CUET-UG Information Practices Sample Paper-13

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 can be used to find the remainder when one number is divided by another?

(A) REM()

(B) MOD()

(C) REMAINDER()

(D) DIV()

Q2. The SQL string function used to extract a part of a string starting from a specific position for a specific number of characters is:

(A) LEFT()

(B) SUBSTR()

(C) EXTRACT()

(D) PART()

Q3. What will be the output of the following SQL command?

```
SELECT ROUND(15.768, 2);
```

(A) 15.7

(B) 15.76

(C) 15.77

(D) 15.8

Q4. Which SQL function is used to return the current system date and time?

(A) CURDATE()

(B) DATE()

(C) SYSDATE()

(D) NOW()

Q5. Which function is used to convert all characters of a string to lowercase in SQL?

(A) LOWERCASE()

(B) LOWER()

(C) LOWER_STR()

(D) MIN()

Q6. If a string is 'Informatics Practices', what will `SELECT INSTR('Informatics Practices', 'Practices');` return?

(A) 12

(B) 13

(C) 11

(D) 10

Q7. Which function returns the name of the month for a given date?

(A) MONTHNAME()

(B) NAME()

(C) MONTH()

(D) DATE_MONTH()

Q8. What is the result of `SELECT TRIM(' CUET 2026 ');`?

(A) ' CUET 2026'

(B) 'CUET 2026 '

(C) 'CUET 2026'

(D) 'CUET2026'

Q9. In a Relational Database, a candidate key that is not selected as the primary key is known as a/an:

(A) Foreign Key

(B) Secondary Key

(C) Alternate Key

(D) Composite Key

Q10. The Relational Algebra operation used to select a subset of columns from a relation is:

(A) Selection (σ)

(B) Projection (π)

(C) Join

(D) Intersection

Q11. A column or a set of columns in one table that references the primary key of another table is called a:

(A) Candidate Key

(B) Super Key

(C) Foreign Key

(D) Primary Key

Q12. Which property of a database ensures that all transactions are treated as a single unit, which either succeeds completely or fails completely?

(A) Isolation

(B) Durability

(C) Atomicity

(D) Consistency

Q13. In a computer network, which device is used to connect multiple segments of a LAN and works at the Data Link Layer using MAC addresses?

(A) Repeater

(B) Router

(C) Switch

(D) Gateway

Q14. Which network topology requires a central controller or a hub to connect all nodes?

(A) Mesh Topology

(B) Star Topology

(C) Bus Topology

(D) Tree Topology

Q15. What is the unique 48-bit hardware address assigned to a Network Interface Card (NIC)?

(A) IP Address

(B) MAC Address

(C) Port Address

(D) URL

Q16. Given a Series 'S' with values [10, 20, 30, 40]. What will be the output of `S[1:3]`?

(A) 10, 20

(B) 20, 30

(C) 20, 30, 40

(D) 10, 20, 30

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

(A) .isnull()

(B) .empty

(C) .size

(D) .is_empty

Q18. In Pandas, the function used to read data from a comma-separated values file is:

(A) read_file()

(B) load_csv()

(C) read_csv()

(D) import_csv()

Q19. Which method is used to display the first five rows of a DataFrame 'df'?

(A) df.first(5)

(B) df.top(5)

(C) df.head()

(D) df.show(5)

Q20. To change the index of an existing DataFrame, we use the attribute:

(A) df.reindex

(B) df.index

(C) df.set_index

(D) df.change_index

Q21. Which of the following is used to access a group of rows and columns by labels in a DataFrame?

(A) .iloc[]

(B) .loc[]

(C) .at[]

(D) .ix[]

Q22. What will be the output of `df . shape` if the DataFrame has 4 rows and 3 columns?

(A) 12

(B) [4, 3]

(C) (4, 3)

(D) (3, 4)

Q23. To delete a column named 'Salary' from a DataFrame 'df', which command is correct?

(A) df.remove('Salary')

(B) del df['Salary']

(C) df.drop_column('Salary')

(D) df.erase('Salary')

Q24. Which attribute returns the number of dimensions of a Pandas object?

(A) .size

(B) .ndim

(C) .shape

(D) .index

Q25. A Pandas Series is a _____ data structure.

(A) One-dimensional

(B) Two-dimensional

(C) Three-dimensional

(D) Multi-dimensional

Q26. Which of the following is used to rename columns in a DataFrame?

(A) df.change_name()

(B) df.rename()

(C) df.columns.name()

(D) df.set_name()

Q27. To export a DataFrame 'df' to a file named 'data.csv', we use:

(A) df.write_csv('data.csv')

(B) df.to_csv('data.csv')

(C) df.save('data.csv')

(D) df.export('data.csv')

Q28. In a DataFrame, `df.T` is used for:

(A) Truncating data

(B) Transposing the DataFrame

(C) Transforming data

(D) Terminating the session

Q29. Given below are Two Statements, Choose correct:
Statement 1: A Series can store only homogeneous data.
Statement 2: A DataFrame can store heterogeneous data.
Choose the correct option:

(A) Statement 1 is correct, Statement 2 is incorrect.

(B) Statement 1 is incorrect, Statement 2 is correct.

(C) Both statements are correct.

(D) Both statements are incorrect.

Q30. Given below are Two Statements, Choose correct:
Statement 1: `loc` is label-based slicing.
Statement 2: `iloc` is integer-location based slicing.
Choose the correct option:

(A) Statement 1 is true but Statement 2 is false.

(B) Statement 1 is false but Statement 2 is true.

(C) Both Statements are true.

(D) Both Statements are false.

Q31. Case Study: A company 'TechCorp' uses a DataFrame 'Emp' to store employee details. The DataFrame has columns: 'EmpID', 'Name', 'Salary', and 'Dept'. How would they display only those employees who belong to the 'IT' department?

(A) Emp['Dept' == 'IT']

(B) Emp[Emp['Dept'] == 'IT']

(C) Emp.select('Dept' == 'IT')

(D) Emp.where('Dept' == 'IT')

Q32. Continuing the Case Study: How can the company calculate the average salary of all employees?

(A) Emp['Salary'].average()

(B) Emp['Salary'].mean()

(C) Emp.mean('Salary')

(D) Emp.avg('Salary')

Q33. Which SQL clause is used to group the result-set by one or more columns?

(A) ORDER BY

(B) GROUP BY

(C) HAVING

(D) WHERE

Q34. The HAVING clause is used in SQL because:

(A) The WHERE keyword could not be used with aggregate functions.

(B) It is faster than the WHERE clause.

(C) It is used to sort the result.

(D) It is used to join tables.

Q35. To sort the data in descending order in SQL, we use the keyword:

(A) ASC

(B) DESC

(C) SORT

(D) DOWN

Q36. Which type of Join returns all records when there is a match in either left or right table records?

(A) INNER JOIN

(B) LEFT JOIN

(C) RIGHT JOIN

(D) FULL OUTER JOIN

Q37. Match the SQL Clauses in List I with their Purposes in List II:

List I	SQL Clause	List II	Purpose
(i)	GROUP BY	(a)	Filters rows
(ii)	ORDER BY	(b)	Groups identical data
(iii)	DISTINCT	(c)	Removes duplicates
(iv)	WHERE	(d)	Sorts data

(A) (i)-b, (ii)-d, (iii)-c, (iv)-a

(B) (i)-a, (ii)-b, (iii)-c, (iv)-d

(C) (i)-b, (ii)-c, (iii)-d, (iv)-a

(D) (i)-d, (ii)-b, (iii)-a, (iv)-c

Q38. Match the Aggregate Functions in List I with their Usages in List II:

List I	Function	List II	Usage
(i)	COUNT()	(a)	Total of values
(ii)	SUM()	(b)	Number of rows
(iii)	MAX()	(c)	Average value
(iv)	AVG()	(d)	Highest value

(A) (i)-b, (ii)-a, (iii)-d, (iv)-c

(B) (i)-a, (ii)-b, (iii)-c, (iv)-d

(C) (i)-c, (ii)-d, (iii)-a, (iv)-b

(D) (i)-d, (ii)-c, (iii)-b, (iv)-a

Q39. Fill in the blank: In SQL, to filter groups returned by a GROUP BY clause, we use the _____ clause.

