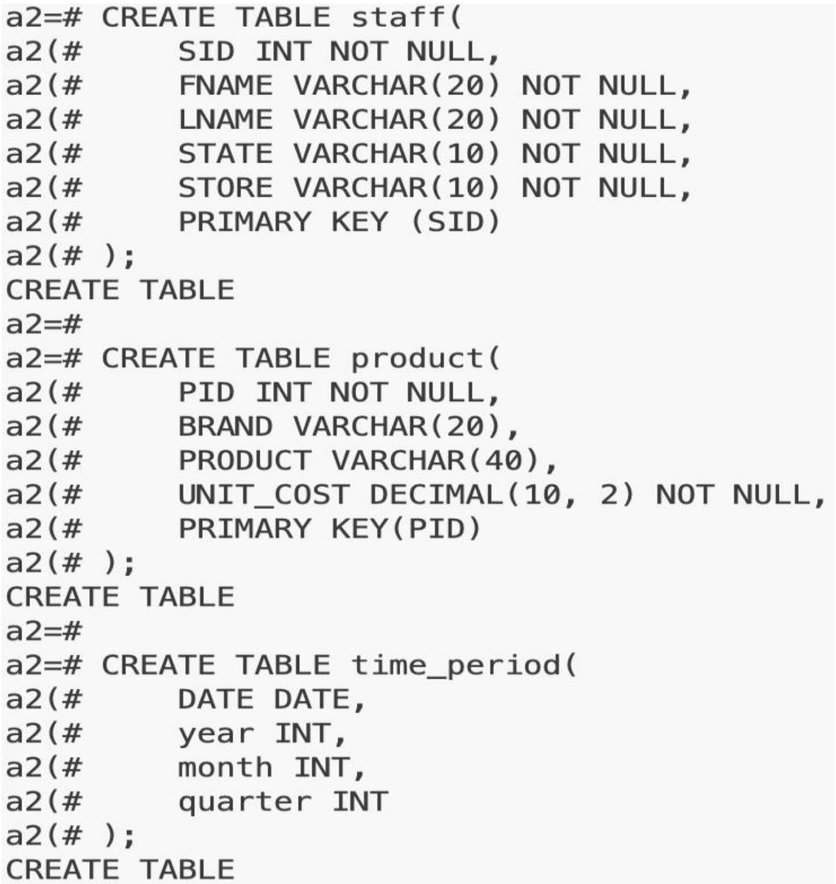
INFS3200 Advanced Database System Assignment 2

Student Name: Wei-Ting, Hong

Student Number: 47523483

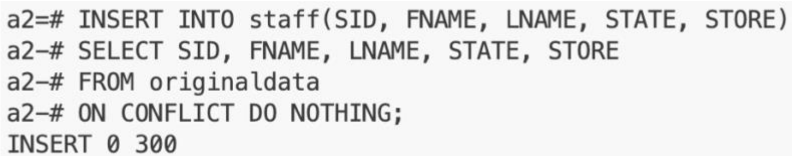
Task 2.

1. In order to separate attributes of raw data into dimension tables appropriately, it should make sense to distribute staff and store information to “staff” table, whereas put PID, BRAND, PRODUCT, UNIT_COST to “product” table. With respect to “time_period” table, the attributes are DATE, year, month, and quarter, just like the statement in task sheet. Corresponding to this design structure, we could create dimension tables by the following SQL.



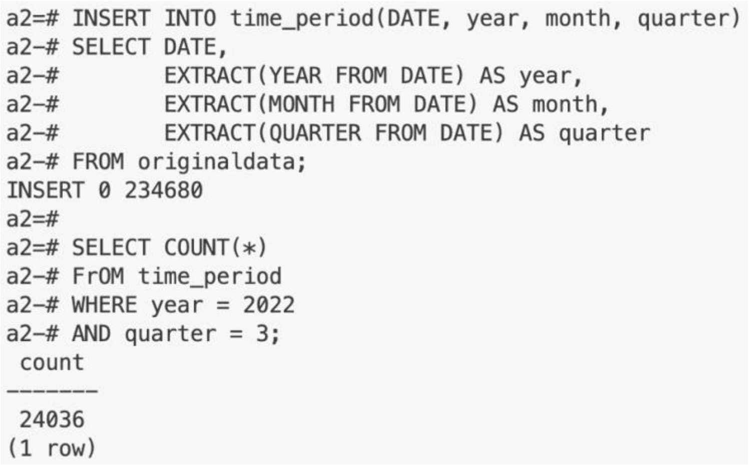
```
a2=# CREATE TABLE staff(
a2(#     SID INT NOT NULL,
a2(#     FNAME VARCHAR(20) NOT NULL,
a2(#     LNAME VARCHAR(20) NOT NULL,
a2(#     STATE VARCHAR(10) NOT NULL,
a2(#     STORE VARCHAR(10) NOT NULL,
a2(#     PRIMARY KEY (SID)
a2(# );
CREATE TABLE
a2=#
a2=# CREATE TABLE product(
a2(#     PID INT NOT NULL,
a2(#     BRAND VARCHAR(20),
a2(#     PRODUCT VARCHAR(40),
a2(#     UNIT_COST DECIMAL(10, 2) NOT NULL,
a2(#     PRIMARY KEY(PID)
a2(# );
CREATE TABLE
a2=#
a2=# CREATE TABLE time_period(
a2(#     DATE DATE,
a2(#     year INT,
a2(#     month INT,
a2(#     quarter INT
a2(# );
CREATE TABLE
```

2_a. By insert information into dimension table from raw data, the terminal return done message with “INSERT 300”. Since SID is a primary key, therefore, we could know there are 300 of employees in total.



```
a2=# INSERT INTO staff(SID, FNAME, LNAME, STATE, STORE)
a2-# SELECT SID, FNAME, LNAME, STATE, STORE
a2-# FROM originaldata
a2-# ON CONFLICT DO NOTHING;
INSERT 0 300
```

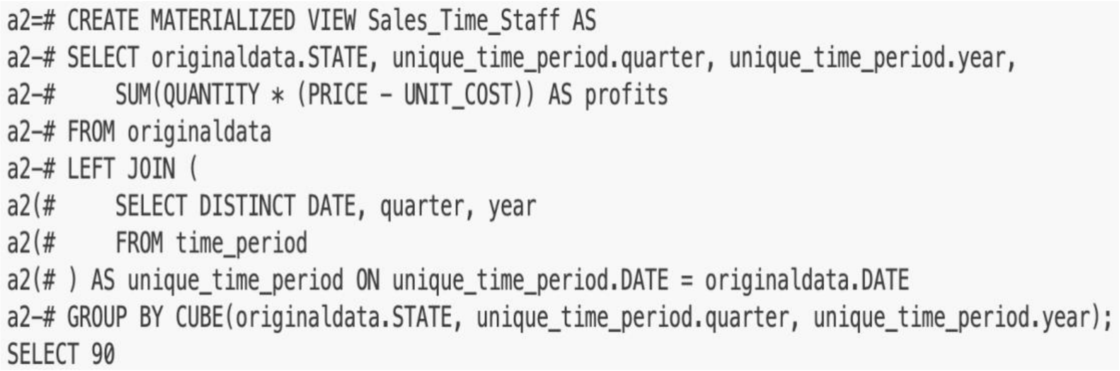
2_b.



```
a2=# INSERT INTO time_period(DATE, year, month, quarter)
a2-# SELECT DATE,
a2-#        EXTRACT(YEAR FROM DATE) AS year,
a2-#        EXTRACT(MONTH FROM DATE) AS month,
a2-#        EXTRACT(QUARTER FROM DATE) AS quarter
a2-# FROM originaldata;
INSERT 0 234680
a2=#
a2=# SELECT COUNT(*)
a2-# FrOM time_period
a2-# WHERE year = 2022
a2-# AND quarter = 3;
 count
-------
 24036
(1 row)
```

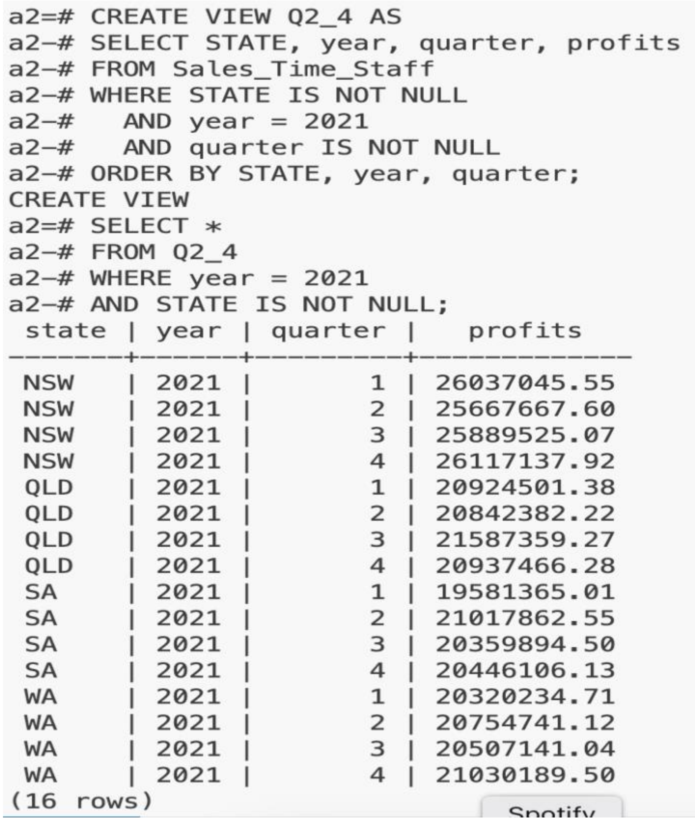
First, we fill up the dimension table related to time by proper SQL. Then do the second SQL to gain the answer specified by task sheet.

3. The following SQL create the specified materialized view. With using sub-query, we could significantly decrease creating time.



```
a2=# CREATE MATERIALIZED VIEW Sales_Time_Staff AS
a2-# SELECT originaldata.STATE, unique_time_period.quarter, unique_time_period.year,
a2-#     SUM(QUANTITY * (PRICE - UNIT_COST)) AS profits
a2-# FROM originaldata
a2-# LEFT JOIN (
a2(#     SELECT DISTINCT DATE, quarter, year
a2(#     FROM time_period
a2(# ) AS unique_time_period ON unique_time_period.DATE = originaldata.DATE
a2-# GROUP BY CUBE(originaldata.STATE, unique_time_period.quarter, unique_time_period.year);
SELECT 90
```

4.



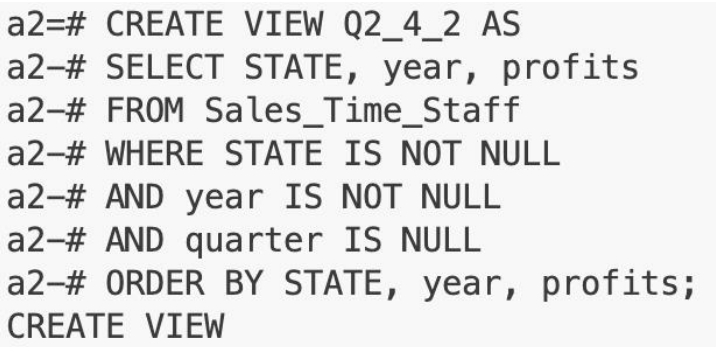
```
a2=# CREATE VIEW Q2_4 AS
a2-# SELECT STATE, year, quarter, profits
a2-# FROM Sales_Time_Staff
a2-# WHERE STATE IS NOT NULL
a2-#   AND year = 2021
a2-#   AND quarter IS NOT NULL
a2-# ORDER BY STATE, year, quarter;
CREATE VIEW
a2=# SELECT *
a2-# FROM Q2_4
a2-# WHERE year = 2021
a2-# AND STATE IS NOT NULL;
 state | year | quarter |   profits
-------+------+---------+-------------
 NSW   | 2021 |       1 | 26037045.55
 NSW   | 2021 |       2 | 25667667.60
 NSW   | 2021 |       3 | 25889525.07
 NSW   | 2021 |       4 | 26117137.92
 QLD   | 2021 |       1 | 20924501.38
 QLD   | 2021 |       2 | 20842382.22
 QLD   | 2021 |       3 | 21587359.27
 QLD   | 2021 |       4 | 20937466.28
 SA    | 2021 |       1 | 19581365.01
 SA    | 2021 |       2 | 21017862.55
 SA    | 2021 |       3 | 20359894.50
 SA    | 2021 |       4 | 20446106.13
 WA    | 2021 |       1 | 20320234.71
 WA    | 2021 |       2 | 20754741.12
 WA    | 2021 |       3 | 20507141.04
 WA    | 2021 |       4 | 21030189.50
(16 rows)
```

From the SQL above, we can complete the table on task sheet.

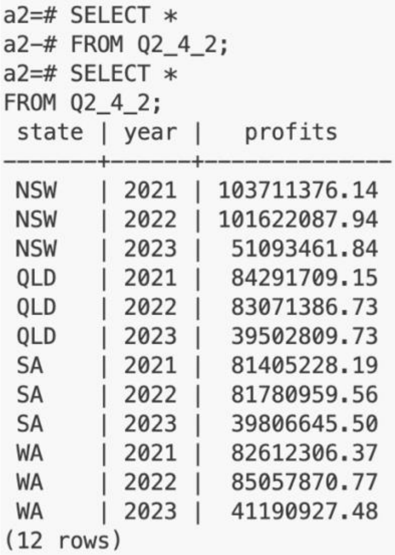
The state sales volumes in each quarter of 2021

	QLD	NSW	WA	SA
2021Q1	20924501.38	26037045.55	20320234.71	19581365.01
2021Q2	20842382.22	25667667.60	20754741.12	21017862.55
2021Q3	21587359.27	25889525.07	20507141.04	20359894.50
2021Q4	20937466.28	26117137.92	21030189.50	20446106.13

And for the second table:



```
a2=# CREATE VIEW Q2_4_2 AS
a2-# SELECT STATE, year, profits
a2-# FROM Sales_Time_Staff
a2-# WHERE STATE IS NOT NULL
a2-# AND year IS NOT NULL
a2-# AND quarter IS NULL
a2-# ORDER BY STATE, year, profits;
CREATE VIEW
```

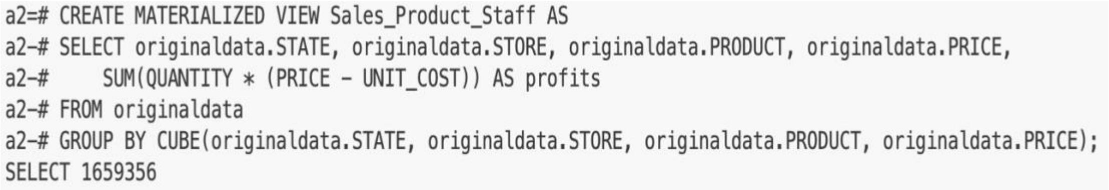


```
a2=# SELECT *
a2-# FROM Q2_4_2;
a2=# SELECT *
FROM Q2_4_2;
 state | year |   profits
-------+------+--------------
 NSW   | 2021 | 103711376.14
 NSW   | 2022 | 101622087.94
 NSW   | 2023 |  51093461.84
 QLD   | 2021 |  84291709.15
 QLD   | 2022 |  83071386.73
 QLD   | 2023 |  39502809.73
 SA    | 2021 |  81405228.19
 SA    | 2022 |  81780959.56
 SA    | 2023 |  39806645.50
 WA    | 2021 |  82612306.37
 WA    | 2022 |  85057870.77
 WA    | 2023 |  41190927.48
(12 rows)
```

The state sales volumes in each year.

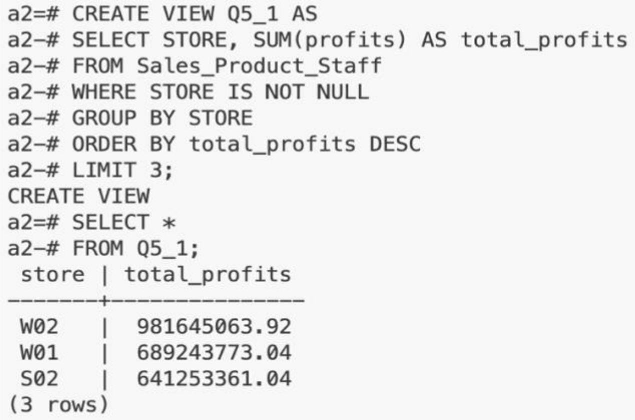
	QLD	NSW	WA	SA
2021	84291709.15	103711376.14	82612306.37	81405228.19
2022	83071386.73	101622087.94	85057870.77	81780959.56
2023	39502809.73	51093461.84	41190927.48	39806645.50

5_a. The SQL of the materialised view "Sales_Product_Staff" should be beloew.



```
a2=# CREATE MATERIALIZED VIEW Sales_Product_Staff AS
a2-# SELECT originaldata.STATE, originaldata.STORE, originaldata.PRODUCT, originaldata.PRICE,
a2-#     SUM(QUANTITY * (PRICE - UNIT_COST)) AS profits
a2-# FROM originaldata
a2-# GROUP BY CUBE(originaldata.STATE, originaldata.STORE, originaldata.PRODUCT, originaldata.PRICE);
SELECT 1659356
```

By creating specified view and select all data on it, we could obtain the answer.



```
a2=# CREATE VIEW Q5_1 AS
a2-# SELECT STORE, SUM(profits) AS total_profits
a2-# FROM Sales_Product_Staff
a2-# WHERE STORE IS NOT NULL
a2-# GROUP BY STORE
a2-# ORDER BY total_profits DESC
a2-# LIMIT 3;
CREATE VIEW
a2=# SELECT *
a2-# FROM Q5_1;
 store | total_profits
-------+---------------
 W02   |  981645063.92
 W01   |  689243773.04
 S02   |  641253361.04
(3 rows)
```