Ibis

and interfaces

Intro

Gil Forsyth
Voltron Data

gforsyth

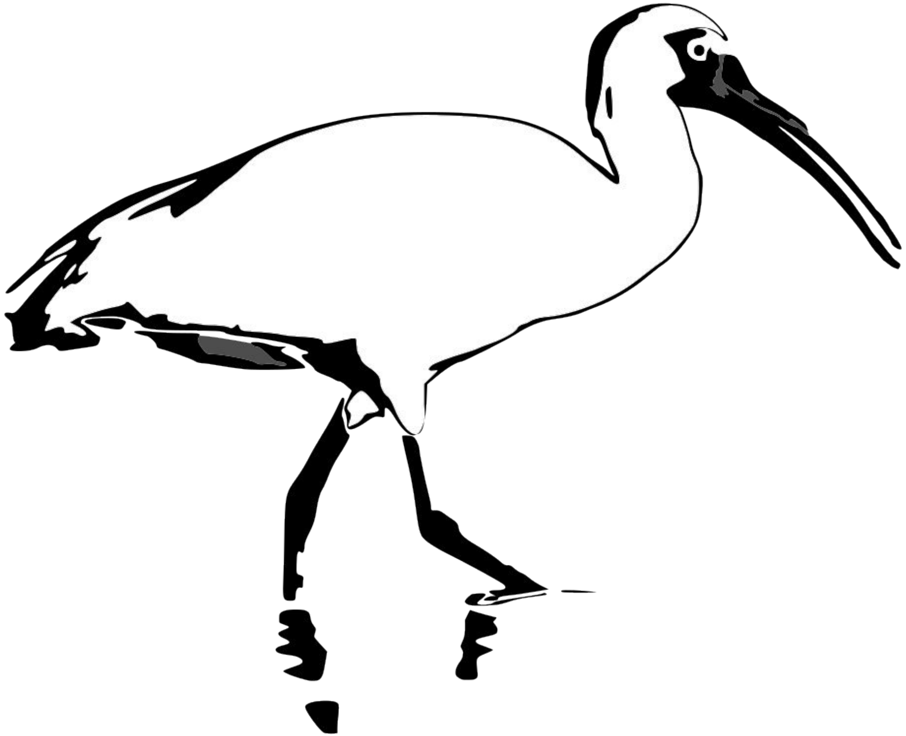
@gforsyth@fosstodon.org

What is Ibis?

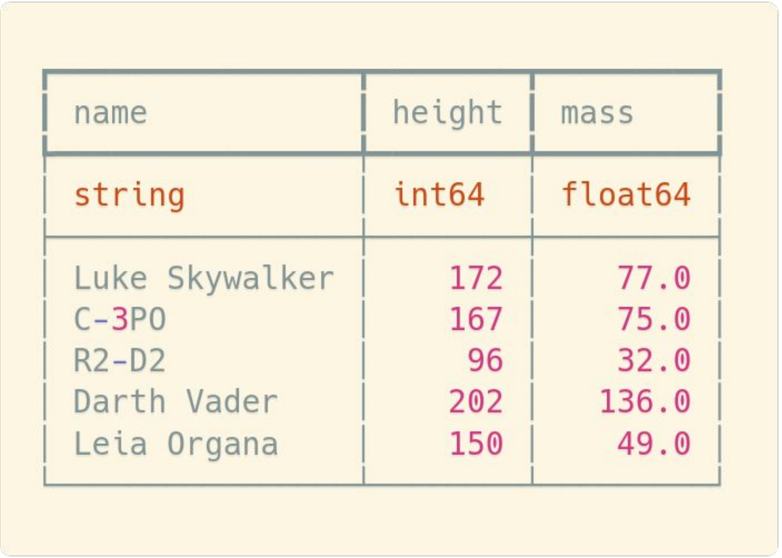
Open-source (Apache 2.0)

Pure Python

DataFrame interface

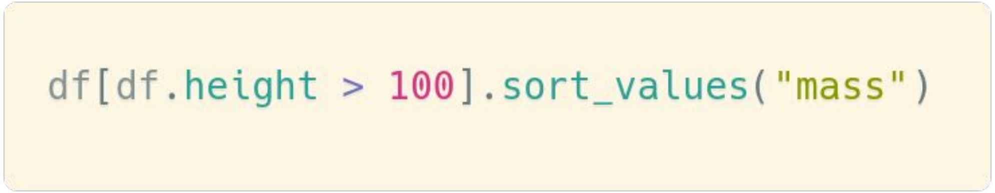


Tabular Data

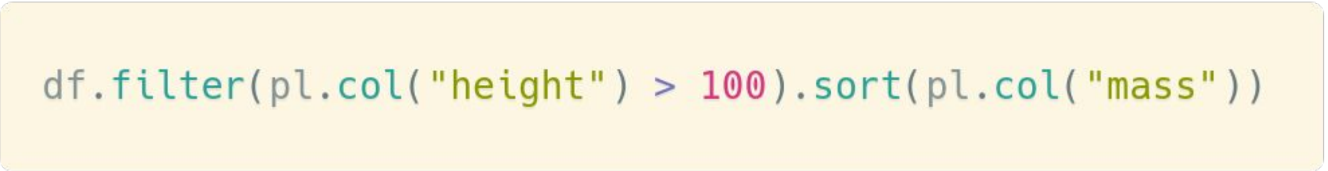


```
┏━━━━━━━━━━━━━━━━┳━━━━━━━━┳━━━━━━━━━┓
┃ name           ┃ height ┃ mass    ┃
┡━━━━━━━━━━━━━━━━╇━━━━━━━━╇━━━━━━━━━┩
│ string         │ int64  │ float64 │
├────────────────┼────────┼─────────┤
│ Luke Skywalker │    172 │    77.0 │
│ C-3PO          │    167 │    75.0 │
│ R2-D2          │     96 │    32.0 │
│ Darth Vader    │    202 │   136.0 │
│ Leia Organa    │    150 │    49.0 │
└────────────────┴────────┴─────────┘
```

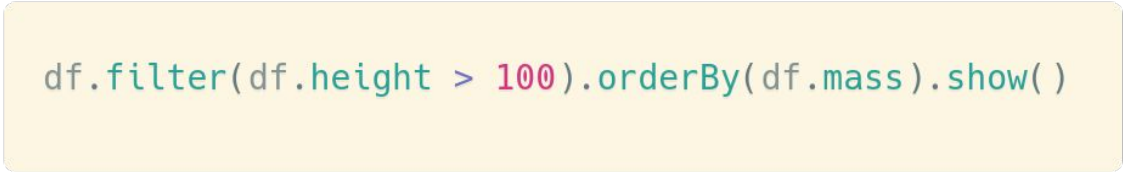
Query tabular data



```
df[df.height > 100].sort_values("mass")
```

