Harshil Patel (0501)

Class Activity:

```
#Exercise: Create a new column named “PayLevel” that has values as 1, 2 or 3. The value of paylevel is 1, if dailyrate>1000 ; 2 if it is > 50

def pay_level(rate):
    if rate > 1000:
        return 1
    elif rate > 500:
        return 2
    else:
        return 3

pay_udf = udf(pay_level, IntegerType())

result_with_pay_level = hrdata.withColumn('PayLevel', pay_udf(hrdata['DailyRate']))
result_with_pay_level.select("DailyRate", "PayLevel").show(5)
```

```
+---------+--------+
|DailyRate|PayLevel|
+---------+--------+
|     1102|       1|
|      279|       3|
|     1373|       1|
|     1392|       1|
|      591|       2|
+---------+--------+
only showing top 5 rows
```

```
!pip install pyspark
```

```
Collecting pyspark
  Downloading pyspark-3.5.1.tar.gz (317.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 317.0/317.0 MB 2.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.5.1-py2.py3-none-any.whl size=317488493 sha256=4e3555d6eba2fd8837cd17b624f80501dbd0ebfb4
  Stored in directory: /root/.cache/pip/wheels/80/1d/60/2c256ed38dddce2fdd93be545214a63e02fbd8d74fb0b7f3a6
Successfully built pyspark
Installing collected packages: pyspark
Successfully installed pyspark-3.5.1
```

```
from pyspark.sql.session import SparkSession
session=SparkSession.builder.appName("employeesql").master("local[2]").getOrCreate()
```

