# Practical-1

**AIM: Write a C program to print the address of a variable using a pointer**

**CODE:**

```c
#include <stdio.h>
int main() {
int num = 42;
int *ptr = &num;
    printf("The address of num is: %p", ptr);
return 0;
}
```

**OUTPUT:**

The address of num is: 0x7fff71670cbc

# Practical-2

**AIM: Write a C program to create a Calculator using a pointer**

**CODE:**

```c
#include <stdio.h>
 int main() {
 double num1, num2, result;
 char op;
 double *ptr1, *ptr2, *ptr_result;
ptr1 = &num1;
ptr2 = &num2;
 ptr_result = &result;
 printf("Enter first number: ");
 scanf("%lf", ptr1);
 printf("Enter operator (+, -, *, /): ");
 scanf(" %c", &op);
 printf("Enter second number: ");
 scanf("%lf", ptr2);
switch(op) {
   case '+':
    *ptr_result = *ptr1 + *ptr2;
    break;
   case '-':
    *ptr_result = *ptr1 - *ptr2;
```

```
      break;
    case '*':
     *ptr_result = *ptr1 * *ptr2;
     break;
    case '/':
     *ptr_result = *ptr1 / *ptr2;
     break;
    default:
     printf("Invalid operator\n");
return 1;
  }
 printf("%.2lf %c %.2lf = %.2lf\n", *ptr1, op, *ptr2, *ptr_result);
  return 0;}
```

**OUTPUT:**

Enter first number: 5

Enter operator (+, -, *, /): *

Enter second number: 5

5.00 * 5.00 = 25.00

# Practical-3

**AIM: Write a C program to swap the two values using call by value and call by reference**

**CODE:**

```c
#include <stdio.h>

// function prototype for call by value

void swapByValue(int x, int y);

// function prototype for call by reference

void swapByReference(int *x, int *y);

int main() {
    int a = 10, b = 20;
 // call swapByValue
    printf("Before swapByValue: a = %d, b = %d\n", a, b);
    swapByValue(a, b);
    printf("After swapByValue: a = %d, b = %d\n", a, b);
 // call swapByReference
    printf("Before swapByReference: a = %d, b = %d\n", a, b);
    swapByReference(&a, &b);
    printf("After swapByReference: a = %d, b = %d\n", a, b);

    return 0;
}

// function definition for call by value
```

```c
void swapByValue(int x, int y) {

    int temp = x;

    x = y;

    y = temp;

}

// function definition for call by reference

void swapByReference(int *x, int *y) {

    int temp = *x;

    *x = *y;

    *y = temp;

}
```

**OUTPUT:**

Before swapByValue: a = 10, b = 20

After swapByValue: a = 10, b = 20

Before swapByReference: a = 10, b = 20

After swapByReference: a = 20, b = 10

# Practical-4

**AIM: Define a structure type struct personal that would contain person name, Date of birth and age using this structure to read this information of 4 people and display the same.**

**CODE:**

```c
#include <stdio.h>

#include <stdlib.h>
struct personal {
  char name[50];
  int birth_year;
  int birth_month;
  int birth_day;
  int age;
};
int main() {
  struct personal people[4];
 // Read information for each person
  for (int i = 0; i < 4; i++) {
    printf("Enter name for person %d: ", i+1);
    fgets(people[i].name, 50, stdin);
    printf("Enter birth year for person %d: ", i+1);
    scanf("%d", &people[i].birth_year);
    printf("Enter birth month for person %d: ", i+1);
```

```c
scanf("%d", &people[i].birth_month);

printf("Enter birth day for person %d: ", i+1);

scanf("%d", &people[i].birth_day);

// Calculate age

int current_year = 2023;

int current_month = 4;

int current_day = 7;

int age = current_year - people[i].birth_year;

if (current_month < people[i].birth_month ||

    (current_month == people[i].birth_month && current_day <
people[i].birth_day)) { age--;}

people[i].age = age;

// Clear the input buffer

while ((getchar()) != '\n');

}

// Display information for each person

for (int i = 0; i < 4; i++) {

printf("Person %d:\n", i+1);

printf("Name: %s", people[i].name);

printf("Date of birth: %d-%02d-%02d\n", people[i].birth_year,
people[i].birth_month, people[i].birth_day);

printf("Age: %d\n", people[i].age);

}

return 0;

}
```

**OUTPUT:**

Enter name for person 1: Aman

Enter birth year for person 1: 2005

Enter birth month for person 1: 06

Enter birth day for person 1: 03


Enter name for person 2: Nikhil

Enter birth year for person 2: 2004

Enter birth month for person 2: 12

Enter birth day for person 2: 01


Enter name for person 3: pradip

Enter birth year for person 3: 2005

Enter birth month for person 3: 06

Enter birth day for person 3: 25


Enter name for person 4: RIYAN

Enter birth year for person 4: 2006

Enter birth month for person 4: 09

Enter birth day for person 4: 09


Person 1:

Name: Aman

Date of birth: 2004-06-03

Age: 19


Person 2:

Name: Nikhil

Date of birth: 2004-12-01

Age: 18

Person 3:

Name: pradip

Date of birth: 2005-06-25

Age: 17

Person 4:

Name: RIYAN

Date of birth: 2006-09-09

Age: 16

# Practical-5

**AIM: Write a C program to calculate the sum of n numbers entered by the user using dynamic memory allocation.**

**CODE:**

```c
#include <stdio.h>
#include <stdlib.h>
int main() {
    int n, *ptr, i, sum = 0;
    printf("Enter the number of elements: ");
    scanf("%d", &n);
// Allocate memory dynamically
    ptr = (int*) malloc(n * sizeof(int));
  if (ptr == NULL) {
        printf("Error! Memory not allocated.");
        exit(0);
    }
  printf("Enter elements: ");
    for (i = 0; i < n; i++) {
        scanf("%d", ptr + i);
        sum += *(ptr + i);
    }
 printf("Sum = %d", sum);
// Free the dynamically allocated memory
    free(ptr);
```

```
  return 0;
}
```

**OUTPUT:**

Enter the number of elements: 5

Enter elements: 1

2

3

4

5

Sum = 15

# Practical-6

**AIM: A file named "New" contains a series of integer numbers. Write a c program to read all numbers from a file and then copy all odd numbers into a file named "odd" and write all even numbers into a file named "even". Then display the values of files odd and even on the screen.**

**CODE:**

```c
#include <stdio.h>
int main() {
    FILE *fp, *even_fp, *odd_fp;
    int num;
    fp = fopen("New.txt", "r");
    if (fp == NULL) {
        printf("Error opening file.");
        return 1;
    }
    even_fp = fopen("even.txt", "w");
    if (even_fp == NULL) {
        printf("Error opening file.");
        return 1;
    }
    odd_fp = fopen("odd.txt", "w");
    if (odd_fp == NULL) {
        printf("Error opening file.");
        return 1;
    }
```

```c
    while (fscanf(fp, "%d", &num) != EOF) {

        if (num % 2 == 0) {

            fprintf(even_fp, "%d\n", num);

        } else {

            fprintf(odd_fp, "%d\n", num);

        }

    }

    fclose(fp);

    fclose(even_fp);

    fclose(odd_fp);

        printf("Values in even file:\n");

    even_fp = fopen("even.txt", "r");

    while (fscanf(even_fp, "%d", &num) != EOF) {

        printf("%d\n", num);

    }

    fclose(even_fp);

  printf("Values in odd file:\n");

  odd_fp = fopen("odd.txt", "r");

  while (fscanf(odd_fp, "%d", &num) != EOF) {

        printf("%d\n", num);

    }

    fclose(odd_fp);

 return 0;

}
```

**OUTPUT:**

Error opening file.

