**Witten by : Eghe Ikponmwosa-Eweka**

**Dated: 07/05/2024**

Table of Contents

Step 1: Establish Remote Desktop Connection2

Step 2: Open SQL server 3

Step 3: Extract Warehouse Data from SQL Server5

Step 4: Upload Data to Google Cloud Storage5

Step 5: Step 5: Open Your GitHub6

## Step 6: Step 6: Store Data in Google Cloud Platform……………………………………………………………………………….9

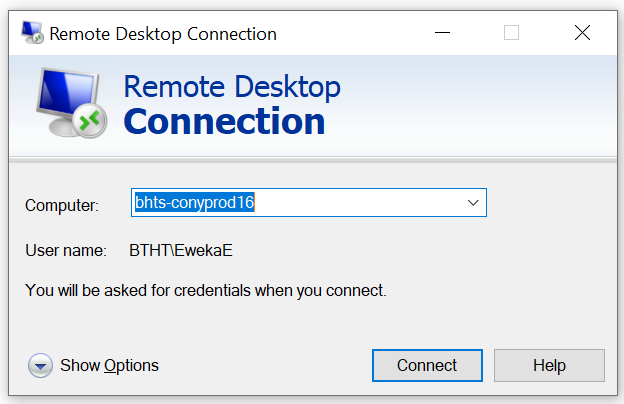
Maintaining and Updating BRI Warehouse Data on SQL and Google Cloud Platform

This guide outlines the steps to maintain and update the BRI Warehouse data on the Google Cloud Platform. The BRI Warehouse data contains pseudonymized data derived from the trust EPR system, which requires regular updates to ensure data accuracy and relevance. The following procedure explains how to transfer data from the SQL server to the Google Cloud Platform using a desktop remote connection and Google Cloud storage.

**Step 1: Establish Remote Desktop Connection**

To access the SQL server and extract warehouse data, you need to establish a remote desktop connection to your designated server. Follow these steps:

* Open the Remote Desktop Connection application on your local machine,see diagram below:



*Figure 1: Diagram showing Remote Desktop Connection*

* In the 'Computer' field, enter the server name: **bhts-conyprod16**.
* Click 'Connect' to initiate the remote desktop session.
* Provide your login credentials (username and password) as required.

Once connected, you should be able to interact with the remote desktop as if you were physically present at the server.

**Step 2: Open SQL server**

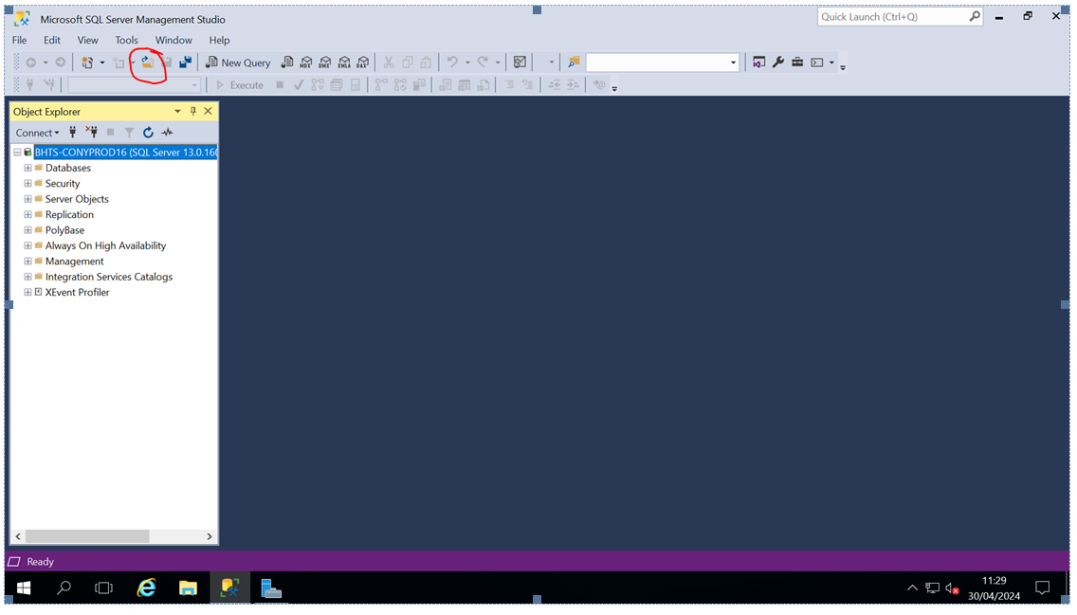
* Open SQL Server Management Studio(SSMS) and connect to bhts-conyprod16
* Then from folder: D:\JDB\Scripts\ src\_StagingDatabase\_Warehouse\ open and run in order :