A spark data frame can be created by reading a csv file using the spark session as demonstrated in the following example. The argument header is considered for specifying whether the file contains a header or not. The schema can also be inferred depending on the value of “inferSchema”.

```
hrdata=session.read.csv("EmployeeAttrition.csv", header=True, inferSchema=True)
```

```
#Displaying the first 2 rows of the dataset
hrdata.show(2, truncate=False)

#View the selected columns of some rows
hrdata.select("Gender", "Age", "Attrition").show(5)
```

```
+---+---------+-----------------+---------+----------------------+----------------+---------+--------------+-------------+--------------
|Age|Attrition|BusinessTravel   |DailyRate|Department            |DistanceFromHome|Education|EducationField|EmployeeCount|EmployeeNumber
+---+---------+-----------------+---------+----------------------+----------------+---------+--------------+-------------+--------------
|41 |Yes      |Travel_Rarely    |1102     |Sales                 |1               |2        |Life Sciences |1            |1
|49 |No       |Travel_Frequently|279      |Research & Development|8               |1        |Life Sciences |1            |2
+---+---------+-----------------+---------+----------------------+----------------+---------+--------------+-------------+--------------
only showing top 2 rows

+------+---+---------+
|Gender|Age|Attrition|
+------+---+---------+
|Female| 41|      Yes|
|  Male| 49|       No|
|  Male| 37|      Yes|
|Female| 33|       No|
|  Male| 27|       No|
+------+---+---------+
only showing top 5 rows
```

```
#Displaying Data type
print("Data type :",type(hrdata))

#Displaying the information of the columns
print("Information of columns:\n",hrdata.columns)

#Displaying the information of number of columns
print("Number of columns:",len(hrdata.columns))

#Displaying Data types of columns
print("Data Types of all columns:",hrdata.dtypes)

#Displaying the first records
print("Details of first record:\n",hrdata.head(1))

#Displaying last two records
print("Last two records:",hrdata.tail(2))

#Displaying the first record using first() action
print("First record:\n",hrdata.first())

#Displaying the first two records using take() action
print("First two records:\n",hrdata.take(2))

#Displaying the number of records using count() action
print("Total Number of records:",hrdata.count())

#Displaying statistics of all columns
print("Information of dataset:\n",hrdata.describe().show())

#Displaying statistics of selected columns
print("Information:",hrdata.describe('Department','MonthlyRate').show())
```

```
Data type : <class 'pyspark.sql.dataframe.DataFrame'>
Information of columns:
 ['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department', 'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount', '
Number of columns: 35
Data Types of all columns: [('Age', 'int'), ('Attrition', 'string'), ('BusinessTravel', 'string'), ('DailyRate', 'int'), ('Department',
Details of first record:
 [Row(Age=41, Attrition='Yes', BusinessTravel='Travel_Rarely', DailyRate=1102, Department='Sales', DistanceFromHome=1, Education=2, Educ
Last two records: [Row(Age=49, Attrition='No', BusinessTravel='Travel_Frequently', DailyRate=1023, Department='Sales', DistanceFromHome=
First record:
 Row(Age=41, Attrition='Yes', BusinessTravel='Travel_Rarely', DailyRate=1102, Department='Sales', DistanceFromHome=1, Education=2, Educa
First two records:
 [Row(Age=41, Attrition='Yes', BusinessTravel='Travel_Rarely', DailyRate=1102, Department='Sales', DistanceFromHome=1, Education=2, Educ
Total Number of records: 1470
+-------+------------------+---------+--------------+------------------+---------------+----------------+------------------+------------
|summary|               Age|Attrition|BusinessTravel|         DailyRate|     Department|DistanceFromHome|         Education|  EducationF
+-------+------------------+---------+--------------+------------------+---------------+----------------+------------------+------------
|  count|              1470|     1470|          1470|              1470|           1470|            1470|              1470|
|   mean|36.923809523809524|     NULL|          NULL| 802.4857142857143|           NULL|9.19251700680272| 2.912925170068027|
| stddev| 9.135373489136729|     NULL|          NULL|403.50909994352804|           NULL|8.10686443566608|1.0241649445978718|
|    min|                18|       No|    Non-Travel|               102|Human Resources|               1|                 1| Human Resou
|    max|                60|      Yes| Travel_Rarely|              1499|          Sales|              29|                 5|Technical De
+-------+------------------+---------+--------------+------------------+---------------+----------------+------------------+------------

Information of dataset:
 None
+-------+---------------+------------------+
|summary|     Department|       MonthlyRate|
+-------+---------------+------------------+
|  count|           1470|              1470|
|   mean|           NULL|14313.103401360544|
| stddev|           NULL| 7117.786044059972|
|    min|Human Resources|              2094|
|    max|          Sales|             26999|
+-------+---------------+------------------+

Information: None
```

Column Transformations: The column transformations are related to adding, deleting and renaming a new column, deleting duplicate rows etc. The withColumn() helps to add a new column according to the specified information. A powerful feature of the withColumn() is that it also helps to add values to a new column with a constant value or from the existing column or using the user defined function. The withColumnRenamed() helps to rename an existing column. The drop() helps to delete the column.

```
#Adding a new column named new daily rate
hrdata=hrdata.withColumn('new_Rate',hrdata['DailyRate']+50)
print("Number of columns after adding:",len(hrdata.columns))
hrdata.select('DailyRate', 'new_Rate').show(2)

#Renaming the column named "new_Rate"
hrdata=hrdata.withColumnRenamed('new_Rate','new_DailyRate')
hrdata.select('new_DailyRate').show(2)

#Using drop() function to delete a column or multiple columns,
hrdata=hrdata.drop('new_DailyRate')
print("Number of columns after deleting:",len(hrdata.columns))

#Writing to a csv file
newhrdata=hrdata.drop('new_DailyRate', 'Attrition','DailyRate')
newhrdata.coalesce(1).write.format('csv').option('header','true').save("newhrdata.csv")
```

```
Number of columns after adding: 36
+---------+--------+
|DailyRate|new_Rate|
+---------+--------+
|     1102|    1152|
|      279|     329|
+---------+--------+
only showing top 2 rows

+-------------+
|new_DailyRate|
+-------------+
|         1152|
|          329|
+-------------+
only showing top 2 rows

Number of columns after deleting: 35
```

```
#Adding a new column based on the user defined function
from pyspark.sql.functions import udf
from pyspark.sql.types import *

#Creating a function
def sat_func(level):
    if level> 3:
        return 'High Satisfaction'
    elif level>2:
        return 'Average Satisfaction'
    else:
        return 'Less Satisfaction'

#Creating a user defined function
sat_udf=udf(sat_func,StringType())

#Adding a new column containing value based on user defined function
result2=hrdata.withColumn('Satisfaction_Level',sat_udf(hrdata['JobSatisfaction']))

#Displaying the results
result2.select("JobSatisfaction","Satisfaction_Level").show(5)
```

```
+---------------+--------------------+
|JobSatisfaction|  Satisfaction_Level|
+---------------+--------------------+
|              4|   High Satisfaction|
|              2|   Less Satisfaction|
|              3|Average Satisfaction|
|              3|Average Satisfaction|
|              2|   Less Satisfaction|
+---------------+--------------------+
only showing top 5 rows
```

```
#Using user defined function for categorical input variable
from pyspark.sql.functions import udf
from pyspark.sql.types import *

def jobfunc(role):
    if role in ['Research Director','Manufacturing Director']:
        return 'Senior Position'
    elif role in ['Sales Executive','Sales Representative']:
        return 'Sales'
    elif role in ['Manager']:
        return 'Manager'
    else:
        return 'Others'

job_udf=udf(jobfunc,StringType())
result3=hrdata.withColumn('NewJob',job_udf(hrdata['JobRole']))
result3.select("JobRole","NewJob").show(3, truncate=False)
```

```
+---------------------+------+
|JobRole              |NewJob|
+---------------------+------+
|Sales Executive      |Sales |
|Research Scientist   |Others|
|Laboratory Technician|Others|
+---------------------+------+
only showing top 3 rows
```

```
#Using Basic SQL functions in PySpark
from pyspark.sql.functions import count, sum, max, min, countDistinct, mean, length, stddev,kurtosis, first, last, skewness, collect_list, v

#Displaying the values of first and last record for "MaritalStatus column"
hrdata.select(first("MaritalStatus"),last("MaritalStatus")).show()

#Displaying the minimum, maximum and mean of "Age" column
hrdata.select(min("Age"), max("Age"), mean("Age")).show()

#Displaying the count and total of "MonthlyRate" column
hrdata.select(count("MonthlyRate"),sum("MonthlyRate")).show()

#Displaying the skewness and kurtosis of "DailyRate" column
hrdata.select(skewness("DailyRate"), kurtosis("DailyRate")).show()

#Displaying the standard deviation and variance of "HourlyRate" column
hrdata.select(stddev("HourlyRate"),variance("HourlyRate")).show()

#Displaying the first and last record of "MaritalStatus" column
hrdata.select(first("MaritalStatus"),last("MaritalStatus")).show()

#Displaying the entire list of items of "Department" column
hrdata.select(collect_list("Department")).show()

#Displaying length of string
hrdata.select(length('JobRole')).show(2)

#Displaying the number of unique categories of "JobRole" column
hrdata.select(countDistinct("JobRole")).show()

#Computing Correlation between Job Level and Daily Rate
hrdata.select(corr('DailyRate','JobLevel')).show()
```

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

+--------+--------+------------------+
|min(Age)|max(Age)|          avg(Age)|
+--------+--------+------------------+
|      18|      60|36.923809523809524|
+--------+--------+------------------+
+------------------+----------------+

+--------------------+-------------------+
| skewness(DailyRate)|kurtosis(DailyRate)|
+--------------------+-------------------+
|-0.00351497695829...|-1.2038109279028495|
+--------------------+-------------------+

