

Low Level Design

“Amazon Data Sales Analysis”

Revision No.: 1

Sachin Solanki

Abstract

The Low-Level Design (LLD) document outlines the detailed technical specifications and architecture for implementing specific functionalities within the Amazon Sales Data Analysis project using Power BI. It delves into the intricacies of data loading, processing, visualization, and deployment aspects of the project. The LLD provides a granular view of the system design, including data sources, Power BI components, and integration strategies. By detailing the data structures, transformations, and visualization techniques, the LLD serves as a blueprint for developers to execute the project efficiently and effectively.

Table of Contents

Low-Level Design.....	1
Abstract.....	2
Table of Contents	3
Chapter 1 Introduction.....	4
1. Introduction.....	4
1.1 What is Low-Level design document?	4
1.2 Scope.....	4
Chapter 2 Architecture.....	5
2.1 Data Sources	5
2.2 Power BI Desktop	6
2.3 Power BI Service	6
2.4 Power BI Report Server	6
2.5 Power BI Gateway	7
2.6 Power BI Mobile.....	7
2.7 Power BI Embedded	7
Chapter 3 Architecture Description.....	8
3.1 Data Description	8
3.2 Data Loading.....	9
3.3 Data Transformation	9
3.4 Data Modelling	10
3.5 Deployment.....	10
Chapter 4 Use Cases	13

Chapter 1 Introduction

1. Introduction

The Low-Level Design (LLD) document elaborates on the intricate technical aspects of the system architecture and provides specific design specifications tailored to each component. This document serves as a detailed guide for developers, offering insights into the implementation phase of the project while maintaining alignment with the overarching goals and requirements.

1.1 What is Low-Level design document?

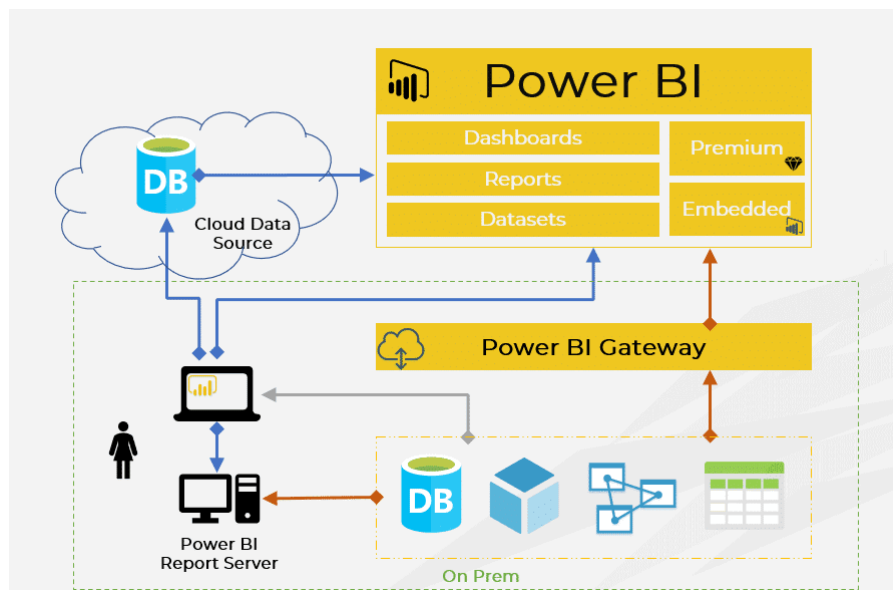
In the context of our project, the Low-Level Design (LLD) document serves as a comprehensive technical blueprint that expands upon the high-level architecture delineated in the High-Level Design (HLD) document. It delves into the granular details of each system component, including modules, classes, functions, and interfaces, elucidating the implementation specifics required to realize the system's functionality effectively.

1.2 Scope

The scope of the Low-Level Design (LLD) document extends to the meticulous design of individual components, modules, and functions within the system architecture. It defines the precise boundaries and responsibilities of each element, elucidating their interrelationships and collaborative mechanisms to achieve the desired system behavior. Furthermore, the LLD specifies the programming languages, frameworks, and technologies to be employed during implementation, ensuring consistency, reliability, and scalability across the entire system.

Chapter 2 Architecture

The architecture of the system defines the structure and arrangement of its components, outlining how they interact and function together to fulfill the system's objectives.



2.1 Data Sources

Data sources refer to the origins of the data used within the system. They could include databases, files, web services, or other data repositories from which data is extracted, transformed, and loaded into the system for analysis and reporting.

