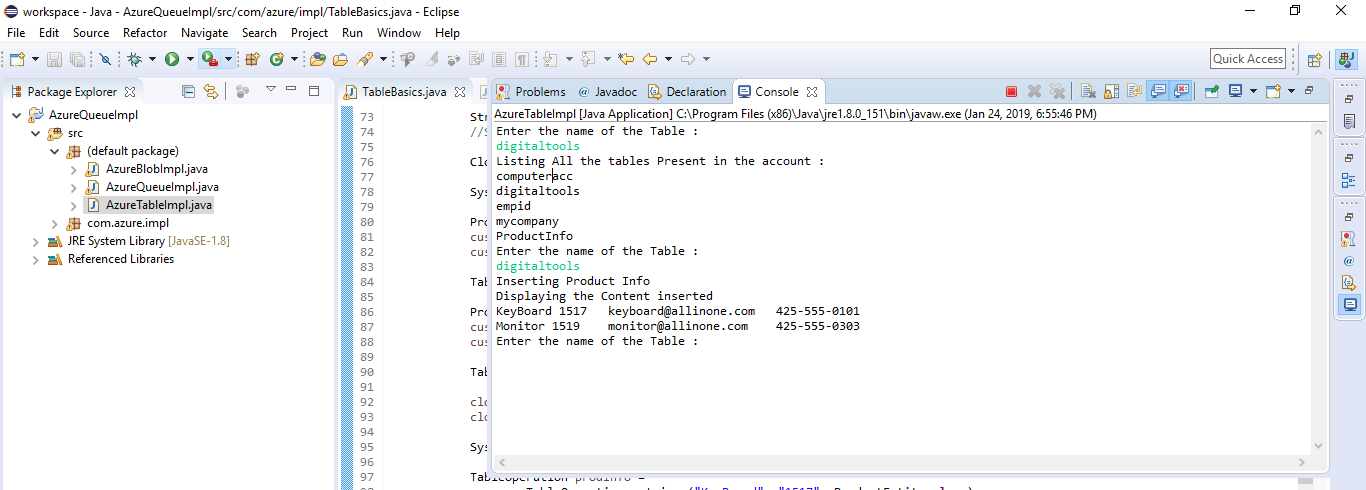
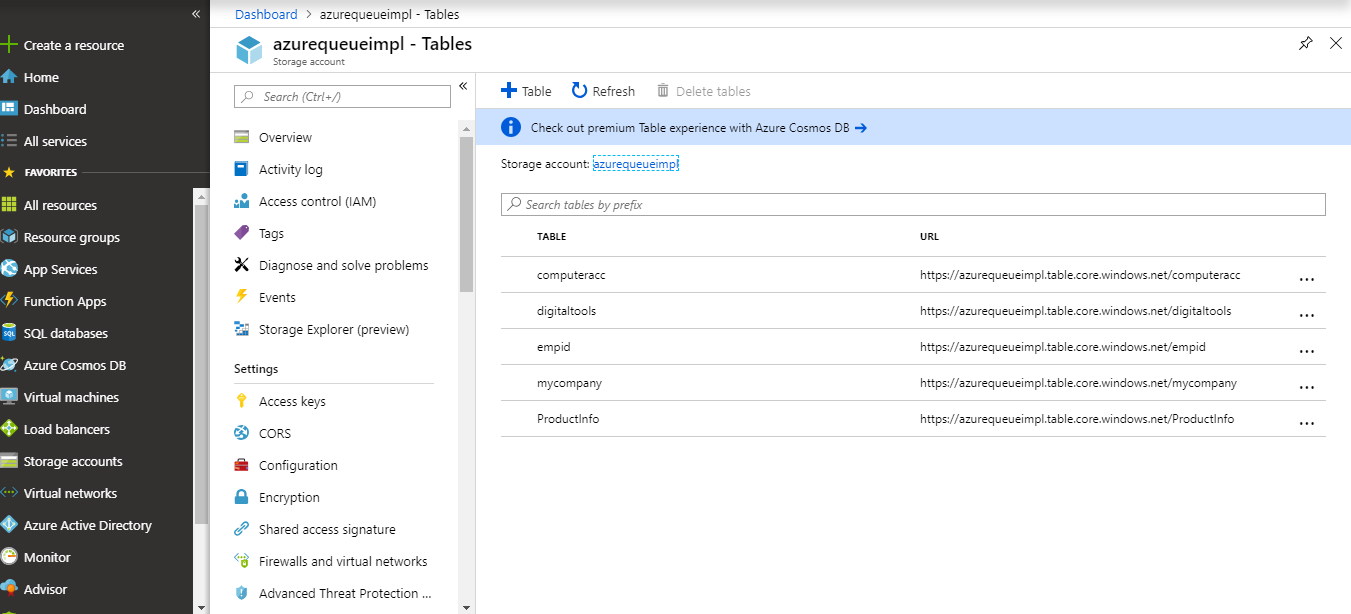
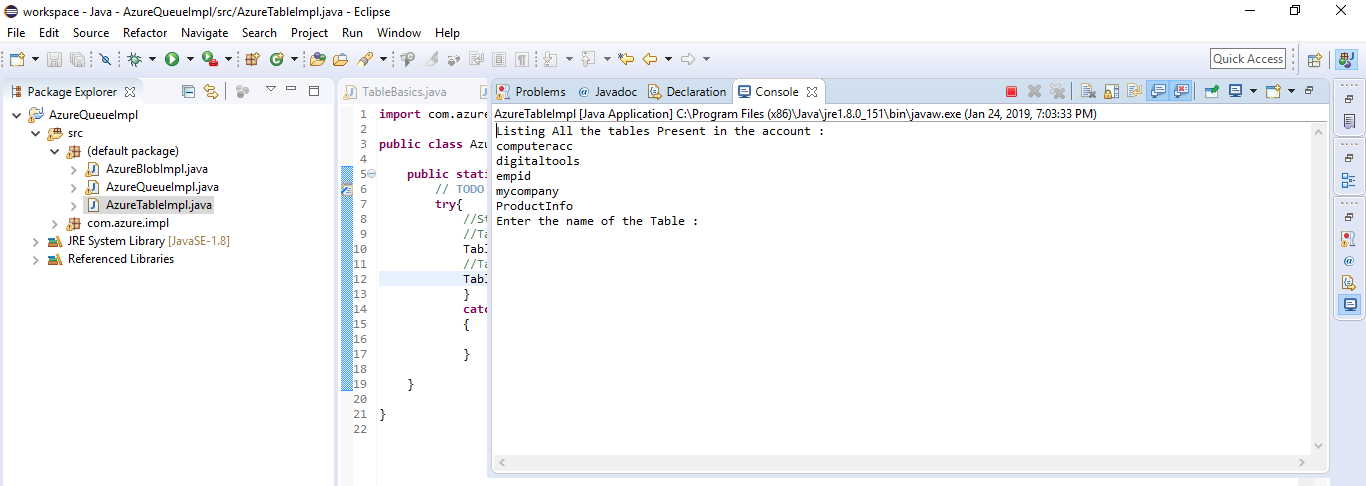
**Demonstration of table in Azure**

