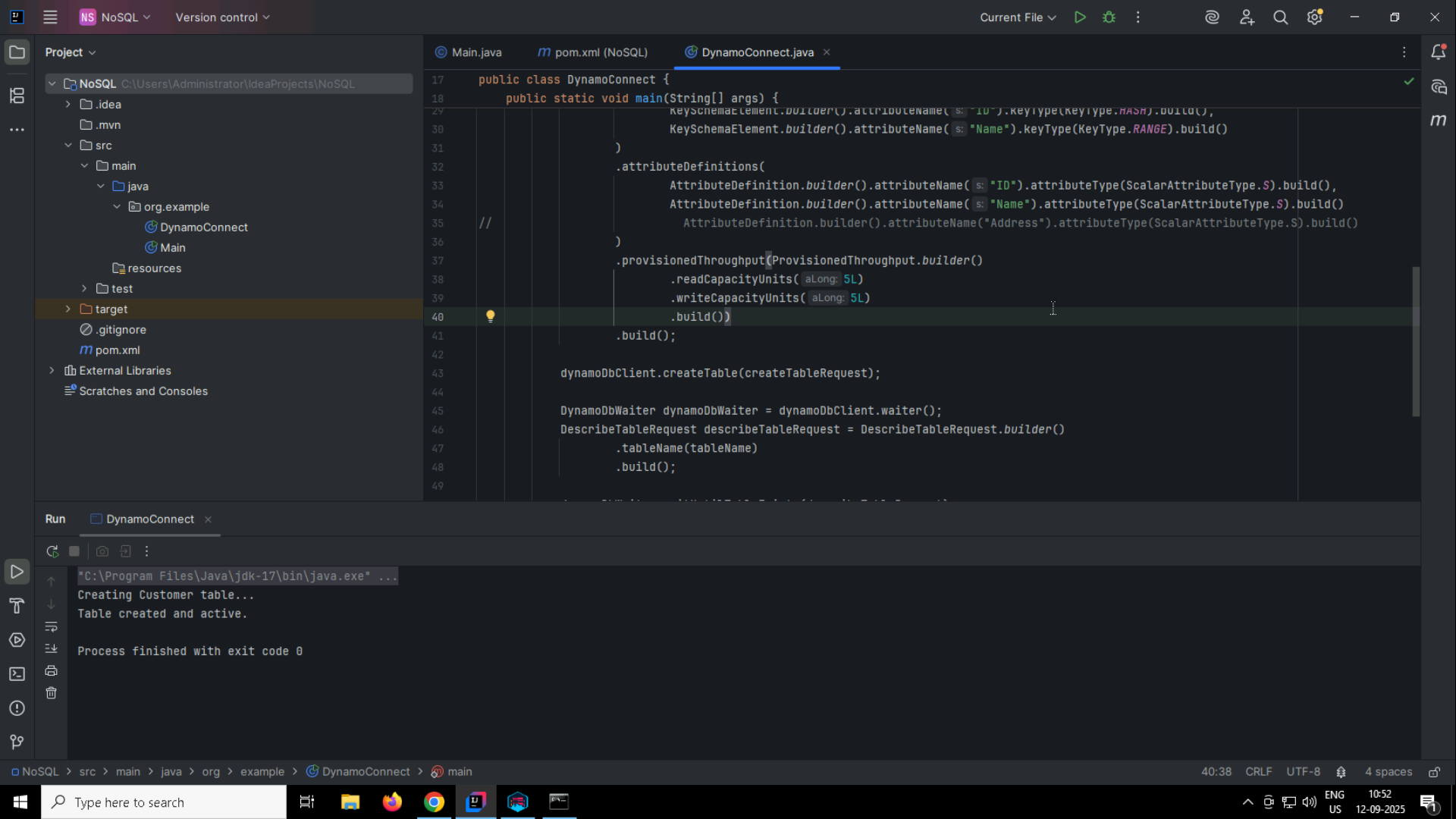