a1\_RebuildLocalCopy\_tmp.sql

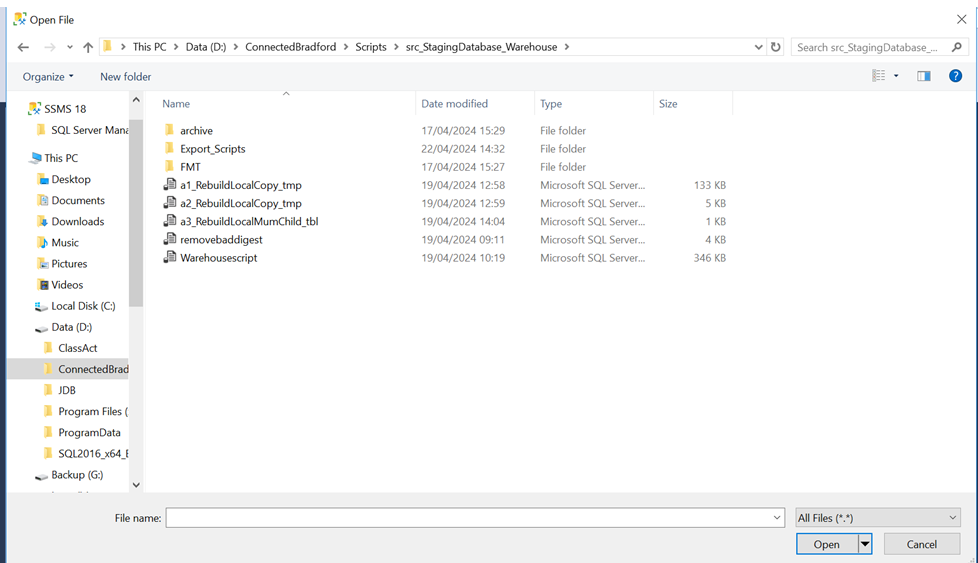
a2\_RebuildLocalCopy\_tmp.sql

a3\_RebuildLocalMumChild\_tbl

See below diagram on how to access folders above



*Figure 2: Diagram shows how to access folder D from SQL server*



*Figure 3: Diagram shows how to access folder D from SQL server*

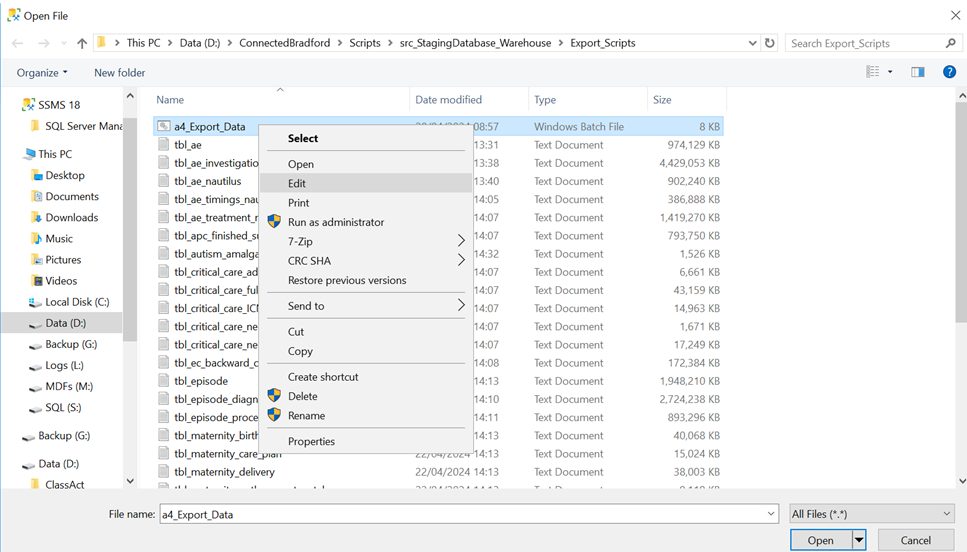
This will update the local copy of the warehouse extract.

