Neelu Tiwari — Updated On July 7th, 2021

Advanced Cloud Computing Data Engineering Data Warehouse python

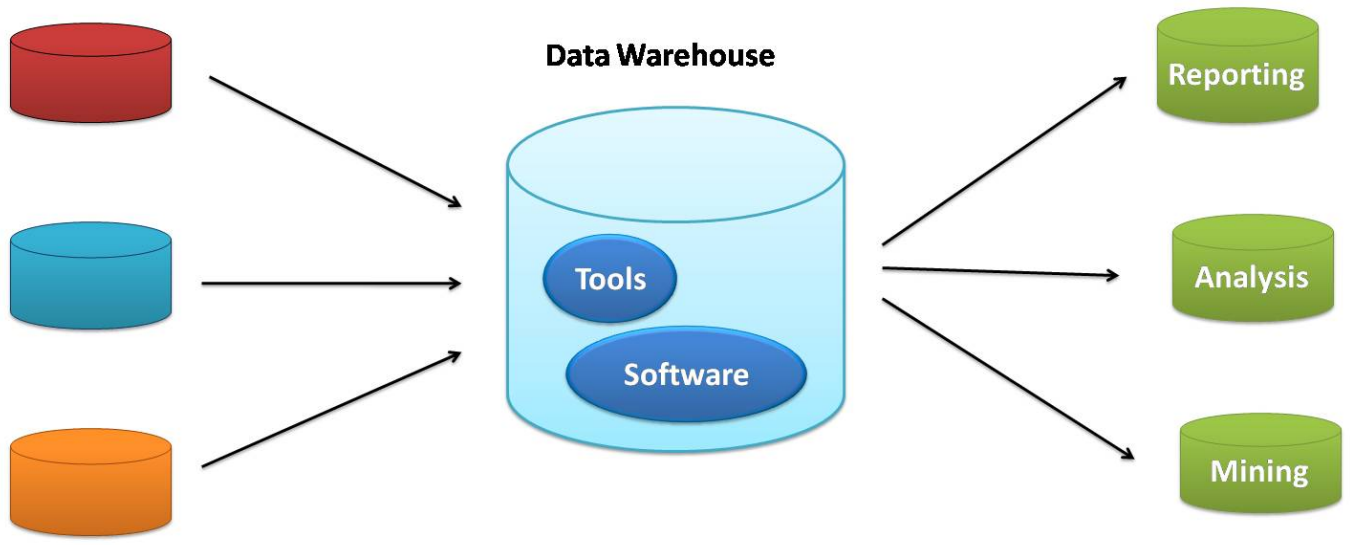
This article was published as a part of the Data Science Blogathon

Introduction

A Data Warehouse is Built by combining data from multiple diverse sources that support analytical reporting, structured and unstructured queries, and decision making for the organization, and Data Warehousing is a step-by-step approach for constructing and using a Data Warehouse. Many data scientists get their data in raw formats from various sources of data and information. But, for many data scientists also as business decision-makers, particularly in big enterprises, the main sources of data and information are corporate data warehouses. A data warehouse holds data from multiple sources, including internal databases and Software (SaaS) platforms. After the data is loaded, it often cleansed, transformed, and checked for quality before it is used for analytics reporting, data science, machine learning, or anything.

What is Data Warehouse?

A Data Warehouse is a collection of software tools that facilitates analysis of a large set of business data used to help an organization make decisions. A large amount of data in data warehouses comes from numerous sources such that internal applications like marketing, sales, and finance; customer-facing apps; and external partner systems, among others. It is a centralized data repository for analysts that can be queried whenever required for business benefits. A data warehouse is mainly a data management system that's designed to enable and support business intelligence (BI) activities, particularly analytics. Data warehouses are alleged to perform queries, cleaning, manipulating, transforming and analyzing the data and they also contain large amounts of historical data.



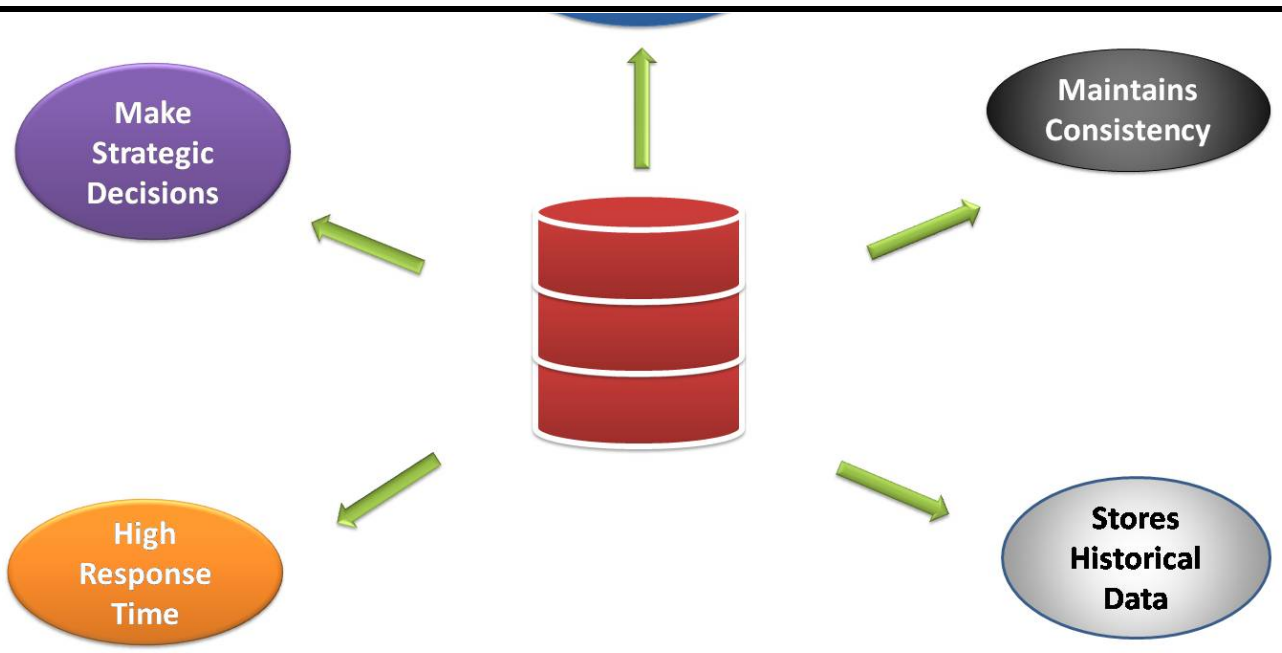
[Source]

What is Data Warehousing?

The process of creating data warehouses to store a large amount of data is named Data Warehousing. Data Warehousing helps to improve the speed and efficiency of accessing different data sets and makes it easier for company decision-makers to obtain insights that will help the business and promoting marketing tactics that set them aside from their competitors. We can say that it is a blend of technologies and components which aids the strategic use of data and information. The main goal of data warehousing is to create a hoarded wealth of historical data that can be retrieved and analyzed to supply helpful insight into the organization's operations.

Need of Data Warehousing.

Data Warehousing is a progressively essential tool for business intelligence. It allows organizations to make quality business decisions. The data warehouse benefits by improving data analytics, it also helps to gain considerable revenue and the strength to compete more strategically in the market. By efficiently providing systematic, contextual data to the business intelligence tool of an organization, the data warehouses can find out more practical business strategies.



[Source]

1. **Business User:** Business users or customers need a data warehouse to look at summarized data from the past. Since these people are coming from a non-technical background also, the data may be represented to them in an uncomplicated way.
2. **Maintains consistency:** Data warehouses are programmed in such a way that they can be applied in a regular format to all collected data from different sources, which makes it effortless for company decision-makers to analyze and share data insights with their colleagues around the globe. By standardizing the data, the risk of error in interpretation is also reduced and improves overall accuracy.
3. **Store historical data:** Data Warehouses are also used to store historical data that means, the time variable data from the past and this input can be used for various purposes.
4. **Make strategic decisions:** Data warehouses contribute to making better strategic decisions. Some business strategies may be depending upon the data stored within the data warehouses.
5. **High response time:** Data warehouse has got to be prepared for somewhat sudden masses and type of queries that demands a major degree of flexibility and fast latency.

Characteristics of Data warehouse:

1. **Subject Oriented:** A data warehouse is often subject-oriented because it delivers may be achieved on a particular theme which means the data warehousing process is proposed to handle a particular theme that is more defined. These themes are often sales, distribution, selling. etc.
2. **Time-Variant**: When the data is maintained via totally different intervals of time like weekly, monthly, or annually, etc. It founds numerous time limits that are unit structured between the big datasets and are command within the online transaction method (OLTP). The time limits for the data warehouse are extended than that of operational systems. The data resided within the data warehouse is predetermined with a particular interval of time and delivers information from the historical perspective. It contains parts of time directly or indirectly.
3. **Non-volatile:** The data residing in the data warehouse is permanent and defined by its names. It additionally means that the data in the data warehouse is cannot be erased or deleted

Operations such as delete, update and insert that is done in a software application over data is lost in the data warehouse environment. There are only two types of data operations that can be done in the data warehouse:

- Data Loading
- Data Access

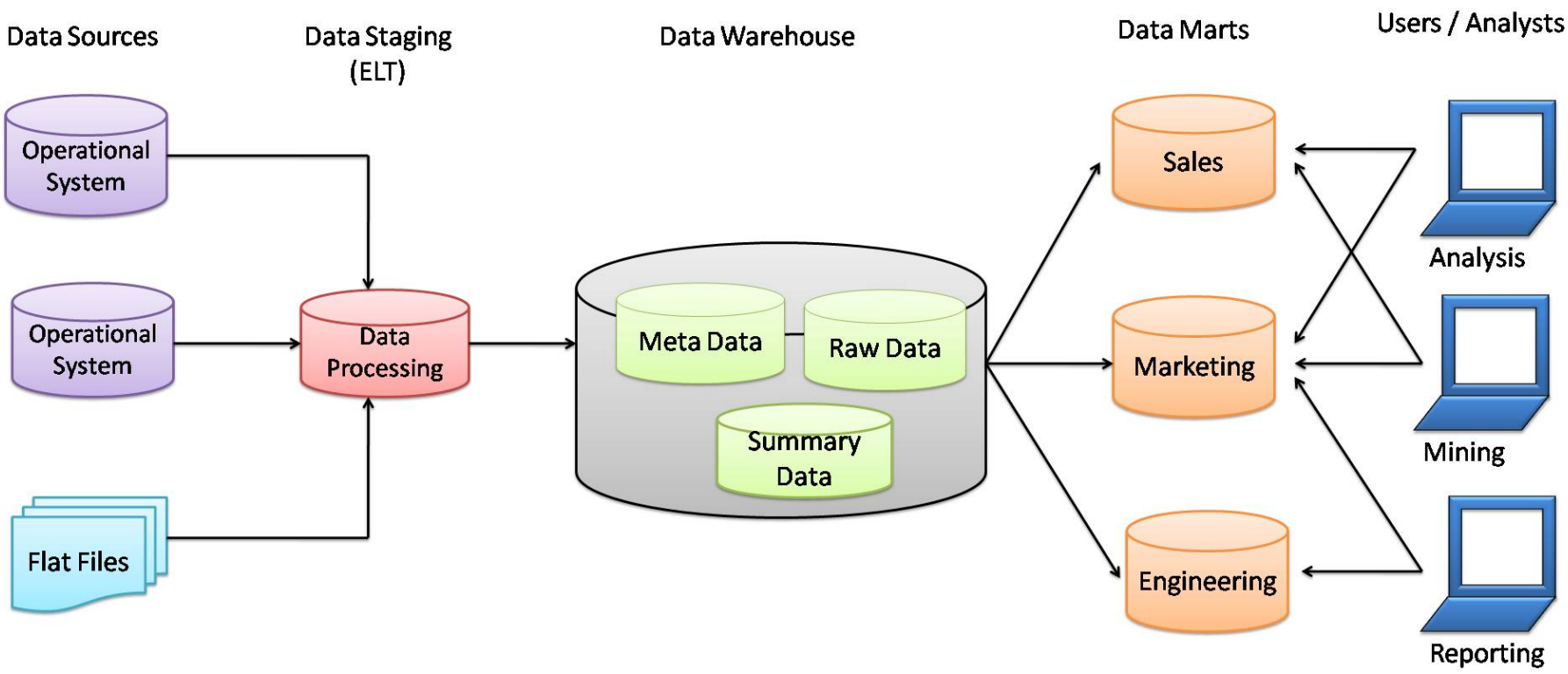
4. **Integrated:** A data warehouse is created by integrating data from numerous different sources such that from mainframe computers and a relational database. Additionally, it should also have reliable naming conventions, formats, and codes. Integration of data warehouse benefits in the successful analysis of data. Dependability in naming conventions, column scaling, encoding structure, etc. needs to be confirmed. Integration of data warehouse handles numerous subject-oriented warehouses.

