Information Practice(065)

Class XI

Chapter –Computer System

1. **Name the software required to make a computer functional. Write down its two primary services.**

Ans. Operating System. Two of its primary services are: managing CPU and managing primary

memory.

2. **What is the need of RAM? How does it differ from ROM ?**

Ans. The RAM is needed for the smooth running of a computer's operations as it provides memory for reading as well writing; its contents can be modified as and when required. ROM, on the other hand, stores some instructions which are permanent. ROM cannot be written onto. One can only read instructions from ROM.

Thus RAM along with ROM, is required for the smooth functioning of a computer system.

4. **Draw the block diagram of computer system. Briefly write about the functionality of each component?**

3. **What is the need for secondary memory?**

Ans. The primary memory holds the data and instructions only during processing and after that, it gets erased. One cannot store data and instructions permanently on RAM.

To permanently store data and instructions, the secondary memory is required.

5. **Differentiate between proprietary software and freeware software. Name two software of each type.**

Ans. Proprietary software is software that is owned by an individual or a company. It comes with a license for its use, and its source code is almost always kept secret.

Freeware is software, most often proprietary, that is distributed at no monetary cost to the end user.

However, its source code is not available and it comes with a license too.

6. Mention any browsers used for browsing the Internet.

Ans. Chrome, Safari, Firefox.

7. Name the input/output device used to do the following:

(a) To output audio

(b) To enter textual data

(c) To make hard copy of a text file

(d) To display the data/information

(e) To enter audio based command

(f) To build 3D models

(g) To assist a visually-impaired individual in entering data

Ans.(a) speaker (b) keyboard

(c) printer (d) monitor (e) MIC

(f) 3D printer

(g) Braille Keyboard

8. Identify the category (system, application, programming tool) of the following software:

(a) Compiler

(b) Assembler

(c) Ubuntu

(d) Text editor

Ans. (a) system software (b) system software (c) system software (d) application software

9. Convert the following into bytes:

(a) 2 MB

(b) 3.7 GB

(c) 1.2 TB

Ans. (a) 2 MB = 2 × 1024x1024 Bytes

(b) 3.7 GB = 3.7 x 1024 x 1024x1024 Bytes

(c) 1.2 TB = 1.2 x 1024 x 1024 x 1024x1024 Bytes

10. What is the security threats involved when we throw away electronic gadgets that are non-functional ?

Ans. While discarding electronic gadgets, if we have not taken care of deleting the stored data permanently by proper tools, then such data may be undeleted and misused.

11. Write down the type of memory needed to do the following:

(a) To store data permanently

(b) To execute the program.

(c) To store the instructions which cannot be overwritten.

Ans. (a) secondary memory (storage devices)

(b) RAM

(c) ROM

1. **WAP to print “Hello World**

```
print("Hello world")
```

```
Hello world
```

2. **WAP to print “Hello + Username” with the user’s name on Python?**

```
usertext = input("What is your name? ")
print("Hello", usertext)
```

```
What is your name? roshni
Hello roshni
```

3. **WAP to to add 2 numbers entered on Python?**

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
sum = num1+num2
print("Sum is = ", sum)
```

```
Enter first number: 50
Enter second number: 80
Sum is =  130.0
```

4. **WAP to find the Average of 2 Entered Numbers on Python?**

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
average =(num1+num2)/2
print("Average is = ", average)
```

```
Enter first number: 50
Enter second number: 60
Average is =  55.0
```

5. **WAP to find the Average of 3 Written Grades entered on Python?**

```
firstexam = float(input("your first exam : "))
```

```
secondexam = float(input("your second exam : "))
thirdexam = float(input("your third exam : "))
average =(firstexam+secondexam+thirdexam)/3
print("average : ", average)
```

```
your first exam : 80
your second exam : 70
your third exam : 90
average :  80.0
```

6. **WAP to show the Class Pass Status (PASSED — FAILED) of the Student whose Written Average Has Been Entered on Python?**

```
average = int(input("enter average : "))

if(int(average)>=40):

   print("Passed")

else:

   print("Failed")
```

```
enter average : 78
Passed
```

7. **WAP to find out if the entered number is odd or even on Python?**

```
num = int(input("Enter a number: "))
if (num % 2) == 0:
  print("Even")
else:
  print("Odd")
```

```
Enter a number: 78
Even
```

8. **WAP to find out if the entered number is Positive, Negative, or 0 on Python?**

```
num = float(input("Enter a number: "))
if num > 0:
  print("Positive number")
elif num == 0:
  print("Zero")
else:
  print("Negative number")
```

```
Enter a number: -98
Negative number
```

9. **WAP to calculate body mass index on Python? (index = weight/(height*height))**

```
print("body mass index calculation program")
height = float(input("enter height (m):"))
weight = int(input("enter weight (kg):"))
index  = weight/(height*height)
print("BMI is ", index)
```

```
body mass index calculation program
enter height (m):1.524
enter weight (kg):55
BMI is  23.680602916761387
```

10. **WAP to show if the person whose age is entered can get a driver's license on Python?**

```
age = input("enter age : ")

if(int(age)<18):

   print("Your Age Is Not Eligible To Get A Driver's License")

else:
```

```
    print("Your Age Is Eligible To Get Your License")
```

```
enter age : 32
Your Age Is Eligible To Get Your License
```

11. WAP to List Numbers 1–100 on the Screen on Python?

```
for i in range(1,101):
    print(i)
```

12. WAP to print Even Numbers 1–100 on Python?

```
for i in range(1,101):
    if i%2==0:
        print(i)
```

13. WAP to List Odd Numbers from 1–100 on Python?

```
for i in range(1,101):
    if i%2!=0:
        print(i)
```

14. WAP to find numbers between 1 and 100 that are divided by 3 and 5 on Python?

```
for i in range(1,101):
    if i%3==0 or i%5==0:
        print(i)
```

15. WAP to list Numbers from 1 to n(User-Entered Number) on Python?

```
num = input('enter number : ')
for i in range(1,int(num)+1):
    print(i)
```

```
enter number : 5
1
2
3
4
5
```

16. WAP to find the Area and Perimeter of a Rectangle with Its Sides on Python?

```
short = int(input("Enter short side : "))
tall = int(input("Enter tall side : "))
area = short*tall
perimeter =2*(short+tall)
print("area: ",area)
print("perimeter: ",perimeter)
```

```
Enter short side : 5
Enter tall side : 10
area:  50
perimeter:  30
```

17. WAP to print the letters of the entered text one under the other on Python?

```
word = input("Enter the word : ")
for i in word:
    print(i)
```

```
Enter the word : computer
c
o
m
p
u
t
e
r
```

18. WAP to show the sum of numbers between two numbers the user has entered on Python?

```
sum=0;

num1 = int(input("first number: "))

num2 = int(input("second number: "))

for i in range(num1+1,num2):

   sum+=i

print("Sum of numbers between {0} and {1} :
{2}".format(num1,num2,sum))
```

```
first number: 5
second number: 10
Sum of numbers between 5 and 10 : 30
>
```

19. For example, let's ask the user about their choice of cinema or theatre. You have to pay 10 dollars to watch movies and 5 dollars for theatre. Students get 50% discount. There is no discount; If he is not a student, let's write a document that calculates the non-discounted amount and prints it.

```
selection = input("Press (1) for Cinema, (2) for Theater : ")
student = input("Are you student(Y/N) : ")
price = 0
#non-discounted fee calculation
if selection == '1':
   price = 10 #cinema
elif selection == '2':
   price = 5 #theatre

#student discount
if student =='Y' or student =='y':
   price=price / 2  #%50
print(" The fee you have to pay : ",price)
```

```
Press (1) for Cinema, (2) for Theater : 1
Are you student(Y/N) : y
The fee you have to pay :  5.0
```

20.WAP to find out if the entered number is Prime or Not on Python?

```
num=int(input("Enter any number : "))
f=0
if num == 0 or num == 1:
   f = 1
for i in range(2,num):
   if num % i == 0:
      f = 1
if f == 1:
   print(num," is not prime")
else:
   print(num, " is prime number")
```

```
Enter any number : 13
13  is prime number
```

21.WAP to calculate the increased salary of the worker whose salary and raise rate is entered on Python?

```
newsalary = 0
salary = int(input("enter new salary : "))
raisesal = int(input("salary raise rate(%) : "))
newsalary = salary+(salary *raisesal/100)
print("increased salary :",newsalary)
```

```
enter new salary : 200
salary raise rate(%) : 5
increased salary : 210.0
```

22. WAP to calculate the area and circumference of the circle whose radius is entered using the function on Python?

```
import math

radius = float(input("Please Enter the radius of a circle: "))
diameter = 2 * radius
circumference = 2 * math.pi * radius
area = math.pi * (radius ** 2)

print("Diameter Of a Circle = ",diameter)
print(" Circumference Of a Circle = ",circumference)
print(" Area Of a Circle = ",area)
```

```
Please Enter the radius of a circle: 5
Diameter Of a Circle =  10.0
 Circumference Of a Circle =  31.41592653589793
 Area Of a Circle =  78.53981633974483
```

23. WAP to sum the digits of the number the user entered on Python?

```
num=int(input("Enter any number : "))
sum =0
while num != 0:
   r = num % 10
   sum = sum + r
   num = num // 10
print("Product of digits = ", sum)
```

```
Enter any number : 145
Product of digits =  10
```

24. WAP to multiply the digits of the number entered by the user (except zero) on Python?

```
num=int(input("Enter any number : "))
prod =1
while num != 0:
   r = num % 10
```

```
    if(r!=0):
        prod = prod * r
    num = num // 10
print("Product of digits = ", prod)
```

```
Enter any number : 340
Product of digits =  12
```

25. WAP to print the programmer's name on the screen using the "for loop" and "while loop" on Python?

```
name=input("Enter your name : ")
count=int(input("How many time do you want to print your name :"))
print("Using for loop --")
for i in range(count):
    print(name)

print()

print("Using while loop --")
i=1
while i<=count:
    print(name)
    i+=1
```

```
Enter your name : roshni
How many time do you want to print your name :5
Using for loop --
roshni
roshni
roshni
roshni
roshni

Using while loop --
roshni
roshni
roshni
roshni
roshni
```

26. WAP to check if he given number is of one digited or two digited or three digited or more than three digited.

```
no=int(input("Enter number greater than zero: "))
if no> 0 and no<10:
    print(no," is one digit no.")
elif no>0 and no<100:
    print(no," is two digit no.")
elif no>0 and no<1000:
    print(no," is three digit no.")
else:
    print(no," is more than three digit no.")
```

```
Enter number greater than zero: 345
345  is three digit no.
```

27. WAP to check if the given character is an uppercase letter or lowercase letter or a digit or a special character.

```
ch = input("Enter a character to check:")
if ord(ch)>=65 and ord(ch)<=90:
    print("The character", ch, "  is an uppercase letter")
elif ord(ch)>=97 and ord(ch)<=122:
```

```
    print("The character", ch, "  is a lowercase letter")
elif ord(ch)>=48 and ord(ch)<=57:
    print("The character", ch, "  is a digit")
else:
    print("The character", ch, "  is a special chacracter")
```

```
Enter a character to check:F
The character F   is an uppercase letter
```

28. WAP that reads three positive numbers a, b, c and determines whether they can form the three sides of a triangle.

```
a = int(input("Enter First Side of a Triangle : "))
b = int(input("Enter Second Side of a Triangle : "))
c = int(input("Enter Third Side of a Triangle : "))
if a + b > c and a + c > b and c + b > a:
    print("Triangle can be formed !")
else:
    print("Triangle cannot be formed.")
```

```
Enter First Side of a Triangle : 6
Enter Second Side of a Triangle : 3
Enter Third Side of a Triangle : 7
Triangle can be formed !
```

29. A toy vendor supplies three types of toys: Battery Based Toys, Key-based Toys, and Electrical Charging Based Toys. The vendor gives a discount of 10% on orders for battery-based toys if the order is for more than Rs. 1000. On orders of more than Rs. 100 for key-based toys, a discount of 5% is given, and a discount of 10% is given on orders for electrical charging based toys of value more than Rs. 500. Assume that the numeric codes 1,2 and 3 are used for battery based toys, key-based toys, and electrical charging based toys respectively. Write a program that reads the product code and the order amount and prints out the net amount that the customer is required to pay after the discount.

```
print("1. For Battery based Toys")
print("2. For Key based Toys")
```

```
print("3. Electric chargin based Toys")
opt = int(input("Enter the product code (1,2 or 3)?:"))
amt = int(input("Enter the amount:"))
if opt==1:
    if amt>1000:
        dis = amt * 0.1
    else:
        dis = 0
elif opt==2:
    if amt>100:
        dis = amt * 0.05
    else:
        dis=0
elif opt==3:
    if amt>500:
        dis = amt*0.1
    else:
        dis = 0
else:
    print("Product is not available")
bill_amt= amt - dis
print("Customer has to pay:",bill_amt)
```

```
1. For Battery based Toys
2. For Key based Toys
3. Electric chargin based Toys
Enter the product code (1,2 or 3)?:2
Enter the amount:300
Customer has to pay: 285.0
```

30. WAP to do the following operations :

Read any two positive integer numbers (say n1 & n2) and one character type operator (say opr). Note that opr is any mathematical operator. Depending upon the operator, do the appropriate operation. e. g. if opr is '+' then the display the value obtained by evaluating the expression (n1 + n2).

```
n1 = int(input("Enter n1:"))
```

```
n2 = int(input("Enter n2:"))
print("+ for addition\n")
print("- for addition\n")
print("* for addition\n")
print("/ for division\n")
ch = input("Enter the choice:")
if ch=='+':
   ans = n1 + n2
elif ch == '-':
   ans = n1 - n2
elif ch == '*':
   ans = n1 * n2
elif ch == '/':
   ans = n1 / n2
else:
   print("Invalid choice")
print("The answer is:",ans)
```

```
Enter n1:7
Enter n2:2
+ for addition

- for addition

* for addition

/ for division

Enter the choice:/
The answer is: 3.5
```

31.The Paschim Gujarat Vij Company Ltd. computes the electricity bill based on the following matrix:

Units Consumed	Charges
0-100	0.50 per unit
101-200	Rs. 50 plus Rs. 1 per unit over 100 units
201-300	Rs. 150 plus 1.50 per unit over 200 units
> 300	Rs. 300 plus Rs.2 per unit over 300 units

1) Ask user to enter the Past meter reading and current meter reading.
2) Find the units consumed.
3)Compute the bill according to given matrix.

```
pmr = int(input("Enter past meter reading:"))
cmr = int(input("Enter current meter reading:"))
uc = cmr - pmr
if uc>0 and uc<=100:
    bill_amt = uc * 0.50
elif uc>100 and uc<=200:
    bill_amt = 50 + (uc-100) * 1
elif uc>200 and uc<=300:
    bill_amt = 150 + (uc-200) * 1.50
else:
    bill_amt = 300 + (uc-300) * 2
print("The bill amount is:",bill_amt)
```

```
Enter past meter reading:100
Enter current meter reading:300
The bill amount is: 150
```

32.A cloth showroom has announced the following discounts on the purchase of specific items :

Amount	Shorts	Pants	Shits/T-Shirts
0-100	–	3%	5%
101-200	5%	8%	10%
201-300	10%	12%	15%
Above 300	18%	20%	22%

Ask user to enter the amount and assign following code for the items such as sh for shorts, p for pans and sht for shirts/t-shirts. Compute the discount and net amount paid by customer.

```
print("sh for Short")
print("p for Pants")
print("t for t-shirts")
opt = input("Enter the product code ('sh','p','t')?:")
amt = int(input("Enter the amount:"))
if opt=="sh":
    if amt>0 and amt<100:
        dis = 0
    elif amt>100 and amt<200:
        dis = amt * 0.05
    elif amt>201 and amt<300:
        dis = amt * 0.1
    else:
        dis = amt * 0.18
elif opt=="p":
    if amt>0 and amt<100:
        dis = amt * 0.03
    elif amt>100 and amt<200:
        dis = amt * 0.08
    elif amt>201 and amt<300:
        dis = amt * 0.12
    else:
        dis = amt * 0.20
elif opt=="t":
    if amt>0 and amt<100:
        dis = amt * 0.05
    elif amt>100 and amt<200:
```

```
        dis = amt * 0.10
    elif amt>201 and amt<300:
        dis = amt * 0.15
    else:
        dis = amt * 0.22
bill_amt = amt - dis
print("Customer has to pay:",bill_amt)
```

```
sh for Short
p for Pants
t for t-shirts
Enter the product code ('sh','p','t')?:p
Enter the amount:250
Customer has to pay: 220.0
```

33.A transport company charges the fare according to following table:

Distance	Charges
1-50	8 Rs./Km
51-100	10 Rs./Km
> 100	12 Rs/Km

Ask user to enter the distance and compute the fare.

```
distance = int(input("Enter distance:"))
if distance>=1 and distance<=50:
    fare = distance * 8
elif distance>=51 and distance<=100:
    fare = distance * 10
elif distance>100:
    fare = distance * 12
else:
    print("Invalid fare")
print("The total fare is:",fare)
```

```
Enter distance:60
The total fare is: 600
```

34.Find the sum of the series (1*1) + (2*2) + (3*3)+ ... + (n*n)

```
n = int(input("Enter the value of n : "))
sums = 0
for i in range(1, n + 1):
    sums += (i * i);
print(sums)
```

```
Enter the value of n : 6
91
```

35.WAP to read 3 numbers and find the highest and smallest number.

```
a = int(input("Enter the first number : "))
b = int(input("Enter the second number : "))
c = int(input("Enter the third number : "))

if a >= b and a >= c:
    max = a
elif b >= a and b >= c:
    max = b
else:
    max = c

print(max, "is the largest of three numbers.")
```

```
Enter the first number : 5
Enter the second number : 7
Enter the third number : 99
99 is the largest of three numbers.
```

Smallest no : Do it yourself..

36.WAP to print the following

- First 10 even numbers

```
for i in range(2,22,2):
    print(i)
```

- First 10 odd numbers

```
for i in range(1,20,2):
    print(i)
```

- First 10 natural numbers

```
for i in range(1,11):
    print(i)
```

- First 10 whole numbers

```
for i in range(0,10):
    print(i)
```

37.WAP to print first 10 integers and their squares.

```
print("Numbers","\t","Squares")
for i in range(1,11):
    print(i,"\t\t\t",i**2)
```

```
Numbers         Squares
1               1
2               4
3               9
4               16
5               25
6               36
7               49
8               64
9               81
10              100
```

38.WAP to print the series 10,20,30,…….300

```
for i in range(10,301,10):
    print(i,end=" ")
```

```
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160
    170 180 190 200 210 220 230 240 250 260 270 280 290
    300 >
```

39.WAP to reverse a number accepted from the user.

```
num = int(input("Enter the number : "))
reversed_num = 0

while num != 0:
    digit = num % 10
    reversed_num = reversed_num * 10 + digit
    num //= 10

print("Reversed Number: " , reversed_num)
```

```
Enter the number : 234
Reversed Number:  432
```

40.WAP to find the factorial of a number.

```
num=int(input("Enter any number : "))
fact=1
i=1
while i <= num:
    fact = fact * i
    i+= 1
print("Factorial is = ", fact)
```

```
Enter any number : 5
Factorial is =  120
```

41.WAP to print Fibonacci series up to n terms.

```
nterms = int(input("How many terms? "))
# first two terms
n1, n2 = 0, 1
count = 0
print("Fibonacci sequence:")
while count < nterms:
    print(n1)
    nth = n1 + n2
    # update values
```

```
    n1 = n2
    n2 = nth
    count += 1
```

```
How many terms? 7
Fibonacci sequence:
0
1
1
2
3
5
8
```

42.WAP to print the sum of the following series.

1 + 2 + 6 + 4 + 120 … n terms

```
n = int(input("Enter the number of terms: "))
sum = 0
prod=1
for i in range(1,n+1):
    prod = i*prod
    print(prod,end="+")
    sum= sum + prod
print("=", sum)
```

```
Enter the number of terms: 7
1+2+6+24+120+720+5040+= 5913
```

Programs on Iteration Statements (Loops)

1. **Write a program to print first 10 natural numbers.**

```
print("====The First 10 Natural Numbers====")
for i in range(1, 11):
    print(i)
```

Output ->

```
====The First 10 Natural Numbers====
1
2
3
4
5
6
7
8
9
10
```

2. **Write a program to print first 10 even numbers.**

```
count = 0
num = 1
while count <= 10:
    if num % 2 == 0:
        print (num)
        count = count+1
    num = num +1
```

Output ->

```
2
4
6
8
10
12
14
16
18
20
22
```

3. **WAP to print first n even numbers in reverse order.**

```
n= int(input("Enter n : "))
num= 0
x= 0
list1= []
while n>x:
    if num %2 ==0:
        list1.append(num)
        x+=1
    num+=1
list1.reverse()
print(list1)
```

Outpt->

```
Enter n : 5
[8, 6, 4, 2, 0]
```

4. **WAP to print table of a number accepted from user.**

```
num=int(input("Enter any number : "))
for i in range(1,11):
    print(num*i)
```

Output->

```
Enter any number : 5
5
10
15
20
25
30
35
40
45
50
```

5. **WAP to display the product of the digits of a number accepted from user.**

```
num=int(input("Enter any number : "))
prod =1
while num != 0:
   r = num % 10
   prod = prod * r
   num = num // 10
print("Product of digits = ", prod)
```

Output->

```
Enter any number : 234
Product of digits =  24
```

6. **WAP to find the factorial of a number.**

```
num=int(input("Enter any number : "))
fact=1
i=1
while i <= num:
   fact = fact * i
   i+= 1
print("Factorial is = ", fact)
```

Output ->

```
Enter any number : 5
Factorial is =  120
```

7. **WAP to check a number is prime or not.**

```
num=int(input("Enter any number : "))
f=0
if num == 0 or num == 1:
   f = 1
for i in range(2,num):
   if num % i == 0:
      f = 1
if f == 1:
   print(num," is not prime")
else:
   print(num, " is prime number")
```

Output->

```
Enter any number : 5
5  is prime number
```

```
Enter any number : 18
18  is not prime
```

8. Accept 10 numbers from the user and calculate their average.

```
sum = 0
for i in range(10):
    n = float(input('Enter a number: '))
    sum = sum + n
average = sum/10
print("The average of these numbers is: ", average)
```

Output->

```
Enter a number: 1
Enter a number: 2
Enter a number: 3
Enter a number: 4
Enter a number: 5
Enter a number: 6
Enter a number: 7
Enter a number: 8
Enter a number: 9
Enter a number: 10
The average of these numbers is:  5.5
```

9. WAP to print the Fibonacci series till n terms.

```
nterms = int(input("How many terms? "))
# first two terms
n1, n2 = 0, 1
count = 0
print("Fibonacci sequence:")
while count < nterms:
    print(n1)
    nth = n1 + n2
    # update values
    n1 = n2
    n2 = nth
    count += 1
```

Output->

```
How many terms? 6
Fibonacci sequence:
0
1
1
2
3
5
```

10. Print all characters in “Computer”

```
for i in "Computer":
    print(i)
```

Output->

```
C
o
m
p
u
t
e
r
```

11. WAP to print the below patterns.

```
*
* *
* * *
* * * *
* * * * *
```

```
for i in range(0,5):
    for j in range(0,i+1):
        print("*",end=" ")
    print()
```

```
0
1 1
2 2 2
3 3 3 3
4 4 4 4 4
```

```
for i in range(0,5):
    for j in range(0,i+1):
        print(i,end=" ")
    print()
```

```
1
1 2
1 2 3
1 2 3 4
```

```
for i in range(1,5):
    for j in range(1,i+1):
        print(j,end=" ")
    print()
```

```
* * * * *
* * * *
* * *
* *
*
```

```
for i in range(5,0,-1):
    for j in range(i):
        print("*",end=" ")
    print()
```

```
5 5 5 5 5
4 4 4 4
3 3 3
2 2
1
```

```
for i in range(5,0,-1):
    for j in range(1,i+1):
        print(i,end=" ")
    print()
```

```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

```
for i in range(5,0,-1):
    for j in range(1,i+1):
        print(j,end=" ")
    print()
```

```
A
B C
D E F
G H I J
K L M N O
```

```
val=65
for i in range(1,6):
    for j in range(1,i+1):
        print(chr(val),end=" ")
        val=val+1
    print()
```

```
val=65
for i in range(1,5):
    for j in range(1,5):
        print(chr(val),end=" ")
    print()
```

Programs on Decision Making Statements

1. Write a program to print monthly telephone bill. Calculate the bill based on the following criteria.

Number of call	Price/call
Upto 100	0.00
>100	2.50
>500	4.50
>1000	7.50

If the number of calls is 250 then no charges for first 100 calls and remaining 150 calls will be charged as 2.50.

Solution:

```
calls=int(input("Enter number of calls : "))

bill=0

if(calls<=100):  # first 100 call free

    bill=0

elif (calls>100 and calls<=500): # 100 free after 2.50 rs/call

    bill=100*0+(calls-100)*2.50

elif (calls>500 and calls<=1000):

    bill=100*0 + 400*2.50 + (calls-500)*4.50

else:

    bill=100*0 + 400*2.50 + 500*4.5 + (calls-1000)*7.50

print("Total bill amount is", bill)
```

Output ->

```
Enter number of calls : 1200

Total bill amount is 4750.0
```

2. Program that takes a number and checks whether the given number is odd or even.

   Solution:

```
num = int(input("Enter any number : "))
if (num % 2) == 0:
   print ("The number is even")
else:
   print ("The provided number is odd")
```

   Output:

```
Enter any number : 56
The number is even
```

3. Program to accept 3 integers and print the largest of the three.

   Solution:

```
a = int(input("Enter the first number : "))
b = int(input("Enter the second number : "))
c = int(input("Enter the third number : "))

if a >= b and a >= c:
   max = a
elif b >= a and b >= c:
   max = b
else:
   max = c

print(max, "is the largest of three numbers.")
```

   Output:

```
Enter the first number : 56
Enter the second number : 78
Enter the third number : 99
99 is the largest of three numbers.
```

4. Program that reads two numbers and an arithmetic operator and displays the result.

   Solution:

```
num1= float(input("Enter first number: "))
num2=float(input("Enter second number: "))
op=input("enter operator [+-*/%] ")
result=0
if op=='+':
   result=num1+num2
elif op =='-':
   result=num1-num2
elif op=='*':
   result=num1*num2
elif op=='/':
   result=num1/num2
else:
```

```
    print("Invalid operator!!")
print(num1,op,num2,'=',result)
```

Output:

```
Enter first number: 8
Enter second number: 2
enter operator [+-*/%] /
8.0 / 2.0 = 4.0
```

5. Program to print whether a given character is an uppercase or lowercase or a digit or a special character.

Solution:

```
ch = input("Enter a single character : ")
if ch >= 'A' and ch <= 'Z':
    print("You have entered an Uppercase character")
elif ch >= 'a' and ch <= 'z':
    print("You have entered an Lowecase character")
elif ch >='0' and ch <= '9':
    print("You have entered a digit")
else:
    print("You have entered a special character")
```

Output:

```
Enter a single character : V
You have entered an Uppercase character
```

Working with Lists and Dictionaries

Qs 1. What will be the output of the following statements?

a)
```
list1 = [12,32,65,26,80,10]
list1.sort()
print(list1)
```

b)
```
list1 = [12,32,65,26,80,10]
print(sorted(list1) )
```

c)
```
list1 = [1,2,3,4,5,6,7,8,9,10]
print(list1[::-2])

print(list1[:3] + list1[3:] )

list1 = [1,2,3,4,5]
print(list1[len(list1)-1])
```

Ans.

(a) [10, 12, 26, 32, 65, 80]

(b) >>> sorted(list1) **[10, 12, 26, 32, 65, 80]**

>>> print(list1) **[12, 32, 65, 26, 80, 10]**

(c) >>> list1[::-2] **[10, 8, 6, 4, 2]**

>>> list1[:3] + list1[3:] **[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]**

(d) 5

len(list1):- len(list1) will return the length of list as 5.

list1 = [1,2,3,4,5]

list1[len(list1)-1]

list1[5-1] list1[4]

Qs 2. Consider the following list myList. What will be the elements of myList after each of the following operations?

myList = [10,20,30,40]

a) myList.append([50,60])

b) myList.extend([80,90])

Ans.

(**a) Append():**

- In python, the append() function adds the given argument as a single element at the end of the list and the length of the list increases by one.
- In this question, the argument is a list containing the element 50, 60.

Outpu: myList = [10, 20, 30, 40, [50, 60]]

(b) Extend():

- In python, the extend() function takes a list as an argument and appends each element of the list in the same order.
- In this question, the argument is a list containing the element 80, 90.

Output: myList = [10, 20, 30, 40, [50, 60], 80, 90]

Qs 3. What will be the output of the following code segment?

```
myList = [1,2,3,4,5,6,7,8,9,10]

for i in range(0,len(myList)):

        if i%2 == 0:

                print(myList[i])
```

Ans.

In this question we use the operation i%2 therefore:

The Output will be the elements at index 0, 2, 4, 6, 8 i.e. [1, 3, 5, 7, 9]

Output:

1

3

5

7

9

Qs 4. What will be the output of the following code segment?

a)
```
myList = [1,2,3,4,5,6,7,8,9,10]
del myList[3:]
print(myList)
```

b)
```
myList = [1,2,3,4,5,6,7,8,9,10]
del myList[:5]
print(myList)
```

c)
```
myList = [1,2,3,4,5,6,7,8,9,10]
del myList[::2]
print(myList)
```

Ans.

(a) The operation in line 2 will delete the elements from index 3 to the last index of the list.

Therefore, the output will be:

myList = [1, 2, 3]

(b) The operation in line 2 will delete the elements from start (index 0) to index 4 of the list.

Therefore, the output will be:

myList = [6, 7, 8, 9, 10]

(c) The operation in line 2 will delete the elements from the start to the end of the list with step size 2. i.e. it will delete elements starting from index 0, index 2, index 4, index 6, and so on.

Therefore, the output will be:

myList = [2, 4, 6, 8, 10]

Qs 5. Differentiate between append() and extend() methods of list.

Ans.

a) Append():

- The append() function takes a single element as an argument and appends it to the list. The argument can be any data type, even a list. The argument will be treated as a single element.

- For example:

  >>list1 = [10, 20, 30]

  >>list1.append(40)

  Now list1 will become [10, 20, 30, 40]

  >>list1 = [10, 20, 30]

  >>list1.append([40, 50])

  Here [40, 50] is a list that will be considered as a single argument, and will be added to the list at index 3.

  Now 'list1' will become [10, 20, 30, [40, 50]]

b) Extend():

- In python, the extend() function takes any iterable data type for e.g. list, tuple, string, a dictionary as an argument, and adds all the elements of the list passed as an argument to the end of the given list. This function is used to add more than one element in the list in a single statement.

- For example;

  >> list = [2, 4, 5, 6]

  >> list.extend ((2, 3, 4, 5))

  >> print(list)

  Output:

  [2, 4, 5, 6, 2, 3, 4, 5]

Qs 6. Consider a list:

list1 = [6,7,8,9]

What is the difference between the following operations on list1:

a) list1 * 2

b) list1 *= 2

c) list1 = list1 * 2

Ans.

(a) The output of the statement will be:

[6, 7, 8, 9, 6, 7, 8, 9]

It will replicate the list1 twice, but the list1 will remain the same.

(b) The output of the statement will be:

list1 = [6, 7, 8, 9, 6, 7, 8, 9]

It will replicate the list1 twice and store it in list1.

(c) The output of the statement will be:

The result will be same as 'list1 *= 2'. So,

list1 = [6, 7, 8, 9, 6, 7, 8, 9]

Qs 7. The record of a student (Name, Roll No, Marks in five subjects and percentage of marks) is stored in the following list:

stRecord = ['Raman','A-36',[56,98,99,72,69], 78.8]

Write Python statements to retrieve the following information from the list stRecord.

a) Percentage of the student

b) Marks in the fifth subject

c) Maximum marks of the student

d) Roll No. of the student

e) Change the name of the student from 'Raman' to 'Raghav'

Ans**.**

a) By using command stRecord[3] we can find the percentage.

stRecord[3] = 78.8

b) By using command stRecord[2][4] we can find the marks in 5th subject.

stRecord[2][4] = 69

c) By using command max(stRecord[2]) we can find the maximum marks of the student.

max(stRecord[2]) = 99

d) By using command stRecord[1] we can find the Roll no. of the student.

stRecord[1] = A-36.

e) By using command stRecord[0] = “Raghav”’ we can change the name from Raman to Raghav.

Qs 8. Consider the following dictionary stateCapital:

stateCapital = {"Assam":"Guwahati", "Bihar":"Patna","Maharashtra":"Mumbai", "Rajasthan":"Jaipur"}

Find the output of the following statements:

a) print(stateCapital.get("Bihar"))

b) print(stateCapital.keys())

c) print(stateCapital.values())

d) print(stateCapital.items())

e) print(len(stateCapital))

f) print("Maharashtra" in stateCapital)

g) print(stateCapital.get("Assam"))

h) del stateCapital["Assam"]

print(stateCapital)

Ans.

a) Output: Patna.

b) Output: dict_keys(['Assam', 'Bihar', 'Maharashtra', 'Rajasthan'])

c) Output: dict_values(['Guwahati', 'Patna', 'Mumbai', 'Jaipur'])

d) Output: dict_items([('Assam', 'Guwahati'), ('Bihar', 'Patna'), ('Maharashtra', 'Mumbai'), ('Rajasthan', 'Jaipur')])

e) Output: 4.

f) Output: True.

g) Output: Guwahati.

h) Output: {'Bihar': 'Patna', 'Maharashtra': 'Mumbai', 'Rajasthan': 'Jaipur'}

Programming Problems

1. Write a program to find the number of times an element occurs in the list.

