Name: Hiti Chouhan

Problem 1:

Figure out the database schema.

SOLUTION:

QUERY:

[gy3500@coe2 ~]$ aws dynamodb list-tables --endpoint-url http://localhost:8000

Output-

{

"TableNames": [

"Forum",

"MusicCollection",

"ProductCatalog",

"Reply",

"Thread"

]

}

QUERY:

aws dynamodb describe-table --table-name Forum --endpoint-url http://localhost:8000

Output-

{

"Table": {

"TableArn": "arn:aws:dynamodb:ddblocal:000000000000:table/Forum",

"AttributeDefinitions": [

{

"AttributeName": "Name",

"AttributeType": "S"

}

],

"ProvisionedThroughput": {

"NumberOfDecreasesToday": 0,

"WriteCapacityUnits": 5,

"LastIncreaseDateTime": 0.0,

"ReadCapacityUnits": 10,

"LastDecreaseDateTime": 0.0

},

"TableSizeBytes": 156,

"TableName": "Forum",

"TableStatus": "ACTIVE",

"KeySchema": [

{

"KeyType": "HASH",

"AttributeName": "Name"

}

],

"ItemCount": 4,

"CreationDateTime": 1582082173.269

}

}

The schema of the table Forum includes KeyType:HASH and AttributeName:Name.

We have a DynamoDB Table named Forum with a Name (HASH) key attribute and type of the attribute is String(AttributeType: "S"). It consists of Simple primary key:Name (String). Attribute of items used is 'name' and attribute type is 'String'.

Table name 'Forum' has a primary key

Partition key: Name (String)

A screenshot of a cell phone

Description automatically generated

Problem 2:  
Using AWS CLI and DynamoDB query functionality for simple data analysis,

Implementing the following queries for DynamoDB with AWS CLI that perform simple analytical tasks on the data stored in the database.

2.1. Obtain an item in the table ProductCatalog, for which the attribute Id is 102. Query returns consistent results.

Query :-

aws dynamodb get-item --consistent-read \

--table-name ProductCatalog \

--key '{ "Id": {"N": "102"} }' --endpoint-url http://localhost:8000

Output:

{

"Item": {

"ISBN": {

"S": "222-2222222222"

},

"Dimensions": {

"S": "8.5 x 11.0 x 0.8"

},

"Title": {

"S": "Book 102 Title"

},

"Price": {

"N": "20"

},

"PageCount": {

"N": "600"

},

"ProductCategory": {

"S": "Book"

},

"InPublication": {

"BOOL": true

},

"Authors": {

"L": [

{

"S": "Author1"

},

{

"S": "Author2"

}

]

},

"Id": {

"N": "102"

}

}

}

A screenshot of a cell phone

Description automatically generated

2.2. Obtain an item in the table Reply, in which the attribute Id (partition key) is "Amazon DynamoDB#DynamoDB Thread 2" and the attribute ReplyDateTime (sort key) is between October 1, 2015 and October 30, 2015.

Query -

aws dynamodb --endpoint-url http://localhost:8000 query --table-name Reply --key-condition-expression 'Id = :id and ReplyDateTime BETWEEN :dt1 AND :dt2' --expression-attribute-values '{

> ":id": {"S": "Amazon DynamoDB#DynamoDB Thread 2"},

> ":dt1": {"S": "2015-10-01"},

> ":dt2": {"S": "2015-10-31"}

> }'

OR

aws dynamodb --endpoint-url http://localhost:8000 query \

--table-name Reply \

--key-conditions '{

"Id": {

"ComparisonOperator": "EQ",

"AttributeValueList": [ {"S": "Amazon DynamoDB#DynamoDB Thread 2"} ]

}

}' \

--query-filter '{

"ReplyDateTime": {

"ComparisonOperator": "BETWEEN",

"AttributeValueList": [ {":dt1": {"S": "2015-10-01"},

":dt2": {"S": "2015-10-31"}} ]

}

}'

OUTPUT:

{

"Count": 1,

"Items": [

{

"PostedBy": {

"S": "User A"

},

"Message": {

"S": "DynamoDB Thread 2 Reply 2 text"

},

"ReplyDateTime": {

"S": "2015-10-05T19:58:22.947Z"

},

"Id": {

"S": "Amazon DynamoDB#DynamoDB Thread 2"

}

}

],

"ScannedCount": 1,

"ConsumedCapacity": null

}

A screenshot of a computer

Description automatically generated

2.3 2.3. The table Thread stores information regarding forum posts in the attributes ForumName (partition key), Subject (sort key), Views and Replies. Obtain item(s) in the table Thread, in which the attribute ForumName is “Amazon DynamoDB” and the attribute Replies is larger than 3.

Query:-

aws dynamodb --endpoint-url http://localhost:8000 query --table-name Thread --key-condition-expression 'ForumName = :a' --filter-expression 'Replies > :t1' --expression-attribute-values '{

> ":a": {"S": "Amazon DynamoDB"},

> ":t1": {"N": "3"}

>

> }'

Output:-

{

"Count": 1,

"Items": [

{

"Tags": {

"L": [

{

"S": "index"

},

{

"S": "primarykey"

},

{

"S": "table"

}

]

},

"LastPostedDateTime": {

"S": "2015-09-22T19:58:22.514Z"

},

"Views": {

"N": "6"

},

"LastPostedBy": {

"S": "User A"

},

"Answered": {

"N": "1"

},

"Replies": {

"N": "5"

},

"Message": {

"S": "DynamoDB thread 1 message"

},

"ForumName": {

"S": "Amazon DynamoDB"

},

"Subject": {

"S": "DynamoDB Thread 2"

}

}

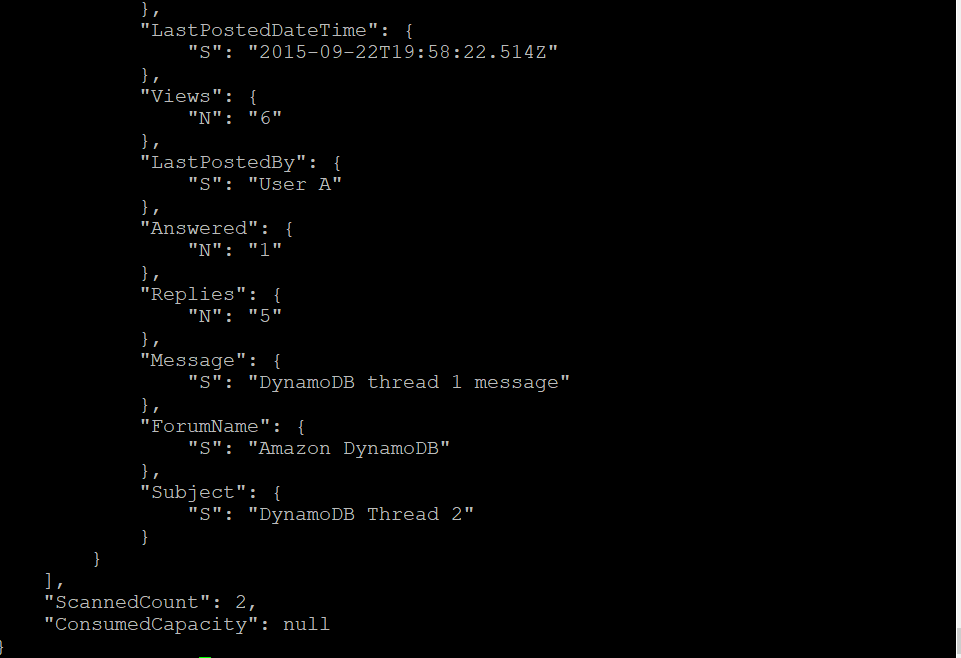
],

"ScannedCount": 2,

"ConsumedCapacity": null

}





2.4 Write a query to return the posts in the forum “Amazon DynamoDB” and the thread “DynamoDB Thread 1” that have been viewed at least once.

[gy3500@coe2 ~]$ aws dynamodb query --table-name Thread --key-condition-expression "ForumName = :name" --expression-attribute-values '{":name":{"S": "Amazon DynamoDB"}}' --endpoint-url http://localhost:8000

Output:

{

"Count": 2,

"Items": [

{

"Tags": {

"L": [

{

"S": "items"

},

{

"S": "attributes"

},

{

"S": "throughput"

}

]

},

"LastPostedDateTime": {

"S": "2015-09-15T19:58:22.514Z"

},

"Views": {

"N": "3"

},

"LastPostedBy": {

"S": "User A"

},

"Answered": {

"N": "1"

},

"Replies": {

"N": "2"

},

"Message": {

"S": "DynamoDB thread 2 message"

},

"ForumName": {

"S": "Amazon DynamoDB"

},

"Subject": {

"S": "DynamoDB Thread 1"

}

},

{

"Tags": {

"L": [

{

"S": "index"

},

{

"S": "primarykey"

},

{

"S": "table"

}

]

},

"LastPostedDateTime": {

"S": "2015-09-22T19:58:22.514Z"

},

"Views": {

"N": "6"

},

"LastPostedBy": {

"S": "User A"

},

"Answered": {

"N": "1"

},

"Replies": {

"N": "5"

},

"Message": {

"S": "DynamoDB thread 1 message"

},

"ForumName": {

"S": "Amazon DynamoDB"

},

"Subject": {

"S": "DynamoDB Thread 2"

}

}

],

"ScannedCount": 2,

"ConsumedCapacity": null

}



