Database Systems
Lab Test Comprehensive Exam - 2
16th April, 2023

Maximum marks 30 **Maximum time: 30 Minutes**

Note: *The database schema is not normalized. The instance provided to you might not follow all the constraints. You are expected to submit the SQL queries only.*

Q1. List the top 5 products with the highest quantity sold, including product name, product_id, and total quantity sold. **[2]**

Q2. Write SQL queries to do the following tasks. **[2+2+2+2+2= 10 Marks]**

a) Find the name, category, and price of the product that has the most reviews.
b) Count the number of reviews for the product with the most reviews.
c) List the names of users who reviewed the product with the highest ratings.
d) List the name of the user, user_id, category of the product, and quantity ordered who placed the highest value order.
e) Find the name of the user who has ordered products from all categories.

Q3. Write SQL queries to do the following tasks. **[3+3 = 6 marks]**

a) Create a stored procedure, 'Get_User_Order_Details', to print a user-wise list of orders. The user inputs the user's ID at runtime. Also, write the test query to call that procedure with user_id_input= 1.
b) Create a stored function, 'get_order_total', to show the total amount for each order, along with the order_id. Also, write the test query to call that function to show the total amount.

Q4. Write SQL queries to do the following tasks. **[4+4+4 = 12 Marks]**

a) Create a trigger, 'stock_log', to maintain the logs of stock changes in a separate table, stock_log<product_id, old_stock, new_stock, update_time>. Write a query for creating the table, creating the trigger, and a test query to check the trigger's functionality. Test functionality will update the database whenever the stock is updated.
b) Write a stored procedure, 'fetch_category_order_details', to fetch order details of a specific category and store these details in a separate table, category_order_details<order_id, product_id, category_id, quantity_ordered, price_each>. The user passes the category_id as an input parameter. Write a query for creating the procedure, creating the table (category_order_details). Call the procedure (fetch_category_order_detail) with category_id_input = 3
c) Write SQL queries to list product name, product_id, and total quantity sold. The column total quantity sold should be generated using a stored function.