**BUILDING A DATA PIPELINE FOR**

**CONSUMER COMPLAINTS**

**COMPANY – INDIA POWER CORPORATION LTD**

**LOCATION – KOLKATA**

**DURATION – 17-06-19 TO 13-07-19 (4 Weeks)**

**SUPERVISORS – ARGHYA ROY**

**SOMNATH CHAKRABORTY**



**ABOUT –**

India Power Corporation Limited (IPCL), formerly known as DPSC Limited and incorporated in 1919, is one of the leading power-generation and utility companies in India. An ISO 9001:2015 entity, it has actively forayed into a diversified portfolio, with renewable and conventional modes of power generation, distribution & power trading.

Having a diversified portfolio of conventional and renewable modes of power generation, multi- location distribution, operation & maintenance and logistic outfits, they are constantly seeking synergic opportunities in the power generation and distribution space to enhance their presence organically and inorganically.

As a Distribution Company, IPCL flies high in its licensed area of 618 sq. km in the Asansol- Raniganj belt of West Bengal. The distribution network is overhead and the current T&D loss is below 3%, lower than the industry average of approximately 25%.

By ensuring 24x7 power supply, competitive pricing, easy payments, instant customer care, customer camps, customer surveys, and online and doorstep services, they have achieved incredible Customer Satisfaction.

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| S.No | Description | Page No. |
| 1. | DATA PIPELINE -  DEFINITION | 1 |
| 2. | WHAT CAN WE DO WITH A DATA PIPELINE | 2 |
| 3. | DATA PIPELINE PROVIDES US WITH | 3-6 |
| 4. | HOW TO BUILD A REAL TIME DATA PIPELINE | 7-12 |
| 5. | TO CREATE A REAL TIME BIG DATA PIPELINE | 13-17 |
| 8. | WHAT IS ETL ? | 18 |
| 9. | DIFFERENCE BETWEEN DATA PIPELINE AND ETL | 19 |
| 10. | VARIOUS ETL TOOLS IN MARKET | 20-27 |
| 11. | BUILDING A REAL TIME DATA PIPELINE  FOR CONSUMER COMPLAINTS | 28-38 |

**DATA PIPELINE**

**DEFINITION –**

In computing , data pipeline is a set of data processing elements connected in series, where the output of one element is the input of the next one. The elements of a pipeline are often executed in parallel or in time-sliced fashion. Some amount of buffer storage is often inserted between elements.

DATA ACQUISITION

DATA PREPARATION

READY FOR

DATA ANALYSIS

**WHAT CAN WE DO WITH A DATA PIPELINE –**

* Convert incoming data to a common format.
* Prepare data for analysis and visualization.
* Migrate between databases.
* Share data processing logic across web apps, batch jobs, and APIs.
* Power your data ingestion and integration tools.
* Consume large XML, CSV, and fixed-width files.



**DATA PIPELINE PROVIDES US**

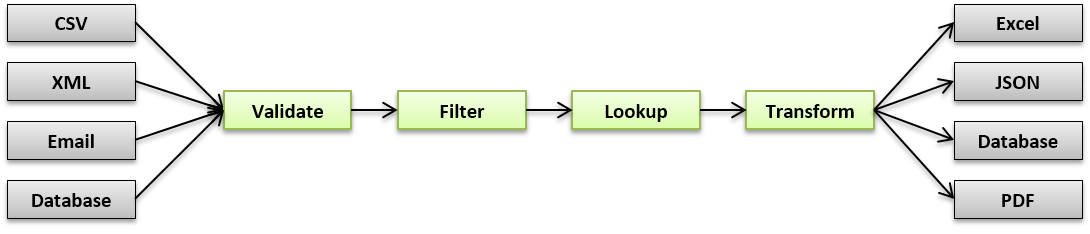
**WITH -**

* **Common Data API -**

Data Pipeline provides you with a single API for working with data. The API treats all data the same regardless of their source, target, format, or structure.

By developing your applications against a single API, you can use the same components to process data regardless of whether they're coming from a database, Excel file, or 3rd-party API. We're also future-proofed when new formats are introduced.

A common API means our team only has one thing to learn, it means shorter development time, and faster time-to-market. It also means less code to create, less code to test, and less code to maintain.



* **Flexible Schema -**

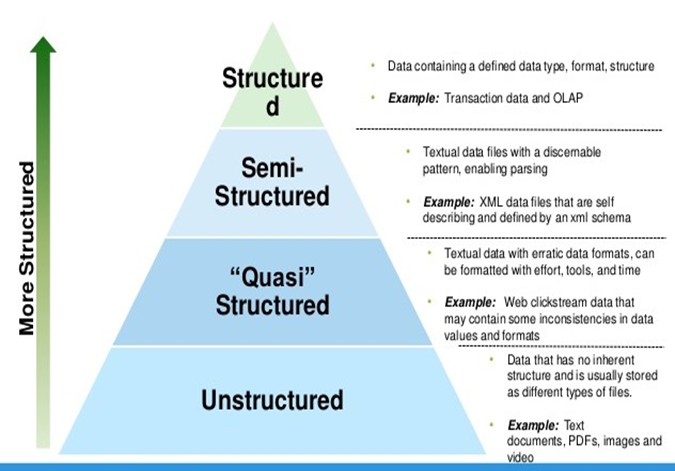
Data Pipeline does not impose a particular structure on your data.

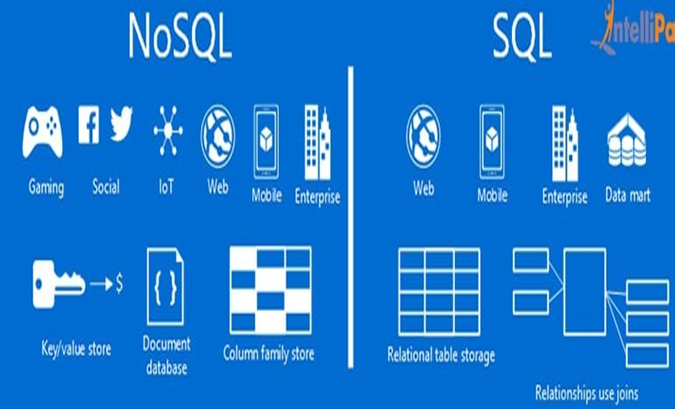
It can store and work with structured data, semi-structured data , quasi-structured data and unstructured data.

SQL databases (MySql ,MS-SQL) are designed to handle structured data and NoSQL databases (Mongo DB , Hbase) are mainly designed to handle unstructured data.

Semi- structured data (XML files) are those data which are not complexly organized as RDBMS but do have some organizational properties (metadata) associated with them which make them a bit easier to analyze than the unstructured data.

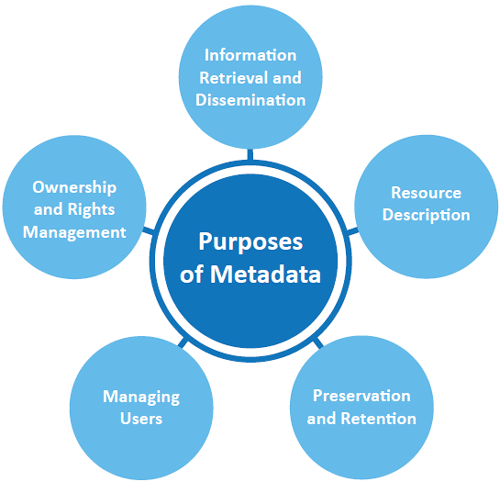
Quasi-structured data are web- clickstream data.





* **Metadata -**

Data Pipeline allows you to associate metadata to each individual record or field. Metadata can be any arbitrary information you like. For example, you can use it to track where the data came from, who created it, what changes were made to it, and who's allowed to see it



* **Process large volumes of data**

Data Pipeline views all data as streaming. Regardless of whether the data is coming from a local Excel file, a remote database, or an online service like Twitter. By breaking dataflows into smaller units, you're able to work with datasets that are orders of magnitude larger than your available memory.



**HOW TO BUILD A REAL TIME DATA PIPELINE -**

To build real-time data pipelines, we need infrastructure and technologies that accommodate ultrafast data capture and processing. Real-time technologies share the following characteristics:

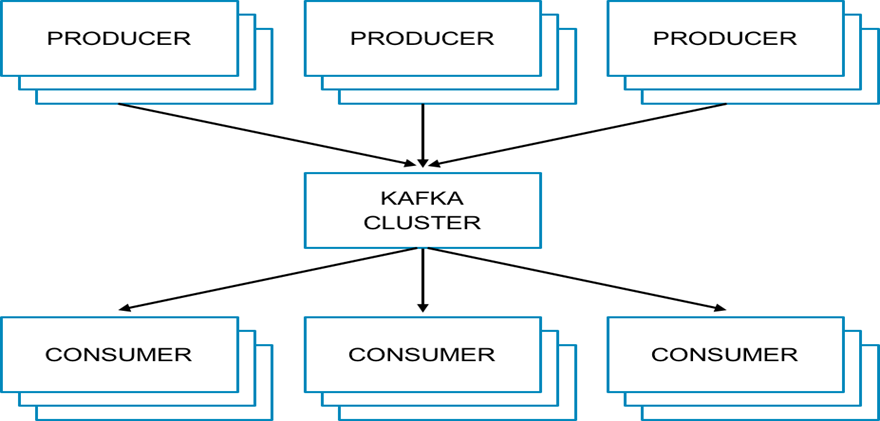
* in-memory data storage for high-speed ingest.
* distributed architecture for horizontal scalability.
* they are queryable for real-time, interactive data exploration.



**High Throughput Messaging System –**

Many real-time data pipelines begin with capturing data at its source and using a high-throughput messaging system to ensure that every data point is recorded in its right place. Data can come from a wide range of sources, including logging information, web events, sensor data, financial market streams, and mobile applications. From there it is written to file systems, object stores, and databases.

Apache Kafka is an example of a high-throughput, distributed messaging system and is widely used across many industries.



Kafka is a distributed, partitioned, replicated commit log service. Kafka can handle terabytes of messages without performance impact.

Because of its distributed characteristics, Kafka is built to scale producers and consumers with ease by simply adding servers to the cluster. Kafka’s effective use of memory, combined with a commit log on disk, provides ideal performance for real-time pipelines and durability in the event of server failure.

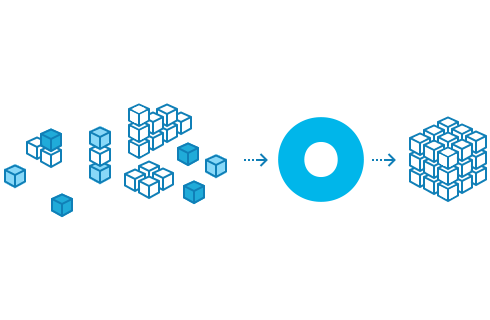
A close up of a sign

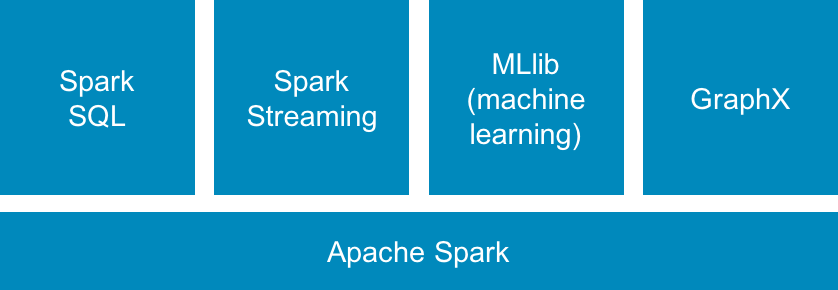
Description automatically generated

**Data Transformation -**

The data transformation tier takes raw data, processes it, and outputs the data in a format more conducive to analysis. Transformers serve a number of purposes including data enrichment, filtering, and aggregation.

Apache Spark is often used for data transformation. Like Kafka, Spark is a distributed, memory-optimized system that is ideal for real-time use cases. Spark also includes a streaming library and a set of programming interfaces to make data processing and transformation easier.





When building real-time data pipelines, Spark can be used to extract data from Kafka, filter down to a smaller dataset, run enrichment operations, augment data, and then push that refined dataset to a persistent datastore.

Spark does not include a storage engine, which is where an operational database comes into play.

