

### Defining Schema

```
from pyspark.sql.types import *
Schema = StructType([
    StructField('Store', StringType(), nullable=True),
    StructField('Store Type', StringType(), nullable=True),
    StructField('Assessment', StringType(), nullable=True),
    StructField('Competition Distance', FloatType(), nullable=True),
    StructField('Competition Open Since Month', IntegerType(), nullable=True),
    StructField('Competition Open Since Year', IntegerType(), nullable=True),
    StructField('Promo 2', IntegerType(), nullable=True),
    StructField('Promo 2 Since Week', IntegerType(), nullable=True),
    StructField('Promo 2 Since Year', IntegerType(), nullable=True),
    StructField('Promo Interval', StringType(), nullable=True)
])
df = spark.read.option("header", True).schema(Schema).csv('store.csv')
# We can drop invalid rows while reading the dataset by setting the read mode as "DROPMALFORMED"
df_l = spark.read.option("header", True).option("mode", 'DROPMALFORMED').csv('store.csv')
df.show()
```

Spark does not detect schema itself properly, so we need to define the schema as well for the data set.

### PySpark DataTypes

| Type        | Size (Byte) | Default | Range (Digits) |        |
|-------------|-------------|---------|----------------|--------|
| byte        | 1           | 0       | 3              | Ints   |
| short       | 2           | 0       | 5              |        |
| int         | 4           | 0       | 10             |        |
| long        | 8           | 0       | Lots           |        |
| floats      | 4           | 0.0f    | Lots           | floats |
| double      | 8           | 0.0d    | Lots           |        |
| DecimalType | 32          | 0.0     | Lots           |        |

### Filtering Data

### String Data Types

#### StringType

**VarcharType(-length)** A variant of StringType which has a length limitation. Data writing will fail if the input string exceeds the length limitation

**CharType(-length)** Reading column of type CharType(n) always returns string values of length n. Char type column comparison will pad the short one to the longer length.

### Adding, renaming and removing columns

### Complex Data Types

**ArrayType(elementType, containsNull)** Represents values comprising a sequence of elements

**MapType(keyType, valueType, valueContainsNull)** Represents values comprising a set of key-value pairs. The data type of keys is described by keyType and the data type of values is described by valueType. For a MapType value, keys are not allowed to have null values. valueContainsNull is used to indicate if values of a MapType value can have null values.

**StructType(fields)** Represents values with the structure described by a sequence of StructFields (fields)

If, elif, else equivalent

```

voter_df.filter(voter_df['name'].isNone)

OR

voter_df.withColumn(
    voter_df.w here(~ voter_df._ cl.is None)) voter_df.d at e.year)
voter_df.f il ter (vo ter _df.da te.year>=1800)
test_d f_sex = test_d f.w ith -
voter_df.w he re( vot er_ df[ '_c 0' Co l o u m n R e n a m e d(' G e n d e r',
n t a i n s ('V O T E'))
'Sex')
#Multiple Conditions
drop
whereDF = flatte nDF.wh ere ((c ol( "voter_ df.d ro p(' unu sed _co -
stN ame ") == " xia ngr ui") | (col("f u n n s ) -
tN a m e") == " mic hae l")) .so rt( asc f o r m a p s p a r k . s q l i m p o r t
stN ame "))
functions as F
whereD F.s how (tr unc ate =False) add_n = udf(lambda x, y: x + y,
#Unique Values Integer T y p e())
voter_df = df.sel ect (df ["VOTER # We register a UDF that adds a
NAME"]).di sti nct() column to the DataFrame,
# Show the rows with 10 highest IDs and we cast the id column to an
set Integer type.
voter_ df.o rd erB y(v o t e r_ d f.R O W I D . d f . w i t h C o l u m n (' i d _
es c() ).s how(10) off set', add_n( F.l it( 1000),
df.id.c as t(I n t e g e r T y p -
e()))

```

### User Defined Functions

1. Define a Python method

```
def revers eSt rin g(m ystr):
return mystr[::-1]
```
2. Wrap the function and store as a variable

```
udfRev ers eString = udf(re -
ver seS tring, String Type())
```
3. Use with Spark

```
user_df = user_d f.w ith Col -
umn ('R eve rse Name',
udfRev ers eSt rin g(u ser -
_df.Name))
```

### Using SQL to clean script

```

df.createOrReplaceTempView("table1")

df2 = spark.sql("SELECT field1, field2
FROM table1")

```

```

.when(<if condition>, <then x>)
df.sel ect (df.Name, df.Age,
.when( df.Age >= 18, " Adu lt")
.when( df.Age < 18, " Min or"))
.other wise() is like else
df.sel ect (df.Name, df.Age,
.when( df.Age >= 18, " Adu lt")
.other wis e("M ino r")

```

### Remove duplicate rows & replace values

```

dropDuplicates()
test_d f_n o_dup =
tes t_d f.s ele ct( 'Us er_ -
ID' , 'G ender', 'Age').dr opD -
upl ica tes()
fillna()
used to replace null value with
any other value
df.fil lna (va lue ==-9 9,s -
ubset=

```

### Validating with Joins

```

parsed_df =
spark.read.parquet('parsed_data.parquet')
company_df = spark.r ea d.p ar q uet ('c -
omp ani es.p ar quet')
verifi ed_df = parsed _df.jo in( com -
pan y_df, parsed _df.co mpany ==
compan y_d f.c ompany)
# This automa tically removes any rows
with a company not in the valid_df !

```

### View data/actions:

```

printSchema(), head(), show(), count(),
columns and describe()

```

show() - Displays/Prints a number of rows in a tabular format. By default it displays 20 rows and to change the default number, you can pass a value to show(n).

where as take(n) returns first n rows as Array of row objects. It is an alias for first().

count() - total rows



By **datamansam**

[cheatography.com/datamansam/](http://cheatography.com/datamansam/)

Published 3rd September, 2022.

Last updated 12th September, 2022.

Page 2 of 3.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

### Remove duplicate rows & replace values (cont)

```
> ["Promo2SinceWeek", "Promo2SinceYear").show()
.withColumn(), when()
creating a new column, with value equal to
1 if
Promo2SinceYear > 2000 otherwise 0
df.withColumn("greater_than_2000",
when(df.CompetitionDistance==2000, 1).otherwise(0)
.alias('value_desc')).show()
```



By **datamansam**

[cheatography.com/datamansam/](https://cheatography.com/datamansam/)

Published 3rd September, 2022.

Last updated 12th September, 2022.

Page 3 of 3.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>