GPA Analysis Project notes

I want to learn more efficient ways to connect bigquery to the sheets data, so that it will update as the data does. The way I set it up was with a set range from each sheet. If existing cells are altered it updated, but what if new rows are added?

tried to use Alter Table to add new column called grade_points, but kept getting error "grade_points is not a supported nested object type at [1:52]". Googling didn't yield helpful results. Went with using a case statement instead

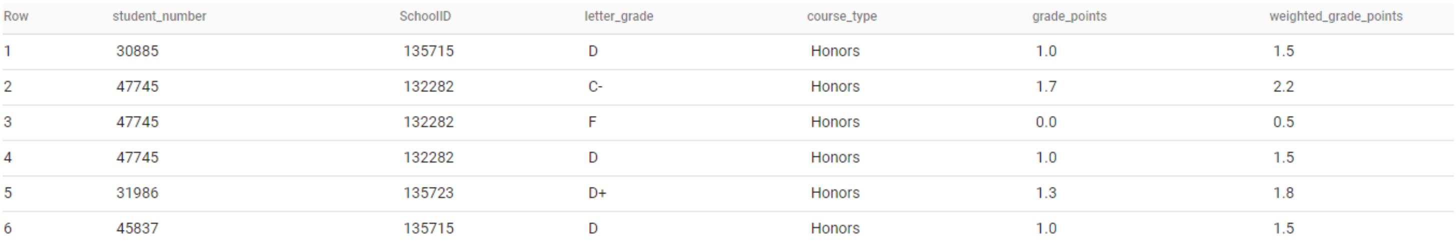
```
select student_number, SchoolID, letter_grade, course_type,
case
    when letter_grade = "A+" then 4.3
    when letter_grade = "A" then 4.0
    when letter_grade = "A-" then 3.7
    when letter_grade = "B+" then 3.3
    when letter_grade = "B" then 3.0
    when letter_grade = "B-" then 2.7
    when letter_grade = "C+" then 2.3
    when letter_grade = "C" then 2.0
    when letter_grade = "C-" then 1.7
    when letter_grade = "D+" then 1.3
    when letter_grade = "D" then 1.0
    when letter_grade = "D-" then 0.7
    when letter_grade = "F" then 0
end as grade_points
from `charter.performance_task.grades`;
```

Now to create weighted grade points

```
with calcing_gradepoints as
(select student_number, SchoolID, letter_grade, course_type,
case
    when letter_grade = "A+" then 4.3
    when letter_grade = "A" then 4.0
    when letter_grade = "A-" then 3.7
    when letter_grade = "B+" then 3.3
    when letter_grade = "B" then 3.0
    when letter_grade = "B-" then 2.7
    when letter_grade = "C+" then 2.3
    when letter_grade = "C" then 2.0
    when letter_grade = "C-" then 1.7
    when letter_grade = "D+" then 1.3
    when letter_grade = "D" then 1.0
    when letter_grade = "D-" then 0.7
    when letter_grade = "F" then 0
end as grade_points
from `charter.performance_task.grades`)

select *,
case
    when course_type = "AP" and letter_grade != "F" then grade_points + 1.0
    when course_type = "Honors" and letter_grade != "F" then grade_points + 0.5
    else grade_points + 0
end as weighted_grade_points
from calcing_gradepoints
```

ordered by course_type? for some reason, but works. Still contains all rows.



