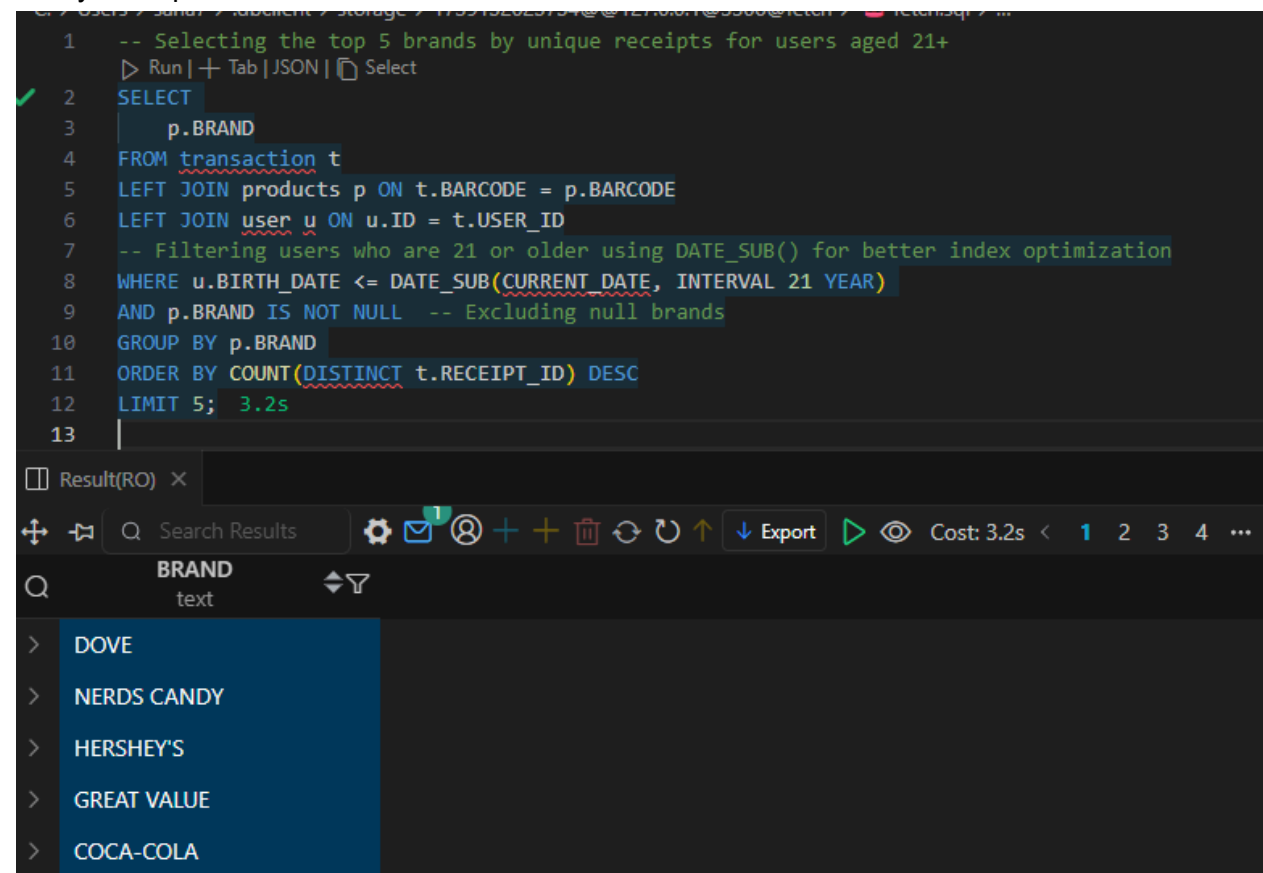


Section 2 - SQL Queries

Query for - What are the top 5 brands by receipts scanned among users 21 and over?

```
-- Selecting the top 5 brands by unique receipts for users aged 21+
SELECT
  p.BRAND
FROM transaction t
LEFT JOIN products p ON t.BARCODE = p.BARCODE
LEFT JOIN user u ON u.ID = t.USER_ID
-- Filtering users who are 21 or older using DATE_SUB() for better index optimization
WHERE u.BIRTH_DATE <= DATE_SUB(CURRENT_DATE, INTERVAL 21 YEAR)
AND p.BRAND IS NOT NULL -- Excluding null brands
GROUP BY p.BRAND
ORDER BY COUNT(DISTINCT t.RECEIPT_ID) DESC
LIMIT 5;
```