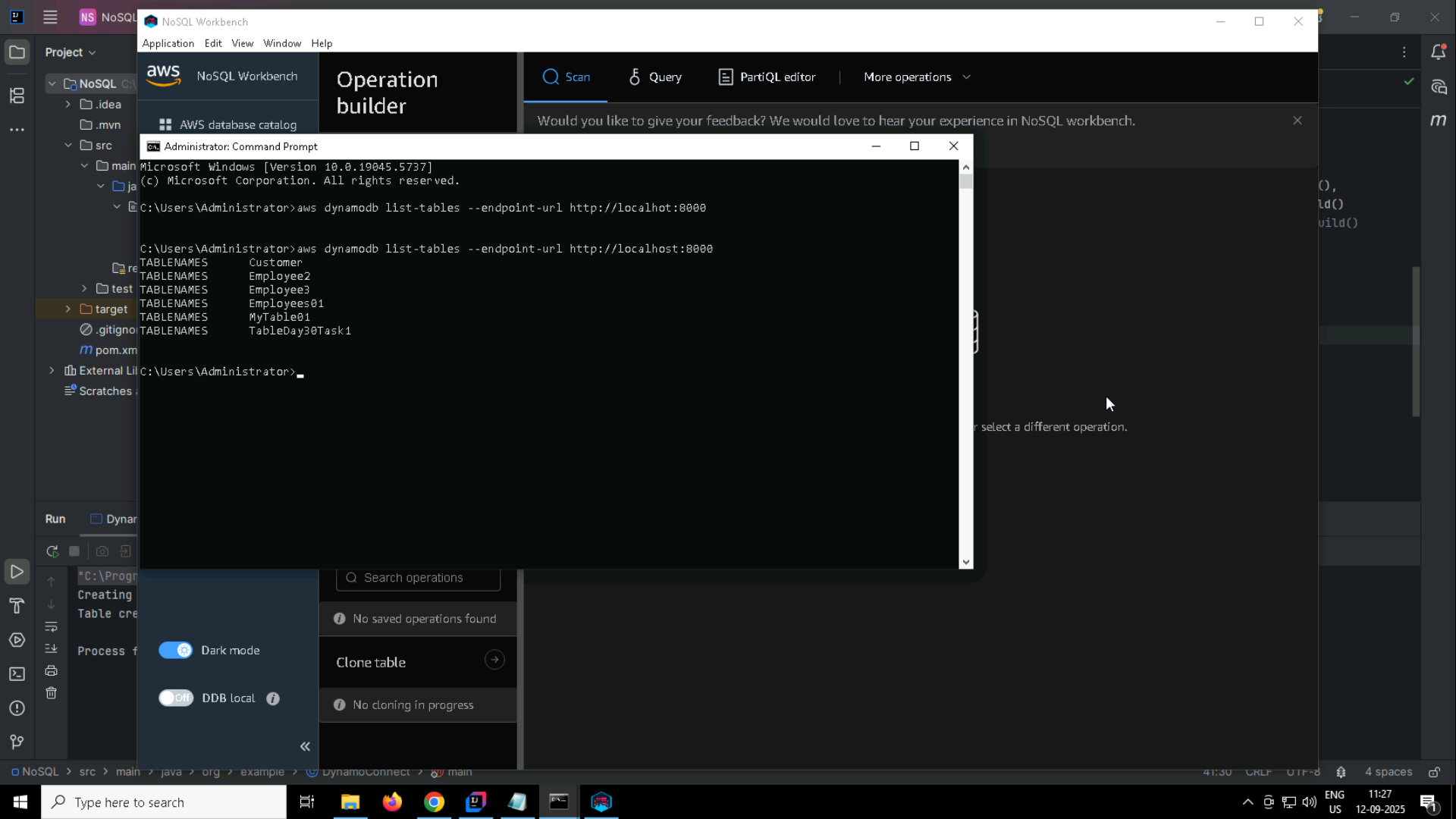
Day 32 – 12/09/2025

