# AZURE SCENARIO 5:

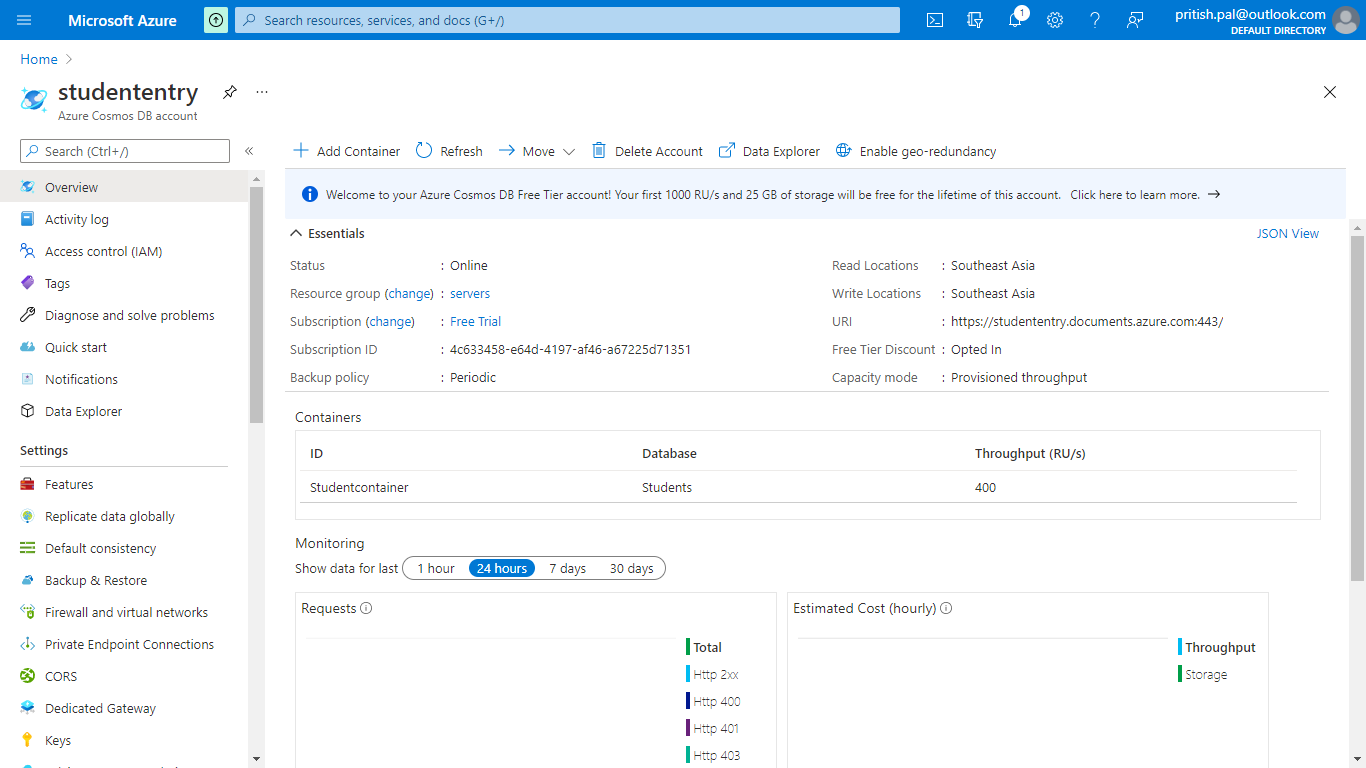
A screenshot of a computer

Description automatically generated with medium confidence

**As per the scenario, student registers himself and his marks on the application and the details are stored in Azure Cosmos DB, the details are thereby processed by Azure logic app which processes records and changes status to “Pass” or “Fail” based on student percentage>60**

## **STEPS:**

* Created Cosmos db named “**studententry”**.



* Created database “**Students”** and container “**StudentContainer”, “MarksContainer”** via code. Also created a web application to register student entry via student class and marks entry via marks class.

