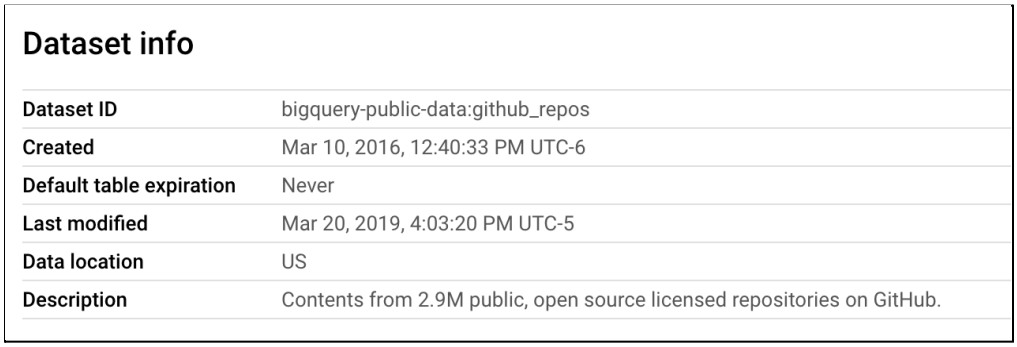
Guide to download Github data from Google BigQuery

We detail the steps to download data from the BigQuery public dataset - **github_repos**.



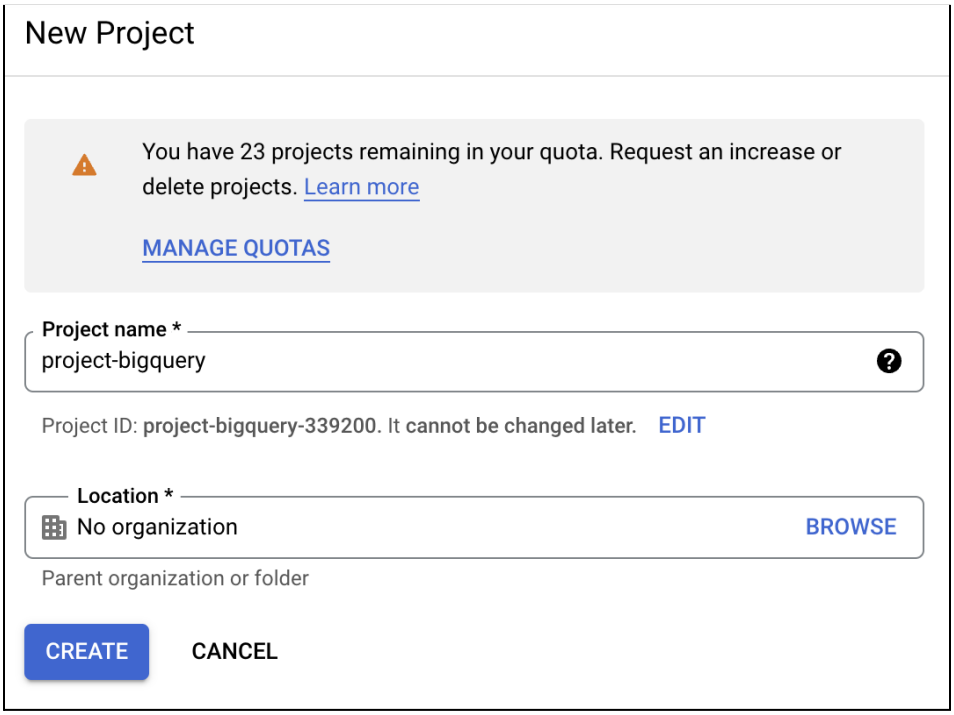
Dataset info

Dataset ID	bigquery-public-data:github_repos
Created	Mar 10, 2016, 12:40:33 PM UTC-6
Default table expiration	Never
Last modified	Mar 20, 2019, 4:03:20 PM UTC-5
Data location	US
Description	Contents from 2.9M public, open source licensed repositories on GitHub.

We discuss the steps as follows.

