Mastering Data Refinement: Unleashing the Power of SQL in Python using Polars

Tom Levy

02/29/24

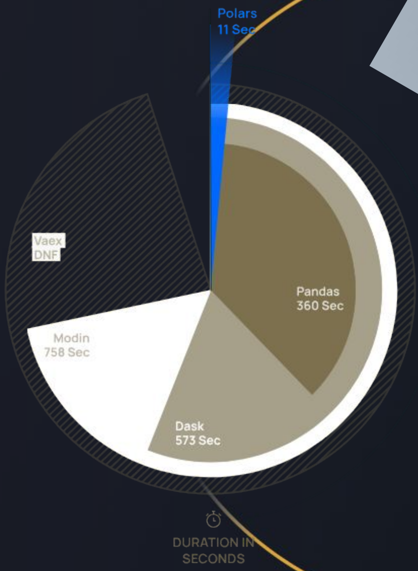
Own

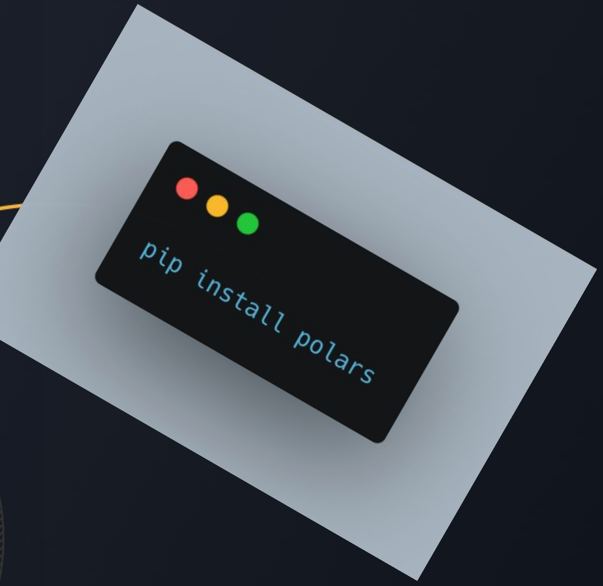
Agenda

- Why Polars?
- Introduction to SQL Context
- How to utilize LazyFrames for big datasets using SQL Context
- Common Table Expressions (CTEs)

Why Polars?

- High Performance - https://pola.rs/posts/benchmarks/
- Memory Efficiency
- I/O
- Lazy Evaluation
- Easy-to-use API
- SQL Context
- Versatile Data Manipulation
- Interoperability
- Open Source - https://github.com/pola-rs/polars/

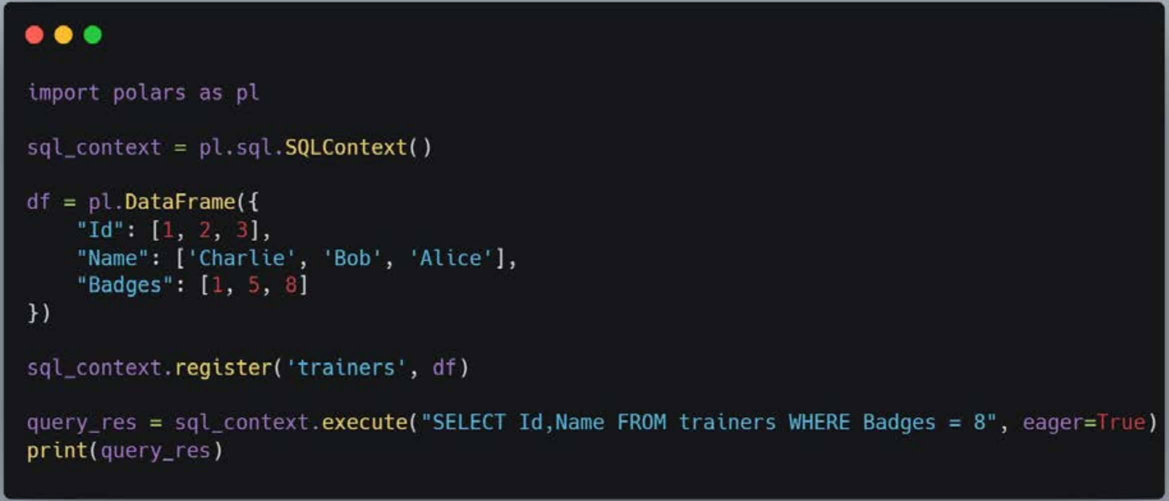




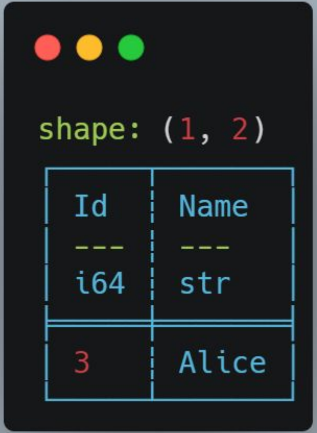
```
pip install polars
```

Official user guide - https://docs.pola.rs/

Introduction to SQL Context

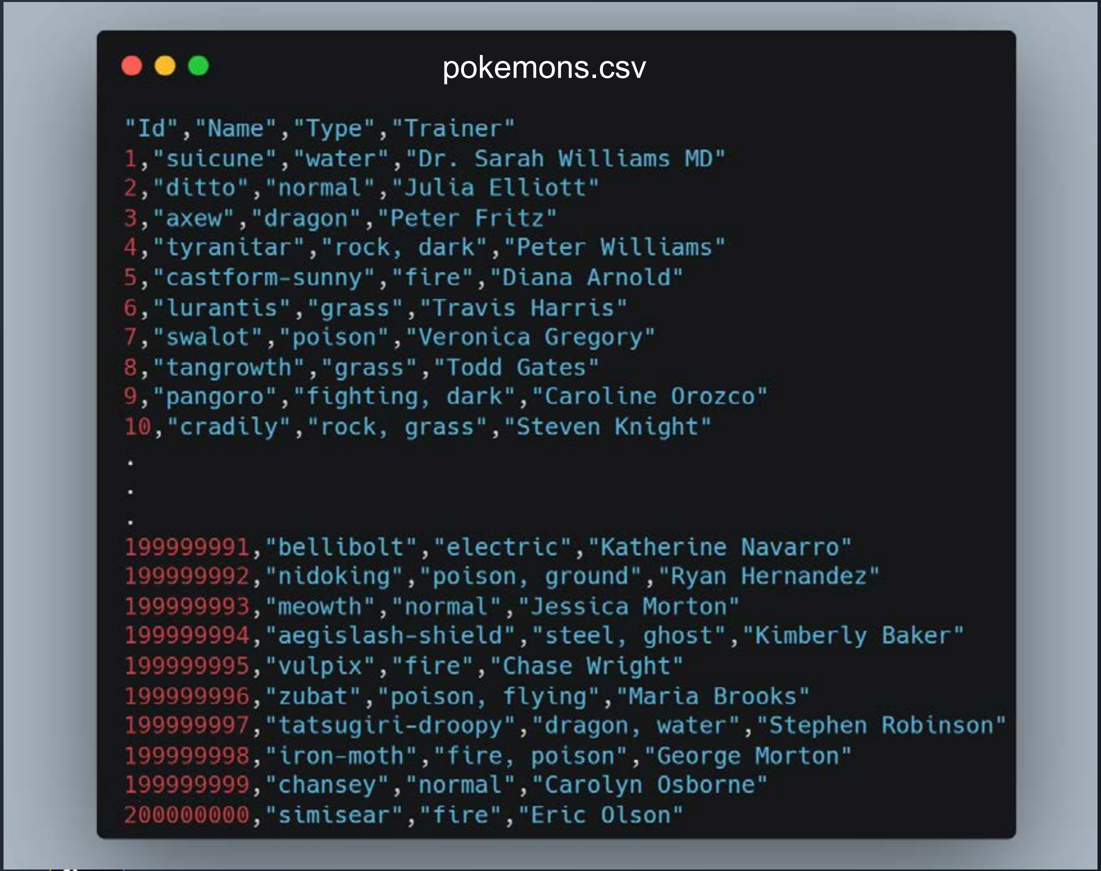


```python
import polars as pl

sql_context = pl.sql.SQLContext()

df = pl.DataFrame({
    "Id": [1, 2, 3],
    "Name": ['Charlie', 'Bob', 'Alice'],
    "Badges": [1, 5, 8]
})

sql_context.register('trainers', df)

query_res = sql_context.execute("SELECT Id,Name FROM trainers WHERE Badges = 8", eager=True)
print(query_res)
```



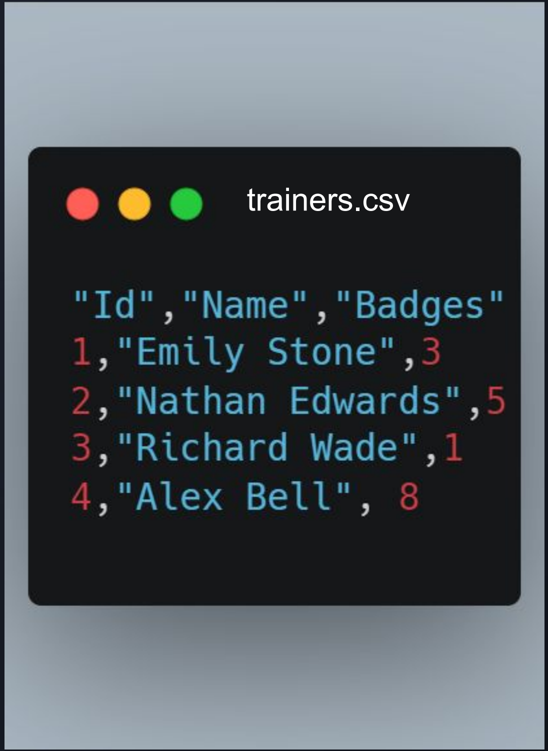
```
shape: (1, 2)
┌─────┬───────┐
│ Id  ┆ Name  │
│ --- ┆ ---   │
│ i64 ┆ str   │
╞═════╪═══════╡
│ 3   ┆ Alice │
└─────┴───────┘
```

Utilize LazyFrames for big datasets



pokemons.csv

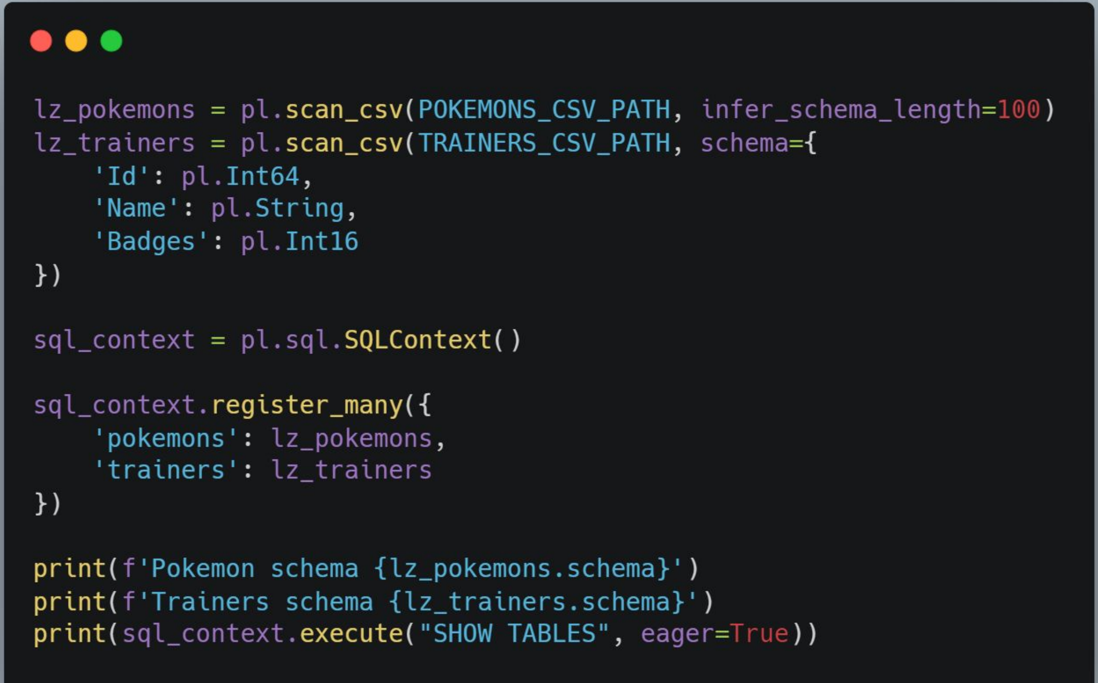
```
"Id","Name","Type","Trainer"
1,"suicune","water","Dr. Sarah Williams MD"
2,"ditto","normal","Julia Elliott"
3,"axew","dragon","Peter Fritz"
4,"tyranitar","rock, dark","Peter Williams"
5,"castform-sunny","fire","Diana Arnold"
6,"lurantis","grass","Travis Harris"
7,"swalot","poison","Veronica Gregory"
8,"tangrowth","grass","Todd Gates"
9,"pangoro","fighting, dark","Caroline Orozco"
10,"cradily","rock, grass","Steven Knight"
.
.
.
199999991,"bellibolt","electric","Katherine Navarro"
199999992,"nidoking","poison, ground","Ryan Hernandez"
199999993,"meowth","normal","Jessica Morton"
199999994,"aegislash-shield","steel, ghost","Kimberly Baker"
199999995,"vulpix","fire","Chase Wright"
199999996,"zubat","poison, flying","Maria Brooks"
199999997,"tatsugiri-droopy","dragon, water","Stephen Robinson"
199999998,"iron-moth","fire, poison","George Morton"
199999999,"chansey","normal","Carolyn Osborne"
200000000,"simisear","fire","Eric Olson"
```



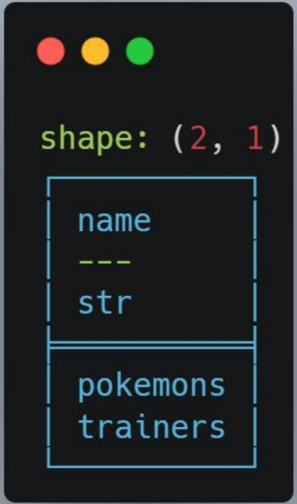
trainers.csv

```
"Id","Name","Badges"
1,"Emily Stone",3
2,"Nathan Edwards",5
3,"Richard Wade",1
4,"Alex Bell", 8
```

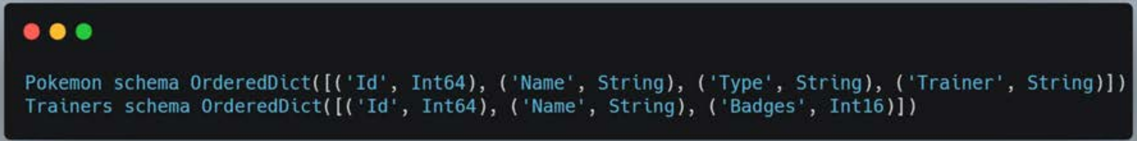
Utilize LazyFrames for big datasets



```python
lz_pokemons = pl.scan_csv(POKEMONS_CSV_PATH, infer_schema_length=100)
lz_trainers = pl.scan_csv(TRAINERS_CSV_PATH, schema={
    'Id': pl.Int64,
    'Name': pl.String,
    'Badges': pl.Int16
})

sql_context = pl.sql.SQLContext()

sql_context.register_many({
    'pokemons': lz_pokemons,
    'trainers': lz_trainers
})

print(f'Pokemon schema {lz_pokemons.schema}')
print(f'Trainers schema {lz_trainers.schema}')
print(sql_context.execute("SHOW TABLES", eager=True))
```

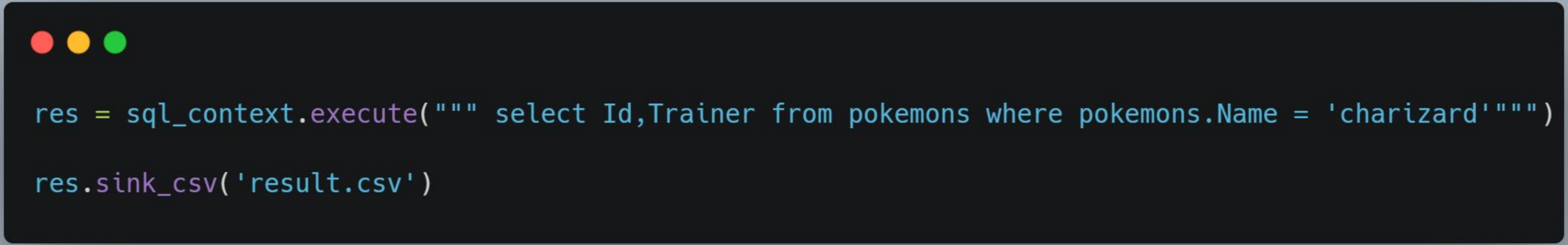


```
shape: (2, 1)
┌──────────┐
│ name     │
│ ---      │
│ str      │
╞══════════╡
│ pokemons │
│ trainers │
└──────────┘
```



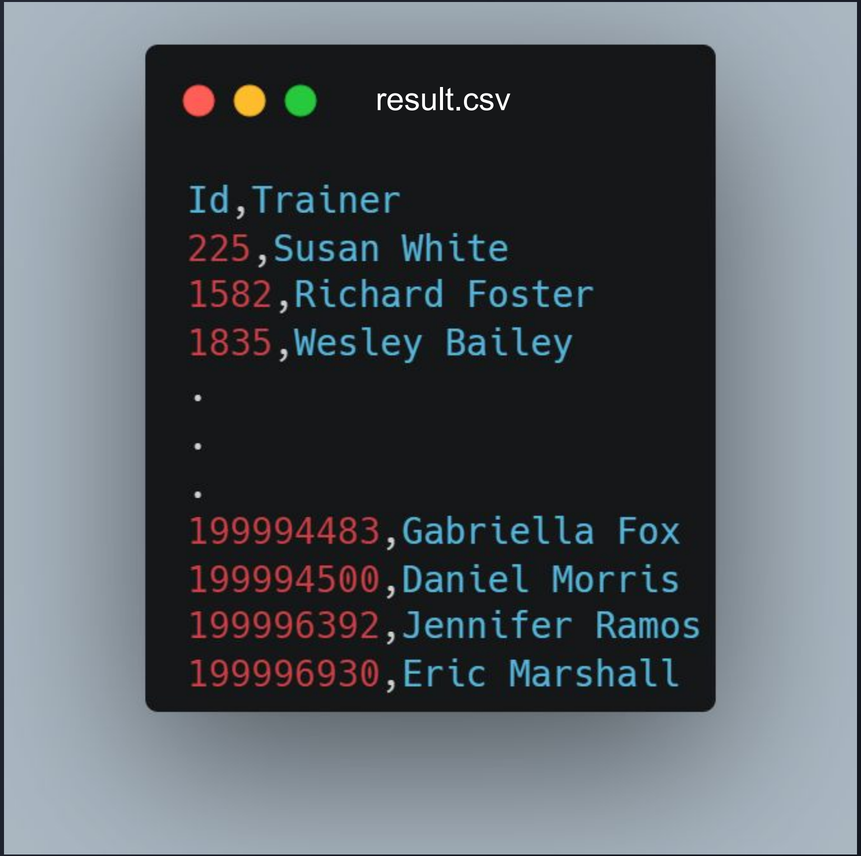
```
Pokemon schema OrderedDict([('Id', Int64), ('Name', String), ('Type', String), ('Trainer', String)])
Trainers schema OrderedDict([('Id', Int64), ('Name', String), ('Badges', Int16)])
```

Utilize LazyFrames for big datasets



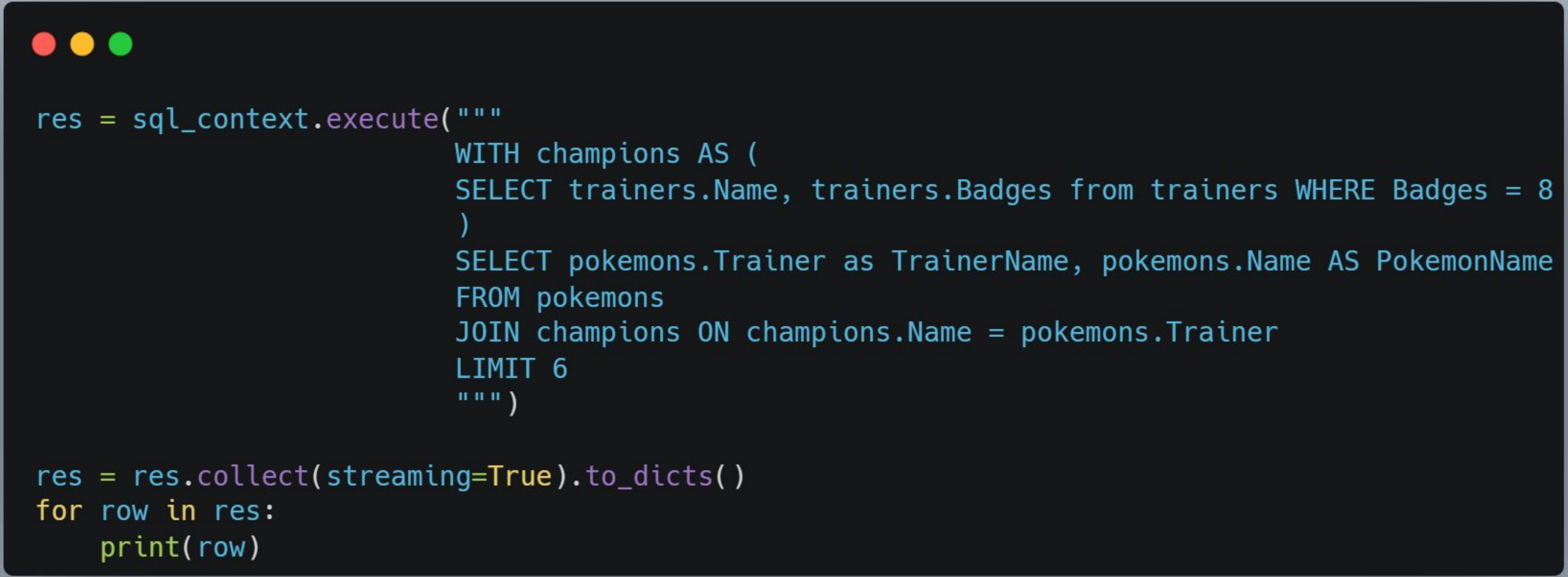
```
res = sql_context.execute(""" select Id,Trainer from pokemons where pokemons.Name = 'charizard'""")

res.sink_csv('result.csv')
```

Utilize LazyFrames for big datasets



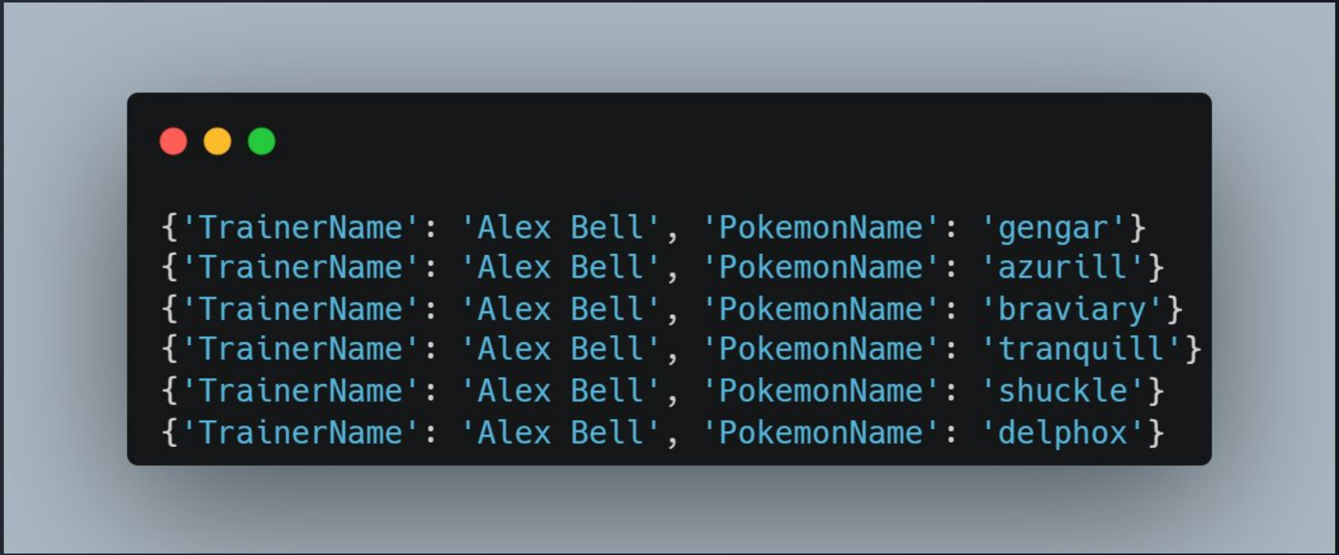
```
Id,Trainer
225,Susan White
1582,Richard Foster
1835,Wesley Bailey
.
.
.
199994483,Gabriella Fox
199994500,Daniel Morris
199996392,Jennifer Ramos
199996930,Eric Marshall
```

Common Table Expressions (CTEs)



```
res = sql_context.execute("""
                          WITH champions AS (
                          SELECT trainers.Name, trainers.Badges from trainers WHERE Badges = 8
                          )
                          SELECT pokemons.Trainer as TrainerName, pokemons.Name AS PokemonName
                          FROM pokemons
                          JOIN champions ON champions.Name = pokemons.Trainer
                          LIMIT 6
                          """)

res = res.collect(streaming=True).to_dicts()
for row in res:
    print(row)
```

Common Table Expressions (CTEs)



```
{'TrainerName': 'Alex Bell', 'PokemonName': 'gengar'}
{'TrainerName': 'Alex Bell', 'PokemonName': 'azurill'}
{'TrainerName': 'Alex Bell', 'PokemonName': 'braviary'}
{'TrainerName': 'Alex Bell', 'PokemonName': 'tranquill'}
{'TrainerName': 'Alex Bell', 'PokemonName': 'shuckle'}
{'TrainerName': 'Alex Bell', 'PokemonName': 'delphox'}
```

Thank You!

tomlevy02@gmail.com