# PRACTICAL-7

**AIM: Write a C++ program to Check if the number is prime or not using a function**

**CODE:**

```cpp
#include <iostream>

#include <cmath>

using namespace std;

bool isPrime(int n) {

   if (n <= 1) {

      return false;

   }

   for (int i = 2; i <= sqrt(n); i++) {

      if (n % i == 0) {

         return false;

      }

   }

   return true;

}

int main() {

   int n;

   cout << "Enter a positive integer: ";

   cin >> n;

   if (isPrime(n)) {

      cout << n << " is a prime number" << endl;
```

```
    } else {

        cout << n << " is not a prime number" << endl;

    }

    return 0;

}
```

**OUTPUT:**

Enter a positive integer: 5

5 is a prime number

# PRACTICAL-8

**AIM: Write a C++ program that prompts the user to enter a letter and check whether a letter is a vowel or constant.**

**CODE:**

```cpp
#include <iostream>
 using namespace std;
 int main() {
   char c;
   cout << "Enter a Character:";
   cin >> c;
   /* Check if input alphabet is member of set{A,E,I,O,U,a,e,i,o,u} */
   if(c == 'a' || c == 'e' || c =='i' || c=='o' || c=='u' || c=='A'
      || c=='E' || c=='I' || c=='O' || c=='U'){
     cout << c << " is VOWEL";
   } else {
     cout << c <<" is CONSONANT";
   }
    return 0;
}
```

**OUTPUT:**

Enter a Character : A

A is VOWEL

# PRACTICAL-9

**AIM: Write a C++ program to demonstrate the concept of constructor and destructor**

**CODE:**

```cpp
#include <iostream>
class MyClass {
public:
  // Constructor
  MyClass() {
    std::cout << "Constructor called" << std::endl;
  }
  // Destructor
  ~MyClass() {
    std::cout << "Destructor called" << std::endl;
  }
};
int main() {
  // Create an object of MyClass
  MyClass obj;
// The object is automatically destroyed when it goes out of scope
  return 0;
}
```

**OUTPUT:**

Constructor called

Destructor called

# PRACTICAL-10

**AIM: Write a C++ program to implement Multilevel Inheritance.**

**CODE:**

```cpp
#include <iostream>

using namespace std;

class A {

public:

    int a;

    void get_A_data()

    {

        cout << "Enter value of a: ";

        cin >> a;

    }

};

class B : public A {

public:

    int b;

    void get_B_data()

    {

        cout << "Enter value of b: ";

        cin >> b;

    }
```

```cpp
};
class C : public B {
private:
    int c;

public:
    void get_C_data()
    {
        cout << "Enter value of c: ";
        cin >> c;
    }
    void sum()
    {
        int ans = a + b + c;
        cout << "sum: " << ans;
    }
};
int main()
{
    C obj;
    obj.get_A_data();
    obj.get_B_data();
    obj.get_C_data();
    obj.sum();
    return 0;
}
```

**OUTPUT:**

Enter value of a: 12

Enter value of b: 2

Enter value of c: 2

sum:16

# PRACTICAL-11

**AIM: Write a C++ program to overload binary + operator.**

**CODE:**

```cpp
#include <iostream>
using namespace std;
class Complex {
  private:
   float real;
   float imag;
 public:
   // Constructor to initialize real and imag to 0
   Complex() : real(0), imag(0) {}
     void input() {
     cout << "Enter real and imaginary parts respectively: ";
     cin >> real;
     cin >> imag;
   }
   // Overload the + operator
   Complex operator + (const Complex& obj) {
     Complex temp;
     temp.real = real + obj.real;
     temp.imag = imag + obj.imag;
     return temp;
```

```cpp
    }
  void output() {
    if (imag < 0)
      cout << "Output Complex number: " << real << imag << "i";
    else
      cout << "Output Complex number: " << real << "+" << imag << "i";
  }
};
int main() {
  Complex complex1, complex2, result;
  cout << "Enter first complex number:\n";
  complex1.input();
  cout << "Enter second complex number:\n";
  complex2.input();
    // complex1 calls the operator function
    // complex2 is passed as an argument to the function
        result = complex1 + complex2;
        result.output();
  return 0;
}
```

**OUTPUT:**

Enter first complex number:

Enter real and imaginary parts respectively: 3

5

Enter second complex number:

Enter real and imaginary parts respectively: 5

Output Complex number: 8+12i

# PRACTICAL-12

**AIM: Write a C++ program to understand the concept of run time polymorphism.**

**CODE:**

```cpp
#include <iostream>
using namespace std;
class base {
public:
    virtual void print()
    {
            cout << "print base class" <<
                        endl;
    }
    void show()
    {
    cout << "show base class" <<
                endl;
    }
};
class derived : public base {
public:
    void print()
    {
```

```cpp
        cout << "print derived class" <<

                endl;

    }
    void show()

    {

    cout << "show derived class" <<

                endl;

    }
};
int main()

{

    base* bptr;

    derived d;

    bptr = &d;

    bptr->print();

    bptr->show();

    return 0;

}
```

