To find the answer for question “what could be a root cause?” I need to answer for several questions firstly. Initially I recognize following crucial questions:

* what are the most common SMTP error messages?
* which domains are affected with this error?
* can we relate number of mail messages sent with bounce rate for particular domains?

To complete it I need to analyse provided data set. As it contains a lot of data, some adjustment activities would be helpful to make analyse easier.

After that I need to familiarize with adjusted set and try to establish conclusions. It will allow me to provide hypothesis about the root cause and maybe to propose a solution.

All activities described above can be grouped into following steps:

1. [Initial data review and SMTP responses recognition](#_Step_1_-)
2. [Data set adjustment](#_Step_2_-)
3. [Extraction of necessary data and transformation](#_Step_3_-)
4. [Import into analytical environment and data analysis](#_Step_4_-)
5. [Final conclusions](#_Step_5_–)

Each point is described in sections below. My hypothesis and propositions are underlined in data analysis section and summarized in final conclusions.

# *Step 1 - Initial data review and SMTP responses recognition*

Data contains information about occurrence of various SMTP responses on pointed date. Date is provided in UNIX format. Both SMTP codes and UNIX date require further activities. Let’s start with SMTP codes recognition.

We face several types of responses in input file. I have checked meaning of each of them using document available [here](https://hosteng.com/faqfiles/SMTP%20Server%20Status%20Codes%20and%20Errors.pdf). There is also [website](https://smtpfieldmanual.com/code/451) focused on SMTP codes which was helpful.

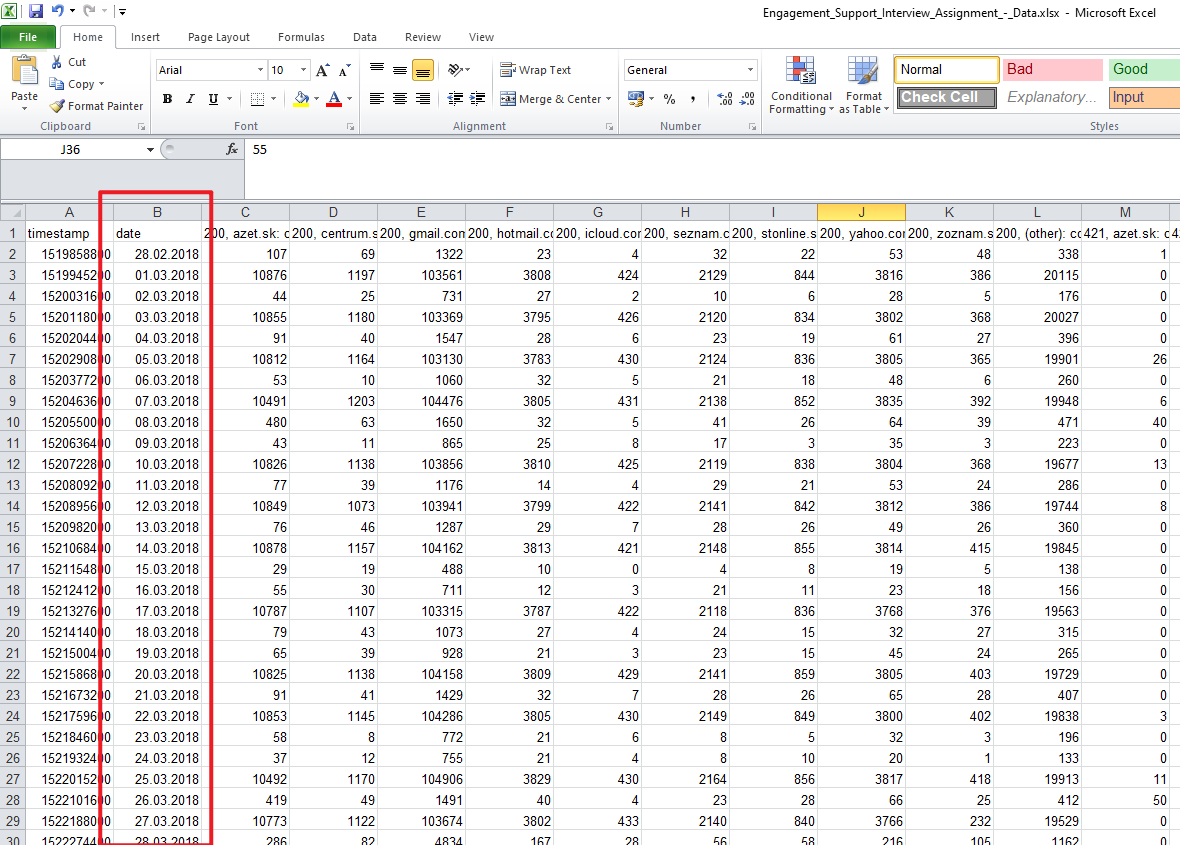
Following table presents codes and respective description:

|  |  |
| --- | --- |
| Code | Description |
| 200 – Positive completion reply | The requested action has been successfully completed. |
| 421 - Service not available | The Mail transfer service is unavailable. This can be caused by many things such as a server administrator stopping the mail service, or rebooting the mail server. |
| 451 - Requested action aborted – Local error in processing | The action has been aborted by the ISP’s server. |
| 452 - Requested action not taken – Insufficient storage | This is usually caused by overloading mail server when attempting to send too many messages at once. |
| 499 - Client closed request | It indicates that the client has closed the connection while the server is still processing the request. (not confirmed) |
| 550 - Requested actions not taken mailbox unavailable | Recipient email address simply does not exist on the remote side. |
| 552 - Requested mail actions aborted – Exceeded storage allocation | Mailbox has reached its maximum allowed size. |
| 554 – Transaction failed | Recipient email address does not exist or there is anti-spam firewall. |
| 605 | Email address is currently suppressed by our system from further delivery attempts. (not confirmed) |

# *Step 2 - Data set adjustment*

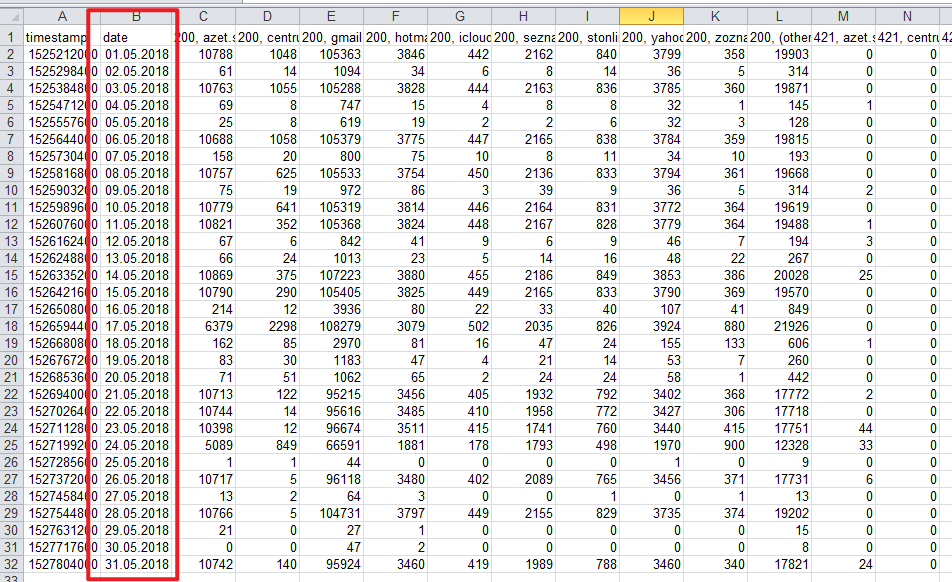
As the task is focused on May campaign (increased bounce rate is also present in other months, but I treat it as out of scope) first thing to do is to filter data from May. To complete it I need to convert UNIX timestamp to date in ‘user friendly’ form. I do it directly in excel sheet using formula:

New column with DD.MM.YYYY format is now displayed next to timestamp:

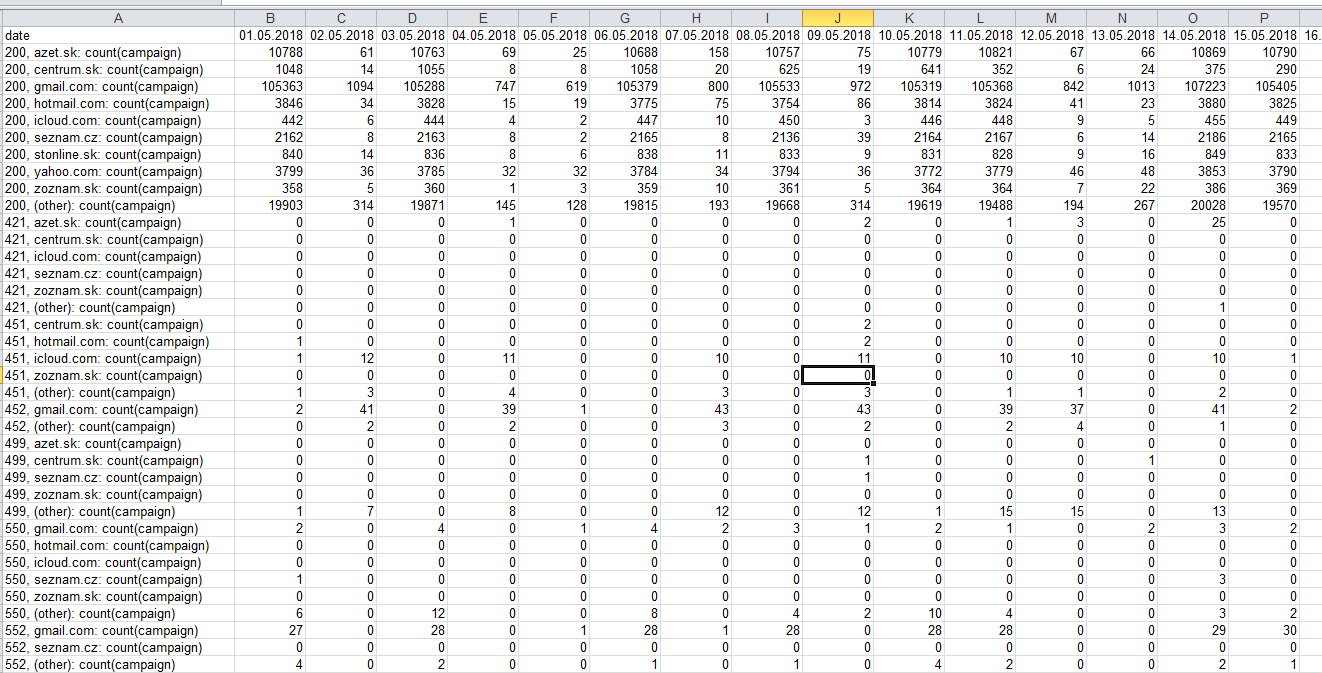


# *Step 3 - Extraction of necessary data and transformation*

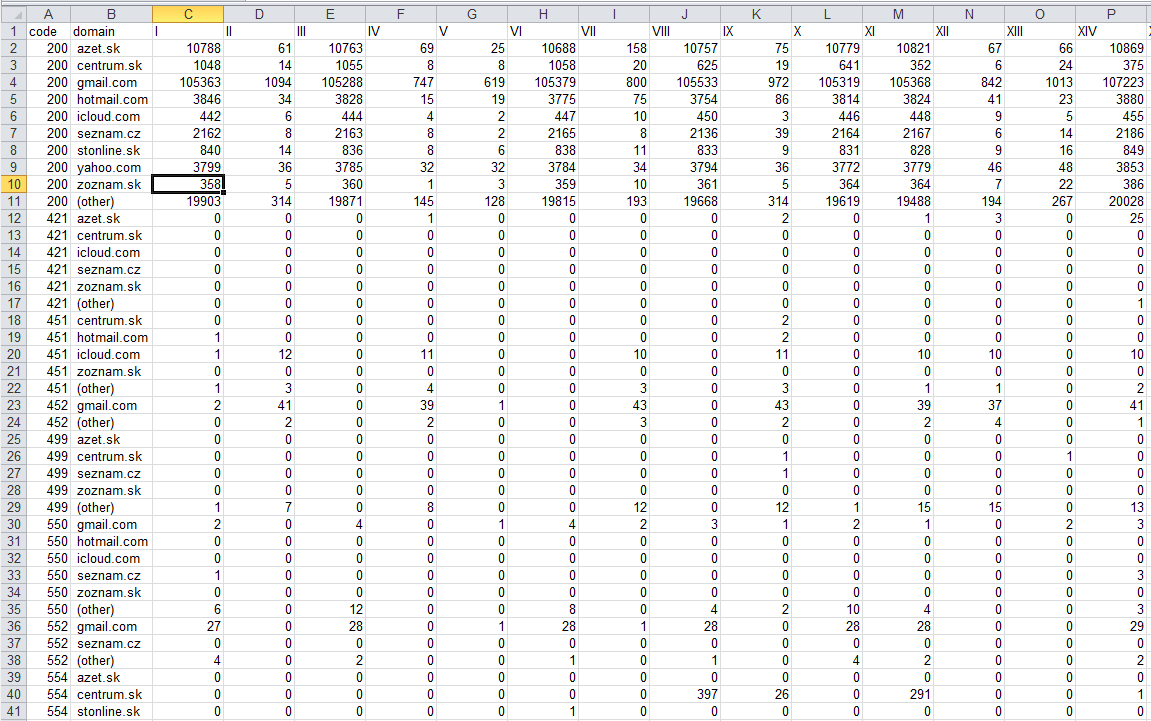
Now I clip data related to May into another sheet:



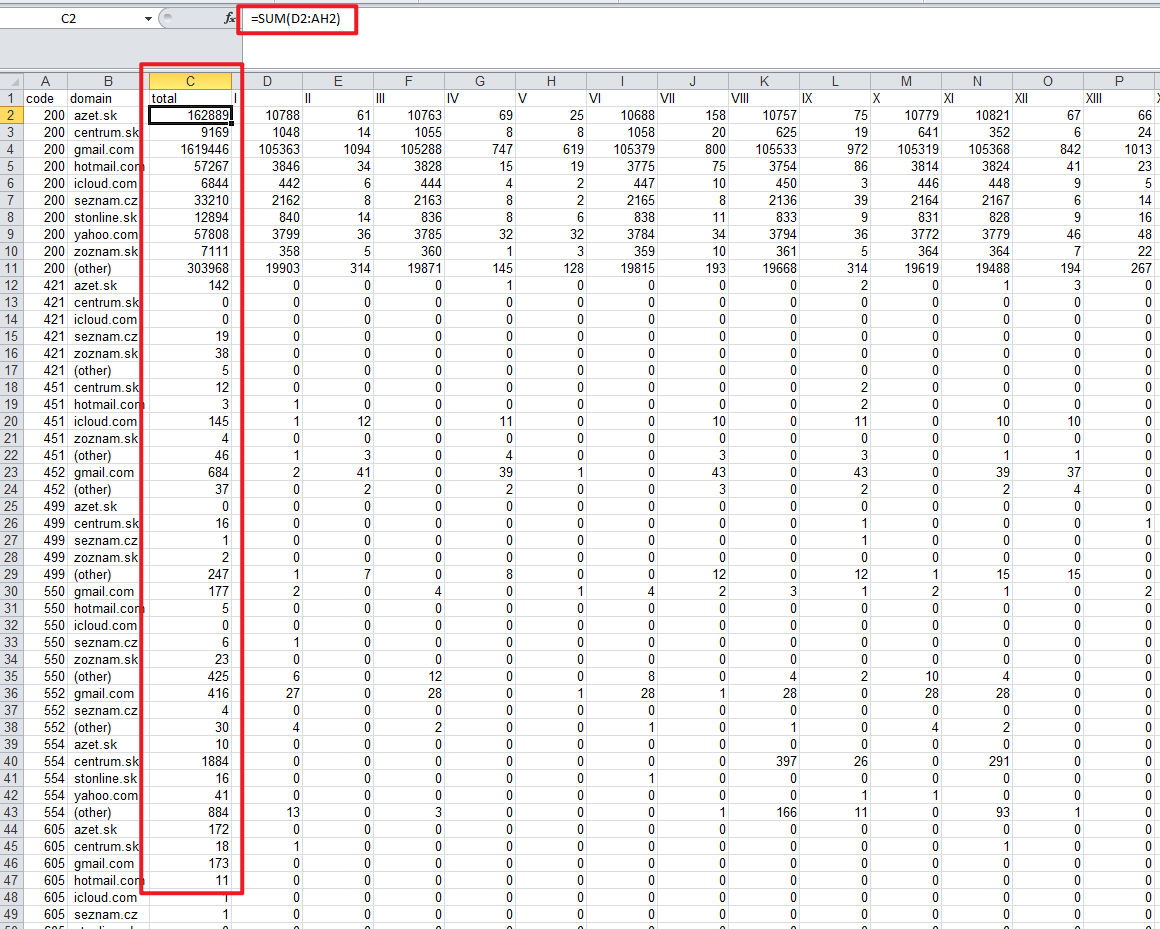
I have reduced the overall number of records but there is still a lot of columns. What is more each column contains 2 information – SMTP code and domain. Analysis of data in this shape is inefficient and as a result it is difficult to find out answers for question listed at the beginning. My idea is to analyse particular codes and domains instead of dates so I will do transposition, separate codes and domains, add new column with total number of responses and load it as a database table to do SQL queries. SQL will allow me to do various checks helpful to answer posted questions. Let’s begin with transposition. I will copy necessary data and paste it with transposition in new sheet:



Now I will separate first column, remove unnecessary string ‘*count(campaign)*’, and rename columns to be more meaningful and database friendly:



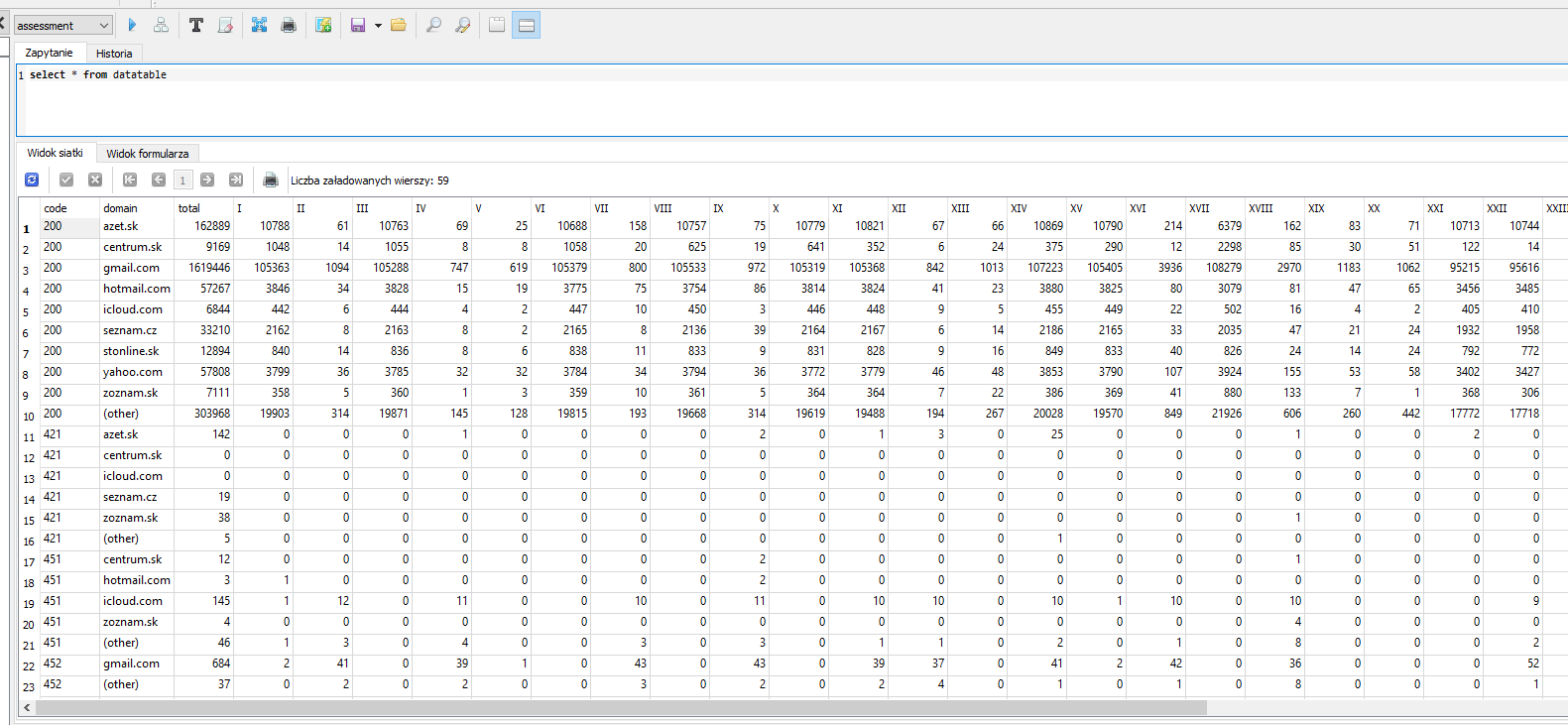
It’s against relational databases rules but in this case it will be useful to add another column with sum of responses from entire month:



Data is ready to be loaded as a database table.

# Step 4 - Import into analytical environment and data analysis

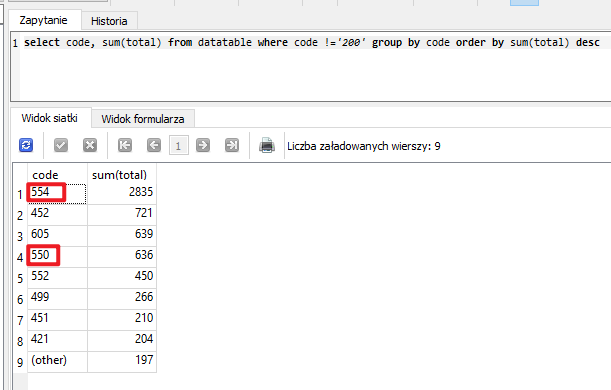
I use SQLite RDBMS as it is free software. Data is extracted from excel into csv format and then loaded into database table:



## *Code 554, 550*

Now database table is ready to do analysis. Let’s start with check which code was returned most often:

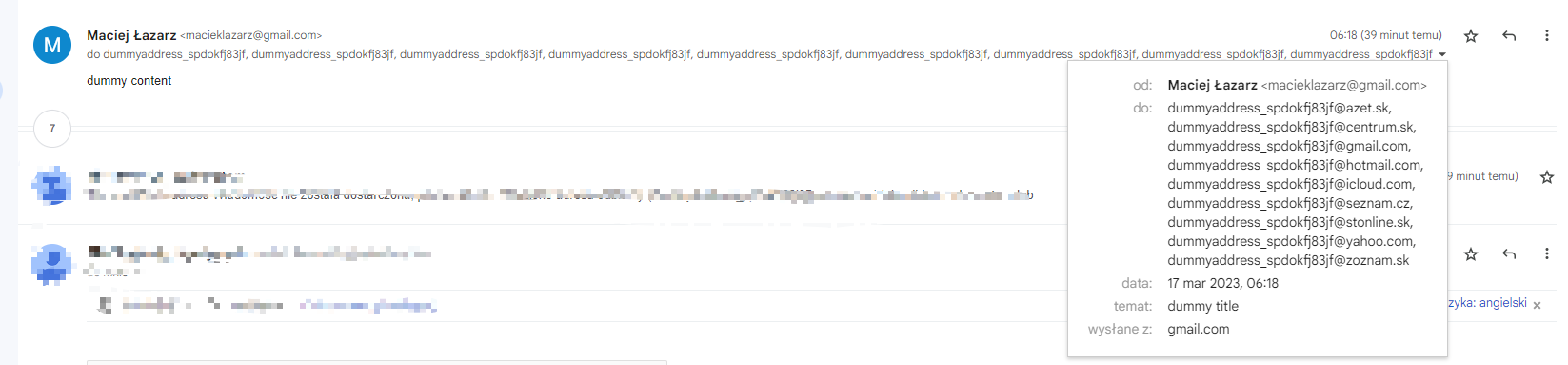
*select code, sum(total) from datatable where code !='200' group by code order by sum(total) desc*



We can see 554 code in position number 1 with large advantage before 452 in position number 2. There is also great number in case of code 550 and I intentionally consider it before I consider codes between 554 and 550 because both 550 and 554 are returned in similar situation – not valid mail address. This is one of supposed reasons of failure for mentioned codes and the second is anti-spam firewall. In this situation (if I was granted with access) I would check if there is some additional message returned with code which will clarify what is the reason of failure, because with error code there could be also message like *“550 Invalid recipient”* or “*550 User account is unavailable”* attached. It will allow me to list invalid mail address and inform client about it.

In case of failure caused by anti-spam firewall I would notify person responsible for the mailing tool about that fact pointing cases (email addresses) when it occur to allow him to do proper reconfiguration or changes in the tool to avoid firewall.

I will do small check. Let’s send a mail to dummy addresses for each domain and analyze response:

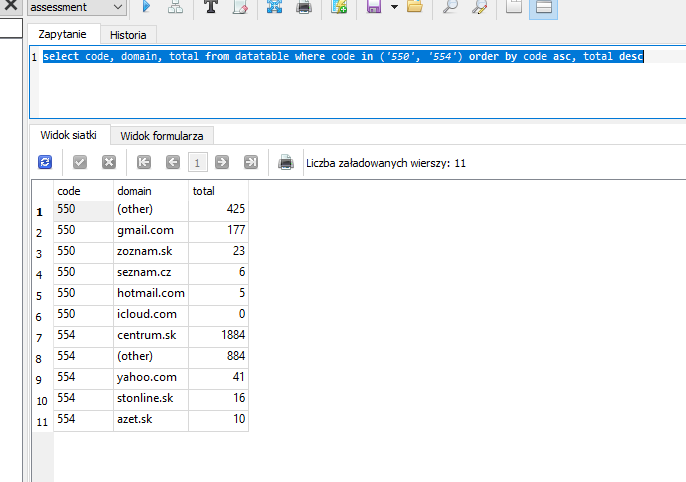


Answers are:

|  |  |
| --- | --- |
| Domain | Response |
| azet.sk | 550 5.1.1 Recipient address rejected |
| centrum.sk | 550 #5.1.0 Address rejected. |
| gmail.com | 550 5.1.1 The email account that you tried to reach does not exist. |
| hotmail.com | 550 5.5.0 Requested action not taken: mailbox unavailable |
| icloud.com | 550 5.1.1 user does not exist |
| seznam.cz | 550 5.1.1 sorry, no such mailbox here |
| stonline.sk | 550 5.1.1 Recipient address rejected: User unknown in local recipient table |
| yahoo.com | 552 1 Requested mail action aborted, mailbox not found |
| zoznam.sk | 550 5.1.1 Recipient address rejected: User unknown in virtual mailbox table |

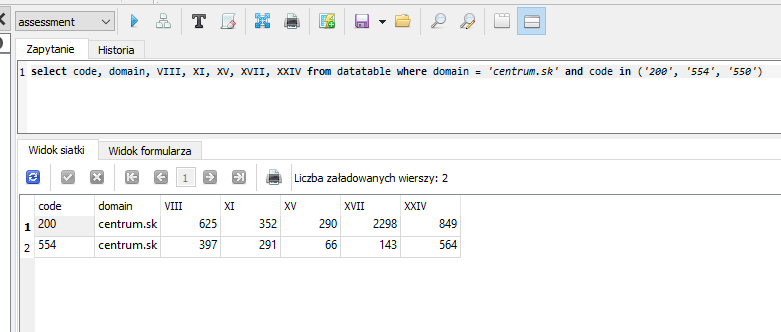
As we can see in almost all cases we have 550 response. It leads me to conclusion that in case of 550 code the reason is wrong address while in case of 554 the reason is probably firewall.

Now let’s check if these errors are most common for some particular domains:

*select code, domain, total from datatable where code in ('554', '550') order by code asc, total desc* 

As we can see centrum.sk domain is most affected in case of 554 code. In this case let’s take a closer look into emails sent to centrum.sk domain:

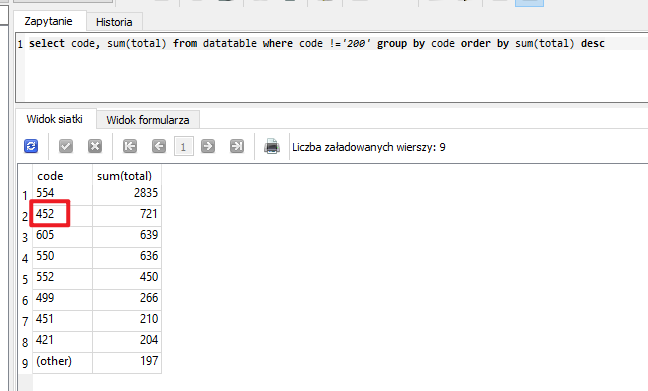
*select code, domain, VIII, XI, XV, XVII, XXIV from datatable where domain = 'centrum.sk' and code in ('200', '554', '550')*



554 code occurred among huge number of successfully delivered messages. It allows me to conclude that email recipients added sender to SPAM list.

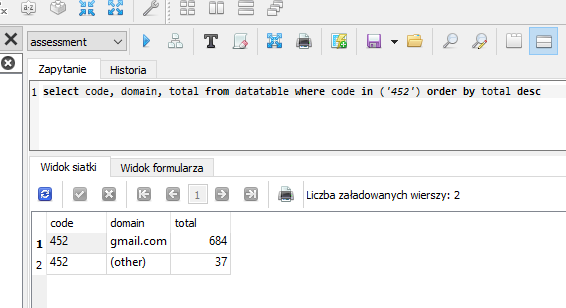
Nevertheless, code comes with description so I would check the description in production database to make sure if my hypothesis is correct.

## *Code 452*



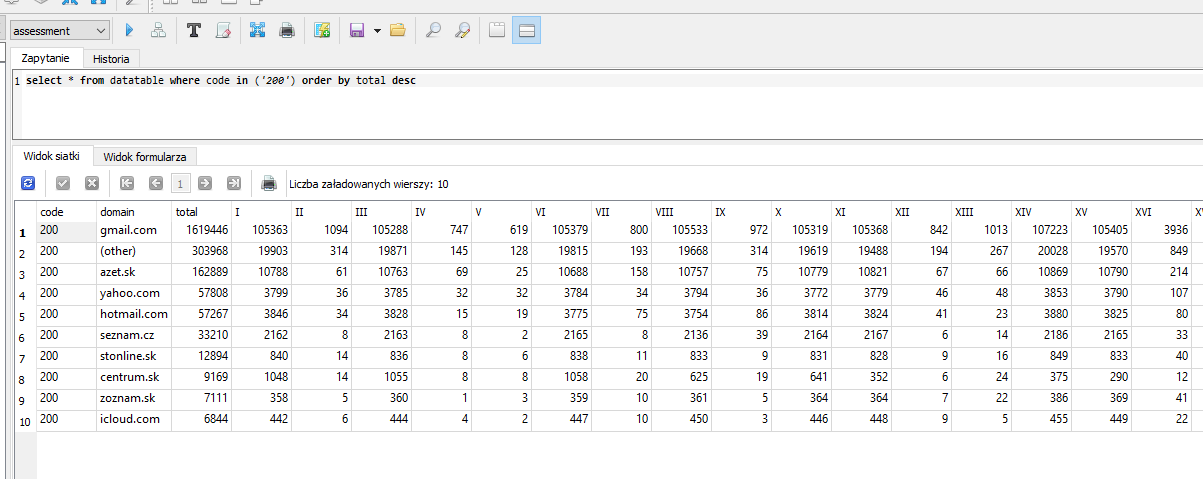
Code 452 is second on our list. It indicates that server is overloaded with too many messages. I suppose it occurs when huge number of mails is tried to be sent at the same time. Let’s check it in our table:

*select code, domain, total from datatable where code in ('452') order by total desc*



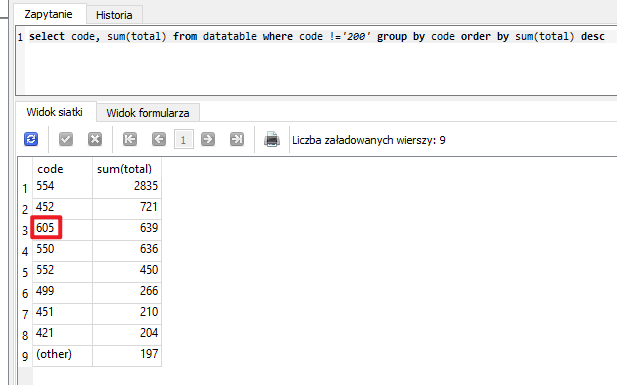
As we can see in this case gmail.com is mostly affected. I assume that it may be caused by the fact that gmail is really popular mail domain (so it needs to serve huge number of requests at the same time) and additionally as we can see below number daily messages delivered there is higher than in case of rest domains summed up:

*select \* from datatable where code in ('200') order by total desc*



In my opinion there is a possibility to deal with this problem by changing ratio of emails sent to gmail domain. There are various mechanisms responsible for mailing queue. From my experience I know that in case of Python there is Celery service responsible for queue management. We don’t know what kind of mechanism is utilized in this case but I suppose that there is a possibility to decrease ratio of message sending.

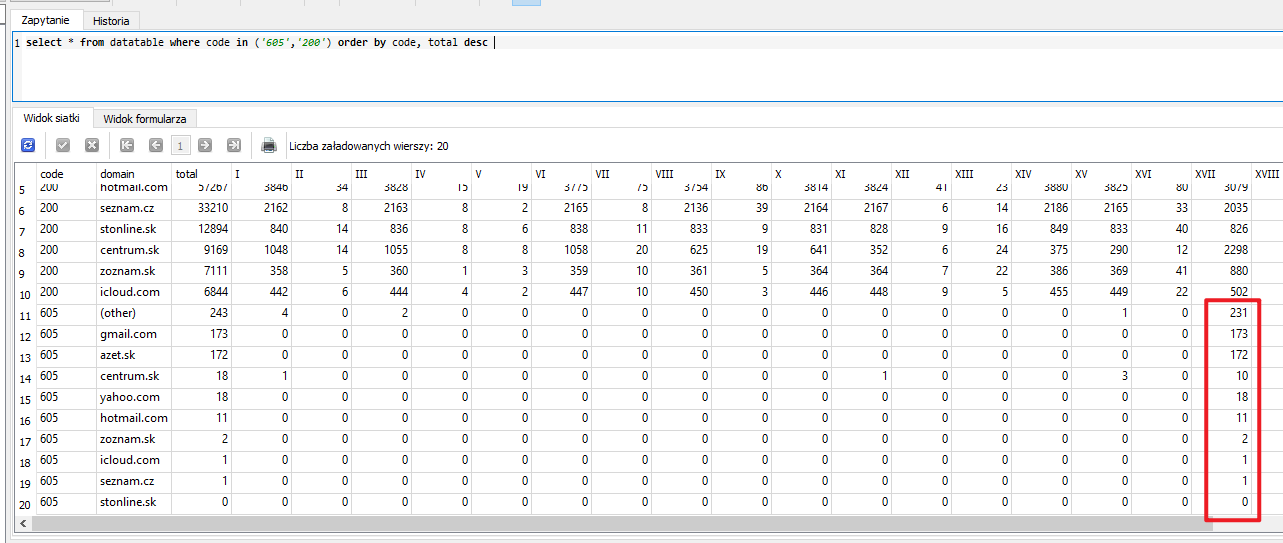
## *Code 605*



It was difficult to find the meaning of code 605 and I’m still not 100% sure if meaning found by me is correct in case of our data. I have found information about this error on [Mailgun](https://help.mailgun.com/hc/en-us/articles/360012152213--Not-Delivering-to-Previously-Bounced-Address-) and [Current RMS](https://help.current-rms.com/en/articles/3132448-troubleshoot-undelivered-discussion-emails) websites, both related to mail sending so I believe it is applicable in case of our data. Description in both cases points that email address is suspended from delivery attempts due to previous failure. Mentioned failure may be caused by various factors so it’s difficult to provide successful solution. Anyway I would recommend to verify if email address is correct (ex. with no spelling mistake) because it is the type of mistake which leads to 605 error in long run.

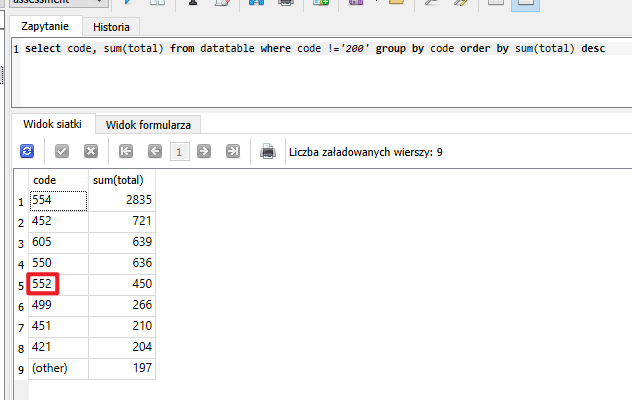
On the other hand we can look at this error from another perspective:

*select \* from datatable where code in ('605','200') order by code, total desc*

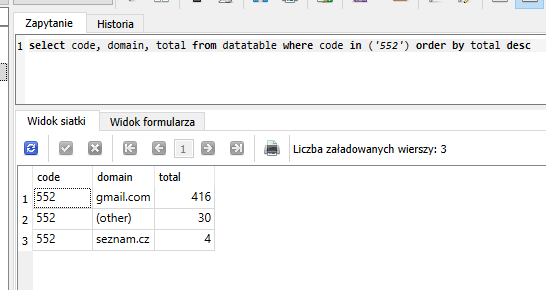


Almost every single example of 605 code occurred at the same date for every domain. In this case it is difficult to point root cause and possible solution but I would point next step of investigation – we should take a closer look into emails sent in that specific date.

## *Code 552*

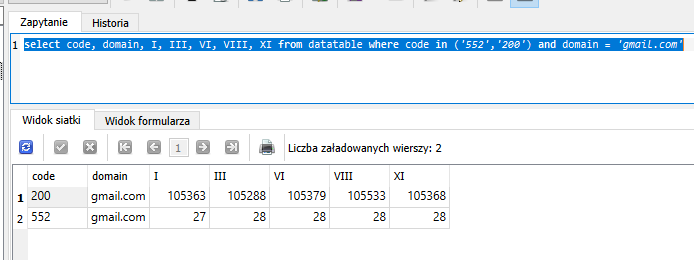


552 code is related to mailbox restrictions. Customers can adjust maximum size of received messages. If our mail exceeds this value it is rejected with code 552 (‘5.3.4 Message size exceeds fixed maximum message size’). Another reason for 552 code reception could be the type of attachment (error 552 – ‘5.7.0 Our system detected an illegal attachment on your message’). Let’s check which domains are affected:



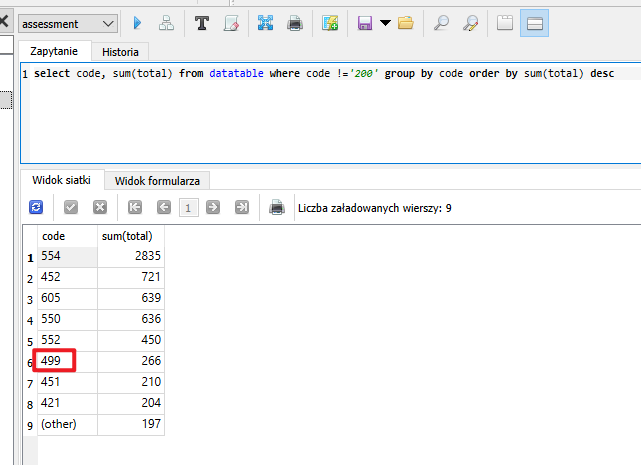
We can check file types blocked by Gmail [here](https://support.google.com/mail/answer/6590?hl=en#zippy=%2Cmessages-that-have-attachments) but in my opinion probably it is not a problem due to the fact that so many messages were delivered successfully at the same date and I suppose that content of both delivered and rejected messages were the same:

*select code, domain, I, III, VI, VIII, XI from datatable where code in ('552','200') and domain = 'gmail.com'*



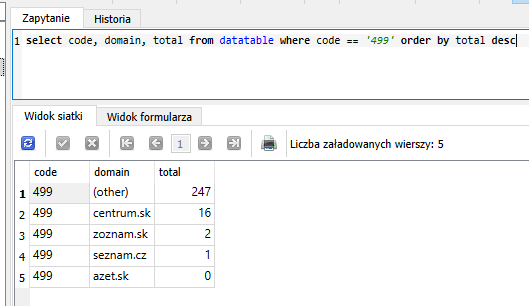
Moreover we can see that number of returned 552 codes is more less the same for different dates so it convinces that particular email recipients set message size restrictions. It is possible to set in Gmail [admin console](https://medhacloud.com/how-to-configure-attachment-compliance-in-gmail/). In May maximum number of rejected messages with 552 code was 31. It is not big number taking into consideration that hundreds of thousands messages are delivered successfully to Gmail at the same time. But if we want to reach affected recipients we could prepare dedicated message for them, with limited size.

## *Code 499*



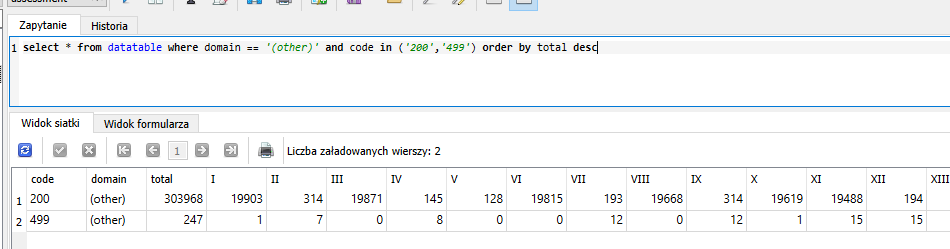
As in case of code 605 it is difficult to find meaning of this code. So let’s try to check if it is typical for some particular domain:

*select code, domain, total from datatable where code == '499' order by total desc*



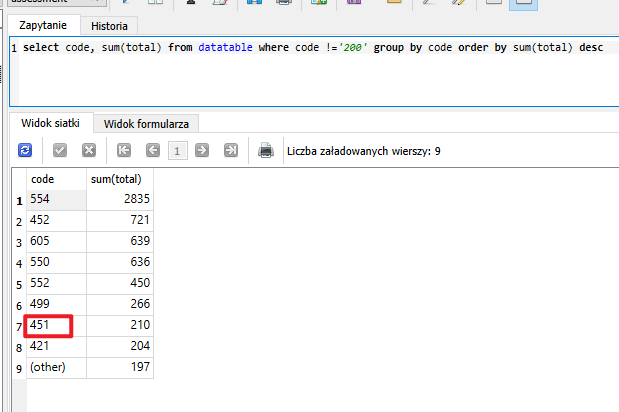
As we see it is impossible to point any domain in this case. So last thing we can do is to check how it occurred during entire month:

*select \* from datatable where domain == '(other)' and code in ('200','499') order by total desc*



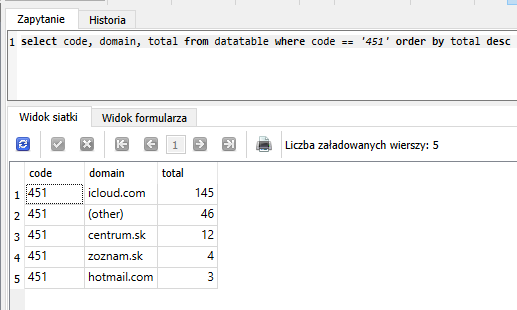
As we can see the occurrence of errors seems not to be correlated to overall number of sent messages. Similar number of errors are returned when huge number of messages are sent and when only few messages are sent (see day VII and XI).

## *Code 451*



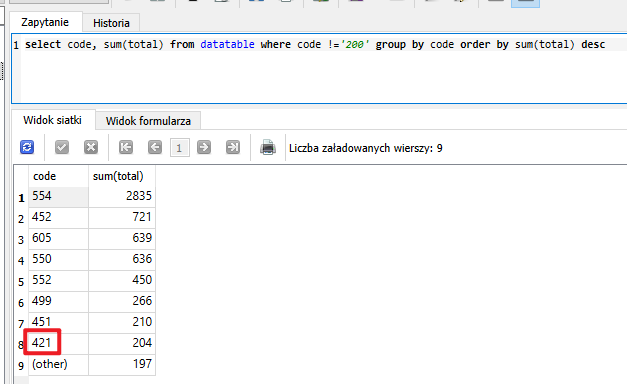
This error in most cases is a result of server temporary problem (see [here](https://smtpfieldmanual.com/code/451)). As usual let’s check affected domains:

*select code, domain, total from datatable where code == '451' order by total desc*



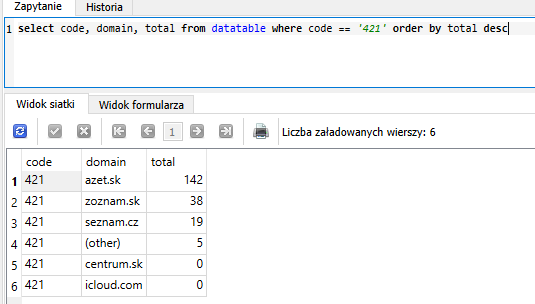
As we see icloud is domain with the biggest number of failures. As it is temporary unavailability problem I would recommend to do retry for failed messages.

## *Code 421*



Last code is 421. It also related to temporary unavailability. It might be caused by events like too many connections, busy server or rebooting mail server. I suppose that cause of problem is the same as in case of 451 and we distinct both 421 and 451 because one domain in certain conditions returns 421 while other under the same conditions returns 451. Let’s list domains affected with 421 and compare it with analogical list for 451:

*select code, domain, total from datatable where code == '421' order by total desc*



We can see that both lists contain different sets of domains. As previously I would recommend retry.

# Step 5 – Final conclusions

It is not possible to point one general root cause and propose one general solution as we face variety of codes. General conclusion could be that if we want to catch the root cause more precisely we need to verify messages received with codes. But with already known set of data we can list following assumptions:

* most common code **554** is most probably caused by SPAM filter on recipients side – mailing tool could be reconfigured the way it avoids firewall
* **550** code is returned in case of invalid mail address – I would recommend to verify if involved addresses are the same as requested by client (e.g. no undesirable mark while copying added) and notify client with list of invalid addresses

In both cases I would propose to develop dedicated ETL tool to support and automatize recommended activities. In first case ETL can extract affected addresses and notify proper team to adjust the tool. In second case it can extract invalid addresses and automatically notify client with list ofrecipients.

* 684 messages did not reach destination due to overload of Gmail (code **552**) – we can reduce it by limiting sending ratio in case of this domain
* it is difficult to explain the meaning of **605** code – if my hypothesis about blocking of previously failed attempts is correct it might help when solutions from above points will be implemented
* we can explain to the client that some recipients did not get the message because they limited allowed size of incoming messages (code **552**) – in addition we can propose that new message fitting the restrictions will be prepared
* both **451** and **421** codes describe temporary unavailability – I would recommend to point affected addresses and do retry after the time span