

## **\*\*Practical Questions in DBMS\*\***

### **1. \*\*School Database:\*\***

- Create a database named "School Database."
- Create a table "Student\_Info" with fields: Name, Class, Address, PhoneNo, Section, RollNo, and DoB.
- Insert at least 10 records.
- Write a query to display all records.

### **2. \*\*Library Database:\*\***

- Create a database "Library Database."
- Create a table "tbl\_Book" with fields: Book\_Id, Book\_Name, Author\_Name, Publication\_Name, Published\_Year, Price, ISBN.
- Insert at least 10 records.
- Write a query to display all records where Author\_Name is "Ramesh."
- Prepare a form based on the table.
- Prepare a report based on the query.

### **3. \*\*SQL Queries for Student Table:\*\***

- Create a database named "Student."
- Create a table "Student\_Data" with fields: SID, Name, Address, Faculty, Contact.
- Insert three records.

### **4. \*\*Employee Table Queries:\*\***

- Write SQL query to delete the record of "Arjun."
- Write SQL query to arrange records in ascending order according to "Ename."
- Write SQL query to update Prakash's salary to 7000.
- Write SQL query to display the final table.

## 5. **\*\*Employee Database:\*\***

- Create a database named "employee" and table "staff" with fields: Emp\_No, Name, Rank, Date\_of\_Join, Salary.

- Insert five records.

- Prepare a query to display Name, Salary, and Tax (13% tax if salary  $\geq$  15000, otherwise 1%).

\*\*\*\*\*

# Answers DBMS

\*\*\*\*\*

## DBMS Solutions

### 1) School Database

#### SQL Commands

-- Create the database

```
CREATE DATABASE School_Database;  
USE School_Database;
```

-- Create the table

```
CREATE TABLE Student_Info (  
    Name VARCHAR(50),  
    Class INT,  
    Address VARCHAR(100),  
    PhoneNo VARCHAR(15),  
    Section CHAR(1),  
    RollNo INT PRIMARY KEY,  
    DoB DATE  
);
```

-- Insert at least 10 records

```
INSERT INTO Student_Info VALUES  
( 'Rajesh', 12, 'Kathmandu', '9801234567', 'A', 1, '2006-05-10'),  
( 'Sita', 12, 'Lalitpur', '9807654321', 'B', 2, '2006-07-15'),  
( 'Bikash', 12, 'Bhaktapur', '9812345678', 'A', 3, '2006-09-20'),  
( 'Aarati', 12, 'Pokhara', '9803456789', 'C', 4, '2006-03-05'),  
( 'Suraj', 12, 'Butwal', '9804567890', 'B', 5, '2006-01-25'),  
( 'Priya', 12, 'Chitwan', '9805678901', 'C', 6, '2006-06-30'),  
( 'Niraj', 12, 'Dharan', '9806789012', 'A', 7, '2006-02-15'),  
( 'Manish', 12, 'Jhapa', '9807890123', 'B', 8, '2006-04-10'),  
( 'Anjali', 12, 'Biratnagar', '9808901234', 'C', 9, '2006-08-22'),  
( 'Pooja', 12, 'Janakpur', '9809012345', 'A', 10, '2006-11-05');
```

-- Display all records

```
SELECT * FROM Student_Info;
```

---

## 2) Library Database

### SQL Commands

```
-- Create the database
CREATE DATABASE Library_Database;
USE Library_Database;

-- Create the table
CREATE TABLE tbl_Book (
    Book_Id INT PRIMARY KEY,
    Book_Name VARCHAR(100),
    Author_Name VARCHAR(100),
    Publication_Name VARCHAR(100),
    Published_Year INT,
    Price DECIMAL(8,2),
    ISBN VARCHAR(20)
);

-- Insert at least 10 records
INSERT INTO tbl_Book VALUES
(1, 'Database Management', 'Ramesh', 'Oxford', 2020, 750.00, '978-1-12345-678-9'),
(2, 'Programming in C', 'Suresh', 'Pearson', 2018, 550.00, '978-1-23456-789-0'),
(3, 'Java Fundamentals', 'Mahesh', 'McGraw-Hill', 2019, 900.00, '978-1-34567-890-1'),
(4, 'Python for Beginners', 'Ramesh', 'Packt', 2021, 650.00, '978-1-45678-901-2'),
(5, 'Computer Networks', 'Rajesh', 'Cambridge', 2017, 800.00, '978-1-56789-012-3'),
(6, 'Operating System Concepts', 'Ramesh', 'Wiley', 2015, 1000.00, '978-1-67890-123-4'),
(7, 'Data Science Essentials', 'Ganesh', 'Springer', 2022, 1200.00, '978-1-78901-234-5'),
(8, 'Machine Learning', 'Ramesh', 'O'Reilly', 2023, 1300.00, '978-1-89012-345-6'),
(9, 'Artificial Intelligence', 'Kiran', 'MIT Press', 2019, 1100.00, '978-1-90123-456-7'),
(10, 'Cybersecurity Basics', 'Dinesh', 'Pearson', 2020, 950.00, '978-1-01234-567-8');

-- Display all books written by "Ramesh"
SELECT * FROM tbl_Book WHERE Author_Name = 'Ramesh';
```

---

## 3) Student Table

### SQL Commands

```
-- Create database
CREATE DATABASE Student;
USE Student;

-- Create table
CREATE TABLE Student_Data (
    SID INT PRIMARY KEY,
    Name VARCHAR(50),
    Address VARCHAR(100),
    Faculty VARCHAR(50),
    Contact VARCHAR(15)
);
```

```
-- Insert three records
INSERT INTO Student_Data VALUES
(1, 'Rohan', 'Kathmandu', 'Science', '9801234567'),
(2, 'Meera', 'Lalitpur', 'Management', '9812345678'),
(3, 'Sunil', 'Bhaktapur', 'Humanities', '9823456789');

-- Display all records
SELECT * FROM Student_Data;
```

---

## 4) Employee Table Queries

### SQL Commands

```
-- Delete record of Arjun
DELETE FROM Employee WHERE Ename = 'Arjun';

-- Arrange records in ascending order by Ename
SELECT * FROM Employee ORDER BY Ename ASC;

-- Update salary of Prakash to 7000
UPDATE Employee SET Salary = 7000 WHERE Ename = 'Prakash';

-- Display final table
SELECT * FROM Employee;
```

---

## 5) Employee Tax Calculation

### SQL Commands

```
-- Create employee database and table
CREATE DATABASE employee;
USE employee;

CREATE TABLE staff (
    Emp_No INT PRIMARY KEY,
    Name VARCHAR(50),
    Rank VARCHAR(20),
    Date_of_Join DATE,
    Salary DECIMAL(10,2)
);

-- Insert 5 records
INSERT INTO staff VALUES
(1, 'Ravi', 'Manager', '2010-05-15', 18000),
(2, 'Sita', 'Assistant Manager', '2015-09-22', 14000),
(3, 'Kiran', 'Accountant', '2017-06-12', 16000),
(4, 'Manoj', 'Clerk', '2019-04-18', 12000),
(5, 'Pooja', 'HR', '2021-01-10', 15500);

-- Display name, salary, and calculated tax
SELECT Name, Salary,
    CASE
        WHEN Salary >= 15000 THEN Salary * 0.13
        ELSE Salary * 0.01
    END AS Tax
FROM staff;
```

---

### **\*\*Practical Questions in Web Technology-II\*\***

1. Create a **\*\*registration form\*\*** with name, address, phone, gender, qualification, email, submit, and cancel button (use HTML and style it).
2. **\*\*Design a login form\*\*** and write PHP code for login (use CSS and JavaScript).
3. **\*\*Form validation\*\***: Design a form with username, address, email, password, and validate it.
4. Explain **\*\*JavaScript function\*\*** with an example.
5. Write a **\*\*server-side script\*\*** to create a database, connect to it, create a table, and insert data.
6. Write a **\*\*JavaScript program\*\*** for a simple calculator.
7. Write a **\*\*PHP program\*\*** to display the largest of three numbers.
8. Write a **\*\*PHP program\*\*** to take a name input and display it.
9. Write a **\*\*database connectivity program\*\***.

### **\*\*Practical Questions in C Programming\*\***

1. WAP to check if a number is **\*\*Armstrong\*\*** or not.
2. WAP to find the **\*\*greatest number\*\*** among different numbers using an array and functions.
3. WAP to check if a number is **\*\*even or odd\*\*** using a function.
4. WAP to **\*\*calculate the area of a circle\*\*** using a function.
5. WAP to **\*\*pass two numbers\*\*** to a function and return their sum.
6. WAP to find the **\*\*sum and square\*\*** of two numbers using a function.
7. WAP to calculate the **\*\*sum of n natural numbers\*\*** using recursion.
8. WAP to **\*\*calculate the area of a rectangle\*\*** using a function.
9. WAP to check if a number is **\*\*prime or composite\*\*** using a function.
10. WAP to **\*\*display the Fibonacci series\*\*** using a function.
11. WAP to **\*\*swap two values\*\*** using **\*\*call by value\*\***.

