High Level Design (HLD)

Analyze International Debt Statistics

Revision Number: 1.0
Last date of revision: 23/05/2024

Rohith Obillaneni

Document Version Control

Date Issued	Version	Description	Author
23-05024	1.0	First Version of Complete HLD	Rohith Obillaneni

Contents

Abstract

Countries often take on debt to support economic stability and fund critical infrastructure projects. The World Bank provides extensive data on the debt of developing nations. This project aims to analyse this data using Power BI, providing visual insights into the debt profiles of various countries, identifying trends, and understanding key financial indicators. The World Bank is the organization that provides debt to countries.

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

The World Bank provides debt data for international countries, aiding in economic management. This project will address questions such as:

- What is the total debt owed by the countries listed in the dataset?
- Which country has the highest debt, and what is the amount?
- What is the average debt across different indicators?

The objective is to leverage Power BI for data visualization, enabling a detailed understanding of debt statistics and economic conditions of developing countries.

2.2 Tools used

The project utilizes various Business Intelligence tools, including:

- **Power BI**: For data visualization and analysis.
- **Microsoft Excel**: For initial data manipulation and cleanup.
- **SQL**: For data querying and manipulation.

3 Design Details

3.1 Functional Architecture

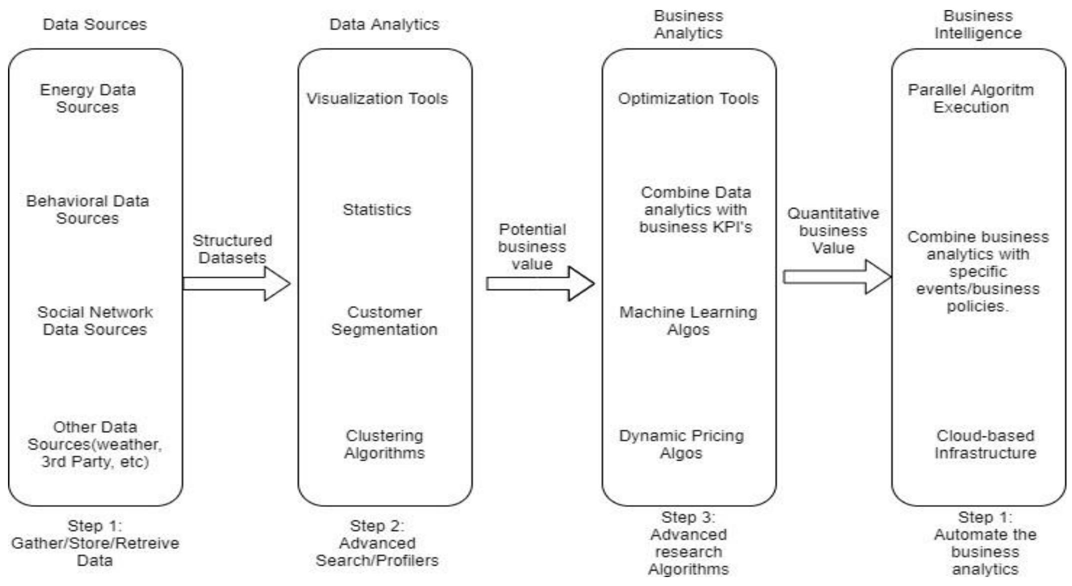
The functional architecture of this project consists of several key stages:

1. **Data Acquisition**: Download the dataset from The World Bank.

2. **Data Cleaning**: Clean and preprocess the data using Power BI.
3. **Data Transformation**: Use DAX in Power BI to create necessary measures and calculated columns.
4. **Data Visualization**: Develop interactive dashboards and reports in Power BI.

Diagram: Functional Architecture

1. **Data Source**: The World Bank dataset.
2. **Data Processing**: Power BI for cleaning and transformation.
3. **Data Storage**: Power BI service.
4. **Data Visualization**: Dashboards and reports in Power BI.
5. **User Interaction**: End-users interact with the Power BI dashboards for insights.


Data Sources
Energy Data Sources
Behavioral Data Sources
Social Network Data Sources
Other Data Sources(weather, 3rd Party, etc)
Step 1: Gather/Store/Retreive Data
Structured Datasets
Data Analytics
Visualization Tools
Statistics
Customer Segmentation
Clustering Algorithms
Step 2: Advanced Search/Profilers
Potential business value
Business Analytics
Optimization Tools
Combine Data analytics with business KPI's
Machine Learning Algos
Dynamic Pricing Algos
Step 3: Advanced research Algorithms
Quantitative business Value
Business Intelligence
Parallel Algoritm Execution
Combine business analytics with specific events/business policies.
Cloud-based Infrastructure
Step 1: Automate the business analytics

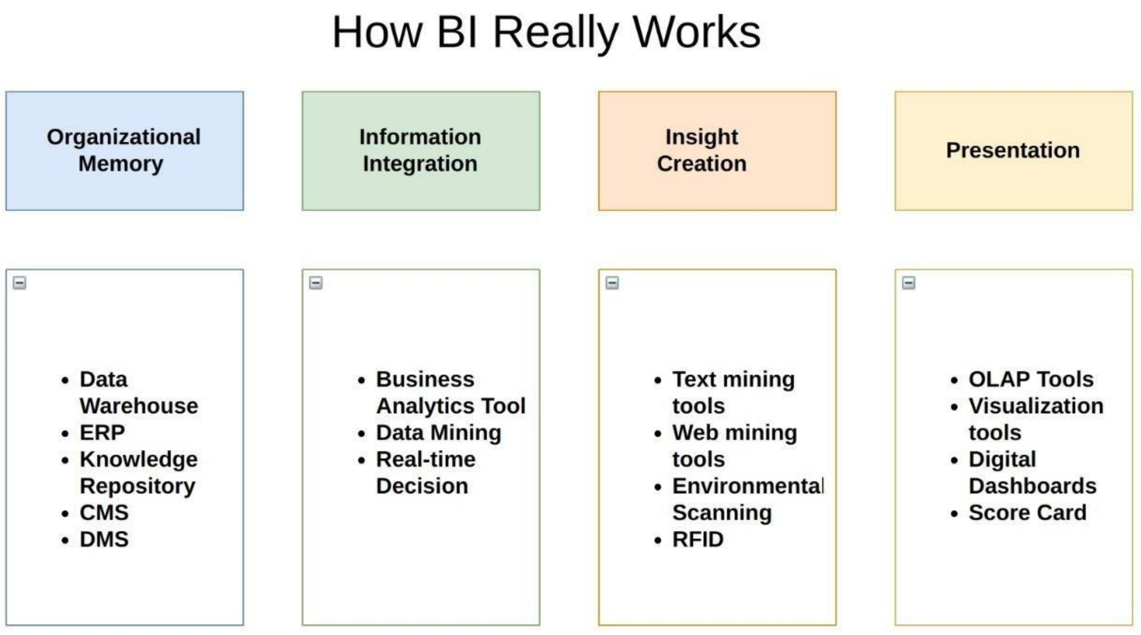


Figure 1: Functional Architecture of Business Intelligence

3.2 Optimization

To ensure optimal performance in Power BI:

- **Minimize Fields and Records:** Reduce the dataset size by removing unnecessary fields and records.
- **Optimize Data Extracts:** Materialize calculations, remove unused columns, and use accelerated views for faster queries.
- **Reduce Marks (Data Points):** Simplify views by limiting the number of data points. Use action filters to connect related views, reducing complexity.
- **Limit Filters:** Use fewer and more efficient filters. Continuous date filters and Boolean/numeric filters are preferable for performance.
- **Efficient Calculations:** Perform calculations within the database where possible. Reduce nested calculations and granularity of Level of Detail (LOD) or table calculations.
- **Simplify Aggregations:** Use MIN or MAX instead of AVG to reduce processing time.

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 AV
- Make groups with calculations. Like include filters, calculated groups load only named members of the domain, whereas Tableau's group function loads the entiredomain.
- 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 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

4.1 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.

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, 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 use 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, 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
- Item relationship