OPERATIONAL

DATABASE

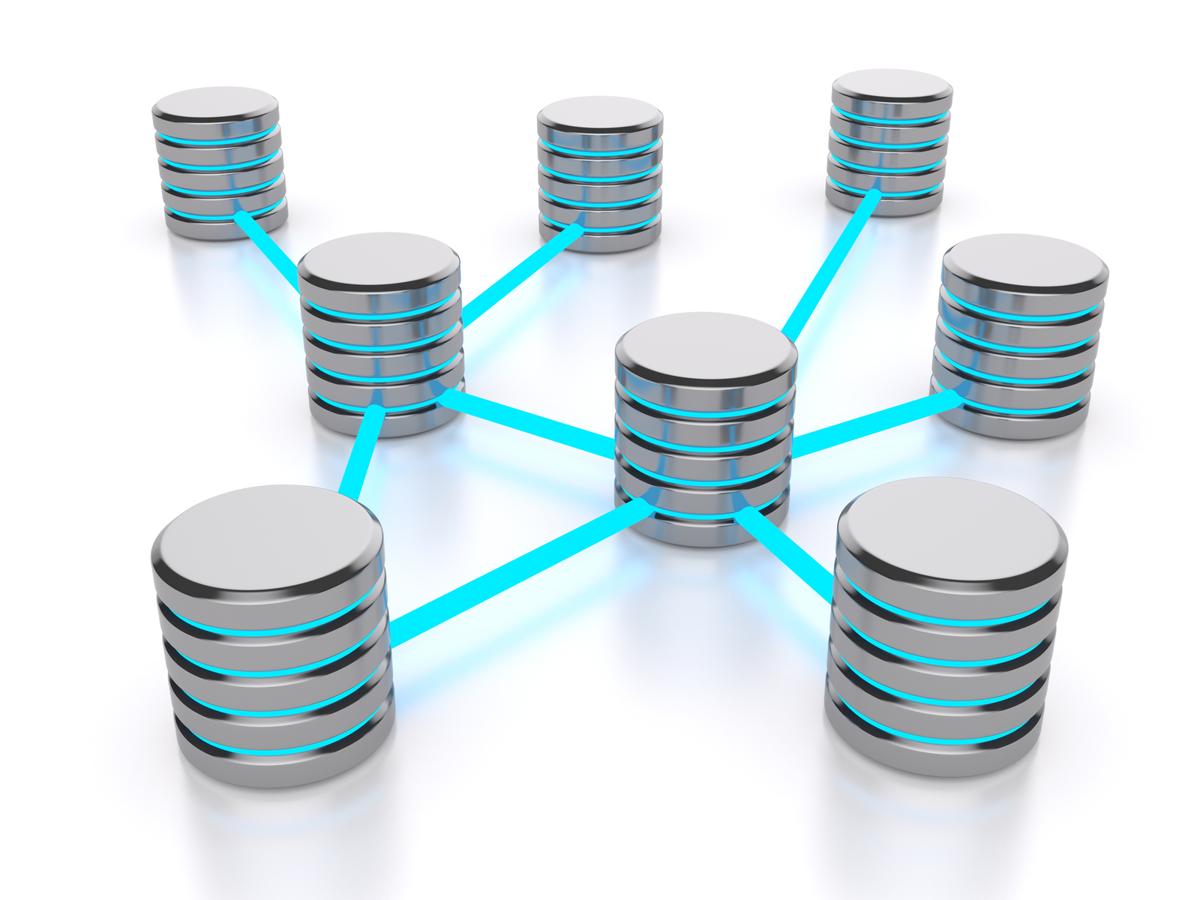
(BI-DIRECTIONAL)

SPARK

**Datastore (Operational Database) –**

To analyze both real-time and historical data, it must be maintained beyond the streaming and transformations layers of our pipeline, and into a permanent datastore. Although unstructured systems like Hadoop Distributed File System (HDFS) or Amazon S3 can be used for historical data persistence, neither offer the performance required for real-time analytics.

On the other hand, a memory-optimized database can provide persistence for real-time and historical data as well as the ability to query both in a single system. By combining transactions and analytics in a memory-optimized system, data can be rapidly ingested from our transformation tier and held in a datastore. This allows applications to be built on top of an operational database that supplies the application with the most recent data available.



**TO CREATE A REAL TIME BIG DATA PIPELINE –**

There are some key points that we need to measure while selecting a tool or technology for building a big data pipeline which is as follows:

Components

Parameters

**Components -**

* The messaging system.
* Message distribution support to various nodes for further data processing.
* Data analysis system to derive decisions from data.
* Data storage system to store results and related information.
* Data representation and reporting tools and alerts system.

**Parameters -**

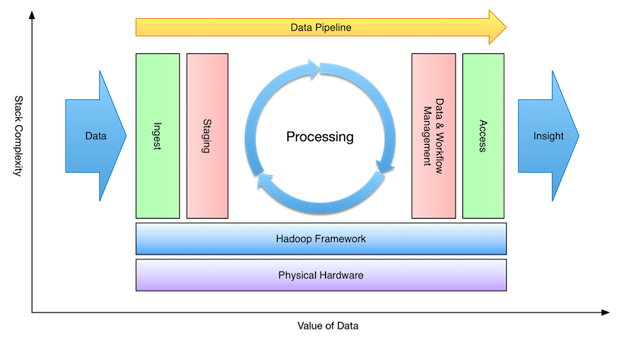
* Compatible with big data
* Low latency
* Scalability
* A diversity that means it can handle various use cases
* Flexibility
* Economic

The choice of technologies like Apache Hadoop, Apache Spark, and Apache Kafka address the above aspects. Hence, these tools are the preferred choice for building a real-time big data pipeline.

The two core processes in real time big data pipeline system –

The messaging system

The data ingestion process



The messaging system is the entry point in a big data pipeline and Apache Kafka is a publish-subscribe messaging system work as an input system. For messaging, Apache Kafka provide two mechanisms utilizing its APIs –

* Producer
* Subscriber

Using the Priority queue, it writes data to the producer. Then the data is subscribed by the listener. It could be a Spark listener or any other listener. Apache Kafka can handle high-volume and high-frequency data.

A screenshot of a cell phone

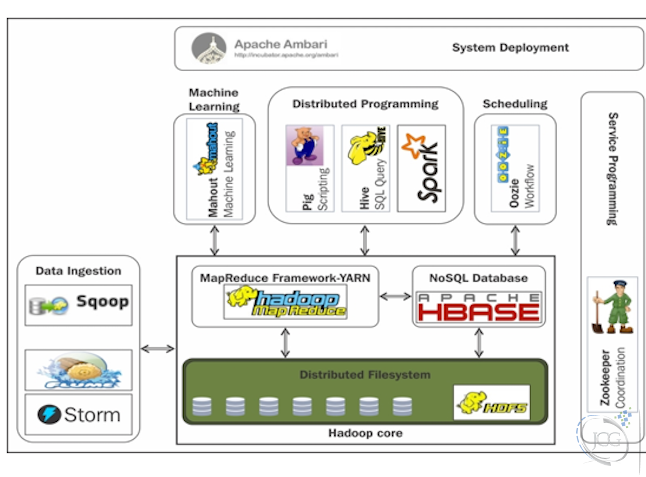
Description automatically generated

Once the data is available in a messaging system, it needs to be ingested and processed in a real-time manner. Apache Spark makes it possible by using its streaming APIs.

A picture containing text, map

Description automatically generated

Apache Hadoop provides an ecosystem for the Apache Spark and Apache Kafka to run on top of it. Additionally, it provides persistent data storage through its HDFS.



The architecture used here is Kappa Architecture. We can also use the Lambda Architecture. The reason behind this is we have to maintain two distinct systems to generate both batch and speed layer. But in some cases, however, having access to a complete set of data in a batch window may yield certain optimizations that would make Lambda better performing and perhaps even simpler to implement.

Kappa architecture is comprised of two layers instead of three layers as in the Lambda architecture.

These layers are -

* Real-time layer/Stream processing
* Serving Layer

**Working –**

In this case, the incoming data is ingested through the real-time layer via a messaging system like Apache Kafka.

In the real-time layer or streaming process data is processed. Usually, Apache Spark is used in this layer as it supports both batch and stream data processing.

The output result from the real-time layer is sent to the serving layer which is a backend system like a NoSQL database.

Apache Hadoop provides the eco-system for Apache Spark and Apache Kafka.

A close up of a logo

Description automatically generated

**WHAT IS ETL ?**

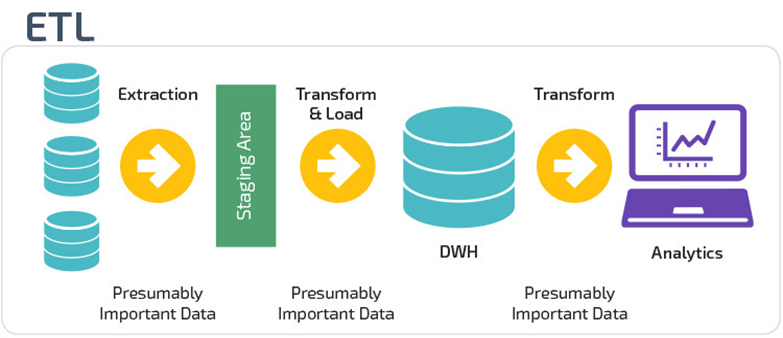
ETL is defined as a process that extracts the data from different RDBMS source systems, then transforms the data (like applying calculations, concatenations, etc.) and finally loads the data into the Data Warehouse system. ETL full-form is Extract, Transform and Load.

The ETL process requires active inputs from various stakeholders including developers, analysts, testers, top executives and is technically challenging.

A close up of a sign

Description automatically generated

In order to maintain its value as a tool for decision-makers, Data warehouse system needs to change with business changes. ETL is a recurring activity (daily, weekly, monthly) of a Data warehouse system and needs to be agile, automated, and well documented.



**DIFFERENCE BETWEEN DATA PIPELINE AND ETL –**

|  |  |
| --- | --- |
| **DATA PIPELINE** | **ETL** |
| Data pipeline is a slightly more generic term. It refers to any set of processing elements that move data from one system to another, possibly transforming the data along the way. | ETL pipeline refers to a set of processes extracting data from one system, transforming it, and loading into some database or data-warehouse. |
| Another type of a data pipeline that is not an ETL pipeline, is an ELT pipeline (note the order of the letters): loading all of your data to the data warehouse, and transforming it only later.  Additionally, a data pipeline doesn’t have to end in loading the data to a database or a data warehouse. It can load it to any other source. | The term ETL pipeline usually implies that the pipeline works in batches - for example, the pipe is run once every 12 hours, while data pipeline can also be run as a streaming computation (meaning, every event is handled as it occurs). |

**VARIOUS ETL TOOLS IN MARKET –**

**HEVO –**

Hevo is a Unified Cloud Data Integration Platform for Real-time Analytics that helps companies analyze data from scattered across multiple sources.

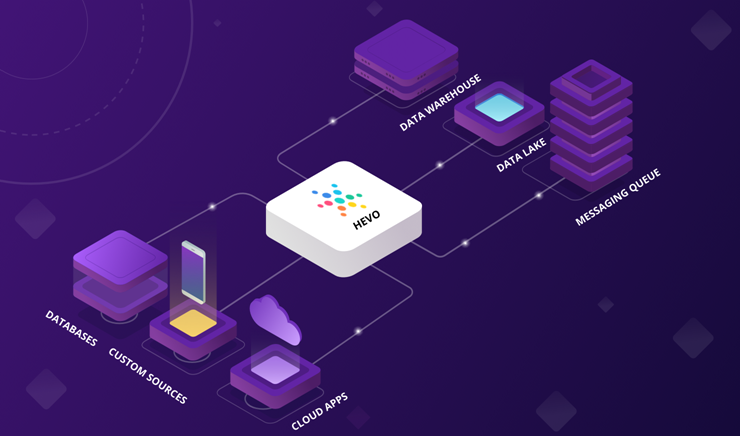
Hevo helps bring data from both structured and unstructured sources like SaaS Applications, Databases, SDKs, Cloud Storage, etc. into Data Warehouse like Amazon Redshift, Snowflake, and BigQuery in real-time.

