# ***Comprehensive Analysis and Data Cleansing of Weekly Sales Data in a Retail Environment***

## 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:

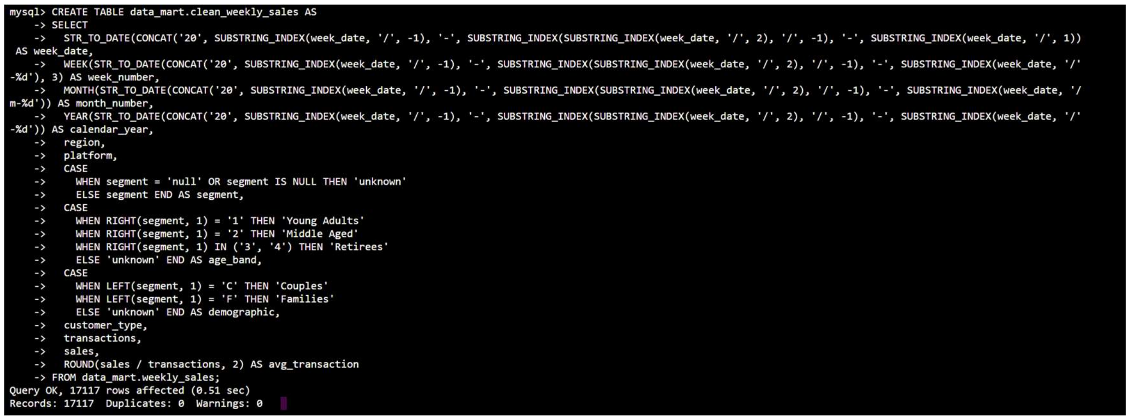
**1**. Convert the week_date to a DATE format
**2**. Add a week_number as the second column for each week_date value, for example any value from the 1st of January to 7th of January will be 1, 8th to 14th will be 2 etc
**3**. Add a month_number with the calendar month for each week_date value as the 3rd column
**4**. Add a calendar_year column as the 4th column containing either 2018, 2019 or 2020 values
**5**. 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 |

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

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

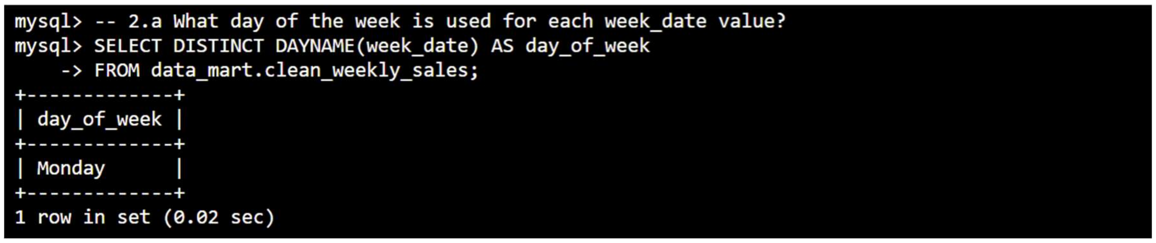
**7**. Ensure all null string values with an "unknown" string value in the original segment column as well as the new age_band and demographic columns
**8**. Generate a new avg_transaction column as the sales value divided by transactions rounded to 2 decimal places for each record



```
mysql> CREATE TABLE data_mart.clean_weekly_sales AS
    -> SELECT
    ->   STR_TO_DATE(CONCAT('20', SUBSTRING_INDEX(week_date, '/', -1), '-', SUBSTRING_INDEX(SUBSTRING_INDEX(week_date, '/', 2), '/', -1), '-', SUBSTRING_INDEX(week_date, '/', 1))
 AS week_date,
    ->   WEEK(STR_TO_DATE(CONCAT('20', SUBSTRING_INDEX(week_date, '/', -1), '-', SUBSTRING_INDEX(SUBSTRING_INDEX(week_date, '/', 2), '/', -1), '-', SUBSTRING_INDEX(week_date, '/'
-%d'), 3) AS week_number,
    ->   MONTH(STR_TO_DATE(CONCAT('20', SUBSTRING_INDEX(week_date, '/', -1), '-', SUBSTRING_INDEX(SUBSTRING_INDEX(week_date, '/', 2), '/', -1), '-', SUBSTRING_INDEX(week_date, '/
m-%d')) AS month_number,
    ->   YEAR(STR_TO_DATE(CONCAT('20', SUBSTRING_INDEX(week_date, '/', -1), '-', SUBSTRING_INDEX(SUBSTRING_INDEX(week_date, '/', 2), '/', -1), '-', SUBSTRING_INDEX(week_date, '/'
-%d')) AS calendar_year,
    ->   region,
    ->   platform,
    ->   CASE
    ->     WHEN segment = 'null' OR segment IS NULL THEN 'unknown'
    ->     ELSE segment END AS segment,
    ->   CASE
    ->     WHEN RIGHT(segment, 1) = '1' THEN 'Young Adults'
    ->     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 data_mart.weekly_sales;
Query OK, 17117 rows affected (0.51 sec)
Records: 17117  Duplicates: 0  Warnings: 0
```

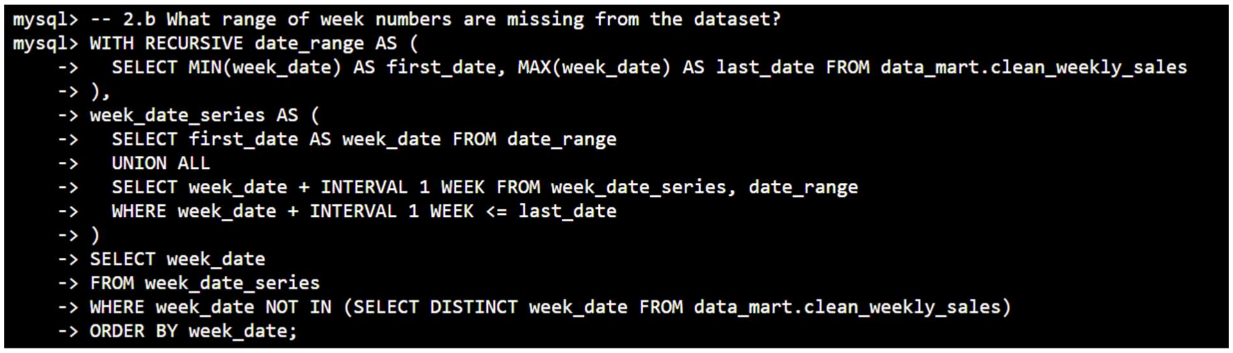
## 2. DATA exploration

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

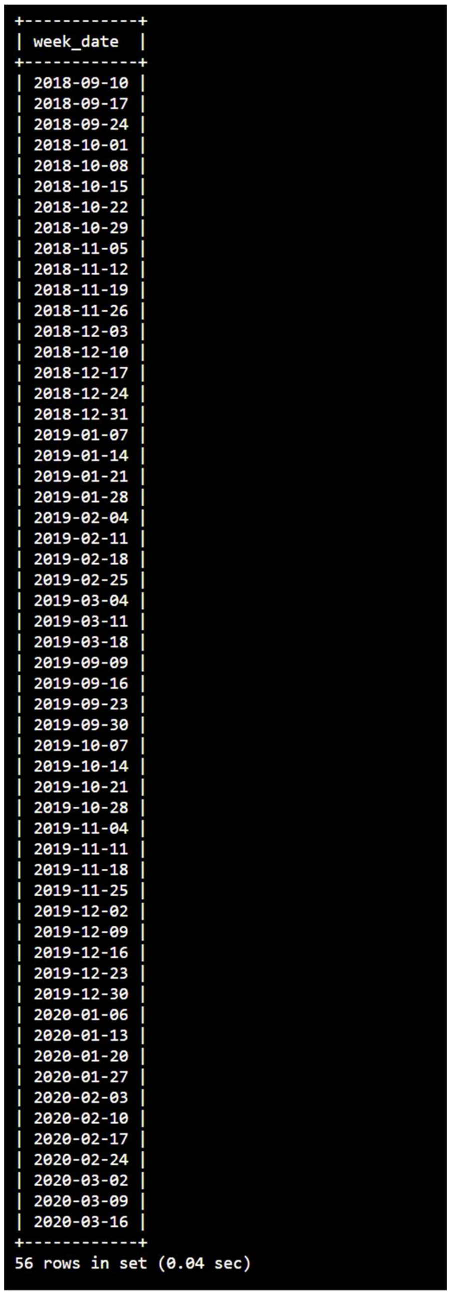


```
mysql> -- 2.a What day of the week is used for each week_date value?
mysql> SELECT DISTINCT DAYNAME(week_date) AS day_of_week
    -> FROM data_mart.clean_weekly_sales;
+-------------+
| day_of_week |
+-------------+
| Monday      |
+-------------+
1 row in set (0.02 sec)
```

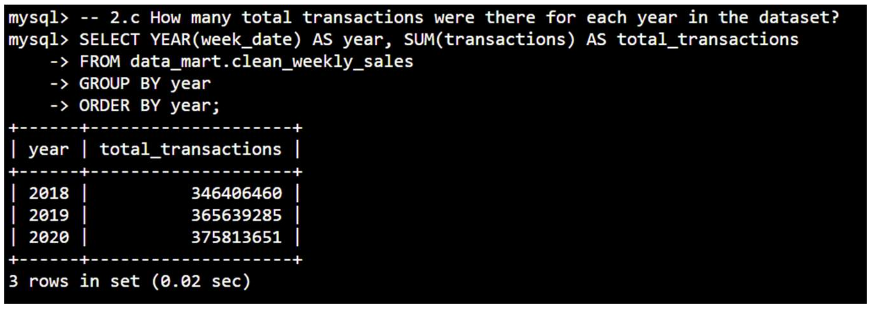
**What range of week numbers are missing from the dataset?**



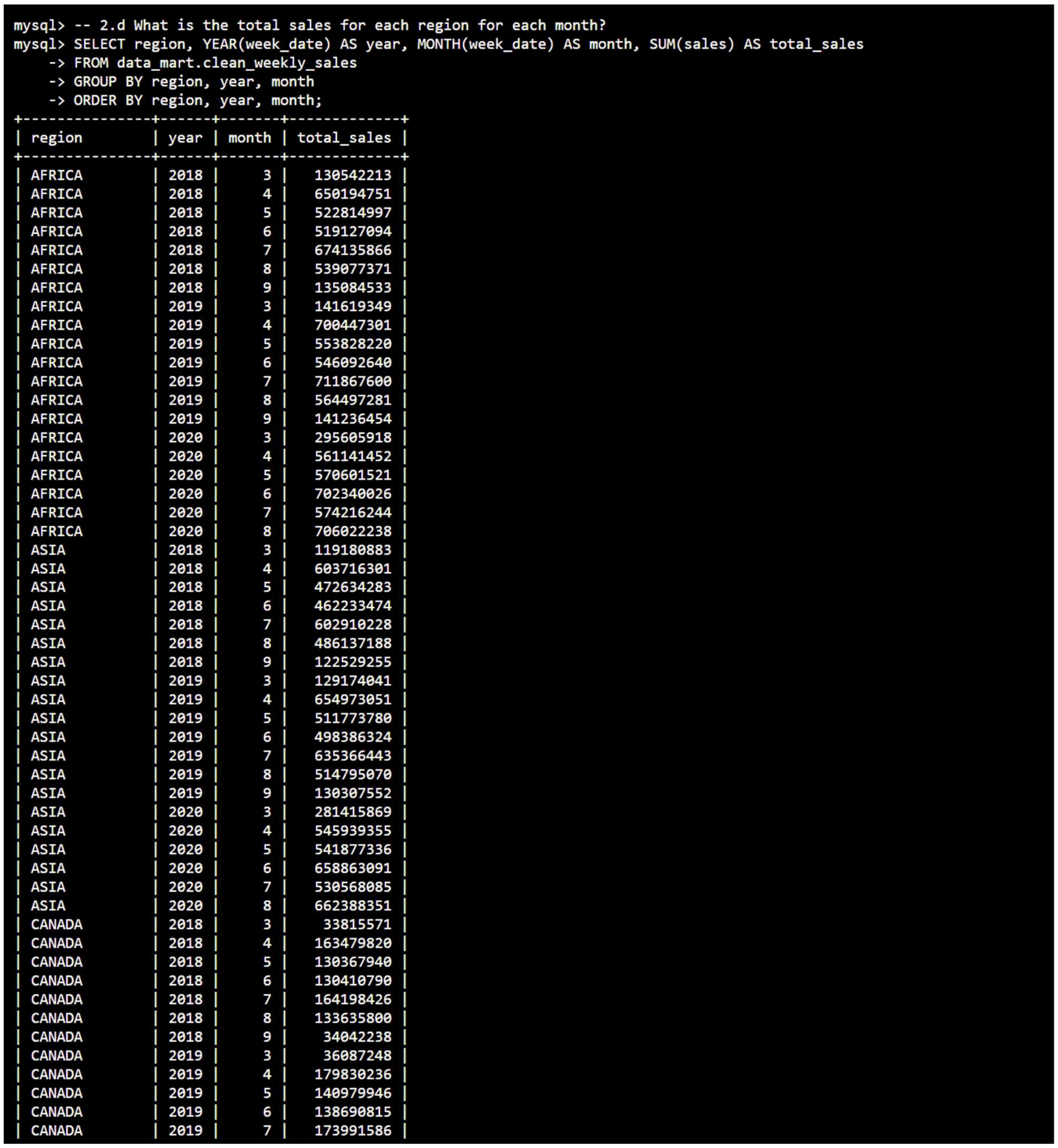
```
mysql> -- 2.b What range of week numbers are missing from the dataset?
mysql> WITH RECURSIVE date_range AS (
    ->   SELECT MIN(week_date) AS first_date, MAX(week_date) AS last_date FROM data_mart.clean_weekly_sales
    -> ),
    -> week_date_series AS (
    ->   SELECT first_date AS week_date FROM date_range
    ->   UNION ALL
    ->   SELECT week_date + INTERVAL 1 WEEK FROM week_date_series, date_range
    ->   WHERE week_date + INTERVAL 1 WEEK <= last_date
    -> )
    -> SELECT week_date
    -> FROM week_date_series
    -> WHERE week_date NOT IN (SELECT DISTINCT week_date FROM data_mart.clean_weekly_sales)
    -> ORDER BY week_date;
```


```
+------------+
| week_date  |
+------------+
| 2018-09-10 |
| 2018-09-17 |
| 2018-09-24 |
| 2018-10-01 |
| 2018-10-08 |
| 2018-10-15 |
| 2018-10-22 |
| 2018-10-29 |
| 2018-11-05 |
| 2018-11-12 |
| 2018-11-19 |
| 2018-11-26 |
| 2018-12-03 |
| 2018-12-10 |
| 2018-12-17 |
| 2018-12-24 |
| 2018-12-31 |
| 2019-01-07 |
| 2019-01-14 |
| 2019-01-21 |
| 2019-01-28 |
| 2019-02-04 |
| 2019-02-11 |
| 2019-02-18 |
| 2019-02-25 |
| 2019-03-04 |
| 2019-03-11 |
| 2019-03-18 |
| 2019-09-09 |
| 2019-09-16 |
| 2019-09-23 |
| 2019-09-30 |
| 2019-10-07 |
| 2019-10-14 |
| 2019-10-21 |
| 2019-10-28 |
| 2019-11-04 |
| 2019-11-11 |
| 2019-11-18 |
| 2019-11-25 |
| 2019-12-02 |
| 2019-12-09 |
| 2019-12-16 |
| 2019-12-23 |
| 2019-12-30 |
| 2020-01-06 |
| 2020-01-13 |
| 2020-01-20 |
| 2020-01-27 |
| 2020-02-03 |
| 2020-02-10 |
| 2020-02-17 |
| 2020-02-24 |
| 2020-03-02 |
| 2020-03-09 |
| 2020-03-16 |
+------------+
56 rows in set (0.04 sec)
```

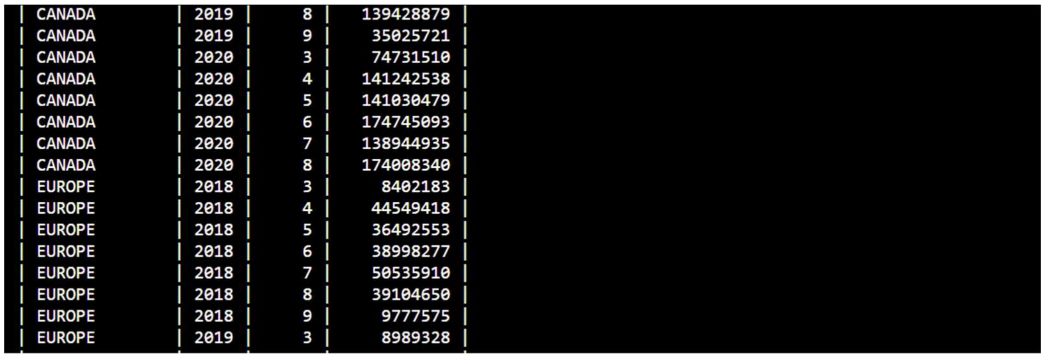
## How many total transactions were there for each year in the dataset?


```
mysql> -- 2.c How many total transactions were there for each year in the dataset?
mysql> SELECT YEAR(week_date) AS year, SUM(transactions) AS total_transactions
    -> FROM data_mart.clean_weekly_sales
    -> GROUP BY year
    -> ORDER BY year;
+------+--------------------+
| year | total_transactions |
+------+--------------------+
| 2018 |          346406460 |
| 2019 |          365639285 |
| 2020 |          375813651 |
+------+--------------------+
3 rows in set (0.02 sec)
```

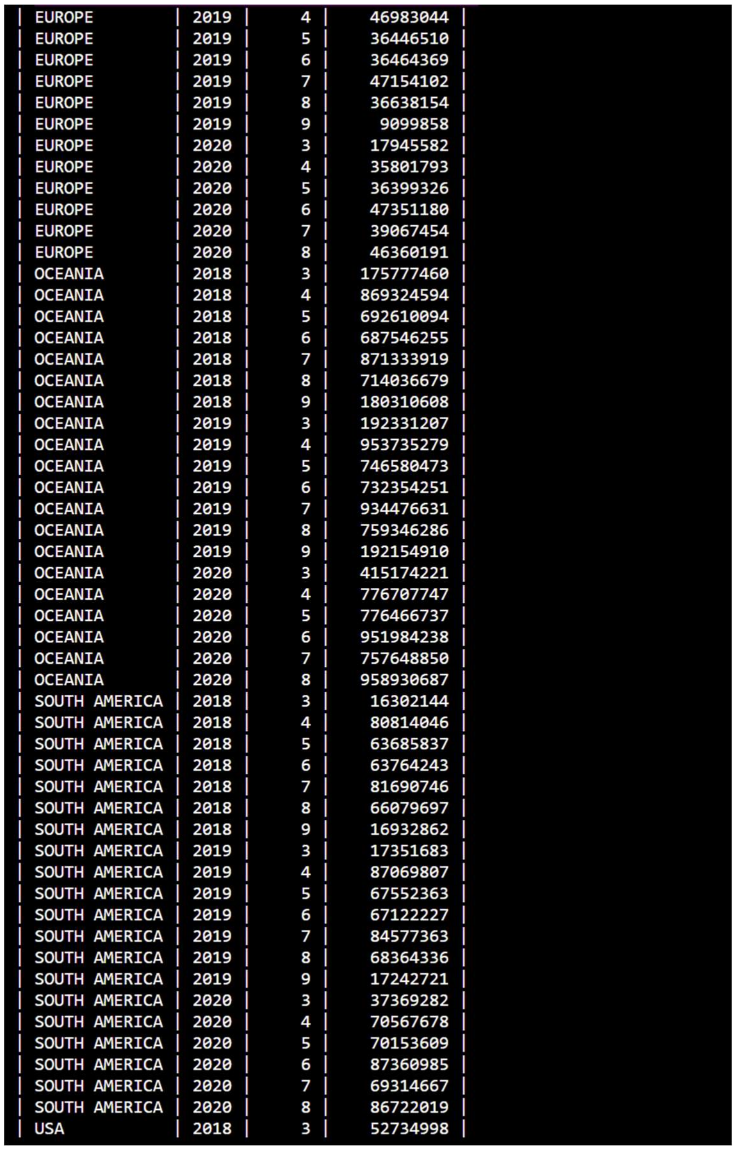
## What is the total sales for each region for each month?



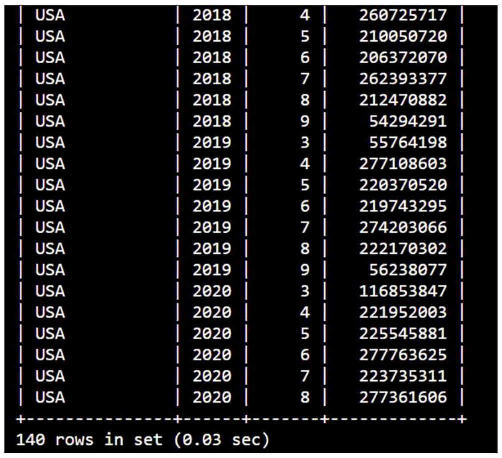
```
mysql> -- 2.d What is the total sales for each region for each month?
mysql> SELECT region, YEAR(week_date) AS year, MONTH(week_date) AS month, SUM(sales) AS total_sales
    -> FROM data_mart.clean_weekly_sales
    -> GROUP BY region, year, month
    -> ORDER BY region, year, month;
+---------------+------+-------+-------------+
| region        | year | month | total_sales |
+---------------+------+-------+-------------+
| AFRICA        | 2018 |     3 |   130542213 |
| AFRICA        | 2018 |     4 |   650194751 |
| AFRICA        | 2018 |     5 |   522814997 |
| AFRICA        | 2018 |     6 |   519127094 |
| AFRICA        | 2018 |     7 |   674135866 |
| AFRICA        | 2018 |     8 |   539077371 |
| AFRICA        | 2018 |     9 |   135084533 |
| AFRICA        | 2019 |     3 |   141619349 |
| AFRICA        | 2019 |     4 |   700447301 |
| AFRICA        | 2019 |     5 |   553828220 |
| AFRICA        | 2019 |     6 |   546092640 |
| AFRICA        | 2019 |     7 |   711867600 |
| AFRICA        | 2019 |     8 |   564497281 |
| AFRICA        | 2019 |     9 |   141236454 |
| AFRICA        | 2020 |     3 |   295605918 |
| AFRICA        | 2020 |     4 |   561141452 |
| AFRICA        | 2020 |     5 |   570601521 |
| AFRICA        | 2020 |     6 |   702340026 |
| AFRICA        | 2020 |     7 |   574216244 |
| AFRICA        | 2020 |     8 |   706022238 |
| ASIA          | 2018 |     3 |   119180883 |
| ASIA          | 2018 |     4 |   603716301 |
| ASIA          | 2018 |     5 |   472634283 |
| ASIA          | 2018 |     6 |   462233474 |
| ASIA          | 2018 |     7 |   602910228 |
| ASIA          | 2018 |     8 |   486137188 |
| ASIA          | 2018 |     9 |   122529255 |
| ASIA          | 2019 |     3 |   129174041 |
| ASIA          | 2019 |     4 |   654973051 |
| ASIA          | 2019 |     5 |   511773780 |
| ASIA          | 2019 |     6 |   498386324 |
| ASIA          | 2019 |     7 |   635366443 |
| ASIA          | 2019 |     8 |   514795070 |
| ASIA          | 2019 |     9 |   130307552 |
| ASIA          | 2020 |     3 |   281415869 |
| ASIA          | 2020 |     4 |   545939355 |
| ASIA          | 2020 |     5 |   541877336 |
| ASIA          | 2020 |     6 |   658863091 |
| ASIA          | 2020 |     7 |   530568085 |
| ASIA          | 2020 |     8 |   662388351 |
| CANADA        | 2018 |     3 |    33815571 |
| CANADA        | 2018 |     4 |   163479820 |
| CANADA        | 2018 |     5 |   130367940 |
| CANADA        | 2018 |     6 |   130410790 |
| CANADA        | 2018 |     7 |   164198426 |
| CANADA        | 2018 |     8 |   133635800 |
| CANADA        | 2018 |     9 |    34042238 |
| CANADA        | 2019 |     3 |    36087248 |
| CANADA        | 2019 |     4 |   179830236 |
| CANADA        | 2019 |     5 |   140979946 |
| CANADA        | 2019 |     6 |   138690815 |
| CANADA        | 2019 |     7 |   173991586 |
```



```
| CANADA        | 2019 |     8 |   139428879 |
| CANADA        | 2019 |     9 |    35025721 |
| CANADA        | 2020 |     3 |    74731510 |
| CANADA        | 2020 |     4 |   141242538 |
| CANADA        | 2020 |     5 |   141030479 |
| CANADA        | 2020 |     6 |   174745093 |
| CANADA        | 2020 |     7 |   138944935 |
| CANADA        | 2020 |     8 |   174008340 |
| EUROPE        | 2018 |     3 |     8402183 |
| EUROPE        | 2018 |     4 |    44549418 |
| EUROPE        | 2018 |     5 |    36492553 |
| EUROPE        | 2018 |     6 |    38998277 |
| EUROPE        | 2018 |     7 |    50535910 |
| EUROPE        | 2018 |     8 |    39104650 |
| EUROPE        | 2018 |     9 |     9777575 |
| EUROPE        | 2019 |     3 |     8989328 |
```

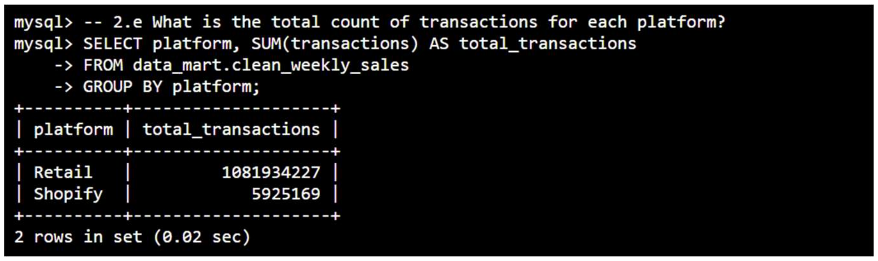


```
| EUROPE        | 2019 |     4 |   46983044 |
| EUROPE        | 2019 |     5 |   36446510 |
| EUROPE        | 2019 |     6 |   36464369 |
| EUROPE        | 2019 |     7 |   47154102 |
| EUROPE        | 2019 |     8 |   36638154 |
| EUROPE        | 2019 |     9 |    9099858 |
| EUROPE        | 2020 |     3 |   17945582 |
| EUROPE        | 2020 |     4 |   35801793 |
| EUROPE        | 2020 |     5 |   36399326 |
| EUROPE        | 2020 |     6 |   47351180 |
| EUROPE        | 2020 |     7 |   39067454 |
| EUROPE        | 2020 |     8 |   46360191 |
| OCEANIA       | 2018 |     3 |  175777460 |
| OCEANIA       | 2018 |     4 |  869324594 |
| OCEANIA       | 2018 |     5 |  692610094 |
| OCEANIA       | 2018 |     6 |  687546255 |
| OCEANIA       | 2018 |     7 |  871333919 |
| OCEANIA       | 2018 |     8 |  714036679 |
| OCEANIA       | 2018 |     9 |  180310608 |
| OCEANIA       | 2019 |     3 |  192331207 |
| OCEANIA       | 2019 |     4 |  953735279 |
| OCEANIA       | 2019 |     5 |  746580473 |
| OCEANIA       | 2019 |     6 |  732354251 |
| OCEANIA       | 2019 |     7 |  934476631 |
| OCEANIA       | 2019 |     8 |  759346286 |
| OCEANIA       | 2019 |     9 |  192154910 |
| OCEANIA       | 2020 |     3 |  415174221 |
| OCEANIA       | 2020 |     4 |  776707747 |
| OCEANIA       | 2020 |     5 |  776466737 |
| OCEANIA       | 2020 |     6 |  951984238 |
| OCEANIA       | 2020 |     7 |  757648850 |
| OCEANIA       | 2020 |     8 |  958930687 |
| SOUTH AMERICA | 2018 |     3 |   16302144 |
| SOUTH AMERICA | 2018 |     4 |   80814046 |
| SOUTH AMERICA | 2018 |     5 |   63685837 |
| SOUTH AMERICA | 2018 |     6 |   63764243 |
| SOUTH AMERICA | 2018 |     7 |   81690746 |
| SOUTH AMERICA | 2018 |     8 |   66079697 |
| SOUTH AMERICA | 2018 |     9 |   16932862 |
| SOUTH AMERICA | 2019 |     3 |   17351683 |
| SOUTH AMERICA | 2019 |     4 |   87069807 |
| SOUTH AMERICA | 2019 |     5 |   67552363 |
| SOUTH AMERICA | 2019 |     6 |   67122227 |
| SOUTH AMERICA | 2019 |     7 |   84577363 |
| SOUTH AMERICA | 2019 |     8 |   68364336 |
| SOUTH AMERICA | 2019 |     9 |   17242721 |
| SOUTH AMERICA | 2020 |     3 |   37369282 |
| SOUTH AMERICA | 2020 |     4 |   70567678 |
| SOUTH AMERICA | 2020 |     5 |   70153609 |
| SOUTH AMERICA | 2020 |     6 |   87360985 |
| SOUTH AMERICA | 2020 |     7 |   69314667 |
| SOUTH AMERICA | 2020 |     8 |   86722019 |
| USA           | 2018 |     3 |   52734998 |
```



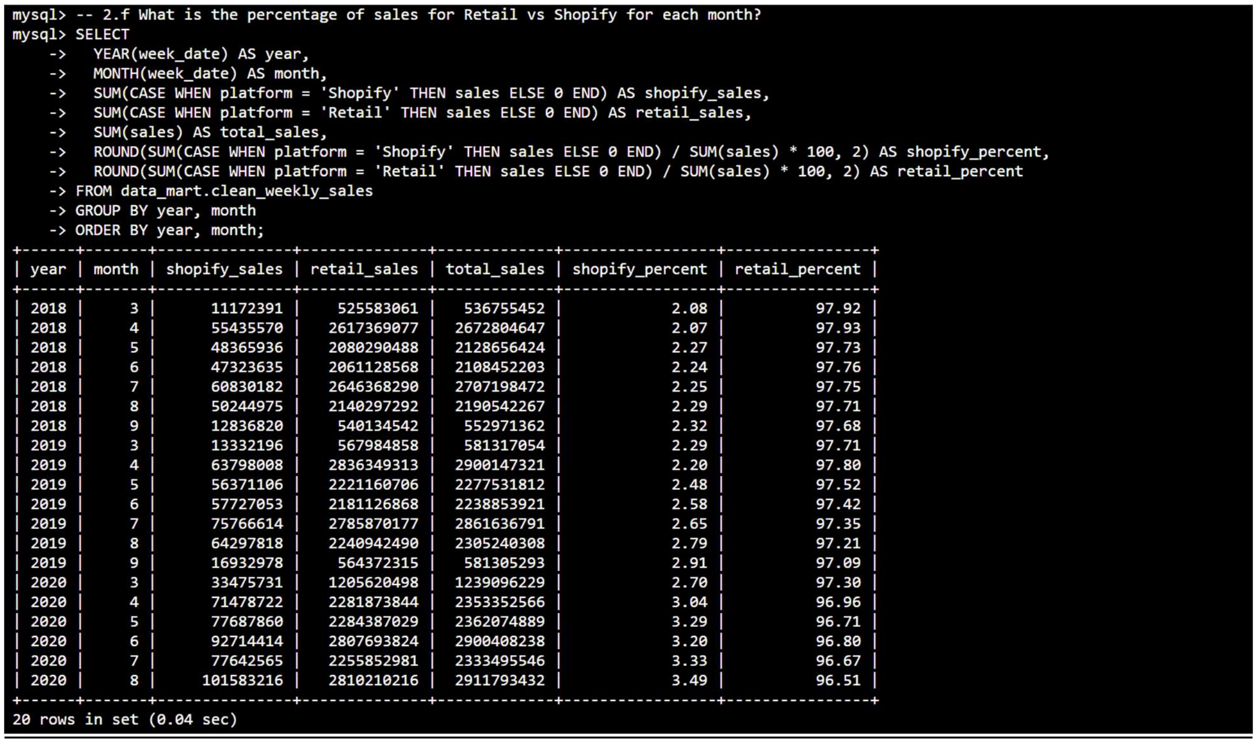
```
| USA           | 2018 |     4 |  260725717 |
| USA           | 2018 |     5 |  210050720 |
| USA           | 2018 |     6 |  206372070 |
| USA           | 2018 |     7 |  262393377 |
| USA           | 2018 |     8 |  212470882 |
| USA           | 2018 |     9 |   54294291 |
| USA           | 2019 |     3 |   55764198 |
| USA           | 2019 |     4 |  277108603 |
| USA           | 2019 |     5 |  220370520 |
| USA           | 2019 |     6 |  219743295 |
| USA           | 2019 |     7 |  274203066 |
| USA           | 2019 |     8 |  222170302 |
| USA           | 2019 |     9 |   56238077 |
| USA           | 2020 |     3 |  116853847 |
| USA           | 2020 |     4 |  221952003 |
| USA           | 2020 |     5 |  225545881 |
| USA           | 2020 |     6 |  277763625 |
| USA           | 2020 |     7 |  223735311 |
| USA           | 2020 |     8 |  277361606 |
+---------------+------+-------+------------+
140 rows in set (0.03 sec)
```

## What is the total count of transactions for each platform?



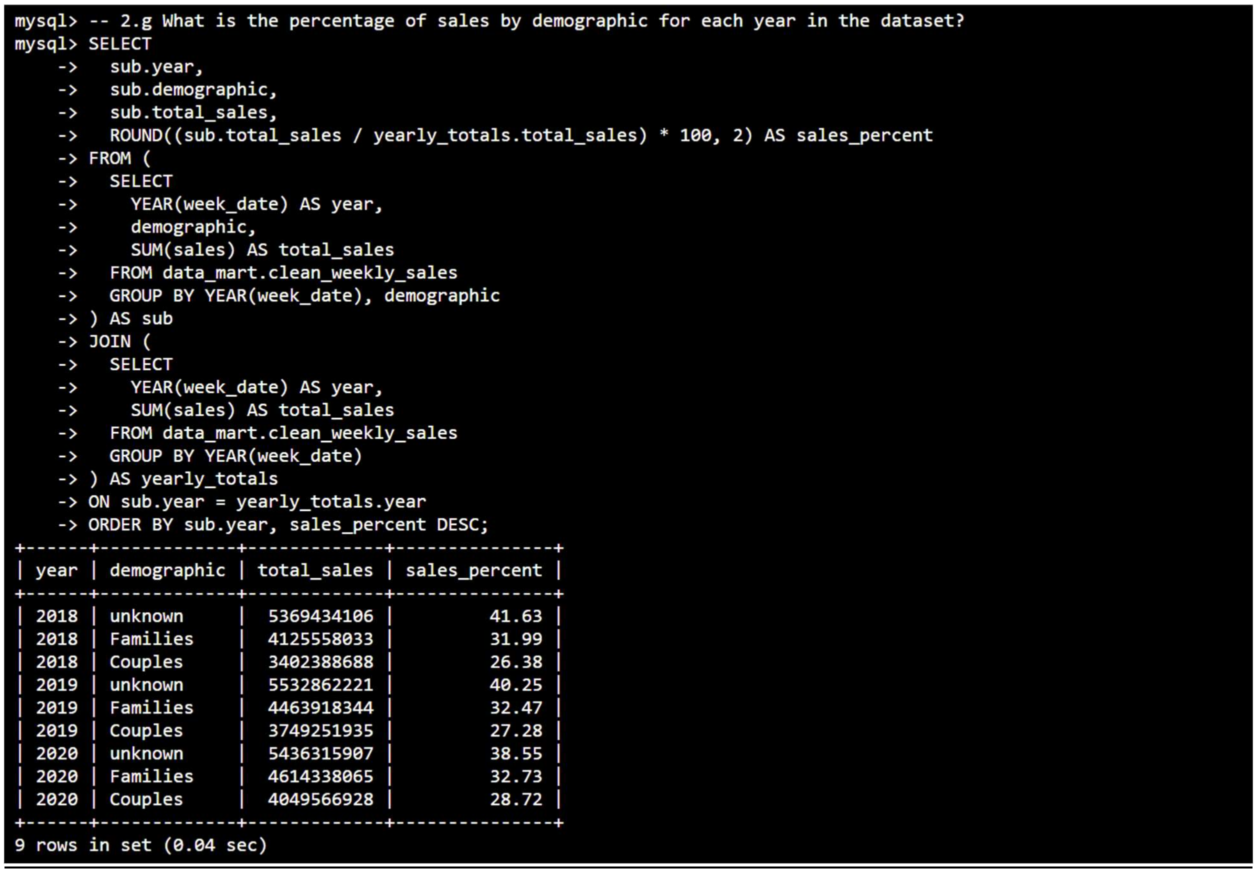
```
mysql> -- 2.e What is the total count of transactions for each platform?
mysql> SELECT platform, SUM(transactions) AS total_transactions
    -> FROM data_mart.clean_weekly_sales
    -> GROUP BY platform;
+----------+--------------------+
| platform | total_transactions |
+----------+--------------------+
| Retail   |         1081934227 |
| Shopify  |            5925169 |
+----------+--------------------+
2 rows in set (0.02 sec)
```

