

Sports Wagering Database Proposal

This document has two purposes. First, it is intended to communicate our current understanding of the data and relationships we need to capture and manage. Second, it serves as first steps for describing business requirements to team members, who in turn will be responsible for implementing the system.

Team members

- Christian Wettre
- Mahima Chandan
- Dale Pippert
- Amrutha Ravi
- Madeline Rys

GitHub URLs

Sports wagering project site

<https://github.com/madelinerys/CS546-Final-Project>

Sports wagering project proposal

<https://github.com/madelinerys/CS546-Final-Project/blob/main/doc/ProjectProposalSportsWagering.pdf>

Sports wagering database proposal (this document)

<https://github.com/madelinerys/CS546-Final-Project/blob/main/doc/DatabaseProposalSportsWagering.pdf>

Games collection

Each document describes a single NFL game, capturing only those few aspects of a game that are relevant to the system, such as its start date/time, and final score. An NFL season is 17 weeks with 14 games per week, so this collection for a full season would have $17 * 14 = 238$ documents in it. The system will insert games into the collection in an automated fashion by either seeding the entire collection all at once or by updating the collection with new games on a week-by-week basis. Either way, it is essential that games be entered into this collection at least several days prior to their start time, so that bettors have a chance to make their wagers on the game.

Game documents are inserted with a null `ascore` and null `hscore`. A system background job is responsible for updating `ascore` and `hscore` with the game's final score.

Games schema

Field	Type	Description
<code>_id</code>	ObjectId	Mongo-generated key for the document
<code>away</code>	String	designator for away (visiting) team
<code>home</code>	String	designator for home team
<code>week</code>	Integer	Week of the season, 1-17. NFL weeks start on Tuesday and end on Monday. As a frame of reference, 10/21/2020 is in Week 7. If you want to know, for example, what is the <i>current week right now</i> as a frame of reference, you can go here and it should default to bring up the current week pre-selected.
<code>start</code>	Date	Game start date and time GMT. The system will not allow bets to be placed on any game where <code>start</code> is \geq current date and time. Upcoming games must be inserted into this collection at least several days prior to <code>start</code> , so that bettors have a chance to place their wagers on the game.
<code>ascore</code>	Integer	Away team final score. Null up until when the game has finished and the system has processed the final score feed.
<code>hscore</code>	Integer	Home team final score. Null up until when the game has finished and the system has processed the final score feed.

Games example document

```
{
  _id: "5f8578c116e3409b5b276d50",
  away: "TEX",
  home: "TIT",
  week: 6,
  start: "2020-10-18T17:00:00.000Z"
  ascore: 36,
  hscore: 42
}
```

Games notes

1. NFL is the [National Football League](#).
2. GMT is four hours ahead of EDT, and five hours ahead of EST. For example, 1:00 PM EDT is 5:00 PM GMT.
3. Designators for teams defined in [Teams](#) collection as [Teams.abbrev](#).

Teams collection

A seeded reference collection to store static identities for all 32 NFL teams. This collection has exactly 32 documents in it, one per team.

Teams schema

Field	Type	Description
<code>_id</code>	ObjectId	Mongo-generated key for the document
<code>abbrv</code>	String	three-letter unique business key to enhance readability
<code>fran</code>	String	franchise name, typically locale/city/state of team
<code>nn</code>	String	team nickname

Teams example document

```
{
  _id: "5f85808c9dad05d358aae011",
  abbrv: "TIT",
  fran: "Tennessee",
  nn: "Titans"
}
```

Lines collection

Stores betting lines for each game. The system populates this collection on a week-by-week, day-by-day basis. It is updated with new lines every day or two via a background job.

Lines schema

Field	Type	Description
<code>_id</code>	ObjectId	Mongo-generated key for the document
<code>gameid</code>	ObjectId	<code>_id</code> from Games collection
<code>ltype</code>	String	AML HML ASP HSP OV UN (see notes)
<code>num</code>	Integer	The number, the meaning of which depends on <code>ltype</code> ; may be positive, negative, or 0.
<code>date</code>	Date	effective date of the line

Lines example document

```
{
  _id: "5f85808c9dad05d358aae00a",
  gameid: "5f8578c116e3409b5b276d50",
  ltype: "ASP",
  num: -2,
  date: "2020-10-16T00:00:00.000Z",
}
```

Lines notes

1. **ltype** valid values are:
 - a. **AML** Away team money line.
 - b. **HML** Home team money line.
 - c. **ASP** Away team spread.
 - d. **HSP** Home team spread.
 - e. **OV** Over points.
 - f. **UN** Under points.
2. There can be many documents for any given **gameid** and **ltype**.
3. For any given **gameid** and **ltype**, the current line for that **gameid** and **ltype** is the line with the most recent date.
4. For any given **gameid** and **ltype**, bets are always placed against the line having the most recent date, i.e., the current line.

Bets collection

Records and stores bets. Each document is a bet from a bettor aka user. The user interface is used to enter bets.

Bets schema

Field	Type	Description
<code>_id</code>	ObjectId	Mongo-generated key for the document
<code>bettorid</code>	ObjectId	<code>_id</code> of the bettor from Bettors collection
<code>lineid</code>	ObjectId	<code>_id</code> of the line from Lines collection
<code>amount</code>	Number	dollar amount of the bet
<code>pays</code>	Number	dollars this bet pays, or null if bet is still live
<code>collects</code>	Number	total dollars this bet collects should bettor win; this is equal to amount + pays
<code>paid</code>	Number	dollar amount this bet paid, or null if bet is still live; may be zero indicating this bet has resolved and was a loss for the bettor
<code>entered</code>	Date	time and date of the bet
<code>closed</code>	Date	time and date the bet resolved, or null if bet is still live

Bets example document

```
{
  _id: "3e85908c9dad05d2589ae104",
  bettorid: "4e9441Cc9dad05d268aff018",
  lineid: "8015617daebe1773199ec12C",
  amount: 55,
  pays: 50,
  collects: 105,
  paid: null,
  entered: "2020-10-22T14:12:35.738Z",
  closed: null
}
```

Bettors collection

These are users aka bettors that have signed up. Possibly (time permitting) seeded with 1000 bettors for demo purposes.

Bettors schema

Field	Type	Description
<code>_id</code>	ObjectId	Mongo-generated key for the document
<code>username</code>	String	
<code>pwd</code>	String	md5 hashed password
<code>balance</code>	Number	dollar balance in account

Bettors example document

```
{
  _id: "3e85908c9dad05d2589ae104",
  username: "foghorn5",
  pwd: "d4ec6fe6cec7f63630376c0af7212a52",
  balance: 250.00
}
```

1. Surprisingly, this collection will be seeded with exactly the same 1000 people from an earlier people.json lab. Apparently they all like to gamble!