

Intermediate Assignment

Fitra Rachma Saphira | Section Seoul - Team 5
FSDA Batch 3

Question 1

Create a query to get the total users who completed the order and total orders per month (Skillset: Intermediate SQL)

Please use time frame from Jan 2019 until Apr 22

Expected output:

- Month
- Total Users
- Total Orders

Question 1 Table Result Schema & Table Result

SCHEMA	DETAILS	PREVIEW
Filter Enter property name or value		
Field name	Type	Mode
month	STRING	NULLABLE
total_users	INTEGER	NULLABLE
total_orders	INTEGER	NULLABLE
EDIT SCHEMA		VIEW ROW ACCESS POLICIES

Question_1

QUERY SHARE COPY SNAPSHOT DELETE EXPORT

Row	month	total_users	total_orders
1	2019-01	5	5
2	2019-02	21	21
3	2019-03	39	39
4	2019-04	56	56
5	2019-05	75	75
6	2019-06	107	107
7	2019-07	133	133
8	2019-08	126	126
9	2019-09	175	175
10	2019-10	201	201
11	2019-11	205	205
12	2019-12	268	271
13	2020-01	240	242
14	2020-02	265	266
15	2020-03	306	307
16	2020-04	338	340
17	2020-05	351	354
18	2020-06	398	399
19	2020-07	454	455
20	2020-08	500	504
21	2020-09	502	505
22	2020-10	529	530
23	2020-11	580	582
24	2020-12	576	577
25	2021-01	721	728
26	2021-02	676	682
27	2021-03	770	772
28	2021-04	834	841
29	2021-05	863	867

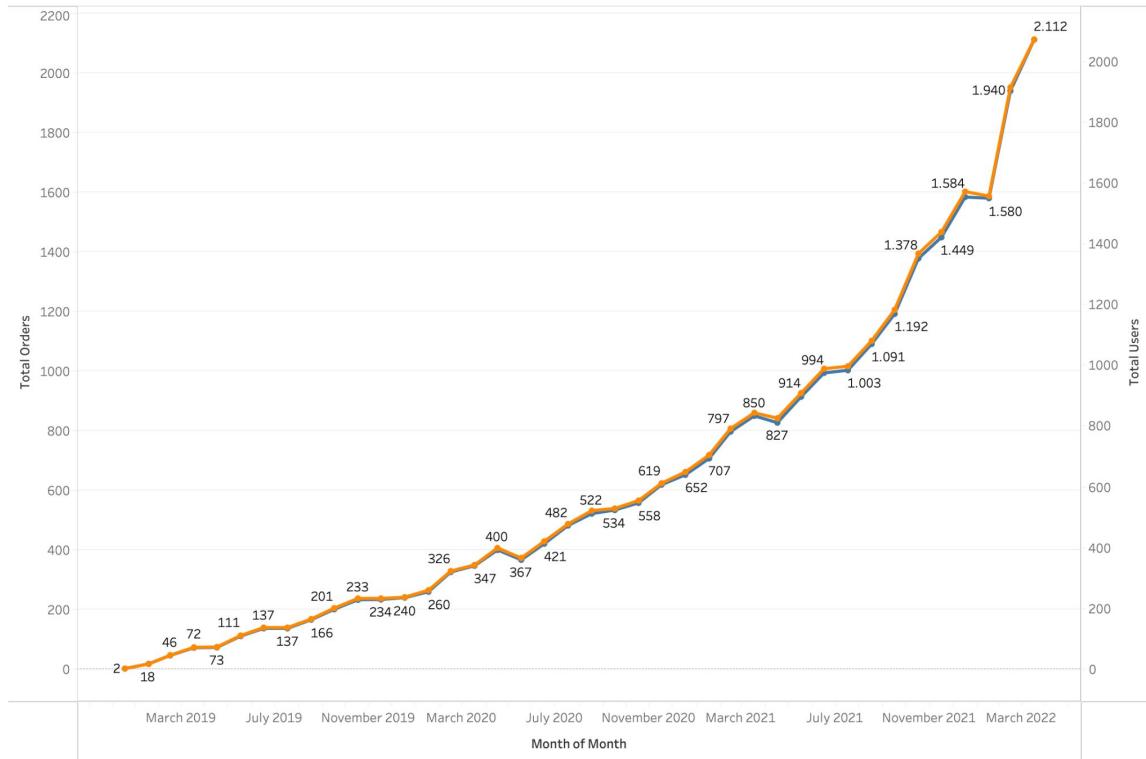
Results per page: 50 ▾ 1 – 40 of 40 | < >

Question 1 SQL Syntax:

```
SELECT format_date ('%Y-%m', a.created_at) month,
       count(distinct a.user_id) total_users,
       count(a.order_id) total_orders,
  FROM `bigquery-public-data.thelook_ecommerce.orders` a
 WHERE created_at BETWEEN '2019-01-01' and '2022-04-30'
       AND status='Complete'
 GROUP BY month
 ORDER BY month
```

Question 1 Insight

Total users who completed the order and total orders per month



- Question 1's line graph shows that the total number of users who completed orders and the total number of orders have both increased.
- The number of customers who completed orders and total orders decreased somewhat in June 2020, May 2021, and February 2022.

Question 2

Create a query to get average order value and total number of unique users, grouped by month (Skillset: Intermediate SQL)

Please use time frame from Jan 2019 until Apr 22

Expected output:

- Month-year
- AOV
- Distinct Users

Question 2 Table Result Schema & Table Result

SCHEMA DETAILS PREVIEW

Filter Enter property name or value

Field name	Type	Mode	Co
month_year	STRING	NULLABLE	
AOV	FLOAT	NULLABLE	
distinct_users	INTEGER	NULLABLE	

[EDIT SCHEMA](#) [VIEW ROW ACCESS POLICIES](#)

Question_2 [QUERY](#) [SHARE](#) [COPY](#) [SNAPSHOT](#) [DELETE](#) [EXPORT](#)

Row	month_year	AOV	distinct_users
1	2019-01	58.24	26
2	2019-02	52.833	87
3	2019-03	53.969	157
4	2019-04	62.017	228
5	2019-05	60.015	349
6	2019-06	61.192	387
7	2019-07	58.317	497
8	2019-08	60.83	553
9	2019-09	62.45	658
10	2019-10	57.806	695
11	2019-11	57.748	819
12	2019-12	57.158	924
13	2020-01	59.183	1054
14	2020-02	60.083	1080
15	2020-03	62.515	1378
16	2020-04	58.899	1359
17	2020-05	60.062	1498
18	2020-06	59.219	1539
19	2020-07	59.28	1723
20	2020-08	60.105	1883
21	2020-09	58.609	1979
22	2020-10	58.342	2292
23	2020-11	58.165	2258
24	2020-12	59.215	2488
25	2021-01	60.26	2721
26	2021-02	60.215	2556
27	2021-03	59.264	2040

