SQL-5 Exercises

1. Inner Joins

Produce a report containing **Employee_Name** and calculated years of service (YOS) as of December 31, 2007, by joining **orion.Employee_Addresses** and **orion.Employee_Payroll** on **Employee_ID**. Label the columns and provide two title lines, as shown in the sample output. Limit the report to employees where YOS > 30. Order the output alphabetically by **Employee_Name**.

- The **orion.Employee_Addresses** table contains the **Employee_Name** column.
- The **orion.Employee_Payroll** table contains the **Employee_Hire_Date** column.
- Both **orion.Employee_Addresses** and **orion.Employee_Payroll** contain columns named **Employee_ID**.
- Use TITLE1 and TITLE2 statements to produce title lines as indicated in the sample report:

Partial PROC SQL Output

Employees With More Than 30 Years of Service
As of December 31, 2007

Name	Years of Service
Abbott, Ray	32
Banchi, Steven	33
Blackley, James	33
Bleu, Henri Le	33
Branly, Wanda	33
Buddery, Jeannette	33
Campbell, Carston	33
Capps, Ramond	33

Age/years of service calculations can be difficult to render precisely. In this course, you use the following:

```
YOS =int(('31DEC2007'd= Employee_Hire_Date)/365.25)
```

Others might leverage the INTCK function, like this:

```
YOS=intck('year',Employee_Hire_Date,'31DEC2007'd)
```

For a more in-depth discussion, see “Accurately Calculating Age with Only One Line of Code” at support.sas.com/kb/24/808.html.

2. Outer Joins

Join **orion.Sales** and **orion.Employee_Addresses** on **Employee_ID** to create a report showing the names and cities of all Orion Star employees, and if an employee is in Sales, the job title. Present the report in alphabetical order by city, job title, and name.

- The **orion.Sales** table contains a record for every employee in the Sales Department and includes columns **Employee_ID** and **Job_Title**.
- The **orion.Employee_Addresses** table contains a record for every employee and includes **Employee_ID**, **Employee_Name**, and **City**.

Partial PROC SQL Output

Name	City	Job_Title
Blanton, Brig	Melbourne	
Catenacci, Reyne	Melbourne	
Dillin, Kerrin	Melbourne	
Fiocca, Jina	Melbourne	
Fouche, Madelaine	Melbourne	
Glattback, Ellis	Melbourne	
Graham-Rowe, Jannene	Melbourne	
Gromek, Gladys	Melbourne	
Harwood, Reece	Melbourne	
Hieds, Merle	Melbourne	
Horsey, Riu	Melbourne	
Mccleary, Bill	Melbourne	
Moffat, Trent	Melbourne	
Pettolino, Peter	Melbourne	
Povey, Liz	Melbourne	
Santomaggio, Pearl	Melbourne	
Sheedy, Sherie	Melbourne	
Streit, Russell	Melbourne	
Zhou, Tom	Melbourne	Sales Manager
Barcoe, Selina	Melbourne	Sales Rep. I
Chantharasy, Judy	Melbourne	Sales Rep. I
Duckett, Shani	Melbourne	Sales Rep. I
Osborn, Hernani	Melbourne	Sales Rep. I
Pa, Koavea	Melbourne	Sales Rep. I
Scordia, Randal	Melbourne	Sales Rep. I
Simms, Doungkamol	Melbourne	Sales Rep. I
Aisbitt, Sandy	Melbourne	Sales Rep. II
George, Vino	Melbourne	Sales Rep. II
Magrath, Brett	Melbourne	Sales Rep. II

3. Joining Multiple Tables

Create a report showing Orion Star Internet customers residing in the U.S. or Australia who purchased foreign manufactured products, that is, a product that was not manufactured in their country of residence. The report should be titled **US and Australian Internet Customers Purchasing Foreign Manufactured Products** and should display the customers' names and the number of foreign purchases made. Present the information so that those with the largest number of purchases appear at the top of the report, and customers who have the same number of purchases are displayed in alphabetical order.

Employee_ID 99999999 is a dummy ID that can be used to identify Internet orders. The data that you need can be found in the listed columns of the following tables:

- **orion.Product_Dim** contains
  - **Product_ID**
  - **Supplier_Country**
- **orion.Order_Fact** contains
  - **Product_ID**
  - **Customer_ID**
- **orion.Customer** contains
  - **Customer_ID**
  - **Country**

Partial PROC SQL Output

US and Australian Internet Customers
Purchasing Foreign Manufactured Products

Name	Purchases
Candy Kinsey	10
Phenix Hill	7
Cynthia Mccluney	5
Korolina Dokter	5
Najma Hicks	4
Robert Bowerman	4

4. Using In-Line Views

a. Produce a report of Orion Star sales force employees' aggregate sales in 2007. Select **Country**, **First_Name**, **Last_Name**, **Value_Sold**, **Orders**, and **Avg_Order** columns by joining **orion.Order_Fact** and **orion.Sales** tables on **Employee_ID**. Group the report by **Country**, **First_Name**, **Last_Name**. Include only employees having an aggregate **Value_Sold** of $200.00 or more. Order the results by **Country**, **Value_Sold** (descending), and **Orders** (descending).

1) Calculate **Value_Sold** by summing **Total_Retail_Price**.

2) Calculate **Orders** by using the COUNT(*) function to count the number of rows returned for each employee.

3) Calculate **Avg_Order** by dividing **Value_Sold** by **Orders**.

4) Title the report as indicated in the sample output.

Partial PROC SQL Output

2007 Sales Force Sales Statistics
For Employees With 200.00 or More In Sales

Country	First_Name	Last_Name	Value_Sold	Orders	Avg_Order
AU	Lucian	Daymond	880.10	5	176.02
AU	Ranj	Stamalis	697.60	3	232.53
AU	Sharryn	Clarkson	400.40	3	133.47
AU	Marinus	Surawski	398.80	2	199.40
AU	Sian	Shannan	306.20	1	306.20
AU	Monica	Kletschkus	239.30	2	119.65
US	Tywanna	Mcdade	1,387.90	2	693.95

b. Using the query created in step **a** as an in-line view, select **Country**, the maximum **Value_Sold**, **Orders**, and **Avg_Order** as well as the minimum **Avg_Order** for each country. Name the report **`2007 Sales Summary by Country`**.

Hint: An in-line view must not use the ORDER BY clause.

PROC SQL Output

2007 Sales Summary by Country

Country	Max Value Sold	Max Orders	Max Average	Min Average
AU	880.10	5.00	306.20	119.65
US	1,387.90	6.00	693.95	66.50

5. Building Complex Queries with In-Line Views

Your ultimate goal in this exercise is to create a report showing each employee's salary expressed as a percentage of the total salary for his department. The report should be sorted by department and, within each department, in descending order of salary percentage.

- The **`orion.Employee_Payroll`** table contains **`Salary`**.
- The **`orion.Employee_Addresses`** table contains **`Employee_Name`**.
- The **`orion.Employee_Organization`** table contains **`Department`**.

Sketch of desired report:

Employee Salaries as a percent of Department Total

Department	Employee_Name	Salary	Percent
Accounts	Mea, Azavi0us	58,200.00	8.6%
Accounts	Miller, Pamela	53,475.00	7.9%
Accounts	Asta, Wendy	52,295.00	7.7%

a. Create a report aggregating the sum of all salaries for each department. The report should include **`Department`** and the sum of all associated salary values as **`Dept_Salary_Total`**. Join **`orion.Employee_Payroll`** and **`orion.Employee_Organization`** **by** **`Employee_ID`** to obtain the information you need.

- The **`orion.Employee_Payroll`** table contains salary values.

- The **orion.Employee_Organization** table contains department information.

Partial PROC SQL Output

Department	Dept_Salary_Total
Accounts	680440
Accounts Management	397175
Administration	1009850
Concession Management	372225
Engineering	276285

b. Create a report that includes the employee ID, name, and department. Join **orion.Employee_Addresses** and **orion.Employee_Organization** by **Employee_ID** to obtain the information you need.

- The **orion.Employee_Addresses** table contains **Employee_Name** and **Employee_ID**.
- The **orion.Employee_Organization** table contains **Employee_ID** and **Department**.

Partial PROC SQL Output

Employee_ID	Employee_Name	Department
121044	Abbott, Ray	Sales
120145	Aisbitt, Sandy	Sales
120761	Akinfolarin, Tameaka	Accounts
120656	Amos, Salley	Logistics Management
121107	Anger, Rose	Sales
121038	Anstey, David	Sales

c. Use the two queries you created in steps **a** and **b** as in-line views. Join the views with **orion.Employee_Payroll** by either **Employee_ID** or **Department** to create the final report.

- The query from step **a** contains **Department** and **Dept_Salary_Total**.
- The query from step **b** contains **Employee_ID**, **Employee_Name**, and **Department**.
- The **orion.Employee_Payroll** table contains **Employee_ID** and individual **Salary** values.

