

```

select monthname(event_time) as mnth ,(count(brand)*price)
sales_amount from events where event_type='purchase'
group by month(event_time)
order by sales_amount desc;

select year, mnth,max(sales_amount) from( select year(event_time)as year,
monthname(event_time) as mnth ,(count(brand)*price)
sales_amount from events where event_type='purchase'
group by month(event_time)
order by sales_amount desc)t1;

select year,mnth,min(sales_amount) from (select year(event_time)as year,
monthname(event_time) as mnth ,(count(brand)*price)as sales_amount
from events where event_type='purchase'
group by month(event_time))t1;

```

The screenshot shows the MySQL Workbench interface. The main window displays a SQL script with three queries. The first query is a simple selection from the 'events' table. The second query updates the 'event_time' column to UTC. The third query is a complex one that calculates monthly sales amounts and orders them by sales amount in descending order.

The 'Result Grid' shows the output of the third query, displaying a table with two columns: 'mnth' and 'sales_amount'. The data is as follows:

mnth	sales_amount
February	1853733.168685913
November	1056724.5123596191
December	302711.84003448486
January	267909.3101501465
September	219092.99737548828
October	171334.795211792

The 'Output' pane at the bottom shows the execution progress of the queries, including the time taken and the number of rows returned.

with anny1 as

(select *,case

when time(event_time) between '00:00:00' and '08:00:00' then 'Latenight or morning'

when time(event_time) between '08:00:00' and '16:00:00' then 'day'

when time(event_time) between '16:00:00' and '23:59:59' then 'evening or night'

end event_view_time

from events),

anny2 as

(select event_view_time,event_type,count(event_view_time) as visits from anny1

group by event_view_time,event_type

order by visits desc)

select event_view_time,event_type,visits from anny2 group by event_type;

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
17 monthname(event_time) as mth,(count(brand)*price)as sales_amount
18 from events where event_type='purchase'
19 group by month(event_time))t1;
20
21 with anny1 as
22 (select *,case
23 when time(event_time) between '00:00:00' and '08:00:00' then 'Latenight or morning'
24 when time(event_time) between '08:00:00' and '16:00:00' then 'day'
25 when time(event_time) between '16:00:00' and '23:59:59' then 'evening or night'
26 end event_view_time
27 from events),
28 anny2 as
29 (select event_view_time,event_type,count(event_view_time) as visits from anny1
group by event_view_time,event_type
order by visits desc)
select event_view_time,event_type,visits from anny2 group by event_type;
```

The Result Grid shows the following data:

event_view_time	event_type	visits
day	view	366652
day	cart	25554
day	purchase	18290

The Output pane shows the execution progress:

#	Time	Action	Message	Duration / Fetch
19	14:50:16	select monthname(event_time) as mth,(count(brand)*price) sales_amount from events where event_type='purchase' group by month(event_time)	6 row(s) returned	5.719 sec / 0.000 sec
20	14:53:30	with anny1 as (select *,case when time(event_time) between '00:00:00' and '08:00:00' then 'Latenight or morning' when time(event_time) between '08:00:00' and '16:00:00' then 'day' when time(event_time) between '16:00:00' and '23:59:59' then 'evening or night' end event_view_time from events), anny2 as (select event_view_time,event_type,count(event_view_time) as visits from anny1 group by event_view_time,event_type order by visits desc) select event_view_time,event_type,visits from anny2 group by event_type;	3 row(s) returned	6.109 sec / 0.000 sec

select (quantity*price)as sale,brand from (select count(brand)as quantity,brand,price from events where event_type='purchase' group by brand order by quantity desc)t1 order by sale desc limit 6;

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
32 select event_view_time,event_type,visits from anny2 group by event_type;
33
34
35 select (quantity*price)as sale,brand from (select count(brand)as quantity,brand,price from events where
36 event_type='purchase' group by brand order by quantity desc)t1 order by sale desc limit 6;
37
38 select count(category_id)as count ,category_code from events where event_type='purchase'
39 group by category_code order by count desc;
40
41 select count(user_id) as count,event_type,user_id
42 from events where event_type='purchase' group by user_id
43 order by count desc;
44
```

The Results Grid shows the output of the query:

sale	brand
847679.2315979004	
834844.3008728027	gigabyte
803723.8974609175	msi
328840.5923461914	canon
279352.26525878906	palt
208375.79315185547	amd

The Action Output pane shows the execution details:

#	Time	Action	Message	Duration / Fetch
20	14:53:30	with anny1 as (select *.case when time(event_time) between '00:00:00' and '08:00:00' then 'Lateng...	3 row(s) returned	6.109 sec / 0.000 sec
21	14:54:18	select (quantity*price)as sale,brand from (select count(brand)as quantity,brand,price from events where eve...	6 row(s) returned	1.750 sec / 0.000 sec

`select count(category_id) as count ,category_code from events where event_type='purchase'`
`group by category_code order by count desc;`

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following queries:

```
32 select event_view_time,event_type,visits from anny2 group by event_type;
33
34
35 select (quantity*price) as sale,brand from (select count(brand) as quantity,brand,price from events where
36 event_type='purchase' group by brand order by quantity desc) t1 order by sale desc limit 6;
37
38 select count(category_id) as count ,category_code from events where event_type='purchase'
39 group by category_code order by count desc;
40
41 select count(user_id) as count,event_type,user_id
42 from events where event_type='purchase' group by user_id
43 order by count desc;
44
```

