

DBS311 Assignment#1

Group Members:

Member 1: James Giroux – 129198164 – jgiroux1@myseneca.ca

Member 2: Soni Dev Alpeshbhai – 130759210 – dasoni4@myseneca.ca

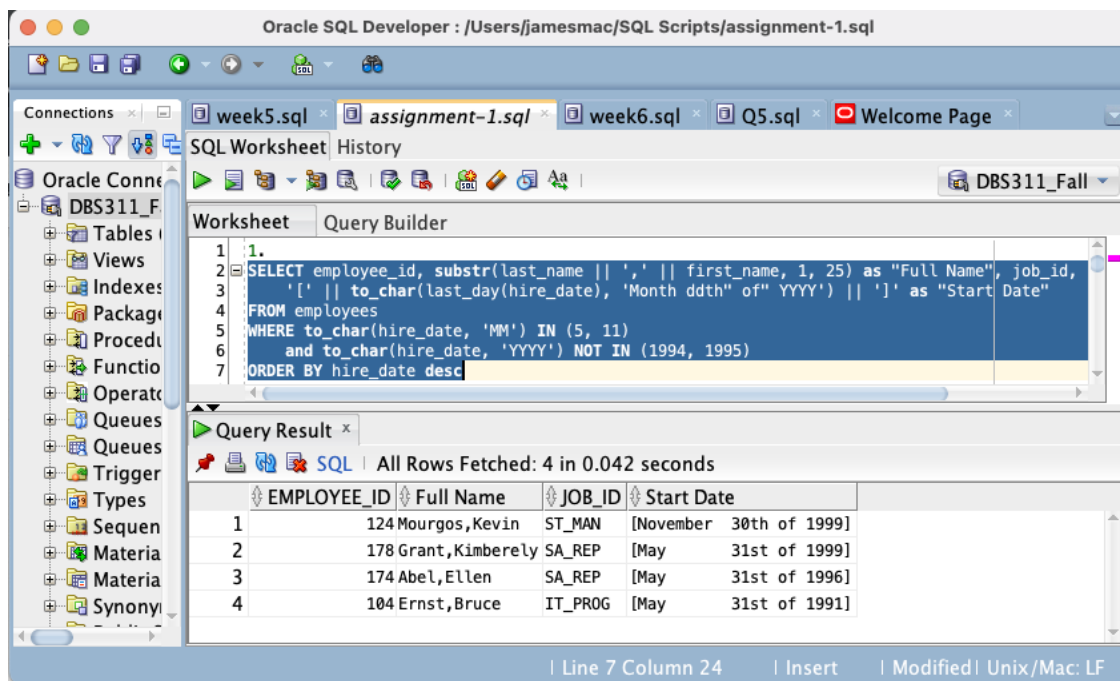
Member 3: Le Chanh Tin Luong – 154574214 – luong7@myseneca.ca

Section: NJJ

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]



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

employees whose monthly earnings (without this increase) are 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.

VPs 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

The screenshot shows the Oracle SQL Developer interface. The 'SQL Worksheet' pane contains the following SQL query:

```
2.
SELECT 'EMP# ' || employee_id || ' : ' || first_name || ' ' || last_name || ' is ' || job_id ||
' and will get a new salary of $' ||
CASE WHEN job_id like '%vp'
THEN to_char(round(salary * 1.3), 'FM999,999')
ELSE to_char(round(salary * 1.2), 'FM999,999')
END as "Employees with pay increase"
FROM employees
WHERE job_id in (
SELECT job_id
FROM jobs
WHERE job_title like('%Manager') or job_title like('%Vice_President'))
and salary NOT BETWEEN 6000 and 11000
ORDER BY salary desc
```

The 'Query Result' pane shows the output of the query, titled 'Employees with pay increase'. It contains 5 rows of data:

EMP#	Last Name	First Name	Job Title	New Salary
101	Neena	Kochhar	AD_VP	\$20,400
102	Lex	De Haan	AD_VP	\$20,400
201	Michael	Hartstein	MK_MAN	\$15,600
205	Shelley	Higgins	AC_MGR	\$14,400
124	Kevin	Mourgos	ST_MAN	\$6,960

3. Display the employee's last name, salary, job title and manager# of all employees not earning a commission OR if they work in the 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 the best-paid employees are shown first.

The output lines should look like this sample line:

De Haan 17000 AD_VP 100

\$216,000.0

The screenshot shows the Oracle SQL Developer interface. The title bar indicates the file path: /Users/jamesmac/SQL Scripts/assignment-1.sql. The main window is divided into several panes. On the left is the 'Connections' pane showing a connection to 'DBS311_Fall'. Below it is the 'Object Explorer' showing a tree view of database objects including Tables, Views, Indexes, Packages, Procedures, Functions, Operators, Queues, Triggers, Types, Sequences, and Materialized Views. The central pane is the 'SQL Worksheet', which contains a SQL query. The query is as follows:

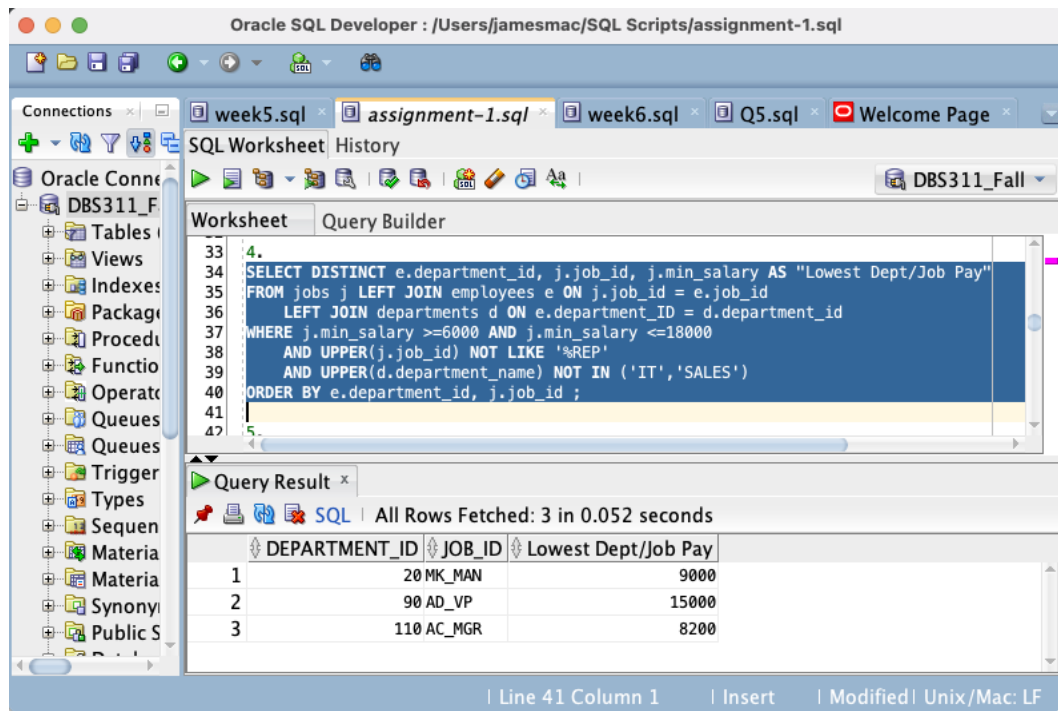
```
3.
SELECT last_name, salary, job_id, decode(e.manager_id, (null), 'NONE', e.manager_id) as "Manager#",
       to_char(12 * (salary + 1000) + 12 * salary * (nvl(commission_pct, 0)), '$999,999.9') as "Total Income"
FROM employees e JOIN departments USING (department_id)
WHERE (department_name = 'Sales' or commission_pct is null)
   and salary*(nvl(commission_pct,0)+1) > 15000
ORDER BY "Total Income" desc
```

Below the SQL Worksheet is the 'Query Result' pane, which displays the results of the query. It shows that 3 rows were fetched in 0.039 seconds. The results are as follows:

	LAST_NAME	SALARY	JOB_ID	Manager#	Total Income
1	King	24000	AD_PRES	NONE	\$300,000.0
2	Kochhar	17000	AD_VP	100	\$216,000.0
3	De Haan	17000	AD_VP	100	\$216,000.0

The status bar at the bottom indicates the current cursor position: Line 31 Column 29, and the status: Insert, Modified, Unix/Mac: LF.

