

Data Engineering 101

Pandas vs PySpark vs SQL



Shwetank Singh
GritSetGrow - GSGLearn.com

Data Loading Example

PYSPARK

```
df = spark.read.csv("file.csv")
```

PANDAS

```
df = pd.read_csv("file.csv")
```

SQL

```
SELECT * FROM table_name
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Filtering Rows

PYSPARK

```
df.filter(df['column'] > 50)
```

PANDAS

```
df[df['column'] > 50]
```

SQL

```
SELECT * FROM table_name  
WHERE column > 50
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Selecting Columns

PYSPARK

```
df.select("column1", "column2")
```

PANDAS

```
df[['column1', 'column2']]
```

SQL

```
SELECT column1, column2  
FROM table_name
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Grouping and Aggregation

PYSPARK

```
df.groupBy("column").agg({"column": "sum"})
```

PANDAS

```
df.groupby('column').sum()
```

SQL

```
SELECT column, SUM(column)  
FROM table_name  
GROUP BY column
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Joining Tables/DataFrames

PYSPARK

```
df1.join(df2, df1.id == df2.id)
```

PANDAS

```
df1.merge(df2, on='id')
```

SQL

```
SELECT * FROM table1  
        JOIN table2  
    ON table1.id = table2.id
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Adding a New Column

PYSPARK

```
df = df.withColumn("new_col", df["column"] + 10)
```

PANDAS

```
df['new_col'] = df['column'] + 10
```

SQL

```
ALTER TABLE table_name  
ADD COLUMN new_col INT;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Removing a Column

PYSPARK

```
df = df.drop("column")
```

PANDAS

```
df = df.drop('column', axis=1)
```

SQL

```
ALTER TABLE table_name DROP COLUMN column;
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Handling Missing Data

PYSPARK

`df.fillna(value)`

PANDAS

`df.fillna(value)`

SQL

```
UPDATE table_name  
    SET column = value  
 WHERE column IS NULL
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Sorting Data

PYSPARK

```
df.orderBy(df['column'].desc())
```

PANDAS

```
df.sort_values('column', ascending=False)
```

SQL

```
SELECT * FROM table_name  
ORDER BY column DESC
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Window Functions

PYSPARK

```
from pyspark.sql.window import Window  
window = Window.partitionBy('column')  
df.withColumn('rank', rank().over(window))
```

PANDAS

```
df['rank'] = df['column'].rank()
```

SQL

```
SELECT column,  
RANK() OVER (PARTITION BY column)
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Saving Data

PYSPARK

```
df.write.csv("output.csv")
```

PANDAS

```
df.to_csv("output.csv")
```

SQL

```
COPY (SELECT * FROM table_name)
      TO 'output.csv' WITH CSV
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Performance

PYSPARK

Optimized for distributed processing

PANDAS

Limited by memory

SQL

Dependent on database capabilities



Shwetank Singh
GritSetGrow - GSGLearn.com



Data Type Support

PYSPARK

Supports various data types including complex types like ArrayType and MapType

PANDAS

Basic data types (e.g., int, float, str, datetime)

SQL

Supports a wide range of data types depending on the database (e.g., int, varchar, date)



Shwetank Singh
GritSetGrow - GSGLearn.com



Handling Null Values

PYSPARK

```
df.fillna(value).dropna().replace(to_replace, value)
```

PANDAS

```
df.fillna(value).dropna().replace(to_replace, value)
```

SQL

COALESCE, NULLIF, IS NULL, IS NOT NULL



Shwetank Singh
GritSetGrow - GSGLearn.com



Pivoting/Unpivoting Data

PYSPARK

```
df.groupBy("column").pivot("column_to_pivot")  
    .sum("value")
```

PANDAS

```
df.pivot_table(values='value', index='column',  
               columns='column_to_pivot', aggfunc='sum')
```

SQL

