Your first priority is to get 2-3 players with high S.R who have faced at least 500 balls.And to do that you have to make a list of 10 players you want to bid in the auction so that when you try to grab them in auction you should not pay the amount greater than you have in the purse for a particular player.

create view Q1 as (select batsman, sum(total_runs) as runs, sum(ball) as balls from balls where extras_type not in ('wides') group by batsman);

Data Output

	batsman character varying	runs bigint	balls bigint
1	A Ashish Reddy	285	709
2	A Chandila	4	23
3	A Chopra	54	255
4	A Choudhary	25	71
5	A Dananjaya	4	22
6	A Flintoff	68	182
7	A Kumble	40	186
8	A Mishra	379	1462
9	A Mithun	37	94
10	A Mukund	19	87
11	A Nehra	48	264
12	A Nortje	7	29
13	A Singh	3	42
14	A Symonds	993	2710
15	A Uniyal	4	21
16	A Zampa	5	34

ssages Notifications Query History Query

```
create table Q1 as (select batsman, sum(total_runs)
as runs, sum(ball) as   balls  from balls
where extras_type not in ('wides') group by batsman);
```

Activate Windows
Go to Settings to activate Windows.

Total rows: 537 of 537 Query complete 00:00:00.278 Ln 1, Col 1

select batsman, runs as total_runs, balls as total_balls,(runs*100)/balls as SR from Q1 where balls > 500 order by SR desc, total_balls desc limit 10 ;

Data Output

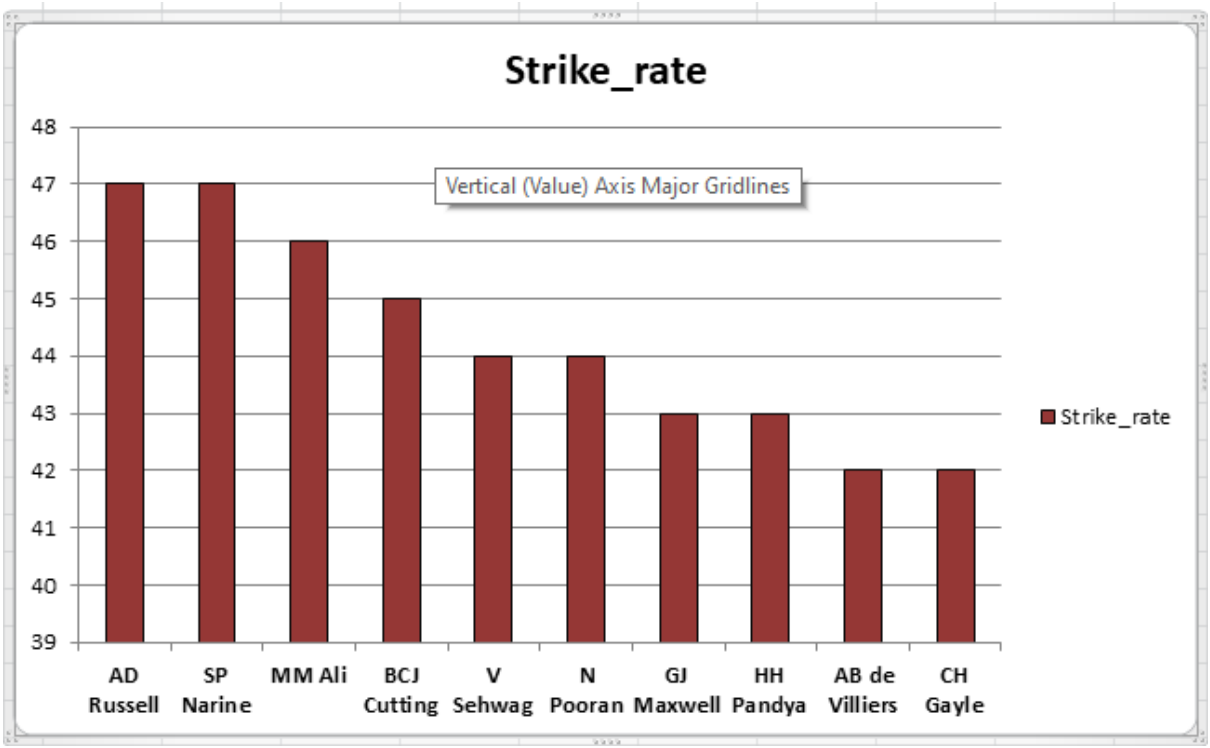
	batsman character varying	total_runs bigint	total_balls bigint	sr bigint
1	AD Russell	1544	3231	47
2	SP Narine	921	1948	47
3	MM Ali	320	694	46
4	BCJ Cutting	242	532	45
5	V Sehwag	2819	6386	44
6	N Pooran	531	1197	44
7	GJ Maxwell	1554	3575	43
8	HH Pandya	1378	3143	43
9	AB de Villiers	4930	11569	42
10	CH Gayle	4912	11501	42

Messages Notifications Query History Query

```
1  create table Q1 as (select batsman, sum(total_runs)
2  as runs, sum(ball) as   balls  from balls
3  where extras_type not in ('wides') group by batsman);
4
5
6  select batsman, runs as total_runs,
7  balls as total_balls,(runs*100)/balls as SR
8  from Q1 where balls > 500
9  order by SR desc, total_balls desc limit 10 ;
10
11
12
13
```

Activate Windows
Go to Settings to activate Windows.

