Rapport TD 1

Cassandra

Table des matières

1. Data cleaning and export

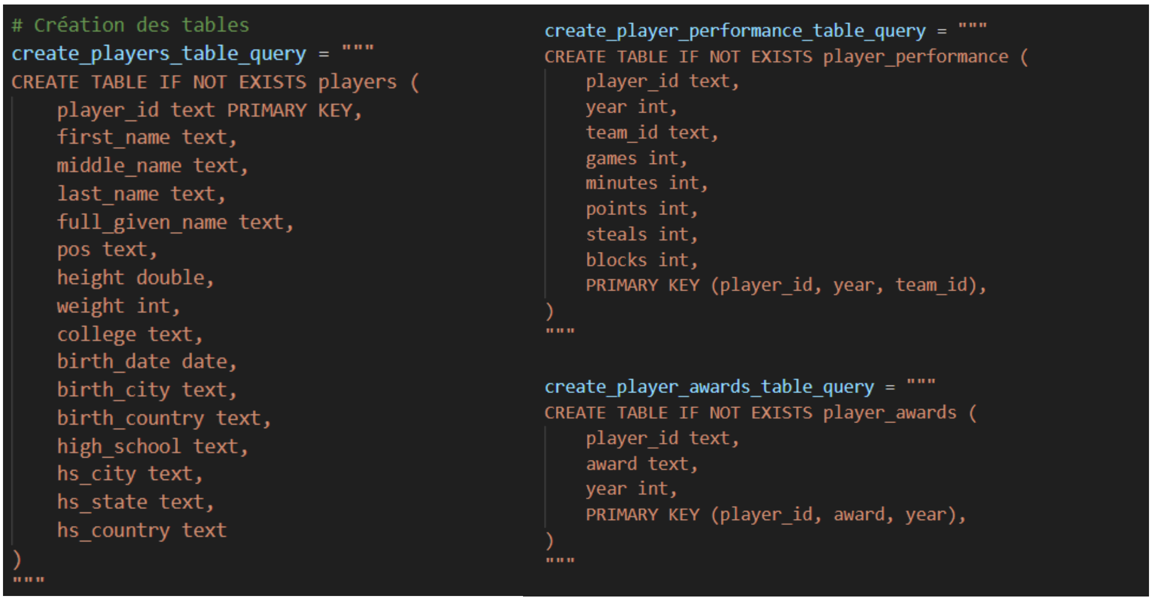
We chose the *basketball_women.json* dataset, its difficulty is 2 (complex dataset) so we have to do 6 simple queries, 2 complex queries, 1 query.

First of all, we checked and we find the format of the json in not valuable. We needed to change so we put everything in a list [] and we add a coma at the end of the first line.

Our data is a json, but we needed to find something to be able to insert it but during our TD we saw that the JSON of the course contains for each new objects “INSERT INTO table ‘(…)’”, so we decided to make a python script that will insert the data into the database.

Moreover, there were a few problems in the dataset. First, some dates were **0000-00-00** and Cassandra can’t recognize it as a date. We decided to replace them with **null**. Then, the symbol “**&**” is not recognized also, so we modified the “**&**” value by “**and**”. Finally, there was some data with “**‘**” in the name and we needed to fix it because it is considered as the end of string, so we replaced it with a space.

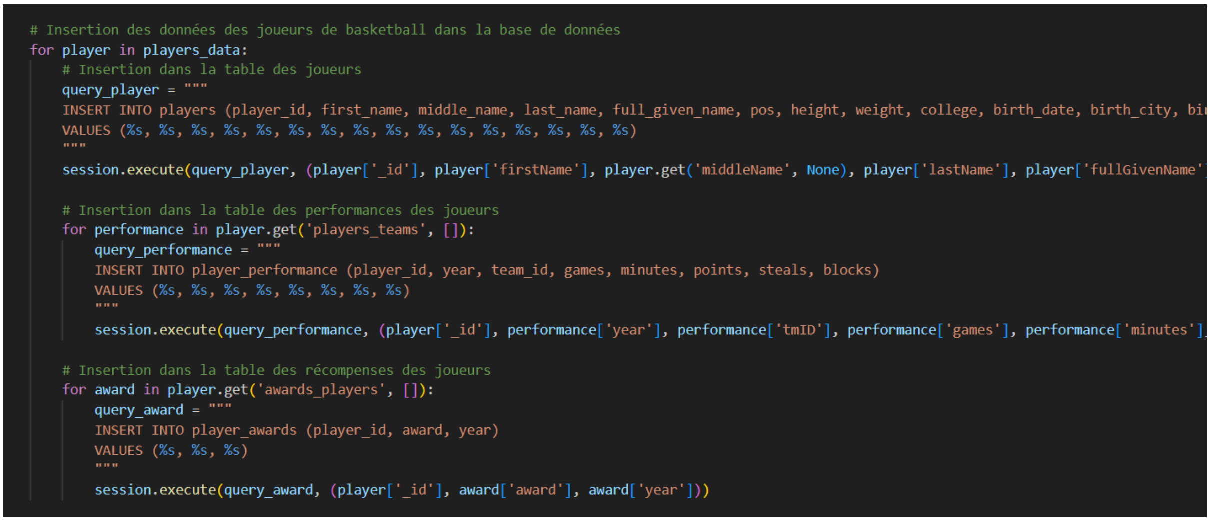
Furthermore, we can create our table “**players**”,**“player_performance”** and **“player_awards”**.



```
# Création des tables
create_players_table_query = """
CREATE TABLE IF NOT EXISTS players (
    player_id text PRIMARY KEY,
    first_name text,
    middle_name text,
    last_name text,
    full_given_name text,
    pos text,
    height double,
    weight int,
    college text,
    birth_date date,
    birth_city text,
    birth_country text,
    high_school text,
    hs_city text,
    hs_state text,
    hs_country text
)
"""
```

