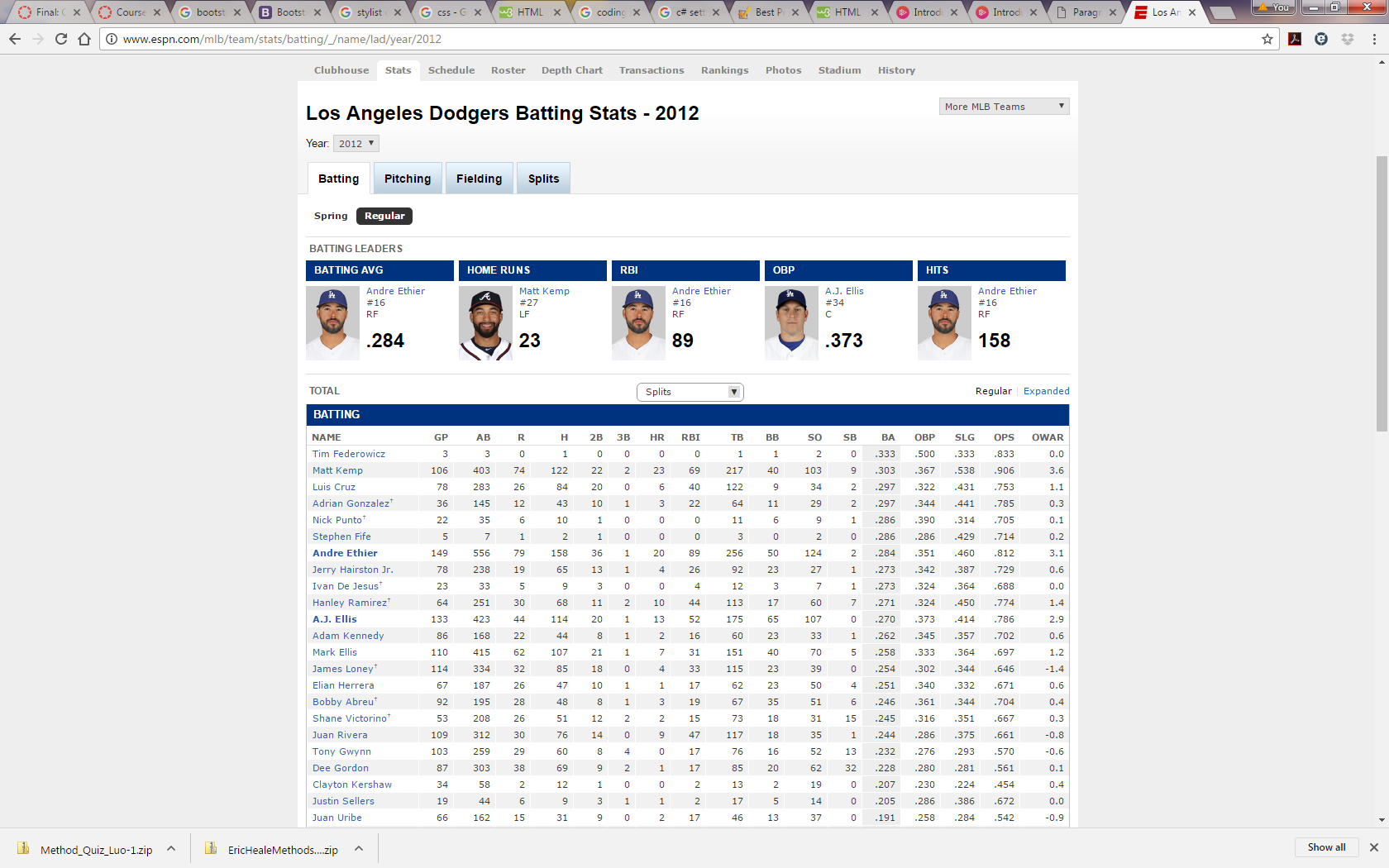
**Final Project  
CIS126 – SQL  
*100 Points***

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 2012 Los Angeles Dodgers Data as seen on:

<http://www.espn.com/mlb/team/stats/batting/_/name/lad/year/2012> 



* 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,…).
* We will not focus on unrelated text or images – 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.
* 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 columns.
* I highly recommend you complete this the easy way – one column at a time.
* You should provide 7 select statements:
  + Top 5 batting leaders (do not worry about player number or position)
  + Body of the grid. I expect the same column headers as ESPN. Name should be formatted the same in a single column.
  + Dodger’s 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 create a view and create a calculated singles column (Singles = Hits-2B-3B-HR). It will likely simplify your work.

**Hints and Instruction:**

*Players/Batting (Batting Tab)*

* Start by building a view between players and batting (for batting tab) if you have not already. I highly recommend you create a calculated (S)ingles column to represent Singles if you have not already. 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.
* Look at the data. We are pulling for 2010. The basic 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 which we might 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.

Normally you would not worry about formatting in SQL. That is the programmer’s responsibility. In the case of ESPN, their data grid is formatted the output to 3 decimal places - do not worry about that.

* 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 sum (a) into (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

* Once you have the data generated, you can implement your grouping to produce the table footers for ONLY totals. Disregard the opponent’s row. This will be a separate SELECT/Group by statement. To produce the output you would run both statements together. I suggest you simply sum the accumulated total columns such as B, H, S, B2, B3…. Your numbers will only match if you perform the same average formulas on the columns as you did above. In other words, do not average the averages. Calculate averages from the totals.

***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.**