# Overview of the Assignment:

In this assignment we will create a few analytical queries on a data warehouse. We will also explore some aspects of Tableau.

# Part 1 – Restore database

Depending on whether you’ve chosen SQL Server or PostgreSQL: download the appropriate file and restore the database to your machine.

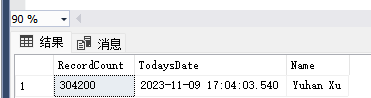
Write a query showing count of records in the manufacture fact database, as well as today’s date and your name (two additional columns), take a screenshot of the result.

**Paste the code and a screen shot of the results.**

**SQL Code:**

|  |
| --- |
| SELECT  COUNT(\*) AS RecordCount,  GETDATE() AS TodaysDate,  'Yuhan Xu' AS Name  FROM  MANUFACTURE\_FACT; |

**Screenshot of result:**



# Part 2 – Examine the Schema and Data to familiarize yourself.

Query the dimension tables to see what they look like. Refer to the schema diagram below. Then answer the questions below.



1. Pick two dimensions and list the hierarchy columns in each of these dimensions which have them. Provide the dimension name and the columns in order from highest to lowest (drill-down).

**Dimension 1:**

Dimension Name:PRODUCT

Hierarchy Columns (highest to lowest): PRODUCT\_KEY, PRODUCT\_ID, PRODUCT\_NAME, BRAND\_NAME, CATEGORY\_NAME, CLASS\_NAME, TOTAL\_PRODUCT

**Dimension 2:**

Dimension Name: CUSTOMER

Hierarchy Columns (highest to lowest): CUSTOMER\_KEY, CUSTOMER\_ID, CUSTOMER\_NAME, ADDRESS, CITY\_NAME, STATE\_PROVINCE\_NAME, COUNTRY\_NAME, REGION\_NAME, POSTAL\_CODE, CUSTOMER\_SEGMENT

1. Would any of the dimensions benefit from SCD2 or SCD3 implementation? Which ones and what would you recommend as a change. Provide two SCD changes you would suggest implementing. Describe the columns you would add, and how the ETL process would change to maintain these. Hint: It’s not just about the above design, review the data as well and explain your reasoning briefly.

**Dimension 1 to change to SCD and how would you change it:**

Dimension Name: CUSTOMER

SCD Change Recommendation: Implement SCD type 2 to track historical changes in the customer's address, city, state, and potentially the customer segment. You would add columns for EFFECTIVE\_DATE, EXPIRATION\_DATE, and IS\_CURRENT flag. The ETL process would need to incorporate logic to insert a new record with the updated address information and update the EXPIRATION\_DATE of the old record to the current date while marking it as not current.

**Dimension 2 to change to SCD and how would you change it:**

Dimension Name: PRODUCT

SCD Change Recommendation: Implement SCD type 3 to maintain the history of a product's category or class if these are subject to frequent changes. You would add columns like PREVIOUS\_CATEGORY\_NAME and PREVIOUS\_CLASS\_NAME. The ETL process would update these columns with the previous values whenever a change occurs, thus maintaining a history of changes.

1. Pick one fact table from the design what are the measures?

**Fact table:** SALES\_FACT

Measures: Typically, the measures in a sales fact table would include ORDER\_AMOUNT, UNIT\_SOLD, and any other numeric fields that represent transactional data. However, the exact measures can only be determined with certainty by looking at the data contained in the SALES\_FACT table and understanding the business logic applied to these fields.

# Part 3 – Dimensional Queries

1. Write and execute a query that identifies for each year, three factories which produced (passed) the most units. Your output should have these five columns and each year should show the top three factories:
   * Year
   * Factory name (label)
   * Total Units produced (passed) for each factory for each year
   * Total Units failed for each factory for each year
   * Factory name rank (based on total units produced)

Sort your result by the latest year first, with oldest year last.

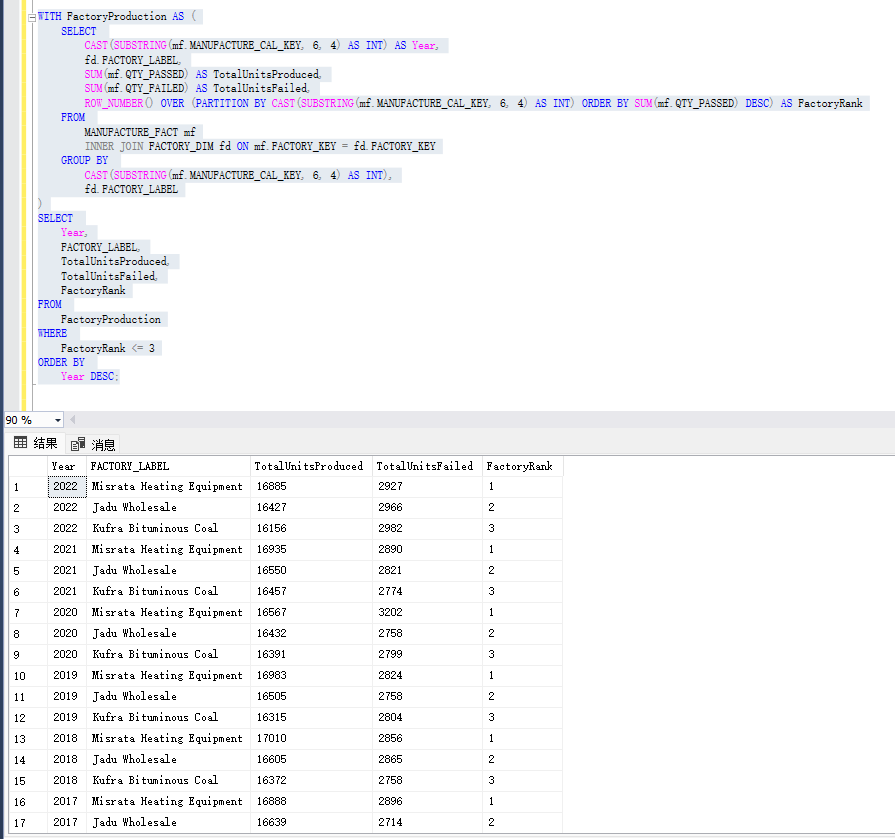
* Hint: Solve this query in multiple steps, then put it all together. For example, join the data first, compute your aggregates, finally filter the aggregates checking your results in each step. Using an inline view or CTE might be helpful to solve this question by breaking it into two steps.

**Paste the SQL code and a screen shot of the results.**

**SQL Code:**

|  |
| --- |
| WITH FactoryProduction AS (  SELECT  CAST(SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 4) AS INT) AS Year,  fd.FACTORY\_LABEL,  SUM(mf.QTY\_PASSED) AS TotalUnitsProduced,  SUM(mf.QTY\_FAILED) AS TotalUnitsFailed,  ROW\_NUMBER() OVER (PARTITION BY CAST(SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 4) AS INT) ORDER BY SUM(mf.QTY\_PASSED) DESC) AS FactoryRank  FROM  MANUFACTURE\_FACT mf  INNER JOIN FACTORY\_DIM fd ON mf.FACTORY\_KEY = fd.FACTORY\_KEY  GROUP BY  CAST(SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 4) AS INT),  fd.FACTORY\_LABEL  )  SELECT  Year,  FACTORY\_LABEL,  TotalUnitsProduced,  TotalUnitsFailed,  FactoryRank  FROM  FactoryProduction  WHERE  FactoryRank <= 3  ORDER BY  Year DESC; |

**Screenshot of result:**



1. Drill down and rollup!

**Question 2A** Write and execute a query that that identifies total units produced (passed) for each of the factories for each month in 2022 with a subtotal for each factory.

Your result set should have the following four fields. You will want to filter the data for 2022.

