**Case Study #8 - Fresh Segments**

**A. Data Exploration and Cleansing**

**1 - Update the fresh\_segments.interest\_metrics table by modifying the month\_year column to be a date data type with the start of the month**

ALTER TABLE interest\_metrics

MODIFY month\_year VARCHAR(10);

UPDATE interest\_metrics

SET month\_year = CONCAT('01-',month\_year) ;

UPDATE interest\_metrics

SET month\_year = STR\_TO\_DATE(month\_year,'%d-%m-%Y');

ALTER TABLE interest\_metrics

MODIFY month\_year DATE

**2 - What is count of records in the fresh\_segments.interest\_metrics for each month\_year value sorted in chronological order (earliest to latest) with the null values appearing first?**

SELECT

month\_year,

COUNT(\*)

FROM interest\_metrics

GROUP BY month\_year

ORDER BY month\_year IS NOT NULL, month\_year ASC

**3 - What do you think we should do with these null values in the fresh\_segments.interest\_metrics?**

DELETE FROM interest\_metrics

WHERE interest\_id IS NULL

**4 - How many interest\_id values exist in the fresh\_segments.interest\_metrics table but not in the fresh\_segments.interest\_map table? What about the other way around?**

SELECT

COUNT(DISTINCT interest\_id) as metric\_count,

COUNT(DISTINCT id) AS map\_count,

COUNT(DISTINCT id) - COUNT(DISTINCT interest\_id) AS not\_in\_metric

FROM interest\_metrics AS metric

RIGHT JOIN interest\_map AS map ON metric.interest\_id = map.id

UNION

SELECT

COUNT(DISTINCT interest\_id) as metric\_count,

COUNT(DISTINCT id) AS map\_count,

COUNT(DISTINCT interest\_id) - COUNT(DISTINCT id) AS not\_in\_metric

FROM interest\_metrics AS metric

LEFT JOIN interest\_map AS map ON metric.interest\_id = map.id

**5 - Summarise the id values in the fresh\_segments.interest\_map by its total record count in this table.**

SELECT

COUNT(DISTINCT id) AS map\_count

FROM interest\_map

**6 - What sort of table join should we perform for our analysis and why? Check your logic by checking the rows where 'interest\_id = 21246' in your joined output and include all columns from fresh\_segments.interest\_metrics and all columns from fresh\_segments.interest\_map except from the id column.**

SELECT

\_month,

\_year,

month\_year,

interest\_id,

composition,

index\_value,

ranking,

percentile\_ranking,

interest\_name,

interest\_summary,

created\_at,

last\_modified

FROM interest\_metrics AS metric

JOIN interest\_map AS map ON metric.interest\_id = map.id

WHERE interest\_id = 21246 AND month\_year IS NOT NULL

**7 - Are there any records in your joined table where the month\_year value is before the created\_at value from the fresh\_segments.interest\_map table? Do you think these values are valid and why?**

SELECT

COUNT(\*)

FROM interest\_metrics AS metric

JOIN interest\_map AS map ON metric.interest\_id = map.id

WHERE month\_year < created\_at

-- 188 records

**B. Interest Analysis**

**1 - Which interests have been present in all month\_year dates in our dataset?**

SELECT

COUNT(DISTINCT month\_year) AS unique\_month\_count

FROM interest\_metrics;

-- 14 unique months

SELECT

interest\_id,

COUNT(\*) AS frequency

FROM interest\_metrics

GROUP BY interest\_id

HAVING frequency = 14

**2 - Using this same total\_months measure - calculate the cumulative percentage of all records starting at 14 months - which total\_months value passes the 90% cumulative percentage value?**

WITH count\_frequency AS

(SELECT

interest\_id,

COUNT(month\_year) as month\_frequency

FROM interest\_metrics

GROUP BY interest\_id),

count\_interest AS

(SELECT

month\_frequency,

COUNT(\*) AS interest\_count

FROM count\_frequency

GROUP BY month\_frequency

ORDER BY month\_frequency),

cal\_cumulative AS

(SELECT

\*,

ROUND(SUM(interest\_count) OVER(ORDER BY month\_frequency DESC) / SUM(interest\_count) OVER() \* 100 , 2) AS cumulative

FROM count\_interest

GROUP BY month\_frequency,interest\_count)

SELECT

\*

FROM cal\_cumulative

WHERE cumulative > 90

**3 - If we were to remove all interest\_id values which are lower than the total\_months value we found in the previous question - how many total data points would we be removing?**

WITH count\_frequency AS

(SELECT

interest\_id,

COUNT(month\_year) as month\_frequency

FROM interest\_metrics

GROUP BY interest\_id

HAVING month\_frequency <= 6 )

SELECT

COUNT(\*) AS record\_count

FROM interest\_metrics

WHERE interest\_id IN (SELECT interest\_id FROM count\_frequency);

**4 - Does this decision make sense to remove these data points from a business perspective? Use an example where there are all 14 months present to a removed interest example for your arguments - think about what it means to have less months present from a segment perspective.**

