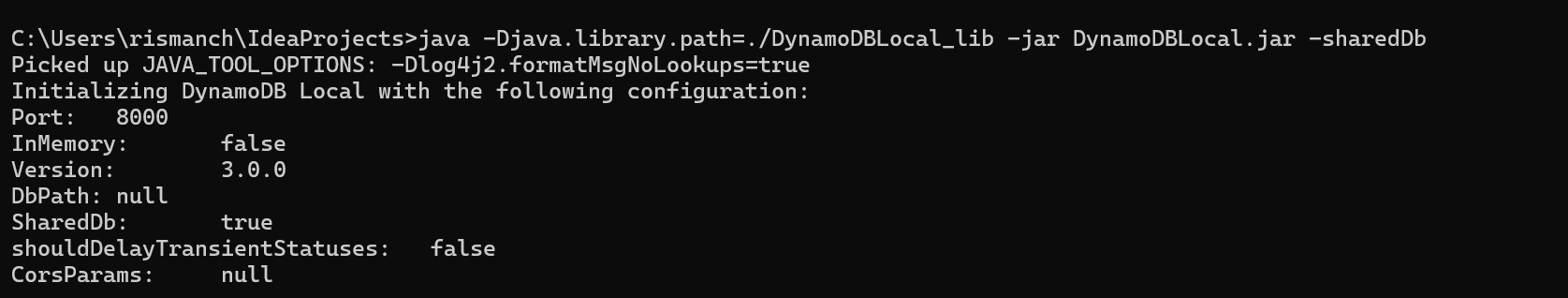
Name – Rishabh Manchanda

Employee ID:

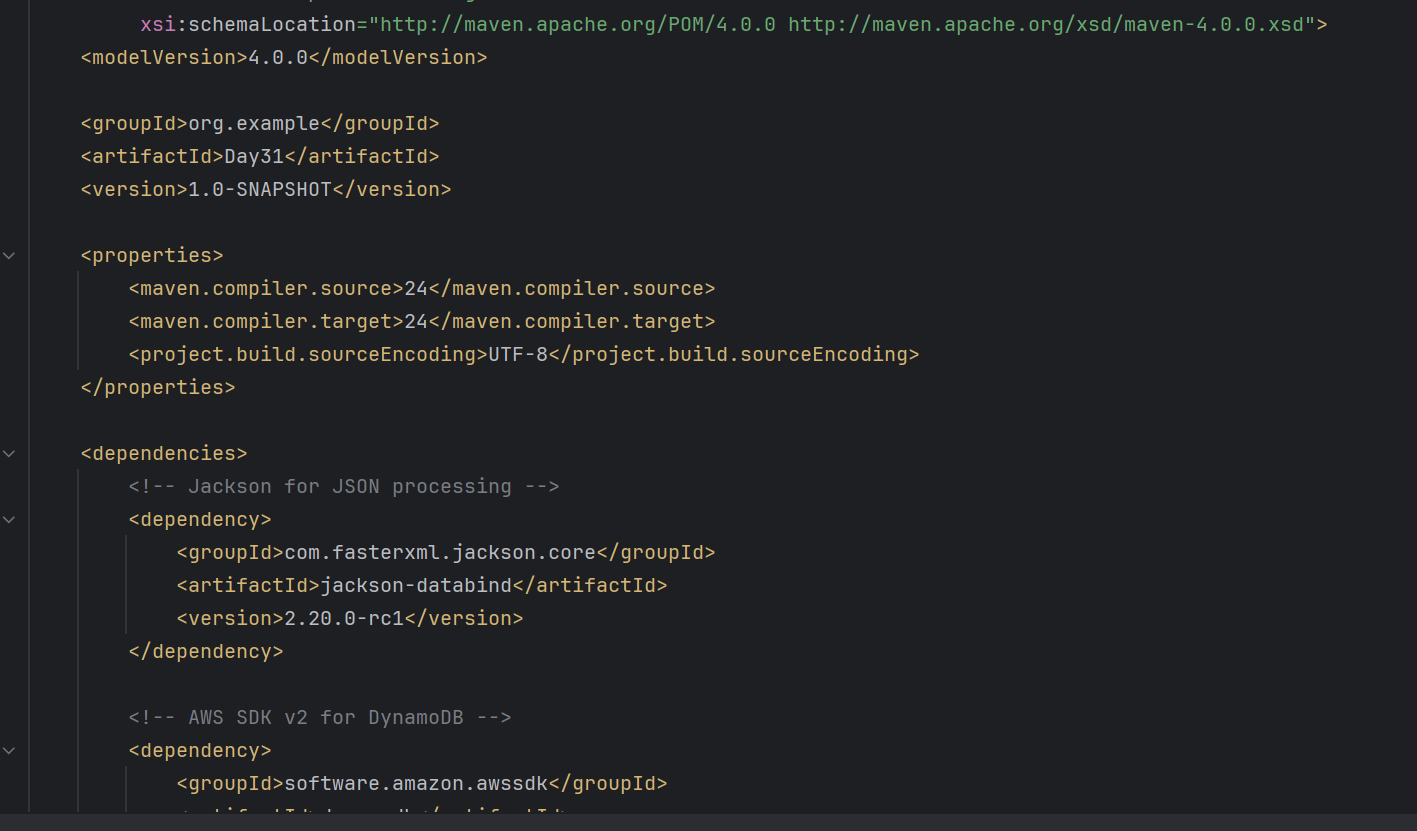
109883998

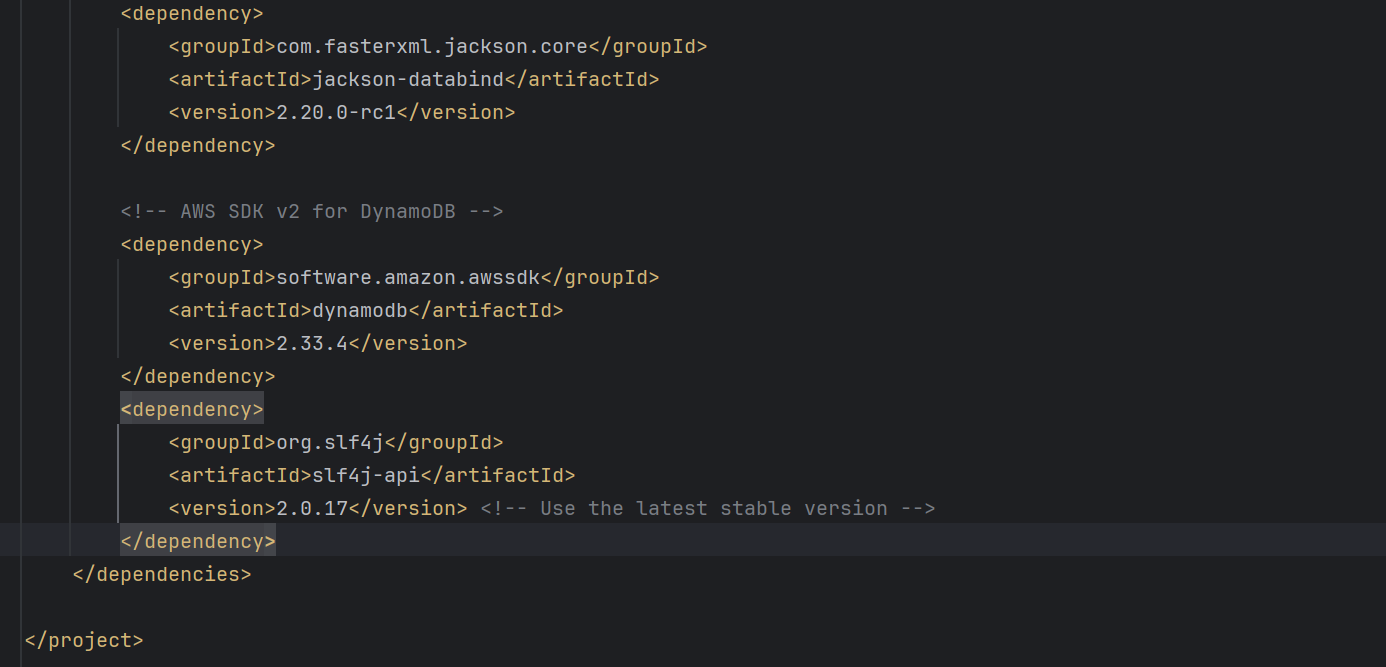
DAY-32

Task 01

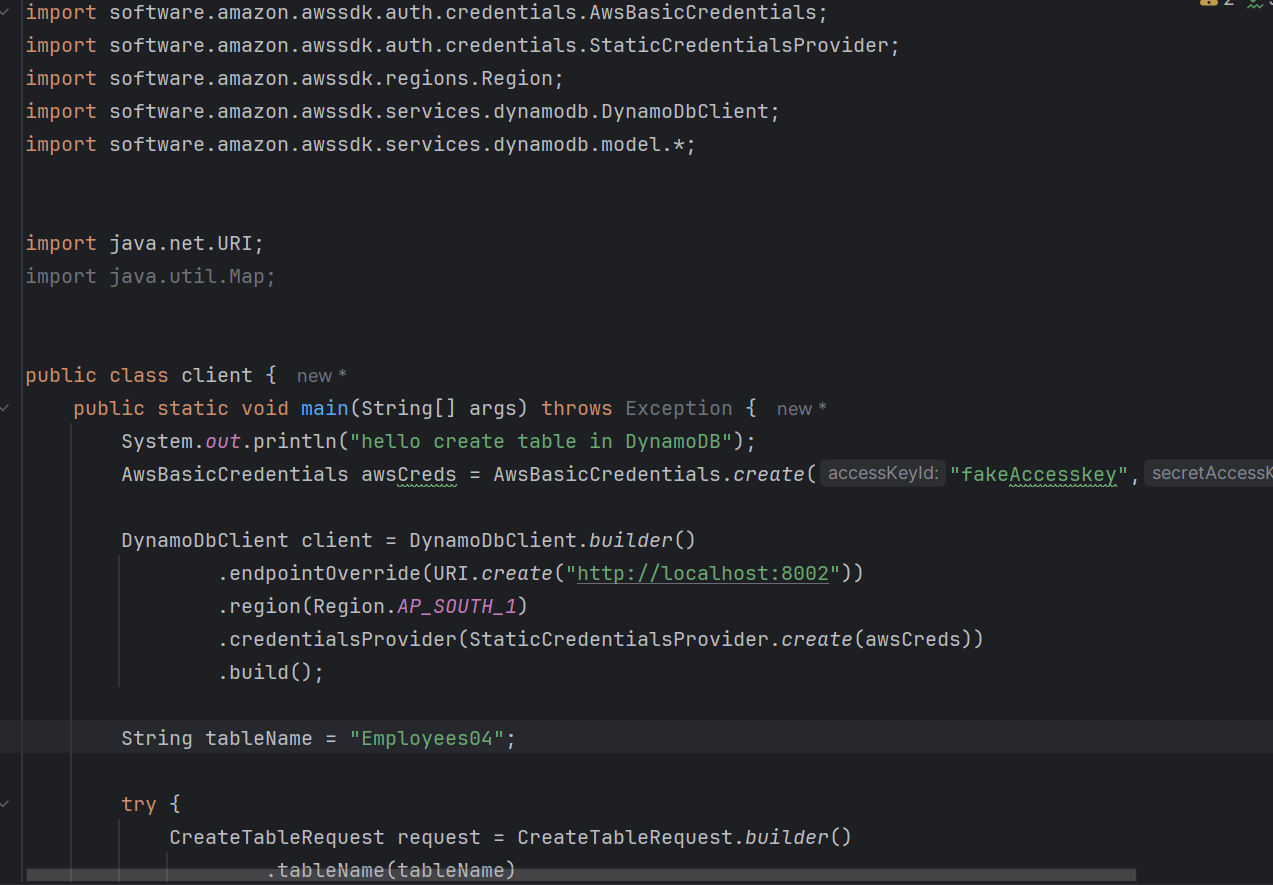


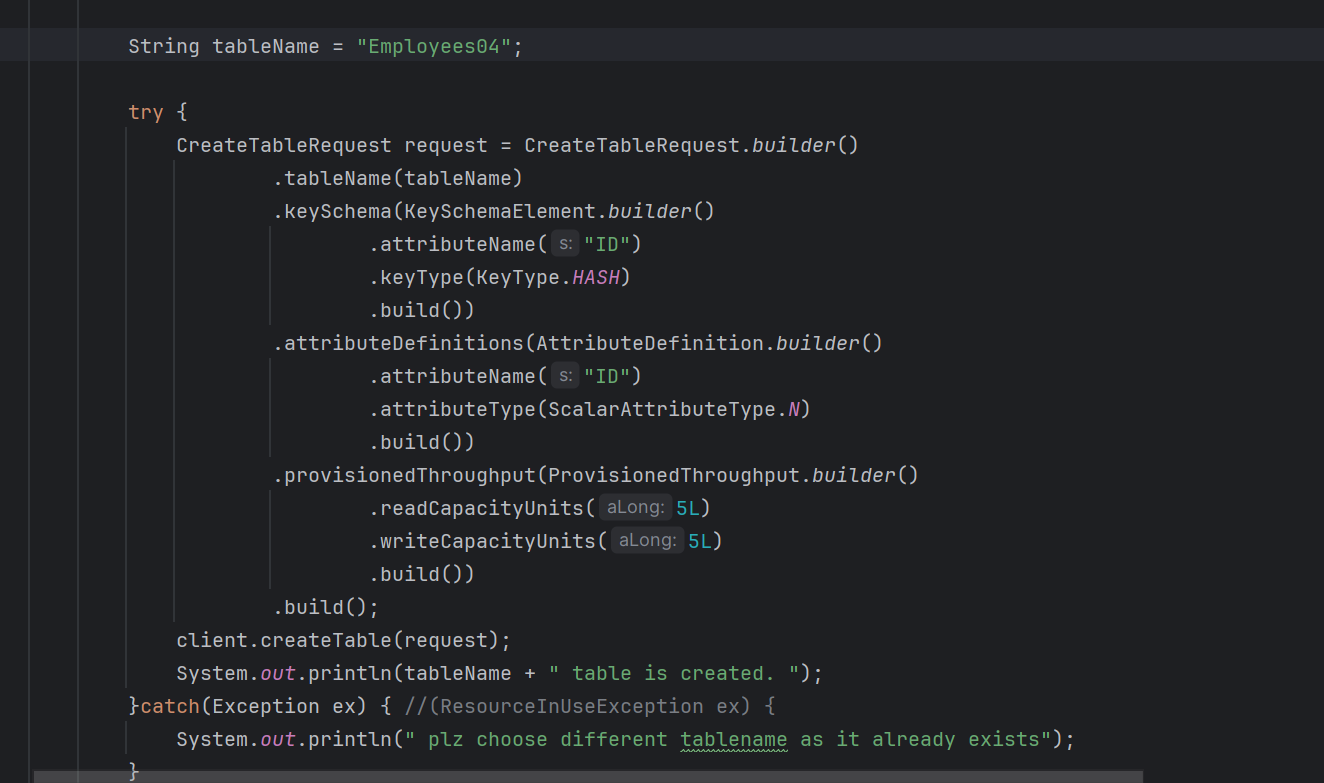
Task02

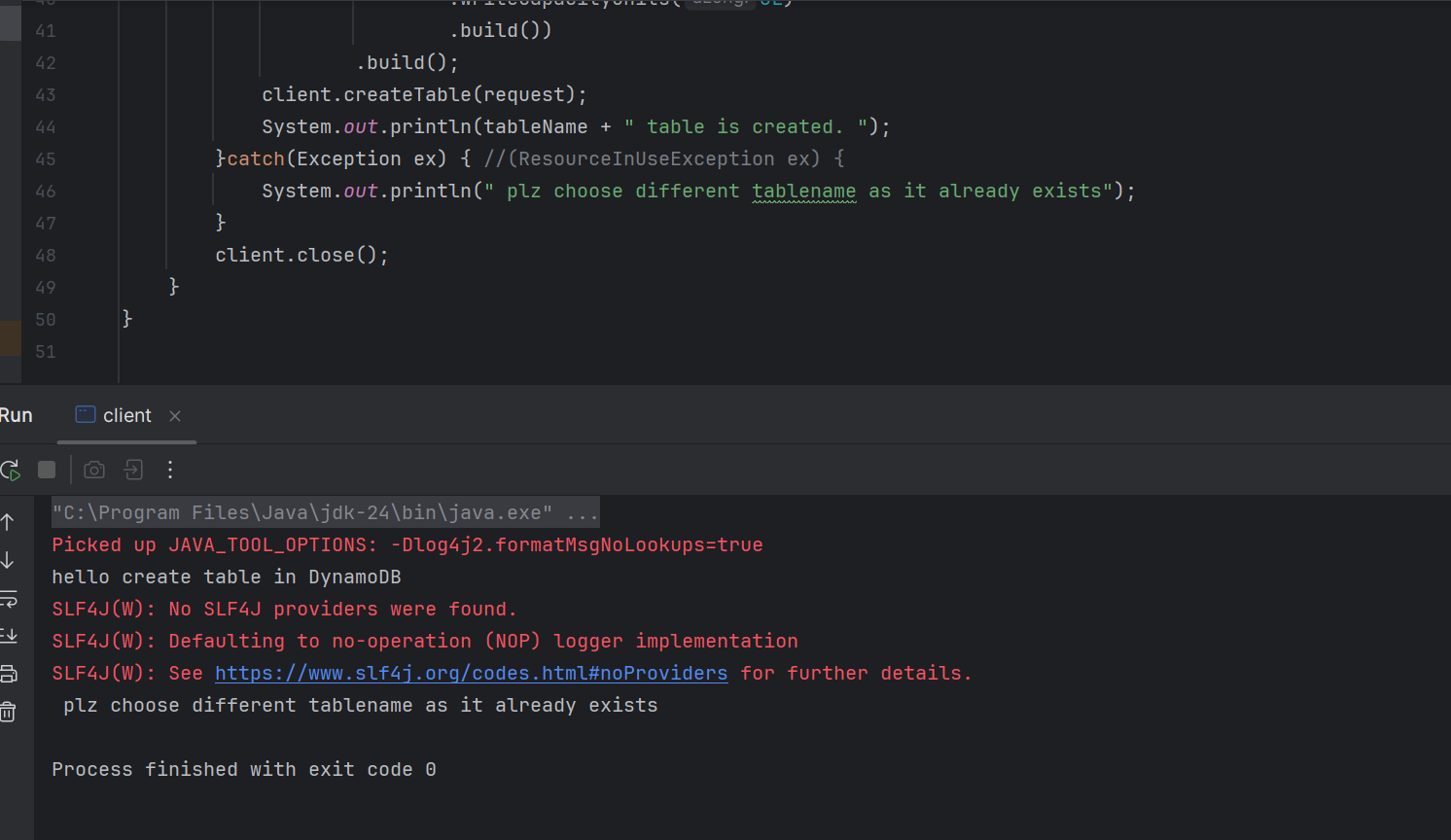


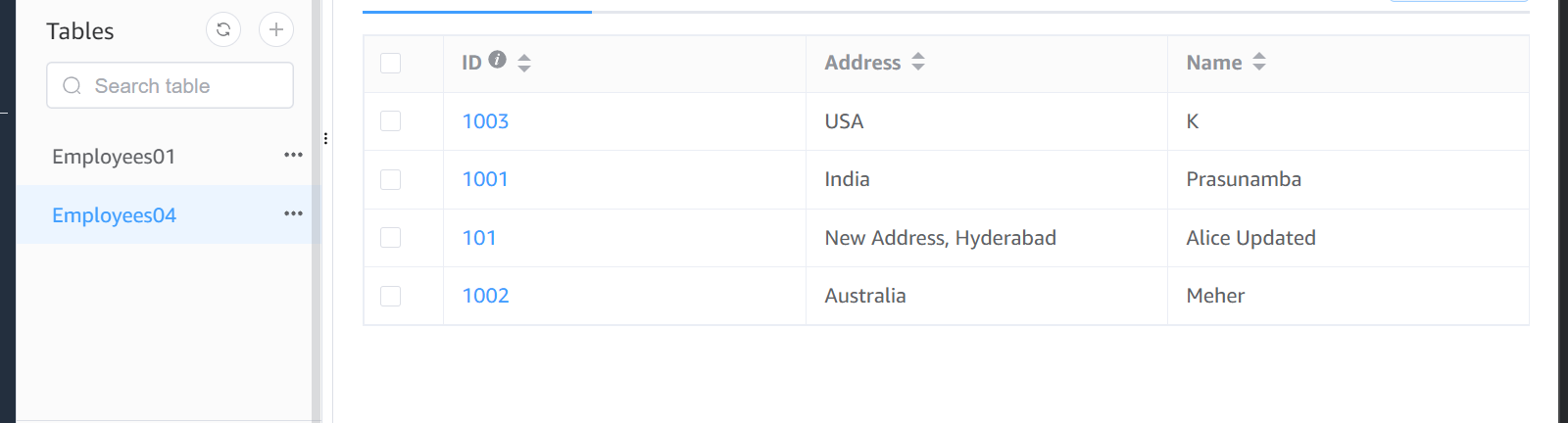