**Step 3: Extract Warehouse Data from SQL Server**

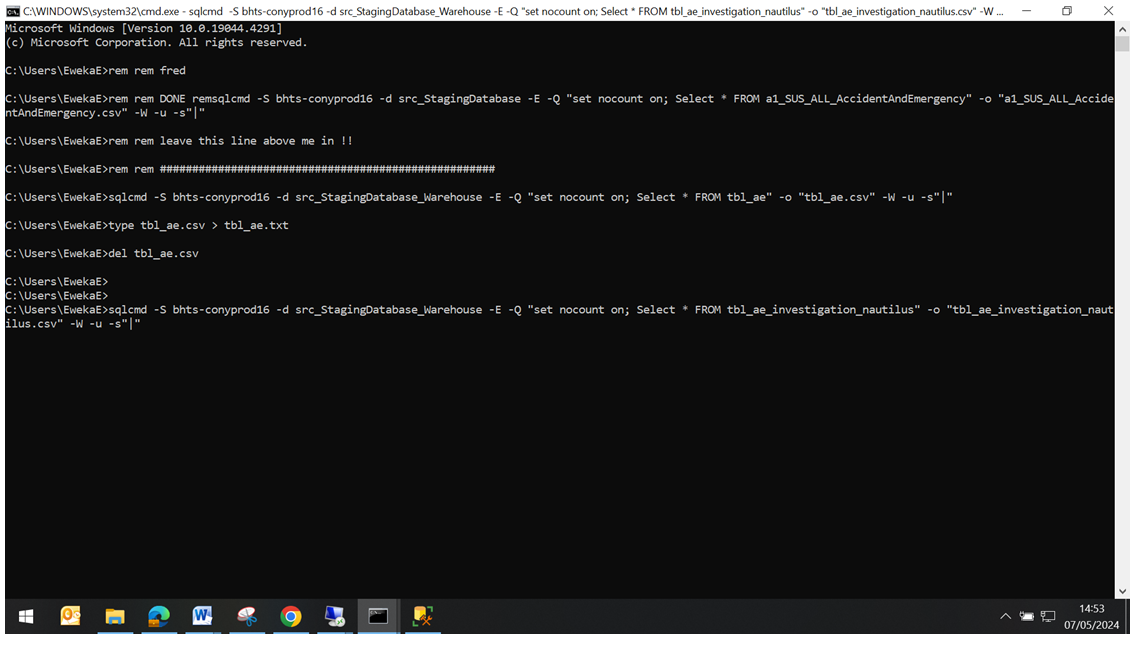
To export warehouse data from the SQL server, follow these instructions to ensure accurate data extraction and preparation for transfer to Google Cloud. This step involves executing a pre-written script to generate the necessary data text files and save them in the appropriate format.

**Accessing the Export Script**

* Navigate to the export script directory on the remote desktop: D:\ConnectedBradford\Scripts\src\_StagingDatabase\_Warehouse\Export\_Scripts.
* Open the script named a4\_Export\_Data. Review its contents to ensure it specifies the correct database and data range for extraction.
* Make any necessary edits to the script, such as adjusting the database connection settings or data extraction parameters, to ensure accurate export. See diagrams below.



*Figure 4: Diagram showing how to edit the export script*



*Figure 5:Diagram showing how to edit the export script*

**Executing the Export Script**

With the script reviewed and updated, execute it to begin the data export process.

* Navigate to the directory containing the script:

D:\ConnectedBradford\Scripts\src\_StagingDatabase\_Warehouse\Export\_Scripts

* Run the export script( a4\_Export\_Data) to extract the warehouse data

The export script will run a series of database queries and operations to create the required data files. This may take several minutes, depending on the size of the data set.

**Locating the Exported Data**

* Once the export script has completed, locate the exported data files on the server:
* Open Windows Explorer and navigate to the output directory specified in the script. Typically, this is where the new data files are created:D:\JDB\Scripts\src\_StagingDatabase\_Warehouse
* Confirm that the expected data files are present and have been generated successfully.