```
val = eval(input("Enter a list : "))
item = int(input("Enter an element : "))
print(item, "occurs in the list",  val.count(item), "times")
```

```
Enter a list : [2,4,1,5,1]
Enter an element : 1
1 occurs in the list 2 times
```

2. Write a program to read a list of n integers (positive notes as well as negative). Create two new lists, one having all positive numbers and the other having all negative numbers from the given list. Print all three lists.

```
listval = eval(input("Enter a list : having positive and negative values "))
positivelist =[]
negativelist =[]
for n in listval:
    if n < 0:
        negativelist.append(n)
    else:
        positivelist.append(n)
print("Original List : ", listval)
```

```
print("Positive List : ", positivelist)
print("Negative List : ", negativelist)
```

```
Enter a list : having positive and negative values [-6,14,25,-36,47,-93]
Original List :  [-6, 14, 25, -36, 47, -93]
Positive List :  [14, 25, 47]
Negative List :  [-6, -36, -93]
```

3. Write a program to find the largest and the second largest elements in a given list of elements.

```
lst = eval(input("Enter a list : "))
lst.sort(reverse=True)
print("The largest number is : ",lst[0])
print("The second largest number is : ",lst[1])
```

Or

```
lst = eval(input("Enter a list : "))
max1=lst[0]
max2=lst[1]
for n in lst:
   if max1 < n:
      max2=max1
      max1=n
   elif max2 < n:
      max2=n
print("The largest number is : ",max1)
print("The second largest number is : ",max2)
```

```
Enter a list : [3,24,1,-6,90,56]
The largest number is :  90
The second largest number is :  56
```

4. Write a program to read a list of n integers and find their median.
Note: The median value of a list of values is the middle one when they are arranged in order. If there are two middle values then take their average.

```
lst = eval(input("Enter a list : "))
```

```
lst.sort()
length = len(lst)
mid =  length // 2
if length%2 !=0:
    median = lst[mid]
else:
    mid1 = mid-1
    mid2 = mid
    median = (lst[mid1]+lst[mid2]) // 2
print("The median value is : ", median)
```

```
Enter a list : [4,15,60,78,90,100]
The median value is :  69
```

5. Write a program to read a list of elements. Modify this list so that it does not contain any duplicate elements i.e. all elements occurring multiple times in the list should appear only once.

```
lst = eval(input("Enter a list : "))
newlist = []
for num in lst:
    if num not in newlist:
        newlist.append(num)
lst = newlist
print("List without diplicate elements : ",lst)
```

```
Enter a list : [45,78,12,45,67,78,45]
List without diplicate elements :  [45, 78, 12, 67]
```

6. Write a program to create a list of elements. Input an element from the user that has to be inserted in the list. Also input the position at which it is to be inserted.

```
lst = eval(input("Enter a list : "))
new_element = int(input("Enter the element to be inserted : "))
position = int(input("Enter the valid index position, where the element to be
inserted : "))
lst.insert(position,new_element)
print("List after insertion : ",lst)
```

```
Enter a list : [56,34,23,78,33]
Enter the element to be inserted : 22
Enter the valid index position, where the element to be inserted : 2
List after insertion :  [56, 34, 22, 23, 78, 33]
```

7. Write a program to read elements of a list and do the following.

a) The program should ask for the position of the element to be deleted from the list and delete the element at the desired position in the list.

b) The program should ask for the value of the element to be deleted from the list and delete this value from the list.

```
lst = eval(input("Enter a list : "))
position = int(input("Enter the valid index position, to delete the element : "))
lst.pop(position)
print("List after deletion : ",lst)
value = int(input("Enter the value to be deleted : "))
lst.remove(value)
print("List after removing element : ",lst)
```

```
Enter a list : [78,45,22,12,566,8]
Enter the valid index position, to delete the element : 3
List after deletion :  [78, 45, 22, 566, 8]
Enter the value to be deleted : 22
List after removing element :  [78, 45, 566, 8]
```

8. Write a Python program to find the highest 2 values in a dictionary.

```
numdict = {"First":555, "Second":23,"Third":254,"Fourth":658}
val = sorted(sorted(numdict.values()),reverse=True)
print("Highest two values are : ", val[0],val[1])
```

```
Highest two values are :  658 555
```

9. Write a Python program to create a dictionary from a string ‘w3resource’ such that each individual character mates a key and its index value for fist occurrence males the corresponding value in dictionary.

Expected output : {'3': 1, 's': 4, 'r': 2, 'u': 6, 'w': 0, 'c': 8, 'e': 3, 'o': 5}

```
str = "w3resource"
dict={}
index=0
for chr in str:
   if chr not in dict:
      dict[chr]=index
   index=index+1
print("Dictionary : ",dict)
```

```
Dictionary :  {'w': 0, '3': 1, 'r': 2, 'e': 3, 's': 4, 'o': 5, 'u': 6, 'c': 8}
```

10. Write a program to input your friend's, names and their phone numbers and store them in the dictionary as the key-value pair. Perform the following operations on the dictionary:

a) Display the Name and Phone number for all your friends.

b) Add a new key-value pair in this dictionary and display the modified dictionary

c) Delete a particular friend from the dictionary

d) Modify the phone number of an existing friend

e) Check if a friend is present in the dictionary or not

f) Display the dictionary in sorted order of names

```
phonedict = {}
n = int(input("How many friend's details want to add : "))

for t in range(n):
   name = input("Enter Friend's name: ")
   phone = input("Enter Phone Number: ")
   if name not in phonedict:
      phonedict[name] = phone

print("Friends dictionary is : ",phonedict)

while True:
      print("1. Display all friends")
      print("2. Add friend")
      print("3. Delete a friend")
      print("4. Modify phone number")
      print("5. Search a friend")
```

```
print("6. Sort on Names")
print("7. Exit")
choice = int(input("Enter your choice 1 - 7 : "))

if choice == 1:
    print("Name \t Phone number")
    for key in phonedict:
        print(key,phonedict[key])

elif choice == 2:
    name = input("Enter friend's name : ")
    phone = input("Enter phone number : ")
    if name not in phonedict:
        phonedict[name] = phone
    else:
        print("Name already exists.")

elif choice == 3:
    name = input("Enter friend's name to delete : ")
    if name in phonedict:
        del phonedict[name]
    else:
        print("No such name exists.")

elif choice == 4:
    name = input("Enter friend's name whose number to modify : ")
    if name in phonedict:
        phone = input("Enter new phone number : ")
        phonedict[name] = phone
    else:
        print("No such name exists.")

elif choice == 5:
    name = input("Enter friend's name to search : ")
    if name in phonedict:
        print(name, "\t", phonedict[name])
    else:
        print("No such name exists.")
```

```
elif choice == 6:
    keys = sorted(phonedict)
    print("Sorted phone directory is : ")
    for key in keys:
        print(key, "\t", phonedict[key])

elif choice == 7:
    break

else:
    print("Invalid input")
```

```
How many friend's details want to add : 5
Enter Friend's name: Arun
Enter Phone Number: 1234
Enter Friend's name: Avi
Enter Phone Number: 2345
Enter Friend's name: Rima
Enter Phone Number: 3456
Enter Friend's name: Robin
Enter Phone Number: 4567
Enter Friend's name: Vishal
Enter Phone Number: 5678
Friends dictionary is :  {'Arun': '1234', 'Avi': '2345', 'Rima': '3456', 'Robin': '4567', 'Vishal': '5678'}
1. Display all friends
2. Add friend
3. Delete a friend
4. Modify phone number
5. Search a friend
6. Sort on Names
7. Exit
```

```
Enter your choice 1 - 7 : 1
Name     Phonbe number
Arun 1234
Avi 2345
Rima 3456
Robin 4567
Vishal 5678
1. Display all friends
2. Add friend
3. Delete a friend
4. Modify phone number
5. Search a friend
6. Sort on Names
7. Exit
Enter your choice 1 - 7 : 2
Enter friend's name : David
Enter phone number : 6789
1. Display all friends
2. Add friend
3. Delete a friend
4. Modify phone number
5. Search a friend
6. Sort on Names
7. Exit
Enter your choice 1 - 7 : 1
Name     Phonbe number
Arun 1234
Avi 2345
Rima 3456
Robin 4567
Vishal 5678
David 6789
1. Display all friends
2. Add friend
3. Delete a friend
4. Modify phone number
5. Search a friend
6. Sort on Names
7. Exit
```

```
Enter your choice 1 - 7 : 3
Enter friend's name to delete : Rima
1. Display all friends
2. Add friend
3. Delete a friend
4. Modify phone number
5. Search a friend
6. Sort on Names
7. Exit
Enter your choice 1 - 7 : 1
Name     Phonbe number
Arun 1234
Avi 2345
Robin 4567
Vishal 5678
David 6789
1. Display all friends
2. Add friend
3. Delete a friend
4. Modify phone number
5. Search a friend
6. Sort on Names
7. Exit
Enter your choice 1 - 7 : 4
Enter friend's name whose number to modify : Avi
Enter new phone number : 1111
1. Display all friends
2. Add friend
3. Delete a friend
4. Modify phone number
5. Search a friend
6. Sort on Names
7. Exit
Enter your choice 1 - 7 : 5
Enter friend's name to search : Arun
Arun     1234
1. Display all friends
2. Add friend
3. Delete a friend
4. Modify phone number
5. Search a friend
6. Sort on Names
7. Exit
```

```
Enter your choice 1 - 7 : 6
Sorted phone directory is :
Arun     1234
Avi      2345
David    6789
Robin    4567
Vishal   5678
1. Display all friends
2. Add friend
3. Delete a friend
4. Modify phone number
5. Search a friend
6. Sort on Names
7. Exit
Enter your choice 1 - 7 : 3
Enter friend's name to delete : fdgdf
No such name exists.
1. Display all friends
2. Add friend
3. Delete a friend
4. Modify phone number
5. Search a friend
6. Sort on Names
7. Exit
Enter your choice 1 - 7 : 7
```

Q. Explain the different types of Token in python
Ans. **Tokens** are the smallest unit of the program.

1. keywords

keywords are nothing but a set of special words, which are reserved by Python and have specific meaning.

2. identifier

identifier in Python are nothing but user defined name to represent programmable entity like variable, function, classes, modules or any object.

3. literals

the other built in object in Python are literal. literal can be define as data that is given in a variable or constant. Python has following literal:-

(a) string

(b) numeric

(c) boolean

(d) collection

4. operators

operators are the symbols which performs the operation on some values. these values are known as operand.

5. Punctuators :

Punctuators are tokens in python employed to put the grammar and structure of syntax into practice.Symbols like '#', '(', '[', '=' etc.

DEBUGGING:

Due to errors, a program may not execute or may generate wrong output.

1. Syntax errors

2.Logical errors

3.Runtime errors

1.Syntax errors: Syntax errors are produced by Python when it is translating the source code into byte code. They usually indicate that there is something wrong with the syntax of the program.

Examples of syntax errors are missing a comma or a quotation mark, or misspelling a word.

2.Logical errors: They occur when the program runs without crashing, but produces an incorrect result. The error is caused by a mistake in the program's logic. You won't get an error message, because no syntax or runtime error has occurred.

For example if we wish to find the average of two numbers 10 and 12 and we write the code as 10+12/2. Run successfully but result is wrong 16 .correct result is 11. [(10+12)/2].

3.Runtime errors: If a program is free of syntax errors, it will be run by the Python interpreter. However, the program may exit if it encounters a runtime error – a problem that went undetected when the program was parsed, but is only revealed when the code is executed. Some examples of Python Runtime errors.

For example we have statement having division operation in the program. By mistake ,if the denominator value is zero then it will give a runtime error like ‘’ division by zero’’

➔ OPERATORS WITH EXAMPLES

1) Python (+) Addition operator

```
X = 10
Y = 20
  # output: x + y = 30
  print ( ‘x + y =’, x + y )
```

2) Python (-) Subtraction operator

```
x = 10
y = 20
# output: x - y = - 10
print (‘x - y =’, x - y )
```

3) Python (*) Multiplication operator

```
x = 10
 y = 20
 # output: x * y = 200
 print (‘x * y =’, x * y )
```

4) Python (/) Division operator

```
  x = 10
 y = 20
 # output: x / y = 0.5
 print (‘x / y =’, x / y )
```

5) Python (%) Percentail operator

```
x = 20
 y = 10
 # output: x % y = 0.
 print (‘x % y =’, x % y )
```

6)Python (//) Floor division operator

```
x = 50
y = 20
# Output: x // y = 2
print('x // y =',x//y)
```

Note:(In the above program, actual value that we get is 2.5, but to match the requirements of the Floor operator, we had to adjust it to the left side, i,e, 2.5 to 2)

7) Python (**) Exponent operator

```
x = 10
y = 4
# output: x ** y =10,000
print (‘x ** y =’, x ** y )
```

Python Comparison Operators

1) Python (==) Equal Comparison Operator

```
x = 20
y = 25
# Output: x == y is False
print('x == y is',x==y)
```

2) Python (!=) Not Equal Comparison Operator

```
x = 20
 y = 25
# Output: x != y is True
print('x != y is',x!=y)
```

Note:[In the above program, the two values are not equal to one another, the Not equal operator is exclusively used to return the True statement if the values are not equal].

3) Python (>) Greater Than Operator

```
x = 20
y = 25
# Output: x > y is False
print('x > y  is',x>y)
```

4) Python (<) Less Than Operator

Ex: 10 is < 12, then the answer becomes True

10 is < 8, then the answer becomes False

Program:

```
x = 20
y = 25
# Output: x < y is True
print('x < y  is',x<y)
```

5) Python >= Greater Than or Equal to operator

```
x = 20
y = 25
# Output: x >= y is False
print('x >= y is',x>=y)
```

6) Python <= Less than or equal to operator

```
x = 20
y = 25
# Output: x <= y is True
print('x <= y is',x<=y)
```

Python Assignment Operators

1) Python (=) Assign value Operator

Example: 5 = 2+3

2) Python (+=) Add AND Operator

It adds the value of right operand to the value of left hand operand.

example; x = 10, x + = 5, which will return the value as 15 by adding the right side value to the left hand one. Therefore, the answer is x= 15.

3) Python(-=) Subtract AND Operator

```
x = 10, y = 7, the  x-=y answer is x = 3
```

4) Python (*=) Multiply AND Operator

```
x = 10, y=5, the x*=y answer is x = 50.
```

5) Python (/=) Divide AND Operator

```
x = 10, x/= 5 the result is x = 2
```

6) Python (%=) Modulus AND Operator

```
x = 10, x%= 5 the result is x = 0
```

7) Python **= Exponent AND operator

```
x = 10, x**= 5 the result is x = 1,00,000
```

8) Python (//=) Floor Division

```
x = 7, x//= 4 the result is x = 1
```

Python logical Operators

Logical operators are used in any programming language to make decision based on multiple conditions. In python, we use Logical operators to determine whether a condition is True or False by taking Operand values as base.

1) Python and Logical AND operator

AND operator returns the logical value 'True' when two Operands are true; otherwise, it will return False.

```
x = 10
y = 10
# Output: x and y is True
print('x and y is',x and y)
```

2) Python or Logical OR operator

In this Logical OR operator, if any one of the two operands are non zero, then it returns the True condition. .

Ex: variable A = 10, B = 20,

Program

```
x = 10
y = 20
# Output: x or y is True
print('x or y is',x or y)
```

3) Python not Logical NOT

```
x = 10
y = 20
# Output: not x is False
print('not x is',not x)
```

Python Bitwise operators

Bitwise operators are used in python to perform the operations on binary numerals or bit patterns.

1) Python & Binary AND Bitwise Operator

Description: This operator performs bit by bit AND operations on the values on either side of the operator.

Example Program

```
a = 10
b = 4
# Print bitwise AND operation
print(a & b)
```

Output: 0

2) Python | Binary OR Bitwise Operator

```
a = 10
b = 4
# Print bitwise OR operation
print(a | b)
```

Output: 14

3) Python ^ Binary XOR Bitwise Operator

```
a = 10
b = 4
# print bitwise XOR operation
print(a ^ b)
```

Output: 14

4) Python ~ Binary Ones Complement Bitwise Operator

```
a = 10
b = 4
# Print bitwise NOT operation
print(~a)
```

Output: -11

5) Python << Binary Left shift Bitwise Operator

```
a = 10
b = 4
# print bitwise left shift operation
print(a << 2)
```

Output: 40

6) Python >> Binary Right shift Bitwise Operator

```
a = 10
b = 4
# print bitwise right shift operation
print(a >> 2)
```

Output: 2

Python Membership operators

Membership Operators are mainly used in python to find whether a value or variable is present in a sequence (string, tuple, list, set, and directory). “In” and “Not In” are the two membership operators available in python.

1) Python In Membership Operator

This In operator is used in python to evaluate if a particular value is there or not in a sequence. The In operator returns True if it the specified element found in the list, otherwise it returns false.

Let’s consider a python program on how In operator works.

Program

```
a  = 10
b  =  7
list = [1,2,3,4,5,6,7,8,9..)

