High Level Design (HLD)

Analyze International Debt Statistics

Revision Number: 1.0

Last date of revision: 19/08/2023

Kushal Chaple

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| **19 th June 2023** | 1.0 | First Version of Complete HLD | Kushal Chaple |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Contents**

[Document Version Control 2](#_bookmark0)

[Abstract 3](#_bookmark1)

1. [Introduction 5](#_bookmark2)
   1. [Why this High-Level Design Document? 5](#_bookmark3)
   2. [Scope 5](#_bookmark4)
2. [General Description 6](#_bookmark5)
   1. [Product Perspective & Problem Statement 6](#_bookmark6)
   2. [Tools used 6](#_bookmark8)
3. [Design Details 7](#_bookmark9)
   1. [Functional Architecture 7](#_bookmark10)
   2. [Optimization 8](#_bookmark11)
4. [KPIs 9](#_bookmark12)
   1. [KPIs (Key Performance Indicators) 9](#_bookmark13)
5. [Deployment 9](#_bookmark14)

# Abstract

It’s not that we humans only take debts to manage our necessities. A country may also take debt to manage its economy. For example, infrastructure spending is one costly ingredient required for a country’s citizens to lead comfortable lives. [The World Bank](https://www.worldbank.org/) is the organization that provides debt to countries.

In this notebook, we are going to analyze international debt data collected by The World Bank. The data set contains information about the amount of debt (in USD) owed by developing countries across several categories. This project is going to use partial data set to explore the debt statistics of developing countries and further analyze their economic conditions and their major loan purpose. This project is entirely conducted in Jupyter Notebook using SQL.

# Introduction

## 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

## 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.

# General Description

## Product Perspective & Problem Statement

The World Bank provides Debt to the International countries to maintain their economy. The World Bank List the yearly debt data and provides the Debt Statistics on The World Bank site.

We are going to find the answers to the questions like:

* What is the total amount of debt that is owed by the countries listed in the dataset?
* Which country owns the maximum amount of debt and what does the amount look like?
* What is the average amount of dent owed by countries across different debt indicators?

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.

## Tools used

Business Intelligence tools works such as Excel, Power Query Tableau, Power BI are used to build the whole framework.



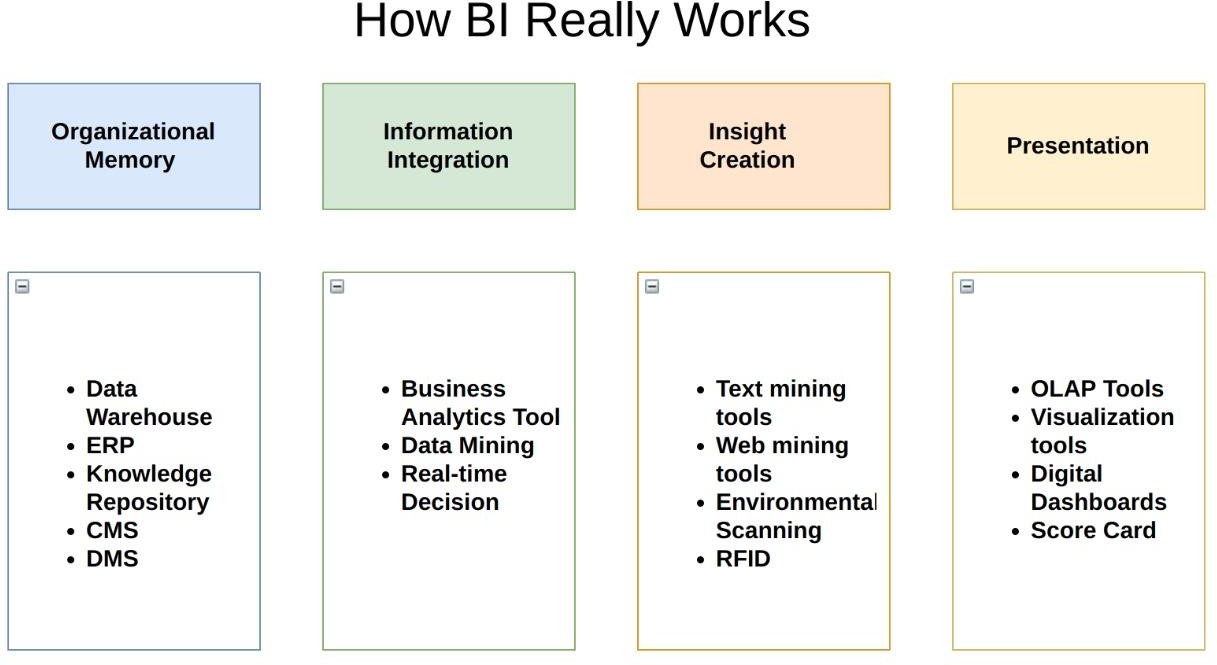


# Design Details

## Functional Architecture



Figure 1: Functional Architecture of Business Intelligence



## 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](http://onlinehelp.tableau.com/current/pro/online/mac/en-us/help.htm#filtering_add_dragfields_dates.html). 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](http://www.tableau.com/learn/tutorials/on-demand/logical-calculations). Computers process integers and Booleans (t/f) much faster than strings.
    - Use [parameters](http://onlinehelp.tableau.com/current/pro/online/en-us/help.htm#parameters.html) and [action filters](http://onlinehelp.tableau.com/current/pro/online/en-us/help.htm#actions.html). 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](http://onlinehelp.tableau.com/current/pro/online/windows/en-us/help.htm#calculations_aggregation.html). 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](http://kb.tableau.com/articles/knowledgebase/creating-groups-using-calculated-fields). 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](http://onlinehelp.tableau.com/current/pro/online/mac/en-us/help.htm#functions_functions_string.html). Computers can process integers and Booleans (t/f) much faster than strings. Boolean>Int>Float>Date>DateTime>String

# KPIs

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



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

## KPIs (Key Performance Indicators)

Key indicators displaying a summary of the International Debt Statistics and its relationship with different metrics

1. Total amount of debt by all countries.
2. Total Debt count.
3. Total Number of Indicators.
4. Total Number of Countries.
5. Countries with highest Debt amount.
6. Countries with lowest Debt amount.
7. Highest amount of Principal Repayment.

# 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, gain 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 at scale, as well as organizing, orchestrating, and unifying disparate sources of data for business users and experts alike to author and consume content.

When content from the current stage is copied to the target stage, Power BI identifies existing content in the target stage and overwrites it. To identify which content item needs to be overwritten, deployment pipelines uses the connection between the parent item and its clones. This connection is kept when new content is created. The overwrite operation only overwrites the content of the item. The item's ID, URL, and permissions remain unchanged.

In the target stage, [item properties that aren't copied](https://learn.microsoft.com/en-us/power-bi/create-reports/deployment-pipelines-process#item-properties-that-are-not-copied), remain as they were before deployment. New content and new items are copied from the current stage to the target stage.

**5.1 Deployed items**

When you deploy content from one pipeline stage to another, the copied content can contain the following Power BI items:

* Datasets
* Reports
* Dataflows
* Datamarts
* Dashboards
* Paginated reports

**5.2 Unsupported items**

* Deployment pipelines doesn't support the following items:
* Datasets that don't originate from a .pbix
* PUSH datasets
* Streaming dataflows
* Reports based on unsupported datasets
* Template app workspaces
* Workbooks
* Metrics

**5.3 Item properties copied during deployment**

During deployment, the following item properties are copied and overwrite the item properties at the target stage:

* Data sources (deployment rules are supported)
* Parameters (deployment rules are supported)
* Report visuals
* Report pages
* Dashboard tiles
* Model metadata
* Itemrelationships