**2020 DB Term Project Report 02組**

Github: <https://github.com/brianchennn/Database_Team_Project>

HackMD: <https://hackmd.io/GFo9xYfMRNCeQgr_60Z1gg?view>

/\*\*\*剩下部分

database： database的maintain(應該不會再有其他更新?)

# Data

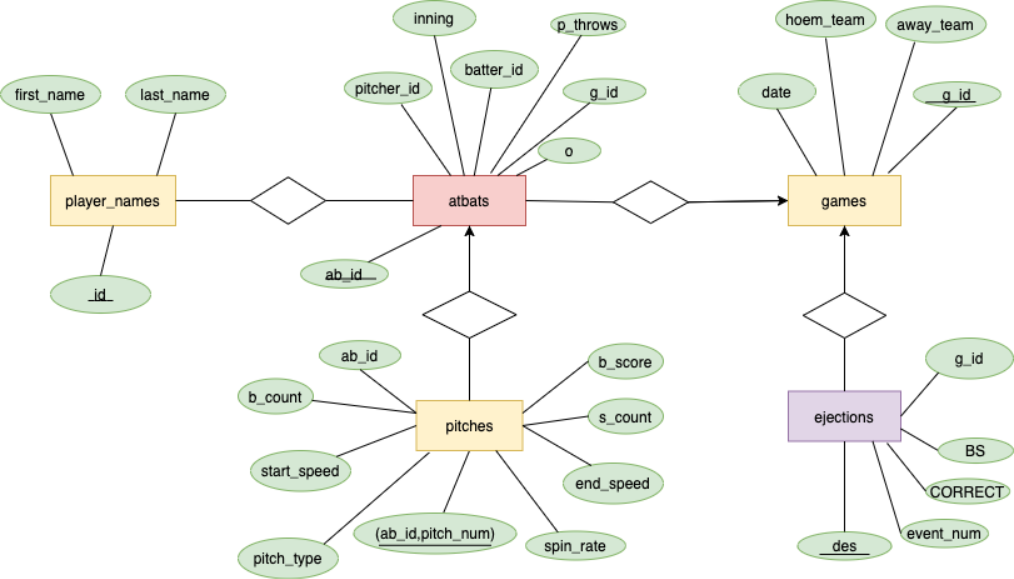
### ****Source****

[Kaggle Pitch Data 2015-2018](https://www.kaggle.com/pschale/mlb-pitch-data-20152018#ejections.csv)

### ****Five Tables From Kaggle****

* **games**
  + **attendance** number of fans who attended (NOTE: for first game of doubleheaders, value is often erroneously 1 or 0. This comes directly from XML files. This data may not be recorded for those games; MLB gameday pages do not report attendance for these game)
  + **away\_final\_score** final score for the visiting team
  + **away\_team** three letter abbreviation for away team; third letter often indicates league(national vs american)
  + **date** date of game
  + **elapsed\_time** length of game, in minutes
  + **g\_id** game ID. Matches with game\_id in atbats.csv
  + **home\_final\_score** final score for the home team
  + **home\_team** three letter abbreviation for home team; third letter often – - indicates league(national vs american)
  + **start\_time** start time of game
  + **umpire\_1B** umpire of 1B
  + **umpire\_2B** umpire of 2B
  + **umpire\_3B** umpire of 3B
  + **umpire\_HP** umpire of HP
  + **venue\_name** name of stadium
  + **weather** description of weather
  + **wind** description of wind
  + **delay** length of delay before game, in minutes
* **ejections**
  + **ab\_id** foreign key for atbats.csv, may be unreliable (ejection happened before, after, during atbat
  + **des** Human readable, in format
  + **event\_num** event number for ejection (from xml file; many event\_nums are skipped)
  + **g\_id** foreign key for games.csv
  + **player\_id** foreign key for player\_names.csv
  + **date** directly from games.csv
  + **BS** ‘Y’ if ejection was for arguing balls and strikes, empty otherwise
  + **CORRECT** Whether the ejection was correct (only for BS ejection). From [closecallsports.com](http://closecallsports.com/)
  + **team** team for player ejected
  + **is\_home\_team** whether that team is the home team-
* **pitches.** (Pitch-level data, including lots of information about the trajectory of the pitch. Match up with atbats.csv for complete picture of game situation. Data comes from unlabeled xmls from MLB website, so the meaning of some fields is not clear.)
  + **px** x-location as pitch crosses the plate. X=0 means right down the middle
  + **pz** z-location as pitch crosses the plate. Z=0 means the ground
  + **start\_speed** Speed of the pitch just as it’s thrown
  + **end\_speed** Speed of the pitch when it reaches the plate
  + **spin\_rate** The pitch’s spin rate, measure in RPM
  + **spin\_dir** Direction in which pitch is spinning, measured in degrees
  + **break\_angle**
  + **break\_length**
  + **break\_y**
  + **ax**
  + **ay**
  + **az**
  + **sz\_bot**
  + **sz\_top**
  + **type\_confidence** Confidence in pitch\_type classification. Goes up to 2 for some reason.
  + **vx0**
  + **vy0**
  + **vz0**
  + **x**
  + **x0**
  + **y**
  + **y0**
  + **z0**
  + **pfx\_x**
  + **pfx\_z**
  + **nasty**
  + **zone**
  + **code** Records the result of the pitch. See dataset description for list of codes and their meaning
  + **type** Simplified code, S (strike) B (ball) or X (in play)
  + **pitch\_type** Type of pitch. See dataset description for list of pitch types
  + **event\_num** event number, used for finding when exactly ejections happen.
  + **b\_score** score for the batter’s team
  + **ab\_id** at-bat ID. Matches up with atbats.csv
  + **b\_count** balls in the current count
  + **s\_count** strikes in the current count
  + **outs** number of outs (before pitch is thrown)
  + **pitch\_num** pitch number (of at-bat)
  + **on\_1b** True if there’s a runner on first, False if empty
  + **on\_2b** True if there’s a runner on second, False if empty
  + **on\_3b** I don’t know
* **atbats.** (This file lists the information that cannot change over the course of an at-bat)
  + **ab\_id** at-bat ID. First 4 digits are year. Matches with ab\_id in pitches.csv
  + **batter\_id** player ID of the batter. Given by MLB, player names found in player\_names.csv
  + **event** description of the result of the at-bat
  + **g\_id** game ID. First 4 digits are year
  + **inning** inning number
  + **o** number of outs after this at-bat
  + **p\_score** score for the pitcher’s team
  + **p\_throws** which hand pitcher throws with. Single character, R or L
  + **pitcher\_id** player ID of the pitcher. Given by MLB, player names found in player\_names.csv
  + **stand** which side batter hits on. Single character, R or L
  + **top** True if it’s the top of the inning, False if it’s the bottom
* **player\_names** (Matches names with player’s ID)
  + **id** matches with batter\_id and pitcher\_id
  + **first\_name** first name
  + **last\_name** last name

### ****ER Diagram****



### Other Tables Extend From The Tables Above

* **batter\_create\_table\_per\_game2** (每場比賽各打者的各種資料)
  + **years** 年度
  + **id** 對應player\_id
  + **first\_name** first name
  + **last\_name** last name
  + **g\_id** game ID
  + **PA** 擔任打席次數
  + **AB** 打數
  + **Single** 一壘安打數
  + **DDouble** 二壘安打數
  + **Triple** 三壘安打數
  + **HR** 全壘打數
  + **K** 三振次數
  + **BB** 四壞球次數
  + **HBP** 觸身球次數
  + **IBB** 故意四壞球次數
  + **SF** 高飛犧牲打次數
  + **GDP**滾地球雙殺次數
  + **Date** 日期
* **batter\_create\_table\_per\_year** (每年度各打者的各種資料)
  + **years** 年度
  + **id** 對應player\_id
  + **first\_name** first name
  + **last\_name** last name
  + **PA** 擔任打席次數
  + **AB** 打數
  + **Single** 一壘安打數
  + **DDouble** 二壘安打數
  + **Triple** 三壘安打數
  + **HR** 全壘打數
  + **K** 三振次數
  + **BB** 四壞球次數
  + **HBP** 觸身球次數
  + **IBB** 故意四壞球次數
  + **DP**滾地球雙殺次數
  + **AVG** 打擊率
  + **OBP** 上壘率
  + **WOBA** 打者加權指數
  + **SLG** 長打率
  + **OPS** 整體攻擊指數
  + **BABIP** 場內安打率
  + **SF** 高飛犧牲打次數
  + **ROE** 野手失誤次數
* **pitcher\_create\_table\_per\_game** (每場比賽各投手的各種資料)
  + **years** 年度
  + **id** 對應player\_id
  + **first\_name** first name
  + **last\_name** last name
  + **g\_id** game ID
  + **Single** 一壘安打數
  + **DDouble** 二壘安打數
  + **Triple** 三壘安打數
  + **HR** 全壘打數
  + **IP** 投球局數
  + **Pitch\_per\_Game** 用球數
  + **K** 三振數
  + **BB** 保送數
  + **HBP** 觸身球數
  + **DP** 雙殺數
  + **Ground** 滾地球數
  + **Fly** 飛球數
  + **ground\_fly\_ratio** 滾飛比(滾地球跟飛球的比例)
  + **date** 日期
* **pitcher\_create\_table\_per\_year2** (每年度各投手的各種資料)
  + **years** 年度
  + **id** 對應player\_id
  + **first\_name** first name
  + **last\_name** last name
  + **IP** 投球局數
  + **pitch\_num** 用球數
  + **Single** 一壘安打數
  + **DDouble** 二壘安打數
  + **Triple** 三壘安打數
  + **HR** 全壘打數
  + **H9** 平均9局被安打次數
  + **K** 三振數
  + **K9** 平均9局三振次數
  + **BB** 保送數
  + **BB9** 平均9局保送次數
  + **IBB** 故意四壞球次數
  + **HBP** 觸身球數
  + **DP** 雙殺數
  + **Ground** 滾地球數
  + **Fly** 飛球數
  + **ground\_fly\_ratio** 滾飛比(滾地球跟飛球的比例)
  + **FIP** FIP
  + **BABIP** 場內被安打率
  + **WHIP** 每局被上壘率
* **pitch\_type\_create\_table\_per\_game** (每場比賽各投手的各球種資料)
  + **years** 年度
  + **id** 對應player\_id
  + **first\_name** first name
  + **last\_name** last name
  + **g\_id** game ID
  + **pitch\_type** 球種
  + **use\_count** 使用次數
  + **use\_ratio** 使用率
  + **strike\_count** 好球次數
  + **strike\_ratio** 好球率
  + **v0\_avg** 初速平均
  + **v\_delta\_avg** 速度變化平均
  + **spin\_rate\_avg** 轉速平均
  + **Date** 日期
* **pitch\_type\_create\_table\_per\_year** (每年度各投手的各球種資料)
  + **years** 年度
  + **id** 對應player\_id
  + **first\_name** first name
  + **last\_name** last name
  + **pitch\_type** 球種
  + **use\_count** 使用次數
  + **use\_ratio** 使用率
  + **strike\_count** 好球次數
  + **strike\_ratio** 好球率
  + **v0\_avg** 初速平均
  + **v\_delta\_avg** 速度變化平均
  + **spin\_rate\_avg** 轉速平均
* **team\_final\_score** (所有對戰的比分)
  + **g\_id** game ID
  + **date** 日期
  + **year** 年度
  + **home\_team** home team
  + **away\_team** away team
  + **home\_final\_score** 主隊最終分數
  + **away\_final\_score** 客隊最終分數
* **team\_region\_status** (所有年度, 所有聯盟分區的戰績)
  + **year** 年度
  + **team** 隊伍
  + **League** 聯盟
  + **Division** 分區
  + **win** 勝場
  + **total** 總比賽場次
  + **win\_rate** 勝率
* **team\_opponent** (某球隊每場比賽的相關資訊)
  + **g\_id** game ID
  + **home\_or\_away** 該隊伍在這場比賽是主隊或客隊
  + **opponent** 對手
  + **score** 該隊伍在這場比賽的分數
  + **opponent\_score** 對手在這場比賽的分數
* **team\_first\_inning\_run\_ratio** (所有球隊每一年的首局得分率，首局總得分數)
  + **year** 年度
  + **team** 隊伍
  + **total\_game** 比賽場次數
  + **scoring\_rate** 首局得分率
  + **total\_score** 首局總得分數
* **team\_LLRR** (左投或右投對左打或右打的打擊率)
  + **year** 年度
  + **pitcher\_batter\_stand** 投手和打者的站位
  + **cnt\_baserun** 安打數
  + **cnt\_atbat** 打數
  + **AVG** 打擊率
* **ejection\_game**(單場比賽驅逐出場次數大於等於3次的場次)
  + **date** 日期
  + **g\_id** game ID
  + **home\_team** home team
  + **away\_team** away team
  + **ejection\_cnt** 驅逐出場次數
* **ejection\_max\_player**(每年度驅逐出場次數最多的選手)
  + **year** 年度
  + **id** 對應player\_id
  + **first\_name** first name
  + **last\_name** last name
  + **cnt** 驅逐出場次數
* **ejection\_team**(每年度所有隊伍的驅逐出場次數，若次數為0的話不會顯示在table裡)
  + **year** 年度
  + **team** 隊伍
  + **cnt** 驅逐出場次數

### ****Data Normalization****

我們為了分析更多資料於是由原始的table中再延伸了許多table，而過程中因想增加可識別度及方便被其他table使用的關係，所以有滿多table都有部分功能相依的狀況(如主鍵為<year, id>時，first\_name及last\_name只相依於id)，因此我們的data在包含新增的table的情況下並沒有滿足2NF

# Database

### ****Database We USE****

MySQL

### ****How Do We Maintain Our Database****

因為使用者在查詢資料不會想花很多時間，所以我們建很多臨時table。

分為

打者單場數據、投手單場數據、打者全年數據、投手全年數據、投球球種全年數據、投球球種單場數據。

單場數據用(g\_id,id)當primary key

全年數據用(years,id)當primary\_key

有了這些key，在新建table時會快很多

我們用單場數據的加總產生全年數據，但不是用create\_table\_per\_game產生cerate\_table\_per\_year，因為在sum的過程非常耗時。

所以是先把小的\*\_per\_game sum成\*\_per\_year。

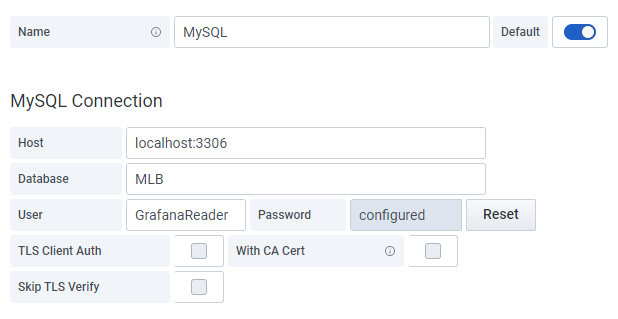
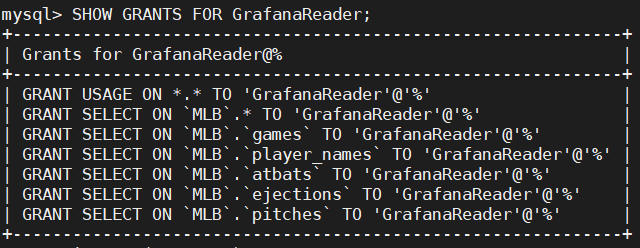
### ****How Do We Connect Our Database To Our Application****

# Application

### ****Interface****

我們使用 Grafana 的 Open Source資料視覺化平台，架在宿舍電腦中的虛擬機中，可以從固定 IP 訪問，亦有申請域名如下。<http://mlb.nctu.me:3000>

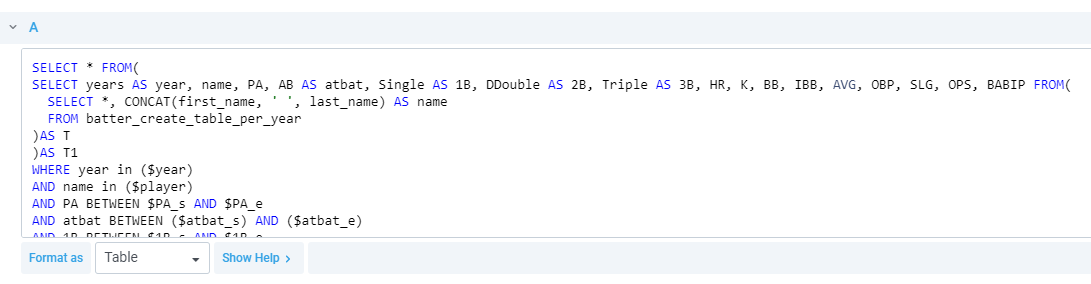
Database 如上述說明，是使用 MySQL，也是架在同一台虛擬機中，藉由 3306 Port ，並開一個唯獨的使用者 GrafanaReader，讓 Grafana 能夠執行 。



如此一來，Grafana 即可讀取 MySQL 的資料。

在 Grafana 內部，我們使用兩種方法進行 query。

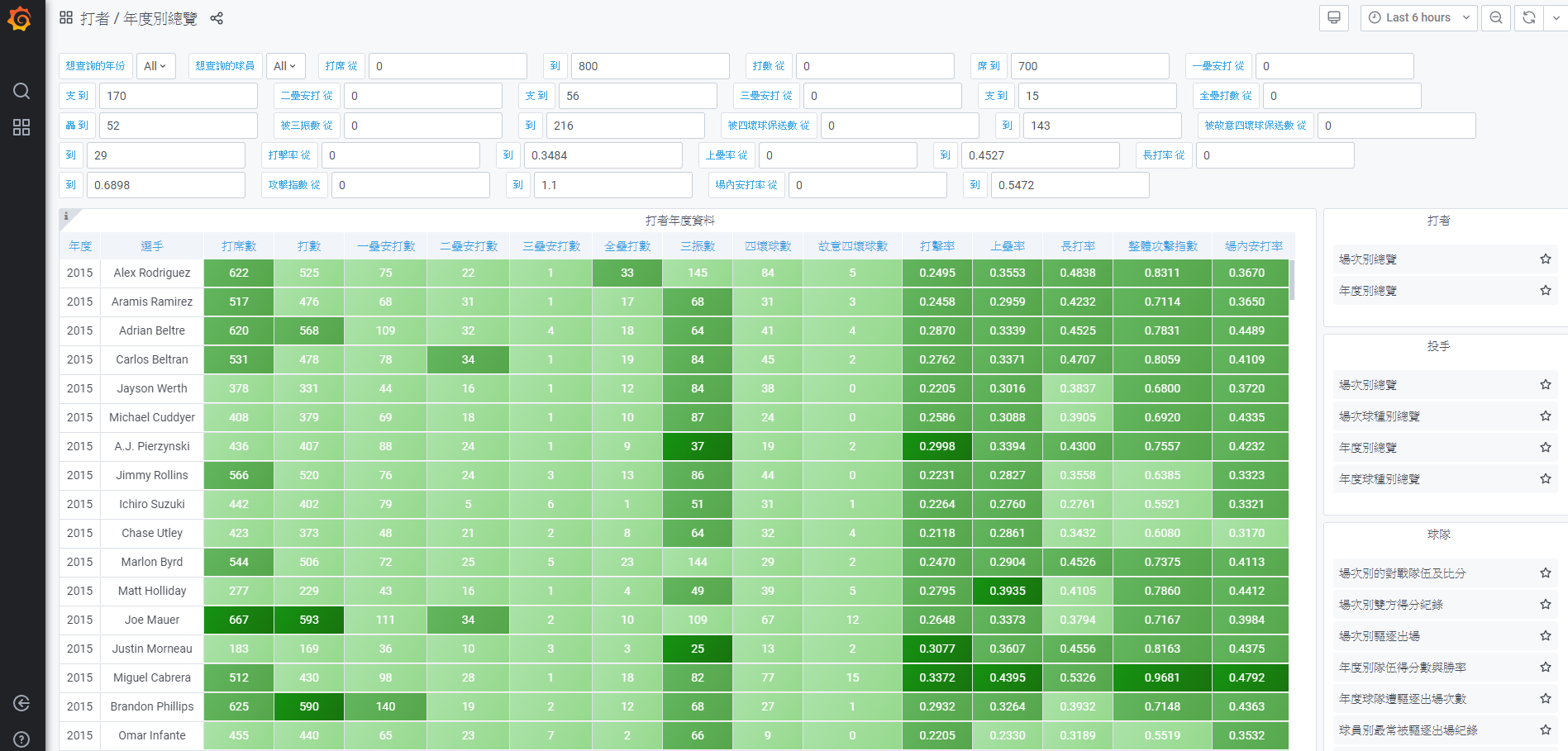
第一種是針對運算量較高的 query，我們會先在 MySQL 跑過一次後，建立相對應的 Table，在網站上再針對特定年分、選手、勝率等資料進行 query，方法如下。



圖中的 batter\_create\_table\_per\_year 是一個已經建立過的 table，我們再根據使用者輸入的數值，對這個 table 進行 query (第二次 query)以滿足使用者進一步縮小範圍的需求，又能避免 query花費過長時間。

### ****Function****

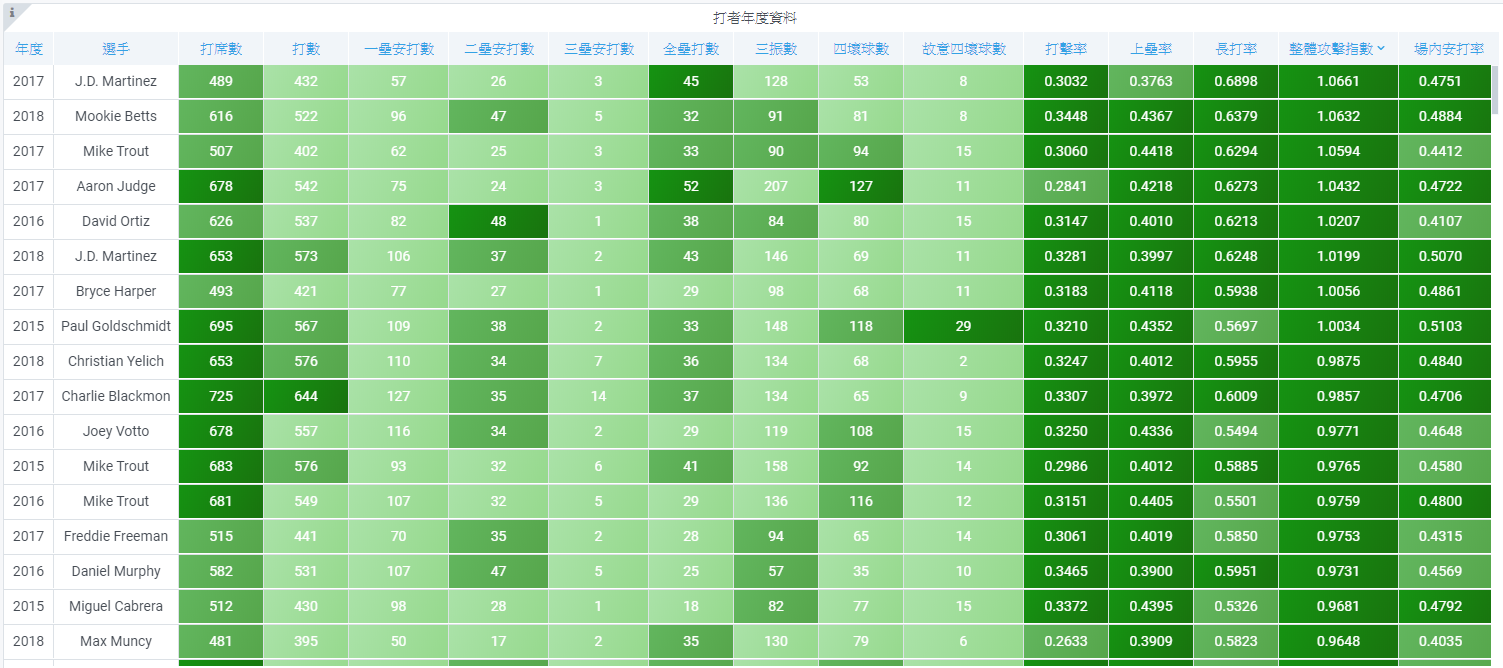
接下來展示面向使用者的UI，當連上[網站](http://mlb.nctu.me:3000)後會出現的主畫面如下。



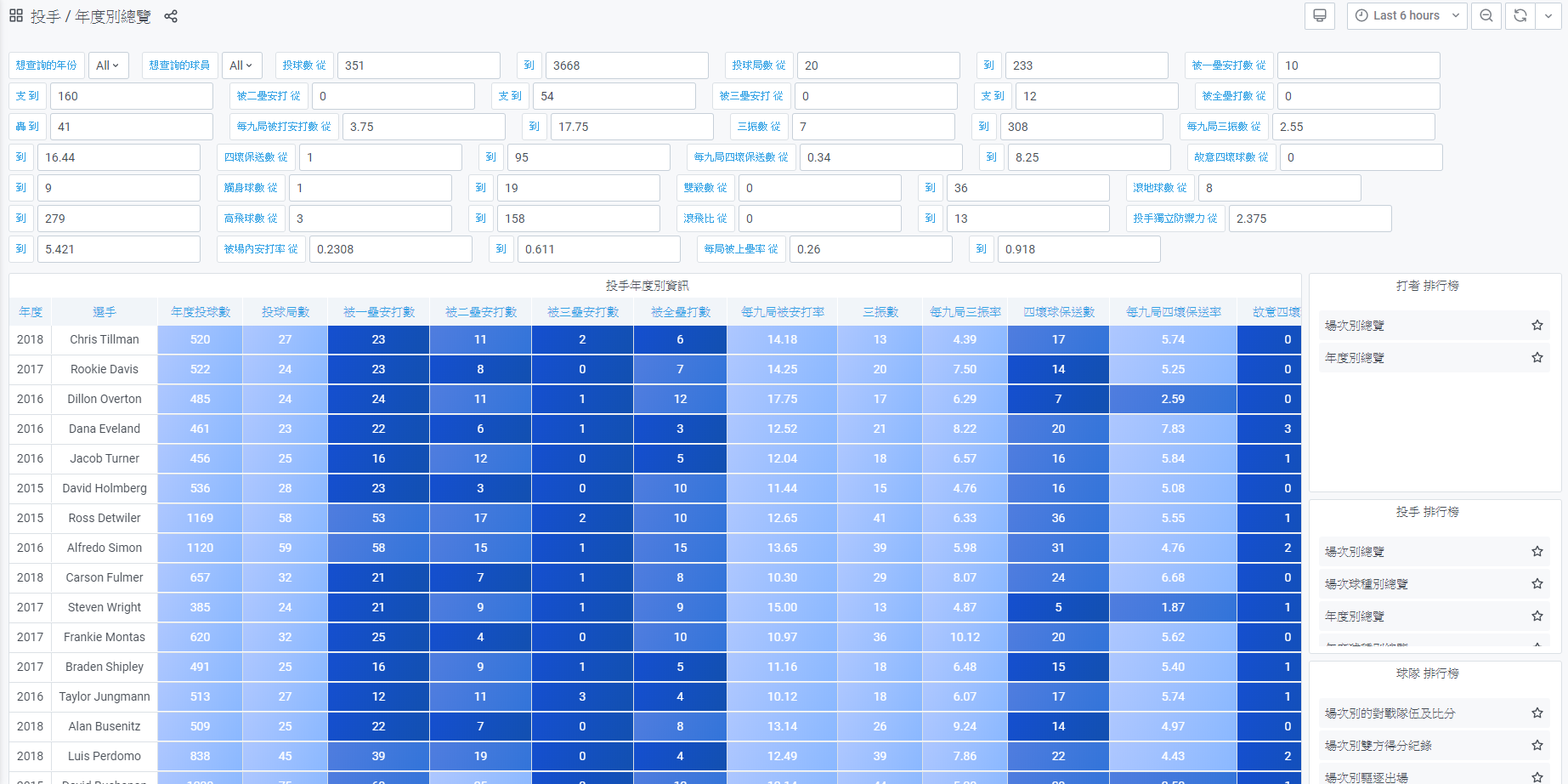
主畫面中可以看到上方是輸入欄，中間最大一塊是呈現數據的 Table，最右邊則是連結到不同類別的 Query，其他畫面的排版跟主畫面皆相同。



圖片中央的 Table 會先列出我們的 Query 結果，並根據數值進行視覺化上色、改動，如圖中會根據每個數值的差到優、低到高進行標色淺到深， PR 60 與 PR 85 是我們預設的變色點，在打者部分，分別會是 淺綠 < 綠 < 深綠，應用上來說，假如點擊「整體攻擊指數」，系統會根據該數值進行排序，根據點擊次數，會在高到低、低到高、取消，三個狀態之間改變，如圖片中是根據整體攻擊指數進行高到低排序，輔以顏色標記後，我們可以看到整體攻擊指數高的球員，在打擊率、上壘率、長打率、場內安打率都是有較佳的表現(多為深色)。



打者區是以綠色為基底。



投手區是以藍色為基底。



其他區會以適合的方式進行視覺化，如勝場數排行、得分比較等。

而主畫面上方是提供使用者縮小範圍的輸入欄。



這些數值都是以變數寫入上述第二次 query 內，會根據使用者輸入的數值不同，進行相對應的 query，例如使用者可以輸入球員姓名，輸入欄相對應的球員就會被反白出來，避免使用者輸入的球員不在我們的資料集中或是打錯字，算是一種防呆機制。

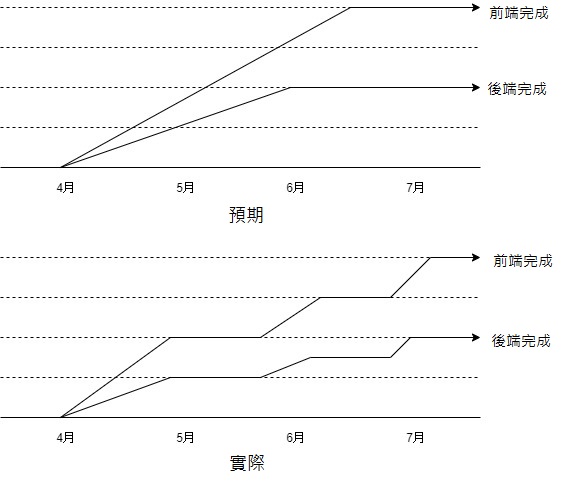


選取完之後，系統會將使用者選取的數值代入上述說的第二次 Query 中相對應變數的地方，然後刷新 Table，如下圖，可以見到原本很多 row 的 table，只剩下選取的球員的數據，其他不符條件的數據都會被屏除，應用上可以讓使用者查詢自己關注的球員、年分、數值排行等，而不會因龐大的數據量而分散注意力。

欄位中都已經預先手動填入所有數值的 max 和 min，因此使用者可以知道數值的分布範圍，可以避免填入偏離資料集的數值而查不到相對應的數據。

# Others

### ****Progress****

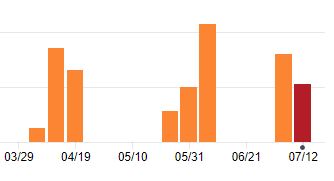


我們在4/21訂出題目並找好source，4/23建好基礎的table

計畫在6月中前完成後端開發並與前端連結，6月底前就能完成project

但實際上因段考期間大家都很忙，中途斷斷續續的進度停滯了一段時間

最後收尾及驗收的部分到了7月才漸漸完成



🡩從Github的commit數量也能看得出來

### ****Problems and Solves****

1. 命名問題：因為batter跟pitcher的query是不同人寫的，命名方式有些不同

🡪 後續要不斷確認這個column的名稱，而且不能亂改。

1. Create table速度：太多table join在一起會讓query速度極慢，可能吃完晚餐都還沒好

🡪 這時用alter table /\*table\_name\*/ add primary key(/\*column\_name\*/);就能加快非常多速度。

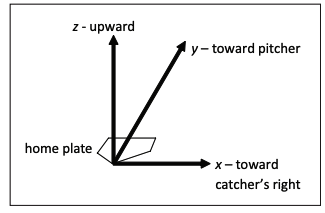
1. 在group by時，如果先篩選出特定row，比如說某打者打一壘安打的row，先篩選，再對打者id做group by並count(\*)，會沒辦法顯示一壘安打是0的打者id，因為已經被篩掉了

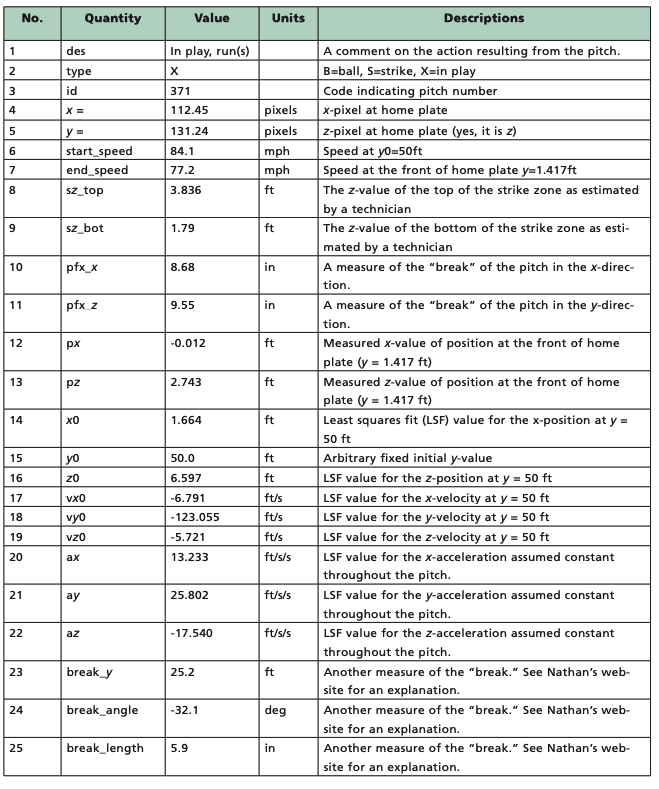
🡪 這時可用if(event=”Single”,1,0) as Single的方式，group by時再用sum(Single)，來正確顯示出0支1壘安打的打者id。

或者先把所有打者id列出來後與一壘安打left join，再用ifnull把count換成0

1. 原始資料意義不明：pitcher這個table紀錄了每球的很多數據，但是某column定義不明，導致我們想要查縱向位移量時，不知道要用哪個column

🡪 我們找到了這篇論文，下面兩張圖是column的單位及定義(<http://baseball.physics.illinois.edu/KaganPitchfx.pdf>)



但是y,z方向的定義很亂。目前正由組員王昶淵研究中。

1. 比較ratio時可能會有data筆數過少導致數值過度異常

🡪 把data筆數也加入條件篩選

1. 無法確認sql的結果是否正確

🡪 與其他現有網站交叉比對

1. data豐富度比不上其它現有網站

🡪 分析其他有趣的data(如驅逐出場次數等)

### ****Contribution****

陳煜盛：

唯一棒球迷，需要解釋術語、傳統、負責計劃研究方向、指派工作。

負責batter\_\*.sql的編寫及pitcher\_\*\_per\_game的部分。

整合proposal PPT與口頭報告所有後端部分。

洪瑋廷：

唯一前端負責人，負責Grafana的排版設計、query 資料根據使用者需求進行二次 query 並視覺化。

維護架設 MySQL 和 Grafana 的 Server。

Proposal負責口頭報告及展示Grafana前端頁面。

影片拍攝。

李嘉盛：

Teams\_\*及其他類query的撰寫

完成陳煜盛指派的任務

報告的撰寫。

王昶淵：

Pitcher\_\*.sql的撰寫

負責完成陳煜盛指派的任務

研究pitches column的物理意義。