using Azure\_TableStorage;

using Microsoft.WindowsAzure.Storage;

using Microsoft.WindowsAzure.Storage.Table;

using System;

using System.Configuration;

class Program

{

static void Main(string[] args)

{

string connectionString = ConfigurationManager.AppSettings["StorageConnectionString"];

CloudStorageAccount cloudStorageAccount = CloudStorageAccount.Parse(connectionString);

CloudTableClient tableClient = cloudStorageAccount.CreateCloudTableClient();

Console.WriteLine("Enter Table Name to create");

string tableName = Console.ReadLine();

CloudTable cloudTable = tableClient.GetTableReference(tableName);

CreateNewTable(cloudTable);

while (true)

{

ScreenOptions();

switch (Console.ReadLine())

{

case "INSERT":

if(cloudTable == null)

{

Console.WriteLine("Table not created");

break;

}

InsertRecordToTable(cloudTable);

break;

case "UPDATE":

if (cloudTable == null)

{

Console.WriteLine("Table not created");

break;

}

UpdateRecordInTable(cloudTable);

break;

case "DELETE":

if (cloudTable == null)

{

Console.WriteLine("Table not created");

break;

}

DeleteRecordinTable(cloudTable);

break;

case "DISPLAY":

if (cloudTable == null)

{

Console.WriteLine("Table not created");

break;

}

DisplayTableRecords(cloudTable);

break;

case "DROP":

if (cloudTable == null)

{

Console.WriteLine("Table not created");

break;

}

DropTable(cloudTable);

cloudTable = null;

break;

default:

Console.WriteLine("Invalid Entry");

break;

}

Console.WriteLine("EXIT to exit the application. Any key t continue ");

if (Console.ReadLine() == "EXIT")

{

return;

}

}

}

public static void CreateNewTable(CloudTable table)

{

if (!table.CreateIfNotExists())

{

Console.WriteLine("Table {0} already exists", table.Name);

return;

}

Console.WriteLine("Table {0} created", table.Name);

}

public static void InsertRecordToTable(CloudTable table)

{

Console.WriteLine("Enter customer type");

string customerType = Console.ReadLine();

Console.WriteLine("Enter customer ID");

string customerID = Console.ReadLine();

Console.WriteLine("Enter customer name");

string customerName = Console.ReadLine();

Console.WriteLine("Enter customer details");

string customerDetails = Console.ReadLine();

Customer customerEntity = new Customer();

customerEntity.CustomerType = customerType;

customerEntity.CustomerID = Int32.Parse(customerID);

customerEntity.CustomerDetails = customerDetails;

customerEntity.CustomerName = customerName;

customerEntity.AssignPartitionKey();

customerEntity.AssignRowKey();

Customer custEntity = RetrieveRecord(table, customerType, customerID);

if (custEntity == null)

{

TableOperation tableOperation = TableOperation.Insert(customerEntity);

table.Execute(tableOperation);

Console.WriteLine("Record inserted");

}

else

{

Console.WriteLine("Record exists");

}

}

public static void UpdateRecordInTable(CloudTable table)

{

Console.WriteLine("Enter customer type");

string customerType = Console.ReadLine();

Console.WriteLine("Enter customer ID");

string customerID = Console.ReadLine();

Console.WriteLine("Enter customer name");

string customerName = Console.ReadLine();

Console.WriteLine("Enter customer details");

string customerDetails = Console.ReadLine();

Customer customerEntity = RetrieveRecord(table, customerType, customerID);

if (customerEntity != null)

{

customerEntity.CustomerDetails = customerDetails;

customerEntity.CustomerName = customerName;

TableOperation tableOperation = TableOperation.Replace(customerEntity);

table.Execute(tableOperation);

Console.WriteLine("Record updated");

}

else

{

Console.WriteLine("Record does not exists");

}

}

public static void DeleteRecordinTable(CloudTable table)

{

Console.WriteLine("Enter customer type");

string customerType = Console.ReadLine();

Console.WriteLine("Enter customer ID");

string customerID = Console.ReadLine();

Customer customerEntity = RetrieveRecord(table, customerType, customerID);

if (customerEntity != null)

{

TableOperation tableOperation = TableOperation.Delete(customerEntity);

table.Execute(tableOperation);

Console.WriteLine("Record deleted");

}

else

{

Console.WriteLine("Record does not exists");

}

}

public static Customer RetrieveRecord(CloudTable table,string partitionKey,string rowKey)

{

TableOperation tableOperation = TableOperation.Retrieve<Customer>(partitionKey, rowKey);

TableResult tableResult = table.Execute(tableOperation);

return tableResult.Result as Customer;

}

public static void DisplayTableRecords(CloudTable table)

{

TableQuery<Customer> tableQuery = new TableQuery<Customer>();

foreach (Customer customerEntity in table.ExecuteQuery(tableQuery))

{

Console.WriteLine("Customer ID : {0}", customerEntity.CustomerID);

Console.WriteLine("Customer Type : {0}", customerEntity.CustomerType);

Console.WriteLine("Customer Name : {0}", customerEntity.CustomerName);

Console.WriteLine("Customer Details : {0}", customerEntity.CustomerDetails);

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

}

public static void DropTable(CloudTable table)

{

if (!table.DeleteIfExists())

{

Console.WriteLine("Table does not exists");

}

}

public static void ScreenOptions()

{

Console.WriteLine("");

Console.WriteLine("INSERT - To Insert record to table");

Console.WriteLine("UPDATE - To update record in table");

Console.WriteLine("DELETE - To delete record in table");

Console.WriteLine("DISPLAY - To display all records in table");

Console.WriteLine("DROP - To drop a table");

Console.WriteLine("Select option");

}

}