The Results grid shows the output of the third query:

count	category_code
7568	
6888	computers.components.videocards
4119	electronics.telephone
2739	stationery.cartridge
2557	computers.peripherals.printer
1297	computers.notebook
1266	computers.components.motherboard
1213	computers.components.cpu

The Output pane shows the execution of the queries:

#	Time	Action	Message	Duration / Fetch
21	14:54:18	select (quantity*price) as sale,brand from (select count(brand) as quantity,brand,price from events where eve...	6 row(s) returned	1.750 sec / 0.000 sec
22	14:55:06	select count(category_id) as count ,category_code from events where event_type='purchase' group by c...	93 row(s) returned	1.516 sec / 0.000 sec

```
select count(user_id) as count,event_type,user_id  
from events where event_type='purchase' group by user_id  
order by count desc;
```

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
select event_view_time,event_type,visits from anny2 group by event_type;  
  
select (quantity*price)as sale,brand from (select count(brand)as quantity,brand,price from events where  
event_type='purchase' group by brand order by quantity desc)t1 order by sale desc limit 6;  
  
select count(category_id)as count ,category_code from events where event_type='purchase'  
group by category_code order by count desc;  
  
select count(user_id) as count,event_type,user_id  
from events where event_type='purchase' group by user_id  
order by count desc;
```

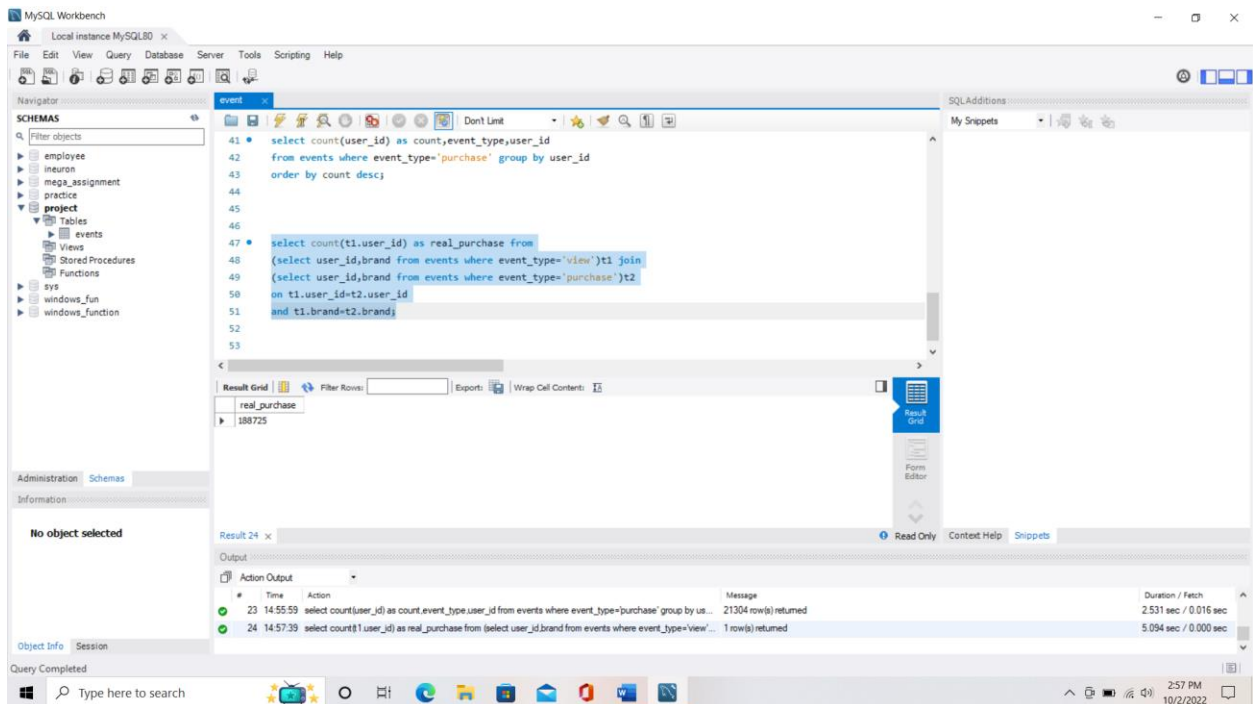
The Results Grid shows the output of the last query:

count	event_type	user_id
56	purchase	1515915625601579158
49	purchase	1515915625591659523
45	purchase	1515915625596534847
43	purchase	1515915625604175669
42	purchase	1515915625605054644
39	purchase	1515915625599634258
36	purchase	1515915625521745364
33	purchase	1515915625603840072

The Output pane shows the execution of the query:

```
22 14:55:06 select count(category_id)as count ,category_code from events where event_type='purchase' group by c... 93 row(s) returned 1.516 sec / 0.000 sec  
23 14:55:59 select count(user_id) as count,event_type,user_id from events where event_type='purchase' group by us... 21304 row(s) returned 2.531 sec / 0.016 sec
```

```
select count(t1.user_id) as real_purchase from  
  
(select user_id,brand from events where event_type='view')t1 join  
  
(select user_id,brand from events where event_type='purchase')t2  
  
on t1.user_id=t2.user_id  
  
and t1.brand=t2.brand;
```



Objectives

- 1)- AS analysis shows after the year change sale was down for January month. need to change the marketing strategies for the new year to grab the attention of customers.
- 2)After viewing the brand the number of customers buying that brand is less which needs to improve To attract the customers to buy the product after seeing it.