12. WAP to **swap two values** using **call by reference**.
13. WAP to **generate the Fibonacci series** up to the 10th term using a user-defined function.
14. WAP to check if a number is **divisible by 5 but not by 7**.
15. WAP to print the **multiplication table** of a given number.
16. WAP to check if a number is **palindrome**.
17. WAP to find the **factorial** using recursion.
18. WAP to read **salary of 300 employees** and count employees earning **between 30,000 and 50,000**.
19. WAP to **pass pointer variables** to a function, sum them, and display the result.
20. WAP to enter **name, roll number, and percentage of 10 students** and store them in a data file.
21. Create a **data file "patient.txt"** and store name, disease, age, and bed number of a patient.
22. Read **"patient.txt"** and display only records of **patients suffering from COVID**.
23. Enter **book name, book ID, and author** into **"library.txt"**, then display the records.
24. Read and display **all records** from **"student.txt"** (contains name, class, and marks).
25. Read and display only records where **name is "Ram"** from **"student.txt"**.
26. Read and display only students who **passed all subjects** from **"student.txt"**.
27. WAP to allow users to **add records until they press "y"**.
28. Read and display **students who failed at least one subject** from **"student.txt"**.
29. Read and display only **name and percentage** from **"student.txt"**.
30. WAP to read **N student records**, store them in a file, and display them using `fprintf`` and `fscanf``.
31. WAP to read a **sentence until Enter is pressed**, store each word in a data file, and display the content.
32. WAP to read an **existing "student.txt" file** and display its content, or print "File does not exist" if not found.

.....

# Answers

That's a big task, but I got you! ☺

I'll now provide solutions to **all 32 C programming practical questions** from your document.

---

## Practical Questions in C Programming - Answers

### 1) Check if a Number is Armstrong

```
#include <stdio.h>
#include <math.h>

int main() {
    int num, original, remainder, result = 0, n = 0;
    printf("Enter a number: ");
    scanf("%d", &num);
    original = num;

    while (original != 0) {
        original /= 10;
        n++;
    }
    original = num;

    while (original != 0) {
        remainder = original % 10;
        result += pow(remainder, n);
        original /= 10;
    }

    if (result == num)
        printf("%d is an Armstrong number.", num);
    else
        printf("%d is not an Armstrong number.", num);

    return 0;
}
```

---

### 2) Find Greatest Number in an Array Using Functions

```
#include <stdio.h>

int findMax(int arr[], int n) {
    int max = arr[0];
    for (int i = 1; i < n; i++) {
        if (arr[i] > max)
            max = arr[i];
    }
    return max;
}
```

```
int main() {
    int arr[5] = {10, 20, 5, 40, 30};
    printf("Greatest number: %d", findMax(arr, 5));
    return 0;
}
```

---

### 3) Check Even or Odd Using Function

```
#include <stdio.h>

void checkEvenOdd(int num) {
    if (num % 2 == 0)
        printf("%d is Even.", num);
    else
        printf("%d is Odd.", num);
}

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    checkEvenOdd(num);
    return 0;
}
```

---

### 4) Area of Circle Using Function

```
#include <stdio.h>

float areaCircle(float r) {
    return 3.1416 * r * r;
}

int main() {
    float r;
    printf("Enter radius: ");
    scanf("%f", &r);
    printf("Area: %.2f", areaCircle(r));
    return 0;
}
```

---

### 5) Sum of Two Numbers Using Function

```
#include <stdio.h>

int sum(int a, int b) {
    return a + b;
}

int main() {
    int x, y;
    printf("Enter two numbers: ");
    scanf("%d %d", &x, &y);
    printf("Sum: %d", sum(x, y));
    return 0;
}
```

---



## 6) Sum and Square of Two Numbers

```
#include <stdio.h>

void sumAndSquare(int a, int b) {
    printf("Sum: %d\n", a + b);
    printf("Square: %d and %d", a * a, b * b);
}

int main() {
    sumAndSquare(5, 3);
    return 0;
}
```

---

## 7) Sum of First N Natural Numbers Using Recursion

```
#include <stdio.h>

int sumN(int n) {
    if (n == 0)
        return 0;
    return n + sumN(n - 1);
}

int main() {
    printf("Sum of first 10 natural numbers: %d", sumN(10));
    return 0;
}
```

---

## 8) Area of Rectangle Using Function

```
#include <stdio.h>

int area(int l, int w) {
    return l * w;
}

int main() {
    printf("Area: %d", area(5, 10));
    return 0;
}
```

---

## 9) Prime or Composite Number Using Function

```
#include <stdio.h>

void checkPrime(int n) {
    int flag = 1;
    if (n <= 1) flag = 0;
    for (int i = 2; i < n; i++) {
        if (n % i == 0) {
            flag = 0;
            break;
        }
    }
    printf(flag ? "Prime" : "Composite");
}

int main() {
    checkPrime(29);
}
```

```
    return 0;
}
```

---

### 10) Fibonacci Series Using Function

```
#include <stdio.h>

void fibonacci(int n) {
    int a = 0, b = 1, temp;
    for (int i = 1; i <= n; i++) {
        printf("%d ", a);
        temp = a + b;
        a = b;
        b = temp;
    }
}

int main() {
    fibonacci(10);
    return 0;
}
```

---

### 11) Swap Two Numbers Using Call by Value

```
#include <stdio.h>

void swap(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
    printf("After swap: %d %d", a, b);
}

int main() {
    swap(5, 10);
    return 0;
}
```

---

### 12) Swap Two Numbers Using Call by Reference

```
#include <stdio.h>

void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main() {
    int x = 5, y = 10;
    swap(&x, &y);
    printf("After swap: %d %d", x, y);
    return 0;
}
```

---

### 13) Generate Fibonacci Series Up to 10th Term

```
#include <stdio.h>
```

```

void fibonacci(int n) {
    int a = 2, b = 3, temp;
    for (int i = 1; i <= n; i++) {
        printf("%d ", a);
        temp = a + b;
        a = b;
        b = temp;
    }
}

int main() {
    fibonacci(10);
    return 0;
}

```

---

#### 14) Check if a Number is Divisible by 5 but Not by 7

```

#include <stdio.h>

void checkDivisibility(int n) {
    if (n % 5 == 0 && n % 7 != 0)
        printf("Yes");
    else
        printf("No");
}

int main() {
    checkDivisibility(25);
    return 0;
}

```

---

#### 15) Print Multiplication Table Using Function

```

#include <stdio.h>

void printTable(int n) {
    for (int i = 1; i <= 10; i++)
        printf("%d x %d = %d\n", n, i, n * i);
}

int main() {
    printTable(5);
    return 0;
}

```

Alright! 🎉 Continuing with the **remaining 17 C programming answers** to complete all 32 questions.

---

#### 16) Check if a Number is Palindrome

```

#include <stdio.h>

int isPalindrome(int n) {
    int reversed = 0, original = n, remainder;
    while (n != 0) {
        remainder = n % 10;
        reversed = reversed * 10 + remainder;
    }
}

```

```

        n /= 10;
    }
    return (original == reversed);
}

int main() {
    int num = 121;
    printf(isPalindrome(num) ? "Palindrome" : "Not Palindrome");
    return 0;
}

```

---

### 17) Calculate Factorial Using Recursion

```

#include <stdio.h>

long factorial(int n) {
    return (n == 0) ? 1 : n * factorial(n - 1);
}

int main() {
    printf("Factorial: %ld", factorial(5));
    return 0;
}

```

---

### 18) Count Employees Earning Between 30,000 and 50,000

```

#include <stdio.h>

int main() {
    int salaries[5] = {25000, 40000, 32000, 51000, 45000}, count = 0;
    for (int i = 0; i < 5; i++) {
        if (salaries[i] >= 30000 && salaries[i] <= 50000)
            count++;
    }
    printf("Employees earning between 30K-50K: %d", count);
    return 0;
}

```

---

### 19) Pass Pointer Variables to Function and Sum Them

```

#include <stdio.h>

int sum(int *a, int *b) {
    return *a + *b;
}

int main() {
    int x = 10, y = 20;
    printf("Sum: %d", sum(&x, &y));
    return 0;
}

```

---

### 20) Store Name, Roll No, and Percentage of 10 Students in a File

```

#include <stdio.h>

int main() {
    FILE *file = fopen("students.txt", "w");
    for (int i = 1; i <= 10; i++) {

```

```
        fprintf(file, "Student%d RollNo%d Percentage%d%%\n", i, i, i * 5);
    }
    fclose(file);
    printf("Data stored in students.txt");
    return 0;
}
```

---

## 21) Create a Patient Data File

```
#include <stdio.h>

int main() {
    FILE *file = fopen("patient.txt", "w");
    fprintf(file, "Name: Raj, Disease: Fever, Age: 45, Bed No: 10\n");
    fclose(file);
    return 0;
}
```

---

## 22) Display Only Patients Suffering from COVID

```
#include <stdio.h>
#include <string.h>

int main() {
    FILE *file = fopen("patient.txt", "r");
    char name[50], disease[50], age[10], bed[10];

    while (fscanf(file, "%s %s %s %s", name, disease, age, bed) != EOF) {
        if (strcmp(disease, "COVID") == 0)
            printf("%s %s %s %s\n", name, disease, age, bed);
    }
    fclose(file);
    return 0;
}
```

---

## 23) Store Book Details in "library.txt" and Display Records

```
#include <stdio.h>

int main() {
    FILE *file = fopen("library.txt", "w");
    fprintf(file, "Book1, ID: 101, Author: John\n");
    fclose(file);

    file = fopen("library.txt", "r");
    char c;
    while ((c = fgetc(file)) != EOF)
        putchar(c);
    fclose(file);
    return 0;
}
```

---

## 24) Display All Records from "student.txt"

```
#include <stdio.h>

int main() {
    FILE *file = fopen("student.txt", "r");
    char line[100];
```

```
while (fgets(line, 100, file))
    printf("%s", line);

fclose(file);
return 0;
}
```

---

### 25) Display Only Records Where Name is "Ram"

```
#include <stdio.h>
#include <string.h>

int main() {
    FILE *file = fopen("student.txt", "r");
    char name[50], class[10], marks[50];

    while (fscanf(file, "%s %s %s", name, class, marks) != EOF) {
        if (strcmp(name, "Ram") == 0)
            printf("%s %s %s\n", name, class, marks);
    }
    fclose(file);
    return 0;
}
```

---

### 26) Display Only Students Who Passed All Subjects

```
#include <stdio.h>

int main() {
    FILE *file = fopen("student.txt", "r");
    char name[50];
    int m1, m2, m3;

    while (fscanf(file, "%s %d %d %d", name, &m1, &m2, &m3) != EOF) {
        if (m1 >= 35 && m2 >= 35 && m3 >= 35)
            printf("%s passed all subjects\n", name);
    }
    fclose(file);
    return 0;
}
```

---

### 27) Add Records Until User Presses "Y"

```
#include <stdio.h>

int main() {
    FILE *file = fopen("student.txt", "a");
    char name[50];
    char choice = 'y';

    while (choice == 'y') {
        printf("Enter name: ");
        scanf("%s", name);
        fprintf(file, "%s\n", name);
        printf("Add more? (y/n): ");
        scanf(" %c", &choice);
    }
}
```

```
    fclose(file);
    return 0;
}
```

---

### 28) Display Students Who Failed in At Least One Subject

```
#include <stdio.h>

int main() {
    FILE *file = fopen("student.txt", "r");
    char name[50];
    int m1, m2, m3;

    while (fscanf(file, "%s %d %d %d", name, &m1, &m2, &m3) != EOF) {
        if (m1 < 35 || m2 < 35 || m3 < 35)
            printf("%s failed\n", name);
    }
    fclose(file);
    return 0;
}
```

---

### 29) Display Only Name and Percentage of All Students

```
#include <stdio.h>

int main() {
    FILE *file = fopen("student.txt", "r");
    char name[50];
    int m1, m2, m3;

    while (fscanf(file, "%s %d %d %d", name, &m1, &m2, &m3) != EOF) {
        printf("%s - %.2f%%\n", name, (m1 + m2 + m3) / 3.0);
    }
    fclose(file);
    return 0;
}
```

---

### 30) Read N Student Records and Display Them

```
#include <stdio.h>

int main() {
    FILE *file = fopen("students.txt", "r");
    char name[50];
    int roll, marks;

    while (fscanf(file, "%s %d %d", name, &roll, &marks) != EOF) {
        printf("Name: %s, Roll: %d, Marks: %d\n", name, roll, marks);
    }
    fclose(file);
    return 0;
}
```

---

### 31) Store Words from a Sentence in a File

```
#include <stdio.h>

int main() {
    FILE *file = fopen("words.txt", "w");
```

```
char word[50];

while (scanf("%s", word) != EOF)
    fprintf(file, "%s\n", word);

fclose(file);
return 0;
}
```

---

### 32) Display Student Records if File Exists

```
#include <stdio.h>

int main() {
    FILE *file = fopen("student.txt", "r");

    if (!file)
        printf("File does not exist");
    else
        printf("File exists");

    fclose(file);
    return 0;
}
```

---