HI-TECH INSTITUTION

CORPORATE CAREER ENHANCEMENT TRAININGS

OUR ROOT LEVEL TRAINING WILL GIVE YOU BETTER GROWTH

Hi-Tech Institution

ABOUT US

Our Vision:

To provide better training by full filing the requirements of our trainee.

Our Mission:

We always ensure to give practical based training. And we make the candidates to get good hands-on experience on any platform.

Philosophy:

Our Root Level Training Will give you Better Growth.

We successfully survived around 5 years in the IT field. Started this is as small Training room. But now we are having 5 branches across India.

Certified Trainers taking the session on various domain with any level of doubts clarification.

For More Details: **www.hitechins.in**

Write feedback to **operations@hitechins.in**

Amazon DynamoDB

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. DynamoDB lets you offload the administrative burdens of operating and scaling a distributed database, so that you don't have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling. Also, DynamoDB offers encryption at rest, which eliminates the operational burden and complexity involved in protecting sensitive data.

With DynamoDB, you can create database tables that can store and retrieve any amount of data, and serve any level of request traffic. You can scale up or scale down your tables' throughput capacity without downtime or performance degradation, and use the AWS Management Console to monitor resource utilization and performance metrics.

Amazon DynamoDB provides on-demand backup capability. It allows you to create full backups of your tables for long-term retention and archival for regulatory compliance needs.

DynamoDB Core Components

In DynamoDB, tables, items, and attributes are the core components that you work with. A *table* is a collection of *items*, and each item is a collection of *attributes*. DynamoDB uses primary keys to uniquely identify each item in a table and secondary indexes to provide more querying flexibility.

Tables, Items, and Attributes

The following are the basic DynamoDB components:

- **Tables** – Similar to other database systems, DynamoDB stores data in tables. A *table* is a collection of data. For example, see the example table called *People* that you could use to store personal contact information about friends, family, or anyone else of interest. You could also have a *Cars* table to store information about vehicles that people drive.
- **Items** – Each table contains zero or more items. An *item* is a group of attributes that is uniquely identifiable among all of the other items. In a *People* table, each item represents a person. For a *Cars* table, each item represents one vehicle. Items in DynamoDB are similar in many ways to rows, records, or tuples in other database systems. In DynamoDB, there is no limit to the number of items you can store in a table.
- **Attributes** – Each item is composed of one or more attributes. An *attribute* is a fundamental data element, something that does not need to be broken down any further. For example, an item in a *People* table contains attributes called *PersonID*, *LastName*, *FirstName*, and so on. For a *Department* table, an item might have attributes such as *DepartmentID*, *Name*, *Manager*, and so on. Attributes in DynamoDB are similar in many ways to fields or columns in other database systems.

People

Each item in the table has a unique identifier, or primary key, that distinguishes the item from all of the others in the table. In the *People* table, the primary key consists of one attribute (*PersonID*).

Other than the primary key, the *People* table is schemaless, which means that neither the attributes nor their data types need to be defined beforehand. Each item can have its own distinct attributes.

Most of the attributes are *scalar*, which means that they can have only one value. Strings and numbers are common examples of scalars.

Some of the items have a nested attribute (*Address*). DynamoDB supports nested attributes up to 32 levels deep.

Primary 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*.

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

The *People* table is an example of a table with a simple primary key (*PersonID*). You can access any item in the *People* table directly 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*.

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.

Secondary Indexes

You can create one or more secondary indexes on a table. A *secondary index* lets you query the data in the table using an alternate key, in addition to queries against the primary key. DynamoDB doesn't require that you use indexes, but they give your applications more flexibility when querying your data. After you create a secondary index on a table, you can read data from the index in much the same way as you do from the table.

DynamoDB supports two kinds of indexes:

- Global secondary index – An index with a partition key and sort key that can be different from those on the table.
- Local secondary index – An index that has the same partition key as the table, but a different sort key.

You can define up to 5 global secondary indexes and 5 local secondary indexes per table.

In the example *Music* table shown previously, you can query data items by *Artist* (partition key) or by *Artist* and *SongTitle* (partition key and sort key). What if you also wanted to query the data by *Genre* and *AlbumTitle*? To do this, you could create an index on *Genre* and *AlbumTitle*, and then query the index in much the same way as you'd query the *Music* table.

