Database Management System

EXPERIMENT 4

NAME: Madhan RV

ROLL: 241801142

First, let's verify our tables have data:

```
-- Check EMPLOYEES table
SELECT COUNT(*) FROM employees;
```

Expected Output:

```
 COUNT(*)
----------
         6
 -- Check DEPARTMENT table
SELECT COUNT(*) FROM department;
```

Expected Output:

```
 COUNT(*)
----------
         7
```

Basic SELECT Statements:

1. Show the structure of departments table and select all data

```
DESC department;
```

Expected Output:

```
Name         Null?    Type
------------ -------- -----------
DEPT_ID      NOT NULL NUMBER(6)
DEPT_NAME    NOT NULL VARCHAR2(20)
MANAGER_ID            NUMBER(6)
LOCATION_ID           NUMBER(4)
 SELECT * FROM department;
```

Expected Output:

```
  DEPT_ID DEPT_NAME               MANAGER_ID LOCATION_ID
---------- -------------------- ---------- -----------
        10 Administration                200        1700
        20 Marketing                     201        1800
        50 Shipping                      124        1500
        60 IT                            103        1400
        80 Sales                         149        2500
        90 Executive                     100        1700
       110 Accounting                    205        1700
```

2. Show the structure of employees table and display specific columns

```
DESC employees;
```

Expected Output:

```
Name            Null?    Type
--------------- -------- ------------
EMPLOYEE_ID     NOT NULL NUMBER(6)
```

```
FIRST_NAME                VARCHAR2(20)
LAST_NAME       NOT NULL  VARCHAR2(25)
EMAIL           NOT NULL  VARCHAR2(25)
PHONE_NUMBER              VARCHAR2(20)
HIRE_DATE       NOT NULL  DATE
JOB_ID          NOT NULL  VARCHAR2(10)
SALARY                    NUMBER(8,2)
COMMISSION_PCT            NUMBER(2,2)
MANAGER_ID                NUMBER(6)
DEPARTMENT_ID             NUMBER(4)
```

```
SELECT employee_id, last_name, job_id, hire_date
FROM employees
ORDER BY employee_id;
```

Expected Output:

```
EMPLOYEE_ID LAST_NAME                 JOB_ID     HIRE_DATE
----------- ------------------------- ---------- ---------
        100 King                      AD_PRES    17-JUN-87
        101 Kochhar                   AD_VP      21-SEP-89
        102 De Haan                   AD_VP      13-JAN-93
        103 Hunold                    IT_PROG    03-JAN-90
        104 Ernst                     IT_PROG    21-MAY-91
        107 Lorentz                   IT_PROG    07-FEB-99
```

3. Provide an alias STARTDATE for the hire date

```
SELECT employee_id, last_name, job_id, hire_date AS
startdate
FROM employees;
```

Expected Output:

```
EMPLOYEE_ID LAST_NAME                 JOB_ID     STARTDATE
----------- ------------------------- ---------- ---------
        100 King                      AD_PRES    17-JUN-87
        101 Kochhar                   AD_VP      21-SEP-89
        102 De Haan                   AD_VP      13-JAN-93
        103 Hunold                    IT_PROG    03-JUN-90
        104 Ernst                     IT_PROG    21-MAY-91
        107 Lorentz                   IT_PROG    07-FEB-99
```

4. Display unique job codes from the employee table

```
SELECT DISTINCT job_id FROM employees;
```

Expected Output:

```
JOB_ID
----------
AD_PRES
AD_VP
IT_PROG
```

5. Display last name concatenated with job ID

```
SELECT last_name || ', ' || job_id AS "Employee and Title"
FROM employees;
```

Expected Output:

```
Employee and Title
-----------------------------------
King, AD_PRES
Kochhar, AD_VP
De Haan, AD_VP
```

```
Hunold, IT_PROG
Ernst, IT_PROG
Lorentz, IT_PROG
```

Advanced SELECT Features:

6. Display all data from employees with comma separation

```
SELECT employee_id || ', ' || first_name || ', ' ||
last_name || ', ' ||
       email || ', ' || phone_number || ', ' || hire_date ||
', ' ||
       job_id || ', ' || salary || ', ' || commission_pct ||
', ' ||
       manager_id || ', ' || department_id AS "THE_OUTPUT"
FROM employees;
```

