SQL Task

By Azzam Gilas Tirani

Case Overview

Data analysts are expected to use SQL to access, read, manipulate, and analyze the data stored in a database and generate useful insights to drive an informed decision-making process.

Dataset Overview

theLook eCommerce

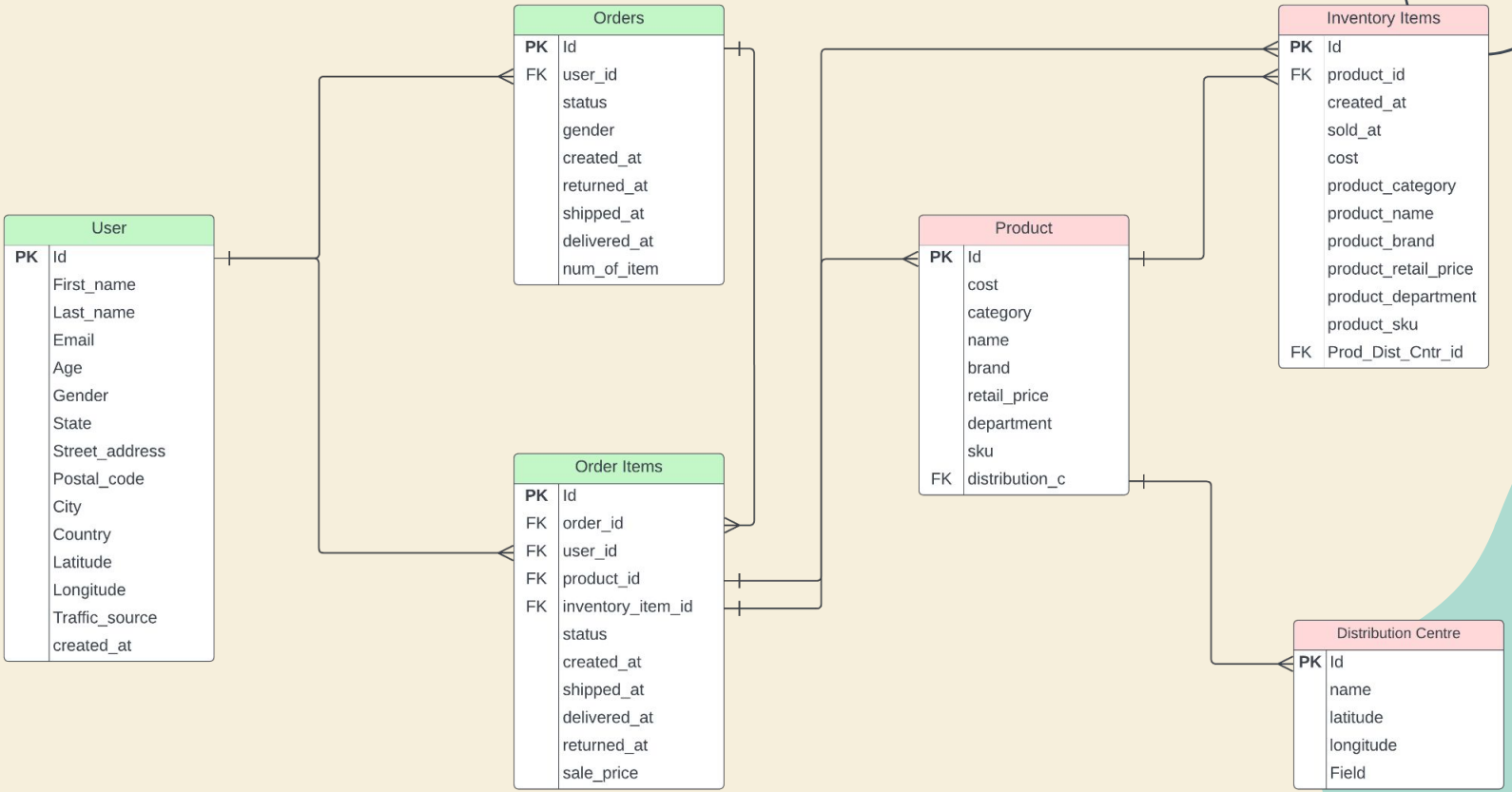
BigQuery Public Data

Synthetic eCommerce and Digital Marketing data

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

ERD

Here


Orders
PK Id
FK user_id
status
gender
created_at
returned_at
shipped_at
delivered_at
num_of_item
Inventory Items
PK Id
FK product_id
created_at
sold_at
cost
product_category
product_name
product_brand
product_retail_price
product_department
product_sku
FK Prod_Dist_Cntr_id
User
PK Id
First_name
Last_name
Email
Age
Gender
State
Street_address
Postal_code
City
Country
Latitude
Longitude
Traffic_source
created_at
Product
PK Id
cost
category
name
brand
retail_price
department
sku
FK distribution_c
Order Items
PK Id
FK order_id
FK user_id
FK product_id
FK inventory_item_id
status
created_at
shipped_at
delivered_at
returned_at
sale_price
Distribution Centre
PK Id
name
latitude
longitude
Field

01

Create a query to get **the number of unique** users, **number of orders**, and total sale price per status and month

From Jan 2019 until Apr 2022

Schema & Preview Q1

Question_1 QUERY SHARE

SCHEMA DETAILS PREVIEW

Filter Enter property name or value

Field name	Type	Mode
Month_Year	DATE	NULLABLE
status	STRING	NULLABLE
Unq_Number_ID	INTEGER	NULLABLE
Number_Order	INTEGER	NULLABLE
total_sales	FLOAT	NULLABLE

Question_1 QUERY SHARE COPY SNAPSHOT DELETE

SCHEMA DETAILS PREVIEW

Row	Month_Year	status	Unq_Numbe...	Number_Or...	total_sales
1	2019-01-01	Shipped	4	6	401.0
2	2019-02-01	Shipped	18	21	1598.0
3	2019-03-01	Shipped	56	87	5432.0
4	2019-04-01	Shipped	70	111	6350.0
5	2019-05-01	Shipped	85	104	6355.0
6	2019-06-01	Shipped	117	169	9372.0
7	2019-07-01	Shipped	134	195	11700.0
8	2019-08-01	Shipped	156	207	11446.0
9	2019-09-01	Shipped	155	209	14344.0
10	2019-10-01	Shipped	191	278	17869.0
11	2019-11-01	Shipped	198	284	15196.0
12	2019-12-01	Shipped	249	356	20529.0

Q1 SQL Sntax : Here

```sql
SELECT
 DATE(DATE_TRUNC(created_at,MONTH)) AS Month_Year
  , status
  , COUNT (distinct user_id) AS Unq_Number_ID
  , COUNT (id) AS Number_Order
  , ROUND(SUM(sale_price)) AS total_sales
FROM `bigquery-public-data.thelook_ecommerce.order_items`
WHERE DATE(DATE_TRUNC(created_at,YEAR)) BETWEEN '2019-01-01' AND '2022-08-31'
GROUP BY 1,2
ORDER BY 1,2
```

02

Create a query to get **frequencies, average order value (AOV)** and **total** number of **unique users** where status is **complete** grouped by month

From Jan 2019 until Apr 2022

Schema & Preview Q2

Question_2 QUERY SHARE

SCHEMA DETAILS PREVIEW

Filter Enter property name or value

Field name	Type	Mode
Month_Year	DATE	NULLABLE
Order_Status	STRING	NULLABLE
AOV	FLOAT	NULLABLE
Frequency	FLOAT	NULLABLE
Unique_Buyers	INTEGER	NULLABLE

Question_2 QUERY SHARE COPY SNAPSHOT DELET

SCHEMA DETAILS PREVIEW

Row	Month_Year	Order_Status	AOV	Frequency	Unique_Buy...
1	2019-01-01	Complete	42.0	1.0	3
2	2019-06-01	Complete	52.0	1.0	86
3	2019-11-01	Complete	56.0	1.0	176
4	2020-03-01	Complete	56.0	1.0	254
5	2021-08-01	Complete	56.0	1.0	871
6	2019-08-01	Complete	58.0	1.0	122
7	2022-01-01	Complete	58.0	1.0	1118
8	2019-10-01	Complete	60.0	1.0	180
9	2019-12-01	Complete	60.0	1.0	178
10	2021-01-01	Complete	60.0	1.0	569
11	2021-02-01	Complete	60.0	1.0	554
12	2021-04-01	Complete	60.0	1.0	606

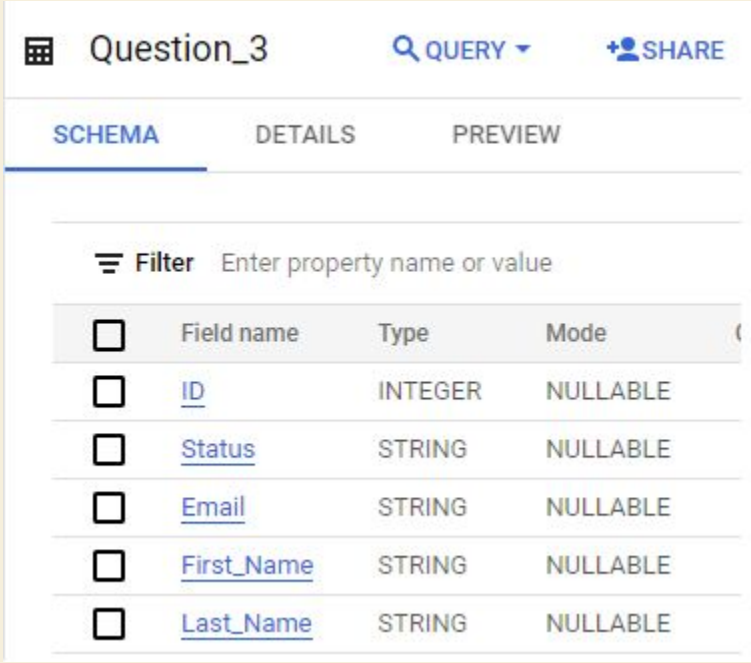
Q2 SQL Sntax : Here

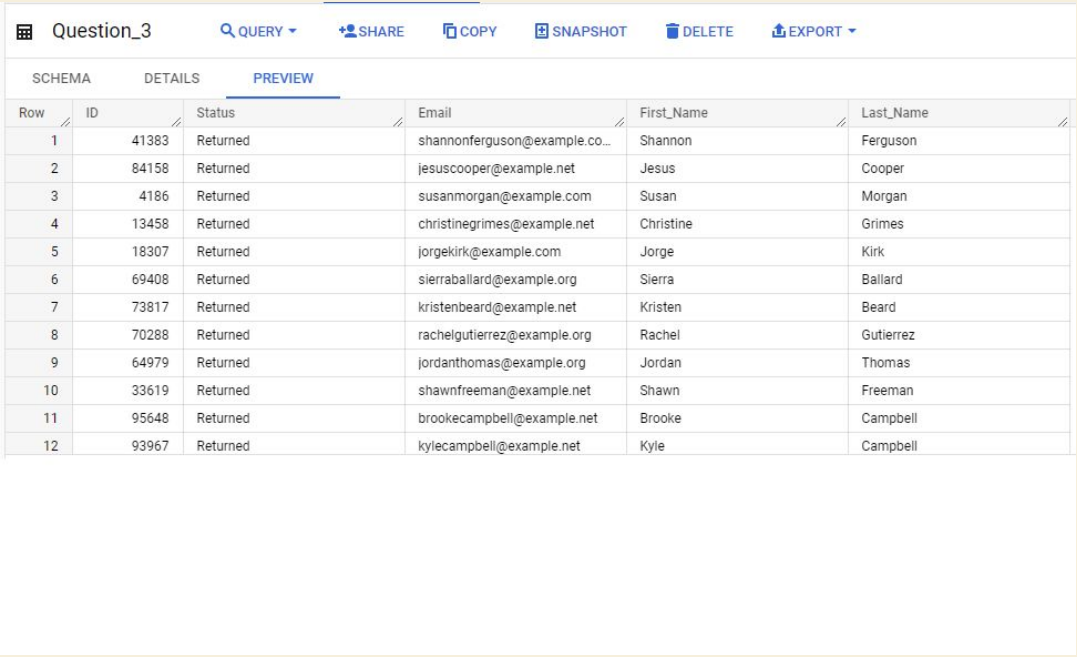
```sql
SELECT
  DATE(DATE_TRUNC(created_at,MONTH)) AS Month_Year
  , status AS Order_Status
  , ROUND((sum(sale_price) / (count(order_id)))) AS AOV
  , ROUND (COUNT(DISTINCT user_id) / (COUNT(order_id))) AS Frequency
  , COUNT (DISTINCT user_id) AS Unique_Buyers
FROM `bigquery-public-data.thelook_ecommerce.order_items`
WHERE DATE(DATE_TRUNC(created_at,YEAR)) BETWEEN '2019-01-01' AND '2022-08-31' AND status= 'Complete'
GROUP BY 1,2
ORDER BY 2,1
```

03

Find the user id, email, first and last name of users whose status is **refunded** on August 2022

Schema & Preview Q3





Row	ID	Status	Email	First_Name	Last_Name
1	41383	Returned	shannonferguson@example.co...	Shannon	Ferguson
2	84158	Returned	jesuscooper@example.net	Jesus	Cooper
3	4186	Returned	susanmorgan@example.com	Susan	Morgan
4	13458	Returned	christinegrimes@example.net	Christine	Grimes
5	18307	Returned	jorgekirk@example.com	Jorge	Kirk
6	69408	Returned	sierraballard@example.org	Sierra	Ballard
7	73817	Returned	kristenbeard@example.net	Kristen	Beard
8	70288	Returned	rachelgutierrez@example.org	Rachel	Gutierrez
9	64979	Returned	jordanthomas@example.org	Jordan	Thomas
10	33619	Returned	shawnfreeman@example.net	Shawn	Freeman
11	95648	Returned	brookecampbell@example.net	Brooke	Campbell
12	93967	Returned	kylecampbell@example.net	Kyle	Campbell

Q3 SQL Sntax : Here

```sql
SELECT
  O.user_id AS ID
  , O.status AS Status
  , U.email AS Email
  , U.first_name AS First_Name
  , U.last_name AS Last_Name
FROM `bigquery-public-data.thelook_ecommerce.orders` AS O
JOIN `bigquery-public-data.thelook_ecommerce.users` AS U
ON O.user_id = U.id
Where O.status = 'Returned'
```

04

Get the Top 5 least and most profitable product over all time

Schema & Preview Q4

Question_4 QUERY SHARE

SCHEMA DETAILS PREVIEW

Filter Enter property name or value

Field name	Type	Mode
Product_Id	INTEGER	NULLABLE
Product_Name	STRING	NULLABLE
Retail_Price	FLOAT	NULLABLE
Cost	FLOAT	NULLABLE
Profit	FLOAT	NULLABLE
Profit_Rank_Alltime	STRING	NULLABLE

Question_4 QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW

Row	Product_Id	Product_Name	Retail_Price	Cost	Profit	Profit_Rank_Alltime
1	24053	The North Face Denali Down M...	903.0	436.149	1867.40398...	Rank 3 Top Profitable
2	24447	Darla	999.0	404.595	2377.61999...	Rank 1 Top Profitable
3	24061	Men's Classic Sheepskin B-3 B...	595.0	270.725	1621.37499...	Rank 5 Top Profitable
4	8425	Canada Goose Women's Whistl...	695.0	296.07	1994.65000...	Rank 2 Top Profitable
5	7804	MiH Jeans Women's Aztec Jac...	495.0	169.785	1626.07499...	Rank 4 Top Profitable
6	9204	Pink Ribbon Breast Cancer Awareness Knee High Socks Great for Sports Teams Fundraising Relay for Life Walk Survivor (Style 21)	1.95	0.772	1.17780003...	Rank 3 Least Profitable
7	15395	Retractable Colorful Rhinestone Lanyards with Breakaway Feature ID Badge Holder & Key	2.67	1.415	1.25490003...	Rank 5 Least Profitable

Q4 SQL Sntax : Here

```sql
WITH
expense AS
(
  SELECT *
  FROM `bigquery-public-data.thelook_ecommerce.products`
),
revenue AS
(
  SELECT *
  FROM `bigquery-public-data.thelook_ecommerce.order_items`
  WHERE status = 'Complete'
),
profit as
(
  SELECT
    expense.id AS prduct_id
    , expense.name AS product_name
    , ROUND(expense.cost,3) AS product_cost
    , ROUND(revenue.sale_price,3) AS retail_price
    , SUM(revenue.sale_price) - (SUM(expense.cost)) AS profit
  FROM revenue
  LEFT JOIN expense
  ON expense.id = revenue.product_id
  GROUP BY 1,2,3,4
  ORDER BY 5
),
bottom_5 as
(
  SELECT *
  FROM profit
  WHERE profit.profit != 0
  GROUP BY 1,2,3,4,5
  ORDER BY 5
  LIMIT 5
),
top_5 as
(
  select *
  from profit
  group by 1,2,3,4,5
  order by 5 desc
  limit 5
```

Q4 SQL Sntax : Here

```
)
select
  top_5.prduct_id AS Product_Id
  , top_5.product_name AS Product_Name
  , top_5.retail_price AS Retail_Price
  , top_5.product_cost AS Cost
  , top_5.profit AS Profit
  , CASE
    WHEN RANK() OVER (order by top_5.profit DESC)= 1 THEN 'Rank 1 Top Profitable'
    WHEN RANK() OVER (order by top_5.profit DESC)= 2 THEN 'Rank 2 Top Profitable'
    WHEN RANK() OVER (order by top_5.profit DESC)= 3 THEN 'Rank 3 Top Profitable'
    WHEN RANK() OVER (order by top_5.profit DESC)= 4 THEN 'Rank 4 Top Profitable'
    WHEN RANK() OVER (order by top_5.profit DESC)= 5 THEN 'Rank 5 Top Profitable'
    END AS Profit_Rank_Alltime
from top_5
union all
select
  bottom_5.prduct_id AS Product_Id
  , bottom_5.product_name AS Product_Name
  , bottom_5.retail_price AS Retail_Price
  , bottom_5.product_cost AS Cost
  , bottom_5.profit AS Profit
  , CASE
    WHEN RANK() OVER (ORDER BY bottom_5.profit)= 1 THEN 'Rank 1 Least Profitable'
    WHEN RANK() OVER (ORDER BY bottom_5.profit)= 2 THEN 'Rank 2 Least Profitable'
    WHEN RANK() OVER (ORDER BY bottom_5.profit)= 3 THEN 'Rank 3 Least Profitable'
    WHEN RANK() OVER (ORDER BY bottom_5.profit)= 4 THEN 'Rank 4 Least Profitable'
    WHEN RANK() OVER (ORDER BY bottom_5.profit)= 5 THEN 'Rank 5 Least Profitable'
    END AS Profit_Rank_Alltime
from bottom_5
ORDER BY 5 DESC
```

05

Create a query to get **the number of unique users**, **number of orders**, and total sale price per status and month

From Jan 2019 until Apr 2022

Schema & Preview Q5

Question_5 QUERY SHARE COPY

SCHEMA DETAILS PREVIEW

Filter Enter property name or value

Field name	Type	Mode	Collatio
Month_Date	DATE	NULLABLE	
Category	STRING	NULLABLE	
Profit_Category_Month	FLOAT	NULLABLE	

Question_5 QUERY SHARE COPY

SCHEMA DETAILS PREVIEW

Row	Month_Date	Category	Profit_Categ...
1	2022-06-01	Plus	1892.43312...
2	2022-06-02	Plus	3742.38684...
3	2022-06-03	Plus	6200.40726...
4	2022-06-04	Plus	8591.87105...
5	2022-06-05	Plus	10759.4274...
6	2022-06-06	Plus	12403.6230...
7	2022-06-07	Plus	14192.8651...
8	2022-06-08	Plus	16306.0258...
9	2022-06-09	Plus	19211.2903...
10	2022-06-10	Plus	21790.0837...
11	2022-06-11	Plus	24219.7502...
12	2022-06-12	Plus	27135.7268...

