

# Low Level Design

## E-commerce Dashboard

<b>Written By</b>	Jay Prakash
<b>Document Version</b>	0.2
<b>Last Revised Date</b>	15/10/2022

## DOCUMENT CONTROL

### Change Record:

VERSION	DATE	AUTHOR	COMMENTS
0.1	15- Oct - 202	Jay Prakash	Introduction and architecture defined
0.2	20 -Oct - 2022	Jay Prakash	Architecture & Architecture description appended and updated.

### Reviews:

VERSION	DATE	REVIEWER	COMMENTS
0.2	22- Oct - 2022	Jay Prakash	Unit test cases to be added

## Contents

1.	Introduction.....	04
1.1	What is Low-Level Design Document? .....	04
1.2	Scope .....	04
2.	Architecture .....	05
3.	Architecture Description .....	08
3.1	Data Description.....	08
3.2	Web Scrapping .....	08
3.3	Data Transformation .....	08
3.4	Data insertion into database .....	08
3.5	Connection with SQL server.....	08
3.5	Export Data from database.....	12
3.6	Deployment.....	12
4.	Unit test cases.....	15

## 1. Introduction

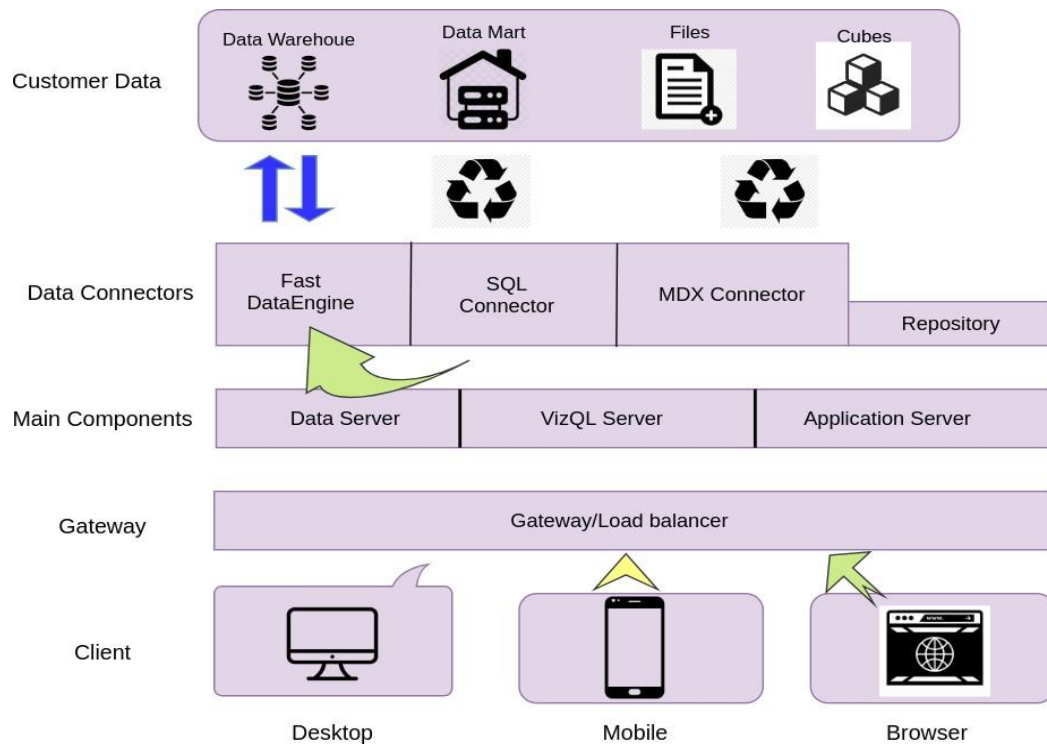
### What is Low-Level design document?

The goal of the LDD or Low-level design document (LLDD) is to give the internal logic design of the actual program code for the E-Commerce Dashboard. LDD describes the class diagrams with the methods and relations between classes and programs specs. It describes the modules so that the programmer can directly code the program from the document.

### Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. The process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

## 2. Architecture



## Excel Services Architecture

Excel Services is part of Microsoft SharePoint Server 2010. Excel Services is built on ASP.NET and SharePoint Foundation technologies. Following are the core components in Excel Services:  
Excel Web Access

- Excel Web Services
- User-defined functions (UDFs)
- ECMAScript (JavaScript, JScript)
- Representational State Transfer (REST) service
- Excel Calculation Services

### Note

Microsoft Excel Online, part of Office Online, also supports Excel workbooks in the browser. For more information about Excel Online, see documentation about Office Web Apps.

The Excel Web Access, Excel Web Services, UDFs, JavaScript, the REST service, and Excel Calculation Services components can be divided into two major groups: the components on a front-end server (also known as the "Web front end") and the component on a back-end application server. Components of a Web front end and a back-end application server

### A Web front end and a back-end application server

#### Web Front-End Servers and Back-End Application Servers

The Excel Web Access, Excel Web Services, UDFs, JavaScript, the REST service, and Excel Calculation Services components can be divided into components on the Web front-end server and components that live on a back-end application server. The Web front end includes Excel Web Access, JavaScript, the REST service, and Excel Web Services. The Excel Calculation Services component resides on the back-end application server, alongside any UDF assemblies that an administrator has added.

In the simplest configuration in SharePoint Server 2010—that is, a single computer running SharePoint Server 2010 as a stand-alone installation—all five components are installed on the same computer. However, in a typical enterprise environment with a large number of users, the components on the Web front-end server and the components on the back-end application server are on different computers in a farm configuration. It is possible to scale out the Web front-end server independently from the back-end application server. For example, you can have more Web front-end servers or more back-end application servers, depending on your organizational needs.

For information about Excel Services topology, scalability, performance, and security, see the SharePoint Server 2010 documentation on TechNet.

### Excel Web Access

Excel Web Access is a viewer page and an Excel Services web part that you can add to any web parts page in SharePoint Server 2010. Excel Web Access renders (in other words, creates the HTML for) live Excel workbooks on a webpage, and enables the user to interact with those workbooks and explore them. Excel Web Access is the visible Excel Services component for the user. You can use Excel Web Access like any other web part in SharePoint Server 2010. Excel Web Access does not require the user to install anything on the client computer.

The Excel Web Access web part properties are also customizable. For more information, see the

## Excel Web Services

Excel Web Services is the Excel Services component that provides programmatic access to its Web service. You can develop applications that call Excel Web Services to calculate, set, and extract values from workbooks, and to refresh external data connections. By using Excel Web Services, you can incorporate server-side workbook logic into an application, automate the updating of Excel workbooks, and create application-specific user interfaces around server-side Excel calculation.

### Note

When you make changes to a workbook—for example, by setting values to a range by using Excel Web Services—the changes to the workbook are preserved only for that session. The changes are not saved or persisted back to the original workbook. When the current workbook session ends (for example, when you call the `CloseWorkbook` method, or when the session times out), the changes that you made are lost.> If you want to save changes that you make to a workbook, you can use the `GetWorkbook` method, and then save the workbook. For more information, see [Microsoft.Office.Excel.Server.WebServices](#) . You can also open the workbook in edit mode and save the changes.

For more information about Excel Web Services, see [Excel Services Development Roadmap](#).

## User-Defined Functions (UDFs)

Excel Services UDFs enable you to use formulas in a cell to call custom functions that are written in managed code and deployed to SharePoint Server 2010. For more information about UDFs in Excel Services, see [Understanding Excel Services UDFs](#).

## ECMAScript (JavaScript, JScript)

The JavaScript object model in Excel Services enables developers to customize, automate, and drive the Excel Web Access web part control on a page. By using the JavaScript object model, you can build mashups and other integrated solutions that interact with one or more Excel Web Access web part controls on a page or an `iframe` with script on the page. It also enables you to add more capabilities to your workbooks and code around them.

