CUET-UG Information Practices Sample Paper-3

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

Instructions

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

Q1. What will be the output of the following SQL query?

```
SELECT ROUND(789.4567, -2);
```

(A) 789.46

(B) 700

(C) 800

(D) 790

Q2. Which SQL function returns the position of the first occurrence of a substring within a string?

(A) MID()

(B) LOCATE()

(C) FIND()

(D) POSITIONAL()

Q3. What will be the result of `SELECT MOD(29,5);` ?

(A) 4

(B) 5

(C) 1

(D) 0

Q4. Which function can be used to extract only the month from a DATE column named `dob`?

(A) `EXTRACT(MONTH FROM dob)`

(B) `MONTHS(dob)`

(C) `GETMONTH(dob)`

(D) `DATE_MONTH(dob)`

Q5. What is the output of `SELECT UCASE(LEFT('informatics',5));` ?

(A) INFOR

(B) infor

(C) INFORMATICS

(D) inforM

Q6. Which query correctly displays the current system date and time in MySQL?

(A) CURRENT()

(B) GETDATE()

(C) NOW()

(D) SYSDATEONLY()

Q7. What will be returned by `SELECT LENGTH(TRIM(' CUET '));` ?

(A) 8

(B) 6

(C) 4

(D) 2

Q8. Which SQL function is used to return the largest integer value less than or equal to a given number?

(A) CEIL()

(B) FLOOR()

(C) ROUND()

(D) INT()

Q9. In relational algebra, which operation is used to extract specific columns from a relation?

(A) Selection

(B) Projection

(C) Join

(D) Union

Q10. A foreign key in a table is primarily used to:

(A) Uniquely identify records

(B) Remove duplicate values

(C) Establish relationship between tables

(D) Encrypt records

Q11. Which of the following properties is NOT associated with a primary key?

(A) Unique values

(B) Cannot contain NULL

(C) Can have duplicate values

(D) Identifies tuples uniquely

Q12. Which relational algebra operation combines tuples from two relations based on a related attribute?

(A) Cartesian Product

(B) Join

(C) Difference

(D) Projection

Q13. Which network topology becomes difficult to troubleshoot because all nodes are connected in a closed loop?

(A) Star

(B) Bus

(C) Ring

(D) Tree

Q14. Which device operates at the Data Link Layer and uses MAC addresses for forwarding frames?

(A) Router

(B) Switch

(C) Gateway

(D) Repeater

Q15. Which statement correctly differentiates MAC address and IP address?

(A) MAC address changes frequently while IP address never changes

(B) MAC address identifies hardware while IP address identifies logical network location

(C) Both are assigned by ISP only

(D) Both operate only at Application Layer

Q16. Which Pandas function is used to read a CSV file into a DataFrame?

(A) `pd.loadcsv()`

(B) `pd.read_csv()`

(C) `pd.import_csv()`

(D) `pd.fetch_csv()`

Q17. What will `df.shape` return for a DataFrame with 12 rows and 5 columns?

(A) (5,12)

12,5

(B) (12,5)

(C) 60

Q18. Which statement correctly selects rows from index 2 to 5 in a DataFrame `df` using `iloc`?

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

(B) `df.iloc[2:6]`

(C) `df.iloc[1:5]`

(D) `df.iloc[2,5]`

Q19. Which attribute returns the column labels of a DataFrame?

(A) `df.axes()`

(B) `df.columns`

(C) `df.rows`

(D) `df.dimension`

Q20. Which command removes rows containing missing values?

(A) `dropna()`

(B) `fillna()`

(C) `remove_null()`

(D) `clearna()`

Q21. What will be the output type of `df['Marks']` ?

(A) DataFrame always

(B) Series

(C) Tuple

(D) Dictionary

Q22. Which method is used to display the first five rows of a DataFrame?

(A) top()

(B) first()

(C) head()

(D) preview()

Q23. Which function is used to write a DataFrame into a CSV file?

(A) save_csv()

(B) export_csv()

(C) to_csv()

(D) write_csv()

Q24. Consider: `s = pd.Series([10,20,30], index=['a','b','c'])`. What is returned by `s['b']` ?

(A) a

(B) 10

(C) 20

(D) Error

Q25. Which parameter of `read_csv()` is used to make a specific column the index?

(A) key_col

(B) header

(C) index_col

(D) indexer

Q26. Which operation will sort a DataFrame by column `Marks` in descending order?

(A) `df.sort_values('Marks', ascending=False)`

(B) `df.sort('Marks','DESC')`

(C) `df.arrange(Marks)`

(D) `df.order(Marks)`

Q27. What does `axis=1` represent in most Pandas operations?

(A) Rows

(B) Columns

(C) Index labels only

(D) Both rows and columns

Q28. Which function combines two DataFrames vertically?

(A) append() / concat()

(B) merge()

(C) joincol()

(D) intersect()

Q29. Which method is used to obtain statistical summary of numeric columns?

(A) summary()

(B) describe()

(C) statistics()

(D) analyze()

Q30. Which statement creates a DataFrame from a dictionary `d`?

(A) `pd.frame(d)`

(B) `pd.DataFrame(d)`

(C) `pd.create(d)`

(D) `DataFrame.make(d)`

Q31. Which indexing method is label-based?

(A) iloc

(B) loc

(C) indexloc

(D) atrow

Q32. What is the output of `len(df.columns)` ?

(A) Number of rows

(B) Total elements

(C) Number of columns

(D) Number of null values

Q33. Which clause is used to filter groups formed by `GROUP BY`?

(A) WHERE

(B) HAVING

(C) ORDER BY

(D) FILTER

Q34. What will be the default sorting order of `ORDER BY marks`?

(A) Random

(B) Descending

(C) Ascending

(D) Alphabetical only

Q35. Which join returns all matching records and also unmatched records from both tables?

(A) INNER JOIN

(B) LEFT JOIN

(C) FULL OUTER JOIN

(D) CROSS JOIN

Q36. Which query correctly counts the number of employees department-wise?

(A) `SELECT dept, COUNT(*) FROM emp GROUP BY dept;`

(B) `SELECT COUNT(dept) FROM emp;`

(C) `SELECT dept FROM emp ORDER BY COUNT(*);`

(D) `SELECT GROUP dept COUNT(*) FROM emp;`

Q37. Which SQL clause must appear after `GROUP BY`?

(A) SELECT

(B) FROM

(C) HAVING

(D) CREATE

Q38. Which join produces Cartesian Product when no condition is specified?

(A) INNER JOIN

(B) NATURAL JOIN

(C) CROSS JOIN

(D) LEFT JOIN

Q39. Which aggregate function ignores NULL values?

(A) COUNT(column_name)

(B) SUM()

(C) AVG()

(D) All of these

Q40. Which query displays departments having average salary greater than 50000?

(A) `SELECT dept FROM emp WHERE AVG(salary)>50000;`

(B) `SELECT dept, AVG(salary) FROM emp GROUP BY dept HAVING AVG(salary)`

(C) `SELECT AVG(salary) FROM emp ORDER BY dept;`

(D) `SELECT dept AVG(salary) WHERE salary>50000;`

Q41. Which keyword is used to eliminate duplicate rows in query output?

(A) UNIQUE

(B) DISTINCT

(C) DIFFERENT

(D) REMOVE

Q42. Which Matplotlib function is used to create a histogram?

(A) `plt.bar()`

(B) `plt.scatter()`

(C) `plt.hist()`

(D) `plt.plotpie()`

Q43. Which parameter is used to display labels in a pie chart?

(A) legends

(B) tags

(C) labels

(D) names

Q44. Which command displays the legend on a graph?

(A) `plt.key()`

(B) `plt.legend()`

(C) `plt.label()`

(D) `plt.note()`

Q45. Which chart type is best suited to represent frequency distribution of continuous data?

(A) Pie Chart

(B) Histogram

(C) Bar Graph

(D) Scatter Plot

Q46. Which function is used to set label for X-axis?

(A) `plt.xname()`

(B) `plt.xlabel()`

(C) `plt.labelx()`

(D) `plt.axisx()`

Q47. Which cybercrime involves fraudulently obtaining sensitive information through fake websites or emails?

(A) Hacking

(B) Phishing

(C) Spoofing

(D) Sniffing

Q48. Which of the following is covered under Intellectual Property Rights (IPR)?

(A) Copyright

(B) Patent

(C) Trademark

(D) All of these

Q49. E-waste primarily refers to:

(A) Waste generated by food industries

(B) Biodegradable waste

(C) Discarded electronic devices and components

(D) Agricultural waste

Q50. Which statement best defines a digital footprint?

(A) Physical storage of files in hard disk

(B) Online trace left by user activities on the internet

(C) Footprint scanner technology

(D) Computer hardware architecture

Detailed Solutions

Q1.

Solution

Concept: The ‘ROUND()‘ function in SQL, especially its behavior with negative decimal places.

Solution: The ‘ROUND(number, decimals)‘ function rounds ‘number‘ to ‘decimals‘ decimal places. When the ‘decimals‘ parameter is a negative number, it rounds to the left of the decimal point.
- ‘ROUND(789.4567, -2)‘ means rounding 789.4567 to the nearest 100 (since -2 indicates two places to the left of the decimal, i.e., the hundreds place).
- 789.4567 is between 700 and 800.
- It is closer to 800 (difference of 10.5433) than to 700 (difference of 89.4567).

Therefore, rounding 789.4567 to the nearest hundred results in 800.

Final Answer : “800”

Answer: (C)

Q2.

Solution

Concept: SQL string functions for finding the position of a substring.

Solution: We need to identify the SQL function that returns the starting position of the first occurrence of a substring within a larger string.
- ‘MID()‘ (or ‘SUBSTRING()‘): This function extracts a substring from a string, it does not return its position.
- ‘LOCATE()‘: This function (common in MySQL, for example) returns the starting position of the first occurrence of a substring within a string. Other similar functions in different SQL dialects include ‘INSTR()‘ (Oracle, PostgreSQL) and ‘CHARINDEX()‘ (SQL Server).
- ‘FIND()‘: This is not a standard SQL function for this specific purpose, though variations exist in some systems (e.g., ‘FIND_IN_SET‘ in MySQL).
- ‘POSITIONAL()‘: This is not a standard SQL function.

Among the given options, ‘LOCATE()‘ is the correct function for this task.

Final Answer : “LOCATE()”

Answer: (B)

Q3.

Solution

Concept: The ‘MOD()‘ function in SQL, which calculates the remainder of a division.

Solution: The ‘MOD(number, divisor)‘ function in SQL returns the remainder of ‘number‘ divided by ‘divisor‘.
In the query ‘SELECT MOD(29,5);‘:
- We are dividing 29 by 5.
- 29 divided by 5 is 5 with a remainder.
- $29 = 5 \times 5 + 4$
- The remainder is 4.

Final Answer : “4”

Answer: (A)

Q4.

Solution

Concept: SQL date and time functions, specifically extracting parts of a date.

Solution: To extract specific parts (like year, month, day, hour, etc.) from a date or timestamp column, standard SQL provides the ‘EXTRACT()‘ function.
- ‘EXTRACT(MONTH FROM dob)‘: This is the standard SQL syntax to extract the month component from a ‘DATE‘ column named ‘dob‘.
- ‘MONTHS(dob)‘: This is not a standard SQL function for extracting the month. ‘MONTHS_BETWEEN‘ is an Oracle function to calculate the number of months between two dates.
- ‘GETMONTH(dob)‘: This is not a standard SQL function.
- ‘DATE_MONTH(dob)‘: This is not a standard SQL function.

Thus, ‘EXTRACT(MONTH FROM dob)‘ is the correct and standard way to perform this operation.

Final Answer : “EXTRACT(MONTH FROM dob)”

Answer: (A)

Q5.

Solution

Concept: Nested SQL string functions, specifically ‘LEFT()‘ for substring extraction and ‘UCASE()‘ (or ‘UPPER()‘) for case conversion.

Solution: The SQL query ‘SELECT UCASE(LEFT(’informatics’,5));‘ involves nested functions, which are evaluated from the innermost function outwards.
1. ‘LEFT(’informatics’, 5)‘: This function extracts a specified number of characters from the left side of a string. Here, it extracts the first 5 characters from ’informatics’.
- Result of ‘LEFT(’informatics’, 5)‘: ’infor’
2. ‘UCASE(’infor’)‘: This function (also known as ‘UPPER()‘ in many SQL dialects) converts all characters in a string to uppercase.
- Result of ‘UCASE(’infor’)‘: ’INFOR’

Therefore, the final output of the query is ’INFOR’.

Final Answer : “INFOR”

Answer: (A)

Q6.

Solution

Concept: SQL functions used to retrieve the current system date and time, specific to MySQL.

Solution: Different SQL database systems use different functions to retrieve the current system date and time.
- ‘CURRENT()‘: Not a standard function for this. ‘CURRENT_DATE‘, ‘CURRENT_TIME‘, ‘CURRENT_TIMESTAMP‘ are standard SQL-99 functions, but ‘CURRENT()‘ by itself is not.
- ‘GETDATE()‘: This function is commonly used in SQL Server to get the current date and time.
- ‘NOW()‘: This function is standard in MySQL (and also PostgreSQL) to return the current date and time. ‘SYSDATE()‘ is another MySQL function for the same purpose.
- ‘SYSDATEONLY()‘: This is not a standard SQL function.

Given the context of common SQL functions and typical database systems, ‘NOW()‘ is the correct choice for MySQL.

Final Answer : “NOW()”

Answer: (C)

Q7.

Solution

Concept: Nested SQL string functions, specifically ‘TRIM()‘ for removing leading/trailing spaces and ‘LENGTH()‘ for determining string length.

Solution: The SQL query ‘SELECT LENGTH(TRIM(’ CUET ’));‘ involves nested functions, which are evaluated from the innermost function outwards.
1. ‘TRIM(’ CUET ’)‘: This function removes both leading (spaces before the text) and trailing (spaces after the text) whitespace from a string.
- The string ‘’ CUET ’‘ has two leading spaces and two trailing spaces.
- Result of ‘TRIM(’ CUET ’)‘: ’CUET’
2. ‘LENGTH(’CUET’)‘: This function returns the number of characters in the resulting string.
- The string ’CUET’ has 4 characters.
- Result of ‘LENGTH(’CUET’)‘: 4

Therefore, the final output of the query is 4.

Final Answer : “4”

Answer: (C)

Q8.

Solution

Concept: SQL mathematical functions for rounding and truncation.

Solution: We need the function that returns the largest integer value less than or equal to a given number, which is effectively rounding down to the nearest integer.
- ‘CEIL()‘ (or ‘CEILING()‘): Returns the smallest integer greater than or equal to the given number (rounds up). For example, ‘CEIL(5.7)‘ is 6, ‘CEIL(5.0)‘ is 5, ‘CEIL(-5.7)‘ is -5.
- ‘FLOOR()‘: Returns the largest integer value less than or equal to the given number (rounds down). For example, ‘FLOOR(5.7)‘ is 5, ‘FLOOR(5.0)‘ is 5, ‘FLOOR(-5.7)‘ is -6.
- ‘ROUND()‘: Rounds a number to the nearest integer or to a specified number of decimal places. For example, ‘ROUND(5.7)‘ is 6, ‘ROUND(5.2)‘ is 5.
- ‘INT()‘: This is not a standard SQL function for this purpose; ‘CAST(number AS INT)‘ might truncate the decimal part, but ‘FLOOR()‘ specifically implements "largest integer less than or equal to".

Therefore, ‘FLOOR()‘ is the correct function.

Final Answer : “FLOOR()”

Answer: (B)

Q9.

Solution

Concept: Basic operations in Relational Algebra.

Solution: Relational Algebra is a procedural query language used to process a relation and return another relation. Let's look at the given operations:
-Selection (σ): Selects a subset of rows (tuples) from a relation that satisfies a specified condition. It extracts horizontal subsets of tuples.
-Projection (π): Selects a subset of columns (attributes) from a relation. It extracts vertical subsets of attributes and eliminates duplicate tuples in the result.
-Join ($\bowtie$): Combines tuples from two relations based on a common attribute or a join condition.
-Union ($\cup$): Combines all tuples from two relations, provided they are union-compatible (same number of attributes and corresponding domains).

The operation used to extract specific columns from a relation is Projection.

Final Answer : "Projection"

Answer: (B)

Q10.

Solution

Concept: Database keys, specifically the role and purpose of a Foreign Key.

Solution: A Foreign Key is a field (or collection of fields) in one table that uniquely identifies a row of another table or the same table. It acts as a cross-reference between tables. Let's analyze the options:
-Uniquely identify records: This is the primary role of a Primary Key.
-Remove duplicate values: While primary keys and unique constraints prevent duplicates, removing them is a data cleaning task, not the primary purpose of a foreign key.
-Establish relationship between tables: This is the core function of a Foreign Key. It links records in two different tables, maintaining referential integrity and showing how data between tables is related. For example, a 'CustomerID' in an 'Orders' table could be a foreign key referencing the 'CustomerID' (primary key) in a 'Customers' table.
-Encrypt records: This relates to data security and encryption, not the structural purpose of a foreign key.

Therefore, a foreign key is primarily used to establish relationships between tables.

Final Answer : "Establish relationship between tables"

Answer: (C)

Q11.

Solution

Concept: Properties and characteristics of a Primary Key in a relational database.

Solution: A Primary Key is a special relational database column (or combination of columns) designated to uniquely identify all table records. Its essential properties are:
-Unique values: Each value in the primary key column(s) must be unique across all rows in the table to ensure distinct identification of records.
-Cannot contain NULL: A primary key column cannot have NULL values. This is known as the entity integrity rule.
-Identifies tuples uniquely: This is the fundamental purpose of a primary key; it serves as a unique identifier for each record (tuple) in the table.

Considering these properties:
- (A) Unique values: Associated.
- (B) Cannot contain NULL: Associated.
- (C) Can have duplicate values: NOT Associated. A primary key must contain unique values. - (D) Identifies tuples uniquely: Associated.

Thus, "Can have duplicate values" is NOT associated with a primary key.

Final Answer : “Can have duplicate values”

Answer: (C)

Q12.

Solution

Concept: Relational Algebra operations for combining relations.

Solution: We are looking for the relational algebra operation that combines tuples from two relations based on a related attribute.

-Cartesian Product ($\times$): Combines every tuple from the first relation with every tuple from the second relation. It generates all possible combinations without any initial filtering based on attributes.

