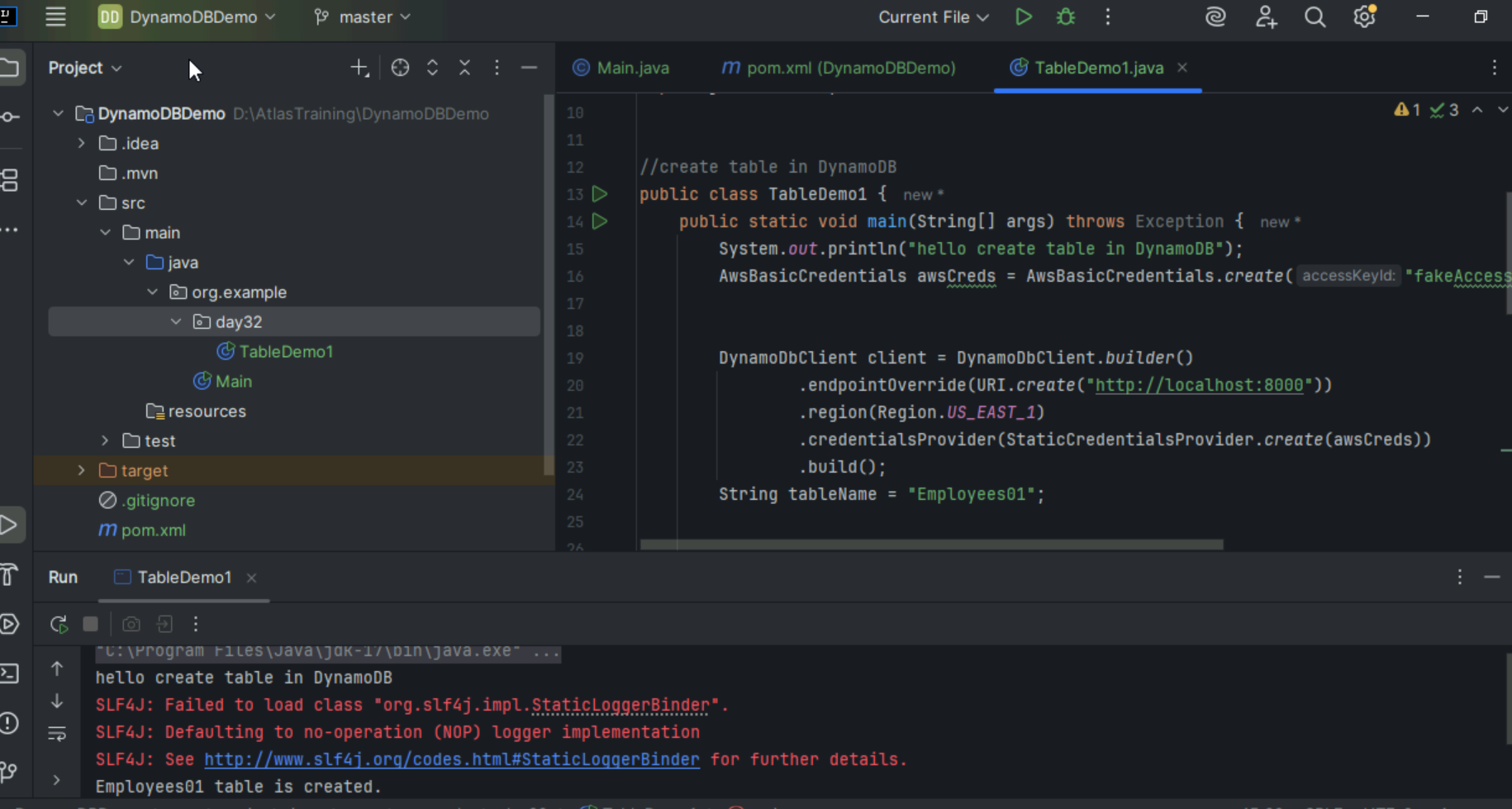
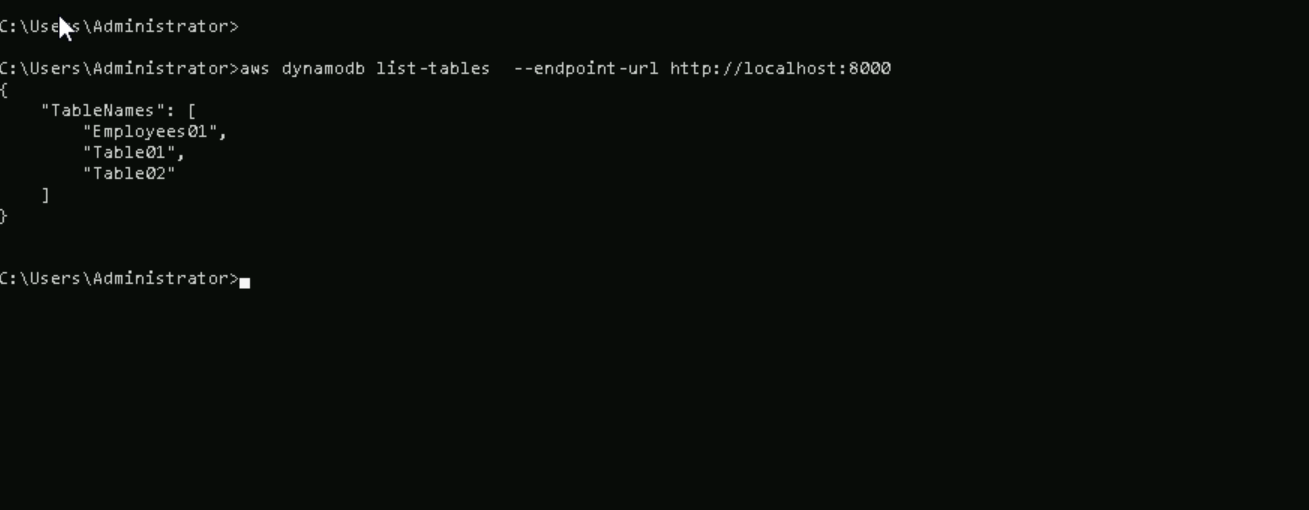
Day 32 - 104608492 - Shirisha Perapagu

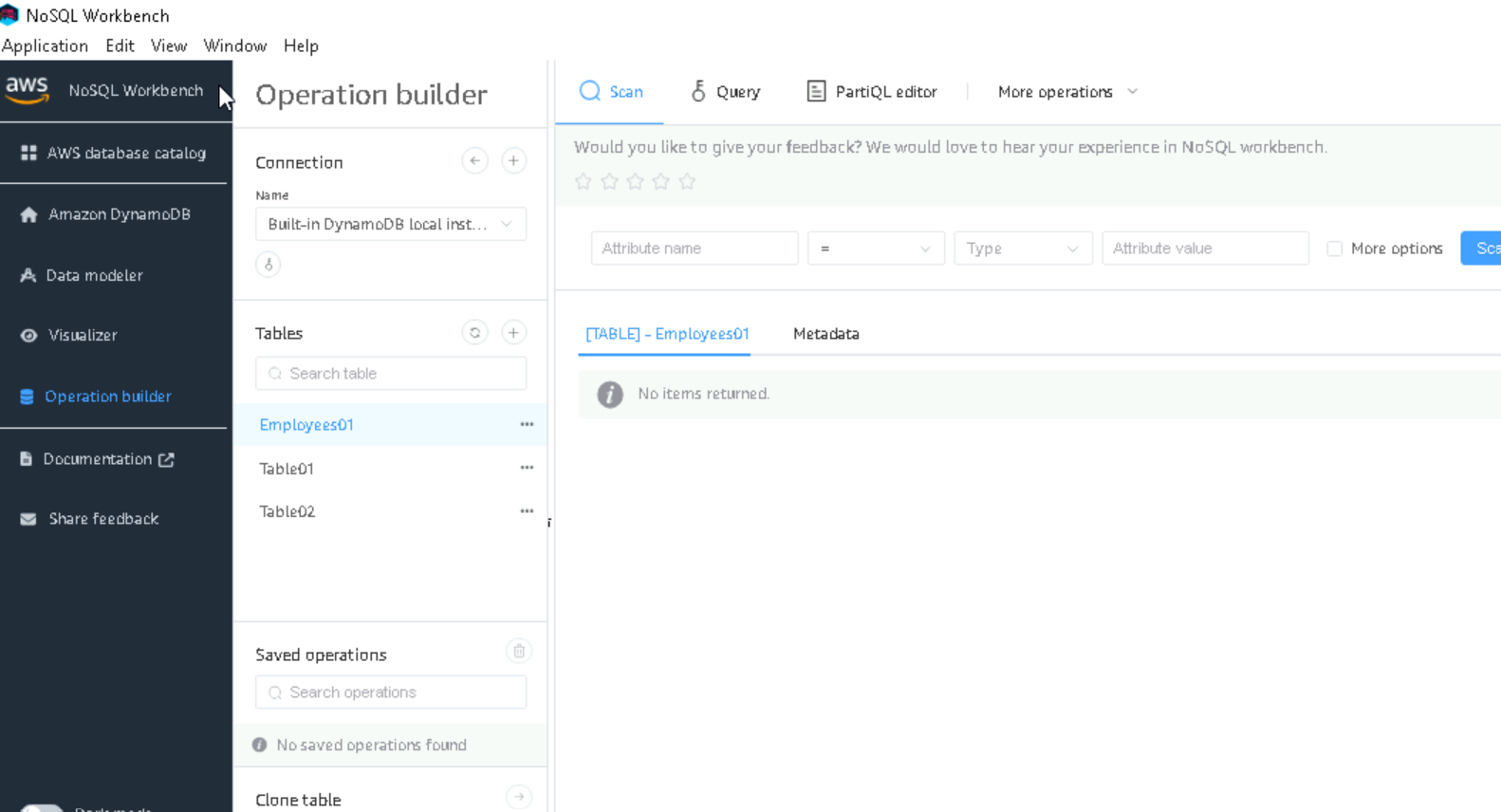
AWS DynamoDB - Part 3

Task 1

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

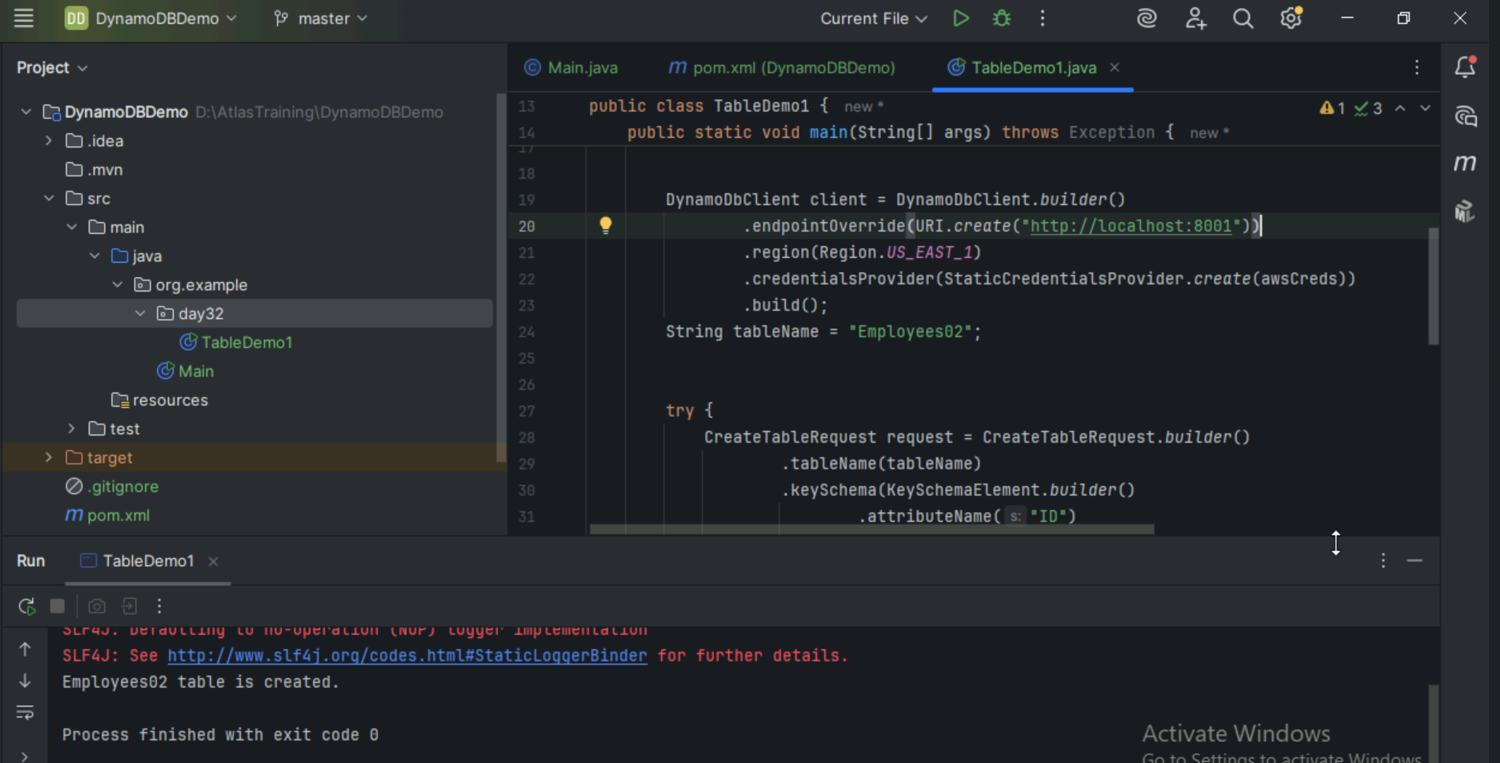


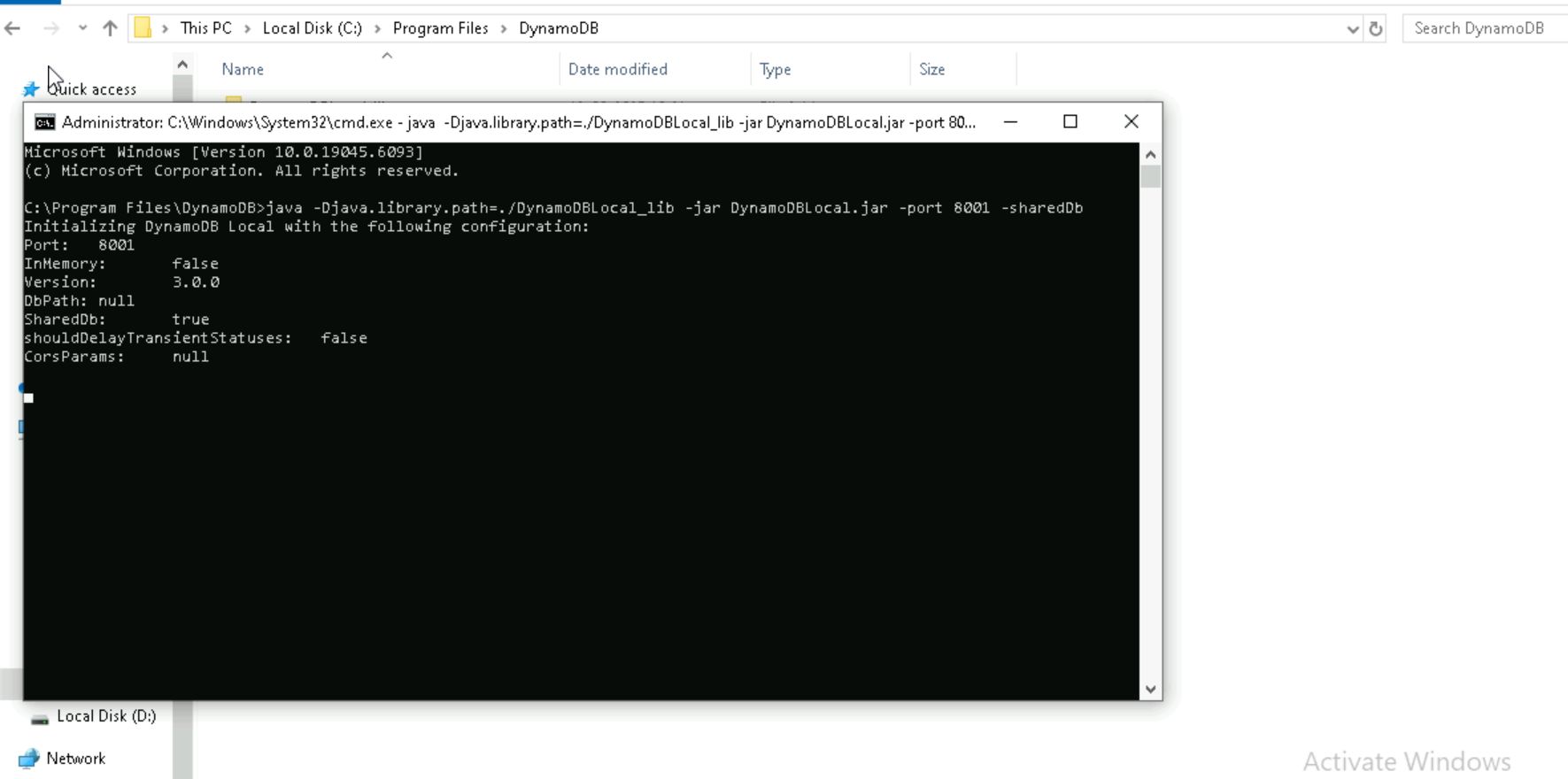


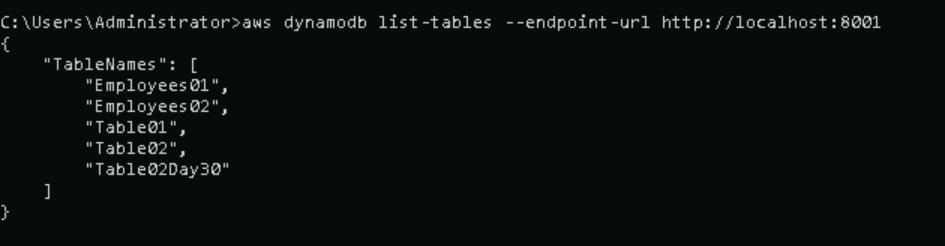


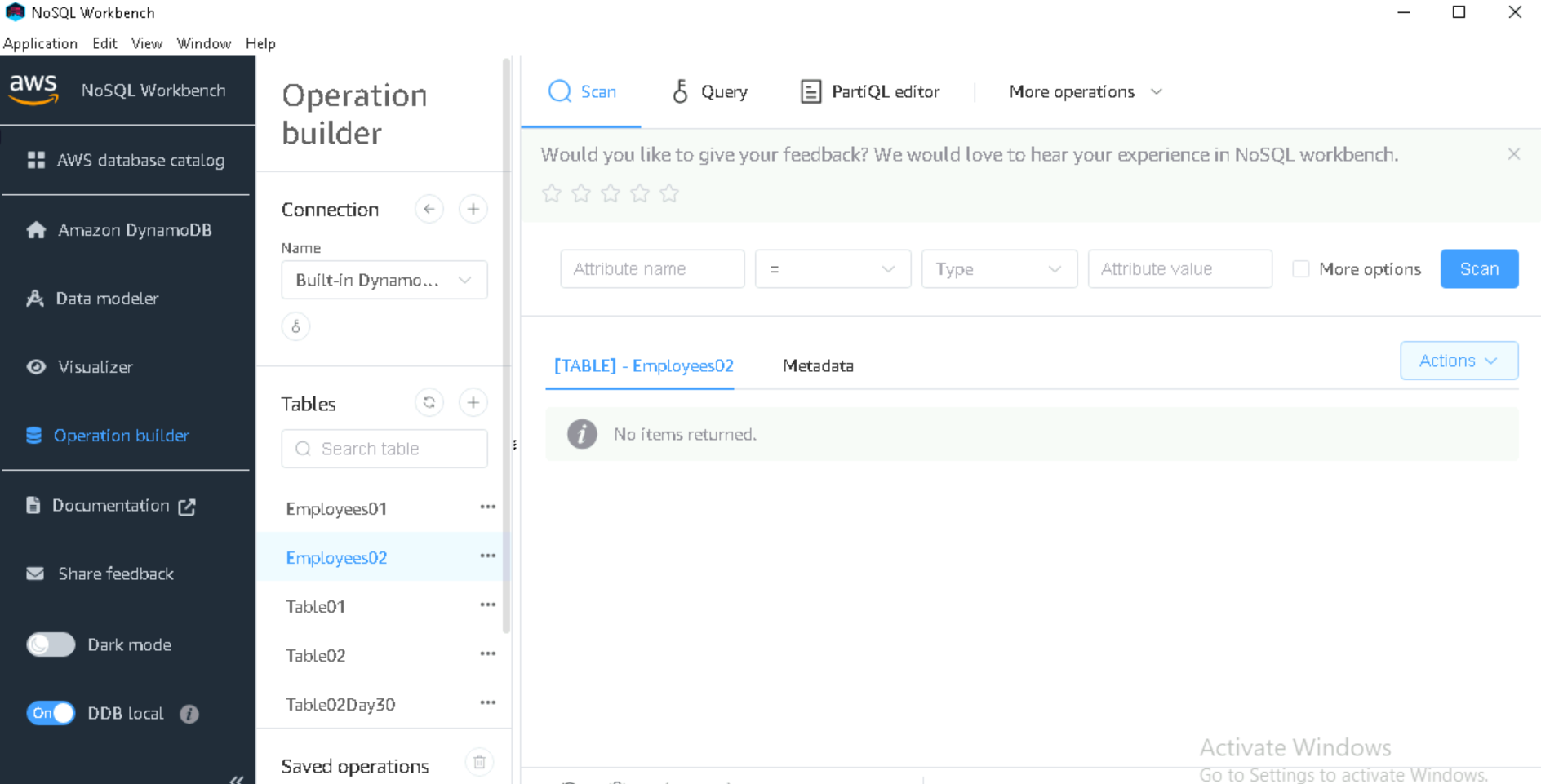
Task 2

Using the same above java code.  Change the port no and table name .To see the table reflecting in your cli prompt:



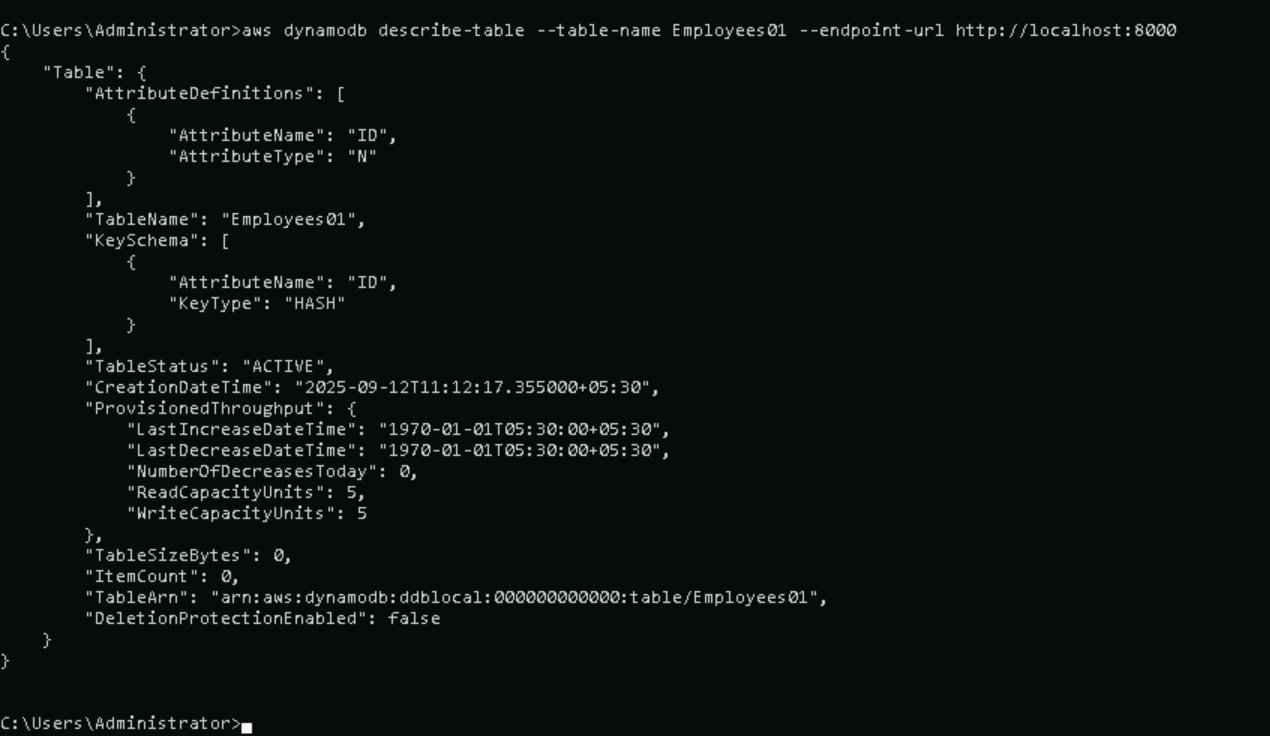


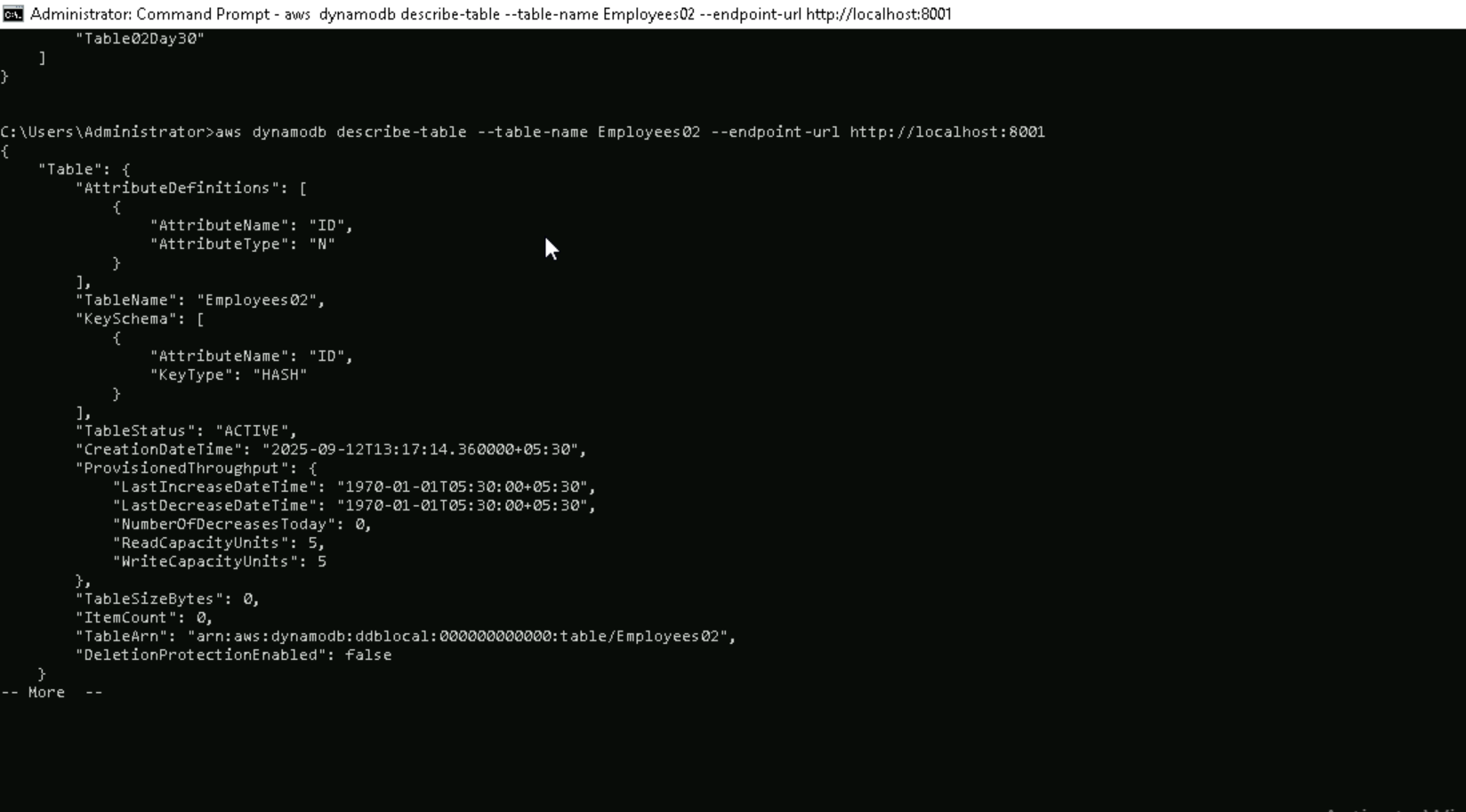




Task 3

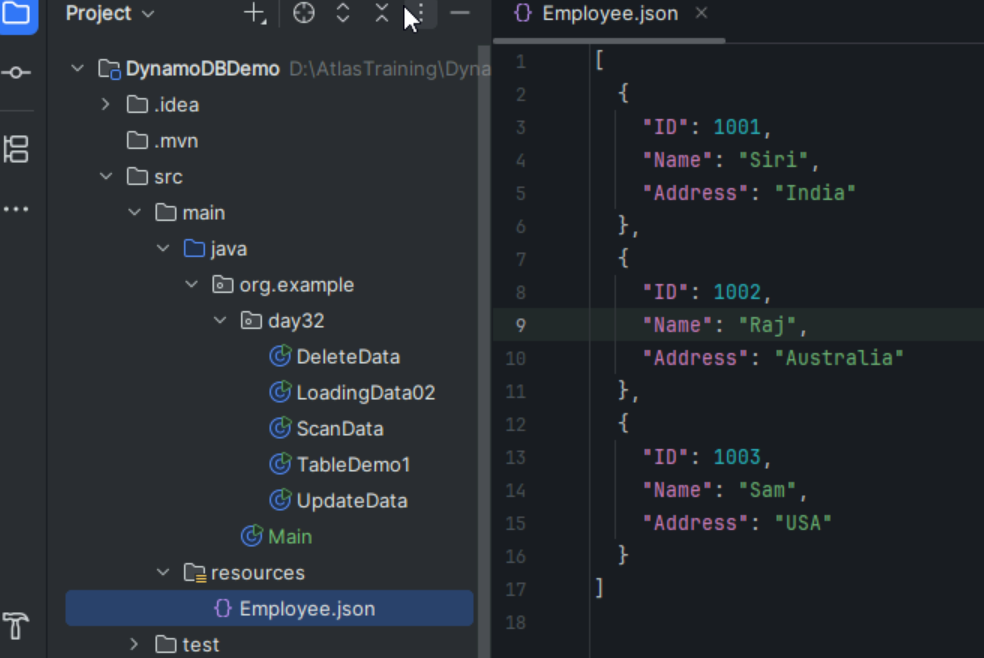
For the same above tables see the description of the tables..

