Agenda

- **Introduction to DynamoDB**
- **Lab**

NoSQL - Dynamo DB

- The concept of NoSQL databases became popular with Internet giants like Google, Facebook, Amazon, etc. who deal with huge volume of data.
- The system response time becomes slow when you use RDBMS for massive volumes of data.
- To resolve this problem, we could “scale up” our systems by upgrading our existing hardware. This process in expensive.
- The alternative for this issue is to distribute dates load on multiple hosts whenever the load increases. This method is know as “scaling out”.

What is Dynamo DB ?

- It flexible data model and reliable performance make it great fit for mobile, web, gaming, ad-tech, IoT, Andy many other applications.
- Transactions provide atomicity, Consistency, isolation, and durability (ACId in DynamoDB)
- The data in DynamoDB is stored in JSOn format.
- It is fully managed database. And supports both document and key-value data models.
- Millions of requests per second, trillions of rows, 100s of TB storage.
- Fast and consistent in performance (low latency on retrieval)

NoSQL - Dynamo DB

- NoSQL means, Not only SQL.
- NoSQL databases are non-relational databases (no fixed columns etc) and are distributed (horizontal scaling).
- NoSQL databases include MongoDB, DynamoDB etc
- NoSQL databases do not support joins
- All the data that is needed for a query is present in one row
- NoSQL databases don’t perform aggregations such as “SUM”
- NoSQL databases scale horizontally

SQL vs NoSQL databases

SQL	NoSQL
SQL is generally used in RDBMS	NoSQL is used for Non-Relational database system
Structured data can be stored in tables	Using JSON data, unstructured data can be stored
The Schemas are static	The Schemas are dynamic
Schemas are rigid and bound to relationships	Schemas are non-rigid, they are flexible
Helpful to design complex queries	No interface to prepare complex queries
Hare we call tables, rows and columns	Here we call collections, collections has documents
MySql, Oracle, Sqlite, Postgres and MS-SQL	MongoDB, BigTable, Redis, RavenDb, Cassandra, Hbase, Neo4j and CouchDb

Use cases

- **Gaming**
- DynamoDB can store analytical and game state records for multiplayer games.

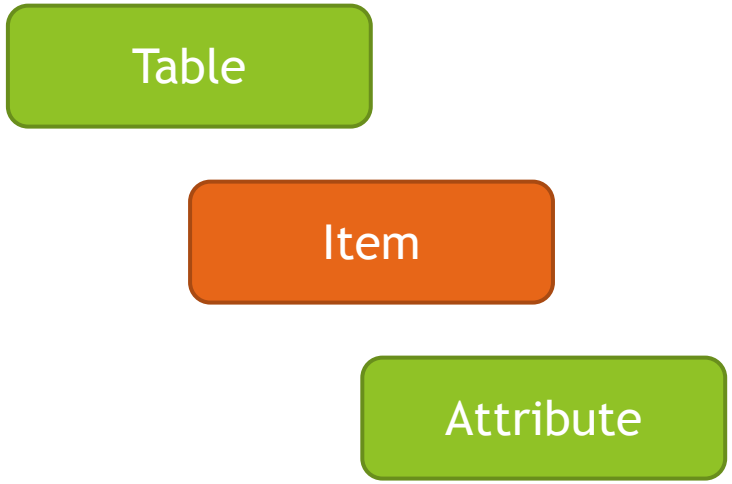
- **Entertainment**
- DynamoDB can process and store analytical data to identify customer likes and dislikes.

- **Social media**
- DynamoDB can provide access to data with single-digit millisecond latency.

- **Financial services**
- DynamoDB can store and process decision-making and results for loan approvals, and validate and persist equities trade orders

Core components of DynamoDB

- In DynamoDB, the collection of items is known as a table. A table in DynamoDB is not a structured table with fixed number of cells or columns.



