**a1-jschoi8-DBS301 Assignment#1**

**Due: BEFORE Monday of week 7**

SUBMISSION:

1 Change the file to read a1-your email id --- only use one of the members. Exmple: a1-nemo17

2 Put the SQL(using good format) and the answer after the question

3 Email the result to me and CC all member in the group

4 Not doing the above will result in a 25% deduction

This assignment can be done in groups. Ideal might be 2 to 4, but is not limited to that number.

It is suggested that you ALL do it individually and then meet to compare answers. Those not doing the work you bar from your group.

**VERY IMPORTANT:**

Being part of a group is the same as being a part of a team for these assignments. When you submitted your work as part of a group you are saying that:

you understood what was submitted and that you fully participated with ALL the group members.

It does not mean letting others do your work for you. It does not mean watching the others do the work. For your full participation, you get a mark equal to all the others in the group. If on the test, which is very much like the assignment, you cannot answer it strongly indicates that you didn’t participate and understand the assignment but depended on others for the mark you received. That is very much like submitting their work and claiming it is your work.

Members in group

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Student ID | Email address | Oracle ID |
| Jungsoo Choi | 134708155 | jschoi8@myseneca.ca | dbs301\_172b06 |
| Kerry Mayers | 118432160 | kmayers1@myseneca.ca | dbs301\_172b10 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Submit the SQL and the results of the SQL after each question

The SQL must be written in good format to get any marks.

1 Display the (a) employee number, (b) full employee name, (c) job and (d) hire date.

- Limit the display to all employees hired in May or November of any year.

- The most recently hired employees are displayed first.

- 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)

- The format is in the form of *May 31st of 1996* – note there is no big gap between month and 31st

- The hire date column should be called *Start Date*.

NOTE: Do NOT use a 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:

|  |  |  |  |
| --- | --- | --- | --- |
| **EMPLOYEE\_ID** | **Full Name** | **JOB\_ID** | **Start Date** |
| 174 | Abel, Ellen | SA\_REP | May 31st of 1996 |

**SELECT EMPLOYEE\_ID AS "Employee number",**

**SUBSTR(LAST\_NAME || ', ' || FIRST\_NAME, 1, 25) AS "Full Name",**

**JOB\_ID AS "Job",**

**TO\_CHAR(LAST\_DAY(HIRE\_DATE), 'fmMonth ddth "of" YYYY') AS "Start Date"**

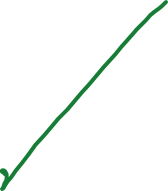
**FROM EMPLOYEES**

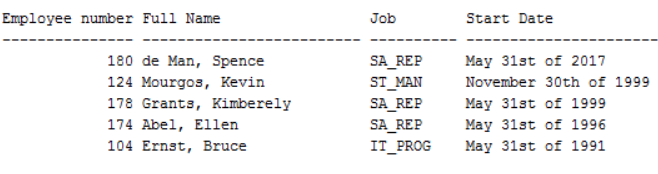
**WHERE TO\_CHAR(HIRE\_DATE, 'MM') IN (05, 11)**

**AND TO\_CHAR(HIRE\_DATE, 'YY') NOT IN (94, 95)**

**ORDER BY HIRE\_DATE DESC;**

Result:





2.List the employee number, full name, job and the modified salary for all employees

- whose monthly earning (without the increase) is outside the range $6,000 – $11,000

- and who are employed as a Vice Presidents or Managers (President is not counted here).

- You should use Wild Card characters for this.

- the modified salary for a VP will be 30% higher

- and managers a 20% salary increase.

- Sort the output by the top salaries (before this increase).

Heading will be: 🡪 *Employees with increased Pay*

The output lines should look like this sample line:

|  |
| --- |
| Emp# 124 named Kevin Mourgos who is ST\_MAN will have a new salary of $6960 |

**SELECT q'[Emp# ]' || EMPLOYEE\_ID ||**

**q'[ named ]' || LAST\_NAME || ' ' || FIRST\_NAME ||**

**q'[ who is ]' || JOB\_ID ||**

**q'[ will have a new salary of $ ]' ||**

**CASE**

**WHEN JOB\_ID LIKE '%VP'**

**THEN (SALARY \* (130/100))**

**WHEN JOB\_ID LIKE '%MGR'**

**THEN (SALARY \* (120/100))**

**END AS "Employees with increased Pay"**

**FROM EMPLOYEES**



**WHERE JOB\_ID LIKE ('%VP')**

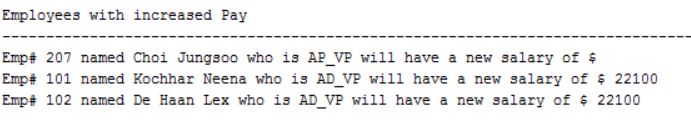
**OR JOB\_ID LIKE ('%MGR')**

**AND SALARY BETWEEN 6000 AND 11000**

**ORDER BY SALARY DESC;**



Result:

****



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 a $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 in the form of $135,600.00

- with the heading 🡪 *Total Income*.

- Sort the result so that the best paid employees are shown first.

The output lines should look like this sample line:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| De Haan | 17000 | AD\_VP | 100 | $216,000.00 |

**SELECT LAST\_NAME AS "Last name",**

**SALARY AS "Salary",**

**JOB\_ID AS "Job title",**

**NVL(TO\_CHAR(MANAGER\_ID), 'NONE'),**

**TO\_CHAR(ROUND(((SALARY + 1000) \* 12), 2), '$999,999.00') AS "Total Income"**

**FROM EMPLOYEES**



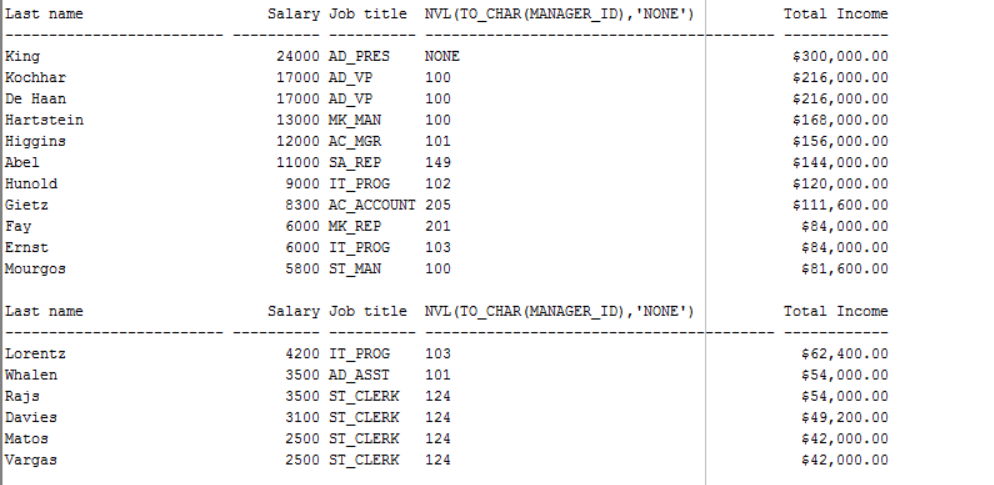
**WHERE COMMISSION\_PCT IS NULL**

**OR JOB\_ID LIKE 'SA\_%'**

**AND (SALARY + 1000 + (SALARY \* NVL((COMMISSION\_PCT), 0))) > 15000**

**ORDER BY 5 DESC;**

Result:



Should be 4 only

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LAST\_NAME** | **SALARY** | **JOB\_ID** | **Manager#** | **Total Income** |
| King | 24000 | AD\_PRES | NONE | $300,000.00 |
| Kochhar | 17000 | AD\_VP | 100 | $216,000.00 |
| De Haan | 17000 | AD\_VP | 100 | $216,000.00 |
| Abel | 11000 | SA\_REP | 149 | $183,600.00 |



4.Display Department\_id, Job\_id and the Lowest salary for this combination but only if that Lowest Pay falls in the range $6000 - $18000.

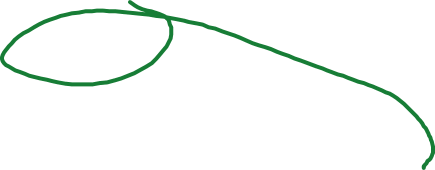
Exclude people who

(a) work as some kind of *Representative* job from this query and

(b) departments IT and SALES

Sort the output according to the Department\_id and then by Job\_id.

You MUST NOT use the Subquery method.



**SELECT DEPARTMENT\_ID AS "Department Number",**

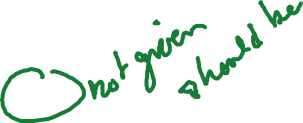
**JOB\_ID AS "Job Title",**

**MIN(SALARY) AS "Salary"**

**FROM EMPLOYEES**



**WHERE SALARY BETWEEN 6000 AND 18000**



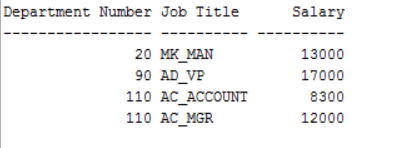
**AND JOB\_ID NOT LIKE '%REP'**

**AND DEPARTMENT\_ID NOT IN (80, 60)**

**GROUP BY DEPARTMENT\_ID, JOB\_ID**

**ORDER BY DEPARTMENT\_ID, JOB\_ID;**

Result:



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 and Joining with the NEWER method. (USING/JOIN)

**SELECT LAST\_NAME AS "Last name",**

**SALARY AS "Salary",**

**JOB\_ID AS "Job"**

**FROM EMPLOYEES**

**WHERE SALARY > ALL**

**(SELECT MIN(SALARY)**

**FROM EMPLOYEES**

**JOIN DEPARTMENTS USING (DEPARTMENT\_ID)**

**JOIN LOCATIONS USING (LOCATION\_ID)**

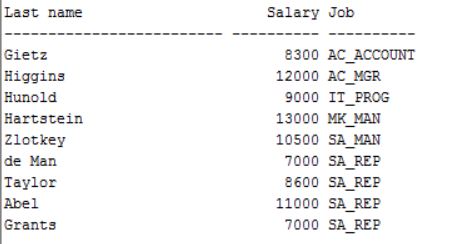
**WHERE COUNTRY\_ID NOT IN ('US'))**

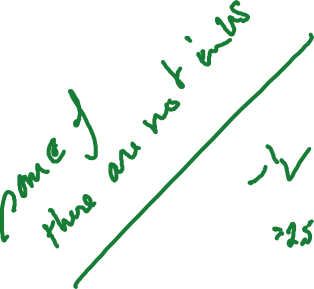
**AND JOB\_ID NOT LIKE ('%VP')**

**AND JOB\_ID NOT LIKE ('%PRES')**

**ORDER BY JOB\_ID;**

Result:

****



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. NOTE: you are not given department numbers, so you can’t use them in place of department names.

You need to use ONLY the Subquery method (NO joins allowed).

**SELECT LAST\_NAME AS "Last name",**

**SALARY AS "Salary",**

**JOB\_ID AS "Job"**

**FROM EMPLOYEES**

**WHERE DEPARTMENT\_ID IN (20, 60)**

**AND SALARY >**

**(SELECT MIN(SALARY)**

**FROM EMPLOYEES**

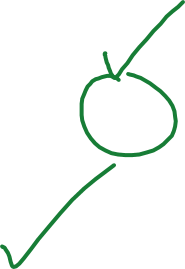
**WHERE DEPARTMENT\_ID =**

**(SELECT DEPARTMENT\_ID**

**FROM DEPARTMENTS**

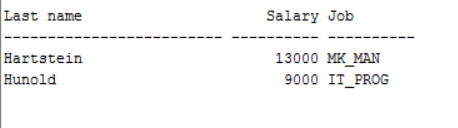
**WHERE DEPARTMENT\_NAME = 'Accounting')**

**)**



**ORDER BY LAST\_NAME;**

Result:



7.Display alphabetically the last name, job, salary, department number for each employee who earns less than the best paid unionized employee

- What is a unionizes employee? Unionized employees are not a manager, presidents or vice president

And do not work in SALES or MARKETING departments

-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 12 characters. It should have an alias *Salary.*

- salary is formatted as a currency amount incl. thousand separators, but no decimals and no $ sign

- display in order of employee last name

The output lines should look like this sample line(not exactly, but close)

|  |  |  |  |
| --- | --- | --- | --- |
| Taylor FirstName | SA\_REP | ===== 8,600 | 80 |

**SELECT**

**INITCAP(LAST\_NAME || ' ' || FIRST\_NAME) AS "Employee",**

**JOB\_ID AS "Job",**

**LPAD(TO\_CHAR(SALARY, '99,999'), 12, '=') AS "Salary",**

**DEPARTMENT\_ID AS "Department number"**



**FROM EMPLOYEES**



**WHERE SALARY <**

**(SELECT MAX(SALARY)**

**FROM EMPLOYEES**

**WHERE DEPARTMENT\_ID NOT IN (80, 20)**

**AND JOB\_ID NOT LIKE '%MGR'**

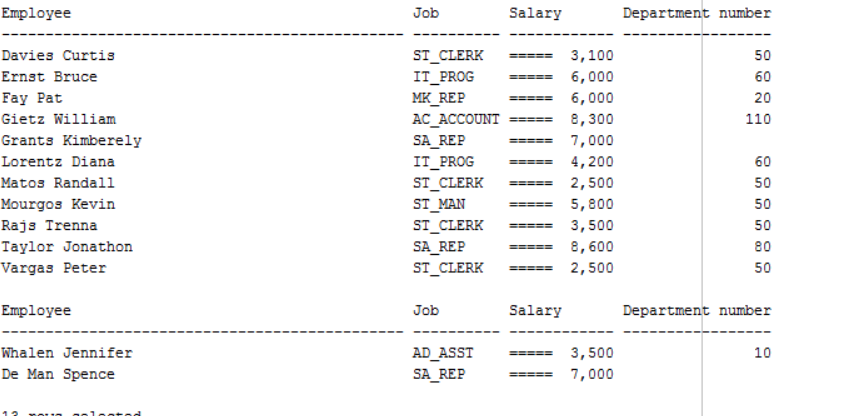
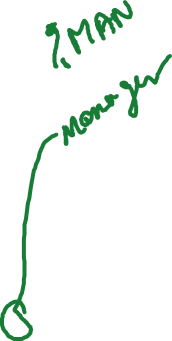
**AND JOB\_ID NOT LIKE '%VP'**

**AND JOB\_ID NOT LIKE '%PRES'**

**)**

**ORDER BY LAST\_NAME;**

Result:



8. DIFFICULT problem

Display department name, city and number of different jobs in each department.

- If city is null, you should print *No Assigned City.*

*-* This column should have alias *City.*

Column that shows # of different jobs in a department should have the heading *# of Jobs*

Limit the width of the *City* to 20 characters.

NOTE: 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 Outer Joining with the NEWER (Oracle) method (not the + symbol method)

**SELECT DEPARTMENT\_NAME AS "Department name",**

**SUBSTR(NVL(CITY, 'No Assigned City'),1,20) AS "City",**

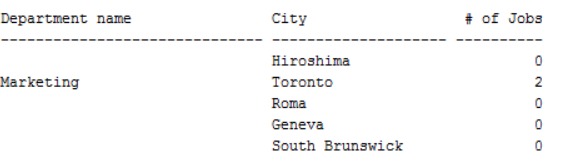
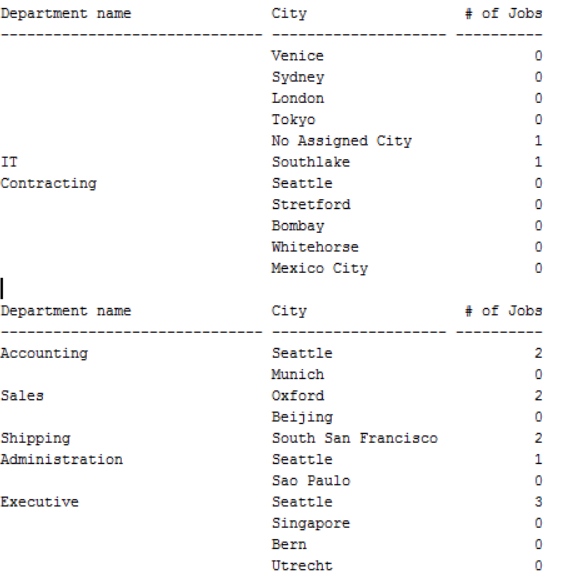
**COUNT(DISTINCT(JOB\_ID)) AS "# of Jobs"**

**FROM LOCATIONS LEFT JOIN DEPARTMENTS USING (LOCATION\_ID)**

**FULL JOIN EMPLOYEES USING (DEPARTMENT\_ID)**

**GROUP BY DEPARTMENT\_NAME, CITY;**

Result:



9. . Write a query that displays the employee’s Full Name and Job Title in the following format:

*CURTIES David is Store Clerk 🡸 requires full job title*

- only employees whose last name ends with *S* and first name starts with *C* or *K*.

- Sort the result by the employees’ last names.

**SELECT UPPER(FIRST\_NAME) || ' ' || INITCAP(LAST\_NAME) || ' is ' ||**

**CASE JOB\_ID**



**WHEN 'ST\_CLERK'**

**THEN 'Store Clerk'**

**WHEN 'SA\_REP'**

**THEN 'Sales Representative'**

**WHEN 'ST\_MAN'**

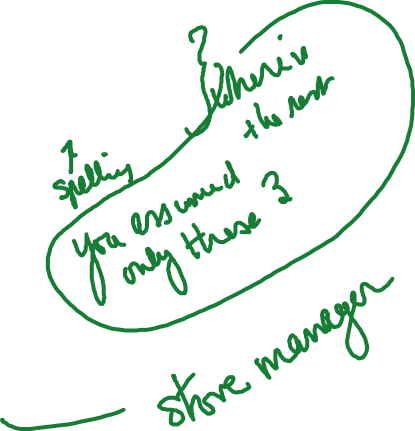
**THEN 'Staion Man'**

**END AS "Job Title"**

**FROM EMPLOYEES**

**WHERE FIRST\_NAME LIKE 'C%'**

**OR FIRST\_NAME LIKE 'K%'**



**AND LAST\_NAME LIKE '%s'**

**ORDER BY LAST\_NAME;**

Result:

