**a1-20163-DBS301 Assignment#1**

**Due: BEFORE Monday of week 7**

SUBMISSION:

1 Change the file to read a1-your email id --- of one of the members

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

3 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 meat to compare.

**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 doesd 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 well then 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 |
| Eunsol Lee | 029-557-154 | elee111@myseneca.ca | dbs301\_163c20 |
| Dariia Mietlova | 113-923-155 | dmietlova@myseneca.ca | dbs301\_163c23 |
| Aleksandr Sledkov | 111-034-153 | asledkov@myseneca.ca | dbs301\_163d14 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

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 |

SQL:

SELECT employee\_id,

CAST(last\_name || ', ' || first\_name AS CHAR(25)) "Full Name",

job\_id,

TO\_CHAR(LAST\_DAY(hire\_date), 'fmMonth ddth "of" YYYY ') "Start Date"

FROM employees

WHERE (SUBSTR(hire\_date, 4, 3) = 'MAY' OR SUBSTR(hire\_date, 4, 3) = 'NOV')

AND (SUBSTR(hire\_date, -2, 2) != 94 AND SUBSTR(hire\_date, -2, 2) != 95);

* WHERE TO\_CHAR(hire\_date, 'MM') IN (05,11)
* AND TO\_CHAR(hire\_date, 'YY') NOT IN (94,95)

ORDER BY hire\_date DESC;

OUTPUT:

EMPLOYEE\_ID Full Name JOB\_ID Start Date

------------------ ----------------------- ---------- ------------------

124 Mourgos, Kevin ST\_MAN November 30th of 1999

178 Grant, Kimberely SA\_REP May 31st of 1999

174 Abel, Ellen SA\_REP May 31st of 1996

104 Ernst, Bruce IT\_PROG May 31st of 1991

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 |

SQL:

SELECT 'Emp# ' || employee\_id || ' named ' || first\_name

|| ' ' || last\_name || ' who is ' || job\_id ||

' will have a new salary of $' ||

CASE WHEN job\_id LIKE '%VP' THEN salary \* 1.3

WHEN job\_id LIKE '%MGR' THEN salary \* 1.2

WHEN job\_id LIKE '%MAN' THEN salary \* 1.2

END

"*Employees with increased Pay*"

FROM employees

WHERE (salary NOT BETWEEN 6000 AND 11000)

AND (job\_id LIKE '%VP' OR job\_id LIKE '%MGR' OR job\_id LIKE '%MAN')

ORDER BY salary DESC;

OUTPUT:

Employees with increased Pay

------------------------------------------------------------------------------

Emp# 101 named Neena Kochhar who is AD\_VP will have a new salary of $22100

Emp# 102 named Lex De Haan who is AD\_VP will have a new salary of $22100

Emp# 201 named Michael Hartstein who is MK\_MAN will have a new salary of $15600

EMP# 205 named Shelley Higgins who is AC\_MGR will have a new salary of $14400

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

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 |

SQL:

SELECT last\_name, salary, job\_id,

nvl(TO\_CHAR(e.manager\_id), TO\_CHAR('NONE')) "Manager#",

TO\_CHAR((salary + 1000 + NVL2(commission\_pct, commission\_pct \* salary, 0)) \* 12, '$999,999.00') "Total Income"

FROM employees e JOIN departments USING(department\_id)

WHERE (commission\_pct IS NULL OR department\_name = 'Sales')

AND (salary + 1000 + NVL2(commission\_pct, commission\_pct \* salary, 0)) > 15000

ORDER BY 5 DESC;

OUTPUT:

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.

SQL:

SELECT department\_id, job\_id, MIN(salary) "Lowest Pay"

FROM employees

JOIN departments USING(department\_id)

WHERE department\_name NOT IN('IT', 'Sales')

AND job\_id NOT LIKE '%REP'

GROUP BY department\_id, job\_id

HAVING MIN(salary) BETWEEN 6000 AND 18000

ORDER BY department\_id, job\_id;

OUTPUT:

DEPARTMENT\_ID JOB\_ID Lowest Pay

---------------------- ---------- ----------------------

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

SQL:

SELECT last\_name, salary, job\_id

FROM employees

WHERE salary > ALL (

SELECT MIN(salary)

FROM employees

JOIN departments USING(department\_id)

JOIN locations USING(location\_id)

WHERE country\_id != 'US'

GROUP BY department\_id )

AND (job\_id NOT LIKE '%VP' AND job\_id NOT LIKE '%PRES')

ORDER BY job\_id;

OUTPUT:

LAST\_NAME SALARY JOB\_ID

------------------------- ---------------------- ----------

Higgins 12000 AC\_MGR

Hunold 9000 IT\_PROG

Hartstein 13000 MK\_MAN

Zlotkey 10500 SA\_MAN

Abel 11000 SA\_REP

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).

SQL:

SELECT last\_name, salary, job\_id

FROM employees

WHERE department\_id IN (

SELECT department\_id

FROM departments

WHERE department\_name IN ('IT', 'Marketing')

)

AND salary > (

SELECT MIN(salary)

FROM employees

WHERE department\_id = (

SELECT department\_id

FROM departments

WHERE department\_name = 'Accounting'

)

)

ORDER BY last\_name;

OUTPUT:

LAST\_NAME SALARY JOB\_ID

------------------------- ---------------------- ----------

Hartstein 13000 MK\_MAN

Hunold 9000 IT\_PROG

7.Display the**first name, last**name, job, salary,  and department number for each employee who earns less than the best paid unionized 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 separator, 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**)

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

SQL:

SELECT CAST(first\_name || ' ' || last\_name AS CHAR(20))"Employee",

job\_id,

LPAD(TO\_CHAR(salary, '99,999'), 12, '=') "Salary",

department\_id

FROM employees

WHERE salary < (

SELECT MAX(salary)

FROM employees JOIN departments USING(department\_id)

WHERE (job\_id NOT LIKE '%MAN'

AND job\_id NOT LIKE '%VP'

AND job\_id NOT LIKE '%PRES')

AND department\_name NOT IN ('Sales', 'Marketing')

)

ORDER BY last\_name;

OUTPUT:

Employee JOB\_ID Salary DEPARTMENT\_ID

-------------------- ---------- ------------ ----------------------

Ellen Abel SA\_REP ===== 11,000 80

Curtis Davies ST\_CLERK ===== 3,100 50

Bruce Ernst IT\_PROG ===== 6,000 60

Pat Fay MK\_REP ===== 6,000 20

William Gietz AC\_ACCOUNT ===== 8,300 110

Kimberely Grant SA\_REP ===== 7,000

Alexander Hunold IT\_PROG ===== 9,000 60

Diana Lorentz IT\_PROG ===== 4,200 60

Randall Matos ST\_CLERK ===== 2,500 50

Kevin Mourgos ST\_MAN ===== 5,800 50

Trenna Rajs ST\_CLERK ===== 3,500 50

Jonathon Taylor SA\_REP ===== 8,600 80

Peter Vargas ST\_CLERK ===== 2,500 50

Jennifer Whalen AD\_ASST ===== 2,500 10

Eleni Zlotkey SA\_MAN ===== 10,500 80

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 25 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)

SQL:

SELECT department\_name,

CAST(NVL(city, 'No Assigned City') AS CHAR(25)) "City",

count(DISTINCT job\_id) "# of Jobs"

FROM departments

RIGHT JOIN employees USING(department\_id)

FULL JOIN locations USING(location\_id)

GROUP BY department\_name, CAST(NVL(city, 'No Assigned City') AS CHAR(25));

OUTPUT:

DEPARTMENT\_NAME City # of Jobs

------------------------------ ------------------------- ---------------------

Shipping South San Francisco 2

Whitehorse 0

Stretford 0

Bern 0

Executive Seattle 3

Singapore 0

IT Southlake 1

Marketing Toronto 2

Venice 0

Hiroshima 0

Administration Seattle 1

Sales Oxford 2

Roma 0

South Brunswick 0

London 0

Geneva 0

Accounting Seattle 2

Beijing 0

Mexico City 0

Sao Paulo 0

Utrecht 0

No Assigned City 1

Tokyo 0

Bombay 0

Sydney 0

Munich 0

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

*DAVIES, CURTIES is Store Clerk*

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

- Sort the result by the employees’ last names.

SQL:

SELECT UPPER(last\_name) || ', ' || UPPER(first\_name) || ' is ' ||

CASE job\_id WHEN 'ST\_CLERK' THEN 'Store Clerk'

WHEN 'ST\_MAN' THEN 'Store Manager'

END "Full Name and Job Title"

FROM employees

WHERE UPPER(last\_name) LIKE '%S'

AND (UPPER(first\_name) LIKE 'C%' OR UPPER(first\_name) LIKE 'K%')

ORDER BY last\_name;

OUTPUT:

Full Name and Job Title

-----------------------------------------------------------------------------

DAVIES, CURTIS is Store Clerk

MOURGOS, KEVIN is Store Manager