OPENSTAT PSA: SUMMARY STATISTICS FOR MANUFACTURING ESTABLISHMENTS BY INDUSTRY DESCRIPTION, SIZE AND DATA ITEM

A capstone project
For PROJECT SPARTA Y5
Data Engineer Pathway

KARL JOSEPH C. CACAO, RCE

November 2024

ACKNOWLEDGEMENT

This section presents the scholar's gratitude for their contributions to the project's success:

The **Development Academy of the Philippines** (DAP), for granting him a scholarship for the PROJECT SPARTA Y5 in order to be enhanced in data literacy.

The **Department of Science and Technology - Philippine Council for Industry, Energy, and Emerging Technology Research and Development** (DOST-PCIEERD), being one of the agencies for science and technology innovation and research play a national role for STEM developments in the country.

The **Phillippine Statistics Authoritiy** (PSA) for providing the Filipinos handful of open data for the benefit of public good. To its website OpenSTAT, it has been an interesting platform accessible to all to gather information about statuses of the Philippine sectors.

Pleasantries and gratitude also to the **Data Engineering Pilipinas**, a PyData community in the Philippines for inspiring young students, professionals and out-of-school people, to be data-driven with the help of their advocacy. The journey of the scholar started from Data Analytics, but keep on moving on Data Engineering to understand more of the big data.

To **Engr. Rammel F. Mistica, Engr. Crispino A. Aranilla**, the scholar's former academic instructors for believing in him for the opportunity of learning the Data Engineer Pathway.

To the scholar's *family* who morally support his decision and potential in developing his skills in data and integrate these to various domain knowledge.

Most especially to **Almighty God** who showed tenacity and determination in completing this research.

K.J.C.C

INTRODUCTION

Philippine economics seemed to be abstract in the point of view of ordinary Filipinos. Most, if not all are not aware of the ongoing status of the economy for various sectors. One of the challenges lie in the information dissemination.

The Philippine Statistics Authority holds all the necessary information for several aspects. They are responsible in the national censuses and surveys as the legal way of collecting information for the record and may be used for academic and socio-political developments. For social statistics, information such as demographics, population and labor could be accessed to the agency. For information on resources, environmental, agricultural—fisheries, mining and construction could be present also. PSA has a website for open data in the Philippine context. OpenSTAT (https://openstat.psa.gov.ph) is the website made accessible to the public for viewing statistics on the different sectors of the economy.

As cited on United Nations Department of Economic and Social Affairs, there are 17 goals under the Sustainable Development Goals (SDGs). In line with the statistics for economy, SDG8 or the goal number 8 states “Promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all. As of 2024 sustainable development report dashboard, (https://dashboards.sdgindex.org/map/goals/SDG8) it is indicated that the Philippines remain on facing major challenges. Indicators supporting the goal are similarly marked as facing major challenges also for the Philippines.

STATEMENT OF THE PROBLEM

Specifically, the scholar aims to conduct a data engineering workflow to deliver clean and understandable data for further analytics as the data has to be presented to the general public. The said workflow will be in a form of a relational database

For economic statistics, the data will focus on manufacturing industry. Available records are for years 2015, 2016 and 2017 only. The summary statistics gathered for the relational database will be based on the **Annual Survey of Philippine Business and Industry** available in the PSA OpenSTAT website.

Furthermore, the statistics are classified into following:

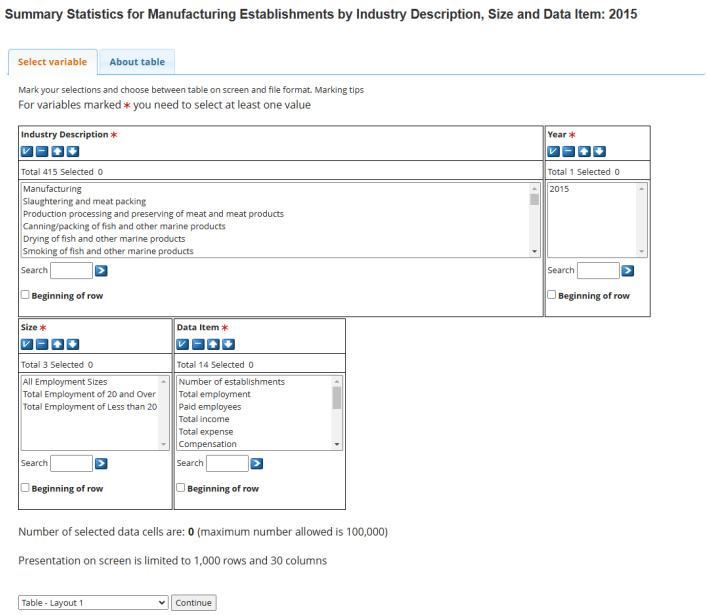
- Industry Description
- Year of observation
- Employment size
- Data Item (Statistical Measure)

METHODOLOGY

Data Integration Approach

Due to accessibility presented in the OpenSTAT website, the **ELT (Extract, Load, Transform)** was used to be the data processing method.

For extracting data, the database is in PXWeb, a web-based tool used for statistical data. The tool provides the statistical data in granulated columns so that the user could easily select the information based on the desired variable and corresponding statistics.

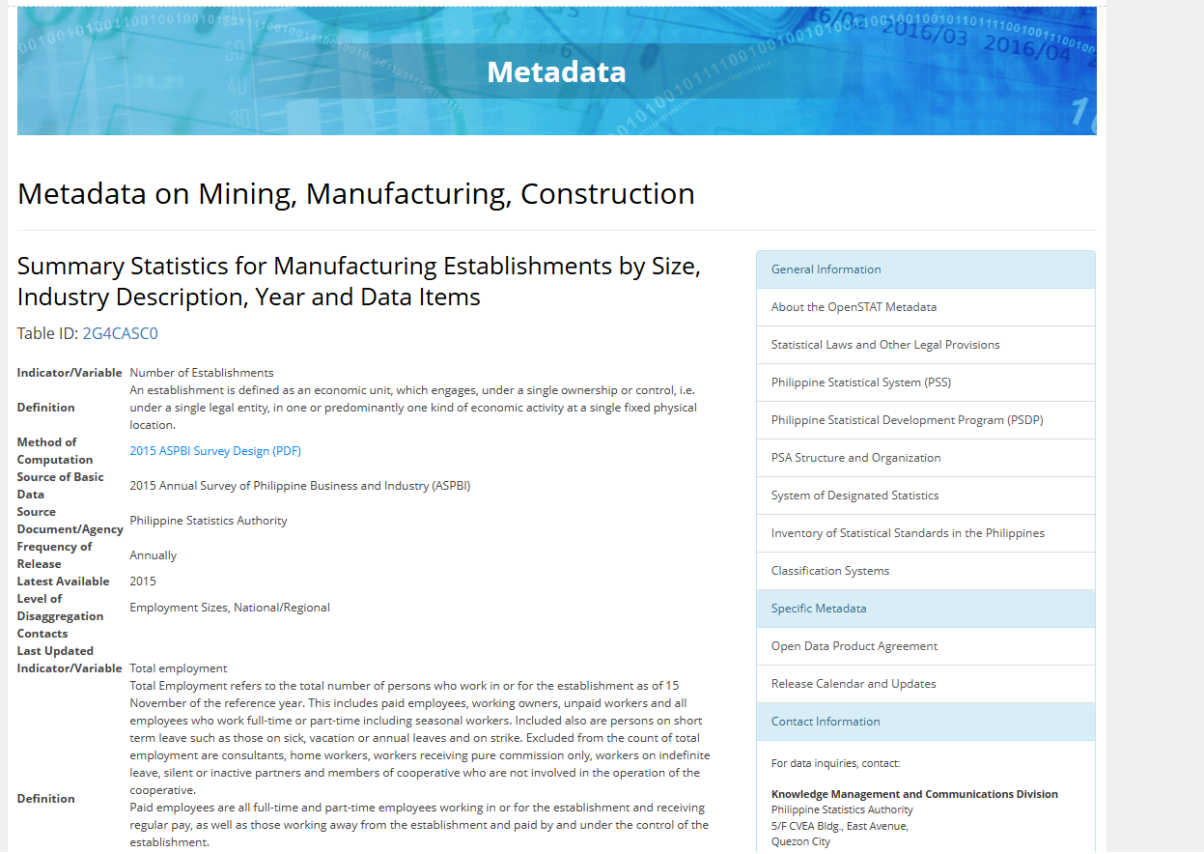


Sample database using the PXWEB tool for data collection

The said platform can extract the necessary data and can be presented into tabulated datasets, graphs and charts. Additionally, the exported data are also available in various file format such as comma-separated values, excel spreadsheet file, etc.

Metadata

Attached to the database are the definitions regarding information about the statistical measure or data item. The data items are the fields of the dataset gathered.



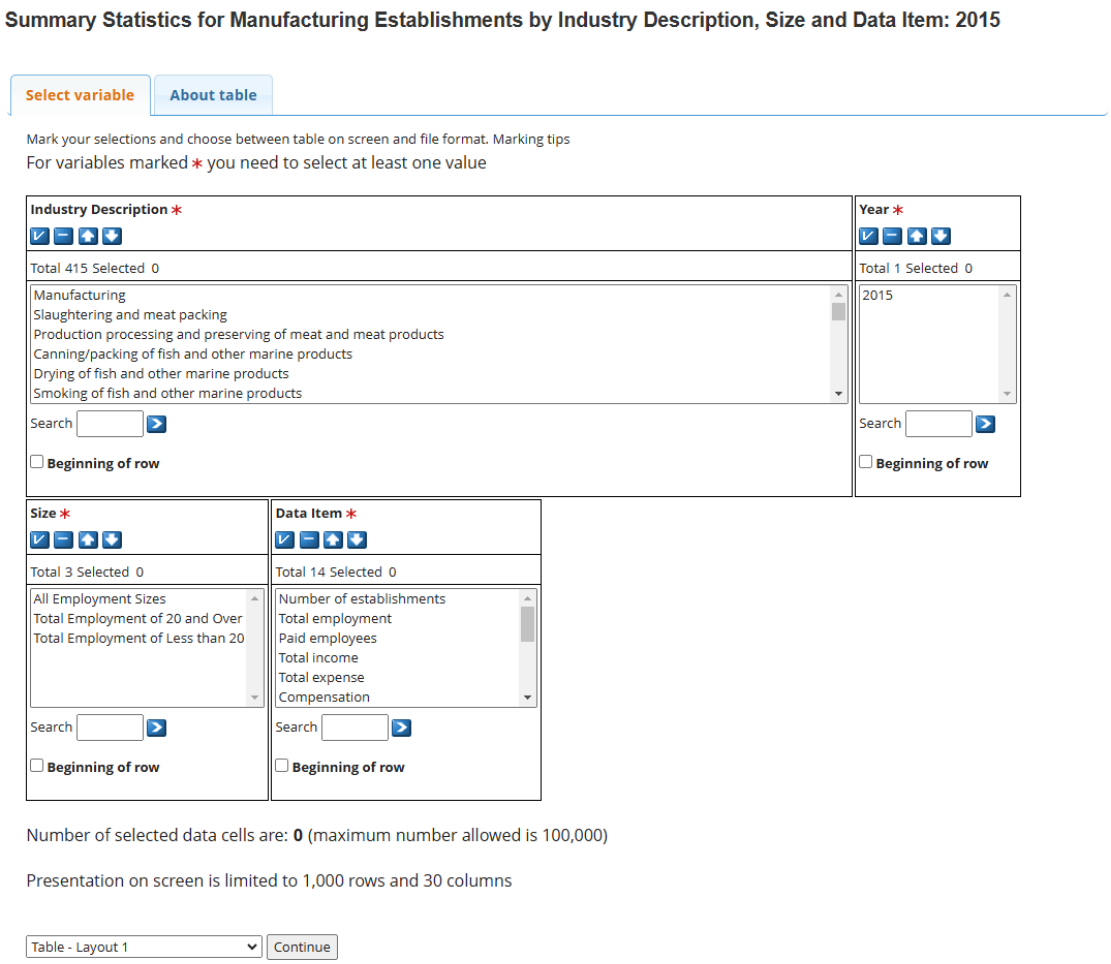
https://openstat.psa.gov.ph/Metadata/2G4CASC0

The values were in thousand pesos; except for the industry description and employment data. In addition, footnotes for the database stated the following:

- "s" – suppressed data
- "c" – combined data
- ".. "– unavailable data

Note that the rows containing these data are to be omitted as these does not help with the summary statistics.

Data Extraction Proper



The simplest extraction for the database here, is aggregating into two “Size(s)”, the **Total Employment of 20 and over**, and the **Total employment of less than 20.** In this situation, if the dataset hasn’t separated data for various years, my extraction will be based on the annual reports before aggregating into employment sizes.

Hence the extraction process was done by six times to six CSV files. Two extractions were done per year as per employment size, and the years **2015, 2016 and 2017** were involved.

Loading Data

Using PostgreSQL, the initial loading of data takes place there. The column names were renamed due to complexity or length. The following are the newly assigned column names and the actual metadata:

- **industry_desc** : Industry Description
- **est_number :** Number of establishments
- **total_employment** : Total employment
- **paid_employees**: Paid employees
- **total_income**: Total income
- **total_exp**: Total Expense
- **compensation**: Compensation
- **other_exp**; Other expense
- **value_of_output**: Value of Output
- **intermediate_exp**: Intermediate Expense
- **value_added**: Value added
- **tf_assets**: Gross addition to tangible fixed assets
- **change_inv** : Change in inventories
- **subsidy**
- **commerce_transaction** : Sales from e-commerce transactions

