# High-Level Design (HLD)

# Budget Sales Analysis

Revision Number : 1.1

Last Date of Revision: 17/06/2022

*Indrajeet Thakare*

## DOCUMENT VERSION CONTROL

| Date Issued | Version | Description | Author |
|---|---|---|---|
| 17-04-2022 | 1.0 | Abstract, Introduction | Indrajeet Thakare |
| 18-04-2022 | 1.1 | Design detail, KPI, deployment | Indrajeet Thakare |
| 17-06-2022 | 1.2 | Final Revision | Indrajeet Thakare |

# Contents



****_____****_____*****_____******_____****____****

## ABSTRACT

Today people are more aware of the things they are supposed to do in order to live a healthy life. Lately people have got to know the health benefits of riding bike in there day to day life. We will be analyzing the sales data of a company which major focused on cycle as their main category. To achieve the goal, we used a data set that is formed by taking into consideration some information of customer details, Date specific details and country specific sales data etc. we have to Extract various information such as Sales, budget, variance.

# 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 before coding and can be used as a reference manual for how the modules interact at a high level.

**The HLD will:**

• Present all 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

The goal of this project is to analyze to forecast future sales occurrence, based on a combination of features that describes the Sales. To achieve the goal, we used a data set that is formed by taking into consideration some information of customer details, Date specific details and country specific sales data etc. we have to Extract various information such as Sales, budget, variance.

### 2.2 Tools used

Business Intelligence tools and libraries works such as NumPy, Pandas, Seaborn, Matplotlib, MS-Excel, MS-Power BI, Jupyter Notebook and Python Programming Language are used to build the whole framework.






