SQL Project

Intermediate & Advanced Assigment

Aslamic Adika

Full Stack Data Analytics (FSDA)
RevoU

Table of Contents

Dataset Overview

TheLook is a fictitious eCommerce clothing site developed by the Looker team. The dataset contains information about customers, product, orders, logistics, web events and digital marketing campaigns. The contents of this dataset are synthetic, and are provided to industry practitioners for the purpose of product discovery, testing, and evaluation.

Query 1

Total users who completed the order and total orders per month between January 2019–April 2022.

```
1
2 SELECT
3
4   format_date('%b %Y', created_at) as Order_Month,
5   count(distinct user_id) as Total_User,
6   count(distinct order_id) as Total_Order
7
8 FROM `bigquery-public-data.thelook_ecommerce.orders`
9 WHERE status = 'Complete' AND created_at BETWEEN '2019-01-01' and '2022-04-30'
10 GROUP BY 1
11 ORDER BY 3 DESC
```

query link: **query1**

Query1 Result

Query results samples: top five total orders per month.

Order Month	Total User	Total Order
Sep 2022	3385	3606
Aug 2022	2348	2416
Jul 2022	1990	2034
Jun 2022	1746	1765
May 2022	1689	1706

Query 2

Average Order Value (AOV) per month between January 2019–April 2022.

$$AOV = \frac{\text{Revenue}}{\text{Total Order}} \quad (1)$$

```
1
2  SELECT
3
4    date(date_trunc(t1.created_at, month)) as Order_Month,
5    round(sum(t1.sale_price) /count(distinct t2.order_id),3) as AOV,
6    count(distinct t1.user_id) as Total_User
7
8  FROM `bigquery-public-data.thelook_ecommerce.order_items` as t1
9    LEFT JOIN `bigquery-public-data.thelook_ecommerce.orders` as t2
10   ON t1.user_id = t2.user_id
11
12 WHERE t1.status = 'Complete'
13   and date(date_trunc(t1.created_at, month)) between '2019-01-01' and '2022-04-30'
14
15 GROUP BY 1
16 ORDER BY 1
```

query link: **query2**

Query2 Result

Query results samples: AOV in the first semester of 2019.

Order Month	AOV	Total User
2019-01-01	47.25	2
2019-02-01	105.59	21
2019-03-01	87.927	37
2019-04-01	103.016	61
2019-05-01	78.25	84
2019-06-01	96.708	91

Query 3 I

The first and last name of users, from the youngest and oldest age of each gender.

```
1
2  with umur AS
3  (
4    SELECT
5
6      gender,
7      first_name,
8      last_name,
9      date(created_at) as dibuat,
10     age,
11     CASE
12       when age = 12 then 'Youngest'
13       when age = 70 then 'Oldest'
14       end as Status
15
16   FROM `bigquery-public-data.thelook_ecommerce.users`
17   WHERE age = 12 or age = 70
18 )
19 SELECT
20
21   gender,
```

Query 3 II

```
22    first_name,
23    last_name,
24    age,
25    Status
26
27 FROM umur
28 WHERE date(umur.dibuat) between '2019-01-01' and '2022-04-30'
```

query link: **query3**

Query3 Result

Query result sample:

gender	first name	last name	age	Status
M	John	Collins	12	Youngest
M	Daniel	Smith	12	Youngest
M	Rodney	Roberts	70	Oldest
M	Justin	Soto	12	Youngest
M	Philip	Williams	12	Youngest
F	Cathy	Ware	12	Youngest
M	Zachary	Campbell	12	Youngest
F	Maureen	Roman	12	Youngest
M	Christopher	Landry	12	Youngest
M	Chad	Watson	70	Oldest

Query 4 I

Top five most profitable product and its details .

```
1
2  WITH order_komplit_dikirim as
3  (
4      SELECT
5        t1.created_at as Order_Month,
6        t1.product_id as Product_ID,
7        t2.name as Product_Name,
8        t2.cost as Cost,
9        t2.retail_price as Price
10     FROM `bigquery-public-data.thelook_ecommerce.order_items` as t1
11     LEFT JOIN `bigquery-public-data.thelook_ecommerce.products` as t2
12     ON t1.product_id = t2.id
13     WHERE t1.status = 'Complete'
14     ORDER BY 1
15   ),
16   Price_Cost as
17   (
18     SELECT
19       date_trunc(date(order_komplit_dikirim.Order_Month), month) as Order_time,
20       order_komplit_dikirim.Product_ID as Product_ID,
21       order_komplit_dikirim.Product_Name as Product_Name,
22       sum(order_komplit_dikirim.Cost) over(partition by order_komplit_dikirim.Product_ID
           order by date_trunc(date(order_komplit_dikirim.Order_Month), month)) as Cost,
```

Query 4 II

```
23        sum(order_komplit_dikirim.Price) over(partition by
              order_komplit_dikirim.Product_ID order by
              date_trunc(date(order_komplit_dikirim.Order_Month), month)) as Price
24      FROM order_komplit_dikirim
25      ORDER BY 1,4
26    ),
27    rangking_product_perbulan as
28    (
29    SELECT
30      Price_Cost.Order_time as Order_Month,
31      Price_Cost.Product_ID,
32      Price_Cost.Product_Name,
33      Round(sum(Price_Cost.Price),3) as Sales,
34      Round(sum(Price_Cost.Cost),3) as Cost,
35      Round(sum(Price_Cost.Price) - sum(Price_Cost.Cost),3) as Profit,
36      RANK() OVER(PARTITION BY Price_Cost.Order_time ORDER BY sum(Price_Cost.Price) -
            sum(Price_Cost.Cost)  DESC) as Rank
37    FROM Price_Cost
38    GROUP BY 1,2,3
39    )
40
41  SELECT *
42  FROM rangking_product_perbulan
43  WHERE Rank IN (1,2,3,4,5)
44  ORDER BY 1,7
```

Query4 Result

query link: **query4**

Query results sample:

Order Month	Product ID	Sales	Cost	Profit	Rank
2019-01-01	28556	129.99	48.096	81.894	1
2019-01-01	15551	91.16	47.494	43.666	2
2019-01-01	18928	79.5	36.967	42.533	3
2019-01-01	11290	78.0	39.468	38.532	4
2019-01-01	26675	38.5	14.168	24.332	5
2019-02-01	2469	308.0	144.452	163.548	1
2019-02-01	23718	129.99	56.156	73.834	2
2019-02-01	24344	129.99	63.175	66.815	3
2019-02-01	1278	99.5	47.163	52.337	4
2019-02-01	8846	89.99	42.925	47.065	5

Query 5 I

Month to Date of total revenue in each product categories of past three months (current date 15 April 2022) breakdown by date.

```
1
2  WITH jan as (
3    SELECT
4      DATE_TRUNC(date(t2.created_at), day) as Date,
5      t1.category as Category,
6      round(sum(t1.retail_price),3) as Revenue
7    FROM `bigquery-public-data.thelook_ecommerce.products`as t1
8    LEFT JOIN `bigquery-public-data.thelook_ecommerce.order_items`as t2
9    ON t1.id = t2.order_id
10   WHERE t2.status = 'Complete'
11   and date(t2.created_at) between '2022-01-01' and '2022-01-15'
12   GROUP BY 1,2
13   ORDER BY 1,3 DESC
14   ),
15   jan_feb as (
16   select *
17   FROM jan
18   union all
19   SELECT
20     DATE_TRUNC(date(t2.created_at), day) as Date,
21     t1.category as Category,
```

Query 5 II

```
22      round(sum(t1.retail_price),3) as Revenue
23    FROM `bigquery-public-data.thelook_ecommerce.products` as t1
24    LEFT JOIN `bigquery-public-data.thelook_ecommerce.order_items` as t2
25    ON t1.id = t2.order_id
26    WHERE t2.status = 'Complete'
27    and date(t2.created_at) between '2022-02-01' and '2022-02-15'
28    GROUP BY 1,2
29    ORDER BY 1,3 DESC
30    ),
31    jan_feb_mar as (
32    SELECT *
33    FROM jan_feb
34    union all
35    SELECT
36      DATE_TRUNC(date(t2.created_at), day) as Date,
37      t1.category as Category,
38      round(sum(t1.retail_price),3) as Revenue
39    FROM `bigquery-public-data.thelook_ecommerce.products` as t1
40    LEFT JOIN `bigquery-public-data.thelook_ecommerce.order_items` as t2
41    ON t1.id = t2.order_id
42    WHERE t2.status = 'Complete'
43    and date(t2.created_at) between '2022-03-01' and '2022-03-15'
44    GROUP BY 1,2
45    ORDER BY 1,3 DESC
46    ),
47    jan_feb_mar_apr
48    as (
```

Query 5 III

```
49      SELECT *
50    FROM jan_feb_mar
51    union all
52    SELECT
53      DATE_TRUNC(date(t2.created_at), day) as Date,
54      t1.category as Category,
55      round(sum(t1.retail_price),3) as Revenue
56    FROM `bigquery-public-data.thelook_ecommerce.products` as t1
57    LEFT JOIN `bigquery-public-data.thelook_ecommerce.order_items` as t2
58    ON t1.id = t2.order_id
59    WHERE t2.status = 'Complete'
60    and date(t2.created_at) between '2022-04-01' and '2022-04-15'
61    GROUP BY 1,2
62    ORDER BY 1,3 DESC
63  )
64  SELECT *
65  FROM jan_feb_mar_apr
```

query link: **query5**

Query5 Result

Query results samples:

Date	Category	Revenue
2022-01-01	Outerwear & Coats	348.95
2022-01-01	Fashion Hoodies & Sweatshirts	119.98
2022-01-01	Sleep & Lounge	114.97
2022-01-01	Tops & Tees	99.99
2022-01-01	Intimates	90.29
2022-01-01	Shorts	29.99

Query 6 I

Monthly growth of order and revenue for each category product.

```
1
2  WITH Total_Order_n_Revenue as
3  (
4    SELECT
5        date_trunc(date(t2.created_at), month) as Date,
6        t1.category as Product_Category,
7        count(distinct t2.order_id) as Total_Order,
8        (sum(t1.retail_price)-sum(t1.cost)) as Revenue
9    FROM `bigquery-public-data.thelook_ecommerce.products` as t1
10   LEFT JOIN `bigquery-public-data.thelook_ecommerce.order_items` as t2
11   ON t1.id = t2.order_id
12   WHERE date(t2.created_at) between '2019-01-01' and '2022-04-30'
13   group by 1,2
14   order by 1 DESC,  4 DESC
15 ),
16 dapur_growth as
17 (
18   SELECT
19     Total_Order_n_Revenue.Date,
20     Total_Order_n_Revenue.Product_Category,
```

Query 6 II

```
21      (Total_Order_n_Revenue.Total_Order - lag(Total_Order_n_Revenue.Total_Order)
            OVER(PARTITION BY Total_Order_n_Revenue.Product_Category ORDER BY
            Total_Order_n_Revenue.Date ))/lag(Total_Order_n_Revenue.Total_Order)
            OVER(PARTITION BY Total_Order_n_Revenue.Product_Category ORDER BY
            Total_Order_n_Revenue.Date )*100 as Growth_Order,
22      ((Total_Order_n_Revenue.Revenue -  lag(Total_Order_n_Revenue.Revenue) OVER(PARTITION
            BY Total_Order_n_Revenue.Product_Category ORDER BY Total_Order_n_Revenue.Date )
             )*100)/(lag(Total_Order_n_Revenue.Revenue) OVER(PARTITION BY
            Total_Order_n_Revenue.Product_Category ORDER BY Total_Order_n_Revenue.Date ) )
            as Growth_Revenue
23    FROM Total_Order_n_Revenue
24    ORDER BY Total_Order_n_Revenue.Date
25  )
26
27  SELECT
28    t1.Date,
29    t1.Product_Category,
30    concat(round(t1.Growth_Order,2),'%') as Growth_Order,
31    concat(round(t1.Growth_Revenue,2),'%') as Growth_Revenue
32  FROM dapur_growth as t1
33  ORDER BY 1 DESC,2
```

Query6 Result

query link: **query6**

Query results samples:

Date	Product Category	Growth Order	Growth Revenue
2022-04-01	Accessories	15.28%	37.79%
2022-04-01	Active	-8.7%	18.46%
2022-04-01	Blazers & Jackets	9.09%	-38.52%
2022-04-01	Clothing Sets	-25%	-37.91%
2022-04-01	Dresses	-27.66%	-46.79%
2022-04-01	Fashion Hoodies & Sweatshirts	-17.39%	-22.47%
2022-04-01	Intimates	32.95%	42.04%
2022-04-01	Jeans	13.19%	18.45%
2022-04-01	Jumpsuits & Rompers	160%	386.8%
2022-04-01	Leggings	33.33%	24.5%

Cohort I

Top five most profitable product and its details .

```sql
WITH cohort_item as
(
  SELECT
    user_id as Users,
    MIN(DATE_TRUNC(DATE(created_at), MONTH)) as cohort_month
  FROM `bigquery-public-data.thelook_ecommerce.orders`
  GROUP BY 1
),
activity as
(
  SELECT

      t1.user_id as user_activ,
      DATE_DIFF(
      date(date_trunc(t1.created_at, MONTH)), t2.cohort_month,
    MONTH) as Periode
    FROM `bigquery-public-data.thelook_ecommerce.orders` as t1
    LEFT JOIN cohort_item as t2
    on t1.user_id = t2.Users

  WHERE EXTRACT(YEAR FROM DATE(t2.cohort_month)) IN (2019,2020,2021,2022)
  GROUP BY 1,2
),
```

Cohort II

```
24 c_size as
25 (
26   SELECT
27     cohort_item.cohort_month,
28     count( cohort_item.Users) as Cohort_Size
29   FROM cohort_item
30   GROUP BY 1
31 ),
32 retention as
33 (
34   SELECT
35
36     t2.cohort_month as CM,
37     t1.Periode,
38     count(t1.user_activ) as Total_User_Act
39
40   FROM activity as t1
41   LEFT JOIN cohort_item as t2
42   ON t1.user_activ = t2.Users
43   GROUP BY 1,2
44   ORDER BY 1,2
45 )
46 SELECT
47
48   t1.cohort_month,
49   t1.Cohort_Size,
50   t2.Periode,
```

Cohort III

```
51    t2.Total_User_Act,
52    (t2.Total_User_Act/t1.Cohort_Size) as Percent_Cohort
53
54 FROM c_size as t1
55 LEFT JOIN retention as t2
56 on t1.cohort_month = t2.CM
57 ORDER BY 1,3
```

query link: query cohort

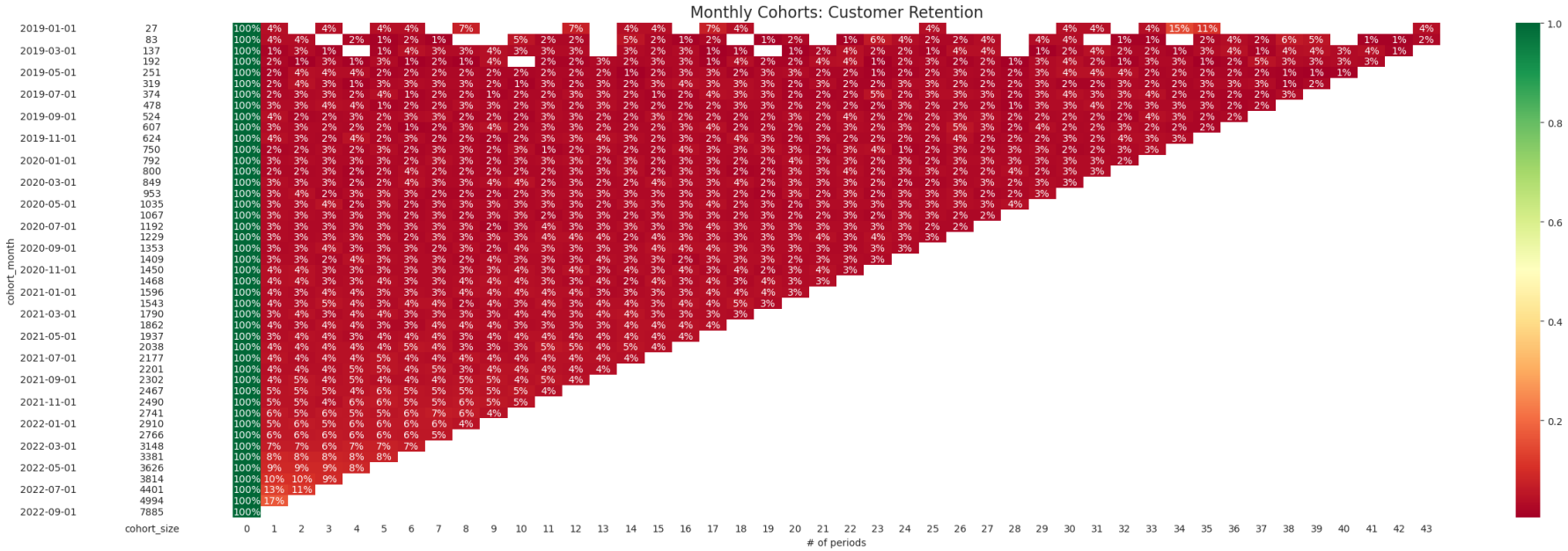
Query Cohort Result

Query results samples:

cohort month	Cohort Size	Periode	Total User Act	Percent Cohort
2019-01-01	22	0	22	1.0
2019-01-01	22	2	1	0.045454545454545456
2019-01-01	22	5	2	0.090909090909090912
2019-01-01	22	21	1	0.045454545454545456
2019-01-01	22	22	1	0.045454545454545456
2019-01-01	22	23	1	0.045454545454545456
2019-01-01	22	24	1	0.045454545454545456
2019-01-01	22	27	1	0.045454545454545456
2019-01-01	22	28	1	0.045454545454545456
2019-01-01	22	30	1	0.045454545454545456
2019-01-01	22	37	1	0.045454545454545456
2019-01-01	22	39	1	0.045454545454545456
2019-01-01	22	40	1	0.045454545454545456
2019-01-01	22	42	1	0.045454545454545456
2019-01-01	22	43	1	0.045454545454545456
2019-01-01	22	44	1	0.045454545454545456
2019-02-01	82	0	82	1.0
2019-02-01	82	1	4	0.04878048780487805
2019-02-01	82	2	2	0.024390243902439025
2019-02-01	82	3	1	0.012195121951219513

Cohort Visualization

Customer retention :



For about four years, user retention buying our product is always 3%!