The data sources used in the project may include various types, such as:

- 1. Relational Databases: Data sourced from relational database management systems (RDBMS) like MySQL, PostgreSQL, SQL Server, or Oracle.
- 2. Flat Files: Data stored in flat file formats such as CSV (Comma-Separated Values), Excel spreadsheets, or text files.
- 3. Cloud-based Storage: Data retrieved from cloud storage services like Amazon S3, Google Cloud Storage, or Microsoft Azure Blob Storage.
- 4. Web APIs: Data obtained from web APIs (Application Programming Interfaces) provided by external services, platforms, or applications.
- 5. Streaming Data: Real-time data streams sourced from IoT devices, sensors, social

LOW LEVEL DESIGN (LLD)

media platforms, or other sources.

- 6. On-Premises Systems: Data extracted from on-premises systems within the organization's infrastructure, such as legacy systems or proprietary databases.
- 7. Business Applications: Data exported from enterprise resource planning (ERP) systems, customer relationship management (CRM) software, or other business applications.
- 8. External Data Providers: Data acquired from third-party data providers, market research firms, or government agencies.

The specific types of data sources utilized in the project would depend on the nature of the analysis, the requirements of the business, and the availability of data relevant to the project objectives.

2.2 Power BI Desktop

Power BI Desktop is a desktop application used for creating and designing interactive reports, dashboards, and data visualizations. It provides a comprehensive set of tools for data preparation, modeling, and visualization, empowering users to analyze and gain insights from their data.

2.3 Power BI Service

Power BI Service, also known as Power BI Online, is a cloud-based platform provided by Microsoft for publishing, sharing, and collaborating on Power BI reports and dashboards. It allows users to access their reports from anywhere, on any device, and enables real-time collaboration and data-driven decision-making.

2.4 Power BI Report Server

Power BI Report Server is an on-premises report server provided by Microsoft, allowing organizations to host and manage Power BI reports and dashboards within their own environment. It provides similar capabilities to the Power BI Service but is deployed within the organization's infrastructure for enhanced security and control.

2.5 Power BI Gateway

Power BI Gateway is a bridge that facilitates secure data transfer between on-premises data sources and the Power BI Service or Power BI Report Server. It ensures that data refreshes, scheduled updates, and live connections to on-premises data sources are executed seamlessly and securely.

2.6 Power BI Mobile

Power BI Mobile is a mobile application that enables users to access and interact with Power BI reports and dashboards on their mobile devices, such as smartphones and tablets. It allows users to stay informed and make data-driven decisions while on the go, providing access to insights anytime, anywhere.

2.7 Power BI Embedded

Power BI Embedded is a platform-as-a-service (PaaS) offering that allows developers to integrate Power BI capabilities, such as reports and dashboards, directly into their own applications. It enables organizations to create customized, white-labeled analytics solutions for their customers or internal users, embedding rich data visualizations seamlessly into their applications.

Chapter 3 Architecture Description

3.1 Data Description

The dataset comprises two primary tables: Customer and Sales data from Amazon spanning the years 2017, 2018, and 2019 and 2020.

Amazon Sales Data Features:

- 1. Custkey: Unique identifier for customers.
- 2. Datekey: Transaction date.
- 3. Discount amount: Variance between sales amount (list price-based) and actual sales amount.
- 4. Invoice Date: Date of delivery and invoice generation.
- 5. Invoice Number: System-generated unique invoice identifier.
- 6. Item Class: Classification of the item.
- 7. Item Number: Unique identifier for items.
- 8. Item: Name of the item involved in the transaction.
- 9. Line Number: Order line number.
- 10. List Price: Manufacturer's quoted price.
- 11. Order Number: Unique order identifier.
- 12. Promised delivery date: Expected delivery date.
- 13. Sales Amount: Total sales value (price multiplied by quantity).
- 14. Sales amount based on List Price: Sales value based on list price.
- 15. Sales Cost amount: Cost incurred for item sales.
- 16. Sales Margin amount: Difference between sales amount and cost amount.
- 17. Sales Price: The selling price of the item.
- 18. Sales Quantity: Quantity of items ordered.
- 19. Sales Rep.: Unique identifier for sales representatives.
- 20. U/M: Unit of measurement for the item.

Customer Details Data Features:

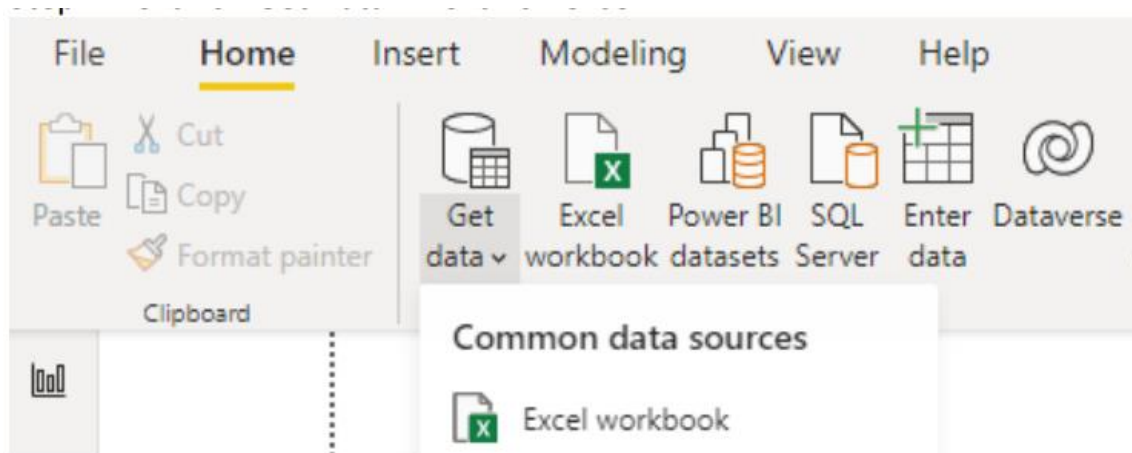
LOW LEVEL DESIGN (LLD)

- 1. Custkey: Unique customer identifier.
- 2. Region Name: Geographic region (e.g., central, southern).
- 3. Division Name: Domestic or international division.
- 4. City: Name of the city.
- 5. Country: Name of the country.
- 6. Zip Code: City's ZIP code.
- 7. Customer Name: Name of the customer.

3.2 Data Loading

Step 1:- Open Power BI Desktop Application

Step 2:- Click on Get Data ◇ Click on excel



Step 3 :- Browse to the data file on your system and select

Step 4: - Once Data is loaded click on Transform data in the bottom for further transformation.

3.3 Data Transformation

In the Transformation Process, we will convert our original datasets with other necessary attributes format and change the features according to the problem statement on Power BI ETL tool Power Query as the data is in excel format. A new Dim_date table created for easy time intelligence analysis the features are:

1. Day of Week

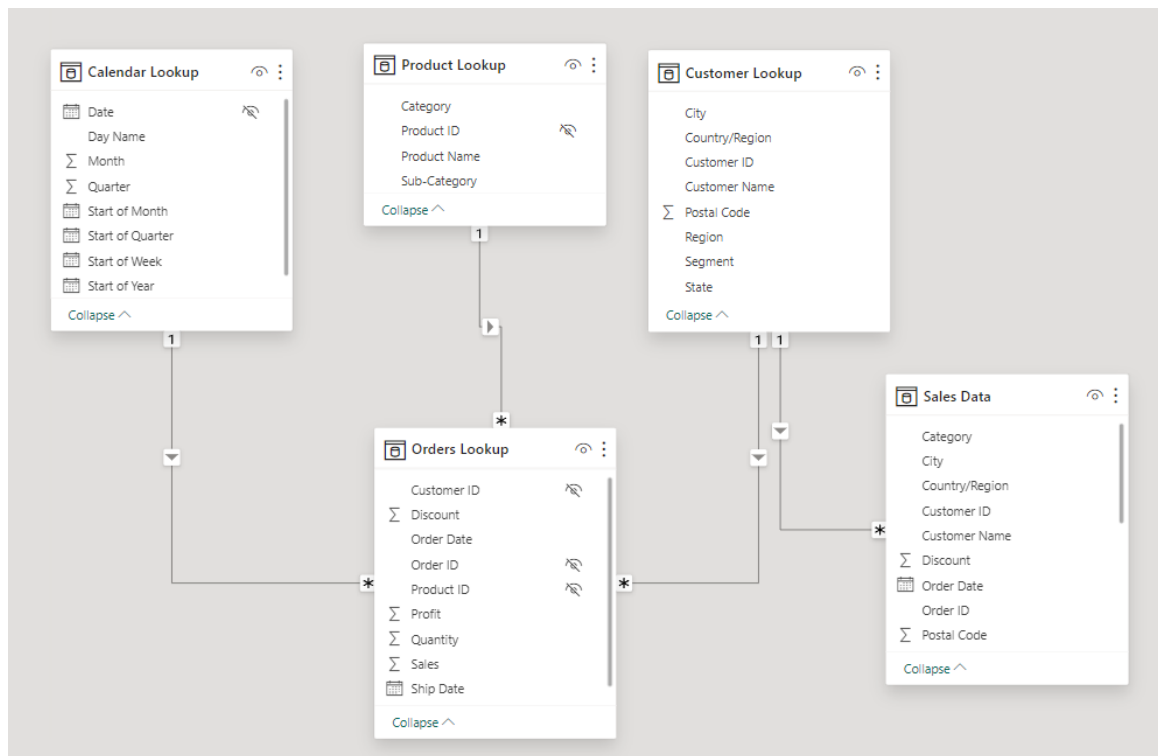
LOW LEVEL DESIGN (LLD)

2. Month name
3. Year
4. Year quarter
5. Month
6. Date

3.4 Data Modelling

In Power BI, you create a data model that defines the relationships between different tables imported from various data sources. This step is crucial for accurate analysis and visualization.