Global Tables

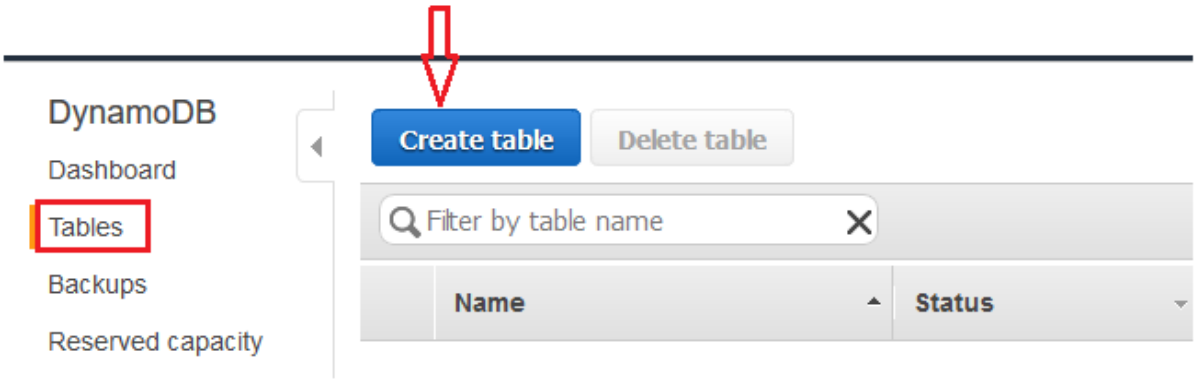
Amazon DynamoDB global tables provide a fully managed solution for deploying a multi-region, multi-master database, without having to build and maintain your own replication solution. When you create a global table, you specify the AWS regions where you want the table to be available. DynamoDB performs all of the necessary tasks to create identical tables in these regions, and propagate ongoing data changes to all of them.

DynamoDB global tables are ideal for massively scaled applications, with globally dispersed users. In such an environment, users expect very fast application performance. Global tables provide automatic multi-master replication to AWS regions world-wide, so you can deliver low-latency data access to your users no matter where they are located.

Creating a Global Table (Console)

Follow these steps to create a global table using the console. The following example creates a global table with replica tables in United States and Europe.

1. Open the DynamoDB console at https://console.aws.amazon.com/dynamodb/home.
2. In the navigation pane on the left side of the console, choose **Tables**.
3. Choose **Create Table**.



For **Table name**, type *Music*.

For **Primary key** type *Artist*. Choose **Add sort key**, and type *SongTitle*. (*Artist* and *SongTitle* should both be strings.)

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* Music

Primary key* Partition key

Artist String

☑ Add sort key

Songtitle String

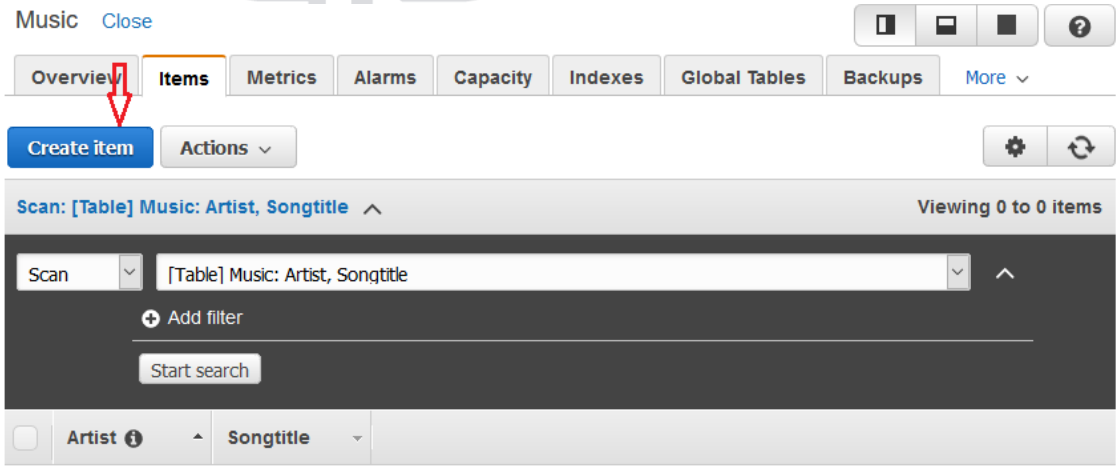
Table settings

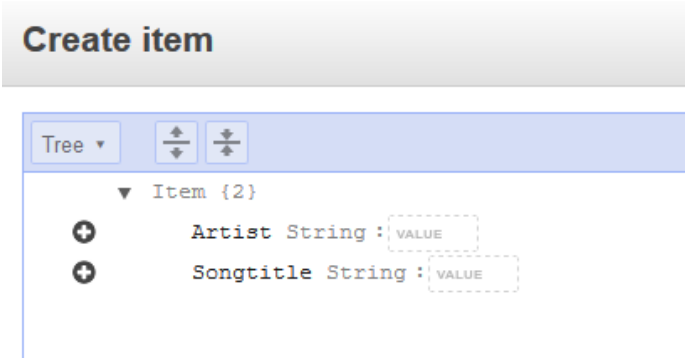
Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