**OUTPUT:**

print derived class

show base class

# PRACTICAL-13

**AIM: Create an HTML Page containing the following Gray Layout using CSS.**

**CODE:**

```html
<!DOCTYPE html>
 <html lang="en">
  <head>
   <meta charset="UTF-8" />
   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
   <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>   <title>Document</title>
   <style type="text/css">
    * {
     font-size: 22px;
     font-weight: bold;
    }
    #r1 {
     background-color: gray;
     height: 50px;     padding-top: 20px;     width: 98%;     padding-left:
2%;
     margin-bottom: 10px;
    }
    #r2,
    #r5 {
     background-color: gray;
```

```css
    height: 33px;        padding-top: 7px;        /* width: 100%; */        padding-left: 2%;        text-align: center;

    margin-bottom: 10px;

  }

  #r3 {

    background-color: gray;

    height: 90px;        width: 100%;        padding-top: 60px;        text-align: center;

    margin-bottom: 10px;

  }    #r4 {

    height: 600px;        width: 100%;


    margin-bottom: 10px;

  }

  #r4c1 {

     width: 28%;        margin-right: 2%;


  }

  #r4c2 {        width: 70%;

  }

  #r4 div {

    float: left;        height: 320px;        padding-top: 280px;        text-align: center;        background-color: gray;

  }

 </style>

 </head>

 <body>
```

```html
    <div>

      <div id="r1">Logo</div>

      <div id="r2">Navigation</div>

      <div id="r3">Header/Banner</div>

      <div id="r4">

        <div id="r4c1">Side bar</div>

        <div id="r4c2">Body Area</div>

      </div>

      <div id="r5">Footer</div>

    </div>

  </body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>

  <style type="text/css">

    *{      font-size: 22px;

      font-weight: bold;

    }

    #r1{

      background-color: gray;

      height: 50px;
```

```css
        height: 50px;

    padding-top: 20px;          width: 98%;          padding-left: 2%;

    margin-bottom: 10px;

}

#r2,#r5{

    background-color: gray;

    height: 33px; padding-top: 7px;  /* width: 100%; */  padding-left: 2%;
text-align: center;

    margin-bottom: 10px;

}

#r3{

    background-color: gray;

    height: 90px; width: 100%;  padding-top: 60px; text-align: center;

    margin-bottom: 10px;

}

#r4{  height: 600px; width: 100%; margin-bottom: 10px;

}

#r4c1{ width: 32%; margin-right: 2%;

}

#r4c2{ width: 32%;  margin-right: 2%;

}

#r4c3{

    width: 32%;

}

    #r4 div{ float: left;  height: 320px; padding-top: 280px;  text-align: center;
background-color: gray;
```

```
        }
    </style>
</head>
<body>
  <div>
    <div id="r1">
        Logo
    </div>
    <div id="r2">
        Navigation
    </div>
    <div id="r3">
      Header/Banner
    </div>
    <div id="r4">
      <div id="r4c1">
        box-1
      </div>
      <div id="r4c2">
        box-2
      </div>
      <div id="r4c3">
      box-3        </div>
    </div>
    <div id="r5">
      Footer
```

```html
        </div>

    </div>

</body>

</html>

<html> <head>

    <title>Demo Layout 3</title>

    <style type="text/css">

        * {        font-size: 22px;

            font-weight: bold;

        }    #R1 {        background-color: gray;        height: 50px;
padding-top: 20px;        width: 98%;        padding-left: 2%;

            margin-bottom: 10px;

        }



 #R5 {

        background-color: gray;

        height: 33px; width: 100%;  padding-top: 7px; text-align: center;
margin-bottom: 10px;

        margin-top: 1%;

    }

table {

        width: 100%;

    }

    .s {

        height: 600px;

        width: 28%;
```

```
        }
        .b {
            background-color: gray;
            height: 100px; text-align: center;
        }
    .bl {
            height: 10px;
        }
     .ba {
            background-color: gray;  height: 490px; text-align: center;
        }
    </style>
</head>
<body>
  <div>
    <div id="R1">
        Logo
    </div>
    <table>
      <tr>
        <td rowspan="3" class="s">
            Side Bar Navigation
        </td>
        <td class="b">
            Header/Banner
        </td>
```

```html
        </tr>
        <tr>
          <td>
            <div class="bl"></div>
          </td>
        </tr>
        <tr>
          <td class="ba">
            Body Area
          </td>
        </tr>
      </table>
      <div id="R5">
        Footer
      </div>
    </div>
  </body>
</html>
<html>
<head>
    <title>Demo Layout 4</title>
    <style type="text/css">
        * {
                font-size: 22px;

                font-weight: bold;
        }
```

```css
#R1 {

        background-color: gray;  height: 50px; padding-top: 20px;

        width: 98%;

        padding-left: 2%;

        margin-bottom: 10px;

}

#R2 {  width: 100%;

        margin-bottom: 10px;

        height: 620px;

        }

#R2 div {  float: left;

                margin-bottom: 1%; margin-bottom: 1%;

}

#R5 {

        background-color: gray; height: 33px;
width: 100%; padding-top: 7px; text-align: center;

        margin-bottom: 10px;

}

 #R3 {

        background-color: gray; height: 90px;
width: 100%;                 padding-top: 60px;

        text-align: center;

}

    .B1,.B2,.B3,.B5,.B6,.B7,.B9,.B10,.B11,.B13,.B14,.B15 {

        background-color: grey; height: 150px;
width: calc(25% - 1%);

        margin-right: 1%;
```

```css
}


.B4,.B8,.B12,.B16 { background-color: grey;
height: 150px;

        width: 25%;


}
 #R3 {

        width: 100%;

        background-color: white;

        height: 222px;

        padding-top: 0px;

        margin-top: 2%;

}


#R3 div { float: left;

        margin-bottom: 0%;

}
#c1 {
#c1 {

        background-color: grey; height: 220px; width: 32%;

        margin-right: 1%; }

#c2 {

        background-color: grey;  height: 220px;
width: 34%;                        margin-right: 1%;

}
#c3 { background-color: grey; height: 220px;
```

```
                width: 32%; }

        .r4 {

        height: 50px; background-color: grey;  margin-top: 1%;
        text-align: center;

                padding-top: 20px; }

    </style>

</head>

 <body>

    <div>

            <div id="R1">

                    Logo

            </div>

            <div id="R2">

                    <div class="B1"></div>

                    <div class="B2"></div>

                    <div class="B3"></div>

                    <div class="B4"></div>

                    <div class="B5"></div>

                    <div class="B6"></div>

                    <div class="B7"></div>

                    <div class="B8"></div>

                    <div class="B9"></div>

        <div class="B10"></div>

        <div class="B11"></div>

        <div class="B12"></div>

        <div class="B13"></div>
```

```
        <div class="B14"></div>

        <div class="B15"></div>

        <div class="B16"></div>

</div>


<div id="R3">

        <div id="c1">Box-1</div>

        <div id="c2">Box-2</div>

        <div id="c3">Box-3</div>

</div>

<div class="r4">

        Footer

</div>

</body>

</html>
```
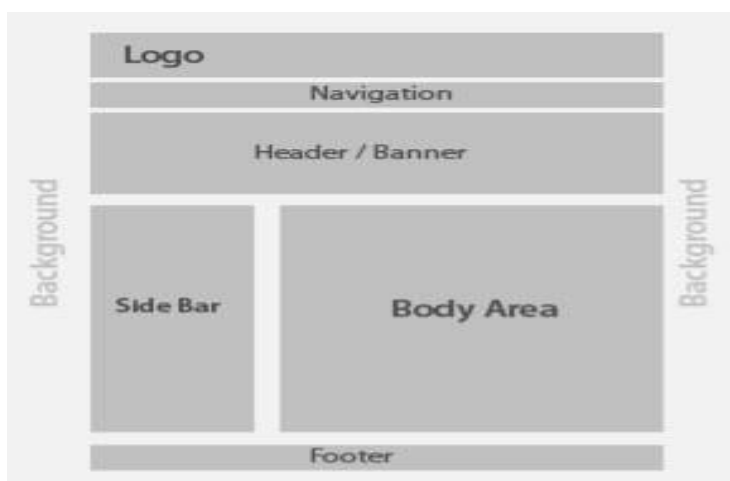
**OUTPUT:**

Logo

Header/Banner

Side Bar Navigation

Body Area

Footer

| Logo | | | |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| Box-1 | Box-2 | | Box-3 |
| | | | |
| Footer | | | |

# PRACTICAL-14

**AIM: Demonstrate JavaScript Form Validation with proper examples.**

**CODE:**

```html
<html>
  <head>
    <title>Form Validation</title>
    <script type = "text/javascript">
      <!--
        // Form validation code will come here.
      //-->
    </script>
  </head>
  <body>
    <form action = "/cgi-bin/test.cgi" name = "myForm" onsubmit = "return(validate());">
      <table cellspacing = "2" cellpadding = "2" border = "1">
        <tr>
          <td align = "right">Name</td>
          <td><input type = "text" name = "Name" /></td>
        </tr>
         <tr>
          <td align = "right">EMail</td>
          <td><input type = "text" name = "EMail" /></td>
```

```html
        </tr>
         <tr>
           <td align = "right">Zip Code</td>
           <td><input type = "text" name = "Zip" /></td>
         </tr>
         <tr>
           <td align = "right">Country</td>
           <td>
             <select name = "Country">
               <option value = "-1" selected>[choose yours]</option>
               <option value = "1">USA</option>
               <option value = "2">UK</option>
               <option value = "3">INDIA</option>
             </select>
           </td>
         </tr>
         <tr>
           <td align = "right"></td>
           <td><input type = "submit" value = "Submit" /></td>
         </tr>
         </table>
     </form>
   </body>
</html>
```

**OUTPUT:**

| Name | |
|---|---|
| EMail | |
| Zip Code | |
| Country | [choose yours] ⌄ |
| | Submit |

**AIM: Write a javascript to check if the number is even or odd.**

**CODE:**

```javascript
const number = prompt("Enter a number: ");
if(number % 2 == 0) {
    console.log("The number is even.");
}
else {
    console.log("The number is odd.");
}
```

**OUTPUT:**

Enter a number: 15

The number is odd.

