




# 8 Week SQL Challenge

Start your SQL learning journey today!



*click here* 

[Data With Danny](#)

[Case Studies](#)

[Getting Started](#)

[Resources](#)

[About](#)

## Case Study #5 - Data Mart

Danny Ma · June 20, 2021

8WeekSQLChallenge.com  
**CASE STUDY #5**



**DATA MART**  
fresh is best

DataWithDanny.com

## Introduction

Data Mart is Danny's latest venture and after running international operations for his online supermarket that specialises in fresh produce - Danny is asking for your support to analyse his sales performance.

In June 2020 - large scale supply changes were made at Data Mart. All Data Mart products now use sustainable packaging methods in every single step from the farm all the way to the customer.

Danny needs your help to quantify the impact of this change on the sales performance for Data Mart and its separate business areas.

The key business question he wants you to help him answer are the following:


- What was the quantifiable impact of the changes introduced in June 2020?
- Which platform, region, segment and customer types were the most impacted by this change?
- What can we do about future introduction of similar sustainability updates to the business to minimise impact on sales?

## Available Data

For this case study there is only a single table: `data_mart.weekly_sales`

The `Entity Relationship Diagram` is shown below with the data types made clear, please note that there is only this one table - hence why it looks a little bit lonely!

data_mart.weekly_sales	
week_date	VARCHAR(7)
region	VARCHAR(13)
platform	VARCHAR(7)
segment	VARCHAR(4)
customer_type	VARCHAR(8)
transactions	INTEGER
sales	INTEGER



## Column Dictionary

The columns are pretty self-explanatory based on the column names but here are some further details about the dataset:

1. Data Mart has international operations using a multi-**region** strategy
2. Data Mart has both, a retail and online **platform** in the form of a Shopify store front to serve their customers
3. Customer **segment** and **customer\_type** data relates to personal age and demographics information that is shared with Data Mart
4. **transactions** is the count of unique purchases made through Data Mart and **sales** is the actual dollar amount of purchases

Each record in the dataset is related to a specific aggregated slice of the underlying sales data rolled up into a **week\_date** value which represents the start of the sales week.

## Example Rows

10 random rows are shown in the table output below from **data\_mart.weekly\_sales**:

week_date	region	platform	segment	customer_type	transa
9/9/20	OCEANIA	Shopify	C3	New	610

week_date	region	platform	segment	customer_type	trans
22/7/20	EUROPE	Shopify	C4	Existing	24
13/5/20	AFRICA	Shopify	null	Guest	5287
24/7/19	ASIA	Retail	C1	New	12734
10/7/19	CANADA	Shopify	F3	New	51
26/6/19	OCEANIA	Retail	C3	New	15292
29/5/19	SOUTH AMERICA	Shopify	null	New	53
22/8/18	AFRICA	Retail	null	Existing	31721
25/7/18	SOUTH AMERICA	Retail	null	New	2136

## Interactive SQL Instance

You can use the embedded DB Fiddle below to easily access these example datasets - this interactive session has everything you need to start solving these questions using SQL.

You can click on the [Edit on DB Fiddle](#) link on the top right hand corner of the embedded session below and it will take you to a fully functional SQL editor where you can write your own queries to analyse the data.

You can feel free to choose any SQL dialect you'd like to use, the existing Fiddle is using PostgreSQL 13 as default.

Serious SQL students will have access to the same relevant schema SQL and example solutions which they can use with their Docker setup from within the course player!

Schema SQL Query SQL Results

Edit on DB Fiddle

```

CREATE SCHEMA data_mart;
SET search_path = data_mart;

DROP TABLE IF EXISTS data_mart.weekly_sales;
CREATE TABLE data_mart.weekly_sales (
  "week_date" VARCHAR(7),
  "region" VARCHAR(13),
  "platform" VARCHAR(7),
  "segment" VARCHAR(4),
  "customer_type" VARCHAR(8),
  "transactions" INTEGER,
  "sales" INTEGER
);

INSERT INTO data_mart.weekly_sales
("week_date", "region", "platform", "segment", "customer_type", "transactions", "sales")
VALUES
('31/8/20', 'ASIA', 'Retail', 'C3', 'New', '120631', '3656163'),
('31/8/20', 'ASIA', 'Retail', 'F1', 'New', '31574', '996575'),
('31/8/20', 'USA', 'Retail', 'null', 'Guest', '529151', '16509610'),
('31/8/20', 'EUROPE', 'Retail', 'C1', 'New', '4517', '141942'),
('31/8/20', 'AFRICA', 'Retail', 'C2', 'New', '58046', '1758388'),
('31/8/20', 'CANADA', 'Shopify', 'F2', 'Existing', '1336', '243878'),
('31/8/20', 'AFRICA', 'Shopify', 'F3', 'Existing', '2514', '519502'),
('31/8/20', 'ASIA', 'Shopify', 'F1', 'Existing', '2158', '371417'),
('31/8/20', 'AFRICA', 'Shopify', 'F2', 'New', '318', '49557'),
('31/8/20', 'AFRICA', 'Retail', 'C3', 'New', '111032', '3888162'),
('31/8/20', 'USA', 'Shopify', 'F1', 'Existing', '1398', '260773'),
('31/8/20', 'OCEANIA', 'Shopify', 'C2', 'Existing', '4661', '882690'),
('31/8/20', 'SOUTH AMERICA', 'Retail', 'C2', 'Existing', '1029', '38762'),
('31/8/20', 'SOUTH AMERICA', 'Shopify', 'C4', 'New', '6', '917'),
('31/8/20', 'EUROPE', 'Shopify', 'F3', 'Existing', '115', '35215'),
('31/8/20', 'OCEANIA', 'Retail', 'F3', 'Existing', '551905', '30371770'),
('31/8/20', 'ASIA', 'Shopify', 'C3', 'Existing', '1969', '374327'),
('31/8/20', 'AFRICA', 'Retail', 'F1', 'Existing', '97604', '5185233'),
('31/8/20', 'OCEANIA', 'Retail', 'C2', 'New', '111219', '2980673'),
('31/8/20', 'USA', 'Retail', 'F1', 'New', '11820', '463738'),
('31/8/20', 'SOUTH AMERICA', 'Retail', 'F3', 'Existing', '1363', '65730'),
('31/8/20', 'AFRICA', 'Retail', 'C3', 'Existing', '284971', '14430196'),
('31/8/20', 'ASIA', 'Retail', 'F2', 'New', '70496', '2176980'),
('31/8/20', 'AFRICA', 'Shopify', 'F1', 'Existing', '2678', '478756'),
('31/8/20', 'USA', 'Shopify', 'C4', 'New', '22', '3319'),
('31/8/20', 'CANADA', 'Retail', 'F3', 'Existing', '94274', '5306746'),

```

## Case Study Questions

The following case study questions require some data cleaning steps before we start to unpack Danny's key business questions in more depth.

### 1. Data Cleansing Steps

In a single query, perform the following operations and generate a new table in the `data_mart` schema named `clean_weekly_sales`:

- Convert the `week_date` to a `DATE` format
- Add a `week_number` as the second column for each `week_date` value, for example any value from the 1st of January to 7th of January will be 1, 8th to 14th will be 2 etc
- Add a `month_number` with the calendar month for each `week_date` value as the 3rd column
- Add a `calendar_year` column as the 4th column containing either 2018, 2019 or 2020 values
- Add a new column called `age_band` after the original `segment` column using the following mapping on the number inside the `segment` value

segment	age_band
1	Young Adults
2	Middle Aged
3 or 4	Retirees

- Add a new `demographic` column using the following mapping for the first letter in the `segment` values:

segment	demographic
C	Couples
F	Families

- Ensure all `null` string values with an `"unknown"` string value in the original `segment` column as well as the new `age_band` and `demographic` columns
- Generate a new `avg_transaction` column as the `sales` value divided by `transactions` rounded to 2 decimal places for each record

## 2. Data Exploration

1. What day of the week is used for each `week_date` value?
2. What range of week numbers are missing from the dataset?
3. How many total transactions were there for each year in the dataset?
4. What is the total sales for each region for each month?
5. What is the total count of transactions for each platform
6. What is the percentage of sales for Retail vs Shopify for each month?
7. What is the percentage of sales by demographic for each year in the dataset?
8. Which `age_band` and `demographic` values contribute the most to Retail sales?
9. Can we use the `avg_transaction` column to find the average transaction size for each year for Retail vs Shopify? If not - how would you calculate it instead?

