**AUDIT PROCESS**

1. Pre-Requirements for ETL Process
2. Source to Blue
3. Blue to Grey
4. Grey to Green
5. Green to Red

**Pre-Requirements for ETL Process:**

Step1: Create ETL\_CONTROLER Table for Source to Blue Zone in AUDIT schema.

Step1.1: Create ETL\_CONTROLER table with all source tables, columns, destination tables, columns and full load conditions.

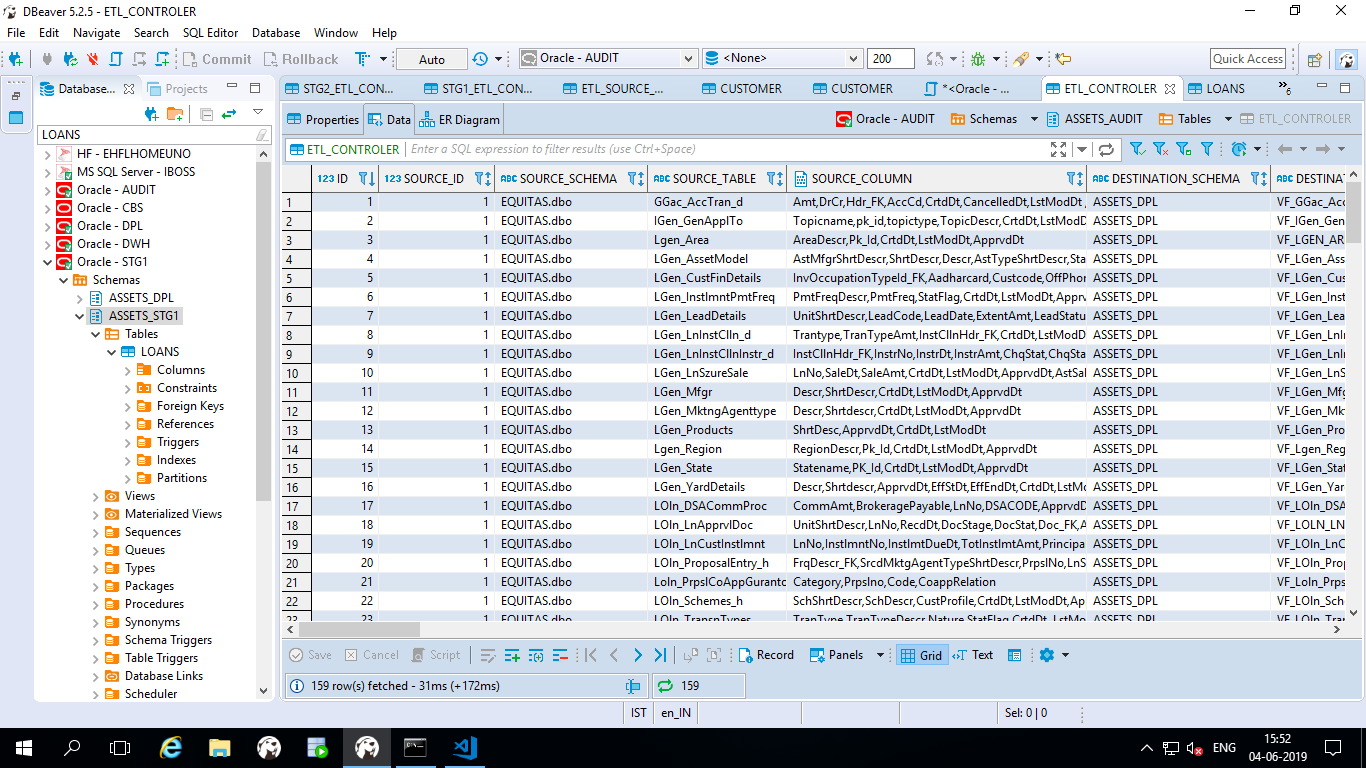


Table: ETL\_CONTROLER

**Full-Load:**

Step1.2: Update full load condition and status is one for all tables in Controller table while loading Full data push.

**Incremental Load:**

Step1.3: Please be careful while updating the Incremental conditions and ETL\_TIME to run the Incremental data.

Step1.4: Update status as one for daily incremental tables and remaining should be zero.

**Step2: Creating the ETL\_SOURCE\_DATABASE table.**

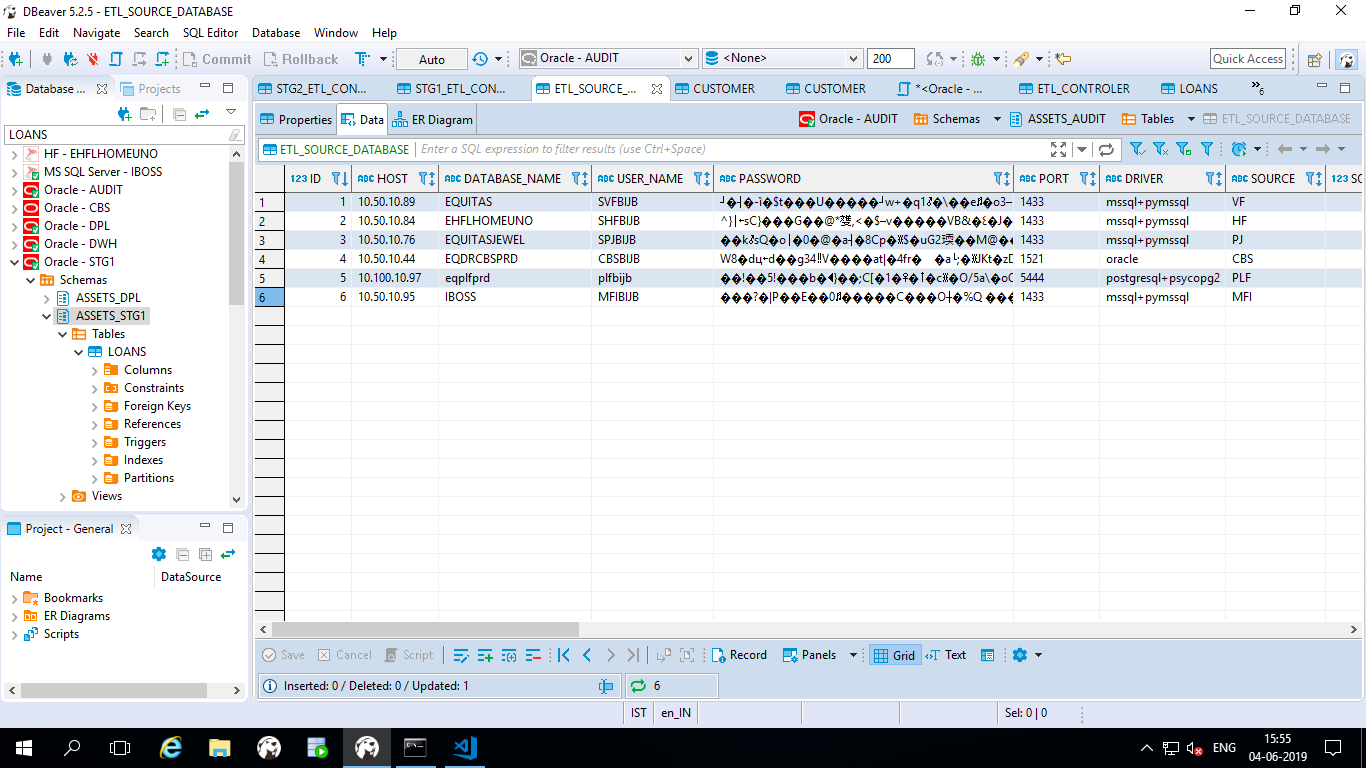


Table: ETL\_SOURCE\_DATABASE

Step2.1: In **ETL\_SOURCE\_DATABASE** table having all Source database credentials.

Step2.2: Here you have to update status as **“One” or “Zero”**

Step2.3: We have a flexibility here to run **Source wise or Zone wise** ETL process.

Step2.4: Update status is 1 which are the sources you are going to run in zone wise.

Step2.5: First it will establish the audit connection and fetch the all the Source credential records which we updated status as one.

Step3: The below one is the python connector file path location .This will give you the audit connection, Source connection, Destination connection and data load Source Details.

**DataLoad\Assets\ETL\utils\Connector.py**

**AUDIT**

**Source to Blue (S2B)**

**dataload\Assets\ETL\S2B\audit.py**

* **Audit Blue:**

1. After loading the data from source to blue check the **Audit\_blue** table in **Audit Schema** whether all tables are loaded or not. Status column will tell which table is in process. If status is ‘**Loading**’ means that table is in process, once completed it will turn to ‘**Done**’.

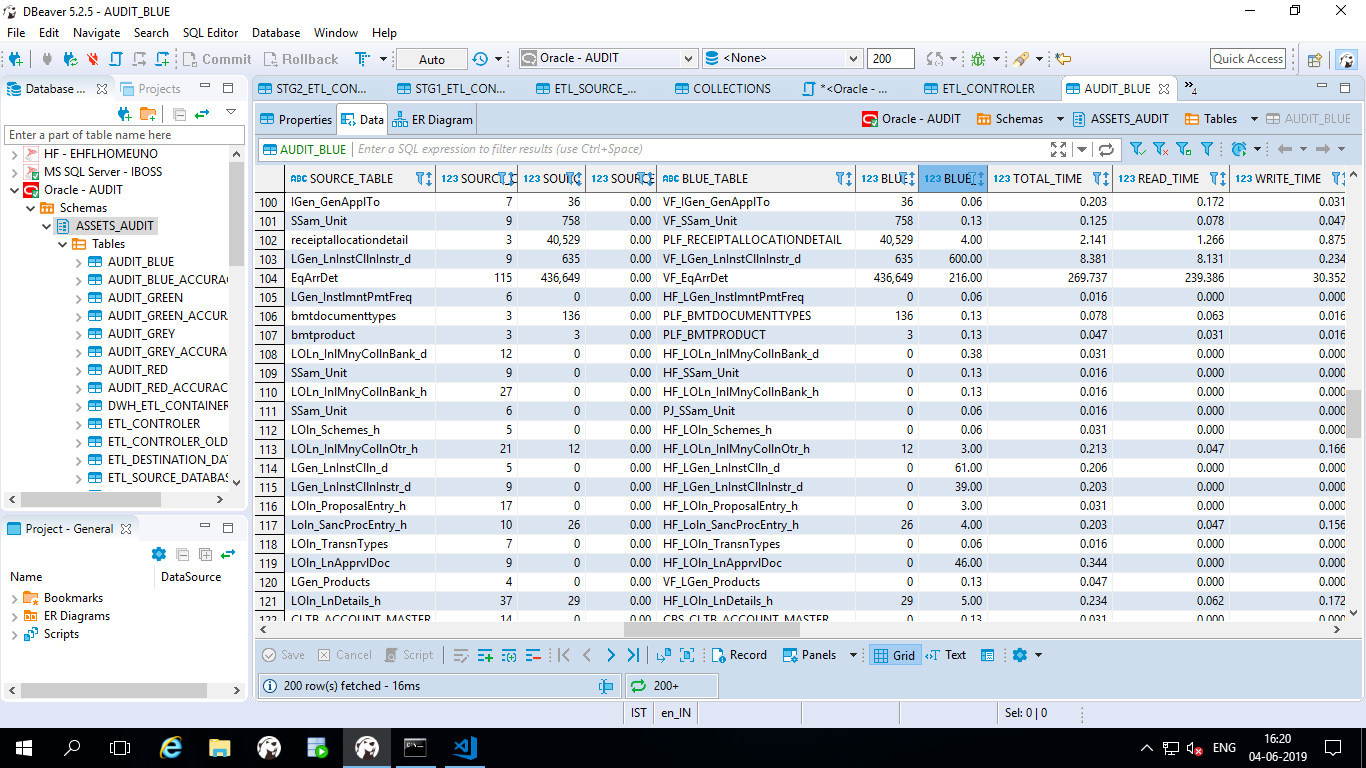


Table: audit\_blue

1. Here you can check all the **statistics** of each table like Start time, End Time, Total Time, Read Time, Write Time, Abnormality, Session\_Id, App\_User name and Status.
2. In Audit blue table you have to check weather abnormality column is YES or No. If you found any yes then you have to investigate data transfer from source to target with incremental condition applied.
3. The audit.py is responsible for audit update during data transfer. Update\_audit\_source and update\_audit\_blue python function used for calculation.
4. We can see all the statistics of Source to blue zone table count level information in Audit Blue table. By default all tables with status **ONE** in etl\_controller table will be audited on this level.

* **Audit Blue Accuracy:**

1. This is the process to check accuracy. In etl\_controller table you can see one column **accuracy\_chk. N**eed to set this column to ONE for the table you want to go for accuracy check.
2. If you set **accuracy\_chk** to ONE, its subsequent **column\_accuracy\_chk** needs to be filled. That is the **column name** of source and destination table.
3. System will check accuracy\_chk column, if found ONE, it will go for mathematical calculation like SUM, AVG, MIN, MAX on specified column in column\_accuracy\_chk.

Example: **SELECT MIN(EmiAmount), MAX(EmiAmount), AVG(EmiAmount), SUM(EmiAmount) FROM LOln\_ProposalEntry\_h**.

1. Make sure the column you set is numeric and present in both source and destination tables.

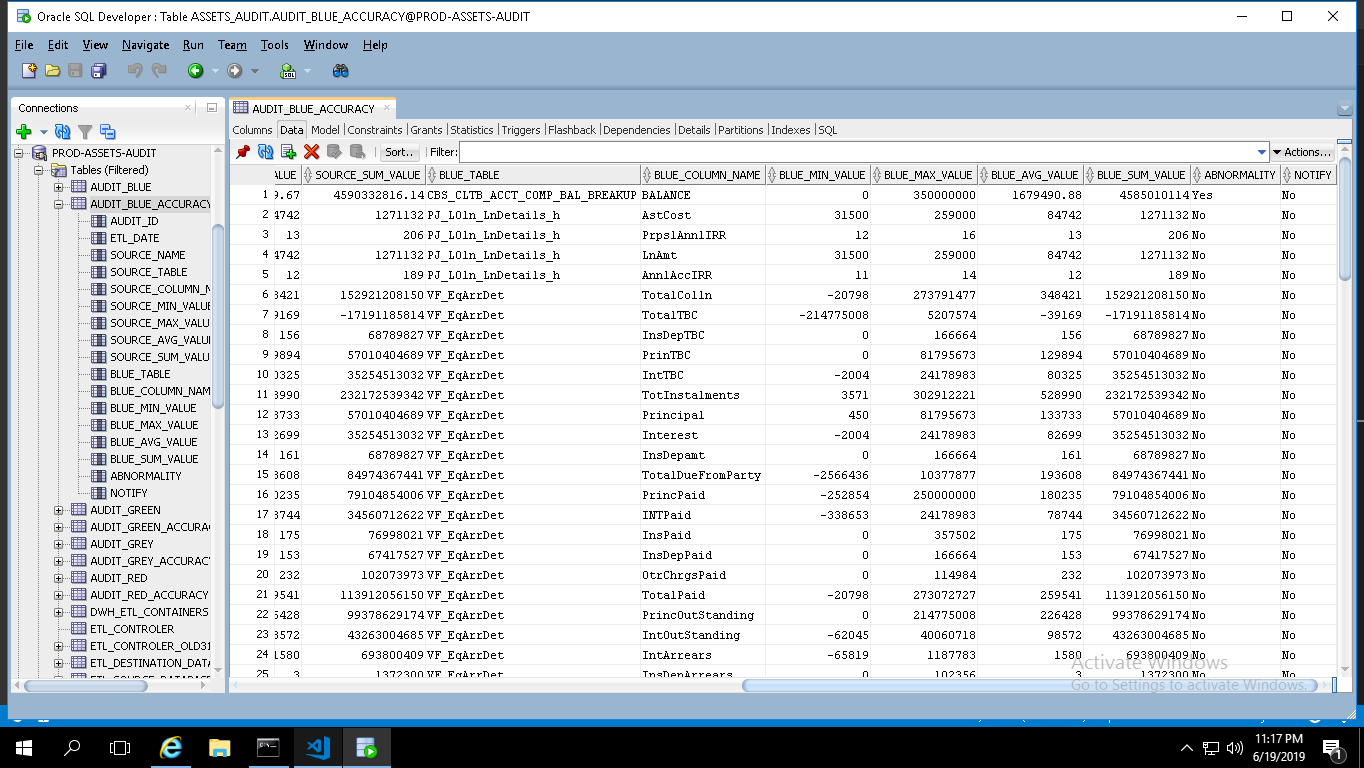


Table: audit\_blue\_accuracy

1. **audit\_accuracy\_blue** python function in audit.py is responsible for this process.
2. Same way as in audit\_blue, if any abnormality comes, a mail will be sending to concern person.

**Blue to Grey (B2G)**

**dataload\Assets\ETL\B2G\audit.py**

1. Next zone is Blue to Grey check whether The **STG1\_ETL\_CONTAINER** table is updated or not. If it is updated go for data push between Blue to Grey.
2. If not update the STG1\_ETL\_CONTAINER. Here we have to match with ETL\_DATE as ETLTIMESTAMP of incremental load.
3. In stg1\_etl\_container we can set priority of the containers by using the sort column in the stg1-container table.
4. Here we are using same query under QUERY\_INCREMENTAL column for both incremental and full load. We replace operator on the basis of full\_load condition in python.
5. Statistic condition and joins are using for updating the audit table in container wise. If you want add any new requirements in future you can add at any point of time.

The below screen short will give you the reference of the process.

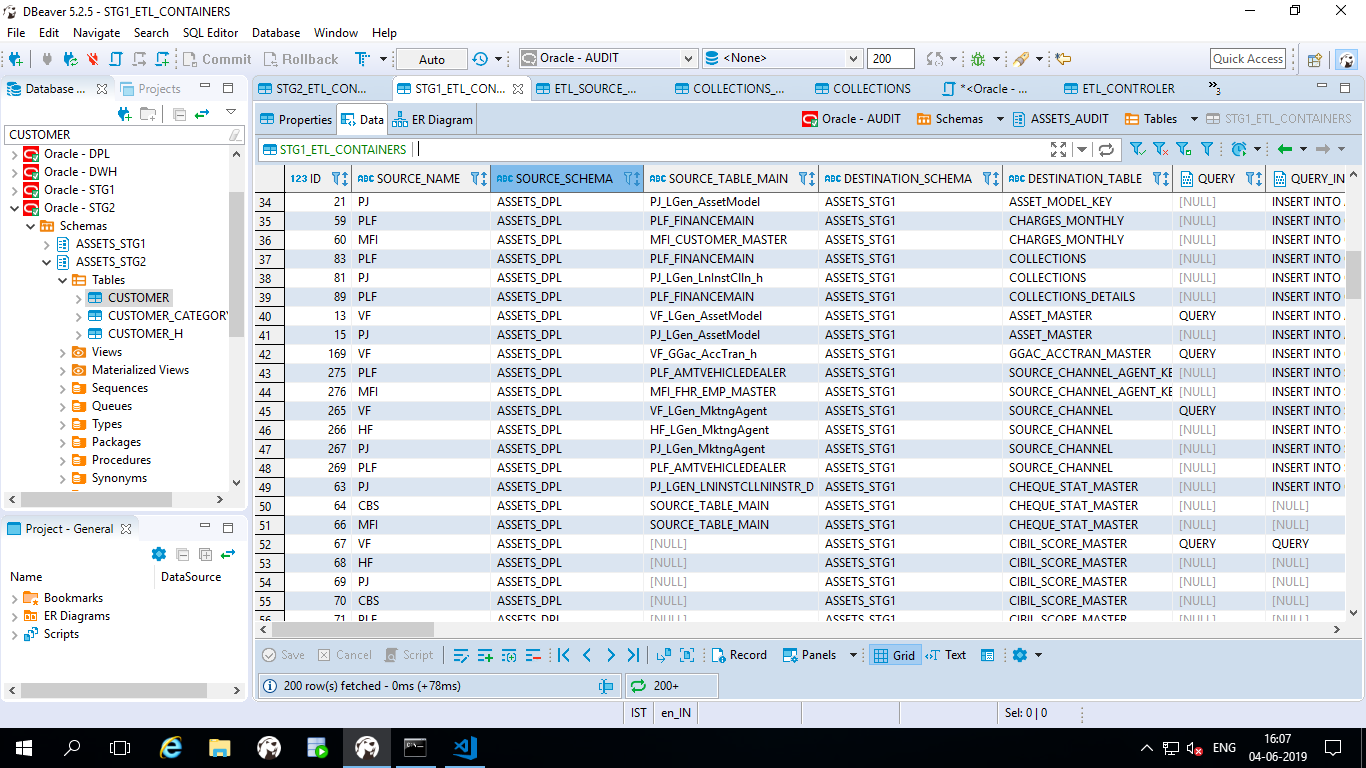
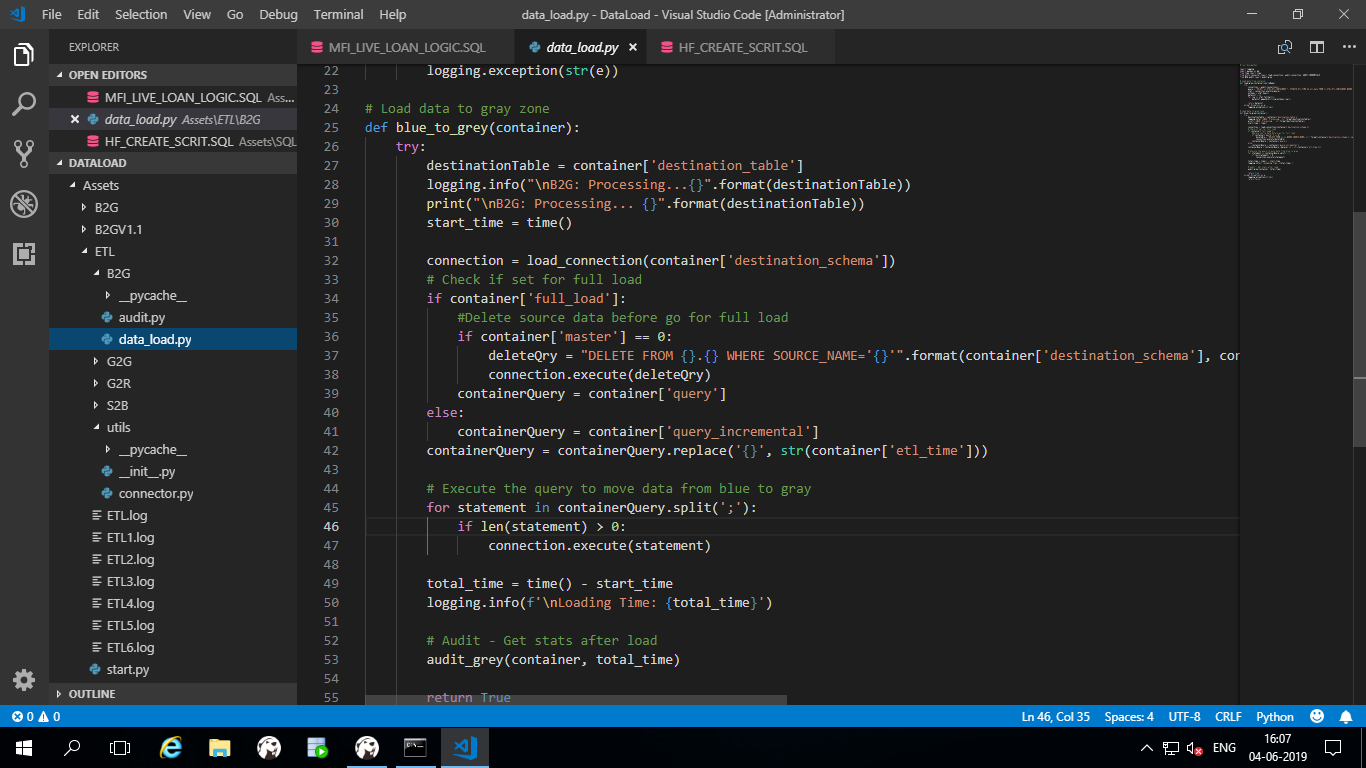


Table: stg1\_containers

1. The below code will tell you the process of pushing the data between Blue to Grey



File: dataload\Assets\ETL\B2G\data\_load.py

1. After loading all containers blue to Grey you can see all the transformed data available in the containers in STG1 Zone i.e Grey Zone and also you can check Audit Grey table in Audit schema.
2. Accuracy check is one feature implement to calculate SUM, AVG, MIN, MAX on specified columns of a table. Table audit\_accuracy\_grey is used to store this information.

* **Audit Grey:**

1. The audit.py is responsible to update audit. This audit table is to capture statistics of data movement from blue to grey.
2. By default every containers statistics will be captured here in table audit\_grey. S**tatistics** like Start time, End Time, Total Time, Read Time, Write Time, Abnormality and Status.
3. For comparison of statistics, we need to provide main source table [source\_table\_main] along with destination container. As in this movement, we have more than one source tables to prepare container. This audit will give you destination count and count of main tables used in query. That might be not match as we used joins to generate container data.

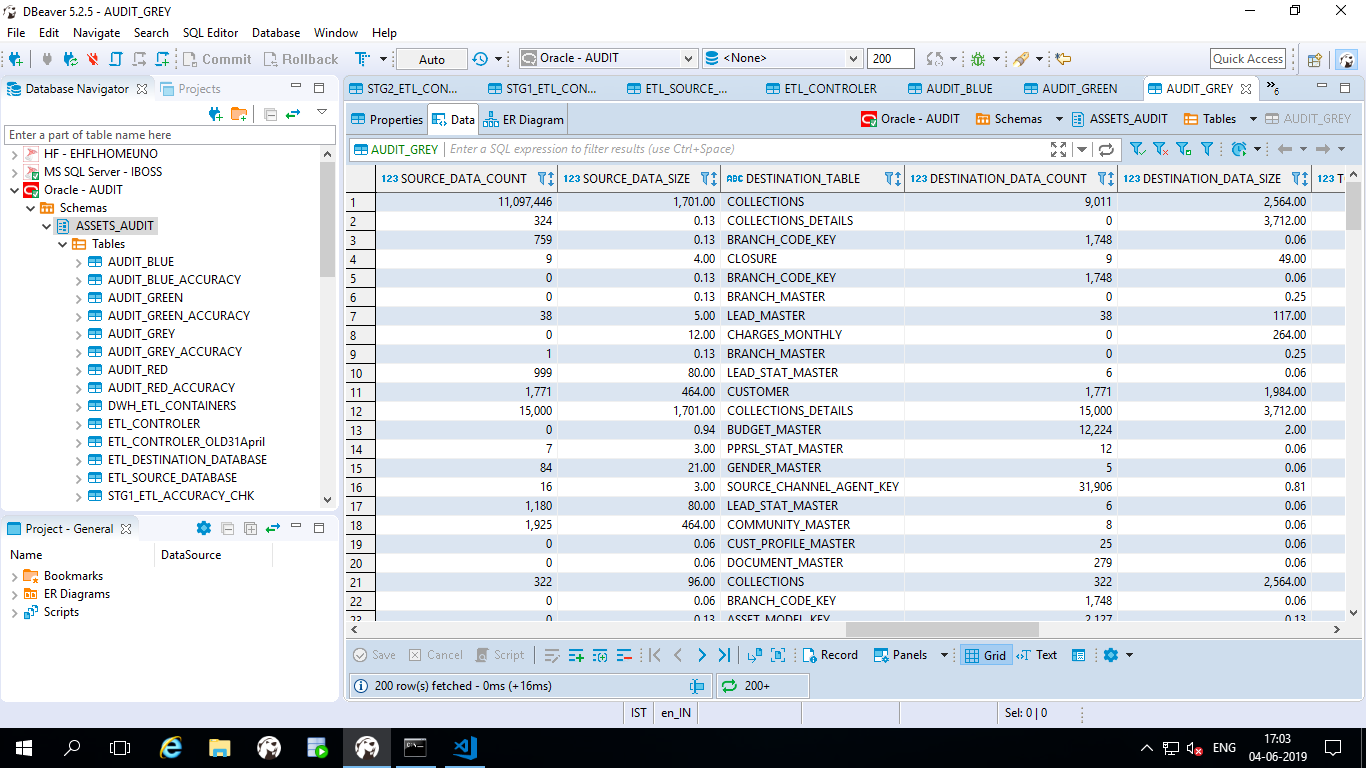


Table: audit\_grey

* **Audit Grey Accuracy:**

1. This is the process to check accuracy. In **STG1\_ETL\_CONTAINER** table you can see one column **accuracy\_chk. N**eed to set this column to ONE for the table you want to go for accuracy check.
2. If you set **accuracy\_chk** to ONE, its corresponding entry needs to be done in table **stg1\_etl\_accuracy\_chk** with container\_id. Need to specify the container columns for accuracy check and its counterpart source table with column name. You can provide Joins if required for calculation match as we know in this movement we have more than one table. You can set status to zero at any point of time if wanted to discontinue check.

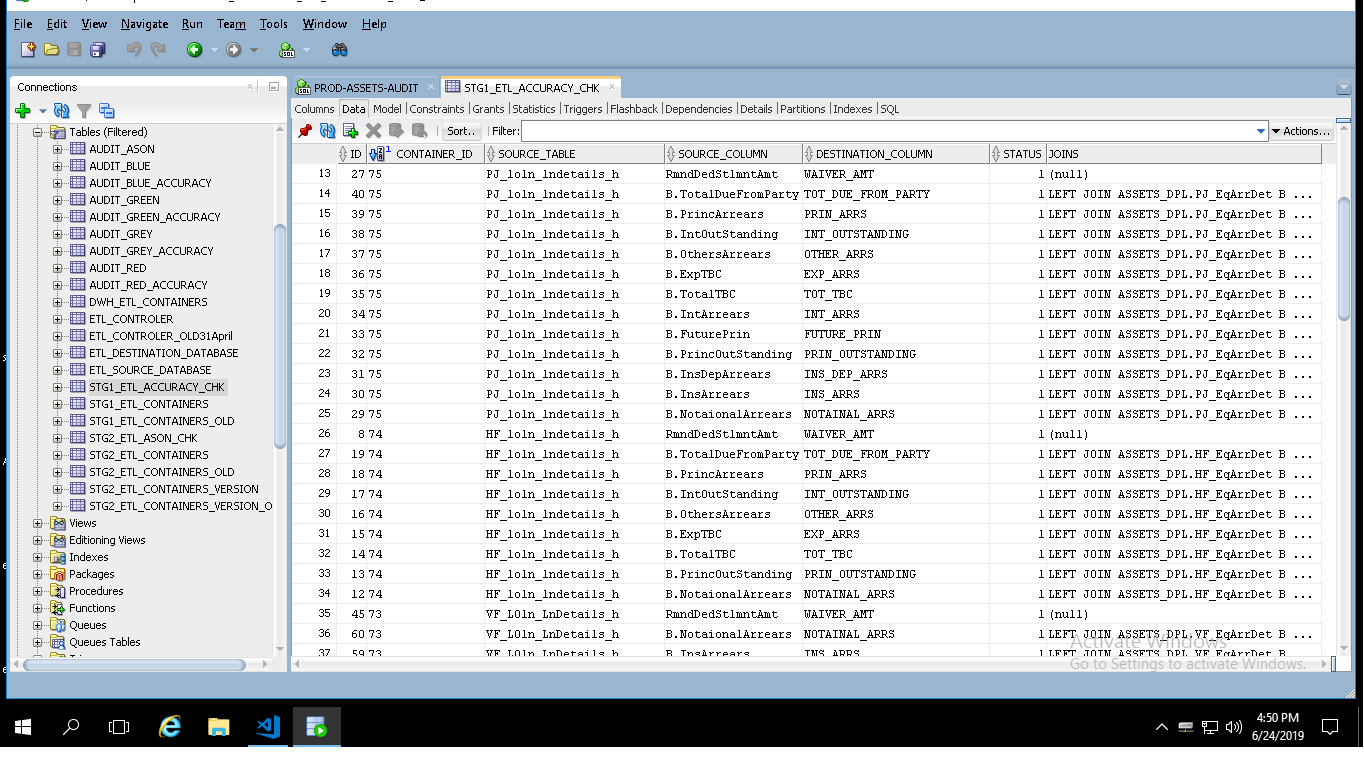
****

Table: stg1\_etl\_accuracy\_chk

1. System will go for mathematical calculation like SUM, AVG, MIN, MAX on specified columns.

Example: **SELECT MIN(PRIN\_OUTSTANDING), MAX(PRIN\_OUTSTANDING), AVG(PRIN\_OUTSTANDING), SUM(PRIN\_OUTSTANDING) FROM CLOSURE**.

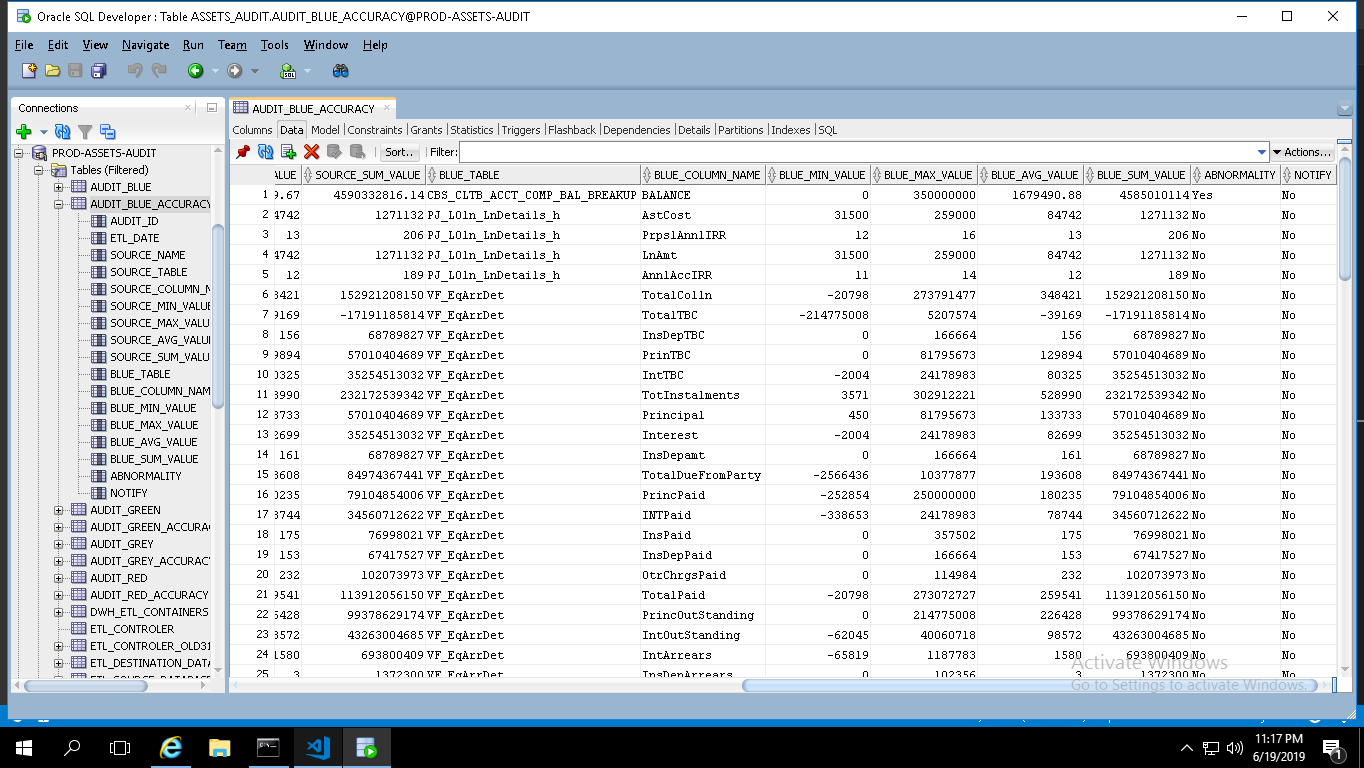
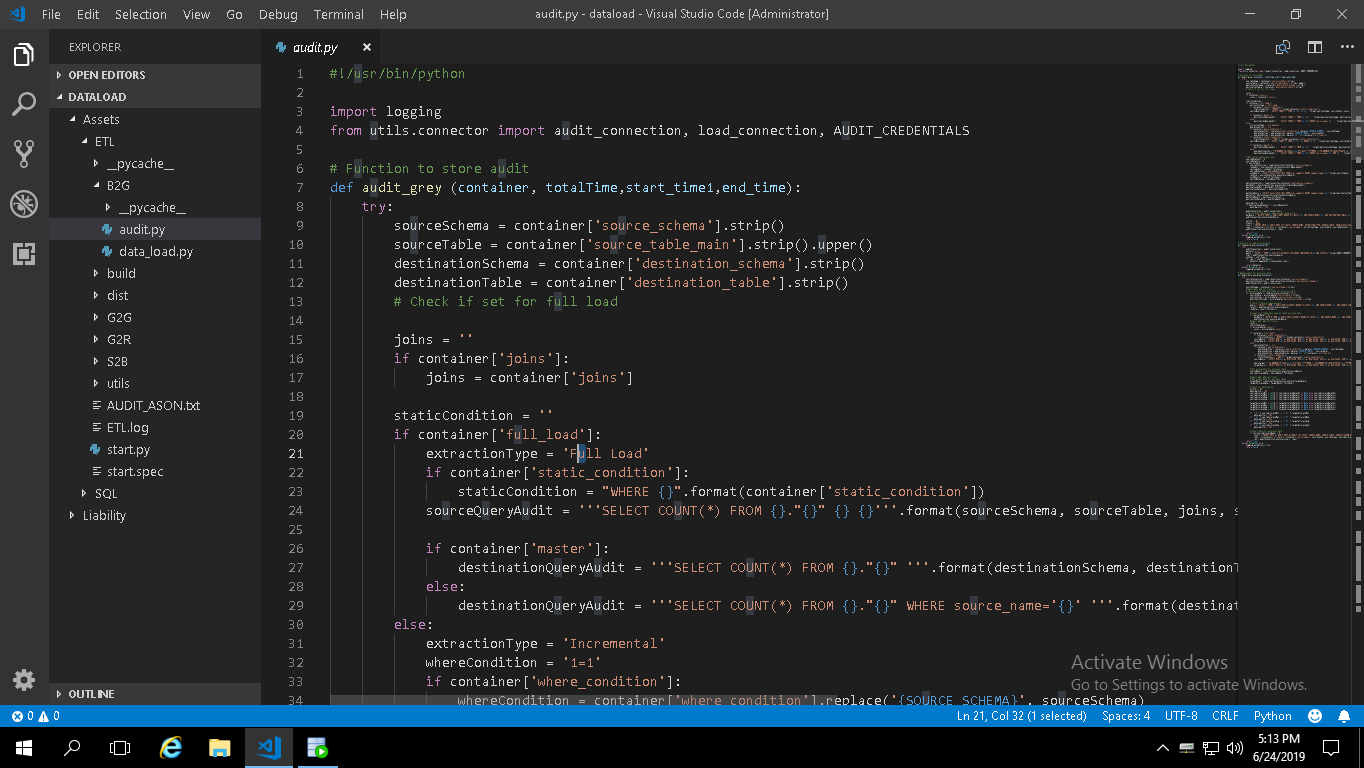
1. Make sure the column you set is numeric and present in both source and destination tables.

Table: audit\_grey\_accuracy

1. **audit\_grey\_accuracy** table given above used to store accuracy check data.
2. Here is the code responsible for B2G audit process.



File: dataload\Assets\ETL\B2G\audit.py

**Grey to Green (G2G)**

**dataload\Assets\ETL\G2G\audit.py**

Actually green zone process is like, few containers will go directly to main container and few will go Version containers. After completing the versioning of those containers we are pushing latest version of the records in to main containers.

**Main Container:**

Step1: Creating STG2\_ETL\_CONTAINER table in Audit Schema.

Step2: In G2G process will start based on the status column for the particular etl\_time.

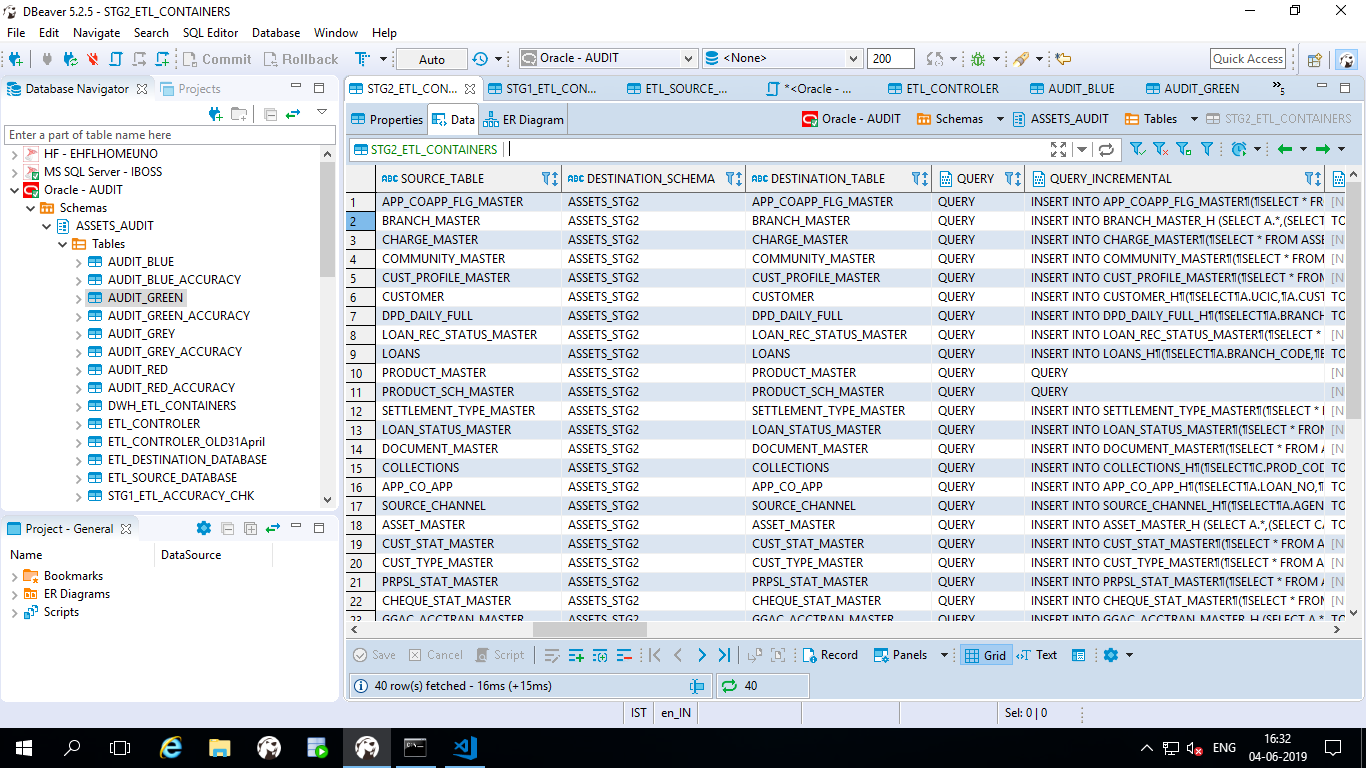
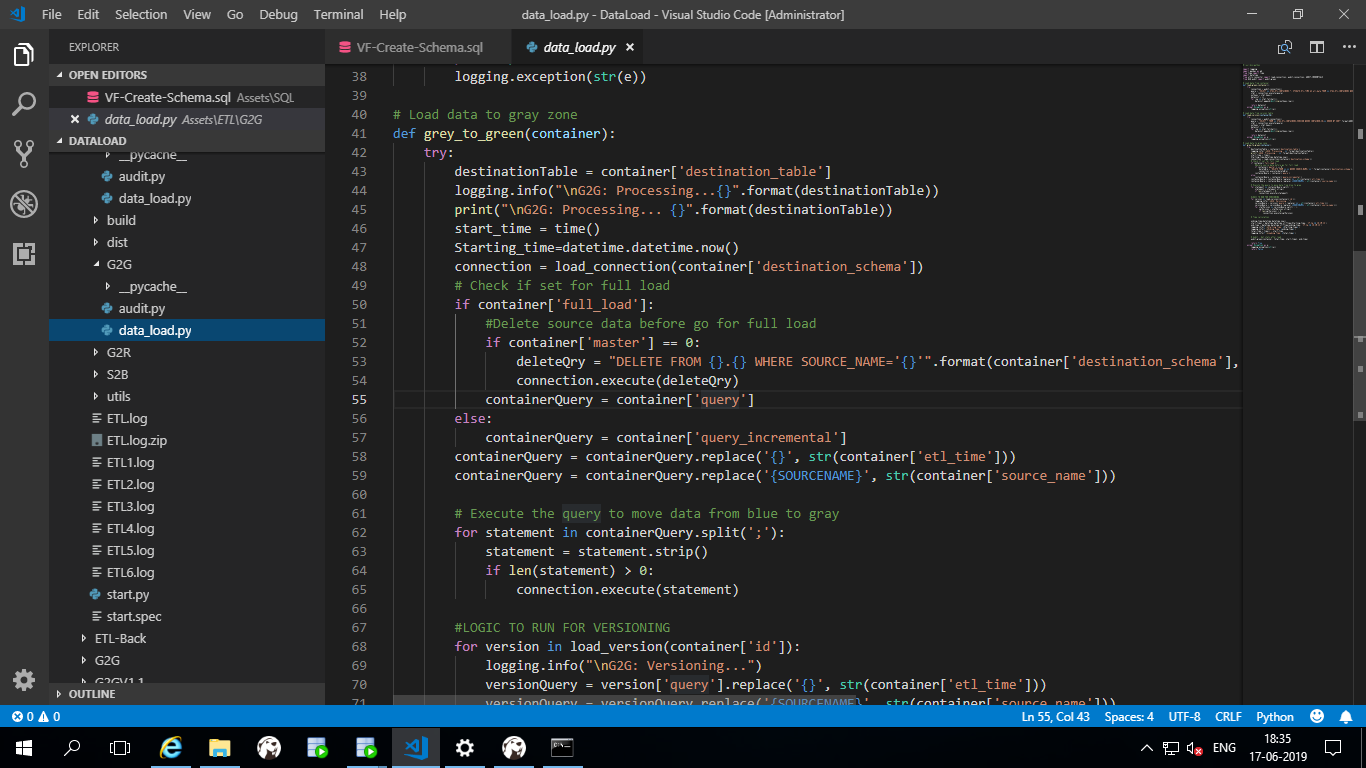


Table: STG2\_ETL\_CONTAINER

Step3: In the STG2 ETL\_CONTAINER you have to update all Queries in container wise for full load and incremental as well as update status as “ONE “which are the containers you are going to load.

**Version Container:**

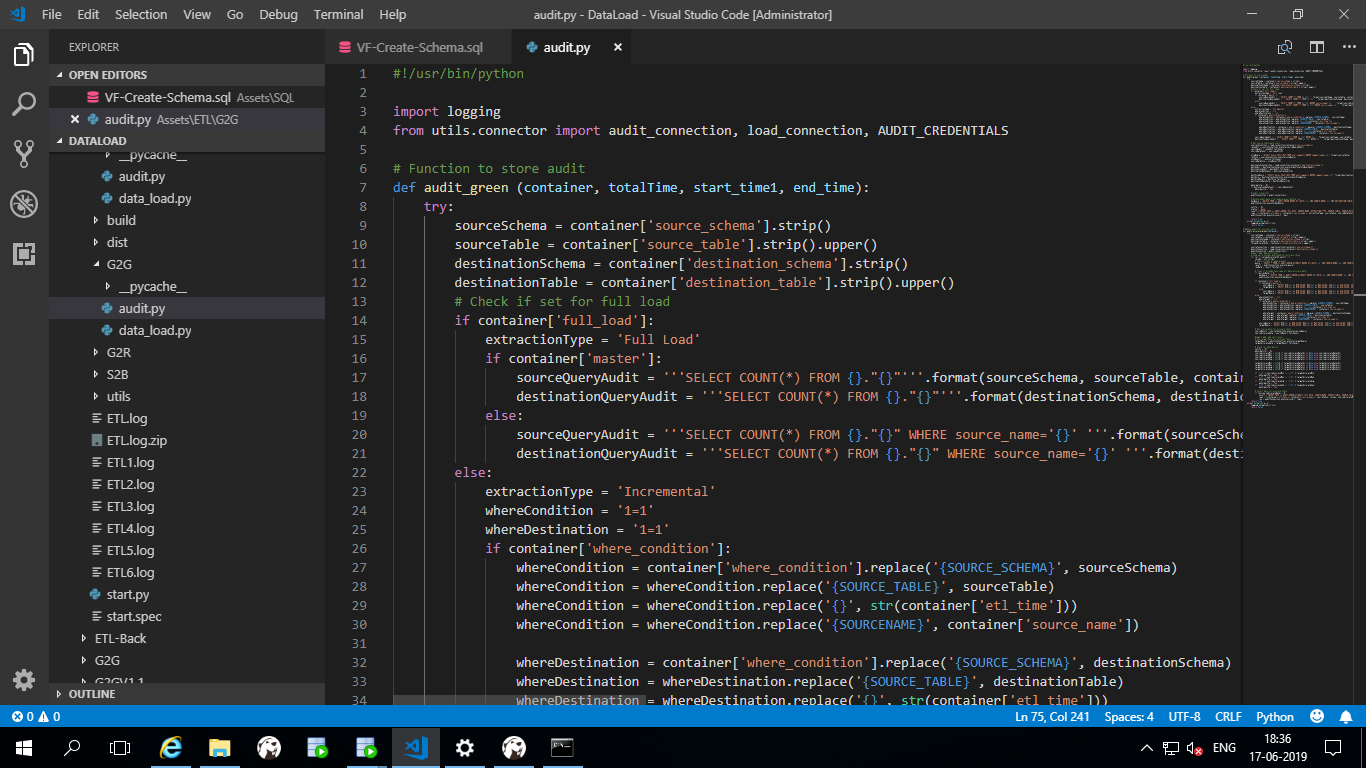
Step4: In STG2\_ETL\_CONTAINERS\_VERSION we have to update all the versioning queries in the Stg2\_etl\_containers\_version table and the main thing is mapping the version containers into main containers.

Step5: The below python code will push the data from Grey2Green.

File: dataload\Assets\ETL\G2G\data\_load.py

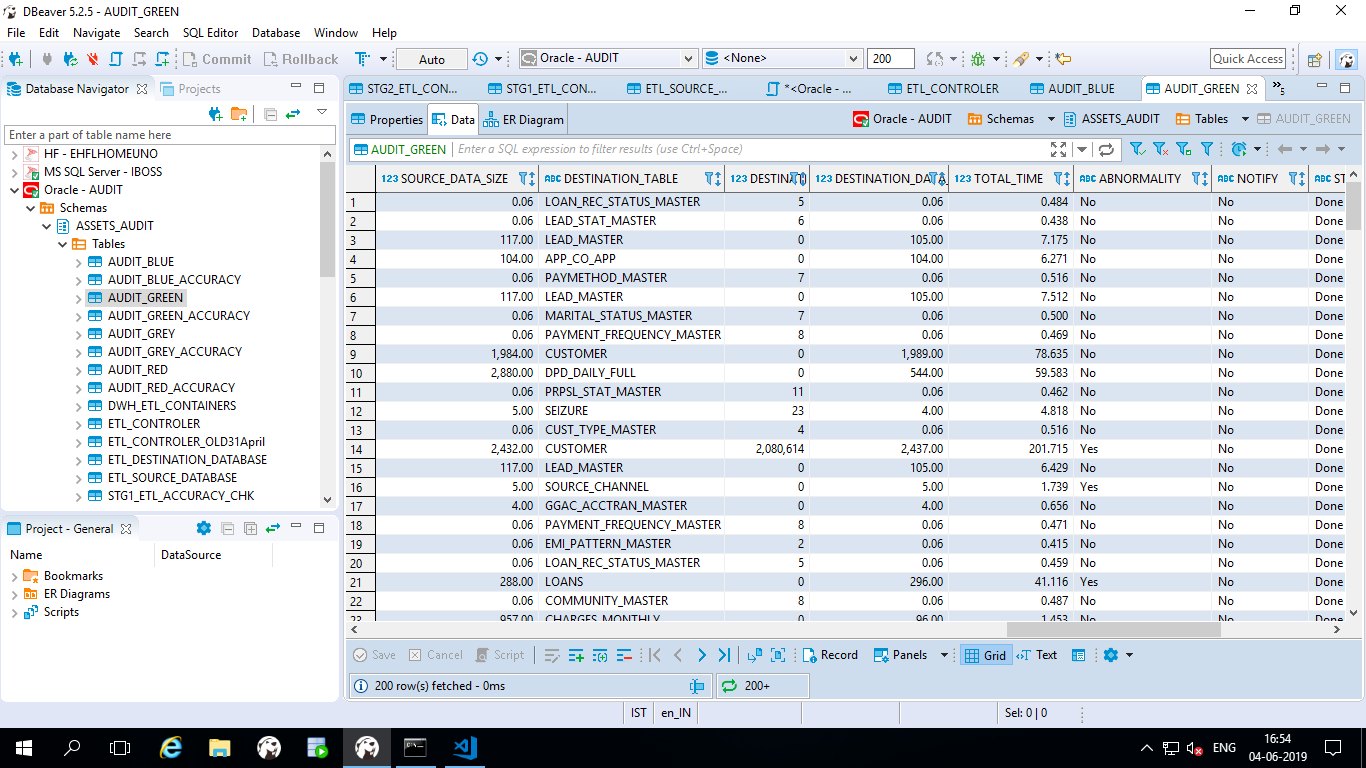
* **Audit Green:**

1. The below python code will update the Grey to Green statistics of the data.



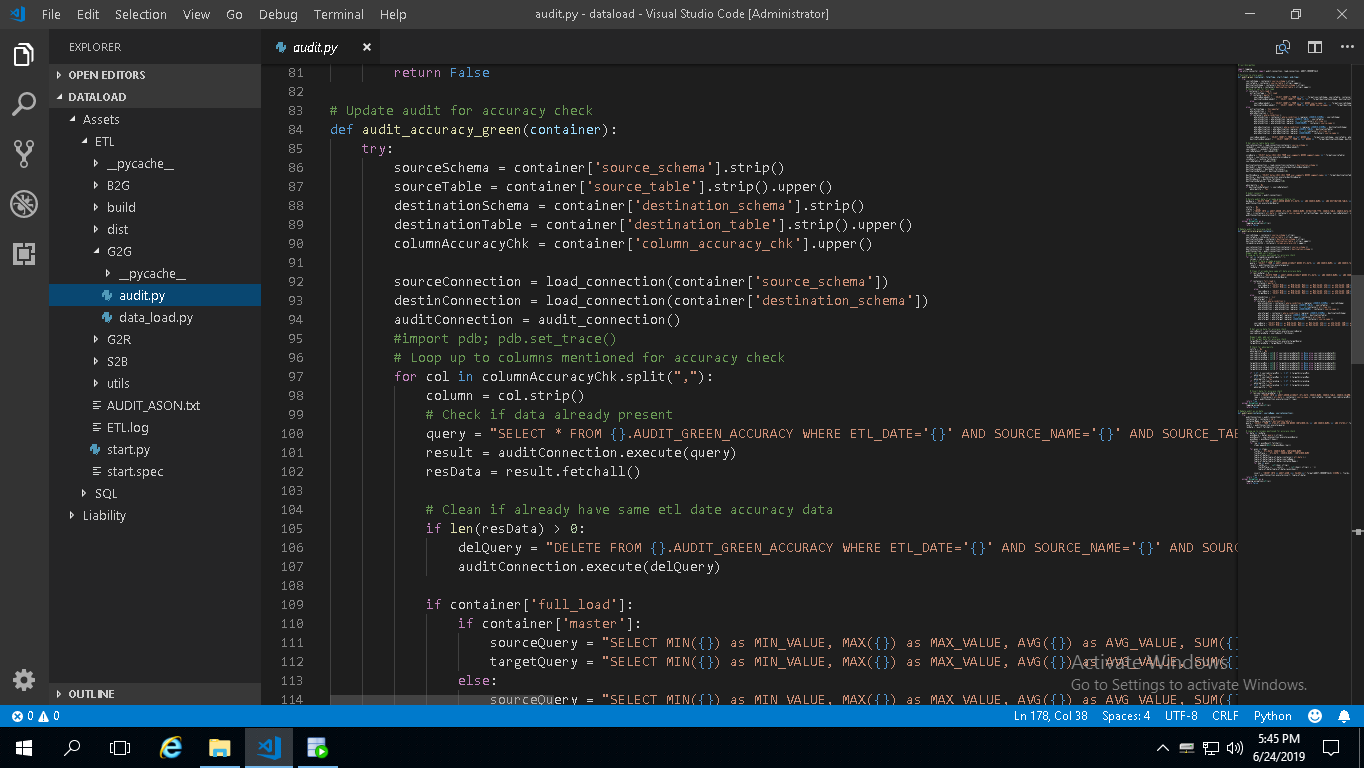
File: dataload\Assets\ETL\G2G\audit.py

1. Container wise statistics will be capture for all active containers. That will capture source and destination container data count along with read time, write time, total time, start and end time.
2. Once container data pushed to green container table you can see all the container level statistics in AUDIT\_GREEN Table.

 Table: audit\_green

* **Audit Green Accuracy:**

1. The below python code will update the Grey to Green statistics of the data.



File: dataload\Assets\ETL\G2G\audit.py

1. This is the containers column level accuracy check. Stg2\_etl\_containers table is the master table for this calculation.
2. Need to update accuracy\_chk column to **ONE** if you want to go for accuracy check. Specify columns on which you need calculation. For more the one column, use comma separated.
3. SUM, AVG, MIN, MAX on specified columns will be stored in **audit\_green\_accuracy** table.

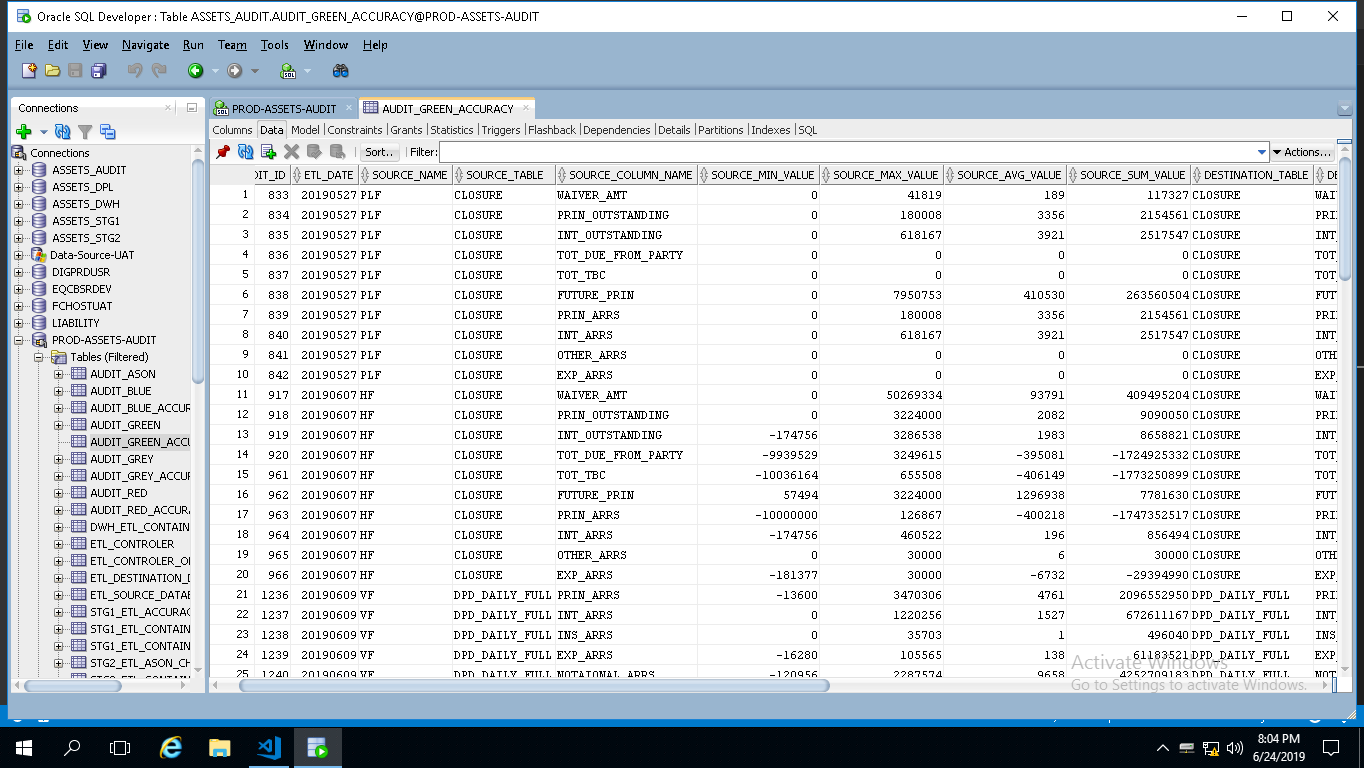


Table: audit\_green\_audit

1. Comparison will be done here between source tables column with destination tables column. As this movement if between containers, their columns must be matched. Make sure your specified column is numeric.

**Green to Red (G2R)**

**dataload\Assets\ETL\G2R\audit.py**

Under this movement, we copy as it is data from green to red zone for business purpose.

Step1: Creating the DWH\_ETL\_CONTAINERS table in audit schema.

Step2: If you’re going for full load, you have to update full load column as “one” and status should be “one”

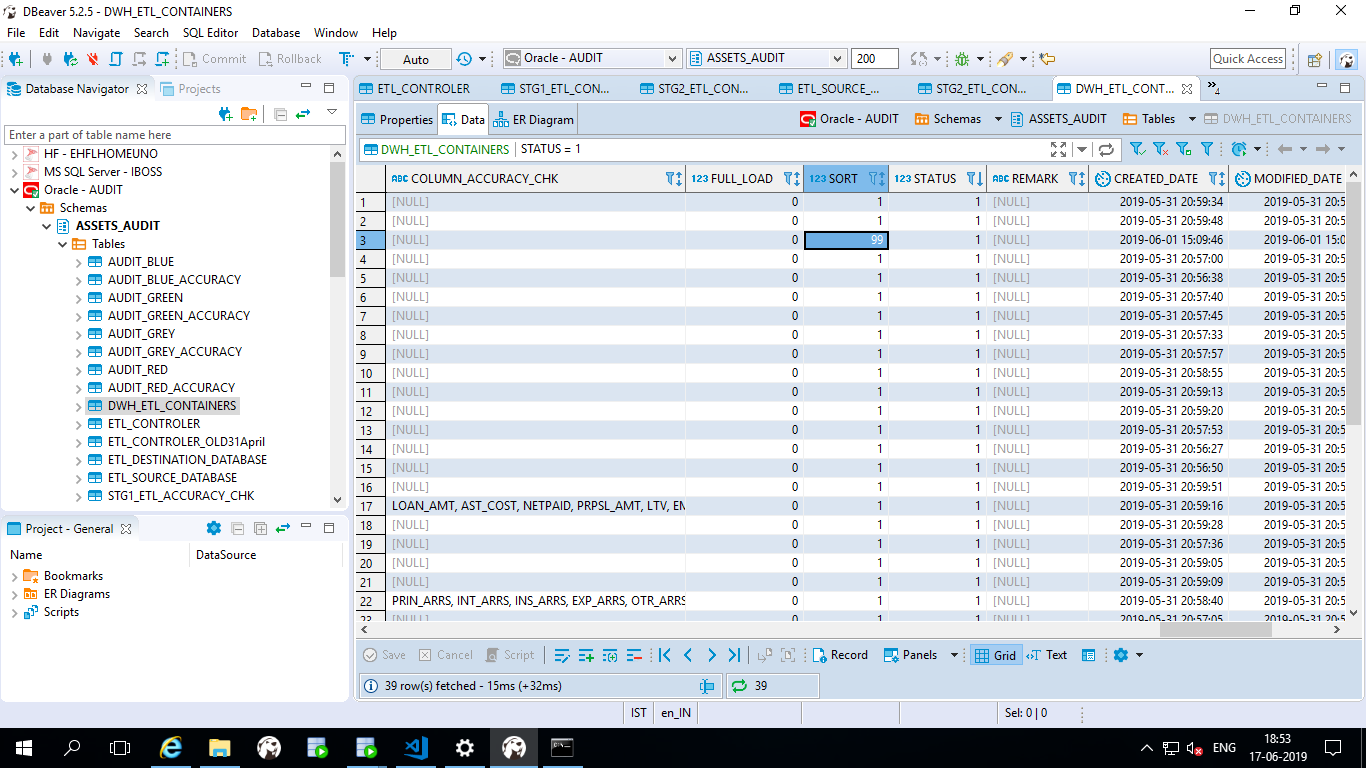
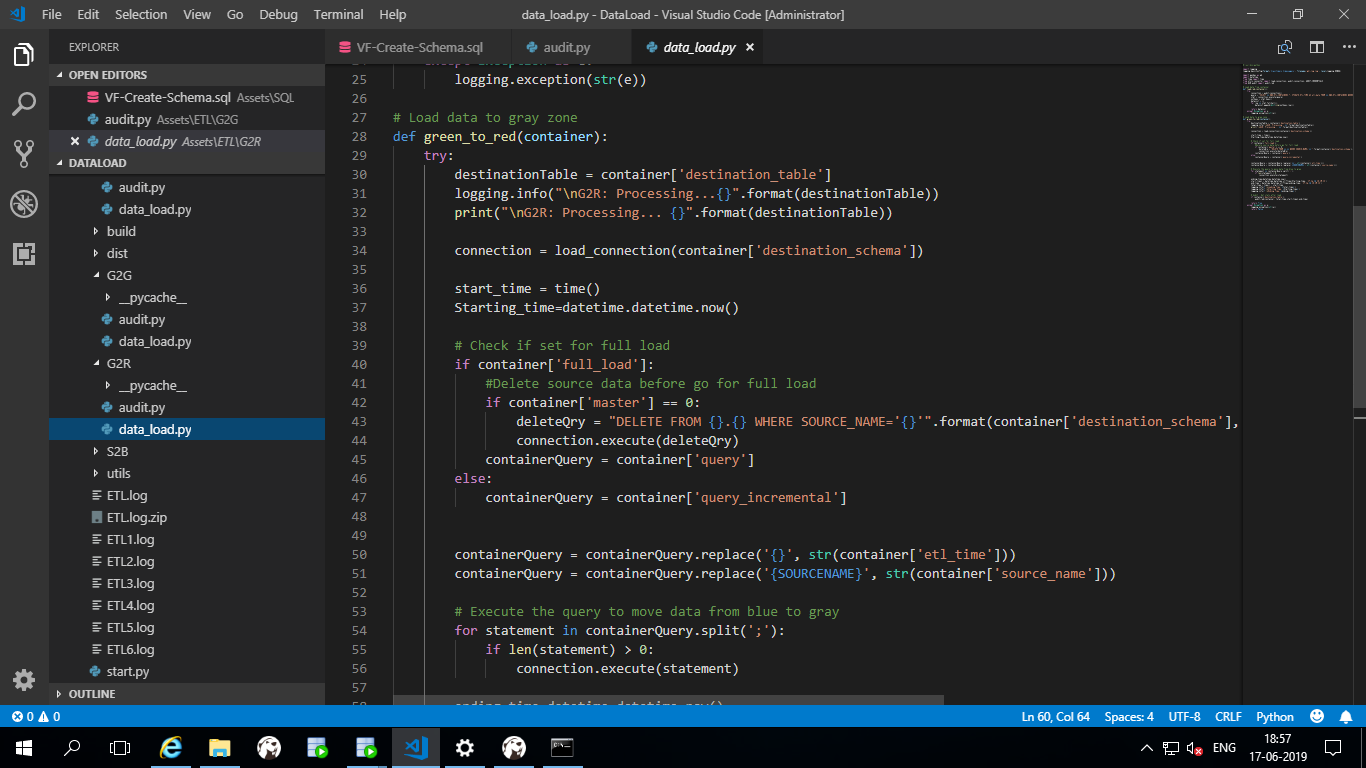


Table: DWH\_ETL\_CONTAINERS

Step3: Using the below python code just we are copying the data from Green zone to Red zone.

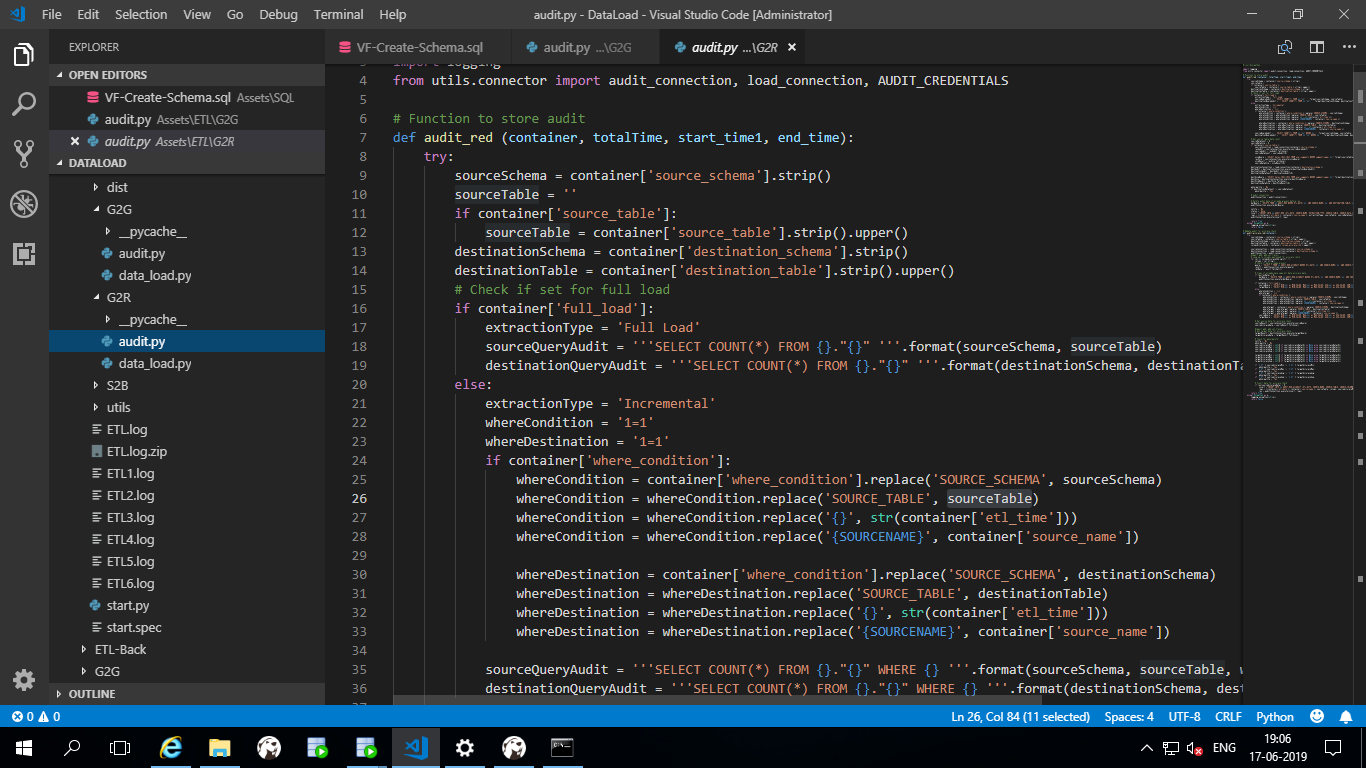
Every time the previous data will be flushed out and insert updated data in Red zone.



File: dataload\Assets\ETL\G2R\data\_load.py

* **Audit Red:**

1. The below script will update statistics of the Grey to Green zone



File: dataload\Assets\ETL\G2R\audit.py

1. After successfully completion of data pushing you can see the red zone audit results.

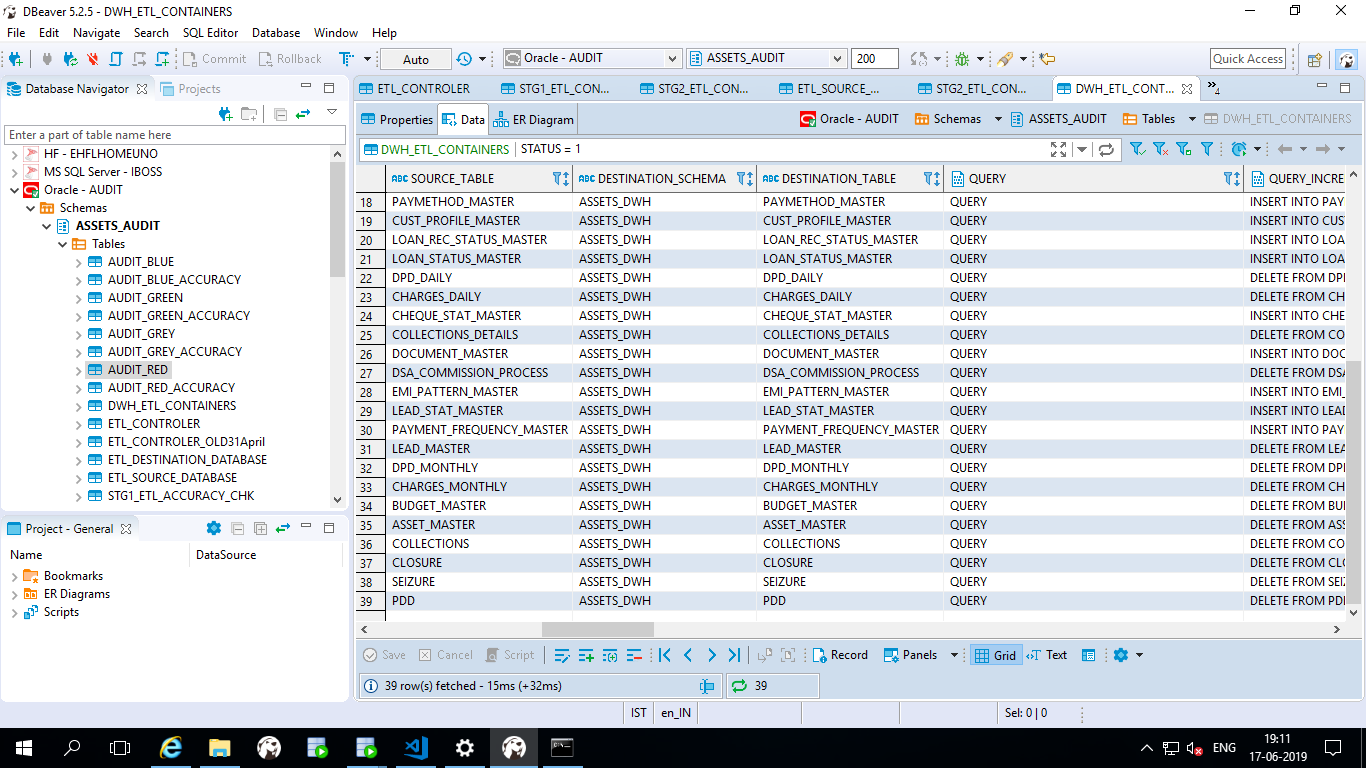
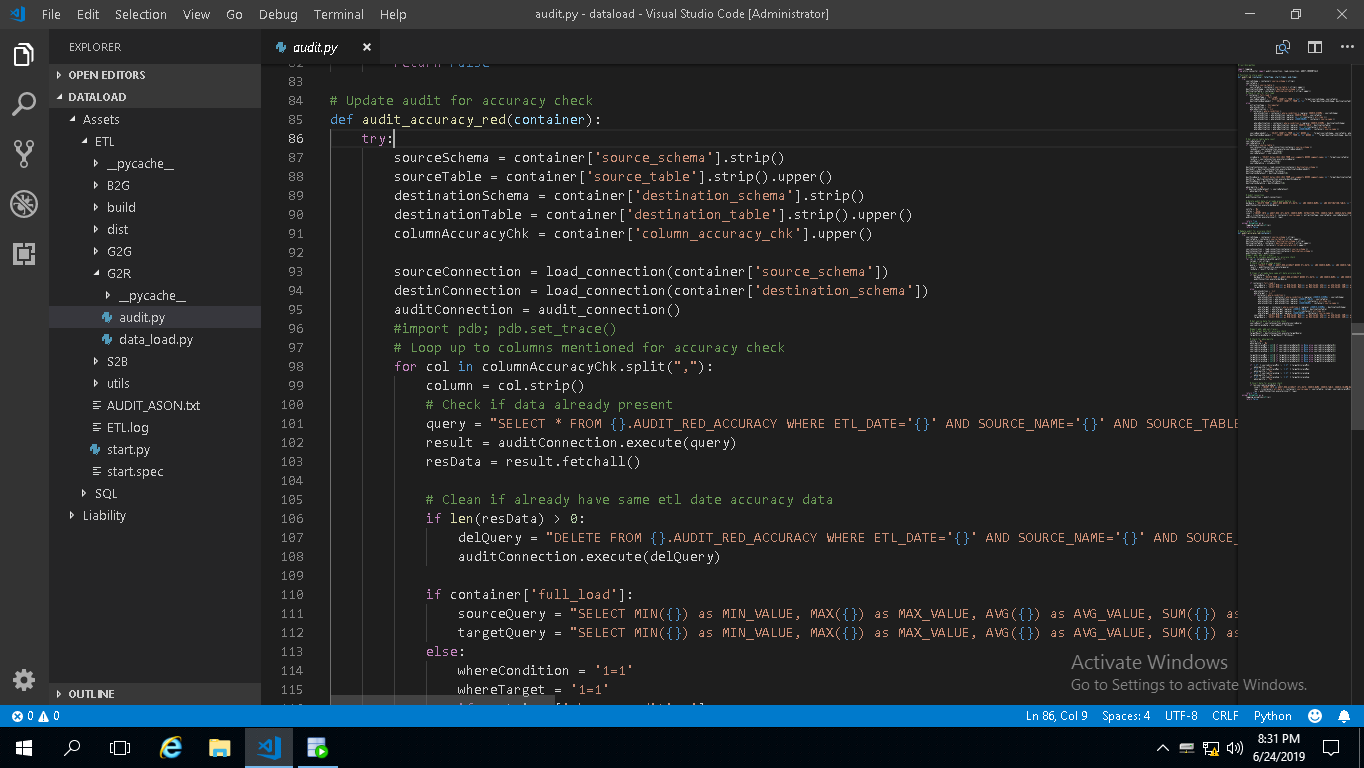


Table: audit\_red

1. This is the statistics capture during data movement. We capture start time, end time, data count on source and destination table, total time taken, read time and write time.
2. By default this statistics will be captured for all active containers.

* **Audit Red Accuracy:**

1. Here is the script audit.py (Function: audit\_accuracy\_red) used to capture accuracy data.



File: dataload\Assets\ETL\G2R\audit.py

1. This is the containers column level accuracy check. DWH\_ETL\_CONTAINERS table is the master table for this calculation.
2. Need to update accuracy\_chk column to **ONE** if you want to go for accuracy check. Specify columns on which you need calculation. For more the one column, use comma separated.
3. SUM, AVG, MIN, MAX on specified columns will be stored in **audit\_red\_accuracy** table.

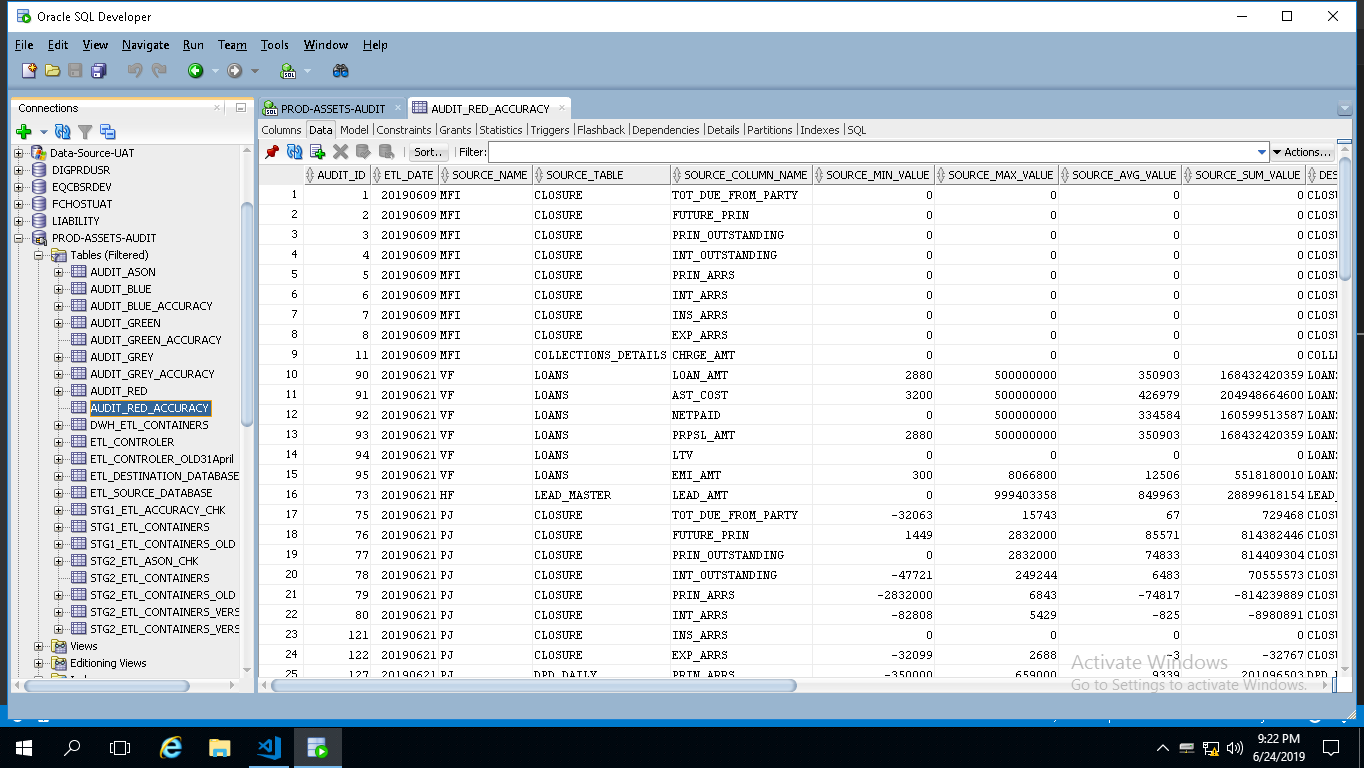


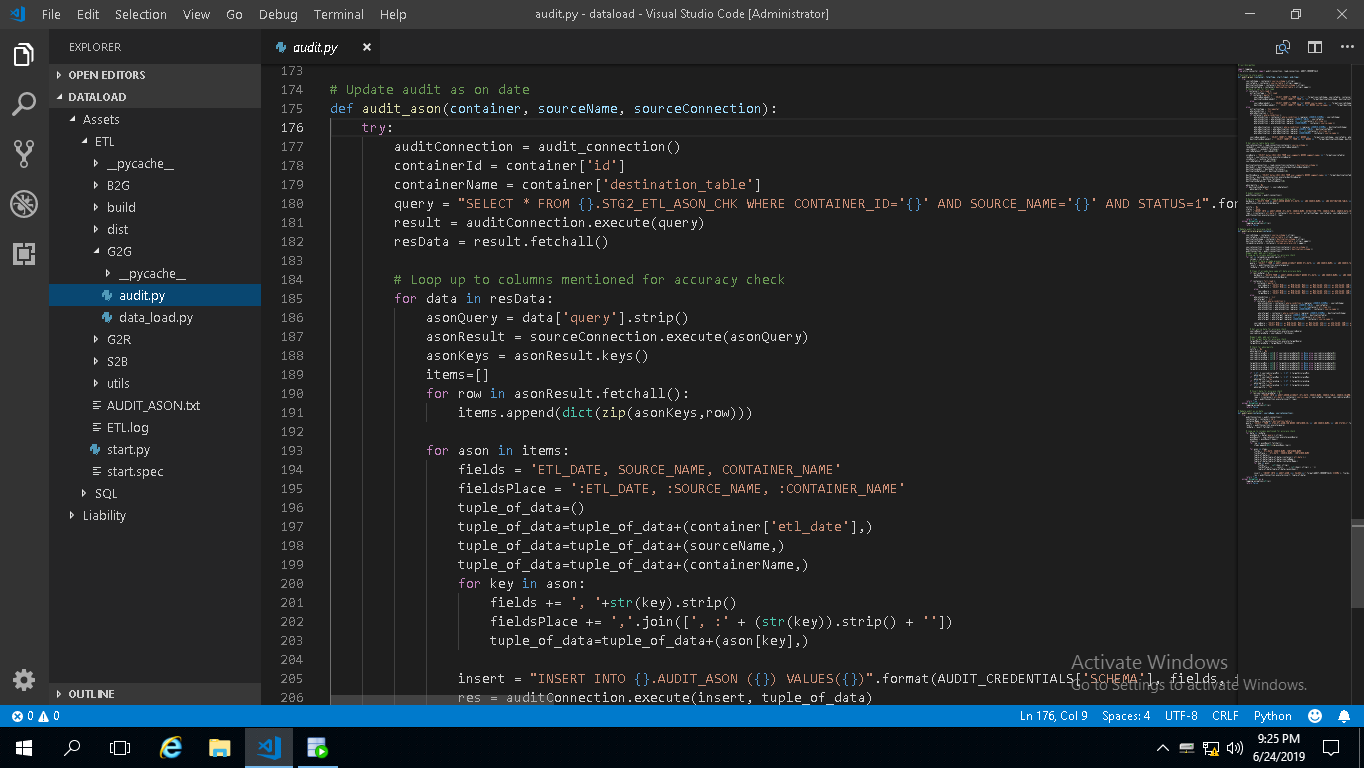
Table: audit\_red\_accuracy

1. Comparison will be done here between source tables column with destination tables column. As this movement if between containers, their columns must be matched. Make sure your specified column is numeric.

**Audit AsOn**

**dataload\Assets\ETL\G2G\audit.py**

1. Audit\_ason function is responsible ason audit calculation.



1. STG2\_ETL\_ASON\_CHK table is controller table for this audit.

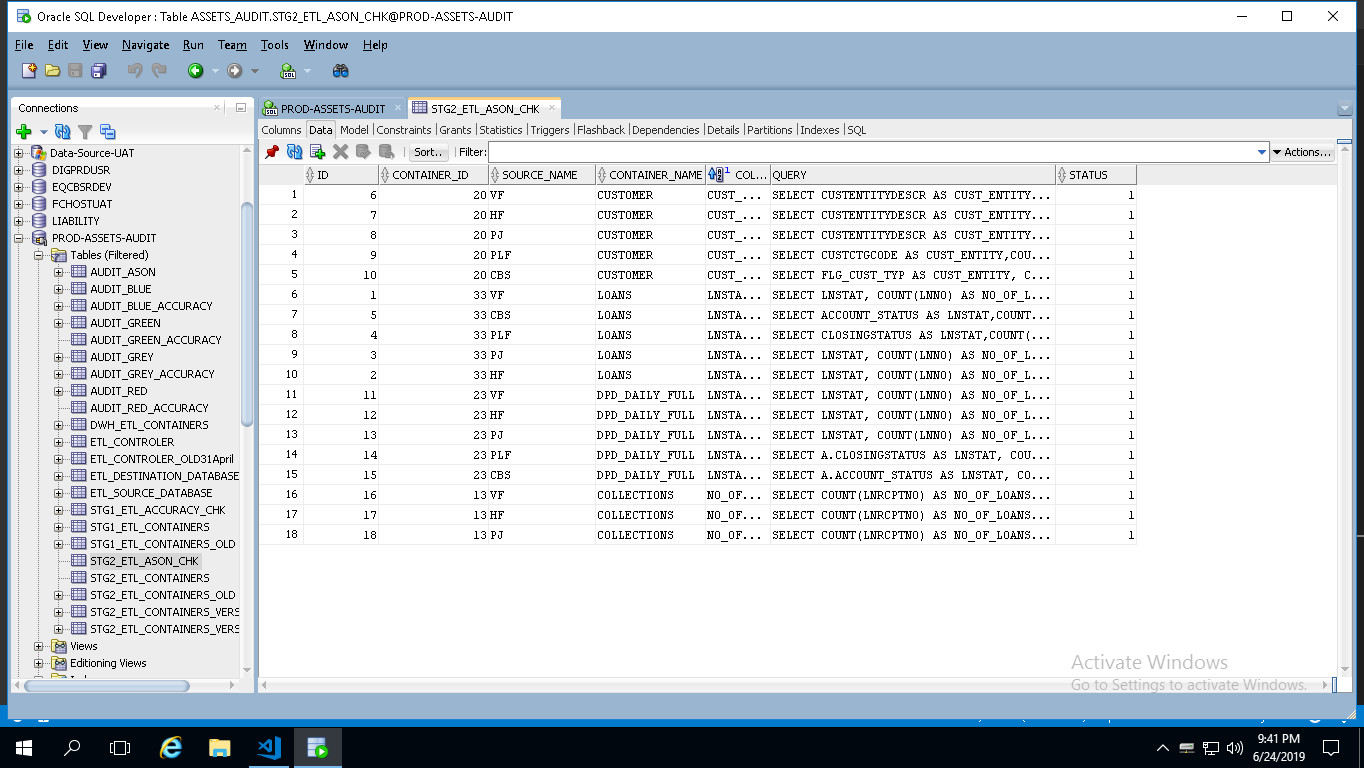


Table: STG2\_ETL\_ASON\_CHK

1. Ason query needs to be updated here along with appropriate container\_id from STG2\_ETL\_CONTAINERS table. You can change status to zero if any point of time wanted to stop ASON check.
2. Comma separated COLUMN\_NAME should be matched with alias in query.
3. After data movement till red, this audit will be captured and stored in table AUDIT\_ASON.

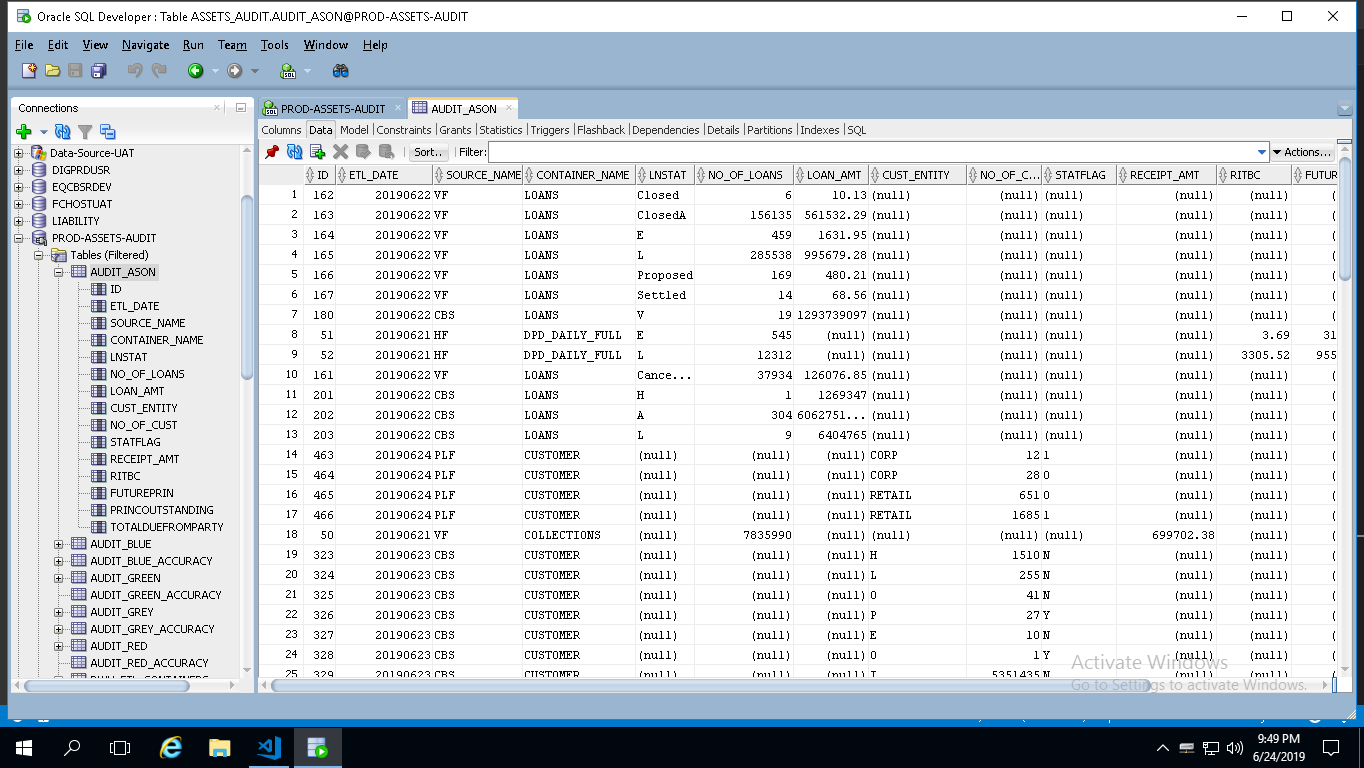


Table: audit\_ason