CUET-UG Information Practices Sample Paper - 10

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. Which of the following SQL functions will return the position of the first occurrence of the substring 'ata' in the string 'Informatics'?

(A) SELECT INSTR('Informatics', 'ata');

(B) SELECT SUBSTR('Informatics', 'ata');

(C) SELECT LOCATE('Informatics', 'ata');

(D) SELECT POSITION('ata' IN 'Informatics');

Q2. What will be the output of the following SQL command?

```
SELECT ROUND(157.48, -2);
```

(A) 157

(B) 160

(C) 200

(D) 157.5

Q3. A database administrator wants to display the names of all employees in uppercase who joined in the month of 'March'. Which query is correct?

(A) SELECT UPPER(Ename) FROM Emp WHERE MONTHNAME(DOJ) = 'March';

(B) SELECT UPPER(Ename) FROM Emp WHERE MONTH(DOJ) = 'March';

(C) SELECT UPPER(Ename) FROM Emp WHERE DATE(DOJ) = 'March';

(D) SELECT UPPER(Ename) FROM Emp WHERE DAY(DOJ) = 'March';

Q4. Consider the string S = 'Global Warming'. What is the output of `SELECT MID(S, 8, 4);`?

(A) 'Warm'

(B) 'al W'

(C) 'l Wa'

(D) 'armi'

Q5. What is the result of the SQL expression: `SELECT MOD(15, 4) + POWER(3, 2);`

(A) 11

(B) 12

(C) 13

(D) 7

Q6. Which function is used to remove only leading and trailing spaces from a string in SQL?

(A) LTRIM()

(B) RTRIM()

(C) TRIM()

(D) STRIP()

Q7. To display the current system date and time in MySQL, we use:

(A) CURDATE()

(B) SYSDATE()

(C) DATE()

(D) NOW()

Q8. The SQL function used to find the length of a string 'Informatics' is:

(A) LENGTH('Informatics')

(B) LEN('Informatics')

(C) COUNT('Informatics')

(D) SIZE('Informatics')

Q9. In a Relational Database, if a table 'Student' has 15 rows and 5 columns, and table 'Course' has 4 rows and 3 columns, what is the Cardinality of the Cartesian Product of these two tables?

(A) 60

(B) 19

(C) 8

(D) 45

Q10. Which relational algebra operation is used to select a subset of columns from a table?

(A) Selection (σ)

(B) Projection (π)

(C) Union ($\cup$)

(D) Join ($\bowtie$)

Q11. A candidate key that is not selected as the primary key is known as a/an _____.

(A) Foreign Key

(B) Composite Key

(C) Alternate Key

(D) Secondary Key

Q12. In a table 'Sales', which constraint ensures that the 'Price' column cannot be left empty?

(A) UNIQUE

(B) CHECK

(C) NOT NULL

(D) DEFAULT

Q13. Which network topology requires a central controller or Hub to function?

(A) Mesh

(B) Star

(C) Bus

(D) Ring

Q14. Which networking device is known as an 'intelligent' device that sends data packets only to the intended destination port based on MAC addresses?

(A) Hub

(B) Repeater

(C) Switch

(D) Gateway

Q15. Identify the correct statement regarding MAC and IP addresses:

(A) MAC address is 32-bit, IP address is 48-bit.

(B) MAC address is logical, IP address is physical.

(C) MAC address is assigned by the manufacturer, IP address is assigned by the ISP.

(D) MAC address changes when we change networks, IP address remains constant.

Q16. If 'S1' is a Series with indices $[10, 20, 30, 40]$ and values $[1, 2, 3, 4]$, what will `S1[1:3]` return?

(A) Indices 10, 20

(B) Indices 20, 30

(C) Indices 10, 20, 30

(D) Values 1, 2

Q17. Which attribute is used to check if a Pandas DataFrame is empty?

(A) df.isnull()

(B) df.empty

(C) df.size == 0

(D) df.void

Q18. Given a DataFrame 'df', which command will display the last 3 rows?

(A) df.head(-3)

(B) df.tail(3)

(C) df.last(3)

(D) df.end(3)

Q19. What is the output of the following code?

```
import pandas as pd
S = pd.Series([10, 20, 30], index=['a', 'b', 'c'])
print(S * 2)
```

(A) 20, 40, 60 with indices a, b, c

(B) 10, 20, 30, 10, 20, 30 with indices a, b, c, a, b, c

(C) Error

(D) 100, 400, 900

Q20. To delete a column named 'Salary' from a DataFrame 'df', we use:

(A) df.drop('Salary', axis=0)

(B) df.drop('Salary', axis=1)

(C) df.remove('Salary')

(D) df.delete('Salary')

Q21. Which Pandas function is used to load data from a CSV file into a DataFrame?

(A) pd.load_csv()

(B) pd.open_csv()

(C) pd.read_csv()

(D) pd.get_csv()

Q22. What does the `df.shape` attribute return for a DataFrame with 5 rows and 4 columns?

(A) 20

(B) (4, 5)

(C) (5, 4)

5, 4

Q23. In a DataFrame, `df.iloc[0:2, 1:3]` will select:

(A) Rows 0, 1 and Columns 1, 2

(B) Rows 1, 2 and Columns 2, 3

(C) Rows 0, 1, 2 and Columns 1, 2, 3

(D) Rows 1, 2 and Columns 1, 2

Q24. To rename a column 'Old' to 'New' in DataFrame 'df', use:

(A) df.rename(columns={'Old':'New'})

(B) df.rename('Old'='New')

(C) df.change('Old', 'New')

(D) df.columns['Old'] = 'New'

Q25. Which attribute returns the number of elements in a Series?

(A) size

(B) length

(C) count

(D) items

Q26. The method `df.info()` is used for:

(A) Displaying the first 5 rows

(B) Displaying statistical summary

(C) Displaying metadata and data types

(D) Displaying only the column names

Q27. How can we add a new row to a DataFrame 'df' using a label 'R1'?

(A) df.loc['R1'] = [values]

(B) df.iloc['R1'] = [values]

(C) df.add_row('R1', [values])

(D) df.append('R1', [values])

Q28. Which of the following is used to handle missing data (NaN) in Pandas?

(A) df.dropna()

(B) df.fillna()

(C) Both (A) and (B)

(D) None of the above

Q29. What will be the output of `pd.Series([1, 2], index=['x', 'y']) + pd.Series([20], index=['y', 'z'])`?

