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 | |

**Find the Solution for the following:**

1. Write a query to display the current date. Label the column Date.

```
select current_date() AS date;
```

2. The HR department needs a report to display the employee number, last name, salary, and increased by 15.5% (expressed as a whole number) for each employee. Label the column New Salary.

```
SELECT employee_id AS 'Employee Number', Last_name
salary, ROUND (salary * (1.155) AS 'New Salary
ROUND (salary 1.155);
```

3. Modify your query lab_03_02.sql to add a column that subtracts the old salary from the new salary. Label the column Increase.

```
select CONCAT(UCASE(LEFT(last_name, 1)), LCASE
(SUBSTRING
(Last_name, 2))) AS 'FORMATTED Last Name',
LENGTH (last_name) AS 'Name length FROM
employees where last_name;
```

4. Write a query that displays the last name (with the first letter uppercase and all other letters lowercase) and the length of the last name for all employees whose name starts with the letters J, A, or M. Give each column an appropriate label. Sort the results by the employees' last names.

```
SELECT (CONCAT(UCASE(LEFT(last_name, 1)), LCASE
(SUBSTRING
(last_name, 2))) AS 'FORMATTED Last Name',
LENGTH (last_name)
As' Name length from employees where last_name;
```

5. Rewrite the query so that the user is prompted to enter a letter that starts the last name. For example, if the user enters H when prompted for a letter, then the output should show all employees whose last name starts with the letter H.

```
SELECT Employee_id AS 'Employee Number', Last_name,
Salary, ROUND (salary * 1.155) AS 'New salary', FR
employees;
```

6. The HR department wants to find the length of employment for each employee. For each employee, display the last name and calculate the number of months between today and the date on which the employee was hired. Label the column MONTHS_WORKED. Order your results by the number of months employed. Round the number of months up to the closest whole number.

```
SELECT last_name, CEIL(TIMESTAMPDIFF(MONTH
, hire_date, current_DATE())) AS MONTHS_WORKED
FROM employees ORDER BY MONTHS_WORKED DESC;
```

**Note:** Your results will differ.

7. 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.

```
SELECT CONCAT (last_name, 'earns', salary, 'monthly
but wants', salary*3) AS 'DREAM Salaries' FROM
employees;
```

8. Create a query to display the last name and salary for all employees. Format the salary to be 15 characters long, left-padded with the $ symbol. Label the column SALARY.

```
SELECT last_name, LPAD(CONCAT('$', salary)
,15,'$') AS salary FROM employees;
```

9. 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."

```
SELECT last_name, hire_date, DATE_FORMAT(
DATE_ADD(hire_date, INTERVAL(8 - DAY OF WEEK
(DATE_ADD)
FROM employees;
```

10. 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.

```
SELECT last_name, hire_date, DAYNAME(hire_date)
AS DATEFROM employees ORDER BY FIELD(DAYNAME
(hire_date), 'MONDAY', 'TUESDAY', 'WEDNESDAY', 'THURSDAY',
'FRIDAY', 'SATURDAY', 'SUNDAY');
```

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