Name: WAQAR RIASAT ALI Section: E Roll num: 2023F - BSE -221

LAB NO 7

Open-Ended Lab

Objective:

A Developer is assigned to develop a database system for a university that also manages enterprise projects. This **University-Enterprise Management System** will track:

- **Students** (admissions, courses, GPA)
- **Employees** (HR, IT, Projects)
- **Products and Orders** (both Local and International)
- **Boating Club** (with sailors and boats)

The system will support various SQL operations: **table creation**, **data manipulation**, **joins**, **views**, **stored procedures**, **functions**, and **triggers**, all while maintaining data integrity.

Theory:

The project includes **8 tables**, **2 views**, **1 stored procedure**, **2 user-defined functions**, and **2 triggers**

Modules and Tables:

1. Academic Management

- **Student**
  - **Columns**: Reg_Id (PK), Name, Contact_No, percentage, test_marks, aggregate, admission_status
- **Student_GPA**
  - **Columns:** Student_id (FK), Batch, Semester, GPA
- **Courses**
  - **Columns**: Course_id (PK), Course_name
- **Teachers**
  - **Columns:** t_id (PK), t_name, coordinator_id (FK – self-referencing)

- **Student_Courses:**
  - Columns: Reg_Id (FK), Course_id (FK)
- **Course_Allocations:**
  - Columns: Course_id (FK), t_id (FK)

2. Employee & Project Management

- **Employee**
  - Columns: Emp_no (PK), E_name, E_address, E_ph_no, Dept_no, Dept_name, Job_id, Salary, Hire_date, Dep_Head, DOJ
- **Attendance_Records**
  - Columns: employee_id (FK), time_in, time_out, total_time
- **Projects**
  - Columns: P_id (PK), Pro_name, Dep_id

3. Inventory & Order Management

- **Products**
  - Columns: Product_id (PK), product_name, quantity, unit_price
- **Products_Log**
  - Columns: product_id (FK), pro_quantity, log_date, action
- **Local_Orders**
  - Columns: Order_No (PK), Cust_Id, Cust_name, Product_ID, Product_name, Cust_address, branch_address, order_status
- **International_Orders**
  - Same schema as Local_Orders

4. Boating Club (Co-curricular)

- **Sailors**
  - Columns: sid (PK), sname, rating, age
- **Boats**
  - Columns: bid (PK), bname, color
- **Reserves**
  - Columns: sid (FK), bid (FK), day

Methodology:

we developed and implemented a set of **views**, **stored procedures**, **user-defined functions**, and **triggers** to handle various operations:

Views (2):

1. v_HighGPA – Students with GPA ≥ 3
2. v_BlueBoatReservations – Sailors who reserved blue boats

Stored Procedure (1):

- sp_GPAStats() – Shows each student's max, min, and average GPA

Functions (2):

- fn_SalaryBreakdown(@salary) – Calculates salary components
- fn_HighGPAStudents() – Returns IDs of students with GPA ≥ 3

Triggers (2):

- trg_InsertProduct – Logs product inserts
- trg_DeleteRestrict – Prevents deletion of Product ID = 5

Name: WAQAR RIASAT ALI Section: E Roll num: 2023F - BSE -221

Results:

Input:

```sql
create database lab8;
use lab8;

-- sub se pehle sare tables create krenge phir on pr functionality apply krenge acc to the labs ;
-- Employee Table
CREATE TABLE Employee ( Emp_no INT PRIMARY KEY,E_name VARCHAR(50),Dept_no INT,Dept_name VARCHAR(50),
Salary DECIMAL(10,2),DOJ DATE);

INSERT INTO Employee VALUES
(1, 'WAQAR', 10, 'HR', 50000, '2020-01-01'),
(2, 'ALI', 10, 'HR', 45000, '2021-02-01'),
(3, 'Umer', 20, 'IT', 60000, '2019-06-15');

-- Student Table
CREATE TABLE Student (Reg_Id INT PRIMARY KEY,Name VARCHAR(50),percentage FLOAT,test_marks FLOAT,aggregate FLOAT);
INSERT INTO Student (Reg_Id, Name, percentage, test_marks, aggregate) VALUES
(101, 'Waqar Riasat', 75.5, 80, 77.75),
(102, 'Huda Mehboob', 62.0, 58, 60.00),
(103, 'Usman sntu', 48.0, 55, 51.50),
(104, 'Nida batool', 85.5, 88, 86.75),
(105, 'Hassan molvi', 59.0, 62, 60.50);

-- Student GPA Table
CREATE TABLE Student_GPA (Student_id INT,GPA FLOAT,Semester INT,Batch VARCHAR(10));
INSERT INTO Student_GPA (Student_id, GPA, Semester, Batch) VALUES
(101, 3.2, 1, 'BSCS20'),
(101, 3.5, 2, 'BSCS20'),
(102, 2.8, 1, 'BSCS20'),
(102, 3.0, 2, 'BSCS20'),
(103, 2.4, 1, 'BSCS20'),
(103, 2.7, 2, 'BSCS20'),
(104, 3.9, 1, 'BSCS20'),
(104, 4.0, 2, 'BSCS20'),
(105, 3.0, 1, 'BSCS20'),
(105, 3.1, 2, 'BSCS20');
```

