Low-Level Design

# Big Game Census Data Visualization

|  |  |
| --- | --- |
| **Written By** | Prachi Kumari |
| **Document Version** | 0.2 |
| **Last Revised Date** | 11/02/2023 |

**DOCUMENT CONTROL**

## Change Record:

|  |  |  |  |
| --- | --- | --- | --- |
| **VERSION** | **DATE** | **AUTHOR** | **COMMENTS** |
| 0.1 | 19- Nov -  2022 | Author 1 | Introduction and architecture defined |
| 0.2 | 11 -Feb-  2023 | Author 2 | Architecture & Architecture description appended and  updated. |
|  |  |  |  |
|  |  |  |  |

**Reviews:**

|  |  |  |  |
| --- | --- | --- | --- |
| **VERSION** | **DATE** | **REVIEWER** | **COMMENTS** |
| 0.2 | 8- Feb -  2023 | Author 3 | Unit test cases to be added |

**Approval Status:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **VERSION** | **REVIEW**  **DATE** | **REVIEWED BY** |  | **APPROVED BY** | **COMMENTS** |
|  |  |  |  |  |  |

# Contents

#### Introduction 04

* 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** |

# Introduction

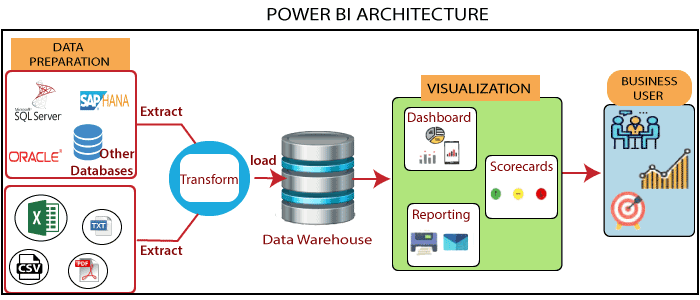
## What is a Low-Level design document?

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 Big Game Census Data Visualization dashboard. LDD describes the class diagrams with the methods and relations between classes and program 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.

# Architecture

****

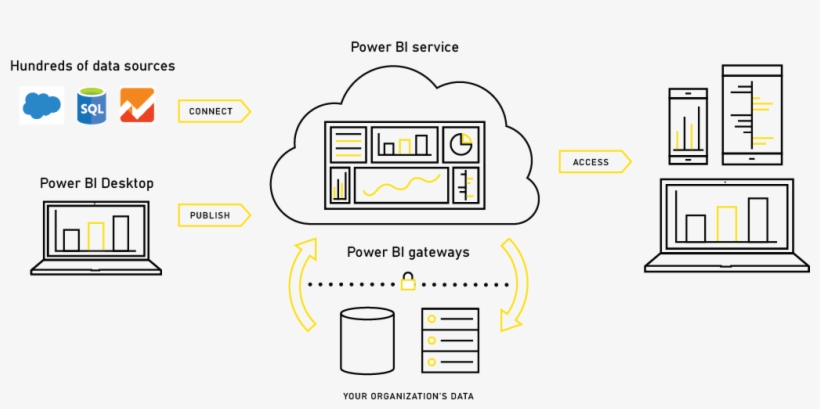
**Power BI Server Architecture**

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

The following diagram shows Power BI Server’s architecture:

**LOW LEVEL DESIGN**

**6**

****

Power BI Server is internally managed by the multiple server processes.

**1. Gateway/Load Balancer**

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

**2) Application Server: -**

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

1. **Repository: -**

Power Bi 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.

1. **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

1. **Data Engine: -**

It Stores data extracts and answers queries.

1. **Backgrounder: -**

The backgrounder Executes server tasks which include refreshing scheduled extracts, and tasks initiated from tab cmd, and manages other background tasks.

1. **Data Server: -**

Data Server Manages connections to Power Bi Server data sources

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

# Architecture Description

## Data Description

The Dataset contains the big game census of 2016 and 2017 Vintage US census that falls under the following category.

* + 1. Player Name: Each player taking part in super bowl 52
    2. Player jersey Number: jersey number of each player
    3. Player position: Position of the players on the ground
    4. Player Age: The age of the players
    5. Player team: Participating teams in the match
    6. Conference: Each conference the players take part in

## Web Scrapping

Web scraping 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 scraping is made simple nowadays, many tools are used for web scrapping. Some of the python libraries used for web scrapping are Beautiful Soup, Scrapy, Selenium, etc.

## 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.

## Data Insertion into Database

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

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

**Step 1: Configuring Power Bi**

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

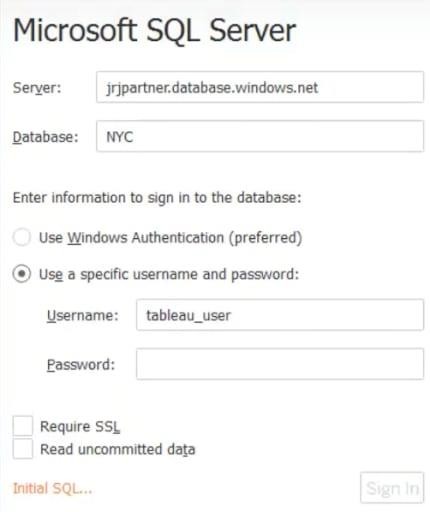
To connect with Bi, you will need to provide information about the server which hosts your database. If you want to connect to a contained database, you can also specify the name of the database.