If (a is in list):
   print: line - 1 “a is presented in the list”
else
    print: line - 1 “a is not presented in the list”
If (b is in list):
   print: line - 2 “b is presented in the list”
else
    print: line - 2 “b is not presented in the list”
```

When you are done with the execution of above program, you will get the below output:

```
Line - 1- a is not available in the list. Hence False.
Line - 2- b is  available in the list. Hence True
```

2) python Not in Membership Operator

"Not in" operator, the name itself is expressing the meaning that something is not inside. It goes through a particular sequence to find weather a value is in the specified list or not. If it finds there is no such value, it returns True, otherwise False.

Let's consider a python program on how Not In operator works:

Program

```
# Not 'in' Operator
x = 23
y = 20
list = [10, 15,20,  30, 35, 40,45, 50];
if (x is not in list)
   print: " x is not available in the list"
else
   print: " x is available in the  given list"
if (y is not in list)
   print: " y is not available in the  given list"
else
   print: " y is available in the given list"
```

Run:

```
Line 1 - x is not there in given list. Hence True.
Line 2 - y is there in given list. Hence False.
```

Python Identity operators

We have two identity operators in python namely is and is not. These are also called as special operators and are used to find if two identical values (or variables) located on the same part of the memory or not. If two variables are equal, that does not mean they are identical. We have two different operators.

1) Python is Identity operator

This operator returns true statement if two identical operands reside in the same memory, otherwise false.

Let's look at the small program on how the '***is***' is used in python.

example

```
x = 6
if (type(x) is int):
    print ("true")
else:
    print ("false")
```

Output

```
True
```

2) Python Is not Identity operator

It will return false if either side of the operator pointing towards the same thing, otherwise it returns true.

Let's look at the program to know how Is not operator will work;

```
x = 4.5
If (type (x) is not int) :
Print (“true”)
else:
    print ("false")
```

Output

```
True
```

DATABASE MANAGEMENT SYSTEM

MYSQL

It is freely available open source Relational Database Management System (RDBMS) that uses **Structured Query Language(SQL).** In MySQL database , information is stored in Tables. A single MySQL database can contain many tables at once and store thousands of individual records.

SQL (Structured Query Language)

SQL is a language that enables you to create and operate on relational databases, which are sets of relatedinformation stored in tables.

DIFFERENT DATA MODELS

A **data model refers to a set of concepts to describe the structure of a database**, and certain constraints (restrictions)
that the database should obey.

Different Data Models:

1. Relational data model
2. Network data model
3. Hierarchical data model

1. **Relational data model:** In this model data is organized into tabular structures called relations. A database may contain many relations providing a better classification of data based on its nature and use. Multiple relations are then linked/ associated together on some common key data values (foreign key).
2. **Network Data Model:** In this model data is represented by collections of records and relationships among data are represented by links. A record is collection of fields i.e. attributes, each of which contents only one data value.
3. **Hierarchical data model:** In this model records are organized as trees, data is represented by collection of records connected to one another through links.

4. RELATIONAL MODEL TERMINOLOGY

•**Relation :** A table storing logically related data is called a Relation.

•**Tuple :** A **row of a relation** is generally referred to as a tuple.

•**Attribute :** A **column** of a relation is generally referred to as an attribute.

•**Degree :** This refers to the **number of attributes** in a relation.

•**Cardinality :** This refers to the **number of tuples** in a relation.

•**Primary Key :** This refers to a set of one or more attributes that can uniquely identify tuples within the relation.

- **Candidate Key :** All attribute combinations inside a relation that can serve as primary key are candidate keys asthese are candidates for primary key position.

•**Alternate Key :** A candidate key that is not primary key, is called an alternate key.

- **Foreign Key :** A non-key attribute, whose values are derived from the primary key of some other table, isknown as foreign key in its current table.

REFERENTIAL INTEGRITY

- A referential integrity is a system of rules that a DBMS uses to ensure that relationships between records in related tables are valid, and that users don't accidentally delete or change related data. This integrity is ensured by foreign key.

CLASSIFICATION OF SQL STATEMENTS

SQL commands can be mainly divided into following categories:

1. **Data Definition Language(DDL) Commands**
   Commands that allow you to perform task, related to data definition e.g;

- Creating, altering and dropping.
- Granting and revoking privileges and roles.
- Maintenance commands.

- **CREATE DATABASE** - creates a new database
- **ALTER DATABASE** - modifies a database
- **CREATE TABLE** - creates a new table
- **ALTER TABLE** - modifies a table
- **DROP TABLE** - deletes a table

2. **Data Manipulation Language(DML) Commands**
   Commands that allow you to perform data manipulation e.g., retrieval, insertion, deletion and modificationof data stored in a database.

- **SELECT** - extracts data from a database
- **UPDATE** - updates data in a database
- **DELETE** - deletes data from a database
- **INSERT INTO** - inserts new data into a database

3. **Transaction Control Language(TCL) Commands**
   Commands that allow you to manage and control the transactions e.g.,

- Making changes to database, permanent
- Undoing changes to database, permanent
- Creating save point
- Setting properties for current transactions.

- **COMMIT** – it make all the changes made by statement issued.
- **ROLLBACK** – it undoes all changes since the beginning of the transaction or since save point.
- **SAVEPOINT** – it marks a point up to successfully completed transaction.
- **SET TRANSACTION** – it establish properties for the current transaction.

MySQL ELEMENTS

1. Literals 2. Datatypes 3. Nulls 4. Comments

LITERALS

It refer to a fixed data value. This fixed data value may be of character type or numeric type. For example,'replay' , 'Raj', '8' , '306' are all character literals.

Numbers not enclosed in quotation marks are numeric literals.

E.g. 22 , 18 , 1997 are all numeric literals.
E.g. 17 is an integer literal but 17.0 and 17.5 are real literals.

DATA TYPES

Data types are means to identify the type of data and associated operations for handling it.
MySQL datatypes are divided into three categories:

- Numeric
- Date and Time
- String types

Numeric Data Type

1. int – used for number without decimal.
2. Decimal(m,d) – used for floating/real numbers. m denotes the total length of number and d is number of decimal digits.

Date and Time Data Type

1. date – used to store date in YYYY-MM-DD format.
2. time – used to store time in HH:MM:SS format.

String Data Types

1. char(m) – used to store a fixed length string. **m** denotes max. number of characters.
2. varchar(m) – used to store a variable length string. **m** denotes max. no. of characters.

DIFFERENCE BETWEEN CHAR AND VARCHAR DATA TYPE

S.NO.	Char Datatype	Varchar Datatype
	It specifies a **fixed length** character String.	It specifies a **variable length** character string.
	When a column is given datatype as CHAR(**n**), then MySQL ensures that all values stored in that column have this length i.e. **n** bytes. If a value is shorter than this length **n** then blanks are added, but the size of value remains **n** bytes.	When a column is given datatype as VARCHAR(**n**), then the maximum size a value in this column can have is **n** bytes. Each value that is stored in this column store exactly as you specify it i.e. no blanks are added if the length is shorter than maximum length **n**.

DATABASE COMMANDS

1. VIEW EXISTING DATABASE

To view existing database names, the command is : **SHOW DATABASES ;**

2. CREATING DATABASE IN MYSQL

For creating the database in MySQL, we write the following command: **CREATE DATABASE** <databasename> ;

e.g. In order to create a database Student, command is :

CREATE DATABASE Student ;

3. ACCESSING DATABASE

For accessing already existing database , we write :

USE <databasename> ;

e.g. to access a database named Student , we write command as :

USE Student ;

4. DELETING DATABASE

For deleting any existing database , the command is :

DROP DATABASE <databasename> ;

e.g. to delete a database , say student, we write command as ; **DROP DATABASE** Student ;

5. VIEWING TABLE IN DATABASE

In order to view tables present in currently accessed database , command is : **SHOW TABLES ;**

CREATING TABLES IN MYSQL

- Tables are created with the CREATE TABLE command. When a table is created, its columns are named, datatypes and sizes are supplied for each column.

Syntax of CREATE TABLE commandis :

CREATE TABLE <table-name>

(<column name> <data type> ,

<column name> <data type> ,

…… **) ;**

E.g. in order to create table STUDENT given below :

ROLLNO	SNAME	DOB	PNAME	ADDRESS

We write the following command :

mysql> CREATE TABLE STUDENT(

ROLLNO INT,

SNAME VARCHAR(20),

DOB DATE,

PNAME VARCHAR(20),
ADDRESS VARCHAR(50),
PRIMARY KEY(ROLLNO));

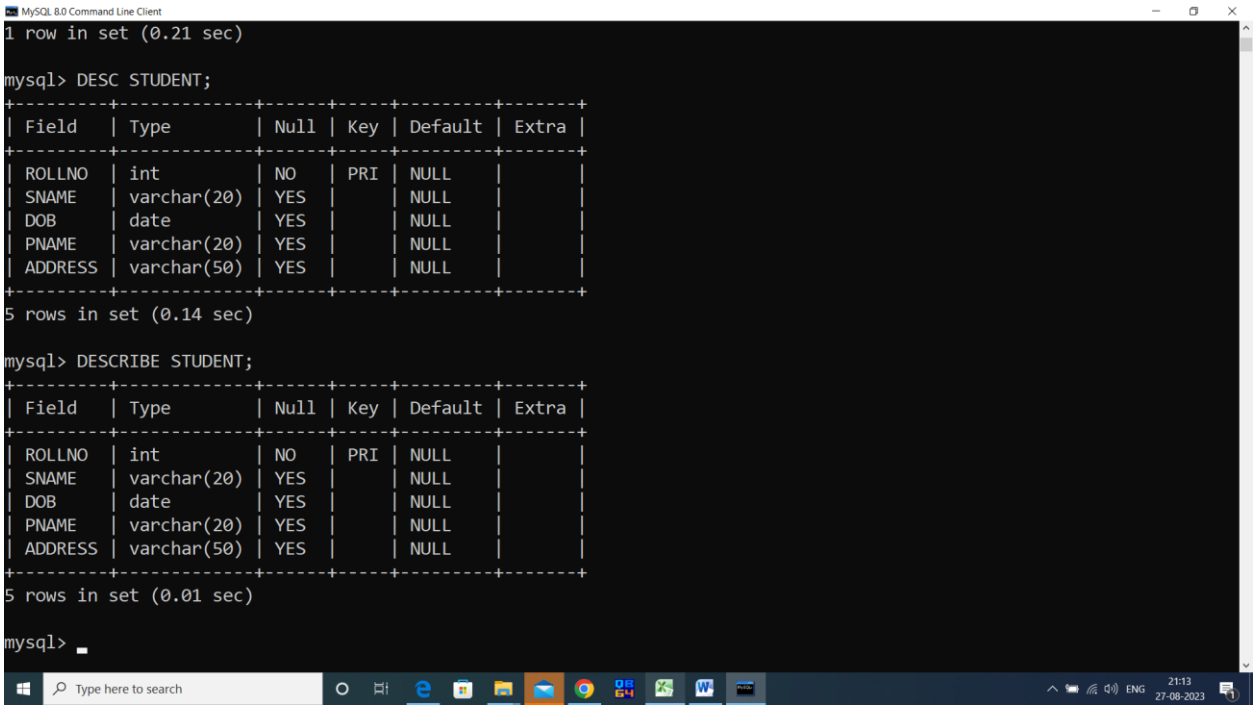
E.g. in order to create table EMPLOYEE given below :

ECODE	ENAME	GENDER	GRADE	GROSS

We write the following command :
mysql> CREATE TABLE employee
(ECODE integer ,
ENAME varchar(20) ,
GENDER char(1) ,
GRADE char(2) ,
GROSS integer) ;

DESCRIBE TABLE

We can view the structure of an already created table using the describe statement.
Syntax: DESCRIBE <TABLENAME>;
OR
mysql> DESC <TABLENAME>;



Show Tables

To display all tables in a database.