**AIM: Create a page and access the LocationAPI.**

**CODE:**

```html
<!DOCTYPE html>

<html>

<body>

  <p>Click the button to get your coordinates.</p>

    <button onclick="getLocation()">Try It</button>

  <p id="demo"></p>

  <script>

var x = document.getElementById("demo");

function getLocation() {

  if (navigator.geolocation) {

    navigator.geolocation.getCurrentPosition(showPosition);

  } else {

    x.innerHTML = "Geolocation is not supported by this browser.";

  }

}

function showPosition(position) {

  x.innerHTML = "Latitude: " + position.coords.latitude +

  "<br>Longitude: " + position.coords.longitude;

}

</script>

</body>

</html>
```

**OUTPUT:**

Click the button to get your coordinates.

Try It

Latitude: 23.0926305
Longitude: 72.5323024

**AIM: Make a resume using the HTML tags without CSS**

**CODE: <!DOCTYPE html>**

**<html lang="en">**

**<head>**

  **<meta charset="UTF-8">**

  **<meta http-equiv="X-UA-Compatible" content="IE=edge">**

  **<meta name="viewport" content="width=device-width, initial-scale=1.0">**

  **<title>DAY-1</title>**

**</head>**

**<body>**

  **<h2>DENISA RABADIYA</h2>**

  **<h3><u>CONTACT</u></h3>**

  **<p class="p"><b>MOBILE : </b>9663456271</p>**

  **<p class="p"><b>EMAIL : </b>denisarabadiya@gmail.com</p>**

  **<h3><u>OBJECTIVE</u></h3>**

  **<ul>**

  **<li><p>Strong in design and integration with intuitive problem-solving skills.       Proficient in CSS, C, C++, and HTML. Passionate about implementing and launching new projects. Ability to translate business requirements into technical solutions. Looking to start the career as an entry-level software engineer with a reputed firm driven by technology.</p></li>   </ul>**

  **<h3><u>EDUCATION</u></h3>**

  **<table border="collapse">**

  **<tr>**

```html
    <th>COURSE</th>

    <th>BOARD</th>

    <th>YEAR</th>
    <th>MARKS</th>
  </tr>
  <tr>

    <td>SSC</td>

    <td>GSEB</td>

    <td>2020</td>

    <td>88%</td>

  </tr>
  <tr>

    <td>HSC</td>

    <td>GSEB</td>

    <td>2022</td>

    <td>71%</td>
  </tr>
  </table>
  <h3><u>SKILLS</u></h3>
  <h3>Technical Skills</h3>
  <ul>
```

```
<li>C,C++</li>

<li>HTML,CSS,JAVASCRIPT</li>
<li>SQL</li>
<li>VS STUDIO,WINDOWS</li>
</ul>
    <h3>Soft Skills</h3>
<ul>
<li>LEADERSHIP</li>
<li>ADAPTABILITY</li>

<li>TEAM WORK</li>
<li>PROBLEM SOLVING</li>
</ul>
<h3><u>PROJECT</u></h3>
<h3>BANK MANAGEMENT SYSTEM</h3>
<ul>
<li>USING SQL</li>
</ul>
<h3>STUDENT MANAGEMENT SYSTEM</h3>

<ul>

<li>USING C</li>

</ul>
```

```html
<h3><u>LANGUAGES</u></h3>
<ul>
<li>GUJARATI</li>
<li>HINDI</li>
<li>ENGLISH</li>
</ul>
<h3><u>HOBBIES</u></h3>

<ul>

<li>TRAVEL</li>

<li>SPORTS</li>
</ul>
<h3><u>DECLARATION</u></h3>
<p>I hereby declare that the details and information given above are complete and true to the best of my knowledge</p>

</body>
</html>
```

**OUTPUT:**

# AMAN NASIT

## CONTACT

**MOBILE : 9875364228**

**EMAIL : amannasit123@gmail.com**

## OBJECTIVE

- Strong in design and integration with intuitive problem-solving skills. Proficient in CSS, C, C++, and HTML. Passionate about implementing and launching new projects. Ability to translate business requirements into technical solutions. Looking to start the career as an entry-level software engineer with a reputed firm driven by technology.

## EDUCATION

| COURSE | BOARD | YEAR | MARKS |
|--------|-------|------|-------|
| SSC | GSEB | 2020 | 88% |
| HSC | GSEB | 2022 | 71% |

## SKILLS

### Technical Skills

- C,C++
- HTML,CSS
- SQL
- VS STUDIO,WINDOWS

### Soft Skills

- LEADERSHIP
- ADAPTABILITY
- TEAM WORK
- PROBLEM SOLVING

## PROJECT

### BANK MANAGEMENT SYSTEM

- USING SQL

### STUDENT MANAGEMENT SYSTEM

- USING C

## LANGUAGES

- ENGLISH
- HINDI
- GUJARATI

## HOBBIES

- TRAVELLING
- SPORTS

## DECLARATION

I hereby declare that the details and information given above are complete and true to the best of my knowledge

# PRACTICAL-18

**AIM: Create an HTML page table and form.**

**CODE: <!DOCTYPE html>**

```
<html>
 <body>
  <h2>Form</h2>
  <table>
    <form>
      <tr>
        <td>


          <label for="name">                    Name


          </label>
        </td>
        <td><input type="text" id="name" />
        </td>
      </tr>
      <tr>
        <td><label for="email">                Email
          </label>
        </td>


        <td><input type="email" id="email" />
```

```html
            </td>
          </tr>
        <tr>
          <td>


            <label for="city">                 City


              </label>
          </td>
          <td><input type="text" id="city" />
          </td>
        </tr>




        <tr>
          <td><label for="telnum">            Tel No.
              </label>


          </td>
          <td><input type="telnum" id="telnum" />
          </td>
        </tr>
        <tr>
          <td><label for="Roll No.">           Roll No.
```

```
        </label>

      </td>

    <td><input type="number" id="rollno" />

      </td>
    </tr>
    <tr>
     <td>

       <input type="submit" value="Submit">
     </td>

           <td>

       <input type="reset" value="Reset">

      </td>
    </tr>
  </form>
 </table>
</body>
 </html>
```

**OUTPUT:**

**Form**

| Name | |
|---|---|
| Email | |
| City | |
| Tel No. | |
| Roll No. | |

[ Submit ] [ Reset ]

**AIM: Create Registration form and do proper validation with HTML 5 inbuilt functionality.**

**CODE:**

```html
<!DOCTYPE html>

<html lang="en">



 <head>

   <meta charset="UTF-8">

   <meta http-equiv="X-UA-Compatible" content="IE=edge">

   <meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Practical 5</title>  </head>

<body bgcolor="powderblue">


    <form action="" method="post">


    <h1 align="center">Registration Form</h1>


    <hr>
    <table border="2" width="50%" height="50%" align="center" >
      <tr>
        <th><b>First Name :</b></th>

        <td><input type="text"  required placeholder="Enter First Name"
    title="Fill the correct data" maxlength="20" ></td>
```

```
        </tr>

        <tr>

            <th> <b>Last Name : </b></th>

            <td><input type="text" required placeholder="Enter Last Name"
maxlength="20"></td>

        </tr>


    <tr>

            <th> <b>Contact no. :</b></th>

            <td><input type="text" placeholder="Enter Your Contact No."
pattern="\d{10}" minlength="10" maxlength="10" required </td>

        </tr>

        <tr>

            <th> <b>Landline no : (079-123456)</b></th>

            <td><input type="text" pattern="[^0-9]{3}-\d{6}" maxlength="10"
required

placeholder="Enter Your Landline No.">


        </tr>

        <tr>

            <th><b>Email :</b></th>

            <td><input type="email" required placeholder="Enter Your Email
ID" pattern="[^@\t\r\n]+@[^@\t\r\n]+\.[^@ \t\r\n]+"></td>


        </tr>

        <tr>
```

```
<th>Password</th>

    <td><input type="password" required placeholder="Enter a
Password"  pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8}"></td>

    </tr>

    <tr>


      <th><b>Date of Birth</b></th>


      <td><input type="date" required></td>


    </tr>

    <tr>

      <th>Age</th>

      <td><input type="Number" min="18" max="40" width="600%"
placeholder="Enter

Age Between 18 to 40" required></td>


    </tr>

    <tr>

      <th> <b>Gender</b></th>

    </tr>

 </table>

 </body>

 </html>
```

