Google Dremel

Vicente Orjales - vorjales@gmail.com
http://www.linkedin.com/in/vorjales

Index

Introduction.What is Dremel?

Google Dremel is a system developed by Google for interactively querying large data sets.

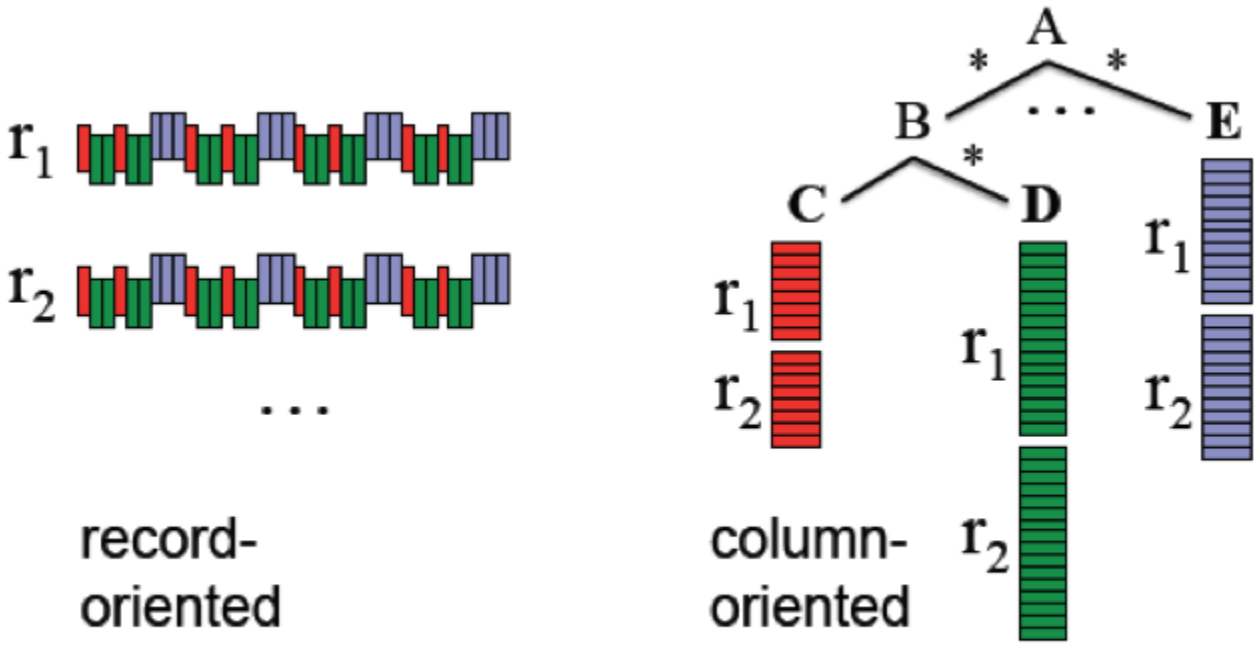
- **Near real time interactive analysis** (instead batch processing). SQL-like query language.
- Nested data (most of unstructured data supported) with a column storage representation.
- Tree architecture (as in web search): multi-level execution trees for query processing.

It is widely used inside Google and available for the public as Google BigQuery. There is also an open source implementation named Apache Drill (formerly Open Dremel) still in beta phase.

Google and Big Data. History.

- **2003**: **Google File System**. Scalable distributed file system for data intensive applications built over cheap commodity hardware.
- **2004**: **Map-Reduce**. Programming model for large data sets. Automatically parallelizes jobs in large cluster of commodity machines. Inspiration for Hadoop.
- **2006**: **Bigtable**. Distributed storage system for structured data. Inspiration for NoSQL databases
- **2010**: **Percolator**. Adds transactions and locks at table and row level.
- **2010**: **Pregel**. Scalable Graph Computing. Mining data from social graphs.
- **2010**: **Dremel**. Ner real time solution with SQL-like language. Model for Google BigQuery.

Columnar Representation of Nested Data



All values of a nested field such as A.B.C are stored contiguously. Therefore, A.B.C can be retrieved without reading A.E, A.B.D, etc.

Columnar storage proved has been successfully used for flat relational data, but making it work for the nested data within Dremel was one the challenges in the Dremel design: how to preserve all structural information and be able to reconstruct records from an arbitrary subset of fields.

