**6210FinalProject Group15 PostgreSQL Implementation**

1.1 Student Discipline from School Administrator perspective:

For each facility, how many students violate the discipline per year?

select df.FacilityName,count(fd.StudentID) as DisciplineStudentNumber,

EXTRACT(YEAR FROM fd.EventDate) AS Year

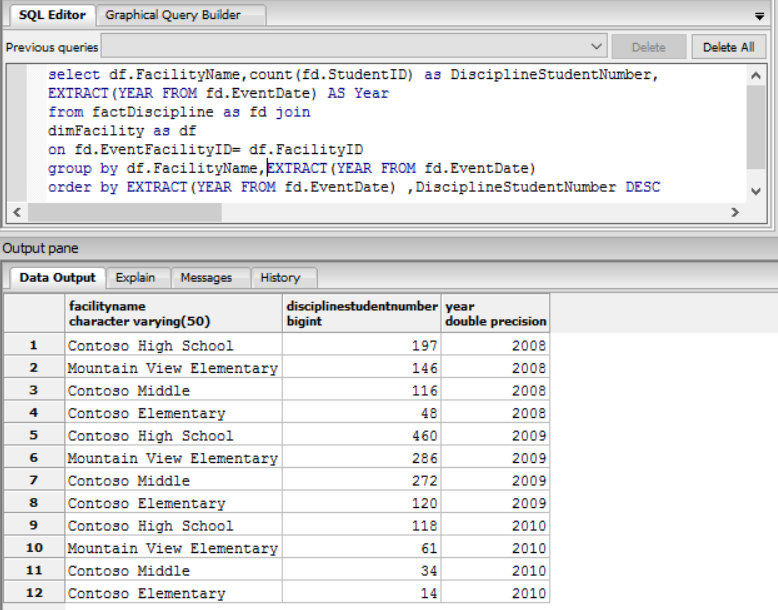
from factDiscipline as fd join

dimFacility as df

on fd.EventFacilityID= df.FacilityID

group by df.FacilityName,EXTRACT(YEAR FROM fd.EventDate)

order by EXTRACT(YEAR FROM fd.EventDate) ,DisciplineStudentNumber DESC



1.2 Student Discipline from parents perspective:

Rank the top 3 common Discipline Events

select dde.DisciplineEventCode, dde.DisciplineEventDesc, count(ds.StudentID) as DisciplineStudentNumber,

DENSE\_RANK()OVER(ORDER BY count(ds.StudentID) DESC) AS DisciplineReason\_RANK

from factDiscipline as fd join

dimDisciplineEventCode as dde

on dde.DisciplineEventCodeID=fd.DisciplineEventCodeID join

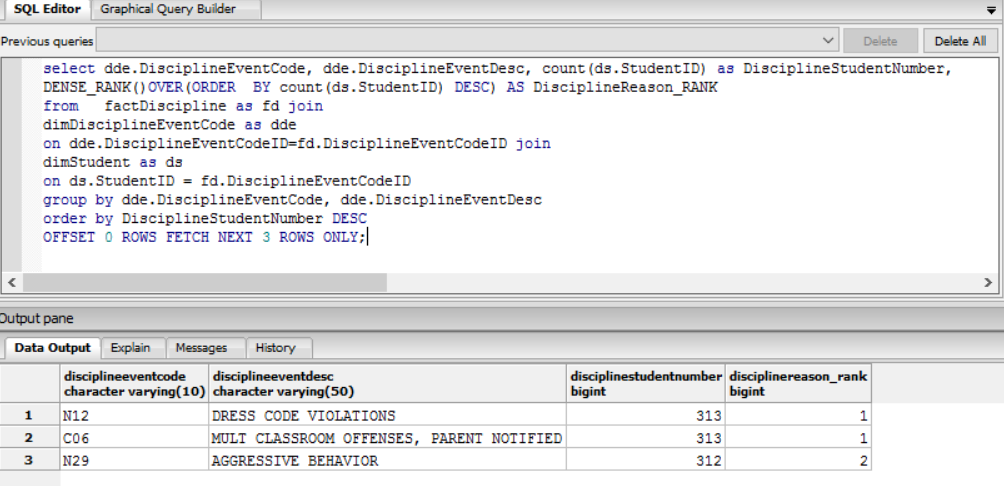
dimStudent as ds

on ds.StudentID = fd.DisciplineEventCodeID

group by dde.DisciplineEventCode, dde.DisciplineEventDesc

order by DisciplineStudentNumber DESC

OFFSET 0 ROWS FETCH NEXT 3 ROWS ONLY;



