DBMS Assignment

Chatbot for Agile Project Management

A chatbot for agile project management is a software application designed to assist individuals and teams in managing agile projects through conversational interfaces.
The platform provides individualized chatbots for each project, granting team members independent access to retrieve updates and details on all ongoing scrum ceremonies within their respective projects, without requiring assistance from others.

Relevance

- **Real-time Updates**: Team members can access real-time updates and details on ongoing scrum ceremonies within their projects, ensuring they are informed of the latest developments.

- **Autonomy**: With individualized chatbots, team members can independently retrieve information without relying on assistance from colleagues or project managers, promoting autonomy and self-sufficiency.

- **Improved Collaboration**: By centralizing project-related communication and information retrieval, the platform encourages collaboration among team members, fostering a more cohesive and efficient working environment.

- **Productivity**: Quick access to relevant project information helps streamline workflows and decision-making processes, leading to increased productivity and faster project execution.

- **Agile Methodology Alignment**: The platform's support for agile project management methodologies ensures that teams can effectively implement scrum ceremonies and adhere to agile principles, thereby enhancing project success rates.

Tables

1. role:
    a. role_id (INT, Primary Key)
    b. role_name (VARCHAR(255))
2. user:
    a. user_id (INT, Primary Key)
    b. role_id (INT, Foreign Key)
    c. user_name (VARCHAR(255))
    d. email (VARCHAR(255))
    e. password (VARCHAR(255))
3. project:
    a. project_id (INT, Primary Key)
    b. project_name (VARCHAR(255))
    c. description (TEXT)
    d. start_date (DATE)
    e. end_date (DATE)
4. ceremony:
    a. ceremony_id (INT, Primary Key)
    b. ceremony_name (VARCHAR(255))
5. messages:
    a. message_id (INT, Primary Key)
    b. user_id (INT, Foreign Key)
    c. message_content (VARCHAR(255))
    d. timestamp (TIMESTAMP)
    e. ceremony_id (INT,Foreign Key)
6. sprints:
    a. sprint_id (INT, Primary Key)
    b. sprint_number (INT)
    c. start_date (DATE)
    d. end_date (DATE)
    e. status (VARCHAR(50))
    f. project_id (INT, Foreign Key)
7. sprint_backlog:
    a. backlog_id (INT, Primary Key)
    b. sprint_id (INT, Foreign Key)
    c. user_story (TEXT)
    d. status (VARCHAR(50))
8. daily_standup_meetings:

    a. standup_id (INT, Primary Key)
    b. sprint_id (INT, Foreign Key)
    c. date (DATE)
    d. notes (TEXT)
9. sprint_review_meetings:
    a. review_id (INT, Primary Key)
    b. sprint_id (INT, Foreign Key)
    c. date (DATE)
    d. results (TEXT)
    e. feedback (TEXT)
10. sprint_retrospective_meetings:
    a. retrospective_id (INT, Primary Key)
    b. sprint_id (INT, Foreign Key)
    c. date (DATE)
    d. attendees (TEXT)
    e. discussions (TEXT)

Normalization

1NF (First Normal Form):

- Ensure each table cell contains only atomic (indivisible) values.
- Eliminate repeating groups by organizing data into rows and columns.
- Each column should have a unique name, and the order of rows and columns shouldn't matter.

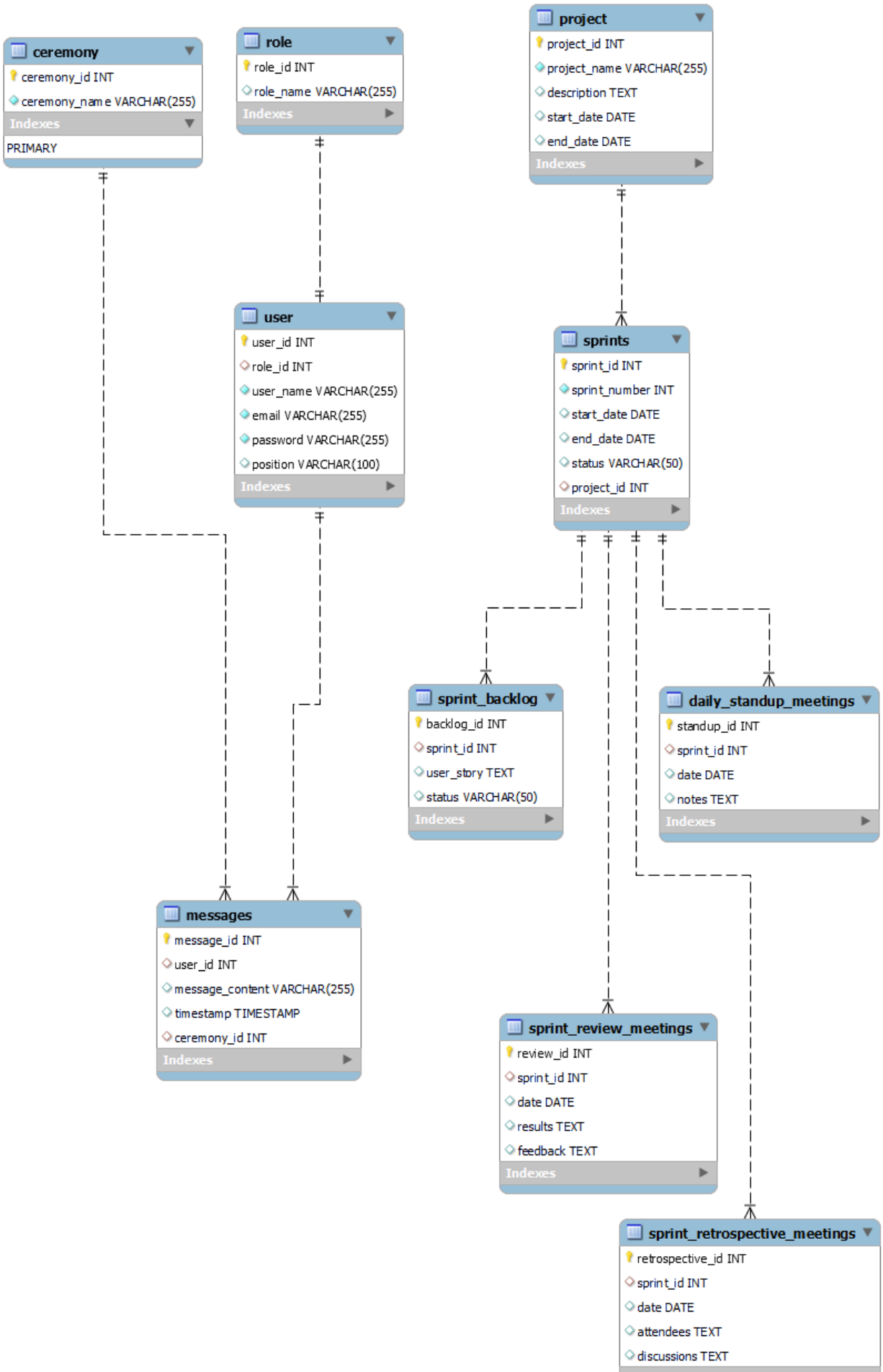
2NF (Second Normal Form):

- Satisfy 1NF requirements.
- Remove partial dependencies, ensuring non-prime attributes depend on the entire primary key.

3NF (Third Normal Form):

- Satisfy 2NF requirements.
- Eliminate transitive dependencies, ensuring non-prime attributes depend only on the primary key, not on other non-prime attributes.

Here all tables conform to the requirements of the first normal form.Here all tables appear to be in the second normal form (2NF) as there are no partial dependencies.
Here all tables appear to be in the third normal form (3NF) as there are no transitive dependencies. Each non-prime attribute depends directly on the primary key of its respective table.


ceremony
ceremony_id INT
ceremony_name VARCHAR(255)
Indexes
PRIMARY
role
role_id INT
role_name VARCHAR(255)
Indexes
project
project_id INT
project_name VARCHAR(255)
description TEXT
start_date DATE
end_date DATE
Indexes
user
user_id INT
role_id INT
user_name VARCHAR(255)
email VARCHAR(255)
password VARCHAR(255)
position VARCHAR(100)
Indexes
sprints
sprint_id INT
sprint_number INT
start_date DATE
end_date DATE
status VARCHAR(50)
project_id INT
Indexes
sprint_backlog
backlog_id INT
sprint_id INT
user_story TEXT
status VARCHAR(50)
Indexes
daily_standup_meetings
standup_id INT
sprint_id INT
date DATE
notes TEXT
Indexes
messages
message_id INT
user_id INT
message_content VARCHAR(255)
timestamp TIMESTAMP
ceremony_id INT
Indexes
sprint_review_meetings
review_id INT
sprint_id INT
date DATE
results TEXT
feedback TEXT
Indexes
sprint_retrospective_meetings
retrospective_id INT
sprint_id INT
date DATE
attendees TEXT
discussions TEXT