High Level Design (HLD) Amazon Sales Data Analysis

Revision Number: 1.0

Last date of revision: 15/07/2023

Harshit Kumar Pathak

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| **15th July 2023** | 1.0 | First Version of Complete HLD | Harshit Kumar Pathak |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Contents**

[Document Version Control 2](#_bookmark0)

[Abstract 4](#_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

This study analyzes Amazon's sales data from 2010 to 2017. The data includes yearly and monthly sales figures for revenue, cost, profit, units sold, item types, and region. The Amazon Sales report describe the sales of the various range of products in International countries. This report may help to take necessary steps after seeing the key insights and analysis. The Detailed analysis of the Sales Data of Amazon for the year of 2010 to Year 2017 is done by using Python and Streamlit tool

which is capable of showing the key insights.

# 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.

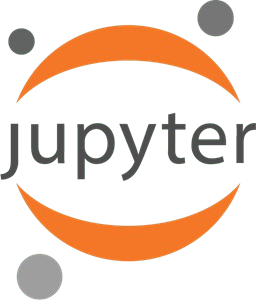
# General Description

## Product Perspective & Problem Statement

Sales management has gained importance to meet increasing competition and the need for improved methods of distribution to reduce cost and to increase profits. Sales management today is the most important function in a commercial and business enterprise.

## Tools used

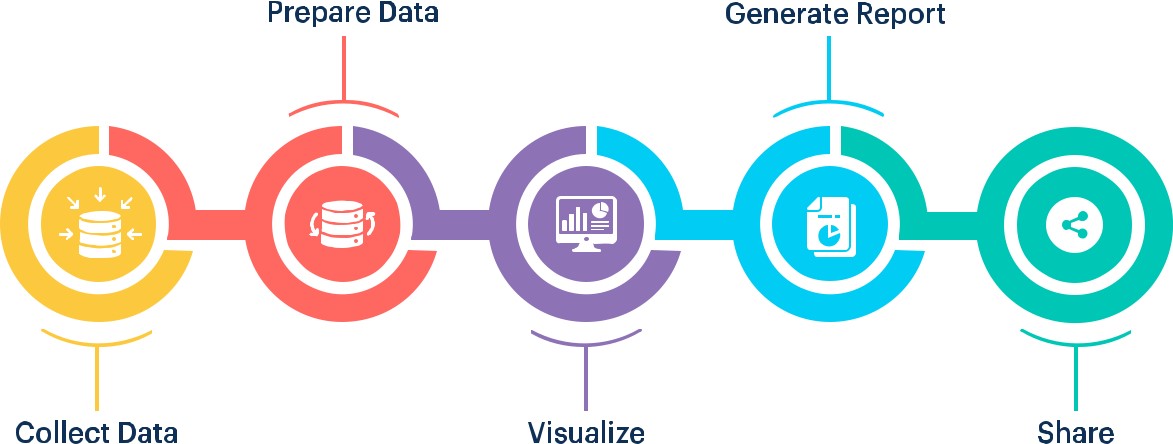
Business Intelligence tools and libraries works such as Jupyter Notebook, Seaborn, Pandas, Plotly & Streamlit are used to build the whole framework