**CosmosDBConnection class: for connection to cosmos DB**

using Microsoft.Azure.Cosmos;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Web;

namespace StuudentEntry.Models

{

public class CosmosDBConnection

{

private string EndpointUrl = Environment.GetEnvironmentVariable("EndpointUrl");

private string PrimaryKey = Environment.GetEnvironmentVariable("PrimaryKey");

private CosmosClient cosmosClient;

private Database database;

private Container container;

private string databaseId = "Students";

private string containerId = "Studentcontainer";

private async Task CreateDatabaseAsync()

{

this.database = await this.cosmosClient.CreateDatabaseIfNotExistsAsync(databaseId);

}

private async Task CreateContainerAsync()

{

this.container = await this.database.CreateContainerIfNotExistsAsync(containerId, "/id");

}

private async Task AddItemsToContainerAsync(Student s)

{

ItemResponse<Student> studentadd = await this.container.CreateItemAsync<Student>(s, new PartitionKey(s.id));

}

public async Task GetStartedDemoAsync(Student student, String url,string key)

{

this.cosmosClient = new CosmosClient(url, key);

await this.CreateDatabaseAsync();

await this.CreateContainerAsync();

await this.AddItemsToContainerAsync(student);

//await this.QueryItemsAsync();

}

}

}

**Students class:**

using Newtonsoft.Json;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace StuudentEntry.Models

{

public class Student

{

[JsonProperty(PropertyName = "id")]

public string Id { get; set; }

public string Name { get; set; }

public string Status { get; set; }

}

}

**Marks class:**

using Newtonsoft.Json;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace StuudentEntry.Models

{

public class Marks

{

[JsonProperty(PropertyName = "id")]

public string Id { get; set; }

public string marks1 { get; set; }

public string marks2 { get; set; }

public string marks3 { get; set; }

public string percent { get; set; }

}

}

**StudentController class:**

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Threading.Tasks;

using System.Net;

using Microsoft.Azure.Cosmos;

using StuudentEntry.Models;

using Newtonsoft.Json;

namespace StuudentEntry.Controllers

{

public class StudentController : Controller

{

public string EndpointUrl = "https://studententry.documents.azure.com:443/";

/// The primary key for the Azure DocumentDB account.

public string PrimaryKey = "yKZkUFZylIA7Z44PO2eMKj9G54CqdtZjNAFRYSrFk60lEibmB3OhjaNSrwiEtha4OXH0tmq8tWNEBhqeaRoxDw==";

// GET: Student

//public ActionResult Index()

//{

// return View();

//}

[HttpGet]

public ActionResult Register()

{

return View();

}

[HttpPost]

public async Task<string> Register(Student s)

{

//SqlConnection sqlConnection = new SqlConnection();

//string connectionString = ConfigurationManager.ConnectionStrings["conn"].ConnectionString;

//SqlCommand sqlCommand = new SqlCommand();

//sqlConnection.ConnectionString = connectionString;

//sqlCommand.CommandType = CommandType.Text;

//sqlCommand.CommandText = "insert into student values ("+s.Id+','+"'"+s.Name+"'"+","+"'"+s.Status+"'"+")";

//sqlCommand.Connection = sqlConnection;

//sqlConnection.Open();

//sqlCommand.ExecuteNonQuery();

//sqlConnection.Close();

CosmosDBConnection dbconn = new CosmosDBConnection();

await dbconn.GetStartedDemoAsync(s,EndpointUrl,PrimaryKey);

return "Student Registered successfully";

}

[HttpGet]

public ActionResult RegisterMarks()

{

return View();

}

[HttpPost]

public async Task<ActionResult> RegisterMarks(Marks s)

{

double student\_percent;

int average\_marks;

int marks1= (int)Convert.ToInt64(s.marks1);

int marks2= (int)Convert.ToInt64(s.marks2);

int marks3 = (int)Convert.ToInt64(s.marks3);

average\_marks = marks1 + marks2 + marks3;

student\_percent = average\_marks\*100/300;

string finalpercent= Convert.ToString(student\_percent);

s.percent = finalpercent;

CosmosDBConnection dbconn = new CosmosDBConnection();

await dbconn.MarksAdder(EndpointUrl, PrimaryKey, s);

//SqlConnection sqlConnection = new SqlConnection();

//string connectionString = ConfigurationManager.ConnectionStrings["conn"].ConnectionString;

//SqlCommand sqlCommand = new SqlCommand();

//sqlConnection.ConnectionString = connectionString;

