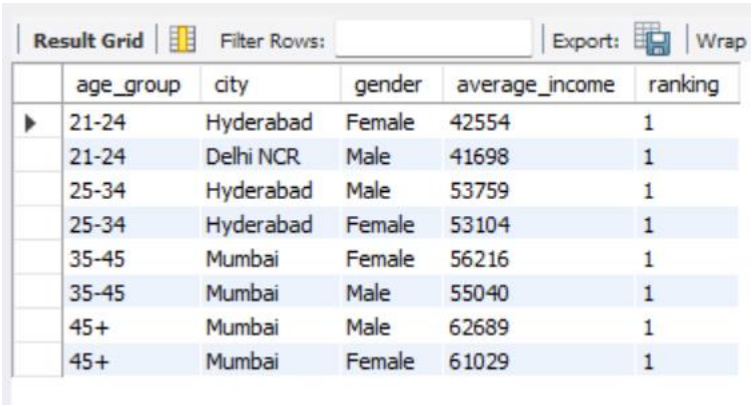
1. Finding the city which has highest average income according to gender across different age groups :

SQL Query :

```
with cte as(

select age_group, city, gender, round(avg(avg_income),0) as average_income,

dense_rank() over (partition by age_group order by round(avg(avg_income),0) desc) as ranking from customers

group by  age_group, city, gender

order by age_group,average_income desc)

select * from cte where ranking =1;
```

Result :



Result Grid | Filter Rows: | Export: | Wrap

age_group	city	gender	average_income	ranking
21-24	Hyderabad	Female	42554	1
21-24	Delhi NCR	Male	41698	1
25-34	Hyderabad	Male	53759	1
25-34	Hyderabad	Female	53104	1
35-45	Mumbai	Female	56216	1
35-45	Mumbai	Male	55040	1
45+	Mumbai	Male	62689	1
45+	Mumbai	Female	61029	1

2. Determining the top occupation according to each city by count:

SQL Query :

```
with cte as(

select city,occupation, count(occupation), dense_rank() over (partition by city order by count(occupation) desc) as ranking

from customers

group by city,occupation

order by city,ranking)

select * from cte where ranking=1;
```

Result:

esult Grid | Filter Rows: | Export: | Wrap Cell

city	occupation	count(occupation)	ranking
Bengaluru	Salaried IT Employees	226	1
Chennai	Salaried IT Employees	257	1
Delhi NCR	Salaried IT Employees	259	1
Hyderabad	Salaried IT Employees	188	1
Mumbai	Salaried IT Employees	364	1

3. Determining average income in each city based on marital_status:

SQL QUERY :

```
select city, marital_status, round(avg(avg_income),0) as average_income from customers

group by city,marital_status

order by city;
```

Result :

city	marital_status	average_income
Bengaluru	Married	52784
Bengaluru	Single	44852
Chennai	Married	52485
Chennai	Single	46382
Delhi NCR	Married	53344
Delhi NCR	Single	46681
Hyderabad	Married	53735
Hyderabad	Single	46101
Mumbai	Married	54141
Mumbai	Single	44920

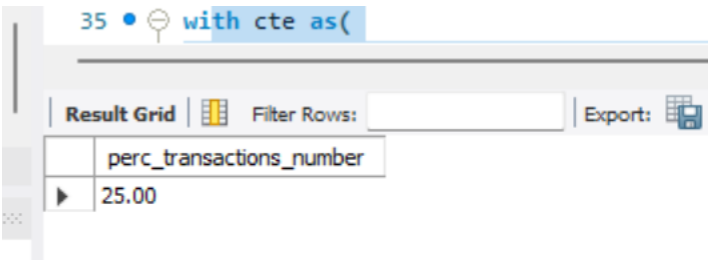
4. Determining Percentage of no. of transactions by credit card as compared to other payment methods:

SQL Query:

```
with cte1 as(

select count(*) as credit_card_transactions_number from spends where payment_type="Credit Card"),

cte2 as(

select count(*) as total_transactions_number from spends)

select
round((credit_card_transactions_number/total_transactions_number)*100,2) as perc_transactions_number from cte1,cte2;
```

-- Out of Total Number of Transactions, 25% of transactions are through credit card.

Result:

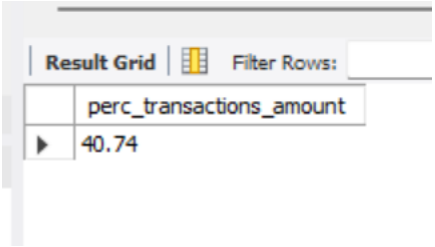


5. Determining Percentage of amount of transactions by credit card as compared to other payment methods

```
with cte1 as(

select sum(spend) as credit_card_transactions_amount from spends where payment_type="Credit Card"),

cte2 as(

select sum(spend) as total_transactions_amount from spends)

select
round((credit_card_transactions_amount/total_transactions_amount)*100,2)
as perc_transactions_amount from cte1,cte2;
```

Insight : 40.74 % of transactions by amount are through credit cards

Result :



6. Analysing top 2 category in each city among different age groups on the basis of amount spent

SQL Query:

```
with cte as(

select
customers.customer_id,customers.age_group,customers.city,customers.occupa
tion,customers.gender,customers.marital_status,customers.avg_income,

spends.months,spends.category,spends.payment_type,spends.spend

from spends left join customers on

spends.customer_id=customers.customer_id where
spends.payment_type='Credit Card'),

cte1 as(

select city, category, sum(spend) as total_transaction_amount, dense_rank()
over ( partition by city order by  sum(spend) desc) as ranking

from cte

group by city,category

order by city,total_transaction_amount desc)

select * from cte1 where ranking in (1,2);
```