**OUTPUT:**

# Registration Form

| | |
|---|---|
| **First Name :** | Enter First Name |
| **Last Name :** | Enter Last Name |
| **Contact no. :** | Enter Your Contact No. |
| **Landline no : (079-123456)** | Enter Your Landline No. |
| **Email :** | Enter Your Email ID |
| **Password** | Enter a Password |
| **Date of Birth** | dd-mm-yyyy |
| **Age** | Enter / |
| **City:** | AHMEDABAD |
| | Submit  Reset |

**AIM: Create a simple XMLHTTPRequest,and retrieve the data from the text file.**

**CODE:**

```html
<!DOCTYPE html>
 <html>

<body>

  <div id="demo">
    <h2>Let AJAX change this text</h2>
  </div>


  <button type="button" onclick="loadDoc()">Change Content</button>

  <script>
    function loadDoc() {var xhttp = new
XMLHttpRequest();xhttp.onreadystatechange = function () {
        if (xhttp.readyState == 4 && xhttp.status == 200) {
          document.getElementById("demo").innerHTML =
xhttp.responseText;
        }
    };
    xhttp.open("GET", "ajax_info.txt", true);xhttp.send();
  }
  </script>
```

**</body>**

**</html>**

**OUTPUT:**

# The XMLHttpRequest Object

Change Content

# AJAX

AJAX is not a programming language.

AJAX is a technique for accessing web servers from a web page.

AJAX stands for Asynchronous JavaScript And XML.
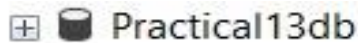
**Aim:** To study DDL-create and DML-insert commands. Create following Tables.

**Code:**

```
CREATE DATABASE Practical13db;
```

**Output:**

Practical13db
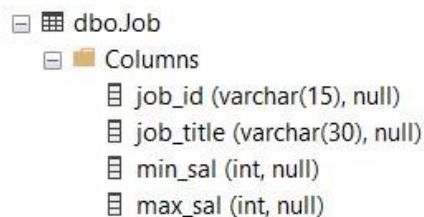
**13.1** Job (job_id, job_title, min_sal, max_sal) **Code:**

```
CREATE TABLE Job (   job_id VARCHAR(15),
job_title VARCHAR(30),
 min_sal INT,
max_sal INT
);
```

**Output:**

dbo.Job
  Columns
    job_id (varchar(15), null)
    job_title (varchar(30), null)
    min_sal (int, null)
    max_sal (int, null)

**13.2** Employee (emp_no, emp_name, emp_sal, emp_comm, dept_no) **Code:**

```
CREATE TABLE Employee (  emp_no INT,
 emp_name VARCHAR(30),
emp_sal DECIMAL(8,2),
 emp_comm DECIMAL(6,1),   dept_no INT
);
```

## Output:

```
□ ⊞ dbo.Employee
  □ ■ Columns
      目 emp_no (int, null)
      目 emp_name (varchar(30), null)
      目 emp_sal (decimal(8,2), null)
      目 emp_comm (decimal(6,1), null)
      目 dept_no (int, null)
```

## 13.3

Deposit(a_no,cname,bname,amount,a_date)

## Code:

```
CREATE TABLE Deposit(   a_no INT IDENTITY (1,1),   cname VARCHAR(50),   bname
VARCHAR(30),

amount DECIMAL(4,2),

a_date Date

);
```

## Output:

```
□ ⊞ dbo.Deposit
  □ ■ Columns
      目 a_no (int, not null)
      目 cname (varchar(50), null)
      目 bname (varchar(30), null)
      目 amount (decimal(4,2), null)
      目 a_date (date, null)
```

## 13.4 Borrow(loanno,cname,bname,amount)

## Code:

```
CREATE TABLE Borrow(

loanno INT,

  cname VARCHAR(25),

bname VARCHAR(20),

amount DECIMAL(6,2)
```

);

## Output:

- 📇 dbo.Borrow
  - 📁 Columns
    - loanno (int, null)
    - cname (varchar(25), null)
    - bname (varchar(20), null)
    - amount (decimal(6,2), null)

**Aim:** Create tables and insert sample data in tables. Write SQL queries to insert following data into tables.

**14.1** Insert following values in the table **Employee.**

| emp_n | emp_name | emp_sal | emp_comm | dept _no |
|-------|----------|---------|----------|----------|
| 101 | Smith | 800 | | 20 |
| 102 | Snehal | 1600 | 300 | 25 |
| 103 | Adama | 1100 | 0 | 20 |
| 104 | Aman | 3000 | | 15 |
| 105 | Anita | 5000 | 50000 | 10 |
| 106 | Sneha | 2450 | 24500 | 10 |
| 107 | Anamika | 2975 | | 30 |

**Code:**

insert into Employee values(101,'Smith',800,Null,20);  insert into Employee

values(102,'Snehal',1600,300,25);  insert into Employee values(103,'Adama',1100,0,20);  insert

into Employee values(104,'Aman',3000,Null,15);  insert into Employee

values(105,'Anita',5000,50000,10); insert into Employee values(106,'Sneha',2450,24500,10);


insert into Employee values(107,'Anamika',2975,Null,30);


select * from Employee;


**Output:**

| | emp_no | emp_name | emp_sal | emp_comm | dept_no |
|---|--------|----------|---------|----------|---------|
| 1 | 101 | Smith | 800.00 | NULL | 20 |
| 2 | 102 | Snehal | 1600.00 | 300.0 | 25 |
| 3 | 103 | Adama | 1100.00 | 0.0 | 20 |
| 4 | 104 | Aman | 3000.00 | NULL | 15 |
| 5 | 105 | Anita | 5000.00 | 50000.0 | 10 |
| 6 | 106 | Sneha | 2450.00 | 24500.0 | 10 |
| 7 | 107 | Anamika | 2975.00 | NULL | 30 |

**14.2** Insert following values in the table **Job.**

| job_id | job_name | min_sal | max_sal |
|--------|----------|---------|---------|
| IT_PROG | Programmer | 4000 | 10000 |
| MK_MGR | Marketing manager | 9000 | 15000 |
| FI_MGR | Finance manager | 8200 | 12000 |
| FI_ACC | Account | 4200 | 9000 |
| LEC | Lecturer | 6000 | 17000 |
| COMP_OP | Computer Operator | 1500 | 3000 |

**Code:**

```
insert into Job values('IT_PROG','Programmer',4000,10000);

insert into Job values('MK_MGR','Marketing manager',9000,15000);
        insert into Job values('FI_MGR','Finance
manager',8200,12000);      insert into Job
values('FI_ACC','Account',4200,9000);      insert into Job
values('LEC','Lecturer',6000,17000);

insert into Job values('COMP_OP','Computer Operator',1500,3000);

select * from Job;
```

**Output:**

| | job_id | job_title | min_sal | max_sal |
|---|--------|-----------|---------|---------|
| 1 | IT_PROG | Programmer | 4000 | 10000 |
| 2 | MK_MGR | Marketing manager | 9000 | 15000 |
| 3 | FI_MGR | Finance manager | 8200 | 12000 |
| 4 | FI_ACC | Account | 4200 | 9000 |
| 5 | LEC | Lecturer | 6000 | 17000 |
| 6 | COMP_OP | Computer Operator | 1500 | 3000 |