Name: WAQAR RIASAT ALI Section: E Roll num: 2023F - BSE -221

```sql
-- Ab methadology apply krenge  tables ke content pr .
-- Creating views:
-- View: Students with GPA >= 3
CREATE VIEW v_HighGPA AS
SELECT Student_id
FROM Student_GPA
GROUP BY Student_id
HAVING AVG(GPA) >= 3;

-- View: Sailors who reserved blue boats
-- Reserves table batati hai ke kis sailor ne kis boat ko kis din reserve kiya.
CREATE TABLE Sailors (sid INT PRIMARY KEY, sname VARCHAR(50), rating INT, age INT);
CREATE TABLE Boats (bid INT PRIMARY KEY, bname VARCHAR(50), color VARCHAR(20));
CREATE TABLE Reserves (sid INT, bid INT, day DATE);

INSERT INTO Sailors VALUES (1, 'Ali', 5, 22);
INSERT INTO Boats VALUES (1, 'Sea Rider', 'Blue'), (2, 'Wave Cutter', 'Red');
INSERT INTO Reserves VALUES (1, 1, '2024-05-01');

CREATE VIEW v_BlueBoatReservations AS
SELECT s.sname, r.day
FROM Sailors s
JOIN Reserves r ON s.sid = r.sid
JOIN Boats b ON r.bid = b.bid
WHERE b.color = 'Blue';
```

Name: WAQAR RIASAT ALI Section: E Roll num: 2023F - BSE -221

```sql
-- Creating procedures

-- GPA Stats Procedure
CREATE PROCEDURE sp_GPAStats()
    SELECT Student_id,
           MAX(GPA) AS Max_GPA,
           MIN(GPA) AS Min_GPA,
           AVG(GPA) AS Avg_GPA
    FROM Student_GPA
    GROUP BY Student_id;

-- Creating Functions
DELIMITER $$

CREATE FUNCTION countHighGPA()  -- students ki total counting batata hai jin ki GPA 3 ya us se zyada hai.
RETURNS INT
DETERMINISTIC  -- fixed output for fixed input.like agar  x*2 ha or x=5 ha to output obviously 10 hi ayega.
BEGIN
    DECLARE total INT;  -- stores the count.

    SELECT COUNT( Student_id)
    INTO total   -- count krke total me daldo.
    FROM Student_GPA
    WHERE GPA >= 3;
```

```sql
    RETURN total;
END$$

DELIMITER ;

-- Creating triggers

-- Products and Products_Log Tables
CREATE TABLE Products (Product_id INT PRIMARY KEY,product_name VARCHAR(50),quantity INT,unit_price DECIMAL(10,2));

CREATE TABLE Products_Log (product_id INT,pro_quantity INT,log_date DATE,action VARCHAR(20));

-- Insert Trigger
DELIMITER $$

CREATE TRIGGER trg_InsertProduct
AFTER INSERT ON Products
FOR EACH ROW
    INSERT INTO Products_Log (product_id, pro_quantity, log_date, action)
    VALUES (NEW.Product_id, NEW.quantity, CURDATE(), 'INSERT');
DELIMITER ;

-- Delete Restriction Trigger
DELIMITER $$;
```

```
-- Delete Restriction Trigger
DELIMITER $$;

CREATE TRIGGER trg_DeleteRestrict
ON Products
INSTEAD OF DELETE
AS
BEGIN
    IF EXISTS (SELECT * FROM deleted WHERE Product_id = 5)
        PRINT 'Deletion of Product ID 5 is not allowed';
    ELSE
        DELETE FROM Products WHERE Product_id IN (SELECT Product_id FROM deleted);
END;
DELIMITER ;
```

Output:

```
SELECT * FROM v_HighGPA ;
```

Result Grid | Filter Rows:

Student_id
101
104
105

```
INSERT INTO Sailors VALUES (1, 'Ali', 5, 22);
INSERT INTO Boats VALUES (1, 'Sea Rider', 'Blue'), (2, 'Wave Cutter', 'Red');
INSERT INTO Reserves VALUES (1, 1, '2024-05-01');

SELECT * FROM v_BlueBoatReservations;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

sname	day
Ali	2024-05-01

Name: WAQAR RIASAT ALI Section: E Roll num: 2023F - BSE -221

```
CALL sp_GPAStats();
```

Result Grid | Filter Rows: | Export: | Wra

Student_id	Max_GPA	Min_GPA	Avg_GPA
101	3.5	3.2	3.350000023841858
102	3	2.8	2.899999976158142
103	2.7	2.4	2.5500000715255737
104	4	3.9	3.950000047683716
105	3.1	3	3.049999952316284

```
SELECT countHighGPA();
```

Result Grid | Filter Rows:

countHighGPA()
7

```
INSERT INTO Products VALUES (101, 'Mouse', 50, 700.00);
SELECT * FROM Products_Log;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_id	pro_quantity	log_date	action
101	50	2025-05-18	INSERT