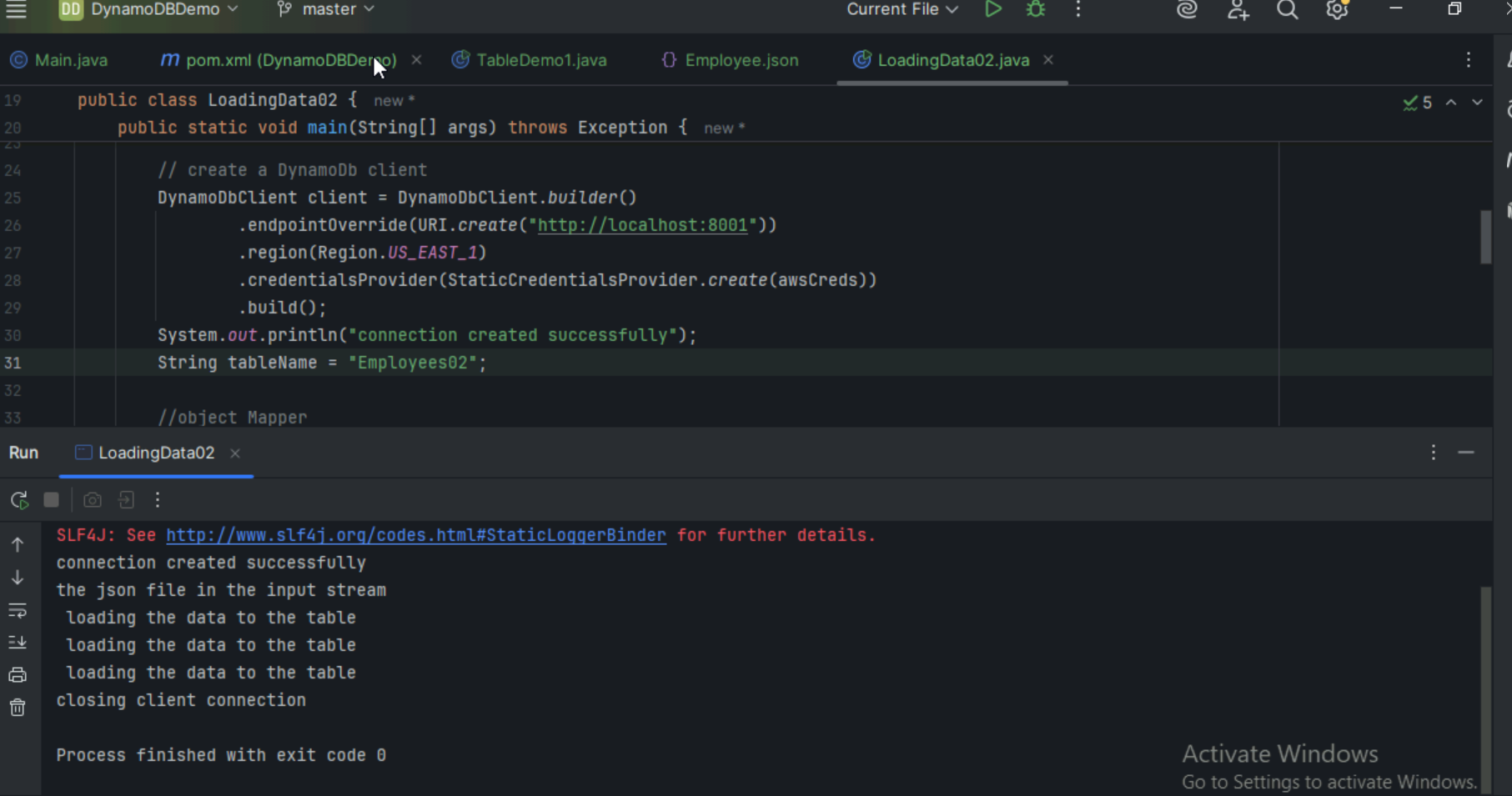


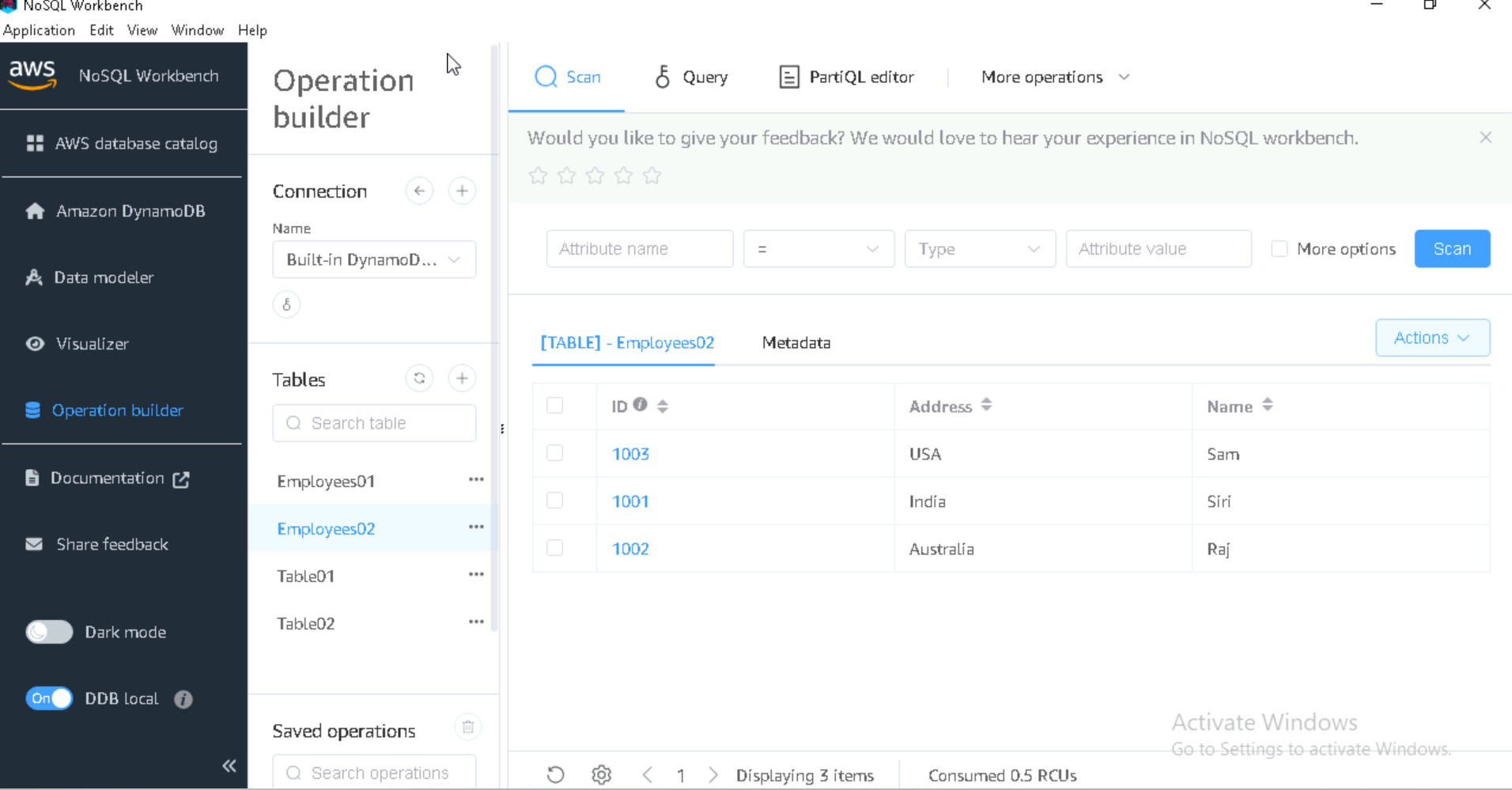


Task 4

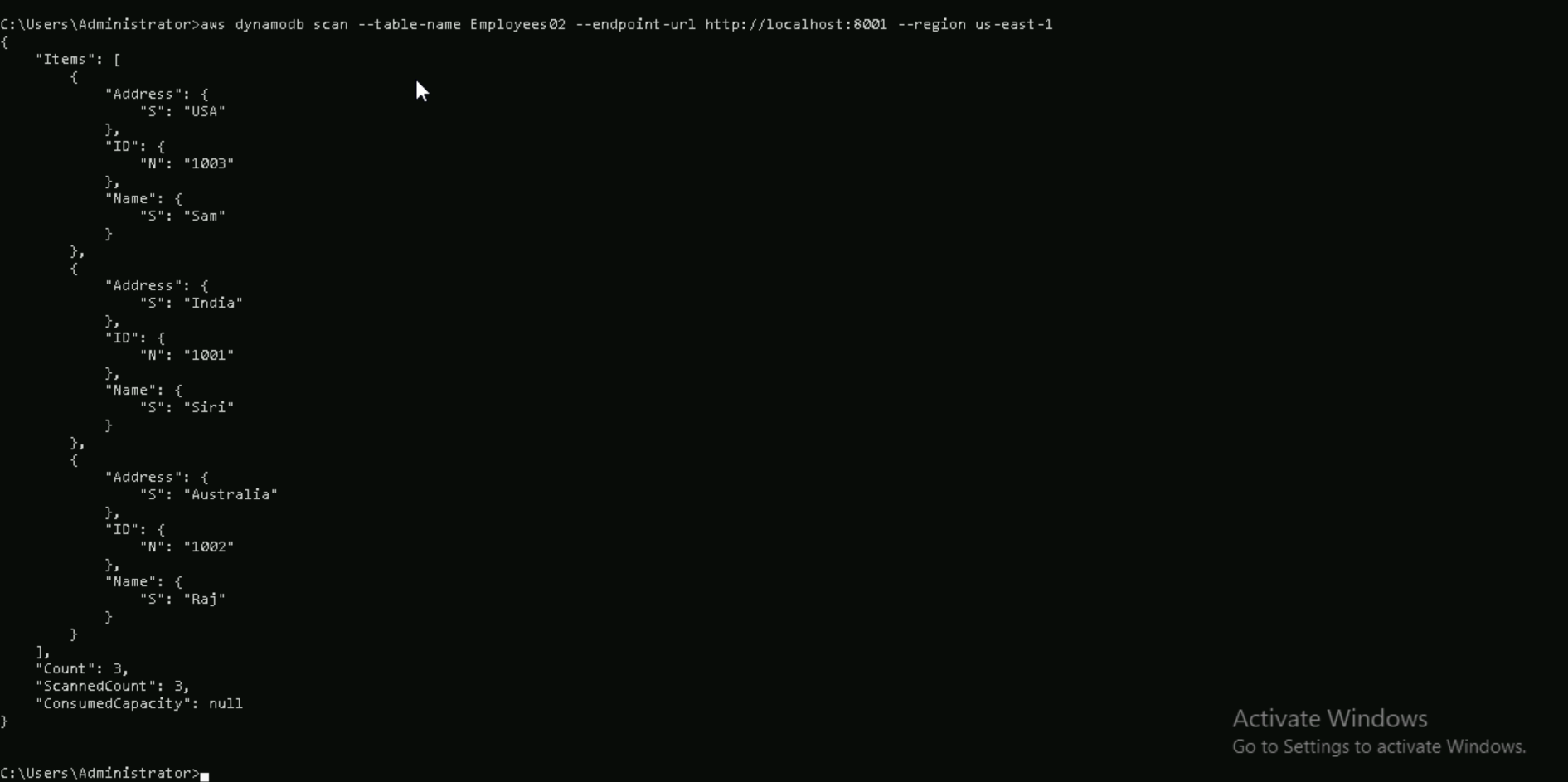
Loading / inserting data to the table





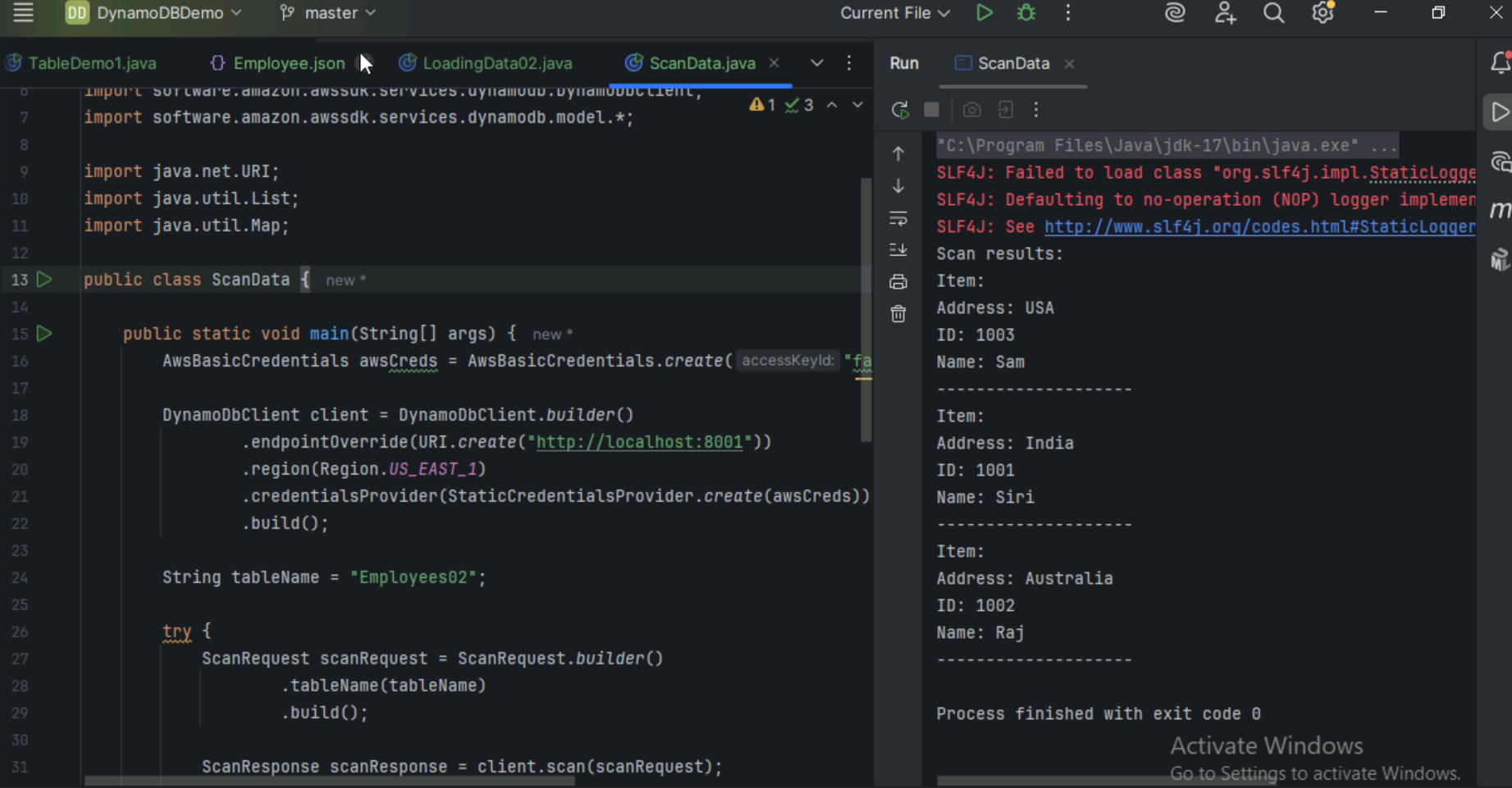


Cmd scan



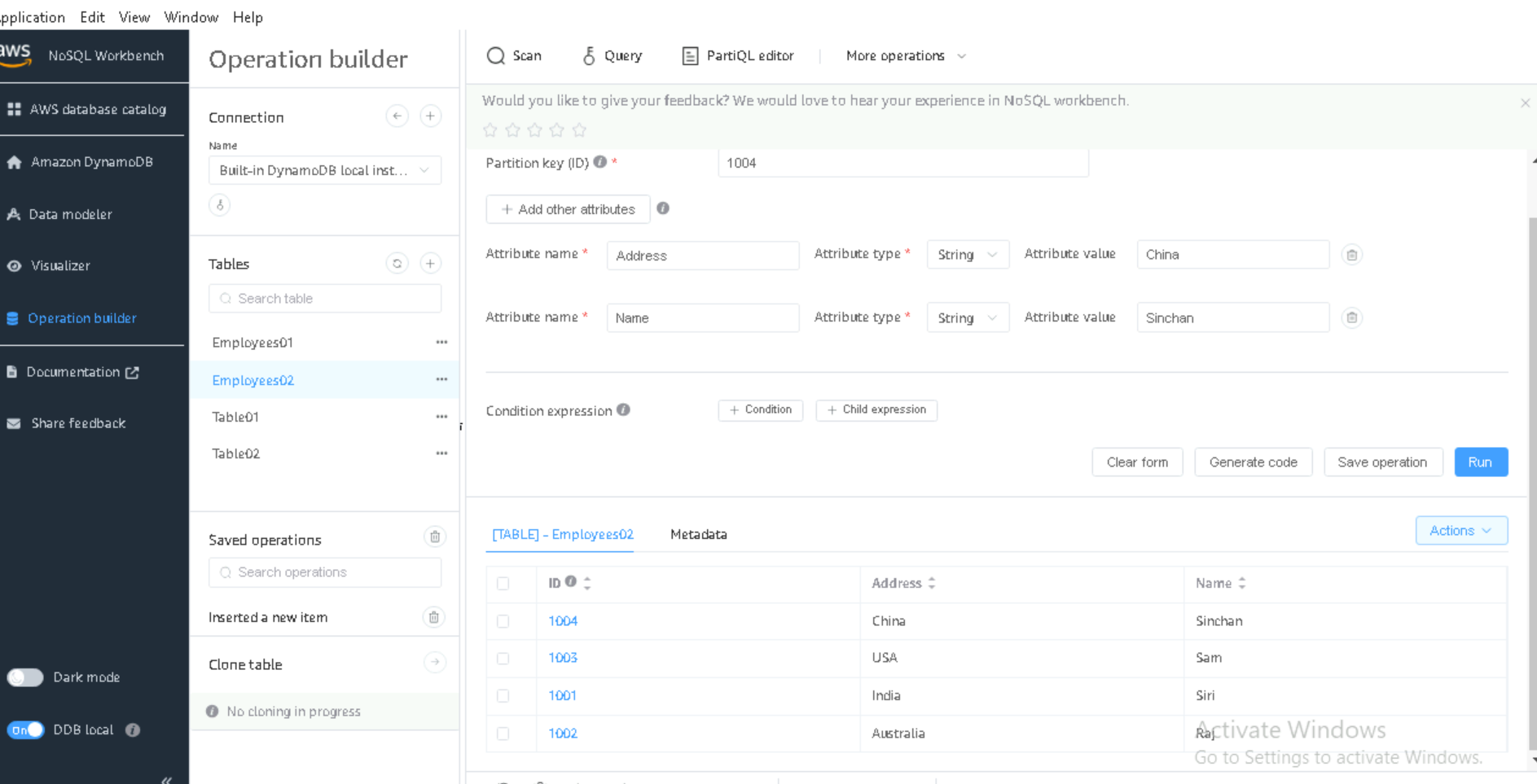
Task 5

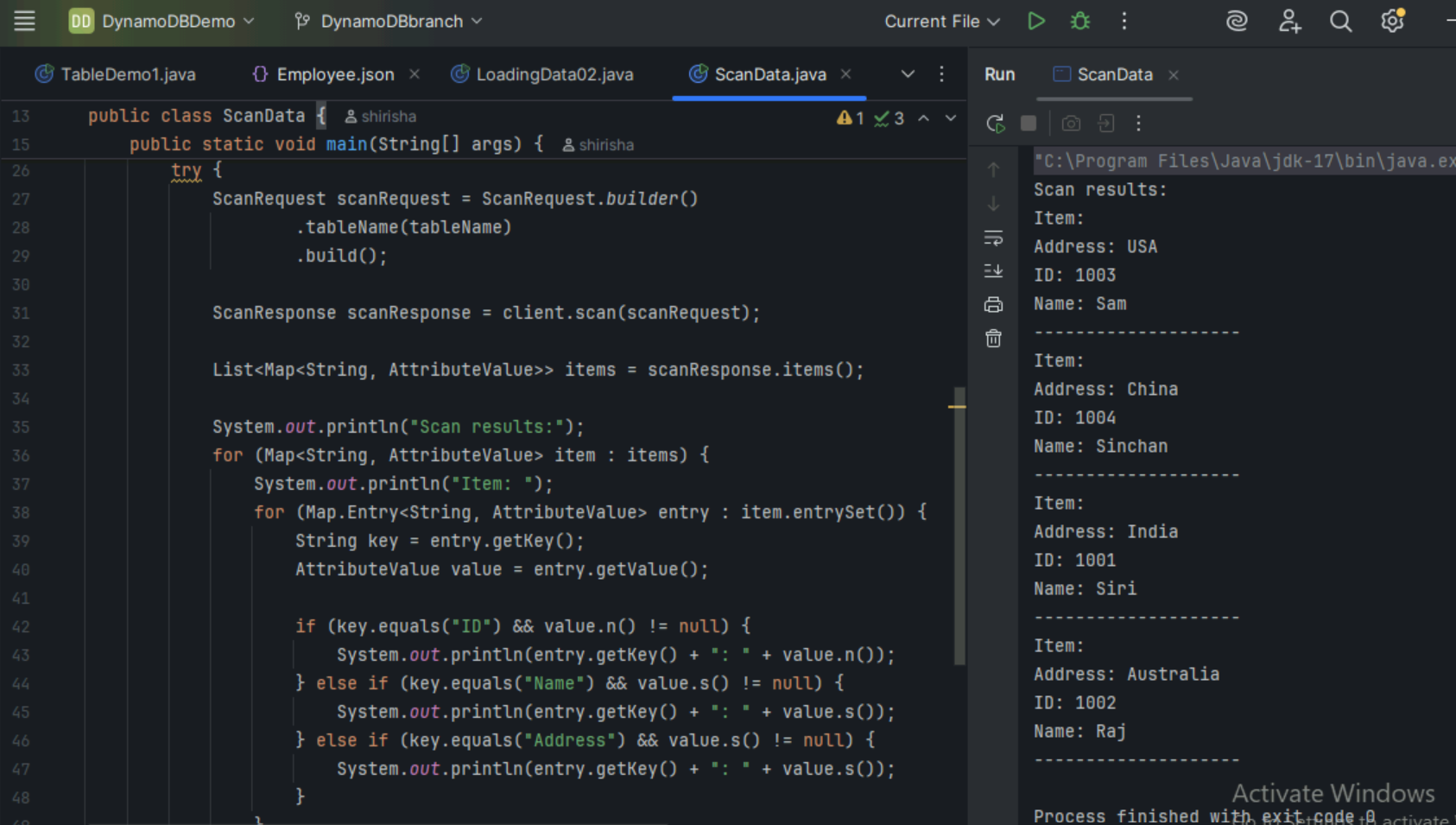
Reading / Scanning the data from the server.. Using java code..



Task 6

