

RESPONSE:

I decided to first keep the teams and games tables very similar since the JSON files both only had the primary key (ID's) along with the name/date without any other relation from a foreign key. However, for a Player I split up the player table into just the player_id that has a joint table with teams found from Player_team since many players can play on multiple teams and multiple teams can have multiple players. Furthermore, I decided shots should be its own table since each player can take multiple shots in a game and relate it to the game_id as well. This is a similar logic to passes although turnovers made it slightly more complicated since a pass can be a turnover and a turnover was its own section within the player JSON. Therefore, I decided to create a separate turnover table that is related directly to the player_id, game_id and pass_id (which is optional). To get if a pass is a turnover requires checking if the given pass_id exists within the turnover table which is required for the backend portion. The passes table is also related to the game_id and player_id and may or may not have a turnover related to the given pass. This creates a normalized structure required to efficiently query the database for given values.

