Business Intelligence.

Practical 1: Import Legacy Data from Excel & load in target System

Step 1: Create a Excel Sheet.

Step 2: Power BI → Home Ribbon → Get Data → Excel → browse your data file → Select your table Sheet from navigator window.

Step 3: Load / Edit

Practical 2: Perform the Extraction, Transformation and Loading (ETL) process to construct the database in SQL Server.

Step 1: Open Power BI → Get Data → Excel → ~~Load your~~ Select your Table Sheet → EDIT

Step 2: Transform 1: Uppercase
↳ Right Click on Column → Transform → UPPERCASE

Transform 2: Sorting
↳ ~~Right Click~~ Select Column → Click on Drop down Arrow → Sort.

Transform 3: Converting INR to Currency.
↳ Add Column from Ribbion → Custom Colum → Name the Column → Enter formula (=INRColumn * 0.014)

Step 3: Apply Changes.

Practical 3: Create data staging area for selected database (Cube)

Step1: Connect to SQL Server Management Studio (Ssms) with Server type as Database Engine & in Database Windows Authentication mode.
→ Copy the Server name.

Step2: Open Business Intelligence Development Studio → (BIDS) → File → New → Project → Analysis project → Rename → OK.

Step3: Right Click on New Data source → New DataSource → New → Server → Paste the Server name → Select Database → Next → Test Connection → Next → Inherit → Next → Finish.

Step4: Right Click on New DataSource view → Next → Next → Choose the Facts table → Click on [>] button → Add related tables → Next → Finish.

Step5: Right Click on Cube → New Cube → Next → Use existing Tables → Next → Select Fact table → Next → Select all Dimensions → Next → Finish.

Step6: 1) Double click on Product Dimension → Drag & Drop Product.Name in Attributes Section.
2) Double click on Date Dimension → Drag & Drop the following Attributes to the Attributes Section.
a) FullDateUK
b) Month
c) MonthName
d) Quater
e) QuaterName
f) WeekOfMonth
g) Year

Step7: Create a hierarchy in the Order :-
→ Year
→ QuarterName
→ MonthName
→ WeekOfMonth
→ FullDateUK

Step8: Right Click on Project SolutionName → Properties → Deployment → Donot Process → Deploy All → Server Name → Apply → Ok

Step9: Deploy

Step10: Process → Run

Step11: Browse the Cube → Add ProductName to Columns → FullDateUK to Row Field → Fact Table to Detail Area.

Practical 5B: Performing MDX Queries.

Step1: ~~Connect~~ Open to SSMS

Step2: Connect to Server using Database Engine
→ File → New → New Query → MDX → Options
→ Browse → Browse your Cube.

Step3: Execute Queries.

Practical 6: What if Analysis.

a) Scenario Manager.

Step1: Make Table.

Step2: Data → What if Analysis → Scenario Manager
→ Add → Provide values → ok.

b) Data Table.

i) One Variable.

Step1: Make Table (3x3)

Step2: Make One Column table → apply formula & give the total value of above column to one cell.

Step3: Select: Table → Data → what if → Data table → Column → Select Column field → OK.

ii) Two Variable.

Step1: Make 2 tables → Apply total value of one Table on another (using formula)

Step2: Select the 2nd Table → what if → Data Table → Select Row & Column values → Ok.

c) Goal Seek.

Step1: Make a Table → Data → what if → Goal Seek → Set values.

a) set Cell (should be a calculated field)
b) To value represents an integer value for achieving the Goal
c) By Changing cell is the input value that needs to be changed in order to achieve the goal.

Practical 7 a) Decision Tree.

i) Using Party Library.

~~> install.packages("party")~~

Step1: Open R (3.5.1) and perform as follows

```
> install.packages("party", dependencies = TRUE)
> library("party")
> print(head(readingSkills))
> input.dat <- readingSkills[c(1:105),]
> png(file = "decision_tree.png")
```

> output.tree <- ctree (nativeSpeaker ~ age + shoeSize + score, data=input.dat)

> ~~plot(output~~
> plot(output.tree)
> dev.off()
> plot(output.tree)

ii) iris.

> str(iris)
> set.seet(1234)
> ind <- sample(2, nrow(iris), replace=TRUE, prob=c(0.7, 0.3))
> trainData <- iris[ind==1,]
> testData <- iris[ind==2,]
> library(party)
> ~~myFormula <- Species ~ Sepal.Length + Sepal.Height~~

> myFormula <- Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width
> iris_ctree <- ctree(myFormula, data=trainData)
> table(predict(iris_ctree), trainData$Species)
> ~~print(iris~~
> print(iris_ctree)
> plot(iris_ctree)
> plot(iris_ctree, type="simple")
> testPred <- Predict(iris_ctree, newdata=testData)
> table(testPred, testData$Species)

7b. iii) Time Series.

```
>rainfall<-c(799,1174.8, 865.1, 1334.6, 635.4, 918.5,
             685.5, 998.6, 784.2, 985, 882.8, 1071)

>rainfall.timeseries<-ts(rainfall,start(2

>rainfall.timeseries<-ts(rainfall, start=c(2012,1),
                         frequency=12)
>print(rainfall.timeseries)

>png(file="rainfall.png")
>plot(rainfal.timesries)
>dev.off()
>plot(rainfall.timeseries
```

Practical 8: Perform Data Clustering Using Clustering Algorithm.

Step1: Open R tool.

```
> install.packages('caret', dependencies=TRUE)
> install.packages('ggplot2', dependencies=TRUE)
> install.packages('lattice', dependencies=TRUE)
> library('caret')
```

Step2: choose (.csv) file

```
> data<-read.csv(file.choose(), header=FALSE)
> require("datasets")
> data("iris")
> str(iris)
> summary(iris)
> head(iris)
```

```
>iris.new <- iris[,c(1,2,3,4)]
>iris.class <- iris["Species"]
>head(iris.new)
>head(iris.class)
>normalize <- function(x){
+return (x-min(x))/(max(x)-min(x))
+}
> iris.new$Sepal.Length <- normalize(iris.new$Sepal.Length)
> iris.new$Sepal.Width <- normalize(iris.new$Sepal.Width)
>iris.new$Petal.Length<normalize(iris.new$Petal.Length)
>iris.new$Petal.Width <- normalize(iris.new$Petal.Width)
>result <- normalize
>result<-kmeans(iris.new,3)
>result$size
> result$cluster
> par(mfrow=c(2,2), mar=c(5,4,2,2))
> plot(iris.new[c(1,2)], col=result$cluster)
> plot(iris.new[c(1,2)], col=iris.class)
> plot(iris.new[c(3,4)], col=iris$cluster)
> plot(iris.new[c(3,4)], col=iris.class)
> table(result$cluster, iris.class)
```

Practical 9: Perform Linear Regression on Datawarehouse Data.

Step1: Open Power BI → GetData → Excel → Select → Edit

Step2: Add Column → XY → Formula:-
XY = Sheet1[Sales(Y)] * Sheet1[Age(X)]
Add Column → XSquare → Formula:-
XSquared = Sheet1[Age(X)] * Sheet1[Age(X)]

Add column → YSquered.
YSquared = Sheet1[Sales(Y)] * Sheet[Sales(Y)]:

Step3: RightClick on fields → Sheet → Add new measure.

1) XSum = SUM(Sheet1[Age(X)]
2) YSum = SUM(Sheet[Sales(Y)]
3) XYSum = SUM(Sheet1[XY])
4) numerator = 10 * [XYSUM] - [XSUM] * [YSum]
5) rdenominator1 = (10 * [XSquaredSum] - (XSum) ^2) * (10 * [YSquaredSum] - [Ysum] ^2)
6) ~~rel~~ rdenominator2 = ~~(10*~~ SQRT([rdenominator1])
7) ~~relvirid~~ r = DIVIDE([numerator], [rdenominator2])
8) rSquared = [r] * [r]
9) Slope = (10 * [XYSum] - [XSum] * [YSum]) / (10 * [XSquaredSum] - [XSum] ^2)
10) YIntercept = DIVIDE(([YSum] * [XSquaredSum] - [XSum] * [XYSum]), (10 * [XSquaredSum] - [XSum] ^2))
11) 10Sales = 0.9675 * 10 - 5.0895.
12) 30Sales = 0.9675 * 30 - 5.0895.
13) 50Sales = 0.9675 * 50 - 5.0895.

Step4: Click on Scatter map → Add trend line.

Practical 10: Data Analysis and Visulization using Excel-
Step1: File → Options → Addins → Manage → Com Addins
→ Go → ~~Add~~ Select All → Ok.

Step2: Insert Data from Access → Data → insert → acces
→ ~~Ok~~. Select file → Ok.

Step 3: Table → matrix → Excel (To visualize data in different forms)

Step 4: Perform Operations as desired.

Step 5: Switch visualization ⇒ bar / line graph for graphical visualization.