(A) WHERE

(B) HAVING

(C) LIKE

(D) LIMIT

Q40. To combine rows from two or more tables based on a related column, we use:

(A) GROUP BY

(B) UNION

(C) JOIN

(D) CONCAT

Q41. Which Matplotlib function is used to create a bar chart?

(A) plt.plot()

(B) plt.bar()

(C) plt.hist()

(D) plt.pie()

(E) plt.scatter()

Q42. To add a label to the X-axis in a Matplotlib plot, we use:

(A) plt.xname()

(B) plt.labelx()

(C) plt.xlabel()

(D) plt.xaxis()

Q43. Which parameter in `plt.pie()` is used to highlight or separate a slice from the pie?

(A) separate

(B) explode

(C) detach

(D) split

Q44. The function used to display the legend in a chart is:

(A) plt.show_legend()

(B) plt.legend()

(C) plt.display()

(D) plt.label()

Q45. Which chart type is best suited for showing the distribution of continuous data?

(A) Bar Chart

(B) Pie Chart

(C) Histogram

(D) Line Chart

Q46. Legal rights given to the inventor or creator to protect their invention or creation for a certain period of time is known as:

(A) Digital Footprint

(B) Intellectual Property Rights (IPR)

(C) Open Source

(D) Privacy Policy

Q47. The trail of data we leave behind when using the internet is called:

(A) Cyber Trail

(B) Digital Footprint

(C) Online History

(D) Data Residue

Q48. Rearrange the following steps to recycle E-waste correctly:
(P) Dismantling, (Q) Collection, (R) Sorting, (S) Extraction of metals.

(A) Q-R-P-S

(B) P-Q-R-S

(C) R-Q-P-S

(D) Q-P-R-S

Q49. Which of the following is a type of Cybercrime?

(A) Phishing

(B) Hacking

(C) Identity Theft

(D) All of the above

Q50. Rearrange the steps for creating a line plot in Matplotlib:
(P) plt.show(), (Q) Import matplotlib.pyplot as plt, (R) plt.plot(x, y), (S) Define data x and y.

(A) Q-S-R-P

(B) S-Q-R-P

(C) Q-R-S-P

(D) P-Q-R-S

Detailed Solutions

Q1.

Solution

Concept:
SQL provides Mathematical functions to perform calculations on numeric data. The operation of finding the remainder of a division is a common requirement in data processing.
Solution:

(a) The question asks for a function that returns the remainder of a division.

(b) In MySQL, the `MOD(n, m)` function returns the remainder of `n` divided by `m`.

(c) Alternatively, the modulo operator `%` or `n MOD m` syntax can also be used.

(d) `REM()` and `REMAINDER()` are not standard MySQL functions for this purpose.

(e) `DIV()` is used for integer division (returning the quotient).

Final Answer: The function is `MOD()`.

Answer: (B)

Q2.

Solution

Concept:
String manipulation is crucial in SQL for cleaning and extracting data. The `SUBSTR()` function (or `SUBSTRING()`) is the standard tool for retrieving a specific portion of a string.
Solution:

(a) Extraction of a substring requires three parameters: the source string, the starting position, and the number of characters.

(b) `SUBSTR(str, pos, len)` fulfills this exact requirement.

(c) `LEFT()` only extracts characters from the beginning (left side).

(d) `EXTRACT()` is generally used for date and time parts, not general strings.

(e) `PART()` is not a standard SQL string function.

Final Answer: The function is `SUBSTR()`.

Answer: (B)

Q3.

Solution

Concept:

The `ROUND()` function is used to round a numeric value to a specified number of decimal places. If the digit at the next decimal place is 5 or greater, the preceding digit is incremented.

Solution:

(a) The input value is 15.768 and the decimal place requested is 2.

(b) We look at the third decimal digit, which is 8.

(c) Since 8 is greater than 5, the second decimal digit (6) is rounded up by 1.

(d) Thus, 15.76 becomes 15.77.

Final Answer: The output is 15.77.

Answer: (C)

Q4.

Solution

Concept:

Date and Time functions allow users to interact with the server's current clock. `NOW()` is one of the most frequently used functions in SQL for timestamps.

Solution:

(a) The task is to retrieve both the current date and the current time.

(b) `CURDATE()` returns only the current date (YYYY-MM-DD).

(c) `DATE()` is used to extract the date part from a datetime expression.

(d) `NOW()` returns the current date and time in 'YYYY-MM-DD HH:MM:SS' format.

(e) `SYSDATE()` is also similar but `NOW()` is the standard answer in the CUET/IP context.

Final Answer: The function is `NOW()`.

Answer: (D)

Q5.

Solution

Concept:

To ensure case-insensitivity or uniform data formatting, SQL provides functions to change the case of strings.

Solution:

(a) Lowering the case of a string is handled by the `LOWER()` function.

(b) The syntax is `LOWER(string_expression)`.

(c) `LOWERCASE()` is not the correct function name in SQL (used in other languages like Python).

(d) `MIN()` is an aggregate function used to find the minimum value in a set, not for case conversion.

Final Answer: The function is `LOWER()`.

Answer: (B)

Q6.

Solution

Concept:

The `INSTR()` function is a fundamental string manipulation tool in SQL used to locate the position of a substring within a larger string. This is particularly useful for parsing data or identifying the presence of specific patterns within text columns. Unlike some programming languages that use zero-based indexing, SQL functions typically follow a one-based indexing system, meaning the very first character of a string is at position 1.

Solution:

(a) The function provided is `INSTR('Informatics Practices', 'Practices')`.

(b) The first argument is the source string: 'Informatics Practices'.

(c) The second argument is the substring we are searching for: 'Practices'.

(d) We count the characters in 'Informatics' to find the start of 'Practices':

- I-1, n-2, f-3, o-4, r-5, m-6, a-7, t-8, i-9, c-10, s-11.

(e) Following 'Informatics' is a space character, which occupies position 12.

(f) The substring 'Practices' starts immediately after the space, which is position 13.

(g) Therefore, the function identifies the character 'P' at the 13th index and returns that integer value.

Final Answer: The output is 13.

Answer: (B)

Q7.

Solution

Concept:

SQL provides a variety of date functions to extract specific components from a `DATE` or `DATETIME` value. While the `MONTH()` function returns the numerical representation of a month (1 through 12), reporting requirements often demand the full name of the month (e.g., 'January', 'February') for better human readability in dashboards and summaries.

Solution:

(a) The objective is to retrieve the alphabetical name of the month rather than its numeric value.

(b) `MONTHNAME()` is the specific MySQL function designed to take a date as an argument and return the corresponding month name as a string.

(c) `MONTH()` would return '5' if the date was in May, whereas `MONTHNAME()` would return 'May'.

(d) `NAME()` is a generic term often used for columns or aliases but is not a built-in date extraction function.

(e) `DATE_MONTH()` is not a valid standard SQL function for this specific purpose.

(f) Understanding the distinction between numeric extraction and string formatting is key for CUET UG Informatics Practices.

Final Answer: The function is `MONTHNAME()`.

Answer: (A)

Q8.

Solution

Concept:

The `TRIM()` function is a vital data-cleaning function used to remove unwanted whitespace from the beginning and end of a string. In real-world databases, data entry errors or file imports often result in leading or trailing spaces that can interfere with string comparisons, searching, and sorting operations. `TRIM()` ensures data integrity by stripping these external spaces while preserving spaces between words.

Solution:

(a) The input string is `' CUET 2026 '`. This string contains three leading spaces and three trailing spaces.

(b) When the `TRIM()` function is applied, it targets the ”outside” of the string.

(c) It removes all spaces located before the first non-space character (’C’).

(d) It also removes all spaces located after the last non-space character (’6’).

