Sanjivani Rural Education Society's College of Engineering, Kopargaon
Department of Electronics and Computer Engineering
TITLE: Experiment Write-up (EW)

Subject: Database Management System Laboratory

EXPERIMENT NO:--07

TITLE: Write a code to implement User defined Functions on DB.

LEARNING OBJECTIVES:

1. To study the fundamental concepts of UDF.
2. To learn the basic issues of functions.

THEORY:

1. Introduction to User-Defined Functions (UDFs)

- Briefly explain what UDFs are: Functions created by the user in a database to perform specific tasks or calculations.
- Mention that UDFs allow for reusable logic within SQL queries.

2. Types of UDFs

- **Scalar Functions**: Return a single value, used for operations like calculations, string manipulations, etc.
- **Table-Valued Functions (TVF)**: Return a table of results, useful for queries involving sets of data.
- **Inline vs. Multi-statement Table-Valued Functions**: Explain that inline TVFs return the result set from a single SELECT statement, while multi-statement TVFs can have multiple SQL statements.

3. Advantages of UDFs

- **Code Reusability**: Encapsulate frequently used logic or calculations.
- **Modularity**: Break down complex queries into simpler, reusable functions.
- **Maintainability**: Centralize logic that can be updated easily, affecting all queries that use the function.
- **Improved Readability**: Using functions can make SQL queries more understandable by abstracting complex logic.

Prepared by Verified by

Mr. N. I. Bhopale Dr. B. S. Agarkar
(Subject Teacher) (HOD, Deptt. of ECE)

4. Syntax of UDFs

```
CREATE FUNCTION FunctionName (@param1 DataType)
RETURNS ReturnType
AS
BEGIN
 -- Function logic here
   RETURN @result
END
```

5.Examples of UDF Usage

```
CREATE FUNCTION CalculateDiscount(@price DECIMAL(10, 2), @discount DECIMAL(5,
2))
RETURNS DECIMAL(10, 2)
AS
BEGIN
   RETURN @price - (@price * @discount / 100)
END
```

Table-Valued Function Example: Returning rows of employees above a certain salary.

```
CREATE FUNCTION EmployeesAboveSalary(@salary INT)
RETURNS TABLE
AS
RETURN (SELECT * FROM Employees WHERE Salary > @salary)
```

6. Parameterization in UDFs

- Explain how UDFs can accept parameters to work dynamically with different inputs.
- Discuss optional parameters and default values.

7. Best Practices

- **Avoid Side-Effects**: UDFs should not modify data in the database (i.e., no INSERT, UPDATE, or DELETE operations).
- **Optimize for Performance**: Explain the performance considerations, such as execution time and indexing, particularly for table-valued functions.
- **Error Handling**: Ensure proper error checking and handling within the UDF.

8. Performance Considerations

- Scalar UDFs may sometimes have performance drawbacks in certain database systems, such as SQL Server, since they may not be fully optimized for parallelism.
- Inline TVFs tend to be faster than multi-statement TVFs because they allow the database optimizer to work more effectively.

9. Security in UDFs

- Discuss how UDFs respect the permissions and privileges set on the objects they access.
- Suggest restricting permissions for UDF creation to authorized users only.

10. Real-World Use Cases

- **Data Transformation**: Use UDFs for transforming data before loading it into reports or dashboards.
- **Custom Business Logic**: Encapsulate custom calculations (e.g., tax computations, financial formulas) inside a UDF.
- **Data Validation**: Create UDFs to validate input data before processing it.

11. Limitations of UDFs

- **No DML Support**: UDFs cannot perform INSERT, UPDATE, DELETE operations.
- **Performance Overhead**: In certain cases, using UDFs can slow down queries, especially when used excessively or inefficiently.

NOTE : Please ensure that you also add the Industrial Problem (2) in your submission/document along with the existing content.

References for Theory:

- Silberschatz A., Korth H., Sudarshan S., "Database System Concepts", MGH
- Connally T, Begg C., "Database Systems", Pearson Education
- Raghurama Krishan, “Database Management Systems”, McGrawHill
- S.K.Singh, “Database Systems : Concepts, Design and Application”, Pearson

CONCLUSION: