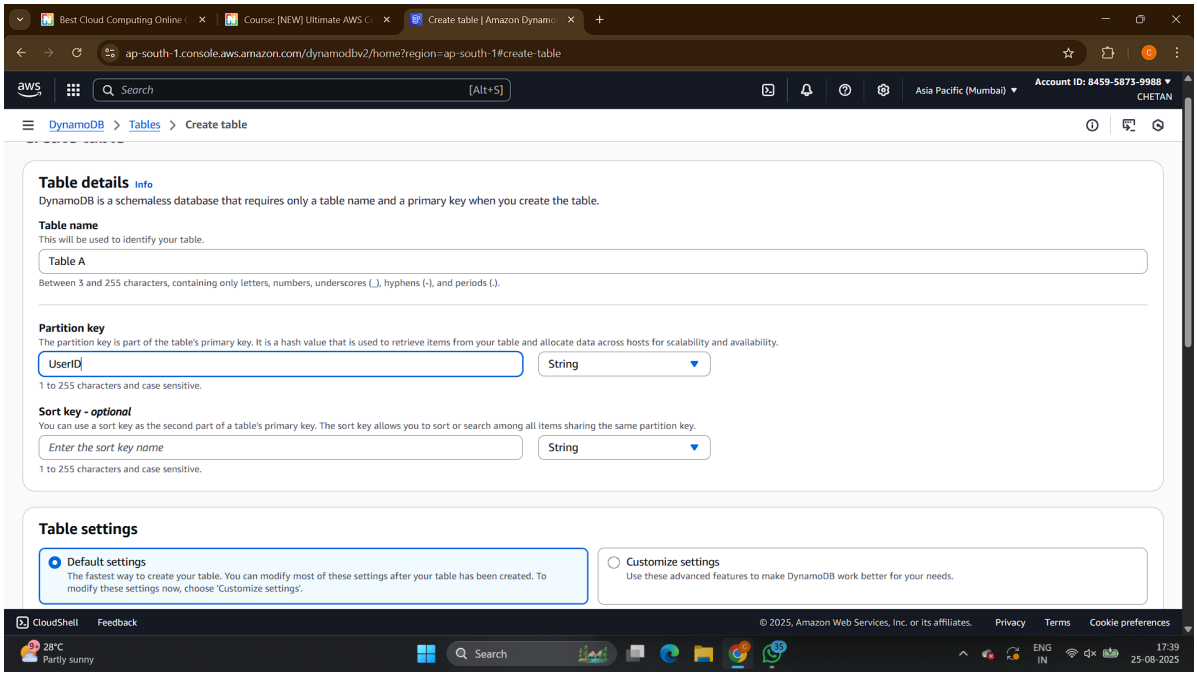
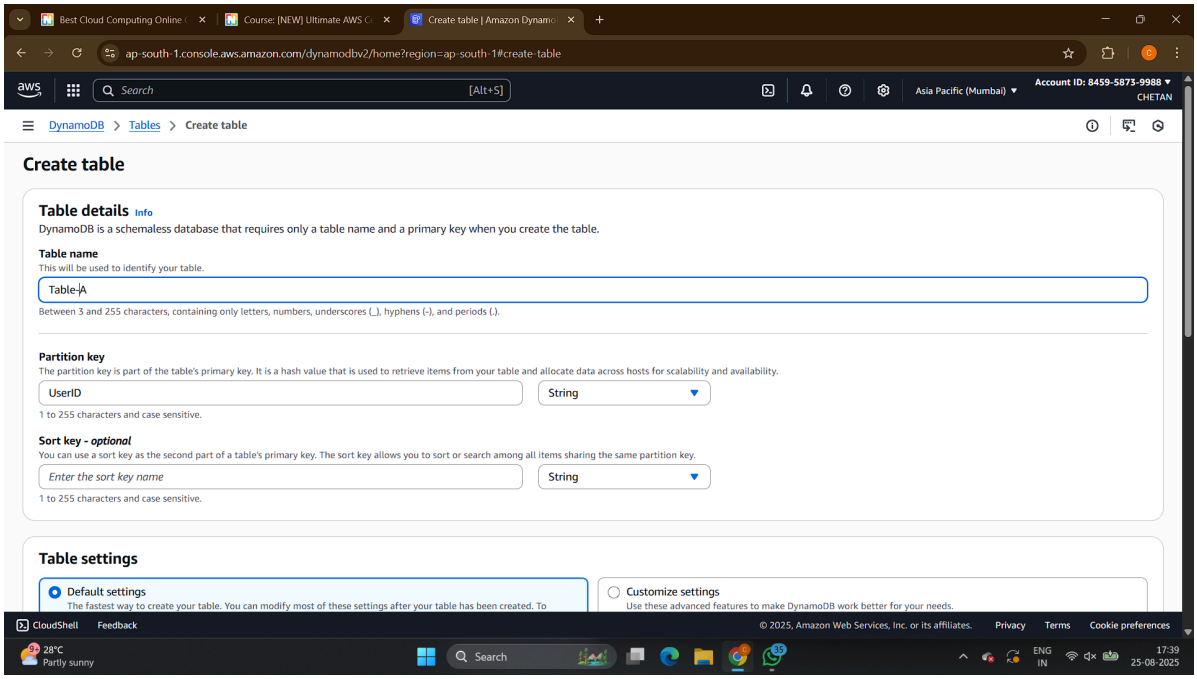
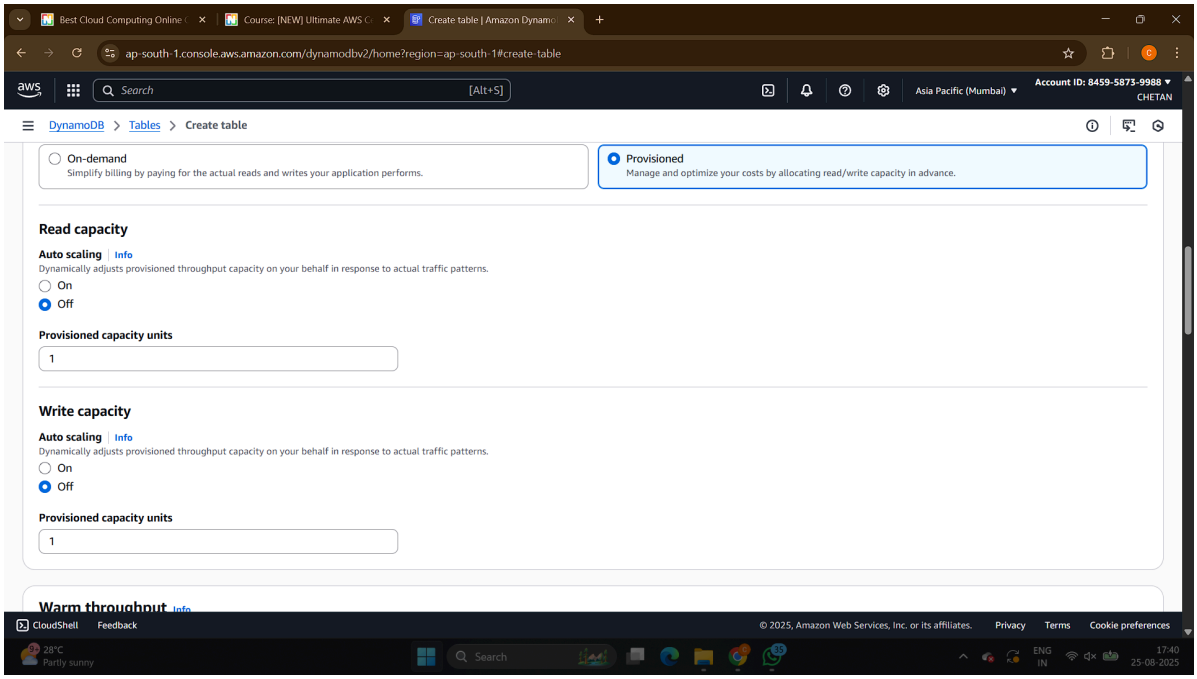
Cycle 08 AWS Homework

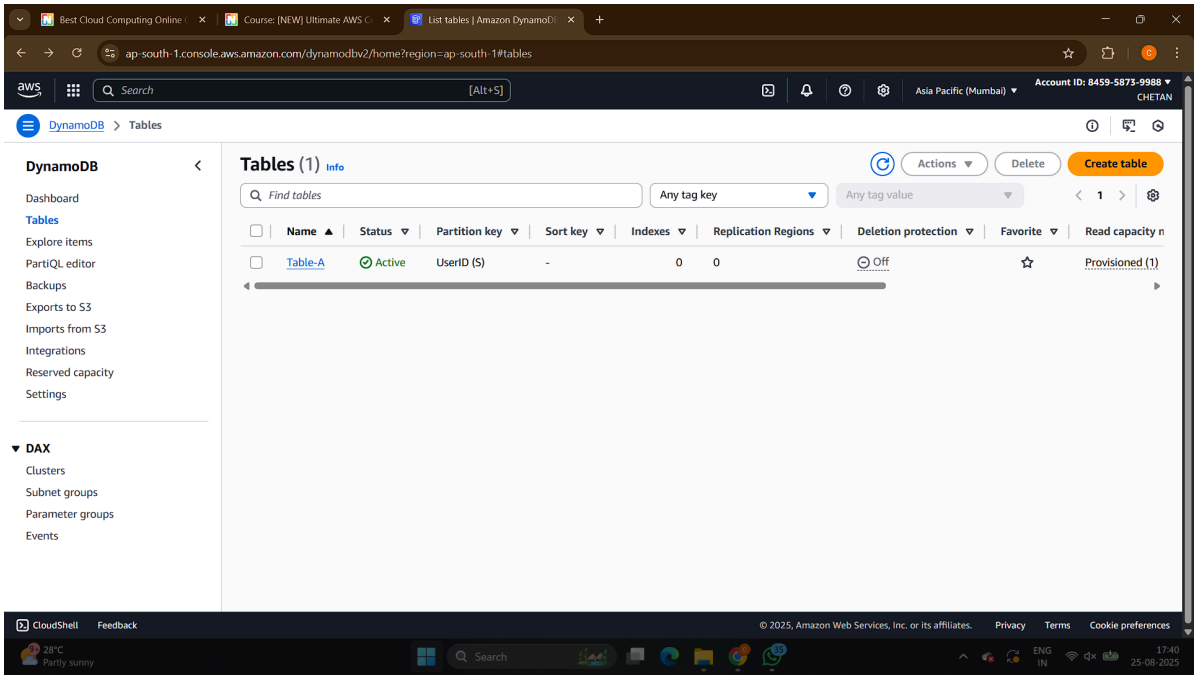
Task 1: Experiment with Partition and Sort Keys

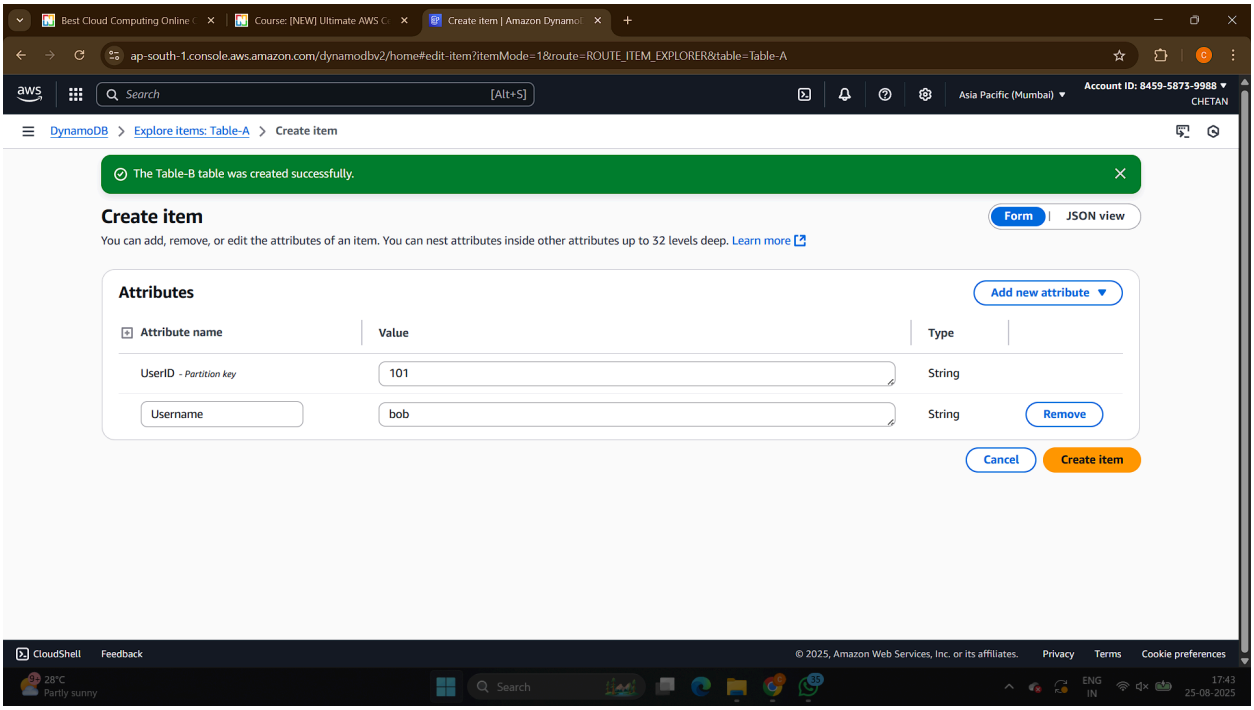
- Create Table A with a simple primary key:
  - Table: **Users**
  - Partition Key: `UserID` (e.g., string or number)

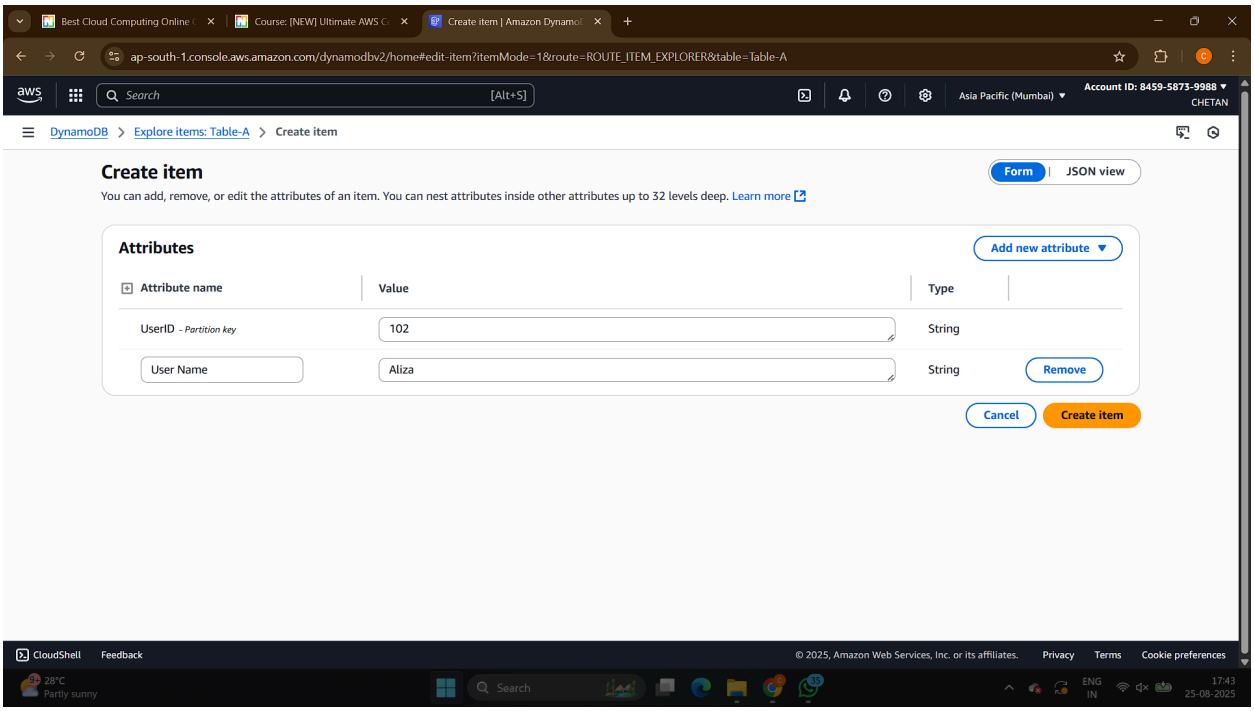


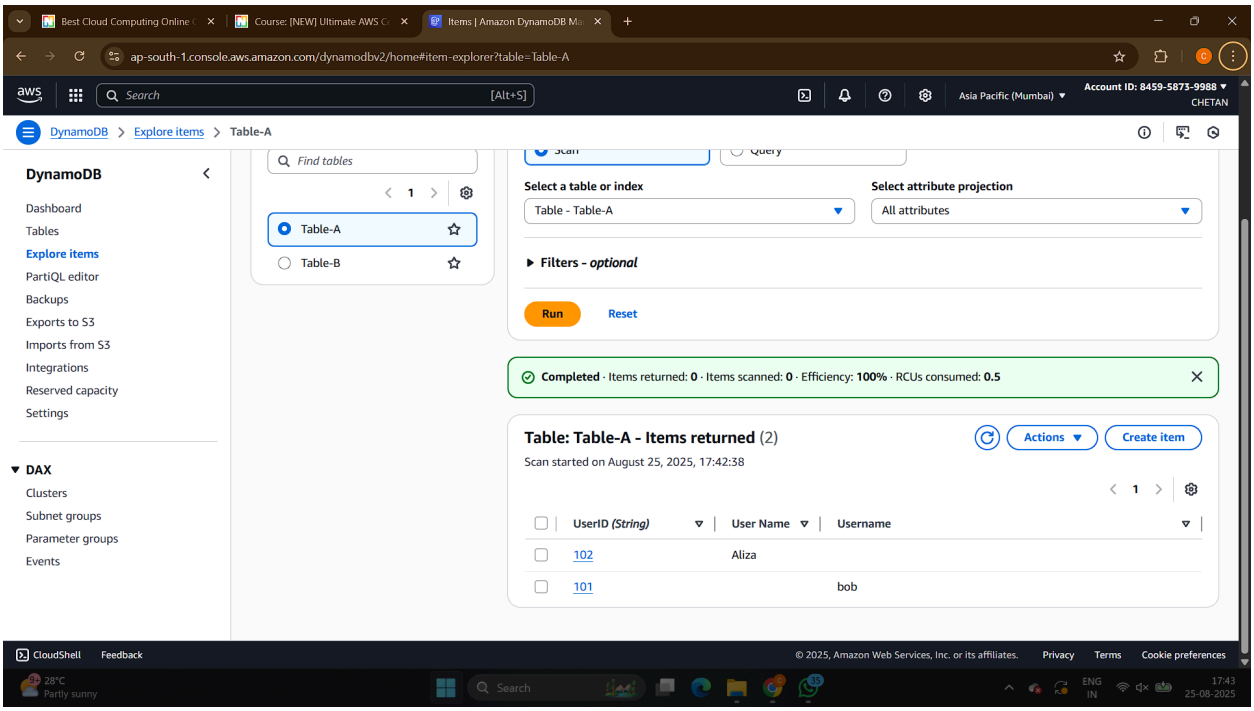

Best Cloud Computing Online
Course: [NEW] Ultimate AWS
Create table | Amazon Dynamo
ap-south-1.console.aws.amazon.com/dynamodbv2/home?region=ap-south-1#create-table
Search
[Alt+S]
Asia Pacific (Mumbai)
Account ID: 8459-5873-9988
CHETAN
DynamoDB > Tables > Create table
Create table
Table details Info
DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.
Table name
This will be used to identify your table.
Table-A
Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).
Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.
UserID
String
1 to 255 characters and case sensitive.
Sort key - optional
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.
Enter the sort key name
String
1 to 255 characters and case sensitive.
Table settings
Default settings
The fastest way to create your table. You can modify most of these settings after your table has been created. To
Customize settings
Use these advanced features to make DynamoDB work better for your needs.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Partly sunny
Search
ENG
IN
17:39
25-08-2025

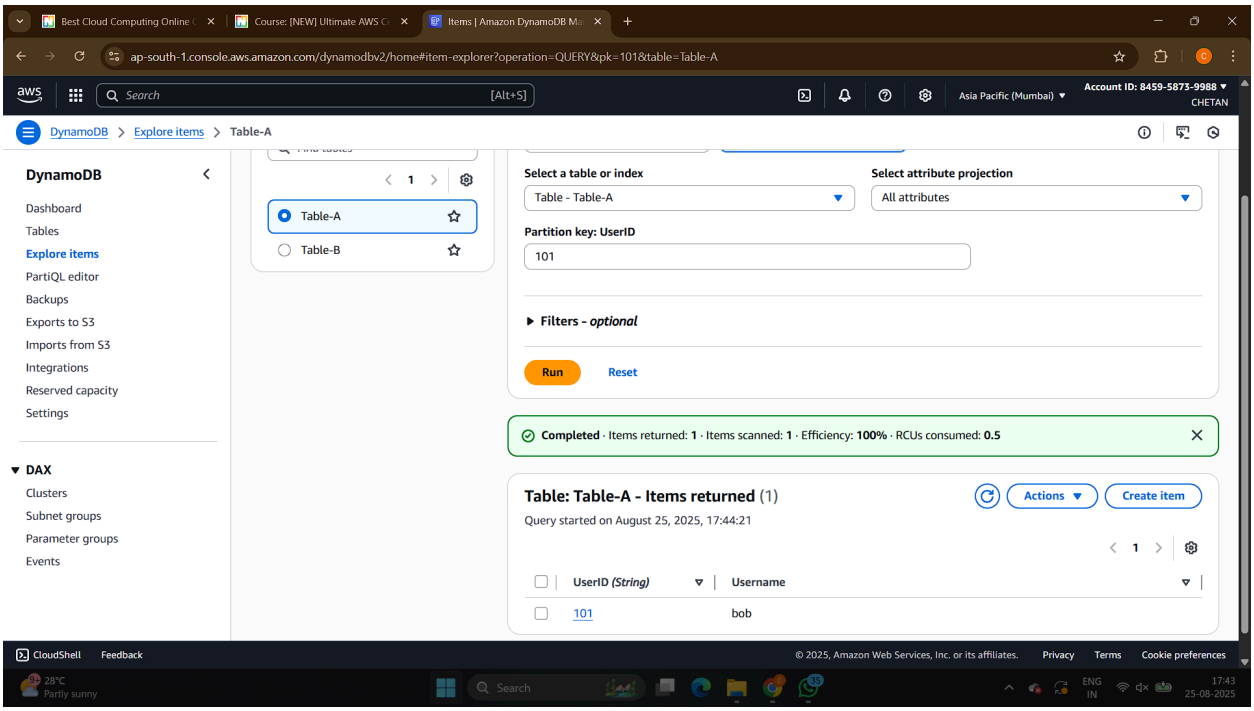

Best Cloud Computing Online
Course: [NEW] Ultimate AWS
Create table | Amazon Dynamo
ap-south-1.console.aws.amazon.com/dynamodbv2/home?region=ap-south-1#create-table
Search
[Alt+S]
Asia Pacific (Mumbai)
Account ID: 8459-5873-9988
CHETAN
DynamoDB > Tables > Create table
On-demand
Simplify billing by paying for the actual reads and writes your application performs.
Provisioned
Manage and optimize your costs by allocating read/write capacity in advance.
Read capacity
Auto scaling Info
Dynamically adjusts provisioned throughput capacity on your behalf in response to actual traffic patterns.
On
Off
Provisioned capacity units
1
Write capacity
Auto scaling Info
Dynamically adjusts provisioned throughput capacity on your behalf in response to actual traffic patterns.
On
Off
Provisioned capacity units
1
Warm throughput Info
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Partly sunny
Search
ENG
IN
17:40
25-08-2025


Best Cloud Computing Online
Course: [NEW] Ultimate AWS
List tables | Amazon DynamoDB
ap-south-1.console.aws.amazon.com/dynamodbv2/home?region=ap-south-1#tables
Search
[Alt+S]
Asia Pacific (Mumbai)
Account ID: 8459-5873-9988
CHETAN
DynamoDB > Tables
DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
Tables (1) Info
Actions
Delete
Create table
Find tables
Any tag key
Any tag value
Name
Status
Partition key
Sort key
Indexes
Replication Regions
Deletion protection
Favorite
Read capacity n
Table-A
Active
UserID (S)
-
0
0
Off
Provisioned (1)
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

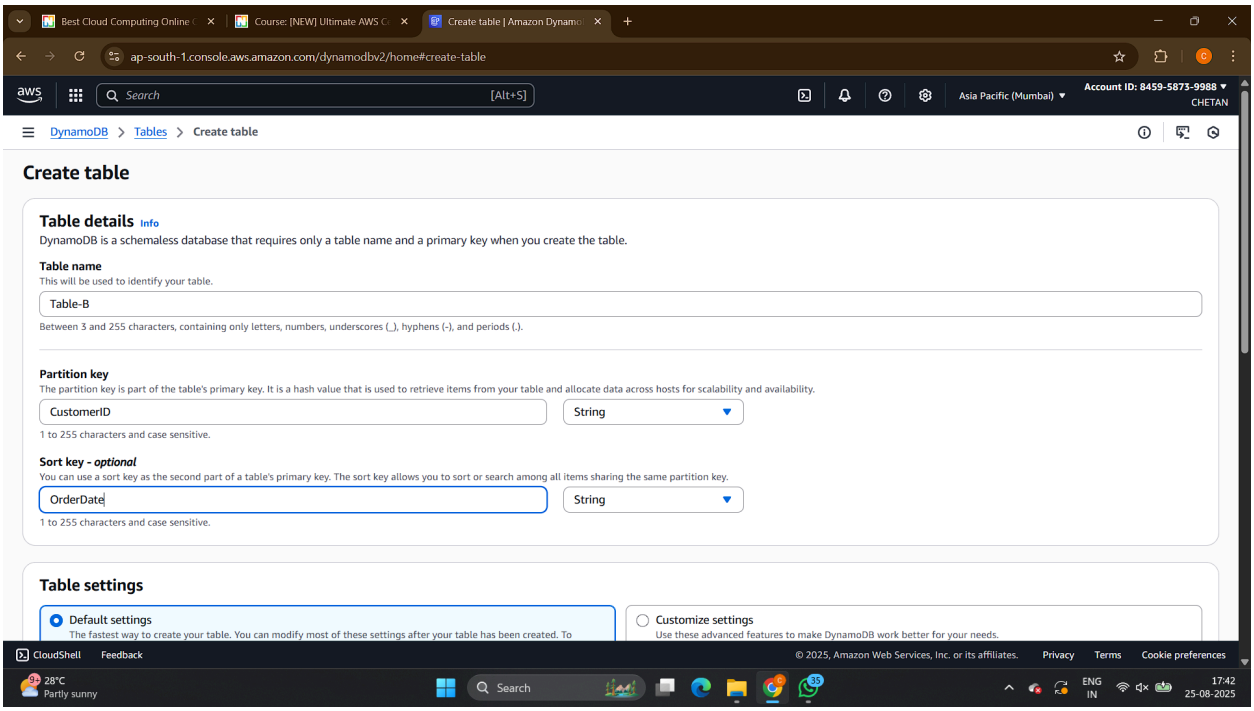

Best Cloud Computing Online
Course: [NEW] Ultimate AWS
Create item | Amazon DynamoDB
ap-south-1.console.aws.amazon.com/dynamodbv2/home#edit-item?itemMode=1&route=ROUTE_ITEM_EXPLORER&table=Table-A
Search
[Alt+S]
Asia Pacific (Mumbai)
Account ID: 8459-5873-9988
CHETAN
DynamoDB > Explore items: Table-A > Create item
The Table-B table was created successfully.
Create item
Form
JSON view
You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. Learn more
Attributes
Add new attribute
Attribute name
Value
Type
UserID - Partition key
101
String
Username
bob
String
Remove
Cancel
Create item
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

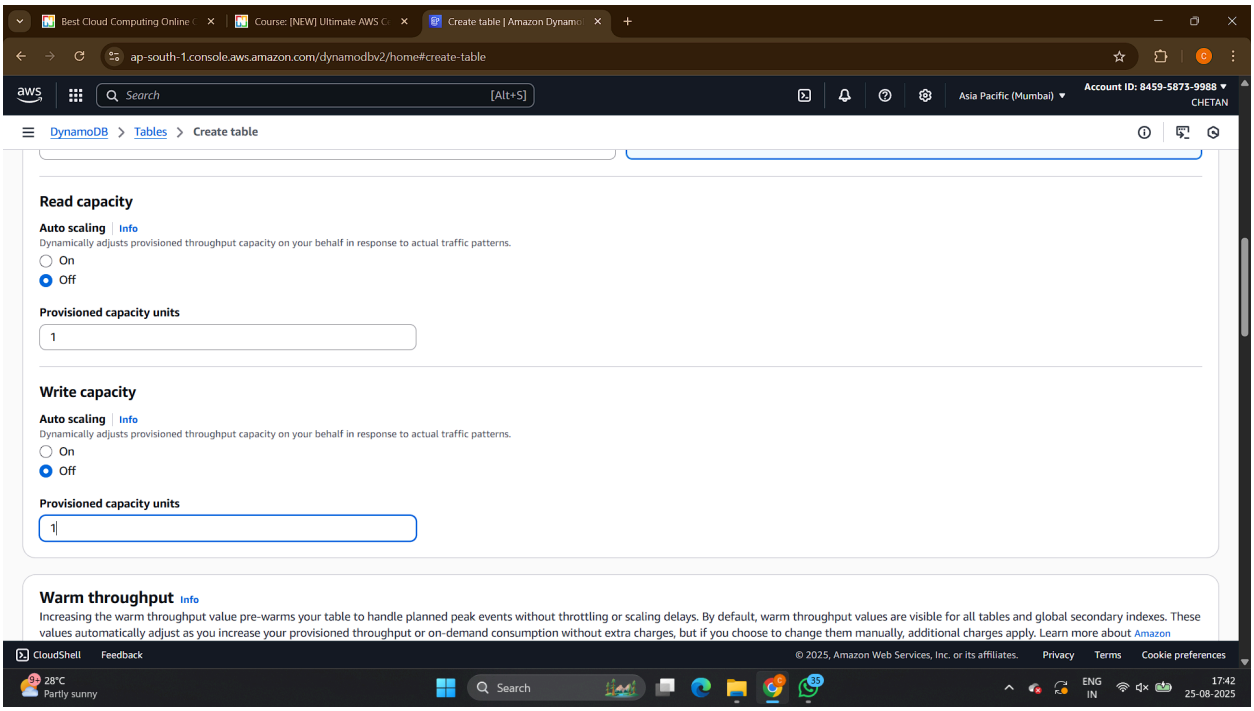

Create item | Amazon DynamoDB
ap-south-1.console.aws.amazon.com/dynamodbv2/home#edit-item?itemMode=1&route=ROUTE_ITEM_EXPLORER&table=Table-A
Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 8459-5873-9988
CHETAN
DynamoDB > Explore items: Table-A > Create item
Create item
Form | JSON view
You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. Learn more
Attributes
Add new attribute
Attribute name
Value
Type
UserID - Partition key
102
String
User Name
Aliza
String
Remove
Cancel
Create item
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

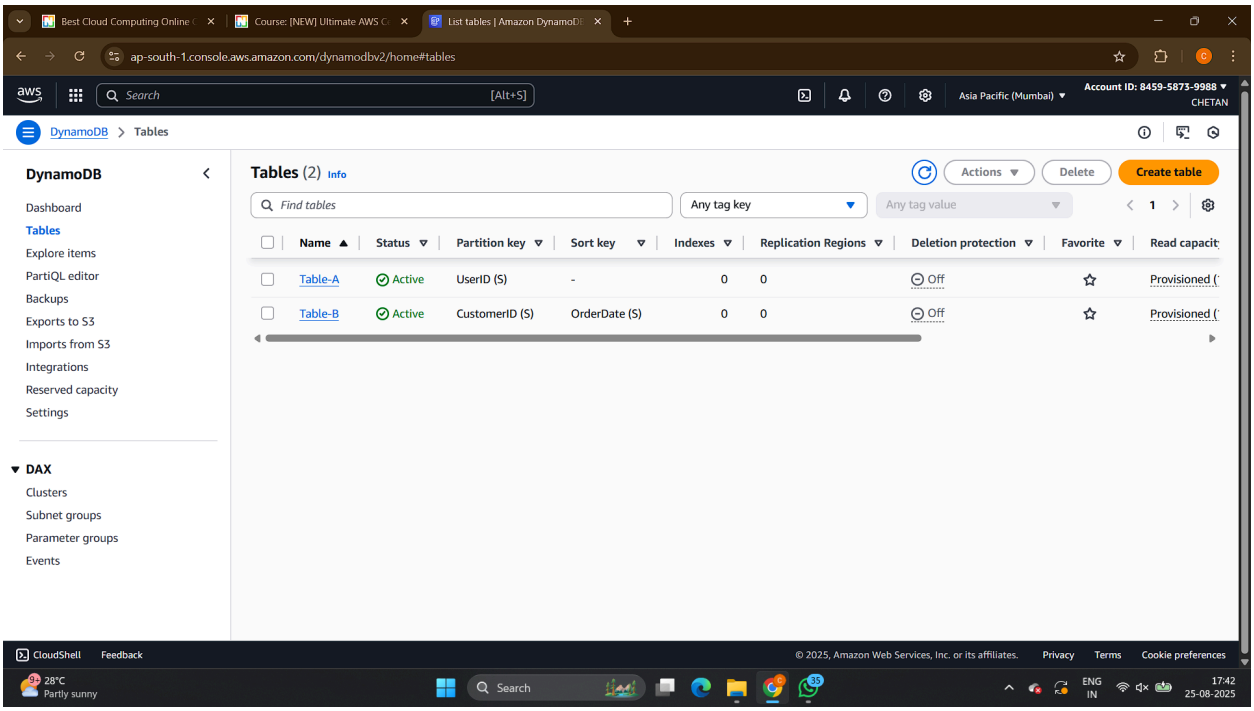

Items | Amazon DynamoDB
ap-south-1.console.aws.amazon.com/dynamodbv2/home#item-explorer?table=Table-A
Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 8459-5873-9988
CHETAN
DynamoDB > Explore items > Table-A
DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
Find tables
Table-A
Table-B
Select a table or index
Table - Table-A
Select attribute projection
All attributes
Filters - optional
Run
Reset
Completed · Items returned: 0 · Items scanned: 0 · Efficiency: 100% · RCUs consumed: 0.5
Table: Table-A - Items returned (2)
Scan started on August 25, 2025, 17:42:38
Actions
Create item
UserID (String)
User Name
Username
102
Aliza
101
bob
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

