# DBS311 Assignment #1

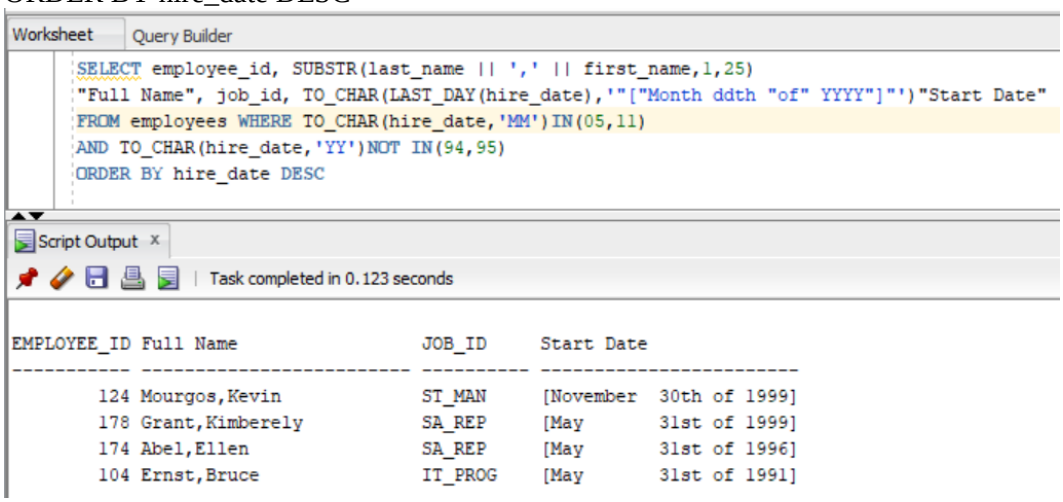
## Instructions:

**Single document submission only -- SELECT statements + ALL Outputs in SQL developer as clear screenshots.**

1. Display the employee number, full employee name, job and hire date of all employees hired in May or November of any year, with the most recently hired employees displayed first. Also, exclude people hired in 1994 and 1995. Full name should be in the form *Lastname, Firstname* with an alias called *Full Name.*
   Hire date should point to the last day in May or November of that year (NOT to the exact day) and be in the form like shown below with the heading *Start Date*. Do NOT use LIKE operator.
   You should display ONE row per output line by limiting the width of the *Full Name* to 25 characters. The output lines should look like this line:
   **174 Abel, Ellen SA_REP [May 31st of 1996]**

**ANS:**

```
SELECT employee_id, SUBSTR(last_name || ', ' || first_name,1,25) "Full Name",
job_id, TO_CHAR(LAST_DAY(hire_date),'"["Month ddth "of" YYYY"]"')"Start Date"
FROM employees WHERE TO_CHAR(hire_date,'MM') IN (05,11)
AND TO_CHAR(hire_date,'YY') NOT IN (94,95)
ORDER BY hire_date DESC
```




2. List the employee number, full name, job and the modified salary for all employees whose monthly earning (without this increase) is outside the range $6,000 – $11,000 and who are employed as Vice Presidents or Managers (President is not counted here).
   You should use Wild Card characters for this.

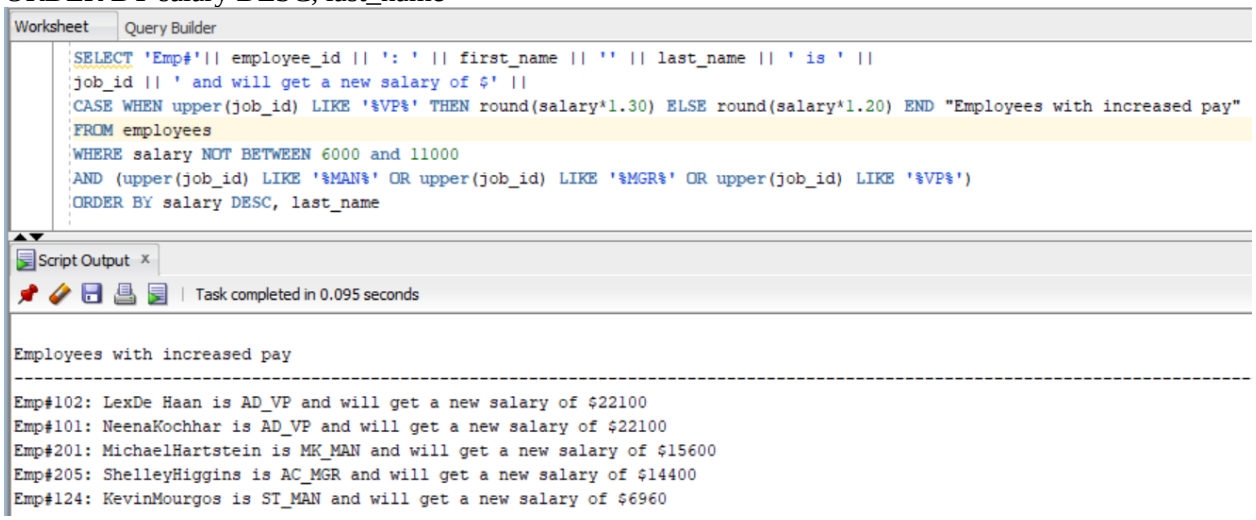
VP's will get 30% and managers 20% salary increase.
Sort the output by the top salaries (before this increase) firstly.
Heading will be like *Employees with increased Pay*

