/DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
CMR TECHNICAL CAMPUS
Kandlakoya (V), Medchal Road, Hyderabad

Course: B.Tech **Date:26-06-2024**
Year: II-II-Sem
DBMS LAB INTERNAL EXAM QUESTION PAPER

1. i. Design a E-R Diagram for Hospital Management System
ii. Perform the following Queries—

a) Find the names of the Sailors who have reserved the boat no 103.

Ans:

- `Sailors(sid, sname, rating, age)`
- `Reserves(sid, bid, day)`
- `Boats(bid, bname, color)`

```
SELECT s.sname
FROM Sailors s
JOIN Reserves r ON s.sid = r.sid
WHERE r.bid = 103;
```

b) Write a query to perform left and right outer join operations on emp and dept tables

Assuming the `emp` (employees) table has columns `emp_id, emp_name, dept_id` and the `dept` (department) table has columns `dept_id, dept_name`, the queries would be:

```
//left
SELECT e.emp_id, e.emp_name, d.dept_name
FROM emp e
LEFT JOIN dept d ON e.dept_id = d.dept_id;
//right
SELECT e.emp_id, e.emp_name, d.dept_name
FROM emp e
RIGHT JOIN dept d ON e.dept_id = d.dept_id;
```

iii. Write a PL/SQL code block to find reverse of a number.

2. i. Design a E-R Diagram for Online Shopping
ii. Perform the following Queries—

a) Find all sailors with a rating above 7.

```
SELECT sname
FROM Sailors
WHERE rating > 7;
```

b) Write a query to perform following functions
i)ascending order ii)descending order iii)concat iv)ltrim

```
///asc
SELECT emp_id, emp_name, salary
FROM Employees
ORDER BY salary ASC;
//desc
SELECT emp_id, emp_name, salary
FROM Employees
ORDER BY salary DESC;

-- Concatenate emp_name and salary
SELECT CONCAT('computer' ,'science') FROM DUAL;
```

-- Left trim select LTRIM('shiva','s') from dual;

iii. write a PL/SQL program to find greatest of 3 numbers.

3. i. Design the following Database using E-R Diagram

Database Schema for a Employee-pay scenario

Employee (**emp_id : integer**, emp_name: string)
Department (**dept_id: integer**, dept_name:string)
Paydetails (**emp_id : integer**, **dept_id: integer**, basic: integer, deductions: integer, additions: integer, DOJ: date)
Payroll (**emp_id : integer**, pay_date: date)

ii. For the above schema, perform the following—

a) Find the names of sailors who have reserved at least one boat.

b) ● Sailors(sid, sname, rating, age)

c) ● Reserves(sid, bid, day)

```
SELECT DISTINCT s.sname
FROM Sailors s
JOIN Reserves r ON s.sid = r.sid;
```

d) Write a query to perform Natural join operation from above tables

Assuming we want to perform a natural join between the `Sailors` and `Reserves` tables based on the common column `sid`, the query would be:

```
SELECT *
FROM Sailors
NATURAL JOIN Reserves;
```

iii. Write a PL/SQL Program using if Condition.

```
DECLARE
  v_number NUMBER := -5; -- Example number to check
BEGIN
  IF v_number > 0 THEN
    DBMS_OUTPUT.PUT_LINE('The number ' || v_number || ' is positive.');
  ELSIF v_number < 0 THEN
    DBMS_OUTPUT.PUT_LINE('The number ' || v_number || ' is negative.');
  ELSE
    DBMS_OUTPUT.PUT_LINE('The number is zero.');
  END IF;
END;
/
```

4. i. Design a E-R Model for Bus Reservation System

ii. Perform the following Queries—

a) Write a query to demonstrate about the following constraints

i)Unique

```
CREATE TABLE Employees (
    emp_id NUMBER PRIMARY KEY,
    emp_name VARCHAR2(100),
    email VARCHAR2(100) UNIQUE
);
```

ii)Not Null

```
CREATE TABLE Employees (
    emp_id NUMBER PRIMARY KEY,
    emp_name VARCHAR2(100) NOT NULL,
    email VARCHAR2(100)
);
```

iii) Foreign Key

```
CREATE TABLE Departments (
    dept_id NUMBER PRIMARY KEY,
    dept_name VARCHAR2(100) NOT NULL
);
```

```
CREATE TABLE Employees (
    emp_id NUMBER PRIMARY KEY,
    emp_name VARCHAR2(100) NOT NULL,
    email VARCHAR2(100),
    dept_id NUMBER,
    CONSTRAINT fk_dept FOREIGN KEY (dept_id) REFERENCES Departments(dept_id)
);
```

b)Find the names of sailors who have reserved a red or a green boat.