Query + Output



The screenshot shows a SQL IDE interface. The top pane displays the SQL query, which is identical to the one in the previous block. The bottom pane shows the results of the query. The results are displayed in a table with one column, 'BRAND', and five rows of data. The brands listed are DOVE, NERDS CANDY, HERSHEY'S, GREAT VALUE, and COCA-COLA. The interface includes a search bar, a filter icon, and a list of results.

```
1  -- Selecting the top 5 brands by unique receipts for users aged 21+
2  Run | + Tab | JSON | Select
3  SELECT
4    p.BRAND
5  FROM transaction t
6  LEFT JOIN products p ON t.BARCODE = p.BARCODE
7  LEFT JOIN user u ON u.ID = t.USER_ID
8  -- Filtering users who are 21 or older using DATE_SUB() for better index optimization
9  WHERE u.BIRTH_DATE <= DATE_SUB(CURRENT_DATE, INTERVAL 21 YEAR)
10 AND p.BRAND IS NOT NULL -- Excluding null brands
11 GROUP BY p.BRAND
12 ORDER BY COUNT(DISTINCT t.RECEIPT_ID) DESC
13 LIMIT 5; 3.2s
```

Result(RO) X

Search Results

BRAND

text

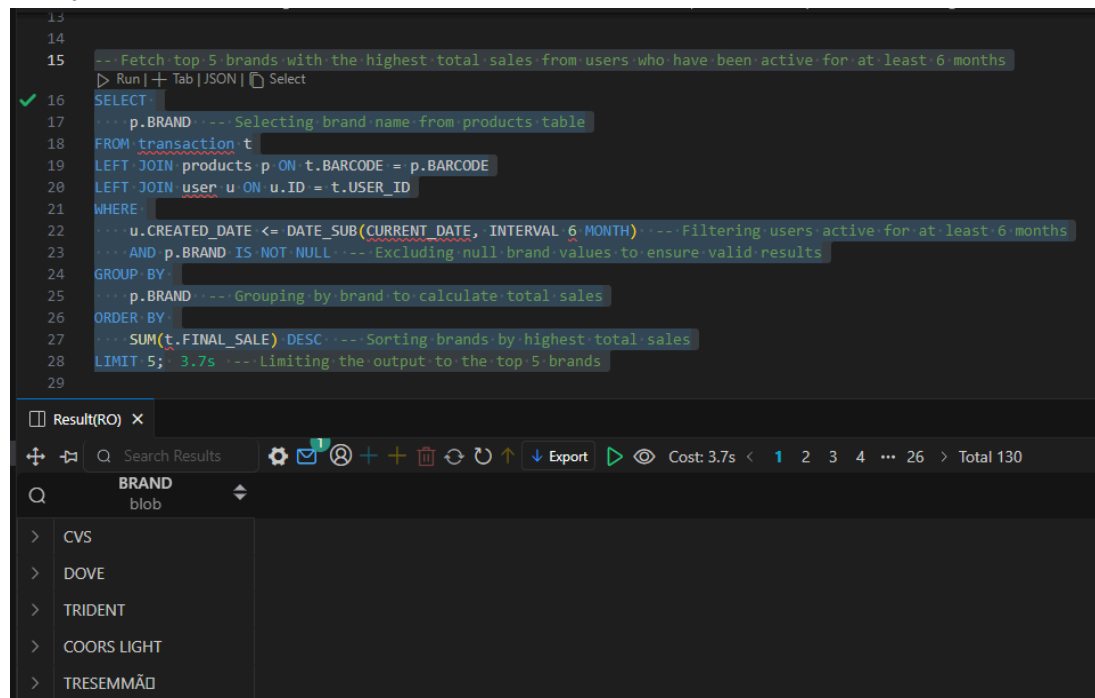
- > DOVE
- > NERDS CANDY
- > HERSHEY'S
- > GREAT VALUE
- > COCA-COLA

Cost: 3.2s < 1 2 3 4 ...

Query for - What are the top 5 brands by sales among users that have had their account for at least six months?

```
-- Fetch top 5 brands with the highest total sales from users who have been active for at least 6 months
SELECT
  p.BRAND -- Selecting brand name from products table
FROM transaction t
LEFT JOIN products p ON t.BARCODE = p.BARCODE
LEFT JOIN user u ON u.ID = t.USER_ID
WHERE
  u.CREATED_DATE <= DATE_SUB(CURRENT_DATE, INTERVAL 6 MONTH) -- Filtering users active for at
least 6 months
  AND p.BRAND IS NOT NULL -- Excluding null brand values to ensure valid results
GROUP BY
  p.BRAND -- Grouping by brand to calculate total sales
ORDER BY
  SUM(t.FINAL_SALE) DESC -- Sorting brands by highest total sales
LIMIT 5; -- Limiting the output to the top 5 brands
```

Query + Output



The screenshot shows a SQL IDE interface. The top pane displays a SQL query with line numbers 13 to 29. The query is the same as the one in the previous block. The bottom pane shows the results of the query, titled "Result(RO) X". The results are displayed in a table with a search bar and a list of brands. The brands listed are CVS, DOVE, TRIDENT, COORS LIGHT, and TRESEMMÃ. The table has a column header "BRAND" and a sub-header "blob". The results are sorted by sales, with CVS at the top and TRESEMMÃ at the bottom. The interface also shows a "Cost: 3.7s" and a "Total 130" count.

```
13
14
15 -- Fetch top 5 brands with the highest total sales from users who have been active for at least 6 months
16 SELECT
17   p.BRAND --- Selecting brand name from products table
18 FROM transaction t
19 LEFT JOIN products p ON t.BARCODE = p.BARCODE
20 LEFT JOIN user u ON u.ID = t.USER_ID
21 WHERE
22   u.CREATED_DATE <= DATE_SUB(CURRENT_DATE, INTERVAL 6 MONTH) --- Filtering users active for at least 6 months
23   AND p.BRAND IS NOT NULL --- Excluding null brand values to ensure valid results
24 GROUP BY
25   p.BRAND --- Grouping by brand to calculate total sales
26 ORDER BY
27   SUM(t.FINAL_SALE) DESC --- Sorting brands by highest total sales
28 LIMIT 5; 3.7s --- Limiting the output to the top 5 brands
29
```

Result(RO) X

BRAND
blob

- > CVS
- > DOVE
- > TRIDENT
- > COORS LIGHT
- > TRESEMMÃ

Cost: 3.7s < 1 2 3 4 ... 26 > Total 130

Question - Which is the leading brand in the Dips & Salsa category?

-- Assumption: "Leading" brand is defined as the brand with the highest total sales in the 'Dips & Salsa' category.

-- This means we are ranking brands based on their cumulative FINAL_SALE value

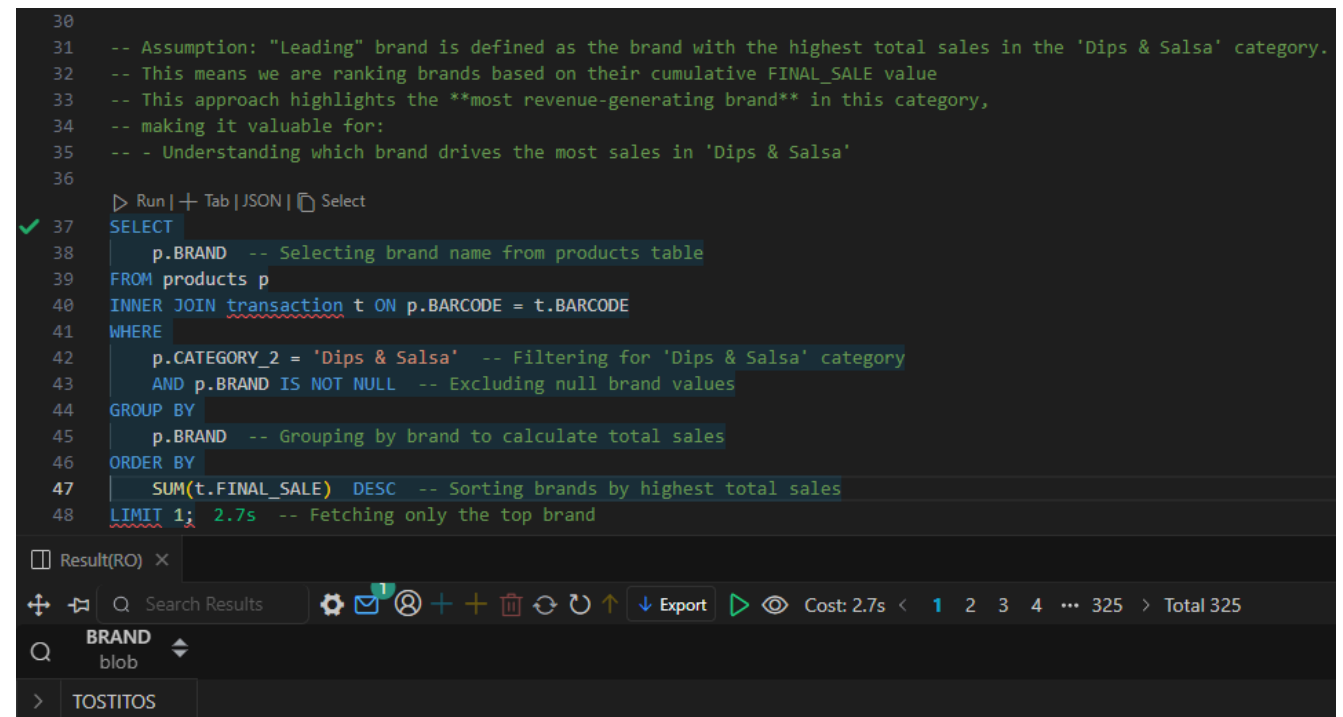
-- This approach highlights the **most revenue-generating brand** in this category,

