CUET-UG Information Practices Sample Paper-19

Duration: 1 Hour **Maximum Marks: 250**

Instructions

- This paper contains a total of 50 Multiple Choice Questions.
- Each correct answer carries **+5 marks**.
- Each incorrect answer carries **-1 mark**.
- No negative marking for unattempted questions.

Q1. What will be the output of `SELECT SQRT(196);` ?

(A) 12

(B) 13

(C) 14

(D) 16

Q2. Which SQL function is used to concatenate two or more strings?

(A) MERGE()

(B) COMBINE()

(C) CONCAT()

(D) JOIN()

Q3. What will be the output of `SELECT RIGHT('COMPUTER',4);` ?

(A) COMP

(B) UTER

(C) PUTE

(D) TER

Q4. Which SQL function returns the current time only?

(A) CURTIME()

(B) NOW()

(C) CURRENT()

(D) TIME()

Q5. What will be the output of `SELECT ROUND(456.789,1);` ?

(A) 456.7

(B) 456.8

(C) 457

(D) 456

Q6. Which key can uniquely identify a tuple and may be selected as the primary key?

(A) Foreign Key

(B) Candidate Key

(C) Composite Key

(D) Secondary Key

Q7. Which relational algebra operation selects rows satisfying a condition?

(A) Projection

(B) Join

(C) Selection

(D) Union

Q8. Which normal form removes transitive dependency from a relation?

(A) 1NF

(B) 2NF

(C) 3NF

(D) 4NF

Q9. Which topology is most affected if the main communication cable fails?

(A) Star

(B) Bus

(C) Mesh

(D) Tree

Q10. Which networking device connects networks using different protocols?

(A) Repeater

(B) Hub

(C) Gateway

(D) Switch

Q11. Which layer of the TCP/IP model is responsible for logical addressing?

(A) Application Layer

(B) Transport Layer

(C) Internet Layer

(D) Network Access Layer

Q12. Which Pandas function is used to create an empty DataFrame?

(A) pd.DataFrame()

(B) pd.Empty()

(C) pd.Blank()

(D) pd.Table()

Q13. What will be returned by `df.axes` ?

(A) Number of rows

(B) Row and column labels

(C) Statistical summary

(D) Data values only

Q14. Which method is used to delete a column from a DataFrame permanently?

(A) remove()

(B) pop()

(C) erase()

(D) clear()

Q15. Which argument in `drop()` specifies deletion of rows?

(A) axis=0

(B) axis=1

(C) row=True

(D) delete=True

Q16. What is the output of `df.ndim` for a Series object?

(A) 0

(B) 1

(C) 2

(D) Depends on size

Q17. Which Pandas function is used to combine DataFrames vertically?

(A) concat()

(B) merge()

(C) join()

(D) relate()

Q18. Which method returns the first valid non-null entry of a DataFrame?

(A) first()

(B) first_valid_index()

(C) valid()

(D) start()

Q19. Which method is used to detect duplicate rows in a DataFrame?

(A) repeated()

(B) duplicated()

(C) clone()

(D) copied()

Q20. What will be the output type of `df.iloc[2:5]` ?

(A) Series

(B) Scalar value

(C) DataFrame

(D) Tuple

Q21. Which SQL clause is used to group records with identical values?

(A) ORDER BY

(B) GROUP BY

(C) HAVING

(D) DISTINCT

Q22. Which SQL query correctly counts total rows in table `student`?

(A) `SELECT TOTAL(*) FROM student;`

(B) `SELECT COUNT(*) FROM student;`

(C) `SELECT NUMBER(*) FROM student;`

(D) `SELECT SUM(*) FROM student;`

Q23. Which join returns all rows from the right table and matching rows from the left table?

(A) LEFT JOIN

(B) INNER JOIN

(C) RIGHT JOIN

(D) CROSS JOIN

Q24. Which SQL operator checks whether a value lies within a range?

(A) LIKE

(B) BETWEEN

(C) IN

(D) EXISTS

Q25. Which aggregate function returns the total sum of values in a numeric column?

(A) ADD()

(B) TOTAL()

(C) SUM()

(D) COUNT()

Q26. Which clause filters grouped records after aggregation?

(A) WHERE

(B) GROUP BY

(C) HAVING

(D) ORDER BY

Q27. Which query correctly displays the second highest salary from a table?

(A) `SELECT MAX(salary) FROM emp;`

(B) `SELECT salary FROM emp ORDER BY salary DESC LIMIT 1,1;`

(C) `SELECT SECOND(salary) FROM emp;`

(D) `SELECT salary FROM emp WHERE MAX(salary);`

Q28. Which SQL statement removes specific rows from a table?

(A) DROP

(B) REMOVE

(C) DELETE

(D) CLEAR

Q29. Which join returns every possible combination of rows from two tables?

(A) INNER JOIN

(B) LEFT JOIN

(C) CROSS JOIN

(D) NATURAL JOIN

Q30. Which SQL clause arranges records in sorted order?

(A) SORT BY

(B) GROUP BY

(C) ORDER BY

(D) ARRANGE BY

Q31. Which Matplotlib function is used to draw a line graph?

(A) plt.line()

(B) plt.plot()

(C) plt.graph()

(D) plt.draw()

Q32. Which parameter specifies marker style in `plt.plot()` ?

(A) color

(B) marker

(C) linestyle

(D) width

Q33. Which Matplotlib function clears the current figure?

(A) plt.clean()

(B) plt.remove()

(C) plt.clf()

(D) plt.reset()

Q34. Which chart is most suitable for representing trends over time?

(A) Pie Chart

(B) Histogram

(C) Line Graph

(D) Bar Graph

Q35. Which argument controls transparency in Matplotlib plots?

(A) shade

(B) alpha

(C) opacity

(D) fade

Q36. Which cyber threat replicates itself automatically across networks?

(A) Trojan Horse

(B) Worm

(C) Spyware

(D) Adware

Q37. Which malicious software disguises itself as a legitimate program?

(A) Worm

(B) Trojan Horse

(C) Antivirus

(D) Firewall

Q38. Which authentication method requires two different forms of verification?

(A) Simple Authentication

(B) Multi-factor Authentication

(C) Password Authentication

(D) Public Authentication

Q39. Which cybercrime involves unauthorized access to computer systems?

(A) Phishing

(B) Hacking

(C) Spamming

(D) Blogging

Q40. Which practice helps in reducing electronic waste?

(A) Burning old devices

(B) Throwing electronics in open landfills

(C) Recycling and refurbishing devices

(D) Mixing e-waste with household waste

Q41. Which type of software is distributed free of cost but with source code unavailable?

(A) Open Source Software

(B) Freeware

(C) Shareware

(D) Proprietary Sourceware

Q42. Which law deals specifically with crimes committed using computers and networks?

(A) Patent Law

(B) Copyright Law

(C) Cyber Law

(D) Trademark Law

Q43. Which networking protocol is used for transferring files between computers?

(A) HTTP

(B) FTP

(C) SMTP

(D) POP3

Q44. Which device provides an interface between a computer and a network cable?

(A) Router

(B) Switch

(C) NIC

(D) Repeater

Q45. What is the full form of DNS?

(A) Domain Name System

(B) Data Naming Service

(C) Digital Network Server

(D) Domain Network Security

Q46. Which Pandas method is used to calculate the sum of values in a Series?

(A) total()

(B) add()

(C) sum()

(D) calculate()

Q47. Which Pandas method returns the minimum value from a Series?

(A) low()

(B) minimum()

(C) min()

(D) smallest()

Q48. Which SQL operator checks for NULL values?

(A) = NULL

(B) IS NULL

(C) NULL =

(D) HAS NULL

Q49. Which wildcard symbol represents zero or more characters in SQL?

(A) _

(B) *

(C) %

(D) #

Q50. Which SQL command completely removes a table along with its structure?

(A) DELETE

(B) TRUNCATE

(C) REMOVE

(D) DROP

Detailed Solutions

Q1.

Solution

Concept: 'SQRT()' is an SQL mathematical function used to calculate the square root of a number. The square root of a number is a value which, when multiplied by itself, gives the original number. Such mathematical functions are commonly used in calculations, statistics, and data analysis queries.

Solution: The query: 'SELECT SQRT(196);'
asks SQL to calculate the square root of '196'.

We know:

$$14 \times 14 = 196$$

Therefore:

$$\sqrt{196} = 14$$

So, the output of the query will be '14'.

The 'SQRT()' function only works properly with non-negative numbers. It is useful in scientific calculations, distance formulas, and statistical operations.

Final Answer : "14"

Answer: (C)

Q2.

Solution

Concept: String functions in SQL are used to manipulate text values. Concatenation means joining two or more strings together to form a single string.

Solution: The ‘CONCAT()‘ function is used in SQL to combine multiple strings into one string. It joins the strings in the same order in which they are written.

Example: ‘SELECT CONCAT(’Hello’, ’ ’, ’World’);‘

Output: ‘Hello World‘

This function is very useful for combining first name and last name, creating full addresses, or formatting output data.

Other options are incorrect because:

- ‘MERGE()‘ is used for combining table records.
- ‘COMBINE()‘ is not a standard SQL function.
- ‘JOIN()‘ is used to combine rows from multiple tables.