Q5 SQL Sntax : Here

```sql
WITH
step_1 AS
(
  SELECT
    category
    , retail_price
    , cost
  FROM `bigquery-public-data.thelook_ecommerce.products`
  group by 1,2,3
),
step_2 AS
(
  SELECT
  DATE(DATE_TRUNC(created_at,DAY)) AS Month_Date
  , sale_price
  FROM `bigquery-public-data.thelook_ecommerce.order_items`
  WHERE status = 'Complete' AND DATE(DATE_TRUNC(created_at,Month)) BETWEEN '2022-06-01' AND '2022-08-31'
),
obs_1 as
(
SELECT
step_2.Month_Date
, step_1.category AS Category
, step_1.cost AS Cost
, step_2.sale_price AS Sale_Price
FROM step_2
LEFT JOIN step_1
ON step_1.retail_price= step_2.sale_price
),
```

```sql
main_1 AS
(
  SELECT
  obs_1.Month_Date
  , obs_1.Category
  , obs_1.Cost
  , obs_1.Sale_Price
  , SUM(obs_1.Sale_Price) OVER (PARTITION BY obs_1.Category ORDER BY obs_1.Month_Date) AS Total_Sale
  , SUM(obs_1.Cost) OVER (PARTITION BY obs_1.Category ORDER BY obs_1.Month_Date) AS Total_Cost
  FROM obs_1
  WHERE obs_1.Month_Date BETWEEN '2022-06-01' AND '2022-06-15'
  GROUP BY 1,2,3,4
),
main_2 AS
(
  SELECT
  obs_1.Month_Date
  , obs_1.Category
  , obs_1.Cost
  , obs_1.Sale_Price
  , SUM(obs_1.Sale_Price) OVER (PARTITION BY obs_1.Category ORDER BY obs_1.Month_Date) AS Total_Sale
  , SUM(obs_1.Cost) OVER (PARTITION BY obs_1.Category ORDER BY obs_1.Month_Date) AS Total_Cost
  FROM obs_1
  WHERE obs_1.Month_Date BETWEEN '2022-07-01' AND '2022-07-15'
  GROUP BY 1,2,3,4
),
```

Q5 SQL Sntax : Here

```
main_3 AS
(
  SELECT
  obs_1.Month_Date
  , obs_1.Category
  , obs_1.Cost
  , obs_1.Sale_Price
  , SUM(obs_1.Sale_Price) OVER (PARTITION BY obs_1.Category ORDER BY obs_1.Month_Date) AS Total_Sale
  , SUM(obs_1.Cost) OVER (PARTITION BY obs_1.Category ORDER BY obs_1.Month_Date) AS Total_Cost
  FROM obs_1
  WHERE obs_1.Month_Date BETWEEN '2022-08-01' AND '2022-08-15'
  GROUP BY 1,2,3,4
)
```

```
SELECT
 main_1.Month_Date
 ,main_1.Category
 ,main_1.Total_Sale - main_1.Total_Cost AS Profit_Category_Month
FROM main_1
UNION DISTINCT
SELECT
main_2.Month_Date
, main_2.Category
, main_2.Total_Sale - main_2.Total_Cost AS Profit_Category_Month
FROM main_2
UNION DISTINCT
SELECT
main_3.Month_Date
, main_3.Category
, main_3.Total_Sale - main_3.Total_Cost AS Profit_Category_Month
FROM main_3
order by 1
```

THANK YOU