



Dhirubhai Ambani  
Institute of Information and Communication Technology

## **Task Management System**

**DBMS (IT214)**  
**Team : S10\_T2**

**Instructor : Prof. Minal Bhise**

**Mentor TA : Mayank Patel**

### **Team Member :**

Student Name	Student ID
Jemish Variya	201901112
Vin Patel	201901288
Piyars Kakadiya	201901305
Raj Patel	201901306

# **Index**

<b>Section : 1 : Final Version of SRS</b>	<b>1</b>
<b>Introduction</b>	<b>2</b>
i. Purpose	2
ii. Intended Audience and Reading Suggestions :	2
iii. Product Scope	2
iv. Problem description :	3
<b>Fact Finding Phase</b>	<b>5</b>
i. Background Reading	5
ii. Interviews :	6
iii. Questionnaires	10
iv. Requirement gathered from Responses :	17
v. Observations:	18
<b>Fact Finding Chart</b>	<b>19</b>
<b>Requirements :</b>	<b>19</b>
<b>User Classes and Characteristics :</b>	<b>20</b>
<b>Operating Environment</b>	<b>21</b>
<b>Product Functions</b>	<b>22</b>
<b>Privileges</b>	<b>23</b>
<b>Assumptions</b>	<b>24</b>
<b>Business Constraints</b>	<b>24</b>
<b>Section : 2 : Noun Analysis</b>	<b>25</b>
<b>Noun Analysis</b>	<b>26</b>
<b>Accepted Nouns and Verbs List</b>	<b>31</b>
<b>Rejected Nouns and Verbs List</b>	<b>33</b>
<b>Section : 3 : ER-Diagram all versions</b>	<b>37</b>
<b>ER Diagram V1</b>	<b>37</b>

<b>ER Diagram V2</b>	<b>39</b>
<b>ER Diagram V3(Final)</b>	<b>40</b>
<b>Section : 4 : Conversion of Final ER-Diagram to Relational Model</b>	<b>41</b>
<b>Mapping E-R Model to Relational Model</b>	<b>42</b>
<b>DDL Scripts</b>	<b>43</b>
<b>Section : 5 : Normalization and Schema Refinement</b>	<b>49</b>
<b>List all the Relations &amp; Schemas with all the details (Original Design of Database)</b>	<b>50</b>
<b>Identify and list all types of dependencies (PK, FK, Functional Dependencies) for each relation</b>	<b>51</b>
<b>Final List of all the Relations &amp; Schemas with all details (Original Design of Database)</b>	<b>59</b>
<b>Data and DDL Snapshots</b>	<b>60</b>
<b>Section : 6 : SQL Queries</b>	<b>76</b>
<b>Section : 7 : Front End</b>	<b>118</b>
<b>Home Page :</b>	<b>119</b>
<b>Functionalities</b>	<b>120</b>
For Inserting data into table :	120
For Searching Queries :	122
For any Query :	123
<b>Appendix :</b>	<b>123</b>

# **Section : 1 : Final Version of SRS**

# **Introduction**

## **i. Purpose**

The purpose of this document is to give a detailed description of the requirements for the “**Task Management System**” software. This Software Requirements Specification(SRS) will describe the processes and functions of the Task Management System. We are going to create a database for a system that helps in tracking and managing the project development job in an IT company. Progress of the entire project and all details of the project can be easily tracked by this software. Task management system is very crucial especially when a company is working on more than one project simultaneously. Apart from this, we can keep track of the contribution of each employee on each project.

- Provides an interface to managers, team members, clients, through which they may log into the system.
- Let project leaders initialize new projects.
- Let project leaders assign tasks to each team member.
- Allows project leaders to dynamically add or remove tasks from a respective project.
- Provides a system to project leaders for checking inconsistencies in process dependencies among the processes chosen for a project.
- Provides an interface to project leaders as well as team members through which they may view all of the processes for a project in the form of a dependency graph.
- Provides a facility to project leaders as well as team members through which they may update the inputs so as outputs, and the current status of a project.
- Obtain information from, and if required make changes to, the database quickly and accurately.
- Provides an intuitive and convenient GUI to maximize ease of use.

## **ii. Intended Audience and Reading Suggestions :**

### **Intended Audience :**

- Project leaders
- Clients
- Technical and Testing team
- Account and Finance department
- Research Development department
- IT Development department

### **Reading Suggestions :**

- <https://aisel.aisnet.org/cais/vol2/iss1/17/>
- <https://onlinelibrary.wiley.com/doi/abs/10.1002/pmj.20052>
- <https://www.northeastern.edu/graduate/blog/project-manager-responsibilities/>
- <https://www.smartsheet.com/content-center/best-practices/project-management/project-management-guide/project-management-IT>

## **iii. Product Scope**

The Task Management System provides an interface to both managers and workers for tracking their daily tasks. The manager interface will allow this operation, allocating new tasks, updating certain aspects of tasks, deleting tasks, and searching for tasks. The manager interface will also allow for creating and deleting users. Lastly, this

interface will have the permission to reset passwords. The worker interface will allow them to search tasks and mark tasks that they have completed. The system will also provide a history of previous tasks assigned or completed.

The scope of the project will cover the following:

1. Get the project's information from client
2. Client interaction
3. Announcements
4. communication with team members
5. Budgeting
6. Time management and approval
7. tracking current progress
8. Task distribution
9. Data sharing

#### **iv. Problem description :**

Today only accomplishing a particular task is not enough for many institutions ranging from multinational companies to local service providing groups like schools, hospitals, government bodies all they require is to perform and track the status of various tasks, of various complexities and probably requiring a team, parallelly and any inconsistency will be resolved at the earliest and for that, they need to have a robust task management system which provides the current status of the workflow for each task and higher authorities can alter the prototype in between the task.

Apart from that, there are many such aspects to be considered for the completion of a project:

- good collaboration, involvement of respective project members
- good resource sharing among various projects and project departments, query resolving
- Task management
- Deadline and related information
- Any important but immediate changes updation to all the project members
- Information of previously completed tasks for the future reference
- Prioritization of task
- Efficient workflow and work distribution
- Record of each assigned tasks
- Clarification regarding queries and doubts of employees
- Presence of announcement feature.

But unfortunately, all the aspects cannot be addressed properly without a structured system, so a Task Management System was introduced. From time to time as new requirements arose, evolution in Task management also happened.

A task management system should be designed to work in this way, with probably some suitable changes depending upon the requirements of the company which is using the Project Manager system:

- A client first approaches the Project Manager of an IT sector company for a solution of a real-world based scenario problem.
- The client puts some requirements in a request manner with certain requirements fulfilled by the proposed system as output and will discuss the overall budget and deadline of the project with the task manager.

- Thereafter the Project Manager selects an experienced employee as a project leader for the regulation of that project throughout the completion of the project. The project Manager allows certain resources to be accessed and shared, with other employees, by leaders with some extra privileges, if any.
- Then the task leader splits the entire project into certain tasks with the respective team, to do a specific task in a time-constrained manner, and allocate any particular resources, if any. He also discusses with the different teams when to get together in a meeting for discussing the progress of their corresponding task and providing guidance for any problem and probably along with the task manager.
- Each employee, belonging to a particular team, is supposed to complete his assigned task. Employees can ask a query to anyone via the “Raise Query” feature provided by the system and can see their own progress for the corresponding task.
- As soon as the respective task of the project is completed their status should be updated in the “project status” menu. After that, if that task is reviewed by the project leader, he makes the status as “approved” and if it should be improved in a specific way then he makes the status as “improve”. And if it’s not been reviewed by the project leader then that status remains “pending” until reviewed.
- The client should be able to view the summary of meetings, without having to attend the meeting itself, and the overall progress of the project throughout the entire project.
- At the end, the project leader segregates all tasks done by the respective team in a final deliverable in a specific manner, and then the Project Manager reviews that final deliverable and possibly discusses some improvements or related stuff with the leader. The Project Manager records this project and all the relevant details in the database for future reference.
- The final deliverable is given to the client.

The new computerized system has an edge over the manual system. The beneficial features of the new computerized system are the following:

**1. Improves Services and Working Conditions :** The proposed system improves the existing working conditions in each individual step of task management. User’s services are improved to a greater extent. The whole process is very easy and efficient.

**2. User Friendly :** If a particular group of users like authorities, employees, clients are able to see their relevant information like for authorities give guidance for any misunderstanding, employees can see their own progress, the client can see the current project status, on their own because the system itself is self-explanatory and don’t require specific knowledge. Thus it’s pretty much user-friendly.

**3. Responsiveness :** Another aspect of the system is its responsiveness. Any change made in the project status, the system makes sure that it will reflect as soon as possible so the client and authority can refer to consistent status.

**4. Reduced Storage Space :** One more important feature of the new system is the reduction of physical storage covered by the existing system. In the new system, records will reside in the computer’s memory, which has a large storage capacity and can be extended further.

**5. Prioritizing tasks :** The order of tasks being performed is also an important factor that needs to be considered. The system enforces real-time tasks to be performed in the order of their schedule except for the scenario when some urgent task has to be prioritized and completed regardless of the schedule.

**6. Coordination of Data Tables :** The user was in trouble while combining figures from different files or registers to find out information and to prepare reports. This trouble will not occur in the new system as the data from different tables will be combined internally and the prepared reports will be presented to the user on-screen and printer.

**7. Minimizing Errors :** The project will also eliminate the creeping in of unintentional errors due to which the system leads to insufficient. The new system will be prepared under all the restrictions of the present system. Whenever the user will attempt any error the new system will respond to it displaying a diagnostic on the screen.

**8. Data Security :** The proposed will also prevent unauthorized access to the system. At the very beginning, the user name and password are to be entered by the user and if it is correct only then the permission to use the software is given to the user. There are different users according to different roles and one user can't use the resources given to another user.

One of the most important benefits of task management is keeping track of projects while employees are on work from home mode. They can see the overall progress of the project, their own progress, can resolve their queries quickly, and refer to the summary of the meetings that they might not be able to attend.

All the improved features come with the high cost of the system. Since there are a lot more demands regarding having the responsiveness as high as possible, a system that saves the information of almost all the previously completed tasks, and the "Raise query" feature, the cost of the task management system should be reliable. That's why the task management system should have a proper balance between the number of features it provides and the cost of making the system.

## **Fact Finding Phase**

### **i. Background Reading**

In this section we are briefly summarizing whatever we have read throughout various magazines, websites, blogs and Wikipedia.

Project management involves planning, organizing, and managing resources in order to successfully meet project goals and objectives. Project management can improve the projects and the performance of the workers.

Generally, whenever we implement a new system it is developed to eliminate the shortcomings of an existing system. The computerized system has more edge over the manual system. It is because of accuracy, high speed, quick result, diligence, etc.

The existing system is totally based on a manual system. The existing system has many drawbacks in comparison to a computerized system. Some of the main drawbacks of the existing system are:

1. Time-consuming: High time consumption is the main weakness of the existing system. A lot of time is consumed in finding the progress of the project.
2. Service and working conditions: With the unusual delay due to various reasons (such as non-availability of record or absence of an employee etc) the client suffers a lot. Without any firm reason, he has to wait which disfigures the working conditions.
3. Difficulties in retrieving information : Sometimes, to get any type of instant information more than one document needs to be searched. This requires the opening of both files and then combining the information to get the required information. Further, the files may not be indexed according to the need, which causes difficulty in retrieving information.

4. Difficulties in debugging : If an error is detected in recorded information from the manual system requires many files or registers to be searched to correct the information.

5. Not easy to understand by new users : To understand the manual of the system the user has to have some experience related to all related fields and have a grasp of all fields that are helped by other employees and have to be in contact with that person. If the person was to leave the user can get the information from him.

6. Need more space to store : Here by using the paperwork system the user needs to maintain all the documents in the proper way to avoid the difficulty in action and need more space to store these documents because they can't destroy the old documents.

7. Very expensive : The existing manual system is a very expensive method as a lot of paperwork has to be done and a lot of wastage of manpower, which increases the expanses of the existing system.

8. Data sharing : The different users in the different departments cannot share data simultaneously.

#### **References :**

- [Guide To Design Database For Task Manager In MySQL](#)
- [Task Management System Database Design](#)
- [PMS Requirements Specification](#)
- <https://www.knack.com/templates/project-management>
- [Build a Project Management Application From Scratch](#)
- [A Project Management Data Model](#)

#### **Required Detail :**

- From Client-side :
  - ❖ Project details
  - ❖ Client information
  - ❖ Deadline of project
  - ❖ Budget
  - ❖ Precise software requirement
- Departments
- Team formation for various subtasks
- Accurately maintain Current activity
- Client must be able to see the current status of the project
- Provide access in a restricted manner to various groups of user
- Remote availability

#### **ii. Interviews :**

##### **Interview : 1**

###### **Role Play Interview plan**

**System:** Task Management System

**Project Reference:** SF/SJ/2021/10

**Interviewee :** 1) Piyars Kakadiya

Designation : Task Manager

**Interviewer :** 1) Jemis Variya

Designation: Software Development Engineer

2) Vin Patel

Designation: Software Development Engineer

**Date :** 24/9/2021

**Time :** 14:30

**Duration :** 30 minutes

**Mode :** Online

**Purpose of Interview :** Preliminary meeting to identify the general and case-specific requirements from the task manager's point of view.

#### **Agenda:**

- Problem with existing management system
- Any specific Requirement
- Any collaboration between different projects
- Allocation of required resources
- Actions required to be taken

#### **Documents to be brought to the interview:**

- Document related to the current situation of task management and workflow.

#### **Interview Summary 1**

**System:** Task Management System

**Project Reference:** SF/SJ/2021/10

**Interviewee :** 1) Piyars Kakadiya (Role Play)

Designation : Project leader

**Interviewer :** 1) Jemis Variya

Designation: Software Development Engineer

2) Vin Patel

Designation: Software Development Engineer

**Date :** 24/9/2021

**Time :** 14:30

**Duration :** 30 minutes

**Mode :** Online

#### **Summary of interview:**

- Preliminary meeting to identify the general and case-specific requirements from the task manager's point of view.
- An essential task of the project leader is to ensure that the project stays within the proposed budget and deadline.
- When there are large projects where goals change constantly based on client feedback, so time management, clear communication with client and management to deliver projects on time and within budget, these aspects must be followed strictly.
- Timely communication with all team members, setting remote working guidelines and weekly goals is more difficult when employees work from home.
- Task Manager should be able to recognize if there is any misunderstanding occurring in a project.
- Three different performance reports: progress report, status report and forecast report. The progress report highlights the completed tasks, while the status report focuses on the discrepancies so far.

## **Interview : 2**

### **Role Play Interview Plan**

**System:** Task Management System

**Project Reference:** SF/SJ/2021/10

**Interviewee:** 1) Jemis Variya(Role Play)

Designation: Client

**Interviewer:** 1) Raj Patel

Designation: Software Development Engineer

2) Piyars Kakadiya

Designation: Software Development Engineer

**Date :** 25/9/2021

**Time :** 14:30

**Duration :** 30 minutes

**Mode :** Online

### **Purpose of Interview:**

Preliminary meeting to collect the information regarding how the system facilitates

### **Agenda:**

- Is he concerned about attending meetings of the project team and manager?
- Responsiveness of the project team and manager
- problem with the consistency of current status of project
- Basic accesses to the system like to be able to see a project status, expected ending date, expected budget.

- Actions required to be taken

## **Interview Summary : 2**

**System:** Task Management System

**Project Reference:** SF/SJ/2021/10

**Interviewee:** 1) Jemis Variya(Role Play)

Designation : Client

**Interviewer:** 1) Raj Patel

Designation : Software Development Engineer

2) Piyars Kakadiya

Designation : Software Development Engineer

**Date :** 25/9/2021

**Time :** 14:30

**Duration :** 30 minutes

**Mode :** Online

### **Summary of interview:**

- Preliminary meeting to collect the information regarding what he expects from the IT company as a result and in between coordination with the project leader and case-specific requirements from client's point of view.
- Clients should have at least some access in a restricted manner like to be able to see a project status, expected ending date, expected budget.

## **Interview : 3**

### **Role Play Interview Plan**

**System:** Task Management System

**Project Reference:** SF/SJ/2021/10

**Interviewee:** 1) Vin Patel(Role Play)

Designation: Project Leader

**Interviewer:** 1) Jemis Variya

Designation : Software Development Engineer

2) Piyars Kakadiya

Designation : Software Development Engineer

**Date :** 27/9/2021

**Time :** 14:30

**Duration :** 30 minutes

**Mode :** Online

**Purpose of Interview :**

Preliminary meeting to identify the general and case specific requirements from the project leader's point of view.

**Agenda :**

- Distribution of subtask to each team member of respective department
- Track of current status of each department
- How he would communicate to team member of respective department
- Announcement for main aspects like change in deadline , flow of project, emergency altering of flow, organizing general meeting
- Allocation of required resources
- Actions required to be taken

**Interview Summary : 3**

**System:** Task Management System

**Project Reference:** SF/SJ/2021/10

**Interviewee:** 1) Vin Patel(Role Play)

Designation: Project Leader

**Interviewer:** 1) Jemis Variya

Designation: Software Development Engineer

2) Piyars Kakadiya

Designation: Software Development Engineer

**Date :** 27/9/2021

**Time :** 14:30

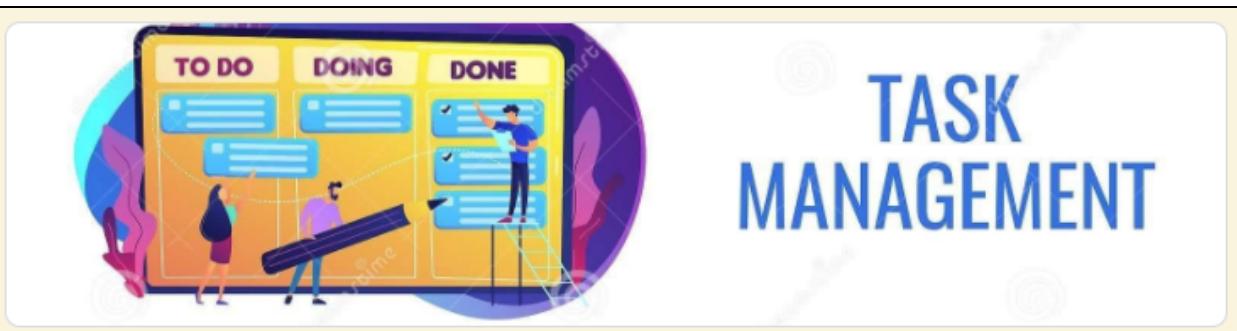
**Duration :** 30 minutes

**Mode :** Online

**Summary of interview :**

- Preliminary meeting to identify the general and case specific requirements from the project leader's point of view.
- For efficient resource allocation and deallocation time to time updation of each department work must be needed.
- Leader should be able to recognise if there is any misunderstanding occurring in a task.
- Inter communication among various departments.
- Compiling each completed component of different departments.

**iii. Questionnaires****Task Management System :**



## Task Management System

Please fill out the following information for our survey. It will take less than a minute.

 201901288@daiict.ac.in (not shared) Switch account 

\* Required

Name \*

Your answer

How often do you use task management system in a day?

- 0 - 3 times
- 4 - 6 times
- 7 - 9 times
- more than 9 times

Rate responsiveness of your existing system towards real time operations.

- Slow
- Moderate
- Fast

Does your existing system provide all the features that required to you? \*

- Yes
- No

What are the features provided by your existing system?

- Easy to access
- Task prioritization
- Track your own progress
- Reminder
- General announcements
- Other:

Do you get notified by your existing system?

- Yes
- No
- Sometimes

Would you like to have "Communication via chat feature" in new system?

- Yes
- No

Would you like to add "Raise query" feature?

- Yes
- No

Suggestions(if any)

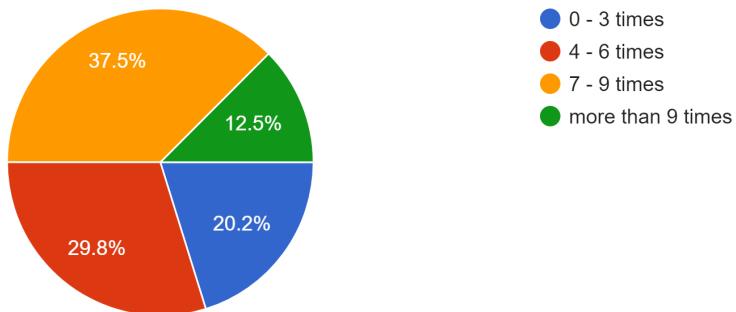
Your answer

---

### **Summary of Questionnaire Responses :**

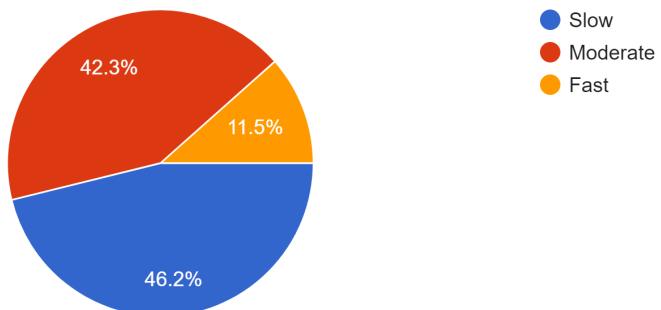
How often do you use task management system in a day?

104 responses



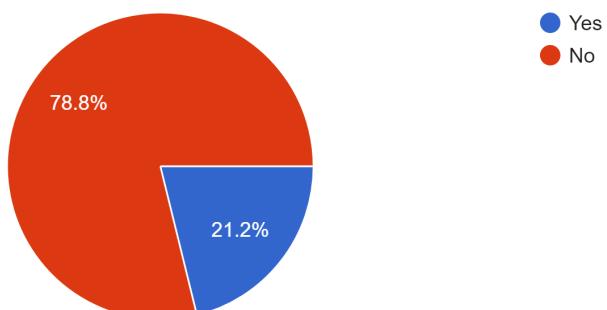
Rate responsiveness of your existing system towards real time operations.

104 responses



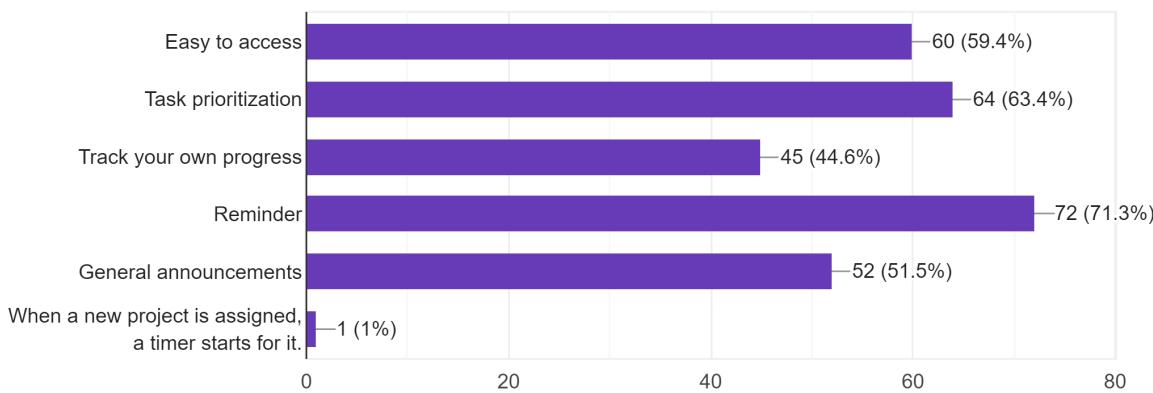
Does your existing system provide all the features that required to you?

104 responses



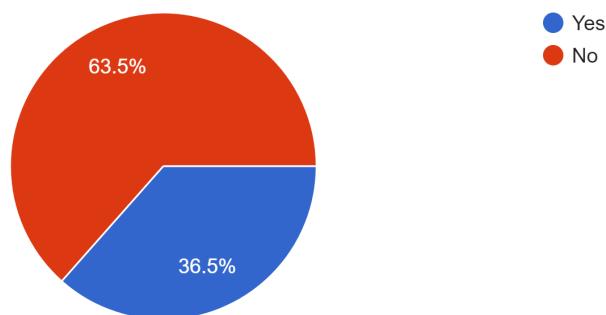
What are the features provided by your existing system?

101 responses



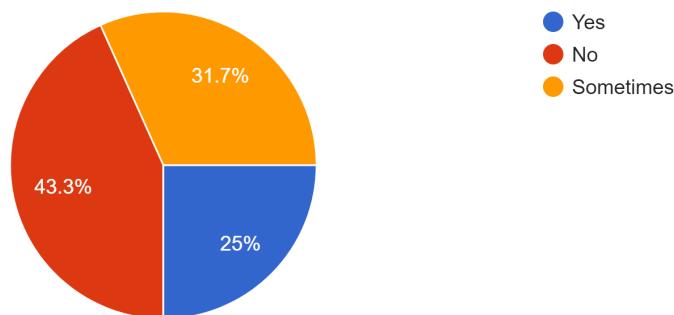
Does your existing system synchronize with calendar?

104 responses



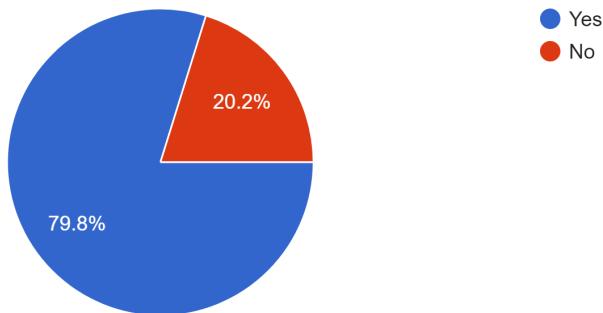
Do you get notified by your existing system?

104 responses



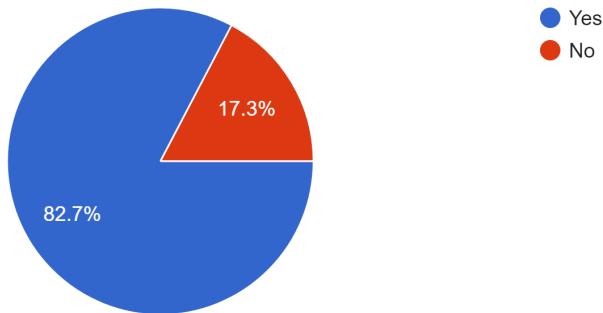
Would you like to have "Communication via chat feature" in new system?

104 responses



Would you like to add "Raise query" feature?

104 responses



#### Short Answer From Responses :

##### Would you like to give any suggestions ?

- The time tracking feature is required for employees because sometimes they work more and get less money.
- It would be better if the system includes a notification system.
- Responsiveness must be as quick as possible.
- My system is a bit advanced but the provider charges me a lot, so I'm looking for a replacement that is somewhat the same in quality but has a deducted charge.
- I need a system that is easy to access in the UI and sensitive to deadline alertness.
- I would like the new system to have the quickest response, as I'm using this system quite a lot in a day.
- The system in my company is a bit complex so new employees usually become unfamiliar with it and even refuse to use it for the initial 2-3 months.
- Recurring task scalability features must be provided by the system.
- In professional projects, a single task can at times serve multiple projects. For instance, the task of updating timesheets by the end of the week can be scheduled on a daily, weekly, or monthly basis for multiple teams.

- It's quite common to have instances in a project where two or multiple tasks are connected to each other. Setting up milestones to pinpoint specific phases within a project will ensure that directionally speaking the project is moving in the right direction.
- A part of task management is to monitor and observe task progress. This is needed to regulate the flow of tasks so as to avoid missing deadlines and make tasks progress more effectively.
- The system shall not have any single point of failure. All critical services of the system (data storage, communication subsystem) must be replicated. The system architecture shall allow the use of cluster hardware and support multi-processor systems.
- The system shall be available for use 24 hours a day, 7 days a week. The database shall be available for use 24 hours a day, 7 days a week. The maintenance weekends are allowed but must be announced 2 months in advance. The maintenance weekends mean that the system is offline for 48 hours for maintenance activities. The system must require not more than 6 maintenance weekends per year.
- The system shall provide full information about failures and errors. The information shall include : time of failure, origin (subsystem or component) where a failure occurred, severity and description of error or failure. Diagnostic information shall be logged and saved in independent data storage (disk file or database).
- The system shall protect the data and services from unauthorized access. The system shall also provide authentication and secure transactions.
- The system shall ensure secure and tamper-proof data exchange between parts of the system and the user. All data sent over the network (LAN or WAN) must be encrypted.
- The installation of the system must be a well-defined and well-documented procedure. The experienced system administrator shall be able to install the system within 1 day.
- The system shall support the concept of user. Every user of the system has a name and a password. The name must be unique within the installed instance of the system.
- The system shall determine what of its functionality is available to the authenticated user according to the user role and permissions and grant or deny access correspondingly.
- The user must be able to change his or her profile and save the changes.
- The system shall provide the concept of the project. The project has properties and contains zero or more tasks. The project must belong to one and only one project portfolio. The properties of the project are: Name, Description, Status, Creation Date, Start Date, Finish Date, Owner, and Project Leader.
- The system shall provide the concept of a task. The task consumes time and it requires resources. The task has properties and zero or more other tasks. These other tasks are called subtasks. The subtask must belong to one and only one task and can have zero or more subtasks. The circular references are not allowed. Hence the task cannot belong to its subtask. The task must belong to one and only one project. The task and subtask is associated with zero or more resources
- The system shall support dependencies between tasks. The dependence is a direct link between two tasks.

#### **iv. Requirement gathered from Responses :**

Most of the candidates are not satisfied with the number of features provided by their current system, so need to add up enough but require new features depending on the class of employees.

We need to improve a bit more overall responsiveness of the system, as there are a lot of candidates who use the system often in a day and already one-third of them reported that their system is not responsive enough. Less responsive and arbitrary delays put the system in the unsupportiveness frame for those employees.

Statistics say that over half of the candidates need two important features, system synchronization with calendar and notification alertness. Though these are the basic features to have still the existing systems are not providing.

Almost three of the fourth candidates reported that they want the in-built chat interface and a raise query feature. These are the integrated features that probably add an overload on the bounded cost of the entire system and need a bit more costly than the existing ones.

The system should remain in a working state 24 X 7, except if there is a maintenance period, generally weekend. But as the maintenance period doesn't allow employees to use any functionalities of the system, it should be as minimal as possible and should be properly notified priorly.

## v. Observations:

**System:** Task Management System

**Project Reference:** SF/SJ/2021/10

**Observations by :** Vin Patel

Raj Patel

Jemis Variya

Piyars Kakadiya

**Date:** 28/9/2021

**Time:** 14:30

**Duration:** 45 minutes

**Place:** Anant's Office (Season's Flavour)

- Undefined delay in updating status of a task, that is sometimes quick and sometimes too much delayed, which leads to unsupportiveness of the time management system from the employees perspective. For example, the immediate alteration by the client, or project leader reaches the employee very late which may cause unwanted repetition of the same task.
- The system was efficient but the system interface was a bit complex in understanding and so as in use. Many senior employees are facing issues related to this.
- Difficulties in collaboration among various departments for even a tiny issue. This issue is severe if it's a work-from-home scenario and leads to time wastage. One possible solution to this problem is to have a communication interface, like chat, among employees of all departments.

- Takes much time to resolve queries manually due to the absence of features like raise query which allows any class of user like an employee, manager, project leader to respond if they have a solution to the query raised by an employee.
- There should be a feature that enables notification, a reminder of deadlines and general announcements so that overall work efficiency can be improved.
- There should be additional data storage which is used for saving all the completed projects, so that in the future it may be useful to refer and reuse.

## Fact Finding Chart

Objective	Technique	Subject(s)	Time Commitment
To get background view of time management system	Background Reading	Company report, Videos	2 days
Find out any problem with the existing system	Interview	Manager	30 minutes
To establish any specific requirement	Questionnaire	Employee	Half-day
To know the problem faced by employee	Questionnaire	Employee	Half-day
To know cooperation from company	Interview	Client	30 minutes
To follow up development of business understanding	Observation	Product Manager	45 minute

## Requirements :

- Budget and deadline provided by the client not be exceeded
- Communication between client, manager and employee via chat interface
- Time tracking feature must be there
- Responsiveness must be as quick as possible
- Should have sensitive deadline alertness
- Easy to access overall environment of the system
- Reusability of previously completed task
- Recurring task should be fast and can be done automatically if possible

- Progress report with statistical analysis
- Prior announcements for maintenance regarding work
- Raise query feature
- The system shall determine what of its functionality is available to the authenticated user according to the user role and permissions in addition to granting or denying access accordingly.
- Detail-related projects like Name, Description, Status, Creation Date, Start Date, Finish Date, Owner, and Project Leader.
- Should give full information related to errors and failure and respective management mechanisms
- Give access to users according to their requirement
- Secure and encrypted data exchange
- Should have reliable service
- Less cost with descent feature
- Consistent with real-time
- Set up milestones after certain phases of project work which ensures the project goes in the right direction
- Dependencies among tasks should be handled in a supportive manner.

## User Classes and Characteristics :

**Task Manager :** The task manager is the one who is responsible for managing different projects at the same time running by the company. He has the highest privileges over all the system.

- Develop and maintain schedule
- Managing distribution of resources to various project teams who requested the same resource at the same time.
- Along with the respective project leader, the task manager fixes meetings with the client and gathers project requirements and estimates overall project cost.
- Analyze and monitor the project's progress.
- Allow proper accesses for viewing the progress of the project to the client
- Ensure the concerns related to client satisfaction

**Project Leader :** The project leader is primarily responsible for the successful completion of a project in the best collaborative manner. The project leader's role is to ensure that the project proceeds within the specified time frame and under the established budget while achieving its objectives. Project leaders make sure that projects are given sufficient resources.

- Develop a project plan

- Manage deliverables according to the plan
- Recruit and manage project team
- Determines methodology used in project
- Establish a project schedule and determine each phase
- Assign task to project team members
- Provide regular update to higher authorities
- Keeping track of the progress of work and if there is any misunderstanding or the project goes in the wrong direction then give guidance regarding this along with the task manager.

**Team member/Employee :** Project team members are the individuals who actively work on one or more phases of the project. They may be in-house staff or external consultants, working on the project on a full-time or part-time basis. Project team member roles can vary according to each project.

- Contributing to overall project objective
- Completing individual deliverables
- Providing expertise
- Working with users to establish and meet business needs
- Documenting to process

**Client :** Clients are the people, for whom the project is being undertaken whether it is an internal company project or an external. Every project has a client or customer, who has business or personal needs that will be satisfied by the successful completion of the project. While the term client generically refers to one or more people from the client's organization, specific large projects also have a dedicated client project lead as a comparable project leader to work on the client-side.

- Approve project plans
- Requesting changes
- Raising issues and risks
- Approving milestones
- Releasing payments

## Operating Environment

### Hardware requirements :

- Processor : Minimum 1 GHz, recommended 2 GHz or more
- Memory : Minimum 2GB, recommended 4GB or more
- Hard disk

### Software requirements :

- Operating system : any Windows, iOS, or Linux based operating system
- SQL server
- Access to third party applications
- Application server
- Front-end web server

- SharePoint server
- Web browser
- Visual reports
- Clock and Timer

#### **Connectivity Requirements :**

- Wi-Fi or internet connectivity
- All data exchanges are encrypted end to end
- If a user has any doubt while using the platform he can communicate with our AI chatbot about general doubts.
- There is feedback on the home page. Using that we continuously improve our system.

#### **External Interface related Requirement:**

- Mathematical Analyzer API of google
- Synchronize Calendar interface of google

## **Product Functions**

First a user logs in the system, after checking credentials the system allows him in the system and shows a dashboard respective to the class of user. Task managers can make changes with the highest level of access to the system, say adding project information. Then the task manager approaches the project leader to give or modify the project prototype and requirements, afterwards, the project leader assigns tasks to the respective team members. and assigns tasks to respective employees. Employees can see their own progress and can edit their profiles. Project leaders can allocate resources and can make project plans. Clients can see the current status of the project and request particular changes. Brief functionalities of the system are as follows:

**Prioritizing tasks :** Users should be able to prioritize their activities. Prioritizing tasks is incredibly helpful whenever urgent problems arise. By using a planning tool with scheduled time slots, Users can quickly make room in your planned schedule without manually rescheduling the whole project plan.

**Shared team calendar :** Establishing a shared calendar that synchronizes with popular calendar tools ensures that everyone receives notifications and attends meetings on time. Project leaders can add important due dates to their team calendar to have a quick overview of upcoming milestones.

**Planning and task scheduling :** A great functionality for teamwork is the planner feature that outlines all your planned project tasks and makes scheduling new activities incredibly easy.

**Quick Overviewing :** Project leaders get insight to whom to assign new tasks in the next step of a project. A quick overview of everyone's tasks and unscheduled time helps to set realistic deadlines and to avoid overbooking people.

**File sharing :** System includes features for file sharing, shared calendars and contact lists. Every team member can share information into the system and others will have quick access whenever and wherever they need it.

**Chat Interface :** System provides an in-built messenger app inside the project management tool. If employees have any kind of misunderstanding related to a task, to resolve it they may query the project leader, or sometimes task manager.

**Announcements :** It is a very useful and privileged feature for project leaders. It is for one-directional communication like announcements, guidelines, to inform about deadlines, to inform about changes in project and goals.

**Storing client and Project information :** It is particularly used for storing those client and project information that the company has taken from start to present.

**Quick access to your data :** Having all the data collected that you can easily share with your team and access it in a few clicks. No need to switch between multiple spreadsheets to find bits of data.

**Schedule your project resources :** Outlining all the scheduled and planned resources and calculating the cost of their use. This helps to avoid over-assignments and possible conflicts for lacking resources.

**Project budget dashboard :** Add graphs and charts to your dashboard to get real-time updates about your project. To share the dashboard or budget reports with team members, set up access rights, ensuring that the right people receive the data.

**Time tracking :** It is tracking how much time is spent by each employee in a project.

## Privileges

### Task Manager :

- He can access information like employee detail, project details, client detail, resources, project status.
- Can see the progress of employees and give guidance if there is some kind of misunderstanding.
- Privilege to announce general information or regarding specific issues.
- He can manage project budgets and time scheduling.
- In the end, he can even alter the prototype and workings of some specific functionalities, if needed, before finalizing the project.
- Allows resources to the project team according to their project.
- Can see the current progress and status of the project.

### Project leader :

- He can access all information regarding a given project.
- He can make a project plan and distribute resources accordingly.
- He can recruit team members and assign them tasks.
- Can see the current progress and status of the project.

### Client :

- He can raise issues and risks regarding the project.
- He can request changes for the project, in any phase of the project.
- He can approve the milestones and rate the work.
- Can see the current progress and status of the project.

### Employee :

- They have limited, primitive, access to the system. In general, they don't have more access regarding changing, and even inserting, something. But an employee can have some specific access if the higher authority has given access.
- They can access their profile and can see their own progress.
- They have a chat interface from where they can raise issues to all other employees for their queries.

## **Assumptions**

- One assumption that could affect the design is that the user runs a Mac X or Windows 8 operating system; a Windows operating system less than version 8 may cause unknown effects to the system functionality. Another assumption that could affect the design is that the user will have an adequate internet connection; this could affect the speed with which the interface communicates with the database. Lastly, this system will be written for users with a basic understanding of how computers work. Users with less computer experience may have a harder time.
- The database, local or provided by the service provider, is large enough to acquire all the necessary data into the system and not be exceeded
- The database service provider is assumed to provide responses 24 X 7.
- All the classes of users are familiar with their privileges, system environment, and qualified on the basis of the technological requirements.
- Only users with data modification privilege are authorized to access the various project management modules.
- Every user is allotted a unique User ID & Password such that no ambiguity appears regarding the same or empty ID.
- Sequence information: The required user-specific information will be present in the system before it can be accessed.

## **Business Constraints**

- With any project, there are limitations and risks that need to be addressed to ensure the project's ultimate success. The three primary constraints that project leaders should be familiar with are time, scope, and cost. These are widely known as the triple constraints or the project management triangle. Each constraint is connected to the other two; so, for example, increasing the scope of the project will likely require more time and money, while sleeping up the timeline for the project may cut costs, but also diminish the scope.
- So at the same time, we can't fulfill all of the three constraints, there are at least one, or even two, constraints that need to be compensated for remaining constraints depending upon the requirements and scenario.
- It has been identified that the computerization process is a necessity and what business benefits can be derived by implementing this improvement. It will lead to better transaction handling in terms of speed, reliability and accuracy.
- Creation of digital infrastructure, good internet connection and fast accessible database server that can handle large amounts of traffic at a time.

## **Section : 2 : Noun Analysis**

## Noun Analysis

<b>Nouns</b>	<b>Verbs</b>
task	raise
institutions	query
multinational	contribute
company	regulate
schools	create
hospitals	transaction
complexities	provide
team	require
Authorities	perform
management	resolve
team	Need
system	Alter
requirements	Have
user	computerise
ID	Improve
timeline	Obtain
constraints	give
risks	response
triangle	change

project	Make
limitations	Reflect
project	refer
digital	check
database	Read
infrastructure	Reliable
document	remove
project	record
manager	reside
client	extend
team member	prioritise
employee	identify
leaders	consider
graph	access
facility	progress
status	complete
database	distribute
audience	request
technical	describe
testing team	assign
account	announce
finance	communicate

research	guide
development	manage
interface	schedule
operation	allow
permission	need
worker	misunderstand
interaction	describe
announcement	track
communication	update
budget	allow
distribution	view
approval	reporting
data	Belongs
institution	Add
multinational	notify
group	check
system	collect
prototype	find
manul	planning
features	attend
condition	adding
individuals	make

process	synchronize
knowledge	prioritize
Action	modify
capacity	show
report	Log in
screen	require
errors	change
restriction	connect
security	exchanges
name	approve
password	develop
resources	analyze
goals	monitor
objectives	collaborate
performance	establish
accuracy	Update
speed	lead
result	create
time	Help
reason	track
emergency	Manage
misunderstanding	develop

Google	Work
accuracy	contribute
empty ID	allocate
modification	assign
Log	
Username	
Meet Summary	
Expected Cost	
Project plans	
phase	
Estimate	
Schedule	
Client Information	
subtasks	
transactions	
authentication	
database	
Recurring task	
Date	
architecture	
Cost	
department	

information	
Team	

## Accepted Nouns and Verbs List

Candidate entity set	Candidate attribute set	Candidate relationship set
Project Manager	<ul style="list-style-type: none"> <li>● <u>PM_ID</u></li> <li>● Emp_ID</li> <li>● Username</li> <li>● Password</li> <li>● Name</li> <li>● Designation</li> <li>● Dept_ID</li> <li>● Address</li> <li>● Contact</li> <li>● Email</li> <li>● Own_progress</li> <li>● Achievements</li> </ul>	Manage, Notify, Request, Allocate
Client	<ul style="list-style-type: none"> <li>● <u>CL_ID</u></li> <li>● Username</li> <li>● Password</li> <li>● Name</li> <li>● Address</li> <li>● Contact</li> <li>● Email</li> <li>● Company</li> </ul>	Request, View
Employee	<ul style="list-style-type: none"> <li>● <u>Emp_ID</u></li> <li>● Username</li> <li>● Password</li> <li>● Name</li> <li>● Designation</li> <li>● Address</li> <li>● Contact</li> <li>● Email</li> <li>● Own_Progress</li> </ul>	Update, Team, Raise_issues, Belongs
Department	<ul style="list-style-type: none"> <li>● <u>Dept_ID</u></li> <li>● Dept_Name</li> </ul>	Belongs

Task	<ul style="list-style-type: none"> <li>• <u>Task_ID</u></li> <li>• Name</li> <li>• Cost</li> <li>• Start_date</li> <li>• End_date</li> <li>• Progress</li> <li>• Status</li> </ul>	Splits, Updates, Team
Query	<ul style="list-style-type: none"> <li>• <u>Query_ID</u></li> <li>• Query</li> <li>• Status</li> <li>• Reply</li> </ul>	Raise_Issue
Logs	<ul style="list-style-type: none"> <li>• <u>Log_ID</u></li> <li>• Log_info</li> <li>• Date</li> </ul>	Updates
Announcements	<ul style="list-style-type: none"> <li>• <u>Announcement_ID</u></li> <li>• Announcement</li> <li>• Date</li> </ul>	Notify
Project	<ul style="list-style-type: none"> <li>• <u>Project_ID</u></li> <li>• Name</li> <li>• Budget</li> <li>• Start_date</li> <li>• End_date</li> <li>• Progress</li> <li>• Status</li> <li>• Expected_Deadline</li> <li>• Meet_summary</li> <li>• Expected_Cost</li> </ul>	Manage, Uses, View, Splits
Resource	<ul style="list-style-type: none"> <li>• <u>RS_ID</u></li> <li>• ID</li> <li>• Password</li> <li>• Description</li> <li>• Issue_date</li> <li>• Return_date</li> </ul>	Allocate, Uses

## Rejected Nouns and Verbs List

Noun	Reject Reason	Verb	Reject Reason
institutions	Irrelevant	establish	Vague
multinational	Irrelevant	monitor	Irrelevant
company	Irrelevant	collaborate	General
schools	Irrelevant	regulate	Irrelevant
hospitals	Irrelevant	create	Irrelevant
complexities	Irrelevant	transaction	Vague
Authorities	Irrelevant	analyze	Vague
management	Irrelevant	develop	Irrelevant
system	Irrelevant	perform	Irrelevant
requirements	General	resolve	Irrelevant
user	Duplicate	Need	General
ID	General	Alter	Vague
timeline	Irrelevant	Have	Irrelevant
constraints	Vague	computerise	General
risks	Irrelevant	Improve	Irrelevant
triangle	Irrelevant	Obtain	Irrelevant
limitations	Irrelevant	give	Irrelevant
digital	Irrelevant	response	Vague
database	Irrelevant	change	Vague
infrastructure	Irrelevant	Make	Irrelevant
document	Irrelevant	Reflect	Irrelevant
manager	General	refer	Vague
leaders	General	check	Irrelevant

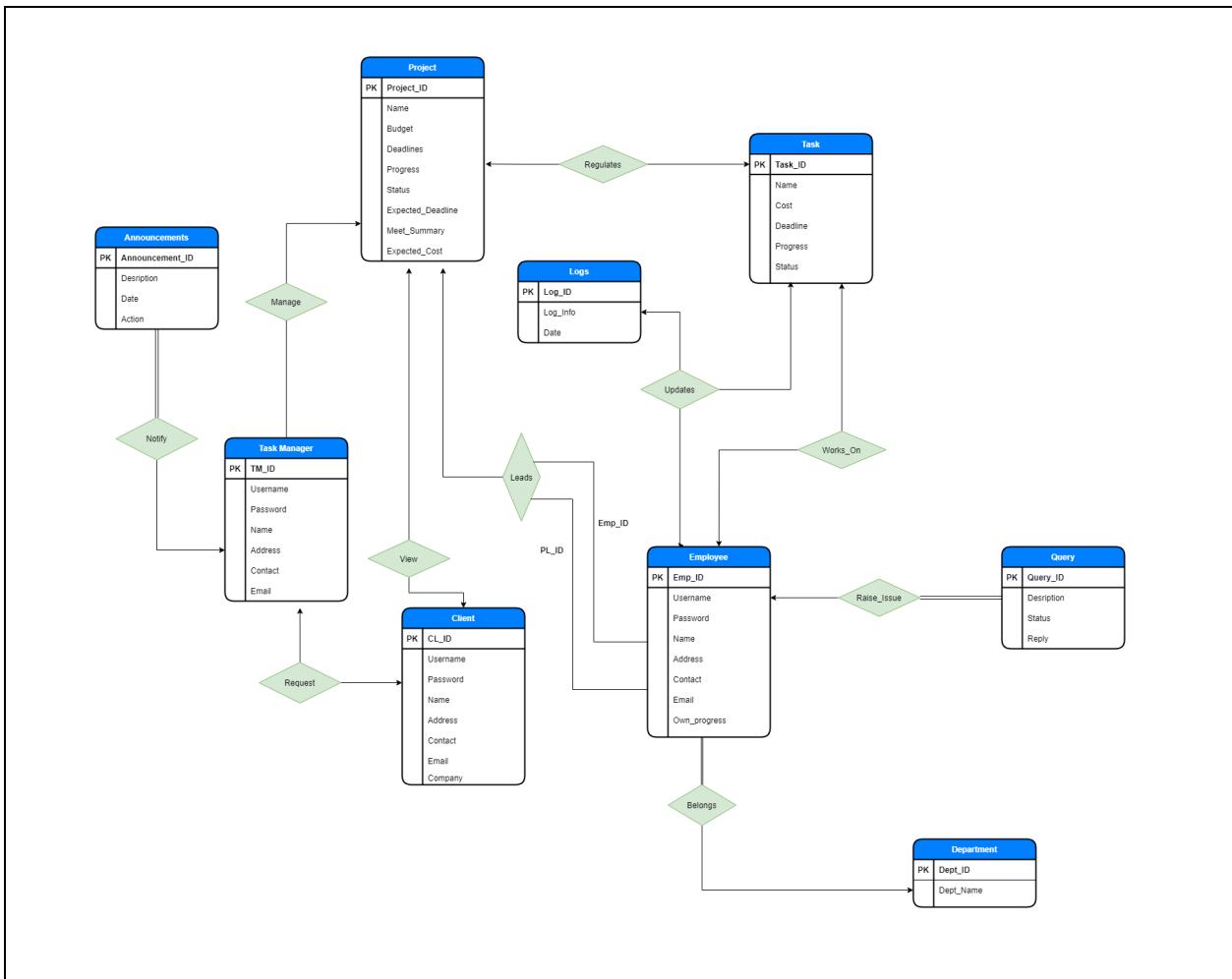
graph	Irrelevant	Read	General
status	Duplicate	Reliable	Irrelevant
database	Vague	remove	Irrelevant
audience	General	approve	Irrelevant
technical	Irrelevant	reside	Vague
testing team	Duplicate	extend	Irrelevant
account	Irrelevant	prioritise	Duplicate
finance	Irrelevant	identify	Irrelevant
research	Vague	consider	Vague
development	General	access	Irrelevant
interface	Irrelevant	progress	Duplicate
operation	Irrelevant	exchanges	Irrelevant
permission	Vague	distribute	Irrelevant
worker	Irrelevant	connect	Irrelevant
interaction	Vague	describe	Irrelevant
budget	Duplicate	modify	Irrelevant
distribution	Vague	require	Vague
data	General	communicate	Irrelevant
institution	Duplicate	adding	Irrelevant
multinational	Irrelevant	manage	Duplicate
group	Duplicate	schedule	Irrelevant
system	Irrelevant	allow	Irrelevant
prototype	Irrelevant	need	Irrelevant
manual	Irrelevant	show	Irrelevant
features	Irrelevant	synchronize	Irrelevant
condition	Irrelevant	track	Duplicate
individuals	Irrelevant	update	Duplicate

process	Vague	make	Duplicate
knowledge	Irrelevant	notify	Duplicate
capacity	Irrelevant	reporting	Vague
report	Vague	planning	Irrelevant
screen	Irrelevant	Add	Irrelevant
errors	Vague	attend	Irrelevant
name	Duplicate	check	Irrelevant
password	Duplicate	collect	Irrelevant
objectives	Irrelevant	find	Irrelevant
accuracy	Irrelevant		
speed	Irrelevant		
result	Irrelevant		
time	Duplicate		
reason	Vague		
emergency	Irrelevant		
Google	Irrelevant		
accuracy	Irrelevant		
empty ID	Vague		
modification	Irrelevant		
Expected Cost	Vague		
phase	Irrelevant		
Estimate	Irrelevant		
Schedule	Duplicate		
Client Information	Vague		
transactions	Duplicate		
authentication	Vague		
database	Irrelevant		

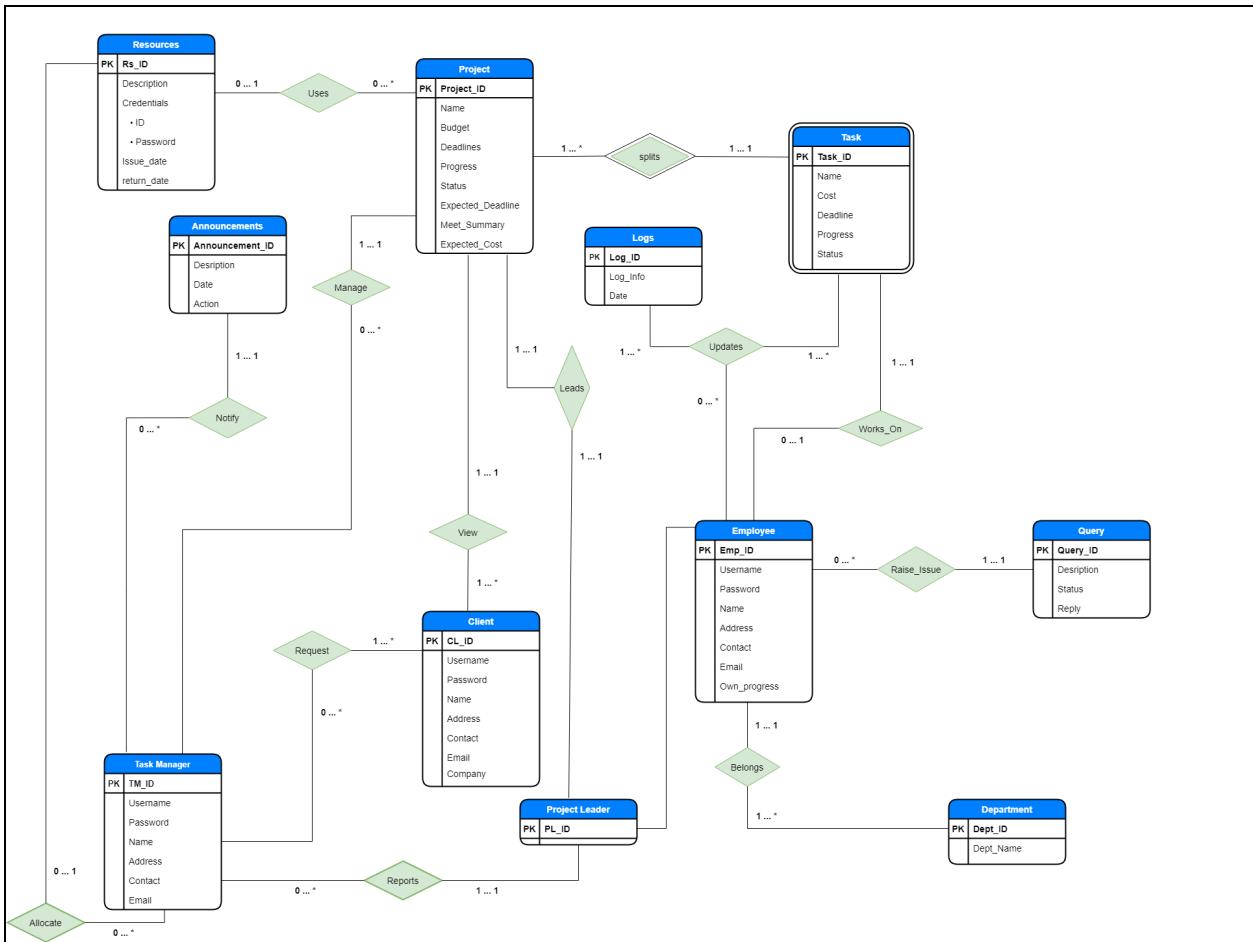
Recurring task	Vague		
Date	General		
architecture	Irrelevant		
Department	Duplicate		
information	General		

## **Section : 3 : ER-Diagram all versions**

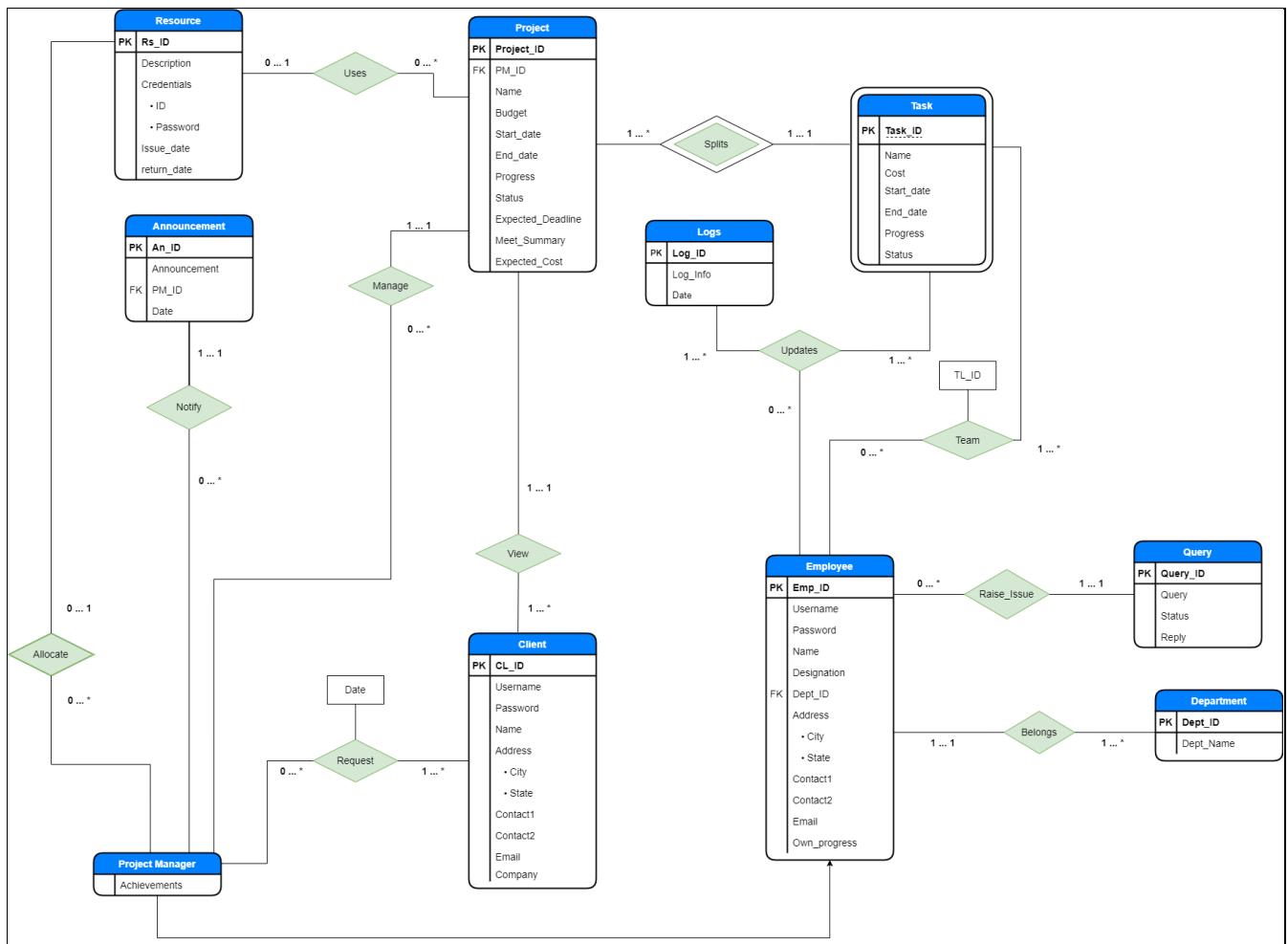
**ER Diagram V1**



## ER Diagram V2



## ER Diagram V3(Final)



## **Section : 4 : Conversion of Final ER-Diagram to Relational Model**

## Mapping E-R Model to Relational Model

1. **Project Manager (PM\_ID, Achievement)**
2. **Client (CL\_ID, Username, Password, Name, City, State, Email, Company, Contact1, Contact2)**
3. **Employee (Emp\_ID, Dept\_ID, Team\_ID, Username, Password, Name, Designation, City, State, Email, Own\_progress, Contact1, Contact2)**
4. **Task (Task\_ID, Project\_ID, Team\_ID, Name, Cost, Start\_date, End\_Date, Progress, Status)**
5. **Project (Project\_ID, PM\_ID, CL\_ID, Name, Budget, Start\_date, End\_date, Progress, Status, Expected\_Deadline, Meet\_Summary, Expected\_Cost)**
6. **Team (Team\_ID)**
7. **Announcement (An\_ID, PM\_ID, Announcement, Date)**
8. **Query (Query\_ID, Emp\_ID, Query, Status, Response)**
9. **Log (Log\_ID, Log\_Info, Date)**
10. **Department (Dept\_ID, Dept\_Name)**
11. **Resource (Rs\_ID, PM\_ID, Project\_ID, Description, ID, Password, Issue\_date, Return\_date)**
12. **Request (PM\_ID, CL\_ID)**
13. **Updates (Emp\_ID, Project\_ID, Task\_ID, Log\_ID)**

## DDL Scripts

### 1. Project Manager

```
create table "Project Manager"
(
    -- ek navu att. add karvanu.
    "PM_ID" bigint not null,
    constraint "PK_ProjectManager PM_ID" primary key ("PM_ID"),
    constraint "FK_Employee Emp_ID" foreign key ("PM_ID") references "Employee"
    ("Emp_ID")
        match simple
        on update cascade
        on delete cascade
        not valid
);
```

### 2. Client

```
create table "Client"
(
    "CL_ID" bigint generated always as identity,
    "Username" varchar(20) not null,
    "Password" varchar(20) not null,
    "Name" varchar(50) not null,
    "City" varchar(20) not null,
    "State" varchar(20) not null,
    "Email" varchar(20) not null,
    "Company" varchar(20),
    "Contact1" bigint not null,
    "Contact2" bigint,
    constraint "PK_Client CL_ID" primary key ("CL_ID"),
    constraint "unique_client_username" unique ("Username")
)
```

### 3. Employee

```
create table "Employee"
(
    "Emp_ID" bigint generated always as identity,
    "Dept_ID" bigint not null,
    "Team_ID" bigint not null,
```

```

"Username" varchar(20) not null,
"Password" varchar(20) not null,
"Designation" varchar(20),
"Name" varchar(50) not null,
"City" varchar(20) not null,
"State" varchar(20) not null,
"Email" varchar(20) not null,
"Own_progress" smallint,
>Contact1" bigint not null,
>Contact2" bigint,

constraint "PK_Employee Emp_ID" primary key ("Emp_ID"),
constraint "FK_Team Team_ID" foreign key ("Team_ID") references "Team" ("Team_ID")
    match simple
    on update cascade
    on delete cascade
    not valid,
constraint "FK_Department Dept_ID" foreign key ("Dept_ID") references "Department"
("Dept_ID")
    match simple
    on update cascade
    on delete cascade
    not valid,
constraint "unique_employee_username" unique ("Username")
)

```

#### 4. Task

```

create table "Task"
(
    "Task_ID" bigint generated always as identity,
    "Project_ID" bigint not null,
    "Team_ID" bigint not null,
    "Name" varchar(50) not null,
    "Cost" bigint,
    "Start_date" date not null,
    "End_date" date,
    "Progress" smallint,
    "Status" varchar(20),

constraint "PK_Task (Project_ID, Task_ID)" primary key ("Project_ID", "Task_ID"),
constraint "FK_Team Team_ID" foreign key ("Team_ID") references "Team" ("Team_ID")
    match simple
    on update cascade
    on delete cascade
    not valid,
constraint "FK_Project Project_ID" foreign key ("Project_ID") references "Project"
("Project_ID")
    match simple
    on update cascade
    on delete cascade
)

```

```

        not valid,
    check ("Status" in ('Approved', 'Pending', 'Improve'))
)

```

## 5. Project

```

create table "Project"
(
    "Project_ID" bigint generated always as identity,
    "PM_ID" bigint not null,
    "CL_ID" bigint not null,
    "Name" varchar(50) not null,
    "Budget" bigint not null check ("Budget" > 0),
    "Start_date" date not null,
    "End_date" date,
    "Progress" smallint,
    "Status" varchar(20),
    "Expected_Deadline" date,
    "Meet_Summary" text,
    "Expected_Cost" bigint,

    constraint "PK_Project Project_ID" primary key ("Project_ID"),
    constraint "FK_ProjectManager PM_ID" foreign key ("PM_ID") references "Project Manager"
    ("PM_ID")
        match simple
        on update cascade
        on delete cascade
        not valid,
    constraint "FK_Client CL_ID" foreign key ("CL_ID") references "Client" ("CL_ID")
        match simple
        on update cascade
        on delete cascade
        not valid,
    check ("Status" in ('Approved', 'Pending', 'Improve'))
)

```

## 6. Team

```

create table "Team"
(
    "Team_ID" bigint generated always as identity,
    constraint "PK_Team Team_ID" primary key("Team_ID")
)

```

## 7. Announcement

```

create table "Announcement"
(
    "An_ID" bigint generated always as identity,
    "PM_ID" bigint not null,
    "Announcement" text,
    "Date" date not null,

    constraint "PK_Announcement An_ID" primary key("An_ID"),
    constraint "FK_ProjectManager PM_ID" foreign key ("PM_ID") references "Project Manager"
    ("PM_ID")
        match simple
        on update cascade
        on delete cascade
        not valid
)

```

## 8. Query

```

create table "Query"
(
    "Query_ID" bigint generated always as identity,
    "Emp_ID" bigint not null,
    "Query" text,
    "Status" varchar(20),
    "Response" text,

    constraint "PK_Query Query_ID" primary key("Query_ID"),
    constraint "FK_Employee Emp_ID" foreign key ("Emp_ID") references "Employee"
    ("Emp_ID")
        match simple
        on update cascade
        on delete cascade
        not valid,
    check ("Status" in ('Pending', 'Solved'))
)

```

## 9. Log

```

create table "Log"
(
    "Log_ID" bigint generated always as identity,
    "Log_Info" text,
    "Date" date,

    constraint "PK_Log Log_ID" primary key("Log_ID")
)

```

## 10. Department

```
create table "Department"
(
    "Dept_ID" bigint generated always as identity,
    "Dept_Name" varchar(20) not null,
    constraint "PK_Department Dept_ID" primary key("Dept_ID")
)
```

## 11. Resource

```
create table "Resource"
(
    "Rs_ID" bigint generated always as identity,
    "PM_ID" bigint not null,
    "Project_ID" bigint not null,
    "Description" text,
    "ID" bigint not null unique,
    "Password" varchar(20) not null,
    "Issue_date" date,
    "Return_date" date,
    constraint "PK_Resource Rs_ID" primary key("Rs_ID"),
    constraint "FK_ProjectManager PM_ID" foreign key ("PM_ID") references "Project Manager"("PM_ID")
        match simple
        on update cascade
        on delete cascade
        not valid,
    constraint "FK_Project Project_ID" foreign key ("Project_ID") references "Project"("Project_ID")
        match simple
        on update cascade
        on delete cascade
        not valid
)
```

## 12. Request

```
create table "Request"
(
    "PM_ID" bigint not null,
    "CL_ID" bigint not null,
    constraint "PK_Request (PM_ID, CL_ID)" primary key("PM_ID", "CL_ID"),
    constraint "FK_ProjectManager PM_ID" foreign key ("PM_ID") references "Project Manager"
```

```

("PM_ID")
    match simple
    on update cascade
    on delete cascade
    not valid,
    constraint "FK_Client CL_ID" foreign key ("CL_ID") references "Client" ("CL_ID")
        match simple
        on update cascade
        on delete cascade
        not valid
)

```

### 13. Updates

```

create table "Updates"
(
    "Emp_ID" bigint not null,
    "Project_ID" bigint not null,
    "Task_ID" bigint not null,
    "Log_ID" bigint not null,

    constraint "PK_Updates (Emp_ID, Project_ID, Task_ID, Log_ID)" primary key("Emp_ID",
    "Project_ID", "Task_ID", "Log_ID"),
        constraint "FK_Employee Emp_ID" foreign key ("Emp_ID") references "Employee"
    ("Emp_ID")
            match simple
            on update cascade
            on delete cascade
            not valid,
        constraint "FK_Task (Project_ID, Task_ID)" foreign key ("Project_ID", "Task_ID") references
    "Task" ("Project_ID", "Task_ID")
            match simple
            on update cascade
            on delete cascade
            not valid,
        constraint "FK_Log Log_ID" foreign key ("Log_ID") references "Log" ("Log_ID")
            match simple
            on update cascade
            on delete cascade
            not valid
)

```

## **Section : 5 : Normalization and Schema Refinement**

## List all the Relations & Schemas with all the details (Original Design of Database)

1. **Project Manager (PM\_ID, Achievement)**
2. **Client (CL\_ID, Username, Password, Name, State, City, Email, Company, Contact1, Contact2)**
3. **Employee (Emp\_ID, Dept\_ID, Username, Password, Name, Designation, State, City, Email, Own\_progress, Contact1, Contact2)**
4. **Task (Project\_ID, Task\_ID, Name, Cost, Start\_date, End\_Date, Progress, Status)**
5. **Project (Project\_ID, PM\_ID, CL\_ID, Name, Budget, Start\_date, End\_date, Progress, Status, Expected\_Deadline, Meet\_Summary, Expected\_Cost)**
6. **Team (Project\_ID, Task\_ID, Member\_ID, TL\_ID)**
7. **Announcement (An\_ID, PM\_ID, Announcement, Date)**
8. **Query (Query\_ID, Emp\_ID, Query, Status, Response)**
9. **Log (Log\_ID, Log\_Info, Date)**
10. **Department (Dept\_ID, Dept\_Name)**
11. **Resource (Rs\_ID, PM\_ID, Project\_ID, Description, ID, Password, Issue\_date, Return\_date)**
12. **Request (PM\_ID, CL\_ID, Date)**
13. **Updates (Emp\_ID, Project\_ID, Task\_ID, Log\_ID)**

➤ **Identify and list all types of dependencies (PK, FK, Functional Dependencies) for each relation**

**1. Project Manager**

Primary Key : PM\_ID

Foreign Key : Emp\_ID

Functional Dependencies :

- Primary : PM\_ID  $\rightarrow$  Achievement

Redundancy: There is no such redundancy present over here.

**Anomalies:**

There is no insert, delete and update anomaly present over here.

**1NF :**

This relation is already in 1NF form as there is no multivalued attribute.

**2NF :**

This relation is already in 2NF form as there are no partial dependencies.

**3NF / BCNF:**

This relation is already in 3NF / BCNF form as there are no transitive dependencies.

**2. Client**

Primary Key : CL\_ID

Foreign Key :

Functional Dependencies :

- Primary : CL\_ID  $\rightarrow$  Username, Password, Name, State, City, Email, Contact1, Contact2
- Other : Username  $\rightarrow$  Password, Name, State, City, Email, Company, Contact1, Contact2

Redundancy: If there are many clients having a single contact number, their corresponding contact2 will remain as NULL so this implies the schema is badly represented and needs to be resolved appropriately.

**Anomalies:**

There is no insert, delete and update anomaly present over here.

**1NF :**

This relation contains multivalued attributes for contact so we have to decompose it accordingly.

Thus, 1NF form is:

**Client(CL\_ID,Username, Password, Name, State, City, Email, Company)**

**Client\_contact(CL\_ID, contact)**

**2NF :**

This relation is already in 2NF form as there are no partial dependencies.

**3NF / BCNF:**

This relation is already in 3NF / BCNF form as there are no transitive dependencies.

In the FD's having Username as discriminant (left side), they all satisfy the 3NF / BCNF condition that the discriminant is super key.

### 3. Employee

Primary Key : Emp\_ID

Foreign Key : Dept\_ID

Functional Dependencies :

- Primary : Emp\_ID  $\rightarrow$  Dept\_ID, Username, Password, Name, Designation, State, City, Email, Own\_progress.
- Other : Username  $\rightarrow$  Dept\_ID, Password, Name, Designation, State, City, Email, Own\_progress, Contact1, Contact2

Redundancy: If there are many employees having a single contact number, their corresponding contact2 will remain as NULL so this implies the schema is badly represented and needs to be resolved appropriately.

Insert Anomaly: There is no insert anomaly present over here.

Delete Anomaly: There is no delete anomaly present over here.

Update Anomaly: There is no update anomaly present over here.

**1NF :**

This relation contains multivalued attribute contact so we have to decompose it accordingly.

Thus, 1NF form is:

**Employee(Emp\_ID,Dept\_ID, Username, Password, Name, Designation, State, City, Email, Own\_progress)**

**Emp\_contact(Emp\_ID, contact)**

**2NF :**

This relation is already in 2NF form as there are no partial dependencies.

**3NF / BCNF:**

This relation is already in 3NF / BCNF form as there are no transitive dependencies.

In the FD's having Username as discriminant (left side), they all satisfy the 3NF / BCNF condition that the discriminant is super key.

#### 4. Task

Primary Key : Project\_ID, Task\_ID

Foreign Key : Project\_ID

Functional Dependencies :

- Primary : Project\_ID, Task\_ID  $\rightarrow$  Name, Cost, Start\_date, End\_Date, Progress, Status

Redundancy: There is no such redundancy present over here.

Insert Anomaly: There is no insert anomaly present over here.

Delete Anomaly: There is no delete anomaly present over here.

Update Anomaly: There is no update anomaly present over here.

**1NF :**

This relation is already in 1NF form as there is no multivalued attribute.

**2NF :**

This relation is already in 2NF form as there are no partial dependencies.

**3NF / BCNF:**

This relation is already in 3NF / BCNF form as there are no transitive dependencies.

#### 5. Project

Primary Key : Project\_ID

Foreign Key : PM\_ID, CL\_ID

Functional Dependencies :

- Primary :- Project\_ID  $\rightarrow$  PM\_ID, CL\_ID, Name, Budget, Start\_date, End\_date, Progress, Status, Expected\_Deadline, Meet\_Summary, Expected\_Cost

Redundancy: There is no such redundancy present over here.

Insert Anomaly: There is no insert anomaly present over here.

Delete Anomaly: There is no delete anomaly present over here.

Update Anomaly: There is no update anomaly present over here.

**1NF :**

This relation is already in 1NF form as there is no multivalued attribute.

**2NF :**

This relation is already in 2NF form as there are no partial dependencies as the primary key contains only a single attribute.

### **3NF / BCNF:**

This relation is already in 3NF / BCNF form because there is no CK other than PK so there are no transitive dependencies.

## **6. Team**

Primary Key : Project\_ID, Task\_ID, Member\_ID

Foreign Key : Project\_ID, Task\_ID, Member\_ID

Functional Dependencies :

- Primary : Project\_ID, Task\_ID, Member\_ID  $\rightarrow$  TL\_ID
- Partial : Project\_ID, Task\_ID  $\rightarrow$  TL\_ID

Redundancy: For the employees working on the same task, have the same Team and so as TL\_ID so that will lead to unnecessary redundancy and need to be addressed properly.

Insert Anomaly: There is no insert anomaly present over here.

Delete Anomaly: If we would delete any particular tuple then we may lose the information regarding which employee did work as team leader for a particular project? And how many times an employee worked as team leader? etc.,

Update Anomaly: Whenever there is sudden change in team leader then would have to make corresponding changes regarding the team leader at many places.

### **1NF :**

This relation is already in 1NF form as there is no multivalued attribute.

### **2NF :**

There is a partial dependency Project\_ID, Task\_ID  $\rightarrow$  TL\_ID so decomposing in two relations.

**Team(Project\_ID, Member\_ID, Task\_ID)**

**Team\_leader(Project\_ID, Task\_ID, TL\_ID)**

### **3NF / BCNF:**

This relation is already in 3NF / BCNF form as there are no transitive dependencies.

## **7. Announcement**

Primary Key : An\_ID

Foreign Key : PM\_ID

Functional Dependencies :

- Primary : An\_ID  $\rightarrow$  PM\_ID, Announcement, Date

Redundancy: There is no such redundancy present over here.

As there is already a separate relation for project manager so it will not create insert, delete, update anomaly.

**1NF :**

This relation is already in 1NF form as there is no multivalued attribute.

**2NF :**

This relation is already in 2NF form as there are no partial dependencies as the primary key contains only a single attribute.

**3NF / BCNF:**

This relation is already in 3NF / BCNF form because there is no CK other than PK so there are no transitive dependencies.

**8. Query**

Primary Key : Query\_ID

Foreign Key : Emp\_ID

Functional Dependencies :

- Primary : Query\_ID  $\rightarrow$  Emp\_ID, Query, Status, Response

Redundancy: There is no such redundancy present over here.

As there is already a separate relation for employees so it will not create insert, delete, update anomaly.

**1NF :**

This relation is already in 1NF form as there is no multivalued attribute.

**2NF :**

This relation is already in 2NF form as there are no partial dependencies as the primary key contains only a single attribute.

**3NF / BCNF:**

This relation is already in 3NF / BCNF form because there is no CK other than PK so there are no transitive dependencies.

**9. Log**

Primary Key : Log\_ID

Functional Dependencies :

- Primary : Log\_ID  $\rightarrow$  Log\_Info, Date

Redundancy: There is no such redundancy present over here.

Insert Anomaly: There is no insert anomaly present over here.

Delete Anomaly: There is no delete anomaly present over here.

Update Anomaly: There is no update anomaly present over here.

**1NF :**

This relation is already in 1NF form as there is no multivalued attribute.

**2NF :**

This relation is already in 2NF form as there are no partial dependencies.

**3NF / BCNF:**

This relation is already in 3NF / BCNF form as there are no transitive dependencies.

**10. Department**

Primary Key : Dept\_ID

Functional Dependencies :

- Primary : Dept\_ID-> Dept\_Name

Redundancy: There is no such redundancy present over here.

Insert Anomaly: There is no insert anomaly present over here.

Delete Anomaly: There is no delete anomaly present over here.

Update Anomaly: There is no update anomaly present over here.

**1NF :**

This relation is already in 1NF form as there is no multivalued attribute.

**2NF :**

This relation is already in 2NF form as there are no partial dependencies.

**3NF / BCNF:**

This relation is already in 3NF / BCNF form as there are no transitive dependencies.

**11. Resource**

Primary Key : Rs\_ID

Foreign Key : PM\_ID, Project\_ID

Functional Dependencies :

- Primary : Rs\_ID-> PM\_ID, Project\_ID, Description, ID, Password, Issue\_date, Return\_date
- Other : ID-> PM\_ID, Project\_ID, Description, Password, Issue\_date, Return\_date

Redundancy: There is no such redundancy present over here.

Insert Anomaly: There is no insert anomaly present over here.

Delete Anomaly: There is no delete anomaly present over here.

Update Anomaly: There is no update anomaly present over here.

**1NF :**

This relation is already in 1NF form as there is no multivalued attribute.

**2NF :**

This relation is already in 2NF form as there are no partial dependencies.

**3NF / BCNF:**

This relation is already in 3NF / BCNF form as there are no transitive dependencies.

**12. Request**

Primary Key : PM\_ID, CL\_ID

Foreign Key : PM\_ID, CL\_ID

Functional Dependencies :

- Primary : PM\_ID, CL\_ID  $\rightarrow$  Date

Redundancy: There is no such redundancy present over here.

Insert Anomaly: There is no insert anomaly present over here.

Delete Anomaly: There is no delete anomaly present over here.

Update Anomaly: There is no update anomaly present over here.

**1NF :**

This relation is already in 1NF form as there is no multivalued attribute.

**2NF :**

This relation is already in 2NF form as there are no partial dependencies.

**3NF / BCNF:**

This relation is already in 3NF / BCNF form as there are no transitive dependencies.

**13. Update**

Primary Key : Emp\_ID, Project\_ID, Task\_ID, Log\_ID

Foreign Key : Emp\_ID, Project\_ID, Task\_ID, Log\_ID

Every single attribute participates in primary key, so there will not be any functional dependency so as no redundancy.

Thus, we can conclude that the relation is already in 3NF / BCNF.

➤ **Final List of all the Relations & Schemas with all details (Original Design of Database)**

1. **Project Manager (PM\_ID, Achievement)**
2. **Client(CL\_ID,Username, Password, Name, State, City, Email, Company)**
3. **Client\_contact(CL\_ID, contact)**
4. **Employee(Emp\_ID,Dept\_ID, Username, Password, Name, Designation, State, City, Email, Own\_progress)**
5. **Emp\_contact(Emp\_ID, contact)**
6. **Task (Project\_ID, Task\_ID, Name, Cost, Start\_date, End\_Date, Progress, Status)**
7. **Project (Project\_ID, PM\_ID, CL\_ID, Name, Budget, Start\_date, End\_date, Progress, Status, Expected\_Deadline, Meet\_Summary, Expected\_Cost)**
8. **Team(Project\_ID, Member\_ID, Task\_ID)**
9. **Team\_leader(Project\_ID, Task\_ID, TL\_ID)**
10. **Announcement (An\_ID, PM\_ID, Announcement, Date)**
11. **Query (Query\_ID, Emp\_ID, Query, Status, Response)**
12. **Log (Log\_ID, Log\_Info, Date)**
13. **Department (Dept\_ID, Dept\_Name)**
14. **Resource (Rs\_ID, PM\_ID, Project\_ID, Description, ID, Password, Issue\_date, Return\_date)**
15. **Request (PM\_ID, CL\_ID, Date)**
16. **Updates (Emp\_ID, Project\_ID, Task\_ID, Log\_ID)**

## ➤ Data and DDL Snapshots

### 1. Department

```
create table "Department"
(
    "Dept_ID" varchar(10) not null,
    "Dept_Name" varchar(20) not null,
    constraint "PK_Department Dept_ID" primary key("Dept_ID")
);
```

Query Editor    Query History

```
1 SELECT * FROM "TMS"."Department"
2 ORDER BY "Dept_ID" ASC
```

Data Output    Explain    Messages    Notifications

Dept_ID [PK] character varying (10)	Dept_Name character varying (20)
1 D0000001	HR
2 D0000002	Finance
3 D0000003	Sales
4 D0000004	Marketing
5 D0000005	Technical
6 D0000006	Development
7 D0000007	Customer Service
8 D0000008	Quality Control
9 D0000009	Testing
10 D0000010	Admin
11 D0000011	Management

## 2. Employee

```

create table "Employee"
(
    "Emp_ID" varchar(100) not null,
    "Dept_ID" varchar(100) not null,
    "Username" varchar(200) not null,
    "Password" varchar(200) not null,
    "Designation" varchar(200),
    "Name" varchar(100) not null,
    "City" varchar(200) not null,
    "State" varchar(200) not null,
    "Email" varchar(100) not null,
    "Own_progress" smallint,

    constraint "PK_Employee_Emp_ID" primary key ("Emp_ID"),
    constraint "FK_Department_Dept_ID" foreign key ("Dept_ID") references "Department" ("Dept_ID")
        match simple
        on update cascade
        on delete cascade
        not valid,
    constraint "unique_employee_username" unique ("Username")
);

```

Query Editor    Query History

```

1 SELECT * FROM "TMS"."Employee"
2 ORDER BY "Emp_ID" ASC

```

Data Output   Explain   Messages   Notifications

	Emp_ID [PK] character varying (100)	Dept_ID character varying (100)	Username character varying (200)	Password character varying (200)	Designation character varying (200)	Name character varying (100)	City character varying (200)
1	E0000001	D0000001	gshadfourth0	rQ9tdNCr	Garwood Shadfourth	Intern	Delaware
2	E0000002	D0000002	arichen1	gsSo5wxlmo	Adina Richen	Senior	Tennessee
3	E0000003	D0000003	hhulstrom2	KfDRNn1	Hephzibah Hulstrom	Intern	District of Columbia
4	E0000004	D0000004	nkadd3	byaiETy0JL	Nikola Kadd	Senior	California
5	E0000005	D0000005	vhilland4	9P6lqKpUbz	Venus Hilland	Senior	Virginia
6	E0000006	D0000006	rmeat5	ulkbsya	Reine Neat	Junior	Massachusetts
7	E0000007	D0000007	vprandoni6	HOzls9	Vinny Prandoni	Intern	Georgia
8	E0000008	D0000008	jtabart7	iWP0bNmUj	Joy Tabart	Junior	Missouri
9	E0000009	D0000009	pflather8	yODcnKC	Pearla Flather	Junior	West Virginia
10	E0000010	D0000010	mcapstick9	HjBjobZT4hJ9	Marius Capstick	Intern	Alaska
11	E0000011	D0000011	heastama	aKGqXqjyw4	Hillary Eastam	Intern	Florida
12	E0000012	D0000001	hvarfolomeevb	ehGnuZh7	Halsey Varfolomeev	Intern	Maryland
13	E0000013	D0000002	mluscombe	1TNXbG	Marie-Jeanne Luscombe	Senior	Missouri
14	E0000014	D0000003	cbrunoned	aqi5qrWZH	Carlyle Brunone	Junior	Massachusetts
15	E0000015	D0000004	pzieme	C8FCDxTyKL9	Pauly Zieme	Senior	California
16	E0000016	D0000005	afeldbergerf	4ShGJK290zAT	Albie Feldberger	Senior	Colorado
17	E0000017	D0000006	cbissattg	3AnSoB	Carissa Bissatt	Junior	Wyoming
18	E0000018	D0000007	apatelyh	pgB2dGTJKAH	Audrye Patey	Junior	California
19	E0000019	D0000008	smorpheyi	3ZKJj53	Salvador Morphey	Intern	Colorado
20	E0000020	D0000009	asimoj	JyKsTTnJq	Amalie Simo	Junior	Minnesota
21	E0000021	D0000010	cwhettonk	qk8PUJG6DU	Colin Whetton	Junior	North Carolina
22	E0000022	D0000011	yshropsheirl	IOJ93D53	Yolanda Shropsheir	Intern	Texas

### 3. Employee Contact

```
create table "Employee Contact"
(
    "Emp_ID" varchar(100) not null,
    "Contact" varchar(100) not null,

    constraint "PK_EmployeeContact" (Emp_ID, Contact) primary key ("Emp_ID", "Contact"),
    constraint "FK_Employee Emp_ID" foreign key ("Emp_ID") references "Employee" ("Emp_ID")
        match simple
        on update cascade
        on delete cascade
        not valid
);
```

Query Editor    Query History

```
1 SELECT * FROM "TMS"."Employee Contact"
2 ORDER BY "Emp_ID" ASC, "Contact" ASC
```

Data Output   Explain   Messages   Notifications

	Emp_ID [PK] character varying (100)	Contact [PK] character varying (100)
1	E0000001	421-305-7751
2	E0000001	510-280-1367
3	E0000002	257-762-9383
4	E0000002	863-346-7784
5	E0000003	107-364-6280
6	E0000003	861-776-3585
7	E0000004	397-907-8530
8	E0000004	957-786-5447
9	E0000005	422-846-7017
10	E0000005	794-281-2383
11	E0000006	434-305-0069
12	E0000006	784-885-8878
13	E0000007	736-154-2999
14	E0000007	779-400-0192
15	E0000008	611-335-9494
16	E0000008	883-952-4160
17	E0000009	593-579-7414
18	E0000009	730-281-3580
19	E0000010	482-989-6573
20	E0000010	610-730-6125
21	E0000011	242-750-1480

## 4. Project Manager

```
create table "Project Manager"
(
    "PM_ID" varchar(100) not null,
    "Achievement" text,

    constraint "PK_ProjectManager PM_ID" primary key ("PM_ID"),
    constraint "FK_Employee Emp_ID" foreign key ("PM_ID") references "Employee" ("Emp_ID")
        match simple
        on update cascade
        on delete cascade
        not valid
);
```

Query Editor    Query History

```
1 SELECT * FROM "TMS"."Project Manager"
2 ORDER BY "PM_ID" ASC
```

Data Output    Explain    Messages    Notifications

PM_ID [PK] character varying (100)	Achievement text
1 E0000011	Vivamus in felis eu sapien cursus vestibulum.
2 E0000022	Ut tellus. Nulla ut erat id mauris vulputate elementum.
3 E0000033	Proin leo odio, porttitor id, consequat in, consequat ut, nulla.
4 E0000044	Integer non velit.
5 E0000055	Integer tincidunt ante vel ipsum.
6 E0000066	Nullam porttitor lacus at turpis. Donec posuere metus vitae ipsum.
7 E0000077	Nulla justo. Aliquam quis turpis eget elit sodales scelerisque.
8 E0000088	In hac habitasse platea dictumst. Aliquam augue quam, sollicitudin vitae, consectetuer eget, rutrum at, lorem.
9 E0000099	Phasellus in felis. Donec semper sapien a libero.
10 E0000191	Vivamus vestibulum sagittis sapien.
11 E0000192	Fusce lacus purus, aliquet at, feugiat non, pretium quis, lectus. Suspendisse potenti.
12 E0000193	Vestibulum rutrum rutrum neque. Aenean auctor gravida sem.
13 E0000194	In tempor, turpis nec euismod scelerisque, quam turpis adipiscing lorem, vitae mattis nibh ligula nec sem. Duis aliquam convallis nunc.
14 E0000195	Aliquam augue quam, sollicitudin vitae, consectetuer eget, rutrum at, lorem.
15 E0000196	Maecenas ut massa quis augue luctus tincidunt. Nulla mollis molestie lorem.
16 E0000197	Sed accumsan felis.
17 E0000198	Curabitur convallis.
18 E0000199	In tempor, turpis nec euismod scelerisque, quam turpis adipiscing lorem, vitae mattis nibh ligula nec sem. Duis aliquam convallis nunc.
19 E0000200	Nullam varius. Nulla facilisi.
20 E0000201	Integer non velit. Donec diam neque, vestibulum eget, vulputate ut, ultrices vel, augue.

## 5. Client

```
create table "Client"
(
    "CL_ID" varchar(100) not null,
    "Username" varchar(200) not null,
    "Password" varchar(200) not null,
    "Name" varchar(500) not null,
    "City" varchar(200) not null,
    "State" varchar(200) not null,
    "Email" varchar(200) not null,
    "Company" varchar(200),

    constraint "PK_Client CL_ID" primary key ("CL_ID"),
    constraint "unique_client_username" unique ("Username")
);
```

Query Editor    Query History

```
1 SELECT * FROM "TMS"."Client"
2 ORDER BY "CL_ID" ASC
```

Data Output   Explain   Messages   Notifications

	CL_ID [PK] character varying (100)	Username character varying (200)	Password character varying (200)	Name character varying (500)	City character varying (200)	State character varying (200)	Email character varying (200)
1	C0000001	amakeswell0	ZWWS3uwB	Antonio Makeswell	Brooklyn	New York	amakeswell0@bloglovin.com
2	C0000002	illiddington1	dkNvYXoW4	Izak Liddington	Sacramento	California	illiddington1@ameblo.jp
3	C0000003	cyggo2	ffffgAv	Charley Yggo	Birmingham	Alabama	cyggo2@google.cn
4	C0000004	ekeneford3	l25F7sD95uIe	Ellary Keneford	Grand Junction	Colorado	ekeneford3@liveinternet.ru
5	C0000005	rflowden4	stflr2GVpnkZ	Rosaleen Fowden	Bloomington	Indiana	rflowden4@japanpost.jp
6	C0000006	lwitson5	7x15oOY	Leona Witson	Long Beach	California	lwitson5@ovh.net
7	C0000007	dchester6	h2hwhc	Danika Chester	Oklahoma City	Oklahoma	dchester6@upenn.edu
8	C0000008	smattimoe7	HFO058	Stesha Mattimoe	Baltimore	Maryland	smattimoe7@unicef.org
9	C0000009	fdomici18	9bhQHO	Fayina Dominici	Tucson	Arizona	fdomici18@umn.edu
10	C0000010	adickman9	8QZUNQS8X	Amandi Dickman	Huntsville	Alabama	adickman9@netvibes.com
11	C0000011	kdavidowsky	J54j5K3j1	Kennie Davidowsky	Tallahassee	Florida	kdavidowsky@kickstarter.com
12	C0000012	llattosb	BXDLrHh3	Lisha Lattos	Dayton	Ohio	llattosb@cocolog-nifty.com
13	C0000013	hcriginc	yubShUJB3No	Hart Crigin	Fresno	California	hcriginc@networksolutions.com
14	C0000014	gsharplyd	jdQSEYPd1A7d	Gianna Sharply	San Jose	California	gsharplyd@t.co
15	C0000015	dsandlande	FspPeUcc2	Dre Sandland	Punta Gorda	Florida	dsandlande@taobao.com
16	C0000016	ugeramf	DnGbaqZ7pz	Ursula Geram	Lincoln	Nebraska	ugeramf@g.co
17	C0000017	cvasilovg	F6SAIQ	Case Vasilov	Fairbanks	Alaska	cvasilovg@bloglovin.com
18	C0000018	Iclemenceauh	j6fgBiU	Lavinie Clemenceau	Kansas City	Missouri	Iclemenceauh@google.cn
19	C0000019	mburni	IDJ1fp6JeI7	Mada Burn	Washington	District of Columbia	mburni@si.edu
20	C0000020	smaccrannj	nrhEHJzb4	Seka MacCrann	Merrifield	Virginia	smaccrannj@vistaprint.com
21	C0000021	abarkshirek	6ql6McGE	Amalita Barkshire	El Paso	Texas	abarkshirek@wordpress.org
22	C0000022	reckl	yLHCBIH	Robinett Eck	Chicago	Illinois	reckl@google.co.jp

## 6. Client Contact

```
create table "Client Contact"
(
    "CL_ID" varchar(100) not null,
    "Contact" varchar(100) not null,
    constraint "PK_ClientContact" (CL_ID, Contact) primary key ("CL_ID", "Contact"),
    constraint "FK_Client CL_ID" foreign key ("CL_ID") references "Client" ("CL_ID")
        match simple
        on update cascade
        on delete cascade
        not valid
);
```

Query Editor    Query History

```
1 SELECT * FROM "TMS"."Client Contact"
2 ORDER BY "CL_ID" ASC, "Contact" ASC
```

Data Output   Explain   Messages   Notifications

	CL_ID [PK] character varying (100)	Contact [PK] character varying (100)
1	C0000001	586-678-3826
2	C0000001	763-514-0374
3	C0000001	804-385-5157
4	C0000002	475-601-7097
5	C0000002	513-172-3862
6	C0000003	109-103-6104
7	C0000003	231-327-0837
8	C0000003	697-870-1271
9	C0000004	245-376-8896
10	C0000004	290-713-1091
11	C0000005	146-855-7362
12	C0000005	193-472-5061
13	C0000005	326-506-6531
14	C0000006	200-952-1347
15	C0000006	726-338-3314
16	C0000007	430-769-9081
17	C0000007	631-356-9383
18	C0000007	678-215-3737
19	C0000008	413-653-5002
20	C0000008	921-861-2196
21	C0000009	455-502-7294
22	C0000009	859-173-8127

## 7. Project

```

create table "Project"
(
    "Project_ID" varchar(100) not null,
    "PM_ID" varchar(100) not null,
    "CL_ID" varchar(100) not null,
    "Name" varchar(500) not null,
    "Budget" bigint not null check ("Budget" > 0),
    "Start_date" date not null,
    "End_date" date,
    "Progress" smallint,
    "Status" varchar(200),
    "Expected_Deadline" date,
    "Meet_Summary" text,
    "Expected_Cost" bigint,
    constraint "PK_Project Project_ID" primary key ("Project_ID"),
    constraint "FK_ProjectManager PM_ID" foreign key ("PM_ID") references "Project Manager" ("PM_ID")
        match simple
        on update cascade
        on delete cascade
        not valid,
    constraint "FK_Client CL_ID" foreign key ("CL_ID") references "Client" ("CL_ID")
        match simple
        on update cascade
        on delete cascade
        not valid,
    check ("Status" in ('Approved', 'Pending', 'Improve'))
);

```

Query Editor    Query History

```

1 SELECT * FROM "TMS"."Project"
2 ORDER BY "Project_ID" ASC

```

Data Output    Explain    Messages    Notifications

	Project_ID [PK] character varying (100)	PM_ID character varying (100)	CL_ID character varying (100)	Name character varying (500)	Budget bigint	Start_date date	End_date date	Progress smallint	Status character varying (200)
1	P0000001	E0000011	C0000001	Baxter Whitcher	2542004	2010-09-04	2015-09-10	87	Approved
2	P0000002	E0000022	C0000002	Agretha Taillard	5179852	2011-04-02	2019-08-04	71	Pending
3	P0000003	E0000033	C0000003	Kim Sommerlie	2255144	2010-01-06	2018-01-02	74	Pending
4	P0000004	E0000044	C0000004	Edi Marle	3808743	2012-12-20	2021-07-25	12	Approved
5	P0000005	E0000055	C0000005	Alissa Coal	9155680	2010-10-15	2018-09-28	96	Improve
6	P0000006	E0000066	C0000006	Keary Postlewhite	6907652	2010-05-17	2017-10-29	5	Pending
7	P0000007	E0000077	C0000007	Gwenora Hyde-Chambers	1137501	2012-06-10	2019-09-30	47	Approved
8	P0000008	E0000088	C0000008	Celinda Rodenburgh	6795926	2010-02-02	2016-07-27	43	Improve
9	P0000009	E0000099	C0000009	Arda Tipperton	349967	2013-10-30	2016-11-11	86	Approved
10	P0000010	E0000191	C0000010	Devondra Hook	4889230	2010-02-18	2017-06-14	65	Improve
11	P0000011	E0000192	C0000011	Arluene Will	5833538	2012-05-09	2017-11-17	93	Pending
12	P0000012	E0000193	C0000012	Jonell Pane	690905	2013-09-11	2015-04-20	14	Approved
13	P0000013	E0000194	C0000013	Tootsie Furbank	4235407	2013-03-29	2015-12-06	97	Approved
14	P0000014	E0000195	C0000014	Pammy Letchford	5730590	2014-11-06	2015-10-22	92	Approved
15	P0000015	E0000196	C0000015	Dean Pandya	294556	2010-08-23	2020-07-27	25	Pending
16	P0000016	E0000197	C0000016	Bradan Peasee	1013750	2013-02-08	2017-03-12	52	Improve
17	P0000017	E0000198	C0000017	Mathias Confort	1384437	2011-04-10	2015-10-10	89	Approved
18	P0000018	E0000199	C0000018	Margaret Goold	7378469	2011-03-26	2020-05-01	43	Improve
19	P0000019	E0000200	C0000019	Constantino Bicheno	1554935	2011-07-11	2015-03-10	85	Improve
20	P0000020	E0000201	C0000020	Melania Deddum	8148778	2014-09-21	2018-04-11	59	Approved
21	P0000021	E0000011	C0000021	Tad Gregoriou	6737621	2011-12-25	2019-11-30	49	Pending
22	P0000022	E0000022	C0000022	Gannon O'Lenane	7574018	2011-04-16	2017-01-09	61	Pending

## 8. Task

```

create table "Task"
(
    "Task_ID" varchar(100) not null,
    "Project_ID" varchar(100) not null,
    "Name" varchar(500) not null,
    "Cost" bigint,
    "Start_date" date not null,
    "End_date" date,
    "Progress" smallint,
    "Status" varchar(200),

    constraint "PK_Task" (Project_ID, Task_ID) primary key ("Project_ID", "Task_ID"),
    constraint "FK_Project Project_ID" foreign key ("Project_ID") references "Project" ("Project_ID")
        match simple
        on update cascade
        on delete cascade
        not valid,
    check ("Status" in ('Approved', 'Pending', 'Improve'))
);

```

Query Editor    Query History

```

1 SELECT * FROM "TMS"."Task"
2 ORDER BY "Task_ID" ASC, "Project_ID" ASC

```

Data Output   Explain   Messages   Notifications

	Task_ID [PK] character varying (100)	Project_ID [PK] character varying (100)	Name character varying (500)	Cost bigint	Start_date date	End_date date	Progress smallint	Status character varying (200)	
1	T0000001	P0000001	Merops nubicus	40863	2010-09-04	2015-09-10	99	Approved	
2	T0000001	P0000002	Cordylus giganteus	28058	2010-02-02	2016-07-27	19	Pending	
3	T0000001	P0000003	Diomedea irrorata	87574	2014-11-06	2015-10-22	76	Approved	
4	T0000001	P0000004	Tachyglossus aculeatus	71311	2011-03-26	2020-05-01	38	Pending	
5	T0000001	P0000005	Colobus guerza	94994	2013-05-06	2017-11-23	88	Approved	
6	T0000001	P0000006	Oxybelis fulgidus	68047	2014-10-19	2017-03-06	73	Approved	
7	T0000001	P0000007	Dicrastonyx groenlandicus	81352	2014-09-16	2017-01-19	97	Approved	
8	T0000001	P0000008	Paradoxurus hermaphroditus	51821	2012-12-09	2020-08-13	76	Approved	
9	T0000001	P0000009	Himantopus himantopus	75725	2010-03-03	2019-05-25	59	Approved	
10	T0000001	P0000010	Macropus giganteus	59288	2014-10-06	2015-04-03	7	Improve	
11	T0000001	P0000011	Aegypius occipitalis	92890	2011-02-01	2020-03-07	25	Approved	
12	T0000001	P0000012	Agkistrodon piscivorus	97972	2014-09-28	2015-02-02	35	Approved	
13	T0000001	P0000013	Otocyon megalotis	37977	2011-10-14	2019-08-23	94	Pending	
14	T0000001	P0000014	Pituophis melanoleucus	93959	2012-11-24	2017-10-31	66	Approved	
15	T0000001	P0000015	Isoodon obesulus	35428	2011-07-11	2015-08-30	98	Pending	
16	T0000001	P0000016	Upupa epops	19926	2012-05-30	2018-08-22	28	Improve	
17	T0000001	P0000017	Sitta canadensis	52285	2011-04-22	2018-04-04	94	Improve	
18	T0000001	P0000018	Podargus strigoides	49348	2013-07-25	2015-10-29	18	Improve	
19	T0000001	P0000019	Equus hemionus	13548	2014-02-26	2018-05-14	44	Approved	
20	T0000001	P0000020	Vulpes vulpes	70916	2010-11-28	2015-11-25	29	Pending	
21	T0000001	P0000021	Colobus guerza	23045	2012-03-19	2018-12-09	74	Approved	
22	T0000001	P0000022	Rangifer tarandus	52367	2012-02-02	2019-06-16	55	Pending	
~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~	~~~~~	~~~~~

## 9. Team

```
create table "Team"
(
    "Project_ID" varchar(100) not null,
    "Task_ID" varchar(100) not null,
    "Member_ID" varchar(100) not null,

    constraint "PK_Team" (Project_ID, Member_ID) primary key("Project_ID", "Member_ID"),
    constraint "FK_Task" (Project_ID, Task_ID) foreign key ("Project_ID", "Task_ID") references "Task" ("Project_ID", "Task_ID")
        match simple
        on update cascade
        on delete cascade
        not valid,
    constraint "FK_Employee Member_ID" foreign key ("Member_ID") references "Employee" ("Emp_ID")
        match simple
        on update cascade
        on delete cascade
        not valid
);
```

Query Editor    Query History

```
1 SELECT * FROM "TMS"."Team"
2 ORDER BY "Project_ID" ASC, "Member_ID" ASC
```

Data Output    Explain    Messages    Notifications

	Project_ID [PK] character varying (100)	Task_ID character varying (100)	Member_ID [PK] character varying (100)
1	P0000001	T0000001	E0000006
2	P0000001	T0000002	E0000011
3	P0000001	T0000003	E0000017
4	P0000001	T0000004	E0000022
5	P0000001	T0000005	E0000028
6	P0000001	T0000006	E0000033
7	P0000001	T0000007	E0000039
8	P0000002	T0000001	E0000044
9	P0000002	T0000002	E0000050
10	P0000002	T0000003	E0000055
11	P0000002	T0000004	E0000061
12	P0000002	T0000005	E0000066
13	P0000002	T0000006	E0000072
14	P0000003	T0000001	E0000077
15	P0000003	T0000002	E0000083
16	P0000003	T0000003	E0000088
17	P0000003	T0000004	E0000094
18	P0000004	T0000001	E0000099
19	P0000004	T0000002	E0000100
20	P0000004	T0000003	E0000101
21	P0000004	T0000004	E0000102

## 10. Team Leader

```
create table "Team_Leader"
(
    "Project_ID" varchar(100) not null,
    "Task_ID" varchar(100) not null,
    "TL_ID" varchar(100) not null,

    constraint "PK_Team_Leader" primary key("Project_ID", "Task_ID"),
    constraint "FK_Task" (Project_ID, Task_ID) foreign key ("Project_ID", "Task_ID") references "Task" ("Project_ID", "Task_ID")
        match simple
        on update cascade
        on delete cascade
        not valid,
    constraint "FK_Employee" TL_ID" foreign key ("TL_ID") references "Employee" ("Emp_ID")
        match simple
        on update cascade
        on delete cascade
        not valid
);
```

Query Editor    Query History

```
1 SELECT * FROM "TMS"."Team_Leader"
2 ORDER BY "Project_ID" ASC, "Task_ID" ASC
```

Data Output    Explain    Messages    Notifications

	Project_ID [PK] character varying (100)	Task_ID [PK] character varying (100)	TL_ID character varying (100)
1	P0000001	T0000001	E0000006
2	P0000001	T0000002	E0000006
3	P0000001	T0000003	E0000006
4	P0000001	T0000004	E0000006
5	P0000001	T0000005	E0000006
6	P0000001	T0000006	E0000006
7	P0000001	T0000007	E0000006
8	P0000002	T0000001	E0000044
9	P0000002	T0000002	E0000044
10	P0000002	T0000003	E0000044
11	P0000002	T0000004	E0000044
12	P0000002	T0000005	E0000044
13	P0000002	T0000006	E0000044
14	P0000003	T0000001	E0000077
15	P0000003	T0000002	E0000077
16	P0000003	T0000003	E0000077
17	P0000003	T0000004	E0000077
18	P0000004	T0000001	E0000099
19	P0000004	T0000002	E0000099
20	P0000004	T0000003	E0000099
21	P0000004	T0000004	E0000099
22	P0000004	T0000005	E0000099

## 11. Request

```
create table "Request"
(
    "PM_ID" varchar(100) not null,
    "CL_ID" varchar(100) not null,
    "Date" date not null,

    constraint "PK_Request" (PM_ID, CL_ID) primary key("PM_ID", "CL_ID"),
    constraint "FK_ProjectManager PM_ID" foreign key ("PM_ID") references "Project Manager" ("PM_ID")
        match simple
        on update cascade
        on delete cascade
        not valid,
    constraint "FK_Client CL_ID" foreign key ("CL_ID") references "Client" ("CL_ID")
        match simple
        on update cascade
        on delete cascade
        not valid
);
```

Query Editor    Query History

```
1 SELECT * FROM "TMS"."Request"
2 ORDER BY "PM_ID" ASC, "CL_ID" ASC
```

Data Output   Explain   Messages   Notifications

	PM_ID [PK] character varying (100)	CL_ID [PK] character varying (100)	Date date
1	E0000011	C0000001	2015-09-10
2	E0000011	C0000002	2019-08-04
3	E0000011	C0000003	2018-01-02
4	E0000011	C0000004	2021-07-25
5	E0000011	C0000005	2018-09-28
6	E0000011	C0000006	2017-10-29
7	E0000022	C0000007	2019-09-30
8	E0000022	C0000008	2016-07-27
9	E0000022	C0000009	2016-11-11
10	E0000022	C0000010	2017-06-14
11	E0000022	C0000011	2017-11-17
12	E0000022	C0000012	2015-04-20
13	E0000033	C0000013	2015-12-06
14	E0000033	C0000014	2015-10-22
15	E0000033	C0000015	2020-07-27
16	E0000033	C0000016	2017-03-12
17	E0000033	C0000017	2015-10-10
18	E0000033	C0000018	2020-05-01
19	E0000044	C0000019	2015-03-10
20	E0000044	C0000020	2018-04-11
21	E0000044	C0000021	2019-11-30
22	E0000044	C0000022	2017-01-09
23	E0000044	C0000023	2018-02-10

## 12. Announcement

```
create table "Announcement"
(
    "An_ID" varchar(100) not null,
    "PM_ID" varchar(100) not null,
    "Announcement" text,
    "Date" date not null,

    constraint "PK_Announcement An_ID" primary key("An_ID"),
    constraint "FK_ProjectManager PM_ID" foreign key ("PM_ID") references "Project Manager" ("PM_ID")
        match simple
        on update cascade
        on delete cascade
        not valid
);
```

Query Editor    Query History

```
1 SELECT * FROM "TMS"."Announcement"
2 ORDER BY "An_ID" ASC
```

Data Output   Explain   Messages   Notifications

An_ID [PK] character varying (100)	PM_ID character varying (100)	Announcement text
1 A0000001	E0000011	Nulla justo. Aliquam quis turpis eget elit sodales scelerisque. Mauris sit amet eros. Suspendisse accumsan tortor quis turpis.
2 A0000002	E0000022	Fusce congue, diam id ornare imperdiet, sapien urna pretium nisl, ut volutpat sapien arcu sed augue. Aliquam erat volutpat. In congue. Etiam justo.
3 A0000003	E0000033	Cras mi pede, malesuada in, imperdiet et, commodo vulputate, justo. In blandit ultrices enim. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
4 A0000004	E0000044	Mauris ullamcorper purus sit amet nulla. Quisque arcu libero, rutrum ac, lobortis vel, dapibus at, diam. Nam tristique tortor eu pede.
5 A0000005	E0000055	Proin interdum mauris non ligula pellentesque ultrices. Phasellus id sapien in sapien iaculis congue. Vivamus metus arcu, adipiscing molestie, hendrerit.
6 A0000006	E0000066	Aliquam quis turpis eget elit sodales scelerisque. Mauris sit amet eros. Suspendisse accumsan tortor quis turpis. Sed ante.
7 A0000007	E0000077	Nulla facilisi. Cras non velit nec nisi vulputate nonummy. Maecenas tincidunt lacus at velit. Vivamus vel nulla eget eros elementum pellentesque.
8 A0000008	E0000088	Cras pellentesque volutpat dui. Maecenas tristique, est et tempus semper, est quam pharetra magna, ac consequat metus sapien ut nunc. Vestibulum.
9 A0000009	E0000099	Integer ac leo. Pellentesque ultrices mattis odio. Donec vitae nisi.
10 A0000010	E0000191	Mauris enim leo, rhoncus sed, vestibulum sit amet, cursus id, turpis. Integer aliquet, massa id lobortis convallis, tortor risus dapibus augue, vel accumsan.
11 A0000011	E0000192	Vestibulum sed magna at nunc commodo placerat. Praesent blandit. Nam nulla.
12 A0000012	E0000193	Donec vitae nisi. Nam ultrices, libero non mattis pulvinar, nulla pede ullamcorper augue, a suscipit nulla elit ac nulla. Sed vel enim sit amet nunc viverra.
13 A0000013	E0000194	Sed ante. Vivamus tortor. Duis mattis egestas metus.
14 A0000014	E0000195	Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nulla dapibus dolor vel est. Donec odio justo, sollicitudin ut, suscipit.
15 A0000015	E0000196	Quisque porta volutpat erat. Quisque erat eros, viverra eget, congue eget, semper rutrum, nulla. Nunc purus. Phasellus in felis.
16 A0000016	E0000197	Quisque erat eros, viverra eget, congue eget, semper rutrum, nulla. Nunc purus. Phasellus in felis. Donec semper sapien a libero.
17 A0000017	E0000198	Nullam molestie nibh in lectus. Pellentesque at nulla. Suspendisse potenti.
18 A0000018	E0000199	In tempor, turpis nec euismod scelerisque, quam turpis adipiscing lorem, vitae mattis nibh ligula nec sem. Duis aliquam convallis nunc. Proin at turpis.
19 A0000019	E0000200	Pellentesque at nulla. Suspendisse potenti. Cras in purus eu magna vulputate luctus.
20 A0000020	E0000201	Suspendisse potenti. In eleifend quam a odio. In hac habitasse platea dictumst. Maecenas ut massa quis augue luctus tincidunt.
21 A0000021	E0000011	Vivamus vel nulla eget eros elementum pellentesque. Quisque porta volutpat erat. Quisque erat eros, viverra eget, congue eget, semper rutrum, nulla.
22 A0000022	E0000022	Morbi non quam nec dui luctus rutrum. Nulla tellus. In sagittis dui vel nisl. Duis ac nibh.

## 13. Resource

```

create table "Resource"
(
    "Rs_ID" varchar(100) not null,
    "PM_ID" varchar(100) not null,
    "Project_ID" varchar(100) not null,
    "Description" text,
    "ID" bigint not null unique,
    "Password" varchar(200) not null,
    "Issue_date" date,
    "Return_date" date,
    constraint "PK_Resource Rs_ID" primary key("Rs_ID"),
    constraint "FK_ProjectManager PM_ID" foreign key ("PM_ID") references "Project Manager" ("PM_ID")
        match simple
        on update cascade
        on delete cascade
        not valid,
    constraint "FK_Project Project_ID" foreign key ("Project_ID") references "Project" ("Project_ID")
        match simple
        on update cascade
        on delete cascade
        not valid
);

```

Query Editor    Query History

```

1 SELECT * FROM "TMS"."Resource"
2 ORDER BY "Rs_ID" ASC

```

Data Output   Explain   Messages   Notifications

	Rs_ID [PK] character varying (100)	PM_ID character varying (100)	Project_ID character varying (100)	Description text
1	R0000001	E0000011	P0000001	Ut tellus. Nulla ut erat id mauris vulputate elementum. Nullam varius.
2	R0000002	E0000022	P0000002	Praesent id massa id nisl venenatis lacinia. Aenean sit amet justo. Morbi ut odio.
3	R0000003	E0000033	P0000003	Maecenas rhoncus aliquam lacus. Morbi quis tortor id nulla ultrices aliquet. Maecenas leo odio, condimentum id, luctus
4	R0000004	E0000044	P0000004	Pellentesque ultrices mattis odio. Donec vitae nisi. Nam ultrices, libero non mattis pulvinar, nulla pede ullamcorper augi
5	R0000005	E0000055	P0000005	Donec diam neque, vestibulum eget, vulputate ut, ultrices vel, augue. Vestibulum ante ipsum primis in fauibus orci luct
6	R0000006	E0000066	P0000006	Aliquam erat volutpat. In congue. Etiam justo. Etiam pretium iaculis justo.
7	R0000007	E0000077	P0000007	Donec ut dolor. Morbi vel lectus in quam fringilla rhoncus. Mauris enim leo, rhoncus sed, vestibulum sit amet, cursus id,
8	R0000008	E0000088	P0000008	In blandit ultrices enim. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin interdum mauris non ligula pell
9	R0000009	E0000099	P0000009	Donec quis orci eget orci vehicula condimentum. Curabitur in libero ut massa volutpat convallis. Morbi odio odio, eleme
10	R0000010	E0000191	P0000010	Donec quis orci eget orci vehicula condimentum. Curabitur in libero ut massa volutpat convallis. Morbi odio odio, eleme
11	R0000011	E0000192	P0000011	Nulla tellus. In sagittis dui vel nisl. Duis ac nibh. Fusce lacus purus, aliquet at, feugiat non, pretium quis, lectus.
12	R0000012	E0000193	P0000012	Curabitur gravida nisi at nibh. In hac habitasse platea dictumst. Aliquam augue quam, sollicitudin vitæ, consectetur et
13	R0000013	E0000194	P0000013	Nulla mollis molestie lorem. Quisque ut erat. Curabitur gravida nisi at nibh.
14	R0000014	E0000195	P0000014	Duis ac nibh. Fusce lacus purus, aliquet at, feugiat non, pretium quis, lectus. Suspendisse potenti.
15	R0000015	E0000196	P0000015	Maecenas leo odio, condimentum id, luctus nec, molestie sed, justo. Pellentesque viverra pede ac diam. Cras pellentes
16	R0000016	E0000197	P0000016	Cras non velit nec nisi vulputate nonummy. Maecenas tincidunt lacus at velit. Vivamus vel nulla eget eros elementum p
17	R0000017	E0000198	P0000017	In sagittis dui vel nisl. Duis ac nibh. Fusce lacus purus, aliquet at, feugiat non, pretium quis, lectus.
18	R0000018	E0000199	P0000018	Praesent blandit. Nam nulla. Integer pede justo, lacinia eget, tincidunt eget, tempus vel, pede.
19	R0000019	E0000200	P0000019	Integer tincidunt ante vel ipsum. Praesent blandit lacinia erat. Vestibulum sed magna at nunc commodo placerat. Praes
20	R0000020	E0000201	P0000020	Praesent blandit. Nam nulla. Integer pede justo, lacinia eget, tincidunt eget, tempus vel, pede.
21	R0000021	E0000011	P0000021	Duis at velit eu est congue elementum. In hac habitasse platea dictumst. Morbi vestibulum, velit id pretium iaculis, dian
22	R0000022	E0000022	P0000022	Vestibulum sed magna at nunc commodo placerat. Praesent blandit. Nam nulla.

## 14. Log

```
create table "Log"
(
    "Log_ID" bigint not null,
    "Log_Info" text,
    "Date" date not null,
    constraint "PK_Log_Log_ID" primary key("Log_ID")
);
```

Query Editor    Query History

```
1  SELECT * FROM "TMS"."Log"
2  ORDER BY "Log_ID" ASC
```

Data Output   Explain   Messages   Notifications

	Log_ID	Log_Info
1	333578	Donec diam neque, vestibulum eget, vulputate ut, ultrices vel, augue. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Donec pharetra, magna vestibulum aliquet.
2	359608	Praesent blandit lacinia erat. Vestibulum sed magna at nunc commodo placerat. Praesent blandit. Nam nulla.
3	558243	In est risus, auctor sed, tristique in, tempus sit amet, sem. Fusce consequat. Nulla nisl.
4	657262	In est risus, auctor sed, tristique in, tempus sit amet, sem. Fusce consequat. Nulla nisl. Nunc nisl.
5	699316	Morbi a ipsum. Integer a nibh. In quis justo. Maecenas rhoncus aliquam lacus.
6	1111111	vcvxcvxc gdgdfgfd ndffd
7	1667329	Vestibulum quam sapien, varius ut, blandit non, interdum in, ante. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Duis faucibus accumsan odio. Curabitur convallis.
8	1988646	Fusce posuere felis sed lacus. Morbi sem mauris, laoreet ut, rhoncus aliquet, pulvinar sed, nisl. Nunc rhoncus dui vel sem.
9	2054259	Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Donec pharetra, magna vestibulum aliquet ultrices, erat tortor sollicitudin mi, sit amet lobortis sapien sapien no.
10	2225855	Quisque porta volutpat erat. Quisque erat eros, viverra eget, congue eget, semper rutrum, nulla. Nunc purus. Phasellus in felis.
11	3004235	Suspendisse potenti. Cras in purus eu magna vulputate luctus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Vivamus vestibulum sagittis sapien.
12	3074752	Donec diam neque, vestibulum eget, vulputate ut, ultrices vel, augue. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Donec pharetra, magna vestibulum aliquet.
13	3417201	Duis mattis egestas metus. Aenean fermentum. Donec ut mauris eget massa tempor convallis.
14	3483423	Duis mattis egestas metus. Aenean fermentum. Donec ut mauris eget massa tempor convallis. Nulla neque libero, convallis eget, eleifend luctus, ultricies eu, nibh.
15	4698253	Duis bibendum. Morbi non quam nec dui luctus rutrum. Nulla tellus. In sagittis dui vel nisl.
16	5831941	Integer ac leo. Pellentesque ultrices mattis odio. Donec vitae nisi. Nam ultrices, libero non mattis pulvinar, nulla pede ullamcorper augue, a suscipit nulla elit ac nulla.
17	6272982	Suspendisse potenti. Nullam porttitor lacus at turpis. Donec posuere metus vitae ipsum. Aliquam non mauris.
18	6602779	Nulla mollis molestie lorem. Quisque ut erat. Curabitur gravida nisi at nibh.
19	8060420	In hac habitasse platea dictumst. Etiam faucibus cursus urna. Ut tellus.
20	8101307	Quisque ut erat. Curabitur gravida nisi at nibh. In hac habitasse platea dictumst. Aliquam augue quam, sollicitudin vitae, consectetuer eget, rutrum at, lorem.
21	9408821	Morbi ut odio. Cras mi pede, malesuada in, imperdiet et, commodo vulputate, justo. In blandit ultrices enim. Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
22	9423419	Nulla facilisi. Cras non velit nec nisi vulputate nonummy. Maecenas tincidunt lacus at velit.

## 15. Update

```

create table "Updates"
(
    "Emp_ID" varchar(100) not null,
    "Project_ID" varchar(100) not null,
    "Task_ID" varchar(100) not null,
    "Log_ID" bigint not null,

    constraint "PK_Updates" (Emp_ID, Project_ID, Task_ID, Log_ID) primary key("Emp_ID", "Project_ID", "Task_ID", "Log_ID"),
    constraint "FK_Employee Emp_ID" foreign key ("Emp_ID") references "Employee" ("Emp_ID")
        match simple
        on update cascade
        on delete cascade
        not valid,
    constraint "FK_Task (Project_ID, Task_ID)" foreign key ("Project_ID", "Task_ID") references "Task" ("Project_ID", "Task_ID")
        match simple
        on update cascade
        on delete cascade
        not valid,
    constraint "FK_Log Log_ID" foreign key ("Log_ID") references "Log" ("Log_ID")
        match simple
        on update cascade
        on delete cascade
        not valid
);

```

Query Editor    Query History

```

1 SELECT * FROM "TMS"."Updates"
2 ORDER BY "Emp_ID" ASC, "Project_ID" ASC, "Task_ID" ASC, "Log_ID" ASC

```

Data Output    Explain    Messages    Notifications

	Emp_ID [PK] character varying (100)	Project_ID [PK] character varying (100)	Task_ID [PK] character varying (100)	Log_ID [PK] bigint
1	E0000006	P0000001	T0000001	98032122
2	E0000006	P0000024	T0000002	67325035
3	E0000006	P0000044	T0000004	72587619
4	E0000011	P0000001	T0000002	25546588
5	E0000011	P0000024	T0000003	24519360
6	E0000011	P0000045	T0000001	30848513
7	E0000017	P0000001	T0000003	95647018
8	E0000017	P0000024	T0000004	72984685
9	E0000017	P0000045	T0000002	64839083
10	E0000022	P0000001	T0000004	96844632
11	E0000022	P0000025	T0000001	29292762
12	E0000028	P0000001	T0000005	60350837
13	E0000028	P0000025	T0000002	43004759
14	E0000033	P0000001	T0000006	37258961
15	E0000033	P0000025	T0000003	95353923
16	E0000039	P0000001	T0000007	34758389
17	E0000039	P0000025	T0000004	31968616
18	E0000044	P0000002	T0000001	84514857
19	E0000044	P0000025	T0000005	23491804
20	E0000050	P0000002	T0000002	9408821
21	E0000050	P0000025	T0000006	49787430
22	E0000055	P0000002	T0000003	90089920

## 16. Query

```

create table "Query"
(
    "Query_ID" varchar(100) not null,
    "Emp_ID" varchar(100) not null,
    "Query" text,
    "Status" varchar(200),
    "Response" text,

    constraint "PK_Query Query_ID" primary key("Query_ID"),
    constraint "FK_Employee Emp_ID" foreign key ("Emp_ID") references "Employee" ("Emp_ID")
        match simple
        on update cascade
        on delete cascade
        not valid,
    check ("Status" in ('Pending', 'Solved'))
)

```

Query Editor			
Query History			
Data Output			
Query_ID [PK] character varying (100)	Emp_ID character varying (100)	Query text	
1 Q0000001	E0000160	Aliquam sit amet diam in magna bibendum imperdiet. Nullam orci pede, venenatis non, sodales sed, tincidunt eu, felis. Fusce posuere felis sed lacus.	
2 Q0000002	E0000161	Quisque ut erat. Curabitur gravida nisi at nibh. In hac habitasse platea dictumst. Aliquam augue quam, sollicitudin vitae, consectetuer eget, rutrum at,	
3 Q0000003	E0000162	Vivamus metus arcu, adipiscing molestie, hendrerit at, vulputate vitae, nisl. Aenean lectus. Pellentesque eget nunc. Donec quis orci eget orci vehicula	
4 Q0000004	E0000163	Integer a nibh. In quis justo. Maecenas rhoncus aliquam lacus.	
5 Q0000005	E0000164	In hac habitasse platea dictumst. Aliquam augue quam, sollicitudin vitae, consectetuer eget, rutrum at, lorem. Integer tincidunt ante vel ipsum. Praese	
6 Q0000006	E0000165	Nunc nisl. Duis bibendum, felis sed interdum venenatis, turpis enim blandit mi, in porttitor pede justo eu massa. Donec dapibus.	
7 Q0000007	E0000166	Vivamus vestibulum sagittis sapien. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam vel augue. Vestibu	
8 Q0000008	E0000167	Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Proin interdum mauris non ligula pellentesque ultrices. Phasellus id sapien in sapien iaculis	
9 Q0000009	E0000168	Nunc rhoncus dui vel sem. Sed sagittis. Nam congue, risus semper porta volutpat, quam pede lobortis ligula, sit amet eleifend pede libero quis orci. N	
10 Q0000010	E0000153	In eleifend quam a odio. In hac habitasse platea dictumst. Maecenas ut massa quis augue luctus tincidunt. Nulla mollis molestie lorem. Quisque ut er	
11 Q0000011	E0000153	Aenean lectus. Pellentesque eget nunc. Donec quis orci eget orci vehicula condimentum. Curabitur in libero ut massa volutpat convallis.	
12 Q0000012	E0000153	Morbi sem mauris, laoreet ut, rhoncus aliquet, pulvinar sed, nisl. Nunc rhoncus dui vel sem. Sed sagittis. Nam congue, risus semper porta volutpat, qu	
13 Q0000013	E0000160	Curabitur at ipsum ac tellus semper interdum. Mauris ullamcorper purus sit amet nulla. Quisque arcu libero, rutrum ac, lobortis vel, dapibus at, diam. N	
14 Q0000014	E0000161	Aliquam non mauris. Morbi non lectus. Aliquam sit amet diam in magna bibendum imperdiet. Nullam orci pede, venenatis non, sodales sed, tincidunt	
15 Q0000015	E0000162	Integer aliquet, massa id lobortis convallis, tortor risus dapibus augue, vel accumsan tellus nisi eu orci. Mauris lacinia sapien quis libero. Nullam sit ar	
16 Q0000016	E0000163	Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Proin interdum mauris non ligula pellentesque ultrices. Phasellus id sapien in sapien iaculis	
17 Q0000017	E0000164	Nulla suscipit ligula in lacus. Curabitur at ipsum ac tellus semper interdum. Mauris ullamcorper purus sit amet nulla.	
18 Q0000018	E0000165	Maecenas ut massa quis augue luctus tincidunt. Nulla mollis molestie lorem. Quisque ut erat.	
19 Q0000019	E0000166	Vivamus in felis eu sapien cursus vestibulum. Proin eu mi. Nulla ac enim. In tempor, turpis nec euismod scelerisque, quam turpis adipiscing lorem, vit	
20 Q0000020	E0000167	Proin at turpis a pede posuere nonummy. Integer non velit. Donec diam neque, vestibulum eget, vulputate ut, ultrices vel, augue. Vestibulum ante ipsu	
21 Q0000021	E0000168	Nulla justo. Aliquam quis turpis eget elit sodales scelerisque. Mauris sit amet eros.	
22 Q0000022	E000006	Maecenas ut massa quis augue luctus tincidunt. Nulla mollis molestie lorem. Quisque ut erat. Curabitur gravida nisi at nibh. In hac habitasse platea di	

## **Section : 6 : SQL Queries**

**1. List down all project ids which were started on or after 1-1-2013.**

**Query :**

```
select "Project_ID", "Start_date"  
from "Project"  
where "Start_date" >= '1-1-2013'
```

**Output :**

Query Editor		Query History
1	set search_path to "TMS";	
2		
3	select "Project_ID", "Start_date"	
4	from "Project"	
5	where "Start_date" >= '1-1-2013'	

Data Output			Explain	Messages	Notifications
	Project_ID	Start_date			
1	P0000009	2013-10-30			
2	P0000012	2013-09-11			
3	P0000013	2013-03-29			
4	P0000014	2014-11-06			
5	P0000016	2013-02-08			
6	P0000020	2014-09-21			
7	P0000024	2013-03-08			
8	P0000025	2013-06-04			
9	P0000026	2013-05-06			
10	P0000029	2014-10-19			
11	P0000032	2014-09-16			
12	P0000033	2014-06-28			
13	P0000042	2014-10-06			
14	P0000044	2014-05-29			

**Tuples : 41**

**2. Find out all the details of the projects which were assigned by the project manager having id equal to E0000011.**

**Query :**

```
select *
from "Project"
where "PM_ID" = 'E0000011'
```

**Output :**

Data Output Explain Messages Notifications											
	Project_ID [PK] character varying (100)	PM_ID character varying (100)	CL_ID character varying (100)	Name character varying (500)	Budget bigint	Start_date date	End_date date	Progress smallint	Status character varying (2)		
1	P000001	E0000011	C000001	Baxter Whitcher	2542004	2010-09-04	2015-09-10	87	Approved		
2	P000021	E0000011	C000021	Tad Gregoriou	6737621	2011-12-25	2019-11-30	49	Pending		
3	P000041	E0000011	C000041	Fowler Sisland	918822	2010-12-04	2021-05-22	24	Pending		
4	P000061	E0000011	C000061	Gustie Bunnell	2735728	2013-06-06	2015-03-09	78	Improve		
5	P000081	E0000011	C000081	Bucky Archibald	310101	2014-07-30	2019-08-03	45	Improve		
6	P000101	E0000011	C000005	Nadiya Delf	6590108	2014-06-17	2020-06-05	61	Pending		

**Tuples : 6**

**3. List down all the details of the projects having status as 'Pending'.**

**Query :**

```
select *  
from "Project"  
where "Status" = 'Pending'
```

**Output :**

Query Editor    Query History

```
1 set search_path to "TMS";  
2  
3 select *  
4 from "Project"  
5 where "Status" = 'Pending'
```

Data Output   Explain   Messages   Notifications

	Project_ID [PK] character varying (100)	PM_ID character varying (100)	CL_ID character varying (100)	Name character varying (500)	Budget bigint	Start_date date	End_date date	Progress smallint	Status character varying
1	P0000002	E0000022	C0000002	Agretha Taillard	5179852	2011-04-02	2019-08-04	71	Pending
2	P0000003	E0000033	C0000003	Kim Sommerlie	2255144	2010-01-06	2018-01-02	74	Pending
3	P0000006	E0000066	C0000006	Keary Postlewhite	6907652	2010-05-17	2017-10-29	5	Pending
4	P0000011	E0000192	C0000011	Arluene Will	5833538	2012-05-09	2017-11-17	93	Pending
5	P0000015	E0000196	C0000015	Dean Pandya	294556	2010-08-23	2020-07-27	25	Pending
6	P0000021	E0000011	C0000021	Tad Gregoriou	6737621	2011-12-25	2019-11-30	49	Pending
7	P0000022	E0000022	C0000022	Gannon O'Lenane	7574018	2011-04-16	2017-01-09	61	Pending
8	P0000023	E0000033	C0000023	Cristine Boyes	6246161	2012-02-28	2018-02-10	65	Pending
9	P0000025	E0000055	C0000025	Ira Boltwood	4188743	2013-06-04	2017-04-23	94	Pending
10	P0000027	E0000077	C0000027	Erie Stirton	4378986	2011-02-15	2015-01-02	53	Pending

**Tuples : 50**

**4. Count the number of queries that are pending.**

**Query :**

```
select count(*)  
from "Query"  
where "Status" = 'Pending'
```

**Output :**

The screenshot shows a PostgreSQL query editor interface. At the top, there are tabs for 'Query Editor' (which is selected) and 'Query History'. Below the tabs, the query code is displayed:

```
1 set search_path to "TMS";  
2  
3 select count(*)  
4 from "Query"  
5 where "Status" = 'Pending'
```

Below the code, there are four tabs: 'Data Output' (selected), 'Explain', 'Messages', and 'Notifications'. Under 'Data Output', a table is shown with one row:

	count	bigint
1	81	

**Tuples : 1**

**5. List down client id having number of contacts greater than 1.**

**Query :**

```
select "CL_ID", count("Contact")
from "Client Contact"
group by "CL_ID"
having count ("Contact") > 1
```

**Output :**

Query Editor    Query History

```
1 set search_path to "TMS";
2
3 select "CL_ID", count("Contact")
4 from "Client Contact"
5 group by "CL_ID"
6 having count ("Contact") > 1
```

Data Output   Explain   Messages   Notifications

	CL_ID	count
1	C0000043	2
2	C0000093	2
3	C0000013	2
4	C0000039	2
5	C0000027	2
6	C0000079	2
7	C0000077	2
8	C0000097	2
9	C0000087	2
10	C0000010	2
11	C0000023	2

**Tuples : 55**

**6. Find out all the details of the employees living in 'Texas' city.**

**Query :**

```
select *
from "Employee"
where "State" = 'Texas'
```

**Output :**

	Emp_ID [PK] character varying (100)	Dept_ID character varying (100)	Username character varying (200)	Password character varying (200)	Name character varying (100)	Designation character varying (200)	State character varying (200)	City character varying (200)
1	E0000022	D0000011	yshropsheirl	IOJ93D53	Yolanda Shropsheir	Intern	Texas	El Paso
2	E0000025	D0000003	chosteno	9BKZiTW	Charmain Hosten	Junior	Texas	Houston
3	E0000035	D0000002	mlangstrathy	YR1QKbD	Monty Langstrath	Senior	Texas	College Station
4	E0000036	D0000003	whannibalz	WPZOysR1wRs7	Worth Hannibal	Intern	Texas	San Antonio
5	E0000037	D0000004	cluckman10	Z4JKNXIY2Arc	Codi Luckman	Senior	Texas	Kaufman
6	E0000055	D0000011	rdumbelton1i	roSA3NXbDKH8	Raquel Dumbelton	Senior	Texas	Houston
7	E0000056	D0000001	wahanhardt1j	xeTLDhrk47	Willis Hanhardt	Junior	Texas	San Antonio
8	E0000064	D0000009	cledgerton1r	zUlygcNP37Z	Corny Ledgerton	Senior	Texas	Austin
9	E0000065	D0000010	kbelbin1s	HGWNh52Lohg	Keelby Belbin	Intern	Texas	Dallas
10	E0000085	D0000008	pnolin2c	r6eZCMm7ll1q	Padgett Nolin	Junior	Texas	Dallas
11	E0000089	D0000001	fdorrance2g	IFXPWd	Felice Dorrance	Junior	Texas	Houston
12	E0000098	D0000010	tklulik2p	4jVNQmmI4sBn	Tori Klulik	Junior	Texas	Houston
13	E0000102	D0000006	dtwinbow2t	nd9BX8HCu	Dniren Twinbow	Intern	Texas	Tyler
14	E0000110	D0000006	achanner31	M5RJmRIVG	Alexa Channer	Senior	Texas	San Antonio
15	E0000113	D0000006	mmacillrick34	C6oxMs	Marjie Macillrick	Intern	Texas	El Paso
16	E0000119	D0000006	cfantham3a	Zw3ZxNh4	Caroljean Fantham	Senior	Texas	Dallas
17	E0000132	D0000006	adavids3n	j5Tg97P	Abbi Davids	Intern	Texas	San Antonio
18	E0000168	D0000006	plandsborough4n	wX2dus9r1Zy	Phella Landsborough	Senior	Texas	El Paso
19	E0000174	D0000009	wsaggs4t	5RQ6LX	Willetta Saggs	Intern	Texas	Dallas
20	E0000180	D0000009	rhlpa47	knRKGNV	Rohinett Da Avenall	Senior	Texas	El Paso

**Tuples : 23**

**7. List down all Announcements which were announced by the project manager having id ‘E0000022’.**

**Query :**

```
select "An_ID", "Announcement"  
from "Announcement"  
where "PM_ID" = 'E0000022'
```

**Output :**

Query Editor		Query History
Data Output	Explain	Messages
1 set search_path to "TMS";		
2		
3 select "An_ID", "Announcement"		
4 from "Announcement"		
5 where "PM_ID" = 'E0000022'		
An_ID [PK] character varying (100)	Announcement text	
1 A000002	Fusce congue, diam id ornare imperdiet, sapien urna pretium nisl, ut volutpat sapien arcu sed augue. Aliquam erat volutpat. In congue. Etiam justo.	
2 A0000022	Morbi non quam nec dui luctus rutrum. Nulla tellus. In sagittis dui vel nisl. Duis ac nibh.	
3 A0000042	Nunc nisl. Duis bibendum, felis sed interdum venenatis, turpis enim blandit mi, in porttitor pede justo eu massa. Donec dapibus.	
4 A0000062	Morbi vestibulum, velit id pretium iaculis, diam erat fermentum justo, nec condimentum neque sapien placerat ante. Nulla justo. Aliquam quis turpis eget elit	
5 A0000082	Maecenas ut massa quis augue luctus tincidunt. Nulla mollis molestie lorem. Quisque ut erat. Curabitur gravida nisi at nibh.	
6 A0000102	In quis justo. Maecenas rhoncus aliquam lacus. Morbi quis tortor id nulla ultrices aliquet.	
7 A0000122	Duis mattis egestas metus. Aenean fermentum. Donec ut mauris eget massa tempor convallis. Nulla neque libero, convallis eget, eleifend luctus, ultricies eu, ni	
8 A0000142	Cras pellentesque volutpat dui. Maecenas tristique, est et tempus semper, est quam pharetra magna, ac consequat metus sapien ut nunc. Vestibulum ante ipsu	
9 A0000162	Pellentesque at nulla. Suspendisse potenti. Cras in purus eu magna vulputate luctus.	
10 A0000182	In sagittis dui vel nisl. Duis ac nibh. Fusce lacus purus, aliquet at, feugiat non, pretium quis, lectus.	

**Tuples : 10**

**8. List down all the details of the tasks having cost greater than 12000.**

**Query :**

```
select *  
from "Task"  
where "Cost" >= 12000
```

**Output :**

Query Editor    Query History

```
1 select *  
2 from "Task"  
3 where "Cost" >= 12000  
4
```

Data Output   Explain   Messages   Notifications

	Task_ID [PK] character varying (100)	Project_ID [PK] character varying (100)	Name character varying (500)	Cost bigint	Start_date date	End_date date	Progress smallint	Status character varying (200)
1	T0000001	P0000001	Merops nubicus	40863	2010-09-04	2015-09-10	99	Approved
2	T0000002	P0000001	Tragelaphus strepsiceros	47598	2011-04-02	2019-08-04	50	Approved
3	T0000003	P0000001	Haliaetus vocifer	49035	2010-01-06	2018-01-02	74	Pending
4	T0000001	P0000002	Geococcyx californianus	50368	2012-12-20	2021-07-25	15	Pending
5	T0000002	P0000002	Lama guanicoe	53962	2010-10-15	2018-09-28	45	Improve
6	T0000001	P0000003	Phascogale calura	60134	2010-05-17	2017-10-29	87	Pending
7	T0000001	P0000004	Capra ibex	46961	2012-06-10	2019-09-30	52	Approved
8	T0000002	P0000004	Cordylus giganteus	28058	2010-02-02	2016-07-27	19	Pending
9	T0000001	P0000005	Aquila chrysaetos	13769	2013-10-30	2016-11-11	66	Improve
10	T0000001	P0000006	Ceratotherium simum	64091	2010-02-18	2017-06-14	41	Approved
11	T0000001	P0000007	Perameles nasuta	26597	2012-05-09	2017-11-17	31	Pending
12	T0000001	P0000008	Cynictis penicillata	29742	2013-09-11	2015-04-20	27	Improve
13	T0000001	P0000009	Petaurus breviceps	82083	2013-03-29	2015-12-06	27	Pending
14	T0000001	P0000010	Diomedea irrorata	87574	2014-11-06	2015-10-22	76	Approved
15	T0000001	P0000011	Oups bengalensis	40622	2010-08-22	2020-07-27	0	Approved

**Tuples : 151**

**9. List down all the Resources which were used in the project having id 'P0000045'.**

**Query :**

```
select *  
from "Resource"  
where "Project_ID" = 'P0000045'
```

**Output :**

Data Output				
	Rs_ID [PK] character varying (100)	PM_ID character varying (100)	Project_ID character varying (100)	
			Description text	
1	R0000045	E0000055	P0000045	Sed accumsan felis. Ut at dolor quis odio consequat varius. Integer ac leo.
2	R0000165	E0000055	P0000045	In sagittis dui vel nisl. Duis ac nibh. Fusce lacus purus, aliquet at, feugiat non, pretium quis, lectus. Suspendisse

**Tuples : 2**

**10. List down all the log\_ID respective to the task and project for the employee having id 'E0000006'.**

**Query :**

```
select "Project_ID", "Task_ID", "Log_ID"
from "Updates"
where "Emp_ID" = 'E0000006'
```

**Output :**

Data Output			
	Project_ID character varying (100)	Task_ID character varying (100)	Log_ID bigint
1	P0000001	T0000001	98032122
2	P0000070	T0000002	67325035
3	P0000098	T0000001	72587619

**Tuples : 3**

**11. Find all details of the client and its corresponding project details whose Project id Is equal to 'P0000007'.**

**Query :**

```
select *  
from "Client", "Project"  
where "Project"."Project_ID" = 'P0000007' and "Client"."CL_ID" = "Project"."CL_ID"
```

**Output :**

Data Output								
	CL_ID	Username	Password	Name	City	State	Email	Company
1	C0000007	dchester6	h2whc	Danika Chester	Oklahoma City	Oklahoma	dchester6@upenn.edu	Flipopia

**Tuples : 1**

**12. Select the Project\_ID, Start\_date, End\_date and Budget of that project in which Resource whose ID is R0000048 is used.**

**Query :**

```
select "Project"."Project_ID", "Start_date" , "End_date", "Budget"  
from "Project" left outer join "Resource"  
on "Project"."Project_ID" = "Resource"."Project_ID"  
where "Resource"."Rs_ID"='R0000048'
```

**Output :**

Query Editor    Query History

```
1 select "Project"."Project_ID", "Start_date" , "End_date", "Budget"  
2 from "Project" left outer join "Resource"  
3 on "Project"."Project_ID" = "Resource"."Project_ID"  
4 where "Resource"."Rs_ID"='R0000048'
```

Data Output   Explain   Messages   Notifications

	Project_ID	Start_date	End_date	Budget
1	P0000048	2011-12-24	2021-09-06	8139377

**Tuples : 1**

**13. Select Emp\_ID, Name and Designation of the employee who has asked the query in the query session.**

**Query :**

```
select "Employee"."Emp_ID", "Employee"."Name", "Employee"."Designation"
from "Employee" right outer join "Query"
on "Employee"."Emp_ID" = "Query"."Emp_ID"
```

**Output :**

Query Editor    Query History

```
1 set search_path to "TMS";
2 select "Employee"."Emp_ID", "Employee"."Name", "Employee"."Designation"
3 from "Employee" right outer join "Query"
4 on "Employee"."Emp_ID" = "Query"."Emp_ID"
5
```

Data Output    Explain    Messages    Notifications

	Emp_ID [PK] character varying (100)	Name character varying (100)	Designation character varying (200)
1	E0000160	Janel Newe	Senior
2	E0000161	Lucias Sallan	Junior
3	E0000162	Catina Bruggeman	Senior
4	E0000163	Davidson Wittke	Senior
5	E0000164	Leslie Massimo	Junior
6	E0000165	Liva Divine	Junior
7	E0000166	Silvester Tracy	Senior
8	E0000167	Kassie Rao	Junior
9	E0000168	Phelia Landsborough	Senior
10	E0000153	Wells Duley	Junior
11	E0000153	Wells Duley	Junior
12	E0000153	Wells Duley	Junior
13	E0000160	Janel Newe	Senior
14	E0000161	Lucias Sallan	Junior

**Tuples : 170**

**14. Find the Employee ID's who is not included in any team.**

**Query :**

```
select "Employee"."Emp_ID"  
from "Employee"  
where "Emp_ID" not in (select "Team"."Member_ID" from "Team" )
```

**Output :**

Query Editor    Query History	
1	set search_path to "TMS";
2	select "Employee"."Emp_ID"
3	from "Employee"
4	where "Emp_ID" not in (select "Team"."Member_ID" from "Team" )
5	

Data Output    Explain    Messages    Notifications	
	Emp_ID [PK] character varying (100)
1	E0000001
2	E0000002
3	E0000003
4	E0000004
5	E0000005
6	E0000007
7	E0000008
8	E0000009
9	E0000010
10	E0000012
11	E0000013
12	E0000014
13	E0000015
14	E0000016

**Tuples : 103**

**15. Display Project name, Manager name, Budget, Start\_date, End\_date, Progress, and Status of the project which is given by client whose ID is C0000005.**

**Query :**

```
select "Project"."Name" as "Project_Name","Employee"."Name" as  
"Manager_Name","Project"."Budget","Project"."Start_date","Project"."End_date","Project"."Progress","Project".  
"Status"  
from "Project Manager","Project","Employee"  
where "Project Manager"."PM_ID" = "Project"."PM_ID" and "Project Manager"."PM_ID" =  
"Employee"."Emp_ID" and "Project"."CL_ID" = 'C0000005'
```

**Output :**

Query Editor    Query History

```
1 set search_path to "TMS";  
2 select "Project"."Name" as "Project_Name", "Employee"."Name" as "Manager_Name", "Project"."Budget", "Project"."Start_date", "Project"."End_date", "Pro  
3 from "Project Manager", "Project", "Employee"  
4 where "Project Manager"."PM_ID" = "Project"."PM_ID" and "Project Manager"."PM_ID" = "Employee"."Emp_ID" and "Project"."CL_ID" = 'C0000005'  
5  
6  
7
```

Data Output   Explain   Messages   Notifications

	Project_Name	Manager_Name	Budget	Start_date	End_date	Progress	Status
1	Nadiya Delf	Hilary Eastam	6590108	2012-03-25	2017-10-04	61	Pending
2	Alissa Coal	Raquel Dumbelton	9155680	2011-02-13	2019-03-08	96	Improve
3	Carmelita Swalteridge	Harrison Gwalter	6025425	2013-06-08	2019-05-16	95	Approved
4	Werner Serjeantson	Cosimo Styen	5856630	2014-04-29	2021-07-16	61	Pending

**Tuples : 4**

**16. Display all the teammates of employee E0000165 of all time.**

**Query :**

```
select distinct("Team"."Member_ID")
from "Team"
where "Team"."Member_ID" <> 'E0000165' and ("Team"."Project_ID","Team"."Task_ID") in
(select "Team"."Project_ID","Team"."Task_ID"
from "Team"
where "Team"."Member_ID" = 'E0000165')
```

**Output :**

The screenshot shows a PostgreSQL query editor interface. The title bar says '201901305\_db/postgres@PostgreSQL 13'. The 'Query Editor' tab is selected. The query text is:

```
1 set search_path to "TMS";
2 select distinct("Team"."Member_ID")
3 from "Team"
4 where "Team"."Member_ID" <> 'E0000165' and ("Team"."Project_ID","Team"."Task_ID") in
5 (select "Team"."Project_ID","Team"."Task_ID"
6 from "Team"
7 where "Team"."Member_ID" = 'E0000165')
8
```

The 'Data Output' tab is selected, showing the results of the query:

Member_ID
character varying (100)
1 E0000166
2 E0000163
3 E0000167
4 E0000164
5 E0000168

**Tuples : 5**

**17. Find Member IDs who worked in Project P0000001 or P0000002.**

**Query :**

```
(select "Team"."Member_ID" from "Team" where "Team"."Project_ID" = 'P0000001')
union
(select "Team"."Member_ID" from "Team" where "Team"."Project_ID" = 'P0000002')
```

**Output :**

Query Editor		Query History
1	set search_path to "TMS";	
2	(select "Team"."Member_ID" from "Team" where "Team"."Project_ID" = 'P0000001')	
3	union	
4	(select "Team"."Member_ID" from "Team" where "Team"."Project_ID" = 'P0000002')	
5		
Data Output Explain Messages Notifications		
	Member_ID character varying (100)	
1	E0000066	
2	E0000039	
3	E0000028	
4	E0000006	
5	E0000061	
6	E0000011	
7	E0000072	
8	E0000017	
9	E0000022	
10	E0000033	
11	E0000050	
12	E0000055	
13	E0000044	

**Tuples : 13**

**18. List down Project ID having a number of tasks greater than 2.**

**Query :**

```
select "Project_ID", count("Task_ID")
from "Task"
group by "Project_ID"
having count("Task_ID") > 2
```

**Output :**

Query Editor    Query History

```
1  (select "Project_ID", count("Task_ID")
2   from "Task"
3   group by "Project_ID"
4   having count("Task_ID") > 2)
5
6 |
```

Data Output   Explain   Messages   Notifications

	Project_ID	count
1	P0000070	3
2	P0000042	3
3	P0000001	3

**Tuples : 3**

**19. List down Emp\_ID, Name, corresponding Project\_Id, its progress having Project's Progress greater than 90.**

**Query :**

```
select "Employee"."Emp_ID","Employee"."Name","Project"."Project_ID","Project"."Progress"
from "Project","Team","Employee"
where ("Project"."Project_ID","Team"."Member_ID")=("Team"."Project_ID","Employee"."Emp_ID") and
"Progress">90
```

**Output :**

Query Editor    Query History

```
1 select "Employee"."Emp_ID","Employee"."Name","Project"."Project_ID","Project"."Progress"
2 from "Project","Team","Employee"
3 where ("Project"."Project_ID","Team"."Member_ID")=("Team"."Project_ID","Employee"."Emp_ID") and "Progress">>90
4
```

Data Output   Explain   Messages   Notifications

	Emp_ID character varying (100)	Name character varying (100)	Project_ID character varying (100)	Progress smallint
1	E0000006	Reine Neat	P0000024	95
2	E0000011	Hillary Eastam	P0000024	95
3	E0000017	Carissa Bissatt	P0000024	95
4	E0000022	Yolanda Shropsheir	P0000025	94
5	E0000028	Antonino Dundredge	P0000025	94
6	E0000033	Patric Dovinson	P0000025	94
7	E0000039	Rriocard Shemming	P0000025	94
8	E0000044	Sidney Emmitt	P0000025	94
9	E0000050	Charlotte Sudell	P0000025	94
10	E0000066	Gae Bolduc	P0000047	98
11	E0000072	Dicky Afield	P0000047	98
12	E0000077	Wyndham Kail	P0000047	98
13	E0000083	Adriianne Paule	P0000047	98
14	E0000107	Marlo Alentyev	P0000005	96
15	E0000108	Fifi Cowden	P0000005	96

**Tuples : 56**

**20. List down all Projects with its Manager, Name and Budget whose Client is living in New York.**

**Query :**

```
select "Project_ID","PM_ID"as Manager_ID,"Name","Budget"  
from "Project"  
where "CL_ID" in (select "CL_ID" from "Client" where "State" = 'New York' )
```

**Output :**

Query Editor    Query History

```
1 select "Project_ID","PM_ID"as Manager_ID,"Name","Budget"  
2 from "Project"  
3 where "CL_ID" in (select "CL_ID" from "Client" where "State" = 'New York' )  
4
```

Data Output   Explain   Messages   Notifications

	Project_ID [PK] character varying (100)	manager_id character varying (100)	Name character varying (500)	Budget bigint
1	P000001	E0000011	Baxter Whitcher	2542004
2	P0000025	E0000055	Ira Boltwood	4188743
3	P0000040	E0000201	Carie O'Brallaghan	439274
4	P0000057	E0000198	Lorelei Beckey	4205216
5	P0000094	E0000195	Nikolaus Anwell	3500902
6	P0000105	E0000055	Merrily Hubert	8199799
7	P0000108	E0000088	Lorri Lymer	8517322
8	P0000113	E0000194	Christian Reid	8553186

**Tuples : 8**

**21. List down all Employees who worked in both Project 'P0000045' and 'P0000001'.**

**Query :**

```
(select "Member_ID" from "Team" where "Project_ID" = 'P0000001')
intersect
(select "Member_ID" from "Team" where "Project_ID" = 'P0000045')
```

**Output :**

Query Editor	Query History		
1 (select "Member_ID" from "Team" where "Project_ID" = 'P0000001') 2 intersect 3 (select "Member_ID" from "Team" where "Project_ID" = 'P0000045')			
Data Output	Explain	Messages	Notifications
Member_ID	character varying (100)		
1	E0000011		
2	E0000017		
3	E0000022		
4	E0000033		
5	E0000028		

**Tuples : 5**

## 22. Find Client\_ID, Name, Budget who has maximum Project's Budget.

Query :

```
with max_budget (value) as
(select max("Budget")
from "Project")
select "Client"."CL_ID","Client"."Name","Project"."Budget"
from "Client","Project","max_budget"
where "Project"."CL_ID" = "Client"."CL_ID" and "Project"."Budget" = "max_budget".value and
"Client"."CL_ID" =
(select "CL_ID"
from "Project","max_budget"
where "Project"."Budget"="max_budget".value)
```

Output :

Query Editor		Query History	
Line	Text	Line	Text
1	with max_budget (value) as		
2	(select max("Budget")		
3	from "Project")		
4	select "Client"."CL_ID","Client"."Name","Project"."Budget"		
5	from "Client","Project","max_budget"		
6	where "Project"."CL_ID" = "Client"."CL_ID" and "Project"."Budget" = "max_budget".value and "Client"."CL_ID" =		
7	(select "CL_ID"		
8	from "Project","max_budget"		
9	where "Project"."Budget"="max_budget".value)		
10			

Data Output			
	CL_ID	Name	Budget
1	C0000035	Corrine Scarbarrow	9954259

Tuples : 1

**23. List down all departments with its number of employees.**

**Query :**

```
select "Department"."Dept_Name",count("Employee"."Emp_ID")
from "Department" natural join "Employee"
group by "Department"."Dept_Name"
```

**Output :**

Query Editor    Query History

```
1 select "Department"."Dept_Name",count("Employee"."Emp_ID")
2 from "Department" natural join "Employee"
3 group by "Department"."Dept_Name"
4 |
```

Data Output    Explain    Messages    Notifications

	Dept_Name	count
1	Marketing	9
2	Finance	9
3	Development	78
4	Management	20
5	Testing	31
6	Admin	9
7	Quality Control	9
8	Sales	9
9	HR	9
10	Customer Service	9
11	Technical	9

**Tuples : 11**

**24. List down Project\_ID, Project wise Total cost, Budget of every project.**

**Query :**

```
select "Project"."Project_ID",sum("Cost") as "Project_Cost","Project"."Budget"
from "Task","Project"
where "Project"."Project_ID"="Task"."Project_ID"
group by "Project"."Project_ID"
order by "Project"."Project_ID"
```

**Output :**

Query Editor    Query History

```
1 select "Project"."Project_ID",sum("Cost") as "Project_Cost","Project"."Budget"
2 from "Task","Project"
3 where "Project"."Project_ID"="Task"."Project_ID"
4 group by "Project"."Project_ID"
5 order by "Project"."Project_ID"
6
```

Data Output   Explain   Messages   Notifications

	Project_ID [PK] character varying (100)	Project_Cost numeric	Budget bigint
1	P0000001	137496	2542004
2	P0000002	104330	5179852
3	P0000003	60134	2255144
4	P0000004	75019	3808743
5	P0000005	13769	9155680
6	P0000006	64091	6907652
7	P0000007	26597	1137501
8	P0000008	29742	6795926
9	P0000009	82083	349967
10	P0000010	87574	4889230
11	P0000011	49623	5833538
12	P0000012	64169	690905
13	P0000013	57527	4235407
14	P0000014	71041	5730500

**Tuples : 120**

**25. List down all employees who haven't updated yet.**

**Query :**

```
(select "Emp_ID" from "Employee")
except
(select "Emp_ID" from "Updates")
```

**Output :**

Query Editor    Query History	
1	(select "Emp_ID" from "Employee")
2	except
3	(select "Emp_ID" from "Updates")
4	
Data Output	Explain    Messages    Notifications
	Emp_ID character varying (100) 
1	E0000187
2	E0000074
3	E0000173
4	E0000068
5	E0000089
6	E0000188
7	E0000092
8	E0000056
9	E0000073
10	E0000190
11	E0000095
12	E0000069
13	E0000090
14	E0000019
15	E0000075

**Tuples : 103**

**26. Create a function returning Employee\_IDs of X department.**

**Query :**

```
CREATE OR REPLACE function "Find_Employees"(x character varying(100))
RETURNS TABLE (a character varying(100))
LANGUAGE 'plpgsql'
AS

$BODY$

DECLARE
R_LIST record;
BEGIN
CREATE TEMP TABLE "RES"("Emp_ID" CHARACTER VARYING (100) ON COMMIT
DROP;
FOR R_LIST IN (SELECT "Emp_ID"
               FROM "Employee"
              WHERE "Dept_ID" IN
                    (SELECT "Dept_ID" FROM "Department" WHERE "Dept_Name"
= x))
LOOP
INSERT INTO "RES" ("Emp_ID") VALUES (R_LIST."Emp_ID");
END LOOP;

RETURN QUERY TABLE "RES";
END;

$BODY$;

select "Find_Employees"('Development');
```

**Output :**

Query Editor    Query History

```
1 set search_path to "TMS";
2
3 CREATE OR REPLACE function "Find_Employees"(x character varying(100))
4 RETURNS TABLE (a character varying(100))
5 LANGUAGE 'plpgsql'
6 AS
7
8 $BODY$
9
10 DECLARE
```

Data Output   Explain   Messages   Notifications

	Find_Employees
1	E000006
2	E000017
3	E000028
4	E000039
5	E000050
6	E000061
7	E000072
8	E000083
9	E000094
10	E000100

Tuples : 78

**27. List down Emp\_ID with Contact whose names are as follows : 'Venus Hiland','Joy Tabart','Pearla Flather','Hilary Eastam'.**

**Query :**

```
select "Emp_ID","Contact"
from "Employee Contact"
where "Emp_ID" in (select "Emp_ID" from "Employee" where "Name" in ('Venus Hiland','Joy Tabart','Pearla
Flather','Hilary Eastam'))
order by "Emp_ID"
```

**Output :**

Query Editor    Query History

```
1 select "Emp_ID","Contact"
2 from "Employee Contact"
3 where "Emp_ID" in (select "Emp_ID" from "Employee" where "Name" in ('Venus Hiland','Joy Tabart','Pearla
Flather','Hilary Eastam'))
4 order by "Emp_ID"
5
```

Data Output    Explain    Messages    Notifications

	Emp_ID [PK] character varying (100)	Contact [PK] character varying (100)
1	E0000005	422-846-7017
2	E0000005	794-281-2383
3	E0000008	883-952-4160
4	E0000008	611-335-9494
5	E0000009	730-281-3580
6	E0000009	593-579-7414
7	E0000011	781-674-7258
8	E0000011	242-750-1489

**Tuples : 8**

**28. Find the Emp\_ID with Own\_Progress greater than that of some employees in the Admin Department and they are not Senior.**

**Query :**

```
select "Emp_ID"
from "Employee"
where "Own_progress" > some(select "Own_progress" from "Employee" where 'Dept_ID' in
                           (select 'Dept_ID' from "Department" where "Dept_Name" = 'Admin')) and "Designation" not in
                           ('Senior')
```

**Output :**

Query Editor		Query History
1	select "Emp_ID"	
2	from "Employee"	
3	where "Own_progress" > some(select "Own_progress" from "Employee" where 'Dept_ID' in	
4	(select 'Dept_ID' from "Department" where "Dept_Name" = 'Admin'))	
5	and "Designation" not in ('Senior')	
Data Output		
	Emp_ID	[PK] character varying (100)
1	E0000001	
2	E0000003	
3	E0000006	
4	E0000007	
5	E0000008	
6	E0000009	
7	E0000010	
8	E0000011	
9	E0000012	
10	E0000014	
11	E0000017	
12	E0000018	

**Tuples : 126**

**29. Find the City of Employees which start with 'W'.**

**Query :**

```
select distinct("City")
from "Employee"
where "City" like 'W%'
```

**Output :**

Query Editor		Query History
1	2	3
		<pre>select distinct("City") from "Employee" where "City" like 'W%'</pre>
Data Output		Explain Messages Notifications
1	City character varying (200)	
1	Washington	
2	Wichita	
3	Winston Salem	
4	Worcester	
5	Wilmington	

**Tuples : 5**

**30. Find the “Name” and concatenation of “Email”, “/”, “Contact” as “Email / Contact” from the “Client” table.**

**Query :**

```
select "Name",concat("Email",'/','Contact') as "Email/Contact"
from "Client","Client Contact"
where "Client"."CL_ID" = "Client Contact"."CL_ID"
```

**Output :**

Query Editor    Query History

```
1 select "Name",concat("Email",'/','Contact') as "Email/Contact"
2 from "Client","Client Contact"
3 where "Client"."CL_ID" = "Client Contact"."CL_ID"
```

Data Output    Explain    Messages    Notifications

	Name character varying (500)	Email/Contact text
1	Antonio Makeswell	amakeswell0@bloglovin.com/763-514-0374
2	Izak Liddington	iliddington1@ameblo.jp/513-172-3862
3	Charley Yggo	cyggo2@google.cn/697-870-1271
4	Ellary Keneford	ekeneford3@liveinternet.ru/245-376-8896
5	Rosaleen Fowden	rpowden4@japanpost.jp/193-472-5061
6	Leona Witson	lwitson5@ovh.net/200-952-1347
7	Danika Chester	dchester6@upenn.edu/430-769-9081
8	Stesha Mattimoe	smattimoe7@unicef.org/413-653-5002
9	Fayina Dominici	fdominici8@umn.edu/455-502-7294
10	Amandi Dickman	adickman9@netvibes.com/618-812-8056
11	Kennie Davidowsky	k davidsky@kickstarter.com/493-480-6409
12	Lisha Lattos	lattosb@cocolog-nifty.com/965-953-5854
13	Hart Crigin	hcriginc@networksolutions.com/961-845-0961
14	Gianna Sharply	gsharplyd@t.co/434-868-7468
15	Dre Sandland	dsandlande@taobao.com/388-681-5308

**Tuples : 160**

**31. Find all the projects which are managed by Senior Managers and also display their managers.**

**Query :**

```
select "Project_ID", "PM_ID"
from "Project"
where "PM_ID" in (select "Emp_ID"
                    from "Employee"
                    where "Employee"."Designation" = 'Senior'
                        and "Employee"."Dept_ID" in (select "Dept_ID"
                            from "Department"
                            where "Dept_Name" = 'Management'
                        )
                    )
                )
```

**Output :**

Query Editor		Query History	
Line	Text	Line	Text
1	select "Project_ID", "PM_ID"		
2	from "Project"		
3	where "PM_ID" in (select "Emp_ID"		
4	from "Employee"		
5	where "Employee"."Designation" = 'Senior'		
6	and "Employee"."Dept_ID" in (select "Dept_ID"		
7	from "Department"		
8	where "Dept_Name" = 'Management'		
9	)		
10	)		

Data Output		Explain	Messages	Notifications
Project_ID [PK] character varying (100)	PM_ID character varying (100)			
1 P0000003	E0000033			
2 P0000004	E0000044			
3 P0000005	E0000055			
4 P0000009	E0000099			
5 P0000011	E0000192			
6 P0000012	E0000193			
7 P0000015	E0000196			
8 P0000020	E0000201			
9 P0000023	E0000033			

**Tuples : 48**

**32. Display the Clients who have given more than 2 projects in the company and display the number of projects given by them.**

**Query :**

```
select "CL_ID", count("Project_ID") as "No_of_Projects"  
from "Project"  
group by "CL_ID"  
having count("Project_ID") >= 2  
order by "CL_ID"
```

**Output :**

Query Editor    Query History

```
1 select "CL_ID", count("Project_ID") as "No_of_Projects"  
2 from "Project"  
3 group by "CL_ID"  
4 having count("Project_ID") >= 2  
5 order by "CL_ID"
```

Data Output   Explain   Messages   Notifications

	CL_ID	No_of_Projects
1	C000005	4
2	C000006	2
3	C000007	2
4	C000008	2
5	C000010	2
6	C000015	3
7	C000020	2
8	C000025	3
9	C000030	2
10	C000035	3
11	C000040	2
12	C000045	3
13	C000050	2
14	C000055	2

**Tuples : 14**

**33. Find the Employee whose own progress is highest in the Development department.**

**Query :**

```
select "Emp_ID"
from "Employee"
where "Own_progress" >= all (select "Own_progress" from "Employee" where "Dept_ID" = 'D0000006')
```

**Output :**

The screenshot shows a database query editor interface. At the top, there are tabs for 'Query Editor' (which is selected) and 'Query History'. Below the tabs, the SQL query is displayed:

```
1 select "Emp_ID"
2 from "Employee"
3 where "Own_progress" >= all (select "Own_progress" from "Employee" where "Dept_ID" = 'D0000006')
```

Below the query, there are four tabs: 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is selected, showing a table with one row of data:

	Emp_ID
1	E0000140

**Tuples : 1**

**34. Find the Client details whose project is divided into 3 or more Tasks.**

**Query :**

```
select *
from "Client"
where "CL_ID" in (select "Project"."CL_ID"
from "Project"
where "Project"."Project_ID" in (select "Task"."Project_ID"
from "Task"
group by "Task"."Project_ID"
having count("Task"."Task_ID")>=3))
```

**Output :**

Query Editor    Query History							
1	select *						
2	from "Client"						
3	where "CL_ID" in (select "Project"."CL_ID"						
4	from "Project"						
5	where "Project"."Project_ID" in (select "Task"."Project_ID"						
6	from "Task"						
7	group by "Task"."Project_ID"						
8	having count("Task"."Task_ID")>=3))						
9							
10							

Data Output    Explain    Messages    Notifications							
	CL_ID [PK] character varying (100)	Username character varying (200)	Password character varying (200)	Name character varying (500)	City character varying (200)	State character varying (200)	Email character varying (200)
1	C0000001	amakeswell0	ZWWS3uwB	Antonio Makeswell	Brooklyn	New York	amakeswell0@bloglo
2	C0000042	hstanton15	cngo1MVp	Hodge Stanton - Skinn	Tacoma	Washington	hstanton15@webnod
3	C0000070	srichold1x	6UGbhU4	Sayre Richold	San Jose	California	srichold1x@fastcomp

**Tuples : 3**

**35. Create View of project status, progres and display client view, the progress and status of particular project, say the one having id P0000112.**

**Query :**

```
create view show_progress as
    select "Project_ID", "CL_ID", "Name", "Progress", "Status"
    from "Project";
select "Progress", "Status"
from show_progress
where "Project_ID" = 'P0000112'
```

**Output :**

The screenshot shows a database query editor interface. At the top, there are tabs for 'Query Editor' (which is selected) and 'Query History'. Below the tabs, the SQL code for creating a view and selecting data from it is displayed. The 'from "Project"' clause in the first select statement is highlighted with a blue background. At the bottom of the interface, there is a preview area titled 'Data Output' which shows a single row of data from the view.

	Progress	Status
1	32	Approved

**Tuples : 1**

### 36. List down all Announcements which were announced after Year 2019.

Query :

```
select *
from "Announcement"
where to_char("Date",'yyyy') > '2019'
```

Output :

Data Output		
An_ID	PM_ID	Announcement
		text
1	E0000044	Mauris ullamcorper purus sit amet nulla. Quisque arcu libero, rutrum ac, lobortis vel, dapibus at, diam. Nam tristique tortor eu pede.
2	E0000196	Quisque porta volutpat erat. Quisque erat eros, viverra eget, congue eget, semper rutrum, nulla. Nunc purus. Phasellus in felis.
3	E0000199	In tempor, turpis nec euismod scelerisque, quam turpis adipiscing lorem, vitae mattis nibh ligula nec sem. Duis aliquam convallis nunc. Proin at turpis a pede.
4	E0000197	Suspendisse potenti. Nullam porttitor lacus at turpis. Donec posuere metus vitae ipsum. Aliquam non mauris.
5	E0000199	Nulla ut erat id mauris vulputate elementum. Nullam varius. Nulla facilisi.
6	E0000041	Aliquam non mauris. Morbi non lectus. Aliquam sit amet diam in magna bibendum imperdiet. Nullam orci pede, venenatis non, sodales sed, tincidunt eu, felis.
7	E0000044	Etiam justo. Etiam pretium iaculis justo. In hac habitasse platea dictumst.
8	E0000055	Fusce consequat. Nulla nisl. Nunc nisl. Duis bibendum, felis sed interdum venenatis, turpis enim blandit mi, in porttitor pede justo eu massa.
9	E0000066	Vivamus vestibulum sagittis sapien. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam vel augue.
10	E0000077	In quis justo. Maecenas rhoncus aliquam lacus. Morbi quis tortor id nulla ultrices aliquet. Maecenas leo odio, condimentum id, luctus nec, molestie sed, justus.
11	E0000088	Donec ut dolor. Morbi vel lectus in quam fringilla rhoncus. Mauris enim leo, rhoncus sed, vestibulum sit amet, cursus id, turpis.
12	E0000201	Nunc nisl. Duis bibendum, felis sed interdum venenatis, turpis enim blandit mi, in porttitor pede justo eu massa. Donec dapibus. Duis at velit eu est congue.
13	E0000033	Proin eu mi. Nulla ac enim. In tempor, turpis nec euismod scelerisque, quam turpis adipiscing lorem, vitae mattis nibh ligula nec sem. Duis aliquam convallis.
14	E0000055	Nulla tellus. In sagittis id vel nisl. Duis ac nibh.
15	E0000066	Nunc nisl. Duis bibendum, felis sed interdum venenatis, turpis enim blandit mi, in porttitor pede justo eu massa. Donec dapibus.
16	E0000088	Mauris sit amet eros. Suspendisse accumsan tortor quis turpis. Sed ante. Vivamus tortor.
17	E0000195	In tempor, turpis nec euismod scelerisque, quam turpis adipiscing lorem, vitae mattis nibh ligula nec sem. Duis aliquam convallis nunc. Proin at turpis a pede.
18	E0000044	Morbi ut odio. Cras mi pede, malesuada in, imperdiet et, commodo vulputate, justo. In blandit ultrices enim.
19	E0000066	Vivamus vestibulum sagittis sapien. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam vel augue. Vestibulum non.
20	E0000192	Vivamus tortor. Duis mattis egestas metus. Aenean fermentum. Donec ut mauris eget massa tempor convallis.
21	E0000195	Mauris lacinia sapien quis libero. Nullam sit amet turpis elementum ligula vehicula consequat. Morbi a ipsum.

Tuples : 47

37. List down all Project Managers with their finished number of projects after 2020 in descending order.

Query :

```
create or replace view latest_by_2020 as
    select "Project_ID", "PM_ID"
    from "Project"
    where to_char("End_date",'yyyy') >= '2020';

select "PM_ID", count("Project_ID")
from latest_by_2020
group by "PM_ID"
order by count("Project_ID") desc
```

Output :

Query Editor		Query History	
Line	Text	Line	Text
1	<code>set search_path to "TMS";</code>		
2			
3	<code>create or replace view latest_by_2020 as</code>		
4	<code>    select "Project_ID", "PM_ID"</code>		
5	<code>    from "Project"</code>		
6	<code>    where to_char("End_date",'yyyy') &gt;= '2020';</code>		
7			
8	<code>select "PM_ID", count("Project_ID")</code>		
9	<code>from latest_by_2020</code>		
10	<code>group by "PM_ID"</code>		
11	<code>order by count("Project_ID") desc</code>		

Data Output	Explain	Messages	Notifications				
<table border="1"><thead><tr><th>PM_ID</th><th>count</th></tr></thead><tbody><tr><td>character varying (100)</td><td>bigint</td></tr></tbody></table>	PM_ID	count	character varying (100)	bigint			
PM_ID	count						
character varying (100)	bigint						
1 E0000044	3						
2 E0000055	3						
3 E0000066	3						
4 E0000199	2						
5 E0000011	2						
6 E0000088	2						
7 E0000195	2						
8 E0000197	1						

Tuples : 15

**38. Display the Task led by every leader.**

**Query :**

```
select "TL_ID", count("Project_ID") as "Task_leaded"  
from "Team_Leader"  
group by "TL_ID"
```

**Output :**

Query Editor    Query History

```
1 set search_path to "TMS";  
2  
3 select "TL_ID", count("Project_ID") as "Task_leaded"  
4 from "Team_Leader"  
5 group by "TL_ID"
```

Data Output   Explain   Messages   Notifications

TLID	Task_leaded
E000017	4
E0000112	5
E0000140	8
E0000157	7
E0000055	4
E0000044	9
E0000197	7
E0000120	10
E0000161	9
E0000143	5
...	...

**Tuples : 77**

**39. List down all pairs of Client and Project Manager who worked together maximum times.**

**Query :**

```
create or replace view "PM-CL Pair" as
    select "PM_ID", "CL_ID", count("Project_ID")
    from "Project"
    group by ("PM_ID", "CL_ID")
    order by count desc;

select *
from "PM-CL Pair"
where count in (select max(count)
                 from "PM-CL Pair")
```

**Output :**

Query Editor    Query History

```
1 set search_path to "TMS";
2
3 create or replace view "PM-CL Pair" as
4     select "PM_ID", "CL_ID", count("Project_ID")
5     from "Project"
6     group by ("PM_ID", "CL_ID")
7     order by count desc;
8 |
9 select *
10 from "PM-CL Pair"
11 where count in (select max(count)
12                  from "PM-CL Pair")
13
```

Data Output    Explain    Messages    Notifications

	PM_ID	CL_ID	count
1	E0000055	C0000025	2
2	E0000191	C0000050	2

**Tuples : 2**

**40. Make a trigger function which checks unique usernames of employees while inserting the data and report on violations accordingly.**

**Query :**

```
CREATE OR REPLACE FUNCTION "TMS"."Trig1"()
RETURNS trigger
LANGUAGE 'plpgsql'
VOLATILE
COST 100
AS $BODY$
DECLARE
tmp varchar(10) = NULL;

BEGIN
select "Username" into tmp from "TMS"."Employee" where "Username" = new."Username";
if (tmp = new."Username") then
RAISE NOTICE 'Already exists..insert another username';
return NULL;
else
return NEW;
end if;
END
$BODY$;
```

**Output :**

The screenshot shows a PostgreSQL Query Editor window. The 'Query History' tab is selected. The query entered is:

```
1 INSERT INTO "TMS"."Employee"
2   "Emp_ID", "Dept_ID", "Username", "Password", "Name", "Designation", "State", "City", "Email", "Own_progress")
3   VALUES ('E00001235','D0000001', 'pziemee', '123456', 'Jyoti', 'Intern', 'GJ', 'Surat', 'dlkfj@gmail.com', '56');
```

Below the query, the 'Messages' tab is selected, showing the output:

```
NOTICE: Already exists..insert another username
INSERT 0 0
```

Data Output Explain Messages Notifications

NOTICE: Already exists..insert another username  
INSERT 0 0

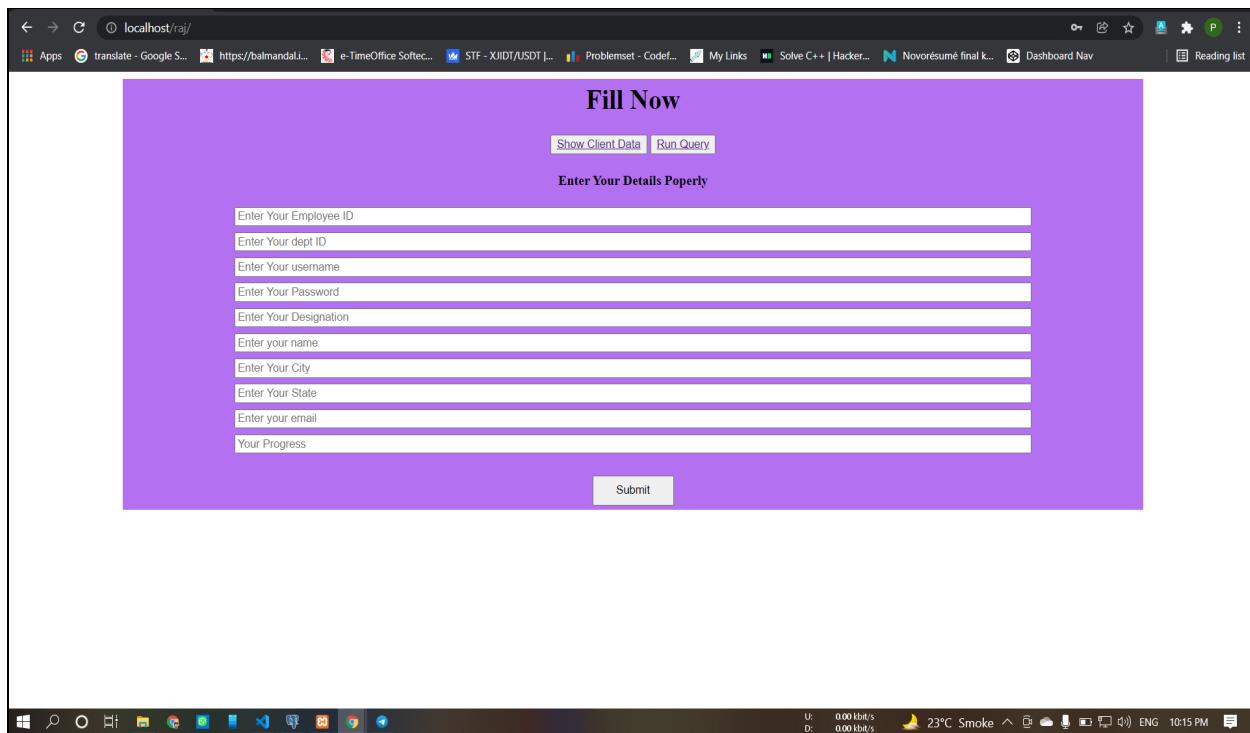
Query returned successfully in 995 msec.

**Tuples : 0**

## **Section : 7 : Front End**

We have created a web based application using a PHP framework named Task Management system. We have added three functionalities named as inserting data, Filter data , and Run any query.

## Home Page :



- Our database is connected to PostgreSQL via inbuilt functionalities provided by the external application XAMPP Control Panel. Here is the - PHP code by which we have connected our database PostgreSQL with this website.

```
<?php  
$host      = "host =localhost";  
$port      = "port = 5433";  
$dbname    = "dbname = example";  
$credentials = "user = postgres password=1234";  
  
// $db = pg_connect( "$host $port $dbname $credentials" );  
$con=pg_connect("host=localhost port=5433 dbname=example user=postgres password=1234");  
if(!$con) {  
echo "Error : Unable to open database\n";
```

```

} else {
    // echo "Opened database successfully\n";
}
?>

```

## Functionalities

### ❖ For Inserting data into table :

→ Here we have inserted our data in the Employee table which contains the details of all employees.

→ Before Inserting data in Employee Table :

The screenshot shows the pgAdmin interface. In the top-left, there's a terminal window with the title 'example/postgres@PostgreSQL 10 (x86)' containing the following SQL code:

```

1 SELECT * FROM "TMS"."Employee"
2 ORDER BY "Emp_ID" ASC

```

In the bottom-right, there's a large data table titled 'Data Output' showing employee records. The columns are: Emp\_ID, Dept\_ID, Username, Password, Designation, Name, City, and State. The data includes 205 rows of employee information.

Emp_ID	Dept_ID	Username	Password	Designation	Name	City	State
195	E0000194	D0000011	aeveque5d	Ovaalj0MGXu	Annelise Eveque	Junior	Michigan
196	E0000195	D0000011	ochominski5e	FmmUyuwrDIOf	Onida Chominski	Junior	West Virginia
197	E0000196	D0000011	lferrick5f	TB6xTA12zC	Leonelle Ferrick	Senior	Tennessee
198	E0000197	D0000011	mdurrand5g	2lgP1EPQOkk	Mariam Durrand	Intern	California
199	E0000198	D0000011	cstyan5h	c7YmlwQGZgo	Cosimo Styan	Intern	Texas
200	E0000199	D0000011	rosan5i	cGT9QSouuW	Ravid Osan	Intern	Michigan
201	E0000200	D0000011	akaliszewski5j	dZ5Zjcsv	Abbott Kaliszewski	Junior	California
202	E0000201	D0000011	fapted5k	Bn4AxYijC	Friederike Apted	Senior	Maryland
203	E111	D0000001	jkbb	dvkv2021	bk	bbk	bl
204	E1345678	D0000001	DVKVh	dvkv2021	d	g	i;
205	E7777777	D0000005	jemish	ljernrinruw	Senior	kdjaby	jekn

→ Employee table contains all details of the employees. We can insert the details of the employee by filling out the form and it will be reflected to our postgres live server. We can see details of that employee in the image that we have inserted through the form.

→ Entering data in Employee Table through web page :

The screenshot shows a web browser window with the URL `localhost/raj/index.php`. The page title is "Fill Now". It contains a form with the following fields:

- Emp\_ID: E9999999
- Dept\_ID: D0000001
- Username: zzzzzz
- Password: \*\*\*\*\*
- Designation: Intern
- Name: vin
- City: vadodra
- State: Gujarat
- Email: kartikpatelwami2@gmail.com
- Mobile: 91

A green success message "submit success" is displayed at the bottom. Below the form is a "Submit" button.

→ After Inserting the data in Employee table :

The screenshot shows a PostgreSQL query editor with the following SQL query:

```

1 SELECT * FROM "TMS"."Employee"
2 ORDER BY "Emp_ID" ASC

```

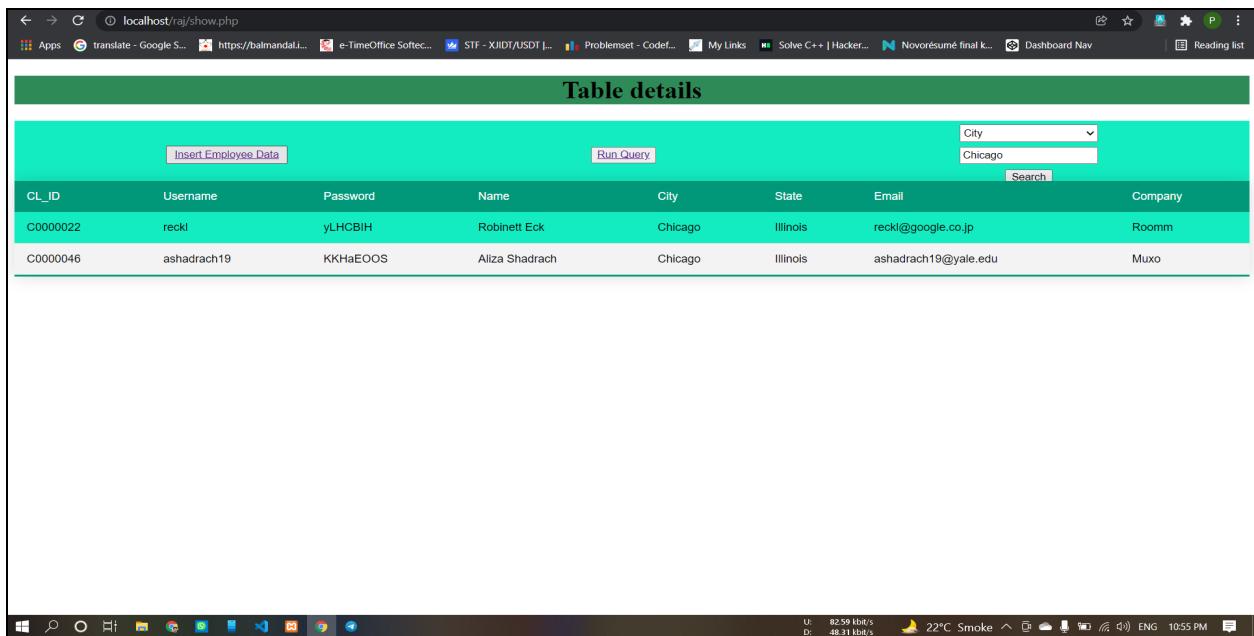
The results of the query are displayed in a table:

	Emp_ID	Dept_ID	Username	Password	Designation	Name	City	State
196	E0000195	D0000011	ochominski5e	FmmUyuwrDIOf	Onida Chominski	Junior	West Virginia	Hi
197	E0000196	D0000011	lferrick5f	TB6xTA12zC	Leonelle Ferrick	Senior	Tennessee	M
198	E0000197	D0000011	mdurrand5g	2lgP1EPQ0kk	Mariam Durrand	Intern	California	Cl
199	E0000198	D0000011	cstyan5h	c7YmlwQGZgo	Cosimo Sty'an	Intern	Texas	Lu
200	E0000199	D0000011	rosan5i	cGT9QSouuW	Ravid Osan	Intern	Michigan	Le
201	E0000200	D0000011	akaliszewski5j	dZ52jcsv	Abbott Kaliszewski	Junior	California	Bc
202	E0000201	D0000011	fapted5k	Bn4AxYljC	Friederike Apted	Senior	Maryland	Bc
203	E111	D0000001	jkbb	dvkv2021	bk	bbk	kb	bk
204	E1345678	D0000001	DVKVh	dvkv2021	d	g	g	l;
205	E7777777	D0000005	jemish	ijerninruw	Senior	kdjabv	jekn	lko
206	E9999999	D0000001	zzzzzz	dvkv2021	Intern	vin	vadodra	Gu

## ❖ For Searching Queries :

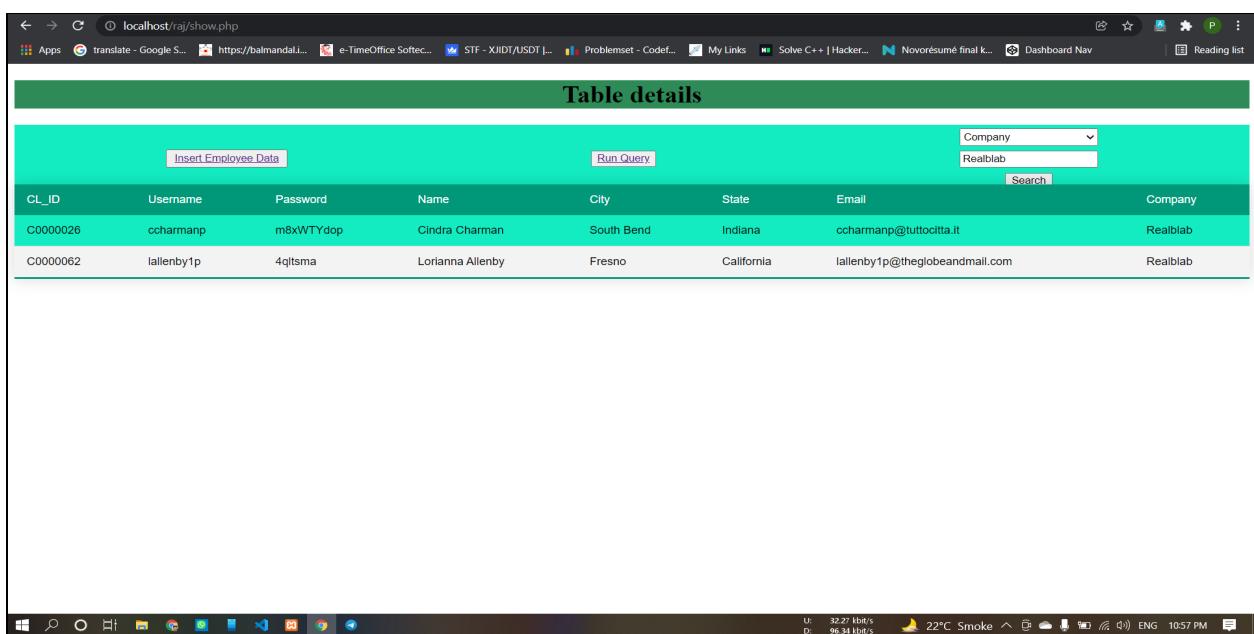
By this functionality we would be able to sort out the client data by first choosing the attribute (i.e., column) for which we want to sort and then entering a value for that attribute. Here we have given two of those choices.

→ Here we have made a search Query which searches the clients who work in Chicago City.



CL_ID	Username	Password	Name	City	State	Email	Company
C0000022	reckl	yLHCBIH	Robinett Eck	Chicago	Illinois	reckl@google.co.jp	Roomm
C0000046	ashadrach19	KKHaEOOS	Aliza Shadrach	Chicago	Illinois	ashadrach19@yale.edu	Muxo

→ This Query will search for the Clients who work in Realblab Company.



CL_ID	Username	Password	Name	City	State	Email	Company
C0000026	ccharmanp	m8xWTYdop	Cindra Charman	South Bend	Indiana	ccharmanp@tuttoicita.it	Realblab
C0000062	lallenby1p	4qltsma	Lorianna Allenby	Fresno	California	lallenby1p@theglobeandmail.com	Realblab

❖ For any Query :

→ Here, we have run a simple query through a web page.

The screenshot shows a web browser window with the URL `localhost/aj/any.php`. The title bar says "Run Any Query". In the main area, there is a code editor containing the following SQL query:

```
select *  
from "TMS"."Project";
```

Below the code editor are three buttons: "Insert Employee Data", "show data", and "Run Query". Underneath these buttons is a table with 10 rows and 8 columns, labeled "field 1" through "field 8". The table contains the following data:

field 1	field 2	field 3	field 4	field 5	field 6	field 7	field 8
P000001	E0000011	C000001	Baxter Whitcher	2542004	2010-09-04	2015-09-10	87
P000002	E0000022	C000002	Agretha Taillard	5179852	2011-04-02	2019-08-04	71
P000003	E0000033	C000003	Kim Sommerlie	2255144	2010-01-06	2018-01-02	74
P000004	E0000044	C000004	Edi Marle	3808743	2012-12-20	2021-07-25	12
P000005	E0000055	C000005	Alissa Coal	9155680	2010-10-15	2018-09-28	86
P000006	E0000066	C000006	Keary Postlewhite	6907652	2010-05-17	2017-10-29	5
P000007	E0000077	C000007	Gwenora Hyde-Chambers	1137501	2012-06-10	2019-09-30	47
P000008	E0000088	C000008	Cellinda Rodenburgh	6795926	2010-02-02	2016-07-27	43
P000009	E0000099	C000009	Arda Tipperton	3489967	2013-10-30	2016-11-11	86
P000010	E0000191	C000010	Devondra Hook	4889230	2010-02-18	2017-06-14	65

The browser's status bar at the bottom shows network activity: U: 0.00 kb/s, D: 0.00 kb/s, 23°C Smoke, ENG, 10:22 PM.

→ We can see the output of the query that we have written which contains all the details of the project.

❖ Appendix :

```
<?php  
$insert= "0";  
$name_error="0";  
$phone_error="0";  
  
if(isset($_POST['name'])){  
  
if($_POST['emp_id']==""){  
$name_error="1";
```

```

}

if($_POST['username']==""){
    $phone_error="1";
}

if($name_error=="0" && $phone_error=="0"){

    include "conection.php";

    $emp_id = $_POST['emp_id'];
    $dept_id = $_POST['dept_id'];
    $username = $_POST['username'];
    $password = $_POST['password'];
    $dest = $_POST['dest'];
    $name = $_POST['name'];
    $city = $_POST['city'];
    $state = $_POST['state'];
    $email = $_POST['email'];
    $pro = $_POST['pro'];

    $sql="INSERT INTO \"TMS\".\"Employee\" VALUES( '$emp_id',
'$dept_id','$username','$password','$dest','$name','$city','$state','$email','$pro' );"

// echo $sql;
$r=pg_query($con,$sql);

if($r){
    $insert=true;
    $insert= "true";

    echo "success";
}
else{
    echo "FAIL";
}
}

```

```

// $insert=false;
echo" ERROR: $sql <br> $con->error";
}

}

}

?>

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Parul Uni</title>
<link rel="stylesheet" href="style.css?v=3">
</head>
<body>
<div class="coin">

<h1>Fill Now</h1>
<button><a href="show.php">Show Client Data</a></button>
<button><a href="any.php">Run Query</a></button>
<h4>Enter Your Details Properly</h4>
<form action="index.php" method="post" class="form_class">

<input type="text" name="emp_id" id="emp_id" placeholder="Enter Your Employee ID ">

<input type="text" name="dept_id" id="dept_id" placeholder="Enter Your dept ID ">

<input type="text" name="username" id="username" placeholder="Enter Your username ">

<input type="password" name="password" id="password" placeholder="Enter Your Password ">

<input type="text" name="dest" id="dest" placeholder="Enter Your Destination">

<input type="text" name="name" id="name" placeholder="Enter your name">

<input type="text" name="city" id="city" placeholder="Enter Your City ">

```

```

<input type="text" name="state" id="state" placeholder="Enter Your State ">

<input type="email" name="email" id="email" placeholder="Enter your email">

<input type="text" name="pro" id="pro" placeholder="Your Progress">

<br>
<?php
if($name_error == "1" || $phone_error == "1")
{
echo"<div class='err'>Employee And Username Number Are Required !! </div>";

}

if($insert=="true"){
    echo"<div class='msg'>submit success</div>";
}

?>
<button class="btn" >Submit</button>
</form>

</div>
<script src="index.js"></script>

</body>
</html>

```

- ❖ GitHub link (For further codes) :  
 → <https://github.com/raj201901306/Task-Management-System>