

Case Study - 8

Tuesday, February 15, 2022 2:39 PM

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 fresh_segments.interest_metrics
alter month_year type date using to_date(month_year,'MM-YYYY');
```

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 fresh_segments.interest_metrics
group by month_year
order by month_year NULLS FIRST;
```

month_year	count
null	1194
2018-07-01T00:00:00.000Z	729
2018-08-01T00:00:00.000Z	767
2018-09-01T00:00:00.000Z	780
2018-10-01T00:00:00.000Z	857
2018-11-01T00:00:00.000Z	928
2018-12-01T00:00:00.000Z	995
2019-01-01T00:00:00.000Z	973
2019-02-01T00:00:00.000Z	1121
2019-03-01T00:00:00.000Z	1136
2019-04-01T00:00:00.000Z	1099
2019-05-01T00:00:00.000Z	857
2019-06-01T00:00:00.000Z	824

3. What do you think we should do with these null values in the `fresh_segments.interest_metrics`

We can't delete row which has NULL values Because in DataSet has other columns has Some values whenever we delete that Rows so entire rows Are delete not only in Month_year which has Null Values.

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?

Yes lots of values in `fresh_segments.interest_metrics` but when you are compare unique interest id of `fresh_segments.interest_metrics` table then it gives me 1202 Records its near to values of `fresh_segments.interest_map` table.

```
select count(distinct interest_id) from fresh_segments.interest_metrics;
select count(distinct id) from fresh_segments.interest_map ;
```

Query #3	Execution time: 5ms
count	

1202
Query #4 Execution time: 11ms
count
1209

5. Summarise the **id** values in the **fresh_segments.interest_map** by its total record count in this table

```
select * from fresh_segments.interest_map;
```

id	interest_name	interest_summary	created_at	last_modified
1	Fitness Enthusiasts	Consumers using fitness tracking apps and websites.	2016-05-26T14:57:59.000Z	2018-05-23T11:30:12.000Z
2	Gamers	Consumers researching game reviews and cheat codes.	2016-05-26T14:57:59.000Z	2018-05-23T11:30:12.000Z
3	Car Enthusiasts	Readers of automotive news and car reviews.	2016-05-26T14:57:59.000Z	2018-05-23T11:30:12.000Z
4	Luxury Retail Researchers	Consumers researching luxury product reviews and gift ideas.	2016-05-26T14:57:59.000Z	2018-05-23T11:30:12.000Z
5	Brides & Wedding Planners	People researching wedding ideas and vendors.	2016-05-26T14:57:59.000Z	2018-05-23T11:30:12.000Z
6	Vacation Planners	Consumers reading reviews of vacation destinations and accommodations.	2016-05-26T14:57:59.000Z	2018-05-23T11:30:13.000Z
7	Motorcycle Enthusiasts	Readers of motorcycle news and reviews.	2016-05-26T14:57:59.000Z	2018-05-23T11:30:13.000Z
8	Business News Readers	Readers of online business news content.	2016-05-26T14:57:59.000Z	2018-05-23T11:30:12.000Z
12	Thrift Store Shoppers	Consumers shopping online for clothing at	2016-05-26T14:57:59.000Z	2018-03-26T14:57:59.000Z

DB Fiddle – Crafted with ♥ by Status200 in the United Kingdom. Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018

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.

Here I used Left Join Because it Returns All The rows of left side table and matched rows with right Side table where interest_id= 21246 and include all columns of both tables except from id column of **fresh_segments.interest_map**.

```
alter table fresh_segments.interest_metrics
alter interest_id type integer using(interest_id::int);
select
    metr.interest_id,
    metr._month,
    metr._year,
    metr.month_year,
    metr.composition,
    metr.index_value,
    metr.ranking,
    Metr.percentage_ranking,
    m.interest_name,
    m.interest_summary,
    m.created_at,
    m.last_modified
```

```

from
fresh_segments.interest_map m Left join fresh_segments.interest_metrics metr
on metr.interest_id=m.id
where interest_id = 21246
group by metr.interest_id,
metr._month,
metr._year,
metr.month_year,
metr.composition,
metr.index_value,
metr.ranking,
metr.percentile_ranking,
m.interest_name,
m.interest_summary,
m.created_at,
m.last_modified;

```

interest_id	_month	_year	month_year	composition	index_value	ranking	percentile_ranking	interest_name	interest_summary	created_at	last_modified
21246	1	2019	01-2019	2.05	0.76	954	1.95	Readers of El Salvadoran Content	People reading news from El Salvadoran media sources.	2018-06-11T17:50:04.000Z	2018-06-11T17:50:04.000Z
21246	3	2019	03-2019	1.75	0.67	1123	1.14	Readers of El Salvadoran Content	People reading news from El Salvadoran media sources.	2018-06-11T17:50:04.000Z	2018-06-11T17:50:04.000Z
21246	11	2018	11-2018	2.25	0.78	908	2.16	Readers of El Salvadoran Content	People reading news from El Salvadoran media sources.	2018-06-11T17:50:04.000Z	2018-06-11T17:50:04.000Z
21246	8	2018	08-2018	2.13	0.59	765	0.26	Readers of El Salvadoran Content	People reading news from El Salvadoran media sources.	2018-06-11T17:50:04.000Z	2018-06-11T17:50:04.000Z
21246	2	2019	02-2019	1.84	0.68	1109	1.07	Readers of El Salvadoran Content	People reading news from El Salvadoran media sources.	2018-06-11T17:50:04.000Z	2018-06-11T17:50:04.000Z
21246	12	2018	12-2018	1.97	0.7	983	1.21	Readers of El Salvadoran Content	People reading news from El Salvadoran media sources.	2018-06-11T17:50:04.000Z	2018-06-11T17:50:04.000Z
21246	null	null	null	1.61	0.68	1191	0.25	Readers of El Salvadoran Content	People reading news from El Salvadoran media sources.	2018-06-11T17:50:04.000Z	2018-06-11T17:50:04.000Z
21246	9	2018	09-2018	2.06	0.61	774	0.77	Readers of El Salvadoran Content	People reading news from El Salvadoran media sources.	2018-06-11T17:50:04.000Z	2018-06-11T17:50:04.000Z
21246	10	2018	10-2018	1.74	0.58	855	0.23	Readers of El Salvadoran Content	People reading news from El Salvadoran media sources.	2018-06-11T17:50:04.000Z	2018-06-11T17:50:04.000Z
								Readers of El Salvadoran	People reading news from El Salvadoran media	2018-06-	2018-06-

DB Fiddle - Crafted with • by Statista200 in the United Kingdom.

Terms of Use • Privacy / Cookie Policy • Statista200 Ltd © 2018

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

```

metr.interest_id,
metr._month,
metr._year,
metr.month_year,
metr.composition,
metr.index_value,
metr.ranking,
metr.percentile_ranking,
m.interest_name,
m.interest_summary,
m.created_at,
m.last_modified

```

from

Fresh_segments.interest_map m inner join fresh_segments.interest_metrics metr

on metr.interest_id=m.id

where m.created_at < metr.month_year

group by metr.interest_id,

metr._month,

metr._year,

metr.month_year,

metr.composition,

metr.index_value,

metr.ranking,

metr.percentile_ranking,

m.interest_name,

m.interest_summary,

m.created_at,

m.last_modified;

interest_id	_month	_year	month_year	composition	index_value	ranking	percentile_ranking	interest_name	interest_summary	created_at	last_modified
1	1	2019	2019-01-01T00:00:00.000Z	2.38	1.59	177	81.81	Fitness Enthusiasts	Consumers using fitness tracking apps and websites.	2016-05-26T14:57:59.000Z	2018-05-23T11:30:12.000Z
1	10	2018	2018-10-01T00:00:00.000Z	3.71	1.84	118	86.23	Fitness Enthusiasts	Consumers using fitness tracking apps and websites.	2016-05-26T14:57:59.000Z	2018-05-23T11:30:12.000Z
1	11	2018	2018-11-01T00:00:00.000Z	2.79	1.84	124	86.64	Fitness Enthusiasts	Consumers using fitness tracking apps and websites.	2016-05-26T14:57:59.000Z	2018-05-23T11:30:12.000Z
1	12	2018	2018-12-01T00:00:00.000Z	2.94	1.83	140	85.93	Fitness Enthusiasts	Consumers using fitness tracking apps and websites.	2016-05-26T14:57:59.000Z	2018-05-23T11:30:12.000Z
1	2	2019	2019-02-01T00:00:00.000Z	2.55	1.32	495	55.84	Fitness Enthusiasts	Consumers using fitness tracking apps and websites.	2016-05-26T14:57:59.000Z	2018-05-23T11:30:12.000Z
1	3	2019	2019-03-01T00:00:00.000Z	2.76	1.54	244	78.52	Fitness Enthusiasts	Consumers using fitness tracking apps and websites.	2016-05-26T14:57:59.000Z	2018-05-23T11:30:12.000Z
1	4	2019	2019-04-01T00:00:00.000Z	2.28	1.5	273	75.16	Fitness Enthusiasts	Consumers using fitness tracking apps and websites.	2016-05-26T14:57:59.000Z	2018-05-23T11:30:12.000Z
1	5	2019	2019-05-01T00:00:00.000Z	1.68	1.62	377	56.01	Fitness Enthusiasts	Consumers using fitness tracking apps and websites.	2016-05-26T14:57:59.000Z	2018-05-23T11:30:12.000Z

<https://tally.so/r/mOJVgn> United Kingdom.

Terms of Use • Privacy / Cookie Policy • Status200 Ltd © 2018

Interest Analysis

1. Which interests have been present in all **month_year** dates in our dataset?

```
alter table fresh_segments.interest_metrics
```

```
alter interest_id type integer using(interest_id::int);
```

```
select
```

```
m.interest_name,
```

```
metr.month_year
```

```
from fresh_segments.interest_map m inner join fresh_segments.interest_metrics metr
```

```
on m.id=metr.interest_id
```

```
group by m.interest_name,metr.month_year
```

```
order by month_year;
```

interest_name

month_year

Accounting & CPA Continuing Education Researchers	2018-07-01T00:00:00.000Z
Conservative Think Tank Readers	2018-07-01T00:00:00.000Z
Tech-Savvy Moms	2018-07-01T00:00:00.000Z
Mexican Food Enthusiasts	2018-07-01T00:00:00.000Z
Marijuana Legalization Advocates	2018-07-01T00:00:00.000Z
Camaro Enthusiasts	2018-07-01T00:00:00.000Z
Lobbyists	2018-07-01T00:00:00.000Z
Natural and Holistic Health Researchers	2018-07-01T00:00:00.000Z
HDTV Researchers	2018-07-01T00:00:00.000Z
Luxury Travel Researchers	2018-07-01T00:00:00.000Z
Asthma Sufferers	2018-07-01T00:00:00.000Z
Flower & Gift Basket Shoppers	2018-07-01T00:00:00.000Z
Florida Gulf Coast Travel Researchers	2018-07-01T00:00:00.000Z

DB Fiddle - Crafted with ♥ by Status200 in the United Kingdom. [Terms of Use](#) • [Privacy / Cookie Policy](#) • Status200 Ltd © 2018

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 cte_interest_months AS (
SELECT
interest_id,
MAX(DISTINCT month_year) AS total_months
FROM fresh_segments.interest_metrics
WHERE interest_id IS NOT NULL
GROUP BY interest_id),

cte_interest_counts AS(
SELECT
total_months,
COUNT(DISTINCT interest_id) AS interest_count
FROM cte_interest_months
GROUP BY total_months
)
SELECT
total_months,
interest_count,
ROUND(100* SUM(interest_count) OVER (ORDER BY total_months DESC) /
(SUM(INTEREST_COUNT) OVER ()),2) AS cumulative_percentage
FROM cte_interest_counts;

```

total_months	interest_count	cumulative_percentage
12-2018	995	82.78
11-2018	15	84.03
10-2018	10	84.86
09-2018	4	85.19
08-2019	161	98.59
08-2018	4	98.92
07-2019	1	99.00
07-2018	6	99.50
03-2019	4	99.83

02-2019	2	100.00
---------	---	--------

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 cte_interest_months AS
(
SELECT
interest_id,
MAX(DISTINCT month_year) AS total_months
FROM fresh_segments.interest_metrics
WHERE interest_id IS NOT NULL
GROUP BY interest_id
),
cte_interest_counts AS
(
SELECT
total_months,
COUNT(DISTINCT interest_id) AS interest_count
FROM cte_interest_months
GROUP BY total_months
)
SELECT
SUM(interest_count) AS total_values_to_be_removed
FROM cte_interest_counts
```

Query #1 Execution time: 21ms

total_values_to_be_removed
1202

4. After removing these interests - how many unique interests are there for each month?

```
DELETE
FROM fresh_segments.interest_metrics
WHERE interest_id IS NOT NULL and month_year=(select MAX(DISTINCT month_year) from
fresh_segments.interest_metrics);
```

```
select
count(Distinct interest_id),
extract(month from month_year) as Month
from fresh_segments.interest_metrics
WHERE month_year IS NOT NULL group by Month ;
```

count	month
973	1
1121	2

1136	3
1099	4
857	5
824	6
1051	7
767	8
780	9
857	10
928	11
995	12

DB Fiddle – Crafted with ♥ by Status200 in the United Kingdom. [Terms of Use](#) • [Privacy / Cookie Policy](#) • Status200 Ltd © 2018

Segment Analysis

- Using our filtered dataset by removing the interests with less than 6 months worth of data, 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**

select

```
(select Distinct(interest_id)as id
from fresh_segments.interest_metrics
where composition=(select max(composition)
from fresh_segments.interest_metrics) order by id LIMIT 10 )As top_10,
```

```
(select Distinct(interest_id)as id
from fresh_segments.interest_metrics
where composition=(select max(composition)
from fresh_segments.interest_metrics)order by id DESC LIMIT 10)As bottom_10;
```

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

select *

```
from fresh_segments.interest_metrics
where ranking = (select avg(rank_min)
from (select min(ranking) as rank_min
from fresh_segments.interest_metrics)t) LIMIT 5;
```

_month	_year	month_year	interest_id	composition	index_value	ranking	percentile_ranking
7	2018	2018-07-01T00:00:00.000Z	32486	11.89	6.19	1	99.86
8	2018	2018-08-01T00:00:00.000Z	6218	5.52	2.84	1	99.87
9	2018	2018-09-01T00:00:00.000Z	6218	4.61	2.84	1	99.87
10	2018	2018-10-01T00:00:00.000Z	6218	6.39	3.37	1	99.88
11	2018	2018-11-01T00:00:00.000Z	6285	7.56	3.48	1	99.89



3. Which 5 interests had the largest standard deviation in their **percentile_ranking** value?

select *

```
from fresh_segments.interest_metrics
where interest_id is not null and percentile_ranking =
(select max(percentile_ranking)
from fresh_segments.interest_metrics )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?

```
select
(select max(percentile_ranking) from fresh_segments.interest_metrics) As
Max_Percetile_Ranking,
(select min(percentile_ranking) from fresh_segments.interest_metrics) As
Min_percentile_ranking;
```

Query #2 Execution time: 3ms

max_percetile_ranking	min_percentile_ranking
99.92	0

Index Analysis

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

```
select
m.interest_id,
i.interest_name,
extract(month from m.month_year)as Month,
avg(m.composition) from
fresh_segments.interest_metrics m inner join fresh_segments.interest_map i on
m.interest_id=i.id
where m.interest_id IS NOT NULL
group by m.interest_id,month,i.interest_name LIMIT 10;
```

Query #3 Execution time: 6ms

interest_id	interest_name	month	avg
1	Fitness Enthusiasts	1	2.38
1	Fitness Enthusiasts	2	2.55
1	Fitness Enthusiasts	3	2.76
1	Fitness Enthusiasts	4	2.28
1	Fitness Enthusiasts	5	1.68
1	Fitness Enthusiasts	7	7.02
1	Fitness Enthusiasts	8	3.04
1	Fitness Enthusiasts	9	2.44
1	Fitness Enthusiasts	10	3.71
1	Fitness Enthusiasts	11	2.79

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

```
select max(interest_name) from
(select
    m.interest_id,
    i.interest_name,
    extract(month from m.month_year)as Month,
    avg(m.composition) from
    fresh_segments.interest_metrics m inner join fresh_segments.interest_map i on
    m.interest_id=i.id
    where m.interest_id IS NOT NULL
    group by m.interest_id,month,i.interest_name LIMIT 10)t;
```

Query #4	Execution time: 13ms
max	
Fitness Enthusiasts	
Query #5	Execution time: 15ms

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

```
select avg(compos) from
(select
    m.interest_id,
    i.interest_name,
    extract(month from m.month_year)as Month,
    avg(m.composition) as compos from
    fresh_segments.interest_metrics m inner join fresh_segments.interest_map
    i on m.interest_id=i.id
    where m.interest_id IS NOT NULL
    group by m.interest_id,month,i.interest_name LIMIT 10)t;
```

Query #5	Execution time: 15ms
avg	
3.065	