Anno Accademico 2020/2021

UNIVERSITÀ DEGLI STUDI DI PALERMO

Dipartimento di Scienze Economiche, Aziendali e Statistiche
Master annuale di secondo livello in

**Data Science and Big Data Analytics**

# ANALYSIS AND CLEANING OF A DATABASE ON BIGQUERY.

Tesi di:
Ludovica Tomaselli

Relatore:
Prof. Marcello Chiodi
Tutor aziendale:
Dott.ssa Francesca Motisi

# Introduction

## Internship location

Cloudtec

## Internship duration

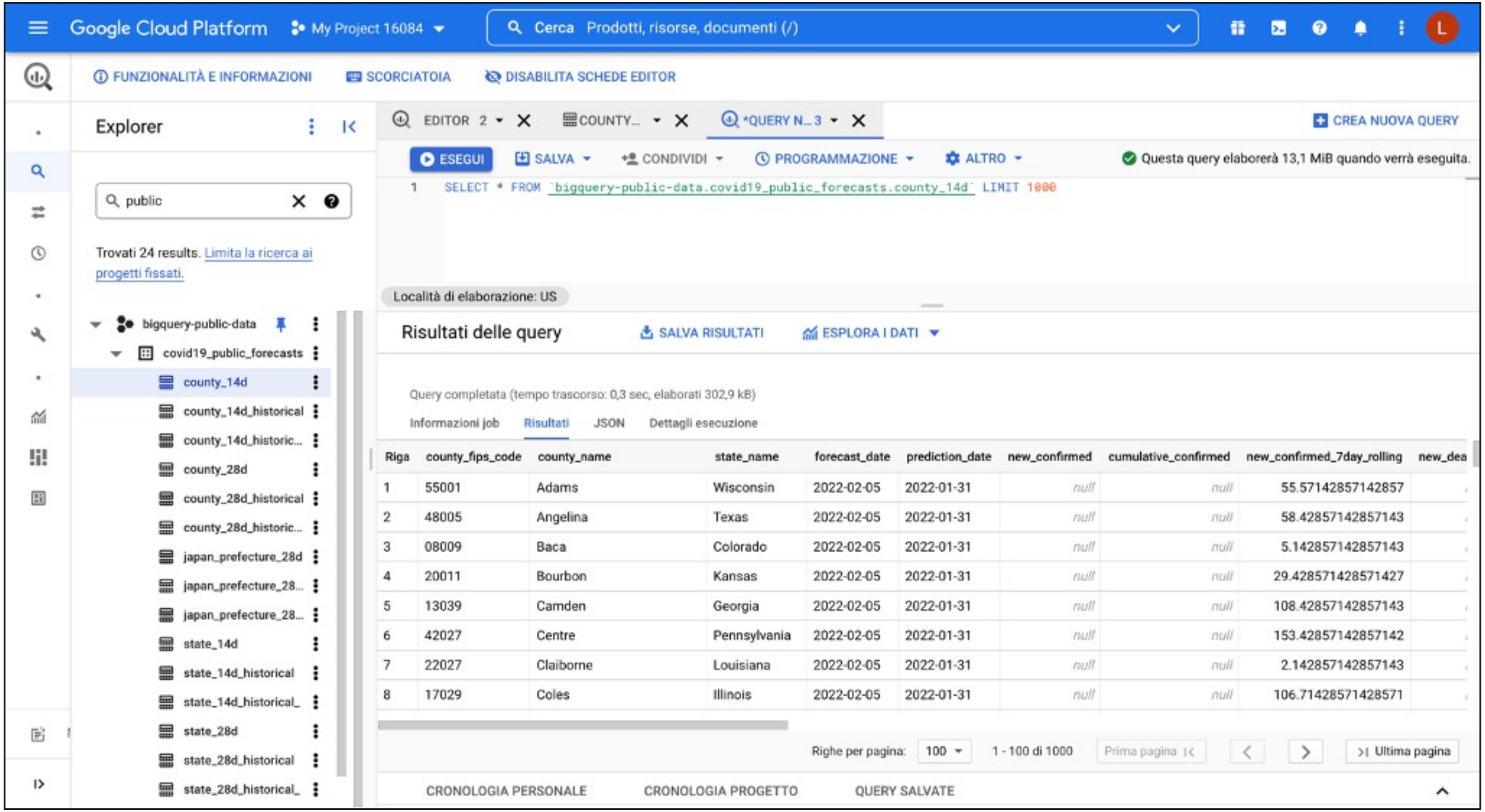
3 months

## Object of the Internship

Google Cloud Platform Study

Database analysis and cleanup

# BigQuery



public

Trovati 24 results. Limita la ricerca ai progetti fissati.

