


Mongo = NO SQL

* User defined system definition could add nearly
 ** for Optim. ~~ization~~
 *** increasing Scope provides richer experience

Users Obj:

Display_name: String
 user_id: integer  for quicker comparison
 email: String

Note, all (type_id) in mod
 text signifies a foreign key

League Obj:

owner: integer (user_id)
 name: String
 description: Text
 logo: bytes
 region: String *
 game_type: String *
 tournaments: array ** (tournament_ids)
 League_id: integer
 Teams: array [integer] (team id)

Tournament Obj:

Teams: array (team_ids)
 Matches: array [integer] (match_id) ← ordering to store bracket data.
 League: integer (league_id)
 Name: String
 description: Text
 Tournament_id: integer
 generated: boolean

↙ No. Let organizers deal with that,
 not all tournaments need this.

Do we want a new field for start time? Doesn't have to do anything
 Special logic wise but could be nice for displaying instead of just being in
 description

Team Obj:

Roster: array [String] *** list of names, 10 per
 Wins: integer
 Losses: integer
 Ties: integer
 Rating: integer
 Owner: integer (user_id)
 Name: String
 Description: text
 Team_id: integer
 Logo: bytes
 Upcoming_matches: array [integer] (match_id)

Match Obj:

Match: String - Query String "{Home}@{Visitor}"
 Home: integer (team_id)
 Visitor: integer (team_id)
 Victor: integer (team_id)
 Loser: integer ** (team_id)
 Victor_Score: integer
 Loser_Score: integer

Confirmed: boolean — Match is not complete until True

in_conflict: boolean — a conflict was detected

Match_id: integer

Victor_to: integer (match_id) — Victor of this match should be added to the match.

Fill_from: 2-tuple [integer, integer] (match_id, match_id) → used by front-end to map a tournament.

in_tournament: integer (tournament_id)