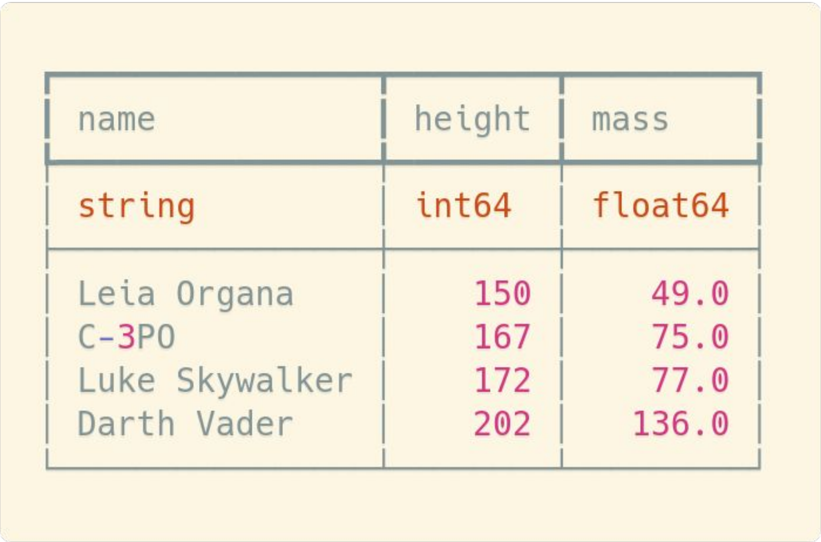


```
df.filter(pl.col("height") > 100).sort(pl.col("mass"))
```



```
df.filter(df.height > 100).orderBy(df.mass).show()
```

Query Result



```
┏━━━━━━━━━━━━━━━━┳━━━━━━━━┳━━━━━━━━━┓
┃ name           ┃ height ┃ mass    ┃
┡━━━━━━━━━━━━━━━━╇━━━━━━━━╇━━━━━━━━━┩
│ string         │ int64  │ float64 │
├────────────────┼────────┼─────────┤
│ Leia Organa    │    150 │    49.0 │
│ C-3PO          │    167 │    75.0 │
│ Luke Skywalker │    172 │    77.0 │
│ Darth Vader    │    202 │   136.0 │
└────────────────┴────────┴─────────┘
```

Interface vs Engine

In PyData land, the interface and the compute engine are tightly* coupled.

`pandas` interface → `pandas` engine

`polars` interface → `polars` engine

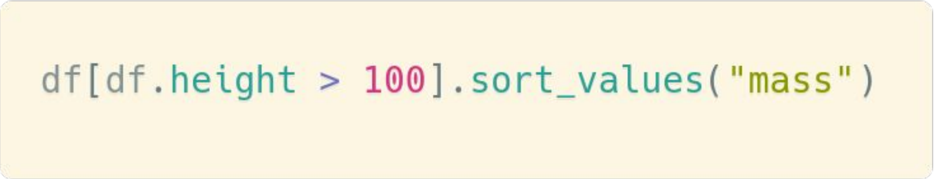
`pyspark` interface → `spark` engine

*: Mostly

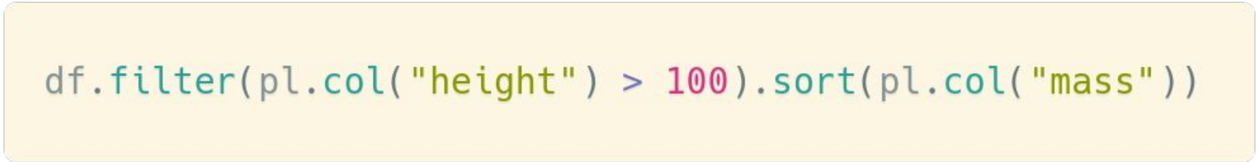
Remember these queries?

Interface

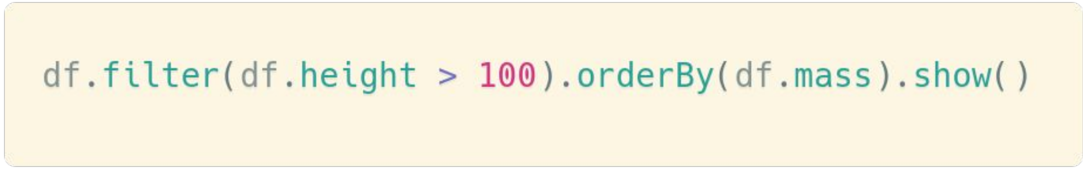
Engine



```
df[df.height > 100].sort_values("mass")
```



```
df.filter(pl.col("height") > 100).sort(pl.col("mass"))
```

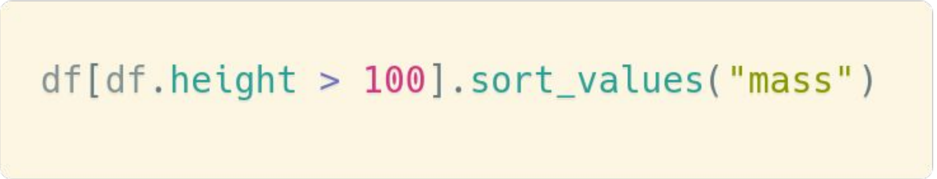


```
df.filter(df.height > 100).orderBy(df.mass).show()
```

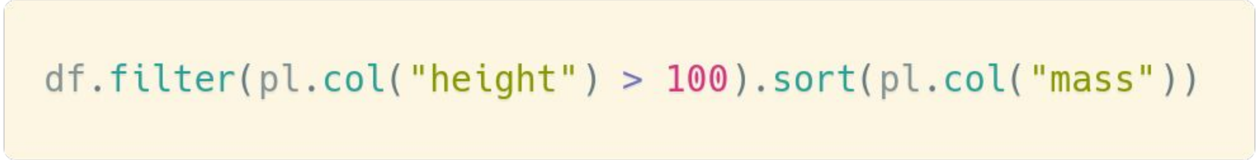
Remember these queries?