```
mysql> SHOW TABLES;
+--------------------+
| Tables_in_student1 |
+--------------------+
| employee           |
| student            |
+--------------------+
2 rows in set (0.00 sec)
```

DROP Statement

- Syntax to drop a table:

DROP TABLE table_name;

- Syntax to drop a database:

DROP DATABASE database_name;

Note: 1) Using the Drop statement to remove a database will ultimately remove all the tables within it.
2) DROP statement will remove the tables or database created by you.

DML(DATA MANIPULATION LANGUAGE)

INSERTING DATA INTO TABLE :

The rows are added to relations(table) using INSERT command of SQL.

Syntax of INSERT is :

INSERT INTO <tablename> (<column list>)
VALUES (<value1> , <value2> , ...) ;

e.g. to enter a row into employee table (created above), we write command as :

mysql> INSERT INTO employee
VALUES(1001 , 'Ravi' , 'M' , 'E4' , 50000);

OR

EX :

mysql> INSERT INTO employee (ECODE , ENAME , GENDER , GRADE , GROSS)
VALUES(1001 , 'Ravi' , 'M' , 'E4' , 50000);

ECODE	ENAME	GENDER	GRADE	GROSS
1001	Ravi	M	E4	50000

In order to insert another row in EMPLOYEE table , we write again INSERT command :

EX:

mysql> INSERT INTO employee
VALUES(1002 , 'Akash' , 'M' , 'A1' , 35000);

ECODE	ENAME	GENDER	GRADE	GROSS
1001	Ravi	M	E4	50000
1002	Akash	M	A1	35000

INSERTING NULL VALUES

- To insert value NULL in a specific column, we can type NULL without quotes and NULL will be inserted in thatcolumn. E.g. in order to insert NULL value in ENAME column of above table, we write INSERT command as :

EX : **mysql>** INSERT INTO EMPLOYEE
VALUES (1004 , NULL , 'M' , 'B2' , 38965) ;

ECODE	ENAME	GENDER	GRADE	GROSS
1001	Ravi	M	E4	50000
1002	Akash	M	A1	35000
1004	NULL	M	B2	38965

SIMPLE QUERY USING SELECT COMMAND

The SELECT command is used to pull information from a table.

Syntax of SELECTcommand is :

SELECT <column name1>,<column name2>
FROM <tablename>
WHERE<condition name> ;

SELECTING ALL DATA

In order to retrieve everything (all columns) from a table, SELECT command is usedas :

mysql> SELECT * FROM <tablename> ;

e.g.: **SELECT * FROM EMPLOYEE;**

In order to retrieve everything from **Employee** table, we write SELECT command as :

EMPLOYEE

ECODE	ENAME	GENDER	GRADE	GROSS
1001	Ravi	M	E4	50000
1002	Akash	M	A1	35000
1004	NULL	M	B2	38965

EX : **mysql> SELECT * FROM** Employee ;

EMPLOYEE

ECODE	ENAME	GENDER	GRADE	GROSS
1001	Ravi	M	E4	50000
1002	Akash	M	A1	35000
1004	Neela	F	B2	38965
1005	Sunny	M	A2	30000
1006	Ruby	F	A1	45000
1009	Neema	F	A2	52000

- A particular column from a table can be selected by specifying column-names with SELECT command. E.g. inabove table, if we want to select ECODE and ENAME column, then command is :

mysql> SELECT ECODE , ENAME
FROM EMPLOYEE ;

E.g.2 in order to select only ENAME, GRADE and GROSS column, the command is :

mysql> SELECT ENAME , GRADE , GROSS
FROM EMPLOYEE ;

SELECTING PARTICULAR ROWS

We can select particular rows from a table by specifying a condition through **WHERE clause** along with SELECTstatement.

E.g. In employee table if we want to select rows where Gender is female, then command is :

mysql> SELECT * FROM **EMPLOYEE**
WHERE **GENDER = 'F' ;**

E.g.2. I In order to select rows where salary is greater than 48000, then command is :

mysql> SELECT * FROM **EMPLOYEE**
WHERE **GROSS > 48000 ;**

ELIMINATING REDUNDANT DATA

The **DISTINCT** keyword eliminates duplicate rows from the results of a SELECT statement.
For example :
SELECT GENDER **FROM** EMPLOYEE ;

GENDER
M
M
F
M
F
F

MySql> SELECT DISTINCT(GENDER) **FROM** EMPLOYEE ;

DISTINCT(GENDER)
M
F

VIEWING STRUCTURE OF A TABLE

- If we want to know the structure of a table, we can use DESCRIBE or DESC command, as per following syntax :

DESCRIBE | **DESC** <tablename> ;

e.g. to view the structure of table **EMPLOYEE**,
command is :

DESCRIBE EMPLOYEE ;
OR
DESC EMPLOYEE ;

USING COLUMN ALIASES

- The columns that we select in a query can be given a different name, i.e. column alias name for output purpose.

Syntax :

SELECT <columnname> **AS** column alias , <columnname> **AS** column alias …..FROM <tablename> ;

e.g. In output, suppose we want to display ECODE column as EMPLOYEE_CODE in output , then command is :

mysql> SELECT **ECODE AS "EMPLOYEE_CODE"**
FROM **EMPLOYEE ;**

CONDITION BASED ON A RANGE

- The **BETWEEN** operator defines a range of values that the column values must fall in to make the conditiontrue. The range include both lower value and upper value.

e.g. to display ECODE, ENAME and GRADE of those employees whose salary is between 40000 and 50000, command is:

mysql> SELECT ECODE , ENAME,GRADE
FROM **EMPLOYEE**
WHERE GROSS **BETWEEN** 40000 AND 50000 ;

Output will be :

ECODE	ENAME	GRADE
1001	Ravi	E4
1006	Ruby	A1

CONDITION BASED ON A LIST

- To specify a list of values, IN operator is used. The IN operator selects value that match any value in a givenlist of values. E.g.

mysql> SELECT * FROM EMPLOYEE WHERE GRADE **IN** ('A1' , 'A2');
Output will be :

ECODE	ENAME	GENDER	GRADE	GROSS
1002	Akash	M	A1	35000
1006	Ruby	F	A1	45000
1005	Sunny	M	A2	30000
1009	Neema	F	A2	52000

- The **NOT IN** operator finds rows that do not match in the list. E.g.
-

mysql> SELECT * FROM EMPLOYEE
WHERE GRADE **NOT IN** ('A1' , 'A2');

Output will be :

ECODE	ENAME	GENDER	GRADE	GROSS
1001	Ravi	M	E4	50000
1004	Neela	F	B2	38965

CONDITION BASED ON PATTERN MATCHES

- LIKE operator is used for pattern matching in SQL. Patterns are described using two special wildcard characters:

1. percent(%) – The % character matches any substring.
2. underscore(_) – The _ character matches any character.

e.g. to display names of employee whose name starts with R in EMPLOYEE table, the command is :

mysql> SELECT ENAME FROM
EMPLOYEE
WHERE ENAME **LIKE 'R%' ;**

Output will be :

ENAME
Ravi
Ruby

e.g. to display details of employee whose second character in name is 'e'.

mysql> SELECT *FROM **EMPLOYEE**
WHERE **ENAME LIKE '_e%' ;**

Output will be :

ECODE	ENAME	GENDER	GRADE	GROSS
1004	Neela	F	B2	38965
1009	Neema	F	A2	52000

e.g. to display details of employee whose name ends with 'y'.

mysql> SELECT *FROM **EMPLOYEE**
WHERE **ENAME LIKE '%y' ;**

Output will be :

ECODE	ENAME	GENDER	GRADE	GROSS
1005	Sunny	M	A2	30000
1006	Ruby	F	A1	45000

SEARCHING FOR NULL

- The NULL value in a column can be searched for in a table using IS NULL in the WHERE clause.
- E.g. to listemployee details whose salary contain NULL, we use the command :

STUDENT

Roll_No	Name	Marks
1	ARUN	NULL
2	RAVI	56
4	SANJAY	NULL

to display the names of those students whose marks is NULL, we use the command :

SELECT **Name**
FROM **STUDENT**
WHERE Marks **IS NULL ;**

Output will be :

Name
ARUN
SANJAY

SORTING RESULTS

Whenever the SELECT query is executed, the resulting rows appear in a predecided order. The **ORDER BY clause** allowsorting of query result. The sorting can be done either in ascending or descending order, the default is ascending.

The **ORDER BY clause is used as :**

SELECT <column name> , <column name>....FROM
<tablename>
WHERE <condition>
ORDER BY <column name> ;

to display the details of employees in EMPLOYEE table in alphabetical order, we use command :

SELECT * FROM EMPLOYEE ORDER BY ENAME ;

Output will be :

ECODE	ENAME	GENDER	GRADE	GROSS
1002	Akash	M	A1	35000
1004	Neela	F	B2	38965
1009	Neema	F	A2	52000
1001	Ravi	M	E4	50000
1006	Ruby	F	A1	45000
1005	Sunny	M	A2	30000

e.g. display list of employee in descending alphabetical order whose salary is greater than 40000.

SELECT ENAME
FROM EMPLOYEE
WHERE GROSS > 40000
ORDER BY ENAME desc ;

Output will be :

ENAME
Ruby
Ravi
Neema

MODIFYING DATA IN TABLES

you can modify data in tables using UPDATE command of SQL. The UPDATE command specifies the rows to bechanged using the WHERE clause, and the new data using the SET keyword.

Syntax of update command is :

UPDATE <tablename>
SET <columnname>=value , <columnname>=valueWHERE
<condition> ;

e.g. to change the salary of employee of those in EMPLOYEE table having employee code 1009 to 55000.

UPDATE EMPLOYEE
SET GROSS = 55000
WHERE ECODE = 1009 ;

UPDATING MORE THAN ONE COLUMNS

e.g. to update the salary to 58000 and grade to B2 for those employee whose employee code is 1001.

UPDATE EMPLOYEE
SET GROSS = 58000, GRADE='B2'
WHERE ECODE = 1001 ;

OTHER EXAMPLES

e.g.1. Increase the salary of each employee by 1000 in the EMPLOYEE table.

UPDATE EMPLOYEE
SET GROSS = GROSS +1000 ;

e.g.2. Double the salary of employees having grade as 'A1' or 'A2' .

UPDATE EMPLOYEE
SET GROSS = GROSS * 2
WHERE GRADE='A1' **OR** GRADE='A2' ;

e.g.3. Change the grade to 'A2' for those employees whose employee code is 1004 and name is Neela.

UPDATE EMPLOYEE
SET GRADE='A2'
WHERE ECODE=1004 **AND** ENAME='NEELA' ;

DELETING DATA FROM TABLES

To delete some data from tables, DELETE command is used.

The DELETE command removes rows from atable.

The syntax of DELETE command is :

DELETE FROM <tablename>
WHERE <condition> ;

For example, to remove the details of those employee from EMPLOYEE table whose grade is A1.

DELETE FROM EMPLOYEE
WHERE GRADE ='A1' ;

TO DELETE ALL THE CONTENTS FROM A TABLE

DELETE FROM EMPLOYEE ;

So if we do not specify any condition with WHERE clause, then all the rows of the table will be deleted.

DROPPING TABLES

The DROP TABLE command lets you drop a table from the database.

The **syntax of DROP TABLE** command is :

DROP TABLE <tablename> ;

e.g. to drop a table employee, we need to write :

DROP TABLE employee ;

Once this command is given, the table name is no longer recognized and no more commands can be given on that table.

After this command is executed, all the data in the table along with table structure will be deleted.

S.NO.	DELETE COMMAND	DROP TABLE COMMAND
1	It is a DML command.	It is a DDL Command.
2	This command is used to delete only rows of data from a table	This command is used to delete all the data of the table along with the structure of the table. The table is no longer recognized when this command gets executed.
3	Syntax of DELETE command is: **DELETE FROM** <tablename> **WHERE** <condition> ;	Syntax of DROP command is : **DROP TABLE** <tablename> ;

ALTER TABLE COMMAND

The ALTER TABLE command is used to change definitions of existing tables.

(adding columns,deleting columnsetc.).

The ALTER TABLE command is used for :

1. adding columns to a table
2. Modifying column-definitions of a table.
3. Deleting columns of a table.
4. Adding constraints to table.
5. Enabling/Disabling constraints.

ADDING COLUMNS TO TABLE

To add a column to a table, ALTER TABLE command can be used as per following syntax:

ALTER TABLE <tablename>
ADD <Column name> <datatype> <constraint> ;

e.g. to add a new column ADDRESS to the EMPLOYEE table, we can write command as :

ALTER TABLE EMPLOYEE
ADD ADDRESS VARCHAR(50) NOT NULL;

A new column by the name ADDRESS will be added to the table, where each row will contain NULLvalue for the new column.

ECODE	ENAME	GENDER	GRADE	GROSS	ADDRESS
1001	Ravi	M	E4	50000	NULL
1002	Akash	M	A1	35000	NULL
1004	Neela	F	B2	38965	NULL
1005	Sunny	M	A2	30000	NULL
1006	Ruby	F	A1	45000	NULL
1009	Neema	F	A2	52000	NULL

However **if you specify NOT NULL constraint while adding a new column**, MySQL adds the new column with thedefault value of that datatype e.g. for INT type it will add 0 , for CHAR types, it will add a space, and so on.

e.g. Given a table namely Testt with the following data in it.

Col1	Col2
1	A
2	G

Now following commands are given for the table. Predict the table contents after each of the following statements:

(i) ALTER TABLE testt ADD col3 INT ;

(ii) ALTER TABLE testt ADD col4 INT NOT NULL ;

(iii) ALTER TABLE testt ADD col5 CHAR(3) NOT NULL ;

(iv) ALTER TABLE testt ADD col6 VARCHAR(3);

MODIFYING COLUMNS

Column name and data type of column can be changed as per following syntax :

ALTER TABLE <table name>
CHANGE <old column name> <new column name> <new datatype>;

If **Only data type of column need to be changed**, then

ALTER TABLE <table name>
MODIFY <column name> <new datatype>;

e.g.1. In table EMPLOYEE, change the column GROSS to SALARY.

ALTER TABLE EMPLOYEE
CHANGE GROSS SALARY INTEGER;

e.g.2. In table EMPLOYEE , change the column ENAME to EM_NAME and data type from VARCHAR(20) to VARCHAR(30).

ALTER TABLE EMPLOYEE
CHANGE ENAME EM_NAME VARCHAR(30);

e.g.3. In table EMPLOYEE , change the datatype of GRADE column from CHAR(2) to VARCHAR(2).

ALTER TABLE EMPLOYEE
MODIFY GRADE VARCHAR(2);

DELETING COLUMNS

To delete a column from a table, the ALTER TABLE command takes the following form :

ALTER TABLE <table name>
DROP <column name>;
e.g. to delete column GRADE from table EMPLOYEE, we will write :
ALTER TABLE EMPLOYEE
DROP GRADE ;

ADDING/REMOVING CONSTRAINTS TO A TABLE

ALTER TABLE statement can be used to add constraints to your existing table by using it in following manner:

TO ADD PRIMARY KEY CONSTRAINT

ALTER TABLE <table name>
ADD PRIMARY KEY (Column name);

e.g. to add PRIMARY KEY constraint on column ECODE of table EMPLOYEE , the command is :
ALTER TABLE EMPLOYEE
ADD PRIMARY KEY (ECODE) ;

TO ADD FOREIGN KEY CONSTRAINT

ALTER TABLE <table name>
ADD FOREIGN KEY (Column name) REFERENCES Parent Table (Primary key of Parent Table);

REMOVING CONSTRAINTS

To remove primary key constraint from a table, we use ALTER TABLE commandas :

ALTER TABLE <table name>
DROP PRIMARY KEY ;

To remove foreign key constraint from a table, we use ALTER TABLE commandas :

ALTER TABLE <table name>
DROP FOREIGN KEY ;

ENABLING/DISABLING CONSTRAINTS

Only foreign key can be disabled/enabled in MySQL.
To disable foreign keys : SET FOREIGN_KEY_CHECKS = 0 ;
To enable foreign keys : SET FOREIGN_KEY_CHECKS = 1 ;

INTEGRITY CONSTRAINTS/CONSTRAINTS

- A constraint is a condition or check applicable on a field(column) or set of fields(columns).
- Common types of constraints include:

	Constraints	Description
	NOT NULL	Ensures that a column cannot have NULL value
	DEFAULT	Provides a default value for a column when none is specified
	UNIQUE	Ensures that all values in a column are different
	CHECK	Makes sure that all values in a column satisfy certain criteria
	PRIMARY KEY	Used to uniquely identify a row in the table
	FOREIGN KEY	Used to ensure referential integrity of the data

NOT NULL CONSTRAINT

By default, a column can hold NULL. If you not want to allow NULL value in a column, then NOT NULL constraintmust be applied on that column.

E.g.

```
CREATE TABLE Customer
(
SID integer NOT NULL ,
Last_Name varchar(30),
First_Name varchar(30)
);
```

Columns **SID** and **Last_Name** cannot include NULL, while **First_Name** can include NULL.

An attempt to execute the following SQL statement,

```
INSERT INTO Customer
VALUES (NULL , 'Kumar' , 'Ajay');
```

will result in an error because this will lead to column SID being NULL, which violates the NOT NULL constrainton that column.

DEFAULT CONSTARINT

The DEFAULT constraint provides a default value to a column when the INSERT INTO statement does notprovide a specific value. **E.g.**

