

## **BDA Suggested Practical List:**

### **1. To create Employee data with salary in a single table using R**

```
→ employee_names <- c("John", "Mary", "David", "Anna")
employee_ids <- c(101, 102, 103, 104)
salaries <- c(50000, 60000, 55000, 62000)
employee_data <- data.frame(EmployeeName = employee_names, EmployeeID =
employee_ids, Salary = salaries)
print(employee_data)
```

### **2. Define a function in a script, source it to the RStudio and print the multiples of a number.**

```
→ file1.R``
print_multiples <- function(n, count) {
  for(i in 1:count) {
    print(n * i)
  }
}
``
RStudio``
source("File1_path.R")
print_multiples(5, 10)
``
```

### **3. Create two data frames roll no, name, class and other data frame as roll no, subject, marks. Merge this two data frame and print the o/p.**

```
→
# Create the first data frame
df1 <- data.frame(
  roll_no = c(1, 2, 3, 4),
  name = c("Alice", "Bob", "Charlie", "David"),
  class = c("A", "B", "A", "C")
)
# Create the second data frame
df2 <- data.frame(
  roll_no = c(1, 2, 3, 4),
  subject = c("Math", "English", "Science", "History"),
  marks = c(90, 85, 92, 88)
)
# Merge the two data frames based on the 'roll_no' column
merged_df <- merge(df1, df2, by = "roll_no")

# Print the merged data frame
print(merged_df)
```

### **4. Create a table in Hive, display the employees record after loading the data into the table.**

### **5. Create a table in HIVE and add a partition as country='AUS'.**

## 6. Hadoop Basic commands

→

1. Hadoop fs -help
2. Hadoop fs -help ls
3. Hadoop fs -mkdir /user/cloudera/myfolder
4. Hadoop fs -ls /user/cloudera:
5. Hadoop fs -copyFromLocal /home/cloudera/Atharva.txt /user/cloudera/myfolder
6. hadoop fs -moveFromLocal /home/cloudera/Atharva.txt /user/cloudera
7. hadoop fs -put sample.txt /user/cloudera/myfolder
8. hadoop fs -cat /user/cloudera/myfolder/Atharva.txt
9. hadoop fs -cp /user/cloudera/myfolder/sample.txt /user/cloudera
10. hadoop fs -mv /user/cloudera/myfolder/myfile.txt /user/cloudera
11. hadoop fs -tail /user/cloudera/myfolder/Atharva.txt
12. hadoop fs -rm /user/cloudera/myfolder/sample.txt
13. hadoop fs -du /user/cloudera/myfolder/Atharva.txt
14. hadoop fs -get /user/cloudera/myfolder/Atharva.txt /home/cloudera
15. hadoop fs -touchz /user/cloudera/myfolder/empty.txt
16. hadoop fs -copyToLocal /user/cloudera/myfolder/empty.txt /home/cloudera
17. hadoop fs -setrep -w 1 /user/cloudera/myfolder/Atharva.txt
18. hadoop fs -chmod 700 /user/cloudera/Atharva.txt
19. hadoop fs -stat /user/cloudera/Atharva.txt
20. sudo -u hdfs hadoop fs -chgrp -R cloudera /ExptFolder
21. sudo -u hdfs hadoop fs -chown -R cloudera /ExptFolder
22. hadoop fs -lsr /user/cloudera
23. hadoop fs -dus /user/cloudera

## 7. Create and insert a post in the NOSQL database for a social media website using MongoDB.

```
data = {
  "_id": "moiz_post_1"
  "image": "image_path.jpeg",
  "caption": "This is the caption of the image"
}
collection.insert_one(data)
```

## 8. Create and update a post in NOSQL database for a social media website using MongoDB

```
key = "moiz_post_1"
new_img_path = "new_image.jpeg"
new_caption = "This is new caption"

collection.update_one(
  {"_id": key},
  {'$set': {"image": new_img_path, "caption": new_caption}}
)
```

**9. Implement FM algorithm**

**10. Implement Bloom's Filter and verify the contents are present or not.**

**11. Implement Word count using map reduce.**

**12. Write a R program to create Data frames which contain details of 5 employees and display the details.**

```
->
# Create a data frame with employee details
employee_data <- data.frame(
  EmployeeID = c(1, 2, 3, 4, 5),
  FirstName = c("John", "Mary", "Robert", "Lisa", "Michael"),
  LastName = c("Smith", "Johnson", "Williams", "Davis", "Brown"),
  Age = c(30, 25, 35, 28, 32),
  Department = c("HR", "IT", "Finance", "Marketing", "Sales")
)

# Display the employee details
print(employee_data)
```

**13. Write a R program to create a list of heterogeneous data, which include character, numeric and logical vectors. Print the lists.**

```
—> # Create character, numeric, and logical vectors
character_vector <- c("Alice", "Bob", "Charlie")
numeric_vector <- c(1, 2, 3, 4, 5)
logical_vector <- c(TRUE, FALSE, TRUE)

# Create a list containing the above vectors
my_list <- list(
  characters = character_vector,
  numbers = numeric_vector,
  logicals = logical_vector
)
# Print the list
print(my_list)
```

**14. Write a R program to create bell curve of a random normal distribution.**

```
→
# Set seed for reproducibility
set.seed(123)

# Generate random data from a normal distribution
data <- rnorm(1000, mean = 0, sd = 1)

# Create a histogram
hist(data, probability = TRUE, col = "lightblue", main = "Bell Curve - Normal Distribution")
```

```
# Add a bell curve using the curve function
curve(dnorm(x, mean = mean(data), sd = sd(data)), col = "darkred", lwd = 2, add = TRUE)
```

```
# Add labels and legend
xlabel <- "Value"
ylabel <- "Density"
title <- "Bell Curve - Normal Distribution"
legend("topright", legend = "Normal Distribution", col = "darkred", lwd = 2)
```

```
# Add labels to the plot
title(main = title)
xlabel <- "Value"
ylabel <- "Density"
title <- "Bell Curve - Normal Distribution"
legend("topright", legend = "Normal Distribution", col = "darkred", lwd = 2)
```

**15. Write a R program to create a simple bar plot of five subjects marks.**

```
→
subjects <- c("Math", "Science", "English", "History", "Art")
marks <- c(85, 90, 78, 88, 92)
barplot(marks, names.arg = subjects, col = "skyblue", main = "Subject Marks", xlab =
"Subjects", ylab = "Marks")
text(1:5, marks + 2, labels = marks, col = "blue", cex = 1)
par(las = 1)
```

**16. Write a R program to create a  $5 \times 4$  matrix,  $3 \times 3$  matrix with labels and fill the matrix by rows and  $2 \times 2$  matrix with labels and fill the matrix by columns.**

```
→
# Create a 5x4 matrix filled by rows
matrix_rows <- matrix(1:20, nrow = 5, ncol = 4, byrow = TRUE)
rownames(matrix_rows) <- c("Row1", "Row2", "Row3", "Row4", "Row5")
colnames(matrix_rows) <- c("Col1", "Col2", "Col3", "Col4")
```

```
# Create a 3x3 matrix with labels
matrix_labels <- matrix(21:29, nrow = 3, ncol = 3)
rownames(matrix_labels) <- c("R1", "R2", "R3")
colnames(matrix_labels) <- c("C1", "C2", "C3")
```

```
# Create a 2x2 matrix filled by columns
matrix_columns <- matrix(30:33, nrow = 2, ncol = 2, byrow = FALSE)
rownames(matrix_columns) <- c("RowA", "RowB")
colnames(matrix_columns) <- c("ColX", "ColY")
```

```
# Print the matrices
```

```
print("Matrix filled by rows:")  
print(matrix_rows)
```

```
print("\nMatrix with labels:")  
print(matrix_labels)
```

```
print("\nMatrix filled by columns:")  
print(matrix_columns)
```

**17. Write a R program to create a sequence of numbers from 20 to 50 and find the mean of numbers from 20 to 60 and sum of numbers from 51 to 91.**

```
->  
# Create a sequence of numbers from 20 to 50  
sequence_20_to_50 <- 20:50  
  
# Calculate the mean of numbers from 20 to 60  
mean_20_to_60 <- mean(20:60)  
  
# Find the sum of numbers from 51 to 91  
sum_51_to_91 <- sum(51:91)  
  
# Display the results  
cat("Sequence from 20 to 50:", sequence_20_to_50, "\n")  
cat("Mean of numbers from 20 to 60:", mean_20_to_60, "\n")  
cat("Sum of numbers from 51 to 91:", sum_51_to_91, "\n")
```