Hence, ‘CONCAT()‘ is the correct function for string concatenation.

Final Answer : “CONCAT()”

Answer: (C)

Q3.

Solution

Concept: The 'RIGHT()' function in SQL extracts a specified number of characters from the right side of a string. It is commonly used in text processing and formatting operations.

Solution: The query: 'SELECT RIGHT('COMPUTER',4);'
extracts the last 4 characters from the string 'COMPUTER'.

The word is:

$$COMPUTER$$

Counting 4 characters from the right:

$$U \quad T \quad E \quad R$$

Therefore, the result will be:

$$UTER$$

The 'RIGHT()' function is useful for extracting file extensions, suffixes, codes, or ending characters from strings.

Final Answer : "UTER"

Answer: (B)

Q4.

Solution

Concept: SQL provides various date and time functions to retrieve current system date or time from the database server. These functions are widely used in transaction processing and record management.

Solution: The ‘CURTIME()‘ function returns only the current system time in the format:

$$HH : MM : SS$$

Example: ‘SELECT CURTIME();‘

Possible Output: ‘14:35:20‘

This function is useful when only the current time is required without the date portion.

Other options are incorrect because:

- ‘NOW()‘ returns both current date and time.
- ‘CURRENT()‘ is not a complete SQL time function.
- ‘TIME()‘ is mainly used to extract the time part from a datetime value.

Therefore, ‘CURTIME()‘ is the correct answer.

Final Answer : “CURTIME()”

Answer: (A)

Q5.

Solution

Concept: The ‘ROUND()‘ function in SQL is used to round a numeric value to a specified number of decimal places. It follows standard mathematical rounding rules.

Solution: The query: ‘SELECT ROUND(456.789,1);‘
rounds the number ‘456.789‘ to one decimal place.

Steps:

- Keep the first digit after decimal point: ‘7‘
- Check the next digit: ‘8‘
- Since ‘8 $\geq$ 5‘, $round$‘7‘$upward$

Therefore:

$$456.789 \rightarrow 456.8$$

Hence, the output will be ‘456.8‘.

The ‘ROUND()‘ function is commonly used in financial reports, averages, and statistical calculations where decimal precision is important.

Final Answer : “456.8”

Answer: (B)

Q6.

Solution

Concept: Keys are essential in relational databases because they help uniquely identify records and maintain relationships between tables. Different types of keys serve different purposes.

Solution: A Candidate Key is an attribute or combination of attributes that can uniquely identify each record in a table. Any candidate key can be selected as the primary key.

Example: In a student table, both ‘RollNo‘ and ‘EmailID‘ may uniquely identify students. Hence, both are candidate keys. Out of them, one can be chosen as the primary key.

Other keys:

- Foreign Key: Maintains relationship between tables
- Composite Key: Combination of two or more attributes
- Secondary Key: Used mainly for searching

Thus, a Candidate Key is the key that may be selected as the primary key.

Final Answer : “Candidate Key”

Answer: (B)

Q7.

Solution

Concept: Relational algebra is the procedural query language used in relational databases. It contains operations that manipulate relations (tables).

Solution: The Selection operation is used to select rows from a table based on a given condition. It is represented by the symbol σ.

Example:

$$\sigma_{\text{Salary}>50000}(Employees)$$

This operation returns only those rows where the salary is greater than ‘50000‘.

Selection works similar to the ‘WHERE‘ clause in SQL.

Other operations:

- Projection: Selects specific columns
- Join: Combines tables
- Union: Combines rows from relations

Therefore, the correct answer is ‘Selection‘.

Final Answer : “Selection”

Answer: (C)

Q8.

Solution

Concept: Normalization is a database design technique used to reduce redundancy and improve data consistency. Different normal forms remove different types of dependency problems.

Solution: Third Normal Form (3NF) removes transitive dependency from a table. A table is in 3NF if:

- It is already in 2NF
- Non-key attributes do not depend on other non-key attributes

Example: Suppose:

$$StudentID \rightarrow DepartmentID$$

and

$$DepartmentID \rightarrow DepartmentName$$

Here, ‘DepartmentName‘ depends indirectly on ‘StudentID‘, creating transitive dependency. This should be removed by creating separate tables.

3NF helps reduce redundancy and prevents update anomalies in databases.

Final Answer : “3NF”

Answer: (C)

Q9.

Solution

Concept: Network topology defines how devices are connected in a network. Different topologies provide different levels of reliability and fault tolerance.

Solution: In a Bus topology, all devices are connected to a single communication cable called the backbone. Data travels through this common cable.

If the main backbone cable fails, communication between all devices stops, causing the entire network to fail. Hence, bus topology is highly affected by cable failure.

Other topologies:

- Star Topology: Failure of one cable affects only one device
- Mesh Topology: Multiple paths provide high reliability
- Tree Topology: Structured hierarchical network

Therefore, the Bus topology is most affected when the main cable fails.

Final Answer : "Bus"

Answer: (B)

Q10.

Solution

Concept: Networking devices are used to establish communication between computers and networks. Each device performs a specific networking function.

Solution: A Gateway is a networking device used to connect networks that use different communication protocols. It performs protocol conversion so that different types of networks can communicate with each other.

For example, a gateway can connect a local network to the internet or connect two networks using different technologies.

Other devices:

- Repeater: Regenerates weak signals
- Hub: Sends data to all connected devices
- Switch: Transfers data within the same network efficiently

Hence, ‘Gateway‘ is the correct answer.

Final Answer : “Gateway”

Answer: (C)

Q11.

Solution

Concept: The TCP/IP model is used for communication over networks and the internet. Each layer performs a different networking function. Logical addressing is necessary for identifying devices uniquely.

Solution: The Internet Layer of the TCP/IP model is responsible for logical addressing using IP addresses. It also handles routing of packets between different networks.

IP addresses uniquely identify devices connected to a network and help data reach the correct destination.

Other layers:

- Application Layer: Provides user-level services
- Transport Layer: Ensures reliable communication
- Network Access Layer: Handles physical transmission of data

Therefore, the correct answer is 'Internet Layer'.

Final Answer : "Internet Layer"

Answer: (C)

Q12.

Solution

Concept: Pandas provides the `DataFrame` data structure for storing tabular data. An empty DataFrame can be created when no initial data is available.

Solution: The `pd.DataFrame()` constructor is used to create an empty DataFrame in Pandas. Example:

```
import pandas as pd

# Create an empty DataFrame
df_empty = pd.DataFrame()

print(df_empty)
```

Output:

```
Empty DataFrame
Columns: []
Index: []
```

We can also create an empty DataFrame with only column names:

```
df = pd.DataFrame(columns=['Name', 'Age'])

print(df)
```

Output:

```
Empty DataFrame
Columns: [Name, Age]
Index: []
```

Other Options:

- **pd.Empty():** Not a valid Pandas function.
- **pd.Blank():** Not a valid Pandas function.
- **pd.Table():** Not a standard Pandas constructor.

Therefore, `pd.DataFrame()` is the correct way to create an empty DataFrame.

Final Answer : "pd.DataFrame()"

Answer: (A)

Q13.

Solution

Concept: A Pandas DataFrame is a two-dimensional labeled data structure with columns of potentially different types. It has two primary axes: the rows and the columns. Each axis has labels associated with it, known as the index for rows and column names for columns.

Solution: The `df.axes` attribute of a Pandas DataFrame returns a list-like object containing both the row labels (index) and column labels (columns). It provides direct access to the DataFrame's row and column headers.

Example:

```
import pandas as pd
data = {'col1': [1, 2], 'col2': [3, 4]}
df = pd.DataFrame(data, index=['rowA', 'rowB'])
print("DataFrame:")
print(df)

# Output:
#       col1  col2
# rowA     1     3
# rowB     2     4

print("\nOutput of df.axes:")
print(df.axes)

# Output:
# [Index(['rowA', 'rowB'], dtype='object'),
#  Index(['col1', 'col2'], dtype='object')]
```

Let's review the other options:
(A) Number of rows: This information is typically obtained using 'len(df)' or 'df.shape[0]'. 'df.axes' provides the labels, not just the count.
(C) Statistical summary: A statistical summary (like mean, median, min, max, std dev) is obtained using the 'df.describe()' method.
(D) Data values only: The raw data values can be accessed using 'df.values' (which returns a NumPy array) or by iterating through the DataFrame. 'df.axes' explicitly provides the labels of the axes, not the data content.

Therefore, 'df.axes' returns both the row and column labels of the DataFrame.

Final Answer : "Row and column labels"

Answer: (B)

Q14.

Solution

Concept: Pandas provides methods to modify DataFrames, including deleting rows or columns. Some methods remove data permanently from the DataFrame.

Solution: The `pop()` method is used to permanently remove a column from a DataFrame and return it as a Series.
Example:

```
import pandas as pd
data = {'col1':[1,2,3],
        'col2':['A','B','C']}
df = pd.DataFrame(data)
removed = df.pop('col2')
print(df)
print(removed)
```

Output:

```
   col1
0     1
1     2
2     3

0    A
1    B
2    C
Name: col2, dtype: object
```

Thus, `pop()` removes the column permanently and also returns its values.

