SQL Project : USING SQL TO UNCOVER TRUTH ¶

The following shall be learned from this project:

- Use the scientific method and critical thinking to glean insights about your data
- Solve real-world problems outside of those described within this book by using the skills that you have acquired
- Convert data and hypotheses into actionable tasks and insights
- Use the skills developed in this book to solve problems in your specific problem domain

This case study will not only demonstrate the processes used in SQL analysis to find solutions for actual problems but will also provide you with confidence and experience in solving such problems. Sales Data: understanding the cause of an unexpected drop in sales

```
sqlda=# SELECT model, base_msrp, production_start_date FROM products WHERE
sqlda-# product_type='scooter';
        model         | base_msrp | production_start_date
-----------------------+-----------+-----------------------
 Lemon                 |    399.99 | 2010-03-03 00:00:00
 Lemon Limited Edition |    799.99 | 2011-01-03 00:00:00
 Lemon                 |    499.99 | 2013-05-01 00:00:00
 Blade                 |    699.99 | 2014-06-23 00:00:00
 Bat                   |    599.99 | 2016-10-10 00:00:00
 Bat Limited Edition   |    699.99 | 2017-02-15 00:00:00
 Lemon Zester          |    349.99 | 2019-02-04 00:00:00
```

- Looking at the results from the search, we can see that we have two scooter products with Bat in the name; Bat and Bat Limited Edition. The Bat Scooter, which started production on October 10, 2016, with a suggested retail price of $599.99; andtheBatLimitedEditionScooter, whichstartedproductionapproximately4monthslater, onFebruary15, 2017, atapriceof$699.99. Looking at the product information supplied, we can see that the Bat Scooter is somewhat unique from a price perspective, being the only scooter with a suggested retail price of $599.99.Therearetwoothersat$699.99 and one at $ 499.99.
- Similarly, if we consider the production start date in isolation, the original Bat Scooter is again unique in that it is the only scooter starting production in the last quarter or even half of the year (date format: YYYY-MM-DD). All other scooters start production in the first half of the year, with only the Blade scooter starting production in June.
- In order to use the sales information in conjunction with the product information available, we also need to get the product ID for each of the scooters.

Extract the model name and product IDs for the scooters available within the database. We will need this information to reconcile the product information with the available sales information:

```
sqlda=# SELECT model, product_id
FROM products
WHERE product_type='scooter';

        model         | product_id
-----------------------+------------
 Lemon                 |          1
 Lemon Limited Edition |          2
 Lemon                 |          3
 Blade                 |          5
 Bat                   |          7
 Bat Limited Edition   |          8
 Lemon Zester          |         12
(7 rows)


Insert the results of this query into a new table called product_names:
```

```
 SELECT model, product_id INTO product_names  FROM products WHERE product_type='scooter';

  model         | product_id
-----------------------+------------
 Lemon                 |          1
 Lemon Limited Edition |          2
 Lemon                 |          3
 Blade                 |          5
 Bat                   |          7
 Bat Limited Edition   |          8
 Lemon Zester          |         12
(7 rows)
```

- As described in the output, we can see that the Bat Scooter lies between the price points of some of the other scooters and that it was also manufactured a lot later in the year compared to the others.
- By completing this very preliminary data collection step, we have the information required to collect sales data on the Bat Scooter as well as other scooter products for comparison. While this exercise involved using the simplest SQL commands, it has already yielded some useful information.

