B5
Practical Viva Questions and Answers

---

a) 1D (Linear) Data Visualization

Q1. What is 1D (Linear) data visualization?
Ans:
1D data visualization represents a single variable across its different values. It helps in understanding the distribution or frequency of one attribute.

Q2. Which charts are used for 1D data visualization?
Ans:
Common charts are Bar charts, Line charts (single line), Histograms, and Pie charts.

---

b) 2D (Planar) Data Visualization

Q1. What is 2D (Planar) data visualization?
Ans:
2D data visualization represents two variables together on a plane, typically using X and Y axes, to observe relationships between them.

Q2. Give an example using the Adult dataset.
Ans:
Example: Plotting "Age" on the X-axis and "Hours-per-week" on the Y-axis using a Scatter plot.

---

c) 3D (Volumetric) Data Visualization

Q1. How is 3D simulated in Tableau?
Ans:
Since Tableau is mainly 2D, 3D is simulated by adding a third variable using **Color**, **Size**, or **Shape** options in the Marks pane.

Q2. Which features are used to create the 3D feeling?
Ans:
Color encoding, Size scaling, and different Shapes are used to represent the third dimension.

---

d) Temporal Data Visualization

Q1. What is temporal data visualization?
Ans:
Temporal data visualization shows how data changes over time, typically using a Date or Time dimension on the X-axis.

Q2. Which dataset field did you use for time?
Ans:
The Adult dataset does not have a direct Date field, so we can either create a Year field based on "Workclass" experience or use any available time-based attribute if present.

Q3. Why is Line Chart preferred for temporal data visualization?
Ans:
Line charts clearly show trends, patterns, and changes over time, making them ideal for temporal analysis.

---

e) Multidimensional Data Visualization

Q1. What is multidimensional data visualization?
Ans:
Multidimensional visualization displays more than three variables at once using multiple visual properties like position (X, Y), color, size, and shape.

Q2. How many variables did you plot together?
Ans:
I plotted four variables together:

- Sepal Length (X-axis)
- Petal Width (Y-axis)
- Species (Color)
- Petal Length (Size)

Q3. What is the use of color and size?
Ans:
Color is used to differentiate categories, while Size represents the magnitude or value of another variable.

---

f) Tree/ Hierarchical Data Visualization

Q1. What is hierarchy in data?
Ans:
Hierarchy in data means a parent-child relationship where one category can be broken into multiple sub-categories.

Q2. What is a Treemap?
Ans:
A Treemap is a chart type that displays hierarchical data as nested rectangles, where size and color represent different variables.

Q3. Which fields did you use for hierarchy?
Ans:
In the Adult dataset, I used "Education" as the parent and "Occupation" as the child to create a Treemap.

---

g) Network Data Visualization

Q1. What is network data visualization?
Ans:
Network data visualization shows the relationship between entities, often represented as nodes (points) and edges (connections).

Q2. Is Tableau good for network visualization?
Ans:
Tableau is not primarily designed for network visualizations. However, basic networks can be simulated using Path and Line charts, and complex networks can be created using extensions like the Sankey Diagram.

Q3. How can you create network diagrams in Tableau?
Ans:
We can create network diagrams by:

- Preparing data with "Source" and "Target" columns.
- Using the Path shelf in Tableau to connect them visually.
- Alternatively, using external extensions like Sankey charts.

Summary Table for Revision

Sr	Type	Example Chart	Dataset Example
1	1D	Bar Chart	Count of Education levels
2	2D	Scatter Plot	Age vs Hours-per-week
3	3D	Color/Size scatter	Age vs Hours-per-week + Education
4	Temporal	Line Chart	Trend over Year (if available)
5	Multidimensional	Bubble Chart	Sepal Length vs Petal Width + Species (Color) + Petal Length (Size)
6	Tree/Hierarchy	Treemap	Education → Occupation
7	Network	Path Diagram / Sankey	Workclass to Occupation connections

---

Final Tip

- Always mention which dataset (Adult or Iris) you used.
- Be ready to show a sample chart on Tableau during the exam if asked.
- Keep answers **short, structured, and to the point** — clarity is important in viva.

A2

Design a distributed application using MapReduce (Java) for Log File Processing

---

Project Description

Design a distributed application using MapReduce (Java) that processes a log file of a system to list out users who have logged for the maximum period. The application runs on Hadoop in pseudo-distributed mode.

All Possible Questions and Answers

1. What is MapReduce?

MapReduce is a programming model for processing large datasets in a distributed manner across a Hadoop cluster. It has two functions: **Map** (process/filter) and **Reduce** (aggregate results).

2. What are Mapper and Reducer in MapReduce?

- **Mapper**: Processes input data and produces intermediate key-value pairs.
- **Reducer**: Combines intermediate results to generate final output.

3. What is Hadoop pseudo-distributed mode?

In pseudo-distributed mode, all Hadoop daemons run on a single machine but behave as if distributed across nodes.

4. How is input split and assigned to Mappers?

Input is split into **InputSplits** (default 128MB). Each split is processed by one Mapper.

5. How does your program handle multiple login and logout for the same user?

The Reducer collects, sorts login/logout records, pairs them, and calculates total session time.

6. What if the log file has missing login or logout entries?

Currently, the program assumes correct pairs. If missing entries exist, results may be incorrect.

7. What is HDFS and why do we use it?

HDFS (Hadoop Distributed File System) reliably stores large files across machines with high throughput.

8. What is the purpose of the Driver class?

The Driver class configures the job (Mapper, Reducer, input, output) and submits it to Hadoop.

9. What is the input and output of your Mapper?

- **Input**: A line from the log file.
- **Output**: (User, login/logout with time) pair.

Example: Input: user1 login 08:00 Output: (user1, login 08:00)

10. What is the input and output of your Reducer?

- **Input**: (User, list of login/logout times)
- **Output**: (User, total minutes logged in)

11. Why are you sorting the login/logout times in the reducer?

Sorting ensures login and logout events are correctly ordered before calculating session durations.

12. Why are you using SimpleDateFormat in the reducer?

SimpleDateFormat parses time strings into Date objects for easy time difference calculation.

13. What does the difference (logout - login) represent?

The duration of a single login session, in minutes.

14. How is data passed between Mapper and Reducer?

Mapper emits key-value pairs, Hadoop shuffles and sorts them before passing to the Reducer.

15. How do you compile and run a MapReduce program?

Steps:

javac -classpath $(hadoop classpath) -d . *.java

jar -cvf loganalysis.jar *.class

hdfs dfs -put logfile.txt /input

hadoop jar loganalysis.jar LogDriver /input /output

16. What happens if login and logout are mismatched?

The current code may fail or produce incorrect results because it expects correct pairs.

17. How can you check if Hadoop services are running?

Use:

jps

Check for NameNode, DataNode, ResourceManager, NodeManager, etc.

18. How do you view the output after job finishes?

Use:

hdfs dfs -cat /output/part-r-00000

19. What if the output directory already exists?

Delete it first:

hdfs dfs -rm -r /output

20. What improvements can you suggest for this project?

- Add validation for login-logout pairing.

- Handle multiple dates.
- Handle missing or corrupted entries.

Short Code Explanation

File	Explanation
Mapper.java	Splits each line to extract user and login/logout info and emits it.
Reducer.java	Gathers all login/logout times for each user, sorts them, calculates total logged time, and emits result.
Driver.java	Configures job setup (Mapper, Reducer, input, output paths) and runs the MapReduce job.

Execution Steps Summary

start-dfs.sh

start-yarn.sh

javac -classpath $(hadoop classpath) -d . *.java

jar -cvf loganalysis.jar *.class

hdfs dfs -mkdir /input

hdfs dfs -put logfile.txt /input/

hadoop jar loganalysis.jar LogDriver /input /output

hdfs dfs -cat /output/part-r-00000

Document Created: For Viva and PR Submission

C1

Q1. What libraries are used in the script?

- BeautifulSoup, requests, and pandas.

Q2. What does the requests.get() function do?

- It sends an HTTP GET request to a webpage and returns the server's response.

Q3. What is BeautifulSoup used for?

- It is used to parse HTML or XML documents and extract data from them easily.

Q4. How do you extract a book's title?

- By selecting the <a> tag inside <h3> using soup.select('article h3 a') and accessing the title attribute.

Q5. How is the rating extracted from the page?

- From the class attribute of the first <p> tag in each article.

Q6. How is the price cleaned before storing?

- By removing the unwanted character (Â£) using .replace('Â£', '').

Q7. What does zip() function do in this code?

- Combines title, rating, and price together to create a dictionary for the DataFrame.

Q8. Why do we use loops over pages?

- Because the website contains multiple pages (1 to 50), and we need to scrape data from all of them.

Q9. What happens inside the second loop (for individual books)?

- For each book, a new request fetches its detailed information like Tax, Availability, etc.

Q10. Why is pd.concat() used instead of append()?

- append() is deprecated in new pandas versions. concat() is the recommended method.