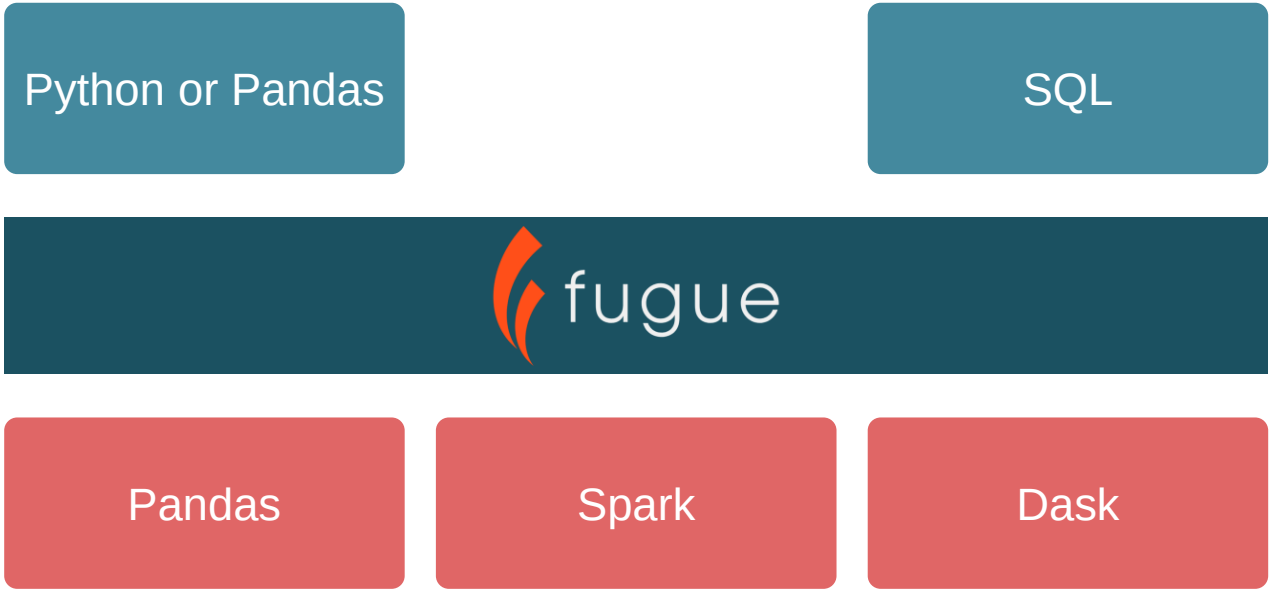
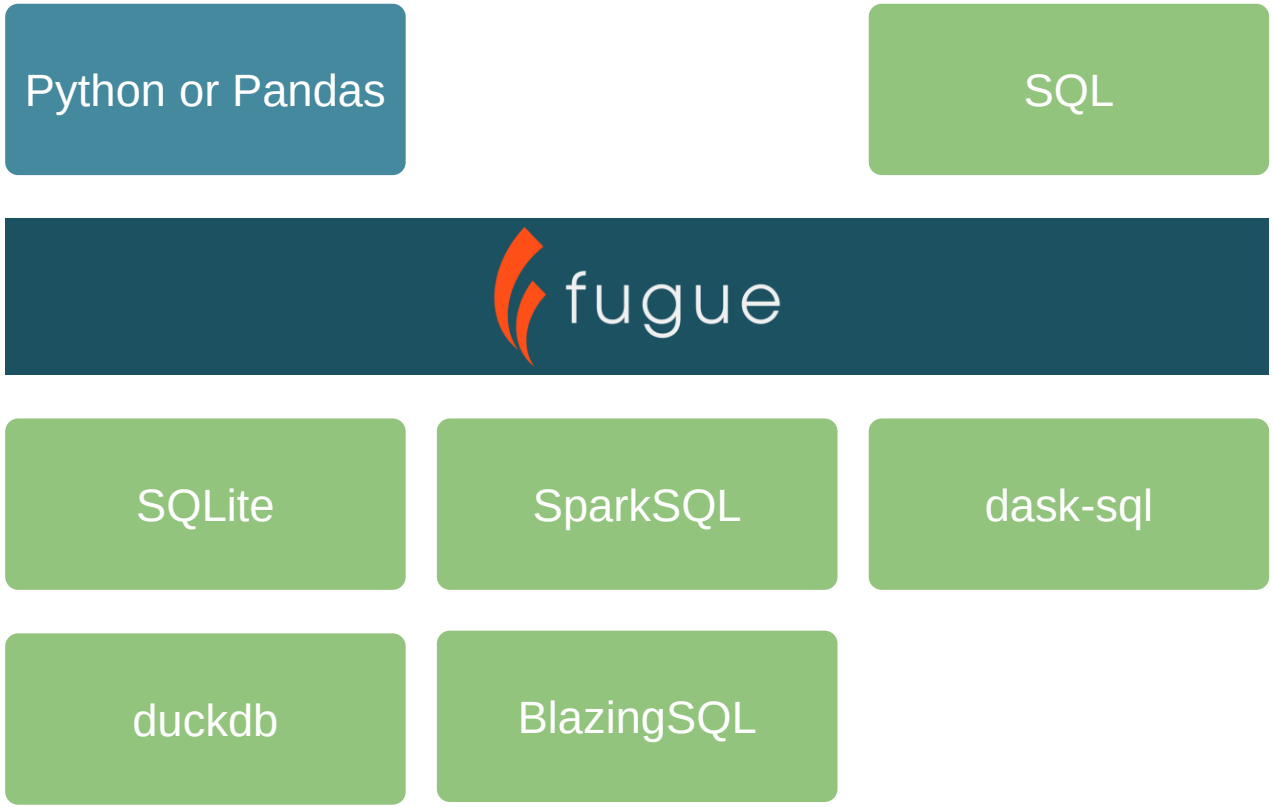
Fugue SQL - SQL for Pandas, Spark and Dask