For more information about the JavaScript object model in Excel Services, see the [Ewa namespace reference documentation](#).

## REST API

The REST API in Excel Services enables you to access workbook parts or elements directly through a URL. The URL contains a "marker" path, which is the entry point to an `.aspx` page, to the workbook file location, and to the path to the requested element inside the workbook.

The discovery mechanisms built into the Excel Services REST API enables developers and users to explore the content of a workbook manually or programmatically.

For more information about the REST API in Excel Services, see [Excel Services REST API](#).

## Excel Calculation Services

The role of Excel Calculation Services is to load workbooks, calculate workbooks, call custom code (UDFs), and refresh external data. It also maintains the session state for interactivity. Excel Calculation Services maintains a session for the duration of interactions with the same workbook by a user or caller. A session is closed when the caller explicitly closes it or when the session times out on the server. Excel Services caches the opened Excel workbooks, calculation states, and external data query results, for improved performance when multiple users access the same set of workbooks.

## Load-Balancing

In multiple-server configurations, Excel Services load-balances requests across multiple Excel Calculation Services occurrences in a farm configuration. If your installation includes multiple application servers, Excel Services will balance the load in an attempt to help ensure that no single application server is overloaded by requests.

Administrators can configure the load-balancing behavior

### 1. Gateway/Load Balancer

It acts as an Entry gate to the Tableau Server and also balances the load to the Server if multiple Processes are configured.

### 2) Application Server:-

Application Server processes (wgserver.exe) handle browsing and permissions for the Tableau Server web and mobile interfaces. When a user opens a view in a client device, that user starts a session on Tableau Server. This means that an Application Server thread starts and checks the permissions for that user and that view.

### 3) Repository:-

Tableau Server Repository is a PostgreSQL database that stores server data. This data includes information about Tableau Server users, groups and group assignments, permissions, projects, data sources, and extract metadata and refresh information.



## VIZQL Server:-

Once a view is opened, the client sends a request to the VizQL process (vizqlserver.exe). The VizQL process then sends queries directly to the data source, returning a result set that is rendered as images and presented to the user. Each VizQL Server has its own cache that can be shared across multiple users

## 5) Data Engine:-

It Stores data extracts and answers queries.

## 6) Backgrounder:-

The backgrounder Executes server tasks which includes refreshes scheduled extracts, tasks initiated from tabcmd and manages other background tasks.

## 7) Data Server:-

Data Server Manages connections to Tableau Server data sources

It also maintains metadata from Tableau Desktop, such as calculations, definitions, and groups.

# 3. Architecture Description

## Data Description

eCommerce datasets contain information about products, stores or marketplaces, sales, and customers. The data can be further segmented by categories, industries, or regions.

Product discovery KPIs: These indicators help eCommerce businesses to understand how customers find their products and online stores. Companies leverage these insights to boost online customer traffic and improve sales.

Online visibility of eCommerce stores

Online and offline impressions, indicating the frequency of serving advertisements to the target market.

Social media reach metrics: impressions, cost per thousand impressions (CPM), frequency

- Video hosting platform impressions
- Influencers and partners reach
- TV, media advertising, and podcast reach

## Customer Demographics:

- Age
- Location
- Gender
- Product favorites
- Brand affinity
- Most recent purchases

Onsite traffic metrics include the volume and frequency of online visits to an eCommerce website. This information can from analytics tools like Google Analytics and includes:

- Website sessions
- Number of users visiting the store
- Pages/Session, or the average number of pages viewed per session
- Bounce rate: percentage of single page visits
- Average session duration
- First-time visitors

## Organic traffic metrics:

- Total clicks from SERPs or Google search results pages
- Average click-through rate (CTR)
- Average ranking position of the eCommerce store

Email engagement metrics indicate customer behavior and intent, based on how they respond to emails from online shopping platforms. Customers receive several daily marketing emails from various eCommerce platforms, and email engagement data helps determine the success of marketing campaigns in driving retail eCommerce sales.

