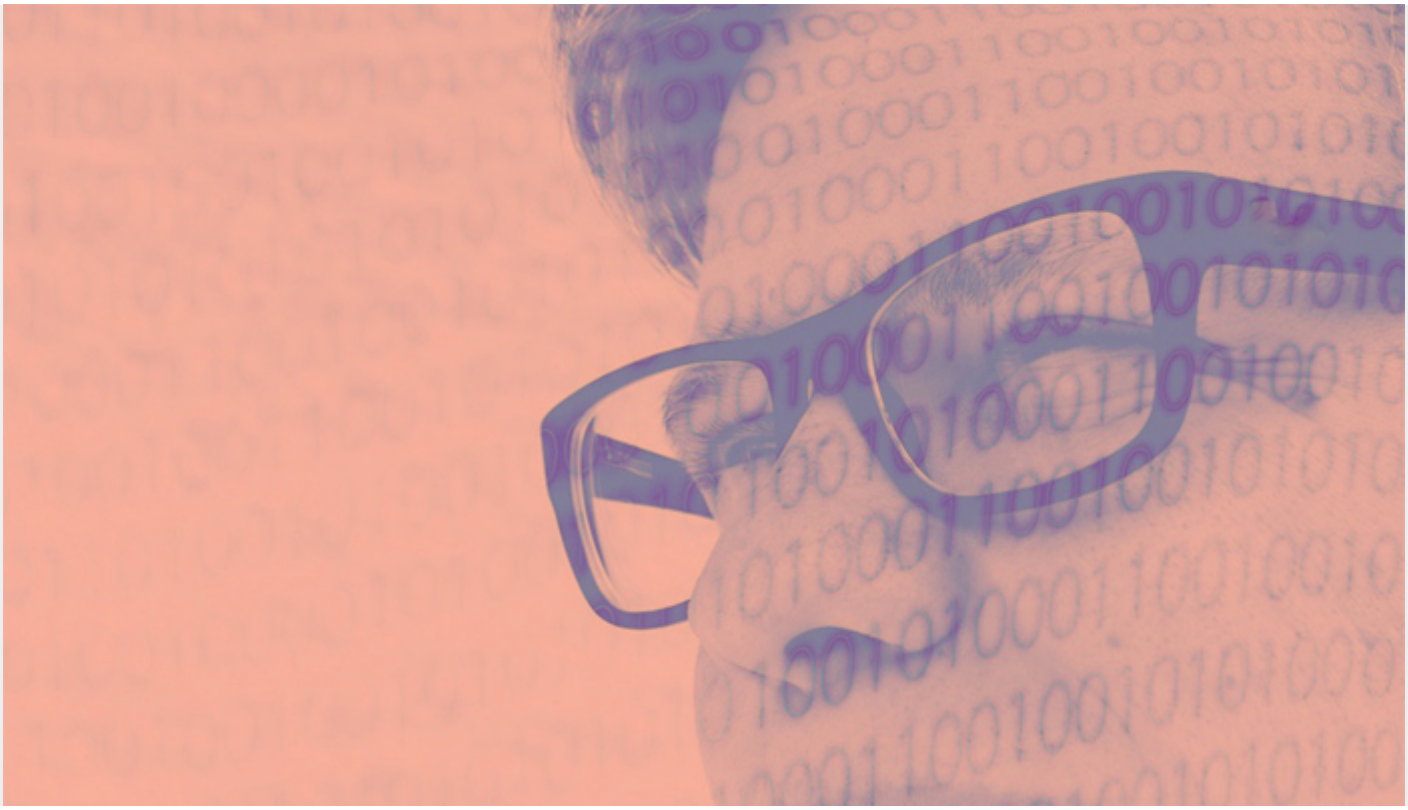
DYNAMODB BULK INSERT: AN EASY TUTORIAL



In this article, we'll show how to do bulk inserts in DynamoDB. If you're new to Amazon DynamoDB, start with these resources:

- Introduction to Amazon DynamoDB
- How To Add Data to Amazon DynamoDB
- How To Query Amazon DynamoDB

(This tutorial is part of our DynamoDB Guide. Use the right-hand menu to navigate.)

Bulk inserts and deletes

DynamoDB can handle bulk inserts and bulk deletes. We use the CLI since it's language agnostic. The file can be up to 16 MB but cannot have more than 25 request operations in one file.

Request operations can be:

- PutRequest
- DeleteRequest

The bulk request does not handle updates.

Data from IMDB

To illustrate, we have pulled 24 items from the IMDB (Internet Movie Database) and put them into JSON format. You can download that data from here.

The format for the bulk operation is:

```
{ "table name:
}
```

Here is an example:

```
{
        "title":
}
```

If you are running DynamoDB locally then start it like this:

```
java -Djava.library.path=./DynamoDBLoc_lib -jar DynamoDBLocal.jar -sharedDb
```

Create a table like this:

```
aws dynamodb create-table \
    --table-name title \
    --attribute-definitions AttributeName=tconst,AttributeType=S \
    --key-schema AttributeName=tconst,KeyType=HASH \
    --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5 \
--endpoint-url http://localhost:8000
```

Then load the data like this, having saved the IMDB data in the file 100.basics.json.

```
aws dynamodb batch-write-item \
         --endpoint-url http://localhost:8000 \
          --request-items
file:////Users/walkerrowe/Documents/imdb/100.basics.json \
         --return-consumed-capacity  TOTAL \
        --return-item-collection-metrics  SIZE
```

It responds:

```
{
    "UnprocessedItems": {},
    "ConsumedCapacity":
}
```

It told you how many records it wrote. You can query that it worked like this:

```
aws dynamodb query  \
     --endpoint-url http://localhost:8000 \
     --table-name title    \
```

```
    --key-condition-expression "tconst = :tconst"     \
    --expression-attribute-values '{ ":tconst":{"S":"tt0276132"}}'
```

Attribute Types and AttributeValue

Here we show some of the **AttributeValues**, meaning attribute or data types supported by DynamoDB. Those are:

- S
- BOOL
- L
- M
- etc.

Note: Even with numeric values you wrap them in quotes.

attribute type	description
S	String Notice that a date is in ISO-8601 value like this:"currentTime": {"S": "2020-07-24T09:25:49+0000" }
BOOL	Boolean. Use **true** or **false**.

L	A list of values without any AttributeValue, meaning no attribute name:
	"other": {
	"L":
	}

M	Map, containing attribute values. This is like a JSON object, except it has attribute values. So, it's like a list of named attributes.
	"map": {
	"M": {"Name": {"S": "Joe"},
	"Age": {"N": "35"}}
	}
	}

Here is an example showing how to use those DynamoDB attribute types.

```
{
        "title":
                                },
                                "actors": {
                                        "SS":
                                },
                                "currentTime": {
                                        "S": "2020-07-24T09:25:49+0000"
                                },
                                "other": {
```

```
                                    "L":
                                    },
                                    "map": {
                                    "M": {"Name": {"S": "Joe"}, "Age": {"N":
"35"}}
                                    }
                            }
                    }
            }]
}
```

Additional resources

For more on this topic, explore the BMC Machine Learning & Big Data Blog and these resources:

- AWS Guide, with 15+ articles and tutorials on AWS
- Availability Regions and Zones for AWS, Azure & GCP
- Databases on AWS: How Cloud Databases Fit in a Multi-Cloud World
- An Introduction to Database Reliability