Report for Final Project

Title of the Project: NBA Fantasy Player Search

Name of the students: Gahan Sudhir, Yoo Jin Oh, Shruthi Prabahara Sundar

I. Introduction

I.1 Motivation:

NBA fantasy players want to select the best historical roster in their estimation to compete with their friends' rosters. To evaluate who would best fit, these players want access to information such as who played for certain teams, played during a certain time period, shot a certain percentage, scored a certain number of points, etc. Therefore, we wanted to design an application that allowed these players to learn more or enter the information they desired and see which players met their criterion. Additionally, our application provides a comparative lens to evaluate players and see which ones appear more frequently under certain criteria.

I.2 Describe the application

Fantasy players can enter personal information about a type of player they have in mind and receive a list of players that match the info they entered. Additionally, they can enter offensive and defensive information and receive a list of players that met their criterion or exceeded it. Also, at the bottom, players can learn more historical information according to advanced metrics used to evaluate players.

I.3 Describe the overall organization of the report and task assignment for each team member

We will first describe our dataset and the implementation of our application. Then, we will display a few queries and stored procedures, ending with an evaluation demonstrating our filter's accuracy.

II. Our Implementation

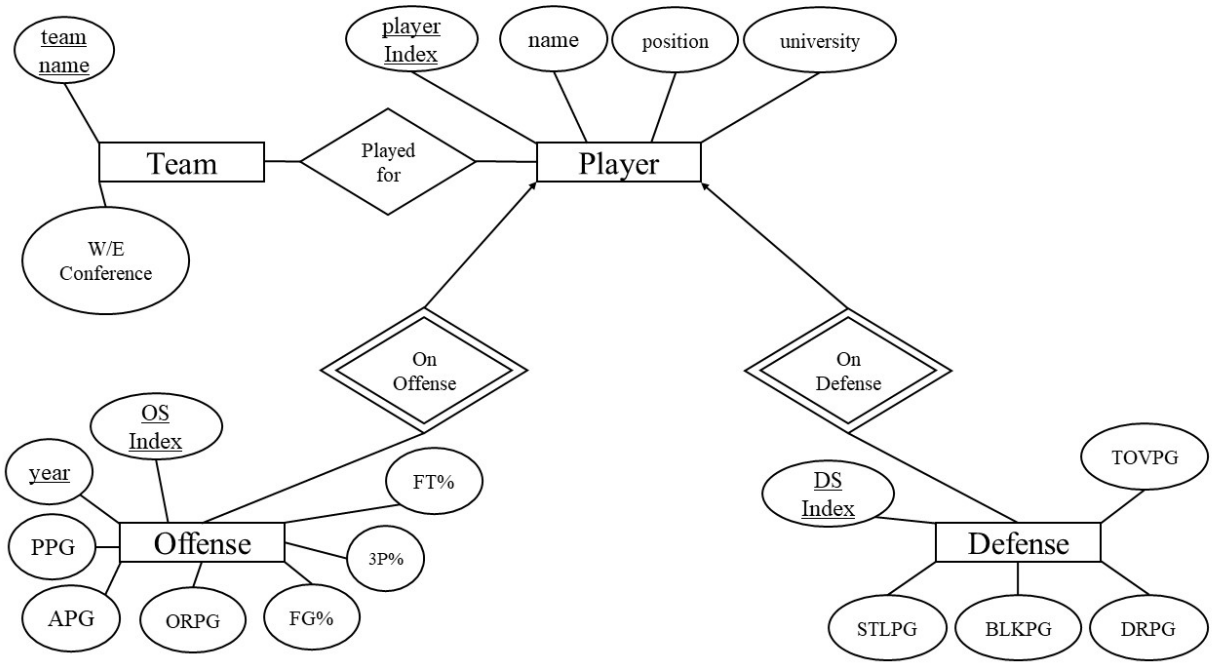
II.1 Description of the system architecture / prototype

The database is located locally and accessed through mysql. Our backend utilized jdbc to send queries to our 'nbaplayers' database with user inputs from our JFrame interface.

