**FROM**

**JOIN**

**WHERE**

**GROUP BY**

**HAVING**

**SELECT**

**DISTINCT**

**ORDER BY**

Sorting Data

1. Find top5 samsung phones with biggest screen size

SELECT model, screen\_size FROM campusx.smartphones

WHERE brand\_name = 'samsung'

ORDER BY screen\_size DESC LIMIT 5

1. Sort all the phone in descending order of number of total cameras

SELECT model, num\_rear\_cameras + num\_front\_cameras AS 'total\_cameras'

FROM campusx.smartphones

ORDER BY total\_cameras DESC

1. Sort data on the basis of ppi in decreasing order

SELECT model,

ROUND(SQRT(resolution\_width \* resolution\_width + resolution\_height \* resolution\_height)/screen\_size) AS 'ppi\_values'

FROM campusx.smartphones

ORDER BY ppi\_values DESC

1. Find the phone with 2nd largest battery

SELECT model, battery\_capacity

FROM campusx.smartphones

ORDER BY battery\_capacity DESC LIMIT 1, 1

Syntax – LIMIT x, y

This means we are telling start from index x = 1 and print y = 1 value

If we give LIMIT 3, 4

Then we mean to start from index 3 and print 4 elements

1. Find the name and rating of the worst rated apple phone

SELECT model, rating/10 AS 'Rating'

FROM campusx.smartphones

WHERE brand\_name = 'apple'

ORDER BY Rating ASC LIMIT 1

1. Sort phones alphabetically and then on the basis of rating in desc order

SELECT \* FROM campusx.smartphones

ORDER BY brand\_name ASC, rating DESC

1. Sort phones alphabetically and then on the basis of price in asc order

SELECT \*

FROM campusx.smartphones

ORDER BY brand\_name ASC, price ASC

1. Find the phone name, price of the costliest phone

SELECT model, price FROM campusx.smartphones

ORDER BY price DESC LIMIT 1

--- ORDER BY animation

<https://infytq.onwingspan.com/web/en/viewer/web-module/lex_auth_0126074240464732165_shared>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Grouping Data

1. Group smartphones by brand and get the count, average price, max rating, avg screen size and avg batter capacity

SELECT brand\_name, COUNT(\*) AS 'num\_phones',

ROUND(AVG(price)) AS 'avg\_price',

MAX(rating) AS 'max\_rating',

ROUND(AVG(screen\_size), 2) AS 'avg\_screen\_size',

ROUND(AVG(battery\_capacity)) AS 'avg\_battery\_capacity'

FROM campusx.smartphones

GROUP BY brand\_name

ORDER BY num\_phones DESC LIMIT 15

--- Group By Animation

<https://infytq.onwingspan.com/web/en/viewer/web-module/lex_auth_0126074240464732165_shared>

1. Group smartphones by whether they have an NFC and get the average price and rating

SELECT has\_nfc,

AVG(price) AS 'avg\_price',

ROUND(AVG(rating), 2) AS 'rating'

FROM campusx.smartphones

GROUP BY has\_nfc

* Fast charging available

SELECT fast\_charging\_available,

AVG(price) AS 'avg\_price',

AVG(rating) AS 'rating'

FROM campusx.smartphones

GROUP BY fast\_charging\_available

1. Group smartphones by the extended memory available and get the average price

SELECT extended\_memory\_available,

AVG(price) AS 'avg\_price',

AVG(rating) AS 'rating'

FROM campusx.smartphones

GROUP BY extended\_memory\_available

1. Group smartphones by the brand and processor brand and get the count of models and the average primary camera resolution (rear)

SELECT brand\_name,

processor\_brand,

COUNT(\*) AS 'num phones',

ROUND(AVG(primary\_camera\_rear)) AS 'avg camera resolution'

FROM campusx.smartphones

GROUP BY brand\_name, processor\_brand

1. Find top 5 most costly phone brands

SELECT brand\_name, ROUND(AVG(price)) AS 'avg\_price'

FROM campusx.smartphones

GROUP BY brand\_name

ORDER BY avg\_price DESC LIMIT 5

1. Which brand makes the smallest screen smartphones

SELECT brand\_name, ROUND(AVG(screen\_size)) AS 'avg\_screen\_size'