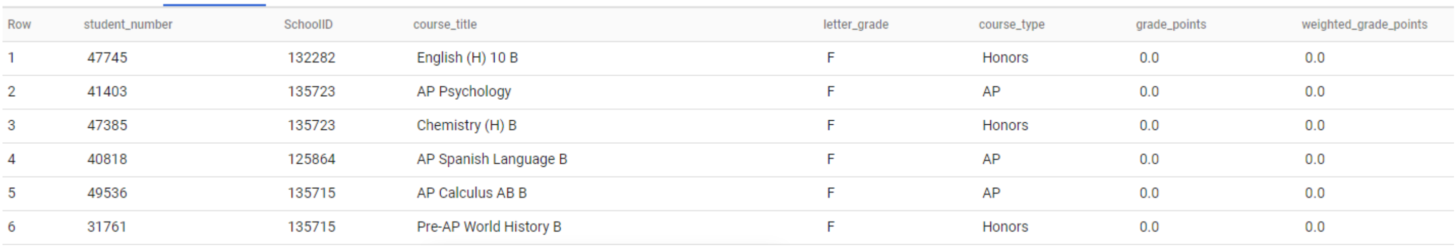
Row	student_number	SchoolID	letter_grade	course_type	grade_points	weighted_grade_points
1	30885	135715	D	Honors	1.0	1.5
2	47745	132282	C-	Honors	1.7	2.2
3	47745	132282	F	Honors	0.0	0.5
4	47745	132282	D	Honors	1.0	1.5
5	31986	135723	D+	Honors	1.3	1.8
6	45837	135715	D	Honors	1.0	1.5

Checking to make sure it did not add weight value for honors/ap that were F's:

```
select *,
case
    when course_type = "AP" and letter_grade != "F" then grade_points + 1.0
    when course_type = "Honors" and letter_grade != "F" then grade_points + 0.5
    else grade_points + 0
```

```
end as weighted_grade_points
from calcing_gradepoints
where letter_grade = "F" and course_type is not null
```

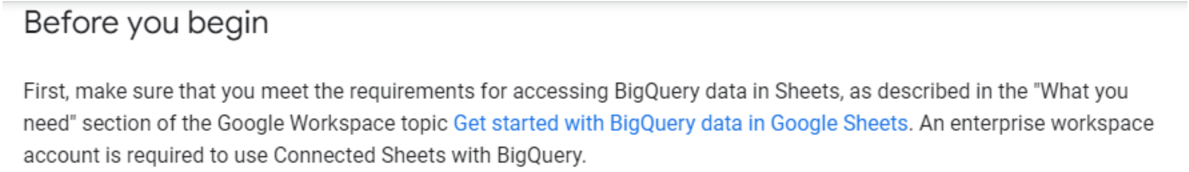
good



Row	student_number	SchoolID	course_title	letter_grade	course_type	grade_points	weighted_grade_points
1	47745	132282	English (H) 10 B	F	Honors	0.0	0.0
2	41403	135723	AP Psychology	F	AP	0.0	0.0
3	47385	135723	Chemistry (H) B	F	Honors	0.0	0.0
4	40818	125864	AP Spanish Language B	F	AP	0.0	0.0
5	49536	135715	AP Calculus AB B	F	AP	0.0	0.0
6	31761	135715	Pre-AP World History B	F	Honors	0.0	0.0

Now I need the Grade Point Average grouped by student. Going to be a little tricky to ignore the null values from advisory classes (which means I'll need course_title after all... need to update that)

Is this why I can't seem to alter the table?



Before you begin

First, make sure that you meet the requirements for accessing BigQuery data in Sheets, as described in the "What you need" section of the Google Workspace topic Get started with BigQuery data in Google Sheets. An enterprise workspace account is required to use Connected Sheets with BigQuery.

Since I don't currently have an enterprise account, I can't seem to connect BigQuery to Google Sheets properly.
This time I will download the data as a csv and import it to BigQuery as such.

Create a column for grade points (which will be averaged for GPA)

```
alter table charter.perf_task.grades
add column grade_points numeric;
```

Update the grade_points column with points based on grading scale

```
update charter.perf_task.grades
set grade_points = case
    when Letter_Grade = "A+" then 4.3
    when Letter_Grade = "A" then 4.0
    when Letter_Grade = "A-" then 3.7
    when Letter_Grade = "B+" then 3.3
    when Letter_Grade = "B" then 3.0
    when Letter_Grade = "B-" then 2.7
    when Letter_Grade = "C+" then 2.3
    when Letter_Grade = "C" then 2.0
    when Letter_Grade = "C-" then 1.7
    when Letter_Grade = "D+" then 1.3
    when Letter_Grade = "D" then 1.0
    when Letter_Grade = "D-" then 0.7
    when Letter_Grade = "F" then 0
    else grade_points
    end
where Letter_Grade is not null;
```