Other Options:

- **remove():** Not a standard Pandas method.
- **erase():** Not a valid Pandas method.
- **clear():** Removes all content, not a specific column.

Therefore, `pop()` is the correct method.

Final Answer : "pop()"

Answer: (B)

Q15.

Solution

Concept: The `drop()` method in Pandas is used to remove rows or columns from a DataFrame. The `axis` argument specifies whether rows or columns should be deleted.

Solution: In the `drop()` method:

- `axis=0` removes rows
- `axis=1` removes columns

Example:

```
import pandas as pd

data = {'col1':[1,2,3],
        'col2':['A','B','C']}

df = pd.DataFrame(data)

df2 = df.drop(1, axis=0)

print(df2)
```

Output:

```
   col1 col2
0     1    A
2     3    C
```

Here, row index 1 is removed.

Other Options:

- **axis=1:** Used for deleting columns.
- **row=True:** Not a valid argument.
- **delete=True:** Not a valid argument.

Therefore, `axis=0` is used to delete rows.

Final Answer : "axis=0"

Answer: (A)

Q16.

Solution

Concept: A Pandas `Series` is a one-dimensional labeled array. The `ndim` attribute returns the number of dimensions of the object.

Solution: Since a Series is always one-dimensional, its `ndim` value is always:

1

Example:

```
import pandas as pd

s = pd.Series([10,20,30])

print(s.ndim)
```

Output:

```
1
```

A DataFrame is two-dimensional, so its `ndim` value is 2.

Other Options:

- **0:** Represents a scalar value.
- **2:** Represents a DataFrame.
- **Depends on size:** Dimension does not depend on size.

Therefore, the output of `s.ndim` is always `1`.

Final Answer : "1"

Answer: (B)

Q17.

Solution

Concept: Pandas provides different methods to combine DataFrames. Vertical combination means stacking rows one below another.

Solution: The `concat()` function is used to combine DataFrames vertically.
Example:

```
import pandas as pd

df1 = pd.DataFrame({'A':[1,2]})
df2 = pd.DataFrame({'A':[3,4]})

result = pd.concat([df1, df2])

print(result)
```

Output:

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

Here, rows of `df2` are added below `df1`.

Other Options:

- **merge():** Combines tables using common columns.
- **join():** Joins DataFrames using indexes.
- **relate():** Not a valid Pandas function.

Therefore, `concat()` is used for vertical combination of DataFrames.

Final Answer : "concat()"

Answer: (A)

Q18.

Solution

Concept: Missing values in Pandas are represented using `NaN`. Pandas provides methods to locate valid (non-null) entries in data.

Solution: The `first_valid_index()` method returns the index of the first non-null value in a Series.
Example:

```
import pandas as pd
import numpy as np

s = pd.Series([np.nan, np.nan, 10, 20])

print(s.first_valid_index())
```

Output:

```
2
```

Here, index 2 contains the first valid value.

Other Options:

- **first():** Not used for finding first non-null value.
- **valid():** Not a valid Pandas method.
- **start():** Not a valid Pandas method.

Therefore, `first_valid_index()` is the correct method.

Final Answer : "first_valid_index()"

Answer: (B)

Q19.

Solution

Concept: Duplicate rows can affect data accuracy. Pandas provides methods to identify duplicate entries in a DataFrame.

Solution: The `duplicated()` method detects duplicate rows and returns `True` for repeated rows. Example:

```
import pandas as pd

data = {'A':[1,2,1],
        'B':['X','Y','X']}

df = pd.DataFrame(data)

print(df.duplicated())
```

Output:

```
0    False
1    False
2     True
dtype: bool
```

Here, the third row is marked as duplicate because it matches the first row.

Other Options:

- **repeated():** Not a valid Pandas method.
- **clone():** Used for copying in other contexts.
- **copied():** Not a valid Pandas method.

Therefore, `duplicated()` is the correct method for detecting duplicate rows.

Final Answer : "duplicated()"

Answer: (B)

Q20.

Solution

Concept: Pandas DataFrames are two-dimensional data structures. The `iloc[]` function is used for integer-based indexing and slicing.

Solution: The expression:

`df.iloc[2:5]`

selects rows from index position 2 up to, but not including, position 5. Therefore, rows 2, 3, and 4 are selected.
Since multiple rows and all columns are returned, the output type is a `DataFrame`.

Example:

```
import pandas as pd

data = {'colA':[10,11,12,13,14],
        'colB':['a','b','c','d','e']}

df = pd.DataFrame(data)

print(df.iloc[2:5])
```

Output:

```
   colA colB
2    12    c
3    13    d
4    14    e
```

Other Options:

- **Series:** Returned when selecting a single row or column.
- **Scalar Value:** Returned when selecting a single element.
- **Tuple:** Not normally returned by `iloc[]`.

Therefore, `df.iloc[2:5]` returns a `DataFrame`.

Final Answer : "DataFrame"

Answer: (C)

Q21.

Solution

Concept: SQL (Structured Query Language) provides powerful clauses for data retrieval and manipulation. When performing aggregations (like calculating sums, averages, counts, or minimum/maximum values), it's often necessary to perform these calculations on subsets of data rather than the entire table. Grouping records allows these aggregate functions to be applied to distinct categories of data.

Solution: The 'GROUP BY' clause in SQL is specifically used to group rows that have identical values in one or more specified columns into a set of summary rows. This clause is almost always used in conjunction with aggregate functions (e.g., 'COUNT()', 'SUM()', 'AVG()', 'MIN()', 'MAX()') in the 'SELECT' statement. When 'GROUP BY' is used, the aggregate functions operate on each group independently, returning a single value for each group.

Example: If you have a table 'Orders' with columns '(OrderID, CustomerID, OrderDate, Amount)', and you want to find the total amount spent by each customer, you would use:

SELECT CustomerID, SUM(Amount) AS TotalSpent FROM Orders GROUP BY CustomerID;

This query groups all orders by the 'CustomerID' and then calculates the sum of 'Amount' for each unique 'CustomerID'.

Let's examine the other options:
(A) ORDER BY: This clause is used to sort the result set of a query in ascending or descending order based on one or more columns. It affects the presentation of the data, not how it's grouped for aggregation.
(C) HAVING: The 'HAVING' clause is used to filter groups of records after they have been grouped by the 'GROUP BY' clause and after aggregate functions have been applied. It filters groups based on conditions involving aggregate values, whereas 'WHERE' filters individual rows before grouping.
(D) DISTINCT: The 'DISTINCT' keyword is used in the 'SELECT' statement to eliminate duplicate rows from the result set, returning only unique combinations of the selected columns. It does not group records for aggregation in the same way 'GROUP BY' does.

Therefore, 'GROUP BY' is the correct SQL clause for grouping records with identical values to perform aggregate calculations.

Final Answer : "GROUP BY"

Answer: (B)

Q22.

Solution

Concept: Counting rows is one of the most basic and frequently performed operations in SQL, used to determine the total number of records in a table or the number of records that satisfy a certain condition. SQL provides specific aggregate functions for this purpose.

Solution: The ‘COUNT()‘ aggregate function is the standard SQL function used to count the number of rows in a table or a result set.
‘COUNT(*)‘: This syntax is used to count all rows in a table, regardless of whether any of their columns contain NULL values. It simply counts the total number of physical rows. This is the most common and efficient way to get a total row count.
‘COUNT(column_name)‘: This syntax counts the number of non-NULL values in a specific column. If a column has NULL values, those rows will not be included in the count for that column.
‘COUNT(DISTINCT column_name)‘: This counts the number of unique, non-NULL values in a specified column.

The query ‘SELECT COUNT(*) FROM student;‘ will correctly count all rows in the ‘student‘ table.

Let’s look at why the other options are incorrect:
(A) SELECT TOTAL(*) FROM student; ‘TOTAL()‘ is not a standard SQL aggregate function for counting rows. While some systems might have proprietary functions, ‘COUNT()‘ is universally recognized.
(C) SELECT NUMBER(*) FROM student; ‘NUMBER()‘ is not a standard SQL aggregate function for counting rows.
(D) SELECT SUM(*) FROM student; The ‘SUM()‘ function is used to calculate the sum of values in a numeric column. It cannot be used with ‘*‘ to count rows, and it expects a numeric column name as an argument (e.g., ‘SUM(grade)‘).

Therefore, ‘SELECT COUNT(*) FROM student;‘ is the universally correct and efficient SQL query to count the total number of rows in the ‘student‘ table.

Final Answer : “SELECT COUNT(*) FROM student;”

Answer: (B)

Q23.

Solution

Concept: SQL JOIN operations are used to combine data from two or more tables based on related columns. Different JOIN types decide which rows are included in the final result.

Solution: A `RIGHT JOIN` (or `RIGHT OUTER JOIN`) returns all rows from the table placed on the right side of the JOIN clause and the matching rows from the left table.
If no matching row exists in the left table, the result still includes the row from the right table, while columns from the left table contain `NULL` values.

Example:

```
SELECT C.Name, O.OrderID, O.Amount
FROM Customers C
RIGHT JOIN Orders O
ON C.CustomerID = O.CustomerID;
```

In this query:

- All rows from the `Orders` table are displayed.
- Matching customer details are fetched from the `Customers` table.
- If an order has no matching customer, the customer name becomes `NULL`.

Explanation of Other JOIN Types:

- **LEFT JOIN:** Returns all rows from the left table and matching rows from the right table.
- **INNER JOIN:** Returns only those rows where matching records exist in both tables.
- **CROSS JOIN:** Returns every possible combination of rows from both tables.

Therefore, `RIGHT JOIN` is the correct JOIN type that returns all rows from the right table and matching rows from the left table.

Final Answer : "RIGHT JOIN"

Answer: (C)

Q24.

Solution

Concept: SQL operators are special keywords or symbols used in ‘WHERE‘ clauses (and sometimes ‘HAVING‘ clauses) to specify conditions that filter data.

Solution: The ‘BETWEEN‘ operator in SQL is used to test whether a value falls within a specified range. The range includes both the lower and upper bounds. It is generally used with numeric values, dates, or strings.

The syntax is:
‘expression BETWEEN value1 AND value2‘

This is equivalent to:
‘expression >= value1 AND expression <= value2‘

Example:
To select products with a price between 10 and 20 (inclusive):

SELECT ProductName, Price FROM Products WHERE Price BETWEEN 10 AND 20;

To select orders placed in January 2023:

SELECT OrderID, OrderDate FROM Orders WHERE OrderDate BETWEEN ’2023-01-01’ AND ’2023-01-31’;

Let’s examine the other operators:
(A) LIKE: The ‘LIKE‘ operator is used for pattern matching within string values. It uses wildcard characters (% for any sequence of zero or more characters, _ for any single character). For example, ‘WHERE Name LIKE ’J%’‘ finds names starting with ’J’.
(C) IN: The ‘IN‘ operator is used to check if a value matches any value in a list of values or a subquery. For example, ‘WHERE City IN (’New York’, ’London’, ’Paris’)‘. It checks for discrete values, not a continuous range.
(D) EXISTS: The ‘EXISTS‘ operator is used in conjunction with a subquery to test for the existence of any rows returned by the subquery. It returns ‘TRUE‘ if the subquery returns one or more rows, and ‘FALSE‘ otherwise. It does not check if a specific value is within a range.

Therefore, ‘BETWEEN‘ is the correct SQL operator for checking if a value lies within a specified range (inclusive).

Final Answer : “BETWEEN”

Answer: (B)

Q25.

Solution

Concept: Aggregate functions in SQL are powerful tools used to perform calculations on a set of rows and return a single summary value. These functions are typically used with the ‘SELECT‘ statement and often with the ‘GROUP BY‘ clause to summarize data based on categories. They are essential for generating reports, statistics, and business insights from raw data.

Solution: The ‘SUM()‘ aggregate function is specifically designed to return the total sum of all non-NULL values in a specified numeric column. It operates exclusively on numeric data types. If a column contains non-numeric data, ‘SUM()‘ will typically result in an error or ‘0‘, depending on the database system and its data type conversion rules.

Example: To calculate the total sales amount from an ‘Orders‘ table:

SELECT SUM(Amount) AS TotalSales FROM Orders;

To calculate the total quantity of items sold for each product:

SELECT ProductID, SUM(Quantity) AS TotalQuantitySold FROM OrderDetails GROUP BY ProductID;

If the ‘Amount‘ or ‘Quantity‘ column contains ‘NULL‘ values, ‘SUM()‘ will ignore those ‘NULL‘ values and only sum the non-NULL entries.

Let’s examine the other options:
(A) ADD(): ‘ADD()‘ is not a standard SQL aggregate function for summing column values. While addition is a basic arithmetic operation, ‘SUM()‘ is the aggregate function for this purpose.
(B) TOTAL(): ‘TOTAL()‘ is not a standard SQL aggregate function. Similar to ‘ADD()‘, it’s not recognized for this functionality across database systems.
(D) COUNT(): The ‘COUNT()‘ aggregate function is used to count the number of rows or non-NULL values in a column. It does not perform summation of values. For example, ‘COUNT(salary)‘ counts how many employees have a non-NULL salary, not the sum of their salaries.

Therefore, ‘SUM()‘ is the correct aggregate function for calculating the total sum of values in a numeric column.

Final Answer : “SUM()”

Answer: (C)

Q26.

Solution

Concept: SQL provides different clauses for filtering data, each operating at a specific stage of query execution. The ‘WHERE‘ clause filters individual rows *before* any grouping or aggregation takes place. However, when you need to filter the results of aggregate functions (i.e., filter the groups themselves), a different clause is required.

Solution: The ‘HAVING‘ clause is specifically used to filter grouped records *after* they have been grouped by the ‘GROUP BY‘ clause and after aggregate functions have been applied. It works similar to a ‘WHERE‘ clause but applies conditions to groups based on aggregate values. You cannot use aggregate functions directly in a ‘WHERE‘ clause because ‘WHERE‘ operates on individual rows before aggregation.

The typical order of clauses in a ‘SELECT‘ statement involving grouping and filtering is:
‘SELECT ... FROM ... WHERE ... GROUP BY ... HAVING ... ORDER BY ...‘

Example:
To find departments where the average salary is greater than 50000:

SELECT Department, AVG(Salary) AS AverageSalary FROM Employees GROUP BY Department HAVING AVG(Salary) > 50000;

Here, ‘GROUP BY Department‘ first groups employees by their department. Then, ‘AVG(Salary)‘ is calculated for each department. Finally, ‘HAVING AVG(Salary) > 50000‘ filters these groups, keeping only those departments whose average salary exceeds 50000.

Let’s examine the other options:
(A) WHERE: The ‘WHERE‘ clause filters individual rows *before* any grouping occurs. It cannot contain aggregate functions directly. For example, ‘WHERE Salary > 50000‘ would filter individual employees before calculating department averages.
(B) GROUP BY: This clause groups rows with identical values in specified columns. It performs the grouping operation but does not filter the groups based on aggregate conditions.
(D) ORDER BY: This clause sorts the final result set (either individual rows or grouped aggregates) based on specified columns in ascending or descending order. It affects presentation, not filtering.

Therefore, ‘HAVING‘ is the correct clause for filtering grouped records after aggregation.

Final Answer : “HAVING”

Answer: (C)

Q27.

Solution

Concept: Finding the Nth highest or lowest value in SQL is commonly done using sorting and row selection. This is usually achieved with the `ORDER BY` clause along with `LIMIT` and `OFFSET`.

Solution: The query:

```
SELECT salary FROM emp ORDER BY salary DESC LIMIT 1,1;
```

is used to display the second highest salary from the `emp` table.

Explanation of the Query:

(a) **SELECT salary FROM emp** Retrieves the `salary` column from the `emp` table.

(b) **ORDER BY salary DESC** Sorts all salaries in descending order so that the highest salary appears first.

(c) **LIMIT 1,1** In MySQL:

- First `1` = number of rows to skip (OFFSET)
- Second `1` = number of rows to display

Thus, the query skips the first row (highest salary) and displays the next row, which is the second highest salary.

Example:
If salaries are:

70000, 50000, 50000, 30000, 10000

After skipping the first value (70000), the next salary displayed will be:

50000

Explanation of Other Options:

- **SELECT MAX(salary) FROM emp;** Returns only the highest salary.
- **SELECT SECOND(salary) FROM emp;** `SECOND()` is not a valid SQL function.
- **SELECT salary FROM emp WHERE MAX(salary);** This query is syntactically incorrect.

Therefore, the correct query to display the second highest salary is:

```
SELECT salary FROM emp ORDER BY salary DESC LIMIT 1,1;
```

Final Answer : "SELECT salary FROM emp ORDER BY salary DESC LIMIT 1,1;"

Answer: (B)

Q28.

Solution

Concept: SQL statements are divided into categories based on their purpose. Data Manipulation Language (DML) commands are used to manage data stored in tables, while Data Definition Language (DDL) commands are used to define or modify the database structure.

Solution: The `DELETE` statement is a DML command used to remove existing rows from a table. It deletes only the data and keeps the table structure, indexes, and constraints unchanged. Usually, the `WHERE` clause is used to specify which rows should be removed.

Example 1: Delete a Specific Record

```
DELETE FROM Employees
WHERE EmployeeID = 101;
```

This command deletes the employee whose `EmployeeID` is 101.

Example 2: Delete Multiple Rows

```
DELETE FROM Employees
WHERE Department = 'Sales';
```

This command removes all employees belonging to the Sales department.

Example 3: Delete All Rows

```
DELETE FROM Employees;
```

If the `WHERE` clause is omitted, all rows from the table are deleted, but the table structure remains intact.

Explanation of Other Options:

- **DROP:** A DDL command used to remove the entire table or database object along with its structure.
- **REMOVE:** Not a standard SQL command.
- **CLEAR:** Not a standard SQL command for deleting rows.

Therefore, `DELETE` is the correct SQL statement used to remove rows from a table while preserving the table structure.

Final Answer : "DELETE"

Answer: (C)

Q29.

