Assignment 4 Report

BugFreeGroup

Github Repo:

https://github.khoury.northeastern.edu/cs6650-24Spring/cs6650-assignment04

Server Design

The servlet listens to two main URL patterns:

- /skiers/*
- /resorts/*

Endpoints and Functions:

1. **Get Number of Unique Skiers (Resort/Season/Day)**

- Endpoint: GET /resorts/{resortID}/seasons/{seasonID}/day/{dayID}/skiers

- Function: Retrieves the number of unique skiers for a specific resort, season, and day. It verifies the path parameters and uses them to construct a key for querying a DynamoDB table to fetch the count of unique skiers.

2. Get Skier Day Verticals

- Endpoint: GET /skiers/{resortID}/seasons/{seasonID}/days/{dayID}/skiers/{skierID}

- Function: Retrieves the total vertical meters skied by a specific skier on a particular day at a specific resort during a specific season. It validates the parameters and fetches the data from DynamoDB.

3. Get Total Vertical for a Skier Across All Seasons

- Endpoint: GET /skiers/{skierID}/vertical

- Function: Retrieves the total vertical meters skied by a specific skier across all seasons at a specified resort. This endpoint processes requests by parsing the skier ID and optional query parameters for resort and season, and then queries the DynamoDB.

Database Design

Database: DynamoDB

Table Name: SkiersData

Primary Key:

Partition Key (ResortSeasonDayLiftId): This is a composite key representing a unique combination of the resort, season, day, and lift. It is designed as a string that concatenates these attributes, e.g., "1#2024#1#10".

Sort Key (SkierId): Represents the unique identifier for a skier. This allows for the efficient retrieval of all records related to a particular skier on a specific lift, during a particular day, in a specific season at a specific resort.

Attributes:
- Vertical: Represents the total vertical meters skied by the skier for the specific lift usage. This attribute is used to accumulate the vertical distance skied by a skier each time they take a lift.
- Time: Stores the time at which the skier took the lift. This can be used for detailed analytics and operational insights, such as determining peak usage times.

ResortSeasonDayLiftId *(String)*	SkierId *(Number)*	Time	Vertical
1#2024#1#9	245	149	90

Database Updates:
- UniqueSkiers: Updates a count of unique skiers using to handle counting unique visitors at a specific day.

ResortSeasonDayLiftId *(String)*	SkierId *(Number)*	DayVertical	UniqueSkiers
1#2024#3	-1		87090

- TotalVertical: Updates the total vertical distance skied for a specific skier over a season at a resort. If the update fails, it writes this data as a new record.

ResortSeasonDayLiftId *(String)*	SkierId *(Number)*	TotalVertical
1#2024	245	770

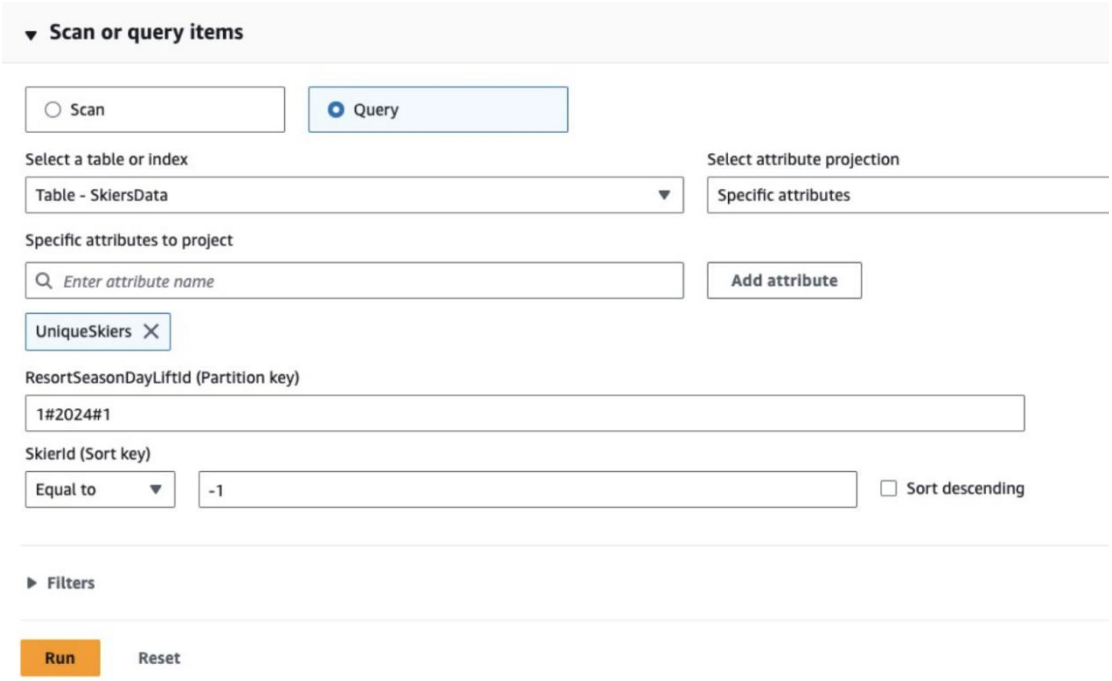
- DayVertical: Updates the vertical distance for a skier at a specific day. If the update fails, it writes this data as a new record.

ResortSeasonDayLiftId *(String)*	SkierId *(Number)*	DayVertical
1#2024#1	245	280

- Vertical Per Lift: Attempts to update the vertical distance for a specific lift on a specific day for a skier. If this fails, it inserts the ski record into the database.