Update the grade_points column again to add 0.5 when Course_type = honors, and add 1.0 when Course_Type = AP

```
update charter.perf_task.grades
set grade_points = case
    when course_type = "Honors" then grade_points + 0.5
    when course_type = "AP" then grade_points + 1.0
    else grade_points
    end
where Letter_Grade != "F";
```

Next: Calculate GPAs grouped by student? or by school? Don't forget to exclude Course_Title like Advisory% or whatever.

First I'll want to make sure I catch all the advisory classes that will need to be excluded. So let me find out what they're called

```
select distinct course_title from `charter.perf_task.grades`
```

It looks like they all have names like "Advisory 9 AM" "Advisory 9PM" etc.

This query will create an honor roll of sorts. It gives all students (from across all the schools) that have a weighted GPA of 4.5 or above:

```
/* update statements above */
select Student_Number, Round(avg(grade_points), 2) as GPA
from charter.perf_task.grades
where Course_Title not like "Advisory%"
group by Student_Number
having GPA >= 4.5
order by GPA desc
```

GPA by school ID

```
select SchoolID, Round(avg(grade_points), 2) as GPA
from charter.perf_task.grades
where Course_Title not like "Advisory%"
group by SchoolID
order by GPA desc
```

Need to get this to join with school names

Final answer? Avg GPA by school (now with school names instead of IDs)

```
/* update statements above */
select School_Name, Round(avg(grade_points), 2) as GPA
from charter.perf_task.grades
join charter.perf_task.schools
on grades.SchoolID = schools.SchoolID
where Course_Title not like "Advisory%"
group by School_Name
order by GPA desc
```

Results:

Row	School_Name	GPA
1	Campus B	3.46
2	Campus A	3.41
3	Campus C	3.39
4	Campus F	3.27
5	Campus D	3.26
6	Campus E	3.07

GPA by English Proficiency

```
/* update statements above */
select English_Proficiency, Round(avg(grade_points), 2) as GPA
from charter.perf_task.demographics
join charter.perf_task.grades
on grades.Student_Number = demographics.Student_Number
where Course_Title not like "Advisory%"
group by English_Proficiency
order by GPA desc
```

Row	English_Proficiency	GPA
1	TBD	4.15
2	IFEP	3.56

3	RFEP	3.43
4	EO	3.24
5	EL	2.89

Create a table with counts & percentages of proficiency categories

```
select English_Proficiency, count(*) as Num_Students, Round((Count(English_Proficiency)* 100 / (Select Count(*) From charter.perf_task.demographics)),1) as
Percent_of_Population
from charter.perf_task.demographics
group by English_Proficiency
```

Row	English_Proficiency	Num_Students	Percent_of_Population
1	EL	380	15.1
2	EO	679	27.0
3	TBD	24	1.0
4	IFEP	174	6.9
5	RFEP	1261	50.1

GPA by IEP Status

```
select IFNULL(Student_has_IEP_, false) as Has_IEP, Round(avg(grade_points), 2) as GPA
from charter.perf_task.demographics
join charter.perf_task.grades
on grades.Student_Number = demographics.Student_Number
where Course_Title not like "Advisory%"
group by Student_has_IEP_
order by GPA desc
```

Row	Has_IEP	GPA
1	false	3.37
2	true	2.89

Population breakdown for IEP status

```
select IFNULL(Student_has_IEP_, false) as Has_IEP, count(*) as Num_Students, Round((Count(IFNULL(Student_has_IEP_, false))* 100 / (Select Count(*) From
charter.perf_task.demographics)),1) as Percent_of_Population
from charter.perf_task.demographics
group by Student_has_IEP_
```

Row	Has_IEP	Num_Students	Percent_of_Population
1	false	2175	86.4
2	true	343	13.6

~~Breakdown of ethnicity that are in AP or honors classes.~~
~~First I'll want the counts & percentages of each ethnicity.~~

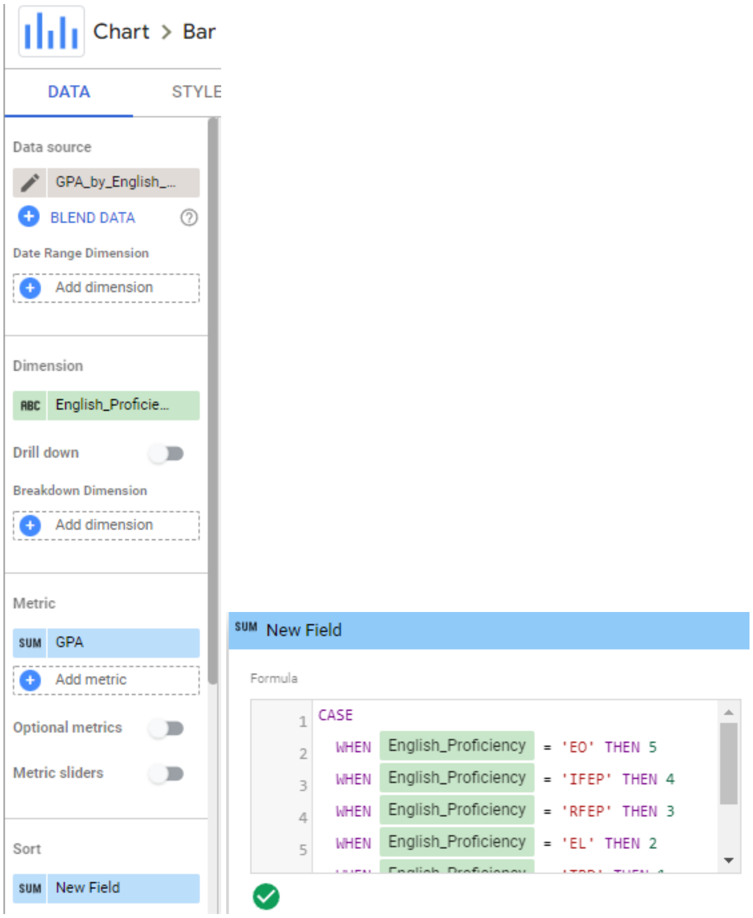
~~Then I'll want to find AP and Honors classes' percent of demographics breakdown~~
~~group by class, group by~~

~~join the demographics and grades tables such that each student's ethnicity is listed with every instance of their student number~~

```
select grades.Student_Number, grades.course_type, demographics.ethnicity
from charter.perf_task.grades
join charter.perf_task.demographics
on grades.Student_Number = demographics.Student_Number
```

Data Studio

I wanted the English Proficiency Data visualization to be ordered in the logical order of the ELPAC levels, so I had to manually add a field to sort by



Data source: GPA_by_English_...

Dimension: ABC English_Proficie...

Metric: SUM GPA

Sort: SUM New Field

SUM New Field

Formula

```
CASE
  WHEN English_Proficiency = 'EO' THEN 5
  WHEN English_Proficiency = 'IFEP' THEN 4
  WHEN English_Proficiency = 'RFEP' THEN 3
  WHEN English_Proficiency = 'EL' THEN 2
```

GPA for each student

```
select Student_Number, SchoolID, Round(avg(grade_points), 2) as GPA
from charter.perf_task.grades
where Course_Title not like "Advisory%"
group by Student_Number, SchoolID
```

Average GPA by school

```
select School_Name, round(avg(GPA),2) as GPA
from charter.perf_task.student_GPA
join charter.perf_task.schools
on student_GPA.SchoolID = schools.SchoolID
group by School_Name
order by 2 desc
```

Row	School_Name	GPA
1	Campus C	3.45
2	Campus F	3.41
3	Campus A	3.38

4	Campus B	3.26
5	Campus D	3.26
6	Campus E	3.07

Table with Number of students and percent of population broken down by English Proficiency status

```
select English_Proficiency, count(*) as Num_Students, Round((Count(English_Proficiency)* 100 / (Select Count(*) From charter.perf_task.demographics)),1) as
Percent_of_Population
from charter.perf_task.demographics
group by English_Proficiency
```

Row	English_Proficiency	Num_Students	Percent_of_Population
1	EO	679	27.0
2	RFEP	1261	50.1
3	EL	380	15.1
4	IFEP	174	6.9
5	TBD	24	1.0

Avg GPA by English Proficiency Status

```
select English_Proficiency, round(avg(GPA),2) as GPA
from charter.perf_task.demographics
join charter.perf_task.student_GPA
on demographics.Student_Number = student_GPA.Student_Number
group by English_Proficiency
```

Row	English_Proficiency	GPA
1	EO	3.23
2	RFEP	3.42
3	EL	2.89
4	IFEP	3.56
5	TBD	4.15

Table with Number of students and percent of population broken down by IEP status

```
select English_Proficiency, count(*) as Num_Students, Round((Count(English_Proficiency)* 100 / (Select Count(*) From charter.perf_task.demographics)),1) as
Percent_of_Population
from charter.perf_task.demographics
group by English_Proficiency;
```

Row	English_Proficiency	Num_Students	Percent_of_Population
1	EO	679	27.0
2	RFEP	1261	50.1
3	EL	380	15.1
4	IFEP	174	6.9
5	TBD	24	1.0

AVG GPA by IEP status

```
select IFNULL(Student_has_IEP_, false) as Has_IEP, round(avg(GPA),2) as GPA
from charter.perf_task.demographics
join charter.perf_task.student_GPA
on demographics.Student_Number = student_GPA.Student_Number
group by Student_has_IEP_
```

Row	Has_IEP	GPA
1	false	3.37
2	true	2.89

Later: I wanted to see a breakdown of ethnicity groups for proportions that are in honors/AP classes

```
with eth_count as (select Ethnicity, count(*) as Num_students
from charter.perf_task.demographics
group by Ethnicity),

honorsplus as (select distinct demographics.Student_Number, course_type, Ethnicity
from charter.perf_task.demographics
join charter.perf_task.grades
on demographics.Student_Number = grades.Student_Number
where course_type = "AP" or course_type = "Honors"),

unique_honorsplus as (select distinct Student_Number, Ethnicity
from honorsplus),

count_unique_honorsplus as (select Ethnicity, count(*) as Num_honorsplus_students
from unique_honorsplus
group by Ethnicity)

select eth_count.Ethnicity, eth_count.Num_students, count_unique_honorsplus.Num_honorsplus_students
from eth_count join count_unique_honorsplus
on eth_count.Ethnicity = count_unique_honorsplus.Ethnicity
```

Then wanted a percentage breakdown

```
with eth_count as (select Ethnicity, count(*) as Num_students
from charter.perf_task.demographics
group by Ethnicity),
honorsplus as (select distinct demographics.Student_Number, course_type, Ethnicity
from charter.perf_task.demographics
join charter.perf_task.grades
on demographics.Student_Number = grades.Student_Number
where course_type = "AP" or course_type = "Honors"),
unique_honorsplus as (select distinct Student_Number, Ethnicity
from honorsplus),
count_unique_honorsplus as (select Ethnicity, count(*) as Num_honorsplus_students
from unique_honorsplus
group by Ethnicity)
select eth_count.Ethnicity, eth_count.Num_students, count_unique_honorsplus.Num_honorsplus_students, round(count_unique_honorsplus.Num_honorsplus_students * 100
/ eth_count.Num_students , 1) as Percent_of_EthnicGroup_inHonorsPlus
from eth_count join count_unique_honorsplus
on eth_count.Ethnicity = count_unique_honorsplus.Ethnicity
```