-- making it valuable for:

-- - Understanding which brand drives the most sales in 'Dips & Salsa'

```
SELECT
  p.BRAND -- Selecting brand name from products table
FROM products p
INNER JOIN transaction t ON p.BARCODE = t.BARCODE
WHERE
  p.CATEGORY_2 = 'Dips & Salsa' -- Filtering for 'Dips & Salsa' category
  AND p.BRAND IS NOT NULL -- Excluding null brand values
GROUP BY
  p.BRAND -- Grouping by brand to calculate total sales
ORDER BY
  SUM(t.FINAL_SALE) DESC -- Sorting brands by highest total sales
LIMIT 1; -- Fetching only the top brand
```

Query + Output



The screenshot shows a SQL IDE interface. The top pane contains a SQL query with line numbers 30 to 48. The query is identical to the one in the previous block. Below the query editor, there is a toolbar with icons for running, saving, and other actions. The bottom pane shows the results of the query. It has a search bar with 'BRAND' and a dropdown menu showing 'blob'. Below that, a table with one row is visible, containing the brand name 'TOSTITOS'.

```
30
31 -- Assumption: "Leading" brand is defined as the brand with the highest total sales in the 'Dips & Salsa' category.
32 -- This means we are ranking brands based on their cumulative FINAL_SALE value
33 -- This approach highlights the **most revenue-generating brand** in this category,
34 -- making it valuable for:
35 -- - Understanding which brand drives the most sales in 'Dips & Salsa'
36
37 SELECT
38   p.BRAND -- Selecting brand name from products table
39 FROM products p
40 INNER JOIN transaction t ON p.BARCODE = t.BARCODE
41 WHERE
42   p.CATEGORY_2 = 'Dips & Salsa' -- Filtering for 'Dips & Salsa' category
43   AND p.BRAND IS NOT NULL -- Excluding null brand values
44 GROUP BY
45   p.BRAND -- Grouping by brand to calculate total sales
46 ORDER BY
47   SUM(t.FINAL_SALE) DESC -- Sorting brands by highest total sales
48 LIMIT 1; 2.7s -- Fetching only the top brand
```

Result(RO) X

Search Results

BRAND
blob

TOSTITOS

Query - At what percent has Fetch grown year over year?

-- Assumption: Fetch's year-over-year growth is measured by the number of new users onboarded each year.
-- This insight demonstrates the platform's expanding reach and increasing consumer participation,
-- making it a valuable tool for CPG brands and retailers to analyze shopping trends, loyalty patterns, and purchase behaviors.

```
WITH sum_of_users_per_year AS (  
  -- Calculate the number of unique users onboarded per year  
  SELECT  
    YEAR(created_date) AS current_year,  
    COUNT(DISTINCT id) AS count_of_unique_users,  
    LAG(COUNT(DISTINCT id)) OVER (ORDER BY YEAR(created_date)) AS count_of_unique_users_last_year  
  FROM user  
  GROUP BY YEAR(created_date)  
)  
-- Calculate the YoY growth percentage  
SELECT  
  current_year,  
  IFNULL(((count_of_unique_users - count_of_unique_users_last_year) * 100 / count_of_unique_users_last_year),0) AS  
yoy_growth_percentage  
FROM sum_of_users_per_year;
```

Query+output

```
55 WITH sum_of_users_per_year AS (  
56   -- Calculate the number of unique users onboarded per year  
57   SELECT  
58     YEAR(created_date) AS current_year,  
59     COUNT(DISTINCT id) AS count_of_unique_users,  
60     LAG(COUNT(DISTINCT id)) OVER (ORDER BY YEAR(created_date)) AS count_of_unique_users_last_year  
61   FROM user  
62   GROUP BY YEAR(created_date)  
63 )  
64 -- Calculate the YoY growth percentage  
65 SELECT  
66   current_year,  
67   IFNULL(((count_of_unique_users - count_of_unique_users_last_year) * 100 / count_of_unique_users_last_year),0) AS yoy_growth_percentage  
68 FROM sum_of_users_per_year; 1.2s  
69
```

current_year	yoy_growth_percentage
int	decimal
> 2014	0.0000
> 2015	70.0000
> 2016	37.2549
> 2017	821.4286
> 2018	236.5891
> 2019	226.7158
> 2020	138.1080
> 2021	13.4999
> 2022	39.8560
> 2023	-42.3589
> 2024	-24.8042