```
PIVOT (SUM(value) FOR column_to_pivot  
       IN (...)) (DBMS dependent)
```



Shwetank Singh
GritSetGrow - GSGLearn.com



Iterating Over Rows

PYSPARK

Not recommended due to distributed nature; use
map or foreach

PANDAS

for index, row in df.iterrows()
(slower for large datasets)

SQL

Typically not used in SQL; handled via queries



Shwetank Singh
GritSetGrow - GSGLearn.com



Performance Optimization

PYSPARK

Catalyst optimizer, Tungsten execution engine

PANDAS

N/A - single-threaded execution

SQL

Indexing, query optimization, parallel execution
(depends on DBMS)



Shwetank Singh
GritSetGrow - GSGLearn.com



DataFrame/Query Caching

PYSPARK

`df.cache()`

PANDAS

N/A - entire DataFrame is in memory

SQL

CACHE TABLE or CREATE INDEX (DBMS dependent)



Shwetank Singh
GritSetGrow - GSGLearn.com



Schema Management

PYSPARK

Supports schema enforcement and evolution via
DataFrame schema

PANDAS

Schema is inferred; can be set manually

SQL

Managed via CREATE TABLE and ALTER TABLE



Shwetank Singh
GritSetGrow - GSGLearn.com



Data Visualization

PYSPARK

Not directly supported, but can integrate with
libraries like matplotlib

PANDAS

Supports matplotlib, seaborn, etc.

SQL

Not supported; data is visualized via external tools
like Tableau, Power BI



Shwetank Singh
GritSetGrow - GSGLearn.com



Debugging

PYSPARK

Log-based debugging, `explain()` for execution plans

PANDAS

Direct inspection with `print`, `info()`, `describe()`

SQL

`EXPLAIN` for query plans, error messages



Shwetank Singh
GritSetGrow - GSGLearn.com



File Formats Supported

PYSPARK

CSV, JSON, Parquet, ORC, Avro, etc.

PANDAS

CSV, Excel, JSON, HDF5, etc.

SQL

Varies; often requires external tools or extensions
for non-tabular formats



Shwetank Singh
GritSetGrow - GSGLearn.com



Complex Aggregations

PYSPARK

```
df.groupBy("column") \  
.agg({"col1": "sum", "col2": "max"})
```

PANDAS

```
df.groupby("column").agg({"col1": "sum", "col2": "max"})
```

SQL

Complex aggregations via GROUP BY, subqueries



Shwetank Singh
GritSetGrow - GSGLearn.com



Window Functions

PYSPARK

Extensive support, e.g., rank(), row_number(), lag()

PANDAS

Limited support via rolling, expanding

SQL

Extensive support, e.g., ROW_NUMBER(), RANK(),
LAG()



Shwetank Singh
GritSetGrow - GSGLearn.com



Date/Time Functions

PYSPARK

`date_add()`, `date_sub()`, `datediff()`, `to_date()`

PANDAS

`pd.to_datetime()`, `df['date'].dt` functions

SQL

`DATEADD`, `DATEDIFF`, `TO_DATE`, `TO_CHAR`



Shwetank Singh
GritSetGrow - GSGLearn.com



Security

PYSPARK

Access control at Spark level, encryption

PANDAS

Depends on the environment (e.g., file permissions)

SQL

Role-based access control, data encryption



Shwetank Singh
GritSetGrow - GSGLearn.com



Error Handling

PYSPARK

Exceptions handling with try-except, logging

PANDAS

Exceptions handling with try-except, logging

SQL

Error codes and messages, transaction control (e.g., rollback)



Shwetank Singh
GritSetGrow - GSGLearn.com



Data Representation

PYSPARK

Distributed DataFrame (Resilient Distributed Dataset)

PANDAS

In-memory DataFrame

SQL

Table



Shwetank Singh
GritSetGrow - GSGLearn.com



Execution Mode

PYSPARK

Lazy Evaluation

PANDAS

Eager Evaluation

SQL

Eager Evaluation



Shwetank Singh
GritSetGrow - GSGLearn.com



Parallel Processing

PYSPARK

Yes, across multiple nodes

PANDAS

No, single machine

SQL

No, typically single machine unless parallel query execution is supported



Shwetank Singh
GritSetGrow - GSGLearn.com



Handling Big Data

PYSPARK

Efficiently handles large datasets

PANDAS

Not designed for big data

SQL

Limited by database capabilities



Shwetank Singh
GritSetGrow - GSGLearn.com



Data Source Integration

PYSPARK

Reads from HDFS, S3, JDBC, etc.

PANDAS

Reads from local files, databases via connectors

SQL

Reads from various database tables



Shwetank Singh
GritSetGrow - GSGLearn.com



THANK
you