

MET CS 689 Designing and Implementing a Data Warehouse

Assignment 4: Presenting Data from a Dimensional Data Warehouse



April 11, 2023

Vincent Nguyen

nvui@bu.edu

# 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 number\_records, 'VincentNguyen' as MyName, Current\_Date as Current\_Date**

**from manufacture\_fact;**

**Screenshot of result:**

# Graphical user interface, text, application 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\_sales\_dim**

**Sales\_year >> sales\_quarterinyear >> sales\_monthofyear >> sales\_day\_date**

**Dimension 2: Person\_sales\_dim**

**Region\_lable >> Territory\_lable >> office\_lable**

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.

* There is person\_foreman\_dim and factory\_dim would benefit from SCD2 and SCD3 implementation.

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

Dimension person\_foreman\_dim would benefit from SCD2 implementation because the person\_location and person\_title might change time to time when the person gets transfer or promotion, so it is important to track the historical changes. I would help to look back the employee career path.

We will add three more columns into person\_foreman\_dim such as effective\_timestamp (start date), expire\_timestamp (end date) and Current\_flag.

ETL process to maintain SCD2. We will have two data loads in this process.

1. With the initial data load, we will have all the records with colums of person\_foreman\_dim plus three columns effective\_timestamp (start date), expire\_timestamp (end date) and Current\_flag. We set the effective\_timestamp (Example: maybe the current date of the day you load data), expire\_timestamp will be infinite time (9999-12-31) and Current\_flag will be Current.
2. With the updated data load (Delta load with same changes of record). We will have to write code to check if record from source exists in dimension.
   1. If matching records exists (we use key to check) – check if any of the columns have updates in the values. If any columns have different values between source (delta load) and dimension (person\_foreman\_dim) 🡪 add new record with Current\_flag on current and effective\_timestamp will be the time data changed, expire\_timestamp is future date. In the old record, the current\_flag will be expired, the effective\_timestamp will be changed to the time data change (usually its equal effective\_timestamp of new record which is added).
   2. If record does not exist, add record with updated effective\_timestamp, expire\_timestamp will be future date, and current\_flag is current.

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

Dimension Factory\_dim would benefit from SCD3 implementation because the factory location (factory\_location) is very infrequent changes, and I would like to keep it simple by tracking the current and the previous changes only. This way we can have track the history location of the factory if the factory moves to new location.

In SCD 3, we will add one more column for changed attribute – “prior”. In this case, factory\_dim will have all the columns, along with a separate column for the original values and another for the previous value (factory\_location\_prev). In another word, in SCD type 3. We create a separate column for the original value of the field and another for the previous value.

ETL process to maintain SCD3. We will have two data loads in this process.

1. Load initial data: Load the initial data from the source system into the dimension table (load all data for the most recent version of each record). This mean that we only load the current value of each field, as we don’t have any historical data to store (factory\_location\_prev will be left empty in this case).
2. Second data load (delta load data):
   1. Check if record from source exists in dimension. If matching record exists – check if SCD 3 columns have updates in the values. In this context, we check if data in the factory\_location has different values from the source (delta load data) and factory\_dim (initial load data). If there is different. We set the current value of factory\_location (initial load data) as factory\_location\_prev, and the new value (from the delta load data) will be set to the factory\_location columns.
   2. If record does not exist, we add record. The value of factory\_location will be in the original column (factory\_location) and factory\_location\_prev will keep empty.
3. Pick one fact table from the design what are the measures?

**Fact table: Sales\_fact**

Measures: item\_quantity and order\_amount

# 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 \* from

(select c.manufacture\_year, f.factory\_label, sum(m.qty\_passed) as total\_units\_passed, sum(m.qty\_failed) as total\_units\_passed,

RANK() OVER (PARTITION BY c.manufacture\_year

ORDER BY sum(m.qty\_passed) DESC ) as Ranking

from factory\_dim f

join manufacture\_fact m ON m.factory\_key = f.factory\_key

join calendar\_manufacture\_dim c on c.manufacture\_cal\_key = m.manufacture\_cal\_key

GROUP BY c.manufacture\_year, f.factory\_label) a

Where a.Ranking <=3

Order By a.manufacture\_year DESC, a.Ranking;

