Analytics Portfolio

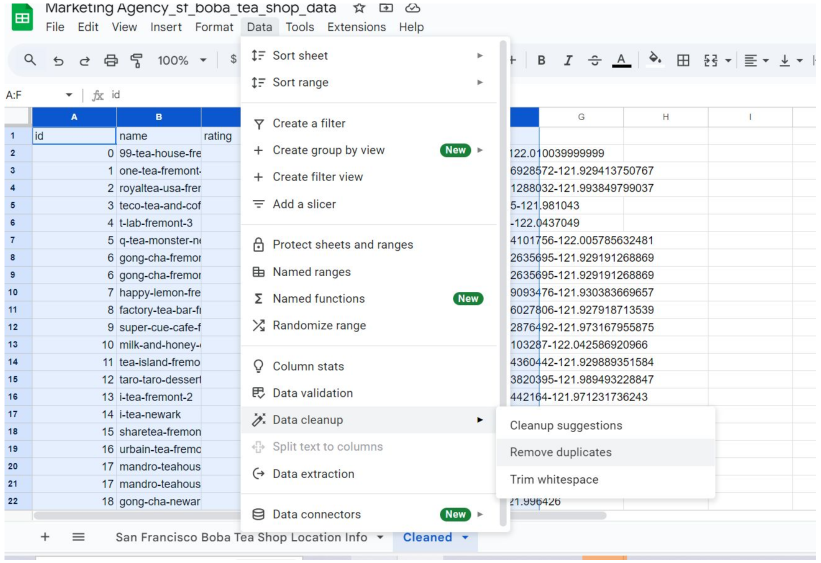
Jessica Gutstein

Dirty to Clean Data - Google Sheets Project

Skills: identify dirty elements using **Google Sheets** in a dataset, remove duplicate data, and use the **`COUNTIF`** and **`SPLIT`** functions to help clean data.

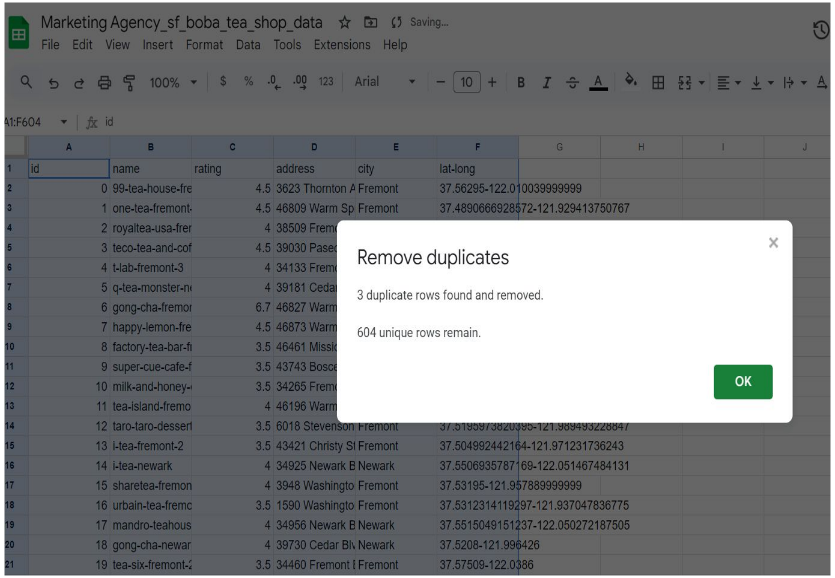
Dataset: Marketing Agency

https://docs.google.com/spreadsheets/d/1p5TiRjV-criLJTI7A0tBdMF0tEUiBIdjY3xL0KOGGHg/edit?usp=sharing



A:F | id

	A	B		
1	id	name	rating	
2	0	99-tea-house-fre		122.010039999999
3	1	one-tea-fremont-		6928572-121.929413750767
4	2	royaltea-usa-frer		1288032-121.993849799037
5	3	teco-tea-and-cof		5-121.981043
6	4	t-lab-fremont-3		-122.0437049
7	5	q-tea-monster-n		4101756-122.005785632481
8	6	gong-cha-fremor		2635695-121.929191268869
9	6	gong-cha-fremor		2635695-121.929191268869
10	7	happy-lemon-fre		9093476-121.930383669657
11	8	factory-tea-bar-fi		6027806-121.927918713539
12	9	super-cue-cafe-f		2876492-121.973167955875
13	10	milk-and-honey-		103287-122.042586920966
14	11	tea-island-fremo		4360442-121.929889351584
15	12	taro-taro-dessert		3820395-121.989493228847
16	13	i-tea-fremont-2		442164-121.971231736243
17	14	i-tea-newark		
18	15	sharetea-fremon		
19	16	urbain-tea-fremc		
20	17	mandro-teahous		
21	17	mandro-teahous		
22	18	gong-cha-newar		21.996426


Marketing Agency_sf_boba_tea_shop_data
Saving...
File Edit View Insert Format Data Tools Extensions Help
100%
Arial
10
OK

A1:F604 | id

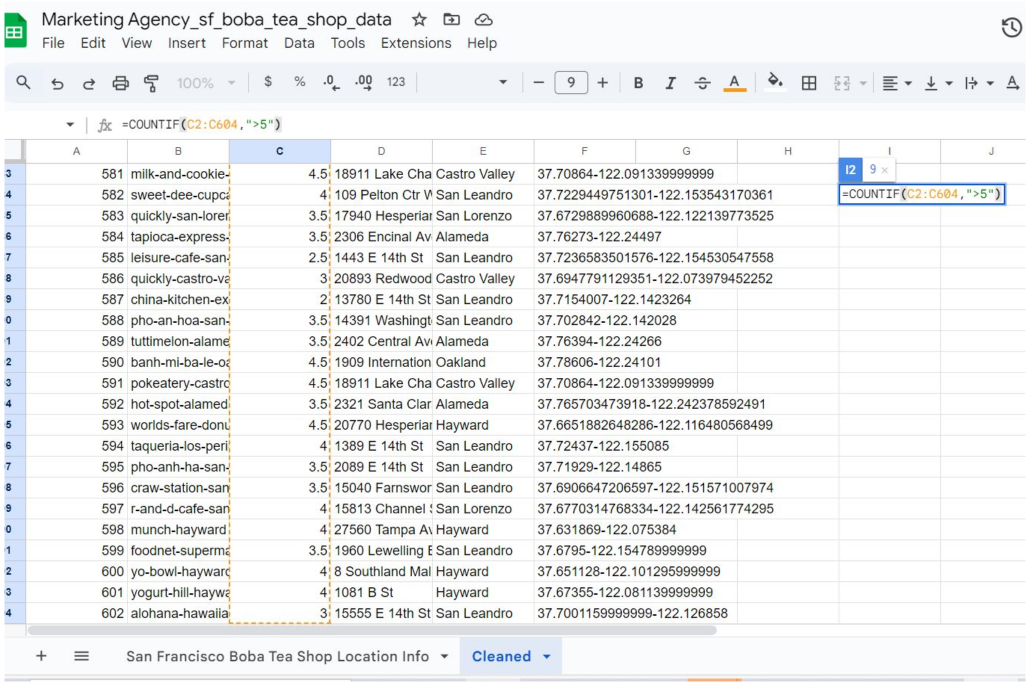
	A	B	C	D	E	F
1	id	name	rating	address	city	lat-long
2	0	99-tea-house-fre	4.5	3623 Thornton A	Fremont	37.56295-122.010039999999
3	1	one-tea-fremont-	4.5	46809 Warm Sp	Fremont	37.4890666928572-121.929413750767
4	2	royaltea-usa-frer	4	38509 Frem		
5	3	teco-tea-and-cof	4.5	39030 Pasec		
6	4	t-lab-fremont-3	4	34133 Frem		
7	5	q-tea-monster-n	4	39181 Cedai		
8	6	gong-cha-fremor	6.7	46827 Warm		
9	7	happy-lemon-fre	4.5	46873 Warm		
10	8	factory-tea-bar-fi	3.5	46461 Missic		
11	9	super-cue-cafe-f	3.5	43743 Bosce		
12	10	milk-and-honey-	3.5	34265 Frem		
13	11	tea-island-fremo	4	46196 Warm		
14	12	taro-taro-dessert	3.5	6018 Stevenson	Fremont	37.5195973820395-121.989493228847
15	13	i-tea-fremont-2	3.5	43421 Christy St	Fremont	37.504992442164-121.971231736243
16	14	i-tea-newark	4	34925 Newark B	Newark	37.5506935787169-122.051467484131
17	15	sharetea-fremon	4	3948 Washingto	Fremont	37.53195-121.957889999999
18	16	urbain-tea-fremc	3.5	1590 Washingto	Fremont	37.5312314119297-121.937047836775
19	17	mandro-teahous	4	34956 Newark B	Newark	37.5515049151237-122.050272187505
20	18	gong-cha-newar	4	39730 Cedar Blv	Newark	37.5208-121.996426
21	19	tea-six-fremont-2	3.5	34460 Fremont I	Fremont	37.57509-122.0386

Remove duplicates

3 duplicate rows found and removed.

604 unique rows remain.

Functions - COUNTIF



Marketing Agency_sf_boba_tea_shop_data

`=COUNTIF(C2:C604,">5")`

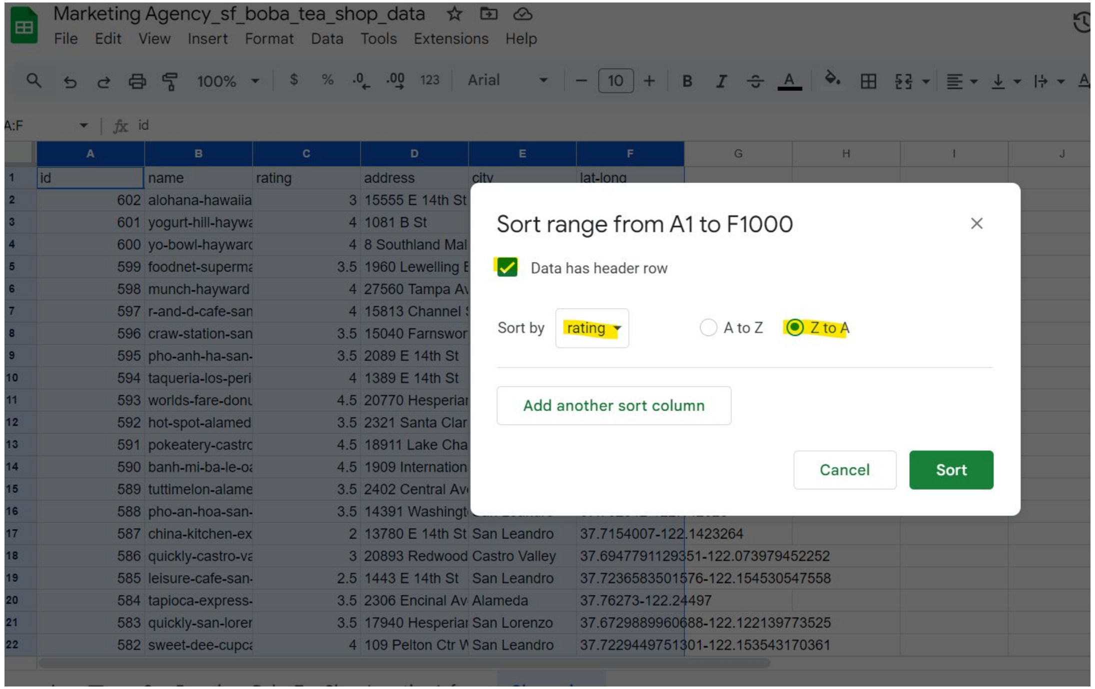
	A	B	C	D	E	F	G	H	I	J
3	581	milk-and-cookie-	4.5	18911 Lake Cha	Castro Valley	37.70864-122.091339999999			9	
4	582	sweet-dee-cupca	4	109 Pelton Ctr W	San Leandro	37.7229449751301-122.153543170361			=COUNTIF(C2:C604,">5")	
5	583	quickly-san-lorer	3.5	17940 Hesperiar	San Lorenzo	37.6729889960688-122.122139773525				
6	584	tapioca-express-	3.5	2306 Encinal Av	Alameda	37.76273-122.24497				
7	585	leisure-cafe-san-	2.5	1443 E 14th St	San Leandro	37.7236583501576-122.154530547558				
8	586	quickly-castro-va	3	20893 Redwood	Castro Valley	37.6947791129351-122.073979452252				
9	587	china-kitchen-ex	2	13780 E 14th St	San Leandro	37.7154007-122.1423264				
0	588	pho-an-hoa-san-	3.5	14391 Washingt	San Leandro	37.702842-122.142028				
1	589	tuttimelon-alame	3.5	2402 Central Av	Alameda	37.76394-122.24266				
2	590	banh-mi-ba-le-oa	4.5	1909 Internation	Oakland	37.78606-122.24101				
3	591	pokeatery-castro	4.5	18911 Lake Cha	Castro Valley	37.70864-122.091339999999				
4	592	hot-spot-alamed	3.5	2321 Santa Clar	Alameda	37.765703473918-122.242378592491				
5	593	worlds-fare-donu	4.5	20770 Hesperiar	Hayward	37.6651882648286-122.116480568499				
6	594	taqueria-los-peri	4	1389 E 14th St	San Leandro	37.72437-122.155085				
7	595	pho-anh-ha-san-	3.5	2089 E 14th St	San Leandro	37.71929-122.14865				
8	596	craw-station-san	3.5	15040 Farnswor	San Leandro	37.6906647206597-122.151571007974				
9	597	r-and-d-cafe-san	4	15813 Channel S	San Lorenzo	37.6770314768334-122.142561774295				
0	598	munch-hayward	4	27560 Tampa Av	Hayward	37.631869-122.075384				
1	599	foodnet-superma	3.5	1960 Lewelling E	San Leandro	37.6795-122.154789999999				
2	600	yo-bowl-haywarc	4	8 Southland Mal	Hayward	37.651128-122.101295999999				
3	601	yogurt-hill-haywa	4	1081 B St	Hayward	37.67355-122.081139999999				
4	602	alohana-hawaiia	3	15555 E 14th St	San Leandro	37.7001159999999-122.126858				

San Francisco Boba Tea Shop Location Info | Cleaned

- Yelp Rates in Column C can only be between (0-5).
- Using COUNTIF to determine if any values are over 5.
- 9 entries that have a rating greater than 5.
- Solution:

1. Research Boba locations on yelp to find the accurate rankings, in this dataset example, the incorrect entries will be replaced with 5.

