

Chapter 5 SQL Questions

- Using the view provided in the assignment page, write a query that uses the RANK function to rank the careerBA column where the careerBA < 0.40. Your results must show the playerid, Full Name, CareerBA and the rank for the players. This query returns 17,658 rows

playerid	Full Name	careerBA	BA_rank
forstte01	Terry Jay (Terry) Forster	0.3974	1
vorhecy01	Henry Bert (Cy) Vorhees	0.3913	2
oterore01	Regino Jose (Reggie) Otero	0.3913	2
underfr01	Frederick Theodore (Fred) Underwood	0.3889	4

- Write the same query as #2 but eliminate any gaps in the ranking. This query returns 17,658 rows

playerid	Full Name	careerBA	BA_rank
forstte01	Terry Jay (Terry) Forster	0.3974	1
vorhecy01	Henry Bert (Cy) Vorhees	0.3913	2
oterore01	Regino Jose (Reggie) Otero	0.3913	2
underfr01	Frederick Theodore (Fred) Underwood	0.3889	3

- Write the same query as #1, but find the ranking within the last year played by the player starting with the most current year and working backwards. Also eliminate any player where the career batting average is = 0. This query returns 14,818 rows

playerid	Full Name	lastplayed	careerBA	BA_rank
ceasedy01	Dylan Edward (Dylan) Cease	2021	0.3750	1
castiiv01	Ivan Enrique (Ivan) Castillo	2021	0.3333	2
camarda01	Daniel Ricardo (Dan) Camarena	2021	0.3333	2

- Write the same query as #3, but show the ranking by quartile (use the NTILE(4) parameter). This query returns 14,818 rows.

playerid	Full Name	lastplayed	careerBA	Ntile
ceasedy01	Dylan Edward (Dylan) Cease	2021	0.3750	1
castiiv01	Ivan Enrique (Ivan) Castillo	2021	0.3333	1
camarda01	Daniel Ricardo (Dan) Camarena	2021	0.3333	1
abbotco01	Cory James (Cory) Abbott	2021	0.3333	1

To help check your answer, Ntile 2 starts at row 281 with

playerid	Full Name	lastplayed	careerBA	Ntile
moncayo01	Yoan Manuel (Yoan) Moncada	2021	0.2610	2

- Using the Salaries table, write a query that compares the averages salary by team and year with the windowed average of the 3 prior years and the 1 year after the current year. This query returns 1,068 rows. **Note:** You will need to use multiple subqueries to get the answer.

teamid	yearid	Avg_Salary	Windowed_Salary
ANA	1997	\$1,004,370.06	\$1,109,258.56
ANA	1998	\$1,214,147.06	\$1,201,073.76
ANA	1999	\$1,384,704.15	\$1,329,673.38
ANA	2000	\$1,715,472.23	\$1,380,639.81

6. For each team and every player that played for that team, write a query that returns the Teamid, playerid, the Team's Average Salary, the Players average salary and the rank of the team against all other teams and the rank of the player against all other players. Limit your data to the teams that are ranked in the top 10. This query should return 2,927 rows.

teamid	playerid	Team Avg	Player Sal	Tsal_Rank	PSal_Rank
BR3	scherma01	\$7,298,831.40	\$34,603,480.00	1	3
BR3	priceda01	\$7,298,831.40	\$32,000,000.00	1	6
BR3	bauertr01	\$7,298,831.40	\$31,333,333.00	1	7

7. and player Write a query that shows that teamid, playerid, Player Full Name, total hits, total at bats, total batting average (calculated by using $\text{sum}(H)*1.0/\text{sum}(AB)$ as the formula) and show the players rank within the team and the rank within all players. Only include players that have a minimum of 150 career hits. This query returns 2,445 rows

teamid	playerid	Full Name	Total Hits	Totals At Bats	Batting Avg	Team Batting Rank	All Batting Rank
ANA	quinlro01	Robb Michael (Robb) Quinlan	55	160	0.3438	1	89
ANA	guerrvl01	Vladimir (Vladimir) Guerrero	206	612	0.3366	2	147
ANA	anderga01	Garret Joseph (Garret) Anderson	1468	4903	0.2994	3	1199
ANA	figgich01	Desmond DeChone (Chone) Figgins	242	817	0.2962	4	1422
ANA	fullmbr01	Bradley Ryan (Brad) Fullmer	187	635	0.2945	5	1544

8. When you were completing the View Assignment, you were not able to include the last college a player attended because you were not familiar with the RANK function. Using the RANK function, write a query that shows the playerid, last year attending college, the schoolid and the rank used to decide on the last school. as well as the last school the player attended. I used multiple subqueries to get my answer. Your query should return 6,577 rows.

playerid	yearid	schoolid	srank
aardsda01	2003	rice	3
abadan01	1993	gamiddl	2
abbeybe01	1892	vermont	4
abbotje01	1994	kentucky	3

9. You've decided that due to the number of queries that use the Salaries table, you need to create a primary key consisting of Playerid, Teamid, Yearid and LGID. When try to create the primary key, you will be told that there are duplicate records. You will receive the following error

The CREATE UNIQUE INDEX statement terminated because a duplicate key was found for the object name 'dbo.Salaries' and the index name 'PK__Salaries__89095551BDD28FCE'. The duplicate key value is (NL, ARI , 2019, GreinZa01).

Note that SQL Server stops when a single error is found when creating indexes. Although a key is provided, that is not the only row that needs the duplicate removed. There are 1,791 rows as identified by the following query

```
select playerid, yearid, lgid, teamid, count(*) as dupcount
from #salaries
group by playerid, yearid, lgid, teamid
having count(*) >1
```

yearid	lgid	teamid	playerid	dupcount
2017	NL	WAS	treinbl.01	2
2017	NL	WAS	turneja.01	2
2017	NL	WAS	turnetr.01	2

Write a query that performs the following steps:

- a. Using a **SELECT * INTO** query, create a temporary copy of the Salaries table named #Salaries
- b. Using the row_number function method described in the hint below, write the query that will remove the duplicate records in the #salaries table so that you can create the primary key
- c. Create the appropriate primary key for the #Salaries table

Hint: Use the Row_Number in a WITH statement and then delete from the WITH statement where row number > 1. The query should delete 3,617

10. Using a recursive CTE, write a query that will generate the days of the week using the DATENAME(DW, N) function where N indicates the day of the week. Your query must recurse and use N+1 to get the next day of the week. The output should be:

Results		Messages	
	weekday		
1	Monday		
2	Tuesday		
3	Wednesday		
4	Thursday		
5	Friday		
6	Saturday		
7	Sunday		