COMP 3005 Term Project

Seyedsajad Hosseini

101205877

Contents

Conceptual Design

The process of designing the database and writing the code to load JSON files into the PostgreSQL database began with a thorough analysis of the data format stored within the JSON files. During this phase, careful attention was given to studying the Open Data documentation files, which provided insights into less commonly used attributes. In accordance with the project specifications, to maximize data integration into the database, a decision was made to adhere closely to the structure of the Open Data, resulting in the creation of 39 tables.

This process resulted in the successful identification of the essential entities necessary for constructing a relational database. The main entities and their relations with other entities include:

- Team: a team has many players, one or more manager, participates in zero or more matches and competitions
- Player: a player can have multiple jersey numbers, has zero or more team (this can happen when a player is transferred to another team), plays in one or more positions, and may receive zero or more cards
- Match: a match has many (two) teams, many positions for the players to play in, exactly one referee (the json files do not include assistant referees), exactly one stadium, and zero or more events
- Event: event is anything that happens during a match including passes, shots, dribbles, fouls, etc. An event is performed by a player in a match, therefore, each event has exactly one player and one match.

After identifying the entities within the database, the next crucial step was to map them to their respective tables. This phase was the most challenging part of the project, as it directly impacted the efficiency of queries and the implementation of the json loader. Key challenges encountered during this step included:

- How to store the events?

  Each events json file contains thousands of different events characterized by shared and unique attributes. One approach to store these events involves creating a single large event table with attributes covering all event types.

While straightforward in its implementation, this approach leads to a multitude of null values. This happens because, for example a pass event doesn’t have attributes such as card type or statsbombXG. Consequently, such a table can result in poor query performance.

Alternatively, a second approach includes establishing individual tables for each event type. While this method yields better query performance, it generates numerous tables, primarily composed of common attributes alongside a handful of unique ones. Despite its efficiency, this approach creates unnecessarily large number of, still, large tables, which was considered undesirable for this project.

The third approach, employed in the submitted database, involves creating a central “events” table housing shared attributes alongside several smaller tables containing unique attributes for each event type. In this approach, some match events possess no distinct attributes and thus do not require separate tables. However, for passes and shots, it was decided to create dedicated events tables to include both shared and unique attributes. This was a deliberate move to further improve the efficiency of queries pertaining to passes and events. Apart from passes and shots, attributes for all other events are distributed between the central events table and type-specific tables. It's worth noting that certain attributes in type-specific tables aren't exclusive to a single event type. However, as these attributes aren't shared by "all" events, they have been relocated to their respective type-specific event tables.

- How to store attributes with ids?

  Some attributes in the json files contain both an id and a name, such as country, pass type, foul type, and event outcome. When storing these attributes in tables, there are two approaches:

  1. Storing only the attribute name.
  2. Using an additional table for each attribute to store both the id and the name, and then referencing the id in other tables.

For instance, in the first approach, the players table would include a "country_name" attribute. Conversely, in the second approach, the players table would have a "country_id" attribute. To retrieve the name of the country to which a player belongs, a JOIN operation is necessary, linking the players table with the countries table where tuples containing the country id and name are stored.

