SOUTH CAROLINA
DEPARTMENT OF EDUCATION

Graduation Rate - Graduates

Scoring Methodology:

Numerator:

Number of students who earned a regular high school diploma within four years.

Denominator:

Number of students enrolled at a given high school by the 180th day who have the same 9GR who have not been removed from the cohort for one of the reasons described in the "Cohort Removal" (student death, emigration, transfer to prison or juvenile facility following adjudication, and properly documented transfer will all need proper documentation).

Data Sources and Validations:

- Ensure student has the following graduation related fields answered: Diploma Earned and Graduation Date (PowerSchool->Compliance->SC Student Information (Diploma Earned & Graduation Date)).
- Ensure student is within correct graduation cohort in PowerSchool **(**PowerSchool->Compliance->SC Student Information (Ninth Grade Code))
- Ensure student has correct cohort status in the 'Cohort Maintenance Worksheet' in PowerSchool (PowerSchool->Compliance->Cohort Maintenance Worksheet ('Student Should Remain in Cohort (denominator)?'))

Progress Monitoring:

- GPS (Growing Pathways for Our Students) Dashboard to check

Graduation Rate - Code

Snowflake Code:

Copy the following code to paste into Snowflake/Stadium to check records:

```
---------------------------------------------- District Level ----------------------------------------------
with student_level as (
  select * from analytics.prod_accountability.msr_student_grad_rate
),
latest_student_demographics as (
  select * from analytics.prod_build.bld_ef3__latest_student_demographics
),
district as (SELECT *
FROM ANALYTICS.PROD_WH.DIM_LEA),
district_aggregate as (
  select
    district.lea_name as District_Name,
    student_level.school_year,
    student_level.cohort_year,
    student_level.cohort_group,
    sum(student_level.did_graduate_4year::int) as n_students_grad_on_time,
    count(*) as n_students,
    round(100 * sum(student_level.did_graduate_4year::int) / nullifzero(count(*)), 2)
 as pct_grad_rate_on_time,
    round(19 * greatest((sum(student_level.did_graduate_4year::int) * 100.0 / nullifzero(count(*)) - 50) / 50, 0),2) as
rating_points
  from student_level
  left join analytics.prod_wh.fct_student_subgroup sg_by_student
   on student_level.k_student = sg_by_student.k_student
  left join latest_student_demographics sg_by_latest
   on student_level.k_student_xyear = sg_by_latest.k_student_xyear
   and student_level.k_student is null
  LEFT JOIN district on district.k_lea = student_level.k_lea
  group by all
)
select *,
  case
    when rating_points >= 15.20 then 'Excellent'
    when rating_points >= 12.67 then 'Good'
    when rating_points >= 10.13 then 'Average'
    when rating_points >= 7.60 then 'Below Average'
    else 'Unsatisfactory'
  end as rating_label
from district_aggregate
```

```
 /* Delete this row to add filters
-- Add filters here:
-- For example, if you only want to see graduation rates for the 2022 cohort and the 2025 school year, use the following code:
WHERE COHORT_YEAR = 2022
AND SCHOOL_YEAR = 2025
*/ --Delete this row to add filters
ORDER BY District_Name, school_year, cohort_year



---------------------------------------------- School Level ----------------------------------------------
with student_level as (
   select * from analytics.prod_accountability.msr_student_grad_rate
),
latest_student_demographics as (
   select * from analytics.prod_build.bld_ef3__latest_student_demographics
),
school as (SELECT *
FROM ANALYTICS.PROD_WH.DIM_SCHOOL),
school_aggregate as (
   select
      s.school_name,
      s.lea_name as District_Name,
      student_level.school_year,
      student_level.cohort_year,
      student_level.cohort_group,
      sum(student_level.did_graduate_4year::int) as n_students_grad_on_time,
      count(*) as n_students,
      round(100 * sum(did_graduate_4year::int) / nullifzero(count(*)), 2)

 as pct_grad_rate_on_time,
      round(
        19 * greatest((sum(student_level.did_graduate_4year::int) * 100.0 / nullifzero(count(*)) - 50) / 50, 0),
```

```
      2) as rating_points
   from student_level
   left join analytics.prod_wh.fct_student_subgroup sg_by_student
    on student_level.k_student = sg_by_student.k_student
   left join latest_student_demographics sg_by_latest
    on student_level.k_student_xyear = sg_by_latest.k_student_xyear
    and student_level.k_student is null
   left join school s
   ON s.k_school = student_level.k_school AND s.tenant_code = student_level.tenant_code
   group by all
)
select *,
   case
      when rating_points >= 15.20 then 'Excellent'
      when rating_points >= 12.67 then 'Good'
      when rating_points >= 10.13 then 'Average'
      when rating_points >= 7.60 then 'Below Average'
      else 'Unsatisfactory'
   end as rating_label
from school_aggregate
/* Delete this row to add filters
-- Add filters here:
-- For example, if you only want to see gradution rates for the 2022 cohort and the 2025 school year and for school 'ABC High
School', use the following code:
WHERE COHORT_YEAR = 2022
AND SCHOOL_YEAR = 2025
AND SCHOOL_NAME = 'ABC High School'
*/ --Delete this row to add filters
ORDER BY school_name, school_year, cohort_year
```

