



# Assignment

Assignment No. – 04

Submission date- 23 February, 2022

Course Title- DBMS (Lab)

Course Code: CSE-2424

Submitted to-

**Mr. Mohammad Aman Ullah**

**Assistant Professor**

01815641524

[ullah047@yahoo.com](mailto:ullah047@yahoo.com), [aman\\_cse@iiuc.ac.bd](mailto:aman_cse@iiuc.ac.bd)

Submitted by-

**MD. SOROWAR MAHABUB RABBY**

Matric ID: **C201032**, Section: **4AM**, Semester: **4<sup>th</sup>**

Department of CSE (Computer Science and Engineering), IIUC

Cell: 01834756433, 01521564157, [c201032@ugrad.iiuc.ac.bd](mailto:c201032@ugrad.iiuc.ac.bd)

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

Answer: True

2. Group functions include nulls in calculations.

Answer: False

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

Answer: True

The HR department needs the following reports:

4. Find the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number. Place your SQL statement in a text file named lab\_04\_04.sql.

```
SELECT  ROUND(MAX(salary),0) "Maximum",
        ROUND(MIN(salary),0) "Minimum",
        ROUND(SUM(salary),0) "Sum",
        ROUND(AVG(salary),0) "Average"
FROM employees;
```

Maximum	Minimum	Sum	Average
24000	2100	691400	6462

1 rows returned

5. Modify the query in lab\_04\_04.sql to display the minimum, maximum, sum, and average salary for each job type. Resave lab\_04\_04.sql as lab\_04\_05.sql. Run the statement in lab\_04\_05.sql.

```
SELECT job_id, ROUND(MAX(salary),0) "Maximum",
        ROUND(MIN(salary),0) "Minimum",
        ROUND(SUM(salary),0) "Sum",
        ROUND(AVG(salary),0) "Average"
FROM employees
GROUP BY job_id;
```

JOB_ID	Maximum	Minimum	Sum	Average
IT_PROG	9000	4200	28800	5760
AC_MGR	12000	12000	12000	12000

AC_ACCOUNT	8300	8300	8300	8300
ST_MAN	8200	5800	36400	7280
PU_MAN	11000	11000	11000	11000

19 rows returned

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

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

JOB_ID	COUNT(*)
AC_ACCOUNT	1
AC_MGR	1
AD_ASST	1
AD_PRES	1
AD_VP	2
FI_ACCOUNT	5
FI_MGR	1
HR_REP	1
IT_PROG	5
MK_MAN	1
MK_REP	1
PR_REP	1
PU_CLERK	5
PU_MAN	1
SA_MAN	5
SA_REP	30
SH_CLERK	20
ST_CLERK	20
ST_MAN	5

19 rows returned

**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\_04\_06.sql.**

```
SELECT job_id, COUNT(job_id)
FROM employees
WHERE job_id = '&JOB_TITLE'
GROUP BY job_id;
```

**Example:**

```
SELECT job_id, COUNT(job_id)
FROM employees
WHERE job_id = 'IT_PROG'
GROUP BY job_id;
```

JOB_ID	COUNT(JOB_ID)
IT_PROG	5

7. Determine the number of managers without listing them. Label the column **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 employees;
```

Number Of Managers
18

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

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

DIFFERENCE
21900

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.

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

MANAGER_ID	MIN(SALARY)
102	9000
205	8300
146	7000
145	7000

108	6900
149	6200
147	6200
148	6100
201	6000

9 rows returned

- 10. Create a query that will 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.**

```
SELECT COUNT(*) AS TOTAL_EMPLOYEES,
SUM(DECODE(TO_CHAR(hire_date, 'YYYY'),1995,1,0)) AS "In_1995",
SUM(DECODE(TO_CHAR(hire_date, 'YYYY'),1996,1,0)) AS "In_1996",
SUM(DECODE(TO_CHAR(hire_date, 'YYYY'),1997,1,0)) AS "In_1997",
SUM(DECODE(TO_CHAR(hire_date, 'YYYY'),1998,1,0)) AS "In_1998"
FROM employees;
```

TOTAL_EMPLOYEES	In_1995	In_1996	In_1997	In_1998
107	4	10	28	23

1 rows returned

- 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.**

```
SELECT job_id "Job",
SUM(DECODE(department_id , 20, salary)) "Dept 20",
SUM(DECODE(department_id , 50, salary)) "Dept 50",
SUM(DECODE(department_id , 80, salary)) "Dept 80",
SUM(DECODE(department_id , 90, salary)) "Dept 90",
SUM(salary) "Total"
FROM employees
GROUP BY job_id;
```

Job	Dept- 20	Dept- 50	Dept- 80	Dept- 90	Total
IT_PROG	-	-	-	-	28800
AC_MGR	-	-	-	-	12000
AC_ACCOUNT	-	-	-	-	8300
ST_MAN	-	36400	-	-	36400
PU_MAN	-	-	-	-	11000
AD_ASST	-	-	-	-	4400
AD_VP	-	-	-	34000	34000
SH_CLERK	-	64300	-	-	64300
FI_ACCOUNT	-	-	-	-	39600
FI_MGR	-	-	-	-	12000
PU_CLERK	-	-	-	-	13900

SA_MAN	-	-	61000	-	61000
MK_MAN	13000	-	-	-	13000
PR_REP	-	-	-	-	10000
AD_PRES	-	-	-	24000	24000
SA_REP	-	-	243500	-	250500
MK_REP	6000	-	-	-	6000
ST_CLERK	-	55700	-	-	55700
HR_REP	-	-	-	-	6500

19 rows returned

Submitted by-

**MD. SOROWAR MAHABUB RABBY**

Matric ID: **C201032**, Section: **4AM**

Department of CSE (Computer Science and Engineering)