**5 - If we include all of our interests regardless of their counts - how many unique interests are there for each month?**

**C. Segment Analysis**

CREATE VIEW filtered\_data AS

WITH count\_frequency AS

(SELECT

interest\_id,

COUNT(month\_year) as month\_frequency

FROM interest\_metrics

GROUP BY interest\_id

HAVING month\_frequency >= 6 )

SELECT

\*

FROM interest\_metrics

WHERE interest\_id IN (SELECT interest\_id FROM count\_frequency)

**1 - Using the complete dataset - which are the top 10 and bottom 10 interests which have the largest composition values in any month\_year? Only use the maximum composition value for each interest but you must keep the corresponding month\_year**

-- Bottom 10

SELECT

month\_year, interest\_id, MAX(composition) AS max\_composition

FROM

filtered\_data

GROUP BY interest\_id , month\_year

ORDER BY max\_composition ASC,month\_year

LIMIT 10;

-- Top 10

SELECT

month\_year, interest\_id, MAX(composition) AS max\_composition

FROM

filtered\_data

GROUP BY interest\_id , month\_year

ORDER BY max\_composition DESC,month\_year

LIMIT 10;

**2 - Which 5 interests had the lowest average ranking value?**

SELECT

DISTINCT interest\_id,

AVG(ranking) OVER(PARTITION BY interest\_id) AS avg\_ranking

FROM

filtered\_data

ORDER BY avg\_ranking

LIMIT 5

**3 - Which 5 interests had the largest standard deviation in their percentile\_ranking value?**

SELECT

interest\_id,

STDDEV(percentile\_ranking) AS standard\_deviation

FROM

filtered\_data

GROUP BY interest\_id

ORDER BY standard\_deviation DESC

LIMIT 5

**4 - For the 5 interests found in the previous question - what was minimum and maximum percentile\_ranking values for each interest and its corresponding year\_month value? Can you describe what is happening for these 5 interests?**

WITH top\_5\_stddev AS

(SELECT

interest\_id,

STDDEV(percentile\_ranking) AS standard\_deviation

FROM

filtered\_data

GROUP BY interest\_id

ORDER BY standard\_deviation DESC

LIMIT 5)

SELECT

interest\_id,

MIN(percentile\_ranking) AS minimum\_pr,

MAX(percentile\_ranking) AS maximum\_pr

FROM filtered\_data

WHERE interest\_id IN (SELECT interest\_id FROM top\_5\_stddev)

GROUP BY interest\_id

**5 - How would you describe our customers in this segment based off their composition and ranking values? What sort of products or services should we show to these customers and what should we avoid?**

SELECT

interest\_id,

interest\_name,

interest\_summary,

AVG(composition) AS avg\_composition,

AVG(ranking) AS avg\_ranking

FROM

filtered\_data AS filtered\_metric

JOIN interest\_map AS map ON filtered\_metric.interest\_id = map.id

GROUP BY interest\_id,

interest\_name,

interest\_summary

ORDER BY avg\_composition DESC, avg\_ranking ASC

LIMIT 10

**D. Index Analysis**

CREATE VIEW metric\_with\_avg\_composition AS

SELECT

\*,

ROUND(composition/index\_value,2) AS avg\_composition

FROM interest\_metrics

The index\_value is a measure which can be used to reverse calculate the average composition for Fresh Segments’ clients. Average composition can be calculated by dividing the composition column by the index\_value column rounded to 2 decimal places.

**1 - What is the top 10 interests by the average composition for each month?**

SELECT

month\_year,

interest\_id,

avg\_composition,

RANK() OVER(PARTITION BY month\_year ORDER BY avg\_composition DESC) AS ranking

FROM metric\_with\_avg\_composition

WHERE ranking <= 10

ORDER BY month\_year

**2 - For all of these top 10 interests - which interest appears the most often?**

WITH cte AS

(SELECT

month\_year,

interest\_id,

RANK() OVER(PARTITION BY month\_year ORDER BY avg\_composition DESC) AS ranking

FROM metric\_with\_avg\_composition

WHERE ranking <= 10

ORDER BY month\_year)

SELECT

interest\_id,

COUNT(\*) AS frequency

FROM cte

GROUP BY interest\_id

ORDER BY frequency DESC

LIMIT 10

**3 - What is the average of the average composition for the top 10 interests for each month?**

WITH cte AS

(SELECT

month\_year,

interest\_id,

avg\_composition,

RANK() OVER(PARTITION BY month\_year ORDER BY avg\_composition DESC) AS ranking

FROM metric\_with\_avg\_composition

WHERE ranking <= 10

ORDER BY month\_year)

SELECT

month\_year,

ROUND(AVG(avg\_composition),2) AS avg\_composition\_each\_month

FROM cte

GROUP BY month\_year;

**4 - What is the 3 month rolling average of the max average composition value from September 2018 to August 2019 and include the previous top ranking interests in the same output shown below.**

**5 - Provide a possible reason why the max average composition might change from month to month? Could it signal something is not quite right with the overall business model for Fresh Segments?**