```
import configparser
import logging
import json
import pyodbc
import textwrap
import time
import datetime
import logging
import collections
import os

import azure.functions as func

def default(o):
  if isinstance(o,(datetime.datetime,datetime.date)):
     return o.isoformat()

def main(req: func.HttpRequest) -> func.HttpResponse:
   logging.info('Python HTTP trigger function processed a request.')
   #Grab the table name from HTTP request
   table_name=req.params.get('table_name')
   schema_name = req.params.get('schema_name')
   r_date=req.params.get('r_date')

   if not table_name or not schema_name or not r_date:
     try:
         req_body = req.get_json()
     except ValueError:
         pass
     else:
         tablename = req_body.get('table_name')
         schema_name = req.params.get('schema_name')
         r_date=req.params.get('r_date')
   #Database Username and password
   database_username = os.getenv('UsernameFromKeyVault')
   database_password = os.getenv('PasswordFromKeyVault')
   logging.info('Fetching  credentials Azure key vault')
   #Print out the Drivers
   logging.info(pyodbc.drivers())
   #Define the Driver
   driver = '{ODBC Driver 17 for SQL Server}'
   # Create the connection String
   connection_string = textwrap.dedent('''
     Driver={driver};
     Server=testingdemo11.database.windows.net,1433;
     Database=testingdemo11;
     Uid={username};
     Pwd={password};
     Encrypt=yes;
     TrustServerCertificate=no;
     Connection Timeout=30;
  '''.format(
     driver=driver,
     username=database_username,
     password=database_password
     ).replace("'",'')
  )
```

```
 # New Connection
    try:
        cnxn: pyodbc.Connection = pyodbc.connect(connection_string)
    except:
        time.sleep(5)
        cnxn: pyodbc.Connection = pyodbc.connect(connection_string)

    #Define The query paramter from Htpp trigger vals
    val1=str(table_name)
    print(val1)
    val2=str(schema_name)
    print(val2)
    val3=str(r_date)
    print(val3)

    ####Selct Query to Fecth Desired output#####
    select_query="SELECT Table_Name,Schema_name,Status,Run_Date  FROM STATS WHERE
Table_Name='%s' and Schema_name= '%s' and Run_Date ='%s' ;" %(val1,val2,val3)
    select_query1="SELECT Table_Name,Schema_name,Status,Run_Date  FROM STATS1 WHERE
Table_Name='%s' and Schema_name= '%s' and Run_Date ='%s' ;" %(val1,val2,val3)
    logging.info(msg='Database Connection Successful.')

    #Creat the Cursor
    cursor_object: pyodbc.Cursor = cnxn.cursor()
    cursor_object1: pyodbc.Cursor = cnxn.cursor()

    #Execute the query
    records2=list()

    cursor_object.execute(select_query)
    columns=[column[0] for column in cursor_object.description]
    records=[]
    for row in cursor_object.fetchall():
        records.append(dict(zip(columns,row)))

    cursor_object1.execute(select_query1)
    columns=[column[0] for column in cursor_object1.description]
    records1=[]
    for row in cursor_object1.fetchall():
        records1.append(dict(zip(columns,row)))

    records2=records+records1
    return func.HttpResponse(
        body=json.dumps(obj=records2, indent=4, default= default),
        status_code=200
    )
```