**Screenshot of result:**

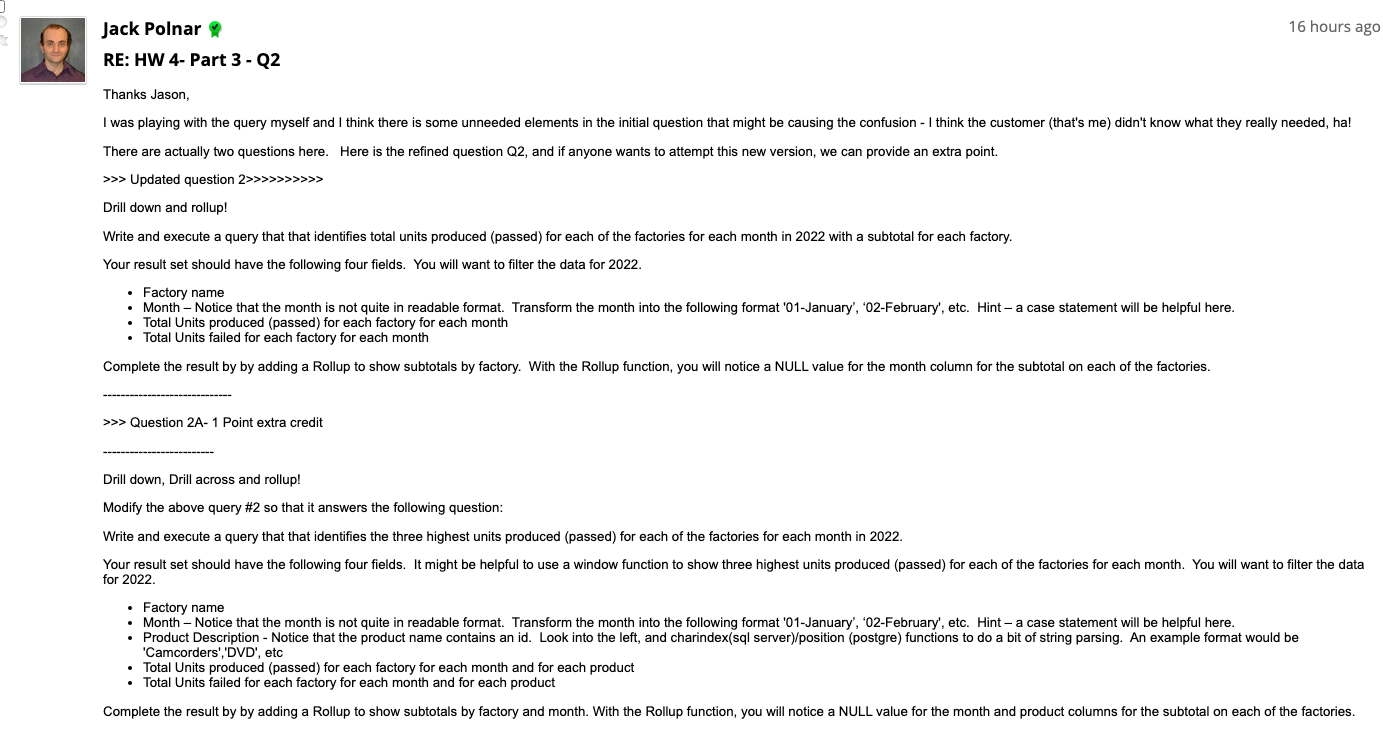
**Table

Description automatically generated**

1. Drill down and rollup! Write and execute a query that that identifies the three highest unit produced (passed) for each of the factories for each month in 2022. Use Rollup to show subtotals by factory. Your output should have these columns:

* 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

I am doing version 2 of this question with extra credit question:



*Question 2 – Version 2:*

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

**SQL Code:**

SELECT f.factory\_label As factory\_name,

CASE

When c.manufacture\_monthofyear is not null then right(c.manufacture\_monthofyear,2) || '-' ||

to\_char(to\_date(right(c.manufacture\_monthofyear,2), 'MM'), 'Month')

ELSE Null

End AS MONTH,

sum(m.qty\_passed) as total\_unit\_passed, sum(m.qty\_failed) as total\_unit\_failed

from factory\_dim f

join manufacture\_fact m ON m.factory\_key = f.factory\_key

join calendar\_manufacture\_dim c on c.manufacture\_cal\_key = m.manufacture\_cal\_key

where c.manufacture\_year = 'mY2022'

group by ROLLUP(f.factory\_label, c.manufacture\_monthofyear)

order by f.factory\_label, c.manufacture\_monthofyear;

**Screenshot of result:**

**Graphical user interface, table

Description automatically generated with medium confidence**

**Question\_2\_Extra Credit**

**SQL code:**

Select pr.factory\_name, pr.month, pr.product\_description, sum(pr.total\_units\_passed) as Total\_units\_passed,

sum(pr.total\_units\_failed) as Total\_units\_failed FROM

(SELECT f.factory\_label As factory\_name,

CASE

When c.manufacture\_monthofyear is not null then right(c.manufacture\_monthofyear,2) || '-' ||

to\_char(to\_date(right(c.manufacture\_monthofyear,2), 'MM'), 'Month')

ELSE Null

End AS MONTH,

left(p.product\_description, position(' id' in p.product\_description)) as product\_description,

sum(m.qty\_passed) as Total\_units\_passed,

sum(m.qty\_failed) as Total\_units\_failed,

rank() over(PARTITION by f.factory\_label, c.manufacture\_monthofyear order by sum(m.qty\_passed) desc) as rank\_s

from factory\_dim f

join manufacture\_fact m ON m.factory\_key = f.factory\_key

join calendar\_manufacture\_dim c on c.manufacture\_cal\_key = m.manufacture\_cal\_key

join product\_dim p on p.product\_key = m.product\_key

where c.manufacture\_year = 'mY2022'

group by f.factory\_label, p.product\_description, c.manufacture\_monthofyear

order by f.factory\_label, c.manufacture\_monthofyear) pr

where pr.rank\_s <=3

group by rollup(pr.factory\_name, pr.month, pr.product\_description)

ORDER By pr.factory\_name, pr.month

**Screenshot:**

**Graphical user interface, application

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:**

One suggestion I would implement as part of the design and ETL to make this question easier to solve above question is create new cumulative fact table with month time grain. This fact table will have factory name, monthID(refer to month\_dimension with month grain), units produced and units failed of each products in each month, productid (refer to product dimension). With this design we can query faster and more efficient, but the ETL process will have to do more transformation(aggregation).

1. Drill down and rollup! Modify the above query in question 2 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.**

I still choose year 2022 (mY2022)

**SQL Code:**

SELECT f.factory\_label As factory\_name,

CASE When c.manufacture\_monthofyear is not null then right(c.manufacture\_monthofyear,2) || '-' ||

to\_char(to\_date(right(c.manufacture\_monthofyear,2), 'MM'), 'Month') ELSE Null End AS MONTH, p.brand\_label as Brand,

sum(m.qty\_passed) as Total\_units\_passed,

sum(m.qty\_failed) as Total\_units\_failed

from factory\_dim f

join manufacture\_fact m ON m.factory\_key = f.factory\_key

join calendar\_manufacture\_dim c on c.manufacture\_cal\_key = m.manufacture\_cal\_key

join product\_dim p on p.product\_key = m.product\_key

where c.manufacture\_year = 'mY2022'

group by rollup(f.factory\_label, c.manufacture\_monthofyear, p.brand\_label)

order by f.factory\_label, c.manufacture\_monthofyear

**Screenshot of result:**

**Table

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:**

SELECT f.factory\_label As factory\_name,

CASE When c.manufacture\_monthofyear is not null then right(c.manufacture\_monthofyear,2) || '-' ||

to\_char(to\_date(right(c.manufacture\_monthofyear,2), 'MM'), 'Month') ELSE Null End AS MONTH, p.brand\_label as Brand,

sum(m.qty\_passed) as Total\_units\_passed,

sum(m.qty\_failed) as Total\_units\_failed

from factory\_dim f

join manufacture\_fact m ON m.factory\_key = f.factory\_key

join calendar\_manufacture\_dim c on c.manufacture\_cal\_key = m.manufacture\_cal\_key

join product\_dim p on p.product\_key = m.product\_key

where c.manufacture\_year = 'mY2022'

group by cube(f.factory\_label, c.manufacture\_monthofyear, p.brand\_label)

order by f.factory\_label, c.manufacture\_monthofyear

**Screenshot of result:**

Table

Description automatically generated

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

Both ROLLUP and CUBE are used to generate the subtotals and the grand total. But there are a few different between rollup and cube in my results. The Cube function give us more flexibility because it generates a result that shows aggregates for all combinations of values in the selected columns. On the other hand, ROLLUP does not generate all possible combinations, ROLLUP generates a result that shows aggregates for a hierarchy of values in the selected columns.