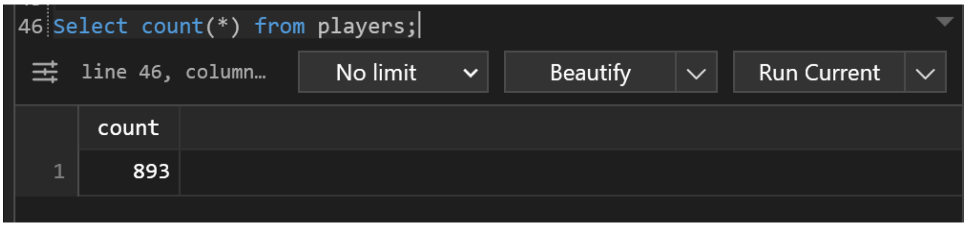
```
create_player_performance_table_query = """
CREATE TABLE IF NOT EXISTS player_performance (
    player_id text,
    year int,
    team_id text,
    games int,
    minutes int,
    points int,
    steals int,
    blocks int,
    PRIMARY KEY (player_id, year, team_id),
)
"""

create_player_awards_table_query = """
CREATE TABLE IF NOT EXISTS player_awards (
    player_id text,
    award text,
    year int,
    PRIMARY KEY (player_id, award, year),
)
"""
```

Then we made a python script that insert our values. The principle is very simple, we first create the table player with the main characteristics. Then, for all the object in performance and award, we load the data into the corresponding tables.



```
# Insertion des données des joueurs de basketball dans la base de données
for player in players_data:
    # Insertion dans la table des joueurs
    query_player = """
    INSERT INTO players (player_id, first_name, middle_name, last_name, full_given_name, pos, height, weight, college, birth_date, birth_city, bi
    VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
    """
    session.execute(query_player, (player['_id'], player['firstName'], player.get('middleName', None), player['lastName'], player['fullGivenName']

    # Insertion dans la table des performances des joueurs
    for performance in player.get('players_teams', []):
        query_performance = """
        INSERT INTO player_performance (player_id, year, team_id, games, minutes, points, steals, blocks)
        VALUES (%s, %s, %s, %s, %s, %s, %s, %s)
        """
        session.execute(query_performance, (player['_id'], performance['year'], performance['tmID'], performance['games'], performance['minutes']]

    # Insertion dans la table des récompenses des joueurs
    for award in player.get('awards_players', []):
        query_award = """
        INSERT INTO player_awards (player_id, award, year)
        VALUES (%s, %s, %s)
        """
        session.execute(query_award, (player['_id'], award['award'], award['year']))
```

We can verify if we have inserted all our values. We have our 893 datas !

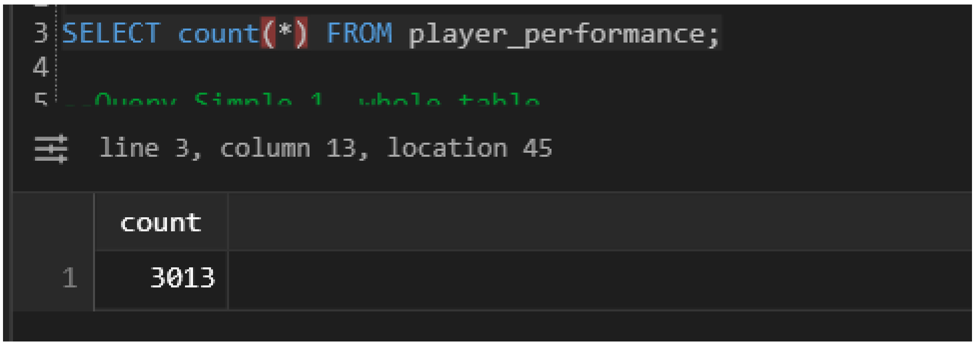


```
Select count(*) from players;
```

line 46, column...

	count
1	893

We have 3013 data for the table player_perfomance with the information of the score, rebound etc..

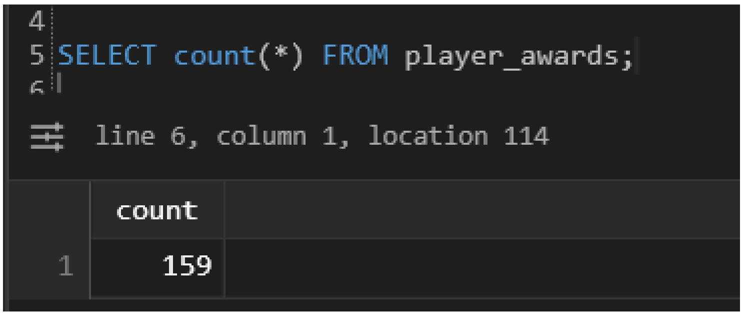


```
SELECT count(*) FROM player_performance;

-- Query Simple 1 - whole table
```

line 3, column 13, location 45

	count
1	3013

And we have 159 datas for the table player_award.



```

SELECT count(*) FROM player_awards;

```

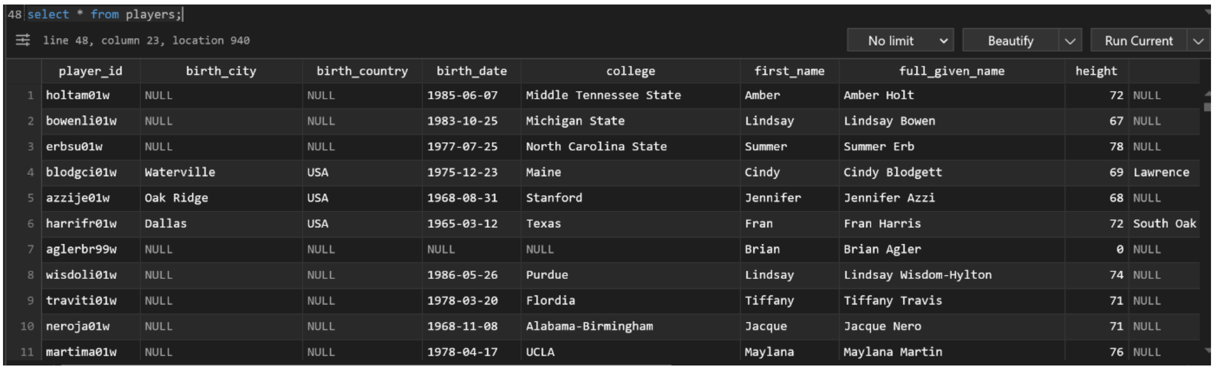
line 6, column 1, location 114

	count
1	159

2. Simple queries (6 queries)

1) Our first query prints the full table of players.

To make this query, we use a SELECT with * that mean all the parameters, and from the table of our choice, in this case players.

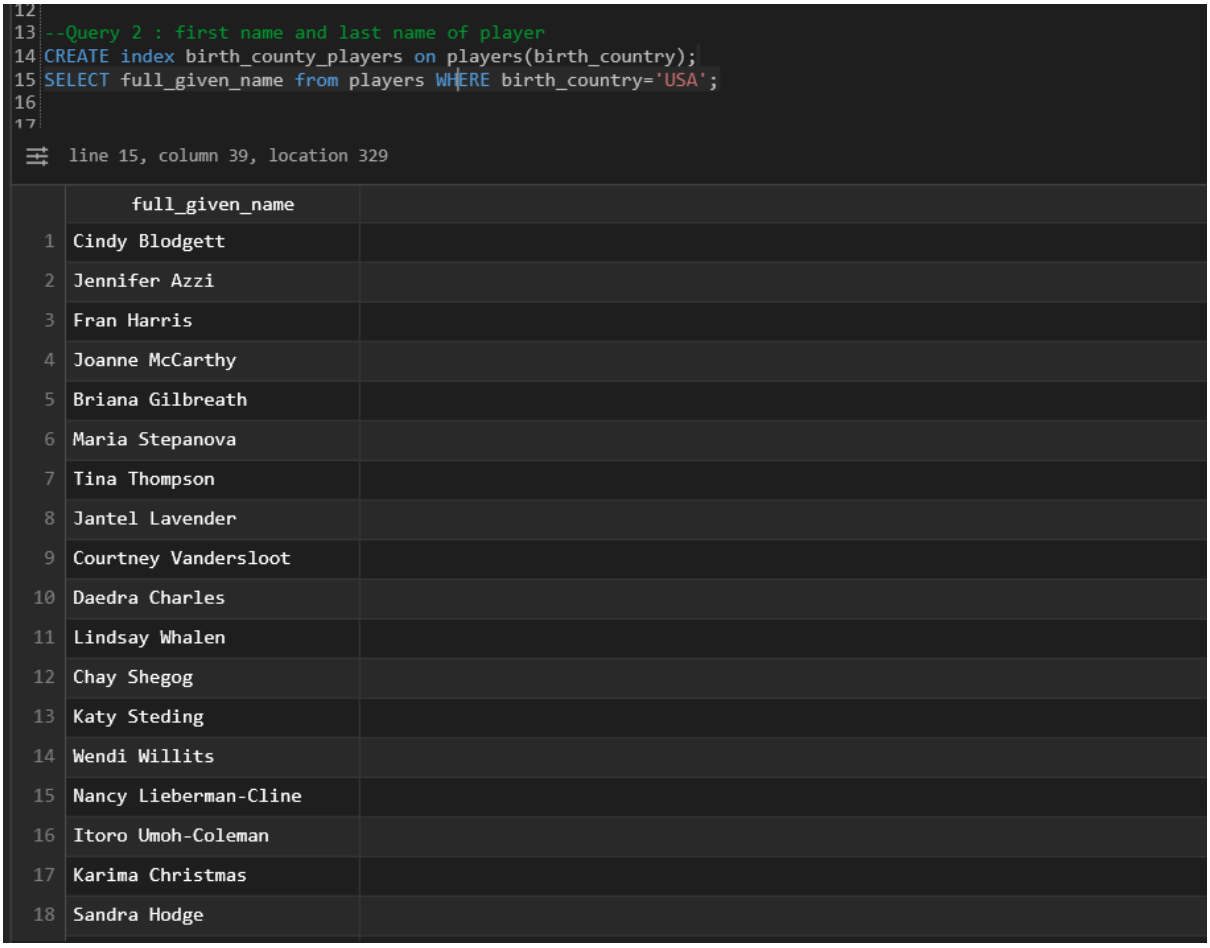


```
select * from players;
```

	player_id	birth_city	birth_country	birth_date	college	first_name	full_given_name	height	
1	holtam01w	NULL	NULL	1985-06-07	Middle Tennessee State	Amber	Amber Holt	72	NULL
2	bowenli01w	NULL	NULL	1983-10-25	Michigan State	Lindsay	Lindsay Bowen	67	NULL
3	erbsu01w	NULL	NULL	1977-07-25	North Carolina State	Summer	Summer Erb	78	NULL
4	blodgci01w	Waterville	USA	1975-12-23	Maine	Cindy	Cindy Blodgett	69	Lawrence
5	azzije01w	Oak Ridge	USA	1968-08-31	Stanford	Jennifer	Jennifer Azzi	68	NULL
6	harrifr01w	Dallas	USA	1965-03-12	Texas	Fran	Fran Harris	72	South Oak
7	aglerbr99w	NULL	NULL	NULL	NULL	Brian	Brian Agler	0	NULL
8	wisdoli01w	NULL	NULL	1986-05-26	Purdue	Lindsay	Lindsay Wisdom-Hylton	74	NULL
9	traviti01w	NULL	NULL	1978-03-20	Flordia	Tiffany	Tiffany Travis	71	NULL
10	neroja01w	NULL	NULL	1968-11-08	Alabama-Birmingham	Jacque	Jacque Nero	71	NULL
11	martima01w	NULL	NULL	1978-04-17	UCLA	Maylana	Maylana Martin	76	NULL

2) In the second query we search the full given names of players who were born in the USA.

To do that, we used an index on the birth_country column in order to make the query more optimized. We use WHERE to filter the result with only the players that were born in the country 'USA'.

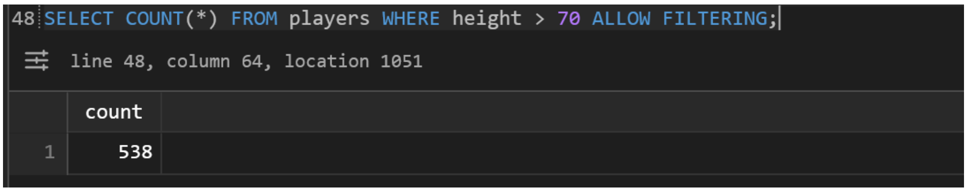


```
--Query 2 : first name and last name of player
CREATE index birth_county_players on players(birth_country);
SELECT full_given_name from players WHERE birth_country='USA';
```

	full_given_name
1	Cindy Blodgett
2	Jennifer Azzi
3	Fran Harris
4	Joanne McCarthy
5	Briana Gilbreath
6	Maria Stepanova
7	Tina Thompson
8	Jantel Lavender
9	Courtney Vandersloot
10	Daedra Charles
11	Lindsay Whalen
12	Chay Shegog
13	Katy Steding
14	Wendi Willits
15	Nancy Lieberman-Cline
16	Itoro Umoh-Coleman
17	Karima Christmas
18	Sandra Hodge

3) In the third query, we search how many players are taller than 70.0 feets.

