

DMDD ASSIGNMENT 3:

USE CASES:

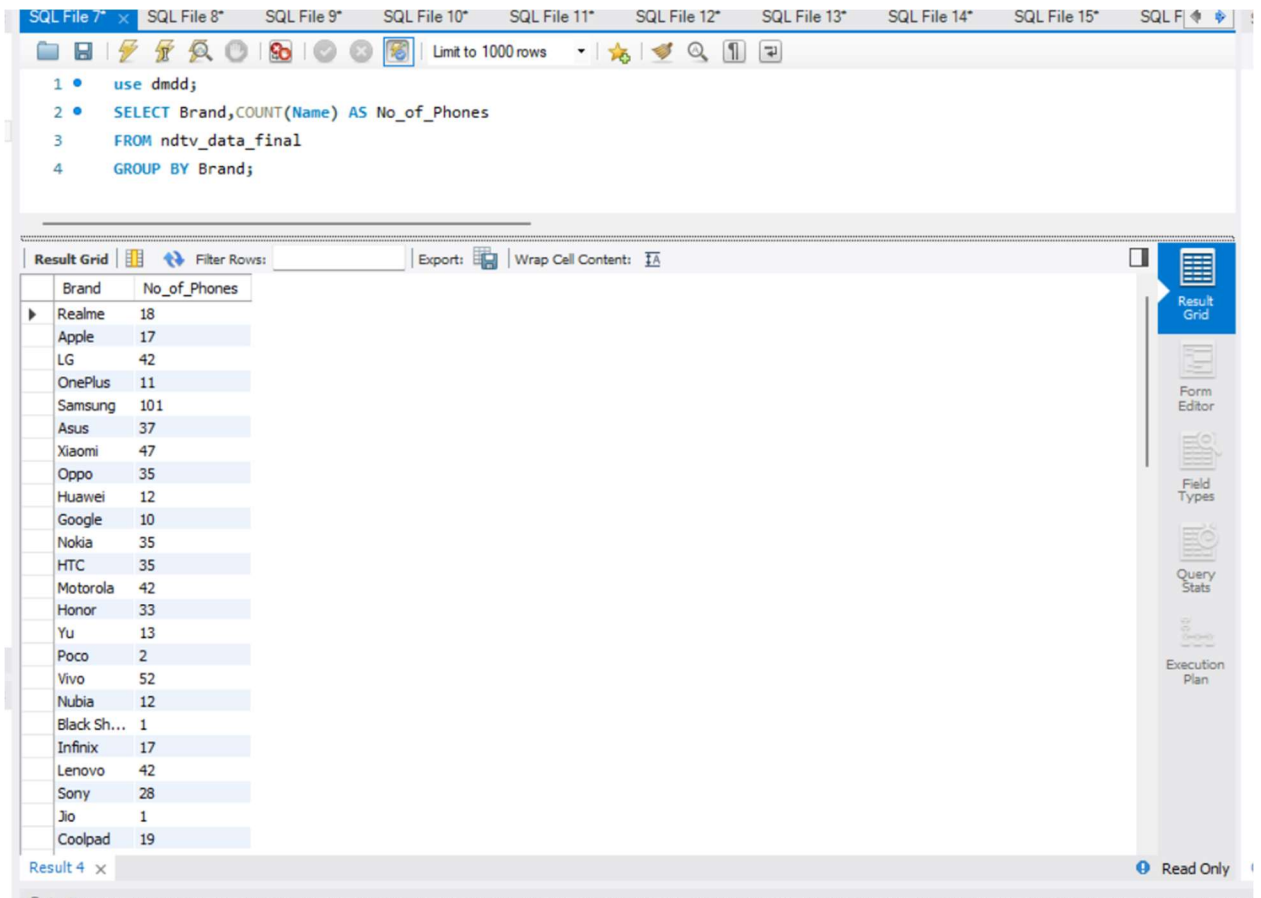
1. Which brand have what quantity of phones?

use dmdd;

```
SELECT Brand,COUNT(Name) AS No_of_Phones
```

```
FROM ndtv_data_final
```

```
GROUP BY Brand;
```



The screenshot shows a SQL IDE interface with multiple tabs for SQL files. The active tab is 'SQL File 7*', which contains the following SQL query:

```
1 • use dmdd;
2 • SELECT Brand,COUNT(Name) AS No_of_Phones
3 • FROM ndtv_data_final
4 • GROUP BY Brand;
```

Below the query editor, the 'Result Grid' is displayed, showing the results of the query. The grid has two columns: 'Brand' and 'No_of_Phones'. The data is as follows:

Brand	No_of_Phones
Realme	18
Apple	17
LG	42
OnePlus	11
Samsung	101
Asus	37
Xiaomi	47
Oppo	35
Huawei	12
Google	10
Nokia	35
HTC	35
Motorola	42
Honor	33
Yu	13
Poco	2
Vivo	52
Nubia	12
Black Sh...	1
Infinix	17
Lenovo	42
Sony	28
Jio	1
Coolpad	19

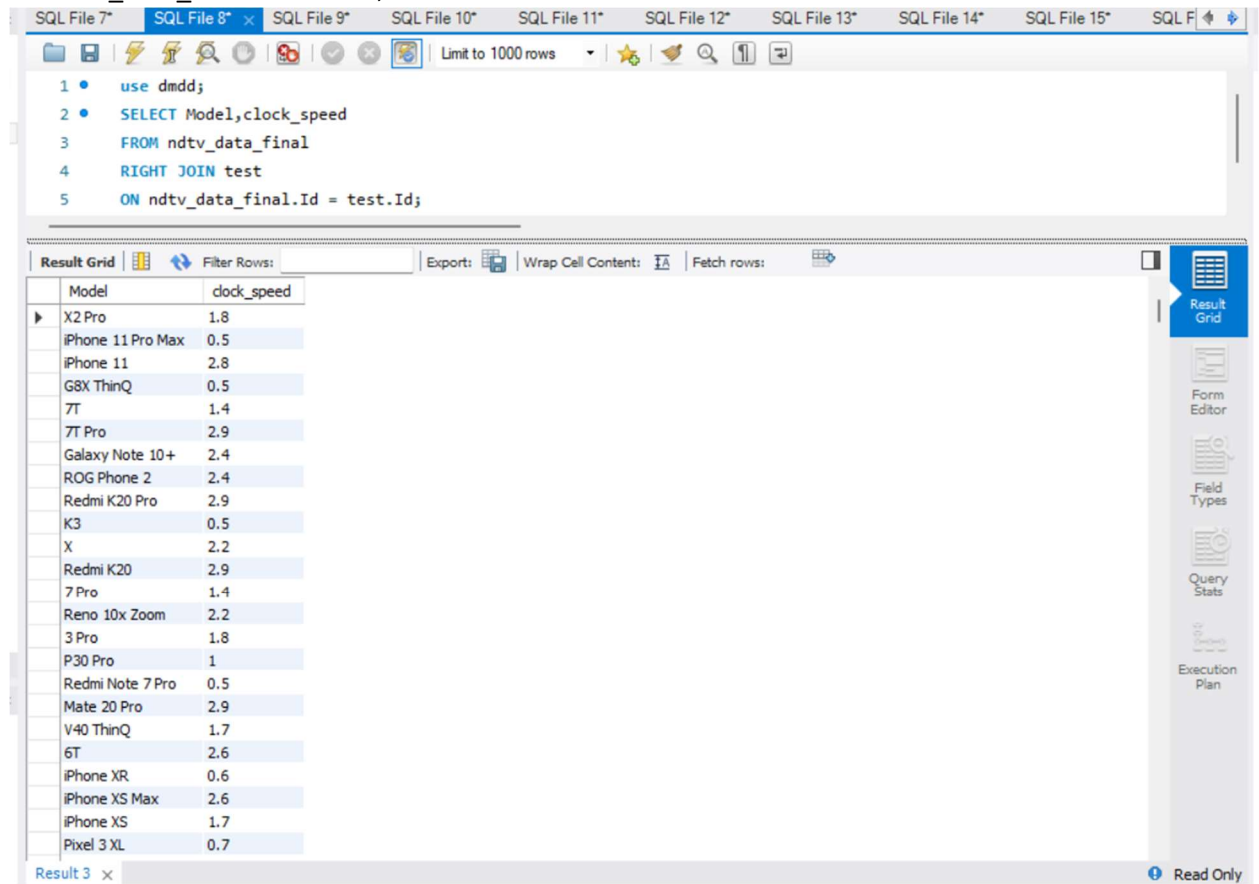
The right sidebar of the IDE contains several icons for 'Result Grid', 'Form Editor', 'Field Types', 'Query Stats', and 'Execution Plan'. The status bar at the bottom indicates 'Result 4' and 'Read Only'.

2. Which model of the smartphone has what clockspeed?

```

use dmdd;
SELECT Model,clock_speed
FROM ndtv_data_final
RIGHT JOIN test
ON ndtv_data_final.Id = test.Id;

```



The screenshot shows a SQL IDE with multiple tabs. The active tab is 'SQL File 8'. The query editor contains the following SQL code:

```

1 • use dmdd;
2 • SELECT Model,clock_speed
3   FROM ndtv_data_final
4   RIGHT JOIN test
5   ON ndtv_data_final.Id = test.Id;

```

Below the query editor, the 'Result Grid' is displayed, showing the results of the query. The table has two columns: 'Model' and 'clock_speed'. The results are as follows:

Model	clock_speed
X2 Pro	1.8
iPhone 11 Pro Max	0.5
iPhone 11	2.8
GSX ThinQ	0.5
7T	1.4
7T Pro	2.9
Galaxy Note 10+	2.4
ROG Phone 2	2.4
Redmi K20 Pro	2.9
K3	0.5
X	2.2
Redmi K20	2.9
7 Pro	1.4
Reno 10x Zoom	2.2
3 Pro	1.8
P30 Pro	1
Redmi Note 7 Pro	0.5
Mate 20 Pro	2.9
V40 ThinQ	1.7
6T	2.6
iPhone XR	0.6
iPhone XS Max	2.6
iPhone XS	1.7
Pixel 3 XL	0.7

The IDE interface includes a toolbar with icons for file operations, a 'Limit to 1000 rows' dropdown, and a sidebar with options like 'Result Grid', 'Form Editor', 'Field Types', 'Query Stats', and 'Execution Plan'.

3. Which brand has what release dates of its phones? (cross join)

```

use dmdd;

SELECT ndtv_data_final.Brand, output.release_date AS Release_Date
FROM output
CROSS JOIN dmdd.ndtv_data_final;

```

SQL File 7* SQL File 8* SQL File 9* x SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL F

Limit to 1000 rows

```

1 • use dmdd;
2 • SELECT ndtv_data_final.Brand, output.release_date AS Release_Date
3 FROM output
4 CROSS JOIN dmdd.ndtv_data_final;

```

Result Grid Filter Rows: Export: Wrap Cell Content: Fetch rows:

	Brand	Release_Date
▶	Realme	20-Oct
	Realme	19-Nov
	Realme	18-Sep
	Realme	18-Sep
	Realme	18-Sep
	Realme	18-Sep
	Realme	18-Sep
	Realme	18-Sep
	Realme	18-Sep
	Realme	18-Nov
	Realme	18-Nov
	Realme	20-Nov
	Realme	18-Sep
	Realme	18-Sep
	Realme	20-Nov
	Realme	18-Sep
	Realme	17-Sep
	Realme	17-Sep
	Realme	16-Mar
	Realme	16-Mar
	Realme	17-Mar
	Realme	20-Nov
	Realme	20-Apr
	Realme	20-Nov

Result 8 x Read Only

4. Which smartphone has what weight, pixel height and pixel width?

use dmdd;

SELECT Name,mobile_wt AS Mobile_Weight,px_height As Pixel_Height,px_width AS Pixel_Width

FROM ndtv_data_final

LEFT JOIN test

ON ndtv_data_final.Id = test.Id;

Limit to 1000 rows

```

1 • use dmdd;
2 • SELECT Name,mobile_wt AS Mobile_Weight,px_height As Pixel_Height,px_width AS Pixel_Width
3   FROM ndtv_data_final
4  LEFT JOIN test
5  ON ndtv_data_final.Id = test.Id;

```

Result Grid

	Name	Mobile_Weight	Pixel_Height	Pixel_Width
▶	Realme X2 Pro	193	226	1412
	iPhone 11 Pro Max	191	746	857
	iPhone 11	186	1270	1366
	LG G8X ThinQ	96	295	1752
	OnePlus 7T	108	749	810
	OnePlus 7T Pro	198	569	939
	Samsung Galaxy Note 10+	156	1283	1374
	Asus ROG Phone 2	111	1312	1880
	Xiaomi Redmi K20 Pro	101	556	876
	Oppo K3	171	52	1009
	Realme X	80	503	1336
	Xiaomi Redmi K20	171	235	1671
	OnePlus 7 Pro	87	829	1893
	Oppo Reno 10x Zoom	158	227	1856
	Realme 3 Pro	193	1315	1323
	Huawei P30 Pro	106	185	1832
	Redmi Note 7 Pro	163	613	1511
	Huawei Mate 20 Pro	186	335	532
	LG V40 ThinQ	194	82	1771
	OnePlus 6T	189	47	559
	Apple iPhone XR	151	504	1930
	Apple iPhone XS Max	101	521	1591
	Apple iPhone XS	115	777	1587
	Google Pixel 2 XL	173	512	1111

Result 5 x

Read Only

5. Which phone model has what release date? (inner join)

use dmdd;

SELECT Model,release_date

FROM ndtv_data_final

INNER JOIN output

ON ndtv_data_final.Id= output.Id;

SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* x SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL F

Limit to 1000 rows

```

1 • use dmdd;
2 • SELECT Model,release_date
3 FROM ndtv_data_final
4 INNER JOIN output
5 ON ndtv_data_final.Id= output.Id;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Model	release_date
X2 Pro	20-Sep
iPhone 11 Pro Max	20-Sep
iPhone 11	20-Sep
7T	18-Jun
Galaxy Note 10+	19-Jan
ROG Phone 2	21-Jan
Redmi K20 Pro	18-Dec
K3	18-Dec
X	18-Dec
P20 Pro	18-May
6	18-May
ZenFone 5Z (ZS6...	18-May
iPhone 8 Plus	18-Nov
iPhone X	18-Nov
Moto Z2 Force	20-Mar
U11	20-Nov
8	18-Sep
Galaxy S8	20-Sep
G6	20-Oct
3T	20-Oct
Galaxy C9 Pro	20-Oct
iPhone 7	20-Sep
Galaxy A8	19-Sep

Result 3 x Read Only

Result Grid
Form Editor
Field Types
Query Stats
Execution Plan

6. What is the total number of brands?

```
use dmdd;
```

```
SELECT Brand FROM ndtv_data_final
```

```
UNION
```

```
SELECT brand_name FROM output;
```

The screenshot shows a SQL IDE interface with multiple tabs at the top labeled 'SQL File 7*' through 'SQL File 15*'. The active tab is 'SQL File 12*'. The query editor contains the following SQL code:

```
1 • use dmdd;  
2 • SELECT Brand FROM ndtv_data_final  
3 • UNION  
4 • SELECT brand_name FROM output;
```

Below the query editor is the 'Result Grid' section. It includes a 'Filter Rows' field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The results are displayed in a table with one column, 'Brand'. The table contains 20 rows of phone brands. On the right side of the interface, there is a vertical toolbar with icons for 'Result Grid', 'Form Editor', 'Field Types', 'Query State', and 'Execution Plan'. At the bottom right, there is a 'Read Only' status indicator.

Brand
Realme
Apple
LG
OnePlus
Samsung
Asus
Xiaomi
Oppo
Huawei
Google
Nokia
HTC
Motorola
Honor
Yu
Poco
Vivo
Nubia
Black Sh...
Infinix
Lenovo
Sony
Jio
Coolpad

7. What is the operating system for what number of phones?

```
use dmdd;
```

```
SELECT Operating_system, count(Id) AS No_of_Phones
```

```
FROM ndtv_data_final
```

```
GROUP BY Operating_system
```

```
HAVING count(Id) > 5;
```

The screenshot shows a SQL IDE with multiple tabs. The active tab, 'SQL File 13', contains the following SQL query:

```
1 • use dmdd;  
2 • SELECT Operating_system, count(Id) AS No_of_Phones  
3 FROM ndtv_data_final  
4 GROUP BY Operating_system  
5 HAVING count(Id) > 5;
```

Below the query editor, the 'Result Grid' displays the results of the query. The grid has two columns: 'Operating_system' and 'No_of_Phones'. The data is as follows:

Operating_system	No_of_Phones
Android	1298
iOS	17
Cyanogen	10
BlackBerry	10
Windows	19

The IDE interface includes a toolbar with various icons, a 'Filter Rows' field, an 'Export' button, and a 'Wrap Cell Content' checkbox. On the right side, there is a vertical toolbar with icons for 'Result Grid', 'Form Editor', 'Field Types', 'Query Stats', and 'Execution Plan'. The bottom status bar shows 'Result 5' and 'Read Only'.

8. What is the total number of phones from both the dataset?

```
use dmdd;
```

```
SELECT Model From ndtv_data_final
```

```
UNION ALL
```

```
SELECT model_name FROM output;
```

SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* x SQL File 15* SQL F

Limit to 1000 rows

```

1 • use dmdd;
2 • SELECT Model From ndtv_data_final
3 • UNION ALL
4 • SELECT model_name FROM output;

```

Result Grid

Model
X2 Pro
iPhone 11 Pro Max
iPhone 11
G8X ThinQ
7T
7T Pro
Galaxy Note 10+
ROG Phone 2
Redmi K20 Pro
K3
X
Redmi K20
7 Pro
Reno 10x Zoom
3 Pro
P30 Pro
Redmi Note 7 Pro
Mate 20 Pro
V40 ThinQ
6T
iPhone XR
iPhone XS Max
iPhone XS
Pixel 3 XL

Result 5 x Read Only

9. What is the average price of all the smartphones?

```
use dmdd;
```

```
SELECT AVG(ndtv_data_final.Price) AS Average_Price
```

```
FROM ndtv_data_final
```

```
INNER JOIN output ON ndtv_data_final.Id = output.Id;
```


The screenshot shows a SQL IDE with multiple tabs labeled 'SQL File 7*' through 'SQL File 15*'. The active tab is 'SQL File 15*'. The query editor contains the following SQL code:

```
1 • use dmdd;  
2 • SELECT AVG(ndtv_data_final.Price) AS Average_Price  
3 FROM ndtv_data_final  
4 INNER JOIN output ON ndtv_data_final.Id = output.Id;
```

Below the query editor is the 'Result Grid' section. It includes a 'Filter Rows:' field, an 'Export:' button, and a 'Wrap Cell Content:' checkbox. The result grid displays a single row with the column 'Average_Price' and the value '11316.2311'. On the right side of the IDE, there is a vertical toolbar with icons for 'Result Grid', 'Form Editor', 'Field Types', 'Query Stats', and 'Execution Plan'. The bottom status bar shows 'Result 5' and 'Read Only'.

10. Which brands have what number of phones with internal storage more than 128GB?

```
use dmdd;
```

```
SELECT Brand,count(Internal_storage) AS Phones_with_higher_storage
```

```
FROM ndtv_data_final
```

```
WHERE Internal_storage>=128
```

```
GROUP BY Brand;
```

SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL File 16* x

Limit to 1000 rows

```

1 use dmdd;
2 SELECT Brand, count(Internal_storage) AS Phones_with_higher_storage
3 FROM ndtv_data_final
4 WHERE Internal_storage >= 128
5 GROUP BY Brand;

```

Result Grid Filter Rows: Export: Wrap Cell Content: [A](#)

Brand	Phones_with_higher_storage
LG	4
OnePlus	5
Samsung	16
Asus	4
Xiaomi	2
Realme	1
Oppo	11
Huawei	6
HTC	2
Vivo	9
Nubia	3
Motorola	2
Black Sh...	1
Honor	6
Nokia	1
Infinix	1
Micromax	1
Meizu	1
Panasonic	1
Tecno	1

Result 5 x Read Only

Result Grid
Form Editor
Field Types
Query Stats
Execution Plan

11. Which phones have what processor count and core capacity?

use dmdd;

SELECT model, pc AS Processor_Count, n_cores AS Core_Capacity

FROM ndtv_data_final

LEFT OUTER JOIN test

ON ndtv_data_final.Id = test.Id

SQL File 10*				SQL File 11*				SQL File 12*				SQL File 13*				SQL File 14*				SQL File 15*				SQL File 16*				SQL File 17* x				SQL File 18*			
Limit to 1000 rows																																			
1 • use dmdd;				2 • SELECT model,pc AS Processor_Count, n_cores AS Core_Capacity				3 • FROM ndtv_data_final				4 • LEFT OUTER JOIN test				5 • ON ndtv_data_final.Id = test.Id																			
Result Grid				Filter Rows:				Export:				Wrap Cell Content:				Fetch rows:																			
model				Processor_Count				Core_Capacity																											
X2 Pro				16				3																											
iPhone 11 Pro Max				12				5																											
iPhone 11				4				3																											
G8X ThinQ				20				8																											
7T				18				6																											
7T Pro				9				8																											
Galaxy Note 10+				3				2																											
ROG Phone 2				2				1																											
Redmi K20 Pro				19				5																											
K3				20				3																											
X				6				7																											
Redmi K20				6				3																											
7 Pro				3				2																											
Reno 10x Zoom				15				5																											
3 Pro				9				8																											
P30 Pro				7				8																											
Redmi Note 7 Pro				17				2																											
Mate 20 Pro				15				4																											
V40 ThinQ				3				2																											
6T				20				4																											
iPhone XR				17				1																											
iPhone XS Max				5				5																											

12. What is the highest and lowest price of the phones of each brand?

use dmdd;

SELECT output.Id, Name,Brand, highest_price,lowest_price

FROM ndtv_data_final

RIGHT OUTER JOIN output

ON ndtv_data_final.Id = output.Id

SQL File 10* SQL File 11* SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL File 16* SQL File 17* SQL File 18* x

Limit to 1000 rows

```

1 • use dmdd;
2 • SELECT output.Id, Name, Brand, highest_price, lowest_price
3   FROM ndtv_data_final
4  RIGHT OUTER JOIN output
5    ON ndtv_data_final.Id = output.Id

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Id	Name	Brand	highest_price	lowest_price
▶	1	Realme X2 Pro	Realme	2489	1659
	2	iPhone 11 Pro Max	Apple	2489	1659
	3	iPhone 11	Apple	2489	1659
	5	OnePlus 7T	OnePlus	11099	10631
	7	Samsung Galaxy Note 10+	Samsung	5295	4733
	8	Asus ROG Phone 2	Asus	5222	4990
	9	Xiaomi Redmi K20 Pro	Xiaomi	5372	4646
	10	Oppo K3	Oppo	5372	4646
	11	Realme X	Realme	5559	4897
	30	Huawei P20 Pro	Huawei	5448	4970
	31	OnePlus 6	OnePlus	5448	4970
	32	Asus ZenFone 5Z (ZS620KL)	Asus	5448	4970
	40	Apple iPhone 8 Plus	Apple	6526	5290
	41	Apple iPhone X	Apple	5759	4599
	42	Motorola Moto Z2 Force	Motorola	5699	5599
	43	HTC U11	HTC	7786	6840
	44	Nokia 8	Nokia	8196	7049
	46	Samsung Galaxy S8	Samsung	3185	2168
	47	LG G6	LG	3299	2889
	48	OnePlus 3T	OnePlus	3899	3581
	49	Samsung Galaxy C9 Pro	Samsung	3899	3581
	52	Apple iPhone 7	Apple	3299	2999

Result 3 x Read Only

13. What is the name, popularity and battery power for each mobile phone where popularity is greater than 500?

```
use dmdd;
```

```
SELECT Name, popularity, battery_power
```

```
FROM ndtv_data_final
```

```
JOIN output
```

```
ON ndtv_data_final.Id = output.Id
```

```
JOIN test
```

```
ON output.Id = test.Id
```

```
WHERE popularity > 500;
```

SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL File 16* SQL File 17* SQL File 18* SQL File 19* x SQL File 20*

Limit to 1000 rows

```

1 • use dmdd;
2 • SELECT Name, popularity, battery_power
3 FROM ndtv_data_final
4 JOIN output
5 ON ndtv_data_final.Id = output.Id
6 JOIN test
7 ON output.Id = test.Id
8 WHERE popularity > 500;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: I

Name	popularity	battery_power
Apple iPhone 8 Plus	1111	567
Apple iPhone X	1144	1952
Motorola Moto Z2 Force	663	822
HTC U11	900	685
Nokia 8	1063	1388
Samsung Galaxy S8	616	1411
LG G6	629	1094
OnePlus 3T	790	1653
Samsung Galaxy C9 Pro	793	916
Apple iPhone 7	1031	632
Samsung Galaxy A8	883	578
Poco X2	1132	1206
Samsung Galaxy Note ...	921	1366
Realme XT	1156	1391
Redmi Note 8 Pro	701	979
Realme 5 Pro	1139	1999
Asus 6Z	1118	1982
Samsung Galaxy M40	1117	1373
Redmi Note 7S	620	1151
Motorola One Vision	798	1650

Result 3 x Read Only

Result Grid
Form Editor
Field Types
Query Stats
Execution Plan

14. Which phones have the mobile weight less than 100?

use dmdd;

SELECT Id, Name

FROM ndtv_data_final

WHERE EXISTS (SELECT mobile_wt FROM test WHERE ndtv_data_final.Id = test.Id AND mobile_wt < 100);

SQL File 18* SQL File 19* SQL File 20* x SQL File 21* SQL File 22* SQL File 23* SQL File 24* SQL File 25* SQL File 26*

Limit to 2000 rows

```

1 • use dmdd;
2 • SELECT Id,Name
3 FROM ndtv_data_final
4 WHERE EXISTS (SELECT mobile_wt FROM test WHERE ndtv_data_final.Id = test.Id AND mobile_wt < 100);

```

Result Grid

Id	Name
4	LG G8X ThinQ
11	Realme X
13	OnePlus 7 Pro
24	Google Pixel 3 XL
28	LG G7+ ThinQ
30	Huawei P20 Pro
37	HTC U11+
40	Apple iPhone 8 Plus
46	Samsung Galaxy S8
60	Google Nexus 6P
61	Apple iPhone 6s Plus
66	Xiaomi Redmi Note
89	Asus 6Z
90	Samsung Galaxy M40
92	Motorola One Vision
96	Samsung Galaxy A70
100	Vivo V15 Pro
103	Samsung Galaxy M20
108	Realme U1
123	Nokia 6.1 Plus
127	Realme 1
136	Infinix Zero 5 Pro
139	Honor 9i
147	Google Pixel 3 XL

ndtv_data_final 3 x

Read Only

15. The brand name APPLE has what number of sellers?

use dmdd;

SELECT ndtv_data_final.Id, Brand, sellers_amount AS No_of_sellers

FROM ndtv_data_final

JOIN output

ON ndtv_data_final.Id = output.Id

where ndtv_data_final.Brand='Apple' and output.sellers_amount<20;

SQL File 18* SQL File 19* SQL File 20* **SQL File 21*** SQL File 22* SQL File 23* SQL File 24* SQL File 25* SQL File 26*

Limit to 2000 rows

```

1 • use dmdd;
2 • SELECT ndtv_data_final.Id, Brand, sellers_amount AS No_of_sellers
3   FROM ndtv_data_final
4  JOIN output
5  ON ndtv_data_final.Id = output.Id
6  where ndtv_data_final.Brand='Apple' and output.sellers_amount<20;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Id	Brand	No_of_sellers
▶	40	Apple	11
	41	Apple	19
	52	Apple	17
	192	Apple	10

Result 5 x Read Only

Form Editor
Field Types
Query Stats
Execution Plan

16. Which of the following mobile phones are black in colour with their prices above 40000?

use dmdd;

SELECT model_name, Price

FROM ndtv_data_final

JOIN output

ON ndtv_data_final.Id = output.Id

where output.model_name like '%Black%' and ndtv_data_final.Price>40000;

SQL File 18* SQL File 19* SQL File 20* SQL File 21* **SQL File 22*** SQL File 23* SQL File 24* SQL File 25* SQL File 26*

