SOLUTIONS

Sample Question Paper-5

SECTION-A

1. True

Why? In Pandas, `DataFrame.rename()` allows changing row indexes and column labels using a dictionary. Example: `df.rename(columns={"old":"new"})`.

Why not? "False" would be wrong because `rename()` indeed supports renaming columns and rows.

2. **(b)** 8.76

Why? `TRUNCATE(num, d)` removes digits after d decimal places without rounding. So, 8.7654 becomes 8.76.

Why not?

(a) 8.77 □ this is rounding, not truncating.

(c) 8.75 □ not mathematically related to truncation here.

(d) 8.7 □ this is truncation to 1 decimal, not 2.

3. **(b)** *Do not open unknown email attachments.* – as it may contain any malicious program.

Why? Other options can be cautiously used.

4. **(c)** `df.shape`

Why? `.shape` returns a tuple `(rows, columns).`

Why not?

(a) **df.shape()** □ shape is not a method, so parentheses are wrong.

(b) **df.len()** □ doesn't exist in Pandas.

(d) **df.size** □ returns total number of elements, not rows/columns separately.

5. **(b)** Modem

Why? Modem converts digital data to analogue (and vice versa) for internet over telephone/cable lines.

Why not?

(a) Repeater □ only regenerates the signal in the LAN.

(c) Bridge □ connects two LAN segments.

(d) Router □ directs packets between networks, but internet access from ISP usually first needs modem.

6. **(b)** MOD(10, 3)

Why? `MOD(a,b)` returns the remainder after division.

Why not?

(a) 10 / 3 □ division result, not remainder.

(c) ROUND() □ rounds numbers.

(d) CEIL() □ rounds up to nearest integer.

7. **(d)** Trademark protects brand symbols/names.

Why not?

(a) Copyright □ protects creative works, not functional design.

(b) Patent □ protects inventions/ideas.

(c) Prototype – It's not an IPRTrademark

8. **(b)** 0, 1, 2

Why? By default, Pandas assigns an integer index starting from 0.

Why not?

(a) 1, 2, 3 □ would require a custom index.

(c) 'a', 'b', 'c' □ also requires a custom index.

(d) Random numbers □ Pandas doesn't assign random indexes.

9. **(a)** It can have NULL values

Why? Primary keys cannot contain NULL.

Why not?

(b) Must be unique □ true.

(c) Can be multiple columns (composite key) □ true.

(d) Used to uniquely identify records □ true.

10. **(b)** SMTP

Why? Simple Mail Transfer Protocol sends outgoing emails.

Why not?

(a) FTP □ file transfer.

(c) VoIP □ voice over IP.

(d) POP □ retrieves incoming emails, not sends.

11. **(a)** Minimum marks

Why? `MIN()` returns the smallest value.

Why not?

(b) Maximum □ MAX().

(c) Average □ AVG().

(d) Total □ SUM().

12. **(a)** `isnull()`

Why? `.isnull()` returns True for NaN.

Why not?

(b) isempty() □ not a Pandas function.

(c) isNaN() □ NumPy equivalent, but not Series method.

(d) nullcheck() □ invalid.

13. **(b)** Cyber crimes and electronic commerce

Why? The IT Act, 2000 provides legal framework for cyber crimes, e-commerce, digital signatures, and electronic records in India.

Why not?

(a) Agriculture is not related to IT Act.

(c) Protection of traditional knowledge comes under IPR laws, not IT Act.

(d) Copyright is covered by the Copyright Act, 1957, not IT Act.

14. **(b)** HAVING

Why? HAVING filters aggregate results.

Why not?

(a) ORDER BY □ sorts.

(c) WHERE □ filters before grouping.

(d) LIMIT □ restricts the rows count.

15. **(b)** Last row

Why? Negative index −1 refers to the last element.

Why not?

(a) First row □ index 0.

(c) Error □ iloc accepts negative indexes.

(d) Second last □ index −2.

16. **(a)** Bus

Why? Bus topology uses a single backbone cable and minimal cabling.

Why not?

(b) Star □ more cable needed.

(c) Mesh □ most cable required.

(d) Hybrid □ depends on combination.

17. **(a)** ATAB

Why? Starts at 2nd character ('A'), takes 4 chars □ "ATAB".

Why not?

(b) DATA □ would be SUBSTR(...,1,4).

(c) DATAB □ would be length 5.

(d) TABA □ would require starting from 3rd char.

18. **(b)** List of column labels

Why? `.columns` is an Index object containing column names.

Why not?

(a) Number of rows □ use len(df).

(c) Column values □ that's df['col'].

(d) Index names □ .index.names.

19. **(b)** COUNT()

Why? COUNT works on any type to count rows.

Why not?

(a) SUM □ numeric only.

(c) AVG □ numeric only.

(d) MOD □ numeric only.

20. **(d)** A false, R true

21. **(d)** A false, R true

SECTION-B

22. **(a)** **Method 1:** From a List

Feature	Series	DataFrame
Structure	1D labelled array	2D labelled data structure (like a table)
Data Format	Similar to a single column	Collection of multiple Series (columns)
Axis	Only one axis (index)	Two axes (rows and columns)
Example Code	pd.Series([10, 20, 30])	pd.DataFrame({'A': [10, 20], 'B': [30, 40]})

OR

(b) **Open-source Python Library:** A freely available package whose source code is open to all for use, modification and distribution.

NumPy:

- Stands for **Numerical Python**
- Helps perform fast mathematical operations on large arrays and matrices
- Useful for statistical analysis, linear algebra, etc.

Example: np.mean([10, 20, 30]) □ 20

Pandas:

- Stands for **Python Data Analysis Library**
- Used to handle structured data using **DataFrames** and **Series**
- Supports data cleaning, analysis, filtering and grouping

Example: df['Sales'].mean() for average sales

23. Creative Commons (CC) Licences:

These are legal tools that allow creators to grant certain usage rights to the public while retaining others.

Benefits for creators:

1. **Control:** Creators can decide how others can use their work (e.g., allow reuse but not commercial use).
2. **Wider Reach:** Easier distribution leads to better visibility and collaboration.
3. **Attribution:** Ensures proper credit is given to the original author.

24. **(i)** SELECT SUBSTRING('Machine Learning with SQL', 9, 8);

(ii) SELECT INSTR('Machine Learning with SQL', 'SQL');

25. **(a) URL (Uniform Resource Locator):** It is the complete web address used to access a specific resource on the Internet.

Example:

- URL: https://www.example.com/page.html
- Domain Name: www.example.com

Difference:

- A domain name is part of a URL.
- URL gives full path including domain, protocol and file.

OR

(b) Website: A collection of interlinked **webpages** hosted under a domain name.

Webpage: A single document (like home, about, contact) on the internet.

Difference with Example:

- Website: www.cbsenotes.com
- Webpage: www.cbsenotes.com/contact.htm

26. **Composite Key:** A primary key formed by combining two or more columns to uniquely identify a row in a table.

Use Case: When no single column is unique by itself.

Example: In a table ENROLLMENT(StudentID, CourseID), combination is unique, not individually.

27. **E-waste:** Discarded electronic devices like old mobiles, computers, batteries, etc.

Management methods:

1. **Recycling:** Reuse parts or recover metals.
2. **Proper disposal:** Use certified e-waste collection centres

25. **(a)**

```
import matplotlib.pyplot as plt
x = ['Jan', 'Feb', 'Mar']
y = [10, 15, 12]
plt.bar(x, y)
plt.xlabel('Months')
plt.show()
```

OR

(b)

```
import pandas as pd
temps = [40, 35, 38]
cities = ['Delhi', 'Mumbai', 'Kolkata']
s = pd.Series(temps, index=cities)
print(s)
```

SECTION-C

29. **(i)** **Throwing electronic waste like mobile phones into household trash can lead to soil and water pollution** because such devices contain hazardous materials like lead, mercury and cadmium. These substances can seep into the ground and contaminate underground water sources.

(ii) **E-waste should be handed over to authorised e-waste recycling centres or collection points**. Many cities have designated drop zones or recycling drives where electronic devices can be safely disposed of.

(iii) **Recycling helps recover valuable materials like gold, copper and plastics**, which reduces the need for mining and conserves natural resources. It also prevents toxic substances from entering the ecosystem, thus protecting both human health and the environment.

30. **(a)**

```
import pandas as pd
books = [
{"Title": "Python 101", "Author": "John"},
{"Title": "Data World", "Author": "Aarti"},
{"Title": "AI Basics", "Author": "Vikram"},
{"Title": "SQL Master", "Author": "Neha"}
]
df = pd.DataFrame(books)
print(df)
```

OR

(b)

```
import pandas as pd
values = ["Guido", "Gosling", "Bjarne"]
indices = ["Python", "Java", "C++"]
s = pd.Series(values, index=indices)
print(s)
```

