

# Sample Question Paper-7

## SECTION-A

1. False

**Why?** shape returns a **tuple** (rows, columns), not the total number of elements. Total elements would be rows \* columns or use .size.

**Why not?** If you think it's True, you're confusing .shape with .size.

2. (c) CONCAT()

**Why?** CONCAT() joins two or more strings in SQL. Example: SELECT CONCAT('Hello', ' World'); → "Hello World".

**Why not?**

(a) JOIN() is for combining rows from multiple tables.

(b) MERGE() is not a standard SQL string function (used in some DB operations, not for strings).

(d) APPEND() is not a standard SQL string function.

3. (c) Keeping software and antivirus up to date

**Why?** Regular updates patch security vulnerabilities and help prevent hacking/malware.

**Why not?**

(a) Same password for all sites = high risk.

(b) Clicking random links = phishing risk.

(d) Ignoring updates leaves you exposed.

4. (c) df.sort\_values()

**Why?** sort\_values() sorts DataFrame rows by a column's values.

**Why not?**

(a) order\_by() is SQL syntax, not Pandas.

(b) sort() in Pandas is deprecated for DataFrames.

(d) sort\_column() doesn't exist.

5. (c) Switch

**Why?** Switches use MAC addresses to forward frames at the data link layer (Layer 2).

**Why not?**

(a) RJ45 is a connector.

(b) Modem connects to ISP.

(d) Repeater only regenerates signal.

6. (c) ROUND(num, 0)

**Why?** ROUND() with precision 0 rounds to the nearest whole number.

**Why not?**

(a) FLOOR() always rounds down.

(b) TRUNCATE() cuts off decimals without rounding.

(d) MOD() gives remainder.

7. (b) To receive legal ownership and protection

**Why?** Copyright registration grants legal proof and rights over your work.

**Why not?**

(a) Hackers = unrelated.

(b) Marketing = not main reason.

(d) Patents = different law.

8. (a) pd.Series([1, 2, 3], index=['x', 'y', 'z'])

**Why?** index parameter sets custom labels.

**Why not?** olabels/name/id aren't valid parameters for index labels.

9. (b) set of attributes that can uniquely identify each record in a table.

**Why?** A candidate key is any set of attributes that uniquely identifies a row.

**Why not?**

(a) incomplete definition

(c) foreign key definition

(d) invalid — keys can't be NULL.

10. (c) Voice communication over data networks

**Why?** VoIP uses the internet/data network to make calls, often cheaper and flexible.

**Why not?**

(a) Requires landline = false

(b) High charges = opposite of VoIP

(d) 4G SIM only = false (works on broadband too)

11. (c) AVG(age)

**Why?** AVG() calculates average in SQL.

**Why not?**

(a) MEAN/AVERAGE not standard SQL

(b) TOTAL() = sum, not average.

12. (c) 'a', 'b', 'c', 'd'

**Why?** Pandas aligns indexes before addition; result index is the union of both.

**Why not?**

(a) ['a', 'b', 'c'] = A only

(b) ['b', 'c'] = intersection

(d) ['b'] = single index only.

- 13. (c)** Information Technology Act, 2000  
**Why?** IT Act covers e-commerce legality and cybercrime prevention.  
**Why not?** Other options are unrelated.
- 14. (a)** COUNT()  
**Why?** COUNT() counts rows (or non-NULL values if column specified).  
**Why not?**  
 (b) SUM = totals numeric values.  
 (c) TOTAL()/NUMBER(\*) not standard SQL.
- 15. (b)** It returns the last two rows  
**Why?** tail(n) shows last n rows.  
**Why not?**  
 (a) Drop = drop() method.  
 (c) Random = sample() method.  
 (d) Rename = rename() method.
- 16. (d)** Star  
**Why?** Star is simple to expand and cost-effective compared to mesh.  
**Why not?**  
 (a) Ring/Tree more complex and costly.  
 (b) Mesh very costly.
- 17. (b)** SUBSTR()  
**Why?** SUBSTR() extracts part of a string from a given position.  
**Why not?**  
 (a) LENGTH() = string length  
 (c) TRIM() = remove spaces  
 (d) INSTR() = position of substring.
- 18. (c)** From scalar value  
**Why?** Pandas DataFrame requires an array-like or dict-like structure; scalar only works for Series, not DataFrame.  
**Why not?** Dictionary, list of lists and tuple of integers are valid.
- 19. (a)** AVG()  
**Why?** AVG() returns mean in SQL.  
**Why not?**  
 (b) MEAN() not SQL standard  
 (c) TOTAL() = sum  
 (d) SUM()/COUNT() works but not the direct SQL function.
- 20. (c)** Assertion True, Reason False  
**Why?** read\_csv() indeed reads data from a CSV into a DataFrame.  
 But it can handle headers, missing values, and different delimiters (using the delimiter or sep parameter).
- 21. (a)** Both True, Reason is the correct explanation.

**Why?** INSERT INTO is DML.

DML includes insert, update, delete.

## SECTION-B

- 22. (a)** Role of dtype in Pandas Series  
 The dtype attribute in a Pandas Series tells you the data type of the values stored in the Series (e.g., int64, float64, object).  
 It helps in:
- Understanding how data is stored in memory.
  - Choosing the right operations (e.g., math ops work for numeric dtypes).
  - Optimising performance by using efficient data types.
- Example:**
- ```
import pandas as pd
s = pd.Series([10, 20, 30])
print(s.dtype) # int64
```
- OR**
- (b)** NumPy in Python  
 NumPy (Numerical Python) is a Python library for fast numerical computation using multi-dimensional arrays.  
**Why use NumPy arrays over Python lists:**
- NumPy arrays are faster because they store elements of the same type in contiguous memory.
  - They support vectorized operations, meaning operations are applied to the whole array without loops.
- Example use case:**
- ```
import numpy as np
arr = np.array([1, 2, 3, 4])
print(arr * 2) # [2 4 6 8] - vectorized operation
```
- 23. Avoiding plagiarism**  
 Plagiarism can be avoided by:
- Rephrasing content in your own words.
  - Giving credit to the original source.
- Two tools/practices:
1. Turnitin / Grammarly — check for plagiarism.
  2. Citation Managers (e.g., Zotero, Mendeley) — help reference sources properly.
- 24. (i)** SELECT SUBSTR('Learning Structured Query Language', 20, 5);  
**(ii)** SELECT LENGTH('Learning Structured Query Language');
- 25. (a)** Website vs Web Page
- Website: A collection of related web pages under a domain.  
**Example:** www.wikipedia.org (whole site).
  - Web Page: A single document on a website.

Example: [https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)) (specific page).

OR

(b) Session Cookies vs Persistent Cookies

- Session Cookies: Stored temporarily, deleted when browser closes.

Example: Shopping cart session on an e-commerce site.

- Persistent Cookies: Stored on disk for a set duration.

Example: "Remember Me" login on Gmail.

26. Choosing Primary Key

If multiple columns qualify:

- Designer chooses the one that is unique, stable and minimal in size.
- Remaining unique keys can be set as candidate keys.
- Only one primary key is chosen; others remain as alternate keys.

27. Ergonomic Safety Tips

1. Adjust chair and monitor height to maintain neutral posture.
2. Take regular breaks to avoid eye strain and muscle fatigue.

25. (a) import pandas as pd

```
data = {'Name': ['Anita', 'Sunil'],
        'Grade': ['A', 'B']} # Added quotes on Name
df = pd.DataFrame(data) # corrected case sensitivity of data
print(df['Grade']) # Changed 'Marks' to 'Grade'
```

OR

(b) import pandas as pd

```
data = [85, 92, 88]
students = ['Ravi', 'Neha', 'Tina']
df = pd.DataFrame(data, index=students, columns=['Marks'])
print(df)
```

## SECTION-C

29. E-Waste & Water Pollution

- (i) Link between e-waste and water pollution:** E-waste contains toxic substances like lead, mercury and cadmium. When dumped in landfills near water bodies, these chemicals leach into the soil and contaminate groundwater and river water, harming plants, animals and humans.
- (ii) Eco-friendly way to manage outdated devices:** The company can use E-waste recycling centres or donate functional laptops to schools/NGOs instead of dumping them.
- (iii) Why follow e-waste laws strictly:**

- Prevents environmental damage.
- Avoids legal penalties and fines.
- Protects public health and company reputation.

30. (a) 

```
import pandas as pd
data = {"City": ["Mumbai", "Kolkata", "Bengaluru", "Jaipur"],
        "Population": [20400000, 14800000, 12000000, 4300000]}
df = pd.DataFrame(data)
print(df)
```

OR

(b) 

```
import pandas as pd
data = {
    "Japan": "Yen",
    "USA": "Dollar",
    "India": "Rupee"
}
s = pd.Series(data)
print(s)
```

- 31. (i)** CREATE TABLE COURSES (  
CourseID VARCHAR(10) PRIMARY KEY,  
Name VARCHAR(40),  
Duration INTEGER,  
Fee DECIMAL(7,2)  
);
- (ii)** INSERT INTO COURSES (CourseID, Name, Duration, Fee)  
VALUES ('C101', 'Data Analysis with Python', 6, 8500.50);

