**Database Management Term Project:**

**Individual Report**

Kyle Galway

Sheridan College

DBAS 32100: Database Management

Haxhi Dvorani

August 13th, 2023

**Project Description**

For this project, our team progressed with largely the same plan as originally designed in the proposal, with minor alterations. We loaded the addresses from the master table in Talend, retrieved the postal codes from the API with a Java function and uploaded these addresses to the central repository. In Task Two we used Talend to load the donations from the CSV files as a list, merged the data streams, removed duplicate values, transformed and parsed the data fields, and uploaded the data to the central repository. In Task Three we created the schema designed using SQL code. In Task Four we created numerous functions and procedures to load and insert the data from the PRC tables into the data mart tables. We created the Task Five views with SQL, and Task Six used simple user role and task role creation and assignment.

There were a few differences between our proposal and the solution. We used a sequential ID in Talend instead of a generated ID in Oracle, as this proved easier to design and allowed Talend to use the Insert or Update action to ensure data consistency. We also did not exclusively use a cursor to retrieve the data in Task Five but instead created data objects in the ETLUser schema to return collections of the required data from the PRC tables to insert into the data mart tables. This change was done to allow for a more functional approach to software design more in line with programming performed in previous courses, but proved substantially more complicated and difficult to implement than the proposed solution. It did allow for Task Five to be split into more organized functional blocks but also required a major amount of work into ensuring that the ETLUser, PRC, and data mart schemas had appropriate access to one another’s required data objects.

**My Role**

I assumed the role of project lead and as a result, I completed a major amount of work on the more programmatic and technical side of this project. Much of the work I completed was on the Java and PL/SQL scripts used in Tasks One, Two, and Five, as well as the creation of the users in Task Six, which was required due to the unforeseen complications that arose because of sharing schema objects between users.

The work which did not fall within my responsibility included designing the process for loading the donations data from the CSV files and outputting transformed and erroneous data into CSV files, designing and creating the Star Schema tables and views for Task Three, creating the volunteer table in the PRC schema, selecting addresses to use from the PRC address table, or creating the views for Task Five.

**Difficulties**

The main difficulty that arose on my part of the project pertained to the necessity for security privileges to be granted for the PRC, datamart, and ETLUser users. I decided to use the ETLUser to complete Task Four and changed my design of the task to use procedures and functions instead of one or more code blocks with cursors. I expected this to make the code more functional and clearer, but I did not foresee the difficulties I would face in making this approach work. To return the correct data types between the procedures and functions I needed to make a single data object owned by one of the users and accessible to each. I also needed to create the procedures and functions for selecting the PRC tables and inserting them into the datamart tables, which required PRC and datamart users to own these functions and likewise share their access with the other users. In all, the bulk of my time in Task Four was spent not on completing the task, but on attempting to grant the correct permissions to each user such that they could access the data objects required to complete the task but not gain extra permissions that would violate the security rules for the database.

Were I to repeat this process, I am unsure about the approach I would use. My original plan was to use five separate PL/SQL blocks to complete Task Four, while my implementation used a single PL/SQL block which is called eight functions to run the entire task at once as a single process. A benefit was that it easily allow me to implement a rollback statement preventing partial transactions, and a commit statement to commit the process only if all operations succeeded. My solution allowed me to write the code in a manner that is more familiar to me from my previous work but also required additional security roles to be created that went beyond the scope of the project. If I was determined to only deliver exactly what was described in the project description, I would likely use my original solution instead – though I would prefer to have the time to do additional research to find if a compromise exists which allows for the benefits of the proposed and implemented solutions without the drawbacks of either.

**Improvements**

The donations data collection and consolidation process is highly flawed and has numerous problems. The leaders of the groups are responsible for collecting the donation data and uploading it to a central directory where they will be loaded by Talend in Task Two. There are an enormous number of errors which can occur because of this – the leaders must manually ensure that the format and order of these CSV files are correct, that there is no missing data, and that the data itself is not formatted incorrectly. The most egregious example of manual work required on the project is that the address fields are input as a single string instead of the fields as contained in the central repository. This required manual and error-prone work to transform the addresses back and forth between database and CSV files and introduced the highest chance for human error in the entire process.

Instead of requiring the leaders to each keep up to date with a CSV file, it would be more reliable to instead create a basic web application and donations input form to allow for front-end error checking – something which the current donations collection process has no implementation of. This would also allow the leaders to catch and potentially correct erroneous data before it goes through the data integration process. Some examples of how this could improve the reliability of the process include using a drop-down menu to select from a list of valid payment types instead of entering them manually, using a date selector field to standardize all data inputs into the same format, and entering the address details into the matching fields instead of requiring the entire address to be parsed as a string. The introduction of these checks would provide valuable redundant safeties, and greatly increase the number of successful donations retrieved by the volunteers.