```
306  *STEP 0 ;
307  /************************************************************************************************
307! ****************************/
308  /*1. Program Name:Vivek235_HW06_Program.sas
308!                                */
309  /* Program Location: C:\Users\vigupta\OneDrive\Learning\DataScience\Statistics Texas A&M
309! University\657\Homework\Assignment06\Assignment\Vivek235_HW06_Program.sas      */
310  /* Date Created: 2/18/17
310!                            */
311  /* Author: Vivek Kumar Gupta
311!                            */
312  /* Purpose: This assignment will primarily utilize, but is not limited to, techniques covered in
312! lectures 5 through 9 but does not require the use of any of the SQL set operators.
313  You will practice using joins, subqueries, inline views and summary functions.
313! */
314  /************************************************************************************************
314! ****************************/
315
316  *STEP 0 - Setup of libraries and fielrefs. Librefs to Orion and homework data must be protected
316! with readonly access. Use a filename
317  statement to define the path to the PDF output file.;
318
319  *1.Create the necessary library references for data sources and destination and file references
319! for output. Turn off page numbering ;
320  libname ncaa  'C:\Users\vigupta\OneDrive\Learning\DataScience\Statistics Texas A&M
320! University\657\Homework\Assignment05\SourceData' access=readonly;
NOTE: Libref NCAA was successfully assigned as follows:
      Engine:        V9
      Physical Name: C:\Users\vigupta\OneDrive\Learning\DataScience\Statistics Texas A&M
      University\657\Homework\Assignment05\SourceData
321  libname givers  'C:\Users\vigupta\OneDrive\Learning\DataScience\Statistics Texas A&M
321! University\657\Homework\Assignment06\SourceData' access=readonly;
NOTE: Libref GIVERS was successfully assigned as follows:
      Engine:        V9
      Physical Name: C:\Users\vigupta\OneDrive\Learning\DataScience\Statistics Texas A&M
      University\657\Homework\Assignment06\SourceData
322  libname orion  'C:\Users\vigupta\OneDrive\Learning\DataScience\Statistics Texas A&M
322! University\657\SQL Files' access=readonly;
NOTE: Libref ORION was successfully assigned as follows:
      Engine:        V9
      Physical Name: C:\Users\vigupta\OneDrive\Learning\DataScience\Statistics Texas A&M
      University\657\SQL Files
323  filename pdfdev 'C:\Users\vigupta\OneDrive\Learning\DataScience\Statistics Texas A&M
323! University\657\Homework\Assignment06\Vivek235_HW06_Output.pdf';
324  option nonumber;
325
326  /*Open destination device*/
327  ods pdf file=pdfdev ;
NOTE: Writing ODS PDF output to DISK destination "PDFDEV", printer "PDF".
328
329  /*STEP 1. Create a report entitled ║2003 NCAA Team Scoring Analysis=, from the scholarship03
329! dataset using inline view and as instructed*/
330  proc sql ;
331  title "2003 NCAA Team Scoring Analysis";
332  select team
333          , count(*) as Players
334          , avg(ppg) as avg_ppg label 'Average PPG' format 5.1
335          , avg(ppg)/sc2.avg_PPG_all as overall_avg label 'Team vs. Overall' format percent8.1
336          , case
337              when  avg(ppg) > avg_PPG_all then 'Above Avg.'
338              else 'Avg. or Below'
339              end as ppg_level label 'PPG Level'
340  from ncaa.scholarship03 sc1,
341  (select  avg(ppg) as avg_PPG_all label "Overall Average PPG"
342          from ncaa.scholarship03
343          where Seed_ not in (15, 16)) as sc2
344  where sc1.Seed_ not in (15, 16)
345  group by team,avg_PPG_all
346  having players >= 5
347  order by avg_ppg desc;
NOTE: The execution of this query involves performing one or more Cartesian product joins that can
      not be optimized.
348  quit;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.06 seconds
      cpu time            0.06 seconds


349
350  /*STEP 3.Create a list of records from givers with duplicate names as shown in the example */
351  proc sql;
352
```

```
353  title"Duplicate Givers";
354
355  select  employee_id,
356          employee_name,
357          qtr1,
358          qtr2,
359          qtr3,
360          qtr4,
361          recipients
362  from givers.givers
363  where employee_name in
364      (select employee_name
365          from givers.givers
366          group by employee_name
367          having count(*) >1)
368  ;
369  quit;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.04 seconds
      cpu time            0.03 seconds



370
371  /*STEP 4.Create a list of Active Employees who are not in the giver list (based on employee_id).
371! Names
372  are found in orion.employee_addresses. The employee_term_date can be read from the
373  orion.employee_payroll table. Use a subquery in the where clause to determine which IDs to
374  eliminate. */
375
376  proc sql;
377
378  title "Active Employees not on Giver List";
379  select payroll.employee_id,
380         address.employee_name
381  from
382      orion.employee_payroll as payroll
383      inner join
384      orion.employee_addresses as address
385      on address.employee_id = payroll.employee_id and not payroll.employee_term_date
386  where payroll.employee_id not in
387  (select employee_id from givers.givers);
```

```
388
389  quit;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.06 seconds
      cpu time            0.06 seconds


390
391  /*STEP 5. Use data in one or more of the tables above to create a list of people from the givers
391! table who
392  are no longer active employees at Orion Star. Show the ID, Name, and Gender of terminated
393  employees.*/
394  proc sql;
395  title"Terminated Givers";
396
397  select payroll.employee_id as ID ,
398         address.employee_name as Name,
399         payroll.employee_gender as Gender
400  from
401      orion.employee_payroll as payroll
402      inner join  orion.employee_addresses as address
403          on address.employee_id = payroll.employee_id and payroll.employee_term_date
404  where payroll.employee_id in
405  (select employee_id from givers.givers);
406  quit;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.03 seconds
      cpu time            0.03 seconds


407
408  /*STEP 6. Create a report entitled ║Orion’s Customers Who Bought Products Other Than Shoes= using
408!  a
409  multiway join.*/
410  proc sql;
411
412  title"Orion's Customers Who Bought Products Other Than Shoes";
413  select distinct c.customer_id,
414         c.customer_name,
415         c.customer_address,
416         c.country,
417         prd.product_group,
418         Month(c.birth_date) label "Birth Month"
419  from
420      orion.order_fact as odf
421      inner join orion.customer as c
422          on c.customer_id=odf.customer_id
423      inner join  orion.product_dim as prd
424          on prd.product_id=odf.product_id
425  where prd.product_group not like '%Shoes%'
426  order by c.country, 6 ,c.customer_name,prd.product_group
427          ;
428
429  quit;
NOTE: PROCEDURE SQL used (Total process time):
      real time           0.20 seconds
      cpu time            0.18 seconds


430
431  /*Housekeeping*/
432  title"";
433  option number;
434
435  /*Close destination*/
436  ods pdf close;
NOTE: ODS PDF printed 21 pages to C:\Users\vigupta\OneDrive\Learning\DataScience\Statistics Texas A&M
      University\657\Homework\Assignment06\Vivek235_HW06_Output.pdf.
```