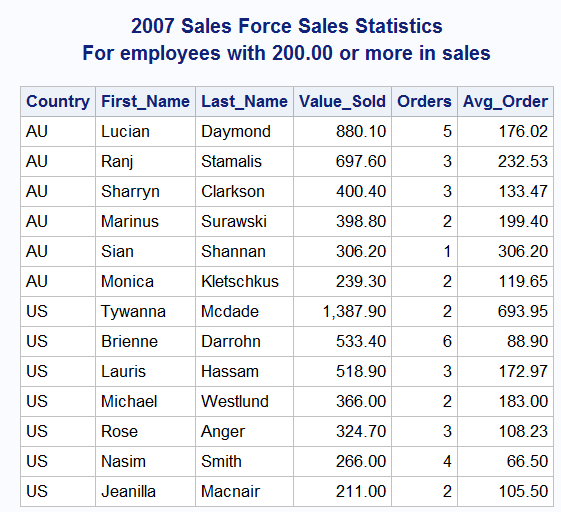
1. **Using In-Line Views**
   1. Produce a report of Orion Star sales force employees’ aggregate sales in 2007. Select **Country**, **First\_Name**, **Last\_Name**, **Value\_Sold**, **Orders**, and **Avg\_Order** columns by joining **orion.Order\_Fact** and **orion.Sales** tables on **Employee\_ID**. 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 (use two lines), the first line should read 2007 Sales Force Sales Statistics and the second line should read For Employees With 200.00 or More In Sales

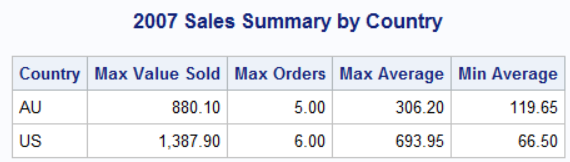
Requested Output:



* 1. 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.

Requested Output:



1. **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 TotalDepartment Employee\_Name Salary PercentAccounts Mea, Azavi0us 58,200.00 8.6%Accounts Miller, Pamela 53,475.00 7.9%Accounts Asta, Wendy 52,295.00 7.7% |

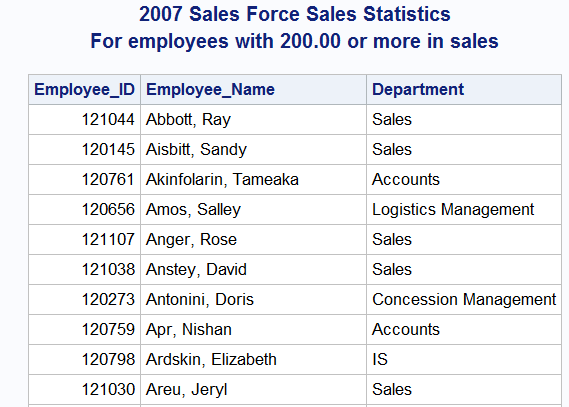
* 1. Create a report aggregating the sum of all salaries for each department. The report should include **Department** and the sum of all associated salary values as **Dept\_Salary\_Total**. Join **orion.Employee\_Payroll** and **orion.Employee\_Organization** **by Employee\_ID** to obtain the information you need.
* The **orion.Employee\_Payroll** table contains salary values.
* The **orion.Employee\_Organization** table contains department information.

Requested output (partial):



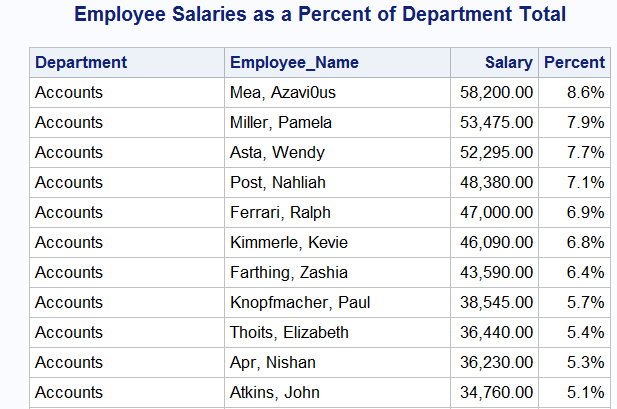
* 1. Create a report that includes the employee ID, name, and department. Join **orion.Employee\_Addresses** and **orion.Employee\_Organization** by **Employee\_ID** to obtain the information you need.
* The **orion.Employee\_Addresses** table contains **Employee\_Name** and **Employee\_ID**.
* The **orion.Employee\_Organization** table contains **Employee\_ID** and **Department**.

Requested output (partial)



* 1. 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.

Requested Output (Partial)



1. **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**)

Requested Output (Partial):

