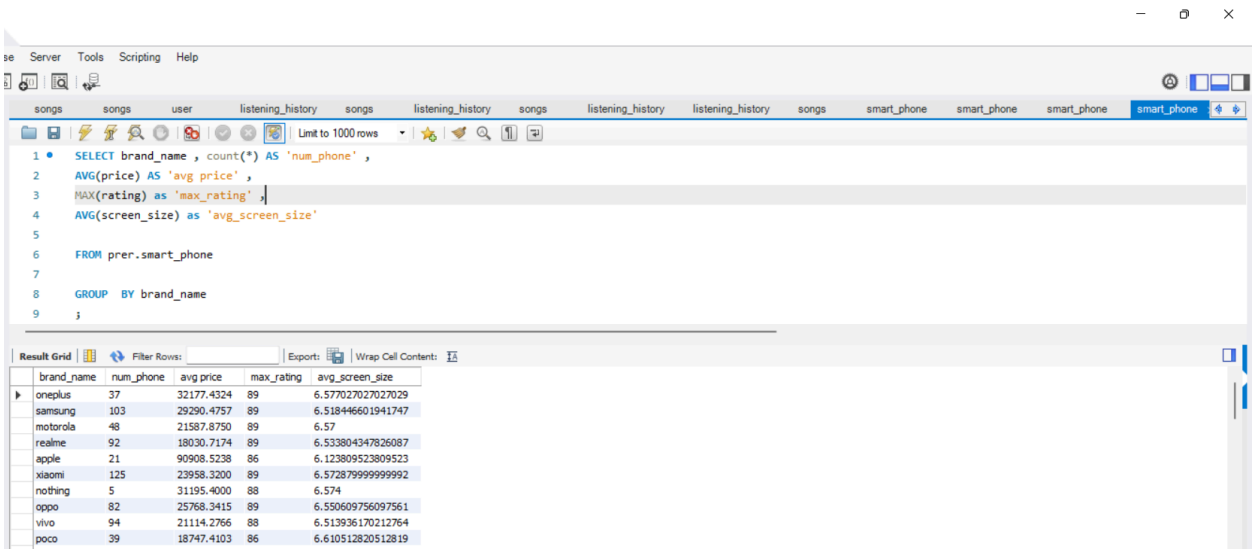
SMARTPHONE DATA ANALYSIS USING GROUPING

Grouping Data

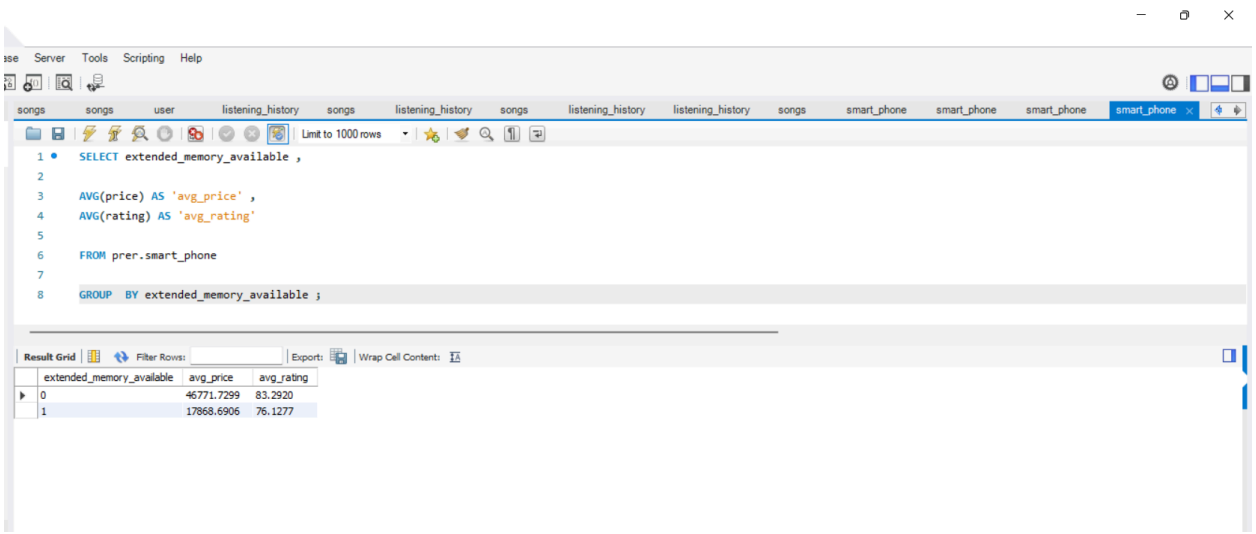
1. Group smartphones by brand and get the count, average price, max rating, avg screen size and avg battery capacity
--> Group By Animation
2. Group smartphones by whether they have an NFC and get the average price and rating
3. Group smartphones by the extended memory available and get the average price
4. Group smartphones by the brand and processor brand and get the count of models and the average primary camera resolution (rear)
5. find top 5 most costly phone brands
6. which brand makes the smallest screen smartphones
7. Avg price of 5g phones vs avg price of non 5g phones
8. Group smartphones by the brand, and find the brand with the highest number of models that have both NFC and an IR blaster
9. Find all samsung 5g enabled smartphones and find out the avg price for NFC and Non-NFC phones
10. find the phone name, price of the costliest phone

QUES 1

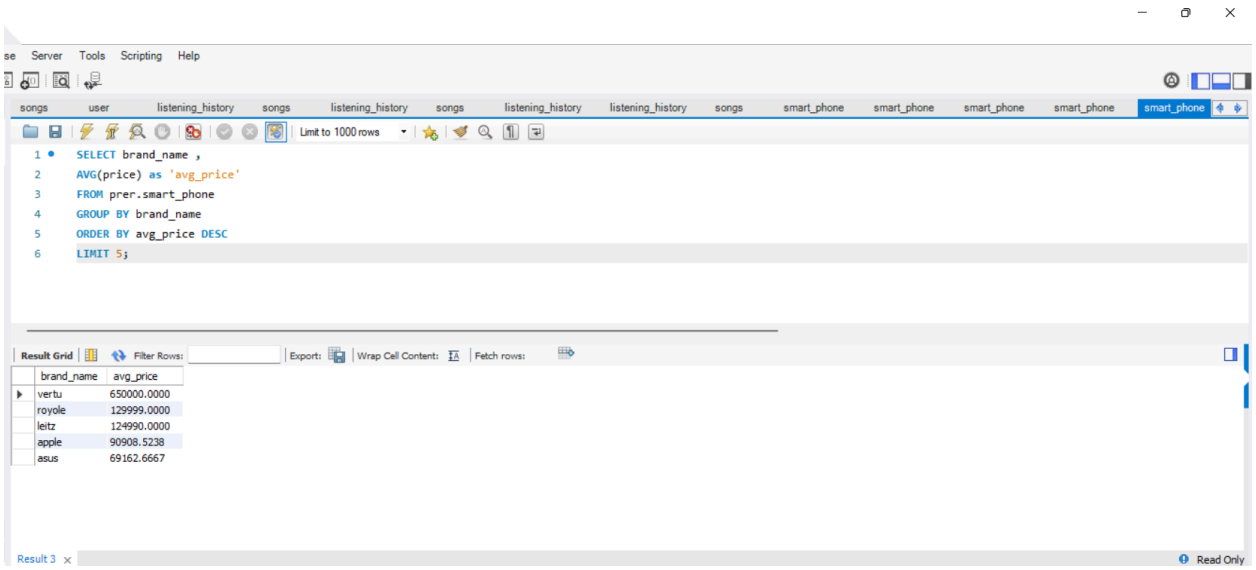


brand_name	num_phone	avg price	max_rating	avg_screen_size
oneplus	37	32177.4324	89	6.577027027027029
samsung	103	29290.4757	89	6.518446601941747
motorola	48	21587.8750	89	6.57
realme	92	18030.7174	89	6.533804347826087
apple	21	90908.5238	86	6.123809523809523
xiaomi	125	23958.3200	89	6.572879999999992
nothing	5	31195.4000	88	6.574
oppo	82	25768.3415	89	6.550609756097561
vivo	94	21114.2766	88	6.513936170212764
poco	39	18747.4103	86	6.610512820512819