+------------------+--------------------+
|stddev(HourlyRate)|var_samp(HourlyRate)|
+------------------+--------------------+
|20.329427593996176|  413.28562629953313|
+------------------+--------------------+

+--------------------+-------------------+
|first(MaritalStatus)|last(MaritalStatus)|
+--------------------+-------------------+
|              Single|            Married|
+--------------------+-------------------+

+------------------------+
|collect_list(Department)|
+------------------------+
|    [Sales, Research ...|
+------------------------+

+---------------+
|length(JobRole)|
+---------------+
|             15|
|             18|
+---------------+
only showing top 2 rows

+-----------------------+
|count(DISTINCT JobRole)|
+-----------------------+
|                      9|
+-----------------------+

+-------------------------+
|corr(DailyRate, JobLevel)|
+-------------------------+
|     0.002966334855111...|
+-------------------------+
```

```
#Filtering Records
#Filter records on the basis of categorical variable
hrdata.filter(hrdata['Attrition']=="No").select('Age', 'Attrition', 'Department').show(2)

#Filter records on the basis of continuous variable
hrdata.filter(hrdata['DailyRate']>=1000).select('DailyRate','Attrition','JobRole').show(2)

#Filter records considering multiple conditions using and
hrdata.filter((hrdata['Age']>=55)&(hrdata['Attrition']=="No")&(hrdata['Department']=="Sales")).select('Age','Department','DailyRate','Attrit

#Filter records considering multiple conditions using or
hrdata.filter((hrdata['Age']>=59)|(hrdata['DailyRate']>=1400)).select('Age','Department','DailyRate','Attrition','JobRole').show(2)

#Filter records considering multiple conditions using and, not
hrdata.filter((hrdata['Age']>59) & (hrdata['MaritalStatus']!="Married") & (hrdata['DailyRate']>1000)). select("Age", "MaritalStatus", "Daily
```

```
+---+---------+--------------------+
|Age|Attrition|          Department|
+---+---------+--------------------+
| 49|       No|Research & Develo...|
| 33|       No|Research & Develo...|
+---+---------+--------------------+
only showing top 2 rows

+---------+---------+--------------------+
|DailyRate|Attrition|             JobRole|
+---------+---------+--------------------+
|     1102|      Yes|     Sales Executive|
|     1373|      Yes|Laboratory Techni...|
+---------+---------+--------------------+
only showing top 2 rows

+---+----------+---------+---------+---------------+
|Age|Department|DailyRate|Attrition|        JobRole|
+---+----------+---------+---------+---------------+
| 59|     Sales|     1435|       No|Sales Executive|
| 59|     Sales|     1225|       No|Sales Executive|
+---+----------+---------+---------+---------------+
only showing top 2 rows

+---+--------------------+---------+---------+--------------------+
|Age|          Department|DailyRate|Attrition|             JobRole|
+---+--------------------+---------+---------+--------------------+
| 59|Research & Develo...|     1324|       No|Laboratory Techni...|
| 44|Research & Develo...|     1459|       No|Healthcare Repres...|
+---+--------------------+---------+---------+--------------------+
only showing top 2 rows

+---+-------------+---------+
|Age|MaritalStatus|DailyRate|
+---+-------------+---------+
| 60|       Single|     1179|
+---+-------------+---------+
```

```
#ORDER BY CLAUSE
#Display records in ascending order
hrdata.orderBy("DailyRate").select("DailyRate","Gender").show(3)

#Display records in descending order
hrdata.orderBy("MonthlyRate", ascending =False).select("Age", "MonthlyRate","DailyRate", "Gender").show(3)

#Display records in ascending order on the basis of multiple columns
hrdata.orderBy("Age", "JobRole","DailyRate").select("Age","JobRole","DailyRate").show(3)

#Display records in descending order on the basis of multiple columns
hrdata.orderBy("Age","Department","MonthlyRate", ascending=False).select("Age","Department","MonthlyRate").show(3)
```

