# High Level Design (HLD)

# Wine Data Analysis

Harshit Soni

## Document Version Control

| Date Issued | Version | Description | Author |
|---|---|---|---|
| **11th July 2023** | 1.0 | First Version of Complete HLD | Harshit Soni |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# Contents

# Abstract

Data analysis is a process of inspecting, transforming, and monitoring to turn raw data into valuable insights. Data Insights helps in making the required decisions for the growth of business and company. To lead a data-driven approach for a business, it is important to analyze the data in deep. Various techniques for Data Analysis in Power BI will be interesting and beneficial for you to learn.

Power BI is no doubt an effective tool for analyzing data. This tool can help data analysts to keep an eye on the huge data generated by the company. Through this blog, we are now familiar with the ways for effective data visualization in Power BI. We have learned some basic techniques that you can use to generate valuable data insights.


# 1 Introduction

## 1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project •
- List and describe the non-functional attributes like:
    - Security ○ Reliability ○ Maintainability
    - ○

      Portability ○ Reusability ○

      Application compatibility ○

      Resource utilization ○ Serviceability

## 1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

# 2 General Description

## 2.1 Product Perspective & Problem Statement

Amazon Sales data has the biggest data records we need to simplify it and visualize it for a given scenario. The objective of the project is to perform data visualization techniques to

understand the insight of the data. This project aims apply various Business Intelligence tools such as Tableau or Power BI to get a visual understanding of the data.

## 2.2 Tools used

Business Intelligence tools and libraries works such as Excel, R, PowerBi , are used to build the whole framework. 3 Design Details

## 3.1 Functional Architecture

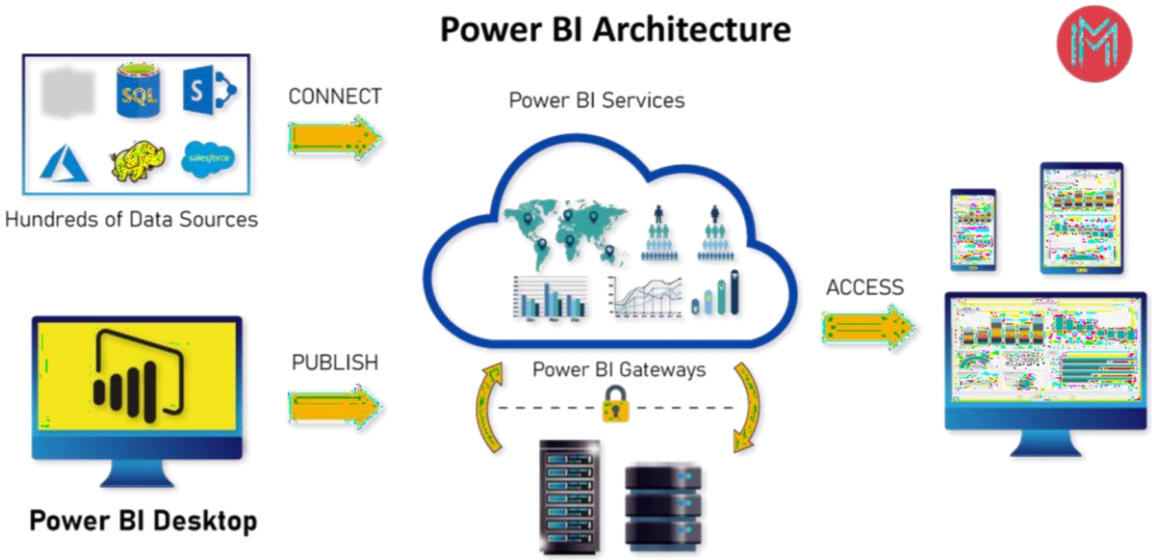
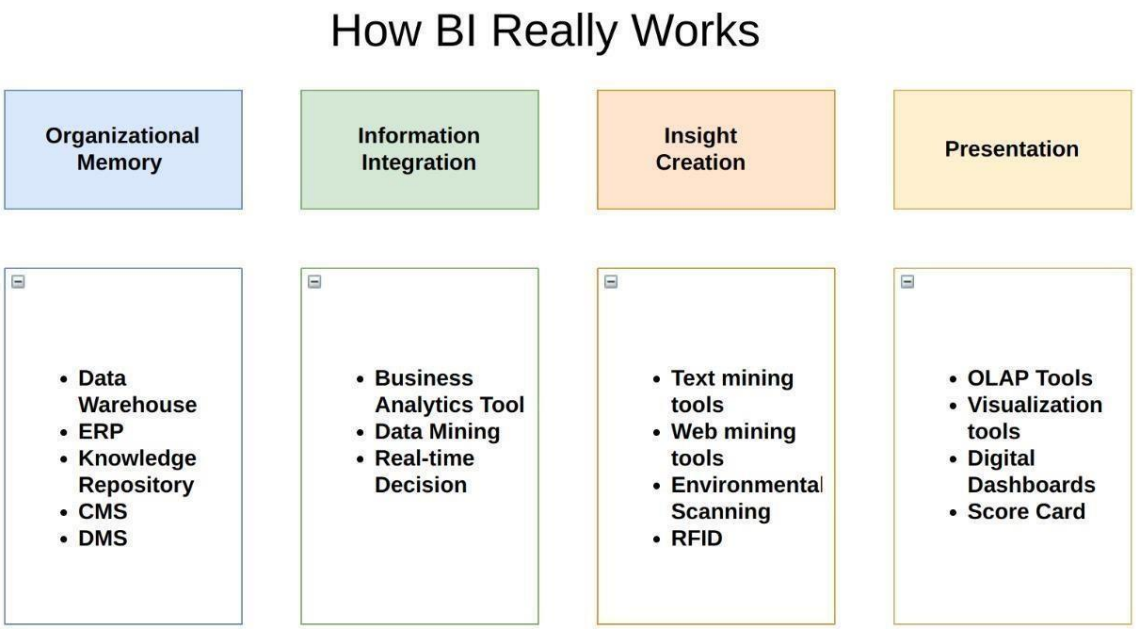



