Introduction

Q-Programming is dedicated to providing numerous resources for secondary school students. This free book is a complete guide for the CBSE 2021-22 Term 2 Board Exam for Computer Science with Python.

Make sure to check out our YouTube channel for detailed lectures on each topic: https://youtube.com/qprogramming.

Visit our website for many more free resources for the CBSE curriculum: https://qprogramming.net/cbse.

Book Structure

The book has been divided into two sections: **notes** and **questions**.

The notes section contains all the relevant notes from NCERT and the official syllabus, and additional information as well for your knowledge (which could potentially appear in the exam). Examples of code have been given with syntax highlighting to demonstrate concepts. You are encouraged to attempt the examples on your system and also play around with the code for better understanding. We have also consulted the computer science reference books authored by Sumita Arora (Computer Science with Python) and Preeti Arora (Computer Science with Python) to ensure all topics are covered.

The authors would like to thank the Q-Programming team and their friends, family and teachers for their support in this endeavour.

User Agreement

This publication is the property of the authors and may not be circulated or sold for profit without the authors' consent and cannot be reproduced or copied without the written permission from **both** authors. Any violation of this condition is liable for legal action.

A large effort has gone into making this publication and we have tried to ensure that all information is accurate and that almost all the relevant material has been covered. In the event that you feel certain material has been left out or is incorrect, you may email us at contact@qprogramming.net for such queries.

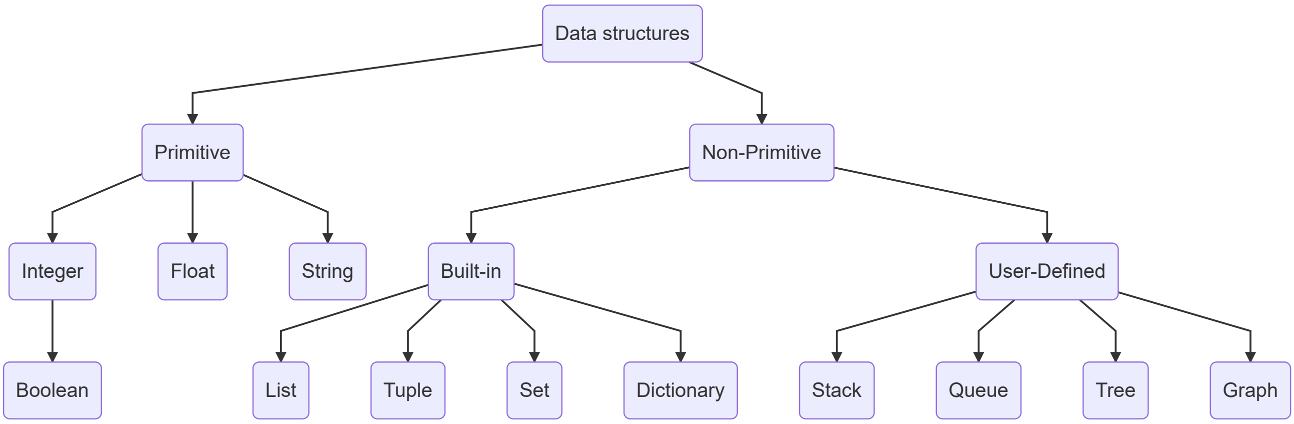
It is notified that by using this publication, you agree to not hold Q-Programming or the authors accountable for any form of errors or discrepancies and agree to all the conditions listed above.

Table of Contents

Chapter 1: Data Structures

Data structures are used to **store, organize and access** data and allow certain **operations** to be performed on data.

They can be classified as follows:



Stack

A stack is a sequential collection of elements.

It follows the principle of **LIFO (Last In First Out)**

This means that the **last element to go in the stack is also the first element to go out**.

Let's understand this through an example. Suppose you are buying study books and want to pile them up:

- When you want to add a new book, you will always add it at the top of the pile.
- To remove a book, it will be inconvenient to remove from the middle or the bottom of the stack, as it may fall. Therefore we remove the book at the top

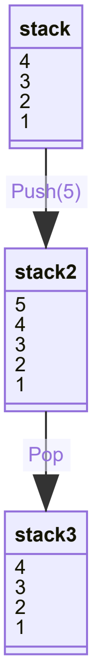
The "TOP" of a stack is the **end position in the stack**, where elements are added and removed.

Operations on Stacks:

1. **PUSH**: It **adds a new element** to the TOP.

   Pushing to a full stack results in an**'overflow'** exception.
2. **POP**: It **removes the TOP**.

   Popping an empty stack results in an **'underflow'** exception.



Applications of Stacks in Programming

1. **Reversing Strings:** Push every character to the stack and then pop one by one.
2. **File History:** Maintains a sequential list of all actions performed and you can undo the last action.
3. **Browser History**
4. **Expression Evaluation:** Evaluating arithmetic expressions and pairing parentheses.

Implementing a Stack in Python

To implement stack in python, we use **lists**, as there is no pre-defined datatype in python specifically meant for stacks.

To start, we must define what the "TOP" of the stack would be. We will be using list methods **append()** and **pop()** for pushing and popping respectively in the stack.

A quick recap of the 2 methods:

1. `append(x)` : It takes in an argument and adds that element to the end of a list

   For example:

   ```
   L = [1,2,3,4]
   L.append(5)
   print(L)
   ```

   The output would be the list `[1,2,3,4,5]`. As you can see, it added append the number 5 to the end of the list (which we call **TOP**)

2. `pop()`: It takes an argument for the index position and removes the element at that position of the list. The default value of the index position is -1 (last element of the list). It also returns the popped element.

For example:

```python
L = [1,2,3,4]
print(L.pop()) #Default is -1, the last element (aka TOP)
print(L)
```

Output:

```
4
[1,2,3]
```

The first output was 4 since that is the element that was popped. After being popped, the list became `[1,2,3]`

The `append()` and `pop()` functions behave like the push and pop operations respectively, which is what we need. They act on the TOP of the stack. Now let's begin implementing stack.

- Defining an empty stack to start

```python
stack = list() # you can also use [] to signify an empty list
```

The `pop()` function in python does not account for underflow, it just throws an IndexError.

We will define a function `isempty()` to check whether the stack is empty or not.

- Checking if stack is empty

```python
def isEmpty(stack):
    if len(stack)==0:
        return True # Return True if the stack is empty
    else:
        return False
```

- Defining the **pop** operation

```python
def pop(stack):
    if isEmpty(stack):
        print("Underflow Error.") # This indicates that the function received an empty stack
        return
    else:
        return stack.pop() # Remove element from end of list and prints
```

- Defining the **push** operation

```python
def push(stack,element):
    stack.append(element) # Adding element to the end, aka TOP
```

An advantage of python is that lists don't have a pre-defined size. Many languages require you to define the array with a fixed size before hand, which could lead to an overflow error.

Python doesn't have that, that is why it is unlikely that you face an overflow error (except in cases where the memory limit of your device is exceeded).

- Checking the **size** of the stack

```
def size(stack):
    return len(stack)
```

- Finding the element at the **top** of the stack ie. the last element in the stack

```
 def top(stack):
     if isEmpty(stack):
         print("Stack is empty")  # An empty stack was received
         return
     else:
         element = stack[-1]      # Return the last element in the stack
         return element
```

We can now create a **menu-driven program** to test our implementation of the stack.

```
stack = list()

def size(stack):
    return len(stack)

def isEmpty(stack):
    if size(stack) == 0:
        return True
    else:
        return False

def push(stack, element):
    stack.append(element)

def pop(stack):
    if isEmpty(stack):
        print("Error: Underflow")
        return
    else:
        return stack.pop()

def top(stack):
    if isEmpty(stack):
        print('Stack is empty')
        return
    else:
        return stack[-1]

def display(stack):
    print("Current elements in the stack are: ",end="")
    for i in range(len(stack)-1,-1,-1):
        print(stack[i], end = " ")
    print()


while True:
    menu = """Select the option:

    1. Push to stack
    2. Pop stack
    3. Get top element
    4. Display stack
    5. exit\nEnter choice: """
    menu = input(menu)

    if menu == "1":
        element = input("Enter element here: ")
        push(stack, element)
    elif menu == "2":
        print("Popped", pop(stack), "| current stack:", stack)
    elif menu == "3":
        print(top(stack))
    elif menu == "4":
        display(stack)
    elif menu == "5":
        break
    else:
        print("Not a valid option, please try again.")
```

Output:

```
Select the option:

    1. Push to stack
    2. Pop stack
    3. Get top element
    4. Display stack
    5. exit
Enter choice: 1
Enter element here: 1
Select the option:

    1. Push to stack
    2. Pop stack
    3. Get top element
    4. Display stack
    5. exit
Enter choice: 1
Enter element here: 2
Select the option:

    1. Push to stack
    2. Pop stack
    3. Get top element
    4. Display stack
    5. exit
Enter choice: 2
Popped 2 | current stack: ['1']
Select the option:

    1. Push to stack
    2. Pop stack
    3. Get top element
    4. Display stack
    5. exit
Enter choice: 3
1
Select the option:

    1. Push to stack
    2. Pop stack
    3. Get top element
    4. Display stack
    5. exit
Enter choice: 1
Enter element here: 3
Select the option:

    1. Push to stack
    2. Pop stack
    3. Get top element
    4. Display stack
    5. exit
Enter choice: 4
Current elements in the stack are: 3 1
Select the option:

    1. Push to stack
    2. Pop stack
    3. Get top element
    4. Display stack
    5. exit
Enter choice: 5
```

Menu Key Points

- All functions to be defined
- Use a infinite while loop so the user can enter the choice as many times as needed
- Ensure there is an exit option to break the loop
- Display output for all options
- Create an `else` option in case the wrong choice is entered

Evaluating Arithmetic Expressions

Notation for arithmetic expressions

We write arithmetic expressions by using an operator between operands, for eg `x+2*y` or `x-y`

We even use **parenthesis** to order the evaluation of complex expressions. The evaluation of these expressions happens using **BODMAS** rule. This is called **Infix Notation**

For example, if you have the expression `6-8/2`. According to the rules of BODMAS, we will first deal with the division of `8/2` then subtract that result from `6`, instead of doing `6-8` first. But this will be **time consuming** for a computer to do since it will have to traverse the expression multiple times

Then 2 new notations were proposed that were easier for computers to deal with using **stacks**:

1. **Prefix**: The operators are written before the operands
   For example: `x+y` becomes `+xy`
2. **Postfix**: The operators are written after the operands
   For example: `x+y` becomes `xy+`

Given a Prefix/Postfix expression, computers can easily evaluate them in a single traversal from left to right, since the operators becomes positioned as per their order of evaluation

Converting an expression from Infix to Postfix

To perform this conversion successfully, you need to know the precedence of operators.

Operator	Use
()	grouping
**	Exponent
~x	Bitwise nor
+x, -x	Unary operation
*,/,//,%	Multiplication, division, floor division, modulus
+, -	Addition and subtraction
&	Bitwise AND
^	Bitwise XOR
|	Bitwise OR
<, <=,>,>=,!=,==,is,is not	Relational Operators and Identity Operators
not	Boolean NOT
and	Boolean AND
or	Boolean OR

Keep this order in mind, since we will be needing it.

Now, for example, we have an infix expression `(P/(Q-R)*S+T)`. You **must** add brackets to the front and the back of the expression.

To convert to postfix, we will follow this process:

Step 1: Create three columns: **Token**, **Stack** and **Postfix**

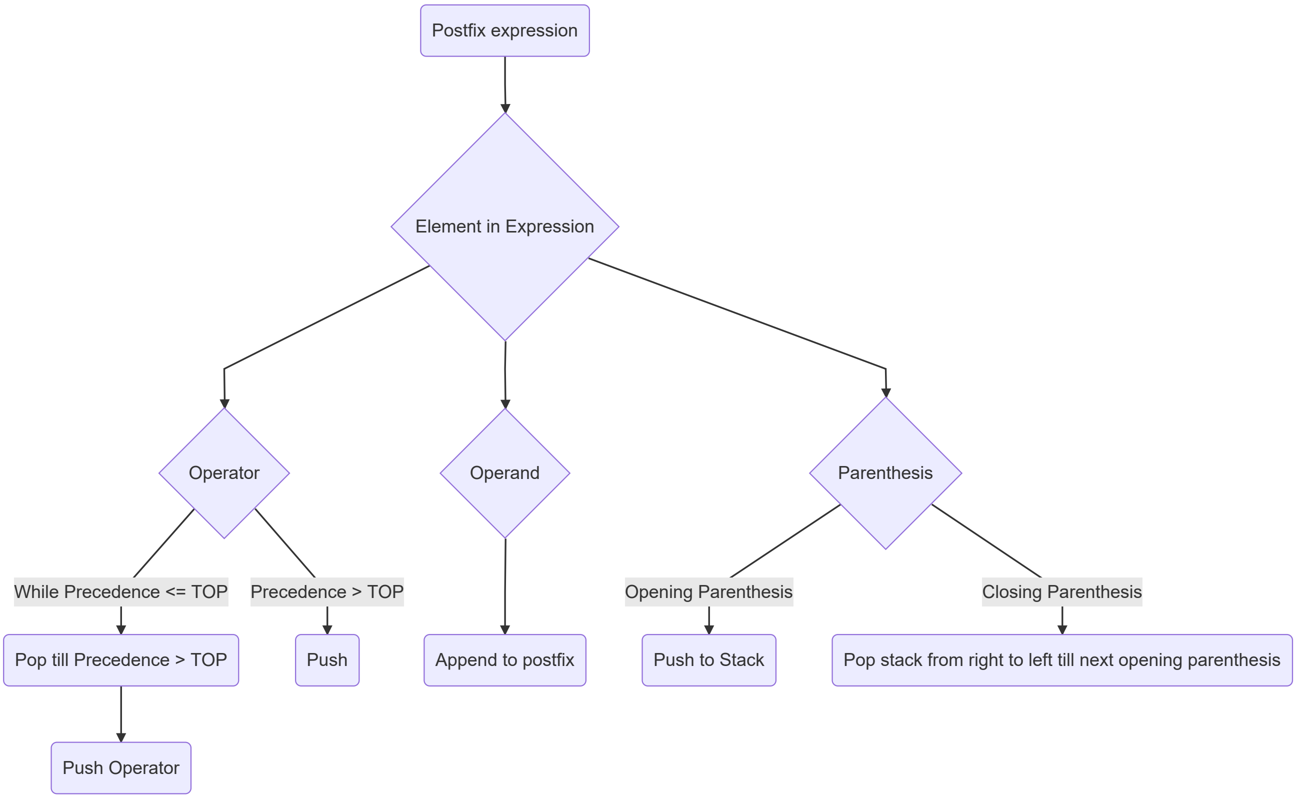
Step 2: Add every element in the expression to the Token column

Step 3: For each element you add, there can be these possibilities:

```
If the element is an operand
    add in Postfix column

Else if element is an operator
    If precedence of element > precedence of operator in stack's TOP
        Push in stack

    else if stack's TOP contains left bracket (
        Push in stack

    else
        Pop all operators which have greater or equal precedence into the Postfix expression

Else if element is parenthesis
    If element is opening parenthesis (
        Push to stack

    else if element is closing parenthesis )
        Pop all elements in stack from right to left until a left bracket is encountered into the Postfix expression
```

When each element in the infix expression has been evaluated through this process, you must pop the remaining elements in the stack and append them to postfix.

We have simplified the process through this flowchart for the case of each possible term in an expression.

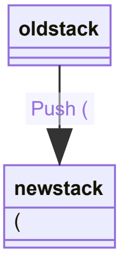


Once all the terms have been evaluated, you must then pop all remaining items in the stack and append them to the postfix expression.

Let's visualize this algorithm using a simple expression `(x + y) * (z - w)`

1. `(`

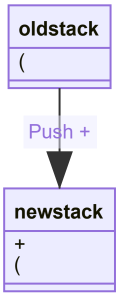
   We must push parenthesis to the stack.



2. `x`

   Add `x` to the postfix expression as it is an operand.

3. `+`

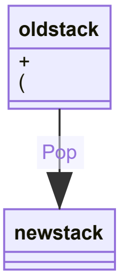


4. `y`

   Add `y` to postfix: `xy` as it is an operand.

5. `)`

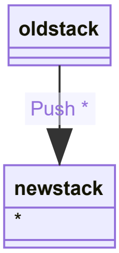
   Pop stack and append to postfix as the parentheses are closed and add operators to the expression.



   Postfix: `xy+`

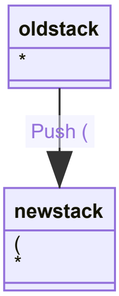
6. `*`

   Push operator to stack



7. `(`

   Push to stack

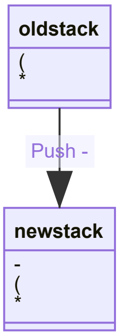


8. `z`

   Add operand to postfix: `xy+z`

9. `-`

   Push to stack

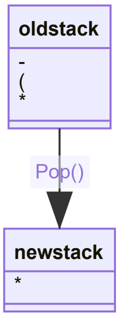


10. `w`

    Add to postfix: `xy+zw`

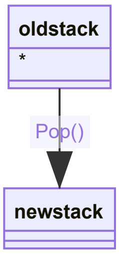
11. `)`

    Pop stack



Postfix: `xy+zw-`

12. Final pop



Final postfix: `xy+zw-*`

In your exam, you can make a table as follows to solve such questions:

Infix: `x + y * z ^ w`

Element	Stack	Postfix	Remark (This is for explanation only)
x		x	Operand added to postfix
+	+	x	+ operator added to stack
y	+	xy	Operand added to postfix
*	+,*	xy	Precedence of * > + so nothing is popped
z	+,*	xyz	Operand added to postfix
^	+,*,^	xyz	Precedence of ^ > *,+
w	+,*,^	xyzw	Operand added to postfix
		xyzw^*+	All elements in stack popped

Let's now look at a slightly more complex expression.

Infix: `x ^ z + (y / w) - a`

Element	Stack	Postfix
x		x
^	^	x
z	^	xz
+	+	xz^
(	+(	xz^
y	+(	xz^y
/	+(/	xz^y
w	+(/	xz^yw
)	+	xz^yw/
-	-	xz^yw/+
a	-	xz^yw/+a
		xz^yw/+a-

Postfix: `xz^yw/+a-`

Note how when `^` was in the stack and `+` was introduced the `^` was popped as it has greater precedence than `+`. A similar case can be seen when `-` was introduced when `+` was in the stack as both have the same precedence.

Evaluating Postfix Expressions

Postfix expressions are evaluated using stacks.

Whenever you have a postfix expression, to evaluate, follow this process:

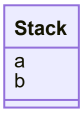
Step 1: Iterate through the expression left to right

Step 2: For every element

```
    If element is operand
        push in stack

    else if element is operator
        perform operation on last 2 elements in stack. Then remove the last 2 elements and push the result in the stack
```

Note: For operations which are **not commutative** (for example subtraction,division,exponent,etc.), you must be mindful of the order of evaluation:

Let the stack be



and the operation is `-`

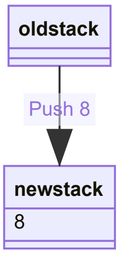
Then perform `b-a`. The element that was in the stack first will be the first operand.

The same logic should be applied for `/`, `^` and all other non commutative operations.

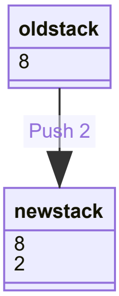
We write the elements separated by commas as to avoid ambiguity (you can also use other delimiters like a whitespace).

Let's evaluate the postfix expression `8,2,+,5,3,-,*,4,/`

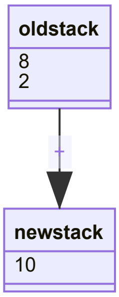
1. `8`
   Push in the stack

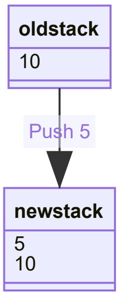
2. `2`
   Push in stack

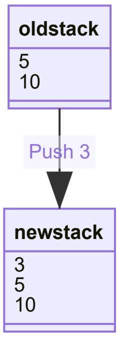
3. `+`
   Perform `2+8` and push to stack



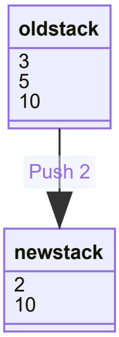
4. `5`
   Push to stack



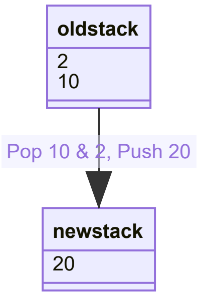
5. `3`
   Push to stack



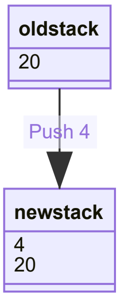
6. `-`
   Perform `5-3` and push to stack. Remember the element that **first went into the stack will be the first operand**



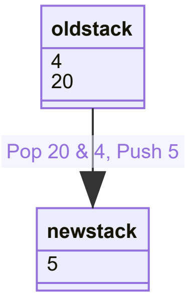
7. `*`
   Perform `10*2` and push to stack



8. `4`
   Push to stack



9. `/`
   Perform `20/4` and push to stack. **Remember to keep in mind order of operation**



Now the stack only has one element, and we have iterated over the entire expression. So the process stops

The final answer is `5`.

In your exams, you will need to write the process in a tabular form.

Let's look at an example of that.

Take the postfix expression `1,5,5,*,5,4,^,/,+`

Element	Stack	Remarks (For explanation only)
1	1	Append operand to stack.
5	1,5	
5	1,5,5	
*	1,25	Pop last two elements and multiply second times first (5 * 5). Push result to stack.
5	1,25,5	
4	1,25,5,4	
^	1,25,625	Pop last two elements and raise second element to the power of the first element Push result to stack.
/	1,(1/25)	Pop last two elements and divide second by first Push result to stack.
+	1 + 1/25 = 1.04	Add the remaining two elements to get the final result

So the result of the expression is `1.04`

NCERT Solutions

1. State TRUE or FALSE for the following cases:
a) Stack is a linear data structure
b) Stack does not follow LIFO rule
c) PUSH operation may result into underflow condition
d) In POSTFIX notation for expression, operators are placed after operands

Ans:

a) True
b) False
c) False
d) True

2. Find the output of the following code:
a)

```
1  result=0
2  numberList=[10,20,30]
3  numberList.append(40)
4  result=result+numberList.pop()
5  result=result+numberList.pop()
6  print(“Result=”,result)
```

b)

```
 1  answer=[]; output=''
 2  answer.append('T')
 3  answer.append('A')
 4  answer.append('M')
 5  ch=answer.pop()
 6  output=output+ch
 7  ch=answer.pop()
 8  output=output+ch
 9  ch=answer.pop()
10  output=output+ch
11  print(“Result=”,output)
```

Ans:

a) In the third line, 40 is added to the list, so the new list is `[10,20,30,40]`

In the fourth line, the last element in the list is added to result, so `result = 40`

In the fifth line, the last element in the list is added to the result, so `result = 70`

So the final output will be: `Result= 70`

b) After executing lines 2-4, `answer = ['T','A','M']`

The code is a simple string reverse code, so the output will be `Result= MAT`

3. Write a program to reverse a string using stack.

Ans:

```python
stack = []

string = input("Enter string: ")

for ch in string:
    stack.append(ch)

while stack != []:
    print(stack.pop(), end="")
```

4. For the following arithmetic expression: ((2+3)*(4/2))+2

Show step-by-step process for matching parentheses using stack data structure.

Element	Stack	Postfix
(	(	
(	((	
2	((	2
+	((+	2
3	((+	2 3
)	(	2 3 +
*	(*	2 3 +
(	(*(	2 3 +
4	(*(	2 3 + 4
/	(*(/	2 3 + 4
2	(*(/	2 3 + 4 2
)	(*	2 3 + 4 2 /
)		2 3 + 4 2 / *
+	+	2 3 + 4 2 / *
2	+	2 3 + 4 2 / * 2

Result: `2 3 + 4 2 / * 2 +`

5. Evaluate following postfix expressions while showing status of stack after each operation given A=3, B=5, C=1, D=4
a) A B + C *
b) A B * C / D *

a)

Element	Stack
3	3
5	3 5
+	8
1	8 1
*	8

Result: 8

b)

Element	Stack3
3	3
5	3 5
*	15
1	15 1
/	15
4	15 4
*	60

6. Convert the following infix notations to postfix notations, showing stack and string contents at each step.
a) A + B - C * D
b) A * ((C + D)/E)

a)

Element	Stack	Postfix
A		A
+	+	
B	+	AB
-	-	AB+
C	-	AB+C
*	-*	AB+C
D	-*	AB+CD
		AB+CD*-

b)

Element	Stack	Postfix
A		A
*	*	A
(	*(	A
(	*((	A
C	*((	AC
+	*((+	AC
D	*((+	ACD
)	*(	ACD+
/	*(/	ACD+
E	*(/	ACD+E
)	*	ACD+E/
		ACD+E/*

7. Write a program to create a Stack for storing only odd numbers out of all the numbers entered by the user. Display the content of the Stack along with the largest odd number in the Stack. (Hint. Keep popping out the elements from stack and maintain the largest element retrieved so far in a variable. Repeat till Stack is empty)

Ans:

```python
def push(stack, element):
    stack.append(element)

def pop(stack):
    if stack == []:
        print("Stack is empty.")
    else:
        return stack.pop()

numbers = input("Enter string of numbers: ").split(",")
stack = []

for num in numbers:
    if int(num) % 2 == 1:
        push(stack, int(num))

stack2 = list(stack)

maxval = stack[0]

for i in stack[::-1]:
    if i > maxval:
            maxval = pop(stack)
    else:
        pop(stack)

print("The stack is", stack2, "and the max odd value is", maxval)
```

Sample Paper Questions

The SQP has 2 questions for a total of 5 marks.

One is a theory based question, the other is a programming based question. This indicates that both components are important for the exam from this chapter.

1. **Give any two characteristics of stacks.** [2]

   Two characteristics of stacks are:

   1. It obeys the Last in First Out (LIFO) rule
   2. Elements are added and removed from the top of the stack

   Other valid points:

   1. It is a linear data structure
2. **Julie has created a dictionary containing names and marks as key value pairs of 6 students. Write a program, with separate user defined functions to perform the following operations:**

   ● Push the keys (name of the student) of the dictionary into a stack, where the corresponding value (marks) is greater than 75.

   ● Pop and display the content of the stack.

   For example: If the sample content of the dictionary is as follows:

   R={"OM":76, "JAI":45, "BOB":89, "ALI":65, "ANU":90, "TOM":82}

   The output from the program should be: TOM ANU BOB OM [3]

```
stack = list()

# 1 mark for push function
def push(stack,element):
    stack.append(element)

# 1 mark for pop function
def pop(stack):
    # make sure to include the if else to account for underflow errors
    if stack != []:
        return stack.pop()
    else:
        return None

R={"OM":76, "JAI":45, "BOB":89, "ALI":65, "ANU":90, "TOM":82}

# 1 mark for output
for key, value in R.items():
    if value > 75:
        push(stack, key)

while stack != []:
    print(pop(stack), end=" ")
```

3. **Alam has a list containing 10 integers. You need to help him create a program with separate user defined functions to perform the following operations based on this list.**

   ● Traverse the content of the list and push the even numbers into a stack.

   ● Pop and display the content of the stack.

   For Example: If the sample Content of the list is as follows:

   N=[12, 13, 34, 56, 21, 79, 98, 22, 35, 38]

   Sample Output of the code should be: 38 22 98 56 34 12

```
stack = []

# 1 mark for push function
def push(stack, element):
    stack.append(element)

# 1 mark for pop function
def pop(stack):
    if stack != []:
        return stack.pop()
    else:
        return None

N = [12, 13, 34, 56, 21, 79, 98, 22, 35, 38]

# 1 mark for displaying output
for i in N:
    if i % 2 == 0:
        push(stack, i)

while stack != []:
    print(pop(stack), end=" ")
```

Analysis

From the questions seen in the sample paper, we can conclude that both theory and programming questions can come in the exam.

Note how one mark is given for defining both the `push` and `pop` functions. Make sure to read the question carefully to see what is being asked and the functions that need to be defined. These are two marks you can easily secur

If a three marker of this kind comes in the exam, you must define the two functions as they are explicitly stated in the answer key. While it says marks are to be awarded for other correct logic, it is in your best interest that you **follow** the format the board uses: define the `push` and `pop` functions and then implement them in the code to solve the problem.

In the theory portion, definition of a stack, properties of a stack, applications of a stack are all important.

You are advised to read all the notes in this book and the NCERT textbook thoroughly to not miss out on key points.

PYQs + Practice

1. **Write PUSH(Books) and POP(Books) methods in Python to add Books and remove Books considering them to act as Push and Pop operations of stack [2015] (4)**
2. **Evaluate the following postfix notation of expression. Show status of stack after every operation**

   84,62,-,14,3,*,+ (2)
3. **Write a program in Python to evaluate a postfix expression separated by commas, inputted by the user, using a stack.**

   Sample input expression: 84,62,-,14,3,*,+ (2)
4. **Write a function in Python, MakePush(Package) and MakePop(Package) to add a new Package and delete a Package from a List of Package Description, considering them to act as push and pop operations of the stack data structure. [2020] (3)**

Solutions

1.

```
def push(Books):
    element = input("Enter book: ")
    Books.append(element)

def pop(Books):
    if Books != []:
        print(Books.pop(), "deleted from stack")
    else:
        return None
```

2.

