EASTERN INTERNATIONAL UNIVERSITY

SCHOOL OF COMPUTING

AND INFORMATION TECHNOLOGY

Practice Assignment 7

Practice Assignment – Quarter 4, 2023-2024

Course Name: Database

Course Code: CSE 301

Student's Full Name:

Student ID:

***Instruction*:**

** Students are allowing to write their answers (like SQL queries, Screen shot of outputs, etc.) in word file (Answer sheet) provided by instructor. After finishing the assignment, students must convert the word file (Answer sheet) into a PDF file. Finally, students upload the file in Moodle.*

1. Create the following tables in a new database 'Assignment3':

Clients(**Client_Number**, Client_Name, Address, City, Pincode, Province, Amount_Paid, Amount_Due)

Product(**Product_Number**, Product_Name, Quantity_On_Hand, Quantity_Sell, Sell_Price, Cost_Price)

Salesman (**Salesman_Number**, Salesman _Name, Address, City, Pincode, Province, Salary, Sales_Target, Target_Achieve, Phone)

Salesorder(**Order_Number**, Order_Date**, Client_Number, Salesman_Number**, Delivery_Status, Delivery_Date, Order_Status)

Salesorderdetails(**Order_Number, Product_Number**, Order_Quantity)

a) SQL UNION

Syntax:

SELECT *column_name(s)* FROM *table1*

UNION

SELECT *column_name(s)* FROM *table2*;

The UNION operator selects only distinct values by default. To allow duplicate values, use UNION ALL:

SELECT *column_name(s)* FROM *table1*

UNION ALL

SELECT *column_name(s)* FROM *table2*;

1. SQL statement returns the cities (only distinct values) from both the "Clients" and the "salesman" table.
2. SQL statement returns the cities (duplicate values also) both the "Clients" and the "salesman" table.
3. SQL statement returns the Ho Chi Minh cities (only distinct values) from the "Clients" and the "salesman" table.
4. SQL statement returns the Ho Chi Minh cities (duplicate values also) from the "Clients" and the "salesman" table.
5. SQL statement lists all Clients and salesman.
6. Write a SQL query to find all salesman and clients located in the city of Ha Noi on a table with information: ID, Name, City and Type.
7. Write a SQL query to find those salesman and clients who have placed more than one order. Return ID, name and order by ID.
8. Retrieve Name, Order Number (order by order number) and Type of client or salesman with the client names who placed orders and the salesman names who processed those orders.
9. Write a SQL query to create a union of two queries that shows the salesman, cities, and target_Achieved of all salesmen. Those with a target of 60 or greater will have the words 'High Achieved', while the others will have the words 'Low Achieved'.
10. Write query to creates lists all products (Product_Number AS ID, Product_Name AS Name, Quantity_On_Hand AS Quantity) and their stock status. Products with a positive quantity in stock are labeled as 'More 5 pieces in Stock'. Products with zero quantity are labeled as ‘Less 5 pieces in Stock'.

b) STORE PROCEDURES

Statements:

1. Create a procedure stores

```
Delimiter $$
CREATE PROCEDURE sp_name () .../
CREATE PROCEDURE sp_name ([IN] param_name type).../
CREATE PROCEDURE sp_name ([OUT] param_name type).../
CREATE PROCEDURE sp_name ([INOUT] param_name type)...
Begin
        Select statements;
End$$
Delimiter ;
```

2. Call a procedure stores
   Call *name_store* (*value of parameter if have*);
3. Drop a procedure stores
   Drop procedure *name_store*;

11. Create a procedure stores get_clients _by_city () saves the all Clients in table. Then Call procedure stores.
12. Drop get_clients _by_city () procedure stores.

13. Create a stored procedure to update the delivery status for a given order number. Change value delivery status of order number “O20006” and “O20008” to “On Way”.
14. Create a stored procedure to retrieve the total quantity for each product.
15. Create a stored procedure to update the remarks for a specific salesman.
16. Create a procedure stores find_clients() saves all of clients and can call each client by client_number.
17. Create a procedure stores salary_salesman() saves all of clients (salesman_number, salesman_name, salary) having salary >15000. Then execute the first 2 rows and the first 4 rows from the salesman table.
18. Procedure MySQL MAX() function retrieves maximum salary from MAX_SALARY of salary table.
19. Create a procedure stores execute finding amount of order_status by values order status of salesorder table.
20. Create a stored procedure to calculate and update the discount rate for orders.
21. Count the number of salesman with following conditions : SALARY < 20000; SALARY > 20000; SALARY = 20000.
22. Create a stored procedure to retrieve the total sales for a specific salesman.
23. Create a stored procedure to add a new product:

    Input variables: Product_Number, Product_Name, Quantity_On_Hand, Quantity_Sell, Sell_Price, Cost_Price.
24. Create a stored procedure for calculating the total order value and classification:
    - This stored procedure receives the order code (p_Order_Number) và return the total value (p_TotalValue) and order classification (p_OrderStatus).
    - Using the cursor (CURSOR) to browse all the products in the order (SalesOrderDetails).
    - LOOP/While: Browse each product and calculate the total order value.
    - CASE WHEN: Classify orders based on total value:

      Greater than or equal to 10000: "Large"

      Greater than or equal to 5000: "Midium"

      Less than 5000: "Small"