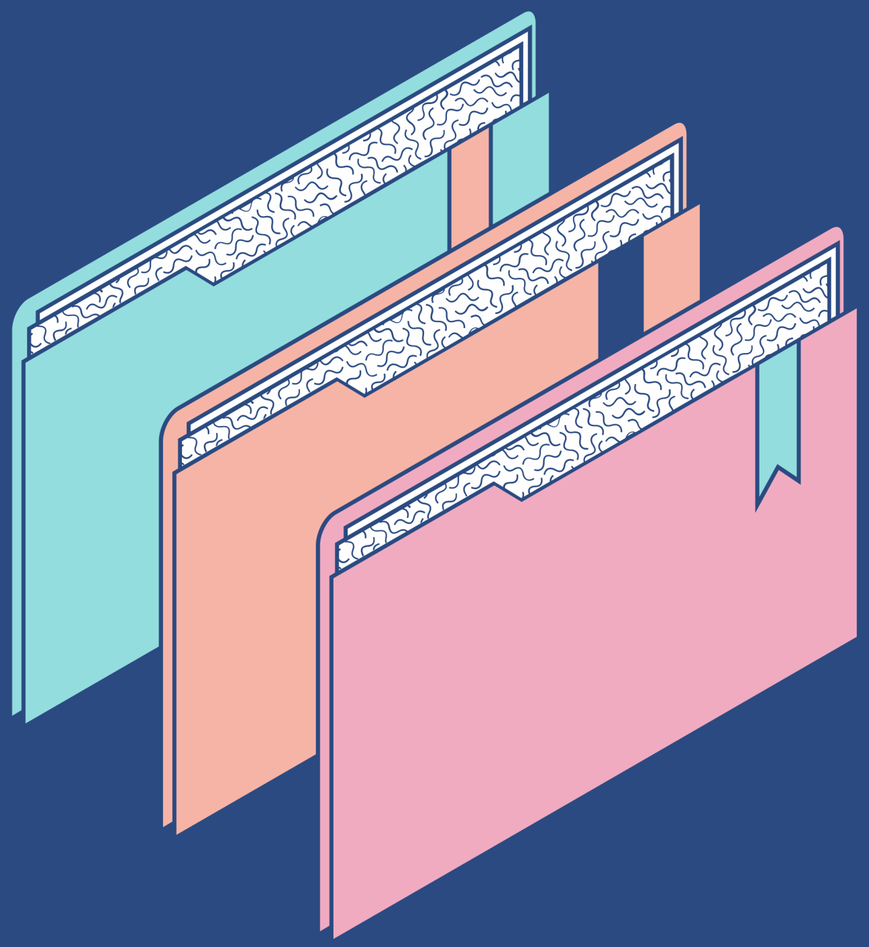
Advanced Assignment SQL

By: Biyan Bahtiar Ramadhan
Team 5, Section Hamburg



Background

As a Data Analyst at eCommerce clothing site company, TheLook, I've been given questions by my user as follow:

1. Find the categories with the lowest business growth (revenue and profit) in the past 1 year.
2. Understand the current retention performance and push new initiative to boost retention rate.

Categories With The Lowest Profit and Revenue Growth

IN THE PAST 1 YEAR

What are the categories with the lowest revenue growth?

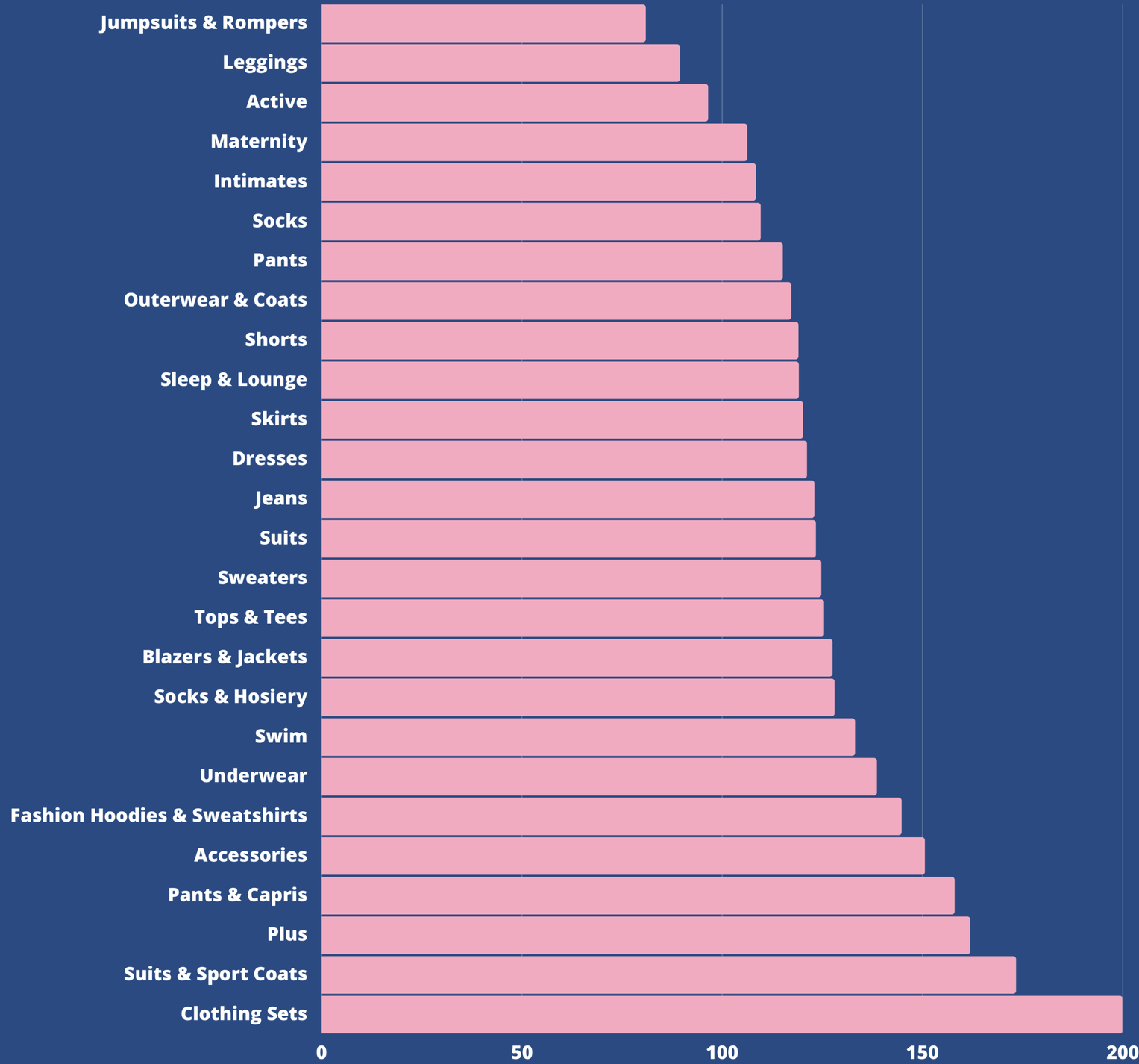
What are the categories with the lowest proft growth?