Interface

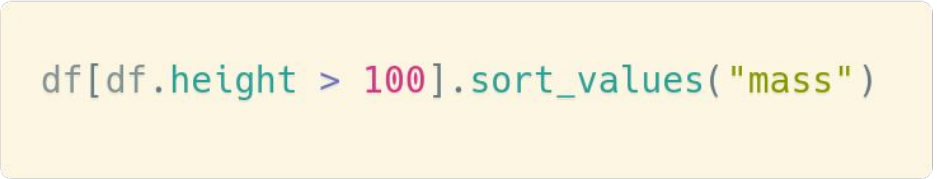
Engine



```
df[df.height > 100].sort_values("mass")
```



```
df.filter(pl.col("height") > 100).sort(pl.col("mass"))
```

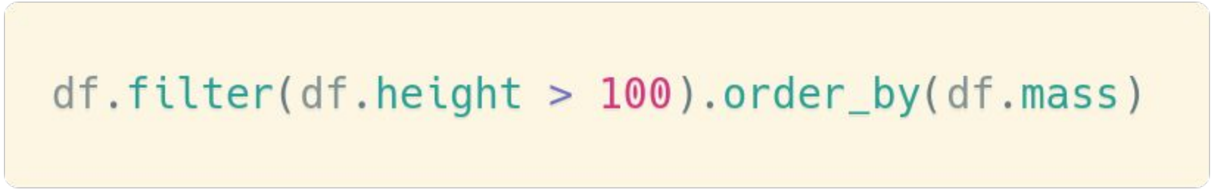


```
df[df.height > 100].sort_values("mass")
```

Remember these queries?

Interface

Engine



```
df.filter(df.height > 100).order_by(df.mass)
```

⋮

What is Ibis?

Ibis provides a Pythonic dataframe interface to 20+ engines.

Ibis helps you build the query, but Ibis is *not* a compute engine

We hand the query to the engine of your choice

What is Ibis?

Ibis provides a Pythonic dataframe interface to 20+ engines.

Ibis helps you build the query, but Ibis is *not* a compute engine

We hand the query to the engine you have access to at `$DAY_JOB`

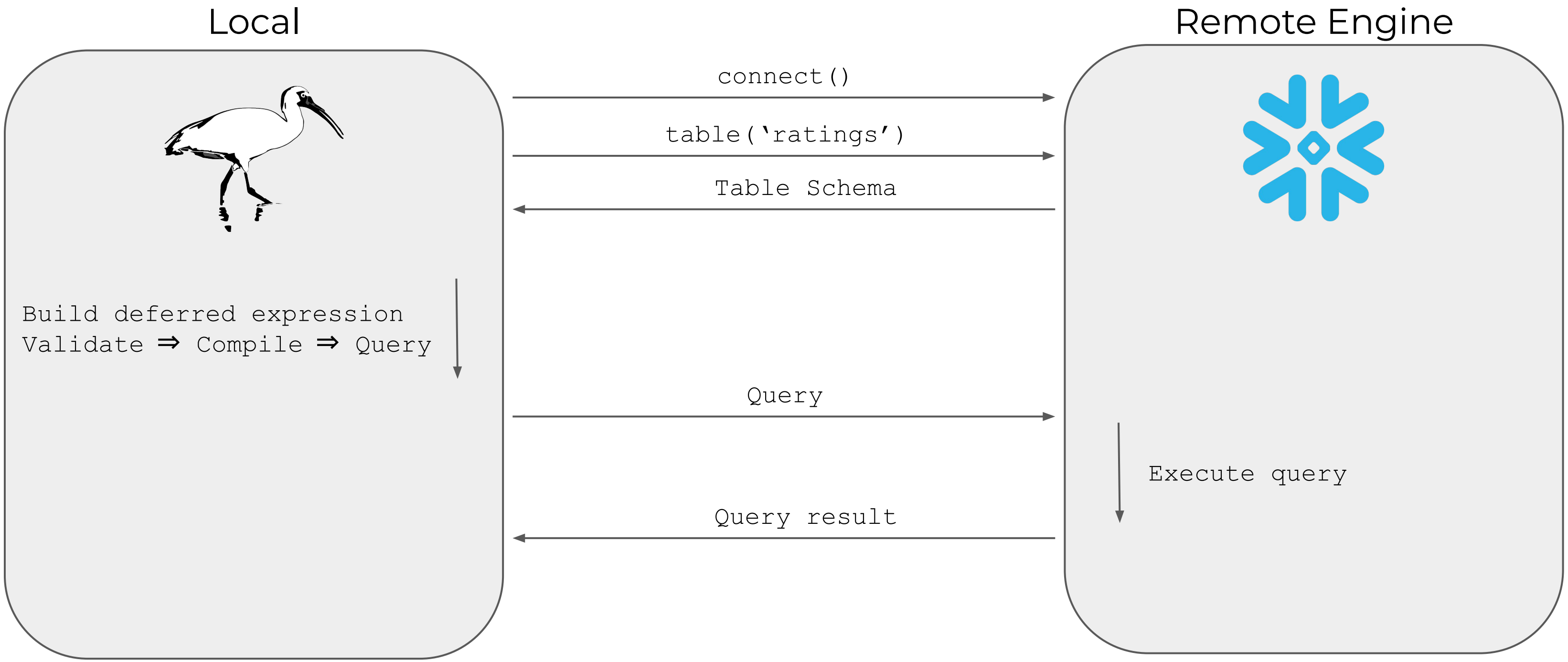
What is Ibis?

Ibis provides a Pythonic dataframe interface to 20+ engines.

Ibis helps you build the query, but Ibis is *not* a compute engine

We hand the query to the engine that has the data you need

Remote processing of remote data



Local

Remote Engine

connect()

table('ratings')

