# CLDV6212

## PoE

Nikki Gordhan
ST10108243

# Contents

Part A: List the Azure Components utilised thus far ..... 2
Part B: Motivation for needed change ..... 3
Part C: Implement necessary change to Part 2 ..... 4
Screenshot of the function code: ..... 4
Screenshot of 5 messages in the Azure Portal: ..... 7
Screenshots of the corresponding messages in the changed component (Azure Table Storage): ..... 7
Part D: Testing ..... 8
Screenshots of the console application (Part 2) running: ..... 8
GitHub Link: ..... 11
References ..... 12

## Part A: List the Azure Components utilised thus far

| Component | Technology Choice | Hosting Model |
|---|---|---|
| Azure SQL Database | Data Storage | PAAS |
| HTTP Trigger (Azure Functions) | Compute | PAAS |
| Queue Trigger (Azure Functions) | Compute | PAAS |
| Azure Queue Storage | Data Storage | PAAS |

## Part B: Motivation for needed change

There are many factors to consider when changing an Azure SQL Database to an Azure Table Storage such the data model, scalability, the performance, the cost, and the features. below is a summary of some of the reasons why I would change the component to an Azure Table Storage:

1. **Cost-effectiveness:** Azure Table Storage is cheaper than using Azure SQL Database. Azure Table Storage is a more cost-effective solution for storing large amounts of data. (Rwakatungu, 2011)
2. **Data model:** Azure storage table is a wide column store that stores data in key-value pairs without any schema or relations. Azure SQL database is a relational DBMS that stores data in tables with predefined schema and supports relations, indexes, constraints. (DB-Engines, unknwon)
3. **Scalability:** Azure storage table can scale up to 100 TB of data and handle large volumes of semi-structured or unstructured data. Azure SQL database has a limit of 150 GB for the business edition and 5 GB for the web edition and is more suitable for structured data. (Patel, 2021)
4. **Performance:** Azure storage table can provide faster access to data if the queries are simple and based on the partition key or row key. Azure SQL database can provide more complex queries and transactions using SQL but may have higher latency and lower throughput. (DB-Engines, unknwon)
5. **Cost:** Azure storage table is cheaper than Azure SQL database for storing large amounts of data. Azure storage table charges based on the amount of data stored, the number of transactions, and the bandwidth used. Azure SQL database charges based on the database size, the number of databases, and the service tier. (DB-Engines, unknwon)
6. **Features:** Azure storage table supports features such as geo-replication, optimistic concurrency, and RESTful API. Azure SQL database supports features such as Transact-SQL, triggers, foreign keys, ACID transactions, and server-side scripts. (Patel, 2021)

In conclusion, the choice between Azure storage table and Azure SQL database depends on the type and size of data, the query and transaction requirements, the budget and the desired features. Azure storage table is better for storing large volumes of non-relational data that can be accessed quickly and cheaply, which meets the primary requirement of Aweh Productions.