(A) x: 1, y: 12, z: 20

(B) x: NaN, y: 12, z: NaN

(C) x: 11, y: 22

(D) Error

Q30. To export a DataFrame 'df' to a file named 'data.csv' without row indices:

(A) df.to_csv('data.csv', index=False)

(B) df.save_csv('data.csv', index=False)

(C) df.write_csv('data.csv', index=None)

(D) pd.to_csv(df, 'data.csv')

Q31. Which parameter in `read_csv()` is used to specify a custom column separator like a semicolon?

(A) delimiter

(B) sep

(C) Both (A) and (B)

(D) separator

Q32. What is the default index type in a Pandas Series if no index is provided?

(A) String labels

(B) RangeIndex starting from 0

(C) RangeIndex starting from 1

(D) Float values

Q33. Which clause is used with aggregate functions to filter groups in SQL?

(A) WHERE

(B) GROUP BY

(C) HAVING

(D) ORDER BY

Q34. To sort the data in descending order based on 'Salary' and then ascending based on 'Name', the query is:

(A) ORDER BY Salary DESC, Name ASC

(B) ORDER BY Salary, Name DESC

(C) SORT BY Salary DESC, Name

(D) ORDER BY Salary DESC AND Name ASC

Q35. Which aggregate function ignores NULL values?

(A) COUNT(*)

(B) COUNT(column_name)

(C) Both ignore NULLs

(D) Neither ignore NULLs

Q36. A join that returns all rows from the left table even if there are no matches in the right table is:

(A) INNER JOIN

(B) RIGHT JOIN

(C) LEFT JOIN

(D) FULL JOIN

Q37. Which command is used to add a new column 'Age' to an existing table 'Student'?

(A) UPDATE Student ADD Age INT;

(B) ALTER TABLE Student ADD Age INT;

(C) MODIFY TABLE Student ADD Age INT;

(D) INSERT INTO Student (Age) VALUES (INT);

Q38. To find the number of unique departments in the 'Emp' table:

(A) SELECT COUNT(Dept) FROM Emp;

(B) SELECT COUNT(DISTINCT Dept) FROM Emp;

(C) SELECT DISTINCT(COUNT Dept) FROM Emp;

(D) SELECT UNIQUE COUNT(Dept) FROM Emp;

Q39. The result of a Join operation is often called:

(A) Cartesian Product

(B) Intersection

(C) Result Set

(D) Equi-Join

Q40. To fetch records where 'City' is 'Delhi', 'Mumbai', or 'Chennai', which operator is most efficient?

(A) LIKE

(B) BETWEEN

(C) IN

(D) OR

Q41. In SQL, `GROUP BY` is applied _____ the `WHERE` clause and _____ the `HAVING` clause.

(A) Before, After

(B) After, Before

(C) After, After

(D) Before, Before

Q42. Which function is used to add a title to a plot in Matplotlib?

(A) plt.label()

(B) plt.header()

(C) plt.title()

(D) plt.heading()

Q43. To display a horizontal bar chart, which function is used?

(A) plt.bar()

(B) plt.barh()

(C) plt.hbar()

(D) plt.plot(kind='barh')

Q44. The parameter used to change the color of the bars in a histogram is:

(A) col

(B) colors

(C) color

(D) palette

Q45. Which command is used to display the legend on the plot?

(A) plt.show_legend()

(B) plt.legend()

(C) plt.label(legend=True)

(D) plt.display()

Q46. To save a plot as an image file named 'graph.png':

(A) plt.save('graph.png')

(B) plt.export('graph.png')

(C) plt.savefig('graph.png')

(D) plt.print('graph.png')

Q47. The legal right given to the creator of a software or literary work is called:

(A) Patent

(B) Trademark

(C) Copyright

(D) Industrial Design

Q48. Which of the following refers to the trail of data you leave behind while using the internet?

(A) Digital Evidence

(B) Digital Footprint

(C) Cyber Trail

(D) Browser History

Q49. Bullying, harassing, or stalking someone using electronic means is called:

(A) Phishing

(B) Cyberstalking

(C) Hacking

(D) Spamming

Q50. The process of recycling or disposing of old electronic devices like broken laptops and phones is known as _____ management.

(A) Solid Waste

(B) Hazardous Waste

(C) E-waste

(D) Bio-waste

Detailed Solutions

Q1.

Solution

Concept: SQL String Functions - INSTR()

The `INSTR()` function is used in MySQL to find the position of the first occurrence of a substring within a larger string. It is a 1-indexed function, meaning the first character is at position 1. If the substring is not found, the function returns 0. It is often used for data validation or string parsing within a database query.

Solution:

Step 1: Analyze the input strings. The primary string is 'Informatics' and the substring we are searching for is 'ata'.

Step 2: Examine the options provided. - `INSTR('Informatics', 'ata')` is the standard syntax to find a position. - `SUBSTR` is used for extracting characters, not finding indices. - `LOCATE` is similar but usually takes parameters in the order `LOCATE(substring, string)`. - `POSITION` uses the `IN` keyword.

Step 3: Logical check. In the string 'Informatics', the sequence 'ata' does not exist (the string contains 'mati'). Therefore, the function will return 0. However, the question asks "which function will return the position," identifying the tool rather than the specific numeric result.

Step 4: Conclusion. Among the choices, `INSTR` is the primary function designated for this task in the CUET-UG IP syllabus.

Final Answer: The function to find the position is INSTR.

Answer: (A)

Q2.

Solution

Concept: SQL Math Functions - ROUND()

The `ROUND(number, decimals)` function is used to round a numeric value. A positive `decimals` value rounds to the right of the decimal point, while a negative `decimals` value rounds to the left (tens, hundreds, etc.). This is a common point of confusion in exams, making it a "Hard" difficulty logic.

Solution:

Step 1: Identify the parameters. The number is 157.48 and the rounding factor is -2.

Step 2: Interpret the negative index. -1 rounds to the nearest ten, and -2 rounds to the nearest hundred.

Step 3: Apply rounding rules to the hundreds place. - The value is 157.48. - Look at the digit to the right of the target place (the tens digit, which is 5). - Since the digit is 5 or greater, we round up the hundreds digit.

Step 4: Calculate the result. The 1 in the hundreds place becomes 2, and all subsequent digits become 0. The result is 200.

Final Answer: The output of the command is 200.

Answer: (C)

Q3.

Solution

Concept: SQL Date Functions and Case Manipulation

This question combines two concepts: string manipulation and date extraction. To solve this, one must know how to nested-functions or use specific date functions like `MONTHNAME()`, which returns the name of the month as a string, unlike `MONTH()` which returns an integer.

Solution:

Step 1: Identify the goal. We need the names in uppercase, requiring the `UPPER()` function.

Step 2: Identify the filtering criteria. The filter is the month 'March'.