What categories need to be deprioritized?

What are the categories with the lowest revenue growth?

IN THE PAST 1 YEAR

Jumpsuits & Rompers, Active and Leggings are the 3 categories with the lowest revenue growth



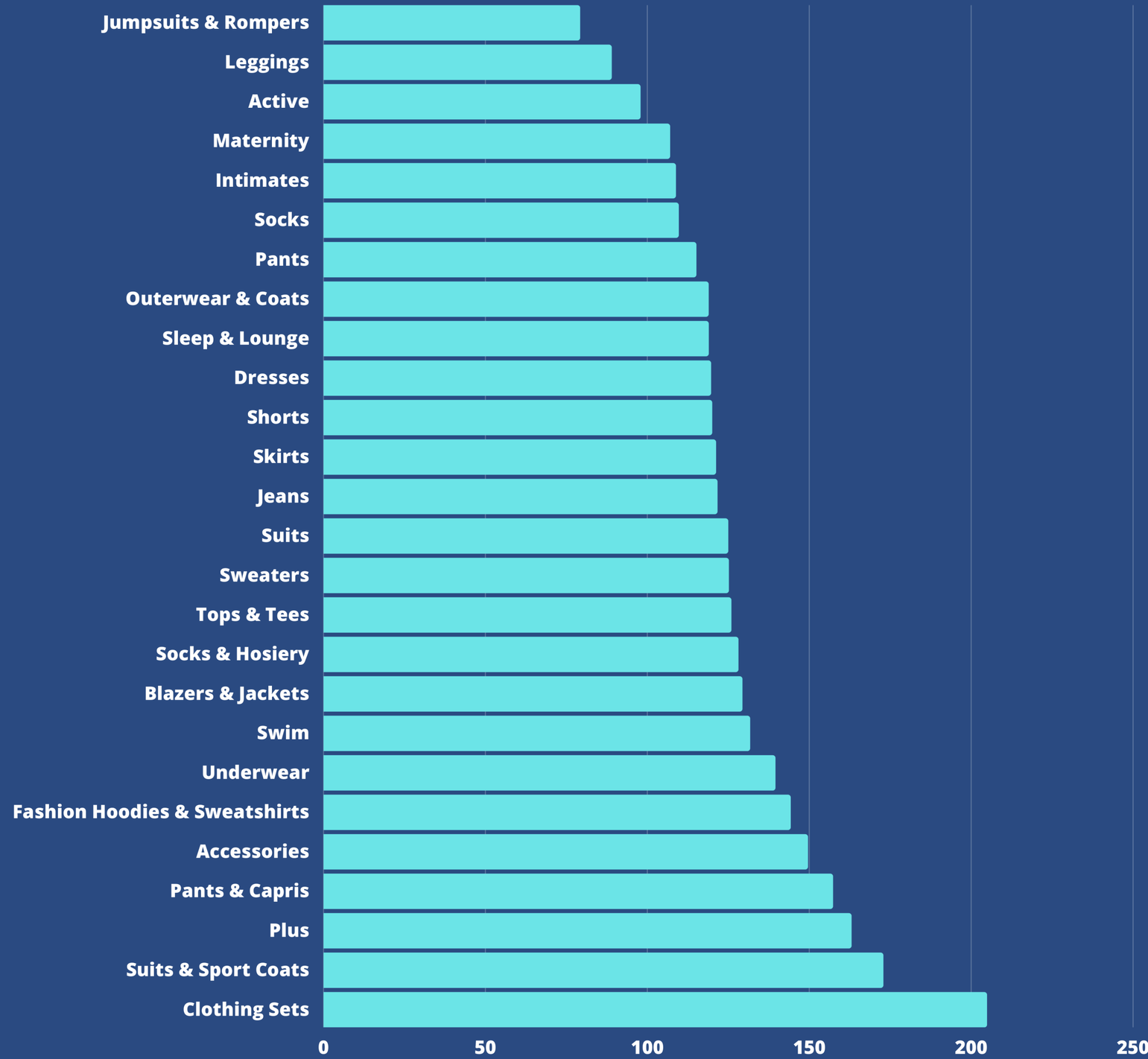
```
WITH SalesPerCategory AS(
  SELECT
    p.category
    , DATE_TRUNC(DATE(o.created_at), YEAR) AS year
    , SUM(oi.sale_price) AS sum_sales
  FROM
    sql-project-376612.thelook_ecommerce.orders o
  INNER JOIN
    sql-project-376612.thelook_ecommerce.order_items oi
    ON
    o.order_id = oi.order_id
  INNER JOIN
    sql-project-376612.thelook_ecommerce.products p
  ON
    oi.product_id = p.id
  WHERE
    oi.status = 'Complete'
  GROUP BY
    1,2
  ORDER BY
    1,2
)
```

```
, SumSalesTable AS(
  SELECT
    category
    , year
    , sum_sales AS SumSales
     , LAG(sum_sales) OVER(PARTITION BY category ORDER BY year) AS
PastSumSales
  FROM
    SalesPerCategory
  WHERE
    year BETWEEN '2021-01-01' AND '2022-01-01'
  ORDER BY
    1,2 ASC
)
SELECT
  category AS Category
      ,   (SumSales   -   PastSumSales)/PastSumSales   *   100   AS
RevenueGrowthRate
FROM
  SumSalesTable
WHERE
  PastSumSales IS NOT NULL
ORDER BY
  2 ASC;
```