Table Schema

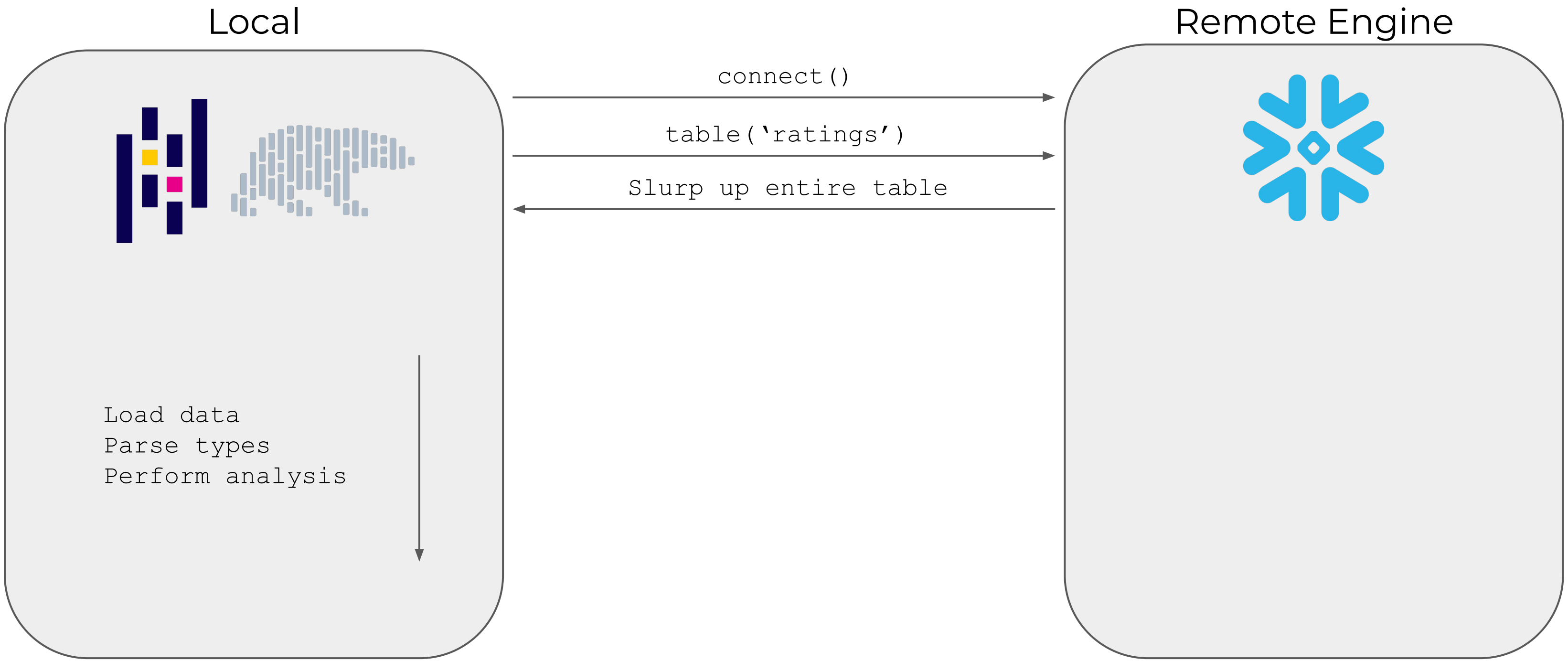
Build deferred expression
Validate ⇒ Compile ⇒ Query

Query

Execute query

Query result

Local processing of remote data



Local

Remote Engine

connect()

table('ratings')

Slurp up entire table

```
Load data
Parse types
Perform analysis
```

Snowflake Demo

It is ok to use the tools you know

If the tools you are using meet your needs and you like them, they are good tools.

Don't let me or anyone else tell you otherwise.

The interface is important!

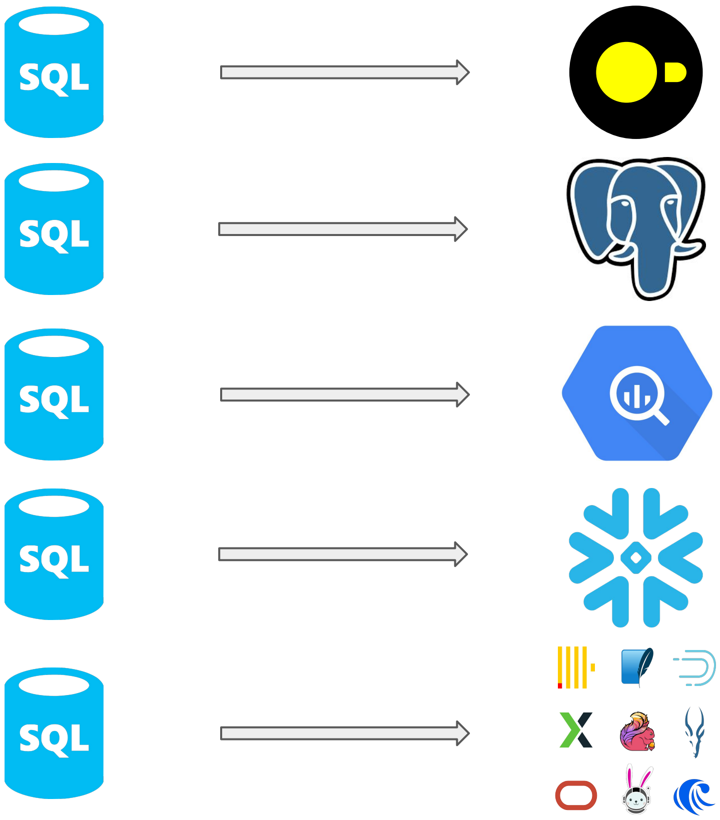
And the engine is important!

Don't let the *engine* dictate the *interface.*

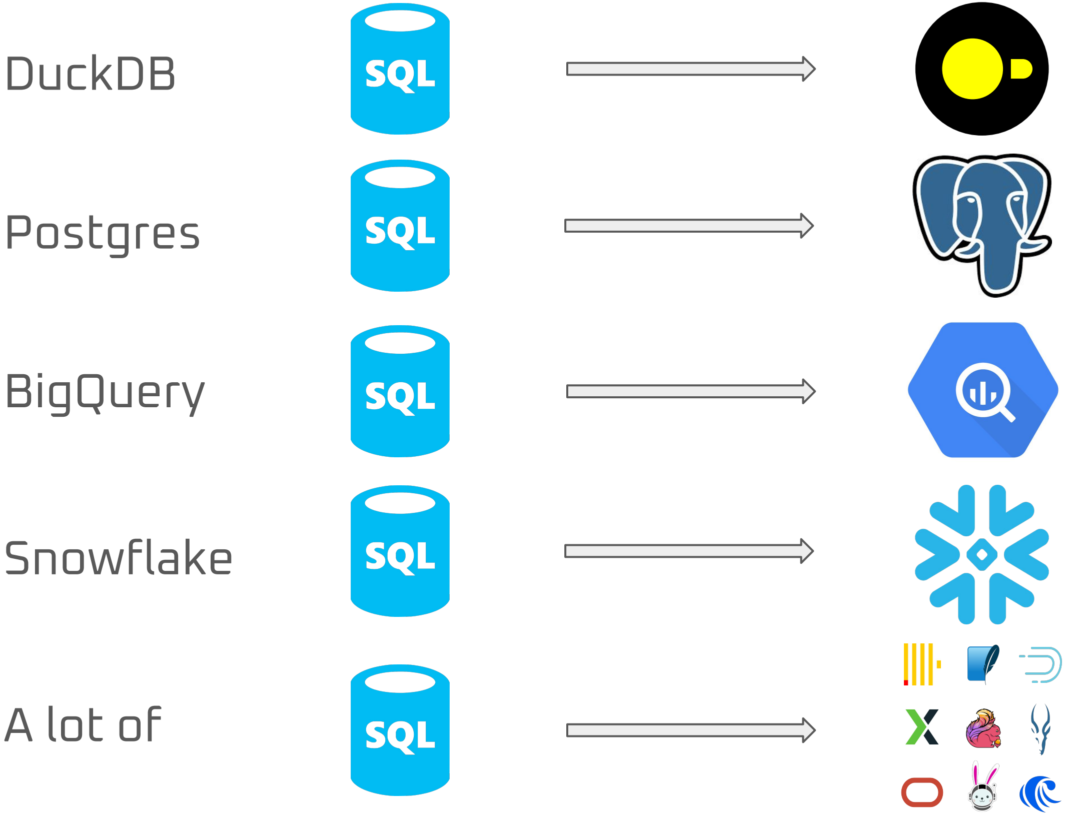
The elephantine duck in the room...

There's another universal interface for working with tabular data

SQL: the ubiquitous interface



SQL: the ubiquitous interface



DuckDB

Postgres

BigQuery

Snowflake

A lot of

Questions with no (single) answer

- Does a week start on Sunday or Monday?
- Are the days of a week 0-indexed or 1-indexed?
- Do nulls sort ascending, or descending, or always first, or always last?
- Given a function to compute $\log_b x$, is the function signature `log(b, x)` or `log(x, b)`?

SQL ain't standard

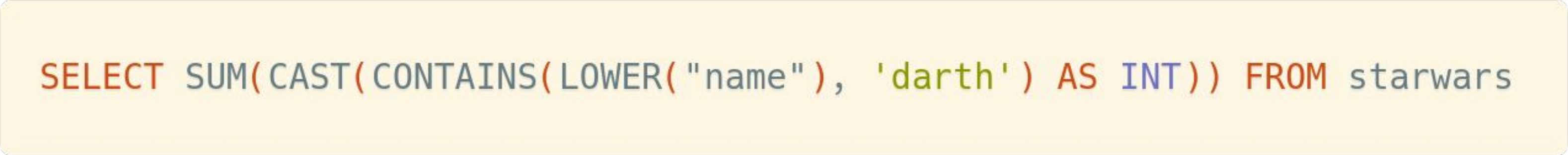
```sql
SELECT SUM(CAST(CONTAINS(LOWER("name"), 'darth') AS INT)) FROM starwars
```

```sql
SELECT SUM(CAST(STRPOS(LOWER("name"), 'darth') > 0 AS INT)) FROM "starwars"
```

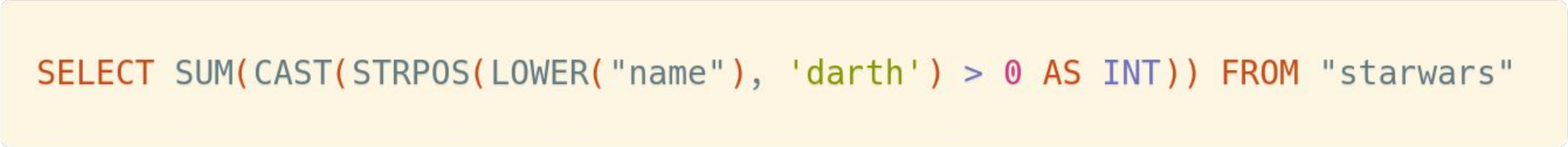
```sql
SELECT SUM(CAST(STRPOS(LOWER(`name`), 'darth') > 0 AS INT64)) FROM `starwars`
```

```sql
SELECT SUM(IIF(CONTAINS(LOWER([name]), 'darth'), 1, 0)) FROM [starwars]
```

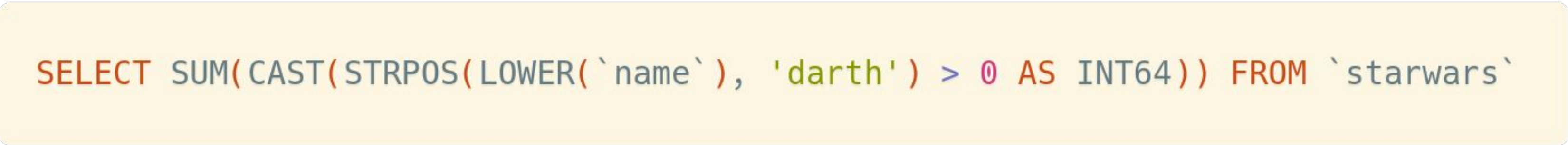
Ibis will do this for you



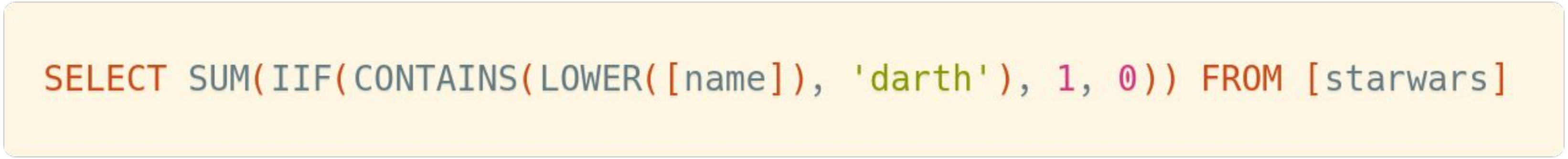
```
SELECT SUM(CAST(CONTAINS(LOWER("name"), 'darth') AS INT)) FROM starwars
```



```
SELECT SUM(CAST(STRPOS(LOWER("name"), 'darth') > 0 AS INT)) FROM "starwars"
```

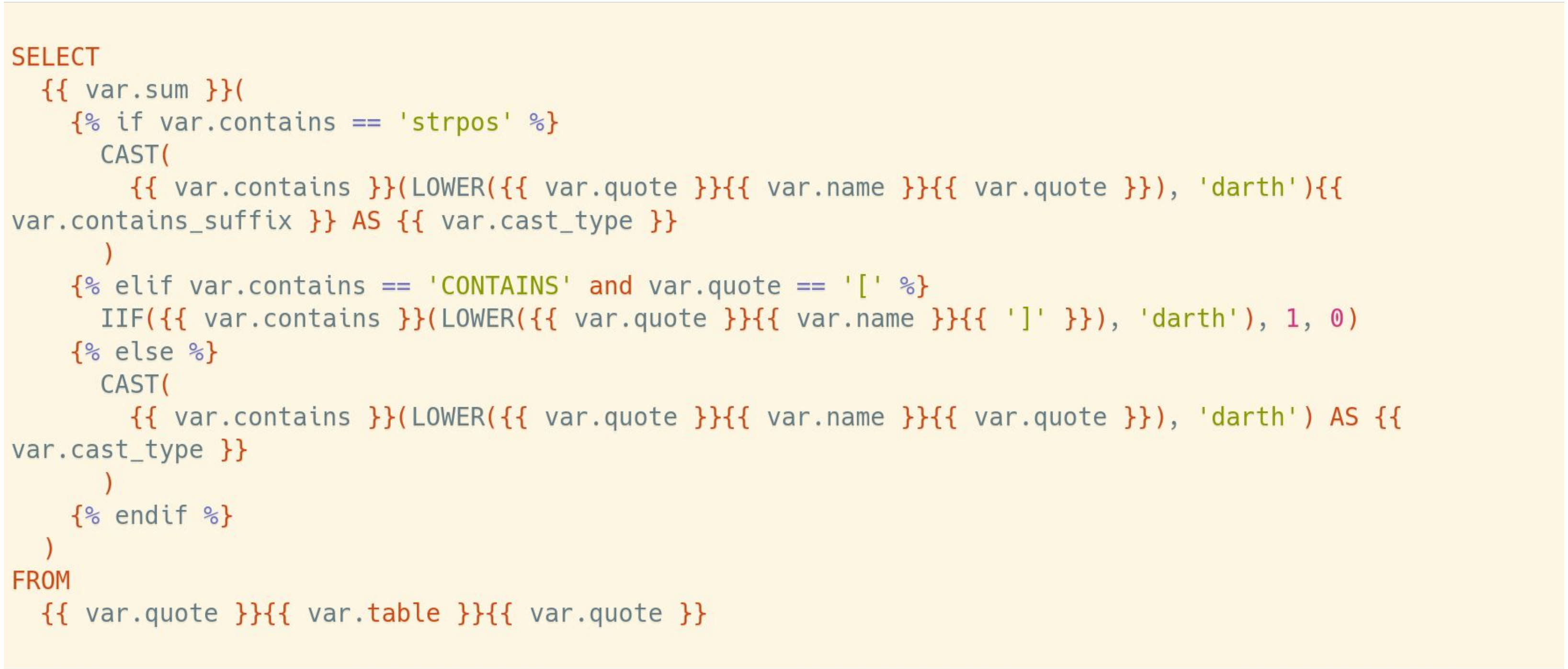


```
SELECT SUM(CAST(STRPOS(LOWER(`name`), 'darth') > 0 AS INT64)) FROM `starwars`
```



```
SELECT SUM(IIF(CONTAINS(LOWER([name]), 'darth'), 1, 0)) FROM [starwars]
```

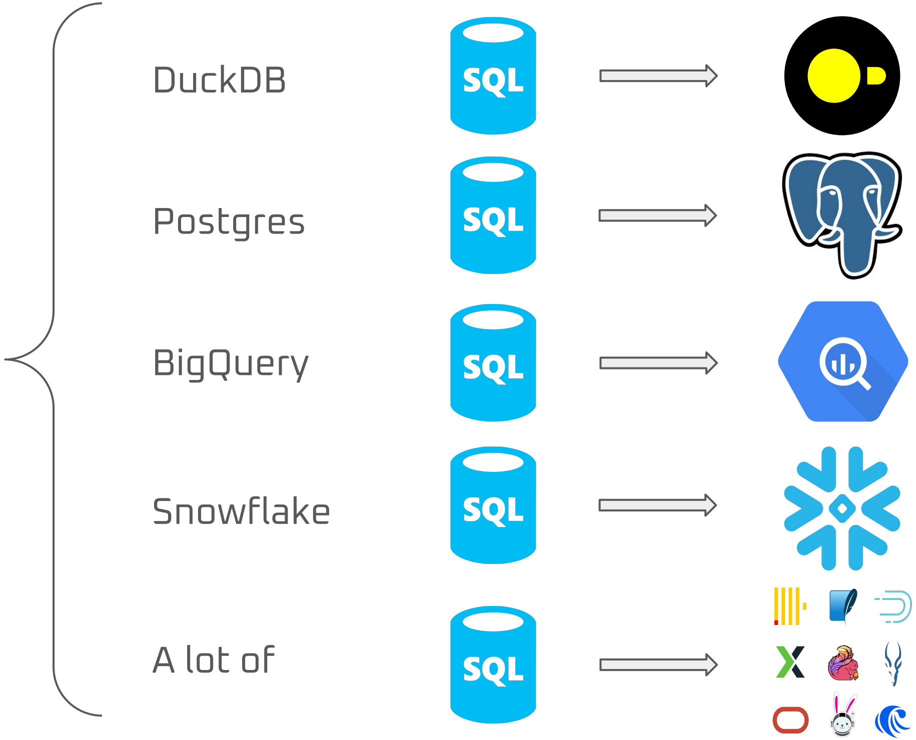
You *could* do this...



```
SELECT
  {{ var.sum }}(
    {% if var.contains == 'strpos' %}
      CAST(
        {{ var.contains }}(LOWER({{ var.quote }}{{ var.name }}{{ var.quote }}), 'darth'){{
var.contains_suffix }} AS {{ var.cast_type }}
      )
    {% elif var.contains == 'CONTAINS' and var.quote == '[' %}
      IIF({{ var.contains }}(LOWER({{ var.quote }}{{ var.name }}{{ ']' }}), 'darth'), 1, 0)
    {% else %}
      CAST(
        {{ var.contains }}(LOWER({{ var.quote }}{{ var.name }}{{ var.quote }}), 'darth') AS {{
var.cast_type }}
      )
    {% endif %}
  )
FROM
  {{ var.quote }}{{ var.table }}{{ var.quote }}
```