Sorting/Filtering



Marketing Agency_sf_boba_tea_shop_data

A:F | fx id

	A	B	C	D	E	F
1	id	name	rating	address	city	lat-long
2	602	alohana-hawaiia	3	15555 E 14th St		
3	601	yogurt-hill-haywa	4	1081 B St		
4	600	yo-bowl-haywarc	4	8 Southland Mal		
5	599	foodnet-superma	3.5	1960 Lewelling E		
6	598	munch-hayward	4	27560 Tampa Av		
7	597	r-and-d-cafe-san	4	15813 Channel S		
8	596	craw-station-san	3.5	15040 Farnswor		
9	595	pho-anh-ha-san-	3.5	2089 E 14th St		
10	594	taqueria-los-peri	4	1389 E 14th St		
11	593	worlds-fare-donu	4.5	20770 Hesperiar		
12	592	hot-spot-alamed	3.5	2321 Santa Clar		
13	591	pokeatery-castrc	4.5	18911 Lake Cha		
14	590	banh-mi-ba-le-oa	4.5	1909 Internation		
15	589	tuttimelon-alame	3.5	2402 Central Av		
16	588	pho-an-hoa-san-	3.5	14391 Washingt		
17	587	china-kitchen-ex	2	13780 E 14th St	San Leandro	37.7154007-122.1423264
18	586	quickly-castro-va	3	20893 Redwood	Castro Valley	37.6947791129351-122.073979452252
19	585	leisure-cafe-san-	2.5	1443 E 14th St	San Leandro	37.7236583501576-122.154530547558
20	584	tapioca-express-	3.5	2306 Encinal Av	Alameda	37.76273-122.24497
21	583	quickly-san-lorer	3.5	17940 Hesperiar	San Lorenzo	37.6729889960688-122.122139773525
22	582	sweet-dee-cupca	4	109 Pelton Ctr W	San Leandro	37.7229449751301-122.153543170361

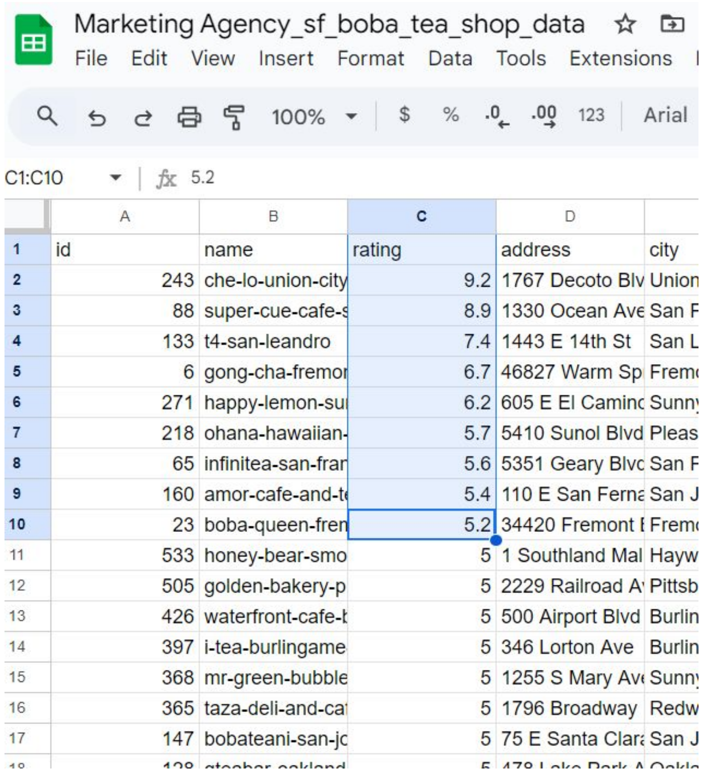
Sort range from A1 to F1000

[✓] Data has header row

Sort by: rating — A to Z [✓ Z to A]

Add another sort column

Cancel | Sort



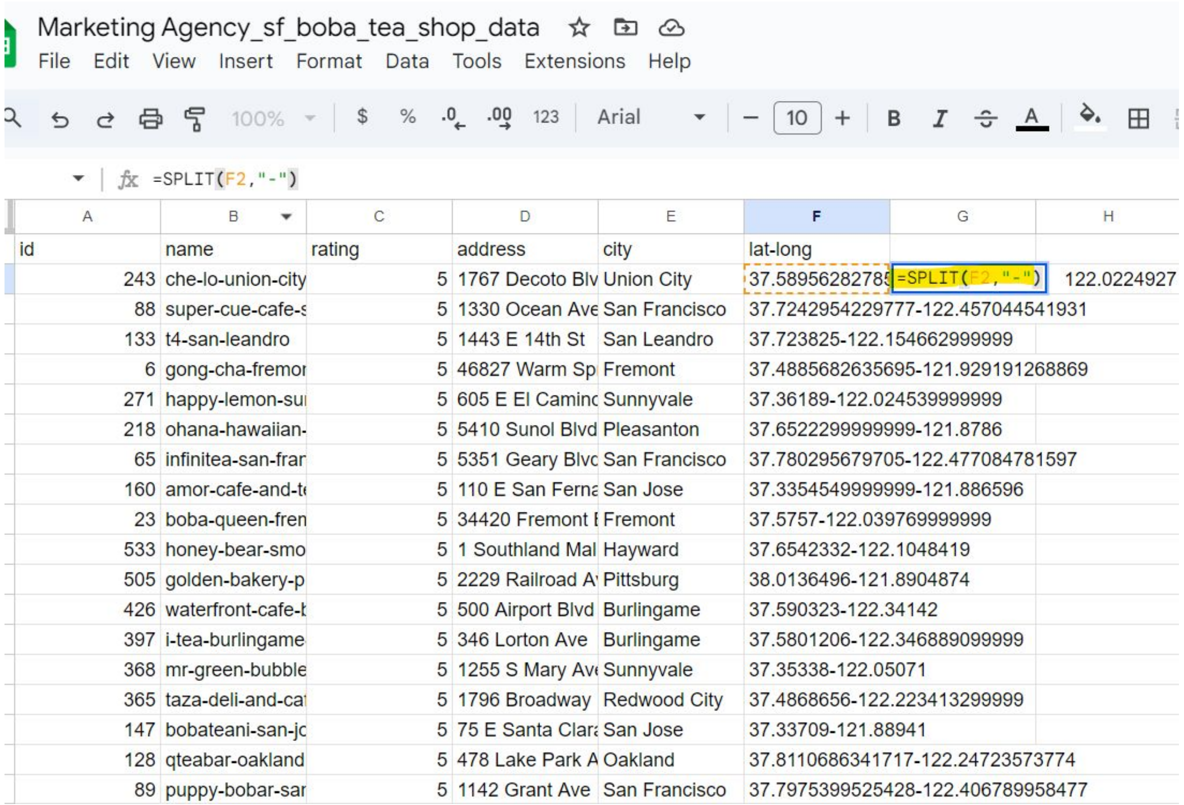
Marketing Agency_sf_boba_tea_shop_data

C1:C10 | fx 5.2

	A	B	C	D	E
1	id	name	rating	address	city
2	243	che-lo-union-city	9.2	1767 Decoto Blv	Union
3	88	super-cue-cafe-s	8.9	1330 Ocean Ave	San F
4	133	t4-san-leandro	7.4	1443 E 14th St	San L
5	6	gong-cha-fremor	6.7	46827 Warm Sp	Fremo
6	271	happy-lemon-su	6.2	605 E El Caminc	Sunny
7	218	ohana-hawaiian-	5.7	5410 Sunol Blvd	Pleas
8	65	infinitea-san-fran	5.6	5351 Geary Blvc	San F
9	160	amor-cafe-and-te	5.4	110 E San Ferna	San J
10	23	boba-queen-fren	5.2	34420 Fremont E	Fremo
11	533	honey-bear-smo	5	1 Southland Mal	Hayw
12	505	golden-bakery-p	5	2229 Railroad A	Pittsb
13	426	waterfront-cafe-b	5	500 Airport Blvd	Burlin
14	397	i-tea-burlingame	5	346 Lorton Ave	Burlin
15	368	mr-green-bubble	5	1255 S Mary Ave	Sunny
16	365	taza-deli-and-ca	5	1796 Broadway	Redw
17	147	bobateani-san-jc	5	75 E Santa Clara	San J

Solution: Replaced all 9 incorrect data entries with 5. To validate correction, the COUNTIF function will show 0 entries >5.