Architecture & Components of Data Warehouse:

Data warehouse architecture defines the comprehensive architecture of data processing and presentation that will be useful for data analysis and decision making within the enterprise and organization. Each organization has different data warehouses depending upon their need, but all of them are characterized by some standard components.

Data Warehouse applications are designed to support the user's data requirements, an example of this is online analytical processing (OLAP). These include functions such as forecasting, profiling, summary reporting, and trend analysis.

The architecture of the data warehouse mainly consists of the proper arrangement of its elements, to build an efficient data warehouse with software and hardware components. The elements and components may vary based on the requirement of organizations. All of these depend on the organization's circumstances.



DATA WAREHOUSE ARCHITECTURE

Source

1. Source Data Component:

In the Data Warehouse, the source data comes from different places. They are group into four categories:

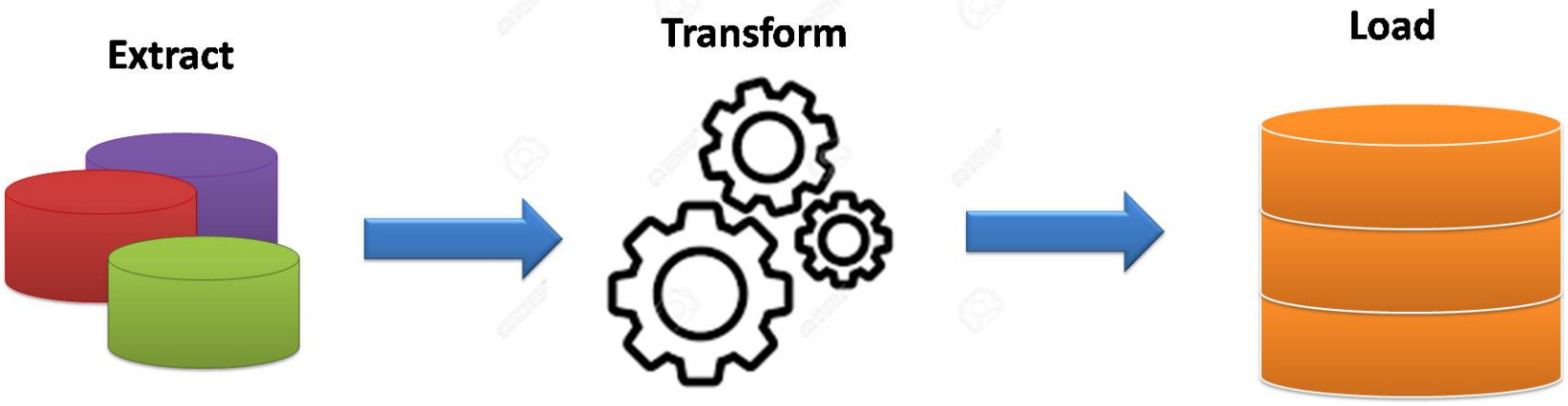
their organization that is brought out by some external sources and department.

