**Northeastern University**

CS 6620 Cloud Computing

**Project #3** [200 points]

**Database with Amazon RDS for SQL and A Complete App on AWS**

# Guidelines

Project #3 should be electronically submitted to Canvas by midnight on the due date. A submission link is provided on Canvas.

# Assignment Overview

For this project, in part 1, you will design SQL schemas for your Django e-commerce website, create tables and run queries locally, then build a complete Web App on AWS following the example given.

Please post any questions to Piazza and attend TA office hours to address any questions. Get started early.

**PART I: SQL with Amazon RDS and Redshift [100 points]**

Let’s assume you’re the seller for your e-commerce website, now designing the SQL tables for your products, customers, and orders, to keep track of the basic sales condition.

Refer to textbook page 244~255 for MySQL, page 269~279 for Amazon Redshift.

1. SQL schema design: draw ER diagram like p244 Figure 6.1 for 3 tables: **[10 points]**

customer : order = 1 : N

order : product = N : M

Customers (cid, customer\_name, gender, phone)

Order (oid, cid, is\_completed, total\_price)

OrderHasProduct (oid, pid, product\_count)

Product (pid, product\_name, price, cost, is\_on\_sale, available\_stock, description)

Types: gender boolean, is\_completed boolean, total\_price int, product\_count int, price int, cost int, is\_on\_sale boolean, available\_stock int. all others are string

*(if you have your own thought about the schemas, please post on Piazza or discuss with TA ahead.)*

1. Locally: create the above tables in your local SQL or MySQL server. **[30 points]**

Insert 8 lines of data for each table and do the following queries.

consider generating some data that can print out at least 2 lines of results when doing the queries.

1. Get all the order.
2. Get top 3 orders with highest total\_price.
3. Get all the orders belong to customer Bob.
4. Get the number of products that is on sale.
5. Get the total income for the completed orders. (income = price – cost)
6. Get the product and the number that need to purchase to fulfill the order. (Available\_stock - product\_count)
7. Amazon RDS: refer to Box 6.4 6.5 in p253 to launch an RDS instance, and create the above MySQL tables, execute the data insertion and doing queries 4~6. You can reuse the code in part 2. **[30 points]**
8. Amazon Redshift: refer to Box 6.13 to load your data into with S3 bucket. You can use any input type other than csv. Show the process to add additional nodes into the cluster, remove nodes from the cluster while remaining operational. The nodes can be duplicated data but different names. Do queries 1~3. **[30 points]**

**PART II: Build a basic Web App and on AWS [100 points][Coding is required]**

Please study this example of building a complete Web App on AWS, using the below infrastructure (IaaS):

[https://medium.com/the-andela-way/designing-a-three-tier-architecture-in-aws-e5c24671f124](https://nam12.safelinks.protection.outlook.com/?url=https%3A%2F%2Fmedium.com%2Fthe-andela-way%2Fdesigning-a-three-tier-architecture-in-aws-e5c24671f124&data=04%7C01%7Ck.saripalli%40northeastern.edu%7C7a7eff3afe924f53167708d99c808b4b%7Ca8eec281aaa34daeac9b9a398b9215e7%7C0%7C0%7C637712898169862053%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=dXftorfu92XVpX%2F2ksPZzSzBDb484qT6j873zyRd8DI%3D&reserved=0)

**Step 1: Create your IaaS** for the Web App: For the IaaS (infrastructure) part, replicate all the steps shown above to create the VMs you need.

**Step 2: deploy a Web App** Then, deploy a basic Web app (SaaS) on this IaaS. The App can be the same as the Java eCommerce app you previously built, OR any other very simple web app you can adopt. Goal here is to deploy a basic web app on the IaaS you build in Step 1.

Explain each subtask using screenshot, code and text as needed to fully answer these questions.

**Other notes:**

Please see TA announcement regarding project submitting guidelines. A PDF report showing the results via screenshots and brief explanations are required. In addition, submit the code file as appropriate.

# Evaluation

Your work will be evaluated on how well the code, results including screenshots and explanations (text) are in conformance to the requirements above.

# Executive Summary

Part of your completed assignment submission should be an executive summary containing an

“Assignment overview” (1 paragraph, up to about 250 words) explaining what you understand to be the purpose and scope of the assignment and a “technical impression” (1–2 paragraphs, about 200–500 words) describing your experiences while carrying out the assignment. The assignment overview shows how well you understand the assignment; the technical impression section helps to determine what parts of the assignment need clarification, improvement, etc., for the future.

# Evaluation

The grade for your executive summary is based on the effort you put into the assignment overview and technical impression. In general, if you put some effort into your writing, you will receive full credit for your executive summary (provided that it is properly formatted and submitted as a plain text file).

# Project Deliverables

The following items should be archived together, e.g., placed in a .zip file or tarball file (\*.tgz or \*.tar.gz), and electronically submitted via the link is provided on the course Module page.

1. All source code files plus any additional support code you developed to answer.
2. Your executive summary and the PDF report answering all questions.