```
#########EDA#####################
USE laptop;

SELECT * FROM laptop;

-- head, tail and sample
SELECT * FROM laptop
ORDER BY `index` LIMIT 5;

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

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

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 laptop
ORDER BY `index` LIMIT 1;

-- missing value
SELECT COUNT(Price)
FROM laptop
WHERE Price IS NULL;

-- outliers
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));

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 laptop) t
```

```
GROUP BY t.buckets;

########vertical####################
##

SELECT Company,COUNT(Company) FROM laptop
GROUP BY Company;

SELECT cpu_speed,Price FROM laptops;

SELECT * FROM laptops;

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;
-- Feature Engineering
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;

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;

-- One Hot Encoding

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
```