```
USE sql_cx_live;

SELECT * FROM laptops;

– Head, Tail, and Sample
SELECT * FROM laptops
ORDER BY `index` LIMIT 5;

SELECT * FROM laptops
ORDER BY `index` DESC LIMIT 5;

SELECT * FROM laptops
ORDER BY rand() LIMIT 5;

– UNIVARIATE ANALYSIS
– In Price Column - [Count, min, max, std, q1, q2, q3]
SELECT COUNT(Price) OVER(),
MIN(Price) OVER(),
MAX(Price) OVER(),
AVG(Price) OVER(),
STD(Price) OVER(),
PERCENTILE_CONT(0.25) WITHIN GROUP(ORDER BY Price) OVER()
AS 'Q1',
PERCENTILE_CONT(0.5) WITHIN GROUP(ORDER BY Price) OVER() AS
'Median',
PERCENTILE_CONT(0.75) WITHIN GROUP(ORDER BY Price) OVER()
AS 'Q3'
FROM laptops
ORDER BY `index` LIMIT 1;

– Missing Values
SELECT COUNT(Price)
FROM laptops
WHERE Price IS NULL;
```

– OUTLIERS
– There are various methods to detect outliers
– If it is Normal Distributed if it’s away from 3-Std then its an outlier
– OR
– We can use Box Plot.

```
SELECT * FROM (SELECT *,
PERCENTILE_CONT(0.25) WITHIN GROUP(ORDER BY Price) OVER()
AS 'Q1',
PERCENTILE_CONT(0.75) WITHIN GROUP(ORDER BY Price) OVER()
AS 'Q3'
FROM laptops) t
WHERE t.Price < t.Q1 - (1.5*(t.Q3 - t.Q1)) OR
t.Price > t.Q3 + (1.5*(t.Q3 - t.Q1));
```

– HISTOGRAM
– CREATING BUCKETS
– DRAWING CONCLUSIONS that which price segment has the most number of laptops and least number of laptops

```
SELECT t.buckets,REPEAT('*',COUNT(*)/5) FROM (SELECT price,
CASE
        WHEN price BETWEEN 0 AND 25000 THEN '0-25K'
    WHEN price BETWEEN 25001 AND 50000 THEN '25K-50K'
    WHEN price BETWEEN 50001 AND 75000 THEN '50K-75K'
    WHEN price BETWEEN 75001 AND 100000 THEN '75K-100K'
        ELSE '>100K'
END AS 'buckets'
FROM laptops) t
GROUP BY t.buckets;
```

– VALUE COUNTS
– Which Company has the most number of laptops or creating a pie chart to understand number of laptops produced by each company

```
SELECT Company,COUNT(Company) FROM laptops
```

```sql
– What is the percentage of touchscreen laptops
– Which company made the most number of touchscreen laptops
– Pie chart of CPU Brand
– Pie Chart of OS
GROUP BY Company;

– Bivariate Analysis
– Making Scatter Plot between 2 numerical columns
SELECT cpu_speed,Price FROM laptops;

SELECT * FROM laptops;

– Bivariate Analysis
– Using 2-Categorical Columns - CROSSTAB
SELECT Company,
SUM(CASE WHEN Touchscreen = 1 THEN 1 ELSE 0 END) AS 'Touchscreen_yes',
SUM(CASE WHEN Touchscreen = 0 THEN 1 ELSE 0 END) AS 'Touchscreen_no'
FROM laptops
GROUP BY Company;

SELECT DISTINCT cpu_brand FROM laptops;

SELECT Company,
SUM(CASE WHEN cpu_brand = 'Intel' THEN 1 ELSE 0 END) AS 'intel',
SUM(CASE WHEN cpu_brand = 'AMD' THEN 1 ELSE 0 END) AS 'amd',
SUM(CASE WHEN cpu_brand = 'Samsung' THEN 1 ELSE 0 END) AS 'samsung'
FROM laptops
GROUP BY Company;

– Categorical Numerical Bivariate analysis

SELECT Company,MIN(price),
MAX(price),AVG(price),STD(price)
FROM laptops
GROUP BY Company;
-- Dealing with missing values
SELECT * FROM laptops
WHERE price IS NULL;
-- UPDATE laptops
-- SET price = NULL
-- WHERE `index` IN (7,869,1148,827,865,821,1056,1043,692,1114)
```

– replace missing values with mean of price

```
UPDATE laptops
SET price = (SELECT AVG(price) FROM laptops)
WHERE price IS NULL;
```

