# TaskMaster


Ahnaf Rais Mahi

Roll:2007054

CSE 3100 :Database Systems Laboratory

***This is*** *Oracle SQL Database Project*

*Runs on run sql command prompt*

## Introduction

**Project Overview**

TaskMaster is a basic task management tool designed to make work completion and tracking in businesses more efficient. It gives customers real-time notifications on task statuses, deadlines, and progress by storing and managing task-related data in an Oracle SQL database.

Databases are Important for Task Management Systems

Task management solutions depend on a well-structured database to efficiently organize work and promote team member collaboration. The database makes it possible for users to assign tasks, track progress, and allocate responsibilities with ease by offering a single repository for task-related data. This enhances productivity and project outcomes.

**Project Objectives**

The following goals are intended to be accomplished by the TaskMaster database project:

Create a Basic Task Management Relational Database Schema.

1. To store task information, such as the task ID, description, status, deadline, and assigned user, create a simple schema.

2. Make sure the task assignment, tracking, and updating are made simple by the schema.
   Use Simple SQL Queries to Manipulate and Retrieve Tasks

3 . Make SQL queries to create, modify, and remove tasks.

4. Create queries to get task data according to different parameters, like assigned user, deadline, and status.

Assure Performance and Data Security

To prevent unwanted access to task data, put basic security measures in place.
In order to guarantee effective query execution and system responsiveness, optimize database performance.

# Database Design

## Overview of the Database Schema

**Employees:**Stores detailed information about individual employees, such as name, email, position, salary, and supervisor ID.

**Tasks**:Contains data about tasks, including task ID, title, description, assigned employee ID, deadline, and status.

**Workdays**: Records details about workdays, such as workday ID, work date, hours worked, and employee ID.

**Timeclock:** Captures clock-in and clock-out times of employees, including clock ID, employee ID, clock-in timestamp, and clock-out timestamp.

**Supervisors:**1.Stores information about supervisors within the organization.

2.Attributes include supervisor ID, name, email, and salary.

3.The supervisor ID serves as the primary key, ensuring each supervisor has a unique identifier.

4.This table enables the tracking of supervisor details and their relationships with employees.

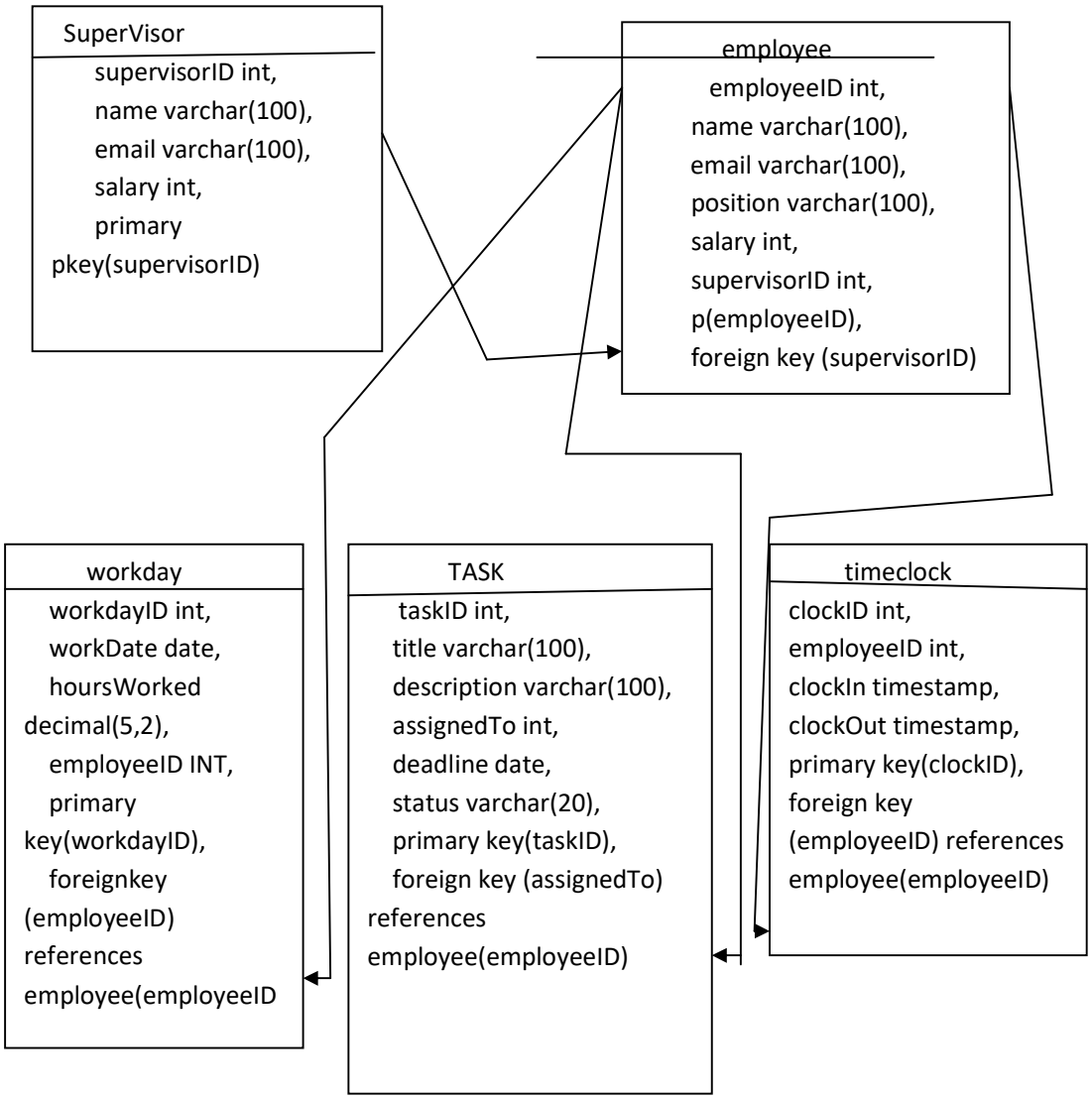
## Creating Tables

```
Create table supervisor (
      supervisorID int,
      name varchar(100),
      email varchar(100),
      salary int,
      primary key(supervisorID)
   );

Create table employee (
       employeeID int,
       name varchar(100),
       email varchar(100),
       position varchar(100),
       salary int,
       supervisorID int,
       primary key(employeeID),
       foreign key (supervisorID) references supervisor(supervisorID)
   );
Create table task (
   taskID int,
   title varchar(100),
   description varchar(100),
   assignedTo int,
   deadline date,
   status varchar(20),
   primary key(taskID),
   foreign key (assignedTo) references employee(employeeID)
);
Create table workday (
   workdayID int,
   workDate date,
   hoursWorked decimal(5,2),
   employeeID INT,
   primary key(workdayID),
   foreign key (employeeID) references employee(employeeID)
);
```

```
Create table timeclock (
    clockID int,
    employeeID int,
    clockIn timestamp,
    clockOut timestamp,
    primary key(clockID),
    foreign key (employeeID) references employee(employeeID)
);
```

# ER DIAGRAM



# SQL QUERIES

1.how to add new task to a particular employee

```
insert into task(taskID, title, description, assignedTo, deadline, status) values (5, 'task 2',
'description 2', 1, to_date('2024-05-10', 'YYYY-MM-DD'), 'pending');
```

2.how to see all tasks assigned to particular employee

```
method no 1: select * from task where assignedTo = 1;
method no 2:select * from task where assignedTo in(select employeeID from employee where
employeeID = 1);
```

3.updating task details

```
update task set title='New Title1', description='New Description1' where taskID = 1;
```

4.to see updated title,description about task details

```
select taskID,title,description from task where taskID=1;
```

5.how to mark a task completed

```
update task set status = 'completed' where taskID = 1;
select taskID,status from task where taskID=1;
```

6.View tasks with a particular status

```
SELECT * FROM task WHERE status = 'pending'
```

7.View tasks with a deadline within a certain date range

```
select * from task where deadline between to_date('2024-05-10', 'YYYY-MM-DD') and
to_date('2024-05-15', 'YYYY-MM-DD');
```

8.Search for tasks by title or description

```
method no 1:select * from task where title like '%New%' or description like '%1%';
method no 2:select t.* from task t join employee e on t.assignedTo = e.employeeID where
(e.position like '%er%' or e.salary >= 5500) and t.status = 'pending';
```

9.no of employee

```
select count(*) as total_employees from employee;
```

10.no of employee under one particular supervisor

```
select count(*) as tot_em_un_super from employee where supervisorID = 1;
```

11.no of tasks assigned to one particular employee

```
select count(*) as total_tasks_assigned from task t join employee e on t.assignedTo = e.employeeID
where e.employeeID = 2;
```

12.To retrieve the names and emails of employees along with the name of their respective supervisor under a specific supervisor

```
insert into employee(employeeID, name, email, position, salary, supervisorID) values (6, 'David
John6', 'david6@example.com', 'junior Project Manager', 6000, 4);
select e.name AS employee_name, e.email as employee_email, s.name as supervisor_name from
employee e join supervisor s on e.supervisorID = s.supervisorID where s.supervisorID = 4;
```

13.trigger for assigning default supervisor

```
CREATE OR REPLACE TRIGGER default_supervisor_trigger
BEFORE INSERT ON employee
FOR EACH ROW
BEGIN
```

```
   IF :new.supervisorID IS NULL THEN
      :new.supervisorID := 1;
   END IF;
END;
/
insert into employee(name, email, position, salary,supervisorID) values ('alice jones',
'alice@example.com', 'developer',5000);
```

## ADVANCED SQL QUERY

14.trigger for auto-increment the employeeID column,

```
CREATE OR REPLACE TRIGGER employee_id_trigger
BEFORE INSERT ON employee
FOR EACH ROW
DECLARE
   max_employee_id INT;
BEGIN

   SELECT MAX(employeeID) INTO max_employee_id FROM employee;
   IF max_employee_id IS NULL THEN
      :new.employeeID := 1;
   ELSE
      :new.employeeID := max_employee_id + 1;
   END IF;
END;
/
insert into employee(name, email, position, salary) values ('alice jones2', 'alice2@example.com',
'developer',7000);
```

15.function creating for no of employee under one distinct supervisor

```
set serveroutput on
CREATE OR REPLACE FUNCTION get_eus(p_supervisor_id INT)
RETURN INT
IS
   v_employee_count INT;
BEGIN
   SELECT COUNT(*) INTO v_employee_count
   FROM employee
   WHERE supervisorID = p_supervisor_id;

   RETURN v_employee_count;
END;
/

set serveroutput on

DECLARE
```

```
  value INT;
BEGIN
  value := get_eus(3);
  DBMS_OUTPUT.PUT_LINE('Employee Count: ' || value); -- Displaying the result
END;
/
```

15. function for marking task as completed

```
CREATE OR REPLACE PROCEDURE mark_task_as_complete(p_task_id INT)
IS
BEGIN
  UPDATE task
  SET status = 'completed'
  WHERE taskID = p_task_id;
  COMMIT;
END;
/
BEGIN
  mark_task_as_complete(1);
END;
/
```

## Targeted Customers/Users

The task management system is designed to cater to various stakeholders involved in task organization, delegation, and tracking. The primary target users include:

1.Project Managers: Project managers are responsible for overseeing project tasks, assigning them to team members, and tracking their progress. The task management system provides them with tools to efficiently allocate resources, set deadlines, and monitor task completion.

2.Team Members: Team members are tasked with executing assigned tasks and meeting project deadlines. They use the task management system to view their assigned tasks, update their status, and communicate with other team members regarding task-related matters.

3.Executives and Stakeholders: Executives and stakeholders require visibility into project progress and performance. They use the task management system to view high-level summaries, monitor key metrics, and assess project health.

4.Freelancers and Contractors: Freelancers and contractors hired for specific project tasks need access to the task management system to receive task assignments, submit deliverables, and communicate with project managers and team members.

5.IT Administrators: IT administrators are responsible for maintaining and configuring the task management system. They ensure system stability, security, and performance, as well as provide technical support to users as needed.

6.Business Analysts: Business analysts analyze task data to identify trends, bottlenecks, and areas for improvement. They use the task management system to extract relevant data, generate reports, and derive actionable insights to optimize project workflows.

7.Human Resources (HR) Personnel: HR personnel may use the task management system for workforce planning, resource allocation, and performance evaluation. They can track employee task assignments, productivity metrics, and performance feedback within the system.

8.Clients and Customers: Clients and customers may have access to the task management system to track project progress, review deliverables, and provide feedback. This transparency fosters collaboration and enhances client satisfaction.

## *Conclusion*

Conclusion

As we round off the task management system project, it's evident that we've advanced significantly.
Important elements including task creation, assignment, and tracking have all been deployed effective ly and help to improve project workflows.
Notwithstanding several challenges, such as guaranteeing data correctness and expandability, we've devised inventive methods to surmount them.
In the end, the task management system is crucial for encouraging teamwork, expediting project man agement procedures, and raising general productivity.
By offering a centralized task management platform, it significantly boosts efficiency and aids teams in successfully completing their goals.
In conclusion, the task management system is a useful instrument for businesses, providing character istics that are critical to optimizing project workflows and fostering success.