A close up of a sign

Description automatically generated

**Features –**

* Hassle-free, code-free ETL. No ETL Script maintenance or Cron jobs required
* Point and Click Interface that allows moving data from any source to any Data Warehouse in minutes
* Support for both ETL and ELT
* Handle data of any scale with Zero data loss
* Automatic Schema Detection and Mapping
* Real-time Monitoring, timely alerts, granular activity logs, and version control
* Priority customer support over slack and email
* Unparallel Data Transformation and Data Cleaning Capabilities
* Capability to build aggregates and joins (Data Models) on Data Warehouse for faster query processing



**Skyvia –**

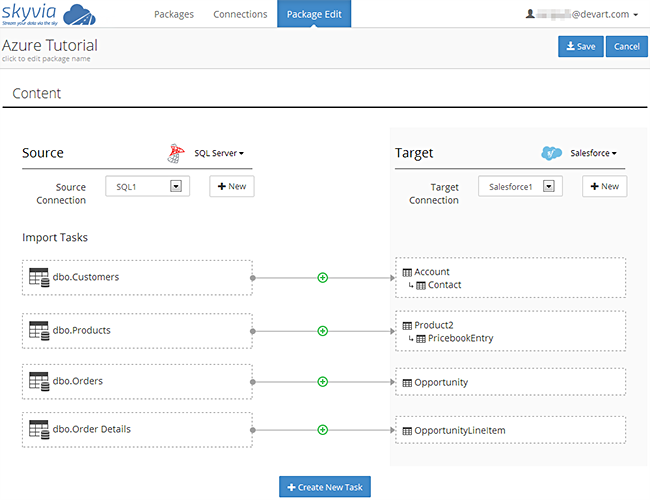
Skyvia is a cloud data platform for no-coding data integration, backup, management and access, developed by Devart.

Skyvia includes an ETL solution for various data integration scenarios with support for CSV files, databases (SQL Server, Oracle, PostgreSQL, MySQL), cloud data warehouses (Amazon Redshift, Google BigQuery), and cloud applications (Salesforce, HubSpot, Dynamics CRM and many others). It also includes cloud data backup tool, online SQL client, and OData server-as-a-service solution.



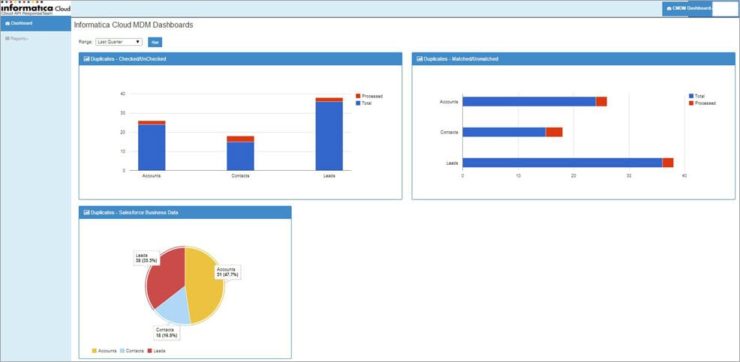
**Features –**

* Skyvia is a commercial, subscription-based cloud solution free plans available
* Wizard-based, no-coding integration configuration does not require much technical knowledge
* Advanced mapping settings with constants, lookups, and powerful expressions for data transformations
* Integration automation by schedule
* Ability to preserve source data relations in target
* Import without duplicates
* Bi-directional synchronization
* Predefined templates for common integration cases



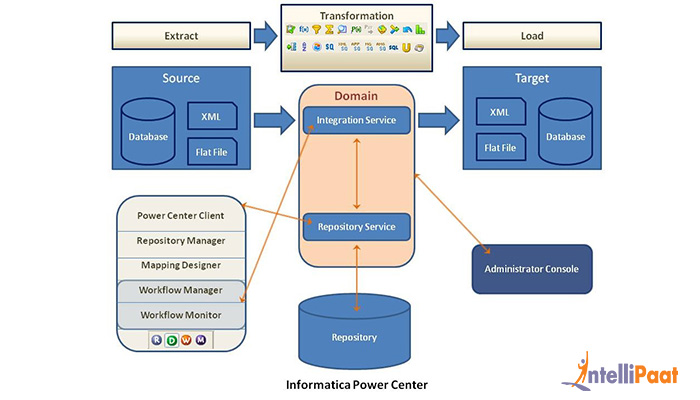
**Informatica – PowerCenter –**

PowerCenter is a product which was developed by Informatica for data integration. It supports data integration lifecycle and delivers critical data and values to the business. PowerCenter supports a huge volume of data and any data type and any source for data integration.



**Features –**

* PowerCenter is a commercial licensed tool.
* It is a readily available tool and has easy training modules.
* It supports data analysis, application migration and data warehousing.
* PowerCenter connects various cloud applications and is hosted by Amazon Web Services and Microsoft Azure.
* PowerCenter supports agile processes.
* It can be integrated with other tools.
* The automated result or data validation across development, testing and production environment.
* A non-technical person can run and monitor jobs which in turn reduces the cost.



**Microsoft – SSIS –**

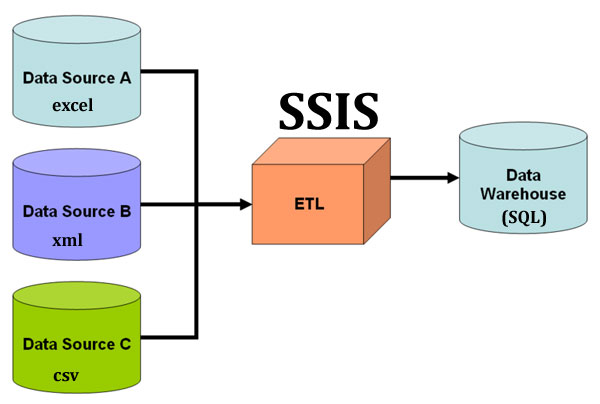
SSIS is a product by Microsoft and was developed for data migration. The data integration is much faster as the integration process and data transformation is processed in the memory. As it is the product of Microsoft, SSIS only supports Microsoft SQL Server.

A screenshot of a social media post

Description automatically generated

**Features –**

* SSIS is a commercial licensed tool.
* SSIS import/export wizard helps to move data from source to destination.
* It automates the maintenance of SQL Server Database.
* Drag and Drop user interface for editing SSIS packages.
* Data transformation includes text files and other SQL server instances.
* SSIS has inbuilt scripting environment available for writing programming code.
* It can be integrated with salesforce.com and CRM using plug-ins.
* Debugging capabilities and easy error handling the flow.
* SSIS can also be integrated with change control software’s like TFS, GitHub etc.



