1. \*\*\*\*\*\*\*Naming Table with respect to no of coloumns/attributes and Number of rows\*\*\*\*\*

Query:

SELECT

sch.name AS table\_name,

COUNT(pti.name) AS number\_of\_attributes,

FROM sqlite\_schema AS sch,

pragma\_table\_info(sch.name) AS pti

WHERE sch.type = 'table'

GROUP BY sch.name;

Output:

|  |  |
| --- | --- |
| table\_name | number\_of \_attributes |
| customers | 13 |
| employees | 8 |
| offices | 9 |
| orderdetails | 5 |
| orders | 7 |
| payments | 4 |
| productlines | 4 |
| products | 9 |

2. We'll need the following two tables to perform these calculations: namely . . .

low stock=SUM(quantityOrdered)/quantityInStock

. . . and . . .

product performance=SUM(quantityOrdered × priceEach)

Query:

SELECT

sum(orderdetails.quantityOrdered)/products.quantityInStock as low\_stock,

sum(orderdetails.quantityOrdered\*orderdetails.priceEach) as product\_performance

FROM orderdetails left join products on products.productCode=orderdetails.productCode

GROUP BY products.productCode

Output:

low\_stock product\_performance

0 90157.77

0 190017.96

0 109998.82

0 170686.0

0 127924.32

0 123123.01

13 161531.48

0 190755.86

0 119085.25

0 135767.03

0 132363.79

0 98718.76

0 152543.02

0 63489.95

0 109946.21

0 104210.62

0 105563.12

0 114648.12

0 102563.52

0 46078.29

0 101778.13

0 144959.91

0 140535.6

0 66455.62

0 119050.95

0 142530.63

1 45306.77

0 117669.66

0 109992.01

0 55835.3

0 68741.91

0 49992.72

1 132275.98

0 100770.12

0 97193.88

0 55172.21

0 74127.24

0 85328.57

0 111528.82

0 276839.98

0 82617.12

0 68783.93

0 87637.63

0 121890.6

0 121653.46

0 54024.87

0 102537.45

0 122254.75

0 71526.22

0 77800.54

0 114232.79

0 44037.84

0 130749.31

0 50101.57

0 53236.67

0 50255.45

0 105266.64

0 42015.54

0 94885.37

0 28052.94

67 67193.49

0 112427.12

0 38449.09

0 118774.33

0 57995.25

0 76670.02

0 31627.96

0 58434.07

0 94248.67

0 30972.87

0 77239.92

0 67357.3

0 52339.53

0 52803.75

0 87404.81

0 71208.18

0 134240.71

0 62269.67

0 29763.39

0 92973.4

0 88434.46

0 68276.35

0 69120.97

0 78067.82

5 89364.89

0 33268.76

0 46519.05

0 53791.99

1 57282.49

7 60493.33

0 84039.24

0 41599.24

0 101137.55

0 52123.81

1 73670.64

0 56455.11

0 73871.22

1 69531.61

0 73298.42

0 89347.8

0 66697.13

0 89272.65

0 102786.38

1 76618.4

0 84992.25

0 78919.06

0 71753.93

0 42692.53

2 47550.4

3. how should we match marketing and communication strategies to customer behaviors? This involves categorizing customers: finding the VIP (very important person) customers and those who are less engaged.

VIP customers bring in the most profit for the store.

Less-engaged customers bring in less profit.

Query:

SELECT o.customerNumber, SUM(quantityOrdered \* (priceEach - buyPrice)) AS profit

FROM products p

JOIN orderdetails od

ON p.productCode = od.productCode

JOIN orders o

ON o.orderNumber = od.orderNumber

GROUP BY o.customerNumber;

Output:

customerNumber profit

103 10063.8

112 31312.72

114 70311.07

119 60875.3

121 41391.52

124 236769.39

128 27728.34

129 28092.43

131 58669.1

141 326519.66

144 25138.38

145 50973.68

146 53211.19

148 60477.38

151 72370.09

157 40429.55

161 41510.76

166 41305.17

167 40592.06

171 27143.39

172 34650.17

173 13734.7

175 41313.51

177 24360.18

181 28279.5

186 38808.53

187 60095.86

189 19588.29

198 6586.02

201 43523.64

202 28882.99

204 24296.9

205 40397.88

209 31526.6

211 16868.68

216 24113.54

219 2610.87

227 34655.62

233 25856.86

239 31594.7

240 26808.79

242 23905.16

249 34100.63

250 28390.81

256 24610.8

259 36744.01

260 24118.93

276 54551.66

278 52309.63

282 52331.45

286 36337.69

298 43393.75

299 25665.76

311 38334.95

314 25861.96

319 31815.01

320 39136.14

321 55674.28

323 60013.99

324 31667.93

328 32240.57

333 22579.18

334 40095.85

339 20737.79

344 18953.3

347 14928.37

350 27184.74

353 52698.66

357 39321.91

362 32549.53

363 45090.39

379 28969.09

381 11693.99

382 54724.68

385 35987.68

386 48516.67

398 43487.88

406 38755.5

412 38399.18

415 13033.35

424 26001.55

447 18957.41

448 46777.54

450 55931.37

452 18358.11

455 29265.9

456 13066.02

458 49192.39

462 36886.05

471 22433.82

473 9532.93

475 17186.93

484 21225.65

486 33598.57

487 17230.12

489 10868.04

495 25244.69

496 51771.5

4. let's find the number of new customers arriving each month. That way we can check if it's worth spending money on acquiring new customers.

Query:

WITH

payment\_with\_year\_month\_table AS (

SELECT \*,

CAST(SUBSTR(paymentDate, 1,4) AS INTEGER)\*100 + CAST(SUBSTR(paymentDate, 6,7) AS INTEGER) AS year\_month

FROM payments p

),

customers\_by\_month\_table AS (

SELECT p1.year\_month, COUNT(\*) AS number\_of\_customers, SUM(p1.amount) AS total

FROM payment\_with\_year\_month\_table p1

GROUP BY p1.year\_month

),

new\_customers\_by\_month\_table AS (

SELECT p1.year\_month,

COUNT(\*) AS number\_of\_new\_customers,

SUM(p1.amount) AS new\_customer\_total,

(SELECT number\_of\_customers

FROM customers\_by\_month\_table c

WHERE c.year\_month = p1.year\_month) AS number\_of\_customers,

(SELECT total

FROM customers\_by\_month\_table c

WHERE c.year\_month = p1.year\_month) AS total

FROM payment\_with\_year\_month\_table p1

WHERE p1.customerNumber NOT IN (SELECT customerNumber

FROM payment\_with\_year\_month\_table p2

WHERE p2.year\_month < p1.year\_month)

GROUP BY p1.year\_month

)

SELECT year\_month,

ROUND(number\_of\_new\_customers\*100/number\_of\_customers,1) AS number\_of\_new\_customers\_props,

ROUND(new\_customer\_total\*100/total,1) AS new\_customers\_total\_props

FROM new\_customers\_by\_month\_table;

Output:

year\_month number\_of\_new\_customers\_props new\_customers\_total\_props

200301 100.0 100.0

200302 100.0 100.0

200303 100.0 100.0

200304 100.0 100.0

200305 100.0 100.0

200306 100.0 100.0

200307 75.0 68.3

200308 66.0 54.2

200309 80.0 95.9

200310 69.0 69.3

200311 57.0 53.9

200312 60.0 54.9

200401 33.0 41.1

200402 33.0 26.5

200403 54.0 55.0

200404 40.0 40.3

200405 12.0 17.3

200406 33.0 43.9

200407 10.0 6.5

200408 18.0 26.2

200409 40.0 56.4

Here are the answers to our questions.

Question 1: Which products should we order more of or less of?

Classic cars are the priority for restocking. They sell frequently, and they are the highest-performance products.

productName productLine

2002 Suzuki XREO Motorcycles

1976 Ford Gran Torino Classic Cars

1995 Honda Civic Classic Cars

1932 Model A Ford J-Coupe Vintage Cars

1965 Aston Martin DB5 Classic Cars

1999 Indy 500 Monte Carlo SS Classic Cars

1968 Dodge Charger Classic Cars

America West Airlines B757-200 Planes

2002 Chevy Corvette Classic Cars

1982 Ducati 996 R Motorcycles

Question 2: How should we match marketing and communication strategies to customer behaviors?

VIP customers

contactLastName contactFirstName city country profit

Freyre Diego Madrid Spain 326519.66

Nelson Susan San Rafael USA 236769.39

Young Jeff NYC USA 72370.09

Ferguson Peter Melbourne Australia 70311.07

Labrune Janine Nantes France 60875.30

Least engaged customers

contactLastName contactFirstName city country profit

Young Mary Glendale USA 2610.87

Taylor Leslie Brickhaven USA 6586.02

Ricotti Franco Milan Italy 9532.93

Schmitt Carine Nantes France 10063.80

Smith Thomas London UK 10868.04

Now that we have the most-important and least-committed customers, we can determine how to drive loyalty and attract more customers.

Question 3: How much can we spend on acquiring new customers?

ltv

39039.594388

LTV tells us how much profit an average customer generates during their lifetime with our store. We can use it to predict our future profit. So, if we get ten new customers next month, we'll earn 390,395 dollars, and we can decide based on this prediction how much we can spend on acquiring new customers.