Step 3: Analyze function outputs. - `MONTH(DOJ)` would return 3, so `MONTH(DOJ) = 'March'` would fail due to type mismatch or incorrect value comparison. - `MONTHNAME(DOJ)` returns 'March' exactly as a string.

Step 4: Assemble the query. `SELECT UPPER(Ename) FROM Emp WHERE MONTHNAME(DOJ) = 'March';` correctly performs the transformation on the selection and applies the correct logical filter.

Final Answer: Option A is the correct query.

Answer: (A)

Q4.

Solution

Concept: SQL String Functions - MID()

The `MID(str, pos, len)` function is used to extract a specific number of characters (`len`) from a string (`str`) starting at a specific position (`pos`). It is synonymous with `SUBSTR()` and `SUBSTRING()`.

Solution:

Step 1: Break down the string S = 'Global Warming'. - 1:G, 2:l, 3:o, 4:b, 5:a, 6:l, 7:(space), 8:W, 9:a, 10:r, 11:m, 12:i, 13:n, 14:g.

Step 2: Identify the starting point. The parameter is 8, which is the letter 'W'.

Step 3: Determine the number of characters to extract. The length parameter is 4.

Step 4: Extract the characters starting from index 8: - Index 8: W - Index 9: a - Index 10: r - Index 11: m

Step 5: Combine the characters. The result is 'Warm'.

Final Answer: The output of the command is 'Warm'.

Answer: (A)

Q5.

Solution

Concept: SQL Arithmetic Operators and Functions

This question tests the ability to combine different mathematical functions within a single `SELECT` statement. It requires knowledge of the modulo operator (`MOD`) and the exponentiation function (`POWER`).

Solution:

Step 1: Solve the `MOD(15, 4)` part. - Divide 15 by 4. - $15 = (4 \times 3) + 3$. - The remainder is 3. So, `MOD(15, 4)` = 3.

Step 2: Solve the `POWER(3, 2)` part. - This represents 3 raised to the power of 2 (3^2). - $3 \times 3 = 9$. So, `POWER(3, 2)` = 9.

Step 3: Add the results together. - 3 + 9 = 12.

Step 4: Final verification. The SQL engine processes functions first and then the addition operator according to operator precedence.

Final Answer: The result of the expression is 12.

Answer: (B)

Q6.

Solution

Concept: SQL String Functions - Space Removal

In SQL, string data often contains unwanted spaces due to data entry errors. To handle these, MySQL provides three distinct functions: `LTRIM()` for the left side, `RTRIM()` for the right side, and `TRIM()` for both. Understanding the difference is crucial for data cleaning tasks in informatics.

Solution:

Step 1: Define the goal. The objective is to remove "leading" (start) and "trailing" (end) spaces simultaneously.

Step 2: Evaluate the options. - `LTRIM()` (Left Trim): Only removes spaces from the beginning of the string. - `RTRIM()` (Right Trim): Only removes spaces from the end of the string. - `TRIM()`: Removes spaces from both ends of the string. - `STRIP()`: This is a Python method, not a standard SQL function.

Step 3: Logical Application. If we have the string `' Data '`, `TRIM()` will result in `'Data'`.

Step 4: Conclusion. The `TRIM()` function is the correct choice for removing spaces from both ends.

Final Answer: The function used is TRIM().

Answer: (C)

Q7.

Solution

Concept: SQL Date Functions - System Time

Retrieving the current date and time is a fundamental operation for logging events or calculating durations in a database. MySQL provides several functions that return the current system timestamp.

Solution:

Step 1: Analyze the requirement. We need both the "current system date" AND "time."

Step 2: Compare similar functions. - `CURDATE()`: Returns only the date (YYYY-MM-DD). - `CURTIME()`: Returns only the time (HH:MM:SS). - `NOW()`: Returns the current date and time (YYYY-MM-DD HH:MM:SS). - `SYSDATE()`: Similar to `NOW()`, it returns the date and time, but specifically the time at which the function executes.

Step 3: Identify the most common standard answer. In the CUET/CBSE curriculum, `NOW()` is the primary function taught for retrieving the combined date and time.

Step 4: Conclusion. `NOW()` is the standard function for this purpose.

Final Answer: The function is NOW().

Answer: (D)

Q8.

Solution

Concept: SQL String Functions - LENGTH()

The `LENGTH()` function in MySQL is used to find the length of a string. In the context of standard ASCII characters (like 'Informatics'), it returns the count of characters in the string.

Solution:

Step 1: Identify the tool required. We need a function that counts the number of characters in a given string.

Step 2: Distinguish between SQL and other languages. - `LEN()` is used in SQL Server (T-SQL) but not in standard MySQL. - `COUNT()` is an aggregate function used for counting rows, not characters in a string. - `SIZE()` is used in some programming languages but not for strings in SQL.

Step 3: Apply the MySQL specific function. `LENGTH('Informatics')` will count the letters: I, n, f, o, r, m, a, t, i, c, s.

Step 4: Conclusion. `LENGTH()` is the correct syntax for MySQL/Informatics Practices.

Final Answer: The function is LENGTH().

Answer: (A)

Q9.

Solution

Concept: Database Concepts - Cartesian Product

The Cartesian Product (also known as a Cross Join) is a fundamental operation in relational algebra. It combines every row of the first table with every row of the second table. The two main properties to remember are: 1. **Cardinality** (Rows) is the product: $C(A \times B) = C(A) \times C(B)$. 2. **Degree** (Columns) is the sum: $D(A \times B) = D(A) + D(B)$.

Solution:

Step 1: Identify the Cardinality of each table. - Table 'Student' has 15 rows. - Table 'Course' has 4 rows.

Step 2: Apply the formula for Cardinality of a Cartesian Product. - Resulting Cardinality = Rows in Student $\times$ Rows in Course.

Step 3: Perform the calculation. - $15 \times 4 = 60$.

Step 4: Clarification. Even though columns are provided (5 and 3), they are only relevant if the question asks for the "Degree" ($5 + 3 = 8$). Since the question asks for Cardinality, we stick to the product of rows.

Final Answer: The Cardinality is 60.

Answer: (A)

Q10.

Solution

Concept: Relational Algebra - Projection (π)

Relational algebra is a formal language for the relational model. Two of the most important operations are Selection and Projection. - **Selection (σ)**: A horizontal operation that retrieves rows based on a condition. - **Projection (π)**: A vertical operation that retrieves specific columns from a relation.

Solution:

Step 1: Identify the requirement of the question. It asks for the operation used to "select a subset of columns."

Step 2: Match the requirement to the operation. - Selecting columns = Vertical operation = Projection.

Step 3: Identify the symbol. The Greek letter Pi (π) is used to denote Projection.

Step 4: Contrast with other options. - Selection (σ) is for rows. - Union ($\cup$) is for combining results from two tables. - Join ($\bowtie$) is for combining rows from two tables based on a related column.

Final Answer: The operation is Projection (π).

Answer: (B)

Q11.

Solution

Concept: Database Keys - Alternate Key

In a database table, there can be multiple columns or sets of columns that can uniquely identify a record. These are all called **Candidate Keys**. One of these is selected to be the **Primary Key**. Any Candidate Key that is not chosen as the Primary Key is officially designated as an **Alternate Key**.

Solution:

Step 1: Define the hierarchy of keys. Every Primary Key is a Candidate Key, but not every Candidate Key is a Primary Key.

Step 2: Identify the specific scenario. The question describes a key that is a candidate (uniquely identifies rows) but was "not selected" for the primary role.

Step 3: Map to terminology. - **Foreign Key:** Used to link two tables. - **Composite Key:** A key made of more than one column. - **Alternate Key:** The specific term for "spare" candidate keys.

Step 4: Conclusion. The correct term for a non-primary candidate key is an Alternate Key.

Final Answer: The key is known as an Alternate Key.

Answer: (C)

Q12.

Solution

Concept: SQL Constraints - NOT NULL

Constraints are rules applied to table columns to ensure data integrity. The **NOT NULL** constraint specifically prevents a column from accepting NULL (empty) values. This ensures that every record must have a valid data point for that specific attribute.

Solution:

Step 1: Analyze the requirement. We need to ensure a column "cannot be left empty."

Step 2: Evaluate SQL constraints. - `UNIQUE`: Prevents duplicate values but may allow one NULL. - `CHECK`: Validates values against a specific condition (e.g., `Price > 0`). - `NOT NULL`: Explicitly forbids the absence of a value. - `DEFAULT`: Provides a value if none is supplied, but doesn't necessarily block NULLs if they are explicitly sent.

Step 3: Conclusion. The `NOT NULL` constraint is the standard mechanism to mandate data entry for a field.

Final Answer: The constraint is NOT NULL.

Answer: (C)

Q13.

Solution

Concept: Network Topologies - Star Topology

Network topology refers to the physical or logical arrangement of computers in a network. In a **Star Topology**, every peripheral node is connected to a central node called a Hub, Switch, or Router. All data traffic passes through this central device, which acts as a signal repeater or director.

Solution:

Step 1: Identify the unique identifier in the question: "central controller or Hub."

Step 2: Compare topologies. - **Bus:** Uses a single main cable (backbone); no central hub. - **Ring:** Each node connects to exactly two other nodes; no central hub. - **Mesh:** Every node is interconnected; no central hub. - **Star:** All nodes radiate from a central point.

Step 3: Logical conclusion. Since all cables lead to a single central device, this configuration is a Star.

Final Answer: The topology is Star.

Answer: (B)

Q14.

Solution

Concept: Networking Devices - Switch

Networking devices vary in ”intelligence.” A **Hub** is a ”dumb” device that broadcasts data to every port. A **Switch** is considered ”intelligent” because it learns the MAC addresses of connected devices and sends data frames only to the specific port intended for the recipient, reducing network traffic and collisions.

Solution:

Step 1: Identify the keywords: ”intelligent,” ”intended destination port,” and ”MAC addresses.”

Step 2: Evaluate devices. - **Repeater:** Simply regenerates a signal to cover longer distances. - **Hub:** Sends data to everyone (broadcast), not just the destination. - **Switch:** Uses a MAC address table to filter and forward data to specific ports. - **Gateway:** Connects two dissimilar networks.

Step 3: Conclusion. The device that performs selective forwarding based on hardware (MAC) addresses is the Switch.

Final Answer: The device is a Switch.

Answer: (C)

Q15.

Solution

Concept: Networking - MAC vs IP Addresses

Understanding the difference between physical (MAC) and logical (IP) addressing is critical in networking. - **MAC (Media Access Control):** A unique, permanent hardware address assigned by the manufacturer (NIC). - **IP (Internet Protocol):** A logical address assigned by the network administrator or ISP, which can change depending on the network location.

Solution:

Step 1: Evaluate Statement A. MAC is 48-bit; IP (v4) is 32-bit. Statement A is inverted.

Step 2: Evaluate Statement B. MAC is physical; IP is logical. Statement B is inverted.

Step 3: Evaluate Statement C. MAC is hardcoded by the manufacturer; IP is assigned by the ISP or network. This is correct.

Step 4: Evaluate Statement D. MAC is fixed for the hardware; IP changes when you move to a different Wi-Fi/network. Statement D is inverted.

Final Answer: Option (C) is the correct statement.

Answer: (C)

Q16.

Solution

Concept: Pandas Series - Slicing and Indexing

In Pandas, a Series can be sliced using positional indices. When using the syntax `Series[start:stop]`, Pandas follows the Python slicing convention: the `start` index is inclusive, while the `stop` index is exclusive. Even if custom labels (like 10, 20) are used as indices, positional slicing still refers to the integer positions 0, 1, 2, ...

Solution:

Step 1: Identify the structure of the Series `S1`.

- Position 0: Index 10, Value 1
- Position 1: Index 20, Value 2
- Position 2: Index 30, Value 3
- Position 3: Index 40, Value 4

Step 2: Apply the slice `[1:3]`.

- Start at position 1 (inclusive).
- Stop at position 3 (exclusive).

Step 3: Extract the corresponding data. This selects positions 1 and 2. - Position 1 has index label 20. - Position 2 has index label 30.

Step 4: Conclusion. The slice returns a new Series containing the index labels 20 and 30.

Final Answer: The slice returns Indices 20, 30.

Answer: (B)

Q17.

Solution

Concept: Pandas DataFrame Attributes - .empty

Pandas objects (Series and DataFrames) have several built-in attributes that provide information about the data structure. The `.empty` attribute is a boolean attribute that returns `True` if the object is empty (contains no elements), and `False` otherwise.

Solution:

Step 1: Review the purpose of the attributes mentioned. - `df.isnull()`: Returns a DataFrame of the same shape with True/False for each individual cell if it is missing data. - `df.empty`: Returns a single boolean value indicating if the entire DataFrame has zero elements. - `df.size == 0`: This logic works, but `size` is an attribute that returns the total number of elements. Checking if it equals 0 is a comparison, not a direct attribute for emptiness.

Step 2: Identify the most direct and efficient attribute. `df.empty` is the standard Pandas property used for this check.

Step 3: Note that a DataFrame with only `NaN` values is not considered empty; it must have zero rows or zero columns to be ”empty.”

Final Answer: The attribute is df.empty.

Answer: (B)

Q18.

