**A group of logos on a green background

Description automatically generated**

**DS6504: Business Intelligence and Big Data**

**Assignment 1: Interactive Dashboard Application**

**Tutor: Abubakar Siddique**

**Student: Adya Sinha (2230449)**

**Frances (Chunguang) Liu (2230873)**

**Vicky Miguel (2211507)**

**Submission Date: 7 September 2024**

**Word Count: 2005 words**

Table of Contents

[Executive Summary 4](#_Toc176643186)

[SQL Views 5](#_Toc176643187)

[Overview 5](#_Toc176643188)

[Views Created 5](#_Toc176643189)

[a. vInventoryByType 5](#_Toc176643190)

[b. vAcceptedByCountry 7](#_Toc176643191)

[c. vRejectedProductsByType 9](#_Toc176643192)

[SQL Stored Procedures 11](#_Toc176643193)

[Overview 11](#_Toc176643194)

[Stored Procedure Created 11](#_Toc176643195)

[a. spMaxInventoryByType 11](#_Toc176643196)

[b. spAcceptedByCountry 13](#_Toc176643197)

[c. spAvgRejected 14](#_Toc176643198)

[Dashboard 16](#_Toc176643199)

[Errors Encountered 16](#_Toc176643200)

[Deployment 16](#_Toc176643201)

Table Of Figures

[Figure 1: Tables and attributes in view ‘vInventoryByType’ 5](#_Toc176643162)

[Figure 2: SQL code to create the view 6](#_Toc176643163)

[Figure 3: Result of the SQL code 7](#_Toc176643164)

[Figure 4: Tables and attributes in view "vAcceptedByCountry" 7](#_Toc176643165)

[Figure 5: SQL code to create the view 8](#_Toc176643166)

[Figure 6: Result of the SQL query 8](#_Toc176643167)

[Figure 7: Tables and attributes in view "vRejectedProductsByType" 9](#_Toc176643168)

[Figure 8: SQL script to create the view 10](#_Toc176643169)

[Figure 9: Output of the view created 10](#_Toc176643170)

[Figure 10: SQL query for the stored procedure 12](#_Toc176643171)

[Figure 11: Output for the "spMaxInventoryByType" by the parameters- Year 2009 & Month 4. 12](#_Toc176643172)

[Figure 12: SQL code for the stored procedure 13](#_Toc176643173)

[Figure 13: Output of the ‘spAcceptedByCountry’ by the parameters- Year 2009 & Month 4. 14](#_Toc176643174)

[Figure 14: SQL query for the stored procedure 15](#_Toc176643175)

[Figure 15: Output of the ‘spAvgRejected’ by the parameters- Year 2009 & Month 4. 15](#_Toc176643176)

[Figure 16: Sample of Dashboard Result - Year 2009 & Month 4. 17](#_Toc176643177)

[Figure 17: Sample of Dashboard Result - Year 2010 & Month 6. 18](#_Toc176643178)

# Executive Summary

This assignment focuses on the creation of an interactive dashboard application using SQL and Microsoft Visual Studio. This report focuses on the development of the SQL views and stored procedures to then support the creation of an interactive dashboard for a production report.

There are three SQL views created, namely the “**vInventoryByType**”, which consolidates the inventory data by product type and subtype which helps provide insights into the inventory levels across all the different categories. There is also the “**vAcceptedByCountry**” view which collects data on accepted products from various plants and countries, and then the “**vRejectedProductsByType**” view which calculates the percentage of the rejected products by product type and subtype.

There is also the creation of three stored procedures developed to help enhance the data processing and reporting. The “**spMaxInventoryByType**” procedure retrieves the maximum inventory levels for all the different product types and subtypes for a given year and month. The procedure “**spAcceptedByCountry**” collects the total number of accepted products by plant and country for any given timeframe. “**spAvgRejected**” is a stored procedure which calculates the average percentage of the rejected products for each of the product subtype.

Finally, the dashboard integrates the SQL views and stored procedures to serve as the perfect tool for data analysis, metrics to help support strategic decision making, and displaying key performance indicators. This report also further highlights the errors encountered during the development and deployment, such as request length issues and map errors.

# SQL Views

## Overview

Views are virtual tables created because of an SQL statement. The fields on views are retrieved from one or more tables in the database and contain consolidated data from queries. Views do not store or change data.