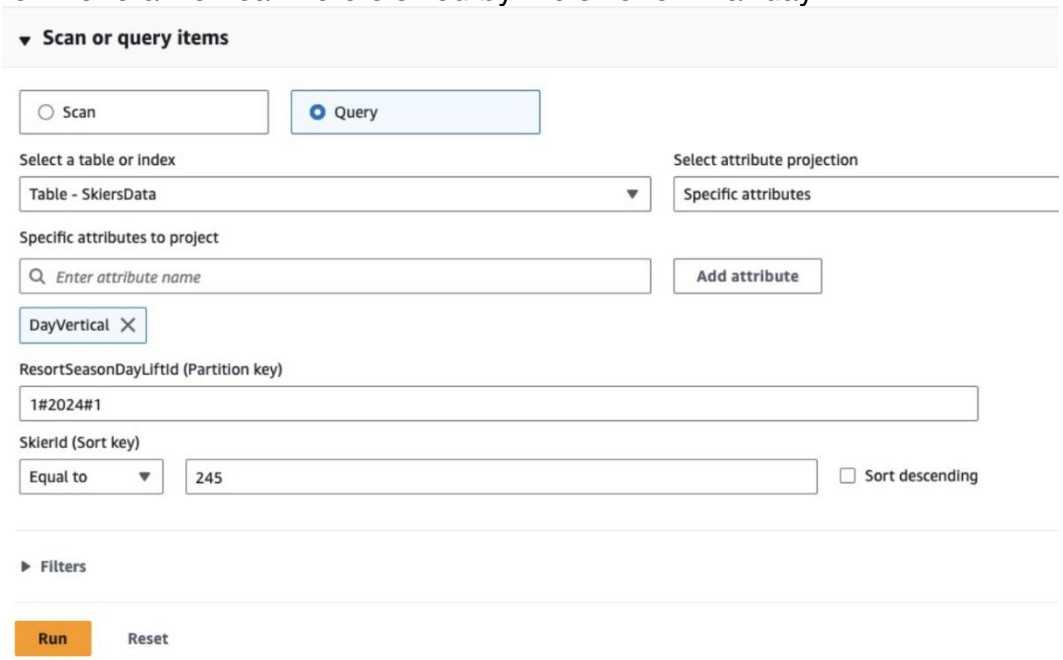
Database Interactions:
Update vs. Write: The system first attempts to update existing records. If no record exists, it falls back to writing a new record using writeVertical method.

Workflow for Data Retrieval
1. Get Number of Unique Skiers
Query DynamoDB: The server queries DynamoDB for a record with this key. The record would contain the count of unique skiers for the specified resort, season, and day.



2. Get Skier Day Verticals

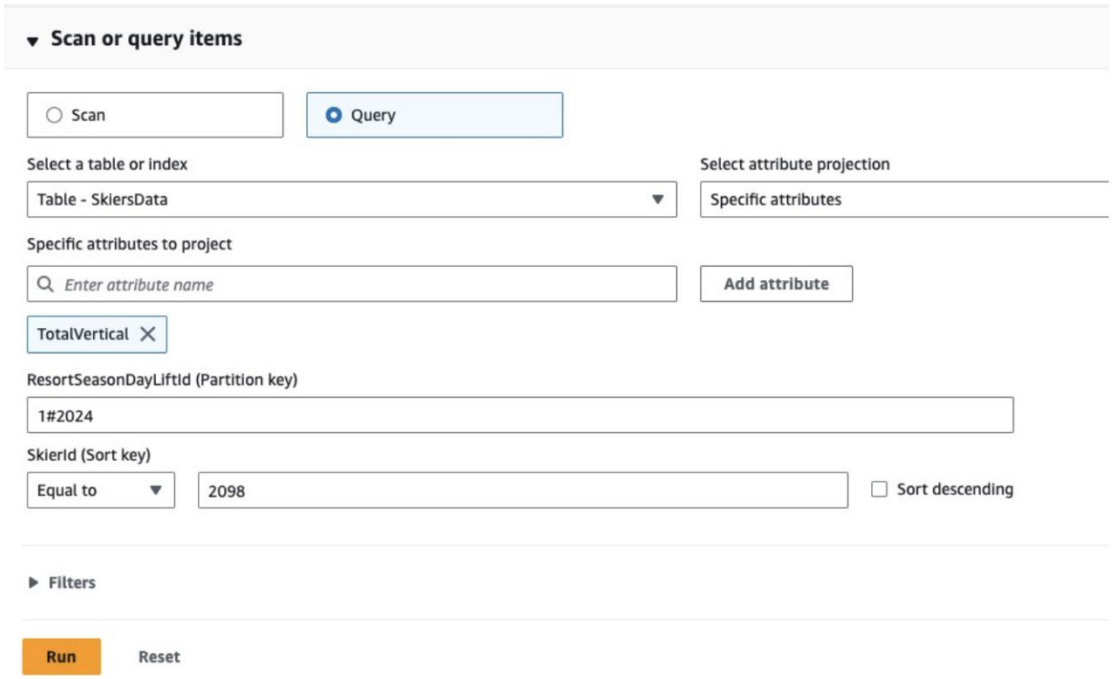
Query DynamoDB: Using the constructed key, the server queries DynamoDB for the total vertical meters skied by the skier on that day.



3. Get Total Vertical for a Skier Across All Seasons

Query DynamoDB: The server constructs a query based on the provided parameters. If specific resort and season are specified, it constructs a more precise query; otherwise, it might need to aggregate data across multiple records.

Aggregate Data: If the request is for total verticals across multiple seasons or resorts, the server might need to perform an aggregation of the verticals from multiple records.



JMeter Test Result