Insert an item in the nosql workbench.. And check if the item reflects in your java output…





Task 7

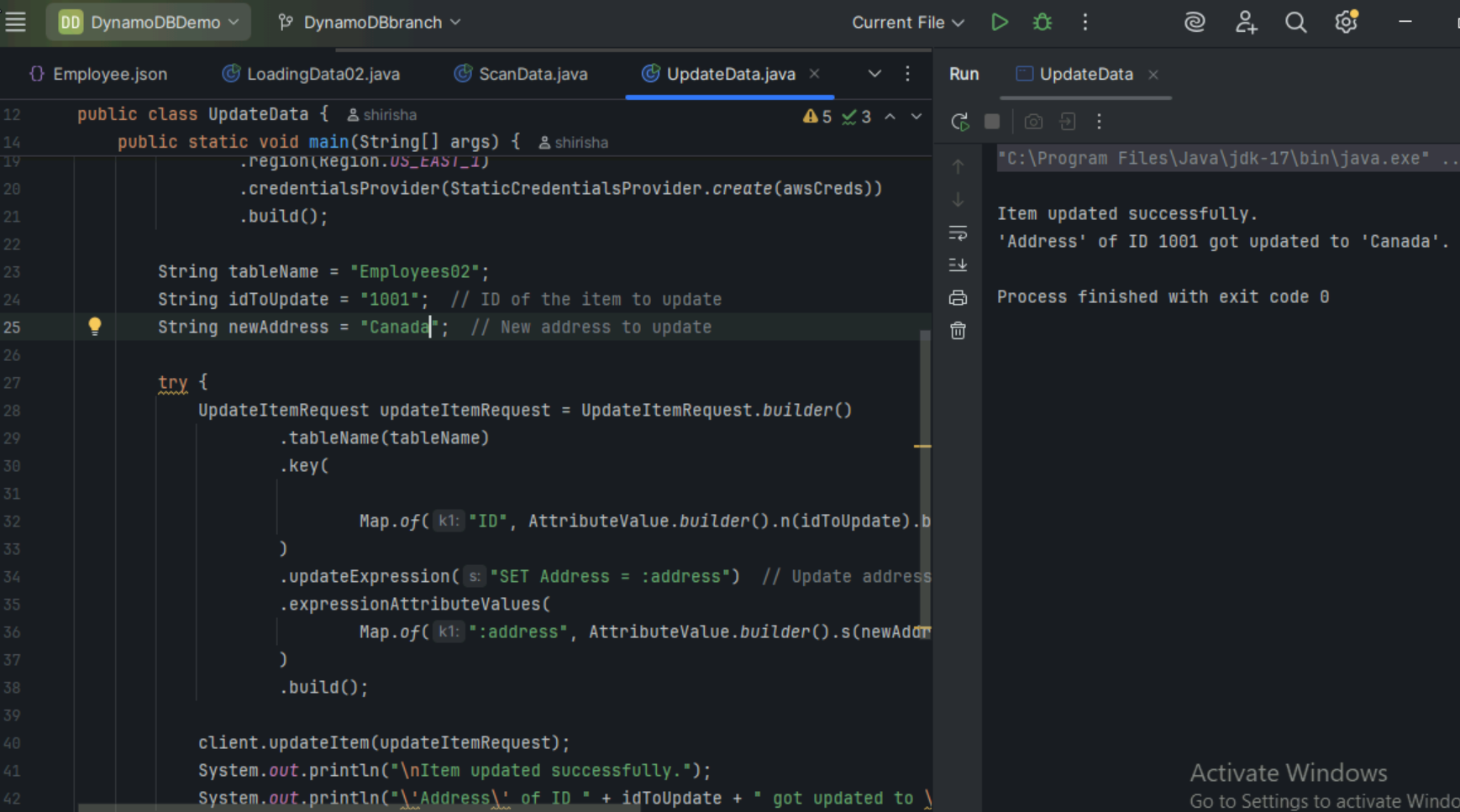
How do you lower the cost of DynomaDB..explain ways to do so…