## Views Created

The following views are created to retrieve and display the required information needed for the Production Report.

### a. vInventoryByType

Description

The view “vInventoryByType” is a virtual table which provides a combined view of the inventory levels categorized by product and subtype. This view joins many of the dimension tables with the fact table “InventoryFact” to help enhance the inventory data with the description about product types and subtypes. There are different columns such as- “ProductTypeName”, “ProductSubtypeName”, “DateOfInventory”, and “InventoryLevel” which are extracted from different tables such as “DimProductType”, “DimProductSubType”, and “InventoryFact”. This view helps create a report on how inventory levels are distributed among all the various product and subproduct categories.

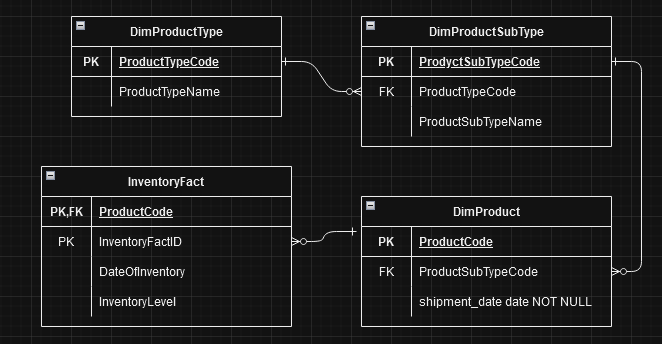


Figure 1: Tables and attributes in view ‘vInventoryByType’

The following table displays the list of table names, primary and foreign keys used to create the view “vInventoryByType”:

|  |  |  |  |
| --- | --- | --- | --- |
| **Table** | **Primary Key** | **Foreign Key** | **Links to Table** |
| DimProductType | ProductTypeCode |  |  |
| DimProductSubType | ProductSubtypeCode | ProductTypeCode | DimProductType |
| DimProduct | ProductCode | ProductSubtypeCode | DimProductSubType |
| InventoryLevel | InventoryFactID  ProductCode | ProductCode | DimProduct |

Table 1. List of tables, primary keys and foreign keys to create view "vInventoryByType"

A screenshot of a computer code

Description automatically generated

Figure 2: SQL code to create the view

A screenshot of a computer

Description automatically generated

Figure 3: Result of the SQL code

### b. vAcceptedByCountry

Description

The view ‘vAcceptedByCountry’ is created to display the number of accepted products across different countries and plants. This view displays a combination of data from several tables, such as ‘DimCountry’, ‘DimPlant’, ‘DimMachine’, and ‘ManufacturingFact’. Detailed information about country code, country name, plant name, the date of manufacture (in a specific format), and the number of accepted products is shown. The purpose of creating this view is to help report and analyze how many of the products were accepted based on the country, plant, and specific dates. Meanwhile, the newly created view will not make any changes to the database.

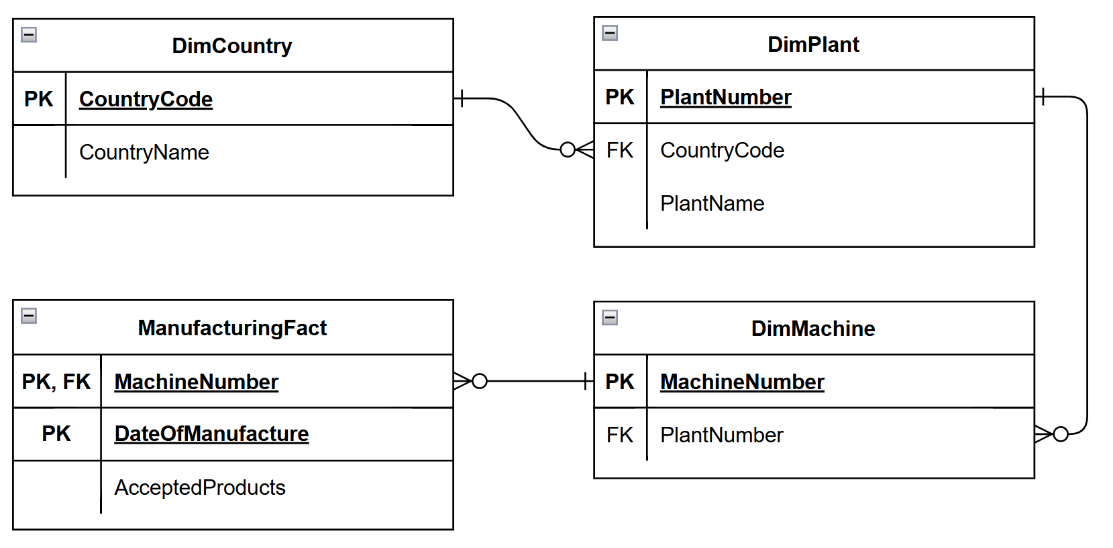


Figure 4: Tables and attributes in view "vAcceptedByCountry"

The table below shows the list of primary and foreign keys used in the query to join the tables.

|  |  |  |  |
| --- | --- | --- | --- |
| **Table** | **Primary Key** | **Foreign Key** | **Links to** |
| DimCountry | CountryCode |  |  |
| DimPlant | PlantNumber | CountryCode | DimCountry.CountryCode |
| DimMachine | MachineNumber | PlantNumber | DimPlant.PlantNumber |
| ManufacturingFact | MachineNumber  DateOfManufacture | MachineNumber | DimMachine.MachineNumber |

Table 2. List of tables, primary keys and foreign keys to create view "vAcceptedByCountry"

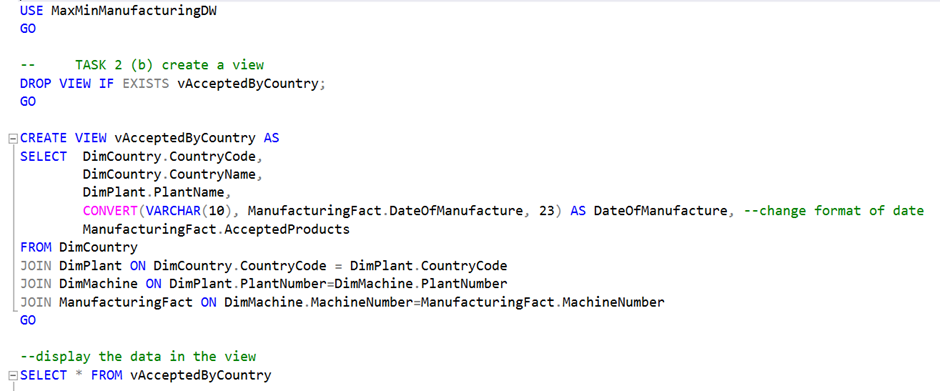


Figure 5: SQL code to create the view

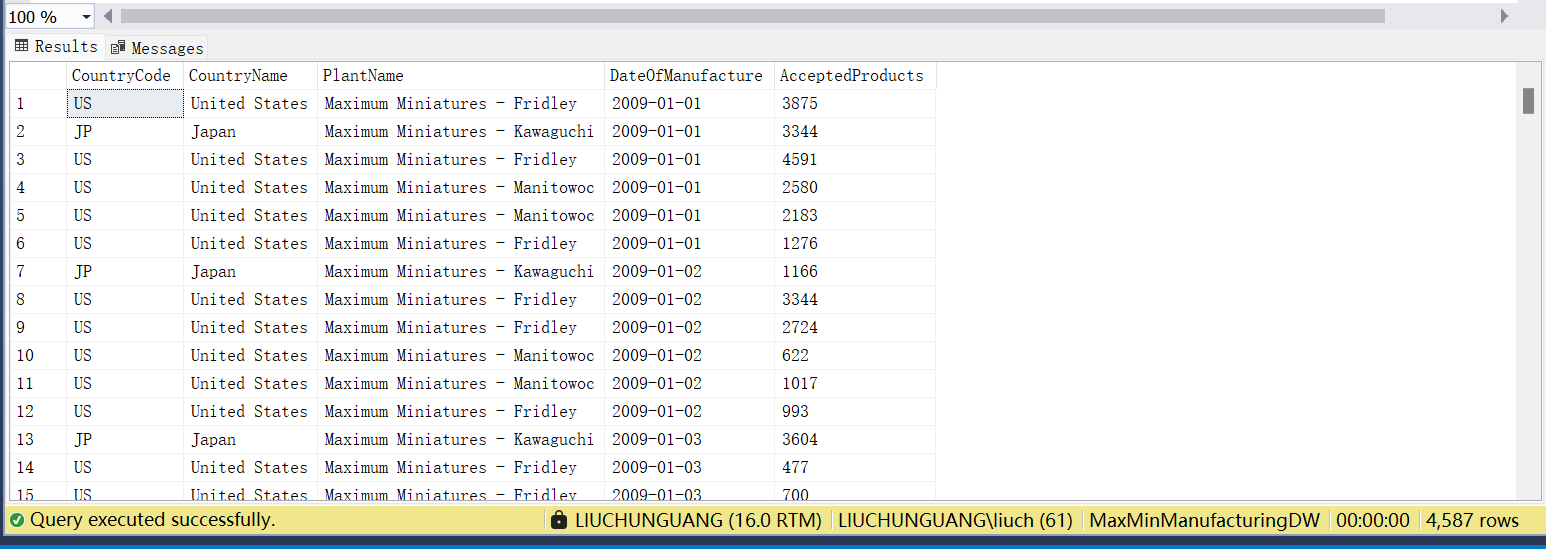


Figure 6: Result of the SQL query

### c. vRejectedProductsByType

Description

The vRejectedProductsByType is a virtual table that displays the number of manufactured products and the percentage of rejected products. This view also provides information about the products such as product type name, product subtype name, and the product manufacture date. The tables “DimProductType”, “DimProductSubtype”, “DimProduct” and “ManufacturingFact” are joined using the INNER JOIN to display the product information.

The purpose of creating the vRejectedProductsByType is to display and calculate the total number of manufactured products and the percentage of rejected products by product type for report and analysis.

A diagram of a product

Description automatically generated

Figure 7: Tables and attributes in view "vRejectedProductsByType"

The table below shows the list of table names, the primary keys, and the foreign keys that are used to create the view vRejectedProductsByType.

|  |  |  |  |
| --- | --- | --- | --- |
| **Table** | **Primary Key** | **Foreign Key** | **Links to Table** |
| DimProductType | ProductTypeCode |  |  |
| DimProductSubType | ProductSubtypeCode | ProductTypeCode | DimProductType |
| DimProduct | ProductCode | ProductSubtypeCode | DimProductSubType |
| ManufacturingFact | DateOfManufacture  ProductCode | ProductCode | DimProduct |

Table 3. List of tables, primary keys and foreign keys to create the view " vRejectedProductsByType "

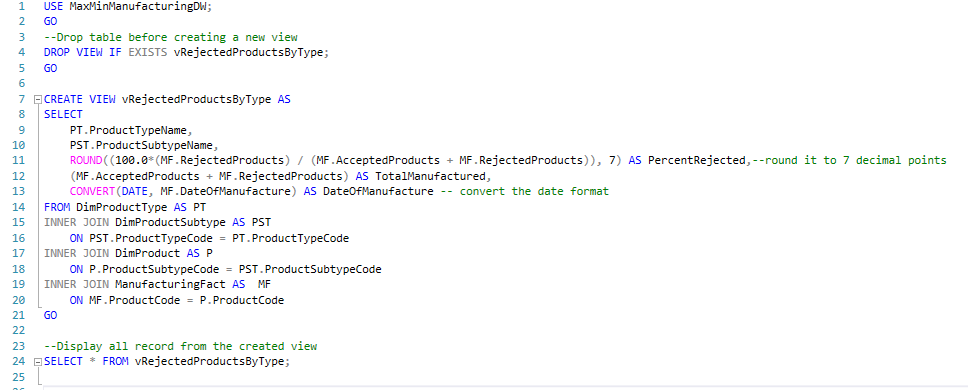


Figure 8: SQL script to create the view

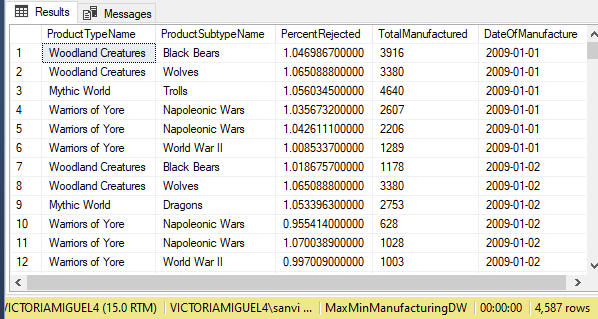


Figure 9: Output of the view created

# SQL Stored Procedures

## Overview

The Stored Procedures are a group of SQL statements compiled, created, and stored in the database. They can be reused by calling the procedure and providing parameter values. Stored procedures are used to maintain data consistency and efficiency.

