Xin Yu (Jason) Jiang

CIS 9440: Data Warehousing and Analytics

HW #2: ETL

ETL Process

For better understanding, please see the .ipynb (Jupyter Notebook) file or view it online here alongside this process doc.

Required Python Packages & Libraries

I chose to do this homework using only Python so there will be some packages and libraries that must be installed before moving forward.

For packages, this includes:

- google-cloud-bigquery.

For libraries, these include:

- pandas
- numpy
- bigquery (from google.cloud)
- service_account (from google.cloud)

Data Profiling

After downloading the dataset from the Google Drive link as per the instructions, I decided to profile and take a closer look at what I'm going to be working with. I read the csv and assigned it to a dataframe variable named, "data".

My main concerns here is missing data and/ or duplicate data. I need the data to be "clean" before moving forward. However, it seems like the data is already clean.

We have all the countries listed under the "country" column and average income per person listed by country, but with each year given its own column. This will be important later when we are making the **GDP Fact Table**.

Creating Dimensions

Country Dimension

I will start with the Country Dimension because we already have the “country_name” column (just needs to be renamed).

I created a new dataframe called “CountryDim” and it only included in the “country” column of the from the “data” dataframe. The column was renamed to “country_name” to better follow the dimensional model.

To make the “country_id” column, I inserted a column in the first index position that gave each country a numbered id from 1 to the number of countries there are.

Date Dimension

For the Date Dimension the initial process is similar. We already have the “year” column from the original “data” dataframe – except it must be transposed first. We can string processes together here, so I dropped all unnecessary columns first before transposing, resetting the index, and then assigning the dataframe to a variable called “DateDim”.

After renaming and formatting the data type in the “year” column I can go ahead and give each year a “date_id”. The process here is similar to what was done for the “country_id” column in the Country Dimension table. I inserted a column into the first index position that gave each year a numbered id from 1 to the total number of years there are in this dataset.

Making the decade column was slightly harder. The logic here is to contain the years (from the “years” column) into decades. For example:

1800, 1801, 1802 …, 1809 → 1800

1810, 1811, 1813 …, 1810 → 1810

. . .

2031, 2032, 2033 …, 2039 → 2030

2040 → 2040

Insertion of this column is the same as all previous insertions. As for calculating each row, I used floor division to divide each year in the “years” column by a decade (10) first. All this does is move the decimal to the left, and taking the yeah 1809 as example, becomes 180. We can then multiply 180 by 10 again to get the decade it’s in.

You’ll see that this trick works for the next decade (1810), and years perfectly because as another example following this logic: 2010 → 201 → 2010.

Creating Fact Table

Creating the fact table for this homework was slightly more difficult because we had to essentially go from wide unstacked data to stacked, narrow, and long. I needed some help with this portion and stumbled across this article on using the .melt() method to bundle related columns with common values into one column called "variable". Values that were initially in these columns are stored in another column called "value".

The "variable" column is just our bundled "year". After renaming, we can go ahead and merge together the foreign keys from our Date Dimension table and Country Dimension table.

Because we are basing the dataframe off the original dataset – we still have our "country" and "year" (was "variable" after performing .melt() method) columns. A simple left-merge allows us to bring in the date_id and country_id columns.

We see here that "value" is our average "income_per_person" – so we will rename that column and drop all unnecessary columns so that what remains reflects what is instructed in our data model.

All of this will be assigned to a variable called "GDP_fact" – our remaining table.

Downloading CSV Files

Because we will need to submit 3 datasets as csv files (GDP_fact, DateDim, and CountryDim) a simple pandas conversion method of .to_csv() allows the notebook to download the files onto your local machine when running the cell.

Deliver Dimensions and Facts to Google BigQuery

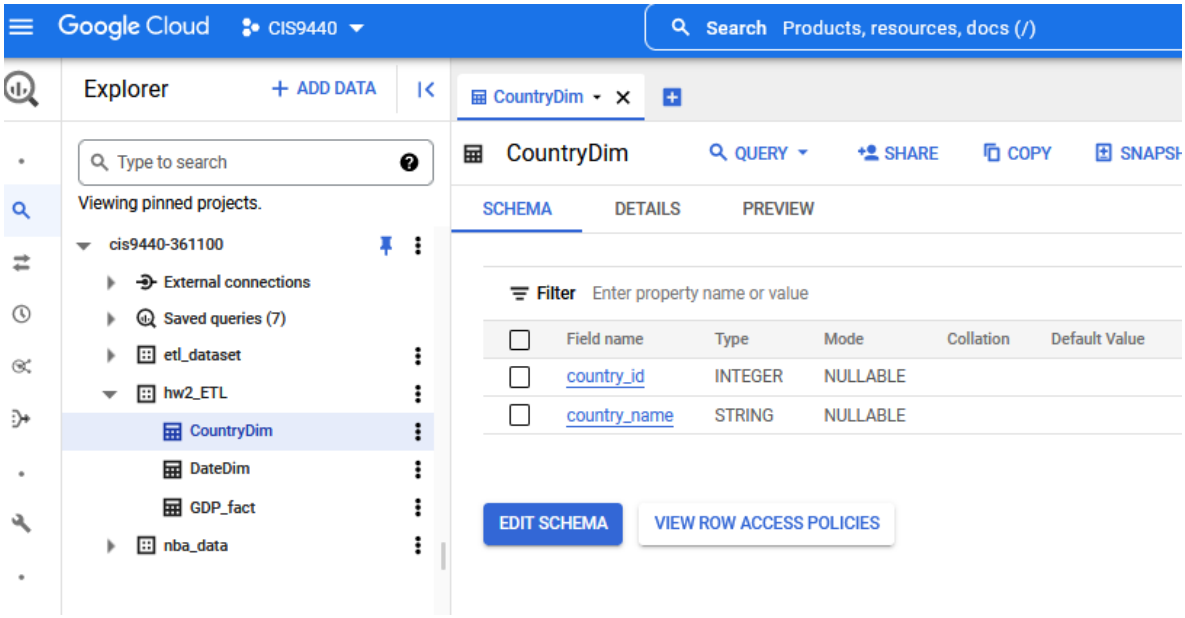
The last portion of this homework is to upload the tables into BigQuery. A lot of what's here is taken from Week 7 of our lecture.

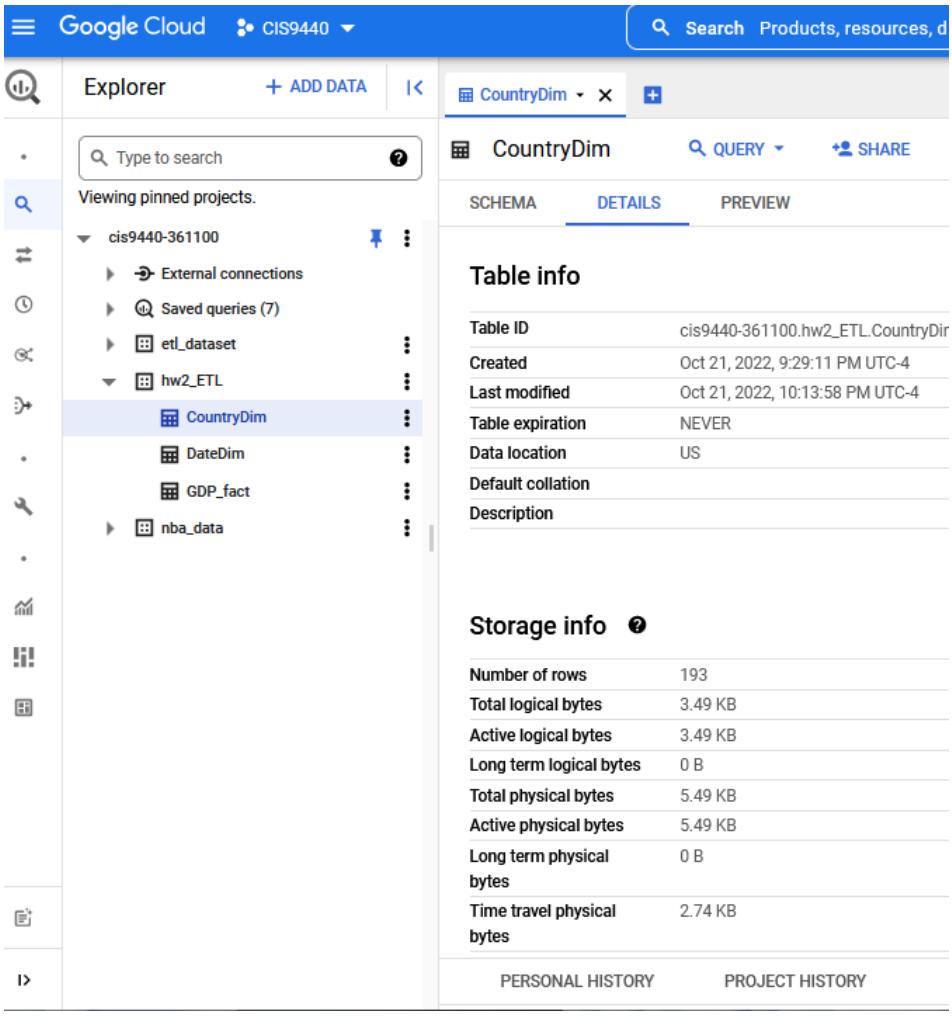
Google BigQuery Variables

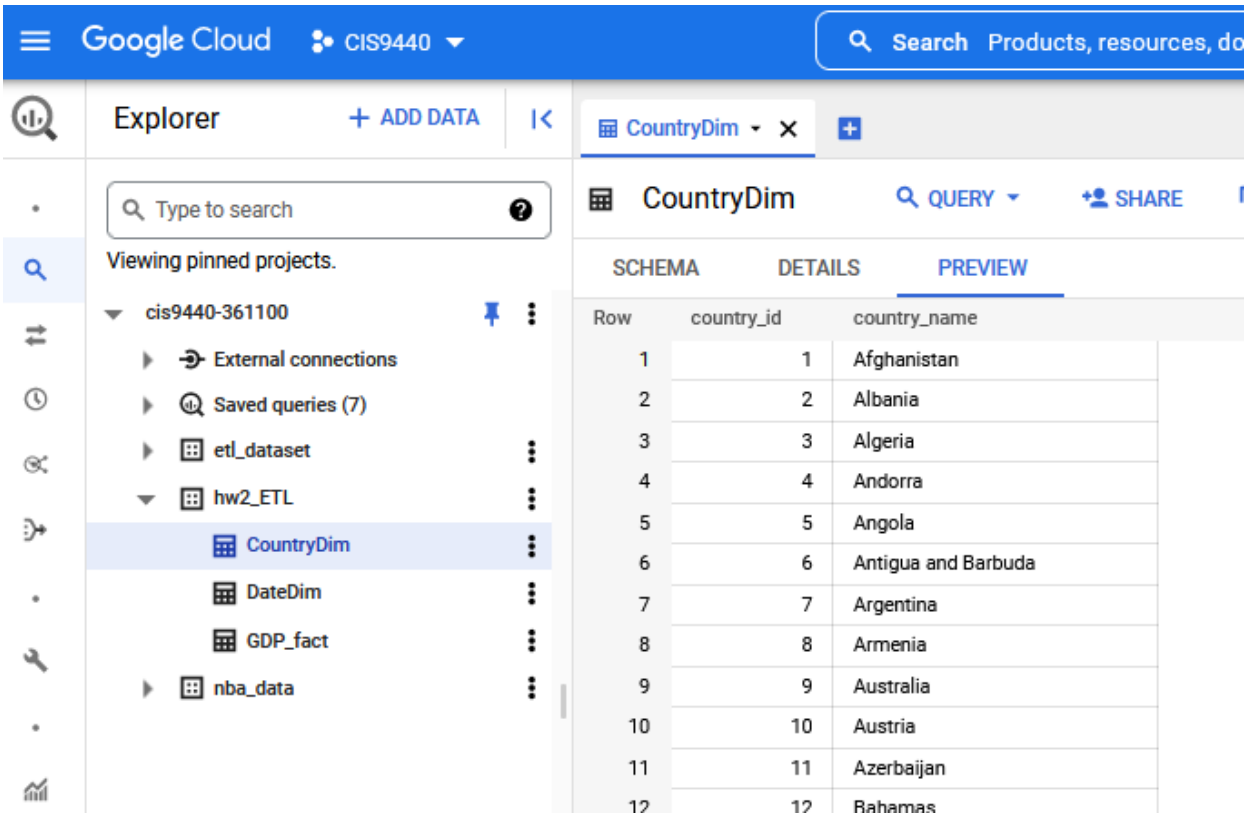
First, we establish our variables and a connection to our Google BigQuery account. The main important