* Factory name
* Month – Notice that the month is not quite in readable format. Transform the month into the following format '01-January’, ‘02-February', etc. Hint – a case statement will be helpful here.
* Total Units produced (passed) for each factory for each month
* Total Units failed for each factory for each month

Complete the result by adding a Rollup to show subtotals by factory. With the Rollup function, you will notice a NULL value for the month column for the subtotal on each of the factories.

**Paste the SQL code and a screen shot of the results.**

**SQL Code:**

|  |
| --- |
| SELECT  fd.FACTORY\_LABEL,  CASE  WHEN CHARINDEX('-', mf.MANUFACTURE\_CAL\_KEY) > 0 THEN  SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 2) + '-' +  DATENAME(MONTH, CONVERT(DATE, SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 10), 120))  END as Month,  SUM(mf.QTY\_PASSED) AS TotalUnitsProduced,  SUM(mf.QTY\_FAILED) AS TotalUnitsFailed  FROM  MANUFACTURE\_FACT mf  INNER JOIN FACTORY\_DIM fd ON mf.FACTORY\_KEY = fd.FACTORY\_KEY  WHERE  CAST(SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 4) AS INT) = 2022  GROUP BY  fd.FACTORY\_LABEL,  CASE  WHEN CHARINDEX('-', mf.MANUFACTURE\_CAL\_KEY) > 0 THEN  SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 2) + '-' +  DATENAME(MONTH, CONVERT(DATE, SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 10), 120))  END  WITH ROLLUP; |

**Screenshot of result:**



**Question 2B:** Outline one suggestion you would implement as part of the design and ETL to make this question easier to solve?

**Short Answer:**

To make the querying process easier, especially when dealing with date parts like months and years, I would suggest implementing a Date Dimension. This Date Dimension would include columns for the year, quarter, month, day, and other potentially useful date parts. The Date Dimension would have a key that could be joined to the MANUFACTURE\_CAL\_KEY in the MANUFACTURE\_FACT table.

The ETL process to populate this Date Dimension would parse the date information from the MANUFACTURE\_CAL\_KEY and store it in a structured format. During ETL, we would extract the date parts and join them to the Date Dimension to avoid complex calculations in the query itself.

**Question 2C (1 Point extra credit):** Drill down, Drill across and rollup!

Modify the above query #2A so that it answers the following question:

Write and execute a query that that identifies the three highest units produced (passed) for each of the factories for each month in 2022.

Your result set should have the following four fields.  It might be helpful to use a window function to show three highest units produced (passed) for each of the factories for each month.  You will want to filter the data for 2022.

* Factory name
* Month – Notice that the month is not quite in readable format.  Transform the month into the following format '01-January’, ‘02-February', etc.  Hint – a case statement will be helpful here.
* Product Description - Notice that the product name contains an id.  Look into the left, and charindex(sql server)/position (postgre) functions to do a bit of string parsing.  An example format would be 'Camcorders','DVD', etc
* Total Units produced (passed) for each factory for each month and for each product
* Total Units failed for each factory for each month and for each product

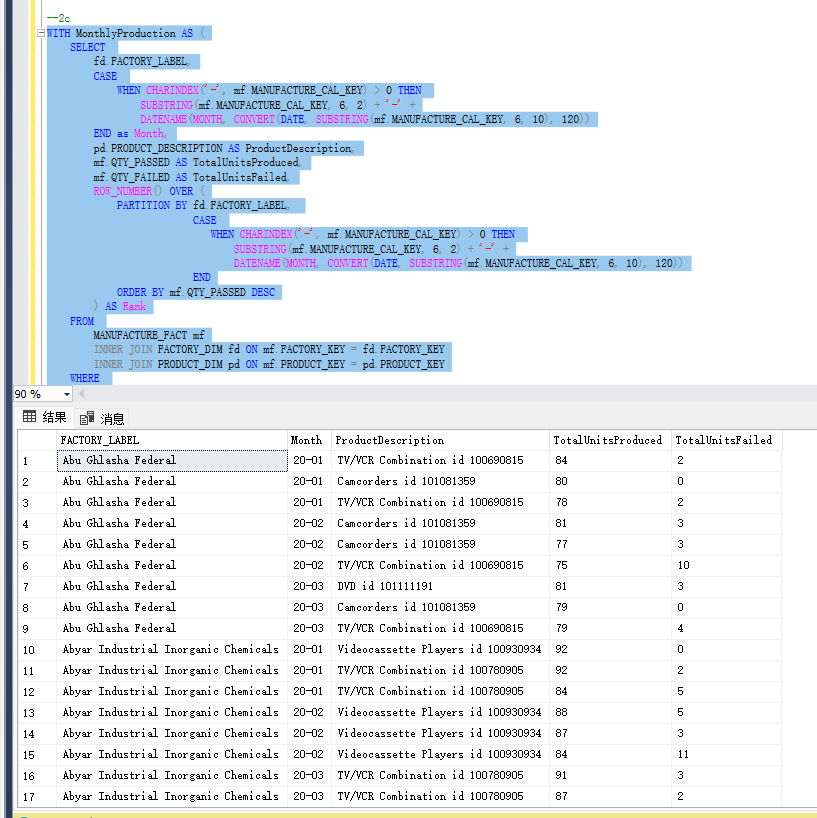
Complete the result by adding a Rollup to show subtotals by factory and month. With the Rollup function, you will notice a NULL value for the month and product columns for the subtotal on each of the factories.

**Paste the SQL code and a screen shot of the results.**

**SQL Code:**

|  |
| --- |
| WITH MonthlyProduction AS (  SELECT  fd.FACTORY\_LABEL,  CASE  WHEN CHARINDEX('-', mf.MANUFACTURE\_CAL\_KEY) > 0 THEN  SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 2) + '-' +  DATENAME(MONTH, CONVERT(DATE, SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 10), 120))  END as Month,  pd.PRODUCT\_DESCRIPTION AS ProductDescription,  mf.QTY\_PASSED AS TotalUnitsProduced,  mf.QTY\_FAILED AS TotalUnitsFailed,  ROW\_NUMBER() OVER (  PARTITION BY fd.FACTORY\_LABEL,  CASE  WHEN CHARINDEX('-', mf.MANUFACTURE\_CAL\_KEY) > 0 THEN  SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 2) + '-' +  DATENAME(MONTH, CONVERT(DATE, SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 10), 120))  END  ORDER BY mf.QTY\_PASSED DESC  ) AS Rank  FROM  MANUFACTURE\_FACT mf  INNER JOIN FACTORY\_DIM fd ON mf.FACTORY\_KEY = fd.FACTORY\_KEY  INNER JOIN PRODUCT\_DIM pd ON mf.PRODUCT\_KEY = pd.PRODUCT\_KEY  WHERE  CAST(SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 4) AS INT) = 2022  )  SELECT  FACTORY\_LABEL,  Month,  ProductDescription,  TotalUnitsProduced,  TotalUnitsFailed  FROM  MonthlyProduction  WHERE  Rank <= 3  ORDER BY  FACTORY\_LABEL,  Month,  Rank; |

**Screenshot of result:**



1. Drill down and rollup! Modify the above query in question 2 (the original query) to now drill down to the brand each of the factories and months within the year you selected. Use ROLLUP to show subtotals by factory, month, and brand. Your output should have these columns:

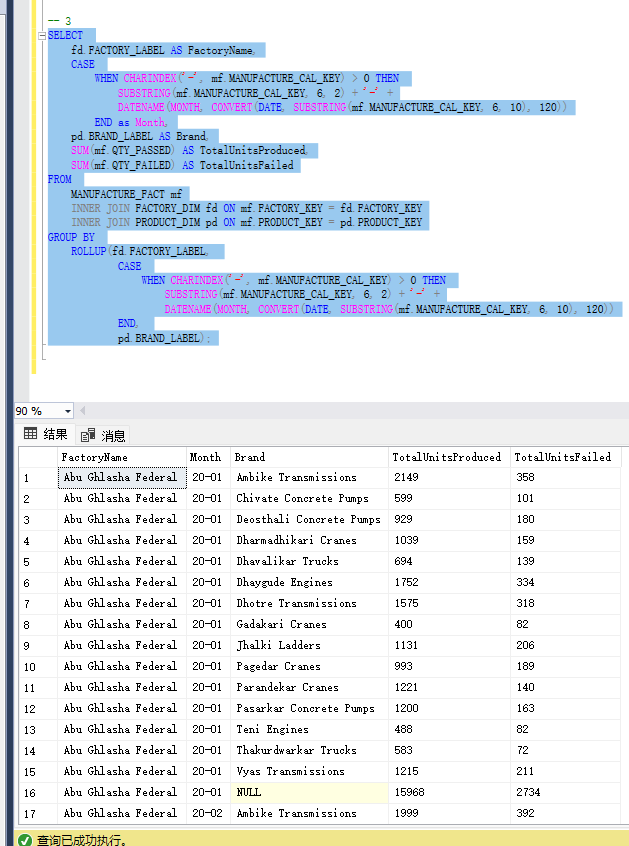
* Factory name
* Month
* Brand (brand label)
* Total Units produced (passed) for each factory for each month, for each brand
* Total Units failed for each factory for each month, for each brand

**Paste the SQL code and a screen shot of the results.**

**SQL Code:**

|  |
| --- |
| SELECT  fd.FACTORY\_LABEL AS FactoryName,  CASE  WHEN CHARINDEX('-', mf.MANUFACTURE\_CAL\_KEY) > 0 THEN  SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 2) + '-' +  DATENAME(MONTH, CONVERT(DATE, SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 10), 120))  END as Month,  pd.BRAND\_LABEL AS Brand,  SUM(mf.QTY\_PASSED) AS TotalUnitsProduced,  SUM(mf.QTY\_FAILED) AS TotalUnitsFailed  FROM  MANUFACTURE\_FACT mf  INNER JOIN FACTORY\_DIM fd ON mf.FACTORY\_KEY = fd.FACTORY\_KEY  INNER JOIN PRODUCT\_DIM pd ON mf.PRODUCT\_KEY = pd.PRODUCT\_KEY  GROUP BY  ROLLUP(fd.FACTORY\_LABEL,  CASE  WHEN CHARINDEX('-', mf.MANUFACTURE\_CAL\_KEY) > 0 THEN  SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 2) + '-' +  DATENAME(MONTH, CONVERT(DATE, SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 10), 120))  END,  pd.BRAND\_LABEL); |

**Screenshot of result:**



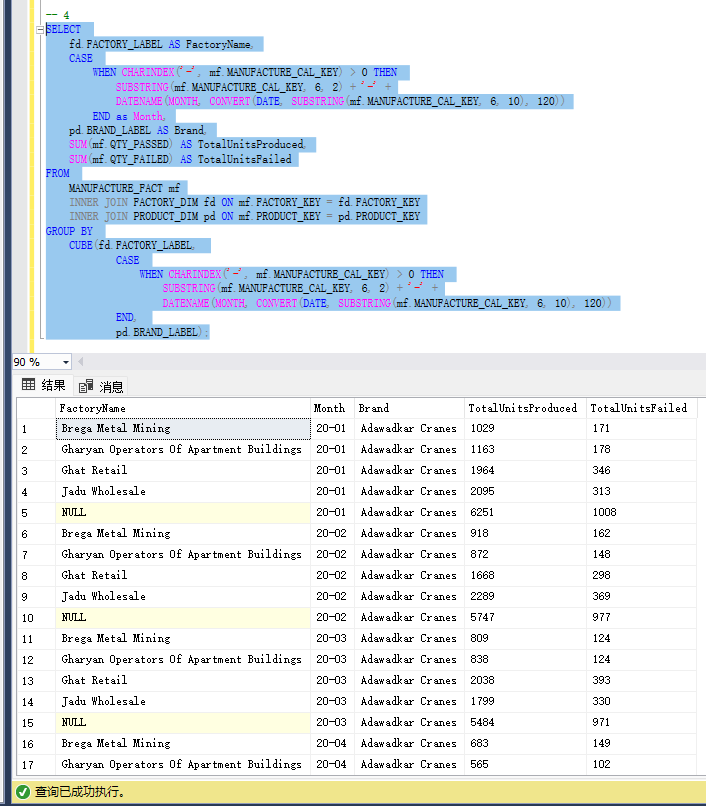
1. Drill down and rollup! Modify the above query in question 3 to use CUBE instead

**Paste the SQL code and a screen shot of the results.**

**SQL Code:**

|  |
| --- |
| SELECT  fd.FACTORY\_LABEL AS FactoryName,  CASE  WHEN CHARINDEX('-', mf.MANUFACTURE\_CAL\_KEY) > 0 THEN  SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 2) + '-' +  DATENAME(MONTH, CONVERT(DATE, SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 10), 120))  END as Month,  pd.BRAND\_LABEL AS Brand,  SUM(mf.QTY\_PASSED) AS TotalUnitsProduced,  SUM(mf.QTY\_FAILED) AS TotalUnitsFailed  FROM  MANUFACTURE\_FACT mf  INNER JOIN FACTORY\_DIM fd ON mf.FACTORY\_KEY = fd.FACTORY\_KEY  INNER JOIN PRODUCT\_DIM pd ON mf.PRODUCT\_KEY = pd.PRODUCT\_KEY  GROUP BY  CUBE(fd.FACTORY\_LABEL,  CASE  WHEN CHARINDEX('-', mf.MANUFACTURE\_CAL\_KEY) > 0 THEN  SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 2) + '-' +  DATENAME(MONTH, CONVERT(DATE, SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, 6, 10), 120))  END,  pd.BRAND\_LABEL); |

**Screenshot of result:**



1. Briefly explain the difference you noticed in results between rollup and cube of your results.

**Your Response:**

ROLLUP creates subtotals that roll up from the most detailed level to the grand total, incrementally adding one group at a time.

CUBE generates subtotals for all combinations of grouped columns, providing a full set of subtotals and the grand total.

1. Reuse the code from your query in question 1 to create the following data set which we will turn into a PIVOT/Crosstab in question 7. The base query will have the following three columns:
   * Year
   * Factory name (label)
   * Quantity passed

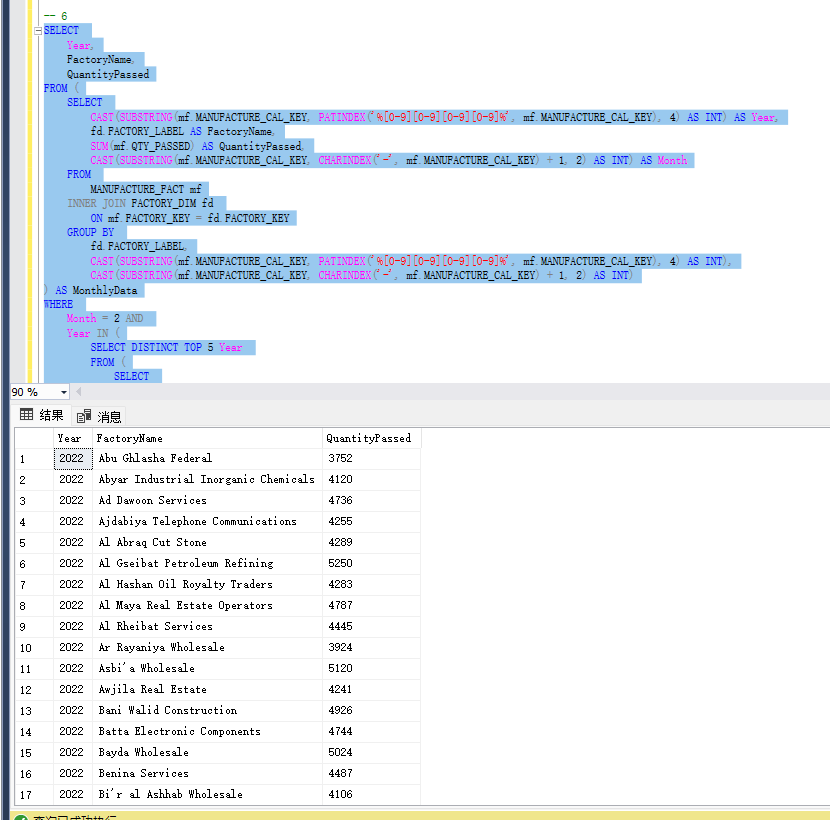
Filter this query to the month of February for the most current five years in the data set (the result will only contain data for February for five latest years)