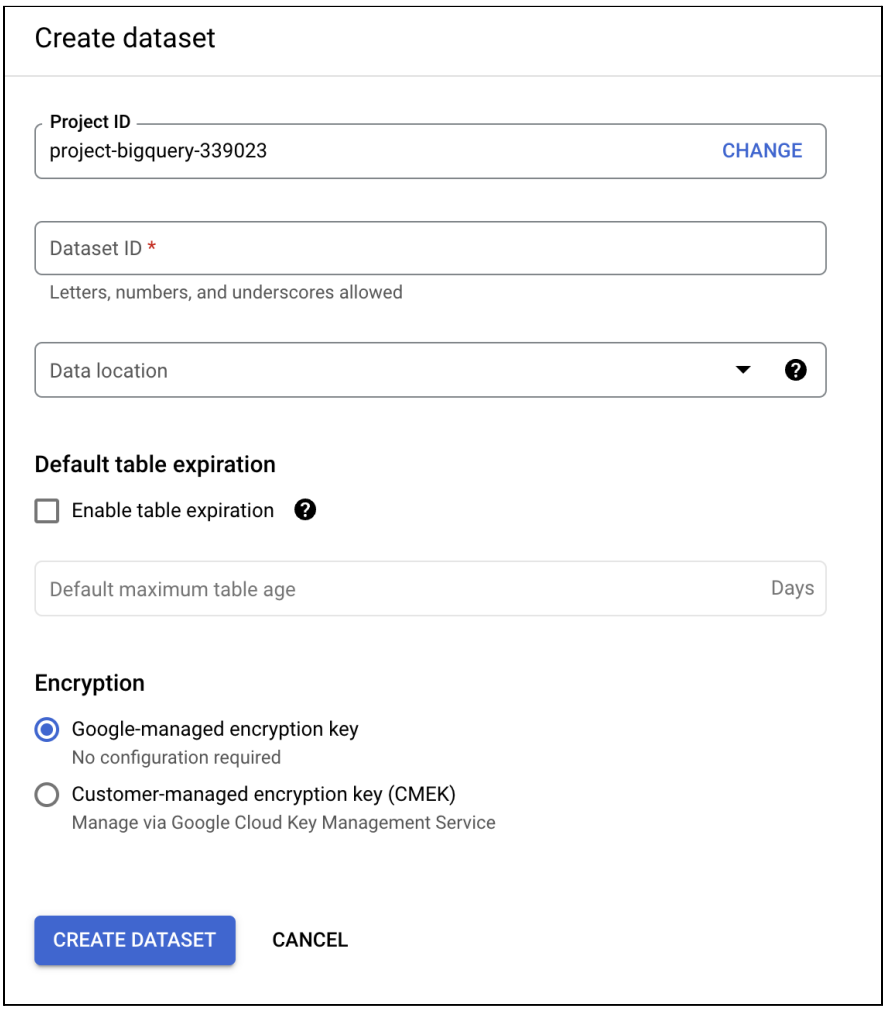
Step1. Create a Google platform account (you will be given $300 free credit that is sufficient to download the Github data).

Step2. Create a Google Big Query project here. The new project creation interface looks like below. Select a project name (e.g., “project-bigquery”) and leave the location field blank.