---------- Student Level ----------

```
with stu_active_grad_cohort as (
   select * from analytics.prod_build.bld_ef3__stu_active_grad_cohort
```

```
  where cohort_group = '4_year_grad_cohort'
),
fct_student_diploma as (
  select * from analytics.prod_wh.fct_student_diploma
  where diploma_award_date <= current_date
),
fct_student_school_association as (
  select * from analytics.prod_wh.fct_student_school_association
),
school as (SELECT *
FROM ANALYTICS.PROD_WH.DIM_SCHOOL
),
student as (SELECT *
FROM ANALYTICS.PROD_WH.DIM_STUDENT
),
cut_off_date_by_student as (
  select
    k_student,
    k_student_xyear,
    cohort_year,
    enrollment_school_year,
    dateadd(day,
    ((8 - date_part('dow', date_from_parts(enrollment_school_year, 8, 1))) % 7) + 14,
    date_from_parts(enrollment_school_year, 8, 1)) as fall_cutoff_date
  from stu_active_grad_cohort
),
grad_status_by_year as (
  select
    d.k_student,
    d.k_student_xyear,
    d.school_year,
    d.diploma_award_date,
    d.diploma_type,
    d.diploma_award_date <= cutoff.fall_cutoff_date as did_graduate
  from fct_student_diploma d
  left join cut_off_date_by_student cutoff on
    d.k_student_xyear = cutoff.k_student_xyear
  where d.diploma_type in ('F','Z')
   and d.diploma_award_date <= cutoff.fall_cutoff_date
  qualify row_number() over (
    partition by d.k_student_xyear, cutoff.enrollment_school_year
    order by d.diploma_award_date asc
  ) = 1
)
select
  sd.student_unique_id,
  s.school_name,
```

```
  s.lea_name as District_Name,
  cohort.enrollment_school_year as school_year,
  cohort.cohort_year,
  cohort.entry_grade_level,
  cohort.cohort_group,
  coalesce(grad_status_by_year.did_graduate, false) as did_graduate_4year,
  case
    when grad_status_by_year.did_graduate then 'Received Diploma On Time'
   else 'Did Not Graduate On Time'
  end as on_time_graduation_status
from stu_active_grad_cohort cohort
LEFT JOIN student sd
ON sd.k_student = cohort.k_student AND sd.tenant_code = cohort.tenant_code
left join grad_status_by_year
ON cohort.k_student_xyear = grad_status_by_year.k_student_xyear
LEFT JOIN school s
ON s.k_school = cohort.k_school AND s.tenant_code = cohort.tenant_code
/* Delete this row to add filters
-- Add filters here:
-- For example, if you only want to see a student with the unique id of '1111111111', use the following code:
WHERE STUDENT_UNIQUE_ID = '1111111111'
*/ --Delete this row to add filters
ORDER BY school_name, school_year, cohort_year
```