Step 1: User entering the table name and displaying all the tables along with the content present in it via console program



Step 2 : Verifying all the tables present via Portal and Via Java Console





Here is the Java Source Code :

AzureTableImpl.java

**import** com.azure.impl.TableBasics;

**public** **class** AzureTableImpl {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**try**{

//String name = reader.readLine();

TableBasics.*createTable*();

TableBasics.*listTable*();

TableBasics.*storeTable*();

TableBasics.*deleteTable*();

}

**catch**(Exception e)

{

e.printStackTrace();

}

}

}

TableBasics.java

package com.azure.impl;

import java.io.\*;

import com.microsoft.azure.storage.\*;

import com.microsoft.azure.storage.table.\*;

import com.microsoft.azure.storage.table.TableQuery.\*;

import com.azure.impl.ProductEntity;

public class TableBasics {

public static final String storageConnectionString =

"DefaultEndpointsProtocol=https;" +

"AccountName=azurequeueimpl;" +

"AccountKey=6E86jjYlvjhjHuUr493ZpRtG2faC4uGY0Bev7nkIIfj+AzKqzKLqHMzxcgNq2SD6YizXow3AAB+dkqPiMDtA1A==";

public static void createTable() throws Exception

{

try{

CloudStorageAccount storageAccount =

CloudStorageAccount.parse(storageConnectionString);