Result :

city	category	total_transaction_amount	ranking
Bengaluru	Bills	8698110	1
Bengaluru	Electronics	6495966	2
Chennai	Bills	7286882	1
Chennai	Electronics	5424885	2
Delhi NCR	Bills	9861773	1
Delhi NCR	Electronics	7430903	2
Hyderabad	Bills	6004091	1
Hyderabad	Electronics	4530384	2
Mumbai	Bills	14481730	1
Mumbai	Electronics	11301627	2

Insight : Top two categories in each city are Bills and Electronics

7. Determining top two months in which maximum amount of credit card transactions by amount are done according to each city

SQL Query:

```
with cte as(select
customers.customer_id,customers.age_group,customers.city,customers.occupa
tion,customers.gender,customers.marital_status,customers.avg_income,

spends.months,spends.category,spends.payment_type,spends.spend

from spends left join customers on

spends.customer_id=customers.customer_id where
spends.payment_type='Credit Card'),

cte1 as(select city, months, sum(spend) as total_transaction_amount,
dense_rank() over ( partition by city order by  sum(spend) desc) as ranking

from cte

group by city,months

order by city,total_transaction_amount desc)

select * from cte1 where ranking in (1,2);
```

Result Grid | Filter Rows: | Export: | Wrap

city	months	total_transaction_amount	ranking
Bengaluru	September	8787166	1
Bengaluru	August	7590904	2
Chennai	September	7230843	1
Chennai	August	6335167	2
Delhi NCR	September	10027562	1
Delhi NCR	August	8687816	2
Hyderabad	September	6035030	1
Hyderabad	August	5273212	2
Mumbai	September	15180925	1
Mumbai	August	13195439	2

Result:

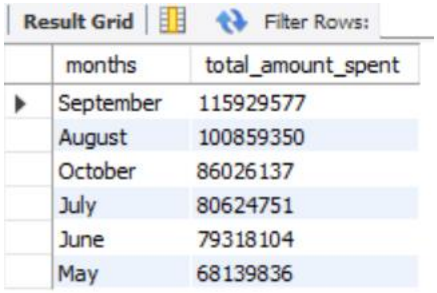
Insight: The top two months in which maximum transactions have been done by amount are common for each city and these months are September and August.

8. Analysing month on month transactions by credit card

SQL Query:

select months,sum(spend) as total_amount_spent from spends group by months order by total_amount_spent desc;

Result:



Result Grid | Filter Rows:

months	total_amount_spent
September	115929577
August	100859350
October	86026137
July	80624751
June	79318104
May	68139836

Insights:

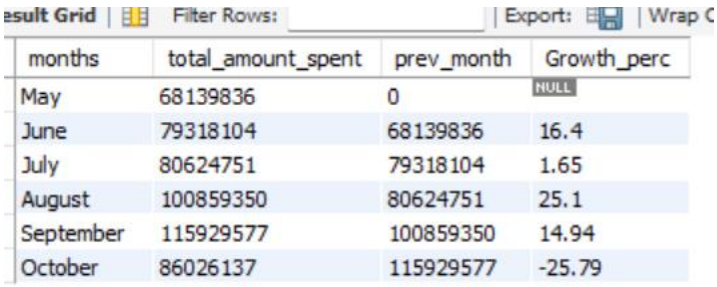
- **Transactions by amount have been on the rise from May to September but there is a dip in October Month.**
- **Maximum Transactions have taken place in September Month.**

9. Determining month on month increase/decrease in transcation amount:

SQL Query:

```
with cte as(

select months,sum(spend) as total_amount_spent from spends group by
months order by

str_to_date(concat('0001',months,'01'),'%Y %M %d')),

cte1 as(

select months,total_amount_spent,lag(total_amount_spent,1,0)  over (order
by

str_to_date(concat('0001',months,'01'),'%Y %M %d')) as prev_month from
cte)

select *, round(((total_amount_spent -prev_month)/prev_month)*100,2) as
Growth_perc from cte1;
```

Result :



esult Grid | Filter Rows: | Export: | Wrap C

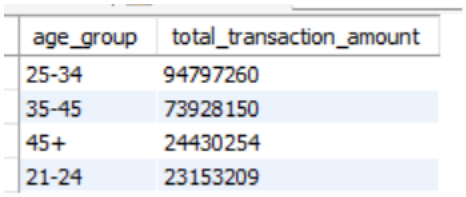
months	total_amount_spent	prev_month	Growth_perc
May	68139836	0	NULL
June	79318104	68139836	16.4
July	80624751	79318104	1.65
August	100859350	80624751	25.1
September	115929577	100859350	14.94
October	86026137	115929577	-25.79

10. Determining the amount of transactions through credit card by age_group

SQL Query:

```
with cte as(select
customers.customer_id,customers.age_group,customers.city,customers.occupa
tion,customers.gender,customers.marital_status,customers.avg_income,

spends.months,spends.category,spends.payment_type,spends.spend

from spends left join customers on

spends.customer_id=customers.customer_id where
spends.payment_type='Credit Card'),

cte1 as(select age_group, sum(spend) as total_transaction_amount

from cte

group by age_group

order by total_transaction_amount desc)

select * from cte1;
```

Result :



age_group	total_transaction_amount
25-34	94797260
35-45	73928150
45+	24430254
21-24	23153209

Insights :

The customers in age_group 25-34 are responsible for maximum transactions (43%) and customers between age_group 21-24 are responsible for minimum transactions.