1.3 Student Discipline from teachers perspective:

For each discipline behaviors, compare the total number of female students and male students

select dde.DisciplineEventCodeID,dde.DisciplineEventDesc,c."F",c."M",c."M>F"

from dimDisciplineEventCode as dde

left join

((select b.DisciplineEventCodeID,

sum(case when b.Gender= 'F' then 1 else 0 end) as "F",

sum(case when b.Gender= 'M' then 1 else 0 end) as "M",

cast((case when sum(case when b.Gender= 'M' then 1 else 0 end) >

sum(case when b.Gender= 'F' then 1 else 0 end) then 1 Else 0 End) as float

) As "M>F"

from(

SELECT distinct fd.StudentID, ds.Gender, fd.DisciplineEventCodeID

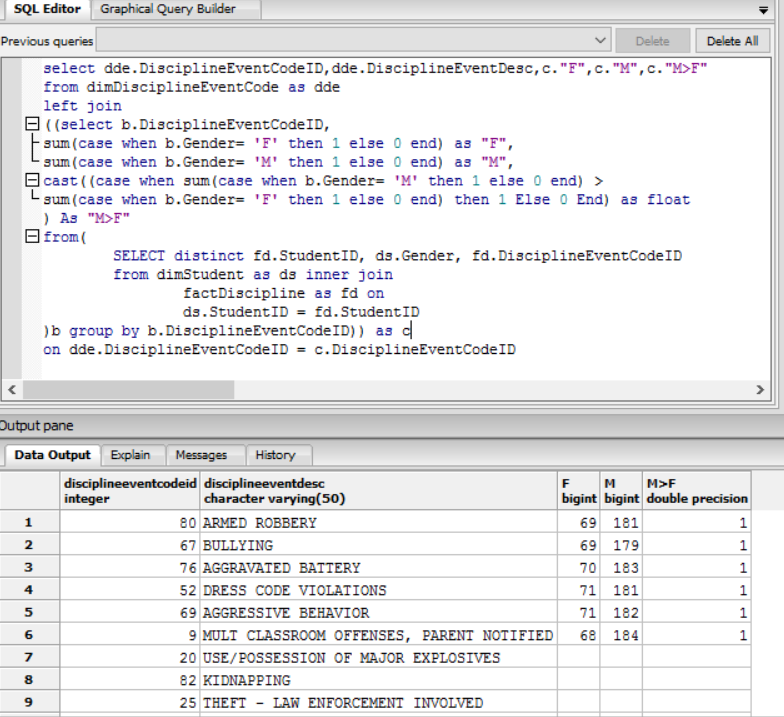
from dimStudent as ds inner join

factDiscipline as fd on

ds.StudentID = fd.StudentID

)b group by b.DisciplineEventCodeID)) as c

on dde.DisciplineEventCodeID = c.DisciplineEventCodeID



2.1 Student Absences from School Administrator perspective:

How many absent students are there in each facility?

SELECT df.FacilityName , count(dd.StudentID) as AbsentStudentNumber

FROM

factAbsence fb join

dimStudent dd on

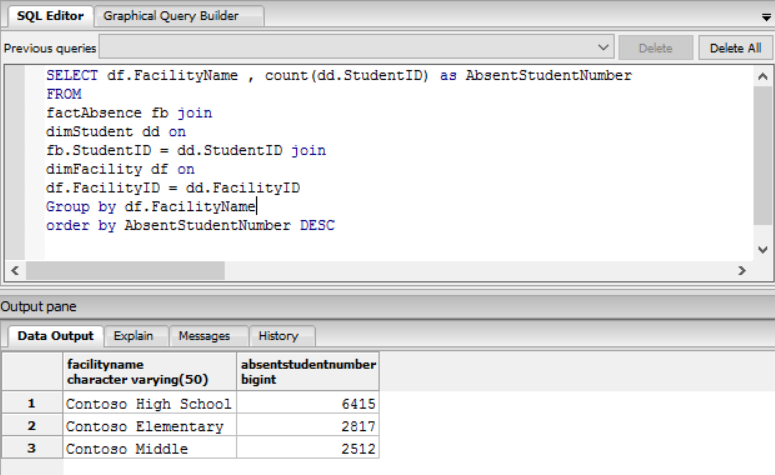
fb.StudentID = dd.StudentID join

dimFacility df on

df.FacilityID = dd.FacilityID

Group by df.FacilityName

order by AbsentStudentNumber DESC



2.2 Student Absences from teachers perspective:

unexcused absence reason percentage of the total absences

select a.Number\_of\_Y, a.Number\_of\_N,

a.Total\_Number,

round(cast((cast(a.Number\_of\_Y as float))/(cast(a.Total\_Number as float))\*100 as int) ,2) as "Percentage%"

