NAME: MOAZZAM FAROOQUI
ROLL NO: CT-24068
COURSE CODE: CT-261
INSTRUCTOR: SIR ROHAIL QAMAR
TASK: DBMS LAB#05

Q1. Group functions work across many rows to produce one result per group. True/False

ANSWER:
True

Q2. Group functions include nulls in calculations. True/False

ANSWER:
False

Q3. The WHERE clause restricts rows prior to inclusion in a group calculation True/False

ANSWER:
True

Q4. Display 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.

SQL QUERY:

```
1   /*Display the highest,lowest,sum,and average salary of all
2   employees.Label the columns Maximum,Minimum,Sum,and
3   Average,respectively.Round your results to the nearest
4   whole number.*/
5
6   SELECT ROUND(MAX(salary)) AS "Maximum",ROUND(MIN(salary))
7   AS "Minimum",ROUND(SUM(salary)) AS "Sum",
8   ROUND(AVG(salary)) AS "Average" FROM HR.EMPLOYEES;
```

OUTPUT:

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.005 seconds

	MAXIMUM	MINIMUM	SUM	AVERAGE
1	24000	2100	691416	6462

Q5. Modify the query in Q4 to display the minimum, maximum, sum, and average salary for each job type.

SQL QUERY:

```
1   /*Modify the query in Q4 to display the minimum, maximum,
2   sum, and average salary for each job type.*/
3
4   SELECT job_id,MIN(salary) AS "Min",MAX(salary) AS "Max",
5   SUM(salary) AS "Sum",AVG(SALARY) AS "Average"
6   FROM HR.EMPLOYEES GROUP BY job_id;
```

OUTPUT:

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.006 seconds

	JOB_ID	MINIMUM	MAXIMUM	SUM	AVERAGE
1	AD_PRES	24000	24000	24000	24000
2	AD_VP	17000	17000	34000	17000
3	IT_PROG	4200	9000	28800	5760
4	FI_MGR	12008	12008	12008	12008
5	FI_ACCOUNT	6900	9000	39600	7920
6	PU_MAN	11000	11000	11000	11000
7	PU_CLERK	2500	3100	13900	2780
8	ST_MAN	5800	8200	36400	7280

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.006 seconds

	JOB_ID	MINIMUM	MAXIMUM	SUM	AVERAGE
12	SH_CLERK	2500	4200	64300	3215
13	AD_ASST	4400	4400	4400	4400
14	MK_MAN	13000	13000	13000	13000
15	MK_REP	6000	6000	6000	6000
16	HR_REP	6500	6500	6500	6500
17	PR_REP	10000	10000	10000	10000
18	AC_MGR	12008	12008	12008	12008
19	AC_ACCOUNT	8300	8300	8300	8300

Q6. Write a query to display the number of people with the same job.

SQL QUERY:

```
1    --Write a query to display the number of people with the same job.
2
3    SELECT job_id,COUNT(*) AS "Employees with same job"
4    FROM HR.EMPLOYEES GROUP BY job_id;
5
```

OUTPUT:

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.005 seconds

	JOB_ID	EMPLOYEES WITH SAME JOB
1	AC_ACCOUNT	1
2	AC_MGR	1
3	AD_ASST	1
4	AD_PRES	1
5	AD_VP	2
6	FI_ACCOUNT	5
7	FI_MGR	1
8	HR_REP	1

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.005 seconds

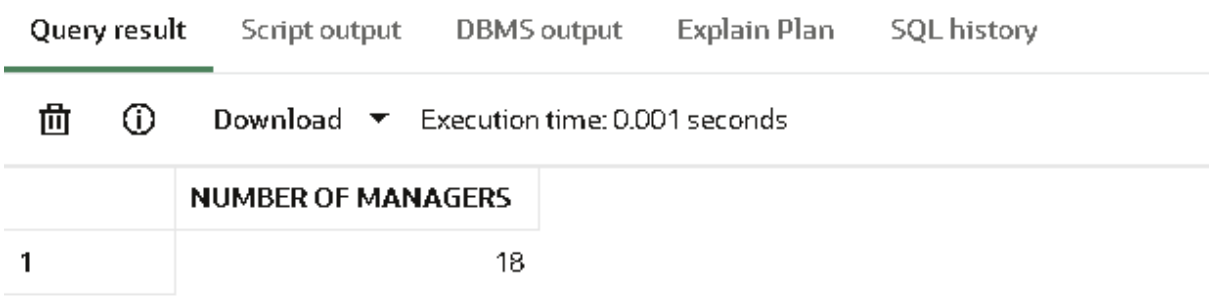
	JOB_ID	EMPLOYEES WITH SAME JOB
12	PR_REP	1
13	PU_CLERK	5
14	PU_MAN	1
15	SA_MAN	5
16	SA_REP	30
17	SH_CLERK	20
18	ST_CLERK	20
19	ST_MAN	5

Q7. 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.

SQL QUERY:

```
1   /*Determine the number of managers without listing them.
2   Label the column Number of Managers.Hint:Use the MANAGER_ID
3   column to determine the number of managers.*/
4
5   SELECT COUNT(DISTINCT manager_id) AS "Number of Managers"
6   FROM HR.EMPLOYEES;
```

OUTPUT:



	NUMBER OF MANAGERS
1	18

Q8. Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE.

SQL QUERY:

```
1   /*Write a query that displays the difference between the highest
2   and lowest salaries. Label the column DIFFERENCE.*/
3
4   SELECT MAX(salary)-MIN(salary) AS "DIFFERENCE"
5   FROM HR.EMPLOYEES;
```

OUTPUT:

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.005 seconds

	DIFFERENCE
1	21900

Q9. 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 less than $6,000. Sort the output in descending order of salary.

SQL QUERY:

```
/*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
less than $6,000.Sort the output in descending order of salary.*/

SELECT manager_id,MIN(salary) AS "Minimum Salary"
FROM HR.EMPLOYEES
WHERE manager_id IS NOT NULL
GROUP BY manager_id
HAVING MIN(salary)>=6000
ORDER BY MIN(salary) DESC;
```

OUTPUT:

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.001 seconds

	MANAGER_ID	MINIMUM SALARY
1	102	9000
2	205	8300
3	145	7000
4	146	7000
5	108	6900
6	147	6200
7	149	6200
8	148	6100
9	201	6000

Q10. Write a query to display each department's name,location, number of employees,and the average salary for all employees in that department. Label the columns Name,Location,Number of People ,and Salary , respectively. Round the average salary to two decimal places.

SQL QUERY:

```
/*Write a query to display each department's name,location,
number of employees,and the average salary for all employees
in that department.Label the columns Name,Location,
Number of People,and Salary,respectively.
Round the average salary to two decimal places.*/

SELECT d.department_name AS "Name",d.location_id AS "Location",
COUNT(e.employee_id) AS "Number of People",
ROUND(AVG(e.salary),2) AS "Salary"
FROM HR.EMPLOYEES e
INNER JOIN HR.DEPARTMENTS d
ON e.department_id=d.department_id
GROUP BY d.department_name,d.location_id
```

OUTPUT:

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.012 seconds

	NAME	LOCATION	NUMBER OF PEOPLE	SALARY
1	Administration	1700	1	4400
2	Marketing	1800	2	9500
3	Purchasing	1700	6	4150
4	Human Resources	2400	1	6500
5	Shipping	1500	45	3475.56
6	IT	1400	5	5760
7	Public Relations	2700	1	10000
8	Sales	2500	34	8955.88
9	Executive	1700	3	19333.33
10	Finance	1700	6	8601.33
11	Accounting	1700	2	10154

Q11. 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.

SQL QUERY:

```
1   /*Create a query that will display the total number of employees
2   and, of that total, the number of employees hired in 1995,1996,
3   1997,and 1998.Create appropriate column headings.*/
4
5   SELECT COUNT(*) "Total",
6     SUM(DECODE(TO_CHAR(hire_date,'YYYY'),'1995',1,0))"1995",
7     SUM(DECODE(TO_CHAR(hire_date,'YYYY'),'1996',1,0))"1996",
8     SUM(DECODE(TO_CHAR(hire_date,'YYYY'),'1997',1,0))"1997",
9     SUM(DECODE(TO_CHAR(hire_date,'YYYY'),'1998',1,0))"1998"
10  FROM HR.EMPLOYEES;
```

OUTPUT:

Query result | Script output | DBMS output | Explain Plan | SQL history

Download ▾ Execution time: 0.002 seconds

	TOTAL	1995	1996	1997	1998
1	107	0	0	0	0

Q12. 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.

SQL QUERY:

```
/*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,
 SUM(DECODE(department_id,20,salary,0))"Dept 20",
 SUM(DECODE(department_id,50,salary,0))"Dept 50",
 SUM(DECODE(department_id,80,salary,0))"Dept 80",
 SUM(DECODE(department_id,90,salary,0))"Dept 90",
 SUM(SALARY) "Total"
FROM HR.EMPLOYEES
GROUP BY job_id;
```

OUTPUT:

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.007 seconds

	JOB_ID	DEPT 20	DEPT 50	DEPT 80	DEPT 90	TOTAL
1	AD_PRES	0	0	0	24000	24000
2	AD_VP	0	0	0	34000	34000
3	IT_PROG	0	0	0	0	28800
4	FI_MGR	0	0	0	0	12008
5	FI_ACCOUNT	0	0	0	0	39600
6	PU_MAN	0	0	0	0	11000
7	PU_CLERK	0	0	0	0	13900
8	ST_MAN	0	36400	0	0	36400

Query result | Script output | DBMS output | Explain Plan | SQL history

Download | Execution time: 0.007 seconds

	JOB_ID	DEPT 20	DEPT 50	DEPT 80	DEPT 90	TOTAL
12	SH_CLERK	0	64300	0	0	64300
13	AD_ASST	0	0	0	0	4400
14	MK_MAN	13000	0	0	0	13000
15	MK_REP	6000	0	0	0	6000
16	HR_REP	0	0	0	0	6500
17	PR_REP	0	0	0	0	10000
18	AC_MGR	0	0	0	0	12008
19	AC_ACCOUNT	0	0	0	0	8300