Group: **Forklift**
Cara Johnson, Michael Sederberg, Nicole Curtis

User	ID	Username	Password	Name							
Project	ID	Title	Summary	OwnerID	Deleted						
Task	ID	UserID	ProjectID	Title	Summary	Due Date	Completed Date	Status	PercentComplete	Deleted	StartDate
Message	ID	projectID	Content	UserID	TimePublished	Deleted	Priority	EditDate			
Hours	ID	UserID	ParentID	TaskType							
ProjectUsers		ProjectID	UserID								
TaskUsers		TaskID	UserID								

Table names, columns, and keys

- User (UserID, Username, Password, FirstName, LastName)
- Project (ProjectID, Title, Summary, OwnerID, Deleted)
    - Foreign Key OwnerID references User
- Task (TaskID, UserID, ProjectID, Title, Summary, DueDate, CompletedDate, Status, PercentComplete, Deleted, StartDate)
    - Foreign Key UserID references User
    - Foreign Key ProjectID references Project
- Message (MessageID, ProjectID, UserID, Content, TimePublished, EditDate, Priority, Deleted)
    - Foreign Key ProjectID references Project
    - Foreign Key UserID references User
- Hours (HoursID, UserID, ParentID, TaskType)
    - Foreign Key UserID references User
    - Foreign Key ParentID references either Project or Task
- ProjectUsers (ProjectID, UserID)
    - Foreign Key UserID references User
    - Foreign Key ProjectID references Project
- TaskUsers (TaskID, UserID)
    - Foreign Key TaskID references Task
    - Foreign Key UserID references User

Explanation of Tables/Columns and Entity Representation

- **User -** holds all of the registered users.
- Entity: This represents users who wish to collaborate on projects with other users
    - UserID: unique identifier for the user
    - Username: user’s username
    - Password: user’s password

- FirstName: user’s first name
- LastName: user’s last name

- **Project -** holds the data for projects.
- Entity: This represents the different projects people can collaborate on.
    - ProjectID: unique identifier for the project
    - Title: title of the project
    - Summary: project summary
    - OwnerID: id of the user who created the project
    - Deleted: if the project has been deleted
- **Task -** holds Task data for a given User in a given Project.
- Entity: This represents a task that is assigned for a specific user to accomplish to help contribute to the current project
    - TaskID: unique identifier for the task
    - UserID: foreign key to associate the task with a user
    - ProjectID: foreign key to associate the task with a project
    - Title: title of task
    - Summary: summary of task
    - StartDate: the start date set for the task
    - DueDate: the due date set for the task
    - CompletedDate: the date the task is completed
    - Status: the current status of the task (inProgress/complete)
    - PercentComplete: the percent of the task completed
    - Deleted: if the task has been deleted
- **Message -** holds info about messages which serve as a pinboard to allow people to communicate in the project.
- Entity: This represents different “sticky note” like messages that members want others of the project to see
    - MessageID: unique identifier for the message
    - ProjectID: foreign key to associate the message with a project
    - UserID: foreign key to associate the message with the user who wrote it
    - Content: content of the message
    - TimePublished: the time the message was created
    - EditDate: the last time the message was edited
    - Priority: whether or not the message is marked as priority
    - Deleted: if the message has been deleted
- **Hours -** holds the hours spent on each project or task
- Entity: This represents the hours that a user spends on their assigned task or in general towards the overall project
    - HoursID: unique identifier for the hour
    - UserID: foreign key to associate the hour with its user
    - ParentID: foreign key to associate the hour with either a project or a task
    - TaskType: enum that is either ‘project’ or ‘task’
- **ProjectUsers -** mapping table to easily get all of the users that belong to a given projectID.

- Entity: This represents all the users that are associated with a given project and all the projects a user is currently a part of
    - ProjectID: foreign key that maps to the project
    - UserID: foreign key that maps to the user
- **TaskUsers** - mapping table to easily get all of the users that are working on a given TaskID
- Entity: This represents all the tasks that a user is currently in charge of, across all their projects
    - TaskID: foreign key that maps to the task
    - UserID: foreign key that maps to the user

Evidence of Normalization

- Each table has a unique key.
- The tables use foreign keys to relate to other tables, instead of having duplicate data between the tables.
- The TaskUsers and ProjectUsers tables are especially useful to show relationships between the data because we don’t want to store multiple Users in each task or Project, but instead have a table to connect them together to know which users belong to each task and project.