Element	Stack
84	84
62	84,62
-	22
14	22,14
3	22,14,3
*	22,42
+	64

Answer: 62

3.

```python
stack = list()

def push(stack, element):
    stack.append(element)

def pop(stack):
    if stack != []:
        return stack.pop()
    else:
        return None

expression = input("Enter a postfix expression: ")

for ch in expression.split(","):
    if ch.isdigit():
        push(stack, ch)
    else:
        a = pop(stack)
        b = pop(stack)

        res = str(eval(b + ch + a))
        push(stack, res)

print("Output of the expression is", pop(stack))
```

Output:

```
Enter a postfix expression: 84,62,-,14,3,*,+
Output of the expression is 64
```

4.

```python
Package  = []

def MakePush(Package):
    element = input("Enter Package Name:")
    stack.append(element)

def MakePop(Package):
    if stack != []:
        print("Element", stack.pop(), "removed")

    else:
        return None
```

Chapter 2: Computer Networks

What are computer networks?

A computer network is defined as "an interconnection among two or more computers or computing devices."

When talking about computer networks, the term "node" is often used to refer to a device in the network that sends, receives, stores or creates data.

Benefits of Computer Networks

1. **Cost Effectiveness:** Computer networks allow users to share software, hardware and other expenses, and thus reduce overall costs. For example, if an office floor is connected to one printer, the entire floor can use the printer without having to buy their own.
2. **Reliability:** Data on one device may get corrupted, but if it is stored on multiple devices, there is a higher probability that the data will remain intact overall.
3. **Security:** Computer networks are secure and authorization is required to access them.
4. **Greater Storage Capacity:** Devices can share howallowing for greater total storage in the entire network.
5. **Information Sharing:** Devices in a network can efficiently share information with each other.

Internet and Intranet

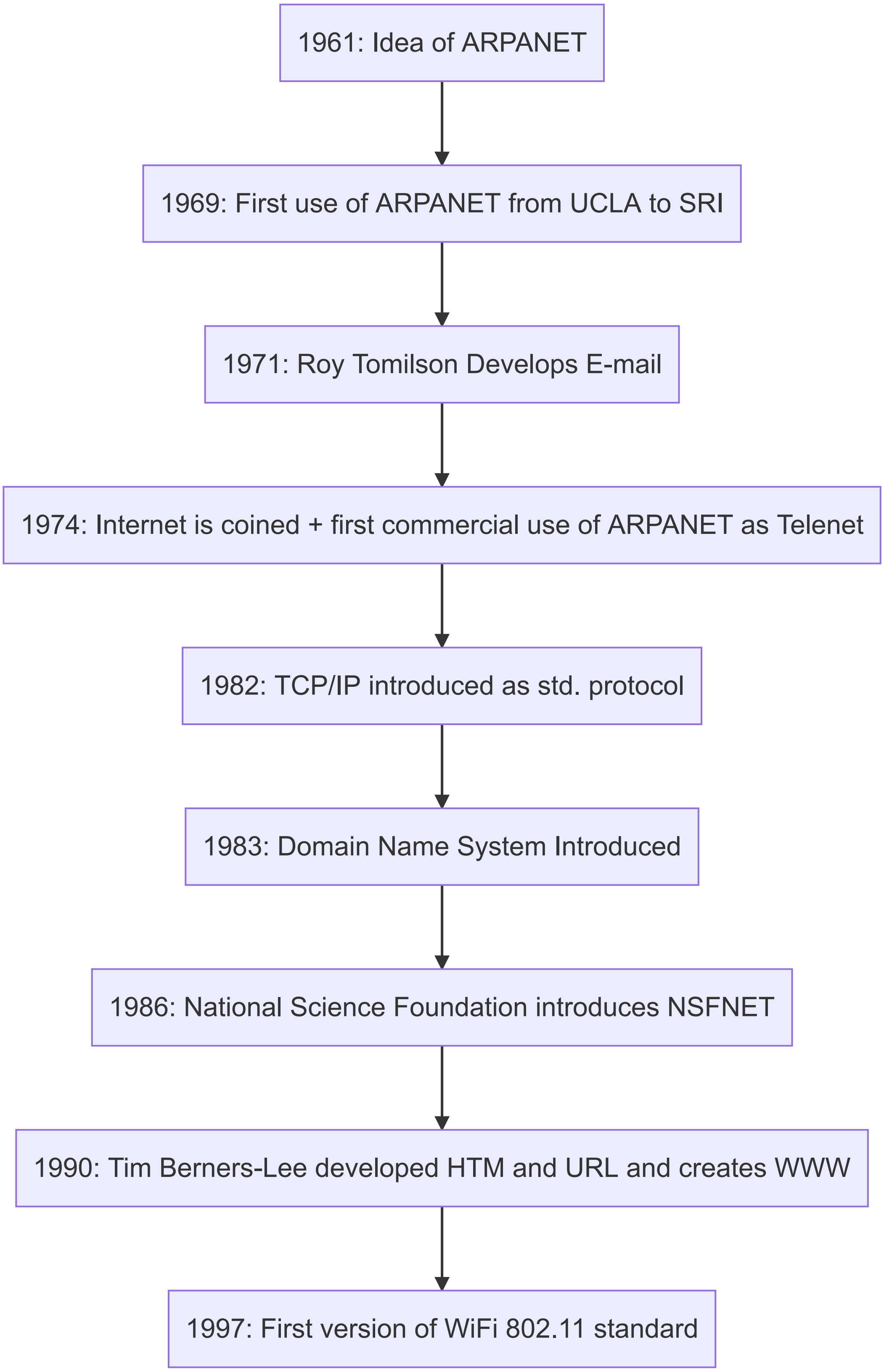
The **internet** refers to a global network of computing devices which can be accessed by anyone with an **internet connection** (Wi-Fi/Mobile Data).

An **intranet** is a private network exclusively created for specific persons or an organization.

Evolution of Networking

ARPANET - Advanced Research Projects Agency Network was created by the US Department of Defence in 1961 to connect many academic and research institutions.

The **first computer network message** was communicated between the University of California, Los Angeles (UCLA) and the Stanford Research Institute (SRI).



1961: Idea of ARPANET

1969: First use of ARPANET from UCLA to SRI

1971: Roy Tomilson Develops E-mail

1974: Internet is coined + first commercial use of ARPANET as Telenet

1982: TCP/IP introduced as std. protocol

1983: Domain Name System Introduced

1986: National Science Foundation introduces NSFNET

1990: Tim Berners-Lee developed HTM and URL and creates WWW

1997: First version of WiFi 802.11 standard

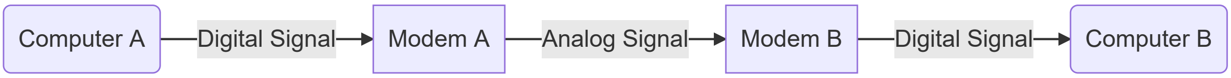
Types of Networks

The following table summarises the main information about the different types of networks. Using this table you will be able to write about each kind individually and also differentiate between any two.

Network	Description	Example	Range	Connection Medium	Network Speed
PAN (Personal Area Network)	A network formed by connecting a few personal devices like computers, laptops, mobile phones, smart phones, printers etc	A network of your phone, laptop and printer.	10m	Wired or wireless	
LAN (Local Area Network)	Connects computers, mobile phones, tablet, mouse, printer, etc. within a limited range.	A school computer lab, or office building.	1km	Ethernet cable, Fibre-optics or WiFi	10 Mbps - 1000 Mbps
MAN (Metropolitan Area Network)	An extended form of LAN which covers a larger geographical area like a city or a town.	Cable TV, Broadband internet	30-40km	Fibre Optics, Broadband	In Mbps, but less than LAN
WAN (Wide Area Network)	Connects computers and other LANs and MANs, which are spread across different geographical locations of a country or in different countries or continents.	Connection of different branches of a business or government organization.	>40km	Satellites, Radio Antennae	

Network Devices

1. **Modem**
   - Modem, which stands for **MO**ulator **DEM**odulator is a device used to convert analog signals to digital bits (binary format of 0s and 1s).
   - The following diagram should help illustrate the transmission of data from one device to another.



   - Modem A acts as a **modulator** and converts the digital signal from Computer A to an analog one.
   - Modem B acts as a **demodulator** and converts the analog signal from Modem A to a digital one for Computer B to receive.
2. **Ethernet Card**
   - It is a network adapter that is used in a wired network and acts as the interface between a computer and the network.

- It is also known as NIC (Network Interface Card).
- The ethernet card is a circuit board that is mounted on the computer board and has a unique identifier known as its MAC (Media Access Control) address.
- The ethernet card connects to the ethernet cable and supports data transfer from 10 Mbps to 1 Gbps

3. **RJ45**
    - RJ45 stands for "Regular Jack-45".
    - It has 8 pins and acts as a small plastic plug to connect ethernet cables to the ethernet card.
    - You can check the wire if you have a wired connection to see the endpoint which should be an RJ45.
4. **Repeater**
    - Let's say you had to communicate with a friend 100 metres away. You could shout their name but they wouldn't be able to hear you since the sound you produce will become quite faint along the way. Instead, if you had two friends in between you and your distant friend, you could shout to them and they would pass along your message by shouting it to each other.
    - This is what a repeater does: it regenerates a weak analog signal to allow it to be transmitted at large distances and not die out.
    - After **100** metres, signals lose their strength and repeaters ensure that the signal continues to be transmitted by regenerating the original signal and it is thus kept alive.
5. **Hub**
    - A hub is used to connect multiple wired devices.
    - Data coming from one connection to the hub is sent to all connected nodes.
    - In the event that two nodes transmit data to the hub, their signals will interfere with one another.
6. **Switch**
    - The switch connects several devices in a LAN. It is used to transmit data to particular nodes in the network.
    - When the switch receives the data, it identifies the destination address and looks it up in a table to identify the receiving node.
    - Unlike the hub which sends the data to all connected devices, the switch sends information only to the desired destination and can even send multiple signals at the same time.
    - It doesn't send signals which are noisy or corrupted and drops them, asking the sender to resend.
    - **Usage:** In homes and offices to LANs or to access the internet.
7. **Router**
    - It receives, analyses and transmits data.
    - It connects a LAN to the internet.
    - It is more advanced than a hub or switch since it can analyse the data: if the received data packet sizes are too large for them to be transmitted to a different network, they can be repackaged by the router.
    - There are two types of routers:
        - Wireless: They provide access to Wi-Fi and are common household items
        - Wired
    - Today, routers act as a modem and a switch: they convert analog signals provided by the ISP to digital ones for internet connectivity.
8. **Gateway**

- A gateway is the portal between a private network and the internet.
- All data coming or going from the network must pass through the gateway.
- Gateways also maintain information about the host network's internal connection paths and the identified paths of other remote networks.
- At home, your internet service provider acts as the gateway to the internet.
- The router works as a gateway device.
- A firewall is usually implemented in the gateway to prevent malicious attacks on networks.

9. **WIFI Card**

Network Topologies and Types

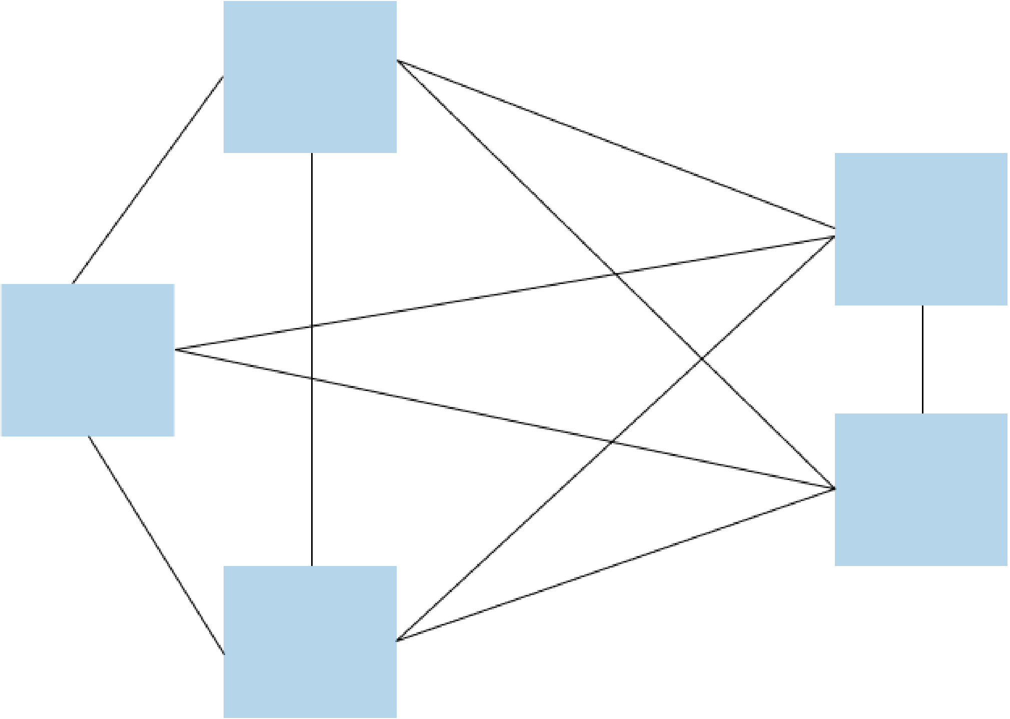
A Network Topology is an arrangement of computers and other devices (called nodes) in a network.

They are broadly classified into these categories

1. **Mesh**

   Arrangement: Every node is connected to every other node in the network, i.e. every possible pair in this topology has a connection

   A fully connected Mesh Topology will have ${}^nC_2 = \frac{n(n-1)}{2}$ connections, if there are n nodes.



   Pros:

   - Network can handle large amounts of traffic since multiple devices can transmit data simultaneously
   - Failure of a single node will not dismantle the network
   - High security and privacy
   - Easy to identify faulty nodes

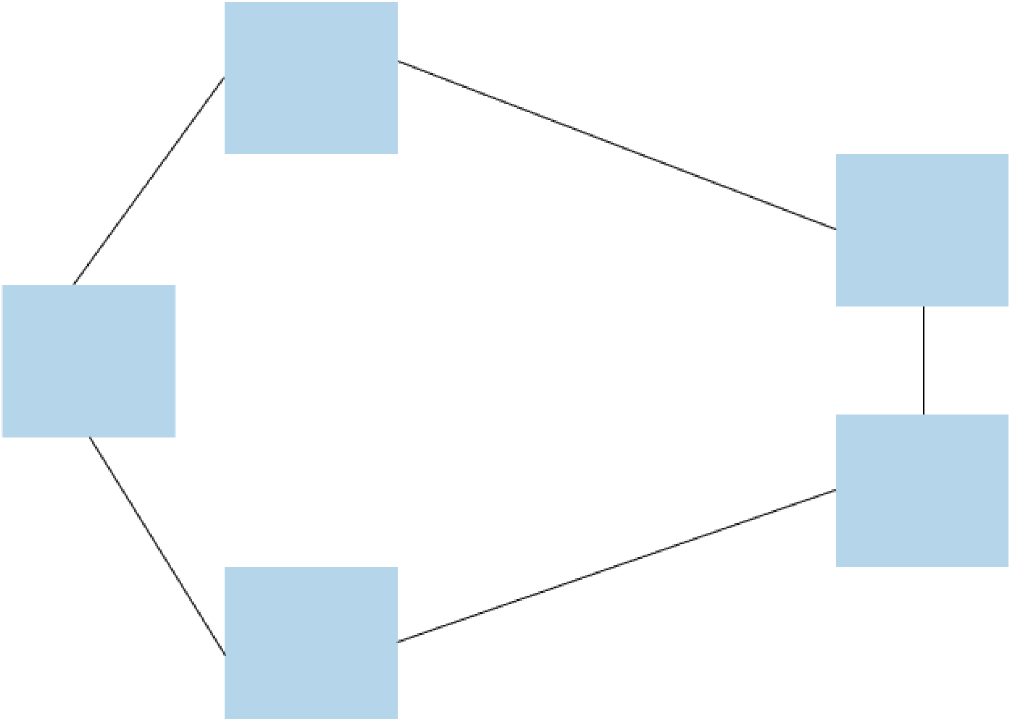
   Cons:

   - Costly set-up, since many cables are required
   - Many redundant and unutilized connections as each and every node is interconnected
   - The wiring becomes very complex

- Very high power requirement since every node needs to stay active to share load of the network

2. **Ring**

   Arrangement: Every node is connected to its two neighbouring/adjacent nodes.



   Advantages:

   - Unidirectional flow of data reduces chances of collision/interference
   - Low-cost and simple setup
   - Additional nodes can be installed without disrupting the network

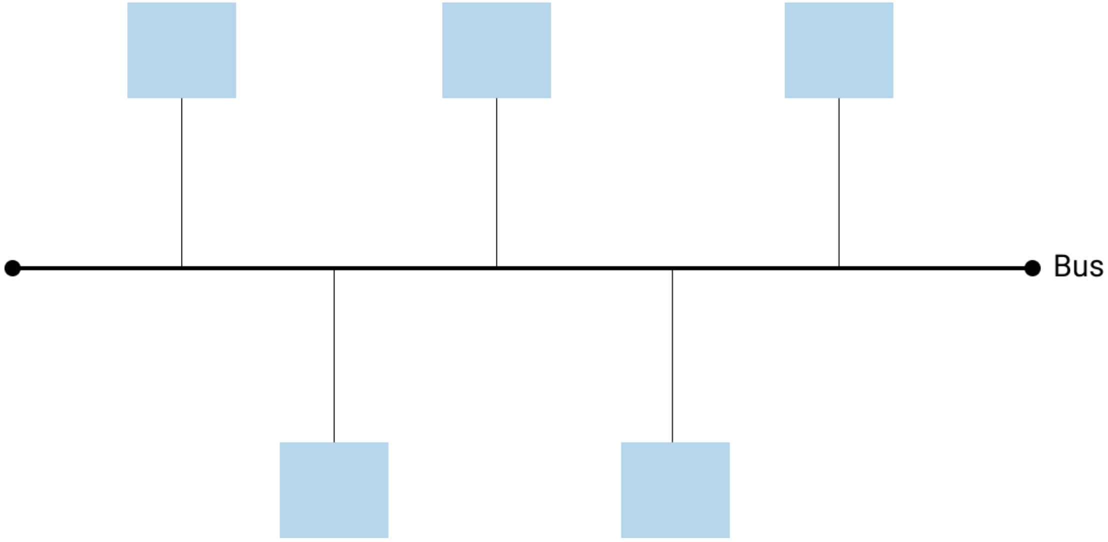
   Disadvantages:

   - Data flow is unidirectional, so it can take time to transmit data from one node to another
   - Failure of any node can disrupt the entire network

3. **Bus**

   Arrangement: Each node is connected to a common transmission medium, called a "Bus". Data transmission happens through this Bus.

   Working: Data from any node reaches the other nodes via the bus in both directions.



Advantages:

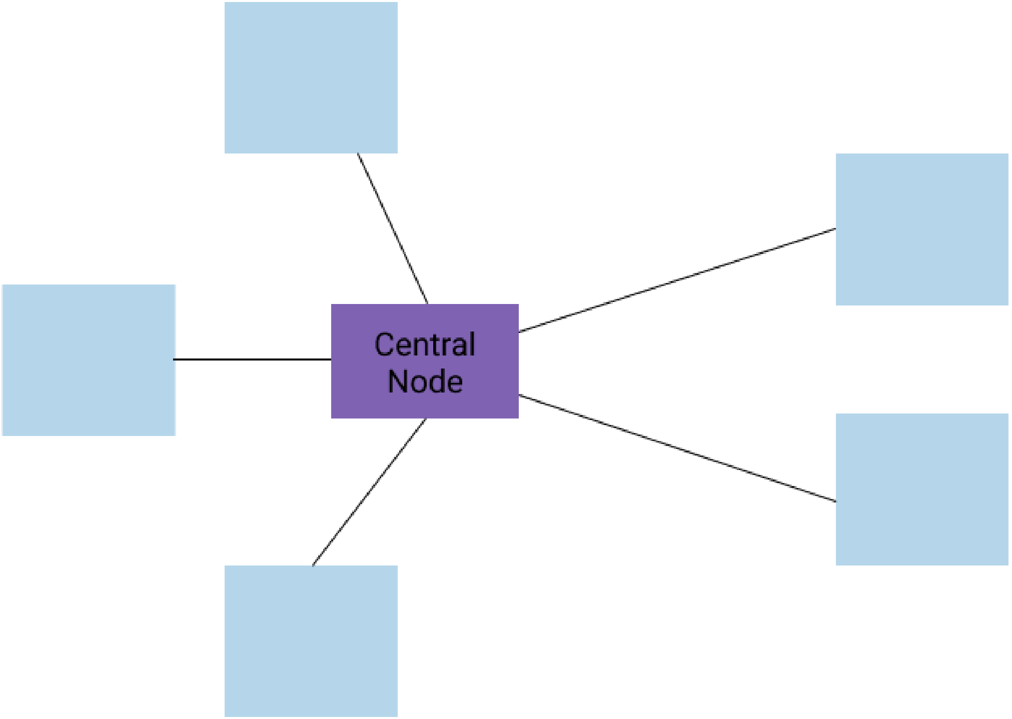
- Very cost effective since a single wire is connected to all the nodes
- It is the easiest topology to use to connect computers in a linear arrangement (like in computer labs)
- It is easy to connect or remove devices without affecting the entire network
- Failure of any one node does not break the network

Disadvantages:

- Less secure and less reliable, since all the nodes receive a signal sent from a source.
- Damage of the bus wire leads to network failure, as it is the main transmission medium

4. **Star**

   Arrangement: Each node is connected to a central node (which can be a hub or switch)



Advantages:

- Very effective, efficient and fast since all nodes are connected to the central device
- Failure of any non-central node will not affect the rest of the network
- High performance since no data collision can occur

Disadvantages:

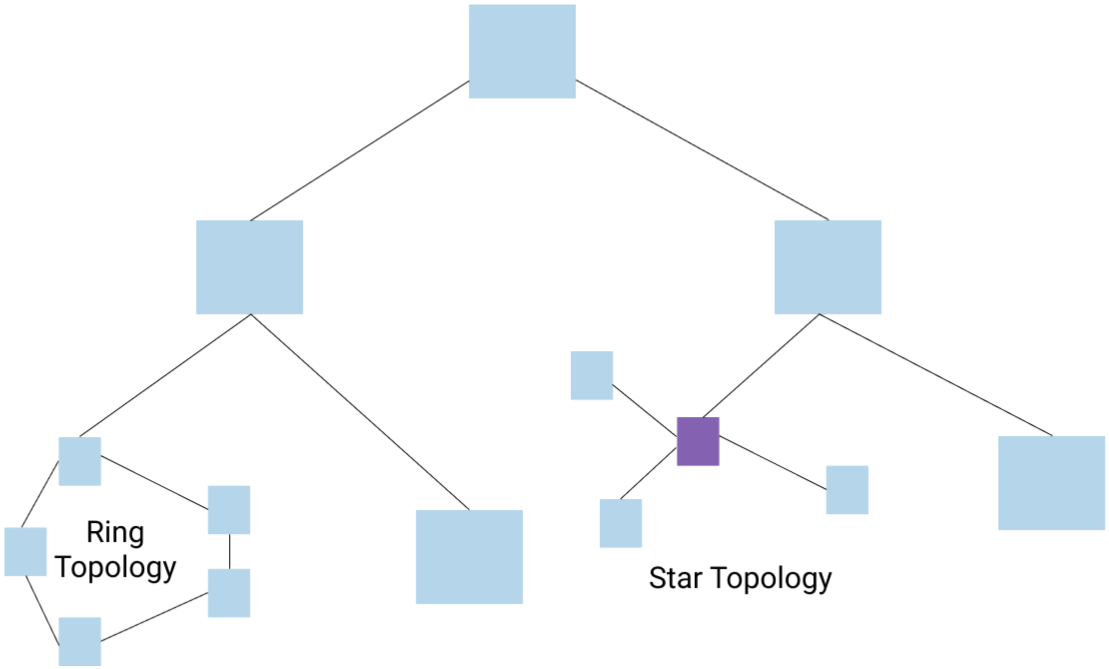
- Failure of central node will cause network failure
- Extra hardware is required (hub or switch) which adds to the cost

5. **Tree/Hybrid**

   Arrangement: Hierarchical topology with multiple branches. Each branch can have more basic topologies like ring, star, bus.

   Usage: Such topologies are used in WAN, connecting multiple LAN's. The LAN's may be in the form of ring, bus or star.

   Working: Data from source first reaches central device and passes through every branch where branches have links to more nodes.



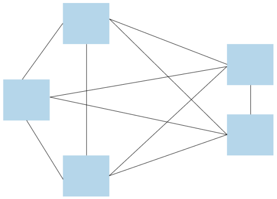
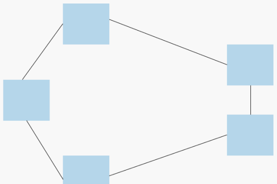
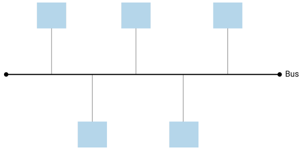
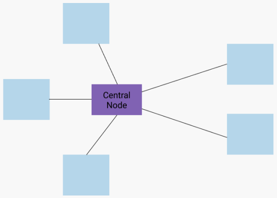
Advantages:

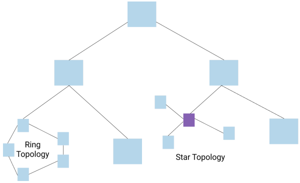
1. Additional nodes can be easily installed
2. Easy to identify faulty nodes
3. Failure of one node does not affect the network

Disadvantages:

1. High setup cost due to wires
2. If central device fails, the network fails

Summary

Topology	Diagram	Arrangement	Advantages	Disadvantages
Mesh		Each node is interconnected with every other node in the network	Fast transmission, High security and privacy, Failure of one node will not affect the network, Easy to identify faulty nodes	High set up cost, Redundant connections, High power requirement as all nodes must be active
Ring		Each node is connected to the adjacent two nodes in the network	Low setup cost, Unidirectional data transmission which implies less risk of collision, Additional nodes can be easily installed	May take time to transmit data, Failure of one node can disrupt the network
Bus		A common transmission medium called a bus connects to each node in the network	Low setup cost, Fast transmission, Additional nodes can easily be added, Failure of one node does not affect the network	Less secure as all nodes receive the data, Damage of the bus wire causes network failure
Star		Each node in the network is connected to a central node such as a hub/switch	Additional nodes can be easily installed, Fast transmission, No data collision	Failure of central node causes network failure, Additional cost due to Hub/Switch at centre

Topology	Diagram	Arrangement	Advantages	Disadvantages
Tree		A central device receives data which is passed down to the branches in the network. It consists of multiple topologies	Additional nodes can be easily installed, Easy to identify faulty nodes, Failure of one node does not affect the network	High setup cost, If central device fails, the network fails

Some Tricks to Remember the Properties of Topologies

1. Know the diagrams as it will be easy to identify the advantages and disadvantages from the diagram
2. If the topology contains a central device, it is likely to fail because of faulty central device
3. If there are many connections, the setup cost will be high and vice-versa
4. If two nodes are connected directly, data transmission is secure and collision free
5. If data has to pass through multiple nodes, transmission time is increased and vice-versa

Identifying Nodes

Each node has certain characteristics which make it unique:

1. **MAC Address**
   - Stands for Media Access Control.
   - Each device has a hardware identifier, associated with its NIC (Network Interface Card).
   - The MAC Address is permanent and unique, and is engraved at the time of manufacture.
   - The MAC address has 12 hexadecimal digits (48 bits). The first 6 contain a unique ID known as Organizational Unique Identifier (OUI) and the second 6 digits are a serial number.
   - Let's say an organization has an OUI `A2:E7:B3`.
   - If it makes two devices, one with the serial number `AB:43:A1` and the other with `AB:43:A2`, their respective MAC addresses would be `A2:E7:B3:AB:43:A1` and `A2:E7:B3:AB:43:A2`.
2. **IP Address**
   - Stands for Internet Protocol address.
   - It is another unique identifier to identify each node in a network that uses IP communication.
   - Unlike MAC, IP address is **not permanent**. If you take a laptop and connect it to your home Wi-Fi, it will show a particular address, however, if you take it to an office and connect it to their Wi-Fi, the IP address will change.
   - **IPV4**
     - 32 bits
     - 4 numbers separated by periods
     - Each number represents an 8 bit binary in decimal form.
     - Just under 4.3 billion possible unique addresses
     - Sample: `192.68.0.24
   - **IPV6**

        - Created to allow more unique addresses.
        - 128 bit identifier.
        - 8 hexadecimal numbers separated by colons.
        - Sample: `1231:ABCD:0000:0000:0000:0000:A234:B312`

The Web and Internet

We have all browsed various websites on the internet. We have a variety of devices nowadays that can connect to the internet.

We are able to connect to the internet through our local Internet Service Provider (ISP). ISPs connect with each other to form regional networks, which further connect to form national networks and the connection of networks in different countries forms the internet.

The internet has also given rise to a lot of new services such as e-learning (like Q-Programming! (https://qprogramming.net)), e-commerce, streaming sites, etc.

IOT

The Internet of Things refers to a network of smart electronic appliances along with other computing devices. It can include TVs, ACs, refrigerators, fans, etc.

A lot of IOT networks can be controlled by your smartphone. These days smart homes are becoming a reality where you can control your appliances and devices with the press of a button.

The World Wide Web

Also known as WWW or web, the World Wide Web consists of trillions of web pages and content which can be accessed by anyone with an internet connection.

Sir Tim Berners-Lee invented the WWW in 1990 with three important technologies:

1. **HTML:**
    - HyperText Markup Langauge.
    - It is used to design webpages and provides a standard structure that can be read by any device.
2. **URI/URL**:
    - Uniform Resource Identifier/Locator.
    - It is a unique path to any resource on the web.
    - Components of a URL:
        - http:// or https://
        - Domain name: eg. qprogramming.net
        - Specific resource: eg `https://qprogramming.net/cs4a` is the URL for the CS4A series!
3. **HTTP:**
    - HyperText Transfer Protocol
    - Rules used to retrieve web pages
    - HTTPS is a secure version with encryption (the S stands for secure)

Web v/s Internet

These terms are used interchangeably but what is the actual difference?

The internet is the global network of **computers** which may or may not have resources to share with the other nodes of the network.

The web is the global network of **resources** accessible to anyone through the internet.

Web Hosting

Web hosting is the storage of web pages and resources on a server in order to be accessed by users with an internet connection.

Website

A website is a collection of web pages which contain hyperlinks to each other. Websites can be accessed by a user through the internet.

Web Server

A web server stores web resources for access through the internet through the HTTP protocol.

Web Browser

A web browser is software used to navigate the internet and retrieve web pages through HTTP requests to web servers. It acts as the client in the client-server model.

XML

XML stands for Extensible Mark-up Language and is used to store metadata about websites such as the website structure.

DNS

We know that each website is hosted on a server. This server acts as a node in the internet and has a unique IP address.

But think about it, can you remember the IP address of every site?

Probably not, that's why we have the Domain Name System!

The domain name system, DNS for short, makes it easier to access resources on the internet by creating an alias for each IP address.

This means that while each website has an actual IP address, we can access the resources through the alias, known as the domain.

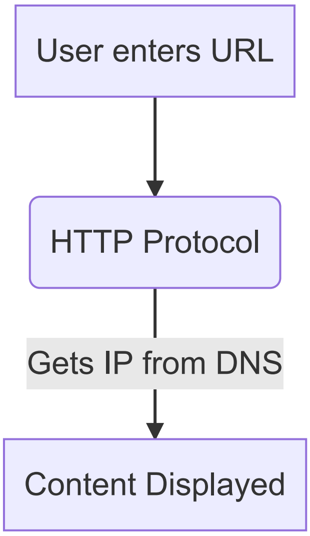
Think of it as your contacts app. You save the number of your friend by their name, and when you want to call them, you type their name and their number appears. This is the same thing!

So, qprogramming.net has an IP address, and you can access its resources by simply typing our domain name rather than our IP!

DNS Server

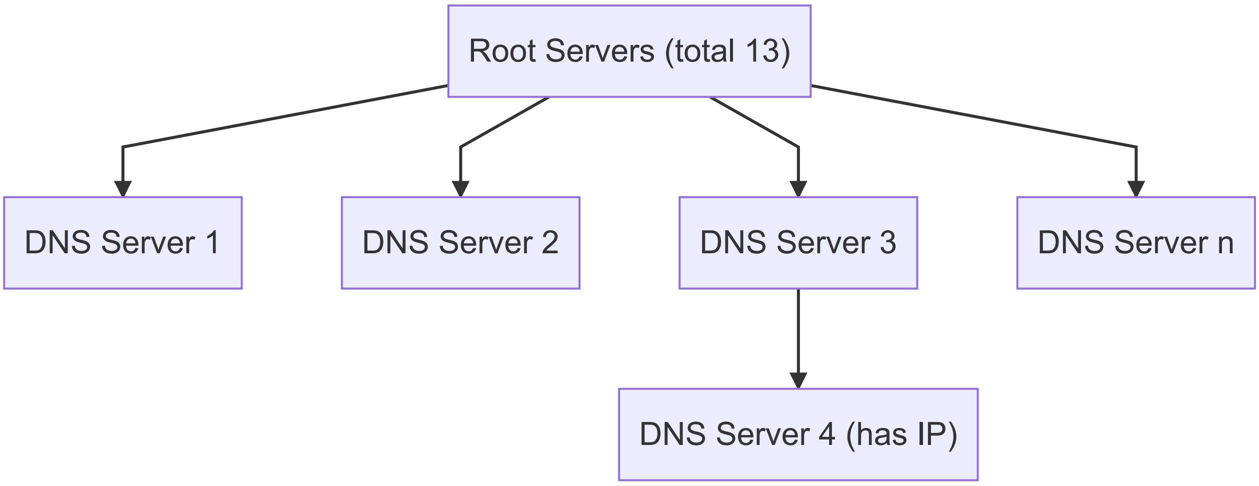
The DNS server is the machine that actually converts domain names to the IP addresses. This process is known as domain name resolution.

How does it work? Let's understand through the following flowchart:



The DNS server has a database of domain names and their respective IP addresses.

The DNS servers are kept in the following hierarchy:



So, when we enter the URL, the HTTP(s) goes to the root servers which point it to the lower level servers until the IP address is found.

In the above example, the order will be Root > DNS Server 3 > DNS Server 4.

The DNS server may contain the IP address of the domain, or the IP address of another server which has the desired in

NCERT Solutions

1. **Expand the following: a) ARPANET b) MAC c) ISP d) URI**

   a) Advanced Research Project Agency Network

   b) Media Access Control

   c) Internet Service Provider

   d) Uniform Resource Identifier

2. **What do you understand by the term network?**

   A network refers to a system of interconnected computing devices which receive, analyze, store and transmit data to each other.

3. **Mention any two main advantages of using a network of computing devices.**

   Two advantages of using networks are:

   a) Cost effectiveness: licensed software can be accessed by multiple computers on the same network, multiple devices can share one printer.

   b) Security: computer networks require specific login credentials to access resources on the network.

4. **Differentiate between LAN and WAN.**

LAN	WAN
Stands for Local Area Network	Stands for Wide Area Network
Network range is approximately 1km	Network range is greater than 40 km
Example: Office floor network	Example: Network of different government branches

5. **Write down the names of few commonly used networking devices**

   Some common networking devices are:

   a) Modem

   b) Router

   c) Hub

   d) Switch

6. **Two universities in different States want to transfer information. Which type of network they need to use for this?**

   The universities need a WAN as it covers ranges greater than 40km and across multiple states and countries

7. **Define the term topology. What are the popular network topologies?**

   A topology refers to the arrangement of devices in a computer network. Some of the popular topologies are:

   a) Mesh

   b) Ring

   c) Bus

d) Star

e) Tree

8. **How is tree topology different from bus topology?**

Tree	Bus
Data from the source first reaches the centralised device and then is passed through every branch	Data transmitted from the source is transmitted along the shared transmission media to all connected devices
Can contain multiple interconnected buses	Only one bus topology
High setup cost	Low setup cost

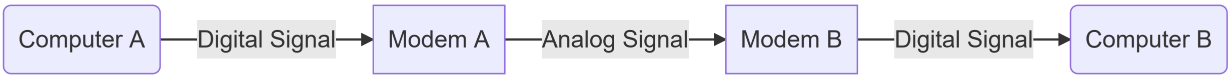
9. **Identify the type of topology from the following:**
a) Each node is connected with the help of a single cable.
b) Each node is connected with central switching through independent cables.

a) Bus topology

b) Star topology

10. **What do you mean by a modem? Why is it used?**

Modem stands for modulator demodulator. It is used to convert both analog signals to digital bits and digital bits to analog signals. This allows a signal to be transmitted from one device to another as follows:



11. **Explain the following devices: a) Switch b) Repeater c) Router d) Gateway e) NIC**

a) A switch is responsible for relaying information to the desired destination address. It can connect to multiple nodes in a network and receive data from one node to transmit to particular nodes. It doesn't send signals which are noisy or corrupted and drops them, asking the sender to resend.

b) A repeater regenerates a signal to amplify it. Signals tend to lose strength after 100 metres so repeaters are necessary to transmit signals across large distances.

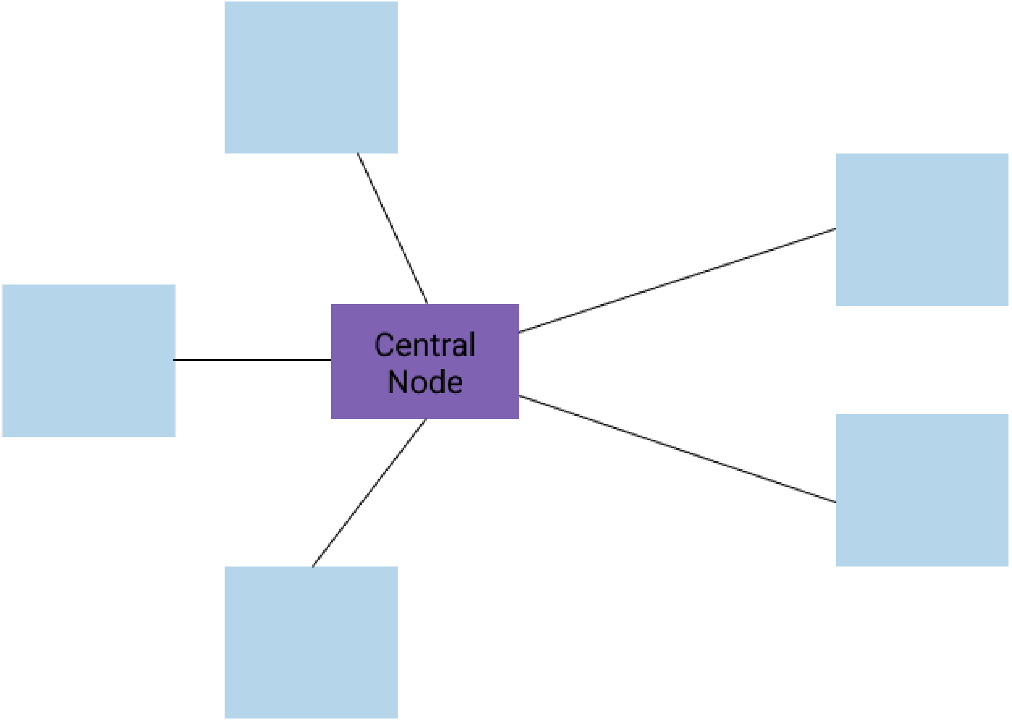
c) Routers are responsible for connecting devices to the internet. They act as both a modem and a switch by converting analog signals from the internet service provider to digital bits to access the internet. They can also analyze data they receive and perform the task of repackaging. In case they receive packets which are too large to transmit to another node in the network, they repackage them and create smaller packets for transmission.

d) Gateways are the portal between the internet and a private network. All incoming and outgoing data must pass through it. They are often equipped with firewalls to prevent malicious attacks to the nodes on the private network. The internet service provider acts as the gateway for home networks.

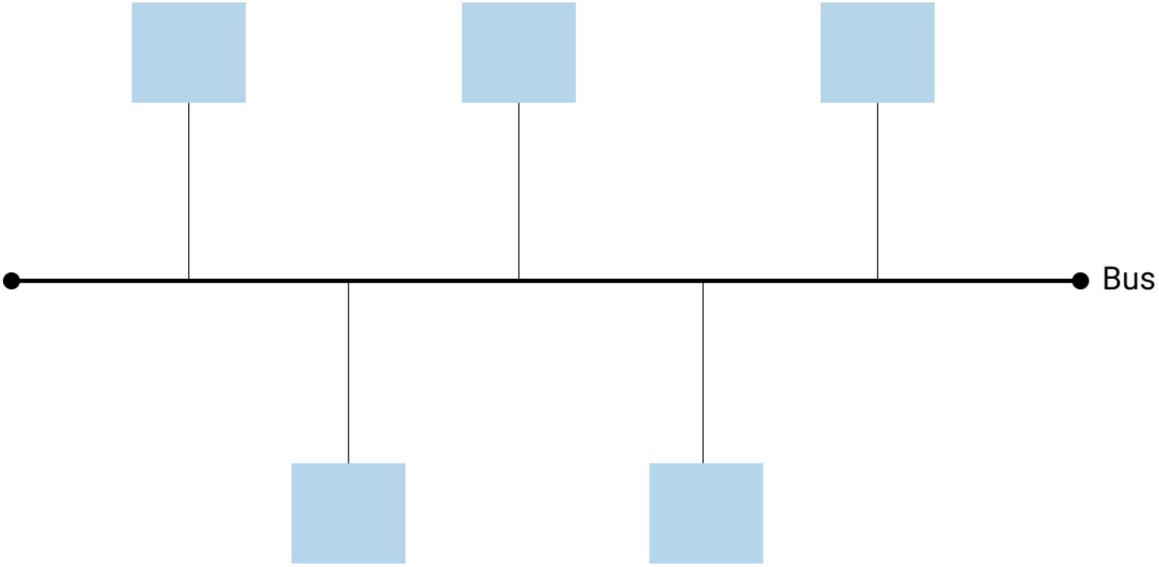
e) NIC stands for Network Interface Card and is an essential part of any internet connected device. It is a network adapter that acts as the interface between a network and the computer. Each NIC has a unique identifier engraved during manufacture called the Media Access Control Address.

12. **Draw a network layout of star topology and bus topology connecting five computers.**

a) Star topology



b) Bus topology



13. **What is the significance of MAC address?**

    The Media Access Control (MAC) address is a unique identifier for the network interface card of a computer. It is useful in identifying nodes in a network as the MAC address is a permanent address engraved to the card at the time of manufacture, and thus can be used to unambiguously identify a node.

14. **How is IP address different from MAC address? Discuss briefly.**

IP Address	MAC Address
Stands for internet protocol address	Stands for media access control address
The address is associated with the entire computer	The address is associated with the ethernet card of the computer
The address is given when the node is connected to the network	The address is engraved at the time of manufacture
The address will change if the node is connected to a different network	The address will remain permanent

15. **What is DNS? What is a DNS server?**

    DNS stands for Domain Name System. It is a system created to allow users to easily access content on the world wide web by referencing the IP Address of a server with a domain name such as ncert.nic.in. This makes it easier to remember the location of the resource.

    A DNS server performs the task of pointing a user to the IP (Internet Protocol) address when they enter the URL (Uniform Resource Locator) of a resource. When the user enters a URL, the HyperText Transfer Protocol (HTTP) accesses the DNS server to retrieve the IP address. Internally, the DNS server may contain the specific IP or a reference to the DNS server containing the IP address, and will search through the connected DNS servers to find the desired address.

16. **Sahil, a class X student, has just started understanding the basics of Internet and web technologies. He is a bit confused in between the terms “World Wide Web” and “Internet”. Help him in understanding both the terms with the help of suitable examples of each.**

World Wide Web	Internet
The network of resources available to anyone with an internet connection	The world wide network of computing devices with or without public resources
Example: qprogramming.net is part of the WWW	Example: the device you are using to read this PDF is a part of the internet

Sample Paper Questions

1. **Give two advantages and two disadvantages of star topology (2)**

   The advantages of star topology are:

   a) Easy to add additional nodes to the network

   b) Fast data transmission

   The disadvantages of star topology are:

   a) If the central node fails, the network fails

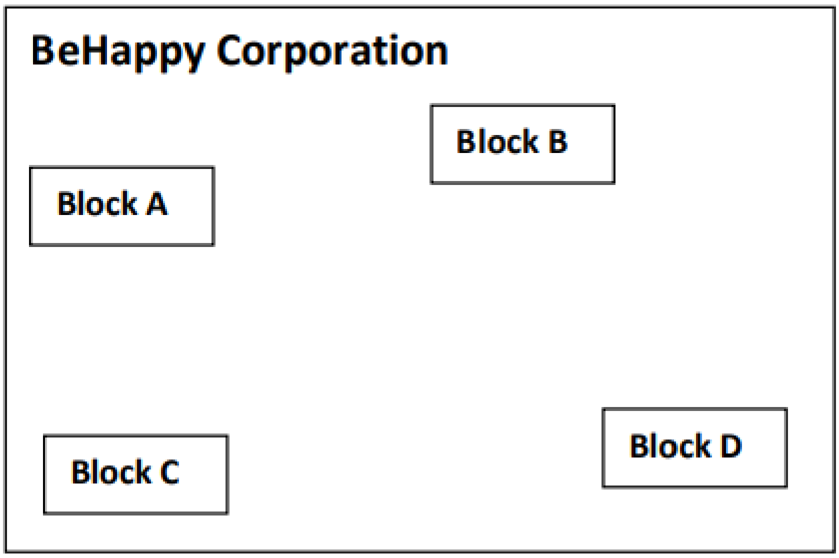
   b) Central hub/switch increases cost

2. **Define the following terms: www, web hosting (2)**

   a) www stands for World Wide Web, which is the web of resources available to anyone with a connection to the internet.

b) Web hosting is the process of storing web pages and resources on a server to be accessed by anyone with an internet connection.

3. **BeHappy Corporation has set up its new centre at Noida, Uttar Pradesh for its office and web-based activities. It has 4 blocks of buildings.**



Distance between the various blocks is as follows:

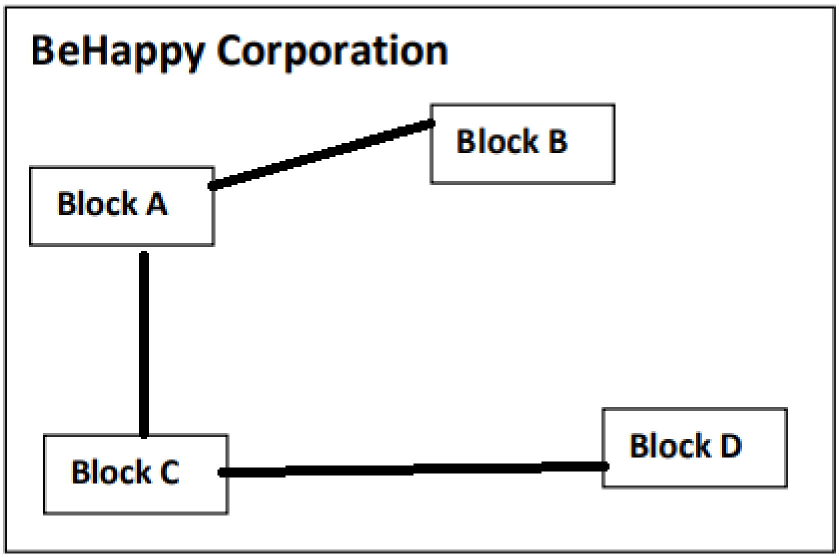
A to B	**40 m**
B to C	**120m**
C to D	**100m**
A to D	**170m**
B to D	**150m**
A to C	**70m**

Numbers of computers in each block

Block A	**25**
Block B	**50**
Block C	**125**
Block D	**10**

(a) Suggest and draw the cable layout to efficiently connect various blocks of buildings within the Noida centre for connecting the digital devices.
(b) Suggest the placement of the following device with justification
i. Repeater
ii. Hub/Switch
(c) Which kind of network (PAN/LAN/WAN) will be formed if the Noida office is connected to its head office in Mumbai?
(d) Which fast and very effective wireless transmission medium should preferably be used to connect the head office at Mumbai with the centre at Noida?

a)



b) i) The repeater should be placed at D as it is 100 m away from the central node

ii) The Hub/Switch should be placed in Block C as it has the maximum number of computers and can acts a the server

c) It will be a WAN as it is connecting two cities and the distance is greater than 40 km.

d) Satellite Communication through microwaves

Analysis

The sample paper contains questions from topologies and definitions of terms. This indicates that the student must be well versed with all the technical terms mentioned in the chapter as well as the advantages/disadvantages of various topologies in addition to their diagrams. Types of networks are also important as they can be asked in application type questions.

The 4 marker question has appeared in different forms in previous year papers as well. Some tricks to solve this are given below:

a) The location with maximum number of computers acts as the server and contains the hub/switch

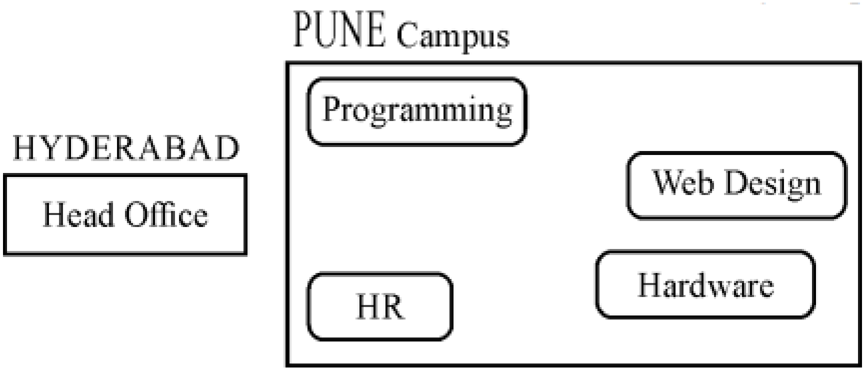
b) The router must be placed at locations >100 m from the server

c) The layout must be drawn such that the cable length used is minimum. First identify the server and then connect the remaining portions keeping length in mind. If possible, name the topology as well.

PYQs

1. **Go-Fast corporation is a Hyderabad based company, which is planning to set up training campuses in various cities in next 3 years. Their first campus is coming up in Pune. At Pune campus, they are planning to have 4 different blocks for HR, Web Design Training, Programming Training and Hardware Training. Each block has number of computers, which are required to be connected in a network for communication, data and resource sharing. As a network consultant of this company, you have to suggest the best network related solutions for them for issues/problems raised in (i) to (iv), keeping in mind the distances between various blocks/locations and other given parameters.**

   Shortest distances between various blocks/locations :



Programming Block to HR Block 60 metres
Programming Block to Web Design Block 50 metres
Programming Block to Hardware Block 70 metres
HR Block to Web Design Block 120 metres
HR Block to Hardware Block 85 metres

HYDERABAD Head Office to PUNE Campus 504 Km

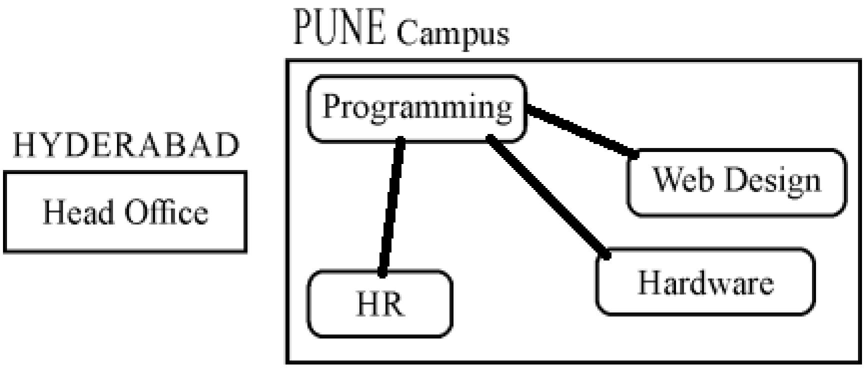
Number of Computers installed at various blocks are as follows :
HR Block 10
Programming Block 100
Web Design Block 60
Hardware 40

(i) Suggest the most appropriate block/location to house the SERVER in the PUNE Campus (out of the 4 blocks) to get the best and effective connectivity. Justify your answer. (1)
(ii) Suggest a device/software to be installed in the PUNE Campus to take care of data security. (1)
(iii) Suggest the best wired medium and draw the cable layout (Block to Block) to efficiently connect various Blocks within the PUNE campus. (1)
(iv) Suggest a device and the protocol that shall be needed to provide Video Conferencing solution between PUNE Campus and Hyderabad Head Office. (1)

i) The programming block should house the server as it has the maximum number of computers

ii) An antivirus program can be installed to prevent malicious attacks by malware.

iii) Optical fibre is the best wired medium for the following layout:



iv) A WiFi card is required and Voice over Internet Protocol (VoIP) should be used for video conferencing.

2. **Name two private ISPs in India (2)**

   a) Airtel

   b) Vodafone

3. **Assume that 50 employees are working in an organization. Each employee has been allotted a separate workstation to work. In this way, all computers are connected through the server and all these workstations are distributed over two floors. In each floor, all the computers are connected to a switch. Identify the type of network?**

   The network is a Local Area Network (LAN).

4. **Software Development Company has set up its new center at Raipur for its office and web based activities. It has 4 blocks of buildings named Block A, Block B, Block C, Block D.**

   Number of Computers

   Block A 25
   Block B 50
   Block C 125
   Block D 10

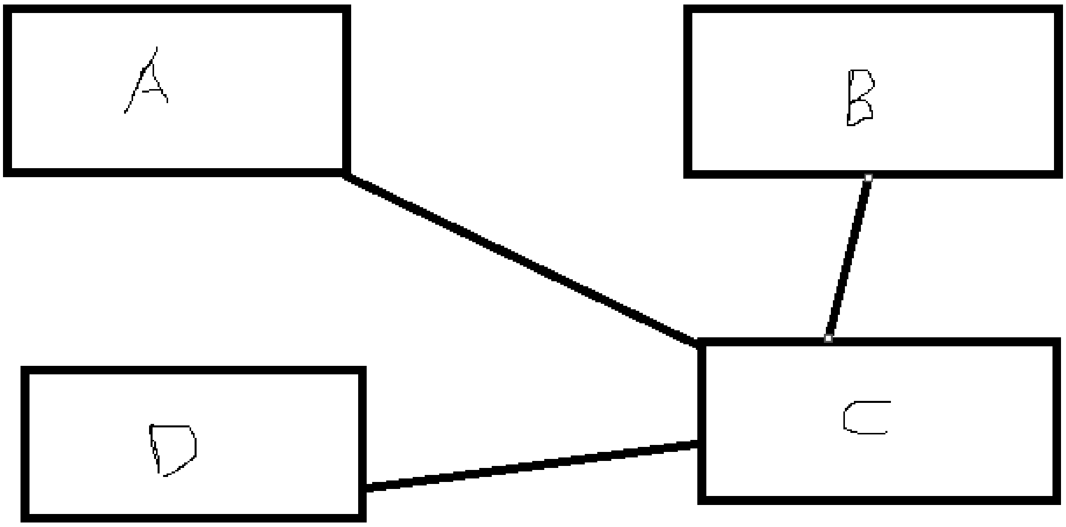
   Shortest distances between various Blocks in meters:
   Block A to Block B 60 m
   Block B to Block C 40 m
   Block C to Block A 30 m
   Block D to Block C 50 m

   (i) Suggest the most suitable place (i.e. block) to house the server of this company with a suitable reason.
   (ii) Suggest the type of network to connect all the blocks with suitable reason .
   (iii)The company is planning to link all the blocks through secure and high-speed wired medium. Suggest a way to connect all the blocks.
   (iv) Suggest the most suitable wired medium for efficiently connecting each computer installed in every block out of the following network cables:

   a) Coaxial Cable
   b) Ethernet Cable
   c) Single Line Telephone Cable

   i) Block C as it has the most number of computers.

   ii) A LAN would be suitable given the short distance between the blocks iii)



iv) Ethernet cable provides the fastest and most secure wired medium.

5. ___ **is a device that forwards data packets along a network.**

   Router is a device that forwards data packets along a network.

Chapter 3: Data Communication

Data communication refers to the exchange of data between nodes in a network.

Components of Data Communication

1. **Sender:** The sender is a node in the network which transmits a message to another node in the network.
2. **Receiver:** The receiver is a node which can receive data from other nodes in the network.
3. **Message:** It is the data that is being transmitted through the network.
4. **Communication Media:** It is the path through which a message is transmitted. For example, an ethernet cable, telephone wire or satellite link are communication media used to transmit various kinds of signals.
5. **Protocols:** They are the rules that must be followed by communication devices for reliable data communication. Example: HTTP, Ethernet.

Measuring Capacity of Communication Media

Bandwidth

- **Definition:** The range of frequencies available for transmission of data through that channel. Bandwidth is the difference of **maximum** and **minimum** frequency contained in the composite signals.
- It is measured in Hertz (Hz)
- 1 Kilohertz (KHz) = 1,000 Hz
- 1 Megahertz (MHz) = 1,000 KHz = 1,000,000 Hz

Data Transfer Rate

- Data transfer rate is the number of bits transmitted between source and destination in one second.
- It is also known as **bit rate**.
- 1 Kbps = 2^{10} bps = 1024 bps
- 1 Mbps = 2^{20} bps = 1024 Kbps
- 1 Gbps = 1024 Mbps
- 1 Tbps = 1024 Gbps

Numerical: A user wants to upload a photo of the size 100 KB (kilobytes) within 10 seconds. What is the required data rate of the channel?

Solution:

Required data rate = $\frac{100 \times 1024 \times 8}{10}$ (each byte has 8 bits)

= $81,920$ bps = 80 Kbps

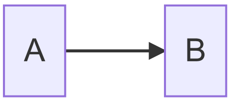
> **Important points:** These are simple numericals. Make sure to convert bytes to bits.

Types of Data Communication

1. Simplex Communication

- One way or **unidirectional** communication from the sender to the receiver.
- The devices use the maximum capacity of the transmission link.
- **Example:** Data from our keyboard is sent through simplex communication to our computer.

  Illustration:

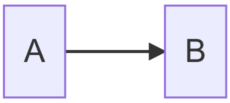


2. Half-Duplex Communication

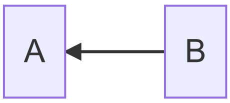
- It is a two way or **bidirectional** communication where both devices can act as the sender and receiver, but cannot be the same simultaneously. In simpler words, if there is data communication with A and B, when A is the sender, B can only act as the receiver and vice-versa.
- It can also be thought of as a simplex channel, but the direction can be switched.
- **Example:** Walkie-talkies. They are devices with a push-to-talk feature and only one person can speak at a time.

  Illustration:

  Instance 1:



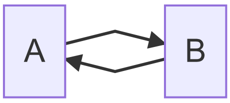
Instance 2:



3. Full-Duplex Communication

- Two way or **bidirectional** communication where both devices can act as the sender and receiver simultaneously.
- The capacity of the transmission link is shared between the signals going in both directions. Either two separate simplex lines are used, or the capacity of a single channel is shared.
- **Example:** during a telephone conversation, both sides can send and receive data.

  Illustration:



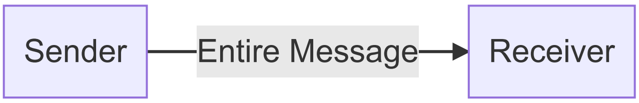
Comparison Table

Category	Simplex	Half-Duplex	Full-Duplex
Direction	Unidirectional	Bidirectional, but a device cannot simultaneously be a sender and receiver	Bidirectional, and a device can simultaneously be a sender and receiver
Capacity	Full capacity of transmission media used	Full capacity of transmission media used	Either two separate simplex links are used, or a single channel is shared
Example	Keyboard	Walkie-Talkie	Telephone

Switching Techniques

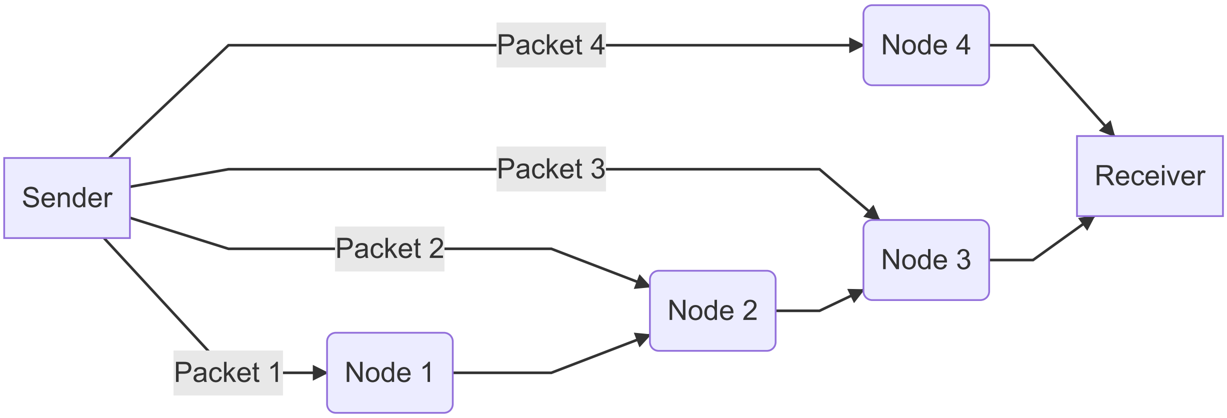
1. Circuit Switching

- A dedicated path is identified between the sender and receiver prior to the start of a communication.
- The path is a connected sequence of links between network nodes.
- All packets follow the same path.
- **Example:** Analog telephones would create a continuous wire circuit to transmit the information.



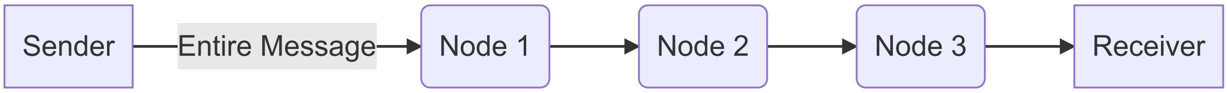
2. Packet Switching

- Information is broken down into smaller pieces called packets, which are independently transmitted and reassembled when received.
- Different packets of the same message may take different routes depending on availability.
- A channel is occupied only during packet transmission and is available for transmission to other senders afterwards.
- Packet components:
  1. **Header:** This contains the address of the destination and other metadata.
  2. **Message:** The actual message to be sent.



3. Message Switching

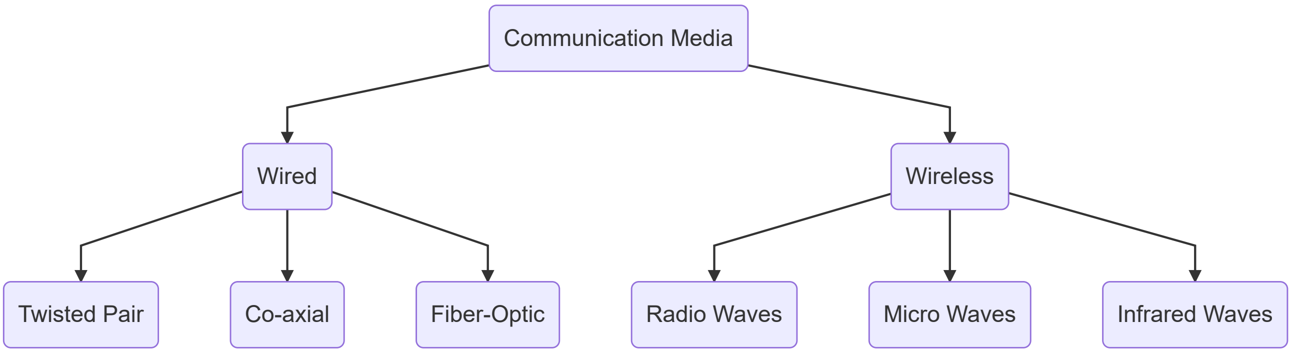
- Store and forward mechanism
- No physical connection
- The message is sent to intermediate nodes which check for errors and forward it to the next node until the destination is reached.



Comparison Table

Category	Circuit	Packet	Message
Data Type	Entire message	Small packets	Entire message
Path	Dedicated path	No dedicated path, packets follow various paths	No physical connection, store and forward mechanism
Example	Analog telephone	Internet Protocol, Ethernet	

Transmission Media



1. Wired (Guided)

NCERT Definition: Any physical link that can carry data in the form of signals belongs to the category of wired transmission media.

Twisted Pair Cable

- Two insulated copper wires twisted around each other to form a helical structure.
- Each wire acts as a single communication link.
- Twisting is done to remove electromagnetic interference.
- Types: unshielded and shielded (contain metal shield).

Coaxial Cable

- Copper wire surrounded by insulating material further surrounded by an outer conductor (copper mesh).
- Shielded design allows fast data transmission without interference.

Optical Fibre

- Data carried as light through a thin fibre of glass.
- The glass fibre is covered by the cladding made of less dense glass.
- Outer Jacket of PVC or Teflon.
- Unidirectional

Comparison Table

Category	Twisted Pair	Coaxial	Optical Fibre
Description	Two insulated copper wires twisted around each other to form a helical structure.	Copper wire surrounded by insulating material further surrounded by an outer conductor (copper mesh).	Data carried as light through a thin fibreof glass.
Speed/Bandwidth		More than twisted	High
Distance		Long distance	Long distance
Cost	Less expensive	More expensive than twisted pair	Most expensive
Use	Telephone lines, LAN	Ethernet	Wi-Fi

2. Wireless

NCERT Definition: In wireless communication technology, information travels in the form of electromagnetic signals through air.

- Uses frequencies of 3KHz to 900 THz

Property	Radio Waves	Microwaves	Infrared
Range of frequencies	3 KHz - 1 GHz	1 Gz - 300 GHz	300 GHz - 400 THz
Direction	Omni-directional	Unidirectional	
Distance	Long distance	Point to point communication or unicast	
Penentration	Susceptible to 3-300KHz can penentrate walls	Cannot penentrate solid objects	Cannot penentrate solid objects
Setup		Communication antenna for line of sight propagation	
Capacity		High	
Use	AM/FM Radio, television, cordless phones	Radar and Satellite	Bluetooth, Remotes

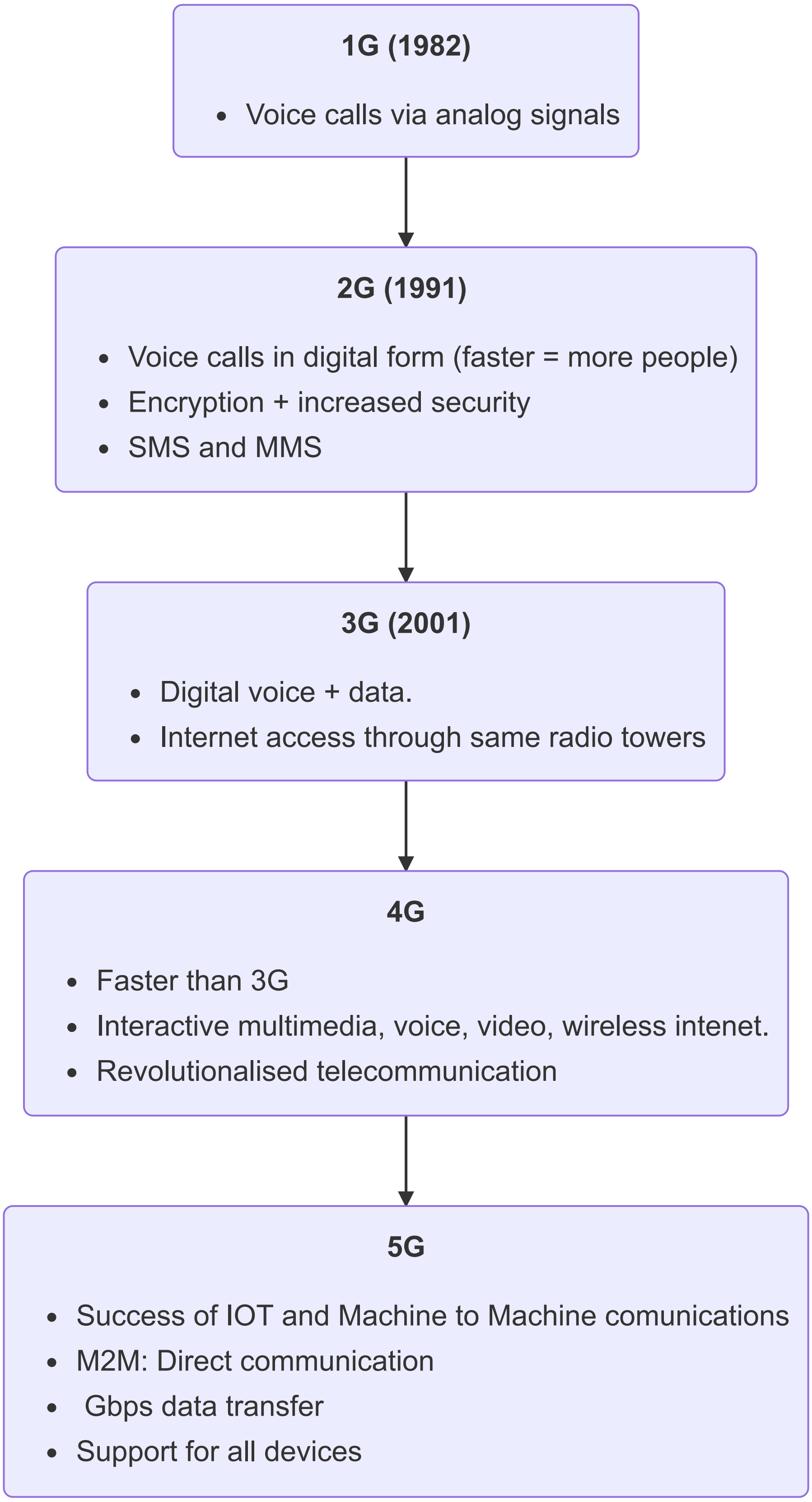
Bluetooth

- Short range
- Connects mobile phones, mouse, headphone, printer
- Low cost transciever chip (2.4 GHz)
- Speed 1-2 Mbps in 10 meter range.
- **Piconet:** Network of devices in 10m range
- 255 devices can form a network, 8 can communicate at the same time

Wireless LAN

- Wireless LAN is 802.11 (Wi-Fi)
- Devices + Access Points on buildings/floors
- **Access Point:** Creates a wireless LAN by connecting wired router, switch or hub
- APs connected through a wired network
- Benefits:
  - Can be used to extend or replace wired connection
  - More access for mobile devices
  - Easy internet access in public places

Mobile Telecommunication Technologies



1G (1982)

- Voice calls via analog signals

2G (1991)

- Voice calls in digital form (faster = more people)
- Encryption + increased security
- SMS and MMS

3G (2001)

- Digital voice + data.
- Internet access through same radio towers

4G

- Faster than 3G
- Interactive multimedia, voice, video, wireless intenet.
- Revolutionalised telecommunication

5G

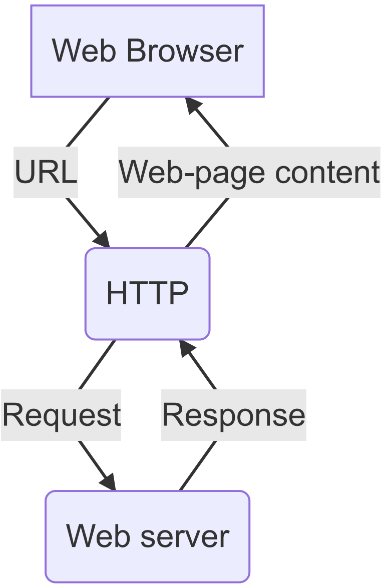
- Success of IOT and Machine to Machine comunications
- M2M: Direct communication
- Gbps data transfer
- Support for all devices

Network Protocols

Need for Protocols

- **Flow control:** Regulating the speed of sending and receiving data between two devices with different data transfer rates.
- **Access Control:** Identifying which nodes share a particular link to prevent packet collision which in turn causes data loss and corruption.
- **Addressing:**
  - Identification of nodes on a network
  - Form of data to be transmitted
  - Decide the final destination of the received data
  - Ensure complete transfer of data
  - Packet compilation at the receiver node

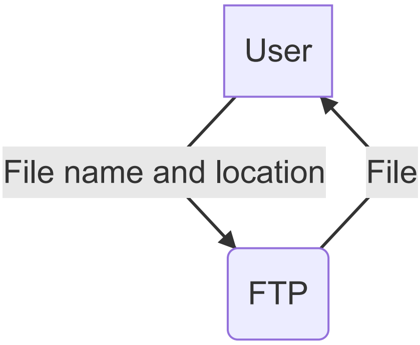
HyperText Transfer Protocol (HTTP)

- Primary protocol to access the World Wide Web
- Developed by Tim Berners-Lee in 1989
- Request response protocol that runs over TCP
- Used by web browser (client) and web server to access hypertext on the WWW
- Defines how information is to be transmitted and how servers and browsers should respond to commands.
- 

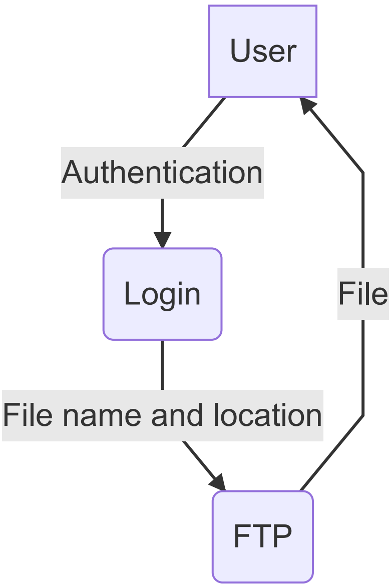
File Transfer Protocol (FTP)

- Used to transfer files between machines
- Client-server model like HTTP

  Case 1: No Authentication Required



Case 2: Authentication Required



Point to Point Protocol (PPP)

- Dedicated and direct connection between two devices
- Defines authentication of two devices and how to create a direct link
- Internet users connect computers to ISP servers with PPP
- Duplex mode required
- Data integrity and sequential order off packet transfer
- Sender is notified about damaged/lost packets and asked to resend

Simple Mail Transfer Protocol (SMTP)

- Used by email services
- Uses information on the message header only
- Email header has email address and the emails are entered into a queue
- Takes mail from outgoing queue and transmits to destination
- Removes mail from queue after sending

Transmission Control Protocol (TCP) / Internet Protocol (IP)

- Client-server model of communication
- IP ensures that all nodes have an IP address
- TCP facilitates IP packet formation
- The IP packets are transmitted through the internet from one router to another
- TCP ensures smooth delivery of IP packets to a particular IP address
- TCP is also responsible for ordering packets
- There can be multiple packet paths since redundant connections exist.

POP3

- Used to retrieve emails from the server
- SMTP sends email, POP3 receives

TELNET

- Used to connect to a local device
- Command line interface

Voice over Internet Protocol (VoIP)

- Used to transmit audio data over the internet
- Converts voice to a digital signal and transmitted to other devices

NCERT Solutions

Q1. What is data communication? What are the main components of data communication?
Data communication is the exchange of data between two or more networked or connected devices
The main components of data communication are:

1. **Sender:** The sender is a node in the network which transmits a message to another node in the network.
2. **Receiver:** The receiver is a node which can receive data from other nodes in the network.
3. **Message:** It is the data that is being transmitted through the network.
4. **Communication Media:** It is the path through which a message is transmitted. For example, an ethernet cable, telephone wire or satellite link are communication media used to transmit various kinds of signals.
5. **Protocols:** They are the rules that must be followed by communication devices for reliable data communication. Example: HTTP, Ethernet.

Q2. Which communication mode allows communication in both directions simultaneously?
Full Duplex Communication allows communication in both directions simultaneously

Q3. Among LAN, MAN and WAN, which has the highest speed and which one can cover the largest area?
Highest speed: LAN
Largest area: WAN

Q4. What are three categories of wired media? Explain them.
The three categories of wired media are:

1. **Twisted Pair Cable:** These are copper wires insulated and twisted together. They have a helical structure.
2. **Coaxial Cable:** These are solid copper wires surrounded by plastic cladding shielded with a wire mesh.
3. **Optical Fibres:** These consist of thin glass fibres and carry data as visible light. Sources converts data using LED or LASER

Q5. Compare wired and wireless media

Wired	Wireless
These use physical path(wires) to transmit data	They use a wireless medium to transmit data
This provides direction to travelling signal	The signals travel in a directionless manner
Eg: Twisted pair cable, coaxial cable, optical fibre	Eg: Radio Waves, Microwaves, Infrared

Q6. Which transmission medium carries signals as forms of light?
Optical fibres

Q7. List out the advantages and disadvantages of optical fibres
Advantages:

- High bandwidth(ie. higher data transfer)
- Faster data transmission
- Negligible loss of data
- Secure

Disadvantages:

- Very costly
- Vulnerable to damage
- Difficult installation

Q8. What is the range of frequencies of radio waves?
3KHz to 1GHz

Q9. 18 Gbps is equal to how many bits per second?
1 Gbps = 2^{10} Mbps = 2^{20} Kbps = 2^{30} bps
therefore 18Gbps = 18×2^{30} = 19327352832 bps

Q10. HTTP stands for?
Hypter Text Transfer Protocol

Q11. Write a short note on the following:
a) HTTP
HTTP is the HyperText Transfer Protocol which is the primary protocol for the World Wide Web(WWW). It is a request-response protocol which runs over TCP. It facilitates how information should be formatted and transmitted, and how web servers/browsers respond to different commands

b) Bandwidth
Bandwidth is the maximum amount of data transmitted by a connection in time. It is calculated as the difference of the maximum and minimum frequency contains in composite signals

c) Bluetooth
Bluetooth is a short-ranged wireless technology to connect devices(like phones, mouse, headphones, keyboards etc) wirelessly over short distance. Bluetooth devices have a low cost transceiver chip

d) DNS
DNS is the Domain Name System, by which web resources are given a name against their IP Address.

e) Data transfer rate
Data transfer rate is the number of bits transferred between source and destination in each second. It is usually measured in bits per second(bps). Other units are:

1Kbps = 2^{10} bps
1Mbps = 2^{10} Kbps = 2^{20} bps
1Gbps = 2^{10} Mbps = 2^{20} Kbps = 2^{30} bps
and so on

Q12. What is protocol in data communication? Explain with an example
Protocol is a set of rules that need to be followed by communicating parties to have successful and reliable communication.
An example of protocol is HTTP, which is the HyperText Transfer Protocol.

Q13. A composite signal contains frequencies between 500MHz and 1GHz. What is the bandwidth of the signal?
To find the bandwidth, we must first convert both the frequencies to the same unit.
Let's convert both to MHz
$f_1 =$500MHz (min frequency)
f_2 = 1GHz = 10^3 MHz = 1000 MHz (max frequency)
Therefore bandwidth = $f_{\text{max}} - f_{\text{min}} = (1000 - 500)$ Mhz = **500 MHz**

Sample Paper Questions

2.
(i) Expand the following: SMTP , XML (1)
SMTP = Simple Mail Transfer Protocol
XML = Extensible Markup Language

You get 0.5 marks for each correct expansion

ii) Out of the following, which is the fastest wired and wireless medium of transmission?
Infrared, coaxial cable, optical fibre, microwave, Ethernet cable
Fastest wired = Optical Fibre
Fastest wireless = Microwave

You get 0.5 marks for identifying each correctly

12. ii) How is packet switching different from circuit switching? (2)

Packet switching	Circuit switching
No physical path is established before data transmission	First a complete path is established between source and destination, then data is transmitted
Message is divided into smaller parts called packets which are then sent forward	Each data unit follows the same established path between sender and receiver
Different packets may take different routes depending on availability	There is a guarantee that the message is delivered

You get 1 mark for each correct difference
Minimum 2 points must be given

PYQs

1. How many pair of wires are there in twisted pair cable (Ethernet)? What is the name of port ,which is used to connect Ethernet cable to a computer or a laptop?

The twisted pair cable consists of two insulated copper wires wound around each other. The ethernet cable is connected to the device through the ethernet port.

2. Tarini Wadhawa is in India and she is interested in communicating with her uncle in Australia. She wants to show one of her own designed gadgets to him and also wants to demonstrate its working without physically going to Australia. Which protocol out of the following will be ideal for the same ?
(1) [CBSE Computer Science Board 2019]
(i) POP3
(ii) SMTP
(iii) VoIP
(iv) HTTP

iii) VoIP (Voice-over Internet Protocol) would be ideal for the same, since it allows delivering of both voice and multimedia communications over Internet Protocol

3. Give two differences between 3G and 4G telecommunication technologies. [CBSE Computer Science Board 2019]

3G	4G
Provides internet access via same radio towers that provide voice services	Supports interactive multimedia voice, video, wireless internet and other broadband services
Relatively slow data transfer speed	Relatively fast data transfer speed

Chapter 4: Database concepts

We have learnt about file handling and we know how it is useful for storing data.

Now we will learn about **Databases**

Databases is a collection of information that is organized, which can be easily managed, accessed and updated

Limitations of Files

1. **Difficulty in access:** Certain file types have specific applications to use them. Since app developers may not foresee all possible ways to use the output files of their programs, not all files are accessible without the required application.
2. **Data redundancy:** When data is repeated unnecessarily in files it is pointless and wastes memory. It may also result in data inconsistency.
3. **Data Inconsistency:** Suppose you have two files containing the same data. If you update data in one of them, the changes will not be reflected in the other, which may cause problems if someone does not have the updated copy.
4. **Data isolation:** We cannot link files containing similar data without creating separate programs. This becomes complex as the files are in different locations and may have different structures.
5. **Data dependence:** Programs are configured to access file contents in a particular format. If the format of the files is changed, the programs must be changed accordingly.
6. **Controlled Data Sharing:** In some cases, organizations and groups may want to limit the access of data to certain members only. Controlling data sharing in files is quite difficult.

Database Management System(DBMS)

DBMS resolves the problems created by file systems.

DBMS is a software that is used to create and manage databases. It allows users to create, store, manage, update and retrieve data from the database.

Examples: MySQL, Oracle, MongoDB

Advantages of DBMS

1. **Easy access:** Data in databases maintain data in one location. A database contains many components of information known as tables.
2. **No inconsistency:** If content in a database is updated, it is updated for all users.
3. **Data security and access control:** DBMS ensures security and only authorized personnel can access the data.
4. **Data Mapping:** Data between tables in a DBMS can be linked using common columns.
5. **No Data Dependence:** If the structure of a table in a database is changed, programs can still access the contents without modification.

DBMS Terminology

Understand a few key terms to understand and use DBMS efficiently.

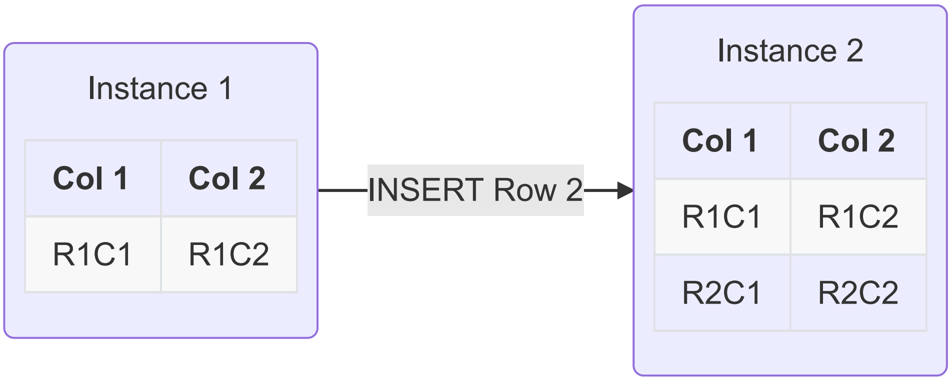
- **Database schema:** It is the outline of the database and the structure of the tables.

Information in schema:

1. Table names
2. Table fields: The columns for individual tables.
3. Data type of columns (fields): Whether the column contains integers, floats, etc.
4. Data constraints: (see next point).
5. Mappings: tables which are linked by a common field

> **Q:** Why is database schema also called the visual or logical architecture?
>
> Ans: Database schema gives us an idea of how a database is structured and the tables included in it.

- **Data constraints:** Constraints are used to ensure that only certain kinds of data can be entered. For example, if we are creating a database of adults with driving licenses, there will be an age constraint where age is greater than or equal to 18.
- **Meta-data/Data dictionary:** Metadata means information that describes data. The metadata of a DBMS comprises of the databases and the constraints on the data.
- **Database instance:** A database instance refers to what the database would look at a particular instance of time. For example in a database with one table:



Instance 1

Col 1	Col 2
R1C1	R1C2

Instance 2

Col 1	Col 2
R1C1	R1C2
R2C1	R2C2

The database instance is also known as a snapshot or state.

- **Query:** A special command used to retrieve specific data from a database. For example we can find the names of citizens who got their driving license in 2022 by querying the database and ensuring that only the required names are selected.
- **Data manipulation:** A Data Manipulation Language (DML) primarily consists of 3 commands: insert, update and delete. These commands are used to *manipulate* the data in a database.

  The commands names are very self explanatory:

  - **Insert:** Adds data at the end of table
  - **Update:** Updates previously added data
  - **Delete:** Deletes data from table
- **Database engine:** It is the component/set of programs used by DBMS to create databases and handle queries for data retrieval/manipulation

Data models

A data model of a database defines the **logical structure** of the database. It describes the way in which data can be stored, organized, represented and manipulated.

In our course, we will be studying the **Relational Data Model** (next section)

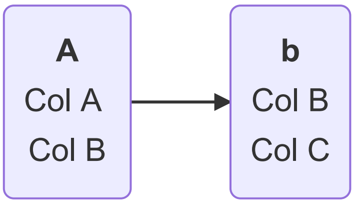
Other examples of data models include object-oriented, entity-relationship, document and hierarchical data models

Relational Data Model

Relational data model is the kind of data model where the database is organized into tables (called relations). Each relation stores data in unique columns.

The advantage of using relational model is that all the relations of a given database are connected to one another

For example:



This database has two relations (A and B) which are linked by Col B.

Terminology

1. **Attribute:** An attribute is the same thing as a column in a table, and is also known as a field.
2. **Tuple:** A tuple is the same thing as a row in a table.
3. **Domain:** The set of possible values in an attribute.
4. **Degree:** The number of attributes.
5. **Cardinality:** The number of tuples.

Relation Properties

We have summarised these points from the NCERT textbook as they are very important:

1. **Attribute Rules**
    - Each attribute has a unique name.
    - The sequence of attributes in a relation is not important.
2. **Tuple Rules**
    - Each tuple is distinct.
    - Sequence of tuples is not important.
3. **State Rules**
    - Attribute values must be in the same domain.
    - Data must be atomic: cannot be divisible into meaningful subparts.
    - Each attribute has only one value in a tuple.
    - We use NULL to represent unknown values.

Keys

We will use the following table to demonstrate the upcoming concepts:

Name	Age	Email	Phone	Grade
A	18	A@qprogramming.net	1234567890	A
B	17	B@qprogramming.net	1234567891	B
C	18	C@qprogramming.net	1234567892	A

Candidate Key

A candidate key is an attribute which has unique values in each tuple.

In the table, we can see that the Email and Phone can both be used to uniquely identify a student.

Hence, "Email" and "Phone" are considered as candidate keys.

Primary Key

A primary is the candidate key chosen for the purpose of uniquely identifying a tuple in the relation.

In this example, if we choose "Email" as the primary key, "Phone" automatically becomes an alternate key.

If there are more than two candidate keys, one key is the primary and the rest are alternate keys.

Note that the primary key **cannot be NULL**.

Composite Primary Key

In the event that a relation has no candidate keys, we group different attributes as a composite primary key.

We choose a set of attributes which when used together, are unique to each tuple.

Foreign Key

We use a foreign key to connect one relation to another.

Let's consider a second relation in our database:

Phone	Address
1234567890	123 ABC road
1234567891	124 PQR road
1234567892	125 CDE road

You can see that the "Phone" column is common to both relations and can be used to link them. Therefore, we can access the address of a student by linking the two relations.

Here, the foreign key in the first table references a common attribute in the second relation.

The referenced attribute in the second relation is also the primary key of the second relation.