## Part C: Implement necessary change to Part 2

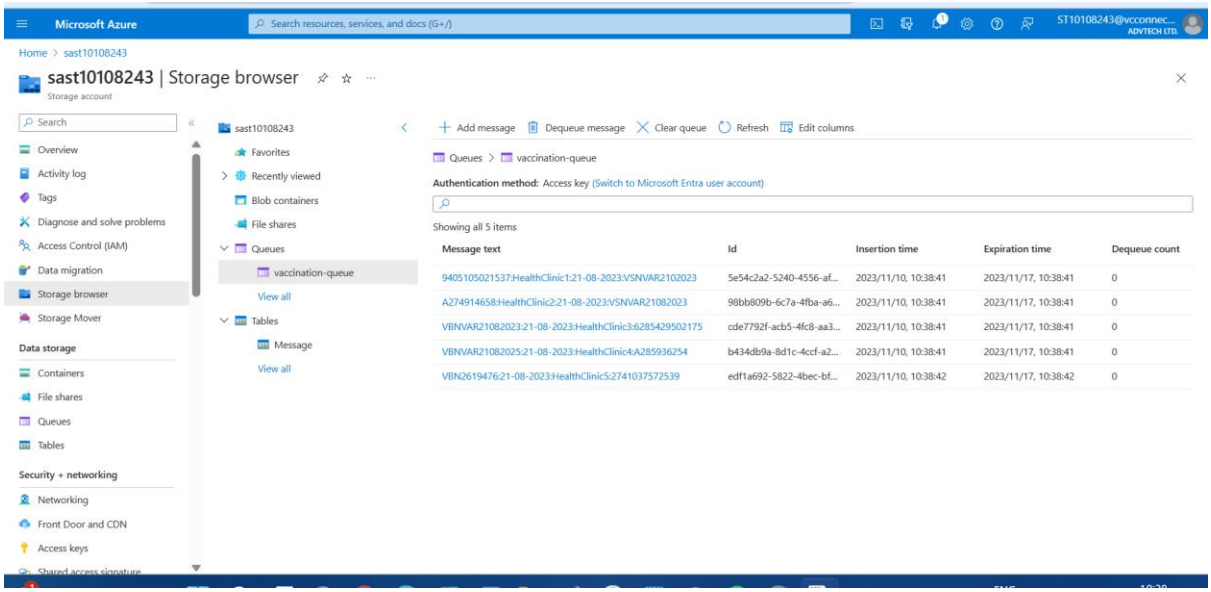
### Screenshot of the function code:

```
using System;
using System.Globalization;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;
using Microsoft.Azure.Cosmos.Table;
using Microsoft.Azure.Storage.Blob.Protocol;
using Microsoft.Azure.WebJobs;
using Microsoft.Azure.WebJobs.Host;
using Microsoft.Extensions.Logging;

namespace ST10108243_AwehProd_QueueTriggerUpdated
{
    public class Function1
    {
        [FunctionName("TableStorage")]
        public async Task Run([QueueTrigger("vaccination-queue", Connection =
"CONNECTION_STRING")] string myQueueItem,
        [Table ("Message", Connection = "CONNECTION_STRING")] CloudTable
cloudTable, ILogger log)
        {
            try
            {
                string sEncodeMessage =
Convert.ToBase64String(Encoding.UTF8.GetBytes(myQueueItem));
                // encodes the message to a 64 byte.
                // the line of code was taken and adapted from StackOverflow.
                // https://stackoverflow.com/questions/11743160/how-do-i-encode-and-
decode-a-base64-string
                // Zeigeist
                // https://stackoverflow.com/users/2649698/zeigeist

                string[] messageParts =
Encoding.UTF8.GetString(Convert.FromBase64String(sEncodeMessage)).Split(':');
                // seperates the string when ever there is a ':'.
                // the line of code above was taken and adapted from Microsoft
Documenets.
                // https://learn.microsoft.com/en-
us/dotnet/api/system.convert.frombase64string?view=net-7.0

                //assumptions
```

