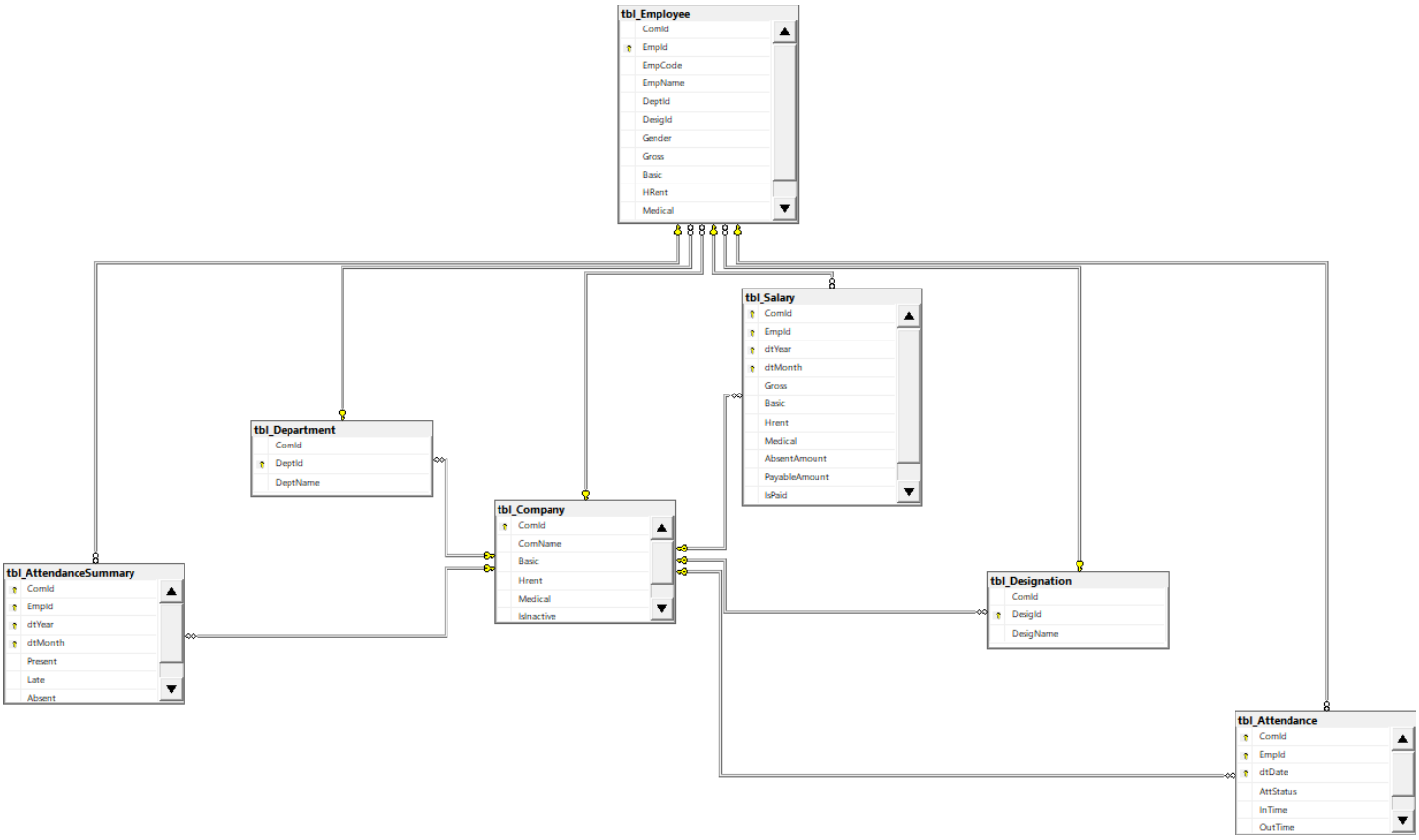
Team Members:

1. Shahinur Begum
2. MD Minhajul Islam Mahin
3. Ashibur Rahman Munna



Database:

```
CREATE DATABASE GTHRTraining;
```

Select Statements:

```
SELECT * FROM Company

SELECT * FROM Department

SELECT * FROM Designation

SELECT * FROM Employee

SELECT * FROM Attendance

SELECT * FROM AttendanceSummary

SELECT * FROM Salary
```

