

AMAZON

DYNAMO DB

Amazon Dynamo DB is key value kind of databases that uses tables to save data.

TABLES:

Tables comprises of columns and rows that contains the data.

ROWS:

Rows in Amazon DynamoDB are called as items, there is no restrictions on number of items in a table.

COLUMNS:

Columns in DynamoDB are called as attributes, there can be any number of attributes of any row or item in a table.

Difference Between Relational Database and Amazon DynamoDB:

Amazon DynamoDB is schema free that is there is no need to predefine any sort of values like number of columns, column length etc.

There can be any number of attributes of different items which is not possible in RDBMS.

The latency is in single digit millisecond which is very less than that of RDBMS.

Throughput for all scales remains constant.

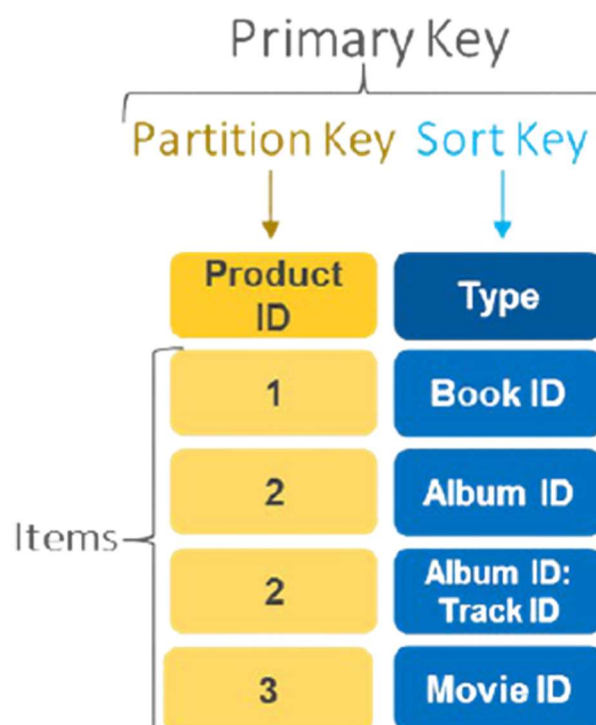
Primary Key in Amazon DynamoDB:

There are two basic keys in amazon DynamoDB 1) Partition key. 2)Sort Key(optional).

1) 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.

Partition key is a key that specifies in which logical partition the data is to be stored.



Partition key can be numeric, string, etc. but needs to be declared at the time of table creation.

The partition key needs to be unique if sort key is not declared.

In the above case product ID (partition key) "2" is repeated but it has different sort key.

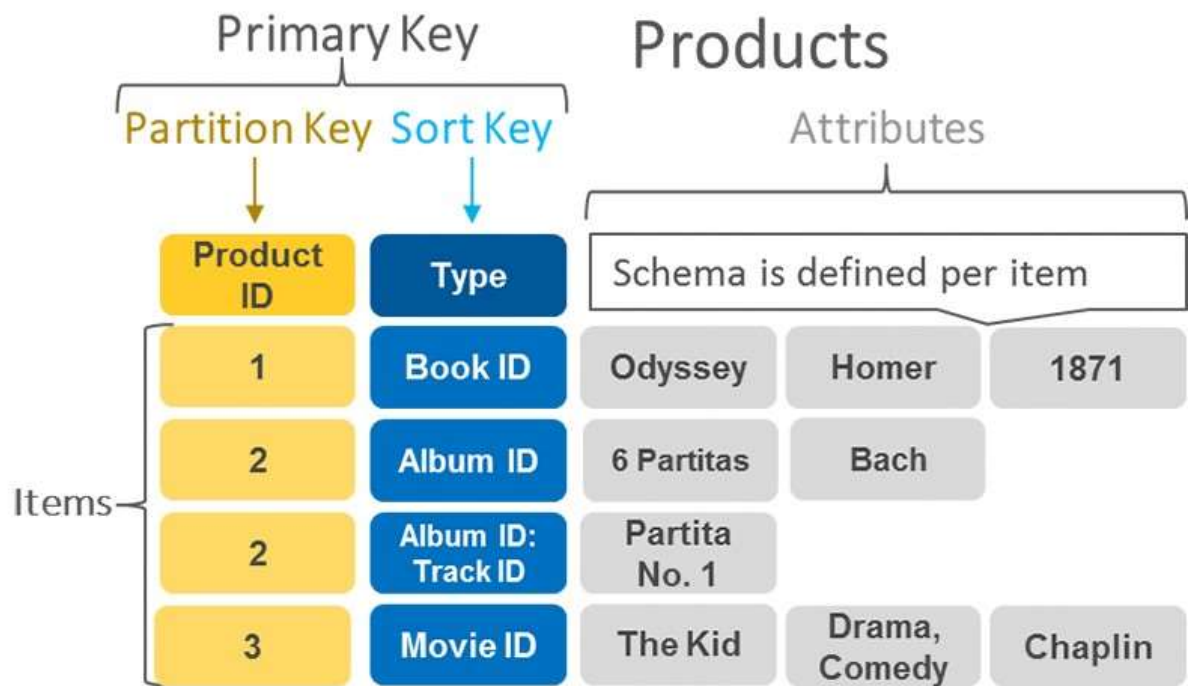
Partition key value is required to enter when inserting new item to table.