Query Query History

```
--------------------------------------------------------------------
--CHECK IF TABLE ALREADY EXISTS--
--------------------------------------------------------------------
DROP TABLE IF EXISTS large_manufacturer_2015;
DROP TABLE IF EXISTS small_manufacturer_2015;
DROP TABLE IF EXISTS large_manufacturer_2016;
DROP TABLE IF EXISTS small_manufacturer_2016;
DROP TABLE IF EXISTS large_manufacturer_2017;
DROP TABLE IF EXISTS small_manufacturer_2017;
--------------------------------------------------------------------
--CREATE TABLE large_manufacturer_2015--
--------------------------------------------------------------------
CREATE TABLE large_manufacturer_2015 (
industry_desc VARCHAR(512),
est_number VARCHAR(64),
total_employment VARCHAR(64),
paid_employees VARCHAR(64),
total_income VARCHAR(64),
total_exp VARCHAR(64),
compensation VARCHAR(64),
other_exp VARCHAR(64),
value_of_output VARCHAR(64),
intermediate_exp VARCHAR(64),
value_added VARCHAR(64),
tf_assets VARCHAR(64),
change_inv VARCHAR(64),
subsidy VARCHAR(64),
commerce_transaction VARCHAR(64)
);

--importing data from OpenSTAT
COPY large_manufacturer_2015
FROM 'C:\kjcc\DA\Project SPARTA Y5\Capstone\OpenstatPSA\EconomicStat\Manufacturing\2015\large_manufacturer_2015.csv'
DELIMITER ','
CSV HEADER;
```

Part of the query in importing a dataset.

As for the footnotes, there are strings (‘c’, ‘s’, ‘..’) involved in the extracted data sets and has to be excluded. Therefore the dataset to import here are initially all in VARCHAR() datatype.

```
Query  Query History
--Check out the industry description
DROP TABLE IF EXISTS industry_description CASCADE;
CREATE TABLE industry_description(
id SERIAL NOT NULL,
industry_desc VARCHAR(512) NOT NULL,
ind_code INTEGER PRIMARY KEY
);

INSERT INTO industry_description (industry_desc, ind_code)
(SELECT industry_desc, ('15' || ROW_NUMBER() OVER()):: INTEGER AS ind_code
FROM large_manufacturer_2015)
;

INSERT INTO industry_description (industry_desc, ind_code)
(SELECT industry_desc, ('16' ||  ROW_NUMBER() OVER()):: INTEGER AS ind_code
FROM large_manufacturer_2016)
;

INSERT INTO industry_description (industry_desc, ind_code)
(SELECT industry_desc, ('17' ||  ROW_NUMBER() OVER()):: INTEGER AS ind_code
FROM large_manufacturer_2017)
;

SELECT industry_desc, COUNT(*) FROM industry_description
GROUP BY ind_code
ORDER BY COUNT(*) DESC;

--Create YEAR Table
DROP TABLE IF EXISTS year;
CREATE TABLE year(
year integer PRIMARY KEY);

INSERT INTO year VALUES (2015);
INSERT INTO year VALUES (2016);
INSERT INTO year VALUES (2017);
```

Creation of keys for future referencing

The query shown is necessary for creating primary and foreign keys of the respective datasets. NOT NULL statements are used to prevent the workflow from having NULL values. The consideration of concatenating the last two digits of year of observation to the ROW_NUMBER statements so that redundancy will be avoided. The *ind_code* serves as a key has to be typecasted into INTEGER.

Cleaning Data

```
Query  Query History
--------------------------------------------------------------------------
--Deleting unnecessary data for manufacturer data--
--------------------------------------------------------------------------

DELETE
    FROM large_manufacturer_2015
    WHERE
    est_number IN ('c','s','..') OR
    total_exp IN ('c','s','..') OR
    total_employment IN ('c','s','..') OR
    paid_employees IN ('c','s','..') OR
    total_income IN ('c','s','..') OR
    compensation IN ('c','s','..') OR
    other_exp IN ('c','s','..') OR
    value_of_output IN ('c','s','..') OR
    intermediate_exp  IN ('c','s','..') OR
    value_added  IN ('c','s','..') OR
    tf_assets IN ('c','s','..') OR
    change_inv IN ('c','s','..') OR
    subsidy IN ('c','s','..') OR
    commerce_transaction IN ('c','s','..');

DELETE
    FROM small_manufacturer_2015
    WHERE
    est_number IN ('c','s','..') OR
    total_exp IN ('c','s','..') OR
    total_employment IN ('c','s','..') OR
    paid_employees IN ('c','s','..') OR
    total_income IN ('c','s','..') OR
    compensation IN ('c','s','..') OR
    other_exp IN ('c','s','..') OR
    value_of_output IN ('c','s','..') OR
    intermediate_exp  IN ('c','s','..') OR
    value_added  IN ('c','s','..') OR
```

Excluding the non-numerical data

As mentioned earlier about the footnotes, these data are not significant to the whole data workflow, if not excluded, summary statistics will not work as expected due to the non-numerical values in the dataset.

Transforming Data

After the cleaning proper, the fields must be able to be converted into the respective and/or necessary datatype. Recall that the values were in thousand pesos; except for the industry description and employment data. Hence we are using the INTEGER and DECIMAL as datatype. In addition the *year*, *emp_code* and *ind_code* are to be the new columns of the datasets for key-connections.

Query | Query History

```
1  ---------------------------------------------------------------------
2  --Transforming Data--
3  ---------------------------------------------------------------------
4
5  --Adding column for foreign keys
6  ALTER TABLE large_manufacturer_2015 ADD COLUMN ind_code INTEGER;
7  ALTER TABLE large_manufacturer_2016 ADD COLUMN ind_code INTEGER;
8  ALTER TABLE large_manufacturer_2017 ADD COLUMN ind_code INTEGER;
9  ALTER TABLE small_manufacturer_2015 ADD COLUMN ind_code INTEGER;
10 ALTER TABLE small_manufacturer_2016 ADD COLUMN ind_code INTEGER;
11 ALTER TABLE small_manufacturer_2017 ADD COLUMN ind_code INTEGER;
12
13 ALTER TABLE large_manufacturer_2015 ADD COLUMN year INTEGER;
14 ALTER TABLE large_manufacturer_2016 ADD COLUMN year INTEGER;
15 ALTER TABLE large_manufacturer_2017 ADD COLUMN year INTEGER;
16 ALTER TABLE small_manufacturer_2015 ADD COLUMN year INTEGER;
17 ALTER TABLE small_manufacturer_2016 ADD COLUMN year INTEGER;
18 ALTER TABLE small_manufacturer_2017 ADD COLUMN year INTEGER;
19
20 ALTER TABLE large_manufacturer_2015 ADD COLUMN emp_code VARCHAR(5);
21 ALTER TABLE large_manufacturer_2016 ADD COLUMN emp_code VARCHAR(5);
22 ALTER TABLE large_manufacturer_2017 ADD COLUMN emp_code VARCHAR(5);
23 ALTER TABLE small_manufacturer_2015 ADD COLUMN emp_code VARCHAR(5);
24 ALTER TABLE small_manufacturer_2016 ADD COLUMN emp_code VARCHAR(5);
25 ALTER TABLE small_manufacturer_2017 ADD COLUMN emp_code VARCHAR(5);
26
```

Additional columns for every dataset

Query | Query History

```
238 ---------------------------------------------------------------------
239 --Key Relationships--
240 ---------------------------------------------------------------------
241 ALTER TABLE large_manufacturer_2015
242     ADD CONSTRAINT fk_ind_code FOREIGN KEY (ind_code) REFERENCES industry_description(ind_code);
243 ALTER TABLE large_manufacturer_2016
244     ADD CONSTRAINT fk_ind_code FOREIGN KEY (ind_code) REFERENCES industry_description(ind_code);
245 ALTER TABLE large_manufacturer_2017
246     ADD CONSTRAINT fk_ind_code FOREIGN KEY (ind_code) REFERENCES industry_description(ind_code);
247
248 ALTER TABLE small_manufacturer_2015
249     ADD CONSTRAINT fk_ind_code FOREIGN KEY (ind_code) REFERENCES industry_description(ind_code);
250 ALTER TABLE small_manufacturer_2016
251     ADD CONSTRAINT fk_ind_code FOREIGN KEY (ind_code) REFERENCES industry_description(ind_code);
252 ALTER TABLE small_manufacturer_2017
253     ADD CONSTRAINT fk_ind_code FOREIGN KEY (ind_code) REFERENCES industry_description(ind_code);
254
255 ALTER TABLE large_manufacturer_2015
256     ADD CONSTRAINT fk_year FOREIGN KEY (year) REFERENCES year(year);
257 ALTER TABLE large_manufacturer_2016
258     ADD CONSTRAINT fk_year FOREIGN KEY (year) REFERENCES year(year);
259 ALTER TABLE large_manufacturer_2017
260     ADD CONSTRAINT fk_year FOREIGN KEY (year) REFERENCES year(year);
261
262 ALTER TABLE small_manufacturer_2015
263     ADD CONSTRAINT fk_year FOREIGN KEY (year) REFERENCES year(year);
264 ALTER TABLE small_manufacturer_2016
265     ADD CONSTRAINT fk_year FOREIGN KEY (year) REFERENCES year(year);
266 ALTER TABLE small_manufacturer_2017
267     ADD CONSTRAINT fk_year FOREIGN KEY (year) REFERENCES year(year);
268
269 ALTER TABLE large_manufacturer_2015
270     ADD CONSTRAINT fk_emp_size FOREIGN KEY (emp_code) REFERENCES employment_size(emp_code);
271 ALTER TABLE large_manufacturer_2016
272     ADD CONSTRAINT fk_emp_size FOREIGN KEY (emp_code) REFERENCES employment_size(emp_code);
273 ALTER TABLE large_manufacturer_2017
274     ADD CONSTRAINT fk_emp_size FOREIGN KEY (emp_code) REFERENCES employment_size(emp_code);
275
```