## Stored Procedure Created

The following stored procedures were created to populate and retrieve required information in the Production Report.

### a. spMaxInventoryByType

The stored procedure “spMaxInventoryByType” helps retrieve the maximum inventory levels for all the different product types and subtypes for a given year and its month. This stored procedure is a useful tool to help analyse the inventory trends according to the filters over time, helping to identify periods of high inventory levels. This is critical for inventory management and business decisions.

There are two parameters – “@Year INT”, which specifies the year for which the inventory data is being queried; and “@Month INT”, which specifies the month within the given year for which the inventory data is being queried.  This query selects three columns, which are- “ProductTypeName”, “ProductSubtypeName”, and the maximum inventory level “(MaxInventoryLevel)” for the specified year and month. The “MAX(InventoryLevel)” is used to find the highest inventory level for each product type and subtype combination within the given time frame.

The results are grouped by the “ProductTypeName” and “ProductSubTypeName” which results for the query to calculate the maximum inventory level for each combination of the product type and subtype. Then the results are ordered alphabetically by the “ProductTypeName” and “ProductSubTypeName”.

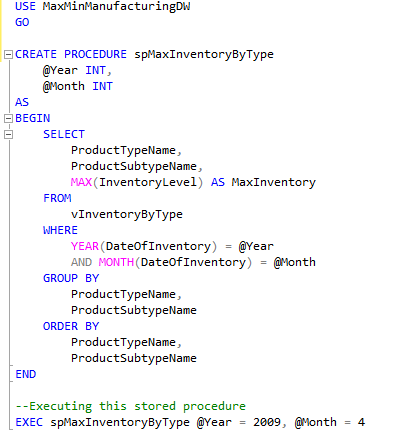


Figure 10: SQL query for the stored procedure

A screenshot of a computer

Description automatically generated

Figure 11: Output for the "spMaxInventoryByType" by the parameters- Year 2009 & Month 4.

### b. spAcceptedByCountry

The stored procedure ‘spAcceptedByCountry’ is created to retrieve information about the total number of accepted products produced in different countries and plants. This procedure takes two parameters, year and month. Thus, the users can filter specific results they need by year and month. This makes analyzing and reporting total manufactured products within specific periods possible. This procedure can return country code, country name, plant name, the total number of accepted products, and plant name with specific format requirement.

Stored procedure ‘spAcceptedByCountry’ query groups the results by ‘CountryCode’, ‘CountryName’, and ‘PlantName’. This makes the data organized or grouped by each plant within each country.

Aggregation contains operations like summing, averaging, or counting. The aggregation in this query is ‘SUM(AcceptedProducts)’. This aggregation operation calculates the total number of accepted products for each country and plant. A REPLACE function is also used to remove the prefix "Maximum Miniatures -" and replace it with an empty string.

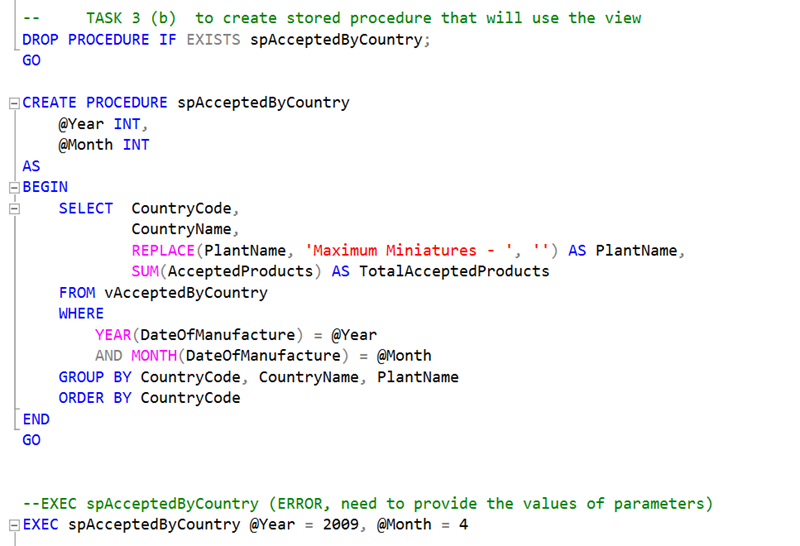


Figure 12: SQL code for the stored procedure

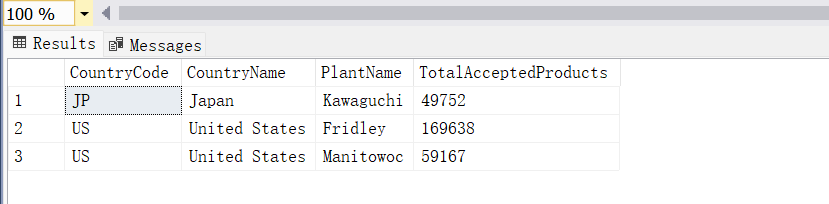


Figure 13: Output of the ‘spAcceptedByCountry’ by the parameters- Year 2009 & Month 4.

### c. spAvgRejected

The stored procedure “spAvgRejected” is designed to display the total number of products manufactured and the average or percentage of rejected products for each product subtype. This procedure uses the view “vRejectedProductsByType” to retrieve the product information and perform the required calculations. A valid Year and Month parameters are required to enter to generate a report, otherwise, an invalid Year or Month message will appear. The report will display the product information such as ProductTypeName, ProductSubTypeName, AvgPercentRejected, and TotManufactured products.

The GROUP BY clause has been used to perform the required calculations, in this case, we grouped it by ProductTypeName and ProductSubTypeName. The aggregate function AVG() has been used in calculating the percentage of rejected products with each product subtype. We also use the SUM() function to calculate the total of manufactured products and used the ORDER BY clause to sort the result by Product type name.

The code below is used to create the stored procedure “spAvgRejected” and the script on how to call and execute it.

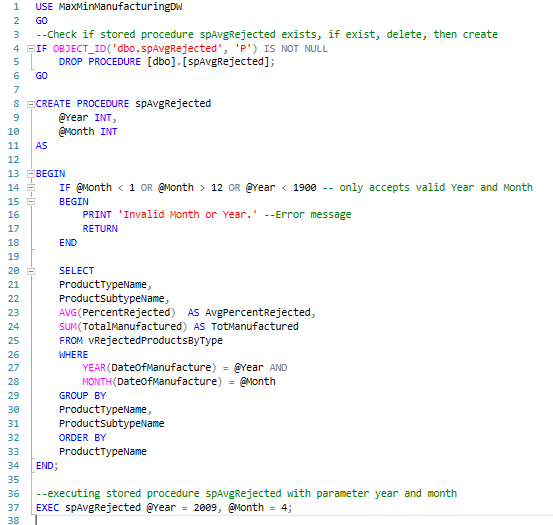


Figure 14: SQL query for the stored procedure

A screenshot of a computer

Description automatically generated

Figure 15: Output of the ‘spAvgRejected’ by the parameters- Year 2009 & Month 4.

# Dashboard

Dashboards are used for data analysis to support the organization in making strategic decisions. It displays valuable information such as key performance indicators, metrics, and other business-related data that are useful to facilitate decision-makers with their analysis and making informed decisions.

## Errors Encountered

Here are some errors encountered during the report and dashboard creation and their solutions.

* **Map Error**

The map error is caused by the missing component that is used for the visualization. To solve this issue, the component SqlServerSpatial160.dll should be saved in the directory where the Visual Studio tools, extensions, and components were saved. In my case, I saved the file to this directory, C:\Program Files (x86)\Microsoft Visual Studio\2019\Enterprise\Common7\IDE\ Extensions\Microsoft\SQLDB\DAC

* **Maximum Request Length Error**

This error is caused by missing or insufficient maxRequestLength value. To solve this issue, update the web.config file. Locate the httpRunTime tag and update or add a maxRequestLength="12582912" to increase the maxRequestLength value.

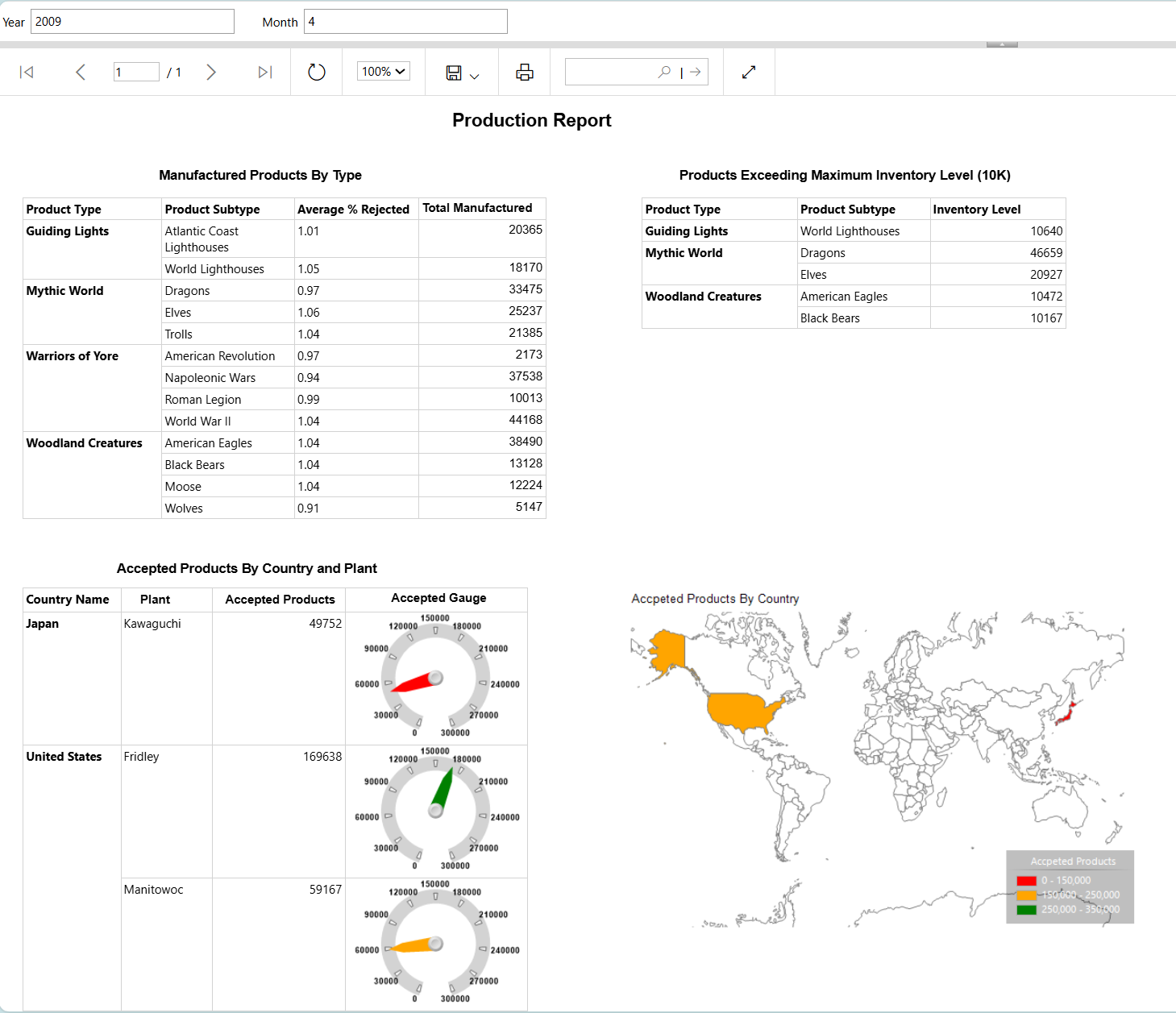
* **Report Definition Error**

The report definition error was encountered because of inconsistencies in data sources and dataset names. Ensure that all the datasets and data sources are linked and configured properly to avoid this error.

## Deployment

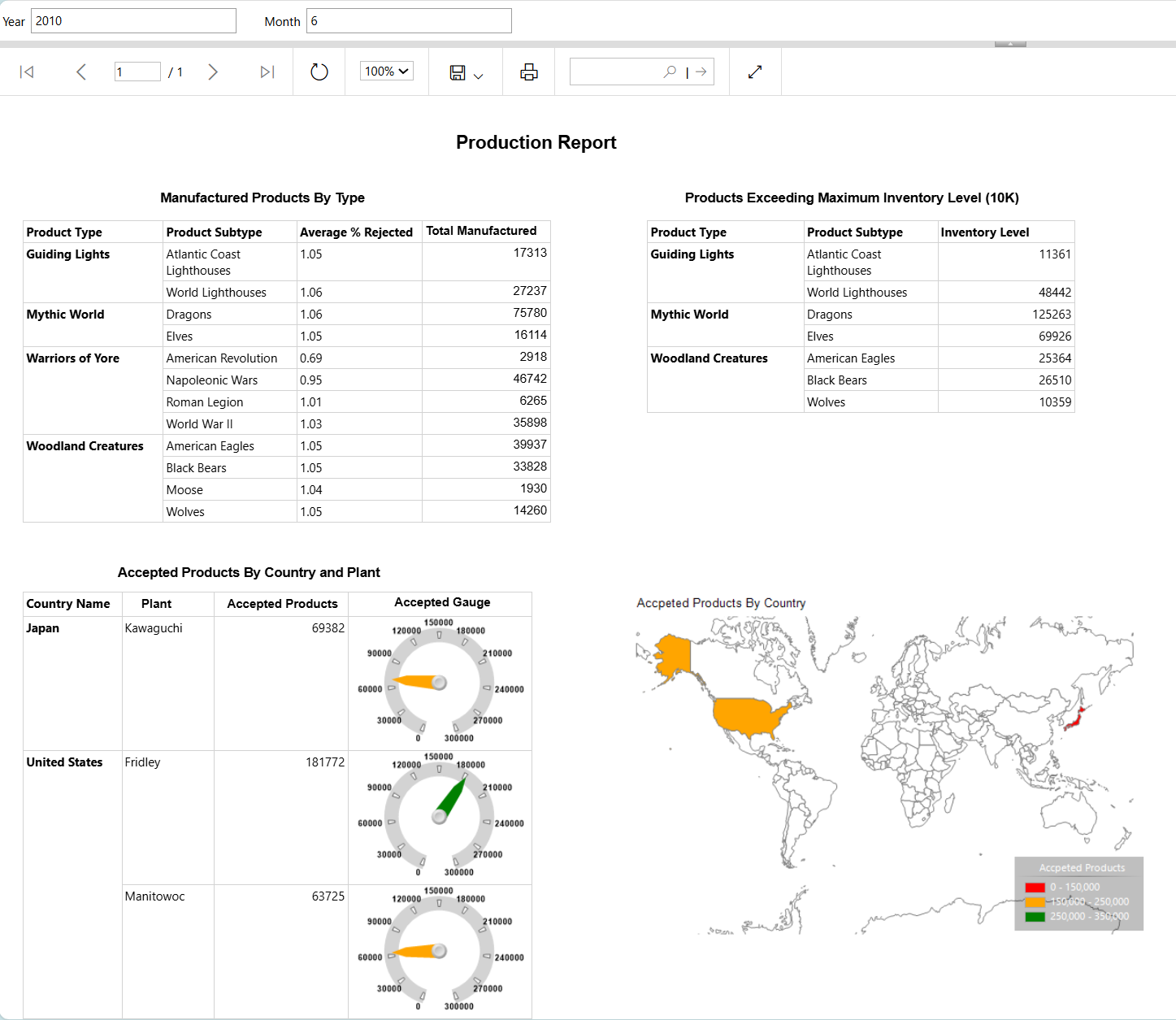
The views and stored procedures mentioned in this report are used to integrate in the data sets and sources to produce the required information in the Production Report.

The screenshot below is an example of the dashboard using 2009 and 4 as the year and month input parameters.



*Figure 16: Sample of Dashboard Result - Year 2009 & Month 4.*

The screenshot below of the dashboard using 2010 and 6 as the year and month input parameters.



*Figure 17: Sample of Dashboard Result - Year 2010 & Month 6.*