```
# Databricks notebook source
from pyspark.sql import SparkSession
spark =
SparkSession.builder.appName('SparkByExamples.com').getOrCreate()

from pyspark.sql.functions import expr
#Concatenate columns
data=[("James","Bond"),("Scott","Varsa")]
df=spark.createDataFrame(data).toDF("col1","col2")
df.withColumn("Name",expr(" col1 ||','|| col2")).show()

#Using CASE WHEN sql expression
data = [("James","M"),("Michael","F"),("Jen","")]
columns = ["name","gender"]
df = spark.createDataFrame(data = data, schema = columns)
df2 = df.withColumn("gender", expr("CASE WHEN gender = 'M' THEN 'Male'
" +
           "WHEN gender = 'F' THEN 'Female' ELSE 'unknown' END"))
df2.show()

#Add months from a value of another column
data=[("2019-01-23",1),("2019-06-24",2),("2019-09-20",3)]
df=spark.createDataFrame(data).toDF("date","increment")
df.select(df.date,df.increment,
     expr("add_months(date,increment)")
  .alias("inc_date")).show()

# Providing alias using 'as'
df.select(df.date,df.increment,
     expr("""add_months(date,increment) as inc_date""")
  ).show()

# Add
df.select(df.date,df.increment,
     expr("increment + 5 as new_increment")
  ).show()

df.select("increment",expr("cast(increment as string) as
str_increment")) \
  .printSchema()
#Use expr()  to filter the rows
data=[(100,2),(200,3000),(500,500)]
df=spark.createDataFrame(data).toDF("col1","col2")
df.filter(expr("col1 == col2")).show()

# COMMAND ----------
```