```
*STEP 0 ;
/*************************************************************************************************
*****************/
/*1. Program Name:Vivek235_HW06_Program.sas
                                                                    */
/* Program Location: C:\Users\vigupta\OneDrive\Learning\DataScience\Statistics Texas A&M
University\657\Homework\Assignment06\Assignment\Vivek235_HW06_Program.sas          */
/* Date Created: 2/18/17
                                                                    */
/* Author: Vivek Kumar Gupta
                                                                */
/* Purpose: This assignment will primarily utilize, but is not limited to, techniques covered in lectures 5
through 9 but does not require the use of any of the SQL set operators.
You will practice using joins, subqueries, inline views and summary functions.
      */
/*************************************************************************************************
*****************/

*STEP 0 - Setup of libraries and fielrefs. Librefs to Orion and homework data must be protected with
readonly access. Use a filename
statement to define the path to the PDF output file.;

*1.Create the necessary library references for data sources and destination and file references for output.
Turn off page numbering ;
libname ncaa  'C:\Users\vigupta\OneDrive\Learning\DataScience\Statistics Texas A&M
University\657\Homework\Assignment05\SourceData' access=readonly;
libname givers  'C:\Users\vigupta\OneDrive\Learning\DataScience\Statistics Texas A&M
University\657\Homework\Assignment06\SourceData' access=readonly;
libname orion  'C:\Users\vigupta\OneDrive\Learning\DataScience\Statistics Texas A&M University\657\SQL
Files' access=readonly;
filename pdfdev 'C:\Users\vigupta\OneDrive\Learning\DataScience\Statistics Texas A&M
University\657\Homework\Assignment06\Vivek235_HW06_Output.pdf';
option nonumber;

/*Open destination device*/
ods pdf file=pdfdev ;

/*STEP 1. Create a report entitled "2003 NCAA Team Scoring Analysis", from the scholarship03 dataset using
inline view and as instructed*/
proc sql ;
title "2003 NCAA Team Scoring Analysis";
```

```
select team
            , count(*) as Players
            , avg(ppg) as avg_ppg label 'Average PPG' format 5.1
            , avg(ppg)/sc2.avg_PPG_all as overall_avg label 'Team vs. Overall' format percent8.1
            , case
                  when  avg(ppg) > avg_PPG_all then 'Above Avg.'
                  else 'Avg. or Below'
                  end as ppg_level label 'PPG Level'
from ncaa.scholarship03 sc1,
(select  avg(ppg) as avg_PPG_all label "Overall Average PPG"
            from ncaa.scholarship03
            where Seed_ not in (15, 16)) as sc2
where sc1.Seed_ not in (15, 16)
group by team,avg_PPG_all
having players >= 5
order by avg_ppg desc;
quit;

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

title"Duplicate Givers";

select      employee_id,
            employee_name,
            qtr1,
            qtr2,
            qtr3,
            qtr4,
            recipients
from givers.givers
where employee_name in
      (select employee_name
            from givers.givers
            group by employee_name
            having count(*) >1)
;
quit;

/*STEP 4.Create a list of Active Employees who are not in the giver list (based on employee_id). Names
are found in orion.employee_addresses. The employee_term_date can be read from the
```

```
orion.employee_payroll table. Use a subquery in the where clause to determine which IDs to
eliminate. */

proc sql;

title "Active Employees not on Giver List";
select payroll.employee_id,
         address.employee_name
from
      orion.employee_payroll as payroll
      inner join
      orion.employee_addresses as address
      on address.employee_id = payroll.employee_id and not payroll.employee_term_date
where payroll.employee_id not in
(select employee_id from givers.givers);

quit;

/*STEP 5. Use data in one or more of the tables above to create a list of people from the givers table who
are no longer active employees at Orion Star. Show the ID, Name, and Gender of terminated
employees.*/
proc sql;
title"Terminated Givers";

select payroll.employee_id as ID ,
         address.employee_name as Name,
         payroll.employee_gender as Gender
from
      orion.employee_payroll as payroll
      inner join  orion.employee_addresses as address
            on address.employee_id = payroll.employee_id and payroll.employee_term_date
where payroll.employee_id in
(select employee_id from givers.givers);
quit;

/*STEP 6. Create a report entitled “Orion’s Customers Who Bought Products Other Than Shoes” using a
multiway join.*/
proc sql;

title"Orion's Customers Who Bought Products Other Than Shoes";
select distinct c.customer_id,
```

```
          c.customer_name,
          c.customer_address,
          c.country,
          prd.product_group,
          Month(c.birth_date) label "Birth Month"
from
      orion.order_fact as odf
      inner join orion.customer as c
            on c.customer_id=odf.customer_id
      inner join  orion.product_dim as prd
            on prd.product_id=odf.product_id
where prd.product_group not like '%Shoes%'
order by c.country, 6 ,c.customer_name,prd.product_group
            ;

quit;

/*Housekeeping*/
title"";
option number;

/*Close destination*/
ods pdf close;
```