

Bug Tracker Database Schema Documentation

This document outlines the database schema for a simplified bug tracking system, often referred to as "Jira-lite." The schema is designed to manage users, projects, issues, and related entities efficiently. It includes essential features and optional tables for extended functionality.

Essential Features

Users Table

The `users` table stores information about each user in the system.

- **id**: Integer, Primary Key
- **name**: Varchar, User's name
- **email**: Varchar, User's email address
- **role**: Varchar, User's role in the system
- **created_at**: Datetime, Timestamp of when the user was created

Projects Table

The `projects` table contains details about each project.

- **id**: Integer, Primary Key
- **name**: Varchar, Project name
- **description**: Text, Detailed description of the project
- **created_by**: Integer, Foreign Key referencing `users.id`
- **created_at**: Datetime, Timestamp of when the project was created

Project Users Table

The `project_users` table manages the association between users and projects.

- **project_id**: Integer, Foreign Key referencing `projects.id`
- **user_id**: Integer, Foreign Key referencing `users.id`
- **role_in_project**: Varchar, Role of the user within the project

Issues Table

The `issues` table tracks individual issues within projects.

- **id**: Integer, Primary Key

- **title:** Varchar, Title of the issue
- **description:** Text, Detailed description of the issue
- **reporter_id:** Integer, Foreign Key referencing `users.id`
- **assignee_id:** Integer, Foreign Key referencing `users.id`
- **project_id:** Integer, Foreign Key referencing `projects.id`
- **status_id:** Integer, Foreign Key referencing `statuses.id`
- **priority_id:** Integer, Foreign Key referencing `priorities.id`
- **created_at:** Datetime, Timestamp of when the issue was created
- **updated_at:** Datetime, Timestamp of the last update

Statuses Table

The `statuses` table defines possible statuses for issues.

- **id:** Integer, Primary Key
- **name:** Varchar, Name of the status

Priorities Table

The `priorities` table defines priority levels for issues.

- **id:** Integer, Primary Key
- **name:** Varchar, Name of the priority level

Comments Table

The `comments` table stores comments made on issues.

- **id:** Integer, Primary Key
- **issue_id:** Integer, Foreign Key referencing `issues.id`
- **author_id:** Integer, Foreign Key referencing `users.id`
- **content:** Text, Content of the comment
- **created_at:** Datetime, Timestamp of when the comment was created

Optional Features

Labels Table

The `labels` table allows categorization of issues with labels.

- **id:** Integer, Primary Key
- **name:** Varchar, Name of the label

- **color:** Varchar, Color associated with the label

Issue Labels Table

The `issue_labels` table manages the association between issues and labels.

- **issue_id:** Integer, Foreign Key referencing `issues.id`
- **label_id:** Integer, Foreign Key referencing `labels.id`

Attachments Table

The `attachments` table stores file attachments related to issues.

- **id:** Integer, Primary Key
- **issue_id:** Integer, Foreign Key referencing `issues.id`
- **file_path:** Varchar, Path to the attached file
- **uploaded_at:** Datetime, Timestamp of when the file was uploaded

Audit Logs Table

The `audit_logs` table records actions taken by users for auditing purposes.

- **id:** Integer, Primary Key
- **user_id:** Integer, Foreign Key referencing `users.id`
- **action:** Varchar, Description of the action performed
- **entity_type:** Varchar, Type of entity affected (e.g., Issue, Project)
- **entity_id:** Integer, ID of the affected entity
- **created_at:** Datetime, Timestamp of when the action was logged

Implementation Notes

- **Foreign Key Constraints:** Ensure all foreign key constraints are correctly implemented to maintain data integrity.
- **Migration Tool:** Use a migration tool like EF Core Migrations in .NET to manage schema changes efficiently.
- **Database Options:** This schema can be implemented in databases like PostgreSQL or SQL Server, which support the necessary features and constraints.

This schema provides a robust foundation for a bug tracking system, allowing for efficient management of users, projects, and issues, with optional features for enhanced functionality.