We use COUNT(*) to print the number of players and we filter with the height >70. Then, we use allow filtering because it is used to permit filtering on non-indexed columns in a query, relaxing the usual requirement for indexed or primary key-based conditions, but it should be used cautiously due to potential performance implications.



```
SELECT COUNT(*) FROM players WHERE height > 70 ALLOW FILTERING;
```

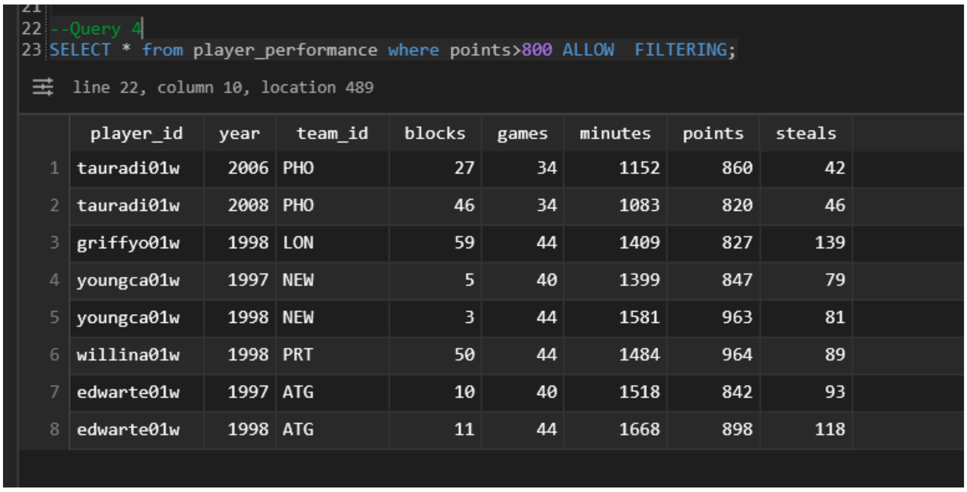
line 48, column 64, location 1051

	count
1	538

We can conclude that 538 women basketball players are taller than 70 feets over 893 in total.

4) In the fourth query, we search for the players that score more than 800 points in one season.

For this query, we use the table player_perfomance that we created with all the performance information of the players. We can see that 3 players are two times in the top 8 because (and only 8 players in the history to shoot more than 800 points in only one season!) it's their points from different season. And even more the first and the second player are the same.



```
--Query 4
SELECT * from player_performance where points>800 ALLOW  FILTERING;
```

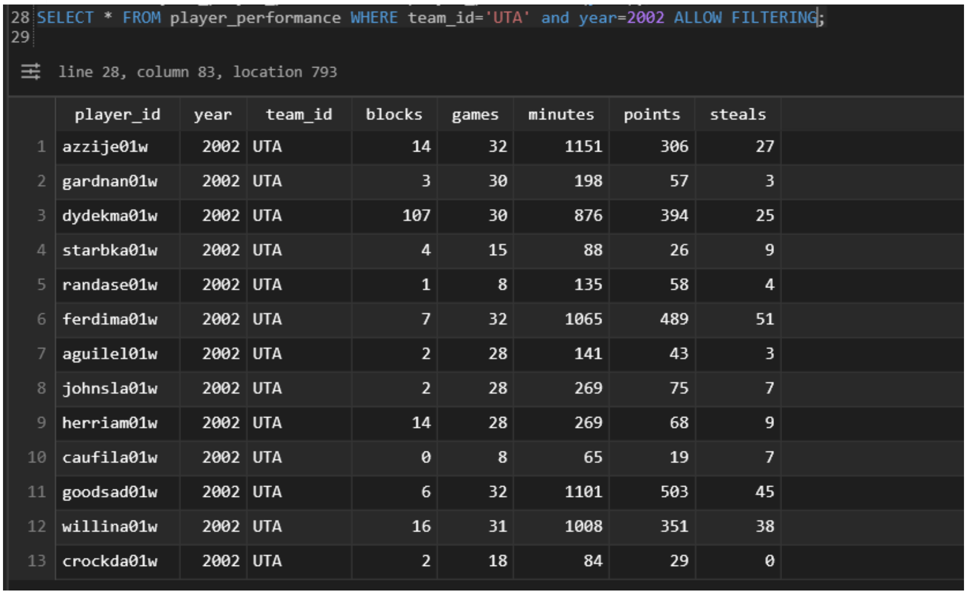
line 22, column 10, location 489

	player_id	year	team_id	blocks	games	minutes	points	steals
1	tauradi01w	2006	PHO	27	34	1152	860	42
2	tauradi01w	2008	PHO	46	34	1083	820	46
3	griffyo01w	1998	LON	59	44	1409	827	139
4	youngca01w	1997	NEW	5	40	1399	847	79
5	youngca01w	1998	NEW	3	44	1581	963	81
6	willina01w	1998	PRT	50	44	1484	964	89
7	edwarte01w	1997	ATG	10	40	1518	842	93
8	edwarte01w	1998	ATG	11	44	1668	898	118

Moreover, we noticed something interesting, the first player to have the most points in a season has played less games and less minutes than the other players! It's just incredible!