Primary Key to Foreign Key links

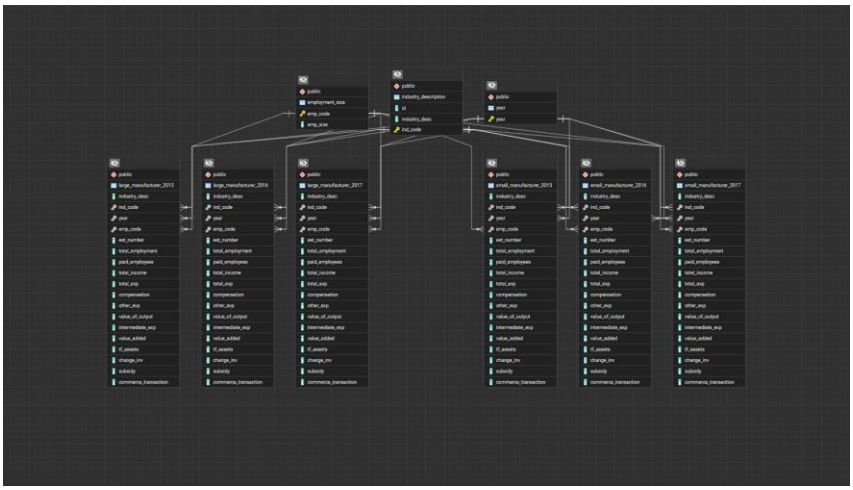
```
Query  Query History

--------------------------------------------------------------------------
--Converting the proper data type--
--------------------------------------------------------------------------

ALTER TABLE large_manufacturer_2015 ALTER COLUMN est_number TYPE INTEGER USING est_number::INTEGER;
ALTER TABLE large_manufacturer_2015 ALTER COLUMN total_employment TYPE INTEGER USING total_employment::INTEGER;
ALTER TABLE large_manufacturer_2015 ALTER COLUMN paid_employees TYPE INTEGER USING paid_employees::INTEGER ;
ALTER TABLE large_manufacturer_2015 ALTER COLUMN total_income TYPE DECIMAL (15,2) USING total_income::DECIMAL(15,2);
ALTER TABLE large_manufacturer_2015 ALTER COLUMN total_exp TYPE DECIMAL (15,2) USING total_exp::DECIMAL(15,2);
ALTER TABLE large_manufacturer_2015 ALTER COLUMN compensation TYPE DECIMAL (15,2) USING compensation::DECIMAL (15,2);
ALTER TABLE large_manufacturer_2015 ALTER COLUMN other_exp TYPE  DECIMAL (15,2) USING other_exp::DECIMAL (15,2);
ALTER TABLE large_manufacturer_2015 ALTER COLUMN value_of_output TYPE DECIMAL (15,2) USING value_of_output::DECIMAL (15,2);
ALTER TABLE large_manufacturer_2015 ALTER COLUMN intermediate_exp TYPE DECIMAL (15,2) USING intermediate_exp::DECIMAL (15,2);
ALTER TABLE large_manufacturer_2015 ALTER COLUMN value_added TYPE DECIMAL (15,2) USING value_added::DECIMAL (15,2);
ALTER TABLE large_manufacturer_2015 ALTER COLUMN tf_assets TYPE DECIMAL (15,2) USING tf_assets::DECIMAL (15,2);
ALTER TABLE large_manufacturer_2015 ALTER COLUMN change_inv TYPE DECIMAL (15,2) USING change_inv::DECIMAL (15,2);
ALTER TABLE large_manufacturer_2015 ALTER COLUMN subsidy TYPE DECIMAL (15,2) USING subsidy::DECIMAL (15,2);
ALTER TABLE large_manufacturer_2015 ALTER COLUMN commerce_transaction TYPE DECIMAL (15,2) USING commerce_transaction::DECIMAL (15,2);

ALTER TABLE large_manufacturer_2016 ALTER COLUMN est_number TYPE INTEGER USING est_number::INTEGER;
ALTER TABLE large_manufacturer_2016 ALTER COLUMN total_employment TYPE INTEGER USING total_employment::INTEGER;
ALTER TABLE large_manufacturer_2016 ALTER COLUMN paid_employees TYPE INTEGER USING paid_employees::INTEGER ;
ALTER TABLE large_manufacturer_2016 ALTER COLUMN total_income TYPE DECIMAL (15,2) USING total_income::DECIMAL(15,2);
ALTER TABLE large_manufacturer_2016 ALTER COLUMN total_exp TYPE DECIMAL (15,2) USING total_exp::DECIMAL(15,2);
ALTER TABLE large_manufacturer_2016 ALTER COLUMN compensation TYPE DECIMAL (15,2) USING compensation::DECIMAL (15,2);
ALTER TABLE large_manufacturer_2016 ALTER COLUMN other_exp TYPE  DECIMAL (15,2) USING other_exp::DECIMAL (15,2);
ALTER TABLE large_manufacturer_2016 ALTER COLUMN value_of_output TYPE DECIMAL (15,2) USING value_of_output::DECIMAL (15,2);
ALTER TABLE large_manufacturer_2016 ALTER COLUMN intermediate_exp TYPE DECIMAL (15,2) USING intermediate_exp::DECIMAL (15,2);
ALTER TABLE large_manufacturer_2016 ALTER COLUMN value_added TYPE DECIMAL (15,2) USING value_added::DECIMAL (15,2);
ALTER TABLE large_manufacturer_2016 ALTER COLUMN tf_assets TYPE DECIMAL (15,2) USING tf_assets::DECIMAL (15,2);
ALTER TABLE large_manufacturer_2016 ALTER COLUMN change_inv TYPE DECIMAL (15,2) USING change_inv::DECIMAL (15,2);
ALTER TABLE large_manufacturer_2016 ALTER COLUMN subsidy TYPE DECIMAL (15,2) USING subsidy::DECIMAL (15,2);
ALTER TABLE large_manufacturer_2016 ALTER COLUMN commerce_transaction TYPE DECIMAL (15,2) USING commerce_transaction::DECIMAL (15,2);
```

