SQL Project – Major League Baseball (MLB)

#1. In each decade, how many schools were there that produced players?

SQL Code

```
SELECT FLOOR(YearID/10)*10 AS decade,
                count(DISTINCT schoolID) AS schools_count
FROM schools
GROUP BY decade
ORDER BY decade ASC;
```

decade	schools_count
1860	2
1870	14
1880	34
1890	89
1900	148
1910	178
1920	196
1930	162
1940	142
1950	176
1960	301
1970	427
1980	473
1990	494
2000	372
2010	57

#2. What are the names of the top 5 schools that produced the most players?

SQL Code

```
SELECT sd.name_full AS school_name,
        COUNT(DISTINCT s.playerID) AS players_num
FROM schools s LEFT JOIN school_details sd
ON s.schoolID = sd.schoolID
GROUP BY s.schoolID
ORDER BY players_num DESC
LIMIT 5;
```

school_name	players_num
University of Texas at Austin	107
University of Southern California	105
Arizona State University	101
Stanford University	86
University of Michigan	76

#3. For each decade, what were the names of the top 3 schools that produced the most players?

```
WITH ps AS (SELECT FLOOR(s.YearID/10)*10 AS decade,
        sd.name_full AS school_name, COUNT(DISTINCT playerID) AS player_count
FROM schools s LEFT JOIN school_details sd
ON s.schoolID = sd.schoolID
GROUP BY decade, school_name
ORDER BY decade, player_count DESC),

top AS (SELECT decade, school_name, player_count,
        ROW_NUMBER() OVER (partition by decade ORDER BY player_count DESC) AS top
FROM ps)

SELECT decade, school_name, player_count
FROM top
WHERE top BETWEEN 1 AND 3
ORDER BY decade DESC, player_count DESC;
```

decade	school_name	player_count
2010	University of Florida	5
2010	University of Texas at Austin	4
2010	Georgia Institute of Technology	3
2000	Arizona State University	23
2000	California State University Long Beach	23
2000	Stanford University	22
1990	Stanford University	25
1990	University of Southern California	23
1990	Louisiana State University	22
1980	University of Arizona	24
1980	Arizona State University	23
1980	University of California, Los Angeles	22
1970	Arizona State University	32
1970	University of Southern California	24
1970	University of Texas at Austin	20
1960	Arizona State University	18
1960	University of Southern California	17
1960	University of Michigan	14
1950	University of Southern California	12
1950	Michigan State University	9
1950	University of Texas at Austin	7
1940	University of Southern California	9
1940	University of Illinois at Urbana-Cham...	8
1940	University of Texas at Austin	7
1930	Duke University	14
1930	University of Texas at Austin	11
1930	College of the Holy Cross	11
1920	University of Alabama	19
1920	College of the Holy Cross	15
1920	University of Texas at Austin	12
1910	St. Mary's College of California	11
1910	College of the Holy Cross	11
1910	Brown University	9
1900	University of Notre Dame	16
1900	Manhattan College	14
1900	College of the Holy Cross	14
1890	College of the Holy Cross	13
1890	Brown University	13
1890	University of Pennsylvania	9
1880	Yale University	6
1880	Brown University	5
1880	College of the Holy Cross	3
1870	Yale University	3
1870	Brown University	3
1870	Dartmouth College	1
1860	Villanova University	1
1860	Fordham University	1

#4. Return the top 20% of teams in terms of average annual spending

```
WITH ts AS (SELECT yearID, teamID,
                SUM(salary /1000000) AS total_spend_mil
FROM salaries
GROUP BY yearID, teamID
ORDER BY total_spend_mil DESC),

tas AS (SELECT teamID, ROUND(avg(total_spend_mil),1) AS avg_spend
FROM ts
GROUP BY teamID
ORDER BY avg_spend DESC),

perc AS (SELECT teamID, avg_spend,
                NTILE(5) OVER (ORDER BY avg_spend DESC) AS per
FROM tas)

SELECT teamID, avg_spend
FROM perc
WHERE per = 1
ORDER BY avg_spend DESC;
```

teamID	avg_spend
SFG	143.5
LAA	118.5
NYA	109.4
BOS	81.1
LAN	74.6
WAS	71.5
ARI	71.2
PHI	66.1

