Name: Jaiswal, Devesh Date:November 10th, 2024

NYU ID: N14682845 dj1380

Course Section: Database Systems CSCI-GA 2433 Section 001

**Database Systems Project Part 2: Logical Schema Optimization and Data Collection**

Total in Points (100 points in total):

Professors Comments:

Affirmation of my Independent Effort: Devesh Jaiswal

**Project Report**

1-Data Lake Implementation folder in the zip file.

2-Project Part 2 pdf from the zip file contains the logical schema as well as more elaborate documentation which were generated from a tool I used called DBschema. These are the following optimization steps I took

Step 1: Validated the conceptual ER. So far according to my use cases from the project proposal the ER model we previously created fulfills the requirements

Step 2: Complete the attributes of the entities. This was already done in the previous part 1 of the project.

Step3:Identify unique keys: For entity CustomerAcct made acctName a unique key since usually customers have to have a unique account name. No other unique keys that I can see so far are needed in the entities created.

Step 4:Identify the primary keys: This was already done in the previous part 1 of the project

Step 5: Normalize the model: I normalized the model into 3nf form. First I ensured the model followed 1nf restrictions by going through attributes in entities and making sure each column can only contain atomic values. With this I had to make 1 change with vehicles\_allowed attribute in OfferedPoliciesVehicle entity where I broke it up into different boolean flags. Then I ensured the model follows 2nf restrictions by ensuring attributes only rely on the primary key. I didn’t have any composite primary keys so need to decompose the entities into smaller entities to eliminate partial dependencies.Finally I converted into 3nf by ensuring no non-prime attributes relied on each other and no changes had to be made for this step.

Step 6:Define relationships: This was already done in the previous part 1 of the project.

Step 7: Define cardinality: I defined the cardinality of all relationships between entities. I realized this wasn’t done in part 1 all that was identified before was if it was an identifying relationship or not. But now I have defined the cardinality constraints and whether it was identifying or not.

Step 8:Create any new entities: I created a few new entities since I realized that some attributes in different entities were repeated like in HealthClaim and VehicleClaim. So created a new entity like Claim and then derived from this entity to recreate HealthClaim and VehicleClaim. Similarly was done for Policy, OfferedPolicies, and Marketing entities. This removed redundancy.

3-Due to utilizing hybrid data when considering a reference architecture one factor is data integration. Usually we are getting data from multiple sources where some may be structured data and some may be unstructured data. So we must integrate the two for efficiency. This can be done using ETL, data API’s, data streaming, or other methods. Another factor to consider is the actual physical storage of the data which must support the different types of data as well. Here we can consider using a data lakes or data warehouse where we can store unstructured data as well and have efficient querying and analytics. Then since now we have the data stored we can actually use the data for things like prediction, business intelligence, etc. In addition, a few properties we may want in the reference architecture is scalability and flexibility to accommodate different data source and evolution of the insurance company. Furthermore since data usually contains important information security layer is another property we would need in the reference architecture to ensure data remains private.

4-I used microsoft azure cloud for the cloud platform to store the different datasets. I first created a storage account, then created a blob container to hold all the datasets, and finally uploaded all the datasets into the blob container. 