II.2 Description of the dataset:

The data set contains 4 tables: Team, Player, Offense, and Defence. Team has 40 records. The candidate key is team name, the other attribute is w/e conference. It is in BCNF with 40 records. The Player table has the candidate key player index, and the other attributes are player index, name, position, and university. It has 2835 records. It is also in BCNF. The third table is Offense with the candidate key OS Index and Year. Other attributes are FT%, 3P%, FG%, ORPG, APF, PPG. It is in 3NF. It has 24690 records. The last tabel is Defense. It has the candidate key DS Index, and other attributes are TOVPG, DRPG, BLKPF, and STLPG. It is in BCNF and has 23980 records.

II.3 ER diagram (final version from the previous checkpoint copied here)



II.4 Relational model (final version from the previous checkpoint copied here)

Team(teamName: String, conference:String) (40 records) in BCNF

Player(playerInd:Int, name:String, position:String, university:String) (2835 records) in BCNF

Offense(OSInd:Int, year:Int, pts:Float, ast:Float, offReb:Float, fgp:Float, tpp:Float, ftp:Float) (24690 records) in 3NF (functional dependency exists to prime attribute ‘year’)

Defense(DSInd:Int, stl:Float, blk:Float, defRev:Float, tov:Float) (23980 records) in BCNF

II.5 Query List

User Input/Output	Query sent to mysql DB	Query Level
name.PNG	select distinct o.player from offstats o, defstats d, players p, team t where o.player like	1

	"Michael%" and o.ind = d.ind and o.player = p.name and o.team = t.name;	
name,team.PNG	select distinct o.player, o.team, t.conf from offstats o, defstats d, players p, team t where o.player like "Michael%" and o.team = "CHI" and o.ind = d.ind and o.player = p.name and o.team = t.name;	2
name,team,ybeg.PNG	select distinct o.player, o.team, t.conf from offstats o, defstats d, players p, team t where o.player like "Michael%" and o.team = "CHI" and o.year > "2000" and o.ind = d.ind and o.player = p.name and o.team = t.name;	2
name,team,ybegend.PNG	select distinct o.player, o.team, t.conf from offstats o, defstats d, players p, team t where o.player like "Michael%" and o.team = "CHI" and o.year > "2000" and o.year < "2010" and o.ind = d.ind and o.player = p.name and o.team = t.name;	2
name,team,ybegend,bo…	select distinct o.player, o.team, t.conf, p.born from offstats o, defstats d, players p, team t where o.player like "Michael%" and o.team = "CHI" and o.year > "1990" and o.year < "2010" and p.born > "1970" and o.ind = d.ind and o.player = p.name and o.team = t.name;	2
ppg.PNG	select o.player, o.year, o.pts/o.games as ppg from offstats o, defstats d, players p	1

	where o.pts/o.games > 30.0 and o.ind = d.ind and o.player = p.name order by o.pts/o.games desc;	
ppg,ast.PNG	select o.player, o.year, o.pts/o.games as ppg, o.ast/o.games as apg from offstats o, defstats d, players p where o.pts/o.games > 30.0 and o.ast/o.games > 5.0 and o.ind = d.ind and o.player = p.name order by o.pts/o.games desc;	1
ppg,ast,oreb.PNG	select o.player, o.year, o.pts/o.games as ppg, o.ast/o.games as apg, o.orb/o.games as orpg from offstats o, defstats d, players p where o.pts/o.games > 30.0 and o.ast/o.games > 5.0 and o.orb/o.games > 1.0 and o.ind = d.ind and o.player = p.name order by o.pts/o.games desc;	1
ppg,ast,oreb,fgp.PNG	select o.player, o.year, o.pts/o.games as ppg, o.ast/o.games as apg, o.orb/o.games as orpg, o.fgp from offstats o, defstats d, players p where o.pts/o.games > 30.0 and o.ast/o.games > 5.0 and o.orb/o.games > 1.0 and o.fgp > 0.5 and o.ind = d.ind and o.player = p.name order by o.pts/o.games desc;	1
ppg,ast,oreb,fgp,tpp.PNG	select o.player, o.year, o.pts/o.games as ppg, o.ast/o.games as apg, o.orb/o.games as orpg, o.fgp, o.tpp from offstats o, defstats d, players p where o.pts/o.games > 30.0 and o.ast/o.games > 5.0 and	1

	o.orb/o.games > 1.0 and o.fgp > 0.5 and o.tpp > 0.3 and o.ind = d.ind and o.player = p.name order by o.pts/o.games desc;	
ppg,ast,oreb,fgp,tpp,ftp…	select o.player, o.year, o.pts/o.games as ppg, o.ast/o.games as apg, o.orb/o.games as orpg, o.fgp, o.tpp, o.ftt as ftp from offstats o, defstats d, players p where o.pts/o.games > 30.0 and o.ast/o.games > 5.0 and o.orb/o.games > 1.0 and o.fgp > 0.5 and o.tpp > 0.3 and o.ftt > 0.8 and o.ind = d.ind and o.player = p.name order by o.pts/o.games desc;	1
ppg,ast,oreb,fgp,tpp,ftp…	select o.player, o.year, o.pts/o.games as ppg, o.ast/o.games as apg, o.orb/o.games as orpg, o.fgp, o.tpp, o.ftt as ftp, d.stl/o.games as stlpg from offstats o, defstats d, players p where o.pts/o.games > 30.0 and o.ast/o.games > 5.0 and o.orb/o.games > 1.0 and o.fgp > 0.5 and o.tpp > 0.3 and o.ftt > 0.8 and d.stl/o.games > 2.5 and o.ind = d.ind and o.player = p.name order by o.pts/o.games desc;	2
ppg,ast,oreb,fgp,tpp,ftp…	select o.player, o.year, o.pts/o.games as ppg, o.ast/o.games as apg, o.orb/o.games as orpg, o.fgp, o.tpp, o.ftt as ftp, d.stl/o.games as stlpg, d.blk/o.games as blkpg from offstats o, defstats d, players p where o.pts/o.games > 30.0 and o.ast/o.games > 5.0 and	2

	o.orb/o.games > 1.0 and o.fgp > 0.5 and o.tpp > 0.3 and o.ftt > 0.8 and d.stl/o.games > 2.5 and d.blk/o.games > 1.0 and o.ind = d.ind and o.player = p.name order by o.pts/o.games desc;	
ppg,ast,oreb,fgp,tpp,ftp…	select o.player, o.year, o.pts/o.games as ppg, o.ast/o.games as apg, o.orb/o.games as orpg, o.fgp, o.tpp, o.ftt as ftp, d.stl/o.games as stlpg, d.blk/o.games as blkpg, d.drb/o.games as drpg from offstats o, defstats d, players p where o.pts/o.games > 30.0 and o.ast/o.games > 5.0 and o.orb/o.games > 1.0 and o.fgp > 0.5 and o.tpp > 0.3 and o.ftt > 0.8 and d.stl/o.games > 2.5 and d.blk/o.games > 1.0 and d.drb/o.games > 3.0 and o.ind = d.ind and o.player = p.name order by o.pts/o.games desc;	2
ppg,ast,oreb,fgp,tpp,ftp…	select o.player, o.year, o.pts/o.games as ppg, o.ast/o.games as apg, o.orb/o.games as orpg, o.fgp, o.tpp, o.ftt as ftp, d.stl/o.games as stlpg, d.blk/o.games as blkpg, d.drb/o.games as drpg, d.tov/o.games as tovpg from offstats o, defstats d, players p where o.pts/o.games > 30.0 and o.ast/o.games > 5.0 and o.orb/o.games > 1.0 and o.fgp > 0.5 and o.tpp > 0.3 and o.ftt > 0.8 and d.stl/o.games > 2.5 and d.blk/o.games > 1.0 and d.drb/o.games > 3.0 and d.tov/o.games < 3.0 and o.ind = d.ind and o.player = p.name	2

	order by o.pts/o.games desc;	
Many more lvl 2 permutations available to be made		2

Stored Procedures

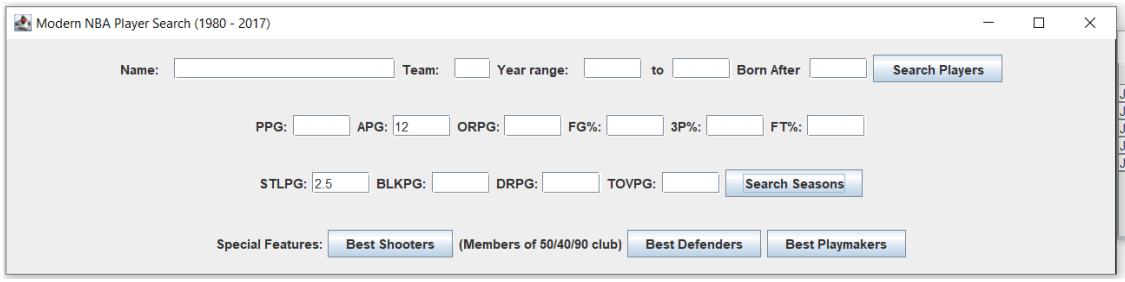
User Input/Output	Procedure in mysql DB
bestShooters.PNG	delimiter $$ drop procedure if exists getShooters; create procedure getShooters() begin select player, year, pts/games as ppg, fgp, tpp, ftt as ftp from offstats where fgp > 0.5 and tpp > 0.4 and ftt > 0.9 and pts/games > 10 order by pts/games desc; end $$ delimiter ;
bestDefenders.PNG	delimiter $$ drop procedure if exists getDefenders; create procedure getDefenders() begin select o.player, o.year, o.pts/o.games as ppg, d.stl/d.games as stlpg, d.blk/d.games as blkpg from defstats d, offstats o where d.ind = o.ind and d.stl/d.games > 2 and d.blk/d.games > 2 order by o.pts/o.games desc; end $$ delimiter ;
bestPlaymakers.PNG	delimiter $$ drop procedure if exists getPlaymakers; create procedure getPlaymakers() begin select o.player, o.year, o.pts/o.games as ppg, o.ast/o.games as apg, o.orb/o.games from defstats d, offstats o

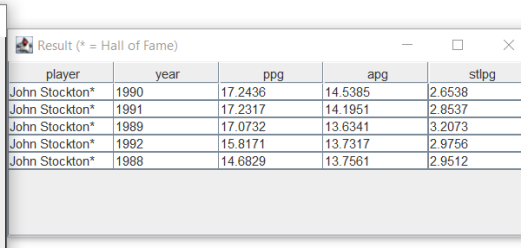
	where d.ind = o.ind and o.pts/o.games > 27 and o.ast/o.games > 5 and o.orb/o.games > 1 order by o.pts/o.games desc; end $$ delimiter ;

II.6 Evaluation: describe how you test your application (e.g, create testing scenarios or queries or something else, running your application through these scenarios/queries/etc.., checking the returned results and see how the results make sense or not).

TestAccuracy.mkv (Video of first test)

Second test on John Stockton:

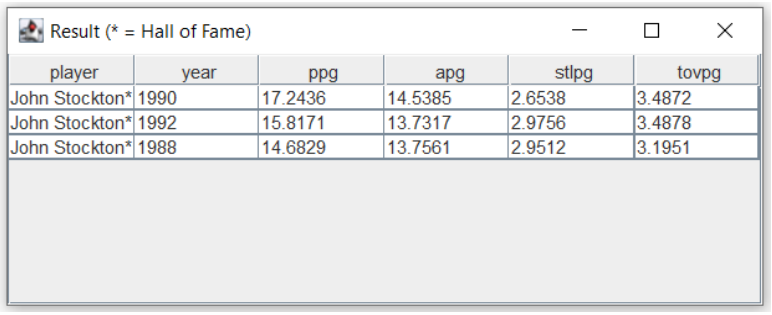




player	year	ppg	apg	stlpg
John Stockton*	1990	17.2436	14.5385	2.6538
John Stockton*	1991	17.2317	14.1951	2.8537
John Stockton*	1989	17.0732	13.6341	3.2073
John Stockton*	1992	15.8171	13.7317	2.9756
John Stockton*	1988	14.6829	13.7561	2.9512

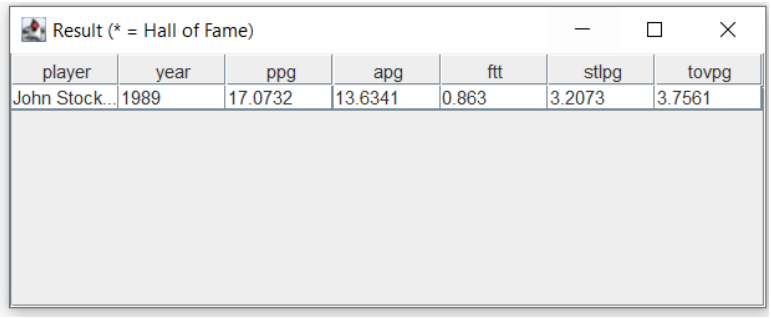
Season	Age	Tm	Lg	Pos	G	GS	MP	FG	FGA	FG%	3P	3PA	3P%	2P	2PA	2P%	eFG%	FT	FTA	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
1984-85	22	UTA	NBA	PG	**82**	5	18.2	1.9	4.1	.471	0.0	0.1	.182	1.9	3.9	.481	.474	1.7	2.4	.736	0.3	1.0	1.3	5.1	1.3	0.1	1.8	2.5	5.6
1985-86	23	UTA	NBA	PG	82	38	23.6	2.8	5.7	.489	0.0	0.2	.133	2.8	5.5	.501	.491	2.1	2.5	.839	0.4	1.8	2.2	7.4	1.9	0.1	2.0	2.8	7.7
1986-87	24	UTA	NBA	PG	**82**	2	22.7	2.8	5.6	.499	0.1	0.5	.179	2.7	5.2	.528	.506	2.2	2.8	.782	0.4	1.5	1.8	8.2	2.2	0.2	2.0	2.7	7.9
1987-88	25	UTA	NBA	PG	82	79	34.7	5.5	9.6	.574	0.3	0.8	.358	5.2	8.8	.594	.589	3.3	4.0	.840	0.7	2.2	2.9	**13.8**	3.0	0.2	3.2	3.0	14.7
1988-89 ★	26	UTA	NBA	PG	**82**	82	38.7	6.1	11.3	.538	0.2	0.8	.242	5.9	10.5	.561	.547	4.8	5.5	.863	1.0	2.0	3.0	**13.6**	**3.2**	0.2	3.8	2.9	17.1
1989-90 ★	27	UTA	NBA	PG	78	78	37.4	6.1	11.8	.514	0.6	1.4	.416	5.4	10.3	.528	.540	4.5	5.5	.819	0.7	1.9	2.6	**14.5**	2.7	0.2	3.5	3.0	17.2
1990-91 ★	28	UTA	NBA	PG	**82**	82	37.8	6.0	11.9	.507	0.7	2.0	.345	5.3	9.9	.541	.537	4.4	5.3	.836	0.6	2.3	2.9	**14.2**	2.9	0.2	3.6	2.8	17.2
1991-92 ★	29	UTA	NBA	PG	82	82	36.6	5.5	11.5	.482	1.0	2.5	.407	4.5	9.0	.503	.527	3.8	4.5	.842	0.8	2.5	3.3	**13.7**	**3.0**	0.3	3.5	2.9	15.8
1992-93 ★	30	UTA	NBA	PG	82	82	34.9	5.3	11.0	.486	0.9	2.3	.385	4.5	8.7	.513	.526	3.6	4.5	.798	0.8	2.1	2.9	**12.0**	2.4	0.3	3.2	2.7	15.1

1) If tovpg is < 3.5, then the ‘89 and ‘91 seasons should be removed from the results:



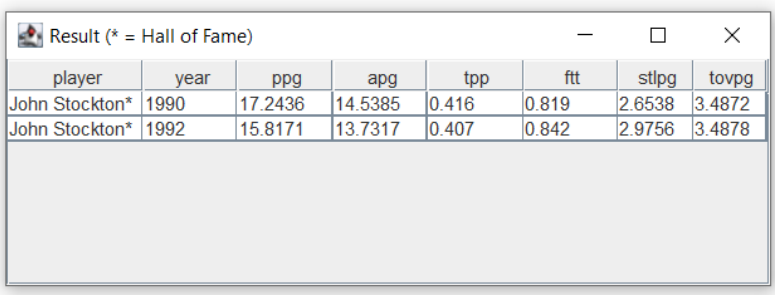
player	year	ppg	apg	stlpg	tovpg
John Stockton*	1990	17.2436	14.5385	2.6538	3.4872
John Stockton*	1992	15.8171	13.7317	2.9756	3.4878
John Stockton*	1988	14.6829	13.7561	2.9512	3.1951

2) If ftp% is > 85%, then only the ‘89 season should remain



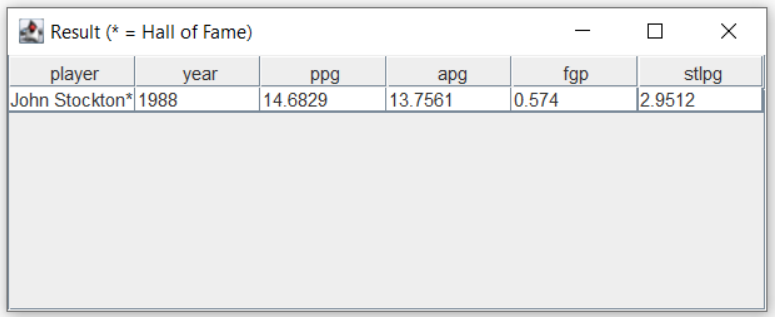
player	year	ppg	apg	ftt	stlpg	tovpg
John Stock...	1989	17.0732	13.6341	0.863	3.2073	3.7561

3) If 3P% is > 40%, only seasons 90 and 92 should remain



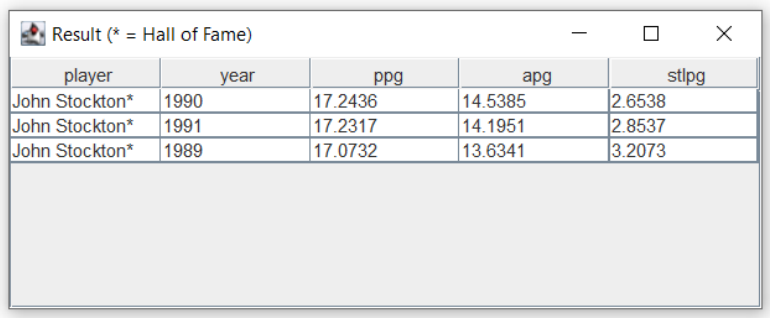
player	year	ppg	apg	tpp	ftt	stlpg	tovpg
John Stockton*	1990	17.2436	14.5385	0.416	0.819	2.6538	3.4872
John Stockton*	1992	15.8171	13.7317	0.407	0.842	2.9756	3.4878

4) If FG% is > 55% only season ‘88 should remain



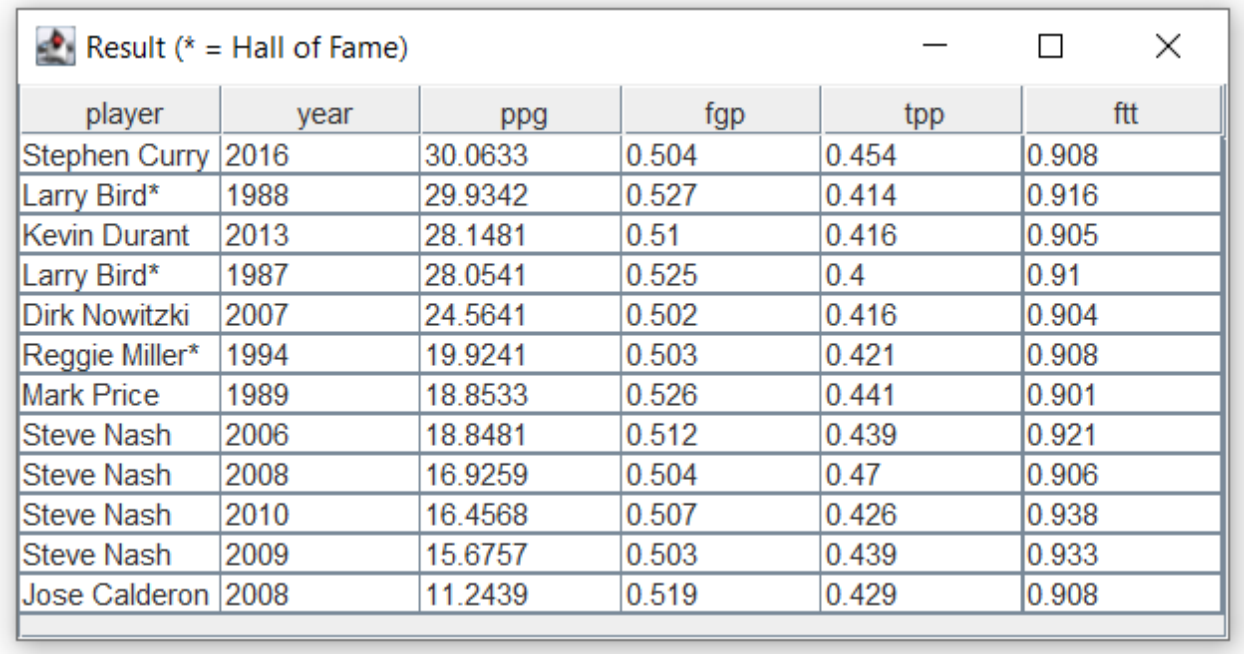
player	year	ppg	apg	fgp	stlpg
John Stockton*	1988	14.6829	13.7561	0.574	2.9512

