

# Sample Question Paper-4

## SECTION-A

1. True

**Why?** `iloc[]` stands for **integer-location based indexing**—it selects rows/columns by position (0-based index).

**Why not?** If it was false, that would mean `iloc[]` does not use integer indexing, which is not true.

2. (d) All of the above

**Why?** • `SYSDATE` (Oracle/MySQL), `CURDATE()` (MySQL), and `NOW()` (MySQL/PostgreSQL) can all return the current date.

3. (c) Cyber Bullying

**Why?** Cyberbullying = repeated online harassment, threats or insults.

**Why not?**

(a) Hacking = unauthorised access to systems.

(b) Phishing = tricking to steal credentials.

(d) Identity theft = stealing personal info to impersonate.

4. (b) `df.describe()`

**Why?** `describe()` gives count, mean, std, min, max and quartiles.

**Why not?**

(a) `stats()` and `summary()` don't exist in Pandas.

(c) `info()` gives column types and non-null counts, not statistics.

5. (b) Hub

**Why?** Hubs send incoming signals to all ports.

**Why not?**

(a) Switch sends only to the intended device.

(c) The router directs between networks.

(d) The modem connects to ISP.

6. (b) 2

**Why?** The remainder isn't 3, 1 or 4.

**Why not?** The remainder isn't 3, 1 or 4.

7. (b) A mathematical formula

**Why?** Abstract ideas, formulas and natural laws are not patentable.

**Why not?** Machines, apps with novel tech and drug formulas can be patented.

8. (c) Index can contain duplicate values

**Why?** Pandas allows duplicate index labels.

**Why not?**

(a) Labels don't have to be strings.

(b) Index can be changed with `.set_index()` or `.index =`

(d) Index doesn't always start from 1.

9. (c) Candidate keys

**Why?** Candidate keys are attributes that can uniquely identify a row.

**Why not?**

(a) Composite key = combination of attributes (not needed here individually).

(b) Foreign key = links to another table.

(d) Super key = candidate key + extra attributes.

10. (d) VLC Media Player

**Why?** VLC is a media player, not a cloud storage platform.

**Why not?** Google Drive, Dropbox, iCloud store files online.

11. (b) `MAX()`

**Why?** `MAX()` returns the largest value in a column.

**Why not?**

(a) `HIGH()` doesn't exist.

(c) `TOP()` limits rows in `SELECT`.

(d) `CEIL()` rounds numbers up.

12. (c) They are included with NaN values

**Why?** Pandas aligns by index; unmatched indices result in NaN.

**Why not?** They aren't discarded, added as 0 or cause errors.

13. (a) Phishing

**Why?** Phishing = fraudulent email to steal info.

**Why not?** Hacking = system breach; cyberbullying = harassment; spamming = mass unsolicited messages.

14. (b) `DISTINCT`

**Why?** Removes duplicate rows from query output.

**Why not?**

(a) `UNIQUE` is a constraint, not for query results.

(c) `ONLY` and `PRIMARY` don't do this.

15. (b) `df.loc[:, 'Marks']`

**Why?** `.loc[:, 'Marks']` selects all rows and only the "Marks" column.

**Why not?**

(a) `select()` not in Pandas.

(c) `.row()` doesn't exist.

(d) `.get()` is not used for columns like that.

16. (b) Ring  
**Why?** Ring topology connects devices in a loop, and data flows one way.  
**Why not?** Tree, Bus and Star are not loops.
17. (a) CONCAT()  
**Why?** CONCAT() joins strings.  
**Why not?** ADD() is for numbers, MERGE() is not SQL standard, UNION() merges result sets.
18. (c) `pandas.read_csv()`  
**Why?** Reads CSV directly into DataFrame.  
**Why not?** `read_table()` is for tab-delimited, `read_txt()` doesn't exist, `read()` isn't a Pandas function.
19. (b) NULL  
**Why?** Aggregate functions ignore NULLs; if all are NULL, the result is NULL.  
**Why not?** It doesn't default to 0, error or infinity.
20. (d) A is False, but R is True.  
**Why?** `df.iloc[1:3]` includes rows at positions 1 and 2 (upper bound excluded), so A is false.  
`iloc[]` does not include upper bound, so R is true.
21. (a) Both A and R are True, and R correctly explains A.  
**Why?** ALTER can add/remove columns.  
 ALTER is a DDL command that modifies the schema — this explains A.

## SECTION-B

22. (a) In Pandas, the `head()` and `tail()` functions are used to view a small portion of a DataFrame or series:
- `head(n)` returns the first `n` rows. By default, it shows the top 5 rows.
  - `tail(n)` returns the last `n` rows. Default is also 5 rows.
- These functions help in quick inspection of large datasets for checking structure, column names and data format.
- Prep Tool:**
- **Concept Applied:** Data preview in Pandas
  - **Common Mistake:** Forgetting parentheses () or using `head[5]` instead of `head(5)`
  - **Answering Tip:** Always mention default value is 5 rows
  - **High-Value Point:** Helpful for data analysis and debugging
22. (b) A Python library is a collection of pre-written functions and modules that help in solving specific problems without writing code from scratch.
- **Pandas:** It supports data handling tasks like reading/writing data, cleaning and analysing datasets using DataFrame and Series structures.
  - **Matplotlib:** It is used for data visualisation tasks such as drawing bar charts, line graphs, pie charts, etc.

