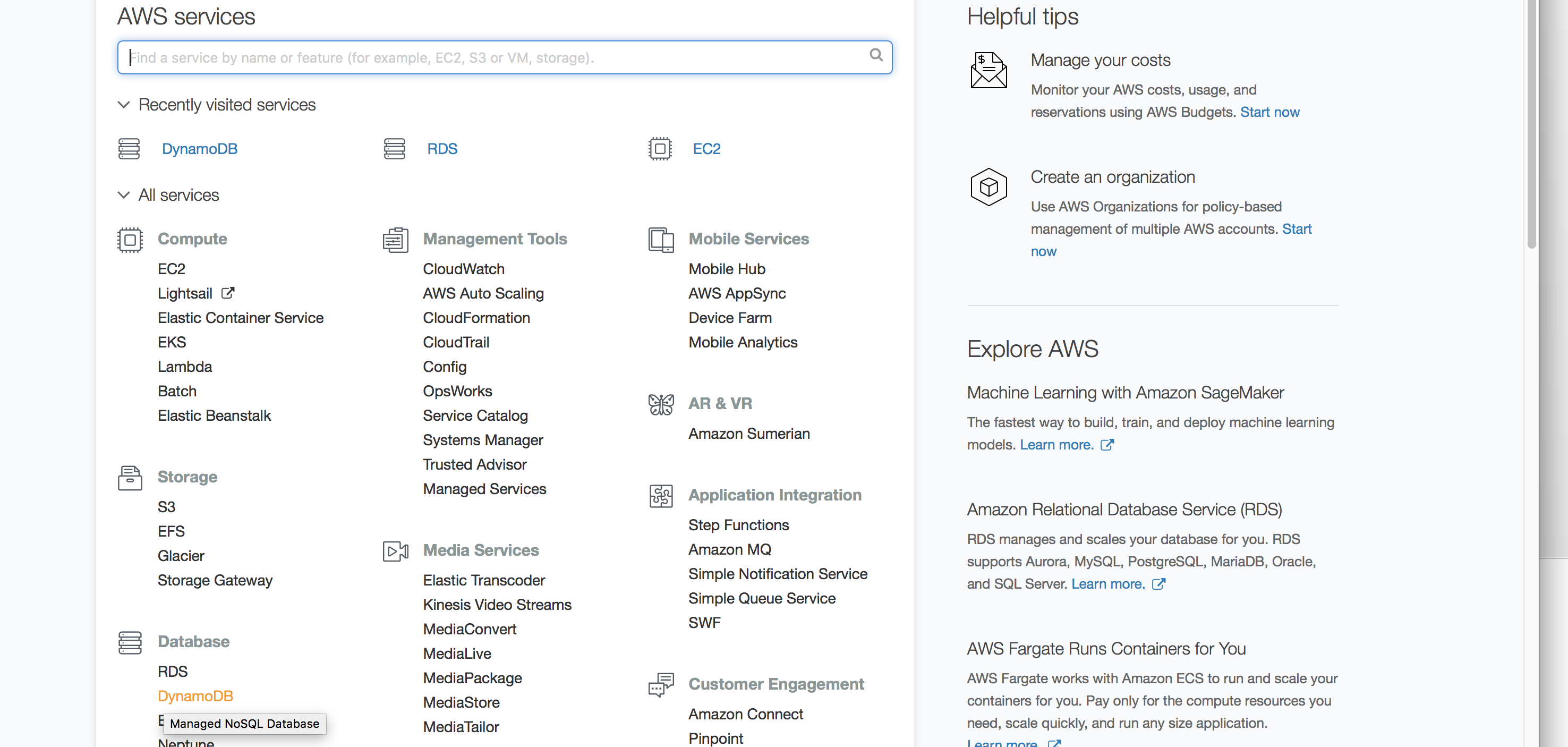
**Lab – DynamoDB Week 1**

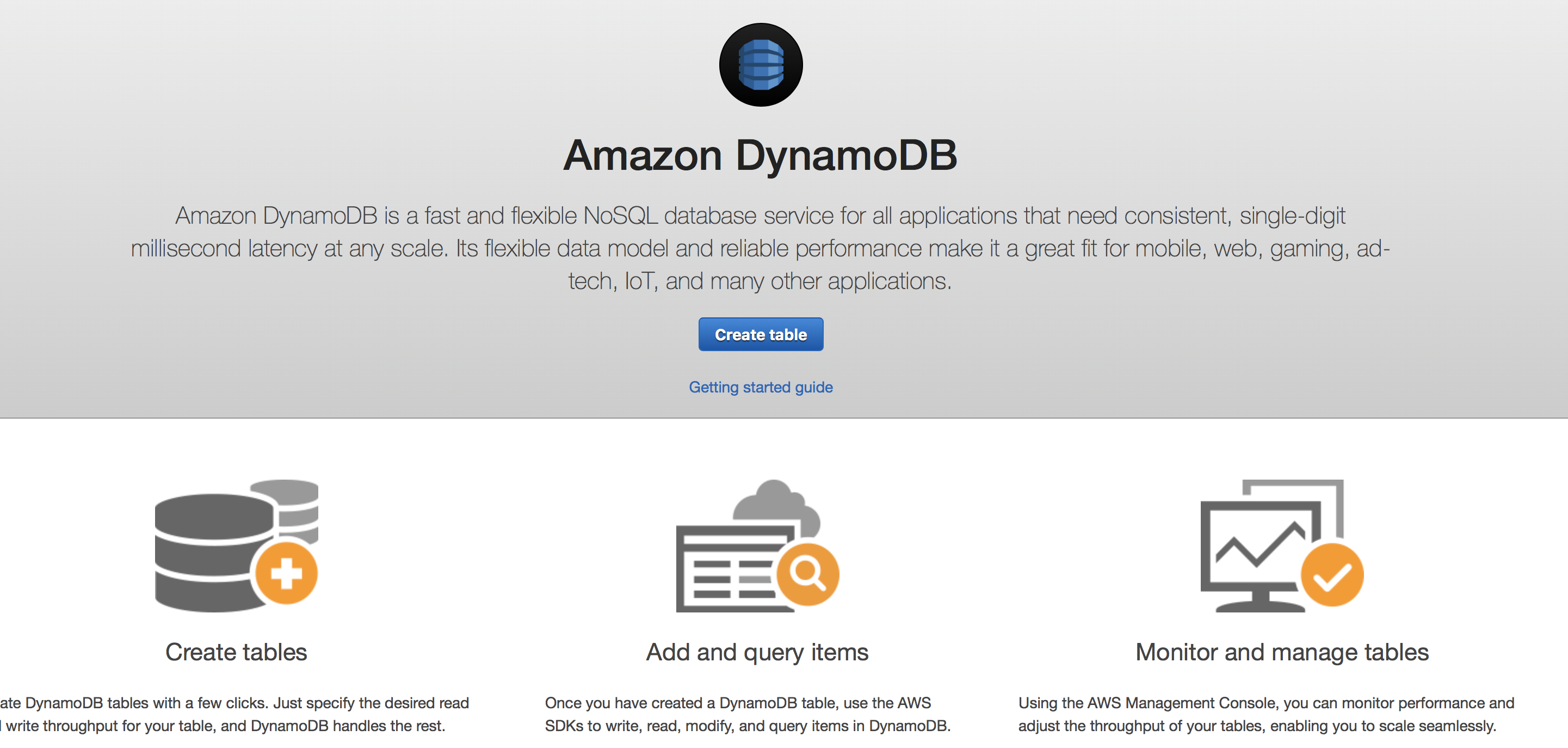
**Interacting with DynamoDB via AWS Console**

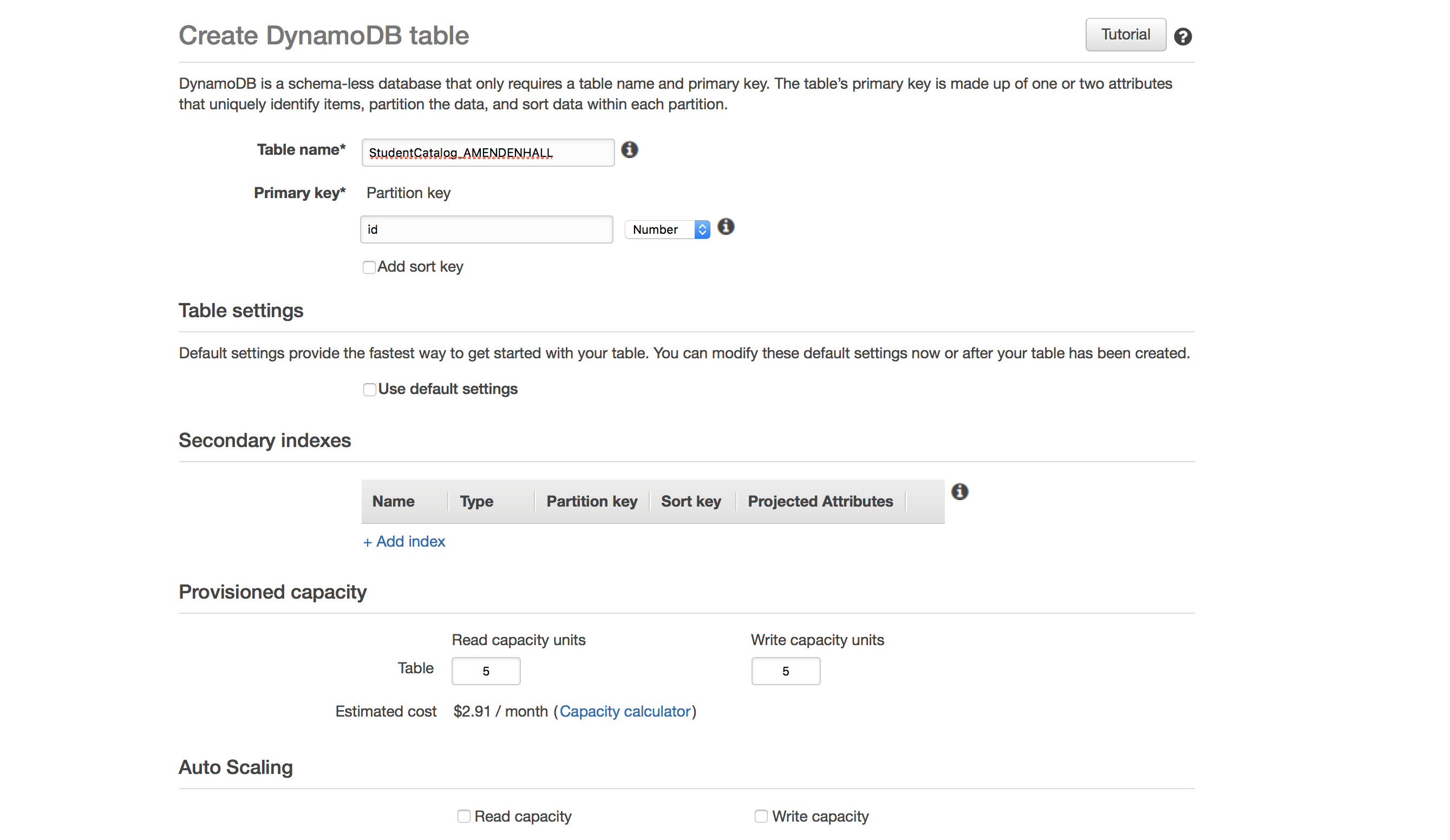
**Exercise 1**

**Learning Objective:**

1. Create table in DynamoDB;
2. Add Primary Keys and Attributes
3. Demonstrate that there is no enforced typing
4. Adding Simple & Complex types to items
5. Delete items from table
6. Create table **StudentCatalog**
   1. Navigate to AWS 🡪 Click on DynamoDB 🡪 Create Table
   2. Table name - StudentCatalog**YOURNETID**
   3. Primary Key – Id (Number)
   4. Uncheck **Use default settings**
   5. Uncheck **Auto Scaling** (Read and write capacity units)
   6. Click **Create table**

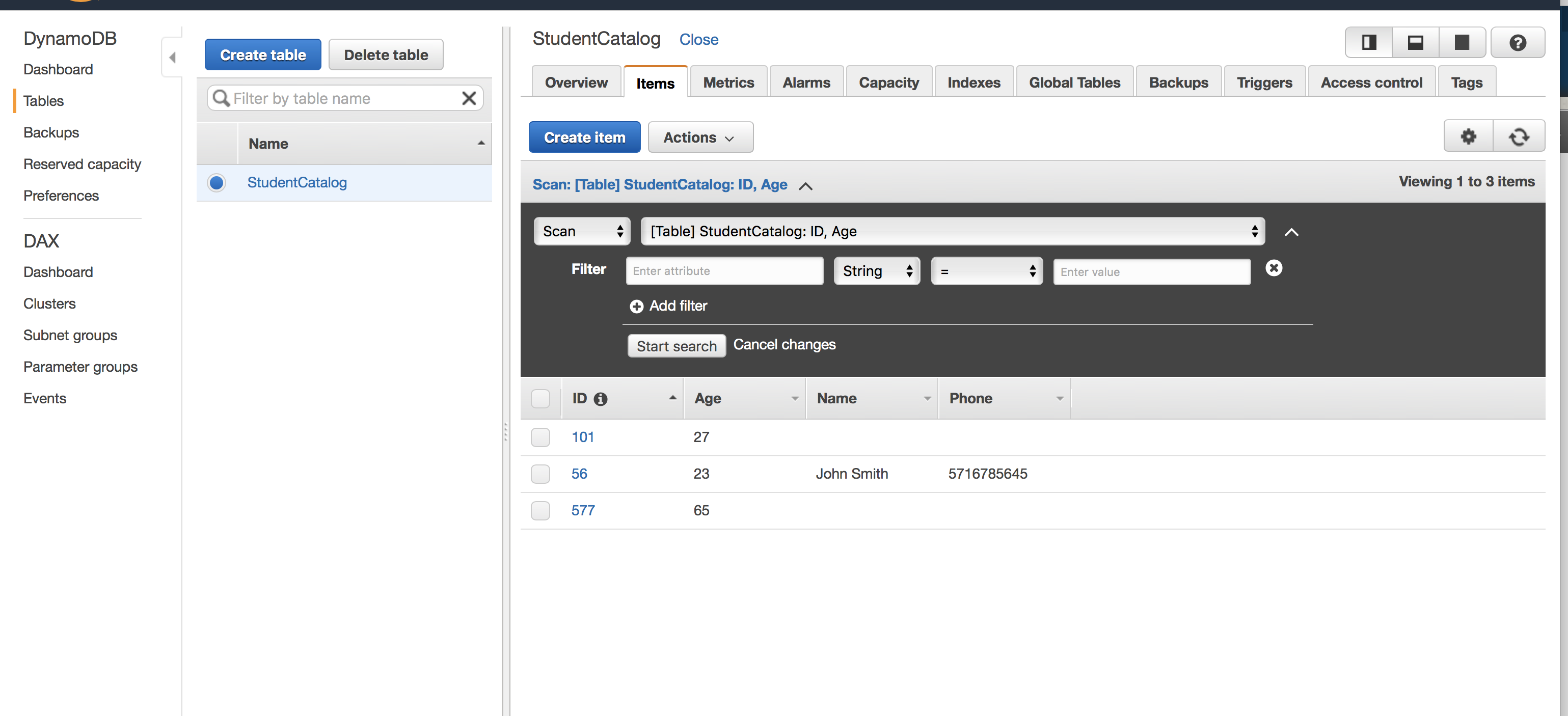


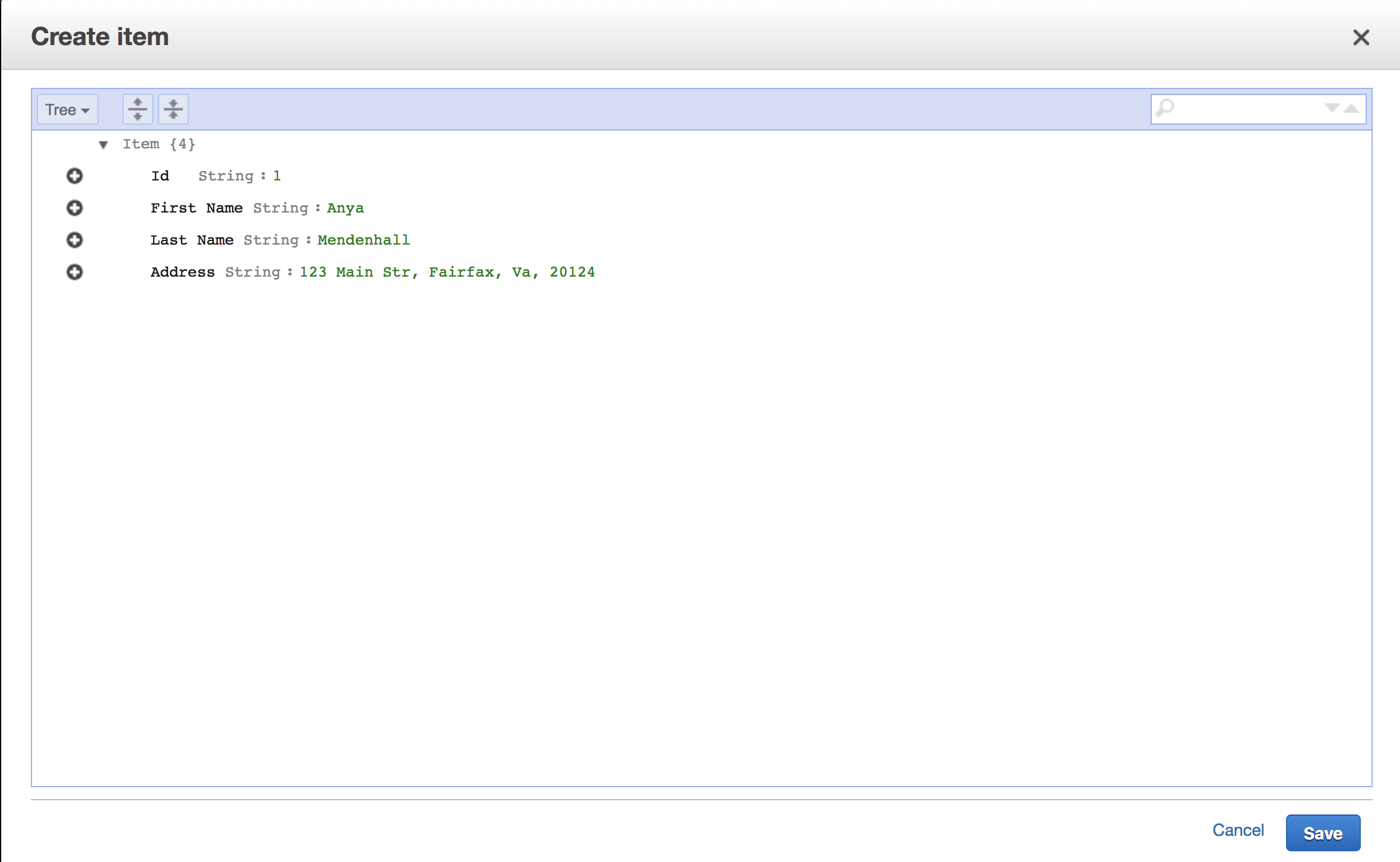




**Task:** Adding items with different attributes to **StudentCatalog** table

1. Adding Items to **StudentCatalog** table
   1. Navigate to Tables 🡪 Chose your table
   2. Items 🡪 Create Item
   3. Id 🡪 Enter 1
   4. Plus Sign 🡪 Append 🡪 First Name (String) 🡪 Enter some name
   5. Plus Sign 🡪 Append 🡪 Last Name (String) 🡪 Enter some name
   6. Plus Sign 🡪 Append 🡪 Address (String) 🡪 Enter some address
   7. Plus Sign 🡪 Append 🡪 Gwid (Number) 🡪 23
   8. Click **Save**