5) If ppg is > 17, ‘89, ‘90, and ‘91 should remain



player	year	ppg	apg	stlpg
John Stockton*	1990	17.2436	14.5385	2.6538
John Stockton*	1991	17.2317	14.1951	2.8537
John Stockton*	1989	17.0732	13.6341	3.2073

Our 50/40/90 filter compared to Wikipedia's



player	year	ppg	fgp	tpp	ftt
Stephen Curry	2016	30.0633	0.504	0.454	0.908
Larry Bird*	1988	29.9342	0.527	0.414	0.916
Kevin Durant	2013	28.1481	0.51	0.416	0.905
Larry Bird*	1987	28.0541	0.525	0.4	0.91
Dirk Nowitzki	2007	24.5641	0.502	0.416	0.904
Reggie Miller*	1994	19.9241	0.503	0.421	0.908
Mark Price	1989	18.8533	0.526	0.441	0.901
Steve Nash	2006	18.8481	0.512	0.439	0.921
Steve Nash	2008	16.9259	0.504	0.47	0.906
Steve Nash	2010	16.4568	0.507	0.426	0.938
Steve Nash	2009	15.6757	0.503	0.439	0.933
Jose Calderon	2008	11.2439	0.519	0.429	0.908

Player	Season	GP	FG	FGA	FG%	3P	3PA	3P%	FT	FTA	FT%	PTS	PPG	Ref.
Larry Bird	1986–87	74	786	1,497	53% (.525)	90	225	40% (.400)	414	455	91% (.910)	2,076	28.1	[9]
Larry Bird (2)	1987–88	76	881	1,672	53% (.527)	98	237	41% (.414)	415	453	92% (.916)	2,275	29.9	[9]
Mark Price	1988–89	75	529	1,006	53% (.526)	93	211	44% (.441)	263	292	90% (.901)	1,414	18.9	[10]
Reggie Miller	1993–94	79	524	1,042	50% (.503)	123	292	42% (.421)	403	444	91% (.908)	1,574	19.9	[11]
Steve Nash	2005–06	79	541	1,056	51% (.512)	150	342	44% (.439)	257	279	92% (.921)	1,489	18.8	[5]
Dirk Nowitzki	2006–07	78	673	1,341	50% (.502)	72	173	42% (.416)	498	551	90% (.904)	1,916	24.6	[12]
Steve Nash (2)	2007–08	81	485	962	50% (.504)	179	381	47% (.470)	222	245	91% (.906)	1,371	16.9	[5]
Steve Nash (3)	2008–09	74	428	851	50% (.503)	108	246	44% (.439)	196	210	93% (.933)	1,160	15.7	[5]
Steve Nash (4)	2009–10	81	499	985	51% (.507)	124	291	43% (.426)	211	225	94% (.938)	1,333	16.5	[5]
Kevin Durant	2012–13	81	731	1,433	51% (.510)	139	334	42% (.416)	679	750	91% (.905)	2,280	28.1	[13]
Stephen Curry	2015–16	79	805	1,598	50% (.504)	402	886	45% (.454)	363	400	91% (.908)	2,375	30.1	[14]
Malcolm Brogdon	2018–19	64	378	748	51% (.505)	104	244	43% (.426)	141	152	93% (.928)	1,001	15.6	[15]
Kyrie Irving	2020–21	54	549	1,086	51% (.506)	152	378	40% (.402)	201	218	92% (.922)	1,451	26.9	[16]

III. Conclusion

What do you learn from this project (both interesting and uninteresting points)? Have you found any relevant database knowledge you have learned in this course helpful and have you encountered any database relevant issues that have been discussed in this course?

This project showed to us how very specific, useful information can be obtained from massive datasets by taking the time to intelligently design a relational schema, prefilter and normalize our data, and understand sql query structure to create increasingly more complex queries.

IV. References (if any)

https://www.kaggle.com/datasets/drgilermo/nba-players-stats (datasets)