**Practical 1: (range)**

Create database adbms;

CREATE TABLE sales (id INT, order\_date DATE, amount DECIMAL(10, 2))

PARTITION BY RANGE (YEAR(order\_date)) (

PARTITION p0 VALUES LESS THAN (1990),

PARTITION p1 VALUES LESS THAN (2000));

INSERT INTO sales VALUES (1, '1985-05-15', 1000.50);

INSERT INTO sales VALUES (2, '1995-08-20', 1500.75);

Select \* from sales partition(p0);

**List partitioning**

Create database adbms;

CREATE TABLE employees (id INT, name VARCHAR(50), department VARCHAR(50))

PARTITION BY LIST (department) (

PARTITION p\_engineering VALUES IN ('Engineering', 'IT'),

PARTITION p\_sales VALUES IN ('Sales', 'Marketing'));

INSERT INTO employees VALUES (1, 'John Doe', 'Engineering');

INSERT INTO employees VALUES (2, 'Jane Smith', 'IT');

Select \* from employees partition(p\_sales);

**Practical 2: (**Roll\_UP, CUBE, First, Last , Lead ,Lag,Rank AND Dense Rank)

CREATE TABLE sales (

order\_date DATE,

product\_id INT,

amount\_sold DECIMAL(10, 2)

);

INSERT INTO sales VALUES ('2022-01-01', 1, 100.50);

INSERT INTO sales VALUES ('2022-01-01', 2, 150.75);

SELECT order\_date, product\_id, SUM(amount\_sold) AS total\_amount

FROM sales

GROUP BY ROLLUP(order\_date, product\_id);

SELECT order\_date, product\_id, SUM(amount\_sold) AS total\_amount

FROM sales

GROUP BY CUBE(order\_date, product\_id);

SELECT order\_date, product\_id, amount\_sold,

FIRST\_VALUE(amount\_sold) OVER (PARTITION BY order\_date ORDER BY product\_id) AS first\_amount

FROM sales;

SELECT order\_date, product\_id, amount\_sold,

LAST\_VALUE(amount\_sold) OVER (PARTITION BY order\_date ORDER BY product\_id ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS last\_amount

FROM sales;

SELECT order\_date, product\_id, amount\_sold,

LEAD(amount\_sold) OVER (PARTITION BY product\_id ORDER BY order\_date) AS next\_amount

FROM sales;

SELECT order\_date, product\_id, amount\_sold,

LAG(amount\_sold) OVER (PARTITION BY product\_id ORDER BY order\_date) AS previous\_amount

FROM sales;

SELECT order\_date, product\_id, amount\_sold,

RANK() OVER (ORDER BY amount\_sold DESC) AS sales\_rank

FROM sales;

SELECT order\_date, product\_id, amount\_sold,

DENSE\_RANK() OVER (ORDER BY amount\_sold DESC) AS dense\_sales\_rank

FROM sales;

**Practical 3:**

CREATE TABLE People (

person\_info PersonType

);

INSERT INTO People VALUES (

PersonType(1, 'John', 'Doe', TO\_DATE('1990-01-15', 'YYYY-MM-DD'))

);

SELECT

person\_info.person\_id,

person\_info.get\_full\_name() AS full\_name,

person\_info.get\_age() AS age

FROM People;

**Practical 5:**

**1. R Program to illustrate Numeric data type**

x = 5.6

print(class(x))

print(typeof(x))

print (x)

**2. R program to illustrate integer data type**

x = as.integer(5)

print(class(x))

print(typeof(x))

y = 5L

print(class(y))

print(type of(y))

**3. R program to illustrate logical data type**

x = 4

y = 3

z = x > y

print(z)

print(class(z))

print(type of(z))

**4. R program to illustrate complex data type**

x = 4 + 3i

print(class(x))

print(type of(x))

**5. R program to illustrate character data type**

char = "Mumbai University"

print(class(char))

print(type of(char))

**6. Read from textfile**

record\_data <- read.table("D:/myrfile/myrecord.txt")

head(record\_data)

**read from csv file.**

record\_data <- read.csv("D:/myrfile/data.csv")

head(record\_data)

**read from excel file**

install.packages("readxl")

library(readxl)

data <- read\_excel("D:/myrfile/data-1.xls", sheet = 1)

print(data)

**the setwd() function to assign working directory.**

setwd("D:/myrfile ")

**To check your current working directory, type**

getwd()

**In R, we can write data easily to a file, using the write.table() command.**

x <- data.frame (name = c("John", "Alice", "Bob"), department = c("Sales", "Marketing",

"Finance"))

write.table(x, file ="data.csv", sep = ",")

z <- data.frame(a = 10, b = 40, c = pi)

write.csv(z, file = "sample.csv")

**read data from mysql**

install.packages("RMySQL")

library(RMySQL)

con <- dbConnect(MySQL(), user='root', password='howareyou', dbname='prac5',

host='localhost',port=3306)

data <- data.frame( name = c("John", "Alice", "Bob"),age = c(30, 25, 35),city = c("New York",

"London", "Paris"))

insert\_query <- paste0("INSERT INTO emp1 (name, age, city) VALUES ")

values <- paste0("('", data$name, "', ", data$age, ", '", data$city, "')")

insert\_query <- paste0(insert\_query, paste(values, collapse = ", "))

dbSendQuery(con, insert\_query)

dbDisconnect(con)

**practical 6:**

**1. Load Data from CSV (make sure to make the CSV files)**

dataframe <- read.csv("file-prac6.csv")

print(dataframe)

**Naming and Renaming Variables**

1. To Rename Variables

names(dataframe)[names(dataframe) == "Name"] <- "FirstName"

print(dataframe)

**2. To Add a New Variable**

dataframe$emailaddress <- c('Alice@gmail.com', 'Bob@gmail.com', 'Charlie@gmail.com',

'Diana@gmail.com', 'Edward@gmail.com','Jasmine@gmail.com')

print(dataframe)

**Dealing with Missing Data**

1. Remove Rows with Missing Values

dataframe <- dataframe[complete.cases(dataframe), ]

print(dataframe)

**2. Fill Missing Values with Mean**

a. To calculate the missing Age

dataframe$Age <- ifelse(

is.na(dataframe$Age),

ave(dataframe$Age, FUN = function(x) mean(x, na.rm = TRUE)),

dataframe$Age

)

print(dataframe)

**b. To calculate the missing salary**

dataframe$Salary <- ifelse(

is.na(dataframe$Salary),

ave(dataframe$Salary, FUN = function(x) mean(x, na.rm = TRUE)),

dataframe$Salary

)

print(dataframe)

**#Dealing with Categorical Data**

**1. Convert Categorical to Numerical (using factorization)**

dataframe$Gender <- as.numeric(factor(dataframe$Gender))

print(dataframe)

**#Data Reduction using Subsetting**

**1. Subset Based on Condition**

subset\_data <- dataframe[dataframe$Age > 30, ]

print(subset\_data)

dataframe <- read.csv("file-prac6.csv")

print(dataframe)

**2. Random Sampling (to reduce data size for analysis)**

sample\_size <- 5

sample\_data <- dataframe[sample(nrow(dataframe), sample\_size, replace = FALSE), ]

print(sample\_data)

**practical 7:**

**add age and height in the excel sheet(age:23,21,22,25& height:150,145,135)**

install.packages("xlsx")

library(xlsx)

ageheight <- read.xlsx("F:\\rstudio1.xls",sheetName = "linear regression")

result <- lm(height~age,data=ageheight)

summary(result)

**practical 8:**

input <- mtcars[,c("am","hp","gear")]

print(head(input))

am.data=glm(formula = am ~ hp + gear,data=input,family = binomial)

print (summary(am.data))

**practical 9: (apriori algorithm)**

install.packages("arules")

library(arules)

transactions <- list(

  c("bread", "milk", "eggs","yogurt"),

  c("bread", "butter","yogurt"),

  c("milk", "butter"),

  c("bread", "milk", "butter"),

  c("bread", "milk","yogurt")

)

trans <- as(transactions, "transactions")

rules <- apriori(trans, parameter = list(supp = 0.5, conf = 0.2))

inspect(rules)

**practical 10: (k-mean)**

install.packages("ggplot2")

install.packages("cluster")

library(ggplot2)

library(cluster)

set.seed(123)

data<-data.frame(

  x=rnorm(100,mean = 0),

  y=rnorm(100,mean = 0)

)

kmeans\_result<-kmeans(data,centers = 4)

data$cluster<-as.factor(kmeans\_result$cluster)

ggplot(data,aes(x,y,color=cluster))+geom\_point()+geom\_point(data=as.data.frame(kmeans\_result$centers), aes(x,y),color="black",size=4,shape=4)

**#Agglomerative**

#install.packages("ggplot2")

#install.packages("cluster")

library(ggplot2)

library(cluster)

set.seed(123)

data<-data.frame(

  x=rnorm(100,mean = 0),

  y=rnorm(100,mean = 0)

)

hclust\_result<- hclust(dist(data))

cut\_tree\_result<-cutree(hclust\_result,k=3)

data$cluster<-as.factor(cut\_tree\_result)

ggplot(data,aes(x,y,color=cluster))+geom\_point()