BASEBALL STATS

Final Project Report

Subject: - ISM6218 Advanced Database Management

Submitted by:

G. SIVA KRISHNA REDDY	**U55391402**
ALEXANDRA NIEVES	**U32118435**
B. PAVAN GOPI	**U99855563**
SK. MOHAMMED ZUBAIR	**U43640420**
V. SRI VAISHNAV	**U29421544**

Major in

BUSINESS ANALYTICS AND INFORMATION SYSTEMS

Under the guidance of

DR. DON BERNDT

MUMA COLLEGE OF BUSINESS

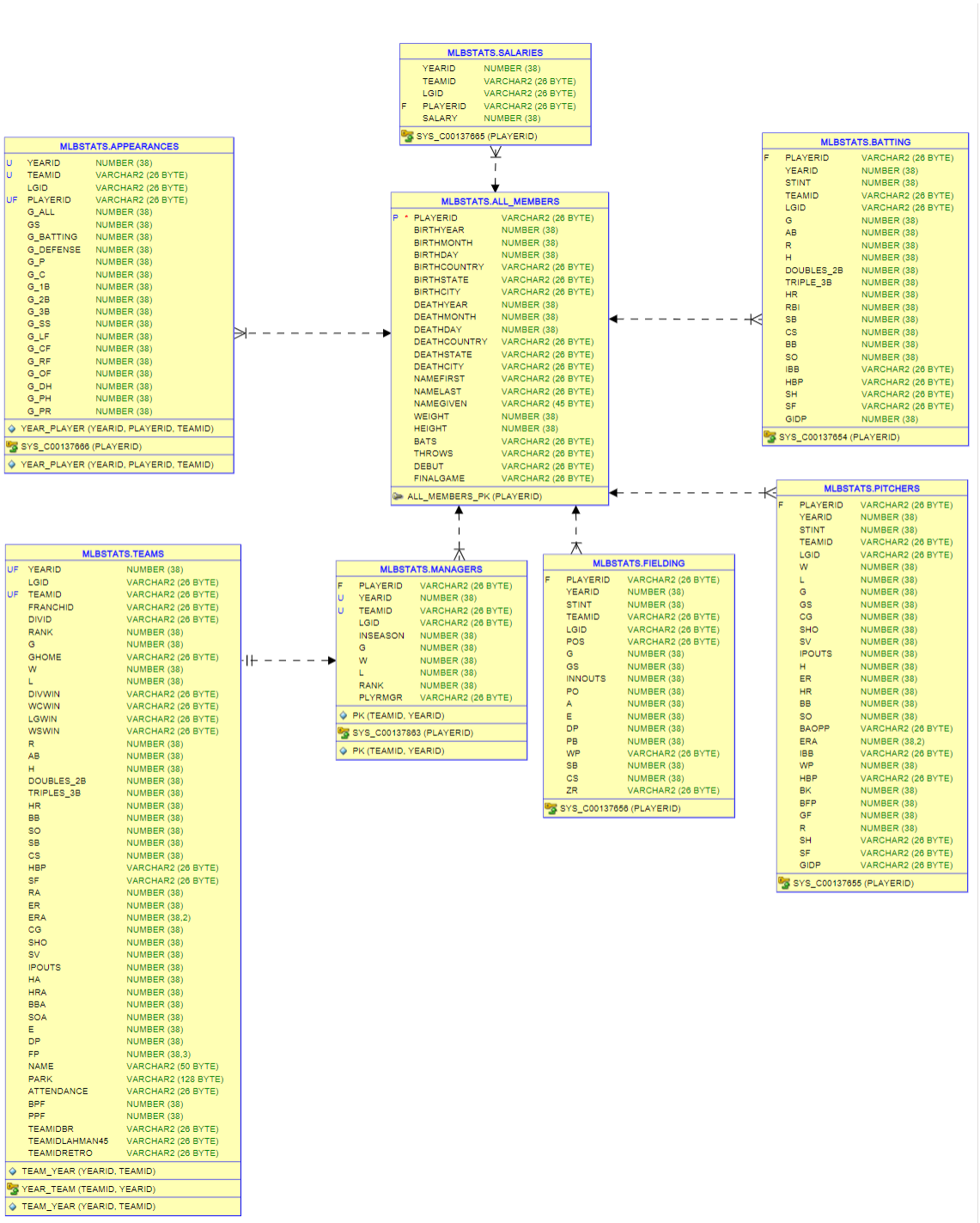
UNIVERSITY OF SOUTH FLORIDA

Table of Contents

1. INTRODUCTION

The real time baseball stats till 2014. Each stats needs to store different data types which may be used by various components. A careful design of database schema is critical to view the stats of each individual player stats. All the data here is Data cubes.

2. DATABASE DESIGN OVERVIEW



3. TABLES

3.1.ALL_MEMBERS

playerID	A unique code asssigned to each player. The playerID links the data in this file with records in the other files.
birthYear	Year player was born
birthMonth	Month player was born
birthDay	Day player was born
birthCountry	Country where player was born
birthState	State where player was born
birthCity	City where player was born
deathYear	Year player died
deathMonth	Month player died
deathDay	Day player died
deathCountry	Country where player died
deathState	State where player died
deathCity	City where player died
nameFirst	Player's first name
nameLast	Player's last name
nameGiven	Player's given name (typically first and middle)
weight	Player's weight in pounds
height	Player's height in inches
bats	Player's batting hand (left, right, or both)
throws	Player's throwing hand (left or right)
debut	Date that player made first major league appearance
finalGame	Date that player made first major league appearance (blank if still active)

3.2.BATTING TABLE

playerID	Player ID code
yearID	Year
stint	player's stint (order of appearances within a season)
teamID	Team
lgID	League
G	Games
AB	At Bats
R	Runs
H	Hits
2B	Doubles
3B	Triples
HR	Homeruns

RBI	Runs Batted In
SB	Stolen Bases
CS	Caught Stealing
BB	Base on Balls
SO	Strikeouts
IBB	Intentional walks
HBP	Hit by pitch
SH	Sacrifice hits
SF	Sacrifice flies
GIDP	Grounded into double plays

3.3.PITCHING TABLE

playerID	Player ID code
yearID	Year
stint	player's stint (order of appearances within a season)
teamID	Team
lgID	League
W	Wins
L	Losses
G	Games
GS	Games Started
CG	Complete Games
SHO	Shutouts
SV	Saves
IPOuts	Outs Pitched (innings pitched x 3)
H	Hits
ER	Earned Runs
HR	Homeruns
BB	Walks
SO	Strikeouts
BAOpp	Opponent's Batting Average
ERA	Earned Run Average
IBB	Intentional Walks
WP	Wild Pitches
HBP	Batters Hit By Pitch
BK	Balks
BFP	Batters faced by Pitcher
GF	Games Finished
R	Runs Allowed

SH	Sacrifices by opposing batters
SF	Sacrifice flies by opposing batters
GIDP	Grounded into double plays by opposing batter

3.4. FIELDING TABLE

playerID	Player ID code
yearID	Year
stint	player's stint (order of appearances within a season)
teamID	Team
lgID	League
Pos	Position
G	Games
GS	Games Started
InnOuts	Time played in the field expressed as outs
PO	Putouts
A	Assists
E	Errors
DP	Double Plays
PB	Passed Balls (by catchers)
WP	Wild Pitches (by catchers)
SB	Opponent Stolen Bases (by catchers)
CS	Opponents Caught Stealing (by catchers)
ZR	Zone Rating

3.5. TEAMS TABLE

yearID	Year
lgID	League
teamID	Team
franchID	Franchise (links to TeamsFranchise table)
divID	Team's division
Rank	Position in final standings
G	Games played
GHome	Games played at home
W	Wins
L	Losses
DivWin	Division Winner (Y or N)
WCWin	Wild Card Winner (Y or N)

LgWin	League Champion(Y or N)
WSWin	World Series Winner (Y or N)
R	Runs scored
AB	At bats
H	Hits by batters
2B	Doubles
3B	Triples
HR	Homeruns by batters
BB	Walks by batters
SO	Strikeouts by batters
SB	Stolen bases
CS	Caught stealing
HBP	Batters hit by pitch
SF	Sacrifice flies
RA	Opponents runs scored
ER	Earned runs allowed
ERA	Earned run average
CG	Complete games
SHO	Shutouts
SV	Saves
IPOuts	Outs Pitched (innings pitched x 3)
HA	Hits allowed
HRA	Homeruns allowed
BBA	Walks allowed
SOA	Strikeouts by pitchers
E	Errors
DP	Double Plays
FP	Fielding percentage
name	Team's full name
park	Name of team's home ballpark
attendance	Home attendance total
BPF	Three-year park factor for batters
PPF	Three-year park factor for pitchers
teamIDBR	Team ID used by Baseball Reference website
teamIDlahman	45 Team ID used in Lahman database version 4.5
teamIDretro	Team ID used by Retrosheet

3.6. SALARIES TABLE

yearID	Year
teamID	Team
lgID	League
playerID	Player ID code
salary	Salary

3.7. APPEARANCES TABLE

yearID	Year
teamID	Team
lgID	League
playerID	Player ID code
G_all	Total games played
GS	Games started
G_batting	Games in which player batted
G_defense	Games in which player appeared on defense
G_p	Games as pitcher
G_c	Games as catcher
G_1b	Games as firstbaseman
G_2b	Games as secondbaseman
G_3b	Games as thirdbaseman
G_ss	Games as shortstop
G_lf	Games as leftfielder
G_cf	Games as centerfielder
G_rf	Games as right fielder
G_of	Games as outfielder
G_dh	Games as designated hitter
G_ph	Games as pinch hitter
G_pr	Games as pinch runner

3.8.MANAGERS TABLE

playerID	Player ID Number
yearID	Year
teamID	Team
lgID	League
inseason	Managerial order. Zero if the individual managed the team the entire year. Otherwise denotes where the manager appeared in the managerial order (1 for first manager, 2 for second, etc.)
G	Games managed
W	Wins
L	Losses
rank	Team's final position in standings that year
plyrMgr	Player Manager (denoted by 'Y')

4. INDEXING

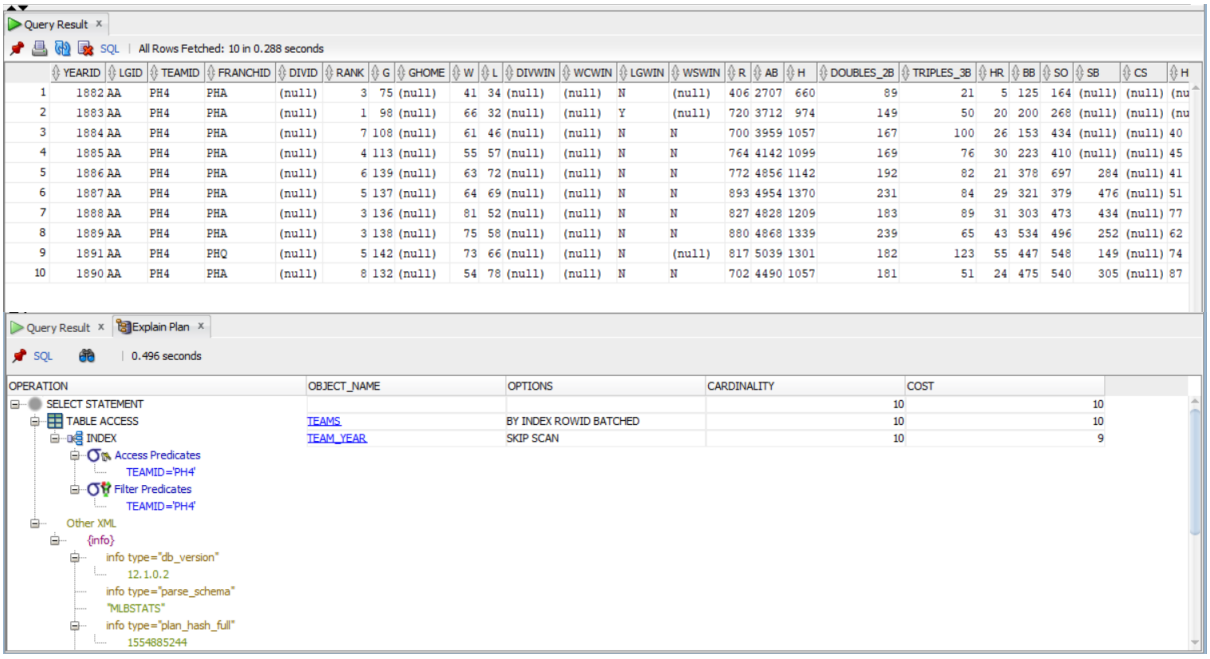
Indexes are used to quickly locate data without having to search every row in a database table every time a database table is accessed. Indexing is optional structure used to improve query performance. For example, if you want to reference all pages in a book that discusses a certain topic, you first refer to the index, which lists all the topics alphabetically and are then referred to one or more specific page numbers. In similar fashion, we have created index.

In the baseball stats dataset, we have created below indexes to improve the performance of various tables.

4.1.SIMPLE QUERY TO SHOW THE COST DUE TO INDEXING

4.1.1. BEFORE DOING INDEXING QUERY 1

```
SELECT * FROM TEAMS
WHERE TEAMID = 'PH4'
```

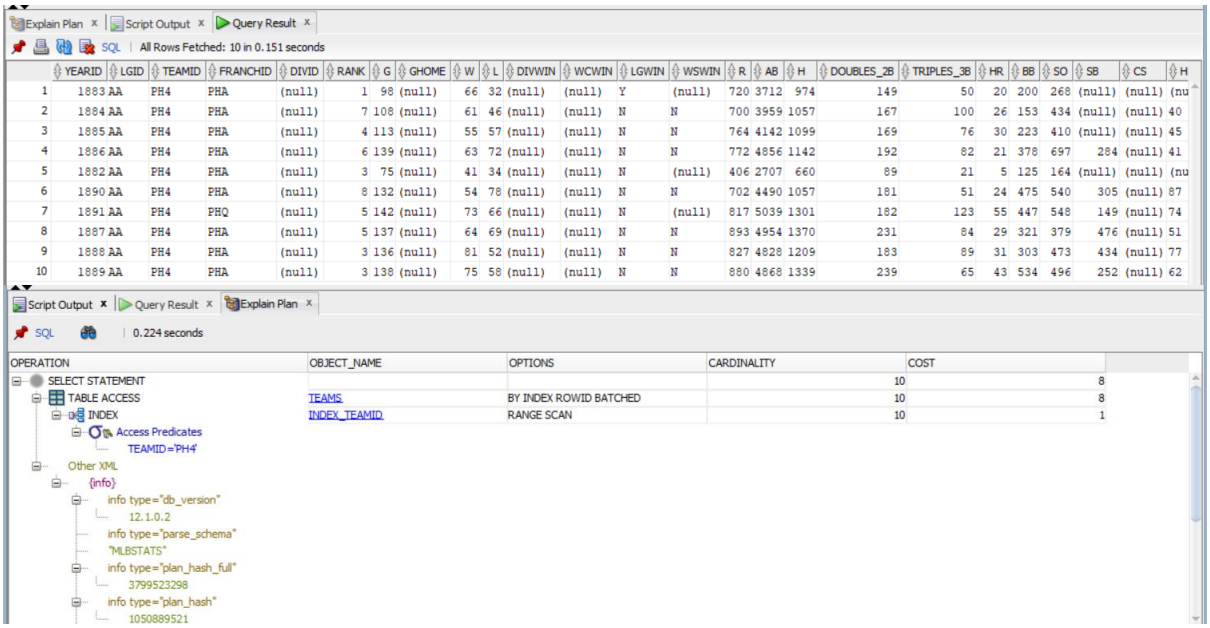


Creating Index to the Teams ID

CREATE INDEX INDEX_TEAMID ON TEAMS("TEAMID")

4.1.2. AFTER INDEXING QUERY 1

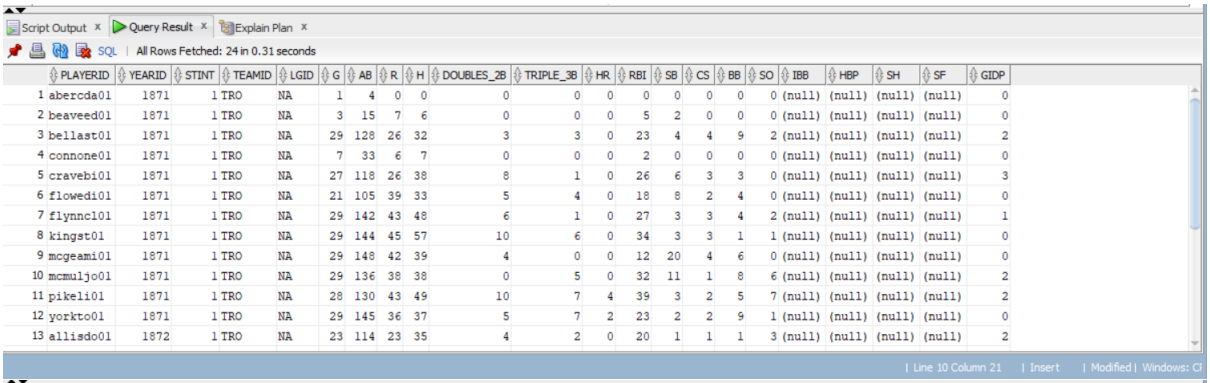
```
SELECT * FROM TEAMS
WHERE TEAMID = 'PH4'
```



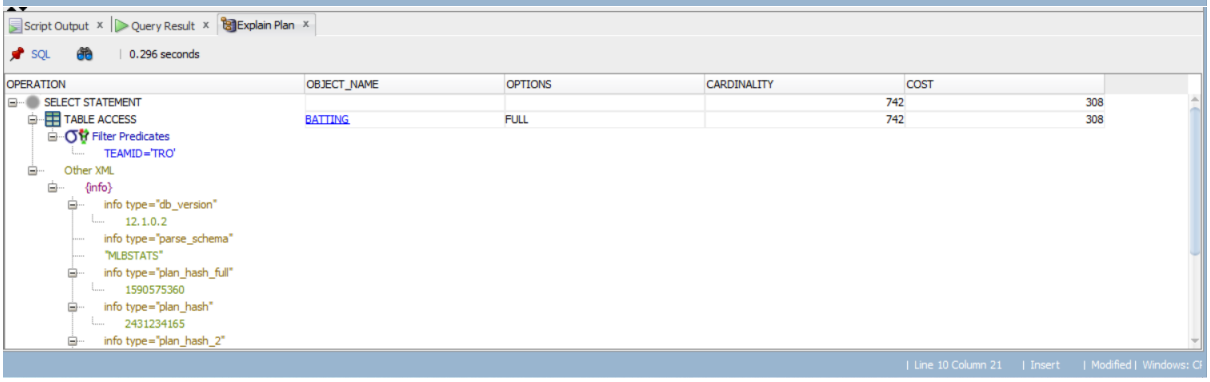
	YEARID	LGID	TEAMID	FRANCHID	DIVID	RANK	G	GHOME	W	L	DIVWIN	WCWIN	LGWIN	WSWIN	R	AB	H	DOUBLES_2B	TRIPLES_3B	HR	BB	SO	SB	CS	H
1	1883	AA	PH4	PHA	(null)	1	98	(null)	66	32	(null)	(null)	Y	(null)	720	3712	974	149	50	20	200	268	(null)	(null)	(nu
2	1884	AA	PH4	PHA	(null)	7	108	(null)	61	46	(null)	(null)	N	N	700	3959	1057	167	100	26	153	434	(null)	(null)	40
3	1885	AA	PH4	PHA	(null)	4	113	(null)	55	57	(null)	(null)	N	N	764	4142	1099	169	76	30	223	410	(null)	(null)	45
4	1886	AA	PH4	PHA	(null)	6	139	(null)	63	72	(null)	(null)	N	N	772	4856	1142	192	82	21	378	697	284	(null)	41
5	1882	AA	PH4	PHA	(null)	3	75	(null)	41	34	(null)	(null)	N	(null)	406	2707	660	89	21	5	125	164	(null)	(null)	(nu
6	1890	AA	PH4	PHA	(null)	8	132	(null)	54	78	(null)	(null)	N	N	702	4490	1057	181	51	24	475	540	305	(null)	87
7	1891	AA	PH4	PHQ	(null)	5	142	(null)	73	66	(null)	(null)	N	(null)	817	5039	1301	182	123	55	447	548	149	(null)	74
8	1887	AA	PH4	PHA	(null)	5	137	(null)	64	69	(null)	(null)	N	N	893	4954	1370	231	84	29	321	379	476	(null)	51
9	1888	AA	PH4	PHA	(null)	3	136	(null)	81	52	(null)	(null)	N	N	827	4828	1209	183	89	31	303	473	434	(null)	77
10	1889	AA	PH4	PHA	(null)	3	138	(null)	75	58	(null)	(null)	N	N	880	4868	1339	239	65	43	534	496	252	(null)	62

4.1.3. BEFORE INDEXING QUERY 2

SELECT * FROM BATTING
WHERE TEAMID = 'TRO'



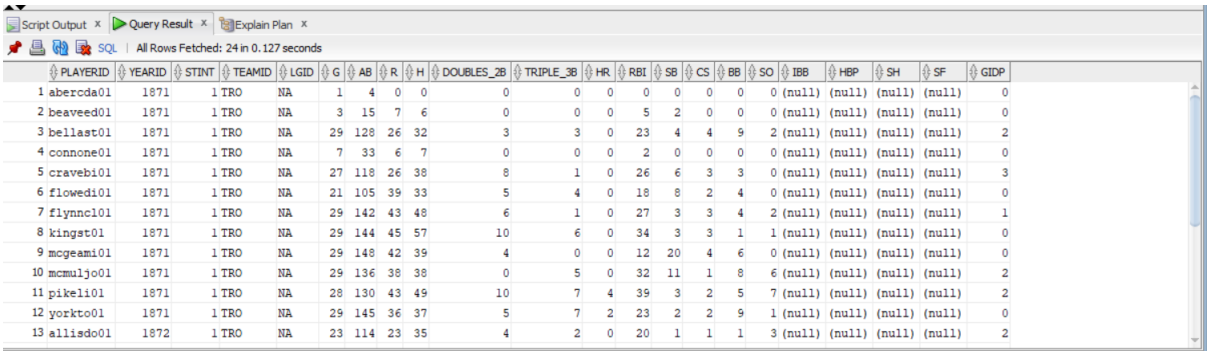
	PLAYERID	YEARID	STINT	TEAMID	LGID	G	AB	R	H	DOUBLES_2B	TRIPLE_3B	HR	RBI	SB	CS	BB	SO	IBB	HBP	SH	SF	GIDP
1	abercda01	1871	1	TRO	NA	1	4	0	0	0	0	0	0	0	0	0	0	(null)	(null)	(null)	(null)	0
2	beaveed01	1871	1	TRO	NA	3	15	7	6	0	0	0	5	2	0	0	0	(null)	(null)	(null)	(null)	0
3	bellast01	1871	1	TRO	NA	29	128	26	32	3	3	0	23	4	4	9	2	(null)	(null)	(null)	(null)	2
4	connone01	1871	1	TRO	NA	7	33	6	7	0	0	0	2	0	0	0	0	(null)	(null)	(null)	(null)	0
5	cravebi01	1871	1	TRO	NA	27	118	26	38	8	1	0	26	6	3	3	0	(null)	(null)	(null)	(null)	3
6	flowedi01	1871	1	TRO	NA	21	105	39	33	5	4	0	18	8	2	4	0	(null)	(null)	(null)	(null)	0
7	flynnci01	1871	1	TRO	NA	29	142	43	48	6	1	0	27	3	3	4	2	(null)	(null)	(null)	(null)	1
8	kingst01	1871	1	TRO	NA	29	144	45	57	10	6	0	34	3	3	1	1	(null)	(null)	(null)	(null)	0
9	mcgeami01	1871	1	TRO	NA	29	148	42	39	4	0	0	12	20	4	6	0	(null)	(null)	(null)	(null)	0
10	mcmuljo01	1871	1	TRO	NA	29	136	38	38	0	5	0	32	11	1	8	6	(null)	(null)	(null)	(null)	2
11	pikeli01	1871	1	TRO	NA	28	130	43	49	10	7	4	39	3	2	5	7	(null)	(null)	(null)	(null)	2
12	yorkto01	1871	1	TRO	NA	29	145	36	37	5	7	2	23	2	2	9	1	(null)	(null)	(null)	(null)	0
13	allisdo01	1872	1	TRO	NA	23	114	23	35	4	2	0	20	1	1	1	3	(null)	(null)	(null)	(null)	2



CREATE INDEX INDEX_TEAM_ID ON BATTING("TEAMID")

4.1.4. AFTER INDEXING QUERY 2

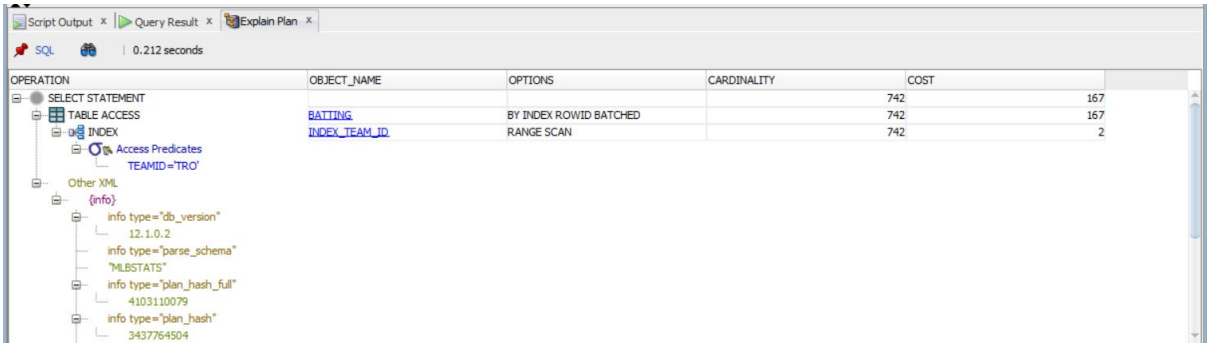
```
SELECT * FROM BATTING
WHERE TEAMID = 'TRO'
```



Script Output | Query Result | Explain Plan

SQL | All Rows Fetched: 24 in 0.127 seconds

	PLAYERID	YEARID	STINT	TEAMID	LGID	G	AB	R	H	DOUBLES_2B	TRIPLE_3B	HR	RBI	SB	CS	BB	SO	IBB	HBP	SH	SF	GIDP
1	abercda01	1871	1	TRO	NA	1	4	0	0	0	0	0	0	0	0	0	0	(null)	(null)	(null)	(null)	0
2	beaveed01	1871	1	TRO	NA	3	15	7	6	0	0	0	5	2	0	0	0	(null)	(null)	(null)	(null)	0
3	bellast01	1871	1	TRO	NA	29	128	26	32	3	3	0	23	4	4	9	2	(null)	(null)	(null)	(null)	2
4	connone01	1871	1	TRO	NA	7	33	6	7	0	0	0	2	0	0	0	0	(null)	(null)	(null)	(null)	0
5	cravebi01	1871	1	TRO	NA	27	118	26	38	8	1	0	26	6	3	3	0	(null)	(null)	(null)	(null)	3
6	flowedi01	1871	1	TRO	NA	21	105	39	33	5	4	0	18	8	2	4	0	(null)	(null)	(null)	(null)	0
7	flynnc101	1871	1	TRO	NA	29	142	43	48	6	1	0	27	3	3	4	2	(null)	(null)	(null)	(null)	1
8	kingst01	1871	1	TRO	NA	29	144	45	57	10	6	0	34	3	3	1	1	(null)	(null)	(null)	(null)	0
9	mcgeami01	1871	1	TRO	NA	29	148	42	39	4	0	0	12	20	4	6	0	(null)	(null)	(null)	(null)	0
10	mcmuljo01	1871	1	TRO	NA	29	136	38	38	0	5	0	32	11	1	8	6	(null)	(null)	(null)	(null)	2
11	pikeli01	1871	1	TRO	NA	28	130	43	49	10	7	4	39	3	2	5	7	(null)	(null)	(null)	(null)	2
12	yorkto01	1871	1	TRO	NA	29	145	36	37	5	7	2	23	2	2	9	1	(null)	(null)	(null)	(null)	0
13	allisdo01	1872	1	TRO	NA	23	114	23	35	4	2	0	20	1	1	1	3	(null)	(null)	(null)	(null)	2



Script Output | Query Result | Explain Plan

SQL | 0.212 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			742	167
TABLE ACCESS	BATTING	BY INDEX ROWID BATCHED	742	167
INDEX	INDEX_TEAM_ID	RANGE SCAN	742	2

Access Predicates
TEAMID='TRO'

Other XML
{info}
info type="db_version"
12.1.0.2
info type="parse_schema"
"MLBSTATS"
info type="plan_hash_full"
4103110079
info type="plan_hash"
3437764504

5. DATA BASE ADMINISTRATION SCRIPTS

The simple SQL script shows the table and column metadata are examples of Database administration scripts. The database system catalog includes metadata tables and views that describe the implementations of individual schemas.

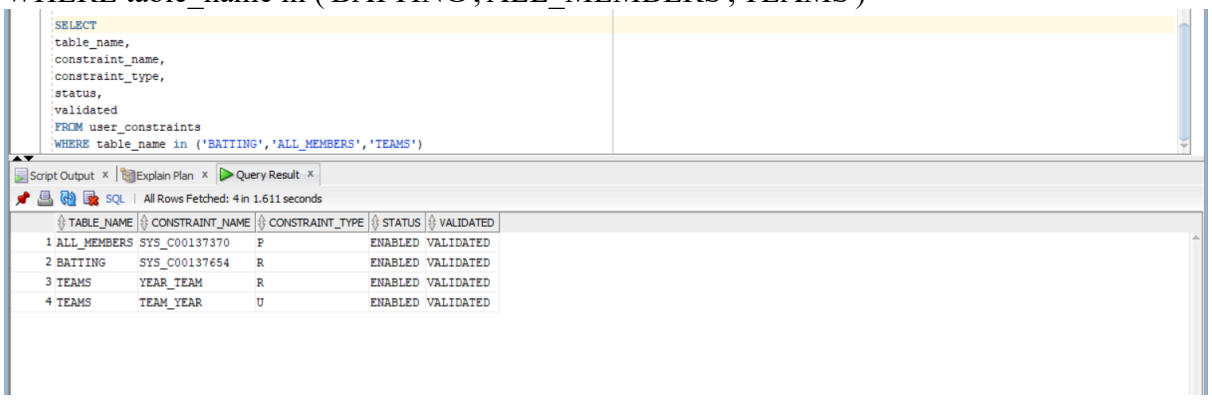
Below are two different views of DBA scripts used for All_Members table:

- List Integrity constraints
- List Index structure

5.1.LIST INTEGRITY CONSTRAINTS

Below query will gives us the list of all constraints which are applied to particular in summary format. Integrity constraints captures the business rules and protect the quality of data. Integrity constraints can be enabled or disabled at any time.

```
SELECT table_name, constraint_name, constraint_type, status, validated
FROM user_constraints
WHERE table_name in ('BATTING','ALL_MEMBERS','TEAMS')
```



5.2.LIST INDEX STRUCTURE

A table can have as many as indexes that improve performance. Monitoring the state and usage of the index structures is an ongoing administrative task.

```
SELECT table_name, index_name, index_type, uniqueness, status
FROM user_indexes
WHERE table_name in ('BATTING','ALL_MEMBERS','TEAMS')
ORDER BY table_name;
```

```
SELECT
table_name,
index_name,
index_type,
uniqueness,
status
FROM user_indexes
WHERE table_name in ('BATTING','ALL_MEMBERS','TEAMS')
ORDER BY table_name;
```

Script Output | Explain Plan | Query Result

SQL | All Rows Fetched: 5 in 0.174 seconds

	TABLE_NAME	INDEX_NAME	INDEX_TYPE	UNIQUENESS	STATUS
1	ALL_MEMBERS	INDEX_BIRTH_COUNTRY	NORMAL	NONUNIQUE	VALID
2	ALL_MEMBERS	SYS_C00137370	NORMAL	UNIQUE	VALID
3	BATTING	INDEX_TEAM_ID	NORMAL	NONUNIQUE	VALID
4	TEAMS	INDEX_TEAMID	NORMAL	NONUNIQUE	VALID
5	TEAMS	TEAM_YEAR	NORMAL	UNIQUE	VALID

6. QUERY WRITING

In this chapter, we wrote multiple queries using different tables:

1. Who is the youngest player query?

```
Select playerid,birthyear,(NameFirst||' '||NameLast) as first_and_last
From all_members
where birthyear is not null
order by birthyear desc;
```

	PLAYERID	BIRTHYEAR	FIRST_AND_LAST
1	francwa01	2001	Wander Franco
2	luciael01	2000	Elvis Luciano
3	garcilu04	2000	Luis Garcia
4	santogr01	1999	Gregory Santos
5	adelljo01	1999	Jo Adell
6	viladry01	1999	Ryan Vilade
7	zerpaan01	1999	Angel Zerpa
8	weathry01	1999	Ryan Weathers

2. Who hit the most homeruns and that player name?

```
Select all_members.playerid,yearid,(NameFirst||' '||NameLast) as first_and_last, HR as
home_runs
From all_members
INNER join batting
on all_members.playerid = batting.playerid
order by HR desc;
```

	PLAYERID	YEARID	FIRST_AND_LAST	HOME_RUNS
1	bondsba01	2001	Barry Bonds	73
2	mcgwima01	1998	Mark McGwire	70
3	sosasa01	1998	Sammy Sosa	66
4	mcgwima01	1999	Mark McGwire	65
5	sosasa01	2001	Sammy Sosa	64
6	sosasa01	1999	Sammy Sosa	63
7	marisro01	1961	Roger Maris	61
8	ruthba01	1927	Babe Ruth	60

3. Highest salaries by year played?

```
select  salaries.salary, salaries.playerid, (NameFirst||' '||NameLast) as
first_and_last,salaries.yearid
from salaries
inner join all_members ON
all_members.playerid = salaries.playerid
order by salary desc;
```

	SALARY	PLAYERID	FIRST_AND_LAST	YEARID
1	33000000	rodrial01	Alex Rodriguez	2009
2	33000000	kershcl01	Clayton Kershaw	2016
3	33000000	rodrial01	Alex Rodriguez	2010
4	32571000	kershcl01	Clayton Kershaw	2015
5	32000000	rodrial01	Alex Rodriguez	2011
6	31799030	greinza01	Zack Greinke	2016

4. Manager with the most wins?

```
select managers.playerid, (namefirst||' '||namelast) as Manager_Name, sum(W)as wins
from managers
left join all_members
on managers.playerid =  all_members.playerid
group by managers.playerid, (namefirst||' '||namelast)
ORDER by wins DESC
```

	PLAYERID	MANAGER_NAME	WINS
1	mackco01	Connie Mack	3719
2	larusto01	Tony LaRussa	2745
3	mcgrajo01	John McGraw	2638
4	coxbo01	Bobby Cox	2457
5	torrejo01	Joe Torre	2249
6	harribu01	Bucky Harris	2158
7	andersp01	Sparky Anderson	2138

5. Who has the most Home runs all the time?

```
select batting.playerid, (namefirst||' '||namelast) as Player_Name, sum(HR)as HR
from batting
```

inner join all_members

on all_members.playerid = batting.playerid

group by batting.playerid, (namefirst||' '||namelast)

ORDER by HR DESC;

	PLAYERID	PLAYER_NAME	HR
1	bondsba01	Barry Bonds	762
2	aaronha01	Hank Aaron	755
3	ruthba01	Babe Ruth	714
4	rodrial01	Alex Rodriguez	696
5	pujolal01	Albert Pujols	679
6	mayswi01	Willie Mays	660
7	griffke02	Ken Griffey	630

6. Which team had more wins and which team had more loss?

select TeamID, SUM(w) as wins from teams

group by teams.teamid

ORDER by wins DESC;

	TEAMID	WINS
1	CHN	11087
2	NYA	10503
3	PIT	10389
4	SLN	10258
5	CIN	10164
6	PHI	9935
7	BOS	9718

select TeamID, SUM(l) as Loss from teams

group by teams.teamid

ORDER by loss DESC;

	TEAMID	LOSS
1	PHI	11112
2	CHN	10521
3	PIT	10251
4	CIN	10105
5	SLN	9731
6	DET	9311

7. Who had more strikeouts?

```
select pitchers.playerid,(namefirst||' '||namelast) as Full_Name, sum(SO) as strike_outs
from pitchers
left join all_members
on pitchers.playerid =  all_members.playerid
group by pitchers.playerid, (namefirst||' '||namelast)
ORDER by  strike_outs DESC;
```

	PLAYERID	FULL_NAME	STRIKE_OUTS
1	ryanno01	Nolan Ryan	5714
2	johnsra05	Randy Johnson	4875
3	clemero02	Roger Clemens	4672
4	carltst01	Steve Carlton	4136
5	blylebe01	Bert Blyleven	3701
6	seaveto01	Tom Seaver	3640
7	suttodo01	Don Sutton	3574

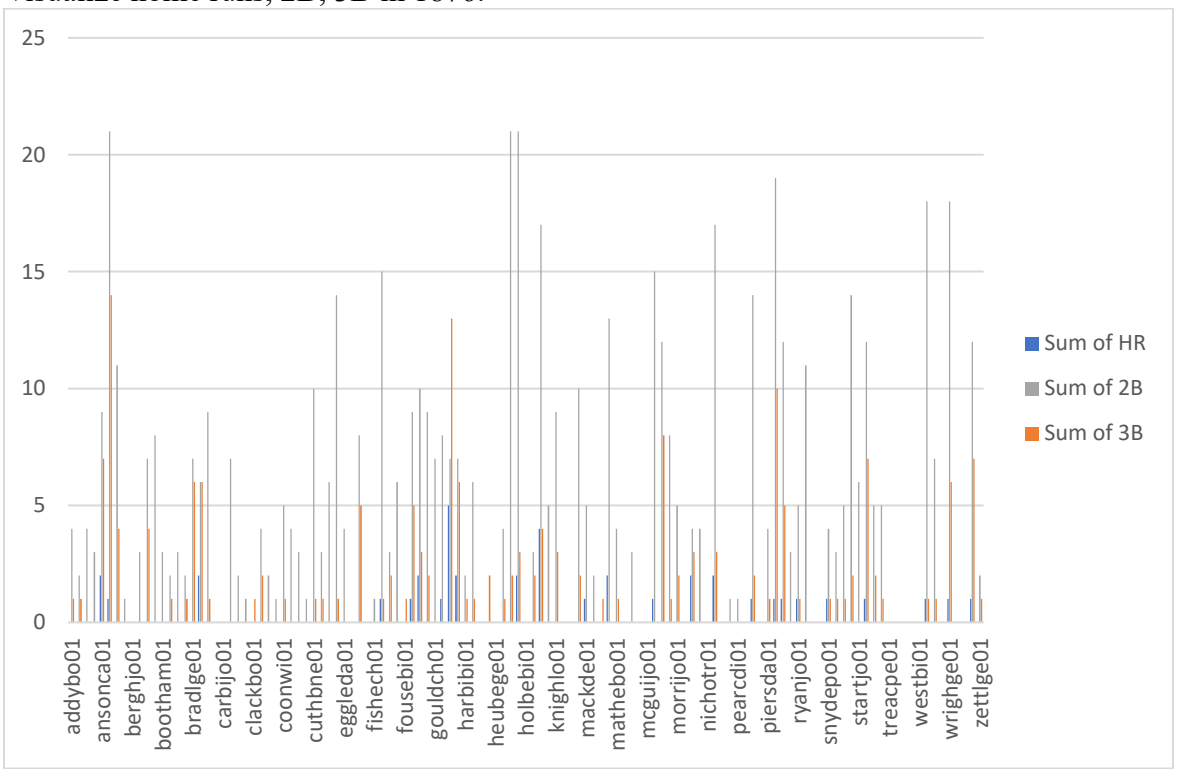
8. How many appearances in G_3B?

```
select appearances.playerid, (namefirst||' '||namelast) as Full_Name , sum(G_3B)
from appearances
inner join all_members
on Appearances.playerid =  all_members.playerid
group by appearances.playerid, (namefirst||' '||namelast)
ORDER by  sum(G_3B) DESC;
```

	PLAYERID	FULL_NAME	SUM(G_3B)
1	robinbr01	Brooks Robinson	2870
2	beltrad01	Adrian Beltre	2759
3	nettlgr01	Graig Nettles	2413
4	gaettga01	Gary Gaetti	2282
5	boggswa01	Wade Boggs	2213
6	schmimi01	Mike Schmidt	2212
7	bellbu01	Buddy Bell	2183

7. DATA VISUALIZATION

Visualize home runs, 2B, 3B in 1876.



8. CONCLUSION

- We created a database in Oracle SQL database which contains base players and their stats
- Created an ERD diagram to show the relations
- We have applied Normalization technique, to reduce the database redundancy
- Written a few sample queries to check whether Database is giving right result. We can confirm that output as expected.
- We used indexing to do the Performance Tuning
- Finally, we have summarized all the important metrics of dataset using data visualization.