OVERVIEW OF THE DATASET

	Outlook	Temperature	Humidity	Wind	Golf
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rainy	Mild	High	Weak	Yes
5	Rainy	Cool	Normal	Weak	Yes
6	Rainy	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rainy	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rainy	Mild	High	Strong	No
15	Sunny	Hot	High	Weak	No

The dataset gives us 15 decisions whether to play golf or not given the information of weather condition. In this dataset, the target attribute that needs forecasting is the Yes/No values in the Golf column. The explanatory attributes are **Outlook, Temperature, Humidity and Wind.**

The job of the ID3 algorithm is **to select the most suitable attribute to be used as nodes in our decision tree.** To determine a node, I follow a 3-step approach:

1. ***Compute the Entropy for dataset Entropy***
2. ***Calculate the Average Entropy and Information Gain of Each Attribute***
3. ***The highest-gain attribute will act as the node.***

Using this approach, I find the Root Node, Decision Node and Leaf Node in order.

1. DETERMINE THE ROOT NODE

```sql
/** STEP 1: FIND THE ATTRIBUTE FOR ROOT NODE **/
/* Calculate Average Entropy of each attribute */
-- Calculate the frequency of each value
SELECT 'Outlook' Col, * INTO OtherAttribute FROM (SELECT DISTINCT Outlook Val, Golf, COUNT(*) Frequency FROM Golf GROUP BY Outlook, Golf) A
UNION SELECT 'Temperature', * FROM (SELECT DISTINCT Temperature, Golf , COUNT(*) Frequency FROM Golf GROUP BY Temperature, Golf) A
UNION SELECT 'Humidity', * FROM (SELECT DISTINCT Humidity, Golf , COUNT(*) Frequency FROM Golf GROUP BY Humidity, Golf) A
UNION SELECT 'Wind', * FROM (SELECT DISTINCT Wind, Golf , COUNT(*) Frequency FROM Golf GROUP BY Wind, Golf) A

SELECT * FROM OtherAttribute
```

Results | Messages

	Col	Val	Golf	Frequency
1	Humidity	High	No	5
2	Humidity	High	Yes	3
3	Humidity	Normal	No	1
4	Humidity	Normal	Yes	6
5	Outlook	Overcast	Yes	4
6	Outlook	Rainy	No	2
7	Outlook	Rainy	Yes	3
8	Outlook	Sunny	No	4
9	Outlook	Sunny	Yes	2
10	Temperature	Cool	No	1
11	Temperature	Cool	Yes	3
12	Temperature	Hot	No	3

In order to implement the Entropy formula, I calculated the components in the formula and store them in tables.

First, I created ‘OtherAttribute’ table, which counts the number of Yes/No decisions associated with each value of each attribute.

1. DETERMINE THE ROOT NODE

```sql
-- Caculate entropy for each values of each attribute
SELECT Col, Val, TotalFreq, -SUM(CAST(Frequency AS FLOAT)/ TotalFreq * LOG(CAST(Frequency AS FLOAT)/ TotalFreq )/LOG(2)) AS AttributeEntropy
INTO #AttributeEntropy
FROM AverageEntropy
GROUP BY Col, Val, TotalFreq

SELECT * FROM #AttributeEntropy
```

Results | Messages

	Col	Val	TotalFreq	AttributeEntropy
1	Humidity	High	8	0,954434002924965
2	Humidity	Normal	7	0,5916727785823274
3	Outlook	Overcast	4	-0
4	Outlook	Rainy	5	0,9709505944546688
5	Outlook	Sunny	6	0,9182958340544896
6	Temperature	Cool	4	0,8112781244591328
7	Temperature	Hot	5	0,9709505944546688
8	Temperature	Mild	6	0,9182958340544896
9	Wind	Strong	6	1
10	Wind	Weak	9	0,9182958340544896

Next, I calculated the Entropy of each attribute value using the formula:

$$Entropy(S) = \frac{-p}{p+n} log_2(\frac{p}{p+n}) - \frac{n}{p+n} log_2(\frac{n}{p+n})$$

1. DETERMINE THE ROOT NODE

```
40  /* Calculate the average information entropy */
41  SELECT Col, SUM((CAST(TotalFreq AS FLOAT)/15) * AttributeEntropy) AS AverageInformationEntropy into #AverageInformationEntropy
42  FROM #AttributeEntropy
43  GROUP BY Col
44
45  SELECT * FROM #AverageInformationEntropy
```

Results | Messages

	Col	AverageInformationEntropy
1	Humidity	0,7851454315650674
2	Outlook	0,690968531773352
3	Temperature	0,9073093649624542
4	Wind	0,9509775004326937

Then, I found the Average Information Entropy of each attribute

$$I(Attribute) = \sum \frac{p_i + n_i}{p + n} Entropy(A)$$

1. DETERMINE THE ROOT NODE

```sql
/* Calculate the gain for each attribute */
-- Calculate the Dataset Entropy
SELECT 'Outlook' Col, * INTO Probability FROM (SELECT DISTINCT Outlook Val, COUNT(*) Frequency, 15 Sum FROM Golf GROUP BY Outlook) A
UNION SELECT 'Temperature', * FROM (SELECT DISTINCT Temperature , COUNT(*) Frequency, 15 Sum FROM Golf GROUP BY Temperature) A
UNION SELECT 'Humidity', * FROM (SELECT DISTINCT Humidity , COUNT(*) Frequency, 15 Sum FROM Golf GROUP BY Humidity) A
UNION SELECT 'Wind', * FROM (SELECT DISTINCT Wind , COUNT(*) Frequency, 15 Sum FROM Golf GROUP BY Wind) A
UNION SELECT 'Golf', * FROM (SELECT DISTINCT Golf , COUNT(*) Frequency, 15 Sum FROM Golf GROUP BY Golf) A

SELECT 'Golf' Col, * INTO TargetAttribute FROM (SELECT DISTINCT Golf , COUNT(*) Frequency, 15 Sum FROM Golf GROUP BY Golf) A

--- Calculate the gain
DECLARE @DatasetEntropy VARCHAR(20)
SELECT @DatasetEntropy = -SUM(CAST(Frequency AS FLOAT)/ 15 * LOG(CAST(Frequency AS FLOAT)/ 15)/LOG(2)) FROM TargetAttribute
SELECT Col, @DatasetEntropy - CAST(AverageInformationEntropy AS FLOAT) AS Gain
FROM #AverageInformationEntropy

/* The information gain of attributes are: Humidity - 0.19; Outlook - 0.28; Temperature - 0.06; Wind = 0.02.
Thus, we choose Outlook - the highest-gain attribute as our Decision Tree Root Node */
```

