UNIVERSITY OF CHITTAGONG

Department of Computer Science and Engineering

Session: 2021-2022
4th semester

Assignment No.	:	02
Course Title	:	Database Systems
Course Code No.	:	CSE-413

Submitted to:
Dr. Rudra Pratap Deb Nath
Associate Professor
Department of Computer Science and Engineering
University of Chittagong

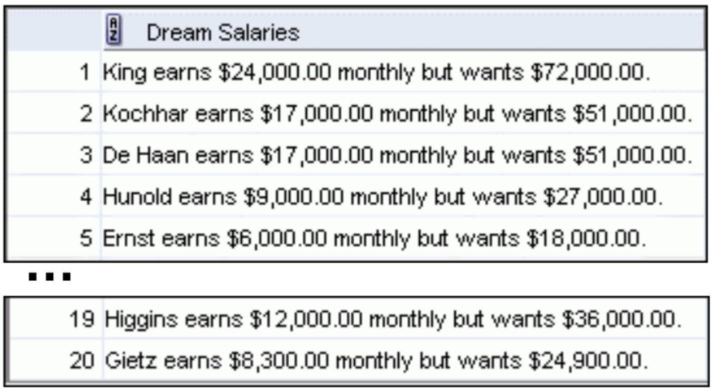
Submitted by:
Sanzid Islam Mahi
ID: 22701065
Department of Computer Science and Engineering
University of Chittagong

Date: Jul 06, 2024

Chapter 4

Practice 4

1. Create a report that produces the following for each employee: <employee last name> earns <salary> monthly but wants <3 times salary.>. Label the column Dream Salaries.

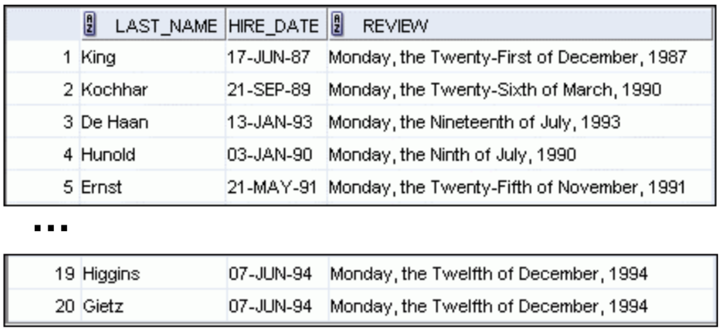


	Dream Salaries
1	King earns $24,000.00 monthly but wants $72,000.00.
2	Kochhar earns $17,000.00 monthly but wants $51,000.00.
3	De Haan earns $17,000.00 monthly but wants $51,000.00.
4	Hunold earns $9,000.00 monthly but wants $27,000.00.
5	Ernst earns $6,000.00 monthly but wants $18,000.00.
...	
19	Higgins earns $12,000.00 monthly but wants $36,000.00.
20	Gietz earns $8,300.00 monthly but wants $24,900.00.

Solution:

```
SELECT Last_Name || '␣earns␣' || Salary ||
    '␣monthly␣but␣wants␣' || (Salary * 3) || '.'
    AS "Dream␣Salaries"
FROM hr.Employees;
```

2. Display each employee's last name, hire date, and salary review date, which is the first Monday after six months of service. Label the column REVIEW. Format the dates to appear in the format similar to "Monday, the Thirty-First of July, 2000."

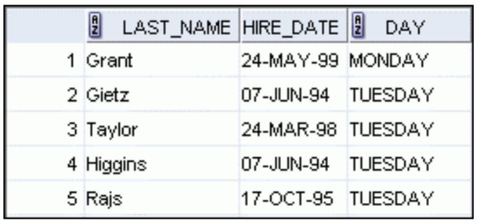


	LAST_NAME	HIRE_DATE	REVIEW
1	King	17-JUN-87	Monday, the Twenty-First of December, 1987
2	Kochhar	21-SEP-89	Monday, the Twenty-Sixth of March, 1990
3	De Haan	13-JAN-93	Monday, the Nineteenth of July, 1993
4	Hunold	03-JAN-90	Monday, the Ninth of July, 1990
5	Ernst	21-MAY-91	Monday, the Twenty-Fifth of November, 1991
...			
19	Higgins	07-JUN-94	Monday, the Twelfth of December, 1994
20	Gietz	07-JUN-94	Monday, the Twelfth of December, 1994

Solution:

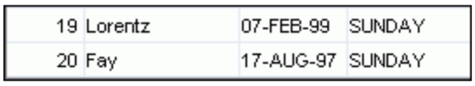
```
SELECT last_name, hire_date,
    TO_CHAR(
        NEXT_DAY(ADD_MONTHS(hire_date, 6) - 1, 'MONDAY'),
        'FMDay,"the"␣fmDdsp␣"of"␣FMMonth,YYYY'
    ) AS review
FROM hr.employees;
```

3. Display the last name, hire date, and day of the week on which the employee started. Label the column DAY. Order the results by the day of the week, starting with Monday.



	LAST_NAME	HIRE_DATE	DAY
1	Grant	24-MAY-99	MONDAY
2	Gietz	07-JUN-94	TUESDAY
3	Taylor	24-MAR-98	TUESDAY
4	Higgins	07-JUN-94	TUESDAY
5	Rajs	17-OCT-95	TUESDAY

...

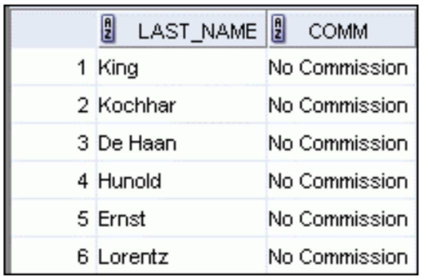


19	Lorentz	07-FEB-99	SUNDAY
20	Fay	17-AUG-97	SUNDAY

Solution:

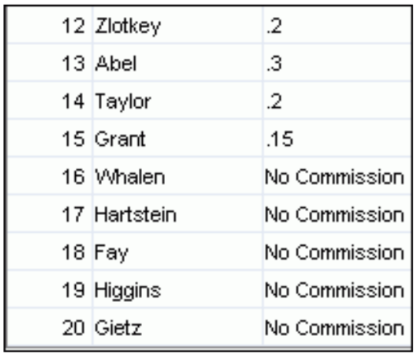
```sql
SELECT last_name,hire_date,
    TO_CHAR(hire_date, 'Day') AS DAY
FROM hr.employees
ORDER BY
    CASE
        WHEN TO_CHAR(hire_date, 'D') = '1' THEN 7
        ELSE TO_CHAR(hire_date, 'D') - 1
    END;
```

4. Create a query that displays the employees' last names and commission amounts. If an employee does not earn commission, show "No Commission." Label the column COMM.



	LAST_NAME	COMM
1	King	No Commission
2	Kochhar	No Commission
3	De Haan	No Commission
4	Hunold	No Commission
5	Ernst	No Commission
6	Lorentz	No Commission

...



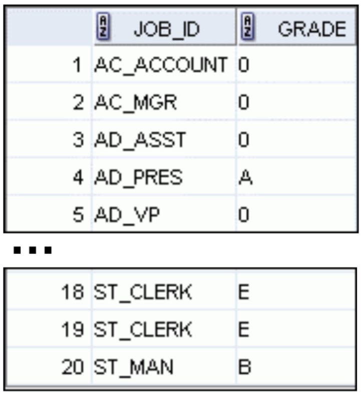
12	Zlotkey	.2
13	Abel	.3
14	Taylor	.2
15	Grant	.15
16	Whalen	No Commission
17	Hartstein	No Commission
18	Fay	No Commission
19	Higgins	No Commission
20	Gietz	No Commission

Solution:

```
SELECT last_name,
    CASE
        WHEN commission_pct IS NULL OR commission_pct = 0
        THEN 'No␣Commission'
        ELSE TO_CHAR(commission_pct,'fm.99')
    END AS comm
FROM hr.employees;
```

5. Using the DECODE function, write a query that displays the grade of all employees based on the value of the column `JOB_ID`, using the following data:

Job	**Grade**
AD_PRES	A
ST_MAN	B
IT_PROG	C
SA_REP	D
ST_CLERK	E
None of the above	0



	JOB_ID	GRADE
1	AC_ACCOUNT	0
2	AC_MGR	0
3	AD_ASST	0
4	AD_PRES	A
5	AD_VP	0

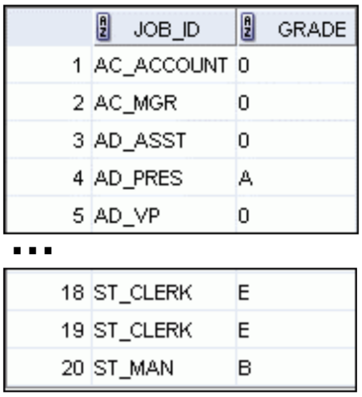
...

	JOB_ID	GRADE
18	ST_CLERK	E
19	ST_CLERK	E
20	ST_MAN	B

Solution:

```
SELECT last_name,job_id,
    DECODE(job_id,
            'AD_PRES', 'A',
            'ST_MAN', 'B',
            'IT_PROG', 'C',
            'SA_REP', 'D',
            'ST_CLERK', 'E',
            '0'
    ) AS grade
FROM hr.employees;
```

6. Rewrite the statement in the preceding exercise using the CASE syntax.



	JOB_ID	GRADE
1	AC_ACCOUNT	0
2	AC_MGR	0
3	AD_ASST	0
4	AD_PRES	A
5	AD_VP	0
...		
18	ST_CLERK	E
19	ST_CLERK	E
20	ST_MAN	B

Solution:

```sql
SELECT last_name,job_id,
    CASE job_id
        WHEN 'AD_PRES' THEN 'A'
        WHEN 'ST_MAN' THEN 'B'
        WHEN 'IT_PROG' THEN 'C'
        WHEN 'SA_REP' THEN 'D'
        WHEN 'ST_CLERK' THEN 'E'
        ELSE '0'
    END AS grade
FROM employees;
```

Chapter 5

practice 5

1. Group functions work across many rows to produce one result per group.
   True/False

   Answer: True

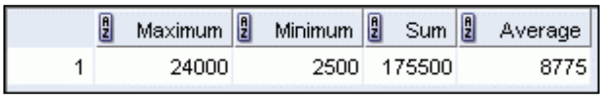
2. Group functions include nulls in calculations.
   True/False

   Answer: False

3. The WHERE clause restricts rows before inclusion in a group calculation.
   True/False

   Answer: True

4. Find the highest, lowest, sum, and average salary of all employees. Label the columns as Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number. Save your SQL statement as `lab_05_04.sql`. Run the query.



	Maximum	Minimum	Sum	Average
1	24000	2500	175500	8775

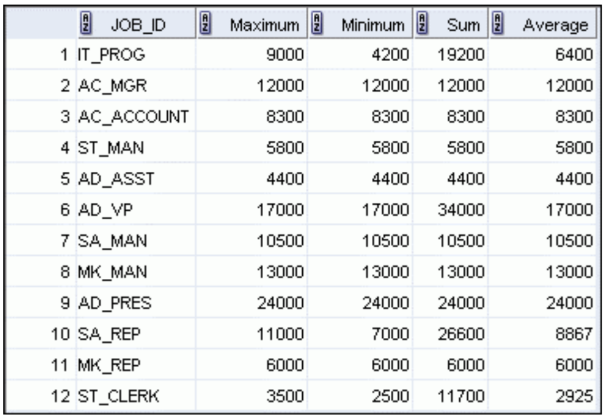
   Solution:

```sql
SELECT
    ROUND(MAX(salary)) AS "Maximum",
    ROUND(MIN(salary)) AS "Minimum",
    ROUND(SUM(salary)) AS "Sum",
    ROUND(AVG(salary)) AS "Average"
FROM hr.employees;
```

5. Modify the query in `lab_05_04.sql` to display the minimum, maximum, sum, and average salary for each job type. Resave `lab_05_04.sql` as `lab_05_05.sql`. Run the statement in `lab_05_05.sql`.

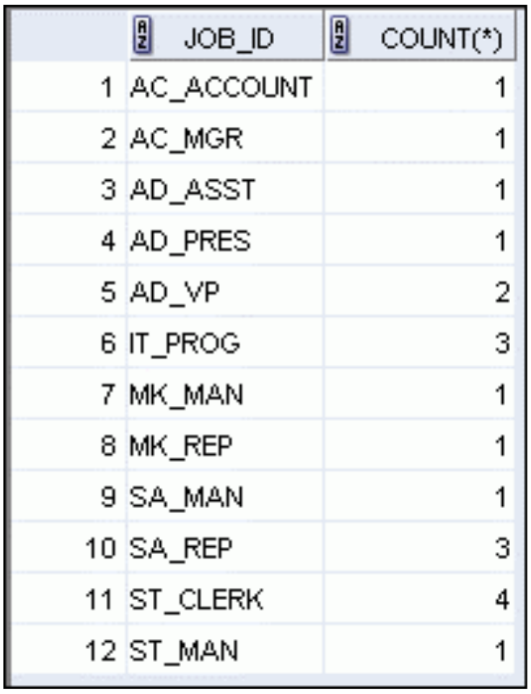
   Solution:

```sql
SELECT job_id,
    ROUND(MIN(salary)) AS "Minimum",
    ROUND(MAX(salary)) AS "Maximum",
    ROUND(SUM(salary)) AS "Sum",
    ROUND(AVG(salary)) AS "Average"
FROM hr.employees
GROUP BY job_id;
```



	JOB_ID	Maximum	Minimum	Sum	Average
1	IT_PROG	9000	4200	19200	6400
2	AC_MGR	12000	12000	12000	12000
3	AC_ACCOUNT	8300	8300	8300	8300
4	ST_MAN	5800	5800	5800	5800
5	AD_ASST	4400	4400	4400	4400
6	AD_VP	17000	17000	34000	17000
7	SA_MAN	10500	10500	10500	10500
8	MK_MAN	13000	13000	13000	13000
9	AD_PRES	24000	24000	24000	24000
10	SA_REP	11000	7000	26600	8867
11	MK_REP	6000	6000	6000	6000
12	ST_CLERK	3500	2500	11700	2925

6. Write a query to display the number of people with the same job.

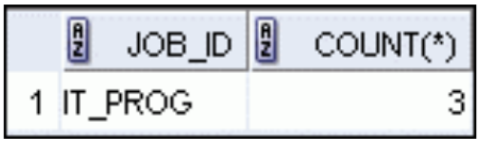


	JOB_ID	COUNT(*)
1	AC_ACCOUNT	1
2	AC_MGR	1
3	AD_ASST	1
4	AD_PRES	1
5	AD_VP	2
6	IT_PROG	3
7	MK_MAN	1
8	MK_REP	1
9	SA_MAN	1
10	SA_REP	3
11	ST_CLERK	4
12	ST_MAN	1

Solution:

```sql
SELECT job_id, COUNT(*)
FROM hr.employees
GROUP BY job_id;
```

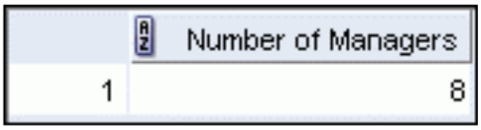
Generalize the query so that the user in the HR department is prompted for a job title. Save the script to a file named `lab_05_06.sql`. Run the query. Enter `IT_PROG` when prompted.



	JOB_ID	COUNT(*)
1	IT_PROG	3

skipped

7. Determine the number of managers without listing them. Label the column as Number of Managers. Hint: Use the MANAGER_ID column to determine the number of managers.

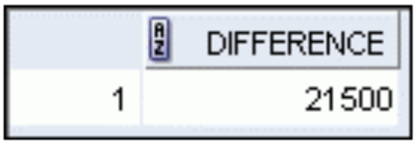


	Number of Managers
1	8

Solution:

```
SELECT
    COUNT(DISTINCT manager_id) AS "Number␣of␣Managers"
FROM hr.employees
WHERE manager_id IS NOT NULL;
```

8. Find the difference between the highest and lowest salaries. Label the column DIFFERENCE.

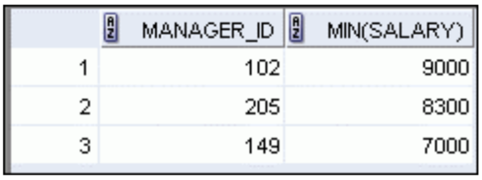


	DIFFERENCE
1	21500

Solution:

```
SELECT (MAX(salary) - MIN(salary)) AS DIFFERENCE
FROM hr.employees;
```

9. Create a report to display the manager number and the salary of the lowest-paid employee for that manager. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is $6,000 or less. Sort the output in descending order of salary.

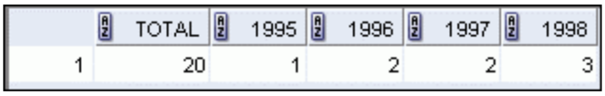


	MANAGER_ID	MIN(SALARY)
1	102	9000
2	205	8300
3	149	7000

Solution:

```
SELECT manager_id,MIN(SALARY)
FROM hr.employees
WHERE manager_id IS NOT NULL
GROUP BY manager_id
HAVING MIN(salary) > 6000
ORDER BY MIN(SALARY) DESC;
```

10. Create a query to display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998. Create appropriate column headings.

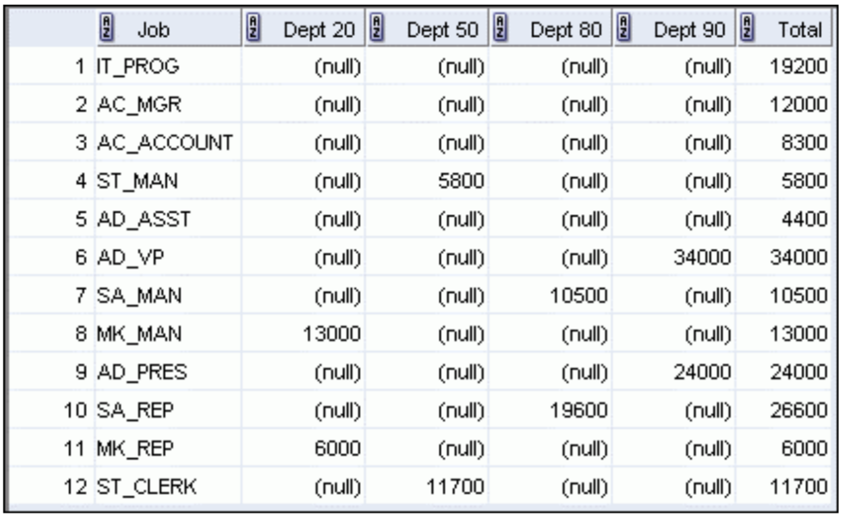


	TOTAL	1995	1996	1997	1998
1	20	1	2	2	3

Solution:

```
SELECT COUNT(*) AS "TOTAL",
    SUM(CASE WHEN TO_CHAR(hire_date, 'YYYY') = '1995'
        THEN 1 ELSE 0 END) AS "1995",
    SUM(CASE WHEN TO_CHAR(hire_date, 'YYYY') = '1996'
        THEN 1 ELSE 0 END) AS "1996",
    SUM(CASE WHEN TO_CHAR(hire_date, 'YYYY') = '1997'
        THEN 1 ELSE 0 END) AS "1997",
    SUM(CASE WHEN TO_CHAR(hire_date, 'YYYY') = '1998'
        THEN 1 ELSE 0 END) AS "1998"
FROM hr.employees;
```

11. Create a matrix query to display the job, the salary for that job based on department number, and the total salary for that job, for departments 20, 50, 80, and 90, giving each column an appropriate heading.



	Job	Dept 20	Dept 50	Dept 80	Dept 90	Total
1	IT_PROG	(null)	(null)	(null)	(null)	19200
2	AC_MGR	(null)	(null)	(null)	(null)	12000
3	AC_ACCOUNT	(null)	(null)	(null)	(null)	8300
4	ST_MAN	(null)	5800	(null)	(null)	5800
5	AD_ASST	(null)	(null)	(null)	(null)	4400
6	AD_VP	(null)	(null)	(null)	34000	34000
7	SA_MAN	(null)	(null)	10500	(null)	10500
8	MK_MAN	13000	(null)	(null)	(null)	13000
9	AD_PRES	(null)	(null)	(null)	24000	24000
10	SA_REP	(null)	(null)	19600	(null)	26600
11	MK_REP	6000	(null)	(null)	(null)	6000
12	ST_CLERK	(null)	11700	(null)	(null)	11700

Solution:

```
SELECT job_id AS "Job",
    SUM(CASE WHEN department_id = 20
        THEN salary ELSE NULL END) AS "Dept␣20",
    SUM(CASE WHEN department_id = 50
        THEN salary ELSE NULL END) AS "Dept␣50",
    SUM(CASE WHEN department_id = 80
        THEN salary ELSE NULL END) AS "Dept␣80",
    SUM(CASE WHEN department_id = 90
        THEN salary ELSE NULL END) AS "Dept 90",
    SUM(salary) AS "Total"
FROM hr.employees
WHERE department_id IN (20, 50, 80, 90)
GROUP BY job_id
ORDER BY job_id;
```