Next Pathway Hack Backpackers problem statement

Introduction

Data industry is fast-tracked and so is Next Pathway. Next Pathway needs experts like you to bring out solutions to the data problems we brought to you today.

Problem statement 1 (LINEAGE PROBLEM STATEMENT)

Identify output columns' originating tables for SQL query, for example

```
37 LOCKING ROW FOR ACCESS
38 SELECT
39 T.SWB_CNTRY_ID,
40 T.CNTRY_TYPE_CD,
41 T.DW_EFF_DT,
42 S.DW_AS_OF_DT
43 FROM
44 (SELECT
45 SWB_CNTRY_ID,
46 CNTRY_TYPE_CD,
47 RCV_IN,
48 DW_EFF_DT,
49 MAX(DW_EFF_DT) MAX_EFF_DT
50 FROM IDW_DATA.CNTRY_MULTI_DEF_CD_T
51 WHERE
52 CURR_IN=1 GROUP BY 1,2,3,4) T,
53 (SELECT
54 SWB_CNTRY_ID,
55 CNTRY_SCHEME_CD,
56 DW_AS_OF_DT,
57 DW_ACTN_IND
58 FROM IDW_STAGE.CNTRY_MULTI_DEF_CD_S) S
59 WHERE
60 S.SWB_CNTRY_ID = T.SWB_CNTRY_ID AND S.CNTRY_SCHEME_CD = T.CNTRY_TYPE_CD
61 AND (S.DW_ACTN_IND='U' OR (S.DW_ACTN_IND='I' AND T.RCV_IN=0))
62 AND S.DW_AS_OF_DT > T.MAX_EFF_DT
```

Acceptance criteria

Write a script in Python or SQL where any SQL Statement can be passed and your program will trace the complete lineage and give the originating table for each output column of the input query.

for example:

sql_script="

SELECT

a.uid, b.uname

FROM

(select * from user) a,

(select * from user_details) b;

"

Sample expected output:

```
function get_originating_tables(sql_script):

  //parse the script and fetch the lineage and gives originating table name and put it in a dictionary or any other data structure

  return //column with the originating table
```

Output of your program

```
column => table

uid => user

uname => user_details
```

Evaluation criteria

1. Participant needs to push the code into a folder with name “Lineage Problem Statement”
2. Participant needs to mention the instructions to run the program in ReadME.md file
3. Evaluator will
    a. Fetch the code to their local
    b. Read the instructions to run
    c. Pass any SQL script to your program
    d. Checks the expected output
    e. Evaluates the program and the approach taken

Judging criteria

1. Based on the output of the program, the correct ones will be called for the interview
2. In a few cases, where the approach is right but due to some technical issues, the program doesn’t run, can also be called for the interview. This is purely the evaluator's decision.

Problem statement 2 (FUNCTION CONVERSION PROBLEM STATEMENT)

Note: You do not necessarily need the knowledge of ADF or Datastage here. You need to understand the meaning of functions and write your logic to convert them. All links with functions, their syntax and explanation are mentioned for reference.

Problem:

DataStage expression and function to ADF expression and dynamic contents

Link for DataStage functions and expressions: DataStage functions and expressions

Understand the functions on which you need to work(mentioned below in **Datastage Functions to work upon**) from the Datastage link mentioned above and create equivalent Python or Javascript code using equivalent ADF functions and expressions (example below),

An example to understand what solution you need to provide:

Below is the sample of writing an equivalent code for Datastage function CurrentDate which returns the current date.

Scenario
ADF does not have a direct function which can return the date value. Hence, In the code, utc_now() is a ADF function which gives datetime in the format 2018-04-15T13:00:00.0000000Z and then we will take the substring till T to return the date.

There can be other better ways to do this which you can put in your code.

Solution in Javascript code

```
function get_curr_date() {
   var day = utc_now(“dd”)
   var month = utc_now(“MM”)
   var year = utc_now(“yyyy”)
   curr_dt = day + “/” + month + “/” + year
   return curr_dt
}
```

Solution in Python code

```
def get_curr_date:
   day = utc_now(“dd”)
   month = utc_now(“MM”)
   year = utc_now(“yyyy”)
   curr_dt = day + “/” + month + “/” + year
```

return curr_dt

Explanation:
utc_now() is the ADF function which will provide the datetime in the format “2022-08-06T11:00:00”

utc_now(“dd”) gives the day part. Here, from 2022-08-06T11:00:00, it will give 06
utc_now(“MM”) gives the month part. Here, from 2022-08-06T11:00:00, it will give 08
utc_now(“yyyy”) gives the year part. Here, from 2022-08-06T11:00:00, it will give 2022

All these functions, their syntax and explanation is mentioned in the ADF links provided.

Important point to note:

Please make sure that the implementation is done by ADF expression only. Python/Javascript is a wrap to get it. In other words, in the given example, when converting from timestamp to date, it needs to be done using ADF expression, not Python or Javascript.

Link for ADF functions, please refer to
https://docs.microsoft.com/en-us/azure/data-factory/data-transformation-functions

Link for Expressions and functions in Azure Data Factory and Azure Synapse Analytics

https://docs.microsoft.com/en-us/azure/data-factory/control-flow-expression-language-functions

** To test the functions, you can have an ADF setup on your local for avoiding any syntactical errors. It is ok if you do not have the setup and it is not a bottleneck in solving the problem, keep writing the functions as per your understanding.

In case you face difficulty in having this setup, please connect us on the **support** channel on Discord.

Datastage Functions to work upon (Total: 19 functions)

Date and time functions

- CurrentTimestamp
- DateFromDaysSince2
- MinutesFromTime
- TimetFromTimestamp

Logical functions

- BitAnd
- BitCompress

Mathematical functions

- Abs
- Floor

Random

Null handling functions

IsNotNull
NullToValue

Number functions

AsDouble
MantissaFromDecimal

Raw functions

RawLength

String functions

AlNum
Change
Compare
Field
Str

Acceptance criteria

Write equivalent python or javascript code for each of the above functions in separate code files and push the code to your Github repository.

Please make sure that the implementation is done by ADF expression only. Python/Javascript is a wrap to get it. In other words, in the given example, when converting from timestamp to date, it needs to be done using ADF expression, not Python or Javascript.

Evaluation criteria

1. Participant needs to push the code into a folder with name “Function Conversion Problem Statement”
2. Participant needs to push one folder for each type of function like folder “Date_time_functions” and inside that you can push each function with a new file like “currenttimestamp.py” or “currenttimestamp.js”
3. Participant needs to mention the instructions to run the program in ReadME.md file
4. Evaluator will
    a. Fetch the code to their local
    b. Read the instructions to run
    c. Checks the expected output
    d. Evaluates the program and the approach taken

Judging criteria

1. Participants will keep pushing their functions as they complete them. We may expect not all the functions to be completed.
2. Based on the output of the scripts pushed, the correct ones will be called for the interview
3. In a few cases, where the approach is right but due to some technical issues or syntactical error while using the ADF function, the program doesn't run, and can also be called for the interview. This is purely the evaluator's decision.

Note: Below is the Incentive problem (Maximum weightage to get you the Cash prize)

PROBLEM STATEMENT 3 - (XML PROBLEM STATEMENT)

Identify how data flows from each transformation (including sql overrides if any) and is loaded into targets on column level for Informatica XML(For reference: Understanding XML document terminologies)

Input

Wf_src_idw_cntry_multi_def_cd.xml

Download the XML file and understand the transformation happening on the data.

Important point to note:

The XML has the tags which are Informatica specific but that does not make it any harder for non-informatica professionals. Being a data enthusiast, you understand the data flow, transformations, source and target. Hence, understand the XML with the help of Google or connect with a mentor wherever you get stuck.

Acceptance criteria

Understand the Source columns, the transformations happening over those columns and the target where they are getting dumped.
Once understood, your task is to

1. Write a python program to parse the XML and fetch the lineage of all downstream transformations over a column from source to target.
2. Pay attention on sources to fetch the columns and then trace the XML document to get the transformations logic over that column
3. Provide the complete lineage of transformation on these columns like below
   Expected output:
   source(tableA.column1) -> filter(col1) -> join(colA) -> target(tableB.column2)
   source(tableB.column2) -> join(colAB) -> target(tableC.column3)

Evaluation criteria

1. Participant needs to push the code into a folder with the name "XML Problem Statement"

2. Participant needs to mention the instructions to run the program in ReadME.md file
3. After running the program on the XML, the output should give the columns lineage like mentioned above.

Judging criteria

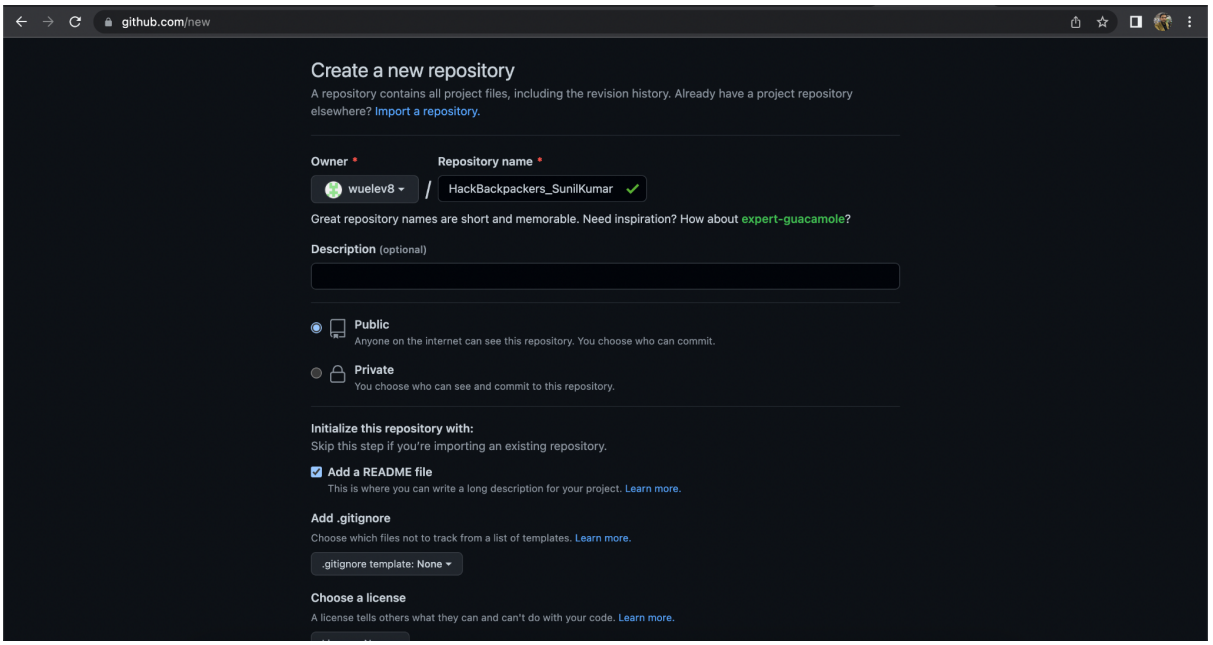
1. Based on the output of the program, the correct ones will be called for the interview
2. In a few cases, where the approach is right but due to some technical issues, the program doesn't run, can also be called for the interview. This is purely the evaluator's decision.

Github Guidelines

Create Repository

1. Create your repository with name "<Hackathon_name>_<Your_name>
2. Keep your repository **public**.
3. Add a README file to your repository

Refer the screenshot below



View [How to create repo and make it public with README.md file video]

Provide us with the repository url while in the registration desk(registration channel on Discord) which starts at 8:30 AM

Instruction to write in README.md

Follow the below format to write in README.md file

```
# HackBackpackers_SunilKumar

Problem Statements

Problem Statement 1 – Lineage problem statement

# Description
<Write some description for your solution>

# Prerequisite
<Write any prerequisites needed to run your solution>

# How to run
<Write steps to run your solution>
Steps
1.
2.

# Any other points to mention
<Any other points if you want to mention>

Problem Statement 2 – Functions problem statement

# Description

# How to run

# Any other points to mention


Problem Statement 3 – XML problem statement

# Description

# How to run
```

README.md content

HackBackpackers_SunilKumar

Problem Statements

Problem Statement 1 - Lineage problem statement

 # Description

<Write some description for your solution>

 # Prerequisite

<Write any prerequisites needed to run your solution>

 # How to run

<Write steps to run your solution>

Steps

1.

2.

 # Any other points to mention

<Any other points if you want to mention>

Problem Statement 2 - Functions problem statement

 # Description

<Write some description for your solution>

 # Prerequisite

<Write any prerequisites needed to run your solution>

 # How to run

<Write steps to run your solution>

Steps

1.

2.

 # Any other points to mention

<Any other points if you want to mention>

Problem Statement 3 - XML problem statement

 # Description

<Write some description for your solution>

 # Prerequisite

<Write any prerequisites needed to run your solution>

 # How to run

<Write steps to run your solution>

Steps

1.

2.

 # Any other points to mention

<Any other points if you want to mention>

How to Submit on checkpoints

Checkpoint 1 (At 3 PM)

This checkpoint is only to ensure that you have understood the problem statements and you are working on this.

We expect you to submit your work on Github irrespective of whether any solution is complete or not. No evaluation is done at this checkpoint.

Checkpoint 2 (At 7 PM)

This checkpoint is to ensure that you do not have any issue and you are able to submit your solution on Github.

We expect you to submit your work on Github. No evaluation is done at this checkpoint.

Final Submission (At 11 PM)

You must have done a great job till the Checkpoint 2 and the last step would be left to fill.

We will share a link to submit the final submission post Checkpoint 2.
In the final submission, you just have to provide the Github repository link again and in case you want to submit a video of your solutions working.

How to reach the support team if you have any query?

JUST reach out to the **support channel on Discord.**