Figure 1: Functional Architecture of Business Intelligence



## 3.2 Optimization

**Your data strategy drives performance**

- Minimize the number of fields
- Minimize the number of records
- Optimize extracts to speed up future queries by materializing calculations, removing columns and the use of accelerated views

**Reduce the marks (data points) in your view**

- Practice guided analytics. There's no need to fit everything you plan to show in a single view. Compile related views and connect them with action filters to travel from overview to highly-granular views at the speed of thought.
- Remove unneeded dimensions from the detail shelf.
- Explore. Try displaying your data in different types of views. **Limit your filters by number and type**

- Reduce the number of filters in use. Excessive filters on a view will create a more complex query, which takes longer to return results. Double-check your filters and remove any that aren't necessary.
- Use an include filter. Exclude filters load the entire domain of a dimension, while include filters do not. An include filter runs much faster than an exclude filter, especially for dimensions with many members.
- Use a continuous date filter. Continuous date filters (relative and range-of-date filters) can take advantage of the indexing properties in your database and are faster than discrete date filters.
- Use Boolean or numeric filters. Computers process integers and Booleans (t/f) much faster than strings.
- Use parameters and action filters. These reduce the query load (and work across data sources).

**Optimize and materialize your calculations**

- Perform calculations in the database
- Reduce the number of nested calculations.
- Reduce the granularity of LOD or table calculations in the view. The more granular the calculation, the longer it takes.
    - LODs - Look at the number of unique dimension members in the calculation.
    - Table Calculations - the more marks in the view, the longer it will take to calculate.
- Where possible, use MIN or MAX instead of AVG. AVG requires more processing than MIN or MAX. Often rows will be duplicated and display the same result with MIN, MAX, or AVG.

- Make groups with calculations. Like include filters, calculated groups load only named members of the domain, whereas Tableau's group function loads the entire domain.
- Use Booleans or numeric calculations instead of string calculations. Computers can process integers and Booleans (t/f) much faster than strings.
  Boolean>Int>Float>Date>DateTime>String

# 4 KPIs

Dashboards will be implemented to display and indicate certain KPIs and relevant indicators for the sales .



As and when, the system starts to capture the historical/periodic data for a user, the dashboards will be included to display charts over time with progress on various indicators or factors .

## 4.1 KPIs (Key Performance Indicators)

Key indicators displaying a summary of the Sales insights and its relationship with different metrics

- Percentage of Leads in Each Lifecycle Stage
- MQL-to-Customer Conversion Rate
- Average Length of Customer Lifecycle
- Volume of New Opportunities
- Cost Per Lead, Cost Per Acquisition
- Customer Retention Rat
- Average Revenue Per Account
- Net Promoter Score (NPS)
- Customer Lifetime Value (CLV))

# 5 Deployment

The deployment process lets you clone content from one stage in the pipeline to another, typically from development to test, and from test to production.

During deployment, Power BI copies the content from the current stage, into the target one. The connections between the copied items are kept during the copy process. Power BI also applies the configured deployment rules to the updated content in the target stage. Deploying content may take a while, depending on the number of items being deployed. During this time, you can navigate to other pages in the Power BI portal, but you can't use the content in the target stage. With on-premises, cloud, and hosted options, there is a version of powerbi to match your requirements.

TYPE PROS CONS

PROS-

- Affordability
- Custom Visualizations
- Excel Integration
- Data Connectivity
- Prompt Updates
- Power Bi Embedded
- Personal Gateway
- Data Accessibility
- Interactive Visulizations.

CONS-

- Table Relationships.
- Configuration of Visuals
- Crowded UI
- Rigid Formulas
- Handling large data volumes
- Complex to understand and master.