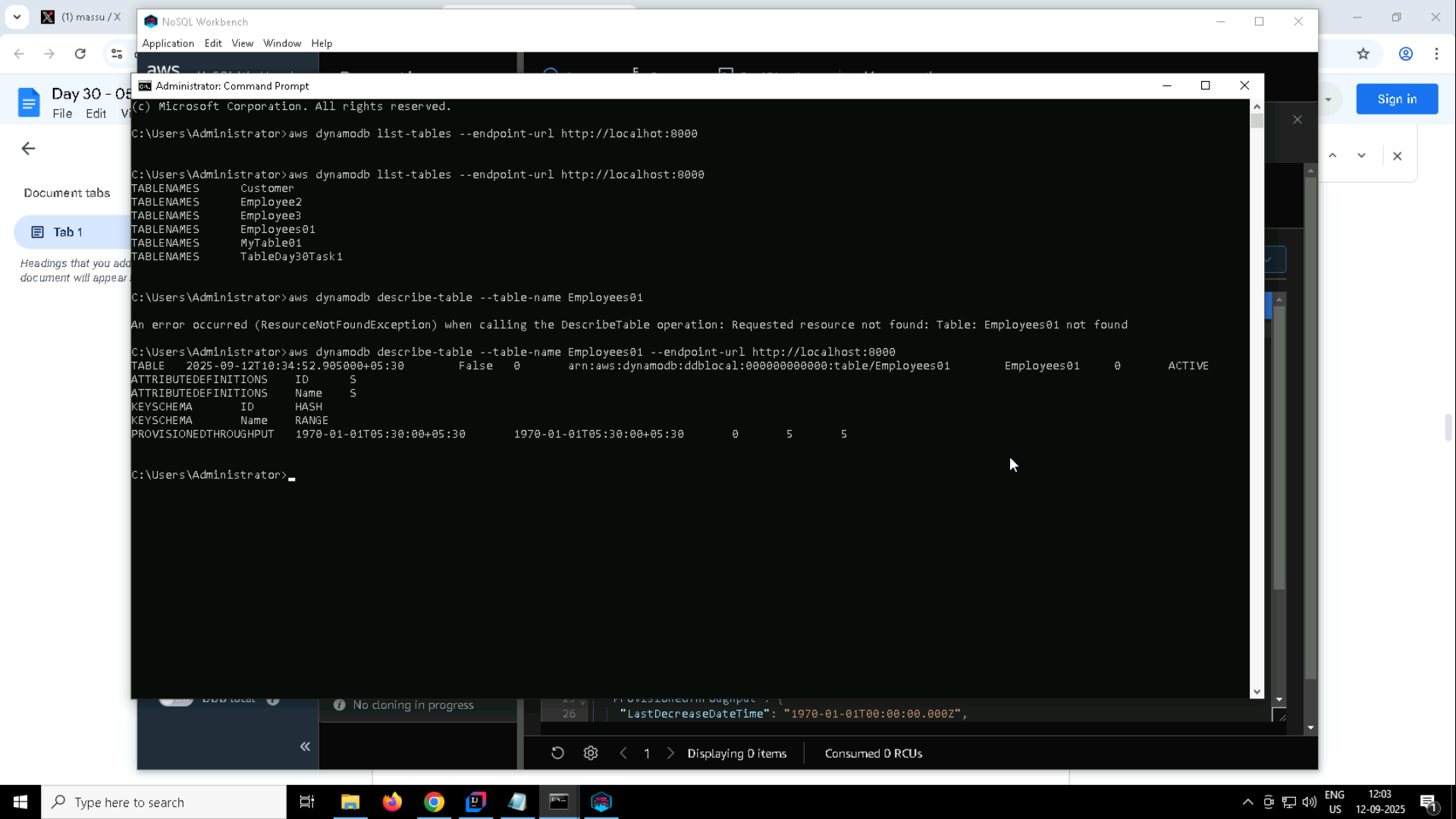
// Task001: Connecting DynamoDB  
  
package org.example;  
  
import software.amazon.awssdk.services.dynamodb.DynamoDbClient;  
import software.amazon.awssdk.services.dynamodb.model.AttributeDefinition;  
import software.amazon.awssdk.services.dynamodb.model.CreateTableRequest;  
import software.amazon.awssdk.services.dynamodb.model.DescribeTableRequest;  
import software.amazon.awssdk.services.dynamodb.model.KeySchemaElement;  
import software.amazon.awssdk.services.dynamodb.model.KeyType;  
import software.amazon.awssdk.services.dynamodb.model.ProvisionedThroughput;  
import software.amazon.awssdk.services.dynamodb.model.ScalarAttributeType;  
import software.amazon.awssdk.services.dynamodb.waiters.DynamoDbWaiter;  
import java.net.URI;  
  
public class DynamoConnect {  
 public static void main(String[] args) {  
 try (DynamoDbClient dynamoDbClient = DynamoDbClient.*builder*()  
 .endpointOverride(URI.*create*("http://localhost:8000"))  
 .build()) {  
 String tableName = "Employees01";  
 System.*out*.println("Creating Customer table...");  
  
