```
SELECT playerID,
           yearID,
           teamID,
           (0.713*(BB-IBB) + 0.742*HBP + 0.898*(H-2B-3B-HR) + 1.257*2B +
1.580*3B + 2.007*HR) / (AB + BB - IBB + SF + HBP) AS wOBA
FROM Batting
WHERE yearID = 1998 AND AB > 300
ORDER BY wOBA DESC
```

TECH 1

Q2

```
SELECT playerID, SO AS Strikeouts, BB as Walks, HR as HomeRuns,
((3*BB +3*HBP + 13*HR - 2*SO)/(IPOuts/3) + 3.132) AS FIP
          FROM Pitching
          WHERE yearID = 2014;
```

Q4

```
SELECT playerID, (BB + SO + HR)/(AB + BB + SF + SH + HBP) AS
TTOPercentage
          FROM Batting
          WHERE yearID = 2000 AND AB >= 500
          ORDER BY TTOPercentage DESC
```

TECH 3

```
SELECT playerID,
    (H/AB) AS BA,
    (H+BB+HBP)/(AB+BB+HBP+SF) AS OBP,
    (H+2B+2*3B+3*HR)/AB AS SLG,
    ((H+BB+HBP)/(AB+BB+HBP+SF)) + ((H+2B+2*3B+3*HR)/AB) AS OPS
FROM Batting
WHERE yearID = 1969 AND AB > 50
HAVING OPS > 0.8
ORDER BY OPS
```

Q1

```
SELECT playerID,
    (H/AB) AS Batting_AVG,
    (H+BB+HBP)/(AB+BB+HBP+SF) AS OBP,
    (H+2B+2*3B+3*HR)/AB AS SLG,
    ((H+BB+HBP)/(AB+BB+HBP+SF)) + ((H+2B+2*3B+3*HR)/AB) AS OPS
FROM Batting
WHERE teamID = 'SEA' AND yearID = 2001 AND AB > 100
ORDER BY OPS
```

Q2

```
SELECT teamID, yearID,
    MAX(HR) AS Leading_HR
FROM Batting
GROUP BY teamID
HAVING SUM(HR) > 45
```

```
ORDER BY Leading_HR DESC, yearID
```

Q3

```
SELECT playerID,
    (H/AB) AS BA,
    (H+BB+HBP)/(AB+BB+HBP+SF) AS OBP,
    (H+2B+2*3B+3*HR)/AB AS SLG,
    ((H+BB+HBP)/(AB+BB+HBP+SF)) + ((H+2B+2*3B+3*HR)/AB) AS OPS
FROM Batting
WHERE teamID = 'SEA' AND yearID = 2001 AND AB > 100
ORDER BY OPS
```

Q4

```
SELECT (((H+BB+HBP)/(AB+BB+HBP+SF)) + ((H+2B+2*3B+3*HR)/AB)) AS OPS
FROM Batting
WHERE playerID = 'kershcl01' AND yearID = 2014


SELECT playerID,
    (H/AB) AS BA,
    (H+BB+HBP)/(AB+BB+HBP+SF) AS OBP,
    (H+2B+2*3B+3*HR)/AB AS SLG,
    ((H+BB+HBP)/(AB+BB+HBP+SF)) + ((H+2B+2*3B+3*HR)/AB) AS OPS
FROM Batting
WHERE yearID = 1969 AND AB > 50
HAVING OPS > (
    SELECT (((H+BB+HBP)/(AB+BB+HBP+SF)) + ((H+2B+2*3B+3*HR)/AB)) AS OPS
    FROM Batting WHERE playerID = 'kershcl01' AND yearID = 2014)
ORDER BY OPS DESC


SELECT ((SO-BB)/BFP) AS KBB
FROM Pitching WHERE playerID = 'kershcl01' AND yearID = 2014 (0.2777)

SELECT playerID, yearID, teamID,
    ((SO-BB)/BFP) AS K_Minus_BB
FROM Pitching
WHERE yearID > 1990 AND IPOuts > 450
HAVING K_Minus_BB > (
    SELECT ((SO-BB)/BFP) AS K_Minus_BB
    FROM Pitching WHERE playerID = 'kershcl01' AND yearID = 2014)
ORDER BY K_Minus_BB DESC
```

WEEK3

LECTURE 1 (9)

QUIZ 3

```
SELECT teamID, yearID, R/G
FROM Teams
WHERE yearID = 1968 OR yearID = 2000
ORDER BY R/G, teamID
```

QUIZ 4

Let's now use a new SQL query to find the Teams throughout the Expansion Era that have most outperformed the Run-Scoring Environment that they played in

```
SELECT
           SUM(R)/SUM(G) as LeagueRunAverage,
           yearID,
           (MAX(R/G))/(SUM(R)/SUM(G)) as BestOffenseRatio,
           SUM(R)/SUM(G)*(MAX(R/G))/(SUM(R)/SUM(G)) as
BestOffenseRunAverage
FROM Teams
```

Let's modify the previous SQL query to now find the Teams throughout the Expansion Era that have most underperformed the Run-Scoring Environment that they played in

```
SELECT
           SUM(R)/SUM(G) as LeagueRunAverage,
           yearID,
           (MIN(R/G))/(SUM(R)/SUM(G)) as WorstOffenseRatio,
           SUM(R)/SUM(G)*(MIN(R/G))/(SUM(R)/SUM(G)) as
WorstOffenseRunAverage
FROM Teams
WHERE yearID >= 1961
GROUP BY yearID
ORDER BY BestOffenseRatio DESC
WHERE yearID >= 1961
GROUP BY yearID
ORDER BY WorstOffenseRatio DESC
```

QUIZ 5

Use the query below to find the Teams in the Expansion Era that struck out the most times per game.

```
SELECT
           SUM(SO)/SUM(G) as LeagueKRate,
           (MAX(SO/G)) as HighestKRate,
        yearID
FROM Teams
WHERE yearID >= 1961
GROUP BY yearID
ORDER BY HighestKRate DESC
```

WEEK 3

LECTURE 2

LECTURE 10

QUIZ 4

Let's use the BU SQL Sandbox to investigate the topic of Win Percentage throughout baseball history. Write a query that Selects teamID, yearID, Wins, Losses, Winning Percentage (W/(W+L)), Runs Scored, Runs Allowed, Run Differential (R - RA), and Run Differential Per Game ((R - RA)/G). Alias the Winning Percentage term as Winning_Percentage, the Run Differential term as Run_Differential, and the Run Differential Per Game as Run_Diff_Per_Game. Select this information from the Teams table. Make sure that your results only come from years after (and including) 1901 - the Modern Era. Furthermore, only select teams with Winning_Percentage greater than 70% (.700). You should order your results to start with the Highest Single Season Winning Percentage in the Modern Era. Additionally, add teamID and yearID as the 2nd and 3rd parts of your ORDER BY command (don't use ASC or DESC for either one).

```
SELECT teamID, yearID, W, L,
        (W/(W+L)) AS Winning_Percentage,
        R,
        RA,
        (R - RA) AS Run_Differential,
        ((R - RA)/G) AS Run_Diff_Per_Game
FROM Teams
WHERE yearID > 1900
HAVING Winning_Percentage > 0.7
ORDER BY Winning_Percentage DESC, teamID, yearID
```

Let's create a metric that determines the actual performance of these great teams (measured by Winning Percentage) compared to how Bill James's Pythagorean Formula would have expected them to perform

```
SELECT teamID, yearID, W, L,
        (W/(W+L)) AS Winning_Percentage,
        R,
        RA,
        (R - RA) AS Run_Differential,
        ((R - RA)/G) AS Run_Diff_Per_Game,
        (R*R)/(R*R+RA*RA) AS BillJames,
        ((R*R)/(R*R+RA*RA)) - (W/(W+L)) AS JamesMinusActual
FROM Teams
WHERE yearID > 1900
HAVING Winning_Percentage > 0.7
ORDER BY JamesMinusActual DESC, teamID, yearID
```

WEEK 3
TECH 1 - VIDEO 1

```
SELECT teamID, yearID, W,
        G * (R*R)/(R*R+RA*RA) AS predictedW,
        ((R*R)/(R*R+RA*RA)) - W AS ActualError
FROM Teams
WHERE yearID = 2013
ORDER BY ActualError DESC, teamID, yearID
```

