**Case study – banking market analysis**

Case study – banking market analysis

1. Load data and create a Hive table

* CREATE TABLE IF NOT EXISTS bankings

(

age int, job String, marital String, education String, default String, balance int, housing String, loan String, contact String, day int, month String, duration int, campaign int, pdays int, previous int, poutcome String, y String

)

COMMENT 'bankings details'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ';'

LINES TERMINATED BY '\n'

STORED AS TEXTFILE;

* LOAD DATA LOCAL INPATH '/mnt/c/MKM/SQL/SARVANA/futurense-dataengg-bootcamp/assignments/datasets/bankmarketdata.csv' OVERWRITE INTO TABLE bankings;

2. Give marketing success rate. (No. of people subscribed / total no. of entries)

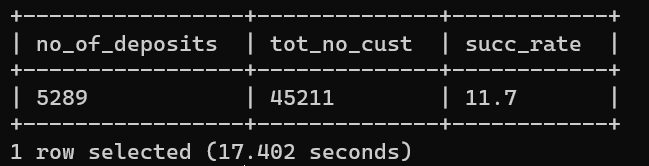
Select COUNT(IF(y = 'yes', 1, NULL)) as no\_of\_deposits,

count(\*) as tot\_no\_cust,

round( (COUNT(IF(y = 'yes', 1, NULL)) / count(\*)) \* 100, 2) as succ\_rate

From bankings

;



3. Give marketing failure rate

Select COUNT(IF(y = 'yes', 1, NULL)) as no\_of\_deposits,

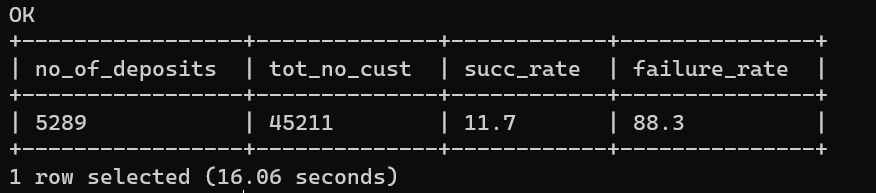
count(\*) as tot\_no\_cust,

round( (COUNT(IF(y = 'yes', 1, NULL)) / count(\*)) \* 100, 2) as succ\_rate,

100 - round( (COUNT(IF(y = 'yes', 1, NULL)) / count(\*)) \* 100, 2) as failure\_rate

From bankings

;



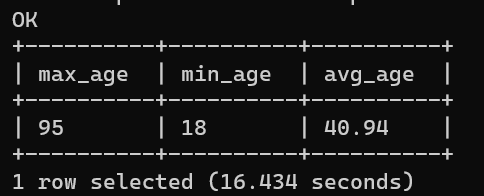
4. Maximum, Mean, and Minimum age of the average targeted customer

select max(age) as max\_age, min(age) as min\_age,

round( avg(age), 2) as avg\_age

from bankings

;



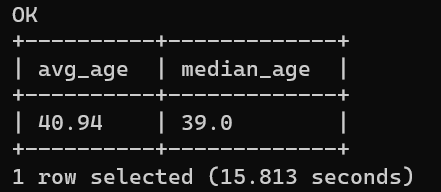
5. Check the quality of customers by checking the average balance, median balance of customers

Select round( avg(age), 2) as avg\_age,

round( percentile(age, 0.5), 2) as median\_age

From bankings

;



