In this tutorial, you do the following:

Create both Dynamodb and RDS.

[What is Dynamodb?](#_95laxyfjm05z)

[What is RDS?](#_xfu2zgvab4h0)

[When should I choose a NoSQL Database vs SQL Database?](#_f5n4ghi24kq9)

[Creating Dynamodb](#_3y74q8txiwlk)

[Connecting EC2 and RDS](#_yhrlxjgqo1ed)

[Creating Dynamodb from CLI](#_1o7ef24eiqne)

### What is Dynamodb?

Amazon DynamoDB is a **fully managed NoSQL database service** that provides fast and predictable performance with seamless scalability. It is a document-oriented database that allows you to store and retrieve data using simple API calls, making it easy to integrate into your applications.

DynamoDB is designed to **handle large amounts of data**, making it an ideal choice for applications that **require high performance, low latency, and scalable storage**.

DynamoDB offers a number of features that make it easy to work with, including automatic scaling, backup and restore, data encryption, and global tables for multi-region deployments. It also supports a flexible data model, allowing you to store and query structured, semi-structured, and unstructured data.

### What is RDS?

Amazon Relational Database Service (RDS) is a **fully managed database service** provided by Amazon Web Services (AWS). It makes it **easy to set up, operate, and scale a relational database** in the cloud. With RDS, you can choose from several popular database engines such as **Amazon Aurora, MySQL, PostgreSQL, MariaDB, Oracle, and Microsoft SQL Server**, and run your database in a fully managed environment.

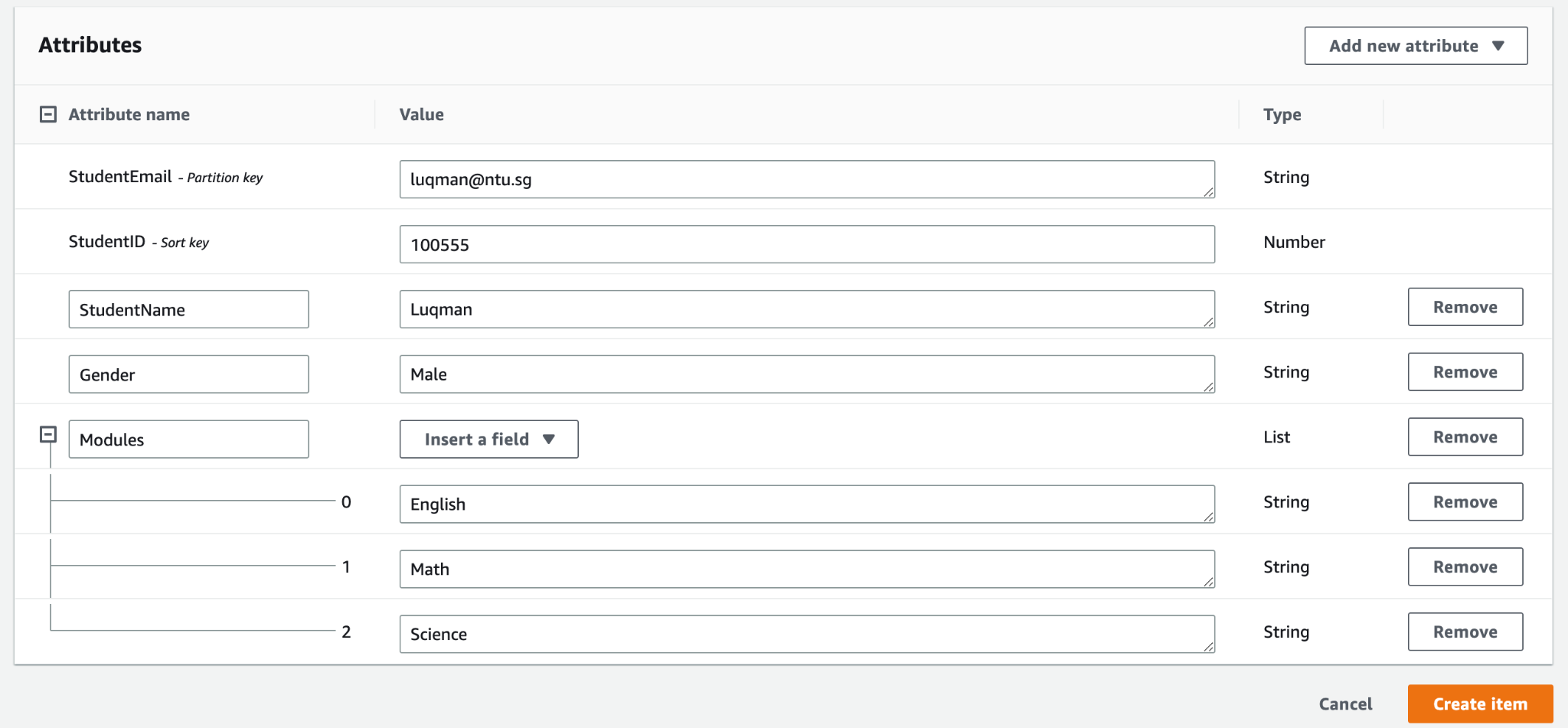
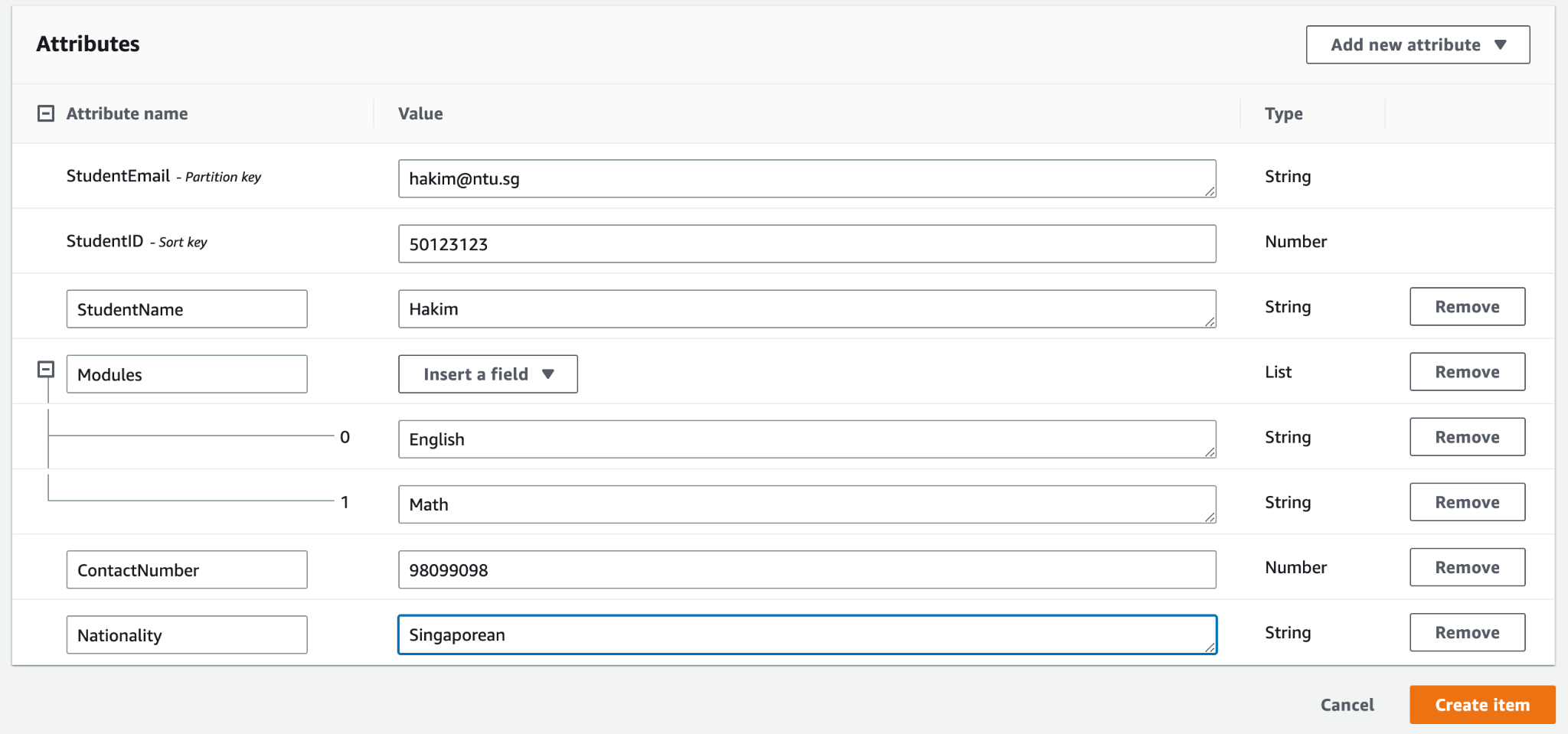
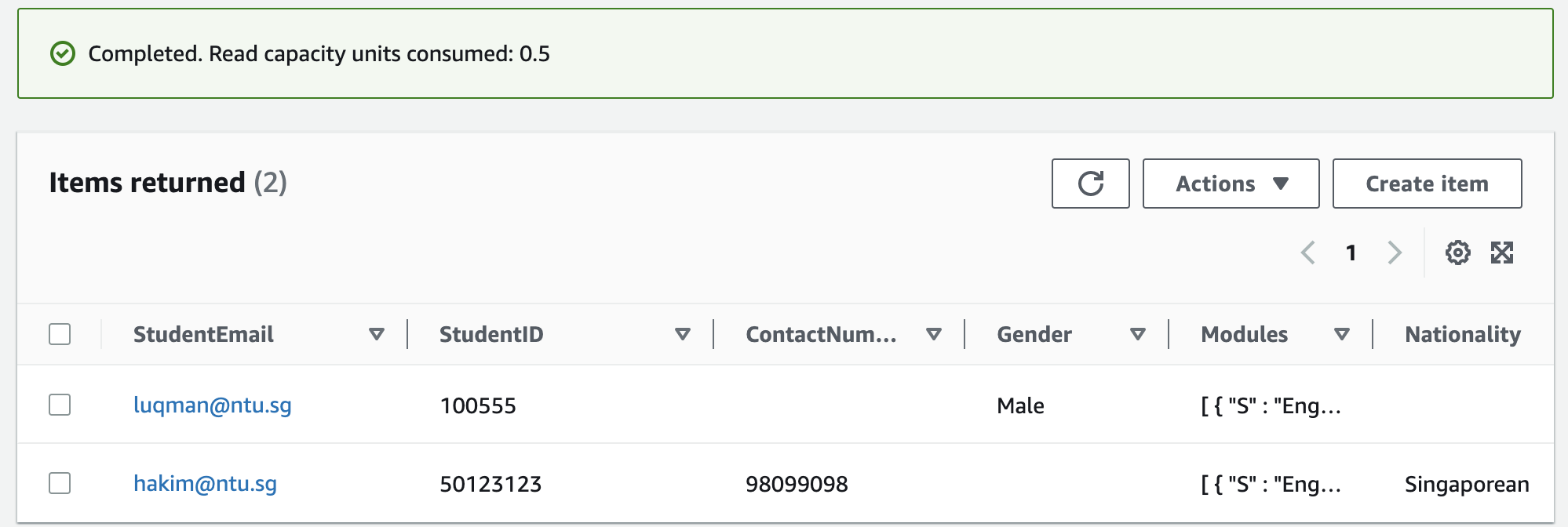
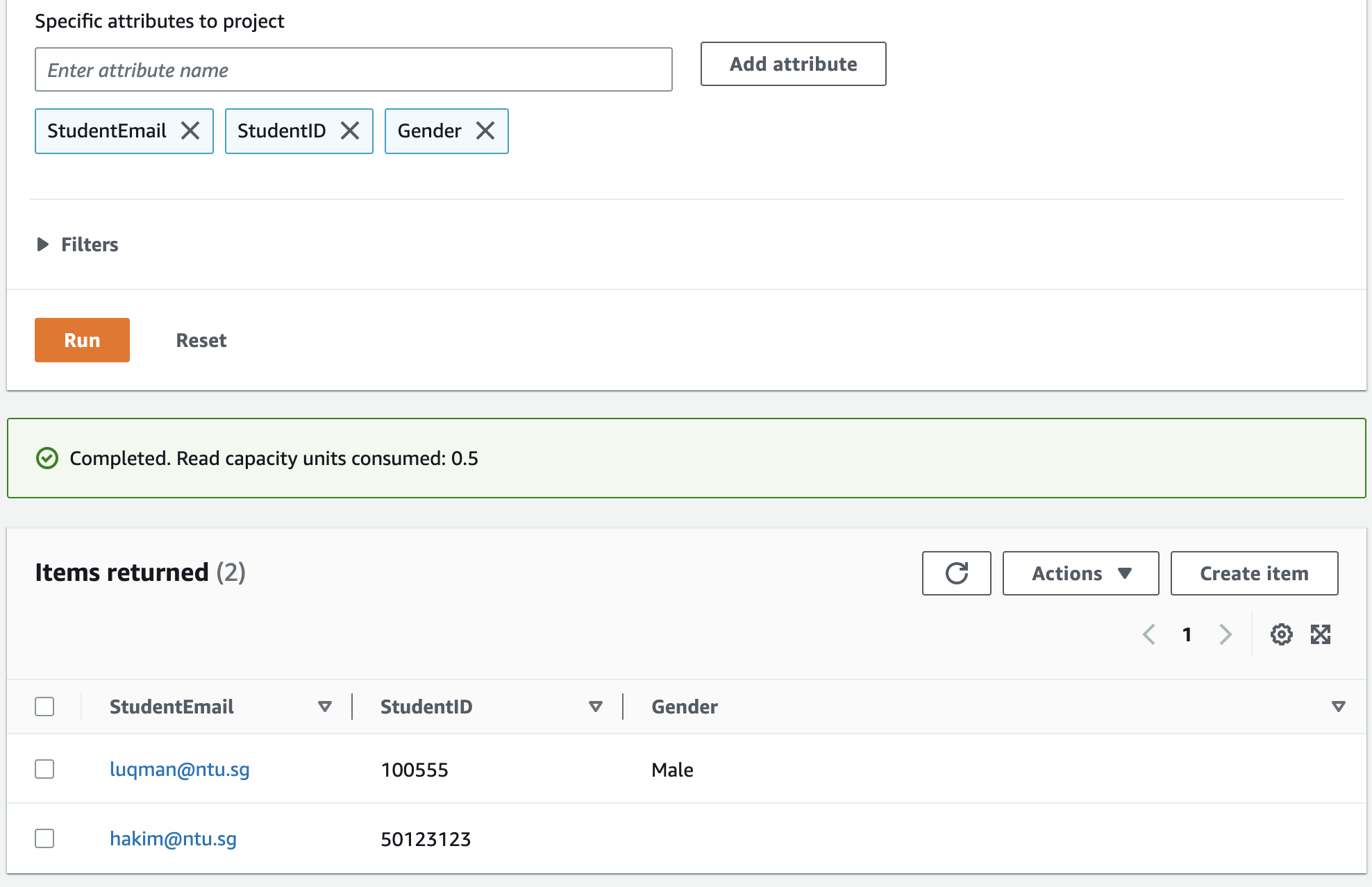
RDS **takes care of database administration tasks such as software installation, patching, backups, and monitoring**. It also provides **automated scaling**, enabling you to increase or decrease database capacity on-demand, **without downtime**. This makes RDS an ideal choice for applications that require **high availability, reliability, and scalability**.

