UNIVERSITY OF TECHNOLOGY, JAMAICA

FACULTY: Engineering and Computing

SCHOOL: Computing and Information Technology

Module Name: **Database Administration** Module Code: **CIT3013**

Given: **Week of February 12** Due: **Week of April 1 (Lab Time)**

Assessment Type: **Group Project** Group Size: **3 - 5**

Objective:

Students are required to develop an Oracle Database Package to support the Data access layer of a Core Banking application. The package will also work as a user management utility that will carry out several DBA functions. The requirements of the utility are listed below.

Method of development:

This project is designed to be completed in a new group environment. Each group should have 3 – 5 members. **Your lab tutor must approve the group**. Within each group one leader must be assigned. The Leader is responsible for the development of the overall project and the coordination between all parts of the project developed by other members.

Section A - Database Application Requirements

---

1. Develop a database application simulating a core banking system. The database should be placed in an application schema to which users will be given access. The database should use indexes for frequently queried fields and joins. (**Marks will be deducted if the database is not in Third Normal Form.)** **[25 Marks]**

Entities:

- **Customer:** ID Number, Name (First and Last), DOB, Gender, Mobile, Price Plan, Customer Status (Active, Suspended, Inactive), Address, Tax registration Number (TRN)
- **Account:** Account Number**,** Account Name, Account Type (Savings, Checking, Deposit), Available Balance, Actual Balance, Minimum Allowed Balance, Daily Debit (Withdrawal) Limit, Daily Credit (Deposit) Limit, Account Status

(Active, Pending, Dormant, Frozen, Closed)

- **Card:** Card Number, Customer ID, PIN (must be encrypted/hashed), Card Status (Active, Deactive, Blocked, Pending)
- **Transaction:** Transaction ID, Transaction Amount, Transaction Type (Deposit/Withdrawal/Transfer), Account Number, Other Account Number, Transaction Date, Customer ID, Card ID, User ID, Approved By, Transaction Status (Pending, Completed, Rejected)
- **User:** User ID, User type (Teller/Supervisor/CSR), Transaction Limit, ReportsTo (also a User)

Business Rules:

The following Business rules must be implemented in your database as relationships, constraints, Stored Procedures or triggers as appropriate:

- Customer can have multiple Accounts
- Account can have multiple Customers
- Each Customer can have maximum of one Card
- Each Card can be owned by only one Customer
- Each Card can be linked to one, two or three accounts
- Account may not exceed Daily Transaction (Debit/Credit) Limit
- Only Checking Accounts may have minimum balance less than 0 (i.e. Overdraft)
- Account may NOT go below Minimum Balance
- Tellers can only conduct Transactions (i.e. Deposits and Withdrawals) as well as Balance Inquiries
- Tellers should NOT be able to create, modify or delete Account, Customer or Card details
- Transactions can only be done against Active accounts
- All Transactions require the use of a Card and correct PIN
- Only Transactions that exceed a Teller's limit would require approval by their Supervisor– if this is below the limit the Teller themselves should be logged as the approver
- CSRs should only be able to create and modify Accounts, Cards, Customers
- CSRs should NOT be able to conduct Transactions
- All modifications to Customer, Accounts or Cards must be authorized by Supervisor
- Supervisor may NOT initiate any Transaction, only approve or reject.
    - When a Transaction is approved, it should be updated
    - If a Transaction is rejected it must be deleted from the Transaction table and appropriately logged in another table.
- Supervisor may NOT initiate the creation or modification of a Customer, Account or Card, they may only approve or reject
- The system should not allow a User to delete Account, Card, Customer or Transactions (for obvious reasons!). The status must be updated to the appropriate value.

1. The database should use a partitioning scheme for your Transaction table by Transaction Date for the month. The Database should automatically create new partitions. **[10 Marks]**

2. Three roles should be implemented that achieves responsibilities for TELLER, CSR, SUPERVISOR (You will also need an OPERATOR role for user who can create and modify other users). Ensure that users have been created to demonstrate each. **[15 Marks]**

3. Stored procedures should be developed for each table in the application that handles ALL DML statements factoring the Business Rules listed on Page 2. **[40 Marks]**

4. Triggers should be used to implement an audit trail for the tables, which holds the most essential data for your application. This should be an update trigger and the past values for that record should be stored along with the user who made the changes. (Any Audit table that is created should have a primary key) **[15 Marks]**

5. Tracking Logons from Various Locations - The database should be able to track and identify when a username as has connected to the database from different machines. For security purposes the utility should be able to identify the username, operating system user name, machine from where the user is logged on and how long they are logged on. Custom tables will need to be designed and implemented for this requirement. **[15 Marks]**

6. Implement a fine grain auditing policy that tracks persons searching for Accounts with balance over $500,000. Additionally, demonstrate that the audit was recorded. **[15 Marks]**

7. Implement a view of your choice for security on any of the main tables e.g. Pending Transactions – and demonstrate a Supervisor having modification privileges can use this.

[10 Marks]

Section B - Package Requirements: (70 Marks)

1
.Managing Users **[20 Marks]**

a. Creating Single User – The utility should be able to accept a user name and role for which the following will be done. (The role must be one that was implemented for the database in Section A)
   - Give a default Table Space
   - Give a 1 Megabyte Quota
   - Password word of Pass1234, which the user can change upon log on.

b. Creating A Set of Users - Using the rules in requirements 1a users should be created from reading usernames and roles from a flat file.

c. Deleting a User (by accepting a username)

d. Deleting a set of Users (by accepting a flat file with usernames only)

2. Exporting Data – The Utility should have a procedure that will be able to export Transaction Records for a single Account or for all Accounts for a particular date range and produce flat files in CSV format. A separate file should be generated for each Account **[20 Marks]**

3. Managing Locked Accounts – The utility should be able to identify each locked account that was locked because of invalid login attempts. The utility should further unlock accounts that have been locked for more than a week. **[10 Marks]**

4. The utility should have a procedure that marks customer Accounts as Dormant if there are no transactions for the last three (3) months. **[20 Marks]**

Section C – Submission Rules and Requirements:

---

1. Projects are due in the Lab Time of the group Leader.
2. The project must have a project document outlining the scenario of the project and the ERD diagram that will be used in the Application Schema. The project document should further have a detailed paragraph for each task explaining what your implementation of the task in trying to accomplish along with an explanation of any decisions taken **(E.g. Which partitioning method you chose and why for your specific scenario. You should highlight why other techniques were not chosen.)** The project document should include the statements use to implement each task. Marks will only be given for tasks that are commented properly. An example of the execution statement should also be included.
3. Each Group must have a task list with who is responsible for which aspect of the Project.
4. Marks will not be awarded for section B if there is no Oracle Package Present.
5. **Each group must submit the project in a compressed file via Google classroom and email (sscale.utech@gmail.com) by the due date as stated above.**
6. If the project is submitted to lecturer after the due day the evaluation interview will still be conducted, but a penalty is attracted for every late day. **Each day after due day (including Saturdays and Sundays) will attract a 10% accumulative penalty which will be deducted from the total mark of project. No Project will be accepted one week after the due date has passed.**
7. Presentation of the final project will be based on "face to face" discussion individually with each member of group.
8. If the presentation time is missed, a written letter must be provided outlining the reasons for missing the appointed time along with a request for another presentation time.
9. Each member of the group should be responsible for a task of project.
10. It is not acceptable for a student to know only their part of project and have no knowledge about other parts or how their part integrates with the project.