- 32. (a) (i)** SELECT Name FROM EMPLOYEE  
WHERE Department = 'IT'  
ORDER BY Name ASC;
- (ii)** SELECT UPPER(Month) FROM SALARY  
WHERE Amount > 60000;
- (iii)** SELECT EMPLOYEE.Name, SALARY.  
Month, SALARY.Amount  
FROM EMPLOYEE  
JOIN SALARY  
ON EMPLOYEE.EmployeeID = SALARY.  
EmployeeID;

OR

- (b) (i)** Primary Key:  
ProductID — because it uniquely identifies each product in the table, no two rows will have the same ProductID.
- (ii)** ALTER TABLE PRODUCT ADD  
StockQuantity NUMERIC;
- (iii)** Output:  

Category	TotalProducts
Electronics	3
Furniture	2

## SECTION-D

**34. Statement-1:** import matplotlib.pyplot as plt → needed for plotting.

**Statement-2:** plt.bar(students, marks, label='Math Marks') → creates bar chart.

**Statement-3:** plt.title('Math Marks of Students') → sets chart title.

**Statement-4:** plt.savefig('math\_marks.png') → saves figure.

- 34. (a)**
- (i) 

```
SELECT LOWER(Name) AS Name,
      LOWER(Department) AS Department
FROM Employee ORDER BY Salary DESC;
```
  - (ii) 

```
SELECT EmpID, YEAR(Joining_Date) AS
      Joining_Year
FROM Employee;
```
  - (iii) 

```
SELECT MAX(Salary) AS Highest_Salary
FROM Employee;
```
  - (iv) 

```
SELECT Department, COUNT(*) AS
      Employee_Count
FROM Employee
GROUP BY Department;
```
- OR**
- (b)
- (i) 

```
SELECT UPPER(Title) AS Title,
      UPPER(Author) AS Author
FROM Books
ORDER BY Author ASC;
```
  - (ii) 

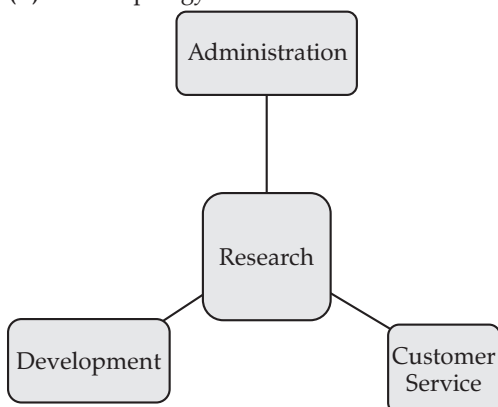
```
SELECT BookID, MONTHNAME(Purchase_
      Date) AS Purchase_Month
FROM Books;
```
  - (iii) 

```
SELECT AVG(Price) AS Average_Price
FROM Books;
```
  - (iv) 

```
SELECT Author, COUNT(*) AS Total_Books
FROM Books
GROUP BY Author;
```

## SECTION-E

- 35. (i)** Research is the most suitable place to place server as it has maximum number of systems.
- (ii)** Star Topology



(iii) Switch

(iv) WAN

(v) Repeater as it is used to restrengthen the weak signals over long distances.

**36. (i)** Display the first two rows

```
print(df.head(2))
```

**(ii)** Add new column "Salary"

```
df["Salary"] = [75000, 68000,
                82000, 70000, 65000]
```

**(iii)** Delete the column "Age"

```
df.drop(columns=["Age"],
         inplace=True)
```

**(iv)** Rename "Department" to "Dept"

```
df.rename(columns={"Department":
                  "Dept"}, inplace=True)
```

**(v)** Display only "Name" and "Dept" columns

```
print(df[["Name", "Dept"]])
```

```
print(students_dropped)
```

**37. (a) (i)**

```
SELECT RIGHT(emp_code, 3) AS last_
      three_chars
FROM Employees;
```

**(ii)**

```
SELECT SUM(salary) AS total_salary
FROM Payroll;
```

**(iii)**

```
SELECT MONTHNAME(join_date) AS
      month_name
FROM Employees;
```

**(iv)**

```
SELECT LCASE(email) AS lower_email
FROM Clients;
```

**(v)**

```
SELECT SYSDATE();
```

**OR**

**(b) (i)**

```
SELECT LENGTH('InformaticsPractice') AS
      total_chars;
```

**(ii)**

```
SELECT INSTR(Emp_Name, 'e')
FROM Employees;
```

**(iii)**

```
SELECT POWER(Product_Price, 3) AS
      price_cube
FROM Products;
```

**(iv)**

```
SELECT MAX(Marks) AS max_marks
FROM Students;
```

**(v)**

```
SELECT SUM(Quantity) AS total_quantity
FROM Orders;
```