## <http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.CoreComponents.html>

## Primary Key = Partition Key Composite Primary Key = Partition Key + Sort Key or Range Key

When you create a table, in addition to the table name, you must specify the primary key of the table. The primary key uniquely identifies each item in the table, so that no two items can have the same key.

DynamoDB supports two different kinds of primary keys:

* **Partition key** – A simple primary key, composed of one attribute known as the partition key.

DynamoDB uses the partition key's value as input to an internal hash function. The output from the hash function determines the partition (physical storage internal to DynamoDB) in which the item will be stored.

In a table that has only a partition key, no two items can have the same partition key value.

The People table described in [Tables, Items, and Attributes](http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.CoreComponents.html#HowItWorks.CoreComponents.TablesItemsAttributes) is an example of a table with a simple primary key (PersonID). You can access any item in the People table immediately by providing the PersonId value for that item.

* **Partition key and sort key** – Referred to as a composite primary key, this type of key is composed of two attributes. The first attribute is the partition key, and the second attribute is the sort key.

DynamoDB uses the partition key value as input to an internal hash function. The output from the hash function determines the partition (physical storage internal to DynamoDB) in which the item will be stored. All items with the same partition key are stored together, in sorted order by sort key value.

In a table that has a partition key and a sort key, it's possible for two items to have the same partition key value. However, those two items must have different sort key values.

The Music table described in [Tables, Items, and Attributes](http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.CoreComponents.html#HowItWorks.CoreComponents.TablesItemsAttributes) is an example of a table with a composite primary key (Artist and SongTitle). You can access any item in the Music table immediately, if you provide the Artistand SongTitle values for that item.

A composite primary key gives you additional flexibility when querying data. For example, if you provide only the value for Artist, DynamoDB retrieves all of the songs by that artist. You could even provide a value for Artist and a range of SongTitle values, to retrieve only a subset of songs by a particular artist.

**Note**

The partition key of an item is also known as its hash attribute. The term hash attribute derives from the use of an internal hash function in DynamoDB that evenly distributes data items across partitions, based on their partition key values.

The sort key of an item is also known as its range attribute. The term range attribute derives from the way DynamoDB stores items with the same partition key physically close together, in sorted order by the sort key value.

Each primary key attribute must be a scalar (meaning that it can hold only a single value). The only data types allowed for primary key attributes are string, number, or binary. There are no such restrictions for other, non-key attributes.