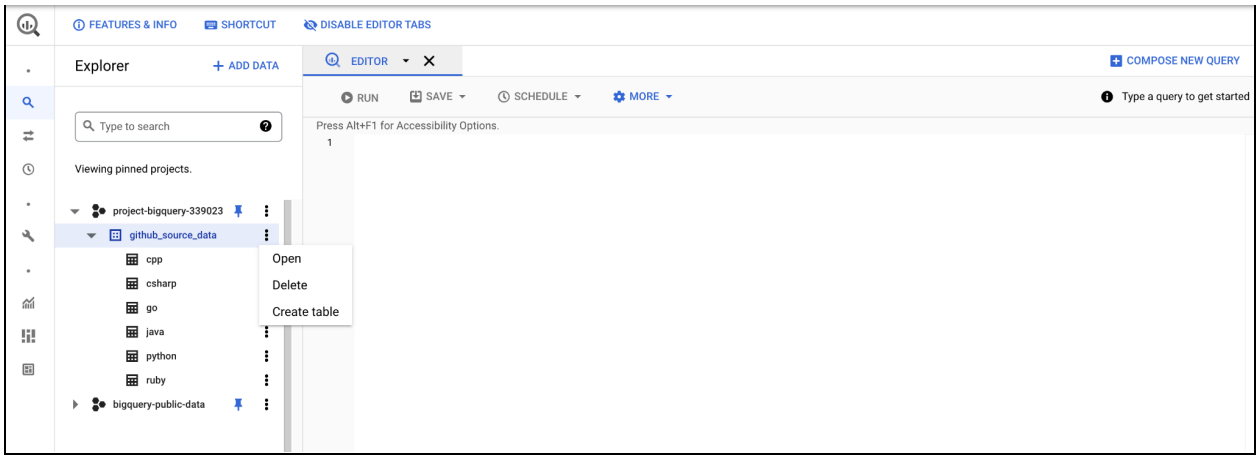


Step3. In this project, create a dataset as shown in the following figure.

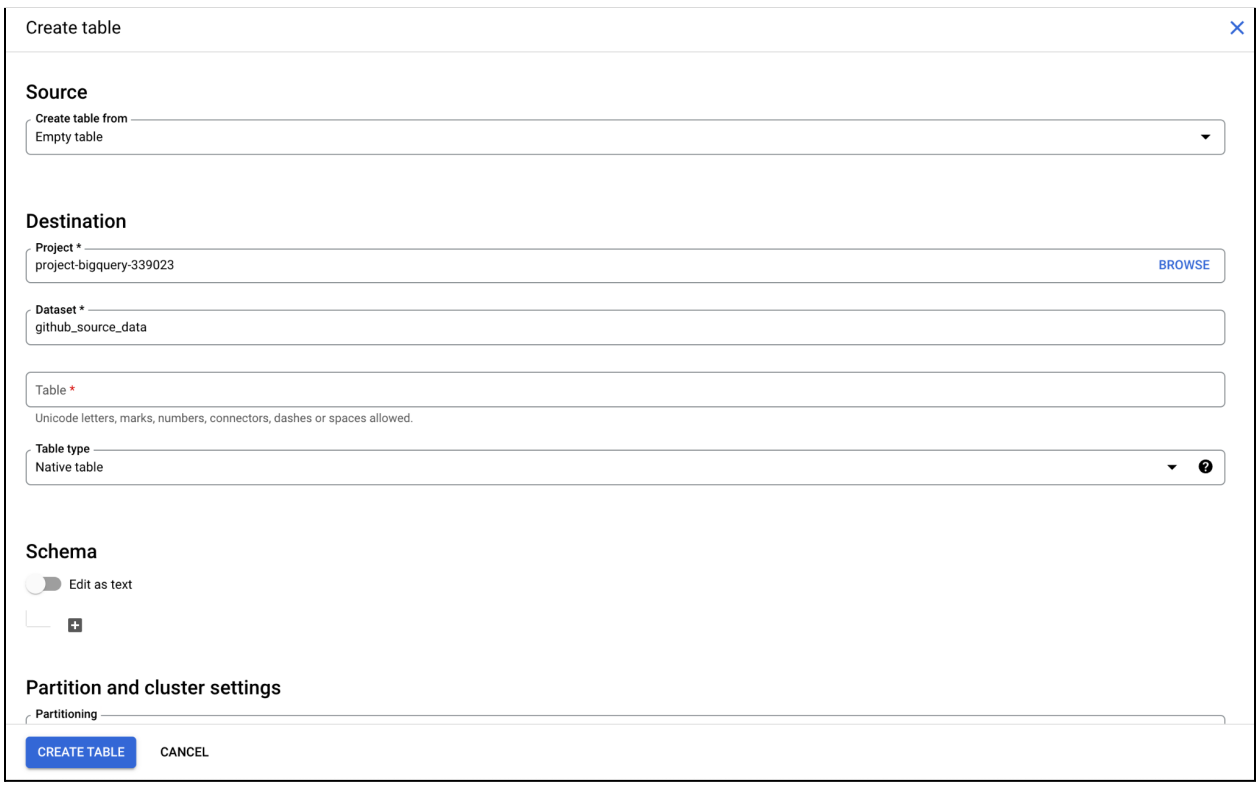
In the “Create dataset” wizard, set the **Dataset ID** as “github_source_data” and leave the **Data location** field empty.



Step4. In this dataset, create one table per programming language. The results of each SQL request (one per language) will be stored in these tables. (ex., we create 6 tables)

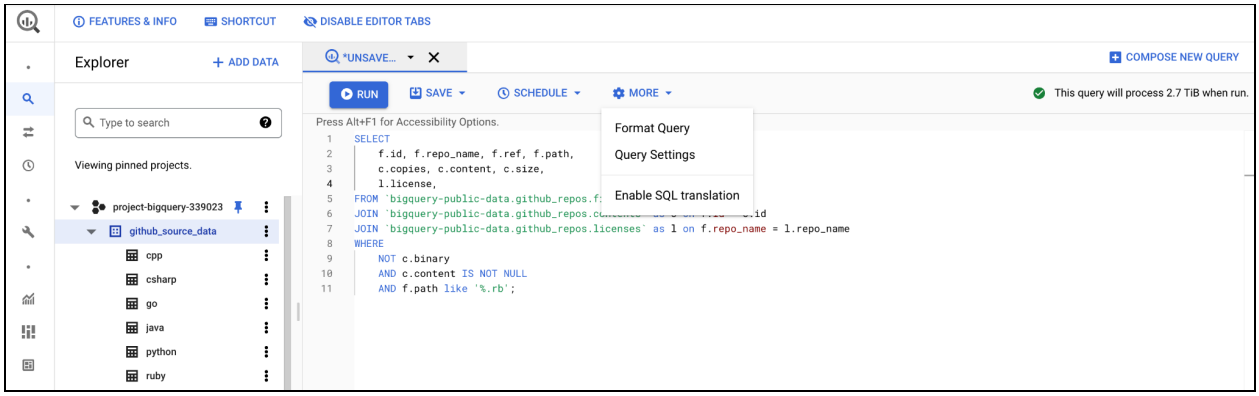


In the “Create table” wizard, set the **Table** name (e.g., “ruby”) and leave other fields to their default value. Make sure the project and dataset name is set accordingly.

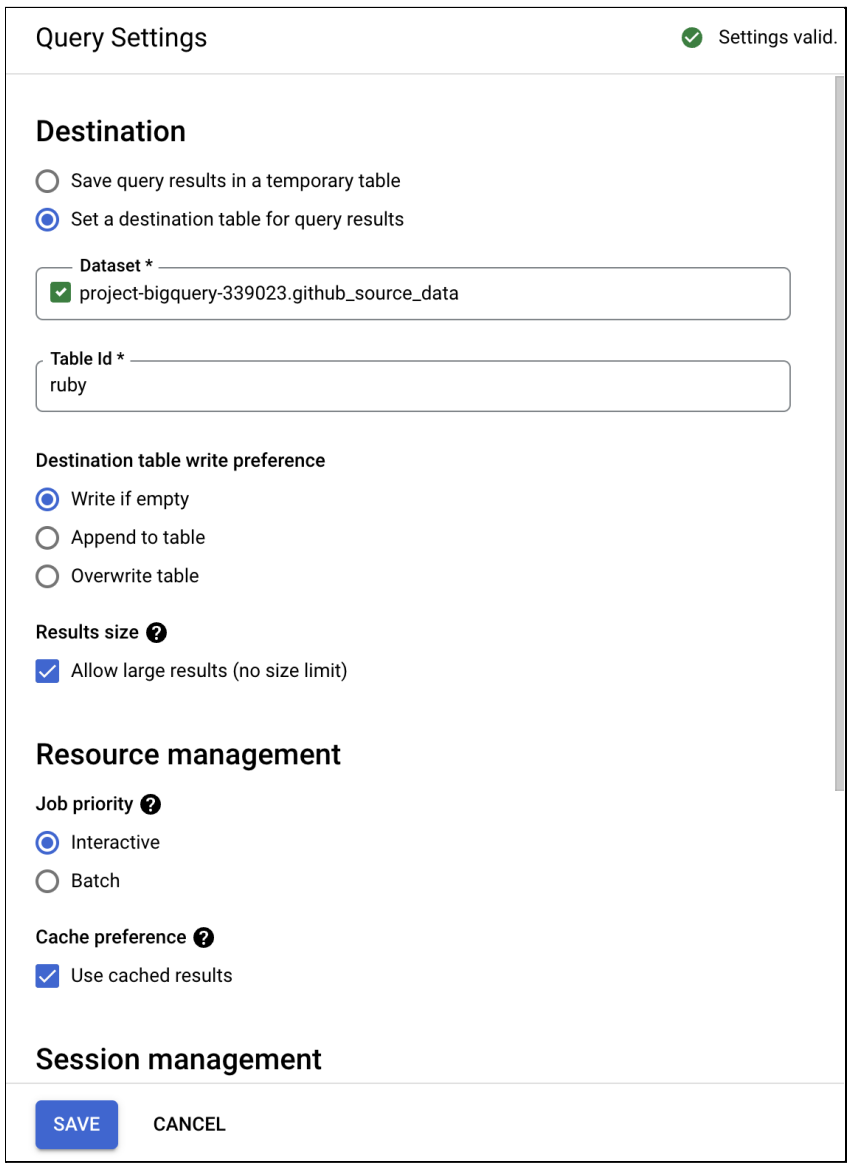


Step5. Before running an SQL request, make sure you change the query settings to save the query results in the dedicated table (more → Query Settings → Destination → table for query results → put table name). See the following figures.

Click on the “Query Settings” choice from the drop-down menu.



In the “Query Settings” wizard, set the **fields** (Destination, Dataset, Table Id, Destination table write preference, Results size) as shown in the following figure.



Step6. Run the SQL request (one per language and don't forget to change the table for each request). We show the SQL query for the **Ruby** language as follows.

```
SELECT
    f.id, f.repo_name, f.ref, f.path,
    c.copies, c.content, c.size,
    l.license,
FROM `bigquery-public-data.github_repos.files` as f
JOIN `bigquery-public-data.github_repos.contents` as c on f.id = c.id
JOIN `bigquery-public-data.github_repos.licenses` as l on f.repo_name =
l.repo_name
WHERE
    NOT c.binary AND c.content IS NOT NULL AND f.path like '%.rb
```

We can consider removing duplicate content from the tables (we save the de-duplicated data in another table; you may consider updating the existing table).

```
SELECT
      id, repo_name, ref, path,
      copies, content, size, license,
FROM (
    SELECT
        *,
        ROW_NUMBER()
            OVER (PARTITION BY content)
            row_number
    FROM dataset_name.table_name
)
WHERE row_number = 1
```

For example, we can replace `dataset_name.table_name` with `github_source_data.ruby` to remove duplicate content for the Ruby language.

For the following nine languages, we can use the corresponding file extensions.

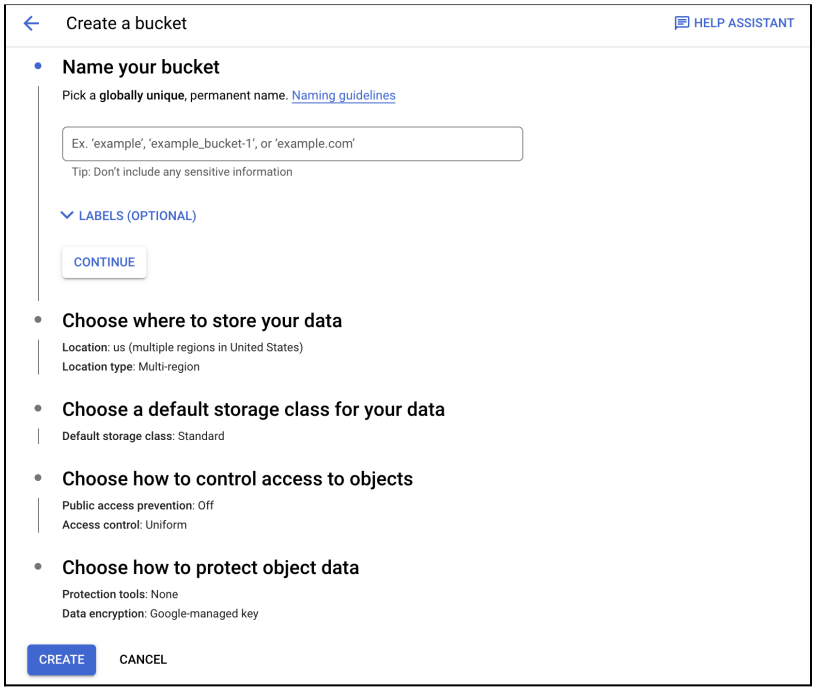
```
C = .c; C++ = .cpp; C# = .cs; Go = .go; Java = .java;
Javascript = .js; PHP = .php; Python = .py; Ruby = .rb;
```

Please take note of the following points.

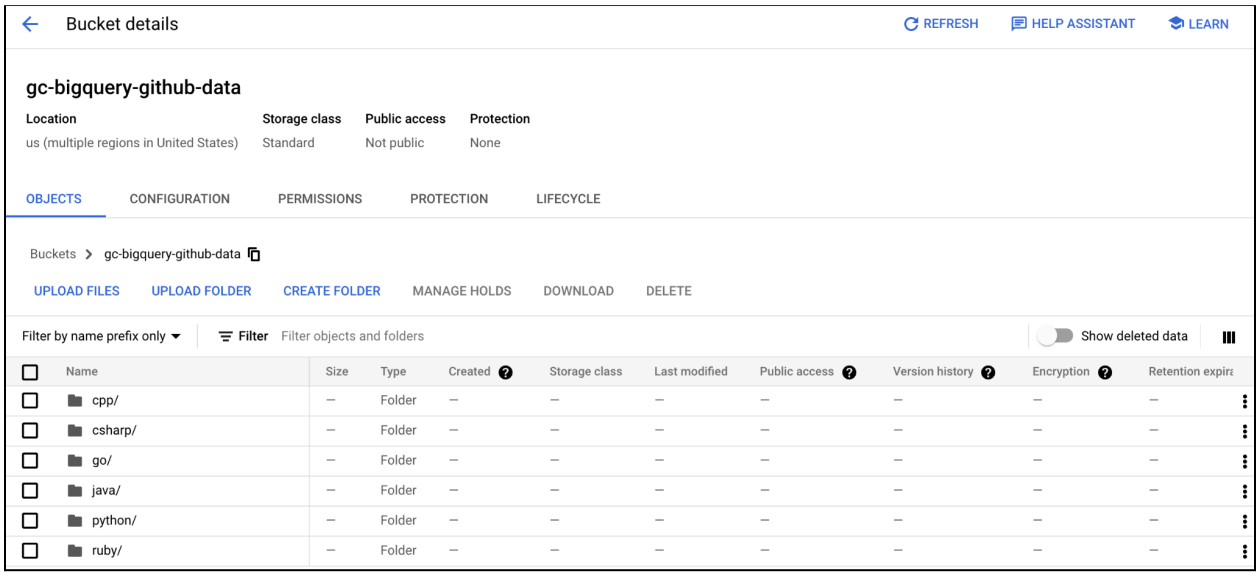
- Check BigQuery pricing from here.
- When we run an SQL request to fetch language-specific data from Github, it processes 2.7 TiB of data, therefore, one request would cost (2.7 x 5) = $13.5 based on the current pricing list in the above-mentioned link.
- It may take a few minutes to run the SQL request and fetch the data into the target table.

Step7. Export your results to Google Cloud Storage (GCS).

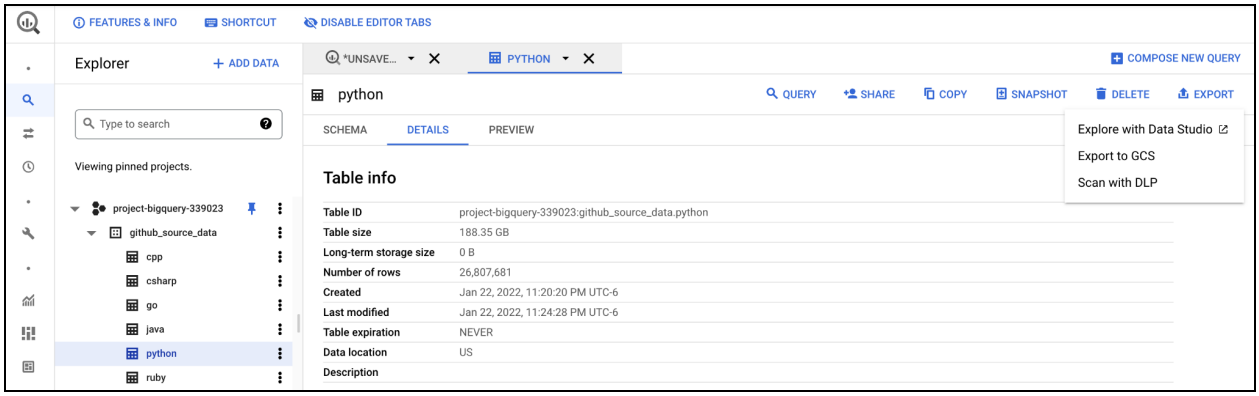
In GCS, create a bucket. In the wizard, chose a bucket name and leave the other fields to their default value (or empty) as shown in the following figure.



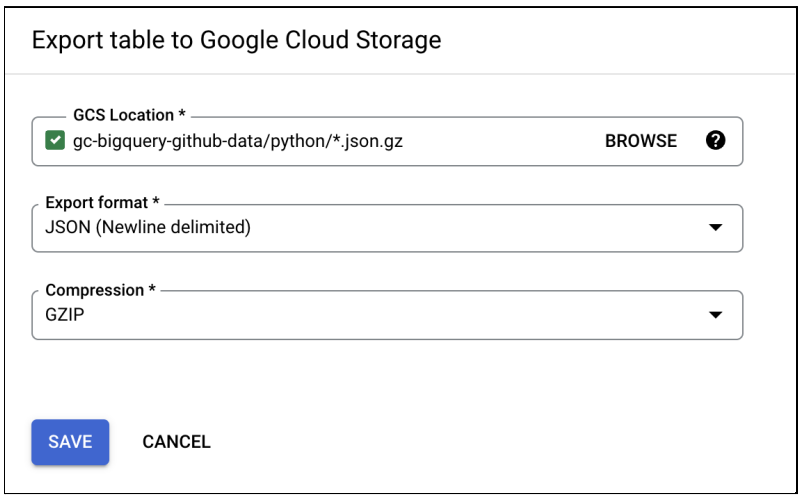
Then create a folder for each language in the bucket. For example, we create six folders in the bucket named `gc-bigquery-github-data` as shown in the following Figure.



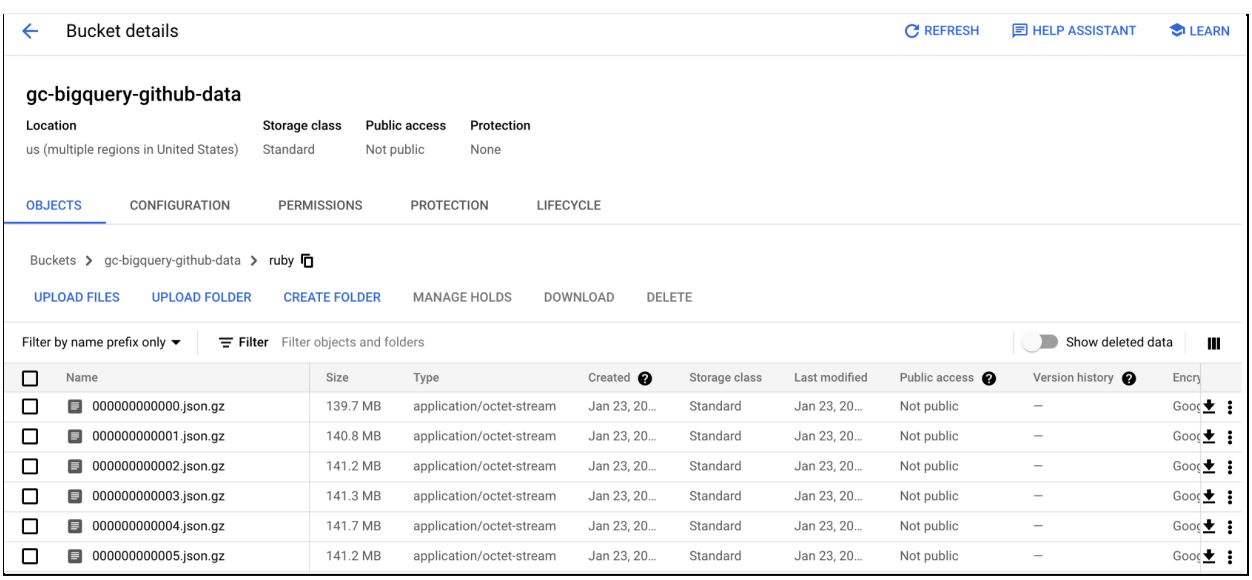
Next, export the dataset tables into this bucket. Do `EXPORT -> Export to GCS` as shown in the following figure.



Then, as shown in the following figure, choose the field values accordingly.



Note that, the “GCS Location” should be “name_of_bucket/name_of_folder/*.json.gz”. The files will be compressed (with the “.json.gz” extension as shown in the following Figure).



Step8. To download the bucket on your machine, use the gsutil tool.

First, install the tool.

```
pip install gsutil
```

Then configure `gsutil` by running `gsutil config` (follow steps as the terminal says). We are ready now to copy the bucket on the local machine.

```
gsutil -m cp -r gs://name_of_bucket/name_of_folder dest_folder
```

For example, we can copy Ruby language data from the bucket by running:

```
gsutil -m cp -r gs://gc-bigquery-github-data/ruby dest_folder
```

To learn about the total size of the data for a particular language, run:

```
gsutil du -sh -a gs://gc-bigquery-github-data/ruby
```

Sample JSON file content

Each line in JSON files are JSON object as follows.


```
"id": "c887ac2a8f597b5ebd7bc746d843f9a7bd05a9b3",
"repo_name": "maisaengineering/dv-seedstarter",
"ref": "refs/heads/master",
"path": "db/migrate/20101227195636_create_oauth_providers.rb",
"copies": "22",
"content": "require 'sexy_pg_constraints'\nclass CreateOauthProviders <
ActiveRecord::Migration\n  def self.up\n    create_table :oauth_providers do
|t|\n      t.text :name, :null => false\n      t.text :key, :null => false\n
t.text :secret, :null => false\n      t.text :scope\n      t.integer :order\n
t.timestamps\n    end\n    constrain :oauth_providers do |t|\n      t.name
:not_blank => true, :unique => true\n      t.key :not_blank => true\n
t.secret :not_blank => true\n    end\n  end\n\n  def self.down\n    drop_table
:oauth_providers\n  end\nend\n\n",
"size": "540",
"license": "mit"
```

Language-wise Github dataset size available in Google BigQuery

We detail the dataset sizes in the following table for nine languages.

Language	Full Data		Deduplicated Data	
	Size(GB)	#Files	Size(GB)	#Files
C	1100	265,849,088	30.3	5,888,554
C++	42.7	16,200,761	14.4	4,999,534
C#	20.7	18,363,942	7.4	6,970,868
Go	49.8	27,734,386	23.0	2,289,362
Java	66.3	58,523,793	23.7	19,958,621
Javascript	1100	283,835,669	48.4	13,706,946
PHP	133.1	91,442,122	17.3	11,932,791
Python	45.5	26,803,167	12.9	7,355,273
Ruby	13.1	20,246,012	3.1	4,558,544