**Paste the SQL code and a screen shot of the results.**

**SQL Code:**

|  |
| --- |
| SELECT  Year,  FactoryName,  QuantityPassed  FROM (  SELECT  CAST(SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, PATINDEX('%[0-9][0-9][0-9][0-9]%', mf.MANUFACTURE\_CAL\_KEY), 4) AS INT) AS Year,  fd.FACTORY\_LABEL AS FactoryName,  SUM(mf.QTY\_PASSED) AS QuantityPassed,  CAST(SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, CHARINDEX('-', mf.MANUFACTURE\_CAL\_KEY) + 1, 2) AS INT) AS Month  FROM  MANUFACTURE\_FACT mf  INNER JOIN FACTORY\_DIM fd  ON mf.FACTORY\_KEY = fd.FACTORY\_KEY  GROUP BY  fd.FACTORY\_LABEL,  CAST(SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, PATINDEX('%[0-9][0-9][0-9][0-9]%', mf.MANUFACTURE\_CAL\_KEY), 4) AS INT),  CAST(SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, CHARINDEX('-', mf.MANUFACTURE\_CAL\_KEY) + 1, 2) AS INT)  ) AS MonthlyData  WHERE  Month = 2 AND  Year IN (  SELECT DISTINCT TOP 5 Year  FROM (  SELECT  CAST(SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, PATINDEX('%[0-9][0-9][0-9][0-9]%', mf.MANUFACTURE\_CAL\_KEY), 4) AS INT) AS Year  FROM MANUFACTURE\_FACT mf  ) AS YearsData  ORDER BY Year DESC  )  ORDER BY  Year DESC,  FactoryName; |

**Screenshot of result:**



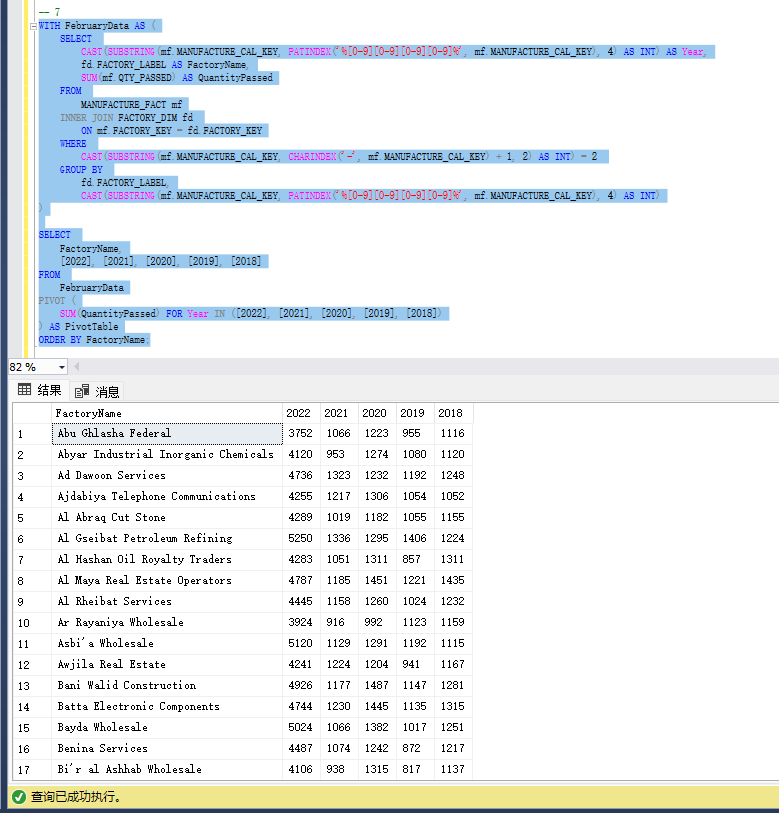
1. Crosstab/PIVOT. Use SQL Server PIVOT or PostgreSQL crosstab to create a table based on question 6
   * Year as column headings
   * Factory Label as rows
   * Quantity Passed as the data in the body of the table.

**Paste the code and a screen shot of the results.**

**SQL Code:**

|  |
| --- |
| WITH FebruaryData AS (  SELECT  CAST(SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, PATINDEX('%[0-9][0-9][0-9][0-9]%', mf.MANUFACTURE\_CAL\_KEY), 4) AS INT) AS Year,  fd.FACTORY\_LABEL AS FactoryName,  SUM(mf.QTY\_PASSED) AS QuantityPassed  FROM  MANUFACTURE\_FACT mf  INNER JOIN FACTORY\_DIM fd  ON mf.FACTORY\_KEY = fd.FACTORY\_KEY  WHERE  CAST(SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, CHARINDEX('-', mf.MANUFACTURE\_CAL\_KEY) + 1, 2) AS INT) = 2  GROUP BY  fd.FACTORY\_LABEL,  CAST(SUBSTRING(mf.MANUFACTURE\_CAL\_KEY, PATINDEX('%[0-9][0-9][0-9][0-9]%', mf.MANUFACTURE\_CAL\_KEY), 4) AS INT)  )  SELECT  FactoryName,  [2022], [2021], [2020], [2019], [2018]  FROM  FebruaryData  PIVOT (  SUM(QuantityPassed) FOR Year IN ([2022], [2021], [2020], [2019], [2018])  ) AS PivotTable  ORDER BY FactoryName; |

**Screenshot of result:**

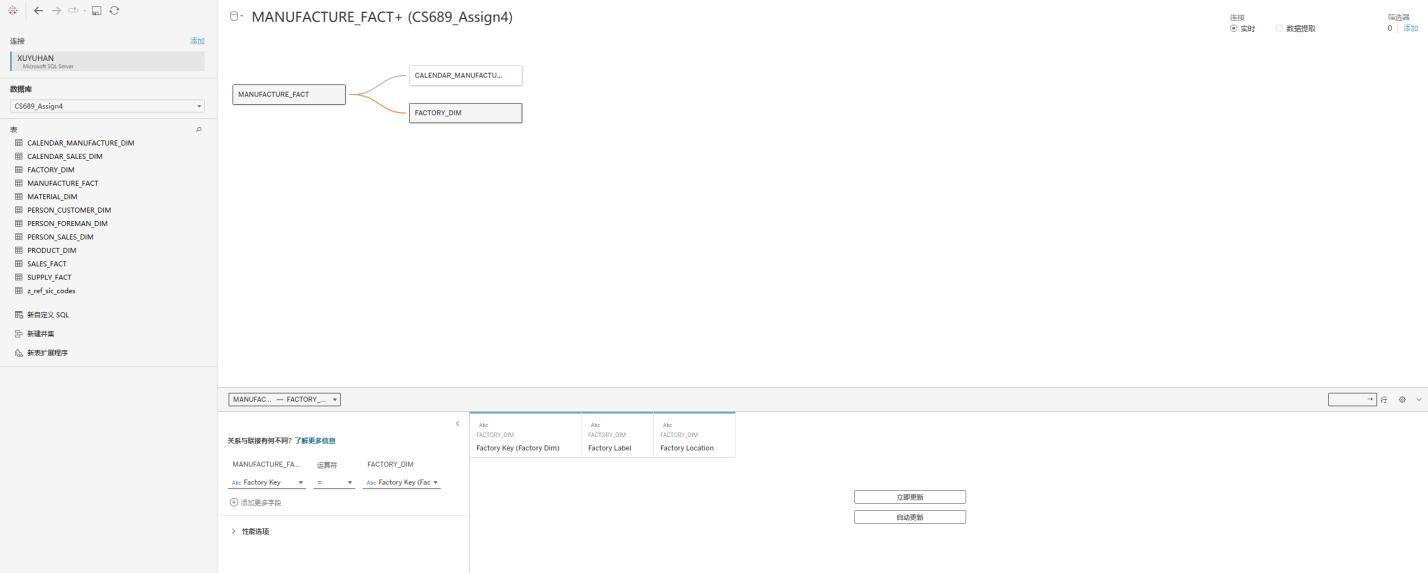


# Part 4 – Tableau Data Presentation

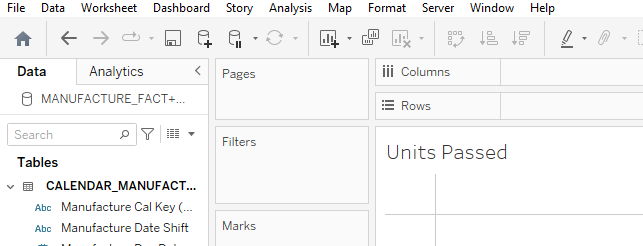
In this section, you will be working in at least two Tableau workbooks.

1. Download and install Tableau. You can get a student version here: <https://www.tableau.com/academic/students>. Note: it make a take few days to get a license. Alternatively, you can download a 14-day free trial version while you wait for your student license: <https://www.tableau.com/products/trial>
2. Start Tableau and connect it to your new data warehouse database.
   * Under Connect to a Server, choose “more” and select your DBMS (SQL Server or PostgreSQL), you will need to use the same connection options as when you sign into your DBMS.
     + If you are using PostgreSQL on a Mac and having issues connecting, please see Appendix section of this assignment.
   * Select the database you have been using for this assignment.
3. You will now see a list of tables you are now familiar with. Drag the MANUFACTURE\_FACT, CALENDAR\_MANUFACTURE\_DIM, AND FACTORY\_DIM from the Tables list within the Data Source tab. (The "Drag Tables Here" is where you drag it to). You will see lines connecting the tree tables, these indicate the joins. Click on each line to review that Tableau has joined the tables correctly.

**Take Screenshot of the Data Source table**



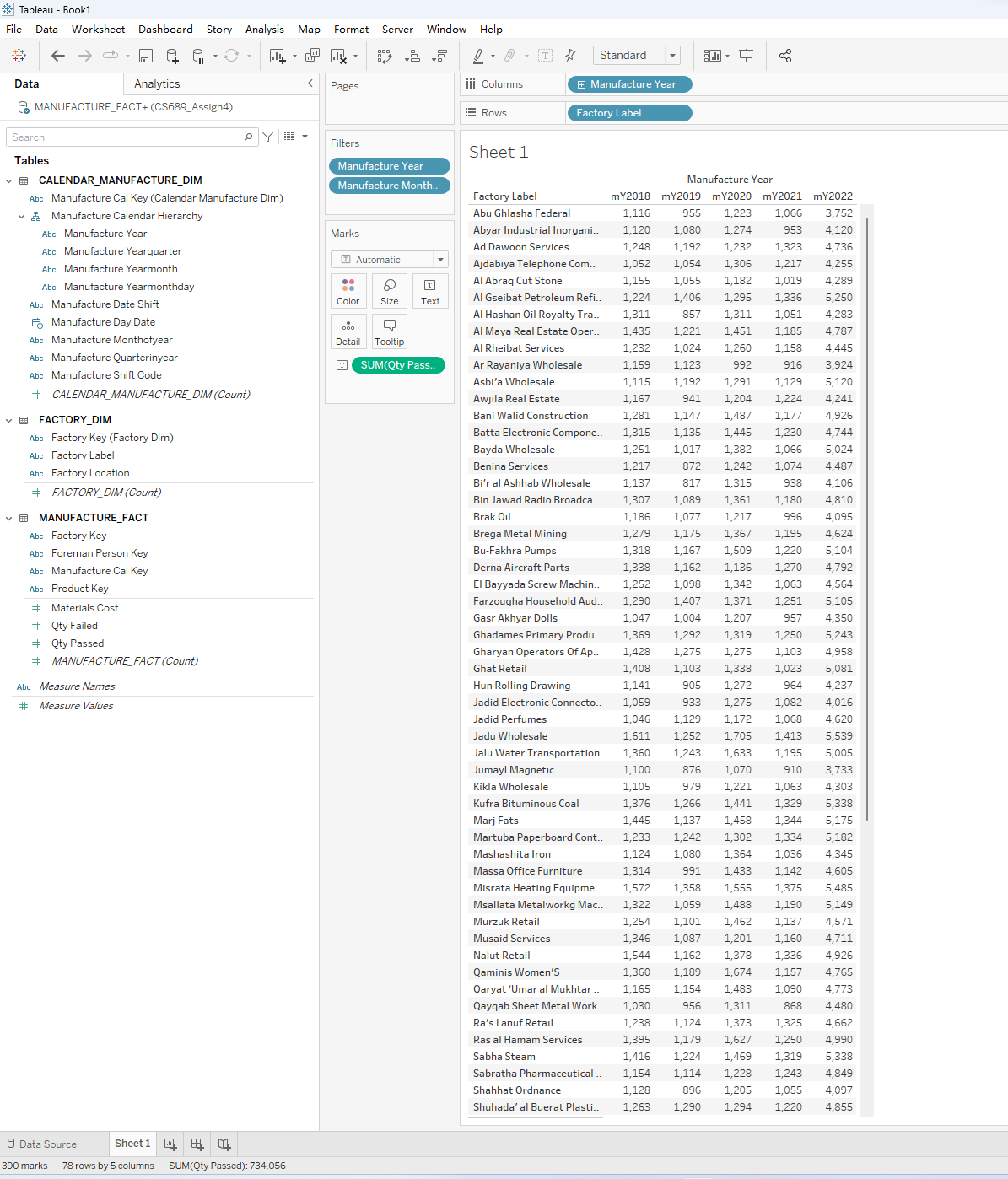
1. Sheet 1 will show in the bottom right corner, with “Go to Worksheet link”.
   * Click on Sheet 1 to open the worksheet.
   * Rename Sheet 1 worksheet to Units Passed (right click on it to show menu option to rename it)
   * Notice the three tables on the right, columns and rows area at the top.



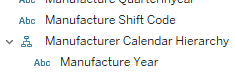
* + Place Manufacture Year (from the CALENDAR\_MANUFACTURE\_DIM table) as columns and Factory Label (from the FACTORY\_DIM table) as Rows. You will now see years as columns and factories as rows.
  + Under the Manufacture fact, drag the Qty Passed into the center of the pivot table (where you see the “abc” for each record). Under the Marks, Sum (Qty Pass will show)
  + Drag the Manufacture Year (from the CALENDAR\_MANUFACTURE\_DIM table) into the Filters box, in the General tab, select the latest 5 years
  + Drag the Manufacture Monthofyear (from the CALENDAR\_MANUFACTURE\_DIM table) into the Filters box, in the General tab, select February.

The pivot table we just created is the exact same one you solved in question 7 from part 3! You should see the same data.