Solution

Concept: Pandas DataFrame Methods - tail()

The `.head(n)` and `.tail(n)` methods are used to quickly inspect the beginning or the end of a DataFrame. By default, both return 5 rows, but this can be customized by passing an integer `n`.

Solution:

Step 1: Determine the requirement. We need to display the ”last 3 rows.”

Step 2: Analyze the methods. - `head(n)`: Returns the first n rows. - `tail(n)`: Returns the last n rows.

Step 3: Apply the parameter. To get exactly 3 rows from the bottom, we use `df.tail(3)`.

Step 4: Evaluate other options. `df.last()` is not a standard Pandas method for general row retrieval, and `df.end()` does not exist in the Pandas library.

Final Answer: The command is df.tail(3).

Answer: (B)

Q19.

Solution

Concept: Pandas Series - Vectorized Operations

Pandas allows for "vectorized operations," meaning you can perform arithmetic directly on a Series or DataFrame without writing loops. When you multiply a Series by a scalar (like a number), the operation is applied to every single element in the Series.

Solution:

Step 1: Examine the initial Series `S`. - Index 'a': 10 - Index 'b': 20 - Index 'c': 30

Step 2: Perform the multiplication operation `S * 2`. - $10 \times 2 = 20$ - $20 \times 2 = 40$ - $30 \times 2 = 60$

Step 3: Observe the indices. Vectorized operations preserve the original indices of the Series.

- 'a' becomes 20
- 'b' becomes 40
- 'c' becomes 60

Step 4: Contrast with Python lists. If this were a list `[10, 20, 30] * 2`, it would result in replication: `[10, 20, 30, 10, 20, 30]`. In Pandas, it performs mathematical multiplication.

Final Answer: The output is 20, 40, 60 with indices a, b, c.

Answer: (A)

Q20.

Solution

Concept: Pandas DataFrame - Dropping Data

The `.drop()` method is used to remove rows or columns from a DataFrame. To specify whether you are removing a row or a column, you must use the `axis` parameter. - `axis=0` or `axis='index'`: Removes rows. - `axis=1` or `axis='columns'`: Removes columns.

Solution:

Step 1: Identify the target to be deleted. The question specifies a "column named 'Salary'."

Step 2: Choose the correct method. `df.drop()` is the correct function.

Step 3: Set the axis. Since we are removing a column, we must set `axis=1`.

Step 4: Analyze other options. `df.remove()` and `df.delete()` are not standard Pandas methods for this purpose. `df.drop('Salary', axis=0)` would attempt to find a row labeled 'Salary', which would likely cause an error if it doesn't exist.

Final Answer: The command is df.drop('Salary', axis=1).

Answer: (B)

Q21.

Solution

Concept: Pandas Data Input - CSV Files

CSV (Comma Separated Values) is the most common format for data exchange in data science. Pandas provides a highly optimized function to parse these files directly into a DataFrame object.

Solution:

Step 1: Identify the standard naming convention in Pandas for input/output. Most input functions start with `read_` and output functions start with `to_`.

Step 2: Evaluate the options. - `load_csv()` and `open_csv()` are not actual Pandas functions. - `read_csv()` is the universal function used to load text-based tabular data. - `get_csv()` is incorrect.

Step 3: Technical Detail. `read_csv()` automatically detects the header and assigns appropriate data types to columns unless specified otherwise.

Final Answer: The function is pd.read_csv().

Answer: (C)

Q22.

Solution

Concept: Pandas DataFrame Attributes - .shape

The `.shape` attribute is used to find the dimensions of a DataFrame. It returns a Python tuple representing the dimensionality of the object.

Solution:

Step 1: Recall what the `shape` attribute returns. It always returns a tuple in the format `(rows, columns)`.

Step 2: Apply the given dimensions. - Rows = 5 - Columns = 4

Step 3: Format as a tuple: `(5, 4)`.

Step 4: Contrast with other attributes. `df.size` would return the total number of elements (5×4 = 20). `df.ndim` would return 2 (since a DataFrame is two-dimensional).

Final Answer: The attribute returns (5, 4).

Answer: (C)

Q23.

Solution

Concept: Pandas Slicing - .iloc

The `.iloc` indexer is used for integer-location based indexing and slicing. It follows the standard Python slicing rule: the start index is inclusive and the end index is exclusive.

Solution:

Step 1: Break down the slice `df.iloc[0:2, 1:3]`. The first part refers to rows, and the second part refers to columns.

Step 2: Analyze row slice `0:2`. - Start at index 0. - Stop before index 2. - Result: Rows 0 and 1.

Step 3: Analyze column slice `1:3`. - Start at index 1. - Stop before index 3. - Result: Columns 1 and 2.

Step 4: Combine the selection. The operation selects the intersection of Rows 0, 1 and Columns 1, 2.

Final Answer: Rows 0, 1 and Columns 1, 2 are selected.

Answer: (A)

Q24.

Solution

Concept: Pandas DataFrame - Renaming Columns

To change the labels of existing columns, the `.rename()` method is used. This method requires a dictionary-like object mapping the old names to the new names.

Solution:

Step 1: Identify the correct syntax for `rename()`. It uses the `columns` parameter.

Step 2: Evaluate the dictionary structure: `{'Old_Name' : 'New_Name'}`.

Step 3: Check the options. - `df.rename(columns={'Old':'New'})` is the standard, correct syntax. - `df.rename('Old'='New')` is syntactically incorrect in Python. - `df.change()` is not a Pandas method.

Step 4: Conclusion. To modify the labels specifically, the dictionary mapping within the `columns` argument is the proper approach.

Final Answer: Option (A) is the correct command.

Answer: (A)

Q25.

Solution

Concept: Pandas Series Attributes - .size

Series attributes provide quick information about the object. While `len()` is a Python function that works on Series, Pandas provides the `.size` attribute for consistency across its objects.

Solution:

Step 1: Define the purpose of `.size`. It returns the total number of elements in the Series (including NaN values).

Step 2: Compare with `count()`. - `size` returns the total length of the index. - `count()` is a method that returns only the number of non-null (non-NaN) values.

Step 3: Evaluate the attribute names. `length` and `items` are not the standard attributes for total element count in this context.

Step 4: Conclusion. The `.size` attribute is the correct property for the total number of elements.

Final Answer: The attribute is size.

Answer: (A)

Q26.

Solution

Concept: Pandas DataFrame Methods - .info()

The `.info()` method is a diagnostic tool used to get a concise summary of a DataFrame. It is essential for understanding the structure of the data before performing analysis, particularly when dealing with large datasets imported from external files.