Partial PROC SQL Output

Employee Salaries as a percent of Department Total

Department	Employee_Name	Salary	Percent
Accounts	Mea, AzaviOus	58,200.00	8.6%
Accounts	Miller, Pamela	53,475.00	7.9%
Accounts	Asta, Wendy	52,295.00	7.7%
Accounts	Post, Nahliah	48,380.00	7.1%
Accounts	Ferrari, Ralph	47,000.00	6.9%
Accounts	Kimmerle, Kevie	46,090.00	6.8%
Accounts	Farthing, Zashia	43,590.00	6.4%
Accounts	Knopfmacher, Paul	38,545.00	5.7%
Accounts	Thoits, Elizabeth	36,440.00	5.4%

Accounts	Apr, Nishan	36,230.00	5.3%
Accounts	Atkins, John	34,760.00	5.1%
Accounts	Voltz, Sal	34,040.00	5.0%
Accounts	Woods, Connie	32,675.00	4.8%
Accounts	Akinfolarin, Tameaka	30,960.00	4.5%
Accounts	Leone, Marvin	30,625.00	4.5%
Accounts	Van Damme, Jean-Claude	30,590.00	4.5%
Accounts	Niemann, Kevin	26,545.00	3.9%
Accounts Management	Kempster, Janelle	53,400.00	13%
Accounts Management	Kokoszka, Nikeisha	51,950.00	13%
Accounts Management	Lightbourne, Abelino	47,990.00	12%

6. Building a Complex Query Using a Multi-Way Join

Create a report using a multi-way inner join, which produces the total of the 2007 sales figures for each Orion Star employee. The report should be titled **`2007 Total Sales Figures`** and must include both the managers' and employees' names (displayed as first name followed by last name), and the total retail value of all sales made by each employee in 2007. Present the information as follows:

- Use one row per employee.
- Organize the report so that the following standards are observed:
  - Employees under one manager are adjacent to each other (grouped together) on the report.
  - Within each manager's group, employees are listed in decreasing order of total sales.
  - The Australian groups are listed first, followed by the U.S. groups.
  - Manager names are in alphabetical order by last name and then first name.

Remember that you can group and order by columns that are not included in the SELECT statement list.

The data that you need can be found in the following tables (variables of interest in parentheses):

- **`orion.Order_Fact`** (**`Employee_ID`**, **`Total_Retail_Price`**)
- **`orion.Employee_Organization`** (**`Employee_ID`**, **`Manager_ID`**)
- **`orion.Employee_Addresses`** (**`Employee_ID`**, **`Employee_Name`**)

Partial PROC SQL Output

2007 Total Sales Figures

Manager	Employee	Total_Sales
Wilson Dawes	Jina Fiocca	223.80
Wilson Dawes	Phu Sloey	17.60
Wilson Dawes	Amanda Liebman	6.40
Tom Zhou	Lucian Daymond	880.10
Tom Zhou	Ranj Stamalis	697.60
Tom Zhou	Sharryn Clarkson	400.40
Tom Zhou	Marinus Surawski	398.80
Tom Zhou	Sian Shannan	306.20
Tom Zhou	Monica Kletschkus	239.30
Tom Zhou	Fancine Kaiser	147.10

Tom Zhou	Shani Duckett	101.50
Tom Zhou	Atul Leyden	92.50
Tom Zhou	Kevin Lyon	73.99
Tom Zhou	Andrew Conolly	60.80
Tom Zhou	Sean Dives	19.10
Tom Zhou	Lynelle Phoumirath	19.10
Renee Capachietti	Brienne Darrohn	533.40
Renee Capachietti	Michael Westlund	366.00

7. Joining Multiple Tables

Create a report of Orion Star employees with more than 30 years of service as of December 31, 2007. Display the employee's name, years of service, and the employee's manager's name. Order the report alphabetically by manager name, by descending years of service, and then alphabetically by employee name. Label the columns and title the report as shown in the sample output.

The data that you need can be found in the listed columns of the following tables:

- **orion.Employee_Addresses** contains
  - **Employee_ID**
  - **Employee_Name**
- **orion.Employee_Payroll** contains
  - **Employee_ID**
  - **Employee_Hire_Date**
- **orion.Employee_Organization** contains
  - **Employee_ID**
  - **Manager_ID** (**Employee_ID** of the person's manager)

Partial PROC SQL Output

Employees with more than 30 years of service
as of December 31, 2007

Employee Name	Years of Service	Manager Name
Marion, Chiorene	33	Ardskin, Elizabeth
Zied, Ahmed	33	Ardskin, Elizabeth
Buddery, Jeannette	33	Billington, Kareen
Hornsey, John	33	Billington, Kareen
Horsey, Riu	33	Billington, Kareen