

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

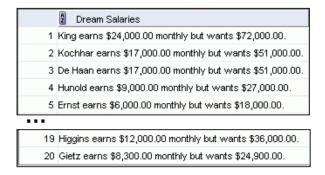
Department of Computer Science and Engineering University of Chittagong

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.

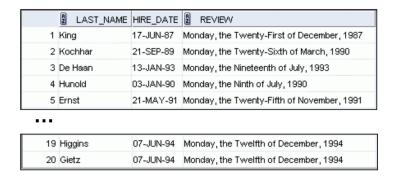


Solution:

```
SELECT Last_Name | | 'uearnsu' | | Salary | |
'umonthlyubutuwantsu' | | (Salary * 3) | | '.'

AS "DreamuSalaries"
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."



```
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.

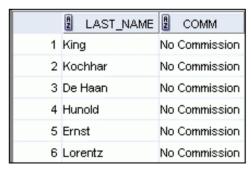


Solution:

```
SELECT Last_Name || 'uearnsu' || Salary ||
'umonthlyubutuwantsu' || (Salary * 3) || '.'

AS "DreamuSalaries"
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.



. . .

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	

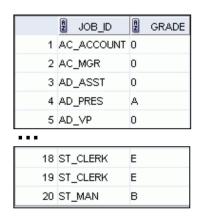
Solution:

```
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;
```

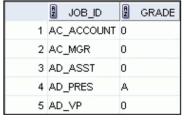
5. Using the DECODE function, write a query that displays the grade of all employees based on the value of the column JOB_ID, using the following data:

Job	\mathbf{Grade}
$AD_{-}PRES$	A
ST_MAN	В
IT_PROG	\mathbf{C}
SA_REP	D
ST_CLERK	\mathbf{E}
None of the above	0



Solution:

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



- - -

18 ST_CLERK	E
19 ST_CLERK	E
20 ST_MAN	В

```
SELECT last_name,job_id,
     CASE job_id
2
          WHEN 'AD_PRES' THEN 'A'
3
          WHEN 'ST_MAN' THEN 'B'
4
          WHEN 'IT_PROG' THEN 'C'
          WHEN 'SA_REP' THEN 'D'
          WHEN 'ST_CLERK' THEN 'E'
7
          ELSE '0' -- Default case for other values
8
     END AS grade
9
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.



Solution:

```
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.

```
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	2 Maximum	2 Minimum	2 Sum	2 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



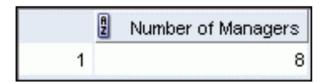
6. Write a query to display the number of people with the same job.

Solution:

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

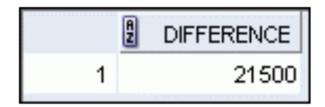
7. Determine the number of managers without listing them. Label the column as Number of Managers. Hint: Use the MANAGER_ID column to determine the number of managers.

```
SELECT
COUNT(DISTINCT manager_id) AS "Number of Managers"
FROM
```



```
hr.employees
WHERE
manager_id IS NOT NULL;
```

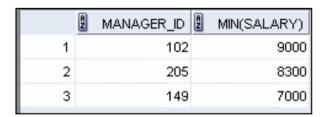
8. Find the difference between the highest and lowest salaries. Label the column DIFFERENCE.



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.



```
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
```

```
HAVING
MIN(salary) > 6000
CRDER BY
min_salary DESC;
```

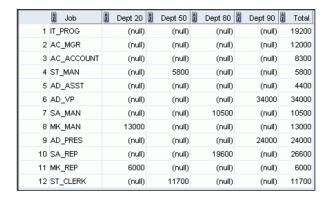
10. Create a query to display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998. Create appropriate column headings.



Solution:

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



```
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
```

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