from (

select sum(case when da.Unexcused = 'Y' then 1 else 0 end) as Number\_of\_Y,

sum(case when da.Unexcused = 'N' then 1 else 0 end) as Number\_of\_N,

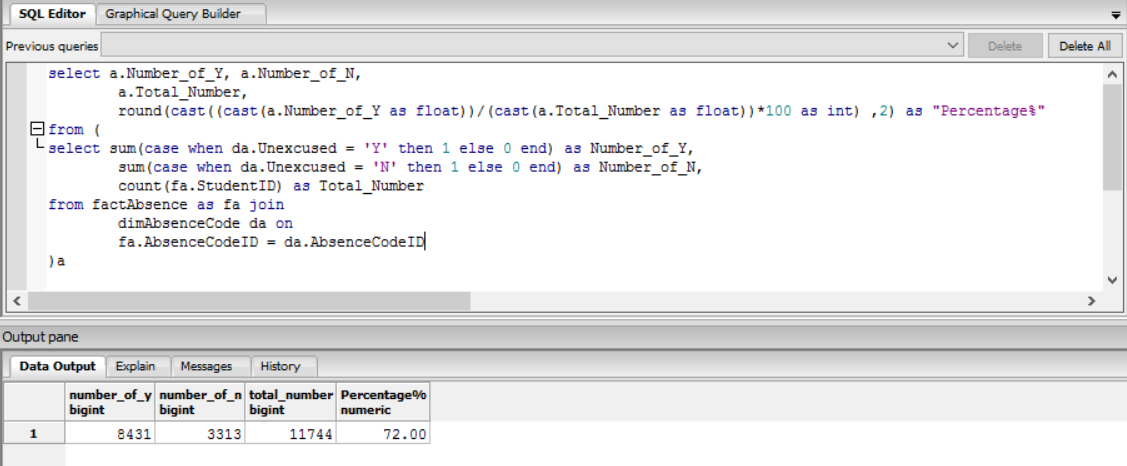
count(fa.StudentID) as Total\_Number

from factAbsence as fa join

dimAbsenceCode da on

fa.AbsenceCodeID = da.AbsenceCodeID

)a



2.3 Student Absences from parents perspective:

Rank the reason why students are absent

select da.AbsenceCodeDesc, count(ds.StudentID) as AbsentStudentNumber,

DENSE\_RANK()OVER(ORDER BY count(ds.StudentID) DESC) AS AbsenceReason\_RANK

from factAbsence as fa join

dimAbsenceCode as da

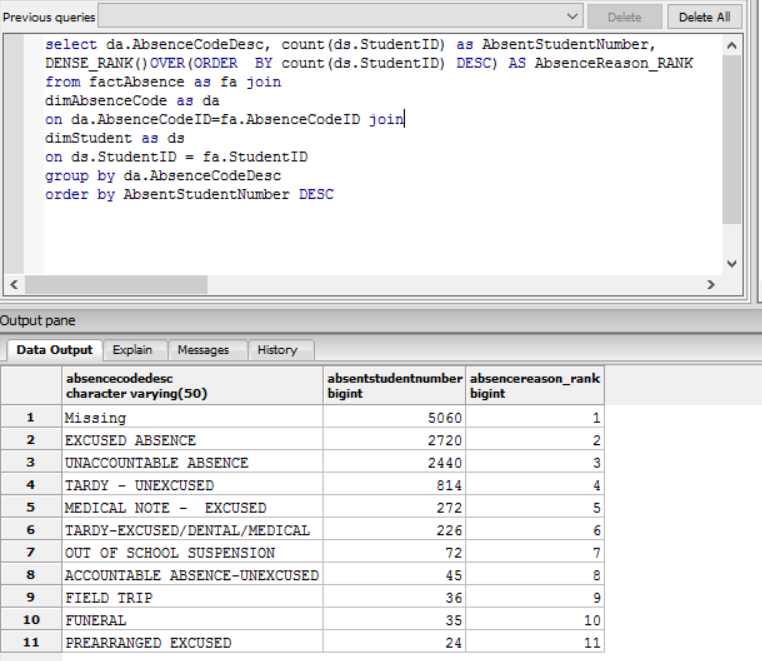
on da.AbsenceCodeID=fa.AbsenceCodeID join

dimStudent as ds

on ds.StudentID = fa.StudentID

group by da.AbsenceCodeDesc

order by AbsentStudentNumber DESC



3.1 Academic performance from School Administrator perspective:

For each facility, list average raw score and scaled score for every subject

select df.FacilityName,fsd.SubjectID,

AVG(fsd.ScaledScore) as ScaledScore,

AVG(fsd.RawScore) as RawScore

from factStandardizedTestScore as fsd

join dimStudent as ds on

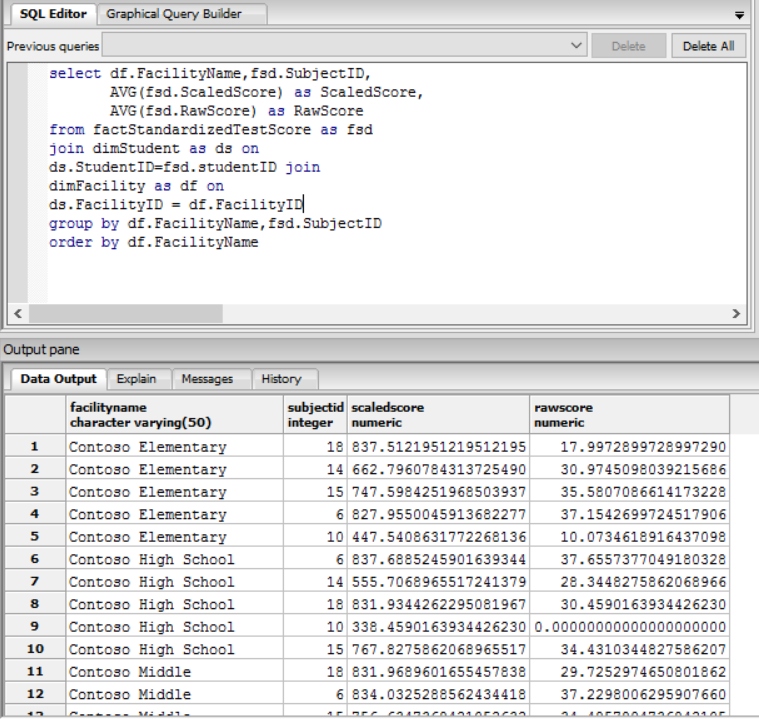
ds.StudentID=fsd.studentID join

dimFacility as df on

ds.FacilityID = df.FacilityID

group by df.FacilityName,fsd.SubjectID

order by df.FacilityName



3.2 Academic performance from teachers perspective:

Average numeric grade of students for each course

select dc.CourseID,dc.CourseTitle,avg(fscg.NumericGrade) as AverageScore

from factStudentCourse as fsc join

dimCourse as dc on

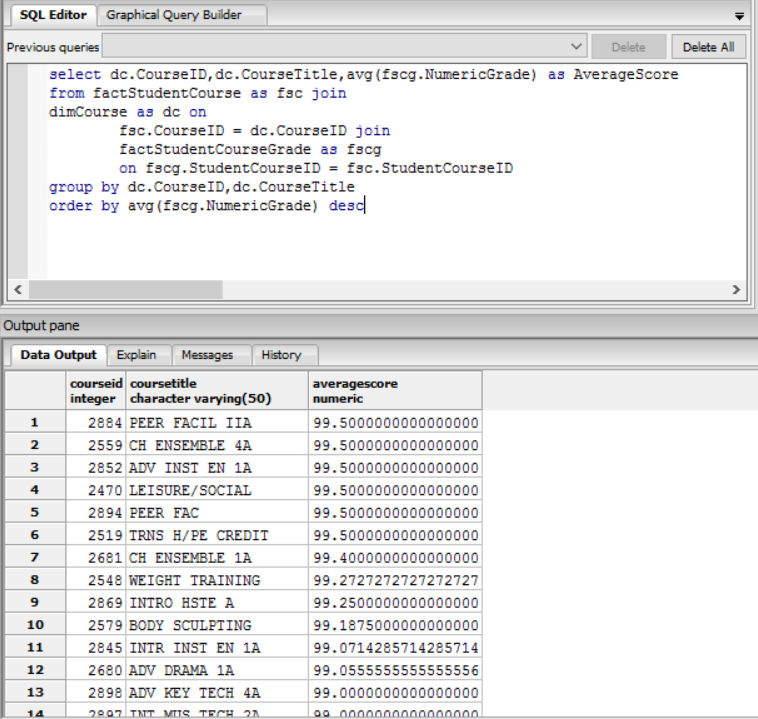