Company Table:

```
CREATE TABLE Company (
  ComId INT PRIMARY KEY,
  ComName VARCHAR(255),
  Basic DECIMAL(10, 2),
  Hrent DECIMAL(10, 2),
  Medical DECIMAL(10, 2),
  IsInactive TINYINT
);
```

Department Table:

```
CREATE TABLE Department (
  ComId INT,
  DeptId INT PRIMARY KEY,
  DeptName VARCHAR(255),
  FOREIGN KEY (ComId) REFERENCES Company (ComId)
);
```

Designation Table

```
CREATE TABLE Designation (
  ComId INT,
  DesigId INT PRIMARY KEY,
  DesigName VARCHAR(255),
  FOREIGN KEY (ComId) REFERENCES Company (ComId)
);
```

Employee Table:

```
CREATE TABLE Employee (
  ComId INT,
  EmpId INT PRIMARY KEY,
  EmpCode VARCHAR(255),
  EmpName VARCHAR(255),
  DeptId INT,
  DesigId INT,
  Gender VARCHAR(10),
  Gross DECIMAL(10, 2),
  Basic DECIMAL(10, 2),
  HRent DECIMAL(10, 2),
  Medical DECIMAL(10, 2),
  Others DECIMAL(10, 2),
```

```
  CONSTRAINT CHK_Gross CHECK (Gross >= 10000),
  FOREIGN KEY (ComId) REFERENCES Company (ComId),
  FOREIGN KEY (DeptId) REFERENCES Department (DeptId),
  FOREIGN KEY (DesigId) REFERENCES Designation (DesigId)
);
```

Attendance Table:

```
CREATE TABLE Attendance (
  ComId INT,
  EmpId INT,
  dtDate DATE,
  AttStatus VARCHAR(50),
  InTime TIME,
  OutTime TIME,
  PRIMARY KEY (ComId, EmpId, dtDate),
  FOREIGN KEY (ComId) REFERENCES Company (ComId),
  FOREIGN KEY (EmpId) REFERENCES Employee (EmpId)
);
```

AttendanceSummary Table:

```
CREATE TABLE AttendanceSummary (
  ComId INT,
  EmpId INT,
  dtYear INT,
  dtMonth INT,
  Present INT,
  Late INT,
  Absent INT,
  PRIMARY KEY (ComId, EmpId, dtYear, dtMonth),
  FOREIGN KEY (ComId) REFERENCES Company (ComId),
  FOREIGN KEY (EmpId) REFERENCES Employee (EmpId)
);
```

Salary Table:

```
CREATE TABLE Salary (
  ComId INT,
  EmpId INT,
  dtYear INT,
  dtMonth INT,
  Gross DECIMAL(10, 2),
  Basic DECIMAL(10, 2),
  Hrent DECIMAL(10, 2),
  Medical DECIMAL(10, 2),
  AbsentAmount DECIMAL(10, 2),
  PayableAmount DECIMAL(10, 2),
  IsPaid BIT,
  PaidAmount DECIMAL(10, 2),
  PRIMARY KEY (ComId, EmpId, dtYear, dtMonth),
  FOREIGN KEY (ComId) REFERENCES Company (ComId),
  FOREIGN KEY (EmpId) REFERENCES Employee (EmpId)
);
```

Triggers

Get Salary:

```
ALTER TRIGGER [dbo].[set_salary]
ON [dbo].[Employee]
AFTER INSERT, UPDATE
AS
BEGIN
  DECLARE @basic DECIMAL(10, 2);
  DECLARE @hRent DECIMAL(10, 2);
  DECLARE @medical DECIMAL(10, 2);
  DECLARE @others DECIMAL(10, 2);

  SELECT @basic = Basic / 100, @hRent = Hrent / 100, @medical = Medical / 100
  FROM Company
  WHERE ComId =  (SELECT ComId FROM inserted);

  UPDATE Employee
  SET Basic = @basic * Gross,
      HRent = @hRent * Gross,
      Medical = @medical * Gross,
      Others = Gross-(Basic+HRent+Medical)
  WHERE EmpId IN (SELECT EmpId FROM inserted);
    UPDATE Employee
  SET  Others = Gross-(Basic+HRent+Medical)
  WHERE EmpId IN (SELECT EmpId FROM inserted);
END;
```

