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**