CloudTableClient tableClient = storageAccount.createCloudTableClient();

BufferedReader reader =

new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the name of the Table :");

String tableName = reader.readLine();

//System.out.println("Table Name 1 :"+tableName);

CloudTable cloudTable = tableClient.getTableReference(tableName);

cloudTable.createIfNotExists();

}//end of try block

catch(Exception e){

e.printStackTrace();

}

}//end of runtable method

public static void listTable() throws Exception

{

try{

CloudStorageAccount storageAccount =

CloudStorageAccount.parse(storageConnectionString);

CloudTableClient tableClient = storageAccount.createCloudTableClient();

System.out.println("Listing All the tables Present in the account :");

for (String table : tableClient.listTables())

{

// Output each table name.

System.out.println(table);

}

}//end of try block

catch(Exception e){

e.printStackTrace();

}

}//end of listtable method

public static void storeTable() throws Exception

{

try{

CloudStorageAccount storageAccount =

CloudStorageAccount.parse(storageConnectionString);

CloudTableClient tableClient = storageAccount.createCloudTableClient();

BufferedReader reader =

new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the name of the Table :");

String tableName = reader.readLine();

//System.out.println("Table Name 2 :"+tableName);

CloudTable cloudTable = tableClient.getTableReference(tableName);

System.out.println("Inserting Product Info");

ProductEntity customer1 = new ProductEntity("KeyBoard", "1517");

customer1.setEmail("keyboard@allinone.com");

customer1.setPhoneNumber("425-555-0101");

TableOperation insertCustomer1 = TableOperation.insertOrReplace(customer1);

ProductEntity customer2 = new ProductEntity("Monitor", "1519");

customer2.setEmail("monitor@allinone.com");

customer2.setPhoneNumber("425-555-0303");

TableOperation insertCustomer2 = TableOperation.insertOrReplace(customer2);

cloudTable.execute(insertCustomer1);

cloudTable.execute(insertCustomer2);

System.out.println("Displaying the Content inserted");

TableOperation prodInfo =

TableOperation.retrieve("KeyBoard", "1517", ProductEntity.class);

ProductEntity specificEntity =

cloudTable.execute(prodInfo).getResultAsType();

//System.out.println("SpecificEntity value :"+ specificEntity);

if (specificEntity != null)

{

System.out.println(specificEntity.getPartitionKey() +

" " + specificEntity.getRowKey() +

"\t" + specificEntity.getEmail() +

"\t" + specificEntity.getPhoneNumber());

}

TableOperation prodInfo1 =

TableOperation.retrieve("Monitor", "1519", ProductEntity.class);

ProductEntity specificEntity1 =

cloudTable.execute(prodInfo1).getResultAsType();

if (specificEntity1 != null)

{

System.out.println(specificEntity1.getPartitionKey() +

" " + specificEntity1.getRowKey() +

"\t" + specificEntity1.getEmail() +

"\t" + specificEntity1.getPhoneNumber());

}

}//end of try block

catch(Exception e){

e.printStackTrace();

}

}//end of storetable method

public static void deleteTable() throws Exception

{

try{

CloudStorageAccount storageAccount =

CloudStorageAccount.parse(storageConnectionString);

CloudTableClient tableClient = storageAccount.createCloudTableClient();

BufferedReader reader =

new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter the name of the Table :");

String tableName = reader.readLine();

CloudTable cloudTable = tableClient.getTableReference(tableName);

cloudTable.deleteIfExists();

System.out.println("Table Deleted Successfully : "+tableName);

System.out.println("Tables After Deletion are - ");

listTable();

}//end of try block

catch(Exception e){

e.printStackTrace();

}

}//end of deletetable method

}