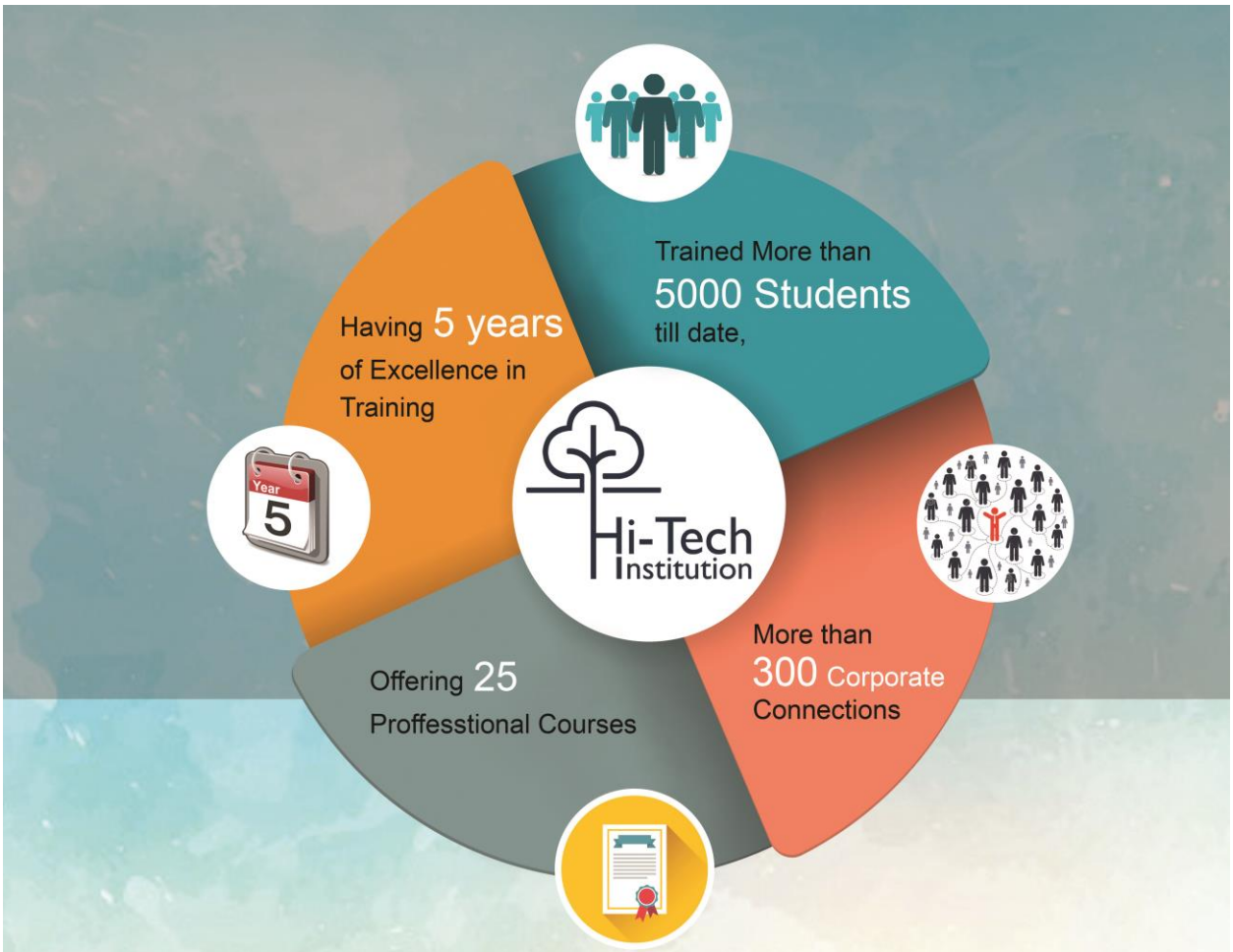
☑ Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".
- Encryption at Rest with DEFAULT encryption type NEW!

4. To Add an Item to Table







TOP RECRUITERS

Capgemini · Mindtree · Tech Mahindra · MISYS · EMC² · subex · FIS · L&T Technology Services · virtusa · TRIGENT · HUAWEI · KENNAMETAL · HCL · Atos · ALLEGIS GROUP · izmo · AWPL · Softtek · Infosys · BOSCH · hp · infinite · CGI · OPENTEXT · IGATE · mellow infosystems · GLOBAL EDGE · ARTECH · Fonezela virtual business assistant · CORPORATE LADDER · Simbus · BENISON TECHNOLOGIES · BRISTLECONE Your Supply Chain. Optimized. · THOUGHTFOCUS · and more...

Hi-Tech Institution

50% offer for School or College students

30% offer for IT Employees

Above offer applicable only technical courses. Terms and conditions apply

Email Us
operations@hitechins.in

Search Us
www.hitechins.in

CONTACT US

7092 90 91 92 / 82 20 21 7640

PONDICHERRY
No.32, 100 feet road,
Ellaipillaichavady,
Pondicherry – 605 005,
Nearby Rajiv Gandhi Hospital

TAMBARAM
No.24, Chithi Vinayagar Kovil street,
KamarajNagar, Tambaram Sanatorium,
Chennai – 600 047,
Nearby Sanatorium Railway Station

VELACHERRY
No: 21, Officer Colony,
100 feet road, VijayaNagar,
Velacherry – 600 042,
Nearby Sathya Home Appliances

Locations
Chennai & Pondicherry