– replace missing values with mean price of corresponding company

```
UPDATE laptops l1
SET price = (SELECT AVG(price) FROM laptops l2 WHERE
                    l2.Company = l1.Company)
WHERE price IS NULL;

SELECT * FROM laptops
WHERE price IS NULL;
-- corresponsing company + processor
SELECT * FROM laptops;
```

– Adding suitable columns that would be beneficial for the analysis
OR
– Replacing columns that are not beneficial for the analysis with new columns
-- Feature Engineering

- **Adding the column PPI with columns (resolution_width, resolution_height, Inches)**

```
ALTER TABLE laptops ADD COLUMN ppi INTEGER;

UPDATE laptops
SET ppi = ROUND(SQRT(resolution_width*resolution_width +
resolution_height*resolution_height)/Inches);

SELECT * FROM laptops
ORDER BY ppi DESC;
```

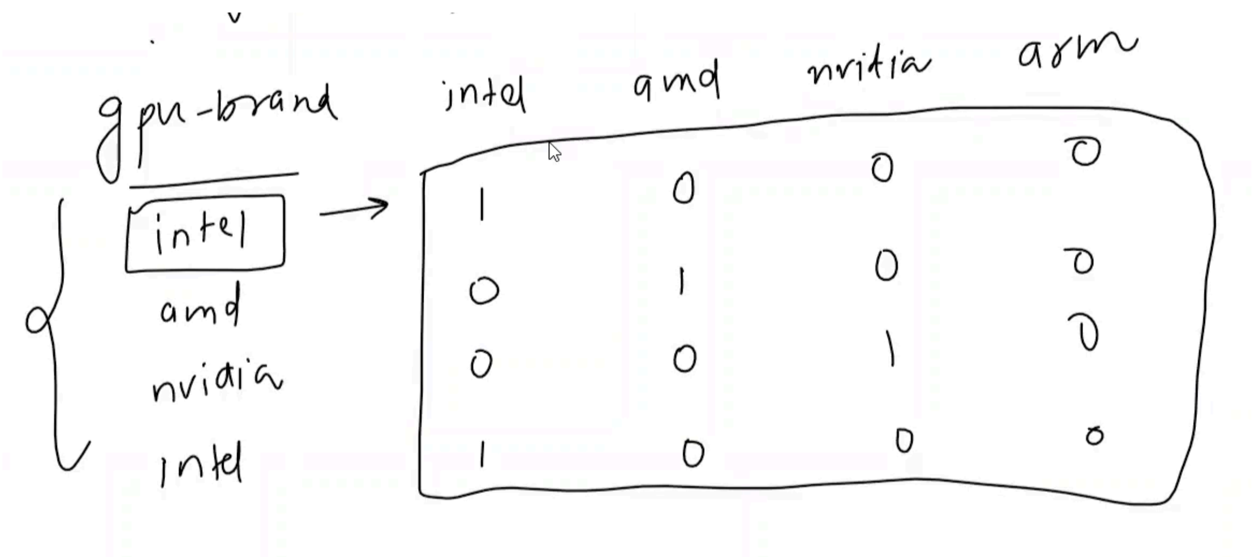
- **Adding column screen size with column (Inches and dividing them into three categories Small screen laptops, Medium Screen, and Large Screen)**

```
ALTER TABLE laptops ADD COLUMN screen_size VARCHAR(255)
AFTER Inches;

UPDATE laptops
SET screen_size =
CASE
        WHEN Inches < 14.0 THEN 'small'
    WHEN Inches >= 14.0 AND Inches < 17.0 THEN 'medium'
        ELSE 'large'
END;

SELECT screen_size,AVG(price) FROM laptops
GROUP BY screen_size;
```

-- When we need to convert a categorical column into a Numerical column we need to do this. As it makes the analysis more efficient.
– One Hot Encoding



gpu-brand	intel	amd	nvidia	arm
intel	1	0	0	0
amd	0	1	0	0
nvidia	0	0	1	0
intel	1	0	0	0

- **Instead of GPU_BRAND we're making 4 new columns that are (INTEL, AMD, NVIDIA, and ARM) and whenever a laptop's GPU is one of the 4 GPUs we mark it as '1' and the remaining 0.**

```
SELECT gpu_brand,
CASE WHEN gpu_brand = 'Intel' THEN 1 ELSE 0 END AS 'intel',
CASE WHEN gpu_brand = 'AMD' THEN 1 ELSE 0 END AS 'amd',
CASE WHEN gpu_brand = 'nvidia' THEN 1 ELSE 0 END AS 'nvidia',
CASE WHEN gpu_brand = 'arm' THEN 1 ELSE 0 END AS 'arm'
FROM laptops
```