**SQL Case Study 1: Data Mart Analysis**



**INTRODUCTION:**

Data Dart is my latest venture and I want your help to analyze the sales and performance of my venture. 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.

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

**SCHEMA USED: WEEKLY\_SALES TABLE**

|  |  |
| --- | --- |
| Column name | Data type |
| week\_date | date |
| region | varchar(20) |
| platform | varchar(20) |
| segment | varchar(10) |
| customer | varchar(20) |
| transactions | int |
| sales | int |

**CASE STUDY QUESTIONS**

## 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:

1. 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.
2. Add a month\_number with the calendar month for each week\_date value as the 3rd column
3. Add a calendar\_year column as the 4th column containing either 2018, 2019 or 2020 values
4. 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 |

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

segment | demographic |  
C | Couples |  
F | Families |

1. 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
2. Generate a new avg\_transaction column as the sales value divided by transactions rounded to 2 decimal places for each record

SOLUTION:

create table clean\_weekly\_sales as

select week\_date,

week(week\_date) as week\_number,

month(week\_date) as month\_number,

year(week\_date) as calender\_year,

region, platform,

case

when segment=null then 'Unknown'

else segment

end as segment,

case

when right(segment,1)='1' then 'Young Adult'

when right(segment,1)='2' then 'Middle Aged'

when right(segment,1) in ('3','4') then 'Retirees'

else 'Unknown'

end as age\_band,

case

when left(segment,1)='C' then 'Couples'

when left(segment,1)='F' then 'Families'

else 'Unknown'

end as demographic,

customer\_type, transactions, sales,

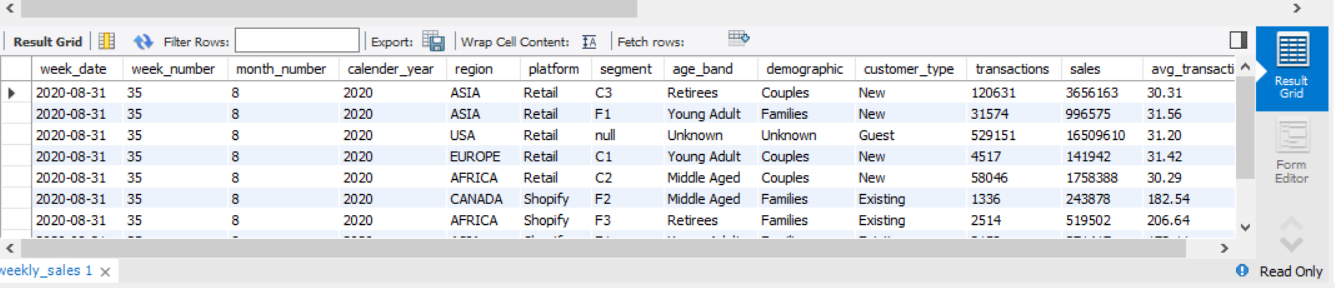
round(sales/transactions,2) as 'avg\_transaction'

from weekly\_sales;

**OUTPUT:**

select \* from clean\_weekly\_sales

limit 10;



## B. Data Exploration

1. Which week numbers are missing from the dataset?
2. How many total transactions were there for each year in the dataset?
3. What are the total sales for each region for each month?
4. What is the total count of transactions for each platform
5. What is the percentage of sales for Retail vs Shopify for each month?
6. What is the percentage of sales by demographic for each year in the dataset?
7. Which age\_band and demographic values contribute the most to Retail sales?

SOLUTION:

**1**.

create table seq100

(x int not null auto\_increment primary key);

insert into seq100 values (),(),(),(),(),(),(),(),(),();

insert into seq100 values (),(),(),(),(),(),(),(),(),();

insert into seq100 values (),(),(),(),(),(),(),(),(),();

insert into seq100 values (),(),(),(),(),(),(),(),(),();

insert into seq100 values (),(),(),(),(),(),(),(),(),();

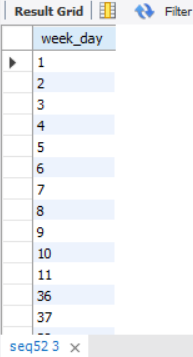
insert into seq100 select x + 50 from seq100;

select \* from seq100;

create table seq52 as (select x from seq100 limit 52);

select distinct x as week\_day from seq52 where x not in(select distinct week\_number from clean\_weekly\_sales);

OUTPUT:



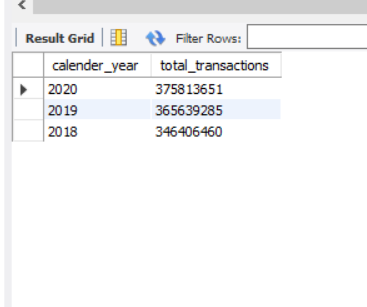
**2**. select calender\_year,

sum(transactions) as total\_transactions

from clean\_weekly\_sales

group by calender\_year;

**OUTPUT:**



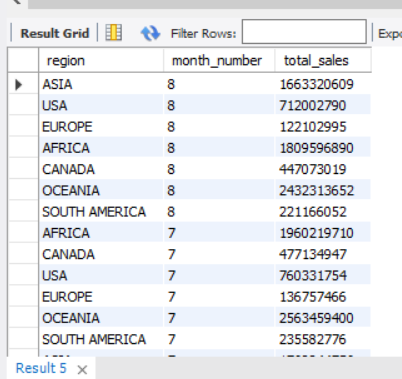
**3.** select region, month\_number,

sum(sales) as 'total\_sales'

from clean\_weekly\_sales

group by month\_number, region;

**OUTPUT:**

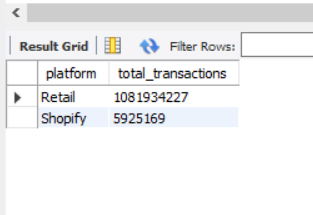


**4.** select platform, sum(transactions) as total\_transactions

from clean\_weekly\_sales

group by platform;

**OUTPUT:**



**5.** with cte\_monthly\_platform\_sales as (

select month\_number, calender\_year, platform, sum(sales) as monthly\_sales

from clean\_weekly\_sales

group by month\_number, calender\_year, platform)

select month\_number, calender\_year,

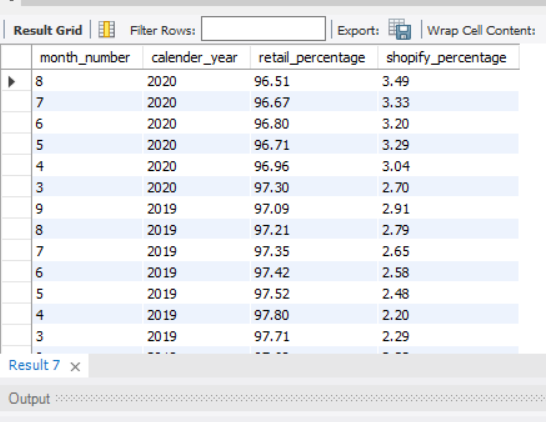
round(100\*max(case when platform='Retail' then monthly\_sales else null end)/sum(monthly\_sales),2) as retail\_percentage,

round(100\*max(case when platform='Shopify' then monthly\_sales else null end)/sum(monthly\_sales),2) as shopify\_percentage

from cte\_monthly\_platform\_sales

group by month\_number, calender\_year;

**OUTPUT:**



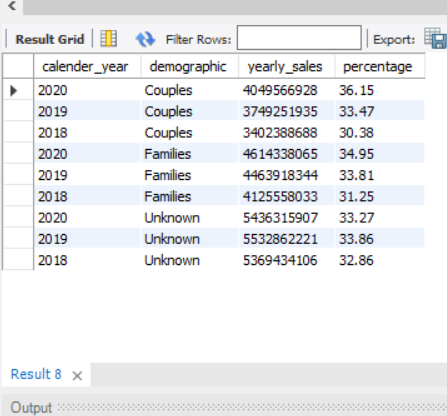
**6.** select calender\_year, demographic ,sum(sales) as yearly\_sales,

round(100\*sum(sales)/sum(sum(sales)) over (partition by demographic),2) as percentage

from clean\_weekly\_sales

group by calender\_year, demographic;

**OUTPUT:**



**7.** select age\_band, demographic, sum(sales) as total\_sales

from clean\_weekly\_sales

where platform='Retail'

group by age\_band, demographic

order by total\_sales desc;

**OUTPUT:**