# 3. Design Details

## 3.1 Functional Architecture

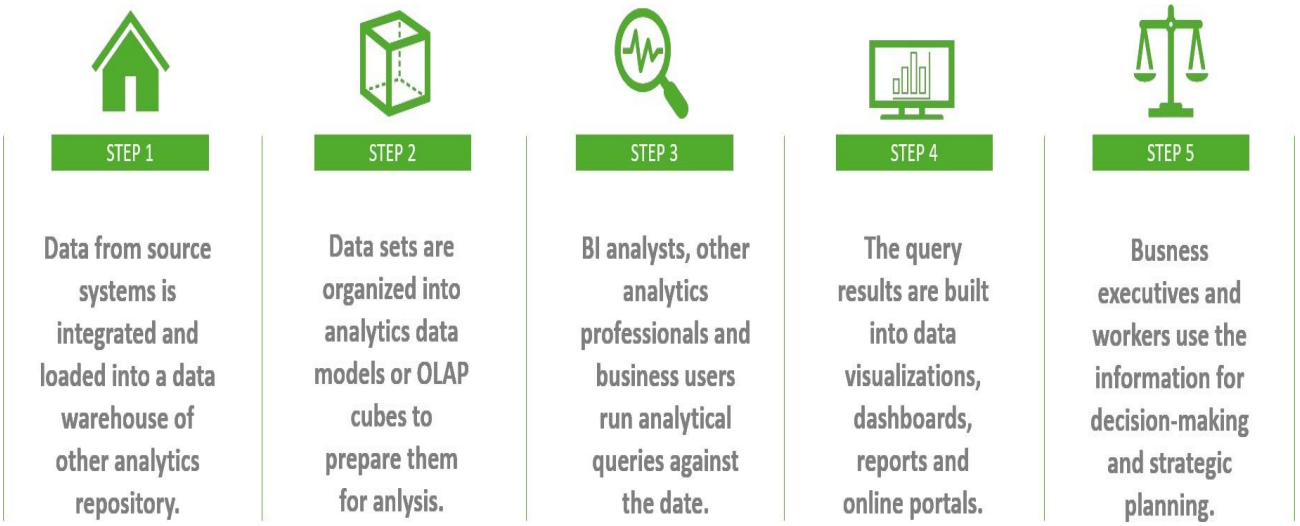
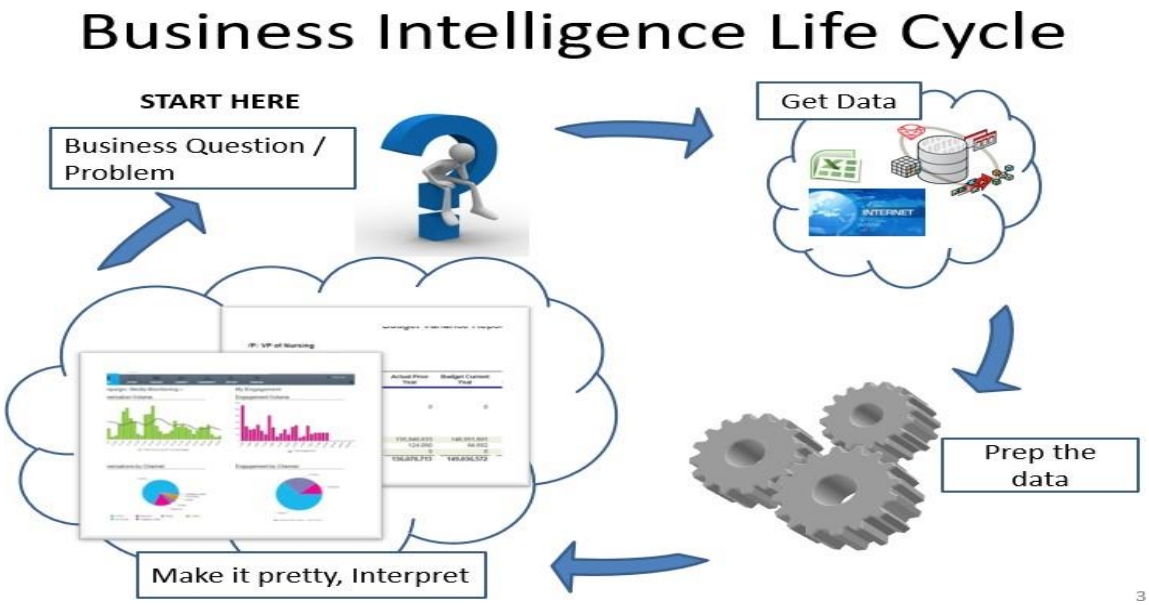



Figure 1: Functional Architecture of Business Intelligence




## 3.2 Optimization

### A. 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.

### B. 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.

### C. 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 included filter. Exclude filters load the entire domain of a dimension, while including filters do not. An include filter runs much faster than an excluded 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 data filters.
- Use Boolean or numeric filters. Computers process integers and Boolean (t/f) much faster than strings.
- Use parameters and action filters. These reduce the query load (and work across data sources)

**D. 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 Boolean or numeric calculations instead of string calculations.

Computers can process integers and Boolean (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 Housing Price and its relationship with different metrics.

1. Target Audience
2. Gender Distribution
3. Average order value per region
4. Sales per region
5. Top 10 product by sales
6. SubCategory based sales data
7. Sales based on weekend and weekdays
8. Sales based of yearly income of customers
9. Sales based on occupation and marital status

## 5. Deployment

Prioritizing data and analytics couldn’t come at a better time. Your company, no matter what size, is already collecting data and most likely analyzing just a

portion of it to solve business problems, gains competitive advantages, and drive enterprise transformation. With the explosive growth of enterprise data, database technologies, and the high demand for analytical skills, today's most effective IT organizations have shifted their focus to enabling self-service by deploying and operating Power BI at scale, as well as organizing, orchestrating, and unifying disparate sources of data for business users and experts alike to author and consume content.

Power BI prioritizes choice in flexibility to fit, rather than dictate, your enterprise architecture. Power BI Desktop and Power BI Service leverage your existing technology investments and integrate them into your IT infrastructure to provide a self-service, modern analytics platform for your users. With on-premises, cloud, and hosted options, there is a version of Power BI to match your requirements.