Notice that the ALTER queries have additional statements such as TYPE and USING. These are significant statements to ensure the typecasting will be secured and none of the records will be completely converted into. The reason is all of these data are actually in numbers. It only differs that some of the records are in Philippine Pesos, that is why it has to be in DECIMAL data type.

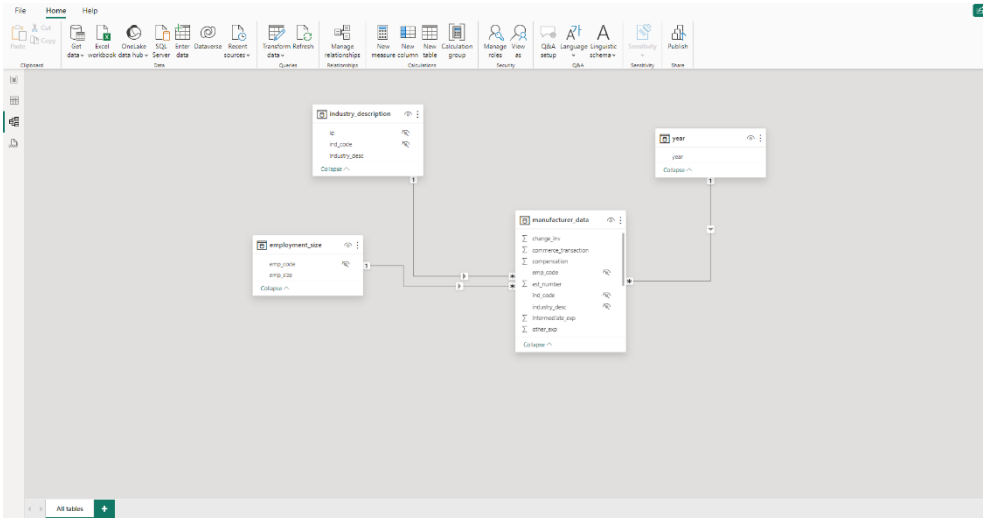
Exporting Data

```
Query  Query History

-----------------------------------------------
--PRODUCT ETL FOR VISUALIZATION--
-----------------------------------------------


--Manufacturer Data for 2015 to 2017
COPY (SELECT * FROM large_manufacturer_2015
UNION ALL
SELECT * FROM small_manufacturer_2015
UNION ALL
SELECT * FROM large_manufacturer_2016
UNION ALL
SELECT * FROM small_manufacturer_2016
UNION ALL
SELECT * FROM large_manufacturer_2017
UNION ALL
SELECT * FROM small_manufacturer_2017) TO 'C:\kjcc\DA\Project SPARTA Y5\Capstone\OpenstatPSA\EconomicStat\Manufacturing\manufacturer_data.csv' WITH (FORMAT CSV, HEADER);


--Export the primary/foreign key tables
COPY (SELECT * FROM year) TO 'C:\kjcc\DA\Project SPARTA Y5\Capstone\OpenstatPSA\EconomicStat\Manufacturing\year.csv' WITH (FORMAT CSV, HEADER);
COPY (SELECT * FROM employment_size) TO 'C:\kjcc\DA\Project SPARTA Y5\Capstone\OpenstatPSA\EconomicStat\Manufacturing\employment_size.csv' WITH (FORMAT CSV, HEADER);
COPY (SELECT * FROM industry_description)TO 'C:\kjcc\DA\Project SPARTA Y5\Capstone\OpenstatPSA\EconomicStat\Manufacturing\industry_description.csv' WITH (FORMAT CSV, HEADER);
```

