CSE2004: DATABASE MANAGEMENT SYSTEMS

FACULTY: SARAVANAKUMAR K

THEORY SLOT: B1+TB1

LAB SLOT: L53 +L54

FALL SEMESTER-2021-22

DIGITAL ASSIGNMENT

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

BY

RISHIT ANAND (20BCE2146)

B-TECH

COMPUTER SCIENCE AND ENGINEERING

DATED AS 18th October 2021

IPL STATISTICS MANAGEMENT SYSTEM

REQUIREMENT ANALYSIS

We shall require the necessary data to manage the statistics of all the IPL seasons along with updation of database for the statistics of each individual player participating in IPL.

We would need data on Season, Matches, Players, Teams, Umpires, Venue, Stadiums, run scored, Balls played, overs played and all necessary statistical data of each player.

INTRODUCTION

As currently the IPL season is going on, I have decided to form an efficient database for the IPL statistics management.

My database would represent the statistics of each match in terms of the player's scores in each team, each match statistics, post-match player scores, winners, season info, etc.

It shall also represent the overall statistics of each player as updation shall take place after each match.

Each match info shall include attributes like total runs by both teams, super over details, umpire details, etc which in turn be connected to entity sets like PLAYER, TEAM, UMPIRE, BALL_BY_BALL, VENUE etc for efficient management of the DATA.

DATA-COLLECTION

- Details of each team participating in the IPL
- Details on each player in each team in IPL
- Details of the umpires in the IPL
- Statistics of each batsman(player) participating in the IPL
- Statistics of each bowler(player) participating in the IPL
- Details on each Match having the date and the timings
- Statistical Details of each match like duration of each innings, runs made by each team, etc.
- Ball by Ball details for each match.
- Post-match statistics of all the batsman of the concerned match.
- Post-match statistics of all the bowler of the concerned match.

- Details on each Season of the IPL
- Details on the Concerned venues in IPL

ENTITY SETS

- **TEAM entity set:** A strong entity set having the details of each team participating in the IPL.

  TEAM (**T_Id**, T_Name, No_of_players, Coach_Id, C_Name)

- **PLAYER entity set:** A strong entity set having the details of each player participating in the IPL.

  PLAYER (**P_Id,** P_Name, P_Age, P_Country, P_Designation, Matches_played, Batting_Skill, Bowling_Skill)

- **Two ROLES have been identified for the PLAYER entity set.**
  1. CAPTAIN
  2. WICKETKEEPER

- **UMPIRE entity set:** Strong entity set having the details of the umpires in the IPL.

  UMPIRE (**U_Id**, U_Name, DOB, Experience)

- **STATS_OF_BATSMAN entity set:** A weak entity set having the statistics of each batsman(player) participating in the IPL

  STATS_OF_BATSMAN (Total_Runs, Avg_Run_Rate, Innings_played)

- **STATS_OF_BOWLER entity set:** A weak entity set having the statistics of each bowler(player) participating in the IPL

  STATS_OF_BOWLER (Overs_played, Wickets_taken, Catches)

- **MATCH_INFO entity set**: A strong entity set having the details on each Match having the date and the timings.

  MATCH_INFO (**M_id**, City_Name, M_Date, Start_Time, End_Time)

- **BALL_BY_BALL entity set**: A weak entity set having the Ball by Ball details for each match.

  BALL_BY_BALL (**B_Id, Over_Id**, B_Num, Runs, Out, Catch, Run_out)

- **MATCH_STATS entity set**: A weak entity set having the statistical Details of each match like duration of each innings, runs made by each team, etc.

  MATCH_STATS(Toss_Winner, Toss_Decision, Innings_Duration1, Innings_Duration2, Runs_By_T1, Runs_By_T2, Sixes, Fours, Super_Over, Man_of_match, Winner, Won_By)

- **POST_MATCH_BATSMAN_STATS entity set**: A weak entity set having post-match statistics of all the batsman of the concerned match

  POST_MATCH_BATSMAN_STATS(T_Runs, Run_rate, No_Sixes, No_Fours, Out_At_Run)

- **POST_MATCH_BOWLER_STATS entity set**: A weak entity set having post-match statistics of all the bowler of the concerned match

  POST_MATCH_BOWLER_STATS(No_Overs_played, No_Catches, Runouts, No_Wickets_taken)

- **SEASON entity set**: A strong entity set having the details on each Season of the IPL

  SEASON(**Season_Id**, S_Year, S_Country, Orange_Cap_Id, Purple_Cap_Id, Man_of_series_Id)

- **VENUE entity set**: a strong entity set having the details on the Concerned venues in IPL

  VENUE(**V_id**, Stadium_name, Capacity, Eatery_Arrangment)

RELATIONSHIP SETS

1. **P_T** -> Many to one strong relationship from team to player

2. **I_S** -> Many to one strong relationship from Match_Info to Season

3. **I_V** -> Many to one relationship from Match_Info to Venue

4. **I_U** -> Many to many relationship from Match_Info to Umpire

5. **I_T** -> Many to many relationship from Match_Info to Team

6. **Info_Stats** -> One to one relationship from Match_Info to Match_Stats

7. **I_B** -> Many to one relationship from ball to match_info

8. **BATSMAN_PLAYER** -> One to one relationship between the Batsman stats and the player.

9. **BOWLER_PLAYER** -> One to one relationship between the Bowler stats and the player.

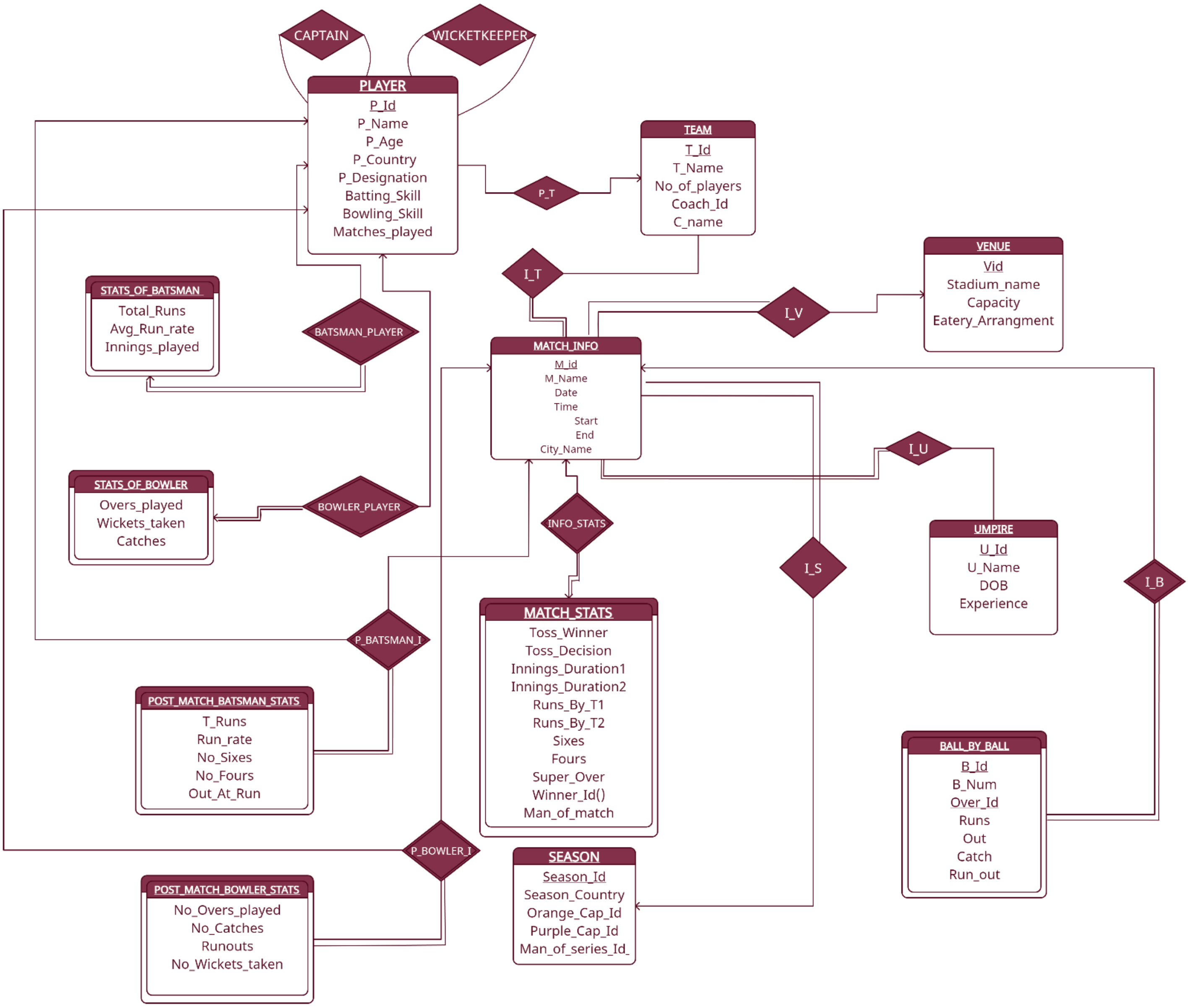
10. **CAPTAIN** -> Role for the captain and the player

11. **WICKETKEEPER** -> Role for the wicketkeeper and the player

12. **P_BATSMAN_I** -> Ternary relationship between Player, Match_Info and the Post_Match_Batsman_Stats

13. **P_BOWLER_I** -> Ternary relationship between Player, Match_Info and the Post_Match_Bowler_Stats

ER DIAGRAM:

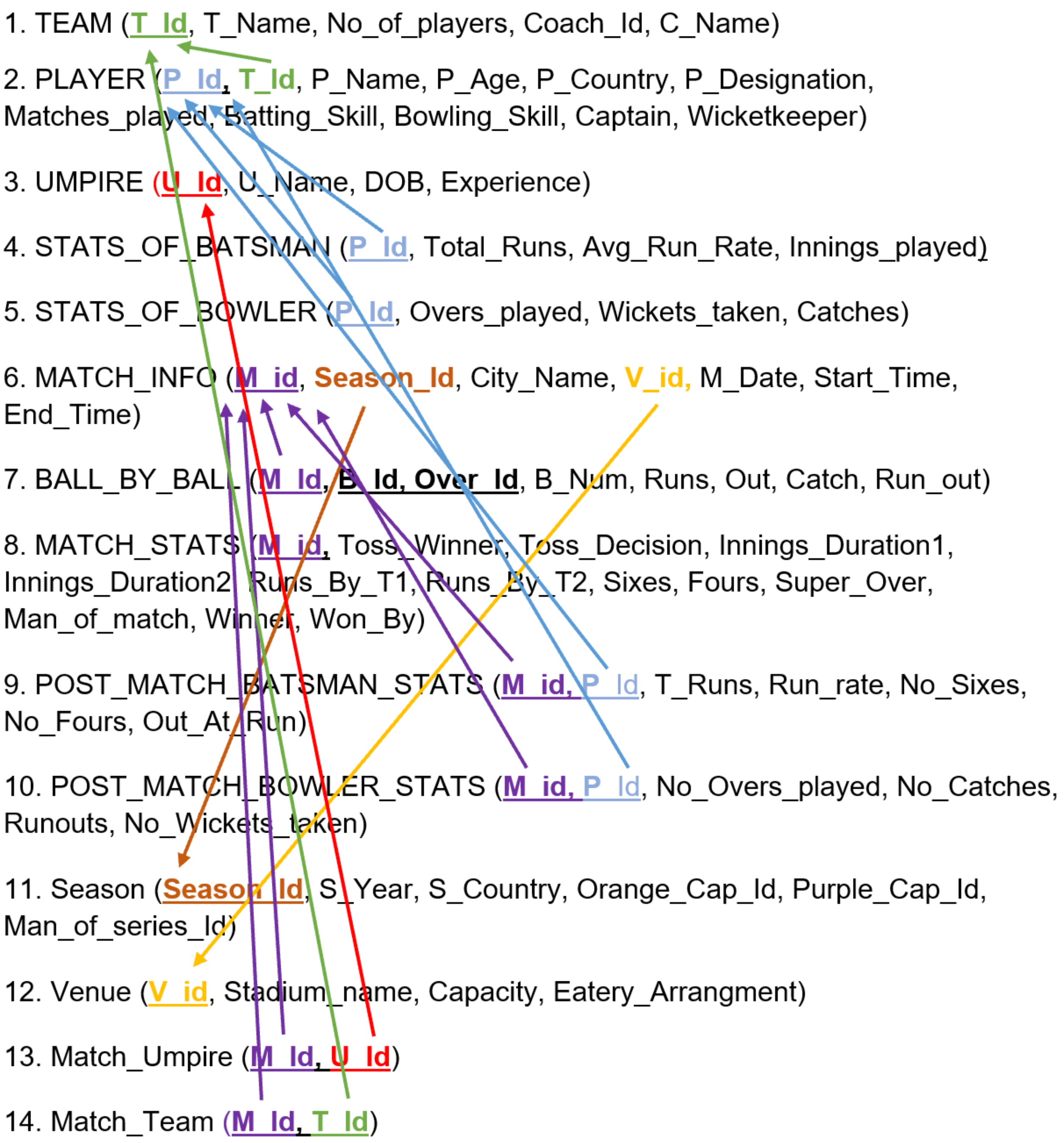


Rules to Convert ER diagram to Schema:

1. Create table for the strong entities
2. Create a table for the weak entity set, include the Primary key of its dependent relationship via the identifying relationship set
3. For Composite attributes, create a separate relation
4. For Multivalued attributes also create a separate relation
5. Don’t mention the derived attributes in the entity sets
6. For 1-1 relationship, add the primary key of any one side onto the other, but favour total participation

7. For 1-n and n-1, add the primary key of the one side to the many side.
8. For m-n relationship, create a separate table containing the primary key of both entity sets.
9. For Roles, put the roles as attributes in the related entity set.

SCHEMA:



1. TEAM (T_Id, T_Name, No_of_players, Coach_Id, C_Name)

2. PLAYER (P_Id, T_Id, P_Name, P_Age, P_Country, P_Designation, Matches_played, Batting_Skill, Bowling_Skill, Captain, Wicketkeeper)

3. UMPIRE (U_Id, U_Name, DOB, Experience)

4. STATS_OF_BATSMAN (P_Id, Total_Runs, Avg_Run_Rate, Innings_played)

5. STATS_OF_BOWLER (P_Id, Overs_played, Wickets_taken, Catches)

6. MATCH_INFO (M_id, Season_Id, City_Name, V_id, M_Date, Start_Time, End_Time)

7. BALL_BY_BALL (M_Id, B_Id, Over_Id, B_Num, Runs, Out, Catch, Run_out)

8. MATCH_STATS (M_id, Toss_Winner, Toss_Decision, Innings_Duration1, Innings_Duration2, Runs_By_T1, Runs_By_T2, Sixes, Fours, Super_Over, Man_of_match, Winner, Won_By)

9. POST_MATCH_BATSMAN_STATS (M_id, P_Id, T_Runs, Run_rate, No_Sixes, No_Fours, Out_At_Run)

10. POST_MATCH_BOWLER_STATS (M_id, P_Id, No_Overs_played, No_Catches, Runouts, No_Wickets_taken)

11. Season (Season_Id, S_Year, S_Country, Orange_Cap_Id, Purple_Cap_Id, Man_of_series_Id)

12. Venue (V_id, Stadium_name, Capacity, Eatery_Arrangment)

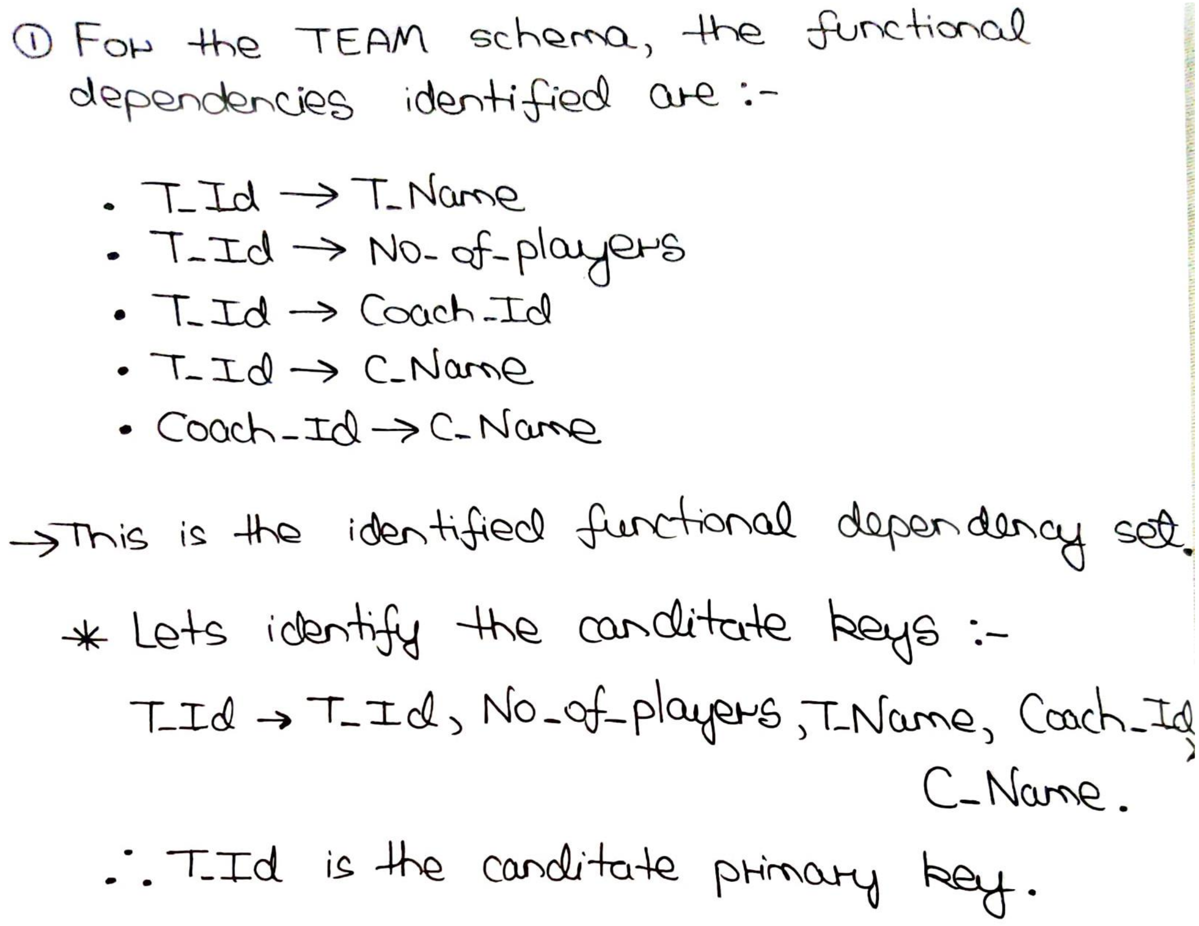
13. Match_Umpire (M_Id, U_Id)

14. Match_Team (M_Id, T_Id)

PHASE 2: NORMALIZATION

1. TEAM entity set:

T_Id	T_Name	No_of_players	Coach_Id	C_Name
T1	Kolkata Knight Riders	14	C1	Smith
T2	Royal Challengers Bangalore	14	C2	Rahul
T3	Chennai Super Kings	20	C3	Adam
T4	Kings XI Punjab	16	C4	Dravid
T5	Rajasthan Royals	13	C5	Sanjay
T6	Delhi Capitals	15	C6	Allen



① For the TEAM schema, the functional dependencies identified are :-

- T_Id → T_Name
- T_Id → No-of-players
- T_Id → Coach-Id
- T_Id → C_Name
- Coach-Id → C-Name

→This is the identified functional dependency set.

* Lets identify the canditate keys :-

T_Id → T_Id, No_of_players, T_Name, Coach_Id, C_Name.

∴ T_Id is the canditate primary key.

Checking for 1NF:

A relation schema is said to be in 1NF if and only if:

- All the attributes of the relation are atomic (indivisible into meaningful subparts)
- Every attribute contains single value (per record).

The table is already in 1NF as all the domains are atomic

Checking for 2NF:

A relation schema is in 2NF if and only if:

- At the first place the table is in 1NF,
- All the non-key attributes of the table are fully functionally dependent on the Primary key of the table.

The table is already in 2NF as there is no partial dependency

Checking for 3NF:

A relation schema is in 3NF if and only if:

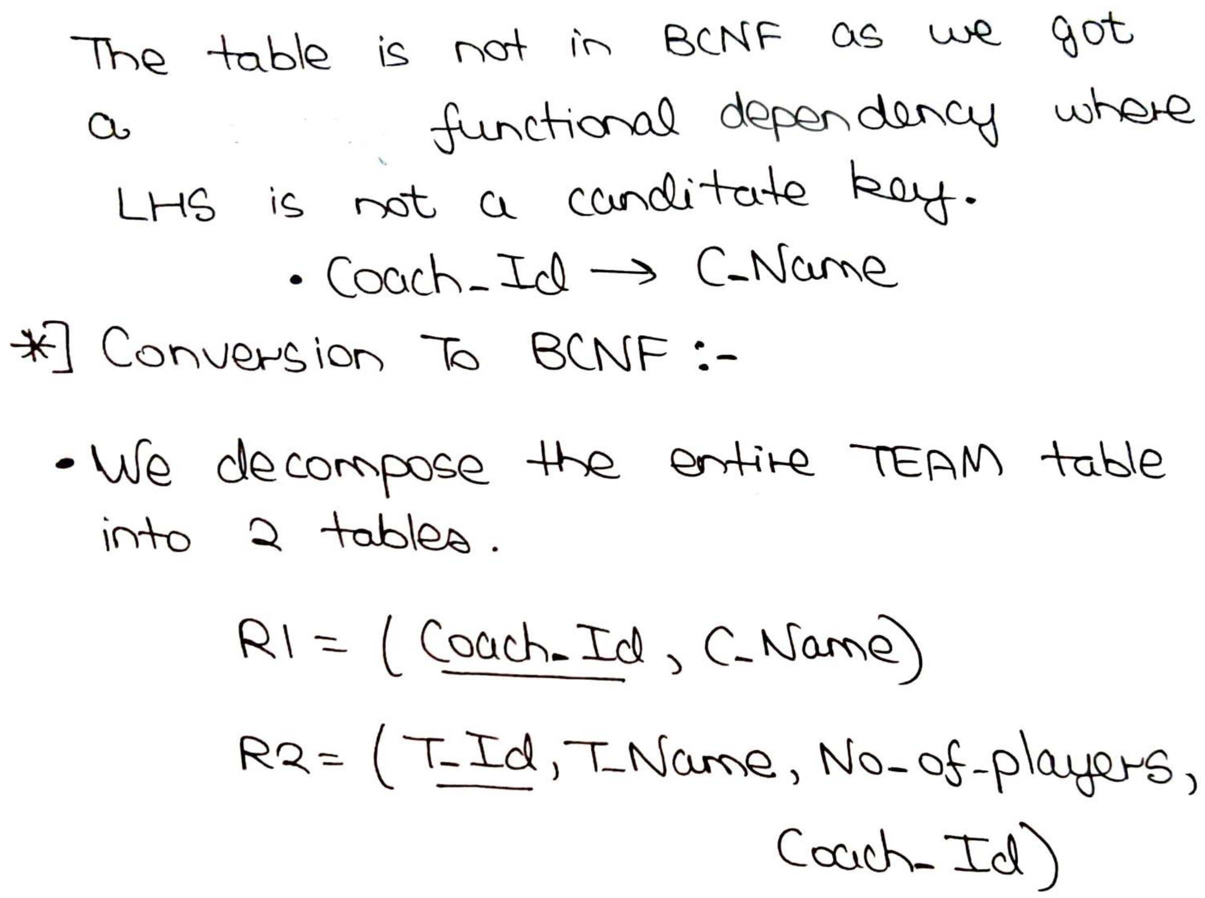
- The table is in 2NF,
- There is no Functional Dependency such that both Left Hand Side and Right-Hand Side attributes of the FD are non-key attributes. In other words, no transitive dependency is allowed.

The table is in 3NF as there is no transitive dependency.

Checking for BCNF:

A relation schema is in BCNF if and only if:

- The table is in 3NF
- For all the non-trivial FDs held on R, the left-hand side of those non-trivial FDs must be Candidate Keys



The table is not in BCNF as we got a functional dependency where LHS is not a canditate key.

- Coach_Id → C_Name

*] Conversion To BCNF :-

- We decompose the entire TEAM table into 2 tables.

R1 = (Coach_Id, C_Name)

R2 = (T_Id, T_Name, No-of-players, Coach_Id)

2. PLAYER entity set:

P_Id	T_Id	P_Name	P_Age	P_Country	P_Designation	Matches_played	Batting_Skill	Bowling_Skill	Captain	Wicketkeeper
CSK1	T3	MS Dhoni	23	India	Batsman	280	Left_Hand	Right-arm medium	C1	W1
CSK2	T3	Suresh Raina	21	India	Bowler	209	NULL	Right-arm medium	NULL	NULL
CSK3	T3	Kedar Jadhav	39	India	All-Rounder	298	Right_Hand	Right-arm medium	NULL	NULL
CSK4	T3	Ravindra Jadeja	26	India	Batsman	294	Right_Hand	Right-arm offbreak	NULL	NULL
CSK5	T3	Dwayne Bravo	31	West Indies	Bowler	202	NULL	Right-arm offbreak	NULL	NULL

② We take the PLAYER schema for normalization.

→ PLAYER schema would have the following functional dependencies :-

- P_Id → T_Id
- P_Id → P_Name
- P_Id → P_Age
- P_Id → P_Country
- P_Id → P_Designation
- P_Id → Matches_Played
- P_Id → Batting_Skill
- P_Id → Bowling_Skill
- P_Id → Captain
- P_Id → Wicketkeeper
- Captain → P_Id
- Wicketkeeper → P_Id

* We have found the possible FD set for the PLAYER schema.

* Now, we shall loo k for the possible canditate keys.

① P_Id ② Captain ③ Wicketkeeper

→ These are the possible candidate keys.

Checking for 1NF:

A relation schema is said to be in 1NF if and only if:

- All the attributes of the relation are atomic (indivisible into meaningful subparts)
- Every attribute contains single value (per record).

The table is already in 1NF as all the domains are atomic

Checking for 2NF:

A relation schema is in 2NF if and only if:

- At the first place the table is in 1NF,
- All the non-key attributes of the table are fully functionally dependent on the Primary key of the table.

The table is already in 2NF as there is no partial dependency

Checking for 3NF:

A relation schema is in 3NF if and only if:

- The table is in 2NF,
- There is no Functional Dependency such that both Left Hand Side and Right-Hand Side attributes of the FD are non-key attributes. In other words, no transitive dependency is allowed.

The table is in 3NF as there is no transitive dependency.

Checking for BCNF:

A relation schema is in BCNF if and only if:

- The table is in 3NF
- For all the non-trivial FDs held on R, the left-hand side of those non-trivial FDs must be Candidate Keys

The table is in BCNF

FINAL:

1. TEAM (**T_Id**, T_Name, No_of_players, **Coach_Id**)

2. COACH (**Coach_Id**, C_Name)

3. PLAYER (**P_Id, T_Id**, P_Name, P_Age, P_Country, P_Designation, Matches_played, Batting_Skill, Bowling_Skill, Captain, Wicketkeeper)

4. UMPIRE (**U_Id**, U_Name, DOB, Experience)

5. STATS_OF_BATSMAN (**P_Id**, Total_Runs, Avg_Run_Rate, Innings_played)

6. STATS_OF_BOWLER (**P_Id**, Overs_played, Wickets_taken, Catches)

7. MATCH_INFO (**M_id**, **Season_Id**, City_Name, **V_id,** M_Date, Start_Time, End_Time)

8. BALL_BY_BALL (**M_Id, B_Id, Over_Id**, B_Num, Runs, Out, Catch, Run_out)

9. MATCH_STATS (**M_id,** Toss_Winner, Toss_Decision, Innings_Duration1, Innings_Duration2, Runs_By_T1, Runs_By_T2, Sixes, Fours, Super_Over, Man_of_match, Winner, Won_By)

10. POST_MATCH_BATSMAN_STATS (**M_id, P**_Id, T_Runs, Run_rate, No_Sixes, No_Fours, Out_At_Run)

11. POST_MATCH_BOWLER_STATS (**M_id, P**_Id, No_Overs_played, No_Catches, Runouts, No_Wickets_taken)

12. Season (**Season_Id**, S_Year, S_Country, Orange_Cap_Id, Purple_Cap_Id, Man_of_series_Id)

13. Venue (**V_id**, Stadium_name, Capacity, Eatery_Arrangment)

14. Match_Umpire (**M_Id, U_Id**)

15. Match_Team (**M_Id, T_Id**)