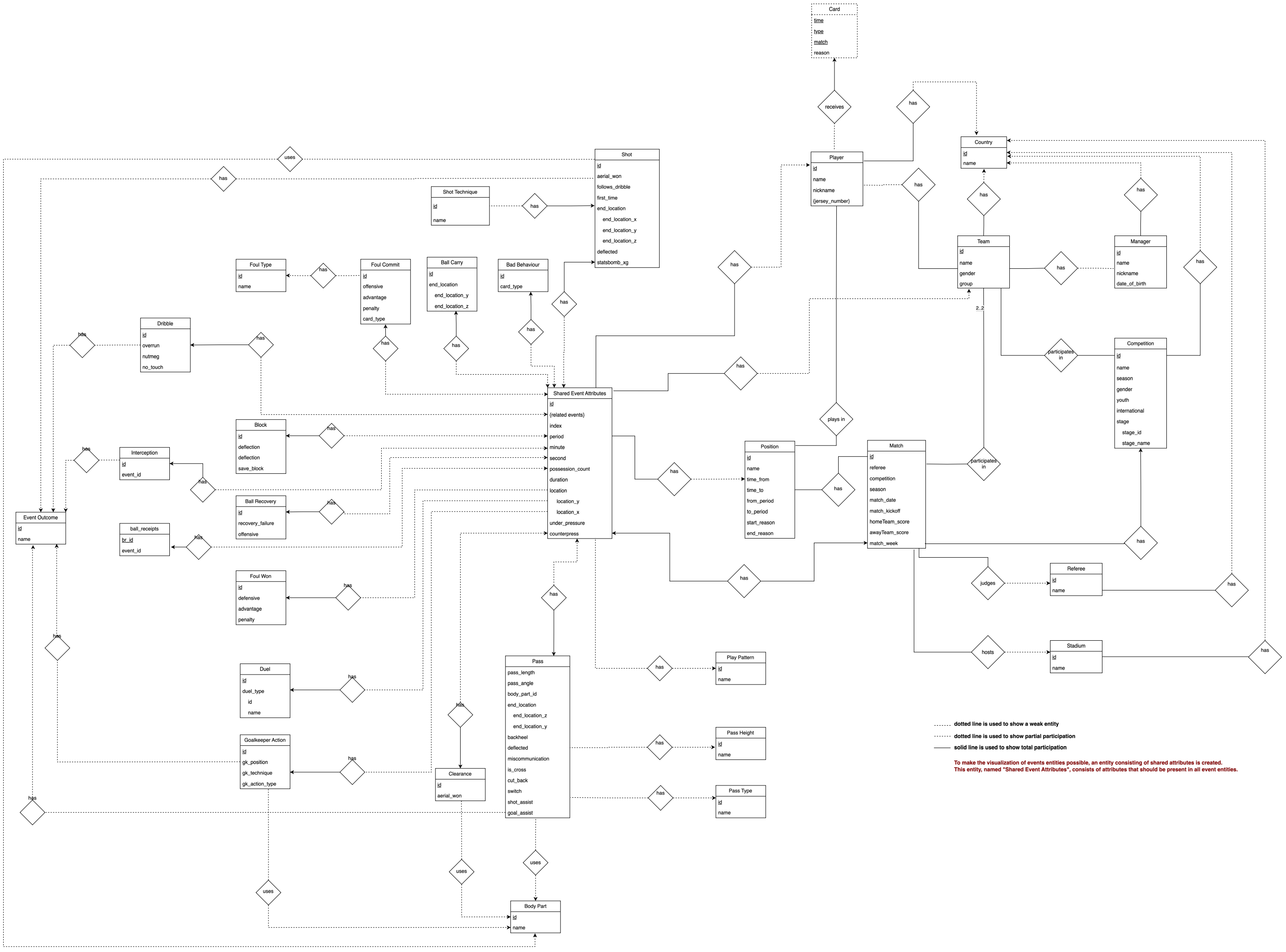
While the initial approach reduces the number of JOIN operations for each attribute, it may lead to slower query execution times when searching for items with long textual variables. This is because comparing an integer field is generally faster than comparing a varchar field. To determine the optimal approach, both methods were implemented and the query times for project queries were compared. Although both approaches yielded similar query times for most questions, the second approach (utilizing separate tables for such attributes) consistently demonstrated more stable query performance. As a result, it was selected the preferable approach.

The ER diagram for this project (page 5) has been saved as an SVG image, allowing you to zoom in to view the details. However, some texts in diamonds have moved out of the place. You can download the image from here:

https://drive.google.com/drive/folders/1i5tU_tl41MoFQ2SjMl8M_ZlKO2fVUo3s?usp=sharing

To make the visualization of event entities possible, a semi-entity[1] consisting of shared attributes is created. This semi-entity, named "Shared Event Attributes", consists of attributes that should be present in all event entities. While a table has been created for this semi-entity, it does not correspond to any single entity in the original or final databases.

[1] Semi-entity is not a technical term. It’s being used here to indicate an object with attributes like an entity but not being an actual entity.



dotted line is used to show a weak entity

dotted line is used to show partial participation

solid line is used to show total participation

To make the visualization of events entities possible, an entity consisting of shared attributes is created.
This entity, named "Shared Event Attributes", consists of attributes that should be present in all event entities.

Reduction to Relation Schemas

countries

sn, country_id, country_name

PK (sn)

stadiums

stadium_id, stadium_name

FK (country_id), PK (stadium_id)

referees

referee_id, referee_name

FK (country_id), PK (referee_id)

managers

manager_id, manager_name, manager_nickname, manager_dob,

FK (country_id), PK (manager_id)

competitions_seasons

competition_id, competition_orig_id, competition_name, competition_gender, competition_youth, competition_international, country_name, season_id, season_name,