What are the categories with the lowest profit growth?

IN THE PAST 1 YEAR

Jumpsuits & Rompers, Leggings and Active are the 3 categories with the lowest profit growth



```
WITH SalesPerCategory AS(
  SELECT
    p.category
    , DATE_TRUNC(DATE(o.created_at), YEAR) AS year
    , SUM(oi.sale_price) AS sum_sales
  FROM
    sql-project-376612.thelook_ecommerce.orders o
  INNER JOIN
    sql-project-376612.thelook_ecommerce.order_items oi
    ON
    o.order_id = oi.order_id
  INNER JOIN
    sql-project-376612.thelook_ecommerce.products p
  ON
    oi.product_id = p.id
  WHERE
    DATE_TRUNC(DATE(o.created_at), YEAR) BETWEEN '2021-01-01' AND
'2022-01-01'
    AND
    oi.status = 'Complete'
  GROUP BY
    1,2
  ORDER BY
    1,2
)
```

```
, CostReference AS(
 SELECT
 DISTINCT(product_id) AS product_id
 , cost
FROM
    sql-project-376612.thelook_ecommerce.inventory_items
)
, CostPerCategory AS(
  SELECT
    p.category AS category
    , DATE_TRUNC(o.created_at, YEAR) AS year
    , SUM(cr.cost) as sum_cost
  FROM
    sql-project-376612.thelook_ecommerce.orders o
  INNER JOIN
    sql-project-376612.thelook_ecommerce.order_items oi
    ON
    o.order_id = oi.order_id
  INNER JOIN
    sql-project-376612.thelook_ecommerce.products p
  ON
    oi.product_id = p.id
  INNER JOIN
    CostReference cr
  ON
    oi.product_id = cr.product_id
```

```
 WHERE
   DATE_TRUNC(DATE(o.created_at), YEAR) BETWEEN '2021-01-01' AND
2022-01-01'
   AND
   oi.status = 'Complete'
 GROUP BY
   1,2
 ORDER BY
   1,2

 CostSalesJoin AS(
 SELECT
   s.category
   , s.year
   , s.sum_sales
   , c.sum_cost
   , (s.sum_sales - c.sum_cost) AS profit
 FROM
   SalesPerCategory s
 INNER JOIN
   CostPerCategory c
 ON
   s.category = c.category
   AND
   DATE(s.year) = DATE(c.year)
 ORDER BY
 1,2)
```

```
, CurrentPastProfit AS (
  SELECT
  category
  , year
  , profit
  , LAG(profit) OVER(PARTITION BY category ORDER BY year) AS
pastyr_profit
FROM
  CostSalesJoin
)
SELECT
  category AS Category
  , (profit-pastyr_profit)/pastyr_profit*100 AS ProfitGrowthRate
FROM
  CurrentPastProfit
WHERE
  pastyr_profit IS NOT NULL
  AND
  year = '2022-01-01'
ORDER BY
  2 ASC;
```

What categories need to be deprioritized?

Jumpsuits & Rompers and Leggings need to be deprioritized.

		Market Share	
		HIGH	LOW
Profit Growth	HIGH	Accessories	Blazers & Jackets
		Fashion Hoodies & Sweatshirts	Clothing Sets
		Suits & Sport Coats	Pants & Capris
		Sweaters	Plus
		Swim	Socks & Hosiery
		Tops & Tees	Suits
			Underwear
	LOW	Active	Jumpsuits & Rompers
		Dresses	Leggings
		Intimates	Maternity
		Jeans	Pants
		Outerwear & Coats	Skirts
		Shorts	Socks
		Sleep & Lounge	

To obtain market share..

```
WITH SalesPerCategory AS(
  SELECT
    p.category
    , DATE_TRUNC(DATE(o.created_at), YEAR) AS year
    , SUM(oi.sale_price) AS sum_sales
  FROM
    sql-project-376612.thelook_ecommerce.orders o
  INNER JOIN
    sql-project-376612.thelook_ecommerce.order_items oi
    ON
    o.order_id = oi.order_id
  INNER JOIN
    sql-project-376612.thelook_ecommerce.products p
  ON
    oi.product_id = p.id
  WHERE
    DATE_TRUNC(DATE(o.created_at), YEAR) BETWEEN '2021-01-01' AND
'2022-01-01'
    AND
    oi.status = 'Complete'
  GROUP BY
    1,2
  ORDER BY
    1,2
)
, SalesPerCategory2Year AS (
  SELECT
    category
    , SUM(sum_sales) AS sum_sales
  FROM
    SalesPerCategory
  GROUP BY
    1
)
, CostReference AS(
  SELECT
    DISTINCT(product_id) AS product_id
    , cost
  FROM
    sql-project-376612.thelook_ecommerce.inventory_items
)
, CostPerCategory AS(
  SELECT
    p.category AS category
    , DATE_TRUNC(o.created_at, YEAR) AS year
    , SUM(cr.cost) as sum_cost
  FROM
    sql-project-376612.thelook_ecommerce.orders o
```

To obtain market share..

```
 INNER JOIN
 sql-project-376612.thelook_ecommerce.order_items oi
 ON
 o.order_id = oi.order_id
 INNER JOIN
 sql-project-376612.thelook_ecommerce.products p
 ON
 oi.product_id = p.id
 INNER JOIN
 CostReference cr
 ON
 oi.product_id = cr.product_id
 WHERE
 DATE_TRUNC(DATE(o.created_at), YEAR) BETWEEN '2021-01-01' AND
'2022-01-01'
 AND
 oi.status = 'Complete'
 GROUP BY
 1,2
 ORDER BY
 1,2
)
```

```
, CostSalesJoin AS(
 SELECT
 s.category
 , s.year
 , s.sum_sales
 , c.sum_cost
 , (s.sum_sales - c.sum_cost) AS profit
 FROM
 SalesPerCategory s
 INNER JOIN
 CostPerCategory c
 ON
 s.category = c.category
 AND
 DATE(s.year) = DATE(c.year)
 ORDER BY
 1,2
)
, CurrentPastProfit AS (
 SELECT
 category
 , year
 , profit
 , LAG(profit) OVER(PARTITION BY category ORDER BY year) AS
pastyr_profit
```

To obtain market share..

```
FROM
 CostSalesJoin
)
, profit_growth_rate as(
 SELECT
 category AS Category
 , (profit-pastyr_profit)/pastyr_profit*100 AS ProfitGrowthRate
 FROM
 CurrentPastProfit
 WHERE
 pastyr_profit IS NOT NULL
 AND
 year = '2022-01-01'
 ORDER BY
 2 ASC
)
, SumSalesAll AS(
 SELECT
 SUM(oi.sale_price) AS sum_sales_all
 FROM
 sql-project-376612.thelook_ecommerce.orders o
 INNER JOIN
 sql-project-376612.thelook_ecommerce.order_items oi
 ON
 o.order_id = oi.order_id
 WHERE
 DATE_TRUNC(DATE(o.created_at), YEAR) BETWEEN '2021-01-01' AND
'2022-01-01'
 AND
 oi.status = 'Complete'
)
SELECT
 sy.category
 , ProfitGrowthRate
 , (sy.sum_sales/sa.sum_sales_all)*100 AS MarketShare
FROM
 SalesPerCategory2Year sy
INNER JOIN
 profit_growth_rate pg
 ON sy.category = pg.Category
CROSS JOIN
 SumSalesAll sa
ORDER BY
 2;
```

Current Retention Performance and Push New Initiative to Boost Retention Rate.

IN THE PAST 1 YEAR,
PER MONTH

How is the retention rate of the businesses

What categories has the worst retention rate? Why?

What categories has the best retention rate? Why?

Is there any new initiatives to boost overall retention rate?

Retention Rate of The Businesses

MONTHLY, ON YEAR 2022

- On average, 1,53% users purchase again on the following month after making their first purchase.
- Users who first purchase at November 2022 do the most purchase in percentage in the following month.

FirstOrder	0	1	2	3	4	5	6	7	8	9	10	11
Jan-22	100	0.92	1.73	1.15	0.92	1.27	1.27	1.04	1.27	1.15	1.04	1.15
Feb-22	100	1.37	1.50	0.87	0.62	1.75	1.12	1.12	0.87	1.62	1.12	
Mar-22	100	1.39	1.79	1.00	1.00	1.39	1.20	1.79	2.19	1.49		
Apr-22	100	0.90	1.30	2.01	1.50	1.50	0.90	1.30	1.60			
May-22	100	1.26	2.03	1.26	1.55	1.65	1.94	1.36				
Jun-22	100	1.56	1.75	0.92	1.19	2.02	2.02					
Jul-22	100	1.46	1.46	2.06	1.97	1.72						
Aug-22	100	2.17	1.78	2.55	2.01							
Sep-22	100	2.07	2.21	1.71								
Oct-22	100	2.69	2.43									
Nov-22	100	2.98										
Dec-22	100											

```sql
with first_order_ref as (
  SELECT
    user_id
    , min(date_trunc(date(created_at), month)) as
first_order_month
  FROM sql-project-376612.thelook_ecommerce.orders
  where date_trunc(date(created_at), month) BETWEEN '2022-01-01'
AND '2022-12-01'
    and status = 'Complete'
    and user_id is not null
  group by 1
)
, act_ref as (
  select
    distinct o.user_id
    , date_trunc(date(o.created_at), month) as period
    , first_order_month
  FROM sql-project-376612.thelook_ecommerce.orders as o
  left join first_order_ref as fst
    on o.user_id = fst.user_id
  where o.user_id is not null and date_trunc(date(o.created_at),
month) BETWEEN '2022-01-01' AND '2022-12-01'
    and status = 'Complete'
)
, raw as (
 select
 *
 , date_diff(period, first_order_month, month) datediff
 from act_ref
)
, cohort_size as (
  select
    first_order_month
    , count(distinct user_id) first_users
  from first_order_ref
  group by 1
)
, cohort as (
  select
    first_order_month
    , datediff
    , count(distinct user_id) users
  from raw
  group by 1,2
)
select
  c.*
  , first_users
  , (users/first_users)*100 cohort_percentage
from cohort as c
left join cohort_size as cs
  on c.first_order_month = cs.first_order_month
order by 1,2;
```

Retention Rate of The Businesses

YEARLY

- On average, 8.35% users purchase again on the following year after making their first purchase.
- Users who first purchase at 2021 do the most purchase in percentage in the following year.
- Compared to monthly, more users are coming back to purchase clothings in the app in year period. However, the percentage is still small.

first_order_year	0	1	2	3
2019	100	6.53	6.33	6.73
2020	100	8.31	7.80	
2021	100	10.46		
2022	100			

```
with first_order_ref as (
  SELECT
    user_id
    , min(date_trunc(date(created_at), YEAR)) as first_order_year
  FROM sql-project-376612.thelook_ecommerce.orders
  where date_trunc(date(created_at), YEAR) BETWEEN '2019-01-01'
AND '2022-12-01'
    and status = 'Complete'
    and user_id is not null
  group by 1
)
, act_ref as (
  select
    distinct o.user_id
    , date_trunc(date(o.created_at), YEAR) as period
    , first_order_year
  FROM sql-project-376612.thelook_ecommerce.orders as o
  left join first_order_ref as fst
    on o.user_id = fst.user_id
  where o.user_id is not null and date_trunc(date(o.created_at),
month) BETWEEN '2019-01-01' AND '2022-12-01'
    and status = 'Complete'
)
, raw as (
 select
 *
 , date_diff(period, first_order_year, year) datediff
 from act_ref
)
, cohort_size as (
 select
 first_order_year
 , count(distinct user_id) first_users
 from first_order_ref
 group by 1
)
, cohort as (
 select
 first_order_year
 , datediff
 , count(distinct user_id) users
 from raw
 group by 1,2
)
select
 c.*
 , first_users
 , (users/first_users)*100 cohort_percentage
from cohort as c
left join cohort_size as cs
 on c.first_order_year = cs.first_order_year
order by 1,2;
```

User Retention Rate Per Category by December 2022

(SEGMENTED BASED ON USER FIRST ORDER MONTH)

category	Jan22	Feb22	Mar22	Apr22	May22	Jun22	Jul22	Aug22	Sep22	Oct22	Nov22	avg
Tops & Tees	0.00	0.00	1.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.11	0.21
Socks & Hosiery	0.00	0.00	0.00	0.00	0.00	2.56	0.00	0.00	0.00	0.00	0.00	0.23
Sleep & Lounge	1.41	0.00	0.00	1.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.25
Socks	0.00	0.00	2.17	0.00	0.00	0.00	0.00	1.35	0.00	1.23	0.00	0.43
Intimates	0.00	2.56	0.00	0.00	0.95	1.75	0.74	0.00	0.00	0.00	0.00	0.55

Insight

- This table shows 5 category with the highest retention rate. For each category and user's first order month, how many of them repeat purchase in the same category in December 2022.
- If we look from monthly retention rate, Intimates, Socks, Sleep & Lounge, Socks & Hosiery and Tops & Tees are the category with highest retention rate, monthly.

Top User Retention Rate Per Category by Year 2022

(SEGMENTED BASED ON USER FIRST ORDER YEAR)

category	2019	2020	2021	average
Jeans	0.86	1.19	1.47	1.17
Sweaters	2.82	0.00	0.93	1.25
Intimates	1.09	0.29	2.48	1.28
Pants	1.72	1.12	1.64	1.49
Leggings	4.35	1.01	0.51	1.96

Insight

- This table shows 5 category with the highest retention rate, for each category and user's first order year, how many of them repeat purchase in the same category in year 2022.
- If we look from yearly retention rate, Jeans, Sweaters, Intimates, Pants and Leggings are the categories with highest retention rate.
- The top categories are different from monthly retention rate, indicating that some categories are repurchased after long period of time, while other categories don't.

```
with first_order_ref as (
  SELECT
    o.user_id
    , p.category
    , MIN(EXTRACT(YEAR FROM DATE(o.created_at))) as first_order_year
  FROM sql-project-376612.thelook_ecommerce.orders o
  INNER JOIN
    sql-project-376612.thelook_ecommerce.order_items oi
    on o.order_id = oi.order_id
  INNER JOIN
    sql-project-376612.thelook_ecommerce.products p
    on oi.product_id = p.id
  WHERE DATE_TRUNC(DATE(o.created_at), YEAR) BETWEEN '2019-01-01' AND
'2022-01-01'
    and o.status = 'Complete'
    and o.user_id is not null
  GROUP BY 1,2
)
, act_ref_raw as (
  SELECT
    o.user_id
    , p.category
    , first_order_year
    , EXTRACT(YEAR FROM DATE(o.created_at)) as period
    , ROW_NUMBER() OVER(
      PARTITION BY
        o.user_id
        , p.category
        , first_order_year
        , EXTRACT(YEAR FROM DATE(o.created_at))
      ORDER BY o.user_id) as dup_count
  FROM sql-project-376612.thelook_ecommerce.orders o
  INNER JOIN
    sql-project-376612.thelook_ecommerce.order_items oi
    on o.order_id = oi.order_id
  INNER JOIN
    sql-project-376612.thelook_ecommerce.products p
    on oi.product_id = p.id
  INNER JOIN
    first_order_ref fst
    ON fst.user_id = o.user_id
    AND fst.category = p.category
  WHERE DATE_TRUNC(DATE(o.created_at), YEAR) BETWEEN '2000-01-01' AND
'2022-01-01'
    and o.status = 'Complete'
    and o.user_id is not null
)
, act_ref as(
  SELECT
    user_id
    , category
    , first_order_year
    , period
  FROM
    act_ref_raw
  WHERE
    dup_count = 1
    and first_order_year <= period
)
```

```
, category_to_fill AS(
 SELECT
 DISTINCT category AS category
 FROM sql-project-376612.thelook_ecommerce.products
)
, fst_to_fill AS(
 SELECT
 DISTINCT EXTRACT(YEAR FROM DATE(created_at)) as first_order_year
 FROM sql-project-376612.thelook_ecommerce.orders
 WHERE DATE_TRUNC(DATE(created_at), YEAR) BETWEEN '2000-01-01' AND
'2022-01-01'
)
, period_to_fill AS(
 SELECT
 DISTINCT EXTRACT(YEAR FROM DATE(created_at)) as period
 FROM sql-project-376612.thelook_ecommerce.orders
 WHERE DATE_TRUNC(DATE(created_at), YEAR) BETWEEN '2000-01-01' AND
'2022-12-01'
)
, table_to_fill AS(
 SELECT
 category
 , first_order_year
 , period
 FROM category_to_fill
 CROSS JOIN fst_to_fill
 CROSS JOIN period_to_fill
 WHERE first_order_year <= period
)
```

```
, ref_first as(
 SELECT
 category
 , first_order_year
 , period
 , COUNT(user_id) AS count_first
 FROM act_ref
 WHERE
 first_order_year = period
 GROUP BY
 1,2,3
)
, ref_year as (
 SELECT
 category
 , first_order_year
 , period
 , COUNT(user_id) AS count_period
 FROM act_ref ar
 WHERE first_order_year <= period
 GROUP BY
 1,2,3
)
```

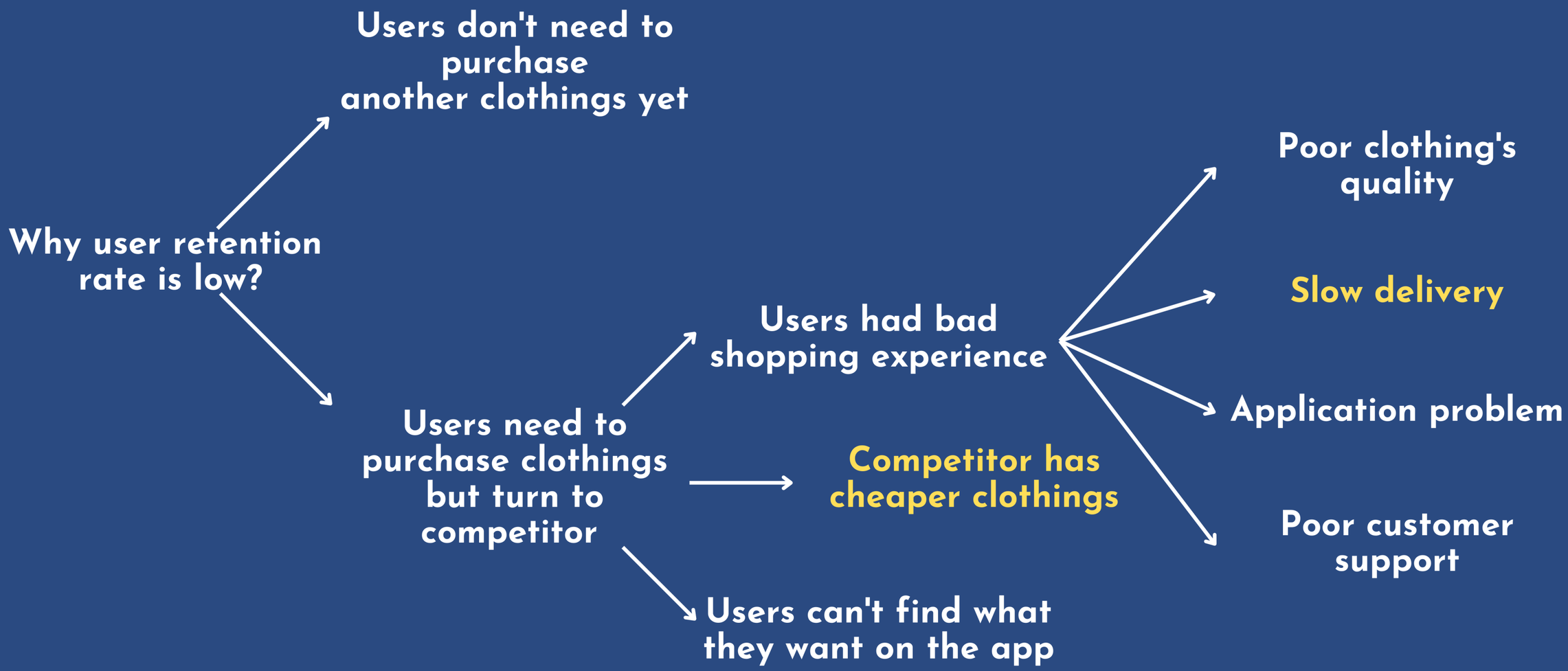
```
, table_1 AS (
 select
 tf.category
 , tf.first_order_year
 , tf.period
 , count_first
from
 table_to_fill tf
LEFT JOIN
 ref_first rf
 ON tf.category = rf.category
 AND tf.first_order_year = rf.first_order_year
)
SELECT
 t1.category
 , t1.first_order_year
 , t1.period
 , COALESCE(count_period,0) AS count_period
 , t1.count_first
 , (COALESCE(count_period,0)/t1.count_first)*100 AS retention_rate
FROM
 table_1 t1
LEFT JOIN
 ref_year ry
 on t1.category = ry.category
 and t1.first_order_year = ry.first_order_year
 and t1.period = ry.period
WHERE
 t1.period = 2022
 and t1.first_order_year < t1.period
ORDER BY 1,2,3
```

User Retention Rate is Low!

Problem Statement:

How to increase user retention rate to 15% by the end of the year?

Root Cause Analysis



Our price is exactly same with retail price

However, it is still not known whether our competitor give discounts.

category	avg_sale_price	avg_retail_price
Socks & Hosiery	16.79	16.79
Socks	20.44	20.44
Underwear	27.03	27.03
Leggings	27.24	27.24
Intimates	33.44	33.44
Plus	38.19	38.19
Tops & Tees	41.48	41.48
Accessories	42.77	42.77
Jumpsuits & Rompers	44.51	44.51
Shorts	46.61	46.61
Sleep & Lounge	49.38	49.38
Active	49.85	49.85
Maternity	51.09	51.09
Pants & Capris	54.10	54.10
Fashion Hoodies & Sweatshirts	54.58	54.58
Skirts	55.20	55.20
Swim	56.72	56.72
Pants	58.98	58.98
Sweaters	76.41	76.41
Dresses	84.39	84.39
Clothing Sets	84.52	84.52
Blazers & Jackets	90.71	90.71
Jeans	97.82	97.82
Suits	117.00	117.00
Suits & Sport Coats	126.74	126.74
Outerwear & Coats	146.70	146.70

```
WITH ref_table AS
  (SELECT
    oi.product_id
    , p.category
    , oi.sale_price
    , p.retail_price
  FROM sql-project-376612.thelook_ecommerce.order_items oi
  INNER JOIN sql-project-376612.thelook_ecommerce.products p
    ON oi.product_id = p.id)
, avg_sale AS(
  SELECT
    category
    , AVG(sale_price) AS avg_sale_price
  FROM
    ref_table
  GROUP BY
    1
)
, avg_retail AS(
  SELECT
    category
    , AVG(retail_price) AS avg_retail_price
  FROM
    ref_table
  GROUP BY
    1
)
```

```
SELECT
  avg_sale.category
  , avg_sale_price
  , avg_retail_price
FROM
  avg_sale
INNER JOIN
  avg_retail
  ON avg_sale.category=avg_retail.category
ORDER BY
  1
```

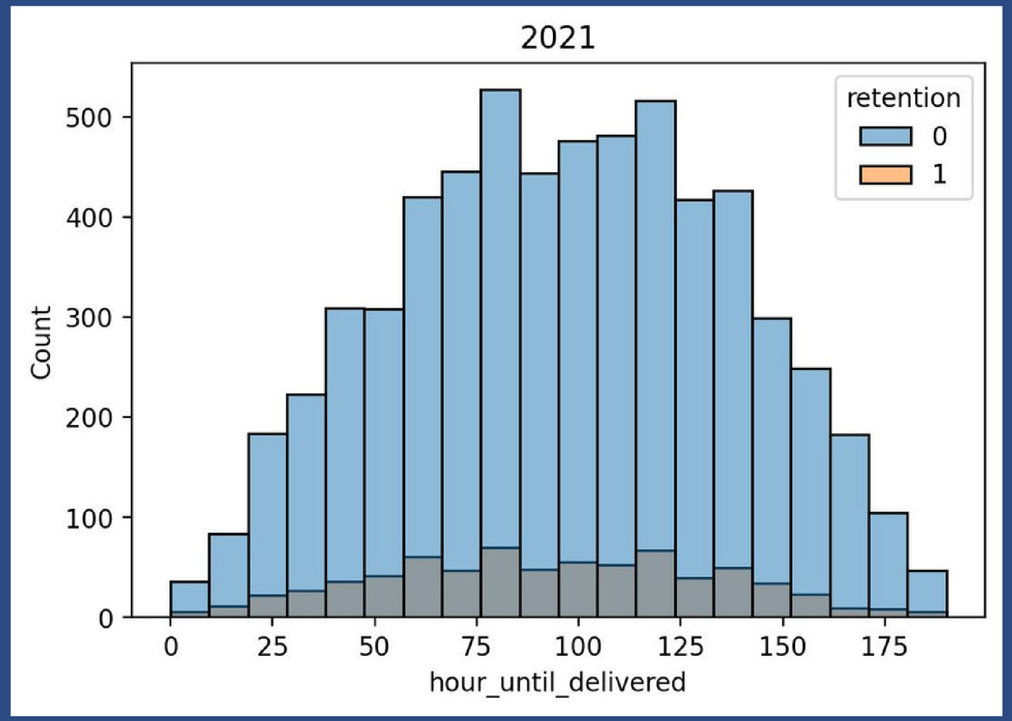
Average time needed from order created to delivered is 96.8 hrs

Clothing sets have the slowest delivery time. This category is also among the lowest retention rate. Coincidence?

category	created_to_shipped	shipped_to_delivered	created_to_delivered
Maternity	35.31	58.41	93.71
Blazers & Jackets	36.08	57.92	94.00
Tops & Tees	35.53	59.18	94.71
Pants & Capris	35.03	59.77	94.80
Pants	35.65	59.21	94.86
Shorts	35.49	59.82	95.31
Jeans	36.05	59.26	95.31
Plus	36.31	59.16	95.46
Skirts	36.15	59.43	95.58
Socks	35.69	60.32	96.01
Intimates	35.81	60.20	96.01
Swim	35.79	60.34	96.12
Accessories	36.03	60.35	96.38
Underwear	35.24	61.24	96.47
Leggings	35.36	61.30	96.67
Fashion Hoodies & Sweatshirts	36.56	60.19	96.75
Outerwear & Coats	36.11	60.66	96.77
Sweaters	36.29	60.50	96.79
Dresses	37.33	59.87	97.19
Sleep & Lounge	35.97	61.54	97.51
Active	36.45	61.07	97.51
Jumpsuits & Rompers	36.76	60.90	97.66
Suits & Sport Coats	37.19	61.07	98.27
Socks & Hosiery	36.60	62.20	98.79
Suits	37.91	61.64	99.55
Clothing Sets	40.68	68.77	109.46

```
WITH table_1 AS(
  SELECT
    o.order_id
    , oi.product_id
    , DATETIME_DIFF(DATETIME(o.shipped_at), DATETIME(o.created_at),
HOUR) AS created_to_shipped
    , DATETIME_DIFF(DATETIME(o.delivered_at), DATETIME(o.shipped_at),
HOUR) AS shipped_to_delivered
    , DATETIME_DIFF(DATETIME(o.delivered_at), DATETIME(o.created_at),
HOUR) AS created_to_delivered
  FROM
    sql-project-376612.thelook_ecommerce.orders o
  INNER JOIN
    sql-project-376612.thelook_ecommerce.order_items oi
    ON o.order_id = oi.order_id
  WHERE
    LOWER(o.status) = 'complete'
)
SELECT
  p.category
  , AVG(created_to_shipped) AS created_to_shipped
  , AVG(shipped_to_delivered) AS shipped_to_delivered
  , AVG(created_to_delivered) AS created_to_delivered
FROM
  table_1 t1
INNER JOIN
  sql-project-376612.thelook_ecommerce.products p
  ON t1.product_id = p.id
GROUP BY 1
```

T-test: Repeat Order Status vs. Hour Until Delivered (2021)




```
ttest2021 = pingouin.ttest(x=df2.loc[df2['retention']==1,'hour_until_delivered'],
              y=df2.loc[df2['retention']==0,'hour_until_delivered'],
              alternative='less')
print(df2.groupby('retention')['user_id'].agg('count'))
print('')
print(f"group 1 mean={df2.loc[df2['retention']==1,'hour_until_delivered'].mean()},\
 group 0 mean={df2.loc[df2['retention']==0,'hour_until_delivered'].mean( )}, p-value={ttest2021['p-val'].values}")
```

```
retention
0    6180
1     717
Name: user_id, dtype: int64

group 1 mean=92.79637377963738, group 0 mean=96.31440129449838, p-value=[0.01283327]
```

	1	0
Mean	92.80	96.31
Observations	717	6180
P-value (1-tail)	0.01283327	

Insight

- The t-test is done on people who first order in 2021. The group 1 is repeat order group, it constitutes of people who do at least 1 repeat order in year 2022. The time is measured in hour from order creation.
- Based on this test, there is evidence to support that in 2021, people who experiences faster delivery time tend to repeat order in the next year.

```
with first_order_ref as (
  SELECT
    user_id
    , order_id
    , min(date_trunc(date(created_at), year)) as first_order_year
  FROM sql-project-376612.thelook_ecommerce.orders
  where date_trunc(date(created_at), year) BETWEEN '2020-01-01' AND
'2020-12-01'
    and status = 'Complete'
    and user_id is not null
  group by 1,2
)
, act_ref as (
  select
    distinct o.user_id
    , fst.order_id
    , date_trunc(date(o.created_at), year) as period
    , first_order_year
  FROM sql-project-376612.thelook_ecommerce.orders as o
  left join first_order_ref as fst
    on o.user_id = fst.user_id
  where o.user_id is not null and date_trunc(date(o.created_at), year)
BETWEEN '2020-01-01' AND '2022-12-01'
    and status = 'Complete'
)
, raw as (
  select
    *
    , date_diff(period, first_order_year, year) datediff
  from act_ref
  WHERE first_order_year IS NOT NULL
)
, table1 as(
  SELECT
  r.user_id
  , r.order_id
  , first_order_year
  , period
  , CASE WHEN
      datediff = 0 THEN 0
      ELSE 1 END AS retention
  , DATETIME_DIFF(DATETIME(o.delivered_at), DATETIME(o.created_at), HOUR)
AS hour_until_delivered
FROM raw r
INNER JOIN sql-project-376612.thelook_ecommerce.orders as o
  ON r.order_id = o.order_id
)
, table2 as(
  SELECT
  user_id
  , order_id
  , first_order_year
  , period
  , hour_until_delivered
  , retention
  , ROW_NUMBER() OVER (PARTITION BY user_id ORDER BY retention DESC) as
row_num
FROM table1
ORDER BY user_id
)
SELECT
  user_id
  , order_id as first_order_id
  , hour_until_delivered
  , retention
FROM table2
WHERE row_num = 1
```

Conclusion & Recommendation

- Jumpsuits & Rompers and Leggings are 2 category that need to be deprioritized since they have low market share and low profit.
- One of the reason to low retention rate could be slow delivery time (measured from order creation).
- Need A/B testing to ascertain the cause and effect relationship.

Thank You