Expected Output:

```
THE_OUTPUT
--------------------------------------------------------------
----------------------------
100, Steven, King, SKING, 515.123.4567, 17-JUN-87, AD_PRES,
24000, , , 90
101, Neena, Kochhar, NKOCHHAR, 515.123.4568, 21-SEP-89,
AD_VP, 17000, , 100, 90
102, Lex, De Haan, LDEHAAN, 515.123.4569, 13-JAN-93, AD_VP,
17000, , 100, 90
103, Alexander, Hunold, AHUNOLD, 590.423.4567, 03-JAN-90,
IT_PROG, 9000, , 102, 60
104, Bruce, Ernst, BERNST, 590.423.4568, 21-MAY-91, IT_PROG,
6000, , 103, 60
107, Diana, Lorentz, DLORENTZ, 590.423.4568, 07-FEB-99,
```

```
IT_PROG, 4200, , 103, 60
```

Arithmetic Operations:

7. Basic arithmetic operations with salaries

```
SELECT last_name, salary, salary + 300 AS "Salary+300"
FROM employees;
```

Expected Output:

```
LAST_NAME                      SALARY  Salary+300
------------------------- ---------- ----------
King                            24000       24300
Kochhar                         17000       17300
De Haan                         17000       17300
Hunold                           9000        9300
Ernst                            6000        6300
Lorentz                          4200        4500
```

8. Complex arithmetic expressions

```
SELECT last_name, salary, 12 * salary + 100 AS "Annual1",
       12 * (salary + 100) AS "Annual2"
FROM employees;
```

Expected Output:

```
LAST_NAME                      SALARY    Annual1    Annual2
------------------------- ---------- ---------- ----------
King                            24000     288100     289200
Kochhar                         17000     204100     205200
De Haan                         17000     204100     205200
```

```
Hunold                          9000     108100     109200
Ernst                           6000      72100      73200
Lorentz                         4200      50500      51600
```

9. Working with commission percentages

```
SELECT last_name, job_id, salary, commission_pct,
       12 * salary * commission_pct AS "Annual Commission"
FROM employees
WHERE commission_pct IS NOT NULL;
```

Expected Output:

```
no rows selected
```

(Let's add some employees with commission for demonstration)

```
INSERT INTO employees VALUES
(149, 'Eleni', 'Zlotkey', 'EZLOTKEY', '011.44.1344.429018',
 TO_DATE('29-JAN-2000', 'DD-MON-YYYY'), 'SA_MAN', 10500,
0.2, 100, 80);

INSERT INTO employees VALUES
(174, 'Ellen', 'Abel', 'EABEL', '011.44.1644.429267',
 TO_DATE('11-MAY-1996', 'DD-MON-YYYY'), 'SA_REP', 11000,
0.3, 149, 80);

COMMIT;
```

Now run the commission query again:

```
SELECT last_name, job_id, salary, commission_pct,
       12 * salary * commission_pct AS "Annual Commission"
```

```
FROM employees
WHERE commission_pct IS NOT NULL;
```

Expected Output:

```
LAST_NAME                 JOB_ID         SALARY
COMMISSION_PCT Annual Commission
------------------------- ---------- ---------- ------------
-- ---------------
Zlotkey                   SA_MAN
10500             .2            25200
Abel                      SA_REP
11000             .3            39600
```

Column Aliases and Literals:

10. Using column aliases with and without AS keyword

```
SELECT last_name AS "Name", salary "Monthly Salary",
       salary * 12 "Yearly Salary"
FROM employees;
```

Expected Output:

```
Name                       Monthly Salary Yearly Salary
------------------------- -------------- -------------
King                                24000        288000
Kochhar                             17000        204000
De Haan                             17000        204000
Hunold                               9000        108000
Ernst                                6000         72000
Lorentz                              4200         50400
Zlotkey                             10500        126000
```

```
Abel                                11000        132000
```

11. Using literal character strings

```
SELECT last_name || ' is a ' || job_id AS "Employee Details"
FROM employees;
```

Expected Output:

```
Employee Details
---------------------------------
King is a AD_PRES
Kochhar is a AD_VP
De Haan is a AD_VP
Hunold is a IT_PROG
Ernst is a IT_PROG
Lorentz is a IT_PROG
Zlotkey is a SA_MAN
Abel is a SA_REP
```

12. More complex literal example

```
SELECT last_name || '''s annual salary is $' ||
       TO_CHAR(salary * 12, '999,999') AS "Salary
Information"
FROM employees
WHERE ROWNUM <= 3;
```

Expected Output:

```
Salary Information
---------------------------------------
King's annual salary is $288,000
```

```
Kochhar's annual salary is $204,000
De Haan's annual salary is $204,000
```