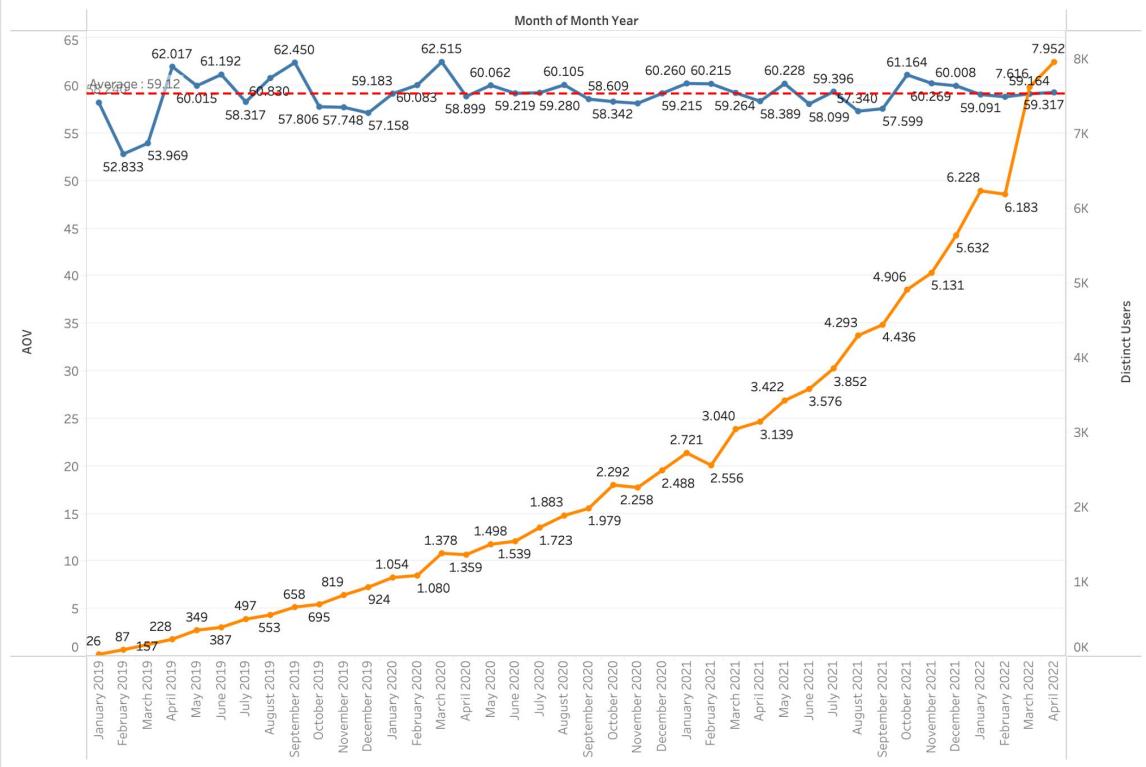
Results per page: 50 ▾ 1 – 40 of 40 | < >

Question 2 SQL Syntax:

```
SELECT format_date ('%Y-%m', a.created_at) month_year,
       round(sum(a.sale_price)/count(a.order_id),3) AOV,
       count(distinct a.user_id) distinct_users
  FROM `bigquery-public-data.thelook_ecommerce.order_items` a
 WHERE a.created_at BETWEEN '2019-01-01' and '2022-04-30'
 GROUP BY month_year
 ORDER BY month_year;
```

Question 2 Insight

The average order value and total number of unique users grouped by month



- The overall trend in the distinct user's line graph is that the number of users has increased significantly, from 26 in January 2019 to 7952 in April 2022.
- The monthly AOV, on the other hand, is rather consistent, averaging 59.12. Despite the fact that the value is a little low from average at the beginning of 2019.

Measure Names

- AOV
- Distinct Users

Question 3

Find the first and last name of users from the youngest and oldest age of each gender. Assume this year is 2022 (Skillset: Intermediate & Advanced SQL)

Expected output:

- Gender
- Youngest Age
- Oldest Age
- First name
- Last name

Question 3 Table Result Schema & Table Result

SCHEMA	DETAILS	PREVIEW																								
<p>Filter Enter property name or value</p> <table><thead><tr><th>Field name</th><th>Type</th><th>Mode</th><th>Collation</th></tr></thead><tbody><tr><td>gender</td><td>STRING</td><td>NULLABLE</td><td></td></tr><tr><td>age</td><td>INTEGER</td><td>NULLABLE</td><td></td></tr><tr><td>type</td><td>STRING</td><td>NULLABLE</td><td></td></tr><tr><td>first_name</td><td>STRING</td><td>NULLABLE</td><td></td></tr><tr><td>last_name</td><td>STRING</td><td>NULLABLE</td><td></td></tr></tbody></table> <p>EDIT SCHEMA VIEW ROW ACCESS POLICIES</p>			Field name	Type	Mode	Collation	gender	STRING	NULLABLE		age	INTEGER	NULLABLE		type	STRING	NULLABLE		first_name	STRING	NULLABLE		last_name	STRING	NULLABLE	
Field name	Type	Mode	Collation																							
gender	STRING	NULLABLE																								
age	INTEGER	NULLABLE																								
type	STRING	NULLABLE																								
first_name	STRING	NULLABLE																								
last_name	STRING	NULLABLE																								

Question_3

QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA	DETAILS	PREVIEW			
Row	gender	age	type	first_name	last_name
1	F	12	Youngest	Amy	Larson
2	F	12	Youngest	Amy	May
3	F	12	Youngest	Amy	Smith
4	F	12	Youngest	Amy	Ford
5	F	12	Youngest	Amy	Arnold
6	F	12	Youngest	Amy	Garcia
7	F	12	Youngest	Amy	Mcclure
8	F	12	Youngest	Amy	Reed
9	F	12	Youngest	Amy	Pugh
10	F	12	Youngest	Amy	Marks
11	F	12	Youngest	Amy	Pacheco
12	F	12	Youngest	Amy	Smith
13	F	12	Youngest	Amy	Ramirez
14	F	12	Youngest	Amy	Mcclure
15	F	12	Youngest	Ana	Thomas
16	F	12	Youngest	Ann	Gilbert
17	F	12	Youngest	Ann	Garcia
18	F	12	Youngest	Ann	Joyce
19	M	12	Youngest	Don	Williams
20	M	12	Youngest	Ian	Edwards
21	M	12	Youngest	Jay	Rangel
22	M	12	Youngest	Jay	Jennings
23	M	12	Youngest	Joe	Schaefer
24	M	12	Youngest	Jon	Foster
25	M	12	Youngest	Jon	Frank
26	M	12	Youngest	Jon	Doyle
27	M	12	Youngest	Jon	Roberts
28	F	12	Youngest	Kim	Chang
29	F	12	Youngest	Kim	Santos

Results per page: 50 ▾ 1 – 50 of 3455 | < < > > |

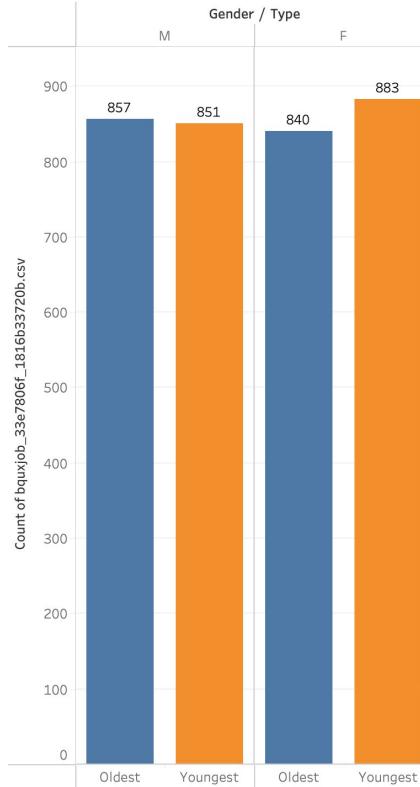
Question 3 SQL Syntax:

```
WITH cte as (
    SELECT a.gender,
           a.age,
           a.first_name,
           a.last_name,
      FROM `bigquery-public-data.thelook_ecommerce.users` a )
SELECT gender,
       age,
CASE WHEN age = (SELECT min(age) FROM `bigquery-public-data.thelook_ecommerce.users`) then 'Youngest'
ELSE 'Oldest' end type,
       first_name,
       last_name
  FROM cte
 WHERE (age = (SELECT min(age) FROM `bigquery-public-data.thelook_ecommerce.users`)
       OR age = (SELECT max(age) FROM `bigquery-public-data.thelook_ecommerce.users`))
       AND gender IS NOT NULL
 ORDER BY 1,2
```

Question 3 Insight

gender	age	type	first_name	last_name
F	12	Youngest	Abigail	Wright
F	12	Youngest	Abigail	Evans
F	12	Youngest	Adrienne	Morrison
F	12	Youngest	Aimee	Hernandez
F	12	Youngest	Alexa	Gray
F	12	Youngest	Alexandra	Rivers
F	12	Youngest	Alexandria	Payne
F	12	Youngest	Alexis	Hunt
F	12	Youngest	Alicia	Jones
F	12	Youngest	Alicia	Sanders
F	12	Youngest	Alisha	Huff
F	12	Youngest	Allison	Koch
F	12	Youngest	Allison	Ponce
F	12	Youngest	Allison	Ward

gender	age	type	first_name	last_name
M	70	Oldest	Aaron	Ryan
M	70	Oldest	Aaron	Bennett
M	70	Oldest	Aaron	Peterson
M	70	Oldest	Aaron	Miller
M	70	Oldest	Aaron	Torres
M	70	Oldest	Aaron	Cardenas
M	70	Oldest	Adam	King
M	70	Oldest	Adam	Williams
M	70	Oldest	Adam	Washington
M	70	Oldest	Adam	Gardner
M	70	Oldest	Adam	Joseph
M	70	Oldest	Adam	Perry
M	70	Oldest	Adam	Gilbert
M	70	Oldest	Adam	Ward



- From the results table for Question 3, we can find out the first and last names of the youngest and oldest users by gender.
- The number of users by gender and age (the oldest and youngest) are not significantly different.

Question 4

Get the top 5 most profitable product and its profit detail breakdown by month (Skillset: Intermediate & Advanced SQL)

Please use time frame from Jan 2019 until Apr 22

Expected output:

- Month
- Product id
- Sales
- Cost
- Profit
- Rank per month

Question 4 Table Result Schema & Table Result

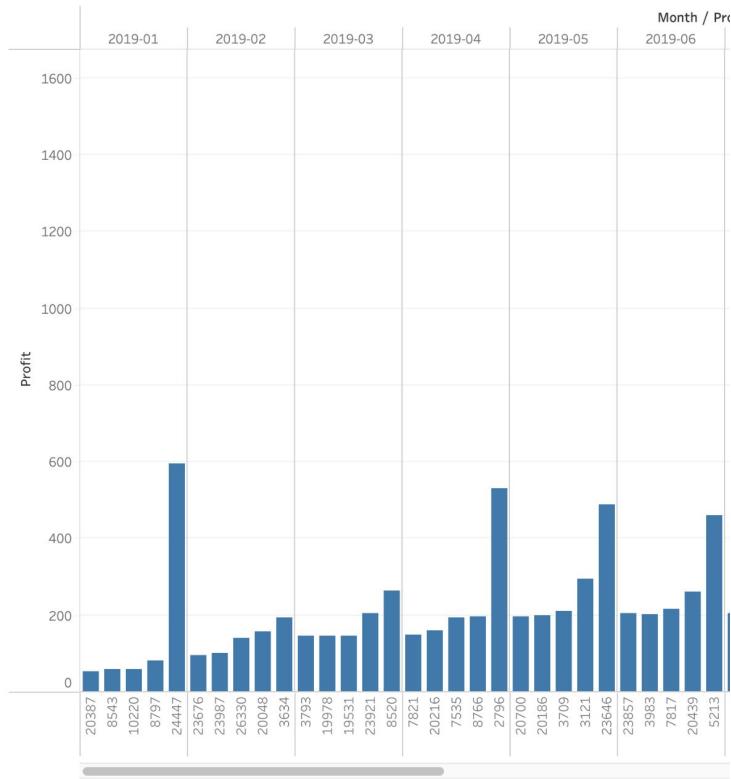
SCHEMA	DETAILS	PREVIEW																																																																																																																																																																															
<input type="text"/> Filter Enter property name or value																																																																																																																																																																																	
<table><thead><tr><th>Field name</th><th>Type</th><th>Mode</th><th>Collation</th></tr></thead><tbody><tr><td>month</td><td>STRING</td><td>NULLABLE</td><td></td></tr><tr><td>product_id</td><td>INTEGER</td><td>NULLABLE</td><td></td></tr><tr><td>sales</td><td>FLOAT</td><td>NULLABLE</td><td></td></tr><tr><td>cost</td><td>FLOAT</td><td>NULLABLE</td><td></td></tr><tr><td>profit</td><td>FLOAT</td><td>NULLABLE</td><td></td></tr><tr><td>ranked</td><td>INTEGER</td><td>NULLABLE</td><td></td></tr></tbody></table>			Field name	Type	Mode	Collation	month	STRING	NULLABLE		product_id	INTEGER	NULLABLE		sales	FLOAT	NULLABLE		cost	FLOAT	NULLABLE		profit	FLOAT	NULLABLE		ranked	INTEGER	NULLABLE																																																																																																																																																				
Field name	Type	Mode	Collation																																																																																																																																																																														
month	STRING	NULLABLE																																																																																																																																																																															
product_id	INTEGER	NULLABLE																																																																																																																																																																															
sales	FLOAT	NULLABLE																																																																																																																																																																															
cost	FLOAT	NULLABLE																																																																																																																																																																															
profit	FLOAT	NULLABLE																																																																																																																																																																															
ranked	INTEGER	NULLABLE																																																																																																																																																																															
EDIT SCHEMA	VIEW ROW ACCESS POLICIES																																																																																																																																																																																
<table><thead><tr><th>Job Information</th><th colspan="3">Results</th><th>JSON</th><th colspan="2">Execution Details</th></tr><tr><th>Row</th><th>Month</th><th>Product ID</th><th>Sales</th><th>Cost</th><th>Profit</th><th>Ranked</th></tr></thead><tbody><tr><td>1</td><td>2019-01</td><td>19548</td><td>265.0</td><td>121.9</td><td>143.1</td><td>1</td></tr><tr><td>2</td><td>2019-01</td><td>28375</td><td>194.95</td><td>75.056</td><td>119.894</td><td>2</td></tr><tr><td>3</td><td>2019-01</td><td>19899</td><td>157.89</td><td>65.209</td><td>92.681</td><td>3</td></tr><tr><td>4</td><td>2019-01</td><td>7697</td><td>114.5</td><td>39.045</td><td>75.455</td><td>4</td></tr><tr><td>5</td><td>2019-01</td><td>5006</td><td>128.0</td><td>66.304</td><td>61.696</td><td>5</td></tr><tr><td>6</td><td>2019-02</td><td>26845</td><td>299.95</td><td>122.68</td><td>177.27</td><td>1</td></tr><tr><td>7</td><td>2019-02</td><td>24129</td><td>249.95</td><td>108.728</td><td>141.222</td><td>2</td></tr><tr><td>8</td><td>2019-02</td><td>16558</td><td>268.0</td><td>145.256</td><td>122.744</td><td>3</td></tr><tr><td>9</td><td>2019-02</td><td>14207</td><td>190.0</td><td>75.62</td><td>114.38</td><td>4</td></tr><tr><td>10</td><td>2019-02</td><td>20466</td><td>199.99</td><td>86.196</td><td>113.794</td><td>5</td></tr><tr><td>11</td><td>2019-03</td><td>19140</td><td>376.0</td><td>186.872</td><td>189.128</td><td>1</td></tr><tr><td>12</td><td>2019-03</td><td>3130</td><td>352.0</td><td>164.736</td><td>187.264</td><td>2</td></tr><tr><td>13</td><td>2019-03</td><td>24262</td><td>310.0</td><td>131.75</td><td>178.25</td><td>3</td></tr><tr><td>14</td><td>2019-03</td><td>2735</td><td>275.0</td><td>105.6</td><td>169.4</td><td>4</td></tr><tr><td>15</td><td>2019-03</td><td>23687</td><td>190.0</td><td>87.97</td><td>102.03</td><td>5</td></tr><tr><td>16</td><td>2019-04</td><td>19956</td><td>600.0</td><td>256.2</td><td>343.8</td><td>1</td></tr><tr><td>17</td><td>2019-04</td><td>19652</td><td>338.0</td><td>165.282</td><td>172.718</td><td>2</td></tr><tr><td>18</td><td>2019-04</td><td>1490</td><td>285.0</td><td>123.975</td><td>161.025</td><td>3</td></tr><tr><td>19</td><td>2019-04</td><td>20235</td><td>259.99</td><td>105.296</td><td>154.694</td><td>4</td></tr><tr><td>20</td><td>2019-04</td><td>24418</td><td>270.0</td><td>118.26</td><td>151.74</td><td>5</td></tr><tr><td>21</td><td>2019-05</td><td>8778</td><td>699.0</td><td>343.908</td><td>355.092</td><td>1</td></tr><tr><td>22</td><td>2019-05</td><td>1357</td><td>658.0</td><td>320.446</td><td>337.554</td><td>2</td></tr><tr><td>23</td><td>2019-05</td><td>8264</td><td>391.6</td><td>160.164</td><td>231.436</td><td>3</td></tr></tbody></table>			Job Information	Results			JSON	Execution Details		Row	Month	Product ID	Sales	Cost	Profit	Ranked	1	2019-01	19548	265.0	121.9	143.1	1	2	2019-01	28375	194.95	75.056	119.894	2	3	2019-01	19899	157.89	65.209	92.681	3	4	2019-01	7697	114.5	39.045	75.455	4	5	2019-01	5006	128.0	66.304	61.696	5	6	2019-02	26845	299.95	122.68	177.27	1	7	2019-02	24129	249.95	108.728	141.222	2	8	2019-02	16558	268.0	145.256	122.744	3	9	2019-02	14207	190.0	75.62	114.38	4	10	2019-02	20466	199.99	86.196	113.794	5	11	2019-03	19140	376.0	186.872	189.128	1	12	2019-03	3130	352.0	164.736	187.264	2	13	2019-03	24262	310.0	131.75	178.25	3	14	2019-03	2735	275.0	105.6	169.4	4	15	2019-03	23687	190.0	87.97	102.03	5	16	2019-04	19956	600.0	256.2	343.8	1	17	2019-04	19652	338.0	165.282	172.718	2	18	2019-04	1490	285.0	123.975	161.025	3	19	2019-04	20235	259.99	105.296	154.694	4	20	2019-04	24418	270.0	118.26	151.74	5	21	2019-05	8778	699.0	343.908	355.092	1	22	2019-05	1357	658.0	320.446	337.554	2	23	2019-05	8264	391.6	160.164	231.436	3
Job Information	Results			JSON	Execution Details																																																																																																																																																																												
Row	Month	Product ID	Sales	Cost	Profit	Ranked																																																																																																																																																																											
1	2019-01	19548	265.0	121.9	143.1	1																																																																																																																																																																											
2	2019-01	28375	194.95	75.056	119.894	2																																																																																																																																																																											
3	2019-01	19899	157.89	65.209	92.681	3																																																																																																																																																																											
4	2019-01	7697	114.5	39.045	75.455	4																																																																																																																																																																											
5	2019-01	5006	128.0	66.304	61.696	5																																																																																																																																																																											
6	2019-02	26845	299.95	122.68	177.27	1																																																																																																																																																																											
7	2019-02	24129	249.95	108.728	141.222	2																																																																																																																																																																											
8	2019-02	16558	268.0	145.256	122.744	3																																																																																																																																																																											
9	2019-02	14207	190.0	75.62	114.38	4																																																																																																																																																																											
10	2019-02	20466	199.99	86.196	113.794	5																																																																																																																																																																											
11	2019-03	19140	376.0	186.872	189.128	1																																																																																																																																																																											
12	2019-03	3130	352.0	164.736	187.264	2																																																																																																																																																																											
13	2019-03	24262	310.0	131.75	178.25	3																																																																																																																																																																											
14	2019-03	2735	275.0	105.6	169.4	4																																																																																																																																																																											
15	2019-03	23687	190.0	87.97	102.03	5																																																																																																																																																																											
16	2019-04	19956	600.0	256.2	343.8	1																																																																																																																																																																											
17	2019-04	19652	338.0	165.282	172.718	2																																																																																																																																																																											
18	2019-04	1490	285.0	123.975	161.025	3																																																																																																																																																																											
19	2019-04	20235	259.99	105.296	154.694	4																																																																																																																																																																											
20	2019-04	24418	270.0	118.26	151.74	5																																																																																																																																																																											
21	2019-05	8778	699.0	343.908	355.092	1																																																																																																																																																																											
22	2019-05	1357	658.0	320.446	337.554	2																																																																																																																																																																											
23	2019-05	8264	391.6	160.164	231.436	3																																																																																																																																																																											
<p>Results per page: 50 ▾ 1 – 50 of 200 < <</p>																																																																																																																																																																																	

Question 4 SQL Syntax:

```
WITH profit_products AS (
    SELECT format_date ('%Y-%m', a.created_at) month,
           a.product_id,
           ROUND(sum(a.sale_price),3) sales,
           ROUND(sum(b.cost),3) cost,
           ROUND(sum(a.sale_price)-sum(b.cost),3) profit,
      FROM `bigquery-public-data.thelook_ecommerce.order_items` a
     LEFT JOIN `bigquery-public-data.thelook_ecommerce.products` b
       ON a.product_id=b.id
      GROUP BY month, a.product_id
),
ranked_table AS (
    SELECT *,
           DENSE_RANK() OVER (PARTITION BY month ORDER BY profit DESC) ranked
      FROM profit_products
)
SELECT *
  FROM ranked_table
 WHERE month BETWEEN '2019-01' AND '2022-04'
   AND ranked <= 5
ORDER BY month, ranked
```

Question 4 Insight

Top 5 most profitable product and its profit detail breakdown by month



- The results of query number 4 show which five products are the most profitable per month from January 2019 to April 2022, as shown in the table and graph.

Question 5

Create a query to get Month to Date of total revenue in each product categories of past 3 months, (current date 15 april 2022) breakdown by date (Skillset: Intermediate and Advanced SQL)

Expected Output:

- Date (in date format)
- Product Categories
- Revenue

Question 5 Table Result Schema & Table Result

SCHEMA	DETAILS	PREVIEW
<hr/>		
<input type="button" value="Filter"/> Enter property name or value		
Field name	Type	Mode
date_order	DATE	NULLABLE
product_category	STRING	NULLABLE
revenue_mtd	FLOAT	NULLABLE

[EDIT SCHEMA](#)[VIEW ROW ACCESS POLICIES](#)

Question_5

QUERY SHARE COPY SNAPSHOT DELETE EXPORT

Row	date_order	product_category	revenue_mtd
1	2022-02-01	Accessories	944.14
2	2022-02-02	Accessories	1854.98
3	2022-02-03	Accessories	2638.31
4	2022-02-04	Accessories	3134.79
5	2022-02-05	Accessories	4175.18
6	2022-02-06	Accessories	5082.32
7	2022-02-07	Accessories	6021.57
8	2022-02-08	Accessories	6887.45
9	2022-02-09	Accessories	8256.0
10	2022-02-10	Accessories	9136.28
11	2022-02-11	Accessories	10433.07
12	2022-02-12	Accessories	11396.25
13	2022-02-13	Accessories	12452.19
14	2022-02-14	Accessories	13128.72
15	2022-02-15	Accessories	13774.24
16	2022-03-01	Accessories	1018.37
17	2022-03-02	Accessories	1915.59
18	2022-03-03	Accessories	2237.7
19	2022-03-04	Accessories	3294.9
20	2022-03-05	Accessories	4178.9
21	2022-03-06	Accessories	4707.36
22	2022-03-07	Accessories	5117.67
23	2022-03-08	Accessories	6444.31
24	2022-03-09	Accessories	7003.9
25	2022-03-10	Accessories	8024.2
26	2022-03-11	Accessories	8736.11
27	2022-03-12	Accessories	9402.49

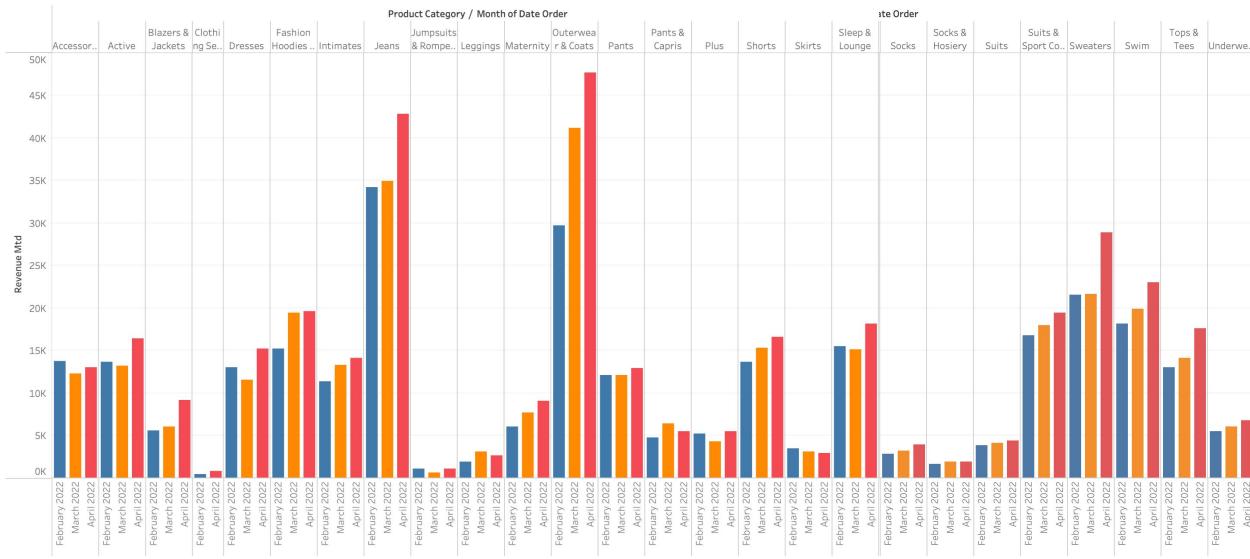
Results per page: 50 ▾ 1 – 50 of 1130 | < >

Question 5 SQL Syntax:

```
WITH cte AS (
    SELECT DATE_TRUNC(a.created_at, day) date_order,
           b.category product_category,
           ROUND(SUM(a.sale_price), 3) revenue
      FROM `bigquery-public-data.thelook_ecommerce.order_items` a
     LEFT JOIN `bigquery-public-data.thelook_ecommerce.products` b
       ON a.product_id=b.id
      GROUP BY 1,2
)
SELECT EXTRACT(date from date_order) date_order,
       product_category,
       ROUND(SUM(revenue) OVER (PARTITION BY product_category,EXTRACT(month from date_order) ORDER BY date_order),3) revenue_mtd
  FROM cte
 WHERE EXTRACT(date from date_order) BETWEEN '2022-02-01' AND '2022-04-15'
   AND EXTRACT(day from date_order) BETWEEN 1 AND 15
 ORDER BY 2,1
```

Question 5 Insight

Month to Date of total revenue in each product categories of past 3 months



- From the results table and graph for Question 5, we can determine the revenue growth of each product categories in the last three months from the present date of 15 April 2022
- April 2022 is the month with the highest total revenue across all product categories.
- Outerwear and coats products have the highest total revenue in the last three months when compared to other product categories, and followed by jeans products.

Advanced Assignment

Question 6

Find monthly growth of TPO (# of completed orders) and TPV (# of revenue) in percentage breakdown by product categories, ordered by time descendingly. After analyzing the monthly growth, is there any interesting insight that we can get?

Please use time frame from Jan 2019 until Apr 2022

Expected output:

- Month
- Categories
- Order Growth
- Revenue Growth (%)

Question 6 Table Result Schema & Table Result

Question_6 QUERY SHARE

SCHEMA DETAILS PREVIEW

Filter Enter property name or value

Field name	Type	Mode	Col
month_order	STRING	NULABLE	
product_category	STRING	NULABLE	
order_growth	STRING	NULABLE	
revenue_growth	STRING	NULABLE	

EDIT SCHEMA VIEW ROW ACCESS POLICIES

Query results

SAVE RESULTS EXPLORE DATA

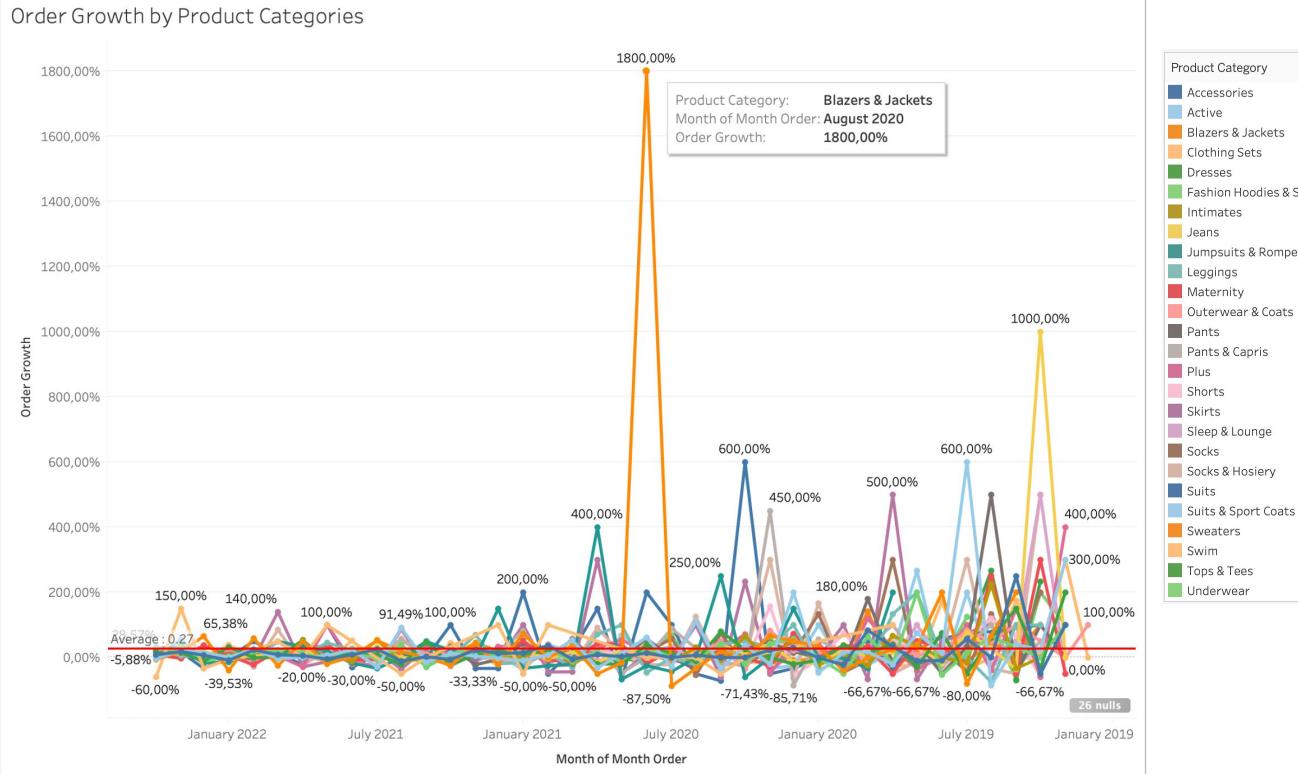
JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
Row	month_order	product_category	order_growth	revenue_growth
1	2022-04	Accessories	8.45%	29.39%
2	2022-03	Accessories	18.33%	15.65%
3	2022-02	Accessories	7.14%	12.25%
4	2022-01	Accessories	-8.2%	-22.45%
5	2021-12	Accessories	23.23%	26.91%
6	2021-11	Accessories	8.79%	8.25%
7	2021-10	Accessories	5.81%	6.92%
8	2021-09	Accessories	-4.44%	-24.71%
9	2021-08	Accessories	8.43%	25.24%
10	2021-07	Accessories	25.76%	71.48%
11	2021-06	Accessories	-10.81%	-28.8%
12	2021-05	Accessories	2.78%	2.86%
13	2021-04	Accessories	-6.49%	-13.82%
14	2021-03	Accessories	24.19%	44.28%
15	2021-02	Accessories	12.73%	18.13%
16	2021-01	Accessories	14.58%	0.34%
17	2020-12	Accessories	37.14%	47.83%
18	2020-11	Accessories	-5.41%	-8.92%
19	2020-10	Accessories	8.82%	12.48%
20	2020-09	Accessories	3.03%	-15.28%
21	2020-08	Accessories	13.79%	6.92%
22	2020-07	Accessories	0%	42.58%
23	2020-06	Accessories	7.41%	14.54%

Results per page: 50 ▾ 1 – 50 of 967 | < > >>

Question 6 SQL Syntax:

```
WITH cte AS (
    SELECT FORMAT_DATE('%Y-%m', DATE_TRUNC(a.created_at, month)) month_order,
           b.category product_category,
           COUNT(DISTINCT a.order_id) num_order,
           ROUND(SUM(a.sale_price), 2) revenue
      FROM `bigquery-public-data.thelook_ecommerce.order_items` a
     LEFT JOIN `bigquery-public-data.thelook_ecommerce.products` b
       ON a.product_id=b.id
     WHERE a.created_at BETWEEN '2019-01-01' AND '2022-04-30'
       AND status='Complete'
     GROUP BY 1,2
),
cte2 AS(
    SELECT *,
           LAG(num_order) OVER(PARTITION BY product_category ORDER BY month_order) prev_num_order,
           LAG(revenue) OVER(PARTITION BY product_category ORDER BY month_order) prev_revenue
      FROM cte
     ORDER BY 2,1
)
SELECT cte2.month_order,
       cte2.product_category,
       CONCAT(ROUND((num_order-prev_num_order)/prev_num_order*100,2),"") order_growth,
       CONCAT(ROUND((revenue-prev_revenue)/prev_revenue*100,2),"") revenue_growth
     FROM cte2
    ORDER BY 2,1 DESC
```

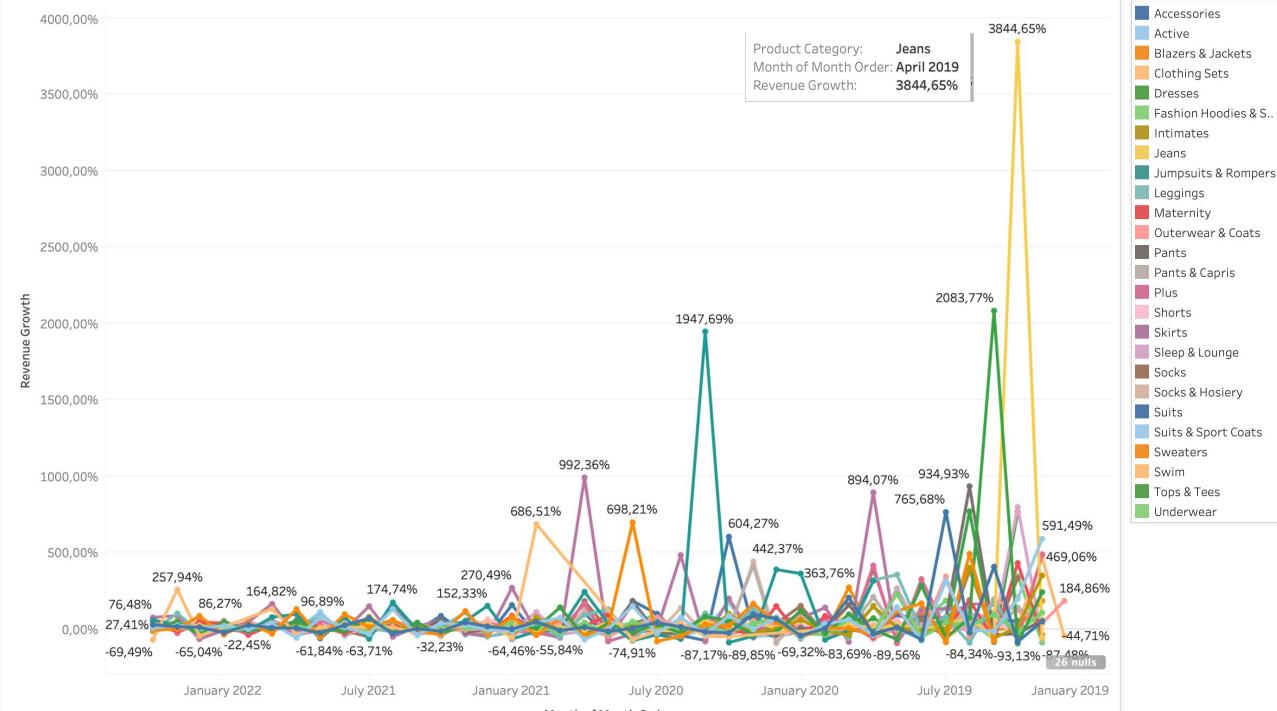
Question 6 Insight



- From January 2019 to April 2022, order growth of overall product categories is tend to decreases.
 - The line graph shows that order growth for blazers and jackets increased significantly in August 2020, from -15.79 percent to 1800 percent.
 - The second large increase in order growth, at 1000 percent, is in jeans products.

Question 6 Insight

Revenue Growth by Product Categories



- Overall revenue growth appears to be decreasing and stable from January 2021 to April 2022.
- The majority of the significant increase in revenue growth occurred between January 2019 and early 2021.
- The most significant increase was in jeans products by 3845% in April 2019.
- Another significant growth is in dress product in May 2019 and jumpsuit product in May 2020.

Question 7

Create monthly retention cohorts (the groups, or cohorts, can be defined based upon the date that a user purchased a product) and then how many of them (%) coming back for the following months in 2019-2022. After analyzing the retention cohort, is there any interesting insight that we can get? (Skillset: Intermediate SQL (Lecture 3 & 4) & Advanced SQL (Lecture 6))

Notes: initial start date can be defined using first purchase from each user in table order

Expected output:

- Month
- M (# of users in current month)
- M1 (# of users in following months)
- M2 (# of users in following two months)
- M3 (# of users in following three months)

Question 7 Table Result Schema & Table Result

Question_7

QUERY SHARE

SCHEMA DETAILS PREVIEW

Filter Enter property name or value

Field name	Type	Mode
cohort_month	STRING	NULLABLE
cohort_size	INTEGER	NULLABLE
next_purchased	STRING	NULLABLE
diff_next_purchased	INTEGER	NULLABLE
num_users	INTEGER	NULLABLE
percentage	NUMERIC	NULLABLE

EDIT SCHEMA **VIEW ROW ACCESS POLICIES**

Question_7

QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW

Row	cohort_month	cohort_size	next_purchased	diff_next_purchased	num_users	percentage
1	2019-01	9	2019-01	0	9	1
2	2019-01	9	2019-02	1	1	0.11111111
3	2019-02	20	2019-02	0	20	1
4	2019-02	20	2019-03	1	1	0.05
5	2019-02	20	2019-05	3	1	0.05
6	2020-02	281	2020-02	0	281	1
7	2020-02	281	2020-03	1	11	0.039145907
8	2020-02	281	2020-04	2	6	0.021352313
9	2020-02	281	2020-05	3	4	0.014234875
10	2022-04	1818	2022-04	0	1818	1
11	2022-04	1818	2022-05	1	265	0.145764576
12	2022-04	1818	2022-06	2	142	0.078107811
13	2020-10	539	2020-10	0	539	1
14	2020-10	539	2020-11	1	20	0.037105751
15	2020-10	539	2020-12	2	26	0.048237477
16	2020-10	539	2021-01	3	25	0.046382189
17	2020-11	542	2020-11	0	542	1
18	2020-11	542	2020-12	1	24	0.044280443
19	2020-11	542	2021-01	2	22	0.040590406
20	2020-11	542	2021-02	3	19	0.035055351
21	2022-06	2339	2022-06	0	2339	1
22	2019-03	42	2019-03	0	42	1
23	2019-03	42	2019-04	1	1	0.023809524
24	2019-03	42	2019-05	2	1	0.023809524
25	2021-02	561	2021-02	0	561	1
26	2021-02	561	2021-03	1	22	0.039215686
27	2021-02	561	2021-04	2	20	0.025665621

