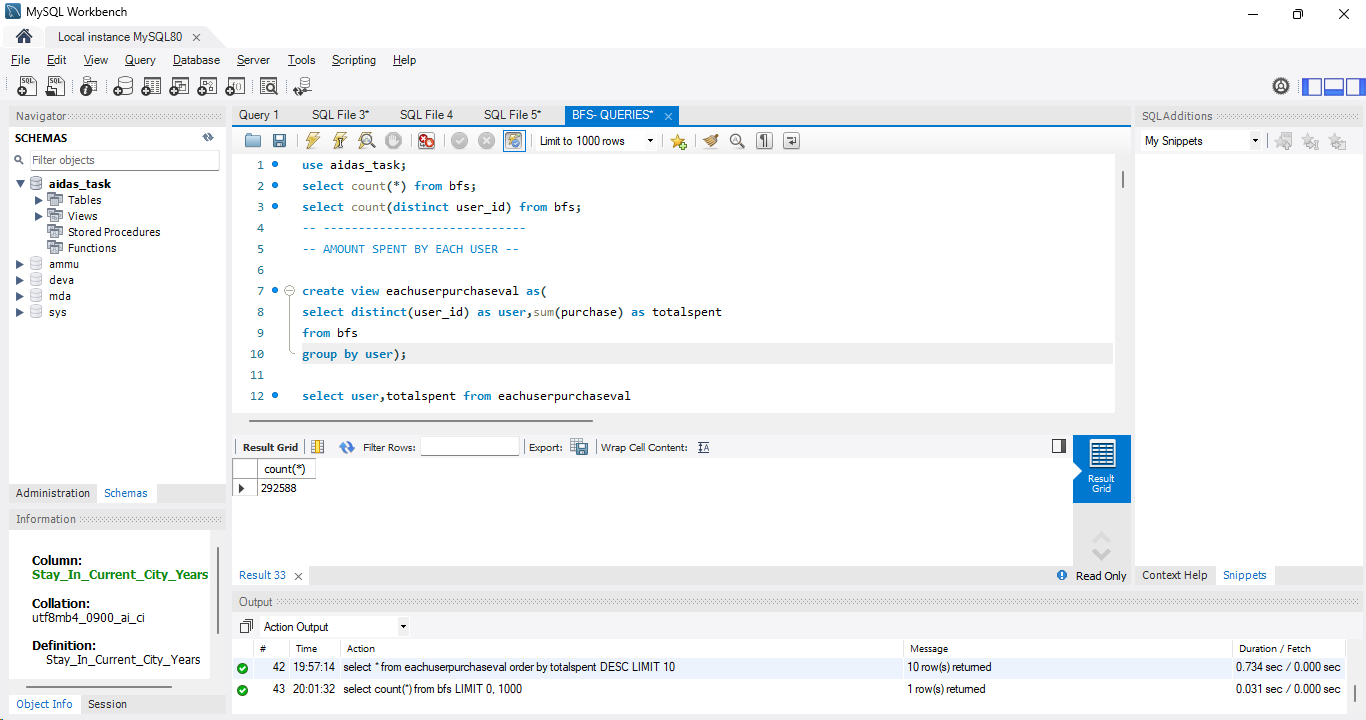
**DEVADHARSHINI N**

* Source - **Black Friday sales.csv**
* Table name – **bfs**
* IDE – **MySQL Workbench 8.0**
* Note :- Due to High number of records only '292588' records were imported in the import process of MySQL workbench.



1. Gender based purchase

select gender,sum(purchase) as 'Sales value'

from bfs

group by gender

order by gender DESC;



1. Highest number of purchase w.r.t userid

-- AMOUNT SPENT BY EACH USER --

create view eachuserpurchaseval as(

select distinct(user\_id) as user,sum(purchase) as totalspent

from bfs

group by user);

select user,totalspent from eachuserpurchaseval

order by totalspent DESC

limit 1;



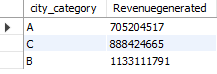
1. Sale based on city category

select city\_category,sum(purchase) as 'Revenuegenerated'

from bfs

group by city\_category

order by 'Revenuegenerated';



1. Which is the best selling product?

select product\_id,count(\*) as 'Count of sold items' from bfs

group by product\_id

order by 'Count of sold items' DESC LIMIT 1;



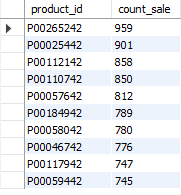
1. Most purchased product top 10 and its category

select product\_id, COUNT(\*) AS count\_sale

from bfs

GROUP BY product\_id

ORDER BY count\_sale DESC LIMIT 10;



1. Number of users based on gender

select gender,count(distinct(user\_id)) as 'Userscount' from BFS

group by gender

order by Userscount DESC;