#5. For each team, show the cumulative sum of spending over the years

```
WITH ts AS (SELECT yearID, teamID,
            SUM(salary /1000000) AS total_spend_mil
FROM salaries
GROUP BY yearID, teamID)

SELECT teamID, yearID,
            ROUND(sum(total_spend_mil)
          OVER (PARTITION BY teamID ORDER BY yearID),1) AS cum_spend
FROM ts;
```

teamID	yearID	cum_spend
ANA	1997	31.1
ANA	1998	72.4
ANA	1999	127.8
ANA	2000	179.3
ANA	2001	226.8
ANA	2002	288.5
ANA	2003	367.6
ANA	2004	468.1
ARI	1998	32.3
ARI	1999	101.1
ARI	2000	182.1
ARI	2001	267.2
ARI	2002	370.0
ARI	2003	450.6
ARI	2004	520.4
ARI	2005	582.7
ARI	2006	642.4
ARI	2007	694.5
ARI	2008	760.7
ARI	2009	833.8
ARI	2010	894.5
ARI	2011	948.2
ARI	2012	1022.0
ARI	2013	1112.1
ARI	2014	1210.0
ATL	1985	14.8
ATL	1986	31.9
ATL	1987	48.5
ATL	1988	61.2
ATL	1989	72.3
ATL	1990	86.9
ATL	1991	105.3
ATL	1992	139.9
ATL	1993	181.5
ATL	1994	230.9
ATL	1995	278.1
ATL	1996	327.8
ATL	1997	380.1
ATL	1998	441.3
ATL	1999	514.4
ATL	2000	599.0
ATL	2001	690.9
ATL	2002	783.8
ATL	2003	890.0
ATL	2004	980.2
ATL	2005	1066.7
ATL	2006	1156.8
ATL	2007	1244.1
ATL	2008	1346.5
ATL	2009	1443.2
ATL	2010	1527.6
ATL	2011	1614.6
ATL	2012	1697.5
ATL	2013	1785.3
ATL	2014	1882.9
BAL	1985	11.6
BAL	1986	24.6

#6. Return the first year that each team's cumulative spending surpassed 1 billion

```
WITH ts AS (SELECT yearID, teamID, sum(salary /1000000000) AS total_spend_in_billions
FROM salaries
GROUP BY yearID, teamID),

tcs AS (SELECT teamID, yearID,
        ROUND(sum(total_spend_in_billions) OVER (PARTITION BY teamID ORDER BY
        yearID),2) AS cum_spend_billion
FROM ts),

tb AS (SELECT teamID, yearID, cum_spend_billion
FROM tcs
WHERE  cum_spend_billion >= 1),

tr AS (SELECT teamID, yearID,cum_spend_billion,
            ROW_NUMBER() OVER (PARTITION BY teamID ORDER BY yearID) As row_num
FROM tb)

SELECT teamID, yearID, cum_spend_billion
FROM tr
WHERE row_num =1;
```

teamID	yearID	cum_spend_billion
ARI	2012	1.02
ATL	2005	1.07
BAL	2007	1.06
BOS	2004	1.00
CHA	2008	1.07
CHN	2007	1.08
CIN	2010	1.06
CLE	2009	1.06
COL	2011	1.05
DET	2008	1.00
HOU	2008	1.03
KCA	2012	1.02
LAA	2013	1.06
LAN	2005	1.08
MIL	2014	1.05
MIN	2011	1.02
NYA	2003	1.06
NYN	2005	1.04
OAK	2011	1.00
PHI	2008	1.03
SDN	2012	1.04
SEA	2007	1.04
SFN	2007	1.04
SLN	2007	1.07
TEX	2007	1.04
TOR	2008	1.05

#7. View the players table and find the number of players in the table

```
SELECT COUNT(PlayerID) AS num_of_players
FROM players;
```

num_of_players
18589

#8. For each player, calculate their age at their first game, their last game, and their career length (all in years). Sort from longest career to shortest career.

```
SELECT    playerID, nameGiven AS player_name,
          Year(debut) - birthyear AS age_first_game, year(finalGame) - birthyear AS age_last_game,
          year(finalgame) - year(debut) AS careerlength
FROM      players
ORDER BY  careerlength DESC;
```

Please note: The output screenshot is only for top rows. It doesn't show all the rows.

playerID	player_name	age_first_game	age_last_game	careerlength
altroni01	Nicholas	22	57	35
orourji01	James Henry	22	54	32
minosmi01	Saturnino Orestes Armas	24	55	31
olearch01	Charles Timothy	29	59	30
lathaar01	Walter Arlington	20	49	29
mcguide01	James Thomas	21	49	28
eversjo01	John Joseph	21	48	27
jennihu01	Hugh Ambrose	22	49	27
ryanno01	Lynn Nolan	19	46	27
streega01	Charles Evard	22	49	27
johnto01	Thomas Edward	20	46	26
moyerja01	Jamie	24	50	26
ansonca01	Adrian Constantine	19	45	26
broutda01	Dennis Joseph	21	46	25
francju01	Julio Cesar	24	49	25
gleaski01	William J.	22	46	24
henderi01	Rickey Nelson Henley	21	45	24
fiskca01	Carlton Ernest	22	46	24
kaatji01	James Lee	21	45	24
houghch01	Charles Oliver	22	46	24
collied01	Edward Trowbridge	19	43	24
wallabo01	Roderick John	21	45	24
ryanja01	John Bernard	21	45	24
wynnea01	Early	19	43	24
newsobo01	Louis Norman	22	46	24
morgami01	Michael Thomas	19	43	24
quinnja01	John Picus	26	50	24
oroscje01	Jesse Russell	22	46	24
lyonste01	Theodore Amar	23	46	23
raineti01	Timothy	20	43	23
maranra01	Walter James Vincent	21	44	23
oconnja01	John Joseph	21	44	23
niekrph01	Philip Henry	25	48	23
coonejo01	John Walter	20	43	23
cobbty01	Tyrus Raymond	19	42	23
clemero02	William Roger	22	45	23
carltst01	Steven Norman	21	44	23
eckerde01	Dennis Lee	21	44	23
dempsri01	John Rikard	20	43	23
griffcl01	Clark Calvin	22	45	23
hartlgr01	Grover Allen	23	46	23
rosepe01	Peter Edward	22	45	23
ruffire01	Charles Herbert	19	42	23
vizquom01	Omar Enrique	22	45	23

#9. How many players started and ended on the same team and also played for over a decade?

```
WITH pi AS (SELECT  playerID, nameGiven AS player_name, year(debut) AS debut_year,
                    year(finalGame) AS end_year,
                    year(finalgame) - year(debut) AS careerlength
FROM players
ORDER BY careerlength DESC),

psy AS (SELECT pi.playerID, pi.player_name, pi.debut_year, s.teamID AS starting_team,
                    pi.end_year, pi.careerlength
FROM pi LEFT JOIN salaries s
ON pi.playerID = s.playerID
WHERE s.yearID = pi.debut_year),

pet AS (SELECT psy.playerID, psy.player_name, psy.debut_year, psy.starting_team,
                    psy.end_year, s.teamID as ending_team, psy.careerlength
FROM psy LEFT JOIN salaries s
ON psy.playerID = s.playerID
WHERE s.yearID = psy.end_year)

SELECT player_name, debut_year, starting_team, end_year, ending_team
FROM pet
WHERE starting_team = ending_team AND careerlength > 10
ORDER BY starting_team;
```

player_name	debut_year	starting_team	end_year	ending_team
Thomas Michael	1987	ATL	2008	ATL
Larry Wayne	1993	ATL	2012	ATL
Ellis Rena	1987	BOS	2004	BOS
Ronald Joseph	1986	CHA	1997	CHA
Kerry Lee	1998	CHN	2012	CHN
Barry Louis	1986	CIN	2004	CIN
Todd Lynn	1997	COL	2013	COL
Brad William	1995	MIN	2006	MIN
Bernabe	1991	NYA	2006	NYA
Andrew Eugene	1995	NYA	2013	NYA
Mariano	1995	NYA	2013	NYA
Chase Cameron	2003	PHI	2014	PHI
David Michael	1990	PHI	2002	PHI
George Kenneth	1989	SEA	2010	SEA
Richard Santo	1995	SFN	2009	SFN
Raymond Lewis	1990	SLN	2004	SLN
Thomas Alan	1987	SLN	1998	SLN
Samuel Peralta	1989	TEX	2007	TEX
Patrick George	1991	TOR	2004	TOR

#10. Which players have the same birthday?

```
WITH p AS (SELECT nameGiven AS player_name, CAST(CONCAT(birthYear, '-', birthmonth, '-', birthDay) AS DATE) AS DOB
FROM players)

SELECT DOB, GROUP_CONCAT(player_name SEPARATOR ',') AS players
FROM p
WHERE YEAR(DOB) IS NOT NULL
GROUP BY DOB
HAVING COUNT(player_name) > 1
ORDER BY DOB;
```

Please note: The output screenshot is only for top rows. It doesn't show all the rows.

DOB	players
1845-01-31	Freeman,Robert Vavasour
1854-05-04	Frank Bernard,James Henry
1854-10-06	Francis,Charles N.
1855-01-01	Thomas Edward,William Henry,William A.
1855-02-14	Louis J.,John Joseph
1855-08-20	George Cresse,David P.
1855-10-02	Cyrus Alban,John Robert
1856-09-05	James,John Parkinson
1857-03-09	George R.,Samuel R.
1857-10-24	Edmund Dana,Edward Nagle
1858-03-03	John P.,Harry Eugene
1858-04-01	John,Fred J.
1858-06-26	Dennis J.,Lorenzo Burroughs
1858-07-15	William J.,John Nelson
1858-07-18	George William,Edward T.
1858-10-24	Tobias Charles,William J.
1858-11-11	Charles Anthony,Robert H.
1858-11-16	Benjamin Franklin,Joseph
1858-11-20	Joseph John,Laurence P.
1859-08-10	Lawrence J.,Sidney Douglas
1859-10-29	John,Charles Hercules
1860-01-12	Henry E.,John William
1861-02-17	Joseph A.,George Edward
1861-06-28	Mortimer Martin,William Thomas
1861-07-01	John Gibson,Charles L.,Charles Franklin
1861-07-21	Percival Wheritt,John
1861-11-12	Patrick E.,John Henry
1861-12-21	Conrad,Harry H.
1861-12-31	Walton Hugh,James J.
1862-06-18	Charles William,Howard Carleton
1862-09-07	Michael Joseph,Edward M.
1862-11-13	Peter James,John Garibaldi
1863-02-26	Simeon Henry Jean,Edward
1863-03-04	Allen A.,John George
1863-05-10	James B.,John F.,John Leckie
1863-08-10	William Crawford,George Washington
1863-09-01	George A.,William Darby
1863-10-04	James Nathaniel,William James

#11. Create a summary table that shows for each team, what percent of players bat right, left and both

```
WITH bat AS (SELECT s.teamID, sum(IF(p.bats = 'R', 1,0)) as right_b,
                    sum(IF(p.bats = 'L',1,0)) AS left_b,
                    sum(IF(p.bats = 'B',1,0)) AS both_b, count(s.playerID) AS total
FROM salaries s LEFT JOIN players p
ON s.playerID = p.playerID
GROUP BY s.teamID)

SELECT              teamID, right_b/total*100 AS right_b_perc,
                            left_b/total*100 AS left_b_perc,
                            both_b/total*100 AS both_b_perc
FROM bat
ORDER BY teamID;
```

Please note: The output screenshot is only for top rows. It doesn't show all the rows.

teamID	right_b_perc	left_b_perc	both_b_perc
ANA	61.1336	31.5789	7.2874
ARI	61.5702	30.3719	7.8512
ATL	61.8329	29.2343	8.9327
BAL	61.8347	29.5583	8.6070
BOS	61.9479	29.3318	8.6070
CAL	60.5978	29.3478	10.0543
CHA	59.6890	33.4928	6.8182
CHN	63.7972	28.5377	7.6651
CIN	62.5858	29.4050	8.0092
CLE	59.5745	29.6753	10.7503
COL	63.6223	27.7090	8.5139
DET	60.8333	28.5714	10.5952
FLO	66.3265	24.3197	9.3537
HOU	62.3030	23.8788	13.8182
KCA	64.3021	27.2311	8.4668
LAA	68.1979	16.6078	15.1943
LAN	62.9339	27.7716	9.1825
MIA	64.1026	29.4872	5.1282
MIL	66.3883	29.4363	4.1754
MIN	60.8696	26.7081	12.4224
ML4	59.5801	29.3963	11.0236
MON	63.7782	24.0901	12.1317
NYA	58.8168	30.7167	10.4664
NYM	66.6667	29.1667	4.1667
NYN	56.0859	30.1909	13.7232
OAK	62.6561	27.4688	9.8751
PHI	58.4546	31.3550	10.1904
PIT	64.3914	27.4175	8.1911
SDN	61.4583	28.9352	9.6065
SEA	61.6845	28.9442	9.3713
SFG	55.5556	25.9259	18.5185
SFN	61.1449	27.5274	11.3276
SLN	61.8510	26.5237	11.6253

#12. How have average height and weight at debut game changed over the years, and what's the decade-over-decade difference?

```
WITH pd AS (SELECT  FLOOR(year(debut)/10)*10 AS yr, avg(weight) AS avg_weight,
                    avg(height) AS avg_height
FROM players
GROUP BY yr)

SELECT yr, avg_weight,
                    avg_weight - lag(avg_weight) OVER (ORDER BY yr) AS dec_weight_diff,
                    avg_height,
                    avg_height - lag(avg_height) OVER (ORDER BY yr) AS dec_height_diff
FROM pd
WHERE yr IS NOT NULL;
```

yr	avg_weight	dec_weight_diff	avg_height	dec_height_diff
1870	163.1394	NULL	68.8415	NULL
1880	169.0087	5.8693	69.5838	0.7423
1890	170.3323	1.3236	69.9861	0.4023
1900	174.0783	3.7460	70.5297	0.5436
1910	171.8658	-2.2125	70.7816	0.2519
1920	173.0967	1.2309	70.9092	0.1276
1930	178.8141	5.7174	71.6435	0.7343
1940	182.3502	3.5361	72.0514	0.4079
1950	184.4131	2.0629	72.4654	0.4140
1960	185.8705	1.4574	72.8793	0.4139
1970	186.0540	0.1835	73.0714	0.1921
1980	187.7023	1.6483	73.3436	0.2722
1990	193.8888	6.1865	73.4896	0.1460
2000	205.8854	11.9966	73.6789	0.1893
2010	207.3201	1.4347	73.6043	-0.0746