- Sailors(sid, sname, rating, age)
- Reserves(sid, bid, day)
- Boats(bid, bname, color)

```
SELECT DISTINCT s.sname
FROM Sailors s
JOIN Reserves r ON s.sid = r.sid
JOIN Boats b ON r.bid = b.bid
WHERE b.color IN ('red', 'green');
```

iii. Write a PL/SQL program using Nested Loop.

```
DECLARE
    outer_num NUMBER;
    inner_num NUMBER;
BEGIN
    -- Outer loop from 1 to 5
    FOR outer_num IN 1..5 LOOP
        DBMS_OUTPUT.PUT_LINE('Multiplication Table for ' || outer_num || ':');

        -- Inner loop from 1 to 5
        FOR inner_num IN 1..5 LOOP
            DBMS_OUTPUT.PUT_LINE(outer_num || ' * ' || inner_num || ' = ' || (outer_num * inner_num));
        END LOOP;

        DBMS_OUTPUT.PUT_LINE(''); -- Print a blank line for better readability
    END LOOP;
END;
/
```

5. i. Design a E-R Diagram for University

ii. Perform the following Queries—

a) Find the names of sailors who have reserved both a red and green boat.

b) • Sailors(sid, sname, rating, age)

c) • Reserves(sid, bid, day)

d) • Boats(bid, bname, color)

```
SELECT s.sname
FROM Sailors s
JOIN Reserves r1 ON s.sid = r1.sid
JOIN Boats b1 ON r1.bid = b1.bid AND b1.color = 'red'
JOIN Reserves r2 ON s.sid = r2.sid
JOIN Boats b2 ON r2.bid = b2.bid AND b2.color = 'green';
```

e) Write a Query to perform any five Character Functions.

```
SELECT LOWER('GEEKSFORGEEKS') FROM DUAL;
```

```
SELECT UPPER('geeksforgeeks') FROM DUAL;
SELECT SUBSTR('Database Management System', 9) FROM DUAL;
```

```
SELECT LENGTH('Learning Is Fun') FROM DUAL;
```

```
SELECT CONCAT('computer' ,'science') FROM DUAL;
```

iii. Create a trigger for each row to perform before and after insertion operations.

6. i. Design a E-R Model for Banking Management System

ii. Perform the following Queries—

a) Find sailors whose rating is better than some sailors called “Horatio”.

```
SELECT sname
FROM Sailors WHERE rating > (SELECT MIN(rating) FROM Sailors WHERE sname = 'Horatio');
```

b) Write a query to perform all number functions.

Abs()
Asin()
Acos
Atan
cos
Ceil
floor

iii. Create a trigger before/after update on employee table for each row/statement.

7. i. Design the following Database using E-R Diagram

Database Schema for a customer-sale scenario

Customer(**Cust id : integer**, cust_name: string)

Item(**item_id: integer**, item_name: string, price: integer)

Sale(**bill_no: integer**, bill_data: date, **cust_id: integer**, **item_id: integer**, qty_sold: integer)

ii. Perform the following Queries—

a) Find the colors of boats reserved by Lubber.

```
SELECT DISTINCT b.color
FROM Sailors s
JOIN Reserves r ON s.sid = r.sid
JOIN Boats b ON r.bid = b.bid
WHERE s.sname = 'Lubber';
```

b) Find the names of sailors who have reserved a red but not a green boat.

```
SELECT DISTINCT s.sname
FROM Sailors s
JOIN Reserves r ON s.sid = r.sid
JOIN Boats b ON r.bid = b.bid
WHERE b.color = 'red'
  AND NOT EXISTS (
    SELECT 1
    FROM Reserves r2
    JOIN Boats b2 ON r2.bid = b2.bid
    WHERE r2.sid = s.sid
      AND b2.color = 'green'
  );
```

iii. Write a PL/SQL program for factorial of a number.

8. i. Design a E-R Model for Library Management System

ii. Perform the following Queries—

a) Find the names of sailors who have reserved a red boat.

```
SELECT DISTINCT s.sname
FROM Sailors s
JOIN Reserves r ON s.sid = r.sid
JOIN Boats b ON r.bid = b.bid
WHERE b.color = 'red';
```

b) Find the ages of sailors whose name begins and ends with B and has not at least three characters.

```
SELECT s.age
FROM Sailors s
WHERE s.sname LIKE 'B%' AND s.sname LIKE '%B' AND LENGTH(s.sname) < 3;
```

iii. Write a PL/SQL program to find greatest of 3 numbers.

9.a. Write a query to display unique jobs from emp table

```
SELECT DISTINCT job FROM emp;
```

b. Fine the names of sailors who have reserved boat no 103

```
SELECT s.sname
FROM Sailors s
JOIN Reserves r ON s.sid = r.sid WHERE r.bid = 103;
```

c. Write a query to perform equi join operation on emp and dept tables

```
SELECT e.emp_id, e.emp_name, e.job, e.sal, d.dept_name, d.location
FROM emp e
JOIN dept d ON e.dept_id = d.dept_id;
```

d. Write a PL/SQL program for sum and average of two numbers.

10. a. List the employee details in ascending order of their salaries.

```
SELECT emp_id, emp_name, salary
FROM Employees
ORDER BY salary ASC;
```

b. Find the names of the sailors who have not reserves boat no 103.

```
SELECT s.sname
FROM Sailors s
WHERE s.sid NOT IN (SELECT r.sid FROM Reserves r WHERE r.bid = 103);
```

c. Write a query to perform natural join operation on emp and dept tables

```
SELECT* from emp Natural join dept
;
```

d. Write a PL/SQL code using simple case statement

11.a. List the employee details who are working for deptno 10 or 20 (IN)

```
SELECT emp_id, emp_name, job, mgr, hiredate, sal, comm, dept_id

FROM emp

WHERE dept_id IN (10, 20);
```

b. Find the names of the sailors who have reserves atleast one boat.

```
SELECT DISTINCT s.sname
FROM Sailors s
JOIN Reserves r ON s.sid = r.sid;
```

c. Write a query to perform outer join operation on emp and dept table

d. Write a PL/SQL program to find simple interest

12.a.List the enames starting with S and with 5 characters

```
SELECT ename

FROM emp WHERE ename LIKE 'S____';
```

b. Find all sids who have a rating of 10 or reserved boat 104

```
SELECT DISTINCT s.sid FROM Sailors s LEFT JOIN Reserves r ON s.sid = r.sidWHERE
s.rating = 10 OR r.bid = 104;
```

c. Write a query to perform following character functions

i)upper ii)initcap iii)concat iv)length

```
SELECT INITCAP('geeksforgeeks is a computer science portal for
geeks') FROM DUAL;
```

d. Write a PL/SQL program to find sum and average .

13.a. List the ename who is not getting any commission

```
SELECT ename

FROM emp

WHERE comm IS NULL OR comm = 0;
```

b. Find the sailor whose rating is better than some sailor called horatio

```
SELECT sname

FROM Sailors

WHERE rating > (SELECT MIN(rating) FROM Sailors WHERE sname = 'Horatio');
```

c. Write a Query to perform all aggregate functions

count
sum
avg
min
max

d. Write a trigger for each row before and after delete.

14.a.Write a query to display the employee details whose commission is greater than salary

```
SELECT emp_id, ename, job, mgr, hiredate, sal, comm, dept_id

FROM emp

WHERE comm > sal;
```

b. Find the sailors whose rating is better than every sailor called “horatio”.

```
SELECT sname FROM SailorsWHERE rating > (SELECT MIN(rating) FROM Sailors
WHERE sname = 'Horatio');
```

c. Write a query to demonstrate about the following constraints

i)Unique ii)Not Null iii) Check

```
CREATE TABLE example_table (
   id INT PRIMARY KEY,
   age INT CHECK (age >= 18)
);
```

d. Write a PL/SQL program using Case Selector.