Total rows: 10 of 10 Query complete 00:00:00.066 Ln 6, Col 1


Strike_rate
48
47
46
45
44
43
42
41
40
39
Vertical (Value) Axis Major Gridlines
AD Russell
SP Narine
MM Ali
BCJ Cutting
V Sehwag
N Pooran
GJ Maxwell
HH Pandya
AB de Villiers
CH Gayle
Strike_rate

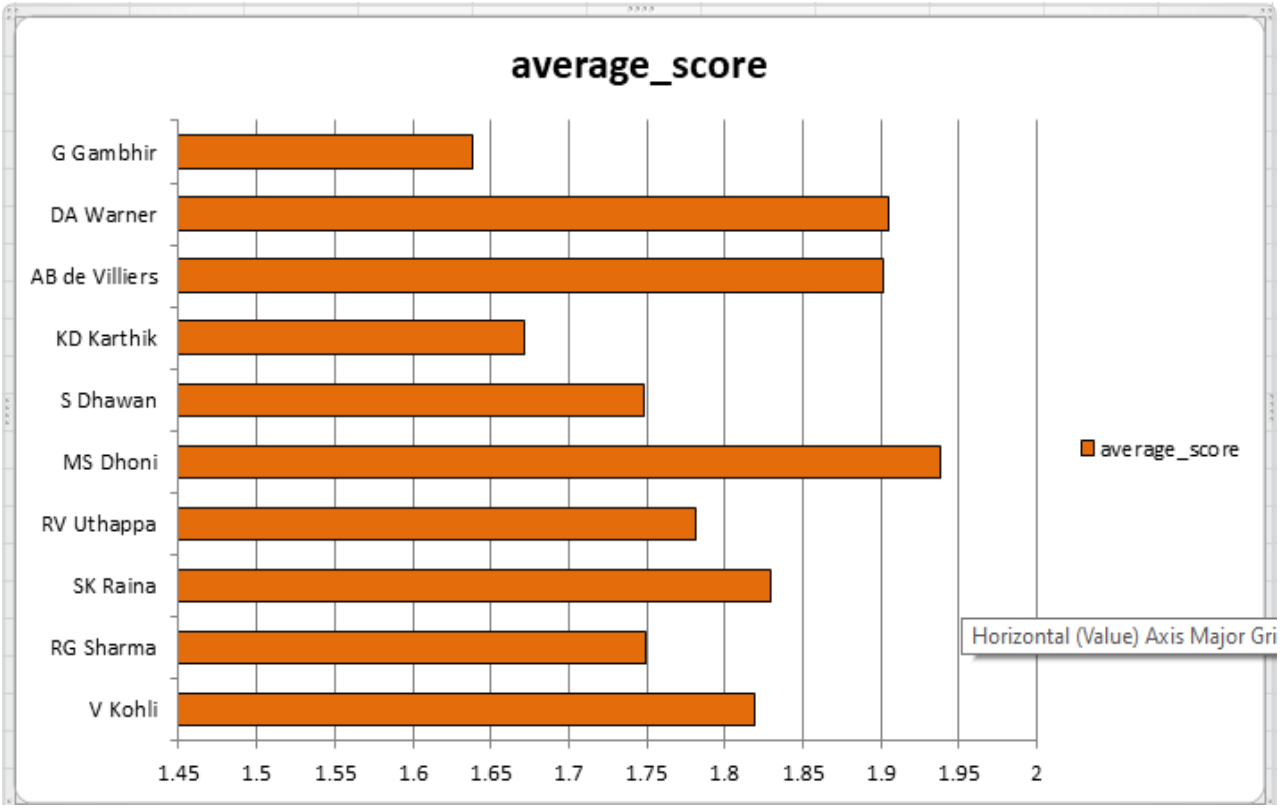
Now you need to get 2-3 players with good Average who have played more the 2 ipl seasons.And to do that you have to make a list of 10 players you want to bid in the auction so that when you try to grab them in auction you should not pay the amount greater than you have in the purse for a particular player.

create view Q2 as (select distinct b.id, b.batsman, extract(year from m.date), batsman_runs, dismissal_kind from balls b left join matches m on b.id = m.id order by b.batsman);

select batsman,count(distinct extract) as season, sum(batsman_runs) as total_runs ,count(dismissal_kind) as total_dismissal, round(sum(batsman_runs)/cast(count(dismissal_kind) as decimal),3) as average_score from q2 group by batsman having count(distinct extract)>2 order by total_dismissal desc limit 10;

Data Output

	batsman character varying	season bigint	total_runs bigint	total_dismissal bigint	average_score numeric
1	V Kohli	13	1608	884	1.819
2	RG Sharma	13	1537	879	1.749
3	SK Raina	12	1604	877	1.829
4	RV Uthappa	13	1516	851	1.781
5	MS Dhoni	13	1647	850	1.938
6	S Dhawan	13	1423	814	1.748
7	KD Karthik	13	1258	753	1.671
8	AB de Villiers	13	1341	705	1.902
9	DA Warner	11	1343	705	1.905
10	G Gambhir	11	1125	687	1.638


average_score
G Gambhir
DA Warner
AB de Villiers
KD Karthik
S Dhawan
MS Dhoni
RV Uthappa
SK Raina
RG Sharma
V Kohli
1.45
1.5
1.55
1.6
1.65
1.7
1.75
1.8
1.85
1.9
1.95
2
average_score
Horizontal (Value) Axis Major Gri

Now you need to get 2-3 Hard-hitting players who have scored most runs in boundaries and have played more the 2 ipl season. To do that you have to make a list of 10 players you want to bid in the auction so that when you try to grab them in auction you should not pay the amount greater than you have in the purse for a particular player.

create view Q3 as (select distinct d.id, d.batsman, extract(year from m.date), d.batsman_runs from deliveries_v02 d left join matches m on d.id = m.id where ball_result='Boundary' order by d.batsman);

Data Output

	id integer	batsman character varying	extract numeric	batsman_runs integer
1	548346	A Ashish Reddy	2012	6
2	548359	A Ashish Reddy	2012	0
3	548359	A Ashish Reddy	2012	4
4	548373	A Ashish Reddy	2012	4
5	598000	A Ashish Reddy	2013	4
6	598004	A Ashish Reddy	2013	6
7	598010	A Ashish Reddy	2013	4
8	598018	A Ashish Reddy	2013	6
9	598021	A Ashish Reddy	2013	4
10	598030	A Ashish Reddy	2013	4
11	598030	A Ashish Reddy	2013	6
12	598032	A Ashish Reddy	2013	4
13	598044	A Ashish Reddy	2013	4
14	829727	A Ashish Reddy	2015	6
15	829731	A Ashish Reddy	2015	4
16	829731	A Ashish Reddy	2015	6

ages Notifications Query History Query

```
select* from q3;
select* from  q3_2
select * from deliveries_v02

create table Q3 as (select distinct d.id, d.batsman,
extract(year from m.date), d.batsman_runs
from deliveries_v02 d
left join matches m on d.id = m.id
where ball_result='Boundary'
order by d.batsman)
```

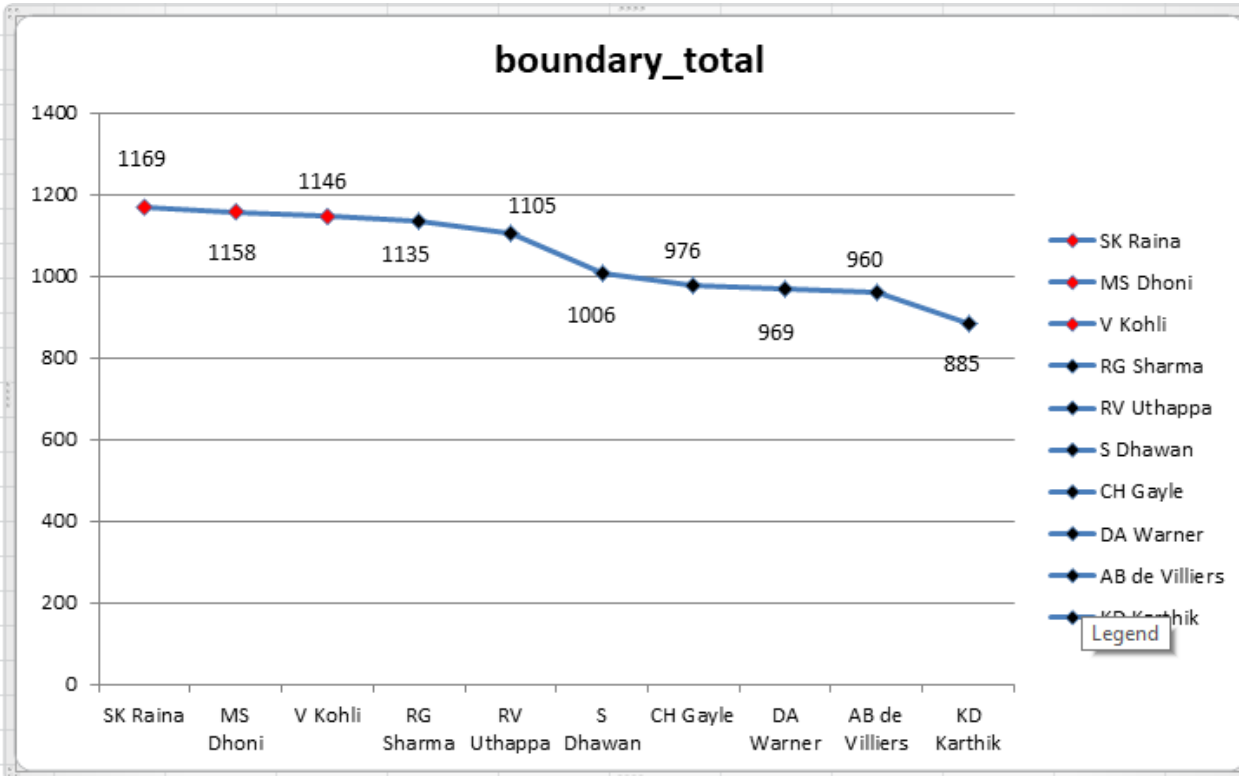

SELECT batsman, count(distinct extract) as seasons, sum(batsman_runs) as boundary_tot from Q3 group by batsman having count(distinct extract)>2 order by boundary_tot desc limit 10;

Data Output

	batsman character varying	seasons bigint	boundary_tot bigint
1	SK Raina	12	1169
2	MS Dhoni	13	1158
3	V Kohli	13	1146
4	RG Sharma	13	1135
5	RV Uthappa	13	1105
6	S Dhawan	13	1006
7	CH Gayle	12	976
8	DA Warner	11	969
9	AB de Villiers	13	960
10	KD Karthik	13	885

Messages Notifications Query History Query

```
select* from q3;
select* from  q3_2
select * from deliveries_v02

create table Q3 as (select distinct d.id, d.batsman,
extract(year from m.date), d.batsman_runs
from deliveries_v02 d
left join matches m on d.id = m.id
where ball_result='Boundary'
order by d.batsman)

SELECT batsman,
count(distinct extract) as seasons,
sum(batsman_runs) as boundary_tot from Q3
group by batsman
having count(distinct extract)>2
order by boundary_tot desc limit 10;
```


boundary_total
1400
1200
1000
800
600
400
200
0
1169
1158
1146
1135
1105
1006
976
969
960
885
SK Raina
MS Dhoni
V Kohli
RG Sharma
RV Uthappa
S Dhawan
CH Gayle
DA Warner
AB de Villiers
KD Karthik
Legend

Your first priority is to get 2-3 bowlers with good economy who have bowled at least 500 balls in IPL so far.To do that you have to make a list of 10 players you want to bid in the auction so that when you try to grab them in auction you should not pay the amount greater than you have in the purse for a particular player.

create view Q4 as (select bowler, count(ball) as ball_tot, sum(total_runs) as conceded_runs from balls group by bowler having count(ball)>500);

