ER Diagram Documentation

ER Diagram Documentation

This document describes the entities, attributes, and relationships within the Entity-Relationship (ER) diagram for a project management system.

1. Entities:

* **User:** Represents a user within the system.
    * **Attributes:**
        * `userId` (PK, INT): Unique identifier for each user. (Primary Key)
        * `email` (VARCHAR, UNIQUE): User's email address. Must be unique.
        * `passwordHash` (VARCHAR): Hashed password for security.
        * `role` (VARCHAR): User's role (e.g., admin, member).
        * `createdAt` (TIMESTAMP): Timestamp indicating when the user was created.
        * `updatedAt` (TIMESTAMP): Timestamp indicating when the user was last updated.
        * `userName` (VARCHAR): User's display name.

* **Project:** Represents a project within the system.
    * **Attributes:**
        * `projectId` (PK, INT): Unique identifier for each project. (Primary Key)
        * `name` (VARCHAR): Name of the project.
        * `userId` (INT, FK): Foreign Key referencing the User who created the project.
        * `createdAt` (TIMESTAMP): Timestamp indicating when the project was created.
        * `updatedAt` (TIMESTAMP): Timestamp indicating when the project was last updated.

* **Task:** Represents a task within a project.
    * **Attributes:**
        * `taskId` (PK, INT): Unique identifier for each task. (Primary Key)
        * `projectId` (INT, FK): Foreign Key referencing the Project the task belongs to.
        * `name` (VARCHAR): Name of the task.
        * `description` (TEXT): Description of the task.
        * `dueDate` (DATE): Due date for the task.
        * `priority` (VARCHAR): Priority level of the task (e.g., high, medium, low).
        * `status` (VARCHAR): Status of the task (e.g., open, in progress, completed).
        * `assignedTo` (INT, FK): Foreign Key referencing the User assigned to the task.
        * `createdAt` (TIMESTAMP): Timestamp indicating when the task was created.
        * `updatedAt` (TIMESTAMP): Timestamp indicating when the task was last updated.

* **ProjectUser:** Represents the relationship between a Project and a User (many-to-many relationship). Tracks user invitations to projects.
    * **Attributes:**
        * `projectUserId` (PK, INT): Unique identifier for each project-user association. (Primary Key)
        * `projectId` (INT, FK): Foreign Key referencing the Project.
        * `userId` (INT, FK): Foreign Key referencing the User.
        * `invitationStatus` (VARCHAR): Status of the invitation (e.g., pending, accepted, rejected).

* **ActivityLog:** Records activities performed within the system.
    * **Attributes:**

* `logId` (PK, INT): Unique identifier for each log entry. (Primary Key)

* `projectId` (INT, FK): Foreign Key referencing the Project (can be NULL for system-wide actions).

* `userId` (INT, FK): Foreign Key referencing the User who performed the action.

* `action` (VARCHAR): Type of action performed.

* `timestamp` (TIMESTAMP): Timestamp indicating when the action was performed.

* `details` (TEXT): Detailed information about the action.

2. Relationships:

* **creates (User 1:N Project):** A User can create multiple Projects.

* **contains (Project 1:N Task):** A Project contains multiple Tasks.

* **assigned to (Task 1:1 User):** A Task is assigned to one User. (Note: Could be 1:0 if a task is unassigned)

* **invites (Project 1:N ProjectUser):** A Project can have multiple User invitations (through ProjectUser).

* **joins (User 1:N ProjectUser):** A User can join multiple Projects (through ProjectUser).

* **generates (Project 1:N ActivityLog):** A Project can generate multiple ActivityLog entries.

* **performs (User 1:N ActivityLog):** A User can perform multiple actions, generating ActivityLog entries.

3. Cardinality and Participation:

The cardinality (1:1, 1:N, M:N) and participation (total or partial) of each relationship are indicated above in parentheses. For instance, "creates (User 1:N Project)" means one user can create many projects. The participation isn't explicitly defined but can be inferred from the context (e.g., a User doesn't *have* to create a Project, so participation is partial).

4. Primary and Foreign Keys:

Primary keys (PK) and foreign keys (FK) are indicated for each attribute. Foreign keys enforce referential integrity between entities.

This documentation provides a clear and concise description of the ER diagram, allowing for a better understanding of the database structure and its relationships. Further details regarding data types (e.g., VARCHAR length, INT size) may be added for increased specificity.