Database Administration Mini-Project

3rd year Computer Science (SCI)

2023/2024

Important Notes :

- The work must be done in pairs (binômes) or singles (monôme).
- The project must be carried out under Oracle Database Express Edition (ODXE) 11g.
- Any ODXE-compatible client can be used.
- Give all the SQL queries that enabled the completion of the requested tasks.
- The examiner must be able to connect to the server and run queries from another machine on the same network. The remote connection should be tested with some requests.

Part 1: Database objects creation (DDL)

1. Create a new Tablespace called `LibraryDataTBS`, with a datafile named `LibraryDataDF.dbf` having a size of 2 MB.
2. Create a user named `LibraryAdmin` and assign him the LibraryDataTBS tablespace as a default tablespace.
3. Give the user an unlimited quota in the `USERS` tablespace.
4. Grant the `LibraryAdmin` user the necessary privileges to perform all the following database operations (using existing database Roles if there are any).

Bonus: create a user that is associated with the connected OS user, and test the connection

We want to design a system for library management.

1. On the `LibraryAdmin` schema create the tables `Books`, `Authors`, `Borrowers`, `Transactions`, and `Genres`, as described in this file. Add appropriate primary key and foreign key constraints to all tables. `Transactions` table should be created in the `USERS` tablespace.
2. Implement a unique constraint on Book ISBNs.
3. The library administrator wants to keep track of the total number of transactions for each book. Add the `Transaction_Count` column to the Books table.
4. Create a view named `PopularBooks` that displays books with more than 10 transactions.

5. Create a view named `RecentFictionBooks` that displays books of the "*Fiction*" genre that have been written in the last 3 years.
6. The database administrator wants to know the owner of the table `Transactions`, how can he do it ?
7. Give the size of the `Transactions` table in Kilobytes.

Part 2: Database data manipulation (DML)

1. Retrieve the list of books published by authors who wrote at least one book whose title starts with “T”.
2. Convert all book titles to uppercase in the database.
3. Display the count and average age (rounded to two decimal places) of borrowers by their membership type.
4. List borrowers who have borrowed books written by more than one author.
5. Calculate the total overdue fine for each borrower.
6. Display the list of burrowers with the overdue duration (in days) of books they borrowed (each book separately, show its title)

Part 3: Users and security management

1. Create a profile named `LibraryProfile` with the following characteristics:
    a. Allow one session at a time.
    b. Limit a system call to a maximum of 45 seconds of CPU time.
    c. Restrict each session to 20 minutes.
    d. Limit each session to allocate a maximum of 40 KB of memory in SGA.
    e. Allow a maximum of 30 minutes of inactivity for each session.
    f. Passwords are valid for 5 days.
    g. Allow 3 connection attempts before locking the account.
    h. Implement a 1-day access ban after 4 unsuccessful connection attempts.
2. Assign the `LibraryProfile` to the `LibraryAdmin` user.
3. Grant the `hr` user the right to read from `PopularBooks` view. Verify.
4. Create the `Librarian` role that can read data from the `Book` and `Burrower` tables, and that can read and manipulate data from the `Transaction` table.
5. Create a `LibrarianUser` user and grant it the `Librarian` role.
6. Verify that the `LibrarianUser` can do the tasks assigned above.
7. What is the used percentage of the `USERS` tablespace for the `LibraryAdmin` objects ?

Part 4: PL/SQL

1. Write a function that calculates the average fine amount paid by a borrower identified by their ID.
2. Write a PL/SQL script to display the average fine amounts paid by each borrower.
8. Write a procedure that updates the value of the `Transaction_Count` column for each book according to current transactions.
3. Create a procedure to record a new book transaction. Update any necessary tables. Insert transactions for 3 different books and borrowers using this procedure.
4. Implement a trigger to ensure that the due date for a book transaction cannot be earlier than the borrow date. Display an appropriate error message for both cases.
5. Create a trigger to ensure that borrowers cannot borrow more than three books at a time.
6. Write a trigger to verify that a borrower cannot have more than 10 overdue transactions.

Part 5: Web Interface

Create a web application with the following functionalities:

1. Log-in using the `Librarian` credentials.
2. Display a list of books that have been borrowed more than 10 times (popular).

<u>Bonus:</u>

- Create a dynamic login page with a username/password form.
- Show a list of tables/views that the logged-in user can access.
- When clicking on a table/view, display all its contents (data).