

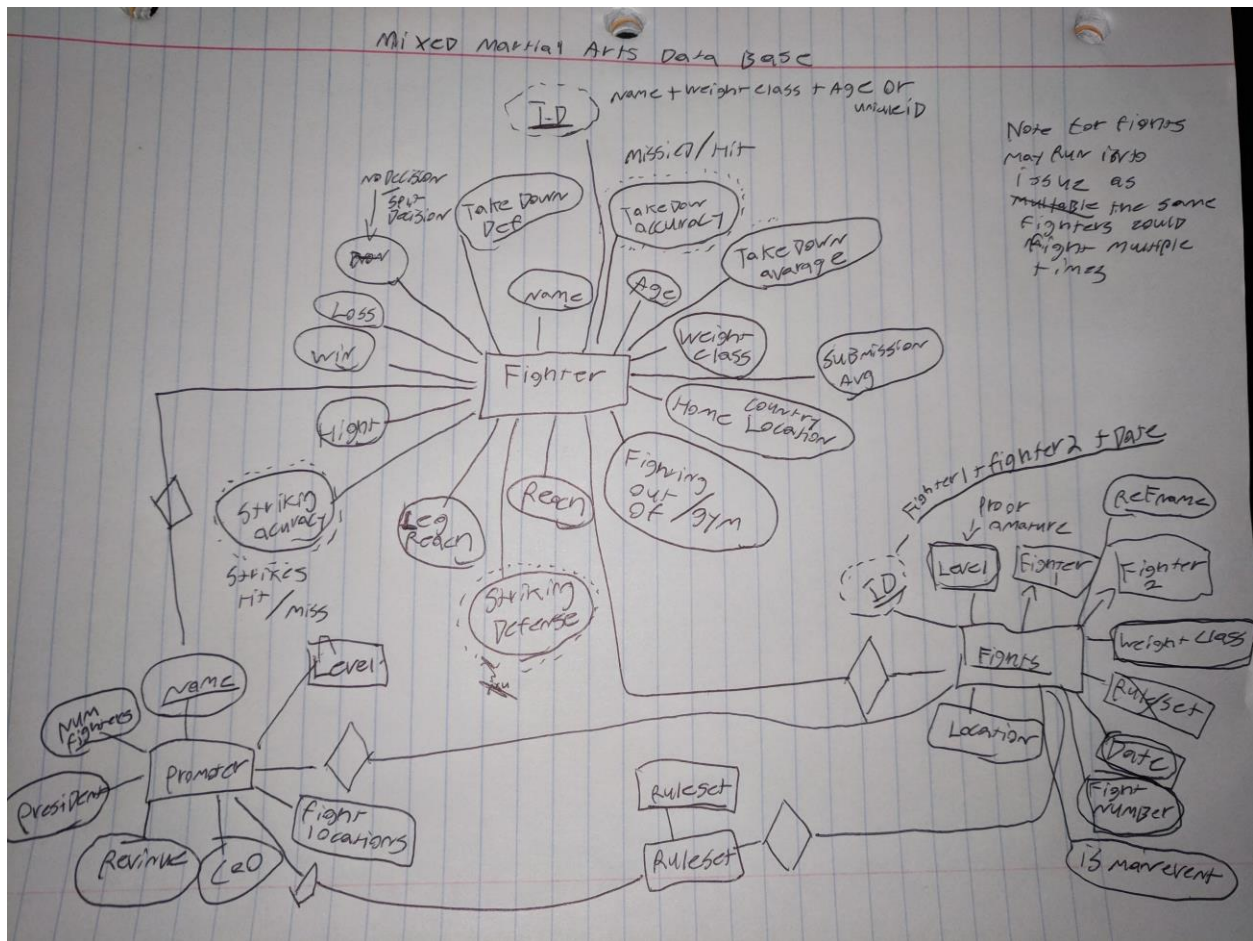
The technical specifications of the database based on the requirements set for by ESPN are as follows.

- Create a relational database.
- Be able to store relevant data effectively.
  - Create a relational design that disallows data redundancy as well as creation, update and deletion errors by progressing to the third normal form.
- Cover relevant aspects of combat sports.
  - Create entities for fighters, stats, fights and the relations between them.
- Have the ability to compound on existing statistics.
  - Allow the insertion of existing statistics to further query.
- Have the ability to generate new insights based on current information.
  - Have a relational design that expresses the nature of combat sports.
- Be general enough to be applicable to all combat sports.
  - Implement additional entities such as commissions and promoters.

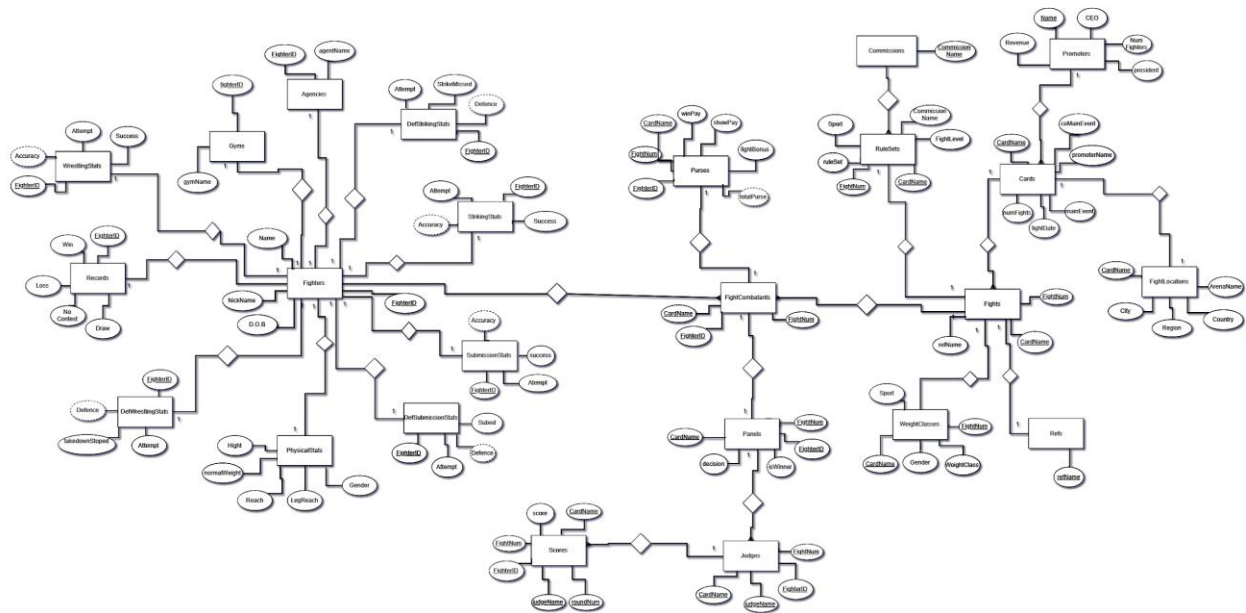
The assumptions made in the database are as follows.

- There should only be two fighters per fight.
- varchar(100) should be large enough to store all text data.
- Each fight should have a promoter, card, location, weight class, ref, and commission.
- Each fighter will have a record, physical stats, gym, agent and discipline(striking, wrestling, etc.) related stats.
- Information that is relevant to only one fighter in a fight is related to the FightCombants entities.

# First ER diagram



## Final ER diagram



My ER diagram is in the third normal form as evident by the ER diagram and how all columns are relevant to the primary key.

## Data Dictionary

## Agencies

The Agencies entity serves the purpose of containing a given fighter's agent. This information may be useful in comparing fighters with the same agent. The attributes of Agencies are AgentName and fighterID. This entity should have at most one record per fighter in the database. This is due to the fighterID being the primary key of Agencies.

### Attributes Types

- AgentName - varchar(100)
- fighterID - int

### Attributes Use

- AgentName - Used to represent the identity of the agent.
- fighterID - Allows a fighter to have an agent.

## Cards

The Cards entity is a representation of a fight card. The purpose of Cards is that they are made by a Promoter and contain fights. In the database, Cards provides a way to link all the fights and fighters that occur on a given card. This entity should have one record per card. The only limitation is that of the primary key cardName. Due to what data is entered it may be possible to have one card per fight or around ten fights per card. Under normal circumstances, the latter of the two bounds is more likely as most cards have nine to fifteen fights.

### *Attributes Types*

- cardName - varchar(100)
- numFights - int
- mainEvent - varchar(200)
- coMainEvent - varchar(200)
- promoterName - varchar(100)

### *Attributes Use*

- cardName - The name of the card.
- numFights - The number of fights on a given card.
- mainEvent - The text representation of the main event fight.
- coMainEvent - The text representation of the co-main event fight.
- promoterName - The name of the promoter that put on the card.

## Commissions

\_\_\_\_\_The commission entity represents a state athletic commission. Commissions are the organization that sanction fights, adopt and enforce rule sets. The number of records in commissions should be one for every state or providence in which a combat sporting event has occurred.