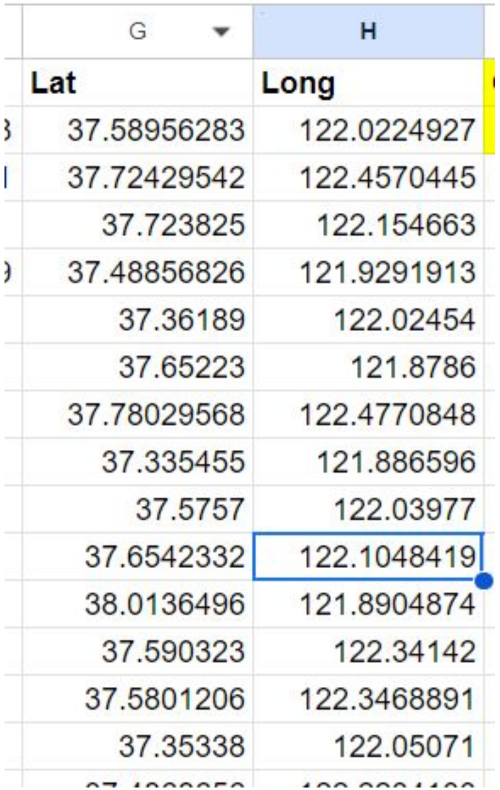
Functions - SPLIT


Marketing Agency_sf_boba_tea_shop_data

File Edit View Insert Format Data Tools Extensions Help

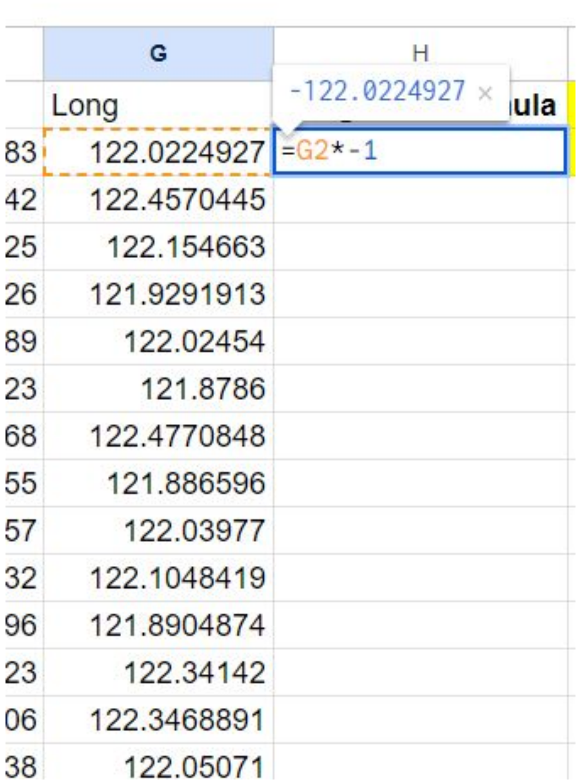
```
fx =SPLIT(F2,"-")
```

	A	B	C	D	E	F	G	H
	id	name	rating	address	city	lat-long		
	243	che-lo-union-city	5	1767 Decoto Blv	Union City	37.58956282785	=SPLIT(F2,"-")	122.0224927
	88	super-cue-cafe-s	5	1330 Ocean Ave	San Francisco	37.7242954229777-122.457044541931		
	133	t4-san-leandro	5	1443 E 14th St	San Leandro	37.723825-122.154662999999		
	6	gong-cha-fremo	5	46827 Warm Sp	Fremont	37.4885682635695-121.929191268869		
	271	happy-lemon-su	5	605 E El Caminc	Sunnyvale	37.36189-122.024539999999		
	218	ohana-hawaiian-	5	5410 Sunol Blvd	Pleasanton	37.6522299999999-121.8786		
	65	infinitea-san-fra	5	5351 Geary Blvc	San Francisco	37.780295679705-122.477084781597		
	160	amor-cafe-and-t	5	110 E San Ferna	San Jose	37.3354549999999-121.886596		
	23	boba-queen-fren	5	34420 Fremont I	Fremont	37.5757-122.039769999999		
	533	honey-bear-smo	5	1 Southland Mal	Hayward	37.6542332-122.1048419		
	505	golden-bakery-p	5	2229 Railroad A	Pittsburg	38.0136496-121.8904874		
	426	waterfront-cafe-l	5	500 Airport Blvd	Burlingame	37.590323-122.34142		
	397	i-tea-burlingame	5	346 Lorton Ave	Burlingame	37.5801206-122.346889099999		
	368	mr-green-bubble	5	1255 S Mary Ave	Sunnyvale	37.35338-122.05071		
	365	taza-deli-and-ca	5	1796 Broadway	Redwood City	37.4868656-122.223413299999		
	147	bobateani-san-jc	5	75 E Santa Clar	San Jose	37.33709-121.88941		
	128	qteabar-oakland	5	478 Lake Park A	Oakland	37.8110686341717-122.24723573774		
	89	puppy-bobar-sar	5	1142 Grant Ave	San Francisco	37.7975399525428-122.406789958477		



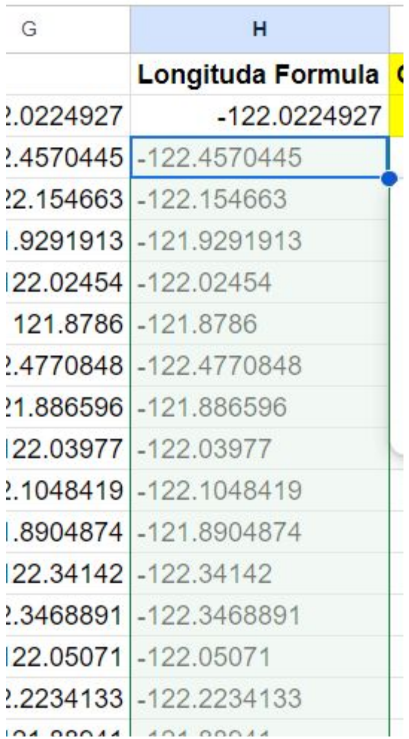
G	H
Lat	Long
37.58956283	122.0224927
37.72429542	122.4570445
37.723825	122.154663
37.48856826	121.9291913
37.36189	122.02454
37.65223	121.8786
37.78029568	122.4770848
37.335455	121.886596
37.5757	122.03977
37.6542332	122.1048419
38.0136496	121.8904874
37.590323	122.34142
37.5801206	122.3468891
37.35338	122.05071

Formulas



	G	H
	Long	ula
83	122.0224927	=G2*-1
42	122.4570445	
25	122.154663	
26	121.9291913	
89	122.02454	
23	121.8786	
68	122.4770848	
55	121.886596	
57	122.03977	
32	122.1048419	
96	121.8904874	
23	122.34142	
06	122.3468891	
38	122.05071	

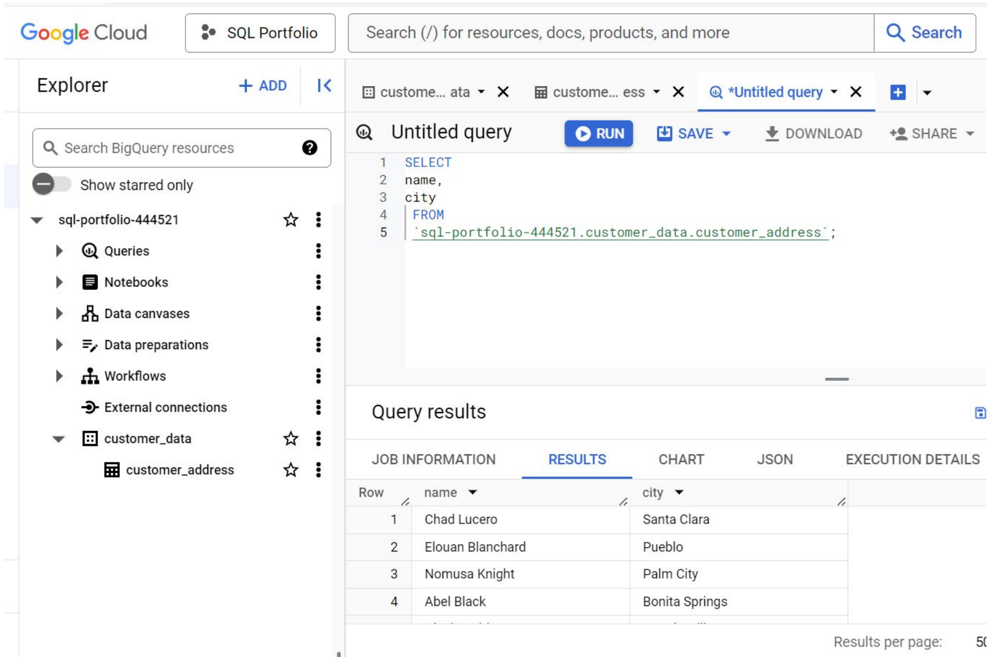
-122.0224927 ×



G	H
	Longituda Formula
.0224927	-122.0224927
.4570445	-122.4570445
2.154663	-122.154663
.9291913	-121.9291913
22.02454	-122.02454
121.8786	-121.8786
.4770848	-122.4770848
1.886596	-121.886596
22.03977	-122.03977
.1048419	-122.1048419
.8904874	-121.8904874
22.34142	-122.34142
.3468891	-122.3468891
22.05071	-122.05071
.2234133	-122.2234133

- Longitude values should be negative so that they are accurate coordinates for mapping.
- **`=G2*-1`**

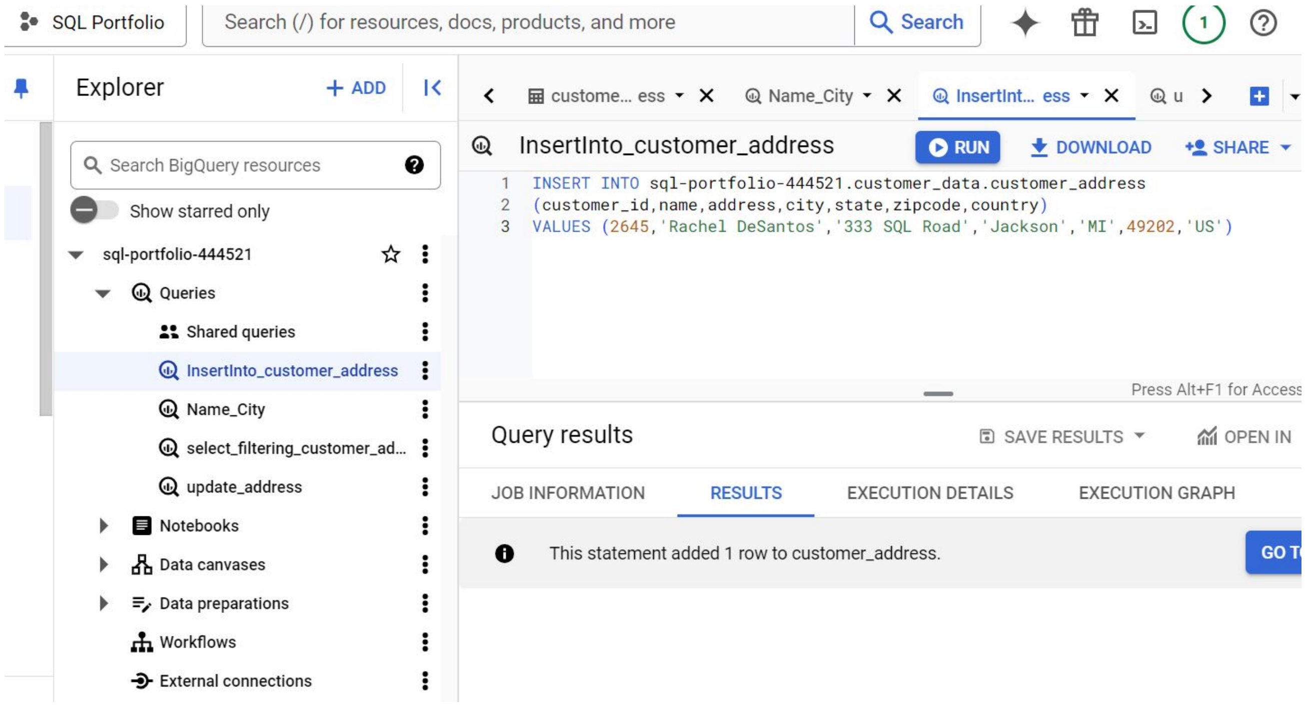
SQL - Common Queries



```sql
SELECT
name,
city
 FROM
 `sql-portfolio-444521.customer_data.customer_address`;
```

Row	name	city
1	Chad Lucero	Santa Clara
2	Elouan Blanchard	Pueblo
3	Nomusa Knight	Palm City
4	Abel Black	Bonita Springs

Identifies variable within a specific row


SQL Portfolio

Search (/) for resources, docs, products, and more

Explorer

Search BigQuery resources

Show starred only

sql-portfolio-444521
- Queries
  - Shared queries
  - InsertInto_customer_address
  - Name_City
  - select_filtering_customer_ad...
  - update_address
- Notebooks
- Data canvases
- Data preparations
- Workflows
- External connections

InsertInto_customer_address

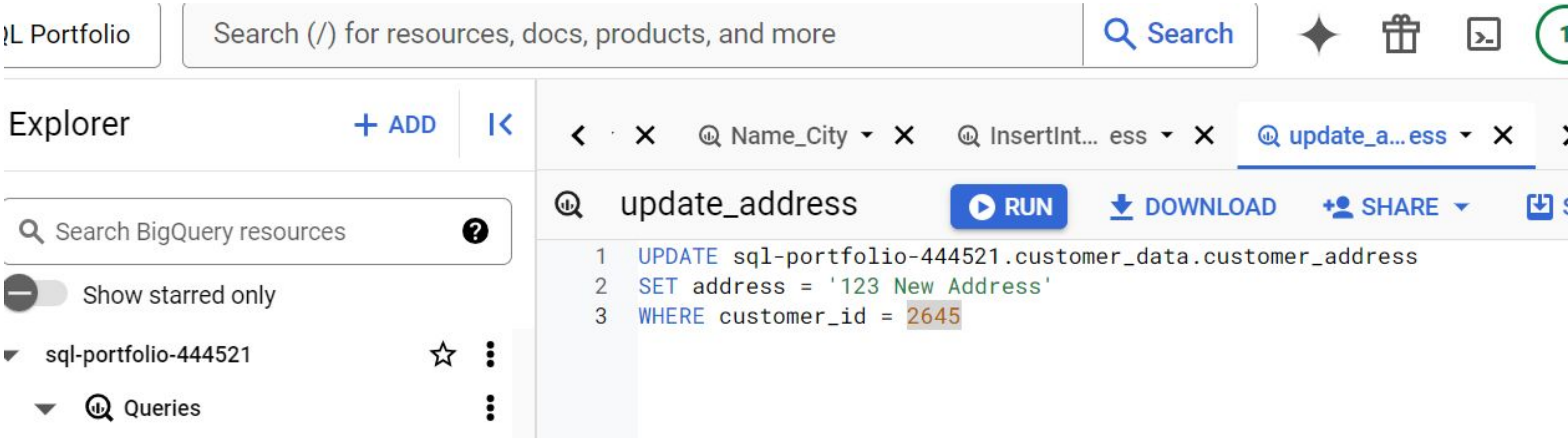
```sql
INSERT INTO sql-portfolio-444521.customer_data.customer_address
(customer_id,name,address,city,state,zipcode,country)
VALUES (2645,'Rachel DeSantos','333 SQL Road','Jackson','MI',49202,'US')
```

Query results

JOB INFORMATION | RESULTS | EXECUTION DETAILS | EXECUTION GRAPH

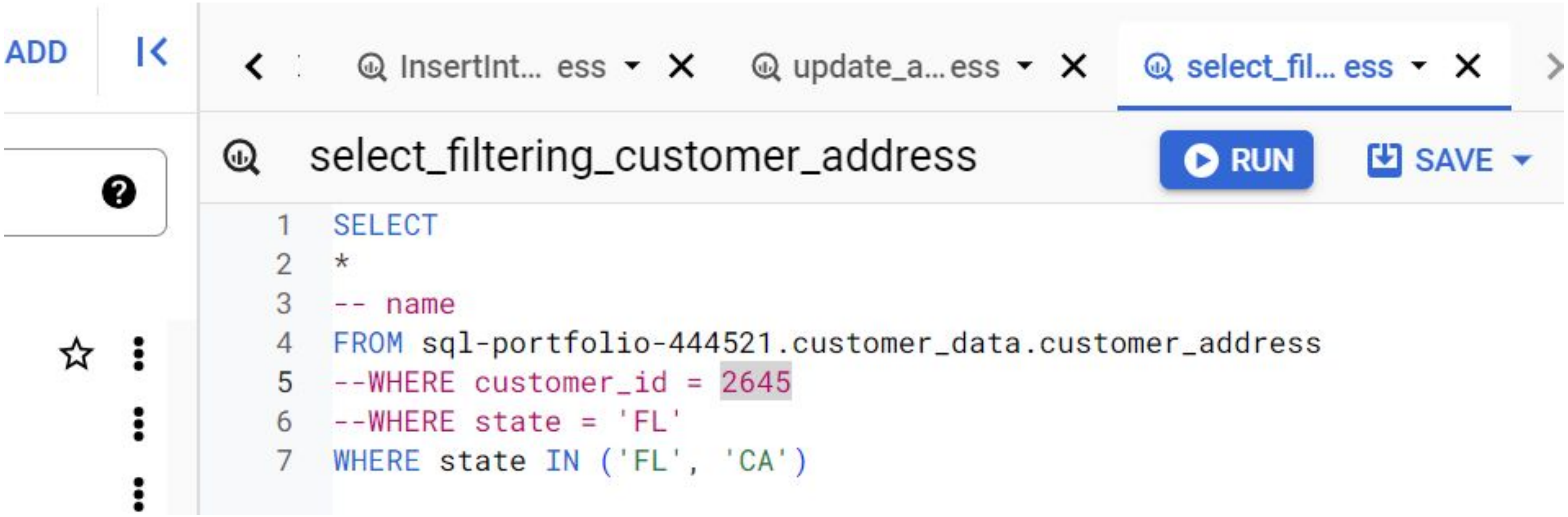
This statement added 1 row to customer_address.

Updating a variable within a row



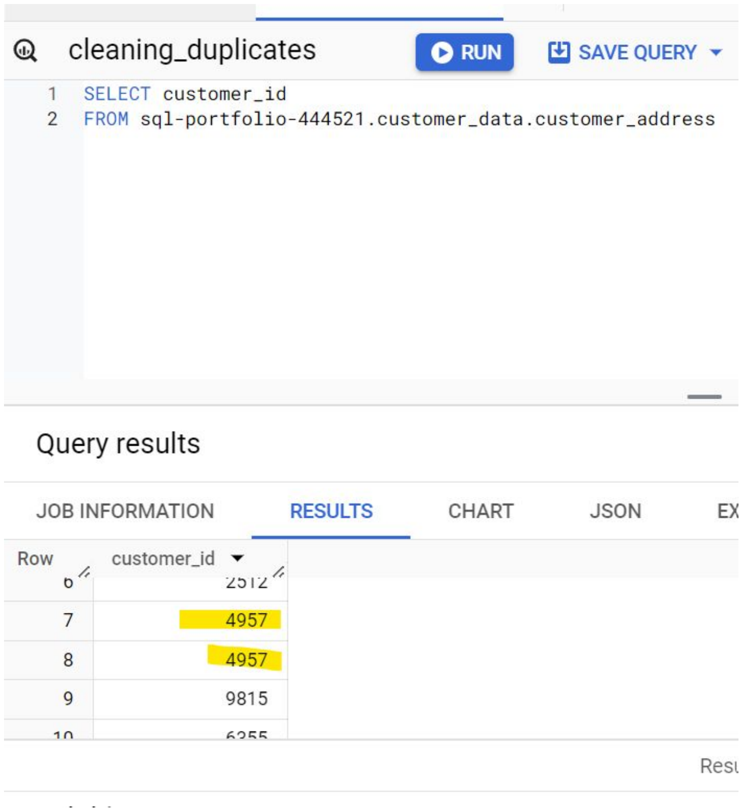
```
UPDATE sql-portfolio-444521.customer_data.customer_address
SET address = '123 New Address'
WHERE customer_id = 2645
```

SQL-Filtering



```
SELECT
*
-- name
FROM sql-portfolio-444521.customer_data.customer_address
--WHERE customer_id = 2645
--WHERE state = 'FL'
WHERE state IN ('FL', 'CA')
```

Cleaning String Variables - removing duplicates



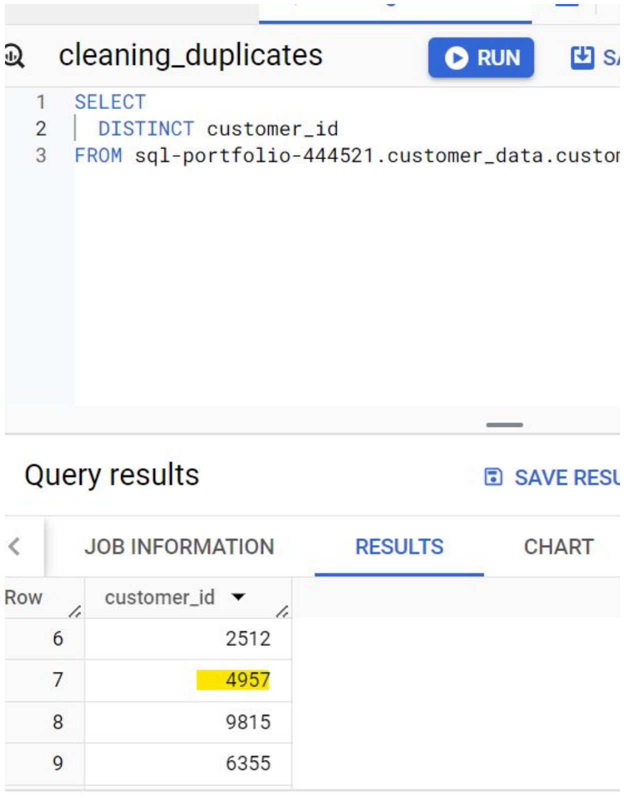
cleaning_duplicates

```
SELECT customer_id
FROM sql-portfolio-444521.customer_data.customer_address
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EX

Row	customer_id
6	2512
7	4957
8	4957
9	9815
10	6355



cleaning_duplicates

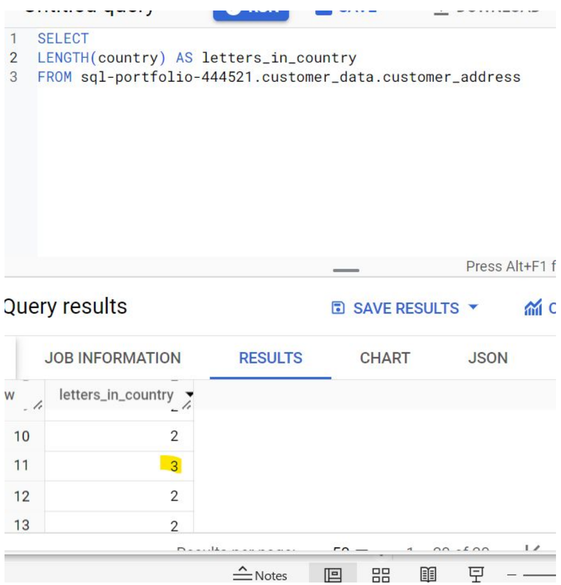
```
SELECT
  DISTINCT customer_id
FROM sql-portfolio-444521.customer_data.custor
```

Query results

JOB INFORMATION | RESULTS | CHART

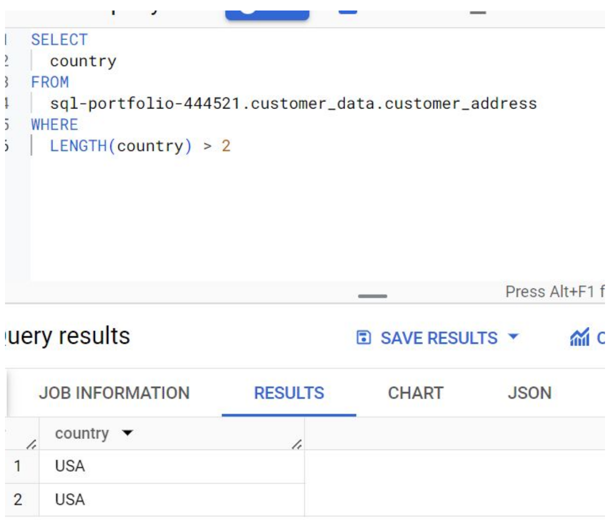
Row	customer_id
6	2512
7	4957
8	9815
9	6355

Functions - LENGTH / DISTINCT / SUBSTR



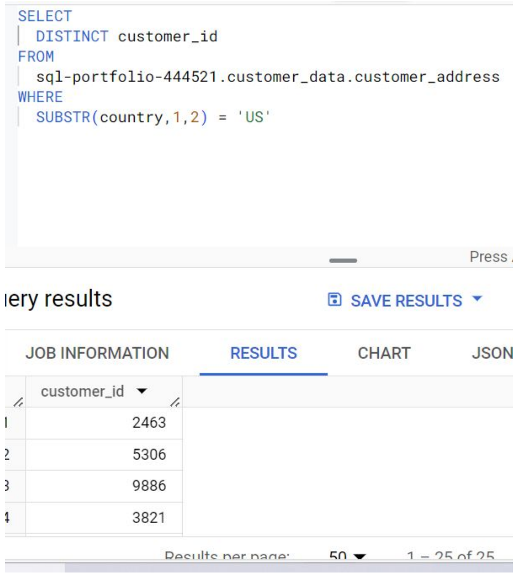
```sql
SELECT
LENGTH(country) AS letters_in_country
FROM sql-portfolio-444521.customer_data.customer_address
```

w	letters_in_country
10	2
11	3
12	2
13	2



```sql
SELECT
  country
FROM
  sql-portfolio-444521.customer_data.customer_address
WHERE
  LENGTH(country) > 2
```

	country
1	USA
2	USA

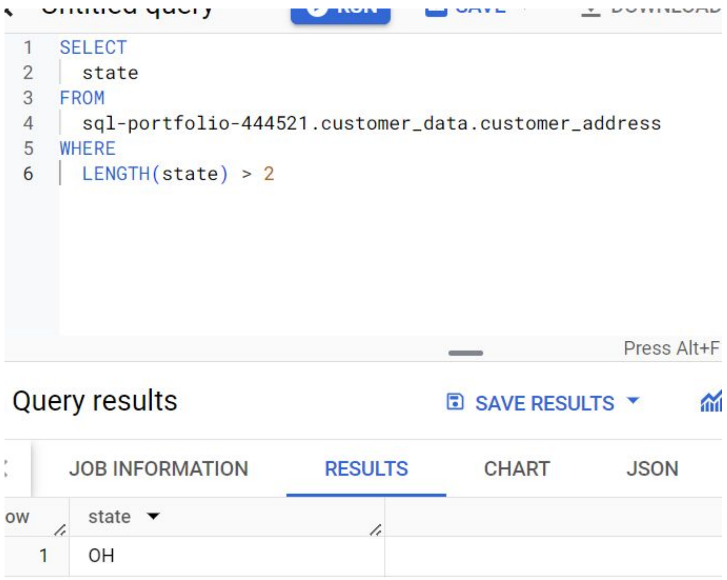


```sql
SELECT
  DISTINCT customer_id
FROM
  sql-portfolio-444521.customer_data.customer_address
WHERE
  SUBSTR(country,1,2) = 'US'
```

	customer_id
1	2463
2	5306
3	9886
4	3821

TRIM()

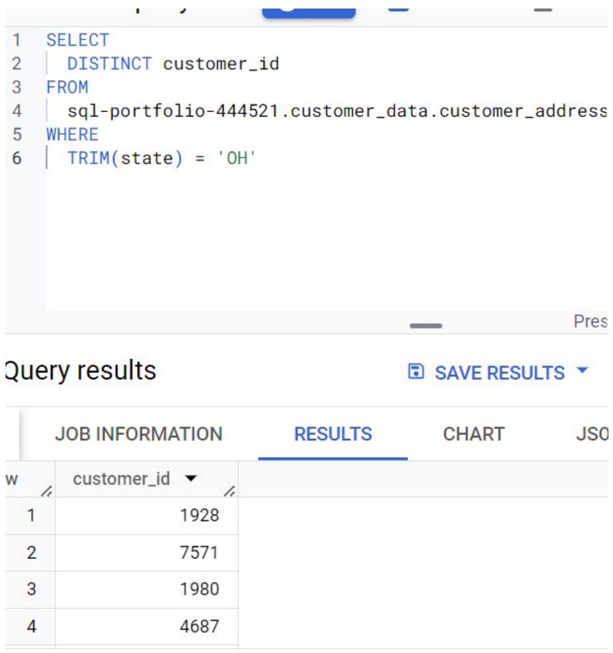
OH is not greater than 2 characters, so using the TRIM function will remove any spaces



```
SELECT
  state
FROM
  sql-portfolio-444521.customer_data.customer_address
WHERE
  LENGTH(state) > 2
```

Query results

Row	state
1	OH



```
SELECT
  DISTINCT customer_id
FROM
  sql-portfolio-444521.customer_data.customer_address
WHERE
  TRIM(state) = 'OH'
```

Query results

Row	customer_id
1	1928
2	7571
3	1980
4	4687

Dirty to Clean Data - SQL Project

Dataset: Automobiles

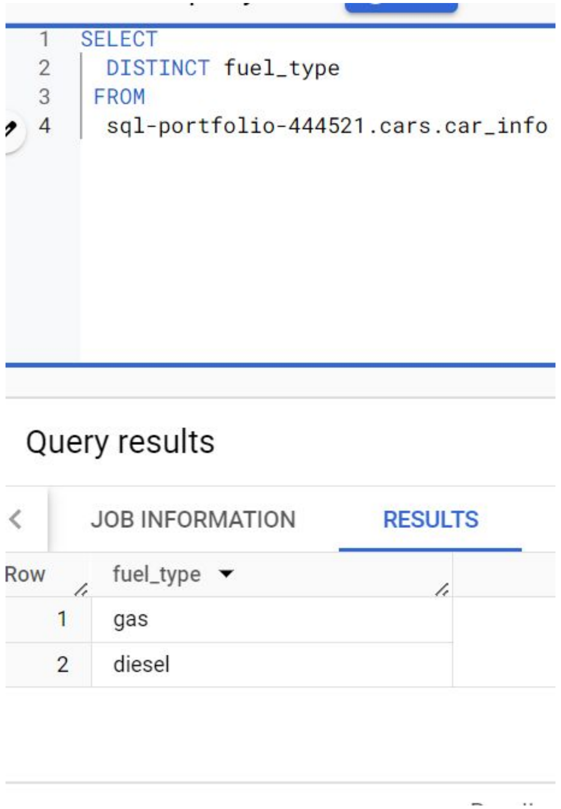
https://docs.google.com/spreadsheets/d/1uc563psJrCU6Gf0ftaWLXKXAF7czdONsKnaCgC5bz9c/edit?usp=sharing

Contains historical sales data, including details such as car features and prices. You can use this data to find the top 10 most popular cars and trims

Goal of project: clean data to avoid presenting inaccurate inventory, which could result in the company to lose money on their car investment

Automobile - UCI Machine Learning Repository - verifies that the fuel_type column should contain only two unique string values 'diesel' and 'gas'.

See screenshot to verify fuel_type has the two unique strings.



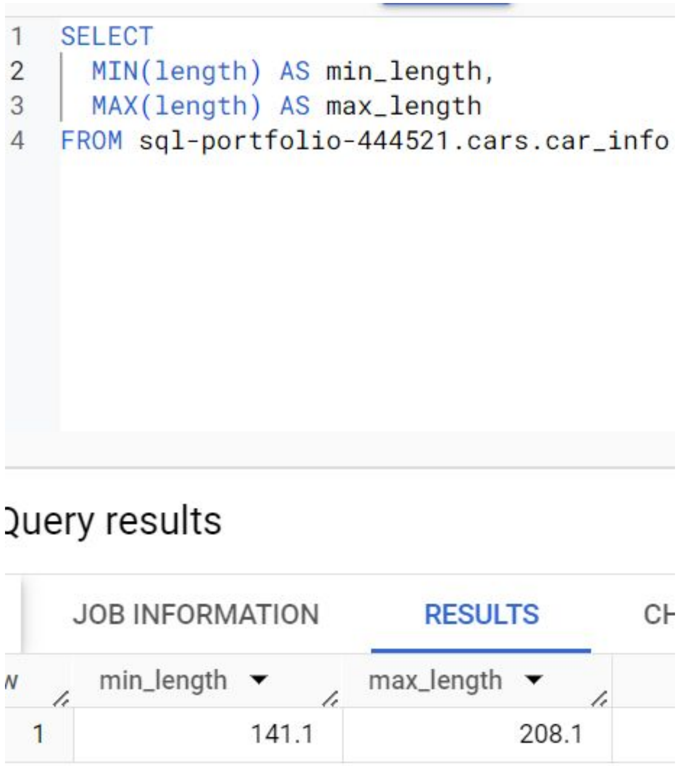
```
SELECT
  DISTINCT fuel_type
FROM
  sql-portfolio-444521.cars.car_info
```

Query results

JOB INFORMATION RESULTS

Row	fuel_type
1	gas
2	diesel

MAX/MIN Functions



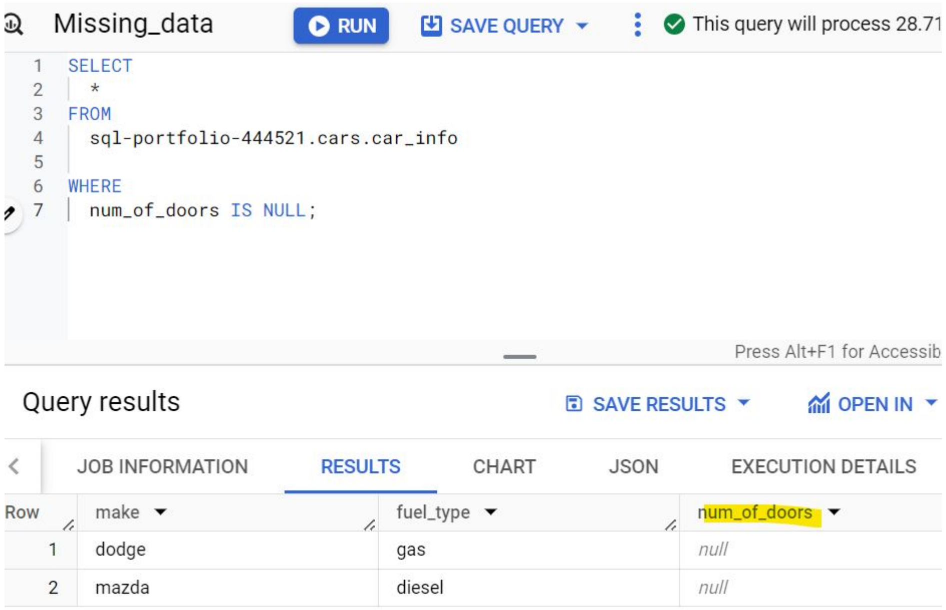
```
SELECT
  MIN(length) AS min_length,
  MAX(length) AS max_length
FROM sql-portfolio-444521.cars.car_info
```

	min_length	max_length
1	141.1	208.1

Within the automobiles dataset, I wanted to know the (max/min) within the length column. When the query returned the values, I asked to make two new columns for the entries to fall into.

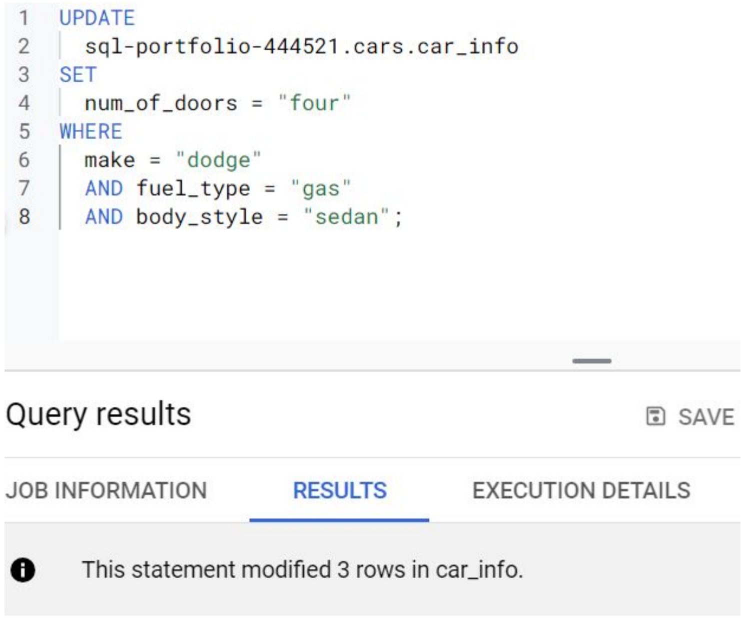
SQL - Missing Data

ALWAYS check data for null or missing values. Confirm replace with team or data owner and make correction.



```sql
SELECT
  *
FROM
  sql-portfolio-444521.cars.car_info

WHERE
  num_of_doors IS NULL;
```

Row	make	fuel_type	num_of_doors
1	dodge	gas	*null*
2	mazda	diesel	*null*

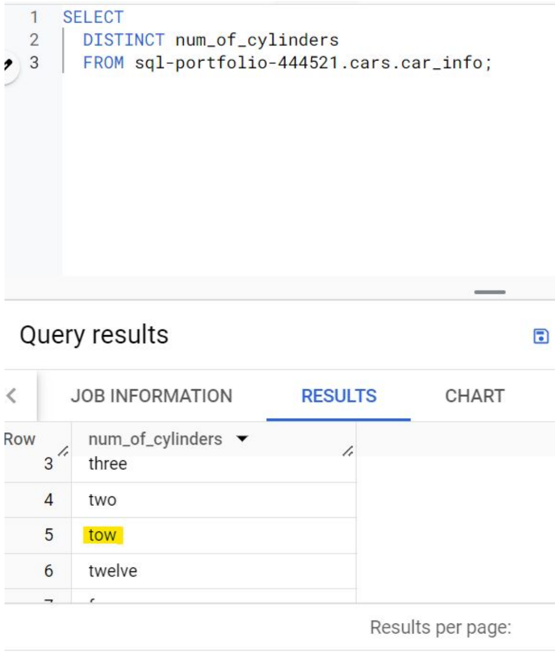


```sql
UPDATE
  sql-portfolio-444521.cars.car_info
SET
  num_of_doors = "four"
WHERE
  make = "dodge"
  AND fuel_type = "gas"
  AND body_style = "sedan";
```

This statement modified 3 rows in car_info.

SQL - Correcting Misspelling

‘Tow’ is now corrected to ‘two’. The DISTINCT function only picks up ‘two’.



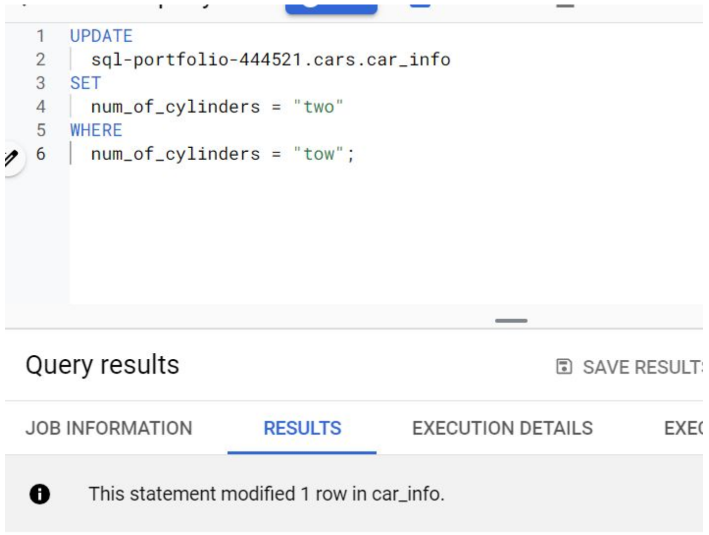
```
SELECT
  DISTINCT num_of_cylinders
  FROM sql-portfolio-444521.cars.car_info;
```

Query results

JOB INFORMATION | RESULTS | CHART

Row	num_of_cylinders
3	three
4	two
5	tow
6	twelve

Results per page:



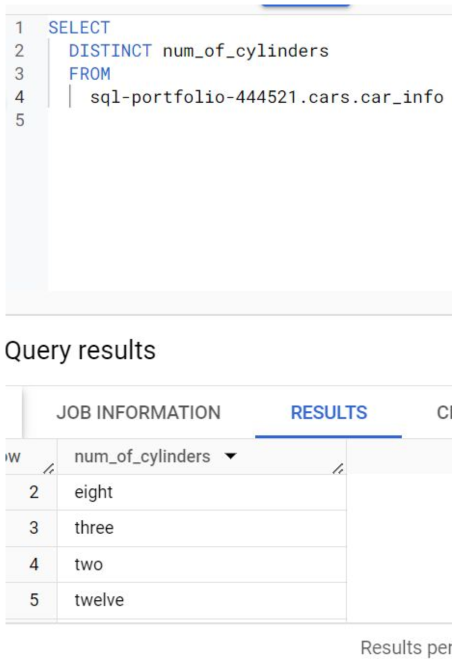
```
UPDATE
  sql-portfolio-444521.cars.car_info
SET
  num_of_cylinders = "two"
WHERE
  num_of_cylinders = "tow";
```

Query results

SAVE RESULT

JOB INFORMATION | RESULTS | EXECUTION DETAILS | EXE

This statement modified 1 row in car_info.



```
SELECT
  DISTINCT num_of_cylinders
  FROM
    sql-portfolio-444521.cars.car_info
```

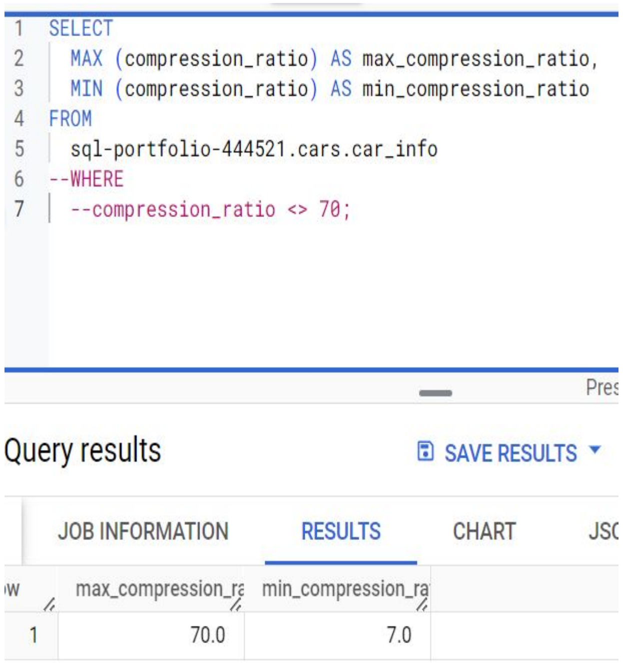
Query results

JOB INFORMATION | RESULTS | C

w	num_of_cylinders
2	eight
3	three
4	two
5	twelve

Results per

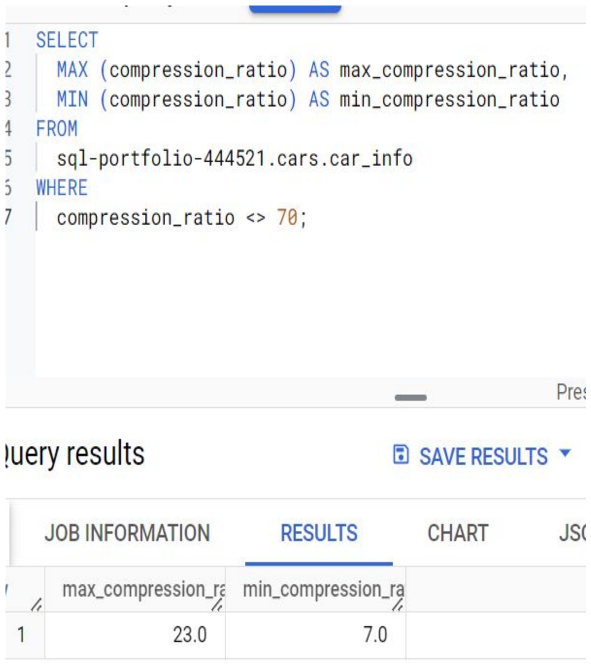
MAX/MIN ranges (removing/deleting <> outliers)



```sql
SELECT
  MAX (compression_ratio) AS max_compression_ratio,
  MIN (compression_ratio) AS min_compression_ratio
FROM
  sql-portfolio-444521.cars.car_info
--WHERE
  --compression_ratio <> 70;
```

Query results

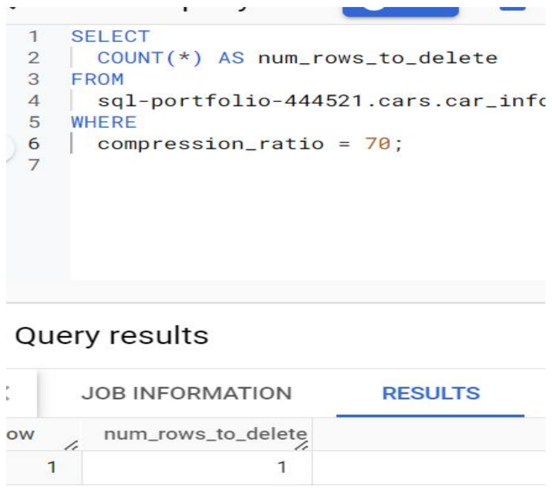
max_compression_ra	min_compression_ra
70.0	7.0



```sql
SELECT
  MAX (compression_ratio) AS max_compression_ratio,
  MIN (compression_ratio) AS min_compression_ratio
FROM
  sql-portfolio-444521.cars.car_info
WHERE
  compression_ratio <> 70;
```

Query results

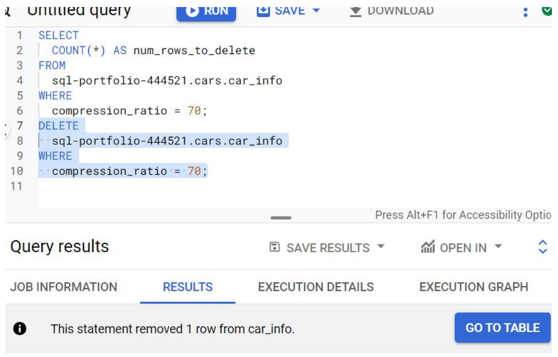
max_compression_ra	min_compression_ra
23.0	7.0



```sql
SELECT
  COUNT(*) AS num_rows_to_delete
FROM
  sql-portfolio-444521.cars.car_info
WHERE
  compression_ratio = 70;
```

Query results

num_rows_to_delete
1

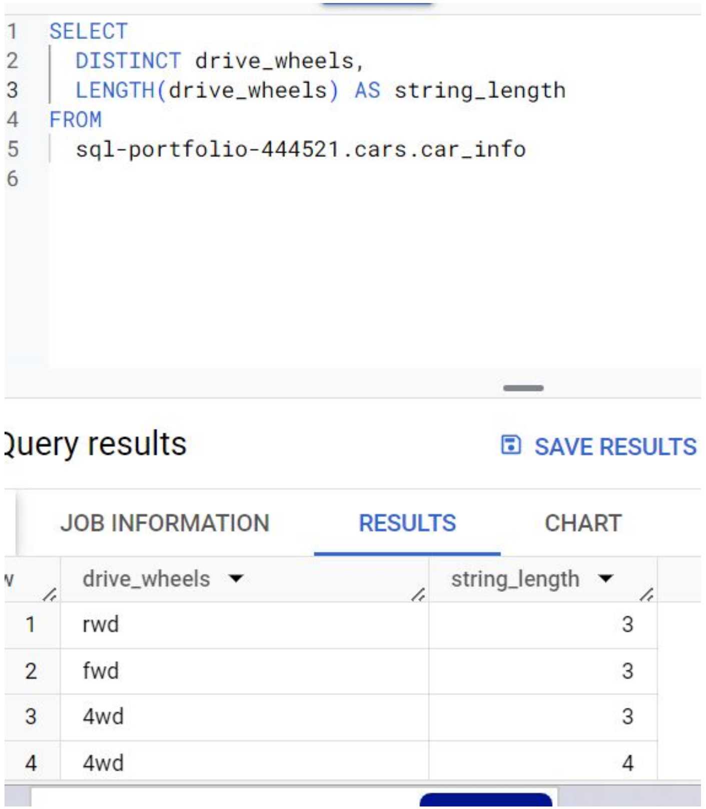


```sql
SELECT
  COUNT(*) AS num_rows_to_delete
FROM
  sql-portfolio-444521.cars.car_info
WHERE
  compression_ratio = 70;
DELETE
  sql-portfolio-444521.cars.car_info
WHERE
  compression_ratio = 70;
```

Query results

This statement removed 1 row from car_info.

SQL - Multiple Functions



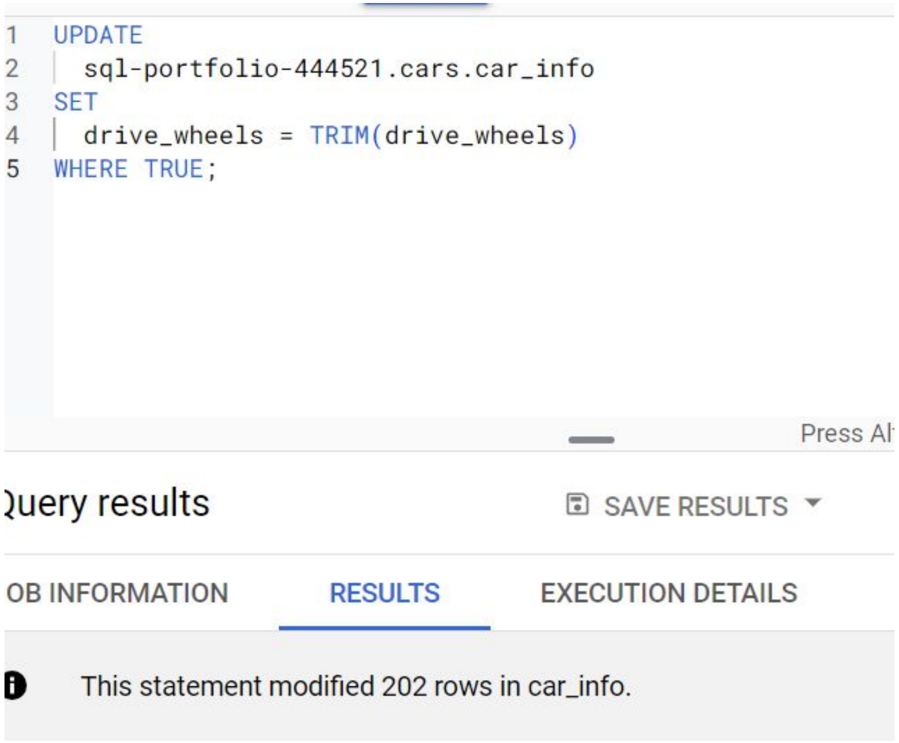
```
SELECT
  DISTINCT drive_wheels,
  LENGTH(drive_wheels) AS string_length
FROM
  sql-portfolio-444521.cars.car_info

```

Query results

SAVE RESULTS

JOB INFORMATION RESULTS CHART

	drive_wheels	string_length
1	rwd	3
2	fwd	3
3	4wd	3
4	4wd	4



```
UPDATE
  sql-portfolio-444521.cars.car_info
SET
  drive_wheels = TRIM(drive_wheels)
WHERE TRUE;
```

Press Al

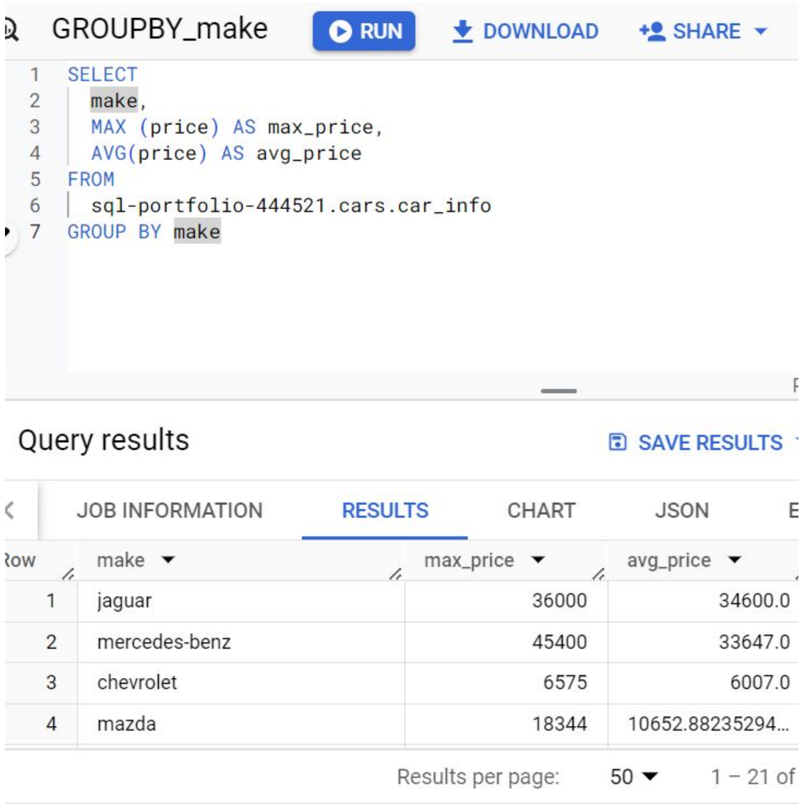
Query results

SAVE RESULTS

OB INFORMATION RESULTS EXECUTION DETAILS

This statement modified 202 rows in car_info.

GROUP BY



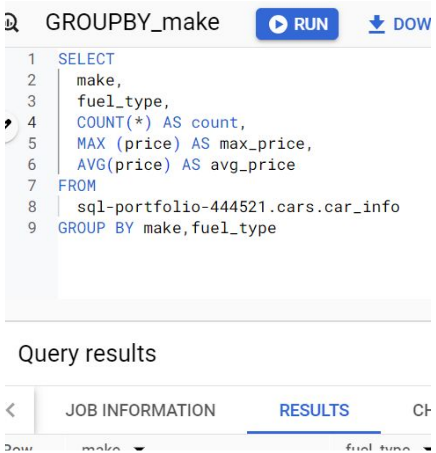
```
SELECT
  make,
  MAX (price) AS max_price,
  AVG(price) AS avg_price
FROM
  sql-portfolio-444521.cars.car_info
GROUP BY make
```

Query results

Row	make	max_price	avg_price
1	jaguar	36000	34600.0
2	mercedes-benz	45400	33647.0
3	chevrolet	6575	6007.0
4	mazda	18344	10652.88235294...

Results per page: 50 1 – 21 of

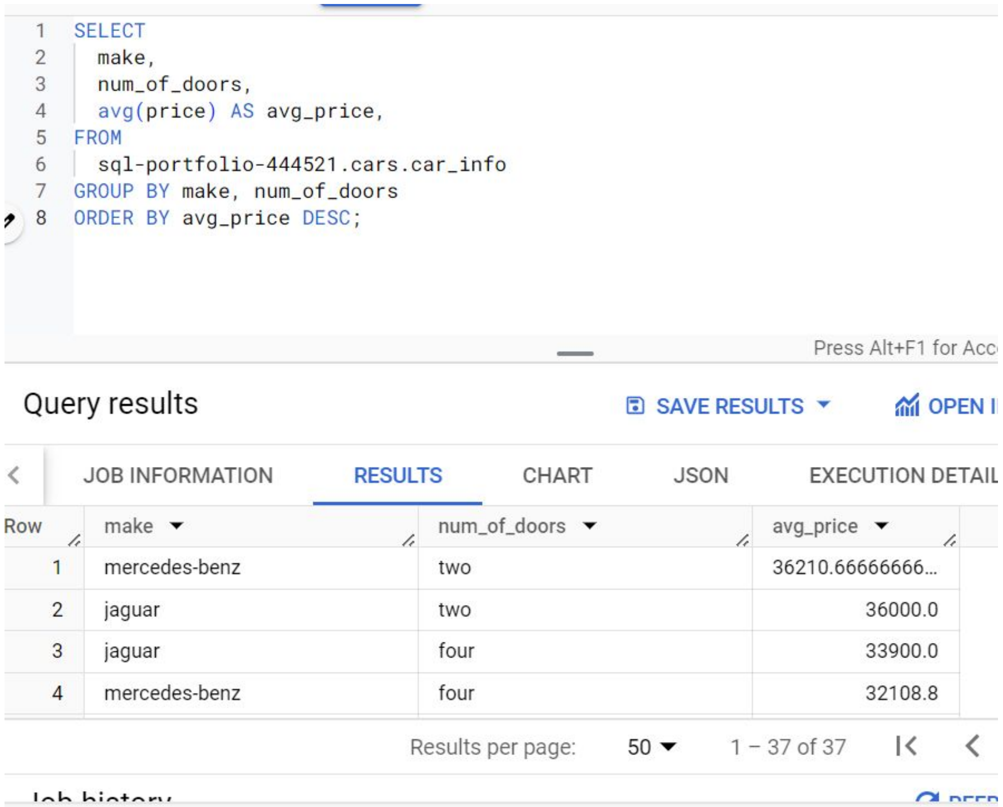
- Applying the GROUPBY to the non-aggregated function for MAX price to align the make with MAX/AVG prices.



```
SELECT
  make,
  fuel_type,
  COUNT(*) AS count,
  MAX (price) AS max_price,
  AVG(price) AS avg_price
FROM
  sql-portfolio-444521.cars.car_info
GROUP BY make,fuel_type
```

Query results

ORDER BY



```
SELECT
  make,
  num_of_doors,
  avg(price) AS avg_price,
FROM
  sql-portfolio-444521.cars.car_info
GROUP BY make, num_of_doors
ORDER BY avg_price DESC;
```

Press Alt+F1 for Acc

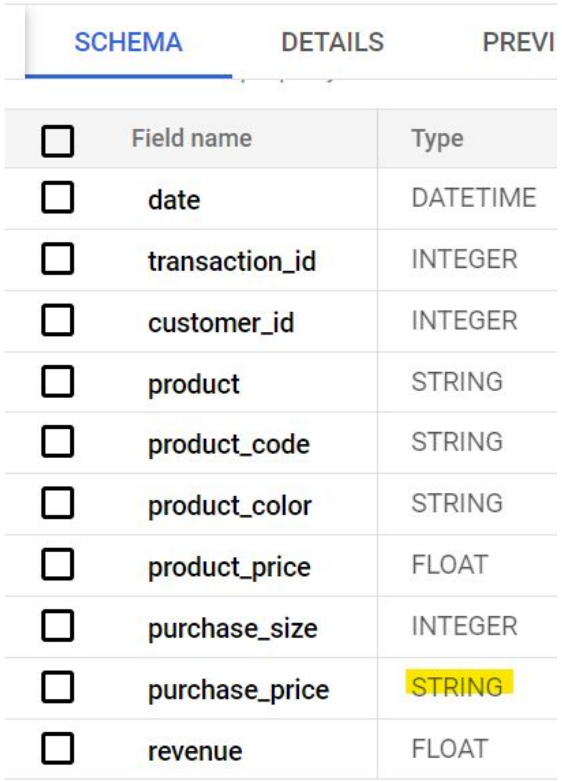
Query results

Row	make	num_of_doors	avg_price
1	mercedes-benz	two	36210.66666666...
2	jaguar	two	36000.0
3	jaguar	four	33900.0
4	mercedes-benz	four	32108.8

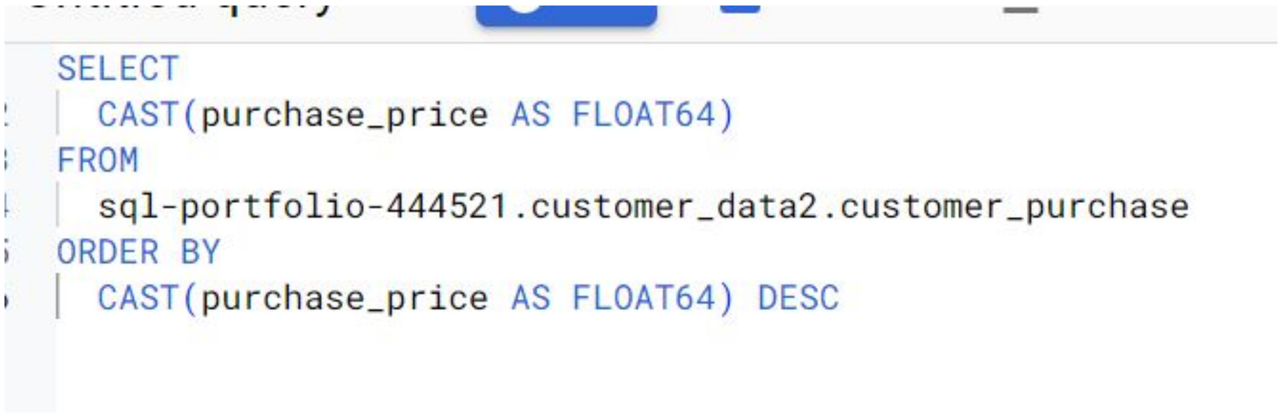
Results per page: 50 1 – 37 of 37

Advanced Function - CAST()

- String -> Float conversion.
- Verify change through SCHEMA.
- We needed to determine the customer purchase price from greatest to least, but the numbers were being ordered by strings not numerical values. By changing to FLOAT, we can accurately order the values.

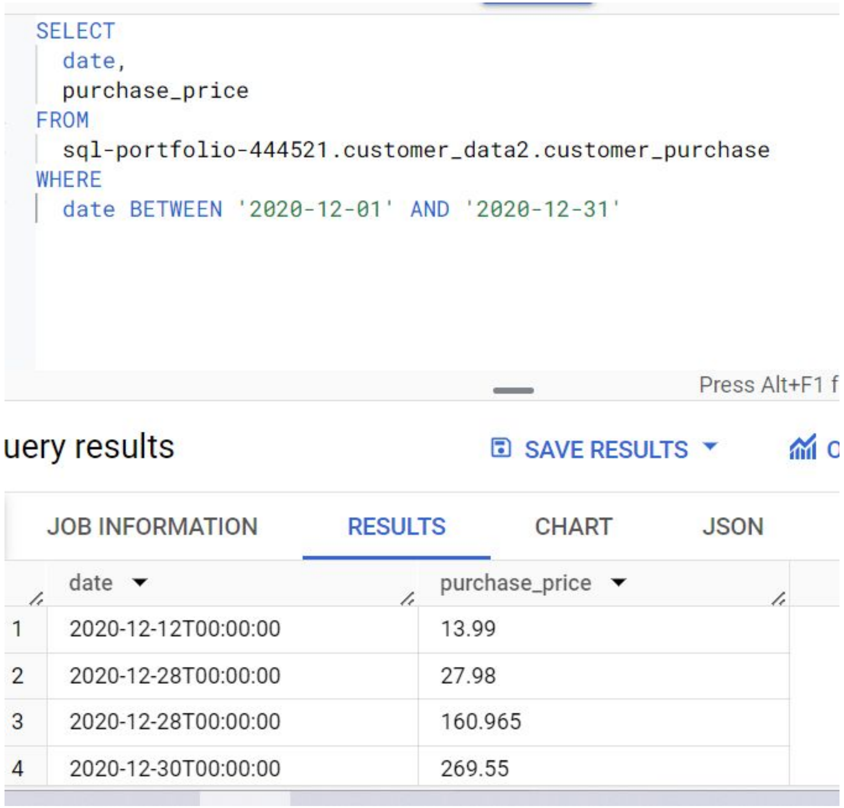


	Field name	Type
☐	date	DATETIME
☐	transaction_id	INTEGER
☐	customer_id	INTEGER
☐	product	STRING
☐	product_code	STRING
☐	product_color	STRING
☐	product_price	FLOAT
☐	purchase_size	INTEGER
☐	purchase_price	STRING
☐	revenue	FLOAT



```
SELECT
  CAST(purchase_price AS FLOAT64)
FROM
  sql-portfolio-444521.customer_data2.customer_purchase
ORDER BY
  CAST(purchase_price AS FLOAT64) DESC
```

Filtering Date Periods (Correcting String -> Date)



```sql
SELECT
  date,
  purchase_price
FROM
  sql-portfolio-444521.customer_data2.customer_purchase
WHERE
  date BETWEEN '2020-12-01' AND '2020-12-31'
```

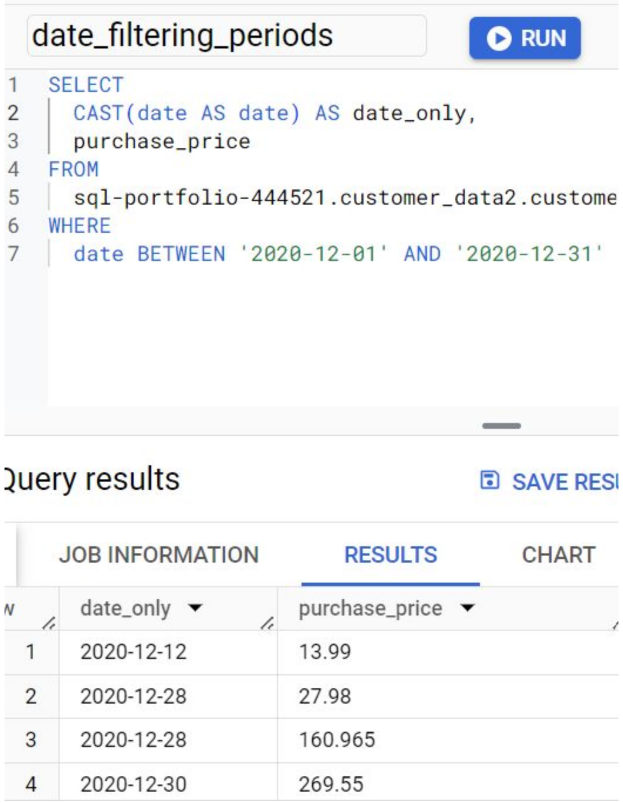
Press Alt+F1 f

uery results

SAVE RESULTS

JOB INFORMATION | RESULTS | CHART | JSON

	date	purchase_price
1	2020-12-12T00:00:00	13.99
2	2020-12-28T00:00:00	27.98
3	2020-12-28T00:00:00	160.965
4	2020-12-30T00:00:00	269.55



date_filtering_periods

RUN

```sql
SELECT
  CAST(date AS date) AS date_only,
  purchase_price
FROM
  sql-portfolio-444521.customer_data2.custome
WHERE
  date BETWEEN '2020-12-01' AND '2020-12-31'
```

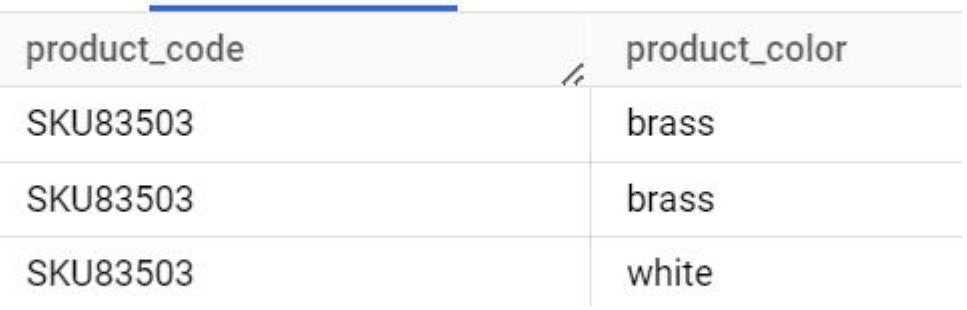
Query results

SAVE RESI

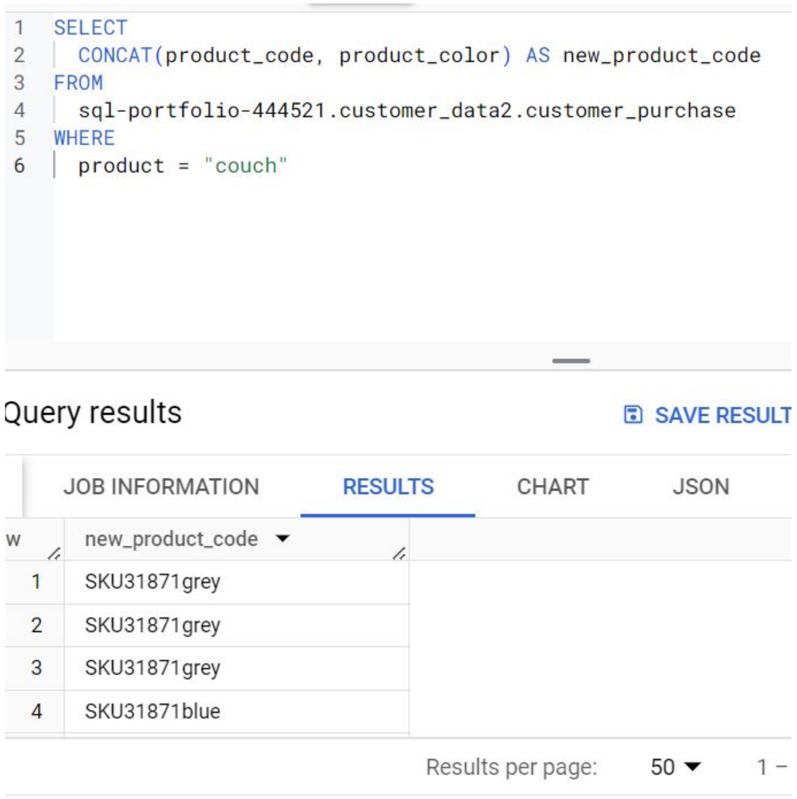
JOB INFORMATION | RESULTS | CHART

	date_only	purchase_price
1	2020-12-12	13.99
2	2020-12-28	27.98
3	2020-12-28	160.965
4	2020-12-30	269.55

CONCAT()



product_code	product_color
SKU83503	brass
SKU83503	brass
SKU83503	white

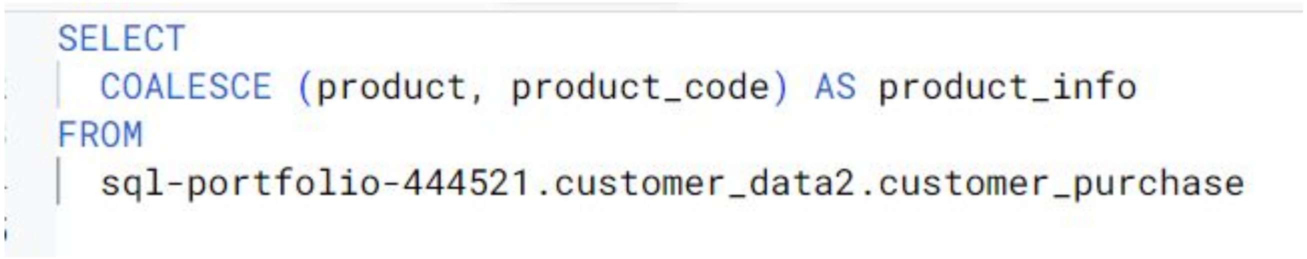


```
SELECT
  CONCAT(product_code, product_color) AS new_product_code
FROM
  sql-portfolio-444521.customer_data2.customer_purchase
WHERE
  product = "couch"
```

Query results

w	new_product_code
1	SKU31871grey
2	SKU31871grey
3	SKU31871grey
4	SKU31871blue

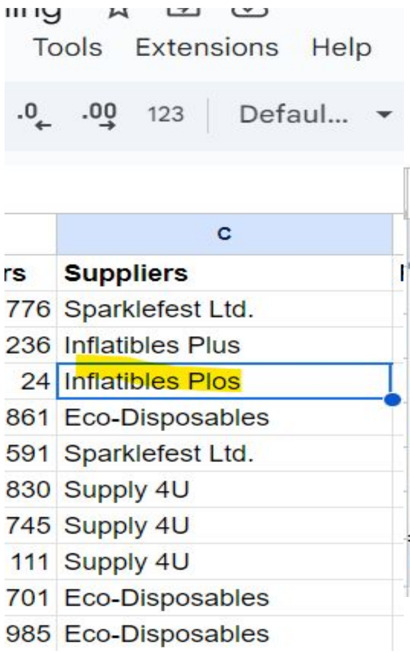
Advanced Function - COALESCE ()



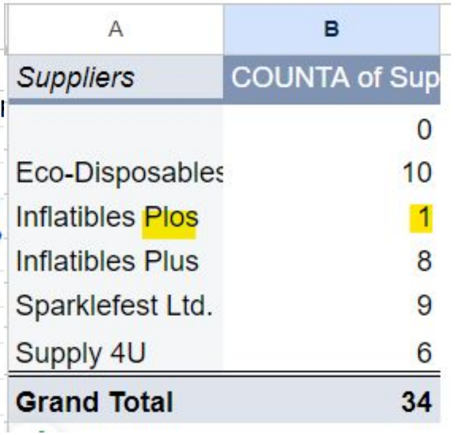
```sql
SELECT
  COALESCE (product, product_code) AS product_info
FROM
  sql-portfolio-444521.customer_data2.customer_purchase
```

- This function replaces NULL’s with another identifier in the column you are trying to calculate total. This would be a placeholder for another field that shows as NULL.

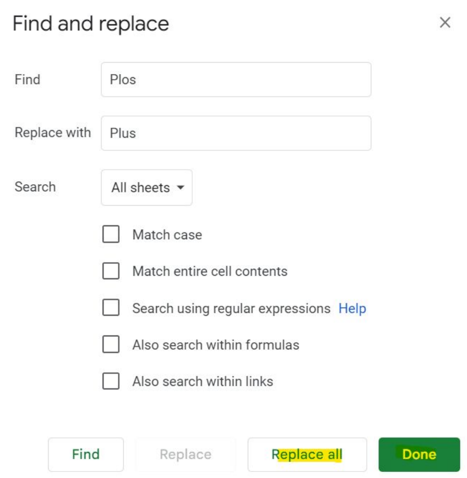
Verifying Data is Cleaned - Excel/Google Sheets Using Pivot Tables/ Find & Replace



	C
rs	Suppliers
776	Sparklefest Ltd.
236	Inflatibles Plus
24	Inflatibles Plos
861	Eco-Disposables
591	Sparklefest Ltd.
830	Supply 4U
745	Supply 4U
111	Supply 4U
701	Eco-Disposables
985	Eco-Disposables



A	B
Suppliers	COUNTA of Sup
	0
Eco-Disposables	10
Inflatibles Plos	1
Inflatibles Plus	8
Sparklefest Ltd.	9
Supply 4U	6
Grand Total	**34**



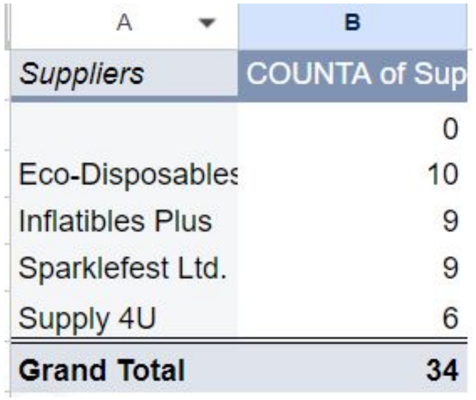
Find and replace

Find: Plos

Replace with: Plus

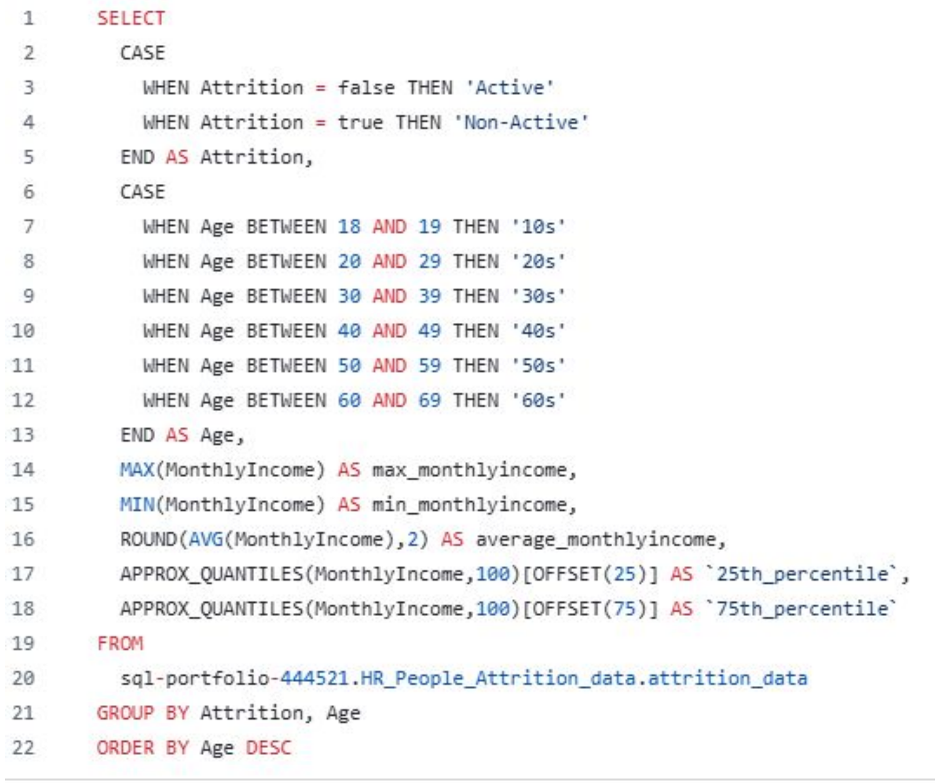
Search: All sheets

- [ ] Match case
- [ ] Match entire cell contents
- [ ] Search using regular expressions Help
- [ ] Also search within formulas
- [ ] Also search within links



A	B
Suppliers	COUNTA of Sup
	0
Eco-Disposables	10
Inflatibles Plus	9
Sparklefest Ltd.	9
Supply 4U	6
Grand Total	**34**

Verifying Data is Cleaning Using SQL CASE()



```sql
SELECT
  CASE
    WHEN Attrition = false THEN 'Active'
    WHEN Attrition = true THEN 'Non-Active'
  END AS Attrition,
  CASE
    WHEN Age BETWEEN 18 AND 19 THEN '10s'
    WHEN Age BETWEEN 20 AND 29 THEN '20s'
    WHEN Age BETWEEN 30 AND 39 THEN '30s'
    WHEN Age BETWEEN 40 AND 49 THEN '40s'
    WHEN Age BETWEEN 50 AND 59 THEN '50s'
    WHEN Age BETWEEN 60 AND 69 THEN '60s'
  END AS Age,
  MAX(MonthlyIncome) AS max_monthlyincome,
  MIN(MonthlyIncome) AS min_monthlyincome,
  ROUND(AVG(MonthlyIncome),2) AS average_monthlyincome,
  APPROX_QUANTILES(MonthlyIncome,100)[OFFSET(25)] AS `25th_percentile`,
  APPROX_QUANTILES(MonthlyIncome,100)[OFFSET(75)] AS `75th_percentile`
FROM
  sql-portfolio-444521.HR_People_Attrition_data.attrition_data
GROUP BY Attrition, Age
ORDER BY Age DESC
```