Cloud9 terminal:

Partition key (required) + Sort key (optional)

1. List table and add data

- aws dynamodb list-tables
- aws dynamodb put-item --table-name Music --item file://song1.json

```
song1.json:
  {
    "Artist" : {"S" : "David Bowie"},
    "SongTitle" : {"S" : "Changes"},
    "AlbumTitle" : {"S" : "Hunky Dory"},
    "Genre" : {"S" : "Rock"}
  }
```

DynamoDB console:

Click on the 'Items' tab and refresh the page to see the data

Could9 terminal

1. Query

- aws dynamodb query --table-name Music --key-condition-expression "Artist = :v1" --expression-attribute-values file://values1.json

```
values1.json:
  {
    ":v1" : {"S" : "David Bowie"}
  }
```

- aws dynamodb query --table-name Music --key-condition-expression "Artist = :v1 AND SongTitle = :v2" --expression-attribute-values file://values2.json

```
values2.json:
  {
    ":v1" : {"S" : "David Bowie"},
    ":v2" : {"S" : "Heros"}
  }
```

2. Delete

- aws dynamodb delete-item --table-name Music --key file://keysToDelete.json

```
keysToDelete.json
  {
    "Artist" : {"S" : "David Bowie"},
    "SongTitle" : {"S" : "Changes"}
  }
```

3. Scan

pull back all data

- aws dynamodb scan --table-name Music

4. Filter with scan

go through every item of the table

- aws dynamodb query --table-name Music --key-condition-expression "Artist = :v1" --filter-expression "Released = :v2" --expression-attribute-values file://filterValues1.json

```
filterValues1.json
  {
    ":v1" : {"S" : "Bryan Adams"},
    ":v2" : {"S" : "1998"}
  }
```

- aws dynamodb scan --table-name Music --filter-expression "SongTitle = :v1" --expression-attribute-values file://filterValues2.json

```
filterValues2.json
  {
    ":v1" : {"S" : "Heroes"}
  }
```

5. Secondary index

- Local secondary index: allows to pick alternate sort key

  Global secondary index: allows to create alternate partition and sort key

- **Can't query non-key attribute without a scan**

Week 1 Notes and Resources

KEY TOPICS

Relational and NoSQL databases

For decades, the predominant data model that was used for application development was the relational data model used by relational databases such as Oracle, DB2, SQL Server, MySQL, and PostgreSQL. It wasn't until the mid to late 2000s that other data models began to gain significant adoption and usage. To differentiate and categorize these new classes of databases and data models, the term "NoSQL" was coined. Often the term "NoSQL" is used interchangeably with "nonrelational."

Neither model is necessarily better, just more appropriate for particular application and workload requirements. More information about Relational and NoSQL databases can be found here.

Amazon DynamoDB

Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multiregion, multimaster database with built-in security, backup and restore, and in-memory caching for internet-scale applications. DynamoDB can handle more than 10 trillion requests per day and can support peaks of more than 20 million requests per second.

DynamoDB lets you offload the administrative burdens of operating and scaling a distributed database, so that you don't have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling.

Accessing and Setting up DynamoDB via the AWS Console

You can access the AWS Management Console for DynamoDB here: https://console.aws.amazon.com/dynamodb/home.

You can use the console to do the following in DynamoDB:

- Monitor recent alerts, total capacity, service health, and the latest DynamoDB news on the DynamoDB dashboard.
- Create, update, and delete tables. The capacity calculator provides estimates of how many capacity units to request based on the usage information you provide.
- Manage streams.
- View, add, update, and delete items that are stored in tables. Manage Time To Live (TTL) to

define when items in a table expire so that they can be automatically deleted from the database.
- Query and scan a table.
- Set up and view alarms to monitor your table's capacity usage. View your table's top monitoring metrics on real-time graphs from CloudWatch.
- Modify a table's provisioned capacity.
- Create and delete global secondary indexes.
- Create triggers to connect DynamoDB streams to AWS Lambda functions.
- Apply tags to your resources to help organize and identify them.
- Purchase reserved capacity.

Additional information on using the DynamoDB with the AWS Management Console can be found here.

Using the CLI to access DynamoDB

You can use the AWS Command Line Interface (AWS CLI) to control multiple AWS services from the command line and automate them through scripts. You can use the AWS CLI for ad hoc operations, such as creating a table. You can also use it to embed DynamoDB operations within utility scripts.

The command line format consists of a DynamoDB operation name, followed by the parameters for that operation. The AWS CLI supports a shorthand syntax for the parameter values, as well as JSON. See the Amazon DynamoDB CLI documentation for details.

ADDITIONAL SERVICES/CONCEPTS

Developing in the Cloud with AWS Cloud9

AWS Cloud9 is a cloud-based integrated development environment (IDE) that lets you write, run, and debug your code with just a browser. It includes a code editor, debugger, and terminal. You can run this development environment on a managed Amazon Elastic Compute Cloud (Amazon EC2) instance that automatically sleeps when you are not using it.

Follow the exercise directions carefully when setting up your AWS Cloud9 instance.

CAP Theorem

The CAP theorem states posits that in a distributed system such as Amazon DynamoDB, it is impossible to have more than two out of the following three:

- [Consistency](https://en.wikipedia.org/wiki/Consistency_(database_systems)
- Availability
- Partition Tolerance

Amazon DynamoDB provides tool to help manage read consistency.

WHAT YOU ACCOMPLISHED THIS WEEK

- You created your AWS Cloud9 instance.

- You created a DynamoDB table and populated it with game data
- You began to build out your application