Salary Payrol:

```
CREATE trigger [dbo].[tri_salaryPayrol]
on Salary
after insert, update
as
begin
declare @gross decimal(10,2),
@Basic decimal(10,2),
@Hrent decimal(10,2),
@Med decimal(10,2),
@ttlAbs int,
@AbsAmnt decimal(10,2),
@payableAmnt decimal(10,2),
@IsPaid bit
select @gross=Gross,@Basic=Basic,@Hrent=HRent,@Med=Medical from  Employee where ComId=(select ComId
from inserted) and EmpId=(select EmpId from inserted);
set @ttlAbs= (select Absent from  AttendanceSummary where ComId=(select ComId from inserted) and
EmpId=(select EmpId from inserted)and dtYear=(select dtYear from inserted) and dtMonth=(select dtMonth
from inserted) ) ;
set @IsPaid=(select IsPaid from inserted)
set @AbsAmnt= (@Basic/30) *@ttlAbs
print @AbsAmnt
if(@IsPaid=1)
begin
       update  Salary
       set Gross=@gross, Basic=@Basic, Hrent=@Hrent,Medical=@Med,AbsentAmount=@AbsAmnt where
ComId=(select ComId from inserted) and EmpId=(select EmpId from inserted)and dtYear=(select dtYear from
inserted) and dtMonth=(select dtMonth from inserted)
       UPDATE  Salary
SET PayableAmount = CASE
        WHEN @AbsAmnt IS NULL THEN Gross
```

```
        ELSE Gross - @AbsAmnt
    END,
    PaidAmount = CASE
        WHEN @AbsAmnt IS NULL THEN Gross
        ELSE Gross - @AbsAmnt
    END
WHERE
    ComId = (SELECT ComId FROM inserted)
    AND EmpId = (SELECT EmpId FROM inserted)
    AND dtYear = (SELECT dtYear FROM inserted)
    AND dtMonth = (SELECT dtMonth FROM inserted);
end
else
begin
       update  Salary
       set Gross=@gross, Basic=@Basic, Hrent=@Hrent,Medical=@Med,AbsentAmount=@AbsAmnt where
ComId=(select ComId from inserted) and EmpId=(select EmpId from inserted)and dtYear=(select dtYear from
inserted) and dtMonth=(select dtMonth from inserted)
       UPDATE  Salary
SET PayableAmount = CASE
        WHEN @AbsAmnt IS NULL THEN Gross
        ELSE Gross - @AbsAmnt
    END,
    PaidAmount = CASE
        WHEN @AbsAmnt IS NULL THEN Gross
        ELSE Gross - @AbsAmnt
    END
WHERE
    ComId = (SELECT ComId FROM inserted)
    AND EmpId = (SELECT EmpId FROM inserted)
    AND dtYear = (SELECT dtYear FROM inserted)
    AND dtMonth = (SELECT dtMonth FROM inserted);

end
end
```

Stored Procedures:

Set Attendance:

```
ALTER PROCEDURE [dbo].[set_attnIN] (
  @comid INT,
  @empid INT,
  @dtdate DATE,
  @in TIME,
  @out TIME
)
AS
BEGIN
  DECLARE @dtYear INT, @dtMonth INT;

  IF (@dtdate IS NULL OR @dtdate = '')
  BEGIN
    SET @dtdate = CAST(GETDATE() AS DATE);
  END

  SET @dtYear = YEAR(@dtdate);
  SET @dtMonth = MONTH(@dtdate);
    PRINT 'dtYear: ' + CAST(@dtYear AS VARCHAR(4));
```

```
  PRINT 'dtMonth: ' + CAST(@dtMonth AS VARCHAR(2));
 IF (@in IS NULL OR @in = '' OR @out IS NULL OR @out = '')
  BEGIN
    INSERT INTO Attendance
    VALUES (@comid, @empid, @dtdate, 'A', '', '');
  END
  ELSE IF (@in <= '09:05:00')
  BEGIN
    INSERT INTO Attendance
    VALUES (@comid, @empid, @dtdate, 'p', @in, @out);
  END
  ELSE IF (@in > '09:05:00' AND @in < '11:00:00')
  BEGIN
    INSERT INTO Attendance
    VALUES (@comid, @empid, @dtdate, 'L', @in, @out);
  END
  ELSE IF (@in > '16:05:00')
  BEGIN
    UPDATE Attendance SET OutTime = @in WHERE ComId = @comid AND EmpId = @empid AND dtDate = @dtdate;
  END
  ELSE
  BEGIN
    INSERT INTO Attendance
    VALUES (@comid, @empid, @dtdate, 'A', '', '');
  END
END;
```

