**Homework, SQL Subqueries**

**All tables are in the sql subdirectory.**

1. **Using Subqueries**

The **orion.Order\_Fact** table contains information about orders that were placed by Orion Star Sales staff. Create a report that lists the Sales staff whose average quantity of items sold exceeds the company average quantity of items sold.

* 1. Write a query that displays the average **Quantity** for all rows in the table.
* Use **AVG(Quantity)** to calculate the average.
* Use the **orion.Order\_Fact** table.

The requested output:



* 1. Write a query that displays **Employee\_ID** and **AVG(Quantity)** for those employees whose average exceeds the company average. The query should do the following:
* Display the values for **Employee\_ID** and **AVG(Quantity)**. Name the second column **MeanQuantity**.
* Use the **orion.Order\_Fact** table.
* Group the data by **Employee\_ID**.
* Include only groups where the employee’s average quantity of items sold exceeds the company average. Use the query from step **1.a.** as a subquery in the HAVING clause.
* Add a title to the report: Employees whose Average Quantity Items Sold Exceeds the Company’s Average Items Sold

The requested output (Partial):



1. **Using a Noncorrelated Subquery**

Each month a memo is posted that lists the employees who have employment anniversaries for that month. Create the report for February and list **Employee\_ID** and the first and last names for all employees hired during the month of February of any year.

You can find **Employee\_Name** in the **orion.Employee\_Addresses** table and **Employee\_Hire\_Date** in the **orion.Employee\_Payroll** table. Both tables contain the column **Employee\_ID**. Order the report by an employee's last name.

* 1. Create a query that returns a list of employee IDs for employees with a February anniversary. The query should do the following:
* Display **Employee\_ID** numbers.
* Use the **orion.Employee\_Payroll** table.
* Return only employees whose **Employee\_Hire\_Date** isin February.
* Add a title to the report: Employee IDs for February Anniversaries

Requested output (Partial):



* 1. Using the query in step **2.a.** as a noncorrelated subquery, write a query that displays the employee IDs and names of employees who have February anniversaries. The final query should do the following:
* Display **Employee\_ID** and split **Employee\_Name** into two new columns: **FirstName** and **LastName**. Both new columns should have a length of $15 and appropriate labels. (See the report below.) The original **Employee\_Name** is stored as **Lastname, Firstname**.
* Use the **orion.Employee\_Addresses** table.
* Select only employee IDs for employees who had February anniversary months.
* Order the final results by **LastName**.
* Create an appropriate title.

Requested output (Partial):



1. **Creating Subqueries Using the ALL Keyword**

In most companies, you can assume that the higher-level job titles have employees that are older than employees with a lower-level job title. Using the **orion.Staff** table, determine whether there are any lower-level purchasing agents (Purchasing Agent I and Purchasing Agent II) that are older than all the higher-level purchasing agents (Purchasing Agent III). The final report should display **Employee\_ID**, **Job\_Title**, **Birth\_Date**, and a calculated **Age** column for the employee as of 24Nov2007.

Hint: Use the SAS date constant ('24Nov2007'd) in the calculation for **Age**.

Requested output:



1. **Using Nested Subqueries**

Orion Star Sales managers are interested in rewarding the top sales person at the company. The **orion.Order\_Fact** table contains information about all sales, including the employee ID of the staff member responsible for making the sale. The **orion.Employee\_Addresses** table contains the ID and name of every employee in the company.

Generate a report that shows **Employee\_ID** and the respective staff member’s calculated total sales figures from the **orion.Order\_Fact** table. Calculate the total sales figures by summing the product of **Total\_retail\_price**\***Quantity**. The **Employee\_ID** number 99999999 is a generic employee ID number that indicates an Internet sale for which no staff member can take credit. Exclude the **Employee\_ID** number 99999999 when you determine the employee with the highest total sales.

* 1. Generate a report that shows **Employee\_ID** and calculated **Total\_Sales** of the employee with the highest sales from the **orion.Order\_Fact** table.



* 1. Generate another report that displays **Employee\_ID** and **Employee\_Name** of the employee with the highest sales. The **orion**.**Employee\_Addresses** table contains the employee names.

Requested output:



* 1. Write a query that combines the two queries above in order to generate a report that adds the **Total\_Sales** column with **Employee\_ID** and **Employee\_Name** and the calculated **Total\_Sales** column.

Requested output:



1. **Creating a Simple Correlated Subquery**

Create a report showing **Employee\_ID** and the birth month (calculated as **month(Birth\_date)**) for all Australian employees, using a correlated subquery.

The table **orion.Employee\_Payroll** contains **Employee\_ID** and **Birth\_Date**.

In the subquery, select only **Country** from **orion.Employee\_Addresses**. Use a WHERE clause to return only rows where the **EmployeeID** in **orion.Employee\_Addresses** matches the **EmployeeID** in **orion.Employee\_Payroll**.

Order the report by birth month.

Requested output (Partial)



1. **Using a Correlated Subquery**

Generate a report that displays **Employee\_ID**, **Employee\_Gender**, and **Marital\_Status** for all employees who donate more than 0.002 of their salary. The table **orion.Employee\_donations** contains **Employee\_ID**, quarterly donations (**Qtr1**-**Qtr4**), and charities (**Recipients**). The **orion.Employee\_Payroll** table contains **Employee\_ID** and **Salary** information.

Requested output:

