Applicant: Yizhou Yao

Position: Data and Policy Analyst - Statistical Programmer

SAS Code Sample

```
/* writing all outputs into a pdf. */
ods pdf file='C:\Users\Victor\Desktop\stsci5010\hw3\Yao_Yizhou_HW3_HTML.pdf';
/* problem 1 */

/*    Created a libref called hw3.
      Added a new column called SumExpenses that accumulated Expenses */
libname hw3 'C:\Users\Victor\Desktop\stsci5010\hw3';
data hw3.RunningSum;
      set hw3.expenses;
      SumExpenses + Expenses;
run;

title1 'problem 1';
title2 'RunningSum Dataset';
title3 'Added a new column called SumExpenses ';
title4 'that accumulated Expenses';
/* Display dataset of RunningSum. */
proc print data=hw3.runningsum;
run;
footnote '-- Produced by Yizhou Yao --';
/* Total expense in DEC 1999 is 8059191. */

/* problem 2 */

/* Sorted data by flightID. */
proc sort data = hw3.expenses out = expenses_sorted;
      by FlightID;
run;

/* Accumulated expenses for each flightID. */
data Sum_by_flight (drop = date expenses);
      set expenses_sorted;
      by FlightID;
      if first.flightID then Sum_by_flight = 0;
      Sum_by_flight + expenses;
      if last.flightID;
run;

title1 'problem 2';
title2 'Sum_by_flight Dataset';
title3 'Summed expenses for each individual flightID';
/* Display dataset of Sum_by_flight. */
proc print data=Sum_by_flight;
run;

/* problem 3 (a) */

/* without creating a dataset */
/* Calculate date values for January 1, 1987
   and September 1, 2015 and calculated weeks in between.
```

```
   And put them in the log. */
data _null_;
      date1 = mdy(1, 1, 87);
      put date1 =;
      date2 = mdy(9, 1, 15);
      put date2 =;
      weeks = intck('week', '01jan1983'd, date2);
      put weeks =;
run;

/* With creating a new dataset */
/* Calculate date values for January 1, 1987
   and September 1, 2015 and calculated weeks in between.*/
data hw3.WeeksBetween;
      date1 = 'Jan 1st, 1987';
      date1_value = mdy(1, 1, 87);
      date2 = 'Sep 1st, 2015';
      date2_value = mdy(9, 1, 15);
      weeksInBetween = intck('week', date1_value, date2_value);
run;
/*
Jan 1st, 1987 = 9862
Sep 1st, 2015 = 20332
weeks = 1705 i.e. There are 1705 weeks in between. */

/* problem 3 (b) */

/* without creating a dataset */
/* Caculate the date 107 weeks after October 15th, 2017 and
   put it in log. Added alignment = 's' so that
   the date of exacltly 107 weeks after will be calculated. */
data _null_;
      targetDate = intnx('week', '15oct2011'd, 107, 's');
      put targetDate = ;
      put targetDate = date9.;
run;

/* with creating a dataset */
/* save the data and use format date9. */
data hw3.weeksAfter2015;
      startDate = '15OCT2011';
      endDate = intnx('week', '15oct2011'd, 107, 's');
      format endDate date9.;
run;
/* Date value = 19664
   OR
   Date is November 2, 2013 */

/* problem 3 (c) */

/* Create a dataset that extracts
   the first and last names in Company dataset. */
data hw3.names;
      set hw3.company;
      lname = scan(name, 1);
      fname = scan(name, 2);
run;
```

```
title1 'problem 3 (c)';
title2 'Names dataset';
title3 'Display the first and last names';
title4 'that were extracted from Company dataset';
/* Display the dataset. */
proc print data = hw3.names;
run;

/* problem 3 (d) */

/* Create a dataset called ssn that
      replace the 4th and 5th ssn digit to 0. */
data hw3.ssn;
      set hw3.company;
      if ssn ne '' then
      substr(ssn, 5, 2) = '00';
run;

title1 'problem 3 (d)';
title2 'SSN dataset';
title3 'Display the SSN with middle digits replaced by 0';
/* Display the dataset. */
proc print data = hw3.ssn;
run;

/* problem 3 (e) */

/* Display the value returned by a SAS function in the form like
01JAN1964: 5 semiyears after January 1, 1983.
 results are in the log file. */
data _null_;
      semi = intnx('semiyear', '01jan1983'd, 5);
      format semi date9.;
      put semi ':5 semiyears after January 1, 1983.';
run;

/* problem 4 */

/* Use a DO loop to calculate the
      accumulated interests for a 30-year investment
      with annual interest rate equal to 8.8%.*/
data Invest (drop = monthInterest Interest c_prev);
      monthInterest = 0.088/12;
      c_prev = 0;
      do Year = 1 to 30;
            Year = Year;
            Capital = (8000 + c_prev) * (1 + monthInterest)**12;
            Interest = Capital - c_prev - 8000;
            Accumulated_interest + Interest;
            Accumulated_month = year * 12;
            c_prev = Capital;
            output;
      end;
run;

title1 'problem 4';
```

```
title2 'Invest dataset';
title3 'Use a DO loop to calculate the';
title4 'accumulated interests for a 30-year investment';
title5 'with annual interest rate equal to 8.8%.';
/* Display the dataset and supress the observations */
proc print data=Invest noobs;
run;

/* close the ods pdf. */
ods pdf close;

ods html close;
ods html;
/* 1.a create libraries*/
libname hw2 "C:\Users\Victor\Desktop\STSCI5010\hw2";
libname file xlsx 'C:\Users\Victor\Desktop\STSCI5010\hw2\FBPandHIV.xlsx';
data hw2.FBP_HIV;
      set file.Data;
      base_bmi = (preweight/(height*height))*10000;
      post_bmi = (postweight/(height*height))*10000;
      delta_bmi = post_bmi - base_bmi;
run;
libname file clear;
title 'Question1 a';
footnote'produced by Yizhou Yao';
proc print data=hw2.fbp_hiv;
run;
/* 1.b create user-defined formats */
libname library 'C:\Users\Victor\Desktop\STSCI5010\hw2';
proc format library=library;
      value karnf
            low-<25 = 'Sick 24 or less'
            25-<75 = 'Disabled 25-74'
            75-high = 'Healthy 75 or greater'
            other='unknown';
      value ynf
            1 = 'Yes'
            0 = 'No';
      value genderf
            1 = 'Male'
            0 = 'Female';
      value mybmif
            low-<18.5 = 'Underweight'
            18.5-<25 = 'Normal Wight'
            25-<30 = 'Overweight'
            30-high = 'Obese'
            other='Unknown';
run;

/* 1.c use formats*/
title 'Question1 c';
footnote'produced by Yizhou Yao';
proc print data=hw2.fbp_hiv (obs=28);
      format FBP ynf. gender genderf. arv ynf.
               prekarn karnf. postkarn karnf.
               preweight 7.2 postweight 7.2 height 7.2
               precd4 7.2 postcd4 7.2 delta_bmi 5.2
```

```
                base_bmi mybmif. post_bmi mybmif.;
run;

/* 2 produce frequency tables*/
/* a */
title 'Question2 a';
footnote'produced by Yizhou Yao';
proc freq data = hw2.fbp_hiv;
      tables fbp * base_bmi;
      format fbp ynf. base_bmi bmif.;
run;
/* b */
title 'Question2 b';
footnote'produced by Yizhou Yao';
proc freq data = hw2.fbp_hiv;
      tables fbp * post_bmi;
      format fbp ynf. post_bmi bmif.;
run;
/* c */
title 'Question2 c';
footnote'produced by Yizhou Yao';
proc freq data = hw2.fbp_hiv;
      tables arv * base_bmi;
      format arv ynf. base_bmi bmif.;
run;
/* d */
title 'Question2 d';
footnote'produced by Yizhou Yao';
proc freq data = hw2.fbp_hiv;
      tables gender * arv * base_bmi;
      format gender genderf. arv ynf. base_bmi bmif.;
run;

/* 3.a */
title 'Question3 a';
footnote'produced by Yizhou Yao';
proc summary data=hw2.fbp_hiv print maxdec=1;
      var precd4 postcd4;
      class fbp;
run;
/* 3.b produce median, qrange table*/
title 'Question3 b';
footnote'produced by Yizhou Yao';
proc means data=hw2.fbp_hiv
    qrange median maxdec=1;
      var prekarn postkarn precd4 postcd4;
run;

/* 3.c */
title 'Question3 c';
footnote1 'Since p-value is greater than 0.05 significance level,';
footnote2 'we cannot reject the null hypothesis and thus';
footnote3 'the difference of BMI is not significantly away from zero.';
proc means data=hw2.fbp_hiv t probt;
      var delta_bmi;
run;
/* Since p-value is greater than 0.05 significance level,
```

```
   we cannot reject the null hypothesis and thus
   the difference of BMI is not significantly away from zero. */

/* 4 */
/* read in excel file */
libname mydata xlsx 'C:\Users\Victor\Desktop\STSCI5010\hw2\Medical.xlsx';
data mydata.Nutrition;
      set hw2.Nutrition;
      if gender="M" then gender="F";
      else if gender="F" then gender="M";
run;


ods html close;
ods html;
/* practice One */
/* Created the libref called lab2 and filref called saledata. */
/* 1. */
libname lab2 'C:\Users\Victor\Desktop\stsci5010';
filename saledata 'C:\Users\Victor\Desktop\stsci5010\Sales.txt';

/* Test the program without reading in the observations. */
/* 2. */
data lab2.sales;
      infile saledata obs=0;
      input LastName 1-7 Month 9-11 Residential 13-21
      Commercial 23-31;
      Total=residential + commercial;
run;

/* There are errors because there are many Notes
   saying Invalid data for Lastname and Month. */
/* 0 records and 5 variables. */
/* In Input line, the dollar sign is missing for the
   LasteName and Month because they are character data
   and must be denoted by a dollar sign. */

/* Read all observations but does not create any data file. */
/* 3. */
data _null_;
      infile saledata;
      input LastName 1-7 Month 9-11 Residential 13-21
      Commercial 23-31;
      Total=residential + commercial;
run;
/* There are errors because there are many Notes
   saying Invalid data for Lastname and Month.
   Invalid data for Month and LastName.
   _ERROR_ = 1. */

/* Fixed the issue by adding $ to denote character data. */
/* 4. */
data _null_;
      infile saledata;
      input LastName $ 1-7 Month $ 9-11 Residential 13-21
      Commercial 23-31;
      Total=residential + commercial;
```

```
run;


/* 5. */
/* Created a dataset called Sales and saved it in lab2. */
data lab2.Sales;
      infile saledata;
      input LastName $ 1-7 Month $ 9-11 Residential 13-21
      Commercial 23-31;
      Total=residential + commercial;
run;


title1 'Practice One Problem 5';
title2 'Sales Dataset';
/* Print out the contents in lab2.Sales. */
proc print data=lab2.Sales;
run;
footnote 'Produced by Yizhou Yao';
/* 12 records and 5 variables. */

/* 6.A */
title1 'Practice One Problem 6.A';
title2 'Frequency Table Of Month';
/* print out the frequency table for month. */
proc freq data=lab2.Sales;
      tables month;
run;
footnote 'Produced by Yizhou Yao';

/* 6.B */
/* Create a dataset called Salesmonths in Lab2.
   Create a column called Type and set to incorrect
   if months is JAN, FEB, MAR. Also put an error message
   to the log if month is incorrect. */
data lab2.Salesmonths;
      set lab2.Sales;
      select(month);
            when('JAN', 'FEB', 'MAR')do;
                type = 'incorrect';
                  put _N_= month= type=;
                  put 'Data step'_N_ 'has an incorrect month: ' month=;
            end;
            when('AAA') type = 'correct';
      end;
run;

title1 'practice one problem 6.B';
title2 'Salesmonths Dataset';
proc print data=lab2.Salesmonths;
run;
footnote 'Produced by Yizhou Yao';

/* practice Two */
/* 1. */
/* Sort the empdata by location and save it in empdata_sorted. */
proc sort data=lab2.empdata out=lab2.empdata_sorted;
```

```
      by location;
run;

title1 'practice two problem 1';
title2 'empdata_sorted Dataset';
proc print data=lab2.empdata_sorted;
run;

/* 2. */
/* Calculate the total salary for each location. */
data lab2.Total_salary (keep=location total_salary);
      set lab2.empdata_sorted;
      by location;
      if first.location then total_salary = 0;
      total_salary + salary;
      if last.location;
run;

/* 3. */
/* Display Total_salary dataset */
title1 'practice two problem 3';
title2 'Total_salary Dataset';
proc print data=lab2.Total_salary noobs;
      sum total_salary;
      format total_salary dollar11.;
run;


/* Practie Three */
/* Created table1 using datalines */
data lab2.table1;
input Year 1-4 Var_X $ 6-7;
datalines;
1991 X1
1993 X3
1992 X2
1995 X5
1994 X4
;

/* Created table2 using datalines */
data lab2.table2;
input Year 1-4 Var_Y $ 6-7;
datalines;
1993 Y3
1991 Y1
1991 Y2
1994 Y4
1995 Y5
;

/* Sorted table1 by year */
proc sort data=lab2.table1;
      by year;
run;

/* Sorted table2 by year */
```

```
proc sort data=lab2.table2;
      by year;
run;

/* Merge table1 and table2 by year. */
data lab2.all;
      merge lab2.table1 lab2.table2;
      by year;
run;

/* Display All dataset */
title1 'Practice Three';
title2 'All';
proc print data=lab2.all;
run;

/* Practice Four */
/* A. */
/* sort demog by id */
proc sort data=lab2.demog;
      by id;
run;

/* sort visit by id */
proc sort data=lab2.visit;
      by id;
run;

/* merge demog and rename into all_matched and save it in lab2,
   without including unmatched records.
   rename date to BirthDate.
   put messages for each step.*/
data lab2.all_matched(keep = ID Sex BirthDate Visit Weight VisitDate);
      merge lab2.demog(in=indemog
                       rename=(date=BirthDate))
            lab2.visit(in=invisit
                       rename=(date=VisitDate));
      by id;
      if indemog=1 and invisit=1 then do;
            put _N_= indemog= invisit=;
            put ' Data step' _N_ 'has output to the target data set.';
      end;
      else do;
            put _N_= indemog= invisit=;
            put ' Data step' _N_ 'has not output to the target data set.';
      end;
      if indemog=1 and invisit=1;
run;

title1 'Practice Four Step A';
title2 'all_matched Dataset';
/* print out the data for all_mathced dataset */
proc print data=lab2.all_matched;
run;

/* B. */
/* craete heavy_female_patient only including women
```

```
   with weight greater than or equal to 250 pounds and
   save it in lab2. */
data lab2.heavy_female_patient;
      set lab2.all_matched;
      where sex='f' and weight>=250;
run;

/* Display heavy_female_patient dataset. */
title1 'Practice Four Step B';
title2 'heavy_female_patient Dataset';
proc print data=lab2.heavy_female_patient;
run;


/* Exercise 1 */
/* In this exercise, I assigned a new SAS library called Lab1,
   printed out the metadata of the Nutrition table,
   and printed out all the data on Nutrition table.*/
ODS HTML CLOSE;
ODS HTML;
options nonumber nodate;
libname Lab1 "C:\Users\Victor\Desktop\Lab1";
title 'metadata of nutrition table';
proc contents data=Lab1.nutrition;
run;
title;
/* Char variables: GENDER, VIT_A, VIT_B6, VIT_B12
   VIT_C, VIT_D, VIT_E, VIT_K. */
title 'nutrition table';
options pagesize=max linesize=max;
proc print data=Lab1.nutrition;
run;
title;
/* Abnormal feature: columns FOLATE and VIT_B2 have no values */
title 'rows 10 to 20 of nutrition table';
proc print data=Lab1.nutrition(firstobs=10 obs=20);
run;
title;

/* Exercise 2 */
/* In this exercise, I created a new table called males3000kcal
   from Nutrition table by selecting the rows that meet the condition
   using where clause.
   I sorted the table by calories and printed out the top 15 records
   with title and footnote. */
options pagesize=30 linesize=100;
data Lab1.males3000kcal;
      set Lab1.nutrition;
      where kcal>=3000 AND gender="M";
run;
proc sort data=Lab1.males3000kcal out=work.sortedM3000;
      by descending kcal;
run;
title1 'Males with calories intake no less than 3000';
title2 'sorted in descending order';
footnote 'Data from Nutrition table ';
proc print data=work.sortedM3000(obs=15);
```

```
       var GENDER KCAL KCAL_FAT KCAL_CHO KCAL_PRO;
run;
title;
footnote;

/* Exercise 3 */
/* In this exercise, I sorted the table in descending order by iron and
   then by fiber. I printed out the subset of table by selecting the rows
   that meet the condition using where clause with title and footnote.*/
proc sort data=lab1.nutrition out=work.sorted_IRON_FIBER;
      by descending iron descending fiber;
run;
title 'Nutrition Table Sorted By Descending Iron and Fiber';
footnote 'Data from sorted_Iron_Fiber';
proc print data=work.sorted_iron_fiber;
      where gender='F' AND iron<4 AND fiber<4;
      var  GENDER KCAL VIT_A VIT_D FIBER IRON PROTEIN;
run;
title;
footnote;
/* 2 women met the criteria and are included in my report. */

/* Exercise 4*/
/* In this exercise, I selected rows that meet condition by using
   where clause and formatted the SODIUM column. I printed out the
   resulting table with title and footnote. */
options pagesize=50 linesize=80;
title 'Males with iron greater than 20 and fat greater than 120';
footnote 'Data from sorted_iron_fiber';
proc print data=work.sorted_iron_fiber;
      where gender='M' AND iron>20 AND fat>120;
      var  IRON FIBER GENDER PROTEIN SODIUM;
      format sodium comma8.2;
run;
title;
footnote;


/* 1 */
ODS HTML CLOSE;
ODS HTML;
/* assign a path to a libref called hw1. */
libname hw1 'C:\Users\Victor\Desktop\stsci5010\hw1';
/* Import the .txt data using datalines into a SAS table,
   and save it as activity in hw1 library. */
data hw1.activity;
      input ID $ Name $ Sex $ Age Date Height Weight ActLevel $ Fee;
      datalines;
      2458 Murray M 27 1 72 168 HIGH 85.24
      2462 Almers F 34 3 66 152 HIGH 124.85
      2501 Bonavent F 31 17 61 123 LOW 155.77
      2523 Johnson F 43 31 63 137 MOD 149.75
      2539 LaMance M 71 4 71 158 LOW 124.86
      2544 Jones M 29 6 76 193 HIGH 124.89
      2552 Reberson F 32 9 67 151 MOD 149.75
      2555 King M 35 13 70 173 MOD 199.75
      2563 Pitts M 65 22 73 154 LOW 124.88
```

```
      2568 Eberhard F 49 27 64 172 LOW 124.81
      2571 Nunnelly F 44 19 66 140 HIGH 149.75
      2572 Oberon F 28 17 62 118 LOW 85.26
      2574 Peterson M 30 6 69 147 MOD 149.75
      2575 Quigley F 40 8 69 163 HIGH 124.83
      2578 Cameron M 47 5 72 173 MOD 124.84
      2579 Underwoo M 60 22 71 191 LOW 180.18
      2584 Takahash F 43 29 65 123 MOD 124.82
      2586 Derber M 25 23 75 188 HIGH 85.26
      2588 Ivan M 66 20 63 139 LOW 85.27
      2589 Wilcox F 41 16 67 141 HIGH 149.75
      2595 Warren M 54 7 71 183 MOD 165.75
;
/* print out the summary table of the data table. */
title 'problem 1';
footnote 'Produced by Yizhou Yao';
options ps=50;
proc contents data=hw1.activity;
run;
title;
footnote;
/* As shown, there are 21 observations and 9 variables and the data
   are all loaded compared to the original text file. */

/* 2 */
/* Create a new temporary data set by selecting
   rows whose activity level is HIGH. */
data work.al_high;
      set hw1.activity;
      where actlevel='HIGH';
run;
/* Create a new temporary data set by selecting
   rows whose activity level is MOD. */
data work.al_mod;
      set hw1.activity;
      where actlevel='MOD';
run;
/* Create a new temporary data set by selecting
   rows whose activity level is LOW. */
data work.al_low;
      set hw1.activity;
      where actlevel='LOW';
run;

/* print out the data in al_high table . */
title 'problem 2';
title2 'people with HIGH activity level';
footnote 'Produced by Yizhou Yao';
options ps=18;
proc print data=al_high;
run;
title;
title2;
footnote;
/* print out the data in al_mod table . */
title 'problem 2';
title2 'people with MOD activity level';
```

```
footnote 'Produced by Yizhou Yao';
proc print data=al_mod;
run;
title;
title2;
footnote;
/* print out the data in al_low table . */
title 'problem 2';
title2 'people with LOW activity level';
footnote 'Produced by Yizhou Yao';
proc print data=al_low;
run;
title;
title2;
footnote;
/* Since WORK library is temporary and the SAS session has been
   terminated, we cannot find those three files again. */

/* 3 */
/* a */
/* print out the data with actlevel is high or mod AND
   with fee between 100 and 130.*/
title 'problem 3 (a)';
title2 'people with HIGH or MOD activity level';
title3 'and activity level between 100 and 130';
footnote 'Produced by Yizhou Yao';
proc print data=hw1.activity;
      where (actlevel='HIGH' or actlevel='MOD') and
             (fee <= 130 and fee>=100);
run;
title;
title2;
title3;

/* b */
/* print out the data with name containing an 'o' and 'n'. */
title 'problem 3 (b)';
title2 'people whose name contains o and n';
footnote 'Produced by Yizhou Yao';
proc print data=hw1.activity;
      var  ID name sex age;
      where name ? 'o' and name ? 'n';
run;
title;
title2;
footnote;

/* c */
/* print out the data who are female and fee is greater than 100. */
title 'problem 3 (c)';
title2 'female whose fee is greater than 100';
footnote 'Produced by Yizhou Yao';
proc print data=hw1.activity label;
      id ID;
      where sex='F' and fee>100;
      format fee dollar7.2;
      label actlevel='Activity Level';
```

```
run;
title;
title2;
footnote;

/* 4 */
/* create and save a new data set called oldmale into
   hw1, by selecting male with age over 65 and setting
   the format and label. */
data hw1.OldMale;
      set hw1.activity;
      where sex='M' and age>=65;
      label fee='Fee charged at the time of admission ($)';
      format fee dollar7.2;
run;

/* print out the data with pre-saved label. */
title 'problem 4';
title2 'male who are at least 65';
footnote 'Produced by Yizhou Yao';
proc print data=hw1.oldmale label;
run;
title;
title2;
footnote;

/* overwrite the previously saved label and print
   out the new table. */
title 'problem 4';
title2 'male who are at least 65 with updated column name';
footnote 'Produced by Yizhou Yao';
proc print data=hw1.oldmale label;
      label fee = 'Admission Fee';
      format fee dollar6.1;
run;
title;
title2;
footnote;
```

SQL Code Sample:

```
/* Fall 2020 STSCI 5060 Final Project */

/* Name: Yizhou Yao */

/* NetID: yy856 */


/* set the pagesize and linesize */

set linesize 5000

set pagesize 1000


/* clear up all tables/views after each session. */
```

```
drop table fedrev_t;

drop table strev_t;

drop table locrev_t;

drop table school_t;

drop view sd#_v;

drop view mfr_v;

drop view msr_v;

drop view mlr_v;

drop view total_rev_v;

drop view fed_contribution_v;

drop view st_contribution_v;

drop view loc_contribution_v;

drop view fsl_contribution_v;

drop table state_t cascade constraints;


ttitle '******** Step 3 ********' skip 2

/* update the state_t table by changing the single-digit values, 1-9, of
state code to two-digit values, 01-09.  */

update state_t

    set stcode='0'||substr(stcode,1,1)

        where cast(stcode as int)<10;


/* display the 9 rows whose Stcode values are less than 10 */

select * from state_t where cast(stcode as int)<10;


ttitle '******** Step 4 ********' skip 2


/********************** please note ***********************************

Because of my computer setting, the numeric data in sql was automatically set

to BINARY_DOUBLE instead of NUMBER. I consulted professor Yang about this

and he said it was OK and CC'ed the grader about this situation.If you have

any additional question please do not hesitate to let me
```

```
and I'm more than willing to provide more info.
Thank you very much for your understanding.
**********************************************************************/

/* see the metadata about school_finance_2010_t table. */
describe school_finance_2010_t;

/* display the top 10 rows of school_finance_2010_t. */
select * from school_finance_2010_t
where rownum <= 10;

ttitle '******** Step 5 ********' skip 2
/* change idcesus's datatype to varchar2(15) */
alter table school_finance_2010_t
    modify idcensus varchar2(15);

/* change name's datatype to varchar2(60) */
alter table school_finance_2010_t
    modify name varchar2(60);

ttitle '******** Step 6 ********' skip 2
/* rename name to SD_NAME */
alter table school_finance_2010_t
rename column name to SD_NAME;

/* rename state to stcode */
alter table school_finance_2010_t
rename column state to STCODE;

ttitle '******** Step 7 ********' skip 2
/* create fedrev_t table by summing up some columns */
create table fedrev_t as
```

```
select idcensus, stcode,
(c14+c15+c16+c17+c18+c19+b11+c20+c25+c36+b10+b12+b13) as fed_rev

from school_finance_2010_t;


/* create strev_t table by summing up some columns */

create table Strev_t as

select idcensus, stcode,
(c01+c04+c05+c06+c07+c08+c09+c10+c11+c12+c13+c24+c35+c38+c39) as st_rev

from school_finance_2010_t;


/* create locrev_t table by summing up some columns */

create table Locrev_t as

select idcensus, stcode,
(t02+t06+t09+t15+t40+t99+d11+d23+a07+a08+a09+a11+a13+a15+a20+a40+u11+u22+u30+
u50+u97) as loc_rev

from school_finance_2010_t;


/* create school_t from school_finance_2010_t */

create table school_t as

select idcensus, stcode, sd_name

from school_finance_2010_t;


ttitle '******** Step 8.A ********' skip 2

/* Set the stcode column as the primary key of the State_t table */

alter table state_t

add constraint stcode_pk primary key (stcode);


ttitle '******** Step 8.B ********' skip 2

/* Set the idcensus column in the Fedrev_t as the primary key. */

alter table fedrev_t

add constraint idcensus_PK primary key(idcensus);


/* Set the idcensus column in the strev_t as the primary key. */

alter table Strev_t
```

```
add constraint idcensus_PK2 primary key(idcensus);

/* Set the idcensus column in the school_t as the primary key. */
alter table school_t
add constraint idcensus_PK3 primary key(idcensus);

/* Set the idcensus column in the locrev_t as the primary key. */
alter table Locrev_t
add constraint idcensus_PK4 primary key(idcensus);

ttitle '******** Step 8.C ********' skip 2
/* Set the idcensus column of the Fedrev_t as the foreign key that
references the idcensus column of the School_t table. */
alter table fedrev_t
add constraint idcensus_fk foreign key (idcensus) references
school_t(idcensus);
/* Set the idcensus column of the strev_t as the foreign key that
references the idcensus column of the School_t table. */
alter table Strev_t
add constraint idcensus_fk2 foreign key (idcensus) references
school_t(idcensus);
/* Set the idcensus column of the locrev_t as the foreign key that
references the idcensus column of the School_t table. */
alter table Locrev_t
add constraint idcensus_fk3 foreign key (idcensus) references
school_t(idcensus);

ttitle '******** Step 8.D ********' skip 2
/* Set the stcode column of the School_t table as the foreign key that
references the stcode
column of the State_t table. */
alter table school_t
add constraint stcode_fk foreign key (stcode) references state_t(stcode);
```

```
ttitle '******** Step 10 ********' skip 2

/* display idcensus, stcode and fed_revenue of school districts with more
than

1000000k funds. */

select idcensus, stcode, to_char(fed_rev,'999999999.9') as fed_revenue from
fedrev_t

where fed_rev > 1000000;


/* display idcensus, stcode and st_revenue of school districts with more than

1000000k funds. */

select idcensus, stcode, to_char(st_rev,'999999999.9') as st_revenue from
strev_t

where st_rev > 1000000;


/* display idcensus, stcode and loc_revenue of school districts with more
than

1000000k funds. */

select idcensus, stcode, to_char(loc_rev,'999999999.9') as loc_revenue from
locrev_t

where loc_rev > 1000000;


ttitle '******** Step 11.A ********' skip 2

/* find the state(s) that with the lowest number of school districts by using
sd#_v. List the

state code, state name and the total number of school districts. */

create view sd#_v as select count(stcode) as SD#,stcode from  school_t

group by stcode;


select v.stcode, stname, sd#

from sd#_v v inner join state_t t on v.stcode=t.stcode

where sd# = (select min(sd#) from sd#_v);


ttitle '******** Step 12.A ********' skip 2

/* create three views in Oracle called mfr_v,
```

```
msr_v, and mlr_v to calculate the maximum federal, state, and local revenues
in each

state. */

create or replace view mfr_v as select stcode,max(fed_rev) as MAX_FED_REV
from fedrev_t

group by stcode

order by stcode;

create or replace view msr_v as select stcode,max(st_rev) as MAX_ST_REV from
strev_t

group by stcode

order by stcode;

create or replace view mlr_v as select stcode,max(loc_rev)as MAX_loc_REV from
locrev_t

group by stcode

order by stcode;


ttitle '******** Step 12C ********' skip 2

/* use the mfslr_t table created by above SAS DATA Step to get the results.
*/

select to_char(m.stcode,'99') as stcode,to_char(max_fed_rev, '999999999.9')

as max_fed_rev, to_char(max_st_rev,'999999999.9') as max_st_rev,

to_char(max_loc_rev,'999999999.9') as max_loc_rev, stname as state_name from
mfslr_t m,state_t s

where m.stcode=s.stcode;


ttitle '******** Step 13 ********' skip 2

/* list the state code and the highest federal revenue (use aliases,
state_code for state code, state_name for stname, and

max_fed_rev for the highest total federal revenue of the school district in
that state */

select to_char(m.stcode,'999999999')as state_code, stname as state_name,
to_char(max_fed_rev, '999999999.9') as max_fed_rev, sd_name

from school_t sc, mfslr_t m, fedrev_t f,state_t st

where f.idcensus=sc.idcensus and

      m.stcode=sc.stcode and

      sc.stcode=f.stcode and
```

```
      st.stcode=sc.stcode and
      m.max_fed_rev=f.fed_rev
order by max_fed_rev desc;

ttitle '******** Step 14 ********' skip 2
/* Create a view called Total_Rev_v from fedrev_t, strev_t, and locrev_t by
including idcensus,
state code, total federal revenue (named tfedrev), total state revenue (named
tstrev), and total
local revenue (named tlocrev) of each school district. */
create or replace view total_rev_v as
select f.idcensus, f.stcode, fed_rev as tfedrev, st_rev as tstrev, loc_rev as
tlocrev
from fedrev_t f, strev_t s, locrev_t l
where f.idcensus=s.idcensus and
      s.idcensus=l.idcensus;

ttitle '******** Step 15 ********' skip 2
/* display the top 100 columns in the order of stcode, stname, idcensus,
total_revenue
and sd_name, in descending order */
select * from
(select t.stcode, stname, t.idcensus,
to_char((tfedrev+tstrev+tlocrev),'999999999.9') as total_revenue, sd_name
from total_rev_v t, school_t s, state_t st
where t.stcode=s.stcode and
      st.stcode=s.stcode and
      t.idcensus=s.idcensus
order by total_revenue desc)
where rownum<=100;

ttitle '******** Step 16 ********' skip 2
/* display stcode, stname, and the total school expenditure of
the state. Sort output with the total school expenditure in descending order.
```

```
 */
select sc.stcode, stname, to_char(sum(totalexp),'999999999.9') as totalexp_st
from school_finance_2010_t sc, state_t st
where sc.stcode = st.stcode
group by sc.stcode, stname
order by sum(totalexp) desc;

ttitle '******** Step 17 ********' skip 2
/* display the total amount of the money that the United State spent on the
public school
systems in 2010 */
set heading off
select 'The total amount that the United States spent on the public school
systems in 2010 was',to_char(sum(totalexp),'$999999999.9'), 'K.'