- bigquery-public-data
  - covid19_public_forecasts
    - county_14d
    - county_14d_historical
    - county_14d_historic...
    - county_28d
    - county_28d_historical
    - county_28d_historic...
    - japan_prefecture_28d
    - japan_prefecture_28...
    - japan_prefecture_28...
    - state_14d
    - state_14d_historical
    - state_14d_historical_
    - state_28d
    - state_28d_historical
    - state_28d_historical_

Questa query elaborerà 13,1 MiB quando verrà eseguita.

```
SELECT * FROM `bigquery-public-data.covid19_public_forecasts.county_14d` LIMIT 1000
```

Località di elaborazione: US

Risultati delle query

Query completata (tempo trascorso: 0,3 sec, elaborati 302,9 kB)

| Riga | county_fips_code | county_name | state_name | forecast_date | prediction_date | new_confirmed | cumulative_confirmed | new_confirmed_7day_rolling | new_dea |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 55001 | Adams | Wisconsin | 2022-02-05 | 2022-01-31 | null | null | 55.57142857142857 | |
| 2 | 48005 | Angelina | Texas | 2022-02-05 | 2022-01-31 | null | null | 58.42857142857143 | |
| 3 | 08009 | Baca | Colorado | 2022-02-05 | 2022-01-31 | null | null | 5.142857142857143 | |
| 4 | 20011 | Bourbon | Kansas | 2022-02-05 | 2022-01-31 | null | null | 29.428571428571427 | |
| 5 | 13039 | Camden | Georgia | 2022-02-05 | 2022-01-31 | null | null | 108.42857142857143 | |
| 6 | 42027 | Centre | Pennsylvania | 2022-02-05 | 2022-01-31 | null | null | 153.42857142857142 | |
| 7 | 22027 | Claiborne | Louisiana | 2022-02-05 | 2022-01-31 | null | null | 2.142857142857143 | |
| 8 | 17029 | Coles | Illinois | 2022-02-05 | 2022-01-31 | null | null | 106.71428571428571 | |

Righe per pagina: 100 1 - 100 di 1000

|  |  |  |
|---|---|---|
| ID | INTEGER | NULLABLE |
| CREATED_DATE | DATETIME | NULLABLE |
| BARCODE | STRING | **REQUIRED** |
| AGENCY_ID | INTEGER | **REQUIRED** |
| AGENCY_NAME | STRING | NULLABLE |
| NOTE | STRING | NULLABLE |
| RECIPIENT_TYPE | STRING | NULLABLE |
| DOCUMENT_TYPE | STRING | NULLABLE |
| DOCUMENT_NUMBER | STRING | NULLABLE |
| PRODUCT_TYPE | STRING | **REQUIRED** |
| ↓ STATE | RECORD | REPEATED |
| NAME | STRING | **REQUIRED** |
| SEGNCOD | STRING | NULLABLE |
| USER_ID | INTEGER | NULLABLE |
| USER_USERNAME | STRING | **REQUIRED** |
| CREATED_DATE | DATETIME | NULLABLE |
| DELIVERY_SEND_STATE | STRING | NULLABLE |
| DELIVERY_CREATED_DATE | DATETIME | NULLABLE |
| DELIVERY_LATITUDE | FLOAT | NULLABLE |
| DELIVERY_LONGITUDE | FLOAT | NULLABLE |

# DATABASE FEATURES

- ❖ Client:
  MySQL8
- ❖ Purpose:
  Mail tracking
- ❖ Structure:
  1 nested table

## MAIN TABLE

| | |
|---|---|
| ID | Identification code of each letter received, less reliable than the barcode since it may be missing. |
| CREATED_DATE | Date of receipt of the letter. |
| BARCODE | It refers to the barcode of the letter and is the primary identification code. |
| AGENCY_ID | Identification code of the agency that accepted the letter. |
| AGENCY_NAME | The name of the agency that accepted the letter or package. |
| NOTE | Any notes. |
| RECIPIENT_TYPE | Recipient of the letter (e.g. recipient, relative, delegate, institution, etc.). |
| DOCUMENT_TYPE | The type of document shown upon receipt (if necessary). |
| DOCUMENT_NUMBER | Document code shown upon receipt (if necessary). |
| PRODUCT_TYPE | Type of letter (sdoc, parcel, registered letter) |

## NESTED TABLE

| | |
|---|---|
| ↓ STATE | |
| NAME | Identify whether the letter is being accepted or delivered |
| SEGNCOD | Identify the delivery method or the reason for a non-delivery |
| USER_ID | Postman identification code |
| USER_USERNAME | Postman's identification name |
| CREATED_DATE | Date of movement indicated in the line (acceptance or attempted delivery) |
| DELIVERY_SEND_STATE | Identify if the letter was sent or if an error occurred |
| DELIVERY_CREATED_DATE | Date the letter was delivered |
| DELIVERY_LATITUDE | The two scopes, latitude and longitude, refer to the location of the delivery |
| DELIVERY_LONGITUDE | |

