Query #1

A screenshot of a computer

Description automatically generated

INSERT INTO public."queryOne\_brands"(

brand1, brand2, year, customer1, customer2, customer3, customer4)

VALUES

('alex', 'ava', '2023', 2, 2, 2, 2),

('ava', 'alex', '2023', 2, 2, 2, 2),

('apple', 'samsung', '2020', 1, 2, 1, 2),

('samsung', 'apple', '2020', 1, 2, 1, 2),

('apple', 'samsung', '2021', 1, 2, 5, 3),

('samsung', 'apple', '2021', 5, 3, 1, 2),

('google', '', '2020', 5, 9,0,0 ),

('oneplus', 'nothing', '2020', 5, 9, 6, 3)

;

select \* from public."queryOne\_brands";

with pairs\_year as(

select \*,case when brand1<brand2 then concat (brand1,brand2,year)

else concat(brand2,brand1,year) end as mark\_value

from public."queryOne\_brands" a

),

brand\_rn as (

select \* ,row\_number()over(partition by mark\_value) rn

from pairs\_year

)

select brand1,brand2,year,customer1,customer2, customer3, customer4

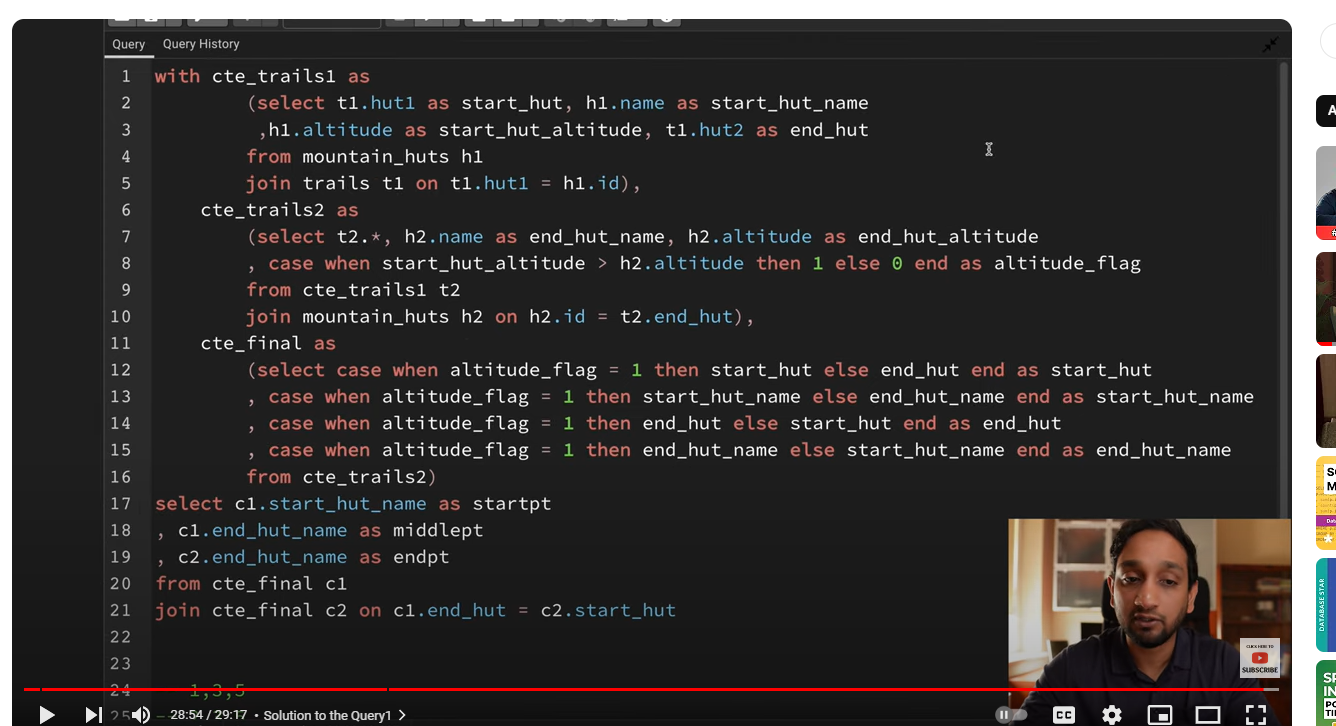
from brand\_rn where rn=1 or (customer1<>customer3 and customer2<>customer4);

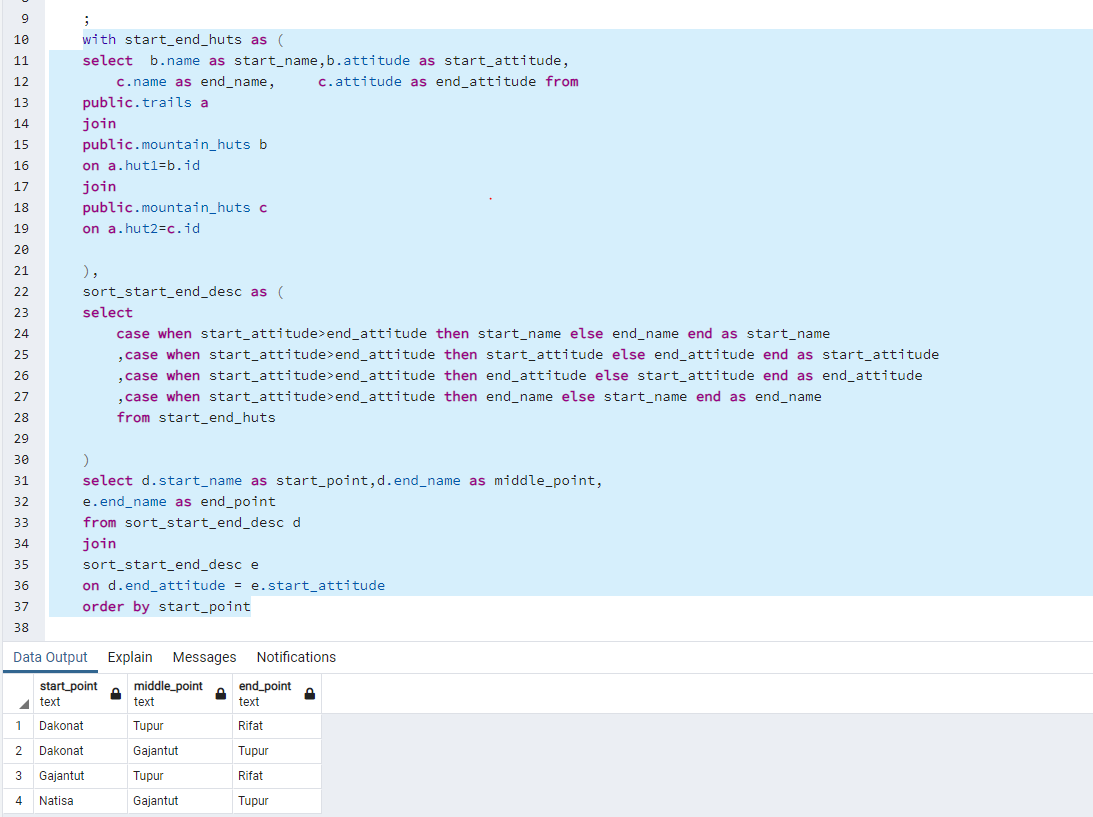
Query2:



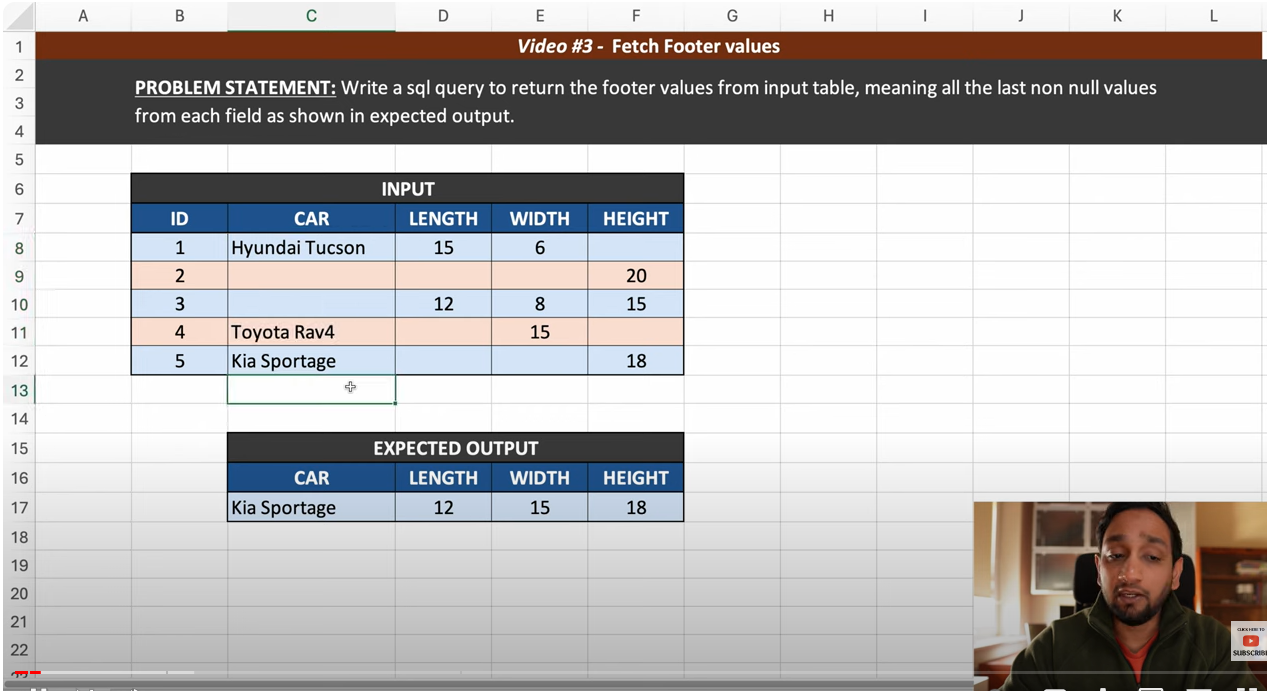
A screenshot of a computer

Description automatically generated





Query 3:



Based on your current role and skills, here's a list of areas you might consider learning or improving:

1. \*\*Advanced SQL:\*\* Since you're already working with SQL, enhancing your skills in this area could make you much more effective at testing and analyzing data. SQL is used heavily in the data world, and more complex knowledge such as stored procedures, triggers, and performance tuning can benefit you greatly.

2. \*\*Learn More About Big Data Technologies:\*\* Besides Spark, which you're already using, you might want to learn more about other big data technologies such as Hadoop, Hive, Pig, HBase, or Kafka. These technologies are commonly used in the big data ecosystem, and understanding them can make you more versatile as a data quality tester.

3. \*\*Data Warehousing Concepts:\*\* Understanding concepts such as ETL processes, data modeling, OLAP vs OLTP, star schema, and snowflake schema can help when working with big data systems.

4. \*\*Python Programming:\*\* Since you mentioned using PySpark and Pandas, you're probably already familiar with Python. Nevertheless, diving deeper into Python, especially aspects related to data manipulation and analysis (like NumPy, SciPy, Matplotlib for data visualization, etc.) could prove to be beneficial. Also, understanding Python's usage in the automation of testing scenarios might be helpful.

5. \*\*Data Visualization Tools:\*\* Tools like Tableau, Power BI, or QlikView can help you visualize data in a more digestible format, which is typically useful in noticing trends or anomalies.

6. \*\*Testing Tools and Frameworks:\*\* There are many tools and frameworks out there that can help automate and streamline the testing process. You might consider learning about tools such as JUnit, Postman (for API testing), Selenium (for web application testing), or Jest (for JavaScript testing).

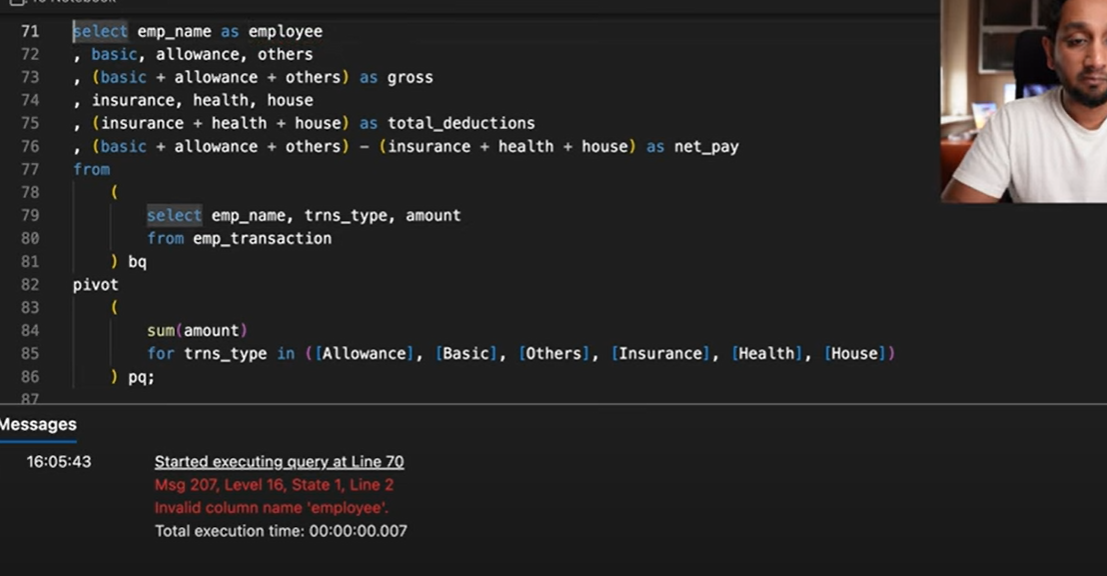
7. \*\*Understanding Machine Learning:\*\* As big data and machine learning are closely intertwined, having a basic understanding of machine learning concepts can often be beneficial in a big data career.

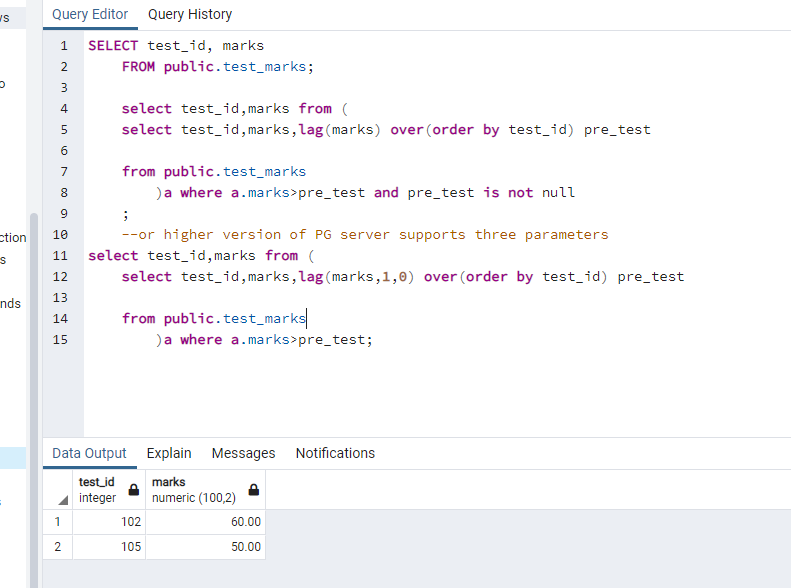
Remember, the path to learning these skills depends on your goals, the needs of your projects, or the technologies your organization is invested in. It's a good practice to discuss this with your manager or team lead to align your learning path with the team's requirements.

Firstly  
 create extension if not exists tablefunc;

A screen shot of a computer program

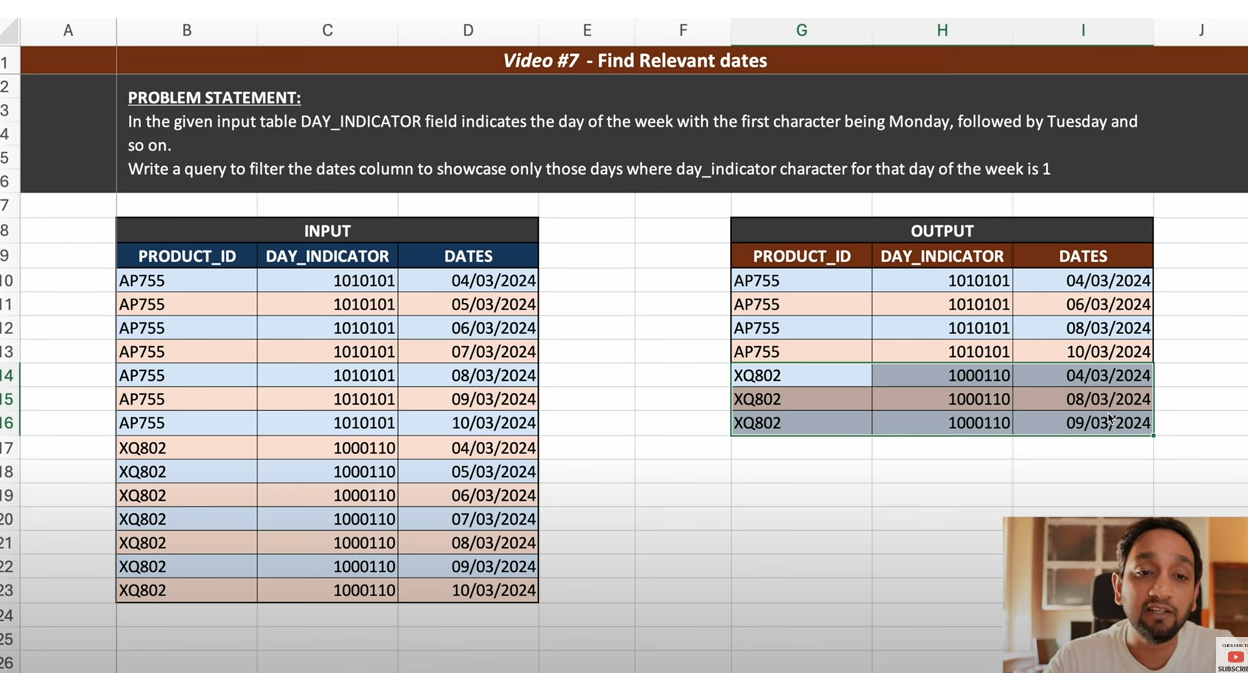
Description automatically generated

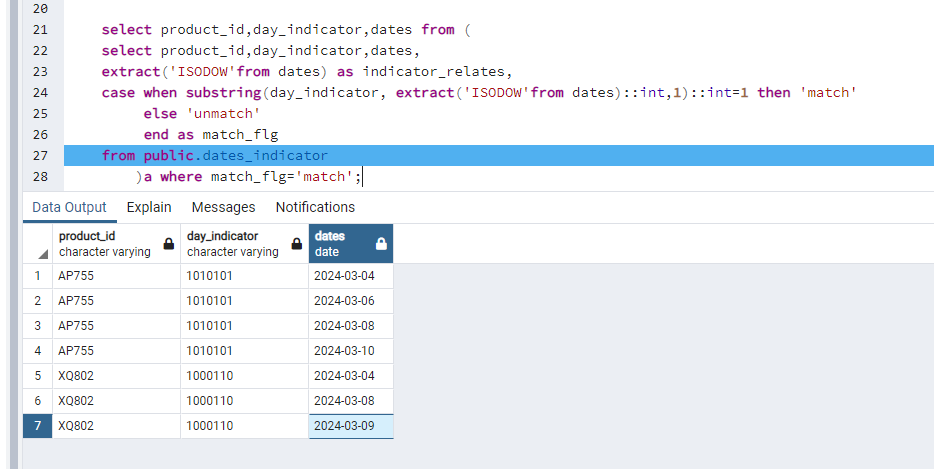


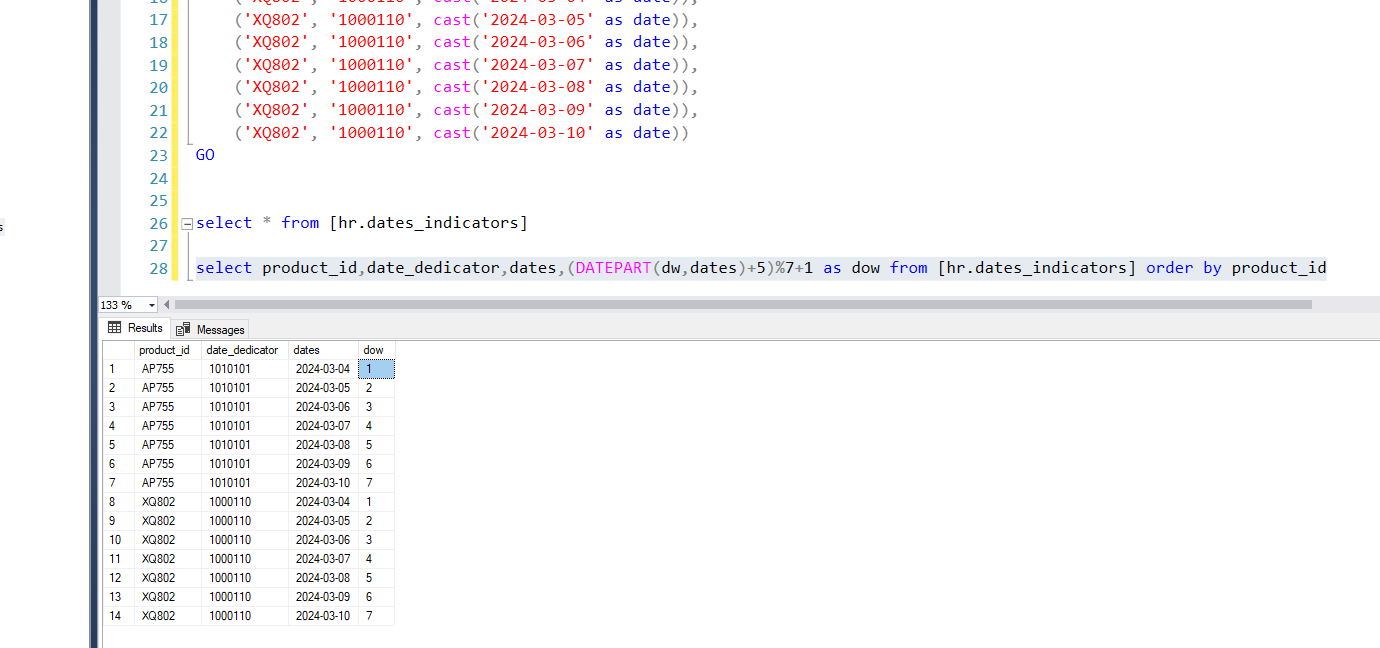


A screenshot of a computer

Description automatically generated

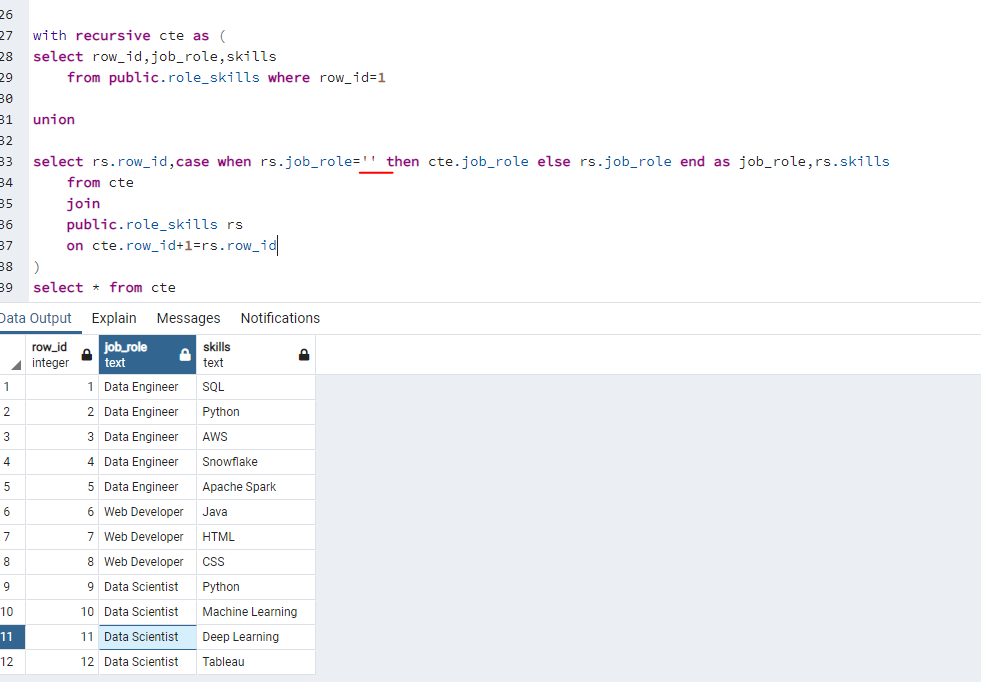


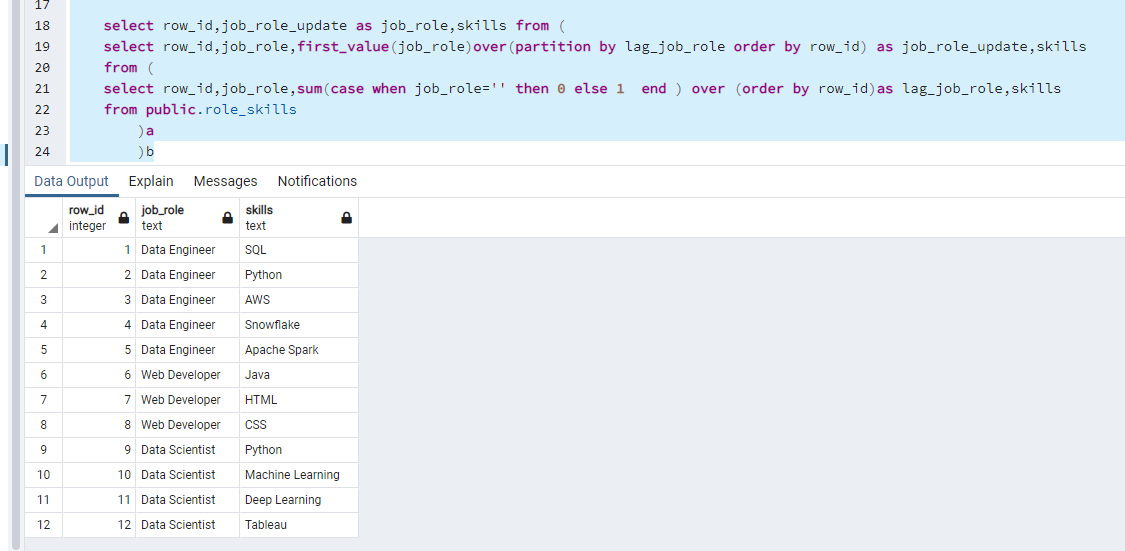




A screenshot of a computer

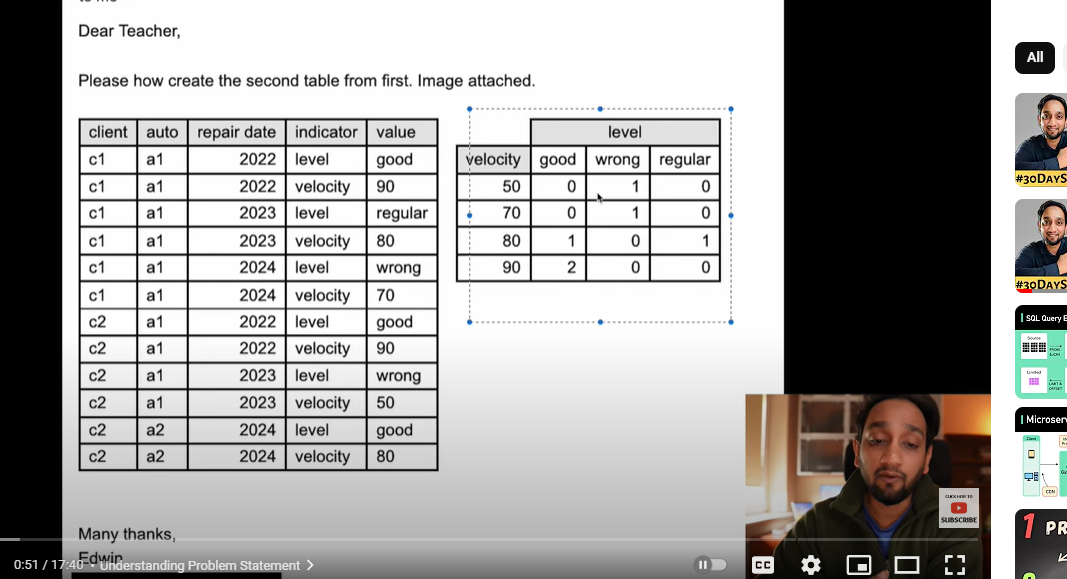
Description automatically generated



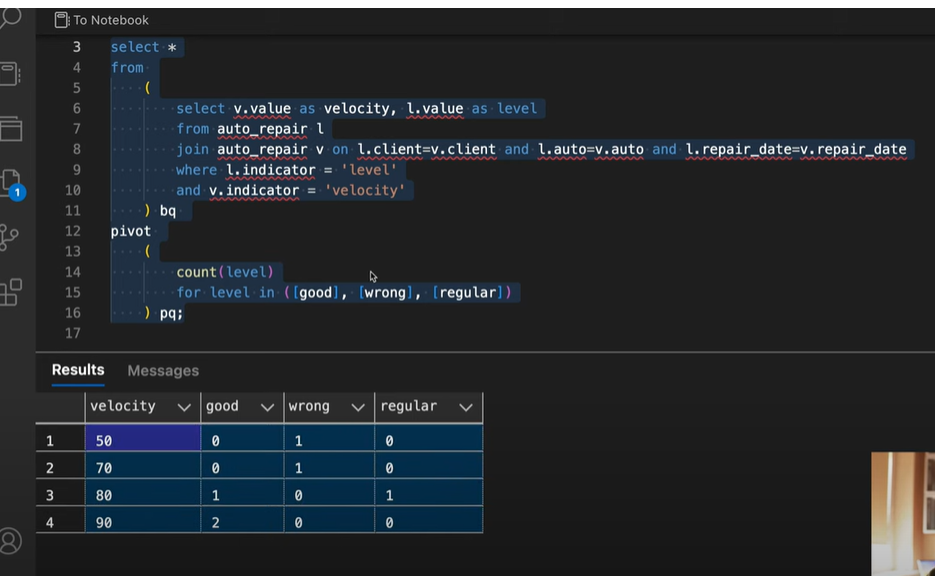


A screenshot of a computer

Description automatically generated



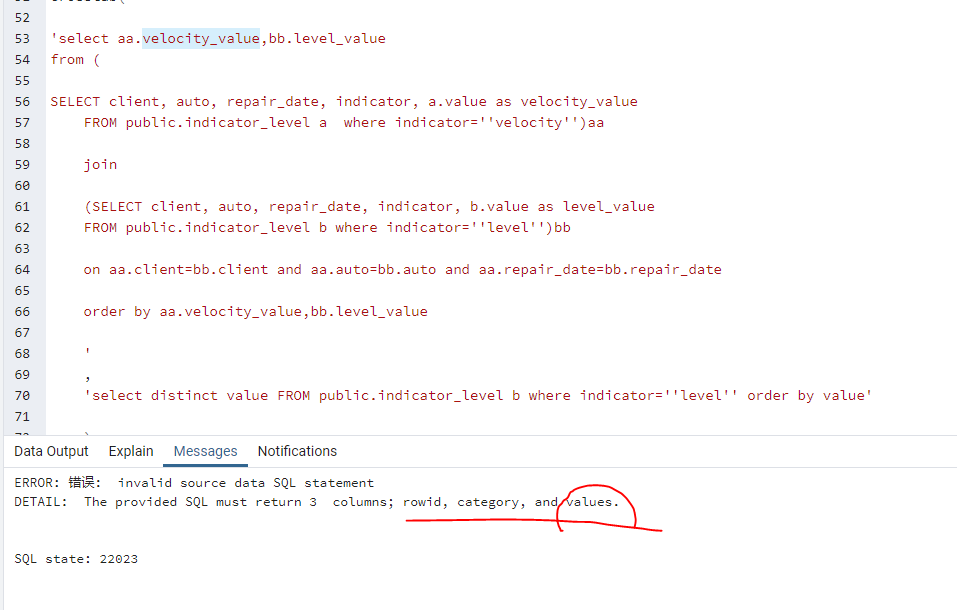
Using sqlserver pivot()

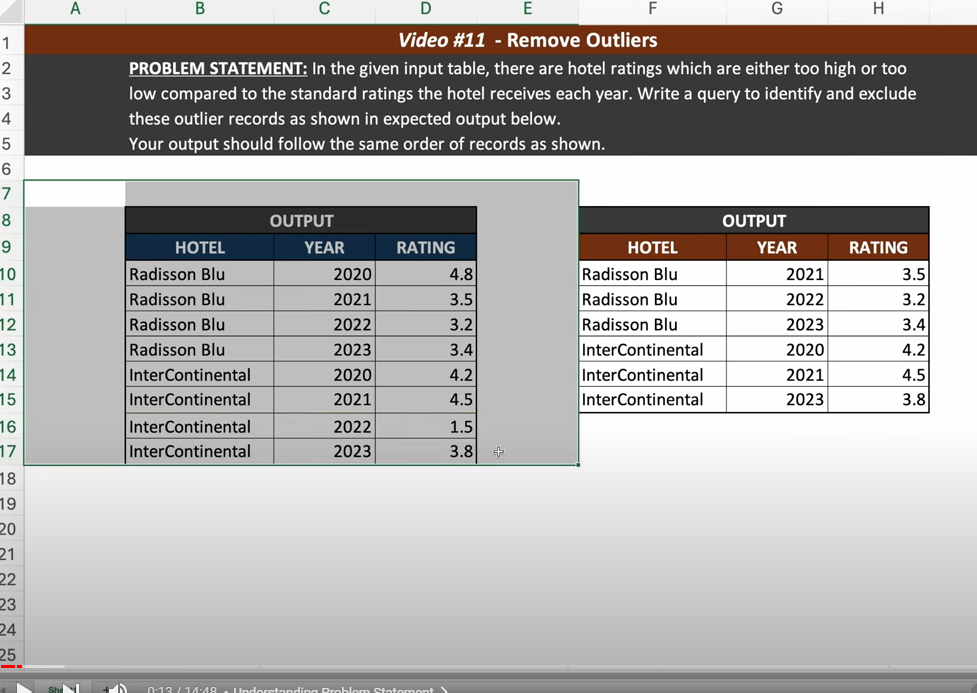


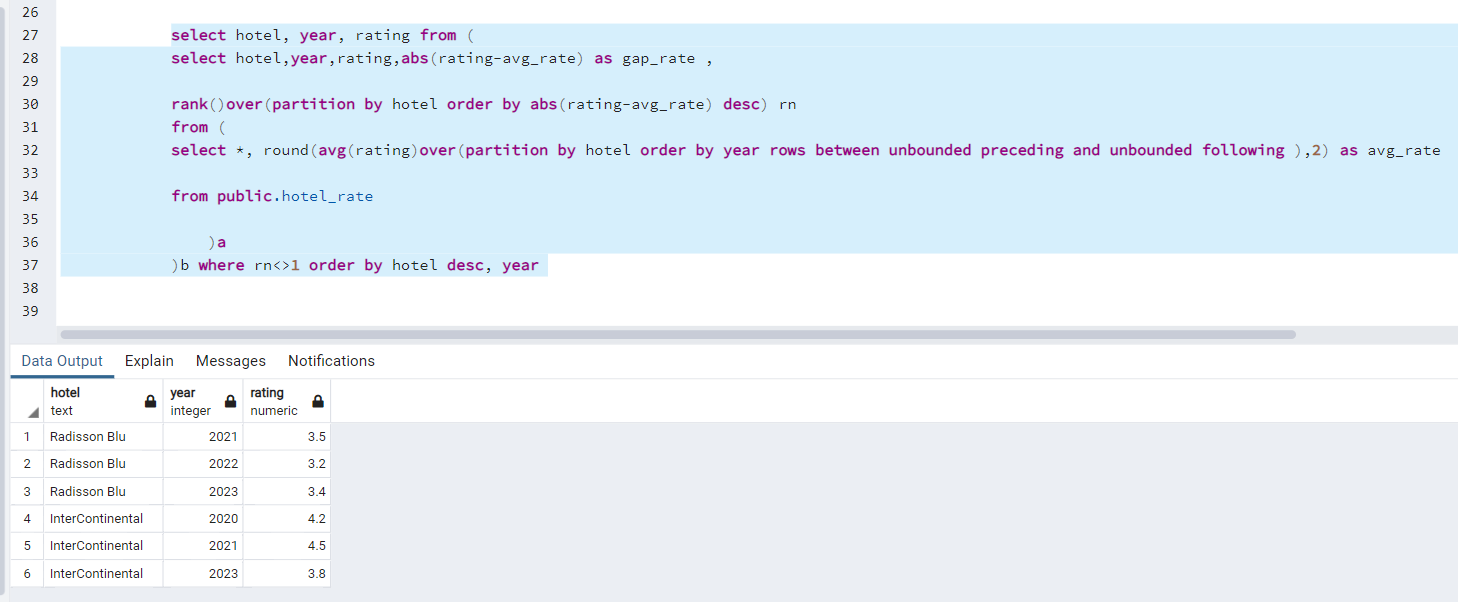
PG crosstab()

A screen shot of a computer

Description automatically generated

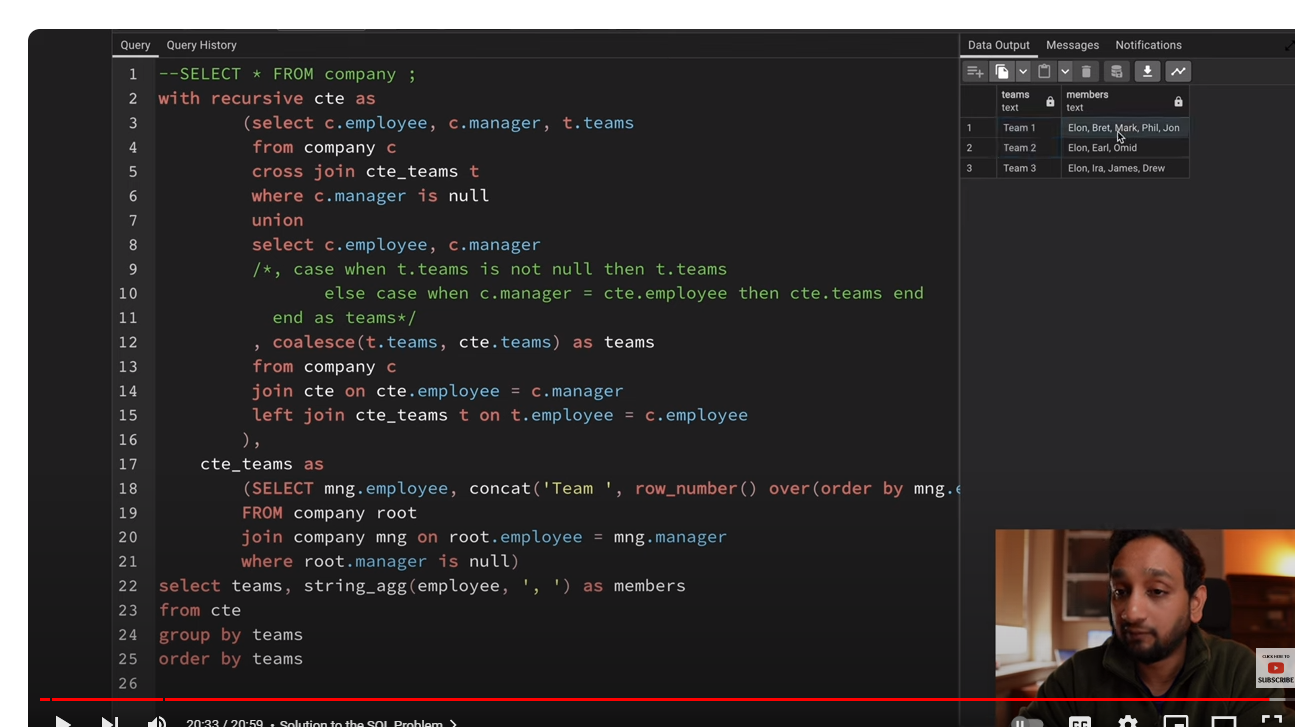








Observe the feature of the output records, found that the sequence is from up to end



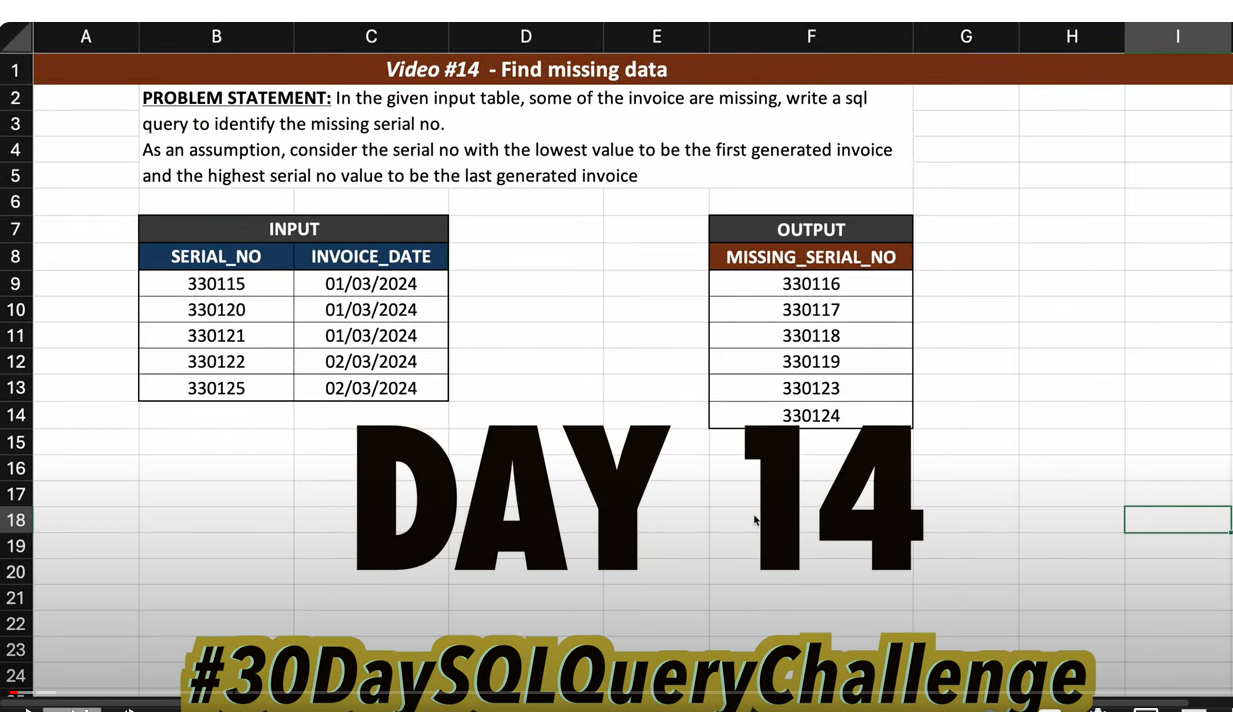
A screenshot of a computer

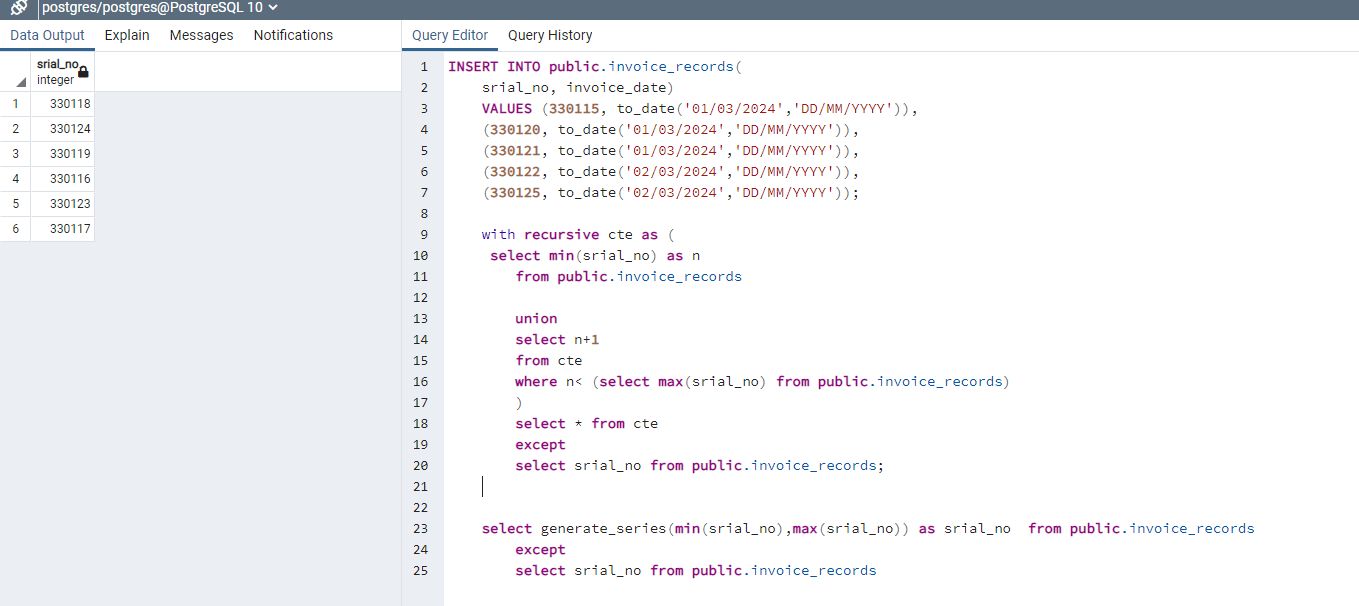
Description automatically generated

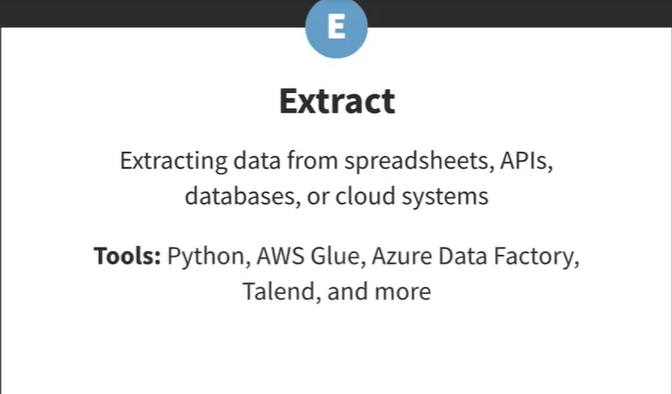
conda install -c anaconda psycopg2

A screenshot of a computer

Description automatically generated







A screen shot of a computer

Description automatically generated

A screenshot of a phone

Description automatically generated

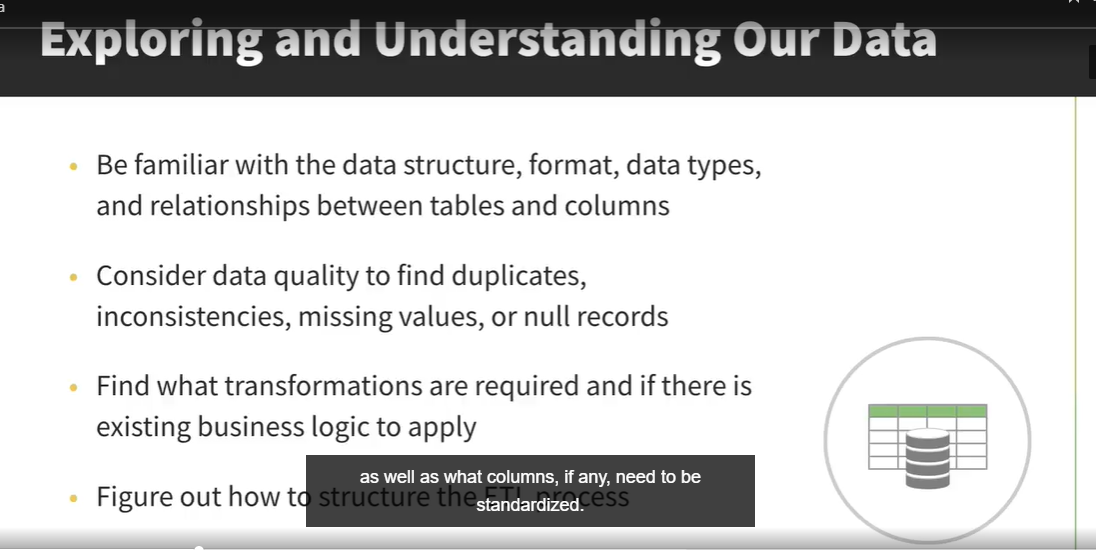
A screenshot of a phone

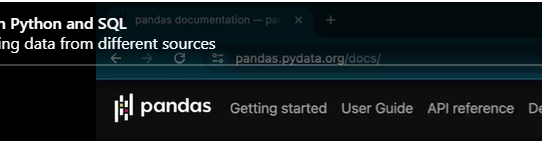
Description automatically generated

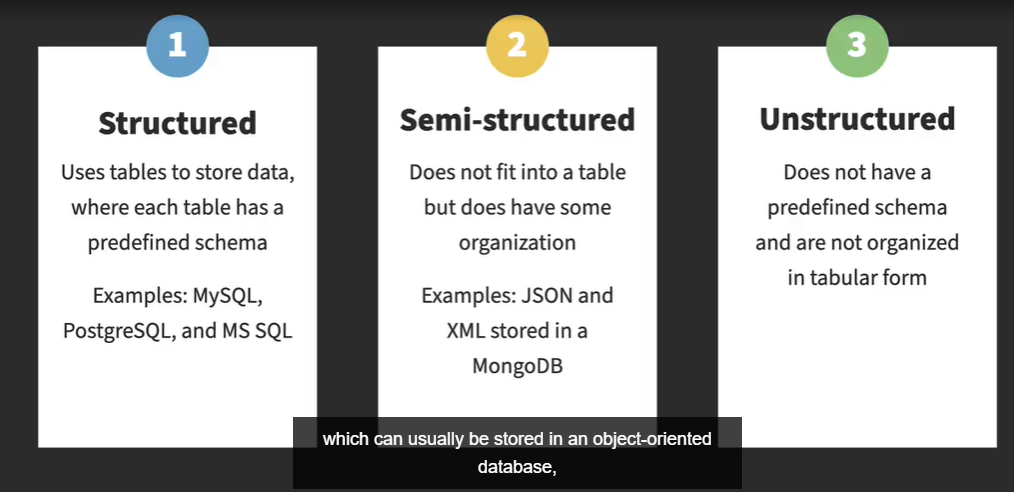
A screenshot of a computer

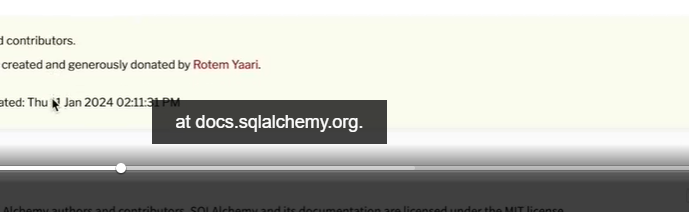
Description automatically generated





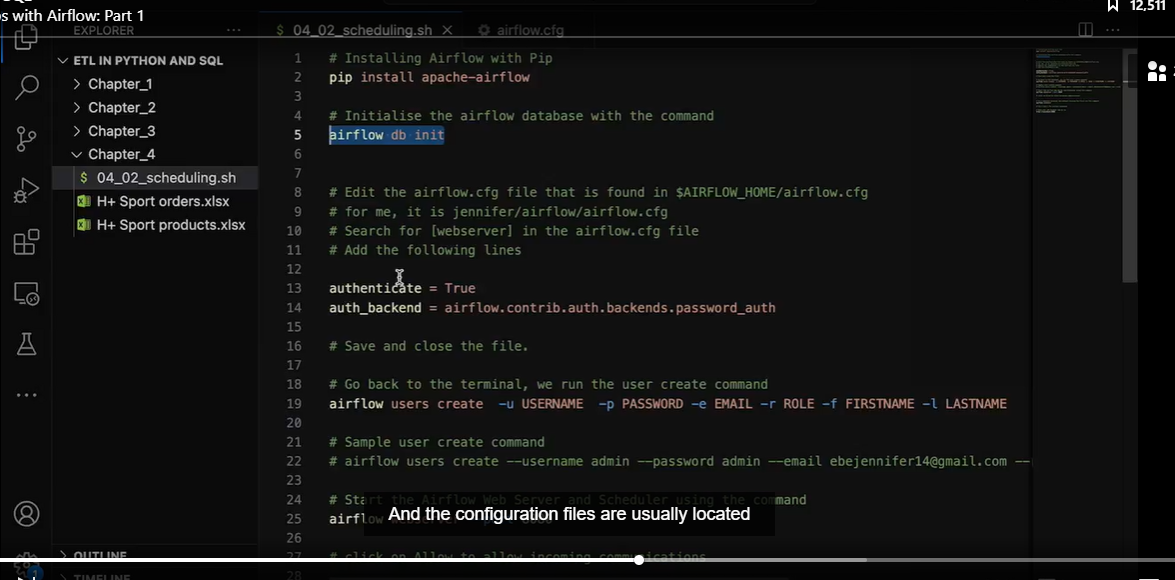


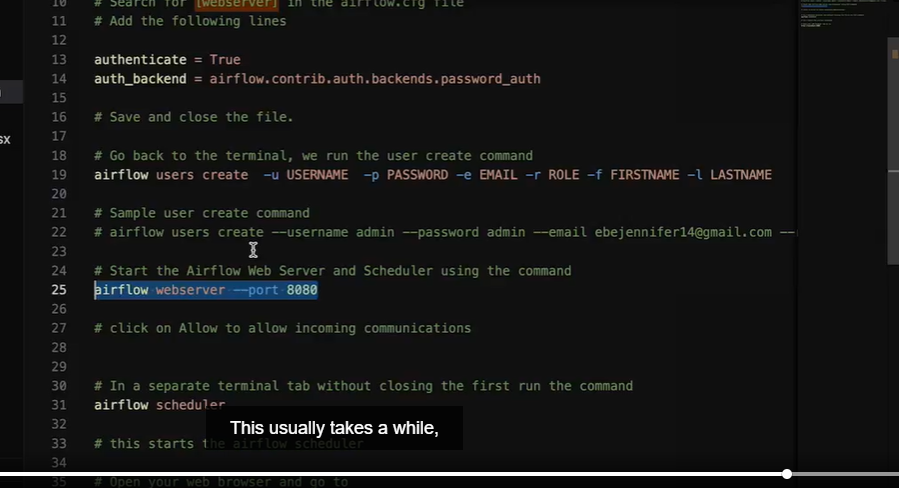




Pip install apache-airflow

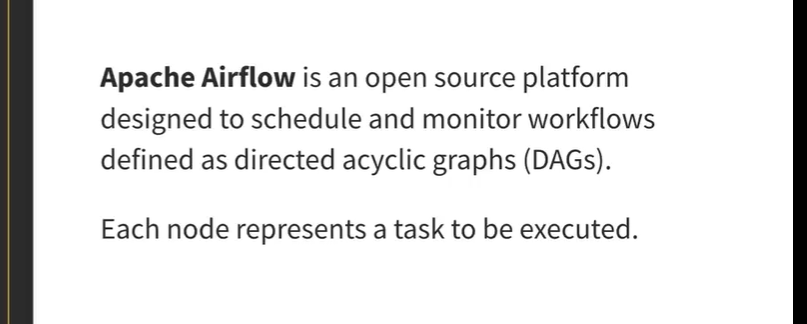
Airflow d binit

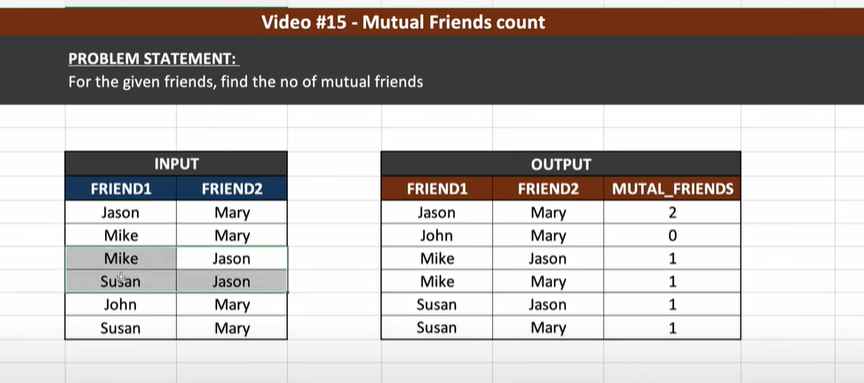




A screenshot of a computer

Description automatically generated





Get consecutive logins spanned 5 days or more, and get start\_date, end\_date:

INSERT INTO public.user\_active(

user\_id, login\_date)

VALUES (1 ,cast('2024-03-01' as date)) ,

(1 ,cast('2024-03-02' as date)) ,

(1 ,cast('2024-03-03' as date)) ,

(1 ,cast('2024-03-04' as date)) ,

(1 ,cast('2024-03-05' as date)) ,

(1 ,cast('2024-03-06' as date)) ,

(1 ,cast('2024-03-07' as date)) ,

(1 ,cast('2024-03-08' as date)) ,

(1 ,cast('2024-03-09' as date)) ,

(1 ,cast('2024-03-10' as date)) ,

(2 ,cast('2024-03-02' as date)) ,

(2 ,cast('2024-03-03' as date)) ,

(2 ,cast('2024-03-04' as date)) ,

(2 ,cast('2024-03-04' as date)) ,

(2 ,cast('2024-03-05' as date)) ,

(2 ,cast('2024-03-06' as date)) ,

(2 ,cast('2024-03-07' as date)) ,

(2 ,cast('2024-03-08' as date)) ,

(2 ,cast('2024-03-09' as date)) ,

(3 ,cast('2024-03-10' as date)) ,

(3 ,cast('2024-03-11' as date)) ,

(3 ,cast('2024-03-12' as date)) ,

(3 ,cast('2024-03-13' as date)) ,

(3 ,cast('2024-03-15' as date)) ,

(3 ,cast('2024-03-16' as date)) ,

(3 ,cast('2024-03-17' as date)) ,

(1 ,cast('2024-03-17' as date)) ,

(1 ,cast('2024-03-18' as date)) ,

(1 ,cast('2024-03-19' as date)) ,

(1 ,cast('2024-03-20' as date)) ,

(1 ,cast('2024-03-21' as date)) ,

(1 ,cast('2024-03-22' as date)) ,

(1 ,cast('2024-03-23' as date)) ,

(4 ,cast('2024-03-12' as date)) ,

(4 ,cast('2024-03-13' as date)) ,

(4 ,cast('2024-03-14' as date)) ,

(4 ,cast('2024-03-15' as date)) ,

(4 ,cast('2024-03-16' as date)) ,

(4 ,cast('2024-03-17' as date)) ,

(4 ,cast('2024-03-18' as date)) ,

(5 ,cast('2024-03-16' as date)) ,

(5 ,cast('2024-03-18' as date)) ,

(5 ,cast('2024-03-18' as date)) ,

(6 ,cast('2024-03-18' as date)) ,

(6 ,cast('2024-03-18' as date)) ,

(6 ,cast('2024-03-19' as date)) ,

(6 ,cast('2024-03-20' as date)) ,

(6 ,cast('2024-03-20' as date)) ,

(6 ,cast('2024-03-22' as date)) ,

(6 ,cast('2024-03-22' as date)) ,

(6 ,cast('2024-03-24' as date)) ;

select \* from public.user\_active;

with user\_logins as (

select

user\_id,

login\_date,

date\_group,

b.gap\_day,

count(date\_group)over(partition by user\_id,date\_group order by user\_id, date\_group) as day\_count

from (

select user\_id,login\_date,

(fol\_login\_dt-login\_date) as gap\_fol,

(login\_date - pre\_login\_dt) as gap\_pre,

gap\_day,

login\_date-gap\_day::int as date\_group

from (

select

user\_id,

login\_date,

lag(login\_date)over(partition by user\_id order by login\_date) as pre\_login\_dt,

lead(login\_date)over(partition by user\_id order by login\_date) as fol\_login\_dt,

dense\_rank()over(partition by user\_id order by login\_date) as gap\_day

from public.user\_active

order by user\_id,login\_date

)a

)b order by user\_id,login\_date,

date\_group ,b.gap\_day

),

login\_start\_end as (

select distinct user\_id,date\_group,

min(login\_date)over(partition by user\_id,date\_group order by user\_id,date\_group) as start\_date,

max(login\_date)over(partition by user\_id,date\_group order by user\_id,date\_group) as end\_date,

min(gap\_day)over(partition by user\_id,date\_group order by user\_id,date\_group) as start\_gap,

max(gap\_day)over(partition by user\_id,date\_group order by user\_id,date\_group) as end\_gap

from user\_logins

where day\_count>=5 order by user\_id

),

day\_gaps as (

select user\_Id,start\_date,end\_date,(end\_date-start\_date)+1 as daygaps

from

login\_start\_end

)

select \* from day\_gaps where daygaps>=5

order by user\_id,start\_date

;