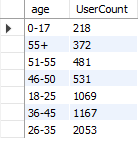
1. Number of users based on age-group

select age,count(distinct user\_id) as UserCount

from bfs

group by age

order by UserCount;



1. Age group with Highest number of users

select age,count(distinct user\_id) as UserCount

from bfs

group by age

order by UserCount

DESC limit 1;



1. Age group with Second Highest number of users

select age,count(distinct user\_id) as UserCount

from bfs

group by age

order by UserCount

DESC limit 1 OFFSET 1;



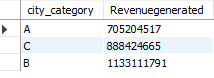
1. Purchase amount of city category

select city\_category,sum(purchase) as 'Revenuegenerated'

from bfs

group by city\_category

order by 'Revenuegenerated';



1. Highest purchase amount of city category

select city\_category,Revenuegenerated from citybasedsale

where revenuegenerated = (select max(revenuegenerated)

from citybasedsale);

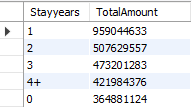


1. Purchase amount by staying years customer

select stay\_in\_current\_city\_years as Stayyears, sum(Purchase) as TotalAmount from bfs

group by Stayyears

order by TotalAmount DESC;



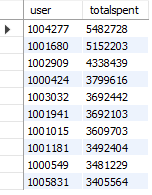
1. Top 10 Customers

create view eachuserpurchaseval as(

select distinct(user\_id) as user,sum(purchase) as totalspent

from bfs group by user);

select \* from eachuserpurchaseval order by totalspent DESC LIMIT 10;



1. Total number of users

select count(distinct(user\_id)) as 'Number of Users' from bfs;



1. Total number of products

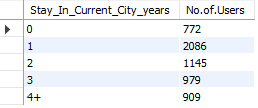
select count(distinct(product\_id)) as 'Total Number of Products' from bfs;



1. Number of users with respect to the stay in current city years

select Stay\_In\_Current\_City\_years, count(distinct(user\_id)) as 'No.of.Users' from bfs

group by stay\_in\_current\_city\_years;



1. Number of user analysis based on age groups to decide promotion levels.

SELECT age, COUNT(DISTINCT user\_id) AS 'No.of.Users',

CASE

WHEN COUNT(DISTINCT user\_id) < 0 THEN 'Invalid count'

WHEN COUNT(DISTINCT user\_id) < 500 THEN 'Attention needed'

WHEN COUNT(DISTINCT user\_id) < 1000 THEN 'Needs Promotion'

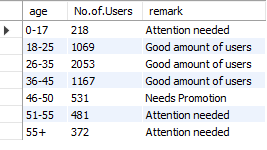
WHEN COUNT(DISTINCT user\_id) > 1000 THEN 'Good amount of users'

ELSE 'Re Examine'

END AS remark

FROM bfs

GROUP BY age;



1. Rank of products based on purchase value

SELECT

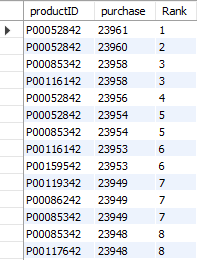
distinct product\_id AS 'product\_ID',

purchase,

DENSE\_RANK() OVER (ORDER BY purchase DESC) AS 'Rank value'

FROM

bfs;



1. Dividing products into clusters Based on Purchase value

SELECT

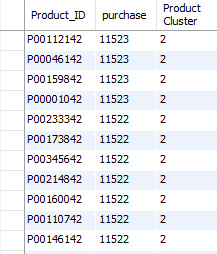
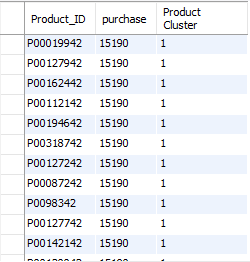
distinct product\_id AS 'Product\_ID',

purchase,

NTILE(5) OVER (ORDER BY purchase DESC) AS 'Product Cluster'

FROM

bfs;



1. Count of Product clusters based on purchase value category

SELECT

count(Product\_ID) as TotalProducts,

SUM(CASE WHEN ProductCluster = 1 THEN 1 ELSE 0 END) AS 'Elite Products',

SUM(CASE WHEN ProductCluster = 2 THEN 1 ELSE 0 END) AS 'Premium Products',

SUM(CASE WHEN ProductCluster = 3 THEN 1 ELSE 0 END) AS 'Midrange Products',

SUM(CASE WHEN ProductCluster = 4 THEN 1 ELSE 0 END) AS 'standard Products',

SUM(CASE WHEN ProductCluster = 5 THEN 1 ELSE 0 END) AS 'Basic Products'

FROM

clusters;

