```
/* to view the table attributs */

proc contents data="/home/u64168505/EPG1V2/data/class_birthdate.sas7bdat";
run;

/*
 The file path has the two pieces of information that are required for SAS to read:
     1. location
     2. name and type of data
  what issues might aries from using a hardcoded path?
     1. long program
     2. data location changing
     3. other data types (like excel)
  These issues is solve by SAS library
*/
/*
    SAS library:
      give you a easy way to spesify the two required pieces of information:
      "location and type"
    library --> is a collection of data files that are the same location and type
    library is global statement
*/

/* create a library

libname libref engine "path";

libfref --> the name of the library
engine  --> the name of the behin scren of instructions for read structure data
            different engine for each data type:
            SAS table(Base), Excel, Teradata, Hadoop...
path    --> location for the information that you want to read

*/
/*
  to access the file table we write the name of the library
  .the name of table we want to access

  */

libname mylib base "/home/u64168505/EPG1V2/output";

proc contents data = mylib.class_copy2;
run;

/*
 Excel file
   first:
    - options statement we need to transfer the excel column heading no rule to sas rule
   using:
    - options validvarname = V7;
   second:
    - engine --> xlsx
   finally:
    - when you define a connection to a data source such as excel or other database
    - it is good to clear or delete the libref at the end of program
   issue:
    - while the library is active it could create a lock that prevents others from accessing file
   using:
    - libname lifbref clear;
```

```
*/
/*1*/
options validvarname=v7;
libname xlclass xlsx "/home/u64168505/EPG1V2/data/class.xlsx";


proc contents data=xlclass.class_birthdate;
run;

libname xlclass clear;

/*2*/
option validvarname=v7;
libname Example xlsx "/home/u64168505/EPG1V2/data/storm.xlsx" ;

proc contents data= Example.storm_summary;
run;

libname Example clear;

/*
 Unstructure data to Structure using SAS
  CSV:
    using import procedure
     - proc import datafile="path/filename" DBMS = filetype
               out = output-table<Replace>;
               <guessingrows=n|Max;>
       run;
     datafile     --> to assign path
     DBMS         --> to assign the type of file
     out          --> to provide the library and name sas output table
     <Replace>    --> to overwrite the sas output
     guessingrows --> to provide a number of rows to examine
     Max          --> used to examine all rows
*/
/*1*/
proc import datafile="/home/u64168505/EPG1V2/data/class_birthdate.csv"
    dbms=csv out=work.class_birthdate_import
    replace;
run;
/*2*/
proc import datafile="/home/u64168505/EPG1V2/data/storm_damage.csv"
     dbms=csv out= work.strom_damage_import
     replace;
run;

/*
  Rather then read and write excel data directly we can import like csv

*/

proc import datafile="/home/u64168505/EPG1V2/data/class.xlsx"
     dbms=xlsx out= work.class_birthdate
     replace;
     sheet=class_test;
run;
```