- With ROLLUP to show subtotals by factory, month, and brand. It only generates only four group sets, assuming the hierarchy factory > month > brand as follows:

(Factory, month, brand) – subtotals combination of three columns factory, month, brand

(Factory, month) – subtotals combination of 2 columns factory and month

(Factory) – subtotal of combination 1 column factory

() – grand total

- With CUBE to show subtotals by factory, month, and brand. It makes all 8 possible grouping sets:

(Factory, month, brand) -subtotals combination of three columns factory, month, brand

(Factory, month) – subtotals combination of 2 columns factory and month

(Month, brand) – subtotals combination of 2 columns month and brand

(Factory, brand) – subtotals combination of 2 columns factory and brand

(Factory) – subtotal of combination 1 column factory

(Month) – subtotal of combination 1 column month

(Brand) – subtotal of combination 1 column brand

() – 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 c.manufacture\_year as year, f.factory\_label as factory\_name, m.qty\_passed as Quantity\_passed

from factory\_dim f

join manufacture\_fact m ON m.factory\_key = f.factory\_key

join calendar\_manufacture\_dim c on c.manufacture\_cal\_key = m.manufacture\_cal\_key

where c.manufacture\_year between 'mY2017' and 'mY2022' and c.manufacture\_monthofyear = 'mMO02'

order by c.manufacture\_year desc;

**Screenshot of result:**

Table

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:**

CREATE EXTENSION IF NOT EXISTS tablefunc;

**----**

SELECT \* FROM crosstab(

'select f.factory\_label as factory\_name, c.manufacture\_year as year, sum(m.qty\_passed) as Quantity\_passed

from factory\_dim f

join manufacture\_fact m ON m.factory\_key = f.factory\_key

join calendar\_manufacture\_dim c on c.manufacture\_cal\_key = m.manufacture\_cal\_key

where c.manufacture\_year between ''mY2017'' and ''mY2022'' and c.manufacture\_monthofyear = ''mMO02''