**Step 4: Upload Data to Google Cloud Storage**

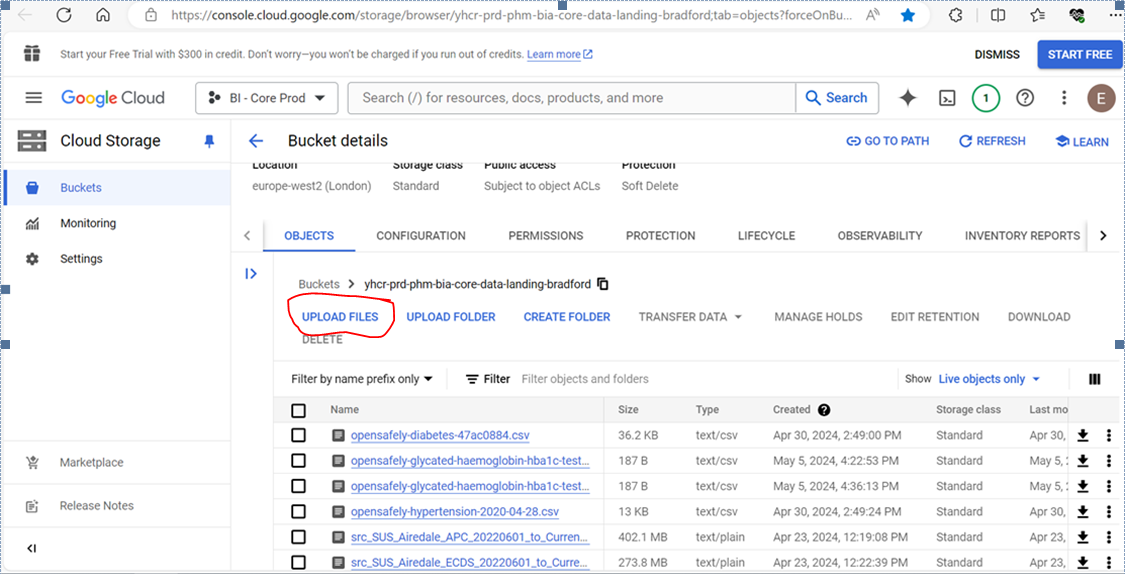
Following the extraction of warehouse data, the next task is to upload it to the Google Cloud Storage (GCS) bucket for further processing. This step-by-step guide provides detailed instructions on how to upload data to a specific GCS bucket.

* Access Google Cloud Platform via a web browser and sign in with your authorized credentials which is your yhcr account details.
* Navigate to 'Storage' from the Google Cloud console menu to access the Google Cloud Storage service.In this case we are navigating to the yhcr bucket below:

<https://console.cloud.google.com/storage/browser/yhcr-prd-phm-bia-core-data-landing-bradford;tab=objects?forceOnBucketsSortingFiltering=false&project=yhcr-prd-phm-bia-core&prefix=&forceOnObjectsSortingFiltering=false>

**Note**: This is the bucket id in this case: yhcr-prd-phm-bia-core-data-landing-bradford and may vary depending on the bucket you intend to upload your files

* Click 'Upload Files' as shown in the diagram below to initiate the upload to Google Cloud Storage .



*Figure 6: Diagram showing how to upload files to the GCP bucket*

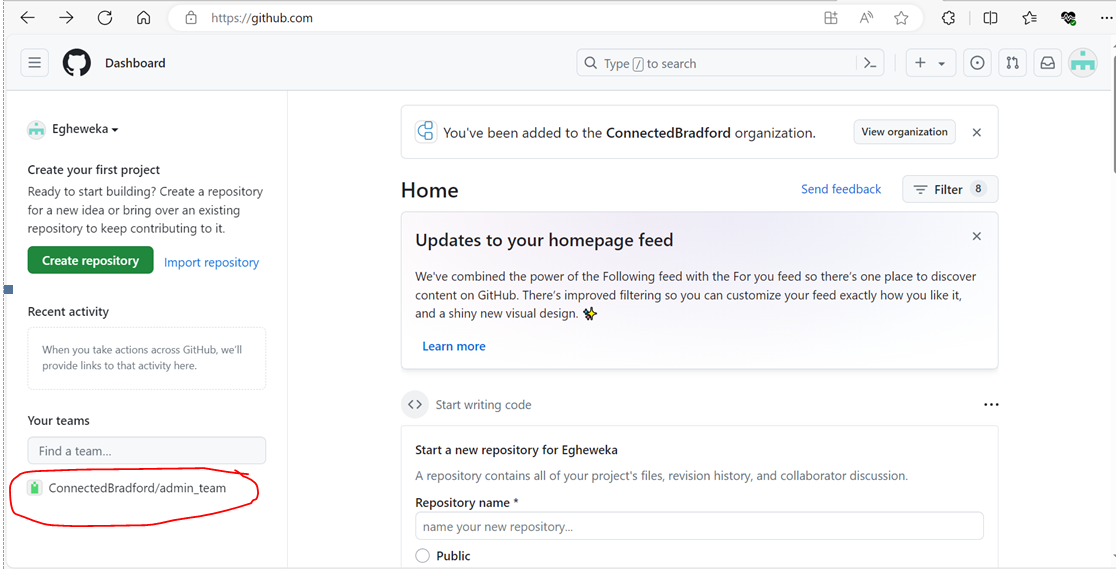
* Once the upload is complete, verify that the data has been successfully transferred to the Google Cloud Storage bucket.

**Verify Uploaded Data**

* Access your GCS bucket to ensure the data has been successfully uploaded.
* Check the file names and timestamps to confirm that the correct data has been uploaded.
* Examine file sizes and other metadata to ensure the upload was completed without errors.

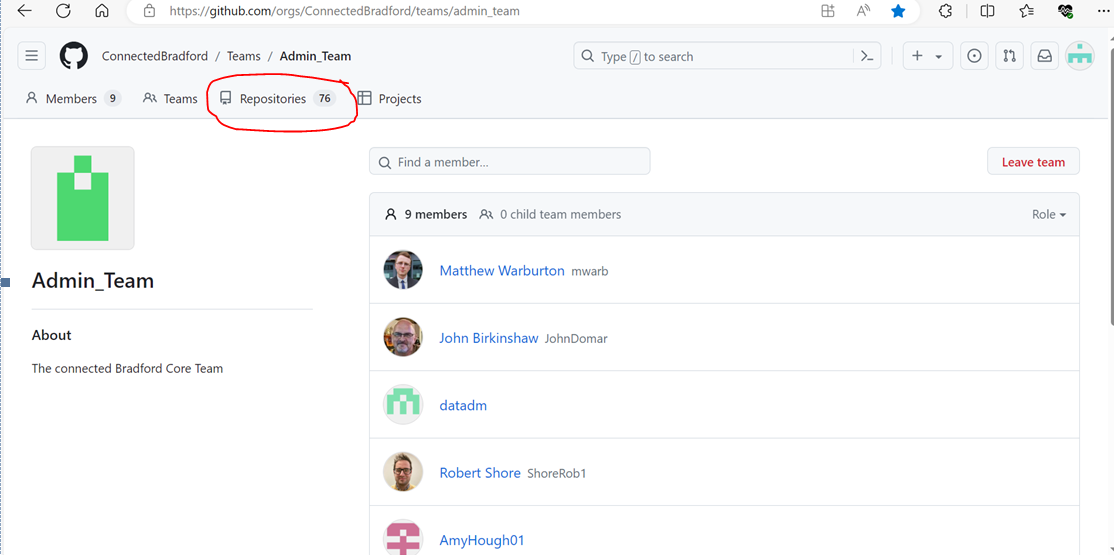
**Step 5: Open Your GitHub**

* Log on to your GitHub account and navigate to the [ConnectedBradford/admin\_team](https://github.com/orgs/ConnectedBradford/teams/admin_team) as shown in diagram below.



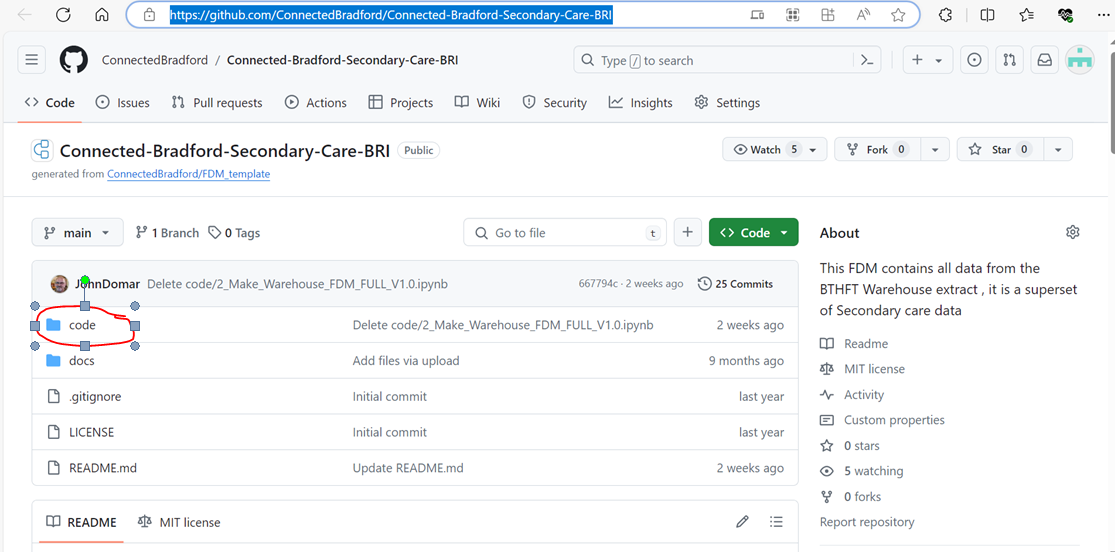
*Figure 7:Diagram showing ConnectedBradford/admin\_team on GitHub*

* Click on repository as shown in diagram below.



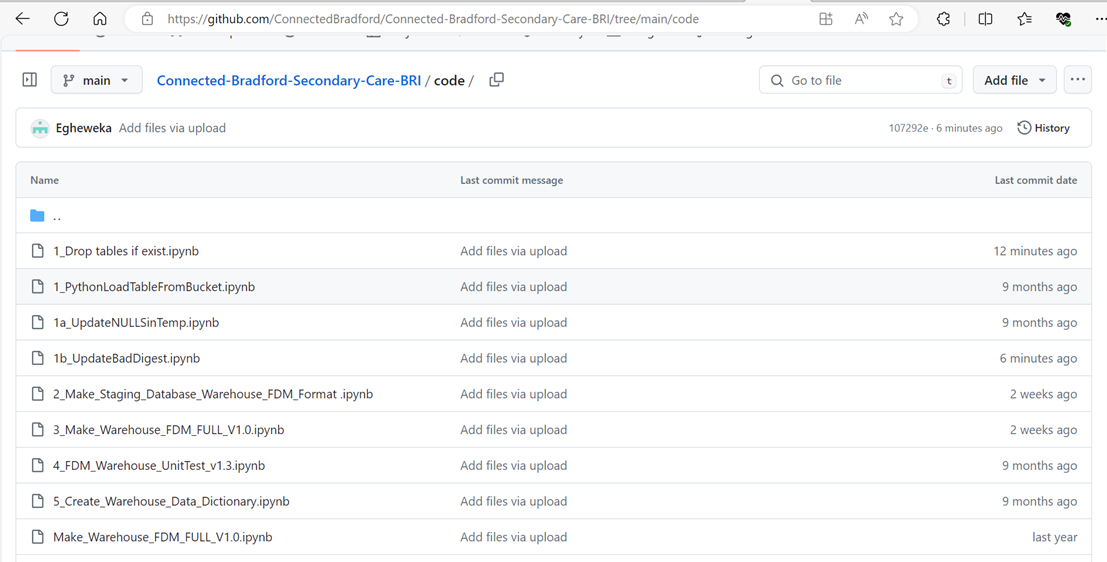
*Figure 8: Diagram showing repositories on GitHub*

* Navigate to <https://github.com/ConnectedBradford/CB_FDM_Warehouse_ALL>, and click on ‘code’ as shown in the diagram below, [Connected-Bradford-Secondary-Care-BRI/code at main · ConnectedBradford/Connected-Bradford-Secondary-Care-BRI (github.com)](https://github.com/ConnectedBradford/Connected-Bradford-Secondary-Care-BRI/tree/main/code)



*Figure 9: Diagram showing how to access the scripts*

* Download in order as shown is screen shot below



*Figure 10 :Diagram showing scripts as on GitHub*

1. 1\_Drop tables if exist.ipynb

[Connected-Bradford-Secondary-Care-BRI/code/1\_Drop tables if exist.ipynb at main · ConnectedBradford/Connected-Bradford-Secondary-Care-BRI (github.com)](https://github.com/ConnectedBradford/Connected-Bradford-Secondary-Care-BRI/blob/main/code/1_Drop%20tables%20if%20exist.ipynb)

1. 1\_PythonLoadTableFromBucket.ipynb

[Connected-Bradford-Secondary-Care-BRI/code/1\_PythonLoadTableFromBucket.ipynb at main · ConnectedBradford/Connected-Bradford-Secondary-Care-BRI (github.com)](https://github.com/ConnectedBradford/Connected-Bradford-Secondary-Care-BRI/blob/main/code/1_PythonLoadTableFromBucket.ipynb)

1. 1b\_UpdateBadDigest.ipynb

[Connected-Bradford-Secondary-Care-BRI/code/1b\_UpdateBadDigest.ipynb at main · ConnectedBradford/Connected-Bradford-Secondary-Care-BRI (github.com)](https://github.com/ConnectedBradford/Connected-Bradford-Secondary-Care-BRI/blob/main/code/1b_UpdateBadDigest.ipynb)

1. 1a\_UpdateNULLSinTemp.ipynb

[Connected-Bradford-Secondary-Care-BRI/code/1a\_UpdateNULLSinTemp.ipynb at main · ConnectedBradford/Connected-Bradford-Secondary-Care-BRI (github.com)](https://github.com/ConnectedBradford/Connected-Bradford-Secondary-Care-BRI/blob/main/code/1a_UpdateNULLSinTemp.ipynb)

1. 2\_Make\_Staging\_Database\_Warehouse\_FDM\_Format .ipynb

[Connected-Bradford-Secondary-Care-BRI/code/2\_Make\_Staging\_Database\_Warehouse\_FDM\_Format .ipynb at main · ConnectedBradford/Connected-Bradford-Secondary-Care-BRI (github.com)](https://github.com/ConnectedBradford/Connected-Bradford-Secondary-Care-BRI/blob/main/code/2_Make_Staging_Database_Warehouse_FDM_Format%20.ipynb)

1. 3\_Make\_Warehouse\_FDM\_FULL\_V1.0.ipynb

[Connected-Bradford-Secondary-Care-BRI/code/3\_Make\_Warehouse\_FDM\_FULL\_V1.0.ipynb at main · ConnectedBradford/Connected-Bradford-Secondary-Care-BRI (github.com)](https://github.com/ConnectedBradford/Connected-Bradford-Secondary-Care-BRI/blob/main/code/3_Make_Warehouse_FDM_FULL_V1.0.ipynb)

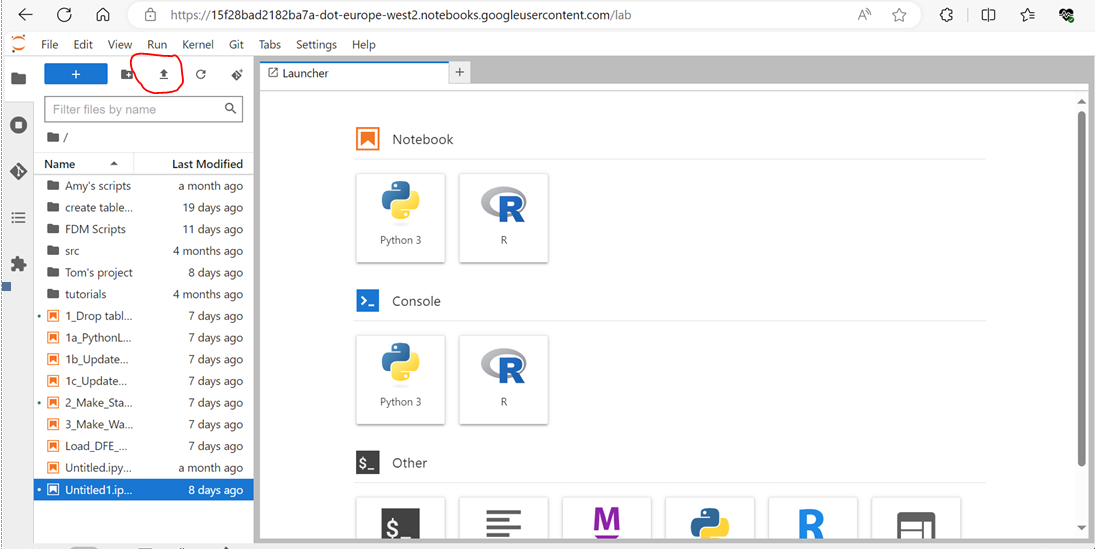
## Step 6: Store Data in Google Cloud Platform

After uploading data to Google Cloud Storage (GCS), the next step is to ensure it is correctly integrated into the Google Cloud Platform (GCP) ecosystem. This involves validating the data, loading it into the appropriate environment, and performing any necessary transformations. Follow these steps to achieve successful integration:

## Confirm the correct data has been uploaded by checking the file names and timestamps in the Google Cloud Storage bucket.

## Open your Vertex Notebook in Google Cloud Platform

* Click on the upload icon as shown in the diagram below to upload all scripts downloaded from your GitHub , <https://github.com/ConnectedBradford/CB_FDM_Warehouse_ALL>



*Figure 11: Diagram showing upload icon on Vertex notebook*

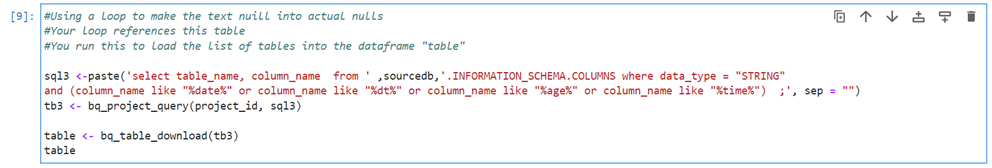
## Open scripts and perform data validation or checks to ensure the data's integrity and consistency, this is done following process outline the process for integrating the data into your GCP environment:

1. **Drop Existing Tables**: Open the ‘1\_Drop tables if exist.ipynb ‘ script and execute it to drop any existing tables in the dataset that might conflict with new data.
2. **Load Data from GCS:** Run the 1\_PythonLoadTableFromBucket.ipynb script to load the demographics master and person data from the GCS bucket into your workspace.

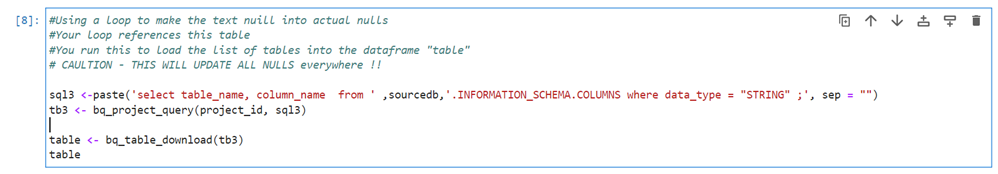
Note: Once loaded, review the data to ensure correct import and structure

1. **Update NULL Values**: Execute the 1a\_UpdateNULLSinTemp.ipynb script to convert "NULL" strings to actual NULL values for all date/time fields.

Note: In amending all datetime data type to null,either of the two options can be done before looping through the data ,see screen shot below for reference.

*Figure 12: Diagram showing code used to loop over list of tables with datatype datetime*

***OR***



*Figure 13 :Diagram showing code used to loop over list of tables with data type string*

* 2\_Make\_Staging\_Database\_Warehouse\_FDM\_Format .ipynb- This rebuilds the staging database for the warehouse.You are now ready to build the FDM:
* 3\_Make\_Warehouse\_FDM\_FULL\_V1.0.ipynb- This builds the FDM.