Ibis will do this for you

```
starwars.name.lower().contains("darth").sum()
```



DuckDB

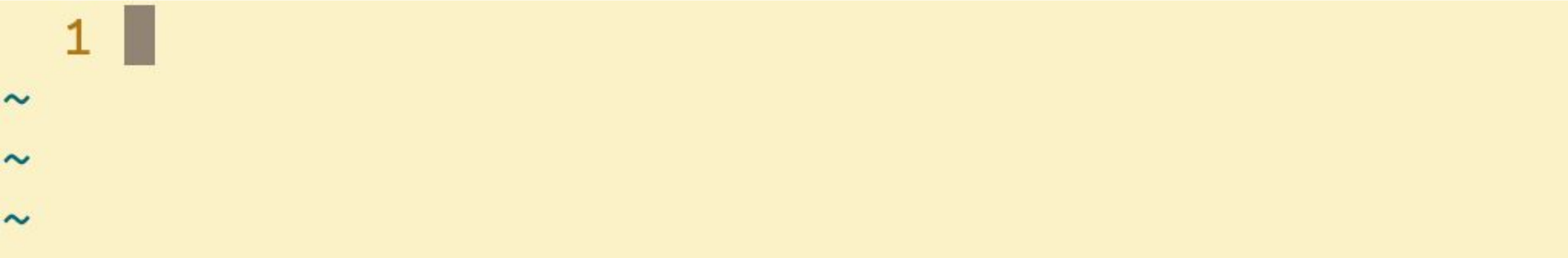
Postgres

BigQuery

Snowflake

A lot of

Building Queries

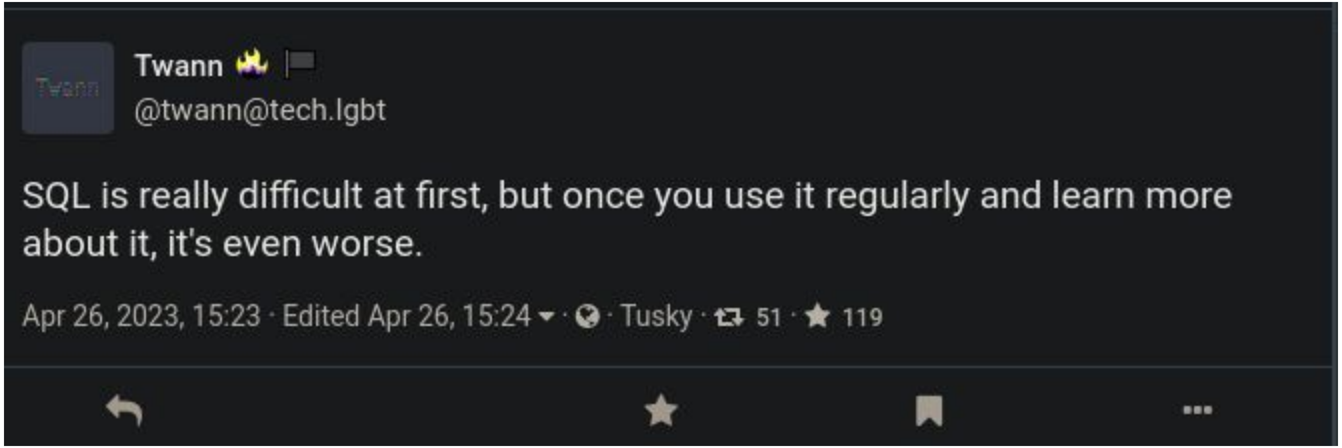


```

~
~
~
```

Building Queries

```
_
~
~
~
~
```



Twann

@twann@tech.lgbt

SQL is really difficult at first, but once you use it regularly and learn more about it, it's even worse.

Apr 26, 2023, 15:23 · Edited Apr 26, 15:24 ▾ · 🌐 · Tusky · 🔁 51 · ★ 119

What comes after the query?

How do you write query results to a parquet file?

Which connector library should you use?

How do you pull query results and put them into an Arrow `RecordBatchReader`?

How do you make `$ENGINE` work with the rest of the PyData ecosystem?

So why would I ever use SQL?

SQL can be fine. It's quite stable, and it's the lingua franca for the data world.

You might know SQL.

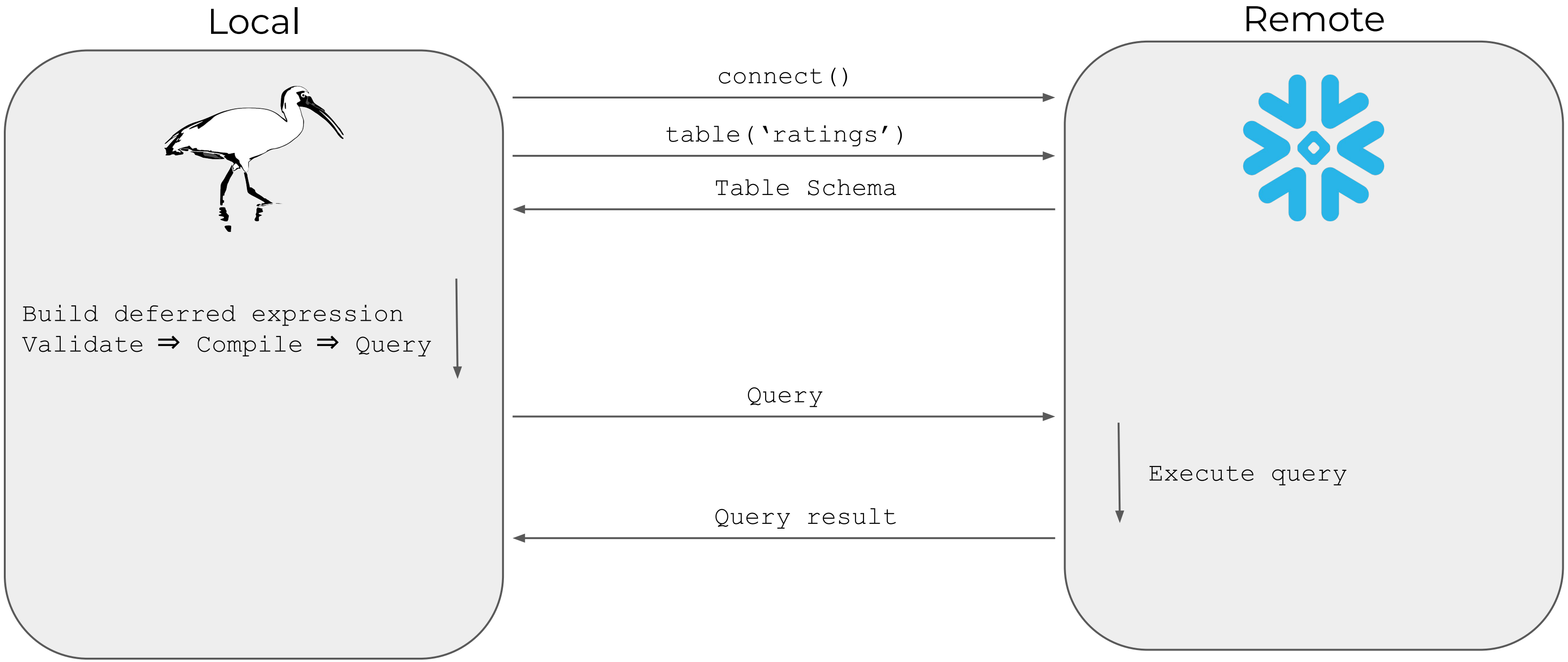
Your coworkers might know SQL.