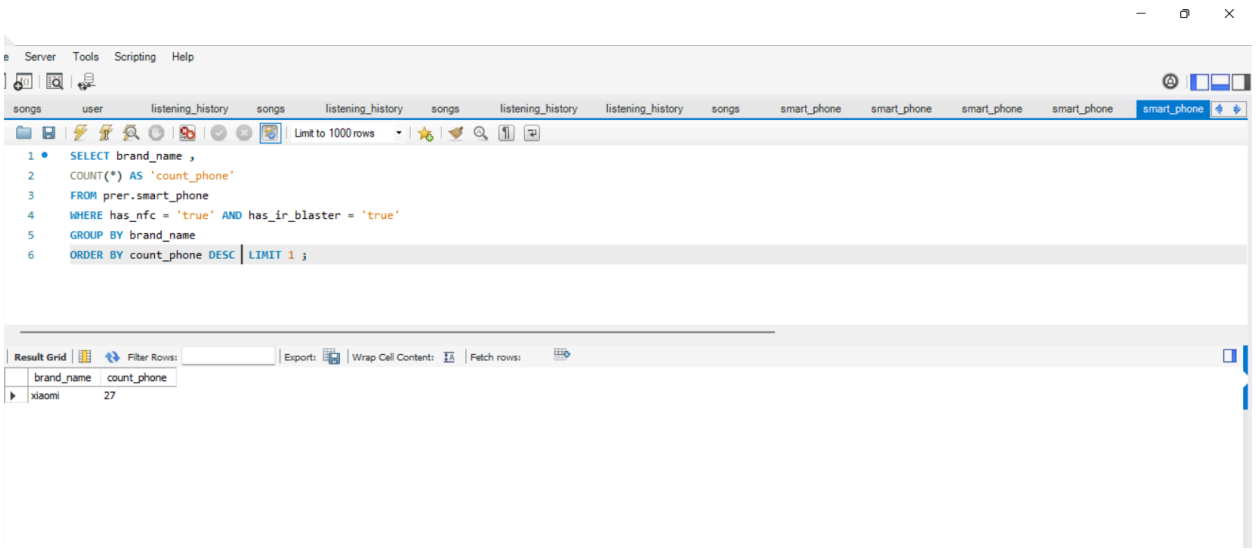
QUES 3


Server Tools Scripting Help
songs songs user listening_history songs listening_history songs listening_history listening_history songs smart_phone smart_phone smart_phone smart_phone
Limit to 1000 rows
1 SELECT extended_memory_available ,
2
3 AVG(price) AS 'avg_price' ,
4 AVG(rating) AS 'avg_rating'
5
6 FROM prer.smart_phone
7
8 GROUP BY extended_memory_available ;
Result Grid Filter Rows: Export: Wrap Cell Content:
extended_memory_available avg_price avg_rating
0 46771.7299 83.2920
1 17868.6906 76.1277

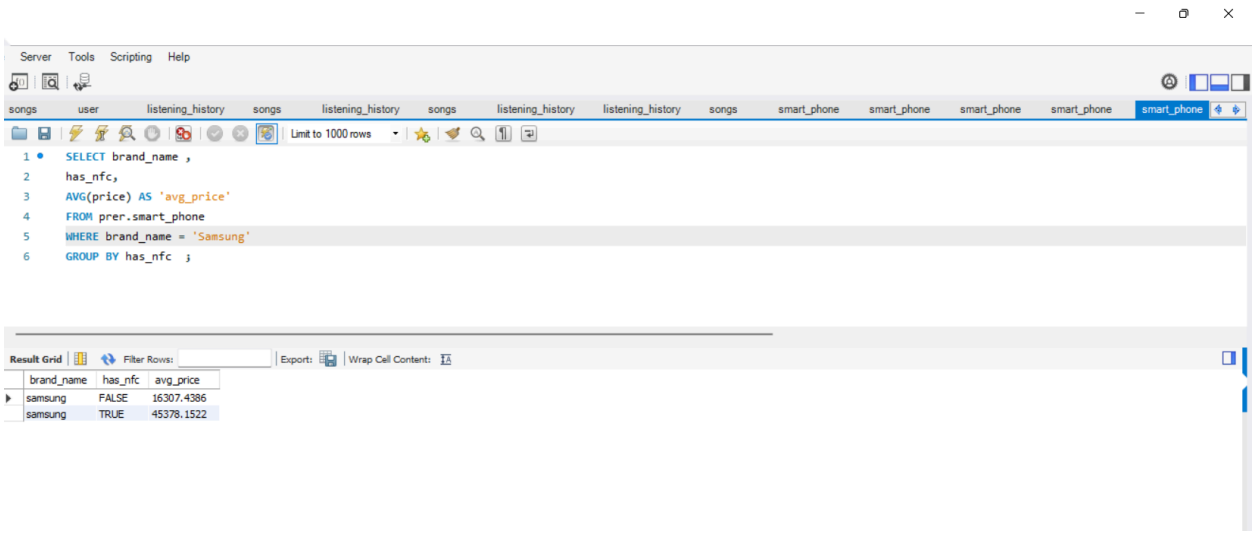
QUES 5


Server Tools Scripting Help
songs user listening_history songs listening_history songs listening_history listening_history songs smart_phone smart_phone smart_phone smart_phone smart_phone
Limit to 1000 rows
1 SELECT brand_name ,
2 AVG(price) as 'avg_price'
3 FROM prer.smart_phone
4 GROUP BY brand_name
5 ORDER BY avg_price DESC
6 LIMIT 5;
Result Grid Filter Rows: Export: Wrap Cell Content: Fetch rows:
brand_name avg_price
vertu 650000.0000
royole 129999.0000
leitz 124990.0000
apple 90908.5238
asus 69162.6667
Result 3
Read Only

QUES 8


Server Tools Scripting Help
songs user listening_history songs listening_history songs listening_history listening_history songs smart_phone smart_phone smart_phone smart_phone smart_phone
Limit to 1000 rows
SELECT brand_name ,
COUNT(*) AS 'count_phone'
FROM prer.smart_phone
WHERE has_nfc = 'true' AND has_ir_blaster = 'true'
GROUP BY brand_name
ORDER BY count_phone DESC LIMIT 1 ;
Result Grid Filter Rows: Export: Wrap Cell Content: Fetch rows:
brand_name count_phone
xiaomi 27

QUES 9


Server Tools Scripting Help
songs user listening_history songs listening_history songs listening_history listening_history songs smart_phone smart_phone smart_phone smart_phone smart_phone
Limit to 1000 rows
SELECT brand_name ,
has_nfc,
AVG(price) AS 'avg_price'
FROM prer.smart_phone
WHERE brand_name = 'Samsung'
GROUP BY has_nfc ;
Result Grid Filter Rows: Export: Wrap Cell Content:
brand_name has_nfc avg_price
samsung FALSE 16307.4386
samsung TRUE 45378.1522