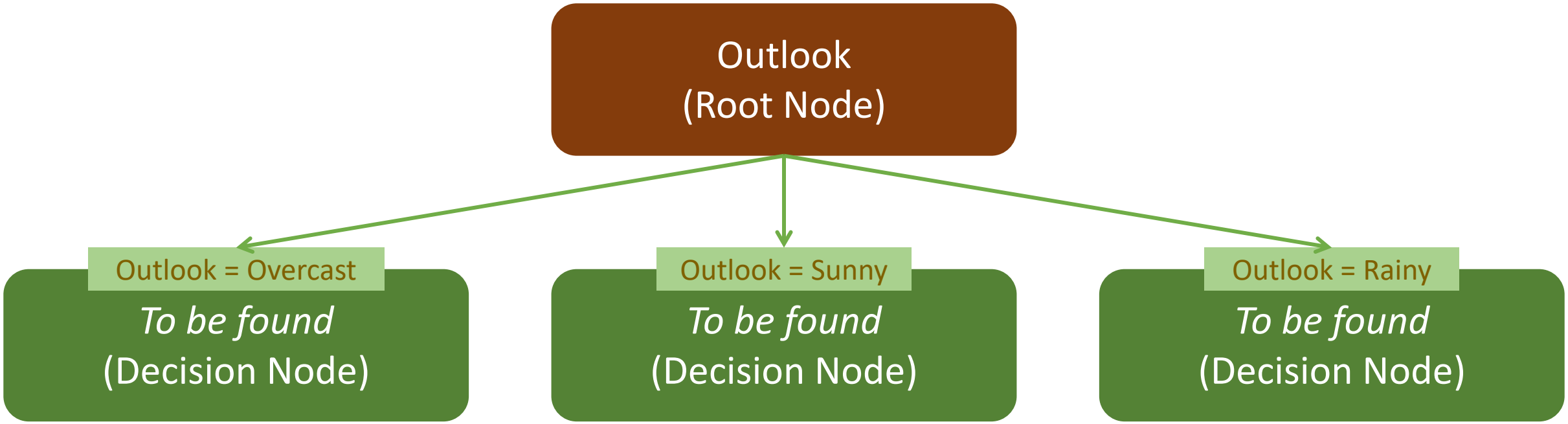
Results | Messages

	Col	Gain
1	Humidity	0,18580556843493257
2	Outlook	0,27998246822664796
3	Temperature	0,06364163503754583
4	Wind	0,01997349956730632

Finally, I calculated the Information Gain of each attribute by using:

$$Gain = Entropy(S) - I(Attribute)$$

1. DETERMINE THE ROOT NODE



2. DETERMINE THE DECISION NODE

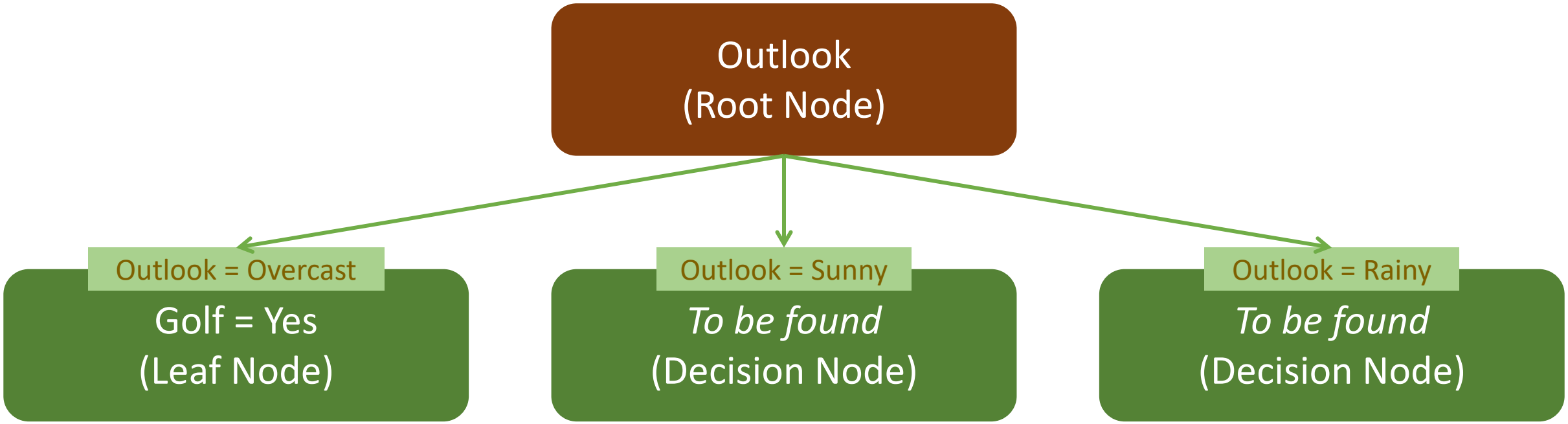
```
69   SELECT * FROM OtherAttribute WHERE Col = 'Outlook'
70   /* Scenario 1: Outlook = Overcast: When the outlook is Overcast, the Golf value is always Yes. Thus, there is no need to go further. */
```

Results | Messages

	Col	Val	Golf	Frequency
1	Outlook	Overcast	Yes	4
2	Outlook	Rainy	No	2
3	Outlook	Rainy	Yes	3
4	Outlook	Sunny	No	4
5	Outlook	Sunny	Yes	2

Given Outlook as the Root Node, I find the next attribute to be the next decision node in each of 3 scenarios corresponding to 3 values of Outlook (Sunny, Rainy and Overcast).

2. DETERMINE THE DECISION NODE



2. DETERMINE THE DECISION NODE

```
72  /* Scenario 2: Outlook = Sunny*/
73  -- Filter and save a dataset where Outlook = Sunny
74  SELECT * INTO OutlookS2
75  FROM Golf WHERE Outlook = 'Sunny'
76  SELECT * FROM OutlookS2
77
```

Results Messages

	Outlook	Temperature	Humidity	Wind	Golf
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Sunny	Mild	High	Weak	No
4	Sunny	Cool	Normal	Weak	Yes
5	Sunny	Mild	Normal	Strong	Yes
6	Sunny	Hot	High	Weak	No

Filter the dataset where Outlook takes value of ‘Sunny’ to be used in this part.

2. DETERMINE THE DECISION NODE

```sql
-- Calculate Average Entropy of each attribute
SELECT 'Temperature' Col, * INTO OutlookS2Attributes FROM (SELECT DISTINCT Temperature Val, Golf, COUNT(*) Frequency FROM OutlookS2 GROUP BY Temperature, Golf)
UNION SELECT 'Humidity', * FROM (SELECT DISTINCT Humidity, Golf , COUNT(*) Frequency FROM OutlookS2 GROUP BY Humidity, Golf) A
UNION SELECT 'Wind', * FROM (SELECT DISTINCT Wind, Golf , COUNT(*) Frequency FROM OutlookS2 GROUP BY Wind, Golf) A
SELECT * FROM OutlookS2Attributes

-- Calculate the total frequency of each value
SELECT DISTINCT Col, Val, SUM(Frequency) AS TotalFreq
INTO #OutlookS2Total
FROM OutlookS2Attributes GROUP BY Col, Val
SELECT * FROM #OutlookS2Total

SELECT OutlookS2Attributes.*, #OutlookS2Total.TotalFreq INTO OutlookS2AverageEntropy
FROM OutlookS2Attributes
LEFT JOIN #OutlookS2Total ON OutlookS2Attributes.Col = #OutlookS2Total.Col AND OutlookS2Attributes.Val = #OutlookS2Total.Val

SELECT * FROM OutlookS2AverageEntropy

-- Caculate entropy for each values of each attribute
SELECT Col, Val, TotalFreq, -SUM(CAST(Frequency AS FLOAT)/ TotalFreq * LOG(CAST(Frequency AS FLOAT)/ TotalFreq )/LOG(2)) AS AttributeEntropy
INTO #OutlookS2AttributeEntropy
FROM OutlookS2AverageEntropy
GROUP BY Col, Val, TotalFreq
SELECT * FROM #OutlookS2AttributeEntropy

-- Calculate the average information entropy
SELECT Col, SUM((CAST(TotalFreq AS FLOAT)/15) * AttributeEntropy) AS AverageInformationEntropy into #OutlookS2AverageInformationEntropy
FROM #OutlookS2AttributeEntropy
GROUP BY Col
SELECT * FROM #OutlookS2AverageInformationEntropy
```

Results | Messages

	Col	Val	Golf	Frequency	TotalFreq
1	Humidity	High	No	4	4
2	Humidity	Normal	Yes	2	2
3	Temperature	Cool	Yes	1	1
4	Temperature	Hot	No	3	3
5	Temperature	Mild	No	1	2

Following the same steps and formula in the first part.

2. DETERMINE THE DECISION NODE

```sql
-- Calculate the Dataset Entropy in Scenario 2
SELECT 'Temperature' Col, * INTO OutlookS2Prob FROM (SELECT DISTINCT Temperature Val, COUNT(*) Frequency, 6 Sum FROM OutlookS2 GROUP BY Temperature) A
UNION SELECT 'Humidity', * FROM (SELECT DISTINCT Humidity , COUNT(*) Frequency, 6 Sum FROM OutlookS2 GROUP BY Humidity) A
UNION SELECT 'Wind', * FROM (SELECT DISTINCT Wind , COUNT(*) Frequency, 6 Sum FROM OutlookS2 GROUP BY Wind) A
UNION SELECT 'Golf', * FROM (SELECT DISTINCT Golf , COUNT(*) Frequency, 6 Sum FROM OutlookS2 GROUP BY Golf) A

SELECT 'Golf' Col, * INTO OutlookS2TargetAttribute FROM (SELECT DISTINCT Golf , COUNT(*) Frequency, 6 Sum FROM OutlookS2 GROUP BY Golf) A

SELECT * FROM OutlookS2TargetAttribute

--- Calculate the gain
DECLARE @OutlookS2DatasetEntropy VARCHAR(20)
SELECT @OutlookS2DatasetEntropy = -SUM(CAST(Frequency AS FLOAT)/ 6 * LOG(CAST(Frequency AS FLOAT)/ 6)/LOG(2)) FROM OutlookS2TargetAttribute
SELECT Col, @OutlookS2DatasetEntropy - CAST(AverageInformationEntropy AS FLOAT) AS Gain
FROM #OutlookS2AverageInformationEntropy

/* The information gain of attributes are: Humidity - 0.92; Temperature - 0.78; Wind = 0.57.
Thus, we choose Humidity - the highest-gain attribute as our Decision Node in Scenario 2: Outlook = Sunny */
```