(e) It does *not* affect the space between ’CUET’ and ’2026’, as that is an internal space.

(f) The resulting cleaned string is `'CUET 2026'`.

(g) This process is essential for matching queries, as `'CUET 2026'` (without spaces) would not equal `' CUET 2026 '` in a strict comparison.

Final Answer: The result is ’CUET 2026’.

Answer: (C)

Q9.

Solution

Concept:

In the Relational Model, ”Keys” are used to uniquely identify records within a table. A table may have multiple columns or sets of columns that could serve as a unique identifier; each of these is called a **Candidate Key**. From this pool of candidates, the Database Administrator (DBA) selects one to be the **Primary Key**. Any other unique candidate keys that were not chosen are categorized under a specific term.

Solution:

(a) A Candidate Key is a minimal set of attributes that can uniquely identify a tuple.

(b) Once a Primary Key is designated for the table to enforce entity integrity, the remaining candidate keys are still unique but serve as ”backups.”

(c) These remaining unique identifiers are called **Alternate Keys**.

(d) For example, if a Student table has both ’Enrollment_No’ and ’Aadhar_No’, and ’Enrollment_No’ is chosen as the Primary Key, then ’Aadhar_No’ becomes the Alternate Key.

(e) A **Foreign Key** is used to link tables, and a **Composite Key** is a key consisting of multiple columns.

(f) Therefore, the correct term for a non-primary candidate key is Alternate Key.

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

Answer: (C)

Q10.

Solution

Concept:

Relational Algebra is the formal mathematical foundation for SQL. It consists of several operations that manipulate relations (tables) to produce new relations. Two of the most fundamental operations are Selection and Projection. While Selection filters the ”rows” (horizontal), there is a specific operation dedicated to filtering ”columns” (vertical).

Solution:

(a) The question focuses on selecting a subset of **columns** from a table.

(b) In Relational Algebra, this is known as the **Projection** operation.

(c) It is represented by the Greek letter Pi (π).

(d) For example, $\pi_{Name,Age}(Student)$ would create a new table containing only the Name and Age columns from the original Student table.

(e) **Selection** (σ) is used to filter rows based on a condition (e.g., Age > 18).

(f) **Join** combines two tables, and **Intersection** finds common rows between two tables.

(g) Thus, Projection is the correct choice for column-wise filtering.

Final Answer: The operation is Projection (π).

Answer: (B)

Q11.

Solution

Concept:

In the Relational Database Management System (RDBMS) architecture, the concept of ”Keys” is fundamental to maintaining data integrity and establishing relationships between disparate tables. While a Primary Key ensures that every record within a single table is unique, a mechanism is needed to ”link” these records to other tables to represent real-world connections. This link is established through referential integrity, where an attribute in a ”child” table points toward a unique identifier in a ”parent” table.

Solution:

(a) The question describes a scenario where a column in one table references the Primary Key of another table.

(b) This specific type of key is known as a **Foreign Key**.

(c) A Foreign Key acts as a cross-reference between tables. For example, in a ’Orders’ table, a ’CustomerID’ column might be a Foreign Key that refers to the ’CustomerID’ Primary Key in the ’Customers’ table.

(d) This relationship ensures that you cannot add a record to the ’Orders’ table for a customer who does not exist in the ’Customers’ table, thereby preventing ”orphan records.”

(e) A **Candidate Key** is a potential primary key, a **Super Key** is any set of attributes that identifies a row, and a **Primary Key** is the main unique identifier within its own table.

(f) Therefore, the cross-table reference is uniquely defined as a Foreign Key.

Final Answer: The key is called a Foreign Key.

Answer: (C)

Q12.

Solution

Concept:

To ensure the reliability of a database, transactions must adhere to a set of properties known as **ACID** (Atomicity, Consistency, Isolation, Durability). These properties guarantee that database transactions are processed reliably even in the event of errors, power failures, or other mishaps. One of the most critical properties among these concerns the ”all-or-nothing” execution of a transaction, ensuring that a database never remains in a partially updated state.

Solution:

(a) The question asks for the property that ensures a transaction is treated as a single, indivisible unit of work.

(b) This property is **Atomicity**.

(c) Atomicity dictates that if any part of a transaction fails, the entire transaction is rolled back, and the database is left unchanged. If every part succeeds, the transaction is committed.

(d) A classic example is a bank transfer: Money must be debited from Account A AND credited to Account B. If the credit fails, the debit must be undone to prevent the money from simply vanishing.

(e) **Consistency** ensures the database follows all rules; **Isolation** prevents concurrent transactions from interfering; and **Durability** ensures changes persist after a crash.

(f) Consequently, the ”single unit” success/failure logic is the definition of Atomicity.

Final Answer: The property is Atomicity.

Answer: (C)

Q13.

Solution

Concept:

Computer networking involves various hardware devices that operate at different layers of the Open Systems Interconnection (OSI) model. To manage traffic effectively within a Local Area Network (LAN), devices must be able to identify where data needs to go. While some older devices simply broadcast data to everyone, modern networking relies on ”intelligent” devices that can direct data to specific hardware based on unique physical addresses.

Solution:

(a) The device mentioned works at the **Data Link Layer** (Layer 2) and uses **MAC addresses** for data delivery.

(b) A **Switch** is the correct device for this role. Unlike a Hub (which broadcasts data to all ports), a Switch maintains a ”MAC address table” to learn which device is connected to which port.

(c) When a frame arrives, the Switch looks at the destination MAC address and forwards the data only to the relevant port, reducing network congestion.

(d) A **Repeater** works at the Physical Layer (Layer 1) and simply boosts signals.

(e) A **Router** works at the Network Layer (Layer 3) and uses IP addresses to connect different networks.

(f) A **Gateway** is a entry/exit point that translates protocols between different network architectures.

(g) Given the requirement for Layer 2 and MAC addresses, the Switch is the definitive answer.

Final Answer: The device is a Switch.

Answer: (C)

Q14.

Solution

Concept:

Network topology defines the physical or logical arrangement of nodes (computers, printers, etc.) and their connections within a network. The choice of topology affects cost, scalability, and fault tolerance. One specific topology is characterized by its centralized nature, where every individual node is connected to a central point, making it the most common layout for modern office and home LANs.

Solution:

(a) The question specifies a central controller or ”hub” that serves as the connection point for all other nodes.

(b) This describes the **Star Topology**.

(c) In a Star layout, if one cable connecting a node to the hub fails, only that node is disconnected, and the rest of the network continues to function. However, if the central hub itself fails, the entire network goes down.

(d) **Mesh Topology** involves redundant paths where nodes connect to many others.

(e) **Bus Topology** uses a single main backbone cable.

(f) **Ring Topology** connects nodes in a circular fashion where data travels in one direction.

(g) Because the configuration relies entirely on a central ”star” hub, the Star Topology is the correct identification.

Final Answer: The topology is Star Topology.

Answer: (B)

Q15.

Solution

Concept:

Every device capable of connecting to a network requires a unique identity to ensure that data packets reach the correct physical hardware. In the world of networking, there are two primary types of addresses: Logical (IP) and Physical (MAC). While an IP address can change depending on the network you join, the hardware address is ”burned” into the device during manufacturing.

Solution:

(a) The question asks for the unique 48-bit hardware address assigned to a Network Interface Card (NIC).

(b) This is the **MAC Address** (Media Access Control address).

(c) A MAC address is usually represented in hexadecimal format (e.g., 00:1A:2B:3C:4D:5E). It is permanent and unique to that specific piece of hardware worldwide.

(d) An **IP Address** is a logical address (usually 32-bit for IPv4) assigned by software or a router to identify a device’s location on a network.

(e) A **Port Address** is used to identify a specific process or service (like HTTP on port 80) on a computer.

(f) A **URL** (Uniform Resource Locator) is a human-readable web address (like www.google.com).

(g) Since the requirement is for a 48-bit hardware identifier on the NIC, MAC Address is the only correct answer.

Final Answer: The address is the MAC Address.

Answer: (B)

Q16.

Solution

Concept:

In the Pandas library, a Series is a one-dimensional array-like object that contains a sequence of values and an associated array of data labels called its index. Accessing a subset of these values is achieved through a process called "Slicing." Pandas slicing follows the standard Python slicing convention, which is essential to master for data manipulation. The syntax `object[start:stop]` allows users to extract a range of data points based on their numerical position.

Solution:

(a) We are given a Series `S` with values `[10, 20, 30, 40]`. By default, these are indexed from 0 to 3.

(b) The command `S[1:3]` uses numerical slicing.

(c) In Python/Pandas slicing, the `start` index is **inclusive**, while the `stop` index is **exclusive**.

(d) Therefore, `S[1:3]` will include the value at index 1 and the value at index 2, but it will **not** include the value at index 3.

(e) Looking at our data:

- Index 0: 10
- Index 1: 20
- Index 2: 30
- Index 3: 40

(f) The slice picks indices 1 and 2, which correspond to the values 20 and 30.

(g) The resulting output is a sub-Series containing these two values.

Final Answer: The output is 20, 30.

Answer: (B)

Q17.

Solution

Concept:

When working with large datasets, it is common to perform operations that might result in an empty container (for example, filtering a DataFrame where no rows meet the criteria). To handle such scenarios programmatically, Pandas provides several attributes and methods to check the state of the data structure. Attributes in Pandas are used to provide metadata about the object without performing a heavy calculation or returning a new copy of the data.

Solution:

(a) The question focuses on determining if a DataFrame contains no data (0 rows and 0 columns).

(b) The attribute `.empty` is a built-in property that returns a Boolean value: `True` if the DataFrame is empty, and `False` otherwise.

(c) It is important to distinguish this from `.isnull()`, which is a method used to find missing or NaN (Not a Number) values within an existing dataset; `isnull()` returns a mask of the same shape as the original data, not a single True/False for the whole object.

(d) The `.size` attribute returns the total number of elements (rows $\times$ columns), which would be 0 for an empty DataFrame, but `.empty` is the most direct and conventional way to check for "emptiness."

(e) Functions like `.is_empty` do not exist in the standard Pandas library.

(f) Thus, `.empty` is the correct attribute for this specific check.

Final Answer: The attribute is .empty.

Answer: (B)

Q18.

Solution

Concept:

Data acquisition is the first step in any data analysis workflow. Since the Comma-Separated Values (CSV) format is the industry standard for storing tabular data in a lightweight text format, Pandas provides a highly optimized and versatile function to load these files into a DataFrame. This function is capable of handling different delimiters, headers, and data types automatically, making it one of the most powerful tools in the library.

Solution:

(a) To import data from a CSV file into a Pandas DataFrame, the standard function used is `read_csv()`.

(b) The typical syntax is `import pandas as pd` followed by `df = pd.read_csv('filename.csv')`.

(c) This function parses the text file and converts it into a structured two-dimensional DataFrame, where the first row of the CSV is usually treated as the column header.

(d) Options like `load_csv()`, `read_file()`, or `import_csv()` are often confused with functions from other libraries or are simply incorrect naming conventions that do not exist in the Pandas namespace.

(e) Understanding the ”read_prefix” is helpful, as Pandas also uses `read_excel()`, `read_json()`, and `read_sql()` for other formats, maintaining a consistent naming pattern for developers.

Final Answer: The function is read_csv().

Answer: (C)

Q19.

Solution

Concept:

When dealing with massive DataFrames containing thousands or millions of records, printing the entire object to the console is impractical and can lead to performance issues. Data scientists often need a quick "sanity check" to see the structure of the data, the column names, and the first few values. To facilitate this, Pandas provides two complementary methods: one for viewing the beginning of the data and one for the end.

Solution:

(a) The question asks for the method to display the first five rows of a DataFrame named `df`.

(b) The `head()` method is specifically designed for this. By default, `df.head()` returns exactly the first five rows.

(c) If a user wants a different number of rows, they can pass an integer as an argument, such as `df.head(10)`.

(d) Conversely, the `tail()` method is used to view the last few rows of a DataFrame.

(e) Options like `df.first(5)` or `df.show(5)` are incorrect; while `.first()` exists in some contexts (like TimeSeries or Spark), it is not the standard method for a general preview in Pandas. `show()` is commonly used in libraries like Matplotlib or PySpark, but not for row retrieval in Pandas.

(f) Therefore, `head()` is the standard and correct tool.

Final Answer: The method is df.head().

Answer: (C)

Q20.

Solution

Concept:

The index in a DataFrame serves as the address for each row, allowing for fast lookups and data alignment. While Pandas automatically assigns a numeric index starting from 0, there are many cases where a specific column (like 'EmployeeID' or 'Date') should serve as the index to make the data more meaningful. Changing or setting this index is a fundamental step in data preparation and reorganization.

Solution:

(a) To assign a specific existing column as the new index of a DataFrame, the correct method is `set_index()`.

(b) For example, `df.set_index('ColumnName')` will remove that column from the data area and move it to the index position.

(c) It is important to distinguish this from `reindex()`, which is used to conform the DataFrame to a new set of labels, potentially adding NaN values where labels do not match. `reindex` does not "promote" a column to an index; it changes the order or content based on a list.

(d) The `.index` property is an attribute used to **view** or directly overwrite the entire index with a new list of values, but it is not a method for converting columns into indices.

(e) `change_index` is not a valid Pandas command.

(f) Thus, `set_index()` is the correct choice for transforming a column into a row identifier.

Final Answer: The attribute (method) is df.set_index.

Answer: (C)

Q21.

Solution

Concept:

In the Pandas library, there are two primary ways to access and slice data within a DataFrame: label-based indexing and integer-based indexing. Label-based indexing is a powerful feature that allows users to retrieve data using the actual names assigned to rows (indices) and columns. This makes the code much more readable and robust, as the labels usually correspond to real-world identifiers like dates, names, or categories, rather than arbitrary row numbers that might change if the data is sorted.

Solution:

(a) The question asks for the tool used to access data by **labels**.

(b) The correct answer is `.loc[]`. This is a label-location based indexer.

(c) For example, `df.loc['Row1', 'ColumnA']` retrieves the specific cell where the row label is 'Row1' and the column label is 'ColumnA'.

(d) It can also be used for slicing, such as `df.loc['StartRow':'EndRow']`, where the "stop" label is actually **inclusive**, which is a notable difference from standard Python slicing.

(e) In contrast, `.iloc[]` is used for **integer** location-based indexing (0, 1, 2...).

(f) `.at[]` is used for accessing a single scalar value by label but is less flexible than `.loc[]`.

(g) `.ix[]` is a hybrid indexer that has been deprecated in newer versions of Pandas in favor of `.loc` and `.iloc`.

Final Answer: The indexer is .loc[].

Answer: (B)

Q22.

Solution

Concept:

When working with DataFrames, understanding the "shape" of your data—specifically the number of rows and columns—is essential for data validation and for ensuring that subsequent operations (like matrix multiplication or joining tables) are mathematically possible. Pandas provides the `.shape` attribute to retrieve this information. This attribute provides a structural overview of the object's dimensions without requiring any computational overhead.

Solution:

(a) The question asks for the output of `df.shape` for a DataFrame with 4 rows and 3 columns.

(b) The `.shape` attribute in Pandas returns a **tuple** representing the dimensionality of the DataFrame.

(c) By convention, the first element of the tuple is the number of rows, and the second element is the number of columns.

(d) For our specific case: Rows = 4, Columns = 3.

(e) Therefore, the output will be `(4, 3)`.

(f) Note that a tuple is always enclosed in parentheses `()` in Python.

(g) The value 12 would be returned by `df.size` (which is the total number of elements), and `[4, 3]` would be a list, which is not the format Pandas uses for this attribute.

(h) Correct indexing and understanding of Python data types (tuples vs. lists) is a common testing point in the Informatics Practices syllabus.

Final Answer: The output is (4, 3).

Answer: (C)

Q23.

Solution

Concept:

Modifying the structure of a DataFrame often involves removing unnecessary or redundant data. In Python, there are multiple ways to delete objects or parts of objects. For a Pandas DataFrame, you can either use a built-in method like `.drop()` or use standard Python keywords. Understanding how to permanently remove a column is a basic but critical task in the data cleaning phase of any project.

Solution:

(a) The objective is to delete a column named 'Salary' from a DataFrame `df`.

(b) The standard Python keyword `del` can be used for this purpose. The syntax is `del df['ColumnName']`.

(c) This operation happens **in-place**, meaning it modifies the original DataFrame immediately and cannot be easily undone.

(d) Another common way is `df.drop(columns=['Salary'])`, but if you use `drop`, you must either reassign it (`df = df.drop(...)`) or use the `inplace=True` parameter.

(e) `df.remove()` is not a method for DataFrames (it is used for Python lists).

(f) `df.drop_column()` and `df.erase()` are not valid Pandas methods.

(g) Given the options provided, the `del` keyword is the most direct and syntactically correct way to remove a column from the DataFrame dictionary-like structure.

Final Answer: The command is del df['Salary'].

Answer: (B)

Q24.

Solution

Concept:

In data science, ”dimensionality” refers to the number of axes a data structure possesses. For example, a single point is zero-dimensional, a line is one-dimensional, and a table is two-dimensional. In Pandas, different objects have different numbers of dimensions. To programmatically determine how many axes an object has, we use a specific attribute that returns an integer representing the dimension count.

Solution:

(a) The attribute that returns the number of dimensions is `.ndim`.

(b) For a Pandas **Series**, which is a 1D array, `ndim` will return **1**.

(c) For a Pandas **DataFrame**, which is a 2D table, `ndim` will return **2**.

(d) It is important not to confuse this with other attributes:

- `.size` returns the total number of elements.
- `.shape` returns a tuple showing the length of each dimension.
- `.index` returns the labels for the rows.

(e) The ”n” in `ndim` stands for ”number,” and ”dim” stands for ”dimensions.” This naming convention is consistent with other libraries like NumPy, making it a cross-compatible concept in the Python data ecosystem.

Final Answer: The attribute is .ndim.

Answer: (B)

Q25.

Solution

Concept:

A Series is one of the two main data structures provided by the Pandas library. It is designed to hold a sequence of data of the same type (homogeneous data), similar to a column in a spreadsheet or a list in Python. However, unlike a standard list, a Series is much more powerful because it includes a set of labels (an index) that allows for labeled access and automatic data alignment during mathematical operations.

Solution:

(a) The question asks about the dimensionality of a Pandas Series.

(b) By definition, a Series has only one axis of data (the rows). Therefore, it is a **one-dimensional** data structure.

(c) Think of it as a single column of data. Even though it has an index and values, it only grows in one direction (downward).

(d) A **DataFrame**, by contrast, is **two-dimensional** because it has both rows and columns.

(e) Higher-dimensional structures (like the Panel, which was 3D) have largely been deprecated in modern Pandas in favor of using Multi-indexing on DataFrames.

(f) In the context of CUET Informatics Practices, remembering that Series = 1D and DataFrame = 2D is a fundamental requirement.

Final Answer: It is a One-dimensional data structure.

Answer: (A)

Q26.

Solution

Concept:

In data analysis, column names (headers) often need to be modified to make them more descriptive, to follow naming conventions (like removing spaces), or to reflect changes made during data processing. While you can overwrite all column names at once by assigning a new list to the `.columns` attribute, this is inefficient if you only want to change one or two specific names. Pandas provides a dedicated method that allows for selective modification using a dictionary-based mapping.

Solution:

(a) To modify specific column labels in a DataFrame, the `rename()` method is used.

(b) The most common way to use it is by passing a dictionary to the `columns` parameter: `df.rename(columns='old_name': 'new_name')`.

(c) This method is highly flexible because it allows you to change as many or as few columns as you like without affecting the rest.

(d) By default, `rename()` returns a new DataFrame with the changes. To modify the existing DataFrame directly, you must use the `inplace=True` parameter.

(e) `change_name()` and `set_name()` are not valid Pandas methods.

(f) `df.columns.name()` is used to set the name of the column axis itself, not the individual labels of the columns.

(g) Therefore, `rename()` is the standard tool for this task.

Final Answer: The method is df.rename().

Answer: (B)

Q27.

Solution

Concept:

After performing data cleaning, transformation, and analysis in Python, the final step is often to save the resulting dataset so it can be shared with others or used in other software like Microsoft Excel or Tableau. Pandas provides a variety of ”writer” functions that correspond to its ”reader” functions. These methods convert the internal DataFrame structure back into a flat file format or a database table, allowing for seamless data persistence.

Solution:

(a) To save or export a DataFrame to a CSV file, the method used is `to_csv()`.

(b) The syntax is `df.to_csv('filename.csv')`.

(c) This method provides several important parameters, such as `index=False` (to prevent saving the row indices as a separate column) and `header=True` (to include the column names in the file).

(d) Functions like `write_csv()`, `save()`, or `export()` are either non-existent in Pandas or belong to different libraries (like `pickle` or `joblib`).

(e) Remembering the ”to_prefix” naming convention is a helpful mnemonic for CUET students, as Pandas uses `to_excel()`, `to_json()`, and `to_sql()` for other export formats.

(f) Since the requirement is specifically for a CSV file, `to_csv()` is the only correct answer.

Final Answer: The method is df.to_csv(’data.csv’).

Answer: (B)

Q28.

Solution

Concept:

In linear algebra and data science, ”transposing” is the operation of flipping a matrix over its diagonal. This results in the rows becoming columns and the columns becoming rows. In the context of a Pandas DataFrame, this is extremely useful when the data orientation provided by a source (like an API or a specific CSV) does not match the orientation required for a specific chart type or statistical model.

Solution:

(a) The attribute `df.T` (or the method `df.transpose()`) is used to swap the axes of the DataFrame.

(b) If a DataFrame has labels A, B, C as columns and 1, 2, 3 as row indices, after using `.T`, 1, 2, 3 will become the columns and A, B, C will become the row indices.

(c) This is not related to ”Truncating,” which refers to cutting off data, or ”Transforming,” which usually refers to applying a function to each element via `.transform()`.

(d) It also has nothing to do with ”Terminating” a session, which would be handled by exiting the Python interpreter or closing a connection.

(e) Transposing is a common operation in Data Science workflows, especially when prepping data for Matplotlib visualizations where the categories are expected in rows rather than columns.

Final Answer: It is used for Transposing the DataFrame.

Answer: (B)

Q29.

Solution

Concept:

Understanding the nature of the data stored within Pandas objects is crucial for efficient memory management and error prevention. Data structures can be classified as homogeneous (all elements must be of the same data type) or heterogeneous (elements can be of different data types). This distinction dictates how Python handles the underlying memory and determines which operations are mathematically valid on the object.

Solution:

(a) **Statement 1** claims that a Series can store only homogeneous data. In a technical sense, a Pandas Series is built on top of a NumPy array, which is designed for a single data type (dtype). If you put multiple types in a Series, Pandas will typically "upcast" them all to the most general type (usually `object`), effectively treating them as a single type.

(b) **Statement 2** claims that a DataFrame can store heterogeneous data. This is absolutely correct. A DataFrame is essentially a collection of Series (columns), where each column can have a completely different data type (e.g., Column 1 is integers, Column 2 is strings, and Column 3 is floats).

(c) Because a DataFrame can handle diverse types across its columns while a Series is optimized for a single type per instance, both statements accurately reflect the design philosophy of the Pandas library.

(d) Thus, the correct choice is that both statements are correct.

Final Answer: Both statements are correct.

Answer: (C)

Q30.

Solution

Concept:

The ability to extract specific portions of a dataset is one of the most important skills in data handling. Pandas provides two distinct attributes for this: `loc` and `iloc`. While they may appear similar at first glance, the logic they use to find data is fundamentally different. Misunderstanding this difference is one of the most common sources of bugs in data processing scripts.