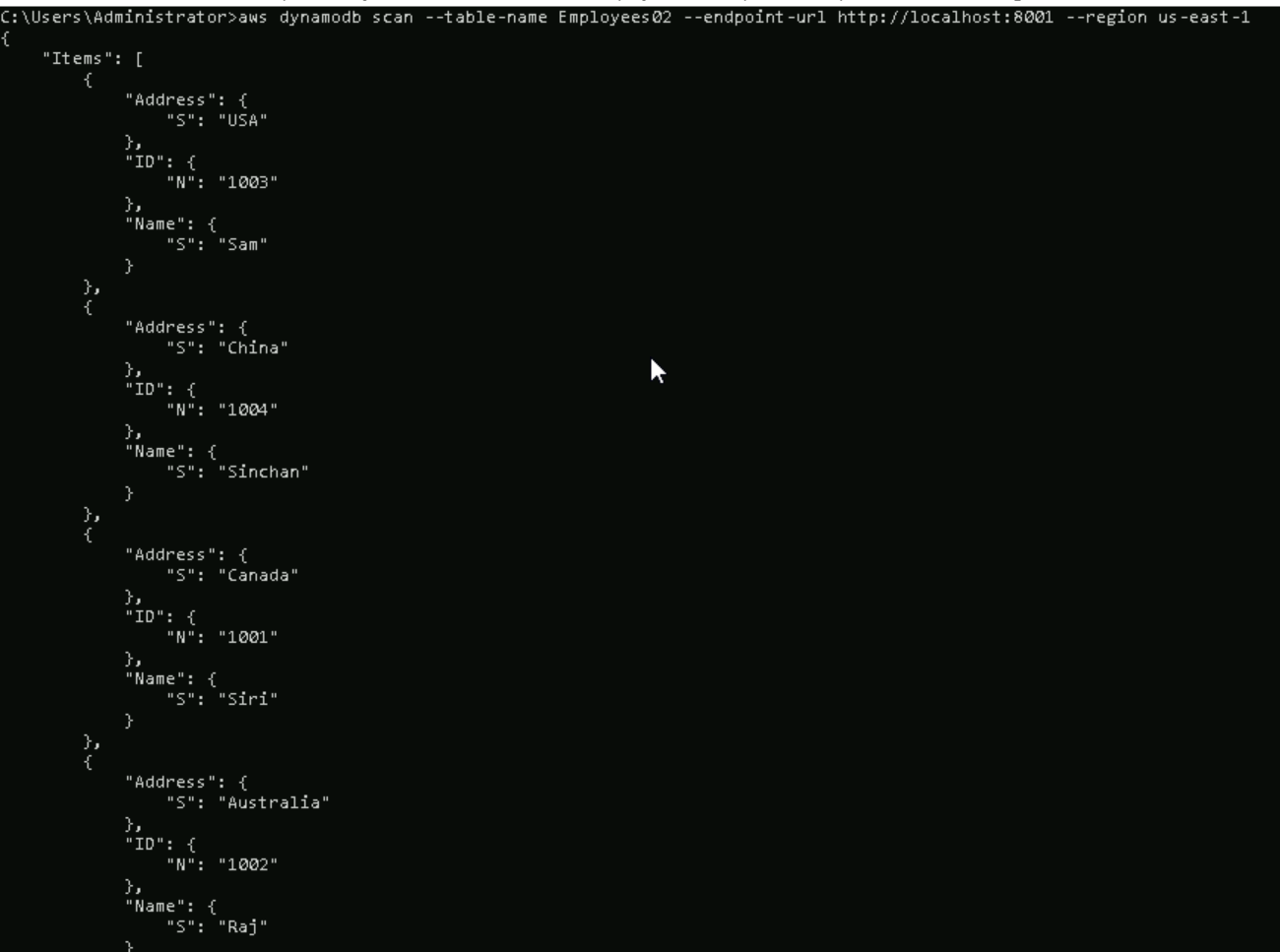
To lower the cost of DynamoDB, we can

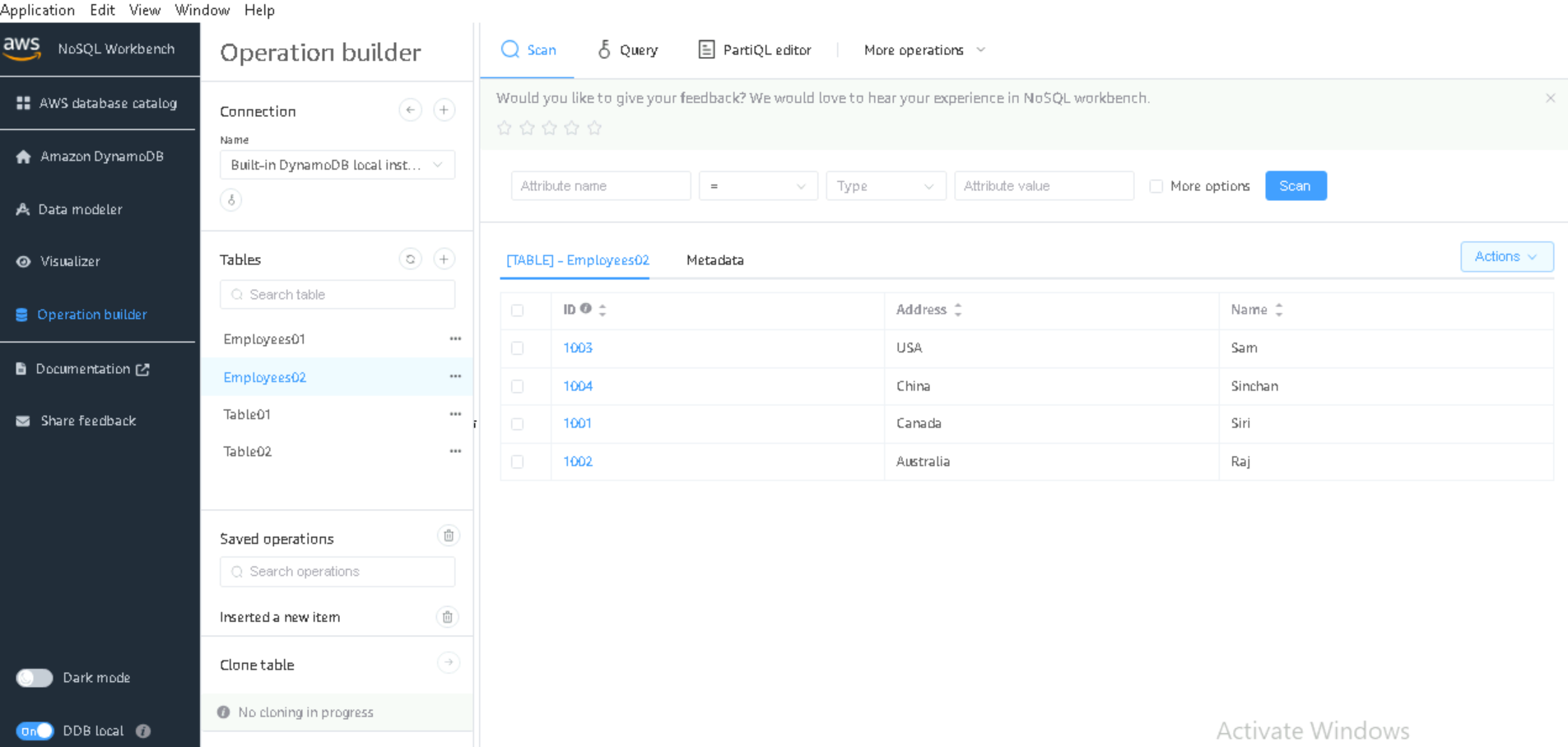
1. **Use On-Demand Pricing** only when needed and switch to **Provisioned Capacity** for predictable workloads.
2. **Optimize Queries** by using efficient keys (partition and sort) to avoid full table scans.
3. **Use Global Secondary Indexes (GSIs)** wisely and delete unused indexes.
4. **Enable Auto Scaling** to adjust throughput based on demand.
5. **Store data efficiently** by archiving old records or using TTL (Time-to-Live) for automatic data expiration.

