TRAINITY DATA ANALYTICS PROJECT-3

**OPERATION ANALYTICS AND INVESTIGATING METRIC SPIKE USING ADVANCED SQL**

# PROJECT DESCRIPTION:

The aim of this project is to examine the patterns of multinational product users in terms of their interactions with fellow users and the frequency of their engagement with their accounts. This evaluation yields valuable and practical knowledge to the multinational product teams, which can be utilized to anticipate and project the future developments of the product. By utilizing diverse database management tools, we can extract informative insights from the raw data and even visualize them, thereby paving the way for optimizing the platform's efficiency.

# APPROACH:

The first step in commencing the project is to analyze the provided datasets and gain a clear understanding of the relationships between the columns in each dataset. After this, we will create a database and import the CSV files into the datasets. Finally, we will utilize SQL queries to extract and organize the data, enabling us to retrieve the necessary insights and information required to respond to inquiries from the management team and investors regarding the metrics of the multinational product.

# INSIGHTS:

My experience working on the project gave me valuable insights into how advanced SQL techniques can be used to effectively extract insights from the database. Through the use of advanced SQL, I was able to conduct operational analytics and investigate metric spikes, enabling me to identify trends and patterns in the data.

# TECH-STACK USED:

* Mysql workbench

# Case Study 1 (Job Data):

**1A)Calculate the number of jobs reviewed per hour per day for November 2020?**

# SQL QUERY:

SELECT ds AS Date ,ROUND(COUNT(job\_id)/SUM(time\_spent)\*3600) AS Total\_Job\_Reviews FROM job\_data GROUP BY 1;

# QUERY OUTPUT:

|  |  |
| --- | --- |
| **Date** | **Total\_Job\_Reviews** |
| 2020-11-30 | 180 |
| 2020-11-29 | 180 |
| 2020-11-28 | 218 |
| 2020-11-27 | 35 |
| 2020-11-26 | 64 |
| 2020-11-25 | 80 |

**1B)Calculate 7 day rolling average of throughput? For throughput, do you prefer daily metric or 7-day rolling and why?**

## SQL QUERY:

WITH grp AS (SELECT ds, COUNT(job\_id) AS num\_jobs, SUM(time\_spent) AS total\_time FROM job\_data GROUP BY ds) SELECT ds AS DATE , ROUND(1.0\*SUM(num\_jobs) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) / SUM(total\_time) OVER (ORDER BY ds ROWS BETWEEN 6 PRECEDING AND

CURRENT ROW),2) AS Rolling\_average\_for\_7days FROM grp;

# QUERY OUTPUT:

|  |  |
| --- | --- |
| **DATE** | **Rolling\_average\_for\_7days** |
| 2020-11-25 | 0.02 |
| 2020-11-26 | 0.02 |

|  |  |
| --- | --- |
| **DATE** | **Rolling\_average\_for\_7days** |
| 2020-11-25 | 0.02 |
| 2020-11-26 | 0.02 |
| 2020-11-27 | 0.01 |
| 2020-11-28 | 0.02 |
| 2020-11-29 | 0.02 |
| 2020-11-30 | 0.03 |

In MySQL, daily metrics and rolling average are both useful in analyzing system performance and making decisions about resource allocation and optimization. Daily metrics provide a snapshot of performance over a fixed time period, while rolling average offers a more flexible and up-to-date view of trends over a sliding time window. The preferred approach depends on the specific use case and the insights required from the data.

# 1c)Calculate the percentage share of each language in the last 30 days. SQL QUERY:

SELECT language,(100\*COUNT(language)/TOTAL) AS Percentage FROM job\_data CROSS JOIN (SELECT COUNT(\*) AS TOTAL FROM job\_data WHERE ds BETWEEN '2020-11-30' - INTERVAL 30 DAY AND '2020-11-30') as tb GROUP BY 1 ;

# QUERY OUTPUT:

|  |  |
| --- | --- |
| **Language** | **Percentage** |
| English | 12.5000 |
| Arabic | 12.5000 |
| Persian | 37.5000 |
| Hindi | 12.5000 |
| French | 12.5000 |
| Italian | 12.5000 |