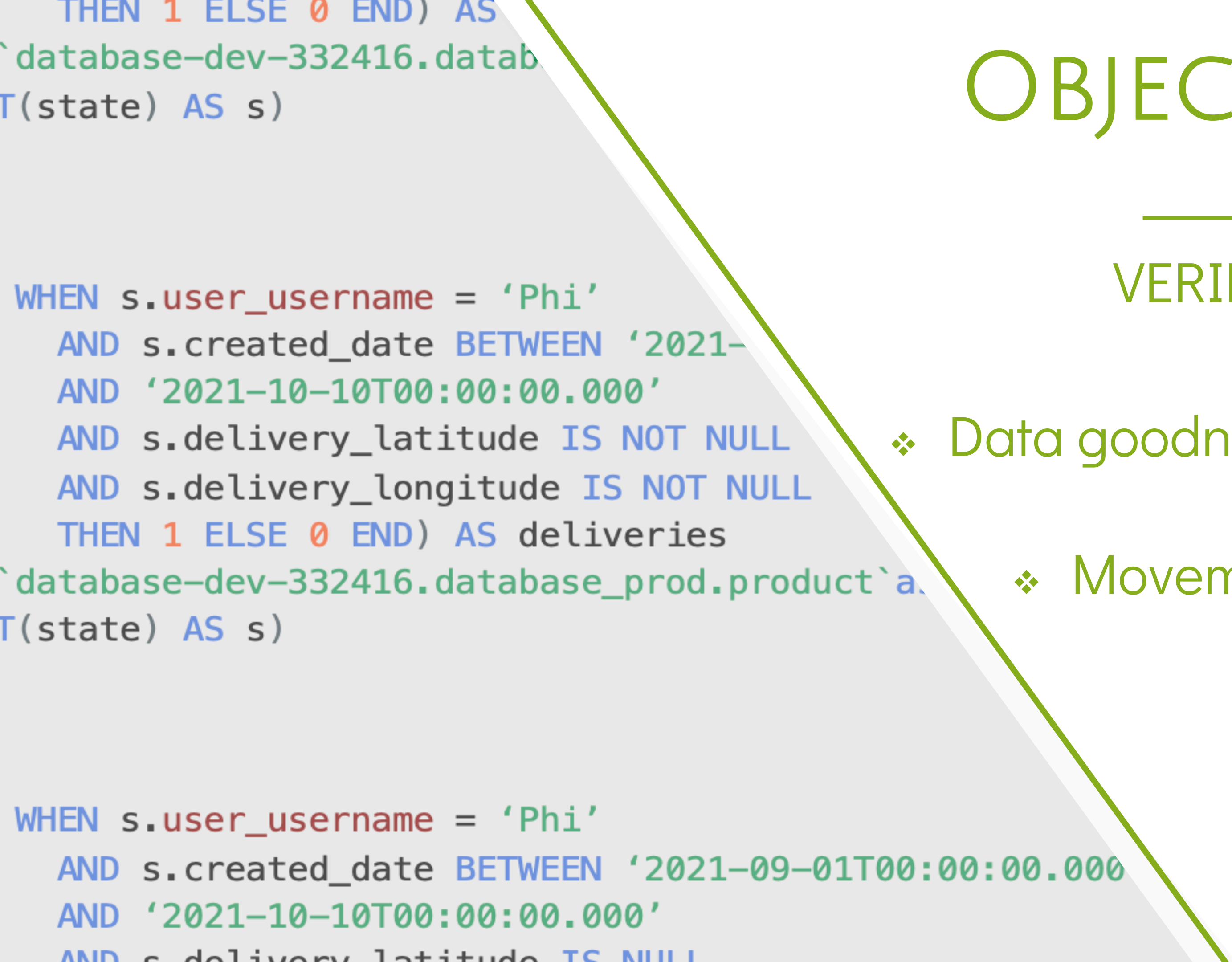
# MAIN TABLE

| | |
|---|---|
| ID | Identification code of each letter received, less reliable than the barcode since it may be missing. |
| CREATED_DATE | Date of receipt of the letter. |
| BARCODE | It refers to the barcode of the letter and is the primary identification code. |
| AGENCY_ID | Identification code of the agency that accepted the letter. |
| AGENCY_NAME | The name of the agency that accepted the letter or package. |
| NOTE | Any notes. |
| RECIPIENT_TYPE | Recipient of the letter (e.g. recipient, relative, delegate, institution, etc.). |
| DOCUMENT_TYPE | The type of document shown upon receipt (if necessary). |
| DOCUMENT_NUMBER | Document code shown upon receipt (if necessary). |
| PRODUCT_TYPE | Type of letter (sdoc, parcel, registered letter) |

# NESTED TABLE

↓ STATE

| | |
|---|---|
| NAME | Identify whether the letter is being accepted or delivered |
| SEGNCOD | Identify the delivery method or the reason for a non-delivery |
| USER_ID | Postman identification code |
| USER_USERNAME | Postman's identification name |
| CREATED_DATE | Date of movement indicated in the line (acceptance or attempted delivery) |
| DELIVERY_SEND_STATE | Identify if the letter was sent or if an error occurred |
| DELIVERY_CREATED_DATE | Date the letter was delivered |
| DELIVERY_LATITUDE | The two scopes, latitude and longitude, refer to the location of the delivery |
| DELIVERY_LONGITUDE | |



```
    THEN 1 ELSE 0 END) AS
`database-dev-332416.datab
T(state) AS s)



 WHEN s.user_username = ‘Phi’
    AND s.created_date BETWEEN ‘2021-
    AND ‘2021-10-10T00:00:00.000’
    AND s.delivery_latitude IS NOT NULL
    AND s.delivery_longitude IS NOT NULL
    THEN 1 ELSE 0 END) AS deliveries
`database-dev-332416.database_prod.product`a
T(state) AS s)



 WHEN s.user_username = ‘Phi’
    AND s.created_date BETWEEN ‘2021-09-01T00:00:00.000
    AND ‘2021-10-10T00:00:00.000’
    AND s.delivery_latitude IS NULL
```

# OBJECTIVE

VERIFY:

- Data goodness
- Movement tracking

# Exploratory analysis

1.261

Number of postmen

361

Days of activity

16.677.672

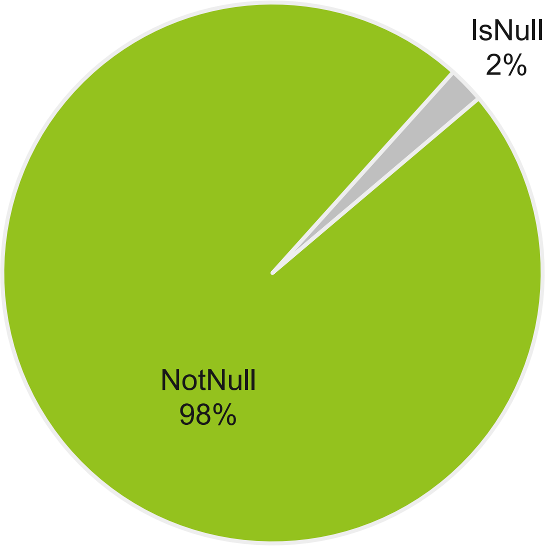
Total barcodes managed

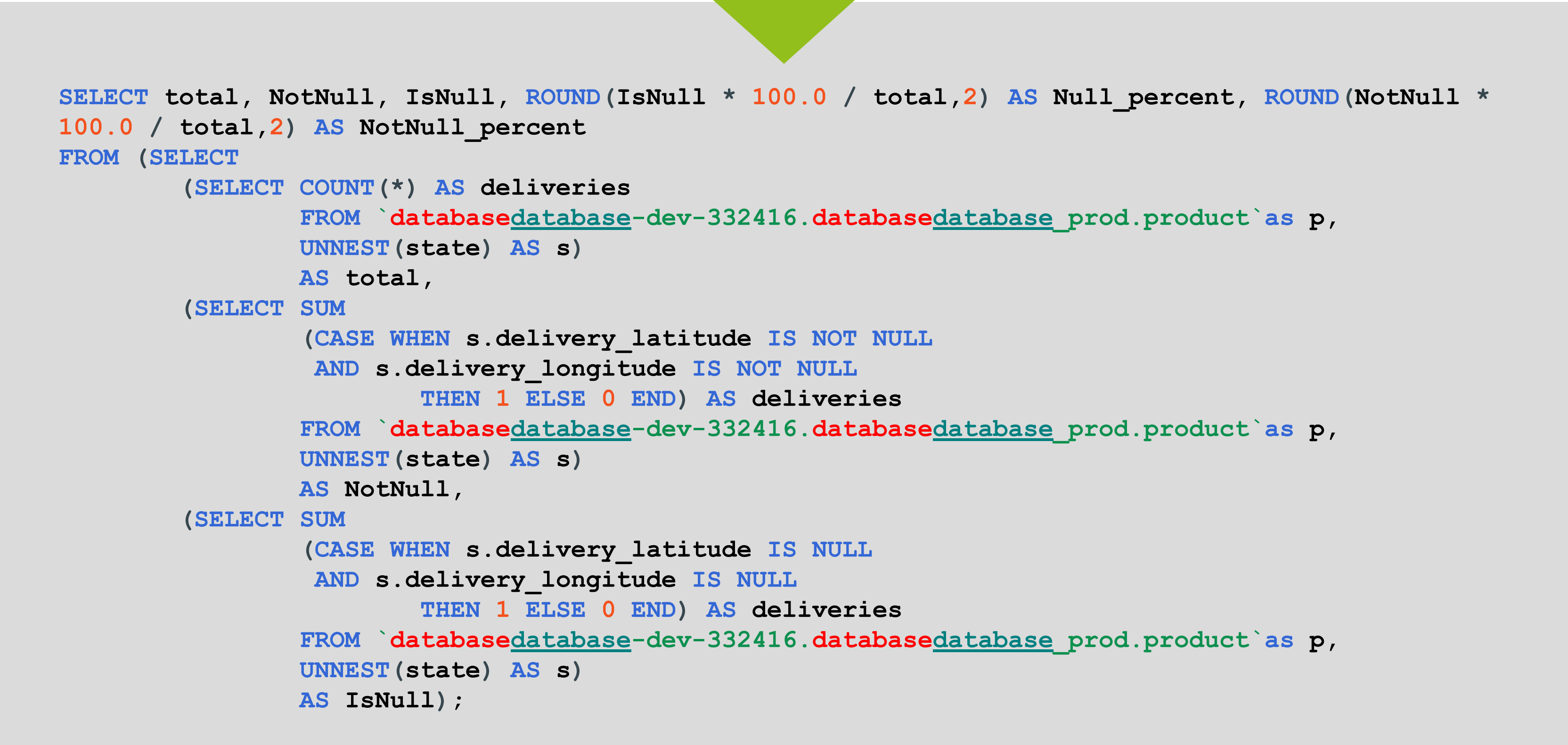
# MISSING DATA ANALYSIS

- Latitude and longitude
- created_date
- state.created_date
-

| total | NotNull | IsNull |
|---|---|---|
| 17672087 | 17290203 | 381884 |

| Null_percent | NotNull_percent |
|---|---|
| 2.16 | 97.84 |



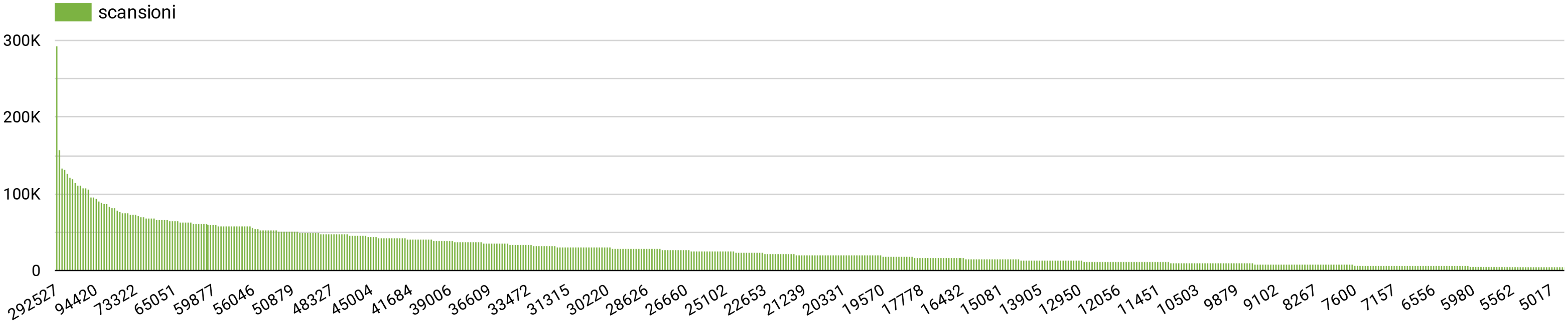


```
SELECT total, NotNull, IsNull, ROUND(IsNull * 100.0 / total,2) AS Null_percent, ROUND(NotNull *
100.0 / total,2) AS NotNull_percent
FROM (SELECT
        (SELECT COUNT(*) AS deliveries
                FROM `databasedatabase-dev-332416.databasedatabase_prod.product`as p,
                UNNEST(state) AS s)
                AS total,
        (SELECT SUM
                (CASE WHEN s.delivery_latitude IS NOT NULL
                 AND s.delivery_longitude IS NOT NULL
                        THEN 1 ELSE 0 END) AS deliveries
                FROM `databasedatabase-dev-332416.databasedatabase_prod.product`as p,
                UNNEST(state) AS s)
                AS NotNull,
        (SELECT SUM
                (CASE WHEN s.delivery_latitude IS NULL
                 AND s.delivery_longitude IS NULL
                        THEN 1 ELSE 0 END) AS deliveries
                FROM `databasedatabase-dev-332416.databasedatabase_prod.product`as p,
                UNNEST(state) AS s)
                AS IsNull);
```

# MISSING DATA ANALYSIS

# Out-of-scale values

## Total scans per postman

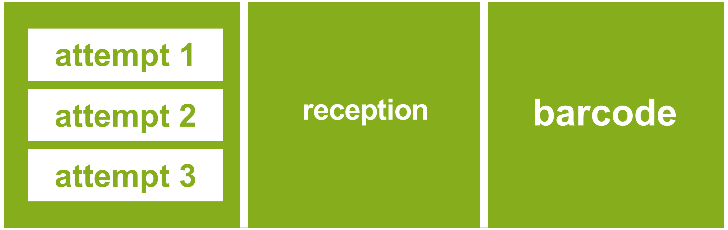


Daily scans for each postman

All scans for single postman

Monthly and weekly scans

# THE PROBLEM OF THE UNNEST



attempt 1

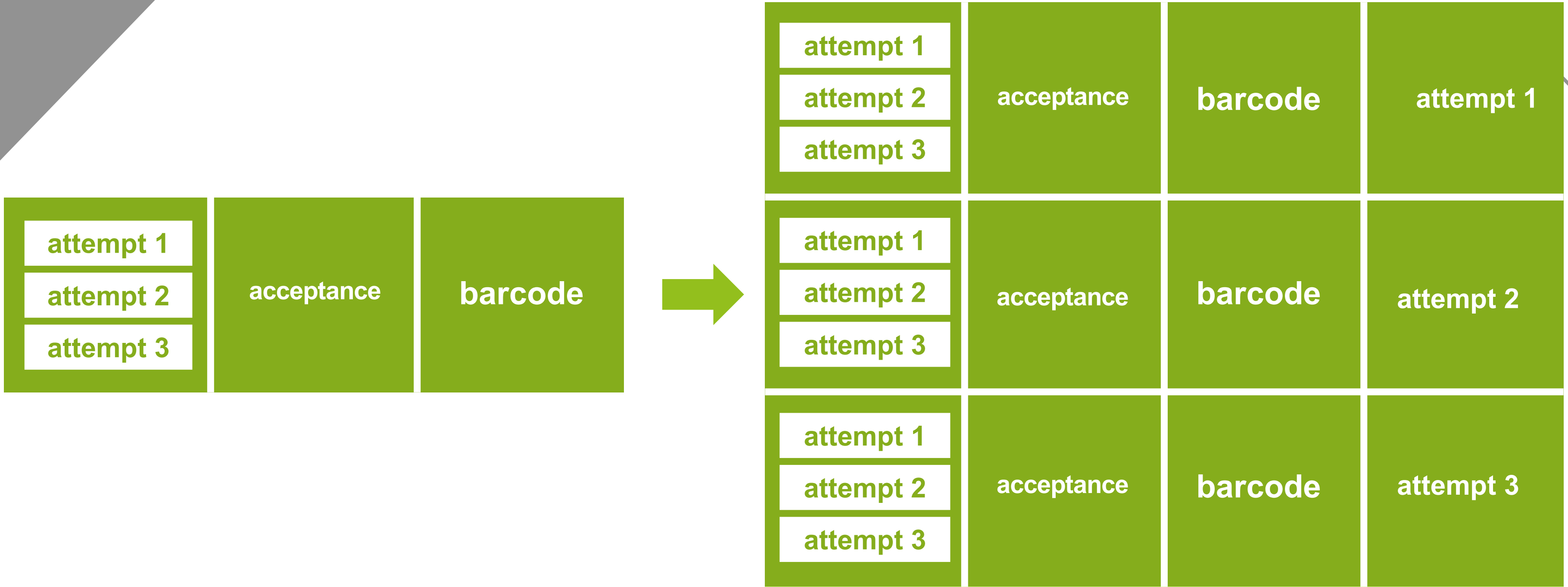
attempt 2

attempt 3

reception

barcode

# THE PROBLEM OF THE UNNEST



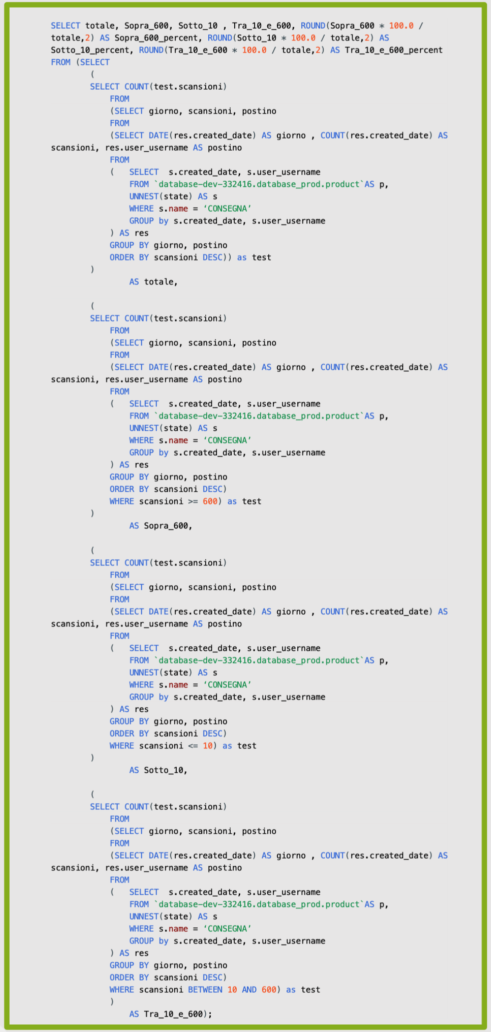
| | | |
|---|---|---|
| attempt 1<br>attempt 2<br>attempt 3 | acceptance | barcode |

| | | | |
|---|---|---|---|
| attempt 1<br>attempt 2<br>attempt 3 | acceptance | barcode | attempt 1 |
| attempt 1<br>attempt 2<br>attempt 3 | acceptance | barcode | attempt 2 |
| attempt 1<br>attempt 2<br>attempt 3 | acceptance | barcode | attempt 3 |

```
SELECT DATE(res.created_date) AS giorno , COUNT(res.created_date) AS
scansioni, res.user_username AS postino
FROM
(
    SELECT  s.created_date, s.user_username
    FROM `databasedatabase-dev-332416.databasedatabase_prod.product`AS p,
    UNNEST(state) AS s
    WHERE s.name = '`CONSEGNA'’
    GROUP by s.created_date, s.user_username
) AS res
GROUP BY giorno, postino
ORDER BY scansioni DESC;
```

# UNNEST SOLUTION

# PULIZIA DEI DATI

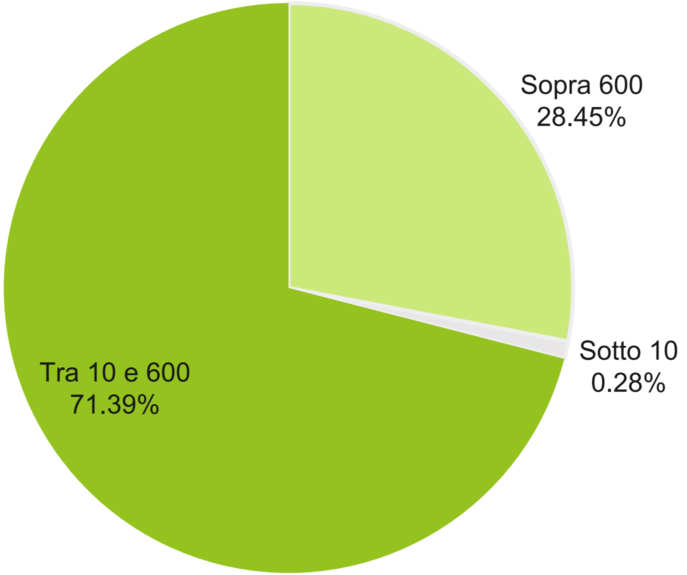


```sql
SELECT totale, Sopra_600, Sotto_10 , Tra_10_e_600, ROUND(Sopra_600 * 100.0 /
totale,2) AS Sopra_600_percent, ROUND(Sotto_10 * 100.0 / totale,2) AS
Sotto_10_percent, ROUND(Tra_10_e_600 * 100.0 / totale,2) AS Tra_10_e_600_percent
FROM (SELECT
        (
        SELECT COUNT(test.scansioni)
            FROM
            (SELECT giorno, scansioni, postino
            FROM
            (SELECT DATE(res.created_date) AS giorno , COUNT(res.created_date) AS
scansioni, res.user_username AS postino
            FROM
            (   SELECT  s.created_date, s.user_username
                FROM `database-dev-332416.database_prod.product`AS p,
                UNNEST(state) AS s
                WHERE s.name = 'CONSEGNA'
                GROUP by s.created_date, s.user_username
            ) AS res
            GROUP BY giorno, postino
            ORDER BY scansioni DESC)) as test
        )
                AS totale,

        (
        SELECT COUNT(test.scansioni)
            FROM
            (SELECT giorno, scansioni, postino
            FROM
            (SELECT DATE(res.created_date) AS giorno , COUNT(res.created_date) AS
scansioni, res.user_username AS postino
            FROM
            (   SELECT  s.created_date, s.user_username
                FROM `database-dev-332416.database_prod.product`AS p,
                UNNEST(state) AS s
                WHERE s.name = 'CONSEGNA'
                GROUP by s.created_date, s.user_username
            ) AS res
            GROUP BY giorno, postino
            ORDER BY scansioni DESC)
            WHERE scansioni >= 600) as test
        )
                AS Sopra_600,

        (
        SELECT COUNT(test.scansioni)
            FROM
            (SELECT giorno, scansioni, postino
            FROM
            (SELECT DATE(res.created_date) AS giorno , COUNT(res.created_date) AS
scansioni, res.user_username AS postino
            FROM
            (   SELECT  s.created_date, s.user_username
                FROM `database-dev-332416.database_prod.product`AS p,
                UNNEST(state) AS s
                WHERE s.name = 'CONSEGNA'
                GROUP by s.created_date, s.user_username
            ) AS res
            GROUP BY giorno, postino
            ORDER BY scansioni DESC)
            WHERE scansioni <= 10) as test
        )
                AS Sotto_10,

        (
        SELECT COUNT(test.scansioni)
            FROM
            (SELECT giorno, scansioni, postino
            FROM
            (SELECT DATE(res.created_date) AS giorno , COUNT(res.created_date) AS
scansioni, res.user_username AS postino
            FROM
            (   SELECT  s.created_date, s.user_username
                FROM `database-dev-332416.database_prod.product`AS p,
                UNNEST(state) AS s
                WHERE s.name = 'CONSEGNA'
                GROUP by s.created_date, s.user_username
            ) AS res
            GROUP BY giorno, postino
            ORDER BY scansioni DESC)
            WHERE scansioni BETWEEN 10 AND 600) as test
            )
                AS Tra_10_e_600);
```



| Totale | Sopra_600 | Sotto_10 | Tra_10_e_600 |
|---|---|---|---|
| 14930960 | 4247380 | 41123 | 10659787 |

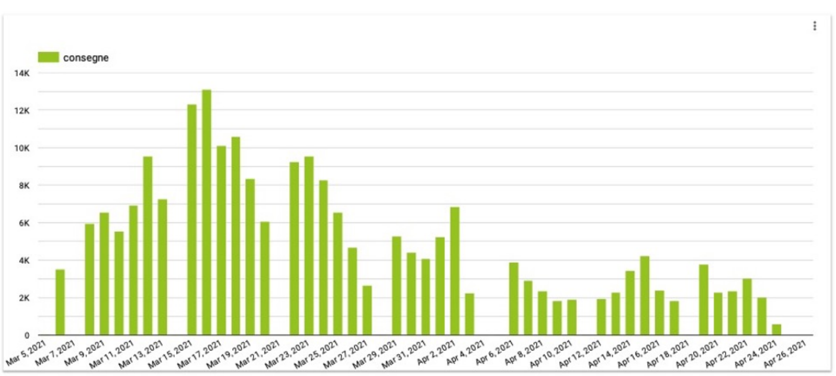
| Sopra_600_percent | Sotto_10_percent | Tra_10_e_600_percent |
|---|---|---|
| 28.45 | 0.28 | 71.39 |

```
SELECT COUNT(test.scansioni)
            FROM
            (SELECT giorno, scansioni, postino
            FROM
            (SELECT DATE(res.created_date) AS giorno , COUNT(res.created_date) AS
scansioni, res.user_username AS postino
            FROM
            (   SELECT  s.created_date, s.user_username
                FROM `databasedatabase-dev-332416.databasedatabase_prod.product`AS p,
                UNNEST(state) AS s
                WHERE s.name = 'CONSEGNA'
                GROUP by s.created_date, s.user_username
            ) AS res
            GROUP BY giorno, postino
            ORDER BY scansioni DESC)
            WHERE scansioni BETWEEN 10 AND 600) as test
```

