

Low Level Design

Amazon Sales Analysis

Written By	Aniket Ghorpade
Document Version	0.1
Last Revised Date	10/06/2022

DOCUMENT CONTROL

Change Record:

VERSION	DATE	AUTHOR	COMMENTS
0.1	10-06 -22	Aniket Ghorpade	Introduction and architecture defined
0.2	11-06-22	Aniket Ghorpade	Architecture & Architecture description appended and updated.

Reviews:

VERSION	DATE	REVIEWER	COMMENTS
0.3	14-06-22	Aniket Ghorpade	Unit test cases to be added

Approval Status:

VERSION	REVIEW DATE	REVIEWED BY		APPROVED BY	COMMENTS

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

1.1 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 Amazon sales review 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.

1.2 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

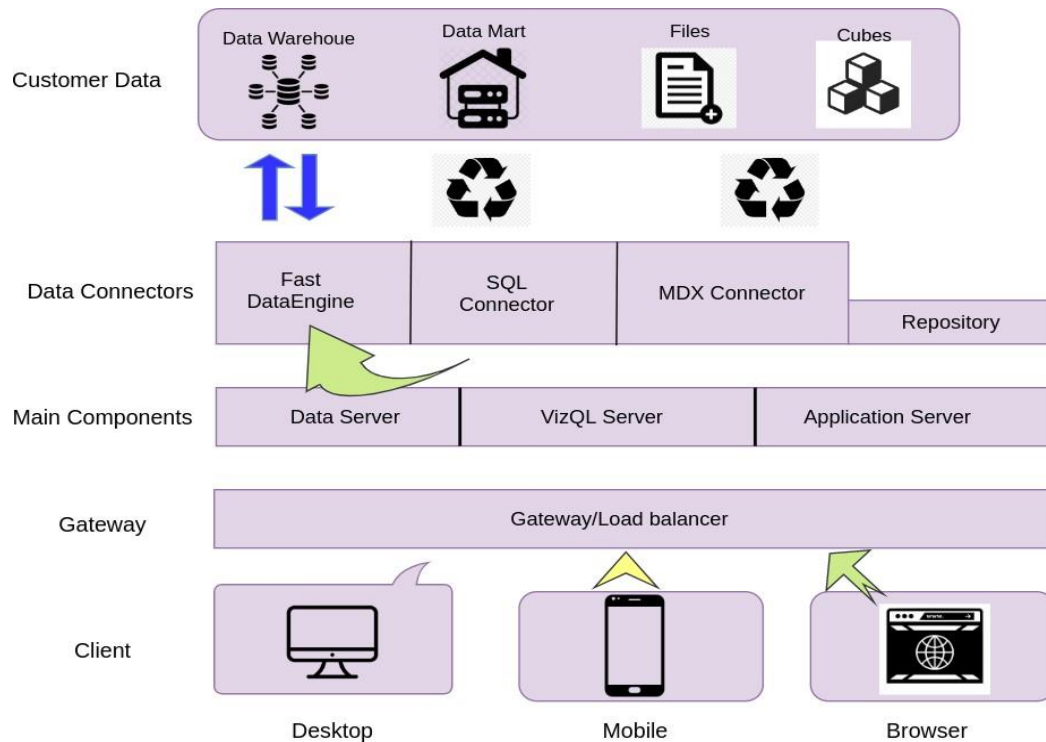


Tableau Server Architecture

Tableau has a highly scalable, n-tier client-server architecture that serves mobile clients, web clients and desktop-installed software. Tableau Server architecture supports fast and flexible deployments.

The following diagram shows Tableau Server's architecture:

Tableau Communication Flow

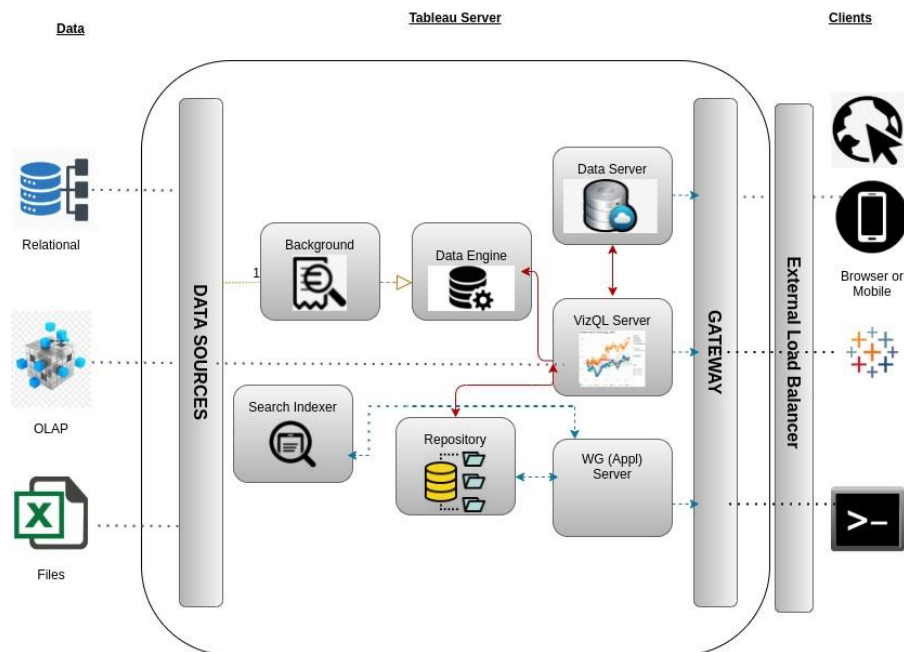


Tableau Server is internally managed by the multiple server processes.

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.

4) 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

3.1. Data Description

The Dataset contains customer information, sales information, discount, customer location etc.

1. Date: Date of order, promised delivery date.
2. Customer type: category of customer
3. Sales price: Details about sales price trade.
4. Discount: Amount if any discount is provided.
5. Item number: details about items that is products
6. Sales Quantity: Details about the quantity of sold product.

3.2. 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.

3.3. 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.

3.4. 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

3.5 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.

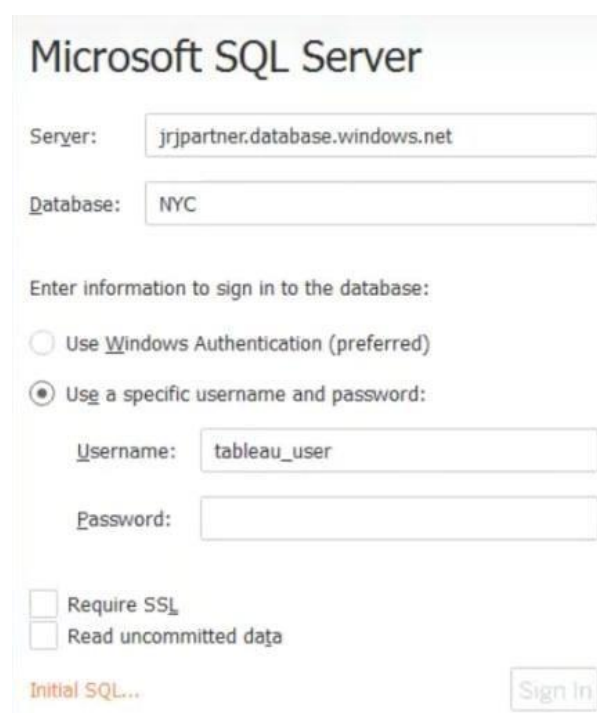
To connect with tableau, you will need to provide information about the server which hosts your database. If you want to connect to a contained database, you can also specify the name of the database.

To connect with a port other than the default port, you need to specify the port and server as follows:

```
<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.

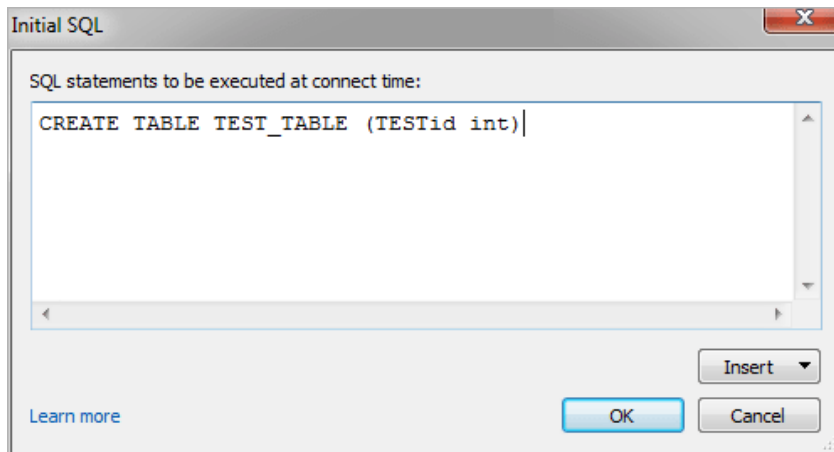


The screenshot shows the 'Microsoft SQL Server' connection window. It has two input fields at the top: 'Server:' with the value 'jrjpartner.database.windows.net' and 'Database:' with the value 'NYC'. Below these is a section titled 'Enter information to sign in to the database:'. It contains two radio 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. On the bottom left, there is a link that says 'Initial SQL...'. The window has a light gray border and a title bar.

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/...	148136626088.vb9...	(direct)	821664097.14813...	2016-12-08T16:10:12.889+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.xd.ku...	(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/...	1481374823776.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

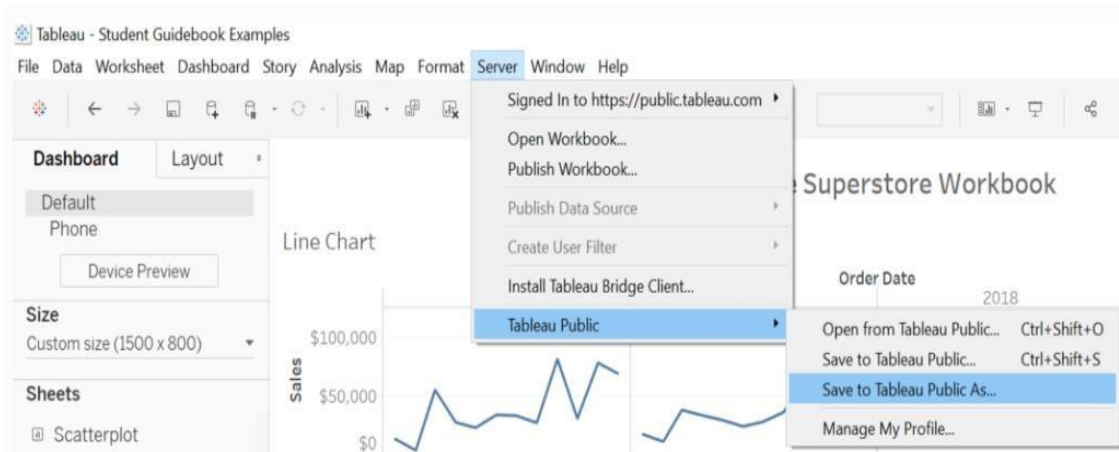
3.5. Export Data from Database

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

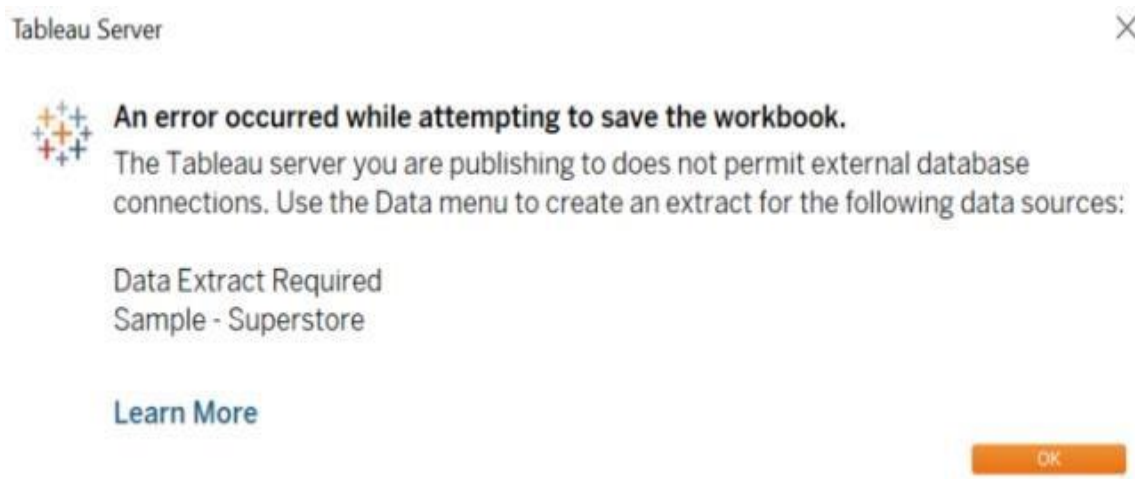
3.6 Deployment.

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.

4. Unit Test Cases

TEST CASE DESCRIPTION	EXPECTED RESULTS
Working of filter for customer selection on dashboard	When selected particular customer's information for that customer should be shown only.
Tooltip information for every sheet	When hovered on particular sheets tooltip information should be shown properly as per necessity.
Working of screen switching button	Screen switching button should work properly and it must go from one screen to another as per desired configuration.
Filter selection of sales quantity per item	Sales quantity per item should be shown as per the filter selection for various zones and items.
Quantity purchased per order	Quantity purchased per order should be checked by using dashboard by different customers, it is important test case.
Performance of Sales manager and sales representatives should be checked	From available bar graphs on dashboard we can check performance of various sales managers and sales representatives.