Database Management Systems II (DBMS-2004)
Milestone 3 – Final Project

Milestone 3 – Stored procedures/Functions/Triggers

Deliverable 1

Create a trigger called ‘Validate_Type’ that fires before an insert or update on the RentalAgreement table is made. The trigger must check the ‘RentalType’ column. If the value to be entered on the table is either ‘M’ or ‘B’ then the trigger does not run. If the value is ‘m’ or ‘b’ then convert the value to be added to the table to uppercase. If the letter is anything else then throw an application error with a value of -20750 and a message of ‘Invalid Rental Type ? specified’, were the ‘?’ is the rental type entered on the insert or update statement.

Deliverable 2

Create a stored function called ‘Calculate_Rental_Amount’ to determine the rental cost of a book. The function is passed the Book ID. Retrieve the Book and return the rental cost. Make sure the Book ID passed in exists. If it does not, raise application error number -20850 with the error message ‘No such book’.

Deliverable 3

Write a stored procedure called ‘Add_Rental_Agreement’ to add a new Rental Agreement. The procedure will be passed the ID of the customer placing the order and the type of the rental order. The Book Rental project does not have the time or funds to code a stored procedure for movies, so if the RentalType entered is ‘m’ or ‘M’, raise the exception ‘-20950’ with the message ‘*** ERROR procedure does not support creation of movie rentals’ and ~~terminate.~~ If the RentalType is anything other than ‘M’, ‘m’, ‘B’, or ‘b’ then ~~do nothing~~ (the trigger created above catches this).

If the rental type is valid, add a new rental agreement to the table as follows:

1. The AgreementID is the highest agreement value on the RentalAgreement table plus 1;
2. The Customer ID is the value passed in;
3. The Agreement Date is today (the date the stored procedure runs);
4. The ItemCount is null;
5. The Duration ID is always 4 (7 days);
6. The RentalType is the value passed in.

Since this procedure is part of a larger transaction, DO NOT commit changes.

Deliverable 4

Write a stored procedure called ‘Add_Book_Rented’ to add a new Book Rental record. The procedure will be passed the ID of the book rented, the Agreement ID this rental is associated with and the date of the rental.

Add a new Book Rented record to the table as follows:

1. The BookID is the value passed in
2. The AgreementID is the value passed in
3. Use the function created above to determine the rental amount for the book rented
4. The Rental Expiry is 7 days after the date the book was rented (which is passed in)

Since this procedure is part of a larger transaction, DO NOT commit changes.

Deliverables

A single .zip archive named ‘*yourlastname,firstinitial.zip’*. This zip file contains the SQL file ‘Processing.sql’ that contains the statements to create or replace the following:

- The trigger ‘Validate_Type’
- The stored Function ‘Calculate_Rental_Amount’
- The stored procedure ‘Add_Rental_Agreement’
- The stored procedure ‘Add_Book_Rented’

IMPORTANT: If the Trigger is not codes correctly it can affect how your procedures work. It is recommended that you TEST the trigger before proceeding to the remaining deliverables. Use several INSERT statements passing a RentalType of 'M', 'm', 'B', 'b'. These should allow the insert. Next INSERT another character that should cause the exception message and number to display.

Your schema with **clean** compiles of all the above objects. Any deliverables that compile **with errors** will NOT be marked.
It is recommended to include the code **SHOW ERRORS;** after each trigger, function and procedure to help you debug.

WARNING: You may only submit this milestone deliverable once, so make sure you are satisfied with what you have done before submitting. Submitting in error is NOT a valid excuse.

Submitting to the drop box will automatically release the next milestone deliverable(s) to you. You must submit *something* to receive the next milestone deliverable(s).