Columnar Representation of Nested Data

```
message Document {
  required int64 DocId;
  optional group Links {
    repeated int64 Backward;
    repeated int64 Forward;
  }
  repeated group Name {
    repeated group Language {
      required string Code;
      optional string Country;
    }
    optional string Url;
  }
}
```

```
DocId: 10                    r1
Links
  Forward: 20
  Forward: 40
  Forward: 60
Name
  Language
    Code: 'en-us'
    Country: 'us'
  Language
    Code: 'en'
  Url: 'http://A'
Name
  Url: 'http://B'
Name
  Language
    Code: 'en-gb'
    Country: 'gb'
```

```
DocId: 20                    r2
Links
  Backward: 10
  Backward: 30
  Forward:  80
Name
  Url: 'http://C'
```

Columnar Representation of Nested Data

DocId

value	r	d
10	0	0
20	0	0

Name.Url

value	r	d
http://A	0	2
http://B	1	2
NULL	1	1
http://C	0	2

Links.Forward

value	r	d
20	0	
40	1	
60	1	
80	0	

Links.Backward

value	r	d

Name.Language.Code

value	r	d
en-us	0	2
en	2	2
NULL	1	1
en-gb	1	2
NULL	0	1

Name.Language.Co

value	r	d
us	0	3
NULL	2	2
NULL	1	1
gb	1	3
NULL	0	1

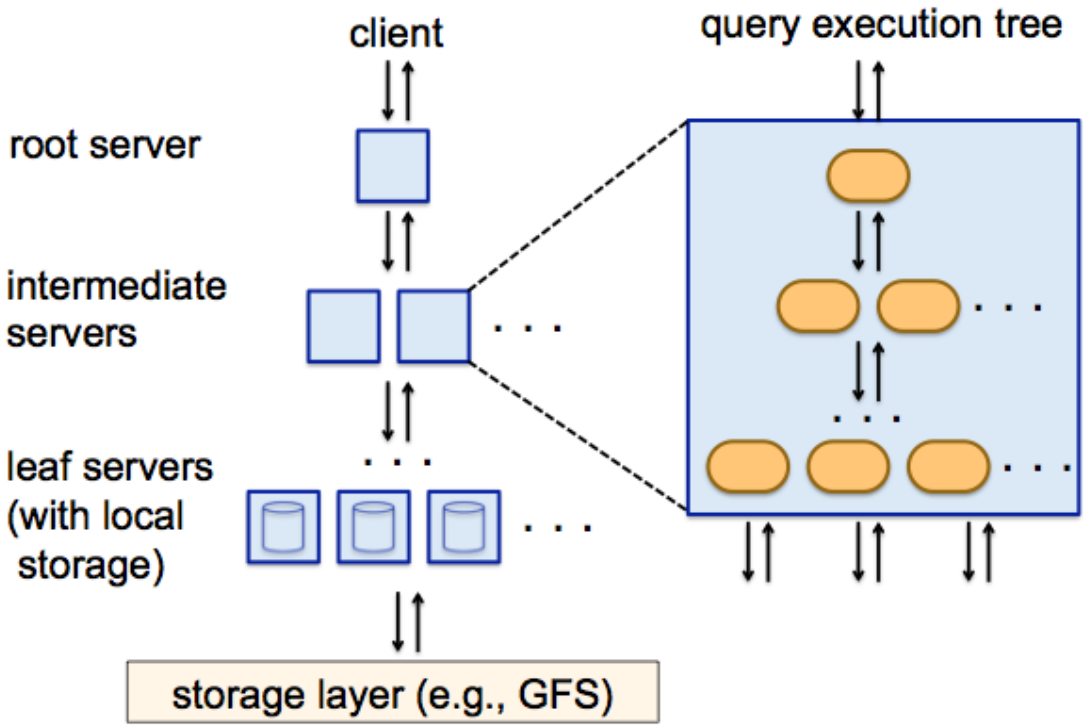
record (r=0) has repeated

Language (r=0) has repeated

```
DocId: 10                    r1
Links
  Forward: 20
  Forward: 40
  Forward: 60
Name
  Language
    Code: 'en-us'
    Country: 'us'
  Language
    Code: 'en'
  Url: 'http://A'
Name
  Url: 'http://B'
Name
  Language
    Code: 'en-gb'
    Country: 'gb'
```

evels

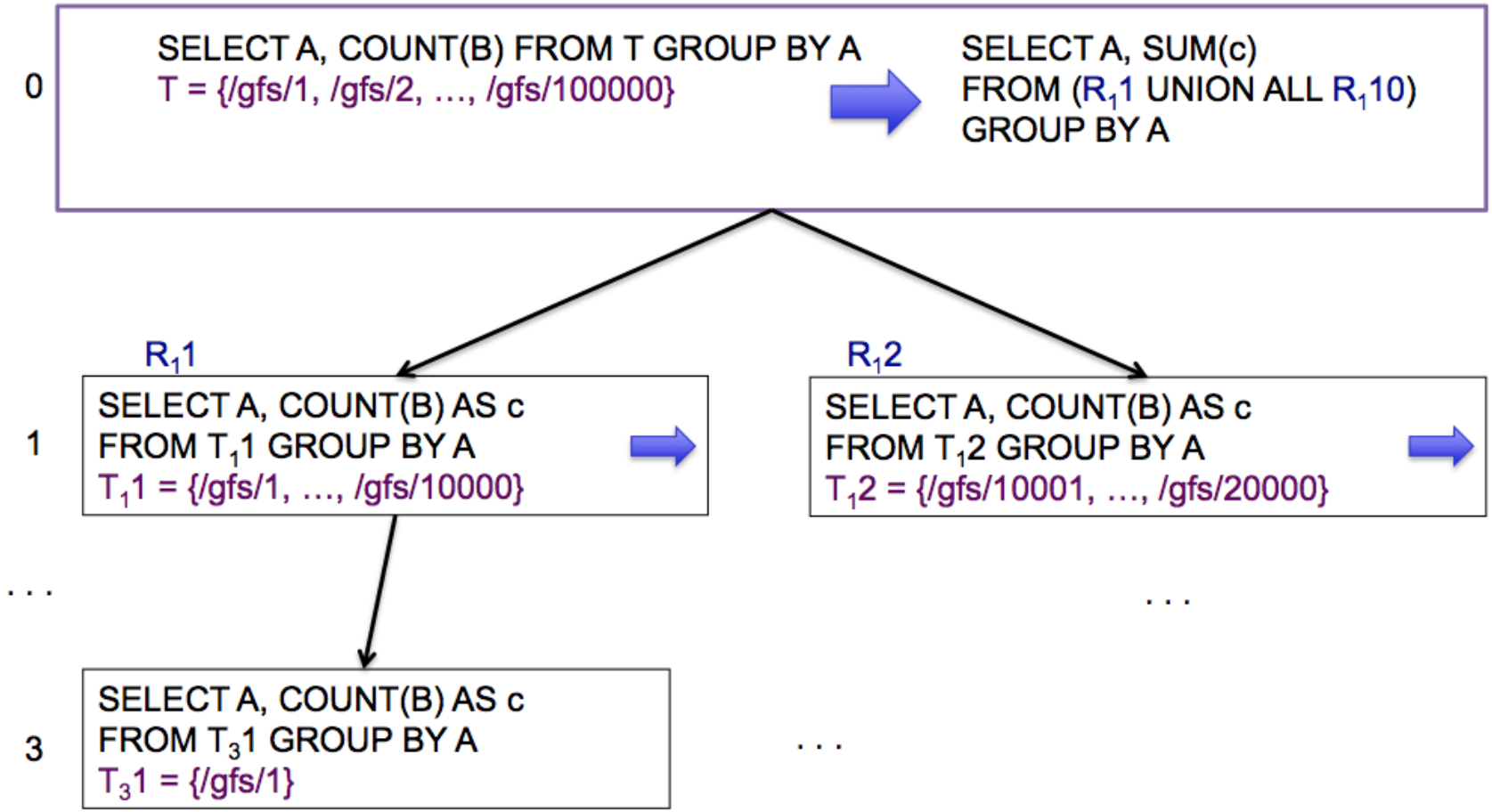
Query execution



Multi-level serving tree for executing queries:

- Root Server: receive incoming queries, read metadata from tables, route query to next level.
- Leaf Server: Communication and access to the data layer for retrieving the results.
- Each server has an internal execution tree that corresponds with a physical query execution plan.

Query execution - Example: count()



Google Dremel vs Map-Reduce.

M-R	Dremel
Batch Processing	Interactive query
Not query language. May be added with extra modules (e.g.: Hive in Hadoop)	SQL-like query language
Data organization depends on the source	Nested data. Column oriented data organization.
Hadoop implementation: Considerable admin effort	BigQuery implementation: Admin transparent to the user.

Google Dremel vs Map-Reduce.

- Emerged at different times and with different motivations: MR focused on distributed processing for large data sets, while Gremel driver was ad hoc. queries in near real time.

- Some authors argue that MR approach is not enough for the new big data requirements companies are confronting today (real time, complex joins, ACID...).

- While web companies has moved forward with their own products (e.g. Google with Dremel or Pregel), most of "conventional" entreprises has decided to buy instead build, selecting and investing in Hadoop based products in most of cases (Cloudera, Hortonworks, EMC).

The market tendency is to keep building Hadoop based solutions and enrich them with extra layers providing better real time support and scalability, while web companies keep innovating with new products. In this scenario Dremel and MR may be complementary instead of alternative approachs.

Dremel Implementations

- Google BigQuery
  - It is the externalization of the Dremel product, making it available to the public as part of the Google Cloud.
  - Data upload: directly to BigQuery or through Google Cloud Storage (better performance for big large data sets).
  - Data manipulation: web interface / Language API / REST API

- Apache Drill (formerly Open Dremel)
  - Open Source implementation of Google Dremel.
  - Apache License (Apache Incubator). http://incubator.apache.org/drill/
  - Still Beta (Alpha?) Phase

BigQuery - Case studies

- Top advertiser network in Brazil.
- Challenge: process millions of records generated daily (impressions, click-through, views) to be sure system is targeting ads effectively.
- Solution: BigQuery to gain real time insights into more than 3 billion ads in 350K blogs and webs.

- Travel agency with bus ticketing in India.
- Challenge: Analyze booking and inventory from systems of hundreds of bus operators serving more than 10,000 routes. Hadoop takes much time and requires specialized staff.
- Solution: BigQuery allows to quickly detect improvement opportunities (e.g.: routes with available seats)

- its customer has 16 bungalow villages in Europe.
- Challenge: Forecast number of vacationers and determine optimal pricing for accommodations. Limitations of current systems. Several architectures has failed (MS, SAS, BO, Excel).
- Solution: Their own application based in BigQuery performs analysis in seconds instead hours and without extra people.

Source: https://developers.google.com/bigquery/case_studies/

BigQuery - Demo - Conclusions

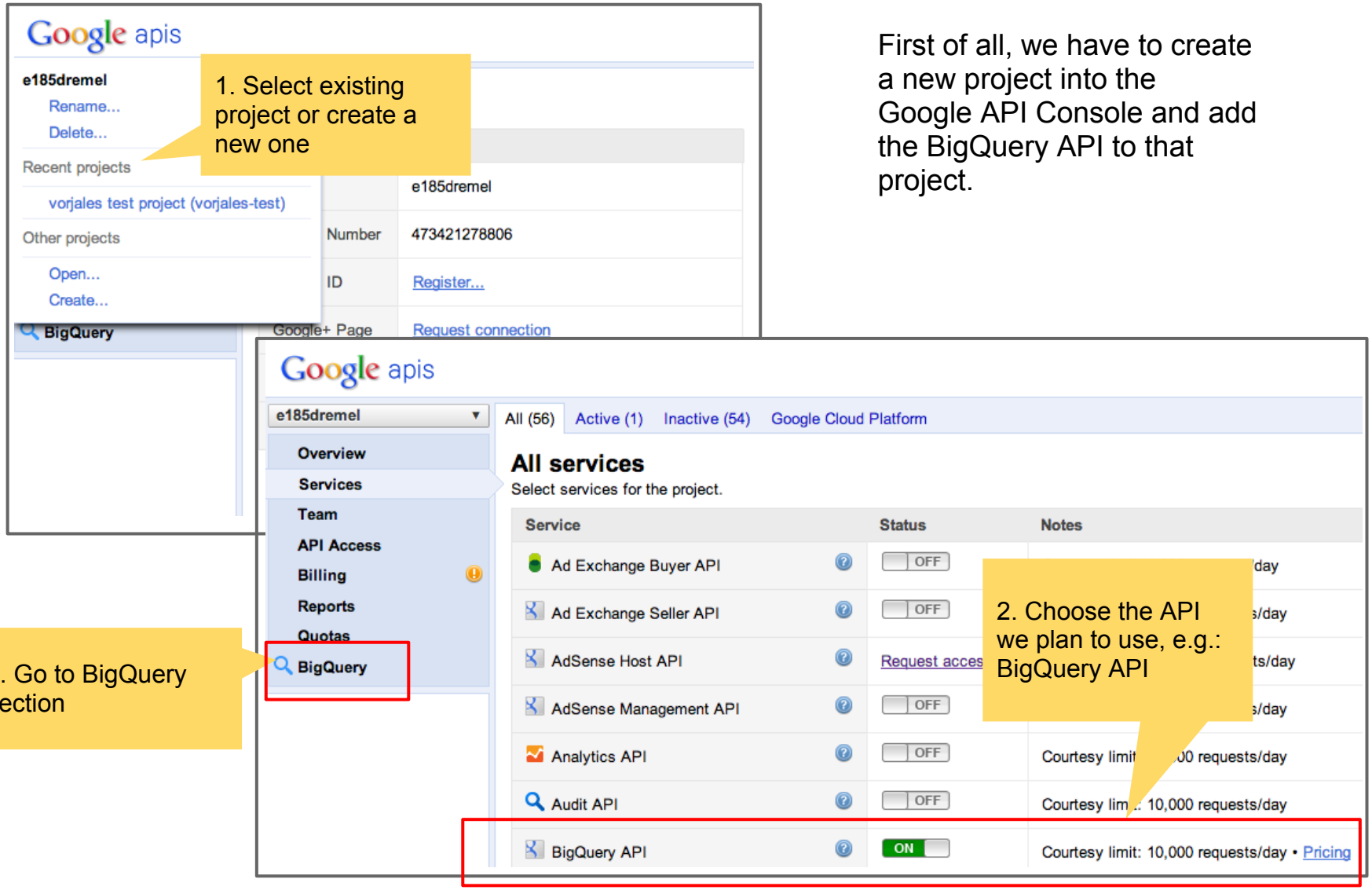
- No any installation or configuration of infrastructure is needed. Everything is in the Google Cloud.
- Interactive system.
- The amount of processed data depends on the selected columns (column oriented storage)
- ...and the billing depends on the amount of data we've processed!!

Appendix

Demo Screens

BigQuery - Demo - Project Creation

https://code.google.com/apis/console/

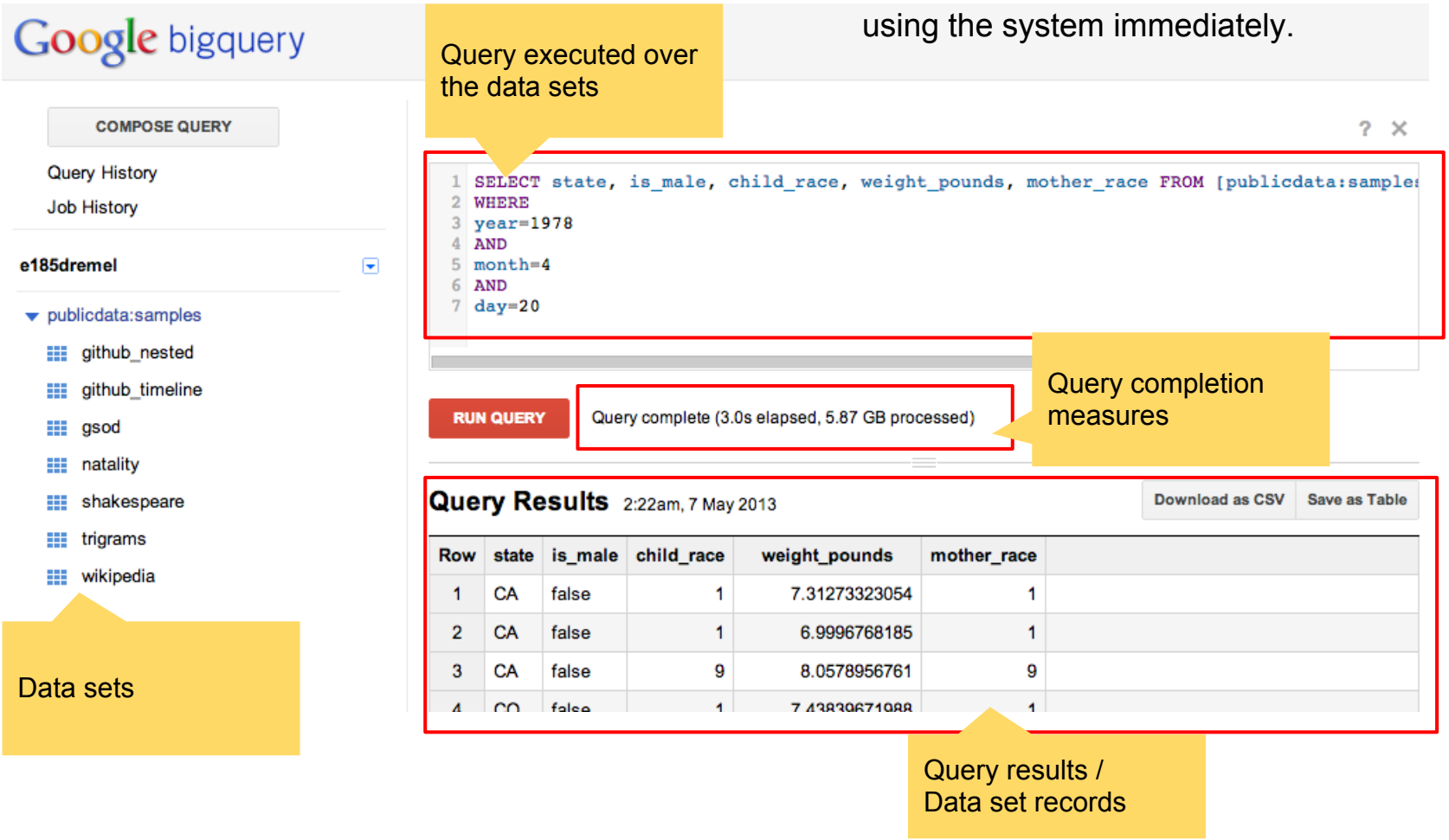


First of all, we have to create a new project into the Google API Console and add the BigQuery API to that project.

BigQuery - Demo - Web Console

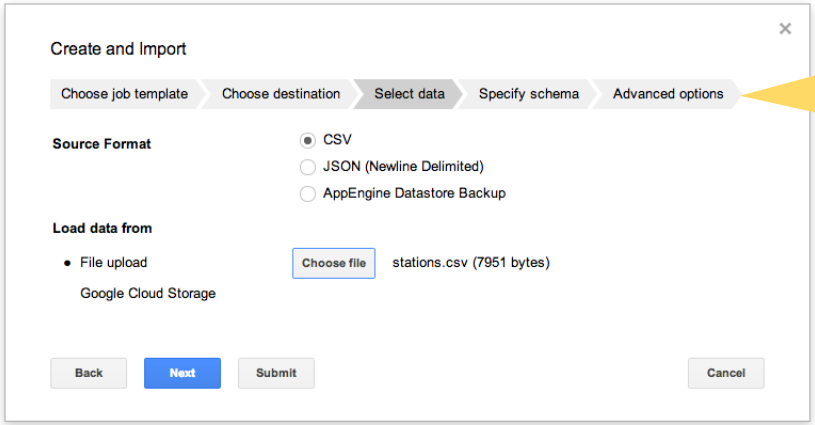
https://bigquery.cloud.google.com/

- BigQuery offers a web console to upload any data set and execute any query.
- Also test data sets are available to start using the system immediately.



BigQuery - Demo - Loading Data

- It will not allow to create data sets neither upload information until we activate the billing for the project.
- Two ways for uploading data:
  - Coming directly from our machine
  - Or we can use Google Cloud Storage as intermediary (better performance for big data sets).
  - Data uploaded as CSV or JSON (flat or nested) files.
- Simple example: Upload list of stations from our machine
  - Create a new dataset named "hubway" and create a new table "stations"
  - Table schema: id:integer, name:string, des:string, install:boolean, locked:boolean, temporary: boolean, lat:float, lng:float



List of steps to upload data from local disk

Data source for the example: http://hubwaydatachallenge.org/submission/leaderboard/

END