fsc.CourseID = dc.CourseID join

factStudentCourseGrade as fscg

on fscg.StudentCourseID = fsc.StudentCourseID

group by dc.CourseID,dc.CourseTitle

order by avg(fscg.NumericGrade) desc



3.3 Academic performance from parents perspective:

Rank student average course score by course

select dc.CourseTitle,ds.StudentID,ds.StudentName,avg(fscg.NumericGrade) as AverageScore,

RANK() over( PARTITION BY dc.CourseTitle order by avg(fscg.NumericGrade) desc) as RankByScore

from factStudentCourse as fsc join

dimstudent as ds on

fsc.StudentID = ds.StudentID join

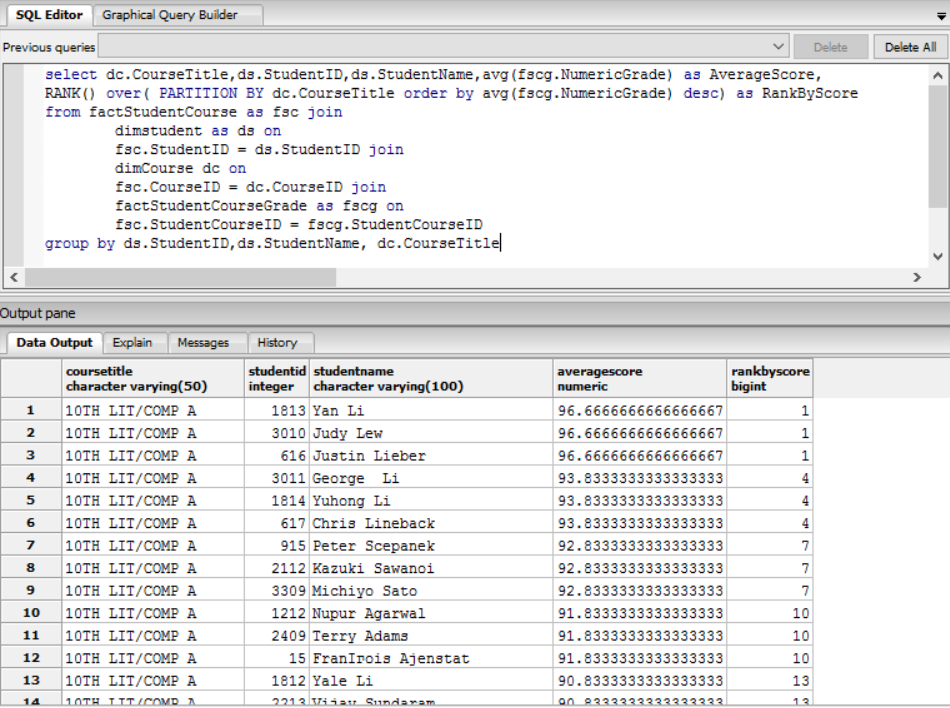
dimCourse dc on

fsc.CourseID = dc.CourseID join

factStudentCourseGrade as fscg on

fsc.StudentCourseID = fscg.StudentCourseID

group by ds.StudentID,ds.StudentName, dc.CourseTitle



4.1 Special Programs from School Administrator perspective:

How many Special Programs are there in every facility per year?

select df.FacilityName,count(fsp.SpecialProgramID) as SpecialProgramNumber,

EXTRACT(YEAR FROM fsp.EntryDate) AS Year

from factSpecialProgram as fsp join

dimStudent as dS

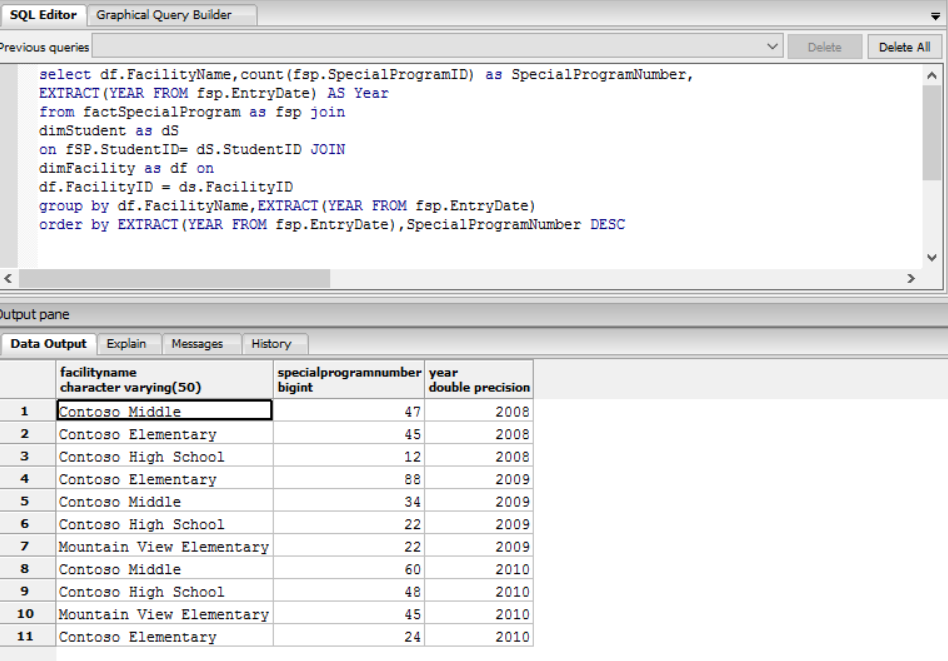
on fSP.StudentID= dS.StudentID JOIN

dimFacility as df on

df.FacilityID = ds.FacilityID

group by df.FacilityName,EXTRACT(YEAR FROM fsp.EntryDate)

order by EXTRACT(YEAR FROM fsp.EntryDate),SpecialProgramNumber DESC



4.2 Special Programs from teacher perspective:

How many Special Programs are there in every program?

select fsp.ProgramID,dsp.ProgramCode,count(fsp.SpecialProgramID) AS SpecialProgramNumber

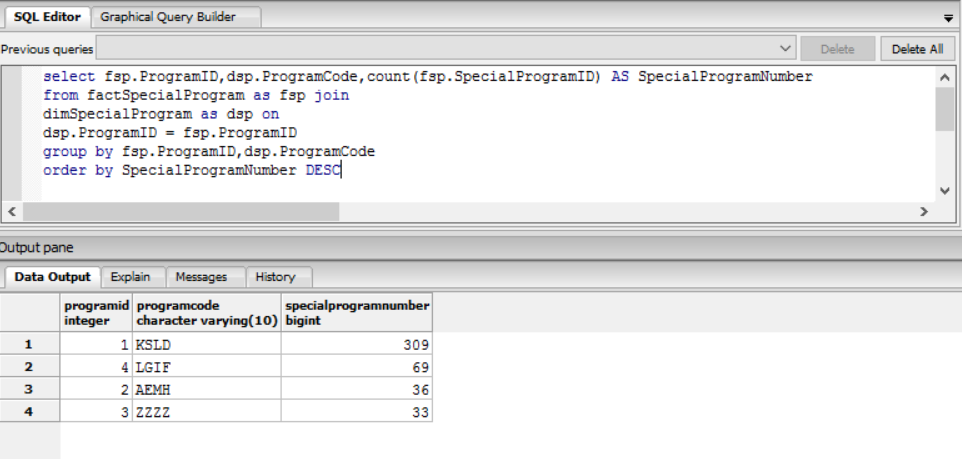
from factSpecialProgram as fsp join

dimSpecialProgram as dsp on

dsp.ProgramID = fsp.ProgramID

group by fsp.ProgramID,dsp.ProgramCode

order by SpecialProgramNumber DESC



4.3 Special Programs from parents perspective:

For each grade, How many student attend special programs in year of 2010

select ds.Grade,count(fsp.StudentID) AS StudentNumber

from factSpecialProgram as fsp join

dimStudent as ds on

ds.StudentID= fsp.StudentID

where EXTRACT(year from fsp.EntryDate)= '2010'

group by ds.Grade

