

Problem 1:

Create a query showing department_name and the count of employees in that department. Use a subquery (count(*) employees) as a column expression and label the column Count_Emp. Order by department name.

SQL Script:

```
SELECT d.department_name, COUNT(*) "Count_Emp"  
FROM departments d  
    INNER JOIN employees e  
        ON e.department_id = d.department_id  
GROUP BY d.department_name;
```

Console Output:

```
SQL> SELECT d.department_name, COUNT(*) "Count_Emp"  
  2  FROM departments d  
  3  INNER JOIN employees e  
  4  ON e.department_id = d.department_id  
[ 5* GROUP BY d.department_name;  


| DEPARTMENT_NAME  | Count_Emp |
|------------------|-----------|
| Sales            | 34        |
| Marketing        | 2         |
| Administration   | 1         |
| Purchasing       | 6         |
| Shipping         | 45        |
| IT               | 5         |
| Executive        | 3         |
| Finance          | 6         |
| Public Relations | 1         |
| Human Resources  | 1         |
| Accounting       | 2         |

  
11 rows selected.
```

Problem 2:

Create a query displaying the last_name concatenated with a comma and blank space and first_name, the hire date, and the salary of any employee working in the same department as Sigal Tobias (employee_id of 117). Exclude that employee from the final result. Sort it by the last name.

SQL Script:

```
SELECT last_name || ',' || first_name name, hire_date, salary
FROM employees
WHERE department_id =
(
  SELECT department_id
  FROM employees
  WHERE employee_id = 117
)
AND employee_id <> 117
ORDER BY last_name;
```

Console Output:

```
SQL> SELECT last_name || ',' || first_name name, hire_date, salary
  2  FROM employees
  3  WHERE department_id =
  4  (
  5  SELECT department_id
  6  FROM employees
  7  WHERE employee_id = 117
  8  )
  9  AND employee_id <> 117
[ 10* ORDER BY last_name;
NAME          HIRE_DATE      SALARY
-----        -----
Baida, Shelli    24-DEC-97    2900
Colmenares, Karen 10-AUG-99    2500
Himuro, Guy       15-NOV-98    2600
Khoo, Alexander   18-MAY-95    3100
Raphaely, Den      07-DEC-94  11000
```

Problem 3:

Create a query displaying the employee_id, last_name, and department_id of all employees who work in a department with any employee whose last name starts with Hi. Order the data by department_id, last_name.

SQL Script:

```
SELECT employee_id, last_name, department_id
FROM employees
WHERE department_id = ANY
(
  SELECT department_id
  FROM employees
  WHERE SUBSTR(last_name, 1, 2) = 'Hi'
)
ORDER BY department_id, last_name;
```

Console Output:

```
SQL> SELECT employee_id, last_name, department_id
  2  FROM employees
  3  WHERE department_id = ANY
  4  (
  5  SELECT department_id
  6  FROM employees
  7  WHERE SUBSTR(last_name, 1, 2) = 'Hi'
  8  )
  9* ORDER BY department_id, last_name;
EMPLOYEE_ID LAST_NAME          DEPARTMENT_ID
-----  -----
      116 Baida                  30
      119 Colmenares             30
      118 Himuro                 30
      115 Khoo                   30
      114 Raphaely                30
      117 Tobias                  30
      206 Gietz                   110
      205 Higgins                 110
8 rows selected.
```

Problem 4:

Display employee_id, last_name, department_id, salary, and hire_date where department_id and salary match the department_id and salary of any employee hired before 1996. (Use a subquery with a compound where clause). Sort it by department_id and last_name.

SQL Script:

```
SELECT employee_id, last_name, department_id, salary, hire_date
FROM employees
WHERE (department_id, salary) IN
(
    SELECT department_id, salary
    FROM employees
    WHERE TO_CHAR(hire_date, 'yyyy') < '1996'
)
ORDER BY department_id, last_name;
```

Console Output:

```

SQL> SELECT employee_id, last_name, department_id, salary, hire_date
  2  FROM employees
  3 WHERE (department_id, salary) IN
  4  (
  5   SELECT department_id, salary
  6   FROM employees
  7  WHERE TO_CHAR(hire_date, 'yyyy') < '1996'
  8  )
[ 9* ORDER BY department_id, last_name;
EMPLOYEE_ID LAST_NAME DEPARTMENT_ID SALARY HIRE_DATE
----- ----- ----- -----
      200 Whalen          10     4400 17-SEP-87
      115 Khoo            30     3100 18-MAY-95
      114 Raphaely        30    11000 07-DEC-94
      203 Mavris           40     6500 07-JUN-94
      189 Dilly            50     3600 13-AUG-97
      122 Kaufling         50     7900 01-MAY-95
      137 Ladwig           50     3600 14-JUL-95
      141 Rajs             50     3500 17-OCT-95
      104 Ernst            60     6000 21-MAY-91
      103 Hunold           60     9000 03-JAN-90
      204 Baer             70    10000 07-JUN-94
      102 De Haan          90    17000 13-JAN-93
      100 King              90    24000 17-JUN-87
      101 Kochhar           90    17000 21-SEP-89
      109 Faviet            100    9000 16-AUG-94
      108 Greenberg         100   12000 17-AUG-94
      206 Gietz             110    8300 07-JUN-94
      205 Higgins           110   12000 07-JUN-94

```

18 rows selected.

Problem 5 Part 1:

Create a query showing the department manager (format: last_name, first_name, aliased dep_manager), department_id, and department_name.

SQL Script:

```

SELECT CASE WHEN e.last_name IS NULL THEN "
  ELSE e.last_name || ',' || e.first_name END dep_manager,
        d.department_id, d.department_name
FROM departments d
  LEFT OUTER JOIN employees e
    ON d.manager_id = e.employee_id;

```

Console Output:

```
|SQL> SELECT CASE WHEN e.last_name IS NULL THEN ''  
2 ELSE e.last_name || ', ' || e.first_name END dep_manager,  
3 d.department_id, d.department_name  
4 FROM departments d  
5 LEFT OUTER JOIN employees e  
[ 6* ON d.manager_id = e.employee_id;  
  
DEP_MANAGER DEPARTMENT_ID DEPARTMENT_NAME  
-----  
-----  
Baer, Hermann 70 Public Relations  
Fripp, Adam 50 Shipping  
Greenberg, Nancy 100 Finance  
Hartstein, Michael 20 Marketing  
Higgins, Shelley 110 Accounting  
Hunold, Alexander 60 IT  
King, Steven 90 Executive  
Mavris, Susan 40 Human Resources  
Raphaely, Den 30 Purchasing  
Russell, John 80 Sales  
Whalen, Jennifer 10 Administration  
120 Treasury  
130 Corporate Tax  
140 Control And Credit  
150 Shareholder Services  
160 Benefits  
170 Manufacturing  
180 Construction  
190 Contracting  
200 Operations  
210 IT Support  
220 NOC  
230 IT Helpdesk  
240 Government Sales  
250 Retail Sales  
260 Recruiting  
270 Payroll
```

27 rows selected.

Problem 5 Part 2:

Create a query showing last_name (aliased emp), department_id, and the employees' manager (format: last_name, first_name, aliased as emp_mngr). Create an outer join as the self join to include the boss Steven King. Since Steven King does not have a manager, display Boss as the emp_mngr.

SQL Script:

```
SELECT e.last_name emp, e.department_id,  
CASE WHEN m.last_name IS NULL THEN 'Boss'  
ELSE m.last_name || ', ' || m.first_name END emp_mngr  
FROM employees e  
LEFT OUTER JOIN employees m  
ON e.manager_id = m.employee_id;
```

Console Output:

```
|SQL> SELECT e.last_name emp, e.department_id,
| 2 CASE WHEN m.last_name IS NULL THEN 'Boss'
| 3 ELSE m.last_name || ', ' || m.first_name END emp_mgr
| 4 FROM employees e
| 5 LEFT OUTER JOIN employees m
| 6* ON e.manager_id = m.employee_id;
EMP                  DEPARTMENT_ID  EMP_MGR
-----
Ozer                      80 Cambrault, Gerald
Bloom                     80 Cambrault, Gerald
Fox                        80 Cambrault, Gerald
Smith                      80 Cambrault, Gerald
Bates                      80 Cambrault, Gerald
Kumar                      80 Cambrault, Gerald
Hunold                     60 De Haan, Lex
Vishney                     80 Errazuriz, Alberto
Greene                     80 Errazuriz, Alberto
McCain                     50 Vollman, Shanta
Jones                      50 Vollman, Shanta
Nayer                      50 Weiss, Matthew
Mikkilineni                50 Weiss, Matthew
Landry                      50 Weiss, Matthew
Markle                      50 Weiss, Matthew
Taylor                      50 Weiss, Matthew
Fleur                       50 Weiss, Matthew
Sullivan                    50 Weiss, Matthew
Geoni                       50 Weiss, Matthew
Abel                         80 Zlotkey, Eleni
Hutton                      80 Zlotkey, Eleni
Taylor                      80 Zlotkey, Eleni
Livingston                  80 Zlotkey, Eleni
Grant                        Zlotkey, Eleni
Johnson                     80 Zlotkey, Eleni
King                         90 Boss
107 rows selected.
```

Problem 5 Part 3:

Create a query showing the average salary (aliased dep_avg_sal) and the count of employees (aliased dep_emp_count) by department_id.

SQL Script:

```
SELECT department_id, AVdG(salary) dep_avg_sal, COUNT(*) dep_emp_count
FROM employees
GROUP BY department_id;
```

Console Output:

```

SQL> SELECT department_id, AVG(salary) dep_avg_sal, COUNT(*) dep_emp_count
  2  FROM employees
  3* GROUP BY department_id;
DEPARTMENT_ID DEP_AVG_SAL DEP_EMP_COUNT
-----
      50    3475.55556      45
      40      6500          1
     110     10150          2
      70     10000          1
     90    19333.3333       3
     30      4150          6
     10      4400          1
            7000          1
     20      9500          2
     60      5760          5
    100     8600          6
     80    8955.88235      34

```

12 rows selected.

Problem 5 Part 4:

Combine the queries from parts 1 – 3 in the following subquery factoring construct: WITH^[1]

q_dep_mgr (Part 1)

q_emp_mgr (Part 2)

q_dep_avg_sal_cnt (Part 3) --Main Query:

Display emp (aliased Employee), department_name (aliased Department Name), dep_avg_sal (rounded to 2 decimal digits, aliased Average Salary), dep_emp_count aliased Dep. Employee Count, dep_manager (aliased Department Manager), and emp_manager (aliased Employee Manager) based on the three subqueries joined on department_id. Make sure to display all employees (107 records). Order the result by department name and employee.

Note: Please shorten the column lengths in the output by either using COLUMN col_name FORMAT a{n} or by using the CAST function. Suggested lengths are:

> emp (20)

> department_name (20) > dep_manager (25)^[1]_[SEP]

> emp_manager (25)

SQL Script:

WITH

q_dep_mgr AS

```

(SELECT CASE WHEN e.last_name IS NULL THEN "
    ELSE e.last_name || ',' || e.first_name END dep_manager,
    d.department_id, d.department_name
FROM departments d
    LEFT OUTER JOIN employees e
        ON d.manager_id = e.employee_id),
q_emp_mgr AS
(SELECT e.last_name emp, e.department_id,
CASE WHEN m.last_name IS NULL THEN 'Boss'
    ELSE m.last_name || ',' || m.first_name END emp_mgr
FROM employees e
    LEFT OUTER JOIN employees m
        ON e.manager_id = m.employee_id),
q_dep_avg_sal_cnt AS
(SELECT department_id, AVG(salary) dep_avg_sal, COUNT(*) dep_emp_count
FROM employees
GROUP BY department_id)

— Main Query:

SELECT em.emp "Employee", d.department_name "Department Name",
    ROUND(dasc.dep_avg_sal, 2) "Average Salary", dasc.dep_emp_count "Dep. Employee Count",
    dm.dep_manager "Department Manager", em.emp_mgr "Employee Manager"
FROM q_emp_mgr em
    LEFT OUTER JOIN q_dep_avg_sal_cnt dasc
        ON em.department_id = dasc.department_id
    LEFT OUTER JOIN departments d
        ON dasc.department_id = d.department_id
    LEFT OUTER JOIN q_dep_mngr dm
        ON em.department_id = dm.department_id
ORDER BY "Department Name", "Employee";

```

Console Output:

```

SQL> WITH q_dep_mgr AS
  2 (SELECT CASE WHEN e.last_name IS NULL THEN ''
  3   ELSE e.last_name || ', ' || e.first_name END dep_manager,
  4   d.department_id, d.department_name
  5   FROM departments d
  6   LEFT OUTER JOIN employees e
  7   ON d.manager_id = e.employee_id),
  8 q_dep_avg_sal_cnt AS
  9 (SELECT e.last_name emp, e.department_id,
 10  CASE WHEN m.last_name IS NULL THEN 'Boss'
 11  ELSE m.last_name || ', ' || m.first_name END emp_mgr
 12  FROM employees e
 13  LEFT OUTER JOIN employees m
 14  ON e.manager_id = m.employee_id),
 15 q_dep_avg_sal AS
 16 (SELECT department_id, AVG(salary) dep_avg_sal, COUNT(*) dep_emp_count
 17  FROM employees
 18 GROUP BY department_id)
 19 SELECT em.emp "Employee", d.department_name "Department Name",
 20  ROUND(dasc.dep_avg_sal, 2) "Average Salary", dasc.dep_emp_count "Dep. Employee Count",
 21 dm.dep_manager "Department Manager", em.emp_mgr "Employee Manager"
 22 FROM q.emp_mgr em
 23 LEFT OUTER JOIN q.dep_avg_sal_cnt dasc
 24 ON em.department_id = dasc.department_id
 25 LEFT OUTER JOIN departments d
 26 ON dasc.department_id = d.department_id
 27 LEFT OUTER JOIN q.dep_mgr dm
 28 ON em.department_id = dm.department_id
 29* ORDER BY "Department Name", "Employee";

```

Employee	Department Name	Average Salary	Dep. Employee Count	Department Manager	Employee Manager
Gietz	Accounting	10150	2	Higgins, Shelley	Higgins, Shelley
Higgins	Accounting	10150	2	Higgins, Shelley	Kochhar, Neena
Whalen	Administration	4400	1	Whalen, Jennifer	Kochhar, Neena
De Haan	Executive	19333.33	3	King, Steven	King, Steven
King	Executive	19333.33	3	King, Steven	Boss
Kochhar	Executive	19333.33	3	King, Steven	King, Steven
Chen	Finance	8600	6	Greenberg, Nancy	Greenberg, Nancy
Faviet	Finance	8600	6	Greenberg, Nancy	Greenberg, Nancy
Greenberg	Finance	8600	6	Greenberg, Nancy	Kochhar, Neena
Ladwig	Shipping	3475.56	45	Fripp, Adam	Vollman, Shanta
Landry	Shipping	3475.56	45	Fripp, Adam	Weiss, Matthew
Mallin	Shipping	3475.56	45	Fripp, Adam	Kaufling, Payam
Markle	Shipping	3475.56	45	Fripp, Adam	Weiss, Matthew
Marlow	Shipping	3475.56	45	Fripp, Adam	Fripp, Adam
Matos	Shipping	3475.56	45	Fripp, Adam	Mourgos, Kevin
McCain	Shipping	3475.56	45	Fripp, Adam	Vollman, Shanta
Mikkilineni	Shipping	3475.56	45	Fripp, Adam	Weiss, Matthew
Mourgos	Shipping	3475.56	45	Fripp, Adam	King, Steven
Nayer	Shipping	3475.56	45	Fripp, Adam	Weiss, Matthew
OConnell	Shipping	3475.56	45	Fripp, Adam	Mourgos, Kevin
Olson	Shipping	3475.56	45	Fripp, Adam	Fripp, Adam
Patel	Shipping	3475.56	45	Fripp, Adam	Vollman, Shanta
Perkins	Shipping	3475.56	45	Fripp, Adam	Kaufling, Payam
Philtanker	Shipping	3475.56	45	Fripp, Adam	Kaufling, Payam
Rajs	Shipping	3475.56	45	Fripp, Adam	Mourgos, Kevin
Rogers	Shipping	3475.56	45	Fripp, Adam	Kaufling, Payam
Sarchand	Shipping	3475.56	45	Fripp, Adam	Fripp, Adam
Seo	Shipping	3475.56	45	Fripp, Adam	Vollman, Shanta
Stiles	Shipping	3475.56	45	Fripp, Adam	Vollman, Shanta
Sullivan	Shipping	3475.56	45	Fripp, Adam	Weiss, Matthew
Taylor	Shipping	3475.56	45	Fripp, Adam	Weiss, Matthew
Vargas	Shipping	3475.56	45	Fripp, Adam	Mourgos, Kevin
Vollman	Shipping	3475.56	45	Fripp, Adam	King, Steven
Walsh	Shipping	3475.56	45	Fripp, Adam	Mourgos, Kevin
Weiss	Shipping	3475.56	45	Fripp, Adam	King, Steven
Grant					Zlotkey, Eleni

107 rows selected.