-Join ($\bowtie$): This operation explicitly combines tuples from two relations based on a common attribute or a specified join condition. For instance, a natural join combines tuples that have equal values on their common attributes.

-Difference ($-$): Returns the tuples that are present in the first relation but not in the second relation.

-Projection (π): Extracts specific columns from a relation, not combines tuples from two relations.

Therefore, the Join operation is used to combine tuples from two relations based on a related attribute.

Final Answer : "Join"

Answer: (B)

Q13.

Solution

Concept: Understanding different network topologies and their characteristics, particularly in terms of troubleshooting and fault isolation.

Solution: Let's analyze the common network topologies:
- Star Topology: All nodes are connected to a central hub or switch. If a cable to a node fails, only that node is affected. If the central device fails, the entire network goes down.
Troubleshooting is generally easy as problems are isolated to individual connections or the central device.
- Bus Topology: All nodes are connected to a single shared communication line (bus). A break in the main cable can bring down the entire segment, and locating the exact point of failure can be difficult if not properly managed.
- Ring Topology: Nodes are connected in a closed loop, where data travels in one direction.
Each node typically regenerates the signal. If a single connection or node in the ring fails, the entire ring can be broken, preventing communication for all nodes. Identifying the exact point of failure in a large ring can be challenging because a break anywhere in the loop affects connectivity across the whole loop. This fits the description of "difficult to troubleshoot because all nodes are connected in a closed loop."
- Tree Topology: A hierarchical structure combining characteristics of star and bus topologies. It's generally easier to troubleshoot than a bus or ring due to its segmented nature.

Given that the problem specifically mentions a "closed loop" and difficulty in troubleshooting due to the interconnected nature, the Ring topology is the most fitting answer.

Final Answer : "Ring"

Answer: (C)

Q14.

Solution

Concept: Understanding the OSI model and the function of various networking devices at different layers.

Solution: We need to identify the device that operates at the Data Link Layer (Layer 2) and uses MAC addresses for forwarding frames.
- Router: Operates at the Network Layer (Layer 3) and uses IP addresses to forward packets between different networks.
- Switch: Operates at the Data Link Layer (Layer 2). It learns MAC addresses of devices connected to its ports and uses these MAC addresses to forward data frames only to the intended recipient, reducing network traffic.
- Gateway: Can operate at various layers, often functioning at the Application Layer, connecting two dissimilar networks that use different protocols.
- Repeater: Operates at the Physical Layer (Layer 1). It simply amplifies or regenerates the signal to extend the network segment, without any knowledge of MAC or IP addresses.

Based on these functions, a Switch is the device that operates at the Data Link Layer and uses MAC addresses for forwarding frames.

Final Answer : "Switch"

Answer: (B)

Q15.

Solution

Concept: Distinguishing between MAC (Media Access Control) addresses and IP (Internet Protocol) addresses in networking.

Solution: Let's evaluate each statement to differentiate MAC and IP addresses:
- MAC Address: A physical hardware address uniquely assigned to a network interface card (NIC) by its manufacturer. It's a Layer 2 (Data Link Layer) address. It generally does not change and is hardcoded into the hardware.
- IP Address: A logical address assigned to a device on a network. It's a Layer 3 (Network Layer) address. IP addresses can be static (fixed) or dynamic (assigned by a DHCP server, and thus can change frequently, especially in dynamic environments).

Now let's check the options:
- (A) MAC address changes frequently while IP address never changes: This is incorrect. MAC addresses are generally static, while IP addresses (especially dynamic ones) can change frequently.
- (B) MAC address identifies hardware while IP address identifies logical network location: This is correct. MAC addresses are hardware-specific, identifying the physical network adapter. IP addresses identify a device's logical position within a network, allowing it to communicate across different networks.
- (C) Both are assigned by ISP only: This is incorrect. MAC addresses are assigned by the manufacturer. IP addresses can be assigned by an ISP (for public IPs), a router's DHCP server (for private IPs), or manually.
- (D) Both operate only at Application Layer: This is incorrect. MAC addresses operate at the Data Link Layer, and IP addresses operate at the Network Layer. The Application Layer is Layer 7.

Therefore, the statement that correctly differentiates them is that MAC address identifies hardware while IP address identifies logical network location.

Final Answer : "MAC address identifies hardware while IP address identifies logical network location"

Answer: (B)

Q16.

Solution

Concept: Basic data input/output operations in the Pandas library for Python.

Solution: The Pandas library is a powerful tool for data manipulation and analysis in Python. One of its most common uses is reading data from various file formats into a DataFrame.
- ‘pd.loadcsv()‘: This is not a standard Pandas function for reading CSV files.
- ‘pd.read_csv()‘: This is the correct and widely used function in Pandas to read data from a Comma Separated Values (CSV) file and create a DataFrame from it.
- ‘pd.import_csv()‘: This is not a standard Pandas function.
- ‘pd.fetch_csv()‘: This is not a standard Pandas function.

Thus, ‘pd.read_csv()‘ is the correct function.

Final Answer : “pd.read_csv()”

Answer: (B)

Q17.

Solution

Concept: Understanding DataFrame attributes in Pandas, specifically ‘shape‘ for obtaining dimensions.

Solution: In Pandas, a DataFrame is a two-dimensional labeled data structure. The ‘shape‘ attribute of a DataFrame is used to get the dimensions of the DataFrame. It returns a tuple ‘(number_of_rows, number_of_columns)‘.
Given a DataFrame ‘df‘ with 12 rows and 5 columns:
- The number of rows is 12.
- The number of columns is 5.
Therefore, ‘df.shape‘ will return the tuple ‘(12, 5)‘.

Let’s check the options:
- (A) (5,12): Incorrect, this reverses rows and columns.
- (B) [12,5]: Incorrect, ‘shape‘ returns a tuple, not a list.
- (C) (12,5): Correct, representing (rows, columns).
- (D) 60: Incorrect, this is the total number of elements (12 * 5), not the shape.

Final Answer : “(12,5)”

Answer: (C)

Q18.

Solution

Concept: Pandas DataFrame indexing and slicing using ‘iloc‘.

Solution: In Pandas, ‘iloc‘ is used for integer-location based indexing, meaning you select rows and columns by their integer position (from 0 to N-1). It works similarly to standard Python list slicing, where the start index is inclusive and the stop index is exclusive.
To select rows from index 2 to 5 (inclusive of rows 2, 3, 4, and 5):
- The start index should be 2.
- Since the stop index is exclusive, if we want to include row 5, the stop index must be 6. - ‘df.iloc[start:stop]‘ will select rows with integer indices from ‘start‘ up to, but not including, ‘stop‘.
- So, ‘df.iloc[2:6]‘ will select rows at index 2, 3, 4, and 5.

Let’s check the options:
- (A) ‘df.iloc[2:5]‘: This would select rows at integer indices 2, 3, and 4 (excluding 5). Incorrect.
- (B) ‘df.iloc[2:6]‘: This correctly selects rows at integer indices 2, 3, 4, and 5 (excluding 6). Correct.
- (C) ‘df.iloc[1:5]‘: This would select rows at integer indices 1, 2, 3, and 4. Incorrect.
- (D) ‘df.iloc[2,5]‘: This syntax would select a single element at row index 2 and column index 5, not a range of rows. Incorrect.

Final Answer : “df.iloc[2:6]”

Answer: (B)

Q19.

Solution

Concept: Understanding DataFrame attributes in Pandas, specifically for accessing column labels.

Solution: A Pandas DataFrame consists of rows and columns. We need an attribute that returns the labels (names) of the columns.
- ‘df.axes()‘: This is incorrect syntax. The ‘df.axes‘ attribute (without parentheses) returns a list containing the row labels (index) and the column labels. While it contains the column labels, ‘df.columns‘ is the more direct and common attribute for just the column names.
- ‘df.columns‘: This is the correct attribute. It returns an ‘Index‘ object containing the labels of the DataFrame’s columns.
- ‘df.rows‘: There is no standard ‘df.rows‘ attribute in Pandas. To get row labels, you would use ‘df.index‘.
- ‘df.dimension‘: There is no ‘df.dimension‘ attribute. ‘df.ndim‘ returns the number of dimensions (2 for a DataFrame).

Therefore, ‘df.columns‘ is the attribute that returns the column labels of a DataFrame.

Final Answer : “df.columns”

Answer: (B)

Q20.

Solution

Concept: Handling missing data (NaN values) in Pandas DataFrames.

Solution: Missing data is a common issue in real-world datasets, and Pandas provides functions to manage it. We are looking for the command that removes rows containing missing values.
- ‘dropna()‘: This is the primary Pandas function used to remove missing values. By default, ‘dropna()‘ drops any row that contains at least one ‘NaN‘ (Not a Number) value. It can also be configured to drop columns or to drop rows/columns only if all values are missing.
- ‘fillna()‘: This function is used to replace ‘NaN‘ values with a specified value (e.g., 0, mean, median, or a forward/backward fill). It does not remove rows.
- ‘remove_null()‘: This is not a standard Pandas function for handling missing data.
- ‘clearna()‘: This is not a standard Pandas function for handling missing data.

Hence, ‘dropna()‘ is the correct command to remove rows containing missing values.

Final Answer : “dropna()”

Answer: (A)

Q21.

Solution

Concept: Understanding data structures in Pandas, specifically what type of object is returned when a single column is selected from a DataFrame.

Solution: In Pandas, a DataFrame is a two-dimensional labeled data structure with columns that can be of different types. A Series, on the other hand, is a one-dimensional labeled array.
When you select a single column from a DataFrame using singlebracket notation (e.g., ‘df[’column_name’]‘ or ‘df.column_name‘), Pandas returns a Series object. If you select multiple columns (e.g., ‘df[[’col1’, ’col2’]]‘), it returns a DataFrame.
Therefore, ‘df[’Marks’]‘ will return a Pandas Series, where ’Marks’ is the label (name) of the Series.

Final Answer : “Series”

Answer: (B)

Q22.

Solution

Concept: Basic DataFrame inspection methods in Pandas.

Solution: Pandas DataFrames provide convenient methods to quickly inspect the data.
- ‘head()‘: This method returns the first ‘n‘ rows of the DataFrame. By default, if no argument is passed, it returns the first 5 rows.
- ‘top()‘: Not a standard Pandas method for this purpose.
- ‘first()‘: Not a standard Pandas method for this purpose.
- ‘preview()‘: Not a standard Pandas method for this purpose.

Thus, ‘head()‘ is the correct method to display the first five rows of a DataFrame.

Final Answer : “head()”

Answer: (C)

Q23.

Solution

Concept: Data output operations in Pandas, specifically writing a DataFrame to a CSV file.

Solution: Pandas DataFrames have a suite of methods prefixed with ‘to_‘ for writing data to various file formats or other data structures.
- ‘save_csv()‘: Not a standard Pandas function for this purpose.
- ‘export_csv()‘: Not a standard Pandas function for this purpose.
- ‘to_csv()‘: This is the correct and standard Pandas method to write a DataFrame to a Comma Separated Values (CSV) file. You can specify the file path, whether to include the index, header, etc.
- ‘write_csv()‘: Not a standard Pandas function for this purpose.

Therefore, ‘to_csv()‘ is the function used to write a DataFrame into a CSV file.

Final Answer : “to_csv()”

Answer: (C)

Q24.

Solution

Concept: Accessing elements in a Pandas Series using its labels (index).

Solution: A Pandas Series is a one-dimensional labeled array. When you create a Series with a custom index, you can access individual elements using these labels.
Given ‘s = pd.Series([10,20,30], index=[’a’,’b’,’c’])‘:
- The value ‘10‘ is associated with the index label ‘’a’‘.
- The value ‘20‘ is associated with the index label ‘’b’‘.
- The value ‘30‘ is associated with the index label ‘’c’‘.

When you call ‘s[’b’]‘, Pandas looks for the element with the label ’b’ and returns its corresponding value. In this case, the value is 20.

Final Answer : “20”

Answer: (C)

Q25.

Solution

Concept: Customizing DataFrame creation from CSV files using ‘pd.read_csv()‘.

