TEAM NAME: Confused_trio

ROLL NUMBERS: 2019101032, 2019101040, 2019101043

OUR MINI-WORLD: INDIAN PREMIER LEAGUE!!

The Mini World we have chosen is Indian Premier League. It is the richest cricket franchise league in the world. The Indian Premier League is a professional Twenty20 cricket league in India contested during March or April of every year. The IPL has been founded by the BCCI in 2008. Other than Indian players foreign players also take part in the tournament.

Thus, the organization requires a database to record data regarding different structures of the organization like teams participating, players playing and the country they belong to, the various venues they plays at, the sponsors of the League and the stages of the tournament. The databases are created in keeping view of users who wish to know the statistics of the team and the winner of the IPL Tournament (score and who won match etcetera).

DATABASE REQUIREMENTS:

- IPL consists of teams (TEAM). Each team should contain at least 20 player and may contain up to twenty-five players. We store hometown of the team (Thome, which is unique for every team), name of the team(Tname, unique for every team), a shorthand name for each team (Tteam, unique for each team Example CSK, SRH, MI).A team can be uniquely indentified either by team name(Tname) or shorthand name(Tteam).Each team will play fixed number of leagues matches and top teams will qualify for playoffs, so we store number of matches played (TPlayed), number of wins(W), and number of loses(L), and number of draws(D). Every match will be between two different teams, the number of players playing the match are 11 per side. Each team will have one and only one Captain. For every match we store number of runs scored by team and number of wickets lost, and which team won. Each team will have a unique identification number (Tid, unique for every team). Each team will have coaches coaching the team.

- Each team is sponsored by some set of sponsors (SPONSER), where sponsors are organizations. We store sponsors name (Sname, unique to every sponsor), and percentage share they have in the team (Spercentage).

- The database should store information about player (PLAYER), each player is further classified into: batsman **(BATSMAN)**, bowler (**BOWLER)** they are **subclasses** of player. Each player will play for a team. A player may captain a team. We store information about the dependents of the player. We store the name of the player (name (Firstname ,Middlename , Lastname)), date of

birth(dob) whose format is DD-MM-YYYY, and age (age, **derived attribute** from dob). We store passport number and country as an attribute(P_id, which is a **composite key attribute,** of country (Pcountry) and passport number (Passport_number)) and the number of matches they have played in this IPL (MatchesPlayed) and also addresses(Paddress, which is **multivalued attribute** since we store multiple addresses of player). For each batsman (BATSMAN) we store the number of runs he scored in this season (Bruns), and also the average runs he scored this season (Bavg, which is derived attribute). For each bowler we store the number of wickets he took this season (Bwickets).

- There are 4 stages (STAGE) in IPL they are league matches, and playoffs which are final, Qualifier, eliminator. The table toppers will qualify for playoffs and the table topper will directly enter finals, third and fourth team plays for berth in qualifier and second and the winning team in eliminator will play qualifier. Every match will have a stage associated with it. We store stage of the match (IPLstage, unique for every stage), and duration from (From, which is unique) and to (To, which is unique) note that both are in DD-MM-YYYY format.

- IPL is played across India in various venues (VENUE), each venue is home venue of some team. We store venue name for each venue (name, unique for every venue), and its capacity(capacity), and the city in which it is located (Vcity).We will analyze the type of pitch by the average of the team scores ,scored in that ground.

- The database stores dependents (DEPENDENT) for each player. We store the name of the dependent (name, which is unique for a particular player dependent partial key), age(age), relation with the player(relation). Here dependent is **weak entity**.

- Every team will have a set of coaches (COACH). Some of them would be assistant coaches and some of them would be main coaches. The assistant coach works under the guidance of main coach. We will store the name(coachName). And the direct guider (in this case assistant coach and coach relationship) of each coach. Here coach is **weak entity**.

RELATIONSHIP TYPES:

- **“CAPTAINED_BY” -** This relates teams (TEAM) with the player(PLAYER).Each team will have a Captain. The cardinality ratio of this relationship is **1:1** i.e. A team will have one and only one Captain and only one

among players captains the team. Here the participation of team is total, and player is partial.

- **“PLAYS”** -This relates a player (PLAYER) to a team (TEAM). Every player plays for a particular team. The cardinality ratio of the relationship type is **1:N** (1 on the teams side) I.e. every player belongs to a particular team and team can contains any number of players (maximum number of players in a team is 25). Here participation of both player and team is total participation.

- **“MATCH”** - Each match is played between two distinct teams team1(TEAM) and team2(TEAM) at a particular venue (VENUE), matches are played at different stages (STAGE) of IPL namely league matches, qualifiers, eliminators, final. This is **n>3 relationship type**. We store the match number (mnumber, which is unique for every match), TeamA score (TeamAscore), TeamB score (TeamBscore) where TeamA and TeamB are the teams which are playing in this match. We also store the team which won (WinTeam). Example: If fixture is RCB vs CSK then TeamA is taken to be RCB and TeamB is CSK TeamAscore 239, TeamBscore 240 so TeamB won. Also note that team which is batting first is assigned to be TeamA. Note that winning team (WinTeam) is derived attribute. Participation of both teams and venue and stage is total participation. Let us denote the cardinality constraints in min max notation. For calculation purposes let us consider ‘T' to be number of teams. So number of stages will be min 1 (which occurs when stage is final, qualifier, eliminator) max (T-1) *T (league matches) . Number of venues min T-1 (which occurs when only home matches are played that too only in league matches), max T (league matches + one playoff match considering that no two playoff matches are played at same venue). And cardinality number for teams will be min 2*(T-1) (These many matches will be played by a team which didn’t get into playoffs) and max 2*(T-1)+3 (league matches + eliminator + qualifier + final).

- **“DEPENDENTS_OF”**: This relates the dependent (DEPENDENT) of the players (PLAYER) to the players on which they are depending. This is the identification relationship type for the weak entity dependents. Every dependent will be related with a player. The cardinality ratio is **1:N** where 1 is for dependent side and N is for player side. The dependents will have total participation ship in this relationship. Player will have partial relationship constraint in this relationship type.

- **“COACHES”:** This relates the team (TEAM) with their coaches (COACH). This is the identification relationship type for the weak entity Coach. Every team will have a coach. Every coach will be related to Team. So, this relationship will have a mutual total participation in this relationship. A team can have more than

one coaches. But a coach can be related to only one Team. So the cardinality relation is **1:N**.(Here 1 is for the coach side and N is for team side).

- **"SPONSORS":** This relates a team (TEAM) with the sponsor (SPONSOR). A Sponsor sponsors any number of teams. A team can have more than one sponsor. This relationship type has total participation on both sides. This have the cardinality ratio of **M:N**. Example JIO sponsors for all team and each team have at least one sponsor.

- **"UNDER_GUIDANCE"**: This is **self-relationship** is between coaches (COACH), who are taking the guidance, to the coaches who are giving the guidance. Both are partial participation. Cardinality ratio is **1:N** where 1 is on main coach side and N is assistant coach.

FUNCTIONALITIES:

Retrieval:

QUERY FUNCTIONS:

1) **Selection:** The select function for all the entities which lists the contents of the database. We can select the players participating the current IPL or to see the total fixtures in this IPL, to name few.
2) **Projection:** The function whose functionality is to retrieve players who scored greater than a particular number of runs(provided as input).OPTIONAL(Or the players whose batting average is above particular number(provided as the input) Or the teams which won more than particular number of Matches. Or the matches held at (say Mumbai) venue to name few.)
3) **Aggregate:** This function displays the average number of the runs scored by team/player in all matches. OPTIONAL (Maximum score among all matches in the current season. Maximum number of runs scored by a player this year, or maximum wickets taken by a bowler.)
4) **Search:** A search facility, to search for a player/sponsor by searching his/its name or part of name.

Analysis:

1. We can display the leading scorer from a particular team. And leading wicket taker from a team. In both case we are exploiting "PLAYS" relationship type between player and team.

2. Average score batting first at a particular type. And we can say whether it is bowling or batting pitch (say score<120 bowling pitch).

MODIFICATION:

Insert: -

1. We insert the details of a match after each match is played. And we insert the players/sponsor/coach of the current season. This will be inserted keeping in view of integrity and other kinds of constraints.

Update: -

1. After every game, the numbers of wins and losses of a team is updated, and also the number of runs ,wickets, average runs of a player are updated.

Delete: -

1. When a player gets injured or dead (so he cannot take part further in the tournament) then we delete player.