from school_finance_2010_t;
set heading on

ttitle '******** Step 18.A ********' skip 2
/* Find out school districts that received federal revenues greater than
the total expense, listing all the columns that exist in the
fed_contribution_v and sorting in
descending order by fed_pcnt. */
create or replace view fed_contribution_v as
select f.idcensus, f.stcode, stname, sd_name,
to_char((fed_rev/totalexp),'9.9999')as fed_pcnt
from fedrev_t f,school_finance_2010_t s, state_t st
where s.idcensus=f.idcensus and
      f.stcode=s.stcode and
      s.stcode=st.stcode and
      totalexp is not null and
      totalexp <> 0;

select * from fed_contribution_v where fed_pcnt > 1 order by fed_pcnt desc;
```

```
ttitle '******** Step 18.B ********' skip 2

/* Find out school districts that received state revenues greater than

the total expense, listing all the columns that exist in the
st_contribution_v and sorting in

descending order by st_pcnt. */

create or replace view st_contribution_v as

select sr.idcensus, sr.stcode, stname, sd_name,
to_char((st_rev/totalexp),'9.9999')as st_pcnt

from strev_t sr, school_finance_2010_t s, state_t st

where sr.idcensus=s.idcensus and

      s.stcode=sr.stcode and

      sr.stcode=st.stcode and

      totalexp is not null and

      totalexp <> 0;

select * from st_contribution_v where st_pcnt > 1 order by st_pcnt desc ;


ttitle '******** Step 18C ********' skip 2

/* Find out school districts that received local revenues greater than

the total expense, listing all the columns that exist in the
loc_contribution_v and sorting in

descending order by loc_pcnt. */

create or replace view loc_contribution_v as select l.idcensus, l.stcode,
stname, sd_name, to_char((loc_rev/totalexp),'99.9999') as loc_pcnt

from state_t st,locrev_t l, school_finance_2010_t sf

where l.idcensus=sf.idcensus and

      l.stcode=sf.stcode and

      sf.stcode=st.stcode and

      totalexp is not null and

      totalexp <> 0;

select * from loc_contribution_v where loc_pcnt > 1 order by loc_pcnt desc;


ttitle '******** Step 19.A ********' skip 2

/* create another view called
```

```
fsl_contribution_v, including these columns: idcensus, stcode, sd_name and
the fsl_pcnt (for the

total ratio, which is the sum of fed_pcnt, st_pcnt and loc_pcnt). Keep 4
decimal points.  */

create or replace view fsl_contribution_v as

select f.idcensus, f.stcode, f.sd_name, to_char((fed_pcnt+st_pcnt+loc_pcnt),
'99.9999') as fsl_pcnt

from fed_contribution_v f, st_contribution_v s, loc_contribution_v l

where f.idcensus = s.idcensus

      and s.idcensus = l.idcensus;

/* display the school districts that received total revenues
(federal+state+local) over 3

times of the total amount they actually spent in that year, in descending
order */

select * from fsl_contribution_v where fsl_pcnt > 3 order by fsl_pcnt desc;


/* display the school districts that received total revenues
(federal+state+local) up to 30%

of the total amount they actually spent in that year, in descending order */

ttitle '******** Step 19.B ********' skip 2

select idcensus, stcode, sd_name, to_char(fsl_pcnt,'90.9999')as fsl_pcnt from
fsl_contribution_v where fsl_pcnt<=0.3 order by fsl_pcnt desc;



ttitle '******** Step 25.A ********' skip 2

/* Change the table definitions to make sure they can be joined */

alter table school_finance_2015_t

modify idcensus varchar2(15);

alter table school_finance_2015_t

modify name varchar2(60);


/* display top 5 school districts that had increased total revenues. */

select stcode, stname, idcensus, sd_name, to_char(revdif,'99999999.9') as
revdif, to_char(change_percentage,'999999999999999.9') as change_percentage

from
```

```
(select s2.state as stcode, stname, s2.idcensus, s2.name as sd_name,
(s2.totalrev-s1.totalrev) as revdif,

(100*(s2.totalrev-s1.totalrev)/s1.totalrev) as change_percentage

from school_finance_2010_t s1, school_finance_2015_t s2, state_t s3

where s2.state=s1.stcode

    and s2.state=s3.stcode

    and s1.idcensus=s2.idcensus

    and s1.totalrev <> 0

    order by revdif desc)

where rownum<=5;



ttitle '******** Step 25.B ********' skip 2

/* display top 5 school districts that had decreased total revenues. */

select stcode, stname, idcensus, sd_name, to_char(revdif,'99999999.9') as
revdif, to_char(change_percentage,'99999999999999.9') as change_percentage

from

(select s2.state as stcode, stname, s2.idcensus, s2.name as sd_name,
(s2.totalrev-s1.totalrev) as revdif,

(100*(s2.totalrev-s1.totalrev)/s1.totalrev) as change_percentage

from school_finance_2010_t s1, school_finance_2015_t s2, state_t s3

where s2.state=s1.stcode

    and s2.state=s3.stcode

    and s1.idcensus=s2.idcensus

    and s1.totalrev <>0

    order by revdif)

where rownum<=5;


ttitle '******** Step 25.C ********' skip 2

/* display all school districts whose total revenues stayed the same. */

select stcode, stname, idcensus, sd_name, to_char(revdif,'99999999.9') as
revdif, to_char(change_percentage,'99999999999999.9') as change_percentage

from
```

```
(select s2.state as stcode, stname, s2.idcensus, s2.name as sd_name,
(s2.totalrev-s1.totalrev) as revdif,

((s2.totalrev-s1.totalrev)/s1.totalrev) as change_percentage

from school_finance_2010_t s1, school_finance_2015_t s2, state_t s3

where s2.state=s1.stcode

    and s2.state=s3.stcode

    and s1.idcensus=s2.idcensus

    and s1.totalrev <> 0)

where revdif=0

order by revdif;
```

Python Code Sample: