<https://dynobase.dev/dynamodb-cli-query-examples/>

Exercise 1: Create a Table  
  
• Objective: Create a DynamoDB table to store information about books.  
• Steps: Follow the previous instructions to create a “Books” table.  
  
Exercise 2: Insert Data  
  
• Objective: Add at least 5 different book records into the “Books” table.  
• Steps: Use the “PutItem” operation to insert data into the table.  
  
Exercise 3: Query Data  
  
• Objective: Retrieve a book using its ISBN.  
• Steps: Use the “Query” operation to search for a book by ISBN.  
  
Exercise 4: Update Data  
  
• Objective: Modify the details of an existing book.  
• Steps: Use the “UpdateItem” operation to change an attribute of a book.  
  
Exercise 5: Delete Data  
  
• Objective: Remove a book record from the table.  
• Steps: Use the “DeleteItem” operation to delete a book by its ISBN.  
  
Exercise 6: Scan Data  
  
• Objective: Retrieve all books written by a specific author.  
• Steps: Use the “Scan” operation to find books by the author’s name.  
  
Exercise 7: Secondary Index  
  
• Objective: Create a Global Secondary Index (GSI) on the “Genre” attribute.  
• Steps: Create a GSI to efficiently query books by genre.  
  
Exercise 8: Backup and Restore  
  
• Objective: Perform a backup of your “Books” table and then restore it.  
• Steps: Explore how to create backups and restore the table from a backup.  
  
Exercise 9: Conditional Writes  
  
• Objective: Implement a conditional write operation to prevent overwriting an existing book.  
• Steps: Use a conditional expression to ensure you only add a book if it doesn’t already exist.  
  
Exercise 10: Batch Write and Delete  
  
• Objective: Use batch operations to add multiple books at once and then delete them.  
• Steps: Experiment with “BatchWriteItem” and “BatchDeleteItem” operations to work with multiple items simultaneously.  
  
These 10 exercises will help you get a comprehensive understanding of Amazon DynamoDB, from table creation to advanced operations like conditional writes and batch operations.

**Solution 1:**

aws dynamodb create-table \

--table-name Books \

--attribute-definitions \

AttributeName=isbn,AttributeType=N \

AttributeName=author,AttributeType=S \

--key-schema \

AttributeName=isbn,KeyType=HASH \

AttributeName=author,KeyType=RANGE \

--provisioned-throughput \

ReadCapacityUnits=5,WriteCapacityUnits=5

On demand table

aws dynamodb create-table \

--table-name BooksTemp \

--attribute-definitions \

AttributeName=BookName,AttributeType=S \

AttributeName=Author,AttributeType=S \

--key-schema \

AttributeName=BookName,KeyType=HASH \

AttributeName=Author,KeyType=RANGE \

--billing-mode=PAY\_PER\_REQUEST

**Solution 2:**

aws dynamodb put-item \

--table-name Books \

--item \

'{"isbn": {"N": "111"}, "author": {"S": "Robert Kiyosaki"}, "name": {"S": "Rich Dad Poor Dad"} , "prize": {"N": "1000"}, "genre": {"S": "finance"}}'

aws dynamodb put-item \

--table-name Books \

--item \

'{"isbn": {"N": "222"}, "author": {"S": "Robert Kiyosaki"}, "name": {"S": "Fake"} , "prize": {"N": "1500"}, "genre": {"S": "finance"}}'

aws dynamodb put-item \

--table-name Books \

--item \

'{"isbn": {"N": "333"}, "author": {"S": "Rowling"}, "name": {"S": "Harry Potter 1"} , "prize": {"N": "500"}, "genre": {"S": "adventure"}}'

aws dynamodb put-item \

--table-name Books \

--item \

'{"isbn": {"N": "444"}, "author": {"S": "Rowling"}, "name": {"S": "Harry Potter 2"}, "prize": {"N": "2000"}, "genre": {"S": "adventure"}}'

aws dynamodb put-item \

--table-name Books \

--item \

'{"isbn": {"N": "555"}, "author": {"S": "Rowling"}, "name": {"S": "Harry Potter 3"}, "prize": {"N": "9000"}, "genre": {"S": "adventure"}}'

**Solution 3:**

aws dynamodb query \

--table-name Books \

--key-condition-expression "isbn = :num" \

--expression-attribute-values '{":num":{"N":"222"}}'

aws dynamodb get-item --consistent-read \

--table-name Books \

--key '{ "isbn": {"N": "222"}}'

**Solution 4:**

aws dynamodb update-item \

--table-name Books \

--key '{ "isbn": {"N": "111"}, "author": {"S": "Robert Kiyosaki"}}' \

--update-expression "SET prize = :newval" \

--expression-attribute-values '{":newval":{"N":"5000"}}' \

--return-values ALL\_NEW

**Solution 5:**

aws dynamodb delete-item \

--table-name Books \

--key '{"isbn" : {"N": "555"},"author" : {"S": "Rowling"}}'

**Solution 6:**

aws dynamodb scan \

--table-name Books \

--filter-expression "author = :name" \

--expression-attribute-values '{":name":{"S":"Rowling"}}'

**Solution 7:**

aws dynamodb update-table \

--table-name Books \

--attribute-definitions AttributeName=genre,AttributeType=S \

--global-secondary-index-updates \

"[{\"Create\":{\"IndexName\": \"genre-index\",\"KeySchema\":[{\"AttributeName\":\"genre\",\"KeyType\":\"HASH\"}], \

\"ProvisionedThroughput\": {\"ReadCapacityUnits\": 1, \"WriteCapacityUnits\": 1 },\"Projection\":{\"ProjectionType\":\"ALL\"}}}]"

aws dynamodb query \

--table-name Books \

--index-name genre-index \

--key-condition-expression "genre = :name" \

--expression-attribute-values '{":name":{"S":"finance"}}'

**Solution 8:**

aws dynamodb create-backup \

--table-name Books \

--backup-name Books-backup

aws dynamodb list-backups --table-name Books

aws dynamodb restore-table-from-backup --target-table-name Books-Backup --backup-arn arn:aws:dynamodb:us-east-1:978244587858:table/Books/backup/01694672809403-03e0bd4f

**Solution 9:**

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Expressions.ConditionExpressions.html>

aws dynamodb put-item \

--table-name Books \

--item '{"isbn": {"N": "555"}, "author": {"S": "Rowling"}, "name": {"S": "Harry Potter 3"}, "prize": {"N": "9000"}, "genre": {"S": "adventure"}}' \

--condition-expression "attribute\_not\_exists(isbn)"

**Solution 10:**

<https://docs.aws.amazon.com/amazondynamodb/latest/APIReference/API_BatchWriteItem.html>

aws dynamodb batch-write-item \

--request-items '{ \

"Books" : [{ \

"PutRequest" : {"Item":{"isbn":{"N":"999"}, "author":{"S":"A1"}}}, \

"PutRequest" : {"Item":{"isbn":{"N":"888"}, "author":{"S":"A2"}}}, \

"DeleteRequest" : {"Key":{"isbn":{"N":"555"}, "author":{"S":"Rowling"}}} \

}]}'

Note. BatchWriteItem cannot update items. If you perform a BatchWriteItem operation on an existing item, that item's values will be overwritten by the operation and it will appear like it was updated. To update items, we recommend you use the UpdateItem action.