```
CREATE TABLE Student(
Student_ID integer ,
Name varchar(30) ,
Score integer DEFAULT 80);
```

When following SQL statement is executed on table created above:

INSERT INTO Student
no value has been provided for score field.
VALUES (10 , 'Ravi');

Then table **Student** looks like the following:

Student_ID	Name	Score
10	Ravi	**80**

score field has got the default value

UNIQUE CONSTRAINT

- The UNIQUE constraint ensures that all values in a column are distinct. In other words, no two rows canhold the same value for a column with UNIQUE constraint.

e.g.
CREATE TABLE Customer
(SID integer **Unique** ,
Last_Name varchar(30) ,
First_Name varchar(30)) ;

Column SID has a unique constraint, and hence cannot include duplicate values. So, if the table alreadycontains the following rows :

SID	**Last_Name**	**First_Name**
1	Kumar	Ravi
2	Sharma	Ajay
3	Devi	Raj

The executing the following SQL statement,
INSERT INTO Customer
VALUES ('3' , 'Cyrus' , 'Grace') ;
will result in an error because the value 3 already exist in the SID column, thus trying to insert another rowwith that value violates the UNIQUE constraint.

CHECK CONSTRAINT

- The CHECK constraint ensures that all values in a column satisfy certain conditions. Once defined, the table willonly insert a new row or update an existing row if the new value satisfies the CHECK constraint.

e.g.
CREATE TABLE Customer
(CID integer **CHECK (CID > 0)**,
Last_Name varchar(30) ,
First_Name varchar(30)) ;

So, attempting to execute the following statement :
INSERT INTO Customer
VALUES (-2 , 'Kapoor' , 'Raj');

will result in an error because the values for CID must be greater than 0.

PRIMARY KEY CONSTRAINT

- A primary key is used to identify each row in a table. A primary key can consist of one or more fields(column)on a table. When multiple fields are used as a primary key, they are called a **composite key.**

- You can define a primary key in CREATE TABLE command through keywords PRIMARY KEY. e.g.

CREATE TABLE Customer
(CID integer **NOT NULL PRIMARY KEY**,
Last_Name varchar(30) ,
First_Name varchar(30)) ;

Or

CREATE TABLE Customer(
SID integer,
Last_Name varchar(30) ,
First_Name varchar(30),**PRIMARY KEY (SID)**) ;

- The latter way is useful if you want to specify a composite primary key, **e.g.**

CREATE TABLE Customer
(Branch integer NOT NULL,
SID integer NOT NULL ,
Last_Name varchar(30) ,
First_Name varchar(30),
PRIMARY KEY (Branch , SID)) ;

FOREIGN KEY CONSTRAINT

- Foreign key is a non key column of a table (**child table**) that draws its values from **primary key** of anothertable(**parent table).**
- The table in which a foreign key is defined is called a **referencing table or child table.** A table to which aforeign key points is called **referenced table or parent table.**

e.g.

TABLE: STUDENT ← **Parent Table**

ROLL_NO	NAME	CLASS
1	ABC	XI
2	DEF	XII
3	XYZ	XI

Primary key

Child Table

TABLE: SCORE

ROLL_NO	MARKS
1	55
2	83
3	90

Here column Roll_No is a foreign key in table SCORE(Child Table) and it is drawing its values fromPrimary key (ROLL_NO) of STUDENT table.(Parent Key).

CREATE TABLE STUDENT
(ROLL_NO integer **NOT NULL PRIMARY KEY** ,
NAME VARCHAR(30) ,
CLASS VARCHAR(3));

CREATE TABLE SCORE
(ROLL_NO integer ,
MARKS integer ,
FOREIGN KEY(ROLL_NO) **REFERNCES** STUDENT(ROLL_NO)) ;

**** Foreign key is always defined in the child table.***

Syntax for using foreign key

FOREIGN KEY(column name) REFERENCES Parent_Table(PK of Parent Table);

REFERENCING ACTIONS

Referencing action with ON DELETE clause determines what to do in case of a DELETE occurs in the parent table. Referencing action with ON UPDATE clause determines what to do in case of a UPDATE occurs in the parent table.

Actions:

1. **CASCADE :** This action states that if a DELETE or UPDATE operation affects a row from the parent table, thenautomatically delete or update the matching rows in the child table i.e., cascade the action to child table.
2. **SET NULL :** This action states that if a DELETE or UPDATE operation affects a row from the parent table, thenset the foreign key column in the child table to NULL.
3. **NO ACTION :** Any attempt for DELETE or UPDATE in parent table is not allowed.
4. **RESTRICT :** This action rejects the DELETE or UPDATE operation for the parent table.

Q: Create two tables

Customer(customer_id, name) Customer_sales(transaction_id, amount , **customer_id**)

Underlined columns indicate primary keys and bold column names indicate foreign key.

Make sure that no action should take place in case of a DELETE or UPDATE in the parent table.

Sol : CREATE TABLE Customer (
customer_id int Not Null Primary Key ,
name varchar(30)) ;

CREATE TABLE Customer_sales (
transaction_id Not Null Primary Key ,
amount int ,
customer_id int ,
FOREIGN KEY(customer_id) REFERENCES Customer (customer_id)
ON DELETE NO ACTION
ON UPDATE NO ACTION);

Q: Distinguish between a Primary Key and a Unique key in a table.

	PRIMARY KEY	UNIQUE KEY
	Column having Primary key can't contain NULL value	Column having Unique Key can contain NULL value
	There can be only one primary key in Table.	Many columns can be defined as Unique key

Q: Distinguish between ALTER Command and UPDATE command of SQL.

	ALTER COMMAND	UPDATE COMMAND
	It is a DDL Command	It is a DML command
	It is used to change the definition of existing table, i.e. adding column, deleting column, etc.	It is used to modify the data values present in the rows of the table.
	Syntax for adding column in a table: ALTER TABLE <tablename> ADD <Column name><Datatype> ;	Syntax for using UPDATE command: UPDATE <Tablename> SET <Columnname>=value WHERE <Condition> ;

Introduction to the Emerging Trends

1. List some of the cloud-based services that you are using at present.
   Ans.
   IaaS – Google Drive, Git, Google Docs
   PaaS – Acquia, AWS (Amazon Web Services), Heroku
   SaaS – Microsoft Office 365, Google Doc, Drop Box

2. What do you understand by the Internet of Things? List some of its potential applications.
   Ans.
   IoT or Internet of Things refers to connecting many devices through hardware or software and communicate with each other as well as human beings through internet to work with collaboration and assist each other and form a networking of things.

   Potential Applications of IoT :

   1. Smart Home
   2. Wearables
   3. Smart City
   4. Smart Grids