Solution:

(a) **Statement 1** says `loc` is label-based slicing. This is correct. `loc` looks at the index labels and column names. If you use a slice like `loc['A':'C']`, it includes every label between A and C, including C itself.

(b) **Statement 2** says `iloc` is integer-location based slicing. This is also correct. `iloc` works exactly like standard Python list slicing, using the 0-indexed position of the data. If you use `iloc[0:3]`, it will return positions 0, 1, and 2, excluding 3.

(c) These two indexers allow the user to choose whether they want to find data by its ”name” (loc) or its ”position” (iloc).

(d) Since both definitions provided in the statements are the standard technical definitions used in the Pandas documentation and Informatics Practices curriculum, both statements are true.

Final Answer: Both Statements are true.

Answer: (C)

Q31.

Solution

Concept:

Filtering data in a Pandas DataFrame is often performed using a technique called ”Boolean Indexing.” This process involves creating a Boolean mask—a Series of True and False values—based on a specific condition applied to a column. When this mask is passed back into the DataFrame using square brackets, Pandas returns only the rows where the mask value is True. This is the programmatic equivalent of a ”WHERE” clause in SQL or a ”Filter” in a spreadsheet.

Solution:

(a) The Case Study requires us to filter the ’Emp’ DataFrame to show only those rows where the ’Dept’ column has the value ’IT’.

(b) The inner expression `Emp['Dept'] == 'IT'` evaluates every row in the ’Dept’ column. If the value is ’IT’, it results in True; otherwise, it is False.

(c) To actually retrieve the data, this result must be wrapped in the DataFrame selection syntax: `Emp[ ... ]`.

(d) Therefore, the correct syntax is `Emp[Emp['Dept'] == 'IT']`.

(e) Option (A) `Emp['Dept' == 'IT']` is incorrect because it tries to use a Boolean comparison as a column label.

(f) Options (C) and (D) use `.select()` and `.where()`, which are common in other languages like SQL or Spark, but in standard Pandas, the square bracket notation is the idiomatic and most common way to filter.

Final Answer: The command is Emp[Emp[’Dept’] == ’IT’].

Answer: (B)

Q32.

Solution

Concept:

In data analysis, summarizing numerical information is a key task. Pandas provides a set of statistical methods that can be applied to both Series and DataFrames to calculate values like the total, maximum, minimum, and average. In the Python scientific ecosystem, the mathematical "average" is typically referred to by its statistical term, the "mean." Understanding the correct method naming is essential for performing accurate data aggregation.

Solution:

(a) The objective is to find the average of the 'Salary' column in the 'Emp' DataFrame.

(b) In Pandas, the method used to calculate the arithmetic average is `.mean()`.

(c) First, we must select the specific column: `Emp['Salary']`.

(d) Then, we apply the method: `.mean()`.

(e) Putting it together, `Emp['Salary'].mean()` calculates the sum of all salaries and divides by the count of non-null entries.

(f) Option (A) `.average()` is incorrect because while "average" is the common English term, it is not a built-in method name in Pandas (though it exists in NumPy as `np.average`).

(g) Option (C) uses `Emp.mean('Salary')`, but `mean()` usually takes an axis (0 or 1) as an argument, not a column name, when called on a whole DataFrame.

(h) Option (D) `.avg()` is also not a valid Pandas method.

Final Answer: The command is Emp['Salary'].mean().

Answer: (B)

Q33.

Solution

Concept:

SQL is designed to handle large datasets by organizing data into meaningful categories. The `GROUP BY` clause is a powerful tool used in collaboration with aggregate functions (like `COUNT`, `SUM`, `AVG`) to group rows that have the same values in specified columns into summary rows. This allows users to generate reports, such as finding the total sales per region or the number of students in each department.

Solution:

(a) The question asks for the clause used to group a result-set by one or more columns.

(b) The `GROUP BY` clause is specifically created for this purpose. It follows the `WHERE` clause (if any) and precedes the `ORDER BY` clause.

(c) For example, `SELECT Dept, COUNT(*) FROM Emp GROUP BY Dept;` would show each department name alongside the count of employees in that department.

(d) `ORDER BY` is used exclusively for sorting the output, not for grouping data.

(e) `HAVING` is used to filter groups *after* they have been created, but it does not perform the grouping itself.

(f) `WHERE` is used to filter individual rows *before* any grouping takes place.

(g) Thus, for the task of categorization, `GROUP BY` is the fundamental requirement.

Final Answer: The clause is GROUP BY.

Answer: (B)

Q34.

Solution

Concept:

In SQL, filtering data is performed using conditions. However, because of the logical order of query execution, the standard `WHERE` clause cannot be used to filter results based on aggregate values (like a sum or an average). This is because `WHERE` filters the raw rows before the grouping and aggregation occur. To solve this, SQL introduced a specific keyword to filter the "summarized" data.

Solution:

(a) The `HAVING` clause was added to SQL because the `WHERE` keyword could not be used with aggregate functions.

(b) Think of the sequence: First, `WHERE` filters the rows. Second, `GROUP BY` organizes them. Third, the aggregate function (like `SUM`) calculates a value for each group. Finally, `HAVING` filters those groups based on the calculated value.

(c) For example: `SELECT Dept FROM Emp GROUP BY Dept HAVING AVG(Salary) > 50000;`. Here, `WHERE` couldn't be used because the "Average Salary" doesn't exist until after the groups are made.

(d) Option (B) is incorrect because `HAVING` is not necessarily "faster"; it is logically different.

(e) Option (C) is incorrect because sorting is the job of `ORDER BY`.

(f) Option (D) is incorrect because joining is done via the `JOIN` clause.

Final Answer: The clause is used because WHERE could not be used with aggregate functions.

Answer: (A)

Q35.

Solution

Concept:

Presenting data in a logical order is a core requirement of database reporting. Whether it is a list of names in alphabetical order or a list of transactions from highest to lowest value, sorting makes information much easier for humans to process. SQL uses a specific clause to handle this, which allows the user to specify both the column to sort by and the direction of the sort.

Solution:

(a) To sort data in SQL, we use the `ORDER BY` clause.

(b) Inside this clause, we specify the direction: `ASC` for Ascending (lowest to highest) or `DESC` for Descending (highest to lowest).

(c) If no keyword is specified, SQL defaults to `ASC`.

(d) For example, `SELECT * FROM Emp ORDER BY Salary DESC;` will show the employee with the highest salary at the top of the list.

(e) Option (A) `ASC` is for ascending order (A to Z, 1 to 10).

(f) Options (C) `SORT` and (D) `DOWN` are not valid SQL keywords for defining sort direction; they might be used in other programming contexts but not in standard SQL queries.

(g) Therefore, `DESC` is the shorthand keyword for descending.

Final Answer: The keyword is DESC.

Answer: (B)

Q36.

Solution

Concept:

In a relational database, data is often spread across multiple tables to reduce redundancy. ”Joins” are SQL operations used to combine rows from two or more tables based on a related column between them. While an Inner Join only returns matching rows, ”Outer Joins” are used when we want to include rows that do not have a corresponding match in the other table. There are three types of Outer Joins: Left, Right, and Full.

Solution:

(a) The question asks for a join that returns all records when there is a match in either the left OR the right table.

(b) This is the definition of a **FULL OUTER JOIN**.

(c) In a Full Outer Join, the result set contains all rows from both tables. If a row in the left table has no match in the right table, the right-side columns will contain NULLs, and vice versa.

(d) An **INNER JOIN** only returns rows where there is a match in both tables.

(e) A **LEFT JOIN** returns all rows from the left table and matched rows from the right.

(f) A **RIGHT JOIN** returns all rows from the right table and matched rows from the left.

(g) Since the requirement is to capture all records from both sides regardless of a match, the Full Outer Join is the correct mechanism.

Final Answer: The join is FULL OUTER JOIN.

Answer: (D)

Q37.

Solution

Concept:

In Structured Query Language (SQL), the ability to manipulate and retrieve data efficiently depends on the correct application of various clauses. Each clause acts as a logical instruction that tells the database engine how to process the dataset. Understanding the sequence and purpose of these clauses—ranging from initial row filtering to the final presentation of sorted results—is fundamental for any database administrator or data analyst working within the relational model.

Solution:

(a) **GROUP BY (i):** This clause is used to arrange raw data into groups of rows that share the same values in specified columns. It is typically used in conjunction with aggregate functions to perform calculations on each group. Therefore, it matches with **(b) Groups identical data**.

(b) **ORDER BY (ii):** This clause is used to sort the result-set of a query. By default, it sorts in ascending order, but it can be configured for descending order using the `DESC` keyword. Therefore, it matches with **(d) Sorts data**.

(c) **DISTINCT (iii):** This keyword is applied within a `SELECT` statement to filter out redundant entries and ensure that only unique values are returned to the user. Therefore, it matches with **(c) Removes duplicates**.

(d) **WHERE (iv):** This is the primary filtering clause used to extract only those records that fulfill a specified condition. It operates on individual rows before any grouping occurs. Therefore, it matches with **(a) Filters rows**.

Final Answer: The correct mapping is (i)-b, (ii)-d, (iii)-c, (iv)-a.

Answer: (A)

Q38.

Solution

Concept:

SQL Aggregate Functions are specialized tools that perform a mathematical calculation on a set of values across multiple rows and return a single, summarized value. These functions are indispensable for generating reports and high-level data insights. A key characteristic of most aggregate functions is that they ignore `NULL` values in their calculations (with the exception of `COUNT(*)`), ensuring that the statistical summary remains accurate based on existing data.

Solution:

(a) **COUNT() (i):** This function is designed to return the number of rows that match a specific criterion or the total number of entries in a column. It effectively tallies the occurrences. Therefore, it matches with **(b) Number of rows**.

(b) **SUM() (ii):** This function is used to calculate the grand total of all numeric values in a particular column. It is commonly used for financial reporting, such as calculating total sales. Therefore, it matches with **(a) Total of values**.

(c) **MAX() (iii):** As the name suggests, this function identifies and returns the largest value from a selected column, whether that value is numeric, a string (alphabetical), or a date. Therefore, it matches with **(d) Highest value**.

(d) **AVG() (iv):** This function calculates the arithmetic mean of a numeric column by dividing the sum of values by the count of non-null entries. Therefore, it matches with **(c) Average value**.

Final Answer: The correct mapping is (i)-b, (ii)-a, (iii)-d, (iv)-c.

Answer: (A)

Q39.

Solution

Concept:

In SQL, the order of operations is critical. When a user wants to filter data, they typically reach for the `WHERE` clause. However, the `WHERE` clause acts on individual rows. Once rows are grouped together using the `GROUP BY` clause, the individual rows lose their identity, and only the group-level data remains. To filter these groups based on a condition, a different clause must be utilized.

Solution:

(a) The question asks for the clause used to filter groups returned by a `GROUP BY` operation.

(b) The correct answer is the **HAVING** clause.

(c) For example, if you group employees by 'Department' and want to see only departments where the 'Total Salary' exceeds 1,000,000, you would use: `GROUP BY Dept HAVING SUM(Salary) > 1000000`.

(d) `WHERE` cannot be used here because `SUM(Salary)` is an aggregate value, and `WHERE` does not support aggregate functions.

(e) `LIKE` is an operator used for pattern matching in strings (e.g., finding names starting with 'A').

(f) `LIMIT` is used to restrict the number of rows returned in the final result set.

(g) Therefore, `HAVING` is the specialized filtering tool for grouped data.

Final Answer: The clause is HAVING.

Answer: (B)

Q40.

Solution

Concept:

Data in a relational database is designed to be normalized, meaning it is split into small, manageable tables to avoid data duplication. To reconstruct the complete information for a report, these tables must be stitched back together. This ”stitching” is done by identifying a common link between the tables, such as an ID number that appears in both. The operation that performs this horizontal combination of rows is a core feature of SQL.

Solution:

(a) The process of combining rows from two or more tables based on a related column is called a **JOIN**.

(b) A JOIN allows you to query data from multiple tables as if they were a single table. For instance, you can join a ’Students’ table with a ’Marks’ table using ’RollNo’ as the common link.

(c) **GROUP BY** is used for data aggregation, not for combining different tables.

(d) **UNION** is used to combine the results of two SELECT statements vertically (stacking rows on top of each other), but both statements must have the same number of columns and data types. It does not ”link” tables based on a key.

(e) **CONCAT** is a string function used to join two or more strings together (e.g., joining ’First_Name’ and ’Last_Name’), not for merging tables.

(f) Thus, for relational data combination, JOIN is the correct technical term.

Final Answer: The operation is JOIN.

Answer: (C)

Q41.

Solution

Concept:

Matplotlib is the most widely used library in Python for creating static, animated, and interactive visualizations. One of its primary strengths is the ability to generate different types of plots using simple, intuitive functions. For comparing discrete categories of data—such as sales figures for different months or the number of students in different subjects—the bar chart is the industry standard. It represents data using rectangular bars with lengths proportional to the values they represent.

Solution:

(a) The question asks for the specific function within the `pyplot` module of Matplotlib used to create a bar chart.

(b) The correct function is `plt.bar()`.

(c) This function requires at least two arguments: the categories (usually on the X-axis) and the heights of the bars (usually on the Y-axis).

(d) `plt.plot()` is a general-purpose function used primarily for creating line plots.

(e) `plt.hist()` is used specifically for histograms, which represent the frequency distribution of continuous numerical data, not discrete categories.

(f) `plt.pie()` is used for circular charts representing proportions of a whole.

(g) `plt.scatter()` is used for scatter plots to show the relationship between two numerical variables.

(h) Therefore, for categorical comparison, `bar()` is the definitive tool.

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

Answer: (B)

Q42.

Solution

Concept:

A chart without labels is often meaningless to the viewer, as they will not know what the numerical values on the axes represent. Professional data visualization requires clear documentation of the data being presented. Matplotlib provides straightforward functions to add metadata to a plot, including titles, legends, and axis labels. These labels help provide context, such as identifying if the X-axis represents ”Time in Seconds,” ”Years,” or ”Product Names.”

Solution:

(a) To add a description or label to the horizontal axis (X-axis), Matplotlib uses a specific function in the `pyplot` module.

(b) The correct function is `plt.xlabel()`.

(c) The syntax typically involves passing a string as an argument, for example: `plt.xlabel("Months")`.

(d) Similarly, `plt.ylabel()` is used for the vertical axis, and `plt.title()` is used for the main heading of the plot.

(e) Options like `plt.xname()`, `plt.labelx()`, and `plt.xaxis()` are incorrect as they do not exist as standard labeling functions in the Matplotlib API.

(f) Mastering these naming conventions is a standard requirement for the Informatics Practices curriculum to ensure students can produce readable and professional-grade charts.

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

Answer: (C)

Q43.

Solution

Concept:

A pie chart is used to show the relative sizes of data as slices of a circle. While a standard pie chart shows all slices touching, it is often useful for a presenter to draw attention to one specific data point (e.g., highlighting a specific region's market share). In visualization terminology, "exploding" a slice means shifting it slightly away from the center of the pie to make it stand out visually.

Solution:

(a) The question focuses on the parameter used to highlight or separate a specific slice in a pie chart.

(b) In the `plt.pie()` function, this is achieved using the `explode` parameter.

(c) The `explode` parameter takes a list or array of values, where each value represents the fraction of the radius by which that specific slice will be offset.

(d) For example, `explode=[0, 0.1, 0, 0]` would pull the second slice out from the center while leaving the others in place.

(e) Options like `separate`, `detach`, and `split` are not valid parameters for the `plt.pie()` function.

(f) Understanding this parameter is key for creating more advanced and readable visualizations where certain data points need emphasis.

