This is a demonstration of different calculations and functions with US Census information. The table to be analyzed is 2019 census estimates. The first calculation performed will be stored under a column titled “natural_increase”, showing the difference between births and deaths recorded in counties. Please note the “cen” before the table name; this merely refers to the database used in GCP.

```
SELECT county_name AS county,
    state_name AS state,
    births_2019 AS births,
    deaths_2019 AS deaths,
    births_2019 - deaths_2019 AS natural_increase
FROM cen.us_counties_pop_est_2019
ORDER BY state_name, county_name;
```

Row	county	state	births	deaths	natural_increase
1	Autauga County	Alabama	624	541	83
2	Baldwin County	Alabama	2304	2326	-22
3	Barbour County	Alabama	256	312	-56
4	Bibb County	Alabama	240	252	-12
5	Blount County	Alabama	651	657	-6
6	Bullock County	Alabama	109	109	0

I wanted to confirm that the information given was correct in order to proceed. I combined components to population differences in one column (“components_total”). I also created a column (“difference”) to confirm the 2018 estimate and components combined should equal the 2019 estimates. If the information is correct, all values in “difference” should be 0.

```
SELECT county_name AS county,
    state_name AS state,
    pop_est_2019 AS pop,
    pop_est_2018 + births_2019 - deaths_2019 +
      international_migr_2019 + domestic_migr_2019 +
      residual_2019 AS components_total,
    pop_est_2019 - (pop_est_2018 + births_2019 - deaths_2019 +
      international_migr_2019 + domestic_migr_2019 +
```

```
        residual_2019) AS difference
FROM cen.us_counties_pop_est_2019
ORDER BY difference DESC;
```

Row	county	state	pop	components_total	difference
1	Dallas County	Iowa	93453	93453	0
2	Linn County	Iowa	226706	226706	0
3	Calhoun County	Iowa	9668	9668	0
4	Carroll County	Iowa	20165	20165	0
5	Chickasaw County	Iowa	11933	11933	0
6	Davis County	Iowa	9000	9000	0

One calculation I performed was the percentage of a county's area that was water. After executing the code, I researched whether or not the calculations were accurate, which they were – plus having been to Nantucket before, I knew the ocean surrounding it is considered part of the county.

```
SELECT county_name AS county,
    state_name AS state,
    area_water / (area_land + area_water) * 100 AS pct_water
FROM cen.us_counties_pop_est_2019
ORDER BY pct_water DESC;
```

Row	county	state	pct_water
1	Keweenaw County	Michigan	90.947237474532145
2	Leelanau County	Michigan	86.288589681165831
3	Nantucket County	Massachusetts	84.796924991855121
4	St. Bernard Parish	Louisiana	82.483711492028945
5	Alger County	Michigan	81.872219406475011
6	Poquoson city	Virginia	80.403894026577589

I wanted to practice using aggregate functions. By using aggregate functions, I was able to calculate the sum of the estimated population of all counties in the country as well as the average population of all counties in the country.

```
SELECT sum(pop_est_2019) AS county_sum,
    round(avg(pop_est_2019), 0) AS county_average
FROM cen.us_counties_pop_est_2019;
```

Row	county_sum	county_average
1	328239523	104468.0

I then calculated the median county population. By adding the percentile_cont() command, I was able to calculate the median county population for the country as 25,726.

```
SELECT sum(pop_est_2019) AS county_sum,
    round(avg(pop_est_2019), 0) AS county_average,
    percentile_cont(.5)
    WITHIN GROUP (ORDER BY pop_est_2019) AS county_median
FROM cen.us_counties_pop_est_2019;
```