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. : 2116241901007

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-~~2~~28 SEMESTER 3rd BRANCH CSE - Cyber security

UNIVERSITY REGISTER No. 2116241901007

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 : 241901007

Find the Solution for the following:

1. What privilege should a user be given to log on to the Oracle Server? Is this a system or an object privilege?

Grant Create session To usename;

2. What privilege should a user be given to create tables?

Grant Create Table To usename

3. If you create a table, who can pass along privileges to other users on your table? only user who created the table are granted Privilage with Grant option

Grant SELECT on dept To user . with Grant Option

4. You are the DBA. You are creating many users who require the same system privileges. What should you use to make your job easier?

Grant Create session, CREATE TabLE TO dept_Job;

Grant dept -rule to user 1, user2;

5. What command do you use to change your password?

Password New_Password; Alter USER name

IDENTIFIED BY New - passwrd;

6. Grant another user access to your DEPARTMENTS table. Have the user grant you query access to his or her DEPARTMENTS table.

ALTER Grant SELECT ON departments TO week;

7. Query all the rows in your DEPARTMENTS table.

SELECT * FROM department

8. Add a new row to your DEPARTMENTS table. Team 1 should add Education as department number 500. Team 2 should add Human Resources department number 510. Query the other team's table.

Insert INTO department (department_id, department_name) values (500 ,'Eduction');

INSERT INTO departments (departments_id, department_name) values (510, 'Human relonces');

commit

SELECT * FROM kam2 department;

9. Query the USER_TABLES data dictionary to see information about the tables that you own.

SELECT table_name ,table space_name, num_rows FROM us_table;

10. Revoke the SELECT privilege on your table from the other team.

Revoke select on department from usb2.

11. Remove the row you inserted into the DEPARTMENTS table in step 8 and save the changes.

```
DELETE FROM departments
WHERE departments - id = 500;
Commit;
```

Evaluation Procedure	Marks awarded
Practice Evaluation (5)	5
Viva(5)	5
Total (10)	10
Faculty Signature	

PROGRAM 1

Write a PL/SQL block to calculate the incentive of an employee whose ID is 110.

```
DECLARE
    . V - emp_id    Number := 110;
      V - salary    Number;
      V -inactive   Number

BEGIN
      SELECT  salary
      INTO  V - salary
      FROM  employees
      WHERE employee -id = v_emp -id;
      V = inactive : = V - salary 0.10;
  DBMS _ out Put . Put - LINE ('Employee ID: || v-emp_id);
  DBMS - Output _ Put - Line (salary : ' || V - salary );
  DBMS - output . Put - LINE ( "INactive (10%) : ' || v-
                                          inactives;

Exception :
     WHEN NO DATA - FOUND THEN
          DBMS_ OutPut Put - LINE (' NO Employee sa
                                          with ID' ||
     V - emP_id ) ;
          WHEN Others  THEN
                 DBMS_ output Put - LINE ('Error
     END;

  /
```

PROGRAM 2

Write a PL/SQL block to show an invalid case-insensitive reference to a quoted and without quoted user-defined identifier.

```
DECLARE
    "My variable" NUMBER := 500;
BEGIN
    DBMS_output.put_CLANE('My variable');
END;
/
```

PROGRAM 3

Write a PL/SQL block to adjust the salary of the employee whose ID 122.

Sample table: employees

```
DEclare:
    v.old. salary
    v. new salary,
Begin
    Select salary INTO v.old. Salary
    From employe
    where employe - id = 122:
  v -new-salary := v-old -salary 1.10:
  Update employee
  set salary = V- new -salary
Commit,
  Exception
    when no-Data fond Then
      DBMS- outPut - Put_Line ('NoEmployee found
                                              w/x
                                              ID(22').
END;
```

PROGRAM 4

Write a PL/SQL block to create a procedure using the "IS [NOT] NULL Operator" and show AND operator returns TRUE if and only if both operands are TRUE.

```
Declare
    V - num1' Number = 10
    V_num2 Number := NULL:
    V = result BOOLEAN;

  Procedure Check_null - and - logic is
 Begin
    If (v.num) is not null is
   DBms output. Put_LINE ('v_num1 is Not null');
  Else
    DMS_ outPUT . Put - LINE ('V_num1 IS N none)':
  END IF ;
     If ( V_num2 is NOT null] AND (V - Num 13s) Then
   DBMS - OUTPut . Put_LINE ('Both anditions are TRUE ->
 AND Ecause )'.
 Else :
   DBMS_ outPut. put - LINE 162 ('one or both condition
                                      are false)
 ENDIF ;
End:
```

PROGRAM 5

Write a PL/SQL block to describe the usage of LIKE operator including wildcard characters and escape character.

```
SET STRVEROUTPUT ON
BEGIN:
    IF 'Raven' LIKE 'Ra%.' THEN
        DBMS- OUTPUT .Put_LINE ('LIKE with) : Name
                                                Starts)
    END IF:
    IF 'Raj LIKE 'R-]" Then
        DBMS_ output-LINE (Like with -Second letter
    can be anything'):
    End IF;
  IF'A-B' LIKE 'A'_B' ESCAPE '\' Then
    DBMS -output .Put -LINE (Like with Escape.
                                  method underscore "_"):
  END if
End;
```

PROGRAM 6

Write a PL/SQL program to arrange the number of two variable in such a way that the small number will store in num_small variable and large number will store in num_large variable.

```
    SET seRued output ON:
DECLARE:
        num1 Number : =25:
        Num2 Number : = 10:
        num_ Small Number:
        num - large Number:
BEGIN
    IF num1 < num2 Then
        num_small: = num1:
        num_Large : = num2:
ElSE
    num_Small . = num2:
    num- large : = num1 :
  END IF
 DBMS- output .Put Line ("smaller number: ' || num (small):
 DBMS- output . Put_Line ("Large number = ' || number -large)-
END
/
```

