

# EXHIBIT 9

## Item Sizes and Capacity Unit Consumption

### Topics

- Capacity Unit Consumption for Reads (p. 297)
- Capacity Unit Consumption for Writes (p. 298)

Before you choose read and write capacity settings for your table, you should first understand your data and how your application will access it. These inputs can help you determine your table's overall storage and throughput needs, and how much throughput capacity your application will require. DynamoDB tables are schemaless, except for the primary key, so the items in a table can all have different attributes, sizes, and data types.

The total size of an item is the sum of the lengths of its attribute names and values. You can use the following guidelines to estimate attribute sizes:

- Strings are Unicode with UTF-8 binary encoding. The size of a string is *(length of attribute name) + (number of UTF-8-encoded bytes)*.
- Numbers are variable length, with up to 38 significant digits. Leading and trailing zeroes are trimmed. The size of a number is approximately *(length of attribute name) + (1 byte per two significant digits) + (1 byte)*.
- A binary value must be encoded in base64 format before it can be sent to DynamoDB, but the value's raw byte length is used for calculating size. The size of a binary attribute is *(length of attribute name) + (number of raw bytes)*.
- The size of a null attribute or a Boolean attribute is *(length of attribute name) + (1 byte)*.
- An attribute of type List or Map requires 3 bytes of overhead, regardless of its contents. The size of a List or Map is *(length of attribute name) + sum (size of nested elements) + (3 bytes)*. The size of an empty List or Map is *(length of attribute name) + (3 bytes)*.

### Note

We recommend that you choose shorter attribute names rather than long ones. This will help you optimize capacity unit consumption and reduce the amount of storage required for your data.

## Capacity Unit Consumption for Reads

The following describes how DynamoDB read operations consume read capacity units:

- **GetItem**—reads a single item from a table. To determine the number of capacity units GetItem will consume, take the item size and round it up to the next 4 KB boundary. If you specified a strongly consistent read, this is the number of capacity units required. For an eventually consistent read (the default), take this number and divide it by two.

For example, if you read an item that is 3.5 KB, DynamoDB rounds the item size to 4 KB. If you read an item of 10 KB, DynamoDB rounds the item size to 12 KB.

- **BatchGetItem**—reads up to 100 items, from one or more tables. DynamoDB processes each item in the batch as an individual GetItem request, so DynamoDB first rounds up the size of each item to the next 4 KB boundary, and then calculates the total size. The result is not necessarily the same as the total size of all the items. For example, if BatchGetItem reads a 1.5 KB item and a 6.5 KB item, DynamoDB will calculate the size as 12 KB (4 KB + 8 KB), not 8 KB (1.5 KB + 6.5 KB).
- **Query**—reads multiple items that have the same partition key value. All of the items returned are treated as a single read operation, where DynamoDB computes the total size of all items and then rounds up to the next 4 KB boundary. For example, suppose your query returns 10 items whose