SOLUTIONS

Sample Question Paper-10

SECTION-A

1. True

Explanation: True. reset_index() reverts the DataFrame to its default integer index after a custom index has been set.

2. **(a)** 3

Explanation: MOD(3, 7) returns the remainder of 3 divided by 7, which is 3.

3. **(b)** Hacking

Explanation: This is Hacking, where someone gains unauthorized access to a digital account or system.

4. **(b)** `pd.read_csv('file.csv', nrows=100)`

Explanation: `pd.read_csv('file.csv', nrows=100)` reads only the first 100 rows into a DataFrame.

5. **(b)** Switch

Explanation: A Switch connects devices within a network and filters traffic using MAC addresses

6. **(a)** 0 decimal places

Explanation: Defaults to rounding to 0 decimal places.

7. **(c)** Trademark

Explanation: A Trademark protects logos, brand names, and symbols used for commercial identity.

8. **(b)** index

Explanation: index returns the labels (index) of a Pandas Series.

9. **(c)** FullName (stored as FirstName + LastName)

Explanation: FullName (FirstName + LastName) Because a composite key is formed by combining two or more columns together to uniquely identify a record.

10. **(c)** Unified Communications (Voice & Video)

Explanation: Unified Communications integrates voice and video over IP, making it VoIP-based.

11. **(b)** SUM(column_name)

Explanation: SUM(column_name) adds all non-NULL numeric values in the specified column

12. **(c)** Both numbers and labels

Explanation: A Pandas Series can have a numeric index (like 0,1,2…) or custom labels (like names, dates, etc.) as its index.

13. **(b)** Certifying Authorities

Explanation: The Controller of Certifying Authorities regulates digital signature providers.

14. **(a)** WHERE

Explanation: The WHERE clause filters individual rows before any aggregation or grouping.

15. **(a)** `df.loc[1:4, ['A','C']]`

Explanation: `df.loc[1:4, ['A','C']]` selects rows 1 to 4 (inclusive) and only columns A and C.

16. **(a)** Ring

Explanation: Ring topology connects each node to two neighbors, forming a closed loop.

17. **(b)** It returns a specified number of characters from within a string starting at a given position.

Explanation: Returns a specified number of characters from a string starting at a given position.

18. **(b)** pandas.DataFrame.from_records ([{'a':1}, {'a':2}])

Explanation: pandas.DataFrame.from_records([{'a':1},{'a':2}]) builds a DataFrame from a list of dicts.

19. **(a)** ROUND()

Explanation: ROUND() is a scalar function used for formatting numbers, not aggregation.

20. **(c)** A is True, but R is False.

Explanation: A is true, but R is false. df.T transposes the DataFrame,(transpose swaps rows and columns, not rotate visually.)

21. **(a)** Both A and R are True and R correctly explains A.

Explanation: Both A and R are true, and R correctly explains A. SELECT is part of DML operations.

SECTION-B

22. **(a)** An Index is an immutable array that labels the axis of a Series or DataFrame.

One property of an Index is is_unique, a boolean that tells whether all labels in the index are unique.

OR

(b) All elements in a Series share the same data type (dtype).

Different columns in a DataFrame can each have their own dtype.

23. E-waste refers to end-of-life electronic equipment.

Open burning of cables and circuit boards releases dioxins and furans into the air contributing to respiratory problems and air pollution.

24.
```
import pandas as pd
data = {
'id': [1, 2],
'name': ['Alice', 'Bob']
}
df = pd.DataFrame(data)
print(df)
```

25. **(a)** A URL (e.g., https://example.com/page) contains the protocol (HTTP/HTTPS), the domain and the resource path.

The browser first resolves the domain via DNS to an IP then opens a TCP connection to that IP and sends an HTTP request for the specified path. The web server at that IP returns the matching files.

OR

(b) Web hosting is a service that stores websites on servers and makes them accessible on the internet. Example: Bluehost, HostGator, or GoDaddy.

26. **(i)** SELECT DAYNAME('2025-01-05');

(ii) SELECT UPPER('Incredible India');

27. **Ethical hacking** is the practice of legally and responsibly testing computer systems, networks, or applications to identify and fix security vulnerabilities. It is performed with the permission of the system owner to improve cybersecurity.

Difference from non-ethical hacking:

Ethical hacking is done with authorization and aims to protect systems, while **non-ethical hacking** involves unauthorized access with malicious intent, such as stealing data or causing damage.

28. **(a)**

	EmpID	Pay
0	101	50000
1	102	60000
2	103	55000

OR

(b)

	Language	Creator
0	Python	Guido van Rossum
2	C++	Bjarne Stroustrup

SECTION-C

29. **(i)** Intellectual Property (IP) refers to creations of the mind such as literary works, inventions, etc.

Intellectual Property Rights (IPR) are legal rights granted to creators for their original work.

(ii) ahul's invention will be covered under a patent.

(iii) Inventors need strong Intellectual Property Rights (IPR) to protect their creations from being copied, to gain recognition, and to earn financial benefits.

30. **(a)**
```
import numpy as np
import pandas as pd
marks = np.array([92, 85, 89, 90])
subjects = ['English', 'Art', 'Music', 'Drama']
s = pd.Series(marks, index=subjects)
print(s)
```

OR

(b)
```
import pandas as pd
data = {
"Country": ["USA", "Canada", "Mexico"],
"Capital": ["Washington", "Ottawa", "Mexico City"]
}df = pd.DataFrame(data)
print(df)
```

31. **(i)**
```
CREATE TABLE STUDENTS (
StudentID INTEGER PRIMARY KEY,
FullName VARCHAR(40),
DOB DATE,
Grade CHAR(2)
);
```

(ii) INSERT INTO STUDENTS (StudentID, FullName, DOB, Grade)
VALUES (501, 'Anjali Das', '2006-08-21', 'A+');

32. **(a)** **(i)** SELECT Class, COUNT(*) AS TotalStudents FROM STUDENT GROUP BY Class HAVING COUNT(*) > 1;

(ii) SELECT Subject, Score FROM MARKS WHERE Score BETWEEN 75 AND 90;

(iii) SELECT STUDENT.Name, MARKS.Subject, MARKS.Score
FROM STUDENT
JOIN MARKS ON STUDENT.StudentID = MARKS.StudentID
ORDER BY MARKS.Score DESC;

OR

(b) **(i)** EmployeeID can be considered as Primary Key because it uniquely identifies each employee in the table.

(ii) ALTER TABLE Employee ADD Experience INT;

(iii)

Department	COUNT(*)
IT	1
Marketing	2
Finance	2

SECTION-D

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

(ii) plt.bar(Days, Sales, label='Sales')

(iii) plt.title('Daily Sales Report')

(iv) plt.savefig('daily_sales.png')

34. **(a)** **(i)** SELECT SELECT UPPER(Title) AS Title, UPPER(Genre) AS Genre
FROM Book ORDER BY Price DESC;

(ii) SELECT BookID, YEAR(Publish_Date) AS Publish_Year
FROM Book;

(iii) SELECT SUM(Price) AS Total_Education_Price FROM Book
WHERE Genre = 'Education';

(iv) SELECT SELECT Genre, COUNT(*) AS Book_Count
FROM Book GROUP BY Genre;

OR

(b) **(i)**

Name	SUBSTR(Name,5)
Arjun Verma	n Verma
Meera Joshi	a Joshi

(ii)

lower(Name)
meera Joshi

(iii)

Average. Age
37.0

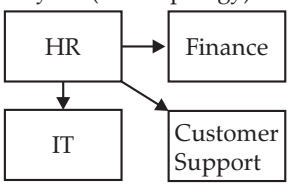
(iv)

Name	Age
Meera Joshi	32
Neha Kapoor	38

SECTION-E

35. **(i)** The server should be installed in the HR department as it has the most number of computers.

(ii) Cable layout (Star Topology)



(iii) Switch/Hub

(iv) WAN (Wide Area Network)

(v) Repeater

36. **(i)** Last three rows

```
df_cars.tail(3)
```

(ii) Add Mileage column

```
df_cars['Mileage'] = [15, 18, 12, 10, 20]
```

(iii) Delete Year column

```
df_cars.drop('Year',        axis=1, inplace=True)
```

(iv) Rename Price to Cost

```
df_cars.rename(columns={'Price': 'Cost'}, inplace=True)
```

(v) Display Make and Cost

```
df_cars[['Make', 'Cost']]
```

37. **(a)** **(i)** SELECT LEFT(user_code, 6) FROM Users;

(ii) SELECT COUNT(Trans_ID) FROM Transactions;

(iii) SELECT YEAR(signup_date) FROM Users;

(iv) SELECT TRIM(Street) FROM Addresses;

(v) SELECT CURRENT_DATE();

OR

(b) **(i)** SELECT LENGTH('Artificial Intelligence');

(ii) SELECT INSTR(Topic, 'i') FROM Seminars;

(iii) SELECT POWER(Duration, 2) FROM Sessions;

(iv) SELECT AVG(Rating) FROM Reviews;

(v) SELECT SUM(Rating) FROM Reviews;

❐❐❐