Final Answer: The parameter is explode.

Answer: (B)

Q44.

Solution

Concept:

When multiple data series are plotted on the same chart (for example, two different lines representing the growth of two different companies), it becomes difficult for the viewer to distinguish which line corresponds to which data. A ”legend” is a key or table that explains what each color, line style, or marker in the plot represents. In Matplotlib, the process of creating a legend involves two steps: labeling the data during plotting and then calling the function to display the key.

Solution:

(a) To show the key/legend on a chart, the function `plt.legend()` is used.

(b) For the legend to be effective, you must first assign a `label` to your plot commands, such as `plt.plot(x, y, label="Company A")`.

(c) Calling `plt.legend()` then tells Matplotlib to look for all these labels and organize them into a small box on the plot area.

(d) `plt.show_legend()` is an incorrect function name and does not exist in Matplotlib.

(e) `plt.display()` is not the standard way to show a legend; `plt.show()` is used to display the entire figure, but it doesn't automatically create a legend.

(f) `plt.label()` is not a valid standalone function for creating a legend box.

(g) Thus, `plt.legend()` is the essential command for making multi-series plots understandable.

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

Answer: (B)

Q45.

Solution

Concept:

Choosing the right type of chart is the most important decision in data visualization. While bar charts are for categories and line charts are for trends over time, a **Histogram** is a specialized plot used to represent the distribution of numerical data. It works by ”binning” the range of values—dividing the entire range of values into a series of intervals—and then counting how many values fall into each interval. This provides a visual representation of data density and spread.

Solution:

(a) The question asks which chart type is best for showing the distribution of continuous data.

(b) A **Histogram** is the correct answer. It allows you to see where values are concentrated (e.g., most students scored between 70 and 80) and where they are sparse.

(c) A **Bar Chart** is for discrete, categorical data (e.g., comparing red cars vs. blue cars), not continuous numerical distributions.

(d) A **Pie Chart** shows parts of a whole (proportions) but does not show the frequency distribution or density.

(e) A **Line Chart** is best for showing how a variable changes over a continuous interval, usually time.

(f) In the context of the Informatics Practices syllabus, understanding the statistical purpose of a histogram (frequency distribution) is a recurring and important theme.

Final Answer: The chart type is Histogram.

Answer: (C)

Q46.

Solution

Concept:

In the digital age, the products of the human intellect—such as software, music, literature, and inventions—are considered valuable assets. Just as physical property is protected by law, intangible creations are protected under a legal framework known as Intellectual Property Rights (IPR). This framework ensures that creators have the exclusive right to benefit from their work, encouraging innovation by providing legal protection against unauthorized copying or distribution.

Solution:

(a) The question describes legal rights given to an inventor or creator to protect their invention for a specific period.

(b) These are collectively known as **Intellectual Property Rights (IPR)**.

(c) IPR includes several categories: **Patents** (for inventions), **Copyrights** (for artistic and literary works), **Trademarks** (for brand names and logos), and **Trade Secrets**.

(d) A **Digital Footprint** is the record of a user's activities online, which is unrelated to legal ownership.

(e) **Open Source** refers to a philosophy where software code is made available for anyone to modify and distribute, which is often the opposite of a restrictive IPR model.

(f) **Privacy Policy** is a statement that explains how an organization handles user data.

(g) Therefore, IPR is the correct term for the legal protection of creative and intellectual output.

Final Answer: The rights are Intellectual Property Rights (IPR).

Answer: (B)

Q47.

Solution

Concept:

Every time we interact with the digital world—whether by sending an email, posting on social media, or even just visiting a website—we leave behind a trail of data. This trail is permanent and can be used to build a comprehensive profile of an individual's behavior, interests, and location. Understanding this concept is critical for digital citizenship, as it highlights the fact that "private" online actions are often recorded and archived by various entities.

Solution:

(a) The question refers to the trail of data left behind during internet usage.

(b) This is officially known as a **Digital Footprint**.

(c) Digital footprints are categorized into two types: **Active** (data the user intentionally shares, like a social media post) and **Passive** (data collected without the user's active participation, like an IP address or browsing history).

(d) **Cyber Trail** and **Online History** are informal terms that describe parts of this concept but are not the standard technical terms used in the IP curriculum.

(e) **Data Residue** usually refers to data that remains on a physical hard drive after a file has been deleted.

(f) Since the "footprint" metaphor perfectly describes the lasting impression of our online movements, it is the standard answer.

Final Answer: The trail is called a Digital Footprint.

Answer: (B)

Q48.

Solution

Concept:

Electronic waste (E-waste) consists of discarded electrical or electronic devices. If not handled correctly, E-waste can leak toxic chemicals into the environment. Recycling E-waste is a complex industrial process aimed at recovering valuable materials like gold, copper, and palladium while safely disposing of hazardous components like lead and mercury. Understanding the logical sequence of these steps is essential for promoting environmental sustainability in information technology.

Solution:

(a) To solve this rearrangement question, we must follow the logical flow of a recycling plant.

(b) **Step 1: Collection (Q)** – Before any processing can happen, the waste must be gathered from households and businesses.

(c) **Step 2: Sorting (R)** – The gathered waste is sorted into categories (e.g., mobile phones, monitors, batteries) because each requires a different processing method.

(d) **Step 3: Dismantling (P)** – The devices are manually or mechanically broken down into their individual components (circuit boards, plastic casings, glass).

(e) **Step 4: Extraction of metals (S)** – This is the final, technical stage where chemical or thermal processes are used to recover precious metals from the circuit boards.

(f) The correct sequence is therefore **Q-R-P-S**.

(g) This logical flow ensures efficiency and safety throughout the recycling lifecycle.

Final Answer: The correct sequence is Q-R-P-S.

Answer: (A)

Q49.

Solution

Concept:

Cybercrime refers to any criminal activity that involves a computer, networked device, or a network. As society becomes more dependent on digital infrastructure, the variety and sophistication of these crimes continue to grow. These crimes can target individuals (to steal identity or money) or organizations (to disrupt service or steal proprietary data). Knowledge of these threats is the first step toward effective cybersecurity.

Solution:

(a) The question asks to identify types of Cybercrime.

(b) **Phishing** is a fraudulent attempt to obtain sensitive information (like passwords) by disguising as a trustworthy entity in an electronic communication.

(c) **Hacking** is the unauthorized access to or control over computer network security systems for some illicit purpose.

(d) **Identity Theft** involves the deliberate use of someone else's identity, usually as a method to gain a financial advantage or obtain credit and other benefits in the other person's name.

(e) Since all three options—Phishing, Hacking, and Identity Theft—are recognized legal and technical categories of criminal activity performed using digital means, the correct choice is "All of the above."

(f) Each of these crimes represents a significant threat to digital privacy and security.

Final Answer: All of the above.

Answer: (D)

Q50.

Solution

Concept:

In Python's Matplotlib library, the workflow for creating a visualization follows a standard, logical sequence. You must move from the general environment setup to data definition, then to the actual plotting command, and finally to the display step. Skipping or reordering these steps will result in errors or an empty output. This question tests the student's understanding of the basic script structure required to generate a graph.

Solution:

(a) To create a plot, we must follow a sequential order of operations.

(b) **Step 1: Import matplotlib.pyplot as plt (Q)** – You must first load the library into your Python environment. You cannot use functions like `plot()` or `show()` until the library is imported.

(c) **Step 2: Define data x and y (S)** – You need values to plot. This involves creating lists or NumPy arrays for your coordinates.

(d) **Step 3: plt.plot(x, y) (R)** – Once you have the library and the data, you call the specific function to draw the line in the computer's memory (the "canvas").

(e) **Step 4: plt.show() (P)** – This is the final step that actually opens a window and renders the plot for the user to see.

(f) The resulting logical sequence is **Q-S-R-P**.

(g) This pattern is foundational for almost all visualization tasks in Python.

Final Answer: The correct sequence is Q-S-R-P.

Answer: (A)

Answer Key

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