   5. Industrial Internet

3. Write a short note on the following:
   Ans.
   a) **Cloud Computing**:

   In cloud computing refers to a virtual computers than can be access a software, services or storage on internet.

   These software, services, and storage provided by cloud services providers based on payment or some of them are free also. It allows to use large applications through the internet without any physical storage.

   If you are using Facebook, google photos, google drive, sky drive, etc. these are examples of cloud computing.

Advantages:

1. Accessible anytime, anywhere on the internet
2. Cost-effective and reasonable cost
3. On-demand sources
4. Allows to use of resources as per the need

b) Big data and its characteristics

Every day, over 2.5 quintillion bytes of data are generated, and the rate is rising due to the Internet of Things' ongoing development (IoT). As a result, big data—data sets with a large volume and high level of complexity—are created.

Such data cannot be handled and analyzed using conventional data processing methods because it is not only large but also unstructured, such as our posts, conversations, and instant messages, as well as the pictures. This data is highly valuable in the businesses, So, there is a strong focus on developing tools and processes to process and analyze big data.

This definition clearly focuses on:

1. Large or Huge volume of data which is going to be processed
2. It can extract information which gives a huge benefit
3. It can help in decision making
4. It can provide insight information

Big data increases the volume day by day. It is in huge amount so cannot be processed or analyzed using conventional methods. Some of the popular terms associated with big data for processing are data mining, data storage, data analysis, data sharing, and data visualization.

Characteristics of Big data

I. **Volume:** It refers to the number of big data in terms of size. It is in very big size or voluminous data.
II. **Velocity:** It refers to the rate of data transfer at very high speed.
III. **Variety:** Data is available in different forms like structured, unstructured, and semi structured. It can be different files like text, images, videos, web pages, etc.

IV. **Veracity:** It refers to the intensity of the reliability of data. When a large volume of data is going to be processed, sometimes chances are there to produce incorrect data.

V. **Value:** It is the quantity of data including amount, reliability of data to be processed.

4. Explain the following along with their applications.
   a) Artificial Intelligence
   b) Machine Learning
   Ans.

Artificial Intelligence (AI)

Artificial intelligence, or I, refers to devices or programmes that resemble human intelligence in order to carry out tasks and have the ability to iteratively improve themselves based on the data they gather.

Artificial intelligence aims to imitate human intelligence naturally in machines so that they will act intelligently. An intelligent machine should be able to mimic some of the cognitive processes that people use to learn, make decisions, and solve problems. Machines are trained to build a knowledge base and base judgments off of it in order to complete jobs with the least amount of human involvement. In order to make new judgments, AI systems can also learn from their prior actions or results.

Machine Learning

Machine learning is a branch of artificial intelligence that enables computers to learn from data using statistical methods without explicit human programming. It includes algorithms that use information to learn on their own and anticipate the future.

5. Differentiate between cloud computing and grid computing with suitable examples.
   Ans.

Cloud computing	Grid computing
Cloud computing is client server architecture	Grid computing is distributed server architecture
In cloud computing, resources are used in centralized pattern	In grid computing, resources are used in collaborative pattern
Cloud service is a paid service	grid service is a free service
It is more flexible	It is less flexible
Example: Salesforce, hubspot	Example: Bluegrid, DISCOM

6. Justify the following statement-
   'Storage of data is cost effective and time saving in cloud computing.'
   Ans.
   Cloud computing stores data into a server and it is available at almost free of cost or nominal cost. When it comes to time-saving by saving data on the internet there is no need to wait to start up any computer or device. It is easily available with a browser program or app of that. So it can be accessed any time from anywhere.

7. What is on-demand service? How it is provided in cloud computing?
   Ans.
   On-demand service means that data is available as and when required. In cloud computing, users can access any file by using a login with a username or email account and password. This is called on-demand service.

8. Write examples of the following:
   a) Government provided cloud computing platform
   b) Large scale private cloud service providers and the services they provide
   Ans.
   a) The government of India has provided a government cloud service named MeghRaj.

b) Google is large scale private cloud service providers which provides storage, web services and many more things.

9. A company interested in cloud computing is looking for a provider who offers a set of basic services such as virtual server provisioning and on-demand storage that can be combined into a platform for deploying and running customised applications. What type of cloud computing model fits these requirements?
   a) Platform as a Service
   b) Software as a Service
   c) Infrastructure as a Service

   Ans.

   The correct answer is option c) Infrastructure as a Service

10. Which is not one of the features of IoT devices?
    a) Remotely controllable
    b) Programmable
    c) Can turn themselves off if necessary
    d) All of the above
    Ans.
    The correct answer is option a) Remotely controllable

11. If Government plans to make a smart school by applying IoT concepts, how can each of the following be implemented in order to transform a school into IoT enabled smart school?
    a) e-textbooks
    b) Smart boards
    c) Online tests
    d) Wifi sensors on classrooms doors
    e) Sensors in buses to monitor their location
    f) Wearables (watches or smart belts) for attendance monitoring
    Ans.
    a) e-textbooks : e-textbooks can be made available e-library by using web services.

    b) Smart Boards: Smart boards can be operated by digital stylus or pen and should be connected with network.

c) Online Tests: A web portal or online examination should be prepared to conduct exams smoothly.

d) Wifi sensors and classrooms doors: Wifi sensors and classroom doors can be used to lock and unlock them. These sensors also provides security to students and other stakeholders.

e) Sensors in buses to monitor their location: These sensors enables quick reporting and monitoring to the coordinates and parents to know the status of child and bus after leaving the school.

f) Wearables (watches or smart belts) for attendance monitoring: These devices can be used or implemented for students in such a way that identifies a student and marks the attendance accordingly.

12. Five friends plan to try a startup. However, they have a limited budget and limited computer infrastructure. How can they avail the benefits of cloud services to launch their startup?
    Ans.
    They can go for registration with any suitable service provider either government or private. They can register for Infrastructure as a service to avail the facility of cloud storage and deploy applications initially.

13. Governments provide various scholarships to students of different classes. Prepare a report on how block chain technology can be used to promote accountability, transparency, and efficiency in distribution of scholarships?

    Ans.
    As we are aware in blockchain technology, a separate ledger is created for individual and stored so the transparency and efficiency automatically improve. The students who can avail the scholarship in their ledger only. The transactions are also more authentic.

14. How IoT and WoT are related?
    Ans.
    IoT and WoT both can be user on network to communicate with multiple devices. They can provide a communication medium between various

devices, this medium can be an app, a web service or any network enabled device.

15. Match the following:

Column A	**Column B**
You got a reminder to take medication	Smart Parking
You got a sms alert that you forgot to lock the door	Smart Wearable
You got the sms alert that parking space is available near your block	Home Automation
You turned off your LED TV from your wrist watch	Smart Health

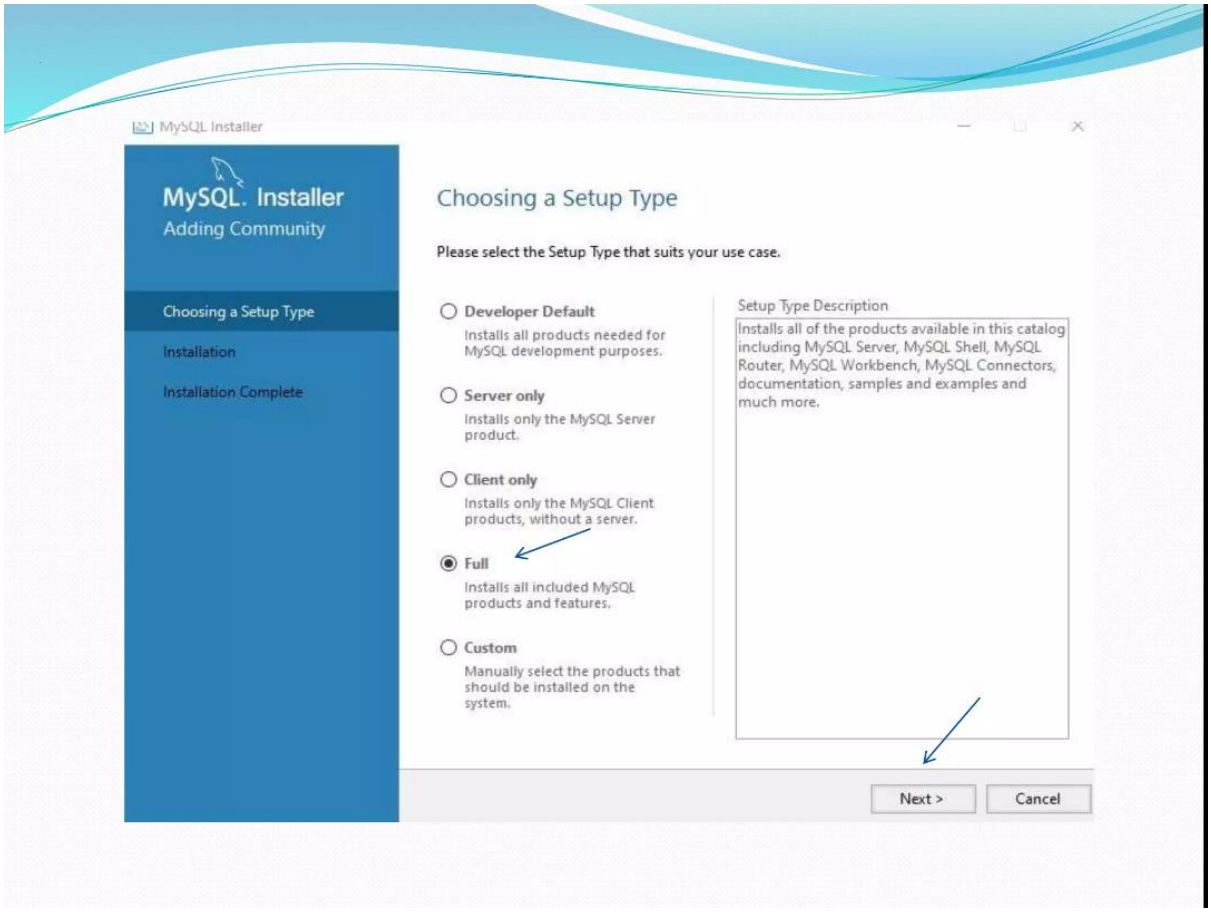
Ans.

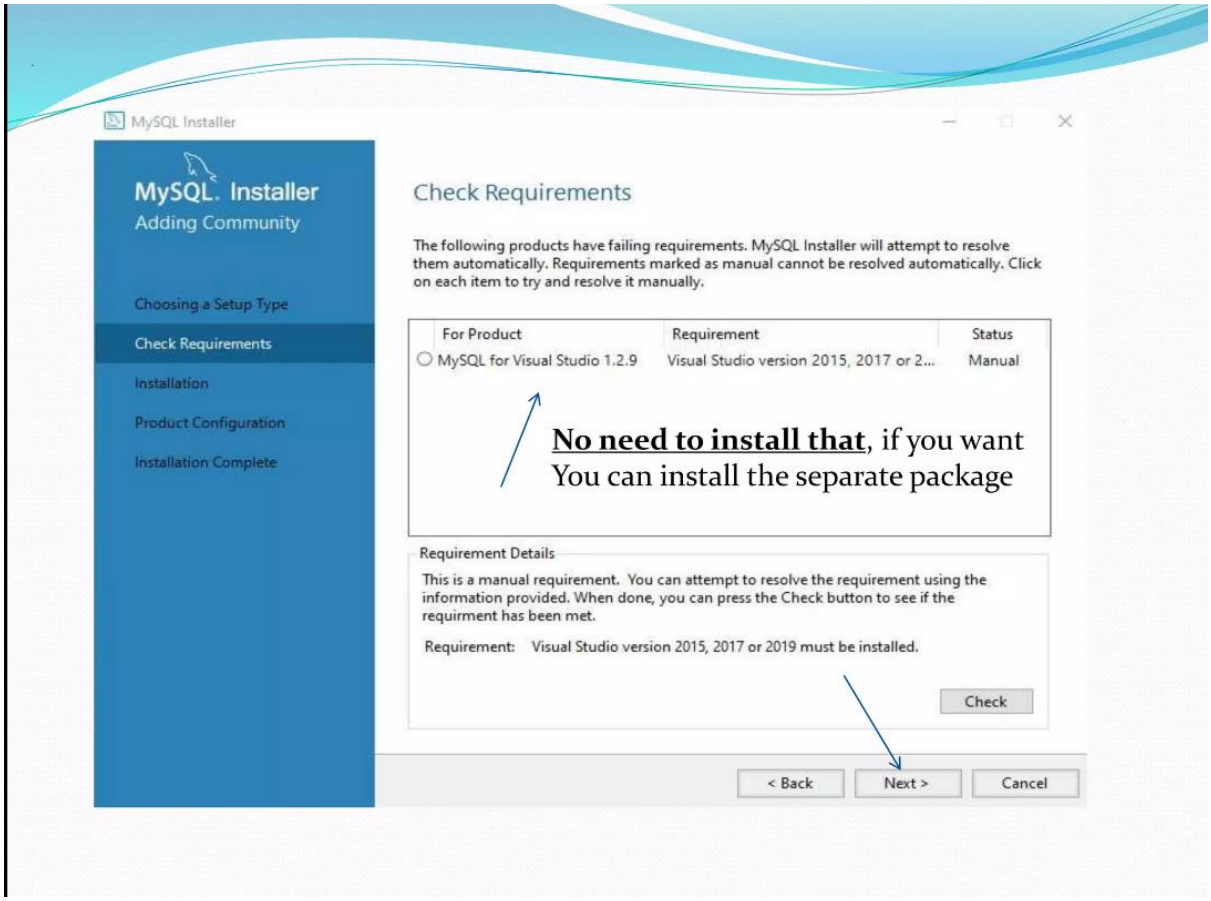
1 -> 4

2 -> 3

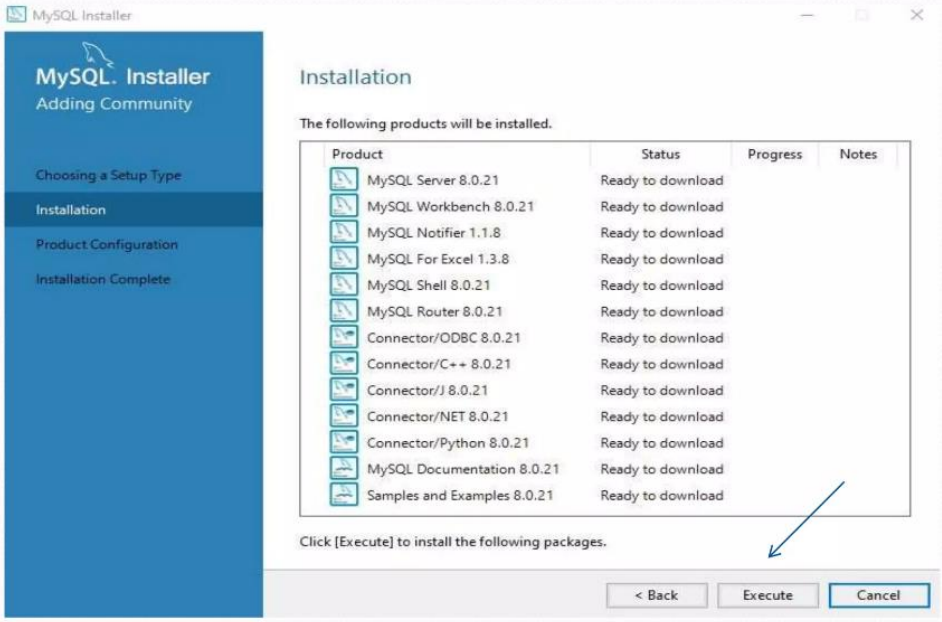
3 -> 1

4 -> 2


MySQL Installer
MySQL. Installer
Adding Community
Choosing a Setup Type
Installation
Installation Complete
Choosing a Setup Type
Please select the Setup Type that suits your use case.
Developer Default
Installs all products needed for MySQL development purposes.
Server only
Installs only the MySQL Server product.
Client only
Installs only the MySQL Client products, without a server.
Full
Installs all included MySQL products and features.
Custom
Manually select the products that should be installed on the system.
Setup Type Description
Installs all of the products available in this catalog including MySQL Server, MySQL Shell, MySQL Router, MySQL Workbench, MySQL Connectors, documentation, samples and examples and much more.
Next >
Cancel

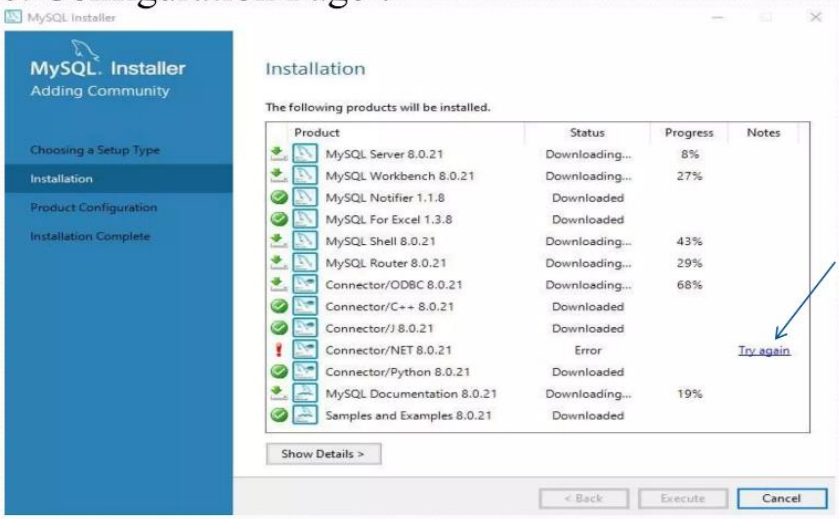

MySQL Installer
MySQL. Installer
Adding Community
Choosing a Setup Type
Check Requirements
Installation
Product Configuration
Installation Complete
Check Requirements
The following products have failing requirements. MySQL Installer will attempt to resolve them automatically. Requirements marked as manual cannot be resolved automatically. Click on each item to try and resolve it manually.
For Product
Requirement
Status
MySQL for Visual Studio 1.2.9
Visual Studio version 2015, 2017 or 2...
Manual
No need to install that, if you want
You can install the separate package
Requirement Details
This is a manual requirement. You can attempt to resolve the requirement using the information provided. When done, you can press the Check button to see if the requirment has been met.
Requirement: Visual Studio version 2015, 2017 or 2019 must be installed.
Check
< Back
Next >
Cancel

Then in next page 'Installation Page' click execute , you will see the files are downloading



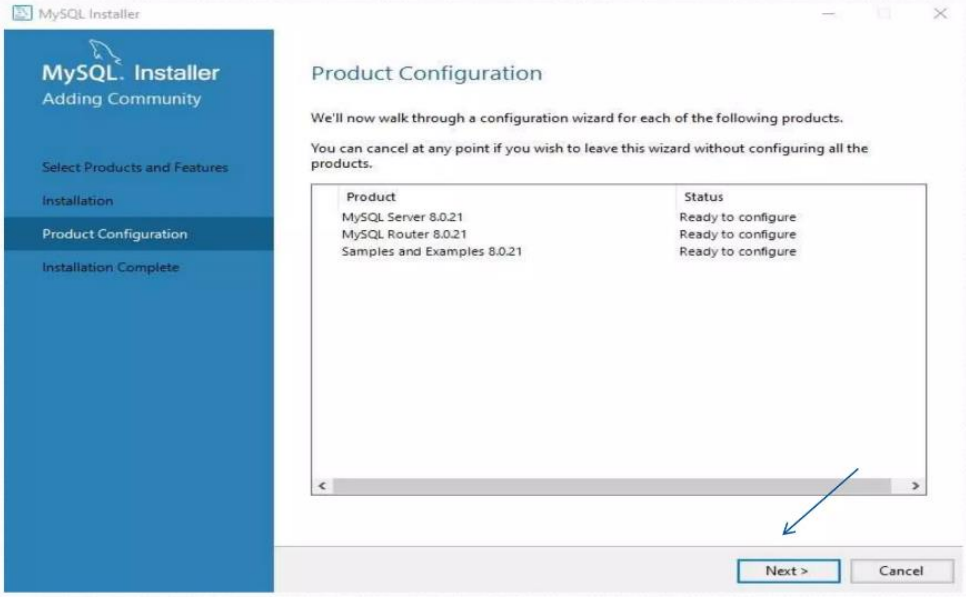
* If it shows error, just **click try again**, it will get downloaded and installed

* Then after finishing the download click next, which leads to 'Product Configuration Page'.

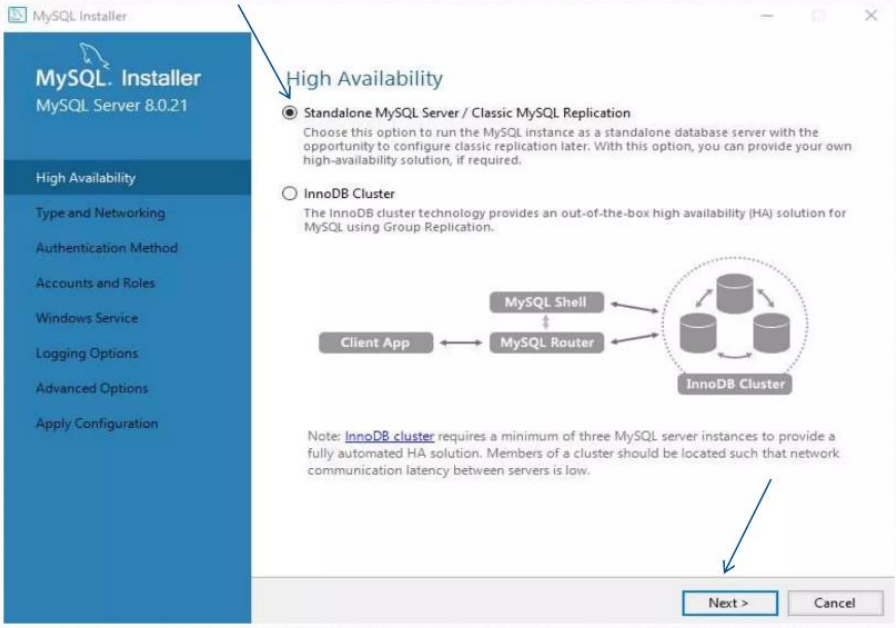


Click 'try again' if it shows error

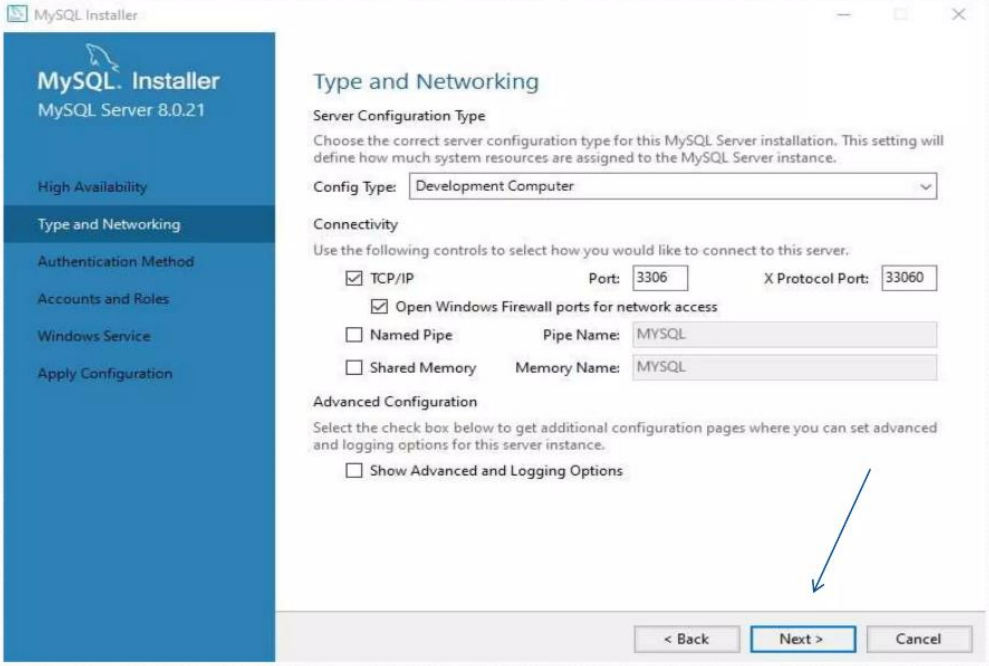
Here choose next to configure the products.



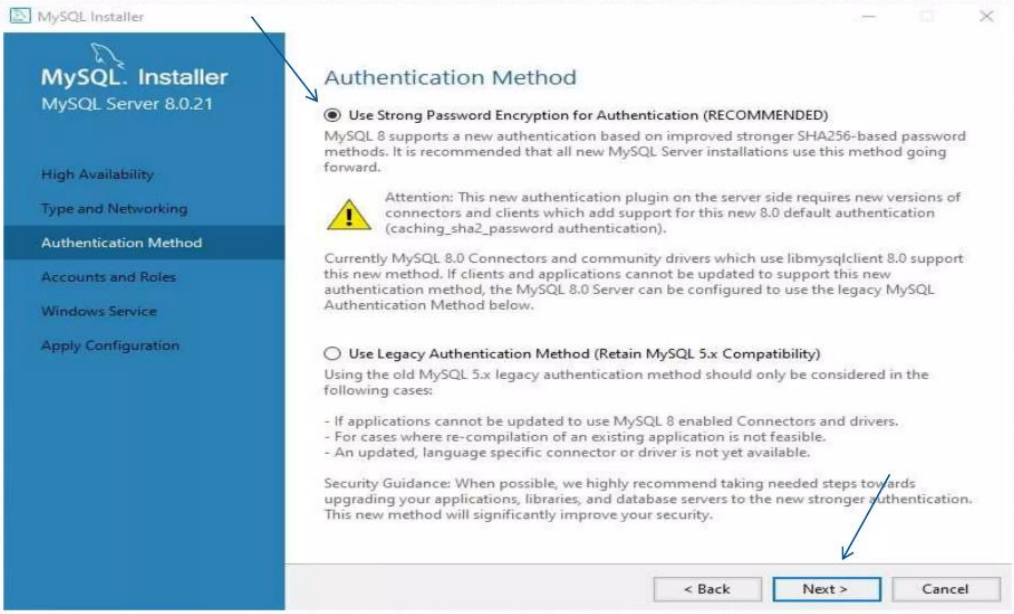
Step-1: Choose Standard MySQL server/ Classic MySQL Replication and 'Next'



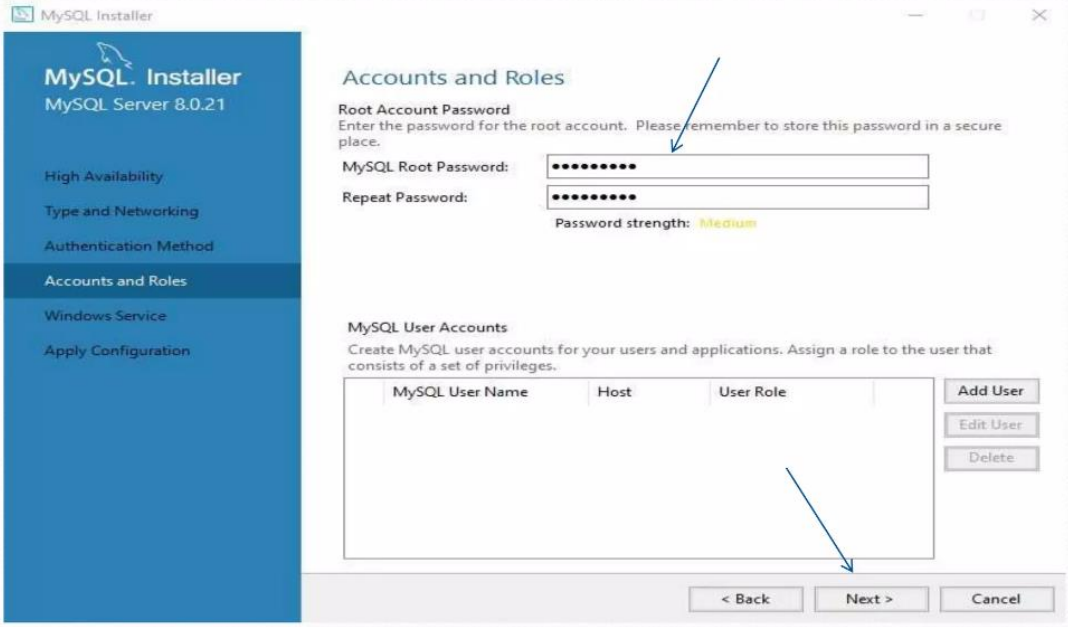
Step-2: Leave the default selections and click 'Next'



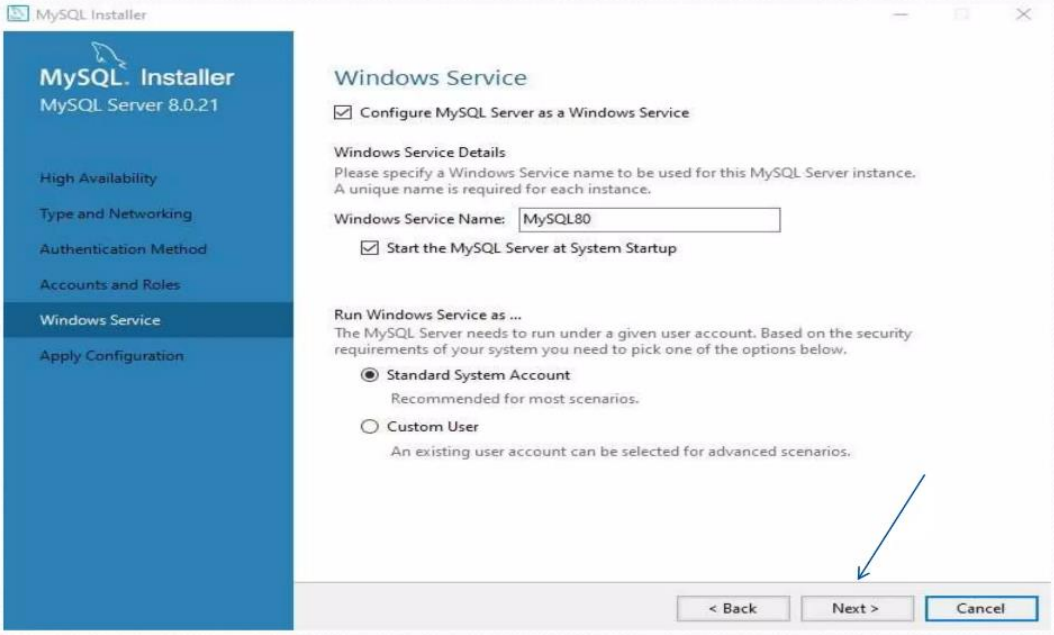
Step-3: Choose the first option 'use strong Password'



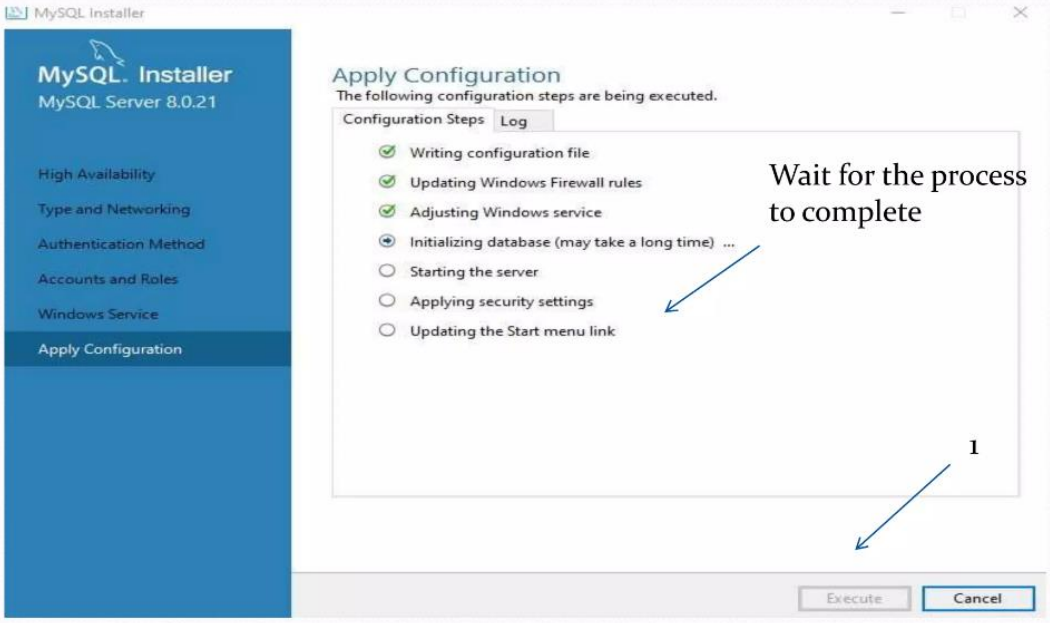
Step-4: Set your password and select 'Next'



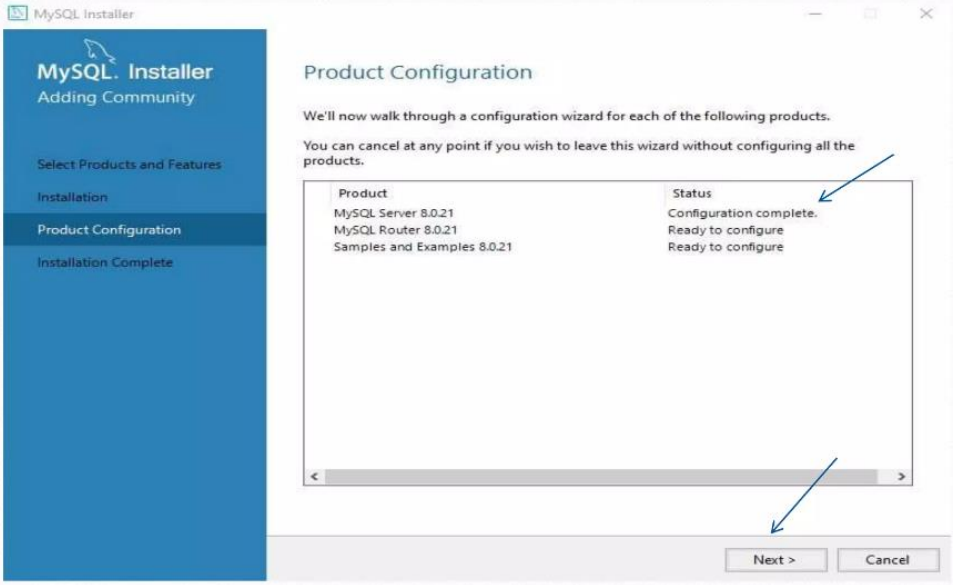
Step-5: Leave the default selections and click 'Next'



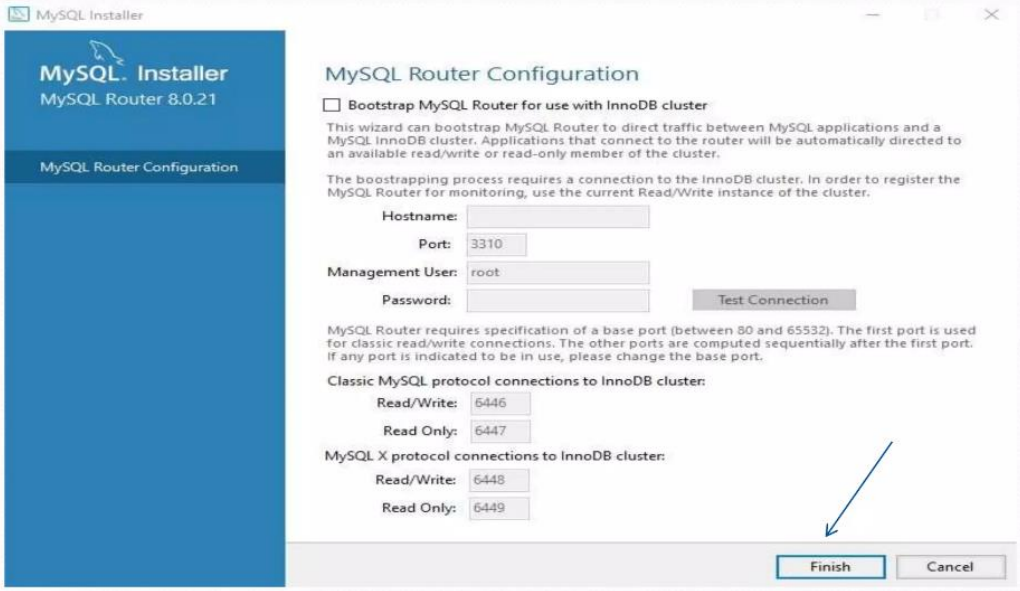
Step-6: Choose Execute and wait for the configuration steps to finish and select 'Finish'



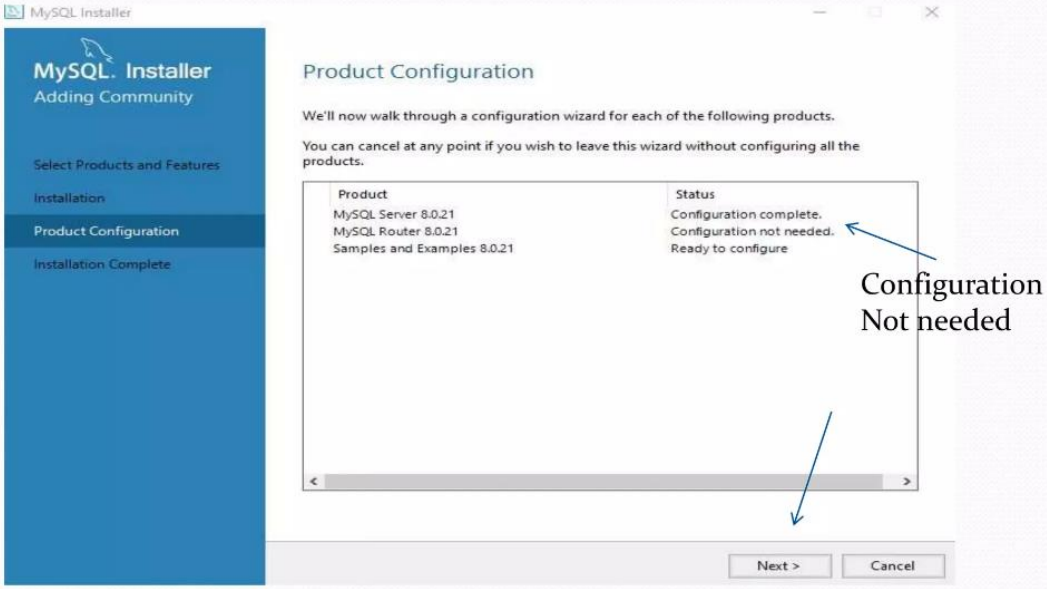
As we can see the **MySQL server** configuration is completed and now we have to configure the **Router** so choose 'Next'

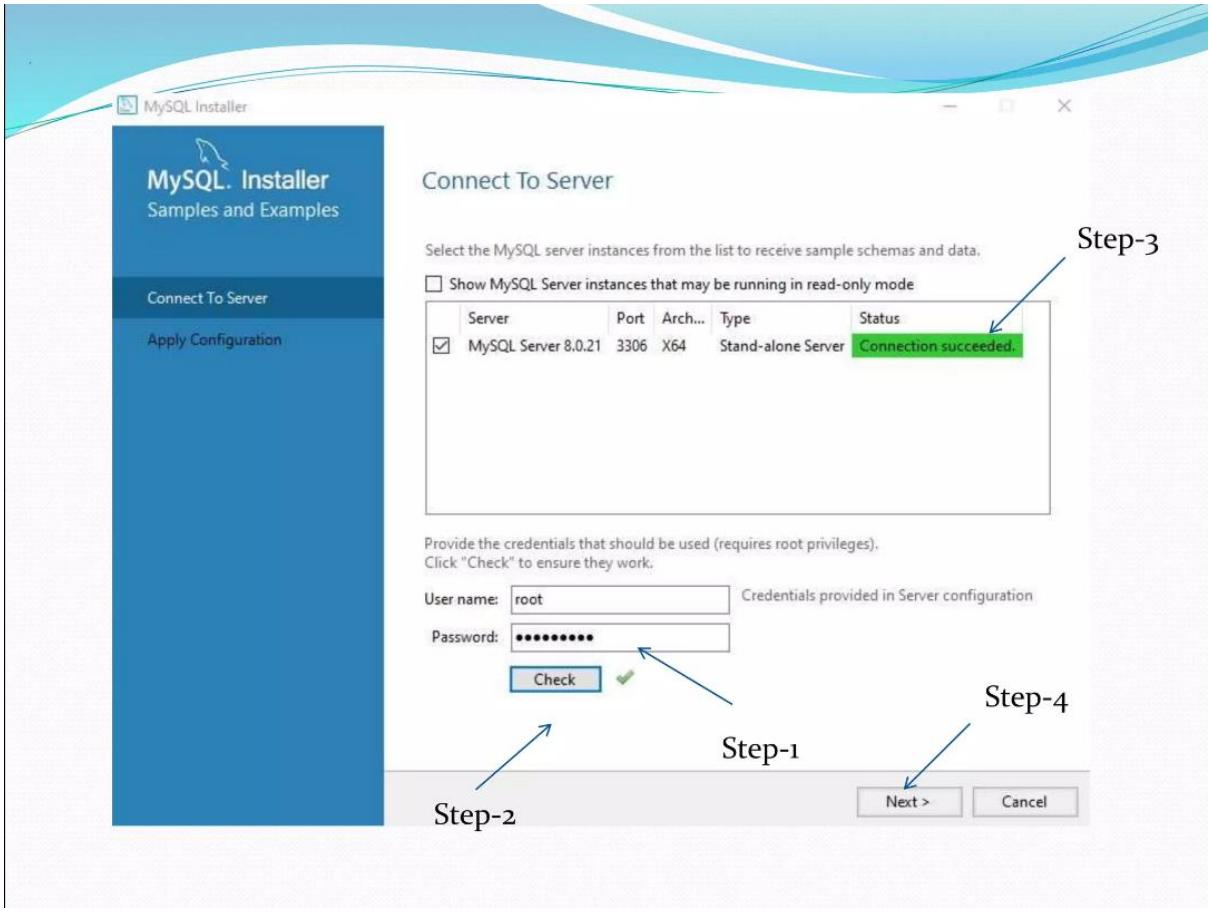


Here we don't configure the 'Router' because mostly its not needed, if we have to do we can **re-configure** later in MySQL installation, so choose 'Finish'.



As we can see we not configured the 'Router', now we have to configure the 'Sample and Example' so click Next



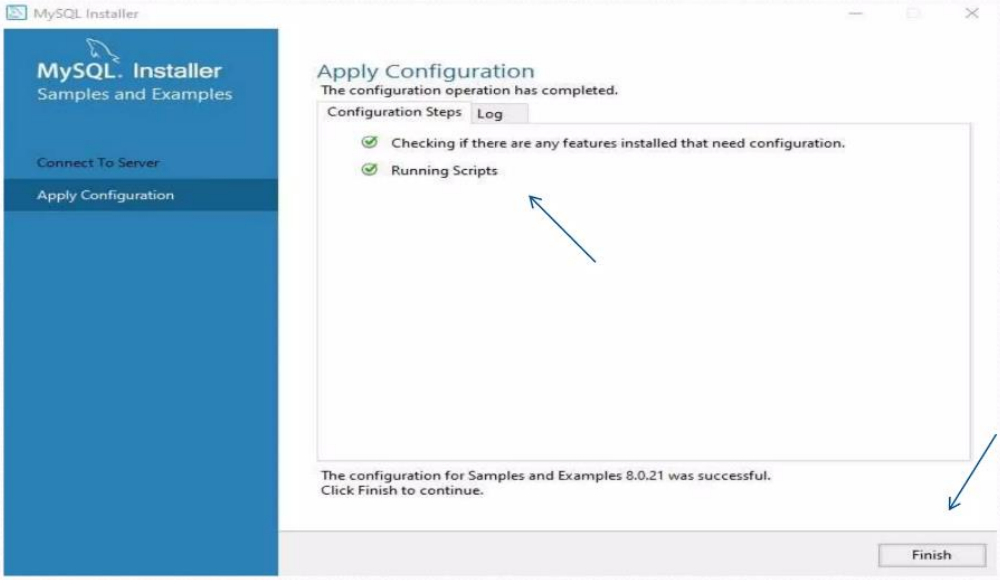


In this 'Apply Configuration' page

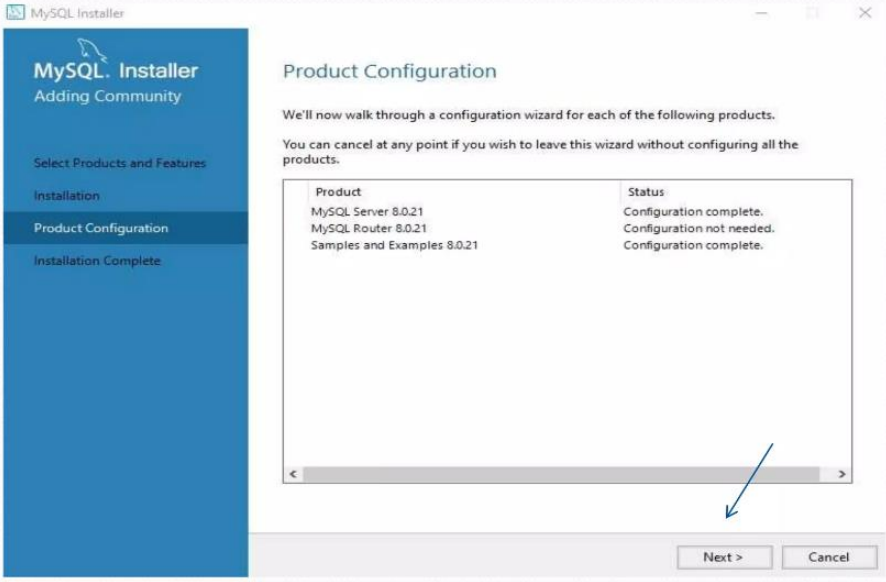
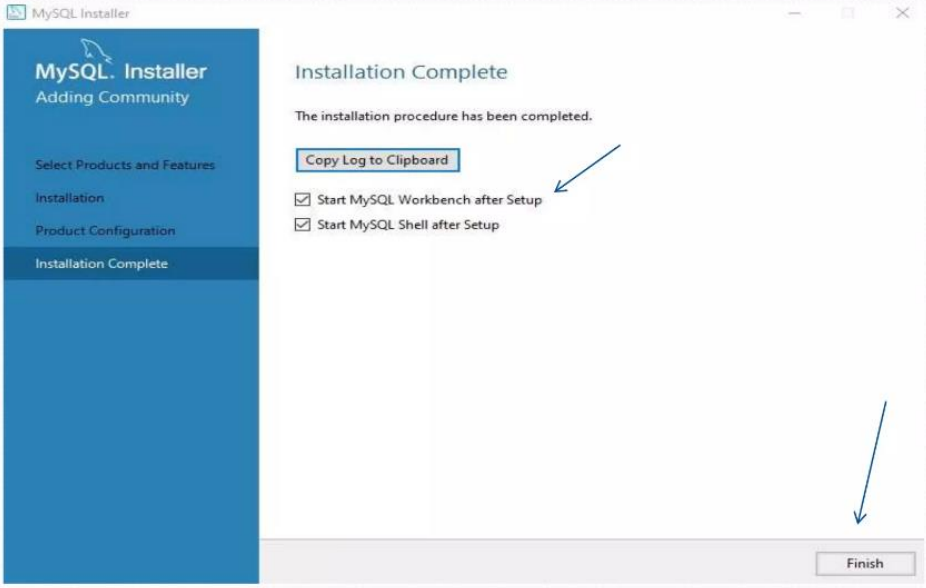
Step-1: Choose execute

Step-2: Wait for the Process to complete

Step-3: After completion select 'Finish'



As we see the configuration for Samples and Example is complete and click 'Next'



We are in a Installation Complete Page, leave the default selections and choose 'Finish' and wait for the MySQL Workbench to popup