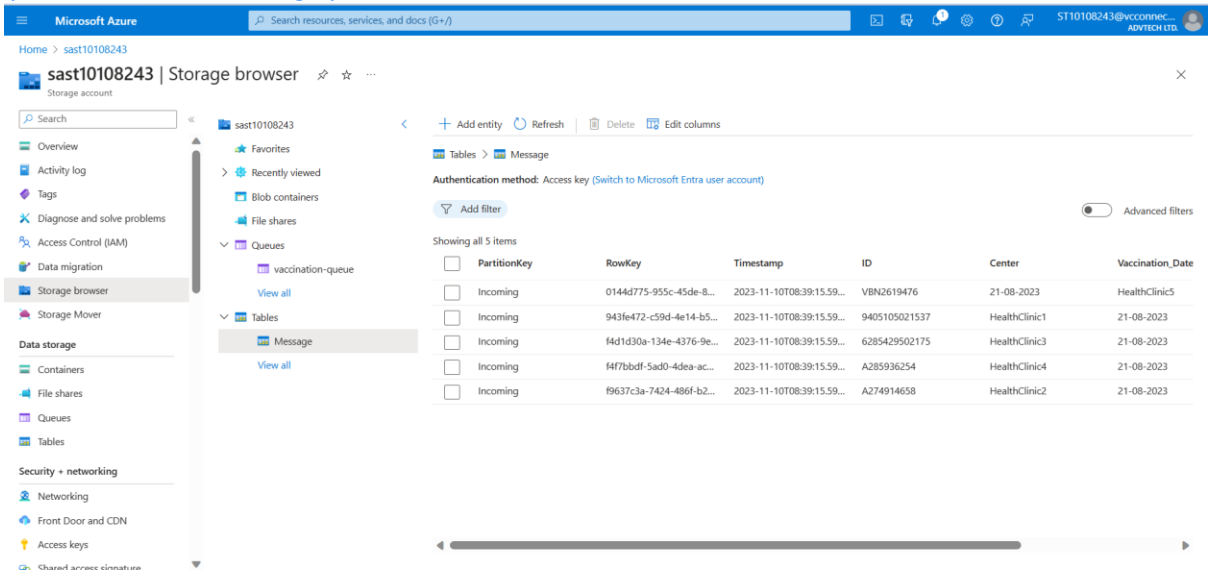
```
            //if else firt element is 13 or 10 then it first format else its the second format
            string sID = "";
            string sCenter = "";
            string sVaccination_Date = "";
            string sSerial_Number = "";


            if (messageParts[0].Length == 13 || messageParts[0].Length == 10)
            {
                // format 1- Id:VaccinationCenter:VaccinationDate:VaccineSerialNumber
                sID = messageParts[0];
                sCenter = messageParts[1];
                sVaccination_Date = messageParts[2];
                sSerial_Number = messageParts[3];
            }
            else
            {
                // format 2- VaccineBarcode:VaccinationDate:VaccinationCenter:Id
                sID = messageParts[3];
                sCenter = messageParts[2];
                sVaccination_Date = messageParts[1];
                sSerial_Number = messageParts[0];
            }
            // assigns the columns into the array elements.

            //CREATE A NEW MESSSAGE ENTITY:
                MessageEntity messageEntity = new
MessageEntity(sID,sCenter,sVaccination_Date,sSerial_Number);
            // create a MessageEnitity object that passes in the values.

            //CREATE A TABLE OPERATION TO INSERT THE ENTITY:
            TableOperation insertOperation = TableOperation.Insert(messageEntity);
            // inserts the messages from the MessageEntity object.

            //EXECUTE THE INSERT OPERATION:
            await cloudTable.ExecuteAsync(insertOperation);
            // executes the insertoperation.

            log.LogInformation($"Queue message has been stored successfully in the
database ID = {sID}, Center = {sCenter}, Vaccination_Date = {sVaccination_Date},
Serial_Number = {sSerial_Number} ");
        }
        catch (Exception ex)
```

```
            {
                log.LogError($"Error in processing the queue message: {ex.Message}");
            }

        }

        public class MessageEntity : TableEntity
        {
            public MessageEntity(string sID, string sCenter, string sVaccination_Date,
string sSerial_Number)
            {
                PartitionKey = "Incoming";
                RowKey = Guid.NewGuid().ToString();
                ID = sID;
                Center = sCenter;
                Vaccination_Date = sVaccination_Date;
                Serial_Number = sSerial_Number;
            }
            // gives the Message entity colunm headings.

            public MessageEntity() { }

            public string ID { get; set; }
            public string Center { get; set; }
            public string Vaccination_Date { get; set; }
            public string Serial_Number { get; set; }
            // getters and setter for each of the columns in the MessageEntity.

        }
    }
}
```

