# 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 count, getdate() as date, 'Varun Nair' as name from MANUFACTURE\_FACT

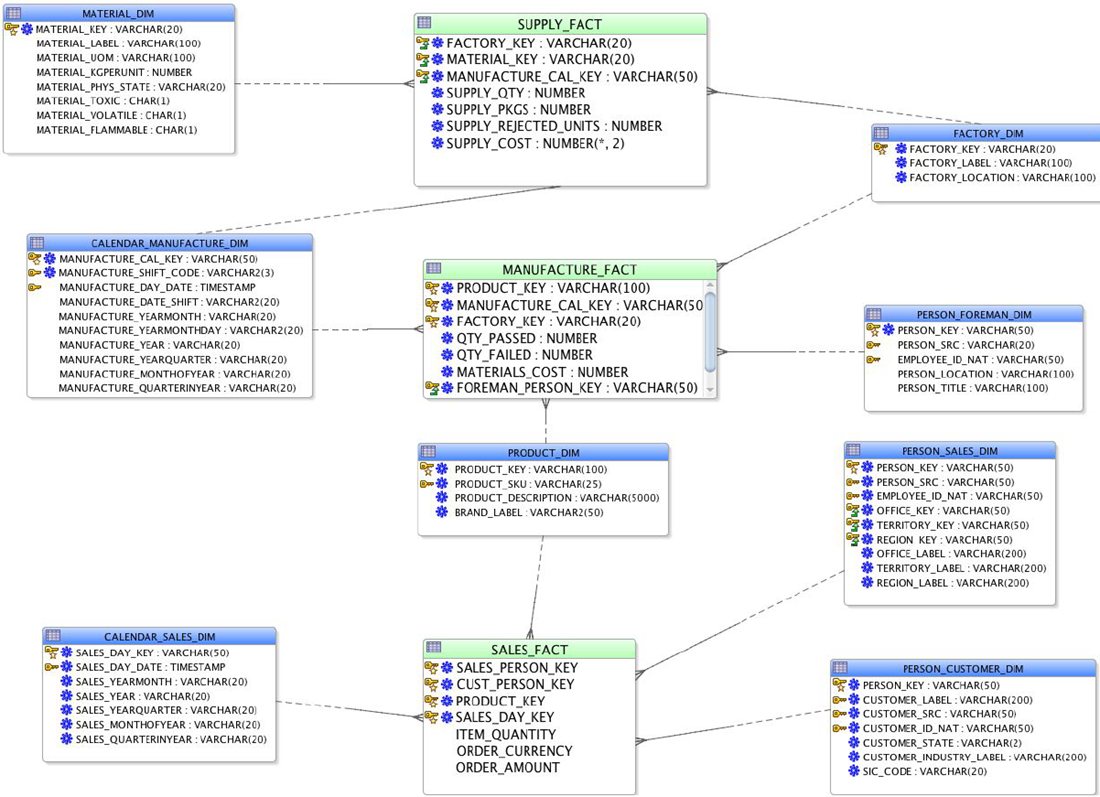
Screenshot of result:

A screenshot of a phone number

Description automatically generated

**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:** CALENDAR\_MANUFACTURE\_DIM : MANUFACTURE\_DAY\_DATE,MANUFACTURE\_QUATERINYEAR,MANUFACTURE\_YEARQUARTER,MANUFACTURE\_MONTHOFYEAR,MANUFACTURE\_YEAR,MANUFACTURE\_YEARMONTH,MANUFACTURE\_YEARMONTHDAY

**Dimension 2:** PERSON\_SALES\_DIM : REGION\_LABEL,REGION\_KEY,TERRITORY\_LABEL,TERRITORY\_KEY,OFFICE\_LABEL,OFFICE\_KEY

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:**

PERSON\_SALES\_DIM can have additional columns like START\_DATE and END\_DATE along with an IsActive flag highlighting the current assignment. Along with this, there can be additional columns detailing each salespersons previous locations (PREV\_OFFICE\_LABEL,PREV\_OFFICE\_KEY)-something to keep in mind that PREV\_OFFICE\_KEY & PREV\_OFFICE\_LABEL will be null for new salespeople and each salesperson’s first assignment AKA when they joined the company, thus making it a SCD 2 and SCD 3 Dimension

For maintenance, if a salesperson changes his location, then the current OFFICE\_LABEL and OFFICE\_KEY will be the PREV\_OFFICE\_LABEL and PREV\_OFFICE\_KEY in the salespersons new record. To make this record further accurate, we can update the start date as the date of when the salesperson join the new location. The end date for their previous assignment will be whenever they have left that location.

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

MATERIAL\_DIM can have a START\_DATE and END\_DATE column talking about whether the material is being provided or not. An IsActive flag can highlight all the materials that are currently being provided. In the future, if certain materials are unavailable, having a record of those materials would be helpful. This would make it a SCD Type 2

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

Fact table: SALES\_FACT

Measures : ITEM\_QUANTITY,ORDER\_AMOUNT,ORDER\_CURRENCY

# 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:

select

YearlyProduction.Year,

YearlyProduction.FactoryLabel,

YearlyProduction.QtyPassed as TotalUnitsPassed,

YearlyProduction.QtyFailed as TotalUnitsFailed,

YearlyProduction.FactoryRank from(

select

c.MANUFACTURE\_YEAR as Year,

f.FACTORY\_LABEL AS FactoryLabel,

sum(m.QTY\_PASSED) as QtyPassed,

sum(m.QTY\_FAILED) as QtyFailed,

ROW\_NUMBER() over (partition by c.MANUFACTURE\_YEAR order by sum(m.QTY\_PASSED) desc) as FactoryRank

from

MANUFACTURE\_FACT m

JOIN

CALENDAR\_MANUFACTURE\_DIM c ON m.MANUFACTURE\_CAL\_KEY = c.MANUFACTURE\_CAL\_KEY

join

FACTORY\_DIM f ON m.FACTORY\_KEY = f.FACTORY\_KEY

GROUP BY

C.MANUFACTURE\_YEAR,f.FACTORY\_LABEL

) AS YearlyProduction where YearlyProduction.FactoryRank <= 3 order by YearlyProduction.Year DESC,YearlyProduction.FactoryRank

**Screenshot of result:**

A screenshot of a computer

Description automatically generated

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:**

WITH MonthlyData AS (

SELECT

f.FACTORY\_LABEL AS FactoryName,

CONCAT(

FORMAT(MONTH(c.MANUFACTURE\_DAY\_DATE), '00'), '-',

CASE MONTH(c.MANUFACTURE\_DAY\_DATE)

WHEN 1 THEN 'January'

WHEN 2 THEN 'February'

WHEN 3 THEN 'March'

WHEN 4 THEN 'April'

WHEN 5 THEN 'May'

WHEN 6 THEN 'June'

WHEN 7 THEN 'July'

WHEN 8 THEN 'August'

WHEN 9 THEN 'September'

WHEN 10 THEN 'October'

WHEN 11 THEN 'November'

WHEN 12 THEN 'December'

END

) AS Month,

MONTH(c.MANUFACTURE\_DAY\_DATE) AS MonthSort,

SUM(m.QTY\_PASSED) AS TotalUnitsPassed,

SUM(m.QTY\_FAILED) AS TotalUnitsFailed

FROM

MANUFACTURE\_FACT m

JOIN

CALENDAR\_MANUFACTURE\_DIM c ON m.MANUFACTURE\_CAL\_KEY = c.MANUFACTURE\_CAL\_KEY

JOIN

FACTORY\_DIM f ON m.FACTORY\_KEY = f.FACTORY\_KEY

WHERE

YEAR(c.MANUFACTURE\_DAY\_DATE) = 2022

GROUP BY

f.FACTORY\_LABEL,

MONTH(c.MANUFACTURE\_DAY\_DATE)

)

SELECT

FactoryName,

Month,

TotalUnitsPassed,

TotalUnitsFailed

FROM MonthlyData

GROUP BY

FactoryName,

Month,

MonthSort,

TotalUnitsPassed,

TotalUnitsFailed

WITH ROLLUP

HAVING

(Month IS NOT NULL OR FactoryName IS NOT NULL)

ORDER BY

COALESCE(FactoryName, 'ZZZZ'),

CASE

WHEN Month IS NULL THEN 'ZZZZ'

ELSE CAST(MonthSort AS VARCHAR(2))

END;

Screenshot of result:

A screenshot of a computer

Description automatically generated

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

**Short Answer:**

There can be a separate column for month which only has integer values of the month in the Calendar dimension. This will skip the need of any case statements where we needed to check the for the month number and then assign it to a number.

The month can be extracted from the date via the month() function and can be inserted into the new month\_number column

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:**

WITH MonthlyData AS (

SELECT

f.FACTORY\_LABEL AS FactoryName,

CONCAT(

FORMAT(MONTH(c.MANUFACTURE\_DAY\_DATE), '00'), '-',

CASE MONTH(c.MANUFACTURE\_DAY\_DATE)

WHEN 1 THEN 'January'

WHEN 2 THEN 'February'

WHEN 3 THEN 'March'

WHEN 4 THEN 'April'

WHEN 5 THEN 'May'

WHEN 6 THEN 'June'

WHEN 7 THEN 'July'

WHEN 8 THEN 'August'

WHEN 9 THEN 'September'

WHEN 10 THEN 'October'

WHEN 11 THEN 'November'

WHEN 12 THEN 'December'

END

) AS Month,

MONTH(c.MANUFACTURE\_DAY\_DATE) as MonthSort,

p.BRAND\_LABEL as Brand,

sum(m.QTY\_PASSED) as TotalUnitsPassed,

sum(QTY\_FAILED) as TotalUnitsFailed

from MANUFACTURE\_FACT m

JOIN CALENDAR\_MANUFACTURE\_DIM c ON m.MANUFACTURE\_CAL\_KEY = c.MANUFACTURE\_CAL\_KEY

JOIN FACTORY\_DIM f ON m.FACTORY\_KEY = f.FACTORY\_KEY

JOIN PRODUCT\_DIM p ON M.PRODUCT\_KEY = P.PRODUCT\_KEY

WHERE YEAR(c.MANUFACTURE\_DAY\_DATE)=2022

group by f.FACTORY\_LABEL,MONTH(c.MANUFACTURE\_DAY\_DATE),p.BRAND\_LABEL

)

select FactoryName,Month,Brand,TotalUnitsPassed,TotalUnitsFailed from MonthlyData

group by FactoryName,Month,Brand,MonthSort,TotalUnitsPassed,TotalUnitsFailed

with rollup having (Month is not null or FactoryName is not null or Brand is not null)

order by coalesce(FactoryName,'ZZZZ'),

CASE

WHEN Month is null then 'ZZZZ'

ELSE CAST(MonthSort as VARCHAR(2))

END,

COALESCE(Brand,'ZZZZ')

Screenshot of result:

A screenshot of a computer

Description automatically generated

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:**

WITH MonthlyData AS (

SELECT

f.FACTORY\_LABEL AS FactoryName,

CONCAT(

FORMAT(MONTH(c.MANUFACTURE\_DAY\_DATE), '00'), '-',

CASE MONTH(c.MANUFACTURE\_DAY\_DATE)

WHEN 1 THEN 'January'

WHEN 2 THEN 'February'

WHEN 3 THEN 'March'

WHEN 4 THEN 'April'

WHEN 5 THEN 'May'

WHEN 6 THEN 'June'

WHEN 7 THEN 'July'

WHEN 8 THEN 'August'

WHEN 9 THEN 'September'

WHEN 10 THEN 'October'

WHEN 11 THEN 'November'

WHEN 12 THEN 'December'

END

) AS Month,

MONTH(c.MANUFACTURE\_DAY\_DATE) as MonthSort,

p.BRAND\_LABEL as Brand,

sum(m.QTY\_PASSED) as TotalUnitsPassed,

sum(QTY\_FAILED) as TotalUnitsFailed

from MANUFACTURE\_FACT m

JOIN CALENDAR\_MANUFACTURE\_DIM c ON m.MANUFACTURE\_CAL\_KEY = c.MANUFACTURE\_CAL\_KEY

JOIN FACTORY\_DIM f ON m.FACTORY\_KEY = f.FACTORY\_KEY

JOIN PRODUCT\_DIM p ON M.PRODUCT\_KEY = P.PRODUCT\_KEY

WHERE YEAR(c.MANUFACTURE\_DAY\_DATE)=2022

group by f.FACTORY\_LABEL,MONTH(c.MANUFACTURE\_DAY\_DATE),p.BRAND\_LABEL

)

select FactoryName,Month,Brand,TotalUnitsPassed,TotalUnitsFailed from MonthlyData

group by FactoryName,Month,Brand,MonthSort,TotalUnitsPassed,TotalUnitsFailed

with CUBE having (Month is not null or FactoryName is not null or Brand is not null)

order by coalesce(FactoryName,'ZZZZ'),

CASE

WHEN Month is null then 'ZZZZ'

ELSE CAST(MonthSort as VARCHAR(2))

END,COALESCE(Brand,'ZZZZ')

Screenshot of result:

A screenshot of a computer

Description automatically generated

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

**Your Response:**

The cube had much more detailed rows in the result, with a lot more nulls than the rollup. The rollup had individual subtotals for both produced units and failed units for each month that it was present in(from Manufacture\_fact)

The cube query also listed out all brand labels for each factory first (with NULL values for both units in those rows) and then lists the subtotals for each kind of units, for each month.

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

YearlyProduction.Year,

YearlyProduction.FactoryLabel,

YearlyProduction.QtyPassed as TotalUnitsPassed FROM(

select

c.MANUFACTURE\_YEAR as Year,

f.FACTORY\_LABEL AS FactoryLabel,

sum(m.QTY\_PASSED) as QtyPassed,

ROW\_NUMBER() over (partition by c.MANUFACTURE\_YEAR order by sum(m.QTY\_PASSED) desc) as FactoryRank

from

MANUFACTURE\_FACT m

JOIN

CALENDAR\_MANUFACTURE\_DIM c ON m.MANUFACTURE\_CAL\_KEY = c.MANUFACTURE\_CAL\_KEY

join

FACTORY\_DIM f ON m.FACTORY\_KEY = f.FACTORY\_KEY

where month(c.MANUFACTURE\_DAY\_DATE) = 2 AND YEAR(c.MANUFACTURE\_DAY\_DATE) in(2022,2021,2020,2019,2018)

GROUP BY

C.MANUFACTURE\_YEAR,f.FACTORY\_LABEL

) AS YearlyProduction where YearlyProduction.FactoryRank <= 3 order by YearlyProduction.Year DESC,YearlyProduction.FactoryRank

Screenshot of result:

A screenshot of a computer

Description automatically generated

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:**

SELECT \*

FROM (

SELECT

YearlyProduction.Year,

YearlyProduction.FactoryLabel,

YearlyProduction.QtyPassed AS TotalUnitsPassed

FROM (

SELECT

YEAR(c.MANUFACTURE\_DAY\_DATE) AS Year,

f.FACTORY\_LABEL AS FactoryLabel,

SUM(m.QTY\_PASSED) AS QtyPassed,

SUM(m.QTY\_FAILED) AS QtyFailed,

ROW\_NUMBER() OVER (PARTITION BY YEAR(c.MANUFACTURE\_DAY\_DATE) ORDER BY SUM(m.QTY\_PASSED) DESC) AS FactoryRank

FROM

MANUFACTURE\_FACT m

JOIN

CALENDAR\_MANUFACTURE\_DIM c ON m.MANUFACTURE\_CAL\_KEY = c.MANUFACTURE\_CAL\_KEY

JOIN

FACTORY\_DIM f ON m.FACTORY\_KEY = f.FACTORY\_KEY

WHERE

YEAR(c.MANUFACTURE\_DAY\_DATE) IN (2022, 2021, 2020, 2019, 2018)

AND MONTH(c.MANUFACTURE\_DAY\_DATE) = 2

GROUP BY

YEAR(c.MANUFACTURE\_DAY\_DATE), f.FACTORY\_LABEL

) AS YearlyProduction

WHERE YearlyProduction.FactoryRank <= 3

) AS SourceTable

PIVOT (

SUM(TotalUnitsPassed)

FOR Year IN ([2022], [2021], [2020], [2019], [2018])

) AS PivotTable

ORDER BY FactoryLabel

Screenshot of result:

A screenshot of a computer

Description automatically generated

# Part 4 – Tableau Data Presentation

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

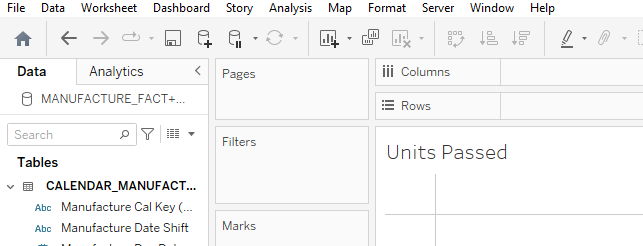
1. Download and install Tableau. You can get a student version here: <https://www.tableau.com/academic/students>. Note: it make a take few days to get a license. Alternatively, you can download a 14-day free trial version while you wait for your student license: <https://www.tableau.com/products/trial>
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

A screenshot of a computer

Description automatically generated

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.

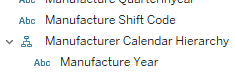
The pivot table we just created is the exact same one you solved in question 7 from part 3! You should see the same data.

A screenshot of a computer

Description automatically generated

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):**



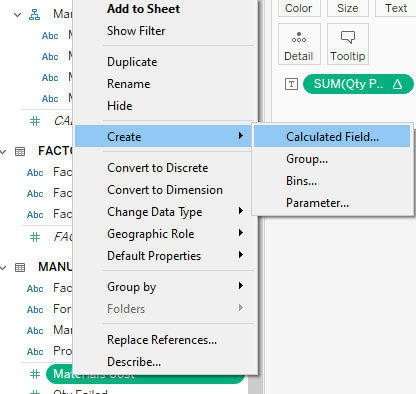
* Remove the Manufacture Year from the Columns at the top and replace it with the Manufacture Calendar Hierarchy you just created.
* You will see a plus next to the Manufacture Year in your column area, click on it to expand date hierarchy column to Months which will appear in your pivot table.
* Remove 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

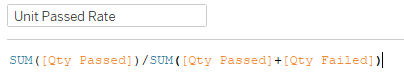
A screenshot of a computer

Description automatically generated

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.

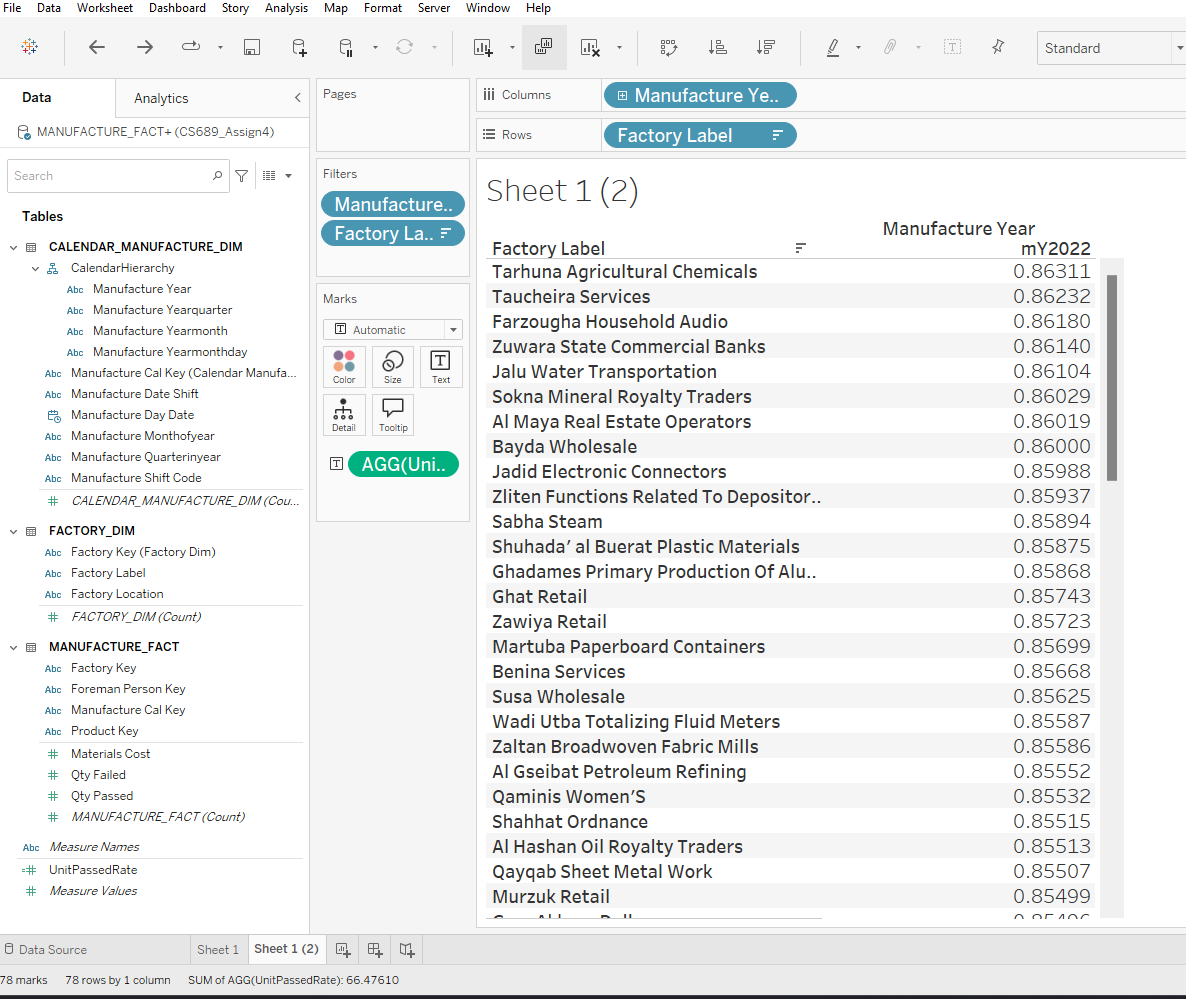


* + The formula should divide the sum of quantity passed by the total quantities (passed and failed) summed.



* + 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? - Tarhuna Agricultural Chemicals

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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.

A screenshot of a graph

Description automatically generated

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

TankSale Cranes in Q3

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

# Tawase Cranes in Q2