Working with NULL Values:

13. Handling NULL values in expressions

```
SELECT last_name, salary, commission_pct,
       salary * (1 + NVL(commission_pct, 0)) AS "Total
Compensation"
FROM employees;
```

Expected Output:

```
LAST_NAME                      SALARY COMMISSION_PCT Total
Compensation
------------------------- ---------- ------------- --------
----------
King                            24000
24000
Kochhar                         17000
17000
De Haan                         17000
17000
Hunold                           9000
9000
Ernst                            6000
6000
Lorentz                          4200
4200
Zlotkey                         10500            .2
12600
Abel                            11000            .3
```

```
14300
```

Additional SELECT Features:

14. Using DUAL table for calculations

```
SELECT 15 * 25 AS "Calculation",
       SYSDATE AS "Current Date",
       USER AS "Current User"
FROM DUAL;
```

Expected Output:

```
Calculation Current Da Current User
----------- --------- --------------------
        375 23-JAN-24 SYSTEM
```

15. String manipulation with DUAL

```
SELECT 'Hello ' || 'World' AS "Greeting",
       LENGTH('Oracle Database') AS "String Length",
       UPPER('hello world') AS "Uppercase"
FROM DUAL;
```

Expected Output:

```
Greeting      String Length UPPERCASE
------------ ------------- ------------
Hello World             14 HELLO WORLD
```

16. Display table structure using data dictionary

```
SELECT column_name, data_type, data_length, nullable
FROM user_tab_columns
WHERE table_name = 'EMPLOYEES'
ORDER BY column_id;
```

Expected Output:

```
COLUMN_NAME                    DATA_TYPE    DATA_LENGTH
NULLABLE
------------------------------ ------------ ----------- ----
----
EMPLOYEE_ID                    NUMBER                6 N
FIRST_NAME                     VARCHAR2             20 Y
LAST_NAME                      VARCHAR2             25 N
EMAIL                          VARCHAR2             25 N
PHONE_NUMBER                   VARCHAR2             20 Y
HIRE_DATE                      DATE                 7 N
JOB_ID                         VARCHAR2             10 N
SALARY                         NUMBER               8 Y
COMMISSION_PCT                 NUMBER               5 Y
MANAGER_ID                     NUMBER               6 Y
DEPARTMENT_ID                  NUMBER               4 Y
```

Final Comprehensive Example:

17. Complete employee information display

```
SELECT
    'Employee: ' || last_name || ', ' || first_name AS "Full
Name",
    'Job: ' || job_id AS "Position",
    'Hired: ' || TO_CHAR(hire_date, 'Month DD, YYYY') AS
"Start Date",
```

```
    'Salary: $' || TO_CHAR(salary, '999,999') AS "Monthly
Pay",
    'Department: ' || department_id AS "Dept ID"
FROM employees
ORDER BY last_name;
```

Expected Output:

```
Full Name                          Position         Start
Date           Monthly Pay      Dept ID
---------------------------------- ---------------- --------
------------ ---------------- ---------
Employee: Abel, Ellen              Job: SA_REP      May
11, 1996   Salary: $11,000  Department: 80
Employee: De Haan, Lex             Job: AD_VP       January
13, 1993   Salary: $17,000  Department: 90
Employee: Ernst, Bruce             Job: IT_PROG     May
21, 1991   Salary: $ 6,000  Department: 60
Employee: Hunold, Alexander        Job: IT_PROG     January
03, 1990   Salary: $ 9,000  Department: 60
Employee: King, Steven             Job: AD_PRES     June
17, 1987   Salary: $24,000  Department: 90
Employee: Kochhar, Neena           Job: AD_VP
September 21, 1989   Salary: $17,000  Department: 90
Employee: Lorentz, Diana           Job: IT_PROG     February
07, 1999   Salary: $ 4,200  Department: 60
Employee: Zlotkey, Eleni           Job: SA_MAN      January
29, 2000   Salary: $10,500  Department: 80
```

Summary of SELECT Statement Capabilities:

1. **Projection** - Choosing specific columns
2. **Column Aliases** - Renaming column headings

3. **Arithmetic Operations** - Calculations in SELECT
4. **Concatenation** - Combining strings and columns
5. **Literal Values** - Including constant values
6. **DISTINCT** - Eliminating duplicate rows
7. **DUAL Table** - For calculations and functions
8. **NULL Handling** - Dealing with null values
9. **Formatting** - Using functions for better display