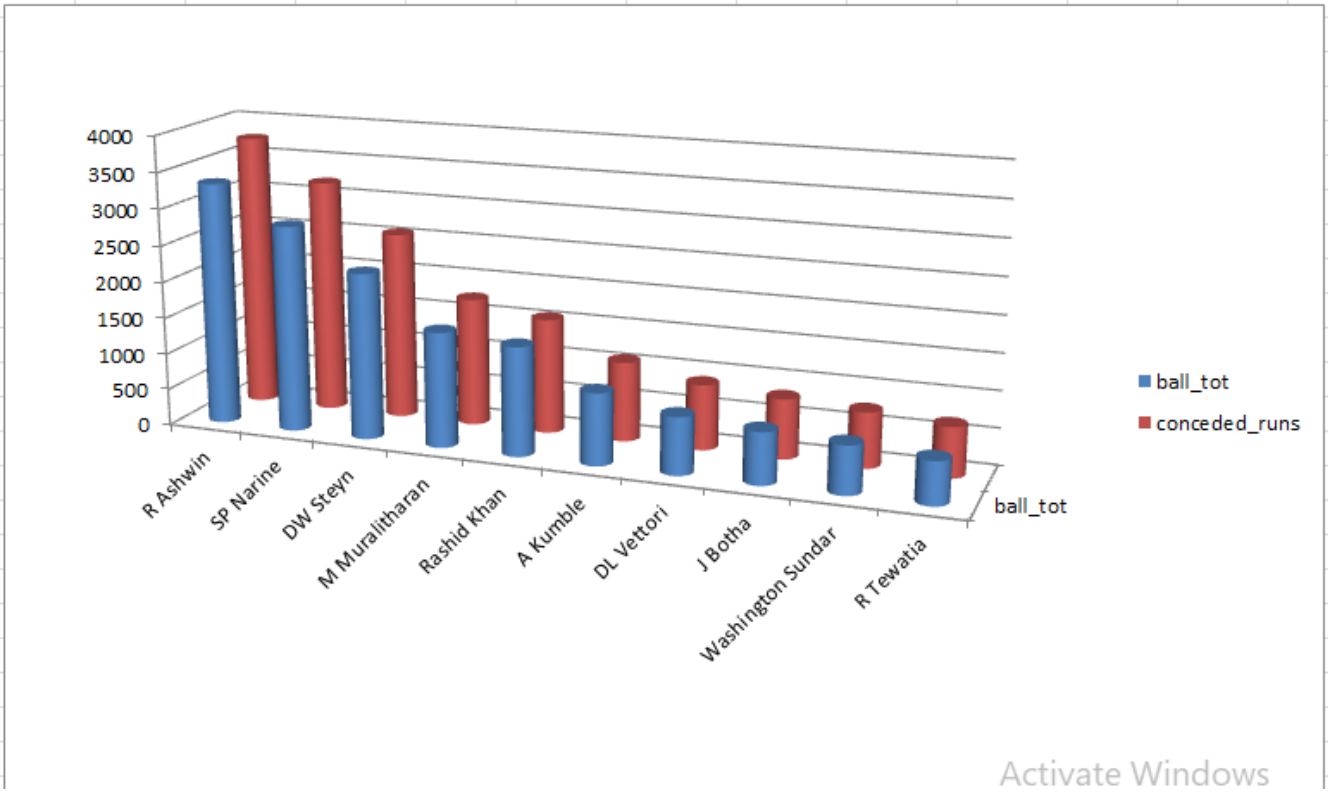
select bowler, ball_tot, conceded_runs, (conceded_runs*6)/ball_tot as economy_Rate from q4 where ball_tot >500 order by economy_rate, ball_tot desc limit 10;

Data Output

	bowler character varying	ball_tot bigint	conceded_runs bigint	economy_rate bigint
1	R Ashwin	3327	3756	6
2	SP Narine	2824	3208	6
3	DW Steyn	2276	2568	6
4	M Muralitharan	1577	1755	6
5	Rashid Khan	1490	1573	6
6	A Kumble	983	1089	6
7	DL Vettori	785	894	6
8	J Botha	709	818	6
9	Washington Sundar	660	758	6
10	R Tewatia	587	684	6

Messages Notifications Query History Query

```
select*from balls

create view Q4 as (select bowler, count(ball)
as ball_tot, sum(total_runs) as conceded_runs
from balls group by bowler
having count(ball)>500);
select* from q4

select bowler, ball_tot, conceded_runs,
(conceded_runs*6)/ball_tot as economy_Rate
from q4 where ball_tot >500
order by economy_rate, ball_tot desc  limit 10;
```



Now you need to get 2-3 bowlers with the best strike rate and who have bowled at least 500 balls in IPL so far.To do that you have to make a list of 10 players you want to bid in the auction so that when you try to grab them in auction you should not pay the amount greater than you have in the purse for a particular player.

select bowler, count(ball) as balls_bowled,sum(batsman_runs) as runs, ROUND(sum(batsman_runs)/cast(count(ball)as decimal),3) as strike_rate from deliveries_v02 WHERE extras_type NOT IN ('wides') group by bowler having count(ball) > 500 order by strike_rate desc limit 10