PK (competition_id),

teams

team_id, team_orig_id, team_name, team_gender, team_group

FK (country_id), FK (competition_id), PK (team_id),

team_managers

FK (manager_id), FK (team_id), PK (manager_id, team_id)

competition_stages

competition_stage_id, competition_stage_name,

PK (competition_stage_id)

matches

match_id, match_date, match_kickoff, homeTeam_score, awayTeam_score, match_week,

FK (homeTeam_id), FK (awayTeam_id), FK (stadium_id), FK (referee_id),

FK (competition_id), FK (competition_stage_id), PK (match_id),

players

player_id, player_name, player_nickname,

FK (country_id), PK (player_id)

player_jerseys

player_id, jersey_number,

FK (competition_id), PK (player_id, competition_id)

player_positions

player_position_id, player_position_name,

PK (player_position_id)

player_match_positions

time_from, time_to, from_period, to_period, start_reason, end_reason,

FK (match_id), FK (team_id), FK (player_id), FK (player_position_id),

PK (match_id, player_id, player_position_id, time_from)

player_match_cards

card_type, time, reason, period,

FK (match_id), FK (team_id), FK (player_id), PK (match_id, player_id, time)

lineups

FK (match_id), FK (team_id), FK (player_id),
PK (match_id, team_id, player_id)

play_patterns

play_pattern_id, play_pattern_name,
PK (play_pattern_id)

events

event_id, event_db_id, index, period, minute, second, location_x, location_y,
event_type_name, possession_count, duration, under_pressure, counterpress,
FK (possession_team_id), FK (play_pattern_id), FK (player_position_id),
FK (match_id), FK (team_id), FK (player_id), PK (event_id)

related_events

re_id, event_db_id, related_event_db_id,
PK (re_id)

event_outcomes

event_outcome_id, event_outcome_name,
PK (event_outcome_id)

pass_heights

pass_height_id, pass_height_name,
PK (pass_height_id)

body_parts

body_part_id, body_part_name,
PK (body_part_id)

shot_techniques

shot_technique_id, shot_technique_name,
PK (shot_technique_id)

pass_types

pass_type_id, pass_type_name,
PK (pass_type_id)

foul_types

foul_type_id, foul_type_name,
PK (foul_type_id)

ball_receipts

br_id,
FK (event_id), FK (event_outcome_id), PK (br_id)

ball_carries

bc_id, end_location_x, end_location_y,
FK (event_id), PK (bc_id)

events_passes

event_id, event_db_id, index, period, minute, second, possession_count, location_x, location_y, duration, under_pressure, counterpress, pass_length, pass_angle, end_location_x, end_location_y, assisted_shot_id, backheel, deflected, miscommunication, is_cross, cut_back, switch, shot_assist, goal_assist,
FK (pass_height_id), FK (body_part_id), FK (recipient_id), FK (match_id), FK (possession_team_id), FK (play_pattern_id), FK (player_position_id), FK (team_id), FK (player_id), FK (pass_type_id), FK (event_outcome_id), FK (pass_technique_id), PK (event_pass_id)

ball_recoveries

br_id, recovery_failure, offensive,

FK (event_id), PK (br_id)

duels

duel_id, duel_type_id, duel_type_name

FK (event_id), FK (event_outcome_id), PK (duel_id)

blocks

block_id, deflection, offensive, save_block,

FK (event_id), PK (block_id)

clearances

clearance_id, aerial_won,

FK (event_id), FK (body_part_id), PK (clearance_id)

interceptions

interception_id,

FK (event_id), FK (event_outcome_id), PK (interception_id)

dribbles

dribble_id, overrun, nutmeg, no_touch,

FK (event_id), FK (event_outcome_id), PK (dribble_id)

events_shots

event_id, event_db_id, index, period, minute, second, possession_count,

location_x, location_y, duration, under_pressure, counterpress, key_pass_id,

aerial_won, follows_dribble, first_time, open_goal, deflected, statsbomb_xg,

end_location_x, end_location_y, end_location_z,

FK (match_id), FK (team_id), FK (player_id), FK (possession_team_id),

FK (play_pattern_id), FK (player_position_id), FK (body_part_id), FK (technique_id), FK (shot_type_id), FK (event_outcome_id), PK (event_id)

substitutions

substitution_id,

FK (event_id), FK (event_outcome_id), FK (replacement_id),

PK (substitution_id)

foul_wons

foul_won_id, defensive, advantage, penalty,

FK (event_id), PK (foul_won_id)

foul_commits

foul_commit_id, offensive, advantage, penalty, card_type,

FK (event_id), FK (foul_type_id), PK (foul_commit_id)

goalkeeper_actions

gk_action_id, gk_position, gk_technique, gk_action_type,

FK (event_id), FK (body_part_id), FK (event_outcome_id), PK (gk_action_id)

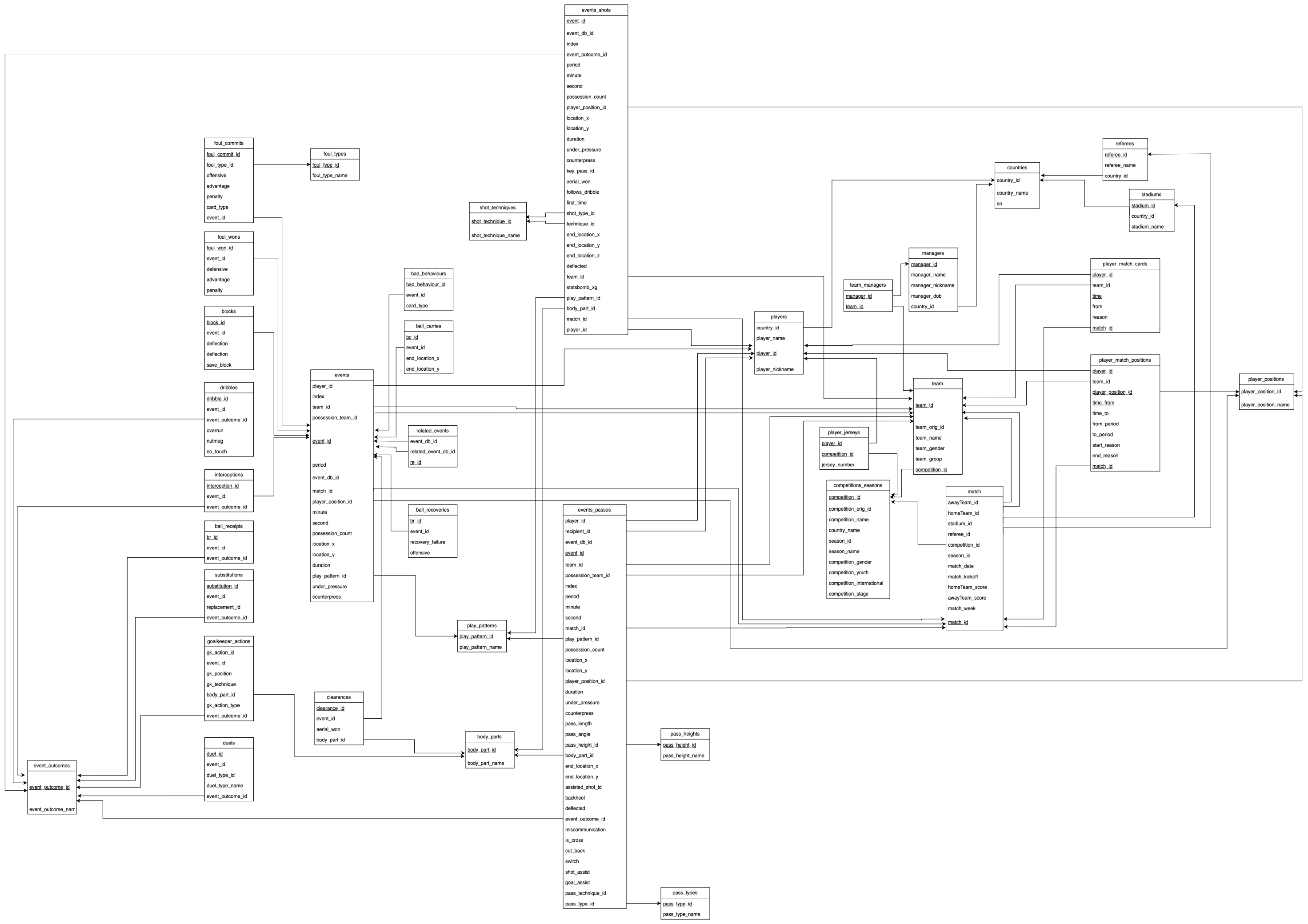
bad_behaviours

bad_behaviour_id, card_type,

FK (event_id), PK (bad_behaviour_id)

Database Schema Diagram

The diagram below has been saved as an SVG image, allowing you to zoom in to view the details. You can download the file from Google Drive.


