Middle East Technical University | Department of Computer Engineering

CENG 495

Cloud Computing

Spring '2016-2017

Assignment 2 - NoSQL Database

Due date: April 23rd 2017, Sunday, 23:55

1 Objective

This homework aims to get you familiar with the NoSQL Databases and Database as a Service (DBaaS) systems namely DocumentDB, which is mainly a DBaaS platform. You will develop and deploy a website using Microsoft Azure Cloud Platform.

In short; develop and deploy your web application which uses DBaaS (Database as a Service) over PaaS (Platform as a Service).

Keywords: *Cloud Computing, PaaS, SaaS, DBaaS, Microsoft, Azure, DocumentDB*

2 Problem Definition

3 Specifications

1. In this homework you will develop a web application and deploy it on Microsoft Azure Platform. You will be provided Azure promo codes via METU Mail!
2. After following the steps specified in the e-mails sent to your ceng mails, you will be able to deploy your application to Azure environment. Your application must be accessible from the address ***your-app-name.azurewebsites.net***
3. Again you are free to use any programming languages and frameworks supported by Azure Cloud Platform (Javascript, Java, Python, etc.).

3.1 Document Format

(a) The application data will be stored on a DocumentDB database which is a database service that is provided by Azure. DocumentDB is Microsoft's multi-tenant distributed database service for managing JSON documents at Internet scale.
For this homework, "Single Tier" will be sufficient for DocumentDB partition selection.

(b) DocumentDB is basically a document based NoSQL database. This database allows data to be stored as JSON documents. The database can provide a RESTful API for accessing the documents and query your indexes with web browser.

A movie sample

Id	Release Date	Movie Name	Tags	Url	Actors
1	2016-05-03	Straight Outta Compton	gangsta,rap,eazy,streets,police,compton	https://straighouttacompton.com	O'Shea Jackson Jr., Corey Hawkins, Jason Mitchell

(c) For NoSQL databases, there is no fixed type of data representation unlike RDBMSes. However, a possible document and the corresponding recipe example is given below:

```
{
  "id": "1",
  "releaseDate": "2016-05-03",
  "movieName": "Straight Outta Compton",
  "tags": "gangsta,rap,eazy,streets,police,compton",
  "url": "https://straighouttacompton.com",
  "actors": "O'Shea Jackson Jr., Corey Hawkins, Jason Mitchell"
}
```

3.2 Application

(a) The main functionality of the application is to show the movies that is previously added to the system. It also allows user to create, update, and delete the movies.

(b) For documentDB interactions, you should provide "id" for all of your documents. The "id" choice of your data structure is up to you (i have used an automatic incremented value for all documents). Also, when dealing with "id" of documents, remember that documentDB only accepts strings as "id".

(c) The application displays movies that are in the database at a certain time. That means, there is no movie in the application window initially as the database does not contain any records. However, if any movie is entered to the system then the records in the database are displayed even if the page is reloaded.

(d) ***The application allows user to update movie documents in the database. Also, user should be able to delete the whole movie document. In short, your application is expected to insert a document with 5 must attributes (release date, movie name, tags, url, actors) and uncertain number of optional attributes, update the attributes that already has been inserted into DB, and delete the whole document.***

(e) You are free to use any page design as long as your application provides the assignment specifications but your documents should be readable on the website. So, do not send an application that shows all document attributes on one line or so!

(f) ***You can assume that any erroneous case will not be tested. That means, you do not have to handle any wrong inputs (Trying to update empty fields etc.), but beware that all functionality of the application should be provided like adding any number of extra fields and main fields, updating current fields. Also, after modifying/deleting a document, website should update itself immediately!***

4 Deliverables

In this assignment, you are expected to both deploy your solution to Azure and submit your source code to COW. For submission on COW, a tar.gz archive file (named hw2.tar.gz) that contains all your source code files and a README file which includes your app-name and your language choice is expected. For your Azure submission, you should deploy your solution to ***your-app-name.azurewebsites.net***.

5 Evaluation

1. Submission schedule will be strict. However you will have ***3*** extra days with penalty. Penalty formula is ***-5xDayxDay***. Submission time will be determined by your last source code submission to coursepage on COW.

2. Your application will be tested only once except the objection. You should also keep in mind that you should attend the objection hour of the day which you submit your code if you want to object your grade. Any objection after that day will not be accepted.

3. You can visit A301 for objections.

6 Useful Links

1. An example application for this assignment can be found at (You can add, delete, modify a movie any way you want.):
`http://ceng495-assignment2.azurewebsites.net/`

2. How to activate your Azure subscription:
   http://www.microsoftazurepass.com/howto

3. Some useful websites for DocumentDB:
   https://docs.microsoft.com/en-us/azure/documentdb/documentdb-create-collection
   https://docs.microsoft.com/en-us/azure/documentdb/documentdb-sql-query

4. Links related with deploment to Azure:
   https://docs.microsoft.com/en-us/azure/app-service-web/web-sites-deploy
   https://docs.microsoft.com/en-us/azure/app-service-web/app-service-deploy-local-git
   https://docs.microsoft.com/en-us/azure/app-service-web/app-service-web-get-started

5. When i was developing the application, i have used Nodejs framework. Below are the links that i have used frequently (I strongly suggest the nodejs framework, btw):
   https://github.com/Microsoft/azure-docs/blob/master/articles/app-service-web/web-sites-nodejs-develop-deploy-mac.md
   https://docs.microsoft.com/en-us/azure/app-service-web/app-service-web-nodejs-get-started

7 Cheating

We have zero tolerance policy for cheating. There is no teaming up! People involved in cheating will be punished according to the university regulations and will get 0. You can discuss design choices or language preferences, but sharing code between each other or using third party code is strictly forbidden. In case a match is found, this will be considered as cheating. Even if you take a "part" of the code from somewhere/somebody else - this is also cheating. Please be aware that there are "very advanced tools" that detect if two codes are similar. So please don't think you can get away with by changing a code obtained from another source.