databricks **Pyspark_Notebook-4**

(https://databricks.com)
Split

2: Create Sample DataFrame

```
emp_data = [(10, "Michel Robinson", "1999-06-01", "100", 2000),
            (20, "James wood", "2003-03-01", "200", 8000),
            (30, "Chrish andreew", "2005-04-01", "100", 6000),
            (40, "Mark Bond", "2008-10-01", "100", 7000),
            (50, "Steve Watson", "1996-02-01", "400", 1000),
            (60, "Methew Simons", "1998-06-01", "500", 5000),
            (70, "Peter Paul", "2001-06-01", "600", 5000)]


emp_schema = ["employee_Id", "name", "doj", "employee_dept_id", "salary"]


emp_df = spark.createDataFrame(data = emp_data, schema = emp_schema)

emp_df.display()
```

▸ ▦ emp_df: pyspark.sql.dataframe.DataFrame = [employee_Id: long, name: string ... 3 more fields]

Table

	¹²₃ employee_Id	ᴬᴮ_C name	ᴬᴮ_C doj	ᴬᴮ_C employee_dept_id	¹²₃ salary
1	10	Michel Robinson	1999-06-01	100	2000
2	20	James wood	2003-03-01	200	8000
3	30	Chrish andreew	2005-04-01	100	6000
4	40	Mark Bond	2008-10-01	100	7000
5	50	Steve Watson	1996-02-01	400	1000
6	60	Methew Simons	1998-06-01	500	5000
7	70	Peter Paul	2001-06-01	600	5000

7 rows

3: Split the name to First and Last name

```
from pyspark.sql.functions import split

df_split = emp_df.withColumn('First Name',split(emp_df['name'], ' ').getItem(0))\
                 .withColumn('Last Name', split(emp_df['name'], ' ').getItem(1))

df_split.display()
```

▸ ▦ df_split: pyspark.sql.dataframe.DataFrame = [employee_Id: long, name: string ... 5 more fields]

Table

	¹²₃ employee_Id	ᴬᴮ_C name	ᴬᴮ_C doj	ᴬᴮ_C employee_dept_id	¹²₃ salary	ᴬᴮ_C First Name	ᴬᴮ_C L...
1	10	Michel Robinson	1999-06-01	100	2000	Michel	Robi...
2	20	James wood	2003-03-01	200	8000	James	woo...
3	30	Chrish andreew	2005-04-01	100	6000	Chrish	andr...
4	40	Mark Bond	2008-10-01	100	7000	Mark	Bon...
5	50	Steve Watson	1996-02-01	400	1000	Steve	Wat...
6	60	Methew Simons	1998-06-01	500	5000	Methew	Sim...

7 rows

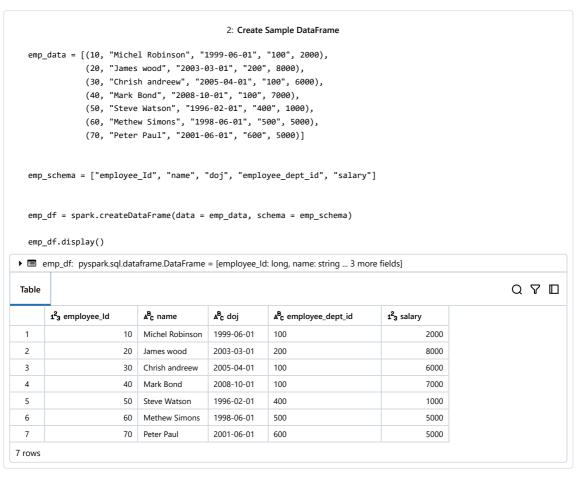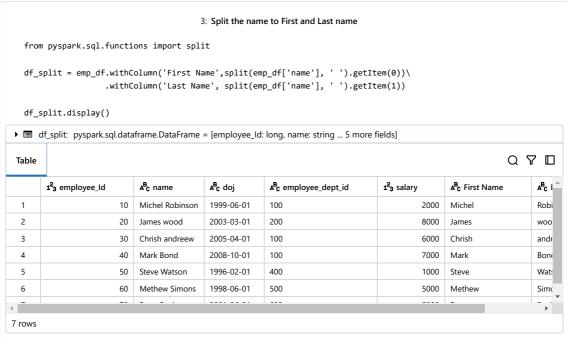### 4: Split & Drop

```
df2 = emp_df.withColumn('First Name',split(emp_df['name'], ' ').getItem(0))\
            .withColumn('Last Name', split(emp_df['name'], ' ').getItem(1))\
            .withColumn('Joining_Year', split(emp_df['doj'], '-').getItem(0))\
            .withColumn('Joining_Month', split(emp_df['doj'], '-').getItem(1))\
            .withColumn('Joining_date', split(emp_df['doj'], '-').getItem(2))\
            .drop(emp_df['name'])\
            .drop(emp_df['doj'])

df2.display()
```

▶ ▦ df2: pyspark.sql.dataframe.DataFrame = [employee_Id: long, employee_dept_id: string … 6 more fields]

Table   🔍 ▼ ▢

	1²₃ employee_Id	ᴬᴮ_C employee_dept_id	1²₃ salary	ᴬᴮ_C First Name	ᴬᴮ_C Last Name	ᴬᴮ_C Joining_Year
1	10	100	2000	Michel	Robinson	1999
2	20	200	8000	James	wood	2003
3	30	100	6000	Chrish	andreew	2005
4	40	100	7000	Mark	Bond	2008
5	50	400	1000	Steve	Watson	1996
6	60	500	5000	Methew	Simons	1998

7 rows

5: When & otherwise

```
data_student = [("Raja", "Science", 80,"p", 90),
                ("Rakesh", "Math", 90,"p", 70),
                ("Rama", "English", 20,"p", 80),
                ("Ramesh", "Science", 45,"F", 75),
                ("Rajesh", "Math", 30,"F", 50),
                ("Raghav", "Math", None,"NA", 70)]

schema = ["name", "subject", "Marks", "status", "Attendence"]


df = spark.createDataFrame(data = data_student, schema = schema)

df.display()
```

▶ ▦ df: pyspark.sql.dataframe.DataFrame = [name: string, subject: string … 3 more fields]

Table   🔍 ▼ ▢

	ᴬᴮ_C name	ᴬᴮ_C subject	1²₃ Marks	ᴬᴮ_C status	1²₃ Attendence
1	Raja	Science	80	p	90
2	Rakesh	Math	90	p	70
3	Rama	English	20	p	80
4	Ramesh	Science	45	F	75
5	Rajesh	Math	30	F	50
6	Raghav	Math	null	NA	70

6 rows

6

```
from pyspark.sql.functions import when

df1 = df.withColumn('status', when(df.Marks >=50, "pass")
                              .when(df.Marks<= 50, "fail")
                              .otherwise("absentee"))


df1.display()
```

▶ ▦ df1: pyspark.sql.dataframe.DataFrame = [name: string, subject: string … 3 more fields]

Table

	ᴬᴮc name	ᴬᴮc subject	₁²₃ Marks	ᴬᴮc status	₁²₃ Attendence
1	Raja	Science	80	pass	90
2	Rakesh	Math	90	pass	70
3	Rama	English	20	fail	80
4	Ramesh	Science	45	fail	75
5	Rajesh	Math	30	fail	50
6	Raghav	Math	null	absentee	70

6 rows

7: **Using And operator**

▸ ▦ df2: pyspark.sql.dataframe.DataFrame = [name: string, subject: string … 4 more fields]

Table

	ᴬᴮc name	ᴬᴮc subject	₁²₃ Marks	ᴬᴮc status	₁²₃ Attendence	ᴬᴮc Grade
1	Raja	Science	80	p	90	distinction
2	Rakesh	Math	90	p	70	Average
3	Rama	English	20	p	80	Average
4	Ramesh	Science	45	F	75	Average
5	Rajesh	Math	30	F	50	Good
6	Raghav	Math	null	NA	70	Average

6 rows