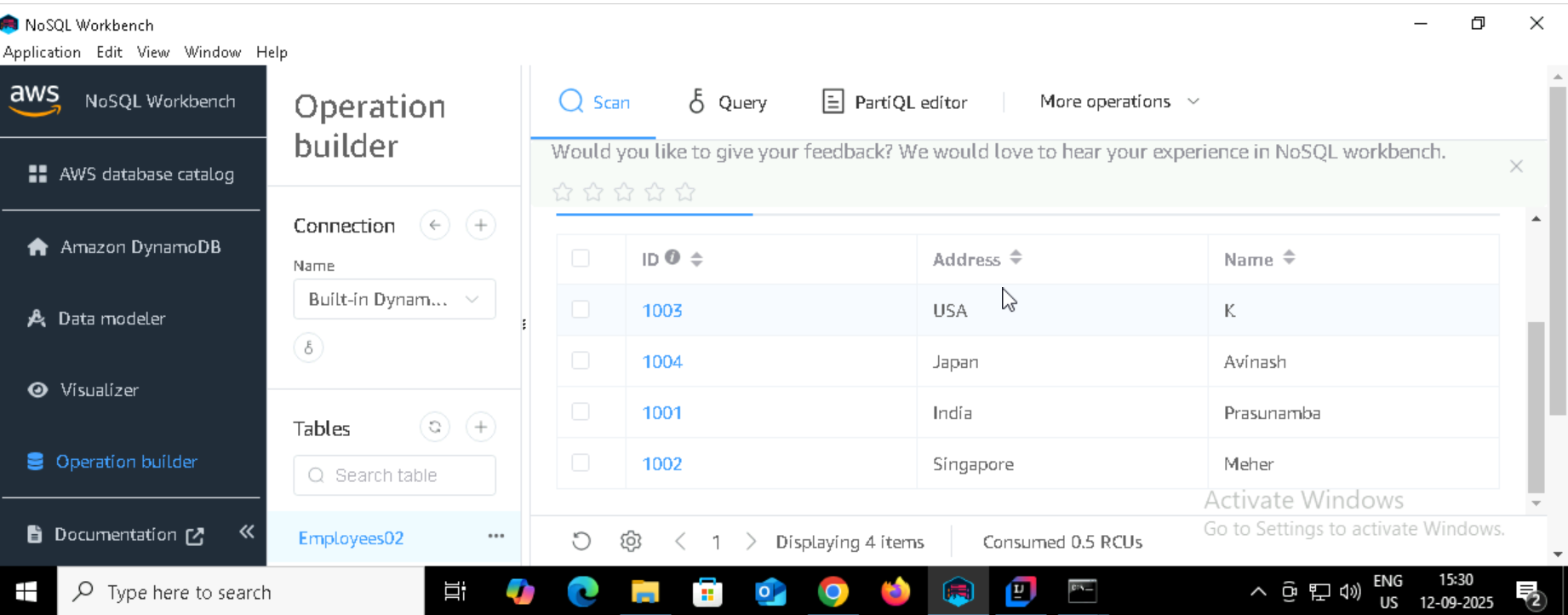
Use CloudWatch metrics and AWS Cost Explorer.

Identify unused tables, over-provisioned capacity, or inefficient queries.

Task 08:

Update item details using java code.. And check if it reflects in the server..





Task 09:

Delete a particular item from the table …

 1002…

