Creating Simple DynamoDB Tables

Stefan Roman
DEVOPS ENGINEER

www.katapult.cloud

What Is DynamoDB?

Fully managed noSQL database service

Designed with elasticity in mind

Highly redundant data storage

Basic Table Components

Item

Equivalent to a row in relational databases

Attribute

Equivalent to a column in relational databases

Table

Same as relational database

Basic Table Components

```
{
   "Name": "Bob",
   "Age": 26,
   "Company": "Globomantics",
   "Position": "DevOps"
}
```

Basic Table Components

```
{
   "Name": "Peter",
   ...
},
{
   "Name": "Jane",
   ...
},
```

```
{
   "Name": "Lucy",
   ...
},
{
   "Name": "Andy",
   ...
},
```

DynamoDB Primary Keys

```
{
  "Name": "Bob",
  "Age": 26,
  "Company": "Globomantics",
  "Position": "DevOps"
}
```

DynamoDB Primary Keys

```
{
  "Name": "Bob",
  "Age": 26,
  "Company": "Globomantics",
  "Position": "DevOps"
}
```

```
{
  "Name": "Bob",
  "Age": 44,
  "Company": "Globomantics",
  "Position": "Administrator"
}
```

DynamoDB Primary Keys

```
{
  "Name": "Bob",
  "Age": 26,
  "Company": "Globomantics",
  "Position": "DevOps"
}
```

```
{
  "Name": "Bob",
  "Age": 44,
  "Company": "Globomantics",
  "Position": "Administrator"
}
```

DynamoDB Primary Keys

Simple Primary Key

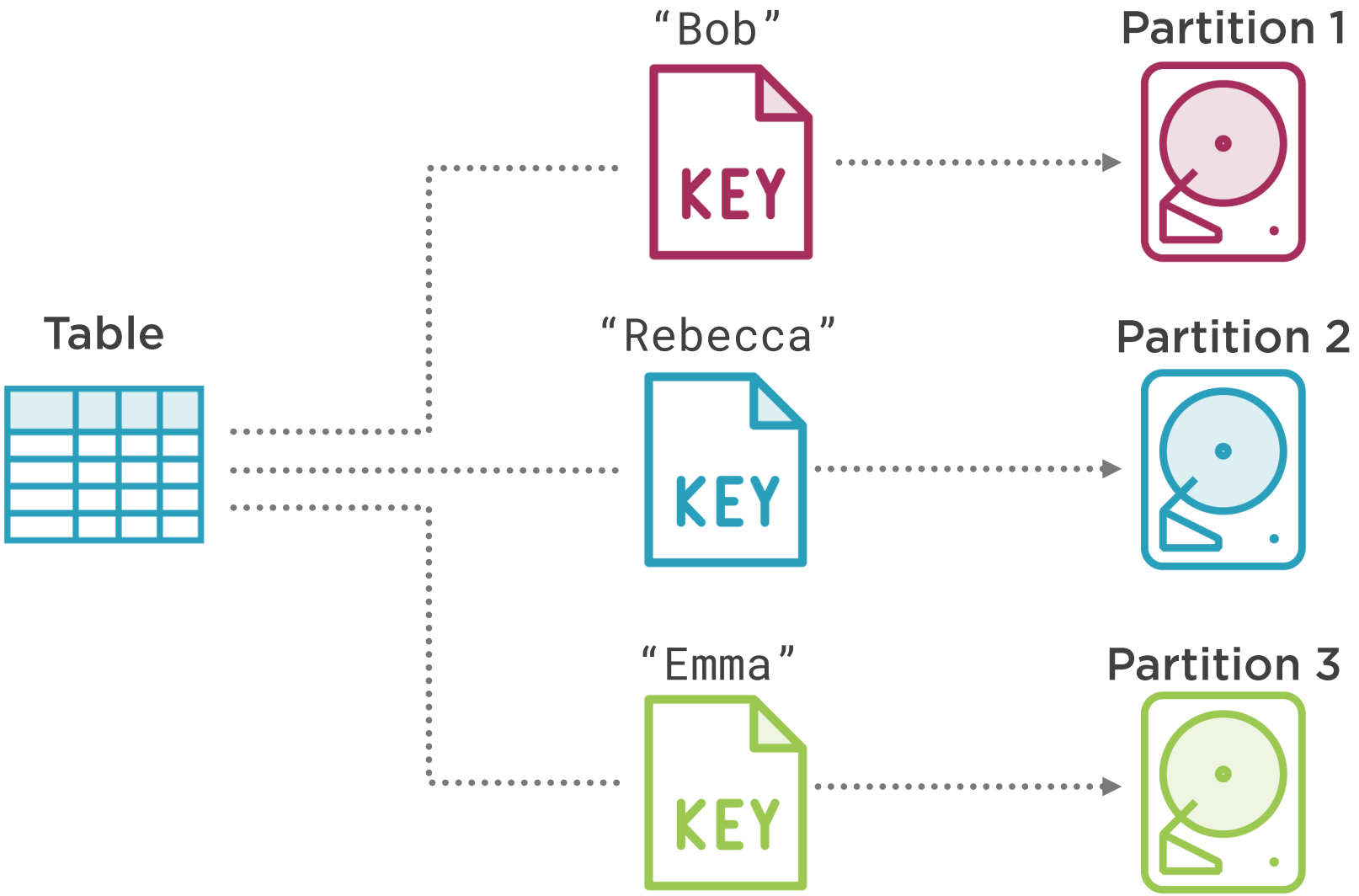
Partition Key

Composite Primary Key

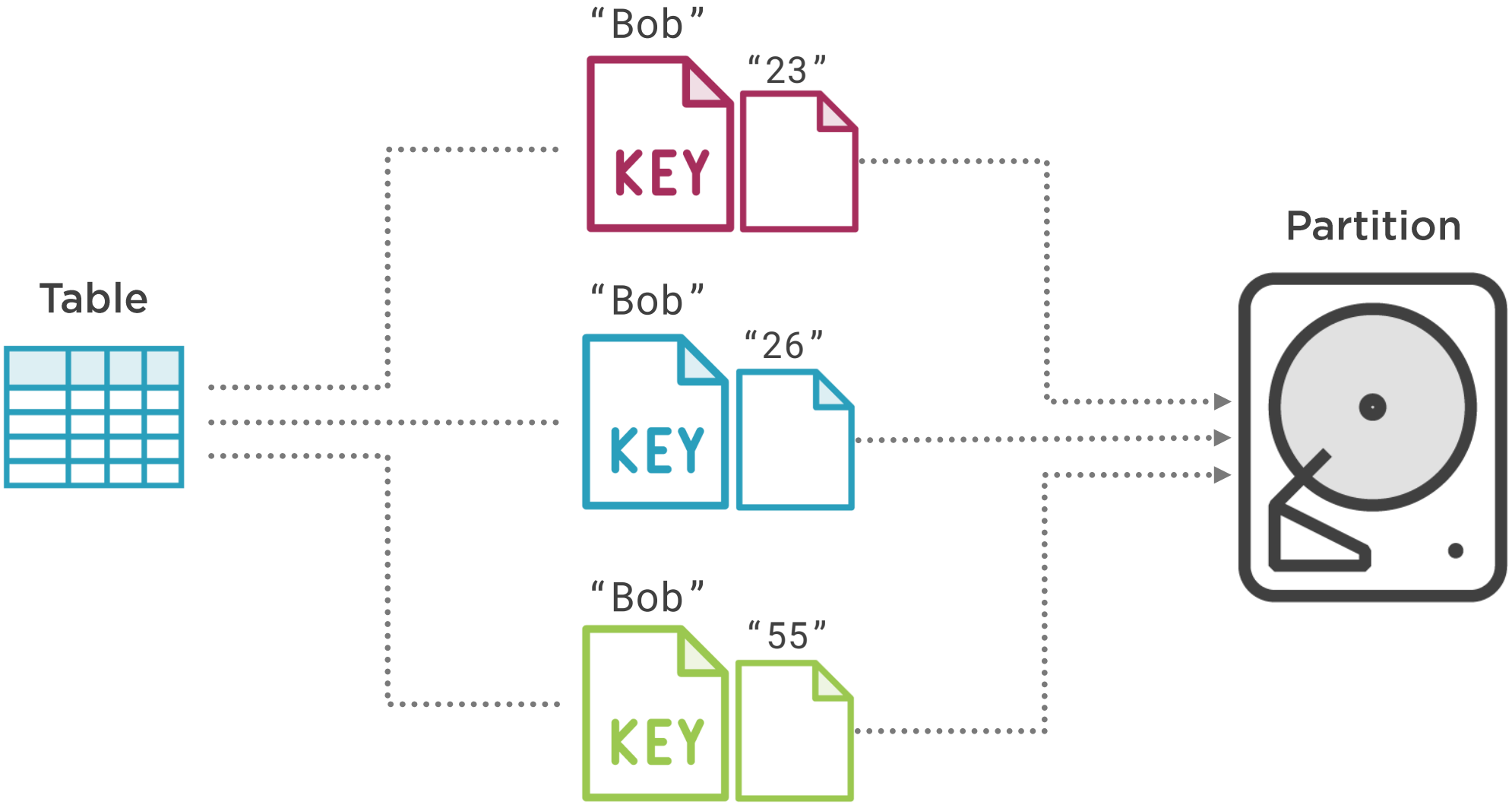
Partition Key

Sort Key

Understanding DynamoDB Partitioning



Understanding DynamoDB Partitioning



Allowed Datatypes

Scalar

One value at a time

Document

Multiple values of different types

Set

Multiple values of the same type

Scalar

```
{
   "Name": "Bob",
   "Age": 26,
   "Company": Null,
   "Employed": False
}
```

- String
- Integer
- Null
- Boolean

Document Type

```
{
   "Name": "Bob",
   "Employed": False
   "Hobbies": [
      ”Skiing”,
      ”Baseball”,
      3.14
   ]
}
```

◀ List

Document Type

```
{
   "Name": "Bob",
   "KitchenTools": [
      { "Spoons": 5 },
      “Pan”,
      “Dishwasher”
   ]
}
```

◀ Map

Sets

```
{
   "Name": "Bob",
   "LotteryNumbers": [
      34,
      22,
      45
   ]
}
```

- Integer set

Sets

```
{
   "Name": "Bob",
   "Kids": [
      “Rebecca”,
      “Emma”,
      “Andy”
   ]
}
```

- String set

```
{
    "Name": "Bob",
    "Secret": "c2VjcmV0",
    "LifeSecrets": [
        "amFuYQ==",
        "bHVjaWE=",
        "cGV0ZXI=",
        "YW5kcmVq"
    ]
}
```

Binary Data

- Binary scalar
- Binary set

Allowed Datatypes

```
{
  "Name": "Bob",
  "Kids": [
      "Rebecca",
      "Emma",
      "Andy"
  ]
}
```

Allowed Datatypes

```
{
   "Name": "Jane"
}
```

Allowed Datatypes

```
{
   "Name": "Peter",
   "KitchenItems": {
       "Spoons": 7,
       "Forks": 10,
       "Knives": 4
    }
}
```

Demo

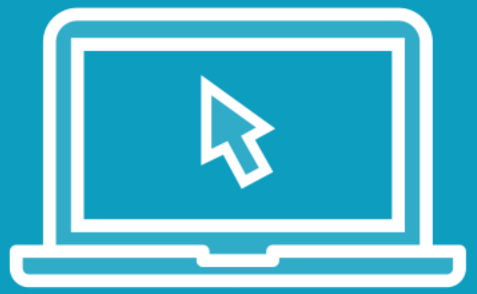


Table prerequisites:

- Table name – `simple_employee_table`
- Partition Key – `Name`
- Sort Key - `Age`

Summary

What is DynamoDB

Basic Table Components

DynamoDB Primary Keys

Storage Partitioning

Allowed Datatypes

Creating Simple Table