 CreateTableRequest createTableRequest = CreateTableRequest.*builder*()  
 .tableName(tableName).keySchema(  
 KeySchemaElement.*builder*().attributeName("ID").keyType(KeyType.*HASH*).build(),  
 KeySchemaElement.*builder*().attributeName("Name").keyType(KeyType.*RANGE*).build()).attributeDefinitions(  
 AttributeDefinition.*builder*().attributeName("ID").attributeType(ScalarAttributeType.*S*).build(),  
 AttributeDefinition.*builder*().attributeName("Name").attributeType(ScalarAttributeType.*S*).build(),  
 AttributeDefinition.*builder*().attributeName("Address").attributeType(ScalarAttributeType.*S*).build())  
 .provisionedThroughput(ProvisionedThroughput.*builder*()  
 .readCapacityUnits(5L)  
 .writeCapacityUnits(5L)  
 .build())  
 .build();  
  
 dynamoDbClient.createTable(createTableRequest);   
 System.*out*.println("Table created and active.");  
 } catch (Exception ex) {  
 System.*err*.println("Error creating table: " + ex.getMessage());  
 }  
 }  
}



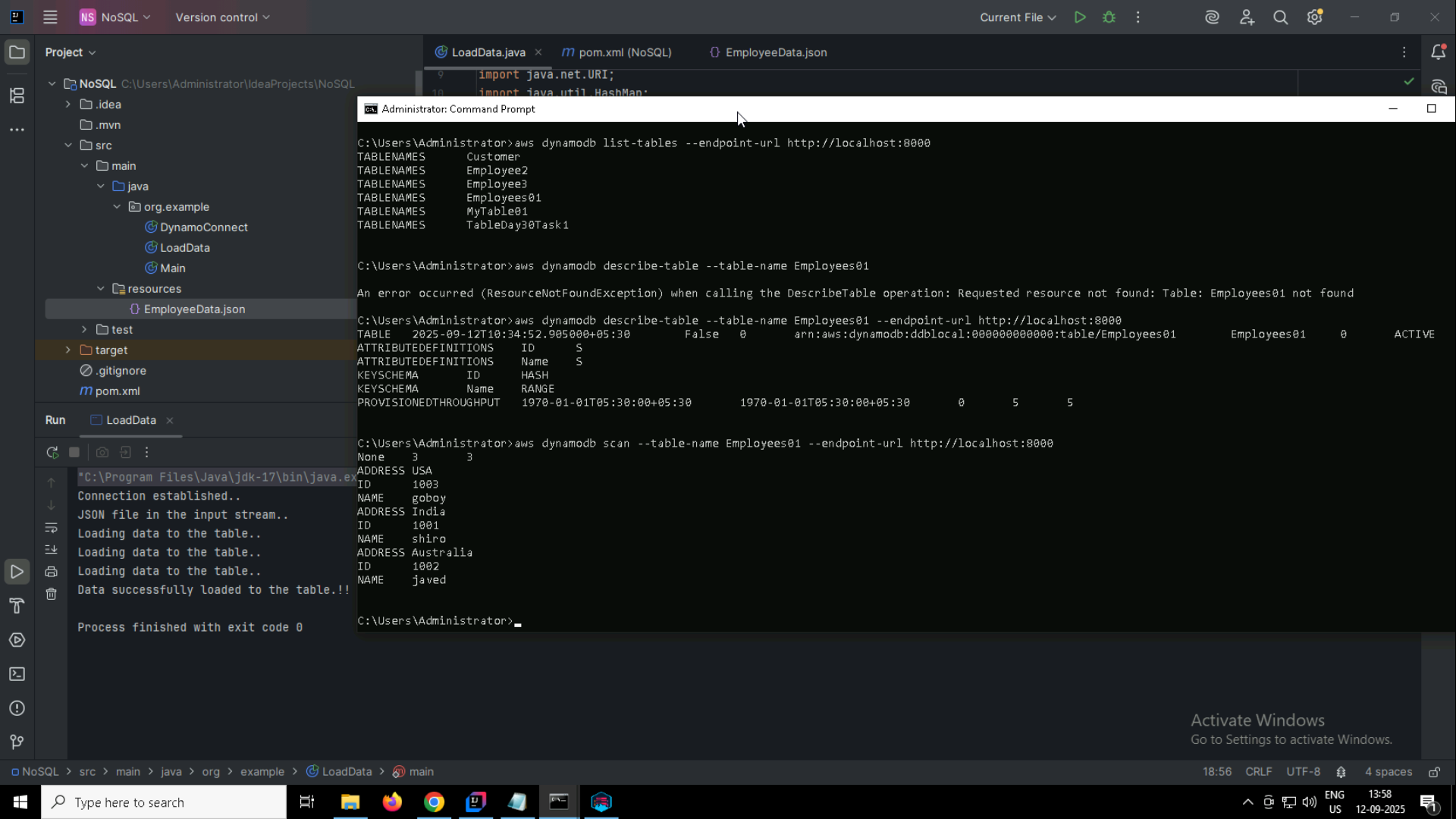
Q2. Check the tables in CLI



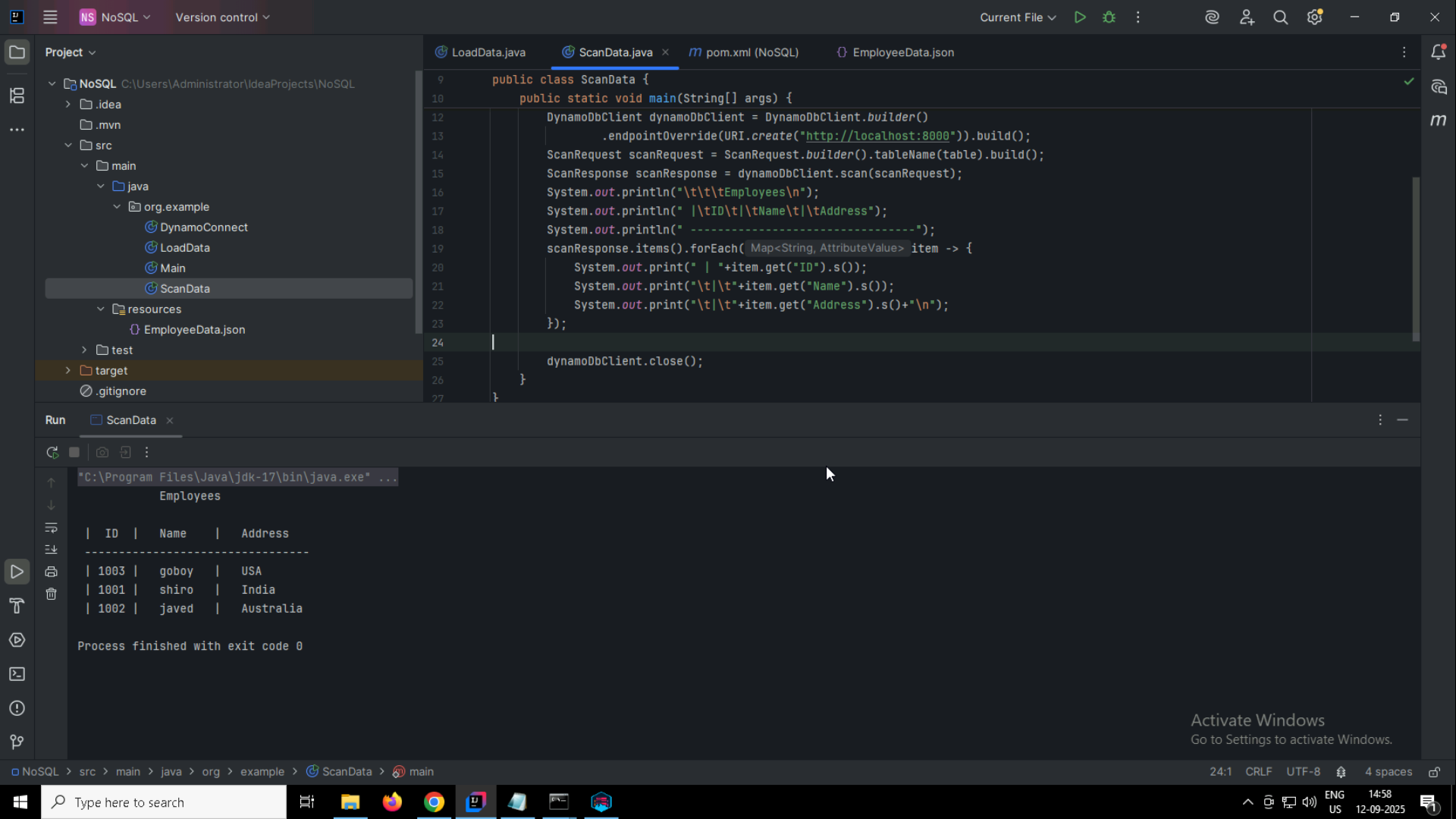
Q3. For the same above tables see the description of the tables.



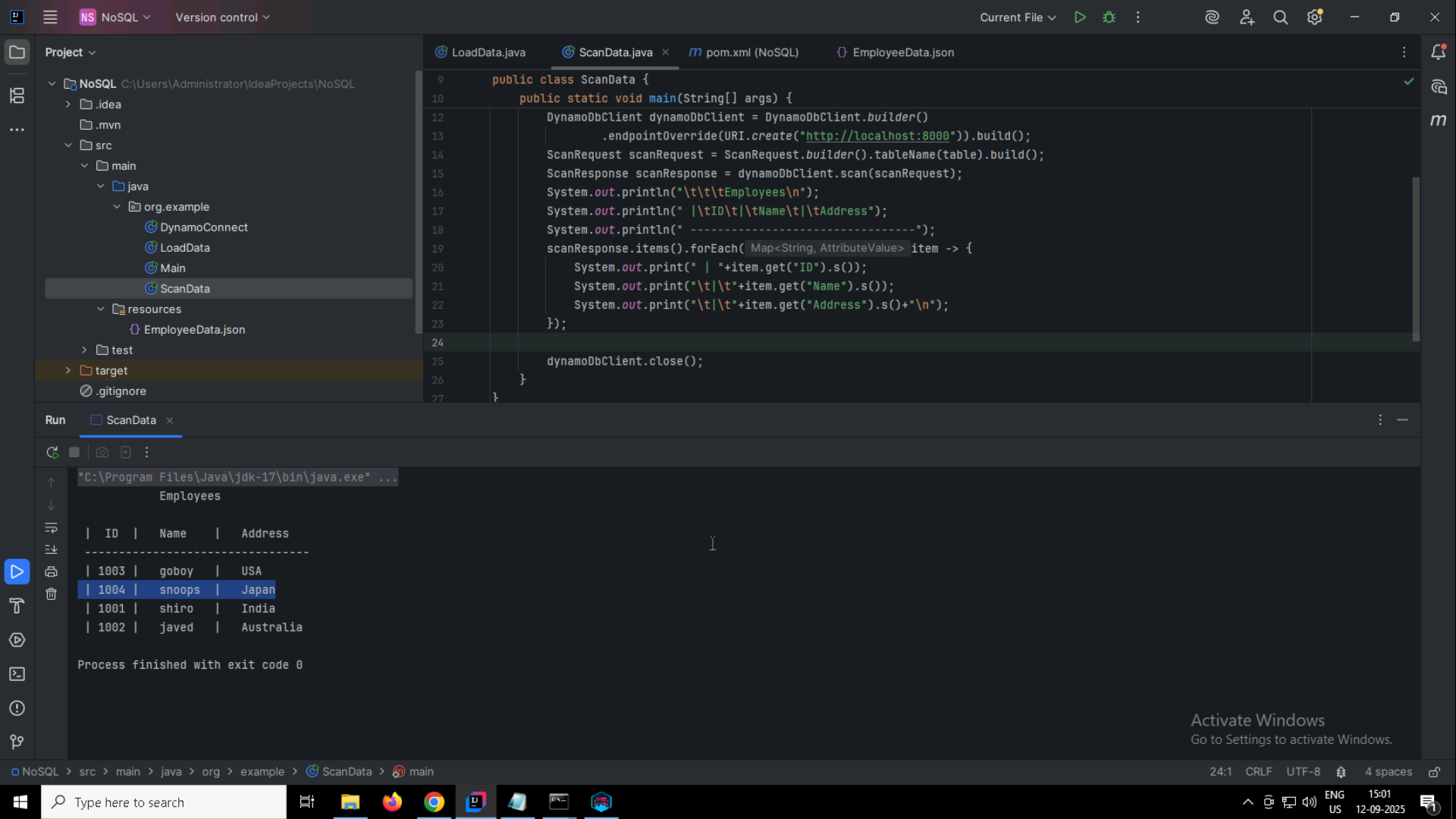
Q4. Loading/inserting data to the table.