## 14.3 Insert following values in the table **deposit**.

| A_no | cname | Bname | Amount | date |
|------|-------|-------|--------|------|
| 101 | Anil | andheri | 7000 | 01-jan-06 |
| 102 | sunil | virar | 5000 | 15-jul-06 |
| 103 | jay | villeparle | 6500 | 12-mar-06 |

| 104 | vijay | andheri | 8000 | 17-sep-06 |
| 105 | keyur | dadar | 7500 | 19-nov-06 |
| 106 | mayur | borivali | 5500 | 21-dec-06 |

## Code:

insert into Deposit values(101,'Anil','andheri',7000,'01-jan-06');
insert into Deposit        values(102,'sunil','virar',5000,'15-jul-06');
insert into Deposit values(103,'jay','villeparle',6500,'12-mar-06');
 insert into Deposit values(104,'vijay','andheri',8000,'17-sep-06');
  insert into Deposit values(105,'keyur','dadar',7500,'19-nov-06');
 insert into Deposit values(106,'mayur','borivali',5500,'21-dec-06');
select * from Deposit;

## Output:

| | a_no | cname | bname | amount | a_date |
|---|---|---|---|---|---|
| 1 | 101 | Anil | andheri | 7000.00 | 2006-01-01 |
| 2 | 102 | sunil | virar | 5000.00 | 2006-07-15 |
| 3 | 103 | jay | villeparle | 6500.00 | 2006-03-12 |
| 4 | 104 | vijay | andheri | 8000.00 | 2006-09-17 |
| 5 | 105 | keyur | dadar | 7500.00 | 2006-11-19 |
| 6 | 106 | mayur | borivali | 5500.00 | 2006-12-21 |

**Aim:** Write the SQL queries to provide constraints on given tables.
Create A Database Sales and Write SQL Queries to create following tables with all
constrains mentioned in image.
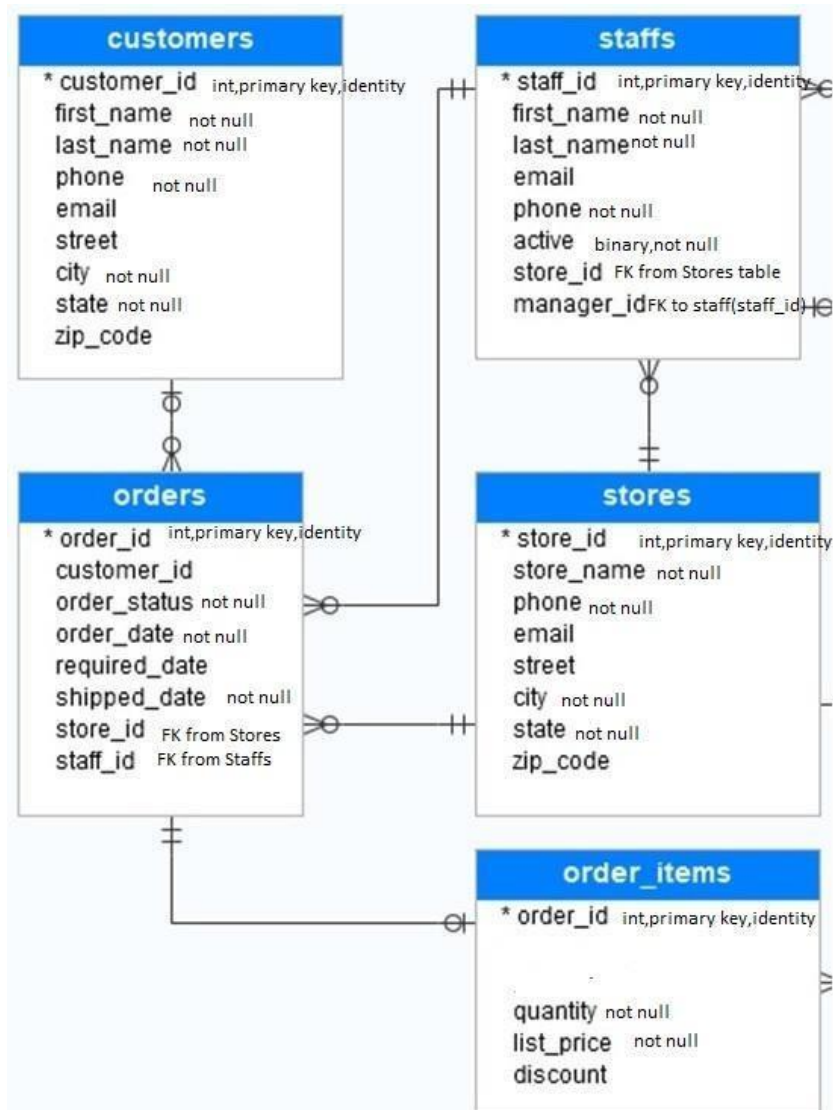


**Code:**

```sql
   create table customers(

 customer_id int not null primary key identity (1,1),

first_name varchar(50) not null,   last_name varchar(50)

not null,   phone int not null,   email varchar(20),   street

varchar(20),  city varchar(20) not null,


  state varchar(20) not null,

  zip_code int

 );
   create table stores(
 store_id int not null primary key identity (1,1),

store_name varchar(50) not null,   phone int not

null,   email varchar(20),   street varchar(20),  city

varchar(20) not null,


  state varchar(20) not null,

  zip_code int

 );
```

```sql
create table staffs(

staff_id int not null primary key identity (1,1),   first_name
varchar(50) not null,   last_name varchar(50) not null,   email
varchar(20),   phone int not null,   active binary not null, store_id
int foreign key references stores(store_id) not null,

 manager_id int foreign key references staffs(staff_id) not null
);

create table orders(

order_id int primary key identity (1,1) not null,
customer_id int foreign key references customers(customer_id),
order_status varchar(20) not null,   order_date Date not null,
required_date Date,   shipped_date Date not null,

store_id int foreign key references stores(store_id),  staff_id int

foreign key references staffs(staff_id)

);

create table order_item(

 oreder_id int foreign key references orders(order_id) not null,

quantity int not null,   list_price
int not null,   discount int
);
```

**Output:**

- ⊟ ▦ dbo.customers
  - ⊟ 🗀 Columns
    - ⊶ customer_id (PK, int, not null)
    - ▤ first_name (varchar(50), not null)
    - ▤ last_name (varchar(50), not null)
    - ▤ phone (int, not null)
    - ▤ email (varchar(20), null)
    - ▤ street (varchar(20), null)
    - ▤ city (varchar(20), not null)
    - ▤ state (varchar(20), not null)
    - ▤ zip_code (int, null)
- ⊟ ▦ dbo.order_item
  - ⊟ 🗀 Columns
    - ⊶ oreder_id (FK, int, not null)
    - ▤ quantity (int, not null)
    - ▤ list_price (int, not null)
    - ▤ discount (int, null)

- ⊟ ▦ dbo.stores
  - ⊟ 🗀 Columns
    - ⊶ store_id (PK, int, not null)
    - ▤ store_name (varchar(50), not null)
    - ▤ phone (int, not null)
    - ▤ email (varchar(20), null)
    - ▤ street (varchar(20), null)
    - ▤ city (varchar(20), not null)
    - ▤ state (varchar(20), not null)
    - ▤ zip_code (int, null)

- ⊟ ▦ dbo.staffs
  - ⊟ 🗀 Columns
    - ⊶ staff_id (PK, int, not null)
    - ▤ first_name (varchar(50), not null)
    - ▤ last_name (varchar(50), not null)
    - ▤ email (varchar(20), null)
    - ▤ phone (int, not null)
    - ▤ active (binary(1), not null)
    - ⊶ store_id (FK, int, not null)
    - ⊶ manager_id (FK, int, not null)

- ⊟ ▦ dbo.orders
  - ⊟ 🗀 Columns
    - ⊶ order_id (PK, int, not null)
    - ⊶ customer_id (FK, int, null)
    - ▤ order_status (varchar(20), not null)
    - ▤ order_date (date, not null)
    - ▤ required_date (date, null)
    - ▤ shipped_date (date, not null)
    - ⊶ store_id (FK, int, null)
    - ⊶ staff_id (FK, int, null)
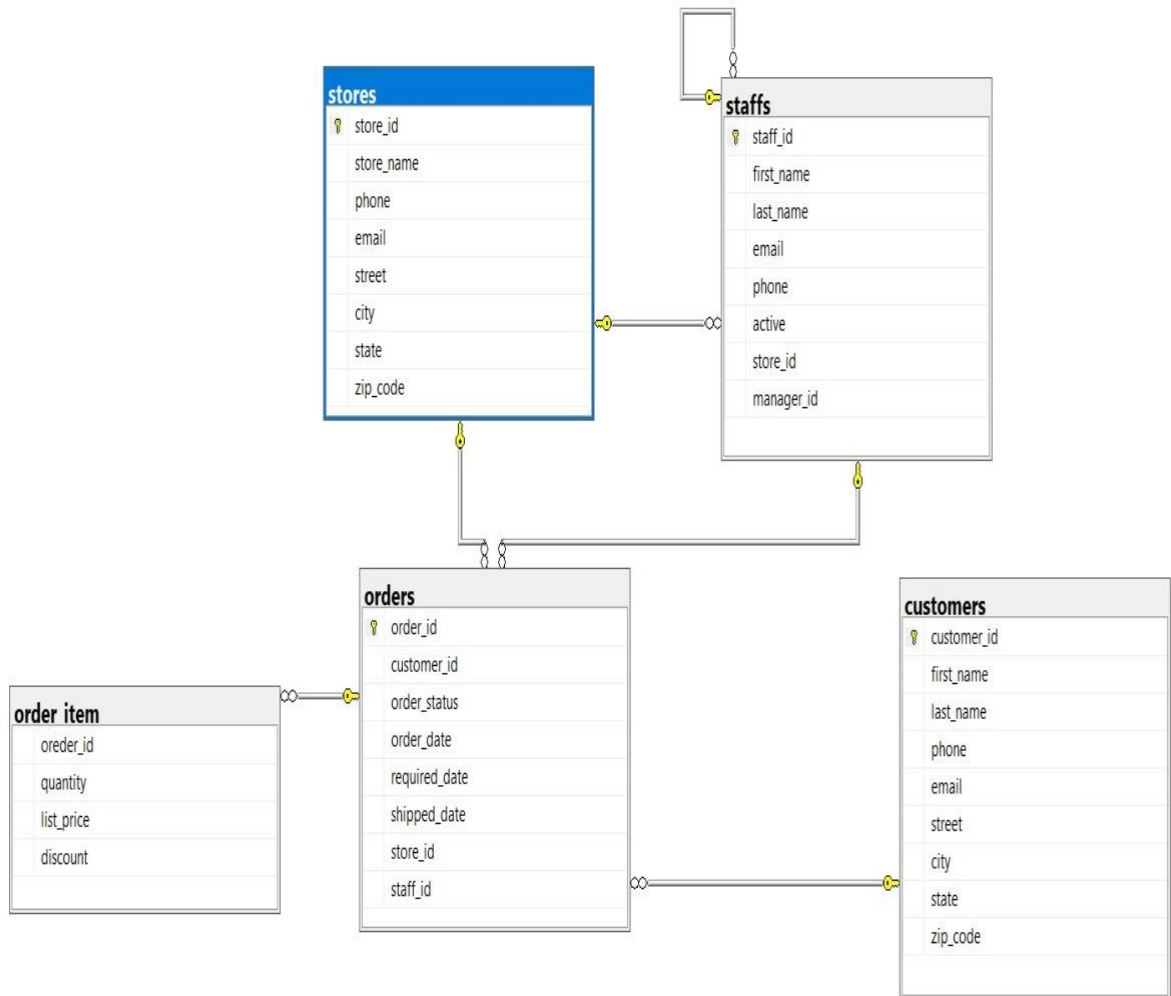
**Aim**: Write the SQL queries to perform various aggregate functions on table data.

1. List total deposit from deposit.

2. List total amount from andheri branch

3. Count total number of customers

4. Count total number of customer's cities.

5. Update the value dept_no to 10 where second character of emp. name is 'm'.

6. Update the value of employee name whose employee number is 103.

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

8. For each employee, display the employee number, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary

9. Modify your previous query to add a column that subtracts the old salary from the new salary. Label the column Increment.

**Code:**

```
select *from deposit;


 select sum(amount) from deposit;

select amount from deposit where bname='andheri';

select count(cname) from deposit;

select bname,count(cname) from deposit group by bname ;

select *from Employee;
```

update Employee set dept_no=10 where emp_name like '_m%';

update Employee set emp_name='panda' where emp_no=105;

```
select emp_no,emp_name,emp_sal,emp_sal+(emp_sal*15/100) as new_sal from
Employee;
select emp_no,emp_name,emp_sal,((emp_sal+(emp_sal*15/100))-emp_sal) as sub_sal
from Employee;


alter table Employee drop column new_salary;
select *from deposit;
select sum(amount) from deposit;
select amount from deposit where bname='andheri';
select count(cname) from deposit;
select bname,count(cname) from deposit group by bname ;


SELECT CAST( GETDATE() AS Date ) ;
```

**Output:**

| a_no | cname | bname | amount | a_date |
|------|-------|-------|--------|--------|
| 101 | Anil | Andheri | 7000 | 01-june-06 |
| 102 | Sunil | virar | 5000 | 15-july-06 |
| 103 | Jay | villeparle | 6500 | 12-march-06 |
| 104 | Vijay | andheri | 8000 | 17-sept-06 |
| 105 | Kayur | dadar | 7500 | 19-nov-06 |
| 106 | Mayur | borivali | 5500 | 21-dec-06 |

| sum(amount) |
|-------------|
| 39500 |

| amount |
|--------|
| 8000 |

| count(cname) |
|--------------|
| 6 |

| bname | count(cname) |
|-------|--------------|
| Andheri | 1 |
| andheri | 1 |
| borivali | 1 |
| dadar | 1 |
| villeparle | 1 |
| virar | 1 |

| emp_no | emp_name | emp_sal | emp_comm | dept_no |
|--------|----------|---------|----------|---------|
| 101 | Smith | 800 | | 10 |
| 102 | Snehal | 1600 | 300 | 25 |
| 103 | Adama | 1100 | | 20 |
| 104 | Aman | 3000 | | 10 |
| 105 | panda | 5000 | 50000 | 10 |
| 106 | Sneha | 2450 | 24500 | 10 |
| 107 | Anamika | 2975 | | 30 |

| emp_no | emp_name | emp_sal | new_sal |
|--------|----------|---------|---------|
| 101 | Smith | 800 | 920 |
| 102 | Snehal | 1600 | 1840 |
| 103 | Adama | 1100 | 1265 |
| 104 | Aman | 3000 | 3450 |
| 105 | panda | 5000 | 5750 |
| 106 | Sneha | 2450 | 2817 |
| 107 | Anamika | 2975 | 3421 |

| emp_no | emp_name | emp_sal | sub_sal |
|--------|----------|---------|---------|
| 101 | Smith | 800 | 120 |
| 102 | Snehal | 1600 | 240 |
| 103 | Adama | 1100 | 165 |
| 104 | Aman | 3000 | 450 |
| 105 | panda | 5000 | 750 |
| 106 | Sneha | 2450 | 367 |
| 107 | Anamika | 2975 | 446 |

# PRACTICAL-25

**Aim**: Write the SQL queries to perform numeric, date and String functions.

1. Retrieve all data from employee, jobs and deposit.

2. Give details of account no. and deposited rupees of customers having account opened between dates 01-01-06 and 25-07-06.