NCERT Solutions

1. **Give the terms for each of the following:**

   a) Collection of logically related records.

   b) DBMS creates a file that contains description about the data stored in the database.

   c) Attribute that can uniquely identify the tuples in a relation.

   d) Special value that is stored when actual data value is unknown for an attribute.

   e) An attribute which can uniquely identify tuples of the table but is not defined as primary key of the table.

   f) Software that is used to create, manipulate and maintain a relational database.

   a) Relation

   b) Schema

   c) Primary Key

   d) NULL

   e) Alternate Key

   f) Database Management System (DBMS)

2. **Why foreign keys are allowed to have NULL values? Explain with an example.**

3. **Differentiate between:**

   a) Database state and database schema

   b) Primary key and foreign key

   c) Degree and cardinality of a relation

   a)

Database State	**Database Schema**
The composition of the database at a particular instance in time.	The structure of how data is organized in a database.
Includes actual data in the database.	Does not include actual data.

   b)

Primary Key	**Foreign Key**
Uniqiuely identifies a tuple in a particular relation.	Uniquely identifies a tuple in a foreign relation (a separate relation).
Cannot be NULL.	May or may not be NULL.

   C)

Degree	Cardinality
A measure of the number of tuples in a relation.	A measure of the number of attributes in a relation.

4. **Compared to a file system, how does a database management system avoid redundancy in data through a database?**

   File systems may contain different versions of the file and different computers may store different versions as well. This creates a redundancy as the data is stored in multiple locations whereas in a database, the data is stored in a central location.

5. **What are the limitations of file system that can be overcome by a relational DBMS?**

   1. **Easy access:** Data in databases maintain data in one location. A database contains many components of information known as tables.
   2. **No inconsistency:** If content in a database is updated, it is updated for all users.
   3. **Data security and access control:** DBMS ensures security and only authorized personnel can access the data.
   4. **Data Mapping:** Data between tables in a DBMS can be linked using common columns.
   5. **No Data Dependence:** If the structure of a table in a database is changed, programs can still access the contents without modification.

6. **A school has a rule that each student must participate in a sports activity. So each one should give only one preference for sports activity. Suppose there are five students in a class, each having a unique roll number. The class representative has prepared a list of sports preferences as shown below. Answer the following:**

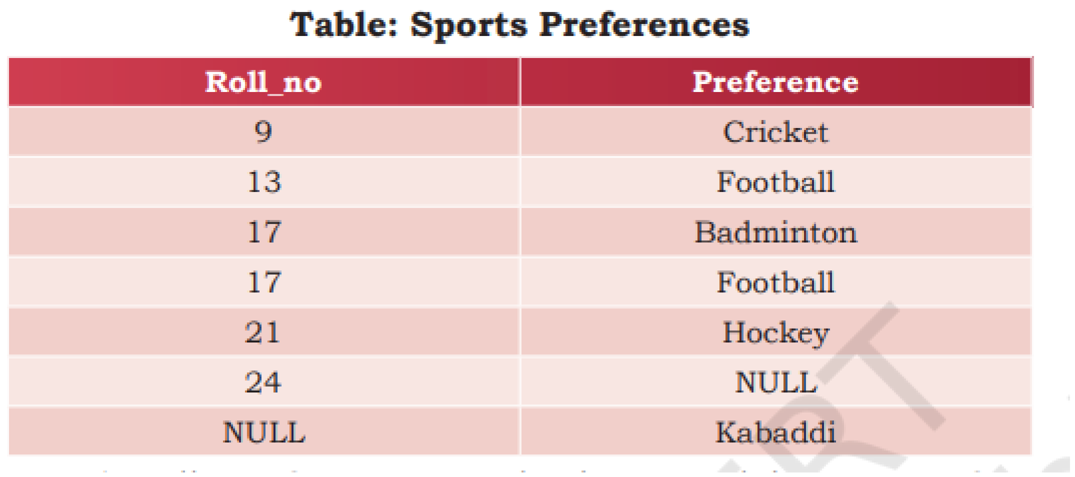


Table: Sports Preferences

Roll_no	Preference
9	Cricket
13	Football
17	Badminton
17	Football
21	Hockey
24	NULL
NULL	Kabaddi

a) Roll no 24 may not be interested in sports. Can a NULL value be assigned to that student's preference field?

No, as the school mandates participation from all students

b) Roll no 17 has given two preferences in sports. Which property of relational DBMC is violated here? Can we use any constraint or key in the relational DBMS to check against such violation, if any?

The tuples are not unique. We can assign the Roll_no as the primary key to avoid the violation

c) Kabaddi was not chosen by any student. Is it possible to have this tuple in the Sports Preferences relation?

No, as there is no primary key value associated with it and the primary key cannot be NULL

7. **In another class having 2 sections, the two respective class representatives have prepared 2 separate Sports Preferences tables, as shown below: Sports preference of section 1 (arranged on roll number column)**

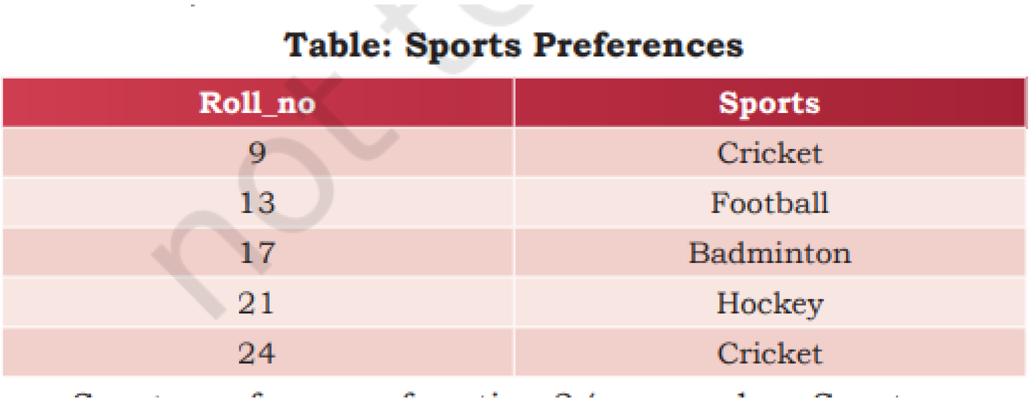


Table: Sports Preferences

Roll_no	Sports
9	Cricket
13	Football
17	Badminton
21	Hockey
24	Cricket

Sports preference of section 2 (arranged on Sports name column, and column order is also different)

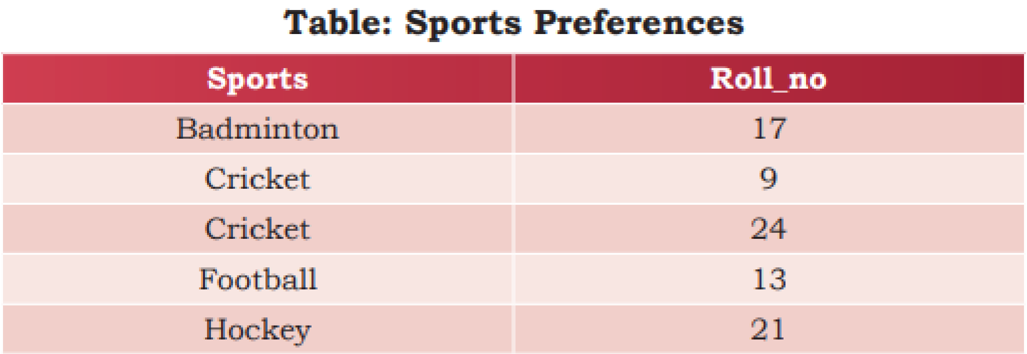


Table: Sports Preferences

Sports	Roll_no
Badminton	17
Cricket	9
Cricket	24
Football	13
Hockey	21

Are the states of both the relations equivalent? Justify.

The states are equivalent as the order of the rows and columns does not matter and the content is same.

8. **The school canteen wants to maintain records of items available in the school canteen and generate bills when students purchase any item from the canteen. The school wants to create a canteen database to keep track of items in the canteen and the items purchased by students. Design a database by answering the following questions:**

   a) To store each item name along with its price, what relation should be used? Decide appropriate attribute names along with their data type. Each item and its price should be stored only once. What restriction should be used while defining the relation?

   A relation "Items" can be used where the attributes are "itemname", a string and "price", an integer. The itemname attribute can be set as the primary key to ensure unqiueness.

   b) In order to generate bill, we should know the quantity of an item purchased. Should this information be in a new relation or a part of the previous relation? If a new relation is required, decide appropriate name and data type for attributes. Also, identify appropriate primary key and foreign key so that the following two restrictions are satisfied:

   i) The same bill cannot be generated for different orders.

   ii) Bill can be generated only for available items in the canteen.

We should use a new relation as the previous one is based on the already known prices and the current relation is for managing purchases which will happen continuously.

The new relation should be called "purchases" and the attributes should be "orderid" a string, "itemname" a string and "quantity", an integer. The "orderid" attribute must act as the primary key in the "purchases" relation and "itemname" must act as the foreign key to connect "Items" and "purchases".

c) The school wants to find out how many calories students intake when they order an item. In which relation should the attribute ‘calories’ be stored?

Calories should be stored in the "Items" relation as it is a known quantity associated with each item.

9. **An organisation wants to create a database EMPDEPENDENT to maintain following details about its employees and their dependent.**

   EMPLOYEE(AadharNumber, Name, Address, Department,EmployeeID)
   DEPENDENT(EmployeeID, DependentName, Relationship)

   a) Name the attributes of EMPLOYEE, which can be used as candidate keys.

   AadharNumber, EmployeeID

   b) The company wants to retrieve details of dependent of a particular employee. Name the tables and the key which are required to retrieve this detail.

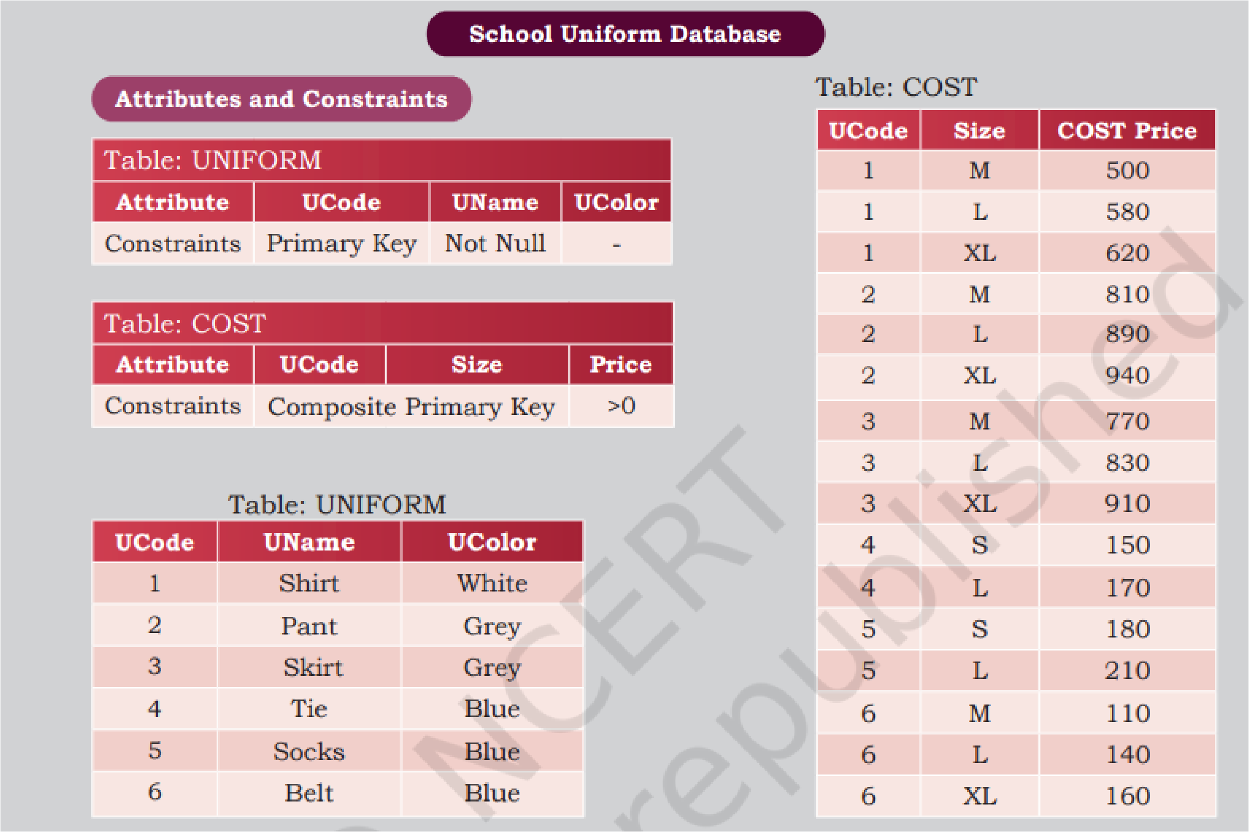
   We require the employee and dependent table as well as the employeeID key to uniquely identify the employee

   c) What is the degree of EMPLOYEE and DEPENDENT relation?

   Employee: 5

   Dependent: 3

10. **School uniform is available at M/s Sheetal Private Limited. They have maintained SCHOOL_UNIFORM Database with two relations viz. UNIFORM and COST. The following figure shows database schema and its state.**



School Uniform Database

Attributes and Constraints

Table: UNIFORM

Attribute	UCode	UName	UColor
Constraints	Primary Key	Not Null	-

Table: COST

Attribute	UCode	Size	Price
Constraints	Composite Primary Key		>0

Table: UNIFORM

UCode	UName	UColor
1	Shirt	White
2	Pant	Grey
3	Skirt	Grey
4	Tie	Blue
5	Socks	Blue
6	Belt	Blue

Table: COST

UCode	Size	COST Price
1	M	500
1	L	580
1	XL	620
2	M	810
2	L	890
2	XL	940
3	M	770
3	L	830
3	XL	910
4	S	150
4	L	170
5	S	180
5	L	210
6	M	110
6	L	140
6	XL	160

a) Can they insert the following tuples to the UNIFORM Relation? Give reasons in support of your answer.

i) 7, Handkerchief, NULL

Yes, as there are no constraints on UColor

ii) 4, Ribbon, Red

No, as a tuple having UCode as 4 already exists and it will violate the primary key constraint

iii) 8, NULL, White

No, as the UName is bound by the NOT NULL constraint

b) Can they insert the following tuples to the COST Relation? Give reasons in support of your answer.

i) 7, S, 0

No, as the price must be greater than zero

ii) 9, XL, 100

No, as UCode is the foreign key and as it is also part of the primary key, it cannot be NULL.

11. **In a multiplex, movies are screened in different auditoriums. One movie can be shown in more than one auditorium. In order to maintain the record of movies, the multiplex maintains a relational database consisting of two relations viz. MOVIE and AUDI respectively as shown below:**

Movie(Movie_ID, MovieName, ReleaseDate)

Audi(AudiNo, Movie_ID, Seats, ScreenType, TicketPrice)

a) Is it correct to assign Movie_ID as the primary key in the MOVIE relation? If no, then suggest an appropriate primary key.

Yes, as the MOVIE_ID will be unique for each movie and can be used to uniquely identify a tuple in the relation.

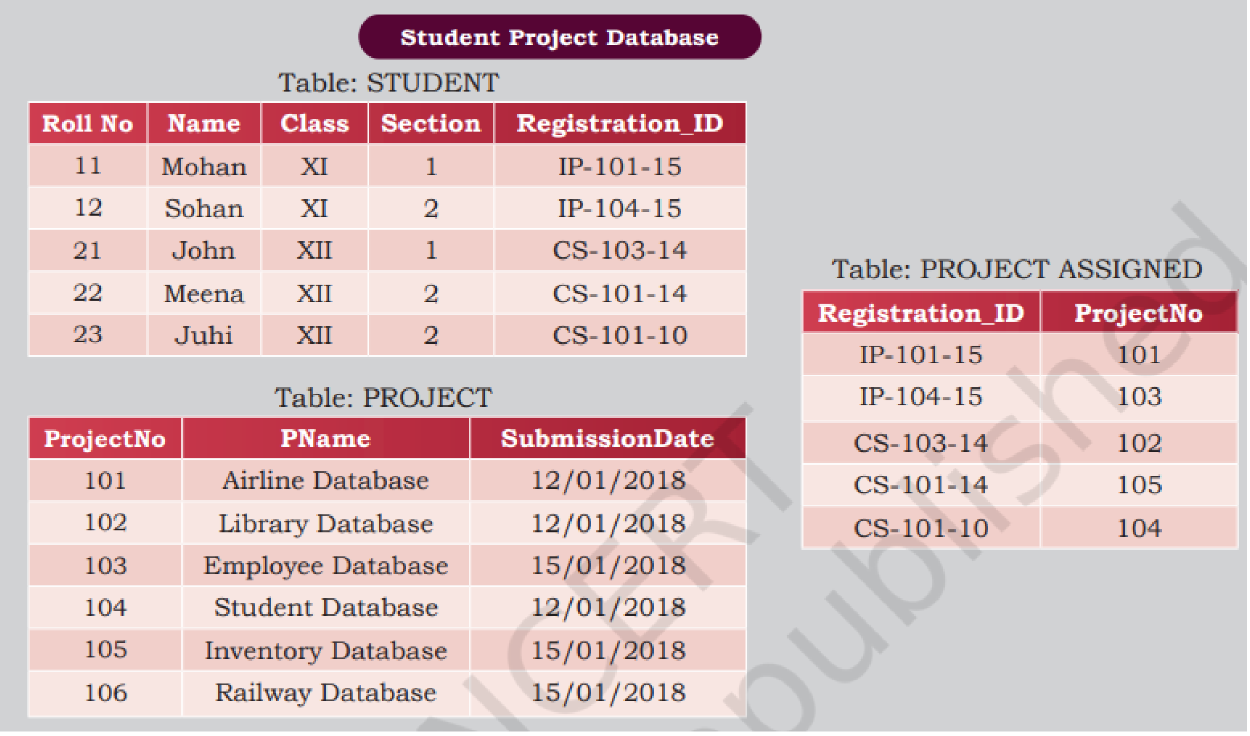
b) Is it correct to assign AudiNo as the primary key in the AUDI relation? If no, then suggest appropriate primary key.

A particular Audi may host different movies so we cannot use it as the primary key. We can use a composite primary key instead consisting of AudiNo and Movie_ID.

c) Is there any foreign key in any of these relations?

Movie_ID acts as the foreign key.

12.



Student Project Database

Table: STUDENT

Roll No	Name	Class	Section	Registration_ID
11	Mohan	XI	1	IP-101-15
12	Sohan	XI	2	IP-104-15
21	John	XII	1	CS-103-14
22	Meena	XII	2	CS-101-14
23	Juhi	XII	2	CS-101-10

Table: PROJECT

ProjectNo	PName	SubmissionDate
101	Airline Database	12/01/2018
102	Library Database	12/01/2018
103	Employee Database	15/01/2018
104	Student Database	12/01/2018
105	Inventory Database	15/01/2018
106	Railway Database	15/01/2018

Table: PROJECT ASSIGNED

Registration_ID	ProjectNo
IP-101-15	101
IP-104-15	103
CS-103-14	102
CS-101-14	105
CS-101-10	104

For the above given database STUDENT-PROJECT, answer the following:

a) Name primary key of each table.

STUDENT: Roll No

PROJECT: ProjectNo

PROJECT ASSIGNED: Registration_ID

b) Find foreign key(s) in table PROJECT-ASSIGNED.

Registration_ID is a foreign key from STUDENT referencing PROJECT ASSIGNED

ProjectNo is a foreign key from PROJECT ASSIGNED referencing PROJECT

c) Is there any alternate key in table STUDENT? Give justification for your answer.

Registration_ID may act as the alternate key since it has a unique value for each tuple.

d) Can a user assign duplicate value to the field RollNo of STUDENT table? Jusify.

RollNo will act as the primary key as each student has a unqiue Roll Number and no duplicated values will be allowed.

13. **For the above given database STUDENT-PROJECT, can we perform the following operations?**

a) Insert a student record with missing roll number value.

No, as RollNo is the primary key and cannot have a NULL value.

b) Insert a student record with missing registration number value.

Yes, as the Registration_ID attribute in the STUDENT relation may be NULL if the student is not registered already.

However it may not be missing in PROJECT ASSIGNED as it acts as the primary key.

c) Insert a project detail without submission-date.

Yes, as the SubmissionDate attribute does not required a value and the student may not have submitted the project.

d) Insert a record with registration ID IP-101-19 and ProjectNo 206 in table PROJECT-ASSIGNED.

No, as referential integrity is violated given that no project in PROJECT has this project number.

Sample Paper Questions

1. **Consider the table, MOVIEDETAILS given below:**

   Table: MOVIEDETAILS
   MOVIEID TITLE LANGUAGE RATING PLATFORM
   M001 Minari Korean 5 Netflix
   M004 MGR Magan Tamil 4 Hotstar
   M010 Kaagaz Hindi 3 Zee5
   M011 Harry Potter and the Chamber of Secrets English 4 Prime Video
   M015 Uri Hindi 5 Zee5
   M020 Avengers: Endgame English 4 Hotstar

   (a) Identify the degree and cardinality of the table.

   Degree (Number of attributes): 5

   Cardinality (Number of tuples): 6

   (b) Which field should be made the primary key? Justify your answer

   MOVIEID should be the primary key as it can be used to uniquely identify tuples in the relation

2. **(a) Identify the candidate key(s) from the table MOVIEDETAILS.**

   MOVIEID and TITLE are candidate keys.

   (b) Consider the table SCHEDULE given below:

   Table: SCHEDULE
   SLOTID MOVIEID TIMESLOT
   S001 M010 10 AM to 12 PM
   S002 M020 2 PM to 5 PM
   S003 M010 6 PM to 8 PM
   S004 M011 9 PM to 11 PM

   Which field will be considered as the foreign key if the tables MOVIEDETAILS and SCHEDULE are related in a database?

   MOVIEID will be considered as the foreign key

Analysis

The sample paper contains only 2 marks worth of questions indicating that this chapter is not very important. However, it is important to know all the material as the concepts are connected to SQL, the largest chapter in the syllabus.

From sample paper questions and PYQs, the important questions tend to be about the degree and cardinality as well as primary and candidate key identification. One must revise these topics thoroughly to secure 2 marks. In addition, one must be familiar with the terminology, advantages of DBMS and properties of RDBMS.

Questions may also show up as subparts of SQL based questions such as identification of the primary key in the given table.

PYQs

1. **Observe the following table and answer the parts (i) and (ii) accordingly**

   Table:Product

   Pno Name Qty PurchaseDate
   101 Pen 102 12-12-2011
   102 Pencil 201 21-02-2013
   103 Eraser 90 09-08-2010
   109 Sharpener 90 31-08-2012
   113 Clips 900 12-12-2011

   (i) Write the names of most appropriate columns, which can be considered as candidate keys. (1)

   Pno and Name are candidate keys

   (ii) What is the degree and cardinality of the above table? (1)

   Degree: 4
   Cardinality: 5

2. **Differentiate between Primary key and Candidate key (2)**

Primary Key	Candidate Key
Each relation has only one primary key	Each relation may have one or more candidate keys
The primary key is always a candidate key	The candidate key may or may not be the primary key

3. **Differentiate between Degree and Cardinality (2)**

Degree	Cardinality
The number of attributes in a relation	The number of tuples in a relation
Cannot be 0	May be 0 (in empty relation)

Structured Query Language (SQL)

The Structured Query Language (SQL) is one of the most popular **query language** used for managing data in Relational Database Management Systems(RDBMS)

It can be used on a DBMS like MySQL, Oracle, etc.

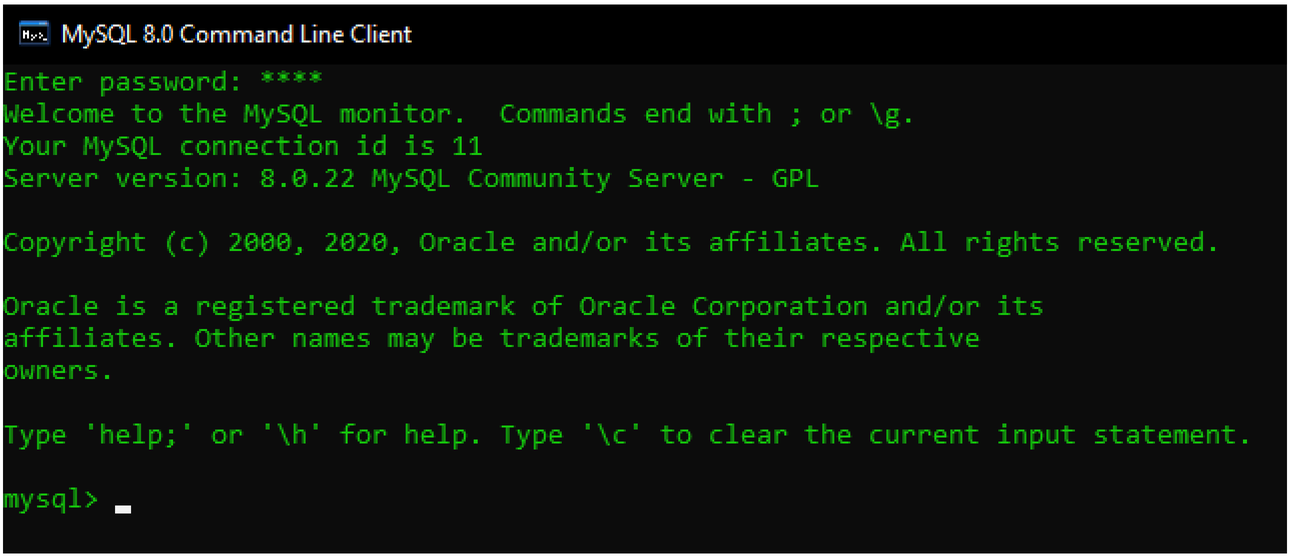
Remember that commands in SQL are **case-insensitive**, so whether you write everything in uppercase letters or lowercase letters, SQL will treat it the same way (similar to HTML).

Installation

In class 12 we will be using MySQL. To install MYSQL go to https://dev.mysql.com/downloads

Set up a password and it is ready to use! Make sure to save this password somewhere so you don't get locked out!

After installing, open the `MySQL Command Line Client` and a screen similar to this would appear:



```
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.22 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

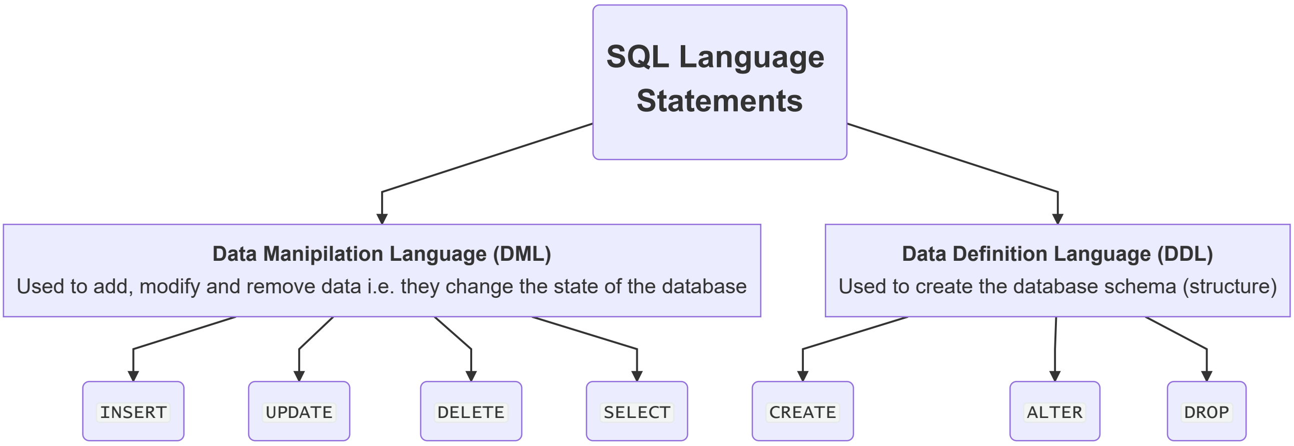
The line which says `mysql>` is where you will type your queries. That is where your cursor will be by default

You can run your first query by typing `show databases;` and pressing enter.

It will list all the databases in your system at the time!

SQL Language statements

SQL has the following categories of language statements


SQL Language Statements
Data Manipilation Language (DML)
Used to add, modify and remove data i.e. they change the state of the database
Data Definition Language (DDL)
Used to create the database schema (structure)
INSERT
UPDATE
DELETE
SELECT
CREATE
ALTER
DROP

> **Important Question:** Divide a set of statements into DML and DDL commands.

Data Types

There are different data types offered by SQL:

1. `CHAR(n)`: A sequence of n characters ($0 \leq n \leq 255$).

   If the value received has less characters than n, then the **remaining are filled with blank spaces on the right**.

   For example, if n=8, and the value received is "apple", then it will be stored as "apple   ", with 3 blank spaces to the right.

2. `VARCHAR(n)`: A sequence of up to n characters ($0 \leq n \leq 65535$).

   If the value received has less characters than n, then it will only occupy as many bytes as needed, without any excess blank spaces (unlike `CHAR`).

   For example, if n=8 and the value received is "apple", it will be stored as "apple".

3. `INT`: Integer values.

   Each value having `INT` occupies 4 bytes. If the data contains larger integers, we will use `BigINT` which can occupy 8 bytea.

4. `FLOAT`: The value must be a float (similar to Python floats). They occupy 4 bytes.

5. `DATE`: These are date timestamps.

   The default format for storing dates in SQL is `YYYY-MM-DD`.

   For example, `2022-02-19` is the 19th of February, 2022.

The names of the datatypes are intuitive. Keep in mind the difference between `CHAR(n)` and `VARCHAR(n)`, and all the formats.

> **Important Question:** Differentiate between char and varchar

> Key points:
>
> char(n): if received characters <n, remaining characters filled with blank spaces, wastage of space, n = 0-255
>
> varchar(n): no space padding, memory efficient, n = 0-65535

Constraints

Everywhere you go, there are a set of rules you must adhere to. Constraints act as rules on data

The following are the different types of constraints in SQL:

1. `NOT NULL`: Makes sure that the field can have no value which is `NULL` (unknown).
2. `UNIQUE`: Ensures that the values in the column are distinct.

   For example in the following table:

Col 1	Col 2
R1C1	R1C2
R2C1	R2C2

   If I try to add a row with values (R2C1,R3C2) but Col 1 has a `UNIQUE` constraint, it will give an error:

```
mysql> INSERT INTO table VALUES("R2C1","R3C2");
ERROR 1062 (23000): Duplicate entry 'R1C1' for key 'table.col1'
```

Don't worry about the command right now, it just tries to add a record to the table.

3. `DEFAULT`: Gives a default value to a record if no value is given, instead of `NULL`.
4. `CHECK(condition)`: Checks whether a condition is satisfied.
5. `PRIMARY KEY`: The primary key is the column in a table which has unique values to identify individual records.
6. `FOREIGN KEY`: It must be the primary key in another table, and serves as a link from one table to another.

> **Important Question:** Why can't the PRIMARY KEY be NULL?
>
> Key Points: The PRIMARY KEY is used to uniquely identify individual records and hence cannot be NULL
>
> **Important Question:** Why can the FOREIGN KEY be NULL in some cases?
>
> Key Points: The FOREIGN KEY can be NULL as the table it references may not contain a record for a particular value in the first table. For example, in a database for a competition, there are two tables. The first is `participants` with attributes StudentID, Name, SubmissionID and `submissions` with attributes SubmissionID and Submission date. In case a student has registered as a participant, but not submitted their project, the `submissions` table will not have a record of the submission and hence the FOREIGN KEY (SubmissionID) in `participants` will be NULL until the participant submits their project.

We will be using both data-types and constraints to create relations in databases.

Data Definition (DDL)

These are the commands by which you can create and modify the **schema** of databases and the relations within them. The schema refers to the structure of a database/relation.

Database Operations

`CREATE`

```
CREATE DATABASE database_name;
```

`database_name` is a placeholder and you can replace it with your desired name.

This is how an example would look:

```
mysql> CREATE DATABASE qprogramming;
Query OK, 1 row affected (0.06 sec)
```

`SHOW`

To see a list of all databases on your computer system, run the command `SHOW DATABASES;`.

It will show you a list of all the databases that exist on your computer, including the one you just created!

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| qprogramming       |
| sakila             |
| sys                |
| world              |
+--------------------+
7 rows in set (0.02 sec)
```

`USE`

To use the database you just created, you can use the command `USE database_name;`.

```
mysql> USE qprogramming;
Database changed
```

Congratulations! You are now using the database you just created.

> **Important:** Remember to add a semicolon at the end of each query!

`DROP`

To remove a database from your computer system, you can use the command `DROP DATABASE database_name;` and the database and everything within it will be deleted.

```
mysql> DROP DATABASE qprogramming;
Query OK, 0 rows affected (0.01 sec)
```

If you now run `SHOW databases;`, you will *not* see the database in the list anymore.

Table Operations

`CREATE`

Once you have entered into a database (using `USE`), you can begin creating tables (or relations) within them

The following is the general syntax:

```
CREATE TABLE table_name(
    col1 type CONSTRAINT,
    col2 type CONSTRAINT,
    col3 type CONSTRAINT
);
```

Keep in mind the following:

- Separate each column definition by ending the row with a comma and moving to the next line.
- The **last** column definition will not have a comma at the end.
- Add a semicolon after the closing bracket or the command will not be executed!

Let's say we want to create a table called `qprog_emps` (we aren't adding values as of now).

Srno	Name	Email
1	Arjun Sodhi	arjun.sodhi@qprogramming.net
2	Sammarth Kumar	sam@qprogramming.net

The following will be the syntax:

```
mysql> create table qprog_emps(
    -> Srno INT PRIMARY KEY, # Srno must be integer and primary key
    -> Name VARCHAR(40),      # Name of the emp must be with characters
    -> Email VARCHAR(40)      # Remember there is no comma here!
    -> );                     # Put semi-colon after closing bracket
Query OK, 0 rows affected (0.11 sec)
```

Remember, right now we are dealing with **Data Definition**, so there are no records in the table as of now.

We have just created the **schema** for the `qprog_emps` table.

`SHOW`

Similar to the `SHOW databses;` command you can run `SHOW tables;` once you are in a database. If you are not in a database you will receive an error.

If your database does not have any tables, you will receive an Empty Set.

Let's see what our `qprogramming` database looks like:

```
mysql> SHOW tables;
+-------------------------+
| Tables_in_qprogramming |
+-------------------------+
| qprog_emps              |
+-------------------------+
1 row in set (0.07 sec)
```

`DESCRIBE`

To see the **schema** of a table, you can use the command `DESCRIBE table_name;`.

```
mysql> DESCRIBE qprog_emps;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| Srno  | int         | NO   | PRI | NULL    |       |
| Name  | varchar(40) | YES  |     | NULL    |       |
| Email | varchar(40) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.06 sec)
```

`ALTER`

We use the `ALTER` command to modify the table schema in multiple ways.

Add Keys

1. **Primary key**

   We can add a primary key to the table by executing:

   ```
   ALTER TABLE tablename
   ADD PRIMARY KEY(column_name);
   ```

   For example,

   ```
   mysql> ALTER TABLE qprog_emps ADD PRIMARY KEY (Srno);
   Query OK, 0 rows affected (0.05 sec)
   Records: 0  Duplicates: 0  Warnings: 0
   ```

   Remember you don't need to do this if you already added a primary key while defining the table!

2. **Foreign key**

   If you want to add a foreign key in table1 which references a key in table 2:

   ```sql
   ALTER TABLE table1
   ADD FOREIGN KEY (column_name)
   REFERENCES table2(referenced_column);
   ```

   Make sure the column in table2 is a **primary key**.

   Note: Notice how we split the command into multiple lines. While this is not necessary for the command to work, it increases readability and you should write your SQL commands in this format in your exams to make it easier for the examiner to check.

   Often, answer keys will also have this formatting so it will be beneficial for you.

3. **Adding UNIQUE Constraint**

   ```
   
   ```

ALTER TABLE tablename
ADD UNIQUE (column_name);

~~~

   For example, we want to make sure all values of the emails are unique in `qprog_emps`. We can do it as follows

   ```mysql
mysql> ALTER TABLE qprog_emps
       -> ADD UNIQUE (Email);
   Query OK, 0 rows affected (0.11 sec)
   Records: 0  Duplicates: 0  Warnings: 0
~~~

To make sure it has run properly, you run the `describe` command to see

```
mysql> DESCRIBE qprog_emps;
   +----------+-------------+------+-----+---------+-------+
   | Field    | Type        | Null | Key | Default | Extra |
   +----------+-------------+------+-----+---------+-------+
   | Srno     | int         | NO   | PRI | NULL    |       |
   | Name     | varchar(40) | YES  |     | NULL    |       |
   | Email    | varchar(40) | YES  | UNI | NULL    |       |
   +----------+-------------+------+-----+---------+-------+
```

As you can see, "UNI" is written for Email, which is short for unique

4. **Add Column to table**
   If you want to add a column to an already existing table, you can add it using the following syntax:

   ```sql
   ALTER TABLE tablename
   ADD column_name datatype;
   ```

For example, if we want to add a phone number column to the `qprog_emps` table, we can do it as follows:

```
mysql> ALTER TABLE qprog_emps
   -> ADD phone_no int;
Query OK, 0 rows affected (0.10 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Now if you execute the describe command, you will see the `phone_no` column showing up as a field

```
mysql> DESCRIBE qprog_emps;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| Srno     | int         | NO   | PRI | NULL    |       |
| Name     | varchar(40) | YES  |     | NULL    |       |
| Email    | varchar(40) | YES  | UNI | NULL    |       |
| phone_no | int         | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
```

5. **Add Default Value**
If you need to add a default value to a specific column of your table, you may use the following syntax:

```
ALTER TABLE tablename
MODIFY column_name datatype
DEFAULT default_val;
```

Remember, by doing this, we are ensuring that the column will never receive a `NULL` value, it will have a default value instead.

For example, let us make the default Name in our table `qprog_emps` as "John Doe"

```
mysql> ALTER TABLE qprog_emps
    -> MODIFY Name varchar(40)
    -> DEFAULT "John Doe";  # Remember to put " " around it, it is a string!
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Now if you execute the describe command, you will see

```
mysql> DESCRIBE qprog_emps;
+----------+-------------+------+-----+----------+-------+
| Field    | Type        | Null | Key | Default  | Extra |
+----------+-------------+------+-----+----------+-------+
| Srno     | int         | NO   | PRI | NULL     |       |
| Name     | varchar(40) | YES  |     | John Doe |       |
| Email    | varchar(40) | YES  | UNI | NULL     |       |
| phone_no | int         | YES  |     | NULL     |       |
+----------+-------------+------+-----+----------+-------+
4 rows in set (0.06 sec)
```

Notice how the "Default" for Name is not NULL, but instead John Doe. This way it will never be NULL

6. **Modify Datatype of Column**

   To modify the datatype of a column, we can use the syntax:

   ```
   ALTER TABLE tablename
   MODIFY column_name datatype;  # Write the new datatype here
   ```

   For example, we want to change the phone number column's datatype to `varchar(10)` instead of int, we can do:

   ```
   mysql> ALTER TABLE qprog_emps
       -> MODIFY phone_no VARCHAR(10);
   Query OK, 0 rows affected (0.05 sec)
   Records: 0  Duplicates: 0  Warnings: 0
   ```

   Now the datatype of the `phone_no` column is `varchar(10)` instead of `int`.

7. **Modify Constraints**

   To change the constraints you have applied on any column of your table, you can use

   ```
   ALTER TABLE tablename
   MODIFY column_name datatype constraint;
   ```

8. **Remove Column**
   You can remove a column from the table as follows:

   ```
   ALTER TABLE tablename
   DROP column_name;
   ```

9. **Remove Primary Key**

   If you want to remove the `PRIMARY KEY` constraint from an attribute you can execute the following command:

   ```
   ALTER tablename
   DROP PRIMARY KEY;
   ```

DROP

If you want to remove a table or database, you can execute the following:

```
DROP TABLE name;
```

or

```
DROP DATABASE name;
```

Data Manipulation(DML)

`INSERT`

Insert one record

```
INSERT INTO tablename
VALUE (val1, val2, val3, ...);
```

Insert multiple records

```
INSERT INTO tablename VALUES
(R1C1, R1C2, R1C3, ...),
(R2C1, R2C2, R2C3, ...),
(R3C1, R3C2, R3C3, ...),
.......................;
```

Insert to specific attributes

```
INSERT INTO tablename (Col1, Col2, Col3)
VALUE (Val1, Val2, Val3);
```

Let's now insert records to the `qprog_emps` table:

srno	name	email	phone_no
1	Arjun Sodhi	arjun.sodhi@qprogramming.net	9910234453
2	Sammarth Kumar	sam@qprogramming.net	9510238481
3	Balpreet Singh	balpreet@qprogramming.net	9323421232

```
INSERT INTO qprog_emps VALUES
(1, "Arjun Sodhi", "arjun.sodhi@qprogramming.net", "9910234453"),
(2, "Sammarth Kumar", "sam@qprogramming.net", "9510238481"),
(3, "Balpreet Singh", "balpreet@qprogramming.net", "9323421232");
```

Output:

```
mysql> INSERT INTO qprog_emps VALUES
    -> (1, "Arjun Sodhi", "arjun.sodhi@qprogramming.net", "9910234453"),
    -> (2, "Sammarth Kumar", "sam@qprogramming.net", "9510238481"),
    -> (3, "Balpreet Singh", "balpreet@qprogramming.net", "9323421232");
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Data Queries

`SELECT`

`SELECT` is used to retrieve particular data from the tables.

Syntax

```sql
SELECT col1, col2, ...
FROM table
WHERE condition;
```

Let's start by selecting all the rows from `qprog_emps`

```sql
SELECT *
FROM qprog_emps
```

Output

```
+------+----------------+----------------------------+------------+
| Srno | Name           | Email                      | phone_no   |
+------+----------------+----------------------------+------------+
|    1 | Arjun Sodhi    | arjunsodhi@qprogramming.net | 9910234453 |
|    2 | Sammarth Kumar | sam@qprogramming.net       | 9510238481 |
|    3 | Balpreet Singh | balpreet@qprogramming.net  | 9323421232 |
+------+----------------+----------------------------+------------+
```

As you can see, it returns all the rows in the table.

What if we want to return a particular column, like `name`?

```sql
SELECT name
FROM qprog_emps;
```

Output

```
+----------------+
| name           |
+----------------+
| Arjun Sodhi    |
| Sammarth Kumar |
| Balpreet Singh |
+----------------+
```

Aliasing

We can "rename" the column in the output only by using `AS`. This **will not** affect the database schema.

Syntax

```sql
SELECT colname AS newname
FROM table
```

For example,

```sql
SELECT phone_no AS Contact
FROM qprog_emps;
```

Output

```
+------------+
| Contact    |
+------------+
| 9910234453 |
| 9510238481 |
| 9323421232 |
+------------+
```

Let the table `qprog_emps` have columns called "role" and "age" with the following values:

qprog_emps

srno	name	email	phone_no	role	Age
1	Arjun Sodhi	arjun.sodhi@qprogramming.net	9910234453	Content Writer	17
2	Sammarth Kumar	sam@qprogramming.net	9510238481	Content Writer	18
3	Balpreet Singh	balpreet@qprogramming.net	9323421232	Video team	18
4	Nihaara	NULL	9321345664	Social media	15

This table will be used for all future examples.

`DISTINCT` Clause

This keyword allows us to return unique records.

Syntax

```sql
SELECT DISTINCT colname
FROM table
```

If we run the command to display the distinct roles in the `qprog_emps` table, this will be the query:

```sql
SELECT DISTINCT role
FROM qprog_emps
```

Output:

```
+-----------------+
| role            |
+-----------------+
| Content Writer |
| Video Team      |
| Social Media    |
+-----------------+
```

`WHERE` Clause

It is used to add conditions to the data being retrieved.

The conditions can be written using comparison operators similar to Boolean operators in programming languages.

Comparison Operators in SQL

Operator	Description
=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
!=	Not equal to

Syntax

```sql
SELECT col1, col2, ...
FROM table
WHERE <condition>
```

Using the `qprog_emps` table

```sql
SELECT srno, name
FROM qprog_emps
WHERE role = "Content Writer";
```

Output:

```
+------+----------------+
| srno | name           |
+------+----------------+
|    1 | Arjun Sodhi    |
|    2 | Sammarth Kumar |
+------+----------------+
```

You can also use comparison operators to check for greater than and less than

```sql
SELECT name
FROM qprog_emps
WHERE age >= 18;
```

Output:

```
+----------------+
| name           |
+----------------+
| Sammarth Kumar |
| Balpreet Singh |
+----------------+
```

You can connect multiple conditions using `AND` , `OR` or `NOT`

```sql
SELECT name
FROM qprog_emps
WHERE age >= 18 AND role = "Content Writer";
```

Output:

```
+----------------+
| name           |
+----------------+
| Sammarth Kumar |
+----------------+
```

`IN`

It is used to check whether a value belongs to a set of values

Syntax:

```sql
SELECT col1, col2
FROM table
WHERE col IN (val1, val2, ...); # Represent all values by separating by comma in circular brackets
```

For example,

```
SELECT name
FROM qprog_emps
WHERE role IN ("Content Writer", "Video Team");
```

Output:

```
+----------------+
| name           |
+----------------+
| Arjun Sodhi    |
| Sammarth Kumar |
| Balpreet Singh |
+----------------+
```

`BETWEEN`

It is used to check whether a value lies in a range of values.

Remember the upper and lower limit are **both** included

Syntax:

```
SELECT col1, col2, ...
FROM table
WHERE col BETWEEN a AND b;  # a and b are the limits
```

For example

```
SELECT name, age
FROM qprog_emps
WHERE age BETWEEN 16 AND 18;
```

Output:

```
+----------------+------+
| name           | age  |
+----------------+------+
| Arjun Sodhi    |   17 |
| Sammarth Kumar |   18 |
| Balpreet Singh |   18 |
+----------------+------+
```

`ORDER BY`

Used to order the retrieved data on the basis of a particular column

Syntax:

```
SELECT col1, col2, ...
FROM table
ORDER BY col;
```

In the order by line, you can write `ASC`for ascending and `DESC` for descending and it will select the data accordingly. The default is ascending

For example,

```
SELECT name,age
FROM qprog_emps
ORDER BY age ASC; # Will also work if you don't write ASC
```

Output

```
+----------------+------+
| name           | age  |
+----------------+------+
| Nihaara        |   15 |
| Arjun Sodhi    |   17 |
| Sammarth Kumar |   18 |
| Balpreet Singh |   18 |
+----------------+------+
```

Using `DESC`

```
SELECT name,age
FROM qprog_emps
ORDER BY age DESC;
```

Output

```
+----------------+------+
| name           | age  |
+----------------+------+
| Sammarth Kumar |   18 |
| Balpreet Singh |   18 |
| Arjun Sodhi    |   17 |
| Nihaara        |   15 |
+----------------+------+
```

`NULL`

We can check whether a value is NULL or not

```
SELECT col1, col2, ...
FROM table
WHERE col IS NULL;
```

To check whether a value is not NULL, we can use

```
SELECT col1,col2,..
FROM table
WHERE col IS NOT NULL;
```

For example,

```
SELECT name
FROM qprog_emps
WHERE email IS NULL;
```

Output

```
+---------+
| name    |
+---------+
| Nihaara |
+---------+
```

Using `IS NOT NULL`

```
SELECT name
FROM qprog_emps
WHERE email IS NOT NULL;
```

Output:

```
+----------------+
| name           |
+----------------+
| Arjun Sodhi    |
| Balpreet Singh |
| Sammarth Kumar |
+----------------+
```

Substring Pattern Matching

Substring pattern matching is used when you want to extract all values (which are strings) with a particular substring in it. There are 2 types of substring pattern matchings:

- **%**: This is used to denote an *arbitrary* amount of characters. For example "A%" means it will match with any strings beginning with "A", regardless of how many characters it has.
- _: This is used to denote a *single* character. For example, "A _ _ _ _ _" will match with all strings beginning with the letter "A" of length 6, including the first letter (notice there are 5 underscores)

These are used in the `WHERE` clause of SQL queries, with the `LIKE` operator

Syntax

```
SELECT col1, col2, ...
FROM table
WHERE col LIKE pattern_str; # pattern_str is the substring you want to match with
```

For example, using "%" substring matching

```
SELECT srno, name, phone_no
FROM qprog_emps
WHERE phone_no LIKE "93%"; # will give all records where phone number begins with 93
```

Output

```
+------+----------------+------------+
| srno | name           | phone_no   |
+------+----------------+------------+
|    3 | Balpreet Singh | 9323421232 |
|    4 | Nihaara        | 9321345664 |
+------+----------------+------------+
```

For example, using "_" substring matching

```
SELECT srno, name
FROM qprog_emps
WHERE name like "N______"; #there are 6 underscores, ie. 7 letter strings will be returned
```

Output

```
+------+---------+
| srno | name    |
+------+---------+
|    4 | Nihaara |
+------+---------+
```

Some other kinds of substrings could be:

- `A%n` which represents any string beginning with A and ending with n
- `S_m%` which represents any string beginning S _ m.

Updation and Deletion

For both `UPDATE` and `DELETE` you must add a `WHERE` clause otherwise all the records would be affected by default.

UPDATE

We can modify the values of records based on a given condition

Syntax

```
UPDATE table
SET col1 = val1, col2 = val2, col3 = val3, ...
WHERE <condition>;
```

For example, if we want to change the age of srno 1 in the table `qprog_emps` to 18, we will use the following

```
UPDATE qprog_emps
SET age = 18
WHERE srno = 1;
```

```
mysql> UPDATE qprog_emps
    -> set age = 18
    -> where srno = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Srno	Name	Email	phone_no	role	age
1	Arjun Sodhi	arjun.sodhi@qprogramming.net	9910234453	Content Writer	18
2	Sammarth kumar	sam@qprogramming.net	9510238481	Content Writer	18
3	Balpreet singh	balpreet@qprogramming.net	9323421232	Video Team	18
4	Nihaara	NULL	9321345664	Social Media	15

This is the new table. Notice the value of age has been updated to 18 where the srno is 1. It was previously 17.

DELETE

We can remove records from a table based on a condition

Syntax

```
DELETE FROM table
WHERE <condition>;
```

For example, if we want to delete the value of srno 4 from the table `qprog_emps` , we can use the following

```
DELETE FROM qprog_emps
WHERE srno = 4;
```

Output

```
mysql> DELETE FROM qprog_emps
    -> WHERE srno = 4;
Query OK, 1 row affected (0.01 sec)
```

Srno	Name	Email	phone_no	role	age
1	Arjun Sodhi	arjun.sodhi@qprogramming.net	9910234453	Content Writer	18
2	Sammarth Kumar	sam@qprogramming.net	9510238481	Content Writer	18
3	Balpreet Singh	balpreet@qprogramming.net	9323421232	Video Team	18

> **Important:** While putting condition in the `WHERE` clause of the update/delete commands, it is recommended to use primary key (or any other candidate key) for comparison, since they uniquely identify every row.

Functions

SQL has some pre-defined functions which help in extracting data. These functions can either act on a single row or on multiple rows of a table.

Single Row (Scalar) Functions

These are the kind of functions which act on a single value and return a single value. There are 3 categories of these:

- **Math Functions**

Function	Description	Example

`POWER(x,y)`
or
`POW(x,y)`
	Returnsx^y	`mysql> SELECT POW(3,4);`
		Output: 81

`ROUND(n,d)`	Returns the number when `n` is rounded to `d` decimal places
	(d=0 means it returns the nearest integer)
	(d<0 means it will ignore the decimal part and round to integers that many places to the left)
		`mysql> SELECT ROUND(32.12234,2);`
		Output: 32.12
		`mysql> SELECT ROUND(34.6345,0);`
		Output: 35
		
		mysql> SELECT ROUND(34.6345,-1);
		Output: 30

`MOD(x,y)`	Returns remainder when x is divided by y (Same as the % operator in programming)	`mysql> SELECT MOD(37,5);` **Output:** 2

- **String functions**

> **Point to remember:** The positioning in SQL strings begins from 1, unlike in programming.

Function	Description	Example
`UCASE(str)` or `UPPER(str)`	Converts all characters of the string to uppercase and returns a new string	`mysql> SELECT UPPER("heLLo");` **Output:** HELLO
`LCASE(str)` or `LOWER(str)`	Converts all characters of the string to lowercase and returns a new string	`mysql> SELECT LOWER("heLLo");` **Output:** hello
`MID(str,pos,n)` or `SUBSTRING(str,pos,n)` or `SUBSTR(str,pos,n)`	Returns a substring of the original string of length `n` beginning from position `pos` If `n` is not provided, it returns substring starting from `pos` till end of string	`mysql> SELECT SUBSTRING("Qprogramming",5,3);` **Output:** gra **Explanation:** It will return positions 5,6 and 7 which are g,r and a respectively
`LENGTH(str)`	Returns length of string	`mysql> SELECT LENGTH("Qprogramming");` **Output:** 12
`LEFT(str,n)`	Returns a substring of the original string of length `n` from the left side	`mysql> SELECT LEFT("Qprogramming",4);` **Output:** Qpro
`RIGHT(str,n)`	Returns substring of original string of length `n` from the right side **Note:** Order is left to right	`mysql> SELECT RIGHT("Qprogramming",4);` **Output:** ming

`INSTR(str,substr)`	Returns position of the first occurrence specified substring `substr` in the original string
	Note: 0 is returned if substring is not present in string
		`mysql> SELECT INSTR("Qprogramming","og");`
		Output: 4
		`mysql> SELECT INSTR("Qprogramming","ap");`
		Output: 0

`LTRIM(str)`	Returns string by removing whitespaces from the left side	`mysql> SELECT LTRIM("     hello");` **Output:** hello
`RTRIM(str)`	Returns string by removing whitespaces from the right side	`mysql> SELECT RTRIM("hello     ");` **Output:** hello
`TRIM(str)`	Returns string by removing whitespaces from left and right of string(does LTRIM and RTRIM)	`mysql> SELECT TRIM("     hello         ");` **Output:** hello

- **Date & Time functions**
  The following is the standard format for date-time in SQL: `YYYY-MM-DD HH:MM:SS`

Function	Description	Example
`NOW()`	Returns the current date and time according to the system	`mysql> SELECT NOW();` **Output:** 2022-04-19 14:14:03

`DATE(d)`	Returns the date of the value provided	`mysql> SELECT DATE("2015-04-13 23:45:03");`
		Output: 2015-04-13
		`mysql> SELECT DATE(NOW());`
		Output: 2022-04-19

`MONTH(d)`	Returns the month of the value provided	`mysql> SELECT MONTH("2015-04-13");` **Output:** 4
`MONTHNAME(d)`	Returns the name of the month of the value provided	`mysql> SELECT MONTHNAME("2015-04-13");` **Output:** April
`YEAR(d)`	Returns the year of the value provided	`mysql> SELECT YEAR("2015-04-13");` **Output:** 2015
`DAY(d)`	Returns the day of the value provided	`mysql> SELECT DAY("2015-04-13");` **Output:** 13
`DAYNAME(d)`	Returns the day of the week of the value provided	`mysql> SELECT DAYNAME("2015-04-13");` **Output:** Monday

Aggregate (Multiple Row) Functions

These act on multiple rows of a table and returns a single value for each column
We will be using these tables for the following queries:

qprog_emps

srno	name	email	phone_no	role	age	date_join
1	Arjun Sodhi	arjun.sodhi@qprogramming.net	9910234453	Content Writer	17	2020-05-13
2	Sammarth Kumar	sam@qprogramming.net	9510238481	Content Writer	18	2020-04-03
3	Balpreet Singh	balpreet@qprogramming.net	9323421232	Video team	18	2020-06-13
4	Nihaara	NULL	9321345664	Social media	15	2021-05-08

These are the aggregate functions in SQL:

Function	Description	Example

`MAX(col)`	Returns the maximum value in the column	`mysql> SELECT MAX(age) FROM qprog_emps`
		Output: 18
		`mysql> SELECT MAX(date_join) FROM qprog_emps`
		Output: 2021-05-08

`MIN(col)`	Returns the minimum value in the column	`mysql> SELECT MIN(age) from qprog_emps`
		Output: 15
		`mysql> SELECT MIN(date_join) from qprog_emps`
		Output: 2020-04-03

`AVG(col)`	Returns the average of the values in the column
	Note: Value returned is always a float value
		`mysql> SELECT AVG(age) from qprog_emps`
		Output: 17.0000
		`mysql> SELECT AVG(DAY(date_join)) from qprog_emps`
		Output: 9.2500

`SUM(col)`	Returns the sum of the values in a column	`mysql> SELECT SUM(age) FROM qprog_emps` **Output:** 68
`COUNT(*)`	Returns the number of records in the table	`mysql> SELECT COUNT(*) FROM qprog_emps` **Output:** 4
`COUNT(col)`	Returns the number of records in the column, which does**not** include NULL values	`mysql> SELECT COUNT(email) FROM qprog_emps` **Output:** 3 **Explanation:** Email for srno 4 is NULL, so it does not count that

`GROUP BY` Clause

It is used to fetch and group rows with common values in a column. The aggregate functions also act on the data.

Syntax:

```sql
SELECT col1, col2, ...
FROM table
GROUP BY col;
```

For example,

```sql
SELECT role, COUNT(role) AS "Number of people" # Using aliasing for columns
FROM qprog_emps
GROUP BY role;
```

Output:

```
+----------------+------------------+
| role           | Number of people |
+----------------+------------------+
| Content Writer |                2 |
| Video Team     |                1 |
| Social Media   |                1 |
+----------------+------------------+
```

The `GROUP BY` clause should be used to group data by using aggregate functions.

`HAVING` Clause

It is used to impose conditions on the data being grouped.

