🔹 DynamoDB – Complete Internship-Level Prep

1. What is DynamoDB?

- **Amazon DynamoDB** is a **fully managed, serverless, NoSQL database service** by AWS.
- It is designed for **fast, consistent performance** at any scale.
- Stores data in **key-value** and **document-oriented** models.

---

2. Related Concepts

NoSQL

- DynamoDB is **NoSQL**, meaning:
    - Data is **not stored in rows and columns** like relational databases.
    - It stores **items** (like rows) inside **tables**.
    - Each item is a collection of **attributes** (key-value pairs).
    - Flexible schema → You don't need to predefine all columns.

Serverless

- No need to manage servers (AWS handles infrastructure, scaling, backups).
- Pay only for what you use (**on-demand pricing**).
- Works well for applications with unpredictable traffic.

---

3. Core DynamoDB Structure

- **Table** → Collection of data (like in SQL).
- **Item** → A single record inside the table (like a row).
- **Attribute** → Key-value pair inside an item (like a column).

Example:
Table Name: **Users**

- **Partition Key (Primary Key)**: UserID (must be unique).

- Attributes: Name, Email, Age, Address.

---

4. Setup Steps (Basic Implementation)

If asked how to create a DynamoDB table:

1. Go to **AWS Management Console → DynamoDB → Create Table**.
2. **Give table name** → Example: Users.
3. **Give unique ID (Partition Key)** → Example: UserID (String).
   - Optionally, add **Sort Key** if you need composite keys.
4. Choose **Capacity Mode**:
   - On-demand or Provisioned.
5. Click **Create** → Table is ready.

---

5. Key Points (Interview-Ready)

- **Serverless** → No server management, auto-handled by AWS.
- **Automatic Scaling** → DynamoDB automatically scales read/write capacity as needed.
- **Zero Downtime** → Always available (designed for high availability).
- **On-Demand Pricing** → Pay per request, no upfront cost.

---

6. Read & Write Capacity Modes

DynamoDB offers **two modes**:

1. **Provisioned Mode** (fixed capacity):
   - You specify **Read Capacity Units (RCU)** and **Write Capacity Units (WCU)**.
   - Suitable for predictable workloads.
2. **On-Demand Mode** (serverless pay-per-use):
   - You don’t set RCU/WCU manually.

    - DynamoDB scales automatically.
    - Best for **unpredictable traffic**.

---

7. Point-in-Time Recovery (PITR)

- **Backup feature** in DynamoDB.
- Allows restoring your table data to any point within the **last 35 days**.
- Useful for accidental deletes or data corruption.

---

8. Example Use Cases (good for interview)

- User profiles (gaming, social media).
- IoT sensor data.
- Shopping carts (e-commerce).
- Session storage for web apps.

---

9. Example Query (Basic CRUD)

Using **AWS CLI / SDK**:

- **Insert (Put Item)**:

```
aws dynamodb put-item \
  --table-name Users \
  --item '{"UserID": {"S": "101"}, "Name": {"S": "Niranjan"}, "Email": {"S": "niranjan@example.com"}}'
```

- **Get Item**:

```
aws dynamodb get-item \
  --table-name Users \
  --key '{"UserID": {"S": "101"}}'
```

✅ DynamoDB Quick Recap (for Interview)

- **Type**: NoSQL, Key-Value & Document store.
- **Serverless**: AWS manages everything, pay-per-use.
- **Tables**: Made up of items (rows) and attributes (columns).
- **Keys**: Partition Key (unique ID), optional Sort Key.
- **Capacity Modes**: Provisioned (RCU/WCU) vs On-Demand.
- **Features**: Automatic scaling, zero downtime, PITR backups.
- **Pricing**: On-demand pay-per-request or provisioned.

👉 My suggestion: Learn DynamoDB **hands-on** by quickly creating a table and inserting/getting data using the AWS Console. They might ask you:

- "How will you create a table?"
- "What is the difference between provisioned and on-demand mode?"
- "What does Point-in-Time Recovery mean?