Solution:

Step 1: Analyze what `df.info()` provides. It displays:

- The class of the object (DataFrame).
- The index range and number of entries.
- Data columns (names and counts).
- Non-null counts for each column.
- Data types (dtypes) of each column.
- Memory usage.

Step 2: Evaluate the options. - It does not just show rows (that's `head()`). - It does not show statistical summaries (that's `describe()`). - It provides "metadata" (data about the data).

Step 3: Conclusion. The correct choice is the display of metadata and data types.

Final Answer: The method is used for displaying metadata and data types.

Answer: (C)

Q27.

Solution

Concept: Pandas DataFrame - Adding Data

Adding a new row to a DataFrame can be done using the `.loc[]` indexer. While `.iloc[]` is for integer positions, `.loc[]` is used for label-based indexing. If the label provided does not exist, Pandas will create a new row with that label.

Solution:

Step 1: Identify the requirement: "add a new row" with a specific label 'R1'.

Step 2: Evaluate the syntax. - `df.loc['R1'] = [values]` is the standard way to assign values to a row label. - `df.iloc` would require an existing integer index and cannot easily append a new labeled row in this manner. - `df.add_row` is not a valid Pandas method.

Step 3: Conclusion. Using `.loc` with a new label effectively appends a new row to the end of the DataFrame.

Final Answer: The correct syntax is df.loc['R1'] = [values].

Answer: (A)

Q28.

Solution

Concept: Handling Missing Data in Pandas

Missing data, represented as `NaN` (Not a Number), is common in real-world datasets. Pandas provides two primary ways to manage this: removing the data or filling it with replacement values.

Solution:

Step 1: Analyze the functions. - `df.dropna()`: This method removes any row or column that contains missing values. - `df.fillna()`: This method replaces `NaN` values with a specific value (like 0 or the mean) or a method (like forward fill).

Step 2: Determine the scope of the question. Since both functions are explicitly designed to "handle missing data," both are correct.

Step 3: Conclusion. The most comprehensive answer is "Both (A) and (B)."

Final Answer: Both (A) and (B) are used.

Answer: (C)

Q29.

Solution

Concept: Pandas Series - Data Alignment

Data alignment is one of the most powerful features of Pandas. When performing arithmetic between two Series, Pandas automatically aligns the data based on the index labels. If a label exists in one Series but not the other, the result for that label will be `NaN`.

Solution:

Step 1: Identify the indices and values for the first Series: `x: 1, y: 2`.

Step 2: Identify the indices and values for the second Series: `y: 10, z: 20`.

Step 3: Perform the addition index by index. - Label `x`: Exists in S1 (1) but not S2. Result: `NaN`. - Label `y`: Exists in both (2 + 10). Result: `12`. - Label `z`: Exists in S2 (20) but not S1. Result: `NaN`.

Step 4: Combine the results: `x: NaN, y: 12, z: NaN`.

Final Answer: The output is x: NaN, y: 12, z: NaN.

Answer: (B)

Q30.

Solution

Concept: Pandas Data Output - Exporting to CSV

The `.to_csv()` method is used to save a DataFrame to a text file. By default, Pandas includes the DataFrame's index as the first column in the file.

Solution:

Step 1: Identify the method for CSV export. It is `df.to_csv()`.

Step 2: Address the "without row indices" requirement. This is controlled by the `index` parameter.

Step 3: Set the parameter. To exclude indices, use `index=False`.

Step 4: Evaluate the options. - Option (A) `df.to_csv('data.csv', index=False)` is correct. - Option (B) `save_csv` is not a valid method. - Option (C) `index=None` is often used as a synonym for False in some contexts, but `index=False` is the canonical Pandas syntax.

Final Answer: The correct command is df.to_csv('data.csv', index=False).

Answer: (A)

Q31.

Solution

Concept: Pandas Data Input - Custom Delimiters

The `read_csv()` function is highly flexible. While the name implies "comma-separated," it can actually read files with any character serving as a separator (tab, semicolon, pipe, etc.) by specifying the delimiter parameter.

Solution:

Step 1: Identify the parameter that defines the separator. In Pandas, `sep` is the most commonly used shorthand.

Step 2: Evaluate the documentation. The parameter `delimiter` is actually an alias for `sep`. Both can be used interchangeably to define the character that splits the data into columns.

Step 3: Analyze the options. Since both `sep` and `delimiter` are valid, the most accurate choice is "Both (A) and (B)."

Step 4: Example syntax: `pd.read_csv('file.csv', sep=';')` or `pd.read_csv('file.csv', delimiter=';')`.

Final Answer: Both (A) and (B) can be used.

Answer: (C)

Q32.

Solution

Concept: Pandas Series - Default Indexing

When a Series is created in Pandas, every data element must have an associated index label. If the user does not provide a specific list of labels via the `index` parameter, Pandas generates them automatically.

Solution:

Step 1: Identify the default behavior. Pandas uses an integer-based sequence for default indexing.

Step 2: Determine the start value. Following standard Python 0-based indexing, the sequence begins at 0.

Step 3: Define the index type. This specific object is known as a `RangeIndex`. It starts from 0 and goes up to $n - 1$, where n is the number of elements.

Step 4: Eliminate incorrect options. It does not start from 1, nor does it use float values or strings by default.

Final Answer: The default is RangeIndex starting from 0.

Answer: (B)

Q33.

Solution

Concept: Advanced SQL - Filtering with Aggregate Functions

Filtering data in SQL can happen at two levels: at the row level (using `WHERE`) or at the group level (using `HAVING`). Aggregate functions like `SUM()`, `COUNT()`, and `AVG()` cannot be used inside a `WHERE` clause.

Solution:

Step 1: Recall the order of execution in an SQL query. 1. `FROM` (Choose table) 2. `WHERE` (Filter rows) 3. `GROUP BY` (Group rows) 4. `HAVING` (Filter groups) 5. `SELECT` (Pick columns)

Step 2: Identify the specific function of `HAVING`. It was introduced specifically because the `WHERE` keyword cannot be used with aggregate functions.

Step 3: Logical conclusion. To filter groups (e.g., "Show departments where the average salary > 50000"), the `HAVING` clause is the correct tool.

Final Answer: The clause is HAVING.

Answer: (C)

Q34.

Solution

Concept: SQL Sorting - Multiple Columns

The `ORDER BY` clause is used to sort the result set. You can sort by multiple columns by listing them separated by commas. Each column can have its own sort direction: `ASC` (Ascending, default) or `DESC` (Descending).

Solution:

Step 1: Analyze the requirements. - Requirement 1: Sort by 'Salary' in descending order. (`Salary DESC`) - Requirement 2: Sort by 'Name' in ascending order. (`Name ASC` or just `Name`)

Step 2: Combine the requirements using proper SQL syntax. The columns are listed in the order of sorting priority.

Step 3: Evaluate the options. - `ORDER BY Salary DESC, Name ASC` is the standard syntax. - Option (D) is incorrect because it uses the keyword `AND`, which is for logical conditions, not for separating columns in `ORDER BY`.

Step 4: Conclusion. Option (A) correctly implements the multi-level sort.

Final Answer: Option (A) is the correct query.

Answer: (A)

Q35.

Solution

Concept: SQL Aggregate Functions - Handling NULLs

Aggregate functions perform a calculation on a set of values. A key distinction in SQL is how these functions treat `NULL` (unknown) values.

Solution:

Step 1: Understand `COUNT(*)`. This function counts every row in the result set, including rows that contain `NULL` values. It effectively counts the number of records.

Step 2: Understand `COUNT(column_name)`. This function counts only the non-NULL entries in the specified column.

Step 3: Compare other functions. `SUM()`, `AVG()`, `MIN()`, and `MAX()` all ignore `NULL` values by default.

Step 4: Conclusion. Since `COUNT(column_name)` specifically ignores NULLs while `COUNT(*)` does not, `COUNT(column_name)` is the function that ignores NULLs.

Final Answer: The function is COUNT(column_name).

Answer: (B)

Q36.

Solution

Concept: SQL Joins - LEFT JOIN

In relational databases, a Join is used to combine rows from two or more tables based on a related column. A **LEFT JOIN** (or LEFT OUTER JOIN) returns all records from the left table (the first table mentioned), and the matched records from the right table. If there is no match, the result is NULL from the right side.

Solution:

Step 1: Analyze the requirement. We need "all rows from the left table" regardless of matches.

Step 2: Evaluate Join types. - **INNER JOIN:** Returns only rows where there is a match in both tables. - **RIGHT JOIN:** Returns all rows from the right table and matches from the left. - **LEFT JOIN:** Ensures the left table is preserved entirely in the output.

Step 3: Logical Application. This is often used to find "orphaned" records (e.g., Customers who haven't placed an order).

Step 4: Conclusion. The LEFT JOIN is the correct operation for this specific behavior.

Final Answer: The join is LEFT JOIN.

Answer: (C)

Q37.

Solution

Concept: SQL Data Definition Language (DDL) - ALTER TABLE

To modify the structure of an existing table (like adding, deleting, or modifying columns), we use the `ALTER TABLE` command. This is part of DDL, whereas `UPDATE` is part of DML (Data Manipulation Language) used for modifying data within rows.

Solution:

Step 1: Identify the goal. We need to "add a new column" to a table that already exists.

Step 2: Choose the correct command. `ALTER TABLE` is the mandatory starting point for structural changes.

Step 3: Apply the correct syntax. The syntax is:

```
ALTER TABLE table_name ADD column_name datatype;
```

Step 4: Evaluate the options. - Option (A) uses `UPDATE`, which is for row data. - Option (C) uses `MODIFY`, which is for changing the type of an existing column, not adding a new one. - Option (B) uses the correct keyword `ADD`.

Final Answer: The command is ALTER TABLE Student ADD Age INT;

Answer: (B)

Q38.

Solution

Concept: SQL Aggregate Functions - DISTINCT with COUNT

When you want to find the number of categories or unique values in a column, you combine the `COUNT()` aggregate function with the `DISTINCT` keyword. `COUNT(column)` counts all values, but `COUNT(DISTINCT column)` eliminates duplicates before counting.

Solution:

Step 1: Define the requirement: "number of unique departments."

Step 2: Evaluate the count logic. - `SELECT COUNT(Dept)` would return the total number of employees assigned to any department. - `SELECT COUNT(DISTINCT Dept)` counts each department name only once.

Step 3: Check syntax. The `DISTINCT` keyword must be inside the parentheses of the `COUNT()` function to apply to the values being counted.

Step 4: Conclusion. Option (B) correctly implements the logic for unique/distinct counts.

Final Answer: The query is SELECT COUNT(DISTINCT Dept) FROM Emp;

Answer: (B)

Q39.

Solution

Concept: Database Operations - Join Results

When two tables are joined in a database query, the logical process creates a combination of rows based on the join condition. While the mechanical step is a Cartesian product followed by a filter, the output presented to the user is a specific structure.

Solution:

Step 1: Define terms. - A **Cartesian Product** is an unfiltered join (every row to every row). - An **Intersection** is a mathematical set concept.

Step 2: Analyze the output of a query. In SQL terminology, the result of any `SELECT` statement (including Joins) is formally referred to as a **Result Set**.

Step 3: Nuance check. While an "Equi-join" is a *type* of join, it is not the name of the result itself.

Step 4: Conclusion. The most general and technically accurate term for the output is the Result Set.

Final Answer: The result is called a Result Set.

Answer: (C)

Q40.

Solution

Concept: SQL Operators - IN Operator

The `IN` operator allows you to specify multiple values in a `WHERE` clause. It is a shorthand for multiple `OR` conditions and is much more readable and efficient when checking against a discrete list of strings or numbers.

Solution:

Step 1: Analyze the filter criteria. We need to match 'City' against 'Delhi', 'Mumbai', or 'Chennai'.

Step 2: Evaluate operator efficiency. - `LIKE` is for pattern matching (using % or _). - `BETWEEN` is for a continuous range of values. - `IN` is specifically designed for membership in a set/list.

Step 3: Compare with `OR`. `WHERE City='Delhi' OR City='Mumbai' OR City='Chennai'` is valid but less efficient for the parser and more verbose for the programmer.

Step 4: Conclusion. The `IN` operator is the most efficient and standard way to handle this requirement.

Final Answer: The operator is IN.

Answer: (C)

Q41.

Solution

Concept: SQL Query Execution Order

The order in which clauses are written in an SQL statement follows a strict syntax. To filter rows before grouping, we use `WHERE`. To group those rows, we use `GROUP BY`. To filter the resulting groups, we use `HAVING`.

Solution:

Step 1: Identify the syntactical position of `GROUP BY`. It must come after the `WHERE` clause (if present) because rows must be filtered before they can be aggregated.

Step 2: Identify the position relative to `HAVING`. The `HAVING` clause acts on the groups created by `GROUP BY`, so it must come after it.

Step 3: Sequence the clauses. The logical sequence is `WHERE` → `GROUP BY` → `HAVING`.

Step 4: Match the blanks. `GROUP BY` is applied **After** the `WHERE` clause and **Before** the `HAVING` clause.

Final Answer: The sequence is After, Before.

Answer: (B)

Q42.

Solution

Concept: Matplotlib Customization - Title

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. The `pyplot` module provides a simple interface to add textual elements to a chart to make it more informative.

Solution:

Step 1: Identify the goal: adding a ”title” to the plot.

Step 2: Evaluate the `pyplot` functions. - `xlabel()` and `ylabel()` add titles to the axes. - `title()` adds a main heading centered above the plot area.

Step 3: Verify the syntax. The command is `matplotlib.pyplot.title("Your Title Here")`.

Step 4: Eliminate incorrect options. Functions like `header()` or `heading()` are not part of the standard Matplotlib API.

Final Answer: The function is plt.title().

Answer: (C)

Q43.

Solution

Concept: Matplotlib Chart Types - Horizontal Bar Chart

Standard bar charts are vertical by default. However, when category labels are long, a horizontal bar chart is often preferred for readability. Matplotlib provides a specific function for this orientation.

Solution:

Step 1: Identify the target chart type: "horizontal bar chart."

Step 2: Compare function names. - `plt.bar()` creates vertical bars. - `plt.barh()` (where 'h' stands for horizontal) creates horizontal bars.

Step 3: Parameter Check. `barh()` takes `y` and `width` as primary arguments instead of `x` and `height`.

Step 4: Conclusion. The specific, built-in function for this task is `barh()`.

Final Answer: The function is plt.barh().

Answer: (B)

Q44.

Solution

Concept: Matplotlib Plot Customization - Color

Visual aesthetics in charts, such as colors, line styles, and markers, are controlled by keyword arguments within the plotting functions (`bar`, `hist`, `plot`, etc.).

Solution:

Step 1: Determine the parameter name used across most Matplotlib functions for color.

Step 2: Evaluate the choices. - `color` is the standard keyword argument. It can accept name strings (e.g., 'red'), hex codes, or RGB tuples. - `colors` (plural) is used in specific charts like `pie()`, but for a histogram or bar chart, `color` is the primary parameter.

Step 3: Syntax example: `plt.hist(data, color='green')`.

Step 4: Conclusion. The keyword argument is `color`.

Final Answer: The parameter is color.

Answer: (C)

Q45.

Solution

Concept: Matplotlib - Chart Legends

A legend is an area describing the elements of the graph. In Matplotlib, even if you provide "label" arguments to your plots, the legend will not appear automatically until a specific function is called.

Solution:

Step 1: Recall the two-step process for legends. 1. Assign a label in the plot function: `plt.plot(x, y, label='Sales')`. 2. Call the legend function to draw it.

Step 2: Identify the function. The function is `plt.legend()`.

Step 3: Understand its functionality. It automatically looks for any labeled plots and creates a key for them. It can take parameters like `loc` to change its position (e.g., 'upper right').

Step 4: Eliminate incorrect options. `plt.display()` or `show_legend()` are not valid Matplotlib commands.

Final Answer: The command is plt.legend().

Answer: (B)

Q46.

Solution

Concept: Matplotlib - Exporting Visualizations

Visualizations created in Python are often required for reports or presentations. Matplotlib provides the `savefig()` function to export the current figure to various file formats like PNG, JPG, PDF, or SVG.

Solution:

Step 1: Identify the correct method for saving a figure. In Matplotlib, this is `savefig()`.

Step 2: Evaluate the parameters. The first argument is the filename (including the desired extension).

Step 3: Syntax Check. `plt.savefig('graph.png')` is the standard call.

Step 4: Eliminate incorrect options. `plt.save()` or `plt.export()` are not valid functions in the Matplotlib library. It is important to call this function *before* `plt.show()`, as `show()` clears the figure buffer in some environments.

Final Answer: The command is plt.savefig('graph.png').

Answer: (C)

Q47.

Solution

Concept: Intellectual Property Rights (IPR) - Copyright

Intellectual Property Rights are the rights given to persons over the creations of their minds. **Copyright** is a legal term used to describe the rights that creators have over their literary, artistic, and scientific works, including software code.

Solution:

Step 1: Define the nature of the work mentioned: "software or literary work."

Step 2: Match with the appropriate IPR. - **Patent:** Protects inventions or functional processes. - **Trademark:** Protects brand names, logos, and slogans. - **Copyright:** Protects the expression of ideas (books, music, software).

Step 3: Conclusion. Software is protected under the Copyright Act in most jurisdictions, including India.

Final Answer: The legal right is Copyright.

Answer: (C)

Q48.

Solution

Concept: Societal Impacts - Digital Footprint

A **Digital Footprint** is the unique set of traceable digital activities, actions, contributions, and communications manifested on the Internet or digital devices. It includes both "Active" footprints (posts, emails) and "Passive" footprints (browsing history, IP logs).

Solution:

Step 1: Analyze the definition provided: "trail of data you leave behind while using the internet."

Step 2: Identify the specific terminology. This metaphorical "trail" is universally referred to as a Digital Footprint.

Step 3: Contrast with other options. - **Browser History:** Only a local record on your device. - **Cyber Trail:** A generic term but not the formal technical term used in the IP syllabus.

Step 4: Conclusion. The persistent record of your online activity is your Digital Footprint.

Final Answer: This refers to a Digital Footprint.

Answer: (B)

Q49.

Solution

Concept: Cybercrime - Cyberstalking/Bullying

Cybercrime involves criminal activities carried out using computers or the internet. When the crime involves targeted harassment or monitoring of an individual to cause fear or distress, it falls under the category of cyberstalking or cyberbullying.

Solution:

Step 1: Identify the characteristics of the act: ”Bullying, harassing, or stalking” via ”electronic means.”

Step 2: Evaluate the terminology. - **Phishing:** Fraudulent attempts to obtain sensitive information. - **Hacking:** Unauthorized access to systems. - **Cyberstalking:** Repeated use of the internet to harass or frighten a person.

Step 3: Conclusion. The specific act of persistent online harassment is Cyberstalking.

Final Answer: This is called Cyberstalking.

Answer: (B)

Q50.

Solution

Concept: Societal Impacts - E-waste Management

E-waste (Electronic Waste) refers to discarded electrical or electronic devices. Proper management of e-waste is critical for environmental sustainability due to the toxic materials (like lead and mercury) often found in these components.

Solution:

Step 1: Categorize the waste mentioned: ”old electronic devices like broken laptops and phones.”

Step 2: Identify the prefix used for electronics. The letter ’e’ stands for electronic.

Step 3: Evaluate the waste types. - **Solid Waste:** General household trash. - **Bio-waste:** Organic/medical waste. - **E-waste:** Specifically designated for electronic circuitry and hardware.

Step 4: Conclusion. The management of these devices is called E-waste management.

Final Answer: The process is known as E-waste management.

Answer: (C)

Answer Key

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