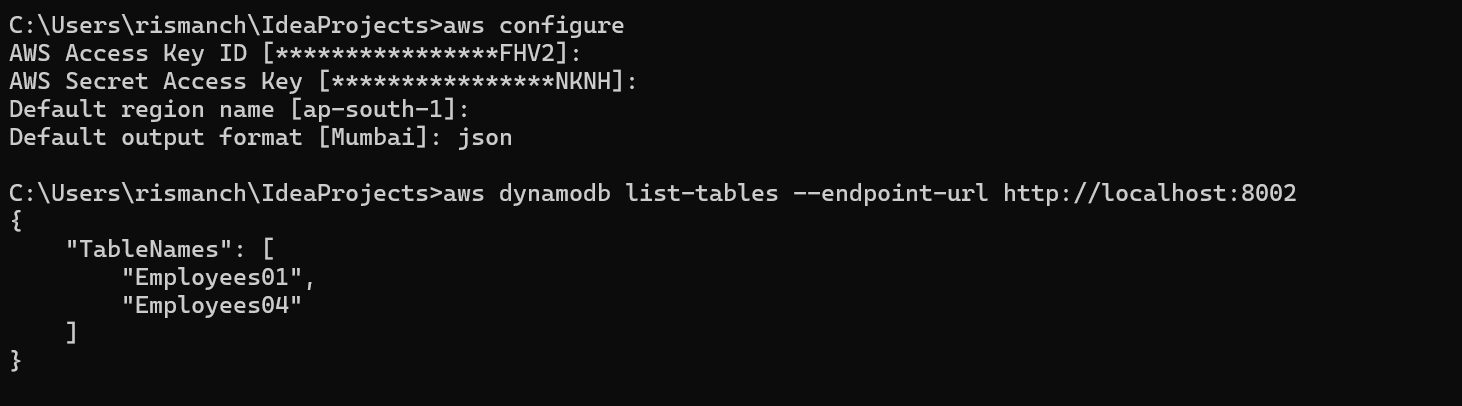
Task03

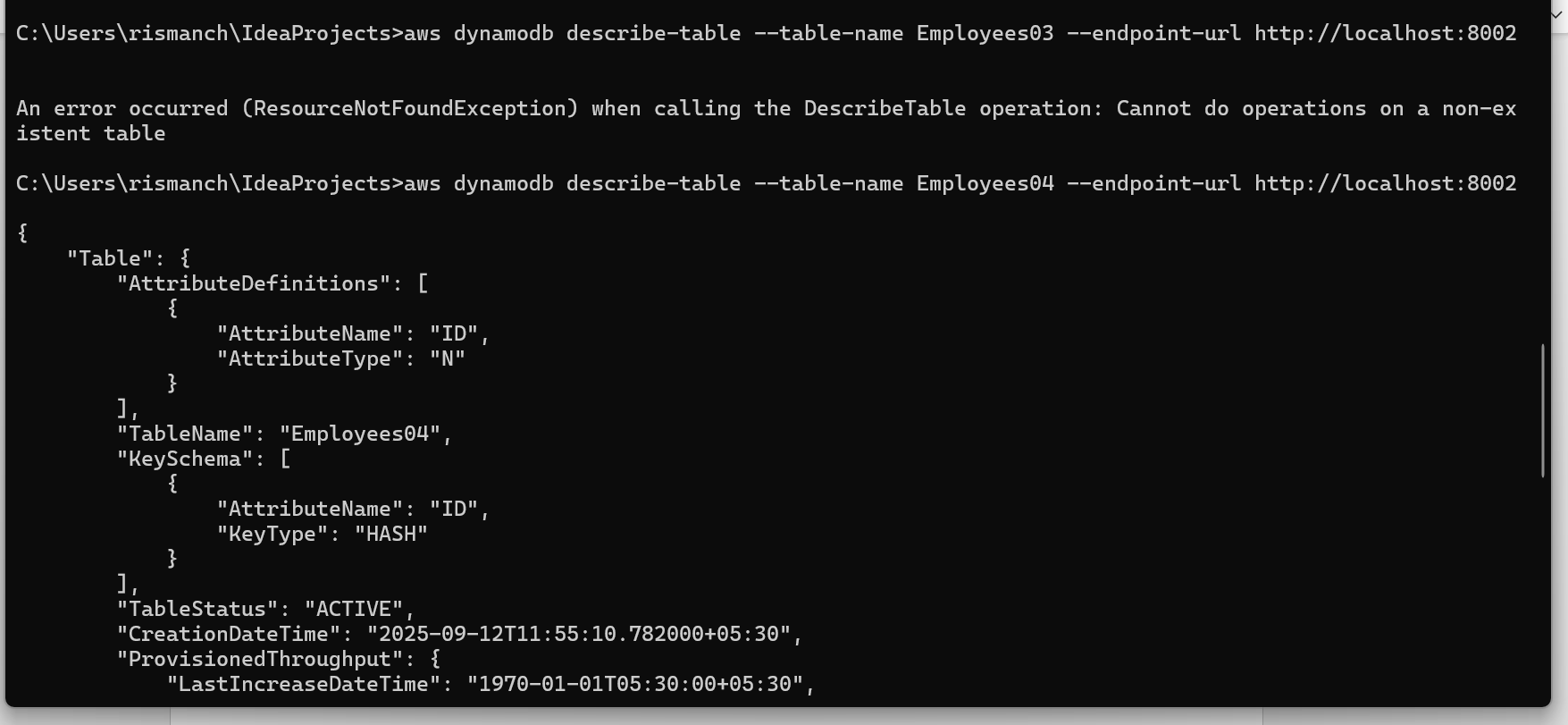


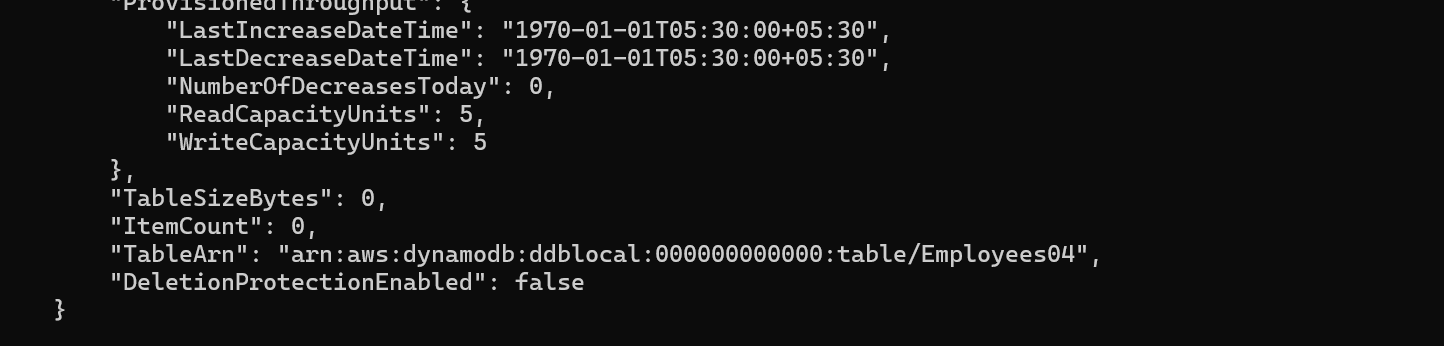


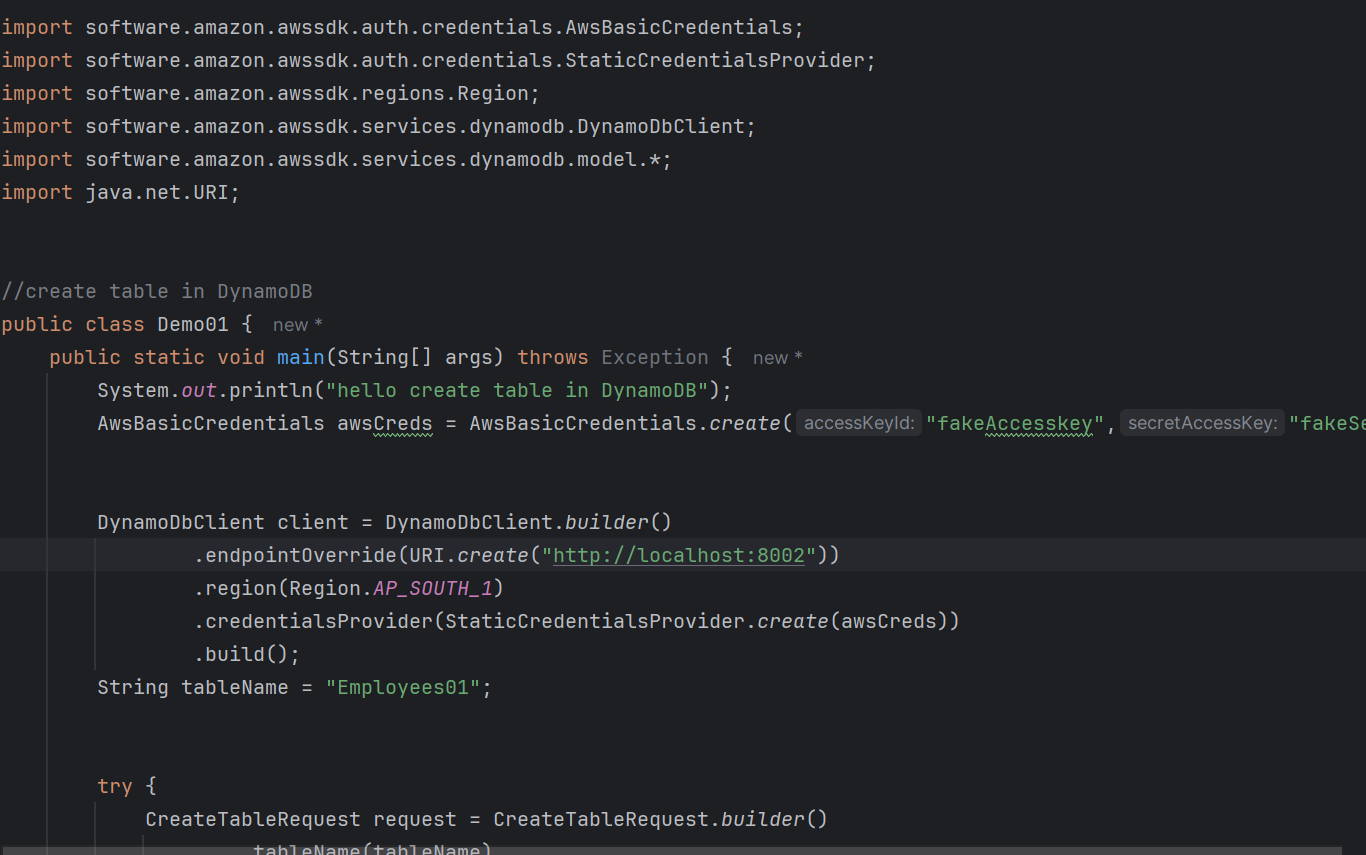


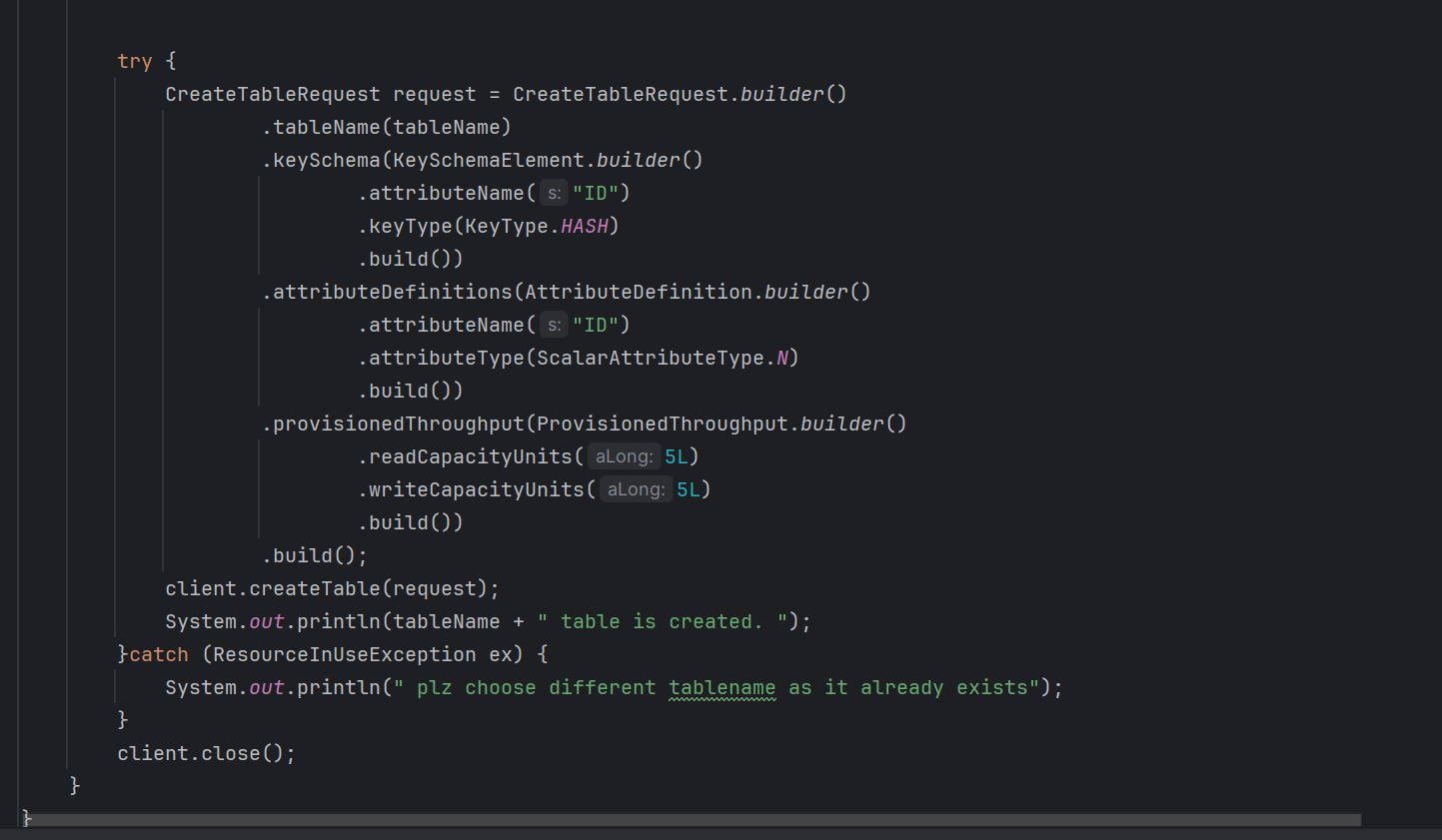


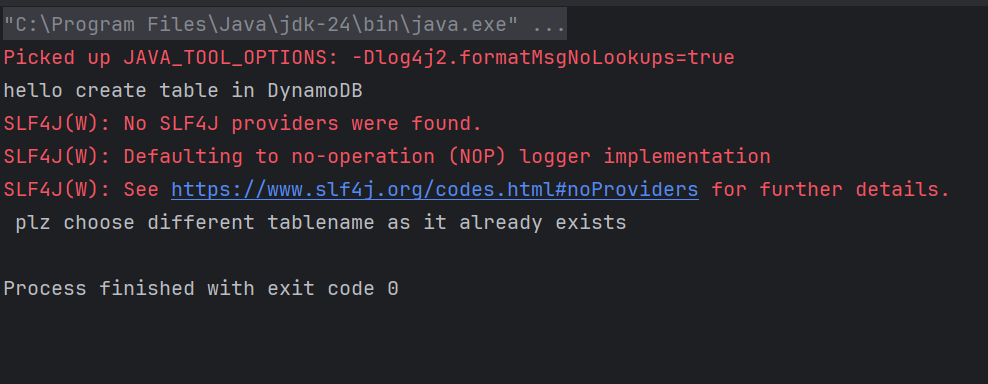


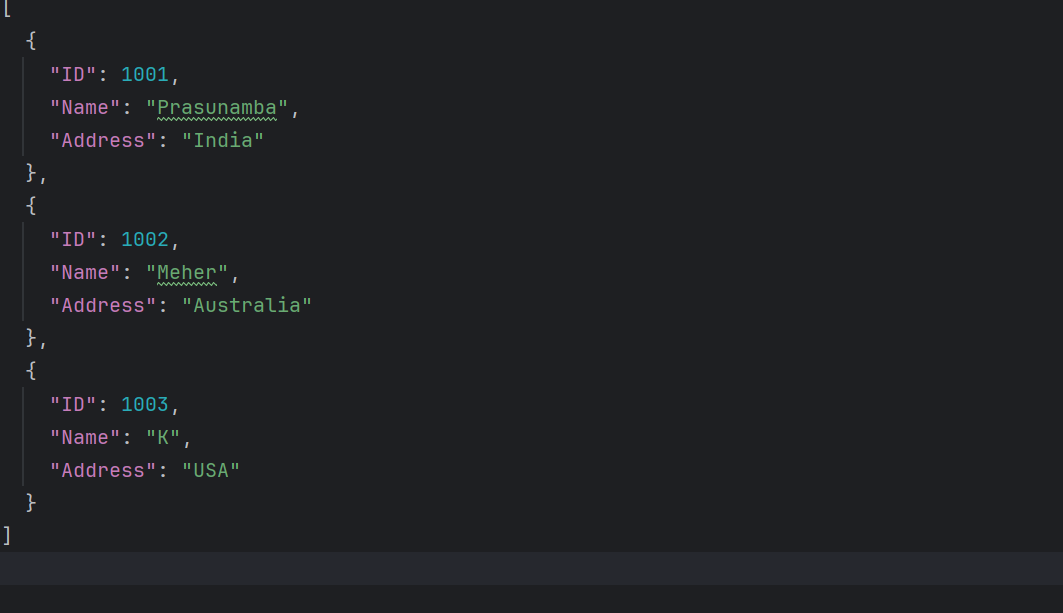