2) Sort Key (Range 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 value are stored together, in sorted order by sort key value.

Sort key works at partition level i.e. inside a partition we can narrow down the search and insert query using sort key.

The combination of Partition key and Sort key acts as a primary key in Amazon DynamoDB.



What Happens Behind the Scenes?

Once we have data with partition key and a sort key in the database, when we try to retrieve it first the query will check in which partition is the data located, then if there are multiple items in the partition it will then check for sort key and narrow down the data items using sort key and then give the output.

Hence, we can say that Partition and Sort key proves important when we need to search or enter a data fast and efficiently.

THROTTLING:

Read Write Capacity Units:

It is the number of users that can at a time read or write data to the database, there is cost difference if we use more or less capacity units.

Now if I have assigned 200 read/write capacity units for a database table with two partitions, the DB will assign 100 R/WCU to partition 1 and 100 to partition 2. But if 2nd partition is getting more than 100 say 200 Read requests then in such case DB would start rejecting the requests which is called as **throttling**.

A new update **adaptive capacity** was made that gives an option for a partition to borrow some capacity units from neighbor partition if the traffic increases.

Create Table DynamoDB:

Create DynamoDB table

Tutorial ?

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* ⓘ

Primary key* Partition key

ⓘ

☒ Add sort key

ⓘ

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

☐ Use default settings

Secondary indexes

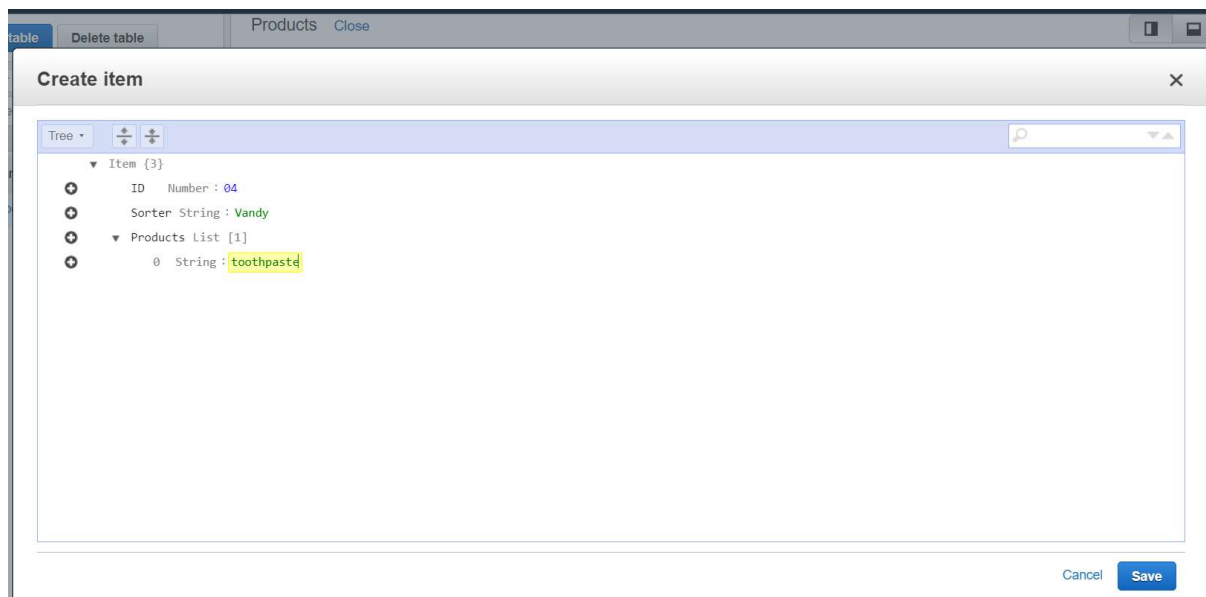
Name	Type	Partition key	Sort key	Projected Attributes
+ Add index				

Read/write capacity mode

Select on-demand if you want to pay only for the read and writes you perform, with no capacity planning required. Select provisioned to save on throughput costs if you can reliably estimate your application's throughput requirements. See the [DynamoDB pricing page](#) and [DynamoDB Developer Guide](#) to learn more.

As we can see while creating a table even using a GUI we need to specify a table name, partition key, sort key if needed. Then we can create table.

Data Insertion:



The partition and sort key are the only compulsory items else we can add any type of data string, list, numbers etc. and add to the table.

DATA RETRIVAL:

There are two methods to retrieve data from database table.

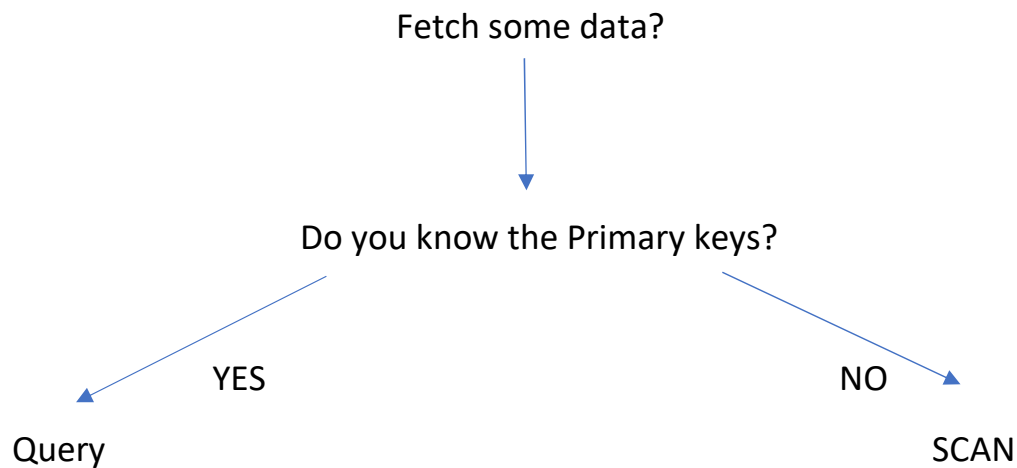
1) SCAN:

Scanning is looking through whole Database for a row that matches the filter.

2) Query:

Querying is method to search for row directly using partition and primary key.

When should we use what?



QUERY vs SCAN:

Query is actually a fast retrieval as the database knows which partition and range is located at which physical position.

Scan on the other hand scans all the data every time.

Query is cost efficient as the cost billed is on the basis of data scanned and not returned, scan method goes through more data at a time and hence require higher cost.

Secondary Indices:

DynamoDB provides us with two secondary Indexes,

- 1) Global Secondary Index.
- 2) Local Secondary Index.

Global Secondary Index (GSI):

Global secondary index is an index that have a partition key and an optional sort key that are different from base table's primary key. It is deemed "global" because queries on the index can access the data across different partitions of the base table. It can view as a different table that contains attributes based on the base table.

Local Secondary Index (LSI):

Local secondary index is an index that must have the same partition key but a different sort key from the base table. It is considered "local" because every partition of a local secondary index is bounded by the same partition key value of the base table. It enables data query with different sorting order of the specified sort key attribute.

Primary Key		
Partition Key	Sort Key	Attribute
Accountid	CreationDate	OriginCountry
1	2019-09-30	USA
2	2019-10-01	Canada
3	2019-10-02	USA
4	2019-10-03	Germany

Primary Key		
Partition Key	Sort Key	Attribute
OriginCountry	CreationDate	Accountid
USA	2019-09-30	1
Canada	2019-10-01	2
USA	2019-10-02	3
Germany	2019-10-03	4

In the above example partition key is account id now if I want data of all accounts created in Germany, I will have to use attribute Origin Country which is not a key and hence the DB will scan all the items of attribute Origin Country and then return the one's in Germany.

But if I created a GSI on attribute Origin Country DB will now create a table with partition key as origin country and it will be faster and easier to get the data in this case original table is kept as it is.

ForumName	LastPostDateTime	Subject
"S3"	"2015-01-03:09:21:11"	"ddd"
"S3"	"2015-01-22:23:18:01"	"bbb"
"S3"	"2015-02-31:13:14:21"	"ccc"
"S3"	"2015-03-15:17:24:31"	"aaa"
"EC2"	"2015-01-18:07:33:42"	"zzz"
"EC2"	"2015-02-12:11:07:56"	"yyy"
"RDS"	"2015-01-19:01:13:24"	"rrr"
"RDS"	"2015-02-22:12:19:44"	"ttt"
"RDS"	"2015-03-11:06:53:00"	"sss"

In the above example Forum name is partition key, LastPostDateTime is the sort key and subject are normal attribute. In case we know the Partition Key and but rather than searching with the date if user wants to search the partition with subject, he/she can assign subject as LSI and hence can now query subjects too as if it is the sort key.

Use Cases Amazon DynamoDB:

- 1) **Duolingo:** Language Learning app uses DynamoDB to store the data.
- 2) **Major League Baseball:** Uses DynamoDB to store the real time images, sounds that require fast read and write capacity offered by DynamoDB.
- 3) **Hess Corporation:** Hess Corporation, a well-known energy company, has been working on the exploration and production of natural gas and crude oil. This business requires strategizing different financial planning which impacts management on the whole. To streamline their business processes, Hess turned towards DynamoDB by shifting its E&P (Energy Exploration and Production) project onto AWS.
- 4) **DOCOMO:** popular mobile operating company, has built a reputation for its voice recognition services which need best performance provided by DynamoDB.

/*****/

Note: Google, Aws documentation, Google images were used as a reference in the above explanation.