Data Output

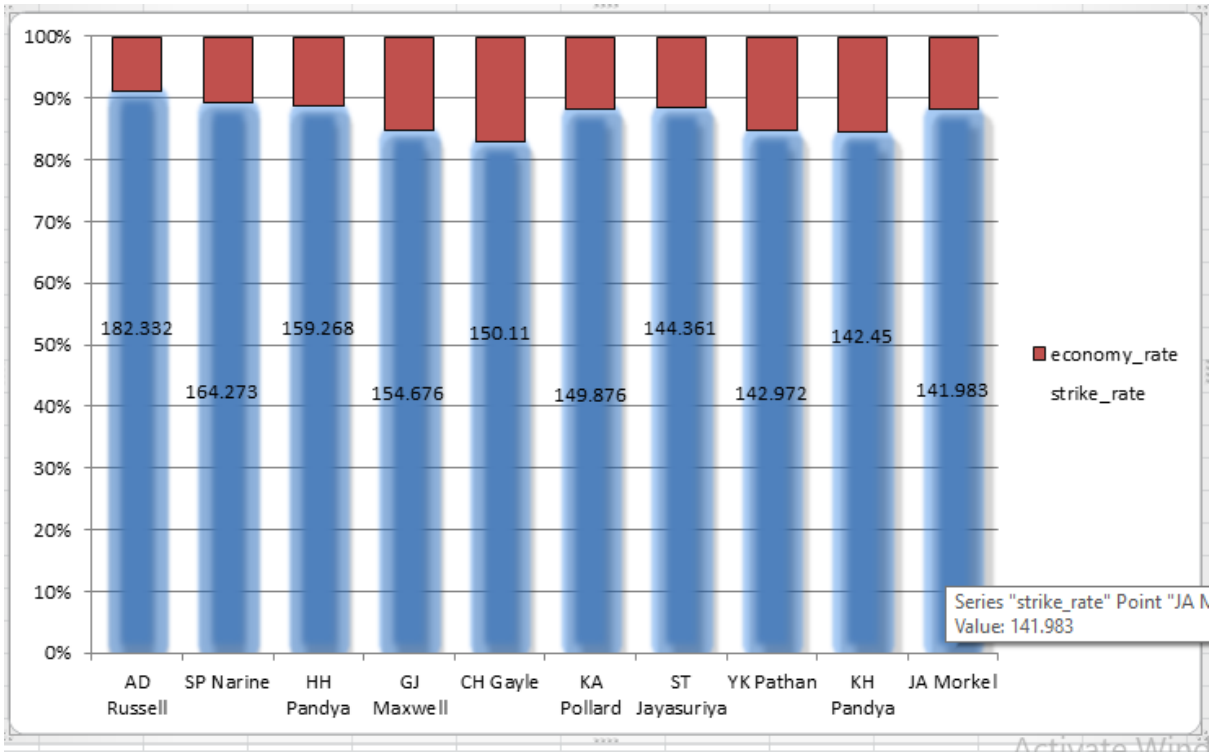
	bowler character varying	balls_bowled bigint	runs bigint	strike_rate numeric
1	MP Stoinis	545	833	1.528
2	M Prasidh Krishna	521	775	1.488
3	HH Pandya	872	1268	1.454
4	DR Smith	543	787	1.449
5	AD Russell	1153	1667	1.446
6	JD Unadkat	1645	2377	1.445
7	Mohammed Siraj	719	1036	1.441
8	AS Rajpoot	539	772	1.432
9	SN Thakur	903	1290	1.429
10	NLTC Perera	701	998	1.424

Now you need to get 2-3 All_rounders with the best batting as well as bowling strike rate and who have faced at least 500 balls in IPL so far and have bowled minimum 300 balls.To do that you have to make a list of 10 players you want to bid in the auction so that when you try to grab them in auction you should not pay the amount greater than you have in the purse for a particular player.

```
SELECT a.batsman AS player_name,
ROUND(SUM(a.batsman_runs)/CAST(COUNT(a.ball) AS DECIMAL)*100,3) AS
strike_Rate,  economy_rate FROM deliveries_v02 AS a

INNER JOIN

(SELECT bowler, COUNT(ball) AS balls_bowled, ROUND(CAST(COUNT(ball) AS
DECIMAL)/SUM(is_wicket),3) AS economy_rate FROM deliveries_v02 GROUP BY
bowler HAVING COUNT(ball) >= 300 ORDER BY economy_rate) AS b

ON a.batsman = b.bowler WHERE extras_type NOT IN ('wides') GROUP BY batsman,
economy_rate HAVING COUNT(ball) >= 500 ORDER BY strike_Rate DESC LIMIT 10;
```

Data Output

	player_name character varying	strike_rate numeric	economy_rate numeric
1	AD Russell	182.332	17.701
2	SP Narine	164.273	19.748
3	HH Pandya	159.268	20.311
4	GJ Maxwell	154.676	27.900
5	CH Gayle	150.110	30.737
6	KA Pollard	149.876	19.915
7	ST Jayasuriya	144.361	18.813
8	YK Pathan	142.972	25.739
9	KH Pandya	142.450	26.184
10	JA Morkel	141.983	18.823



Wicket keeper criteria

Keeping the Best wicket keepers in mind, I think a wicket keeper should be a hard hitter as well as average economy bowler.

Wicket keeper is not just an ordinary batsman or bowler, he is the most sharp and agile player of the team, the one who binds the team in need of moment drives the team to the peak.

A hard hitter would be a good criteria as he when the team other batsman are set on the pitch he can go help to jump the scores really well , boost the upcoming batsman as well as keep the momentum of the team up.