Execution Command:

```
-- EXEC set_attnIn 1,9,'2023-06-13','',''

-- SELECT * FROM Attendance
```

Data Entry in Employee Table:

```
ALTER PROCEDURE [dbo].[data_entry_in_employee](
       @ComId INT,
       @EmpId INT,
       @EmpCode VARCHAR(255),
       @EmpName VARCHAR(255),
       @DeptId INT,
       @DesigId INT,
       @Gender VARCHAR(10),
       @Gross DECIMAL(10, 2)
)
AS
BEGIN
DECLARE @grossMinimumLimit DECIMAL(10,2)= 10000;

IF @Gross>=@grossMinimumLimit
BEGIN
INSERT INTO Employee (ComId, EmpId, EmpCode, EmpName, DeptId, DesigId, Gender, Gross)
    VALUES (@ComId, @EmpId, @EmpCode, @EmpName, @DeptId, @DesigId, @Gender, @Gross);
END
ELSE
BEGIN
PRINT 'Please enter minimum gross=10000'
END
END
```

Execution Command:

```
--- EXEC data_entry_in_employee 2, 9, 'EE-90', 'Mahin', 2, 2, 'Male', 900000;

-- select * from Employee
```

Data Entry in Attendance Summary:

```
ALTER PROCEDURE [dbo].[InsertAttendanceSummary]
  @ComId INT,
  @EmpId INT,
  @dtYear INT,
  @dtMonth INT
AS
BEGIN
  DECLARE @dtDate DATE = CONVERT(DATE, CONVERT(VARCHAR(10), @dtYear) + '-' + RIGHT('0' +
CONVERT(VARCHAR(2), @dtMonth), 2) + '-13');

  INSERT INTO AttendanceSummary (ComId, EmpId, dtYear, dtMonth, Present, Late, Absent)
  SELECT
    @ComId AS ComId,
    @EmpId AS EmpId,
    @dtYear AS dtYear,
    @dtMonth AS dtMonth,
    COUNT(CASE WHEN AttStatus = 'Present' THEN 1 END) AS Present,
    COUNT(CASE WHEN AttStatus = 'Late' THEN 1 END) AS Late,
    COUNT(CASE WHEN AttStatus = 'Absent' THEN 1 END) AS Absent
  FROM Attendance
  WHERE ComId = @ComId
    AND EmpId = @EmpId
    AND dtDate = @dtDate;
END
```

Execution Command:

```
--- EXEC GenerateAttendanceSummary 2, 6, 2023, 6, 20, 5, 3;

-- select * from AttendanceSummary
```