Solution: The ‘pd.read_csv()‘ function is highly configurable. One common requirement is to designate a specific column from the CSV file as the index (row labels) of the resulting DataFrame, instead of using the default 0-based integer index.
- ‘key_col‘: This is not a parameter of ‘read_csv()‘.
- ‘header‘: This parameter specifies which row (or rows) should be used as the column names (header) of the DataFrame. It does not control the row index.
- ‘index_col‘: This is the correct parameter. You can pass the name of the column (as a string) or its 0-based integer position to ‘index_col‘ to specify which column should be used as the DataFrame’s index.
- ‘indexer‘: This is not a parameter of ‘read_csv()‘.

Therefore, ‘index_col‘ is the parameter used to make a specific column the index.

Final Answer : “index_col”

Answer: (C)

Q26.

Solution

Concept: Sorting DataFrames in Pandas based on column values.

Solution: To sort a Pandas DataFrame by the values in one or more columns, the ‘sort_values()‘ method is used.
- The ‘by‘ parameter specifies the column(s) by which to sort. In this case, ‘by=’Marks’‘.
- The ‘ascending‘ parameter controls the sort order. ‘ascending=True‘ (default) sorts in ascending order, while ‘ascending=False‘ sorts in descending order.

Let’s evaluate the options:
- (A) ‘df.sort_values(’Marks’, ascending=False)‘: This correctly sorts the DataFrame by the ’Marks’ column in descending order.
- (B) ‘df.sort(’Marks’,’DESC’)‘: ‘sort()‘ is an older, deprecated method. The syntax ‘’DESC’‘ is not used with ‘ascending‘ parameter.
- (C) ‘df.arrange(Marks)‘: ‘arrange()‘ is a function commonly found in R’s dplyr package, not standard Pandas.
- (D) ‘df.order(Marks)‘: ‘order()‘ is an older, deprecated method.

Final Answer : “df.sort_values(’Marks’, ascending=False)”

Answer: (A)

Q27.

Solution

Concept: Understanding the ‘axis‘ parameter in Pandas operations.

Solution: Many Pandas DataFrame methods that involve applying a function or performing an operation across a DataFrame have an ‘axis‘ parameter to specify whether the operation should be performed along rows or columns.
- ‘axis=0‘: Refers to the index (rows). Operations are applied column-wise. For example, ‘df.sum(axis=0)‘ would sum values down each column.
- ‘axis=1‘: Refers to the columns. Operations are applied row-wise. For example, ‘df.sum(axis=1)‘ would sum values across each row.

Therefore, ‘axis=1‘ represents columns in most Pandas operations.

Final Answer : “Columns”

Answer: (B)

Q28.

Solution

Concept: Combining DataFrames in Pandas, specifically vertical concatenation.

Solution: Pandas provides powerful functions for combining DataFrames.
- ‘append()‘ / ‘concat()‘: To combine two DataFrames vertically (stacking one DataFrame’s rows on top of another’s), the ‘pd.concat()‘ function is used, typically with ‘axis=0‘ (which is the default for vertical concatenation). The ‘append()‘ method (e.g., ‘df1.append(df2)‘) is also used for vertical concatenation and is essentially a specialized call to ‘concat()‘.
- ‘merge()‘: Used to combine DataFrames horizontally based on common column values, similar to SQL joins.
- ‘joincol()‘: Not a standard Pandas function for combining DataFrames.
- ‘intersect()‘: Not a standard Pandas function for combining DataFrames. (Set operations are possible but not with this name for DataFrames).

Thus, ‘append() / concat()‘ are the primary functions for combining DataFrames vertically.

Final Answer : “append() / concat()”

Answer: (A)

Q29.

Solution

Concept: Generating descriptive statistics for DataFrames in Pandas.

Solution: Data analysis often begins with obtaining a quick statistical summary of the dataset.
- ‘summary()‘: Not a standard Pandas method.
- ‘describe()‘: This is the correct and widely used Pandas method. When called on a DataFrame, it generates descriptive statistics that summarize the central tendency, dispersion, and shape of the distribution of a DataFrame’s numeric columns (by default). This includes count, mean, std (standard deviation), min, 25th percentile (Q1), 50th percentile (median), 75th percentile (Q3), and max.
- ‘statistics()‘: Not a standard Pandas method.
- ‘analyze()‘: Not a standard Pandas method.

Therefore, ‘describe()‘ is the method used to obtain a statistical summary of numeric columns.

Final Answer : “describe()”

Answer: (B)

Q30.

Solution

Concept: Creating a Pandas DataFrame from common Python data structures, specifically a dictionary.

Solution: The ‘pd.DataFrame()‘ constructor is the primary way to create a DataFrame in Pandas. It can accept various types of input, including dictionaries. When a dictionary is passed, the keys typically become the column labels, and the values (which should be array-like structures like lists or Series) become the data for those columns.
- ‘pd.frame(d)‘: Incorrect capitalization and naming.
- ‘pd.DataFrame(d)‘: This is the correct syntax for creating a DataFrame from a dictionary ‘d‘.
- ‘pd.create(d)‘: Not a standard Pandas function for DataFrame creation.
- ‘DataFrame.make(d)‘: Incorrect syntax; ‘DataFrame‘ is a class, but ‘make()‘ is not its constructor or a standard factory method in this form.

Final Answer : “pd.DataFrame(d)”

Answer: (B)

Q31.

Solution

Concept: Distinguishing between different DataFrame indexing methods in Pandas.

Solution: Pandas DataFrames offer powerful and flexible indexing methods:
- ‘iloc‘: Stands for integer-location based indexing. It is used to select data by its integer position (from 0 to N-1) along the index.
- ‘loc‘: Stands for label-based indexing. It is primarily used to select data by its explicit index labels (for rows) and column labels (for columns).
- ‘indexloc‘: Not a standard Pandas indexing method.
- ‘atrow‘: Not a standard Pandas indexing method. (‘.at‘ is used for fast single-value label-based access).

Therefore, ‘loc‘ is the indexing method that is label-based.

Final Answer : “loc”

Answer: (B)

Q32.

Solution

Concept: Data structure inspection and metadata retrieval in the Pandas library.

Solution: A Pandas DataFrame is a two-dimensional, size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns).
The attribute `df.columns` is used to access the column labels of the DataFrame. It returns an `Index` object containing the names of all columns.
In Python, the built-in `len()` function is used to return the number of items in an object. When applied to the `df.columns` object, it counts how many column labels exist in that index.
Therefore, `len(df.columns)` gives the total count of columns in the DataFrame.
For example, if a DataFrame `df` is created from a CSV with headers "ID", "Name", and "Age", `df.columns` will be `['ID', 'Name', 'Age']` and its length will be 3.

Final Answer : “Number of columns”

Answer: (C)

Q33.

Solution

Concept: Filtering grouped data versus individual rows in SQL (WHERE vs. HAVING).

Solution: In SQL, the order of execution is crucial. When we use the `GROUP BY` clause, the database collapses multiple rows into summary rows (like finding the total sales per region).
The `WHERE` clause is processed **before** the data is grouped. It filters individual rows based on non-aggregate criteria.
The `HAVING` clause is processed **after** the `GROUP BY` clause. It is specifically designed to filter the "groups" created by the `GROUP BY` statement.
If you need to filter data based on an aggregate function (such as `COUNT`, `SUM`, or `AVG`), you cannot use `WHERE`; you must use `HAVING`.

Final Answer : "HAVING"

Answer: (B)

Q34.

Solution

Concept: Standard sorting mechanisms in Structured Query Language (SQL).

Solution: The `ORDER BY` clause is used to sort the result-set of a query. It can sort data in two directions:
Ascending (ASC): From lowest to highest (e.g., 1 to 10 or A to Z).
Descending (DESC): From highest to lowest (e.g., 10 to 1 or Z to A).
By default, most relational database management systems (like MySQL, PostgreSQL, and SQL Server) assume the user wants the data in increasing order if no specific keyword is provided. Thus, `ORDER BY marks` is functionally identical to `ORDER BY marks ASC`.

Final Answer : "Ascending"

Answer: (C)

Q35.

Solution

Concept: Comprehensive set logic in SQL Join operations.

Solution: SQL Joins allow users to retrieve data from multiple tables based on logical relationships.
INNER JOIN: Returns only the intersection (rows with matches in both tables).
LEFT JOIN: Returns everything from the left table and matching rows from the right.
RIGHT JOIN: Returns everything from the right table and matching rows from the left.
FULL OUTER JOIN: This join returns the **Union** of the two tables. It provides all records from Table A and Table B. If there is a match, the values are joined; if there is no match, the missing side is filled with `NULL` values. It ensures that no data from either table is omitted from the result set.

Final Answer : "FULL OUTER JOIN"

Answer: (C)

Q36.

Solution

Concept: Syntactic structure of aggregate queries in SQL.

Solution: To generate a count of employees for each department, we must follow a specific query structure:
Select the category: We need to see the `dept` name in the output.
Apply the aggregate: `COUNT(*)` counts the number of rows (employees) in each group.
Group the data: The `GROUP BY dept` clause tells the database to create a separate bucket for each unique department value.
Option (A) `SELECT dept, COUNT(*) FROM emp GROUP BY dept;` is the only query that correctly pairs the selected column with the grouping column and utilizes the correct syntax for an aggregate report.

Final Answer : "SELECT dept, COUNT(*) FROM emp GROUP BY dept;"

Answer: (A)

Q37.

Solution

Concept: The logical query processing phases of a SELECT statement.

Solution: In SQL, the syntax follows a predefined hierarchy. When a query involves grouping, the sequence is:
`FROM`: Identify the tables.
`WHERE`: Filter the raw rows.
`GROUP BY`: Categorize the remaining rows into groups.
`HAVING`: Filter those groups based on aggregate results.
`SELECT`: Determine which columns/aggregates to display.
`ORDER BY`: Sort the final output.
Since `HAVING` acts upon the groups created by `GROUP BY`, it is the clause that must appear after the `GROUP BY` clause to evaluate conditions on those summarized groups.

Final Answer : "HAVING"

Answer: (C)

Q38.

Solution

Concept: Multiplicative relationships between tables (Cartesian Product).

Solution: A Cartesian Product is a mathematical operation that returns a set from two sets. In database terms, it means every single row from Table A is combined with every single row from Table B. If Table A has n n rows and Table B has m m rows, the result has n × m n×m rows. In SQL, the `CROSS JOIN` keyword is explicitly used to produce this effect.
While an `INNER JOIN` (or a comma-separated join) without a join condition (`ON` or `WHERE` clause) also results in a Cartesian Product, `CROSS JOIN` is the standard name for this specific join type.

Final Answer : "CROSS JOIN"

Answer: (C)

Q39.

Solution

Concept: The "Three-Valued Logic" and treatment of NULLs in SQL calculations.

Solution: In SQL, `NULL` represents a missing or unknown value. If aggregate functions included `NULL`s as zero, it would mathematically skew results like averages. Therefore:
`SUM(column)` ignores `NULL` entries and only totals known numbers.
`AVG(column)` calculates the mean by dividing the sum by the count of **non-null** values.
`COUNT(column_name)` only counts rows where that specific column contains a value.
Because all standard aggregate functions are designed to skip `NULL` values to ensure statistical accuracy, the answer is "All of these."

Final Answer : "All of these"

Answer: (D)

Q40.

Solution

Concept: Combined use of GROUP BY and HAVING for statistical filtering.

Solution: To solve this requirement, the SQL engine must perform several steps:
Gather all rows from the `emp` table.
Divide those rows into groups based on the `dept` column (`GROUP BY dept`).
For each group, calculate the average salary (`AVG(salary)`).
Apply a filter to keep only those groups where the calculated average is greater than 50,000 (`HAVING AVG(salary) > 50000`).
Option (B) is the only statement that correctly places the aggregate condition in the `HAVING` clause, which is mandatory because aggregate functions are prohibited in the `WHERE` clause.

Final Answer : "SELECT dept, AVG(salary) FROM emp GROUP BY dept HAVING AVG(salary)>50000;"

Answer: (B)

Q41.

Solution

Concept: Result set purification and redundancy control in SQL.

Solution: Databases often contain multiple rows with identical data in certain columns (e.g., many employees in the "Sales" department). If you want a list of which departments exist without seeing "Sales" repeated a hundred times, you use the `DISTINCT` keyword.
`SELECT DISTINCT` works by comparing the rows in the final result set.
If two rows are identical across all selected columns, only one is kept in the output.
It is placed immediately after `SELECT` and before the column names.

Final Answer : "DISTINCT"

Answer: (B)

Q42.

Solution

Concept: Graphical representation of data distributions in Python.

Solution: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations.
`plt.bar()`: Used for bar charts (comparing discrete categories).
`plt.scatter()`: Used for scatter plots (showing correlation between two variables).
`plt.hist()`: This is the specific function for creating a Histogram. A histogram takes a large set of data and groups it into "bins" (ranges) to show the underlying frequency distribution (shape) of a set of continuous data.
There is no standard function called `plt.plotpie()`; the standard for pie charts is `plt.pie()`.

Final Answer : "plt.hist()"

Answer: (C)

Q43.

Solution

Concept: Mapping data values to descriptive text in Matplotlib visualizations.

Solution: When generating a pie chart using `plt.pie(x)`, the chart shows slices of different sizes. To identify what each slice represents, we use the `labels` parameter.
The `labels` argument takes a list of strings (e.g., `['Apples', 'Bananas', 'Cherries']`). These strings are then printed next to their respective wedges.
Other parameters exist like `autopct` (for percentages) or `colors`, but for the specific purpose of displaying names/tags for each slice, `labels` is the correct keyword.

Final Answer : "labels"

Answer: (C)

Q44.

Solution

Concept: Plot annotation and legend management in Data Science.

Solution: A legend is an essential part of a graph that identifies what different lines, colors, or markers represent.
When plotting data, you typically provide a label: `plt.plot(x, y, label='Growth')`.
Simply providing the label is not enough; Matplotlib will not show the legend box by default.
You must explicitly call the `plt.legend()` function to instruct Matplotlib to create the box and place the labels inside it on the plot area.

Final Answer : "plt.legend()"

Answer: (B)

Q45.

Solution

Concept: Statistical tools for continuous vs. discrete data visualization.

Solution: Data is often classified as either discrete (countable, like number of cars) or continuous (measurable, like time or height).
Bar Graphs are best for discrete categories.
Pie Charts are used to show parts of a whole.
Histograms are specifically designed to represent the distribution of **continuous data**. By dividing the data into ranges (bins) and plotting the frequency of occurrences within those ranges, a histogram provides a visual interpretation of numerical data distribution (e.g., Normal Distribution).

Final Answer : "Histogram"

Answer: (B)

Q46.

Solution

Concept: Axes labeling and plot documentation in Matplotlib.

Solution: Effective data visualization requires clear labeling so the viewer knows what the units and variables are.
`plt.xlabel()`: Sets the label for the horizontal X-axis.
`plt.ylabel()`: Sets the label for the vertical Y-axis.
`plt.title()`: Adds a main heading to the top of the chart.
Using these functions makes the chart professional and readable. Options like `xname` or `labelx` do not exist in the Matplotlib API.

Final Answer : "plt.xlabel()"

Answer: (B)

Q47.

Solution

Concept: Social Engineering and Cyber Security threats.

Solution: Cybercriminals use various methods to exploit users.
Phishing: This involves sending fraudulent communications (emails, SMS, or fake websites) that appear to come from a reputable source. The goal is to trick the user into typing in sensitive information like passwords or bank details.
Hacking: Gaining unauthorized access to a computer system.
Spoofing: Disguising an identity (like a fake caller ID or IP address).
Sniffing: Monitoring or capturing data packets as they pass through a network.
Because the question specifically mentions "fraudulently obtaining sensitive information through fake websites," **Phishing** is the exact term.

Final Answer : "Phishing"

Answer: (B)

Q48.

Solution

Concept: Scope and classification of Intellectual Property Rights (IPR).

Solution: Intellectual Property Rights (IPR) are legal frameworks designed to protect the "creations of the mind." These rights allow creators and inventors to enjoy the commercial benefits of their work while preventing unauthorized use. IPR is a broad umbrella term that encompasses several distinct legal protections:

- **Copyright:** This protects original literary, dramatic, musical, and artistic works. In the digital age, this also extends to computer software, source code, and website content.
- **Patent:** This is a right granted to an inventor for a technical invention. It must be a new way of doing something or offer a new technical solution to a problem. Patents prevent others from making, using, or selling the invention without a license.
- **Trademark:** This protects signs, logos, symbols, or names used to distinguish the goods or services of one enterprise from those of others. It helps in brand identification and prevents consumer confusion.

Since Copyrights, Patents, and Trademarks are all primary components of Intellectual Property Rights, the correct answer is "All of these."

Final Answer : "All of these"

Answer: (D)

Q49.

Solution

Concept: Environmental impact and definition of Electronic Waste (E-waste).

Solution: E-waste, short for Electronic Waste, describes electrical or electronic devices that are discarded, surplus, obsolete, or broken. As technology evolves rapidly, the lifespan of gadgets decreases, leading to a massive increase in e-waste.

- **Primary Examples:** Discarded items such as old mobile phones, laptops, desktop computers, printed circuit boards (PCBs), cathode ray tubes (CRTs), monitors, and even household appliances like refrigerators and microwaves.
- **Why it is unique:** Unlike biodegradable waste (like food) or agricultural waste, e-waste contains hazardous components like lead, mercury, cadmium, and brominated flame retardants. If not disposed of correctly, these toxic substances can leach into the soil and groundwater.
- **Categorization:** It specifically refers to discarded electronic devices and their internal components. It does not relate to the food industry or biological materials.

Final Answer : "Discarded electronic devices and components"

Answer: (C)

Q50.

Solution

Concept: Digital citizenship and the persistence of online data (Digital Footprint).

Solution: A digital footprint refers to the trail, traces, or "footprints" of data that people leave behind when they use the internet. This trail is often permanent and can be searched or tracked by others. It is divided into two main categories:

- **Active Digital Footprint:** This is data that a user intentionally leaves behind. For example, posting on social media (Facebook/Instagram), sending emails, or filling out online forms.
- **Passive Digital Footprint:** This is data collected without the user's active participation. For example, websites tracking your IP address, your browsing history stored via cookies, or apps accessing your GPS location in the background.

Essentially, any activity performed online—from searching on Google to liking a photo—contributes to this "online trace." It is not a physical object like a hard disk or a biometric scanner, but a digital record of behavior.

Final Answer : "Online trace left by user activities on the internet"

Answer: (B)

Answer Key

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