TECH 1 - Q1
Write a query that selects teamID, yearID, Wins (W), Pythagorean Predicted Wins (predictedW), and Pythagorean Error (Error) from the teams table. Include only results for the Dodgers (teamID LAN) for years since and including 1980 and order your results with the greatest positive error first. Remember that pythagorean winning *percentage* is equal to R*R/(R*R + RA*RA) and error is equal to predicted wins minus actual wins. Put games (not W+L) in the front of your multiplication in calculating predicted wins and errors to avoid issues with the SQL Sandbox grader.

```
SELECT teamID, yearID, W,
        G * (R*R)/(R*R+RA*RA) AS predictedW,
        G * ((R*R)/(R*R+RA*RA)) - W AS Error
FROM Teams
WHERE yearID >= 1980 AND teamID = 'LAN'
```

```
ORDER BY Error DESC, teamID, yearID
```

TECH 1 - VIDEO 2

```
SELECT w.yearID, W, AVG(ABS(w.Error)) AS AverageError
FROM (SELECT teamID, yearID, W,
        G * (R*R)/(R*R+RA*RA) AS predictedW,
        G * ((R*R)/(R*R+RA*RA)) - W AS Error
        FROM Teams
        WHERE yearID >= 1950
        ) w
GROUP BY w.yearID
```

TECH 1 - Q2

Modify the nested select query from the previous video to select the w.yearID, MINError, MAXError, STDError for all Pythagorean win predictions from 1955 onwards (do not use the ABS function for any of these). Recall that (R*R)/(R*R + RA*RA) is the formula for pythagorean win *percentage*, and error is defined as pythagorean wins minus actual wins. Order by the highest standard deviation (using the STDDEV function). ROUND your standard deviation to three decimal places using the ROUND function i.e. ROUND(STDDEV(...),3)

We can see from the MINError and MAXError that in some cases, teams greatly over- or under-perform Bill James's Pythagorean prediction. To get the grader to accept your answer as correct, please multiply by games at the *beginning* of your predicted win and error formulas.

```
SELECT w.yearID, MIN(w.Error) AS MINError, MAX(w.Error) AS MAXError,
ROUND(STDDEV(w.Error),3) AS STDDEVError
FROM (SELECT teamID, yearID, W,
        G * (R*R)/(R*R+RA*RA) AS predictedW,
        G * ((R*R)/(R*R+RA*RA)) - W AS Error
        FROM Teams
        WHERE yearID >= 1955
        ) w
GROUP BY w.yearID
ORDER BY STDDEVError DESC
```

TECH 1 - VIDEO 3

```
SELECT CONCAT (nameFirst, " ", nameLast) AS PlayerName,
    G, AB, 2B, 3B, R, RBI, SB,
    H / AB AS BA,
    (H+BB+HBP)/(AB+BB+HBP+SF) AS OBP,
    ((H+2B+2*3B+3*HR)/AB) AS SLG
FROM Batting b
JOIN Master m
    ON b.playerId = m.playerID
WHERE yearID = 2013 AND teamID = 'SEA'
ORDER BY SLG DESC
```

TECH 1 - Q3.1

Write a query that selects ERA and FIP (3*BB + 3*HBP + 13*HR - 2*SO)/(IPOuts/3) + 3.139 from the pitching table for all pitchers on the 1998 Braves (teamID = ATL) with at least 10 games started. Order your results by ERA-FIP, with the largest positive differential first.

```
SELECT ERA, (3*BB + 3*HBP + 13*HR - 2*SO)/(IPOuts/3) + 3.139 AS FIP,
    ((3*BB + 3*HBP + 13*HR - 2*SO)/(IPOuts/3) + 3.139) - ERA AS DIFF
FROM Pitching
WHERE teamID = 'ATL' AND yearID = 1998 AND GS > 10
ORDER BY (ERA - FIP) DESC
```

TECH 1 - Q3.2

Now modify your previous query to include the player's full name (as Name) as the first column. Remember to do a join with the Master Table, on playerID (the common column between the Master table and the Pitching table). Alias the tables as p and m. The code to create a full name is CONCAT(m.nameFirst, ' ', m.nameLast)

```
SELECT CONCAT (nameFirst, " ", nameLast) AS Name, ERA, (-2*SO + 3*BB +
3*HBP + 13*HR)/(IPOuts/3) + 3.139 AS FIP
FROM Pitching p
JOIN Master m
    ON p.playerId = m.playerID
WHERE teamID = 'ATL' AND yearID = 1998 AND GS >= 10
ORDER BY (ERA - FIP) DESC
```

TECH 1 - VIDEO 4

```
SELECT CONCAT (nameFirst, " ", nameLast) AS PlayerName,
    G, AB, 2B, 3B, R, RBI, SB,
    H / AB AS BA,
    (H+BB+HBP)/(AB+BB+HBP+SF) AS OBP,
    ((H+2B+2*3B+3*HR)/AB) AS SLG
FROM Batting b
JOIN Master m
    ON b.playerId = m.playerID
WHERE nameFirst = 'Miguel' AND nameLast = 'Cabrera'
```

TECH 1 - Q4

Modify your last query, selecting the same columns with the same ordering, plus yearID as column 2, but for all Pedro Martinez seasons. Do this by searching for his NameLast and NameFirst, not by searching for his playerID.

```
SELECT CONCAT (nameFirst, " ", nameLast) AS Name, yearID, ERA, (-2*SO +
3*BB + 3*HBP + 13*HR)/(IPOuts/3) + 3.139 AS FIP
FROM Pitching p
JOIN Master m
    ON p.playerId = m.playerID
WHERE nameFirst = 'Pedro' AND nameLast = 'Martinez'
ORDER BY (ERA - FIP) DESC
```

TECH 1 - VIDEO 5

```
SELECT b.playerID, SUM(b.AB) AS CAREER_AB, SUM(b.HR) AS CAREER_HR,
    SUM(p.IPOuts) AS CAREER_Ipouts, SUM(p.SO) AS CAREER_SO
FROM Batting b
JOIN Pitching p
    ON (b.playerId = p.playerID
    AND b.yearID = p.yearID
    AND b.stint = p.stint)
GROUP BY playerID DESC
```

TECH 1 - VIDEO 5

Write a query that joins the batting table (aliased as b) with the pitching table (aliased as p) on playerID, yearID, and stint. Then, group by yearID to get league numbers for players who pitched. Use aggregate functions to select b.yearID, total hits for pitchers as Total_H, total at bats for pitchers as Total_AB, and league batting average for pitchers as League_AVG. When calculating league batting average, be sure to divide *total* hits by *total* at bats. Order your results with the highest league batting average for pitchers first. Secondarily, order your results by yearID.

```
SELECT p.yearID, SUM(b.H) AS Total_H, SUM(b.AB) AS Total_AB,
    (SUM(b.H) / SUM(b.AB)) AS League_AVG
FROM Batting b
JOIN Pitching p
    ON (b.playerId = p.playerID
    AND b.yearID = p.yearID
    AND b.stint = p.stint)
GROUP BY yearID
ORDER BY League_AVG DESC, b.yearID
```

TECH 1 VIDEO 6

```
SELECT t.yearID, t.teamID, t.W, s.Payroll, s.Payroll/t.W AS DOLLARperW
FROM Teams t
JOIN (
    SELECT yearID, teamID, SUM(salary) AS Payroll
    FROM Salaries
    GROUP BY yearID, teamID
    ) s
ON t.yearID = s.yearID
    AND t.teamID = s.teamID
WHERE t.yearID >= 2000
ORDER BY DOLLARperW DESC
```

TECH 1 QUIZ 6

Now, use the query from the above problem to join the Teams table (aliased as t) to the table the is the result of the above query (aliased as s), on teamID and yearID. Select yearID, teamID, and attendance from Teams, and Payroll from the derived table (s). Also select attendance per payroll dollar, as FansPerDollar. Order your result so that the team with the fewest fans per dollar is first.

```
SELECT t.yearID, t.teamID, t.attendance, s.Payroll,
t.attendance/s.Payroll AS FansPerDollar
FROM Teams t
JOIN (
    SELECT yearID, teamID, SUM(salary) AS Payroll
    FROM Salaries
    GROUP BY yearID, teamID
    ) s
ON t.yearID = s.yearID
    AND t.teamID = s.teamID
ORDER BY FansPerDollar
```

TECH 1 VIDEO 7

```
SELECT CONCAT (nameFirst, " ", nameLast) AS PlayerName,
    G, AB, 2B, 3B, R, RBI, SB,
    H / AB AS BA,
    (H+BB+HBP)/(AB+BB+HBP+SF) AS OBP,
    ((H+2B+2*3B+3*HR)/AB) AS SLG
FROM Batting b
JOIN Master m
    ON b.playerId = m.playerID
WHERE birthcity = 'Cambridge'
AND AB > 50
ORDER BY SLG DESC
```

TECH 1 VIDEO 7

```
SELECT CONCAT (nameFirst, " ", nameLast) AS playerName,
    m.BirthCountry AS Country, p.yearID AS Year, (IPOuts/3) AS IP, p.ERA
AS ERA,
    ((3*BB + 3*HBP + 13*HR - 2*SO)/((IPOuts/3)) + 3.2) as FIP
FROM Pitching p
JOIN Master m
    ON p.playerId = m.playerID
WHERE m.BirthCountry = "Venezuela"
HAVING IP > 60
```

TECH 2 VIDEO 1

```
SELECT CONCAT (nameFirst, " ", nameLast) AS PlayerName,
    G, AB, 2B, 3B, R, RBI, SB,
    H / AB AS BA,
    (H+BB+HBP)/(AB+BB+HBP+SF) AS OBP,
    ((H+2B+2*3B+3*HR)/AB) AS SLG
FROM Batting b
JOIN Master m
    ON b.playerId = m.playerID
JOIN CollegePlaying sp
      ON s.playerId = sp.playerid
JOIN Schools s
      ON s.schoolID = sp.schoolID
WHERE schoolName = "Boston University"
AND AB > 50
ORDER BY SLG DESC
```

TECH 2 VIDEO 2

```
SELECT CONCAT (nameFirst, " ", nameLast) AS PlayerName,
b.G, b.AB, b.SB, b.W/b.AB AS BA,
(b.H+b.BB+b.HBP)/(b.AB+b.BB+b.HBP+b.SF) AS OBP,
(b.H+b.2B+2*b.3B+3*b.HR)/b.AB) AS SLG
f.POS, f.DP,
f.A + f.PO + f.E As TotalChances
(f.A + f.PO)/(f.A + f.PO + f.E) AS FPct
FROM Batting b
JOIN Master m
    ON b.playerId = m.playerID
JOIN Fielding f
      ON f.playerId = b.playerID
      AND f.yearID = b.yearID
WHERE b.yearID = 2013 AND b.teamID = 'DET'
ORDER BY TotalSchances DESC
```

TECH 2 QUIZ 1

```
SELECT CONCAT (nameFirst, " ", nameLast) AS playerName,  b.yearID AS
Year,
    (H/AB) AS BA,
    (H+BB+HBP)/(AB+BB+HBP+SF) AS OBP,
    (H+2B+2*3B+3*HR)/AB AS SLG
FROM Batting b
JOIN Master m
    ON b.playerId = m.playerID
JOIN AwardsPlayers a
      ON a.playerId = b.playerID
      AND a.yearID = b.yearID
WHERE a.awardID = 'Most Valuable Player' AND b.yearID > 1990
ORDER BY SLG DESC, b.playerID
```

TECH 2 VIDEOS LEFT JOIN / RIGHT JOIN

```
SELECT b.playerID, b.yearID, b.teamID, b.lgID, b.AB, b.H/b.AB AS BA,
p.IPouts, p.ERA
FROM Batting b
LEFT JOIN Pitching p
ON b.playerID = p.playerID
      AND b.yearID = p.yearID
      AND b.stint = p.stint
WHERE b.yearID = 2013 # AND b.AB > 0
1409 results
```

```
SELECT b.playerID, b.yearID, b.teamID, b.lgID, b.AB, b.H/b.AB AS BA,
p.IPouts, p.ERA
FROM Batting b
LEFT JOIN Pitching p
ON b.playerID = p.playerID
      AND b.yearID = p.yearID
      AND b.stint = p.stint
WHERE b.yearID = 2013 AND b.AB > 0
ORDER BY AB DESC
1008 results
```

```
SELECT p.playerID, p.yearID, p.teamID, b.lgID, b.AB, b.H/b.AB AS BA,
p.IPouts, p.ERA
FROM Batting b
RIGHT JOIN Pitching p
ON b.playerID = p.playerID
      AND b.yearID = p.yearID
      AND b.stint = p.stint
WHERE p.yearID = 2013
726 results
```

TECH 2 QUIZ 2

```
SELECT a.playerID, a.yearID, a.startingPos, SUM(b.H + b.2B + 2*b.3B +
3*b.HR) AS Post_TB
FROM BattingPost b
RIGHT JOIN AllstarFull a
    ON b.playerID = a.playerID
    AND b.yearID = a.yearID
WHERE a.yearID >= 1950
GROUP BY a.playerID, a.yearID
```

```
ORDER BY Post_TB DESC, a.playerID, a.yearID
```

FIELDING METRICS
TotalChances (TC) (PO + A + E)
FLD% = (PO + A) / (PO + A + E)
Range Factor (PO + A) / G
RF/9 = 9 * (PO/A) / Inn
FRAR(FIelding Runs Above Average

WEEK 4
LECTURE 2 Q2 - INCORRECT QUERY

```
SELECT yearID, playerID,
    (SUM(ER) * 9 / SUM(IPOuts/3)) AS ERA, 9 * (PO/A) AS RA9,
    (SUM(ER)/SUM(R)) AS ERPercentage
FROM Pitching
WHERE yearID >= 2000 AND yearID =< 2009 AND IPOuts > 3000
GROUP BY playerID
ORDER BY ERPercentage, playerID
```

```
SELECT playerID, SUM(ER*9)/SUM(IPOUTS/3) AS ERA, SUM(R*9)/SUM(IPOUTS/3)
AS RA9, SUM(ER/R) AS ERPercentage FROM Pitching WHERE yearID BETWEEN 2000
AND 2009
HAVING BY IPOuts > 3000
GROUP BY playerID ORDER BY ERPercentage, playerID
```

QUIZ 3 - INCORRECT 3/5

```
SELECT playerID, yearID,
    (50*GS + IPOuts + SO - (2*H) - (4*ER) - 2*(R-ER) - BB + 2*(IPOuts/3 -
GS*4))/GS AS GameScore
FROM Pitching
```

QUIZ 4

```
WHERE yearID >= 1980 AND GS > 20
ORDER BY GS DESC, playerID, yearID
```

PITCHING METRICS
WHIP = (W+H) / IP
K/9
BB/9
K/BB
K%
BB%
K-BB%
Pitching Runs = IP *   (LgERA/9) -   ER

```
      Pedro Martinez   IP 217 LgERA 4.83    42 =76.14
Clayton Kershaw = 216 * (3.87/9)     -    48 =

FIP = (13*HR) + (3*(BB+HBP))-(2*K) + C(3.1) / IP
--> ERA ~ 3.1 (C)
XFIP = ((13*(FB * (lgHR/FB))) + (3 * (BB+HBP)) - (2*K)) / IP + C
lgHR / FB ~ 10.8*
Line Drive Rate ~ 20%
FB (FlyBall) Rate ~ 36%
GB (GroundBall) ~ 44%
```

WEEK 5

```
SELECT b12.playerID, b12.teamID,
      (b12.AB + b12.BB + b12.HBP + b12.SF + b12.SH) AS PA12,
      (b12.H + b12.BB + b12.HBP) / (b12.AB + b12.BB + b12.HBP + b12.SF) +
(b12.H + b12.2B +
    2*b12.3B + 3*b12.HR) / b12.AB AS OPS12,
      (b13.AB + b13.BB + b13.HBP + b13.SF + b13.SH) AS PA13,
      (b13.H + b13.BB + b13.HBP) / (b13.AB + b13.BB + b13.HBP + b13.SF) +
(b13.H + b13.2B +
    2*b13.3B + 3*b13.HR) / b13.AB AS OPS13
FROM Batting b12, Batting b13
WHERE (b12.playerID = b13.playerID) AND (b12.yearID = 2012 AND b13.yearID
= 2013)
HAVING (PA12 > 100 AND PA13 > 100)
ORDER BY OPS12 DESC
```

```
SELECT p13.playerID, p13.teamID
, (p13.IPOuts/3) as IP13
, (p14.IPOuts/3) as IP14
, (p13.SO/p13.BFP) as KRate13
, (p14.SO/p14.BFP) as KRate14
, (p13.ERA) as ERA13
, (p14.ERA) as ERA14
, (p13.H - p13.HR)/(p13.BFP - p13.BB - p13.HBP - p13.HR - p13.SO -
p13.SH) as BABIP13
, (p14.H - p14.HR)/(p14.BFP - p14.BB - p14.HBP - p14.HR - p14.SO -
p14.SH) as BABIP14
FROM Pitching p13, Pitching p14
WHERE (p13.playerID = p14.playerID) AND (p13.yearID = 2013 AND p14.yearID
= 2014)
```

WEEK5 - LECTURE 2

Using the SQL Sandbox, let's explore the idea of using a multiplier on On-Base Percentage to get the best Run Extimator possible using OBP and SLG. This idea is mentioned in Moneyball, by Michael Lewis, and has sabermetric backing. Run the following code.

```
SELECT teamID,
yearID,
(H+BB+HBP)/(AB+BB+HBP+SF) as OBP,
(H+2B+2*3B+3*HR)/AB AS SLG,
(H+BB+HBP)/(AB+BB+HBP+SF) + (H+2B+2*3B+3*HR)/AB as OPS,
1.8*(H+BB+HBP)/(AB+BB+HBP+SF) + (H+2B+2*3B+3*HR)/AB as Adj_OPS,
(H+BB+HBP)/(AB+BB+HBP+SF) / ((H+BB+HBP)/(AB+BB+HBP+SF) +
(H+2B+2*3B+3*HR)/AB) as OBP_Ratio
FROM Teams
WHERE yearID >= 2000
HAVING OPS >= .800
ORDER BY OBP_Ratio DESC
LIMIT 30
HAVING (IP13 > 100 AND IP14 > 100)
ORDER BY KRate13 DESC
```

WEEK 5 - LECTURE 3 PYTHAGOREAN EXPECTATIONS

```
SELECT yearID, teamID,
        R, RA,
        R / (W+L) AS RperG,
        RA / (W+L) AS RAperG,
        W, L,
        W / (W+L) AS WPct,
        W/L AS WinRatio,
        R/RA AS RunRatio
FROM Teams
WHERE yearID >= 1901
ORDER BY WPct DESC
```

WEEK 6

MODULE 1

```
SELECT playerID, yearID, teamID
    , (AB + BB + HBP + SF + SH) as PA
    , (H + BB + HBP)/(AB + BB + HBP + SF) as OBP
    , (H + 2B + 2*3B + 3*HR)/AB as SLG
    , (H + BB + HBP)/(AB + BB + HBP + SF) + (H + 2B + 2*3B + 3*HR)/AB as
OPS
FROM Batting
WHERE yearID = 2013
HAVING PA > 299
ORDER BY OPS DESC
```

MODULE 3 - QUIZ2

```
SELECT (SUM(R) / (SUM(IPOuts) / 27)) AS League_RA9
FROM Pitching
WHERE yearid = 2003

SELECT playerID, R/(IPOuts/27) AS RA9,
    ((SUM(R) / (SUM(IPOuts) / 27)) - (R/(IPOuts/27))*(IPOuts/27)/10.2 +
IPOuts/270) AS WAR
FROM Pitching
WHERE yearID = 2003
ORDER BY WAR DESC, playerID
```

FINAL ASSIGNMENT

Now, we will modify our query to get statistics for the AL and NL separately, by year. Use a group by clause to ensure that each row represents one year of one league (note that lgID is the identifier for

AL or NL). Select yearID, lgID, and SO_Pct and BB_Pct, but make sure that these represent the statistics for the entire league. Include results for all years since and including 1995, with no AB minimum. Order your results by yearID, and secondarily by lgID

```
SELECT yearID, lgID, SUM(SO)/SUM(AB + SF + SH + BB + HBP) AS SO_Pct,
SUM(BB)/SUM(AB + SF + SH + BB + HBP) AS BB_Pct
          FROM Batting
          WHERE yearID >= 1995
          GROUP BY lgID, yearID
          ORDER BY yearID, lgID
```

Q3 - INCORRECT

```
SELECT playerID,
    ((.7*(BB+HBP) + .9*(H-2B-3B-HR) + 1.25*2B + 1.6*3B + 2*HR)/(AB + BB +
HBP + SF + SH)) AS wOBA,
    (((AB + BB + HBP + SF + SH)*((.7*(BB+HBP) + .9*(H-2B-3B-HR) + 1.25*2B
+ 1.6*3B + 2*HR) /((AB + BB + HBP + SF + SH))/(1.2) AS wRAA
FROM Batting
WHERE yearID = 2000
ORDER BY wRAA DESC, playerID
```

Q4 - CORRECT

```
SELECT CONCAT (nameFirst, " ", nameLast) AS Name,  pit.yearID, pit.ERA,
(IPOuts/3) AS IP
FROM Pitching pit
JOIN Master m
    ON pit.playerId = m.playerID
WHERE m.birthState = 'MA' AND (IPOuts/3) > 100 AND ERA < 3 AND yearID >=
1960
ORDER BY ERA, m.playerID, yearID
```