Kevin Kho
Rowan Molony

Fugue - An Abstraction Layer



FugueSQL - Different Backends



Enhanced Syntax

```
%%fsql
SELECT * FROM data
SAVE OVERWRITE "/tmp/f.csv"(header=true)

temp = SELECT *
       FROM (LOAD "/tmp/f.csv" (header=true))
       WHERE number = 1

output = SELECT word FROM temp
SAVE OVERWRITE "/tmp/output.csv"(header=true)

new = LOAD "/tmp/output.csv"(header=true)
PRINT new
```

Added Keywords

```
%%fsql
a = SELECT * FROM df
TAKE 2 ROWS PRESORT number DESC          -- a is consumed by TAKE
PRINT
b = SELECT * FROM df
TAKE 2 ROWS FROM b PRESORT number DESC   -- equivalent explicit synax
PRINT
```

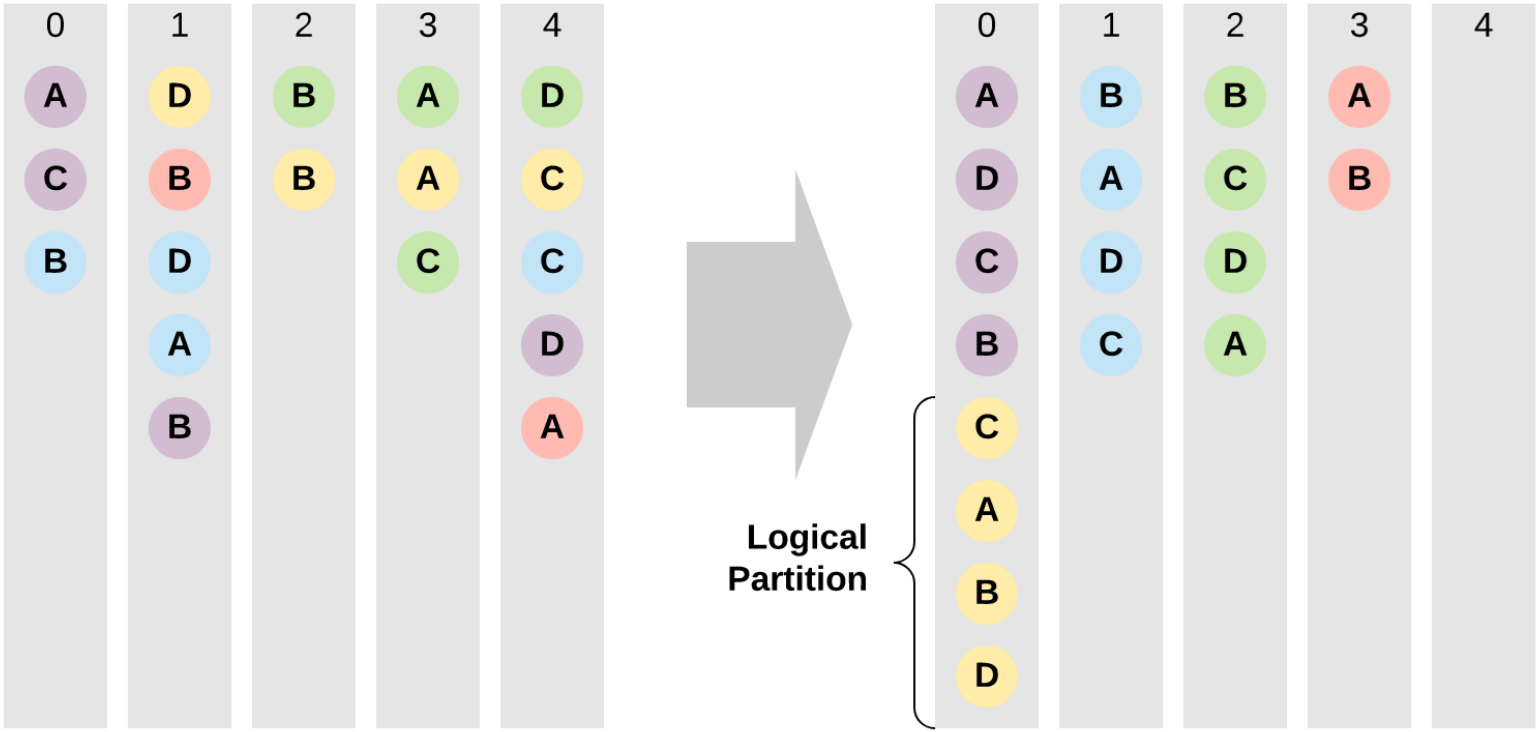
Operators

- RENAME COLUMNS
- FILL NULL
- DROP ROWS
- SAMPLE
- TAKE
- TRANSFORM
- OUTPUT

Scaling to Distributed Compute

	id	date	value	shift
0	A	2020-01-03	30.0	NaN
1	A	2020-01-02	NaN	30.0
2	A	2020-01-01	10.0	NaN
3	B	2020-01-03	40.0	NaN
4	B	2020-01-02	NaN	40.0
5	B	2020-01-01	20.0	NaN

Partitions in Distributed Compute



Scaling to Distributed Compute

```
# schema: *, shift:double
def shift(df: pd.DataFrame) -> pd.DataFrame:
    df['shift'] = df['value'].shift()
    return df
```

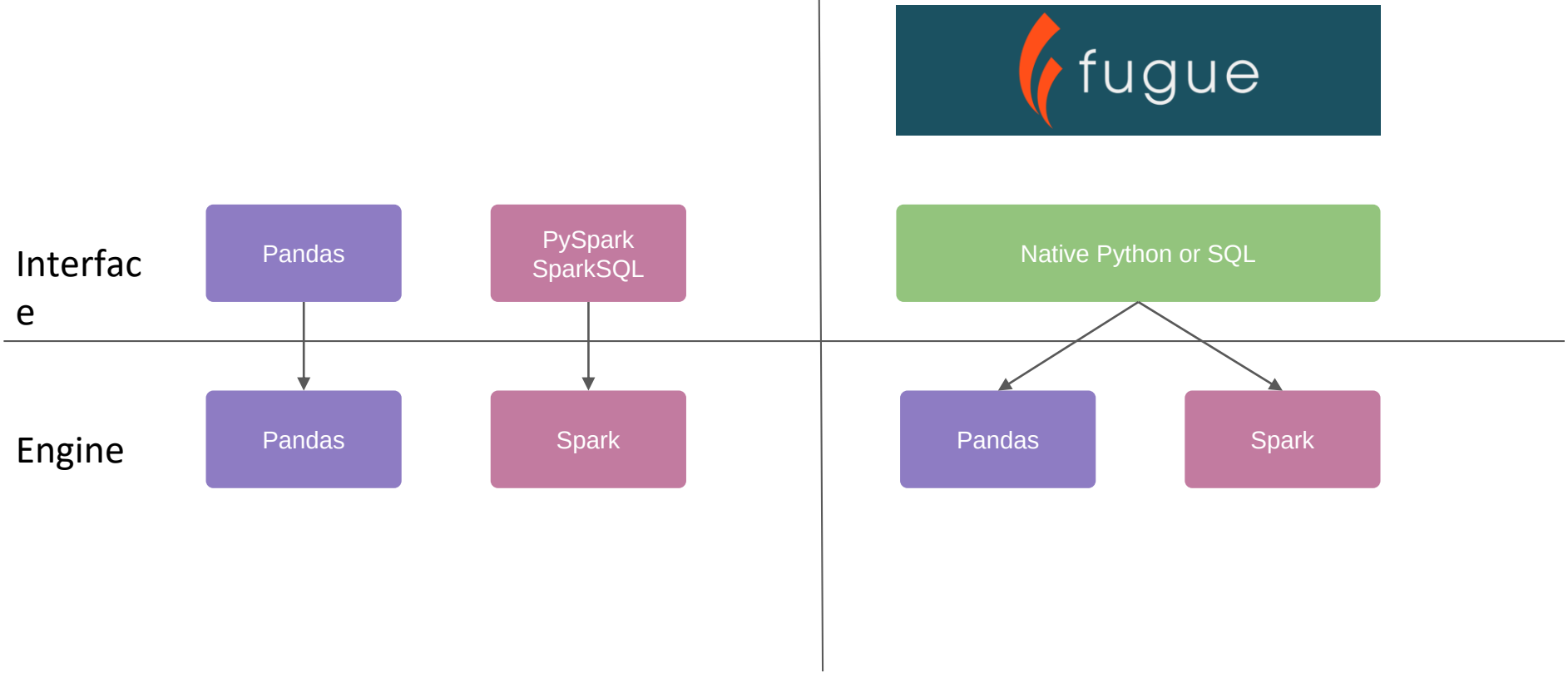
```
%%fsql spark
a = SELECT * FROM df
TRANSFORM a PREPARTITION BY id PRESORT date DESC USING shift
PRINT

b = SELECT * FROM df
TRANSFORM a USING shift     -- default partition
PRINT
```

Big Data Operations

- PARTITION
- BROADCAST
- PERSIST

Decoupling Logic and Execution



Fugue SQL Notebook

In [ ]:
```
import pandas as pd
```

In [ ]:
```
from fugue_notebook import setup
setup()
```

In [ ]:
```
df = pd.DataFrame({'a':[1,2,3,4], 'b':[1,2,3,4]})
df.to_csv('df.csv', index=False)
```

In [ ]:
```
%%fsql
-- This SQL cell sees the dataframe defined in the previous cell
SELECT *
  FROM df
 WHERE a > 2
 PRINT
```