31.

```
CREATE TABLE STUDENTS (
StudentID NUMERIC PRIMARY KEY,
FirstName VARCHAR(20),
LastName VARCHAR(10),
DateOfBirth DATE,
Percentage FLOAT(10,2)
);
INSERT INTO STUDENTS
VALUES (1, 'Supriya', 'Singh', '2010-08-18', 75.5);
```

34. **(a)** **(i)** SELECT TITLE, AUTHOR FROM BOOKS WHERE PRICE > 500;

(ii) SELECT ISSUE.STUDENT_NAME, BOOKS.TITLE FROM ISSUE JOIN BOOKS ON ISSUE.BOOK_ID = BOOKS.BOOK_ID WHERE ISSUE.RETURNED = 'NO';

(iii) SELECT TITLE, PRICE FROM BOOKS ORDER BY PRICE DESC;

OR

(b) **(i)** SELECT CUSTOMER_NAME FROM CUSTOMERS WHERE CUSTOMER_ID IN (SELECT CUSTOMER_ID FROM ORDERS GROUP BY CUSTOMER_ID HAVING COUNT(*) > 1);

(ii) SELECT ORDER_ID, CUSTOMER_NAME, PRODUCT, ORDER_DATE FROM ORDERS JOIN CUSTOMERS ON ORDERS.CUSTOMER_ID = CUSTOMERS.

CUSTOMER_ID WHERE CITY IN ('Mumbai', 'Delhi');

(iii) SELECT COUNT(*) FROM ORDERS WHERE PRODUCT = 'Laptop';

SECTION-D

33. **(i)** matplotlib.pyplot

(ii) sales

(iii) ylabel

(iii) 'Mobile Sales in Q1'

34. **(a)** **(i)** SELECT UPPER(TITLE) FROM MOVIE;

(ii) SELECT MAX(RATING) FROM MOVIE;

(iii) SELECT TITLE, LENGTH(TITLE) FROM MOVIE;

(iv) SELECT MCODE, RATING FROM MOVIE ORDER BY RATING ASC;

OR

(b) **(i)** SELECT LOWER(NAME) FROM STUDENT;

(ii) SELECT COUNT(*) FROM STUDENT WHERE COURSE = 'BCA';

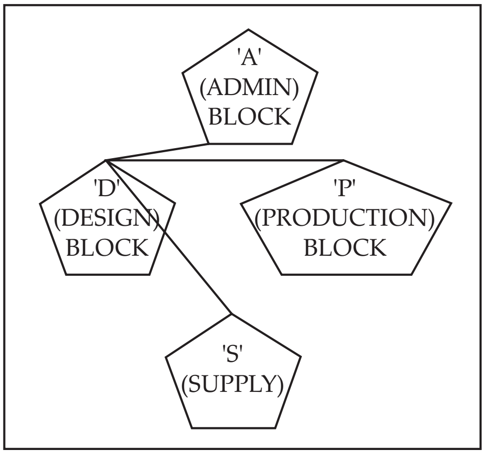
(iii) SELECT NAME, LENGTH(NAME) FROM STUDENT;

(iv) SELECT SID, GRADE FROM STUDENT ORDER BY GRADE DESC;

SECTION-E

35. **(i)** LAN (Local Area Network)

(ii) Star Topology



(iii) **(a)** Repeater – Repeater is required between block P to D as distance between them is 150m which will make signals weak.

(b) Hub/Switch – To be placed inside each block for connectivity

(iv) Video Conferencing software: Teams, Zoom, Skype, etc. Protocol of Video Conferencing software: VoIP.

(v) Since the block D is having the greatest number of computers, which is 80. So, as per 80- 20 rule the most suitable place to set up the server is block D.

36. **(i)** `print(df1.tail(3))`

(ii) `print(df1["Name"])`

(iii)
```
df1 = df1.drop("Marks", axis=1)
print(df1)
```

(iv) `print(df1.loc[1:3, "Name"])`

(v)
```
df1 = df1.rename(columns={"Name":
"StudentName"})
print(df1)
```

37. **(a)** **(i)** SELECT AVG(price) FROM Products;

(ii) SELECT LEFT(product_name, 4) FROM Products;

(iii) SELECT COUNT(*) FROM Products;

(iv) SELECT MAX(stock_quantity) FROM Products;

(v) SELECT UPPER(product_name) FROM Products;

OR

(b) **(i)** SELECT LENGTH(full_name) FROM Customers;

(ii) SELECT LOWER(email) FROM Customers;

(iii) SELECT COUNT(DISTINCT city) FROM Customers;

(iv) SELECT TRIM(full_name) FROM Customers;

(v) SELECT * FROM Customers ORDER BY full_name ASC;