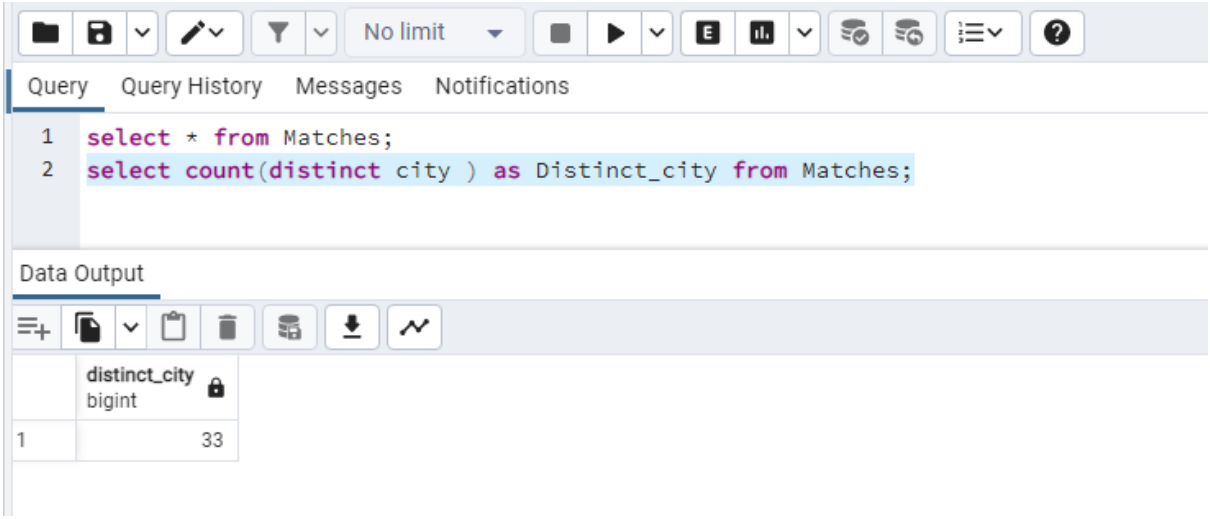
Same goes with a average economy bowler, when the team need a bit of change in plan or there is a need to surprise the enemy batsman but can't afford to lose many runs a good economy bowler can help to keep the pressure up and let's the team settle for the rest of game.

```
CREATE TABLE Balls(
  id                INTEGER  NOT NULL
 ,inning            int NOT NULL
 ,over              int  NOT NULL
 ,ball              int NOT NULL
 ,batsman           VARCHAR NOT NULL
 ,non_striker       VARCHAR NOT NULL
 ,bowler            VARCHAR NOT NULL
 ,batsman_runs      int NOT NULL
 ,extra_runs        int NOT NULL
 ,total_runs        int NOT NULL
 ,is_wicket         int NOT NULL
 ,dismissal_kind    VARCHAR NOT NULL
 ,player_dismissed  VARCHAR NOT NULL
 ,fielder           VARCHAR NOT NULL
 ,extras_type       VARCHAR NOT NULL
 ,batting_team      VARCHAR NOT NULL
 ,bowling_team      VARCHAR NOT NULL
 ,CONSTRAINT f_key FOREIGN KEY (id) REFERENCES Matches(id)
);

copy Balls from 'C:\Program Files\PostgreSQL\15\data\IPL_Ball.csv' delimiter ',' csv
header;


CREATE TABLE Matches(
  id              integer  NOT NULL PRIMARY KEY
 ,city            varchar NOT NULL
 ,date            date  NOT NULL
 ,player_of_match varchar NOT NULL
 ,venue           varchar NOT NULL
 ,neutral_venue   bit  NOT NULL
 ,team1           varchar NOT NULL
 ,team2           varchar NOT NULL
 ,toss_winner     varchar NOT NULL
 ,toss_decision   varchar NOT NULL
 ,winner          varchar NOT NULL
 ,result          varchar NOT NULL
 ,result_margin   integer
 ,eliminator      varchar NOT NULL
 ,method          varchar NOT NULL
 ,umpire1         varchar NOT NULL
 ,umpire2         varchar NOT NULL
);

copy Matches from 'C:\Program Files\PostgreSQL\15\data\IPL_matches.csv' delimiter
',' csv header
```

Additional query

1. *Get the count of cities that have hosted an IPL match*

select count(distinct city) from Matches;



2. ***Create table deliveries_v02 with all the columns of the table 'deliveries' and an additional column ball_result containing values boundary, dot or other depending on the total_run (boundary for >= 4, dot for 0 and other for any other number)***
(Hint 1 : CASE WHEN statement is used to get condition based results)
(Hint 2: To convert the output data of the select statement into a table, you can use a subquery. Create table table_name as [entire select statement].

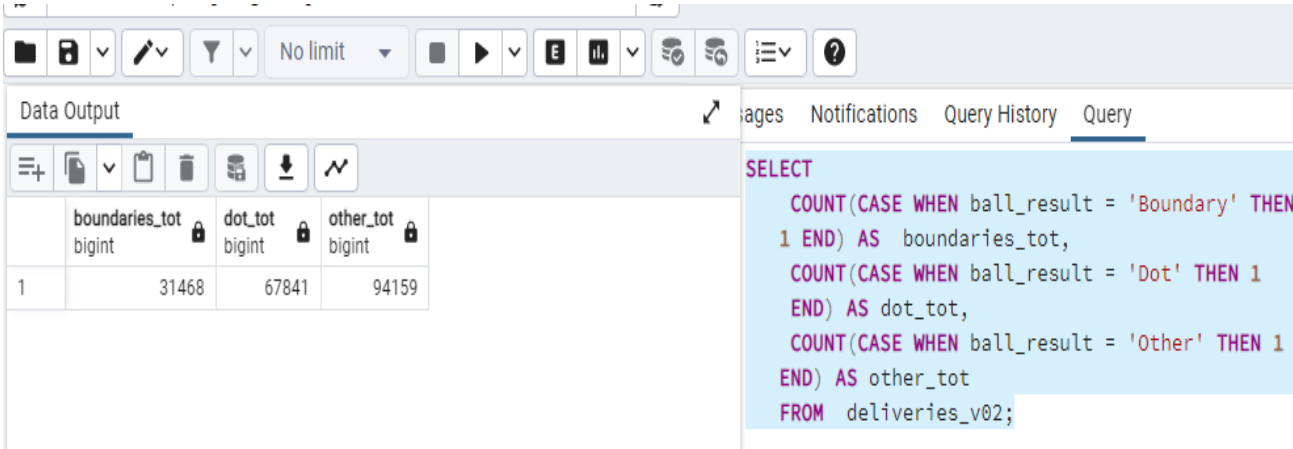
create table deliveries_v02 as (select *, case when total_runs >=4 then 'Boundary' when total_runs = 0 then 'Dot' else 'Other' end as ball_result from Balls);

	icket ger	dismissal_kind character varying	player_dismissed character varying	fielder character varying	extras_type character varying	batting_team character varying	bowling_team character varying	ball_result text
1	0	NA	NA	NA	NA	Kolkata Knight Riders	Royal Challengers Bangalore	Other
2	0	NA	NA	NA	NA	Kolkata Knight Riders	Royal Challengers Bangalore	Other
3	0	NA	NA	NA	NA	Kolkata Knight Riders	Royal Challengers Bangalore	Dot
4	0	NA	NA	NA	NA	Kolkata Knight Riders	Royal Challengers Bangalore	Other
5	0	NA	NA	NA	NA	Kolkata Knight Riders	Royal Challengers Bangalore	Other
6	0	NA	NA	NA	NA	Kolkata Knight Riders	Royal Challengers Bangalore	Other
7	0	NA	NA	NA	NA	Kolkata Knight Riders	Royal Challengers Bangalore	Other
8	0	NA	NA	NA	NA	Kolkata Knight Riders	Royal Challengers Bangalore	Other
9	0	NA	NA	NA	NA	Kolkata Knight Riders	Royal Challengers Bangalore	Dot
10	0	NA	NA	NA	NA	Kolkata Knight Riders	Royal Challengers Bangalore	Dot
11	0	NA	NA	NA	NA	Kolkata Knight Riders	Royal Challengers Bangalore	Dot
12	0	NA	NA	NA	NA	Kolkata Knight Riders	Royal Challengers Bangalore	Other
13	0	NA	NA	NA	NA	Kolkata Knight Riders	Royal Challengers Bangalore	Other
14	0	NA	NA	NA	NA	Kolkata Knight Riders	Royal Challengers Bangalore	Other
15	0	NA	NA	NA	NA	Kolkata Knight Riders	Royal Challengers Bangalore	Other
16	0	NA	NA	NA	NA	Kolkata Knight Riders	Royal Challengers Bangalore	Other
17	0	NA	NA	NA	NA	Kolkata Knight Riders	Royal Challengers Bangalore	Other

Total rows: 1000 of 193468 | Query complete 00:00:01.084 | Ln 4, Col 1

3. ***Write a query to fetch the total number of boundaries and dot balls from the deliveries_v02 table.***

SELECT COUNT(CASE WHEN ball_result = 'Boundary' THEN 1 END) AS boundaries_tot, COUNT(CASE WHEN ball_result = 'Dot' THEN END) AS dot_tot, COUNT(CASE WHEN ball_result = 'Other' THEN 1 END) AS other_to FROM deliveries_v02;



4. ***Write a query to fetch the total number of boundaries scored by each team from the deliveries_v02 table and order it in descending order of the number of boundaries scored.***

select distinct batting_team, count(case when ball_result = 'Boundary' then 1 end) as Boundary_scored from deliveries_v02 group by batting_team order by Boundary_scored Desc;

Data Output

	batting_team character varying	boundary_scored bigint
1	Mumbai Indians	4118
2	Royal Challengers Bangalore	3800
3	Kings XI Punjab	3780
4	Kolkata Knight Riders	3739
5	Chennai Super Kings	3496
6	Rajasthan Royals	3041
7	Delhi Daredevils	3022
8	Sunrisers Hyderabad	2306
9	Deccan Chargers	1387
10	Pune Warriors	733
11	Delhi Capitals	659
12	Gujarat Lions	624
13	Rising Pune Supergiant	290
14	Rising Pune Supergiants	242
15	Kochi Tuskers Kerala	231

Messages Notifications Query History Query

```
select distinct batting_team,
count(case when ball_result = 'Boundary' then 1 end)
as Boundary_scored
from deliveries_v02
group by batting_team order by Boundary_scored
Desc;
```

5. ***Write a query to fetch the total number of dot balls bowled by each team and order it in descending order of the total number of dot balls bowled.***

select distinct bowling_team, count(case when ball_result = 'Dot' then 1 end) as Dot_scored from deliveries_v02 group by bowling_team order by Dot_scored Desc;

Data Output

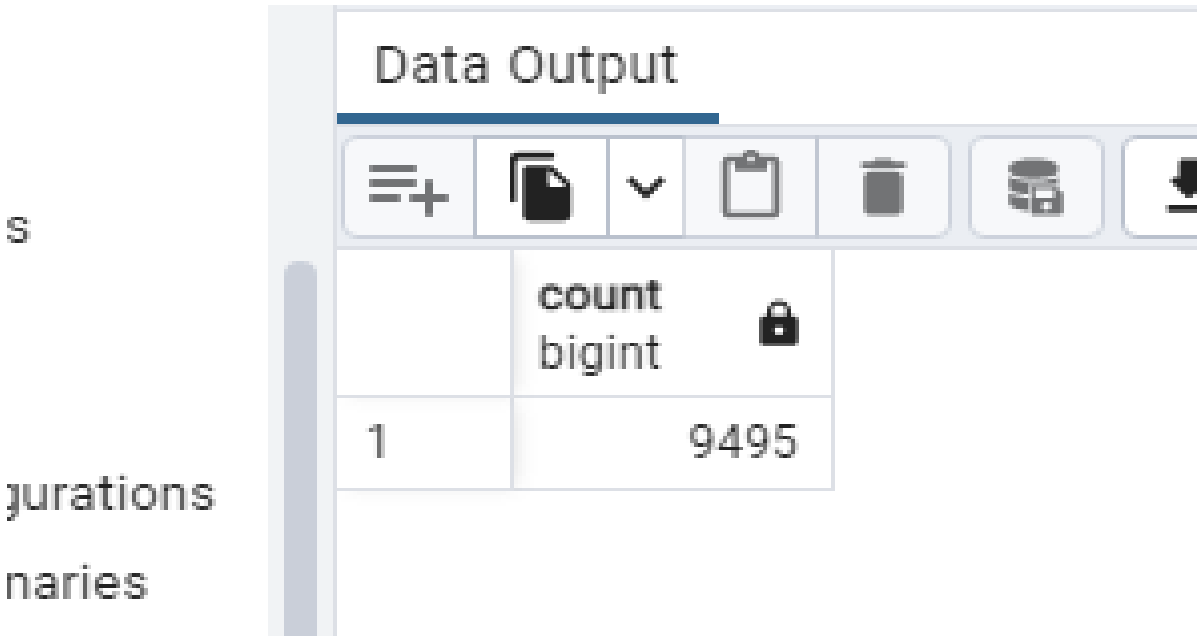
	bowling_team character varying	dot_scored bigint
1	Mumbai Indians	8714
2	Royal Challengers Bangalore	7955
3	Kolkata Knight Riders	7894
4	Kings XI Punjab	7679
5	Chennai Super Kings	7593
6	Rajasthan Royals	6665
7	Delhi Daredevils	6520
8	Sunrisers Hyderabad	5248
9	Deccan Chargers	3306
10	Pune Warriors	1900
11	Delhi Capitals	1338
12	Gujarat Lions	1095
13	Rising Pune Supergiant	698
14	Kochi Tuskers Kerala	626
15	Rising Pune Supergiants	539

Messages Notifications Query History Query

```
1 select distinct bowling_team,
2 count(case when ball_result = 'Dot' then 1 end)
3 as Dot_scored
4 from deliveries_v02 where bowling_team != 'NA'
5 group by bowling_team order by Dot_scored Desc;
6
7
8
```

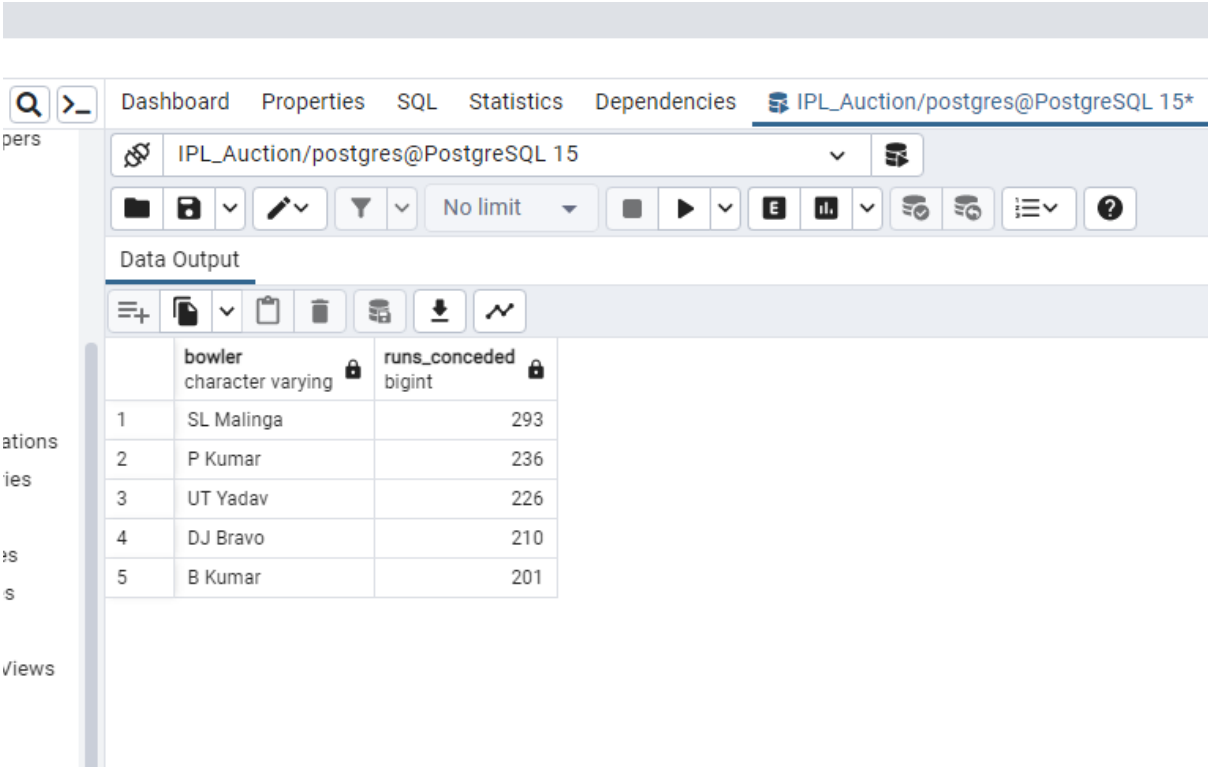
6. ***Write a query to fetch the total number of dismissals by dismissal kinds where dismissal kind is not NA***

select count(case when dismissal_kind != 'NA' then 1 end) as Count from deliveries_v02;



7. ***Write a query to get the top 5 bowlers who conceded maximum extra runs from the deliveries table***

select bowler, sum(extra_runs) as runs from deliveries_v02 group by bowler order by runs desc limit 5;



	bowler character varying	runs_conceded bigint
1	SL Malinga	293
2	P Kumar	236
3	UT Yadav	226
4	DJ Bravo	210
5	B Kumar	201

8. ***Write a query to create a table named deliveries_v03 with all the columns of deliveries_v02 table and two additional column (named venue and match_date) of venue and date from table matches***

CREATE TABLE deliveries_v03 AS SELECT d.*, m.venue as venue, m.date AS date FROM deliveries_v02 d JOIN Matches m ON d.id = m.id;

9. ***Write a query to fetch the total runs scored for each venue and order it in the descending order of total runs scored.***

select venue, sum(total_runs)as runs_tot from deliveries_v03 group by venue order by runs_tot desc;

	venue character varying	runs_tot bigint
1	Eden Gardens	23658
2	Wankhede Stadium	23390
3	Feroz Shah Kotla	22947
4	M Chinnaswamy Stadium	20237
5	Rajiv Gandhi International Stadium, Uppal	19484
6	MA Chidambaram Stadium, Chepauk	17821
7	Sawai Mansingh Stadium	14264
8	Punjab Cricket Association Stadium, Mohali	10987
9	Dubai International Cricket Stadium	10402
10	Sheikh Zayed Stadium	8830
11	Punjab Cricket Association IS Bindra Stadium, Moh...	7021
12	Maharashtra Cricket Association Stadium	6780
13	Sharjah Cricket Stadium	5924
14	M.Chinnaswamy Stadium	5127
15	Dr DY Patil Sports Academy	4810
16	Subrata Roy Sahara Stadium	4755
17	Kingsmead	4353

Total rows: 36 of 36 Query complete 00:00:00.196

10. *Write a query to fetch the year-wise total runs scored at Eden Gardens and order it in the descending order of total runs scored.*

select extract(year from date) as year, sum(total_runs) as runs_tot from deliveries_v03 where venue = 'Eden Gardens' group by year order by runs_tot desc;

Data Output

	year numeric	runs_tot bigint
1	2018	2885
2	2019	2651
3	2015	2386
4	2013	2304
5	2017	2194
6	2010	2167
7	2016	2073
8	2012	2012
9	2011	1854
10	2008	1843
11	2014	1289