DYNAMO DATABASE

Amazon **DynamoDB** is a **fully managed NoSQL database service** designed for **high availability, scalability, and performance**. It is a **key-value and document database** that delivers **single-digit millisecond latency** at any scale.

Key Features of DynamoDB

🚀 1. NoSQL, Key-Value & Document Store

- Stores **structured** and **semi-structured** data in **key-value** or **document** format.
- Supports **flexible schema** (unlike relational databases).

📈 2. Fully Managed & Scalable

- **On-demand scaling**: Automatically handles large workloads.
- **Supports millions of requests per second**.
- **Auto-sharding** distributes data across multiple partitions.

🔄 3. High Availability & Durability

- **Multi-AZ replication** (across 3 AWS availability zones).
- **Automatic backups & point-in-time recovery**.

⚡ 4. Fast Performance

- **Single-digit millisecond latency**.
- Supports **DAX (DynamoDB Accelerator)** for caching to improve read performance.

🔐 5. Security & Access Control

- **IAM-based authentication** (fine-grained access control).
- **Encryption at rest and in transit**.

📊 6. Querying & Indexing

- **Primary Key** (Partition Key + Optional Sort Key).
- **Secondary Indexes**:
    - **Global Secondary Index (GSI)**: Allows querying on non-primary attributes.
    - **Local Secondary Index (LSI)**: Enables sorting within partitions.

Use Cases of DynamoDB

- **Real-time applications**: Leaderboards, chat applications.
- **IoT & Sensor Data**: Stores rapidly incoming device data.
- **E-commerce**: Shopping carts, product catalogs.
- **Gaming**: Storing game state & session history.
- **Financial & Transactional Applications**.

How to Create a DynamoDB Table

✅ 1. Using AWS Console

1. Go to **AWS Management Console** → **DynamoDB**.
2. Click **Create Table**.
3. **Enter Table Name** (`Users`).
4. **Define Primary Key**:
    - Partition Key: `Usernames` (String)
    - (Optional) Sort Key: `Timestamp` (Number)
5. Select **Provisioned** or **On-Demand Capacity**.
6. Click **Create Table**.