**BUILDING A REAL TIME**

**DATA PIPELINE FOR CONSUMER**

**COMPLAINTS –**

* **Extraction of data from SAP –**

The raw data is extracted from the SAP ERP and is dumped in the machine 10.90.3.12 in .TSV (Tab Separated Values) format. This is handled by the SAP ERP administrator.

* **Transferring the data from the machine 10.90.3.12 –**

The data is transferred from the machine 10.90.3.12 to this machine (localhost) via a scheduler (.bat file).

* **Uploading the file in SQL Server using Import - Export Wizard –**

Upload the raw file in the SQL Server by creating a custom table.

The columns in the table were –

1. [Notification]

2. [Notifictntype]

3. [Description]

4. [Notif date]

5. [Notif time]

6. [Customer]

7. [Businessarea]

8. [Completn date]

9. [Completion time]

* **Transform –**

Transform the raw file by creating a view (summary\_notf) in

the SQL Server.

The code written for view created is as –

select [Notifictntype] ,

(CASE

WHEN [Businessarea]='' THEN 'Others'

ELSE [Businessarea]

END) as Business\_area ,

MONTH(CAST(STUFF(STUFF([Notif date],5,0,'-'),8,0,'-') AS

DATE)) AS [Month] ,

YEAR(CAST(STUFF(STUFF([Notif date],5,0,'-'),8,0,'-') AS

DATE)) AS [Year] ,

SUM( CASE

WHEN [Completn date]='0' THEN 1

ELSE 0

END) as complaints\_open,

(count([Notification]) - SUM( CASE

WHEN [Completn date]='0' THEN 1

ELSE 0

END)) as complaints\_closed

FROM [Consumer Complaint].[dbo].[Complaint]

WHERE [Businessarea] <> 'BG'

group by [Notifictntype],

(CASE

WHEN [Businessarea]='' THEN 'Others'

ELSE [Businessarea]

END) ,

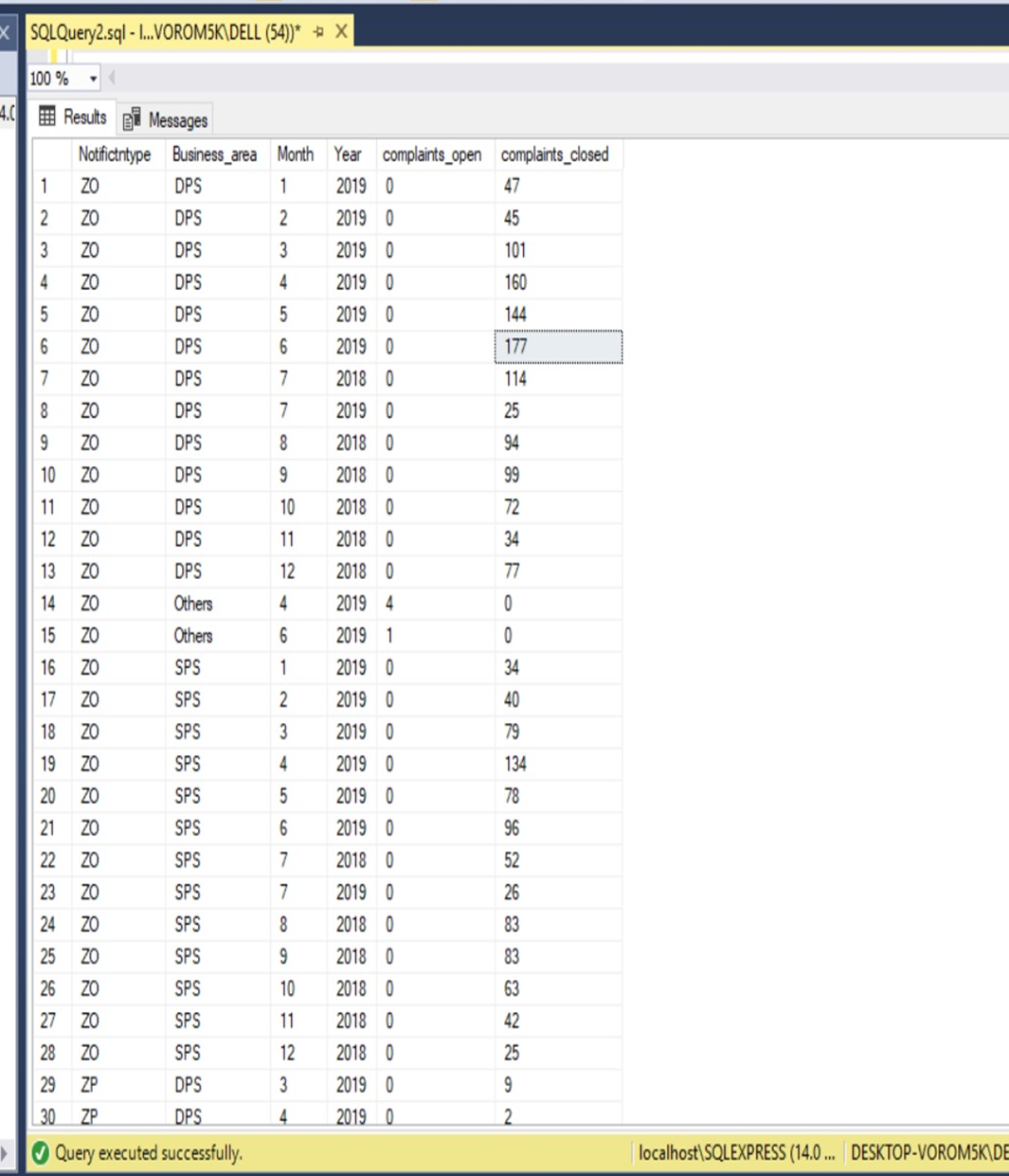
MONTH(CAST(STUFF(STUFF([Notif date],5,0,'-'),8,0,'-') AS DATE)) ,

YEAR(CAST(STUFF(STUFF([Notif date],5,0,'-'),8,0,'-') AS DATE))

* **Execution of the code in the view –**

After executing this code in the view a new table is generated

with necessary details.