**Take Screenshot of the worksheet showing the pivot table**

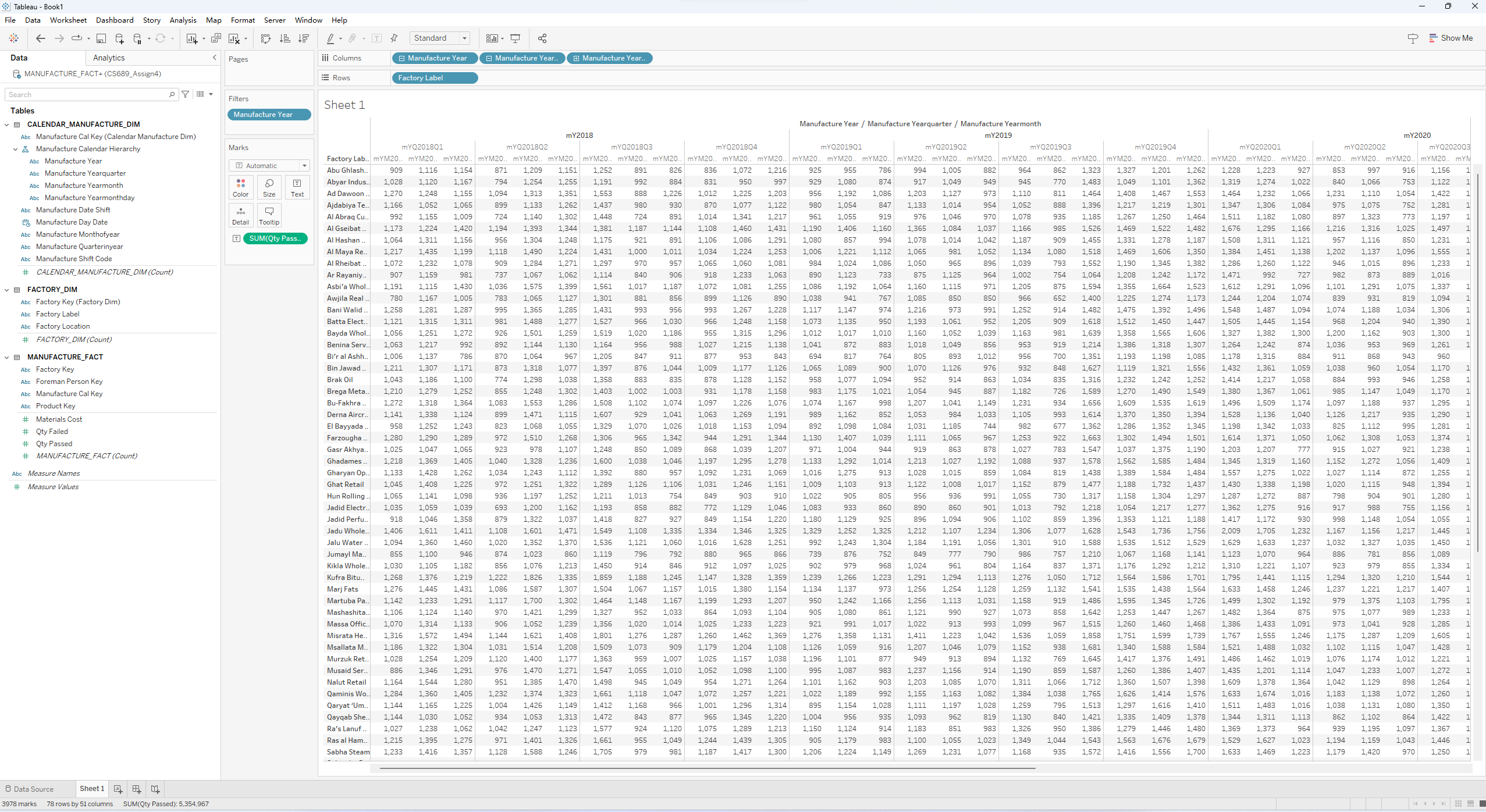


1. Create a Manufacture Calendar Hierarchy.
   * Under the CALENDAR\_MANUFACTURE\_DIM select the four attributes which represent the calendar hierarchy, right click on them and from the Hierarchy option select create Hierarchy. It should use four of the table’s fields with the following order – Year, Yearquarter, Yearmonth, Yearmonthdat. The hierarchy should increase in detail as you move down. Here is a hint to what it should look like (the first field is shown):

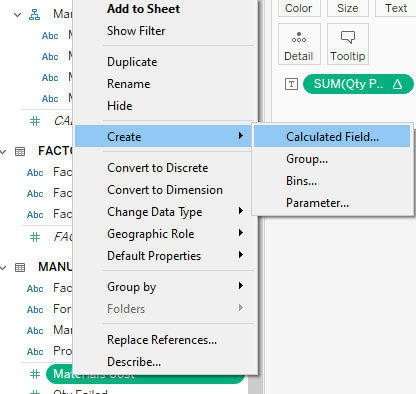


* Remove the Manufacture Year from the Columns at the top and replace it with the Manufacture Calendar Hierarchy you just created.
* You will see a plus next to the Manufacture Year in your column area, click on it to expand date hierarchy column to Months which will appear in your pivot table.
* Remove the February month filter, you should see an expanded workbook of the year and all the months under it.

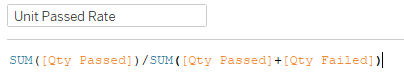
**Take Screenshot of the worksheet showing the pivot table including the Manufacture Calendar Hierarchy in the tables area on the left**



1. Let’s create a calculated measure called Unit Passed Rate.
   * First copy the worksheet you have been using 
   * Right-click in the Measures section at lower left (on any of the existing measures) and select Create to add a calculated field called Unit Passed Rate.



* + The formula should divide the sum of quantity passed by the total quantities (passed and failed) summed.

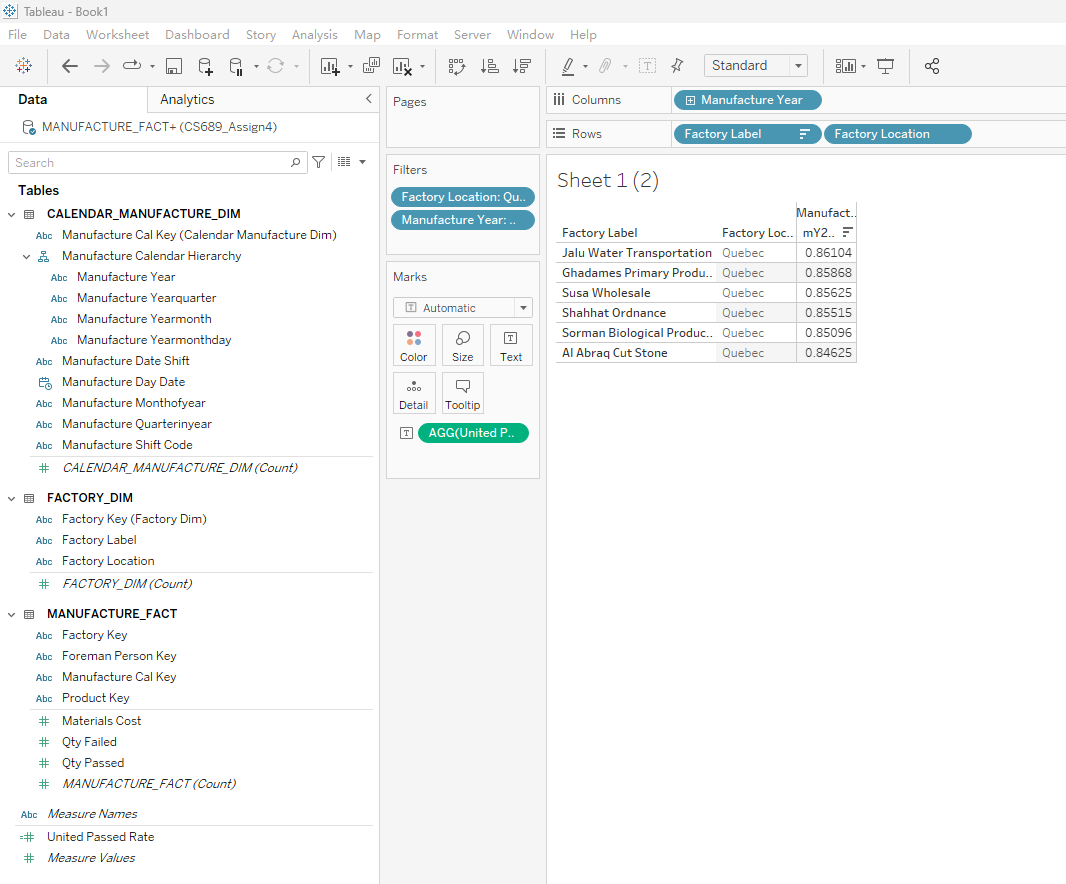


* + Replace the Qty Passed sum measure with Unit Passed Rate calculated measure in the pivot, and in addition add Factory location as the first hierarchy in the rows.

**What factory in Quebec has the highest unit pass rate for 2022?**

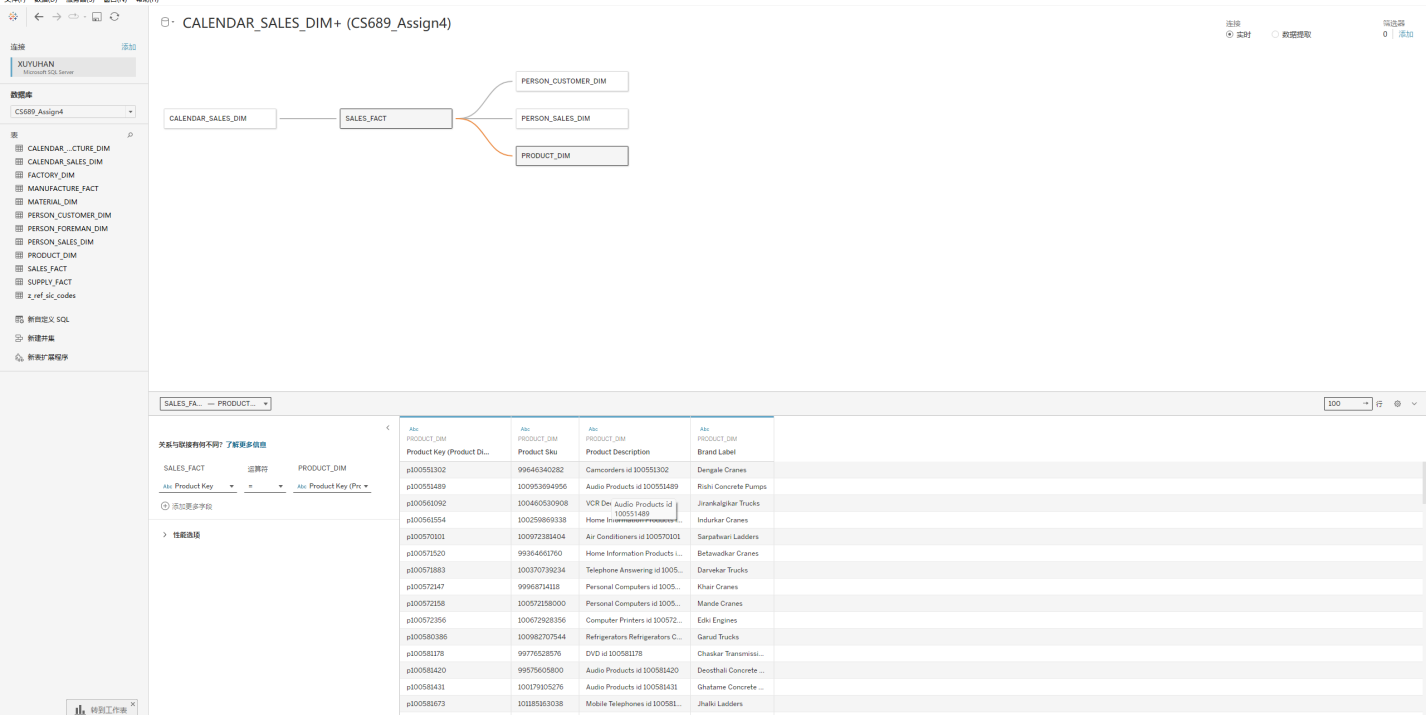
Jalu Water Transportation, Quebec has the highest unit pass rate for 2022

**Take a screenshot showing Quebec and its factories along with the unit pass rate**



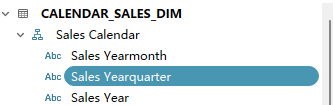
1. Create a New Tableau Workbook for Sales and set up a Data Source for Sales Facts.
   * You will need CALENDAR\_SALES\_DIM, SALES\_FACT, PERSON\_CUSTOMER\_DIM, PERSON\_SALES\_DIM, AND PRODUCT\_DIM. When adding the two person DIM tables, review the key relationships as the names are not consistent, match the salesperson key and customer person key with the appropriate person key in both of the DIM tables.

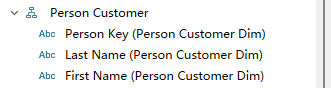
**Take a screenshot of the Sales Data Source**



1. Set up hierarchies.
   * Click into "Sheet 1" at the bottom to start a worksheet. You will see a completely empty sheet, with dimensions and measures.
   * Review all the dimension tables and set up hierarchies which make sense to you. Two hierarchies in total should be fine here.

