PYSPARK TIP

CLEAN EVERY COLUMN NAME IN ONE LINE

```
# Before
["CustID", "trnxAmt", "ACCT-bal", "pmt$Status"]

# After
["cust_id", "trnx_amt", "acct_bal", "pmt_status"]
```

THE MESS

Sign Up to the newsletter

Databricks.news

Find more databricks tips @

dailydatabricks.tips

EVERY SOURCE SYSTEM IS DIFFERENT

Vendor feeds, CSVs, APIs, legacy databases. Column names arrive in every format imaginable.

```
# Source system column names
df.columns = [
"CustID", # PascalCase
"custName", # camelCase
"trnx_date", # abbreviated
"ACCT-bal", # UPPER + hyphen
"pmt$Status", # dollar sign
"addr.Line1", # dot notation
]
```

ONE LINE FIX

Sign Up to the newsletter

Databricks.news

Find more databricks tips @

dailydatabricks.tips

DICT COMPREHENSION + .WITHCOLUMNSRENAMED()

Loop over df.columns, apply a regex, pass the dict. One plan node, every column renamed.

```python
import re

# Strip special chars in one dict comprehension
df_clean = df.withColumnsRenamed(
{
c: re.sub(r'[.$\-@#!]', '_', c).lower()
for c in df.columns
}
)
```

"ACCT-bal" becomes "acct_bal", "pmt$Status" becomes "pmt_status"

MAKE IT GENERIC

A REUSABLE CLEAN_COLUMNS() FUNCTION

Strips special characters, splits camelCase, lowercases everything. Works on any DataFrame.

```python
def clean_columns(df):
return df.withColumnsRenamed({
c: re.sub(r'[.$\-@#!]', '_',
re.sub(r'([a-z])([A-Z])',
r'\1_\2', c)).lower()
for c in df.columns
})

# Use it anywhere
df_clean = clean_columns(df)
```

EXPAND ABBREV -IATIONS

CHAIN IT WITH ABBREVIATION EXPANSION

"trnx_amt" becomes "transaction_amount", "acct_bal" becomes "account_balance"

```python
ABBR = {"trnx": "transaction",
"cust": "customer",
"acct": "account",
"pmt": "payment",
"amt": "amount",
"bal": "balance"}

def expand(name):
return "_".join(ABBR.get(p, p)
for p in name.split("_"))

# Chain: clean then expand
df.withColumnsRenamed(
{c: expand(clean(c)) for c in df.columns})
```

Sign Up to the newsletter

Databricks.news

Find more databricks tips @

dailydatabricks.tips

WANT MORE TIPS?

FOLLOW FOR DAILY DATABRICKS TIPS

SIGN UP TO THE NEWSLETTER

Databricks.news

dailydatabricks.tips