Importing modules

In [36]:

```
from pyspark.sql import SparkSession
```

Creating spark session

In [37]:

```
spark = SparkSession.builder.appName("Erick")\
        .config('spark.jars.packages', 'mysql:mysql-connector-java:8.0.32')\
        .getOrCreate()
sqlContext = SparkSession(spark)
spark.sparkContext.setLogLevel("ERROR")
```

Creating connection to mysql

In [78]:

```
sql_df = spark.read \
    .format("jdbc") \
    .option("driver","com.mysql.cj.jdbc.Driver") \
    .option("url", "jdbc:mysql://192.168.0.101:3306/erick") \
    .option("dbtable", "BreastCancer") \
    .option("user", "root") \
    .option("password", "mysql") \
    .load()
```

Showing the data types of columns in the dataset

In [79]:

```
sql_df.printSchema()
```

```
root
 |-- id: integer (nullable = true)
 |-- diagnosis: string (nullable = true)
 |-- radius_mean: double (nullable = true)
 |-- texture_mean: double (nullable = true)
 |-- perimeter_mean: double (nullable = true)
 |-- area_mean: double (nullable = true)
 |-- smoothness_mean: double (nullable = true)
 |-- compactness_mean: double (nullable = true)
 |-- concavity_mean: double (nullable = true)
 |-- concave points_mean: double (nullable = true)
 |-- symmetry_mean: double (nullable = true)
 |-- fractal_dimension_mean: double (nullable = true)
 |-- radius_se: double (nullable = true)
 |-- texture_se: double (nullable = true)
 |-- perimeter_se: double (nullable = true)
 |-- area_se: double (nullable = true)
 |-- smoothness_se: double (nullable = true)
 |-- compactness_se: double (nullable = true)
 |-- concavity_se: double (nullable = true)
 |-- concave points_se: double (nullable = true)
 |-- symmetry_se: double (nullable = true)
 |-- fractal_dimension_se: double (nullable = true)
 |-- radius_worst: double (nullable = true)
 |-- texture_worst: double (nullable = true)
 |-- perimeter_worst: double (nullable = true)
 |-- area_worst: double (nullable = true)
 |-- smoothness_worst: double (nullable = true)
 |-- compactness_worst: double (nullable = true)
 |-- concavity_worst: double (nullable = true)
 |-- concave points_worst: double (nullable = true)
 |-- symmetry_worst: double (nullable = true)
 |-- fractal_dimension_worst: double (nullable = true)
```

Count the number of rows in the dataset

In [80]:

```
sql_df.count()
```

Out[80]: 569

show columns names present in the dataset

In [81]:

```
print(sql_df.columns)
```

```
['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity
_mean', 'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean', 'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'sm
oothness_se', 'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se', 'fractal_dimension_se', 'radius_worst', 'textur
e_worst', 'perimeter_worst', 'area_worst', 'smoothness_worst', 'compactness_worst', 'concavity_worst', 'concave points_worst', 'sym
metry_worst', 'fractal_dimension_worst']
```

Since there so many columns i have decided to display the columns in pandas for clearity purposes

In [115…

```
import pandas as pd
pd.DataFrame(sql_df.take(10), columns=sql_df.columns).head(100)
```

Out[115]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concav points_mea
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.1471
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.0701
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.1279
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.1052
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.1043
5	843786	M	12.45	15.70	82.57	477.1	0.12780	0.17000	0.15780	0.0808
6	844359	M	18.25	19.98	119.60	1040.0	0.09463	0.10900	0.11270	0.0740
7	84458202	M	13.71	20.83	90.20	577.9	0.11890	0.16450	0.09366	0.0598
8	844981	M	13.00	21.82	87.50	519.8	0.12730	0.19320	0.18590	0.0935
9	84501001	M	12.46	24.04	83.97	475.9	0.11860	0.23960	0.22730	0.0854

10 rows × 32 columns

Data cleaning, checking for null values

In [117…

```
from pyspark.sql.functions import isnan, when, count, col
```

In [118…

```
sql_df.filter(sql_df['radius_mean'].isNull()).count()
```

Out[118]: 0

Using sql to write query from the dataset

In [119…

```
sql_df.createOrReplaceTempView("sql_df")
```

In [120…

```
spark.sql('select area_worst from sql_df').show(5)
```

```
+----------+
|area_worst|
+----------+
|    2019.0|
|    1956.0|
|    1709.0|
|     567.7|
|    1575.0|
+----------+
only showing top 5 rows
```

In [121…

```
spark.sql('select count(diagnosis) from sql_df').show(5)
```

```
+----------------+
|count(diagnosis)|
+----------------+
|             569|
+----------------+
```

In [122…

```
spark.sql('select diagnosis,\
    perimeter_mean,\
    perimeter_worst from sql_df where fractal_dimension_worst>0.07678').show()
```

```
+---------+--------------+---------------+
|diagnosis|perimeter_mean|perimeter_worst|
+---------+--------------+---------------+
|        M|         122.8|          184.6|
|        M|         132.9|          158.8|
|        M|         130.0|          152.5|
|        M|         77.58|          98.87|
|        M|         82.57|          103.4|
|        M|         119.6|          153.2|
|        M|          90.2|          110.6|
|        M|          87.5|          106.2|
|        M|         83.97|          97.65|
|        M|         102.7|          123.8|
|        M|         103.6|          136.5|
|        M|         132.4|          151.7|
|        M|          93.6|          108.8|
|        M|         96.73|          124.1|
|        M|         94.74|          123.4|
|        M|         108.1|          136.8|
|        B|         85.63|          96.09|
|        B|         60.34|          65.13|
|        M|         102.5|          125.1|
|        M|         110.0|          177.0|
+---------+--------------+---------------+
only showing top 20 rows
```

In [ ]:

In [ ]: