



DynamoDB Core Components

1

Introductoin

- 1 In Amazon DynamoDB, data is organized into tables, where each table stores items (similar to rows in traditional databases)
- 2 Each item contains attributes (data fields), and every item is uniquely identified by a Primary Key
- 3 The primary key can be a Partition Key alone or a combination of Partition Key and Sort Key

2

Table

- 1 The first step to work with Amazon DynamoDB is to create a table
- 2 The top-level container that holds your data.
- 3 It is similar to a table in relational databases but designed for high performance and scalability

4

Table

5 Example Table for an Order Management Application

```
{
  "CustomerID": "C001",
  "OrderID": "O001",
  "OrderDate": "2024-12-01",
  "OrderAmount": 150.00,
  "OrderStatus": "Delivered"
}
```

CustomerID	OrderID	OrderDate	OrderAmount	OrderStatus
C001	O001	2024-12-01	150.00	Delivered
C002	O002	2024-12-03	250.00	Pending
C003	O003	2024-12-05	100.00	Delivered

3

Items

- 1 An item represents a single record in a DynamoDB table
- 2 It is similar to a row in a relational database
- 3 The table's Primary Key is used to uniquely identify each item

4 Example Table for an Order Management Application

Items

CustomerID	OrderID	OrderDate	OrderAmount	OrderStatus
C001	O001	2024-12-01	150.00	Delivered
C002	O002	2024-12-03	250.00	Pending
C003	O003	2024-12-05	100.00	Delivered

4

Attributes

- 1 Attributes are the individual pieces of data within an item
- 2 They are similar to columns in a relational database.
- 3 Attributes can store various types of data, including strings, numbers, lists, maps, and more
- 4 DynamoDB does not enforce a fixed schema for attributes
- 5 You can define attributes dynamically when adding records, which means items in the same table can have different attributes

6 Example Table for an Order Management Application

Items

Attributes

CustomerID	OrderID	OrderDate	OrderAmount	OrderStatus
C001	O001	2024-12-01	150.00	Delivered
C002	O002	2024-12-03	250.00	Pending
C003	O003	2024-12-05	100.00	Delivered

5

DynamoDB Keys

1 Primary Key

- 1 When you create a table, in addition to the table name, you must specify the primary key of the table
- 2 The primary key uniquely identifies each item in the table, so that no two items can have the same key
- 3 DynamoDB supports two different kinds of primary keys
 - 1 Partition key(Simple Primary Key)
 - 2 Partition Key + Sort Key (Composite Primary Key)

2 Partition key(Simple Primary Key)

- 1 A simple primary key, composed of one attribute known as the partition key
- 2 DynamoDB uses the partition key's value as input to an internal hash function.
- 3 The output from the hash function determines the partition (physical storage internal to DynamoDB) in which the item will be stored
- 4 In a table that has only a partition key, no two items can have the same partition key value

5 Example Table for an Order Management Application with Partition key

Partition key(Simple Primary Key)

CustomerID (Partition Key)	OrderID	OrderDate	OrderAmount	OrderStatus
C001	O001	2024-12-01	150.00	Delivered
C002	O002	2024-12-03	250.00	Pending
C003	O003	2024-12-05	100.00	Delivered

3 Partition Key + Sort Key (Composite Primary Key)

- 1 Referred to as a composite primary key, this type of key is composed of two attributes
- 2 The first attribute is the partition key, and the second attribute is the sort key
- 3 In a table that has a partition key and a sort key, it's possible for multiple items to have the same partition key value.
- 4 However, those items must have different sort key values

5 Example Table for an Order Management Application with Partition key & Sort k

Partition Key + Sort Key (Composite Primary Key)

CustomerID (Partition Key)	OrderID (Sort Key)	OrderDate	OrderAmount	OrderStatus
C001	O001	2024-12-01	150.00	Delivered
C001	O004	2024-12-03	200.00	Pending
C002	O002	2024-12-05	250.00	Delivered
C003	O003	2024-12-06	100.00	Shipped