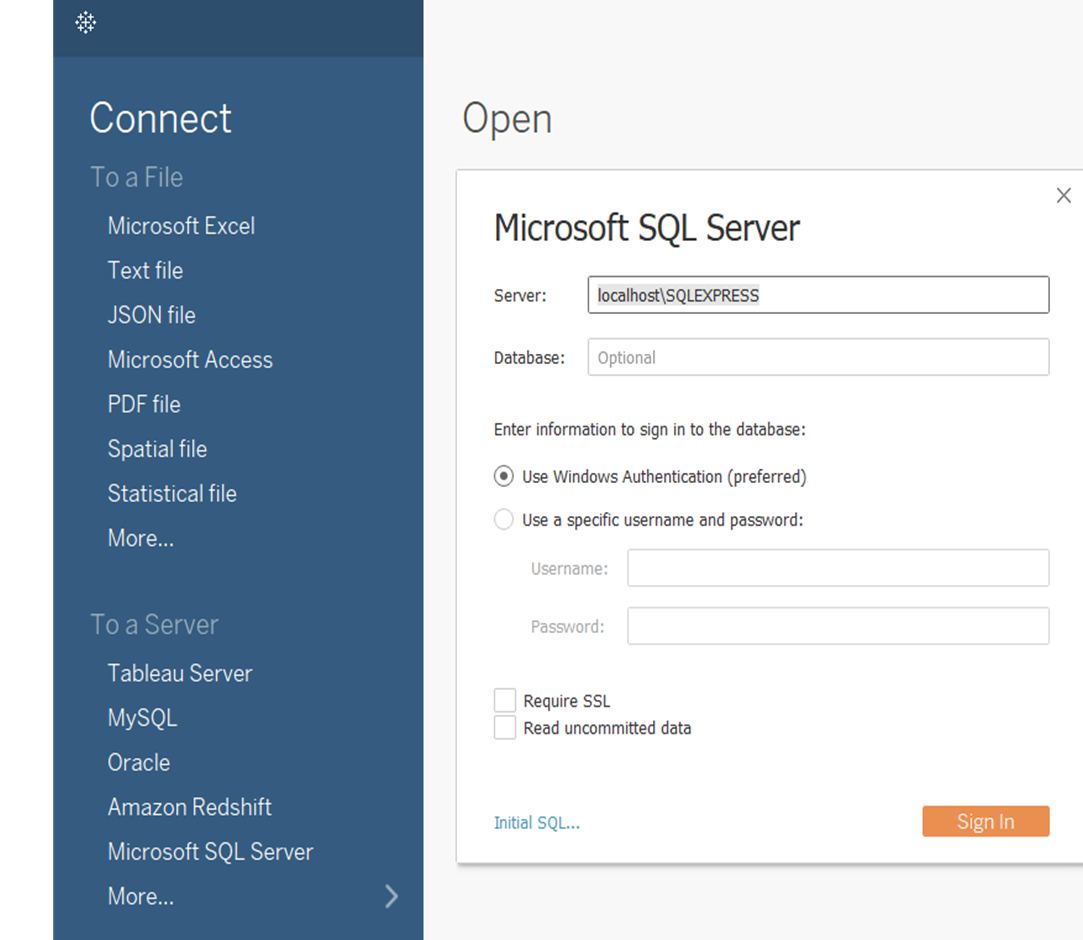


* **Connecting to SQL Server by Tableau** -

Open the Tableau workspace. Then under **To a Server** click on

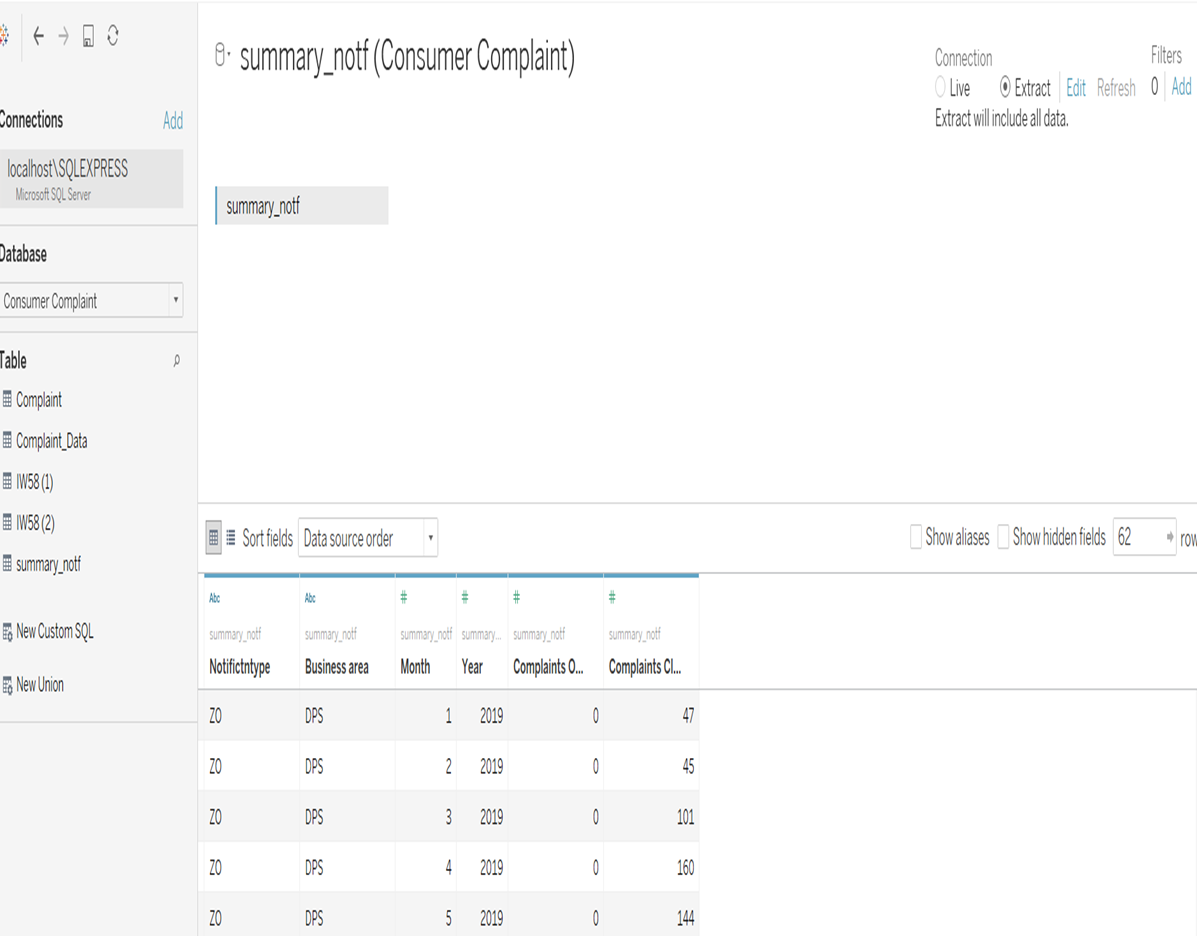
**Microsoft SQL Server**. Specify the **Server** and select **Use**

