Untitled

January 15, 2024

In [7]:
```
#QUESTION 1

from pyspark.sql import SparkSession
from pyspark.sql.functions import col

# Create a SparkSession object
spark = SparkSession.builder.appName("DataFrameExample").getOrCreate()

# Create a sample DataFrame
data = [("pencil", 10), ("notebook", 30), ("paint", 35)]
df = spark.createDataFrame(data, ["item", "price"])

# Apply filter transformation
df_filtered = df.filter(col("price") > 25)

# Apply withColumn transformation
df_with_column = df_filtered.withColumn("price after applying GST", col("price") + 22.5

# Display the final DataFrame
df_with_column.show()
```

```
+--------+-----+-----------------------+
|    item|price|price after applying GST|
+--------+-----+-----------------------+
|notebook|   30|                   52.5|
|   paint|   35|                   57.5|
+--------+-----+-----------------------+
```

In [10]:
```
# QUESTION 2

from pyspark.sql import SparkSession
from pyspark.sql.functions import col


# Create SparkSession
spark = SparkSession.builder.appName('PySpark').getOrCreate()
```

```
        # Create DataFrame
        data = [("pencil", 10), ("notebook", 30), ("paint", 35), ("fevicol", 28)]
        columns = ["item", "price"]
        df = spark.createDataFrame(data, columns)

        # Count rows in DataFrame
        rows = df.count()
        print(f"The number of rows in the DataFrame is {rows}.")

        # Show DataFrame
        df.show()
```

```
The number of rows in the DataFrame is 4.
+--------+-----+
|    item|price|
+--------+-----+
|  pencil|   10|
|notebook|   30|
|   paint|   35|
| fevicol|   28|
+--------+-----+
```

In [17]:
```
#QUESTION 3

        from pyspark.sql import SparkSession
        from pyspark.sql.functions import sum, avg

        # Create SparkSession
        spark = SparkSession.builder.appName('PySpark').getOrCreate()

        # Create DataFrame
        data = [("Alice", "Sales", 3000), ("Bob", "IT", 4000), ("Charlie", "Sales", 3500), ("
        columns = ["Name", "Department", "Salary"]
        df = spark.createDataFrame(data, columns)

        # Display the DataFrame
        df.show()

        # Total salary by department
        total_salary = df.groupBy("Department").agg(sum("Salary"))

        # Display the result
        print("Total salary by department:")
        total_salary.show()
```

```
        # Average salary by department
        average_salary = df.groupBy("Department").agg(avg("Salary"))

        # Display the result
        print("Average salary by department:")
        average_salary.show()

        # Stop the Spark session
        spark.stop()
```

```
+-------+----------+------+
|   Name|Department|Salary|
+-------+----------+------+
|  Alice|     Sales|  3000|
|    Bob|        IT|  4000|
|Charlie|     Sales|  3500|
|  David|        IT|  4500|
+-------+----------+------+

Total salary by department:
+----------+-----------+
|Department|sum(Salary)|
+----------+-----------+
|     Sales|       6500|
|        IT|       8500|
+----------+-----------+

Average salary by department:
+----------+-----------+
|Department|avg(Salary)|
+----------+-----------+
|     Sales|     3250.0|
|        IT|     4250.0|
+----------+-----------+
```

In [19]:
```
        #QUESTION 4
        from pyspark.sql import SparkSession

        # Create SparkSession
        spark = SparkSession.builder.appName('PySpark').getOrCreate()

        # Create DataFrame
        data = [("Alice", "Sales", 3000), ("Bob", "IT", 4000), ("Charlie", "Sales", 3500), ("
        columns = ["Name", "Department", "Salary"]
        df = spark.createDataFrame(data, columns)
```

```
# Write DataFrame to CSV file
df.write.csv("/home/lplab/Desktop/210962018/q4.csv")
```

1 question 1,2,4 combined

```
In [1]: from pyspark.sql import SparkSession
        from pyspark.sql.functions import col, sum, avg

        # Create SparkSession
        spark = SparkSession.builder.appName('PySpark').getOrCreate()

        # Example: DataFrame with filter and withColumn transformations
        data = [("Aryan", "backend", 75000), ("rohit", "frontend", 47000), ("yashveer", "fronte
        columns = ["Name", "Department", "Salary"]
        df = spark.createDataFrame(data, columns)
        df.show()
        # Apply filter transformation
        df_filtered = df.filter(col("Salary") > 50000)  # Corrected missing parenthesis

        # Apply withColumn transformation
        df_with_column = df_filtered.withColumn("Salary after cutting GST", col("Salary") - 570
        df_with_column.show()
        # Count rows in DataFrame
        rows = df.count()
        print(f"The number of rows in the DataFrame is {rows}.")

        # Specify the output CSV file path
        output_path = "/home/lplab/Desktop/210962018/output.csv"

        # Write DataFrame to CSV
        df.write.csv(output_path, header=True, mode="overwrite")  # Corrected df3 to df

        # Stop the Spark session
        spark.stop()
```

```
/home/lplab/anaconda3/lib/python3.7/site-packages/pyspark/context.py:317: FutureWarning: Python
  warnings.warn("Python 3.7 support is deprecated in Spark 3.4.", FutureWarning)


+--------+----------+------+
|    Name|Department|Salary|
+--------+----------+------+
|   Aryan|   backend| 75000|
|   rohit|  frontend| 47000|
|yashveer|  frontend| 55000|
|   rehan|   backend| 65000|
+--------+----------+------+
```

```
+--------+----------+------+-----------------------+
|    Name|Department|Salary|Salary after cutting GST|
+--------+----------+------+-----------------------+
|   Aryan|   backend| 75000|                69299.72|
|yashveer|  frontend| 55000|                49299.72|
|   rehan|   backend| 65000|                59299.72|
+--------+----------+------+-----------------------+

The number of rows in the DataFrame is 4.
```

```
In [2]: #QUESTION 5
        # QUESTION 5
        from pyspark.sql import SparkSession
        from pyspark.sql.functions import explode, split

        # Create SparkSession
        spark = SparkSession.builder.appName('WordCountExample').getOrCreate()

        # Sample text data
        text_data = ["malpe beach","mattu beach","baga beach", "hoode beach"]

        # Create a DataFrame from the text data
        df = spark.createDataFrame([(line,) for line in text_data], ["text"])

        # Split the text into words using space as a delimiter and explode the array of words
        word_count = df.select(explode(split("text", " ")).alias("word")).groupBy("word").count

        # Display the result
        print("Word Count:")
        word_count.show()

        # Stop the Spark session
        spark.stop()
```

```
/home/lplab/anaconda3/lib/python3.7/site-packages/pyspark/context.py:317: FutureWarning: Python
  warnings.warn("Python 3.7 support is deprecated in Spark 3.4.", FutureWarning)


Word Count:
+-----+-----+
| word|count|
+-----+-----+
|beach|    4|
|malpe|    1|
|mattu|    1|
| baga|    1|
|hoode|    1|
```

```
+-----+-----+
```

In [ ]: