19/02/2022, 14:34 OneNote

Case study –5

Monday, February 7, 2022 11:30 AM

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
С	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

First I create A new table named as Clean_weekly_sales

```
CREATE TABLE clean_weekly_sales as (select * from data_mart.weekly_sales);
     -- select * from clean_weekly_sales;
```

ALTER Table clean_weekly_sales alter column week_date TYPE date using to_date(week_date,'DD-MM-YYYY');

ALTER table clean_weekly_sales add column week_number int NOT NULL DEFAULT(0); UPDATE clean_weekly_sales SET week_number=extract(weekday from week_date);

ALTER table clean_weekly_sales add column month_number int NOT NULL DEFAULT(0); UPDATE clean_weekly_sales SET month_number=extract(month from week_date);

ALTER table clean_weekly_sales add column year_number int NOT NULL DEFAULT(0); UPDATE clean_weekly_sales SET year_number=extract(year from week_date);

ALTER table clean weekly sales add column age band varchar DEFAULT 'J';

```
UPDATE clean_weekly_sales SET age_band = CASE
WHEN right(segment,1) = '1' THEN 'Young Adults'
                      WHEN right(segment,1) = '2' THEN 'Middle Aged'
                      WHEN right(segment,1) = '3' OR
                      right(segment,1) = '4' THEN 'Retirees'
                      ELSE NULL
```

19/02/2022, 14:34 OneNote

END;

Choose file No file chosen

```
Choose file No file chosen ALTER table clean_weekly_sales add column demographic varchar DEFAULT 'J';
             UPDATE clean_weekly_sales SET demographic = CASE
             WHEN left(segment,1) = 'C' THEN 'Couples'
                                     WHEN left(segment,1) = 'F' THEN 'Families'
                                     ELSE NULL
                                    END;
```

UPDATE clean_weekly_sales SET segment = CASE segment WHEN 'null' then 'unkn' else segment END, age_band = CASE age_band WHEN 'null' then 'unkn' else age_band END, demographic = CASE demographic WHEN 'null' then 'unkn' else demographic END;

ALTER table clean_weekly_sales add column avg_transaction int NOT NULL DEFAULT(0); UPDATE clean_weekly_sales SET avg_transaction=Round((sales/transactions),2);

select * from clean_weekly_sales;

Data Exploration

1. What day of the week is used for each week_date value?

select Distinct(to_char(week_date,'Day')) As DAY from clean_weekly_sales;



2. What range of week numbers are missing from the dataset?

select Distinct(52 – week_number) as Missing_Range from clean_weekly_sales



3. How many total transactions were there for each year in the dataset?

select Distinct(extract(year from week_date)) as Year, count(transactions) as Total_transactions from clean_weekly_sales group by year order by year;



20 5711

4. What is the total sales for each region for each month?

 $select\ Distinct(region)\ as\ Regions, to_char(week_date, 'month')\ AS\ Month,\ sum(sales)\ from\ clean_weekly_sales\ group\ by\ Regions, Month\ order\ by\ Regions;$



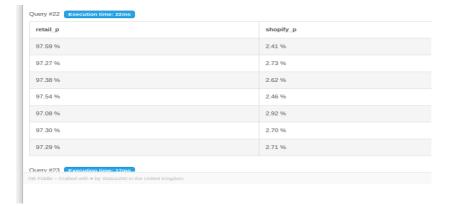
5. What is the total count of transactions for each platform

 $select\ Distinct (Platform)\ as\ Platforms, count (transactions)\ as\ Transactions\ from\ clean_weekly_sales\ group\ by\ Platforms;$



6. What is the percentage of sales for Retail vs Shopify for each month?

select round((sum(CASE WHEN platform='Retail' then sales



7. What is the percentage of sales by demographic for each year in the dataset?

select round((sum(CASE WHEN demographic='Couples' then sales ${\tt ELSE~0}$

19/02/2022, 14:34 OneNote

```
END)::NUMERIC/
sum(sales))* 100,2) || ' %' as Couples_p,
round((sum(CASE WHEN demographic='Families' then sales
   ELSE 0
END)::NUMERIC/
sum(sales))* 100,2) | | '%' as Families_p,
extract(year from week_date) as year from clean_weekly_sales group by year;
```



8. Which age_band and demographic values contribute the most to Retail sales?

select age band, demographic from clean weekly sales where platform='Retail' group by ${\tt age_band,platform,demographic\ having\ age_band\ is\ not\ null\ order\ by\ platform\ DESC\ LIMIT\ 1;}$



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?

Select sum(sales)/sum(transactions), extract(year from week_date) as year from clean_weekly_sales where platform='Retail' group by year; select sum(sales)/sum(transactions), extract(year from week_date) as year from clean_weekly_sales where platform='Shopify' group by year;



Before & After Analysis

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?

select

(select sum(sales) from clean_Weekly_sales where week_date_IN (select '0020-06-15T00:00:00.000Z'::date + INTERVAL '4 week' as date from clean_weekly_sales)) As After_4_week,

(select sum(sales) from clean_Weekly_sales where week_date_IN (select '0020-06-15T00:00:00.000Z'::date - INTERVAL '4 week' as date from clean_weekly_sales)) As Before_4_week; 19/02/2022, 14:34 OneNote



select week_date,

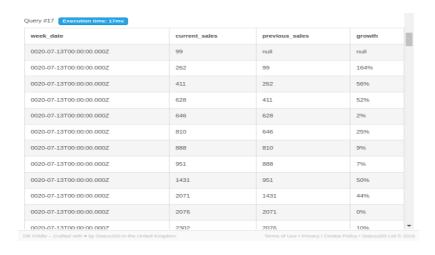
sum(sales) as Current_sales,

lag(sum(sales), 1) over (order by week_date) as Previous_sales,

(100 * (sum(sales) - lag(sum(sales), 1) over (order by week_date)) / lag(sum(sales), 1) over

(order by sales)) | | '%' as growth from clean_Weekly_sales

 $where \ week_date \ IN \ (select \ '0020-06-15T00:00:00.000Z'::date + INTERVAL \ '4 \ week' \ as \ date from \ clean_weekly_sales) \ group \ by \ week_date, sales;$



select

(select Round(((a.sales_week)::Numeric/b.total_sales)*100,2) as Percent from

(select sum(sales) as sales_week from clean_Weekly_sales where week_date IN (select '0020-06-15T00:00:00.000Z'::date + INTERVAL '4 week' as date from clean_weekly_sales) group by week_date)a,

(select sum(sales) as total_sales from clean_weekly_sales)b)As After_4_week,

(select Round(((a.sales_week)::Numeric/b.total_sales)*100,2) as Percent from

(select sum(sales) as sales_week from clean_Weekly_sales where week_date IN (select '0020-06-15T00:00:00.000Z'::date + INTERVAL '4 week' as date from clean_weekly_sales) group by week_date)a,

(select sum(sales) as total_sales from clean_weekly_sales)b)As after_4_week;



2. What about the entire 12 weeks before and after?

select

(select sum(sales) from clean_Weekly_sales where week_date IN (select '0020-06-15T00:00:00.000Z'::date + INTERVAL '12 week' as date from clean_weekly_sales))as After_12, (select sum(sales) from clean_Weekly_sales where week_date IN (select '0020-06-15T00:00:00.000Z'::date - INTERVAL '12 week' as date from clean_weekly_sales))as Before_12;



3. How do the sale metrics for these 2 periods before and after compare with the previous years in 2018 and 2019?

WITH myconstants (total) as (values (40743634227))

 $\hbox{-- select sum (sales), Round ((sum (sales)::Numeric/total)*100,2) as Percentage, extract (year from the sales) and the sales of the$ week_date) as year from clean_weekly_sales,myconstants where week_date IN (select '0020-06-15T00:00:00.000Z'::date + INTERVAL '4 week' as date from clean_weekly_sales) group by year, myconstants.total;

select sum(sales), Round((sum(sales)::Numeric/total)*100,2) as Percentage, extract(year from week_date) as year from clean_weekly_sales,myconstants where week_date IN (select '0020-06-15T00:00:00.000Z'::date - INTERVAL '4 week' as date from clean_weekly_sales) group by year, myconstants.total;