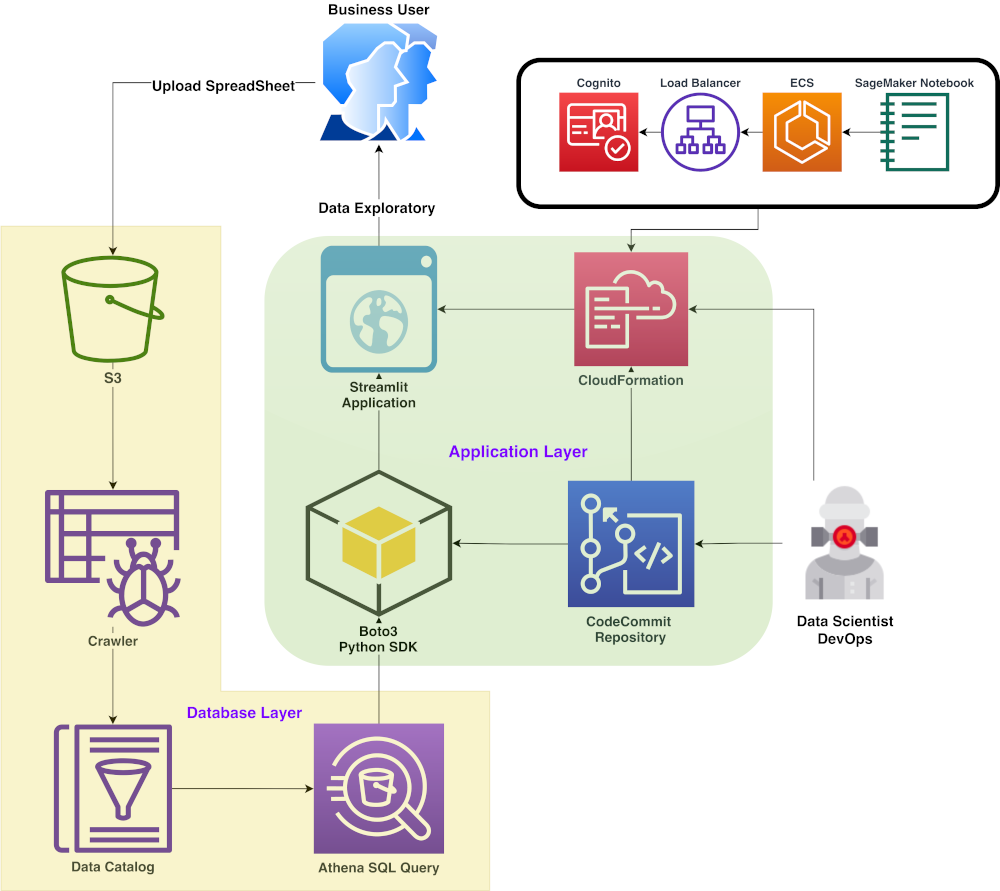
  

# Design Details

## Functional Architecture



**Figure 1: Functional Architecture of Business Intelligence**



**Figure 2: Functional Architecture of Streamlit**

## 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 data 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](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 Amazon sales, 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 Amazon Sales Data and its relationship with different metrics

1. Total Revenue, Total Profit, Total Cost and Total Units Sold
2. Total Revenue, Profit, Cost, Units Sold over the Years
3. Total Revenue, Profit, Cost, Units Sold over months
4. Sales Channels Yearly
5. Product category wise sales
6. Region Wise Sales and so on.

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

Streamlit Python prioritizes choice in flexibility to fit, rather than dictate, your enterprise architecture. Streamlit Cloud and Streamlit Sharing leverage your existing technology investments and integrate into your IT infrastructure to provide a self-service, modern analytics platform for your users. With cloud and hosted options, there is a version of Streamlit Python to match your requirements. Below is a comparison of the two types:

* + - Streamlit Cloud is a fully managed service that takes care of all the infrastructure and operational overhead. This is the easiest way to get started with Streamlit Python.
    - Streamlit Sharing gives you more control over the deployment process. You can host your Streamlit Python app on your own infrastructure or on a third-party cloud platform.

Depending on your organizational roles and responsibilities, Streamlit Cloud should be set up by a systems administrator and the designated Streamlit Cloud Administrator in coordination with the appropriate IT roles. For Streamlit Sharing, you will integrate with your existing technology and configure the deployment settings. The Data & Analytics Survey, completed by business teams, identifies and prioritizes data use cases, audience size, and users. You will use the information collected in both surveys to plan your deployment strategy, including sizing, installation, and configuration of your Streamlit Cloud or integration and configuration of Streamlit Sharing. In addition to installing Streamlit Cloud or configuring Streamlit Sharing, administrators will also need to plan for the client software installation of Streamlit, Python, and any other dependencies that your app requires.