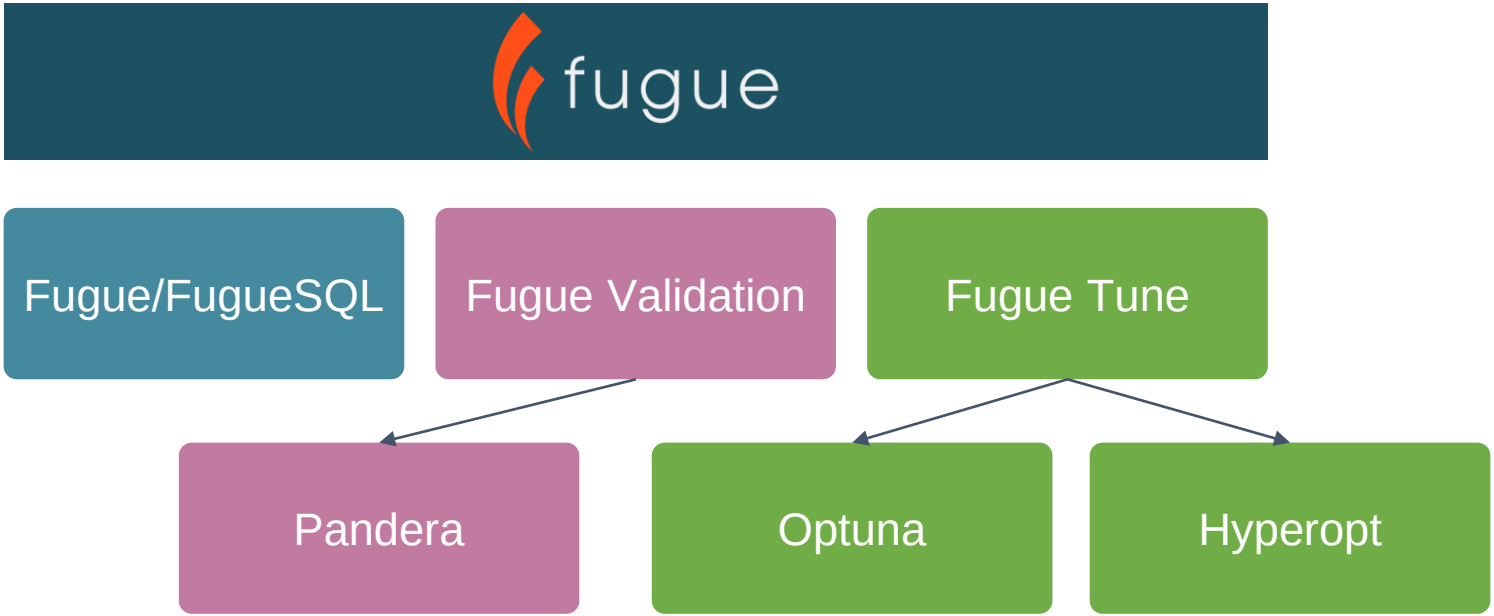
In [ ]:
```
%%fsql
df2 = LOAD "/Users/kevinkho/Work/fugue/df.csv" (header=TRUE, infer_schema=TRUE)
SELECT *
  FROM df2
 WHERE b < 2
 PRINT
  SAVE OVERWRITE "/Users/kevinkho/Work/fugue/df.csv"
```

In [ ]:

Takeaways

- FugueSQL provides a SQL interface for in memory DataFrames (Pandas, Spark and Dask)
- An enhanced syntax allows it to be the dominant grammar for compute workflows
- It can scale to distributed compute with additional keywords
- Use cases around SQL workflows that get send to BI tools

Broader Fugue Project



Contact Us

Github

- https://github.com/fugue-project/fugue

Fugue Tutorials

- https://fugue-tutorials.readthedocs.io/tutorials/fugue_sql/

Slack

- https://slack.fugue.ai/

Questions