RAJALAKSHMI ENGINEERING COLLEGE
RAJALAKSHMI NAGAR, THANDALAM – 602 105

RAJALAKSHMI
ENGINEERING
COLLEGE

CS23332 DATABASE MANAGEMENT SYSTEMS LAB

Laboratory Record Note Book

Name : P. M. Arunesh

Year / Branch / Section : 2025 / CSE - Cyber security

University Register No. : 211624190 1007

College Roll No. : 241901007

Semester : III

Academic Year : 2024 - 28

RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

BONAFIDE CERTIFICATE

NAME P.M. Arunesh

ACADEMIC YEAR 2024-28 SEMESTER 3rd BRANCH CSE - Cyber Security

UNIVERSITY REGISTER No. 211624901007

Certified that this is the bonafide record of work done by the above student in the

Database management System Laboratory during the year 2025 - 2026

Signature of Faculty - in - Charge

Submitted for the Practical Examination held on .....

Internal Examiner

External Examiner

INDEX

Name : P.M.Arunesh Branch : CSE-CS Sec : A Roll No : 2241901007

S.No.	Date	Title	Page No.	Teacher's Sign / Remarks
1.	16/7/25	Creating & managing table	1	
2.	16/7/25	Manipulating data	7	
3.	17/7/25	including constraints	17	
4.	23/7/25	writing basic SQL SELECT statement.	26	
5.	24/7/25	Restricting & sorting data	34	
6.	30/7/25	single row function	45	
7.	31/7/25	Displaying data from multiple table	61	
8.	7/8/25	Aggregating data using group function	70	
9.	13/8/25	Sub Queries	83	
10.	14/8/25	Creating Queries	91	
11.	20/8/25	using the set operators	99	
12.	3/9/25	Not Null & UNIQUE constraints.	141	
13.	17/9/25	Creating queries	149	
14.	18/9/25	other database object	158	
15.	24/9/25	Controlling User Access	163	
16.	25/9/25	PL/SQL	171	
17.	25/9/25	control structure.	172	
18.	8/10/25	Procedure & function	192	
19.	9/10/25	Trigger	199	
20.	15/10/25	Mango DB	220	

SELECT department_id, AVG(salary) FROM employees GROUP BY department_id
HAVING max(salary)>10000;

Example displays the job ID and total monthly salary for each job that has a total payroll exceeding $13,000. The example excludes sales representatives and sorts the list by the total monthly salary.

SELECT job_id, SUM(salary) PAYROLL FROM employees WHERE job_id NOT LIKE '%REP%'
GROUP BY job_id HAVING SUM(salary) > 13000 ORDER BY SUM(salary);

Nesting Group Functions

Display the maximum average salary:
Group functions can be nested to a depth of two. The slide example displays the maximum average salary.

SELECT MAX(AVG(salary)) FROM employees GROUP BY department_id;

Summary

In this exercise, students should have learned how to:

- Use the group functions COUNT, MAX, MIN, and AVG
- Write queries that use the GROUP BY clause
- Write queries that use the HAVING clause

SELECT *column, group_function*
FROM *table*
[WHERE *condition*]
[GROUP BY *group_by_expression*]
[HAVING *group_condition*]
[ORDER BY *column*];

Find the Solution for the following:

Determine the validity of the following three statements. Circle either True or False.

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

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

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

The HR department needs the following reports:

4. Find the highest, lowest, sum, and average salary of all employees. Label the columns

SELECT ROUND (MAX (salary)) AS maximum, ROUND (MIN (salary)) AS minimum, ROUND (SUM (salary)) AS Sum, ROUND (AVG (salary)) AS Average By job_id;

Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number

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

SELECT job_id, ROUND (MIN (salary)) AS minimum, ROUND (MAX (salary)) AS maximum, ROUND (sum, Salary)) AS Sum ROUND (AVG) (salary));

6. Write a query to display the number of people with the same job. Generalize the query so that the user in the HR department is prompted for a job title.

SELECT COUNT (*) AS COUNT FROM employees WHERE job_id = ?;

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 (DISTing manager_id) AS 'Number OA Managers'. FROM employees WHERE manager_id IS NOT NULL;

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

SELECT MAX (salary) - MIN (salary) AS 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.

```
SELECT manager_id, min (salary) AS lowest_salary
FROM employees WHERE manager_id IS NOT NULL
GROUP by manager_id HAVING MIN (salary) >
6000 ORDER BY lowest_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.

```
SELECT COUNT(*) AS Total_Employees, SUM(YEAR
(hire_date) = 1995) AS Hired_1995, Sum (year(
hire_date) = 1996) AS Hired_1996, SUM(YEAR(
date) = 1997 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.

```
SELECT job_id, SUM (CASE WHEN department_id = 20
THEN Salary ELSE 0 END) AS Dept_20_Salary,
SUM (CASE WHEN department_id = 50 THEN
Salary ELSE 0 END) AS Dept_50_Salary,
SUM (CASE WHEN department_id = 80;
```

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

```
SELECT d.department_name AS 'Name', Location,
l.city AS LOCATION, COUNT(e.employee_id) AS
Number of people', ROUND(AVG(e.salary),2) AS
SALARY FROM employees e JOIN departments d ON
e.department_id = d.department_id GROUP BY
d.department_name, l.city;
```

Evaluation Procedure	Marks awarded
Query(5)	5
Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	