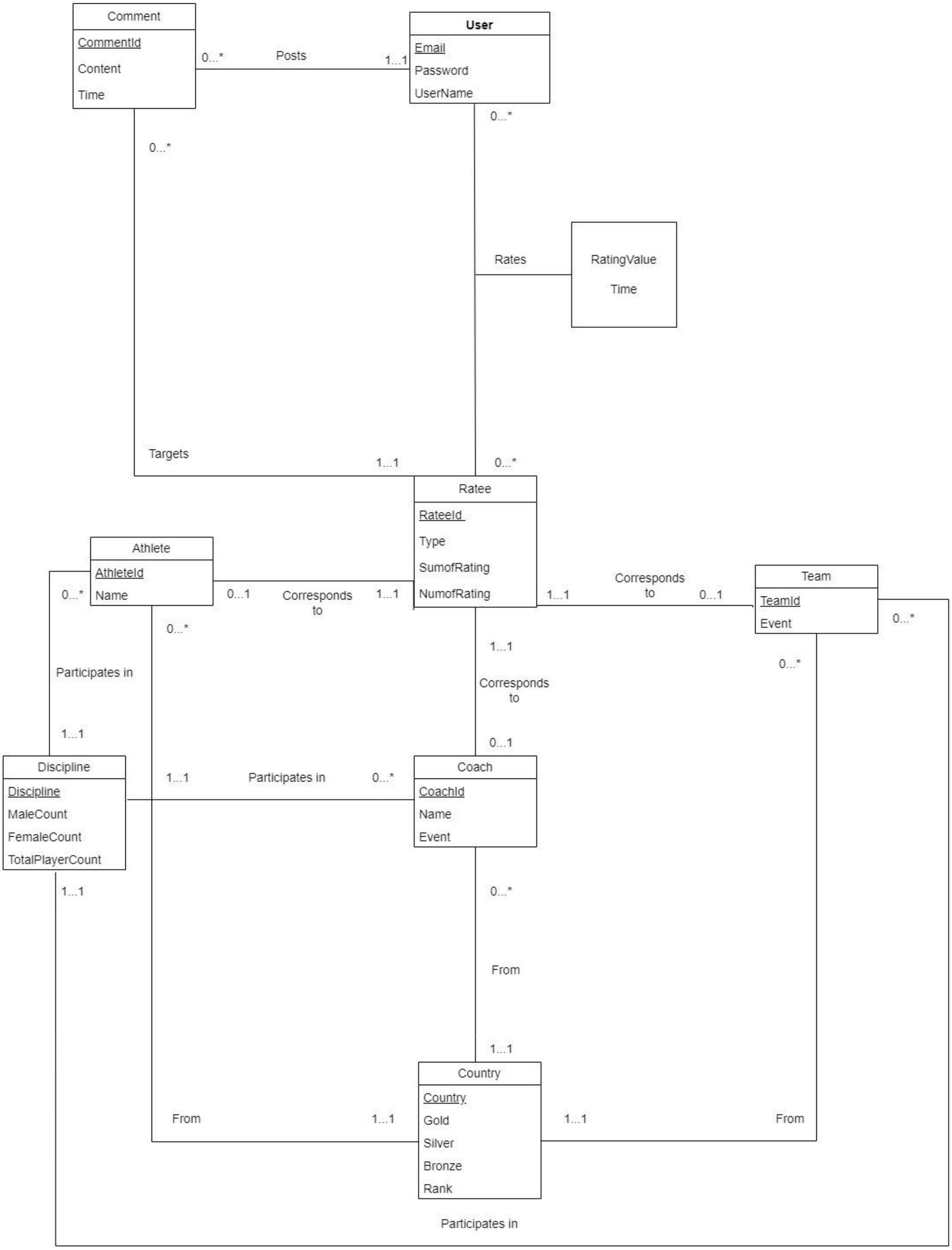
Stage 2: Database Design (Team-003)



Description and Assumption of UML Diagram

Entity

- User
  - Represents a registered user
  - Assume the length of Email will not exceed 127 characters
- Comment
  - Represents a single comment entity.
  - Assume the content of the comment will not exceed 4095 characters
- Ratee
  - Represents every participating entity that can be rated, including team, athlete, and coach
  - Assume the type of Ratee is in [athlete, team, coach]
- Athlete
  - Represents an athlete entity.
  - Assume the length of name does not exceed 50 characters
  - Assume every athlete participates in one and only one discipline
  - Assume every athlete has one and only one nationality
- Coach
  - Represents a coach entity.
  - Assume the length of name does not exceed 50 characters
  - Assume every coach coaches one and only one discipline
  - Assume every coach has one and only one nationality
- Team
  - Represents a team entity.
  - Assume every team participates in one and only one discipline
  - Assume every team has one and only one nationality
- Country
  - Represents a physical country participating in the Olympics.
  - Assume the length of country names do not exceed 50 characters
  - Assume every country name is unique
  - Assume the rank is a positive integer
- Discipline
  - Represents a branch of sport in Olympics
  - Assume each team can only participate in one discipline

Relationship

- Posts(User - Comment)

    - The relationship between "User - Comment" describes that a user can post comment entities, and we assume that each user entity can comments zero or more times. However, each comment entity can only be provided by one user.
- Targets(Comment - Ratee)
    - The relationship between "Comment - Ratee" describes that a comment entity targets towards a specific ratee, and we assume that each comment or retee entity can targets one and only one retee. However, each ratee can be commented zero or more comment or retee entity.
- Rates(User - Ratee)
    - The relationship between "User - Ratee" describes that any users can rate any ratess, and we assume that each user entity can rates zero or more times and each ratees can be rated zero or more times.
- Corresponds to(Ratee - Athlee / Ratee - Coach / Ratee - Team)
    - The relationship between "Ratee - Athlete" , "Ratee - Coach" and "Ratee - Team" describes that a retee can be corresponds to one and only one Athlete, Coach, or Team. However, an Athlete, Coach, or Team entity can be associated with at most one retee entity.
- From(Athlee - Country / Coach - Country / Team - Country)
    - The relationship between "Athlee - Country", "Coach - Country" and "Team - Country" describes that an Athlete, Coach, or Team entity can be from one and only one country, similarly, one country can have multiple Athlete, Coach, or Team associated.
- Participates in (Athlee - Discipline/ Coach - Discipline/ Team - Discipline)
    - The relationship between "Athlee - Discipline", "Coach - Discipline" and "Team - Discipline" describes that an Athlete, Coach, or Team entity can participates in a certain discipline. We assume an Athlete, Coach, or Team entity can only be dedicated for one and only one discipline, and a specific discipline can have multiple Athlete, Coach, or Team associated.

Relational Schema Design

User (
    Email: VARCHAR(127) [PK],
    Password: VARCHAR(31) ,
    Username: VARCHAR(25)
)

Comment (
    CommentId: INT [PK]
    Content: VARCHAR(4095),
    Time: TIMESTAMP,
    PostBy: VARCHAR(127) [FK to User.Email],
    Target: INT [FK to Ratee.RateeId]
)

```
Rates (
        RateBy: VARCHAR(127) [PK] [FK to User.Email]
        Target: INT [PK] [FK to Ratee.RateeId]
        RatingValue: INT,
        Time: TIMESTAMP
)

Ratee (
        RateeId: INT [PK],
        Type: VARCHAR(25),
        SumofRating: INT,
        NumofRating: INT
)

Athlete (
        AthleteId: INT [PK],
        Name: VARCHAR(50),
        Discipline: VARCHAR(25) [FK to Discipline.Discipline],
        Country: VARCHAR (25) [FK to Country.Country],
        RateeId: INT [FK to Ratee.RateeId]
)

Coach (
        CoachId: INT [PK],
        Name: VARCHAR (50),
        Country: VARCHAR (25) [FK to Country.Country],
        Discipline: VARCHAR (25) [FK to Discipline.Discipline],
        Event: VARCHAR (25),
        RateeId:  INT [FK to Ratee.RateeId]
)

Team (
        TeamId: INT [PK],
        Country: VARCHAR (25) [FK to Country.Country],
        Discipline: VARCHAR (25) [FK to Discipline.Discipline],
        Event: VARCHAR (25),
        RateeId:  INT [FK to Ratee.RateeId]
)

Country (
        Country: VARCHAR(50) [PK],
        Gold: INT,
        Silver: INT,
        Bronze: INT,
        Rank: INT
)

Discipline (
        Discipline: VARCHAR (25) [PK],
        MaleCount: INT,
        FemaleCount: INT,
        TotalPlayerCount: INT
)
```

Normalization

Functional Dependencies and Closures:

User (Email,Password,Username):
Email -> Password,Username
{Email}$^+$ = {Email, Password,Username}

Comment (CommentId, Content, Time, PostBy, Target):
CommentId -> Content, Time, PostBy, Target
{CommentId}$^+$ = {CommentId, Content, Time, PostBy, Target}

Rates (RatingId, RatingValue, Time):
(RateBy, Target) -> RatingValue, Time,
{RateBy, Target}$^+$ = {RatingValue, Time, Target, RateBy}

Athlete (AthleteId, Name, Discipline, Country, RateeId)
AthleteId -> Name, Discipline, Country, RateeId
{AthleteId}$^+$ = {AthleteId, Name, Discipline, Country, RateeId}

RateeId -> Name, Discipline, Country, AthleteId
{RateeId}$^+$ = {RateeId, Name, Discipline, Country, AthleteId}

Coach (CoachId, Name, Discipline, Country, Event, RateeId)
CoachId -> Name, Discipline, Country, Event, RateeId
{CoachId}$^+$ = {CoachId, Name, Discipline, Country, Event, RateeId}

RateeId -> Name, Discipline, Country, Event, CoachId
{RateeId}$^+$ = {RateeId, Name, Discipline, Country, Event, CoachId}

Team (TeamId, Country, Discipline, Event, RateeId)

TeamId -> Country, Discipline, Event, RateeId
{TeamId}$^+$ = {TeamId, Country, Discipline, Event, RateeId}

RateeId : Country, Discipline, Event, TeamId
{RateeId}$^+$: {RateeId, Country, Discipline, Event, TeamId}

Country (Country, Gold, Silver, Bronze, Rank)
Country -> Gold, Silver, Bronze, Rank
{Country}$^+$ -> {Country, Gold, Silver, Bronze, Rank}

Discipline (Discipline, MaleCount, FemaleCount, TotalPlayerCount)
Discipline -> MaleCount, FemaleCount, TotalPlayerCount
{Discipline}$^+$ = {Discipline, MaleCount, FemaleCount, TotalPlayerCount}

Reasoning for normalization

Upon examination of our inferred functional dependencies, it’s evident that on each functional dependency, each left hand side attribute’s closure implies all the attributes of the relation. Consequently, it means that each left hand side attribute is a super key. Therefore, all of our schemas adhere to Boyce-Codd Normal Form (BCNF).

The reason that we choose BCNF is that it minimizes the redundancy and prevents information loss.