```
+---------+------+
|DailyRate|Gender|
+---------+------+
|      102|Female|
|      103|  Male|
|      104|Female|
+---------+------+
only showing top 3 rows

+---+-----------+---------+------+
|Age|MonthlyRate|DailyRate|Gender|
+---+-----------+---------+------+
| 20|      26999|     1362|  Male|
| 27|      26997|      511|Female|
| 29|      26968|     1107|Female|
+---+-----------+---------+------+
only showing top 3 rows

+---+--------------------+---------+
|Age|             JobRole|DailyRate|
+---+--------------------+---------+
| 18|Laboratory Techni...|      230|
| 18|Laboratory Techni...|      247|
| 18|Laboratory Techni...|     1124|
+---+--------------------+---------+
only showing top 3 rows

+---+----------+-----------+
|Age|Department|MonthlyRate|
+---+----------+-----------+
| 60|     Sales|      11924|
| 60|     Sales|      10893|
| 60|     Sales|       2845|
+---+----------+-----------+
only showing top 3 rows
```

```
#Group by function on single categorical variable
#Displaying the total daily rate on the basis of department.
hrdata.groupBy("Department").sum("DailyRate").show()

#Displaying the maximum monthly rate on the basis of education field.
hrdata.groupBy("EducationField").max("MonthlyRate").show()

#Displaying the minimum daily rate on the basis of gender
hrdata.groupBy("Gender").min("DailyRate").show()

#Displaying the mean age for different job roles
hrdata.groupBy("JobRole").mean("Age").show()

#Displaying number of observations for business travel options
hrdata.groupBy("BusinessTravel").count().show()
```

```
+--------------------+--------------+
|          Department|sum(DailyRate)|
+--------------------+--------------+
|               Sales|        356923|
|Research & Develo...|        775384|
|     Human Resources|         47347|
+--------------------+--------------+

+----------------+----------------+
|  EducationField|max(MonthlyRate)|
+----------------+----------------+
|Technical Degree|           26849|
|           Other|           26537|
|       Marketing|           26959|
|         Medical|           26999|
|   Life Sciences|           26968|
| Human Resources|           25811|
+----------------+----------------+

+------+--------------+
|Gender|min(DailyRate)|
+------+--------------+
|Female|           102|
|  Male|           103|
+------+--------------+

+--------------------+------------------+
|             JobRole|          avg(Age)|
+--------------------+------------------+
|     Sales Executive| 36.88957055214724|
|Manufacturing Dir...|38.296551724137935|
|Laboratory Techni...|  34.0965250965251|
|Sales Representative| 30.36144578313253|
|Healthcare Repres...| 39.80916030534351|
|  Research Scientist|34.236301369863014|
|             Manager| 46.76470588235294|
|   Research Director|              44.0|
|     Human Resources|              35.5|
+--------------------+------------------+

+-----------------+-----+
|   BusinessTravel|count|
+-----------------+-----+
|Travel_Frequently|  277|
|       Non-Travel|  150|
|    Travel_Rarely| 1043|
+-----------------+-----+
```

```
#Group by on multiple categorical variables
#Grouping using sum () function on the basis of Department and Gender
hrdata.groupBy("Department", "Gender").sum("DailyRate").show()

#Grouping using average for multiple columns
hrdata.groupBy("Attrition", "Gender", "MaritalStatus").avg("MonthlyRate").show()

#Grouping using count for multiple columns
hrdata.groupBy("Department","Gender","Attrition").count().show()
```

```
+--------------------+------+--------------+
|          Department|Gender|sum(DailyRate)|
+--------------------+------+--------------+
|Research & Develo...|Female|        298980|
|     Human Resources|Female|         17802|
|               Sales|  Male|        198440|
|Research & Develo...|  Male|        476404|
|     Human Resources|  Male|         29545|
|               Sales|Female|        158483|
+--------------------+------+--------------+

+---------+------+-------------+------------------+
|Attrition|Gender|MaritalStatus|  avg(MonthlyRate)|
+---------+------+-------------+------------------+
|      Yes|  Male|     Divorced|15110.541666666666|
|       No|Female|       Single|14958.631578947368|
|      Yes|Female|     Divorced|17565.444444444445|
|       No|Female|      Married|14684.933609958507|
|      Yes|  Male|       Single|13983.561643835616|
|       No|  Male|       Single|14641.585858585859|
|       No|Female|     Divorced|14005.898148148148|
|      Yes|  Male|      Married|14412.377358490567|
|       No|  Male|      Married|13561.020114942528|
|       No|  Male|     Divorced|14225.908602150537|
|      Yes|Female|       Single|15220.595744680852|
|      Yes|Female|      Married|13864.193548387097|
+---------+------+-------------+------------------+

+--------------------+------+---------+-----+
|          Department|Gender|Attrition|count|
+--------------------+------+---------+-----+
|Research & Develo...|  Male|       No|  492|
|     Human Resources|  Male|       No|   37|
|Research & Develo...|Female|       No|  336|
|               Sales|Female|       No|  151|
|     Human Resources|Female|      Yes|    6|
|               Sales|  Male|      Yes|   54|
|     Human Resources|Female|       No|   14|
|Research & Develo...|Female|      Yes|   43|
|               Sales|Female|      Yes|   38|
|     Human Resources|  Male|      Yes|    6|
|               Sales|  Male|       No|  203|
|Research & Develo...|  Male|      Yes|   90|
+--------------------+------+---------+-----+
```

```
#Aggregate functions
hrdata.agg({"Age":"min", "DailyRate":"sum", "Department":"count", "HourlyRate":"sum"}).show(truncate=False)

#Group by and aggregate functions together
hrdata.groupBy("Department").agg({"DailyRate":"sum", "Age":"min","MonthlyRate":"max"}).show(truncate=False)
```

```
+--------------+-----------------+--------+---------------+
|sum(DailyRate)|count(Department)|min(Age)|sum(HourlyRate)|
+--------------+-----------------+--------+---------------+
|1179654       |1470             |18      |96860          |
+--------------+-----------------+--------+---------------+

+----------------------+--------------+----------------+--------+
|Department            |sum(DailyRate)|max(MonthlyRate)|min(Age)|
+----------------------+--------------+----------------+--------+
|Sales                 |356923        |26997           |18      |
|Research & Development|775384        |26999           |18      |
|Human Resources       |47347         |26894           |19      |
+----------------------+--------------+----------------+--------+
```

```
#Display the age of male employees who are manager belonging to human resources department and have not left the organization
hrdata.filter((hrdata['Department']=='Human Resources')&(hrdata['JobRole']=='Manager')&(hrdata['Attrition']=="No")&(hrdata['Gender']=="Male"

#Display the total daily rate of male divorced employees possessing different education fields
hrdata.filter((hrdata['Gender']=="Male")&(hrdata['MaritalStatus']=="Divorced")).groupBy("EducationField").sum("DailyRate").show()

#Display maximum monthly rate and total daily rate of single or divorced employees who left the organization belonging to different job role
hrdata.filter((hrdata['Attrition']=="Yes")&((hrdata['MaritalStatus']=="Single")|(hrdata['MaritalStatus']=="Divorced"))).groupBy("JobRole").a

#Display average daily rate of male and female Sales Executives of different marital status belonging to sales department who have not left
hrdata.filter((hrdata['JobRole']=='Sales Executive')&(hrdata['Attrition']=="No")&(hrdata['Department']=="Sales")).groupBy("Gender","MaritalS

