Cleaning with PySpark Cheat Sheet

Defining Schema

```
from pyspark.sql.types import *
Schema =St ruc tType([
    Str uct Fie ld( 'St ore ',S tri ngT ype (), nul lab le= True),
    Str uct Fie ld( 'St ore Typ e', Str ing Typ e() ,nu lla ble =True),
    Str uct Fie ld( 'As sor tme nt' ,St rin gTy pe( ),n ull abl e=T rue),
    Str uct Fie ld( 'Co mpe tit ion Dis tan ce' ,Fl oat Typ e() ,nu lla ble =True),
    Str uct Fie ld( 'Co mpe tit ion Ope nSi nce Mon th' ,In teg erT ype (), nul lab le= T
  rue), Struct Fie -
ld( 'Co mpe tit ion Ope nSi nce Yea r', Int ege rTy pe( ),n ull abl e=T rue),
Struct Fie ld( 'Pr omo 2', - Int ege rTy pe( ),n ull abl e=T rue),
    Str uct Fie ld( 'Pr omo 2Si nce Wee k', Int ege rTy pe( ),n ull abl e=T rue),
    Str uct Fie ld( 'Pr omo 2Si nce Yea r', Int ege rTy pe( ),n ull abl e=T rue),
    Str uct Fie ld( 'Pr omo Int erv al' ,St rin gTy pe( ),n ull abl e=True)
])
df = spark.r ea d.o pti on( "he ade r", Tru e).s ch ema (Sc hem a).c sv ('s tor e.csv')
# We can drop invalid rows while reading the dataset by setting the read mode as "DROPM ALF ORMED"
df_1=s par k.r ead.op tio n(" hea der ",T rue ).o pti on( "mo de" ,'D ROP MAL FOR MED ').c
  sv ('s tor -
e.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	
Decima - lType	32	0.0	Lots	

String Data Types

StringType	
Varcha rTy pe(- length)	A variant of StringType which has a length limita tion. Data writing will fail if the input string exceeds the length limitation
CharT y pe(- length)	Reading column of type CharTy - pe(n) always returns string values of length n. Char type column comparison will pad the short one to the longer length.

Complex Data Types

ArrayT - ype (el - eme - ntType, contai - nsNull)	nts values comprising a sequence of elements
MapTyp - e(k eyTyp e, valueType, valueC on t - ain sNull)	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. valueC ont ain sNull is used to indicate if values of a MapType value can have null values.
Struct Typ - e(f ields)	Represents values with the structure described by a sequence of Struct Fields

Filtering Data

Adding, renaming and removing columns

(fields)

```
voter_df.filter(voter_df['name'].isNotNull())add-withColumn
   ORvoter_ df.w it hCo lum n(' -
voter_ df.w here(~ voter_ df._ c1.is Null())year',
voter_ df.d at e.year)
voter_ df.f il ter (vo ter _df.da te.yrearnaming> -
withCo lum nRe named
1800)                                test_d f_sex =
test_d f.w ith -
voter_ df.w he re( vot er_ df[ '_c 0']Col.co   umn- Ren ame
  d(' Gen der',
nta ins ('V OTE'))                   'Sex')
#Multiple Conditions                  drop
whereDF =
flatte nDF.wh ere ((c ol( " voterfir - df.d ro p(' unu sed
  _co -
stN ame ") == " xia ngr ui") |
(col("flumn') irs -
tNa me") ==
" mic hae l")).so rt( asc from("la
  pyspar-                                    k.sql import
stN ame "))                          functions as F
whereD F.s how (tr unc ate =Fa
lse)                                 add_n = udf(lambda x, y: x + y,
                                     Intege rTy pe())
#Unique Values
voter_df =                           # We register a UDF that adds a
df.sel ect (df ["VOTER               column to the DataFrame,
NAME"]).di sti nct()                 # Show the rows with 10 highest IDs inandthewe cast the id column to an
set                                  Integer type.
voter_ df.o rd erB y(v ote r_d       f.R OWdf ID=.
d f-.wit hCo lum n(' id_ -
                                     off set',
es c() ).s how(10)                   add_n( F.l it( 1000),
                                     df.id.c as t(I nte ger Typ
                                       -
User Defined Functions               e())))
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 se
r - _df.Name))
```

```
parsed_df =
spark.read.parquet('parsed_data.parquet')
company_df = spark.r ea d.p arq 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/a ctions:

printS che ma(), head(), show(), count(), columns and describe()

show() - Displa ys/ 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).

```
.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=
```

df.cre ate OrR epl ace Tem pVi e w(" tab le1 ")

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

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

count() - total rows

Published 3rd September, 2022.
Last updated 12th September, 2022.

Remove duplicate rows & replace values (cont)

```
> [“Prom o2S inc eWe ek” ,”P rom o2S inc eYe -
ar” ]).s how()
.withC olumn() ,when()
creating a new column, with value equal to
1 if
Promo2 Sin ceYear > 2000 otherwise 0
df.wit hCo lum n(“ gre_ ate r_t han 2000”,
when(d f.C omp eti tio nDi sta nce ==2 000 ,1).ot -
her wise(0)
.alias (‘v alu e_d esc ’)).show()
```

By datamansam

cheatography.com/datamansam/

Published 3rd September, 2022.
Last updated 12th September, 2022.