dbutils.fs.ls('FileStore/tables/')

from pyspark.sql import SparkSession

from pyspark.sql.types import \*

from pyspark.sql.functions import \*

from pyspark.sql.window import Window

# read.format()

# schema

# header

# load

spark = SparkSession.builder.appName('Basics').getOrCreate()

df = spark.read.format('csv')\ # read.format

.option('inferSchema', True)\ # schema

.option('header', True)\ # header

.load('/FileStore/tables/BigMart\_Sales.csv') # load

df.select('Item\_Identifier', 'Item\_Weight', 'Item\_Fat\_Content')

df.select(col('Item\_Identifier'), col('Item\_Weight'), col('Item\_Fat\_Content'))

col('Item\_Identifier').alias('Item\_ID')

df.filter(col('Item\_Fat\_Content') == 'Regular')

df.filter( (col('Item\_Type') == 'Soft Drinks') & (col('Item\_Weight') < 10) )

df.filter(

    col('Outlet\_Location\_Type').isin('Tier 1', 'Tier 2') &

    col('Outlet\_Size').isNull()

)

df.withColumnRenamed('Item\_Weight', 'Item Kg')

df.withColumn(

    'multiply',

    col('Item\_Weight') \* col('Item\_MRP')

)

df.withColumn(

    'Item\_Fat\_Content2',

    regexp\_replace( col('Item\_Fat\_Content'), 'Regular', 'Standart' ))

df.sort(

    col('Item\_Weight').desc())

df.sort(

    ['Item\_Weight', 'Item\_Visibility'],

    ascending = [False, True] )

df.limit(10)

df.drop('Item\_Weight', 'Item\_Visibility', 'Item\_Fat\_Content')

df.dropDuplicates(['Item\_Type'])

df1.union(df2)

df1.unionByName(df2)

df.select(upper('Item\_Type'))

df.dropna(how='any')

df.fillna('NotAvailable')

df.groupBy('Item\_Type').mean()

df.groupBy('Item\_Type').agg(mean('Item\_MRP'))

df.groupBy('Item\_Type').pivot('Outlet\_Size').mean()

df.groupBy('Item\_Type').pivot('Outlet\_Size').agg(mean('Item\_MRP'))

df.withColumn(

    'veg\_flag',

    when(col('Item\_Type')=='Meat', 'Non\_Veg').otherwise('Veg')

)

df.withColumn(

    'veg\_flag',

    when(

        (col('Item\_Type') != 'Meat') & (col('Item\_MRP') < 100),

        'Veg\_Inexpensive'

    ).when(

        (col('Item\_Type') != 'Meat') & (col('Item\_MRP') >= 100),

        'Veg\_Expensive'

    ).otherwise(

        'Non\_Veg'

    ))

df1.join(

    df2,

    df1.dept\_id == df2.dept\_id,

    'inner')

# row\_number: 1 2 3 4 5 6 7 8 9

df.withColumn(

    'rowCol',

    row\_number().over(Window.orderBy('Item\_Identifier')))

# Rank: 1 1 1 1 5 5 5 8 8

df.withColumn(

    'rank',

    rank().over(Window.orderBy('Item\_Identifier')))

# Dense Rank 1 1 1 1 2 2 2 3 3

df.withColumn(

    'denseRank',

    dense\_rank().over(Window.orderBy(col('Item\_Identifier').desc())))

def my\_func(x):

    return x\*x

my\_udf = udf(my\_func)

df.withColumn(

    'mynewcol',

    my\_udf('Item\_MRP'))

df.write.format('csv').save('/FileStore/tables/CSV/data.csv')

df.createTempView('my\_view')

spark.sql("select \* from my\_view where Item\_Fat\_Content = 'Low Fat' ")