RDS supports various features such as multi-availability zone deployments, read replicas, and automatic backups, which help you build fault-tolerant and highly available applications. It also provides security features such as network isolation, encryption at rest and in-transit, and authentication and authorization mechanisms to help you secure your data.

### When should I choose a NoSQL Database vs SQL Database?

* **Data model**: DynamoDB is a NoSQL database, which means it is designed to handle unstructured data and provides flexible data modeling capabilities. RDS, on the other hand, is a relational database that uses SQL and is best suited for structured data.
* **Performance requirements**: DynamoDB is optimized for high-speed, low-latency access to data, making it an excellent choice for applications that require quick, predictable performance at scale. RDS can also provide high performance, but it may require more resources to achieve the same level of performance as DynamoDB.
* **Scaling requirements**: DynamoDB is designed to scale horizontally by adding more capacity as needed, while RDS is designed to scale vertically by increasing the size of the database instance. If you anticipate a high volume of traffic and rapid growth, DynamoDB may be a better choice.
* **Cost considerations**: DynamoDB is generally more cost-effective than RDS for applications that require high write throughput or have unpredictable traffic patterns. RDS can be more cost-effective for applications with consistent traffic patterns or where SQL is a requirement.
* **Database management**: DynamoDB is a fully managed service, which means AWS takes care of all the administrative tasks such as software updates, backups, and scaling. RDS is also fully managed, but you have more control over the database management tasks.

### Creating Dynamodb

* 1. For this activity, you will be creating a Dynamodb table for a school to manage its student list.
  2. From the AWS console, go to Dynamodb and click create table.
  3. Enter a unique table name you’d like e.g. Group2StudentTable
  4. Partition Keys are also the primary key of the database, and a good partition key would be one that is unique like NRICs, Student IDs, emails etc. Read more here - <https://aws.amazon.com/blogs/database/choosing-the-right-dynamodb-partition-key/>   
     For partition key, you may choose any that you feel is valid, e.g. StudentEmail as a String object.
  5. Sort Keys are optional second primary keys of your database. They gather related information together in one place where it can be queried efficiently - <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-sort-keys.html#:~:text=Well%2Ddesigned%20sort%20keys%20have,%2C%20%3C%20%2C%20and%20so%20on>.   
     For sort key, you may choose one that is also unique e.g. StudentID as a String object.
  6. Create the table with the default settings.
  7. Once created, click into your newly created Dynamodb table.
  8. From here, you would want to start entering new records into the table. To do so, click on “Explore table items”.
  9. At the bottom of the page, you would see “Create item”. Click this to start inserting records into your Dynamodb table.
  10. Fill up all the necessary fields which are the partition key and sort key, and you can click on “Add new attribute” to add new fields as well. Follow the template below for your first record.
  11. 
  12. Create a second entry with the below template as well. Notice that there are different fields in this second record.
  13. Create as many records as you’d like! The more the merrier.
  14. Head back to the table again. Under Explore table items, you will see two options to view your data - Scan or Query.
  15. Scan, as the name suggests, will browse table items from start to finish. A Query will rely on the primary-key to find information. Query can point directly to a particular item (or set of items) and retrieve them in a fast and efficient way.
  16. For now, select “Scan”, select your Table under the drop down and click “Run”. This would be a sample output.
  17. 
  18. You can also scan for specific attributes e.g. StudentEmail, StudentID, Gender. Once done, click “Run” to view the output:
  19. To-Do: Explore what query is and how query differs from Scan.
  20. To-Do: Try doing delete records too.
  21. Once done, please remember to delete your table.

### Connecting EC2 and RDS

* 1. Try it out - <https://medium.com/geekculture/create-a-web-server-and-an-amazon-rds-database-instance-8259a352b790>
  2. If you’d like to try this out, do try to share RDS as a group as much as possible, or use different regions to avoid running into resource limit issues.

### Creating Dynamodb from CLI

* 1. Remember our activity previously where we connected to AWS using CLI? Let’s build on that again.
  2. Follow the same steps above in part 1 but this time, let’s use CLI. The commands can be found in this guide - <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/getting-started-step-1.html> Follow from step 1 to step 8 for the AWS CLI option.