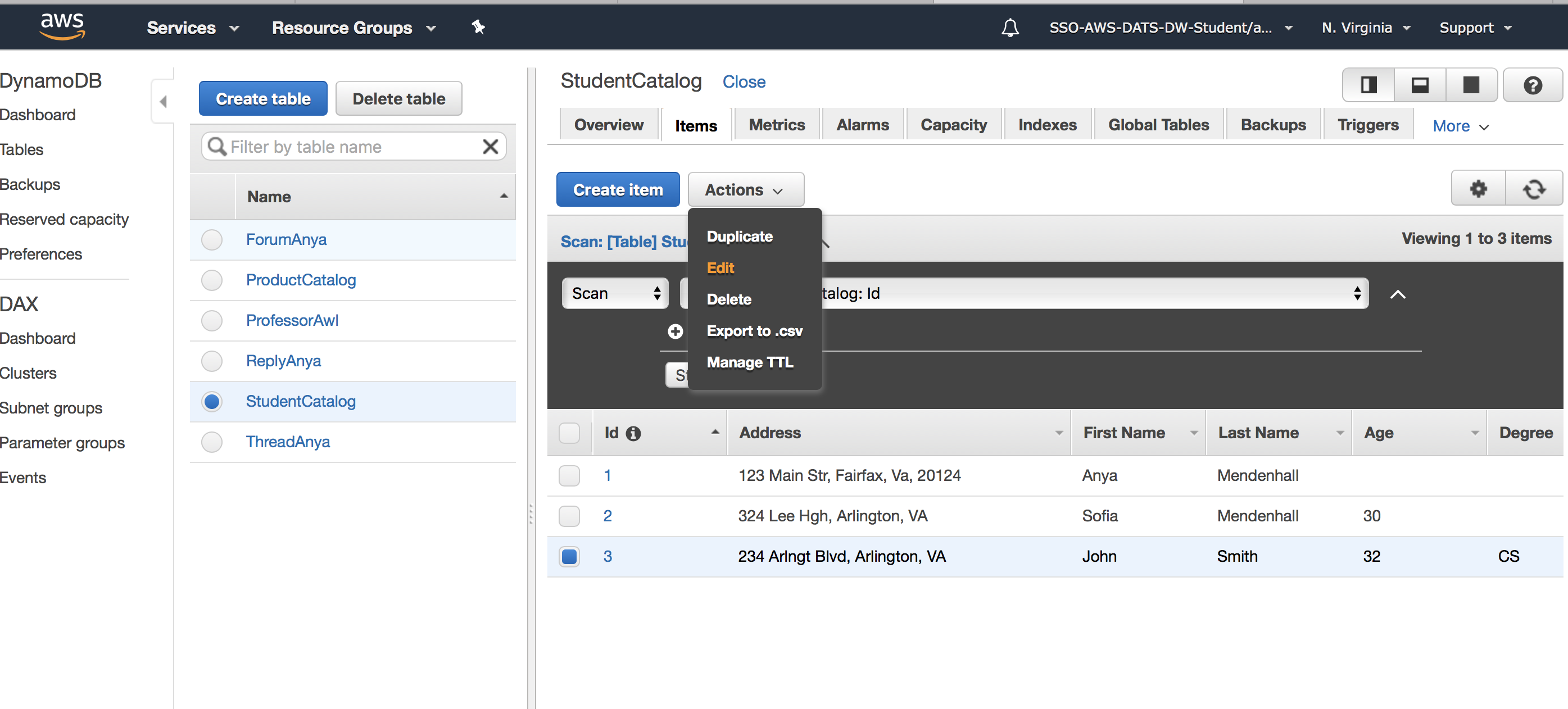


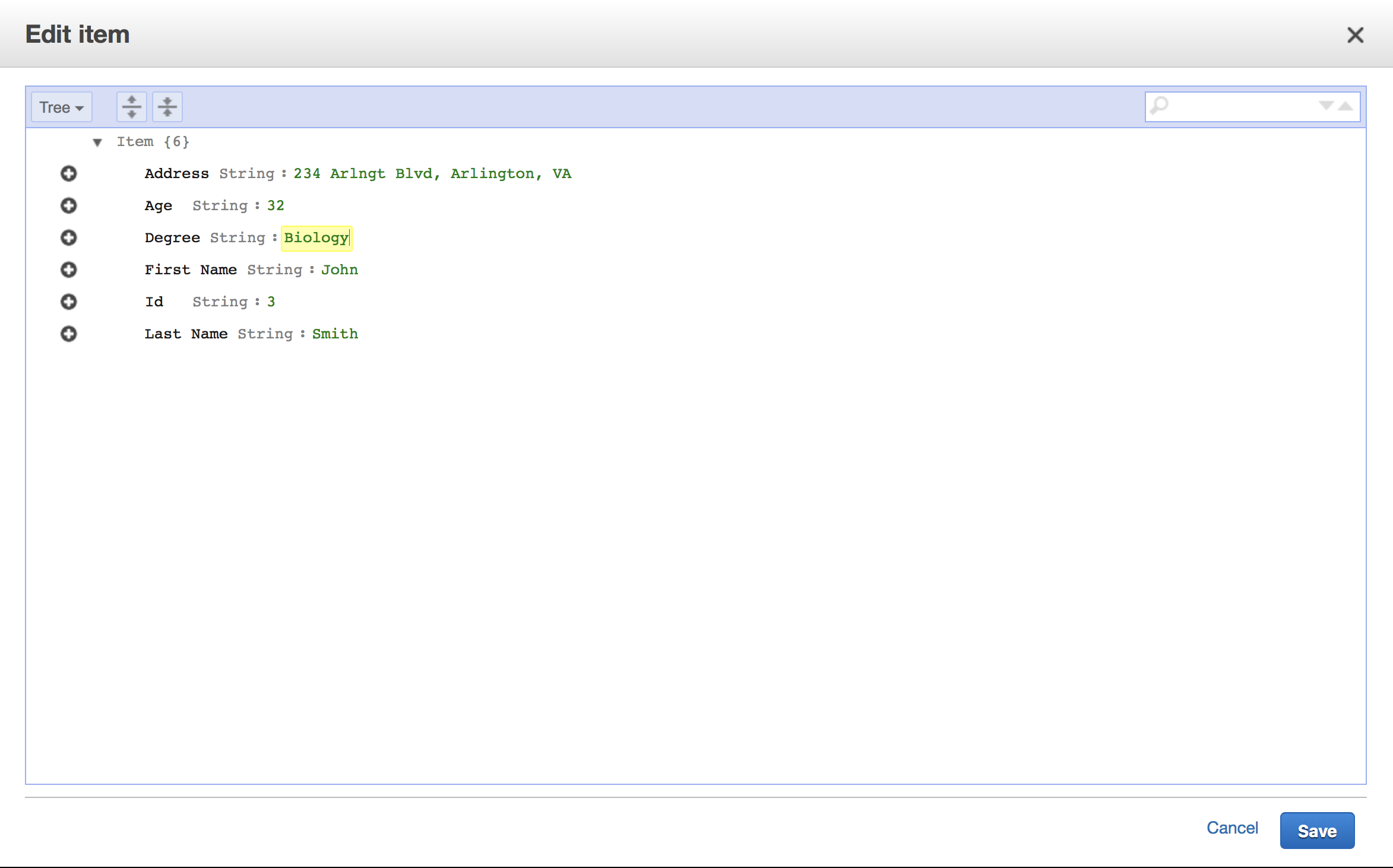


1. Adding Items to **StudentCatalog** table
   1. Navigate to Tables 🡪 Chose your table
   2. Items 🡪 Create Item
   3. Id 🡪 Enter **2**
   4. Plus Sign 🡪 Append 🡪 First Name (String) 🡪 Enter some name
   5. Plus Sign 🡪 Append 🡪 Last Name (String) 🡪 Enter some name
   6. Plus Sign 🡪 Append 🡪 Address (String) 🡪 Enter some address
   7. Plus Sign 🡪 Append 🡪 Age (Number) 🡪 Enter some age
   8. Plus Sign 🡪 Append 🡪 Gwid (Number) 🡪 67
   9. Click **Save**
2. Adding Items to **StudentCatalog** table
   1. Navigate to Tables 🡪 Chose your table
   2. Items 🡪 Create Item
   3. Id 🡪 Enter **3**
   4. Plus Sign 🡪 Append 🡪 First Name (String) 🡪 Enter some name
   5. Plus Sign 🡪 Append 🡪 Last Name (String) 🡪 Enter some name
   6. Plus Sign 🡪 Append 🡪 Address (String) 🡪 Enter some address
   7. Plus Sign 🡪 Append 🡪 Age (Number) 🡪 Enter some age
   8. Plus Sign 🡪 Append 🡪 Degree (String) 🡪 Some degree
   9. Plus Sign 🡪 Append 🡪 Gwid (Number) 🡪 90
   10. Click **Save**

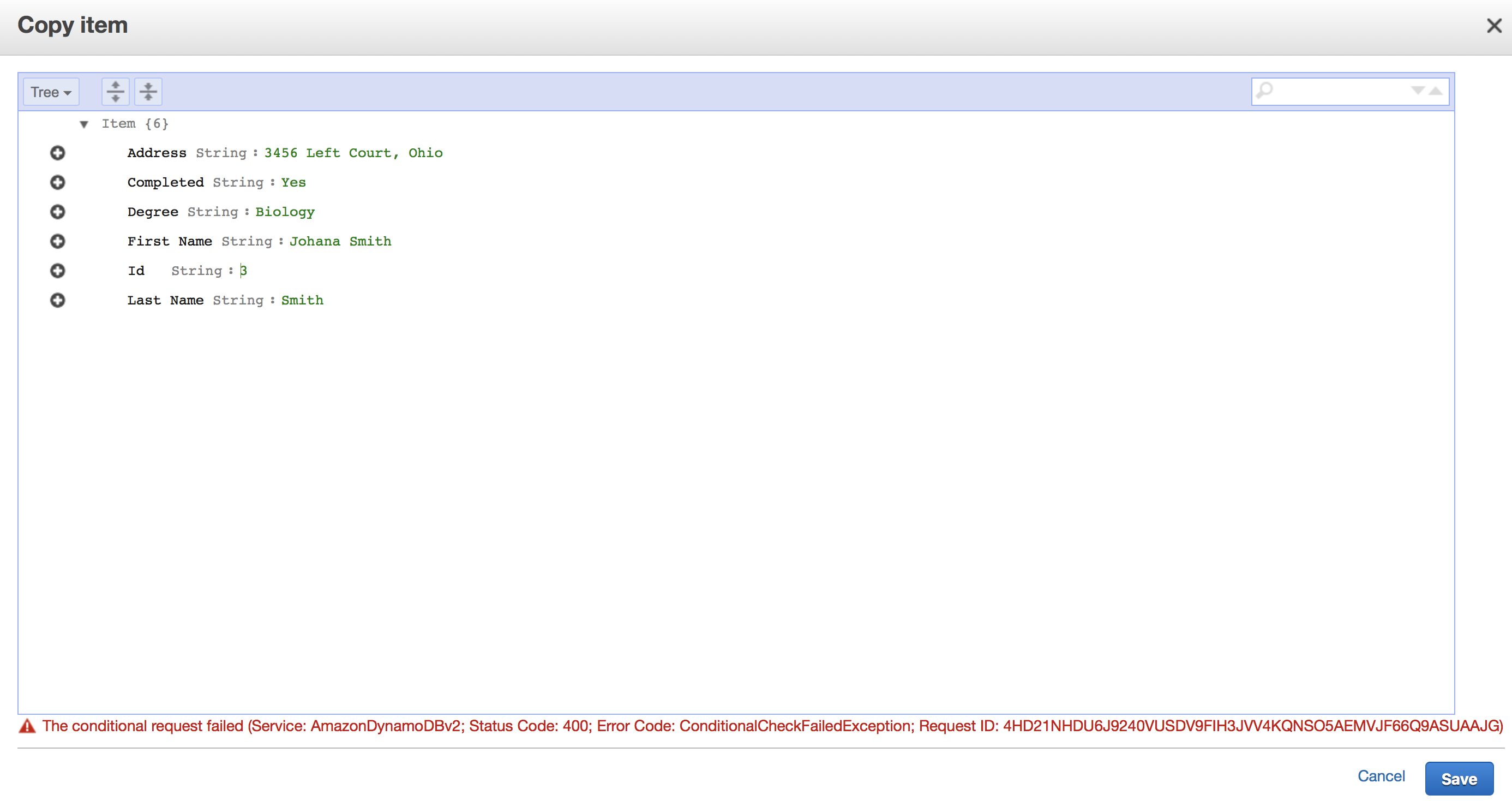


1. Updating Items in **StudentCatalog** table
   1. Navigate to Tables 🡪 Chose your table
   2. Items Choose any item🡪 Actions 🡪 Edit
   3. Change Degree to Biology
   4. Click **Save**

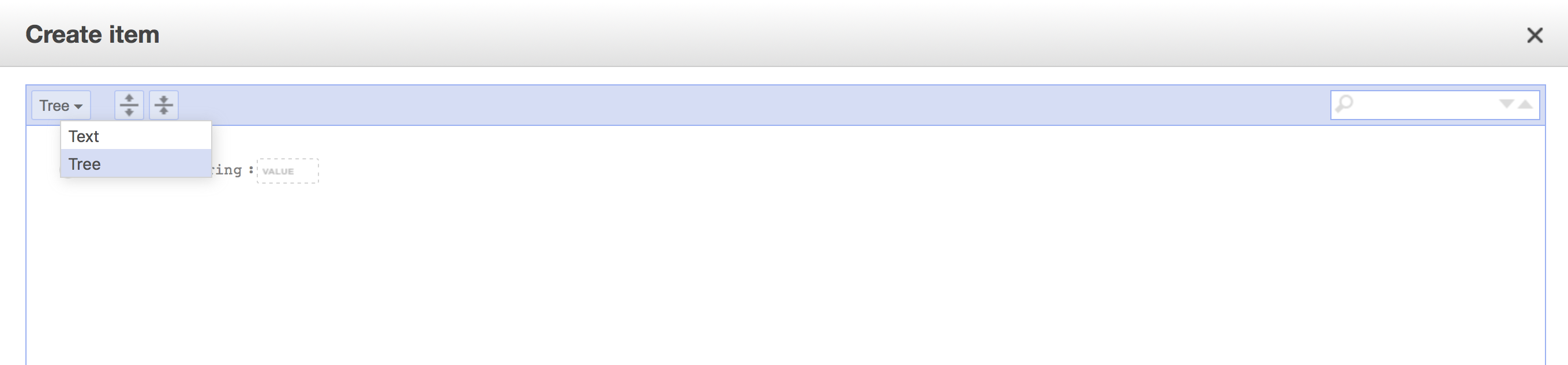


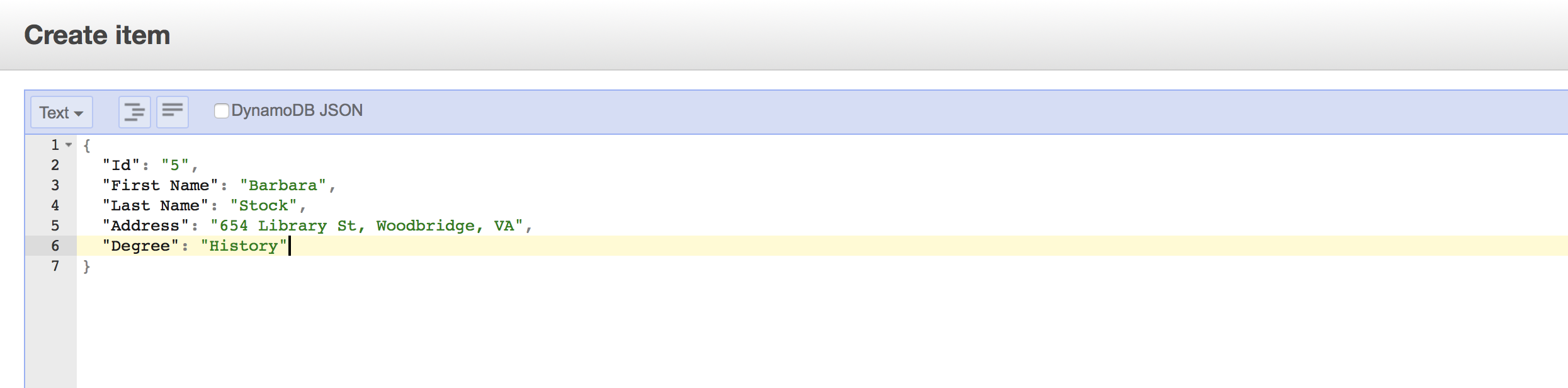


1. Append vs Insert Items in **StudentCatalog** table (observe the difference)
   1. Navigate to Tables 🡪 Chose your table
   2. Items 🡪 Choose any item 🡪 Actions 🡪 Edit
   3. Plus Sign 🡪 Insert 🡪 Finished (Boolean) 🡪 True
   4. Click **Save**
2. Insert item with numeric attribute into **StudentCatalog** table
   1. Navigate to Tables 🡪 Chose your table
   2. Items 🡪 Choose any item 🡪 Actions 🡪 Edit
   3. Plus Sign 🡪 Insert 🡪 Finished (Boolean) 🡪 True
   4. Click **Save**
3. Duplicating Items in **StudentCatalog** table
   1. Navigate to Tables 🡪 Chose your table
   2. Items 🡪 Choose any item 🡪 Actions 🡪 Duplicate
   3. Change Id to 4 (if you do not provide unique to this table Id, you will get an error)
