BCNF PROOFS

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}

Compute the Closure of Match_id :

match_id+ = { match_id, match_date, toss_decision, status, team1, team2, toss_winner, venue }

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

Compute the Closure of { match_id, player_id } :

{ match_id, player_id }+ = { match_id, player_id, team_id, is_substitute }

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,ball_speed}

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 ,ball_speed}

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}

Compute the Closure of Wicket_id :

wicket_id+ = { wicket_id, wicket_type, player_got_out, caught_by, run_out_by, on_ball }

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. Live_score Relation

Given FDs:

{ match_id, inning } → {team_batting, current_score, wickets_down, overs_bowled, last_ball_description}

Compute the Closure of { match_id, inning } :

{ match_id, inning }+ = { match_id, inning, team_batting, current_score, wickets_down, overs_bowled, last_ball_description }

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

For all functional dependencies in this relation, the left side is { match_id, inning }, which is a key.

Hence, the Live_score relation satisfies BCNF.

11. 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.

12. Player_performance_in_a_match Relation

Given FDs:

{ in_match, player } → {runs, balls_played, wickets_taken, overs_bowled, runs_conceded}

Compute the Closure of { in_match, player } :

{ in_match, player }+ = { in_match, player, runs, balls_played, wickets_taken, overs_bowled, runs_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.

13. Points_table Relation

Given FDs:

team_id → {matches_played, lost, won, no_result, points, NRR}

Compute the Closure of Team_id :

team_id+ = { team_id, matches_played, lost, won, 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.

14. 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.