thing here is the .json credentials file (what was download after setting up our IAM service account on Google Cloud).

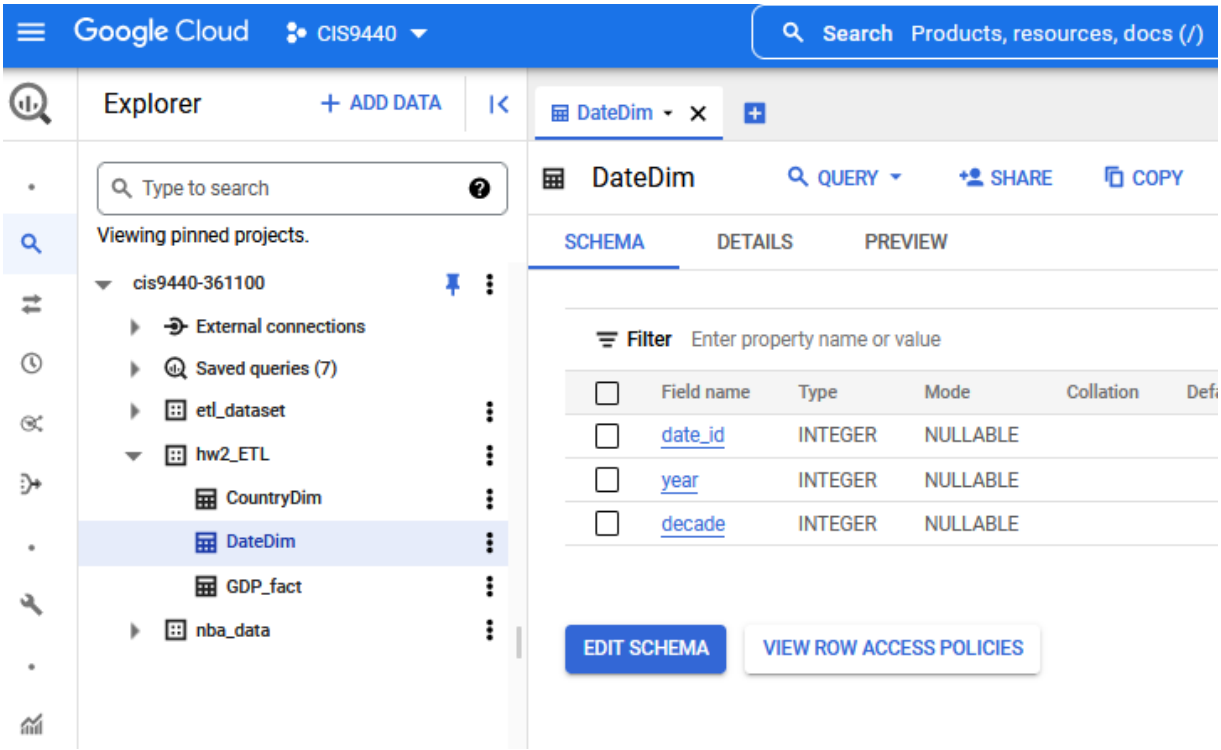
Uploading Dimensions and Facts to Google BigQuery

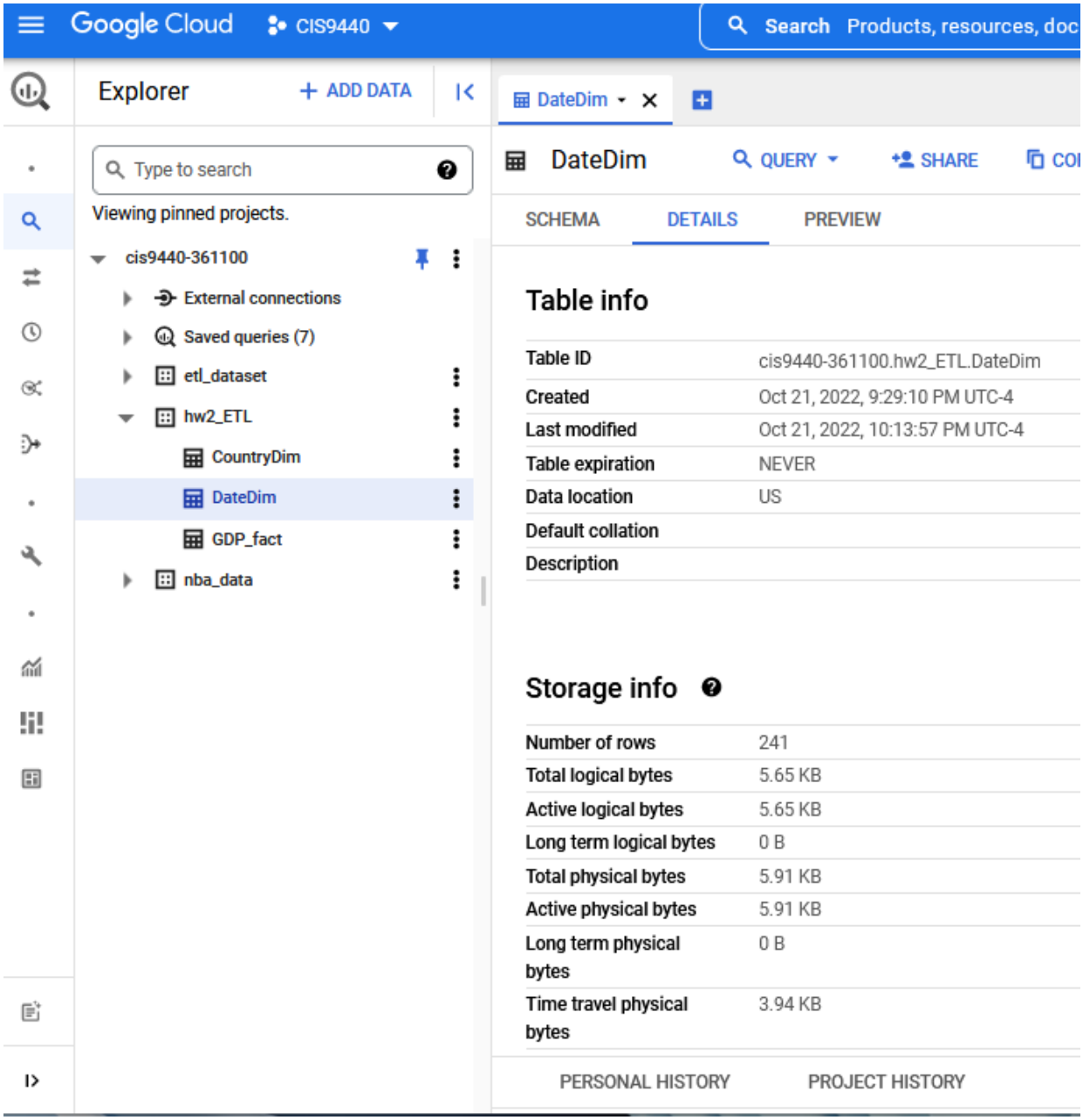
Next, we specify which project dataset to upload out tables to. I created a empty dataset specifically for this homework called "hw2_ETL" within my existing cis9440-361100 project. This is the dataset where I will upload the tables to:

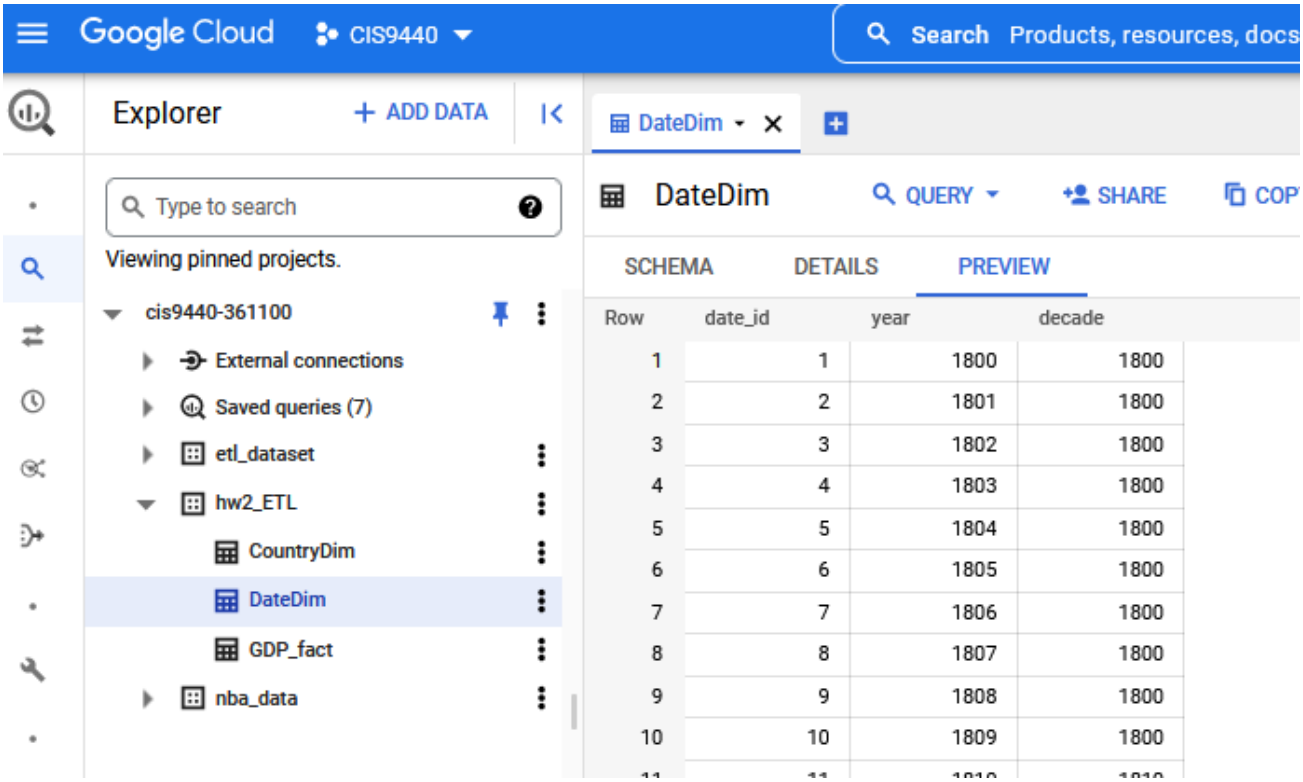

Google Cloud
CIS9440
Search Products, resources, docs (/)
Explorer
+ ADD DATA
Type to search
Viewing pinned projects.
cis9440-361100
External connections
Saved queries (7)
etl_dataset
hw2_ETL
CountryDim
DateDim
GDP_fact
nba_data
CountryDim
CountryDim
QUERY
SHARE
COPY
SNAPSH
SCHEMA
DETAILS
PREVIEW
Filter
Enter property name or value
Field name
Type
Mode
Collation
Default Value
country_id
INTEGER
NULLABLE
country_name
STRING
NULLABLE
EDIT SCHEMA
VIEW ROW ACCESS POLICIES

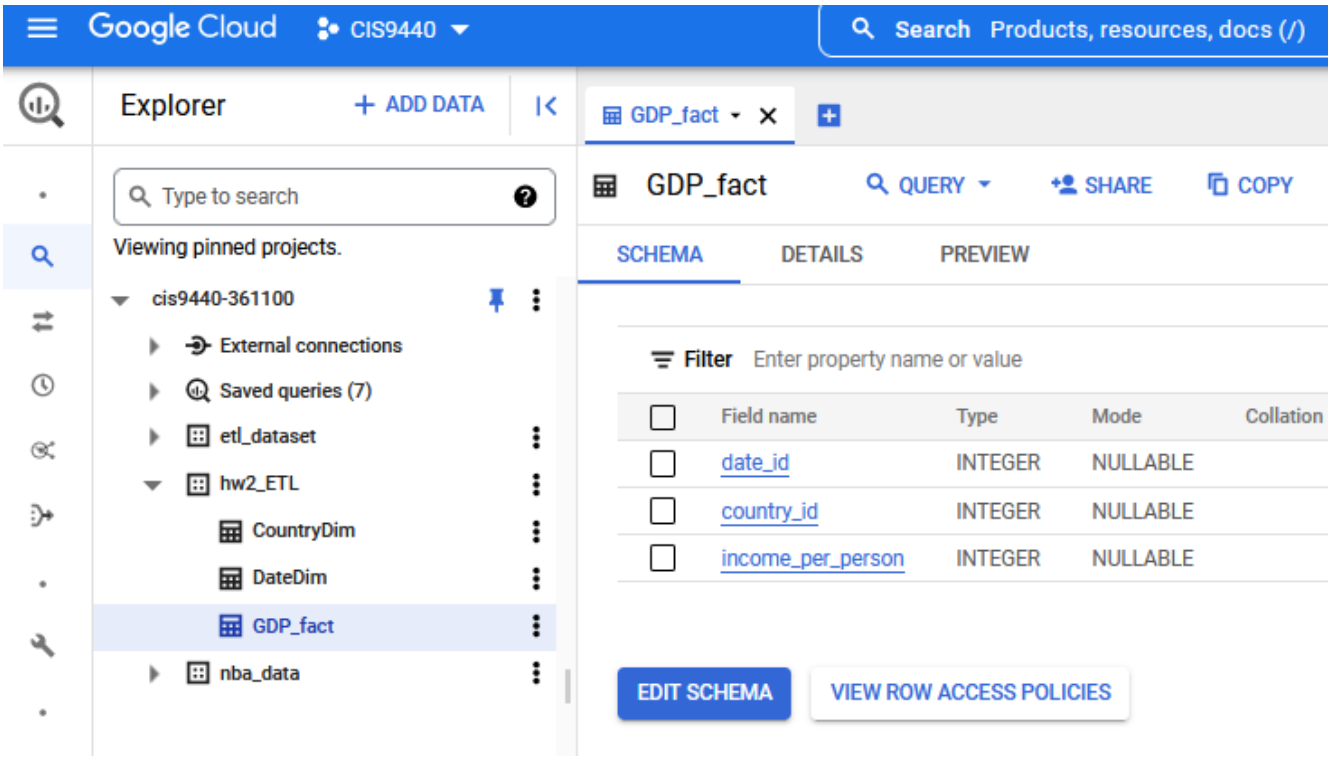

Google Cloud
CIS9440
Search Products, resources, d
Explorer
+ ADD DATA
Type to search
Viewing pinned projects.
cis9440-361100
External connections
Saved queries (7)
etl_dataset
hw2_ETL
CountryDim
DateDim
GDP_fact
nba_data
CountryDim
QUERY
SHARE
SCHEMA
DETAILS
PREVIEW
Table info
Table ID
cis9440-361100.hw2_ETL.CountryDir
Created
Oct 21, 2022, 9:29:11 PM UTC-4
Last modified
Oct 21, 2022, 10:13:58 PM UTC-4
Table expiration
NEVER
Data location
US
Default collation
Description
Storage info
Number of rows
193
Total logical bytes
3.49 KB
Active logical bytes
3.49 KB
Long term logical bytes
0 B
Total physical bytes
5.49 KB
Active physical bytes
5.49 KB
Long term physical bytes
0 B
Time travel physical bytes
2.74 KB
PERSONAL HISTORY
PROJECT HISTORY

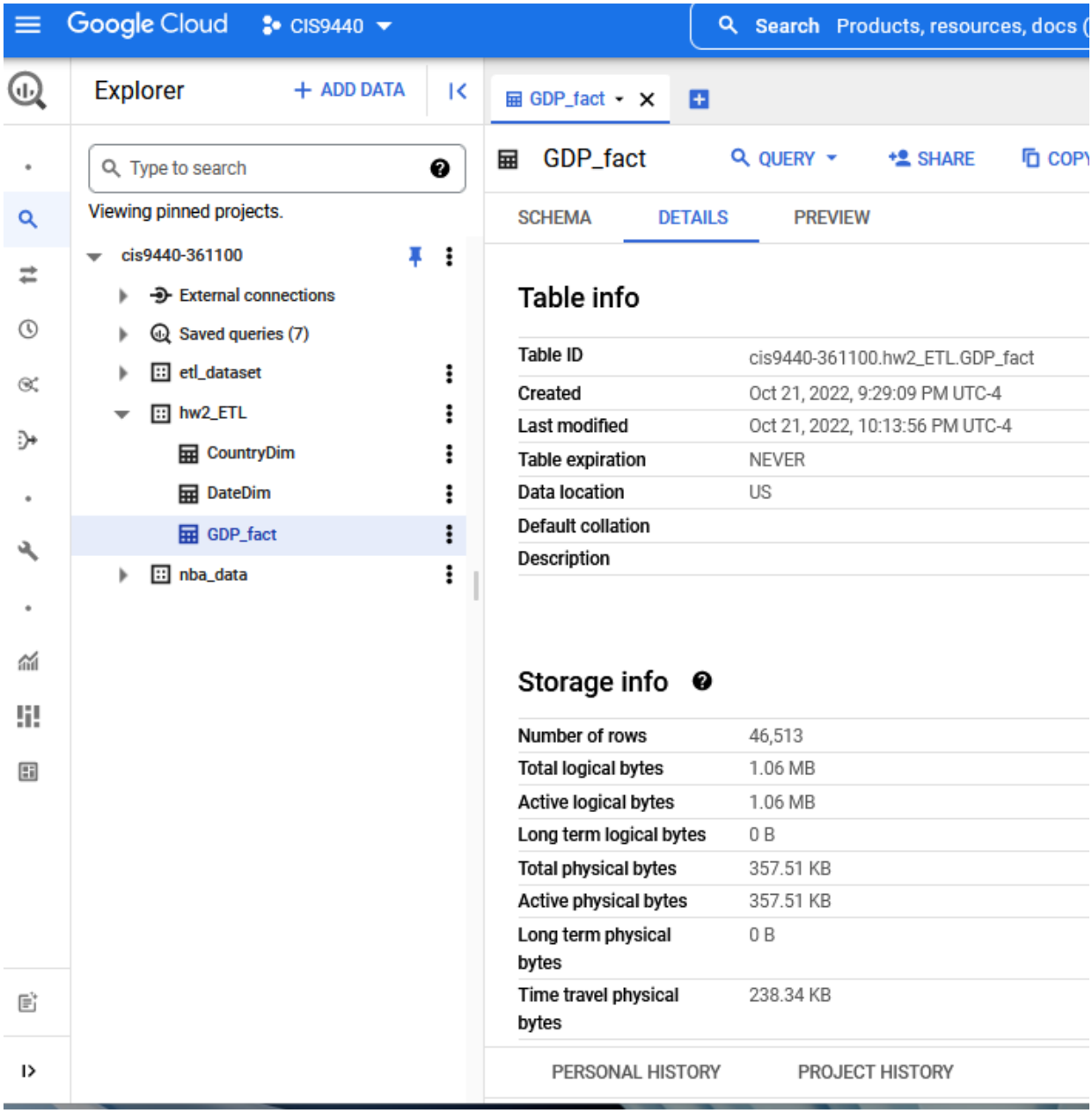

Google Cloud
CIS9440
Search Products, resources, do
Explorer
+ ADD DATA
CountryDim
Type to search
Viewing pinned projects.
cis9440-361100
External connections
Saved queries (7)
etl_dataset
hw2_ETL
CountryDim
DateDim
GDP_fact
nba_data
CountryDim
QUERY
SHARE
SCHEMA
DETAILS
PREVIEW
Row country_id country_name
1 1 Afghanistan
2 2 Albania
3 3 Algeria
4 4 Andorra
5 5 Angola
6 6 Antigua and Barbuda
7 7 Argentina
8 8 Armenia
9 9 Australia
10 10 Austria
11 11 Azerbaijan
12 12 Bahamas

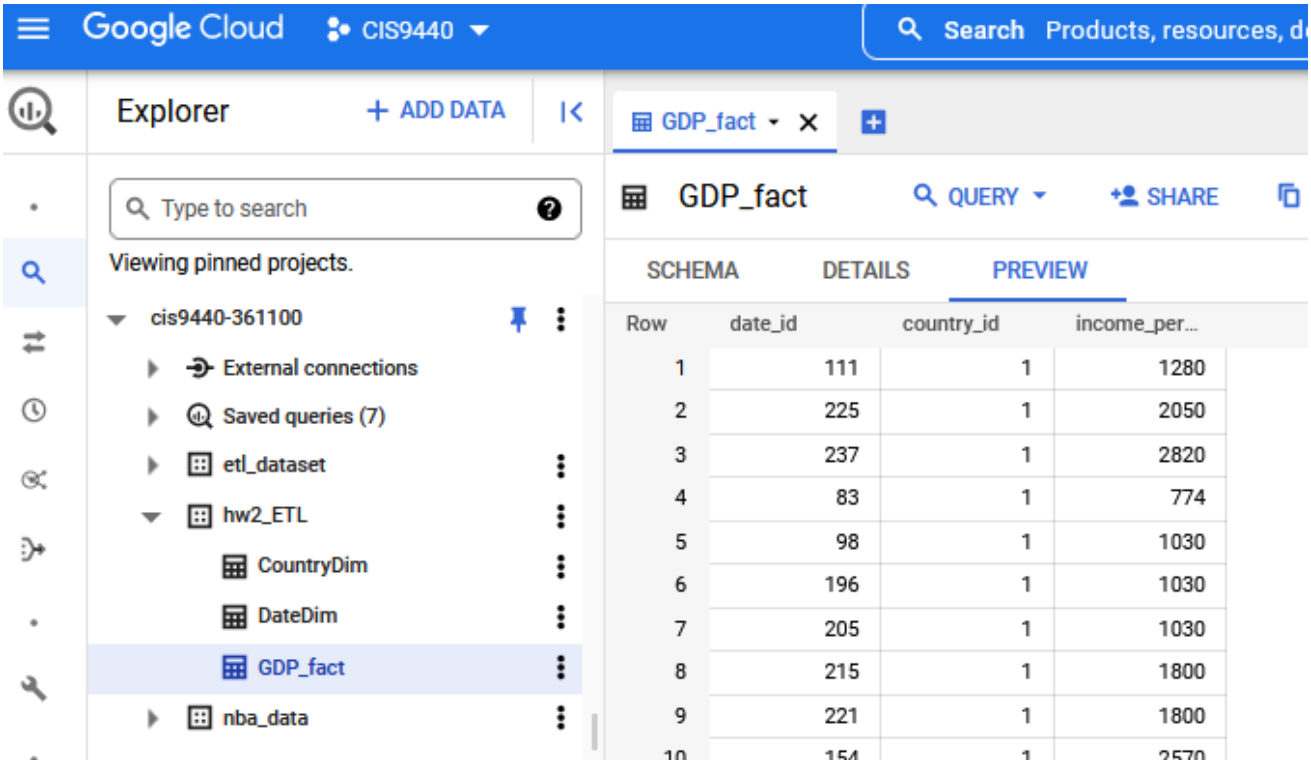

Google Cloud
CIS9440
Search Products, resources, docs (/)
Explorer
+ ADD DATA
DateDim
Type to search
Viewing pinned projects.
cis9440-361100
External connections
Saved queries (7)
etl_dataset
hw2_ETL
CountryDim
DateDim
GDP_fact
nba_data
DateDim
QUERY
SHARE
COPY
SCHEMA
DETAILS
PREVIEW
Filter Enter property name or value
Field name Type Mode Collation
date_id INTEGER NULLABLE
year INTEGER NULLABLE
decade INTEGER NULLABLE
EDIT SCHEMA
VIEW ROW ACCESS POLICIES


Google Cloud
CIS9440
Search Products, resources, doc
Explorer
+ ADD DATA
Type to search
Viewing pinned projects.
cis9440-361100
External connections
Saved queries (7)
etl_dataset
hw2_ETL
CountryDim
DateDim
GDP_fact
nba_data
DateDim
DateDim
QUERY
SHARE
SCHEMA
DETAILS
PREVIEW
Table info
Table ID
cis9440-361100.hw2_ETL.DateDim
Created
Oct 21, 2022, 9:29:10 PM UTC-4
Last modified
Oct 21, 2022, 10:13:57 PM UTC-4
Table expiration
NEVER
Data location
US
Default collation
Description
Storage info
Number of rows
241
Total logical bytes
5.65 KB
Active logical bytes
5.65 KB
Long term logical bytes
0 B
Total physical bytes
5.91 KB
Active physical bytes
5.91 KB
Long term physical bytes
0 B
Time travel physical bytes
3.94 KB
PERSONAL HISTORY
PROJECT HISTORY


Google Cloud
CIS9440
Search Products, resources, docs
Explorer
+ ADD DATA
Type to search
Viewing pinned projects.
cis9440-361100
External connections
Saved queries (7)
etl_dataset
hw2_ETL
CountryDim
DateDim
GDP_fact
nba_data
DateDim
DateDim
QUERY
SHARE
COP
SCHEMA
DETAILS
PREVIEW
Row
date_id
year
decade
1 1 1800 1800
2 2 1801 1800
3 3 1802 1800
4 4 1803 1800
5 5 1804 1800
6 6 1805 1800
7 7 1806 1800
8 8 1807 1800
9 9 1808 1800
10 10 1809 1800


Google Cloud
CIS9440
Search Products, resources, docs (/)
Explorer
+ ADD DATA
Type to search
Viewing pinned projects.
cis9440-361100
External connections
Saved queries (7)
etl_dataset
hw2_ETL
CountryDim
DateDim
GDP_fact
nba_data
GDP_fact
GDP_fact
QUERY
SHARE
COPY
SCHEMA
DETAILS
PREVIEW
Filter
Enter property name or value
Field name
Type
Mode
Collation
date_id INTEGER NULLABLE
country_id INTEGER NULLABLE
income_per_person INTEGER NULLABLE
EDIT SCHEMA
VIEW ROW ACCESS POLICIES


Google Cloud
CIS9440
Search Products, resources, docs
Explorer
ADD DATA
Type to search
Viewing pinned projects.
cis9440-361100
External connections
Saved queries (7)
etl_dataset
hw2_ETL
CountryDim
DateDim
GDP_fact
nba_data
GDP_fact
QUERY
SHARE
COPY
SCHEMA
DETAILS
PREVIEW
Table info
Table ID
cis9440-361100.hw2_ETL.GDP_fact
Created
Oct 21, 2022, 9:29:09 PM UTC-4
Last modified
Oct 21, 2022, 10:13:56 PM UTC-4
Table expiration
NEVER
Data location
US
Default collation
Description
Storage info
Number of rows
46,513
Total logical bytes
1.06 MB
Active logical bytes
1.06 MB
Long term logical bytes
0 B
Total physical bytes
357.51 KB
Active physical bytes
357.51 KB
Long term physical bytes
0 B
Time travel physical bytes
238.34 KB
PERSONAL HISTORY
PROJECT HISTORY



References

- "Pandas Melt, Stack and wide_to_long For Reshaping Columns into Rows"
  Credits: Susan Maina | Date written: Jul 17th, 2021
  https://towardsdatascience.com/wide-to-long-data-how-and-when-to-use-pandas-melt-stack-and-wide-to-long-7c1e0f462a98

- Week 7 Lecture on ETL using sodapy and Google BigQuery
  Credits: Michael O'Donnell | Date published: Oct 19th, 2022
  https://github.com/xyjiang970/cis9440-dataWarehousing/blob/main/lecture7/sodapy_ETL_20221018_class.ipynb