Limit to 2000 rows

```

1 • use dmdd;
2 • SELECT model_name, Price
3   FROM ndtv_data_final
4  JOIN output
5  ON ndtv_data_final.Id = output.Id
6  where output.model_name like '%Black%' and ndtv_data_final.Price>40000;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	model_name	Price
▶	1 5033D 1/16GB Volcano Black (5033D-2LALUAF)	106900
	1 5033D 1/16GB Volcano Black (5033D-2LALUAF)	62900
	10 Lite 4/64GB Black	79699
	15 4/64GB Black	49990
	16 6/128GB Black	47000
	16 6/64GB Black	69999
	8x Max 4/64GB Black	59090
	A60 Pro 3/16GB Black	54665
	BV9600 Pro 6/128GB Black	50650
	BV9600 Pro 6/128GB Black	56699
	BV9800 6/128Gb Black	57915
	Galaxy A20e SM-A202F 3/32GB Black SM-A202F...	45999
	Galaxy M10 SM-M105F 2/16GB Black (SM-M105...	42021
	Mix 2 M2 4/64GB Black	92999
	Pixel 32GB (Quite Black)	74990
	RAZR 2019 XT2000-2 Noir Black	57900
	ROG Phone 3 ZS661KS 12/512GB Black	67900
	Redmi Note 9T 4/128GB Nightfall Black	44990
	iPhone 7 Plus 32GB Black (MNQM2)	88719

Result 4 x Read Only

17. Brand name, Operating systems and model name of one of the dataset were inserted into another dataset with similar attributes whose price is greater than 10000

use dmdd;

INSERT INTO ndtv_data_final(Brand,Model,Operating_system)

SELECT brand_name, model_name,os FROM output

WHERE highest_price>10000;

SELECT * FROM dmdd.ndtv_data_final;

SQL File 18* SQL File 19* SQL File 20* SQL File 21* SQL File 22* SQL File 23* x SQL File 24* SQL File 25* SQL File 26*

Limit to 2000 rows

```

1 • use dmdd;
2 • INSERT INTO ndtv_data_final(Brand,Model,Operating_system)
3 • SELECT brand_name, model_name,os FROM output
4 • WHERE highest_price>10000;
5 • SELECT * FROM dmdd.ndtv_data_final;

```

Result Grid Filter Rows: Export: Wrap Cell Content: A

	ID	Name	Brand	Model	Battery_capacity	Screen_size	Touchscreen	Resolution_x	Resolution_y	Process
	1353	Intex Aqua Y2 Ultra	Intex	Aqua Y2 Ultra	1400	4	Yes	480	800	4
	1354	Intex Aqua A2	Intex	Aqua A2	1500	4	Yes	480	800	4
	1355	Videocon Infinium Z51 No...	Videocon	Infinium Z51 Nova+	2000	5	Yes	480	854	4
	1356	Intex Aqua Y4	Intex	Aqua Y4	1700	4.5	Yes	480	854	2
	1357	iBall Andi4 B20	iBall	Andi4 B20	1250	4	Yes	480	800	1
	1358	iBall Andi Avonte 5	iBall	Andi Avonte 5	2150	5	Yes	480	854	4
	NULL	NULL	Honor	10 6/64GB Black	NULL	NULL	NULL	NULL	NULL	NULL
	NULL	NULL	OnePlus	6T 8/256GB Midni...	NULL	NULL	NULL	NULL	NULL	NULL
	NULL	NULL	OnePlus	7 8/256GB Mirror ...	NULL	NULL	NULL	NULL	NULL	NULL
	NULL	NULL	OnePlus	8 12/256GB Onyx...	NULL	NULL	NULL	NULL	NULL	NULL
	NULL	NULL	OnePlus	8 8/128GB Onyx...	NULL	NULL	NULL	NULL	NULL	NULL
	NULL	NULL	OnePlus	8 Pro 12/256GB G...	NULL	NULL	NULL	NULL	NULL	NULL
	NULL	NULL	OnePlus	8 Pro 8/128GB On...	NULL	NULL	NULL	NULL	NULL	NULL
	NULL	NULL	Nokia	8.3 5G 8/128GB P...	NULL	NULL	NULL	NULL	NULL	NULL
	NULL	NULL	OnePlus	8T 12/256GB Cyb...	NULL	NULL	NULL	NULL	NULL	NULL
	NULL	NULL	OnePlus	8T 12/256GB Lun...	NULL	NULL	NULL	NULL	NULL	NULL
	NULL	NULL	OnePlus	8T 8/128GB Aqua...	NULL	NULL	NULL	NULL	NULL	NULL
	NULL	NULL	Ulefone	Armor 10 5G 8/12...	NULL	NULL	NULL	NULL	NULL	NULL
	NULL	NULL	Ulefone	Armor 9 8/128GB ...	NULL	NULL	NULL	NULL	NULL	NULL
	NULL	NULL	Ulefone	Armor 9E 8/128G...	NULL	NULL	NULL	NULL	NULL	NULL
	NULL	NULL	Ulefone	Armor 9F 8/128G...	NULL	NULL	NULL	NULL	NULL	NULL

ndtv_data_final 1 x Read Only

18. Brand name, Operating systems and model name of one of the dataset were inserted into another dataset with similar attributes whose brand name is OnePlus

```
use dmdd;
```

```
INSERT INTO output(brand_name,model_name,screen_size,best_price)
```

```
SELECT Brand, Model, Screen_size, Price FROM ndtv_data_final
```

```
WHERE Brand like 'OnePlus';
```

```
SELECT * FROM output;
```

SQL File 18* SQL File 19* SQL File 20* SQL File 21* SQL File 22* SQL File 23* SQL File 24* x SQL File 25* SQL File 26*

Limit to 2000 rows

```

1 • use dmdd;
2 • INSERT INTO output(brand_name,model_name,screen_size,best_price)
3   SELECT Brand, Model, Screen_size, Price FROM ndtv_data_final
4   WHERE Brand like 'OnePlus';
5 • SELECT * FROM output;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: 1

	Id	brand_name	model_name	os	popularity	best_price	lowest_price	highest_price	sellers_amount
	1212	Apple	iPhone XR Dual Sim 128GB Black (MT192)	iOS	765	22520	18399	24855	15
	1213	Apple	iPhone XR Dual Sim 64GB Black (MT122)	iOS	695	20143	16862	22849	26
	1214	Apple	iPhone XS 256GB Gold (MT9K2)	iOS	1126	21463	15281	25617	68
	1215	Apple	iPhone XS 512GB Silver (MT9M2)	iOS	803	22842	17832	27500	62
	1216	Apple	iPhone XS 64GB Space Gray (MT9E2)	iOS	1187	19790	12505	23928	53
	1217	Apple	iPhone XS Max 256GB Gold (MT552)	iOS	1128	24184	18399	30600	37
	1218	Apple	iPhone XS Max 512GB Space Gray (MT622)	iOS	842	27190	21150	30200	47
	1219	Apple	iPhone XS Max 64GB Gold (MT522)	iOS	1101	22685	16018	27900	61
	1220	Apple	iPhone XS Max Dual Sim 64GB Gold (MT732)	iOS	530	24600	21939	33720	28
	1221	HUAWEI	nova 5T 6/128GB Black (51094MEU)	Android	1174	8804	7999	9999	18
	1222	ZTE	nubia Red Magic 5G 8/128GB Black	Android	752	18755	18500	19010	2
		OnePlus	7T			34930			
		OnePlus	7T Pro			52990			
		OnePlus	7 Pro			39995			
		OnePlus	6T			31999			
		OnePlus	6			34999			
		OnePlus	3T			19999			
		OnePlus	3			22999			
		OnePlus	7			29999			
		OnePlus	5T			28499			
		OnePlus	5			17499			

output 1 x Read Only

19. Which of the following mobile phones have following screen size, where the quantity is above 10?

```

use dmdd;
SELECT Screen_size, count(Id) AS No_of_Phones
FROM ndtv_data_final
GROUP BY Screen_size
HAVING count(Id) > 10;

```

SQL File 18* SQL File 19* SQL File 20* SQL File 21* SQL File 22* SQL File 23* SQL File 24* SQL File 25* x SQL File 26*

Limit to 2000 rows

```

1 • use dmdd;
2 • SELECT Screen_size, count(Id) AS No_of_Phones
3 FROM ndtv_data_final
4 GROUP BY Screen_size
5 HAVING count(Id) > 10;
6
7

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: I A

Screen_size	No_of_Phones
6.4	21
6.3	28
5.5	251
6	53
5.99	14
6.2	23
4.7	37
5.7	51
5.2	82
5	404
4	71
4.5	80
6.22	15
5.45	25

Result 21 x Read Only

Result Grid
Form Editor
Field Types
Query Stats
Execution Plan

20. Self joined brands from both the tables

```

use dmdd;
SELECT ndtv_data_final.Brand, output.brand_name
FROM ndtv_data_final, output
WHERE ndtv_data_final.Id <> output.Id

```

[illegible]

```
In [6]: import pandas as pd
import numpy as np
import seaborn as sns

In [7]: df = pd.read_csv("phones_data.csv")

In [8]: df.isnull().sum()

Out[8]: Unnamed: 0      0
brand_name      0
model_name      0
os             197
popularity      0
best_price      0
lowest_price    260
highest_price   260
sellers_amount  0
screen_size     2
memory_size    112
battery_size    10
release_date    0
dtype: int64

In [9]: df1 = df.dropna()

In [10]: df1
```

Out[10]:

	Unnamed: 0	brand_name	model_name	os	popularity	best_price	lowest_price	highest_price	sellers_amount	screen_size	memory_size	battery_size
0	0	ALCATEL	1 1/8GB Bluish Black (5033D-2JALUAA)	Android	422	1690.0	1529.0	1819.0	36	5.00	8.0	2000.0
1	1	ALCATEL	1 5033D 1/16GB Volcano Black (5033D-2LALUAF)	Android	323	1803.0	1659.0	2489.0	36	5.00	16.0	2000.0
2	2	ALCATEL	1 5033D 1/16GB Volcano Black (5033D-2LALUAF)	Android	299	1803.0	1659.0	2489.0	36	5.00	16.0	2000.0
3	3	ALCATEL	1 5033D 1/16GB Volcano Black (5033D-2LALUAF)	Android	287	1803.0	1659.0	2489.0	36	5.00	16.0	2000.0
5	5	Honor	10 6/64GB Black	Android	71	10865.0	10631.0	11099.0	2	5.80	64.0	3400.0
...
1218	1218	Apple	iPhone XS Max 512GB Space Gray (MT622)	iOS	842	27190.0	21150.0	30200.0	47	6.50	512.0	3174.0
1219	1219	Apple	iPhone XS Max 64GB Gold (MT522)	iOS	1101	22685.0	16018.0	27900.0	61	6.50	64.0	3174.0
1220	1220	Apple	iPhone XS Max Dual Sim 64GB Gold (MT732)	iOS	530	24600.0	21939.0	33720.0	28	6.50	64.0	3174.0
1221	1221	HUAWEI	nova 5T 6/128GB Black (51094MEU)	Android	1174	8804.0	7999.0	9999.0	18	6.26	128.0	3750.0
1222	1222	ZTE	nubia Red Magic 5G 8/128GB Black	Android	752	18755.0	18500.0	19010.0	2	6.65	128.0	4500.0

780 rows × 13 columns

In [11]:

df1.isnull().sum()

Out[11]:

Unnamed: 00
brand_name0
model_name0
os0
popularity0
best_price0
lowest_price0
highest_price0
sellers_amount0
screen_size0
memory_size0
battery_size0
release_date0
dtype: int64

In [12]:

df1.to_csv(r'C:\Users\Harsh\Desktop\Dmdd Ass3\output.csv', index=False, header=True)

In [13]:

data = pd.read_csv("ndtv_data_final.csv")

In [14]:

data.head()

Out[14]:

		Id	Name	Brand	Model	Battery_capacity	Screen_size	Touchscreen	Resolution_x	Resolution_y	Processor	...	Rear_camera	Front_camera	Operating
0	1	Realme X2 Pro	Realme	X2 Pro		4000	6.50	Yes	1080	2400	8	...	64.0	16.0	
1	2	iPhone 11 Pro Max	Apple	iPhone 11 Pro Max		3969	6.50	Yes	1242	2688	6	...	12.0	12.0	
2	3	iPhone 11	Apple	iPhone 11		3110	6.10	Yes	828	1792	6	...	12.0	12.0	
3	4	LG G8X ThinQ	LG	G8X ThinQ		4000	6.40	Yes	1080	2340	8	...	12.0	32.0	
4	5	OnePlus 7T	OnePlus	7T		3800	6.55	Yes	1080	2400	8	...	48.0	16.0	

5 rows × 22 columns

In [15]:

data.tail()

Out[15]:

		Id	Name	Brand	Model	Battery_capacity	Screen_size	Touchscreen	Resolution_x	Resolution_y	Processor	...	Rear_camera	Front_camera	C
1353	1354	Intex Aqua A2	Intex	Aqua A2		1500	4.0	Yes	480	800	4	...	5.0	0.3	
1354	1355	Videocon Infinium Z51 Nova+	Videocon	Infinium Z51 Nova+		2000	5.0	Yes	480	854	4	...	8.0	5.0	
1355	1356	Intex Aqua Y4	Intex	Aqua Y4		1700	4.5	Yes	480	854	2	...	5.0	2.0	
1356	1357	iBall Andi4 B20	iBall	Andi4 B20		1250	4.0	Yes	480	800	1	...	2.0	0.3	
1357	1358	iBall Andi Avonte 5	iBall	Andi Avonte 5		2150	5.0	Yes	480	854	4	...	8.0	0.0	

5 rows × 22 columns

```
In [16]: data.describe()
```

```
Out[16]:
```

	Id	Battery_capacity	Screen_size	Resolution_x	Resolution_y	Processor	RAM	Internal_storage	Rear_camera	Front_camera	Num
count	1358.000000	1358.000000	1358.000000	1358.000000	1358.000000	1358.000000	1358.000000	1358.000000	1358.000000	1358.000000	1358.0
mean	679.500000	2937.645066	5.290295	811.080265	1489.578056	5.549337	2481.773932	30.488925	12.043741	7.031370	1.8
std	392.165144	873.281081	0.670559	270.268687	556.228763	2.196365	1644.898180	36.453776	8.898299	6.293063	0.3
min	1.000000	1010.000000	2.400000	240.000000	320.000000	1.000000	64.000000	0.064000	0.000000	0.000000	1.0
25%	340.250000	2300.000000	5.000000	720.000000	1280.000000	4.000000	1000.000000	8.000000	8.000000	2.000000	2.0
50%	679.500000	3000.000000	5.200000	720.000000	1280.000000	4.000000	2000.000000	16.000000	12.200000	5.000000	2.0
75%	1018.750000	3500.000000	5.700000	1080.000000	1920.000000	8.000000	3000.000000	32.000000	13.000000	8.000000	2.0
max	1358.000000	6000.000000	7.300000	2160.000000	3840.000000	10.000000	12000.000000	512.000000	108.000000	48.000000	3.0

```
In [17]: data.nunique()
```

```
Out[17]: Id          1358
Name          1358
Brand          76
Model         1320
Battery_capacity 165
Screen_size    80
Touchscreen     2
Resolution_x    28
Resolution_y    53
Processor        6
RAM             13
Internal_storage 15
Rear_camera     32
Front_camera    30
Operating_system 7
Wi-Fi           2
Bluetooth        2
GPS              2
Number of SIMs   3
3G               2
4G               2
Price           626
dtype: int64
```

```
In [18]: data.isnull().sum()
```

```
Out[18]: Id          0
Name          0
Brand          0
Model          0
Battery_capacity 0
Screen_size    0
Touchscreen     0
Resolution_x    0
Resolution_y    0
Processor        0
RAM             0
Internal_storage 0
Rear_camera     0
Front_camera    0
Operating_system 0
Wi-Fi           0
Bluetooth        0
GPS              0
Number of SIMs   0
3G               0
4G               0
Price           0
dtype: int64
```

```
In [21]: mobile = data.drop(['Id', 'Resolution_x', 'Resolution_y', 'Front_camera', 'Rear_camera', 'Number of SIMs'], axis = 1)
```

```
In [22]: mobile.head()
```

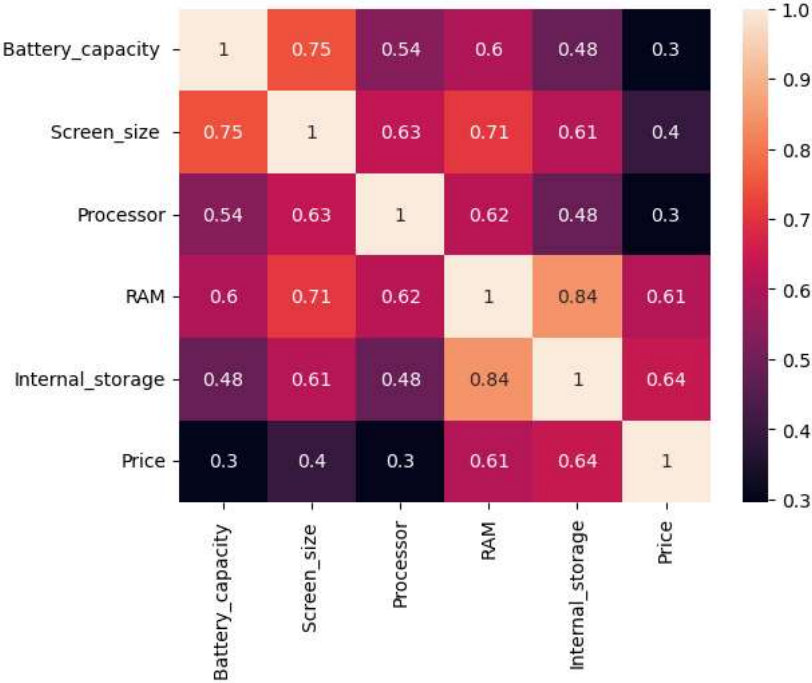
Out[22]:

	Name	Brand	Model	Battery_capacity	Screen_size	Touchscreen	Processor	RAM	Internal_storage	Operating_system	Wi-Fi	Bluetooth	GPS	3G	4G
0	Realme X2 Pro	Realme	X2 Pro	4000	6.50	Yes	8	6000	64.0	Android	Yes	Yes	Yes	Yes	Yes
1	iPhone 11 Pro Max	Apple	iPhone 11 Pro Max	3969	6.50	Yes	6	4000	64.0	iOS	Yes	Yes	Yes	Yes	Yes
2	iPhone 11	Apple	iPhone 11	3110	6.10	Yes	6	4000	64.0	iOS	Yes	Yes	Yes	Yes	Yes
3	LG G8X ThinQ	LG	G8X ThinQ	4000	6.40	Yes	8	6000	128.0	Android	Yes	Yes	Yes	No	No
4	OnePlus 7T	OnePlus	7T	3800	6.55	Yes	8	8000	128.0	Android	Yes	Yes	No	Yes	Yes

```
In [23]: corelation= mobile.corr()
```

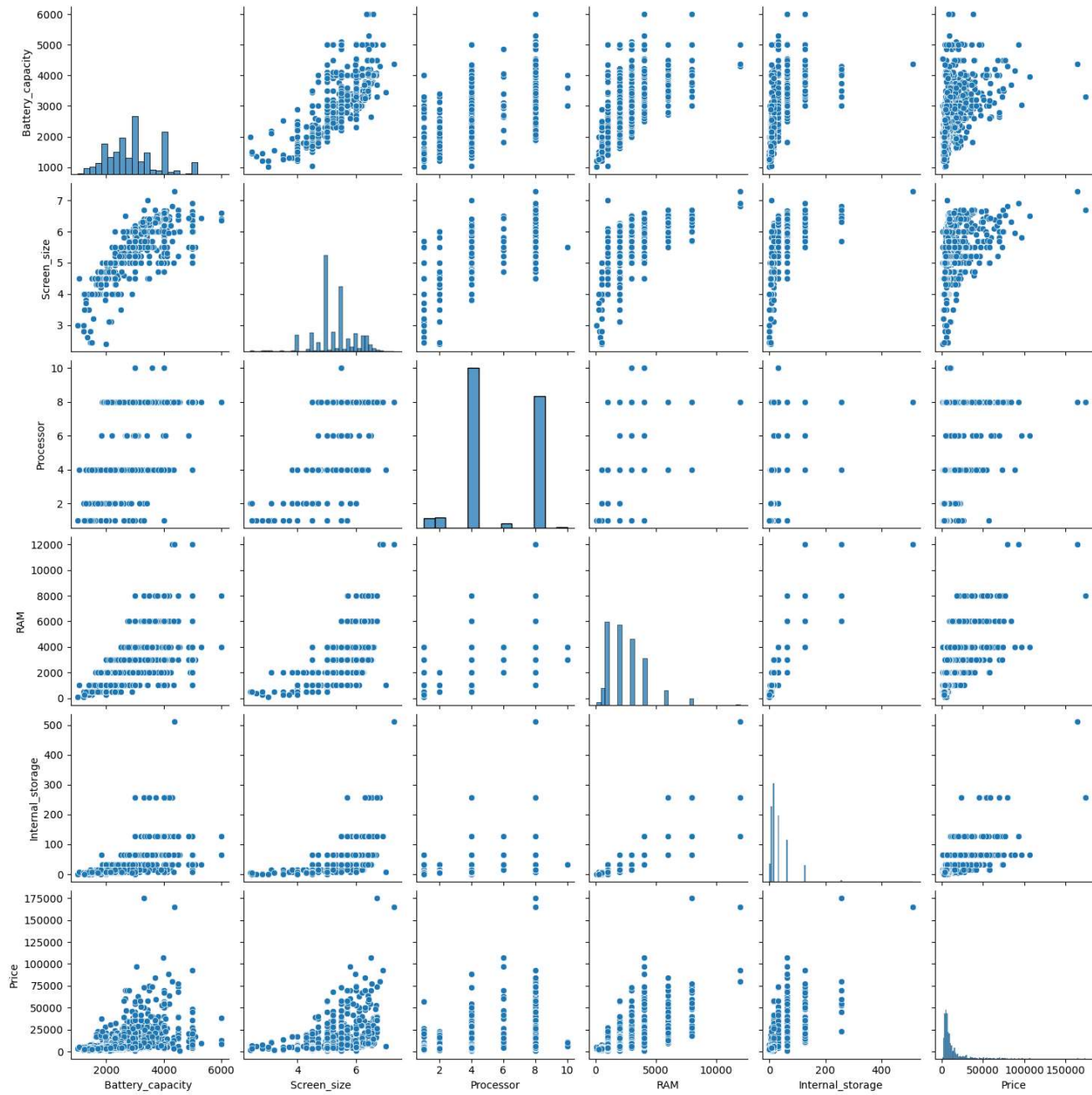
```
In [24]: sns.heatmap(corelation,xticklabels=corelation.columns, yticklabels=corelation.columns, annot=True)
```

Out[24]: <AxesSubplot:>



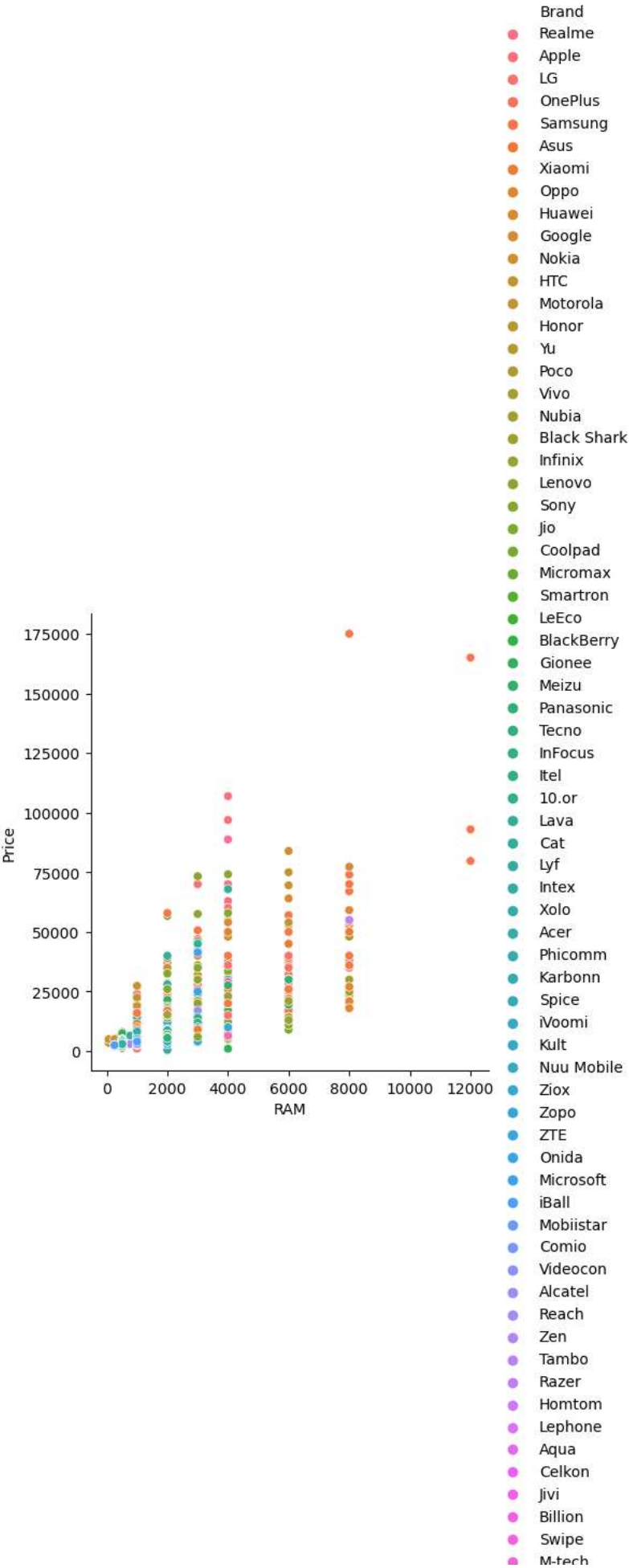

```
In [25]: sns.pairplot(mobile)
```

```
Out[25]: <seaborn.axisgrid.PairGrid at 0x22e77def190>
```



```
In [31]: sns.relplot(x= 'RAM', y='Price',hue='Brand',data=mobile)
```

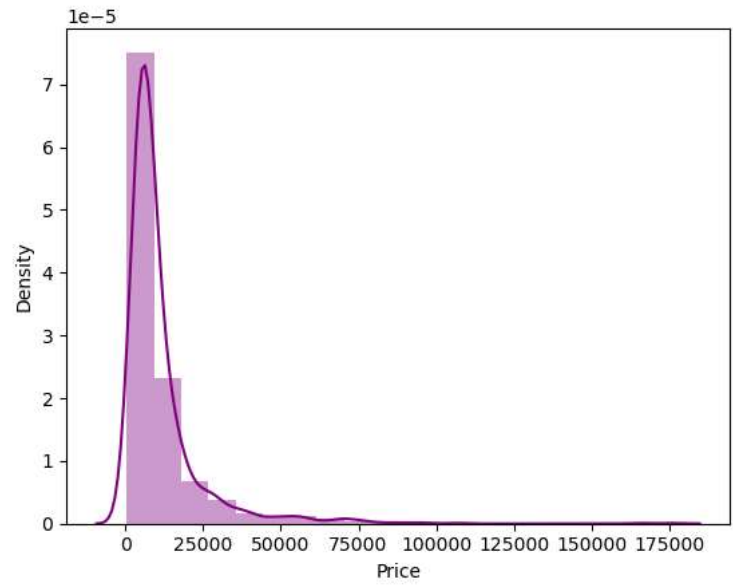
```
Out[31]: <seaborn.axisgrid.FacetGrid at 0x22e78c14dc0>
```

```
In [49]: sns.distplot(mobile['Price'],bins=20, color='purple')
```

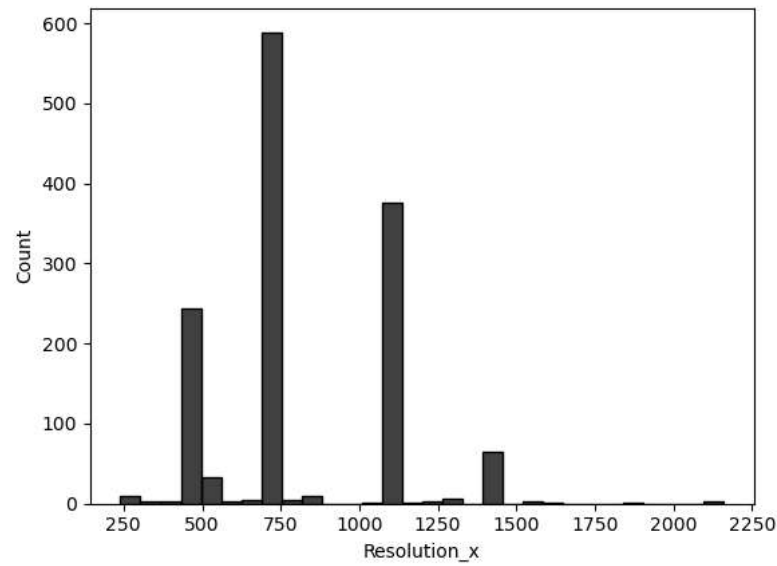
C:\Users\Harsh\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)

```
Out[49]: <AxesSubplot:xlabel='Price', ylabel='Density'>
```



```
In [52]: sns.histplot(data['Resolution_x'], bins=30, color='black')
```

```
Out[52]: <AxesSubplot:xlabel='Resolution_x', ylabel='Count'>
```



```
In [61]: sns.catplot(x='RAM', kind='box', data=mobile)
```

```
Out[61]: <seaborn.axisgrid.FacetGrid at 0x22e0210e5e0>
```

