# Phase 2: Advanced SQL Tasks

## Stored Procedures

1. **CreateTables**

   - **Description**: Create the entire database tables.

   - **Tables Involved**:

     - `Employees`: Stores employee details.
     - `Departments`: Stores department details.
     - `Projects`: Stores project details.
     - `Assignments`: Stores assignment details including start and end dates.

   - **Stored Procedure**: `CreateTables`

   - **Example**:

```
CREATE PROCEDURE CreateTables
AS
BEGIN
    -- SQL statements to create all tables
END;
```

2. **CreateConstraints**

   - **Description**: Create all constraints and relationships between tables.

   - **Tasks**:

     - Add primary keys to all tables.
     - Add foreign keys to link `Employees` to `Departments` and `Assignments` to `Employees` and `Projects`.
     - Add any necessary unique or check constraints.

   - **Stored Procedure**: `CreateConstraintsAndRelationships`

   - **Example**:

```
CREATE PROCEDURE CreateConstraintsAndRelationships
AS
BEGIN
    -- SQL statements to create constraints and relationships
END;
```

3. **InsertDataFromSourceDB**

   - **Description**: Insert data from another database (`SourceDB`) using `SELECT INTO`.

   - **Tasks**:

     - Transfer data from `SourceDB` tables to the corresponding tables in the new database.
     - Ensure data integrity during the transfer process.

   - **Stored Procedure**: `InsertDataFromAnotherDatabase`

   - **Example**:

     ```
     CREATE PROCEDURE InsertDataFromAnotherDatabase
     AS
     BEGIN
         -- SQL statements to insert data using SELECT INTO
     END;
     ```

4. **QueryTasks**

   - **Description**: Perform specific query tasks to test and manipulate data.

   - **Tasks**:

     - Write queries to retrieve, filter, and order data using T-SQL functions.
     - Use subqueries and joins to answer complex data-related questions.

   - **Stored Procedure**: `PerformQueryTasks`

   - **Example**:

     ```
     CREATE PROCEDURE PerformQueryTasks
     AS
     BEGIN
         -- SQL statements to perform specific queries
     END;
     ```

5. **CreateViews**

   - **Description**: Create views for different purposes.

   - **Tasks**:

     - Create a specific view called `EmployeeProjectAssignments` which provides detailed information about employee assignments to projects.

    - Ensure the view includes employee names, departments, project names, roles, and assignment durations.
    - Use table expressions where necessary.
  - **Stored Procedure**: `CreateViews`
  - **Example**:

```
CREATE PROCEDURE CreateViews
AS
BEGIN
    -- SQL statements to create views
END;
```

5.1. **Use Table Expressions** `OPTIONAL`

- **Example Task**: Use common table expressions (CTEs) within the stored procedures to simplify complex queries.
- **Example**:

```
WITH ExampleCTE AS (
    -- CTE query
)
SELECT * FROM ExampleCTE;
```

6. **RunAllProcedures**
   - **Description**: Create a stored procedure to run the stored procedures for creating the database tables and constraints.
   - **Tasks**:
     - Write a stored procedure that calls `CreateTables`, `CreateConstraints`, and any other necessary procedures in the correct order.
   - **Stored Procedure**: `RunAllSetupProcedures`
   - **Example**:

```
CREATE PROCEDURE RunAllSetupProcedures
AS
BEGIN
    EXEC CreateTables;
    EXEC CreateConstraintsAndRelationships;
END;
```

## Specific View: `EmployeeProjectAssignments`

**Specification:**

- **View Name**: `EmployeeProjectAssignments`
- **Columns**:
  - `EmployeeName`: Concatenation of `FirstName` and `LastName` from `Employees` table.
  - `DepartmentName`: From `Departments` table.
  - `ProjectName`: From `Projects` table.
  - `Role`: From `Assignments` table.
  - `AssignmentDuration`: Calculated field showing the duration of the assignment (difference between `EndDate` and `StartDate`).

**Pseudocode for the View:**

1. Join `Employees` table with `Departments` table on `DepartmentID`.
2. Join the result with `Assignments` table on `EmployeeID`.
3. Join the result with `Projects` table on `ProjectID`.
4. Select and format the necessary columns:
   - Concatenate `FirstName` and `LastName` as `EmployeeName`.
   - Include `DepartmentName`, `ProjectName`, and `Role`.
   - Calculate `AssignmentDuration` as the difference between `EndDate` and `StartDate`.

## Separate Script to Run Stored Procedures

**Script to Run All Setup Stored Procedures:**

This script will execute the stored procedures for creating tables and setting up constraints in the correct order.

```
USE YourDatabaseName;
GO

EXEC CreateTables;
EXEC CreateConstraintsAndRelationships;
EXEC InsertDataFromAnotherDatabase;
EXEC PerformQueryTasks;
EXEC CreateViews;
GO
```

# Deliverables

Each group will submit:

1. Scripts for each stored procedure:
   - `CreateTables`

   - `CreateConstraints`
   - `InsertDataFromSourceDB`
   - `QueryTasks`
   - `CreateViews`
   - `RunAllProcedures`

2. A separate script to run all the stored procedures in the correct order.
3. A script for the `EmployeeProjectAssignments` view.
4. Documentation explaining their approach and any challenges encountered.

This structured approach ensures that students practice critical aspects of database management, including table creation, data integrity, and complex query formulation using T-SQL.