```sql
SELECT col1,col2,...
FROM table
GROUP BY col
HAVING <condition>;
```

Important: Make sure to use `HAVING` when the condition is on some aggregate function

For example,

```sql
SELECT role, COUNT(role) AS "Number of people"
FROM qprog_emps
GROUP BY role
HAVING COUNT(role) > 1;
```

Output:

```
+----------------+------------------+
| role           | Number of people |
+----------------+------------------+
| Content Writer |                2 |
+----------------+------------------+
```

If in this query, if we use `WHERE COUNT(role) > 1` and instead used `HAVING COUNT(role) > 1`, we would get an error

```
mysql> SELECT role,COUNT(role)
    -> FROM qprog_emps
    -> WHERE COUNT(role)>1
    -> GROUP BY role;
ERROR 1111 (HY000): Invalid use of group function
```

To impose conditions on aggregate functions, you **must** use `HAVING`

Sequence of commands

Now that you know the parts of a SQL query, remember this sequence of commands which you must follow while writing any query in SQL

```
SELECT ...
FROM ...
WHERE ...
GROUP BY ...
HAVING ...
ORDER BY ...
```

Queries should be written in **this order only**

Cartesian Product / Cross join

The cartesian product is the result of joining all tuples(rows) of one table to that of another table

If the first table has m_1 rows and n_1 columns, then the second table has m_2 rows and n_2 columns, then the resulting cartesian product table has $(m_1 \times m_2)$ rows and $(n_1 + n_2)$ columns

To perform a cartesian product we execute:

```
SELECT *
FROM table1, table2;
```

or

```
SELECT *
FROM table1 CROSS JOIN table2;
```

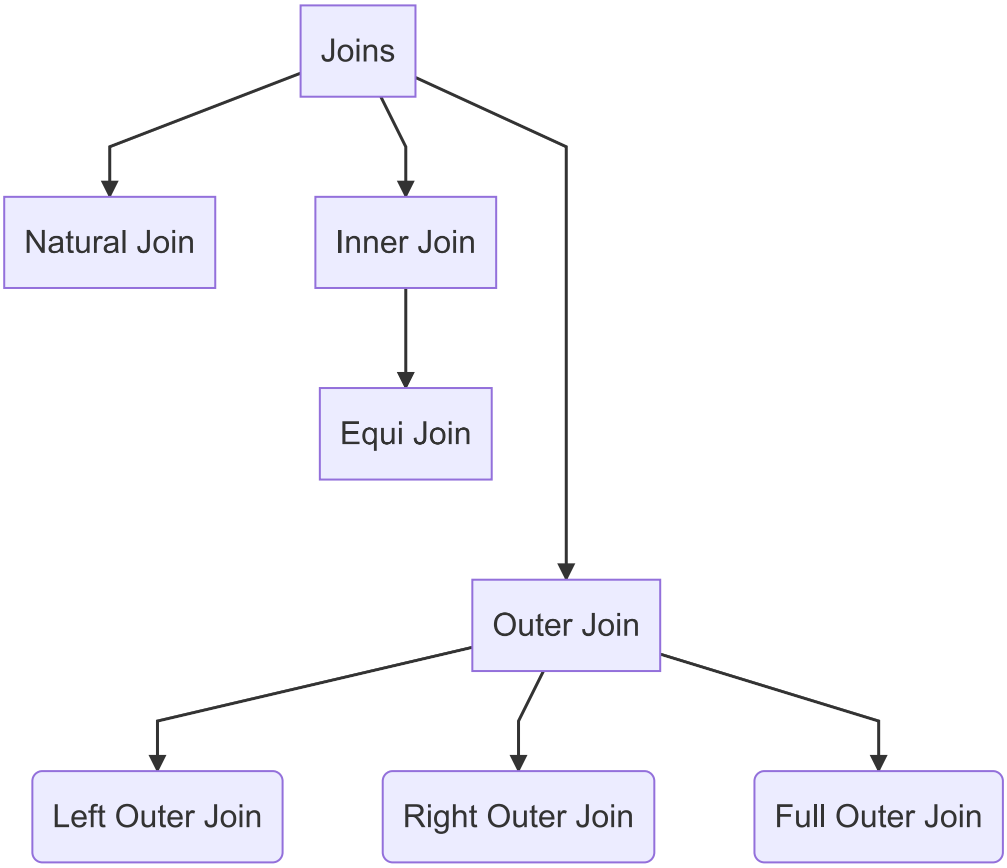
For example,

```
mysql> SELECT * FROM table1;
+------+------+
| a    | b    |
+------+------+
|    1 |    2 |
|    3 |    4 |
+------+------+
2 rows in set (0.00 sec)

mysql> SELECT * FROM table2;
+------+------+
| c    | d    |
+------+------+
|    5 |    6 |
|    7 |    8 |
+------+------+
2 rows in set (0.00 sec)

mysql> SELECT * FROM table1, table2;
+------+------+------+------+
| a    | b    | c    | d    |
+------+------+------+------+
|    1 |    2 |    5 |    6 |
|    3 |    4 |    5 |    6 |
|    1 |    2 |    7 |    8 |
|    3 |    4 |    7 |    8 |
+------+------+------+------+
4 rows in set (0.00 sec)
```

Notice how every row of `table1` was paired to every row of `table2`.

Joins

Joins in SQL can be classified in the following ways



We will be using the following tables for examples to understand joins:

Soccer

srno	name	class
1	Charles	11B
2	John	12A
3	William	11C
4	Micheal	11A
5	Joe	12D

Badminton

number	name	section
1	Charles	B
2	Jordan	B
3	Joe	C

Natural Join

To join two tables, you must have a column with the **same name and datatype** in both tables to join.

Syntax:

```
SELECT *
FROM table1
NATURAL JOIN table2;
```

There is no repetition of a column, also known as **duplicity** in this case.

For example,

```
mysql> SELECT *
    -> FROM soccer
    -> NATURAL JOIN badminton;
+---------+------+-------+--------+---------+
| name    | srno | class | number | section |
+---------+------+-------+--------+---------+
| Charles |    1 | 11B   |      1 | B       |
| Joe     |    5 | 12D   |      3 | C       |
+---------+------+-------+--------+---------+
2 rows in set (0.00 sec)
```

Notice how the common column `name` was the one used to join the two tables. It only came once in the resultant table as the first column.

Equi Join/Inner join

This involves comparison for equality of columns by which you want to join two tables.

Syntax:

```
SELECT *
FROM table1, table2
WHERE table1.col = table2.col;
```

or

```
SELECT *
FROM table1
INNER JOIN table2
ON table1.col = table2.col;
```

For example,

```
mysql> SELECT *
    -> FROM soccer, badminton
    -> WHERE soccer.name = badminton.name;
+------+---------+-------+--------+---------+---------+
| srno | name    | class | number | name    | section |
+------+---------+-------+--------+---------+---------+
|    1 | Charles | 11B   |      1 | Charles | B       |
|    5 | Joe     | 12D   |      3 | Joe     | C       |
+------+---------+-------+--------+---------+---------+
2 rows in set (0.06 sec)
```

The common column `name` is repeated in this case, as you can see.

> **Important Question:** Differentiate between Natural Join and Equijoin
>
> Key points:
>
> Natural Join: No duplicity, not required to state columns used to join
>
> Equi Join: Duplicity, required to state columns used to join

Outer Joins

- **Left Outer Join**

  This will return all records from the first table(the table on the left) and the common records in the second table(the table on the right)

  The values in the left table and not the right table are shown as `NULL` values.

  Syntax:

  ```
  SELECT *
  FROM table1
  LEFT JOIN table2
  ON table1.col = table2.col
  ```

  For example,

  ```
  mysql> SELECT *
      -> FROM soccer
      -> LEFT JOIN badminton
      -> ON soccer.name=badminton.name;
  +------+---------+-------+--------+---------+---------+
  | srno | name    | class | number | name    | section |
  +------+---------+-------+--------+---------+---------+
  |    1 | Charles | 11B   |      1 | Charles | B       |
  |    2 | John    | 12A   |   NULL | NULL    | NULL    |
  |    3 | William | 11C   |   NULL | NULL    | NULL    |
  |    4 | Micheal | 11A   |   NULL | NULL    | NULL    |
  |    5 | Joe     | 12D   |      3 | Joe     | C       |
  +------+---------+-------+--------+---------+---------+
  5 rows in set (0.05 sec)
  ```

Notice how all the values not in the `badminton` table having common value of `name` with `soccer` came as `NULL`.

- **Right Outer Join**

  This will return all records from the second table(the table on the right) and the common records in the first table(the table on the left)

  The values in the right table and not the left table are shown as `NULL` values.

  Syntax:

```
SELECT *
FROM table1
RIGHT JOIN table2
ON table1.col=table2.col
```

  For example,

```
mysql> SELECT *
    -> FROM soccer
    -> RIGHT JOIN badminton
    -> ON soccer.name = badminton.name;
+------+---------+-------+--------+---------+---------+
| srno | name    | class | number | name    | section |
+------+---------+-------+--------+---------+---------+
|    1 | Charles | 11B   |      1 | Charles | B       |
| NULL | NULL    | NULL  |      2 | Jordan  | B       |
|    5 | Joe     | 12D   |      3 | Joe     | C       |
+------+---------+-------+--------+---------+---------+
```

- **Full Outer Join**
  This will return the records which have values in either the **left table or the right table**. It is basically the union of the `left join` and `right join` commands.

- **Syntax:**

```
SELECT * FROM table1 LEFT JOIN table2 ON table1.col = table2.col
UNION
SELECT * FROM table1 RIGHT JOIN table2 ON table1.col = table2.col;
```

  For example

```
mysql> SELECT * FROM soccer LEFT JOIN badminton ON soccer.name=badminton.name
    -> UNION
    -> SELECT * FROM soccer RIGHT JOIN badminton ON soccer.name=badminton.name;
+------+---------+-------+--------+---------+---------+
| srno | name    | class | number | name    | section |
+------+---------+-------+--------+---------+---------+
|    1 | Charles | 11B   |      1 | Charles | B       |
|    2 | John    | 12A   |   NULL | NULL    | NULL    |
|    3 | William | 11C   |   NULL | NULL    | NULL    |
|    4 | Micheal | 11A   |   NULL | NULL    | NULL    |
|    5 | Joe     | 12D   |      3 | Joe     | C       |
| NULL | NULL    | NULL  |      2 | Jordan  | B       |
+------+---------+-------+--------+---------+---------+
6 rows in set (0.06 sec)
```

NCERT

Q1. Answer the following questions:

a) Define RDBMS. Name any two RDBMS software.

RDBMS stands for Relational Database Management System, which is a software to manage, query and retrieve data from relational databases.
Examples are MySQL and Oracle.

b) What is the purpose of the following clauses in a select statement?

i) ORDER BY

The `ORDER BY` statement sorts the data being retrieved in ascending or descending order based on a specific column.

ii) GROUP BY

The `GROUP BY` statement groups the rows together that contain the same values in a particular column.

c) Cite any two differences between Single Row Functions and Aggregate Functions.

Single Row Functions	Aggregate Functions
It operates on a single row	It operates on multiple rows
It returns a single result for one row	It returns a single result for all the rows

d) What do you understand by Cartesian Product?

The Cartesian Product of two tables is obtained when every row of one table is paired with every row of another table to create the resultant table. If the first table has `m` rows and the second has `n` rows, then the resultant table has $m \times n$ rows and $m + n$ columns.

e) Differentiate between the following statements:

i) ALTER and UPDATE

Alter	Update
This is used to alter the tables schema	This is used to update the values stored in a table
This is a DDL command	This is a DML command

ii) DELETE and DROP

DELETE	DROP
This is used to delete values stored in a table	This is used to drop or remove columns, tables or databases
This is a DML command	This is a DDL command

f) Write the name of the functions to perform the following operations:
i) To display the day like “Monday”, “Tuesday”, from the date when India got independence.

```
mysql> SELECT DAYNAME("1947-08-15");
+-----------------------+
| DAYNAME("1947-08-15") |
+-----------------------+
| Friday                |
+-----------------------+
1 row in set (0.02 sec)
```

ii) To display the specified number of characters from a particular position of the given string.

```
mysql> SELECT SUBSTRING("Q-Programming",6,4);    # You can also use MID() or SUBSTR()
+--------------------------------+
| SUBSTRING("Q-Programming",6,4) |
+--------------------------------+
| gram                           |
+--------------------------------+
1 row in set (0.00 sec)
```

iii) To display the name of the month in which you were born.

```
mysql> SELECT MONTHNAME("2004-06-12");
+-------------------------+
| MONTHNAME("2004-06-12") |
+-------------------------+
| June                    |
+-------------------------+
1 row in set (0.01 sec)
```

iv) To display your name in capital letters.

```
mysql> SELECT UPPER("arjun");
+----------------+
| UPPER("arjun") |
+----------------+
| ARJUN          |
+----------------+
1 row in set (0.01 sec)
```

2. Write the output produced by the following SQL statements:

a) SELECT POW(2,3);

```
mysql> SELECT POW(2,3);
+----------+
| POW(2,3) |
+----------+
|        8 |
+----------+
1 row in set (0.00 sec)
```

b) SELECT ROUND(342.9234, -1);

```
mysql> SELECT ROUND(342.9234,-1);
+--------------------+
| ROUND(342.9234,-1) |
+--------------------+
|                340 |
+--------------------+
1 row in set (0.05 sec)
```

c) SELECT LENGTH("Informatics Practices");

```
mysql> SELECT LENGTH("Informatics Practices");
+---------------------------------+
| LENGTH("Informatics Practices") |
+---------------------------------+
|                              21 |
+---------------------------------+
1 row in set (0.00 sec)
```

d) SELECT YEAR(“1979/11/26”), MONTH(“1979/11/26”), DAY(“1979/11/26”), MONTHNAME(“1979/11/26”);

YEAR("1979/11/26")	MONTH("1979/11/26")	DAY("1979/11/26")	MONTHNAME("1979/11/26")
1979	11	26	November

e) SELECT LEFT("INDIA",3), RIGHT("Computer Science",4), MID("Informatics",3,4), SUBSTR("Practices",3);

LEFT("INDIA",3)	RIGHT("Computer Science",4)	MID("Informatics",3,4)	SUBSTR("Practices",3)
IND	ence	form	actices

3. Consider the following MOVIE table and write the SQL queries based on it.

MovieID	MovieName	Category	ReleaseDate	ProductionCost	BusinessCost
001	Hindi_Movie	Musical	2018-04-03	124500	130000
002	Tamil_Movie	Action	2016-05-17	112000	118000
003	English_Movie	Horror	2017-08-06	245000	360000
004	Bengali_Movie	Adventure	2017-01-04	72000	100000
005	Telugu_Movie	Action	-	100000	-
006	Punjabi_Movie	Comedy	-	30500	-

a) Display all the information from the Movie table.

```
SELECT *
FROM MOVIE;
```

b) List business done by the movies showing only MovieID, MovieName and Total_Earning. Total_Earning to be calculated as the sum of ProductionCost and BusinessCost.

```
SELECT MovieID, MovieName, (ProductionCost + BusinessCost) as "Total_Earning"
FROM MOVIE;
```

c) List the different categories of movies.

```
SELECT DISTINCT Category # To show each category once
FROM MOVIE;
```

d) Find the net profit of each movie showing its MovieID, MovieName and NetProfit. Net Profit is to be calculated as the difference between Business Cost and Production Cost.

```
SELECT MovieID, MovieName, (BusinessCost - ProductionCost) as "NetProfit"
FROM MOVIE;
```

e) List MovieID, MovieName and Cost for all movies with ProductionCost greater than 10,000 and less than 1,00,000.

```
SELECT MovieID, MovieName, ProductionCost
FROM MOVIE
WHERE ProductionCost between 10000 and 100000;
```

f) List details of all movies which fall in the category of comedy or action.

```
SELECT *
FROM MOVIE
WHERE Category IN ("Comedy","Action");
```

or

```sql
SELECT *
FROM MOVIE
WHERE Category="Comedy" OR Category="Action";
```

g) List details of all movies which have not been released yet.

```sql
SELECT *
FROM MOVIE
WHERE NOW() < RELEASEDATE;  # Not been released yet, then date right now is less than release date
```

4. Suppose your school management has decided to conduct cricket matches between students of Class XI and Class XII. Students of each class are asked to join any one of the four teams – Team Titan, Team Rockers, Team Magnet and Team Hurricane. During summer vacations, various matches will be conducted between these teams. Help your sports teacher to do the following:
a) Create a database “Sports”.

```sql
CREATE DATABASE SPORTS;
```

b) Create a table “TEAM” with following considerations:
i) It should have a column TeamID for storing an integer value between 1 to 9, which refers to unique identification of a team.
ii) Each TeamID should have its associated name (TeamName), which should be a string of length not less than 10 characters.

```sql
CREATE TABLE TEAM(
    TeamID int,
    TeamName varchar(20),
    CHECK(TeamID >= 1 AND TeamID <= 9 AND LENGTH(TeamName) > 9)
);
```

c) Using table level constraint, make TeamID as the primary key.

```sql
ALTER TABLE TEAM
ADD PRIMARY KEY(TeamID);
```

d) Show the structure of the table TEAM using a SQL statement.

```sql
DESCRIBE TEAM;
```

e) As per the preferences of the students four teams were formed as given below. Insert these four rows in TEAM table:

Row 1: (1, Team Titan)
Row 2: (2, Team Rockers)
Row 3: (3, Team Magnet)
Row 4: (4, Team Hurricane)

```
INSERT INTO TEAM VALUES
    (1,"Team Titan"),
    (2,"Team Rockers"),
    (3,"Team Magnet"),
    (4,"Team Hurricane");
```

f) Show the contents of the table TEAM using a DML statement.

```
SELECT *     # SELECT is a DML Statement
FROM TEAM;
```

g) Now create another table MATCH_DETAILS and insert data as shown below. Choose appropriate data types and constraints for each attribute.

Table: MATCH_DETAILS

MatchID	MatchDate	FirstTeamID	SecondTeamID	FirstTeamScore	SecondTeamStore
M1	2018-07-17	1	2	90	86
M2	2018-07-18	3	4	45	48
M3	2018-07-19	1	3	78	56
M4	2018-07-29	2	4	56	67
M5	2018-07-18	1	4	32	87
M6	2018-07-17	2	3	67	51

```
CREATE TABLE MATCH_DETAILS(
    MatchID VARCHAR(3) PRIMARY KEY,
    MatchDate date,
    FirstTeamID int,
    SecondTeamID int,
    FirstTeamScore int,
    SecondTeamScore int
);

INSERT INTO MATCH_DETAILS VALUES
    ("M1","2018-07-17",1,2,90,86),
    ("M2","2018-07-18",3,4,45,48),
    ("M3","2018-07-19",1,3,78,56),
    ("M4","2018-07-19",2,4,56,67),
    ("M5","2018-07-18",1,4,32,87),
    ("M6","2018-07-17",2,3,67,51);
```

**5. Using the sports database containing two relation (TEAM, MATCH_DETAILS) and write the queries for the
following:**
a) Display the MatchID of all those matches where both the teams have scored more than 70.
b) Display the MatchID of all those matches where FirstTeam has scored less than 70 but SecondTeam has scored more than 70.
c) Display the MatchID and date of matches played by Team 1 and won by it.

d) Display the MatchID of matches played by Team 2 and not won by it.
e) Change the name of the relation TEAM to T_DATA.
Also change the attributes TeamID and TeamName to T_ID and T_NAME respectively.

The two tables:

```
mysql> select * from TEAM;
+--------+----------------+
| TeamID | TeamName       |
+--------+----------------+
|      1 | Team Titan     |
|      2 | Team Rockers   |
|      3 | Team Magnet    |
|      4 | Team Hurricane |
+--------+----------------+
4 rows in set (0.00 sec)

mysql> select * from Match_Details;
+---------+------------+-------------+--------------+----------------+-----------------+
| matchid | MatchDate  | FirstTeamID | SecondTeamID | FirstTeamScore | SecondTeamScore |
+---------+------------+-------------+--------------+----------------+-----------------+
| M1      | 2018-07-17 |           1 |            2 |             90 |              86 |
| M2      | 2018-07-18 |           3 |            4 |             45 |              48 |
| M3      | 2018-07-19 |           1 |            3 |             78 |              56 |
| M4      | 2018-07-19 |           2 |            4 |             56 |              67 |
| M5      | 2018-07-18 |           1 |            4 |             32 |              87 |
| M6      | 2018-07-17 |           2 |            3 |             67 |              51 |
+---------+------------+-------------+--------------+----------------+-----------------+
6 rows in set (0.00 sec)
```

a)

```
SELECT MatchID
FROM MATCH_DETAILS
WHERE FirstTeamScore > 70 AND SecondTeamScore > 70;
```

b)

```
SELECT MatchID
FROM MATCH_DETAILS
WHERE FirstTeamScore < 70 AND SecondTeamScore > 70;
```

c)

```
SELECT MatchID, MatchDate
FROM MATCH_DETAILS
WHERE (FirstTeamID = 1 AND FirstTeamScore > SecondTeamScore)
OR (SecondTeamID = 1 AND SecondTeamScore > FirstTeamScore);
```

We want those where Team number 1 played and won, so they could have played as first team or second team

d)

```
SELECT MATCHID
FROM MATCH_DETAILS
WHERE (FirstTeamID = 2 AND FirstTeamScore < SecondTeamScore)
OR (SecondTeamID = 2 AND SecondTeamScore < FirstTeamScore);
```

Same logic as part (c), except here we want records where Team 2 looses, hence the swap the inequality signs.

e)

```
ALTER TABLE TEAM
RENAME TO T_DATA;
```

```
ALTER TABLE T_DATA
RENAME COLUMN TeamID to T_ID;
```

```
ALTER TABLE T_DATA
RENAME COLUMN TeamName to T_NAME;
```

6. A shop called Wonderful Garments who sells school uniforms maintains a database SCHOOLUNIFORM as shown below. It consisted of two relations - UNIFORM and COST. They made UniformCode as the primary key for UNIFORM relations. Further, they used UniformCode and Size to be composite keys for COSTrelation. By analysing the database schema and database state, specify SQL queries to rectify the following anomalies.
a) M/S Wonderful Garments also keeps handkerchiefs of red colour, medium size of Rs. 100 each.
b) INSERT INTO COST (UCode, Size, Price) values (7, 'M',100);
When the above query is used to insert data, the values for the handkerchief without entering its details in the UNIFORM relation is entered. Make a provision so that the data can be entered in the COST table only if it is already there in the UNIFORM Table
c) Further, they should be able to assign a new UCode to an item only if it has a valid UName. Write a query to add appropriate constraints to the SCHOOLUNIFORM database.
d) Add the constraint so that the price of an item is always greater than zero.

The following are the relations (Used in examples on page 173 of the NCERT):

```
mysql> select * from cost;
+-------+------+-------+
| Ucode | size | price |
+-------+------+-------+
|     1 | L    |   580 |
|     1 | M    |   500 |
|     2 | L    |   890 |
|     2 | M    |   810 |
+-------+------+-------+
4 rows in set (0.10 sec)

mysql> select * from uniform;
+-------+-------+--------+
| Ucode | Uname | Ucolor |
+-------+-------+--------+
|     1 | Shirt | White  |
|     2 | Pant  | Grey   |
|     3 | Tie   | Blue   |
+-------+-------+--------+
3 rows in set (0.02 sec)
```

b) We want to make sure any values being inserted in cost table already has values in the Uniform table. To do so, we have to connect the two tables via a foreign key in the Cost table. To add the foreign key, the command is:

```
ALTER TABLE COST
ADD FOREIGN KEY(UCode)
REFERENCES UNIFORM(UCode);
```

Make sure while doing this that `Ucode` is already a primary key in the Uniform relation. Foreign key can only reference primary keys of another table

c)We want to make sure that one can add a UCode to an item only if it has a valid Uname. This means that Uname cannot be `NULL` , so to add that constraint we use

```
ALTER TABLE Uniform
MODIFY Uname varchar(20) NOT NULL;
```

Now if you print the table schema of UNIFORM you will see this constraint

d) To add this constraint, we have to use the `CHECK()` function

```
ALTER TABLE Cost
MODIFY price int CHECK(price>0);
```

7. Consider the following table named “Product”, showing details of products being sold in a grocery shop.

PCode	PName	UPrice	Manufacturer
P01	Washing Powder	120	Surf
P02	Toothpaste	54	Colgate
P03	Soap	25	Lux
P04	Toothpaste	65	Pepsodent
P05	Soap	38	Dove
P06	Shampoo	245	Dove

Write SQL queries for the following:
a) Create the table Product with appropriate data types and constraints.

```sql
CREATE TABLE Product(
    PCode varchar(4) PRIMARY KEY,
    PName varchar(25),
    UPrice int,
    Manufacturer varchar(30)
);

INSERT INTO Product VALUES
    ("P01","Washing Powder",120,"Surf"),
    ("P02","Toothpaste",54,"Colgate"),
    ("P03","Soap",25,"Lux"),
    ("P04","Toothpaste",65,"Pepsodent"),
    ("P05","Soap",38,"Dove"),
    ("P06","Shampoo",245,"Dove");
```

b) Identify the primary key in Product.

The primary key in the relation `Product` is `Pcode`, since every record in it uniquely identifies every row, and there is no repetition.

c) List the Product Code, Product name and price in descending order of their product name. If PName is the same, then display the data in ascending order of price.

```sql
SELECT PCode, PName, UPrice
FROM Product
ORDER BY PName DESC, UPrice ASC;
```

d) Add a new column Discount to the table Product.

```sql
ALTER TABLE Product
ADD Discount FLOAT NOT NULL;
```

e) Calculate the value of the discount in the table Product as 10 per cent of the UPrice for all those products where the UPrice is more than 100, otherwise the discount will be 0.

```sql
UPDATE Product
SET Discount = 0.1 * UPrice         # Set to 10%
WHERE UPrice > 100;
```

Writing `NOT NULL` constraint while defining it makes sure that it's default value is 0

f) Increase the price by 12 per cent for all the products manufactured by Dove.

```sql
UPDATE Product
SET UPrice = Uprice + 0.12 * UPrice  # Increase of 12%
WHERE Manufacturer = "Dove";
```

g) Display the total number of products manufactured by each manufacturer.

```sql
SELECT Manufacturer, COUNT(*)
FROM Product
GROUP BY Manufacturer;
```

Write the output(s) produced by executing the following queries on the basis of the information given above in the table Product:
We will be using the original table and not the modified version of it

PCode	PName	UPrice	Manufacturer
P01	Washing Powder	120	Surf
P02	Toothpaste	54	Colgate
P03	Soap	25	Lux
P04	Toothpaste	65	Pepsodent
P05	Soap	38	Dove
P06	Shampoo	245	Dove

Although the logic would be same in the modified version too

h) SELECT PName, Avg(UPrice) FROM Product GROUP BY Pname;

```
mysql> SELECT PName, Avg(UPrice) FROM Product GROUP BY Pname;
+----------------+-------------+
| PName          | Avg(UPrice) |
+----------------+-------------+
| Washing Powder |    120.0000 |
| Toothpaste     |     59.5000 |
| Soap           |     31.5000 |
| Shampoo        |    245.0000 |
+----------------+-------------+
4 rows in set (0.06 sec)
```

Remember, Avg(UPrice) will be in decimal values, since Avg() returns decmial values

i) SELECT DISTINCT Manufacturer FROM Product;

```
mysql> SELECT DISTINCT Manufacturer FROM Product;
+--------------+
| Manufacturer |
+--------------+
| Surf         |
| Colgate      |
| Lux          |
| Pepsodent    |
| Dove         |
+--------------+
5 rows in set (0.00 sec)
```

j) SELECT COUNT(DISTINCT PName) FROM Product;

```
mysql> SELECT COUNT(DISTINCT PName) FROM Product;
+-----------------------+
| COUNT(DISTINCT PName) |
+-----------------------+
|                     4 |
+-----------------------+
1 row in set (0.00 sec)
```

k) SELECT PName, MAX(UPrice), MIN(UPrice) FROM Product GROUP BY PName;

```
mysql> SELECT PName, MAX(UPrice), MIN(UPrice) FROM Product GROUP BY PName;**
+----------------+-------------+-------------+
| PName          | MAX(UPrice) | MIN(UPrice) |
+----------------+-------------+-------------+
| Washing Powder |         120 |         120 |
| Toothpaste     |          65 |          54 |
| Soap           |          38 |          25 |
| Shampoo        |         245 |         245 |
+----------------+-------------+-------------+
4 rows in set (0.06 sec)
```

8. Using the CARSHOWROOM database given in the chapter, write the SQL queries for the following:

The table `INVENTORY` is :

CarId	CarName	Price	Model	YearManufacture	Fueltype
D001	Drize	582613.00	LXI	2017	Petrol
D002	Drize	673112.00	VXI	2018	Petrol
B001	Baleno	567031.00	Sigma1.2	2019	Petrol
B002	Baleno	647858.00	Delta1.2	2018	Petrol
E001	EECO	355205.00	5 STR STD	2017	CNG
E002	EECO	654914.00	CARE	2018	CNG
S001	SWIFT	514000.00	LXI	2017	Petrol
S002	SWIFT	614000.00	VXI	2018	Petrol

a) Add a new column Discount in the INVENTORY table.

```
ALTER TABLE INVENTORY
ADD Discount FLOAT NOT NULL;
```

b) Set appropriate discount values for all cars keeping in mind the following:
(i) No discount is available on the LXI model.
(ii) VXI model gives a 10 per cent discount.
(iii) A 12 per cent discount is given on cars other than LXI model and VXI model.

```
UPDATE INVENTORY
SET Discount = 0.1 * Price WHERE Model = "VXI";
```

```
UPDATE INVENTORY
SET Discount = 0.12 * Price WHERE Model NOT IN ("LXI","VXI");
```

We don't need to set discount 0 for LXI model since we defined the Discount attritbute with the `NOT NULL` constraint

c) Display the name of the costliest car with fuel type “Petrol”.

```
SELECT MAX(Price)
FROM INVENTORY
WHERE Fueltype = "Petrol";
```

d) Calculate the average discount and total discount available on Baleno cars.

```
SELECT AVG(Discount), SUM(Discount)
FROM INVENTORY
WHERE CarName = "Baleno";
```

e) List the total number of cars having no discount.

```sql
SELECT COUNT(*)
FROM INVENTORY
WHERE Discount = 0;
```

Sample Paper Questions

The sample paper has 6 questions of SQL, worth a total of 16 marks.

Out of these 16 marks, 5 are theory based, 2 are output finding and 9 are for command writing.

Q3. Differentiate between char(n) and varchar(n) data types with respect to databases. (2)

char(n)	varchar(n)
1. This datatype stores a string of fixed length n (between 1 and 255)	1. This datatype stores a string of length up to length n
2. If the length of the string is less than n, the rest of the characters are stored with blank spaces	2. If the length of the string is less than n, no blank spaces are used
3. There is wastage of space	3. There is no wastage of space

You can write **any two** points of difference. There is **1 mark for each correct difference**

Q5. Write the output of the queries (a) to (d) based on the table Furniture given below: (2)
Table : FURNITURE

FID	NAME	DATEOFPURCHASE	COST	DISCOUNT
B001	Double Bed	03-Jan-2018	45000	10
T010	Dining Table	10-Mar-2020	51000	5
B004	Single Bed	19-Jul-2021	22000	0
C003	Long Back Chair	30-Dec-2016	12000	3
T006	Console Table	17-Nov-2019	15000	12
B006	Bunk Bed	01-Jan-2021	28000	14

(a) SELECT SUM(DISCOUNT) FROM FURNITURE WHERE COST>15000;

The `SUM(DISCOUNT)` will find the sum of the values in the `DISCOUNT` column.

But there is a condition attached to it, which is `COST>15000`. So it will return the sum of the discounts where the value of `COST` is more than 15000, ie.

The final value is $10 + 5 + 0 + 14 = 15 + 14 = 29$ (the values 3 and 12 were ignored since their COST is less than 15000)

Hence the final answer is `29`

(b) SELECT MAX(DATEOFPURCHASE) FROM FURNITURE;

The `MAX(DATE OF PURCHASE)` will return the date in `DATEOFPURCHASE` which is the maximum, i.e. which is the latest date out of them.

Evidently the latest date of purchase is `19-Jul-2021` so that is what will be printed.

(c) SELECT * FROM FURNITURE WHERE DISCOUNT>5 AND FID LIKE "T%";

The query is using `SELECT *` which means all the columns will be printed.

The values printed will be the ones which follow the 2 conditions given: `DISCOUNT>5` and `FID LIKE "T%"` .

The second condition is a substring pattern matching where we want the string `FID` to begin with the letter T

So tallying with the original table, this will be the output:

FID	NAME	DATEOFPURCHASE	COST	DISCOUNT
T006	Console Table	17-Nov-2019	15000	12

The value with FID `T010` was neglected since its `DISCOUNT` value is 5, and we want `DISCOUNT>5`

(d) SELECT DATEOFPURCHASE FROM FURNITURE WHERE NAME IN ("Dining Table", "Console Table");

According to the query, we want to print the `DATEOFPURCHASE` of those records where the `NAME` belongs to the list `("Dining Table", "Console Table")`

Tallying with the table, we get

DATEOFPURCHASE
10-Mar-2020
17-Nov-2019

You will receive **0.5 marks for each correct output**, so make sure to not make mistakes here, since there will not be any partial marking!

Q6.
(i) Which command is used to view the list of tables in a database? (1)

The command to view the list of tables in a database is:`SHOW TABLES;`

(ii) Give one point of difference between an equi-join and a natural join. (1)

Natural Join	Equi Join
This joins the two tables by using identical columns (same name and datatype) in both tables	This joins the two tables by comparing for equality
There is no duplicating of columns	There is duplication of the common column

You will receive **1 mark** for any one correct difference

Q9.
(i) A table, `ITEM` has been created in a database with the following fields:
ITEMCODE, ITEMNAME, QTY, PRICE

Give the SQL command to add a new field, DISCOUNT (of type Integer) to the ITEM table. (1)
The correct command to add a new field in a relation in SQL is:

```sql
ALTER TABLE ITEM
ADD DISCOUNT INT;
```

(ii) Categorize following commands into DDL and DML commands?

INSERT INTO, DROP TABLE, ALTER TABLE, UPDATE...SET (2)

DDL: DROP, ALTER
DML: INSERT INTO, UPDATE...SET

Explanation: `DROP` and `ALTER` are in DDL since they change the schema of the relations.
`INSERT` and `UPDATE` are in DML since they change the content stored in tables.

You will get **0.5 marks** for identifying each command correctly

Q10. Charu has to create a database named MYEARTH in MYSQL. She now needs to create a table named CITY in the database to store the records of various cities across the globe. The table CITY has the following structure: (3)

Table: CITY

FIELD NAME	DATA TYPE	REMARKS
CITYCODE	CHAR(5)	Primary Key
CITYNAME	CHAR(30)	
SIZE	INTEGER	
AVGTEMP	INTEGER	
POLLUTIONRATE	INTEGER	
POPULATION	INTEGER	

Help her to complete the task by suggesting appropriate SQL commands.

Ans:

The first thing she needs to do is create the database, then create the table with the specified schema. These are the correct commands to do so

```
CREATE DATABASE MYEARTH;            #1 mark for create database command
USE MYEARTH;

CREATE TABLE CITY(                  #2 marks for create table commmands
    CITYCODE CHAR(5) PRIMARY KEY,
    CITYNAME CHAR(30),
    SIZE INT,
    AVGTEMP INT,
    POLLUTIONRATE INT,
    POPULATION INT
);
```

Q11. Write queries (a) to (d) based on the tables EMPLOYEE and DEPARTMENT given below:(4)

Table: EMPLOYEE

EMPID	NAME	DOB	DEPTID	DESIG	SALARY
120	Alisha	23-Jan-1978	D001	Manager	75000
123	Nitin	10-Oct-1977	D002	AO	59000
129	Navjot	12-Jul-1971	D003	Supervisor	40000
130	Jimmy	30-Dec-1980	D004	Sales Rep	
131	Faiz	06-Apr-1984	D001	Dep Manager	65000

Table: DEPARTMENT

DEPID	DEPTNAME	FLOORNO
D001	Personal	4
D002	Admin	10
D003	Production	1
D004	Sales	3

(a) To display the average salary of all employees, department wise.

```
SELECT AVG(SALARY)
FROM EMPLOYEE
GROUP BY DEPTID;
```

(b) To display name and respective department name of each employee whose salary is more than 50000.

```sql
SELECT NAME,DEPTNAME
FROM EMPLOYEE, DEPARTMENT
WHERE EMPLOYEE.DEPTID=DEPARTMENT.DEPTID AND SALARY>5000; # Joining tables by foreign key
```

(c) To display the names of employees whose salary is not known, in alphabetical order.

```sql
SELECT NAME
FROM EMPLOYEE
WHERE SALARY IS NULL
ORDER BY NAME;
```

(d) To display DEPTID from the table EMPLOYEE without repetition.

```sql
SELECT DISTINCT DEPTID
FROM EMPLOYEE;
```

You will get **1 mark** for each correct query

Important note: Notice how the queries are written in multiple lines, for each of the types of queries (select,from, where, ...)

Always write queries in such a manner to avoid confusion for the examiner who will grade you.

Analysis

The SQL questions tend to revolve around writing and executing commands in various situations. One must be well versed with all the commands as described in detail in this book. Furthermore, the theory topics must be well known, especially the differences between data types, constraints, commands, etc.

PYQs

1. Anmol maintains that database of Medicines for his pharmacy using SQL to store the data. The structure of the table PHARMA for the purpose is as follows : [CBSE Computer Science Compartment 2021]

- **Name of the table - PHARMA**
- **The attributes of PHARMA are as follows :**
  MID - numeric
  MNAME - character of size 20
  PRICE - numeric
  UNITS - numeric
  EXPIRY - date

Table : `PHARMA`

MID	MNAME	PRICE	UNITS	EXPIRY
M1	PARACETAMOL	12	120	2022-12-25
M2	CETRIZINE	6	125	2022-10-12
M3	METAFORMIN	14	150	2022-05-23
M4	VITAMIN B-6	12	120	2022-07-01
M5	VITAMIN D3	25	150	2022-06-30
M6	TELMISARTAN	22	115	2022-02-25

(a) Write the degree and cardinality of the table `PHARMA`. (1)
Degree = 5 (Number of columns/attributes)
Cardinality = 6 (Number of rows/tuples)

(b) Identify the attribute best suitable to be declared as a primary key. (1)
`MID` is best suited as the primary key, since every value in it uniquely identifies every row

(c) Anmol has received a new medicine to be added into his stock, but for which he does not know the number of UNITS. So he decides to add the medicine without its value for UNITS. The rest of the values are as follows: (1)

MID	MNAME	PRICE	EXPIRY
M7	SUCRALFATE	17	2022-03-20

Write the SQL command which Anmol should execute to perform the required task. (1)

```
INSERT INTO PHARMA
VALUES ("M7","SUCRALFATE", 17, NULL ,"2022-03-20"); # NULL for UNITS
```

(d) Anmol wants to change the name of the attribute UNITS to QUANTITY in the table PHARMA. Which of the following commands will he use for the purpose ? (1)
(i) UPDATE
(ii) DROP TABLE
(iii) CREATE TABLE
(iv) ALTER TABLE

iv) ALTER TABLE

Since it is a DDL command, which changes the schema of the table

(e) Now Anmol wants to increase the PRICE of all medicines by 5. Which of the following commands will he use for the purpose? (1)
(i) UPDATE SET
(ii) INCREASE BY
(iii) ALTER TABLE
(iv) INSERT INTO

i) UPDATE SET

Since it is a DML command, which changes the data stored in the table.

2. Write the names of any two constraints and their respective uses in SQL. (2) [CBSE Computer Science Compartment 2021]
Constraints in SQL (write any 2):

- `NOT NULL`: Ensures the field has no value which is `NULL`
- `UNIQUE`: Ensures the values in the column are distinct
- `DEFAULT`: Gives a default value to the specific column
- `PRIMARY KEY`: The column becomes the primary key of the table
- `FORIEGN KEY`: The column becomes the foreign key of the table which references another table's primary key

3. Write the outputs of the SQL queries (i) to (iii) based on the relations `CUSTOMER` and `TRANSACTION` given below :
[CBSE Computer Science Compartment 2021]

Table: `CUSTOMER`

ACNO	NAME	SEX	BALANCE
C1	RISHABH	M	15000
C2	AAKASH	M	12500
C3	INDIRA	F	9750
C4	TUSHAR	M	14600
C5	ANKITA	F	22000

Table: `TRANSACTION`

ACNO	TDATE	AMOUNT	TYPE
C1	2020-07-21	1000	DEBIT
C5	2019-12-31	1500	CREDIT
C3	2020-01-01	2000	CREDIT

(i) SELECT MAX(BALANCE), MIN(BALANCE) FROM CUSTOMER
WHERE SEX = 'M';

MAX(BALANCE)	MIN(BALANCE)
15000	12500

(ii) SELECT SUM(AMOUNT), TYPE FROM TRANSACTION
GROUP BY TYPE;

SUM(AMOUNT)	TYPE
1000	DEBIT
3500	CREDIT

iii) SELECT NAME, TDATE, AMOUNT
FROM CUSTOMER C, TRANSACTION T
WHERE C.ACNO = T.ACNO AND TYPE = 'CREDIT';

NAME	TDATE	AMOUNT
INDIRA	2020-01-01	2000
ANKITA	2019-12-31	1500

4. Write SQL statements for the following queries (i) to (v) based on the relations CUSTOMER and TRANSACTION given below(same tables as Q3) : [CBSE Computer Science Compartment 2021]

Table: `CUSTOMER`

ACNO	NAME	SEX	BALANCE
C1	RISHABH	M	15000
C2	AAKASH	M	12500
C3	INDIRA	F	9750
C4	TUSHAR	M	14600
C5	ANKITA	F	22000

Table: `TRANSACTION`

ACNO	TDATE	AMOUNT	TYPE
C1	2020-07-21	1000	DEBIT
C5	2019-12-31	1500	CREDIT
C3	2020-01-01	2000	CREDIT

(a) To display all information about the CUSTOMER whose NAME starts with 'A'.

```sql
SELECT *
FROM CUSTOMER
WHERE NAME LIKE "A%";
```

(b) To display the `NAME` and `BALANCE` of Female CUSTOMERs (SEX as 'F') whose TRANSACTION Date (TDATE) is in the year 2019.

```
SELECT NAME,BALANCE
FROM CUSTOMER,TRANSACTION
WHERE CUSTOMER.ACNO=TRANSACTION.ACNO
AND SEX='F' AND YEAR(TDATE)='2019';
```

(c) To display the total number of CUSTOMERs for each SEX.

```
SELECT COUNT(*)
FROM CUSTOMER
GROUP BY SEX;
```

(d) To display the CUSTOMER NAME and BALANCE in ascending order of SEX.

```
SELECT NAME, BALANCE
FROM CUSTOMER
ORDER BY SEX ASC;
```

(e) To display CUSTOMER NAME and their respective INTEREST for all CUSTOMERs where INTEREST is calculated as 8% of BALANCE.

```
SELECT NAME, (BALANCE * 0.08) AS "INTERST"
FROM CUSTOMER;
```

5. Write SQL queries for (i) to (iv) and write outputs for SQL queries (v) to (viii), which are based on the table given below :
[CBSE 2019 Board Exam]

Table: `TRAINS`

TNO	TNAME	START	END
11096	Ahimsa Express	Pune Junction	Ahemdabad Junction
12015	Ajmer Shatabdi	New Delhi	Ajmer Junction
1651	Pune Hbj Special	Pune junction	Habibganj
13005	Amritsar Mail	Howrah Junction	Amritsar Junction
12002	Bhopal Shatabdi	New Delhi	Habibganj
12417	Prayag Raj Express	Allahbad Junction	New Delhi
14673	Shaheed Express	Jaynagar	Amritsar Junction
12314	Sealdah Rajdhani	New Delhi	Sealdah
12498	Shane Punjab	Amritsar Junction	New Delhi
12451	Shhram Shaki Express	Kanpur Central	New Delhi
12030	Swarna Shatabdi	Amritsar Junction	New Delhi

Table: `PASSENGERS`

PNR	TNO	PNAME	SEX	AGE	TRAVELDATE
p001	13005	R N AGARWAL	MALE	45	2018-12-25
p002	12015	P TIWARY	MALE	28	2018-11-10
p003	12015	S TIWARY	FEMALE	22	2018-11-10
p004	12030	S K SAXENA	MALE	42	2018-10-12
p005	12030	S SAXENA	FEMALE	35	2018-10-12
p006	12030	P SAXENA	FEMALE	12	2018-10-12
p007	13005	N S SINGH	MALE	52	2018-05-09
p008	12030	J K SHARMA	MALE	65	2018-05-09
p009	12030	R SHARMA	FEMALE	58	2018-05-09

(i) To display details of all Trains which Start from New Delhi.

```sql
SELECT *
FROM TRAINS
WHERE START = "NEW DELHI";
```

(ii) To display the PNR, PNAME, SEX and AGE of all Passengers whose AGE is below 50.

```sql
SELECT PNR, PNAME, SEX, AGE
FROM PASSENGER
WHERE AGE < 50;
```

(iii) To display total number of MALE and FEMALE Passengers.

```sql
SELECT SEX,COUNT(*)
FROM PASSENGER
GROUP BY SEX;
```

(iv) To display details of all Passengers travelling in Trains whose TNO is 12015.

```sql
SELECT *
FROM PASSENGERS
WHERE TNO =12015;
```

(v) SELECT MAX (TRAVELDATE), MIN(TRAVELDATE) FROM PASSENGERS WHERE SEX = 'FEMALE';

MAX(TRAVELDATE)	MIN(TRAVELDATE)
2018-11-10	2018-05-09

(vi) SELECT END, COUNT(*) FROM TRAINS GROUP BY END HAVING COUNT(*)>1;

END	COUNT(*)
Habibganj	2
Amritsar Junction	2
New Delhi	3

(vii) SELECT DISTINCT TRAVELDATE FROM PASSENGERS;

TRAVELDATE
2018-12-25
2018-11-10
2018-10-12
2018-05-09

(viii) SELECT TNAME, PNAME FROM TRAINS T, PASSENGERS P WHERE T.TNO = P.TNO AND AGE BETWEEN 50 AND 60;

TNAME	PNAME
Amritsar Mail	N S SINGH
Swarna Shatabdi	J K SHARMA
R SHARMA	R SHARMA

6. (a) Observe the following tables VIDEO and MEMBER carefully and write the name of the RDBMS operation out of (i) SELECTION (ii) PROJECTION (iii) UNION (iv) CARTESIAN PRODUCT, which has been used to produce the output as shown below. Also, find the Degree and Cardinality of the final result [CBSE 2018 Board Exam]

Table: `VIDEO`

VNO	VNAME	TYPE
F101	The Last Battle	Fiction
C101	Angels and Devils	Comedy
A102	Daredevils	Adventure

Table: `MEMBER`

MNO	MNAME
M101	Namish Gupta
M102	Sana Sheikh
M103	Lara James

Table: `FINAL RESULT`

VNO	VNAME	TYPE	MNO	MNA
F101	The Last Battle	Fiction	M101	Namish Gupta
F101	The Last Battle	Fiction	M102	Sana Sheikh
F101	The Last Battle	Fiction	M103	Lara James
C101	Angels and Devils	Comedy	M101	Namish Gupta
C101	Angels and Devils	Comedy	M102	Sana Sheikh
C101	Angels and Devils	Comedy	M103	Lara James
A102	Daredevils	Adventure	M101	Namish Gupta
A102	Daredevils	Adventure	M102	Sana Sheikh
A102	Daredevils	Adventure	M103	Lara James

Answer:
This is a cartesian product, since every row of `VIDEO` has been mapped with `MEMBER` to produce the final result table
Of `FINAL RESULT`
degree= 5
cardinality= 9

(b) Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii), which are based on the tables.

Table: `ACCOUNT`

ANO	ANAME	ADDRESS
101	Nirja Singh	Bangalore
102	Rohan Gupta	Chennai
103	Alir Reza	Hyderabad
104	Rishabh Gupta	Chennai
105	Simran Kaur	Chandigarh

Table: `TRANSACT`

TRNO	ANO	AMOUNT	TYPE	DOT
T001	101	2500	Withdraw	2017-12-21
T002	103	3000	Deposit	2017-06-01
T003	102	2000	Withdraw	2017-05-12
T004	103	1000	Deposit	2017-10-22
T005	101	12000	Deposit	2017-11-06

(i) To display details of all transactions of TYPE Deposit from Table TRANSACT

```
SELECT *
FROM TRANSACT
WHERE TYPE="Deposit";
```

(ii) To display the ANO and AMOUNT of all Deposits and Withdrawals done in the month of October 2017 from table TRANSACT.

```
SELECT ANO,AMOUNT
FROM TRANSACT
WHERE MONTH(DOT)="10" and YEAR(DOT)="2017";
```

(iii) To display the last date of transaction (DOT) from the table TRANSACT for the Accounts having ANO as 103

```
SELECT DOT
FROM TRANSACT
WHERE ANO=103;
```

(iv) To display all ANO, ANAME and DOT of those persons from tables ACCOUNT and TRANSACT who have done transactions less than or equal to 3000

```
SELECT ANO,ANAME,DOT
FROM ACCOUNT,TRANSACT
WHERE ACCOUNT.ANO=TRANSACT.ANO
    AND AMOUNT<=3000;
```

(v) SELECT ANO, ANAME FROM ACCOUNT WHERE ADDRESS NOT IN ('CHENNAI', 'BANGALORE');

ANO	ANAME
103	Ali Reza
105	Chandigarh

(vi) SELECT DISTINCT ANO FROM TRANSACT;

ANO
101
103
102

(vii) SELECT ANO, COUNT(*), MIN(AMOUNT) FROM TRANSACT GROUP BY ANO HAVING COUNT(*)> 1;

ANO	COUNT(*)	MIN(AMOUNT)
101	2	2500
103	2	1000

(viii) SELECT COUNT(*), SUM(AMOUNT) FROM TRANSACT WHERE DOT <= '2017-06-01';

COUNT(*)	SUM(AMOUNT)
2	5000

7. **Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii), which are based on the tables : [CBSE 2019 Board Exa]**

   TABLE : SALESPERSON

CODE	NAME	SALARY	ITCODE
1001	TANDEEP JHA	60000	I2
1002	YOGRAJ SINHA	70000	I5
1003	TENZIN JACK	45000	I2
1005	ANOKHI RAJ	50000	I7
1004	TARANA SEN	55000	I7

TABLE: ITEM

ITCODE	ITEMTYPE	TURNOVER
I5	STATIONARY	3400000
I7	HOSIERY	6500000
I2	BAKERY	10090000

(i) To display the CODE and NAME of all SALESPERSON having “I7” Item Type Code from the table SALESPERSON.

```
SELECT CODE, NAME
FROM SALESPERSON
WHERE ITCODE = "I7";
```

(ii) To display all details from table SALESPERSON in descending order of SALARY.

```
SELECT *
FROM SALESPERSON
ORDER BY SALARY DESC;
```

(iii) To display the number of SALESPERSON dealing in each TYPE of ITEM. (Use ITCODE for the same)

```
SELECT ITCODE, COUNT(*)
FROM SALESPERSON
GROUP BY ITCODE;
```

(iv) To display NAME of all the salespersons from the SALESPERSON table along with their corresponding ITEMTYPE from the ITEM table.

```
SELECT NAME, ITEMTYPE
FROM SALESPERSON, ITEM
WHERE SALESPERSON.ITCODE = ITEM.ITCODE;
```

(v) SELECT MAX(SALARY) FROM SALESPERSON;

```
70000
```

(vi) SELECT DISTINCT ITCODE FROM SALESPERSON;

```
I2
I5
I7
```

(vii) SELECT CODE,NAME, I.ITCODE
FROM SALESPERSON S, ITEM I
WHERE S.ITCODE=I.ITCODE AND TURNOVER>=7000000;

```
1001 TANDEEP JHA I2
1003 TENZIN JACK I2
```

(viii) SELECT SUM(SALARY) FROM SALESPERSON WHERE ITCODE=”I2”;

```
105000
```

Python-SQL Connectivity

This part of the database management unit is very similar to the SQL portion. The main difference is that we are using Python for our queries.

Installing Modules

You will need to install the following package to use SQL with Python: https://pypi.org/project/mysql-connector-python/

If you are familiar with installing packages in Python via PIP, this should be easy.

Steps

1. Open your terminal
2. Type `pip install mysql-connector-python`
3. Wait for the download to complete

Note: there is a chance this method may not work for everyone. You are advised to do your own research in the event this does not work for you.

Basic Setup

Once you have the package installed, you can test the connection:

```python
import mysql.connector as mysql

db = mysql.connect(
    host = "",
    user = "",
    passwd = "",
)
```

What values should you use?

You can try to use the following values:

1. `host = "localhost"`
2. `user = "root"`
3. `passwd = "<your MySQL password>"`

If that does not work, here are the steps to follow:

1. In your MySQL Command Line Client, type `status`

   You should get a bunch of data on your screen.

2. Look at the first few rows which may look like this:

   ```
   Connection id:          8
   Current database:       test
   Current user:           root@localhost
   ```

3. To find the user, look at the name before the `@` in the `Current user:` row.
4. To find the host, look at the name after the `@` in the same row.

Try to run the code now:

```python
import mysql.connector as mysql

db = mysql.connect(
    host = "localhost",
    user = "root",
    passwd = "password" # substitute with your password
)

print(db.is_connected())
```

If the output is `True`, you have successfully connected MySQL to Python!

For this book, we will be using this connection throughout.

Performing Queries

To perform queries, we need to create a `cursor` object:

```python
import mysql.connector as mysql

db = mysql.connect(
    host = "localhost",
    user = "root",
    passwd = "password"
)

cursor = db.cursor()
```

Here, the `cursor` variable is our cursor object.

Let's start off by executing a simple query:

```python
import mysql.connector as mysql

db = mysql.connect(
    host = "localhost",
    user = "root",
    passwd = "password"
)

cursor = db.cursor()
cursor.execute("USE DATABASE qprogramming;") # execute is used for command execution

db.close() # add .close() to disconnect from MySQL
```

This will not give us any output; we are just trying to test the program.

`fetchone()`

The `fetchone()` method returns a **tuple** of the first record in the table after making a query.

For example:

```python
import mysql.connector as mysql

db = mysql.connect(
    host = "localhost",
    user = "root",
    passwd = "password",
    database = "qprogramming" # we can define the database we want to use as well in the connection
)

cursor = db.cursor()
cursor.execute("SELECT * FROM qprog_emps")

print(cursor.fetchone())
db.close()
```

Output

```
(1, 'Arjun Sodhi', 'arjun.sodhi@qprogramming.net')
```

The data used is from here as it was originally defined (ignore all the modifications we made afterwards).

If we execute `print(cursor.fetchone())` twice it will return the second record for the second execution:

```python
import mysql.connector as mysql

db = mysql.connect(
    host = "localhost",
    user = "root",
    passwd = "password",
    database = "qprogramming"
)

cursor = db.cursor()
cursor.execute("SELECT * FROM qprog_emps")

print(cursor.fetchone())
print(cursor.fetchone())

db.close()
```

Output

```
(1, 'Arjun Sodhi', 'arjun.sodhi@qprogramming.net')
(2, 'Sammarth Kumar', 'sam@qprogramming.net')
```

`fetchall()`

`fetchall()` returns all the records from the query as a **list of tuples**:

```python
import mysql.connector as mysql

db = mysql.connect(
    host = "localhost",
    user = "root",
    passwd = "password",
    database = "qprogramming"
)

cursor = db.cursor()
cursor.execute("SELECT * FROM qprog_emps")
cursor.fetchall()

db.close()
```

Output

```
[(1, 'Arjun Sodhi', 'arjun.sodhi@qprogramming.net'), (2, 'Sammarth Kumar', 'sam@qprogramming.net')]
```

`rowcount`

The `rowcount` attribute of the `cursor` object tells us how many rows were retrieved by our Python program.

For example:

```python
import mysql.connector as mysql

db = mysql.connect(
    host = "localhost",
    user = "root",
    passwd = "password",
    database = "qprogramming"
)

cursor = db.cursor()
cursor.execute("SELECT * FROM qprog_emps")

cursor.fetchone()
print(cursor.rowcount)

db.close()
```

Output

```
1
```

Since `fetchone()` retrieves only one value from the cursor, the output is one.

However, if we use `fetchall()`:

```python
import mysql.connector as mysql

db = mysql.connect(
    host = "localhost",
    user = "root",
    passwd = "password",
    database = "qprogramming"
)

cursor = db.cursor()
cursor.execute("SELECT * FROM qprog_emps")

cursor.fetchall()
print(cursor.rowcount)

db.close()
```

Output

```
2
```

`INSERT`

Let's now try to add data through Python.

We will simply use the `INSERT` syntax in SQL as follows:

```python
import mysql.connector as mysql

db = mysql.connect(
    host = "localhost",
    user = "root",
    passwd = "password",
    database = "qprogarmming"
)

cursor = db.cursor()
cursor.execute("INSERT INTO qprog_emps VALUE (3, 'Balpreet Juneja', 'balpreet@qprogramming.net');")

db.commit() # important

# Testing to see if the INSERT worked
cursor.execute("SELECT * FROM qprog_emps;")
print(cursor.fetchall())
```

```
18
19  db.close()
```

Output

```
[(1, 'Arjun Sodhi', 'arjun.sodhi@qprogramming.net'), (2, 'Sammarth Kumar',
'sam@qprogramming.net'), (3, 'Balpreet Juneja', 'balpreet@qprogramming.net')]
```

As you can see, we have inserted a value into the `qprog_emps` table.

Note on line 13 we have used the statement `db.commit()` this was done to ensure that the changes made in Python are also made in MySQL.

`UPDATE`

We are now going to perform an update query. The process is similar to the insert query.

In this example, we assume that Arjun Sodhi has changed his email to `arjun@qprogramming.net`, so we will need to update our table accordingly.

```python
import mysql.connector as mysql

db = mysql.connect(
    host = "localhost",
    user = "root",
    passwd = "password",
    database = "qprogramming"
)

cursor = db.cursor()
cursor.execute("UPDATE qprog_emps SET Email='arjun@qprogramming.net' WHERE Srno=1;")

db.commit() # important

# Testing to see if the UPDATE worked
cursor.execute("SELECT * FROM qprog_emps;")
print(cursor.fetchall())

db.close()
```

Output

```
[(1, 'Arjun Sodhi', 'arjun@qprogramming.net'), (2, 'Sammarth Kumar',
'sam@qprogramming.net'), (3, 'Balpreet Juneja', 'balpreet@qprogramming.net')]
```

As you can see, the email has successfully been changed!

Note: We need to use `db.commit()` only for `INSERT` and `UPDATE` operations (and also for the other DML operations).

Now, you may be asking what's the point of using Python when we can perform these queries in MySQL itself?

Well, Python allows you to do many more things that MySQL cannot. For example, you can transfer content from a CSV to MySQL using an automated Python program.

You can also customize the input for automatic programs in Python using string formatting.

Sample Paper Questions

The sample paper has only one question of Python-SQL connectivity, which is of 2 marks. This indicates that this topic is not very important and has a limited set of questions that can be asked. Most likely you will be asked about the cursor functions so revise them well and know the return types of the functions so you can write output for code.

4. A resultset is extracted from the database using the cursor object (that has been already created) by giving the following statement.

Mydata=cursor.fetchone()

(a) How many records will be returned by fetchone() method?
Only **one** record is returned by the `fetchone()` method

(b) What will be the datatype of Mydata object after the given command is executed?
The `fetchone()` method returns a **tuple**, so that will be the datatype of `Mydata`

You will get 1 mark each for both parts

PYQs

For the following SQL Table named `PASSENGERS` in a database `TRAVEL`:[CBSE Compartment Exam 2021]

TNO	NAME	START	END
T1	RAVI KUMAR	DELHI	MUMBAI
T2	NISHANT JAIN	DELHI	KOLKATA
T3	DEEPAK PRAKASH	MUMBAI	PUNE

A cursor named `Cur` is created in Python for a connection of a host which contains the database `TRAVEL`. Write the output for the execution of the following Python statements for the above SQL Table `PASSENGERS`:

```python
Cur.execute('USE TRAVEL')
Cur.execute('SELECT * FROM PASSENGERS')
Recs = Cur.fetchall()
for R in Recs:
    print(R[1])
```

Solution:

Remember the `fetchall()` method returns a list of all the tuples in the table

so if we loop through `Recs` (which is a list) and take the 2nd element of all of them(first index), we will get:

```
RAVI KUMAR
NISHANT JAIN
DEEPAK PRAKASH
```