Big Data in the Cloud

This exercise looks and how you can use Google bigquery to interrogate large datasets quickly.

Preparing data

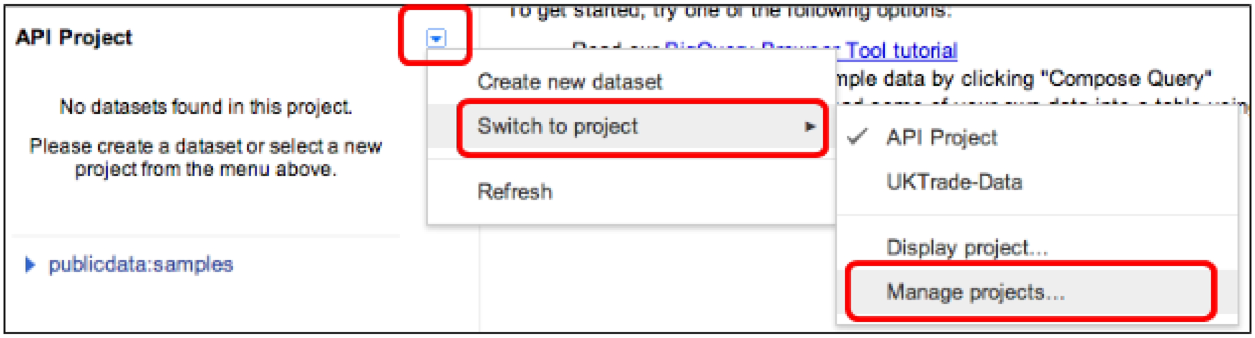
Before you can load data into big query it is advisable to normalise the data into one big flat table. This way the query will scale to billions of records. In this exercise this operation has already been performed.

Step 1 - Sign up

This exercise requires a bigquery account. To set one up go to https://bigquery.cloud.google.com and follow the instructions to set up billing on your first project. This exercise does not involve scaling to sizes of dataset that are billed however if you wish to ensure you won't be billed please ask the trainer at the end of the session ensure your account is safe.

Step 2 - Creating a project

Create a new project with a new dataset.



API Project

No datasets found in this project.

Please create a dataset or select a new project from the menu above.

Create a new project and then make sure that you click the "Billing and Settings" in the project management screen and enable billing. You will need this in order to upload data.

If you wish to load your own data files over 10Mb into big query then you first have to upload these into project cloud storage. This option is available from the project management screen (see right hand panel on next page).

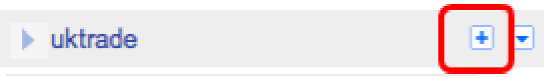
In this exercise, we shall load data I have already uploaded and publically shared, thus saving this step.

Step 3 - Create a dataset

Once done, return to https://bigquery.cloud.google.com and switch to the project you have just created (see the screenshot in step 2)

Now we need to create a dataset (again see the screenshot in step 2) and call this **uktrade**.

Once you have a dataset created we need to add a table, in this case called **exports**. To do this press the plus button next to the dataset and follow the steps below.



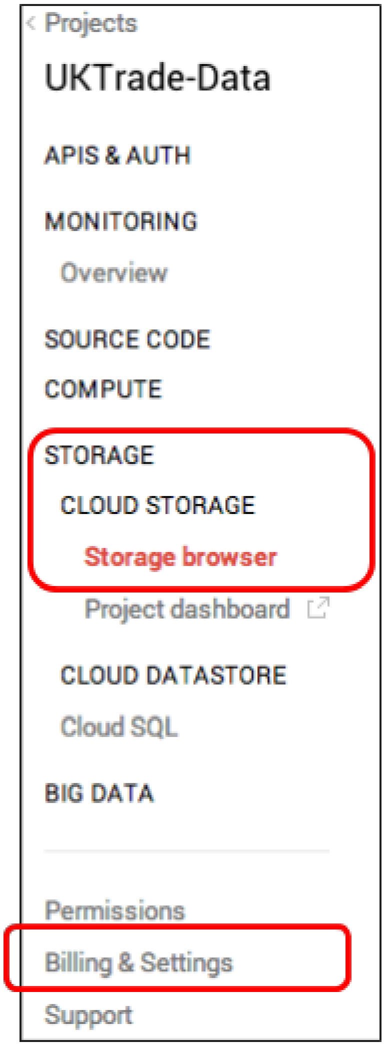
Choose Destination:

Dataset ID: **uktrade**
Table ID: **exports**

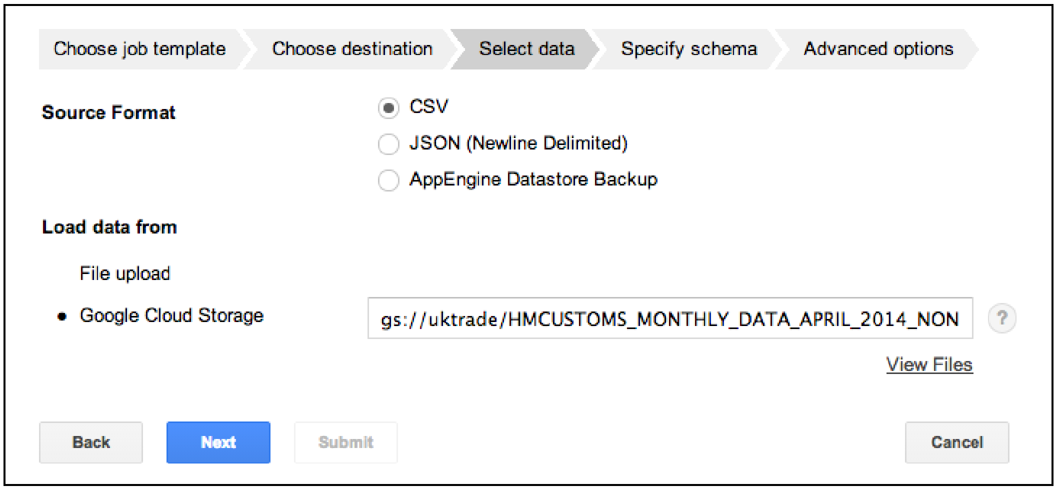
Select Data

Ensure you select **CSV**.

Load data from **Google Cloud Storage** using the following location:



```
gs://uktrade/HMCUSTOMS_MONTHLY_DATA_APRIL_2014_NON-EU_EXPORTS.csv
```



Source Format

● CSV
○ JSON (Newline Delimited)
○ AppEngine Datastore Backup

Load data from

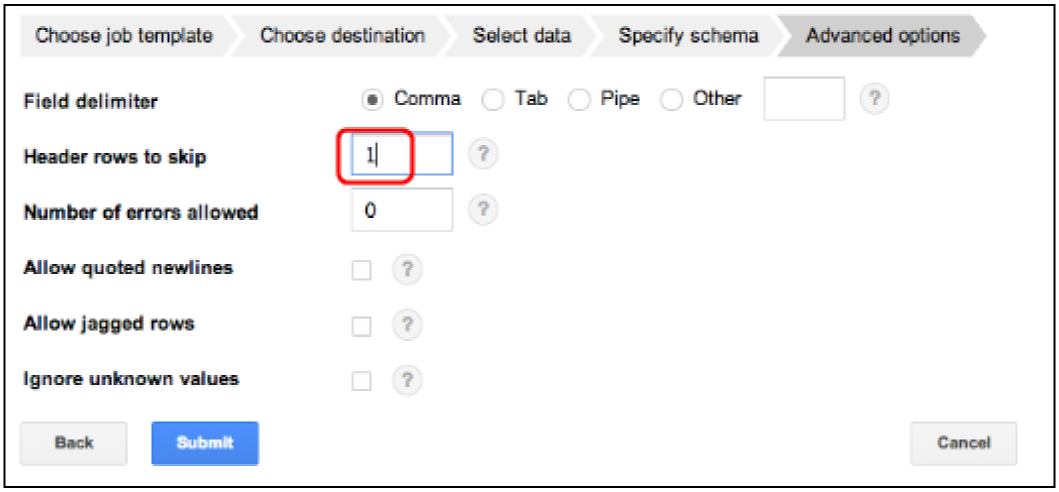
File upload
- Google Cloud Storage: gs://uktrade/HMCUSTOMS_MONTHLY_DATA_APRIL_2014_NON

Schema

Copy and paste the following into the schema box:

```
MAF_COMCODE_SECTION:STRING,
MAF_COMCODE_SUBSECTION:STRING,
MAF_COMCODE_CHAPTER:STRING,
MAF_COMCODE_TITLE:STRING,
MAF_COMCODE_DESCRIPTION:STRING,
MAF_COMCODE:INTEGER,
MAF_SITC:INTEGER,
MAF_RECORD_TYPE:INTEGER,
MAF_COD_SEQUENCE:INTEGER,
MAF_COD_ALPHA:STRING,
MAF_DATE:TIMESTAMP,
MAF_PORT_SEQUENCE:INTEGER,
MAF_PORT_ALPHA:STRING,
MAF_FLAG_SEQUENCE:INTEGER,
MAF_FLAG_ALPHA:STRING,
MAF_TRADE_INDICATOR:INTEGER,
MAF_CONTAINER:INTEGER,
MAF_MODE_OF_TRANSPORT:INTEGER,
MAF_INLAND_MOT:INTEGER,
MAF_GOLO_SEQUENCE:INTEGER,
MAF_GOLO_ALPHA:STRING,
MAF_SUITE_INDICATOR:STRING,
MAF_PROCEDURE_CODE:INTEGER,
MAF_VALUE:INTEGER,
MAF_QUANTITY_1:INTEGER,
MAF_QUANTITY_2:INTEGER,
MAF_INDUSTRIAL_PLANT_COMCODE:INTEGER
```

Advanced Options



Field delimiter ● Comma ○ Tab ○ Pipe ○ Other

Header rows to skip: 1

Number of errors allowed: 0

Allow quoted newlines □

Allow jagged rows □

Ignore unknown values □

Ensure that you skip the first header row of the data being imported as this contains the column headers.

Finally click submit and be patient while the table loads.

Step 4 - Querying the data

With the data loaded, clicking on the table will allow you to view the schema, a preview of the data and also access the query interface.

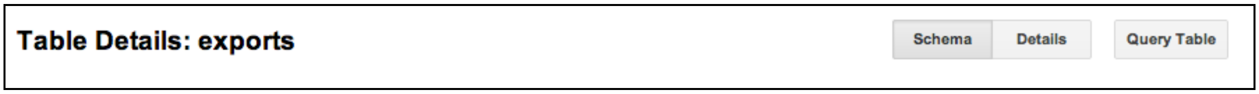


Table Details: exports

With the data loaded, we now have an enterprise level database in the cloud, optimised for fast query over very large datasets. So far we have only loaded one file but we could equally load millions of rows and terrabytes of data into the bigquery engine.

Lets run some queries on the data:

Query 1: The input file was for one month of exports, was it?

```
SELECT MAF_DATE,count(MAF_COMCODE) FROM [uktrade.exports] GROUP BY MAF_DATE
```

Query 2: How many exports per COMCODE subsection?

```
SELECT MAF_COMCODE_SUBSECTION,count(MAF_COMCODE) as number FROM
[uktrade.exports] GROUP BY MAF_COMCODE_SUBSECTION order by number desc
```

Query 3: What was the most valuable export (by COMCODE subsection)?

```
SELECT MAF_COMCODE_SUBSECTION,sum(MAF_VALUE) as value FROM [uktrade.exports]
GROUP BY MAF_COMCODE_SUBSECTION order by value desc;
```

Query 4: I’m concerned about this classification of Nuclear reactors, what exactly are we exporting. Is this an error somewhere in the data or really true?

```
SELECT MAF_COMCODE_DESCRIPTION,sum(MAF_VALUE) as value FROM [uktrade.exports]
where MAF_COMCODE_SUBSECTION="Nuclear reactors" GROUP BY
MAF_COMCODE_DESCRIPTION order by value desc;
```

Step 5 - Moving to the command line (Advanced)

More on using the web interface to Google Bigquery can be found in this guide:

https://developers.google.com/bigquery/bigquery-browser-tool

Unfortunately to load further datasets into Bigquery requires using the command line toolkit which works on all platforms (Windows, Mac and Linux).

For detailed instructions on installing the toolkit, visit the following URL:

https://developers.google.com/cloud/sdk

Once you have the toolkit installed, from you command line ensure you first login

```
gcloud auth login
```

Once logged in we need to ensure that the toolkit is manipulating the correct project. To find out which project you are using go to the following URL

https://console.developers.google.com/project

On the screen you will see your projects and their associated **IDs**. Copy the ID of your project and then issue the following command on the command line:

```
gcloud config set project project_id
```

You can now open a bigquery interface on your project with the command:

```
bq shell
```

Having successfully executed these command you should now have a project shell that that starts with your project id.

To show the datasets in your project try the following command

```
ls
```

Use the dataset name, you can then show the tables in your dataset:

```
ls dataset_id
```

As per the web interface we can also show the schema of a table:

```
show dataset_id.table_id
```

and also view the top 10 records:

```
head -n 10 dataset_id.table_id
```

In order to query the data simply issue the `query` command and enclose the query in quote marks, e.g.

```
query ‘SELECT MAF_DATE,count(MAF_COMCODE) FROM [uktrade.exports] GROUP BY
MAF_DATE’
```

You can press the up and down arrows to navigate between previous commands you have issued to save some typing. You can also type help with any command to get additional information. e.g. `help`, `help query`, `help load`.

Step 6 - Adding more data to your table (Advanced)

The following command appends another dataset to your existing table. As before this dataset is pre-shared ready for use in this exercise. You could equally upload your own to google storage as per the guide in step

```
load --skip_leading_rows=1 uktrade.exports
gs://uktrade/HMCUSTOMS_MONTHLY_DATA_FEBRUARY_2014_NON-EU_EXPORTS.csv
```

As before, you will need to be patient while the data loads.

Step 7 - Self exploration (Advanced)

There is a guide on using the command line tool for big query at the following location:

https://developers.google.com/bigquery/bq-command-line-tool-quickstart

If you wish to load more datasets into your table then datasets are available from the gs://uktrade storage dating all the way back to 2007.