FROM campusx.smartphones

GROUP BY brand\_name

ORDER BY avg\_screen\_size ASC LIMIT 5

1. Avg price of 5g phones Vs avg price of non 5g phones

SELECT has\_5g,

AVG(price) AS 'avg\_price',

ROUND(AVG(rating), 2) AS 'rating'

FROM campusx.smartphones

GROUP BY has\_5g

1. Group smartphones by the brand, and find the brand with the highest number of models that have both NFC and an IR blaster

SELECT brand\_name, COUNT(\*) AS 'count'

FROM campusx.smartphones

WHERE has\_nfc = 'True' AND has\_ir\_blaster = 'True'

GROUP BY brand\_name

ORDER BY count DESC LIMIT 1

1. Find all Samsung 5g enabled smartphones and find out the avg price for NFC and Non-NFC phones

SELECT has\_5g, AVG(price) AS 'avg\_price'

FROM campusx.smartphones

WHERE brand\_name = 'samsung'

GROUP BY has\_5g

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Having Clause**

Just like WHERE is for SELECT, HAVING is for GROUP BY

SELECT 🡪 WHERE

GROUP BY 🡪 HAVING

We use HAVING when we must apply filtering on aggregate

We use WHERE when we must apply filtering on rows

* Finding the average price of phones, which are more than 20 in count

SELECT brand\_name, COUNT(\*) AS 'count', AVG(price) AS 'avg\_price'

FROM campusx.smartphones

GROUP BY brand\_name

HAVING count > 20

ORDER BY count DESC;

1. Find the avg rating of smartphone brands which have more than 20 phones

SELECT brand\_name, COUNT(\*) AS 'count', ROUND(AVG(rating), 2) AS 'avg\_rating'

FROM campusx.smartphones

GROUP BY brand\_name

HAVING count > 20

ORDER BY avg\_rating DESC

1. Find the top 3 brands with the highest avg ram that have a refresh rate of at least 90 Hz and fast charging available and don’t consider brands which have less than 10 phones

SELECT brand\_name, ROUND(AVG(ram\_capacity), 2) AS 'avg\_ram' FROM campusx.smartphones

WHERE refresh\_rate > 90 AND fast\_charging\_available = 1

GROUP BY brand\_name

HAVING COUNT(\*) > 10

ORDER BY avg\_ram DESC LIMIT 3

1. Find the avg price of all the phone brands with avg rating > 70 and num\_phones more than 10 among all 5g enables phones

Avg rating > 70 means we must use HAVING clause

Num\_phones more than 10 ==🡺 COUNT

All 5g enabled =🡺 WHERE

SELECT brand\_name, AVG(price) AS 'avg\_price'

FROM campusx.smartphones

WHERE has\_5g = 'True'

GROUP BY brand\_name

HAVING AVG(rating) > 70 AND COUNT(\*) > 10

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Practice

* Find the top 5 batsman in IPL

SELECT batter, SUM(batsman\_run) AS 'runs'

FROM campusx.ipl

GROUP BY batter

ORDER BY runs DESC LIMIT 10

* Find the 2nd Highest 6 hitter in IPL

SELECT batter, COUNT(\*) AS 'num\_sixes'

FROM campusx.ipl

WHERE batsman\_run = 6

GROUP BY batter

ORDER BY num\_sixes DESC LIMIT 1, 1

* Find Virat Kohli’s performance against all IPL teams (information not available)
* Find all batsmen with centuries in IPL

SELECT batter, ID, SUM(batsman\_run) AS 'score'

FROM campusx.ipl

GROUP BY batter, ID

HAVING score >= 100

ORDER BY batter DESC

* Find the top 5 batsman with highest strike rate who played a min of 1000 balls

SELECT batter, SUM(batsman\_run), COUNT(batsman\_run),

ROUND((SUM(batsman\_run)/COUNT(batsman\_run)) \* 100, 2) AS 'strike\_rate'

FROM campusx.ipl

GROUP BY batter

HAVING COUNT(batsman\_run) > 1000

ORDER BY strike\_rate DESC LIMIT 5