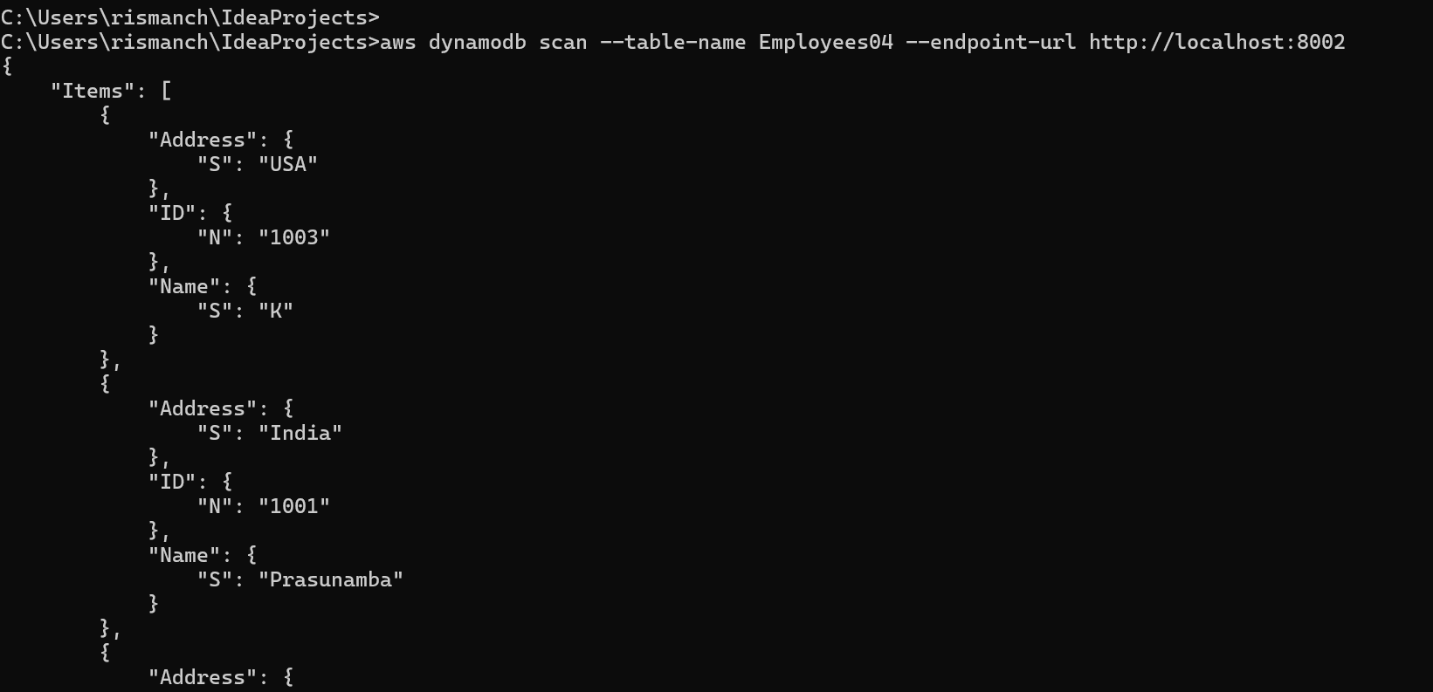




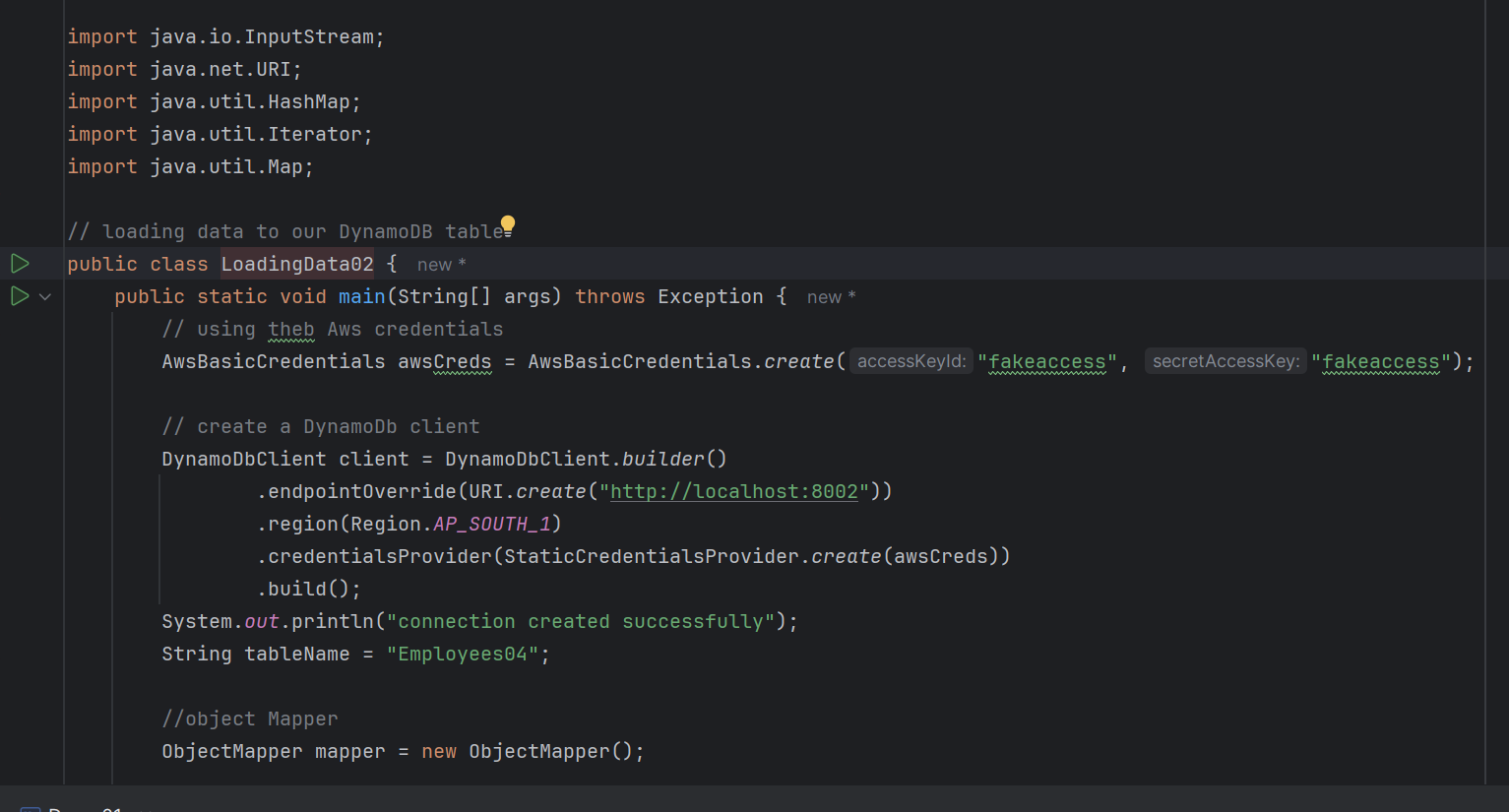




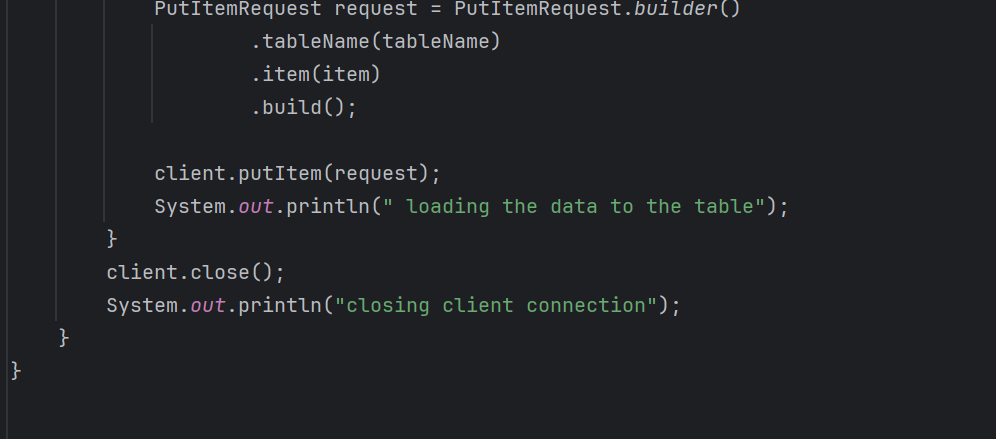


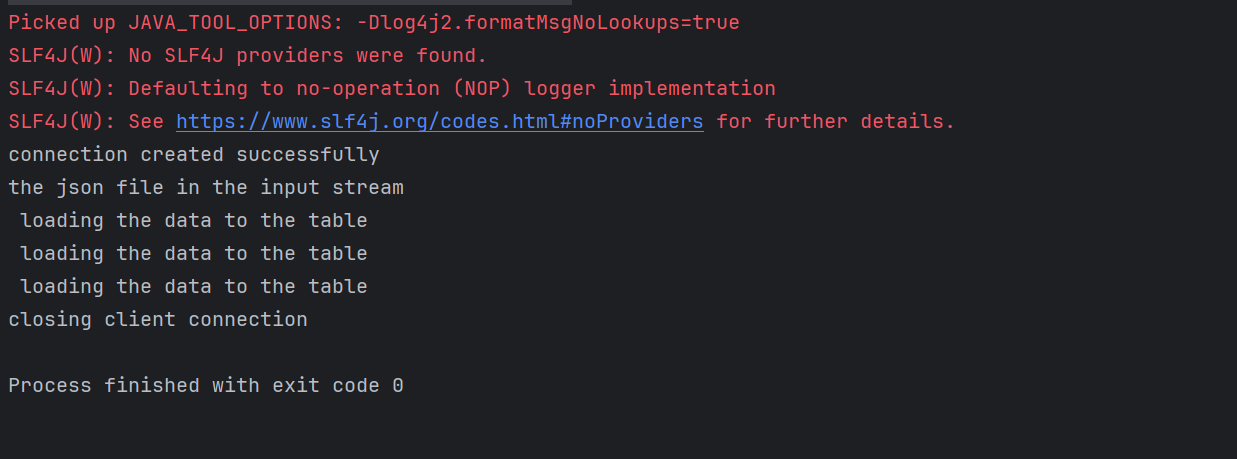


Loading data

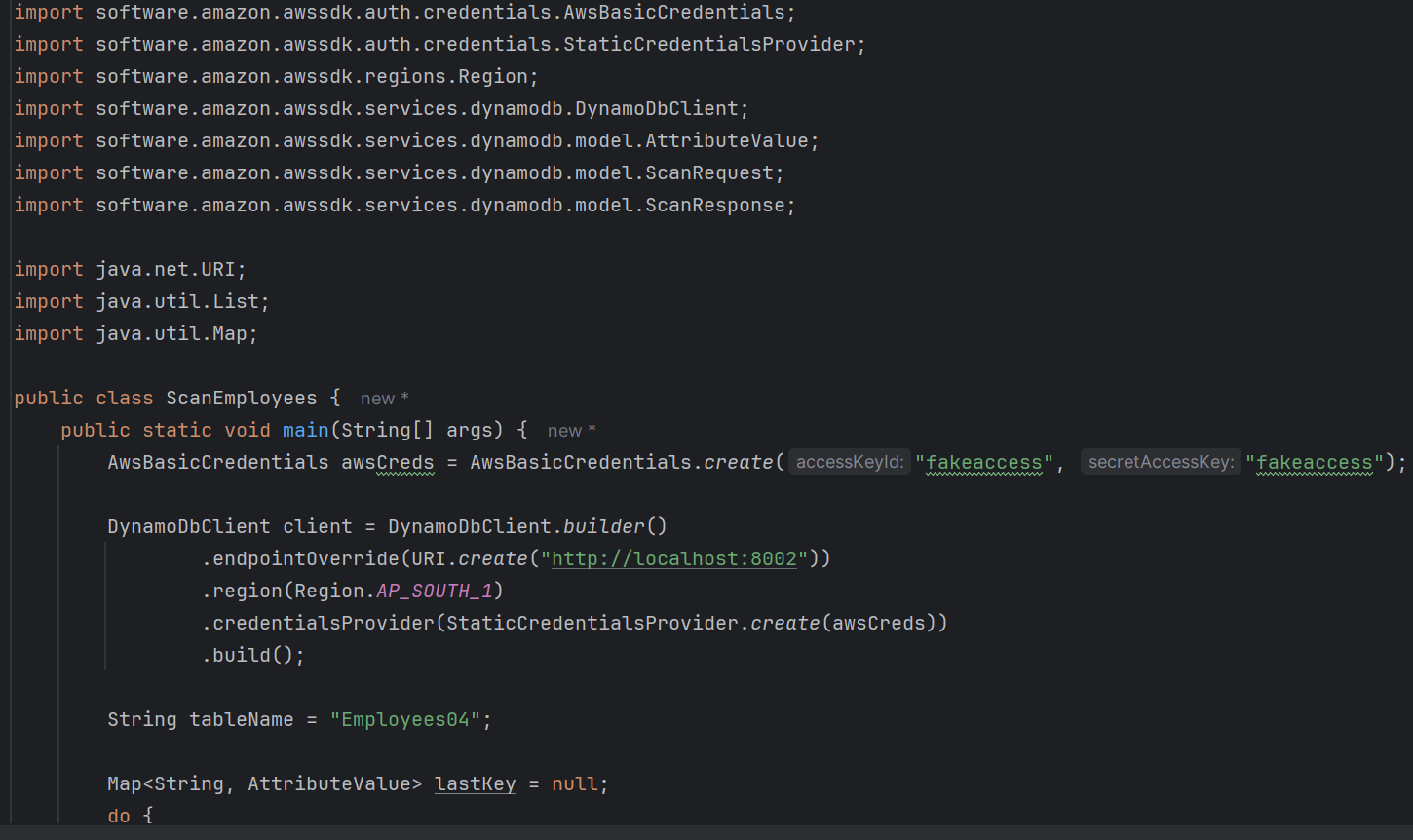




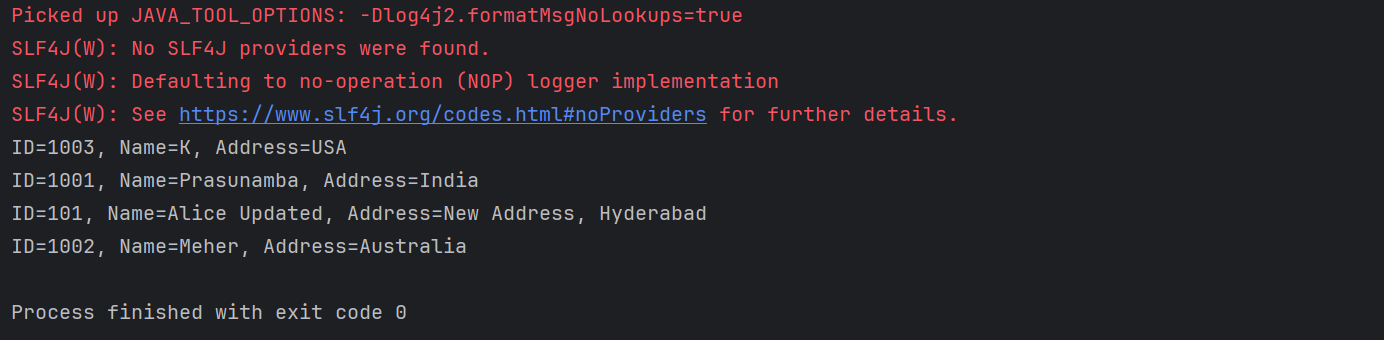