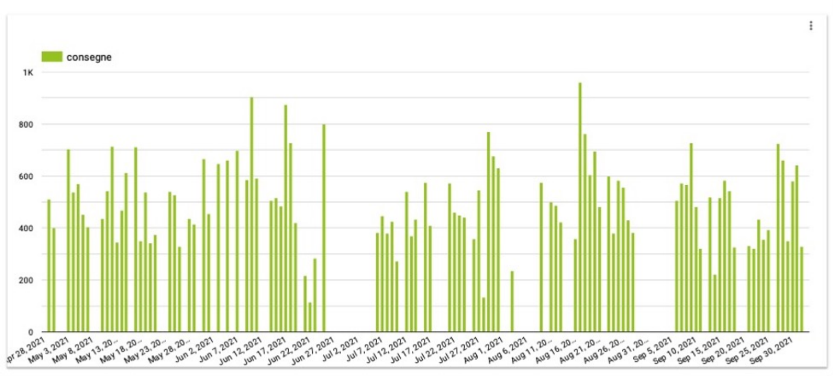
# Code excerpt

# Daily scans

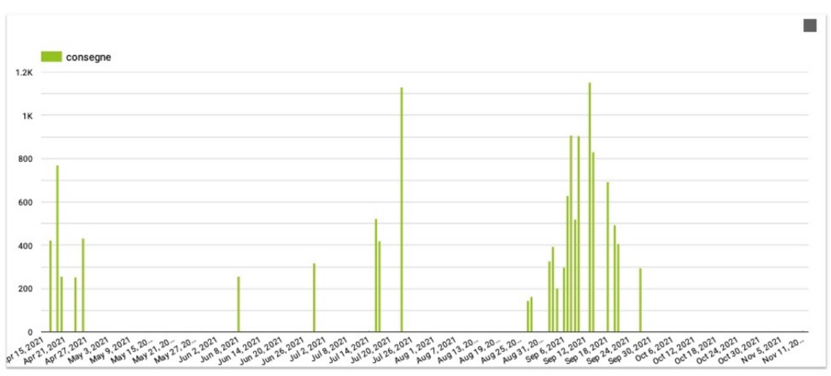
## Postman lambda



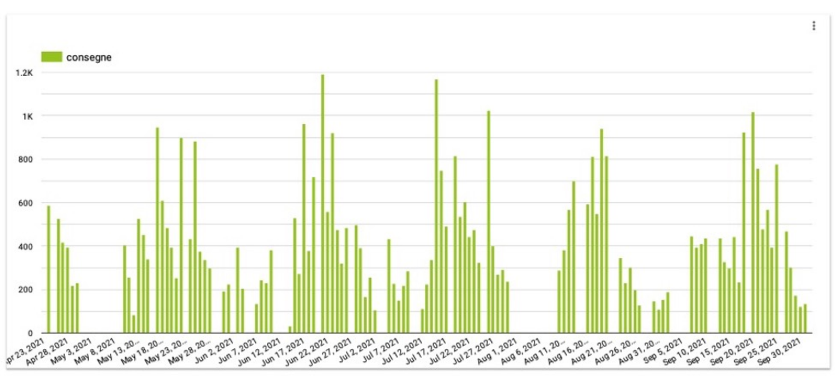
## Postman MI



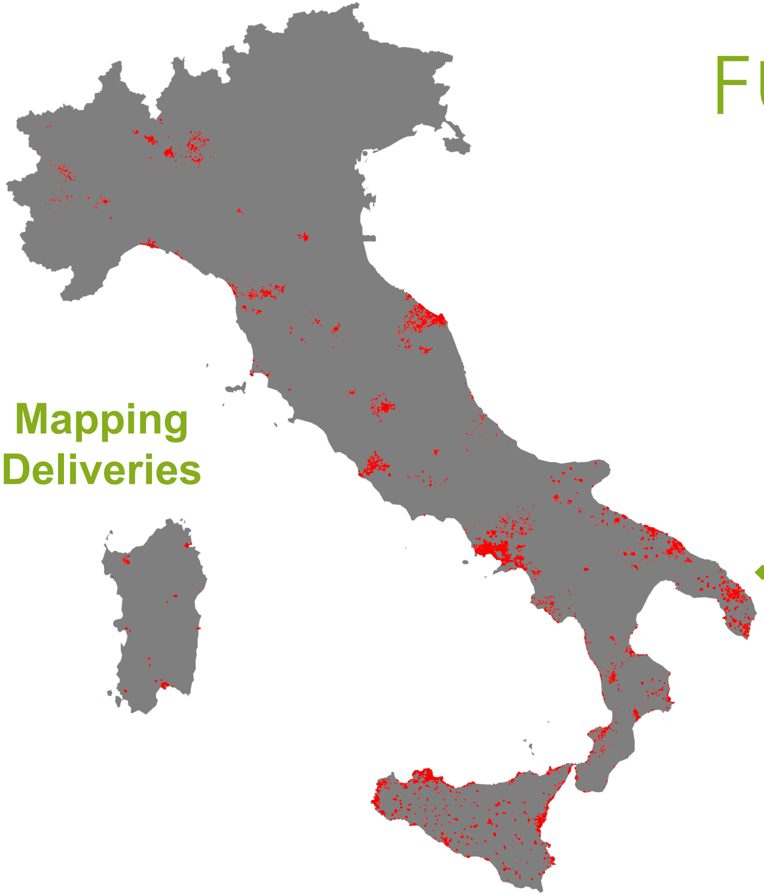
## Postman CHI



## Postman NI



# Future developments



- ❖ Route extraction
- ❖ Optimal route generation
- ❖ Timing calculation
- ❖ Analysis of the most active areas
- ❖ Growth trend analysis
- ❖ Economic growth comparison

# THANKS