//sqlCommand.CommandType = CommandType.Text;

//sqlCommand.CommandText = "insert into marks values (" + s.Id + ',' + student\_percent + ")";

//sqlCommand.Connection = sqlConnection;

//sqlConnection.Open();

//sqlCommand.ExecuteNonQuery();

//sqlConnection.Close();

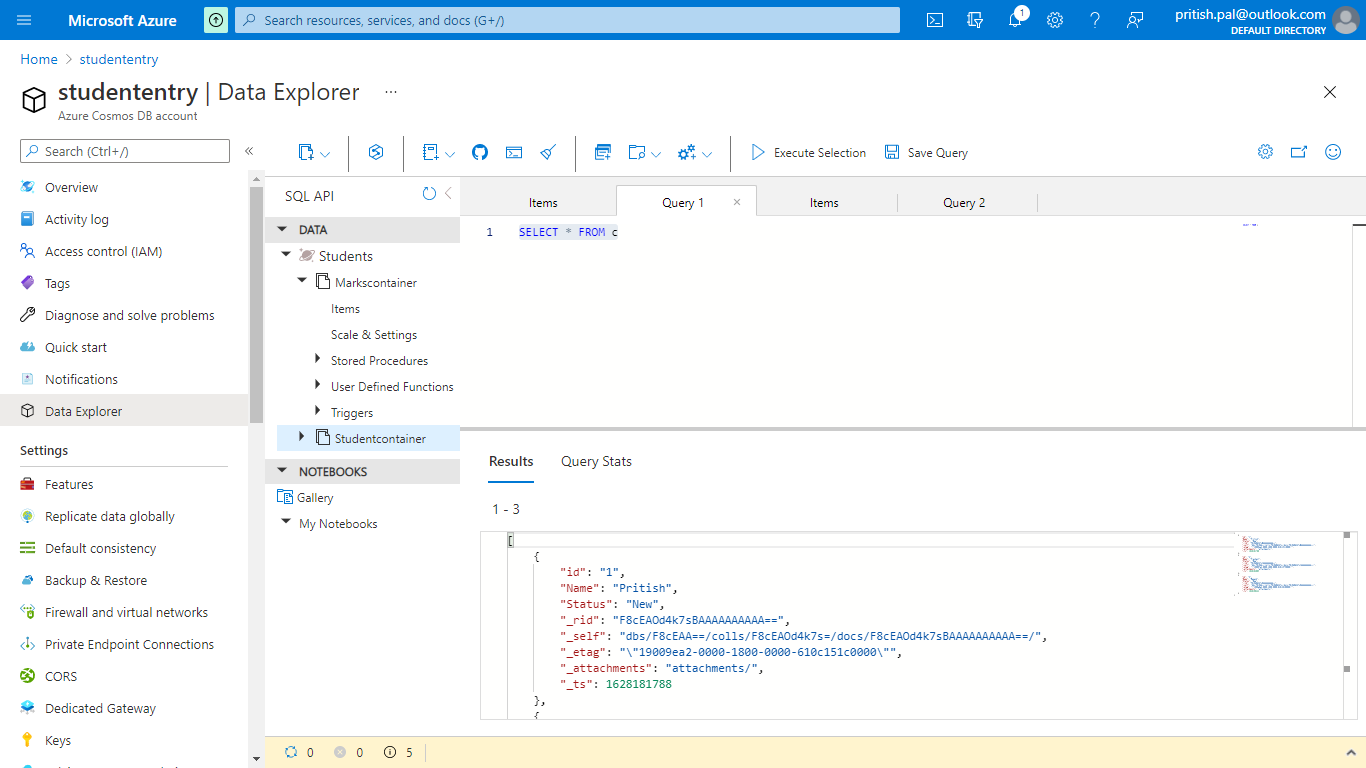
return View();

