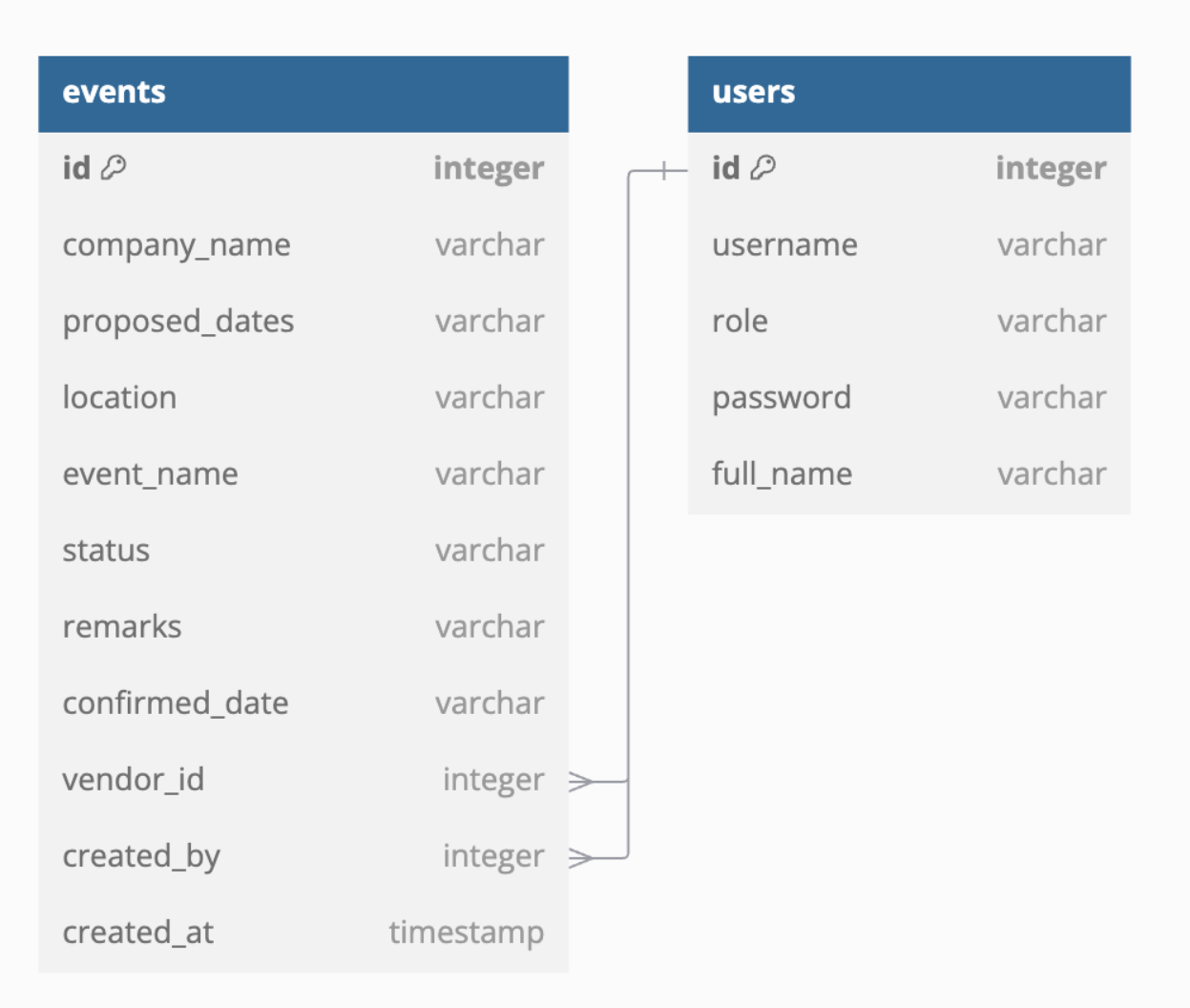
ERD



1. Tables and Columns

events Table

The events table stores details about events being managed. It contains the following columns:

- **id** *(integer)*: The primary key that uniquely identifies each event.
- **company_name** *(varchar)*: The name of the company organizing or related to the event.
- **proposed_dates** *(varchar)*: Proposed date(s) for the event.
- **location** *(varchar)*: Location where the event will take place.
- **event_name** *(varchar)*: The name or title of the event.
- **status** *(varchar)*: Status of the event (e.g., pending, approved, rejected).
- **remarks** *(varchar)*: Additional comments or remarks regarding the event.
- **confirmed_date** *(varchar)*: The date when the event is confirmed.

- **vendor_id** *(integer)*: A foreign key referencing the id in the users table (likely identifying the vendor responsible for the event).
- **created_by** *(integer)*: A foreign key referencing the id in the users table (identifying the user who created the event).
- **created_at** *(timestamp)*: The date and time when the event record was created.

users Table
The users table stores user information. It contains the following columns:

- **id** *(integer)*: The primary key that uniquely identifies each user.
- **username** *(varchar)*: The username of the user.
- **role** *(varchar)*: The role of the user (e.g., admin, vendor, manager).
- **password** *(varchar)*: The user's password.
- **full_name** *(varchar)*: The full name of the user

2. Relationships

- **vendor_id in events → id in users**:
  - This represents a foreign key relationship.
  - The vendor_id identifies which user (from the users table) is responsible for the event.
  - This could signify vendors managing specific events.
- **created_by in events → id in users**:
  - This foreign key indicates which user created the event record in the events table.
  - This helps track accountability for event creation.

3. Use Case Explanation

- **Events Table**: It holds details about various events like name, status, location, and date.
- **Users Table**: It manages user information, including roles and credentials.
- **Relationships**:
  - Each event has a **vendor** assigned (through vendor_id).
  - Each event also logs who created it (through created_by).

4. Example Scenario

- A user with id = 5 (a vendor role) is assigned to manage an event with vendor_id = 5.
- Another user with id = 2 (an admin role) created the event and is tracked using created_by = 2.
- The status column indicates the current state of the event, such as "pending," "approved," or "rejected."