Task 8

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



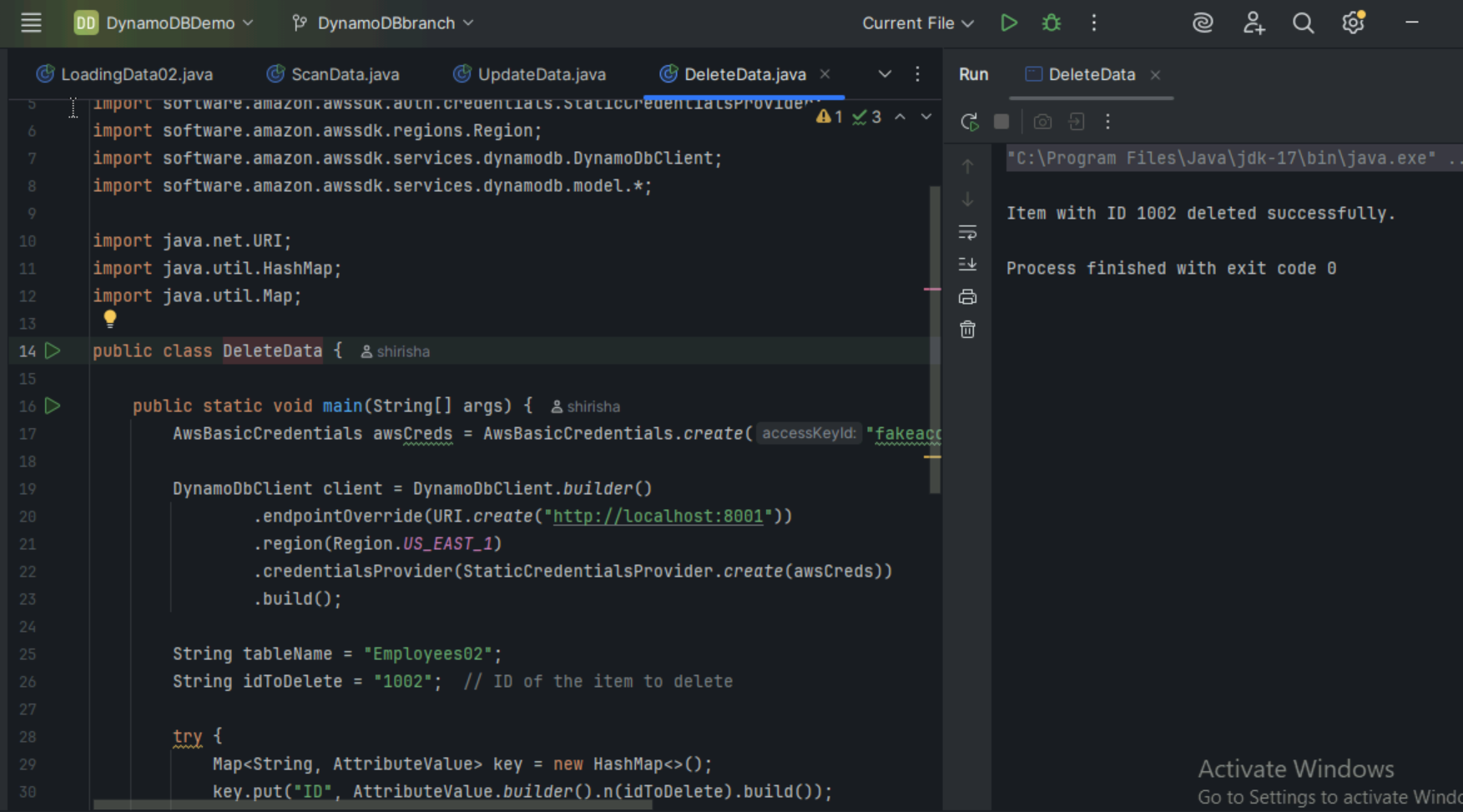


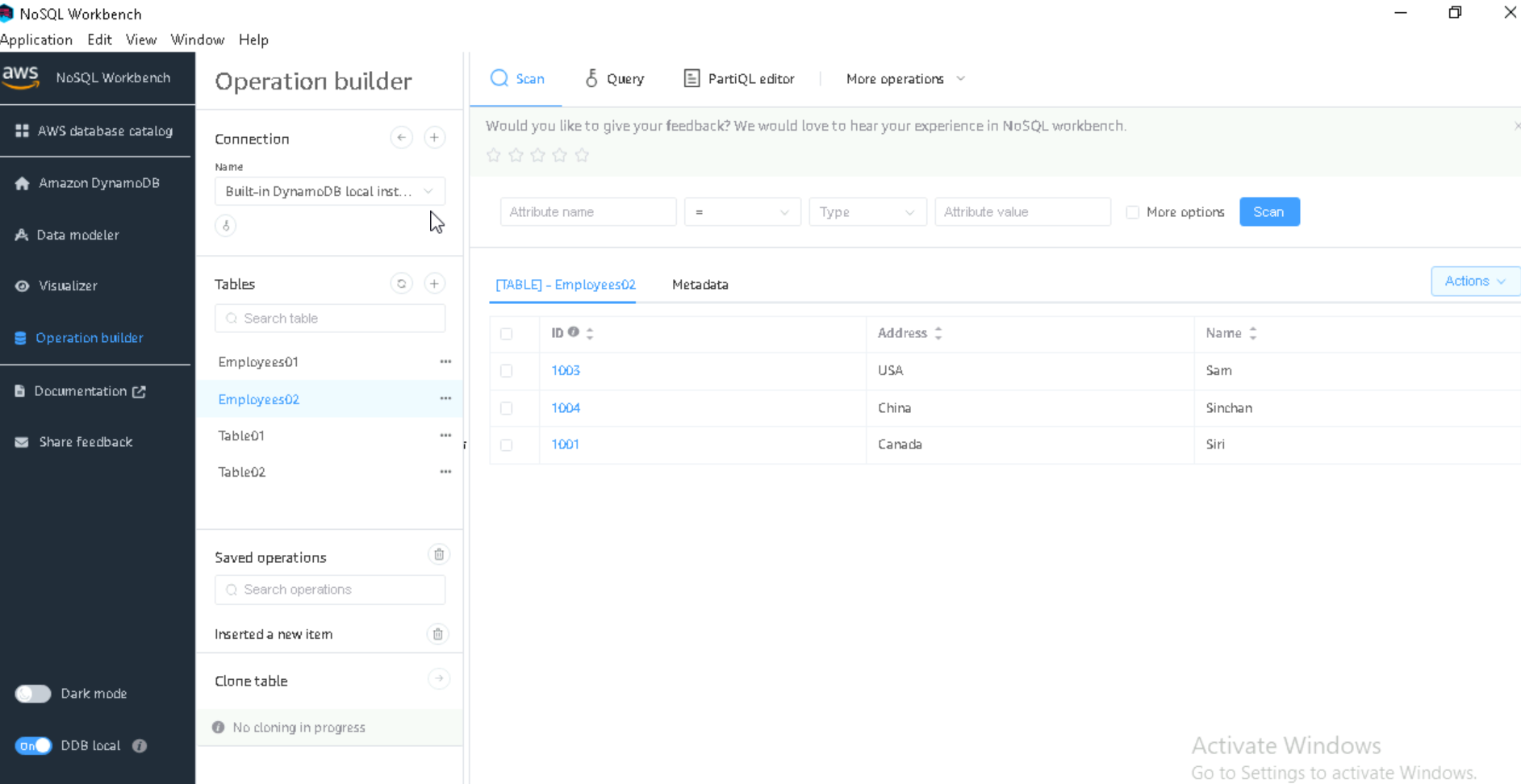


Task 9

Delete a particular item from the table …id

1002…





===============================================================

Home Tasks

================================================================

Task 01:

What are the features of DynamoDB?

DynamoDB is a fully managed NoSQL database provided by AWS with features.

**Scalability**: Automatically handles large data and traffic.

**High Availability**: Data is always available with replication.

**Low Latency**: Fast response times for real-time use.

**Flexible Schema**: No fixed structure for data.

**Fully Managed**: AWS takes care of maintenance and scaling.

Task 02:

What are the advantages and disadvantages of Dynamodb

Advantages of DynamoDB:

**Fully Managed**: No need for manual management of servers or databases.

**Scalable**: Automatically scales to handle large volumes of data and traffic.

**Low Latency**: Provides fast read and write performance.

**High Availability**: Built-in replication across multiple regions for fault tolerance.

**Flexible Schema**: Allows easy storage of structured and unstructured data.

**Integrated with AWS**: Seamlessly integrates with other AWS services.

Disadvantages of DynamoDB:

**Complex Pricing**: The pricing model (read/write capacity units, storage) can be complex and costly at high scale.

**Limited Querying**: Limited querying capabilities compared to relational databases (e.g., JOINs, complex queries).

**Size Limitations**: Item size is limited to 400 KB, which can be restrictive for some use cases.

**Eventual Consistency**: By default, read operations are eventually consistent.

**No Multi-table Joins:** Unlike relational databases, DynamoDB does not support multi-table joins.

Task 03:

Where do we use dynamoDB(the uses cases of DynamoDb)

**1.Web & Mobile Apps**: Store user data and sessions.

**2.Gaming**: Track player stats and leaderboards.

**3.IoT**: Manage data from connected devices.

**4.E-commerce**: Handle orders and inventory.

**5.Social Media**: Store posts and user content.

Task 04:

What is DynamoDBMapper?

**DynamoDBMapper** is a high-level Java library in the AWS SDK that allows us to map Java objects to DynamoDB items and vice versa, making it easier to interact with the database without writing low-level code.

Simplifies Create, Read, Update, and Delete operations on DynamoDB tables.

Uses annotations to define how Java objects are mapped to DynamoDB tables.

Supports Batch Operations- reads and writes for improved performance.

Task 05:

What are projections in DynamoDB?

**Projections** in DynamoDB define which attributes from the base table are copied into a secondary index. This helps in querying data directly from the index without accessing the main table.

There are three types: KEYS\_ONLY, INCLUDE, and ALL.

Projections improve query performance and can help reduce read costs.

Task 06:

How can you say DynamoDB prevents Data Loss?

DynamoDB prevents data loss via multi-AZ replication, durable writes to SSD, auto backups, and point-in-time recovery.

Task 07:

What is in-place atomic update? Does DynamoDB support it?

An **in-place atomic update** modifies a value (like incrementing a counter) without reading or rewriting the entire item.

Yes, DynamoDB supports it using UpdateItem with atomic operations.

Task 08:

What are Streams in DynamoDB?

**DynamoDB Streams** capture item-level changes (Insert, Modify, Remove) in near real-time and store them in a log for 24 hours.

Task 09:

What are DynamoDB Pricing tiers?

DynamoDB has 2 main pricing options

1. **On-Demand Mode** (pay per request) – we pay only for the read and writes. No need to plan capacity.
2. **Provisioned Mode** (set RCU/WCU) – we choose how many reads/writes per second we need and pay based on that.

Other costs are

* Storage for how much data we store,
* DynamoDB Streams,
* DAX(cache)
* Data transfer between regions

Task 10:

Max item size in DynamoDB?

The maximum item size is **400 KB** (including all attributes and overhead).

Task 11:

Max number of GSIs per table?

You can create up to **20** Global Secondary Indexes (GSIs) per table.

Task 12:

What is DynamoDB Accelerator (DAX)?

**DAX i**s an in-memory cache for DynamoDB that improves read performance (microsecond latency) without modifying your app.

Task 13:

What are DynamoDB Global Tables?

**Global Tables** provide multi-region, fully replicated tables for high availability and low-latency global access.

Task 14:

What are indexes and secondary indexes in DynamoDB?

Indexes helps us find data faster in a DynamoDB table. Types:

1. **Primary Index:** Main index based on the table's partition + optional sort key
2. **Secondary Index:** Alternate query patterns, allowing us to query data in different ways without scanning whole table.

There are 2 types of secondary indexes

**Local Secondary Index LSI:** Uses same partition key, different sort key

**Global Secondary Index GSI:** Uses different partition + sort key

Task 15:

What are Hot Keys and Hot Partitions?

**Hot Key:** A key that is accessed too frequently causing performance issues.

**Hot Partition:** A partition receiving too many read/write requests, causing overloading issues.

Task 16:

Table-level vs Item-level operations in DynamoDB?

**Table-level operations** - Actions that affect the whole table like CreateTable, DeleteTable, UpdateTable

**Item-level operations** - Actions on individual items(rows) like GetItem, PutItem, UpdateItem, DeleteItem