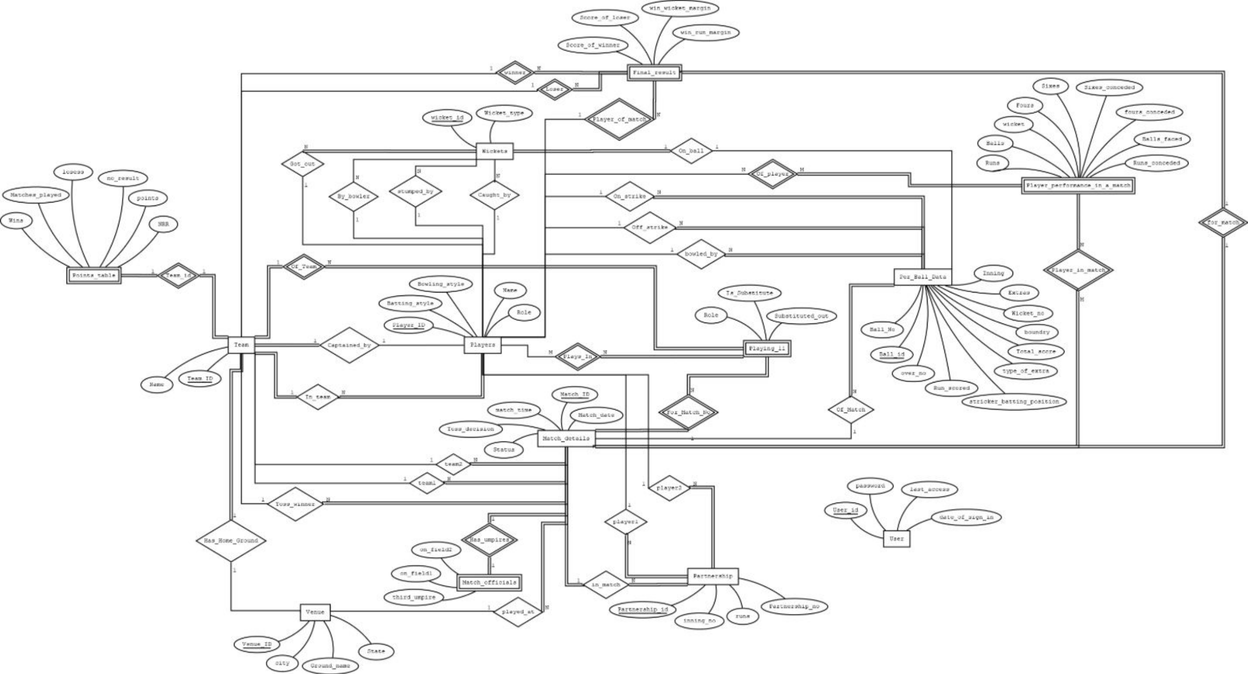
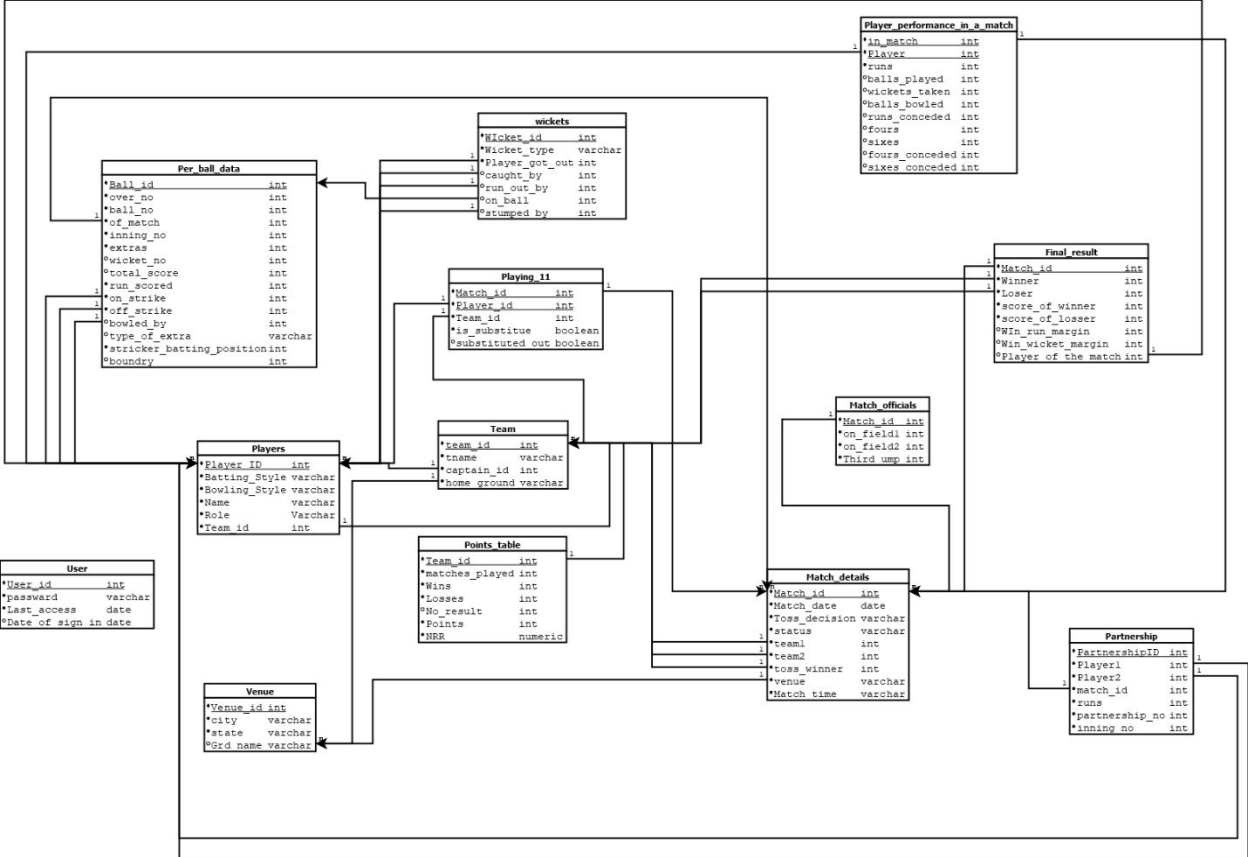


Score_of_loser
win_wicket_margin
win_run_margin
Score_of_winner
winner
Final_result
Loser
Player_of_match
wicket_id
Wicket_type
Wickets
On_ball
Of_player
Sixes
Sixes_conceded
Fours
fours_conceded
wicket
Balls_faced
Balls
Runs
Runs_conceded
Player_performance_in_a_match
for_match
Got_out
By_bowler
stumped_by
Caught_by
On_strike
Off_strike
bowled_by
Player_in_match
losses
no_result
Matches_played
points
Wins
NRR
Points_table
Team_id
Of_Team
Per_Ball_Data
Inning
Extras
Wicket_no
boundry
Total_score
type_of_extra
stricker_batting_position
Run_scored
over_no
Ball_id
Ball_No
Bowling_style
Name
Batting_style
Role
Player_ID
Is_Substitute
Role
Substituted_out
Team
Captained_by
Players
Plays_in
Playing_11
Name
Team_ID
In_team
for_Match_No
Of_Match
Match_ID
match_time
Match_date
Toss_decision
Status
Match_details
team2
team1
player2
Toss_winner
player1
password
last_access
User_id
date_of_sign_in
User
Has_Home_Ground
Has_umpires
on_field2
on_field1
Match_officials
third_umpire
in_match
Partnership
Venue
played_at
Partnership_id
inning_no
runs
Partnership_no
Venue_ID
city
Ground_name
State


Player_performance_in_a_match
*in_match int
*Player int
*runs int
°balls_played int
°wickets_taken int
°balls_bowled int
°runs_conceded int
°fours int
°sixes int
°fours_conceded int
°sixes_conceded int
wickets
*WIcket_id int
*Wicket_type varchar
*Player_got_out int
°caught_by int
°run_out_by int
°on_ball int
°stumped by int
Per_ball_data
*Ball_id int
*over_no int
*ball_no int
*of_match int
*inning_no int
*extras int
°wicket_no int
°total_score int
*run_scored int
*on_strike int
*off_strike int
°bowled_by int
°type_of_extra varchar
*stricker_batting_position int
°boundry int
Playing_11
*Match_id int
*Player_id int
*Team_id int
*is_substitue boolean
°substituted out boolean
Final_result
*Match_id int
*Winner int
*Loser int
*score_of_winner int
*score_of_losser int
°WIn_run_margin int
°Win_wicket_margin int
°Player of the match int
Match_officials
*Match_id int
*on_field1 int
*on_field2 int
*Third ump int
Team
*team_id int
*tname varchar
*captain_id int
*home_ground varchar
Players
*Player_ID int
*Batting_Style varchar
*Bowling_Style varchar
*Name varchar
*Role Varchar
*Team_id int
Points_table
*Team_id int
*matches_played int
*Wins int
*Losses int
°No_result int
*Points int
*NRR numeric
User
*User_id int
*password varchar
*Last_access date
°Date of sign in date
Match_details
*Match_id int
*Match_date date
*Toss_decision varchar
*status varchar
*team1 int
*team2 int
*toss_winner int
*venue varchar
*Match_time varchar
Partnership
*PartnershipID int
*Player1 int
*Player2 int
*match_id int
*runs int
*partnership_no int
*inning_no int
Venue
*Venue_id int
*city varchar
*state varchar
°Grd_name varchar

1. Players Relation

Given FDs:
player_id → {batting_style, bowling_style, name, role, team_Id}

Compute the Closure of Player_ID :
player_id+ = { player_id, batting_style, bowling_style, name, role, team_id }

Since player_id+ includes all attributes of the relation, player_id is a Key.

For all functional dependencies in this relation, the left side is player_id, which is a Key.
Hence, the Players relation satisfies BCNF.

2. Team Relation

Given FDs:
team_id → {tname, caption_id, home_ground}

Compute the Closure of team_id :
team_id+ = { team_id, tname, caption_id, home_ground }

Since team_id+ includes all attributes of the relation, team_id is a Key.

For all functional dependencies in this relation, the left side is team_id, which is a Key.
Hence, the Team relation satisfies BCNF.

3. Venue Relation

Given FDs:
venue_id → {city, state, grd_name}

Compute the Closure of Venue_id :
venue_id+ = { venue_id, city, state, grd_name }

Since venue_id+ includes all attributes of the relation, venue_id is a Key.

For all functional dependencies in this relation, the left side is venue_id, which is a Key.
Hence, the Venue relation satisfies BCNF.

4. Match_details Relation

Given FDs:
match_id → {match_date, toss_decision, status, team1, team2, toss_winner, venue, match_time}

Compute the Closure of Match_id :
match_id+ = { match_id, match_date, toss_decision, status, team1, team2, toss_winner, venue, match_time }

Since match_id+ includes all attributes of the relation, match_id is a Key.

For all functional dependencies in this relation, the left side is match_id, which is a Key.
Hence, the Match_details relation satisfies BCNF.

5. Playing_11 Relation

Given FDs:
{ match_id, player_id } → team_id
{ match_id, player_id } → is_substitute
{ match_id, player_id } → substituted_out

Compute the Closure of { match_id, player_id } :
{ match_id, player_id }+ = { match_id, player_id, team_id, is_substitute, substituted_out }

Since { match_id, player_id }+ includes all attributes of the relation, { match_id, player_id } is a Key.

For all functional dependencies in this relation, the left side is { match_id, player_id }, which is a Key.
Hence, the Playing_11 relation satisfies BCNF.

6. Match_officials Relation

Given FDs:
match_id → {on_field1, on_field2, third_ump}

Compute the Closure of match_id :
match_id+ = { match_id, on_field1, on_field2, third_ump }

Since match_id+ includes all attributes of the relation, match_id is a Key.

For all functional dependencies in this relation, the left side is match_id, which is a Key.
Hence, the Match_officials relation satisfies BCNF.

7. Partnership Relation

Given FDs:
partnership_id → {player1, player2, match_id, runs, partnership_no, inning_no}

Compute the Closure of partnership_id :
partnership_id+ = { partnership_id, player1, player2, match_id, runs, partnership_no, inning_no }

Since partnership_id+ includes all attributes of the relation, partnership_id is a Key.

For all functional dependencies in this relation, the left side is partnership_id, which is a Key.
Hence, the Partnership relation satisfies BCNF.

8. Per_ball_data Relation

Given FDs:
ball_id → {over_no, ball_no, of_match, inning, extras, wicket_no, total_score, runs_scored, on_strike, off_strike, bowled_by, stricker_batting_position, type_of_extra, boundry}

Compute the Closure of ball_id :
ball_id+ = { ball_id, over_no, ball_no, of_match, inning, extras, wicket_no, total_score, runs_scored, on_strike, off_strike, bowled_by, stricker_batting_position, type_of_extra, boundry }

Since ball_id+ includes all attributes of the relation, ball_id is a Key.

For all functional dependencies in this relation, the left side is ball_id, which is a Key.
Hence, the Per_ball_data relation satisfies BCNF.

9. Wickets Relation

Given FDs:
wicket_id → {wicket_type, player_got_out, caught_by, run_out_by, on_ball, stumped_by}

Compute the Closure of Wicket_id :
wicket_id+ = { wicket_id, wicket_type, player_got_out, caught_by, run_out_by, on_ball, stumped_by }

Since wicket_id+ includes all attributes of the relation, wicket_id is a Key.

For all functional dependencies in this relation, the left side is wicket_id, which is a Key.
Hence, the Wickets relation satisfies BCNF.

10. Final_result Relation

Given FDs:
match_id → {winner, loser, score_of_winner, score_of_loser, win_run_margin, win_wicket_margin, player_of_the_match}

Compute the Closure of Match_id :
match_id+ = { match_id, winner, loser, score_of_winner, score_of_loser, win_run_margin, win_wicket_margin, player_of_the_match }

Since match_id+ includes all attributes of the relation, match_id is a Key.

For all functional dependencies in this relation, the left side is match_id, which is a Key.
Hence, the Final_result relation satisfies BCNF.

11. Player_performance_in_a_match Relation

Given FDs:
{ in_match, player } → {runs, balls_played, wickets_taken, overs_bowled, runs_conceded, fours, sixes, fours_conceded, sixes_conceded}

Compute the Closure of { in_match, player } :
{ in_match, player }+ = { in_match, player, runs, balls_played, wickets_taken, overs_bowled, runs_conceded, fours, sixes, fours_conceded, sixes_conceded }

Since { in_match, player }+ includes all attributes of the relation, { in_match, player } is a Key.

For all functional dependencies in this relation, the left side is { in_match, player }, which is a Key.
Hence, the Player_performance_in_a_match relation satisfies BCNF.

12. Points_table Relation

Given FDs:
team_id → {matches_played, losses, wins, no_result, points, NRR}

Compute the Closure of Team_id :
team_id+ = { team_id, matches_played, losses, wins, no_result, points, NRR }

Since team_id+ includes all attributes of the relation, team_id is a Key.

For all functional dependencies in this relation, the left side is team_id, which is a Key.
Hence, the Points_table relation satisfies BCNF.

13. Users Relation

Given FDs:
user_id → {password, last_access, date_of_sign_in}

Compute the Closure of User_id :

user_id+ = { user_id, password, last_access, date_of_sign_in }

Since user_id+ includes all attributes of the relation, user_id is a Key.

For all functional dependencies in this relation, the left side is user_id, which is a Key.
Hence, the User relation satisfies BCNF.

Hance, All relations are in Boyes Code Normal Form.