Solution

Concept: SQL JOIN clauses are used to combine rows from two or more tables based on a relationship between them. Each type of JOIN produces a different result set based on how it handles matching and non-matching rows. One specific type of join creates a Cartesian product, which results in every possible combination of rows from the participating tables.

Solution: A ‘CROSS JOIN‘ produces a Cartesian product of the two tables involved. This means that every row from the first table is combined with every row from the second table. The result set will contain ‘(number of rows in table1) * (number of rows in table2)‘ rows. There is no ‘ON‘ or ‘WHERE‘ clause specified with a ‘CROSS JOIN‘ to define a matching condition, as its purpose is to create all possible combinations.

Example: If ‘TableA‘ has 3 rows and ‘TableB‘ has 4 rows:
SELECT * FROM TableA CROSS JOIN TableB;

This query would return ‘3 * 4 = 12‘ rows, where each row from ‘TableA‘ is paired with each row from ‘TableB‘.

Cross joins are rarely used for combining related data, but they can be useful in specific scenarios, such as:
Generating all possible combinations for testing purposes.
Creating a set of permutations or combinations.
When combined with a ‘WHERE‘ clause, a ‘CROSS JOIN‘ can sometimes simulate an ‘INNER JOIN‘ (though an explicit ‘INNER JOIN‘ is preferred for clarity and performance).

Let’s examine the other JOIN types:
(A) INNER JOIN: Returns only the rows where there is a match in *both* tables based on a specified join condition. It does not produce all possible combinations if the condition is not met for all pairs.
(B) LEFT JOIN: Returns all rows from the *left* table and the matching rows from the right table. Non-matching rows from the right table are represented by NULLs. It does not produce every possible combination.
(D) NATURAL JOIN: This is a type of ‘INNER JOIN‘ that implicitly joins tables based on columns with the same name and data type in both tables. It only includes matching rows and removes duplicate columns, so it does not return every possible combination.

Therefore, the ‘CROSS JOIN‘ is the specific join type that returns every possible combination of rows from two tables.

Final Answer : “CROSS JOIN”

Answer: (C)

Q30.

Solution

Concept: When retrieving data from a database, the order in which the records are returned by default is not guaranteed and can vary. To present data in a meaningful and readable sequence, SQL provides a specific clause to sort the result set based on the values of one or more columns. This allows users to arrange records in ascending or descending order.

Solution: The ‘ORDER BY‘ clause in SQL is used to arrange the rows (records) in the result set of a ‘SELECT‘ query in a sorted order. You can specify one or more columns by which to sort, and for each column, you can define the sort order as ascending or descending.

Syntax:
‘SELECT column1, column2, ...‘ ‘FROM table_name‘ ‘ORDER BY column_name1 [ASC|DESC], column_name2 [ASC|DESC], ...;‘

‘ASC‘ (Ascending): Sorts the records from the lowest value to the highest value. This is the default sort order if neither ‘ASC‘ nor ‘DESC‘ is specified.
‘DESC‘ (Descending): Sorts the records from the highest value to the lowest value.

Example:
To select all employees and sort them by their last name in ascending order:

SELECT EmployeeID, FirstName, LastName FROM Employees ORDER BY LastName ASC;

To select products and sort them by price in descending order, and then by product name in ascending order for products with the same price:

SELECT ProductID, ProductName, Price FROM Products ORDER BY Price DESC, ProductName ASC;

Let’s examine the other options:
(A) SORT BY: ‘SORT BY‘ is not a standard SQL clause. The correct clause is ‘ORDER BY‘.
(B) GROUP BY: This clause is used to group rows that have identical values in specified columns, typically for performing aggregate calculations. It does not arrange the records in sorted order.
(D) ARRANGE BY: ‘ARRANGE BY‘ is not a standard SQL clause.

Therefore, the ‘ORDER BY‘ clause is the correct SQL clause used to arrange records in a sorted order within the result set.

Final Answer : “ORDER BY”

Answer: (C)

Q31.

Solution

Concept: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It provides a wide array of plotting functions to generate various types of charts.

Solution: The ‘plt.plot()‘ function (assuming ‘import matplotlib.pyplot as plt‘) is the primary and most versatile function in Matplotlib used to draw a wide range of plots, including line graphs. When provided with a single list or array of data, ‘plt.plot()‘ will plot them against their indices. When given two lists/arrays, it will plot the first against the second (e.g., x-coordinates vs. y-coordinates), connecting the points with lines by default.

Example:

```
import matplotlib.pyplot as plt

# Data for a line graph
x_values = [1, 2, 3, 4, 5]
y_values = [2, 4, 1, 5, 3]

# Create a line graph
plt.plot(x_values, y_values, marker='o', linestyle='-')
plt.title("Sample Line Graph")
plt.xlabel("X-axis Label")
plt.ylabel("Y-axis Label")
plt.grid(True)
plt.show()
```

Let’s examine the other options:
(A) plt.line(): There is no standard Matplotlib function explicitly named ‘plt.line()‘ to draw a line graph. While you can draw lines with ‘plt.axhline()‘ or ‘plt.axvline()‘ for horizontal/vertical lines, ‘plt.plot()‘ is the general-purpose function for data series.
(C) plt.graph(): This is not a standard Matplotlib function.
(D) plt.draw(): The ‘plt.draw()‘ function is used to redraw the current figure, typically after modifications have been made to an existing plot, but it does not *create* a new graph itself. It is usually called in interactive mode to update the display.

Therefore, ‘plt.plot()‘ is the correct and most commonly used Matplotlib function for drawing line graphs.

Final Answer : “plt.plot()”

Answer: (B)

Q32.

Solution

Concept: The ‘plt.plot()‘ function in Matplotlib is highly customizable, allowing users to control various aspects of the line and markers displayed on the graph. Customizing markers helps in distinguishing data points and improving readability.

Solution: The ‘marker‘ parameter in ‘plt.plot()‘ is specifically used to specify the style of markers that are plotted at each data point. Markers are symbols (like circles, squares, triangles, stars) that represent the actual data points, distinct from the lines connecting them.

Example:

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [2, 4, 1, 5, 3]

# Plotting with different marker styles
plt.plot(x, y, marker='o', label='Circle Markers') # 'o' for circle
plt.plot(x, [val + 0.5 for val in y], marker='s', label='Square Markers') # 's' for square
plt.plot(x, [val - 0.5 for val in y], marker='^',label='TriangleUpMarkers')#' for triangleup
plt.title("Plot with Different Marker Styles")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.legend()
plt.grid(True)
plt.show()
```

Common marker styles include: ‘’o’‘ (circle), ‘’s’‘ (square), ‘’^‘ (triangle up), ‘’v’‘ (triangle down), ‘’*’‘ (star), ‘’+’‘ (plus), ‘’x’‘ (x), ‘’.’‘ (point).

Let’s examine the other parameters:
(A) color: The ‘color‘ parameter specifies the color of the line and/or markers. For example, ‘color=’red’‘.
(C) linestyle: The ‘linestyle‘ (or ‘ls‘) parameter specifies the style of the line connecting the data points (e.g., ‘’-’‘ for solid, ‘’–’‘ for dashed, ‘’:’‘ for dotted).
(D) width: While there’s a ‘linewidth‘ parameter for the line thickness, ‘width‘ itself is not a standard parameter for marker style. Marker size is controlled by ‘markersize‘ or ‘ms‘.

Therefore, ‘marker‘ is the parameter specifically used to specify the marker style in ‘plt.plot()‘.

Final Answer : “marker”

Answer: (B)

Q33.

Solution

Concept: When working with Matplotlib, especially in interactive environments or scripts that generate multiple plots, it's often necessary to manage the display of figures and axes. This includes creating new plots, displaying existing ones, and clearing previously drawn elements.

Solution: The 'plt.clf()' function (assuming 'import matplotlib.pyplot as plt') is used to clear the current figure. This means it removes all axes, subplots, and any other elements from the currently active figure window, but it does not close the figure window itself.

Example:

```
import matplotlib.pyplot as plt
import numpy as np

# Create the first plot
plt.figure(1) # Create a figure with ID 1
plt.plot(np.sin(np.linspace(0, 2*np.pi, 100)))
plt.title("Plot 1")
plt.show(block=False) # Show without blocking, so we can clear it

# Wait a moment, then clear the figure and plot something new
plt.pause(2) # Pause for 2 seconds
plt.clf() # Clear Figure 1
plt.plot(np.cos(np.linspace(0, 2*np.pi, 100)), 'r--')
plt.title("Plot 2 (in the same figure window)")
plt.show()
```

Let's examine the other options: (A) plt.clean(): This is not a standard Matplotlib function for clearing figures.
(B) plt.remove(): This is not a standard Matplotlib function for clearing figures. There are methods like 'ax.remove()' to remove an axis from a figure, but 'plt.remove()' is not for clearing the entire figure.
(D) plt.reset(): This is not a standard Matplotlib function for clearing figures.
'matplotlib.pyplot.close()' is used to close a figure window, and 'plt.cla()' clears the current *axes* (subplot), not the entire figure.

Therefore, 'plt.clf()' is the correct Matplotlib function specifically designed to clear the current figure.

Final Answer : "plt.clf()"

Answer: (C)

Q34.

Solution

Concept: Data visualization involves choosing the most appropriate chart type to effectively communicate insights from data. Different chart types are best suited for different kinds of data and different analytical goals. Representing trends over time is a very common task in data analysis, requiring a chart that can clearly show changes and patterns over a continuous period.

Solution: A Line Graph is the most suitable chart for representing trends over time.
Purpose: Line graphs are specifically designed to illustrate how one or more data points (variables) change continuously over a period. The x-axis typically represents time (e.g., dates, months, years, hours), and the y-axis represents the quantitative value of the variable being measured.
Effectiveness: By connecting data points with lines, a line graph makes it very easy to visualize direction (upward, downward), magnitude (steepness of slope), and overall patterns or trends (e.g., seasonality, growth, decline, cyclical behavior) in the data as time progresses. Multiple lines can be plotted on the same graph to compare trends of different categories or variables.

Example scenarios for line graphs:
Stock prices over months.
Temperature changes throughout a day.
Website traffic over weeks.
Sales figures over fiscal quarters.

Let's examine why the other chart types are less suitable for representing trends over time: (A) Pie Chart: Pie charts are used to show the proportion or percentage of different categories that make up a whole. They are static and do not show change over time.
(B) Histogram: A histogram is used to represent the distribution of a single numerical variable. It shows the frequency of data points falling into different bins or ranges. While it deals with quantitative data, it doesn't primarily show changes *over time*.
(D) Bar Graph (or Bar Chart): Bar graphs are excellent for comparing discrete categories or showing quantities at specific points. While they can show changes over time if the x-axis represents distinct time intervals (e.g., sales per year), a line graph is generally more effective for emphasizing the *continuity* and *flow* of a trend, especially when there are many time points or when interpolation between points is meaningful. For discrete time points, a bar chart can work, but for continuous trends, line graphs excel.

Therefore, a Line Graph is the most appropriate and effective choice for visualizing trends over time due to its ability to clearly depict continuous change and patterns.

Final Answer : "Line Graph"

Answer: (C)

Q35.

Solution

Concept: Matplotlib provides several customization options to improve the appearance of plots. One such feature is transparency, which controls how visible or see-through plot elements appear.

Solution: The `alpha` parameter in Matplotlib is used to control the transparency of plot elements such as lines, bars, scatter points, and filled regions.
The value of `alpha` ranges from:

- `alpha = 0.0` : Completely transparent (invisible)
- `alpha = 1.0` : Completely opaque (solid)
- Values between 0 and 1 produce semi-transparent effects

Example:

```
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 10, 100)
y = np.sin(x)

plt.plot(x, y, color='blue', alpha=0.5)

plt.title("Transparency Example")
plt.show()
```

In this example, the plotted line appears semi-transparent because `alpha=0.5`.

Explanation of Other Options:

- **shade:** Not a standard Matplotlib parameter for transparency.
- **opacity:** Commonly used in CSS and graphics software, but Matplotlib uses `alpha` instead.
- **fade:** Not a valid Matplotlib parameter.

Therefore, `alpha` is the correct parameter used to control transparency in Matplotlib plots.

Final Answer : "alpha"

Answer: (B)

Q36.

Solution

Concept: Cyber threats are malicious acts that aim to damage, disrupt, or gain unauthorized access to computer systems, networks, or data. These threats come in various forms, each with distinct characteristics and methods of propagation. Understanding these different types is crucial for implementing effective cybersecurity measures.

Solution: A Worm is a type of standalone malicious software (malware) that replicates itself automatically across computer networks without human interaction.
Self-replication: Unlike viruses (which require a host program to spread), worms are self-contained and self-replicating. They find vulnerabilities in network protocols, operating systems, or applications, and then exploit these weaknesses to copy themselves to other computers.
Network Propagation: Worms are specifically designed to spread across networks. They can use email attachments, network shares, remote execution vulnerabilities, or instant messaging to propagate rapidly.
Impact: Worms consume network bandwidth, slow down systems, and can carry payloads that perform other malicious activities, such as deleting files, installing backdoors, or launching denial-of-service attacks. The speed and independence of their propagation make them particularly dangerous.

Let's examine the other cyber threats:
(A) Trojan Horse: A Trojan horse is a type of malware that disguises itself as legitimate software. It does not self-replicate like a worm. Instead, it relies on user deception to get installed. Once inside, it performs malicious actions like creating backdoors, stealing data, or installing other malware.
(C) Spyware: Spyware is software that secretly observes the user's computer activities without their permission. It collects personal information, browsing habits, and other data, typically for advertising or nefarious purposes. It does not self-replicate.
(D) Adware: Adware is software that automatically displays or downloads unwanted advertisements (pop-ups, banners). While annoying and potentially a privacy concern, its primary purpose is usually commercial (generating revenue for its creator), not self-replication or direct system damage, though some adware can be bundled with spyware.

Therefore, a Worm is the cyber threat that replicates itself automatically across networks.

Final Answer : "Worm"

Answer: (B)

Q37.

Solution

Concept: Malicious software, or malware, encompasses various types of programs designed to perform unwanted and harmful actions on computer systems. These programs employ different tactics to gain access, evade detection, and execute their malicious payloads. Understanding how they disguise themselves is key to preventing infection.

Solution: A Trojan Horse is a type of malicious software that disguises itself as legitimate or desirable software. It appears to be a harmless, useful, or innocent program (e.g., a free game, a utility, a media player, a fake security update). However, once executed, it carries a hidden malicious payload that performs unauthorized actions.
Deception: Trojans rely on social engineering, tricking users into downloading and executing them. They do not self-replicate like viruses or worms.
Payloads: Once inside, a Trojan can perform various malicious activities, such as:
Creating backdoors for remote access.
Stealing personal data, credentials, or financial information.
Installing other types of malware (e.g., ransomware, spyware).
Logging keystrokes.
Using the infected computer in a botnet for DDoS attacks.

Example: A user might download a "free software activator" that is actually a Trojan designed to steal their banking credentials.

Let's examine the other options:
(A) Worm: A worm is a self-replicating malware that spreads across networks independently, without needing to attach to a legitimate program or deceive a user into running it. It doesn't primarily disguise itself as legitimate software in the same way a Trojan does.
(C) Antivirus: Antivirus software is a *security tool* designed to detect, prevent, and remove malware (including Trojans, worms, and viruses). It is not a type of malicious software.
(D) Firewall: A firewall is a network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules. It is a protective measure, not a malicious software program.

Therefore, a Trojan Horse is the correct term for malicious software that disguises itself as a legitimate program.

Final Answer : "Trojan Horse"

Answer: (B)

Q38.

Solution

Concept: Authentication is the process of verifying the identity of a user, process, or device. It is a critical component of cybersecurity, ensuring that only authorized entities can access resources. Different authentication methods offer varying levels of security.

Solution: Multi-factor Authentication (MFA) is an authentication method that requires a user to provide two or more different forms of verification (or "factors") to gain access to an account or system. This significantly enhances security because even if one factor is compromised (e.g., a password is stolen), an attacker still needs to overcome the other factors.

The three main categories of authentication factors are:
1. Something you know: (Knowledge factor) e.g., password, PIN, security questions.
2. Something you have: (Possession factor) e.g., a physical token, a smartphone (for receiving a code via SMS or an authenticator app), smart card.
3. Something you are: (Inherence factor) e.g., fingerprint, facial recognition, iris scan (biometrics).

For MFA, you typically combine factors from at least two *different* categories. For example, a common MFA setup involves a password (something you know) plus a code sent to your phone (something you have).

Let's examine the other options:
(A) Simple Authentication: This usually refers to single-factor authentication, where only one form of verification (e.g., just a password) is required.
(C) Password Authentication: This is a specific type of single-factor authentication that relies solely on a password (something you know). It's a component of authentication but not a multi-factor method itself.
(D) Public Authentication: This is not a standard term for an authentication method.
Authentication can involve public keys (as in public-key cryptography), but "Public Authentication" as a general method requiring two forms of verification is incorrect.

Therefore, Multi-factor Authentication is the correct method that requires two different forms of verification for enhanced security.

Final Answer : "Multi-factor Authentication"

Answer: (B)

Q39.

Solution

Concept: Cybercrime refers to criminal activities carried out using computers or networks. These crimes encompass a wide range of illicit activities, from data theft and fraud to system damage and unauthorized access. Understanding the terminology associated with various cybercrimes is important for identifying and addressing security threats.

Solution: Hacking is the cybercrime that broadly involves gaining unauthorized access to computer systems or networks. It often implies a deliberate act of breaking into a system, bypassing security measures, or exploiting vulnerabilities to gain control or access that is not intended for the perpetrator.
Methods: Hacking can involve various techniques, such as exploiting software vulnerabilities, guessing passwords, using social engineering to trick users, or deploying malware to gain entry.
Motives: Motives for hacking can range from curiosity and challenge to financial gain, espionage, political activism (hacktivism), or simply causing disruption.

The term "hacking" specifically describes the act of unauthorized access.

Let's examine the other options: (A) Phishing: Phishing is a type of social engineering attack where attackers attempt to trick individuals into revealing sensitive information (like usernames, passwords, credit card details) by impersonating a trustworthy entity in electronic communications (e.g., fake emails, websites, texts). While it can lead to unauthorized access, phishing is the method of obtaining credentials, not the unauthorized access itself.
(C) Spamming: Spamming refers to the sending of unsolicited messages, usually commercial advertising, in bulk. It is a nuisance and can be used for spreading malware or phishing, but it is not inherently about gaining unauthorized access to systems.
(D) Blogging: Blogging is the act of creating and publishing content on a blog (an online journal or informational website). It is a legitimate online activity and not a cybercrime.

Therefore, Hacking is the term that directly describes the cybercrime involving unauthorized access to computer systems.

Final Answer : "Hacking"

Answer: (B)

Q40.

Solution

Concept: Electronic waste, or e-waste, refers to discarded electrical or electronic devices. It is a growing environmental concern due to its volume and the hazardous materials (like lead, mercury, cadmium) it often contains, which can leach into the soil and water if not properly managed. Sustainable practices are crucial for mitigating the negative impacts of e-waste.

Solution: Recycling and refurbishing devices is the most effective and environmentally responsible practice for reducing electronic waste.
Recycling: This involves processing discarded electronic devices to recover valuable materials (e.g., gold, silver, copper, plastics) and safely dispose of hazardous components. Recycling reduces the need for new raw materials, conserves energy, and prevents toxic substances from polluting the environment. Specialized facilities are designed to handle e-waste.
Refurbishing: This involves repairing, restoring, and upgrading used electronic devices to make them fully functional again, extending their lifespan. Refurbished devices can then be resold or donated, reducing the demand for new products and keeping existing electronics out of landfills for longer. This practice is part of the "reduce, reuse, recycle" hierarchy, focusing on reuse.

Let's examine why the other options are harmful or incorrect:
(A) Burning old devices: Burning e-waste, especially in uncontrolled environments, releases toxic chemicals (e.g., dioxins, furans, heavy metals) into the air, soil, and water. This is extremely harmful to human health and the environment and is an illegal practice in many places.
(B) Throwing electronics in open landfills: Disposing of electronics in general landfills allows hazardous materials to leach into the ground and contaminate soil and groundwater. It also means valuable resources are lost and cannot be recovered. Most e-waste should not go into regular landfills.
(D) Mixing e-waste with household waste: This practice is essentially the same as throwing it in landfills, as most household waste ends up there. It prevents proper sorting and specialized recycling, leading to environmental contamination and loss of valuable materials.

Therefore, recycling and refurbishing devices are the primary and responsible methods for reducing electronic waste.

Final Answer : "Recycling and refurbishing devices"

Answer: (C)

Q41.

Solution

Concept: Software licensing and distribution models vary significantly, affecting how users can obtain, use, modify, and distribute software. These models dictate the cost, access to source code, and rights granted to the end-user. Understanding these distinctions is crucial for both software users and developers.

Solution: Freeware is a type of software that is distributed free of charge to the end-user. While it is free to use, its source code is typically unavailable, meaning users cannot modify, study, or redistribute modified versions of the software. The copyright holder usually retains all rights, and users are bound by the terms of the license agreement, which typically permits free use but restricts other actions.

Examples of Freeware include Adobe Reader, Skype, Google Chrome, and many mobile apps. Users can download and use these applications without paying, but they cannot access or change the underlying programming code.

Let's examine the other software types:
(A) Open Source Software: This software is characterized by its source code being publicly available. Users are typically granted the right to use, study, change, and distribute the software and its modified versions to anyone and for any purpose. While often free, open-source software can sometimes have a cost for support or enterprise versions. The key differentiator is the availability of the source code and the rights granted.
(C) Shareware: Shareware is distributed free for an evaluation period or with limited functionality. Users are encouraged to try it out, and if they wish to continue using it beyond the trial period or unlock full features, they are expected to pay a fee to register the software. The source code is generally not provided.
(D) Proprietary Sourceware: This term is not standard. The general term is Proprietary Software (or closed-source software). This type of software is legally owned by an individual or company, and its source code is kept secret. Users must typically purchase a license to use it, and they have very limited rights to modify or redistribute it. Freeware can be considered a subset of proprietary software (where the cost is zero).

Therefore, Freeware is the correct term for software distributed free of cost but with unavailable source code.

Final Answer : "Freeware"

Answer: (B)

Q42.

Solution

Concept: The rapid evolution of computer and network technology has led to the emergence of new forms of crime. Legal frameworks need to adapt to address these new challenges, including cybercrime, data protection, and digital rights. Specific laws and legal branches have been developed to govern activities in the digital realm.

Solution: Cyber Law (also known as Internet Law or IT Law) is the branch of law that specifically deals with legal issues related to the Internet, cyberspace, and computing. It covers a wide range of topics, including:
Cybercrime: Crimes committed using computers and networks (e.g., hacking, data theft, online fraud, cyberstalking, phishing, malware distribution).
Data Protection and Privacy: Laws governing the collection, storage, and use of personal data (e.g., GDPR, HIPAA).
Electronic Commerce: Legal aspects of online transactions and digital contracts.
Intellectual Property Rights in the Digital Age: Protecting copyrights, trademarks, and patents in the context of digital content and software.
Freedom of Speech and Censorship online.

Cyber law aims to regulate digital interactions and ensure justice and order in the virtual world.

Let's examine the other options, which are branches of intellectual property law but not general laws covering all computer and network crimes:
(A) Patent Law: This law protects inventions (new and useful processes, machines, manufactures, or compositions of matter). While software can sometimes be patented, it's not the general law for cybercrimes.
(B) Copyright Law: This law protects original works of authorship (e.g., literary, dramatic, musical, artistic, and certain other intellectual works), including software code, books, music, and movies. It deals with unauthorized copying or distribution, but not the full spectrum of cybercrimes like hacking or online fraud.
(D) Trademark Law: This law protects brand names, logos, and slogans used to identify and distinguish goods and services of one party from those of others. It deals with brand infringement, not broadly with computer crimes.

Therefore, Cyber Law is the specific legal field that deals comprehensively with crimes committed using computers and networks.

Final Answer : "Cyber Law"

Answer: (C)

Q43.

Solution

Concept: Networking protocols are sets of rules that govern how data is formatted, transmitted, and received across networks. Different protocols are designed for specific purposes, such as browsing web pages, sending emails, or transferring files. Understanding these protocols is fundamental to network communication.

Solution: FTP (File Transfer Protocol) is a standard network protocol used for transferring computer files between a client and a server on a computer network. It is one of the oldest and most widely used application-layer protocols for moving files.
Functionality: FTP allows users to upload files to a server, download files from a server, list directory contents, and manipulate files/directories on the server.
Mechanism: FTP typically uses two separate channels: a command channel for controlling the conversation (sending commands, receiving replies) and a data channel for actually transferring file data.

Example use cases for FTP:
Web developers uploading web pages to a web server.
Users downloading software updates from a server.
Backing up files to a remote server.

Let's examine the other networking protocols:
(A) HTTP (Hypertext Transfer Protocol): This is the foundation of data communication for the World Wide Web. It is used to transfer web pages (HTML files, images, videos, etc.) from web servers to web browsers. While it transfers files, its primary role is web content delivery, not general file transfer.
(C) SMTP (Simple Mail Transfer Protocol): This protocol is used for sending outgoing email messages from an email client to an email server, and between email servers. It is strictly for email transmission.
(D) POP3 (Post Office Protocol version 3): This protocol is used by email clients to retrieve (download) email messages from an email server to a local computer. It's for receiving email, often deleting it from the server after download.

Therefore, FTP is the specific networking protocol used for transferring files between computers.

Final Answer : "FTP"

Answer: (B)

Q44.

Solution

Concept: Connecting a computer to a network requires specific hardware components that enable the computer to send and receive data over the network medium. These components convert digital data into signals that can travel across cables or wireless channels and vice versa.

Solution: A NIC (Network Interface Card), also known as a network adapter or Ethernet card, is the device that provides the physical and logical interface between a computer and a network cable (or wireless network).
Function:
Physical Connection: It provides the port (e.g., RJ45 for Ethernet) where the network cable plugs in.
Data Conversion: It converts the digital data from the computer's CPU into electrical signals (for wired networks) or radio waves (for wireless networks) that can travel over the network medium. It also performs the reverse conversion for incoming signals.
MAC Address: Each NIC has a unique physical address called a MAC (Media Access Control) address, which is used for identifying the device within a local network segment. OSI Layers: NICs primarily operate at the Physical (Layer 1) and Data Link (Layer 2) layers of the OSI model.

Essentially, without a NIC, a computer cannot physically connect to a wired network or communicate over a wireless one.

Let's examine the other devices:
(A) Router: A router is a network device that forwards data packets between different computer networks. It operates at the Network Layer (Layer 3) of the OSI model and uses IP addresses to determine the best path for data. It connects networks, not primarily a single computer to a cable.
(B) Switch: A switch is a network device that connects multiple devices within a Local Area Network (LAN). It operates at the Data Link Layer (Layer 2) and intelligently forwards data frames to specific destination devices based on their MAC addresses. It connects multiple computers within a LAN, but not a single computer to the raw network cable.
(D) Repeater: A repeater is a physical layer (Layer 1) device that regenerates and extends network signals over longer distances. It strengthens a weakened signal but does not provide an interface for a computer to connect to a network.

Therefore, a NIC is the specific device that provides the interface between a computer and a network cable.

Final Answer : "NIC"

Answer: (C)

Q45.

Solution

Concept: The Internet relies on a hierarchical and distributed naming system to translate human-readable domain names (like ‘www.example.com‘) into numerical IP addresses (like ‘192.0.2.1‘) that computers use to identify each other on the network. This translation service is fundamental to how we navigate the web and access online resources.

Solution: The full form of DNS is Domain Name System.
Purpose: DNS acts like the phonebook of the Internet. Instead of remembering complex IP addresses for every website or service, users can type easy-to-remember domain names. DNS translates these domain names into the corresponding IP addresses that computers use to locate and connect to resources on the network.
Mechanism: When you type a website address into your browser, your computer sends a query to a DNS server. The DNS server then looks up the domain name and returns the associated IP address.
Your computer then uses this IP address to establish a connection with the website’s server.
Distributed System: DNS is a distributed database system, meaning no single server holds all the information. Instead, it’s organized hierarchically with many DNS servers collaborating to manage the vast amount of domain name records.

Without DNS, users would have to manually enter IP addresses to access websites, which would be highly impractical and difficult to manage.

Let’s examine the other options:
(B) Data Naming Service: While DNS deals with naming data (domains), "Data Naming Service" is not its official full form.
(C) Digital Network Server: This is a generic term that does not specifically refer to the domain name resolution system.
(D) Domain Network Security: While DNSSEC (Domain Name System Security Extensions) adds security to DNS, "Domain Network Security" is not the full form of DNS itself.

Therefore, the full form of DNS is Domain Name System.

Final Answer : “Domain Name System”

Answer: (A)

Q46.

Solution

Concept: Pandas `Series` objects are one-dimensional labeled arrays used for storing and analyzing data. Pandas provides built-in methods such as `sum()`, `mean()`, `min()`, and `max()` for statistical calculations.

Solution: The `sum()` method in Pandas is used to calculate the total sum of values in a Series. It works mainly with numeric data and ignores missing values (`NaN`) by default.

Example 1: Sum of a Numeric Series

```
import pandas as pd
s = pd.Series([10, 20, 30, 40, 50])
print(s.sum())
```

Output:

```
150
```

Calculation:

$$10 + 20 + 30 + 40 + 50 = 150$$

Example 2: Series with NaN Values

```
import pandas as pd
import numpy as np
s = pd.Series([1, 2, np.nan, 4, 5])
print(s.sum())
```

Output:

```
12.0
```

Here, the `NaN` value is ignored:

$$1 + 2 + 4 + 5 = 12$$

Other Options:

- **total():** Not a standard Pandas method.
- **add():** Used for element-wise addition, not for calculating total sum.
- **calculate():** Not a valid Pandas method.

Therefore, `sum()` is the correct method used to calculate the sum of values in a Pandas Series.

Final Answer : "sum()"

Answer: (C)

Q47.

Solution

Concept: Pandas provides built-in methods to calculate descriptive statistics such as minimum, maximum, mean, and median values from Series or DataFrames.

Solution: The ‘min()‘ method in Pandas is used to return the minimum value from a Series. It ignores ‘NaN‘ values by default while calculating the result.

Example:

```
import pandas as pd
s = pd.Series([10, 5, 20, 3, 15])
print(s.min())
```

Output:

```
3
```

Example with ‘NaN‘ values:

```
import pandas as pd
import numpy as np
s = pd.Series([1.0, 2.0, np.nan, 0.5, 3.0])
print(s.min())
```

Output:

```
0.5
```

Other options:
(A) ‘low()‘ is not a valid Pandas method.
(B) ‘minimum()‘ is not a standard Pandas method.
(D) ‘smallest()‘ is not directly used to return the minimum value of a Series.

Therefore, ‘min()‘ is the correct Pandas method used to find the minimum value in a Series.

Final Answer : “min()”

Answer: (C)

Q48.

Solution

Concept: In SQL, 'NULL' represents the absence of a value or an unknown value. It is distinct from zero, an empty string, or any other value. Because 'NULL' is not a value but a marker for missing data, standard comparison operators ('=', '!=', '<', '>') do not work as expected with 'NULL'. Special operators are required to correctly identify or exclude 'NULL' values.

Solution: The 'IS NULL' operator is the correct and standard SQL operator used to check whether a value in a column is 'NULL'.
Conversely, 'IS NOT NULL' is used to check if a value is not 'NULL'.

Example:
To select all employees who do not have an assigned department (i.e., their 'DepartmentID' is 'NULL'):

SELECT EmployeeName FROM Employees WHERE DepartmentID IS NULL;

To select all employees who do have an assigned department:

SELECT EmployeeName FROM Employees WHERE DepartmentID IS NOT NULL;

It's crucial to understand that 'NULL = NULL' does not evaluate to 'TRUE'. Instead, it evaluates to 'UNKNOWN' because, by definition, two unknown values cannot be determined to be equal. This is why standard comparison operators are unsuitable for 'NULL' checks.

Let's examine the other options:
(A) = NULL: This is an incorrect syntax and logic. 'WHERE column_name = NULL' will typically not return any rows, as 'NULL = NULL' is 'UNKNOWN', not 'TRUE'.
(C) NULL =: This is an incorrect syntax.
(D) HAS NULL: This is not a standard SQL operator for checking for NULL values.

Therefore, 'IS NULL' is the only correct SQL operator specifically designed to check for 'NULL' values in a column.

Final Answer : "IS NULL"

Answer: (B)

Q49.

Solution

Concept: SQL wildcard symbols are used with the ‘LIKE‘ operator to search for specific patterns in text data.

Solution: The ‘%‘ (percent sign) wildcard in SQL represents zero or more characters. It can match any sequence of characters, making it very useful for flexible searches.

Examples:

‘LIKE ’J%’‘ → Matches values starting with J (‘John‘, ‘Jane‘, ‘Jack‘)

‘LIKE ’%er’‘ → Matches values ending with “er” (‘Computer‘, ‘Mixer‘)

‘LIKE ’%new%’‘ → Matches values containing “new” (‘New York‘, ‘Newark‘)

Other options:

(A) ‘_‘ matches exactly one character.

(B) ‘*‘ is used in ‘SELECT *‘ for all columns, not for pattern matching.

(D) ‘#‘ is not a standard SQL wildcard symbol.

Therefore, ‘%‘ correctly represents zero or more characters in SQL.

Final Answer : “%”

Answer: (C)

Q50.

Solution

Concept: SQL provides Data Definition Language (DDL) commands for managing the database schema and Data Manipulation Language (DML) commands for managing data within tables.

Solution: The ‘DROP TABLE‘ command is a DDL (Data Definition Language) statement that completely removes an entire table from the database. This includes:
Table Structure (Schema): The definition of the table, including its column names, data types, constraints, and indexes, is permanently deleted.
All Data: All rows and data contained within the table are irrevocably removed.

Once a table is dropped, it is generally difficult or impossible to recover without a database backup. ‘DROP TABLE‘ operations are typically auto-committed and cannot be rolled back in most database systems.

Example:
To completely remove a table named ‘Employees‘:

DROP TABLE Employees;

This command would eliminate the ‘Employees‘ table and all its data from the database.

Let’s examine the other commands:
(A) DELETE: This is a DML (Data Manipulation Language) command. ‘DELETE FROM table_name;‘ or ‘DELETE FROM table_name WHERE condition;‘ removes *rows* (data) from a table, but it leaves the table’s structure (schema) intact. ‘DELETE‘ operations can usually be rolled back.

(B) TRUNCATE: This is a DDL command (though some consider it DML due to its effect on data).
‘TRUNCATE TABLE table_name;‘ quickly removes all rows from a table by deallocating the data pages, but it leaves the table’s structure (schema) intact. It is much faster than ‘DELETE‘ for removing all rows, uses fewer system resources, and often cannot be rolled back.

(C) REMOVE: ‘REMOVE‘ is not a standard SQL command for removing tables or rows.

Therefore, ‘DROP TABLE‘ is the correct SQL command that completely removes a table along with its structure (schema) and all its data.

Final Answer : “DROP”

Answer: (D)

Answer Key

Q	Ans	Q	Ans	Q	Ans	Q	Ans	Q	Ans
1	C	2	C	3	B	4	A	5	B
6	B	7	C	8	C	9	B	10	C
11	C	12	A	13	B	14	B	15	A
16	B	17	A	18	B	19	B	20	C
21	B	22	B	23	C	24	B	25	C
26	C	27	B	28	C	29	C	30	C
31	B	32	B	33	C	34	C	35	B
36	B	37	B	38	B	39	B	40	C
41	B	42	C	43	B	44	C	45	A
46	C	47	C	48	B	49	C	50	D