**Low Level Design (LLD)**

**Deloitte Case Study**

|  |  |
| --- | --- |
| **Written By** | **Mukilan S**  **Balachandran E**  **Divakar S**  **Sabanayagam B**  **Dharun Anand S**  **Chandru A**  **Azim Ahamed** |
| **Document Version** | **1.0** |
| **Last Revised Date** |  |

**DOCUMENT CONTROL**

**Change Record:**

|  |  |  |  |
| --- | --- | --- | --- |
| **VERSION** | **DATE** | **AUTHOR** | **COMMENTS** |
| 0.1 | **28 MAY 2023** | **Mukilan S** |  |
| 0.2 |  |  |  |

**Reviews:**

|  |  |  |  |
| --- | --- | --- | --- |
| **VERSION** | **DATE** | **REVIEWER** | **COMMENTS** |
|  |  |  |  |

**Approval Status:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **VERSION** | **REVIEW**  **DATE** | **REVIEWED BY** | **APPROVED**  **BY** | **COMMENTS** |
|  |  |  |  |  |

**Contents**

**1. Introduction........................................................................................................................ 04**

**1.1 What is Low-Level Design Document? ....................................................................... 04**

**1.2 Scope ............................................................................................................................ 04**

**2. Functional Architecture...................................................................................................... 05**

**3. Architecture Description ................................................................................................... 08**

**3.1 Data Description .......................................................................................................... 08**

**3.2 Data Transformation .................................................................................................... 12**

**3.3 Tableau Configuration ................................................................................................. 12**

**3.4 Deployment ................................................................................................................. 13**

**4. Reference .......................................................................................................................... 13**

**1 Introduction**

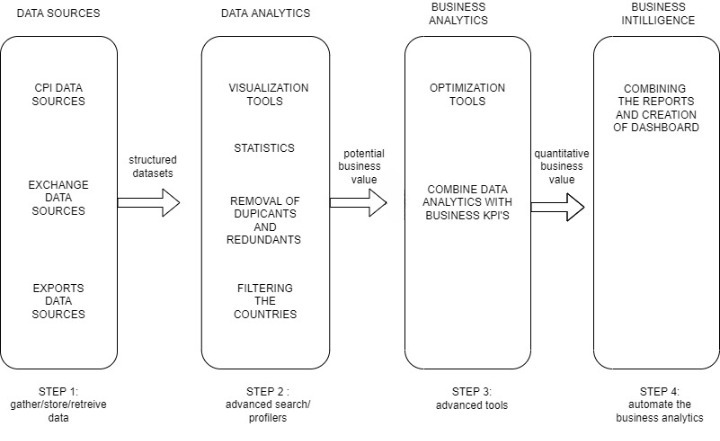
**1.1 Why this Low-Level Design Document?**

The goal of the LDD or Low-level design document (LLDD) is to give the internal logic design of the actual program code for the deloitte case dashboard. LDD describes the class diagrams with the methods and relations between classes and programs specs. It describes the modules so that the programmer can directly code the program from the document.

**1.2 Scope**

**Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. The process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.**

**2 Functional Architecture**



**Tableau Server Architecture**

Tableau has a highly scalable, n-tier client-server architecture that serves mobile clients, web clients and desktop-installed software. Tableau Server architecture supports fast and flexible deployments.

The following diagram shows Tableau Server’s architecture:

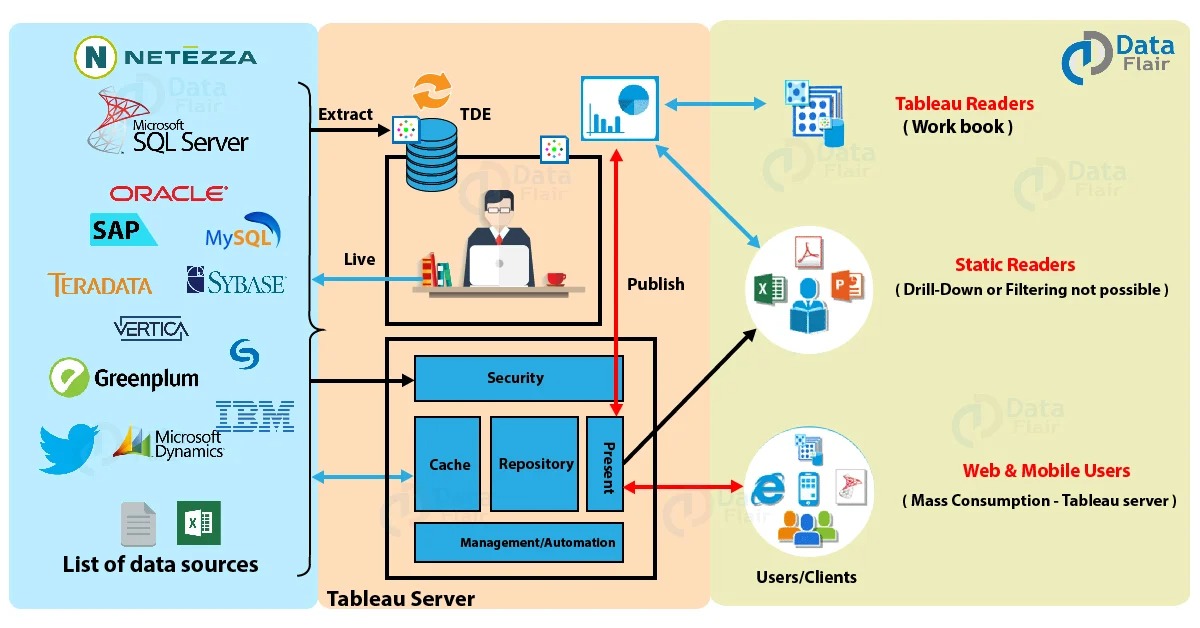


Tableau Server is internally managed by the multiple server processes [1]

**1. Gateway/Load Balancer**

It acts as an Entry gate to the Tableu Server and also balances the load to the Server if multiple Processes are configured.

**2. Application Server**

Application Server processes (wgserver.exe) handle browsing and permissions for the Tableau Server web and mobile interfaces. When a user opens a view in a client device, that user starts a session on Tableau Server. This means that an Application Server thread starts and checks the permissions for that user and that view.

**3. Repository**

Tableau Server Repository is a PostgreSQL database that stores server data. This data includes information about Tableau Server users, groups and group assignments, permissions, projects, data sources, and extract metadata and refresh information.

**4.VIZQL Server**

Once a view is opened, the client sends a request to the VizQL process (vizqlserver.exe). The VizQL process then sends queries directly to the data source, returning a result set that is rendered as images and presented to the user. Each VizQL Server has its own cache that can be shared across multiple users

**5.Data Engine**

It Stores data extracts and answers queries.

**6.Backgrounder**

The backgrounder Executes server tasks which includes refreshes scheduled extracts, tasks initiated from tabcmd and manages other background tasks.

**7.Data Server**

Data Server Manages connections to Tableau Server data sources It also maintains metadata from Tableau Desktop, such as calculations, definitions, and groups.

**3. Architecture Description**

**3.1. Data Description**

The Dataset contains three datasets CPI, EXPORTS and EXCHANGE monthly details on different countries for different year.

**CPI**

Consumer Price Index (CPI) is one of the most popular measures of inflation and deflation. It measures the average change in prices.

**EXPORTS**

Exports are defined as movable goods produced within the boundaries of one country, which are traded with another country. The sale of these goods generates foreign currency earnings in the country that produces them and boosts its economic growth. The greater the proportion of exports in relation to a country’s Gross Domestic Product (GDP), the larger the boost will be to overall growth when overseas demand increases.

**EXCHANGE**

In finance, an exchange rate is the rate at which one currency will be exchanged for another currency. There are many ways to measure an exchange rate. The most common way is tomeasure a bilateral exchange rate. A bilateral exchange rate refers to the value of one currency relative to another. Bilateral exchange rates are typically quoted against the US dollar (USD), as it is the most traded currency globally.

**INFLATION**

Inflation is the percentage change in the value of the Wholesale Price Index (WPI) on a year-on year basis. It effectively measures the change in the prices of a basket of goods and services in a year. In India, inflation is calculated by taking the WPI as base.

Inflation is often used to describe the impact of rising oil or food prices on the economy. For example, if the price of oil goes from $75 a barrel to $100 a barrel, input prices for businesses will increase and transportation costs for everyone will also increase. This may cause many other prices to rise in response.

## Benefits of Inflation:

When the economy is not running at capacity, meaning there is unused labor or resources, inflation theoretically helps increase production. More dollars translate to more spending, which equates to more aggregated demand. More demand, in turn, triggers more production to meet that demand.

British economist John Maynard Keynes believed that some inflation was necessary to prevent the Paradox of Thrift. This paradox states that if consumer prices are allowed to fall consistently because the country is becoming too productive, consumers learn to hold off their purchases to wait for a better deal. The net effect of this paradox is to reduce aggregate demand, leading to less production, layoffs, and a faltering economy.

Inflation also makes it easier on debtors, who repay their loans with money that is less valuable than the money they borrowed. This encourages borrowing and lending, which again increases spending on all levels.

Pros of inflation:

* Leads to higher resale value of assets
* Optimum levels of inflation encourages spending

Cons of inflation:

* Buyers have to pay more for products and services
* Impose higher prices on economy
* Drives some prices up first and others later

Formula of inflation:

**Inflation Rate** = ((B – A) / A) x 100

Where,

A = starting cost

B = ending cost

**Global Data has raised its global inflation rate forecast for the end of 2022 by 2.7 percentage points, reaching 7.5%\***

**Year of Year (YOY)**

Year-on-year growth rates are rates of change expressed over the corresponding period (month or quarter in relation to the frequency of the data) of the previous year.

## Benefits of YOY:

YOY measurements facilitate the cross comparison of sets of data. For a company’s first-quarter revenue using YOY data, a financial analyst or an investor can compare years of first-quarter revenue data and quickly ascertain whether a company’s revenue is increasing or decreasing.

Pros of YOY:

* Minimizes the impact of seasonality on projections by comparing two static points in time
* Objectifies volatile values by comparing overall results
* Offers a simple calculation most people can do easily
* Results are formatted as percentages easy to understand, compare and draw quick insight from

Cons of YOY:

* The insights gained are improved by repeating the procedure over multiple points in time
* It's not valuable to provide insight into individual months where performance could be improved
* Is completely negated if one point of data expresses negative growth

Formula of YOY:

**Year-over-year Growth** = [(This Year – Last Year) / Last Year] X 100

**Compound annual growth (CAGR)**

The compound annual growth rate (CAGR) is the annualized average rate of revenue growth between two given years, assuming growth takes place at an exponentially compounded rate.

Benefits of CAGR:

The CAGR can be used to calculate the average growth of a single investment. As we saw in our example above, due to market volatility, the year-to-year growth of an investment will likely appear erratic and uneven.

Pros of CAGR:

* CAGR is an important tool which is used to measure the performance or profitability of any investment.
* As it considers the period of investment, it helps in providing a better understanding of the performance of the investment or the company in comparison to another.
* This helps in making accurate investment decisions which can result in higher profitability for the investor.

Cons of CAGR:

* CAGR does not consider market fluctuations or market volatility while evaluating the performance of any stocks. This is a hurdle in getting the true picture of the performance of such stock and can mislead the investor in making any investment decision.
* CAGR cannot be used as the sole basis to make investment decisions related to stock markets. Investors have to consider other parameters that incorporate the risk involved on account of market volatility and have to accordingly make their investment decisions.
* CAGR does not reflect on the short-term variations in the market and hence cannot be used as sole yardstick to make an investment decision between two funds or portfolios.

Formula of CAGR:

***CAGR*=**(*BVEV* )*n*1 −1×100

Where,

*EV*=Ending value

*BV*=Beginning value

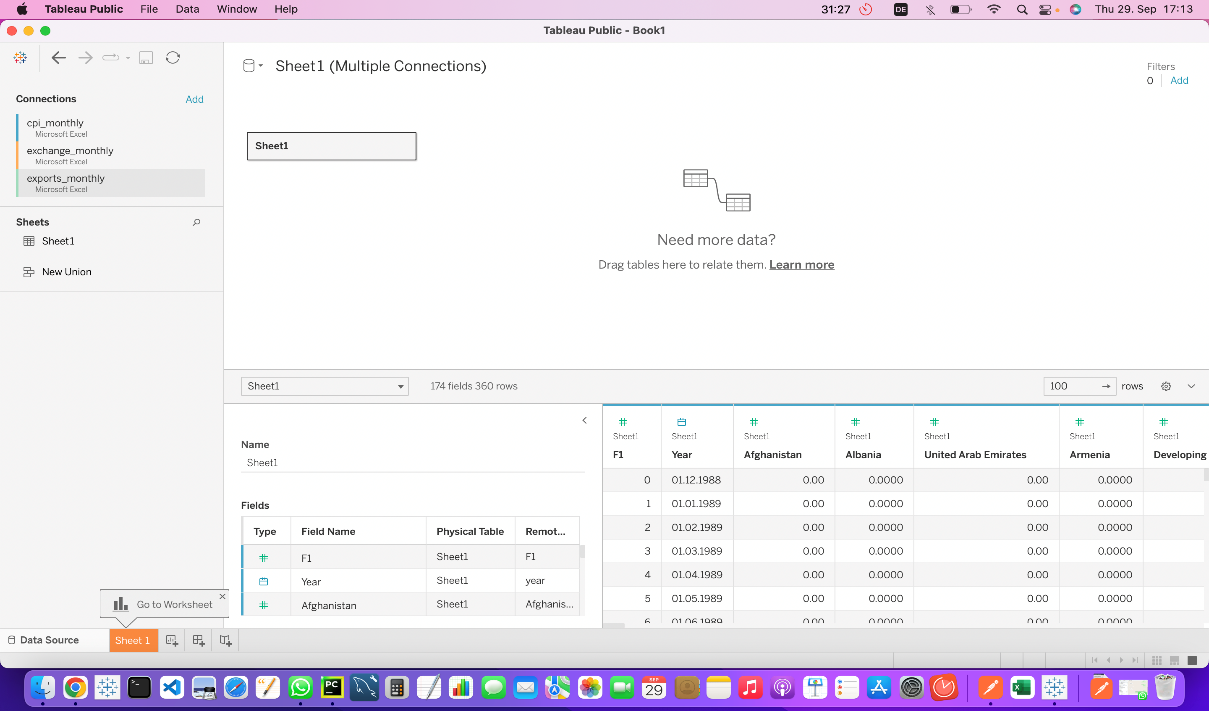
*n*=Number of years

**3.2. Data Transformation**

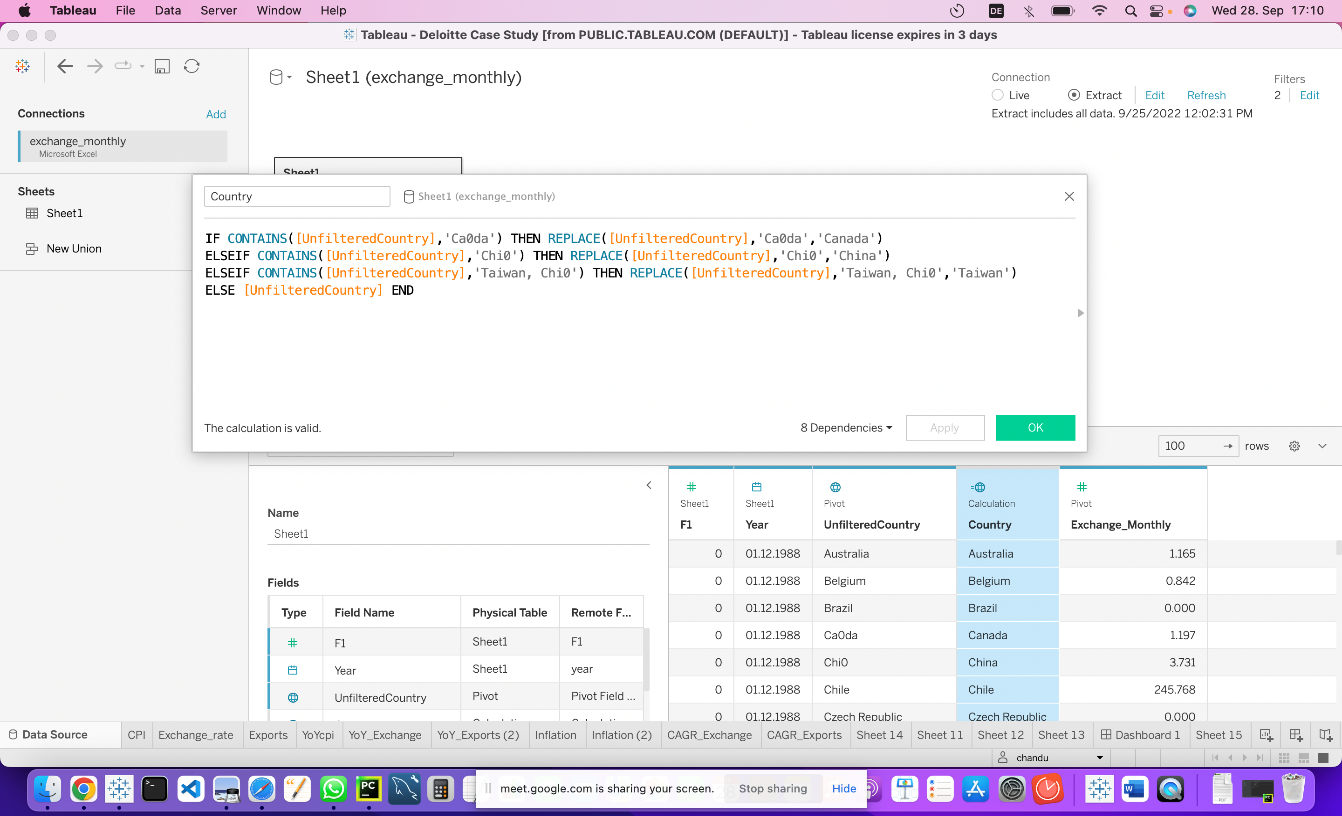
In the Transformation Process, we will convert our original datasets with other necessary attributes format. Originally datasets are in the form of wide dataset we converted into long datatype which will be useful for analysis Removing of NaN values and duplicate values. Conversion of monthly data into quarterly and yearly data. Filtering data according to countries

**3.3. Tableau Configuration**

**Step 1: Configuring Data Source**



**Step 2: Filtering the data**



**3.4 Deployment**

Once you’ve completed your dashboard, follow these steps:-

Server, Tableau Public, Save to Tableau Public As You may be prompted to log into your Tableau Public profile first if this is your first time publishing.

**4.Reference**

**1. Website URL :** <https://data-flair.training/blogs/tableau-architecture/>