Calculate Attendance Summary:

```
ALTER PROCEDURE [dbo].[CalculateAttendanceSummary] (
  @comId INT,
  @empId INT,
  @dtYear INT,
  @dtMonth INT
)
AS
BEGIN
       DECLARE @totalAbsent VARCHAR(100);
  -- Clear the table for matching ComId and EmpId
  DELETE FROM AttendanceSummary
  WHERE ComId = @comId AND EmpId = @empId;

  -- Calculate attendance summary for a specific month
  IF (@dtMonth IS NOT NULL)
  BEGIN
       SELECT @totalAbsent = SUM(CASE WHEN AttStatus = 'A' THEN 1 ELSE 0 END)
        FROM Attendance
    WHERE ComId = @comId AND EmpId = @empId
      AND YEAR(dtDate) = @dtYear
      AND MONTH(dtDate) = @dtMonth;
         PRINT 'TOTAL ABSENT '+ @totalAbsent
```

```
    INSERT INTO AttendanceSummary (ComId, EmpId, dtYear, dtMonth, Present, Late, Absent)
    SELECT
      @comId,
      @empId,
      @dtYear,
      @dtMonth,
      SUM(CASE WHEN AttStatus = 'p' THEN 1 ELSE 0 END) AS Present,
      SUM(CASE WHEN AttStatus = 'L' THEN 1 ELSE 0 END) AS Late,
      SUM(CASE WHEN AttStatus = 'A' THEN 1 ELSE 0 END) AS Absent
    FROM Attendance
    WHERE ComId = @comId AND EmpId = @empId
      AND YEAR(dtDate) = @dtYear
      AND MONTH(dtDate) = @dtMonth;
  END
  -- Calculate attendance summary for the whole year
  ELSE
  BEGIN
    INSERT INTO AttendanceSummary (ComId, EmpId, dtYear, dtMonth, Present, Late, Absent)
    SELECT
      @comId,
      @empId,
      @dtYear,
      NULL,
      SUM(CASE WHEN AttStatus = 'p' THEN 1 ELSE 0 END) AS Present,
      SUM(CASE WHEN AttStatus = 'L' THEN 1 ELSE 0 END) AS Late,
      SUM(CASE WHEN AttStatus = 'A' THEN 1 ELSE 0 END) AS Absent
    FROM Attendance
    WHERE ComId = @comId AND EmpId = @empId
      AND YEAR(dtDate) = @dtYear;
  END
END
```

Execution Command:

```
-- EXEC CalculateAttendanceSummary 1,3,'2023','06'

-- SELECT * FROM AttendanceSummary
```

Attendance Report Daily:

```
ALTER PROCEDURE [dbo].[Attendance_Report](
       @comId INT,
       @empId INT,
       @dtDate DATE
)
AS
BEGIN
IF (@empId IS NULL OR @empId = '') AND (@dtDate IS NULL OR @dtDate = '')
       BEGIN
       SELECT * FROM Attendance
       WHERE ComId=@comId
       END
--ELSE IF (@month IS NOT NULL)
--     BEGIN
--     SELECT * FROM AttendanceSummary
--     WHERE ComId=@comId AND EmpId=@empId AND  dtMonth=@month
--     END
ELSE IF (@dtDate IS NULL OR @dtDate='')
       BEGIN
       SELECT * FROM Attendance
       WHERE ComId=@comId AND EmpId=@empId
       END
--ELSE IF (@month IS NOT NULL AND @dtDate = '')
--     BEGIN
```

```
--     SELECT * FROM AttendanceSummary
--     WHERE ComId=@comId AND EmpId=@empId AND  dtMonth=@month
--     END
ELSE IF (@empId IS NULL OR @empId='')
       BEGIN
       SELECT * FROM Attendance
       WHERE ComId=@comId AND dtDate=@dtDate
       END

ELSE IF (@empId IS NOT NULL AND @dtDate IS NOT NULL)
       BEGIN
       SELECT * FROM Attendance
       WHERE ComId=@comId AND EmpId=@empId AND dtDate=@dtDate
       END
END
```

Execution Command:

```
-- exec Attendance_Report 1,'',''

-- SELECT * FROM AttendanceSummary
```