events_shots
event_id
event_db_id
index
event_outcome_id
period
minute
second
possession_count
player_position_id
location_x
location_y
duration
under_pressure
counterpress
key_pass_id
aerial_won
follows_dribble
first_time
shot_type_id
technique_id
end_location_x
end_location_y
end_location_z
deflected
team_id
statsbomb_xg
play_pattern_id
body_part_id
match_id
player_id
foul_commits
foul_commit_id
foul_type_id
offensive
advantage
penalty
card_type
event_id
foul_types
foul_type_id
foul_type_name
shot_techniques
shot_technique_id
shot_technique_name
foul_wons
foul_won_id
event_id
defensive
advantage
penalty
bad_behaviours
bad_behaviour_id
event_id
card_type
ball_carries
bc_id
event_id
end_location_x
end_location_y
blocks
block_id
event_id
deflection
deflection
save_block
dribbles
dribble_id
event_id
event_outcome_id
overrun
nutmeg
no_touch
events
player_id
index
team_id
possession_team_id
event_id
period
event_db_id
match_id
player_position_id
minute
second
possession_count
location_x
location_y
duration
play_pattern_id
under_pressure
counterpress
related_events
event_db_id
related_event_db_id
re_id
interceptions
interception_id
event_id
event_outcome_id
ball_recoveries
br_id
event_id
recovery_failure
offensive
ball_receipts
br_id
event_id
event_outcome_id
substitutions
substitution_id
event_id
replacement_id
event_outcome_id
goalkeeper_actions
gk_action_id
event_id
gk_position
gk_technique
body_part_id
gk_action_type
event_outcome_id
clearances
clearance_id
event_id
aerial_won
body_part_id
duels
duel_id
event_id
duel_type_id
duel_type_name
event_outcome_id
event_outcomes
event_outcome_id
event_outcome_nam
play_patterns
play_pattern_id
play_pattern_name
body_parts
body_part_id
body_part_name
events_passes
player_id
recipient_id
event_db_id
event_id
team_id
possession_team_id
index
period
minute
second
match_id
play_pattern_id
possession_count
location_x
location_y
player_position_id
duration
under_pressure
counterpress
pass_length
pass_angle
pass_height_id
body_part_id
end_location_x
end_location_y
assisted_shot_id
backheel
deflected
event_outcome_id
miscommunication
is_cross
cut_back
switch
shot_assist
goal_assist
pass_technique_id
pass_type_id
pass_heights
pass_height_id
pass_height_name
pass_types
pass_type_id
pass_type_name
players
country_id
player_name
player_id
player_nickname
countries
country_id
country_name
sn
referees
referee_id
referee_name
country_id
stadiums
stadium_id
country_id
stadium_name
managers
manager_id
manager_name
manager_nickname
manager_dob
country_id
team_managers
manager_id
team_id
player_match_cards
player_id
team_id
time
from
reason
match_id
team
team_id
team_orig_id
team_name
team_gender
team_group
competition_id
player_jerseys
player_id
competition_id
jersey_number
competitions_seasons
competition_id
competition_orig_id
competition_name
country_name
season_id
season_name
competition_gender
competition_youth
competition_international
competition_stage
match
awayTeam_id
homeTeam_id
stadium_id
referee_id
competition_id
season_id
match_date
match_kickoff
homeTeam_score
awayTeam_score
match_week
match_id
player_match_positions
player_id
team_id
player_position_id
time_from
time_to
from_period
to_period
start_reason
end_reason
match_id
player_positions
player_position_id
player_position_name

Bonus Queries

The solution to bonus queries are

1.

```
SELECT player_name, COUNT(*) as shot_count
FROM competitions_seasons
JOIN matches
     ON competitions_seasons.competition_id =
matches.competition_id
JOIN events_shots
     ON matches.match_id = events_shots.match_id
          AND (end_location_x >= 119.9)
          AND (end_location_z >= 2.67*2/3 AND end_location_z
<= 2.67)
          AND ((end_location_y >= 36 AND end_location_y <=
36+8/3)
               OR (end_location_y >= 44-8/3 AND
end_location_y <= 44))
JOIN players
     ON events_shots.player_id = players.player_id
WHERE competitions_seasons.competition_name = 'La Liga'
          AND (competitions_seasons.season_name = '2018/2019'
               OR competitions_seasons.season_name =
'2019/2020'
               OR competitions_seasons.season_name =
'2020/2021')
GROUP BY player_name
ORDER BY shot_count DESC;
```

2.

```
SELECT team_name, COUNT(*) as pass_count
FROM competitions_seasons
JOIN matches
     ON competitions_seasons.competition_id =
matches.competition_id
           AND competitions_seasons.competition_name = 'La
Liga'
           AND competitions_seasons.season_name = '2020/2021'
JOIN events_passes
     ON matches.match_id = events_passes.match_id
           AND (end_location_x >= 102.4 AND end_location_y >=
19.9 AND end_location_y <= 60.1)
JOIN teams
     ON events_passes.team_id = teams.team_id
GROUP BY team_name
ORDER BY pass_count DESC;
```

You can find the YouTube video of the results here: https://youtu.be/WSV_4Svxqj8