**Database Group Project Report**

Sienna Truong

Camila Esguerra

Dong Nguyen

Sheridan College

Software Development and Networking Engineering

Sami Ahmed

Database management

DBAS32100

November 27 of 2024

**Description**

The team successfully completed the project tasks based on the outline in the group project document provided in class, and all results were documented with screenshots and straightforward descriptions in this report. In Part A, using Leader1's login, the team created the necessary tables in **Oracle SQL** based on the provided Excel data, ensuring proper table structures and relationships. They then used Talend Open Studio to export data from CSV files into Oracle tables, verifying the data import with SQL Developer. Additional jobs were created in Talend to display data and print customer rental details.

In Part B, using Leader2's account, the team developed PL/SQL scripts for inserting, updating, and deleting customer data, making the scripts flexible with substitution variables. They also wrote SQL queries for analytical tasks, such as listing customer names and DVD rental costs, and created PL/SQL procedures and functions for specific tasks.

In Part C, using Leader3's account, the team analyzed the DVD table to identify dimensions and facts, created a Star Schema and ER diagram, and populated the fact and dimension tables using insert and select queries. They ensured data accuracy and consistency.

**Part A – Sienna**

**Task 1**

Prompt: “Create the tables in Oracle as per data given in the excel file (GP - DBAS32100). Feel free to add any fields you think may be necessary, and feel free to change any existing fields to appropriate data (i.e. ID fields/ data types). Make sure to mark Primary key, foreign key relationships, unique constraints and check constraints wherever required, for this task make sure to observe data present in csv files. Once you have the database/tables with appropriate tables/fields/relationships ready. Create appropriate “.cvs” files for the given tables in the excel sheet if there is any change introduced by you in tables. Once ready take screenshots of table structure that you have created.”

**Execution**

All the Enums where created first to preserve data integrity across the tables and ensure that the foreign key constraints can be properly established later with other tables.

First, these three tables where created to store the possible ratings for DVDs, to store the possible statuses for customers, and to store the possible actions to be taken when a DVD is returned.

A screenshot of a computer

Description automatically generated

In the following images, we can observe how the code went through and created the Enum tables successfully.

**A screenshot of a computer program

Description automatically generated**

After this, we can now populate each Enum table with the rightful categories that are the same as the ones provided on the excel table.

A screenshot of a computer

Description automatically generated

**A screenshot of a computer program

Description automatically generated**

And finally, we use some select statements just to make sure the output went through.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**A screenshot of a computer

Description automatically generated**

After creating the Enum tables, the creation of the rest of the following tables was done as well. Including proper primary and foreign keys, as well as the same table columns from excel file.  
A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

And the rightful output as well when running the code, creating the tables.

A screenshot of a computer

Description automatically generated

After this, all left is populating all of the tables with the rightful data and run a select script to make sure it made effect. Ensuring it matches the value for the table and the information on the excel.

**Category table**

A screenshot of a computer

Description automatically generated

**Customer table**

A group of black text

Description automatically generated A close-up of a computer code

Description automatically generated

**A screenshot of a computer

Description automatically generated**

**Dvd table**

**A screenshot of a computer

Description automatically generated**

**A number and numbers on a white background

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Rental table**

**A group of words on a white background

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Task 2**

Prompt: “Using Talend Open Studio, export the data from the CSV file one by one into the respective oracle tables that you have created in the SQL developer, and later show the data from the tables in SQL developer using select statements. Take screenshots of all tables and jobs showing data is successfully populated.”

**(fill up all the rest of the tasks with your part of the report now lol)**

**Task 3**

Prompt: “Using Talend Open Studio, Create a new Job. Select any 1 table from Oracle that you have populated in last step and In that job, display the data from the oracle tables to log and take screenshots.”

Using Talend Open Studio, Create a new Job. Select any 1 table from Oracle that you have populated in last step and In that job, display the data from the oracle tables to log and take screenshots.

**Task 4**

Prompt: “Using Talend Open Studio, Create a new Job to print customer names, dvd’s that they have rented and rental cost using TMap and Joins. Make sure to display customers who have not rented any movie yet as well.”

**Part B – Camila**

**Task 5**

Prompt: “ Create PL/SQL scripts(you can use PL/SQL procedures or functions as per your choice) to support the following operations for customer table. Once done paste code for all three scripts and their testing screenshots. Remember PL/SQL scripts required not just SQL Queries.

* Insert Data
* Update Data
* Delete Data

[Make sure to use substitution variables to take input from user and test your script to insert data by inserting one record for each group member where customer name should be name of the group members in your group, then test update script by updating data for any one of you and then test delete data script by deleting any one-of yours’ record. Script should be executable as many numbers of times end user wants without any change.]”

This procedure inserts a new customer record into the Customer table. It takes various customer details as input parameters and inserts them into the table.

A screenshot of a computer

Description automatically generated

Testing the procedure after running

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**A screenshot of a computer error

Description automatically generated**

**A screenshot of a computer error

Description automatically generated**

**A screenshot of a phone number

Description automatically generated**

A screenshot of a computer

Description automatically generated

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer error

Description automatically generated**

**A screenshot of a phone number

Description automatically generated**

This is the update data procedure for the Customer table, where the user should be able to update some values like phone number or status.

A screenshot of a computer

Description automatically generated

A screenshot of a computer error

Description automatically generated

A screenshot of a computer error

Description automatically generated

A screenshot of a computer

Description automatically generated

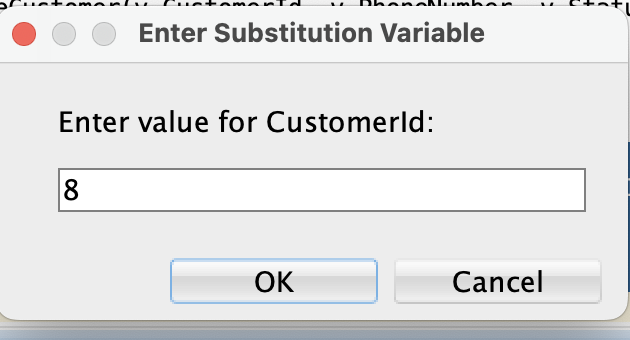
A screenshot of a computer

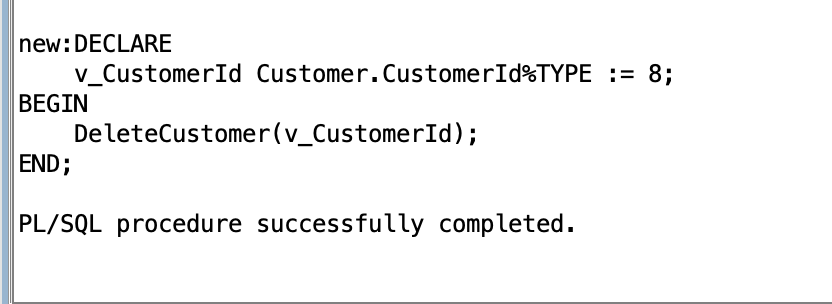
Description automatically generated

This is the delete data procedure for the Customer table, where the user should be able to delete a whole customer based on their id.

A screenshot of a computer code

Description automatically generated





A screenshot of a computer

Description automatically generated

**Task 6**

Prompt: “Write SQL Queries for following tasks as per instructions given below;-

* 1. Show the customer name and DVD name along with rental cost in list starting from which has least renting cost to highest renting cost using analytical functions?
  2. Show the titles and cost of DVDs who are with third and fourth highest price using Analytical Functions?”

**Part a**

This query lists customer names, DVD titles, and rental costs, sorted from the least to the highest rental cost using the RANK() analytical function.

A screenshot of a computer

Description automatically generated

**Part b**

This query shows the titles and costs of DVDs with the third and fourth highest prices using the ROW\_NUMBER() analytical function.

**A screenshot of a computer program

Description automatically generated**

**Task 7**

1. Prompt: “Pick any 2 questions from below of your choice and create procedures/functions in PL/SQL . You can choose any argument (of your choice) for the function/procedure.
2. Show the names of the customers, movie name and the number of movies rented by them. Make sure to show subtotals as per customer (number of movies rented) and subtotal for each movie showing how many times that movie was rented.
3. What is the difference between DVD cost and the average cost of all the DVDs of each year.”

For the first point of task 7, I divided the tasks into two procedures so it’s easier for the coding and output.

**First procedure (a)**

This procedure displays the names of customers, the movies they rented, and the number of times each movie was rented.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

**First procedure (b)**

This procedure displays subtotals of rentals per customer and per movie and per customer.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Second procedure**

This function calculates the difference between the cost of each DVD and the average cost of all DVDs for each year.

A screenshot of a computer

Description automatically generated

**Part C - Dong**

**Task 8**

Prompt: “Consider DVD table from the excel sheet. What can be the possible dimensions and facts in this table? Explain.”

**Task 9**

Prompt: “Create a Star Schema and an ER diagram (using draw.io) out of DVD table showing Star Schema, to find out the total amount of cost for each category, each year and Rating. ”

**Task 10**

Prompt: “Populate each of the tables (fact and dimension) with appropriate values from the DVD table using insert and select query, manual insertion of data will not get you any marks. Note that you need to use surrogate keys for the dimension tables and use cursor to populate the values in the fact table without losing any information.”

Notes from the group project requirements: **[Take Screen shot of each step and paste the screen in a word file, explaining each step. Marks will be deducted for improper submission.]**