Report Attendance Monthly:

```
LTER PROCEDURE [dbo].[Report_Attendance_Monthly]
@comid int, @empid int, @deptId int, @month int, @year int
AS
BEGIN
if @deptId=0
begin
Select C.ComName,E.EmpCode,E.EmpId,E.EmpName,d.DeptName, asm.dtMonth, asm.Present, asm.Late, asm.Absent
from  Employee as e

       Inner join  Company as c on c.ComId=e.ComId
       Inner join  Department as d on d.DeptId=e.DeptId
       Inner join  AttendanceSummary as asm on asm.EmpId = e.EmpId
       Where c.ComId=e.ComId and e.ComId=@Comid and Convert(varchar, e.EmpId) like Case When @EmpId = 0
Then '%' Else Convert(varchar, @empid) End
       AND asm.dtMonth=@month
end
else
begin
       Select C.ComName,E.EmpCode,E.EmpId,E.EmpName,d.DeptName, asm.dtMonth, asm.Present, asm.Late,
asm.Absent  from  Employee as e

       Inner join  Company as c on c.ComId=e.ComId
       Inner join  Department as d on d.DeptId=e.DeptId
       Inner join  AttendanceSummary as asm on asm.EmpId = e.EmpId
       Where c.ComId=e.ComId and e.ComId=@Comid and Convert(varchar, e.EmpId) like Case When @EmpId = 0
Then '%' Else Convert(varchar, @empid) End
       AND asm.dtMonth=@month  and d.DeptId=@deptId
END
END
```

Execution Command:

```
--- exec Report_Attendance_Monthly 1,0,0,6,0
```

Employee List Report:

```
ALTER PROCEDURE [dbo].[EmployeeList](
       @comId INT,
       @deptId INT,
       @desigId INT
)
AS
DECLARE @deptIdGet VARCHAR(50);
BEGIN
IF (@deptId='' AND @desigId='')
BEGIN
SELECT * FROM Employee
WHERE ComId=@comId
END
ELSE IF (@deptId IS NOT NULL AND @desigId='')
BEGIN
SELECT * FROM Employee
WHERE ComId=@comId AND DeptId = @deptId
END
ELSE IF (@desigId IS NOT NULL AND @deptId='')
BEGIN
SELECT * FROM Employee
WHERE ComId=@comId AND DesigId = @desigId
END
ELSE
BEGIN
SELECT * FROM Employee
WHERE ComId=@comId AND DesigId = @desigId
END

END
```

Execution Command:

```
--  EXEC EmployeeList 1,'2','2'

--- select * from Employee
```

Salary Report:

```
ALTER PROCEDURE [dbo].[Salary_Report]
(
    @ComId INT,
    @EmpId INT = NULL
)
AS
BEGIN
    -- Create a temporary table to store the result
    CREATE TABLE #TempAttendance
    (
        ComId INT,
        EmpId INT,
        dtYear INT,
        dtMonth INT,
        Present INT,
        Late INT,
        Absent INT,
        EmpName VARCHAR(50),
        DeptId INT,
```

```
        DeptName VARCHAR(50),
        PayableAmount DECIMAL(18, 2),
        PaidAmount DECIMAL(18, 2)
    )

    -- Insert the desired columns into the temporary table
    INSERT INTO #TempAttendance (ComId, EmpId, dtYear, dtMonth, Present, Late, Absent, EmpName, DeptId,
DeptName, PayableAmount, PaidAmount)
    SELECT A.ComId, A.EmpId, A.dtYear, A.dtMonth, A.Present, A.Late, A.Absent, E.EmpName, E.DeptId,
D.DeptName,
           CASE WHEN S.IsPaid = 0 THEN S.PayableAmount ELSE 0 END AS PayableAmount,
           CASE WHEN S.IsPaid = 1 THEN S.PaidAmount ELSE 0 END AS PaidAmount
    FROM AttendanceSummary A
    INNER JOIN Employee E ON A.ComId = E.ComId AND A.EmpId = E.EmpId
    INNER JOIN Department D ON A.ComId = D.ComId AND E.DeptId = D.DeptId
    LEFT JOIN Salary S ON A.ComId = S.ComId AND A.EmpId = S.EmpId
    WHERE A.ComId = @ComId AND (@EmpId IS NULL OR A.EmpId = @EmpId)

    -- Select the data from the temporary table
    SELECT ComId, EmpId, dtYear, dtMonth, Present, Late, Absent, EmpName, DeptId, DeptName,
PayableAmount, PaidAmount
    FROM #TempAttendance

    -- Drop the temporary table
    DROP TABLE #TempAttendance
END
```

Execution Command:

```
-- Show all data for a specific @ComId
--- EXEC Salary_Report @ComId = 1

-- Show data for a specific @ComId and @EmpId
-- EXEC Salary_Report @ComId = 1, @EmpId = 2
```