}

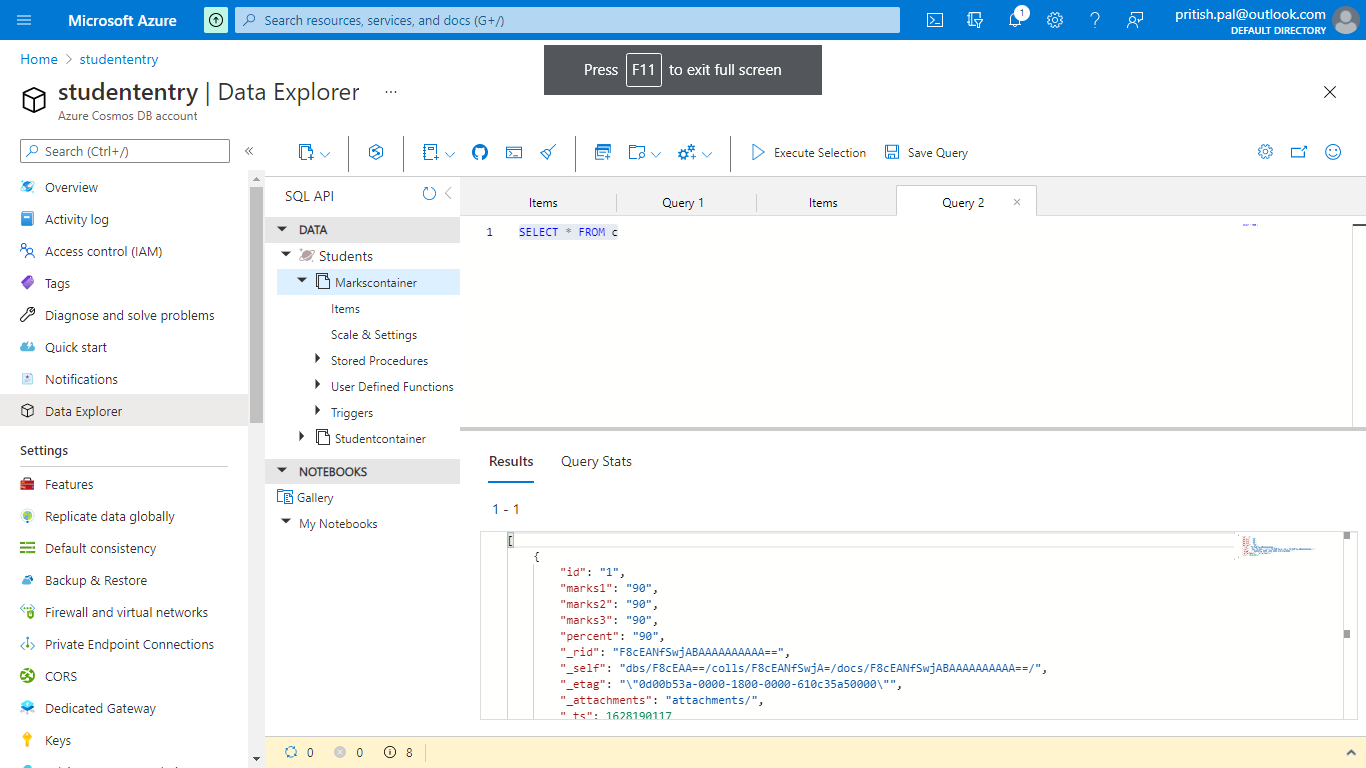
}

}

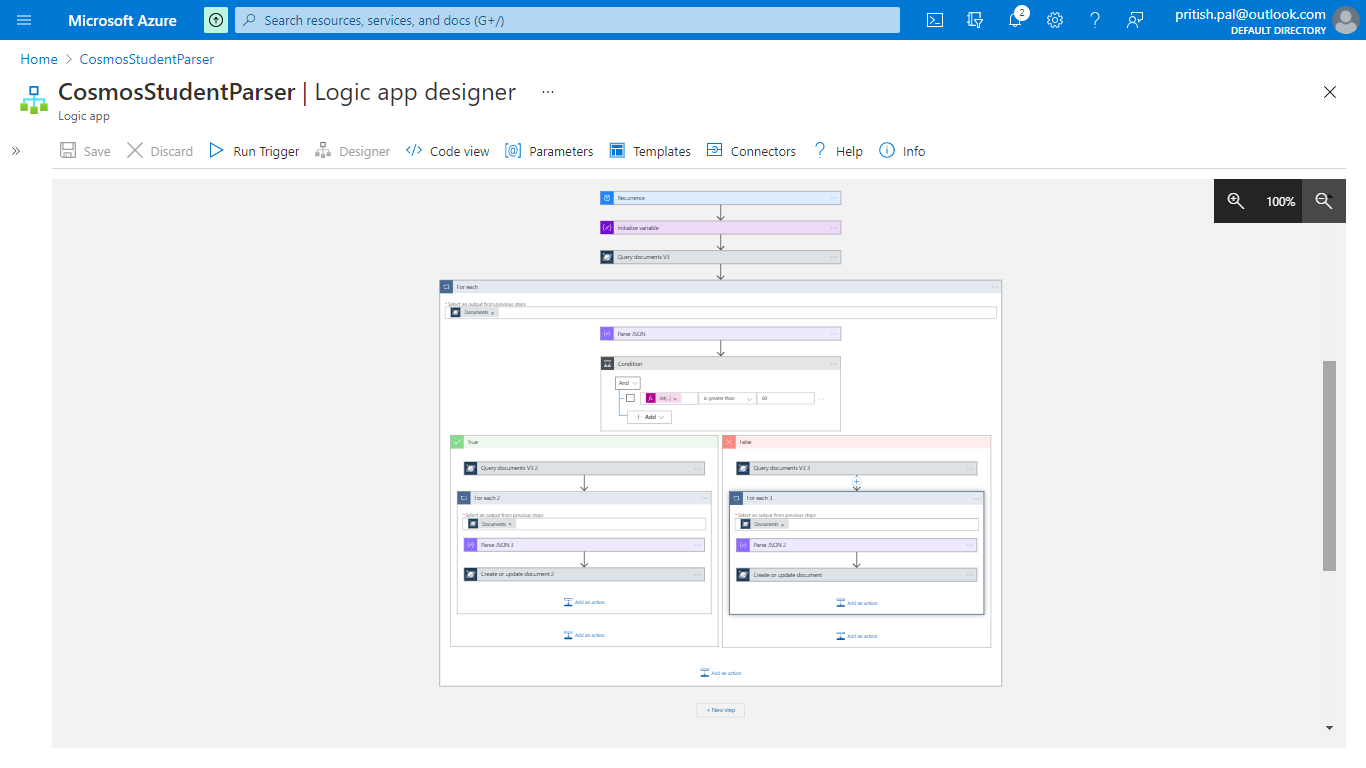
Student data



Marks data:



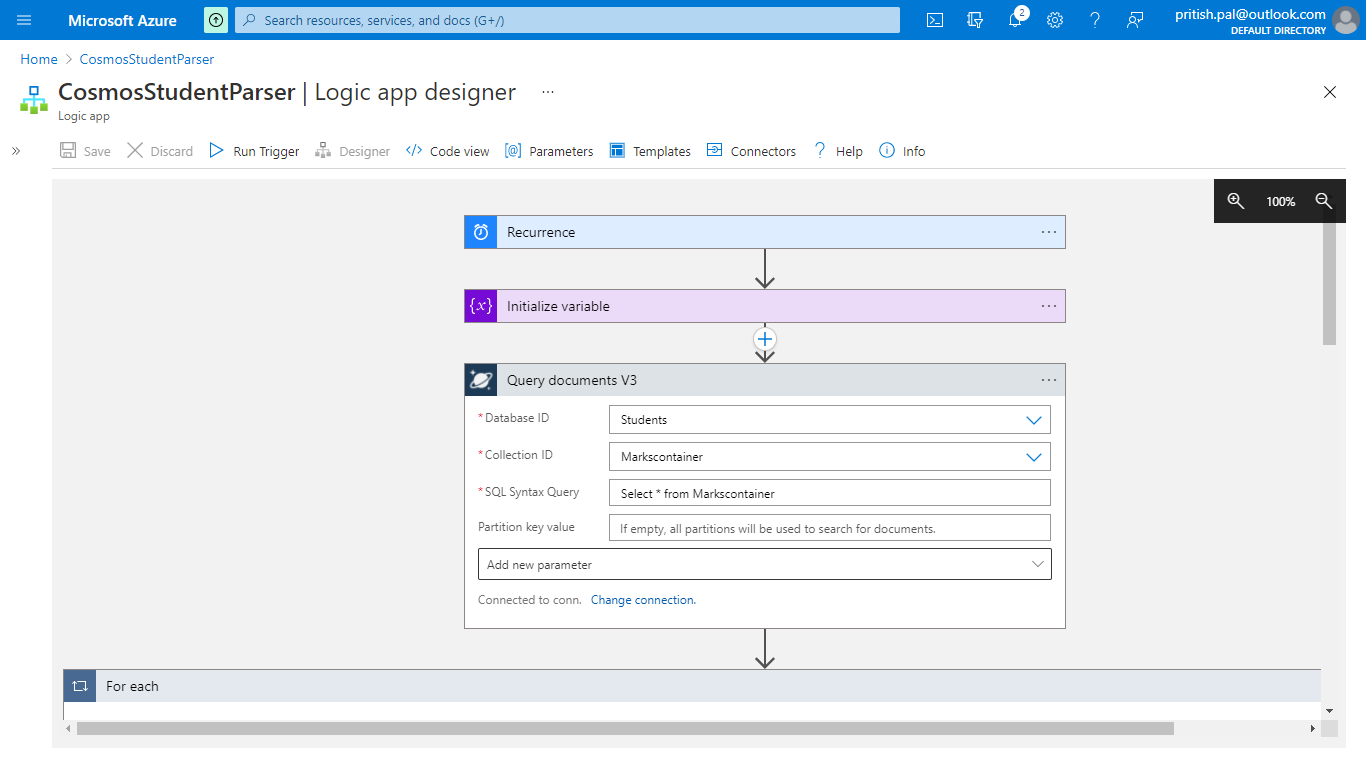
* Designed Logic app which processes student records and changes status to “Pass” or “Fail” based on percentage>60.



