Project Phase 1 - Database Requirements

> *Little Timmy was 7 years old. Once, he played the best game ever - Mario Bros. - and he decided that was what he wanted to do when he grew up. However, Little Timmy's Mommy looked down upon playing video games as a profession and scolded him severely, shattering Little Timmy's dreams and his hopes for the future. Little Timmy then started concentrating on his studies and never thought about video games. Or so his mother thought.*

> *Little Timmy has been keeping track of how his favorite players are playing, what video games are being played competitively, the tournaments that are being conducted and any relevant information he might need. One day, Little Timmy will be listed on GameDhaBa. He will show his Mommy his total earnings on the database, and will prove his mother wrong. Video games do earn you money. That will be the day, Little Timmy isn't Little anymore, Little Timmy will be Rich Timmy.*

Traditional sports like Football and Cricket have a huge number of enthusiasts. At this age where everything is being digitalized, enthusiasm for eSports is rising steadily. In eSports Events, players compete individually or in teams to see who play(s) a video game the best. It is estimated that around half a billion viewers would tune in to watch at least one eSports Event during 2020. **GameDhaBa** keeps track of these video games, the teams that compete, the players who participate and the organizations that conduct the events in this growing field.

Some of the things users of this database would be able to do is to search for players with the most winnings, see the playerbase for a particular game, look up organizations that made their favorite game, find the coaches who coached the top ranking teams and many more.

Team Name - RaIS

- Rahul Goel - 2019111034
- Ishaan Shah - 2019111028
- Sriram Devata - 2019113007

Entities

- **Player -** The person who participates in video game tournaments.
    - **Player ID (Primary Key) -** Numeric, Unique, not NULL
    - **Username -** Alphanumeric String, Unique, not NULL
    - **Name (Composite Key) -**
        - **First Name -** String, First name of the person, not NULL
        - **Last Name -** String, Last name of the person, not NULL

    - **Nationality** - String, Nationality of the Player, should belong to the set of existing nationalities, not NULL
    - **Winnings -** Numeric, Total amount of money earned by the Player throughout their career, greater than or equal to zero, not NULL
    - **Number of Wins (Derived) -** Numeric, Sum of the total wins of the Team(s) the Player is a part of, greater than or equal to zero, less than or equal to number of games played by the Player's Team
    - **Number of Losses (Derived) -** Numeric, Sum of the total losses of the Team(s) the Player is a part of, greater than or equal to zero, less than or equal to number of games played by the Player's Team
    - **Date of Birth** - Date, not NULL
    - **Age (Derived) -** Numeric, greater than zero, not NULL

- **Video Game** - A video game played by Players

    - **Name** - Alphanumeric String, Unique, not NULL
    - **Game ID (Primary Key)** - Numeric, Unique, not NULL
    - **Platforms (Multivalued) -** Non-empty array of Strings
    - **Genres (Multivalued) -** Non-empty array of Strings
    - **Release Date -** Date, not NULL
    - **Latest Patch** - String, Latest stable release version, not NULL
    - **Registered Players -** Numeric, Number of registered Players of the video game

- **Organisation** - A recognized group of people that are related to the gaming community

    - **Organisation ID (Primary Key) -** Numeric, Unique, not NULL
    - **Name** - Alphanumeric String, not NULL
    - **Headquarters** - String, Must be a valid location, can be NULL
    - **Founded -** Date, Date when the organization came into being, not NULL
    - **Earnings -** Numeric, Total earnings of the Organisation, greater than or equal to zero
    - **Developer (Subclass, Partial, Disjoint) -** An Organisation that develops video games
        - **CEO** - String, Name of the CEO of the Organisation, not NULL
    - **Team (Subclass, Partial, Disjoint)** - An Organisation of Players that participates in eSports events
        - **Manager** - String, Name of the Manager of the Team, not NULL

- **eSports Event** - A tournament where Teams / Individual Players compete in video games

    - **Event ID (Primary Key) -** Numeric, Unique, not NULL
    - **Name** - Alphanumeric String, not NULL
    - **Start Date -** Date, not NULL

- **End Date** - Date, not NULL
- **Prize Pool** - Numeric, Greater than or equal to zero

- **Ranklist (Weak Entity)** - The top Teams in an eSports event ranked according to their earnings in the entire event
  - **First Place** - Alphanumeric String, not NULL
  - **Second Place** - Alphanumeric String, not NULL
  - **Third Place** - Alphanumeric String, not NULL

- **Coach (Weak Entity)** - The person who trains a Team for getting better at a video game
  - **Name** - String, The name of the coach, not NULL
  - **Start Date** - Date, Date when the Coach started training Team to play Video Game, not NULL
  - **End Date** - Date, Date when Coach stopped training Team to play Video Game, can be NULL

Relationships

- **Developed** - A 1:N (one-to-many) relationship between a Game Developer and a Video Game, participation of both the entities is complete
- **Organised** - A M:N (many-to-many) relationship between Organisations and an eSports Event, Organisation has optional participation and the event has complete participation
- **Has Result** - A 1:1 (one-to-one) identifying relationship for the entity Ranklist with an eSports Event, both the entities have total participation
- **Coaches** - A ternary identifying relationship for the Coach entity with Team and Video Game. The Coach entity has complete participation while the other two have optional participation
- **Played** - A quaternary relationship between a Player, Team, Video Game and an eSports Event. eSports event has to have total participation. All other entities has optional participation
- **Owns** - A 1:N (one-to-many) recursive relationship between a parent organisation and a subsidary
  - **Acquired on** - Date, not NULL

Functional Requirements

Insertion

- **Add Organisation** - Add a new organization to the database that falls neither in the subclass of developer nor Team. Assign an OrganisationID (incrementing counter). All non-nullable values must be entered correctly

- **Add VideoGame -** Add a new Video Game to the database. Assign a VideoGameID (incrementing counter). All non-nullable values must be entered correctly. VideoGame must be associated with an existing Organisation
- **Add Developer -** Add a new organization to the databse. Assign an OrganisationID (incrementing counter). All non-nullable values must be entered correctly
- **Add Team** - Add a organization Team to the database. Assign an OrganisationID (incrementing counter). All non-nullable values must be entered correctly. Assign a manager
- **Add eSportEvent -** Add an upcoming eSport Event to the database. Assign an EventID (incrementing counter). All non-nullable values must be entered correctly. If the event is conducted by an organization existing in the database, associate it with that organization
- **Create Ranklist** - After an eSportEvent is over, create a ranklist of the first, second and third positions and associate it with the eSportEvent
- **Add Player** - Add a Player to the database. Assign a PlayerID (incrementing counter). All non-nullable values must be entered correctly. Associate the Player with the Team, eSportEvent and the VideoGame
- **Add Coach** - Add a coach to the database. Since the coach is a weak entity that is in a ternary relationship with VideoGame and Team. The TeamID and VideoGameID together form the foreign key

Modification

- **Update VideoGameLatestPatch** - Update the LatestPatch attribute in VideoGame whenever an update to the video game is released
- **Acquirement of Organisation -** When Organisation A acquires Organisation B, build the acquirement relationship between A and B
- **Acquire Organisation -** Update the information for the organizations when one acquires another. Build the relationshipt acquires between the organizations
- **Update eSportEventStartDate** - Change the inaugral date for the event. This will happen when the event gets preponed or postponed
- **Update eSportEventEndDate -** Change the closing date for the event. This will happen when the event gets preponed or postponed
- **Update eSportEventPrizePool** - Change the winning prize for an event.

- **Update PlayerWinnings** - Update the Player Winnings. This has to be done after every event for the Players who have won.

Deletion

- **Remove eSportEvent** - When an event gets cancelled, remove it from the database and all the associations with it too.

Projections

- **eSports Events after** - Get a list of all eSports events after a particular date.

Aggregate

- **Find the youngest player's age** - Find the smallest PlayerAge.

Search

- **Search PlayerByName** - Search for a Player by his First or Last name.
- **Search OrganisationByName** - Search organisation by its registered name.

Report Generation

- **Find AllVideoGamesByDeveloper** - Returns all the VideoGames made by a certain developer.
- **Find TeamParticipations** - Returns all the Events that a Team has particpated in. Also returns the respective Players and video games.
- **Find EventTeams** - Returns all the Teams taking part in an event.
- **Find EventVideoGames** - Returns all the the video games in an event.
- **Find EventRanklist** - Returns the Event Ranklist.
- **Find PlayerTeams** - Returns all the Teams the Player is a part of.
- **Find TeamGameCoach** - Returns the coach of a particular Team for a particular game.