The Particulars of the Performance Report

Last update: 12/29/2000:
Recent changes.
1. Change Time to Degree from Average to Median

1) **Characteristics of the Faculty Teaching First Year Students**
   a) In the data base queries for these reports:
      i) First Year Students selected are those with Rank of FR or NU in AU 1998 and Year and Term of First Enrollment should be SM or AU 1998. We include Rank NU because there are significant numbers of NU students with Year and Term of First Enrollment in SM or AU 1998 and NU is the proper Rank for a first year student with no degree intention. The reason for using SM and AU 1998 instead of 1999 is because the FD data we have is for all terms of FY 1999.
      ii) Instructors selected are those who taught a course section reported in the ST in AU 1998 and whose demographics are in the FD in FY 1999 with Work Category Code = FA (Faculty) or IR (Instruction / Research Assistant). The reason for restricting to FA and IR is to avoid counting full time administrators who are part time instructors.
      iii) Selected Instructors are classified full time part time as follows:
         (1) Full Time if Appointment Code is 09 or 11,
         (2) GA if Appointment Code is GA,
         (3) Part Time, No Tenure if Appointment Code is PT, and Tenure Code is NT (Non-tenure-track),
         (4) Part Time, Tenure otherwise.
         (5) This report is produced for all sectors.
      iv) Selected Instructors are classified by Rank as follows:
         (1) GA if Appointment Code is GA,
         (2) Faculty Rank Code if Appointment Code is not GA, except combine Other, No Academic Rank and Unknown Ranks into one category called All Others.
         (3) Senior Faculty are considered as those with Rank of Professor (PRP or Associate Professor (AP).
         (4) This report is produced for four-year institutions, main campus and branches only.
      v) Selected Instructors are classified by Highest Degree as follows:
         (1) Graduate Degree if Highest Degree in the FD is Doctoral, Masters, Professional, or Educational Specialist,
         (2) Undergrad Degree if Highest Degree in the FD is Certificate, Assoc, or Bac,
         (3) Other Degree if the Highest Degree Code is None of the Above.
         (4) This report is produced for two-year institutions only and not branches of four-year institutions.
      vi) The queries show the number of SCH generated by the selected students by the various classifications of instructor, and the percentage of each classification compared to the whole. In the case of team teaching, or sections taught by more than one instructor the SCH are credited to each instructor.
      vii) In a separate query, we calculate the %'s of faculty in the DC that are Full Time, Senior and have Graduate Degrees as separate %s. These %'s are at the institution level because the FD file is at the institution level. We also calculate the % of SCH earned in AU 1998 by First Year Students, as defined above, compared to total SCH on each campus.
      viii) Spreadsheet with SQL is in O:\Data Queries \Faculty Load \FTE for FR by Faculty Char ver 3 SQL Adv.xls.
   b) In the published report the SCH generated by the various faculty characteristics are converted to %s and shown in tables and bar charts.

2) **The Experience of Under-Prepared First Year Students** (in AU 1998, English and Math remediation by Age Category)
   a) In the data base queries for these reports:
      i) Developmental coursework in AU 1998 is defined as:

(1) Developmental sections selected from AU 1998 are those with Level of V,
(2) Subject Code that maps to either the English or Math Subject Field. The mapping of Subject Code to Subject Field is on the WWW at http://www.regents.state.oh.us/hei/datasubdoc/vertables/subjcodes.html . Include Subject Code of 320101 or 320199 (Basic Skills) together with LIT, READ, COMPO, WRIT, or (ENG, but without SEC) as part of the Course Title in the English Subject Field and Subject Code of 320101 or 320199 (Basic Skills) together with MATH or ALGEBRA as part of the Course Title in the Math Subject Field.
    (a) Since some schools did not have their developmental courses coded as V, they may supply a list of Developmental Course IDs, and we will use this to select enrollments, this becomes an alternative to the selection based on Course Subject Code and Level. The attached table is the list of such courses.
(3) Course Enrollment Credit Hours >0,
(4) Course Enrollment Award of Academic Credit of Y.
(5) Course Enrollment Attempt of Academic Credit of I.
(6) Student Rank of either FR or NU and Term of First Enrollment should be SM or AU 1998,

ii) Age Category breaks the students into two groups, 20 and Under are students with Birth Year in the SE 1978 or later, all others are in the over 20 group, UNKN Birth Year is in the over 20 group.

iii) Subsequent non-developmental coursework is defined as:
(1) Course enrollment in any term WI 1999, SP 1999, SM 1999, AU 1999, and
(2) On the same campus as the developmental course work, and
(3) Level not equal V, and
(4) Course ID not included in the Developmental Course IDs that are not coded as developmental, and
(5) Subject Field the same as the Subject Field of the developmental course in AU 1998, and
(6) Course Enrollment Credit Hours >0.
(7) The subsequent course is considered completed successfully if the Award of Academic Credit is Y.

iv) The report counts students who completed all of their subsequent non-developmental coursework in the same field (English or Math) successfully, some successfully, and none successfully, as well as students who completed any developmental course work in AU 1998 regardless of subsequent enrollments and the total freshmen class in AU 1998.

v) Total Freshmen class is all students enrolled in AU 1998 with Rank of FR or NU and Year and Term of First Enrollment should be SM or AU 1998.

vi) Spreadsheet with SQL is in O:\Data Queries \developmental student credit hours \Continuing Students in Dev Math and Engl PR3 ver4.xls.

b) In the published report Total Remediation statistics are not taken from the HEI unit record data because too much data is missing. For the subsequent success of students who went on to college level work, percentages are shown of:

i) Successful all college level (English or Math): Students who successfully completed developmental (English or Math) in AU 1998 and attempted college level work in the next year in the same field and earned academic credit for all such instruction divided by Students who successfully completed developmental (English or Math) in AU 1998 and attempted college level work in the next year in the same field.

ii) Successful some college level (English or Math): Students who successfully completed developmental (English or Math) in AU 1998 and attempted college level work in the next year in the same field and earned academic credit for some but not all such instruction divided by Students who successfully completed developmental (English or Math) in AU 1998 and attempted college level work in the next year in the same field.

iii) Unsuccessful all college level (English or Math): Students who successfully completed developmental (English or Math) in AU 1998 and attempted college level work in the next year in the same field and did not earn academic credit for any such instruction divided by Students who successfully completed developmental (English or Math) in AU 1998 and attempted college level work in the next year in the same field.

iv) Non Persisting (English or Math): Students who successfully completed developmental (English or Math) in AU 1998 and did not attempt college level work in the next year in the same field at any Ohio public institution divided by Students who successfully completed developmental (English or Math) in AU 1998.

v) Continuing (English or Math): Students who successfully completed developmental (English or Math) in AU 1998 and did complete college level work in the next year in the same field at any Ohio public institution divided by Students who successfully completed developmental (English or Math) in AU 1998.

3) **Year to Year Persistence of Full time, First Time, Degree Seeking First Year Students (FTFTDSFR)**

a) The FTFTDSFR cohort for AU 1998 is defined as having all of these characteristics:

i) Full time: The total SCH in the Course Enrollment file (CN) for the student must be >= 12 in AU 1998.

ii) First term: Year and Term of First Enrollment must be SM or AU 1998, and if it comes from the SE, Rank in AU 1998 must be FR or NU, if it comes from the RF, Rank may be any undergraduate Rank.

iii) First time: First Time Ever in College Switch in the SE file must be Y. EDSN is excluded from this test because First Time in College Switch is N for all new students in SM and AU 1998. Also, CLRK has requested to be excluded from this test due to inaccurate reporting of the First Time in College Switch.

iv) Degree seeking: Intention Code in the SE must be 06, 07, or 08. University main campuses and branches of CINC, KENT, and BGSU are excluded from this test For campuses excluded from the Intention Code, Rank must not by NU. NU is not a degree seeking Rank, but for campuses that use the Intention Code, we use that to determine degree seeking. Also, for RGCC, 04 is a degree seeking Intention Code.

b) Persisting students are those included in the AU 1998 cohort who enrolled in AU 1999 on the same campus, different campus of the same institution, or different public institution. For students in the cohort who enrolled on more than one campus in AU 1999, only one enrollment counts, and the same campus is the first choice, same institution second and different institution third.

c) The report shows, by campus, the number of FTFTDSFR, the number who persisted in each of the three cases above and the total first year students. And the percentages of retained to FTFTDSFR and the percentage of FTFTDSFR to the first year class.

d) First year class is First Term and First Time as defined above.

e) Spreadsheet with SQL is in O:\Data Queries \Retention \FTFTDSFR first term in base ver 4.xls. Note ver 5 has the Executive summary data.

4) **Diversity (Demographics) Breakdown of Undergraduate Students in AU 1999**

a) Students selected are those enrolled in AU 1999 with Admissions Area = UND.

b) Full-time part-time is full-time for students enrolled in 12 or more credit hours and part-time for all others.

c) Age Category breaks the students into two groups, 24 and Under are students with Birth Year in the SE of 1975 or later, all others are in the over 24 group, UNKN Birth Year is in the over 24 group.

d) Residency at the time of admission is classified as:

i) Ohio if State of Residency in the SE corresponding to the AU 1999 student enrollment is OH,

ii) Foreign if it is ZZ,

iii) Out of State otherwise.

e) Race Ethnic Code is broken down by the codes on the database.

f) The report is summarized at the campus level and shows total number of enrolled students (head count) and the % of enrolled students in each of the above categories.

g) Spreadsheet with SQL is in O:\Data Queries\Student Demographics\undergrad AU 1999.xls.

5) **Diversity (Demographics) Breakdown of Graduate Students in AU 1999**

a) Students selected are those enrolled in AU 1999 with Rank = "MS","DS","NM","ND", or "PR".

b) Residency at the time of admission is classified as:

i) Ohio if State of Residency in the SE corresponding to the AU 1999 student enrollment is OH,

ii) Foreign if it is ZZ,

iii) Out of State otherwise.

c) The report is summarized at the institution level and shows total number of enrolled students (head count) and the % of enrolled students in each of the above categories. The count is unduplicated at the institution level, so if a student is enrolled at more than one campus he/she counts only once.

d) Spreadsheet with SQL is in O:\Data Queries\Student Demographics\grad AU 1999.xls.

6) Time and Credits to Degree

a) Time to degree

i) Time to degree is measured for all level 03 and 05 degrees in the DC file earned in any term of FY 1999. The time is measured in years and is the time between the First Year and Term enrolled from the SE and the Year and Term the degree was earned in the DC. However, if the student has a record in the RF file for this institution, use that Year and Term for the beginning year and term Both the beginning and ending term are included in the calculated time and each semester term is 1/3 year and each quarter is ¼ year. Time to degree is calculated both with and without an adjustment for extended programs. The institution calendar used in the term that the degree was earned is assumed to be the calendar for the student's entire duration. Extended programs are adjusted by multiplying the time calculated by the ratio of the normal time for Assoc or Bac degrees (2 or 4 years) to the extended time for this degree. However, the adjusted time is not used in the final reports.

ii) Note CNST has five terms per year, two of which are labeled WI; this may cause their duration to be miscalculated by one or two terms.

iii) Only students who are assumed not to be transfer students are included. This is determined by the SCH earned for the degree in the DC file. Students with more than the assumed SCH for the degree are assumed not to be transfer students. The credits assumed for the degree are 60 semester and 90 quarter credits for associate and 120 semester credits and 180 quarter credits for Bac. Appropriate records in the EP or ES files can override these.

iv) A special selection is included for CNST. Only count their degrees earned by students whose Year and Term of First Enrollment in the SE is other than WI 1998. They used this term, the beginning of HEI, for many students who started earlier.

v) The reports show median time to degree, total graduates, total graduates in an extended program, and % of graduates assumed to be non-transfer by level, statewide, by sector and by campus. Total graduates include all level 03 and 05 records in any term of FY 1999.

vi) Spreadsheet with SQL is in O:\Data Queries\Time to Degree\TTD median via SPSS.xls

b) Credits to Degree for all Levels of Degree

i) Credits to degree are computed for degree Levels 03 (Assoc), 05 (Bac), 07 (Masters) in any term of FY 1999 for students who are assumed to be non-transfer students.

ii) For undergraduate degrees non-transfer students are determined by the SCH earned for the degree in the DC file. Students with more than the assumed SCH for the degree are assumed not to be transfer students. The credits assumed for the degree are 60 semester and 90 quarter credits for associate and 120 semester and 180 quarter credits for Bac. Appropriate records in the EP or ES files can override these. The institution calendar used in the term that the degree was earned is assumed to be the calendar for the student's entire duration.

iii) For graduate degrees all students are assumed to be non-transfer.

iv) Credits are calculated as semester credits and converted if the degree was earned on a quarter calendar. The schools calendar in the term of graduation is used to determine if the credits in the DC are semester or quarter credits. Exceptions are LCCC and CYCC in SM 1998 had a quarter calendar, but the DC records had semester hours.

v) Dual degrees are excluded from this calculation because they are reported as multiple records in the DC file and frequently with total credits necessary for both degrees. In particular if dual degrees are earned in the same term (with the same Level, but not the same Subject Code), both DC records are excluded and in they are earned in different terms, the degree earned latest is excluded.

vi) The reports show average credits to degree, total graduates and % of graduates assumed to be non-transfer by level, statewide, by sector and by campus. Total graduates include all level 03 and 05 records in any term of FY 1999

vii) Spreadsheet with SQL is in O:\Credits to Degree\Avg credits for Assoc Bac MS ver4.xls.

7) Employment Outcomes of Graduates

a) All graduates in SP term of FY 1998 and 1999 with a Degree Level of 03 (Assoc), 05 (Bac), 07 (Masters) or 09 (Doc) are selected from the DC. Also include Med and Law. Law is Level 10 and Subject Code 220101, Med is Level 10 and any of the Subject Codes in the Medical Funding Unit verification table. Note, CLEV has no Law grads recorded in SP 1998, therfore, their Law grads represent only one year.

b) To determine their residency, we join with the appropriate SE record based on the ADM area stored in the DC.

c) To see if they returned to school in the year after their graduation, we join with the SN for any term later than their Year and Term Degree Earned in the DC and not more than one year later.

d) To see if they had employment in Ohio according to the OBES, we join with the OBES data to see if they were employed in 4th quarter of the Year Degree Earned in the DC. If there was more than one OBES record we sum the wages in the 4 th quarter. The annual wages shown on the report is the sum of all amounts earned in the 4th quarter times 4. Annual wage < $16K is assumed part time.

e) We summarize the DC records according to three residency types:
   i) Ohio if the State of Residency in the SE is OH,
   ii) Foreign if the State of Residency in the SE is ZZ,
   iii) Other State Otherwise.

f) We summarize the employment outcome according to four cases:
   i) Returned to school and working in Ohio,
   ii) Returned to school and not working in Ohio,
   iii) Did not return to school and working in Ohio,
   iv) Did not return to school and not working in Ohio.

g) We summarize Major Field of Study by the first two digits of Subject Code and map to the subject categories used in the "Common Data Set".

h) Since some graduates have dual majors or degrees, generating multiple records in the DC for the same Year, Term and Subject Code, we summarize the selected DC records by Student ID, Year, Term, Level of Degree and Subject Code, and in the case of multiple Subject Codes we substituted the generic subject Multiple Fields.

i) Subject is the 2 digit Subject Code mapped to the Common Data Set discipline.

j) Spreadsheet with SQL is in O:\Data Queries\Obes\PR 10 ver 5 reports.xls.

8) Degrees Earned by Major Subject Code and Level

a) Count all degrees earned in any term of FY 1999 broken down by Campus, Level and the Two high order digits of Subject Code.

b) The 2 digit Subject Codes are mapped to a field using the mapping from the "Common Data Set", which is listed below.

c) Spreadsheet with SQL is in O:\Data Queries\Degrees\Degrees for Performance Report 2000.xls

9) Mobility and Concurrent Enrollments

a) Undergrad students enrolled in SP 2000 and SP 1999are counted as mobile if they were enrolled in any other campus in any pervious term of HEI (i.e. back to WI 1998).

b) Undergrad students enrolled in AU 1999 and AU 1998 are counted as concurrent if they were enrolled in any other campus in AU 1999.

c) Spreadsheet is at O:\Data Queries\Student Mobility Queries\Revised Queries for Performance Report.xls.

10) Comparing GPAs for Transfer Students to Student who did not Transfer

a) Students in AU 1999 with undergraduate Rank = ("FR","SO","JR","SR") attending university main campuses were selected. We excluded students with zero cumulative GPA hours in AU 1999 because this might represent incomplete data reporting. They were divided into three groups:
   i) Those who attended a two year school in SP 1999,
   ii) Those who attended a two year campus previously, in WI 1999, or WI, SP or AU 1998, but not in SP 1999,
   iii) Those who did not attend a two year campus previously in any of the above terms.

b) The average cumulative GPA in AU 1999 was calculated for each group by the campus they attended in AU 1999. This average GPA for a campus just averages the GPAs of the students in

the aggregation, i.e. it does not sum points and hours within the aggregate. This means that a new student's GPA weighs as heavily as one who has been at the school for a long time. At the state level we multiply the GPA for the campus and rank by the number of students, i.e. a weighted average.

c) Spreadsheet is at O:\Data Queries\GPA Studies\transfer gpa vs others ver2.xls.

11) I & G Cost for Undergraduates

a) The Unrestricted I & G cost per FTE for undergraduate instruction was calculated using the RA Query on the WWW in the Restricted Query area. We selected all levels (G, V, T, B) and calculated the cost on a per campus basis. The Level we use is the same as the Course Level except in the case of an undergraduate students in graduate courses, Level B is used.

b) Spreadsheet is at O:\Data Queries\I & G Cost\cost of ug.xls.

12) Enrollments in AU 1999

a) Simply count student enrollments (SN records) by campus in AU 1999.

b) Break them down by undergrad (Ranks "FR","SO","JR","SR","NU" for undergrads, Ranks "MS","PR","NM" for masters and Prof., and Ranks "DS","ND" for doctoral.

c) Spreadsheet is at O:\Data Queries\Headcounts\AU 1999 by campus for PR.xls.

13) Graduation Rates
14) Board Examination Passing Rates

Files used in the Performance Reports

1. CN: AU 1998, WI 1999, SP 1999, SM 1999, AU 1999
2. ST: AU 1998
3. SN: SM 1998, AU 1998, WI 1999, SP 1999, SM 1999, AU 1999, WI 2000
4. SE (for students with SN in SM or AU 1998, and students with DC in FY 1998 or 1999)
5. CI: AU 1998, WI 1999, SP 1999, SM 1999, AU 1999
6. RF (for students SN in SM or AU 1998, and students with DC in FY 1998 or 1999)
7. DC: FY 1998, FY 1999
8. EP: FY 1998, FY 1999
9. ES: FY 1998, FY 1999

2 Digit Subject Codes mapped to field via the Common Dataset Mapping

X01	Agriculture
X02	Agriculture
X04	Architecture
X05	Area and ethnic studies
X26	Biological/life sciences
X08	Business/marketing
X52	Business/marketing
X09	Communications/communication technologies
X10	Communications/communication technologies
X11	Computer and information sciences
X13	Education
X14	Engineering/engineering technologies
X15	Engineering/engineering technologies
X23	English
X16	Foreign languages and literature
X51	Health professions and related sciences
X19	Home economics and vocational home economics
X20	Home economics and vocational home economics
X30	Interdisciplinary studies

X22	Law/legal studies
X24	Liberal arts/general studies
X25	Library science
X27	Mathematics
X28	Military science and technologies
X29	Military science and technologies
X03	Natural resources/environmental science
X31	Parks and recreation
X12	Personal and miscellaneous services
X38	Philosophy, religion, theology
X39	Philosophy, religion, theology
X40	Physical sciences
X41	Physical sciences
X43	Protective services/public administration
X44	Protective services/public administration
X42	Psychology
X45	Social sciences and history
X46	Trade and industry
X47	Trade and industry
X48	Trade and industry
X49	Trade and industry
X50	Visual and performing arts
Xdu	Multiple Degrees
X32	Basic Skills
XTR	Transfer Module

SQL

1. Characteristics of the Faculty Teaching First Year Students

```
/* Basic Steps ***********************************/
/*Step 1 Join CN ST SN SE, sum to instructor*/
SELECT  crse_enroll.campus_code,
 crse_sect_taught.obrid,
 inst_campus.campus_type_code,
 inst_campus.campus_name,
 inst_campus.inst_code,
Sum( crse_enroll.crse_enroll_cr_hrs) instructor_SCH
into #step1
FROM crse_enroll, crse_sect_taught, inst_campus, stud_entrance, stud_enroll
WHERE ( crse_enroll.crse_sect_id =  crse_sect_taught.crse_sect_id)
AND ( crse_enroll.crse_id =  crse_sect_taught.crse_id)
AND ( crse_enroll.term_code =  crse_sect_taught.term_code)
AND ( crse_enroll.yr_num =  crse_sect_taught.yr_num)
AND ( crse_enroll.campus_code =  crse_sect_taught.campus_code)
AND  (crse_sect_taught.campus_code =  inst_campus.campus_code)
AND ( crse_enroll.term_code =  stud_enroll.term_code)
AND ( crse_enroll.yr_num =  stud_enroll.yr_num)
AND ( crse_enroll.campus_code =  stud_enroll.campus_code)
AND ( crse_enroll.obrid =  stud_enroll.obrid)
AND ( crse_enroll.admis_area_code =  stud_entrance.admis_area_code)
AND ( crse_enroll.inst_code =  stud_entrance.inst_code)
AND ( crse_enroll.obrid =  stud_entrance.obrid)
AND  (stud_enroll.rank_code In ("FR","NU"))
AND (( crse_enroll.yr_num)=1998)
AND (( crse_enroll.term_code)="AU")
AND (( stud_entrance.first_enroll_yr_num)=1998)
AND (( stud_entrance.first_enroll_term_code) In ("SM","AU"))
/*AND crse_enroll.campus_code ="CCCE"*/
GROUP BY  crse_enroll.campus_code,  crse_sect_taught.obrid,
 inst_campus.campus_type_code,
 inst_campus.campus_name,
 inst_campus.inst_code
/*Step 2 Join DC classify the instructor*/
SELECT  #step1.campus_type_code Campus_Type,
 #step1.inst_code Inst,
 #step1.campus_code Campus,
 fac_demographics.obrid Faculty_ID,
CASE WHEN appt_code In ("09","11") THEN "Full Time"
WHEN appt_code="GA" THEN "Graduate Asst"
WHEN fac_tenure_code="NT" THEN "Part Time, no Tenure"
ELSE "Part Time, Tenure or Ten Track" END Faculty_Type,
CASE WHEN appt_code="GA" THEN "GA"
ELSE fac_rank_code END Faculty_Rank,
CASE WHEN fac_hi_degree_code In ("A","B","C") THEN "Undergrad Degree"
```

```
WHEN fac_hi_degree_code In ("D","E","M","P") THEN "Grad Degree"
ELSE "Other Degree" END Highest_Degree,
Sum( instructor_SCH)  SCH,
 #step1.campus_name Campus_Name,
 inst_campus.campus_name Inst_Name
INTO #step2
FROM  #step1, fac_demographics, inst_campus
WHERE #step1.obrid =  fac_demographics.obrid
AND  #step1.inst_code =  fac_demographics.inst_code
AND  fac_demographics.inst_code =  inst_campus.inst_code
AND  fac_demographics.work_cat_code In ("FA","IR")
AND  fac_demographics.year_num = 1999
AND  inst_campus.campus_code=inst_campus.inst_code
GROUP BY  #step1.campus_type_code,
 #step1.inst_code,  #step1.campus_code,
 fac_demographics.obrid,
CASE WHEN appt_code In ("09","11") THEN "Full Time"
WHEN appt_code="GA" THEN "Graduate Asst"
WHEN fac_tenure_code="NT" THEN "Part Time, no Tenure"
ELSE "Part Time, Tenure or Ten Track" END,
CASE WHEN appt_code="GA" THEN "GA"
ELSE fac_rank_code END,
CASE WHEN fac_hi_degree_code In ("A","B","C") THEN "Undergrad Degree"
WHEN fac_hi_degree_code In ("D","E","M","P") THEN "Grad Degree"
ELSE "Other Degree" END,
  #step1.campus_name,  inst_campus.campus_name
/*Step 3 sum to faculty type, rank and highest degree*/
SELECT  #step2.Campus_Type,
 #step2.Inst,
#step2.Campus,
 #step2.Faculty_Type,
 #step2.Faculty_Rank,
 #step2.Highest_Degree,
Sum( #step2.SCH) AS group_SCH,
Count(*) AS Number_of_Teachers,
 #step2.Inst_Name,
 #step2.Campus_Name
INTO #step3
FROM  #step2
GROUP BY  #step2.Campus_Type,
 #step2.Inst,
#step2.Campus,
 #step2.Faculty_Type,
 #step2.Faculty_Rank,
 #step2.Highest_Degree,
 #step2.Inst_Name,
 #step2.Campus_Name
SELECT * from #step3
/*Full Time Part Time Reprots **********************************/
```

```
/*step4 sum to campus for full time*/
SELECT Campus_Type,
Inst_Name,
Campus_Name,
Sum(group_SCH) campus_sch,
Sum(CASE WHEN Faculty_Type="Full Time" THEN group_SCH ELSE 0 END) full_time_sch
INTO #step4
FROM #step3
GROUP BY  Campus_Type,Inst_Name,Campus_Name
/*step5 report full time*/
SELECT Campus_Type,
Inst_Name,
Campus_Name,
full_time_sch/campus_sch percent_full_time
FROM #step4
/*step5a sum to sector for full time part time*/
SELECT Campus_Type,
Sum(group_SCH) campus_sch,
Sum(CASE WHEN Faculty_Type="Full Time" THEN group_SCH ELSE 0 END) full_time_sch,
Sum(CASE WHEN Faculty_Type="Graduate Asst" THEN group_SCH ELSE 0 END) ga_sch,
Sum(CASE WHEN Faculty_Type="Part Time, no Tenure" THEN group_SCH ELSE 0 END) ptnt_sch,
Sum(CASE WHEN Faculty_Type="Part Time, Tenure or Ten Track" THEN group_SCH ELSE 0 END) pttn_sch
INTO #step5
FROM #step3
GROUP BY  Campus_Type
/*step6 report sector full time part time*/
SELECT Campus_Type,
full_time_sch/campus_sch percent_full_time,
ga_sch/campus_sch percent_ga,
ptnt_sch/campus_sch percent_ptnt,
pttn_sch/campus_sch percent_pttn
FROM #step5
/*step7 sum to state for full time part time*/
SELECT
Sum(group_SCH) campus_sch,
Sum(CASE WHEN Faculty_Type="Full Time" THEN group_SCH ELSE 0 END) full_time_sch,
Sum(CASE WHEN Faculty_Type="Graduate Asst" THEN group_SCH ELSE 0 END) ga_sch,
Sum(CASE WHEN Faculty_Type="Part Time, no Tenure" THEN group_SCH ELSE 0 END) ptnt_sch,
Sum(CASE WHEN Faculty_Type="Part Time, Tenure or Ten Track" THEN group_SCH ELSE 0 END) pttn_sch
INTO #step7
FROM #step3
/*step8 report state full time part time*/
SELECT
full_time_sch/campus_sch percent_full_time,
ga_sch/campus_sch percent_ga,
ptnt_sch/campus_sch percent_ptnt,
pttn_sch/campus_sch percent_pttn
FROM #step7
/*Rank Reports  ****************************************/
```

```
/*step10 sum to campus for senior rank*/
SELECT Campus_Type,
Inst_Name,
Campus_Name,
Sum(group_SCH) campus_sch,
Sum(CASE WHEN Faculty_Rank in("PR","AP") THEN group_SCH ELSE 0 END) senior_sch
INTO #step10
FROM #step3
WHERE Campus_Type in ("UM","UB")
GROUP BY  Campus_Type,Inst_Name,Campus_Name
/*step11report senior by campus*/
SELECT Campus_Type,
Inst_Name,
Campus_Name,
senior_sch/campus_sch percent_senior
FROM #step10
/*step12 sum to sector for rank*/
SELECT Campus_Type,
Sum(group_SCH) campus_sch,
Sum(CASE WHEN Faculty_Rank="GA" THEN group_SCH ELSE 0 END) ga_sch,
Sum(CASE WHEN Faculty_Rank in ("NR","OT","UK") THEN group_SCH ELSE 0 END) other_sch,
Sum(CASE WHEN Faculty_Rank="AP" THEN group_SCH ELSE 0 END) ap_sch,
Sum(CASE WHEN Faculty_Rank="AS" THEN group_SCH ELSE 0 END) as_sch,
Sum(CASE WHEN Faculty_Rank="IN" THEN group_SCH ELSE 0 END) in_sch,
Sum(CASE WHEN Faculty_Rank="LE" THEN group_SCH ELSE 0 END) le_sch,
Sum(CASE WHEN Faculty_Rank="PR" THEN group_SCH ELSE 0 END) pr_sch
INTO #step12
FROM #step3
WHERE Campus_Type in ("UM","UB")
GROUP BY  Campus_Type
/*step13 report sector rank*/
SELECT Campus_Type,
pr_sch/campus_sch percent_pr,
ap_sch/campus_sch percent_ap,
as_sch/campus_sch percent_as,
ga_sch/campus_sch percent_ga,
in_sch/campus_sch percent_in,
le_sch/campus_sch percent_le,
other_sch/campus_sch percent_other,
campus_sch/campus_sch percent_total
FROM #step12
/*step14 sum to state for rank*/
SELECT
Sum(group_SCH) campus_sch,
Sum(CASE WHEN Faculty_Rank="GA" THEN group_SCH ELSE 0 END) ga_sch,
Sum(CASE WHEN Faculty_Rank in ("NR","OT","UK") THEN group_SCH ELSE 0 END) other_sch,
Sum(CASE WHEN Faculty_Rank="AP" THEN group_SCH ELSE 0 END) ap_sch,
Sum(CASE WHEN Faculty_Rank="AS" THEN group_SCH ELSE 0 END) as_sch,
Sum(CASE WHEN Faculty_Rank="IN" THEN group_SCH ELSE 0 END) in_sch,
```

```
Sum(CASE WHEN Faculty_Rank="LE" THEN group_SCH ELSE 0 END) le_sch,
Sum(CASE WHEN Faculty_Rank="PR" THEN group_SCH ELSE 0 END) pr_sch
INTO #step14
FROM #step3
WHERE Campus_Type in ("UM","UB")
/*step15 report state Rank*/
SELECT
pr_sch/campus_sch percent_pr,
ap_sch/campus_sch percent_ap,
as_sch/campus_sch percent_as,
ga_sch/campus_sch percent_ga,
in_sch/campus_sch percent_in,
le_sch/campus_sch percent_le,
other_sch/campus_sch percent_other,
campus_sch/campus_sch percent_total
From #step14
/*Degree Reports  ************************************/
/*step16 sum to campus for grad degree*/
SELECT Campus_Type,
Inst_Name,
Campus_Name,
Sum(group_SCH) campus_sch,
Sum(CASE WHEN Highest_Degree = "Grad Degree" THEN group_SCH ELSE 0 END) grad_sch
INTO #step16
FROM #step3
WHERE Campus_Type in ("UB","CC","SC","TC")
GROUP BY  Campus_Type,Inst_Name,Campus_Name
/*step17report grad degree by campus*/
SELECT Campus_Type,
Inst_Name,
Campus_Name,
grad_sch/campus_sch percent_grad
FROM #step16
/*step18 sum to sector for highest degree*/
SELECT Campus_Type,
Sum(group_SCH) campus_sch,
Sum(CASE WHEN Highest_Degree = "Grad Degree" THEN group_SCH ELSE 0 END) grad_sch,
Sum(CASE WHEN Highest_Degree = "Undergrad Degree" THEN group_SCH ELSE 0 END) ug_sch,
Sum(CASE WHEN Highest_Degree = "Other Degree" THEN group_SCH ELSE 0 END) other_sch
INTO #step18
FROM #step3
WHERE Campus_Type in ("UB","CC","SC","TC")
GROUP BY  Campus_Type
/*step19 report sector highest degree*/
SELECT Campus_Type,
grad_sch/campus_sch percent_grad,
ug_sch/campus_sch percent_ug,
other_sch/campus_sch percent_other,
campus_sch/campus_sch percent_total
```

```
FROM #step18
/*step20 sum to state for highest degree*/
SELECT
Sum(group_SCH) campus_sch,
Sum(CASE WHEN Highest_Degree = "Grad Degree" THEN group_SCH ELSE 0 END) grad_sch,
Sum(CASE WHEN Highest_Degree = "Undergrad Degree" THEN group_SCH ELSE 0 END) ug_sch,
Sum(CASE WHEN Highest_Degree = "Other Degree" THEN group_SCH ELSE 0 END) other_sch
INTO #step20
FROM #step3
WHERE Campus_Type in ("UB","CC","SC","TC")
/*step21 report state highest degree*/
SELECT
grad_sch/campus_sch percent_grad,
ug_sch/campus_sch percent_ug,
other_sch/campus_sch percent_other
From #step20
```

SQL for Faculty Qualifications

```
Summary by Inst
SELECT dbo_inst_campus.campus_type_code,
dbo_inst_campus.campus_name,
Count(*) AS [total faculty],
Sum(IIf([appt_code] In ("09","11"),1,0)) AS [Full Time],
Sum(IIf([fac_rank_code] In ("PR","AP"),1,0)) AS [Senior Fac],
Sum(IIf([fac_hi_degree_code] In ("D","E","M","P"),1,0)) AS [Grad Degree]
FROM dbo_fac_demographics
INNER JOIN dbo_inst_campus
ON dbo_fac_demographics.inst_code = dbo_inst_campus.inst_code
WHERE (((dbo_fac_demographics.yr_num)=1999)
AND ((dbo_inst_campus.inst_code)=[campus_code]))
GROUP BY dbo_inst_campus.campus_type_code,
dbo_inst_campus.campus_name;
```

2. Outcomes of Remedial Instruction

```
/*Step 0a create dev course table*/
CREATE TABLE #dev_course
(inst char(4),
course_id char(12),
field char(7))
/*Step 0b populate the dev course table*/
INSERT #dev_course values ('CLEV',      '010705',      'English')
INSERT #dev_course values ('CLRK',      'DEV 061',     'English')
INSERT #dev_course values ('CLRK',      'DEV 062',     'English')
INSERT #dev_course values ('CLRK',      'DEV 071',     'English')
INSERT #dev_course values ('CLRK',      'DEV 091',     'Math')
INSERT #dev_course values ('CLRK',      'DEV 101',     'Math')
INSERT #dev_course values ('CLRK',      'DEV 102',     'Math')
```

```
INSERT #dev_course values ('CLRK',      'DEV 103',      'Math')
INSERT #dev_course values ('KENT',      'ENG 10000',    'English')
INSERT #dev_course values ('KENT',      'MATH10005',    'Math')
INSERT #dev_course values ('MIAM',      'ENG001    ',   'English')
INSERT #dev_course values ('MIAM',      'ENG001A   ',   'English')

INSERT #dev_course values ('MIAM',      'ENG002    ',   'English')
INSERT #dev_course values ('MIAM',      'ENG002A   ',   'English')

INSERT #dev_course values ('MIAM',      'ENG007    ',   'English')
INSERT #dev_course values ('MIAM',      'MTH001    ',   'Math')
INSERT #dev_course values ('MIAM',      'MTH002    ',   'Math')
INSERT #dev_course values ('MIAM',      'MTH007    ',   'Math')
INSERT #dev_course values ('MIAM',      'MTH007E   ',   'Math')
INSERT #dev_course values ('MIAM',      'MTH007P   ',   'Math')
INSERT #dev_course values ('NCTC',      'DEV110    ',   'English')
INSERT #dev_course values ('NCTC',      'DEV115    ',   'English')
INSERT #dev_course values ('NCTC',      'DEV116    ',   'English')
INSERT #dev_course values ('NCTC',      'DEV120    ',   'English')
INSERT #dev_course values ('NCTC',      'DEV140    ',   'English')
INSERT #dev_course values ('NCTC',      'DEV160    ',   'English')
INSERT #dev_course values ('NCTC',      'DEV161    ',   'English')
INSERT #dev_course values ('NCTC',      'DEV180    ',   'English')
INSERT #dev_course values ('NCTC',      'MTH100    ',   'Math')
INSERT #dev_course values ('NCTC',      'MTH101    ',   'Math')
INSERT #dev_course values ('NSCC',      'ENG090',       'English')
INSERT #dev_course values ('RGCC',      'MTH11404',     'Math')

INSERT #dev_course values ('NSCC',      'MTH090',       'Math')
INSERT #dev_course values ('SNCL',      'DEV 085',      'Math')
INSERT #dev_course values ('TRRA',      'ENG051    ',   'English')
INSERT #dev_course values ('TRRA',      'ENG060    ',   'English')
INSERT #dev_course values ('TRRA',      'ENG071    ',   'English')
INSERT #dev_course values ('TRRA',      'ENG072    ',   'English')
INSERT #dev_course values ('TRRA',      'ENG081    ',   'English')
INSERT #dev_course values ('TRRA',      'ENG082    ',   'English')
INSERT #dev_course values ('TRRA',      'ENG099    ',   'English')
INSERT #dev_course values ('TRRA',      'MTH060    ',   'Math')
INSERT #dev_course values ('TRRA',      'MTH071    ',   'Math')
INSERT #dev_course values ('TRRA',      'MTH072    ',   'Math')
INSERT #dev_course values ('TRRA',      'MTH091    ',   'Math')
INSERT #dev_course values ('TRRA',      'MTH092    ',   'Math')
INSERT #dev_course values ('TRRA',      'MTH099    ',   'Math')
INSERT #dev_course values ('YNGS',      'UMATH 503',    'Math')

INSERT #dev_course values ('YNGS',      'UMATH 504',    'Math')

INSERT #dev_course values ('YNGS',      'UMATH 505',    'Math')
```

```
INSERT #dev_course values ('YNGS',                    'UMATH 511',          'Math')

INSERT #dev_course values ('YNGS',                    'UMATH 513',          'Math')
```

/*Step 3a Identify the developmental courses in the CI*/

```
SELECT
inst_code,
crse_id,
CASE
WHEN subject_field="Math" THEN "Math"
WHEN subject_field="English" THEN "English"
WHEN crse_inventory.subject_code In ("320101","320199") And (course_title Like "%MATH%" Or
course_title Like "%ALGEBRA%") THEN "Math"
WHEN crse_inventory.subject_code In ("320101","320199") And (course_title Like "%LIT%" Or
course_title Like "%READ%" Or course_title Like "%COMPO%" Or course_title Like "%WRIT%") THEN "English"
WHEN crse_inventory.subject_code In ("320101","320199") And course_title Like "%ENG%" And
course_title Not Like "%SEC%" THEN "English"
 ELSE "Other"
END field
INTO #step3a
FROM crse_inventory,  subject
WHERE crse_inventory.subject_code = subject.subject_code
AND crse_inventory.yr_num = subject.yr_num
AND crse_inventory.term_code = subject.term_code
AND (subject_field  = "Math"
  OR subject_field = "English"
  OR (crse_inventory.subject_code In ("320101","320199")
    AND (course_title Like "%MATH%" OR course_title Like "%ALGEBRA%"))
  OR (crse_inventory.subject_code In ("320101","320199")
    AND (course_title Like "%LIT%" Or course_title Like "%READ%" Or course_title Like "%COMPO%" Or
course_title Like "%WRIT%"))
  OR (crse_inventory.subject_code In ("320101","320199") And course_title Like "%ENG%" And course_title
Not Like "%SEC%"))
AND crse_inventory.yr_num=1998
AND crse_inventory.term_code="AU"
AND crse_inventory.crse_level_code="V"
```

/*Step 3b Append Dev Course IDs*/

```
INSERT INTO #step3a
(inst_code, crse_id, field)
SELECT
inst, course_id, field
FROM #dev_course
/*SELECT * FROM #step3a*/
```

/*Step 3b sum to elim dups*/

```
SELECT
inst_code, crse_id, Max(field) field
INTO #step3b
FROM #step3a
GROUP by inst_code, crse_id
```

```
/*Step 4 select students taking developmental in AU 1998*/
SELECT stud_enroll.campus_code,
 stud_enroll.obrid,
field,
CASE WHEN birth_yr  = "UNKN" or birth_yr <"1978" THEN "over 20"
ELSE "20 or under" END Age_Category,
Count(*) AS Dev_courses_taken
INTO #step4
FROM  stud_enroll, stud_entrance, crse_enroll, #step3a
WHERE stud_enroll.obrid =  stud_entrance.obrid
AND  stud_enroll.inst_code =  stud_entrance.inst_code
AND  stud_enroll.admis_area_code =  stud_entrance.admis_area_code
AND  stud_enroll.term_code =  crse_enroll.term_code
AND  stud_enroll.yr_num =  crse_enroll.yr_num
AND  stud_enroll.campus_code =  crse_enroll.campus_code
AND  stud_enroll.obrid =  crse_enroll.obrid
AND  crse_enroll.inst_code = #step3a.inst_code
AND  crse_enroll.crse_id = #step3a.crse_id
AND stud_enroll.yr_num=1998
AND  stud_enroll.term_code="AU"
AND  stud_enroll.rank_code In ("FR","NU")
AND  stud_entrance.first_enroll_yr_num=1998
AND  stud_entrance.first_enroll_term_code In ("SM","AU")
AND  crse_enroll.crse_enroll_cr_hrs>0
AND  crse_enroll.attempt_acad_cr_code="I"
AND  crse_enroll.crse_enroll_audit_sw="Y"
/*AND  stud_enroll.inst_code="CLRK"*/
GROUP BY  stud_enroll.campus_code,
 stud_enroll.obrid,
 #step3a.field,
CASE WHEN birth_yr  = "UNKN" or birth_yr <"1978" THEN "over 20"
ELSE "20 or under" END
/*Step 5 sum to campus for report*/
SELECT campus_code,
field,
Age_Category,
Count(*) AS Students_taking_dev,
0 extra_field1,
0 extra_field2
INTO #step5
FROM #step4
GROUP BY campus_code,
field,
Age_Category
/*Step 6 select subsequent enrollments in 1999*/
SELECT #step4.obrid,
#step4.campus_code,
#step4.field,
#step4.Age_Category,
```

```
crse_enroll.crse_id,
#dev_course.course_id,
crse_enroll.crse_enroll_audit_sw AS sub_completed
INTO #step6
FROM #step4, crse_enroll, crse_inventory, subject, #dev_course
WHERE #step4.obrid = crse_enroll.obrid
AND crse_enroll.crse_id *= #dev_course.course_id
AND crse_enroll.inst_code *= #dev_course.inst
/*AND #dev_course.course_id is null*/
AND #step4.campus_code = crse_enroll.campus_code
AND crse_enroll.term_code = crse_inventory.term_code
AND crse_enroll.yr_num = crse_inventory.yr_num
AND crse_enroll.inst_code = crse_inventory.inst_code
AND crse_enroll.crse_id = crse_inventory.crse_id
AND #step4.field = subject.subject_field
AND crse_enroll.term_code = subject.term_code
AND crse_enroll.yr_num = subject.yr_num
AND crse_inventory.subject_code = subject.subject_code
AND crse_enroll.yr_num=1999
AND crse_enroll.crse_enroll_cr_hrs>0
AND crse_enroll.attempt_acad_cr_code="I"
AND crse_inventory.crse_level_code<>"V"
/*select * from #step6*/
/*step 7 sum to student for unduplicated head count*/
SELECT campus_code,
obrid,
field,
 Age_Category,
Sum(CASE WHEN sub_completed="Y" THEN 1 ELSE 0 END)  sub_success,
Sum(CASE WHEN sub_completed="N" THEN 1 ELSE 0 END) sub_not_success
INTO #step7
FROM #step6
WHERE #step6.course_id is null
GROUP BY campus_code,
obrid,
field,
Age_Category
/*select * from #step7*/
/*Step 8 sum to campus*/
SELECT campus_code,
field,
Age_Category,
Sum(CASE WHEN sub_not_success=0 THEN 1 ELSE 0 END) All_successful,
Sum(CASE WHEN sub_success>0 And sub_not_success>0 THEN 1 ELSE 0 END) Some_successful,
Sum( CASE WHEN sub_success=0 THEN 1 ELSE 0 END) No_successful
INTO #step8
FROM #step7
GROUP BY campus_code,
field,
```

```
Age_Category
/*Step 9 join Rem with Continuing*/
SELECT
#step8.campus_code,
#step8.field,
#step8.Age_Category,
#step5.Students_taking_dev,
All_successful,
Some_successful,
No_successful,
inst_campus.campus_type_code sector,
inst_campus.campus_name campus,
inst_campus_1.campus_name inst
INTO #step9
FROM #step5, #step8, inst_campus, inst_campus inst_campus_1
WHERE
#step8.campus_code = #step5.campus_code
AND #step8.field = #step5.field
AND #step8.Age_Category = #step5.Age_Category
AND #step8.campus_code = inst_campus.campus_code
AND inst_campus.inst_code = inst_campus_1.inst_code
AND inst_campus_1.inst_code = inst_campus_1.campus_code
SELECT * from #step9
/*Campus report    ************************************/
/*Step 10 Campus report*/
SELECT
field,
sector,
inst,
campus,
All_successful+Some_successful+No_successful Persisting,
convert(float,All_successful)/convert(float,(All_successful+Some_successful+No_successful)) All_successful,
convert(float,Some_successful)/convert(float,(All_successful+Some_successful+No_successful))
Some_successful,
convert(float,No_successful)/convert(float,(All_successful+Some_successful+No_successful)) No_successful,
1 all_persisting_percent,
convert(float,(Students_taking_dev-(All_successful+Some_successful+No_successful)))/
convert(float,Students_taking_dev) percent_non_persisting
FROM #step9
WHERE Age_Category = "20 or under"
ORDER BY
field,
sector,
inst,
campus
/*Step 11 sector sum*/
SELECT
field,
sector,
```

```
Age_Category,
sum(Students_taking_dev) Students_taking_dev,
sum(All_successful) All_successful,
sum(Some_successful) Some_successful,
sum(No_successful) No_successful
INTO #step11
FROM #step9
GROUP BY
field,
sector,
Age_Category
/*Step 11a sector report*/
SELECT
field,
sector,
Age_Category,
convert(float,All_successful)/convert(float,(All_successful+Some_successful+No_successful)) All_successful,
convert(float,Some_successful)/convert(float,(All_successful+Some_successful+No_successful))
Some_successful,
convert(float,No_successful)/convert(float,(All_successful+Some_successful+No_successful)) No_successful,
1 all_persisting_percent,
convert(float,(All_successful+Some_successful+No_successful))/ convert(float,Students_taking_dev)
percent_continuing
FROM #step11
ORDER BY
field,
sector,
Age_Category
/*Step 12 statewide sum*/
SELECT
field,
Age_Category,
sum(Students_taking_dev) Students_taking_dev,
sum(All_successful) All_successful,
sum(Some_successful) Some_successful,
sum(No_successful) No_successful
INTO #step12
FROM #step9
GROUP BY
field,
Age_Category
/*Step 12a statewide report*/
SELECT
field,
Age_Category,
convert(float,All_successful)/convert(float,(All_successful+Some_successful+No_successful)) All_successful,
convert(float,Some_successful)/convert(float,(All_successful+Some_successful+No_successful))
Some_successful,
convert(float,No_successful)/convert(float,(All_successful+Some_successful+No_successful)) No_successful,
1 all_persisting_percent,
```

```
convert(float,(All_successful+Some_successful+No_successful))/ convert(float,Students_taking_dev)
percent_continuing
FROM #step12
ORDER BY
field,
Age_Category
```

3.Year to Year Persistence of Full time, First Time, Degree Seeking First Year Students (FTFTDSFR)

/*Step 1 SN CN SE select under grads in AU 1998*/

```
SELECT  stud_enroll.obrid,
 inst_campus.campus_type_code AU_1998_campus_type,
 stud_enroll.inst_code AU_1998_inst,
 stud_enroll.campus_code AU_1998_campus,
 stud_enroll.rank_code,
 stud_enroll.admis_area_code,
Sum( crse_enroll.crse_enroll_cr_hrs) SCH,
 stud_entrance.intention_code,
 stud_entrance.first_enroll_term_code Adm_FT,
 stud_entrance.first_enroll_yr_num Adm_FY,
 stud_entrance.first_time_in_college_sw
INTO #step1
FROM stud_enroll, crse_enroll, inst_campus, stud_entrance
WHERE stud_enroll.obrid =  crse_enroll.obrid
AND  stud_enroll.campus_code =  crse_enroll.campus_code
AND  stud_enroll.yr_num =  crse_enroll.yr_num
AND  stud_enroll.term_code =  crse_enroll.term_code
AND stud_enroll.campus_code =  inst_campus.campus_code
AND stud_enroll.admis_area_code =  stud_entrance.admis_area_code
AND  stud_enroll.inst_code =  stud_entrance.inst_code
AND  stud_enroll.obrid =  stud_entrance.obrid
AND stud_enroll.admis_area_code =  "UND"
AND stud_enroll.yr_num=1998
AND stud_enroll.term_code="AU"
/*AND stud_enroll.inst_code  = "RGCC"*/
GROUP BY  stud_enroll.obrid,
 inst_campus.campus_type_code,
 stud_enroll.inst_code,
 stud_enroll.campus_code,
 stud_enroll.rank_code,
stud_enroll.admis_area_code,
 stud_entrance.intention_code,
 stud_entrance.first_enroll_term_code,
 stud_entrance.first_enroll_yr_num,
 stud_entrance.first_time_in_college_sw
HAVING  stud_enroll.rank_code In ("FR","SO","NU","JR","SR")
AND  stud_enroll.admis_area_code="UND"
```

```
/*Step 2 Join RF*/
SELECT #step1.*,
CASE WHEN SCH>=12 THEN "Full" ELSE "Part" END Full_Part_Time,
rev_first_enroll.first_enroll_term_code Rev_FT,
rev_first_enroll.first_enroll_yr_num Rev_FY
INTO #step2
FROM #step1, rev_first_enroll
WHERE #step1.obrid *= rev_first_enroll.obrid
/*Step 3 Calculate FTFTDSFR*/
SELECT #step2.*,
(CASE WHEN (rank_code In ("FR","NU") Or Rev_FT is not null)
And ((Rev_FT is null And Adm_FY=1998 And Adm_FT In ("SM","AU"))
  Or (Rev_FY=1998 And Rev_FT In ("SM","AU") And rank_code In ("FR","NU","SO","JR","SR")))
 THEN 1 ELSE 0 END) first_term,

(CASE WHEN Rev_FY=1998 And Rev_FT In ("SM","AU") THEN 1 ELSE 0 END) first_term_from_RF,

(CASE WHEN first_time_in_college_sw="Y" Or AU_1998_campus in("EDSN","CLRK") THEN 1 ELSE 0 END)
first_time,

(CASE WHEN (AU_1998_campus_type="UM" And rank_code<>"NU" )
  Or intention_code In ("06","07","08")
  Or (AU_1998_campus_type="UB" And AU_1998_inst In ("CINC","KENT","BGSU") And rank_code<>"NU")
  Or (AU_1998_inst = "RGCC" and intention_code = "04")
 THEN 1 ELSE 0 END) degree_intent,

(CASE WHEN (Full_Part_Time="Full")
And (rank_code In ("FR","NU") Or Rev_FT is not null)
And ((Rev_FT is null And Adm_FY=1998 And Adm_FT In ("SM","AU"))
  Or (Rev_FY=1998 And Rev_FT In ("SM","AU") And rank_code In ("FR","NU","SO","JR","SR")))
And (first_time_in_college_sw="Y" Or AU_1998_campus in("EDSN","CLRK"))
And ((AU_1998_campus_type="UM" And rank_code<>"NU")
   Or intention_code In ("06","07","08")
   Or (AU_1998_campus_type="UB" And AU_1998_inst In ("CINC","KENT","BGSU") And rank_code<>"NU")
  Or (AU_1998_inst = "RGCC" and intention_code = "04"))
 THEN 1 ELSE 0 END)  FTFTDSFR
INTO #step3
FROM #step2
/*Step 4 sum to campus*/
SELECT  #step3.AU_1998_campus_type,
 #step3.AU_1998_inst,
 #step3.AU_1998_campus,
Sum(CASE WHEN Full_Part_Time ="Full" THEN 1 ELSE 0 END) Num_Full_Time,
Sum(first_term) Num_first_term,
sum(case when first_term + first_time =2 then 1 else 0 end) Num_first_term_and_time,
Sum(first_term_from_RF) Num_first_term_from_RF,
Sum( first_time) Num_first_time,
Sum(degree_intent) Num_degree_seekeers,
Sum(FTFTDSFR) Num_FTFTDSFR
```

```
INTO #step4
FROM  #step3
GROUP BY AU_1998_campus_type,
AU_1998_inst,
AU_1998_campus
/*Step 5 join SN from AU 1999 for FTFTDSFR*/
SELECT  #step3.obrid,
 #step3.AU_1998_campus_type,
 #step3.AU_1998_inst,
 #step3.AU_1998_campus,
 #step3.FTFTDSFR,
(CASE WHEN AU_1998_campus=stud_enroll.campus_code THEN "campus"
WHEN AU_1998_inst=stud_enroll.inst_code THEN "inst" ELSE "state" END) Campus_Rel
INTO #step5
FROM  #step3, stud_enroll
WHERE  #step3.obrid = stud_enroll.obrid
AND #step3.FTFTDSFR=1
AND stud_enroll.yr_num=1999
AND stud_enroll.term_code="AU"
/*Step 6 Sum to Obrid to pick best campus, if FTFTDSFR enrolled in > 1 campus in AU 1999*/
SELECT  #step5.obrid,
 #step5.AU_1998_campus_type,
 #step5.AU_1998_inst,
 #step5.AU_1998_campus,
 #step5.FTFTDSFR,
Min( #step5.Campus_Rel) Closest_campus
INTO #step6
FROM  #step5
GROUP BY  #step5.obrid,
 #step5.AU_1998_campus_type,
 #step5.AU_1998_inst,
 #step5.AU_1998_campus,
 #step5.FTFTDSFR
/*Step 7 sum to campus*/
SELECT  #step6.AU_1998_campus_type,
 #step6.AU_1998_inst,
 #step6.AU_1998_campus,
Sum(CASE WHEN Closest_campus="campus" THEN 1 ELSE 0 END) Same_campus,
Sum(CASE WHEN Closest_campus="inst" THEN 1 ELSE 0 END) Same_inst,
Sum(CASE WHEN Closest_campus="state" THEN 1 ELSE 0 END) Different_inst,
Count(*) AS Retained
INTO #step7
FROM  #step6
GROUP BY  #step6.AU_1998_campus_type,
 #step6.AU_1998_inst,
 #step6.AU_1998_campus
/*Step8 join step 4 and 7*/
SELECT
#step4.AU_1998_campus_type,
```

```
#step4.AU_1998_inst,
#step4.AU_1998_campus,
Num_Full_Time,
Num_first_term,
Num_first_term_and_time,
Num_first_term_from_RF,
Num_first_time,
Num_degree_seekeers,
Num_FTFTDSFR,
Same_campus,
Same_inst,
Different_inst,
Retained
INTO #step8
FROM #step4, #step7
WHERE #step4.AU_1998_campus = #step7.AU_1998_campus
SELECT * FROM #step8
/*Step9 campus report*/
SELECT
#step8.AU_1998_campus_type,
inst_campus_1.campus_name,
inst_campus.campus_name,
convert(float,(Same_campus+Same_inst))/convert(float,Num_FTFTDSFR) Percent_same_inst,
convert(float,(Same_campus+Same_inst+Different_inst))/convert(float,Num_FTFTDSFR) Percent_any_public,
convert(float,Num_FTFTDSFR)/convert(float,Num_first_term_and_time) Percent_FTFTDSFR
FROM #step8,inst_campus,inst_campus inst_campus_1
WHERE
AU_1998_campus  = inst_campus.campus_code
AND inst_campus.inst_code = inst_campus_1.inst_code
AND inst_campus_1.campus_code = inst_campus_1.inst_code
/*Step10 sum to sector*/
SELECT
#step8.AU_1998_campus_type,
sum(Same_campus+Same_inst) same_inst,
sum(Same_campus+Same_inst+Different_inst) any_public,
sum(Num_FTFTDSFR) FTFTDSFR,
sum(Num_first_term_and_time) first_term_and_time
INTO #step10
FROM #step8
GROUP BY #step8.AU_1998_campus_type
/*Step11 sector report*/
SELECT
AU_1998_campus_type,
convert(float,(same_inst))/convert(float,FTFTDSFR) Percent_same_inst,
convert(float,(any_public))/convert(float,FTFTDSFR) Percent_any_public,
convert(float,FTFTDSFR)/convert(float,first_term_and_time) Percent_FTFTDSFR
FROM #step10
/*Step12 sum to state*/
SELECT
```

```
sum(Same_campus+Same_inst) same_inst,
sum(Same_campus+Same_inst+Different_inst) any_public,
sum(Num_FTFTDSFR) FTFTDSFR,
sum(Num_first_term_and_time) first_term_and_time
INTO #step12
FROM #step8
/*Step13 state report*/
SELECT
convert(float,(same_inst))/convert(float,FTFTDSFR) Percent_same_inst,
convert(float,(any_public))/convert(float,FTFTDSFR) Percent_any_public,
convert(float,FTFTDSFR)/convert(float,first_term_and_time) Percent_FTFTDSFR
FROM #step12
```

4. Demographics of undergraduates

Access SQL (used this due to formatting problems with SQL Advantage)
Step 1join SN CN SE

```
SELECT dbo_stud_enroll.yr_num,
dbo_stud_enroll.term_code,
dbo_stud_enroll.campus_code,
dbo_stud_enroll.obrid,
dbo_stud_entrance.birth_yr,
dbo_stud_entrance.race_ethnic_code,
dbo_stud_entrance.state_abbrev,
Sum(dbo_crse_enroll.crse_enroll_cr_hrs) AS [Student SCH]
FROM (dbo_stud_enroll
INNER JOIN dbo_stud_entrance
ON (dbo_stud_enroll.admis_area_code = dbo_stud_entrance.admis_area_code)
AND (dbo_stud_enroll.inst_code = dbo_stud_entrance.inst_code)
AND (dbo_stud_enroll.obrid = dbo_stud_entrance.obrid))
INNER JOIN dbo_crse_enroll
ON (dbo_stud_enroll.term_code = dbo_crse_enroll.term_code)
AND (dbo_stud_enroll.yr_num = dbo_crse_enroll.yr_num)
AND (dbo_stud_enroll.campus_code = dbo_crse_enroll.campus_code)
AND (dbo_stud_enroll.obrid = dbo_crse_enroll.obrid)
WHERE (((dbo_stud_enroll.admis_area_code)="UND"))
GROUP BY dbo_stud_enroll.yr_num,
dbo_stud_enroll.term_code,
dbo_stud_enroll.campus_code,
dbo_stud_enroll.obrid,
dbo_stud_entrance.birth_yr,
dbo_stud_entrance.race_ethnic_code,
dbo_stud_entrance.state_abbrev
HAVING (((dbo_stud_enroll.yr_num)=1999)
AND ((dbo_stud_enroll.term_code)="AU"));
```

Step 2 sum to campus

```
SELECT [dug s1 join SN CN SE].campus_code,
Count(*) AS students,
Sum(IIf([Student SCH]<12,0,1)) AS [full time],
Sum(IIf([Student SCH]<12,1,0)) AS [part time],
Sum(IIf([birth_yr]="UNKN" Or [birth_yr]<"1975",1,0)) AS [over 24],
Sum(IIf([birth_yr]="UNKN" Or [birth_yr]<"1975",0,1)) AS under25,
Sum(IIf([race_ethnic_code]="AI",1,0)) AS AI,
Sum(IIf([race_ethnic_code]="AS",1,0)) AS [AS],
Sum(IIf([race_ethnic_code]="BL",1,0)) AS BL,
Sum(IIf([race_ethnic_code]="HS",1,0)) AS HS,
Sum(IIf([race_ethnic_code]="NR",1,0)) AS NR,
Sum(IIf([race_ethnic_code]="UK",1,0)) AS UK,
Sum(IIf([race_ethnic_code]="WH",1,0)) AS WH,
Sum(IIf([state_abbrev]="OH",1,0)) AS Ohio,
Sum(IIf([state_abbrev]="ZZ",1,0)) AS [Foreign],
Sum(IIf([state_abbrev] Not In ("OH","ZZ"),1,0)) AS [Other State]
FROM [dug s1 join SN CN SE]
GROUP BY [dug s1 join SN CN SE].campus_code;
```

Step 3 Campus %s

```
SELECT dbo_inst_campus.campus_type_code,
dbo_inst_campus_1.campus_name,
dbo_inst_campus.campus_name,
[dug s2 sum to campus].students,
[full time]/[students] AS [full time percent],
[part time]/[students] AS [part time percent],
[over 24]/[students] AS [over24 percent],
[under25]/[students] AS [under 25 percent],
[AI]/[students] AS [AI percent],
[AS]/[students] AS [AS percent],
[BL]/[students] AS [BL percent],
[HS]/[students] AS [HS percent],
[NR]/[students] AS [NR percent],
[UK]/[students] AS [UK percent],
 [WH]/[students] AS [WH percent],
[Ohio]/[students] AS [Ohio percent],
[Other State]/[students] AS [other state percent],
[Foreign]/[students] AS [foreign percent]
FROM ([dug s2 sum to campus]
INNER JOIN dbo_inst_campus
ON [dug s2 sum to campus].campus_code = dbo_inst_campus.campus_code)
INNER JOIN dbo_inst_campus AS dbo_inst_campus_1
ON dbo_inst_campus.inst_code = dbo_inst_campus_1.inst_code
WHERE (((dbo_inst_campus_1.inst_code)=[dbo_inst_campus_1]![campus_code]));
```

Step 4 sum to sector

```
SELECT dbo_inst_campus.campus_type_code,
Count(*) AS students,
Sum(IIf([Student SCH]<12,0,1)) AS [full time],
Sum(IIf([Student SCH]<12,1,0)) AS [part time],
```

```
Sum(IIf([birth_yr]="UNKN" Or [birth_yr]<"1975",1,0)) AS [over 24],
Sum(IIf([birth_yr]="UNKN" Or [birth_yr]<"1975",0,1)) AS under25,
Sum(IIf([race_ethnic_code]="AI",1,0)) AS AI,
Sum(IIf([race_ethnic_code]="AS",1,0)) AS [AS],
Sum(IIf([race_ethnic_code]="BL",1,0)) AS BL,
Sum(IIf([race_ethnic_code]="HS",1,0)) AS HS,
Sum(IIf([race_ethnic_code]="NR",1,0)) AS NR,
Sum(IIf([race_ethnic_code]="UK",1,0)) AS UK,
Sum(IIf([race_ethnic_code]="WH",1,0)) AS WH,
Sum(IIf([state_abbrev]="OH",1,0)) AS Ohio,
Sum(IIf([state_abbrev]="ZZ",1,0)) AS [Foreign],
Sum(IIf([state_abbrev] Not In ("OH","ZZ"),1,0)) AS [Other State]
FROM [dug s1 join SN CN SE]
INNER JOIN dbo_inst_campus
ON [dug s1 join SN CN SE].campus_code = dbo_inst_campus.campus_code
GROUP BY dbo_inst_campus.campus_type_code;
Step 5 sector report
SELECT [dug s4 sum to sector].campus_type_code,
[full time]/[students] AS FT,
[dug s4 sum to sector].[part time]/[students] AS PT,
[dug s4 sum to sector].[over 24]/[students] AS Over_24,
[dug s4 sum to sector].under25/[students] AS Under_25,
[dug s4 sum to sector].AI/[students] AS AI,
[dug s4 sum to sector].AS/[students] AS [AS],
[dug s4 sum to sector].BL/[students] AS BL,
[dug s4 sum to sector].HS/[students] AS HS,
[dug s4 sum to sector].NR/[students] AS NR,
[dug s4 sum to sector].UK/[students] AS UK,
[dug s4 sum to sector].WH/[students] AS WH,
[dug s4 sum to sector].Ohio/[students] AS Ohio,
[dug s4 sum to sector].Foreign/[students] AS [Foreign],
[dug s4 sum to sector].[Other State]/[students] AS Other_state
FROM [dug s4 sum to sector];
Step6 state sum
SELECT Count(*) AS students,
Sum(IIf([Student SCH]<12,0,1)) AS [full time],
Sum(IIf([Student SCH]<12,1,0)) AS [part time],
Sum(IIf([birth_yr]="UNKN" Or [birth_yr]<"1975",1,0)) AS [over 24],
Sum(IIf([birth_yr]="UNKN" Or [birth_yr]<"1975",0,1)) AS under25,
Sum(IIf([race_ethnic_code]="AI",1,0)) AS AI,
Sum(IIf([race_ethnic_code]="AS",1,0)) AS [AS],
Sum(IIf([race_ethnic_code]="BL",1,0)) AS BL,
Sum(IIf([race_ethnic_code]="HS",1,0)) AS HS,
Sum(IIf([race_ethnic_code]="NR",1,0)) AS NR,
Sum(IIf([race_ethnic_code]="UK",1,0)) AS UK,
Sum(IIf([race_ethnic_code]="WH",1,0)) AS WH,
Sum(IIf([state_abbrev]="OH",1,0)) AS Ohio,
Sum(IIf([state_abbrev]="ZZ",1,0)) AS [Foreign],
Sum(IIf([state_abbrev] Not In ("OH","ZZ"),1,0)) AS [Other State]
```

```
FROM [dug s1 join SN CN SE];
Step 7 state report
SELECT [full time]/[students] AS FT,
[part time]/[students] AS PT,
[over 24]/[students] AS Over_24,
[under25]/[students] AS Under_25,
AI/[students] AS Percent_AI,
[AS]/[students] AS Percent_AS,
BL/[students] AS Percent_BL,
HS/[students] AS Percent_HS,
NR/[students] AS Percent_NR,
UK/[students] AS Percent_UK,
WH/[students] AS Percent_WH,
Ohio/[students] AS Percent_Ohio,
Foreign/[students] AS Percent_Foreign,
[Other State]/[students] AS Percent_Other_state
FROM [dug s6 sum to state];
```

5. Demographics for graduates

```
/* step1 join SN SE*/
SELECT DISTINCT
stud_enroll.yr_num,
stud_enroll.term_code,
stud_enroll.inst_code,
stud_enroll.obrid,
stud_entrance.state_abbrev
INTO #step1
FROM stud_enroll, stud_entrance
WHERE
stud_enroll.obrid = stud_entrance.obrid
AND stud_enroll.inst_code = stud_entrance.inst_code
AND stud_enroll.admis_area_code = stud_entrance.admis_area_code
AND stud_enroll.yr_num = 1999
AND stud_enroll.term_code = "AU"
AND stud_enroll.rank_code in("MS","DS","NM","ND","PR")
/*AND stud_enroll.campus_code = "FIRE"*/
GROUP BY
stud_enroll.yr_num,
stud_enroll.term_code,
stud_enroll.inst_code,
stud_enroll.obrid,
stud_entrance.state_abbrev
/*step2 sum to inst*/
SELECT
inst_code,
count(*) students,
sum(CASE WHEN state_abbrev = "OH" THEN 1 ELSE 0 END) Ohio,
sum(CASE WHEN state_abbrev = "ZZ" THEN 1 ELSE 0 END) over_seas,
```

```
sum(CASE WHEN state_abbrev not in("OH","ZZ") THEN 1 ELSE 0 END) other_state
INTO #step2
FROM #step1
GROUP BY inst_code
/*step3 calc %s*/
SELECT
inst_campus.campus_name,
students,
convert(float,Ohio)/convert(float,students) Ohio,
convert(float,other_state)/convert(float,students) other_state,
convert(float,over_seas)/convert(float,students) over_seas
FROM #step2, inst_campus
WHERE
#step2.inst_code = inst_campus.campus_code
/*step 4 sum to state*/
SELECT Count(*) students,
Sum(CASE WHEN state_abbrev="OH" THEN 1 ELSE 0 END) Ohio,
Sum(CASE WHEN state_abbrev="ZZ" THEN 1 ELSE 0 END) over_seas,
Sum(CASE WHEN state_abbrev Not In ("OH","ZZ") THEN 1 ELSE 0 END) Other_State
INTO #step4
FROM #step1
/*step 5 state report*/
SELECT students,
Convert(float,Ohio)/convert(float,students) Ohio,
Convert(float,Other_State)/convert(float,students) other_state,
Convert(float,over_seas)/convert(float,students) over_seas
FROM #step4
```

6.Time and Credits to Degree

time to degree

```
/*Step 0a create term off set table*/
CREATE TABLE #term_offset
(term_type char(1),
term char(2),
offset integer)
/*Step 0b populate the dev course table*/
INSERT #term_offset values ('Q',        'WI',1)
INSERT #term_offset values ('Q',        'SP',2)
INSERT #term_offset values ('Q',        'SM',3)
INSERT #term_offset values ('Q',        'AU',4)
INSERT #term_offset values ('S',        'WI',1)
INSERT #term_offset values ('S',        'SP',1)
INSERT #term_offset values ('S',        'SM',2)
INSERT #term_offset values ('S',        'AU',3)
/*Step 1 Outter joins*/
SELECT
degree_cert.*,
```

```
 "UND" AS UG_ADM,
 extend_bacc_pgrm.exp_time_compl AS EP_Time,
 extend_bacc_pgrm.cr_hrs_compl AS EP_SCH,
 extend_bacc_stud.exp_time_compl AS ES_Time,
 extend_bacc_stud.cr_hrs_compl AS ES_SCH,
 /*#aca_cal.term_type_code,*/
hei_sys..acad_cal.term_type_code,
 rev_first_enroll.first_enroll_term_code AS RF_first_term,
 rev_first_enroll.first_enroll_yr_num AS RF_first_year
into #step1
/*FROM degree_cert, #aca_cal, rev_first_enroll,extend_bacc_pgrm, extend_bacc_stud*/
/*where degree_cert.degree_cert_term_earned  *=  #aca_cal.term_code*/
/*AND  degree_cert.degree_cert_yr_earned  *=  #aca_cal.yr_num*/
/*AND  degree_cert.inst_code  *=  #aca_cal.inst_code*/
FROM degree_cert, hei_sys..acad_cal, rev_first_enroll,extend_bacc_pgrm, extend_bacc_stud
where degree_cert.degree_cert_term_earned  *=  hei_sys..acad_cal.term_code
AND  degree_cert.degree_cert_yr_earned  *= hei_sys..acad_cal.yr_num
AND  degree_cert.inst_code  *=  hei_sys..acad_cal.inst_code
and degree_cert.inst_code  *=  rev_first_enroll.inst_code
AND  degree_cert.obrid  *=  rev_first_enroll.obrid
and degree_cert.yr_num  *=  extend_bacc_pgrm.yr_num
AND  degree_cert.inst_code  *=  extend_bacc_pgrm.inst_code
AND  degree_cert.subject_code  *=  extend_bacc_pgrm.subject_code
and degree_cert.yr_num  *=  extend_bacc_stud.yr_num
 AND  degree_cert.subject_code  *=  extend_bacc_stud.subject_code
AND  degree_cert.obrid  *=  extend_bacc_stud.obrid
and degree_cert.yr_num =1999
AND  degree_cert.degree_cert_level_code In ("03","05")
/*and degree_cert.inst_code = "CNST"*/

/*Step 2 Join SE*/
SELECT #step1.*,
 stud_entrance.first_enroll_yr_num,
 stud_entrance.first_enroll_term_code,
case when RF_first_term is null then first_enroll_term_code else RF_first_term end AS First_Term,
case when RF_first_year is null then first_enroll_yr_num else RF_first_year end  AS First_Year
into #step2
FROM #step1, stud_entrance
where (#step1.UG_ADM =  stud_entrance.admis_area_code
AND #step1.inst_code =  stud_entrance.inst_code
AND #step1.obrid =  stud_entrance.obrid
AND #step1.inst_code <>"CNST" )
OR (#step1.UG_ADM =  stud_entrance.admis_area_code
AND #step1.inst_code =  stud_entrance.inst_code
AND #step1.obrid =  stud_entrance.obrid
AND #step1.inst_code = "CNST"
and  (stud_entrance.first_enroll_yr_num <>1998
or  stud_entrance.first_enroll_term_code<>"WI"))
```

```
/*Step 3 calc duration*/
SELECT #step2.subject_code,
#step2.degree_cert_cr_hrs AS Credits_in_DC,
#step2.degree_cert_yr_earned AS Year_Earned_from_DC,
#step2.degree_cert_term_earned AS Term_Earned_from_DC ,
#step2. EP_Time ,
#step2. EP_SCH ,
#step2. ES_Time ,
#step2. ES_SCH ,
#step2.first_enroll_yr_num AS  Year_from_SE ,
#step2.first_enroll_term_code AS  Term_from_SE ,
#step2. RF_first_year ,
#step2. RF_first_term ,
#step2. First_Year ,
#step2. First_Term ,
case when ES_Time is not null then ES_Time
       when EP_Time is not null then EP_Time
       when degree_cert_level_code ="03" then 2 else 4 end as Degree_time,

case when degree_cert_level_code ="03" and term_type_code ="S" then 60
    when degree_cert_level_code ="03" and term_type_code ="Q" then 90
    when ES_SCH is not null then ES_SCH
    when EP_SCH is not null then EP_SCH
    when term_type_code ="S" then 120 else 180 end as Degree_credits,

 #term_offset.offset AS  Start_Term ,
 #term_offset_1.offset AS  End_Term ,
#step2.term_type_code,

convert(float,case
when #step2.inst_code ="CNST" then
convert(float,(( convert(float,degree_cert_yr_earned) *5+ convert(float,#term_offset_1.offset) )
-( convert(float,First_Year) *5+ convert(float,#term_offset.offset) )+1)/5)
when term_type_code ="Q" then
convert(float,(( convert(float,degree_cert_yr_earned) *4+ convert(float,#term_offset_1.offset) )
-( convert(float,First_Year) *4+ convert(float,#term_offset.offset) )+1)/4)
else
convert(float,(( convert(float,degree_cert_yr_earned) *3+ convert(float,#term_offset_1.offset ))
-( First_Year *3+ #term_offset.offset )+1)/3) end) AS Duration,
 #step2.obrid,
#step2.campus_code AS Campus,
#step2.inst_code AS  Inst_Code ,
#step2.degree_cert_level_code AS  Level_from_DC
into #step3
FROM #step2 , #term_offset, #term_offset #term_offset_1
where #step2.term_type_code =  #term_offset.term_type
AND #step2. First_Term  =  #term_offset.term
and #step2.term_type_code =  #term_offset_1.term_type
AND #step2.degree_cert_term_earned =  #term_offset_1.term
```

```
  /*step3a sort and prepare SPSS input*/
select
Level_from_DC,
inst_campus.campus_type_code as Sector,
inst_campus_1.campus_name as Inst,
inst_campus.campus_name as Campus,
Duration
from #step3, inst_campus, inst_campus inst_campus_1
where Credits_in_DC >= Degree_credits
and #step3.Campus = inst_campus.campus_code
and #step3.Inst_Code = inst_campus_1.inst_code
and inst_campus_1.inst_code = inst_campus_1.campus_code
order by
Level_from_DC,
inst_campus.campus_type_code,
inst_campus_1.campus_name,
/*Step 0a create term off set table*/
CREATE TABLE #term_offset
(term_type char(1),
term char(2),
offset integer)
/*Step 0b populate the dev course table*/
INSERT #term_offset values ('Q',                    'WI',1)
INSERT #term_offset values ('Q',                    'SP',2)
INSERT #term_offset values ('Q',                    'SM',3)
INSERT #term_offset values ('Q',                    'AU',4)
INSERT #term_offset values ('S',                    'WI',1)
INSERT #term_offset values ('S',                    'SP',1)
INSERT #term_offset values ('S',                    'SM',2)
INSERT #term_offset values ('S',                    'AU',3)
/*Step 1 Outter joins*/
SELECT
degree_cert.*,
 "UND" AS UG_ADM,
 extend_bacc_pgrm.exp_time_compl AS EP_Time,
 extend_bacc_pgrm.cr_hrs_compl AS EP_SCH,
 extend_bacc_stud.exp_time_compl AS ES_Time,
 extend_bacc_stud.cr_hrs_compl AS ES_SCH,
 /*#aca_cal.term_type_code,*/
hei_sys..acad_cal.term_type_code,
 rev_first_enroll.first_enroll_term_code AS RF_first_term,
 rev_first_enroll.first_enroll_yr_num AS RF_first_year
into #step1
/*FROM degree_cert, #aca_cal, rev_first_enroll,extend_bacc_pgrm, extend_bacc_stud*/
/*where degree_cert.degree_cert_term_earned  *=  #aca_cal.term_code*/
/*AND  degree_cert.degree_cert_yr_earned  *=  #aca_cal.yr_num*/
/*AND  degree_cert.inst_code  *=  #aca_cal.inst_code*/
FROM degree_cert, hei_sys..acad_cal, rev_first_enroll,extend_bacc_pgrm, extend_bacc_stud
where degree_cert.degree_cert_term_earned  *=  hei_sys..acad_cal.term_code
```

```
AND  degree_cert.degree_cert_yr_earned  *= hei_sys..acad_cal.yr_num
AND  degree_cert.inst_code  *=  hei_sys..acad_cal.inst_code
and degree_cert.inst_code  *=  rev_first_enroll.inst_code
AND  degree_cert.obrid  *=  rev_first_enroll.obrid
and degree_cert.yr_num  *=  extend_bacc_pgrm.yr_num
AND  degree_cert.inst_code  *=  extend_bacc_pgrm.inst_code
AND  degree_cert.subject_code  *=  extend_bacc_pgrm.subject_code
and degree_cert.yr_num  *=  extend_bacc_stud.yr_num
 AND  degree_cert.subject_code  *=  extend_bacc_stud.subject_code
AND  degree_cert.obrid  *=  extend_bacc_stud.obrid
and degree_cert.yr_num =1999
AND  degree_cert.degree_cert_level_code In ("03","05")
/*and degree_cert.inst_code = "CNST"*/

/*Step 2 Join SE*/
SELECT #step1.*,
 stud_entrance.first_enroll_yr_num,
 stud_entrance.first_enroll_term_code,
case when RF_first_term is null then first_enroll_term_code else RF_first_term end AS First_Term,
case when RF_first_year is null then first_enroll_yr_num else RF_first_year end  AS First_Year
into #step2
FROM #step1, stud_entrance
where (#step1.UG_ADM =  stud_entrance.admis_area_code
AND #step1.inst_code =  stud_entrance.inst_code
AND #step1.obrid =  stud_entrance.obrid
AND #step1.inst_code <>"CNST" )
OR (#step1.UG_ADM =  stud_entrance.admis_area_code
AND #step1.inst_code =  stud_entrance.inst_code
AND #step1.obrid =  stud_entrance.obrid
AND #step1.inst_code = "CNST"
and  (stud_entrance.first_enroll_yr_num <>1998
or  stud_entrance.first_enroll_term_code<>"WI"))

/*Step 3 calc duration*/
SELECT #step2.subject_code,
#step2.degree_cert_cr_hrs AS Credits_in_DC,
#step2.degree_cert_yr_earned AS Year_Earned_from_DC,
#step2.degree_cert_term_earned AS Term_Earned_from_DC ,
#step2. EP_Time ,
#step2. EP_SCH ,
#step2. ES_Time ,
#step2. ES_SCH ,
#step2.first_enroll_yr_num AS  Year_from_SE ,
#step2.first_enroll_term_code AS  Term_from_SE ,
#step2. RF_first_year ,
#step2. RF_first_term ,
#step2. First_Year ,
#step2. First_Term ,
case when ES_Time is not null then ES_Time
```

```
        when EP_Time is not null then EP_Time
        when degree_cert_level_code ="03" then 2 else 4 end as Degree_time,

case when degree_cert_level_code ="03" and term_type_code ="S" then 60
     when degree_cert_level_code ="03" and term_type_code ="Q" then 90
     when ES_SCH is not null then ES_SCH
     when EP_SCH is not null then EP_SCH
     when term_type_code ="S" then 120 else 180 end as Degree_credits,

 #term_offset.offset AS  Start_Term ,
 #term_offset_1.offset AS  End_Term ,
#step2.term_type_code,

convert(float,case
when #step2.inst_code ="CNST" then
convert(float,(( convert(float,degree_cert_yr_earned) *5+ convert(float,#term_offset_1.offset) )
-( convert(float,First_Year) *5+ convert(float,#term_offset.offset) )+1)/5)
when term_type_code ="Q" then
convert(float,(( convert(float,degree_cert_yr_earned) *4+ convert(float,#term_offset_1.offset) )
-( convert(float,First_Year) *4+ convert(float,#term_offset.offset) )+1)/4)
else
convert(float,(( convert(float,degree_cert_yr_earned) *3+ convert(float,#term_offset_1.offset ))
-( First_Year *3+ #term_offset.offset )+1)/3) end) AS Duration,
 #step2.obrid,
#step2.campus_code AS Campus,
#step2.inst_code AS  Inst_Code ,
#step2.degree_cert_level_code AS  Level_from_DC
into #step3
FROM #step2 , #term_offset, #term_offset #term_offset_1
where #step2.term_type_code =  #term_offset.term_type
AND #step2. First_Term  =  #term_offset.term
and #step2.term_type_code =  #term_offset_1.term_type
AND #step2.degree_cert_term_earned =  #term_offset_1.term
  /*step3a sort and prepare SPSS input*/
select
Level_from_DC,
inst_campus.campus_type_code as Sector,
inst_campus_1.campus_name as Inst,
inst_campus.campus_name as Campus,
Duration
from #step3, inst_campus, inst_campus inst_campus_1
where Credits_in_DC >= Degree_credits
and #step3.Campus = inst_campus.campus_code
and #step3.Inst_Code = inst_campus_1.inst_code
and inst_campus_1.inst_code = inst_campus_1.campus_code
order by
Level_from_DC,
inst_campus.campus_type_code,
inst_campus_1.campus_name,
```

```
inst_campus.campus_name,
Duration
/*Step 0a create term off set table*/
CREATE TABLE #term_offset
(term_type char(1),
term char(2),
offset integer)
/*Step 0b populate the dev course table*/
INSERT          'WI',1)
#term_offset
values ('Q',
INSERT          'SP',2)
#term_offset
values ('Q',
INSERT          'SM',3)
#term_offset
values ('Q',
INSERT          'AU',4)
#term_offset
values ('Q',
INSERT          'WI',1)
#term_offset
values ('S',
INSERT          'SP',1)
#term_offset
values ('S',
INSERT          'SM',2)
#term_offset
values ('S',
INSERT          'AU',3)
#term_offset
values ('S',
/*Step 1 Outter joins*/
SELECT
degree_cert.*,
 "UND" AS UG_ADM,
 extend_bacc_pgrm.exp_time_compl AS EP_Time,
 extend_bacc_pgrm.cr_hrs_compl AS EP_SCH,
 extend_bacc_stud.exp_time_compl AS ES_Time,
 extend_bacc_stud.cr_hrs_compl AS ES_SCH,
 /*#aca_cal.term_type_code,*/
hei_sys..acad_cal.term_type_code,
 rev_first_enroll.first_enroll_term_code AS RF_first_term,
 rev_first_enroll.first_enroll_yr_num AS RF_first_year
into #step1
/*FROM degree_cert, #aca_cal, rev_first_enroll,extend_bacc_pgrm, extend_bacc_stud*/
/*where degree_cert.degree_cert_term_earned  *=  #aca_cal.term_code*/
/*AND  degree_cert.degree_cert_yr_earned  *=  #aca_cal.yr_num*/
/*AND  degree_cert.inst_code  *=  #aca_cal.inst_code*/
FROM degree_cert, hei_sys..acad_cal, rev_first_enroll,extend_bacc_pgrm, extend_bacc_stud
where degree_cert.degree_cert_term_earned  *=  hei_sys..acad_cal.term_code
AND  degree_cert.degree_cert_yr_earned  *= hei_sys..acad_cal.yr_num
```

```
AND  degree_cert.inst_code  *=  hei_sys..acad_cal.inst_code
and degree_cert.inst_code  *=  rev_first_enroll.inst_code
AND  degree_cert.obrid  *=  rev_first_enroll.obrid
and degree_cert.yr_num  *=  extend_bacc_pgrm.yr_num
AND  degree_cert.inst_code  *=  extend_bacc_pgrm.inst_code
AND  degree_cert.subject_code  *=  extend_bacc_pgrm.subject_code
and degree_cert.yr_num  *=  extend_bacc_stud.yr_num
 AND  degree_cert.subject_code  *=  extend_bacc_stud.subject_code
AND  degree_cert.obrid  *=  extend_bacc_stud.obrid
and degree_cert.yr_num =1999
AND  degree_cert.degree_cert_level_code In ("03","05")
/*and degree_cert.inst_code = "CNST"*/

/*Step 2 Join SE*/
SELECT #step1.*,
 stud_entrance.first_enroll_yr_num,
 stud_entrance.first_enroll_term_code,
case when RF_first_term is null then first_enroll_term_code else RF_first_term end AS First_Term,
case when RF_first_year is null then first_enroll_yr_num else RF_first_year end  AS First_Year
into #step2
FROM #step1, stud_entrance
where (#step1.UG_ADM =  stud_entrance.admis_area_code
AND #step1.inst_code =  stud_entrance.inst_code
AND #step1.obrid =  stud_entrance.obrid
AND #step1.inst_code <>"CNST" )
OR (#step1.UG_ADM =  stud_entrance.admis_area_code
AND #step1.inst_code =  stud_entrance.inst_code
AND #step1.obrid =  stud_entrance.obrid
AND #step1.inst_code = "CNST"
and  (stud_entrance.first_enroll_yr_num <>1998
or  stud_entrance.first_enroll_term_code<>"WI"))

/*Step 3 calc duration*/
SELECT #step2.subject_code,
#step2.degree_cert_cr_hrs AS Credits_in_DC,
#step2.degree_cert_yr_earned AS Year_Earned_from_DC,
#step2.degree_cert_term_earned AS Term_Earned_from_DC ,
#step2. EP_Time ,
#step2. EP_SCH ,
#step2. ES_Time ,
#step2. ES_SCH ,
#step2.first_enroll_yr_num AS  Year_from_SE ,
#step2.first_enroll_term_code AS  Term_from_SE ,
#step2. RF_first_year ,
#step2. RF_first_term ,
#step2. First_Year ,
#step2. First_Term ,
case when ES_Time is not null then ES_Time
       when EP_Time is not null then EP_Time
```

```
        when degree_cert_level_code ="03" then 2 else 4 end as Degree_time,

case when degree_cert_level_code ="03" and term_type_code ="S" then 60
     when degree_cert_level_code ="03" and term_type_code ="Q" then 90
     when ES_SCH is not null then ES_SCH
     when EP_SCH is not null then EP_SCH
     when term_type_code ="S" then 120 else 180 end as Degree_credits,

 #term_offset.offset AS  Start_Term ,
 #term_offset_1.offset AS  End_Term ,
#step2.term_type_code,

convert(float,case
when #step2.inst_code ="CNST" then
convert(float,(( convert(float,degree_cert_yr_earned) *5+ convert(float,#term_offset_1.offset) )
-( convert(float,First_Year) *5+ convert(float,#term_offset.offset) )+1)/5)
when term_type_code ="Q" then
convert(float,(( convert(float,degree_cert_yr_earned) *4+ convert(float,#term_offset_1.offset) )
-( convert(float,First_Year) *4+ convert(float,#term_offset.offset) )+1)/4)
else
convert(float,(( convert(float,degree_cert_yr_earned) *3+ convert(float,#term_offset_1.offset ))
-( First_Year *3+ #term_offset.offset )+1)/3) end) AS Duration,
 #step2.obrid,
#step2.campus_code AS Campus,
#step2.inst_code AS  Inst_Code ,
#step2.degree_cert_level_code AS  Level_from_DC
into #step3
FROM #step2 , #term_offset, #term_offset #term_offset_1
where #step2.term_type_code =  #term_offset.term_type
AND #step2. First_Term  =  #term_offset.term
and #step2.term_type_code =  #term_offset_1.term_type
AND #step2.degree_cert_term_earned =  #term_offset_1.term
  /*step3a sort and prepare SPSS input*/
select
Level_from_DC,
inst_campus.campus_type_code as Sector,
inst_campus_1.campus_name as Inst,
inst_campus.campus_name as Campus,
Duration
from #step3, inst_campus, inst_campus inst_campus_1
where Credits_in_DC >= Degree_credits
and #step3.Campus = inst_campus.campus_code
and #step3.Inst_Code = inst_campus_1.inst_code
and inst_campus_1.inst_code = inst_campus_1.campus_code
order by
Level_from_DC,
inst_campus.campus_type_code,
inst_campus_1.campus_name,
inst_campus.campus_name,
```

```
Duration
inst_campus.campus_name,
Duration
```

Credits to Degree

```
/*step1 select degree with highest SCH*/
SELECT
campus_code,
degree_cert_level_code,
degree_cert_term_earned,
degree_cert_yr_earned,
inst_code,
obrid,
yr_num,
max(subject_code) subject
INTO #step1
FROM degree_cert
WHERE
yr_num = 1999
and degree_cert_level_code In ("03","05","07")
/*and inst_code = "LCCC"*/
GROUP BY
campus_code,
degree_cert_level_code,
degree_cert_term_earned,
degree_cert_yr_earned,
inst_code,
obrid
/*step2 select prev degrees*/
SELECT
#step1.*,
degree_cert.yr_num prev_yr_num,
degree_cert.degree_cert_yr_earned prev_yr,
degree_cert.degree_cert_term_earned prev_term
INTO #step2
FROM #step1, degree_cert
WHERE #step1.obrid =degree_cert.obrid
AND #step1.degree_cert_level_code = degree_cert.degree_cert_level_code
AND #step1.campus_code = degree_cert.campus_code
AND (#step1.yr_num > degree_cert.yr_num
        OR #step1.yr_num = degree_cert.yr_num
        AND #step1.degree_cert_term_earned > degree_cert.degree_cert_term_earned)
/*step3 Join mult degrees*/
SELECT
#step1.*,
prev_yr,
prev_term
INTO #step3
```

```
FROM #step1, #step2
WHERE #step1.obrid *= #step2.obrid
AND #step1.degree_cert_level_code *= #step2.degree_cert_level_code
AND #step1.campus_code *= #step2.campus_code
/*step4 elim 2nd degrees and join in extended programs*/
SELECT
#step3.*,
extend_bacc_pgrm.cr_hrs_compl EP_sch,
extend_bacc_pgrm.exp_time_compl EP_time,
extend_bacc_stud.cr_hrs_compl ES_sch,
extend_bacc_stud.exp_time_compl ES_time
INTO #step4
FROM #step3, extend_bacc_pgrm, extend_bacc_stud
WHERE #step3.subject *= extend_bacc_pgrm.subject_code
AND #step3.inst_code *= extend_bacc_pgrm.inst_code
AND #step3.yr_num *= extend_bacc_pgrm.yr_num
AND #step3.obrid *= extend_bacc_stud.obrid
AND #step3.subject *= extend_bacc_stud.subject_code
AND #step3.yr_num *= extend_bacc_stud.yr_num
AND #step3.inst_code *= extend_bacc_stud.inst_code
AND prev_yr is null
AND prev_yr is null
/*step5 join back to DC to pick up DC credits*/
SELECT
#step4.*,
degree_cert.degree_cert_cr_hrs DC_sch
INTO #step5
FROM #step4, degree_cert
WHERE
#step4.campus_code = degree_cert.campus_code
AND #step4.degree_cert_level_code = degree_cert.degree_cert_level_code
AND #step4.degree_cert_term_earned = degree_cert.degree_cert_term_earned
AND #step4.degree_cert_yr_earned = degree_cert.degree_cert_yr_earned
AND #step4.inst_code = degree_cert.inst_code
AND #step4.obrid = degree_cert.obrid
AND #step4.subject = degree_cert.subject_code
/*Step 6 create calendar table*/
CREATE TABLE #calendar
(inst char(4),
year integer,
term char(2),
cal_type char(1))
/*Step 0b populate the table*/
INSERT #calendar values (	'AKRN',1998,		'AU','S')
INSERT #calendar values (	'AKRN',1999,		'SP','S')
INSERT #calendar values (	'AKRN',1999,		'SM','S')
INSERT #calendar values (	'AKRN',1999,		'AU','S')
INSERT #calendar values (	'AKRN',2000,		'SP','S')
INSERT #calendar values (	'BGSU',1998,		'SM','S')
```

```
INSERT #calendar values (	'BGSU',1998,	'AU','S')
INSERT #calendar values (	'BGSU',1999,	'SP','S')
INSERT #calendar values (	'BGSU',1999,	'SM','S')
INSERT #calendar values (	'BGSU',1999,	'AU','S')
INSERT #calendar values (	'BGSU',2000,	'SP','S')
INSERT #calendar values (	'BLTC',1998,	'SM','Q')
INSERT #calendar values (	'BLTC',1998,	'AU','Q')
INSERT #calendar values (	'BLTC',1999,	'WI','Q')
INSERT #calendar values (	'BLTC',1999,	'SP','Q')
INSERT #calendar values (	'BLTC',1999,	'SM','Q')
INSERT #calendar values (	'BLTC',1999,	'AU','Q')
INSERT #calendar values (	'BLTC',2000,	'WI','Q')
INSERT #calendar values (	'BLTC',2000,	'SP','Q')
INSERT #calendar values (	'CINC',1998,	'SM','Q')
INSERT #calendar values (	'CINC',1998,	'AU','Q')
INSERT #calendar values (	'CINC',1999,	'WI','Q')
INSERT #calendar values (	'CINC',1999,	'SP','Q')
INSERT #calendar values (	'CINC',1999,	'SM','Q')
INSERT #calendar values (	'CINC',1999,	'AU','Q')
INSERT #calendar values (	'CINC',2000,	'WI','Q')
INSERT #calendar values (	'CINC',2000,	'SP','Q')
INSERT #calendar values (	'CLEV',1998,	'SM','Q')
INSERT #calendar values (	'CLEV',1998,	'AU','S')
INSERT #calendar values (	'CLEV',1999,	'SP','S')
INSERT #calendar values (	'CLEV',1999,	'SM','S')
INSERT #calendar values (	'CLEV',1999,	'AU','S')
INSERT #calendar values (	'CLEV',2000,	'SP','S')
INSERT #calendar values (	'CLRK',1998,	'SM','Q')
INSERT #calendar values (	'CLRK',1998,	'AU','Q')
INSERT #calendar values (	'CLRK',1999,	'WI','Q')
INSERT #calendar values (	'CLRK',1999,	'SP','Q')
INSERT #calendar values (	'CLRK',1999,	'SM','Q')
INSERT #calendar values (	'CLRK',1999,	'AU','Q')
INSERT #calendar values (	'CLRK',2000,	'WI','Q')
INSERT #calendar values (	'CLRK',2000,	'SP','Q')
INSERT #calendar values (	'CNST',1998,	'SM','Q')
INSERT #calendar values (	'CNST',1998,	'AU','Q')
INSERT #calendar values (	'CNST',1999,	'WI','Q')
INSERT #calendar values (	'CNST',1999,	'SP','Q')
INSERT #calendar values (	'CNST',1999,	'SM','Q')
INSERT #calendar values (	'CNST',1999,	'AU','Q')
INSERT #calendar values (	'CNST',2000,	'WI','Q')
INSERT #calendar values (	'CNST',2000,	'SP','Q')
INSERT #calendar values (	'CNTL',1998,	'SM','Q')
INSERT #calendar values (	'CNTL',1998,	'AU','Q')
INSERT #calendar values (	'CNTL',1999,	'WI','Q')
INSERT #calendar values (	'CNTL',1999,	'SP','Q')
INSERT #calendar values (	'CNTL',1999,	'SM','Q')
INSERT #calendar values (	'CNTL',1999,	'AU','Q')
```

```
INSERT #calendar values (	'CNTL',2000,	'WI','Q')
INSERT #calendar values (	'CNTL',2000,	'SP','Q')
INSERT #calendar values (	'COTC',1998,	'SM','Q')
INSERT #calendar values (	'COTC',1998,	'AU','Q')
INSERT #calendar values (	'COTC',1999,	'WI','Q')
INSERT #calendar values (	'COTC',1999,	'SP','Q')
INSERT #calendar values (	'COTC',1999,	'SM','Q')
INSERT #calendar values (	'COTC',1999,	'AU','Q')
INSERT #calendar values (	'COTC',2000,	'WI','Q')
INSERT #calendar values (	'COTC',2000,	'SP','Q')
INSERT #calendar values (	'CSCC',1998,	'SM','Q')
INSERT #calendar values (	'CSCC',1998,	'AU','Q')
INSERT #calendar values (	'CSCC',1999,	'WI','Q')
INSERT #calendar values (	'CSCC',1999,	'SP','Q')
INSERT #calendar values (	'CSCC',1999,	'SM','Q')
INSERT #calendar values (	'CSCC',1999,	'AU','Q')
INSERT #calendar values (	'CSCC',2000,	'WI','Q')
INSERT #calendar values (	'CSCC',2000,	'SP','Q')
INSERT #calendar values (	'CYCC',1998,	'SM','Q')
INSERT #calendar values (	'CYCC',1998,	'AU','S')
INSERT #calendar values (	'CYCC',1999,	'SP','S')
INSERT #calendar values (	'CYCC',1999,	'SM','S')
INSERT #calendar values (	'CYCC',1999,	'AU','S')
INSERT #calendar values (	'CYCC',2000,	'SP','S')
INSERT #calendar values (	'EDSN',1998,	'SM','S')
INSERT #calendar values (	'EDSN',1998,	'AU','S')
INSERT #calendar values (	'EDSN',1999,	'SP','S')
INSERT #calendar values (	'EDSN',1999,	'SM','S')
INSERT #calendar values (	'EDSN',1999,	'AU','S')
INSERT #calendar values (	'EDSN',2000,	'SP','S')
INSERT #calendar values (	'HOCK',1998,	'SM','Q')
INSERT #calendar values (	'HOCK',1998,	'AU','Q')
INSERT #calendar values (	'HOCK',1999,	'WI','Q')
INSERT #calendar values (	'HOCK',1999,	'SP','Q')
INSERT #calendar values (	'HOCK',1999,	'SM','Q')
INSERT #calendar values (	'HOCK',1999,	'AU','Q')
INSERT #calendar values (	'HOCK',2000,	'WI','Q')
INSERT #calendar values (	'HOCK',2000,	'SP','Q')
INSERT #calendar values (	'JEFF',1998,	'SM','S')
INSERT #calendar values (	'JEFF',1998,	'AU','S')
INSERT #calendar values (	'JEFF',1999,	'SP','S')
INSERT #calendar values (	'JEFF',1999,	'SM','S')
INSERT #calendar values (	'JEFF',1999,	'AU','S')
INSERT #calendar values (	'JEFF',2000,	'SP','S')
INSERT #calendar values (	'KENT',1998,	'SM','S')
INSERT #calendar values (	'KENT',1998,	'AU','S')
INSERT #calendar values (	'KENT',1999,	'SP','S')
INSERT #calendar values (	'KENT',1999,	'SM','S')
INSERT #calendar values (	'KENT',1999,	'AU','S')
```

```
INSERT #calendar values (	'KENT',2000,	'SP','S')
INSERT #calendar values (	'KETT',1999,	'SM','S')
INSERT #calendar values (	'KETT',1999,	'AU','S')
INSERT #calendar values (	'KETT',2000,	'SP','S')
INSERT #calendar values (	'LCCC',1998,	'SM','Q')
INSERT #calendar values (	'LCCC',1998,	'AU','S')
INSERT #calendar values (	'LCCC',1999,	'SP','S')
INSERT #calendar values (	'LCCC',1999,	'SM','S')
INSERT #calendar values (	'LCCC',1999,	'AU','S')
INSERT #calendar values (	'LCCC',2000,	'SP','S')
INSERT #calendar values (	'LKCC',1998,	'SM','Q')
INSERT #calendar values (	'LKCC',1998,	'AU','Q')
INSERT #calendar values (	'LKCC',1999,	'WI','Q')
INSERT #calendar values (	'LKCC',1999,	'SP','Q')
INSERT #calendar values (	'LKCC',1999,	'SM','Q')
INSERT #calendar values (	'LKCC',1999,	'AU','Q')
INSERT #calendar values (	'LKCC',2000,	'WI','Q')
INSERT #calendar values (	'LKCC',2000,	'SP','Q')
INSERT #calendar values (	'LMTC',1998,	'SM','Q')
INSERT #calendar values (	'LMTC',1998,	'AU','Q')
INSERT #calendar values (	'LMTC',1999,	'WI','Q')
INSERT #calendar values (	'LMTC',1999,	'SP','Q')
INSERT #calendar values (	'LMTC',1999,	'SM','Q')
INSERT #calendar values (	'LMTC',1999,	'AU','Q')
INSERT #calendar values (	'LMTC',2000,	'WI','Q')
INSERT #calendar values (	'LMTC',2000,	'SP','Q')
INSERT #calendar values (	'MATC',1998,	'SM','Q')
INSERT #calendar values (	'MATC',1998,	'AU','Q')
INSERT #calendar values (	'MATC',1999,	'WI','Q')
INSERT #calendar values (	'MATC',1999,	'SP','Q')
INSERT #calendar values (	'MATC',1999,	'SM','Q')
INSERT #calendar values (	'MATC',1999,	'AU','Q')
INSERT #calendar values (	'MATC',2000,	'WI','Q')
INSERT #calendar values (	'MATC',2000,	'SP','Q')
INSERT #calendar values (	'MCOT',1998,	'SM','S')
INSERT #calendar values (	'MCOT',1998,	'AU','S')
INSERT #calendar values (	'MCOT',1999,	'SP','S')
INSERT #calendar values (	'MCOT',1999,	'SM','S')
INSERT #calendar values (	'MCOT',1999,	'AU','S')
INSERT #calendar values (	'MCOT',2000,	'SP','S')
INSERT #calendar values (	'MIAM',1998,	'SM','S')
INSERT #calendar values (	'MIAM',1998,	'AU','S')
INSERT #calendar values (	'MIAM',1999,	'SP','S')
INSERT #calendar values (	'MIAM',1999,	'SM','S')
INSERT #calendar values (	'MIAM',1999,	'AU','S')
INSERT #calendar values (	'MIAM',2000,	'SP','S')
INSERT #calendar values (	'MRTC',1998,	'SM','Q')
INSERT #calendar values (	'MRTC',1998,	'AU','Q')
INSERT #calendar values (	'MRTC',1999,	'WI','Q')
```

```
INSERT #calendar values (	'MRTC',1999,	'SP','Q')
INSERT #calendar values (	'MRTC',1999,	'SM','Q')
INSERT #calendar values (	'MRTC',1999,	'AU','Q')
INSERT #calendar values (	'MRTC',2000,	'WI','Q')
INSERT #calendar values (	'MRTC',2000,	'SP','Q')
INSERT #calendar values (	'NCTC',1998,	'SM','Q')
INSERT #calendar values (	'NCTC',1998,	'AU','Q')
INSERT #calendar values (	'NCTC',1999,	'WI','Q')
INSERT #calendar values (	'NCTC',1999,	'SP','Q')
INSERT #calendar values (	'NCTC',1999,	'SM','Q')
INSERT #calendar values (	'NCTC',1999,	'AU','Q')
INSERT #calendar values (	'NCTC',2000,	'WI','Q')
INSERT #calendar values (	'NCTC',2000,	'SP','Q')
INSERT #calendar values (	'NECM',1998,	'SM','Q')
INSERT #calendar values (	'NECM',1998,	'AU','Q')
INSERT #calendar values (	'NECM',1999,	'WI','Q')
INSERT #calendar values (	'NECM',1999,	'SP','Q')
INSERT #calendar values (	'NECM',1999,	'AU','Q')
INSERT #calendar values (	'NECM',2000,	'WI','Q')
INSERT #calendar values (	'NECM',2000,	'SP','Q')
INSERT #calendar values (	'NSCC',1998,	'SM','S')
INSERT #calendar values (	'NSCC',1998,	'AU','S')
INSERT #calendar values (	'NSCC',1999,	'SP','S')
INSERT #calendar values (	'NSCC',1999,	'SM','S')
INSERT #calendar values (	'NSCC',1999,	'AU','S')
INSERT #calendar values (	'NSCC',2000,	'SP','S')
INSERT #calendar values (	'OHSU',1998,	'SM','Q')
INSERT #calendar values (	'OHSU',1998,	'AU','Q')
INSERT #calendar values (	'OHSU',1999,	'WI','Q')
INSERT #calendar values (	'OHSU',1999,	'SP','Q')
INSERT #calendar values (	'OHSU',1999,	'SM','Q')
INSERT #calendar values (	'OHSU',1999,	'AU','Q')
INSERT #calendar values (	'OHSU',2000,	'WI','Q')
INSERT #calendar values (	'OHSU',2000,	'SP','Q')
INSERT #calendar values (	'OHUN',1998,	'SM','Q')
INSERT #calendar values (	'OHUN',1998,	'AU','Q')
INSERT #calendar values (	'OHUN',1999,	'WI','Q')
INSERT #calendar values (	'OHUN',1999,	'SP','Q')
INSERT #calendar values (	'OHUN',1999,	'SM','Q')
INSERT #calendar values (	'OHUN',1999,	'AU','Q')
INSERT #calendar values (	'OHUN',2000,	'WI','Q')
INSERT #calendar values (	'OHUN',2000,	'SP','Q')
INSERT #calendar values (	'OSCC',1998,	'SM','S')
INSERT #calendar values (	'OSCC',1998,	'AU','S')
INSERT #calendar values (	'OSCC',1999,	'SP','S')
INSERT #calendar values (	'OSCC',1999,	'SM','S')
INSERT #calendar values (	'OSCC',1999,	'AU','S')
INSERT #calendar values (	'OSCC',2000,	'SP','S')
INSERT #calendar values (	'RGCC',1998,	'SM','Q')
```

```
INSERT #calendar values (	'RGCC',1998,	'AU','Q')
INSERT #calendar values (	'RGCC',1999,	'WI','Q')
INSERT #calendar values (	'RGCC',1999,	'SP','Q')
INSERT #calendar values (	'RGCC',1999,	'SM','Q')
INSERT #calendar values (	'RGCC',1999,	'AU','Q')
INSERT #calendar values (	'RGCC',2000,	'WI','Q')
INSERT #calendar values (	'RGCC',2000,	'SP','Q')
INSERT #calendar values (	'SHAW',1998,	'SM','Q')
INSERT #calendar values (	'SHAW',1998,	'AU','Q')
INSERT #calendar values (	'SHAW',1999,	'WI','Q')
INSERT #calendar values (	'SHAW',1999,	'SP','Q')
INSERT #calendar values (	'SHAW',1999,	'SM','Q')
INSERT #calendar values (	'SHAW',1999,	'AU','Q')
INSERT #calendar values (	'SHAW',2000,	'WI','Q')
INSERT #calendar values (	'SHAW',2000,	'SP','Q')
INSERT #calendar values (	'SNCL',1998,	'SM','Q')
INSERT #calendar values (	'SNCL',1998,	'AU','Q')
INSERT #calendar values (	'SNCL',1999,	'WI','Q')
INSERT #calendar values (	'SNCL',1999,	'SP','Q')
INSERT #calendar values (	'SNCL',1999,	'SM','Q')
INSERT #calendar values (	'SNCL',1999,	'AU','Q')
INSERT #calendar values (	'SNCL',2000,	'WI','Q')
INSERT #calendar values (	'SNCL',2000,	'SP','Q')
INSERT #calendar values (	'SSCC',1998,	'SM','Q')
INSERT #calendar values (	'SSCC',1998,	'AU','Q')
INSERT #calendar values (	'SSCC',1999,	'WI','Q')
INSERT #calendar values (	'SSCC',1999,	'SP','Q')
INSERT #calendar values (	'SSCC',1999,	'SM','Q')
INSERT #calendar values (	'SSCC',1999,	'AU','Q')
INSERT #calendar values (	'SSCC',2000,	'WI','Q')
INSERT #calendar values (	'SSCC',2000,	'SP','Q')
INSERT #calendar values (	'SSCT',1998,	'SM','S')
INSERT #calendar values (	'SSCT',1998,	'AU','S')
INSERT #calendar values (	'SSCT',1999,	'SP','S')
INSERT #calendar values (	'SSCT',1999,	'SM','S')
INSERT #calendar values (	'SSCT',1999,	'AU','S')
INSERT #calendar values (	'SSCT',2000,	'SP','S')
INSERT #calendar values (	'TLDO',1998,	'SM','S')
INSERT #calendar values (	'TLDO',1998,	'AU','S')
INSERT #calendar values (	'TLDO',1999,	'SP','S')
INSERT #calendar values (	'TLDO',1999,	'SM','S')
INSERT #calendar values (	'TLDO',1999,	'AU','S')
INSERT #calendar values (	'TLDO',2000,	'SP','S')
INSERT #calendar values (	'TRRA',1998,	'SM','Q')
INSERT #calendar values (	'TRRA',1998,	'AU','Q')
INSERT #calendar values (	'TRRA',1999,	'WI','Q')
INSERT #calendar values (	'TRRA',1999,	'SP','Q')
INSERT #calendar values (	'TRRA',1999,	'SM','Q')
INSERT #calendar values (	'TRRA',1999,	'AU','Q')
```

```
INSERT #calendar values (	'TRRA',2000,		'WI','Q')
INSERT #calendar values (	'TRRA',2000,		'SP','Q')
INSERT #calendar values (	'WSCC',1998,		'SM','Q')
INSERT #calendar values (	'WSCC',1998,		'AU','Q')
INSERT #calendar values (	'WSCC',1999,		'WI','Q')
INSERT #calendar values (	'WSCC',1999,		'SP','Q')
INSERT #calendar values (	'WSCC',1999,		'SM','Q')
INSERT #calendar values (	'WSCC',1999,		'AU','Q')
INSERT #calendar values (	'WSCC',2000,		'WI','Q')
INSERT #calendar values (	'WSCC',2000,		'SP','Q')
INSERT #calendar values (	'WSUN',1998,		'SM','Q')
INSERT #calendar values (	'WSUN',1998,		'AU','Q')
INSERT #calendar values (	'WSUN',1999,		'WI','Q')
INSERT #calendar values (	'WSUN',1999,		'SP','Q')
INSERT #calendar values (	'WSUN',1999,		'SM','Q')
INSERT #calendar values (	'WSUN',1999,		'AU','Q')
INSERT #calendar values (	'WSUN',2000,		'WI','Q')
INSERT #calendar values (	'WSUN',2000,		'SP','Q')
INSERT #calendar values (	'YNGS',1998,		'SM','Q')
INSERT #calendar values (	'YNGS',1998,		'AU','Q')
INSERT #calendar values (	'YNGS',1999,		'WI','Q')
INSERT #calendar values (	'YNGS',1999,		'SP','Q')
INSERT #calendar values (	'YNGS',1999,		'SM','Q')
INSERT #calendar values (	'YNGS',1999,		'AU','Q')
INSERT #calendar values (	'YNGS',2000,		'WI','Q')
INSERT #calendar values (	'YNGS',2000,		'SP','Q')
/*step7 calc credits to degree*/
SELECT
EP_sch,
ES_sch,
DC_sch,
obrid,
degree_cert_level_code,
inst_code,
campus_code,
degree_cert_term_earned,
degree_cert_yr_earned,
cal_type,
CASE WHEN EP_sch is null and ES_sch is null
      and degree_cert_level_code = "03" THEN 60
    WHEN EP_sch is null and ES_sch is null
      and degree_cert_level_code = "05" THEN 120
    WHEN ES_sch is not null THEN ES_sch
    WHEN EP_sch is not null THEN EP_sch
  ELSE 0 END Degree_credits,
CASE WHEN inst_code in ("LCCC","CYCC") and degree_cert_yr_earned = 1998 and degree_cert_term_earned =
"SM" then DC_sch
WHEN cal_type = "Q" THEN DC_sch * 2/3 ELSE DC_sch END Semester_sch
INTO #step7
```

```
FROM #step5, #calendar
WHERE inst_code = inst
AND degree_cert_term_earned = term
AND degree_cert_yr_earned = year
/*step7.5 sum to degree to elim dups*/
SELECT
EP_sch,
ES_sch,
DC_sch,
obrid,
degree_cert_level_code,
inst_code,
campus_code,
degree_cert_term_earned,
degree_cert_yr_earned,
cal_type,
Degree_credits,
Semester_sch
INTO #step7p5
FROM #step7
group by
EP_sch,
ES_sch,
DC_sch,
obrid,
degree_cert_level_code,
inst_code,
campus_code,
degree_cert_term_earned,
degree_cert_yr_earned,
cal_type,
Degree_credits,
Semester_sch
/*step8 sum to campus*/
SELECT
campus_code,
degree_cert_level_code,
count(*) non_transfer_grads,
avg(Semester_sch) Avg_semester_sch
INTO #step8
FROM #step7p5
WHERE degree_cert_level_code > "05" or Semester_sch > Degree_credits
GROUP BY
campus_code,
degree_cert_level_code
select * from #step8
```

7. Employment Outcomes of Graduates

```
/* Step 1 Join DC OBES SE */
SELECT   degree_cert.obrid,
  Degree_cert.campus_code,
  Degree_cert.yr_num,
  Degree_cert.degree_cert_level_code,
  Degree_cert.subject_code,
  Degree_cert.degree_cert_yr_earned,
  Degree_cert.degree_cert_term_earned,
  Degree_cert.inst_code,
  Degree_cert.admis_area_code,
  empl_qtr_wage.employer_id,
  empl_qtr_wage.weeks_of_work,
  empl_qtr_wage.wage_qtr_num,
  empl_qtr_wage.wage_yr_num,
  Isnull(empl_qtr_wage.wage_amt,0) AS wage_amt,
Degree_cert_yr_earned*4+
(CASE
    WHEN degree_cert_term_earned="WI" THEN 0
    WHEN degree_cert_term_earned="SP" THEN 1
    WHEN degree_cert_term_earned="SM" THEN 2
    ELSE 3
END) AS DC_term,
  stud_entrance.state_abbrev
INTO #step1
FROM  degree_cert,
           empl_qtr_wage,
stud_entrance
WHERE degree_cert.obrid *=   empl_qtr_wage.obrid
AND  degree_cert.obrid *=   stud_entrance.obrid
AND   degree_cert.inst_code *=   stud_entrance.inst_code
AND   degree_cert.admis_area_code *=   stud_entrance.admis_area_code
 AND degree_cert.degree_cert_yr_earned  in(1998,1999)
  AND degree_cert.degree_cert_term_earned = "SP"
 AND degree_cert.degree_cert_yr_earned  *=  empl_qtr_wage.wage_yr_num
 AND empl_qtr_wage.wage_qtr_num = 4
AND   degree_cert.degree_cert_level_code In ("03","05","07","09","10")
 /*AND degree_cert.inst_code = "BLTC"*/

/*Step 2 sum wages */
SELECT obrid,
 campus_code,
 yr_num,
 degree_cert_level_code,
 subject_code,
 degree_cert_yr_earned,
 degree_cert_term_earned,
Sum(wage_amt) AS Max_wage,
inst_code,
```

```
admis_area_code,
DC_term,
state_abbrev
INTO #step2a
FROM #step1
GROUP BY  obrid,
 campus_code,
 yr_num,
 degree_cert_level_code,
 subject_code,
 degree_cert_yr_earned,
 degree_cert_term_earned,
inst_code,
admis_area_code,
DC_term,
state_abbrev
```

/*Step 3a sum to degree level*/

```
SELECT   obrid,
   campus_code,
   yr_num,
  degree_cert_level_code,
  degree_cert_yr_earned,
  degree_cert_term_earned,
  inst_code,
  admis_area_code,
  state_abbrev,
 DC_term,
 Max_wage,
 Count(*) AS dual_major
INTO #step3a
FROM #step2a
GROUP BY   obrid,
  campus_code,
  yr_num,
  degree_cert_level_code,
  degree_cert_yr_earned,
  degree_cert_term_earned,
  inst_code,
  admis_area_code,
   state_abbrev,
  DC_term,
  Max_wage
```

/*Step 4 join subject codes*/

```
SELECT DISTINCT
  #step3a.*,
(CASE
    WHEN dual_major >1 THEN "dual major"
```

```
   ELSE subject_code
END) AS Subject
INTO #step4
FROM   #step3a,  #step2a
WHERE  (#step3a.obrid =   #step2a.obrid )
AND (  #step3a.campus_code =   #step2a.campus_code)
AND (  #step3a.yr_num =   #step2a.yr_num)
AND (  #step3a.degree_cert_level_code =   #step2a.degree_cert_level_code)
AND (  #step3a.degree_cert_yr_earned =   #step2a.degree_cert_yr_earned)
AND (  #step3a.degree_cert_term_earned =   #step2a.degree_cert_term_earned)
 AND (  #step3a.inst_code =   #step2a.inst_code)
AND (  #step3a.admis_area_code =   #step2a.admis_area_code)
AND (  #step3a.DC_term =   #step2a.DC_term)
AND (  #step3a.state_abbrev =   #step2a.state_abbrev)
```

/*Step 5 sum to degree*/

```
SELECT  obrid,
 campus_code,
  yr_num,
 degree_cert_level_code,
 degree_cert_yr_earned,
  degree_cert_term_earned,
  inst_code,
  admis_area_code,
 state_abbrev,
DC_term,
 Max_wage,
 Subject
INTO #step5
FROM #step4
GROUP BY  obrid,
 campus_code,
 yr_num,
 degree_cert_level_code,
 degree_cert_yr_earned,
 degree_cert_term_earned,
 inst_code,
 admis_area_code,
 state_abbrev,
 DC_term,
 Max_wage,
 Subject
```

/*Step 6 join SN*/

```
SELECT  #step5.*,
stud_enroll.campus_code AS SN_campus,
stud_enroll.yr_num AS SN_year,
stud_enroll.term_code,
isnull(stud_enroll.yr_num,0)*4+
```

```
(CASE
WHEN stud_enroll.term_code = "WI" THEN 0
WHEN stud_enroll.term_code = "SP" THEN 1
WHEN stud_enroll.term_code = "SM" THEN 2
ELSE 3
END) AS return_term
INTO #step6
FROM  #step5, stud_enroll
WHERE  #step5.obrid *= stud_enroll.obrid
```

/*Step 7 classify return*/

```
SELECT obrid,
inst_code,
state_abbrev,
Max_wage,
 campus_code,
 yr_num,
 degree_cert_level_code,
 degree_cert_yr_earned,
 degree_cert_term_earned,
 Subject,
DC_term,
return_term,
(CASE
 WHEN return_term > DC_term AND return_term < DC_term+5 THEN "y"
ELSE "n"
END) AS return_class
INTO #step7
FROM #step6
```

/*Step 8 sum to max return*/

```
SELECT obrid,
inst_code,
state_abbrev,
Max_wage,
 campus_code,
 yr_num,
 degree_cert_level_code,
 degree_cert_yr_earned,
 degree_cert_term_earned,
 Subject,
Max(return_class) AS max_return_term
INTO #step8
FROM #step7
GROUP BY obrid,
inst_code,
state_abbrev,
Max_wage,
 campus_code,
```

```
 yr_num,
 degree_cert_level_code,
 degree_cert_yr_earned,
 degree_cert_term_earned,
 Subject
```

/*Step 9 summarize*/

```
/*Step 9 summarize*/
SELECT  inst_code,
 campus_code,
 yr_num,
 degree_cert_level_code,
(CASE
 WHEN state_abbrev ="OH" THEN "Ohio"
WHEN state_abbrev ="ZZ" THEN "Foreign"
ELSE "Other State"
END) AS Residency,
(CASE
 WHEN isnull(Max_wage,0)=0 THEN "No Job"
WHEN Max_wage > 0 and Max_wage < 4000 THEN "Part time job"
ELSE "Full time job"
END) AS Job_Status,
Avg(isnull(Max_wage,0)*4) AS Avg_Annual_Wage ,
substring(Subject,1,2) AS Subject_Code_2_digit,
(CASE
 WHEN max_return_term="n" THEN "No return"
ELSE "Return to school"
END) AS Return_Status,
Count(*) AS Graduates
INTO #step9
FROM  #step8
GROUP BY  inst_code,
 campus_code,
 yr_num,
 degree_cert_level_code,
(CASE
 WHEN state_abbrev ="OH" THEN "Ohio"
WHEN state_abbrev ="ZZ" THEN "Foreign"
ELSE "Other State"
END),
(CASE
 WHEN isnull(Max_wage,0)=0 THEN "No Job"
WHEN Max_wage > 0 and Max_wage < 4000 THEN "Part time job"
ELSE "Full time job"
END),
substring(Subject,1,2)  ,
(CASE
 WHEN max_return_term="n" THEN "No return"
ELSE "Return to school"
```

```
END)
SELECT * from #step9
/* Step 10 campus report*/
SELECT
  degree_cert_level_code degree_level,
inst_campus.campus_type_code sector,
inst_campus_1.campus_name inst,
inst_campus.campus_name campus,
sum (Graduates) total_grads,
sum(CASE WHEN Return_Status = "Return to school"
       AND Job_Status <> "No Job" THEN Graduates
     ELSE 0 END) job_and_return,
sum(CASE WHEN Return_Status = "No return"
       AND Job_Status <> "No Job" THEN Graduates
     ELSE 0 END) job_and_no_return,
sum(CASE WHEN Return_Status = "Return to school"
       AND Job_Status = "No Job" THEN Graduates
     ELSE 0 END) no_job_and_return,
sum(CASE WHEN Return_Status = "Return to school"
       OR Job_Status <> "No Job" THEN Graduates
     ELSE 0 END) job_or_return,
sum(CASE WHEN Return_Status = "No return"
       AND Job_Status = "No Job" THEN Graduates
     ELSE 0 END) no_job_and_no_return
FROM #step9, inst_campus, inst_campus inst_campus_1
WHERE
inst_campus.campus_code = #step9.campus_code
AND inst_campus_1.campus_code = #step9.inst_code
AND inst_campus_1.campus_code = inst_campus_1.inst_code
GROUP BY
inst_campus.campus_type_code,
inst_campus_1.campus_name,
inst_campus.campus_name,
 degree_cert_level_code
/* Step 11a level outcome report*/
SELECT
 degree_cert_level_code,
sum (Graduates) total_grads,
sum(CASE WHEN Return_Status = "Return to school"
       AND Job_Status <> "No Job" THEN Graduates
     ELSE 0 END) job_and_return,
sum(CASE WHEN Return_Status = "No return"
       AND Job_Status <> "No Job" THEN Graduates
     ELSE 0 END) job_and_no_return,
sum(CASE WHEN Return_Status = "Return to school"
       AND Job_Status = "No Job" THEN Graduates
     ELSE 0 END) no_job_and_return,
sum(CASE WHEN Return_Status = "Return to school"
       OR Job_Status <> "No Job" THEN Graduates
```

```
    ELSE 0 END) job_or_return,
sum(CASE WHEN Return_Status = "No return"
      AND Job_Status = "No Job" THEN Graduates
    ELSE 0 END) no_job_and_no_return
FROM #step9, inst_campus
WHERE
inst_campus.campus_code = #step9.campus_code

GROUP BY
 degree_cert_level_code
/* Step 11b sector outcome report*/
SELECT
inst_campus.campus_type_code,
sum (Graduates) total_grads,
sum(CASE WHEN Return_Status = "Return to school"
      AND Job_Status <> "No Job" THEN Graduates
    ELSE 0 END) job_and_return,
sum(CASE WHEN Return_Status = "No return"
      AND Job_Status <> "No Job" THEN Graduates
    ELSE 0 END) job_and_no_return,
sum(CASE WHEN Return_Status = "Return to school"
      AND Job_Status = "No Job" THEN Graduates
    ELSE 0 END) no_job_and_return,
sum(CASE WHEN Return_Status = "Return to school"
      OR Job_Status <> "No Job" THEN Graduates
    ELSE 0 END) job_or_return,
sum(CASE WHEN Return_Status = "No return"
      AND Job_Status = "No Job" THEN Graduates
    ELSE 0 END) no_job_and_no_return
FROM #step9, inst_campus
WHERE
inst_campus.campus_code = #step9.campus_code
 AND degree_cert_level_code = "03"
GROUP BY
inst_campus.campus_type_code
/* Step 12 salaries for Assoc degrees*/
SELECT
 inst_campus.campus_type_code sector,
sum (Graduates) full_time_job,
sum(Avg_Annual_Wage*Graduates) total_wage
INTO #step12
FROM #step9, inst_campus
WHERE
inst_campus.campus_code = #step9.campus_code
AND Job_Status = "Full time job"
AND degree_cert_level_code = "03"
GROUP BY
inst_campus.campus_type_code
/*step 13 print report*/
```

```
SELECT #step12.*,
total_wage / full_time_job avg_wage
FROM #step12

/*Step 0a create common subjects table*/
CREATE TABLE #common_subjects
(subject char(3),
name char(40))
/*Step 0b populate the table*/
INSERT #common_subjects values ('X01',	'Agriculture')
INSERT #common_subjects values ('X02',	'Agriculture')
INSERT #common_subjects values ('X04',	'Architecture')
INSERT #common_subjects values ('X05',	'Area and ethnic studies')
INSERT #common_subjects values ('X26',	'Biological/life sciences')
INSERT #common_subjects values ('X08',	'Business/marketing')
INSERT #common_subjects values ('X52',	'Business/marketing')
INSERT #common_subjects values ('X09',	'Communications/communication technologies')
INSERT #common_subjects values ('X10',	'Communications/communication technologies')
INSERT #common_subjects values ('X11',	'Computer and information sciences')
INSERT #common_subjects values ('X13',	'Education')
INSERT #common_subjects values ('X14',	'Engineering/engineering technologies')
INSERT #common_subjects values ('X15',	'Engineering/engineering technologies')
INSERT #common_subjects values ('X23',	'English')
INSERT #common_subjects values ('X16',	'Foreign languages and literature')
INSERT #common_subjects values ('X51',	'Health professions and related sciences')
INSERT #common_subjects values ('X19',	'Home economics and vocational home economics')
INSERT #common_subjects values ('X20',	'Home economics and vocational home economics')
INSERT #common_subjects values ('X30',	'Interdisciplinary studies')
INSERT #common_subjects values ('X22',	'Law/legal studies')
INSERT #common_subjects values ('X24',	'Liberal arts/general studies')
INSERT #common_subjects values ('X25',	'Library science')
INSERT #common_subjects values ('X27',	'Mathematics')
INSERT #common_subjects values ('X28',	'Military science and technologies')
INSERT #common_subjects values ('X29',	'Military science and technologies')
INSERT #common_subjects values ('X03',	'Natural resources/environmental science')
INSERT #common_subjects values ('X31',	'Parks and recreation')
INSERT #common_subjects values ('X12',	'Personal and miscellaneous services')
INSERT #common_subjects values ('X38',	'Philosophy, religion, theology')
INSERT #common_subjects values ('X39',	'Philosophy, religion, theology')
INSERT #common_subjects values ('X40',	'Physical sciences')
INSERT #common_subjects values ('X41',	'Physical sciences')
INSERT #common_subjects values ('X43',	'Protective services/public administration')
INSERT #common_subjects values ('X44',	'Protective services/public administration')
INSERT #common_subjects values ('X42',	'Psychology')
INSERT #common_subjects values ('X45',	'Social sciences and history')
INSERT #common_subjects values ('X46',	'Trade and industry')
INSERT #common_subjects values ('X47',	'Trade and industry')
INSERT #common_subjects values ('X48',	'Trade and industry')
INSERT #common_subjects values ('X49',	'Trade and industry')
```

```
INSERT #common_subjects values ('X50',          'Visual and performing arts')
INSERT #common_subjects values ('Xdu',          'Multiple Degrees')
INSERT #common_subjects values ('X32',          'Basic Skills')
INSERT #common_subjects values ('XTR',          'Transfer Module')
/* Step 14 salaries for subjects*/
SELECT
name,
sum(CASE WHEN degree_cert_level_code = "03" THEN Graduates ELSE 0 END) assoc_grads,
sum(CASE WHEN degree_cert_level_code = "03" THEN Avg_Annual_Wage*Graduates ELSE 0 END)
assoc_wage,
sum(CASE WHEN degree_cert_level_code = "05" THEN Graduates ELSE 0 END) bac_grads,
sum(CASE WHEN degree_cert_level_code = "05" THEN Avg_Annual_Wage*Graduates ELSE 0 END)
bac_wage,
sum(CASE WHEN degree_cert_level_code = "07" THEN Graduates ELSE 0 END) masters_grads,
sum(CASE WHEN degree_cert_level_code = "07" THEN Avg_Annual_Wage*Graduates ELSE 0 END)
masters_wage,
sum(CASE WHEN degree_cert_level_code = "10" THEN Graduates ELSE 0 END) prof_grads,
sum(CASE WHEN degree_cert_level_code = "10" THEN Avg_Annual_Wage*Graduates ELSE 0 END)
prof_wage,
sum(CASE WHEN degree_cert_level_code = "09" THEN Graduates ELSE 0 END) doc_grads,
sum(CASE WHEN degree_cert_level_code = "09" THEN Avg_Annual_Wage*Graduates ELSE 0 END)
doc_wage
INTO #step14
FROM #step9, #common_subjects
WHERE
Subject_Code_2_digit = substring(subject,2,2)
AND Job_Status = "Full time job"
GROUP BY
name
/*step 14a print report*/
SELECT
name,
CASE WHEN assoc_grads =0 THEN -1 WHEN assoc_grads < 5 THEN 0 ELSE assoc_grads END assoc_grads,
CASE WHEN assoc_grads =0 THEN -1 WHEN assoc_grads < 5 THEN 0 ELSE assoc_wage/assoc_grads END
assoc_wage,
CASE WHEN bac_grads =0 THEN -1 WHEN bac_grads < 5 THEN 0 ELSE bac_grads END bac_grads,
CASE WHEN bac_grads =0 THEN -1 WHEN bac_grads < 5 THEN 0 ELSE bac_wage/bac_grads END
bac_wage,
CASE WHEN masters_grads =0 THEN -1 WHEN masters_grads < 5 THEN 0 ELSE masters_grads END
masters_grads,
CASE WHEN masters_grads =0 THEN -1 WHEN masters_grads < 5 THEN 0 ELSE
masters_wage/masters_grads END masters_wage,
CASE WHEN prof_grads =0 THEN -1 WHEN prof_grads < 5 THEN 0 ELSE prof_grads END prof_grads,
CASE WHEN prof_grads =0 THEN -1 WHEN prof_grads < 5 THEN 0 ELSE prof_wage/prof_grads END
prof_wage,
CASE WHEN doc_grads =0 THEN -1 WHEN doc_grads < 5 THEN 0 ELSE doc_grads END doc_grads,
CASE WHEN doc_grads =0 THEN -1 WHEN doc_grads < 5 THEN 0 ELSE doc_wage/doc_grads END
doc_wage
FROM #step14
/* step15 statewide average salary*/
SELECT
sum(CASE WHEN degree_cert_level_code = "03" THEN Graduates ELSE 0 END) assoc_grads,
```

```
sum(CASE WHEN degree_cert_level_code = "03" THEN Avg_Annual_Wage*Graduates ELSE 0 END)
assoc_wage,
sum(CASE WHEN degree_cert_level_code = "05" THEN Graduates ELSE 0 END) bac_grads,
sum(CASE WHEN degree_cert_level_code = "05" THEN Avg_Annual_Wage*Graduates ELSE 0 END)
bac_wage,
sum(CASE WHEN degree_cert_level_code = "07" THEN Graduates ELSE 0 END) masters_grads,
sum(CASE WHEN degree_cert_level_code = "07" THEN Avg_Annual_Wage*Graduates ELSE 0 END)
masters_wage,
sum(CASE WHEN degree_cert_level_code = "10" THEN Graduates ELSE 0 END) prof_grads,
sum(CASE WHEN degree_cert_level_code = "10" THEN Avg_Annual_Wage*Graduates ELSE 0 END)
prof_wage,
sum(CASE WHEN degree_cert_level_code = "09" THEN Graduates ELSE 0 END) doc_grads,
sum(CASE WHEN degree_cert_level_code = "09" THEN Avg_Annual_Wage*Graduates ELSE 0 END)
doc_wage
INTO #step15
FROM #step9
WHERE
Job_Status = "Full time job"
/*step16 statewide report*/
SELECT
CASE WHEN assoc_grads =0 THEN -1 WHEN assoc_grads < 5 THEN 0 ELSE assoc_grads END assoc_grads,
CASE WHEN assoc_grads =0 THEN -1 WHEN assoc_grads < 5 THEN 0 ELSE assoc_wage/assoc_grads END
assoc_wage,
CASE WHEN bac_grads =0 THEN -1 WHEN bac_grads < 5 THEN 0 ELSE bac_grads END bac_grads,
CASE WHEN bac_grads =0 THEN -1 WHEN bac_grads < 5 THEN 0 ELSE bac_wage/bac_grads END
bac_wage,
CASE WHEN masters_grads =0 THEN -1 WHEN masters_grads < 5 THEN 0 ELSE masters_grads END
masters_grads,
CASE WHEN masters_grads =0 THEN -1 WHEN masters_grads < 5 THEN 0 ELSE
masters_wage/masters_grads END masters_wage,
CASE WHEN prof_grads =0 THEN -1 WHEN prof_grads < 5 THEN 0 ELSE prof_grads END prof_grads,
CASE WHEN prof_grads =0 THEN -1 WHEN prof_grads < 5 THEN 0 ELSE prof_wage/prof_grads END
prof_wage,
CASE WHEN doc_grads =0 THEN -1 WHEN doc_grads < 5 THEN 0 ELSE doc_grads END doc_grads,
CASE WHEN doc_grads =0 THEN -1 WHEN doc_grads < 5 THEN 0 ELSE doc_wage/doc_grads END
doc_wage
FROM #step15
```

8. Degrees by Subject and Level

```
for bac and above report
SELECT
inst_campus.campus_name,
sum(CASE WHEN degree_cert.degree_cert_level_code = "05" THEN 1 ELSE 0 END) Bacs,
sum(CASE WHEN degree_cert.degree_cert_level_code = "07" THEN 1 ELSE 0 END) MS,
sum(CASE WHEN degree_cert.degree_cert_level_code = "10" THEN 1 ELSE 0 END) Prof,
sum(CASE WHEN degree_cert.degree_cert_level_code = "09" THEN 1 ELSE 0 END) Doc
From degree_cert, inst_campus
WHERE
inst_campus.campus_code = degree_cert.campus_code
```

```
AND degree_cert.yr_num = 1999
AND degree_cert_level_code in("05","07","09","10")
GROUP BY inst_campus.campus_name
```

Assoc by campus report

```
SELECT
inst_campus.campus_type_code Sector,
inst_campus_1.campus_name Inst,
inst_campus.campus_name Campus,
Count(*) Assoc_Degrees
from degree_cert, inst_campus, inst_campus inst_campus_1
WHERE
degree_cert.campus_code = inst_campus.campus_code
AND inst_campus.inst_code = inst_campus_1.inst_code
AND inst_campus_1.inst_code = inst_campus_1.campus_code
AND degree_cert.degree_cert_level_code = "03"
AND degree_cert.yr_num = 1999
GROUP BY
inst_campus.campus_type_code,
inst_campus_1.campus_name,
inst_campus.campus_name
```

for subject report for certificates

/*Step 0a create common subjects table*/

```
CREATE TABLE #common_subjects
(subject char(3),
name char(40))
```

/*Step 0b populate the table*/

```
INSERT #common_subjects values ('X01',  'Agriculture')
INSERT #common_subjects values ('X02',  'Agriculture')
INSERT #common_subjects values ('X04',  'Architecture')
INSERT #common_subjects values ('X05',  'Area and ethnic studies')
INSERT #common_subjects values ('X26',  'Biological/life sciences')
INSERT #common_subjects values ('X08',  'Business/marketing')
INSERT #common_subjects values ('X52',  'Business/marketing')
INSERT #common_subjects values ('X09',  'Communications/communication technologies')
INSERT #common_subjects values ('X10',  'Communications/communication technologies')
INSERT #common_subjects values ('X11',  'Computer and information sciences')
INSERT #common_subjects values ('X13',  'Education')
INSERT #common_subjects values ('X14',  'Engineering/engineering technologies')
INSERT #common_subjects values ('X15',  'Engineering/engineering technologies')
INSERT #common_subjects values ('X23',  'English')
INSERT #common_subjects values ('X16',  'Foreign languages and literature')
INSERT #common_subjects values ('X51',  'Health professions and related sciences')
INSERT #common_subjects values ('X19',  'Home economics and vocational home
                                         economics')
INSERT #common_subjects values ('X20',  'Home economics and vocational home
                                         economics')
INSERT #common_subjects values ('X30',  'Interdisciplinary studies')
```

```
INSERT #common_subjects values ('X22',                    'Law/legal studies')
INSERT #common_subjects values ('X24',                    'Liberal arts/general studies')
INSERT #common_subjects values ('X25',                    'Library science')
INSERT #common_subjects values ('X27',                    'Mathematics')
INSERT #common_subjects values ('X28',                    'Military science and technologies')
INSERT #common_subjects values ('X29',                    'Military science and technologies')
INSERT #common_subjects values ('X03',                    'Natural resources/environmental science')
INSERT #common_subjects values ('X31',                    'Parks and recreation')
INSERT #common_subjects values ('X12',                    'Personal and miscellaneous services')
INSERT #common_subjects values ('X38',                    'Philosophy, religion, theology')
INSERT #common_subjects values ('X39',                    'Philosophy, religion, theology')
INSERT #common_subjects values ('X40',                    'Physical sciences')
INSERT #common_subjects values ('X41',                    'Physical sciences')
INSERT #common_subjects values ('X43',                    'Protective services/public administration')
INSERT #common_subjects values ('X44',                    'Protective services/public administration')
INSERT #common_subjects values ('X42',                    'Psychology')
INSERT #common_subjects values ('X45',                    'Social sciences and history')
INSERT #common_subjects values ('X46',                    'Trade and industry')
INSERT #common_subjects values ('X47',                    'Trade and industry')
INSERT #common_subjects values ('X48',                    'Trade and industry')
INSERT #common_subjects values ('X49',                    'Trade and industry')
INSERT #common_subjects values ('X50',                    'Visual and performing arts')
INSERT #common_subjects values ('Xdu',                    'Multiple Degrees')
INSERT #common_subjects values ('X32',                    'Basic Skills')
INSERT #common_subjects values ('XTR',                    'Transfer Module')

SELECT
inst_campus.campus_type_code,
inst_campus_1.campus_name,
inst_campus.campus_name,
name,
sum(CASE WHEN degree_cert.degree_cert_level_code = "01" THEN 1 ELSE 0 END) cert_01,
sum(CASE WHEN degree_cert.degree_cert_level_code = "02" THEN 1 ELSE 0 END) cert_02,
sum(CASE WHEN degree_cert.degree_cert_level_code = "04" THEN 1 ELSE 0 END) cert_04,
sum(CASE WHEN degree_cert.degree_cert_level_code = "06" THEN 1 ELSE 0 END) cert_06,
sum(CASE WHEN degree_cert.degree_cert_level_code = "08" THEN 1 ELSE 0 END) cert_08,
sum(CASE WHEN degree_cert.degree_cert_level_code = "TM" THEN 1 ELSE 0 END) cert_TM
From degree_cert, #common_subjects, inst_campus, inst_campus inst_campus_1
WHERE
degree_cert.campus_code = inst_campus.campus_code
AND inst_campus.inst_code = inst_campus_1.inst_code
AND inst_campus_1.inst_code = inst_campus_1.campus_code
AND yr_num = 1999
AND  degree_cert_level_code in("01","02","06","08","TM")
AND substring(degree_cert.subject_code,1,2) = substring(subject,2,2)
GROUP BY
inst_campus.campus_type_code,
inst_campus_1.campus_name,
inst_campus.campus_name,
```

name

for subject report for degrees

```
/*Step 0a create common subjects table*/
CREATE TABLE #common_subjects
(subject char(3),
name char(40))
/*Step 0b populate the table*/
INSERT #common_subjects values ('X01',  'Agriculture')
INSERT #common_subjects values ('X02',  'Agriculture')
INSERT #common_subjects values ('X04',  'Architecture')
INSERT #common_subjects values ('X05',  'Area and ethnic studies')
INSERT #common_subjects values ('X26',  'Biological/life sciences')
INSERT #common_subjects values ('X08',  'Business/marketing')
INSERT #common_subjects values ('X52',  'Business/marketing')
INSERT #common_subjects values ('X09',  'Communications/communication technologies')
INSERT #common_subjects values ('X10',  'Communications/communication technologies')
INSERT #common_subjects values ('X11',  'Computer and information sciences')
INSERT #common_subjects values ('X13',  'Education')
INSERT #common_subjects values ('X14',  'Engineering/engineering technologies')
INSERT #common_subjects values ('X15',  'Engineering/engineering technologies')
INSERT #common_subjects values ('X23',  'English')
INSERT #common_subjects values ('X16',  'Foreign languages and literature')
INSERT #common_subjects values ('X51',  'Health professions and related sciences')
INSERT #common_subjects values ('X19',  'Home economics and vocational home
                                         economics')
INSERT #common_subjects values ('X20',  'Home economics and vocational home
                                         economics')
INSERT #common_subjects values ('X30',  'Interdisciplinary studies')
INSERT #common_subjects values ('X22',  'Law/legal studies')
INSERT #common_subjects values ('X24',  'Liberal arts/general studies')
INSERT #common_subjects values ('X25',  'Library science')
INSERT #common_subjects values ('X27',  'Mathematics')
INSERT #common_subjects values ('X28',  'Military science and technologies')
INSERT #common_subjects values ('X29',  'Military science and technologies')
INSERT #common_subjects values ('X03',  'Natural resources/environmental science')
INSERT #common_subjects values ('X31',  'Parks and recreation')
INSERT #common_subjects values ('X12',  'Personal and miscellaneous services')
INSERT #common_subjects values ('X38',  'Philosophy, religion, theology')
INSERT #common_subjects values ('X39',  'Philosophy, religion, theology')
INSERT #common_subjects values ('X40',  'Physical sciences')
INSERT #common_subjects values ('X41',  'Physical sciences')
INSERT #common_subjects values ('X43',  'Protective services/public administration')
INSERT #common_subjects values ('X44',  'Protective services/public administration')
INSERT #common_subjects values ('X42',  'Psychology')
```

```
INSERT #common_subjects values ('X45',                    'Social sciences and history')
INSERT #common_subjects values ('X46',                    'Trade and industry')
INSERT #common_subjects values ('X47',                    'Trade and industry')
INSERT #common_subjects values ('X48',                    'Trade and industry')
INSERT #common_subjects values ('X49',                    'Trade and industry')
INSERT #common_subjects values ('X50',                    'Visual and performing arts')
INSERT #common_subjects values ('Xdu',                    'Multiple Degrees')
INSERT #common_subjects values ('X32',                    'Basic Skills')
INSERT #common_subjects values ('XTR',                    'Transfer Module')
SELECT
name,
sum(CASE WHEN degree_cert.degree_cert_level_code = "03" THEN 1 ELSE 0 END) Assoc,
sum(CASE WHEN degree_cert.degree_cert_level_code = "05" THEN 1 ELSE 0 END) Bac,
sum(CASE WHEN degree_cert.degree_cert_level_code = "07" THEN 1 ELSE 0 END) Masters,
sum(CASE WHEN degree_cert.degree_cert_level_code = "10" THEN 1 ELSE 0 END) Prof,
sum(CASE WHEN degree_cert.degree_cert_level_code = "09" THEN 1 ELSE 0 END) Doc
From degree_cert, #common_subjects
WHERE
yr_num = 1999
AND  degree_cert_level_code in("03","05","07","10","09")
AND substring(degree_cert.subject_code,1,2) = substring(subject,2,2)
GROUP BY name
```

sector and level

```
SELECT
degree_cert_level_code degree_level,
inst_campus.campus_type_code sector,
count(*) degrees
From degree_cert, inst_campus
WHERE
inst_campus.campus_code = degree_cert.campus_code
AND degree_cert.yr_num = 1999
AND degree_cert_level_code in("01","02","03","TM")
GROUP BY
degree_cert_level_code,
inst_campus.campus_type_code
```

9. Mobility and Concurrent Enrollments

Mobility

```
/**Spring 00 Step 1*/
SELECT  stud_enroll.obrid obrid1,
 Stud_enroll.campus_code campus1
 INTO #step1
FROM  stud_enroll,  stud_enroll stud_enroll_1
WHERE   stud_enroll.admis_area_code="UND"
AND  stud_enroll.yr_num=2000
AND  stud_enroll.term_code="SP"
```

```
AND stud_enroll_1.campus_code<> stud_enroll.campus_code
AND stud_enroll.obrid =  stud_enroll_1.obrid
AND ( stud_enroll_1.yr_num<1999
   OR stud_enroll_1.yr_num=2000
AND  stud_enroll_1.term_code="WI")
/*AND stud_enroll.campus_code = "CNTL"*/
/*step 2 select distinct obrids*/
SELECT DISTINCT
obrid1,
campus1
INTO #step2
FROM #step1
/*step3*/
SELECT
campus1,
count(*) students_mobile
INTO #step3
FROM #step2
GROUP BY campus1
/*SP  Step 4 get all students*/
SELECT DISTINCT
stud_enroll.campus_code,
stud_enroll.obrid
INTO #step4
FROM stud_enroll
WHERE stud_enroll.yr_num=2000
AND stud_enroll.term_code="SP"
AND stud_enroll.admis_area_code="UND"
/*AND stud_enroll.campus_code = "CNTL"*/
/*step5 sum to campus*/
SELECT
#step4.campus_code,
inst_campus.campus_type_code sector,
inst_campus_1.campus_name inst,
inst_campus.campus_name campus,
count(*) students_total_UND
INTO #step5
FROM #step4, inst_campus, inst_campus inst_campus_1
WHERE
#step4.campus_code = inst_campus.campus_code
AND inst_campus.inst_code = inst_campus_1.inst_code
AND inst_campus_1.inst_code = inst_campus_1.campus_code
GROUP BY
#step4.campus_code,
inst_campus.campus_type_code,
inst_campus_1.campus_name,
inst_campus.campus_name
/*step6 join total with mobils*/
SELECT
```

```
sector,
inst,
campus,
students_total_UND,
students_mobile
FROM #step5, #step3
WHERE
#step5.campus_code *= #step3.campus1

concurrent
/*step 7 select AU 1999 for concurrent*/
SELECT  stud_enroll.obrid obrid1,
 stud_enroll.campus_code campus1
 INTO #step7
FROM  stud_enroll,  stud_enroll stud_enroll_1
WHERE   stud_enroll.admis_area_code="UND"
AND  stud_enroll.yr_num=1999
AND  stud_enroll.term_code="AU"
AND stud_enroll_1.campus_code<> stud_enroll.campus_code
AND stud_enroll.obrid =  stud_enroll_1.obrid
AND stud_enroll.yr_num = stud_enroll_1.yr_num
AND stud_enroll.term_code = stud_enroll_1.term_code
/*AND stud_enroll.campus_code = "CNTL"/*
/*step 8 select distinct obrids*/
SELECT DISTINCT
obrid1,
campus1
INTO #step8
FROM #step7
/*step9*/
SELECT
campus1,
count(*) students_mobile
INTO #step9
FROM #step8
GROUP BY campus1
/*SP  Step 10 get all students*/
SELECT DISTINCT
stud_enroll.campus_code,
stud_enroll.obrid
INTO #step10
FROM stud_enroll
WHERE stud_enroll.yr_num=1999
AND stud_enroll.term_code="AU"
AND stud_enroll.admis_area_code="UND"
/*AND stud_enroll.campus_code = "CNTL"*/
/*step11 sum to campus*/
SELECT
#step10.campus_code,
```

```
inst_campus.campus_type_code sector,
inst_campus_1.campus_name inst,
inst_campus.campus_name campus,
count(*) students_total_UND
INTO #step11
FROM #step10, inst_campus, inst_campus inst_campus_1
WHERE
#step10.campus_code = inst_campus.campus_code
AND inst_campus.inst_code = inst_campus_1.inst_code
AND inst_campus_1.inst_code = inst_campus_1.campus_code
GROUP BY
#step10.campus_code,
inst_campus.campus_type_code,
inst_campus_1.campus_name,
inst_campus.campus_name
/*step12 join total with mobils*/
SELECT
sector,
inst,
campus,
students_total_UND,
students_mobile
FROM #step11, #step9
WHERE
#step11.campus_code *= #step9.campus1
```

10. GPAs for transfers vs others

```
/* step 1 select undergrads on UMs in AU 1999*/
Select
stud_enroll.inst_code,
stud_enroll.obrid,
stud_enroll.rank_code,
(CASE WHEN gpa.cum_gpa_hrs <> 0 THEN gpa.cum_gpa_points /gpa.cum_gpa_hrs
    ELSE 0 END) gpa
INTO #step1
FROM stud_enroll, inst_campus, gpa
WHERE
stud_enroll.campus_code = inst_campus.campus_code
AND stud_enroll.inst_code = gpa.inst_code
AND stud_enroll.obrid = gpa.obrid
AND stud_enroll.yr_num = gpa.yr_num
AND stud_enroll.term_code = gpa.term_code
AND inst_campus.campus_type_code = "UM"
AND stud_enroll.yr_num =1999
AND stud_enroll.term_code = "AU"
AND stud_enroll.rank_code in("FR","SO","JR","SR")
AND gpa.cum_gpa_hrs<>0
/*AND stud_enroll.inst_code = "CNTL"*/
```

```
/*step 2 find pervious terms of enrollment*/
SELECT
#step1.*,
stud_enroll.campus_code prev_campus,
stud_enroll.term_code prev_term,
stud_enroll.yr_num prev_year,
stud_enroll.yr_num * 10 +
CASE WHEN stud_enroll.term_code = "WI" THEN 1
        WHEN stud_enroll.term_code = "SP" THEN 2
        WHEN stud_enroll.term_code = "SM" THEN 3
        ELSE 4 END prev_yt
INTO #step2
FROM #step1, stud_enroll, inst_campus
WHERE
#step1.obrid = stud_enroll.obrid
AND (stud_enroll.yr_num = 1998
   AND stud_enroll.term_code in ("WI","SP","AU")
 OR stud_enroll.yr_num = 1999
AND stud_enroll.term_code ="SP")
AND  stud_enroll.campus_code = inst_campus.campus_code
AND inst_campus.campus_type_code in ("CC","SC","TC","UB")
/*step3 join step 2 and 1*/
SELECT
#step1.inst_code,
#step1.obrid,
#step1.rank_code,
#step1.gpa,
#step2.prev_campus,
#step2.prev_term,
#step2.prev_year,
#step2.prev_yt
INTO #step3
FROM #step1, #step2
WHERE
#step1.inst_code *= #step2.inst_code
AND #step1.obrid *= #step2.obrid
AND #step1.rank_code *= #step2.rank_code
AND #step1.gpa *= #step2.gpa
/*step4 pick most recent other term*/
SELECT
#step3.inst_code,
#step3.obrid,
#step3.rank_code,
#step3.gpa,
Max(#step3.prev_yt) max_term
INTO #step4
FROM #step3
GROUP BY
#step3.inst_code,
```

```
#step3.obrid,
#step3.rank_code,
#step3.gpa
/*step5 classify cases*/
SELECT
#step4.inst_code,
#step4.obrid,
#step4.rank_code,
#step4.gpa,
CASE WHEN max_term is null THEN "no transfer"
  WHEN max_term = 19992 THEN "recent transfer"
ELSE "older transfer" END student_class
INTO #step5
FROM #step4
/*step6 sum to Inst*/
SELECT
#step5.inst_code,
rank_code,
student_class,
count(*) students,
AVG(gpa) average_gpa
FROM #step5
GROUP BY
#step5.inst_code,
rank_code,
student_class
```