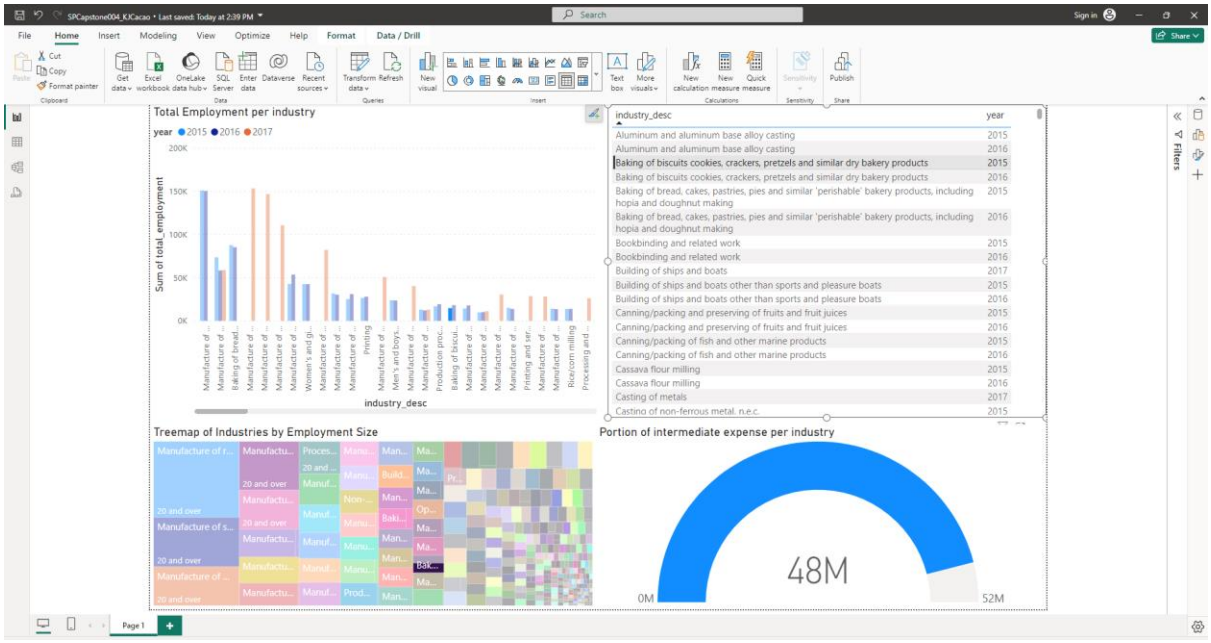
CONCLUSION



The ERD for the manufacturing data schema showcases a well-structured database design, capturing comprehensive information about large and small manufacturers across three years (2015, 2016, and 2017). The central industry_description table, identified by the ind_code, anchors the relationships with various attributes of the summary statistics. This normalization ensures data consistency and integrity, facilitating robust data analysis, reporting, and decision-making. Through this ERD, audience, or the general public can efficiently navigate and interpret the interconnected datasets, leading to more informed insights and strategies within the manufacturing sector.



The figure above shows the simplified schema made for visualization. In Power BI, a schema is essential for linking various visualizations used.



The visualization above effectively highlights the summary statistics across various industries. These figures can be filtered based on the year of observation, industry type, or employment sizes.

The following visualizations used are bar chart, table, treemap and gauge/circular chart. For summary statistics for manufacturing, these are the nearest visualization appropriate for understanding the economy for these relative sectors. The visualization is interactive, and audience may acquire various insights based on the selection of variables on the visualization.

REFERENCES:

Summary Statistics for Manufacturing Establishments by Industry Description, Size and Data Item: 2015-2017;

https://openstat.psa.gov.ph/PXWeb/pxweb/en/DB/DB__2G__MNFG__ASPBI/?tablelist=true&rxid=bdf9d8da-96f1-4100-ae09-18cb3eaeb313; 2024

Metadata on Mining, Manufacturing, Construction;

https://openstat.psa.gov.ph/Metadata/2G4CASC0

Dashboards on Sustainable Development Goal 8;

https://dashboards.sdgindex.org/map/goals/SDG8