

## LAB 2 – HOMETASKS

23K-0594

Q1.

The screenshot shows the SQL Developer interface. The top toolbar includes buttons for Welcome Page, homeuser, and HR. The main window is divided into a Worksheet and a Query Builder. The Worksheet contains the following SQL query:

```
SELECT job_id, MAX(salary) AS max_salary FROM employees GROUP BY job_id ORDER BY max_salary DESC;
```

Below the query, the Query Result tab is active, displaying the results of the query. The status bar indicates "All Rows Fetched: 19 in 0.004 seconds". The results are as follows:

JOB_ID	MAX_SALARY
AD_PRES	24000
AD_VP	17000
SA_MAN	14000
MK_MAN	13000
AC_MGR	12008
FI_MGR	12008
SA_REP	11500
PU_MAN	11000
PR_REP	10000
FI_ACCOUNT	9000
IT_PROG	9000
AC_ACCOUNT	8300
ST_MAN	8200
HR_REP	6500
MK_REP	6000
AD_ASST	4400
SH_CLERK	4200
ST_CLERK	3600
PU_CLERK	3100

Q2.

The screenshot shows the SQL Developer interface. The top toolbar includes buttons for Welcome Page, homeuser, and HR. The main window is divided into a Worksheet and a Query Builder. The Worksheet contains the following SQL query:

```
SELECT department_id, ROUND(AVG(salary),2) AS avg_salary FROM employees GROUP BY department_id ORDER BY department_id;
```

Below the query, the Query Result tab is active, displaying the results of the query. The status bar indicates "All Rows Fetched: 12 in 0.003 seconds". The results are as follows:

DEPARTMENT_ID	AVG_SALARY
10	4400
20	9500
30	4150
40	6500
50	3475.56
60	5760
70	10000
80	8955.88
90	19333.33
100	8601.33
110	10154
(null)	7000

Q3.

The screenshot shows the SQL Developer interface with a query in the Worksheet tab. The query is: `SELECT department_id, COUNT(*) AS num_employees FROM employees GROUP BY department_id HAVING COUNT(*) > 3;`. The Query Result tab shows the results of the query, which are 5 rows of data.

DEPARTMENT_ID	NUM_EMPLOYEES
100	6
30	6
50	45
80	34
60	5

Q4.

The screenshot shows the SQL Developer interface with a query in the Worksheet tab. The query is: `SELECT job_id, MIN(salary) AS min_salary FROM employees GROUP BY job_id ORDER BY min_salary ASC;`. The Query Result tab shows the results of the query, which are 19 rows of data.

JOB_ID	MIN_SALARY
ST_CLERK	2100
SH_CLERK	2500
PU_CLERK	2500
IT_PROG	4200
AD_ASST	4400
ST_MAN	5800
MK_REP	6000
SA_REP	6100
HR_REP	6500
FI_ACCOUNT	6900
AC_ACCOUNT	8300
PR_REP	10000
SA_MAN	10500
PU_MAN	11000
FI_MGR	12008
AC_MGR	12008
MK_MAN	13000
AD_VP	17000
AD_PRES	24000

Q5.

The screenshot shows the SQL Developer interface with a query in the Worksheet tab. The query is: `SELECT EXTRACT(YEAR FROM hire_date) AS hire_year, COUNT(*) AS num_employees_hired FROM employees GROUP BY EXTRACT(YEAR FROM hire_date);`. The Query Result tab shows the results of the query, which are 8 rows of data.

HIRE_YEAR	NUM_EMPLOYEES_HIRED
2005	29
2001	1
2006	24
2007	19
2003	6
2004	10
2002	7
2008	11

Q6.

Welcome Page homeuser AR	
Worksheet Query Builder	
SELECT department_id, SUM(commission_pct * salary) AS total_commission FROM employees WHERE commission_pct IS NOT NULL GROUP BY department_id ORDER BY department_id;	
Query Result x	
All Rows Fetched: 2 in 0.008 seconds	
DEPARTMENT_ID	TOTAL_COMMISSION
1 80	72640
2 (null)	1050

Q7.

Welcome Page homeuser AR

Worksheet

Query Builder

SELECT first\_name || ' ' || last\_name AS employee\_name, hire\_date, TO\_CHAR(hire\_date, 'Month') AS hire\_month FROM employees ORDER BY TO\_NUMBER(TO\_CHAR(hire\_date, 'MM'));

Query Result x

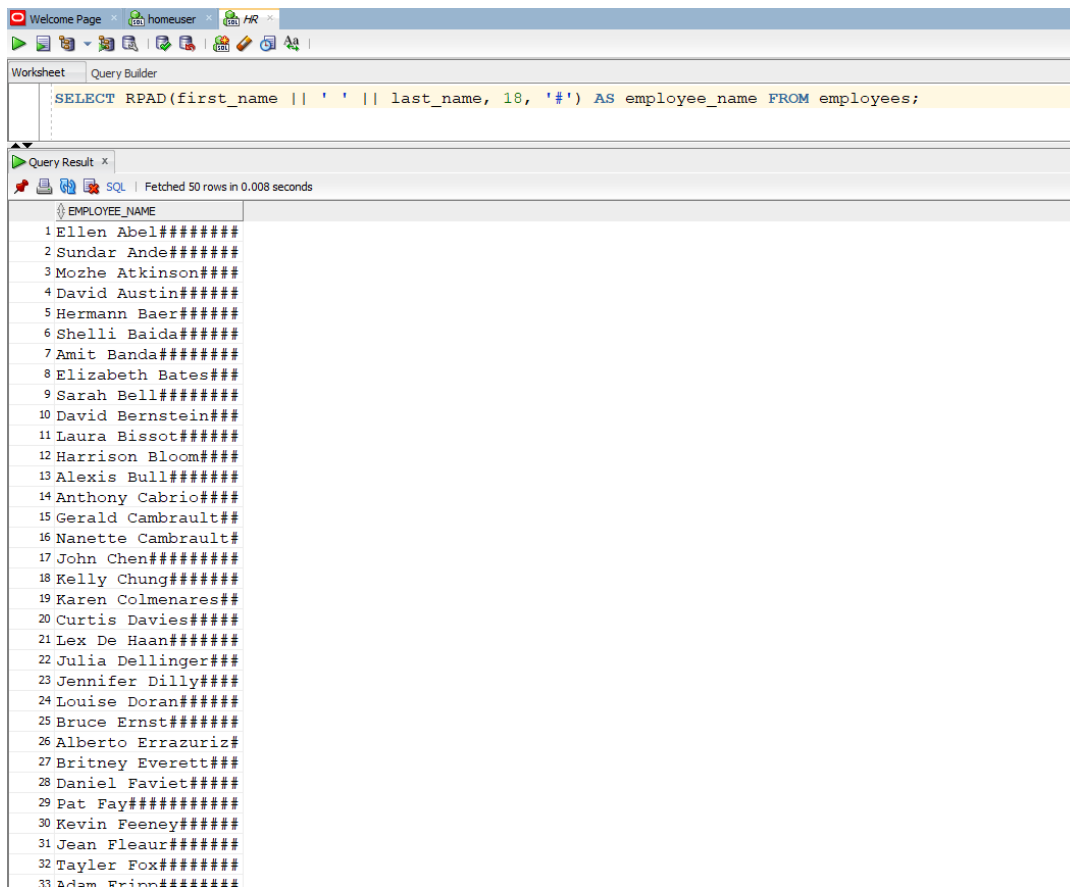
Fetched 50 rows in 0.008 seconds

	EMPLOYEE_NAME	HIRE_DATE	HIRE_MONTH
1	Lex De Haan	13-JAN-01	January
2	Alexander Hunold	03-JAN-06	January
3	Douglas Grant	13-JAN-08	January
4	Nandita Sarchand	27-JAN-04	January
5	Winston Taylor	24-JAN-06	January
6	Charles Johnson	04-JAN-08	January
7	Taylor Fox	24-JAN-06	January
8	Mattea Marvins	24-JAN-08	January
9	Janette King	30-JAN-04	January
10	Peter Tucker	30-JAN-05	January
11	Eleni Zlotkey	29-JAN-08	January
12	Karen Partners	05-JAN-05	January
13	Curtis Davies	29-JAN-05	January
14	James Landry	14-JAN-07	January
15	David Lee	23-FEB-08	February
16	Jean Fleaur	23-FEB-06	February
17	Sarah Bell	04-FEB-04	February
18	Alexis Bull	20-FEB-05	February
19	Diana Lorentz	07-FEB-07	February
20	Michael Hartstein	17-FEB-04	February
21	Girard Geoni	03-FEB-08	February
22	William Smith	23-FEB-07	February
23	James Marlow	16-FEB-05	February