group by f.factory\_label, c.manufacture\_year

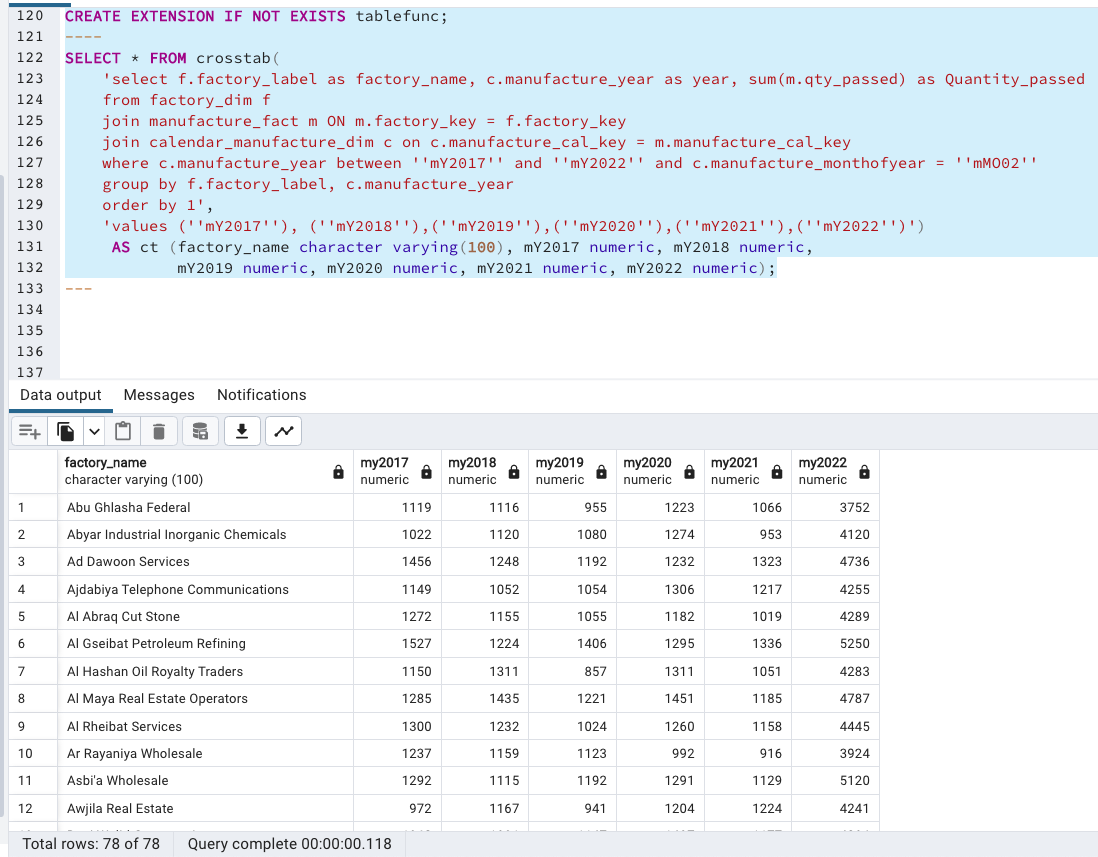
order by 1',

'values (''mY2017''), (''mY2018''),(''mY2019''),(''mY2020''),(''mY2021''),(''mY2022'')')

AS ct (factory\_name character varying(100), mY2017 numeric, mY2018 numeric,

mY2019 numeric, mY2020 numeric, mY2021 numeric, mY2022 numeric);

**Screenshot of result:**



# Part 4 – Tableau Data Presentation

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

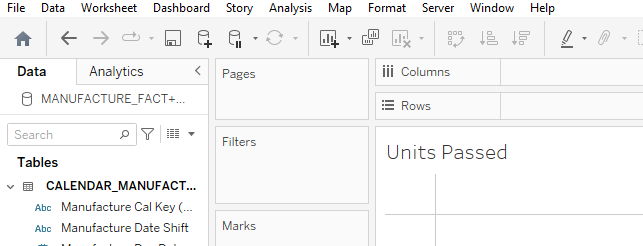
1. Download and install Tableau. You can get a student version here: <https://www.tableau.com/academic/students>. Note: it make a take few days to get a license. Alternatively, you can download a 14-day free trial version while you wait for your student license: <https://www.tableau.com/products/trial>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

Graphical user interface, application

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.*

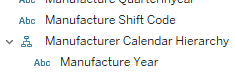
*The pivot table we just created is the exact same one you solved in question 7 from part 3! You should see the same data. Yes, I got the same data with question 7 part 3*.

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

**Graphical user interface, application, table

Description automatically generated**

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, Yearmonth, Yearquarter, 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):



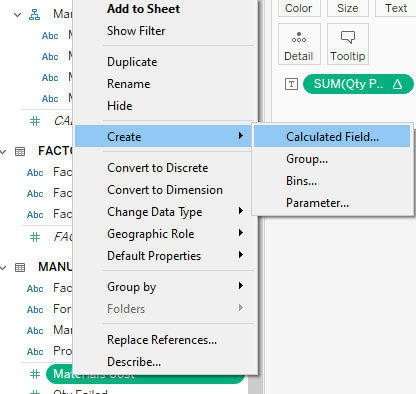
* Remove the Manufacture Year from the Columns at the top and replace it with the Manufacture Calendar Hierarchy you just created.
* You will see a plus next to the Manufacture Year in your column area, click on it to expand date hierarchy column to Months which will appear in your pivot table.
* Remove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

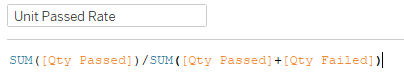
**Text

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?**

**Jalu Water Transportation** has the highest unit pass rate for 2022.

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

**Graphical user interface

Description automatically generated**

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**

**Graphical user interface

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**

**Graphical user interface, application

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.**

**Chart, bar chart

Description automatically generated**

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

Tanksale Cranes has the worst quarter in 2019. It has only 312 order amounts

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

Tawase Cranes has the best quarter in 2019. It has 22441 order amounts.

# 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.

* + - What is the most popular product sale in quarter 4 of 2020 in Vancouver British Columbia and its total quantity?

Personal Computers items are the most popular and its total quantity is 251.

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

**Chart, bar chart

Description automatically generated**

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.