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.



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.

The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL worksheet with the following query:

```

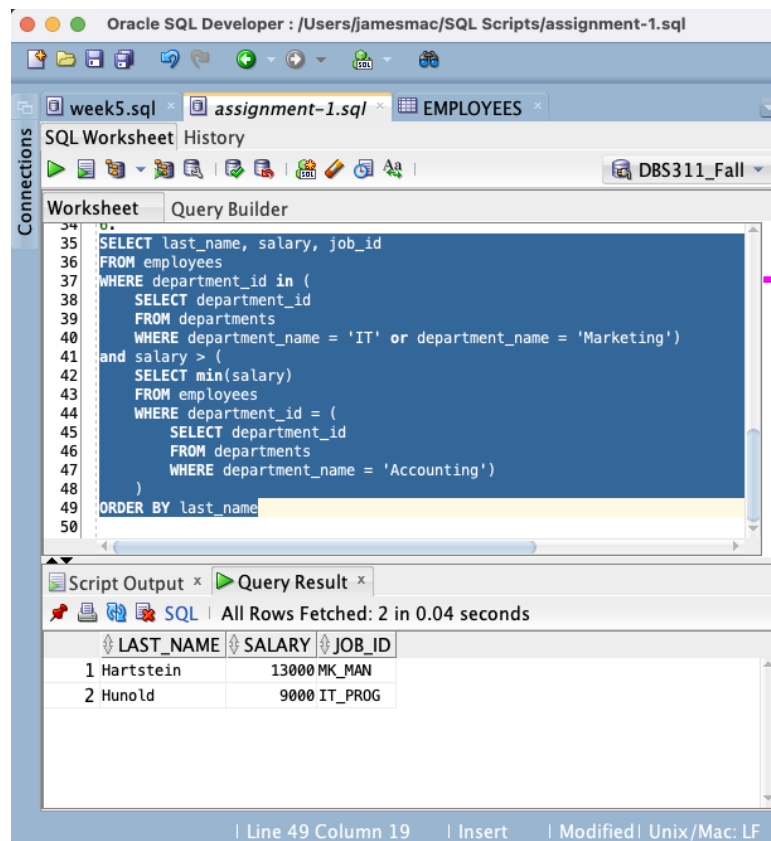
43 SELECT last_name, salary, job_title
44 FROM employees e
45 LEFT JOIN jobs j ON e.job_id = j.job_id
46 WHERE salary > ANY(SELECT MIN(salary)
47 FROM employees
48 GROUP BY department_id
49 HAVING department_id NOT IN
50 (SELECT d.department_id
51 FROM departments d LEFT JOIN locations l
52 ON d.location_id = l.location_id
53 WHERE UPPER(l.country_id) <> 'US'))
54 AND e.job_id NOT IN ('AD_PRES', 'AD_VP')
55 ORDER BY job_title
56

```

The query results are displayed in the Script Output window, showing 16 rows fetched in 0.042 seconds. The results are sorted by job title.

	LAST_NAME	SALARY	JOB_TITLE
1	Higgins	12000	Accounting Manager
2	Whalen	4400	Administration Assistant
3	Hartstein	13000	Marketing Manager
4	Fay	6000	Marketing Representative
5	Hunold	9000	Programmer
6	Lorentz	4200	Programmer
7	Ernst	6000	Programmer
8	Gietz	8300	Public Accountant
9	Zlotkey	10500	Sales Manager
10	Abel	11000	Sales Representative
11	Grant	7000	Sales Representative
12	Taylor	8600	Sales Representative
13	Davies	3100	Stock Clerk
14	Matos	2600	Stock Clerk
15	Rajs	3500	Stock Clerk
16	Mourgos	5800	Stock Manager

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).



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.

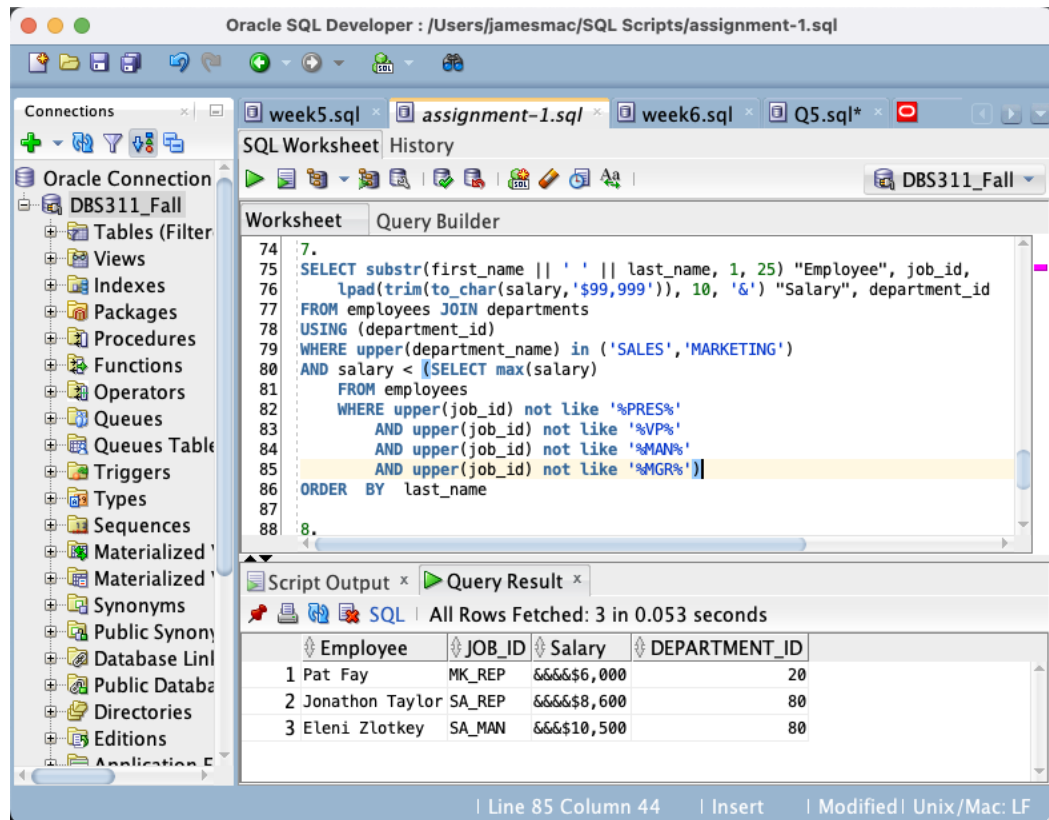
The output lines should look like this [sample line](#):

Jonathon Taylor

SA_REP

&&&& \$8,600

80



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.

Oracle SQL Developer : /Users/jamesmac/SQL Scripts/assignment-1.sql

Connections x week5.sql x assignment-1.sql x week6.sql x ...

SQL Worksheet History

0.047 seconds

Worksheet Query Builder

```

8.
SELECT d.department_name,
       SUBSTR(NVL(l.city,'Not Assigned Yet'),1,25) as City,
       COUNT(DISTINCT(job_id)) as "# 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 x Query Result x

SQL | All Rows Fetched: 26 in 0.043 seconds

	DEPARTMENT_NAME	CITY	# of Jobs
1	Accounting	Seattle	2
2	Administration	Seattle	1
3	Executive	Seattle	2
4	IT	Southlake	1
5	Marketing	Toronto	2
6	Sales	Oxford	2
7	Shipping	South San Francisco	2
8	(null)	Beijing	0
9	(null)	Bern	0
10	(null)	Bombay	0
11	(null)	Geneva	0
12	(null)	Hiroshima	0
13	(null)	London	0
14	(null)	Mexico City	0
15	(null)	Munich	0
16	(null)	Roma	0
17	(null)	Sao Paulo	0
18	(null)	Singapore	0
19	(null)	South Brunswick	0
20	(null)	Stretford	0
21	(null)	Sydney	0
22	(null)	Tokyo	0
23	(null)	Utrecht	0
24	(null)	Venice	0
25	(null)	Whitehorse	0
26	(null)	Not Assigned Yet	1

| Line 102 Column 1 | Insert | Modified | Unix/Mac: LF