   4. Click **Save**



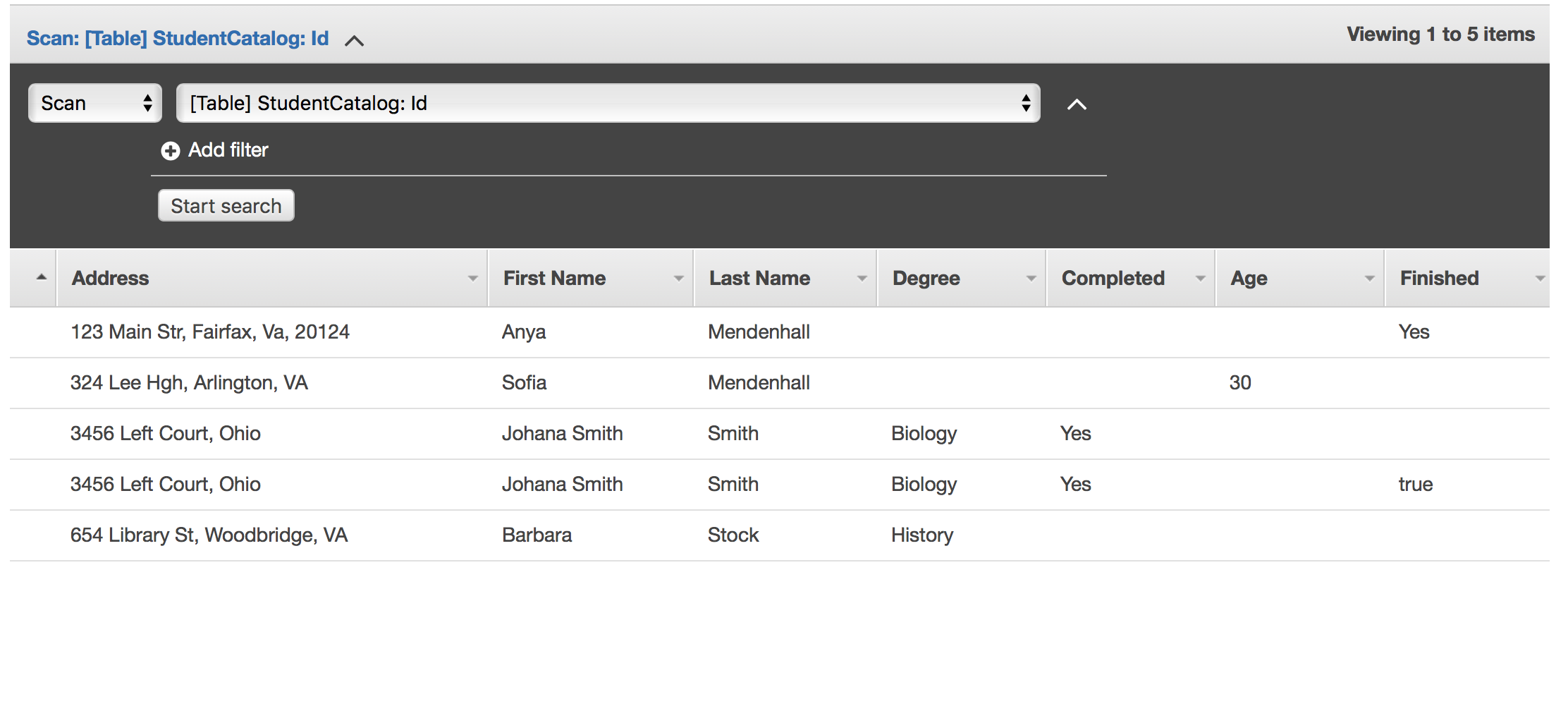
1. Adding items to **StudentCatalog** table in **JSON** format
   1. Navigate to Tables 🡪 Chose your table
   2. Items Choose any item🡪 Actions 🡪 Create Item
   3. From a drop down menu choose Text instead of Tree
   4. Enter attributes in JSON format (make sure to give unique ID)





1. Demonstrating that there isn’t enforced typing for attributes:
   1. Navigate to Tables 🡪 Chose your table
   2. Items Item with Id = 1 🡪 Actions 🡪 Edit
   3. Append 🡪 Finished (String) and set it to Yes
   4. Click Save
   5. Note that Finished is Boolean and String for different items. RDBMS would not allow this, as we cannot mix types in relational databases. So these checks (enforced typing) will not be in place when items are inserted.

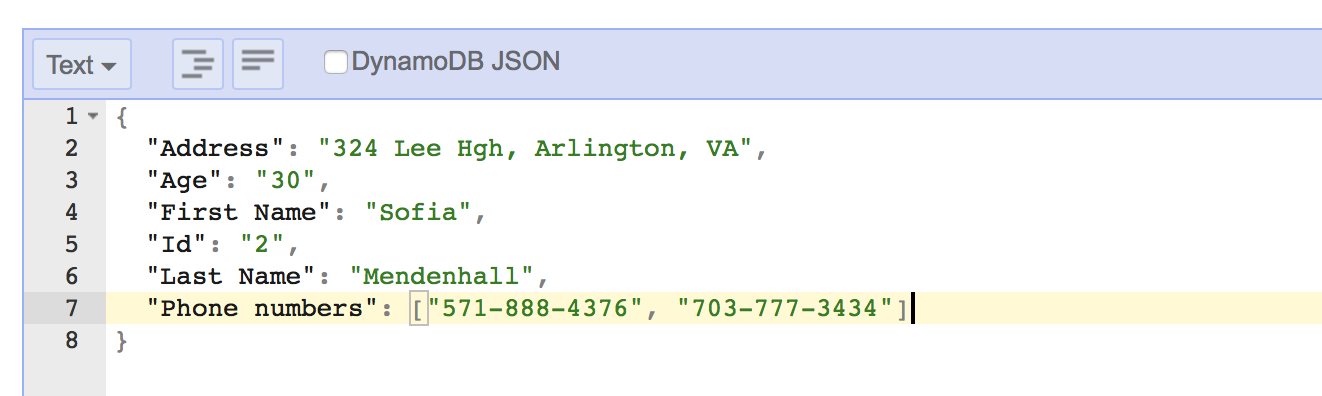




1. Adding Complex Types (**Map type**) to Items. Attributes can have attributes themselves, so we can have nested representation of data.
   1. Navigate to Tables 🡪 Chose your table
   2. Items Choose item with Id 3🡪 Actions 🡪 Edit
   3. Drop down to Text view (JSON)
   4. Enter the following JSON and click Save:



1. Adding Complex Types (**List type**) to Items.
   1. Items Choose item with Id 2🡪 Actions 🡪 Edit
   2. Drop down to Text view (JSON)
   3. Enter the following JSON and click Save:



**Note:**

**{} – represents Map**

**[] – represents List**

1. Deleting Items in **StudentCatalog** table
   1. Navigate to Tables 🡪 Chose your table
   2. Items Choose any item🡪 Actions 🡪 Delete
   3. Delete Item with Id 3

**Exercise 2**

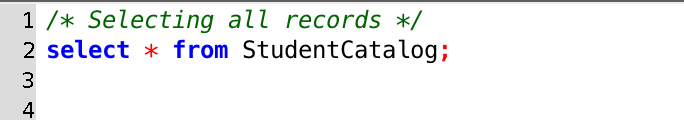
**Learning Objective:** Install RazorSQL

**Task:** Follow instructions posted on Blackboard to download and configure RazorSQL (Mac and Windows instructions are separate). **You have to use authorization and security key that was provided to you via email.**

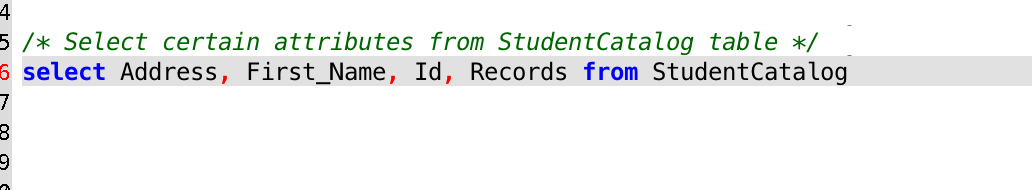
**Exercise 3**

**Learning Objective:** Verify that RazorSQL is connected to your AWS and run basic queries against it.

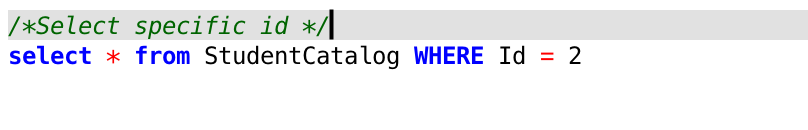
1. Select all records from StudentCatalog table



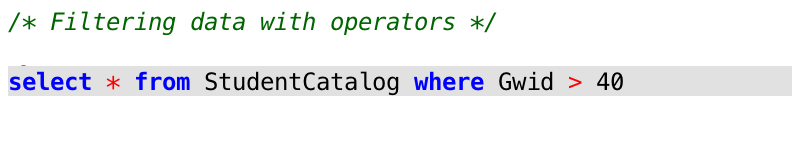
1. Select certain attributes from StudentCatalog table



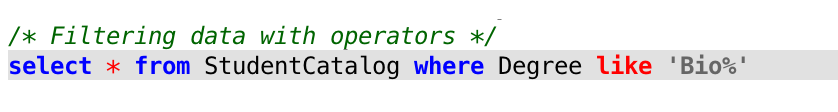
1. Select specific Id



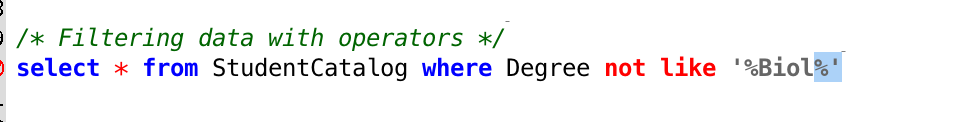
1. Filtering data with operators >



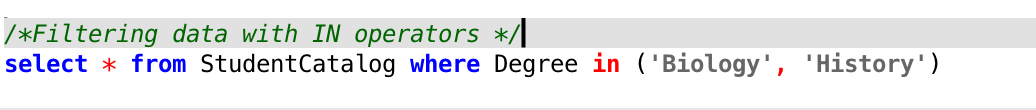
1. Filtering with operators LIKE



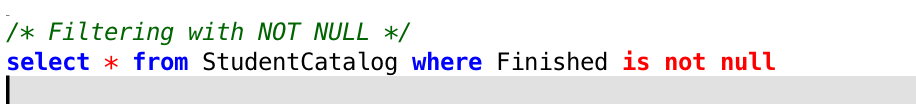
1. Filtering with operators NOT LIKE



1. Filtering data with IN operators



1. Filtering with NOT NULL



1. Filtering records with AND, there is no OR in RazorSQL

