DATABASE MANAGEMENT SYSTEM - CSA0593

ASSIGNMENT 3

K.SAI YASWANTH

192372374

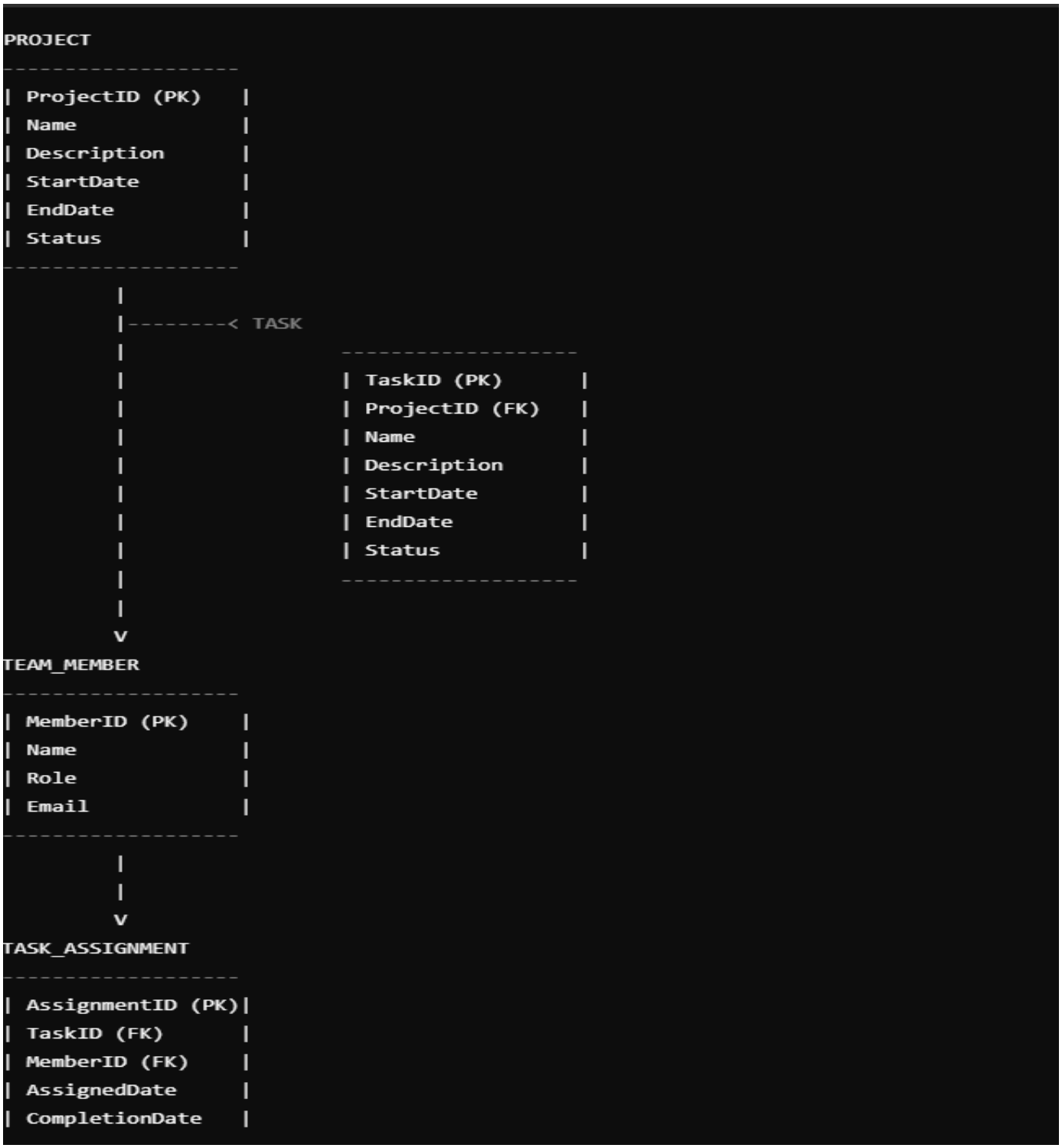
QUESTION:

Design a database schema and write SQL code for managing projects, tasks, team members, and task assignments

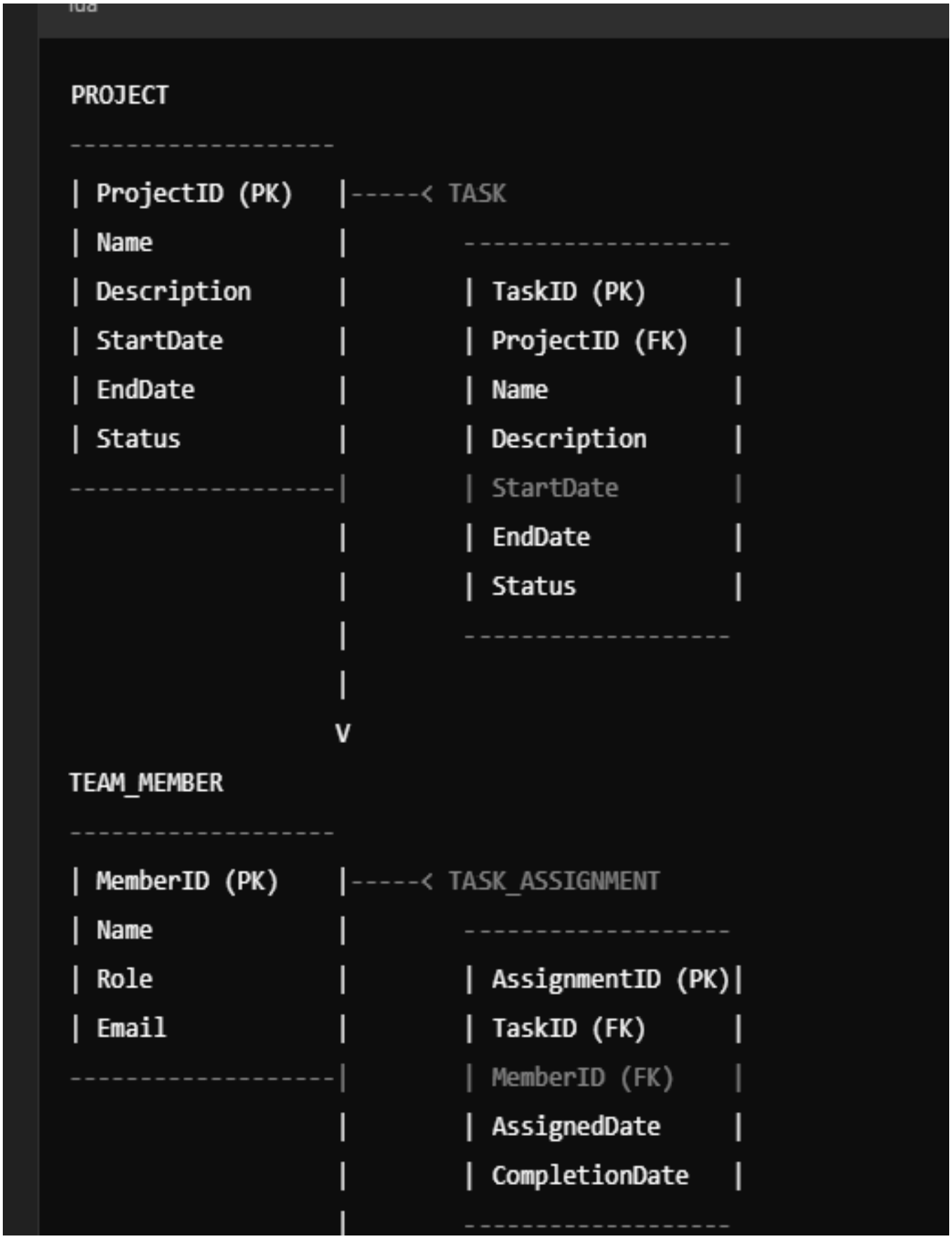
- Model tables for projects, tasks, team members, and task assignments.
- Write stored procedures for assigning tasks and tracking project progress.
- Implement triggers to update project statuses when all tasks are completed.
- Write SQL queries to analyze project completion rates and task distribution among team members.

ANSWER:

CONCEPTUAL E.R.DIAGRAM:



LOGICAL E.R.DIAGRAM:



PHYSICAL E.R.DIAGRAM:

MYSQL STATEMENTS:

```
mysql
CREATE DATABASE ProjectManagement;

USE ProjectManagement;

CREATE TABLE Projects (
  ProjectID INT AUTO_INCREMENT PRIMARY KEY,
  ProjectName VARCHAR(100),
  ProjectDescription VARCHAR(255),
  StartTime DATE,
  EndTime DATE,
  ProjectStatus VARCHAR(20)
);

CREATE TABLE Tasks (
  TaskID INT AUTO_INCREMENT PRIMARY KEY,
  ProjectID INT,
  TaskName VARCHAR(100),
  TaskDescription VARCHAR(255),
  StartTime DATE,
  EndTime DATE,
  TaskStatus VARCHAR(20),
  FOREIGN KEY (ProjectID) REFERENCES Projects(ProjectID)
);
```

```
CREATE TABLE TeamMembers (
  MemberID INT AUTO_INCREMENT PRIMARY KEY,
  MemberName VARCHAR(100),
  Email VARCHAR(100),
  Role VARCHAR(50)
);

CREATE TABLE TaskAssignments (
  AssignmentID INT AUTO_INCREMENT PRIMARY KEY,
  TaskID INT,
  MemberID INT,
  AssignmentDate DATE,
  FOREIGN KEY (TaskID) REFERENCES Tasks(TaskID),
  FOREIGN KEY (MemberID) REFERENCES TeamMembers(MemberID)
);
```

Stored Procedures:

mysql

```
DELIMITER //

CREATE PROCEDURE sp_AssignTask(
  IN taskID INT,
  IN memberID INT,
```

```
  IN assignmentDate DATE
)
BEGIN
  INSERT INTO TaskAssignments (TaskID, MemberID, AssignmentDate)
  VALUES (taskID, memberID, assignmentDate);
END //

CREATE PROCEDURE sp_UpdateTaskStatus(
  IN taskID INT,
  IN taskStatus VARCHAR(20)
)
BEGIN
  UPDATE Tasks
  SET TaskStatus = taskStatus
  WHERE TaskID = taskID;

  -- Update project status if all tasks are completed
  UPDATE Projects
  SET ProjectStatus = 'Completed'
  WHERE ProjectID = (SELECT ProjectID FROM Tasks WHERE TaskID = taskID)
  AND (SELECT COUNT(*) FROM Tasks WHERE ProjectID = ProjectID AND
TaskStatus != 'Completed') = 0;
END //

DELIMITER ;
```

Triggers:

mysql

```
DELIMITER //

CREATE TRIGGER tr_UpdateProjectStatus
AFTER UPDATE ON Tasks
FOR EACH ROW
BEGIN
  IF NEW.TaskStatus = 'Completed' THEN
    UPDATE Projects
    SET ProjectStatus = 'Completed'
    WHERE ProjectID = NEW.ProjectID
    AND (SELECT COUNT(*) FROM Tasks WHERE ProjectID = ProjectID AND
TaskStatus != 'Completed') = 0;
  END IF;
END //

DELIMITER ;
```

SQL Queries for Analysis:

mysql

```
-- Project Completion Rates
```

```
SELECT
  ProjectName,
  ProjectStatus,
  COUNT(*) AS TotalTasks,
  SUM(CASE WHEN TaskStatus = 'Completed' THEN 1 ELSE 0 END) AS
CompletedTasks
FROM
  Projects
  JOIN Tasks ON Projects.ProjectID = Tasks.ProjectID
GROUP BY
  ProjectName, ProjectStatus;


-- Task Distribution among Team Members
SELECT
  MemberName,
  COUNT(*) AS TotalTasks,
  SUM(CASE WHEN TaskStatus = 'Completed' THEN 1 ELSE 0 END) AS
CompletedTasks
FROM
  TeamMembers
  JOIN TaskAssignments ON TeamMembers.MemberID =
TaskAssignments.MemberID
  JOIN Tasks ON TaskAssignments.TaskID = Tasks.TaskID
GROUP BY
  MemberName;
```

Conclusion:

This database design provides a comprehensive foundation for managing projects, tasks, team members, and task assignments. The stored procedures simplify task assignment and project progress tracking, while the triggers ensure data consistency and accuracy. The SQL queries enable analysis of project completion rates and task distribution among team members.