Actividad Módulo 48: Big Data Parte 2

Generar un archivo PDF que contenga las siguientes salidas:

- Exportación de archivos desde Spark
- Conexión desde Python al servicio Spark donde se instaló la información
- Ejecución de comandos Python que traigan la misma salida de información obtenida en el ejercicio anterior

In [ ]:
```
# Conexión desde Python al servicio Spark donde se instaló la información

# Generar sesión de spark
from pyspark.sql import SparkSession

spark = SparkSession\
            .builder\
            .appName('ActModulo48')\
            .getOrCreate()

spark
```

Out[ ]:

SparkSession - in-memory

SparkContext

Spark UI

Version	v3.5.1
Master	local[*]
AppName	ActModulo48

In [ ]:
```
# Cargar datos de housing
path_archivo = "D:/DataAnalysis_EBAC/ebac/Python/Modulo44/kc_house_data.csv"
df = spark.read.csv(path_archivo, header=True, inferSchema=True)
df.printSchema()
```

```
root
 |-- id: long (nullable = true)
 |-- date: string (nullable = true)
 |-- price: double (nullable = true)
 |-- bedrooms: integer (nullable = true)
 |-- bathrooms: double (nullable = true)
 |-- sqft_living: integer (nullable = true)
 |-- sqft_lot: integer (nullable = true)
 |-- floors: double (nullable = true)
 |-- waterfront: integer (nullable = true)
 |-- view: integer (nullable = true)
 |-- condition: integer (nullable = true)
 |-- grade: integer (nullable = true)
 |-- sqft_above: integer (nullable = true)
 |-- sqft_basement: integer (nullable = true)
 |-- yr_built: integer (nullable = true)
 |-- yr_renovated: integer (nullable = true)
 |-- zipcode: integer (nullable = true)
 |-- lat: double (nullable = true)
 |-- long: double (nullable = true)
 |-- sqft_living15: integer (nullable = true)
 |-- sqft_lot15: integer (nullable = true)
```

In [ ]:
```
df.select('zipcode').distinct().count()
```

Out[ ]:
```
70
```

In [ ]:
```
# Exportación de datos desde Spark - Hacer particiones (Por Genre)
df.write.option('header', True).partitionBy('zipcode').mode('overwrite').csv('./par
```

In [ ]:
```
# Ejecución de comandos Python que traigan la misma salida de información obtenida
# Ejecución de comandos SQL con la misma salida de información del ejercicio del Mó

# Para el zipcode con mayor número de casas, calcular el promedio de precio y tamañ
df.createOrReplaceTempView('KC_HOUSING')

sql_str = """
            select  zipcode,
                    count(distinct id) as id_count,
                    avg(price) as avg_price,
                    avg(sqft_living) * 0.0929 as sqft_living_m2
            from KC_HOUSING
            group by zipcode
            order by count(distinct id) desc
            limit 3
"""
spark.sql(sql_str).show(10)
```

```
+-------+--------+-----------------+------------------+
|zipcode|id_count|        avg_price|    sqft_living_m2|
+-------+--------+-----------------+------------------+
|  98103|     600|584919.2109634551|153.36215946843853|
|  98038|     587|         366867.6|199.52274711864408|
|  98115|     576|619900.5471698113|  170.498907890223|
+-------+--------+-----------------+------------------+
```

In [ ]:
```
# Agrupamiento en Spark, por zipcode, por número de habitaciones y baños, precio pr

df.createOrReplaceTempView('KC_HOUSING')

sql_str = """
            select  zipcode,
                    bedrooms,
                    bathrooms,
                    avg(price) as avg_price
            from KC_HOUSING
            group by zipcode, bedrooms, bathrooms
            order by 1,2,3
"""

spark.sql(sql_str).show(20)
```

```
+-------+--------+---------+------------------+
|zipcode|bedrooms|bathrooms|         avg_price|
+-------+--------+---------+------------------+
|  98001|       0|      0.0|          139950.0|
|  98001|       1|      1.0|          166000.0|
|  98001|       1|      2.0|          171000.0|
|  98001|       2|      1.0|197428.57142857142|
|  98001|       2|      1.5|          350000.0|
|  98001|       2|     1.75|          246112.5|
|  98001|       2|      2.5|          214100.0|
|  98001|       2|     2.75|          239475.0|
|  98001|       3|     0.75|          363000.0|
|  98001|       3|      1.0|205182.80952380953|
|  98001|       3|      1.5|          224108.5|
|  98001|       3|     1.75| 260531.0810810811|
|  98001|       3|      2.0|256841.42857142858|
|  98001|       3|     2.25|          265999.0|
|  98001|       3|      2.5| 308581.8604651163|
|  98001|       3|     2.75|          255000.0|
|  98001|       3|      3.0|          309500.0|
|  98001|       4|      1.0|          229790.0|
|  98001|       4|      1.5|246406.85714285713|
|  98001|       4|     1.75| 251114.2857142857|
+-------+--------+---------+------------------+
only showing top 20 rows
```