**Provide screenshots of the two hierarchies**



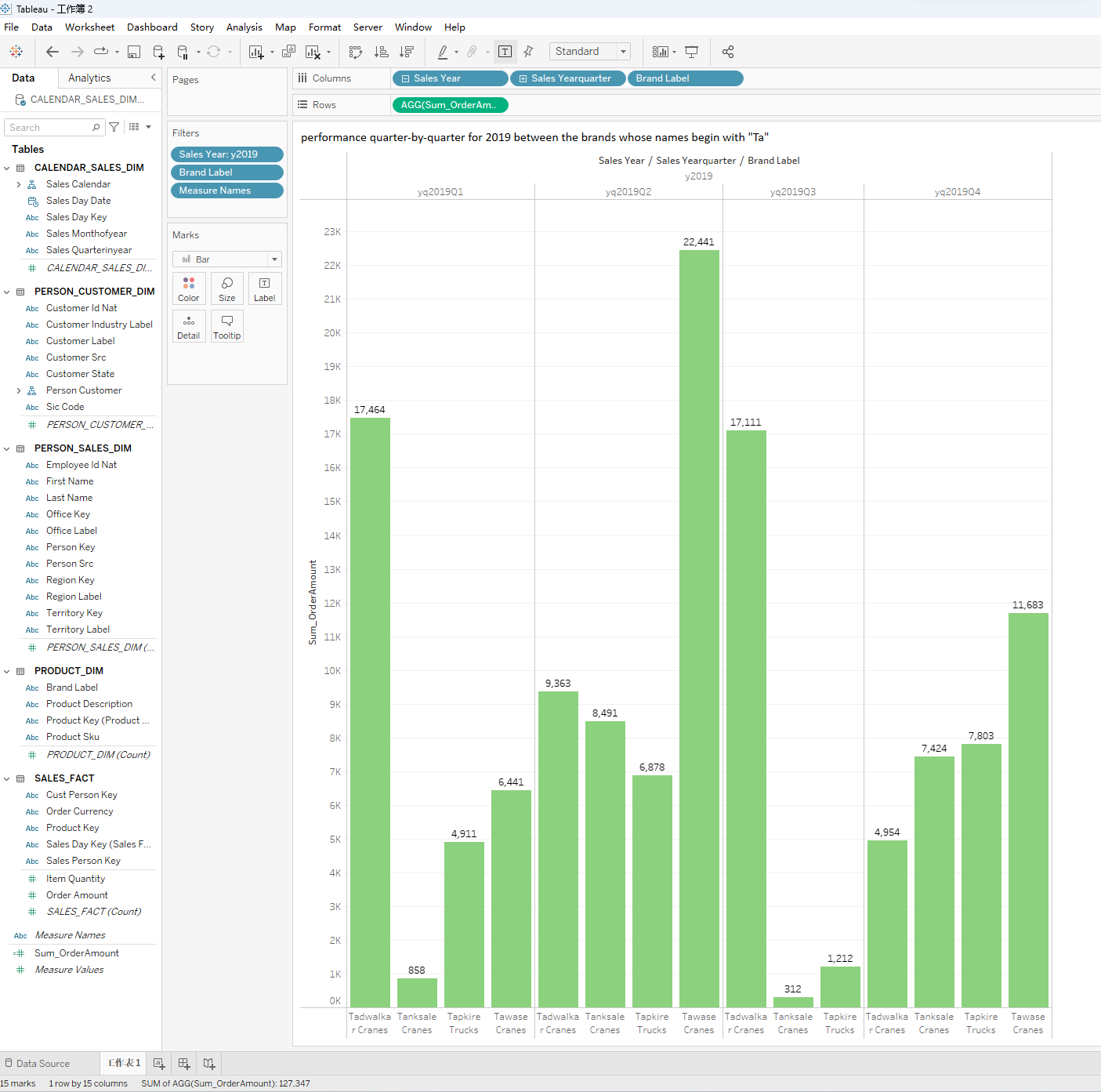


1. Create a Bar Chart for Sales Revenue

We want to compare performance quarter-by-quarter for 2019 between the brands whose names begin with "Ta". As you complete each of these bullet points, the bar chart will take shape.

* Add the date hierarchy as a column and expand to quarters.
* Rows don’t just have to be dimensions, we can add a measure for the sum of order amounts from the sales fact here.
* Add a filter from the calendar dimension for 2019.
* Add Brand Label from the Products dimension to the columns
* Add a filter for brand label, selecting only the brands beginning with "Ta", there should be 4 product labels that being with “Ta”
* Your bar chart is complete, however let’s add color! Select use the “Side-by-Side” bars. If you recall we are comparing products by a single variable, and side-by-side chart works great here!
* In the Marks click on the Label, and check the box to Show mark labels so that the amounts are shown.
* In the columns area, you can drag the Sales Yearquarter and the Brand label to have them switch spaces, note how the chart will re-organize.

**Provide screenshot of your chart, including all the filters, columns, rows, etc.**



**Which brand had the worst quarter in 2019 and what was the order amount?**

Tanksale Cranes had the worst quarter in 2019, the order amount is 0

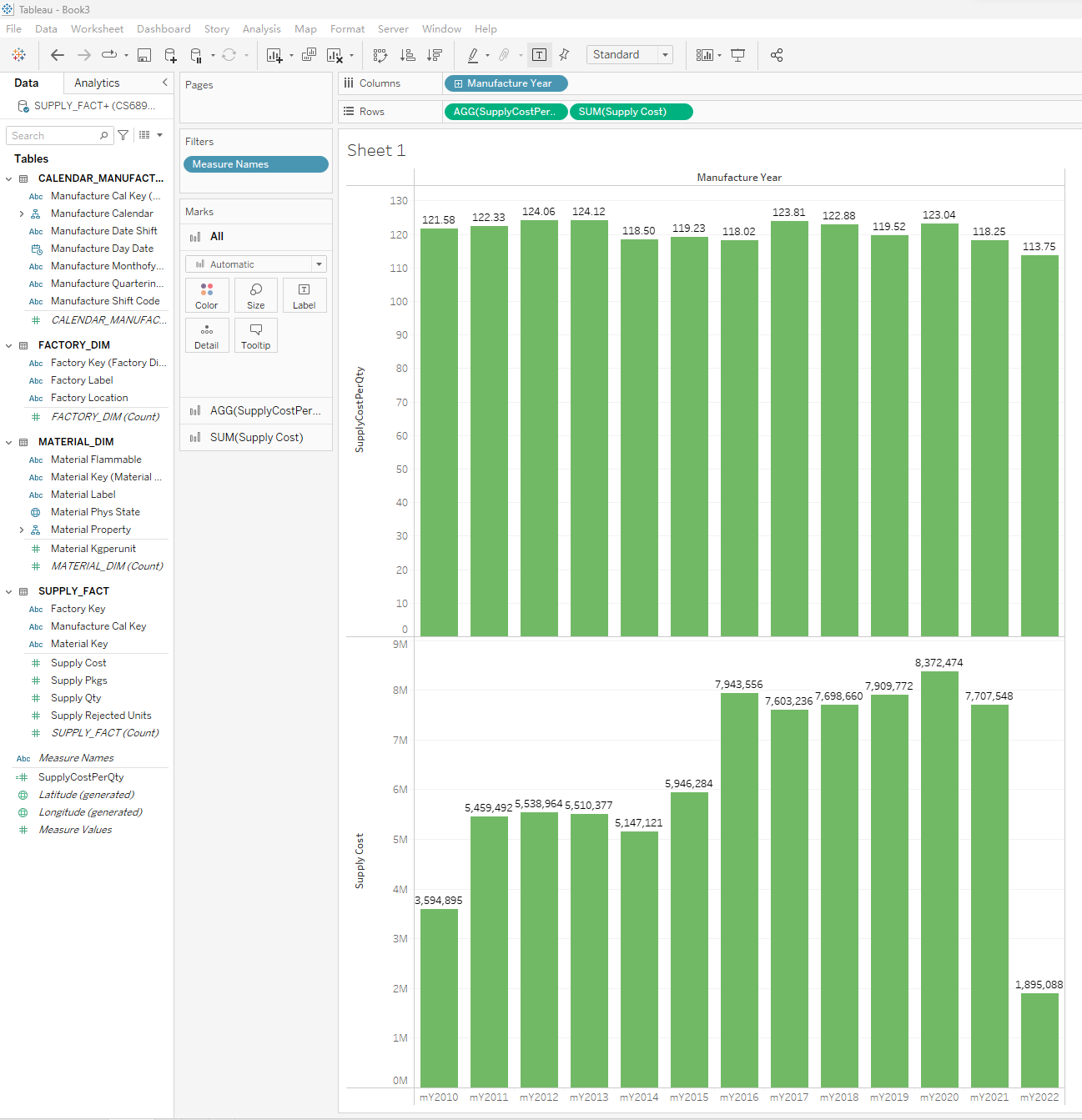
**Which brand had the best quarter in 2019 and what was the order amount?**

Tawase Cranes had the best quarter in 2019, the order amount is 22441

# Part 5 Extra Credit

Define and implement a visualization of your choice. The more complexity, the more extra credit (1 to 3 extra credit points). Look to experiment with different types charts depending on the number of variables, incorporate hierarchies, calculated measures, etc. Define the question and the provide the screenshots of your visualization.

**Provide screenshot of your chart, including all the filters, columns, rows, etc.**



In the graph above, we can get that it has highest supply Cost in 2020, lowest Supply Cost in 2022, and It has the highest SupplyCostPerQty in 2013, lowest in 2022

# Appendix

**Tableau and PostgreSQL connection on Mac**

If you are having an issue with Tableau unable to connect to PostgreSQL on your mac, the following steps might help:

1. Click into this link https://www.tableau.com/support/drivers and find PostgreSQL section, download the jar file.
2. Move the jar file to 'Library/Tableau/Drivers'

If you're unable to locate Tableau folder in Library, try this method: open any folder or turn on 'Finder' on your mac. Find the 'Go' tab from the top bar, click 'Go to Folder’

Then enter '~/Library' in the search bar, then press 'return' key. You should be able to find Tableau folder in this directory.

You should now be able to connect Tableau with your PostgreSQL.

Solution reference: <https://community.tableau.com/s/question/0D54T00000dYsbrSAC/postgressql-connection-failed>

Use the **Ask the Teaching Team Discussion Forum** if you have any questions regarding the how to approach this assignment.

Save your assignment as ***lastnameFirstname\_assign4\_0.docx*** and submit it in the *Assignments* section of the course.

For help uploading files please refer to the *Technical Support* page in the syllabus.