## Screenshot of 5 messages in the Azure Portal:

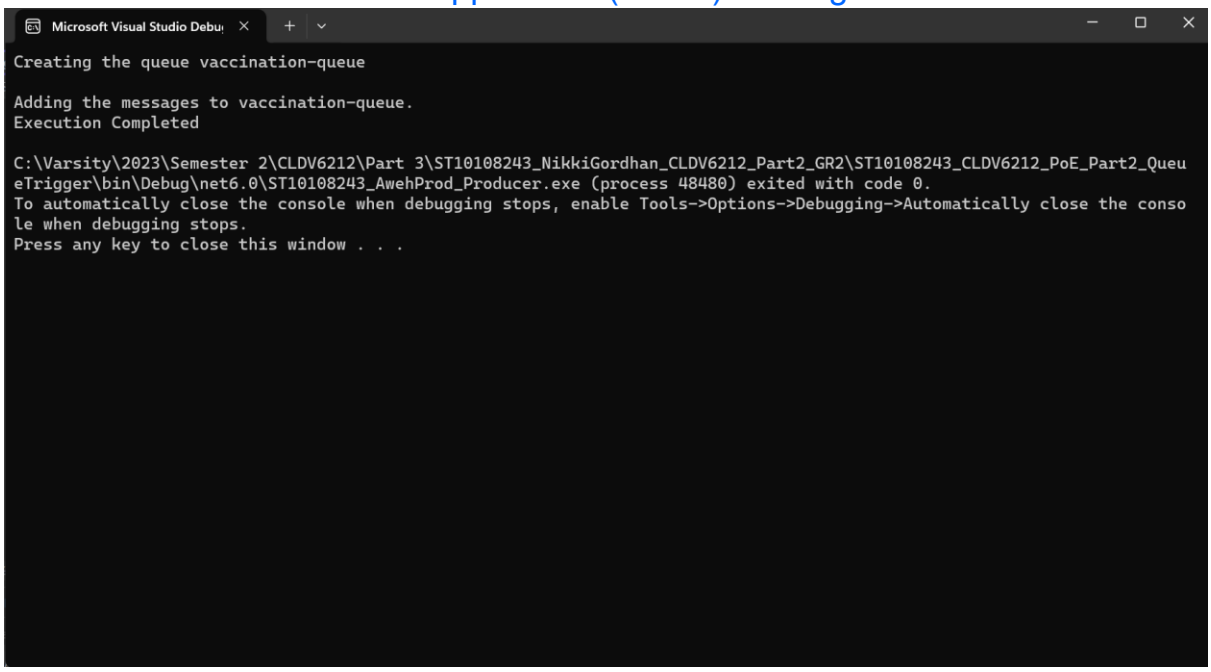


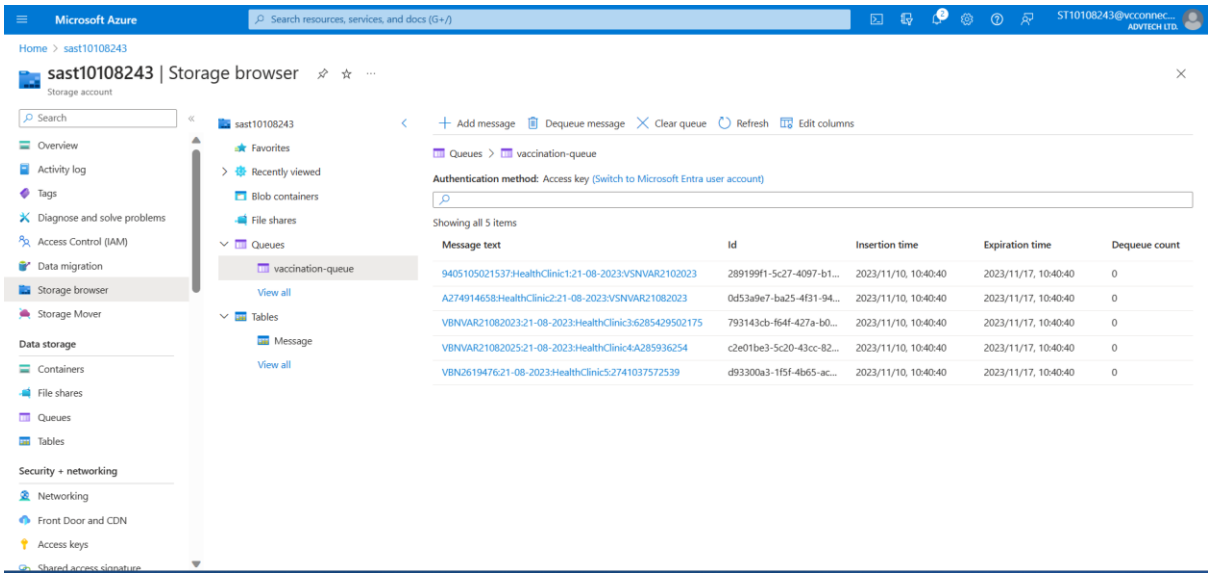

## Screenshots of the corresponding messages in the changed component (Azure Table Storage):

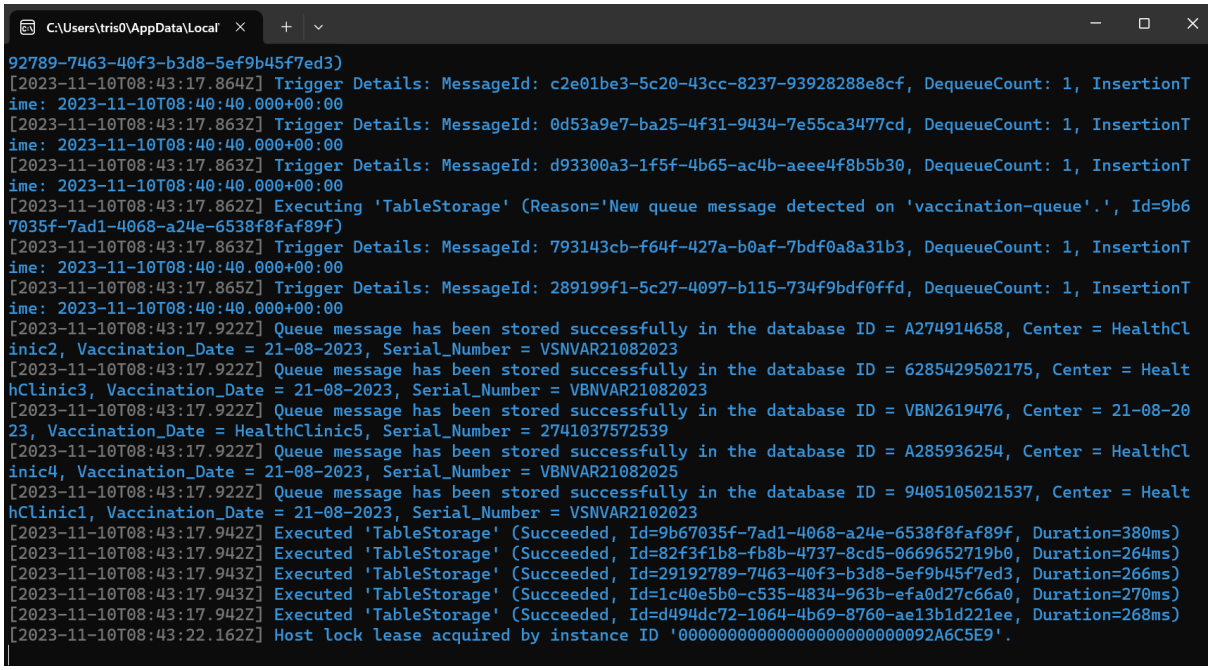


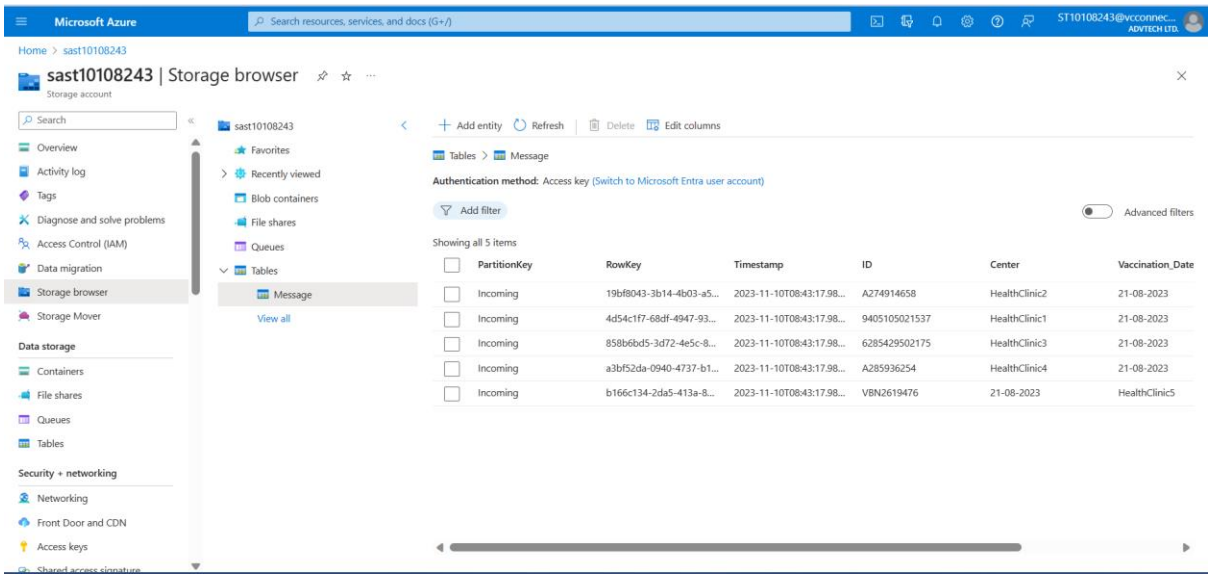
# Part D: Testing

## Screenshots of the console application (Part 2) running:






Message 1:

| Message text | Id | Insertion time | Expiration time | Dequeue count |
|---|---|---|---|---|
| 9405105021537:HealthClinic1:21-08-2023:VSNVAR2102023 | 289199f1-5c27-4097-b1... | 2023/11/10, 10:40:40 | 2023/11/17, 10:40:40 | 0 |

