**Python Questions:**

**TASK 1**

One of our products is in charge of downloading and ingesting millions of records from our clients.

Recently during ingesting a large dataset we had our entire DB(postgres) go down and the entire ingestion process from a pandas dataframe to sql took around 2-3 hours because of the RAM unavailability. Now this has two simple fixes

- Increase ram/ scale the db on demand

[RESULT]:

One of the possible way would be to limit the data to a smaller subset

I would consider an example:

by probing every-nth value in a source.

- change our code to accommodate these restrictions and make the entire ingestion process much faster on the way.

[RESULT]:

import pandas as pd

importnumpy as np

fromos import path

n\_rows = 1\_000\_000

n\_cols = 1000

fori in range(1, 3):

filename = 'analysis\_%d.csv' % i

file\_path = path.join('csv\_files', filename)

df = pd.DataFrame(np.random.uniform(0, 100, size=(n\_rows, n\_cols)), columns=['col%d' % i for i in range(n\_cols)])

print('Saving', file\_path)

df.to\_csv(file\_path, index=False)

df.head()

How would you approach this? We are not looking for a full blown ingestion logic. Just a small script to take a given csv file and upload it to DB in an efficient manner.

[RESULT]:

import pandas as pd

data = pd.read\_csv (r'C:\Users\Vinay\Desktop\Test\products.csv')

df = pd.DataFrame(data)

print(df)

[RESULT]:

importpyodbc

conn = pyodbc.connect('Driver={SQL Server};'

'Server=VINAY\SQLEXPRESS;'

'Database=test\_database;'

'Trusted\_Connection=yes;')

cursor = conn.cursor()

Write code to take a large csv file(> 1GB ) and ingest it to table - public.test\_od

[RESULT]:

start = time.time()

#read data in chunks of 1 million rows at a time

chunk = pd.read\_csv('huge\_data.csv',chunksize=1000000)

end = time.time()

print("Read csv with chunks: ",(end-start),"sec")

pd\_df = pd.concat(chunk)

=======================================================

**TASK 2**

=======================================================

Our customer records are sometimes a little messy and contain duplicate entires.

For instance, we might have two records like "Pfizer" and "Pfizer Incorporated (old DO NOT USE)".

Please write a small basic function to merge these records together. We've provided a few example inputs below.

[RESULT]:

list\_2\_nodups = list\_2.drop\_duplicates()

pd.merge(list\_1 , list\_2\_nodups , on=['names'])

We are aware this is a very difficult problem: we're looking for what you can do quickly as a basic case, not an ideal solution.

Equipment ONLY - Saama Technologies

Saama Technologies

SaamaTech, Inc.

Takeda Pharmaceutical SA - Central Office

\*\*\* DO NOT USE \*\*\* Takeda Pharmaceutical

Takeda Pharmaceutical, SA

Ship to AstraZeneca

AstraZeneca, gmbh Munich

AstraZeneca (use AstraZeneca, gmbh Munich acct 84719482-A)

Use your own interpretation of the question and feel free to provide a written explanation for your choices as well.

=======================================================

**SQL Questions**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Visits** |  |  |  | **Customer** |  |  |  |
| Customer\_id | City\_id\_visited | Date\_visited |  | Customer\_id | Customer\_name | Gender | Age |
| 1001 | 2003 | 1-Jan-03 |  | 1001 | John | M | 25 |
| 1001 | 2004 | 1-Jan-04 |  | 1002 | Mark | M | 40 |
| 1002 | 2001 | 1-Jan-01 |  | 1003 | Martha | F | 55 |
| 1004 | 2003 | 1-Jan-03 |  | 1004 | Selena | F | 34 |

|  |  |  |
| --- | --- | --- |
| **City** |  |  |
| City\_id | City\_name | Expense |
| 2001 | Chicago | 500 |
| 2002 | Newyork | 1000 |
| 2003 | SFO | 2000 |
| 2004 | Florida | 800 |

======================================================

**TASK 1**

CREATETABLE Visits(Customer\_idINT,

City\_id\_visitedINT,

Date\_visitedDATE);

CREATETABLE Customer(Customer\_idINT,

Customer\_nameVARCHAR(50),

Gender VARCHAR(50),

AGE INT);

CREATETABLE City(City\_idINT,

City\_nameVARCHAR(50),

Expense INT);

INSERTINTO Visits VALUES(1001,2003,'1-Jan-03')

INSERTINTO Visits VALUES(1001, 2004,'1-Jan-04')

INSERTINTO Visits VALUES(1002,2001,'1-Jan-01')

INSERTINTO Visits VALUES(1004,2003,'1-Jan-03')

INSERTINTO Customer VALUES(1001,'John','M',25)

INSERTINTO Customer VALUES(1002,'Mark','M',40)

INSERTINTO Customer VALUES(1003,'Martha','F',55)

INSERTINTO Customer VALUES(1004,'Selena','F',34)

INSERTINTO City VALUES(2001,'Chicago',500)

INSERTINTO City VALUES(2002,'Newyork',1000)

INSERTINTO City VALUES(2003,'SFO',2000)

INSERTINTO City VALUES(2004,'Florida',800)

1. Cities frequently visited?

SELECT A.CITY\_NAME FROM (

SELECTC.City\_name CITY\_NAME,COUNT(C.City\_name)COUNTFROM City C

LEFTJOIN VISITS V ONC.City\_id=V.City\_id\_visited

GROUPBYC.City\_name) A

WHERECOUNT>1

1. Customers visited more than 1 city?

SELECT CUSTOMER\_NAME FROM (

SELECT C.Customer\_name,COUNT(V.CUSTOMER\_ID) VISIT

FROM Customer C

LEFT JOIN Visits V

ON C.Customer\_id=V.Customer\_id

GROUP BY Customer\_name) A

WHERE VISIT>1

1. Cities visited breakdown by gender?

SELECT CT.CITY\_NAME,C.GENDER

FROM Customer C

LEFT JOIN Visits V

ON C.Customer\_id=V.Customer\_id

INNER JOIN CITY CT

ON CT.City\_id=V.City\_id\_visited

1. List the city names that are not visited by every customer and order them by the expense budget in ascending order?

SELECT CT.City\_name,Expense

FROM City CT LEFT JOIN Visits V

ON CT.City\_id=V.City\_id\_visited

LEFT JOIN Customer C

ON C.Customer\_id=V.Customer\_id

WHERE Customer\_name IS NULL

ORDER BY Expense

1. Visit/travel Percentage for every customer?

SELECT Customer\_name,COUNT(V.City\_id\_visited)/COUNT(CITY\_ID) TRAVEL\_PER

FROM Customer C

LEFT JOIN Visits V

ON C.Customer\_id=V.Customer\_id

LEFT JOIN City CT

ON CT.City\_id=V.City\_id\_visited

WHERE City\_id IS NOT NULL

GROUP BY Customer\_name

ORDER BY Customer\_name

1. Total expense incurred by customers on their visits?

SELECT Customer\_name,ISNULL(SUM(Expense),0) TOTAL\_EXPENSE

FROM Customer C

LEFT JOIN Visits V

ON C.Customer\_id=V.Customer\_id

LEFT JOIN City CT

ON CT.City\_id=V.City\_id\_visited

GROUP BY Customer\_name

ORDER BY Customer\_name

1. list the Customer details along with the city they first visited and the date of visit?

SELECT CUSTOMER\_NAME,AGE,GENDER,DATE\_VISITED,EXPENSE FROM (

SELECT Customer\_name,Gender,AGE,Date\_visited,Expense,RANK() OVER(PARTITION BY CUSTOMER\_NAME ORDER BY DATE\_VISITED) r

FROM Customer C

LEFT JOIN Visits V

ON C.Customer\_id=V.Customer\_id

LEFT JOIN City CT

ON CT.City\_id=V.City\_id\_visited

WHERE Date\_visited IS NOT NULL

) A

WHERE r=1