### Prep Tool:

- **Concept Applied:** Python libraries in Data Analysis
- **Common Mistake:** Confusing Pandas with data visualisation
- **Answering Tip:** Use one-liners with keyword functions like DataFrame and plot()
- **High-Value Point:** Libraries save time and improve efficiency

23. • **Open-source software** is freely available, and its source code can be modified and distributed.

**Example:** LibreOffice, Python

- Proprietary software is owned by companies; the source code is not shared.

**Example:** MS Office, Windows OS

**Difference:** Open-source allows modification, proprietary software does not.

### Prep Tool:

- **Concept Applied:** Types of software licensing
- **Common Mistake:** Using free = open source (not always true)
- **Answering Tip:** Always include an example for each
- **High-Value Point:** Licensing control is the key differentiator

24. (i) `SELECT SUBSTRING('Information Technology Department', 13, 10);`  
 (ii) `SELECT INSTR('Information Technology Department', 'Depart');`

### Prep Tool:

- **Concept Applied:** String functions in SQL – SUBSTRING() and INSTR()
- **Common Mistake:** Wrong position (index starts at 1 in SQL), incorrect length
- **Answering Tip:** Count character positions carefully
- **High-Value Point:** INSTR returns the first match position

25. (a) **Incognito mode** is a private browsing feature in web browsers that does not save your browsing history, cookies or form inputs.

**Benefit:** It helps maintain privacy on shared or public devices.

- (b) **Pop-up blockers** are browser tools that prevent unwanted pop-up windows from appearing.

They **enhance safety** by stopping malicious ads or phishing attempts from being triggered automatically.

### Prep Tool:

- **Concept Applied:** Privacy & safety features in browsers
- **Common Mistake:** Confusing incognito with full anonymity (it doesn't hide from ISP)

- **Answering Tip:** Give practical benefit (e.g., avoid tracking)
- **High-Value Point:** Both features are used for privacy protection

**26.** A **Foreign Key** is a field in one table that refers to the *Primary Key* in another table.

It ensures **referential integrity** by maintaining valid relationships between tables. If a record is inserted with a non-existing reference, it will be rejected.

**Prep Tool:**

- **Concept Applied:** Database relationships
- **Common Mistake:** Confusing with Primary Key
- **Answering Tip:** Use terms like *reference*, *related table*
- **High-Value Point:** Maintains data accuracy and consistency

**27.** Two **netiquettes** while using social media are:

- (1) **Be respectful:** Do not post offensive, abusive, or hurtful comments.
- (2) **Protect privacy:** Do not share personal or confidential information of yourself or others publicly.

**28. (a)** import pandas as pd

```
marks = [90, 85, 88, 76] # Changed from tuple to list for clarity
```

```
s = pd.Series(marks) # 'Series' should be capitalized
```

```
print(s) # print() uses parentheses, not square brackets
```

**(b)** import pandas as pd

```
data = ['Chennai', 'Lucknow', 'Imphal']
```

```
indx = ['Tamil Nadu', 'Uttar Pradesh', 'Manipur']
```

```
s = pd.Series(data, indx)
```

```
print(s)
```

**Prep Tool:**

- **Concept Applied:** Creating Series with custom index
- **Common Mistake:** Lowercase 'series', square brackets in print
- **Answering Tip:** Use correct data structures (list or tuple)
- **High-Value Point:** Capitalisation and syntax matter in Pandas

## SECTION-C

**29. (i) Health Hazard:**

Burning electronic waste releases toxic chemicals like lead, mercury and cadmium. These can cause respiratory issues, skin diseases and even cancer in humans and animals.

**(ii) Safer Handling Suggestion:**

E-waste should be handed over to authorised e-waste recyclers or collected at designated e-waste collection centres for proper recycling.

**(iii) Sustainable Future:**

Proper disposal of e-waste helps recover valuable materials (like copper and gold), **reduces environmental** pollution and promotes reuse and recycling, thus supporting a greener planet.

**Prep Tool:**

- **Concept Applied:** Cyber safety and environmental awareness

- **Common Mistake:** Confusing e-waste with general waste

- **Answering Tip:** Give health + environment + reuse angle

- **High-Value Point:** Mention "authorized recyclers" and "resource recovery"

**30. (a)** import pandas as pd

```
students = [
    {"Name": "Riya", "Grade": "A"},
    {"Name": "Mohan", "Grade": "B"},
    {"Name": "Sneha", "Grade": "A+"},
    {"Name": "Rahul", "Grade": "C"}
]
```

```
df = pd.DataFrame(students)
print(df)
```

**30. (b)** import pandas as pd

```
data = {
    "Gujarat": "Gandhinagar",
    "Rajasthan": "Jaipur",
    "Punjab": "Chandigarh"
}
```

```
s = pd.Series(data)
print(s)
```

**Prep Tool:**

- **Concept Applied:** Pandas DataFrame and Series creation

- **Common Mistake:** Using a tuple instead of a dictionary; wrong case in Series

- **Answering Tip:** Emphasise key-value logic for Series

- **High-Value Point:** Clean structure and capitalized method names

**31. (a)** CREATE TABLE EMPLOYEES (  
EmpID INTEGER PRIMARY KEY,  
EmpName VARCHAR(30),  
Department VARCHAR(20),  
Salary FLOAT(8,2),  
DOJ DATE  
);

**(b)** INSERT INTO EMPLOYEES

```
VALUES (101, 'Ramesh Sharma',  
'Finance', 55000.75, '2019-06-01');
```

### Prep Tool:

- **Concept Applied:** Table creation, constraints, data types
- **Common Mistake:** Using DATE without proper format or missing constraints
- **Answering Tip:** Always list data types + constraints (e.g., PRIMARY KEY)
- **High-Value Point:** Include precision like FLOAT(8,2) and formatted date

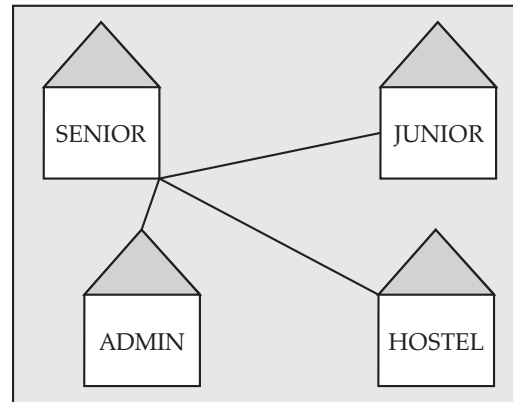
32. (a) (i) `SELECT SUBJECT, AVG(SCORE) AS Avg_Score FROM GRADES GROUP BY SUBJECT;`  
(ii) `SELECT DISTINCT GRADE, SCORE FROM GRADES ORDER BY SCORE DESC;`  
(iii) `SELECT STUDENTS.STU_NAME, GRADES.SUBJECT FROM STUDENTS JOIN GRADES ON STUDENTS.STU_ID = GRADES.STU_ID;`
- (b) (i) The attribute ProductID can be considered as the Primary Key because it uniquely identifies each product in the table.  
(ii) SQL query to increase the price of all "Electronics" items by 10%:  
`UPDATE PRODUCT SET Price = Price * 1.10 WHERE Category = 'Electronics';`  
(iii) Output of the query `SELECT Category, AVG(Price) FROM PRODUCT GROUP BY Category;` will be:  
Stationery 30  
Electronics 600  
Accessories 800

## SECTION-D

33. (i) `import matplotlib.pyplot as plt`  
(ii) `plt.bar(quantities, labels=fruits)`  
(iii) `plt.title('FruitSaleDistribution')`  
(iv) `plt.show()`
34. (a) (i) `SELECT DISTINCT Transporter FROM Bus;`  
(ii) `SELECT Transporter, SUM(Charges) FROM Bus GROUP BY Transporter;`  
(iii) `SELECT Transporter, AVG(Distance) FROM Bus GROUP BY Transporter;`  
(iv) `SELECT Transporter, MIN(Nooftud) FROM Bus GROUP BY Transporter;`
34. (b) (i) 193  
(ii) 194  
(iii) 93.75  
(iv) 6

## SECTION-E

35. (i) Star Topology is suggested.



- (ii) Recommended Wing to install server: Wing S as it has maximum number of computers which will keep maximum traffic local to server.  
(iii) Install the firewall between the School Network and the Internet (main router/modem)  
(iv) Install Wi-Fi routers in central locations of each wing (especially Wing S and J) for even signal distribution  
(v) Switch should be placed in each wing.
36. (i) `print(students_df.head(3))`  
(ii) `print(students_df['Name'])`  
(iii) `students_df.drop('Marks', axis=1, inplace=True)`  
(iv) `print(students_df.loc[1:3, 'Name'])`  
(v) `students_df.rename(columns={'Class': 'Standard'}, inplace=True)`
37. (a) (i) `SELECT SUM(pages) FROM Books;`  
(ii) `SELECT LEFT(isbn_code, 5) FROM Books;`  
(iii) `SELECT TRIM(author_name) FROM Authors;`  
(iv) `SELECT MAX(price) FROM Books;`  
(v) `SELECT COUNT(*) FROM Authors;`
- (b) (i) `SELECT AVG(marks_obtained) FROM Results;`  
(ii) `SELECT RIGHT(roll_no, 2) FROM Students;`  
(iii) `SELECT TRIM(student_name) FROM Students;`  
(iv) `SELECT MAX(attendance) FROM Attendance;`  
(v) `SELECT COUNT(*) FROM Students;`