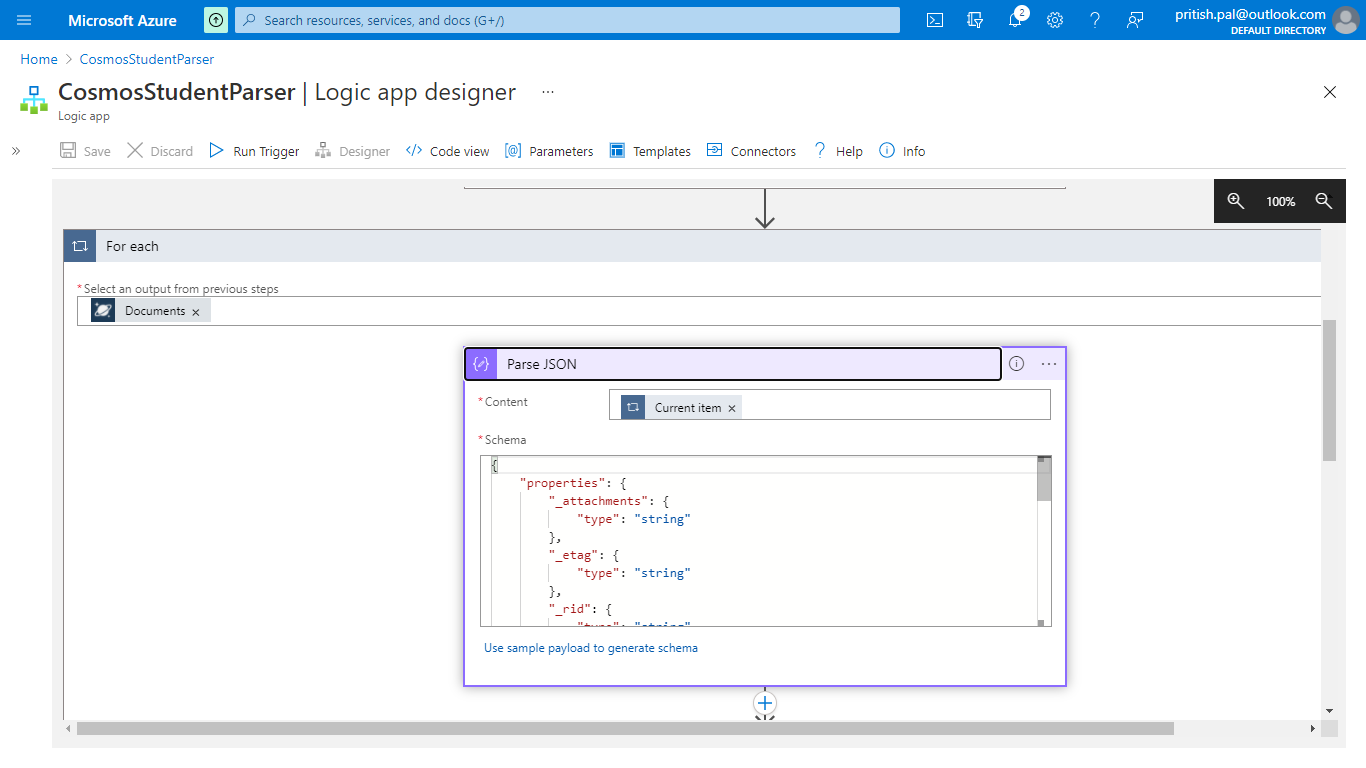
Connectors:

**Query Document DB:** QUERIES THE MENTIONED DATABASE AND CONTAINER AND GETS THE RELATED DOCUMENTS BASED ON THE QUERY

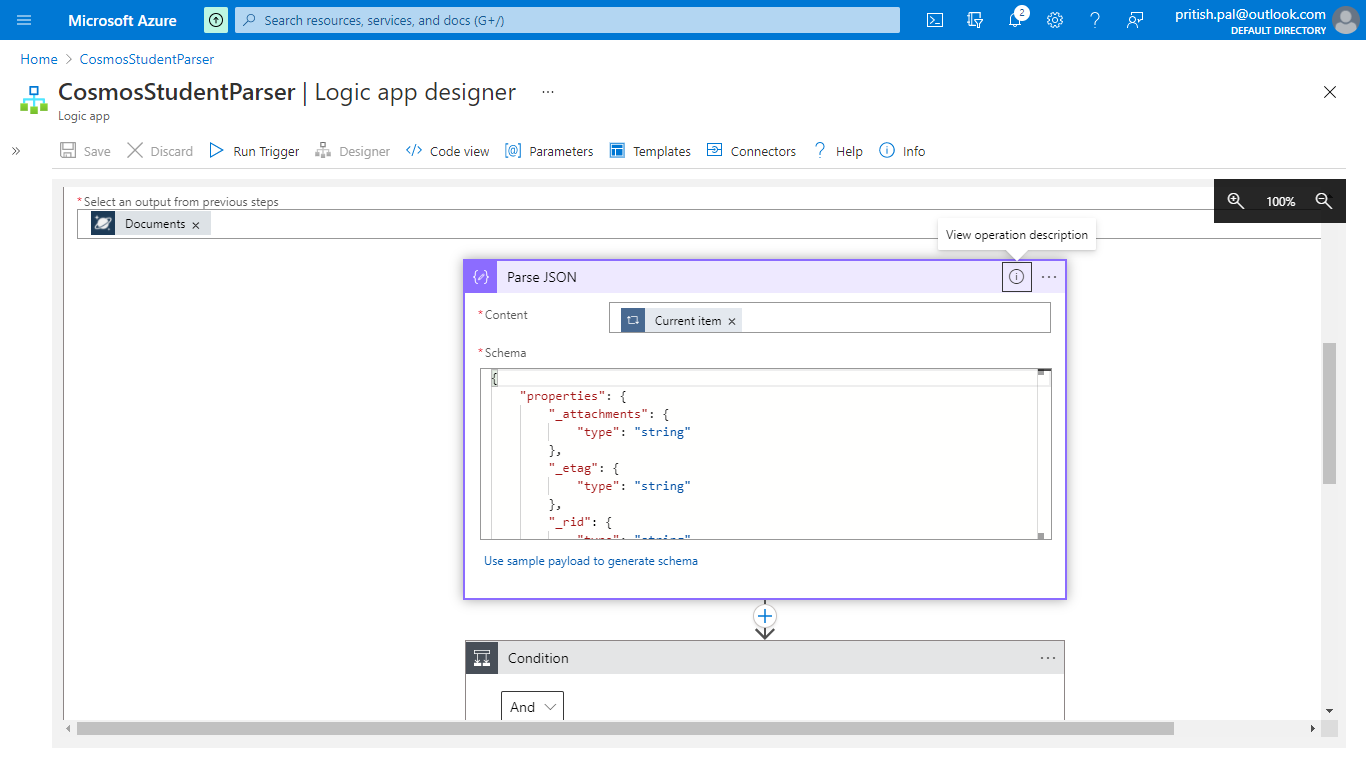
**Ex:** Select \* from Markscontainer



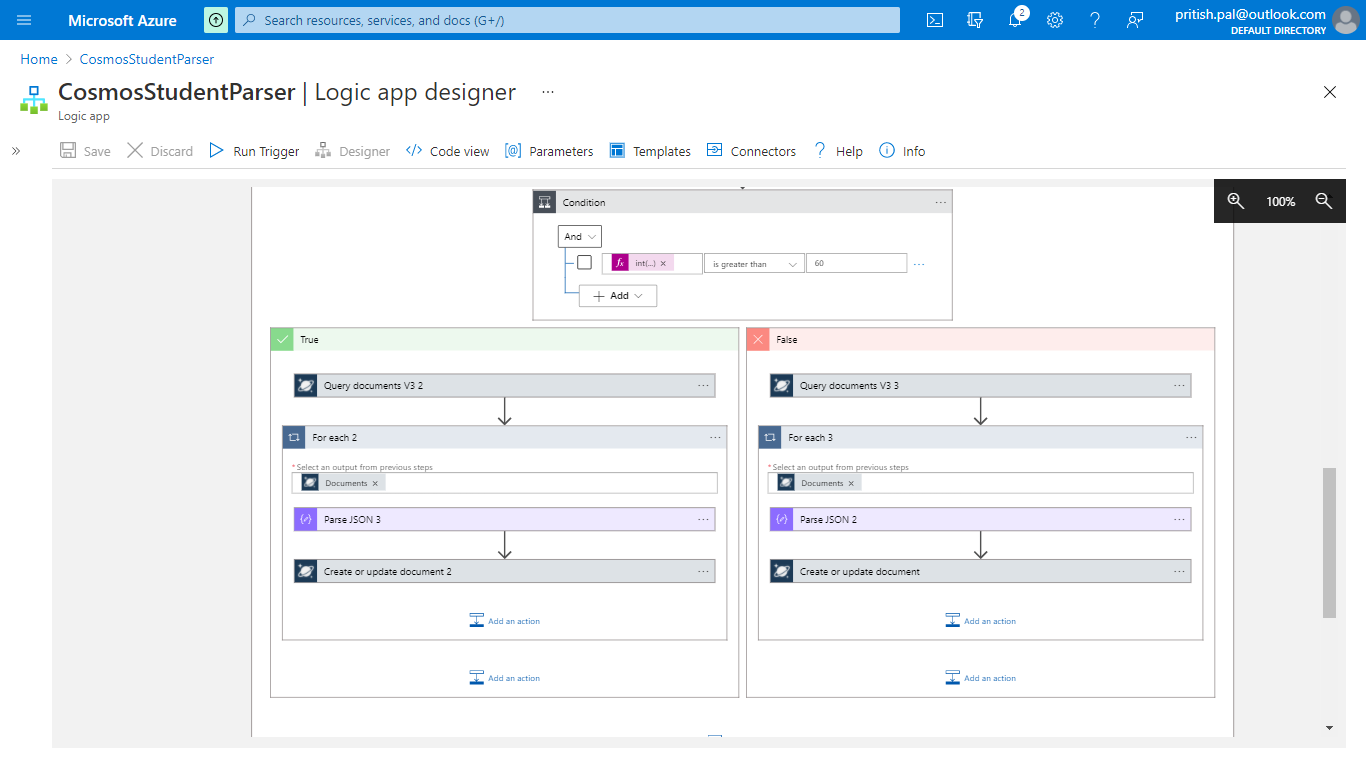
**FOREACH:** TO ITERATE THROUGH EACH DOCUMENT



**PARSE JSON:** TO READ THE JSON SCHEMA CONTENT BASED ON THE SCHEMA PROVIDED BY YOU



**IF CONDITION:** TO CHECK FOR A CONDITION



**References:**

**https://soltisweb.com/blog/detail/2020-28-28-updating-cosmos-db-documents-with-azure-logic-apps**

**https://docs.microsoft.com/en-us/azure/cosmos-db/create-sql-api-dotnet**