### *Attributes Types*

- commissionName - varchar(100)

### *Attributes Use*

- commissionName - used to identify the commission sanctioning a fight.

## DefStrikingStats

\_\_\_\_\_The DefStrikingStats entity serves the purpose of containing a given fighter's defensive striking statistics. The statistics in DefStrikingStats are primarily concerned with how many strikes has a fighter avoided. There will be at most one record in DefStrikingStats per fighter in the database.

### *Attributes Types*

- fighterID - int
- attempt - float
- strikeMissed - float
- defense - computed as  $((\text{attempt} - \text{strikeMissed}) * 100)$

### *Attributes Use*

- fighterID - A fighter's ID so they can be linked to their defensive striking stats.
- attempt - The number of strikes that have been thrown at a fighter.
- strikeMissed - The number of strikes that have missed hitting a fighter
- defense - A representation of a fighter's striking defense, the percentage of strikes that miss them.

## DefSubmissionStats

\_\_\_\_\_The DefSubmissionStats entity serves the purpose of containing a given fighter's defensive submission statistics. The statistics in DefSubmissionStats are primarily concerned with how many submissions has a fighter avoided. There will be at most one record in DefSubmissionStats per fighter in the database.

### *Attributes Types*

- fighterID - int
- attempt - float
- subed - float
- defense - computed as  $((\text{attempt} - \text{subed}) * 100)$

### *Attributes Use*

- fighterID - A fighter's ID so they can be linked to their defensive submission stats.
- attempt - The number of submissions attempts a fighter has faced.
- subed - The number of times a fighter has been submitted.
- defense - - A representation of a fighter's submission defense, the percentage of submissions they defend.

## DefWrestlingStats

The DefWrestlingStats entity serves the purpose of containing a given fighter's defensive submission statistics. The statistics in DefWrestlingStats are primarily concerned with how many takedowns has a fighter avoided. There will be at most one record in DefWrestlingStats per fighter in the database.

### *Attributes Types*

- fighterID - int
- attempt - float
- takedownStoped- - float
- defense - computed as  $((\text{attempt} - \text{takedownStoped}) * 100)$

### *Attributes Use*

- fighterID - A fighter's ID so they can be linked to their defensive submission stats.
- attempt - The number of shots attempts a fighter has had to defend.
- takedownStoped- The number of times a fighter has been taken down.
- defense - - A representation of a fighter's wrestling defense, the percentage of shots they defend.

## FightCombatants

\_\_\_\_\_The FightCombatants entity is needed as the relationship between Fighters and Fights is a many to many relationship. FightCombatants represents the occurrence of a fight. As such data that is dependent on the outcome of the fight is related to the FightCombatants entity. There should be two records in FightCombatants for each record in the Fights entity.

### *Attributes Types*

- fighterID - int
- fightNum - int
- cardName - varchar(100)

### *Attributes Use*

- fighterID - A fighter's ID so they can be linked to a fight.
- fightNum - The number into a card a given fight occurs, part of Fights primary key.
- cardName - The name of the card on which the fight has occurred, part of Fights primary key

## Fighters

\_\_\_\_\_The Fighters entity is needed as it is responsible for containing the data about a given fighter. In addition, the entity is needed as fighters are the subject matter on which most the statistics will be generated. There should be two records in Fighters for each record in Fights to receive the best quality of statistics.

### *Attributes Types*

- fighterID - int
- fighterName - varchar(100)
- dateOfBirth - date
- nickName - varchar(100)

### *Attributes Use*

- fighterID - Represent a fighter in the database.
- fighterName - Contains the fighters name.
- dateOfBirth - The date of birth of a fighter.
- nickName - Contains the fighters nick-name.

## FightLocations

The FightLocations entity is needed to when progressing to the third normal form as there needs to be a way to represent a fights location. There should be one record in FightLocations for each record in Cards.

### *Attributes Types*

- cardName - varchar(100)
- city - varchar(100)
- country -varchar(100)
- region - varchar(100)
- arenaName -varchar(100)

### *Attributes Use*

- cardName - Link back to Cards as a card should have a location.
- city - City in which the card is being held.
- country - Country in which the card is being held.
- region - Used to specify less specific information in the US like northeast or provinces in other countries.
- arenaName - The arena in which the card is being held.

## Fights

The Fights entity is needed to contain the core information of a fight and as a method of matching fighters to fights. Also, the entity is needed as the table to which all static information of a fight is related to such as a referee. There should be about nine to fifteen records in Fights for each record in Cards. This is due to the usual number of fights on a card being reliably between nine to fifteen.

### *Attributes Types*

- cardName - varchar(100)
- fightNum - int
- refName - varchar(100)

### *Attributes Use*

- cardName - Link back to Cards.
- fightNum - The number of fights into a card the given fight is.
- refName - The referee of the fight, links back to the Refs entity.



## Gyms

The Gyms entity is needed to contain the fighter's main gym and future filter data when searching for insights. There should be one record in Gyms for each record in Fighters.

### *Attributes Types*

- gymName - varchar(100)
- fighterID - int

### *Attributes Use*

- gymName - The name of a fighter's main gym
- fighterID - Link back to a fighter and show a fighter's main gym.

## Judges

The Judges entity is needed as a point to relate the Scores entity to which in turn is needed to provide more detailed information about a fight. In addition, having a Judges entity allows having many judges per fight on a panel. There should be around six to ten records in Judges for each record in FightCombatants. This is due to most fights having three to five judges which must give scores to two fighters.

### *Attributes Types*

- judgeName - varchar(100)
- cardName - varchar(100)
- fightNum - int
- fighterID - int

### *Attributes Use*

- judgeName - The name used to identify a given judge.
- cardName - Link to Panels.
- fightNum - Link to Panels.
- fighterID - Link to Panels.

## Panels

The Panels entity is needed to contain the fighter specific decision and judging information about the fight. This entity also allows for there to be many judges per fight as is common. There should be one record in Panels for each record in FightCombatants.

### *Attributes Types*

- judgingSystem - varchar(100)
- decision - varchar(100)
- fightNum - int
- cardName - varchar(100)
- fighterID - int
- isWinner - bit

### *Attributes Use*

- judgingSystem - A short description of the judging system ex. 10 points must.
- decision - The fighter specific description of the fight outcome ex. TKO, doctor stoppage.
- fightNum - Provides a link to FightCombatants.
- cardName - Provides a link to FightCombatants
- fighterID - Provides a link to FightCombatants
- isWinner - Bit representation if the fighter won the fight.

## PhysicalStats

\_\_\_\_\_The PhysicalStats entity is needed to represent a fighter's physical stats. The PhysicalStats entity allows of greater depth of searching and filtering when querying the database. There should be one record in PhysicalStats for each fighter in the database.

### *Attributes Types*

- fighterID - int
- hight - int
- reach - int
- legReach - int
- Gender - varchar(100)
- normalWeight - int

### *Attributes Use*

- fighterID - Link back to a fighter, a fighter's physical stats.
- hight - Fighter's hight in inches.
- reach - Fighter's arm reach in inches.
- legReach - Fighter's leg reach in inches.
- Gender - Fighter's gender.
- normalWeight - Fighter's normal weight class.

## Promoters

The Promoters entity is needed as the starting point of a card for which most of the database is reliant on or related to. In addition, the ability to have different promoters allows for the separation of cards. There should be a minimal amount of records in Promoters as only one is needed for many cards. The maximum amount of records in Promoters should be no more than the number of cards. However, it is most likely that the records in Promoters will trend towards the minimum.

### *Attributes Types*

- promoterName - varchar(100)
- CEO - varchar(100)
- preisdent - varchar(100)
- numFighters - int
- revenue - money

### *Attributes Use*

- promoterName - The name of the promotion.
- CEO - The name of the promotion's CEO.
- president - The name of the promotion's president.
- numFighters - The number of fighters on a promotions roster.
- revenue - Store how much the promotion makes.

### Purses

The Purses entity is needed since there should be a way to store the pay a fighter receives for a fight. The pay information is of importance to the relation between fighters and fights. In addition, any media coverage of a fight normal has the fighter's purse disclosed. There should be one record in Purses for each record in FightCombatants.

### *Attributes Types*

- fightNum - int
- cardName - varchar(100)
- fighterID - int
- winPay - money
- showPay - money
- fightBounus - money
- totalPurse - computed as (fightBounus + showPay + winPay)

### *Attributes Use*

- fightNum - Provides a link to FightCombatants.
- cardName - Provides a link to FightCombatants
- fighterID - Provides a link to FightCombatants
- winPay - Payment received for winning a fight.
- showPay - Payment received for fight fighting.
- fightBonus - The amount of the bonus if the fighter receives one.
- totalPurse - The total amount of money a fighter received.

### Records

The records entity is needed as it contains a fighter's record. The goal was to have triggers update this table as decisions are entered as a record should be related to a fighter. However, to due time constraints this feature was not implemented. There should be one record in Records for each record in Fighters.

### *Attributes Types*

- win - int
- loss - int
- draw - int
- noContest - int
- fighterID - int

### *Attributes Use*

- win - The number of fights won.
- loss - The number of fights lost.
- draw - The number of fights ending in a draw.
- noContest - The number of fights ending in a no contest.
- fighterID - Link back to fighters, a given fighters record.

### Refs

### *Attributes Types*

- refName - varchar(100)

### *Attributes Use*

- refName - The name of the referee.

### RuleSets

The RuleSets entity is needed as it is the primary method how different combat sports are differentiated. Also, RuleSets is needed to correctly represent the relation between commissions and fights. There should be one record in RuleSets for each record in Fights.

### *Attributes Types*

- ruleSet - varchar(100)
- commissionName - varchar(100)
- sport - varchar(100)
- fightLevel - varchar(100)
- fightNum - int
- cardName - varchar(100)

### *Attributes Use*

- ruleSet - Name of rule set

- commissionName - The name of the commission sanctioning the fight.
- sport - Name of the sport the ruleset is under.
- fightLevel - description of the fight level ex. professional, amateur, high school
- fightNum - Link to a given fight for which this rule set is for.
- cardName - Link to a given fight for which this rule set is for.

## Scores

The Scores entity is needed as it allows for a many to one relation with Judges. The entity also facilitates entering of scores based on rounds which is how scores are kept. The number of records should be around (number of rounds of a fight \* number of judges on the panel \* 2). For a max five round MMA fight, this would be thirty records and a 12 round boxing fight would have one hundred and twenty.

### *Attributes Types*

- judgeName - varchar(100)
- cardName - varchar(100)
- fightNum - int
- fighterID - int
- roundNum - int
- score - int

### *Attributes Use*

- judgeName - Judges name, and the one giving the score.
- cardName - Link to Judges, fight for which this score is relevant to.
- fightNum - Link to Judges, fight for which this score is relevant to.
- fighterID - Link to Judges, the fighter for which this score is relevant to.
- roundNum - Round of the fight that the score is relevant to.
- score - The numeric score given.

## StrikingStats

\_\_\_\_\_The StrikingStats entity is needed as there should be a way to represent the fighter's striking stats. These striking stats are used to generate the statistics produced by the database. There should be one record in StrikingStats for each record in Fighters.

### *Attributes Types*

- fighterID - int
- success - float

- attempt - float
- accuracy - computed as  $((\text{attempt} - \text{success}) * 100)$

#### *Attributes Use*

- fighterID - Link to Fighters, a given fighters striking stats.
- success - The number of successful strikes a fighter has landed.
- attempt - Total number of strikes thrown.
- accuracy - representation of a fighter's offensive striking, the percentage of successful to unsuccessful strikes.

#### SubmissionStats

The SubmissionStats entity is needed as there should be a way to represent the fighter's submission stats. These submission stats are used to generate the statistics produced by the database. There should be one record in SubmissionStats for each record in Fighters.

#### *Attributes Types*

- fighterID - int
- success - float
- attempt - float
- accuracy - computed as  $((\text{attempt} - \text{success}) * 100)$

#### *Attributes Use*

- fighterID - Link to Fighters, a given fighter's striking stats.
- success - The number of submissions a fighter has.
- attempt - Total number of submissions attempted.
- accuracy - representation of a fighter's offensive submission skills, the percentage of successful to unsuccessful submission attempts

#### WeightClasses

The WeightClass entity is need as the weight class of a fight need as a way to be represented in the database. There will be one record in WeightClasses for each record in Fights.

#### *Attributes Types*

- weightClass - int
- gender - varchar(100)

- sport - varchar(100)
- fightNum - int
- cardName - varchar(100)

#### *Attributes Use*

- weightClass - Numeric representation of the weight class.
- gender - The gender that the weight class is for.
- sport - the sport that the weight class is for.
- fightNum - Link to fights, the fight that is under the weight class.
- cardName - Link to fights, the fight that is under the weight class.

### WrestlingStats

The WrestlingStats entity is needed as there should be a way to represent the fighters wrestling stats. These wrestling stats are used to generate the statistics produced by the database. There should be one record in WrestlingStats for each record in Fighters.

#### *Attributes Types*

- fighterID - int
- success - float
- attempt - float
- accuracy - computed as  $((\text{attempt} - \text{success}) * 100)$

#### *Attributes Use*

- fighterID - Link to Fighters, a given fighter's striking stats.
- success - The number of successful takedowns a fighter has.
- attempt - Total number of takedowns attempted.
- accuracy - representation of a fighter's offensive wrestling, the percentage of successful to unsuccessful takedowns.