Day 5 – Store scores

## **Objective (one-liner for the doc)**

Persist evaluation scores and test-run metadata into **AWS DynamoDB**. This lets the system store results scalable & serverless.

## **High level design & table schema**

Table name: EvaluationResults

Primary Key:

* Partition key: studentId (String)
* Sort key: submissionId (String) — unique per submission, e.g. assignmentId\_timestamp or UUID

Attributes (per item):

* studentId (S) — partition key
* submissionId (S) — sort key (e.g., A001\_20250907T100000)
* assignmentId (S)
* score (N)
* maxScore (N) (optional)
* passedCount (N)
* totalTests (N)
* timestamp (S) — ISO8601
* results (S) — JSON string of detailed test results (or store separately)
* metadata (S) — JSON for any extra info (optional)

Why this schema?

* Query by studentId to get all submissions.
* Use submissionId to uniquely identify each run (and order by time).
* Storing JSON result lets you keep detailed info without a fixed schema.

1. **Add Maven dependencies (pom.xml)**

<!-- AWS SDK v2 DynamoDB -->

<dependency>

<groupId>software.amazon.awssdk</groupId>

<artifactId>dynamodb</artifactId>

<version>2.20.8</version>

</dependency>

<!-- DynamoDB Enhanced Client (convenience mapping) -->

<dependency>

<groupId>software.amazon.awssdk</groupId>

<artifactId>dynamodb-enhanced</artifactId>

<version>2.20.8</version>

</dependency>

<!-- (Optional) Jackson for JSON serialization -->

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

<version>2.15.2</version>

</dependency>

1. **Java code**

* 4.1 ScoreRecord.java
* 4.2 DynamoDbScoreRepository.java
* 4.3 — Integrate into EvaluationEngine

1. **Example wiring & demo (AtlasApplication update)**

package org.example;

import software.amazon.awssdk.regions.Region;

import java.util.Date;

public class AtlasApplication {

public static void main(String[] args) {

// If using DynamoDB Local:

String endpoint = "http://localhost:8000"; // set to null when using real AWS

Region region = Region.AP\_SOUTH\_1; // or Region.US\_EAST\_1

DynamoDbScoreRepository repo = new DynamoDbScoreRepository("EvaluationResults", region, endpoint);

// Optional: create table locally if doesn't exist

repo.createTableIfNotExists();

EvaluationEngine engine = new EvaluationEngine(repo);

Student student = new Student("S001", "Alice", "alice@example.com");

Assignment assignment = new Assignment("A001", "Java Basics", "Complete Java assignment", new Date(), "uploads/S001\_A001.txt");

CandidateSolution solution = new MockSolution();

int score = engine.evaluateAndPersist(student, assignment, solution);

System.out.println("Score persisted: " + score);

// Query saved records for demo

var records = repo.queryByStudentId("S001");

System.out.println("Records for S001: " + records.size());

repo.close();

}

}

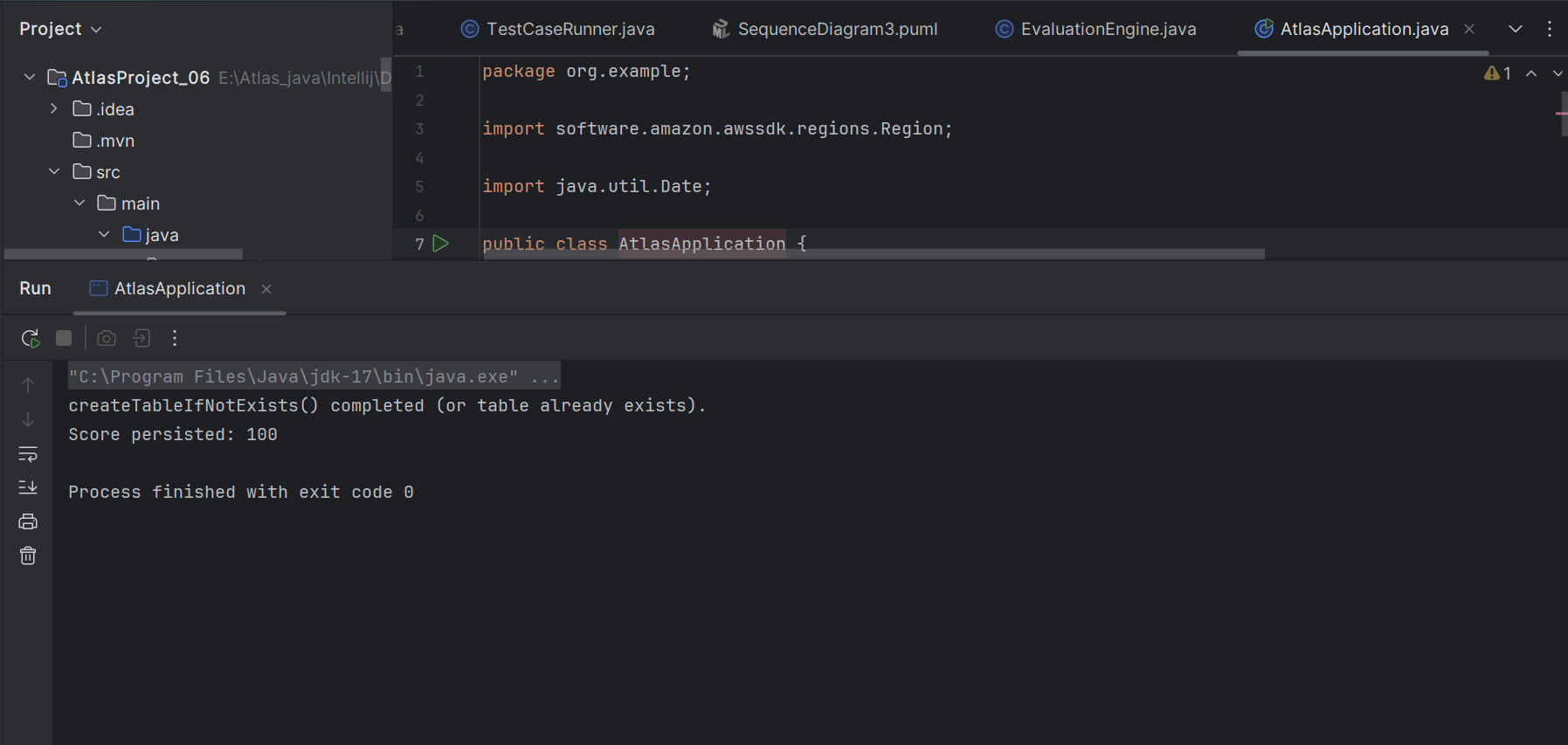
1. **Creating a table**

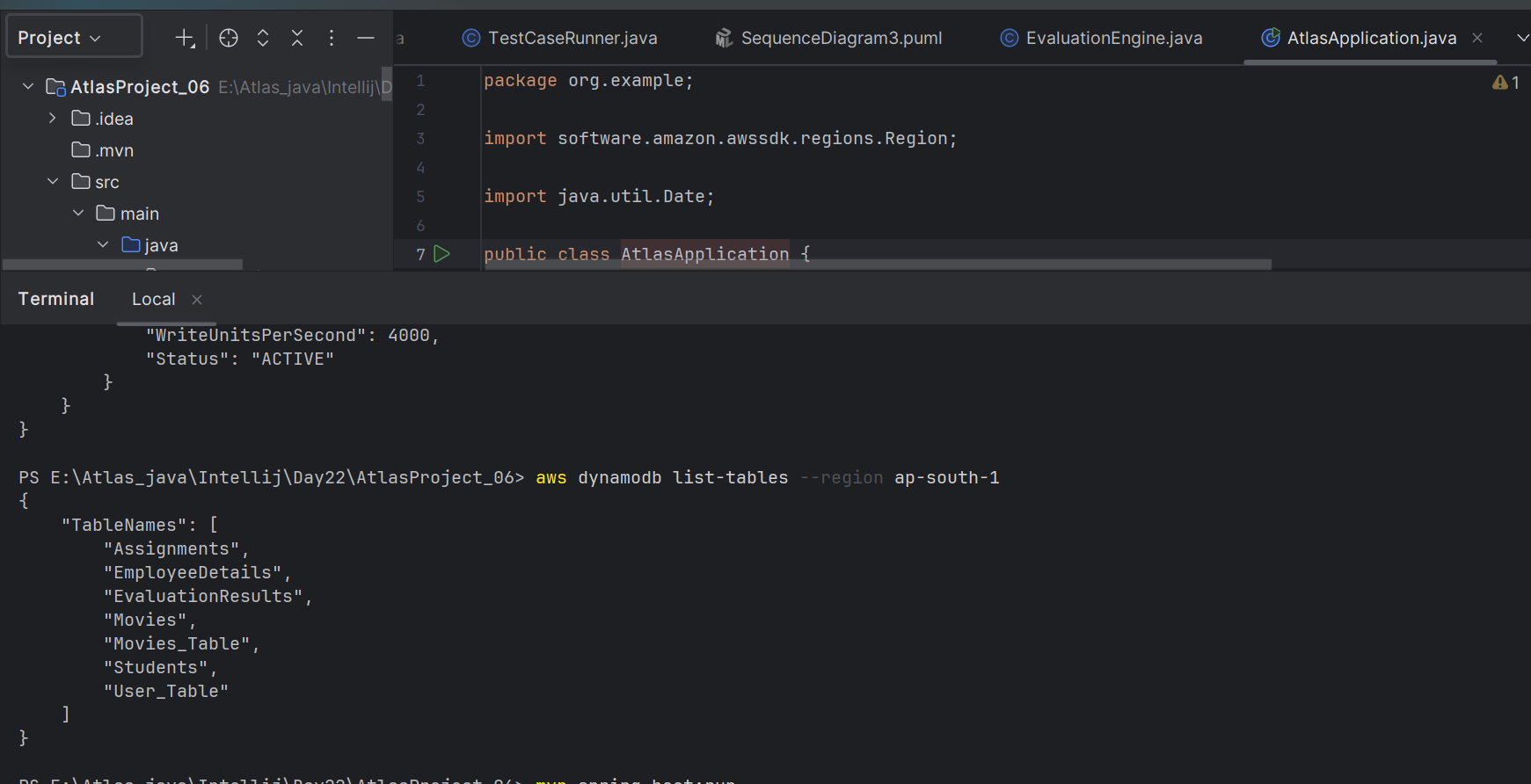
### Real AWS (Console or CLI)

Use AWS Console or the aws dynamodb create-table command same as above but omit --endpoint-url and set --region appropriately.

1. **Testing & verification**

In demo AtlasApplication you print the number of records — verify it increases after each run.

****

****

**Day 5 – Conclusion:**

* Implemented DynamoDB integration to persist evaluation results (replacing MySQL).
* Designed EvaluationResults table with studentId (partition key) and submissionId (sort key).
* Created ScoreRecord POJO and DynamoDbScoreRepository for saving & querying results.
* Updated EvaluationEngine to persist score, test counts and JSON test details immediately after evaluation.
* Included local development instructions using DynamoDB Local and sample CLI commands.