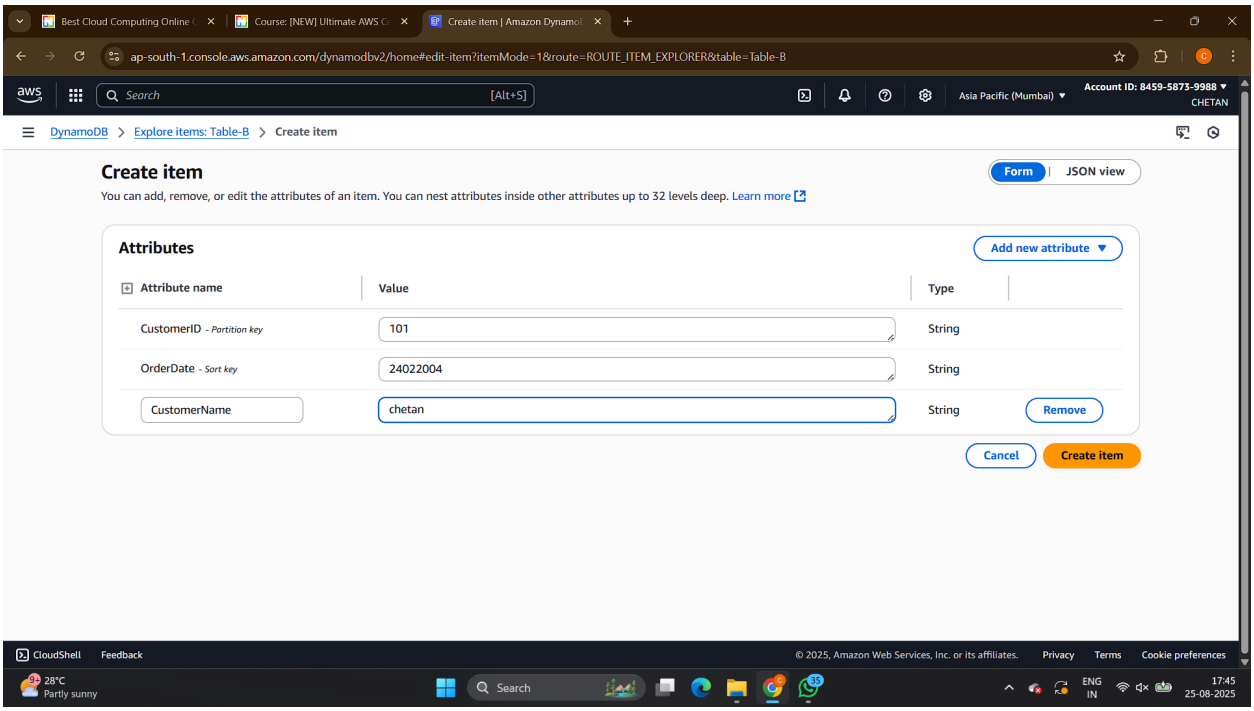


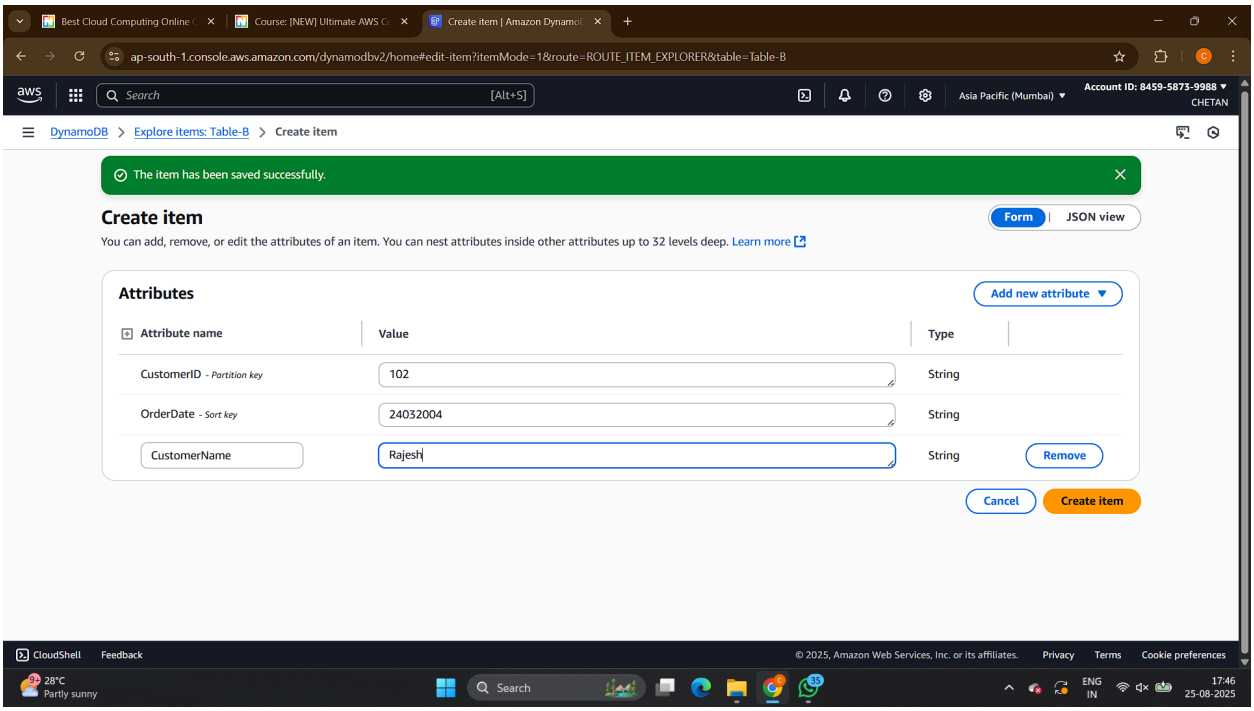
- Create Table B with a composite primary key:
  - Table: **Orders**
  - Partition Key: `CustomerID`
  - Sort Key: `OrderDate` (allows for sorting and range querying)
- Load sample data into both tables.

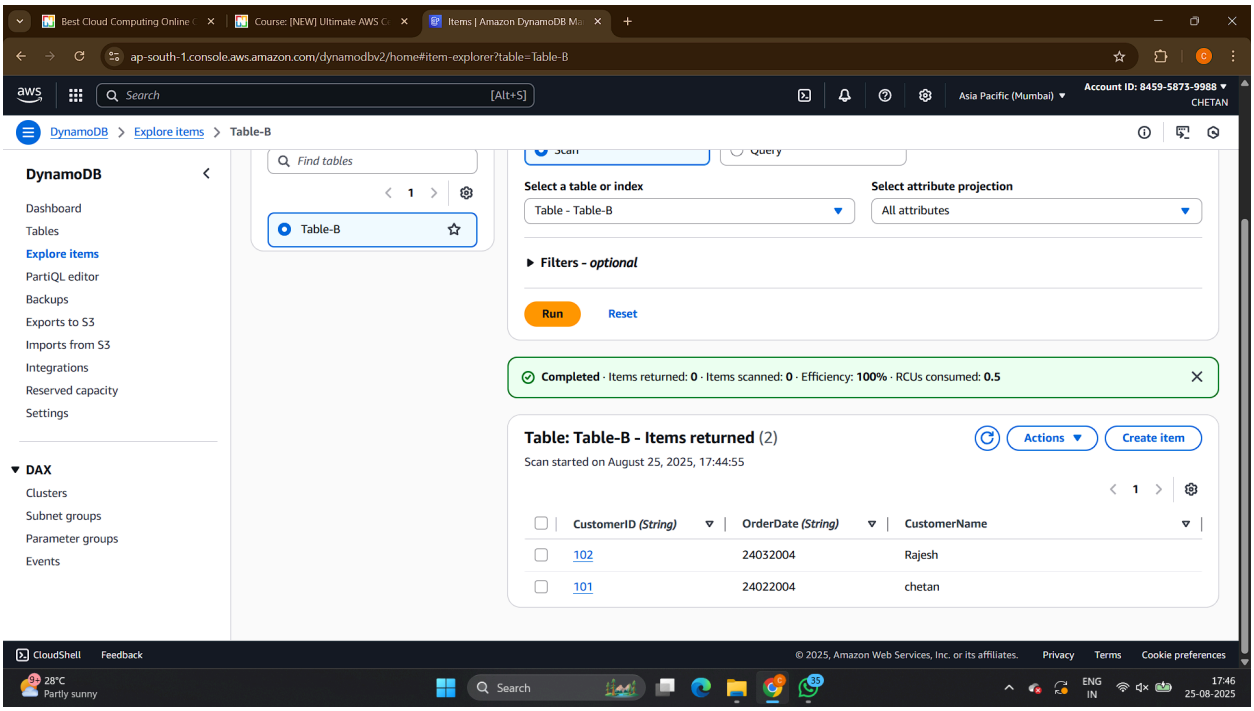

Best Cloud Computing Online
Course: [NEW] Ultimate AWS
Create table | Amazon Dynamo
ap-south-1.console.aws.amazon.com/dynamodbv2/home#create-table
Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 8459-5873-9988
CHETAN
DynamoDB > Tables > Create table
Create table
Table details Info
DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.
Table name
This will be used to identify your table.
Table-B
Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).
Partition key
The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.
CustomerID
String
1 to 255 characters and case sensitive.
Sort key - optional
You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.
OrderDate
String
1 to 255 characters and case sensitive.
Table settings
Default settings
The fastest way to create your table. You can modify most of these settings after your table has been created. To
Customize settings
Use these advanced features to make DynamoDB work better for your needs.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Partly sunny
Search
ENG
IN
17:42
25-08-2025

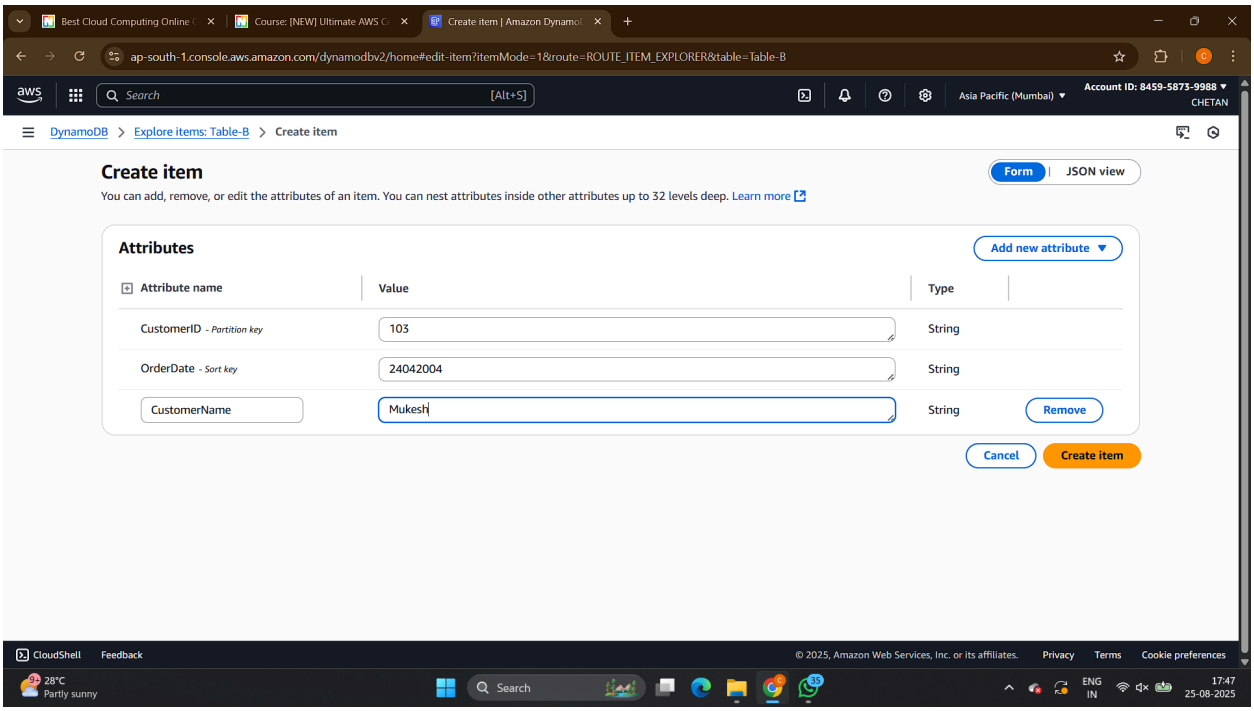

Best Cloud Computing Online
Course: [NEW] Ultimate AWS
Create table | Amazon Dynamo
ap-south-1.console.aws.amazon.com/dynamodbv2/home#create-table
Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 8459-5873-9988
CHETAN
DynamoDB > Tables > Create table
Read capacity
Auto scaling Info
Dynamically adjusts provisioned throughput capacity on your behalf in response to actual traffic patterns.
On
Off
Provisioned capacity units
1
Write capacity
Auto scaling Info
Dynamically adjusts provisioned throughput capacity on your behalf in response to actual traffic patterns.
On
Off
Provisioned capacity units
1
Warm throughput Info
Increasing the warm throughput value pre-warms your table to handle planned peak events without throttling or scaling delays. By default, warm throughput values are visible for all tables and global secondary indexes. These values automatically adjust as you increase your provisioned throughput or on-demand consumption without extra charges, but if you choose to change them manually, additional charges apply. Learn more about Amazon
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Partly sunny
Search
ENG
IN
17:42
25-08-2025

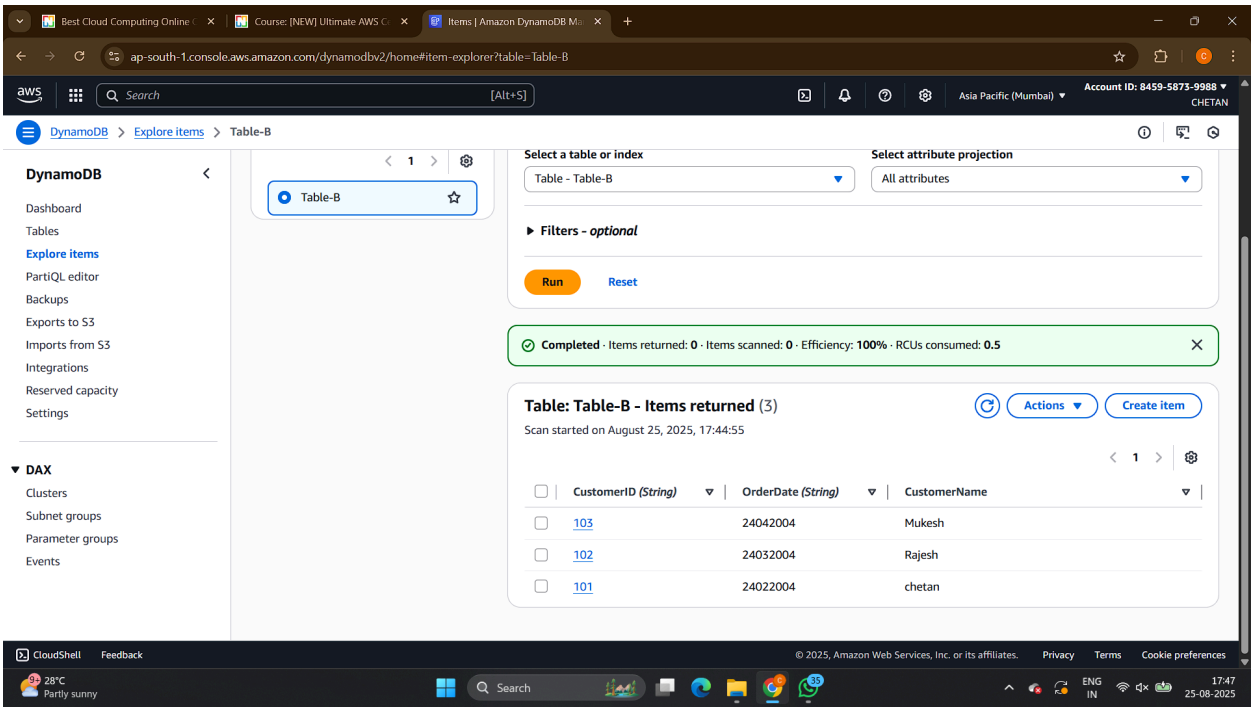

Best Cloud Computing Online
Course: [NEW] Ultimate AWS
List tables | Amazon DynamoDB
ap-south-1.console.aws.amazon.com/dynamodbv2/home#tables
Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 8459-5873-9988
CHETAN
DynamoDB > Tables
DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
Tables (2) Info
Actions
Delete
Create table
Find tables
Any tag key
Any tag value
Name
Status
Partition key
Sort key
Indexes
Replication Regions
Deletion protection
Favorite
Read capacity
Table-A
Active
UserID (S)
-
0
0
Off
Provisioned
Table-B
Active
CustomerID (S)
OrderDate (S)
0
0
Off
Provisioned
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Partly sunny
Search
ENG
IN
17:42
25-08-2025

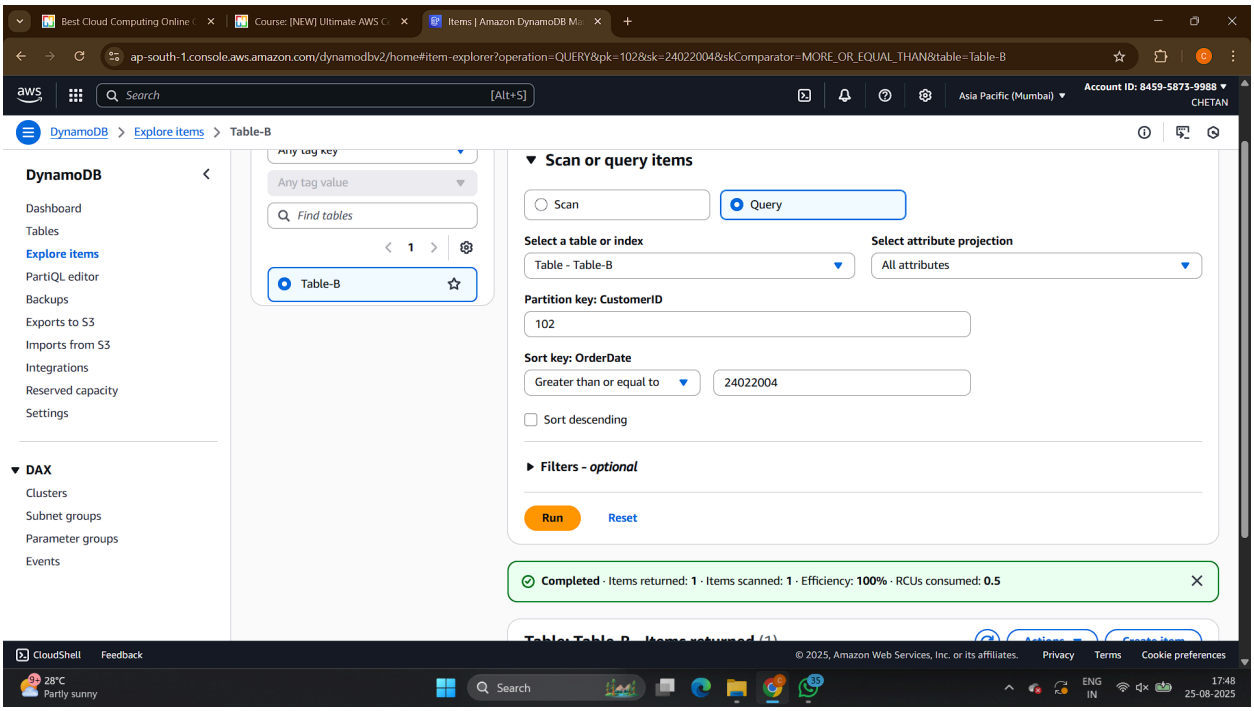

Best Cloud Computing Online
Course: [NEW] Ultimate AWS
Create item | Amazon DynamoDB
ap-south-1.console.aws.amazon.com/dynamodbv2/home#edit-item?itemMode=1&route=ROUTE_ITEM_EXPLORER&table=Table-B
Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 8459-5873-9988
CHETAN
DynamoDB > Explore items: Table-B > Create item
Create item
Form
JSON view
You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. Learn more
Attributes
Add new attribute
Attribute name
Value
Type
CustomerID - Partition key
101
String
OrderDate - Sort key
24022004
String
CustomerName
chetan
String
Remove
Cancel
Create item
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Partly sunny
Search
ENG
IN
17:45
25-08-2025

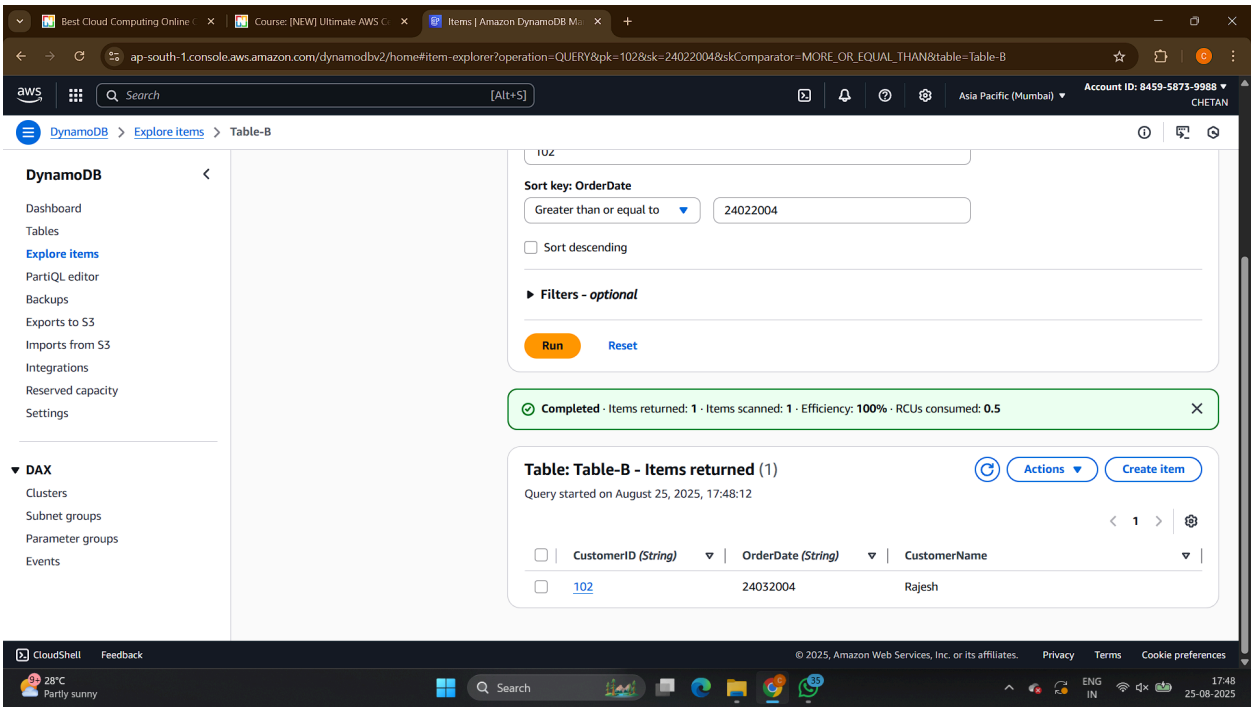

Create item | Amazon DynamoDB
ap-south-1.console.aws.amazon.com/dynamodbv2/home#edit-item?itemMode=1&route=ROUTE_ITEM_EXPLORER&table=Table-B
Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 8459-5873-9988
CHETAN
DynamoDB > Explore items: Table-B > Create item
The item has been saved successfully.
Create item
Form | JSON view
You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. Learn more
Attributes
Add new attribute
Attribute name
Value
Type
CustomerID - Partition key
102
String
OrderDate - Sort key
24032004
String
CustomerName
Rajesh
String
Remove
Cancel
Create item
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Partly sunny
Search
ENG
IN
17:46
25-08-2025


Items | Amazon DynamoDB
ap-south-1.console.aws.amazon.com/dynamodbv2/home#item-explorer?table=Table-B
Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 8459-5873-9988
CHETAN
DynamoDB > Explore items > Table-B
DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
Find tables
Table-B
Select a table or index
Table - Table-B
Select attribute projection
All attributes
Filters - optional
Run
Reset
Completed · Items returned: 0 · Items scanned: 0 · Efficiency: 100% · RCUs consumed: 0.5
Table: Table-B - Items returned (2)
Scan started on August 25, 2025, 17:44:55
Actions
Create item
CustomerID (String)
OrderDate (String)
CustomerName
102
24032004
Rajesh
101
24022004
chetan
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Partly sunny
Search
ENG
IN
17:46
25-08-2025


Create item | Amazon DynamoDB
ap-south-1.console.aws.amazon.com/dynamodbv2/home#edit-item?itemMode=1&route=ROUTE_ITEM_EXPLORER&table=Table-B
Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 8459-5873-9988
CHETAN
DynamoDB > Explore items: Table-B > Create item
Create item
Form | JSON view
You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. Learn more
Attributes
Add new attribute
Attribute name
Value
Type
CustomerID - Partition key
103
String
OrderDate - Sort key
24042004
String
CustomerName
Mukesh
String
Remove
Cancel
Create item
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Partly sunny
Search
ENG
IN
17:47
25-08-2025


Items | Amazon DynamoDB
ap-south-1.console.aws.amazon.com/dynamodbv2/home#item-explorer?table=Table-B
Search [Alt+S]
Asia Pacific (Mumbai)
Account ID: 8459-5873-9988
CHETAN
DynamoDB > Explore items > Table-B
DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
Table-B
Select a table or index
Table - Table-B
Select attribute projection
All attributes
Filters - optional
Run
Reset
Completed · Items returned: 0 · Items scanned: 0 · Efficiency: 100% · RCUs consumed: 0.5
Table: Table-B - Items returned (3)
Scan started on August 25, 2025, 17:44:55
Actions
Create item
CustomerID (String)
OrderDate (String)
CustomerName
103
24042004
Mukesh
102
24032004
Rajesh
101
24022004
chetan
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Partly sunny
Search
ENG
IN
17:47
25-08-2025





- Use AWS CLI for:
  - `GetItem` on **Users** table (retrieves single item by `UserID`).

    - `Query` on **Orders** table (retrieves all orders for a `CustomerID` and can filter by date range).
- Understanding:
    - Simple key allows unique retrieval by partition key only.
    - Composite key offers more querying flexibility by partition + sort key.

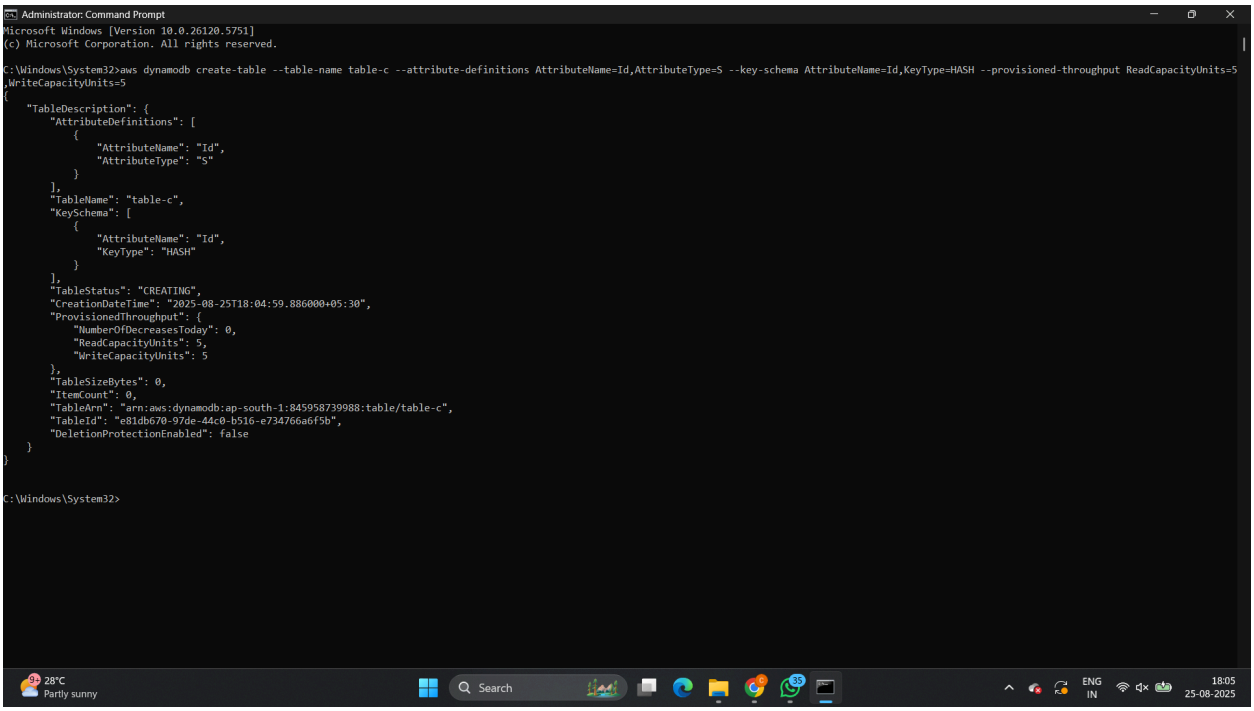
Example AWS CLI for creating tables with provisioned throughput:

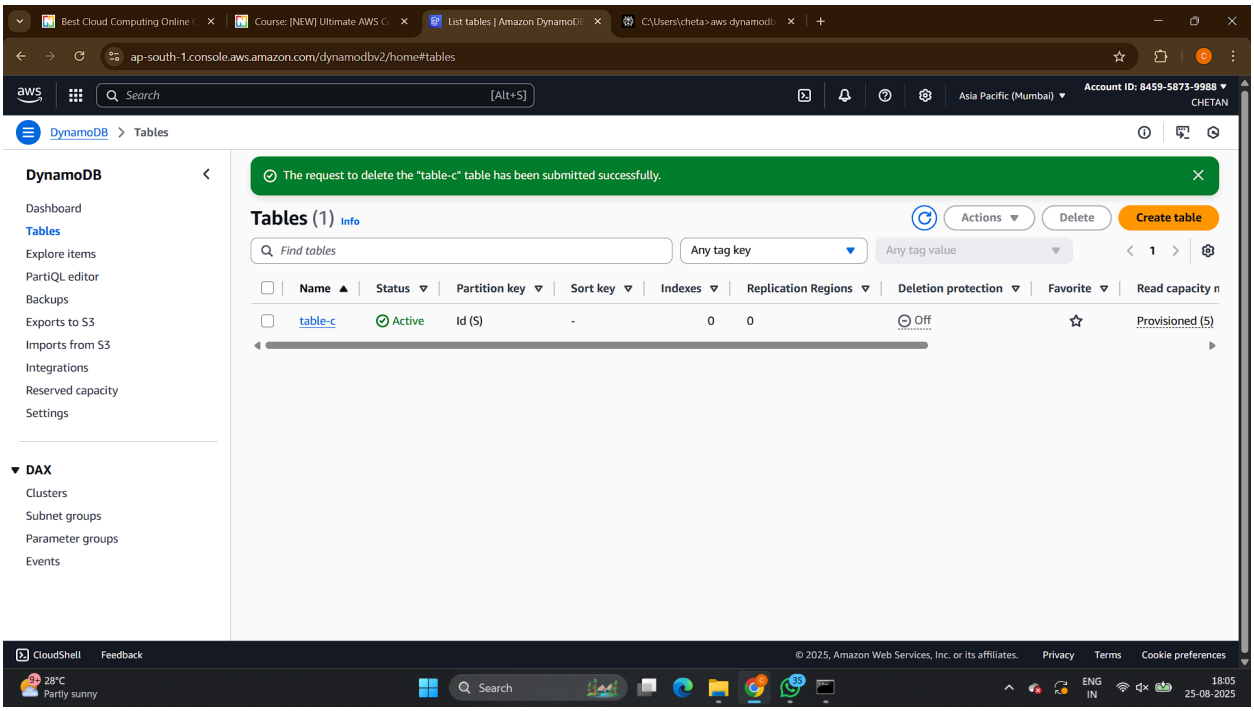
```
bashaws dynamodb create-table \
 --table-name Users \
 --attribute-definitions AttributeName=UserID,AttributeType=S \
 --key-schema AttributeName=UserID,KeyType=HASH \
 --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5
```

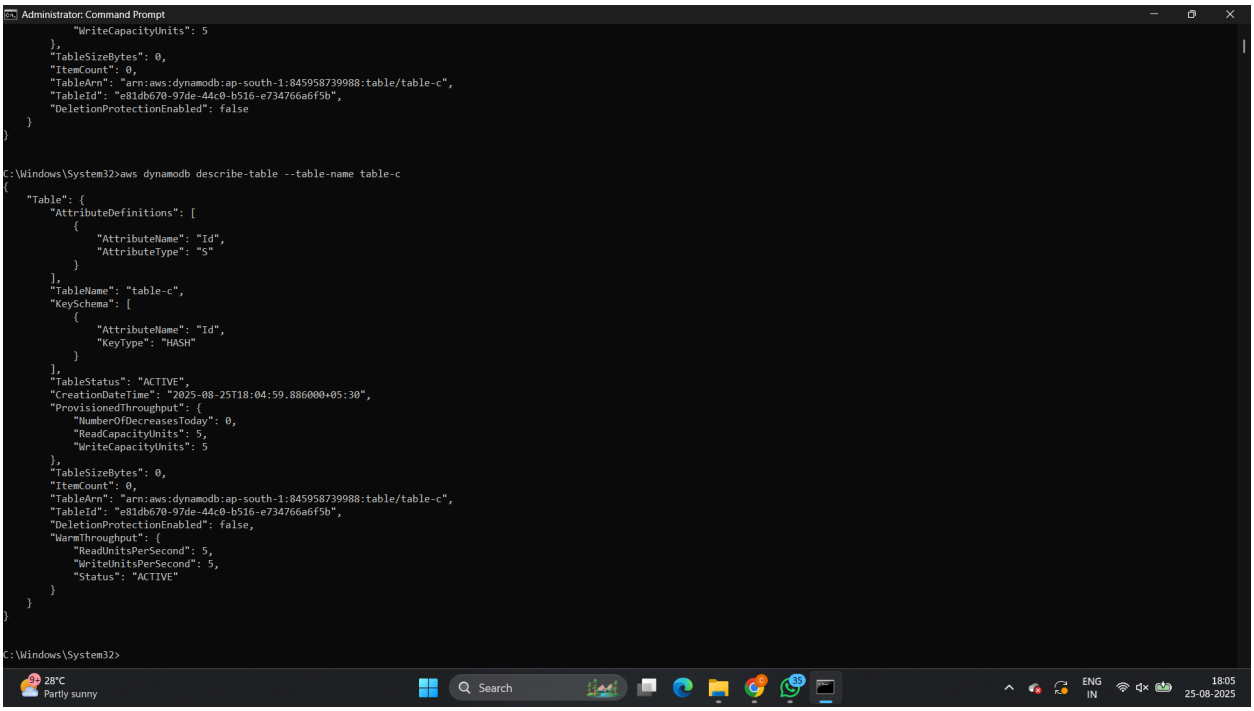
```
aws dynamodb create-table \
 --table-name Orders \
 --attribute-definitions AttributeName=CustomerID,AttributeType=S AttributeName=OrderDate,AttributeType=S \
 --key-schema AttributeName=CustomerID,KeyType=HASH AttributeName=OrderDate,KeyType=RANGE \
 --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5
```

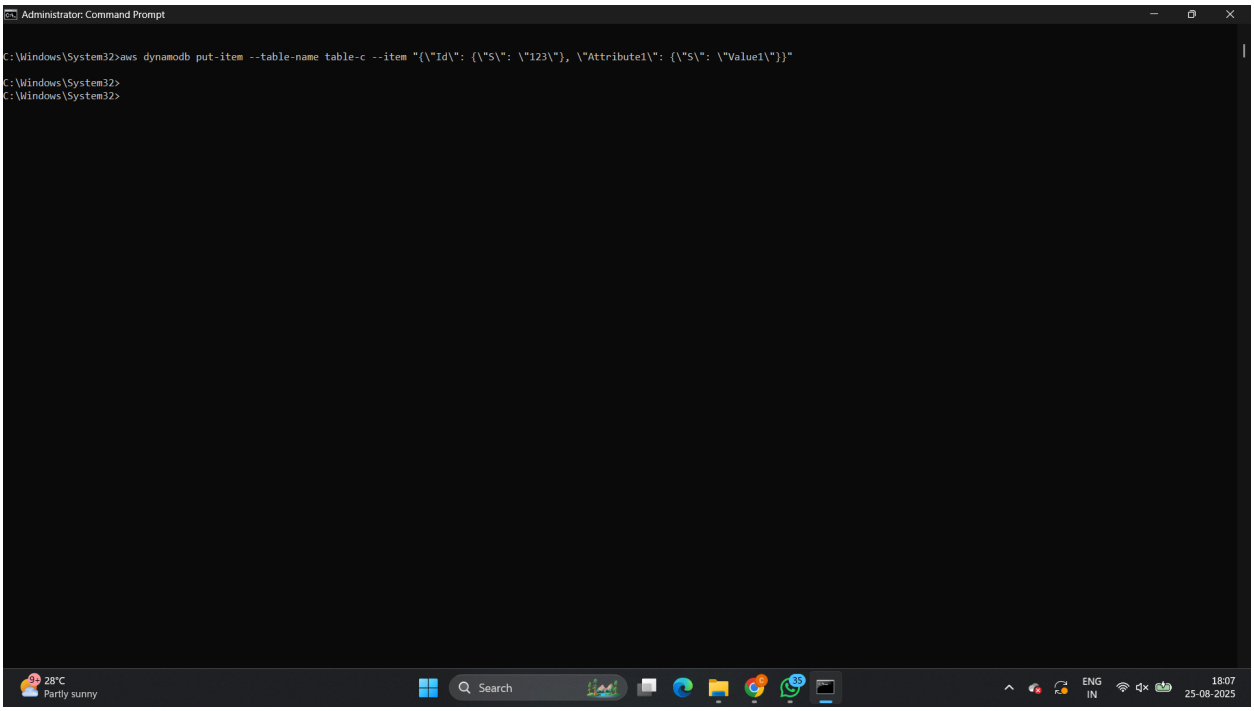
Task 2: Compare Consistency Models

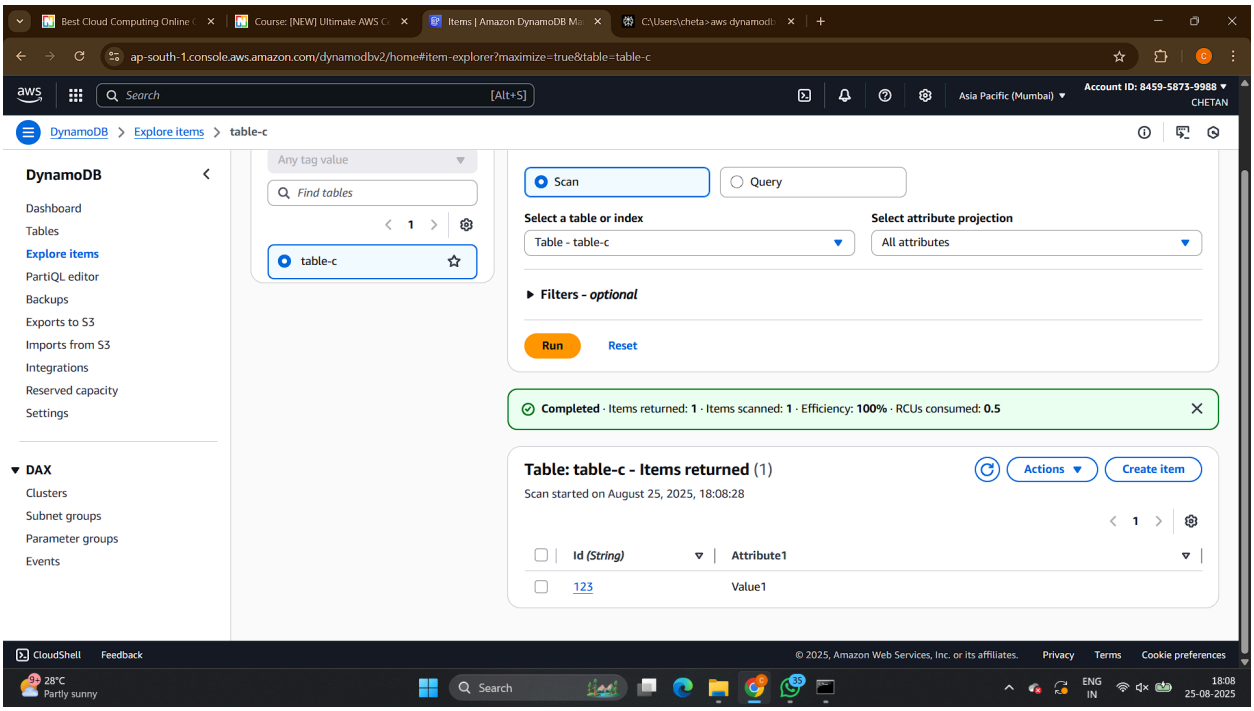
- Write an item to a DynamoDB table using AWS CLI.
- Read the item immediately:
    - Eventually consistent read: `-consistent-read false` (default)
    - Strongly consistent read: `-consistent-read true`
- Result:
    - Eventually consistent may show stale data if read too soon after write.
    - Strongly consistent read guarantees the latest committed data.
- Use cases for eventual consistency favor lower cost and latency, whereas strong consistency guarantees fresh data at a higher cost.

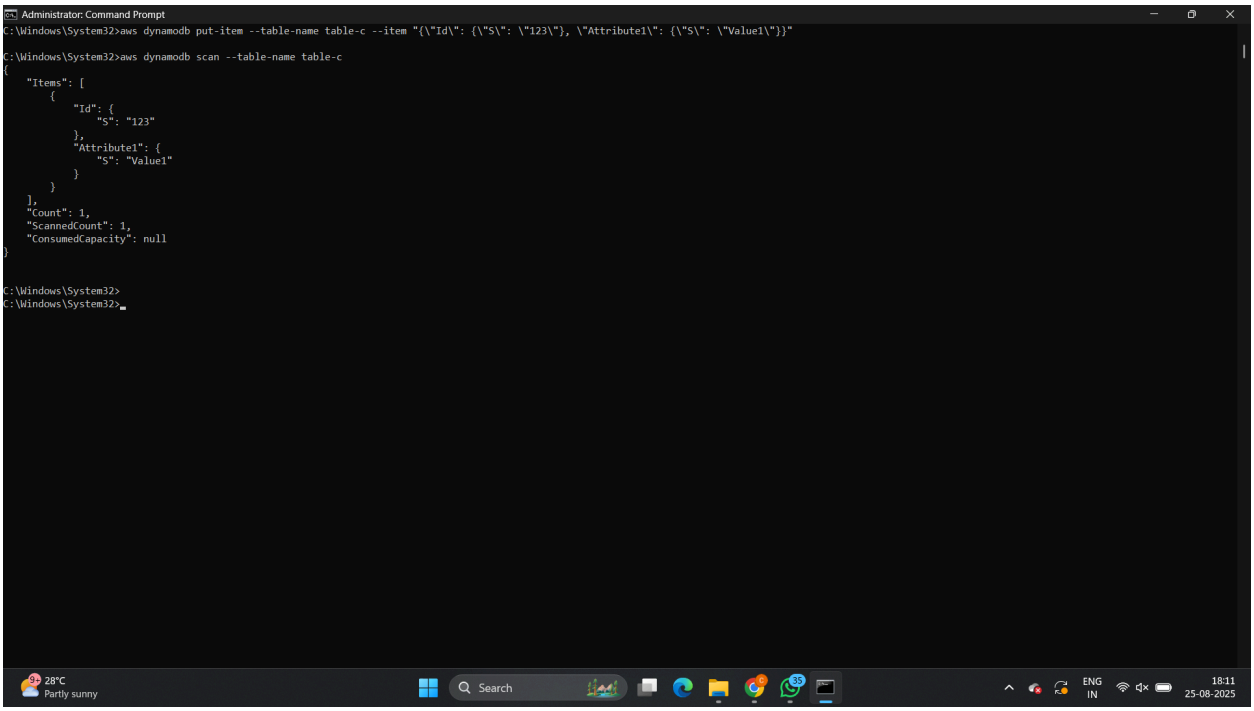

Administrator: Command Prompt
Microsoft Windows [Version 10.0.26120.5751]
(c) Microsoft Corporation. All rights reserved.
C:\Windows\System32>aws dynamodb create-table --table-name table-c --attribute-definitions AttributeName=Id,AttributeType=S --key-schema AttributeName=Id,KeyType=HASH --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5
{
"TableDescription": {
"AttributeDefinitions": [
{
"AttributeName": "Id",
"AttributeType": "S"
}
],
"TableName": "table-c",
"KeySchema": [
{
"AttributeName": "Id",
"KeyType": "HASH"
}
],
"TableStatus": "CREATING",
"CreationDateTime": "2025-08-25T18:04:59.886000+05:30",
"ProvisionedThroughput": {
"NumberOfDecreasesToday": 0,
"ReadCapacityUnits": 5,
"WriteCapacityUnits": 5
},
"TableSizeBytes": 0,
"ItemCount": 0,
"TableArn": "arn:aws:dynamodb:ap-south-1:845958739988:table/table-c",
"TableId": "e81db670-97de-44c0-b516-e734766a6f5b",
"DeletionProtectionEnabled": false
}
}
C:\Windows\System32>
28°C
Partly sunny
Search
ENG
IN
18:05
25-08-2025

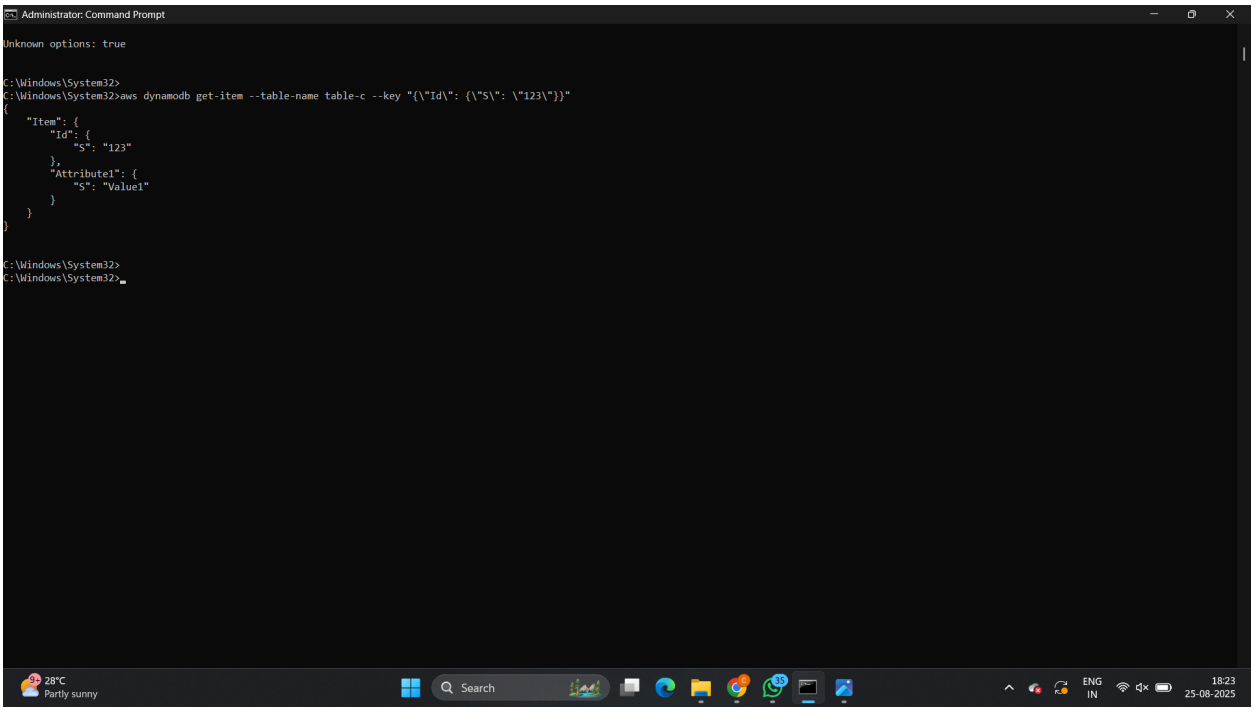

ap-south-1.console.aws.amazon.com/dynamodbv2/home#tables
Search
[Alt+S]
Asia Pacific (Mumbai)
Account ID: 8459-5873-9988
CHETAN
DynamoDB > Tables
DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
The request to delete the "table-c" table has been submitted successfully.
Tables (1) Info
Actions
Delete
Create table
Find tables
Any tag key
Any tag value
Name
Status
Partition key
Sort key
Indexes
Replication Regions
Deletion protection
Favorite
Read capacity
table-c
Active
Id (S)
-
0
0
Off
Provisioned (5)
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Partly sunny
Search
ENG
IN
18:05
25-08-2025

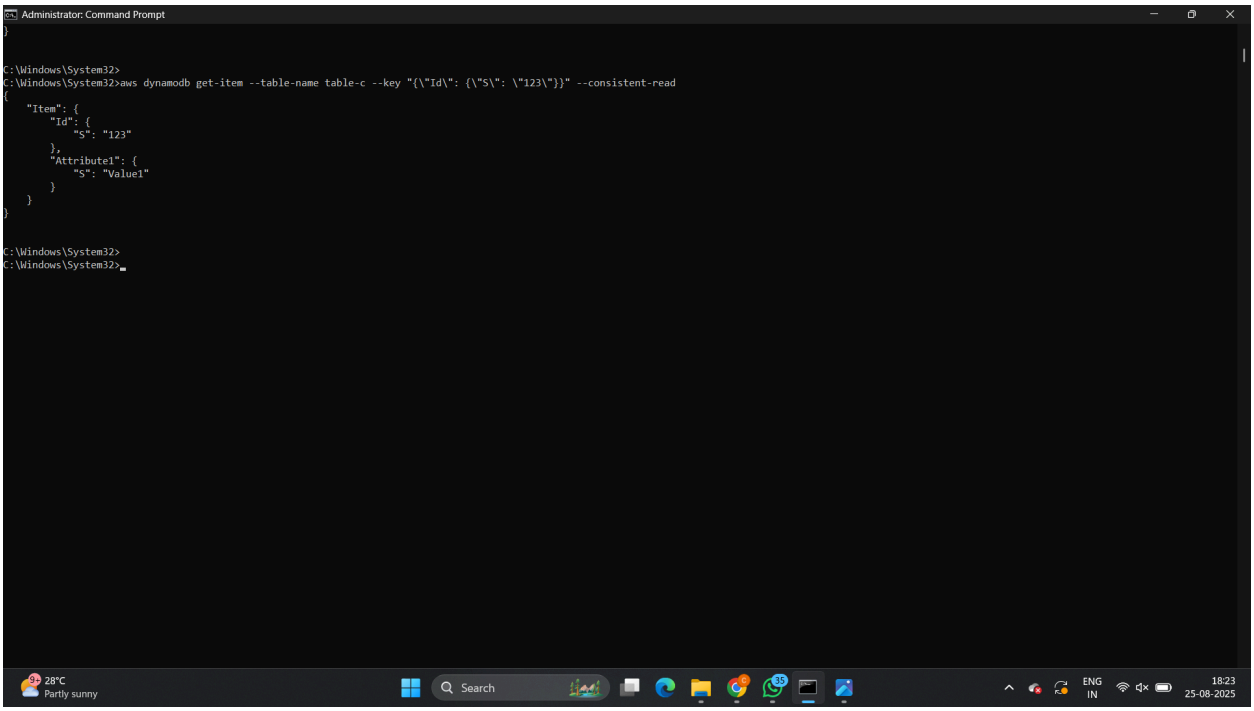

Administrator: Command Prompt
"WriteCapacityUnits": 5
},
"TableSizeBytes": 0,
"ItemCount": 0,
"TableArn": "arn:aws:dynamodb:ap-south-1:845958739988:table/table-c",
"TableId": "e81db670-97de-44c0-b516-e734766a6f5b",
"DeletionProtectionEnabled": false
}
}
C:\Windows\System32>aws dynamodb describe-table --table-name table-c
{
"Table": {
"AttributeDefinitions": [
{
"AttributeName": "Id",
"AttributeType": "S"
}
],
"TableName": "table-c",
"KeySchema": [
{
"AttributeName": "Id",
"KeyType": "HASH"
}
],
"TableStatus": "ACTIVE",
"CreationDateTime": "2025-08-25T18:04:59.886000+05:30",
"ProvisionedThroughput": {
"NumberOfDecreasesToday": 0,
"ReadCapacityUnits": 5,
"WriteCapacityUnits": 5
},
"TableSizeBytes": 0,
"ItemCount": 0,
"TableArn": "arn:aws:dynamodb:ap-south-1:845958739988:table/table-c",
"TableId": "e81db670-97de-44c0-b516-e734766a6f5b",
"DeletionProtectionEnabled": false,
"WarmThroughput": {
"ReadUnitsPerSecond": 5,
"WriteUnitsPerSecond": 5,
"Status": "ACTIVE"
}
}
}
C:\Windows\System32>
28°C
Partly sunny
Search
ENG
IN
18:05
25-08-2025


Administrator: Command Prompt
C:\Windows\System32>aws dynamodb put-item --table-name table-c --item "{\"Id\": {\"S\": \"123\"}, \"Attribute1\": {\"S\": \"Value1\"}}"
C:\Windows\System32>
C:\Windows\System32>
28°C
Partly sunny
Search
ENG
IN
18:07
25-08-2025


Best Cloud Computing Online
Course: [NEW] Ultimate AWS
Items | Amazon DynamoDB
C:\Users\cheta>aws dynamodb
ap-south-1.console.aws.amazon.com/dynamodbv2/home#item-explorer?maximize=true&table=table-c
Search
[Alt+S]
Asia Pacific (Mumbai)
Account ID: 8459-5873-9988
CHETAN
DynamoDB > Explore items > table-c
DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
Any tag value
Find tables
table-c
Scan
Query
Select a table or index
Table - table-c
Select attribute projection
All attributes
Filters - optional
Run
Reset
Completed · Items returned: 1 · Items scanned: 1 · Efficiency: 100% · RCUs consumed: 0.5
Table: table-c - Items returned (1)
Scan started on August 25, 2025, 18:08:28
Actions
Create item
Id (String)
Attribute1
123
Value1
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Partly sunny
Search
ENG
IN
18:08
25-08-2025


Administrator: Command Prompt
C:\Windows\System32>aws dynamodb put-item --table-name table-c --item "{\"Id\": {\"S\": \"123\"}, \"Attribute1\": {\"S\": \"Value1\"}}"
C:\Windows\System32>aws dynamodb scan --table-name table-c
{
"Items": [
{
"Id": {
"S": "123"
},
"Attribute1": {
"S": "Value1"
}
}
],
"Count": 1,
"ScannedCount": 1,
"ConsumedCapacity": null
}
C:\Windows\System32>
C:\Windows\System32>
28°C
Partly sunny
Search
ENG
IN
18:11
25-08-2025





Task 3: Experience Throttling in Provisioned Mode

- Create a table with low provisioned throughput, e.g., 1 RCU and 1 WCU.

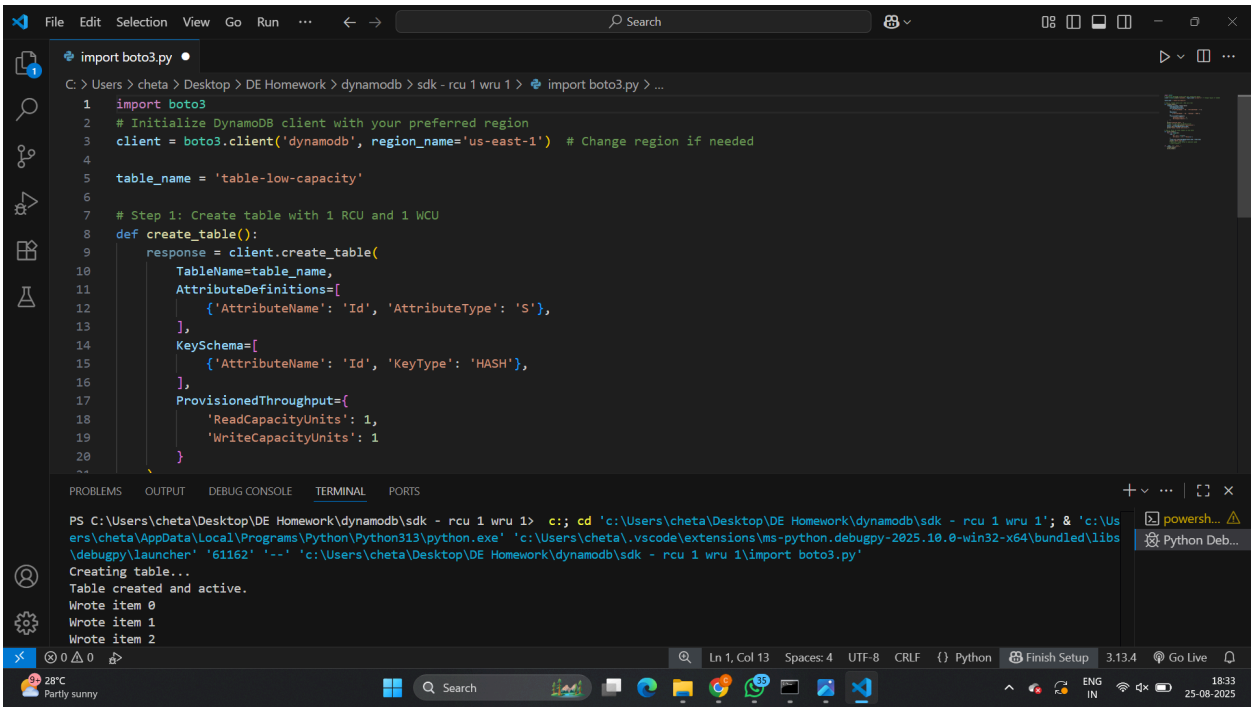
- Use a script (e.g., Python Boto3) to attempt rapid write operations (e.g., 10 writes in a loop).
- Observe `ProvisionedThroughputExceededException` errors indicating throttling.
- Learn importance of:
  - Proper capacity provisioning based on load.
  - Implementing retry and backoff logic.
  - Using Auto Scaling or On-Demand mode for variable workloads.

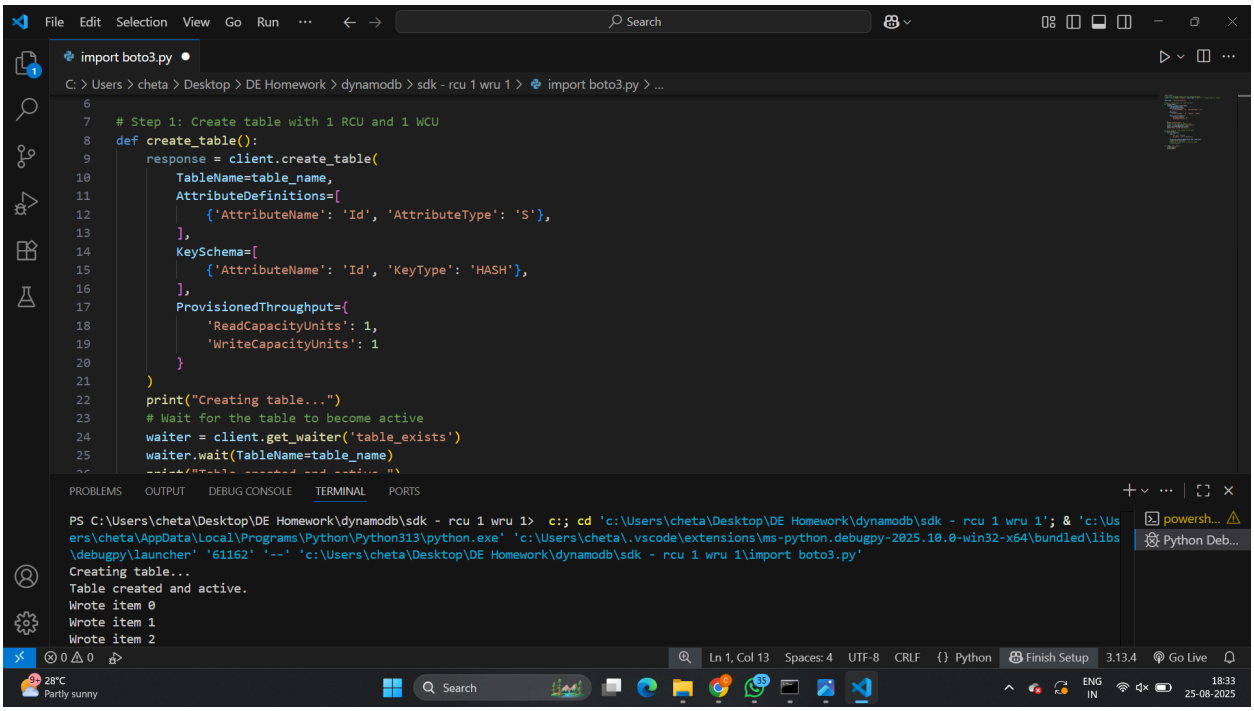
CODE:

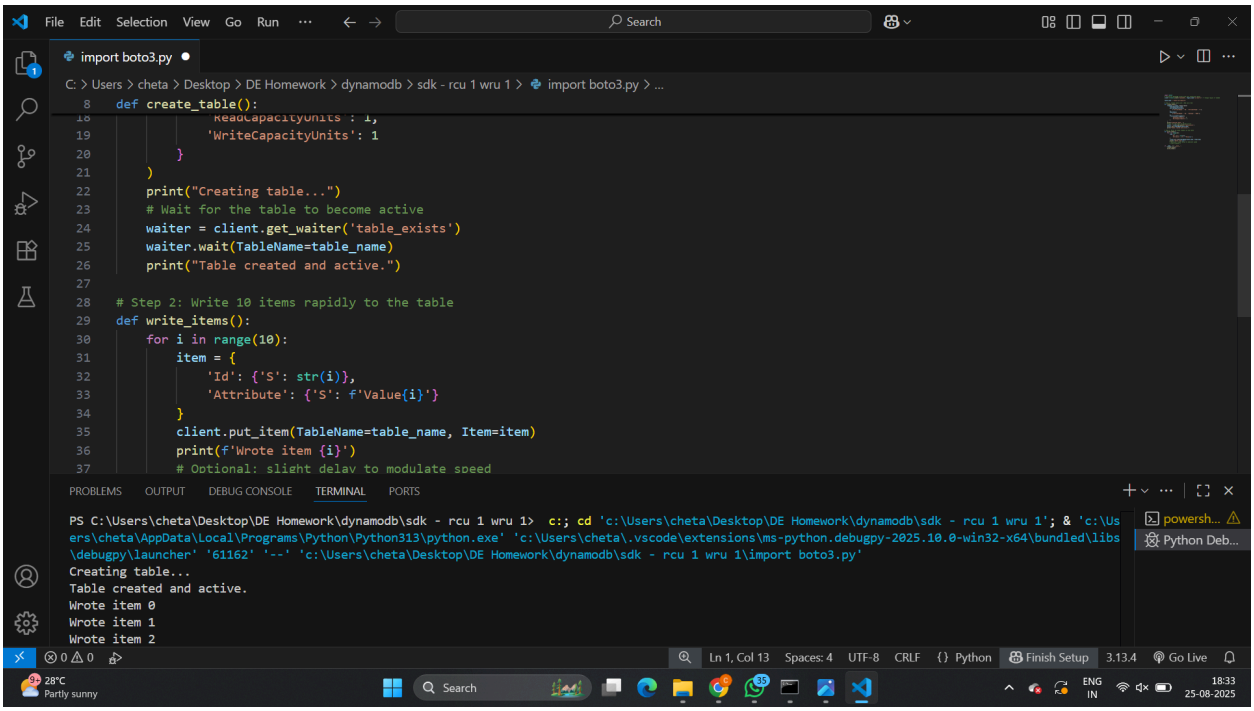
```
import boto3
import time

# Initialize DynamoDB client with your preferred region

client = boto3.client('dynamodb', region_name='us-east-1')

# Change region if needed

table_name = 'table-low-capacity'

# Step 1: Create table with 1 RCU and 1 WCU

def create_table():
    response = client.create_table(
        TableName=table_name,
        AttributeDefinitions=[
            {'AttributeName': 'Id', 'AttributeType': 'S'},
        ],
        KeySchema=[
            {'AttributeName': 'Id', 'KeyType': 'HASH'},
        ],
        ProvisionedThroughput={
            'ReadCapacityUnits': 1,
            'WriteCapacityUnits': 1
        }
```

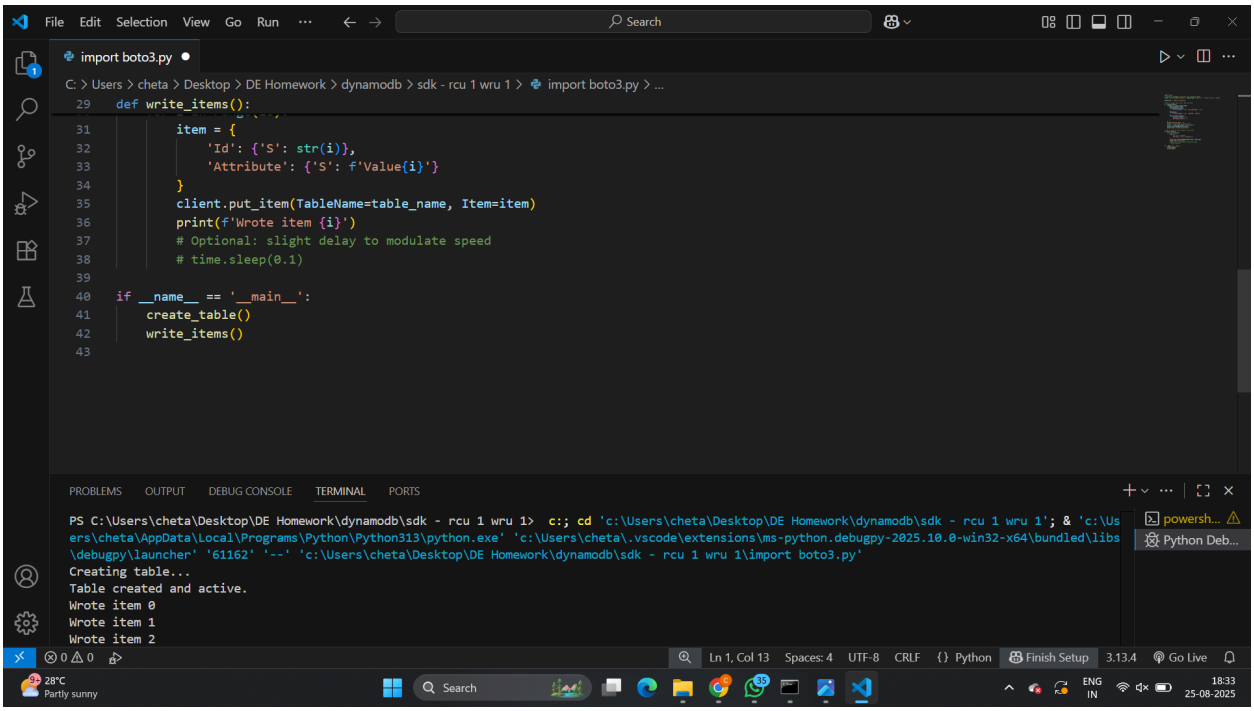
```
    )
    print("Creating table...")

# Wait for the table to become active

waiter = client.get_waiter('table_exists')
    waiter.wait(TableName=table_name)
    print("Table created and active.")

# Step 2: Write 10 items rapidly to the table

def write_items():
    for i in range(10):
        item = {
            'Id': {'S': str(i)},
            'Attribute': {'S': f'Value{i}'}
        }
        client.put_item(TableName=table_name, Item=item)
        print(f'Wrote item {i}')

# Optional: slight delay to modulate speed

# time.sleep(0.1)

if __name__ == '__main__':
    create_table()
    write_items()
```

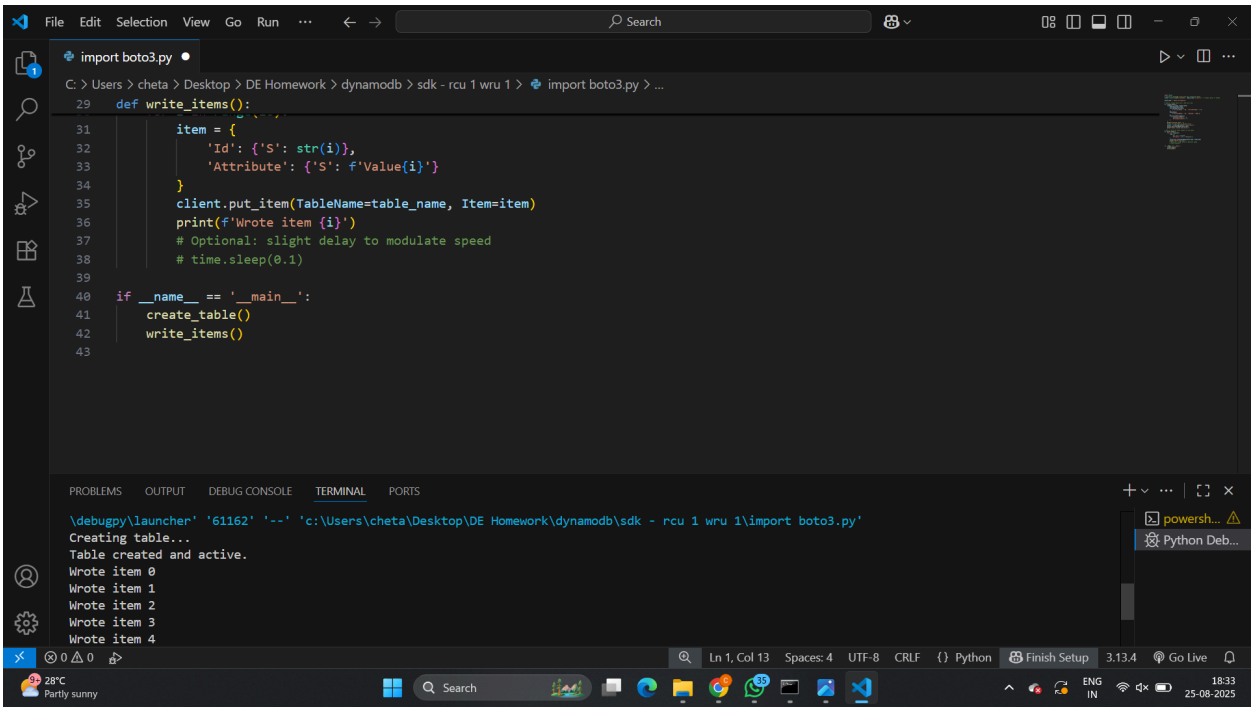
Screenshots:

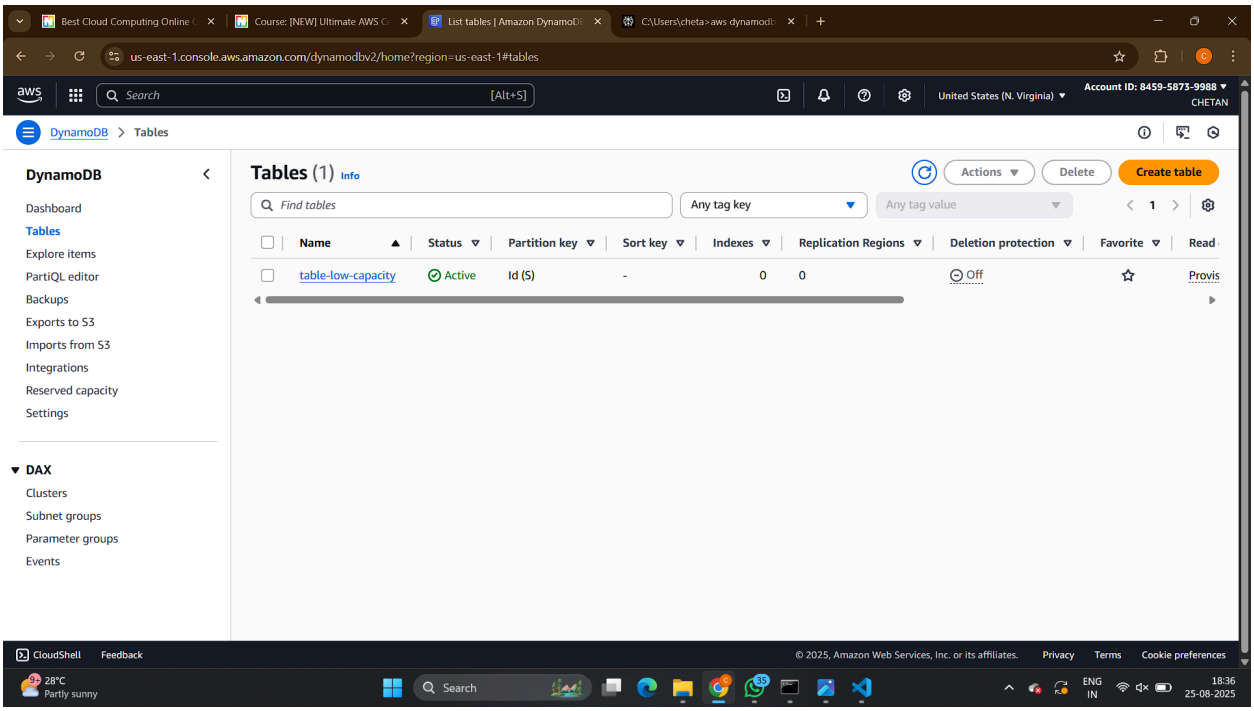

File Edit Selection View Go Run
Search
import boto3.py
C: > Users > cheta > Desktop > DE Homework > dynamodb > sdk - rcu 1 wru 1 > import boto3.py > ...
import boto3
# Initialize DynamoDB client with your preferred region
client = boto3.client('dynamodb', region_name='us-east-1') # Change region if needed
table_name = 'table-low-capacity'
# Step 1: Create table with 1 RCU and 1 WCU
def create_table():
response = client.create_table(
TableName=table_name,
AttributeDefinitions=[
{'AttributeName': 'Id', 'AttributeType': 'S'},
],
KeySchema=[
{'AttributeName': 'Id', 'KeyType': 'HASH'},
],
ProvisionedThroughput={
'ReadCapacityUnits': 1,
'WriteCapacityUnits': 1
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1> c:; cd 'c:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1'; & 'c:\Users\cheta\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\cheta\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '61162' '--' 'c:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1\import boto3.py'
Creating table...
Table created and active.
Wrote item 0
Wrote item 1
Wrote item 2
powersh...
Python Deb...
Ln 1, Col 13 Spaces: 4 UTF-8 CRLF {} Python Finish Setup 3.13.4 Go Live
28°C Partly sunny
Search
ENG IN
18:33 25-08-2025

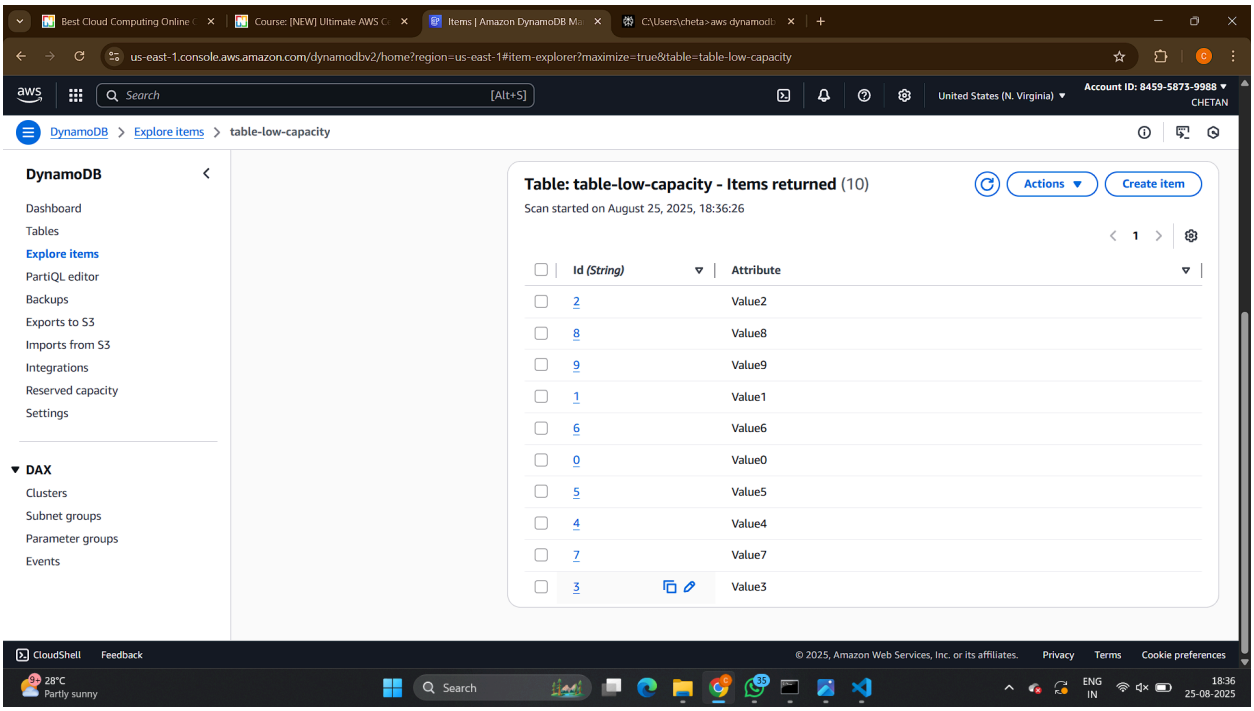

File Edit Selection View Go Run
Search
import boto3.py
C: > Users > cheta > Desktop > DE Homework > dynamodb > sdk - rcu 1 wru 1 > import boto3.py > ...
# Step 1: Create table with 1 RCU and 1 WCU
def create_table():
response = client.create_table(
TableName=table_name,
AttributeDefinitions=[
{'AttributeName': 'Id', 'AttributeType': 'S'},
],
KeySchema=[
{'AttributeName': 'Id', 'KeyType': 'HASH'},
],
ProvisionedThroughput={
'ReadCapacityUnits': 1,
'WriteCapacityUnits': 1
}
)
print("Creating table...")
# Wait for the table to become active
waiter = client.get_waiter('table_exists')
waiter.wait(TableName=table_name)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1> c:; cd 'c:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1'; & 'c:\Users\cheta\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\cheta\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '61162' '--' 'c:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1\import boto3.py'
Creating table...
Table created and active.
Wrote item 0
Wrote item 1
Wrote item 2
powersh...
Python Deb...
Ln 1, Col 13 Spaces: 4 UTF-8 CRLF {} Python Finish Setup 3.13.4 Go Live
28°C Partly sunny
Search
ENG IN
18:33 25-08-2025


File Edit Selection View Go Run
import boto3.py
C: > Users > cheta > Desktop > DE Homework > dynamodb > sdk - rcu 1 wru 1 > import boto3.py > ...
def create_table():
'WriteCapacityUnits': 1
print("Creating table...")
# Wait for the table to become active
waiter = client.get_waiter('table_exists')
waiter.wait(TableName=table_name)
print("Table created and active.")
# Step 2: Write 10 items rapidly to the table
def write_items():
for i in range(10):
item = {
'Id': {'S': str(i)},
'Attribute': {'S': f'Value{i}'}
}
client.put_item(TableName=table_name, Item=item)
print(f'Wrote item {i}')
# Optional: slight delay to modulate speed
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1> c:; cd 'c:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1'; & 'c:\Users\cheta\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\cheta\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '61162' '--' 'c:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1\import boto3.py'
Creating table...
Table created and active.
Wrote item 0
Wrote item 1
Wrote item 2
Ln 1, Col 13 Spaces: 4 UTF-8 CRLF Python Finish Setup 3.13.4 Go Live
28°C Partly sunny
Search
ENG IN
18:33 25-08-2025


File Edit Selection View Go Run
import boto3.py
C: > Users > cheta > Desktop > DE Homework > dynamodb > sdk - rcu 1 wru 1 > import boto3.py > ...
def write_items():
item = {
'Id': {'S': str(i)},
'Attribute': {'S': f'Value{i}'}
}
client.put_item(TableName=table_name, Item=item)
print(f'Wrote item {i}')
# Optional: slight delay to modulate speed
# time.sleep(0.1)
if __name__ == '__main__':
create_table()
write_items()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1> c:; cd 'c:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1'; & 'c:\Users\cheta\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\cheta\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '61162' '--' 'c:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1\import boto3.py'
Creating table...
Table created and active.
Wrote item 0
Wrote item 1
Wrote item 2
Ln 1, Col 13 Spaces: 4 UTF-8 CRLF Python Finish Setup 3.13.4 Go Live
28°C Partly sunny
Search
ENG IN
18:33 25-08-2025


File Edit Selection View Go Run
import boto3.py
C: > Users > cheta > Desktop > DE Homework > dynamodb > sdk - rcu 1 wru 1 > import boto3.py > ...
def write_items():
item = {
'Id': {'S': str(i)},
'Attribute': {'S': f'Value{i}'}
}
client.put_item(TableName=table_name, Item=item)
print(f'Wrote item {i}')
# Optional: slight delay to modulate speed
# time.sleep(0.1)
if __name__ == '__main__':
create_table()
write_items()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
\debugpy\launcher' '61162' '--' 'c:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1\import boto3.py'
Creating table...
Table created and active.
Wrote item 0
Wrote item 1
Wrote item 2
Wrote item 3
Wrote item 4
Ln 1, Col 13 Spaces: 4 UTF-8 CRLF Python Finish Setup 3.13.4 Go Live
28°C Partly sunny
Search
ENG IN
18:33 25-08-2025


us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#tables
Search [Alt+S]
United States (N. Virginia)
Account ID: 8459-5873-9988
CHETAN
DynamoDB > Tables
DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
Tables (1) Info
Actions
Delete
Create table
Find tables
Any tag key
Any tag value
Name
Status
Partition key
Sort key
Indexes
Replication Regions
Deletion protection
Favorite
Read
table-low-capacity
Active
Id (S)
-
0
0
Off
Provis
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C Partly sunny
Search
ENG IN
18:36 25-08-2025



Task 4: Cost Calculation Exercise for Provisioned Throughput

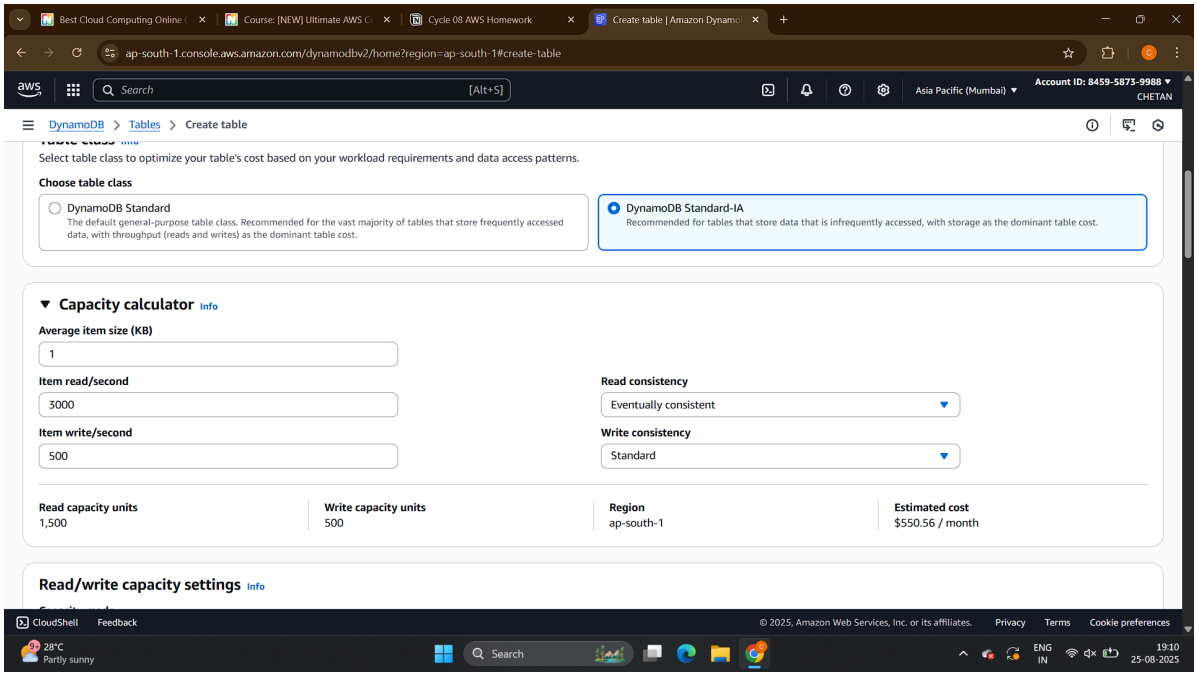
Given workload:

- Writes: 500/sec, item size 2.3 KB
- Reads: 3000/sec (eventual consistency), item size 5.1 KB

Calculations:

- **Write Capacity Units (WCU)**:
  - 1 WCU = 1 write per second for up to 1 KB
  - Each write is 2.3 KB ≈ 3 WCUs per write
  - Total WCU = 500 writes/sec * 3 = 1500 WCUs
- **Read Capacity Units (RCU)**:
  - 1 RCU = 1 strongly consistent or 2 eventually consistent reads per second for up to 4 KB
  - 5.1 KB item > 4 KB, use ceil(5.1/4) = 2 RCUs per read

- Eventual consistency halves RCU usage (1 RCU supports 2 reads)
- Total RCU = (3000 reads / 2) * 2 = 3000 RCUs



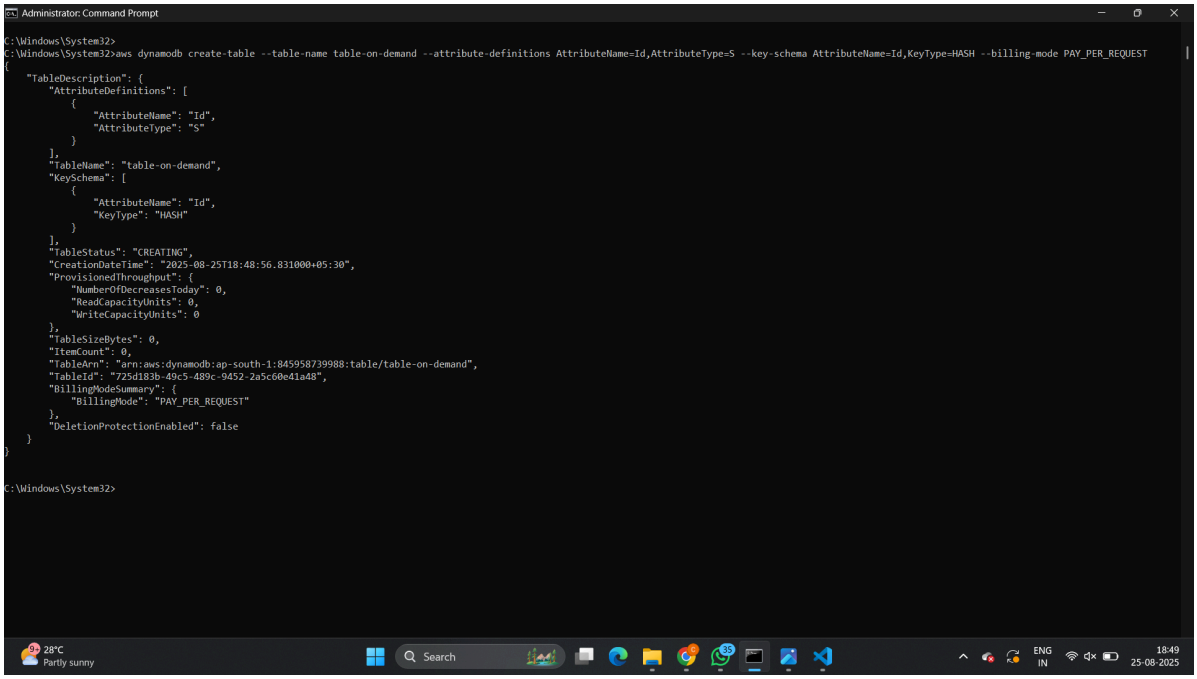
Task 5: Explore Autoscaling / On-Demand Mode

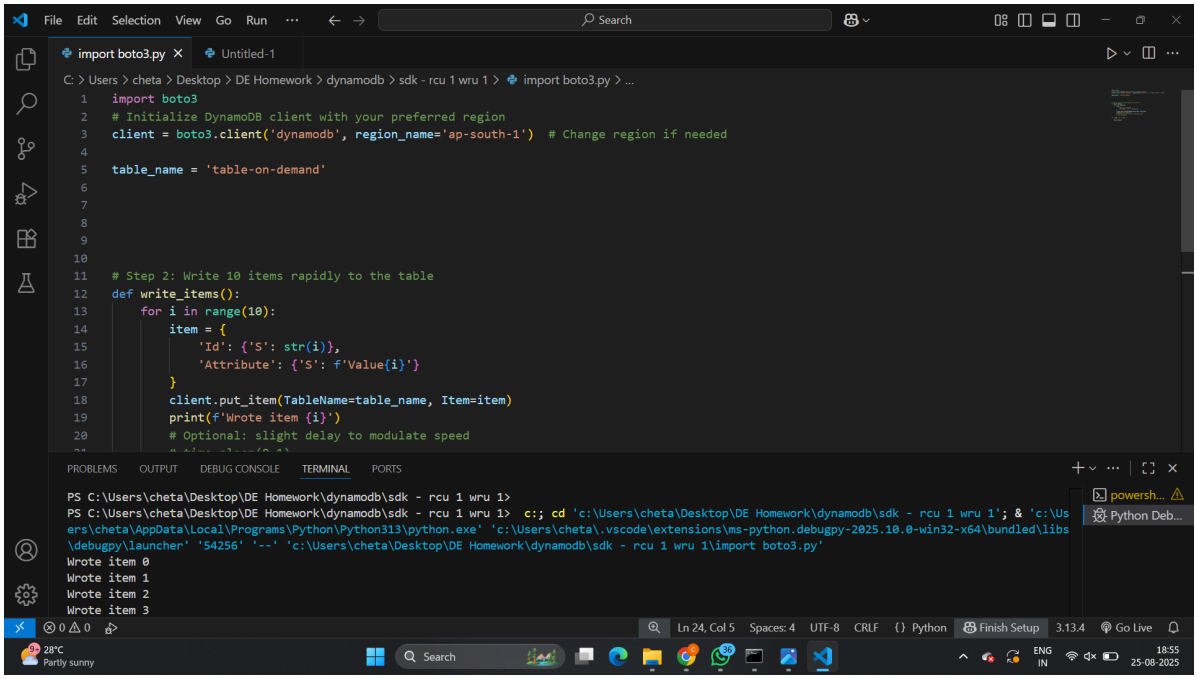
- Create a new DynamoDB table with **On-Demand** capacity mode (no manual RCU/WCU settings).
- Run rapid write tests as in Task 3 script.
- Observe no throttling occurs unless hitting overall account limits.
- Understand that On-Demand mode automatically adjusts capacity instantly to match load.
- Consider pricing implications:
  - On-Demand is more expensive but offers flexibility.
  - Provisioned mode with Auto Scaling offers cost savings but slower capacity adjustment.

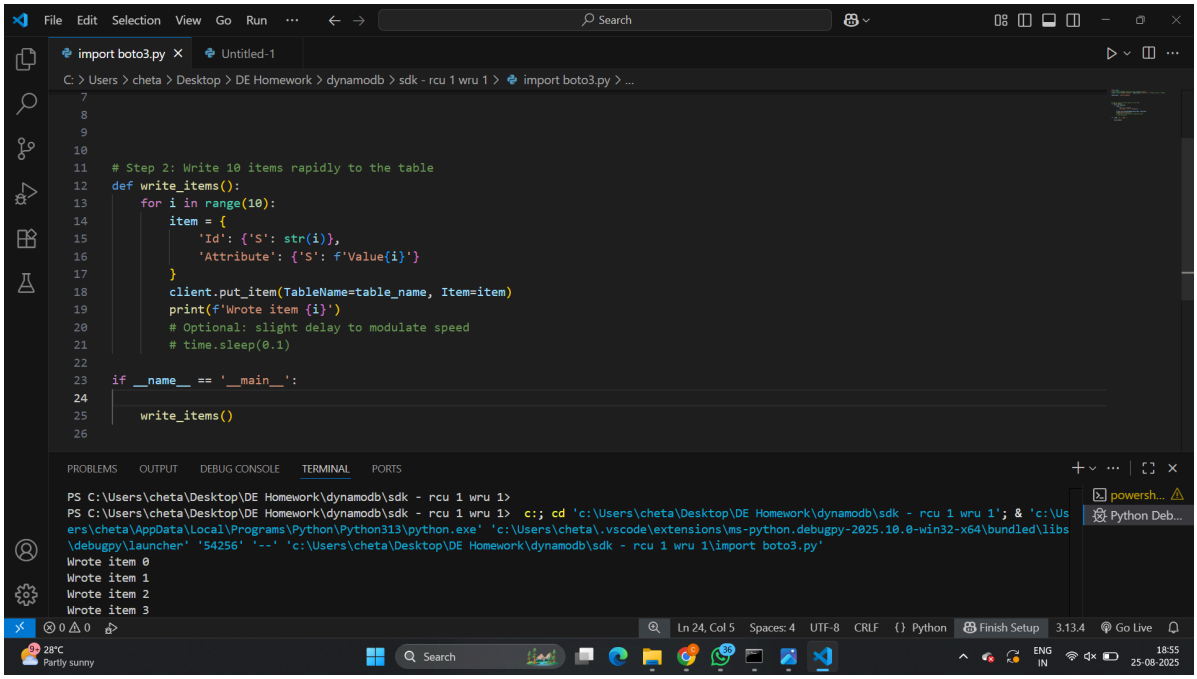
CODE:

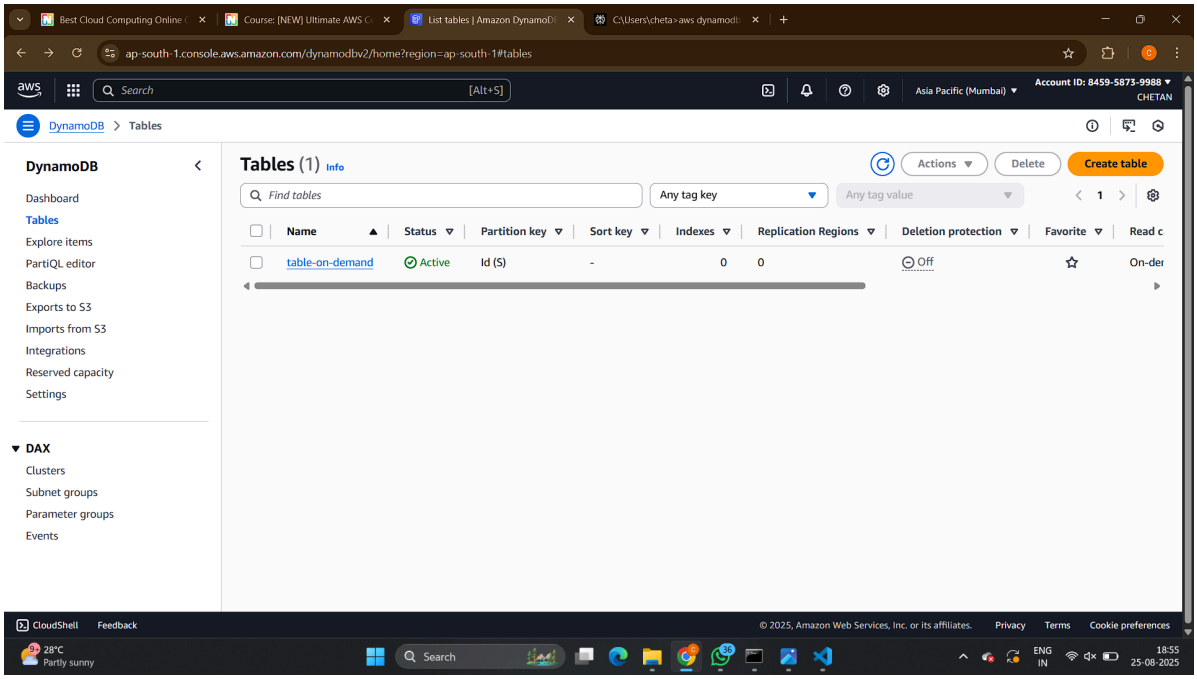
```
import boto3
# Initialize DynamoDB client with your preferred region
client = boto3.client('dynamodb', region_name='ap-south-1')  # Change re
gion if needed

table_name = 'table-on-demand'




# Step 2: Write 10 items rapidly to the table
def write_items():
    for i in range(10):
        item = {
            'Id': {'S': str(i)},
            'Attribute': {'S': f'Value{i}'}
        }
        client.put_item(TableName=table_name, Item=item)
        print(f'Wrote item {i}')
        # Optional: slight delay to modulate speed
        # time.sleep(0.1)

if __name__ == '__main__':

    write_items()
```

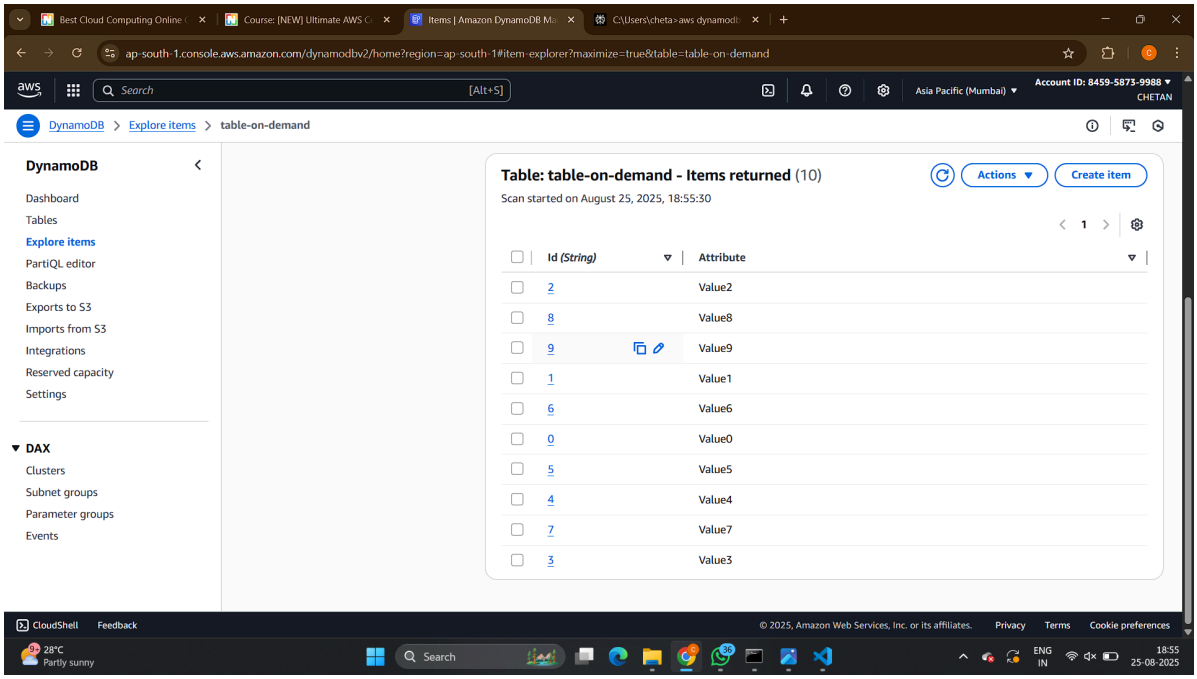
SCREENSHOTS:


Administrator: Command Prompt
C:\Windows\System32>
C:\Windows\System32>aws dynamodb create-table --table-name table-on-demand --attribute-definitions AttributeName=Id,AttributeType=S --key-schema AttributeName=Id,KeyType=HASH --billing-mode PAY_PER_REQUEST
{
"TableDescription": {
"AttributeDefinitions": [
{
"AttributeName": "Id",
"AttributeType": "S"
}
],
"TableName": "table-on-demand",
"KeySchema": [
{
"AttributeName": "Id",
"KeyType": "HASH"
}
],
"TableStatus": "CREATING",
"CreationDateTime": "2025-08-25T18:48:56.831000+05:30",
"ProvisionedThroughput": {
"NumberOfDecreasesToday": 0,
"ReadCapacityUnits": 0,
"WriteCapacityUnits": 0
},
"TableSizeBytes": 0,
"ItemCount": 0,
"TableArn": "arn:aws:dynamodb:ap-south-1:845958739988:table/table-on-demand",
"TableId": "725d183b-49c5-489c-9452-2a5c60e41a48",
"BillingModeSummary": {
"BillingMode": "PAY_PER_REQUEST"
},
"DeletionProtectionEnabled": false
}
}
C:\Windows\System32>
28°C
Partly sunny
Search
ENG
IN
18:49
25-08-2025


File Edit Selection View Go Run
Search
import boto3.py
Untitled-1
C: > Users > cheta > Desktop > DE Homework > dynamodb > sdk - rcu 1 wru 1 > import boto3.py > ...
import boto3
# Initialize DynamoDB client with your preferred region
client = boto3.client('dynamodb', region_name='ap-south-1') # Change region if needed
table_name = 'table-on-demand'
# Step 2: Write 10 items rapidly to the table
def write_items():
for i in range(10):
item = {
'Id': {'S': str(i)},
'Attribute': {'S': f'Value{i}'}
}
client.put_item(TableName=table_name, Item=item)
print(f'Wrote item {i}')
# Optional: slight delay to modulate speed
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1>
PS C:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1> c:; cd 'c:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1'; & 'c:\Users\cheta\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\cheta\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '54256' '--' 'c:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1\import boto3.py'
Wrote item 0
Wrote item 1
Wrote item 2
Wrote item 3
powersh...
Python Deb...
Ln 24, Col 5 Spaces: 4 UTF-8 CRLF {} Python Finish Setup 3.13.4 Go Live
28°C
Partly sunny
Search
ENG
IN
18:55
25-08-2025


import boto3.py
Untitled-1
C: > Users > cheta > Desktop > DE Homework > dynamodb > sdk - rcu 1 wru 1 > import boto3.py > ...
# Step 2: Write 10 items rapidly to the table
def write_items():
for i in range(10):
item = {
'Id': {'S': str(i)},
'Attribute': {'S': f'Value{i}'}
}
client.put_item(TableName=table_name, Item=item)
print(f'Wrote item {i}')
# Optional: slight delay to modulate speed
# time.sleep(0.1)
if __name__ == '__main__':
write_items()
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1>
PS C:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1> c:; cd 'c:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1'; & 'c:\Users\cheta\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\cheta\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '54256' '--' 'c:\Users\cheta\Desktop\DE Homework\dynamodb\sdk - rcu 1 wru 1\import boto3.py'
Wrote item 0
Wrote item 1
Wrote item 2
Wrote item 3
Ln 24, Col 5
Spaces: 4
UTF-8
CRLF
Python
Finish Setup
3.13.4
Go Live


ap-south-1.console.aws.amazon.com/dynamodbv2/home?region=ap-south-1#tables
Asia Pacific (Mumbai)
Account ID: 8459-5873-9988
CHETAN
DynamoDB > Tables
DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
Tables (1) Info
Actions
Delete
Create table
Find tables
Any tag key
Any tag value
Name
Status
Partition key
Sort key
Indexes
Replication Regions
Deletion protection
Favorite
table-on-demand
Active
Id (S)
0
0
Off
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Best Cloud Computing Online
Course: [NEW] Ultimate AWS
Items | Amazon DynamoDB Ma
C:\Users\cheta>aws dynamodb
ap-south-1.console.aws.amazon.com/dynamodbv2/home?region=ap-south-1#item-explorer?maximize=true&table=table-on-demand
Search
[Alt+S]
Asia Pacific (Mumbai)
Account ID: 8459-5873-9988
CHETAN
DynamoDB
Explore items
table-on-demand
DynamoDB
Dashboard
Tables
Explore items
PartiQL editor
Backups
Exports to S3
Imports from S3
Integrations
Reserved capacity
Settings
DAX
Clusters
Subnet groups
Parameter groups
Events
Table: table-on-demand - Items returned (10)
Scan started on August 25, 2025, 18:55:30
Actions
Create item
Id (String)
Attribute
2
Value2
8
Value8
9
Value9
1
Value1
6
Value6
0
Value0
5
Value5
4
Value4
7
Value7
3
Value3
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Partly sunny
Search
ENG
IN
18:55
25-08-2025