Entity Relationship Diagram

For

REVELS MANAGEMENT SYSTEM

Version 1.0

Prepared by

Akhil Varanasi 57 - 230953496

Saurabh Sharma 43 - 230953374

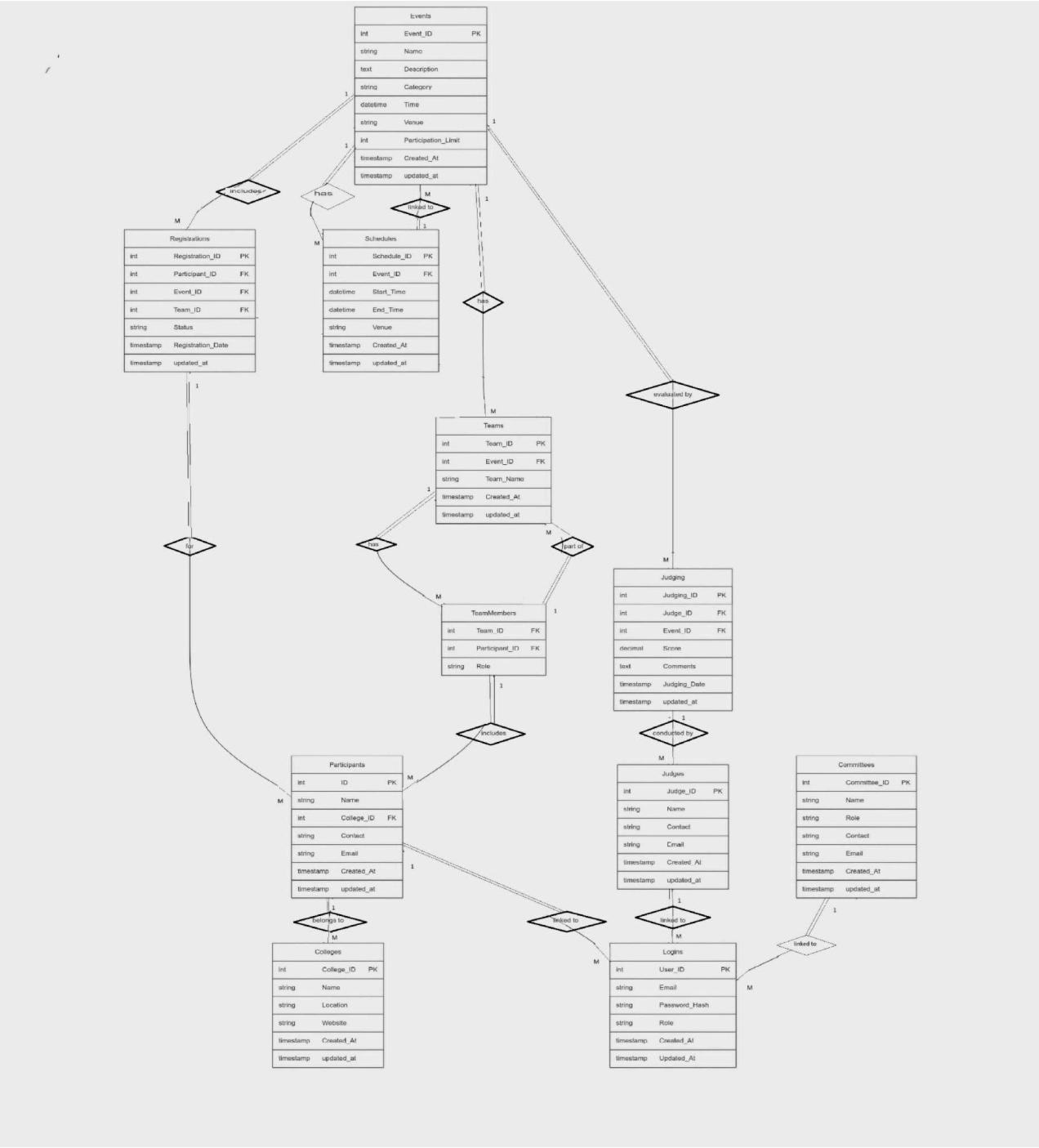
Shreyas Kumar PM 33- 230953300

Manipal Institute of Technology

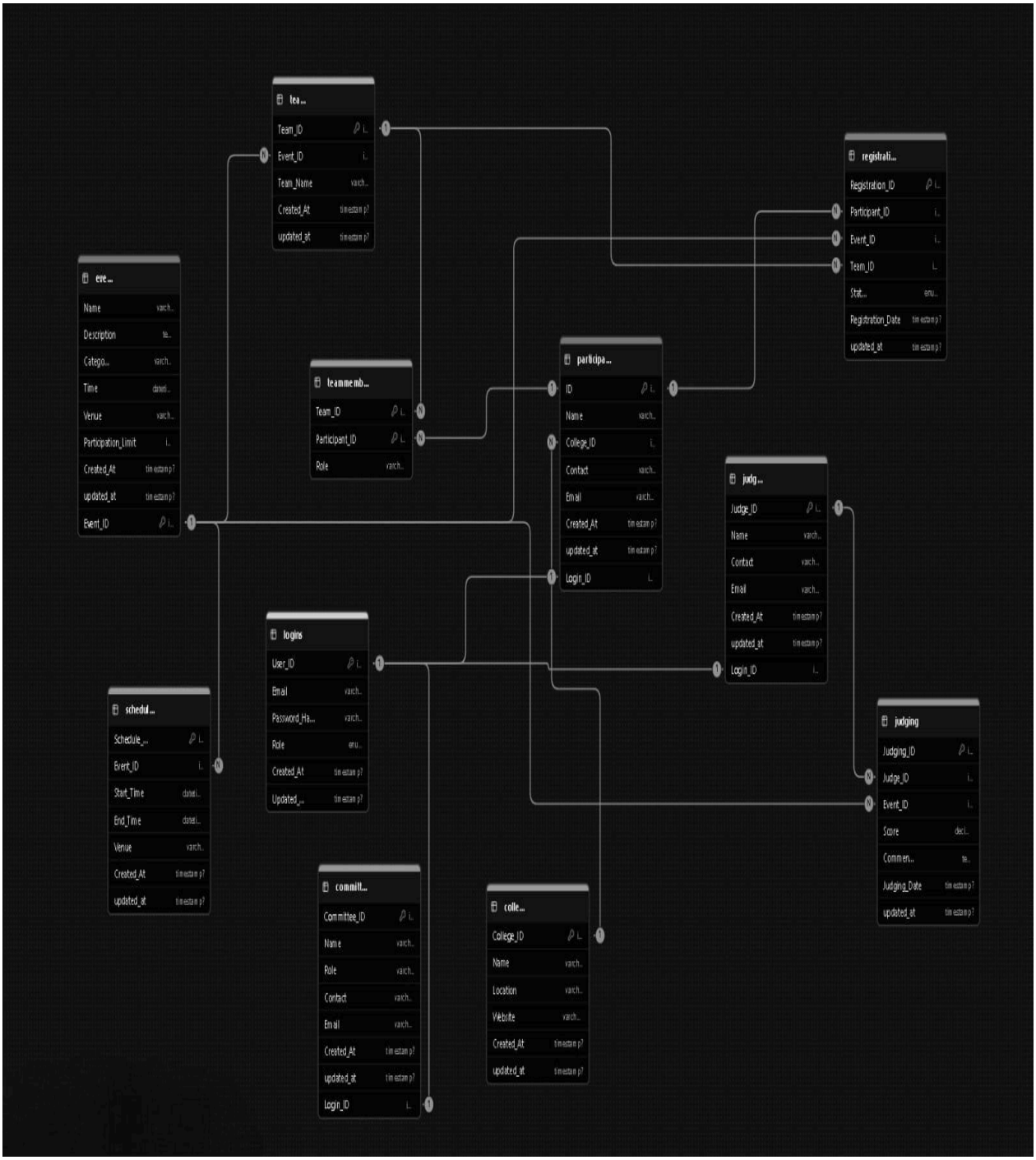
1st April 2025

Analysis Models

- **Entity-Relationship Diagrams (ERD):** Diagrams depicting the relationships between key data entities such as Users, Events, Teams, and Scores.



- **Schema Diagram:** A visual representation of a database structure. It shows the tables, columns, data types, primary keys, foreign keys, and relationships between tables in a database.



Schema Reduction Using Functional Dependency-Based Normalization

Step 1: Identify Relations and Functional Dependencies

From the ERD, we identify the following relations and their attributes:

Events

Attributes: (Event_ID, Name, Description, Category, Time, Venue, Participation_Limit, Created_At, Updated_At)
Functional Dependency: Event_ID → Name, Description, Category, Time, Venue, Participation_Limit, Created_At, Updated_At

Schedules

Attributes: (Schedule_ID, Event_ID, Start_Time, End_Time, Created_At, Updated_At)
Functional Dependency: Schedule_ID → Event_ID, Start_Time, End_Time, Created_At, Updated_At

Registrations

Attributes: (Registration_ID, Participant_ID, Event_ID, Team_ID, Status, Registration_Date, Updated_At)
Functional Dependency: Registration_ID → Participant_ID, Event_ID, Team_ID, Status, Registration_Date, Updated_At

Teams

Attributes: (Team_ID, Event_ID, Team_Name, Created_At, Updated_At)
Functional Dependency: Team_ID → Event_ID, Team_Name, Created_At, Updated_At

TeamMembers

Attributes: (Team_ID, Participant_ID, Role)
Functional Dependency: (Team_ID, Participant_ID) → Role

Participants

Attributes: (ID, Name, College_ID, Contact, Email, Created_At, Updated_At)
Functional Dependency: ID → Name, College_ID, Contact, Email, Created_At, Updated_At

Colleges

Attributes: (College_ID, Name, Location, Website, Created_At, Updated_At)
Functional Dependency: College_ID → Name, Location, Website, Created_At, Updated_At

Judging

Attributes: (Judging_ID, Judge_ID, Event_ID, Score, Comments, Judging_Date, Updated_At)
Functional Dependency: Judging_ID → Judge_ID, Event_ID, Score, Comments, Judging_Date, Updated_At

Judges

Attributes: (Judge_ID, Name, Contact, Email, Created_At, Updated_At)
Functional Dependency: Judge_ID → Name, Contact, Email, Created_At, Updated_At

Committees

Attributes: (Committee_ID, Name, Role, Contact, Email, Created_At, Updated_At)
Functional Dependency: Committee_ID → Name, Role, Contact, Email, Created_At, Updated_At

Logins

Attributes: (User_ID, Email, Password_Hash, Role, Created_At, Updated_At)
Functional Dependency: User_ID → Email, Password_Hash, Role, Created_At, Updated_At

---

Step 2: Decomposing into Reduced Schemas

To remove redundancy and ensure **3rd Normal Form (3NF)**, we decompose the relations where needed.

Events

- No decomposition needed (already in 3NF).

Schedules

- Schedule_ID is the primary key, and Event_ID is a foreign key.

- No redundant dependencies exist → **Remains unchanged**.

Registrations

- Already in 3NF since Registration_ID determines all attributes.

Teams & TeamMembers

- **Teams** (Team_ID, Event_ID, Team_Name, Created_At, Updated_At)
- **TeamMembers** (Team_ID, Participant_ID, Role)
- No further reduction needed.

Participants & Colleges

- **Participants** (ID, Name, College_ID, Contact, Email, Created_At, Updated_At)
- **Colleges** (College_ID, Name, Location, Website, Created_At, Updated_At)
- No further decomposition needed.

Judging & Judges

- **Judging** (Judging_ID, Judge_ID, Event_ID, Score, Comments, Judging_Date, Updated_At)
- **Judges** (Judge_ID, Name, Contact, Email, Created_At, Updated_At)
- No decomposition needed.

Committees

- Already in 3NF, no further reduction.

Logins

- Already in 3NF, no further reduction.

Step 3: Verify Lossless Join and Dependency Preservation

- **All relations still allow natural joins** to reconstruct the original data.
- **All functional dependencies are preserved**.

Final Reduced Schema

Events

(Event_ID, Name, Description, Category, Time, Venue, Participation_Limit, Created_At, Updated_At)

Schedules

(Schedule_ID, Event_ID, Start_Time, End_Time, Created_At, Updated_At)

Registrations

(Registration_ID, Participant_ID, Event_ID, Team_ID, Status, Registration_Date, Updated_At)

Teams

(Team_ID, Event_ID, Team_Name, Created_At, Updated_At)

TeamMembers

(Team_ID, Participant_ID, Role)

Participants

(ID, Name, College_ID, Contact, Email, Created_At, Updated_At)

Colleges

(College_ID, Name, Location, Website, Created_At, Updated_At)

Judging

(Judging_ID, Judge_ID, Event_ID, Score, Comments, Judging_Date, Updated_At)

Judges

(Judge_ID, Name, Contact, Email, Created_At, Updated_At)

Committees

(Committee_ID, Name, Role, Contact, Email, Created_At, Updated_At)

Logins

(User_ID, Email, Password_Hash, Role, Created_At, Updated_At)

Conclusion

This reduction removes redundancy, prevents anomalies, and preserves dependencies while maintaining a lossless join.