Results per page: 50 ▾ 1 – 50 of 157 | < >

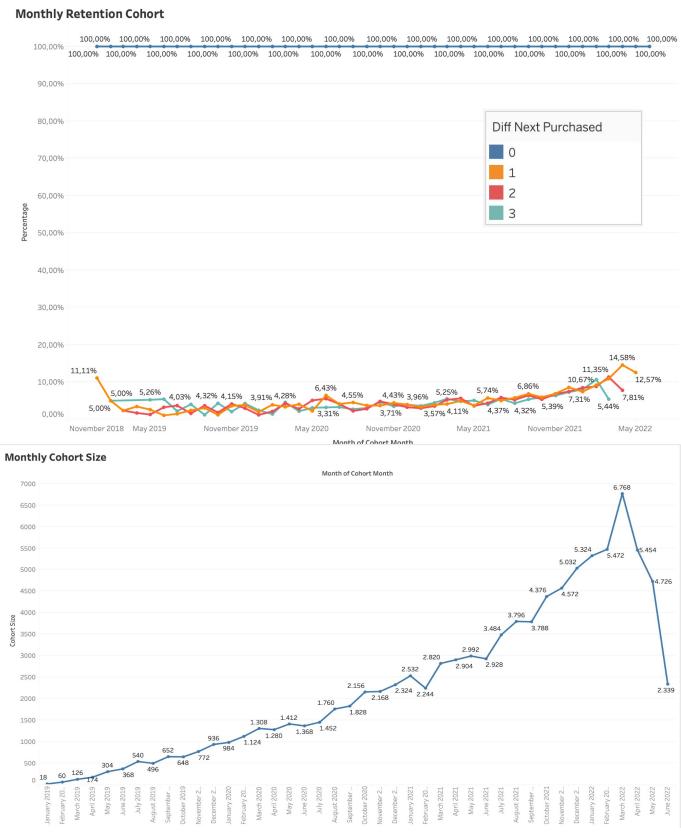
Question 7 SQL Syntax:

```
With cohort_items as(
    SELECT a.user_id,
        MIN(date(date_trunc(a.created_at, month))) cohort_month
    FROM `bigquery-public-data.thelook_ecommerce.orders` a
    WHERE a.status = 'Complete'
    AND EXTRACT(year from a.created_at) BETWEEN 2019 AND 2022
    GROUP BY 1
),
user_reten as(
    SELECT a.user_id,
        date(date_trunc(a.created_at, month))next_purchased,
        date_diff(date(date_trunc(a.created_at, MONTH)),b.cohort_month,month)
        diff_next_purchased
    FROM `bigquery-public-data.thelook_ecommerce.orders` a
    LEFT JOIN cohort_items b
    ON a.user_id=b.user_id
    WHERE
        date_diff(date(date_trunc(a.created_at, MONTH)),b.cohort_month,month)BETWEEN 0
        AND 3
        AND EXTRACT(year from a.created_at) BETWEEN 2019 AND 2022
        GROUP BY 1,3,2
),
cohort_size as (
    SELECT cohort_month,
        COUNT(1) as num_users
    FROM cohort_items
    GROUP BY 1
    ORDER BY 1
),
```

```
retention_table as (
    SELECT c.cohort_month,
        a.next_purchased,
        a.diff_next_purchased,
        COUNT(1) as num_users
    FROM user_reten a
    LEFT JOIN cohort_items c ON a.user_id = c.user_id
    group by 1, 2, 3
)
SELECT format_date('%Y-%m',a.cohort_month) cohort_month,
    b.num_users as cohort_size,
    format_date('%Y-%m',a.next_purchased) next_purchased,
    a.diff_next_purchased,
    a.num_users,
    CONCAT(CAST(a.num_users as decimal)/ b.num_users*100, "%") as percentage
FROM retention_table a
LEFT JOIN cohort_size b
ON a.cohort_month = b.cohort_month
WHERE a.cohort_month IS NOT NULL
ORDER BY 1,3,4
```

Question 7 Insight

	SUM of percents diff_next_purchased				cohort_size
cohort_month	0	1	2	3	
2019-01	100.00%	11.11%			18
2019-02	100.00%	5.00%	5.00%		60
2019-03	100.00%	2.38%	2.38%		126
2019-04	100.00%	3.45%	1.72%		174
2019-05	100.00%	2.63%	1.32%	5.26%	304
2019-06	100.00%	1.09%	3.26%	5.43%	368
2019-07	100.00%	1.48%	3.70%	2.22%	540
2019-08	100.00%	2.42%	1.61%	4.03%	496
2019-09	100.00%	3.07%	3.68%	1.23%	652
2019-10	100.00%	1.23%	1.85%	4.32%	648
2019-11	100.00%	3.63%	4.15%	2.07%	772
2019-12	100.00%	3.85%	2.99%	4.27%	936
2020-01	100.00%	2.03%	1.22%	2.44%	984
2020-02	100.00%	3.91%	2.14%	1.42%	1124
2020-03	100.00%	3.36%	4.28%	4.59%	1308
2020-04	100.00%	4.06%	2.81%	2.19%	1280
2020-05	100.00%	2.27%	5.10%	3.12%	1412
2020-06	100.00%	6.43%	5.56%	3.22%	1368
2020-07	100.00%	4.13%	4.13%	3.31%	1452
2020-08	100.00%	4.55%	2.27%	2.73%	1760
2020-09	100.00%	3.72%	2.84%	3.06%	1828
2020-10	100.00%	3.71%	4.82%	4.64%	2156
2020-11	100.00%	4.43%	4.06%	3.51%	2168
2020-12	100.00%	3.96%	3.27%	3.96%	2324
2021-01	100.00%	3.32%	3.00%	3.63%	2532
2021-02	100.00%	3.92%	3.57%	4.46%	2244
2021-03	100.00%	4.11%	5.25%	5.53%	2820
2021-04	100.00%	4.96%	5.65%	4.82%	2904
2021-05	100.00%	3.74%	3.61%	5.08%	2992
2021-06	100.00%	5.74%	4.37%	3.96%	2928
2021-07	100.00%	5.05%	5.86%	5.51%	3484
2021-08	100.00%	5.80%	5.27%	4.32%	3796
2021-09	100.00%	6.86%	6.44%	5.39%	3788
2021-10	100.00%	5.94%	5.39%	5.94%	4376
2021-11	100.00%	6.91%	6.82%	6.39%	4572
2021-12	100.00%	8.59%	7.47%	7.31%	5032
2022-01	100.00%	7.44%	8.49%	8.04%	5324
2022-02	100.00%	9.21%	8.85%	10.67%	5472
2022-03	100.00%	10.93%	11.35%	5.44%	6768
2022-04	100.00%	14.58%	7.81%		5454
2022-05	100.00%	12.57%			4726
2022-06	100.00%				2339
AVERAGE	100.00%	5.06%	4.43%	4.40%	95809



- According to this cohort analysis, user retention from the current month to the next three months tends to be constant from 2019 to 2022
- The line graph cohort size shows that it gradually increased from early 2019 to March 2022, but then dropped significantly from 6768 users to 2339 users from March to June 2022.

