STSCI 5060 Final Project

(Fall 2023)

Important: This project must be submitted by 4:30 PM, December 10, 2023, and any overdue project will not be accepted. It is an individual project that is completed by only one student. Cornell's Code of Academic Integrity is strictly enforced. Do your own work! Please carefully read the following General Instructions before you start.

General Instructions

1. Coding:
   A. At the very beginning, start your Oracle and SAS code with the following comment block.
      /* Fall 2023 STSCI 5060 Final Project */
      /* Name: */
      /*NetID: */
   B. Use ORACLE SQL DEVELOPER for SQL coding unless otherwise specified.
   C. If a task can be done with Oracle SQL*PLUS or with SAS PROC SQL, use Oracle SQL*PLUS. You use SAS PROC SQL only when you are explicitly told to do so.
   D. Always leave a blank line between any two blocks of code.
   E. When you code in Oracle, start each project step with a title like:
      i. ttitle '******** Step # ********' skip 2 (# is the step number).
   F. When you code in SAS, start your project step with a title like:
      i. title '******** Step # ********'; (# is the step number).
      If this guideline is not followed, up to 5 points will be taken off.
   G. Set your line size (e.g., 5000) and page size (e.g., 1000) properly so that you produce well-organized outputs, e.g., your Oracle output should not wrap to next line(s). If your output is messy without following this guideline, up to 5 extra points will be taken off.
   H. Insert a brief comment at the beginning of a code block to briefly explain what the code does. Up to 5 extra points will be taken off if these comments are missing.

2. Submission:
   A. Your Oracle code/script, named LastName_FirstName_Oracle_STSCI5060FP.sql (Note: use your real name to replace LastName_FirstName, the same below).
   B. Your SAS code, named LastName_FirstName_SAS_STSCI5060FP.sas.
   C. All the Oracle script outputs, i.e., your Oracle query results, named LastName_FirstName_SQL_output.txt. You may produce this file in a new, single run after all your code works correctly, but you will need to first drop some tables, e.g., State_t, School_Finance_2010_t, School_Finance_2015_t, and School_t. (hint: drop table state_t CASCADE CONSTRAINTS;) and then create them again when you run your SQL script.
   D. Your SAS log (LastName_FirstName_SAS_log.log).
   E. An MS Word (or PDF) file containing all your screenshots, code entered in the Oracle command line interface, the ERD from Step 9, SAS analysis results, and answers to the questions (LastName_FirstName_FP_Report.docx or LastName_FirstName_FP_Report.PDF).

F. At the end, compress all the above files with 7-ZIP (or a similar tool) into one final file named LastName_FirstName_STSCI5060FP.zip and submit it to the course website.

The Project Background and Steps

This project utilizes much knowledge and programming skills (both SQL and SAS) covered in the class. You will create a database in Oracle, based on two big SAS datasets called *School_Finance_2010. sas7bdat* and *School_Finance_2015. sas7bdat*, and then use this database to analyze the data with sql*plus and SAS. The *School_Finance_201X.sas7bdat* data files contain real data about the finance of US public schools in the years of 2010 and 2015. For our international students who are not familiar with the US school systems, here is some short explanation. In each state, the public schools are divided into different school districts. A school district normally includes many different levels of schools (elementary schools, middle schools, and high schools in a city/town/area). A school district is normally funded by three sources: the local revenue, the state revenue and the federal revenue. The datasets describe these revenues and expenditures. You need to read *Column Description.PDF* so that you understand the datasets and the meaning of column names. Please strictly follow the following steps as you do the project and write your project report.

1. In Oracle command line, log in as the Oracle SYSTEM user and create a new user called **LastName_FirstName_STSCI5060FP** and grant all the privileges to this user. If you encounter an error of invalid name, you may need to issue an "ALTER SESSION" command (see Lab 1 for details). Then, in ORACLE SQL DEVELOPER create a new connection to this new user with the same name, LastName_FirstName_STSCI5060FP.

