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

4. Find the highest, lowest, sum, and average salary of all employees. Label the columnsas Maximum, Minimum, Sum, and Average, respectively. Round your results to the nea rest whole number

Query : select max(salary),min(salary),sum(salary),round(avg(salary)) from employees;

5. Modify the query in lab_05_04.sql to display the minimum, maximum, sum, and average salary for each job type.

Query : select distinct job_id,max(salary),min(salary),sum(salary),round(avg(salary)) from employees group by job_id order by 1;

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

Query : select distinct job_id,count(job_id) from employees group by job_id;

6(b)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

Query : select distinct job_id,count(job_id) "count(*)" from employees group by job_id having count(job_id) = 5 order by 1;

7. Determine the number of managers without listing them. Label the column as Number of Managers.

Query : select count(distinct manager_id) "Numberofmanagers" from employees;

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

QUERY : select max(salary)-min(salary) "DIFFERENCE" from 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

```
QUERY : 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;
```

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.

```
QUERY : select count (*) ,sum(decode(to_char(hire_date,'yyyy'),1995,1,0)) "1995",
       sum(decode(to_char(hire_date,'yyyy'),1996,1,0)) "1996",
       sum(decode(to_char(hire_date,'yyyy'),1997,1,0)) "1997",
       sum(decode(to_char(hire_date,'yyyy'),1998,1,0)) "1998"
       from 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.

```
QUERY : 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;
```