

Brian Kane & Adrian Wilson
CS 348: Databases
Ed Harcourt
12/4/13

Functional Dependency Analysis Document

Conferences (conference_name)

- conference_name → conference_name

Primary Key: conference_name

Only having one attribute, it is clear that conference_name depends on itself.

Divisions (division_name, conference_name)

- division_name → conference_name

Primary Key: division_name

A division can only belong to one conference, so the conference a division belongs to depends solely on that particular divisions name. Therefore, division_name is enough to distinguish between rows in this relation.

Teams (team_name, division_name)

- team_name → division_name
- team_name → state
- team_name → city
- team_name → stadium

Primary Key: team_name

Based on the principle that every team must have a unique name, it is easy to see why team_name determines all other attributes in this relation. A team's division alone isn't going to be enough, since multiple teams can be in the same division.

Locations (state, city, stadium)

- (state, city, stadium) → state
- (state, city, stadium) → city

- (state, city, stadium) → stadium

Primary Key: (state, city, stadium)

Simply using state as the primary key wouldn't be enough to uniquely distinguish a tuple in this relation, as multiple teams can reside in the same state. Multiple teams can also reside in the same city, therefore including the stadium is necessary to distinguish between locations in this table.

Coaches (team_name, first_name, last_name)

- team_name → first_name

- team_name → last_name

Primary Key: team_name

A coach's first and/or last name is not enough to uniquely distinguish him, since it is very possible for the coaches of two different teams to have the same name. Thus, since each team can only have one head coach, simply providing the team name is enough.

Owners (team_name, first_name, last_name)

- team_name → first_name

- team_name → last_name

Primary Key: team_name

An owner's first and/or last name is not enough to uniquely distinguish him, since it is very possible for the owners of two different teams to have the same name. Thus, since each team can only have one owner, simply providing the team name is enough.

Players (team_name, player_number, position_id, first_name, last_name)

- (team_name, player_number) → position_id

- (team_name, player_number) → first_name

- (team_name, player_number) → last_name

Primary Key: (team_name, number)

Simply providing a player's position or name is not enough to uniquely identify a row in this relation, since the football league contains many players of the same position and can easily contain multiple players with the same name. Since no two players on the same team can share a number, but players from different teams may have the same number, both team_name and player_number are needed to ID a row.

Season (start_year)

- start_year → start_year

Primary Key: start_year

The year in which a season starts obviously determines itself.

Games (home_team_name, away_team_name, game_date, start_year, game_type_id)

- (home_team_name, away_team_name, game_date) → start_year

- (home_team_name, away_team_name, game_date) → game_type_id

Primary Key: (home_team_name, away_team_name, game_date)

It is easy to see why game_date is not enough to ID a tuple, as multiple games are played on the same date every week of the season. By the same type of reasoning, simply providing the names of the competing teams is not enough either, as teams can matchup more than once during a single season. The combination of start_year and game_type_id does not uniquely identify a row either, as many games of the same game_type (i.e. a regular season game) are played within a single season.

Franchises (team_name, start_year, wins, losses)

- (team_name, start_year) → wins

- (team_name, start_year) → losses

Primary Key: (team_name, start_year)

Wins and losses alone cannot uniquely determine a franchise, as many times over the course of many seasons have had identical win/loss records. Therefore, team_name and start_year are enough to decipher between tuples in this relation, since all team names are different and each team can only have one season a year.

Schedules (team_name, player_number, game_date)

- (team_name, player_number, game_date) → (team_name, player_number, game_date)

Primary Key: (team_name, player_number, game_date)

Since teams have multiple games, and games have multiple teams, a schedule is needed to keep track of which teams play each other and when. Since two teams can play each other more than once during a single season and across multiple seasons, the game_date is needed with the names of the teams to uniquely determine tuples in this relation.

Game_Locations (home_team_name, away_team_name, game_date, state, city, stadium)

- (home_team_name, away_team_name, game_date) → state

- (home_team_name, away_team_name, game_date) → city

- (home_team_name, away_team_name, game_date) → stadium

Primary Key: (home_team_name, away_team_name, game_date)

Since each team is associated with a state, city, stadium tuple, simply providing the teams playing a particular game and the date on which the game is held is enough to uniquely determine rows in this table.

Game Types (game_type_id)

- game_type_id → game_type

Primary Key: game_type_id

Obviously game_type_id determines itself as a two to three-letter code.

Positions (position_id)

- position_id → position_name

Primary Key: position_id

Position_id determines itself as a two-letter code.

Performances (team_name, player_number, game_date)

- (team_name, player_number, game_date) → (team_name, player_number, game_date)

Primary Key: (team_name, player_number, game_date)

Since teams have many games throughout the season, neither just a team's name, player's number, or game date is sufficient for uniquely identifying a tuple in this relation, nor is any combination of two of these attributes. Without game_date, performances between different weeks of a season cannot be differentiated. Without player_number, we don't know which player gave a particular performance. Without team_name, we don't know to which player a particular player_number belongs to.

Offensive Performances (team_name, player_number, game_date, passing_yards, rushing_yards, receiving_yards, touchdowns, interceptions, fumbles, quarterback_rating, yards_per_carry, yards_per_reception)

- (team_name, player_number, game_date) → passing_yards
- (team_name, player_number, game_date) → rushing_yards
- (team_name, player_number, game_date) → receiving_yards
- (team_name, player_number, game_date) → touchdowns
- (team_name, player_number, game_date) → interceptions
- (team_name, player_number, game_date) → fumbles
- (team_name, player_number, game_date) → quarterback_rating
- (team_name, player_number, game_date) → yards_per_carry
- (team_name, player_number, game_date) → yards_per_reception

Primary Key: (team_name, player_number, game_date)

It is easy to see that any arbitrary statistic by itself cannot uniquely identify a tuple in this relation, nor can any combination of statistics. The team name coupled with a player's number and the date the game took place, however, does uniquely identify a performance for a particular player for a particular game.

Defensive Performances (team_name, player_number, game_date, tackles, tackles_for_loss, sacks, forced_fumbles, fumble_recoveries, interceptions, passes_defended, tackle_assists)

- (team_name, player_number, game_date) → tackles
- (team_name, player_number, game_date) → tackles_for_loss
- (team_name, player_number, game_date) → sacks
- (team_name, player_number, game_date) → forced_fumbles
- (team_name, player_number, game_date) → fumble_recoveries
- (team_name, player_number, game_date) → interceptions
- (team_name, player_number, game_date) → passes_defended

- (team_name, player_number, game_date) → tackle_assists

Primary Key: (team_name, player_number, game_date)

It is easy to see that any arbitrary statistic by itself cannot uniquely identify a tuple in this relation, nor can any combination of statistics. The team name coupled with a player's number and the date the game took place, however, does uniquely identify a performance for a particular player for a particular game.

Kicker Performances (team_name, player_number, game_date, field_goals_attempted, field_goals_made, field_goal_long, punts, punts_inside_20, punt_avg_distance)

- (team_name, player_number, game_date) → field_goals_attempted
- (team_name, player_number, game_date) → field_goals_made
- (team_name, player_number, game_date) → field_goal_long
- (team_name, player_number, game_date) → punts
- (team_name, player_number, game_date) → punts_inside_20
- (team_name, player_number, game_date) → punt_avg_distance

Primary Key: (team_name, player_number, game_date)

It is easy to see that any arbitrary statistic by itself cannot uniquely identify a tuple in this relation, nor can any combination of statistics. The team name coupled with a player's number and the date the game took place, however, does uniquely identify a performance for a particular player for a particular game.

For all tables, all non-trivial superkeys imply a join dependency.