Results | Messages

	Col	Gain
1	Humidity	0,918296
2	Temperature	0,7849626666666667
3	Wind	0,5686218334775646

Results grid

These are results of Information Gain of the rest attributes when Outlook is Sunny.

2. DETERMINE THE DECISION NODE

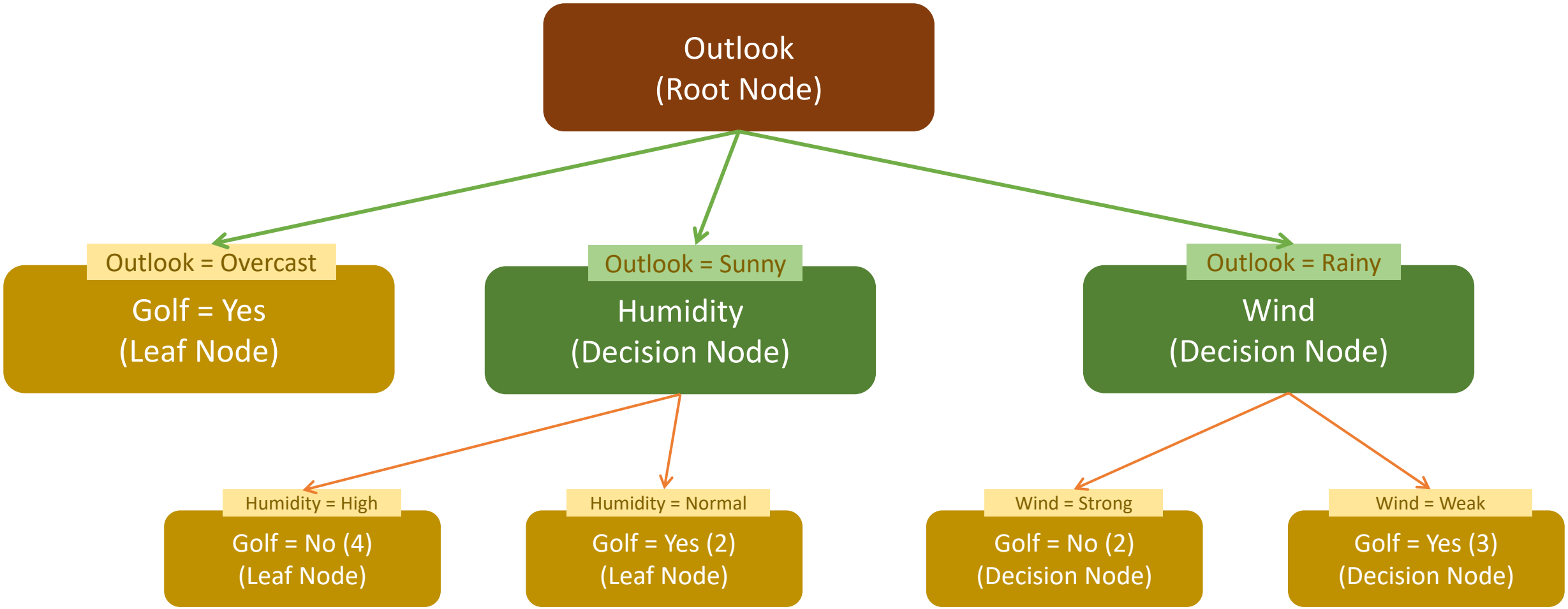
```
176   --- Calculate the gain
177   DECLARE @OutlookS3DatasetEntropy VARCHAR(20)
178   SELECT @OutlookS3DatasetEntropy = -SUM(CAST(Frequency AS FLOAT)/ 5 * LOG(CAST(Frequency AS FLOAT)/ 5)/LOG(2)) FROM OutlookS3TargetAttribute
179   SELECT Col, @OutlookS3DatasetEntropy - CAST(AverageInformationEntropy AS FLOAT) AS Gain
180   FROM #OutlookS3AverageInformationEntropy
181
182   /* The information gain of attributes are: Humidity - 0.65; Temperature - 0.365 Wind = 097.
183   Thus, we choose Wind - the highest-gain attribute as our Decision Node in Scenario 3: Outlook = Rainy */
```

Results | Messages

	Col	Gain
1	Humidity	0,6539584998557688
2	Temperature	0,6539584998557688
3	Wind	0,970951

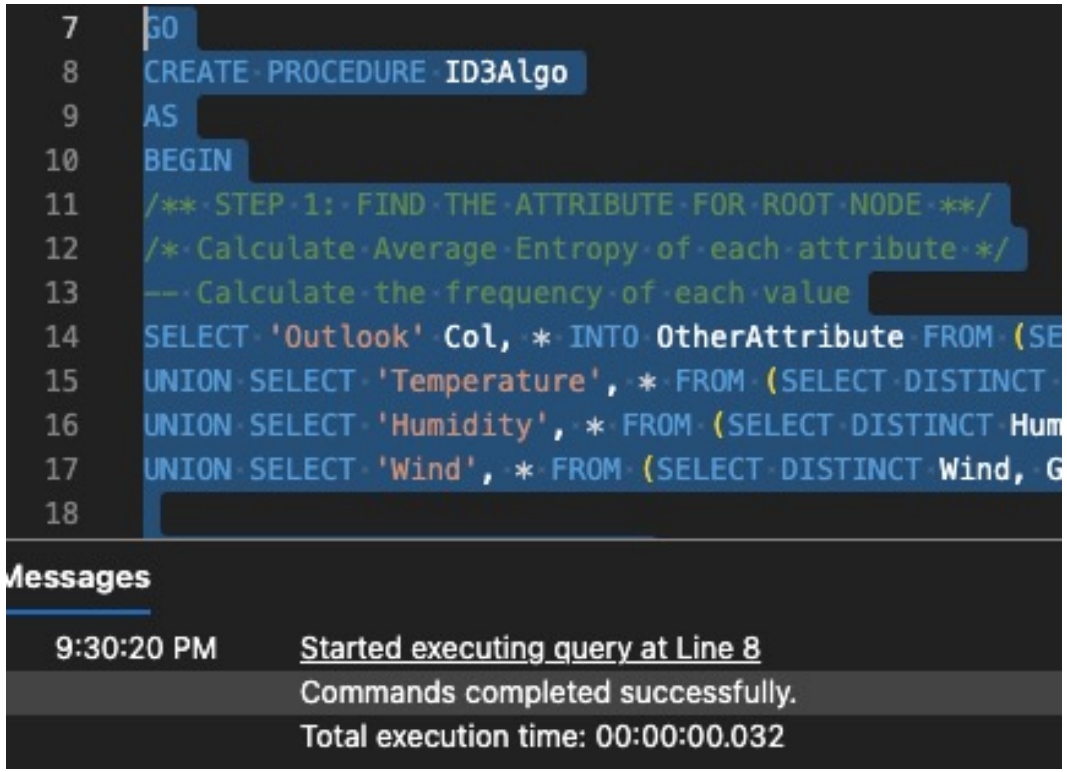
Doing the same for the scenario where Outlook is Rainy, I got these results.

2. DETERMINE THE DECISION NODE



3. CREATE A STORED PROCEDURE

```
/* ID3 Algorithm Implementation:
Step 1: Find the Root Node by calculating entropy and information gain of each attribute and choose the one with the most gain
Step 2: Given the Root Node, continue finding the Decision Nodes by repeating the same mechanism */
GO
CREATE PROCEDURE ID3Algo
AS
BEGIN
```



Successfully created
a stored procedure.