24	Anthony Cabrio	07-FEB-07	February
25	Hazel Philtanker	06-FEB-08	February
26	John Seo	12-FEB-06	February
27	Valli Pataballa	05-FEB-06	February
28	Alyssa Hutton	19-MAR-05	March
29	Jonathon Taylor	24-MAR-06	March
30	Jose Manuel Urman	07-MAR-06	March
31	Elizabeth Bates	24-MAR-07	March
32	Britney Everett	03-MAR-05	March

Q8.

Welcome Page homeuser AR	
Worksheet Query Builder	
SELECT INITCAP(first_name    ' '    last_name) AS employee_name FROM employees;	
Query Result x	
Fetched 50 rows in 0.007 seconds	
EMPLOYEE_NAME	
1 Ellen Abel	
2 Sundar Ande	
3 Mozhe Atkinson	
4 David Austin	
5 Hermann Baer	
6 Shelli Baida	
7 Amit Banda	
8 Elizabeth Bates	
9 Sarah Bell	
10 David Bernstein	
11 Laura Bissot	
12 Harrison Bloom	
13 Alexis Bull	
14 Anthony Cabrio	
15 Gerald Cambrault	
16 Nanette Cambrault	
17 John Chen	
18 Kelly Chung	
19 Karen Colmenares	
20 Curtis Davies	
21 Lex De Haan	
22 Julia Dellinger	
23 Jennifer Dilly	
24 Louise Doran	
25 Bruce Ernst	
26 Alberto Errazuriz	
27 Britney Everett	
28 Daniel Faviet	
29 Pat Fay	
30 Kevin Feeney	
31 Jean Fleaur	
32 Tayler Fox	

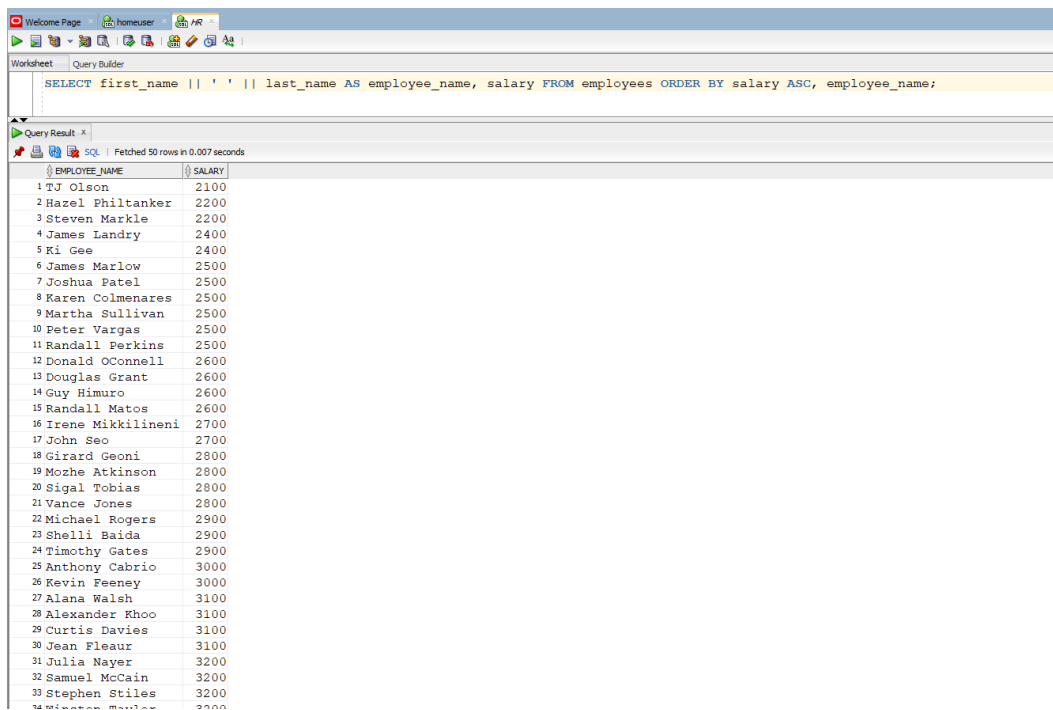
Q9.



The screenshot shows a database query interface with a 'Query Builder' tab. The SQL query is: `SELECT RPAD(first_name || ' ' || last_name, 18, '#') AS employee_name FROM employees;` The 'Query Result' tab shows 50 rows of data. The first column is 'EMPLOYEE\_NAME' and the second column is 'SALARY'.

EMPLOYEE_NAME	SALARY
1 Ellen Abel#####	2100
2 Sundar Ande#####	2200
3 Mozhe Atkinson####	2200
4 David Austin#####	2400
5 Hermann Baer#####	2400
6 Shelli Baida#####	2500
7 Amit Banda#####	2500
8 Elizabeth Bates###	2500
9 Sarah Bell#####	2500
10 David Bernstein###	2500
11 Laura Bissot#####	2500
12 Harrison Bloom###	2500
13 Alexis Bull#####	2500
14 Anthony Cabrio####	2600
15 Gerald Cambrault##	2600
16 Nanette Cambrault#	2600
17 John Chen#####	2600
18 Kelly Chung#####	2600
19 Karen Colmenares##	2600
20 Curtis Davies#####	2600
21 Lex De Haan#####	2600
22 Julia Dellinger###	2600
23 Jennifer Dilly###	2600
24 Louise Doran#####	2600
25 Bruce Ernst#####	2600
26 Alberto Errazuriz#	2600
27 Britney Everett###	2600
28 Daniel Faviot#####	2600
29 Pat Fay#####	2600
30 Kevin Feeney#####	2600
31 Jean Fleaur#####	2600
32 Tayler Fox#####	2600
33 Adam Fynn#####	2600

Q10.



The screenshot shows a database query interface with a 'Query Builder' tab. The SQL query is: `SELECT first_name || ' ' || last_name AS employee_name, salary FROM employees ORDER BY salary ASC, employee_name;` The 'Query Result' tab shows 50 rows of data. The first column is 'EMPLOYEE\_NAME' and the second column is 'SALARY'.

EMPLOYEE_NAME	SALARY
1 TJ Olson	2100
2 Hazel Philtanker	2200
3 Steven Markle	2200
4 James Landry	2400
5 Ki Gee	2400
6 James Marlow	2500
7 Joshua Patel	2500
8 Karen Colmenares	2500
9 Martha Sullivan	2500
10 Peter Vargas	2500
11 Randall Perkins	2500
12 Donald OConnell	2600
13 Douglas Grant	2600
14 Guy Himuro	2600
15 Randall Matos	2600
16 Irene Mikkilineni	2700
17 John Seo	2700
18 Girard Geoni	2800
19 Mozhe Atkinson	2800
20 Sigal Tobias	2800
21 Vance Jones	2800
22 Michael Rogers	2900
23 Shelli Baida	2900
24 Timothy Gates	2900
25 Anthony Cabrio	3000
26 Kevin Feeney	3000
27 Alana Walsh	3100
28 Alexander Khoo	3100
29 Curtis Davies	3100
30 Jean Fleaur	3100
31 Julia Nayer	3200
32 Samuel McCain	3200
33 Stephen Stiles	3200