The output lines should look like this sample line:

**Emp# 124 : Kevin Mourgos is ST_MAN and will get a new salary of $6,960**

**ANS:**

```
SELECT 'Emp#'|| employee_id || ': ' || first_name || '' || last_name || ' is ' ||
job_id || ' and will get a new salary of $' ||
CASE WHEN upper(job_id) LIKE '%VP%' THEN round(salary*1.30) ELSE round(salary*1.20) END
"Employees with increased pay"
FROM employees
WHERE salary NOT BETWEEN 6000 and 11000
AND (upper(job_id) LIKE '%MAN%' OR upper(job_id) LIKE '%MGR%' OR upper(job_id) LIKE
'%VP%')
ORDER BY salary DESC, last_name
```

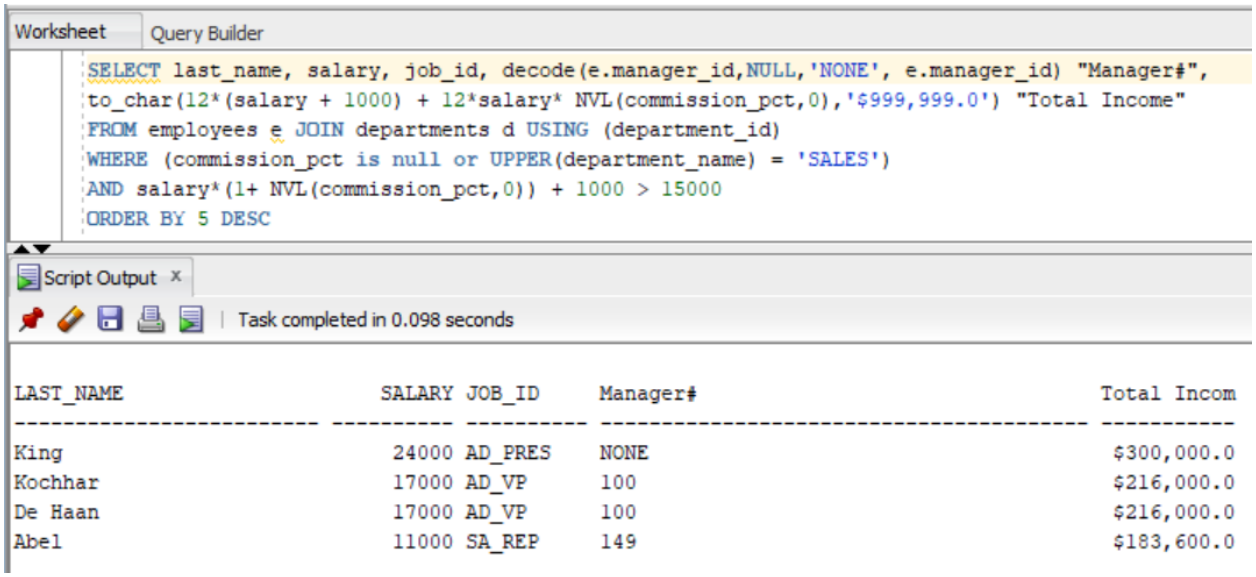



3. Display the employee last name, salary, job title and manager# of all employees not earning a commission OR if they work in SALES department, but only if their total monthly salary with $1000 included bonus and commission (if earned) is greater than $15,000.
Let's assume that all employees receive this bonus.
If an employee does not have a manager, then display the word NONE instead. This column should have an alias *Manager#.*
Display the Total annual salary as well in the form of $135,600.00 with the heading *Total Income*. Sort the result so that best paid employees are shown first. The output lines should look like this sample line:

**De Haan 17000 AD_VP 100 $216,000.0**

**ANS:**

```
SELECT last_name, salary, job_id, decode(e.manager_id,NULL,'NONE', e.manager_id) "Manager#",
to_char(12*(salary + 1000) + 12*salary* NVL(commission_pct,0),'$999,999.0') "Total Income"
FROM employees e JOIN departments d USING (department_id)
WHERE (commission_pct is null or UPPER(department_name) = 'SALES')
AND salary*(1+ NVL(commission_pct,0)) + 1000 > 15000
ORDER BY 5 DESC
```




4. Display Department_id, Job_id and the Lowest salary for this combination under the alias *Lowest Dept/Job Pay*, but only if that Lowest Pay falls in the range $6000 - $18000.
   Exclude people who work as some kind of *Representative* job from this query and departments IT and SALES as well.
   Sort the output according to the Department_id and then by Job_id.
   You MUST NOT use the Subquery method.

**ANS:**

```
SELECT E.department_id, E.job_id, MIN(E.salary) "Lowest Salary"
FROM EMPLOYEES E JOIN DEPARTMENTS D
ON E.department_id = D.department_id
WHERE
(E.job_id NOT LIKE '%REP'
AND LOWER(D.department_name) NOT IN ('it', 'sales')
AND (E.salary BETWEEN 6000 AND 18000))
GROUP BY E.department_id, E.job_id
ORDER BY E.department_id, E.job_id
```

```
SELECT E.department_id, E.job_id, MIN(E.salary) "Lowest Salary"
FROM EMPLOYEES E JOIN DEPARTMENTS D ON E.department_id = D.department_id
WHERE (E.job_id NOT LIKE '%REP' AND LOWER(D.department_name) NOT IN ('it', 'sales')
AND (E.salary BETWEEN 6000 AND 18000)) GROUP BY E.department_id, E.job_id ORDER BY E.department_id, E.job_id
```

Task completed in 0.097 seconds

| DEPARTMENT_ID | JOB_ID | Lowest Salary |
|---|---|---|
| 20 | MK_MAN | 13000 |
| 90 | AD_VP | 17000 |
| 110 | AC_ACCOUNT | 8300 |
| 110 | AC_MGR | 12000 |

5. Display last_name, salary and job for all employees who earn more than all lowest paid employees per department outside the US locations. Exclude President and Vice Presidents from this query.
Sort the output by job title ascending.
You need to use a Subquery.

**ANS:**

SELECT last_name, salary, job_id FROM EMPLOYEES JOIN DEPARTMENTS USING (department_id)
WHERE salary > ALL(SELECT MIN(salary)
FROM EMPLOYEES JOIN DEPARTMENTS USING (department_id) JOIN LOCATIONS USING (location_id)
WHERE country_id NOT IN UPPER(('US'))
GROUP BY department_id)AND(UPPER(job_id) NOT LIKE UPPER('%PRES')AND UPPER(job_id) NOT LIKE UPPER('%VP'))

```
SELECT last_name, salary, job_id FROM EMPLOYEES JOIN DEPARTMENTS USING (department_id)
WHERE salary > ALL(SELECT MIN(salary)
FROM EMPLOYEES JOIN DEPARTMENTS USING (department_id) JOIN LOCATIONS USING (location_id)
WHERE country_id NOT IN UPPER(('US'))
GROUP BY department_id)AND(UPPER(job_id) NOT LIKE UPPER('%PRES')AND UPPER(job_id) NOT LIKE UPPER('%VP'))
```

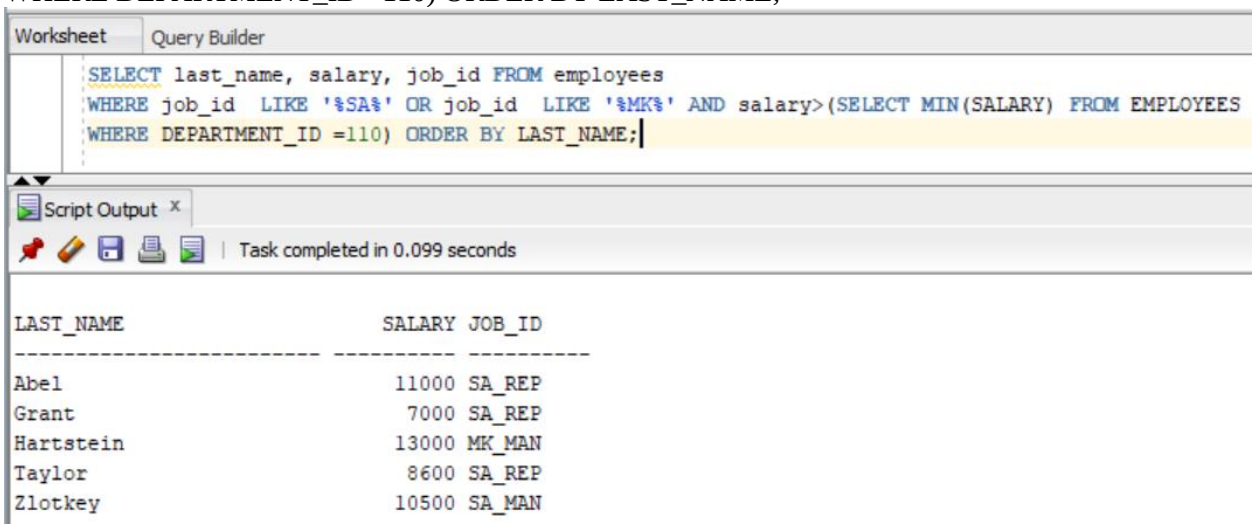
Task completed in 0.128 seconds

| LAST_NAME | SALARY | JOB_ID |
|---|---|---|
| Hunold | 9000 | IT_PROG |
| Zlotkey | 10500 | SA_MAN |
| Abel | 11000 | SA_REP |
| Higgins | 12000 | AC_MGR |
| Hartstein | 13000 | MK_MAN |

6. Who are the employees (show last_name, salary and job) who work either in IT or MARKETING department and earn more than the worst paid person in the ACCOUNTING department.
Sort the output by the last name alphabetically.
You need to use ONLY the Subquery method (NO joins allowed).

**ANS:**

```
SELECT last_name, salary, job_id FROM employees
WHERE job_id  LIKE '%SA%' OR job_id  LIKE '%MK%' AND salary>(SELECT MIN(SALARY)
FROM EMPLOYEES
WHERE DEPARTMENT_ID =110) ORDER BY LAST_NAME;
```

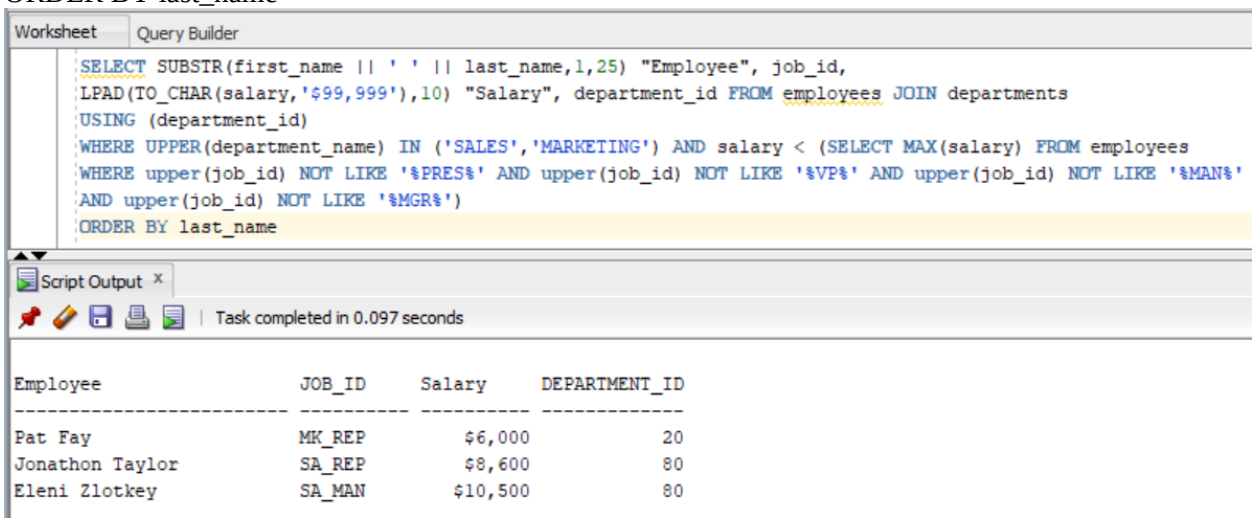



7. Display alphabetically the full name, job, salary (formatted as a currency amount incl. thousand separator, but no decimals) and department number for each employee who earns less than the best paid unionized employee (i.e. not the president nor any manager nor any VP), and who work in either SALES or MARKETING department.
Full name should be displayed as *Firstname Lastname* and should have the heading *Employee*. Salary should be left-padded with the **&** symbol till the width of 10 characters. It should have an alias *Salary*.
You should display ONE row per output line by limiting the width of the *Employee* to 25 characters.
The output lines should look like this sample line:

**Jonathon Taylor SA_REP &&&& $8,600 80**

**ANS:**

```
SELECT SUBSTR(first_name || ' ' || last_name,1,25) "Employee", job_id,
LPAD(TO_CHAR(salary,'$99,999'),10) "Salary", department_id FROM employees JOIN departments
USING (department_id)
WHERE UPPER(department_name) IN ('SALES','MARKETING') AND salary < (SELECT
MAX(salary) FROM employees
WHERE upper(job_id) NOT LIKE '%PRES%' AND upper(job_id) NOT LIKE '%VP%' AND
upper(job_id) NOT LIKE '%MAN%'
AND upper(job_id) NOT LIKE '%MGR%')
ORDER BY last_name
```




8. "Tricky One"

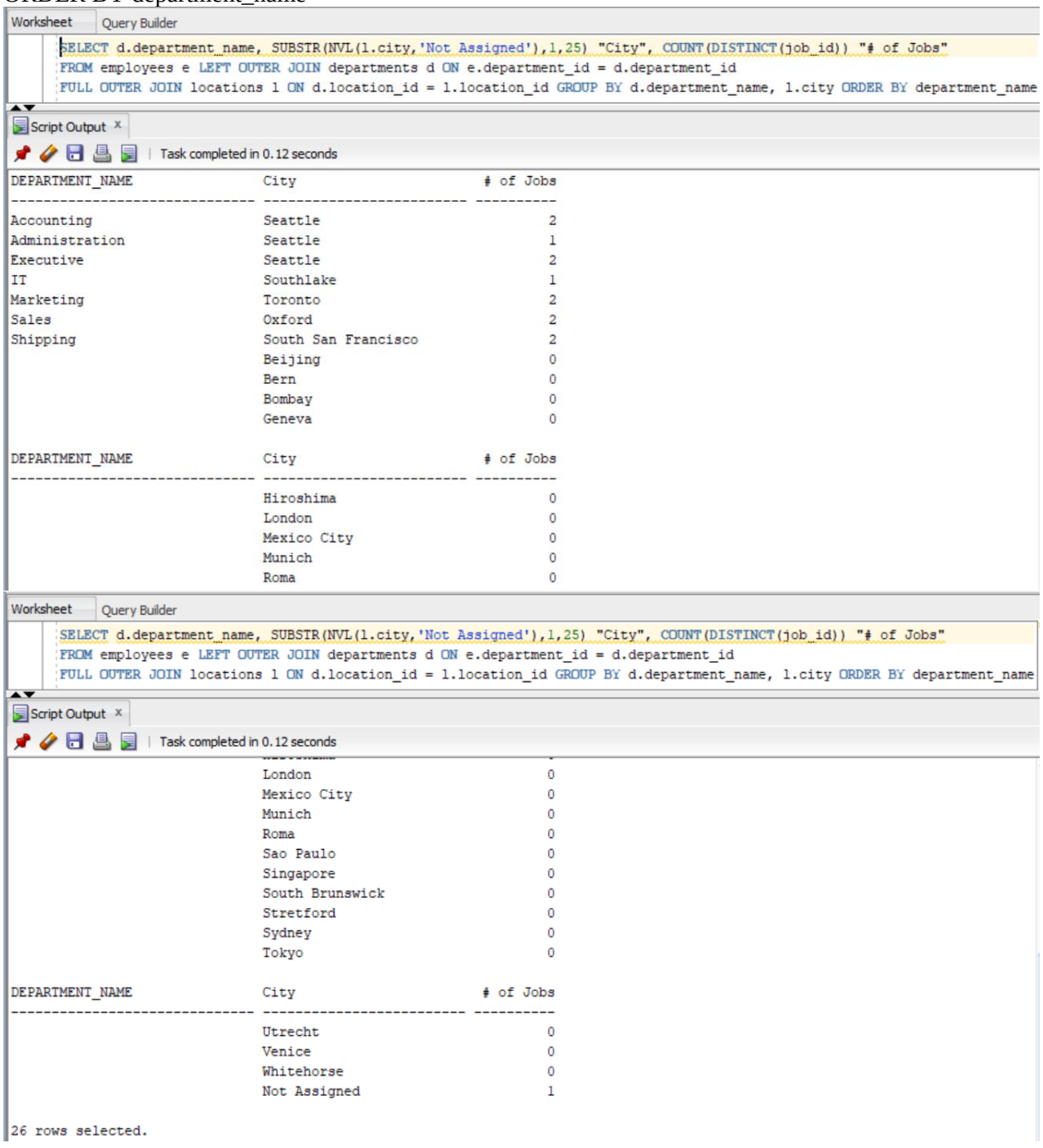
Display department name, city and number of different jobs in each department. If city is null, you should print *Not Assigned Yet.* This column should have alias *City.*
Column that shows # of different jobs in a department should have the heading *# of Jobs*
You should display ONE row per output line by limiting the width of the *City* to 25 characters.

You need to show complete situation from the EMPLOYEE point of view, meaning include also employees who work for NO department (but do NOT display empty departments) and from the CITY point of view meaning you need to display all cities without departments as well.

You need to use Join method.

**ANS:**

```
SELECT d.department_name, SUBSTR(NVL(l.city,'Not Assigned'),1,25) "City",
COUNT(DISTINCT(job_id)) "# of Jobs"
FROM employees e LEFT OUTER JOIN departments d ON e.department_id = d.department_id
FULL OUTER JOIN locations l ON d.location_id = l.location_id GROUP BY d.department_name, l.city
ORDER BY department_name
```



Worksheet | Query Builder

```
SELECT d.department_name, SUBSTR(NVL(l.city,'Not Assigned'),1,25) "City", COUNT(DISTINCT(job_id)) "# of Jobs"
FROM employees e LEFT OUTER JOIN departments d ON e.department_id = d.department_id
FULL OUTER JOIN locations l ON d.location_id = l.location_id GROUP BY d.department_name, l.city ORDER BY department_name
```

Script Output | Task completed in 0.12 seconds

```
DEPARTMENT_NAME               City                      # of Jobs
----------------------------- ------------------------- ----------
Accounting                    Seattle                            2
Administration                Seattle                            1
Executive                     Seattle                            2
IT                            Southlake                          1
Marketing                     Toronto                            2
Sales                         Oxford                             2
Shipping                      South San Francisco                2
                              Beijing                            0
                              Bern                               0
                              Bombay                             0
                              Geneva                             0

DEPARTMENT_NAME               City                      # of Jobs
----------------------------- ------------------------- ----------
                              Hiroshima                          0
                              London                             0
                              Mexico City                        0
                              Munich                             0
                              Roma                               0
```

Worksheet | Query Builder

```
SELECT d.department_name, SUBSTR(NVL(l.city,'Not Assigned'),1,25) "City", COUNT(DISTINCT(job_id)) "# of Jobs"
FROM employees e LEFT OUTER JOIN departments d ON e.department_id = d.department_id
FULL OUTER JOIN locations l ON d.location_id = l.location_id GROUP BY d.department_name, l.city ORDER BY department_name
```

Script Output | Task completed in 0.12 seconds

```
                              London                             0
                              Mexico City                        0
                              Munich                             0
                              Roma                               0
                              Sao Paulo                          0
                              Singapore                          0
                              South Brunswick                    0
                              Stretford                          0
                              Sydney                             0
                              Tokyo                              0

DEPARTMENT_NAME               City                      # of Jobs
----------------------------- ------------------------- ----------
                              Utrecht                            0
                              Venice                             0
                              Whitehorse                         0
                              Not Assigned                       1

26 rows selected.
```