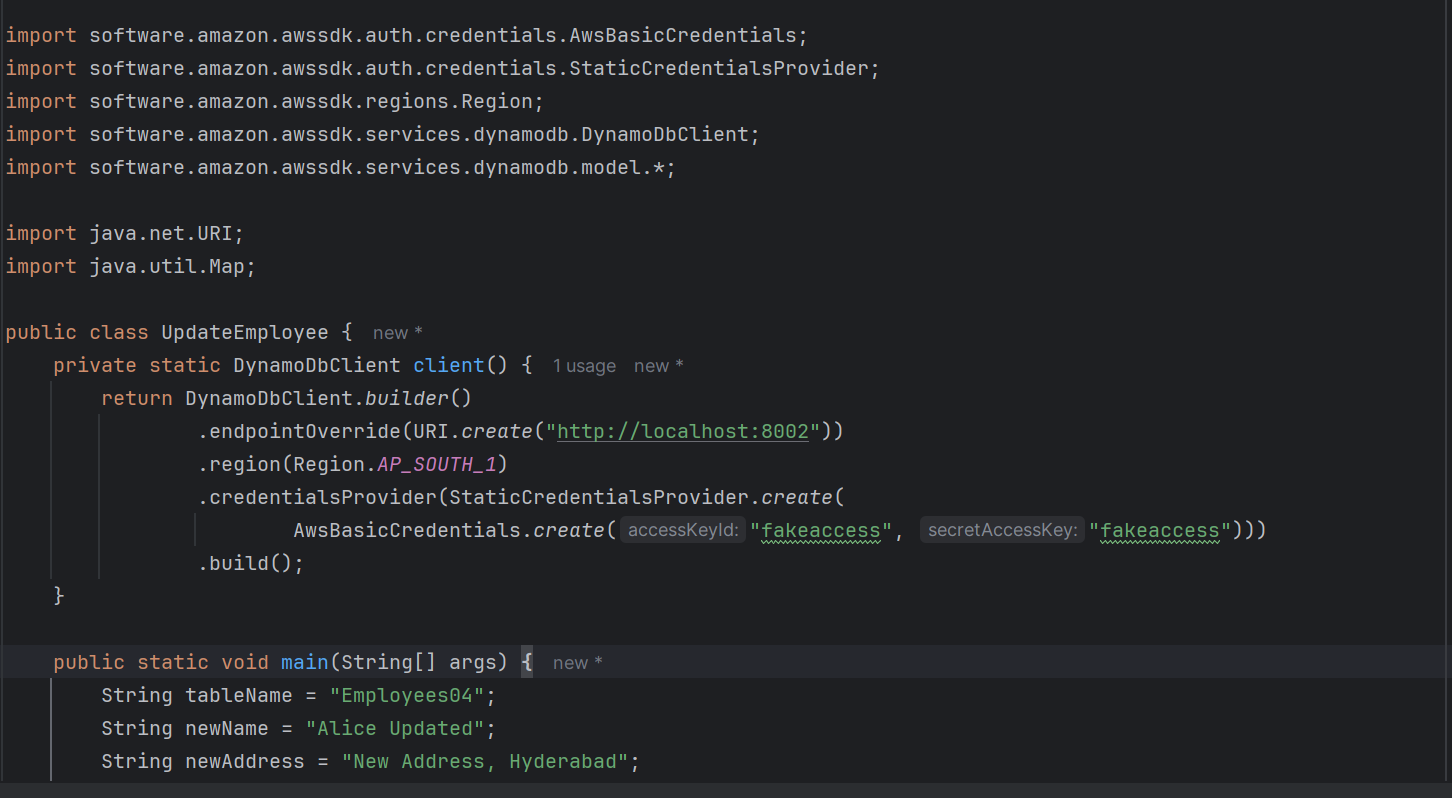
Scan Employees

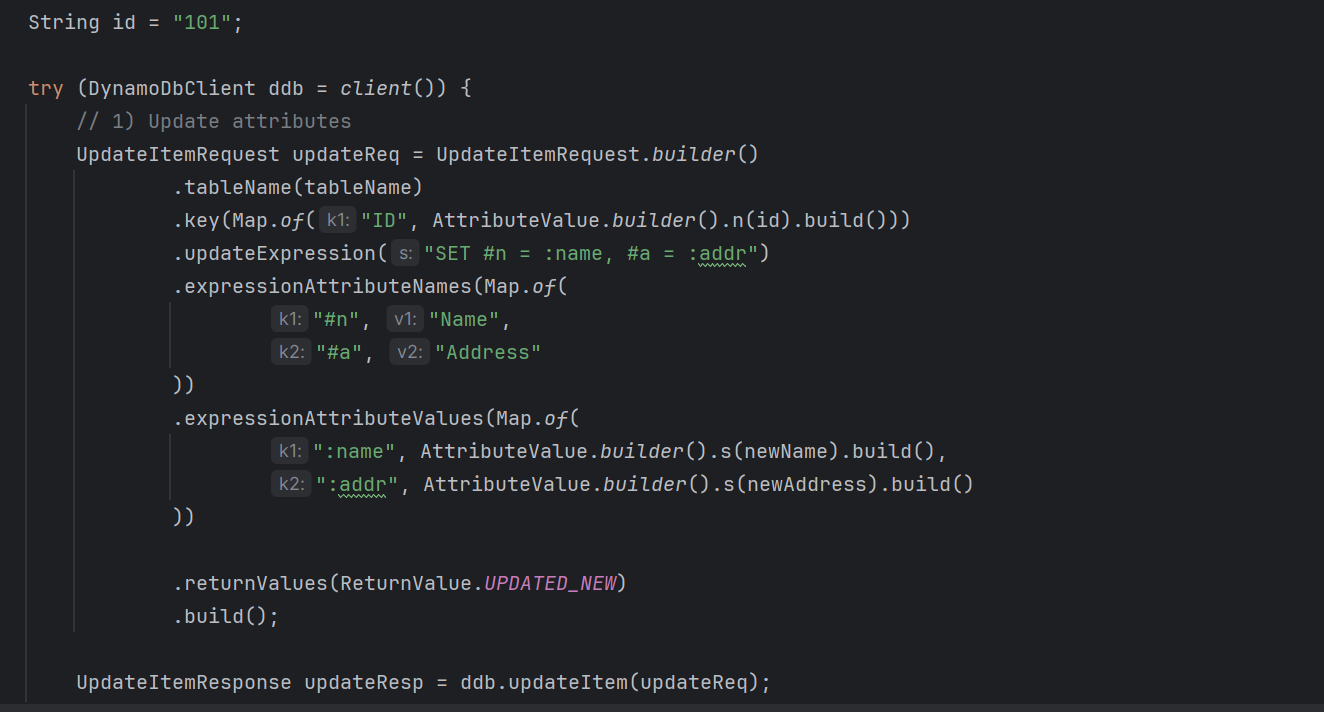


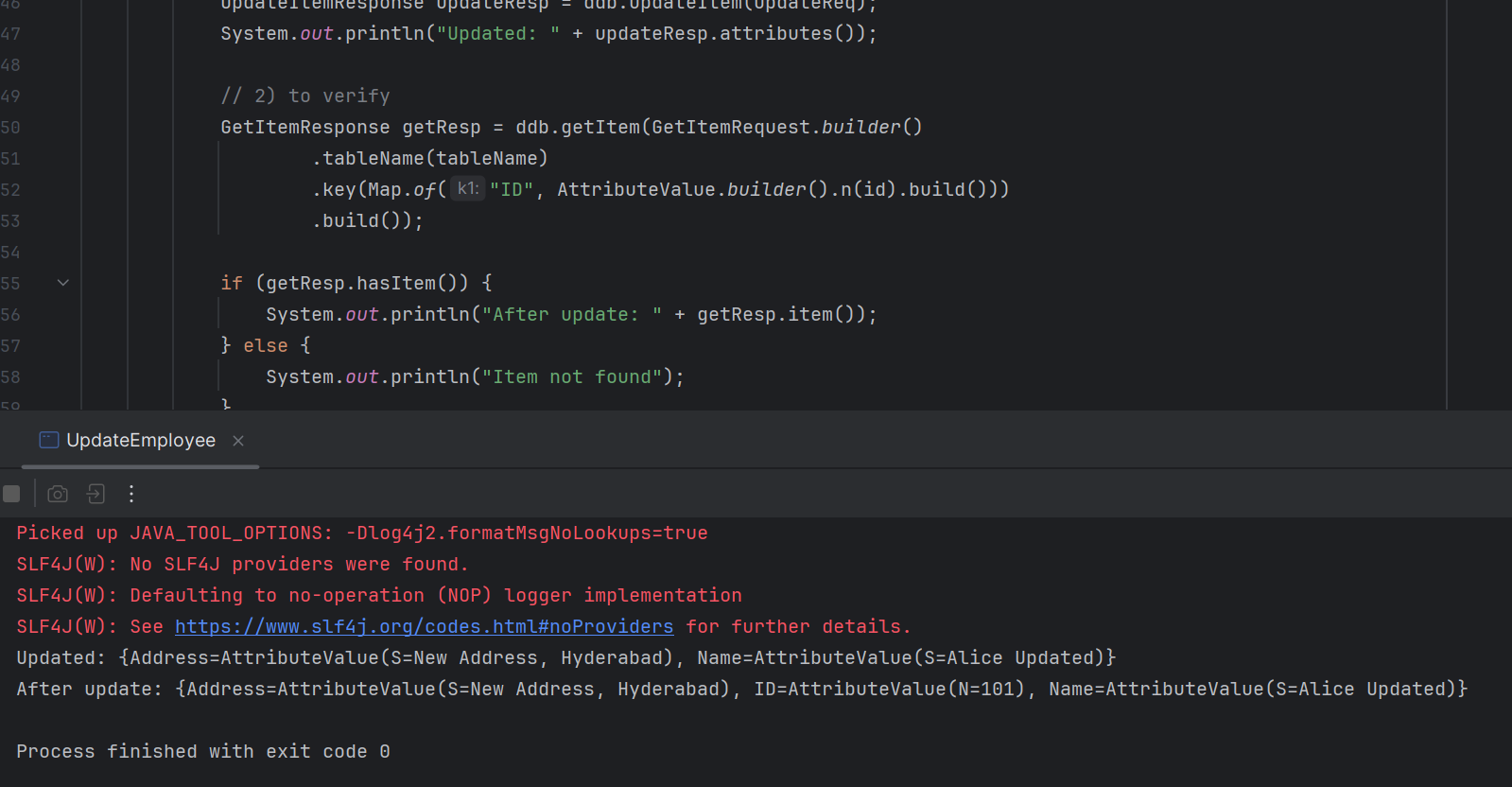




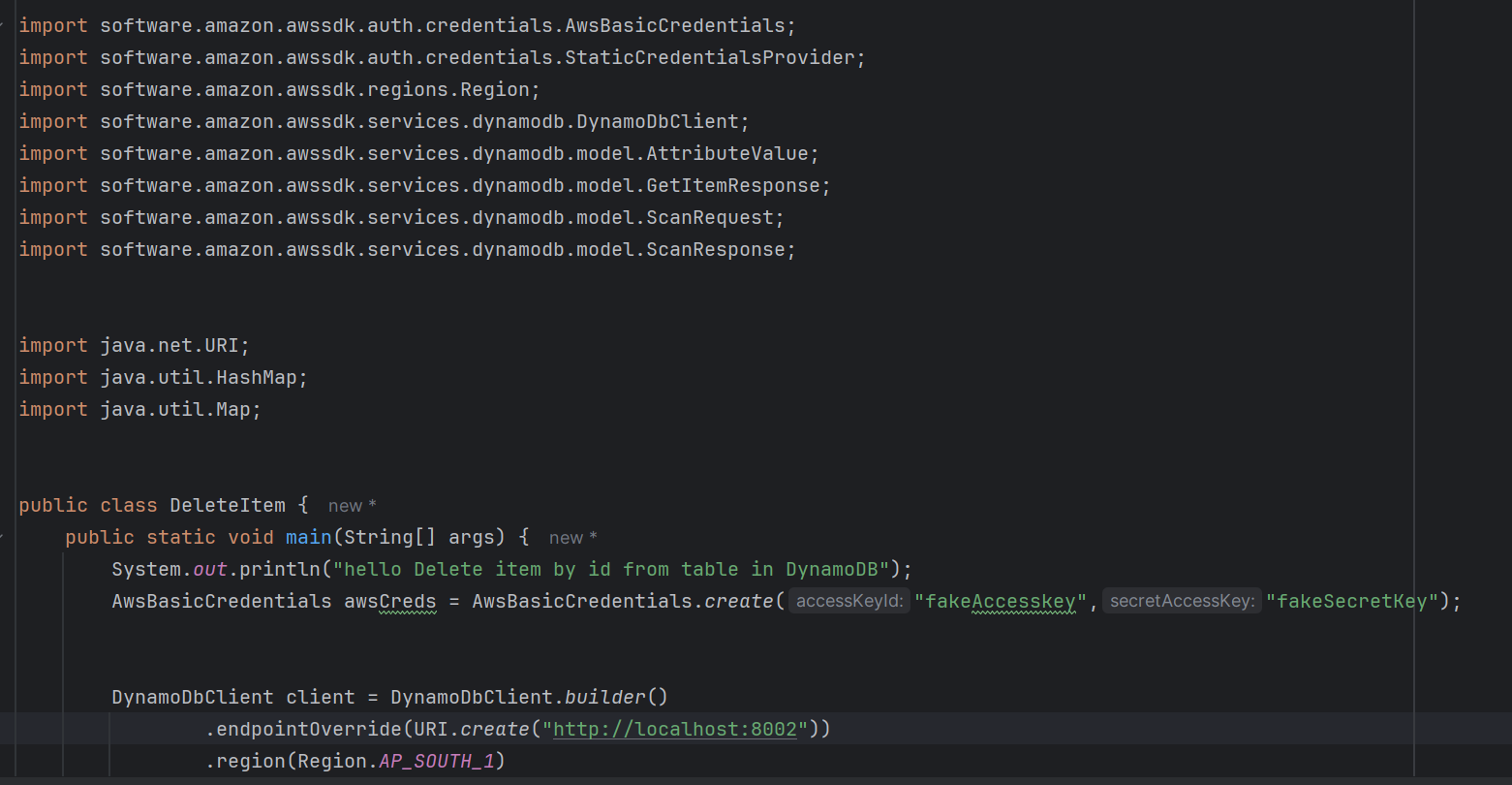
Update Employees



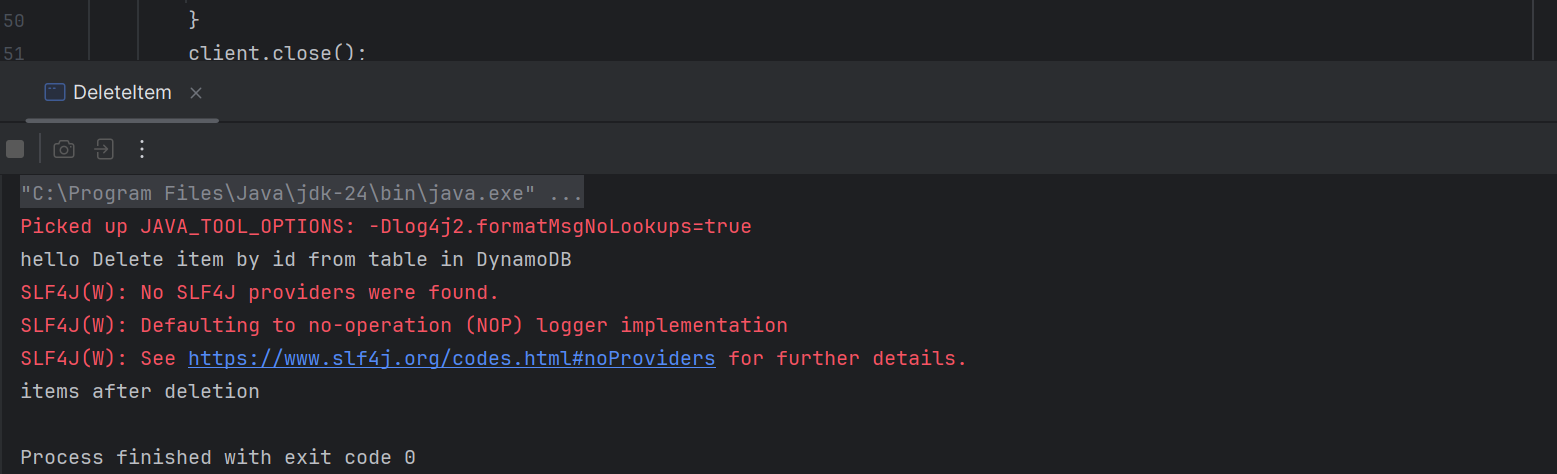




Delete Items







Task-7

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

**Ways to lower DynamoDB cost**

* Pick the right capacity mode
* On-demand: best for spiky/unknown traffic; avoid overprovisioning.
* Provisioned + Auto Scaling: best for steady traffic; set sane min/max to avoid paying for peaks 24/7.

**Reduce read/write units**

* Prefer Query over Scan. Scans read full partitions and are expensive.
* Use eventually consistent reads (½ the RCU of strong).
* Use pagination and fetch only what you need.
* Use BatchGet/BatchWrite to reduce round trips (still charged per item, but saves retries/overhead).

**Optimize data model and access**

* Choose a partition key that spreads traffic evenly; avoid hot partitions (prevents scaling RCUs/WCUs just to handle a hotspot).
* Use write sharding for extremely hot keys.
* Keep items small: short attribute names, remove unused fields, compress large JSON strings if appropriate. You