
GROUP ID:-16

BUG MANAGEMENT SYSTEM

Group Representative:-

Vivek Doshi ID: 202112063

Group Members:-

Mihir Popat ID: 202112073

Shail Parekh ID: 202112128

Yash Trivedi ID: 202112060

Scope

Every software development company encounters issues in their project which we call **bugs**. Our structure focuses on dealing with these bugs in an organized and efficient way. Here, our task will be to build a database which will help the organization to manage these data.

This system is advantageous for both customers and the firm. For a company it gives a bird's eye view of the current bugs which helps in prioritizing severe bugs to clear backlog, keeps the bug history hence eliminating repetition of work. Also allows managers to generate reports which can be used to find any discrepancies. It helps a customer to find the current status of their bug and when the fix is available to them.

Here, information will be available in context of Customer, Employee, Projects, Bug Details and their Patch Details.

Requirements

- Customer is an entity which has the details of customers for whom the company create the projects. Details will include customer_Id, customer_name, email_Id, phone_No, address, city, state.
- Employee is an entity which will store details of all the employees of the company like, employee_Id, employee_name, team_Id, email_Id, phone_No.
- Teams is an entity which has information of which employee belongs to which team and each team belongs to which department. So details would be team_id, employee_id, department_id, team_name, team_lead_id.
- Departments is an entity which stores information about various departments of the company. Details stored will be department_id, department_name, manager_id.
- Projects is an entity which stores information about every project, so details will be project_id, project_name, customer_id, project_type, project_startDate, project_endDate.
- Bug_Details is an entity which stores details of the bug reported like bug_id, bug_title, bug_description, status, type, severity, project_id, report_date, reporter_id.

- Patch_Details is an entity which stores which team is working on the bug and fixed it, so details will include patch_id, bug_id, team_id, patch_date, patch_description.
- Update_Details is an entity which stores all the updates done to a specific project. Multiple bugs will be fixed for a project but will be updated together under 1 version. Details include version_id, project_id, and status.
- Release_Details is an entity which stores which bug is fixed under which version. Details include Patch_id, version_id, release_date.
- One employee will be part of only one team.
- One team can have more than one employee but only one team lead.
- One department can have more than one team but only one manager.
- One customer can have more than one project.
- A customer or a team can report more than one bug.
- One project can have more than one bug.

- One bug will be assigned to and fixed by only one team.
- One bug will have more than one patch.
- One version will have more than one patch.

Queries

- 1) List all the bugs reported by customers.
- 2) Give bug count for each project.
- 3) Give customers details who reported high priority bug.
- 4) Find the bug which required most number of fixes.
- 5) Give details about the team that worked on bugs reported by a specific customer.
- 6) Count the number of bus fixed before any date.
- 7) Give customer details which reported bugs as well as enhancements.
- 8) Give details of top 3 team (including employee name and their manager) which fixed most number of bugs.
- 9) Give details of bugs which were reopened by the reporter.
- 10) Give details of QA team (Employee names) which detected most number of bugs.

- 11) Find highest priority bug fixed the earliest.
- 12) Give project details with no bug reported after 30 days of release.
- 13) Give month wise number of bugs were logged in year 2020.
- 14) Give project details where more than one team fixed the bugs.
- 15) How many bugs were logged for a specific project type?
- 16) Find all the teams who have worked on more patches then average number of patches.
- 17) Fetch top 3 bugs reported from customers coming from a specific city.
- 18) List down details of project updates and the team that worked on its release
- 19) Find the projects with more updates than the average total updates
- 20) Give team details which fixed all types of bug
- 21) List down all the resolved bugs and their update release status
- 22) List down the project which had an update within 30 days of their release