Q5. Reading/scanning data from the server.

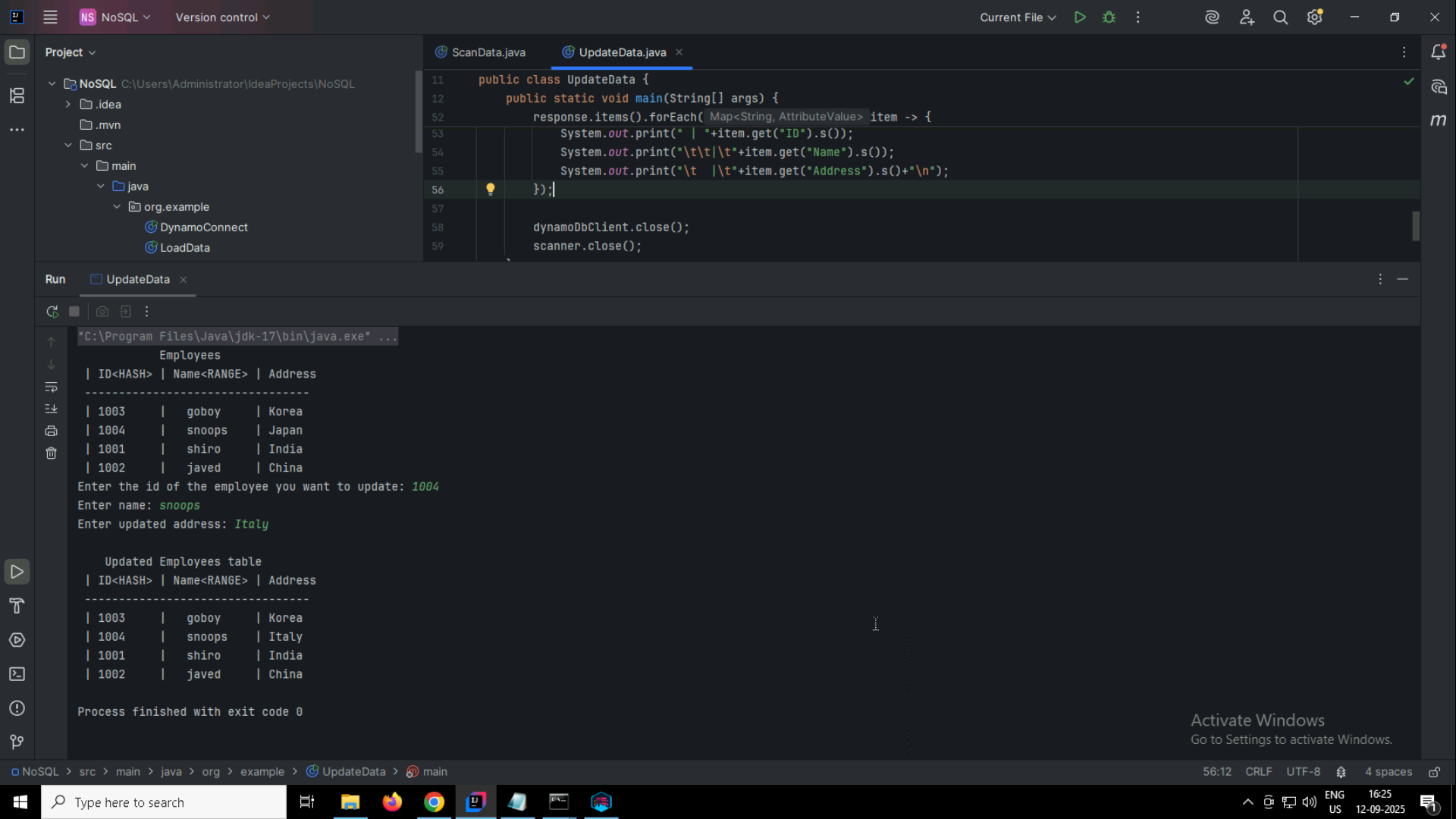


Q6. Insert an item in the NoSQL workbench and check if the item reflects in your java output.



Q7. How do you lower the cost of DynamoDB? Explain ways to do so.  
Ans. homework

Q8. Update item details using java code and check if it reflects in the server.



Q9. Delete a particular item from the table