You might need (or want) to use a SQL database (they are *very* fast).

Don't let the engine dictate the interface

If you want to try DuckDB, don't get blocked by needing to write SQL.

Remember this?



Local

Remote

connect()

table('ratings')

Table Schema

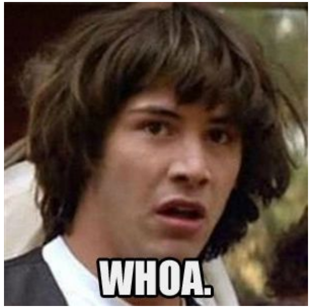
Build deferred expression
Validate ⇒ Compile ⇒ Query

Query

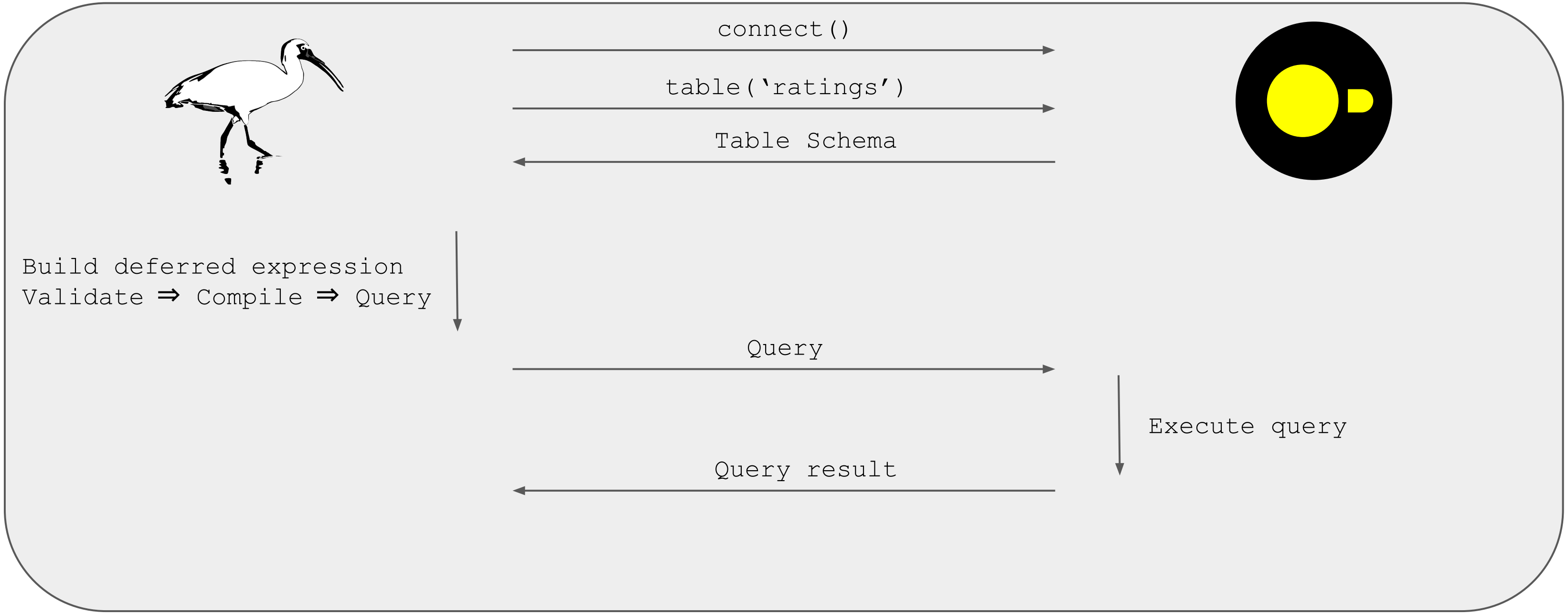
Execute query

Query result

Remember this?



Local



connect()

table('ratings')

Table Schema

Build deferred expression
Validate ⇒ Compile ⇒ Query

Query

Execute query

Query result

A look at local / laptop workflows

There are blazing fast in-process OLAP query engines that can run on your laptop and are orders-of-magnitude faster than pandas.

Run your local analysis faster AND if you need to run it on some huge remote cluster, you can do that without rewriting your whole query.

Demo time

Takeaways

`duckdb`, `polars`, and `datafusion` are all very fast operating on local parquet files.

It is very easy to switch between them using Ibis.

Features I may not have mentioned

I/O for CSV, Parquet, PyArrow, PyArrow streaming, torch, pandas, polars, `__arrow_c_stream__`, `__dataframe__`

Escape valves so you can always talk directly to the engine if there's something Ibis doesn't expose

Integration with other libraries (Altair, VegaLite, Plotly, Streamlit, Hamilton)

Some closing thoughts

SQL isn't going anywhere (truly, it will outlive us all) and the engines are pretty awesome.

Some closing thoughts

Use tools that let you interact with the engine of your choice, and play nice with the software ecosystem you work in.

Some closing thoughts

If you need to work with multiple engines, or if you are thinking of checking out the (very) fast new options, consider using Ibis and future-proofing your queries.

Questions?

https://ibis-project.org/

ibis-project/ibis

ibisData

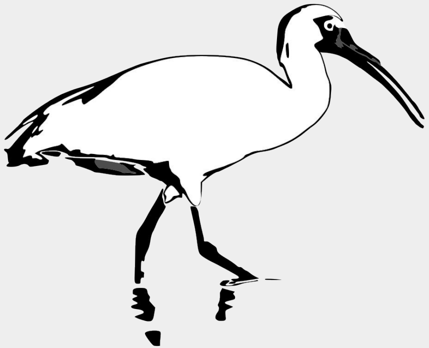
https://ibis-project.zulipchat.com/

Phillip in the Cloud
cpcloud

https://www.linkedin.com/company/ibis-project

```
pip install ibis-framework
pip install ibis-framework[{backend}]

conda install -c conda-forge ibis-framework
                             ibis-bigquery
                             ibis-clickhouse
                             ibis-dask
                             ibis-datafusion
                             ibis-druid
                             ibis-duckdb
                             ibis-exasol
                             ibis-flink
                             ibis-impala
                             ibis-mssql
                             ibis-mysql
                             ibis-oracle
                             ibis-polars
                             ibis-postgres
                             ibis-pyspark
                             ibis-risingwave
                             ibis-snowflake
                             ibis-sqlite
                             ibis-trino
```



Is it faster than ...?

Ibis isn't a thing that can be fast by itself.

Is DuckDB faster than pandas? Yes.

Why is it called Ibis?