3. Display all jobs with minimum salary is greater than 4000.

4. Display name and salary of employee whose department no is 20. Give alias name to name of employee.

5. Display employee no,name and department details of those employee whose department lies in(10,20)

6. Display all employee whose name start with 'A' and third character is ' 'a'. 7. Display name, number and salary of those employees whose name is 5 characters long and first three characters are 'Ani'.

8. Display the non-null values of employees and also employee name second charactershould be 'n' and string should be 5 character long.

9. Display the null values of employee and also employee name's third character should be 'a'.

**Code:**

```
select *from employee;   select
*from job;
select *from deposit;

select a_no,amount from deposit where a_date between '06-01-01'and'06-07-25';

select *from job where min_sal > 4000;

select emp_name from employee  as name_of_employee where dept_no = 20;

select emp_no,emp_name from employee where dept_no in (10,20);

select *from employee where emp_name like '__a%';
```

```
select emp_no,emp_name,emp_sal from employee where emp_name like 'Ani%' and
length(emp_name)=5;

select emp_no,emp_name,emp_sal from employee where emp_name like '_n____' and
 emp_comm is not null;
  select emp_no,emp_name,emp_sal from employee where emp_name like '__a%' and emp_comm
is not null;
```

## Output:

| emp_no | emp_name | emp_sal | emp_comm | dept_no |
|--------|----------|---------|----------|---------|
| 101 | Smith | 800 | | 10 |
| 102 | Snehal | 1600 | 300 | 25 |
| 103 | Adama | 1100 | | 20 |
| 104 | Aman | 3000 | | 10 |
| 105 | panda | 5000 | 50000 | 10 |
| 106 | Sneha | 2450 | 24500 | 10 |
| 107 | Anamika | 2975 | | 30 |

| job_id | job_title | min_sal | max_sal |
|--------|-----------|---------|---------|
| IT PROG | Programmer | 4000 | 10000 |
| MK MGR | Marketing Manager | 9000 | 15000 |
| FI_MGR | Finance Manager | 8200 | 12000 |
| FI ACC | Account | 4200 | 9000 |
| LEC | Lecturer | 6000 | 17000 |
| COMP_OP | Computer Operator | 1500 | 3000 |

| a_no | cname | bname | amount | a_date |
|------|-------|-------|--------|--------|
| 101 | Anil | Andheri | 7000 | 01-june-06 |
| 102 | Sunil | virar | 5000 | 15-july-06 |
| 103 | Jay | villeparle | 6500 | 12-march-06 |
| 104 | Vijay | andheri | 8000 | 17-sept-06 |
| 105 | Keyur | dadar | 7500 | 19-nov-06 |
| 106 | Mayur | borivali | 5500 | 21-dec-06 |

| job_id | job_title | min_sal | max_sal |
|--------|-----------|---------|---------|
| MK MGR | Marketing Manager | 9000 | 15000 |
| FI_MGR | Finance Manager | 8200 | 12000 |
| FI ACC | Account | 4200 | 9000 |
| LEC | Lecturer | 6000 | 17000 |

| emp_name |
|----------|
| Adama |

| emp_no | emp_name |
|--------|----------|
| 101 | Smith |
| 103 | Adama |
| 104 | Aman |
| 105 | panda |
| 106 | Sneha |

| emp_no | emp_name | emp_sal | emp_comm | dept_no |
|--------|----------|---------|----------|---------|
| 103 | Adama | 1100 | | 20 |
| 104 | Aman | 3000 | | 10 |
| 107 | Anamika | 2975 | | 30 |

| emp_no | emp_name | emp_sal |
|--------|----------|---------|
| 102 | Snehal | 1600 |

**AIM:**

**CODE:** **<!DOCTYPE html>**

**<html>**

**<head>**

**<style>**

**table {**

**width: 100%;**

**border-collapse: collapse;**

**}**

**th, td {**

**padding: 8px;**

**text-align: left;**

**border-bottom: 1px solid #ddd;**

**}**

**th {**

**background-color: #f2f2f2;**

**}**

**caption {**

**font-size: 24px;**

**font-weight: bold;**

**margin-bottom: 10px;**

```
      }

      .highlight {
        background-color: yellow;
      }

      .center {
        text-align: center;
      }
    </style>
  </head>
  <body>
    <table>
      <caption>Poster Presentation</caption>
      <thead>
        <tr>
          <th>Topic</th>
          <th>Author</th>
          <th>Time</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td rowspan="2">Introduction to HTML</td>
          <td>John Doe</td>
```

```
   <td>9:00 AM</td>
 </tr>
 <tr>
  <td>Jane Smith</td>
  <td class="highlight">10:00 AM</td>
 </tr>
 <tr>
  <td colspan="3" class="center">Break</td>
 </tr>
 <tr>
  <td>JavaScript Fundamentals</td>
  <td>Michael Johnson</td>
  <td>11:00 AM</td>
 </tr>
 <tr>
  <td rowspan="3">CSS Styling Techniques</td>
  <td>Emily Davis</td>
  <td>1:00 PM</td>
 </tr>
 <tr>
  <td>Robert Wilson</td>
  <td>2:00 PM</td>
 </tr>
 <tr>
  <td>Lisa Thompson</td>
```

```
        <td>3:00 PM</td>
      </tr>
    </tbody>
  </table>
</body>
</html>
```

**OUTPUT:**



## Poster Presentation

| Topic | Author | Time |
|---|---|---|
| Introduction to HTML | John Doe | 9:00 AM |
| | Jane Smith | 10:00 AM |
| | Break | |
| JavaScript Fundamentals | Michael Johnson | 11:00 AM |
| CSS Styling Techniques | Emily Davis | 1:00 PM |
| | Robert Wilson | 2:00 PM |
| | Lisa Thompson | 3:00 PM |

**PRACTICAL-27**

**AIM:** Make a Resume using the HTML tags with CSS.

**CODE:**

```html
<!DOCTYPE html>
<html>
<head>
 <title>Intern Resume</title>
 <style>
  body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 20px;
  }

  h1 {
    color: #333;
  }

  .section {
    margin-bottom: 20px;
  }

  .section h2 {
    color: #777;
    font-size: 18px;
```

```
      margin-bottom: 10px;

    }


  .section p {

    margin: 0;

  }


  .section ul {

    margin: 0;

    padding: 0;

    list-style-type: none;

  }


  .section li {

    margin-bottom: 5px;

  }
 </style>
</head>
<body>
 <h1>Intern Resume</h1>


 <div class="section">

   <h2>Contact Information</h2>

   <p>Name: AMAN NASIT</p>

   <p>Email: amannasit123@gmail.com</p>
```

```html
    <p>Phone:+919423541673</p>
  </div>

  <div class="section">
   <h2>Education</h2>
   <ul>
      <li>Bachelor of Technology in information Technology, Silver Oak University</li>
      <li>Expected Graduation: May 2026</li>
   </ul>
  </div>

  <div class="section">
   <h2>Skills</h2>
   <ul>
     <li>Programming: HTML, CSS, JavaScript</li>
     <li>Database: SQL</li>
     <li>Version Control: Git</li>
   </ul>
  </div>

  <div class="section">
   <h2>Experience</h2>
   <p>Intern at ABC Company, June 2022 - August 2022</p>
   <ul>
```

```
<li>Assisted with front-end web development tasks</li>

<li>Collaborated with team members on project
deliverables</li>

<li>Tested and debugged website issues</li>

</ul>

</div>


</body>

</html>
```

**OUTPUT:**