- **Internal Data:** In every organization, the consumer keeps their "private" spreadsheets, reports, client profiles, and generally even department databases. This is often the interior information, a part that might be helpful in every data warehouse.
- **Operational System data:** Operational systems are principally meant to run the business. In each operation system, we periodically take the old data and store it in achieved files.
- **Flat files:** A flat file is nothing but a text database that stores data in a plain text format. Flat files generally are text files that have all data processing and structure markup removed. A flat file contains a table with a single record per line.

2. Data Staging:

After the data is extracted from various sources, now it's time to prepare the data files for storing in the data warehouse. The extracted data collected from various sources must be transformed and made ready in a format that is suitable to be saved in the data warehouse for querying and analysis. The data staging contains three primary functions

that take place in this part:



Source

- **Data Extraction:** This stage handles various data sources. Data analysts should employ suitable techniques for every data source.
- **Data Transformation:** As we all know, information for a knowledge warehouse comes from many alternative sources. If information extraction for a data warehouse posture huge challenges, information transformation gifts even important challenges. We tend to perform many individual tasks as a part of information transformation. First, we tend to clean the info extracted from every source of data. Standardization of information elements forms an outsized part of data transformation. Data transformation contains several kinds of combining items of information from totally different sources. Information transformation additionally contains purging supply information that's not helpful and separating outsourced records into new mixtures. Once the data transformation performs ends, we've got a set of integrated information that's clean, standardized, and summarized.
- **Data Loading:** When we complete the structure and construction of the data warehouse and go live for the first time, we do the initial loading of the data into the data warehouse storage. The initial load moves high volumes of

Data storage for data warehousing is split into multiple repositories. These data repositories contain structured data in a very highly

normalized form for fast and efficient processing.

- **Metadata:** Metadata means data about data i.e. it summarizes basic details regarding data, creating findings & operating with explicit instances of data. Metadata is generated by an additional correction or automatically and can contain basic information about data.
- **Raw Data:** Raw data is a set of data and information that has not yet been processed and was delivered from a particular data entity to the data supplier and hasn't been processed nonetheless by machine or human. This data is gathered out from online sources to deliver deep insight into users' online behavior.
- **Summary Data or Data summary:** Data summary is an easy term for a brief conclusion of an enormous theory or a paragraph. This is often one thing where analysts write the code and in the end, they declare the ultimate end in the form of summarizing data. Data summary is the most essential thing in data mining and processing.

4. Data Marts:

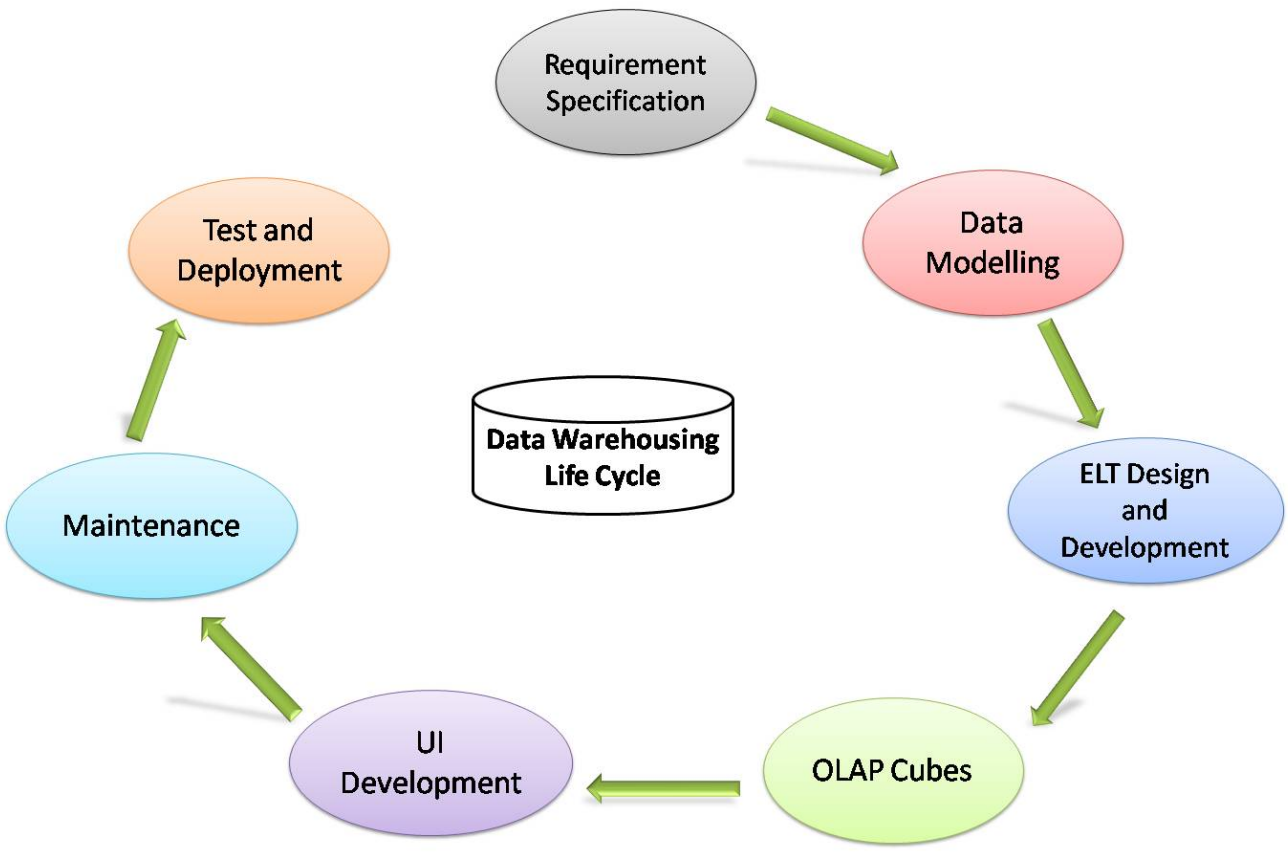
Data marts are also the part of storage component in a data warehouse. It can store the information of a specific function of an organization that is handled by a

single authority. There may be any number of data marts in a particular organization depending upon the functions. In short, data marts contain subsets of the data stored in data warehouses.

Now, the users and analysts can use data for various applications like reporting, analyzing, mining, etc. The data is made available to them whenever required.

Data Warehousing life Cycle:

As we know the data warehouse is made by combining data from multiple diverse sources and the tools that support analytical reporting, structured and unstructured queries, and decision making for the organization. We need to follow the step by step approach for building and successfully implementing the Data Warehouse:



Source

data flows from the transactional system and relational databases. A data warehouse timely pulls out the data from various apps and systems, after then, the data goes through various processing and formatting and makes the data in a format that matches the data already in the warehouse. This processed data is stored in the data warehouses that ready for further analysis for decision making. The data formatting and processing depends upon the need of the organization

The Data could be in one of the following formats:

1. Structured
2. Semi-structured
3. Unstructured data

The data is processed and transformed so that users and analysts can access the processed data in the Data Warehouse through Business Intelligence tools, SQL clients, and spreadsheets. A data warehouse merges all information coming from various sources into one global and complete database. By merging all of this information in one place, it becomes easier for an organization to analyze its customers more comprehensively.

Latest Tools and Technologies for Data Warehousing:

Data warehousing had improved the access to information, reduced query-response time, and also allows businesses to get deep insights from huge big data. Earlier, companies had to build lots of infrastructure for data warehousing. But today the cloud technology has remarkably reduced the cost and effort of data warehousing for businesses.

The field of data warehousing is most emerging and there various cloud data warehousing tools and technologies are developed for better decision making. The cloud-based data warehousing tools are fast, highly scalable, and available on a pay-per-use basis. Following are some data warehousing tools:

1. Amazon Redshift
2. Microsoft Azure
3. Google BigQuery
4. Snowflake
5. Micro Focus Vertica
6. Teradata
7. Amazon DynamoDB
8. PostgreSQL
9. Amazon RD
10. Amazon S3

All these are the top 10 Data Warehousing Tools. In this article, we are going to use Google BigQuery for data warehousing.

easily moving businesses. Google BigQuery is a cloud-based enterprise data warehouse that provides supports to rapid SQL queries and interactive analysis of Big datasets. So, let's get started:

[Source]

Before we begin

We must use the Cloud Resource Manager to Create a Cloud Platform project if you do not already have one, enable billing for the project and then enable BigQuery APIs for the project.

Now, Provide your credentials to the runtime

```
from google.colab import auth
auth.authenticate_user()
print('Authenticated')
```

Enable data table display

Colab includes the google.colab.data_table package that can be used to display large pandas dataframes as an interactive data table. It can be enabled with:

```
%load_ext google.colab.data_table
```

If we want to return the classic Pandas dataframe display, we can disable this by running:

```
%unload_ext google.colab.data_table
```

Use BigQuery via magics

The google.cloud.bigquery library also includes a magic command which runs a query and either displays the result or saves it to a variable as a DataFrame.

```
# Display query output immediately
```

```
%%bigquery --project yourprojectid
```

```
SELECT
  COUNT(*) as total_rows
FROM `bigquery-public-data.samples.gsod`
```

```
# Save output in a variable `df`
```

```
%%bigquery --project yourprojectid df
```

```
SELECT
  COUNT(*) as total_rows
FROM `bigquery-public-data.samples.gsod`
df
```

Use BigQuery through google-cloud-bigquery and declare the Cloud project ID which will be used throughout this notebook:

```
project_id = '[your project ID]'
```

```
sample_count = 2000
row_count = client.query('''
  SELECT
    COUNT(*) as total
  FROM `bigquery-public-data.samples.gsod`''').to_dataframe().total[0]
df = client.query('''
  SELECT
    *
  FROM
    `bigquery-public-data.samples.gsod`
  WHERE RAND() < %d/%d
''' % (sample_count, row_count)).to_dataframe()
print('Full dataset has %d rows' % row_count)
```

Describe the sampled data

```
df.describe()
```

View the first 10 rows

```
df.head(10)
```

Use BigQuery through pandas-gbq

The pandas-gbq library is a community-led project by the pandas community. It covers basic functionality, such as writing a DataFrame to BigQuery and running a query, but as a third-party library, it may not handle all BigQuery features or use cases.

```
import pandas as pd

sample_count = 2000
df = pd.io.gbq.read_gbq('''

  SELECT name, SUM(number) as count

  FROM `bigquery-public-data.usa_names.usa_1910_2013`

  WHERE state = 'TX'
  GROUP BY name
  ORDER BY count DESC
  LIMIT 100
  ''', project_id=project_id, dialect='standard')


df.head()
```

```
SELECT

  COUNT(*) as total_rowsFROM

  `bigquery-public-data.samples.gsod`

''')

pd.io.gbq.read_gbq(query, project_id=project_id, dialect='standard')
```

This brings us an end to this article. Hope you enjoyed reading the article.

Thanks for reading. Do let me know your comments and feedback in the comment section.

For more articles click here.

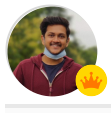
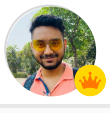
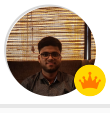
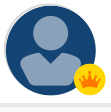
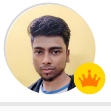
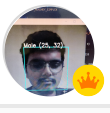
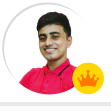
The media shown in this article are not owned by Analytics Vidhya and are used at the Author's discretion.

blogathon Data Engineering data warehouse python

About the Author

Neelu Tiwari

Our Top Authors

view more

Download

Analytics Vidhya App for the Latest blog/Article

Next Post

Facial Landmark Detection Simplified With Opencv

Leave a Reply

Your email address will not be published. Required fields are marked *