5) In the fifth query, we search all the players and stats from the team UTA in 2002

It allowed us to see the entire team's composition and repartition of games.
We used WHERE on team_id to find the USA team and year search the 2002 year.



```
28 SELECT * FROM player_performance WHERE team_id='UTA' and year=2002 ALLOW FILTERING;
29
```

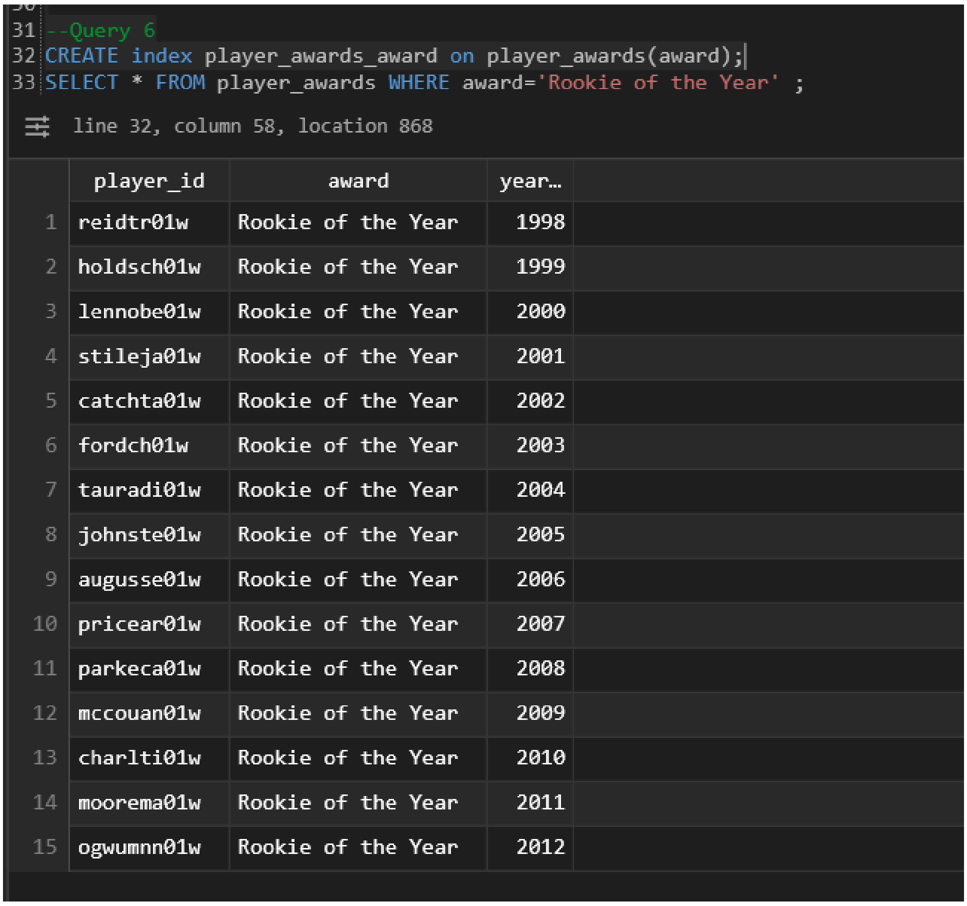
line 28, column 83, location 793

	player_id	year	team_id	blocks	games	minutes	points	steals
1	azzije01w	2002	UTA	14	32	1151	306	27
2	gardnan01w	2002	UTA	3	30	198	57	3
3	dydekma01w	2002	UTA	107	30	876	394	25
4	starbka01w	2002	UTA	4	15	88	26	9
5	randase01w	2002	UTA	1	8	135	58	4
6	ferdima01w	2002	UTA	7	32	1065	489	51
7	aguilel01w	2002	UTA	2	28	141	43	3
8	johnsla01w	2002	UTA	2	28	269	75	7
9	herriam01w	2002	UTA	14	28	269	68	9
10	caufila01w	2002	UTA	0	8	65	19	7
11	goodsad01w	2002	UTA	6	32	1101	503	45
12	willina01w	2002	UTA	16	31	1008	351	38
13	crockda01w	2002	UTA	2	18	84	29	0

Thanks to these statistics, we can easily find the 5 starting players (lines 1, 3, 6, 11 and 12) for the most of the games because they have played a lot of minutes on the basketball field!

6) In the last query we search all the players that get the Rookie award.

For this query, we needed to select from the table awards. We created an index on the award field to make the query more optimized with the WHERE on this field. This allows us to see which player win the Rookie of the Year every year.



```
31 --Query 6
32 CREATE index player_awards_award on player_awards(award);
33 SELECT * FROM player_awards WHERE award='Rookie of the Year' ;
```

line 32, column 58, location 868

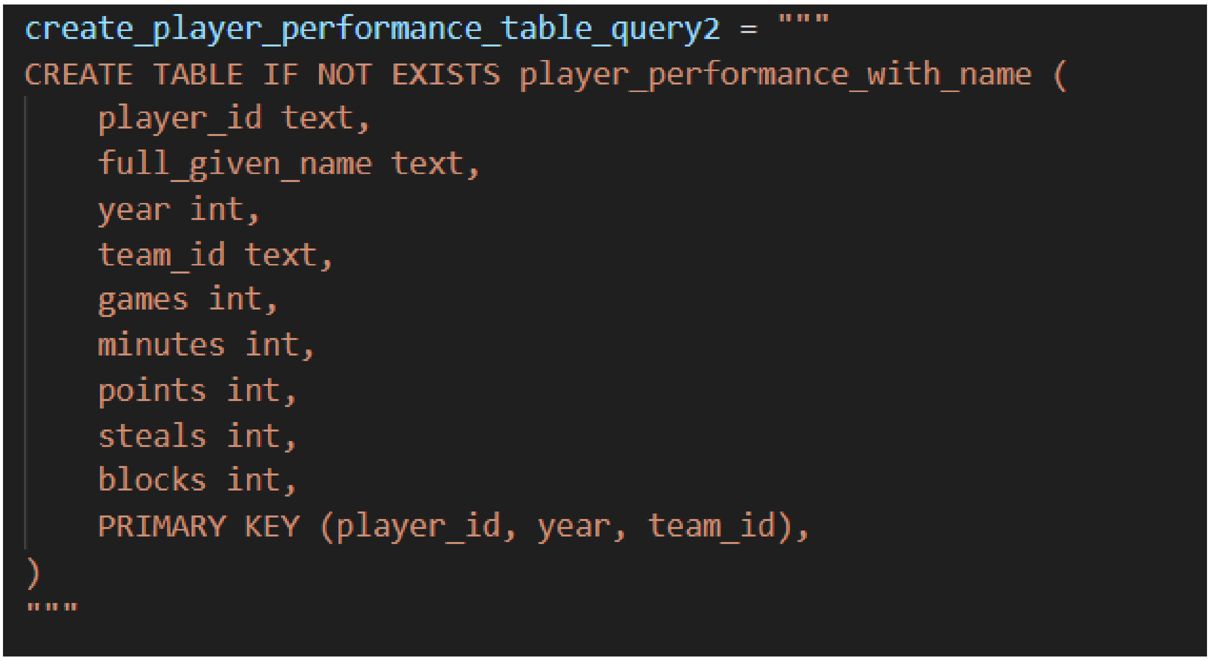
	player_id	award	year…
1	reidtr01w	Rookie of the Year	1998
2	holdsch01w	Rookie of the Year	1999
3	lennobe01w	Rookie of the Year	2000
4	stileja01w	Rookie of the Year	2001
5	catchta01w	Rookie of the Year	2002
6	fordch01w	Rookie of the Year	2003
7	tauradi01w	Rookie of the Year	2004
8	johnste01w	Rookie of the Year	2005
9	augusse01w	Rookie of the Year	2006
10	pricear01w	Rookie of the Year	2007
11	parkeca01w	Rookie of the Year	2008
12	mccouan01w	Rookie of the Year	2009
13	charlti01w	Rookie of the Year	2010
14	moorema01w	Rookie of the Year	2011
15	ogwumnn01w	Rookie of the Year	2012

Overall, we noticed that the award of the Rookie of the year was created in 1998 for the women basketball players!

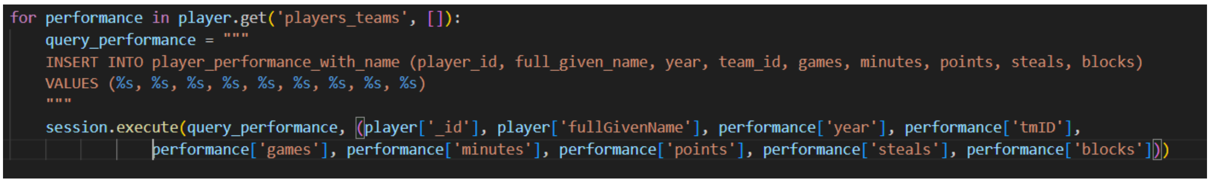
3. Complex queries (2 queries)

1) Let's count the number of participations for each player in their whole career (number of seasons).

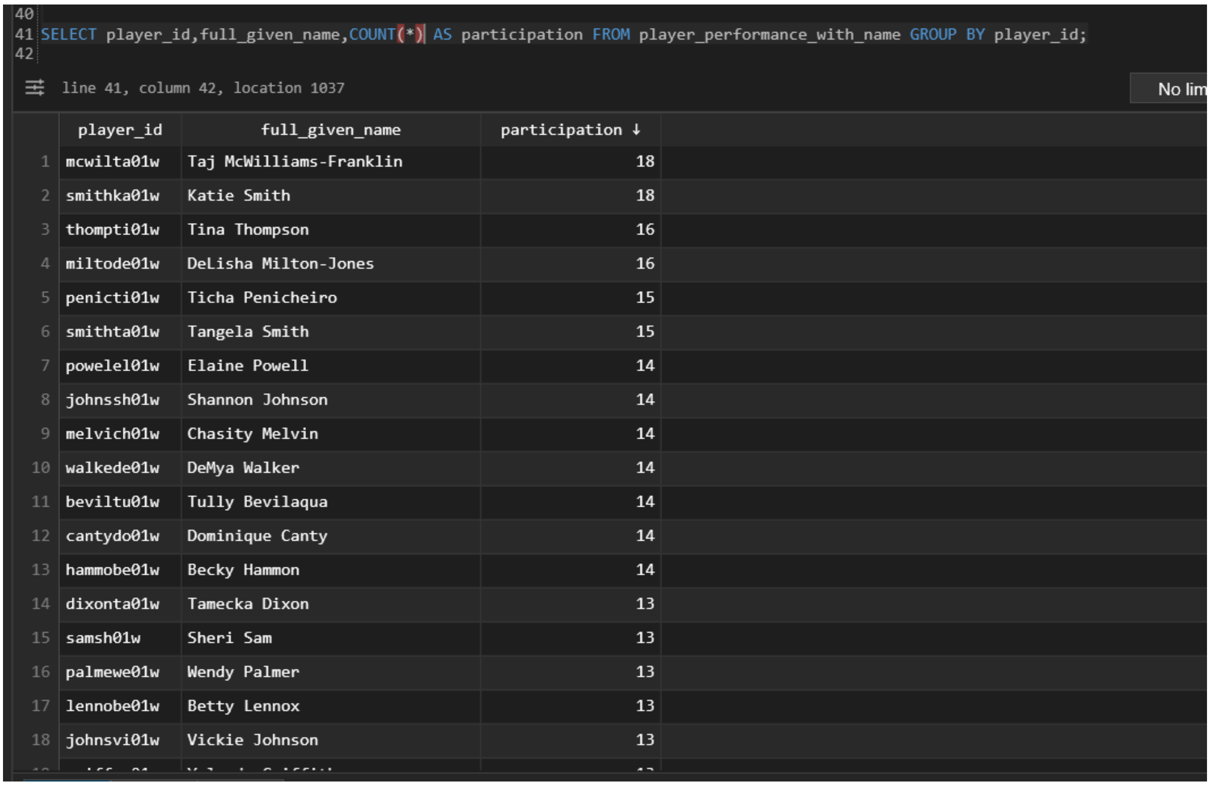
However, we would like to add the names of the players (and not only the player_id), for this, we modified the python script in order to retrieve the full given name in addition to a new table.



```
create_player_performance_table_query2 = """
CREATE TABLE IF NOT EXISTS player_performance_with_name (
    player_id text,
    full_given_name text,
    year int,
    team_id text,
    games int,
    minutes int,
    points int,
    steals int,
    blocks int,
    PRIMARY KEY (player_id, year, team_id),
)
"""
```



```
for performance in player.get('players_teams', []):
    query_performance = """
    INSERT INTO player_performance_with_name (player_id, full_given_name, year, team_id, games, minutes, points, steals, blocks)
    VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)
    """
    session.execute(query_performance, (player['_id'], player['fullGivenName'], performance['year'], performance['tmID'],
                performance['games'], performance['minutes'], performance['points'], performance['steals'], performance['blocks']))
```



```
SELECT player_id,full_given_name,COUNT(*) AS participation FROM player_performance_with_name GROUP BY player_id;
```

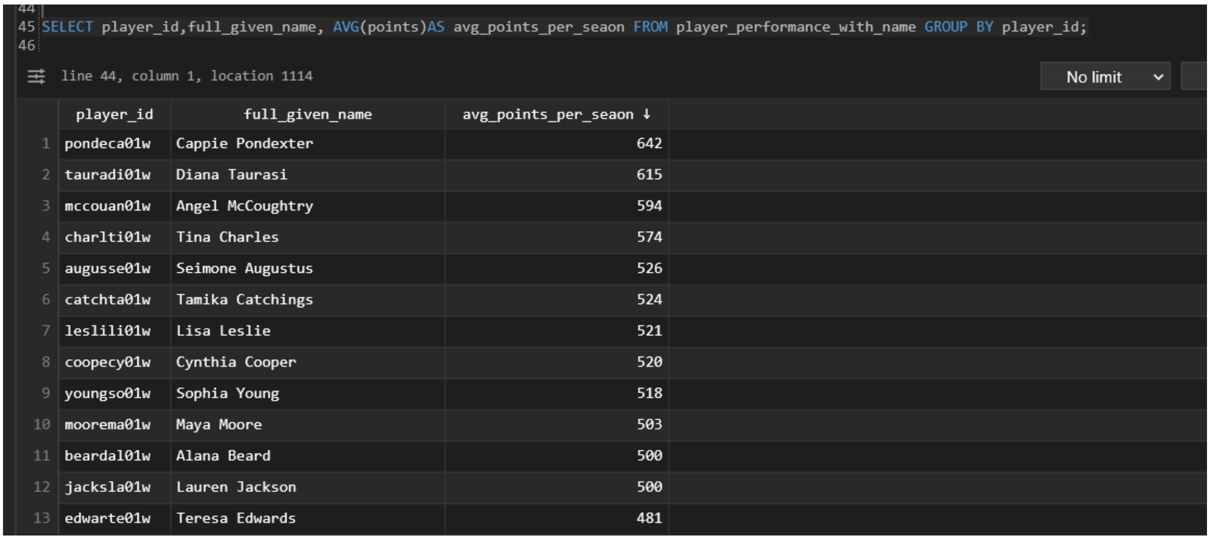
line 41, column 42, location 1037

	player_id	full_given_name	participation ↓
1	mcwilta01w	Taj McWilliams-Franklin	18
2	smithka01w	Katie Smith	18
3	thompti01w	Tina Thompson	16
4	miltode01w	DeLisha Milton-Jones	16
5	penicti01w	Ticha Penicheiro	15
6	smithta01w	Tangela Smith	15
7	powelel01w	Elaine Powell	14
8	johnssh01w	Shannon Johnson	14
9	melvich01w	Chasity Melvin	14
10	walkede01w	DeMya Walker	14
11	beviltu01w	Tully Bevilaqua	14
12	cantydo01w	Dominique Canty	14
13	hammobe01w	Becky Hammon	14
14	dixonta01w	Tamecka Dixon	13
15	samsh01w	Sheri Sam	13
16	palmewe01w	Wendy Palmer	13
17	lennobe01w	Betty Lennox	13
18	johnsvi01w	Vickie Johnson	13

Thus, Taj McWilliamsFranklin and Katie Smith are the two players who have played the most season. Attention! To play the most season doesn't mean they were a good player or they play lot of games or got many points, it's to be nuanced!

2) In the second query, we will count the average points per season for each player.

We use a group by player_id and use the function avg to select the average points per season. This allowed us to see the players that are the most consistent.



```
SELECT player_id,full_given_name, AVG(points)AS avg_points_per_seaon FROM player_performance_with_name GROUP BY player_id;
```

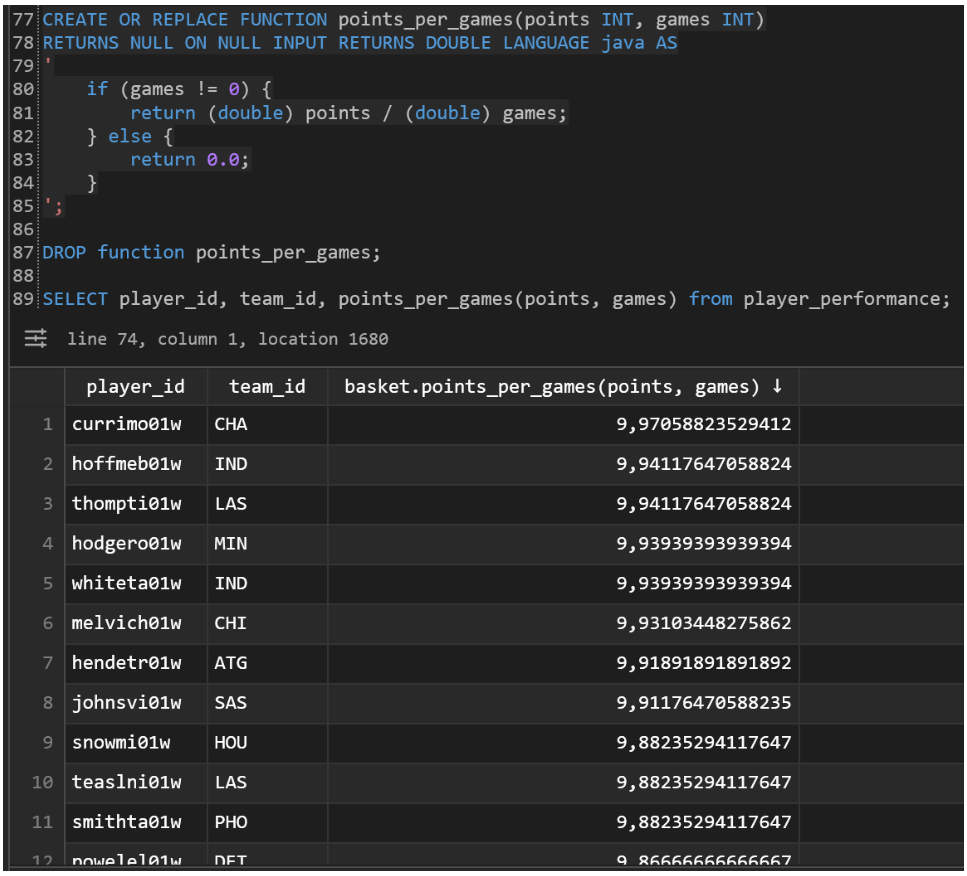
line 44, column 1, location 1114

	player_id	full_given_name	avg_points_per_seaon ↓
1	pondeca01w	Cappie Pondexter	642
2	tauradi01w	Diana Taurasi	615
3	mccouan01w	Angel McCoughtry	594
4	charlti01w	Tina Charles	574
5	augusse01w	Seimone Augustus	526
6	catchta01w	Tamika Catchings	524
7	leslili01w	Lisa Leslie	521
8	coopecy01w	Cynthia Cooper	520
9	youngso01w	Sophia Young	518
10	moorema01w	Maya Moore	503
11	beardal01w	Alana Beard	500
12	jacksla01w	Lauren Jackson	500
13	edwarte01w	Teresa Edwards	481

Indeed, compared to the previous query, there isn't any players who participate in most season here than the previous query. For instance, maybe they can just play few season to have a good average of points per season!

4. Hard query (1 query)

Let’s find the average of each players for the number of points per game :



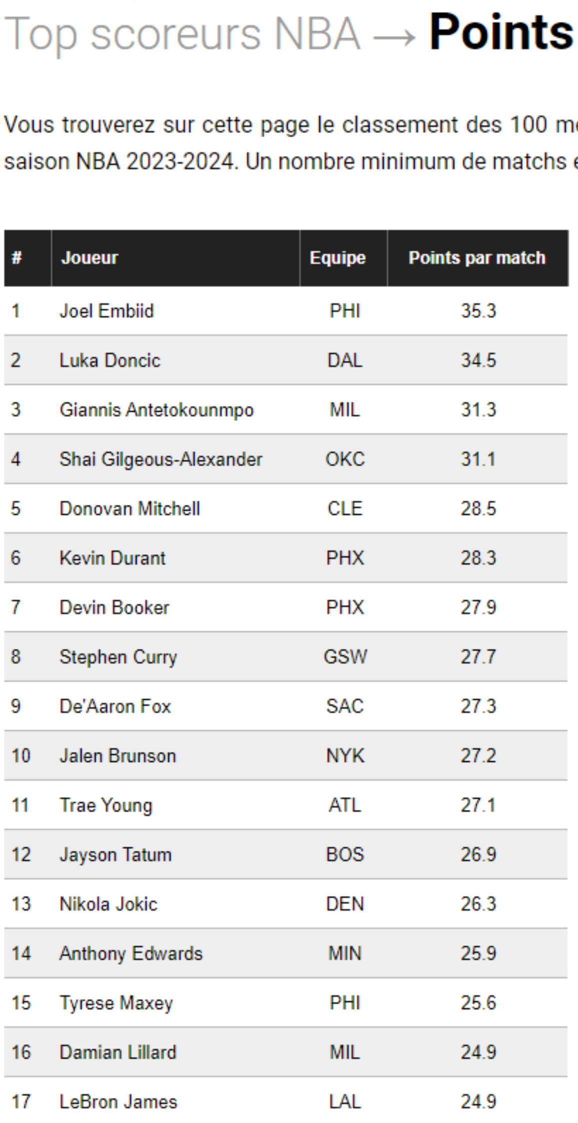
```
CREATE OR REPLACE FUNCTION points_per_games(points INT, games INT)
RETURNS NULL ON NULL INPUT RETURNS DOUBLE LANGUAGE java AS
'
    if (games != 0) {
        return (double) points / (double) games;
    } else {
        return 0.0;
    }
';

DROP function points_per_games;

SELECT player_id, team_id, points_per_games(points, games) from player_performance;
```

line 74, column 1, location 1680

	player_id	team_id	basket.points_per_games(points, games) ↓
1	currimo01w	CHA	9,97058823529412
2	hoffmeb01w	IND	9,94117647058824
3	thompti01w	LAS	9,94117647058824
4	hodgero01w	MIN	9,93939393939394
5	whiteta01w	IND	9,93939393939394
6	melvich01w	CHI	9,93103448275862
7	hendetr01w	ATG	9,91891891891892
8	johnsvi01w	SAS	9,91176470588235
9	snowmi01w	HOU	9,88235294117647
10	teaslni01w	LAS	9,88235294117647
11	smithta01w	PHO	9,88235294117647
12	powelel01w	DET	9,86666666666667

We can see that the best average of points per game is around 10 points, it’s rather low compared to the men players.

Here is the statistics of the 2013-2024 season below and we can see that the player with the best average of points per games has more than the triple than the best women player!



Top scoreurs NBA → **Points**

Vous trouverez sur cette page le classement des 100 meilleurs joueurs de NBA dans la catégorie "Points" pour la saison NBA 2023-2024. Un nombre minimum de matchs est nécessaire pour figurer dans le classement.

#	Joueur	Equipe	Points par match
1	Joel Embiid	PHI	35.3
2	Luka Doncic	DAL	34.5
3	Giannis Antetokounmpo	MIL	31.3
4	Shai Gilgeous-Alexander	OKC	31.1
5	Donovan Mitchell	CLE	28.5
6	Kevin Durant	PHX	28.3
7	Devin Booker	PHX	27.9
8	Stephen Curry	GSW	27.7
9	De'Aaron Fox	SAC	27.3
10	Jalen Brunson	NYK	27.2
11	Trae Young	ATL	27.1
12	Jayson Tatum	BOS	26.9
13	Nikola Jokic	DEN	26.3
14	Anthony Edwards	MIN	25.9
15	Tyrese Maxey	PHI	25.6
16	Damian Lillard	MIL	24.9
17	LeBron James	LAL	24.9

Link : https://www.basketusa.com/top-stats/points/