### 3. Before & After Analysis

This technique is usually used when we inspect an important event and want to inspect the impact before and after a certain point in time.

Taking the `week_date` value of `2020-06-15` as the baseline week where the Data Mart sustainable packaging changes came into effect.

We would include all `week_date` values for `2020-06-15` as the start of the period **after** the change and the previous `week_date` values would be **before**

Using this analysis approach - answer the following questions:

1. What is the total sales for the 4 weeks before and after `2020-06-15`? What is the growth or reduction rate in actual values and percentage of sales?
2. What about the entire 12 weeks before and after?
3. How do the sale metrics for these 2 periods before and after compare with the previous years in 2018 and 2019?

### 4. Bonus Question

Which areas of the business have the highest negative impact in sales metrics performance in 2020 for the 12 week before and after period?

- `region`
- `platform`

- `age_band`
- `demographic`
- `customer_type`

Do you have any further recommendations for Danny's team at Data Mart or any interesting insights based off this analysis?

## Conclusion

This case study actually is based off a real life change in Australia retailers where plastic bags were no longer provided for free - as you can expect, some customers would have changed their shopping behaviour because of this change!

Analysis which is related to certain key events which can have a significant impact on sales or engagement metrics is always a part of the data analytics menu. Learning how to approach these types of problems is a super valuable lesson and hopefully these ideas can help you next time you're faced with a tough problem like this in the workplace!

Ready for the next 8 Week SQL challenge case study? Click on the banner below to get started with case study #6!



8WEEKSQLCHALLENGE.COM  
CASE STUDY # 6



CliqueBait  
ATTENTION CAPTURING

DATAWITHDANNY.COM

### Official Solutions

If you'd like to see the official code solutions and explanations for this case study and a whole lot more, please consider joining me for the [Serious SQL course](#) - you'll get access to all course materials and I'm on hand to answer all of your additional SQL questions directly!

[Serious SQL](#) is priced at \$49USD and \$29 for students and includes access to all written course content, community events as well as live and recorded SQL training videos!

Please send an email to [support@datawithdanny.com](mailto:support@datawithdanny.com) from your educational email or include your enrolment details or student identification for a speedy response!

## Community Solutions

This section will be updated in the future with any community member solutions with a link to their respective GitHub repos!

## Final Thoughts

The 8 Week SQL Challenge is proudly brought to you by me - Danny Ma and the [Data With Danny](#) virtual data apprenticeship program.

Students or anyone undertaking further studies are eligible for a \$20USD student discount off the price of Serious SQL please send an email to [support@datawithdanny.com](mailto:support@datawithdanny.com) from your education email or include information about your enrolment for a fast response!

We have a large student community active on the official DWD Discord server with regular live events, trainings and workshops available to all Data With Danny students, plus early discounted access to all future paid courses.

There are also opportunities for 1:1 mentoring, resume reviews, interview training and more from myself or others in the DWD Mentor Team.

From your friendly data mentor, Danny :)

## All 8 Week SQL Challenge Case Studies

All of the 8 Week SQL Challenge case studies can be found below:

- [Case Study #1 - Danny's Diner](#)
- [Case Study #2 - Pizza Runner](#)
- [Case Study #3 - Foodie-Fi](#)
- [Case Study #4 - Data Bank](#)
- [Case Study #5 - Data Mart](#)
- [Case Study #6 - Clique Bait](#)
- [Case Study #7 - Balanced Tree Clothing Co.](#)

- [Case Study #8 - Fresh Segments](#)

**Share:** [Twitter](#), [Facebook](#)