- Email list growth rate
- Email bounce rate
- Open rate
- Email conversion rate
- Email click-through rate
- Unsubscribes

Social media engagement attributes provide real-time data about social media sentiment.

- Likes per post
- Shares per post
- Comments per post
- Clicks per post

Conversion attributes: Metrics about conversation rates of online customers collected through eCommerce data scraping, including:

- Number of online transactions
- Average order value (AOV) of customers
- Specific sales data
- Number of visits to sale
- Sales conversion rates
- Shopping cart abandonment rate
- Cost Per Acquisition (CPA)

Custom datasets can also include attributes that match your specific requirements.

### **3.1. Web Scrapping**

Web scrapping is a technique to automatically extract content and data from websites using bots. It is also known as web data extraction or web harvesting. Web scrapping is made simple now days, many tools are used for web scrapping. Some of python libraries used for web scrapping are BeautifulSoup, Scrapy, Selenium, etc.

### **3.2. Data Transformation**

In the Transformation Process, we will convert our original datasets with other necessary attributes format. And will merge it with the Scrapped dataset.

### **3.3. Data Insertion into Database**

- a. Database Creation and connection - Create a database with name passed. If the database is already created, open the connection to the database.
- b. Table creation in the database.
- c. Insertion of files in the table

## **Make the SQL connection and set up the data source**

### **Step 1: Configuring Tableau**

Launch Tableau on your workstation and select SQL Server from the connect column on the left. This will open a dialogue box where you need to provide the connection details for SQL Server.

To connect with tableau, you will need to provide information about the server which hosts your database. If you want to connect to a contained database, you can also specify the name of the

<server\_name><port\_number>

Example query: my\_server 8051

There are two ways in which you can sign-in to the server, either by using Windows authentication or by using the username and password. Using the username and password becomes a must if you're working with a password-protected server in a non-Kerberos environment.

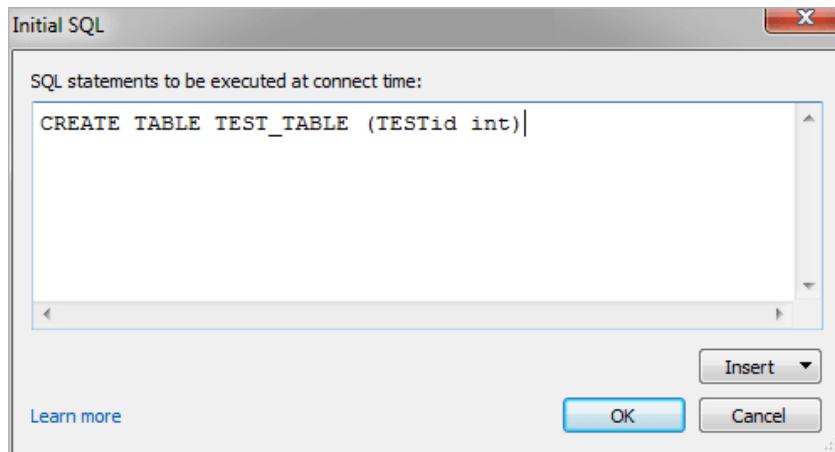


The screenshot shows the 'Microsoft SQL Server' connection window. It has two input fields at the top: 'Server:' with the value 'jrjpartner.database.windows.net' and 'Database:' with the value 'NYC'. Below these is a section titled 'Enter information to sign in to the database:'. It contains two radio buttons: 'Use Windows Authentication (preferred)' which is unselected, and 'Use a specific username and password:' which is selected. Under the selected option, there are two input fields: 'Username:' with the value 'tableau\_user' and 'Password:' which is empty. At the bottom left, there are two unchecked checkboxes: 'Require SSL' and 'Read uncommitted data'. At the bottom right, there is a 'Sign In' button. In the bottom left corner, there is a link that says 'Initial SQL...'. The entire window has a light gray background and a white border.

Click on Sign in to establish a connection. This will enable a connection without SSL. To establish an SSL enabled connection, click the Require SSL checkbox before you sign in.