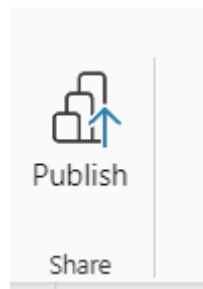
For this project:



3.5 Deployment

Final steps include deployment. In PowerBI we could publish our report..

Step 1: Find publish option in Home Tab



Step 2: Select the workspace

Publish to Power BI



Select a destination

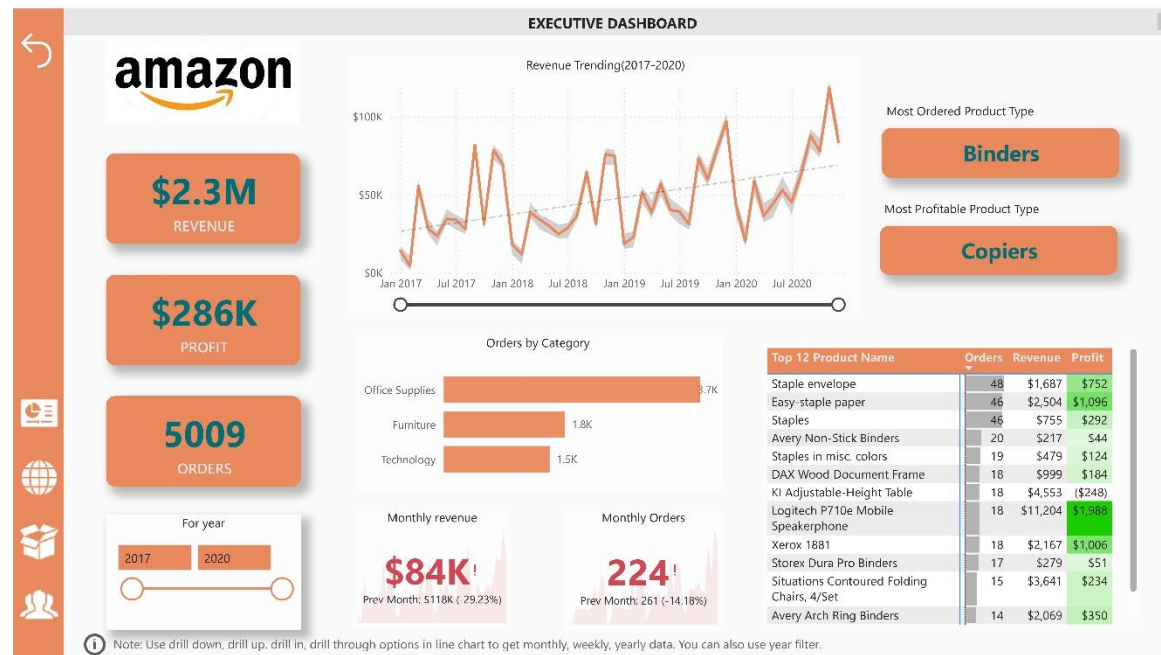
Search

My workspace

Select

Cancel

Step 3: The report is now published



Chapter 4 Unit Test Cases

Visual/Image	Test Cases	Purpose	Assertions
Executive Dashboard for Amazon	- Total sales calculation - Profit calculation - Profit margin calculation - Year-over-year (YoY) total sales calculation - YoY percentage change calculation	Verify accuracy of calculations	Compare results with manual calculations
Product Analysis Dashboard for Staples	- Total orders calculation - Revenue calculation - Profit targets assessment	Validate functionality	Cross-check with manual calculations
Customer Analysis Dashboard	- Total customers calculation - Revenue per customer calculation - Customer behavior insights	Ensure correctness	Compare results with expected values
AI Dashboard with Map	- Geographical data representation - Interactive filtering	Confirm data accuracy	Validate visual behavior

For DAX measures, consider the following unit test cases:

Essential Functionality Tests:

- Total sales calculation
- Profit calculation
- Profit margin calculation
- YoY total sales calculation
- YoY percentage change calculation

Boundary Case Tests:

- Check for empty or zero sales amounts

LOW LEVEL DESIGN (LLD)

- Ensure profit amounts are not negative

Edge Scenario Tests:

- Test missing data (e.g., missing sales amounts)
- Validate year-over-year calculations

Assertions:

- Use DAX queries (e.g., COUNTROWS, CONTAINS) or the built-in data view for assertions.