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

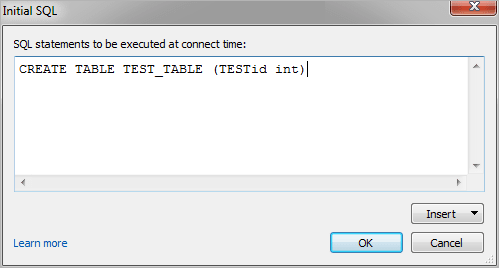
There are two ways in which you can sign in to the server, either by using Windows authentication or by using the username and password. Using the username and password becomes a must if you’re working with a password-protected server in a non-Kerberos environment.



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.

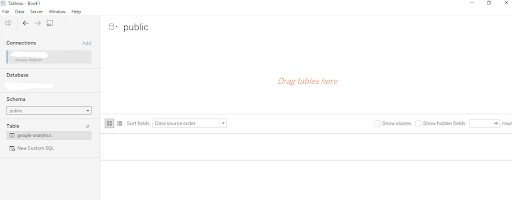
SQL Server provides an option to let the user queries access the modified rows even before they have been committed. This option is called Read Uncommitted data. It saves time by preventing complex queries such as extract refreshes from locking the database and causing a delay. If this option is unchecked,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



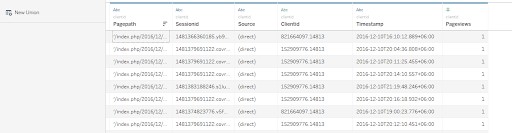
#### Step 2: Configuring Data Source

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



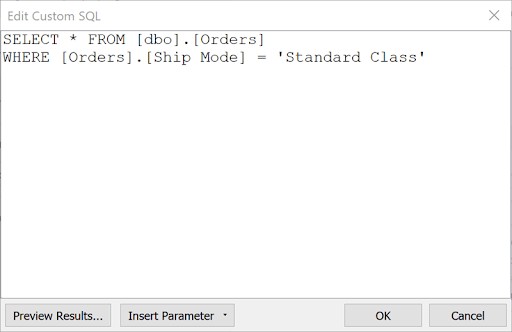
Select the data source name option and give a unique name to the database you are using. It’s considered a good practice to have a unique name as it makes it much easier for users to identify the database from which data is being fetched.

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.



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.



## Export Data from the Database

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

## Deployment.

Once you’ve completed your dashboard, follow these steps: **- Server, Bi Public, Save as to Power Bi public**

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

### LOW LEVEL DESIGN

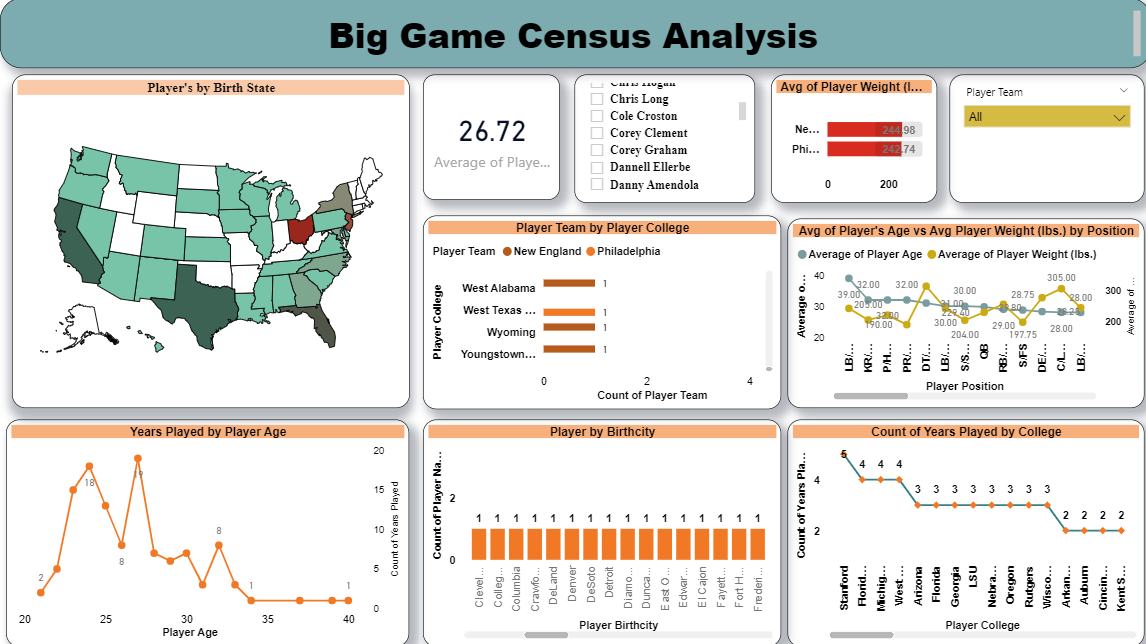
**12**

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 out workbook has been published to tableau public.

 **13 LOW-LEVEL DESIGNS**

****

# Unit Test Cases

|  |  |
| --- | --- |
| **TEST CASE DESCRIPTION** | **EXPECTED RESULTS** |
| Player weight slicer | When clicked on the slicer, a dropdown should occur which has  various parameters of the player. |
| Average player’s weight, players age by position | The time series graph shows the following changes as clicked on the marked points. |
| Relation Between player team and player college | Here a bar graph is shown of college VS player team |
| Years played by players’ age | Various city category is shown and a visualization is created  that shows the players and years played. |
| Relation between college and birth city and Built-up Parameters across the Cities | The visual should show a bubble diagram of the relation between various built-up parameters across various cities. |
| Min, Max & Avg. players Comparison by categories | This is an important visual in the bar graph which shows the category of Max players, birthplace, and Avg weight of the team across Built-up parameters and City categories. |