| PartitionKey | RowKey | Timestamp | ID | Center | Vaccination_Date | Serial_Number |
|---|---|---|---|---|---|---|
| Incoming | 19bf8043-3b14-4b03-a5... | 2023-11-10T08:43:17.98... | A274914658 | HealthClinic2 | 21-08-2023 | VSNVAR21082023 |
| Incoming | 4d54c1f7-68df-4947-93... | 2023-11-10T08:43:17.98... | 9405105021537 | HealthClinic1 | 21-08-2023 | VSNVAR2102023 |
| Incoming | 858b6bd5-3d72-4e5c-8... | 2023-11-10T08:43:17.98... | 6285429502175 | HealthClinic3 | 21-08-2023 | VBNVAR21082023 |
| Incoming | a3bf52da-0940-4737-b1... | 2023-11-10T08:43:17.98... | A285936254 | HealthClinic4 | 21-08-2023 | VBNVAR21082025 |
| Incoming | b166c134-2da5-413a-8... | 2023-11-10T08:43:17.98... | VBN2619476 | 21-08-2023 | HealthClinic5 | 2741037572539 |

## Message 2:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A274914658:HealthClinic2:21-08-2023:VSNVAR21082023 | 0d53a9e7-ba25-4f31-9434-7e55ca3477cd | 2023/11/10, 10:40:40 | 2023/11/17, 10:40:40 | 0 | | | |

| | PartitionKey | RowKey | Timestamp | ID | Center | Vaccination_Date | Serial_Number |
|---|---|---|---|---|---|---|---|
| ☑ | Incoming | 19bf8043-3b14-4b03-a5... | 2023-11-10T08:43:17.98... | A274914658 | HealthClinic2 | 21-08-2023 | VSNVAR21082023 |

## Message 3:

| | | | | | |
|---|---|---|---|---|---|
| VBNVAR21082023:21-08-2023:HealthClinic3:6285429502175 | 793143cb-f64f-427a-b0af-7bdf0a8a31b3 | 2023/11/10, 10:40:40 | 2023/11/17, 10:40:40 | 0 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ☑ | Incoming | 858b6bd5-3d72-4e5c-8... | 2023-11-10T08:43:17.98... | 6285429502175 | HealthClinic3 | 21-08-2023 | VBNVAR21082023 |

## Message 4:

| | | | | | |
|---|---|---|---|---|---|
| VBNVAR21082025:21-08-2023:HealthClinic4:A285936254 | c2e01be3-5c20-43cc-8237-93928288e8cf | 2023/11/10, 10:40:40 | 2023/11/17, 10:40:40 | 0 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ☑ | Incoming | a3bf52da-0940-4737-b1... | 2023-11-10T08:43:17.98... | A285936254 | HealthClinic4 | 21-08-2023 | VBNVAR21082025 |

## Message 5:

| | | | | | |
|---|---|---|---|---|---|
| VBN2619476:21-08-2023:HealthClinic5:2741037572539 | d93300a3-1f5f-4b65-ac4b-aeee4f8b5b30 | 2023/11/10, 10:40:40 | 2023/11/17, 10:40:40 | 0 | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ☑ | Incoming | b166c134-2da5-413a-8... | 2023-11-10T08:43:17.98... | VBN2619476 | 21-08-2023 | HealthClinic5 | 2741037572539 |

## GitHub Link:

https://github.com/IIEWFL/cldv6212-poe-nikkigordhan.git

# References

DB-Engines, unknwon. *System Properties Comparison Microsoft Azure SQL Database vs. Microsoft Azure Table Storage.* [Online]
Available at: https://db-engines.com/en/system/Microsoft+Azure+SQL+Database%3BMicrosoft+Azure+Table+Storage
[Accessed 10 Novvember 2023].

Patel, A., 2021. *Azure — Difference between Azure SQL Database and SQL Server on VM.* [Online]
Available at: https://medium.com/awesome-azure/azure-difference-between-azure-sql-database-and-sql-server-on-vm-comparison-azure-sql-vs-sql-server-vm-cf02578a1188
[Accessed 10 November 2023].

Rwakatungu, A., 2011. *When should I use Sql Azure and when should I use table Storage?.* [Online]
Available at: https://stackoverflow.com/questions/4930368/when-should-i-use-sql-azure-and-when-should-i-use-table-storage
[Accessed 10 November 2023].

VKinfotek Inc., unknown. *Difference between SQL database table and Azure storage table?.* [Online]
Available at: https://vkinfotek.com/azureqa/difference-between-sql-database-table-and-azure-storage-table.html
[Accessed 10 November 2023].