2. In SQL DEVELOPER, create a table, **State_t**, by importing a csv file called State_Code.csv. To import, right click "Tables (filtered)" and choose "Import Data …" and then follow the instructions. In Step 3, make sure that you only select the three required columns (StCode, StName and StAbb), and then choose Column Definition to continue. In the Column Definition window (in the 4th Step of Data Import Wizard), you need to delete the trailing blank(s) for all the variable names that are listed in the Source Data Columns window, in order to have valid column names. Use the default setting when you import the table, except setting the attributes of columns StCode, StName and StAbb to varchar2(2), varchar2(26) and char(2) respectively.

   Attach the following screenshots of ORACLE SQL DEVELOPER: the **Columns** tab and **Data** tab of your State_T table (just showing a portion of the data of the table is OK).

3. In Oracle, programmatically (i.e., you have to code it with SQL) update your State_t table by changing the single-digit values, 1-9, of state code to two-digit values, 01-09. This will make sure that these values are consistent with those in other tables. In this step, you are required to accomplish the task just with one update statement with the following requirements:
   A. Use a WHERE clause to specify a condition.
   B. Use the concatenation operator.
   C. Use the SUBSTR() function.

D. Use the CAST function to explicitly convert the data type for your comparison in the WHERE clause (although in this case Oracle can do data type automatic conversion without using this function).

Query your updated State_t table to confirm that you have successfully made the changes. Only display the 9 rows whose Stcode values are less than 10.

4. In SAS 9.4, create a libref called "**Final**" that references your file location of all your final project SAS files, and then connect SAS to the Oracle database user, "LastName_FirstName_STSCI5060FP," with the LIBNAME statement by creating a libref called **myoracle.** Create an Oracle database table, **School_Finance_2010_t**, using the myoracle libref and PROC SQL by querying the School_Finance_2010.sas7bdat dataset (which is in SAS format).

   Right after you ran your SAS code, refresh the connections in your ORACLE SQL Developer. In Oracle, describe the **School_ Finance_2010_t** table and display the first 10 rows of the table (hint: use rownum<=10).

5. Change table properties of the School_Finance_2010_t table:
   A. Change the property of IDCENSUS to varchar2(15).
   B. Change the property of NAME to varchar2(60).

6. Change column names: rename the column "NAME" to "SD_NAME" and the column "State" to "STCODE" in the School_Finance_2010_t table.

7. The School_Finance_2010_t table is big table with many functional dependencies. You will create four tables from this big table. The first three tables are Fedrev_t, Strev_t and Locrev_t for the federal, state and local revenues respectively. The fourth table is called School_t.
   A. Your Fedrev_t table should include the following columns: idcensus, stcode, and fed_rev, which is the sum of the values of columns c14, c15, c16, c17, c18, c19, b11, c20, c25, c36, b10, b12, and b13.
   B. Your Strev_t table should include the following columns: idcensus, stcode, and st_rev, which is the sum of the values of columns c01, c04, c05, c06, c07, c08, c09, c10, c11, c12, c13, c24, c35, c38, and c39.
   C. Your Locrev_t table should include the following columns: idcensus, stcode, and loc_rev, which is the sum of the values of columns t02, t06, t09, t15, t40, t99, d11, d23, a07, a08, a09, a11, a13, a15, a20, a40, u11, u22, u30, u50, and u97.
   D. Your School_t table should include these columns: idcensus, stcode, and sd_name.

8. Programmatically set primary keys and foreign keys for the following tables:
   A. Set the stcode column as the primary key of the State_t table.
   B. Set the idcensus column in the Fedrev_t, Strev_t, Locrev_t, and School_t tables as the primary key.

C. Set the idcensus column of the Fedrev_t, Strev_t, Locrev_t tables as the foreign key that references the idcensus column of the School_t table.
D. Set the stcode column of the School_t table as the foreign key that references the stcode column of the State_t table.

[Hint: To set the foreign key, use the following syntax: **ALTER TABLE** *table_name* **add constraint** *foreign_key_name* **foreign KEY** (column_name) **references** *name_of_the_table_referenced* (*name_of_the_column_referenced*);]

9

9. Each of the table you have created is correspondent to an entity. As a result, you have five entities: STATE, SCHOOL, FEDREV, STREV and LOCREV. Draw an ERD (using the book's HRT notation) to represent the relationships of these entities. You are required to list all the attributes. Then, convert your ERD to relations and use arrows to represent the primary key/foreign key constraints. If you draw the diagrams with hand, you may scan it and then insert it into the MS Word (or PDF) file.

4.5

10. Based on Fedrev_t, list all the school districts that received more than $1,000,000 K from the federal source (note that the revenue values are expressed in thousands of dollars in the database tables). You should display three columns: idcensus, stcode and an alias, fed_revenue for the federal revenue. Do the same for the Strev_t and Locrev_t tables and find out all the school districts that received more than $1,000,000 K from the state or local sources. You should name the state and local total revenue aliases as st_revenue and loc_revenue. The revenue values you display should not be in scientific notation (**do the same for the rest of the project**). Hint: use the function **to_char(value, '999999999.9')** to achieve this; the numbers of the digit "9" represent the scale and precision of the number.

4

11. Create a view called **sd#_v** to calculate the total number of school districts (SD#) in each state. This view has two columns, SD# and stcode. Then,
    A. find the state(s) that with the highest number of school districts by using sd#_v. In your output, list the state code, state name and the total number of school districts.
    B. 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.

8

12. In each state, among all the school districts, find out the highest local, state and federal revenues, and sort your result by state. Your final output should be listed in the following column order and format:

STCODE	MAX_FED_REV	MAX_ST_REV	MAX_LOC_REV	STATE_NAME
1	xxxxx.0	xxxxx.0	xxxxx.0	Alabama
2	xxxxx.0	xxxxx.0	xxxxx.0	Alaska
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

To achieve this, you need to do the following:

A. Based on the tables you created in Step 7, 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. Each view should only contain two columns: stcode and the value of the maximum revenue, sorted by stcode. Name the maximum revenues as MAX_FED_REV, MAX_ST_REV and MAX_LOC_REV, respectively.

B. Connect SAS 9.4 to Oracle using the libname approach. In a SAS DATA step, use an appropriate dataset combining method to combine the data in the above three views to directly create a table called **mfslr_t** in Oracle. This table should contain four columns: stcode, MAX_FED_REV, MAX_ST_REV and MAX_LOC_REV.

C. Go back to Oracle and use the mfslr_t table created by above SAS DATA Step to get your results. You should space your four columns as shown above, i.e., do not leave too much white space between the columns (hint: use the to_char() function to achieve this).

13. This is an extension of above question. In addition to listing the state name, 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), you are required to list the name of the school district that received the highest federal revenue in that state, as the 4th column. Sort your result by the revenue in descending order.

14. 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.

15. From Total_Rev_v, calculate the total revenues (use an alias, total_revenue) of these three sources (tfedrev, tstrev, and tlocrev) for each school district in the US. With the information from other tables, display the columns in the order of stcode, stname, idcensus, total_revenue and sd_name. The result should be sorted by the total revenue in descending order. Just output the first 100 rows (Hint: use ROWNUM <= 100).

16. The total expenditure of a school district is indicated by the TOTALEXP column in the SCHOOL_FINANCE_2010_T table. Find out the total school expenditure of each state. Include the following columns in your query output: stcode, stname, and the total school expenditure of the state. Sort your output with the total school expenditure in descending order.

17. Calculate the total amount of the money that the United States spent on the public school systems in that year. Your query output **must** read something like

"The total amount that the United States spent on the public school systems in 2010 was $ XXXXXXXXXXXXX.X K."

Note:

- The integer part of the dollar amount is not necessary the same length as indicated by the number of X'es here, but you do need to keep one decimal point.
- There is no column heading displayed. (Hint: use the "SET HEADING OFF" command to suppress headings and use "SET HEADING ON" command to restore to the normal condition).

18. Based on fedrev_t or strev_t, or locrev_t and other necessary tables, create 3 views, excluding any school districts that had no expenses (otherwise, you will have an error in calculation). You should check the value range of the ratios you are getting, which will affect how you choose the display format of your ratios (all the values must be displayed as required).
    A. fed_contribution_v, to calculate the federal revenue contribution to each school district (federal revenue/total expense), including these columns: idcensus, stcode, stname, sd_name and the fed_pcnt (for the federal revenue/total expense ratio). Keep 4 decimal points for the ratio. 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.
    B. st_contribution_v, to calculate the state revenue contribution to each school district (state revenue/total expense), including these columns: idcensus, stcode, stname, sd_name and the st_pcnt (for the state revenue/total expense ratio). Keep 4 decimal points for the ratio. List all the columns that exist in the st_contribution_v, and sort in descending order by st_pcnt. Find out school districts that received state revenues greater than the total expense, and sort in descending order by st_pcnt.
    C. loc_contribution_v, to calculate the local revenue contribution to each school district (local revenue/total expense), including these columns: idcensus, stcode, stname, sd_name and the loc_pcnt (for the revenue/total expense ratio). Keep 4 decimal points for the ratio. Find out school districts that received local revenues greater than the total expense. List all the columns that existed in the loc_contribution_v, and sort in descending order by loc_pcnt.

19. Based on the three views that were created in Step 18, 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.
    A. Find out the school districts that received total revenues (federal+state+local) over 3 times of the total amount they actually spent in that year. List all the columns that exist in the fsl_contribution_v, and sort in descending order by fsl_pcnt.
    B. Find out the school districts that received total revenues (federal+state+local) up to 30% of the total amount they actually spent in that year. List all the columns that exist in the fsl_contribution_v, and sort in descending order fsl_pcnt.

20. In SASUSER, use PROC SQL to create a dataset in SAS called **Total_Rev** by querying the Total_Rev_v view you created in Step 14 in the Oracle database. This will create a SAS dataset that has the same contents as the view.

21. Do a correlation analysis (PROC CORR) on the three variables (tfedrev, tstrev, and tlocrev) of the Total_Rev dataset. Put the analysis results in the MS Word (or PDF) file and comment on the results.

    Hint: use the following syntax of PROC CORR

```
proc corr data=…
plots(maxpoints=NONE)=matrix(histogram);
var variable1 variable2 variable3;
run;
```

22. Do a regression analysis (PROC REG) on variables (tfedrev, tstrev, and tlocrev) of the Total_Rev dataset. Use tfedrev as the dependent variable and tstrev and tlocrev as the independent variables. Put the analysis results in the MS Word (or PDF) file and comment on the results.

    Hint: use the following syntax of PROC REG

```
proc reg data=…;
model dependent-variable = independent-variable1 independent-variable2;
run;
```

23. Do the correlation analysis and regression analysis by directly getting data from the Total_Rev_v view in the Oracle database through the myoracle libref you created earlier. You should achieve the same results as above. Is this an instance of in-database processing? Why or why not?

24. Using PROC SQL and the librefs (myoracle and Final) created earlier in Step 4, create an Oracle database table, **School_Finance_2015_t** by querying the School_Finance_2015.sas7bdat dataset.

25. From 2010 to 2015, some school districts total revenues increased, some decreased and a small number of them stayed the same. Using sql*plus, find out

    A. The top 5 school districts that had increased total revenues.
    B. The top 5 school districts that had decreased total revenues.
    C. All the school districts whose total revenues stayed the same.

    Display the state code, state name, IDCENSUS, the name of school district, the difference of the total revenues of the two years (revdif=2015's totalrev – 2010's totalrev) and the change percentage (change_percentage = 100*revdif/2010's totalrev). The results are sorted by revdif. For all the values, keep one decimal point.

26. Report a screenshot of your final ORACLE SQL DEVELOPER interface, showing **all** your tables, views and the connection that you have created in the whole project. Make sure that you expand the Tables and Views folders so that all your tables and Views are visible.