Table

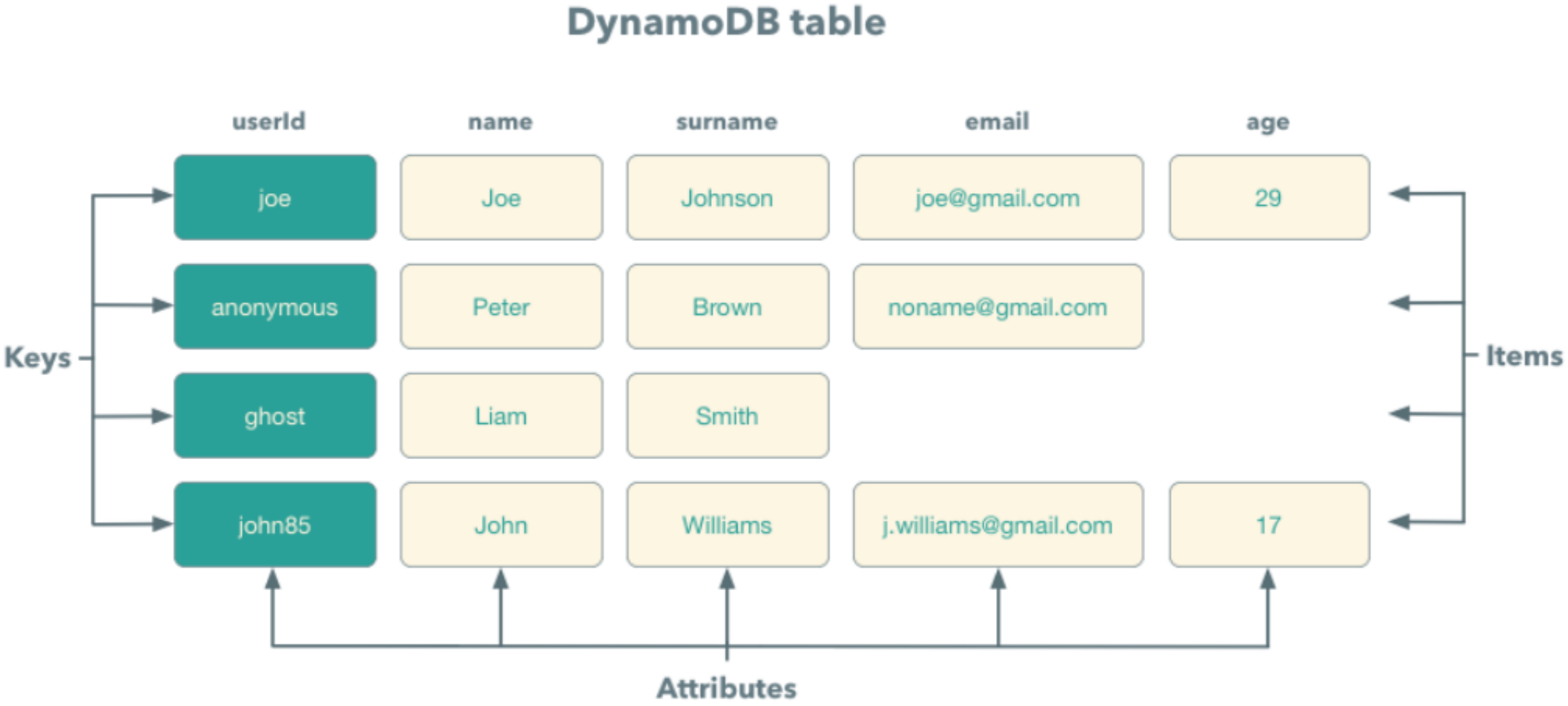
The collection of items is know as a table.

Item

Each table in DynamoDB contains one or more than one items.

Attribute

Attributes in DynamoDB are fundamental data elements or values that reside in item.


DynamoDB table
userId
name
surname
email
age
joe
Joe
Johnson
joe@gmail.com
29
anonymous
Peter
Brown
noname@gmail.com
ghost
Liam
Smith
john85
John
Williams
j.williams@gmail.com
17
Keys
Items
Attributes

Read Consistency of DynamoDB

- **Eventual Consistent Reads**
  - Consistency across all copies of data is usually reached within a second. Repeating a read after a short time should return the updated data.

- **Strongly Consistent Reads**
  - A strongly consistent read returns a result that reflects all writes that received a successful response prior to the read.

Provisioning Capacity

- **Example**
- **Read Capacity Requirements**
  - If you have a table and you want to read 100 items per second with strongly consistent reads and your items are 8KB in size, you would calculate the required provisioned capacity as follows.
  - 8KB/4KB = 2 capacity units
  - 2 read capacity units per item x 100 reads per seconds = 200/1=200 read capacity units
  - Note : Eventual consistent Reads would require 200/2=100 read capacity units

- **Write Capacity Requirements**
  - If you have a table and you want to write 50 items per second and your items are 4 KB in size, you would calculate the required provisioned capacity as follows:
  - 4KB/1KB = 4 capacity units
  - 4 write capacity units per item x 50 writes per second = 200/1 = 200 write capacity units

DynamoDB API

- DynamoDB operations has to be done by provided APIs
  - Create Table – Create a new table. Can be used to create indexes as well.
  - Describe Table: Returns information about tables.
  - List Tables: Returns all tables.
  - Scan : retrieves all of the items in the specified table in index.
  - Delete Item: Deletes a single item with a specific Primary Key.