PROGRAM 7

Write a PL/SQL procedure to calculate the incentive on a target achieved and display the message either the record updated or not.

```
Declar

    Emp_id  Number := 123;
    target  Number := 75000;
    uncentive number;
    row _updated Number;
Begin

    IF target >= 10000 Then
          uncentic := 5000;
     EISEIF target >= 50000   Then
          uncentive := 2000;
Else
        incetive := 0;
ENDif
        update  Emploze
        set   uncentive := Incetie
        where  employee_id = emp_id;
Else:
     DBMS_Output.Put_LINE ('No Record foun for emp id:'||
                                     emp.id);
ENDIF;
END;
```

PROGRAM 8

Write a PL/SQL procedure to calculate incentive achieved according to the specific sale limit.

```
Set  serveroutput on;
Creat or Replace Producer calc_uncitiel
        P.emp_id IN Number,
        P.sales  IN Number
) is
      v_Incentie Number;

Begin  if  P_sales >= 10000 Then
          v_incentive := 500;
      Else _ v_incentive := 0;

Else:
    DBMS_ output. Put_ LINE ('No record foun
 for  employee  ID'|| P_ emp_id);

End IF;
 End ;
 /
```

PROGRAM 9

Write a PL/SQL program to count number of employees in department 50 and check whether this department have any vacancies or not. There are 45 vacancies in this department.

```
SET SERVEROUTPUT ON;
Declare
        V_em_count Number:
        V_vacaries Number:=45:
        V- ramaing number;
Begin
    select count (*) INTO V_emp_count;
DBMS_output.Put_LINE ('Number of employe in department 50'||
IF V_remaining >0 Then
    DBMS_OUTPUT.Put_LINE ("Vaconies available:'|| v_remaing);
Else
    DBMS_output.Put_LINE ("no vaccenies in Department 50);
End IF;
End;
/
```

PROGRAM 10

Write a PL/SQL program to count number of employees in a specific department and check whether this department have any vacancies or not. If any vacancies, how many vacancies are in that department.

```
Set serveroutput on;
declare
        v_dept_id Number:=a dept_id;
        v_emp_count Number;
        V: total - Posts Number:=45.
        v - vacencies Number:
Begin   select count(V) INTO v_emp_count
        from employee
        where department_id = V_dept_id;
        V_vacancies := v_total_posts_v_emp_count;

  ElSE
        DBMS.output.Put_LINE('No vacconies in the depart
END IF:
EXCEPTION
        when no_data_found Then
        DBMS.Output.Put_LINE ("Invalid Department ID
        when Others Then
        DBMS.Output.Put_LINE('Error: '|| SQLERRM);
  END;
  /
```

PROGRAM 11

Write a PL/SQL program to display the employee IDs, names, job titles, hire dates, and salaries of all employees.

```
set   Serveroutput on.
declare:
      V_emp_id   employee. employee_id %.type:
      V_name   employee. first_name %.Type:
      V_job   employee job_id % Type
      V salary  employee .salary %. Type:
Cursor  emp_cur IS
    select  employee_id. first_name.'job_id. hire.date.salary
  From  employee:
Begin
   DBMS_output.Put_Line ('Emp_id . Name  JOB ID  hire_Date
For  emp_rec in  emp_ Loop                                  salary);
    DBMS_output .Put_Line(
        RPad (emp_rec. job_id, 12) ||' ' ||
        To_char (emp_rec. hire_date , 'DD_MON_YYYY')||' '||
        emp_rec.salary
    ):
  End  Loop:
End ;
/
```

PROGRAM 12

Write a PL/SQL program to display the employee IDs, names, and department names of all employees.

```
set  Serveroutput  ON:
declare
      Cursor  emp_cur IS
      select  e.employee_id.
       e.first_name.
       d.department_name
       from employee e
       join  department d
       on  e.department_id = de.department.id
Begin
    DBms_output.Put_Line
    Dbms_output. Put_line ('_ _ _ _ _ _ _ _ _ _ _ _ _
    For .emp_rec   IN   emp_cur Loop

             emp_rec. employee_id || " ||
             RPAD (emp_rec .first_name. 12)||' ',
              emp_ rec. department_name
    );
   End Loop;
  End;
 /
```

PROGRAM 13

Write a PL/SQL program to display the job IDs, titles, and minimum salaries of all jobs.

```
set serveroutput on;
declare
    cursor job-cur is
        select job-id job-title, min salary
    From jobs;
Begin DBMS_ output. Put_Line ('JOB_ID  JOB-TITLE MIN');
DBMS_ output Put-Line('- - - - - - - - - - -');
   For job-rec IN job-cur loop

        RPAD (job_rec_job-title.25) ||' '||
        Job_rec. min_salary

  );
 End Loop;
End;
/
```

PROGRAM 14

Write a PL/SQL program to display the employee IDs, names, and job history start dates of all employees.

```
set serveroutput on;
declare
        cursor emp_ is
        Select e.employee-id
        e.first_name
        jh.start-date.
        From employee e
        JOIN job-history jh

BEGIN
   DBMS_output.Put_Line('Emp_ID Name Start-date');
  DBMS_output.Put_Line('- - - - - - - - - - - -');
     FOR emp-rec IN emp_cur Loop
     DBMS_output.Put_LINE(
     emp_rec employee_id ||' '||
     RPad(emp_rec first_name.12) || ' ' ||

);
End Loop;
End;
/
```