# IPL – INDIAN PRIMIERE LEAGUE DB PHASE - III

**Group Name: Confused_trio**

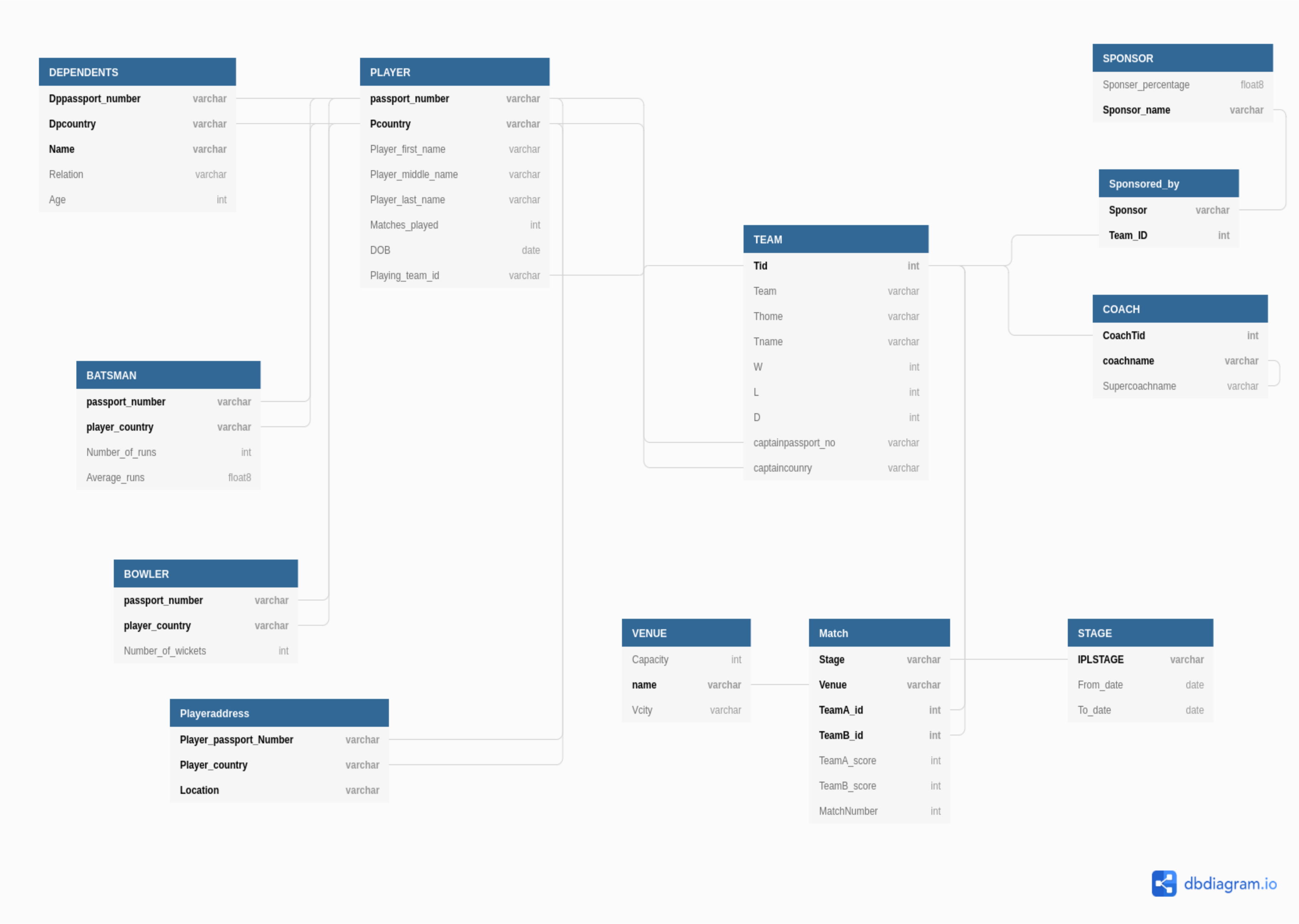
**Roll No: 2019101040, 2019101043, 2019101032**

## Explanation:

### ER TO RELATIONAL MODEL:

- We followed the steps given in textbook for converting into the relation model. The following are the details that are done while converting ER to relational model, NOTE that the names of some of the attributes are changed in the relational model to enhance the ease of understanding of the properties by its names, few of them are mentioned in the explaination below:
- When converting TEAM entity type to TEAM relation, Team, THome, Tname, captainpassport_no, captaincountry , Tid are candidate keys among we chose Tid as the primary key among many other which are there for the TEAM entity type. Derived attribute Tplayed which is there in ER model is omitted (This will be calculated at application level from L D W which are attributes in the TEAM relation). Two attributes namely captainpassport_no and captaincountry are added to TEAM relation to satisfy CAPTAINED _BY relationship type in ER model, so these acts as foreign key in the TEAM relation.
- PLAYER in ER model is converted to PLAYER relation. We omitted Paddresses since it is a multivalued attribute. And the key P_id is divided into simpler attributes passport_number and Pcountry combination of which is primary key of the table. Age being derived attribute is also omitted from this relation. The name of the player being a composite attribute in the ER model is further divided into Player_first_name, Player_middle_name, Player_last_name to ensure that only simple attributes are part of the relation. Rest of the attributes are added without any change.
- The subclasses of the PLAYER i.e. BATSMAN and BOWLER are converted into BATSMAN and BOWLER relations respectively. The primary key of the player is the primary key in the both the subclasses (passport_number, Pcountry is the primary key of the subclasses), We also added Number_of_runs, Average_runs attributes to the BATSMAN and Number_of_wickets to the BOWLER relation.

- VENUE entity type is converted to VENUE relation in which name (Name) acts as primary key and remaining attributes capacity and vcity are added without any change as they are simple attribute in relation.
- STAGE entity type is converted to STAGE relation. The name of the stage in the IPL acts as a primary key(IPLstage). The composite attribute Duration is converted into simple attributes From_date and To _date.
- COACH weak entity type depends upon TEAM entity. The COACH entity type is converted into COACH relation in which the Tid of the Team, which the coach trains, the name of the coach(coachname) and the Tid acts as a primary key which is inherited from the TEAM relation. We also store the name of the supervisor coach (Supercoachname) under the relation Under_Guidance.
- We change the SPONSOR entity into SPONSOR relation. Since every attribute is a simple attribute in the SPONSOR entity, we have included them without any change. Here the Sponsor_name acts as a primary key of the relation.
- The DEPENDENT entity is a weak entity, and the PLAYER entity is the identifying entity of DEPENDENT entity. So, we convert the DEPENDENT entity into DEPENDENT relation. The primary key of the relation is combination Name of the dependent (Name), passportnumber of the player(Dppassport_number), country of the player(Dpcountry). We store the remaining simple attributes without any change.
- Since the SPONSORED_BY is a M:N relation we have created a new relation with the participating entities primary keys as the foreign keys of the relation(Sponsor, Teamid). The participating entities are TEAM relation and the SPONSOR relation. The combination of both foreign keys acts as primary keys. The SPONSORED_BY relation doesn't have any attributes
- The MATCH relation is n-ary relation with 4 participating entities. None of the participating entities have the cardinality as 1. Hence, we have created a new relation MATCH with the participating entities primary keys as the foreign keys of the relation. The foreign key Venue points to the primary key(name) of the VENUE relation, the foreign key TeamA_id points to the primary key(Tid) of the Team relation,the foreign key TeamB_id points to the primary key(Tid) of the team relation. We include the simple attributes of the entity relation MATCH without any change. The combinatin of four foreign keys act as the primary key of the relation.
- Since the player has a multiple valued attribute address, we create a new table named ADDRESS to store the addresses of the player. The relation includes passport number of the player (Player_passport_number) , player country(Player_country) and the location.



| DEPENDENTS | |
|---|---|
| Dppassport_number | varchar |
| Dpcountry | varchar |
| Name | varchar |
| Relation | varchar |
| Age | int |

| PLAYER | |
|---|---|
| passport_number | varchar |
| Pcountry | varchar |
| Player_first_name | varchar |
| Player_middle_name | varchar |
| Player_last_name | varchar |
| Matches_played | int |
| DOB | date |
| Playing_team_id | varchar |

| SPONSOR | |
|---|---|
| Sponser_percentage | float8 |
| Sponsor_name | varchar |

| Sponsored_by | |
|---|---|
| Sponsor | varchar |
| Team_ID | int |

| TEAM | |
|---|---|
| Tid | int |
| Team | varchar |
| Thome | varchar |
| Tname | varchar |
| W | int |
| L | int |
| D | int |
| captainpassport_no | varchar |
| captaincounry | varchar |

| COACH | |
|---|---|
| CoachTid | int |
| coachname | varchar |
| Supercoachname | varchar |

| BATSMAN | |
|---|---|
| passport_number | varchar |
| player_country | varchar |
| Number_of_runs | int |
| Average_runs | float8 |

| BOWLER | |
|---|---|
| passport_number | varchar |
| player_country | varchar |
| Number_of_wickets | int |

| VENUE | |
|---|---|
| Capacity | int |
| name | varchar |
| Vcity | varchar |

| Match | |
|---|---|
| Stage | varchar |
| Venue | varchar |
| TeamA_id | int |
| TeamB_id | int |
| TeamA_score | int |
| TeamB_score | int |
| MatchNumber | int |

| STAGE | |
|---|---|
| IPLSTAGE | varchar |
| From_date | date |
| To_date | date |

| Playeraddress | |
|---|---|
| Player_passport_Number | varchar |
| Player_country | varchar |
| Location | varchar |

## RELATIONAL TO FIRST NORMAL FORM:

- There are no changes in converting our relational model from relational to first normal form. All the multivalued attributes are already represented using a separate relation. And all the composite attributes are also separated into simple attributes, so there are no changes in converting g from relational to first normal from. To be specific we seperated addresses of the players and represented it using a relation Playeraddress, in our database this is the only multivalued attrributes hence we done with this part. Coming to composite attributes like Name in PLAYER relation and Duration in IPL stage so on are divided in to Player_first_name, Player_middle_name, Player_last_name and From_date, To _date respectively.

## FIRST NORMAL FORM TO SECOND NORMAL FORM:

- There are no changes in going from first to second normal from, all the relations satisfy the condition that every non-prime key attribute is fully dependent on the candidate key. To name few consider TEAM relation in which Tid, Team, Thome, Tname,(captainpassport_no, captaincounry) are the candidate keys in this relation. Consider non – prime attributes W,L,D which are fully dependent on the candidate key, this is fairly simple in case of Tid, Team, Thome, Tname as the number of attributes in these candidate keys are only one, So it is bound to be fully dependent. Coming to (captainpassport_no, captaincounry) removing any of them will fail to identify the team so it is impossible for us to say the number of matches L,W,D therefore if is also fully dependent on the (captainpassport_no, captaincounry). Hence TEAM is in second NORMAL from.

## SECOND NORMAL TO THIRD NORMAL FORM:

- The current model is free from Transitive function dependencies. No non-prime attribute determines a non-prime attribute equivalently there are no functional dependency Y depending on X such that X is not super key and Y is not a prime attribute. For example, consider the relation Team in which none of the non-prime attributes(TeamA_score, TeamB_score) can’t determine another non-prime attributes .The remaining attributes area prime attributes. Hence it is in third normal form.