

SAS SQL for CPS fiscal year dataset

Source: Corporate Personnel System (CPS) data warehouse

Database name: uc_cdw

Database server: dwp1.ucop.edu

Source schema name: Personnel

Source table name: PER_FY_ALL

Result table name: FiscalYear_allyears

Date of run: January 27, 2021

*/*This code pulls all data between the fiscal years listed to create a Fiscal Year Personnel data source*/*

/ This runs a macro to handle each year individually for a faster processing*/*

%macro YearCycle;

*/*This step pulls all years including and between the years listed. Each year is substituted in the table names and the where clause.*/*

%do yr = 2011 %to 2020;

/ Grabs fields required for output and derivations*/*

PROC SQL;

create table FY_FTE_&yr as

select

 eff_date_fy,
 employee_number,
 empl_emplid,
 dos_code_cps,
 dos_desc_cps,
 pers_pgm,
 Personnel_program,
 location_1,
 location_2,
 location_3,
 loc_indicator,
 empl_bus_unit,
 appt_type_code,
 appt_type_code_pri,
 academic_rank_code,
 paid_over_code,
 appt_basis_code,

function_code,
function_name,
fund_group_code,
fund,
TRIM(Org_1_Code) as org_1_code,
Org_1_Title,
TRIM(Org_2_Code) as org_2_code,
Org_2_Title,
TRIM(Org_3_Code) as org_3_code,
Org_3_Title,
SAU_Code,
UAS_Discipline_Code,
Home_Department_Name,
Home_Department_Code,
Class_Ttl_Outl_Code_Appt,
TRIM(Class_Ttl_Outl_Name_Appt) as Class_Ttl_Outl_Name_Appt,
title_code,
title_name,
title_code_pri_appt,
primary_job_ind,
payroll_title_unit_rep_code,
empl_unit_code,
sex_code,
ethnic_origin_code,
IPEDS_hispanic_flag,
IPEDS_ethnic_origin_1_code,
IPEDS_ethnic_origin_1_name,
IPEDS_ethnic_origin_2_code,
IPEDS_ethnic_origin_3_code,
IPEDS_ethnic_origin_4_code,
IPEDS_ethnic_origin_5_code,
citizen_status_code,
visa_type,
pps_separation_reason_code,
pps_separation_reason_name,
separation_rsn_code,
separated_rsn_code_acad,
Separation_date format = MMDDYY10.,
date_of_most_recent_hire format = MMDDYY10.,
original_date_of_hire format = MMDDYY10.,
appt_pay_rate,
appt_rate_code,
cat(location_1, employee_number) as unique_id,

*/*Sums hours worked in the fiscal year*/*

sum(TOT_FY_HOURS_WORKED) as TOT_FY_HOURS_WORKED format = n.3,

```
/*Sums Earnings while excluding for certain DOS codes and partial FY 2017-2018 earnings for ASUCLA*/
sum
  (case
    when dos_code_cps in ('XX', 'LV', 'DB', 'DS') then 0
    when eff_date_fy in ('JUN2018') and location_1 = '25' then 0 /*Excludes partial FY
2017-2018 earnings for ASUCLA due to its transition to UCPath in January 2019 */
    else TOT_FY_GROSS_EARNINGS end)
as Total_FY_Earnings format = dollarn.2,
```

```
/*Calculates FTE by dividing over 12 months*/
sum
  (case
    when eff_date_fy in ('JUN2011') and location_1 = '04' and FUND = '69085' then 0 /*
UCLA had a minor amount of negative FTE found under the systemwide assessment in 2010-11*/
    when eff_date_fy in ('JUN2018') and location_1 = '25' then 0
    /*Excludes partial FY 2017-2018 FTE for ASUCLA due to its transition to UCPath in January 2018
*/
    else TOT_FY_TIME_DISTRIB end/12)
as FTE_Monthly format = n.3,
```

```
/* Calculates FTE dividing by Paid Over time period or 12 months*/
input(paid_over_code, n.0) as Paid_over_num format = n.0,
case when missing(calculated paid_over_num) then 12 else calculated paid_over_num END as
Paid_over_full,
sum
  (case
    when eff_date_fy in ('JUN2011') and location_1 = '04' and FUND = '69085' then 0
    /* UCLA had a minor amount of negative FTE found under the systemwide assessment in 2010-
11*/
    when eff_date_fy in ('JUN2018') and location_1 = '25' then 0
    /*Excludes partial FY 2017-2018 FTE for ASUCLA due to its transition to UCPath in
January 2018 */
    else TOT_FY_TIME_DISTRIB end/calculated Paid_over_full)
as FTE_PaidOver format = n.3,
```

```
/* Calculates FTE dividing by Appt Basis time period or 12 months*/
input(appt_basis_code, n.0) as appt_basis_num format = n.0,
case when missing(calculated appt_basis_num) then 12 else calculated appt_basis_num END as
appt_basis_full,
sum
  (case
    when eff_date_fy in ('JUN2011') and location_1 = '04' and FUND = '69085' then 0
    /* UCLA had a minor amount of negative FTE found under the systemwide assessment in 2010-
11*/
    when eff_date_fy in ('JUN2018') and location_1 = '25' then 0
    /*Excludes partial FY 2017-2018 FTE for ASUCLA due to its transition to UCPath in
January 2018 */
```

```
else TOT_FY_TIME_DISTRIB end/calculated appt_basis_full)
as FTE_Appt_Basis format = n.3,
```

```
/*Creates a flag for ASUCLA exclusion from partial FY 2017-2018*/
```

```
Case
  When eff_date_fy in ('JUN2018') and location_1 = '25' Then 'Exclude'
  Else 'Include'
End as ASUCLA_exclusion_flag,
```

```
/*Appointment Pay Rate calculation based on the nature of the rate of pay for the appointment. */
```

```
Case
  When appt_rate_code = 'H' Then (appt_pay_rate* 2088)
  When appt_rate_code = 'M' Then (appt_pay_rate * 12)
  Else appt_pay_rate
  End as Appt_Pay_Rate_Annualized,
```

```
/*Creates DOS flag for only Base FTE Codes*/
```

```
Case
  When dos_code_cps in ('AW','AV','AX','AM','HR','HD','HY','NG') Then 'Base'
  Else 'Non-Base'
  End as DOS_BaseFTE,
```

```
/*Creates FTE for only base pay codes*/
```

```
Case
  When calculated DOS_BaseFTE = 'Non-Base' Then 0
  Else calculated FTE_Monthly
  End as FTE_Base format = n.3,
```

```
/*Creates DOS flag for only Regular FTE Codes*/
```

```
Case
  When dos_code_cps in ('AW','AV','AX','AM','HR','HD','HY','RT','NG') Then 'Regular'
  Else 'Non-Regular'
  End as DOS_RegularFTE,
```

```
/*Creates FTE for only Regular pay codes*/
```

```
Case
  When calculated DOS_RegularFTE = 'Non-Regular' Then 0
  Else calculated FTE_Monthly
  End as FTE_Regular format = n.3,
```

```
/*Creates DOS flag for only Base Earnings Codes*/
```

```
Case
  When dos_code_cps in ('AW','AV','AX','AM','HR','HD') Then 'Base'
  Else 'Non-Base'
  End as DOS_BaseEarnings,
```

```
/* Assigns Names to Location 1*/
```

Case

```
When location_1 = '01' then 'Berkeley'  
When location_1 = '02' then 'San Francisco'  
When location_1 = '03' then 'Davis'  
When location_1 = '04' then 'Los Angeles'  
When location_1 = '05' then 'Riverside'  
When location_1 = '06' then 'San Diego'  
When location_1 = '07' then 'Santa Cruz'  
When location_1 = '08' then 'Santa Barbara'  
When location_1 = '09' then 'Irvine'  
When location_1 = '10' then 'Merced'  
When location_1 = '18' then 'Agriculture & Natural Resources'  
When location_1 = '25' then 'ASUCLA'  
END as Location_1_Name,
```

/* Uses location 2, location 1, and SAU code to separate UCOP and ANR and create a complete location field called "Location Full"*/

Case

```
When location_2 = '2' and sau_code = '4' THEN 'Agriculture & Natural Resources'  
When empl_bus_unit = 'UCANR' THEN 'Agriculture & Natural Resources'  
When location_2 = '2' and location_1 <> '25' and sau_code <> '4' THEN 'Office of the President &
```

SWP'

```
Else calculated location_1_Name  
END as Location_full,
```

/*General Campus/UC Health split, using loc 3, CTO, org titles, and home department code to correct for campus miscategorizations*/

Case

```
When LOCATION_3 = '2' or class_ttl_outl_code_appt = '446' THEN 'UC Health'
```

/*Fixes for Irvine, Riverside, Los Angeles, and San Diego*/

```
When LOCATION_3 = '1' AND (  
    (LOCATION_1 = '09' AND TRIM(ORG_1_CODE) in ('99030','9MEDC','92000','99701')) OR  
    (LOCATION_1 = '05' AND TRIM(ORG_1_CODE) = '5ORG40') OR  
    (LOCATION_1 = '04' AND TRIM(ORG_1_CODE) in ('44000','42200','43200')) OR  
    (LOCATION_1 = '06' AND TRIM(ORG_1_CODE) in ('6000003','6 3','6000005','6  
5','60003','60008','6000008','60005')) OR  
    (LOCATION_1 = '06' AND SUBSTR(Home_Department_Code,1,3)='003') OR  
    (LOCATION_1 = '06' AND SUBSTR(Home_Department_Code,1,4)='0003')  
)
```

```
THEN 'UC Health'
```

```
When (calculated Location_full = 'Office of the President & SWP' AND TRIM(ORG_2_CODE) in  
( '48610','48620')) THEN 'UC Health'
```

```
When (empl_bus_unit = 'UCOP1' AND TRIM(ORG_2_CODE) in ('48610','48620')) THEN 'UC  
Health'
```

```
ELSE 'General Campus'
```

```
END as Service_focus,
```

/*Med Center Breakout by function*/

Case

When function_code = '42' and calculated Service_focus = 'UC Health' Then 'Med Center'
Else 'Non-Med Center'
END as MedCntr_breakout_Func42,

/*Flags Excluded Groups (Emeriti, Supp Pay, etc.)*/

Case

When Class_Ttl_Outl_Code_Appt in ('016','216','316','346','999') then 'Exclude'
Else 'Include'
END as CTO_Exclude_Flag,

/*This statement flags if the title codes we choose as primary are the same as the primary title code in CPS*/

Case

When title_code <> title_code_pri_appt THEN 'Non-primary CPS'
Else 'Primary CPS'
END as Primary_Title_CPS_flag,

/*fills in blanks for appt_type_code using appt_type_code_pri and creates new field appt_type_use*/

Case

When appt_type_code = '' THEN appt_type_code_pri
Else appt_type_code
END as Appt_type_use,

/*Personnel Program Academic and Non-Academic grouping*/

case

When personnel_program = 'A' Then 'Academic'
Else 'Non-Academic'
END as Pers_Prgm_Academic,

/*Sorts appt_type_use in career/non-career and creates new field*/

Case

When personnel_program = 'A' then 'Academic'
When calculated appt_type_use in ('2','7') THEN 'Career Staff'
Else 'Non-career Staff'
END as Appt_career,

/*Personnel Program Categories breaks managers out from senior professionals in MSP, breaks students out from PSS, and for several years some employees were miscategorized with the CTO of M05, which is used for those in the SMG personnel program when they should have been in the 'Managers' CTO of M10.*/

Case

```
When class_ttl_outl_code_appt in
('010','011','012','030','031','040','041','042','114','124','210','211','212','214','218','224','520','521','522','
530','531','532') THEN 'Faculty - Ladder-rank and Equivalent'
When class_ttl_outl_code_appt in ('311','317','323','335','341') THEN 'Faculty - Clinical/In-
Residence/Adjunct'
When class_ttl_outl_code_appt in ('221','225','357') THEN 'Faculty - Lecturers'
When class_ttl_outl_code_appt = '446' THEN 'Medical Interns/Residents'
When class_ttl_outl_code_appt in ('575','577') THEN 'Postdoctoral Scholars'
When class_ttl_outl_code_appt in ('426','436','456','467','477') THEN 'Student
Teaching/Research Assistants'
When calculated Pers_Prgm_Academic = 'Academic' then 'Other Academic Employees'
When calculated appt_type_use = '4' THEN 'Student Staff'
When class_ttl_outl_code_appt = 'M10' THEN 'MSP - Managers'
When PERS_PGM = 'MSP' AND class_ttl_outl_code_appt = 'M05' THEN 'MSP - Managers'
When PERS_PGM = 'SMG' or Personnel_program = '3' or class_ttl_outl_code_appt = 'M05' THEN
'Senior Management Group'
When PERS_PGM = 'MSP' or personnel_program = '2' THEN 'MSP - Senior Professionals'
else 'PSS - Non-Students'
END as Pers_Prgm_CTO,
```

/*Groups employee CTO categories above*/

```
Case
when calculated Pers_prgm_CTO in ('Faculty - Ladder-rank and Equivalent','Faculty - Clinical/In-
Residence/Adjunct','Faculty - Lecturers') then 'Faculty'
when calculated Pers_prgm_CTO in ('MSP - Managers','Senior Management Group') then
'Management'
when calculated Pers_prgm_CTO in ('Student Staff','Student Teaching/Research Assistants') then
'Student Employees'
when calculated Pers_prgm_CTO in ('MSP - Senior Professionals','PSS - Non-Students') then
'Non-Management Staff'
else 'Non-Faculty Academics'
END as Pers_Prgm_CTO_groups,
```

/* Creates flag for faculty CTOs according to the Academic Personnel Manual (APM)*/

```
Case
When Class_Ttl_Outl_Code_Appt in
('010','011','012','030','031','040','041','042','114','124','210','211','212','214','218','221','224','225','311','
317','323','335','341','357') THEN 'Faculty (excl. Astro/Agro)'
When Class_Ttl_Outl_Code_Appt in ('520','521','522','523','524','530','531','532','533','534')
then 'Agronomist/Astronomer'
else 'Exclude'
END as Faculty_APM_Flag,
```

/* Flags Faculty as Tenure, Tenure-Track, SOE, PSOE or NonTenure */

```
Case
When calculated Faculty_APM_Flag <> 'Exclude' Then
```

```
Case
    When class_ttl_outl_code_appt in ('010','030','040','520','530') then 'Tenured'
    When class_ttl_outl_code_appt in ('011','031','041','214','521','524','531','534')
then 'Tenure-track'
    When class_ttl_outl_code_appt in ('114','124') then 'Tenure-Track Acting'
    When class_ttl_outl_code_appt = '210' then 'SOE'
    When class_ttl_outl_code_appt in ('211','221','224') then 'PSOE'
    Else 'Non-tenure track'
End
Else 'Non-Faculty'
End as Tenure_Flag,
```

/*Academic Senate Flag*/

```
Case
    When calculated Pers_Prgm_Academic = 'Academic' and class_ttl_outl_code_appt in
('010','011','012','114','210','212','211','311','317') THEN 'Senate'
    When calculated Pers_Prgm_Academic = 'Academic' THEN 'Non-Senate'
    Else 'NA'
END as Senate_Flag,
```

/*Flag for Campus Leaders*/

```
Case
    When TITLE_CODE = '000030' THEN 'Chancellor'
    When Class_Ttl_Outl_Code_Appt = 'S27' THEN 'Provost'
    When Class_Ttl_Outl_Code_Appt = 'S21' THEN 'Dean'
    Else 'None'
END as Campus_Leaders,
```

/*Fund Categories: General Funds, Stud Tuition and Fees, Systemwide Assessment, Non-Core*/

```
Case
    When TRIM(FUND_GROUP_CODE) in ('401','402','40411') THEN 'General Funds'
    When TRIM(FUND_GROUP_CODE) in
('40311','40312','403141','403142','403143','403144','403145','403146','403147','403148','403149','403
15','403151','403152','403153','4032') THEN 'Student Tuition & Fees'
    When FUND = '69085' THEN 'Systemwide Assessment'
    ELSE 'Non-Core Funds'
END as Fund_Categories,
```

/*Fund Core vs NonCore*/

```
Case
    When calculated Fund_Categories = 'Non-Core Funds' or calculated Fund_Categories =
'Systemwide Assessment' then 'Non-Core Funds'
    ELSE 'Core Funds'
END as Fund_Core,
```

/*Detailed Fund Categories for AAUDE survey*/

```
Case
```



```
When TRIM(fund_group_code) in ('401','402') THEN 'General Funds'
When TRIM(fund_group_code) in ('4111','4112') THEN 'Reserves'
When TRIM(fund_group_code) in
('40311','40312','40313','403141','403142','403143','403144','403145','403146','403147','403148','4031
49','40315','403151','403152','403153','403161','403162','403163','403164','40319','4032','4034') THEN
'Student Tuition & Fees'
When TRIM(fund_group_code) in ('40411','40412') THEN 'State Appropriations'
When TRIM(fund_group_code) in ('40421','40422') THEN 'State Govt Contracts & Grants'
When TRIM(fund_group_code) in ('40521','40522') THEN 'Local Govt Contracts & Grants'
When TRIM(fund_group_code) in
('4061','40611','40621','40623','40622','40631','40634','40632','40633','41021','41025') THEN 'Federal
Govt Contracts & Grants'
When TRIM(fund_group_code) in
('4071','4073','40721','40722','40723','4081','4082','301','302','303','304') THEN 'Endowment Funds'
When TRIM(fund_group_code) in ('4083','40841','40842','40843') THEN 'Private Contracts &
Grants'
When TRIM(fund_group_code) in ('40911','40912','40913','40914','40919','4092','4093') THEN
'Sales & Service'
ELSE 'Other Operating Funds'
END as Fund_Categories_Details,
```

/*Student Faculty Ratio FTE Flag*/

```
Case
When function_code = '40' and calculated service_focus = 'General Campus'
and fund_group_code not in ('4034','403161','403162') and calculated DOS_BaseFTE = 'Base'
and Class_Ttl_Outl_Code_Appt in
('010','011','012','016','040','042','114','124','210','211','212','214','224','216','221','225','311','323','335','
357','467','577')
Then 'Include'
Else 'Exclude'
END as SFR_flag,
```

/*By Agreement FTE Flag*/

```
Case
When function_code = '40' and calculated service_focus = 'General Campus'
and fund_group_code not in ('4034','403161','403162') and dos_code_cps = 'AF'
and Class_Ttl_Outl_Code_Appt in
('010','011','012','016','040','042','114','124','210','211','212','214','224','216','221','225','311','323','335','
357','467','577')
Then 'Include'
Else 'Exclude'
END as ByAgr_flag,
```

/*Ethnic Origin Groups based on CPS data dictionary and utilizing IPEDS flags and codes (updated on 2/5/2019)*/

Case

```

    when IPEDS_hispanic_flag = 'Y' THEN 'Hispanic'
    when CATS(IPEDS_ethnic_origin_1_code, IPEDS_ethnic_origin_2_code,
IPEDS_ethnic_origin_3_code, IPEDS_ethnic_origin_4_code, IPEDS_ethnic_origin_5_code) in ('AA', 'AAA',
'AAAA', 'AAAAA') THEN 'Black'
    when CATS(IPEDS_ethnic_origin_1_code, IPEDS_ethnic_origin_2_code,
IPEDS_ethnic_origin_3_code, IPEDS_ethnic_origin_4_code, IPEDS_ethnic_origin_5_code) in ('CC', 'CCC',
'CCCC', 'CCCCC') THEN 'American Indian'
    when CATS(IPEDS_ethnic_origin_1_code, IPEDS_ethnic_origin_2_code,
IPEDS_ethnic_origin_3_code, IPEDS_ethnic_origin_4_code, IPEDS_ethnic_origin_5_code) in ('ZZ', 'ZZZ',
'ZZZZ', 'ZZZZZ') THEN 'Native Hawaiian'
    when CATS(IPEDS_ethnic_origin_1_code, IPEDS_ethnic_origin_2_code,
IPEDS_ethnic_origin_3_code, IPEDS_ethnic_origin_4_code, IPEDS_ethnic_origin_5_code) in ('XX', 'XXX',
'XXXX', 'XXXXX') THEN 'Asian'
    when CATS(IPEDS_ethnic_origin_1_code, IPEDS_ethnic_origin_2_code,
IPEDS_ethnic_origin_3_code, IPEDS_ethnic_origin_4_code, IPEDS_ethnic_origin_5_code) in ('FF', 'FFF',
'FFFF', 'FFFFF') THEN 'White'
    when LENGTHN(CATS(IPEDS_ethnic_origin_1_code, IPEDS_ethnic_origin_2_code,
IPEDS_ethnic_origin_3_code, IPEDS_ethnic_origin_4_code, IPEDS_ethnic_origin_5_code)) > 1 THEN
'Two or More Races'
    when IPEDS_ethnic_origin_1_code = 'A' OR IPEDS_ethnic_origin_2_code = 'A' OR
IPEDS_ethnic_origin_3_code = 'A' OR IPEDS_ethnic_origin_4_code = 'A' OR IPEDS_ethnic_origin_5_code
= 'A' THEN 'Black'
    when IPEDS_ethnic_origin_1_code = 'C' OR IPEDS_ethnic_origin_2_code = 'C' OR
IPEDS_ethnic_origin_3_code = 'C' OR IPEDS_ethnic_origin_4_code = 'C' OR IPEDS_ethnic_origin_5_code
= 'C' THEN 'American Indian'
    when IPEDS_ethnic_origin_1_code = 'Z' OR IPEDS_ethnic_origin_2_code = 'Z' OR
IPEDS_ethnic_origin_3_code = 'Z' OR IPEDS_ethnic_origin_4_code = 'Z' OR IPEDS_ethnic_origin_5_code
= 'Z' THEN 'Native Hawaiian'
    when IPEDS_ethnic_origin_1_code = 'X' OR IPEDS_ethnic_origin_2_code = 'X' OR
IPEDS_ethnic_origin_3_code = 'X' OR IPEDS_ethnic_origin_4_code = 'X' OR IPEDS_ethnic_origin_5_code
= 'X' THEN 'Asian'
    when IPEDS_ethnic_origin_1_code = 'F' OR IPEDS_ethnic_origin_2_code = 'F' OR
IPEDS_ethnic_origin_3_code = 'F' OR IPEDS_ethnic_origin_4_code = 'F' OR IPEDS_ethnic_origin_5_code
= 'F' THEN 'White'
    when ethnic_origin_code in ('10','11','12','17') THEN 'Hispanic'
    when ethnic_origin_code in ('02','18') THEN 'Black'
    when ethnic_origin_code in ('01','19') THEN 'American Indian'
    when ethnic_origin_code in ('16','21') THEN 'Native Hawaiian'
    when ethnic_origin_code in ('03','04','05','06','07','08','09','20','23','24') THEN 'Asian'
    when ethnic_origin_code in ('13','14','22','25','26','27') THEN 'White'
    when ethnic_origin_code = '28' THEN 'Two or More Races'
    ELSE 'Unknown'
END as Ethnic_Origin_Groups,

```

/* Defines URG (Under-represented groups) among ethnic origin groups*/
Case

```
when calculated Ethnic_Origin_Groups in ('Black','Hispanic','American Indian','Native Hawaiian')
then 'URG'
else 'Non-URG'
END as URG_flag,
```

```
/* Turns Sex Code into men/women*/
```

```
case sex_code
when 'M' then 'Men'
when 'F' then 'Women'
else 'Unknown'
END as Gender,
```

```
/* Domestic vs. International using citizenship*/
```

```
Case Citizen_Status_Code
When 'C' then 'Domestic'
Else 'International'
End as Citizenship,
```

```
/*IPEDS resident vs. NonResident*/
```

```
Case
When Citizen_Status_Code = 'C' OR Visa_type in ('PR','RF','TP') then 'Resident'
Else 'Non-resident'
END as IPEDS_Residency,
```

```
/*Representation flag - recoded in September 2020*/
```

```
CASE
WHEN SUBSTR(empl_unit_code,1,1) = 'A' THEN 'Non-represented'
WHEN (payroll_title_unit_rep_code = 'RACOV' AND empl_unit_code IN ('FX',' ',' ')) THEN
'Represented'
WHEN empl_unit_code IN ('99','87','??','FX',' ',' ') THEN 'Non-represented'
WHEN empl_unit_code IN ('M2','M3','M4','M6','M9','MO','BX','RA') THEN 'Represented'
WHEN SUBSTRN(payroll_title_unit_rep_code,LENGTH(payroll_title_unit_rep_code)-2,3) = 'UNC'
THEN 'Non-represented'
ELSE 'Represented'
END as Representation_flag,
```

```
/* Separation Flag by Separation Date, additional days on either end of separation window to account
for delayed recordings*/
```

```
Case
When Eff_Date_Fy = 'JUN2020' AND Separation_date BETWEEN '29JUN2019'd AND
'02JUL2020'd then 'Separated'
When Eff_Date_Fy = 'JUN2019' AND Separation_date BETWEEN '29JUN2018'd AND
'02JUL2019'd then 'Separated'
When Eff_Date_Fy = 'JUN2018' AND Separation_date BETWEEN '29JUN2017'd AND
'02JUL2018'd then 'Separated'
When Eff_Date_Fy = 'JUN2017' AND Separation_date BETWEEN '29JUN2016'd AND
'02JUL2017'd then 'Separated'
```

```
When Eff_Date_Fy = 'JUN2016' AND Separation_date BETWEEN '29JUN2015'd AND '02JUL2016'd then 'Separated'
When Eff_Date_Fy = 'JUN2015' AND Separation_date BETWEEN '29JUN2014'd AND '02JUL2015'd then 'Separated'
When Eff_Date_Fy = 'JUN2014' AND Separation_date BETWEEN '29JUN2013'd AND '02JUL2014'd then 'Separated'
When Eff_Date_Fy = 'JUN2013' AND Separation_date BETWEEN '29JUN2012'd AND '02JUL2013'd then 'Separated'
When Eff_Date_Fy = 'JUN2012' AND Separation_date BETWEEN '29JUN2011'd AND '02JUL2012'd then 'Separated'
When Eff_Date_Fy = 'JUN2011' AND Separation_date BETWEEN '29JUN2010'd AND '02JUL2011'd then 'Separated'
When Eff_Date_Fy = 'JUN2010' AND Separation_date BETWEEN '29JUN2009'd AND '02JUL2010'd then 'Separated'
When Eff_Date_Fy = 'JUN2009' AND Separation_date BETWEEN '29JUN2008'd AND '02JUL2009'd then 'Separated'
When Eff_Date_Fy = 'JUN2008' AND Separation_date BETWEEN '29JUN2007'd AND '02JUL2008'd then 'Separated'
When Eff_Date_Fy = 'JUN2007' AND Separation_date BETWEEN '29JUN2006'd AND '02JUL2007'd then 'Separated'
When Eff_Date_Fy = 'JUN2006' AND Separation_date BETWEEN '29JUN2005'd AND '02JUL2006'd then 'Separated'
When Eff_Date_Fy = 'JUN2005' AND Separation_date BETWEEN '29JUN2004'd AND '02JUL2005'd then 'Separated'
When Eff_Date_Fy = 'JUN2004' AND Separation_date BETWEEN '29JUN2003'd AND '02JUL2004'd then 'Separated'
When Eff_Date_Fy = 'JUN2003' AND Separation_date BETWEEN '29JUN2002'd AND '02JUL2003'd then 'Separated'
Else 'NA'
END as Separation_flag_by_SepDate,
```

```
/*Separation Flag by PPS reason*/
```

```
Case
When pps_separation_reason_name = '' then 'NA'
Else 'Separated'
END as Separation_flag_by_PPSreason,
```

```
/*Separation Flag by reason code*/
```

```
Case
When separation_rsn_code = '' then 'NA'
Else 'Separated'
END as Separation_flag_by_ReasonCode,
```

```
/* Separation Reason Names added, based on CPS data dictionary*/
```

```
Case separation_rsn_code
when '01' then 'Resignation (other than for health reasons)'
when '02' then 'Resignation (health reasons)'
```

```
when '03' then 'Medical separation (Staff Career, Limited and Casual/ Restricted, Librarians)'  
when '04' then 'Death'  
when '05' then 'Dismissal (Academic, Staff Career)'  
when '06' then 'Release (Staff Career, per diem nurses)'  
when '07' then 'Expiration of Appointment'  
when '08' then 'Layoff (Academic, Staff Career)'  
when '09' then 'Termination (Staff Limited and Casual/Restricted)'  
when '10' then 'Retirement/Change to Emeritus Status'  
when '11' then 'Faculty Early Retirement (Academic)'  
when '12' then 'Termination (coaches, management)'  
when '13' then 'Inactivity for four (4) consecutive months'  
when '14' then 'Resignation - accept job at newly created health corporation'  
when '15' then 'Layoff - accept job at newly created health corporation'  
when '16' then 'Inter-location Transfer with no break in service'  
else 'NA'  
END as separation_rsn_name,
```

/* New Hire Flag by Date of Most Recent Hire, additional days on either end of window to account for delayed recordings*/

Case

```
When Eff_Date_Fy = 'JUN2020' AND date_of_most_recent_hire BETWEEN '29JUN2019'd AND '02JUL2020'd then 'New Hire'  
When Eff_Date_Fy = 'JUN2019' AND date_of_most_recent_hire BETWEEN '29JUN2018'd AND '02JUL2019'd then 'New Hire'  
When Eff_Date_Fy = 'JUN2018' AND date_of_most_recent_hire BETWEEN '29JUN2017'd AND '02JUL2018'd then 'New Hire'  
When Eff_Date_Fy = 'JUN2017' AND date_of_most_recent_hire BETWEEN '29JUN2016'd AND '02JUL2017'd then 'New Hire'  
When Eff_Date_Fy = 'JUN2016' AND date_of_most_recent_hire BETWEEN '29JUN2015'd AND '02JUL2016'd then 'New Hire'  
When Eff_Date_Fy = 'JUN2015' AND date_of_most_recent_hire BETWEEN '29JUN2014'd AND '02JUL2015'd then 'New Hire'  
When Eff_Date_Fy = 'JUN2014' AND date_of_most_recent_hire BETWEEN '29JUN2013'd AND '02JUL2014'd then 'New Hire'  
When Eff_Date_Fy = 'JUN2013' AND date_of_most_recent_hire BETWEEN '29JUN2012'd AND '02JUL2013'd then 'New Hire'  
When Eff_Date_Fy = 'JUN2012' AND date_of_most_recent_hire BETWEEN '29JUN2011'd AND '02JUL2012'd then 'New Hire'  
When Eff_Date_Fy = 'JUN2011' AND date_of_most_recent_hire BETWEEN '29JUN2010'd AND '02JUL2011'd then 'New Hire'  
When Eff_Date_Fy = 'JUN2010' AND date_of_most_recent_hire BETWEEN '29JUN2009'd AND '02JUL2010'd then 'New Hire'  
When Eff_Date_Fy = 'JUN2009' AND date_of_most_recent_hire BETWEEN '29JUN2008'd AND '02JUL2009'd then 'New Hire'  
When Eff_Date_Fy = 'JUN2008' AND date_of_most_recent_hire BETWEEN '29JUN2007'd AND '02JUL2008'd then 'New Hire'
```

```
When Eff_Date_Fy = 'JUN2007' AND date_of_most_recent_hire BETWEEN '29JUN2006'd AND  
'02JUL2007'd then 'New Hire'  
When Eff_Date_Fy = 'JUN2006' AND date_of_most_recent_hire BETWEEN '29JUN2005'd AND  
'02JUL2006'd then 'New Hire'  
When Eff_Date_Fy = 'JUN2005' AND date_of_most_recent_hire BETWEEN '29JUN2004'd AND  
'02JUL2005'd then 'New Hire'  
When Eff_Date_Fy = 'JUN2004' AND date_of_most_recent_hire BETWEEN '29JUN2003'd AND  
'02JUL2004'd then 'New Hire'  
When Eff_Date_Fy = 'JUN2003' AND date_of_most_recent_hire BETWEEN '29JUN2002'd AND  
'02JUL2003'd then 'New Hire'  
Else 'NA'  
END as NewHire_flag_by_date
```

```
/*Source table name in CPS*/  
from PERS.PER_FY_ALL
```

```
/* Dates to gather data for. 'JUN 20**' represents the full fiscal year from July to June. Years filled in  
based on macro years*/  
where eff_date_fy = "JUN&yr"
```

```
/* List of "selected" fields above, minus the ones that are within a function (i.e. sum, count, etc.). Also  
include calculated fields that increase granularity*/
```

```
group by  
    eff_date_fy,  
    employee_number,  
    empl_emplid,  
    dos_code_cps,  
    dos_desc_cps,  
    pers_pgm,  
    Personnel_program,  
    location_1,  
    location_2,  
    location_3,  
    loc_indicator,  
    empl_bus_unit,  
    appt_type_code,  
    appt_type_code_pri,  
    academic_rank_code,  
    paid_over_code,  
    appt_basis_code,  
    function_code,  
    function_name,  
    fund_group_code,  
    fund,  
    org_1_code,  
    Org_1_Title,  
    org_2_code,
```

Org_2_Title,
org_3_code,
Org_3_Title,
SAU_Code,
UAS_Discipline_Code,
Home_Department_Name,
Home_Department_Code,
Class_Ttl_Outl_Code_Appt,
Class_Ttl_Outl_Name_Appt,
title_code,
title_name,
title_code_pri_appt,
primary_job_ind,
payroll_title_unit_rep_code,
empl_unit_code,
sex_code,
ethnic_origin_code,
IPEDS_hispanic_flag,
IPEDS_ethnic_origin_1_code,
IPEDS_ethnic_origin_1_name,
IPEDS_ethnic_origin_2_code,
IPEDS_ethnic_origin_3_code,
IPEDS_ethnic_origin_4_code,
IPEDS_ethnic_origin_5_code,
citizen_status_code,
visa_type,
pps_separation_reason_code,
pps_separation_reason_name,
separation_rsn_code,
separated_rsn_code_acad,
Separation_date,
date_of_most_recent_hire,
original_date_of_hire,
appt_pay_rate,
appt_rate_code,
unique_id;

quit;

*/*ends the do loop*/*

%end;

%mend YearCycle;

*/*Executes the macro*/*

%YearCycle;

```
/*Combines all the individual year tables into one table and drops duplicative or unneeded fields*/  
data FiscalYear_allyears (drop= paid_over_code paid_over_num appt_basis_code appt_basis_num  
sex_code);
```

```
    set FY_FTE_;
```

```
run;
```

```
quit;
```

```
/*Deletes single year files; Saves combined set file to new library for consumption. */
```

```
proc datasets library=work memtype=data nolist;
```

```
    delete FY_FTE_;
```

```
    copy in = work out = <new_library>;
```

```
    select FiscalYear_allyears;
```

```
run;
```

```
quit;
```