SQL Server provides an option to let the user queries access the modified rows even before they have been committed. This option is called Read Uncommitted data. It saves time by preventing complex queries such as extract refreshes from locking the database and causing a delay. If this option is unchecked, Tableau makes use of default isolation levels.

If you want to run a specific SQL command every-time a new connection is established, you can use the Initial SQL option. This will open a dialogue box, where you can specify your desired SQL query.



## Step 2: Configuring Data Source

The data source page loads up after configuring the Tableau connector and successfully signing in. This is how the page looks like:



Select the data source name option and give a unique name to the database you are using. It's considered a good practice to have a unique name as it makes it much easier for users to identify the database from which data is being fetched.

To select the desired schema, you can use the schema drop-down list from the column on the left. You can also perform a text-based search to find the desired option. Now similarly find and select the desired table and drag it onto the canvas.

File Name	File Name	File Name	File Name	File Name	File Name	File Name
Pagepath	Sessionid	Source	Clientid	Timestamp	Pageviews	
/index.php/2016/12/...	1481366260388.vb9...	(direct)	821664097.14813...	2016-12-08T16:10:12.888+06:00	1	
/index.php/2016/12/...	1481379601122.covr...	(direct)	152909776.14813...	2016-12-10T20:04:35.888+06:00	1	
/index.php/2016/12/...	1481379601122.covr...	(direct)	152909776.14813...	2016-12-08T09:11:25.455+06:00	1	
/index.php/2016/12/...	1481379601122.covr...	(direct)	152909776.14813...	2016-12-10T20:14:20.337+06:00	1	
/index.php/2016/12/...	1481368188046.xdu...	(direct)	152909776.14813...	2016-12-08T21:19:48.246+06:00	1	
/index.php/2016/12/...	1481379601122.covr...	(direct)	152909776.14813...	2016-12-10T20:16:18.932+06:00	1	
/index.php/2016/12/...	1481374823379.vf9...	(direct)	821664097.14813...	2016-12-08T19:00:28.776+06:00	1	
/index.php/2016/12/...	1481379601122.covr...	(direct)	152909776.14813...	2016-12-10T20:12:10.451+06:00	1	

This is how you can connect SQL Server with Tableau. Now click on the sheets tab to begin the analysis.

Custom SQL features can be used to focus on specific SQL statements, rather than querying the entire database. Click on the Custom SQL option from the panel on the left. A new dialogue box will now open up, where you can provide the query you want to execute.

Edit Custom SQL

SELECT \* FROM [dbo].[Orders]  
WHERE [Orders].[Ship Mode] = 'Standard Class'

Preview Results...
Insert Parameter
OK
Cancel

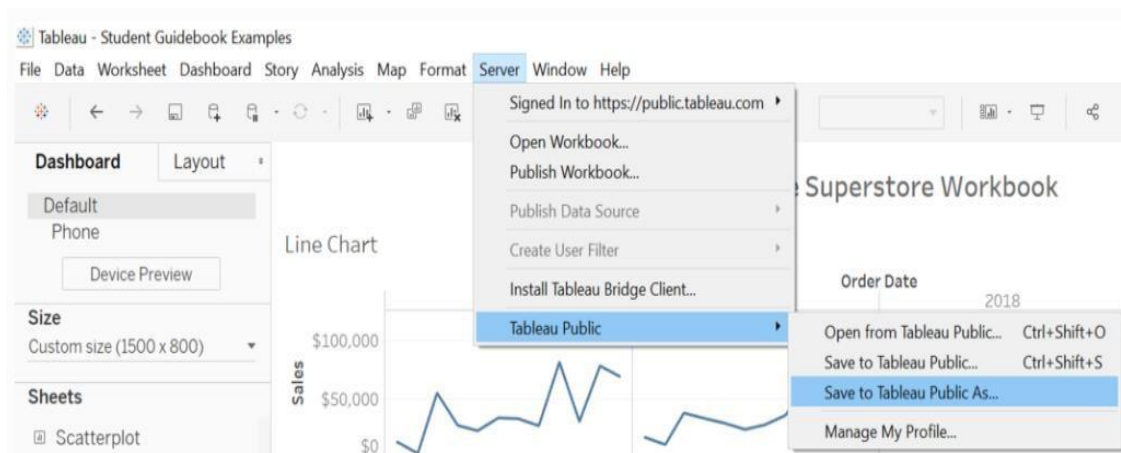
### 3.4. Export Data from Database

Data Export from Database - The data in a stored database is exported as a CSV file to be used for Data Pre-processing.

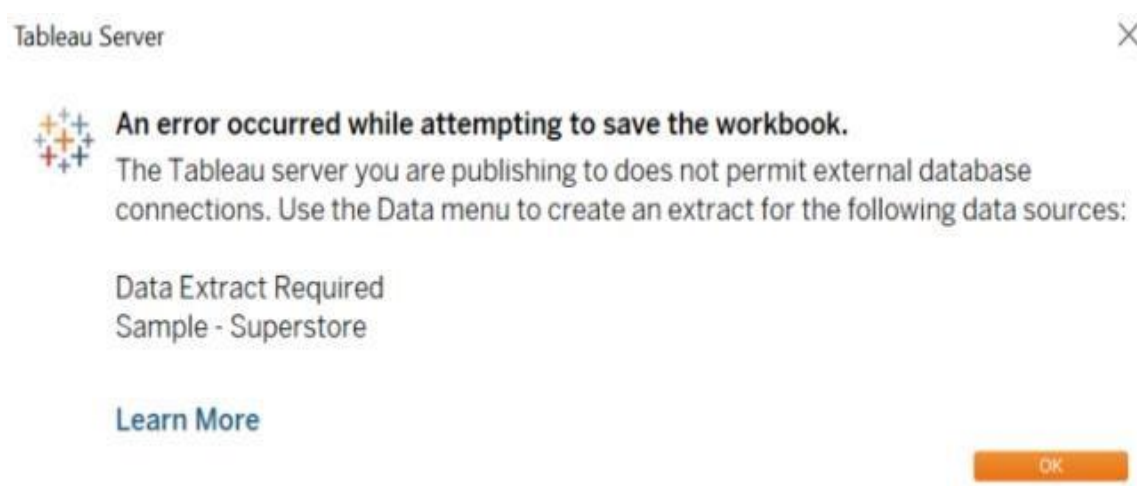
### 3.5 Deployment.

Once you've completed your dashboard, follow these steps:- **Server, Tableau Public, Save to Tableau Public As**

You may be prompted to log into your Tableau Public profile first if this is your first time publishing.



Next, fill out the title you want your viz to have and click “save”.



This message means that your connection to the Sample-Superstore data set is a live connection. Tableau Public cannot host live connections, so you’ll need to convert your connection to an extract (like a frozen screenshot of your data).

Here in the below screenshot, we can see that our workbook has been published to tableau public.

## E Commerce Sales Dashboard

Electronic ▼

**Sales**

394738

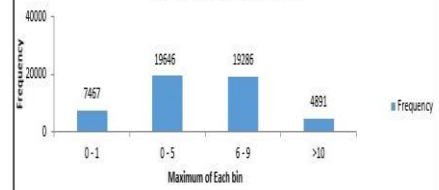
**Quantity**

31280

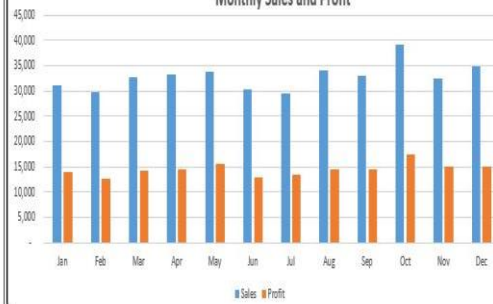
**Profit**

174175.0433

No. of orders Aging Wise



Monthly Sales and Profit



Region wise Sales