6. Check if age matters in marketing subscription for deposit

Select age, COUNT(IF(y = 'yes', 1, NULL)) as no\_of\_deposits,

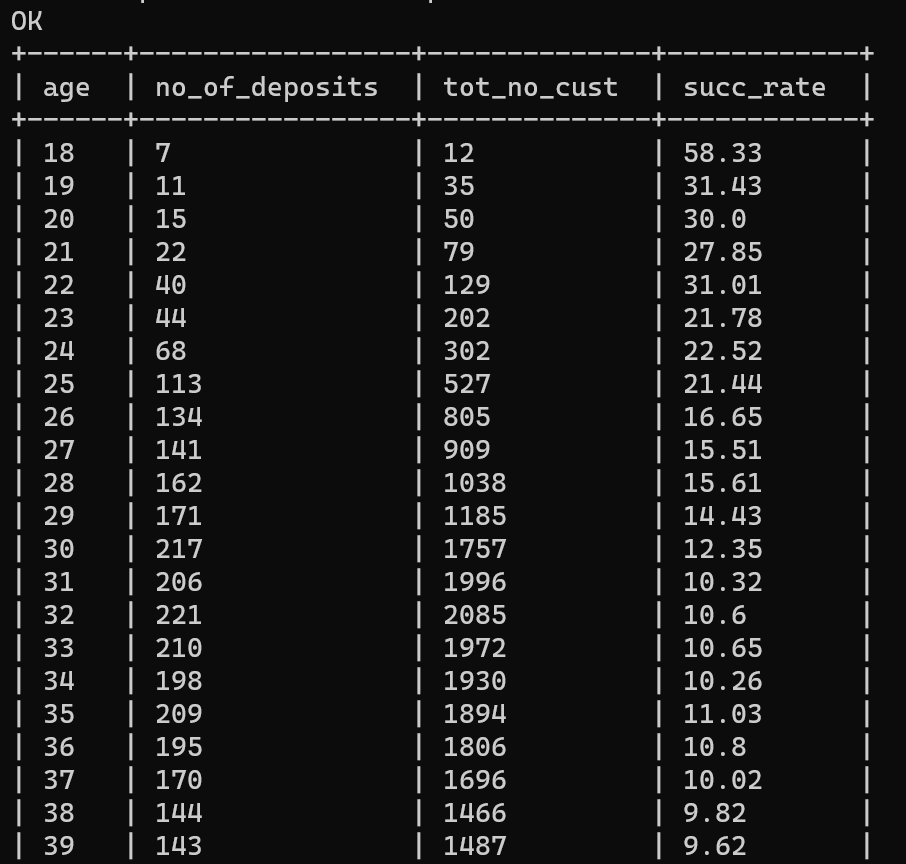
count(\*) as tot\_no\_cust,

round( (COUNT(IF(y = 'yes', 1, NULL)) / count(\*)) \* 100, 2) as succ\_rate

From bankings

group by age

;



select

case

when age < 20 then 'Under 20'

when age between 21 and 30 then '21-30'

when age between 31 and 40 then '31-40'

when age between 41 and 50 then '41-50'

when age between 51 and 60 then '51-60'

when age > 60 then 'Over 60'

END as age\_range,

COUNT(IF(y = 'yes', 1, NULL)) as no\_of\_deposits,

count(\*) as tot\_no\_cust,

round( (COUNT(IF(y = 'yes', 1, NULL)) / count(\*)) \* 100, 2) as succ\_rate

from bankings

group by case

when age < 20 then 'Under 20'

when age between 21 and 30 then '21-30'

when age between 31 and 40 then '31-40'

when age between 41 and 50 then '41-50'

when age between 51 and 60 then '51-60'

when age > 60 then 'Over 60'

END

;



7. Check if marital status mattered for subscription to deposit.

Select marital, COUNT(IF(y = 'yes', 1, NULL)) as no\_of\_deposits,

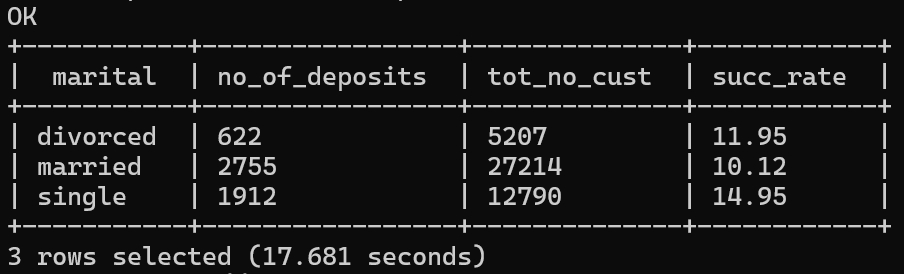
count(\*) as tot\_no\_cust,

round( (COUNT(IF(y = 'yes', 1, NULL)) / count(\*)) \* 100, 2) as succ\_rate

From bankings

group by marital

;



select

case

when age < 20 then 'Under 20'

when age between 21 and 30 then '21-30'

when age between 31 and 40 then '31-40'

when age between 41 and 50 then '41-50'

when age between 51 and 60 then '51-60'

when age > 60 then 'Over 60'

END as age\_range,marital,

COUNT(IF(y = 'yes', 1, NULL)) as no\_of\_deposits,

count(\*) as tot\_no\_cust,

round( (COUNT(IF(y = 'yes', 1, NULL)) / count(\*)) \* 100, 2) as succ\_rate

from bankings

group by case

when age < 20 then 'Under 20'

when age between 21 and 30 then '21-30'

when age between 31 and 40 then '31-40'

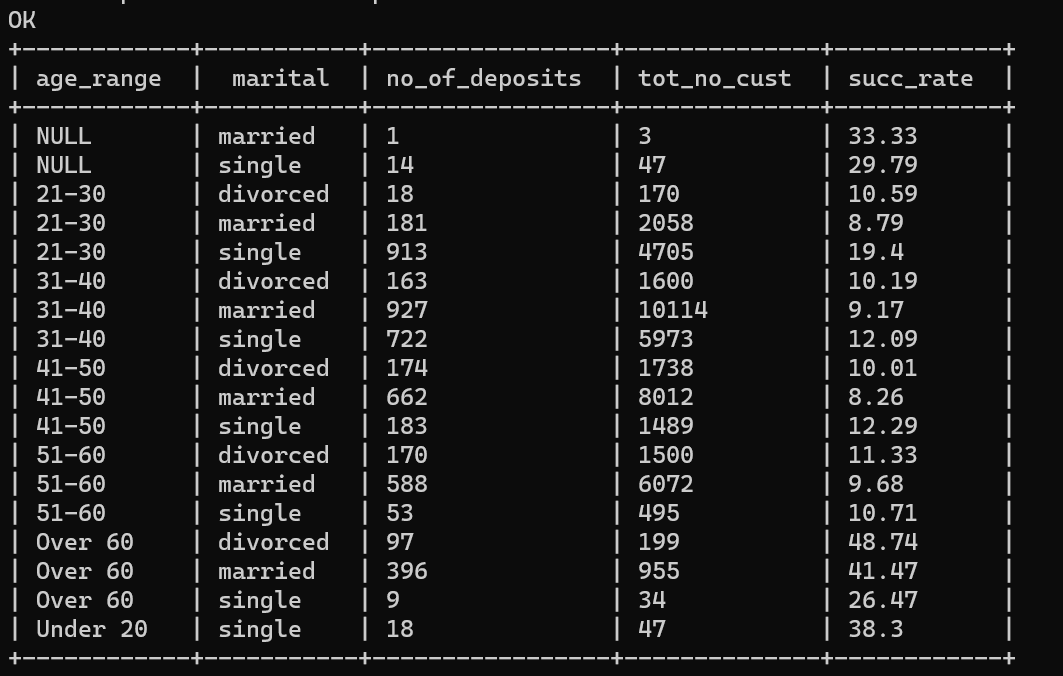
when age between 41 and 50 then '41-50'

when age between 51 and 60 then '51-60'

when age > 60 then 'Over 60'

END , marital

;



8. Check if age and marital status together mattered for subscription to deposit scheme

Select age, marital, COUNT(IF(y = 'yes', 1, NULL)) as no\_of\_deposits,

count(\*) as tot\_no\_cust,

round( (COUNT(IF(y = 'yes', 1, NULL)) / count(\*)) \* 100, 2) as succ\_rate

From bankings

group by age, marital

;

