



# 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 , 2024

## Chapter 3

### Practice 3

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:**

```
1 SELECT Last_Name || ' earns ' || Salary ||  
2        ' monthly but wants ' || (Salary * 3) || '.'  
3        AS "Dream Salaries"  
4 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:**

```
1 SELECT  
2     last_name ,  
3     hire_date ,  
4     TO_CHAR(  
5         NEXT_DAY(ADD_MONTHS(hire_date, 6) - 1, 'MONDAY'),  
6         'FMDay, "the" fmDdsp "of" fMMonth, YYYY'  
7     ) AS review  
8 FROM  
9     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:

```

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

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:**

```
1 SELECT last_name ,
2        CASE
3            WHEN commission_pct IS NULL OR commission_pct = 0
4            THEN 'No Commission'
5            ELSE TO_CHAR(commission_pct,'fm.99')
6        END AS comm
7 FROM
8        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
...		
18	ST_CLERK	E
19	ST_CLERK	E
20	ST_MAN	B

**Solution:**

```
1 SELECT last_name , job_id ,
2        DECODE(job_id ,
3            'AD_PRES' , 'A' ,
4            'ST_MAN' , 'B' ,
5            'IT_PROG' , 'C' ,
6            'SA_REP' , 'D' ,
7            'ST_CLERK' , 'E' ,
8            '0'
9        ) AS grade
10 FROM hr.employees;
```

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

**Solution:**

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

...

18	ST_CLERK	E
19	ST_CLERK	E
20	ST_MAN	B

```

1 SELECT last_name,job_id,
2        CASE job_id
3            WHEN 'AD_PREP' THEN 'A'
4            WHEN 'ST_MAN' THEN 'B'
5            WHEN 'IT_PROG' THEN 'C'
6            WHEN 'SA_REP' THEN 'D'
7            WHEN 'ST_CLERK' THEN 'E'
8            ELSE '0' -- Default case for other values
9        END AS grade
10 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:** True

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:**

```
1 SELECT
2     ROUND(MAX(salary)) AS Maximum,
3     ROUND(MIN(salary)) AS Minimum,
4     ROUND(SUM(salary)) AS Sum,
5     ROUND(AVG(salary)) AS Average
6 FROM
7     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:**

```
1 SELECT job_id,
2     ROUND(MIN(salary)) AS Minimum,
3     ROUND(MAX(salary)) AS Maximum,
4     ROUND(SUM(salary)) AS Sum,
5     ROUND(AVG(salary)) AS Average
6 FROM hr.employees
7 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

	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

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

**Solution:**

```

1 SELECT job_id,
2        COUNT(*)
3 FROM hr.employees
4 GROUP BY job_id;

```

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.

**Solution:**

```

1 SELECT
2        COUNT(DISTINCT manager_id) AS "Number of Managers"
3 FROM

```

	AZ	Number of Managers
1		8

```

4     hr.employees
5 WHERE
6     manager_id IS NOT NULL;

```

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

	AZ	DIFFERENCE
1		21500

**Solution:**

```

1 SELECT
2     (MAX(salary) - MIN(salary)) AS DIFFERENCE
3 FROM
4     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.

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

**Solution:**

```

1 SELECT
2     manager_id,
3     MIN(salary) AS min_salary
4 FROM
5     hr.employees
6 WHERE
7     manager_id IS NOT NULL
8 GROUP BY
9     manager_id

```



```

10 HAVING
11     MIN(salary) > 6000
12 ORDER BY
13     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:**

```

1 SELECT
2     COUNT(*) AS "TOTAL",
3     SUM(CASE WHEN TO_CHAR(hire_date, 'YYYY') = '1995'
4           THEN 1 ELSE 0 END) AS "1995",
5     SUM(CASE WHEN TO_CHAR(hire_date, 'YYYY') = '1996'
6           THEN 1 ELSE 0 END) AS "1996",
7     SUM(CASE WHEN TO_CHAR(hire_date, 'YYYY') = '1997'
8           THEN 1 ELSE 0 END) AS "1997",
9     SUM(CASE WHEN TO_CHAR(hire_date, 'YYYY') = '1998'
10          THEN 1 ELSE 0 END) AS "1998"
11 FROM
12     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:**

```

1 SELECT
2     job_id AS "Job",
3     SUM(CASE WHEN department_id = 20
4           THEN salary ELSE NULL END) AS "Dept_20_Salary",
5     SUM(CASE WHEN department_id = 50

```

```
6         THEN salary ELSE NULL END) AS "Dept_50_Salary",
7     SUM(CASE WHEN department_id = 80
8         THEN salary ELSE NULL END) AS "Dept_80_Salary",
9     SUM(CASE WHEN department_id = 90
10        THEN salary ELSE NULL END) AS "Dept_90_Salary",
11    SUM(salary) AS "Total_Salary"
12 FROM
13     hr. employees
14 WHERE
15     department_id IN (20, 50, 80, 90)
16 GROUP BY
17     job_id
18 ORDER BY
19     job_id;
```