**1D)Let’s say you see some duplicate rows in the data. How will you display duplicates from the table?**

# SQL QUERY:

WITH TAB AS (SELECT \*, ROW\_NUMBER() OVER (PARTITION BY

ds,job\_id,actor\_id,event,language,time\_spent,org) AS row\_num FROM job\_data )SELECT

\* FROM TAB WHERE row\_num>1;

## QUERY OUTPUT:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ds** | **Job\_id** | **actor\_id** | **event** | **language** | **time\_spent** | **org** | **row\_num** |

**There are no duplicate in this table**

# Case Study 2 (Investigating metric spike):

**2A)Calculate the weekly user engagement? SQL QUERY:**

SELECT WEEK(occurRed\_at) AS Week,COUNT(user\_id) AS No\_of\_users FROM events WHERE event\_type='engagement' GROUP BY 1 ORDER BY 1;

# QUERY OUTPUT:

|  |  |
| --- | --- |
| **Week** | **No\_of\_users** |
| 17 | 8019 |
| 18 | 17341 |
| 19 | 17224 |
| 20 | 17911 |
| 21 | 17151 |
| 22 | 18413 |
| 23 | 18280 |
| 24 | 19052 |
| 25 | 18642 |

|  |  |
| --- | --- |
| 26 | 19061 |
| 27 | 19881 |
| 28 | 20776 |
| 29 | 20067 |
| 30 | 21533 |
| 31 | 18556 |
| 32 | 16612 |
| 33 | 16145 |
| 34 | 16127 |
| 35 | 784 |

**2B)Calculate the user growth for product? SQL QUERY:**

WITH CTE AS (SELECT MONTH(created\_at) as MONTH,COUNT(user\_id) AS USERS FROM users GROUP BY 1)SELECT MONTH,USERS,ROUND((USERS/LAG(USERS,1) OVER (ORDER BY MONTH)-1)\*100,2) AS G Growth\_Percent FROM CTE;

# QUERY OUTPUT:

|  |  |  |
| --- | --- | --- |
| **MONTH** | **USERS** | **Growth\_Percent** |
| 1 | 1415 |  |
| 2 | 1382 | -2.33 |
| 3 | 1614 | 16.79 |
| 4 | 1829 | 13.32 |
| 5 | 2083 | 13.89 |

|  |  |  |
| --- | --- | --- |
| 6 | 2213 | 6.24 |
| 7 | 2591 | 17.08 |
| 8 | 2626 | 1.35 |
| 9 | 699 | -73.38 |
| 10 | 826 | 18.17 |
| 11 | 816 | -1.21 |
| 12 | 972 | 19.12 |

**2C)Calculate the weekly retention of users-sign up cohort? SQL QUERY:**

SELECT first\_week,

SUM(CASE WHEN week\_number = 0 THEN 1 ELSE 0 END) AS Week\_0, SUM(CASE WHEN week\_number = 1 THEN 1 ELSE 0 END) AS Weeek\_1, SUM(CASE WHEN week\_number = 2 THEN 1 ELSE 0 END) AS Week\_2, SUM(CASE WHEN week\_number = 3 THEN 1 ELSE 0 END) AS Week\_3, SUM(CASE WHEN week\_number = 4 THEN 1 ELSE 0 END) AS Week\_4, SUM(CASE WHEN week\_number = 5 THEN 1 ELSE 0 END) AS Week\_5, SUM(CASE WHEN week\_number = 6 THEN 1 ELSE 0 END) AS Week\_6, SUM(CASE WHEN week\_number = 7 THEN 1 ELSE 0 END) AS Week\_7, SUM(CASE WHEN week\_number = 8 THEN 1 ELSE 0 END) AS Week\_8, SUM(CASE WHEN week\_number = 9 THEN 1 ELSE 0 END) AS Week\_9, SUM(CASE WHEN week\_number = 10 THEN 1 ELSE 0 END) AS Week\_10, SUM(CASE WHEN week\_number = 11 THEN 1 ELSE 0 END) AS Week\_11, SUM(CASE WHEN week\_number = 12 THEN 1 ELSE 0 END) AS Week\_12,

SUM(CASE WHEN week\_number = 13 THEN 1 ELSE 0 END) AS Week\_13, SUM(CASE WHEN week\_number = 14 THEN 1 ELSE 0 END) AS Week\_14, SUM(CASE WHEN week\_number = 15 THEN 1 ELSE 0 END) AS Week\_15, SUM(CASE WHEN week\_number = 16 THEN 1 ELSE 0 END) AS Week\_16, SUM(CASE WHEN week\_number = 17 THEN 1 ELSE 0 END) AS Week\_17, SUM(CASE WHEN week\_number = 18 THEN 1 ELSE 0 END) AS Week\_18, SUM(CASE WHEN week\_number = 19 THEN 1 ELSE 0 END) AS Week\_19

FROM (SELECT a.user\_id,a.login\_week,b.first\_week as first\_week,a.login\_week-first\_week as week\_number FROM (SELECT user\_id,week(occurred\_at) AS login\_week FROM events GROUP BY

user\_id,week(occurred\_at)) a,(SELECT user\_id, min(week(occurred\_at)) AS first\_week FROM events GROUP BY user\_id) b where a.user\_id=b.user\_id) as with\_week\_number group by first\_week order by first\_week;

# QUERY OUTPUT:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| first  \_we ek | we ek\_ 0 | we ek\_ 1 | we ek\_ 2 | we ek\_ 3 | we ek\_ 4 | we ek\_ 5 | we ek\_ 6 | we ek\_ 7 | we ek\_ 8 | we ek\_ 9 | we ek\_ 10 | we ek\_ 11 | we ek\_ 12 | we ek\_ 13 | we ek\_ 14 | we ek\_ 15 | we ek\_ 16 | we ek\_ 17 | we ek\_ 18 | we ek\_ 19 |
| 17 | 740 | 472 | 324 | 251 | 205 | 187 | 167 | 146 | 145 | 145 | 136 | 131 | 132 | 143 | 116 | 91 | 82 | 77 | 5 | 0 |
| 18 | 788 | 362 | 261 | 203 | 168 | 147 | 144 | 127 | 113 | 122 | 106 | 118 | 127 | 110 | 97 | 85 | 67 | 4 | 0 | 0 |
| 19 | 601 | 284 | 173 | 153 | 114 | 95 | 91 | 81 | 95 | 82 | 68 | 65 | 63 | 42 | 51 | 49 | 2 | 0 | 0 | 0 |
| 20 | 555 | 223 | 165 | 121 | 91 | 72 | 63 | 67 | 63 | 65 | 67 | 41 | 40 | 33 | 40 | 0 | 0 | 0 | 0 | 0 |
| 21 | 495 | 187 | 131 | 91 | 74 | 63 | 75 | 72 | 58 | 48 | 45 | 39 | 35 | 28 | 2 | 0 | 0 | 0 | 0 | 0 |
| 22 | 521 | 224 | 150 | 107 | 87 | 73 | 63 | 60 | 55 | 48 | 41 | 39 | 31 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 23 | 542 | 219 | 138 | 101 | 90 | 79 | 69 | 61 | 54 | 47 | 35 | 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 24 | 535 | 205 | 143 | 102 | 81 | 63 | 65 | 61 | 38 | 39 | 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 25 | 500 | 218 | 139 | 101 | 75 | 63 | 50 | 46 | 38 | 35 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 26 | 495 | 181 | 114 | 83 | 73 | 55 | 47 | 43 | 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 27 | 493 | 199 | 121 | 106 | 68 | 53 | 40 | 36 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 28 | 486 | 194 | 114 | 69 | 46 | 30 | 28 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 29 | 501 | 186 | 102 | 65 | 47 | 40 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 30 | 533 | 202 | 121 | 78 | 53 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 31 | 430 | 145 | 76 | 57 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 32 | 496 | 188 | 94 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 33 | 499 | 202 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 34 | 518 | 44 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 35 | 32 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**2D)Calculate the weekly engagement per device? SQL QUERY:**

SELECT WEEK(occurred\_at) AS WEEK\_NUM,

COUNT(DISTINCT CASE WHEN device IN('dell inspiron notebook')THEN user\_id ELSE NULL END) AS "Dell inspiron notebook",

COUNT(DISTINCT CASE WHEN device IN('iphone 5')THEN user\_id ELSE NULL END) AS "Iphone 5",

COUNT(DISTINCT CASE WHEN device IN('iphone 4s')THEN user\_id ELSE NULL END) AS "Iphone 4s",

COUNT(DISTINCT CASE WHEN device IN('windows surface')THEN user\_id ELSE NULL END) AS "Windows surface",

COUNT(DISTINCT CASE WHEN device IN('macbook air')THEN user\_id ELSE NULL END) AS "Macbook air",

COUNT(DISTINCT CASE WHEN device IN('iphone 5s')THEN user\_id ELSE NULL END) AS "Iphone 5s",

COUNT(DISTINCT CASE WHEN device IN('macbook pro')THEN user\_id ELSE NULL END) AS "Macbook pro",

COUNT(DISTINCT CASE WHEN device IN('kindle fire')THEN user\_id ELSE NULL END) AS "Kindle fire",

COUNT(DISTINCT CASE WHEN device IN('ipad mini')THEN user\_id ELSE NULL END) AS "Ipad mini",

COUNT(DISTINCT CASE WHEN device IN('nexus 7')THEN user\_id ELSE NULL END) AS "Nexus 7",

COUNT(DISTINCT CASE WHEN device IN('nexus 5')THEN user\_id ELSE NULL END) AS "Nexus 5",

COUNT(DISTINCT CASE WHEN device IN('samsung galaxy s4')THEN user\_id ELSE NULL END) AS "Samsung galaxy s4",

COUNT(DISTINCT CASE WHEN device IN('lenovo thinkpad')THEN user\_id ELSE NULL END) AS "Lenovo thinkpad",

COUNT(DISTINCT CASE WHEN device IN('samsung galaxy tablet')THEN user\_id ELSE NULL END) AS "Samsung galaxy tablet",

COUNT(DISTINCT CASE WHEN device IN('acer aspire notebook')THEN user\_id ELSE NULL END) AS "Acer aspire notebook",

COUNT(DISTINCT CASE WHEN device IN('asus chromebook')THEN user\_id ELSE NULL END) AS "Asus chromebook",

COUNT(DISTINCT CASE WHEN device IN('htc one')THEN user\_id ELSE NULL END) AS "Htc one",

COUNT(DISTINCT CASE WHEN device IN('nokia lumia 635')THEN user\_id ELSE NULL END) AS "Nokia lumia 635",

COUNT(DISTINCT CASE WHEN device IN('samsung galaxy note')THEN user\_id ELSE NULL END) AS "Samsung galaxy note",

COUNT(DISTINCT CASE WHEN device IN('acer aspire desktop')THEN user\_id ELSE NULL END) AS "Acer aspire desktop",

COUNT(DISTINCT CASE WHEN device IN('mac mini')THEN

user\_id ELSE NULL END) AS "Mac mini",

COUNT(DISTINCT CASE WHEN device IN('hp pavilion desktop')THEN user\_id ELSE NULL END) AS "Hp pavilion desktop",

COUNT(DISTINCT CASE WHEN device IN('Dell inspiron desktop')THEN user\_id ELSE NULL END) AS "Dell inspiron desktop",

COUNT(DISTINCT CASE WHEN device IN('ipad air')THEN user\_id ELSE NULL END) AS "Ipad air",

COUNT(DISTINCT CASE WHEN device IN('amazon fire phone')THEN user\_id ELSE NULL END) AS "Amazon fire phone",

COUNT(DISTINCT CASE WHEN device IN('nexus 10')THEN

user\_id ELSE NULL END) AS "Nexus 10"

FROM events WHERE event\_type='engagement' GROUP BY 1 ORDER BY 1;

# QUERY OUTPUT:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **WEE K\_NU M** | **Del l ins pir on not ebo ok** | **Ip ho ne 5** | **Ip ho ne 4s** | **Wi nd ow s sur fac e** | **Ma cbo ok air** | **Ip ho ne 5s** | **Ma cbo ok pro** | **Ki ndl e fire** | **Ip ad mi ni** | **Ne xus 7** | **Ne xus 5** | **Sa ms un g gal axy s4** | **Le no vo thi nk pa d** | **Sa ms un g gal axy tab let** | **Ac er asp ire not ebo ok** | **Asus chro mebo ok** | **Ht c on e** | **No ki a lu mi a 63**  **5** | **Sa ms un g gal axy not e** | **Ac er asp ire des kto p** | **M**  **ac mi ni** | **Hp pa vili on des kto p** | **Del l ins pir on des kto p** | **Ip ad air** | **A**  **ma zon fire ph one** | **Ne xus 10** |
| **17** | **46** | **65** | **21** | **10** | **54** | **42** | **143** | **6** | **19** | **18** | **40** | **52** | **86** | **8** | **20** | **21** | **16** | **17** | **7** | **9** | **6** | **14** | **18** | **27** | **4** | **16** |
| **18** | **77** | **113** | **46** | **10** | **121** | **73** | **252** | **27** | **30** | **30** | **73** | **82** | **153** | **11** | **33** | **42** | **19** | **33** | **15** | **26** | **13** | **37** | **58** | **52** | **9** | **30** |
| **19** | **83** | **115** | **44** | **16** | **112** | **79** | **266** | **21** | **36** | **41** | **87** | **91** | **178** | **6** | **41** | **27** | **30** | **23** | **11** | **23** | **18** | **40** | **36** | **55** | **12** | **25** |
| **20** | **84** | **125** | **55** | **21** | **119** | **79** | **256** | **23** | **32** | **32** | **103** | **93** | **173** | **9** | **40** | **41** | **29** | **22** | **18** | **23** | **26** | **30** | **52** | **59** | **11** | **22** |
| **21** | **80** | **137** | **45** | **17** | **110** | **74** | **247** | **30** | **23** | **29** | **91** | **84** | **167** | **6** | **47** | **38** | **21** | **25** | **20** | **29** | **18** | **44** | **41** | **51** | **5** | **25** |
| **22** | **92** | **125** | **45** | **15** | **145** | **71** | **251** | **21** | **34** | **45** | **96** | **105** | **176** | **10** | **41** | **52** | **24** | **25** | **19** | **25** | **25** | **38** | **52** | **58** | **5** | **27** |
| **23** | **103** | **152** | **53** | **14** | **124** | **79** | **266** | **25** | **33** | **36** | **88** | **99** | **176** | **14** | **43** | **49** | **20** | **31** | **14** | **22** | **18** | **54** | **53** | **41** | **16** | **45** |
| **24** | **99** | **142** | **53** | **22** | **152** | **79** | **255** | **25** | **39** | **49** | **87** | **101** | **165** | **11** | **40** | **43** | **20** | **35** | **20** | **24** | **29** | **56** | **59** | **57** | **11** | **38** |
| **25** | **105** | **137** | **40** | **22** | **121** | **78** | **275** | **24** | **30** | **51** | **89** | **99** | **197** | **12** | **47** | **38** | **21** | **37** | **14** | **28** | **21** | **52** | **52** | **57** | **13** | **29** |
| **26** | **89** | **152** | **50** | **21** | **134** | **94** | **269** | **26** | **43** | **46** | **87** | **112** | **192** | **12** | **35** | **49** | **23** | **42** | **9** | **29** | **11** | **46** | **60** | **56** | **13** | **29** |
| **27** | **89** | **163** | **67** | **33** | **142** | **83** | **302** | **25** | **35** | **40** | **84** | **116** | **202** | **15** | **49** | **52** | **27** | **31** | **15** | **29** | **15** | **56** | **53** | **55** | **10** | **37** |
| **28** | **103** | **151** | **61** | **33** | **148** | **93** | **295** | **31** | **35** | **39** | **85** | **122** | **220** | **9** | **49** | **50** | **26** | **35** | **10** | **30** | **28** | **56** | **56** | **54** | **6** | **26** |
| **29** | **113** | **144** | **60** | **28** | **148** | **90** | **295** | **37** | **34** | **45** | **77** | **123** | **209** | **13** | **53** | **49** | **31** | **43** | **16** | **28** | **31** | **58** | **54** | **52** | **12** | **25** |
| **30** | **127** | **152** | **65** | **19** | **159** | **103** | **322** | **25** | **35** | **62** | **84** | **103** | **206** | **9** | **60** | **56** | **31** | **34** | **15** | **33** | **23** | **42** | **54** | **70** | **12** | **36** |
| **31** | **113** | **135** | **56** | **19** | **147** | **71** | **321** | **14** | **27** | **38** | **69** | **100** | **207** | **8** | **55** | **56** | **13** | **28** | **14** | **31** | **24** | **51** | **44** | **55** | **14** | **24** |
| **32** | **104** | **119** | **34** | **10** | **125** | **67** | **307** | **12** | **30** | **25** | **67** | **82** | **179** | **6** | **55** | **62** | **18** | **28** | **12** | **35** | **20** | **51** | **57** | **48** | **12** | **30** |
| **33** | **110** | **110** | **35** | **15** | **133** | **65** | **312** | **14** | **28** | **30** | **70** | **80** | **191** | **12** | **46** | **49** | **19** | **27** | **13** | **39** | **32** | **38** | **37** | **40** | **14** | **23** |
| **34** | **105** | **101** | **50** | **18** | **136** | **70** | **292** | **13** | **25** | **33** | **70** | **90** | **193** | **14** | **63** | **47** | **25** | **17** | **13** | **30** | **30** | **36** | **49** | **39** | **11** | **25** |
| **35** | **9** | **2** | **6** | **3** | **10** | **3** | **17** | **3** | **2** | **2** | **4** | **6** | **16** | **0** | **3** | **6** | **2** | **2** | **1** | **1** | **2** | **1** | **1** | **0** | **0** | **2** |

**2E) Calculate the email engagement metrics? SQL QUERY:**

SELECT WEEK(occurred\_at) AS Week,

COUNT(DISTINCT CASE WHEN action IN('sent\_weekly\_digest')THEN user\_id ELSE NULL END) AS "Sent weekly digest",

COUNT(DISTINCT CASE WHEN action IN('email\_open')THEN user\_id ELSE NULL END) AS "Email open",

COUNT(DISTINCT CASE WHEN action IN('email\_clickthrough')THEN user\_id ELSE NULL END) AS "Email clickthrough",

COUNT(DISTINCT CASE WHEN action IN('sent\_reengagement\_email')THEN user\_id ELSE NULL END) AS "Sent reengagement email"

FROM email\_events GROUP BY 1 ORDER BY 1;

# QUERY OUTPUT:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week** | **Sent weekly digest** | **Email open** | **Email clickthrough** | **Sent**  **reengagement email** |
| 17 | 908 | 310 | 166 | 73 |
| 18 | 2602 | 900 | 425 | 157 |
| 19 | 2665 | 961 | 476 | 173 |
| 20 | 2733 | 989 | 501 | 191 |
| 21 | 2822 | 996 | 436 | 164 |
| 22 | 2911 | 965 | 478 | 192 |
| 23 | 3003 | 1057 | 529 | 197 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 24 | 3105 | 1136 | 549 | 226 |
| 25 | 3207 | 1084 | 524 | 196 |
| 26 | 3302 | 1149 | 550 | 219 |
| 27 | 3399 | 1207 | 613 | 213 |
| 28 | 3499 | 1228 | 594 | 213 |
| 29 | 3592 | 1201 | 583 | 213 |
| 30 | 3706 | 1363 | 625 | 231 |
| 31 | 3793 | 1338 | 444 | 222 |
| 32 | 3897 | 1318 | 416 | 200 |
| 33 | 4012 | 1417 | 490 | 264 |
| 34 | 4111 | 1502 | 481 | 261 |
| 35 | 0 | 41 | 38 | 48 |

**RESULT:**

Overall, my experience working with SQL during the project proved to be an invaluable skill, as it enabled me to derive insightful information and make informed decisions based on the data. The expertise I gained in this area will undoubtedly benefit me in future projects and analysis work.

# DRIVE LINK:

[https://docs.google.com/document/d/170EqlN4l-uxXD\_YEdP\_HLZ7xerUZegkADi7zT4MM](https://docs.google.com/document/d/170EqlN4l-uxXD_YEdP_HLZ7xerUZegkADi7zT4MMkXU/edit?usp=sharing) [kXU/edit?usp=sharing](https://docs.google.com/document/d/170EqlN4l-uxXD_YEdP_HLZ7xerUZegkADi7zT4MMkXU/edit?usp=sharing)