Contents

[Task 1 2](#_Toc198461819)

[Task 1 Question 1 2](#_Toc198461820)

[Task 1 Question 2 3](#_Toc198461821)

[Task 1 Question 3 4](#_Toc198461822)

[Task 2 5](#_Toc198461823)

[Task 2 Question 1.a 5](#_Toc198461824)

[Task 2 Question 2.a 7](#_Toc198461825)

[Task 2 Question 2.b 8](#_Toc198461826)

[Task 2 Question 3 8](#_Toc198461827)

[Task 2 Question 4 9](#_Toc198461828)

[Task 2 Question 5 10](#_Toc198461829)

[Task 2 Question 5.a 10](#_Toc198461830)

[Task 2 Question 5.b 11](#_Toc198461831)

[References 14](#_Toc198461832)

# Task 1

To answer the questions below, I first created the db, setup the table and populated it with CSV data, and setup a user so that I could run python code. This code is located in /home/s4828041/a2/code.sql.

*A computer screen with text on it

AI-generated content may be incorrect.*

*Setting up for the task.*

I copied the python code over from practical 3 and modified it to answer the questions. ChatGPT (OpenAI, 2025) was used to create a function for calculating Levenshtein distance which was placed into the similarity.py file and referenced in there also. All of the python code used for this task is found in /home/s4828041/a2/a2t1/.

*A computer screen shot of a program

AI-generated content may be incorrect.*

*The levenshtein\_distance function which ChatGPT (OpenAI, 2025) assisted with.*

## Task 1 Question 1

I used several different thresholds to match records based on edit-distance. That is, how many modifications would be required for one word to be the same as another.

For each iteration, I printed the edit-distance threshold I used, the total number of similar records, and the precision, recall, and f-measure results.

As the threshold increased, the precision decreased sharply, that is, that all matched records were correct. The recall increased slightly when increasing the threshold from 1 to 2, which shows how many correct matches the program found.

The f-measure is a combination of both of these results, and it was shown that this dropped slightly when the threshold was increased from 1 to 2, and sharply with further decreases.

**After this testing, a threshold value of 1 was deemed to be the best and matched 90 records, with a precision of 0.84, recall of 0.72, and f-measure of 0.78.**

*A screenshot of a computer screen

AI-generated content may be incorrect.*

*The results of my python code when matching restaurants based on different edit-distance thresholds using the Levenshtein distance method.*

## Task 1 Question 2

For this question, I adjusted two difference values. The first was the q value, which determines the length of the tokens a string is split up into. The second was the threshold, which determines how similar two strings need to be to each other to be considered a match based on the Jaccard coefficient.

First I adjusted the q value from 3 down to 1, this change decreased the precision, but increased the recall.

Then when I found a q value I liked, I changed the threshold from 0.75 down to 0.5 and then up to 1. The higher the threshold, the higher the precision, but the lower the recall.

**At the end of this testing, a q value of 2 and threshold of 0.75 was deemed to be the best and matched 92 records, with a precision of 0.88, recall of 0.76, and f-measure of 0.82.**

A screenshot of a computer

AI-generated content may be incorrect.

*The results of my python code when matching restaurants based on different Jaccard coefficient thresholds and q values using the Jaccard coefficient method.*

## Task 1 Question 3

To answer this question we can compare the two most successful iterations of both measures.

At it’s best, the edit-distance method matched 90 records, with a precision of 0.84, recall of 0.72, and f-measure of 0.78.

The best Jaccard coefficient method matched 92 records, with a precision of 0.88, recall of 0.76, and f-measure of 0.82.

**These results show that the Jaccard coefficient similarity measure is better for the restaurant dataset. It matched more correct records, with a greater accuracy than the edit-distance method.**

# Task 2

## Task 2 Question 1.a

**To achieve this question I created a conceptual model** to ensure that I included all of the necessary values in the dimension tables that would be required to feed into the sales\_fact table. **The conceptual model is shown below.**

This included making a staff, product, and time\_period table which included all of the distinct values in the sales table based on their primary keys.

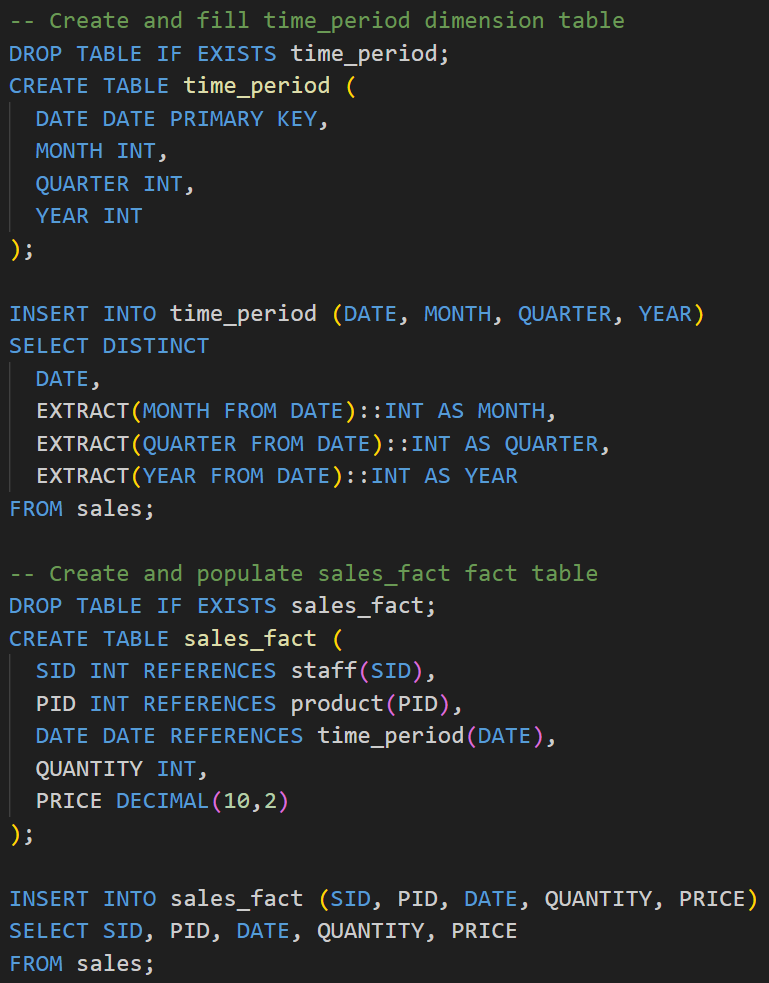
A sales\_fact table was then created and populated which utilises these distinct values, and based on the date of transactions, holds the staff id, product id, date, quantity, and price of a sale on any given date.

Screenshots of this process are shown below.

A diagram of a product

AI-generated content may be incorrect.

***Conceptual model*** *of my data warehouse with star schema design.*

**

*Some example code showing how tables were created, s4828041/a2/code.sql on my zone contains all sql used for the assignment.*

*A screenshot of a computer

AI-generated content may be incorrect.*

*Creating and populating the dimension and fact tables.*

*A screenshot of a computer screen

AI-generated content may be incorrect.*

*Schema for product and staff dimension tables.*

*A screenshot of a computer

AI-generated content may be incorrect.*

*Schema for the time\_period dimension table and sales\_fact fact table.*

## Task 2 Question 2.a

**There are 300 unique staff members**, as shown in the query below. The staff table was already populated with distinct values, so only a select count star query was required.

*A black screen with white text

AI-generated content may be incorrect.*

*A select count statement was all that was required to find the total number of unique staff.*

## Task 2 Question 2.b

**There were 24036 unique sales in the year 2022**, as shown below. For this question, I did a count star query on the sales\_fact table, but only included rows where the date primary key was in quarter 3 of the year 2022.

*A screen shot of a computer

AI-generated content may be incorrect.*

*The sales\_fact table is used in conjunction with the time\_period dimension table to narrow down search results.*

## Task 2 Question 3

For this question, I included all of the hierarchies of staff and time, as well as quantity and price from the sales fact table to create the cube.

*A computer screen shot of a black screen

AI-generated content may be incorrect.*

***Query and result*** *of making a cube stored in a materialized view with staff and time hierarchies.*

## Task 2 Question 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| The state sales profits in each quarter of 2021 | | | | |
|  | **QLD** | **NSW** | **WA** | **SA** |
| 2021 Q1 | 73561844.64 | 89890611.04 | 69745631.52 | 69745631.52 |
| 2021 Q2 | 72304672.48 | 89222413.92 | 72783921.28 | 72939556.00 |
| 2021 Q3 | 74336622.40 | 91341606.24 | 71703951.20 | 71299514.56 |
| 2021 Q4 | 71776438.72 | 90140304.16 | 75456568.00 | 72323697.92 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| The state sales profits in each year | | | | |
|  | **QLD** | **NSW** | **WA** | **SA** |
| 2021 | 291979578.24 | 360594935.36 | 292142394.88 | 286308400.00 |
| 2022 | 289045992.32 | 360728529.76 | 298798987.84 | 283899422.88 |
| 2023 | 141054228.96 | 177465655.52 | 143314486.88 | 137340762.40 |

For this question, I made a view which includes profits from all of the years and quarters for the different states. I was in the process of making a second view for total profits for the year when I found that if the value of quarter is null, then the profit column will show the total profits for the year.

Screenshots of the queries to make and get information from the view is shown below. I could have used the view by itself to answer the questions but decided to query it to provide neater screenshots.

*A computer screen with white text

AI-generated content may be incorrect.*

*The query to create the view.*

*A screenshot of a computer

AI-generated content may be incorrect.*

*Querying the view to fill in the first table.*

*A screenshot of a computer

AI-generated content may be incorrect.*

*Querying the view to fill out the second table.*

## Task 2 Question 5

*A computer screen shot of a black screen

AI-generated content may be incorrect.*

*Query and result to create the sales\_product\_staff cube.*

## Task 2 Question 5.a

**277, 286, and 15 are the PID’s for the top 3 highest grossing stores**. To complete this question, I had to separately sum up gross profit for each of the products at each of the stores, as the sale\_price value would fluctuate regularly.

After doing this, I just had to sum up this gross profit value for each of the stores, order by gross profit descending, and limit it to three stores.

A screenshot of a computer screen

AI-generated content may be incorrect.

*The query and result for creating the top\_three\_gross\_profit\_stores view.*

## Task 2 Question 5.b

For this question, I used a similar strategy to above to get the gross profit of each product. I then had to get some help from ChatGPT (OpenAI, 2025) to partition these products and stores into groups, order them based on the sum of their gross profit, and then be assigned row numbers.

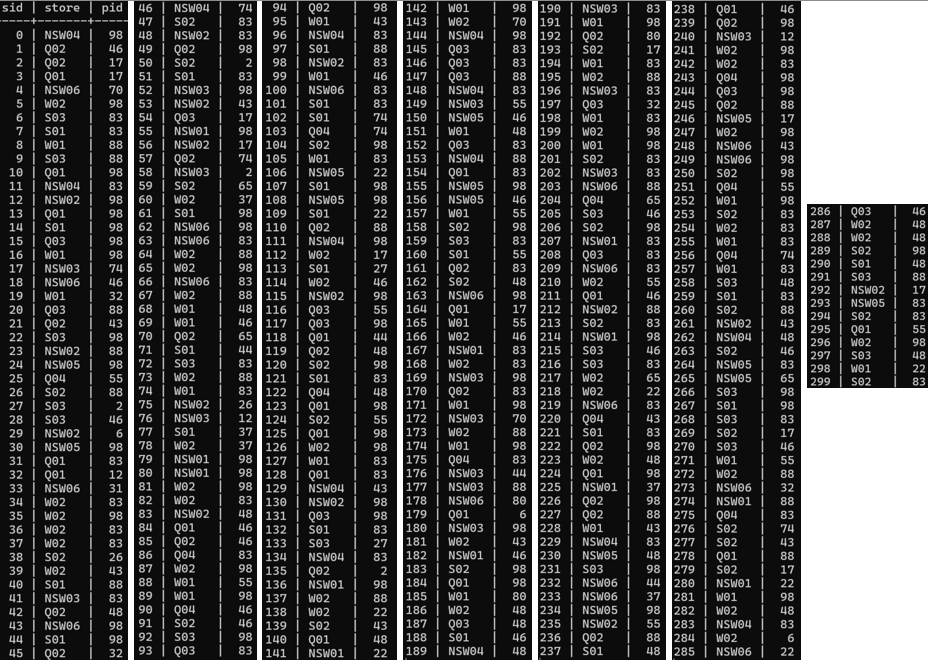
To create the view, I just took the product ID with the highest row value for each store.

I included screenshots of the query as well as the contents of the view, as the highest grossing product for each store was 300 rows and that seemed like too much to list here.

*A screen shot of a computer code

AI-generated content may be incorrect.*

*The result and query to create the top\_product\_per\_store view, which shows the most profitable product for each store, based on gross profit.*



*The contents of the top\_product\_per\_store view, which shows the store id, name, and the id of the most profitable product (based on gross profit).*

# References

OpenAI. (2025). ChatGPT (May 2025 version) [Large language model]. https://chat.openai.com/