



LAB REPORT

CSE312: Database Management System Lab

03

Topic: **Union, Trigger, View, and Stored Procedure**

Submitted To

Shadman Rabbi

Lecturer

Department of CSE

Daffodil International University

Submitted By

Student ID: **221-15-5520**

Section: **61_J1**

Student Name: **Debasish Mondal**

Date of Assignment Distribution : 10th December, 2024

Date of Assignment Submission: 14th December, 2024

Experiment No: 05	Mapping: CO1 and CO2
Experiment Name	Union, Trigger, View, and Stored Procedure

Experiment Details:

This lab experiment demonstrates the use of advanced SQL features such as UNION, TRIGGER, VIEW, and STORED PROCEDURE. These commands allow for more sophisticated database management and processing.

- **Union** - Combines the results of two or more SELECT statements, removing duplicates.
- **Trigger** - A set of SQL statements that automatically executes in response to certain events on a table (e.g., INSERT, UPDATE, DELETE).
- **View** - A virtual table created by a query, which can simplify complex queries or provide a layer of security.
- **Stored Procedure** - A prepared SQL code that can be saved and reused to perform specific operations.

Commands Executed:

Creating Tables:

```
CREATE TABLE Students (  
    StudentID INT PRIMARY KEY,  
    Name VARCHAR(50),  
    Department VARCHAR(50)  
);
```

```
CREATE TABLE Graduates (  
    GraduateID INT PRIMARY KEY,  
    Name VARCHAR(50),  
    Department VARCHAR(50)  
);
```

Using UNION:

```
SELECT Name, Department FROM Students  
UNION  
SELECT Name, Department FROM Graduates;
```

This command combines results from Students and Graduates tables, displaying unique names and departments from both.

Creating a Trigger:

```
CREATE TRIGGER update_student_count
AFTER INSERT ON Students
FOR EACH ROW
BEGIN
    UPDATE DepartmentStats
    SET StudentCount = StudentCount + 1
    WHERE DepartmentName = NEW.Department;
END;
```

This trigger automatically updates the StudentCount in the DepartmentStats table whenever a new student is added to the Students table.

Creating a view:

```
CREATE VIEW CS_Students AS
SELECT Name
FROM Students
WHERE Department = 'Computer Science';
```

This view simplifies access to the names of students in the "Computer Science" department.

Creating a Stored Procedure:

```
CREATE PROCEDURE GetStudentDetails (IN student_id INT)
BEGIN
    SELECT * FROM Students WHERE StudentID = student_id;
END;
```

This stored procedure retrieves all details of a student based on their **StudentID**.

Obtained Output:

Command	Description	Results
UNION	Combined results of Students and Graduates tables.	Displayed unique Name and Department from both tables.
Trigger	Automatically updated StudentCount on new insert.	Updated the count in DepartmentStats after each INSERT event.
View	Created a virtual table for Computer Science students.	Displayed names of students in the CS department.
Stored Procedure	Retrieved student details by ID.	Displayed all information for a specified StudentID .

Desired Output: YES

Alternative Steps/Solution (If any):

- **Using UNION ALL:** If duplicate rows need to be preserved, UNION ALL could be used instead of UNION.
- **Conditional Triggers:** Adding conditions in the trigger could update specific columns based on other criteria.

Observation/ Comments:

- **UNION for Data Merging:**
The UNION command is useful for merging data from different tables with similar structures, especially in cases like consolidating student and graduate records.
- **Automated Actions with Triggers:**
Triggers allow for automatic actions in response to table events, reducing manual work and ensuring data integrity by synchronizing related tables.
- **Simplified Access with Views:**
Views provide an easy way to access and work with specific data subsets, enhancing security by restricting access to only the required data.
- **Reusable Stored Procedures:**
Stored procedures save time and ensure consistent results for frequently used queries, as they can be called with different parameters as needed.