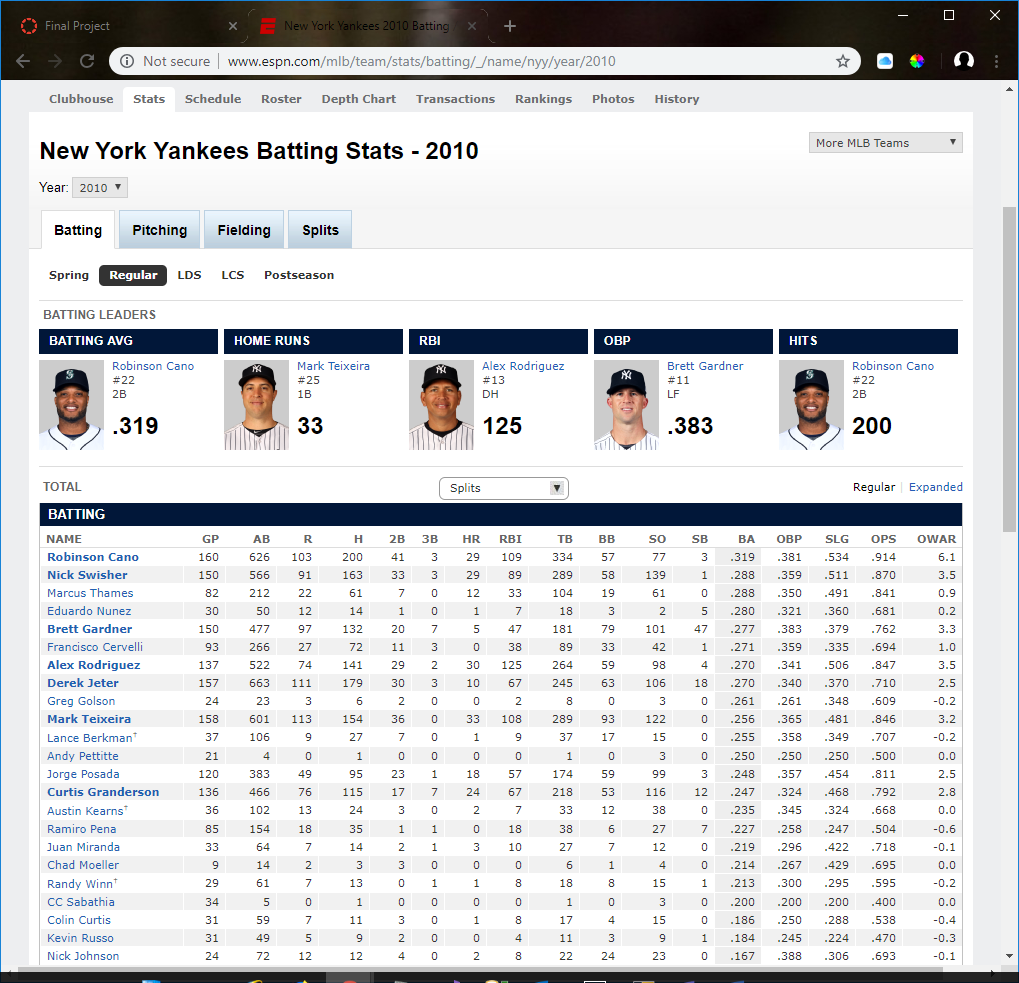
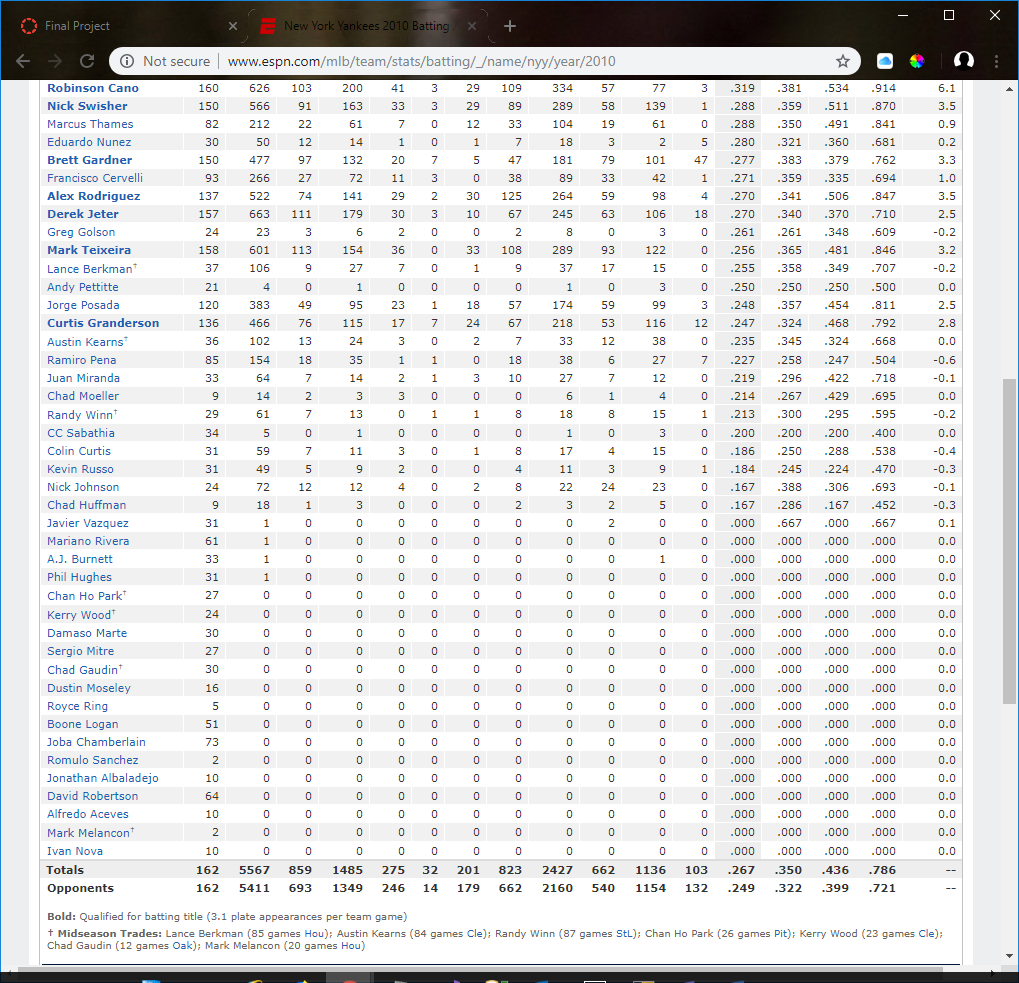
**Final Project  
CIS126 – SQL  
*100 Points***

You will likely be amazed as to what you can actually do with your newly honed SQL skills. ESPN maintains the same data as we do in our Baseball database. To demonstrate your skills you will l be recreating the data portion of the 2010 New York Yankees data as seen on:

http://www.espn.com/mlb/team/stats/batting/\_/name/nyy/year/2010





***Note: You can select any team and/or year that you can select off the linked page. Do not pick dates after 2015 (which I think is the latest data we maintain).***

* Albeit long, the grid body is a simple SELECT with several calculated columns. About 50% of the fields can be plugged into the SELECT directly (i.e. GP, AB, R, H, …) from existing columns.
* We will not focus on unrelated text, images or formatting data – just the data that will later be applied to a web page by a web developer. Keep in mind formatting does not matter from a data perspective. It is the responsibility of the programmer to format the data you provide. You will likely lose points if you format.
* You will not be creating the Opponents row (bottom) or the OWAR column (right). I have provided a hints page with the formulas for the remaining columns.
* I highly recommend you complete this the easy way – one column, left to right, at a time.
* You should provide 7 select statements:
  + Top 5 batting leaders (do not worry about player number, photo, or position).
  + Body of the grid. I expect the same column headers as ESPN. Name should be formatted the same in a single column.
  + The data in the body is sorted by batting average.
  + The total row (your totals might differ *slightly* depending on how you base your calculations).
* Be careful and read the hints carefully. Division by zero errors will be what you will need to avoid. I am providing you with instructions on how to implement the TSQL Case statement you might appreciate.
* Be careful with the hits column. It is misleading. In reality, the “Hits” column is a total of singles + double+triples+home runs. It is not equal to singles. In my suggestions I recommend you modify your vwPlayersBatting view to include a calculated singles column (Singles = Hits-2B-3B-HR). It will simplify your work.

**Hints and Instruction:**

*Players/Batting (Batting Tab)*

* For a Batting Average or OBP leader (two of the top 5 SELECTS) to qualify, they should have batted (AB) at least ***500*** times in a season. This prevents a pitcher with 1 at bat and 1 hit from being a leader with a batting average of 100%. But we will find some teams where a leader was not at bat 500 times. The 2010 Yankees had only 5 players at bat 500 times or more. But they list Bret Gardner as OPB leader and he was at bat 477 times. There is an obscure formula to determine team leaders if they were at bat less than 500 times but I can’t find it published. You can either just give me the overall leader simply based on best OBP or for OBP only set minimum games to 400. Home Run, RBI and Hits leaders do not take into account number of games so they are simply a sum.
* There are 162 games in a season. You will need this in calculations.
* Start by building a view between players and batting (for batting tab) if you have not already. We want all players even if they didn’t bat (*Left Outer Join*).
* I highly recommend you create a calculated (S)ingles column to represent Singles. You will need singles in some formulas. Otherwise you can represent singles as *Hits-doubles-triples-hr*. My hints below will be based on a calculated Singles column existing.
* Look at the data. We are pulling for 2010. The basic (body) table data should be a simple SELECT based on our view.
* Start building the table left to right. Do one column at a time. You might have to research some of the abbreviations (<http://mlb.mlb.com/mlb/official_info/baseball_basics/abbreviations.jsp>). For example, on the ESPN site GP is games played whereas it is simply G in our table. Most match. Disregard the last OWAR column.
* TB (Total bases) is a total of the bases hit. For example, a single is worth 1 base whereas a homerun is worth 4. In other words, a single is worth 1 TB, a double is 2 TB, etc. If a player hit 5 singles, 3 doubles, a triple and two home runs, his TB is calculated as:

5 x 1 singles = 5

3 x 2 doubles

1 x 3 triple = 3

1 x 4 home run = 4

18 TB

* We have already visited how to calculate a percentage (*part/whole*). The Batting average is total hits (H) divided by At-bats. The challenge is that the number is rounded to 0 as an integer. Handle this accordingly.

(H \*1.0)/AB

We do have a slight challenge with BA (Batting Average) is a player, such as a pitcher, never batted. You will get a division by 0 error. You can eliminate these by including in your WHERE statement:

WHERE AB>0

**But** if we do that we will eliminate pitchers that never batted, which we want. If you want pitchers included, you will have to use a special CASE statement. If we treat no At-Bats as 1 rather than 0, any number divided by 1 is itself and we will not have a division by zero error. You could use this as your BA column,

H/(Case when AB=0 then 1 else AB end) as 'BA'

The TSQL case statement says, *if, when you divide, the AB is 0, replace it with 1*. This prevents division by 0 errors. Any number divided by 1 is the same number. Remember to deal with the *integer/integer* returning an *integer* issue as I described above. Do not worry about the fact your answers will be more precise than ESPN’s. The programmer will format the data.

* On Base Percentage (OBP)

(a) *Hits (H + Walks (BB) + Times Hit by Pitch (HBP)* This sum is the total Times on Base

(b) *Calculate Total Plate Appearances by adding At Bats (AB), Walks (BB) and times Hit by Pitch (HBP)*

(c) Then divide (a) Times on base by (b) Total Plate Appearances

You will have to deal with the division by zero error as we previously discussed.

* Slugging Percentage (SLG)

Total Bases *divided by* At-Bats.

* On Base plus Slugging (OPS)

On Base Percentage plus Slugging Percentage

* Note that the body is sorted by Batting Average high to low.
* Once you have the data generated, you can produce the table footers for ONLY totals. Disregard the opponent’s row. This will be a separate SELECT statement. I suggest you simply sum the accumulated total columns such as B, H, S, B2, B3, etc. GP will be hard coded 162. The remaining columns are based on the columns you summed so you should not need to GROUP. Use the same formulas for the calculated columns you did in the body. Rather than referring to B2 in your formula, for example, you would use sum(B2). Your numbers will only match if you perform the same average formulas on the columns as you did in the body SELECT. In other words, **do not average the averages**.

***Extra Credit: 5 Points***

Turn your homework in as a Stored Procedure named spGetTeam. When I enter “***spGetTeam***” followed by a teamid and a year:

*spGetTeam ‘Sea’, 2010*

I get that team’s data for that respective year. Include only the body data of the table.

**You MUST tell me to look for your stored procedure at submission else no credit.**