**Windows Authentication**. Then click on **Sign in**.

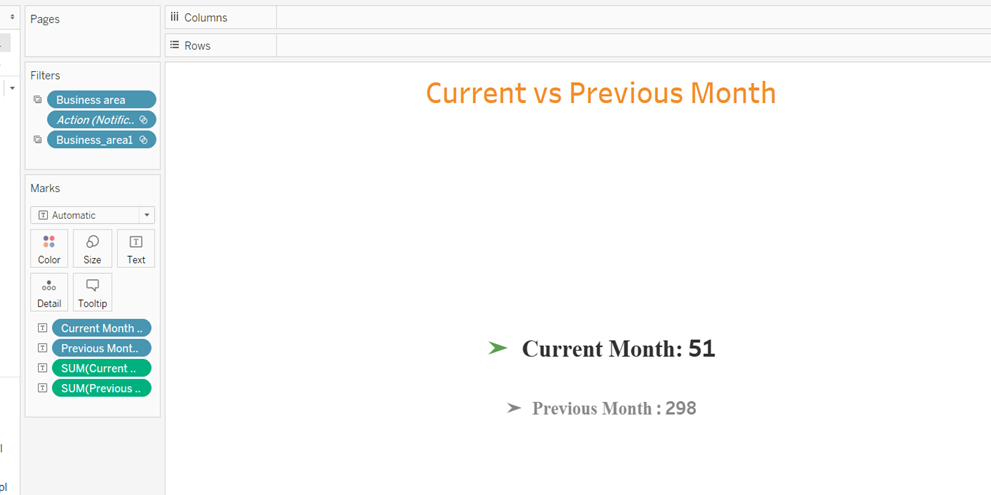


* **Creating Extract –**

Drag the table summary\_notf and create Extract Connection.



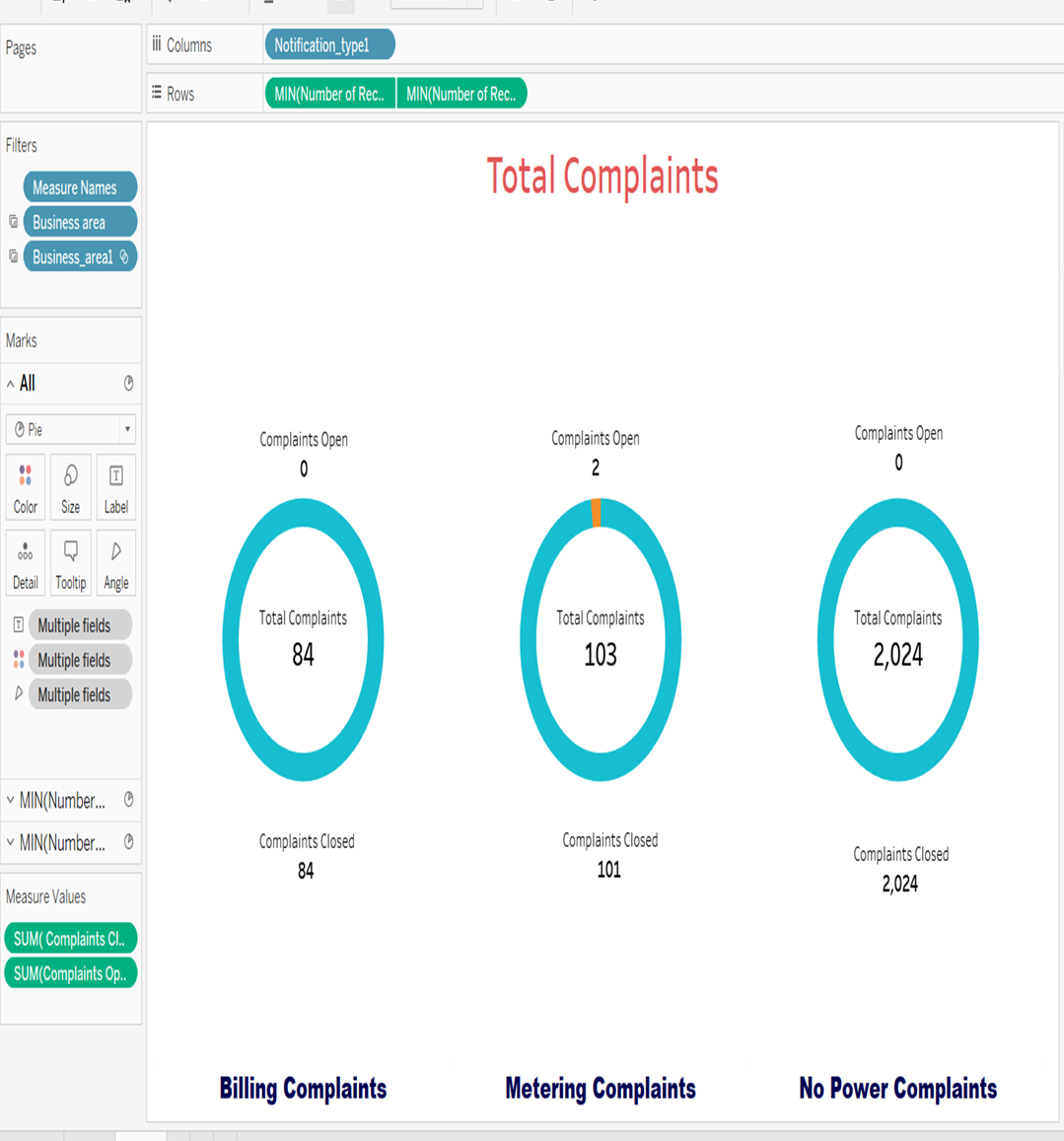
* **Creation of sheet – Current vs Previous Month –**



* **Creation of sheet – Frequency of Complaints – Last Twelve Months –**



* **Creation of sheet – Total Complaints -**



* **Creation of Dashboard in Tableau -**