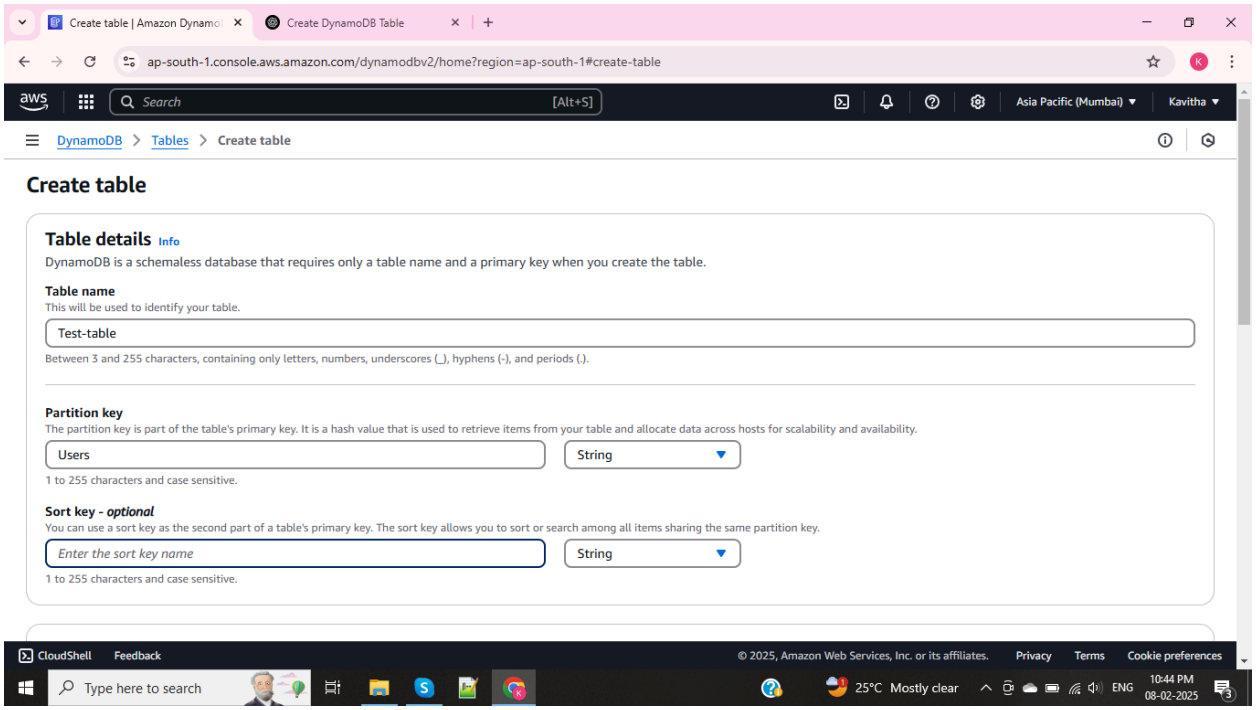
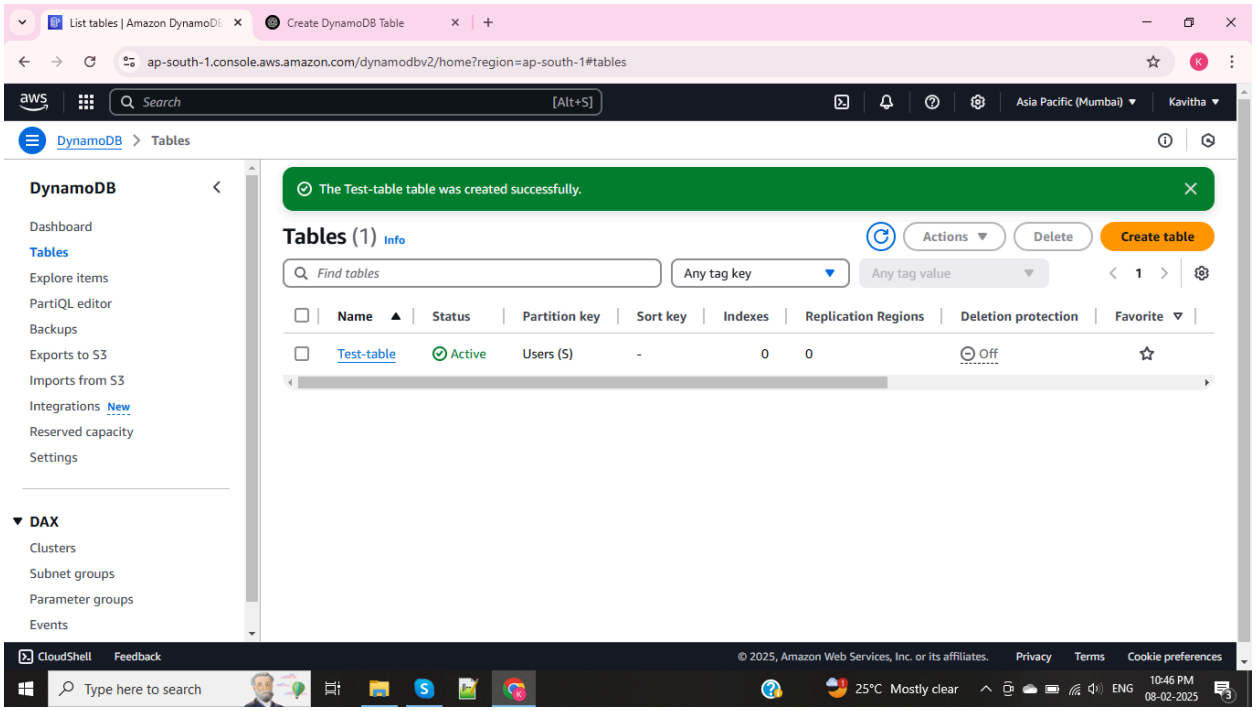
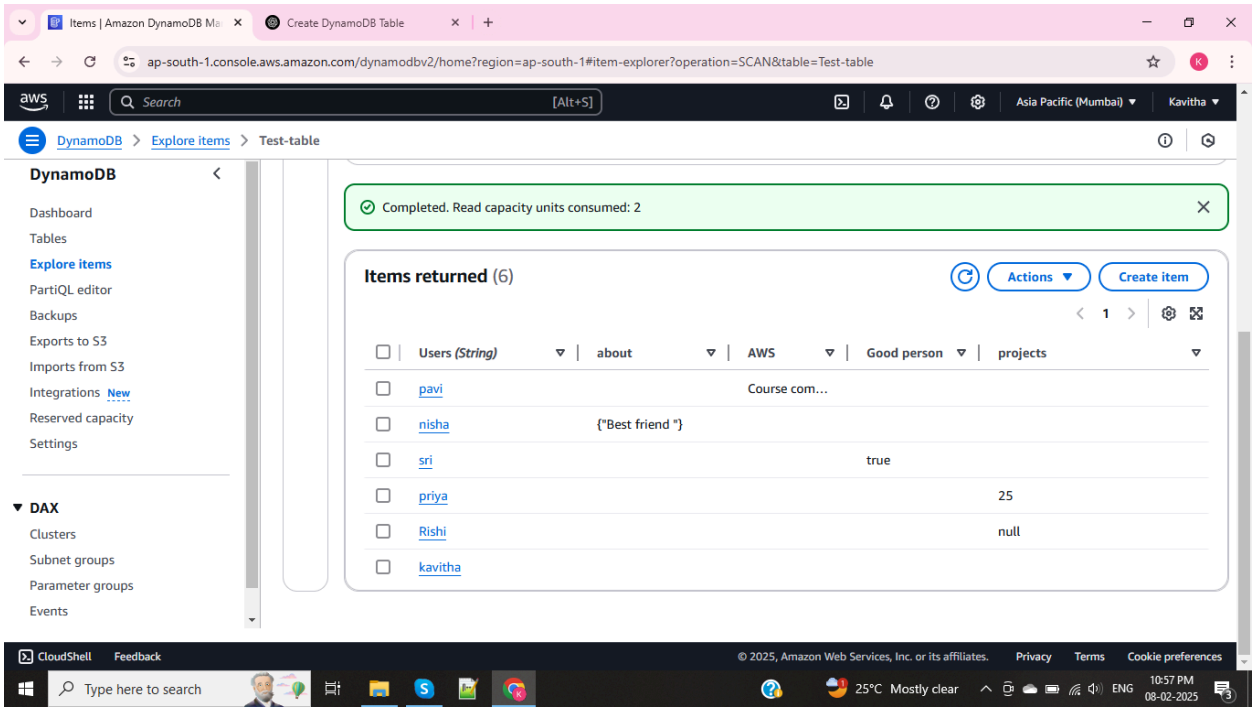


TABLE created successfully



Click table > **Create item** > using many attributes (**String, Boolean, Number, Null, String set**) explore a items in table.



DynamoDB vs RDS (Relational Database)

Feature	DynamoDB (NoSQL)	RDS (SQL)
Schema	Flexible	Fixed
Scalability	Auto-scaled	Manual scaling
Performance	Millisecond latency	Optimized for complex queries
Query Type	NOSQL queries	SQL queries
Use Case	Real-time apps, IoT, gaming	Traditional applications