#Display the minimum monthly rate and maximum daily rate of married managers belonging to different departments and genders
hrdata.filter((hrdata['JobRole']=="Manager")&(hrdata['MaritalStatus']=="Married")).groupBy("Department", "Gender").agg({"DailyRate":"max", "
     +---+---------------+------+---------+-------+
     |Age|     Department|Gender|Attrition|JobRole|
     +---+---------------+------+---------+-------+
     | 50|Human Resources|  Male|       No|Manager|
     | 41|Human Resources|  Male|       No|Manager|
     +---+---------------+------+---------+-------+
     only showing top 2 rows

     +----------------+--------------+
     |  EducationField|sum(DailyRate)|
     +----------------+--------------+
     |Technical Degree|         11709|
     |           Other|         10894|
     |       Marketing|         17187|
     |         Medical|         56600|
     |   Life Sciences|         74326|
     | Human Resources|          3995|
     +----------------+--------------+

     +-------------------------+--------------+----------------+
     |JobRole                  |sum(DailyRate)|max(MonthlyRate)|
     +-------------------------+--------------+----------------+
     |Sales Executive          |27440         |26959           |
     |Manufacturing Director   |3469          |25150           |
     |Laboratory Technician    |33747         |26619           |
     |Sales Representative     |17921         |26820           |
     |Healthcare Representative|1906          |22930           |
     |Research Scientist       |22732         |26999           |
     |Manager                  |1449          |2493            |
     |Research Director        |286           |25761           |
     |Human Resources          |3660          |23648           |
     +-------------------------+--------------+----------------+

     +------+-------------+-----------------+
     |Gender|MaritalStatus|   avg(DailyRate)|
     +------+-------------+-----------------+
     |  Male|       Single|804.8666666666667|
     |  Male|     Divorced|            766.5|
     |Female|       Single|743.4666666666667|
     |Female|     Divorced|821.3913043478261|
     |  Male|      Married|792.6388888888889|
     |Female|      Married|911.8135593220339|
     +------+-------------+-----------------+

     +----------------------+------+--------------+----------------+
     |Department            |Gender|max(DailyRate)|min(MonthlyRate)|
     +----------------------+------+--------------+----------------+
     |Research & Development|Female|1490          |3854            |
     |Human Resources       |Female|1420          |5220            |
     |Sales                 |Male  |1099          |7770            |
     |Research & Development|Male  |1315          |4933            |
     |Human Resources       |Male  |1246          |7999            |
     |Sales                 |Female|1402          |5404            |
     +----------------------+------+--------------+----------------+
```

```
deptdata=session.read.csv("DepartmentDetails.csv", header=True, inferSchema=True)
deptdata.show()
```

```
+---------------+-------+
|     Department|Manager|
+---------------+-------+
|        Finance|  David|
|          Sales|  Peter|
|Human Resources| George|
+---------------+-------+
```

```
#Joining in SPARK SQL
#Inner Join
hrdata.join(deptdata, hrdata["Department"].startswith(deptdata["Department"]),"inner").groupBy("Manager").count().show()
#Left Outer Join
hrdata.join(deptdata, hrdata["Department"].startswith(deptdata["Department"]),"left_outer").groupBy("Manager").count().show()
#Right Outer Join
hrdata.join(deptdata, hrdata["Department"].startswith(deptdata["Department"]),"right_outer").groupBy("Manager").count().show()
#Full Outer Join
hrdata.join(deptdata, hrdata["Department"].startswith(deptdata["Department"]),"full_outer").groupBy("Manager").count().show()
```

```
+-------+-----+
|Manager|count|
+-------+-----+
| George|   63|
|  Peter|  446|
+-------+-----+

+-------+-----+
|Manager|count|
+-------+-----+
|   NULL|  961|
| George|   63|
|  Peter|  446|
+-------+-----+

+-------+-----+
|Manager|count|
+-------+-----+
| George|   63|
|  Peter|  446|
|  David|    1|
+-------+-----+

+-------+-----+
|Manager|count|
+-------+-----+
|   NULL|  961|
| George|   63|
|  Peter|  446|
|  David|    1|
+-------+-----+
```