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



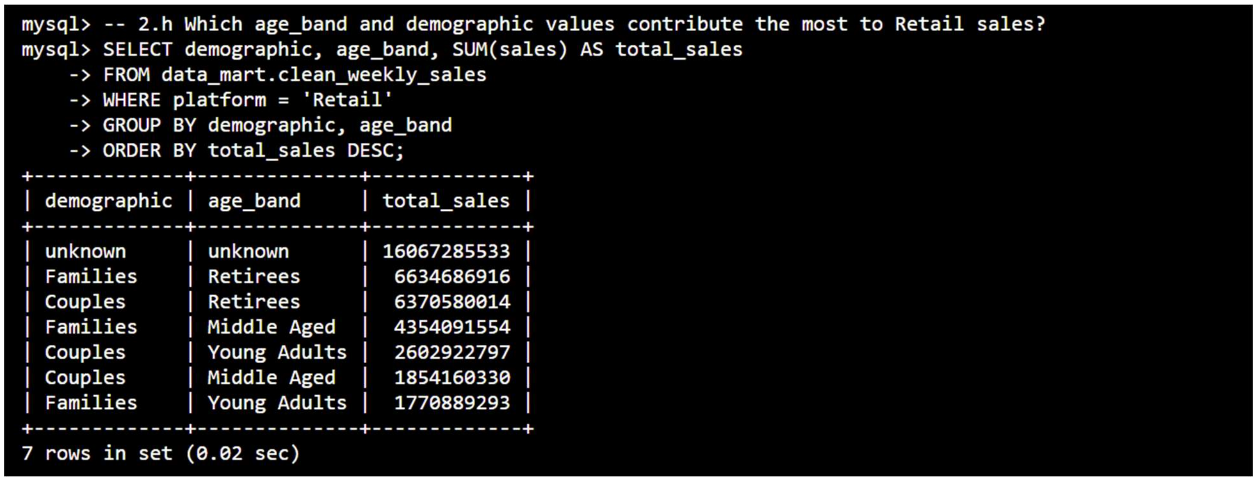
```
mysql> -- 2.f What is the percentage of sales for Retail vs Shopify for each month?
mysql> SELECT
    ->   YEAR(week_date) AS year,
    ->   MONTH(week_date) AS month,
    ->   SUM(CASE WHEN platform = 'Shopify' THEN sales ELSE 0 END) AS shopify_sales,
    ->   SUM(CASE WHEN platform = 'Retail' THEN sales ELSE 0 END) AS retail_sales,
    ->   SUM(sales) AS total_sales,
    ->   ROUND(SUM(CASE WHEN platform = 'Shopify' THEN sales ELSE 0 END) / SUM(sales) * 100, 2) AS shopify_percent,
    ->   ROUND(SUM(CASE WHEN platform = 'Retail' THEN sales ELSE 0 END) / SUM(sales) * 100, 2) AS retail_percent
    -> FROM data_mart.clean_weekly_sales
    -> GROUP BY year, month
    -> ORDER BY year, month;
+------+-------+---------------+--------------+-------------+-----------------+----------------+
| year | month | shopify_sales | retail_sales | total_sales | shopify_percent | retail_percent |
+------+-------+---------------+--------------+-------------+-----------------+----------------+
| 2018 |     3 |      11172391 |    525583061 |   536755452 |            2.08 |          97.92 |
| 2018 |     4 |      55435570 |   2617369077 |  2672804647 |            2.07 |          97.93 |
| 2018 |     5 |      48365936 |   2080290488 |  2128656424 |            2.27 |          97.73 |
| 2018 |     6 |      47323635 |   2061128568 |  2108452203 |            2.24 |          97.76 |
| 2018 |     7 |      60830182 |   2646368290 |  2707198472 |            2.25 |          97.75 |
| 2018 |     8 |      50244975 |   2140297292 |  2190542267 |            2.29 |          97.71 |
| 2018 |     9 |      12836820 |    540134542 |   552971362 |            2.32 |          97.68 |
| 2019 |     3 |      13332196 |    567984858 |   581317054 |            2.29 |          97.71 |
| 2019 |     4 |      63798008 |   2836349313 |  2900147321 |            2.20 |          97.80 |
| 2019 |     5 |      56371106 |   2221160706 |  2277531812 |            2.48 |          97.52 |
| 2019 |     6 |      57727053 |   2181126868 |  2238853921 |            2.58 |          97.42 |
| 2019 |     7 |      75766614 |   2785870177 |  2861636791 |            2.65 |          97.35 |
| 2019 |     8 |      64297818 |   2240942490 |  2305240308 |            2.79 |          97.21 |
| 2019 |     9 |      16932978 |    564372315 |   581305293 |            2.91 |          97.09 |
| 2020 |     3 |      33475731 |   1205620498 |  1239096229 |            2.70 |          97.30 |
| 2020 |     4 |      71478722 |   2281873844 |  2353352566 |            3.04 |          96.96 |
| 2020 |     5 |      77687860 |   2284387029 |  2362074889 |            3.29 |          96.71 |
| 2020 |     6 |      92714414 |   2807693824 |  2900408238 |            3.20 |          96.80 |
| 2020 |     7 |      77642565 |   2255852981 |  2333495546 |            3.33 |          96.67 |
| 2020 |     8 |     101583216 |   2810210216 |  2911793432 |            3.49 |          96.51 |
+------+-------+---------------+--------------+-------------+-----------------+----------------+
20 rows in set (0.04 sec)
```

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



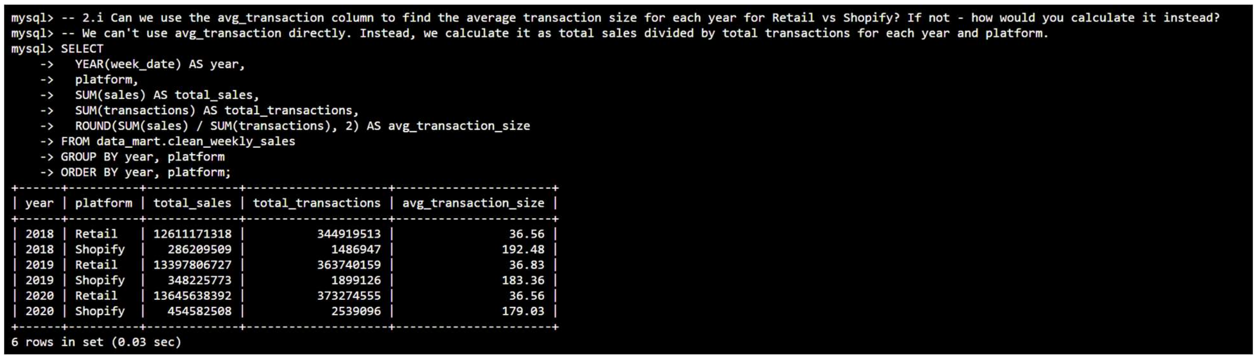
```
mysql> -- 2.g What is the percentage of sales by demographic for each year in the dataset?
mysql> SELECT
    ->   sub.year,
    ->   sub.demographic,
    ->   sub.total_sales,
    ->   ROUND((sub.total_sales / yearly_totals.total_sales) * 100, 2) AS sales_percent
    -> FROM (
    ->   SELECT
    ->     YEAR(week_date) AS year,
    ->     demographic,
    ->     SUM(sales) AS total_sales
    ->   FROM data_mart.clean_weekly_sales
    ->   GROUP BY YEAR(week_date), demographic
    -> ) AS sub
    -> JOIN (
    ->   SELECT
    ->     YEAR(week_date) AS year,
    ->     SUM(sales) AS total_sales
    ->   FROM data_mart.clean_weekly_sales
    ->   GROUP BY YEAR(week_date)
    -> ) AS yearly_totals
    -> ON sub.year = yearly_totals.year
    -> ORDER BY sub.year, sales_percent DESC;
+------+-------------+-------------+---------------+
| year | demographic | total_sales | sales_percent |
+------+-------------+-------------+---------------+
| 2018 | unknown     |  5369434106 |         41.63 |
| 2018 | Families    |  4125558033 |         31.99 |
| 2018 | Couples     |  3402388688 |         26.38 |
| 2019 | unknown     |  5532862221 |         40.25 |
| 2019 | Families    |  4463918344 |         32.47 |
| 2019 | Couples     |  3749251935 |         27.28 |
| 2020 | unknown     |  5436315907 |         38.55 |
| 2020 | Families    |  4614338065 |         32.73 |
| 2020 | Couples     |  4049566928 |         28.72 |
+------+-------------+-------------+---------------+
9 rows in set (0.04 sec)
```

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



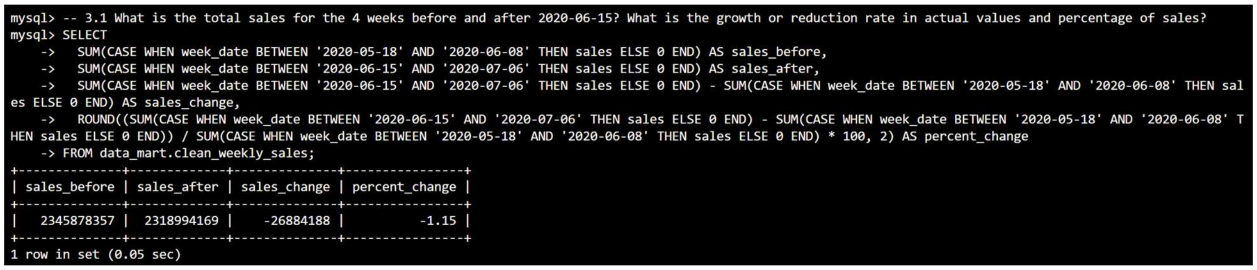
```
mysql> -- 2.h Which age_band and demographic values contribute the most to Retail sales?
mysql> SELECT demographic, age_band, SUM(sales) AS total_sales
    -> FROM data_mart.clean_weekly_sales
    -> WHERE platform = 'Retail'
    -> GROUP BY demographic, age_band
    -> ORDER BY total_sales DESC;
+-------------+--------------+-------------+
| demographic | age_band     | total_sales |
+-------------+--------------+-------------+
| unknown     | unknown      | 16067285533 |
| Families    | Retirees     |  6634686916 |
| Couples     | Retirees     |  6370580014 |
| Families    | Middle Aged  |  4354091554 |
| Couples     | Young Adults |  2602922797 |
| Couples     | Middle Aged  |  1854160330 |
| Families    | Young Adults |  1770889293 |
+-------------+--------------+-------------+
7 rows in set (0.02 sec)
```

## 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?



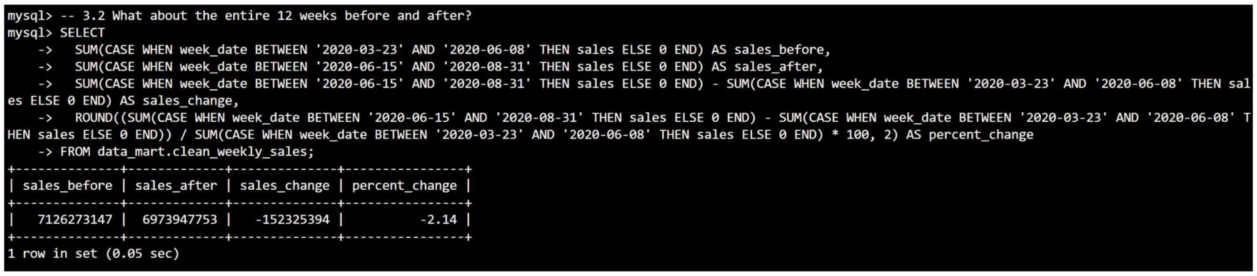
```
mysql> -- 2.i Can we use the avg_transaction column to find the average transaction size for each year for Retail vs Shopify? If not - how would you calculate it instead?
mysql> -- We can't use avg_transaction directly. Instead, we calculate it as total sales divided by total transactions for each year and platform.
mysql> SELECT
    ->   YEAR(week_date) AS year,
    ->   platform,
    ->   SUM(sales) AS total_sales,
    ->   SUM(transactions) AS total_transactions,
    ->   ROUND(SUM(sales) / SUM(transactions), 2) AS avg_transaction_size
    -> FROM data_mart.clean_weekly_sales
    -> GROUP BY year, platform
    -> ORDER BY year, platform;
+------+----------+-------------+--------------------+----------------------+
| year | platform | total_sales | total_transactions | avg_transaction_size |
+------+----------+-------------+--------------------+----------------------+
| 2018 | Retail   | 12611171318 |          344919513 |                36.56 |
| 2018 | Shopify  |   286209509 |            1486947 |               192.48 |
| 2019 | Retail   | 13397806727 |          363740159 |                36.83 |
| 2019 | Shopify  |   348225773 |            1899126 |               183.36 |
| 2020 | Retail   | 13645638392 |          373274555 |                36.56 |
| 2020 | Shopify  |   454582508 |            2539096 |               179.03 |
+------+----------+-------------+--------------------+----------------------+
6 rows in set (0.03 sec)
```

## 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?



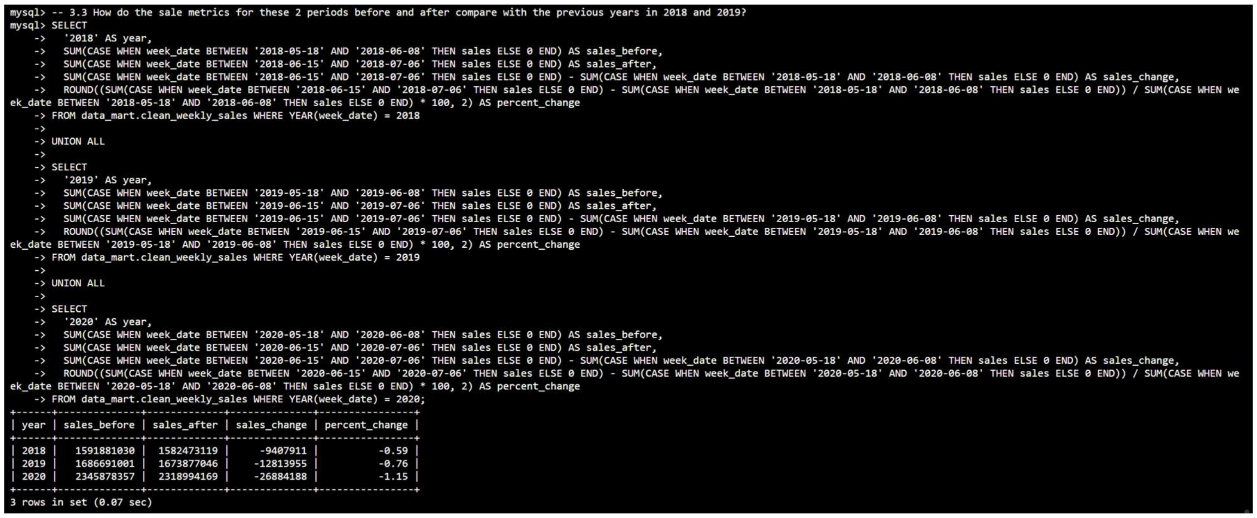
```
mysql> -- 3.1 What is the total sales for the 4 weeks before and after 2020-06-15? What is the growth or reduction rate in actual values and percentage of sales?
mysql> SELECT
    ->   SUM(CASE WHEN week_date BETWEEN '2020-05-18' AND '2020-06-08' THEN sales ELSE 0 END) AS sales_before,
    ->   SUM(CASE WHEN week_date BETWEEN '2020-06-15' AND '2020-07-06' THEN sales ELSE 0 END) AS sales_after,
    ->   SUM(CASE WHEN week_date BETWEEN '2020-06-15' AND '2020-07-06' THEN sales ELSE 0 END) - SUM(CASE WHEN week_date BETWEEN '2020-05-18' AND '2020-06-08' THEN sal
es ELSE 0 END) AS sales_change,
    ->   ROUND((SUM(CASE WHEN week_date BETWEEN '2020-06-15' AND '2020-07-06' THEN sales ELSE 0 END) - SUM(CASE WHEN week_date BETWEEN '2020-05-18' AND '2020-06-08' T
HEN sales ELSE 0 END)) / SUM(CASE WHEN week_date BETWEEN '2020-05-18' AND '2020-06-08' THEN sales ELSE 0 END) * 100, 2) AS percent_change
    -> FROM data_mart.clean_weekly_sales;
+--------------+-------------+--------------+----------------+
| sales_before | sales_after | sales_change | percent_change |
+--------------+-------------+--------------+----------------+
|   2345878357 |  2318994169 |    -26884188 |          -1.15 |
+--------------+-------------+--------------+----------------+
1 row in set (0.05 sec)
```

## What about the entire 12 weeks before and after?



```
mysql> -- 3.2 What about the entire 12 weeks before and after?
mysql> SELECT
    ->   SUM(CASE WHEN week_date BETWEEN '2020-03-23' AND '2020-06-08' THEN sales ELSE 0 END) AS sales_before,
    ->   SUM(CASE WHEN week_date BETWEEN '2020-06-15' AND '2020-08-31' THEN sales ELSE 0 END) AS sales_after,
    ->   SUM(CASE WHEN week_date BETWEEN '2020-06-15' AND '2020-08-31' THEN sales ELSE 0 END) - SUM(CASE WHEN week_date BETWEEN '2020-03-23' AND '2020-06-08' THEN sal
es ELSE 0 END) AS sales_change,
    ->   ROUND((SUM(CASE WHEN week_date BETWEEN '2020-06-15' AND '2020-08-31' THEN sales ELSE 0 END) - SUM(CASE WHEN week_date BETWEEN '2020-03-23' AND '2020-06-08' T
HEN sales ELSE 0 END)) / SUM(CASE WHEN week_date BETWEEN '2020-03-23' AND '2020-06-08' THEN sales ELSE 0 END) * 100, 2) AS percent_change
    -> FROM data_mart.clean_weekly_sales;
+--------------+-------------+--------------+----------------+
| sales_before | sales_after | sales_change | percent_change |
+--------------+-------------+--------------+----------------+
|   7126273147 |  6973947753 |   -152325394 |          -2.14 |
+--------------+-------------+--------------+----------------+
1 row in set (0.05 sec)
```

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



```
mysql> -- 3.3 How do the sale metrics for these 2 periods before and after compare with the previous years in 2018 and 2019?
mysql> SELECT
    ->   '2018' AS year,
    ->   SUM(CASE WHEN week_date BETWEEN '2018-05-18' AND '2018-06-08' THEN sales ELSE 0 END) AS sales_before,
    ->   SUM(CASE WHEN week_date BETWEEN '2018-06-15' AND '2018-07-06' THEN sales ELSE 0 END) AS sales_after,
    ->   SUM(CASE WHEN week_date BETWEEN '2018-06-15' AND '2018-07-06' THEN sales ELSE 0 END) - SUM(CASE WHEN week_date BETWEEN '2018-05-18' AND '2018-06-08' THEN sales ELSE 0 END) AS sales_change,
    ->   ROUND((SUM(CASE WHEN week_date BETWEEN '2018-06-15' AND '2018-07-06' THEN sales ELSE 0 END) - SUM(CASE WHEN week_date BETWEEN '2018-05-18' AND '2018-06-08' THEN sales ELSE 0 END)) / SUM(CASE WHEN we
ek_date BETWEEN '2018-05-18' AND '2018-06-08' THEN sales ELSE 0 END) * 100, 2) AS percent_change
    -> FROM data_mart.clean_weekly_sales WHERE YEAR(week_date) = 2018
    ->
    -> UNION ALL
    ->
    -> SELECT
    ->   '2019' AS year,
    ->   SUM(CASE WHEN week_date BETWEEN '2019-05-18' AND '2019-06-08' THEN sales ELSE 0 END) AS sales_before,
    ->   SUM(CASE WHEN week_date BETWEEN '2019-06-15' AND '2019-07-06' THEN sales ELSE 0 END) AS sales_after,
    ->   SUM(CASE WHEN week_date BETWEEN '2019-06-15' AND '2019-07-06' THEN sales ELSE 0 END) - SUM(CASE WHEN week_date BETWEEN '2019-05-18' AND '2019-06-08' THEN sales ELSE 0 END) AS sales_change,
    ->   ROUND((SUM(CASE WHEN week_date BETWEEN '2019-06-15' AND '2019-07-06' THEN sales ELSE 0 END) - SUM(CASE WHEN week_date BETWEEN '2019-05-18' AND '2019-06-08' THEN sales ELSE 0 END)) / SUM(CASE WHEN we
ek_date BETWEEN '2019-05-18' AND '2019-06-08' THEN sales ELSE 0 END) * 100, 2) AS percent_change
    -> FROM data_mart.clean_weekly_sales WHERE YEAR(week_date) = 2019
    ->
    -> UNION ALL
    ->
    -> SELECT
    ->   '2020' AS year,
    ->   SUM(CASE WHEN week_date BETWEEN '2020-05-18' AND '2020-06-08' THEN sales ELSE 0 END) AS sales_before,
    ->   SUM(CASE WHEN week_date BETWEEN '2020-06-15' AND '2020-07-06' THEN sales ELSE 0 END) AS sales_after,
    ->   SUM(CASE WHEN week_date BETWEEN '2020-06-15' AND '2020-07-06' THEN sales ELSE 0 END) - SUM(CASE WHEN week_date BETWEEN '2020-05-18' AND '2020-06-08' THEN sales ELSE 0 END) AS sales_change,
    ->   ROUND((SUM(CASE WHEN week_date BETWEEN '2020-06-15' AND '2020-07-06' THEN sales ELSE 0 END) - SUM(CASE WHEN week_date BETWEEN '2020-05-18' AND '2020-06-08' THEN sales ELSE 0 END)) / SUM(CASE WHEN we
ek_date BETWEEN '2020-05-18' AND '2020-06-08' THEN sales ELSE 0 END) * 100, 2) AS percent_change
    -> FROM data_mart.clean_weekly_sales WHERE YEAR(week_date) = 2020;
+------+--------------+-------------+--------------+----------------+
| year | sales_before | sales_after | sales_change | percent_change |
+------+--------------+-------------+--------------+----------------+
| 2018 |   1591881030 |  1582473119 |     -9407911 |          -0.59 |
| 2019 |   1686691001 |  1673877046 |    -12813955 |          -0.76 |
| 2020 |   2345878357 |  2318994169 |    -26884188 |          -1.15 |
+------+--------------+-------------+--------------+----------------+
3 rows in set (0.07 sec)
```