Extracting the Sales Information

- Use an inner join on the product_id columns of both the product_names table and the sales table. From the result of the inner join, select the model, customer_id, sales_transaction_date, sales_amount, channel, and dealership_id, and store the values in a separate table called product_sales:

```
SELECT model, customer_id, sales_transaction_date, sales_amount, channel, dealership_id INTO products_sales
FROM sales INNER JOIN product_ names ON sales.product_id=product_names.product_id;

model | customer_id | sales_transaction_date |    sales_amount    | channel  | dealership_id
-------+-------------+------------------------+--------------------+----------+---------------
 Lemon |        2067 | 2010-07-30 20:27:15    | 359.99100000000004 | internet |
 Lemon |       10948 | 2010-08-09 22:19:22    | 359.99100000000004 | internet |
 Lemon |       14578 | 2010-06-21 04:19:10    |             399.99 | internet |
 Lemon |        2687 | 2010-07-25 04:44:25    |             399.99 | internet |
 Lemon |       17256 | 2011-04-08 08:55:11    |             399.99 | internet |
 Lemon |       42685 | 2012-02-10 20:48:55    | 359.99100000000004 | internet |
 Lemon |       16115 | 2010-09-06 02:27:42    | 359.99100000000004 | internet |
 Lemon |        4561 | 2010-04-26 23:50:40    |             399.99 | internet |
 Lemon |       36904 | 2010-09-21 07:29:42    | 359.99100000000004 | internet |
 Lemon |       44087 | 2010-08-09 05:12:41    |             399.99 | internet |
```

Select all the information from the product_sales table that is available for the Bat Scooter and order the sales information by sales_transaction_date in ascending order. By selecting the data in this way, we can look at the first few days of the sales records in detail: sqlda=# SELECT * FROM products_sales WHERE model='Bat' ORDER BY sales_ transaction_date;

```
model | customer_id | sales_transaction_date | sales_amount |  channel   | dealership_id
-------+-------------+------------------------+--------------+------------+---------------
 Bat   |        4319 | 2016-10-10 00:41:57    |       599.99 | internet   |
 Bat   |       40250 | 2016-10-10 02:47:28    |       599.99 | dealership |             4
 Bat   |       35497 | 2016-10-10 04:21:08    |       599.99 | dealership |             2
 Bat   |        4553 | 2016-10-10 07:42:59    |       599.99 | dealership |            11
 Bat   |       11678 | 2016-10-10 09:21:08    |       599.99 | internet   |
 Bat   |       45868 | 2016-10-10 10:29:29    |       599.99 | internet   |
 Bat   |       24125 | 2016-10-10 18:57:25    |       599.99 | dealership |             1
 Bat   |       31307 | 2016-10-10 21:22:38    |       599.99 | internet   |
 Bat   |       42213 | 2016-10-10 21:27:36    |       599.99 | internet   |
 Bat   |       47790 | 2016-10-11 01:28:58    |       599.99 | dealership |            20
 Bat   |        6342 | 2016-10-11 03:04:57    |       599.99 | internet   |
 Bat   |       45880 | 2016-10-11 04:09:19    |       599.99 | dealership |             7
 Bat   |       43477 | 2016-10-11 05:24:50    |       599.99 | internet   |
 Bat   |        6322 | 2016-10-11 08:48:07    |       599.99 | internet   |
 Bat   |       46653 | 2016-10-11 15:47:01    |       599.99 | dealership |             6
 Bat   |        9045 | 2016-10-12 00:15:20    |       599.99 | dealership |            19
 Bat   |       23679 | 2016-10-12 00:17:53    |      539.991 | internet   |
 Bat   |       49856 | 2016-10-12 00:26:15    |       599.99 | dealership |            10
 Bat   |       45256 | 2016-10-12 02:08:01    |      539.991 | dealership |             7
 Bat   |       48809 | 2016-10-12 05:08:43    |       599.99 | internet   |
 Bat   |       42625 | 2016-10-12 06:17:55    |       599.99 | internet   |
 Bat   |       39653 | 2016-10-12 06:28:25    |       599.99 | dealership |             7
 Bat   |       49226 | 2016-10-12 10:26:13    |      539.991 | internet   |
 Bat   |       18602 | 2016-10-12 15:09:53    |       599.99 | internet   |
 Bat   |       43013 | 2016-10-12 18:45:07    |       599.99 | dealership |            16
 Bat   |       14298 | 2016-10-13 01:43:24    |       599.99 | internet   |
 Bat   |       13470 | 2016-10-13 02:58:32    |       599.99 | dealership |            15
 Bat   |       17049 | 2016-10-13 04:57:15    |       599.99 | dealership |             5
 Bat   |       35191 | 2016-10-13 06:24:16    |      539.991 | dealership |             7
 Bat   |       22577 | 2016-10-13 08:30:32    |      479.992 | internet   |
 Bat   |        2552 | 2016-10-13 08:50:01    |      539.991 | dealership |            19
 Bat   |       24564 | 2016-10-13 09:14:55    |       599.99 | dealership |             2
 Bat   |         369 | 2016-10-13 14:47:51    |       599.99 | dealership |             1
 Bat   |       21305 | 2016-10-13 16:52:19    |       599.99 | dealership |            19
 Bat   |       11343 | 2016-10-13 23:25:58    |       599.99 | internet   |
 Bat   |       42792 | 2016-10-14 00:38:23    |       599.99 | dealership |            11
```

Count the number of records available by using the following query:

- sqlda=# SELECT COUNT(model) FROM products_sales WHERE model='Bat';

The model count for the 'Bat' model is as shown here:

count

7328 (1 row)

So, we have 7328 sales, beginning October 10, 2016. Check the date of the final sales record by performing the next step.

Determine the last sale date for the Bat Scooter by selecting the maximum (using the MAX function) for sales_transaction_date:

- sqlda=# SELECT MAX(sales_transaction_date) FROM products_sales WHERE model='Bat'; The last sale date is shown here:

  max

2019-05-31 22:15:30 (1 row)

The last sale in the database occurred on May 31, 2019.

Collect the daily sales volume for the Bat Scooter and place it in a new table called bat_sales to confirm the information provided by the sales team stating that sales dropped by 20% after the first 2 weeks:

- sqlda=# SELECT* INTO bat_sales FROM products_sales WHERE model='Bat' ORDER BY sales_transaction_date;

Remove the time information to allow tracking of sales by date, since, at this stage, we are not interested in the time at which each sale occurred. To do so, run the following query:

- sqlda=# UPDATE bat_sales SET sales_transaction_date=DATE(sales_transaction_date);

```
#### Display the first five records of bat_sales ordered by sales_transaction_date:
sqlda=# SELECT * FROM bat_sales ORDER BY sales_transaction_date LIMIT 5;
```

```
model | customer_id | sales_transaction_date | sales_amount |  channel   | dealership_id
-------+-------------+------------------------+--------------+------------+---------------
 Bat   |        4553 | 2016-10-10 00:00:00    |       599.99 | dealership |            11
 Bat   |       35497 | 2016-10-10 00:00:00    |       599.99 | dealership |             2
 Bat   |       40250 | 2016-10-10 00:00:00    |       599.99 | dealership |             4
 Bat   |        4319 | 2016-10-10 00:00:00    |       599.99 | internet   |
 Bat   |       11678 | 2016-10-10 00:00:00    |       599.99 | internet   |
(5 rows)
```

Create a new table (bat_sales_daily) containing the sales transaction dates and a daily count of total sales:

- sqlda=# SELECT sales_transaction_date, COUNT(sales_transaction_date) INTO bat_sales_daily FROM bat_sales GROUP BY sales_transaction_date ORDER BY sales_transaction_date;

Examine the first 22 records (a little over 3 weeks), as sales were reported to have dropped after approximately the first 2 weeks:

- sqlda=# SELECT * FROM bat_sales_daily LIMIT 22;

```
sales_transaction_date | count
------------------------+-------
 2016-10-10 00:00:00    |     9
 2016-10-11 00:00:00    |     6
 2016-10-12 00:00:00    |    10
 2016-10-13 00:00:00    |    10
 2016-10-14 00:00:00    |     5
 2016-10-15 00:00:00    |    10
 2016-10-16 00:00:00    |    14
 2016-10-17 00:00:00    |     9
 2016-10-18 00:00:00    |    11
 2016-10-19 00:00:00    |    12
 2016-10-20 00:00:00    |    10
 2016-10-21 00:00:00    |     6
 2016-10-22 00:00:00    |     2
 2016-10-23 00:00:00    |     5
 2016-10-24 00:00:00    |     6
 2016-10-25 00:00:00    |     9
 2016-10-26 00:00:00    |     2
 2016-10-27 00:00:00    |     4
 2016-10-28 00:00:00    |     7
 2016-10-29 00:00:00    |     5
 2016-10-30 00:00:00    |     5
 2016-10-31 00:00:00    |     3
(22 rows)
```

- We can see a drop-in sales after October 20, as there are 7 days in the first 11 rows that record double-digit sales, and none over the next 11 days.
- At this stage, we can confirm that there has been a drop off in sales, although we are yet to quantify precisely the extent of the reduction or the reason for the drop off in sales.

```
## Quantifying the Sales Drop
* we will use our knowledge of the windowing methods. We identified the occurrence of the sales drop as being approximately 10
days after launch. Here, we will try to quantify the drop off in sales for the Bat Scooter
```

Compute the daily cumulative sum of sales using the OVER and ORDER BY statements. Insert the results into a new table called bat_sales_growth:

- sqlda=# SELECT *, sum(count) OVER (ORDER BY sales_transaction_date) INTO bat_sales_growth FROM bat_sales_daily;
- The following table shows the daily cumulative sum of sales:

```
 sales_transaction_date | count | sum
------------------------+-------+-----
 2016-10-10 00:00:00    |     9 |   9
 2016-10-11 00:00:00    |     6 |  15
 2016-10-12 00:00:00    |    10 |  25
 2016-10-13 00:00:00    |    10 |  35
 2016-10-14 00:00:00    |     5 |  40
 2016-10-15 00:00:00    |    10 |  50
 2016-10-16 00:00:00    |    14 |  64
 2016-10-17 00:00:00    |     9 |  73
 2016-10-18 00:00:00    |    11 |  84
 2016-10-19 00:00:00    |    12 |  96
(10 rows)
```

Compute a 7-day lag function of the sum column and insert all the columns of bat_sales_daily and the new lag column into a new table, bat_sales_daily_delay.

- This lag column indicates what the sales were like 1 week before the given record:
- sqlda=# SELECT *, lag(sum, 7) OVER (ORDER BY sales_transaction_date) INTO bat_sales_daily_delay FROM bat_sales_growth

```
 sales_transaction_date | count | sum | lag
------------------------+-------+-----+-----
 2016-10-10 00:00:00    |     9 |   9 |
 2016-10-11 00:00:00    |     6 |  15 |
 2016-10-12 00:00:00    |    10 |  25 |
 2016-10-13 00:00:00    |    10 |  35 |
 2016-10-14 00:00:00    |     5 |  40 |
 2016-10-15 00:00:00    |    10 |  50 |
 2016-10-16 00:00:00    |    14 |  64 |
 2016-10-17 00:00:00    |     9 |  73 |   9
 2016-10-18 00:00:00    |    11 |  84 |  15
 2016-10-19 00:00:00    |    12 |  96 |  25
 2016-10-20 00:00:00    |    10 | 106 |  35
 2016-10-21 00:00:00    |     6 | 112 |  40
 2016-10-22 00:00:00    |     2 | 114 |  50
 2016-10-23 00:00:00    |     5 | 119 |  64
 2016-10-24 00:00:00    |     6 | 125 |  73
(15 rows)
```

#### Compute the sales growth as a percentage, comparing the current sales volume to that of 1 week prior. Insert the resulting table into a new table called bat_sales_delay_vol:

```
* sqlda=# SELECT*, (sum-lag)/lag AS volume INTO bat_sales_delay_vol FROM bat_sales_daily_delay ;
* Compare the first 22 values of the bat_sales_delay_vol table:
* sqlda=# SELECT* FROM bat_sales_daily_delay_vol LIMIT 22;
```

In [ ]:

In [ ]:

In [ ]: