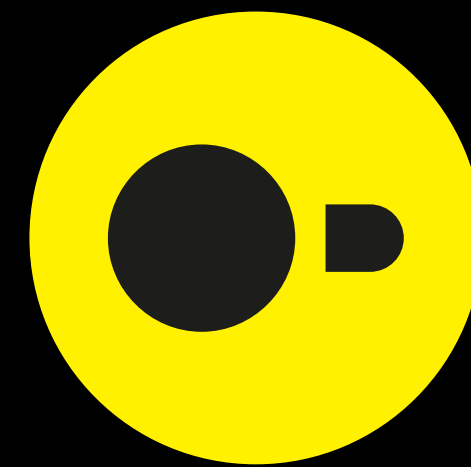
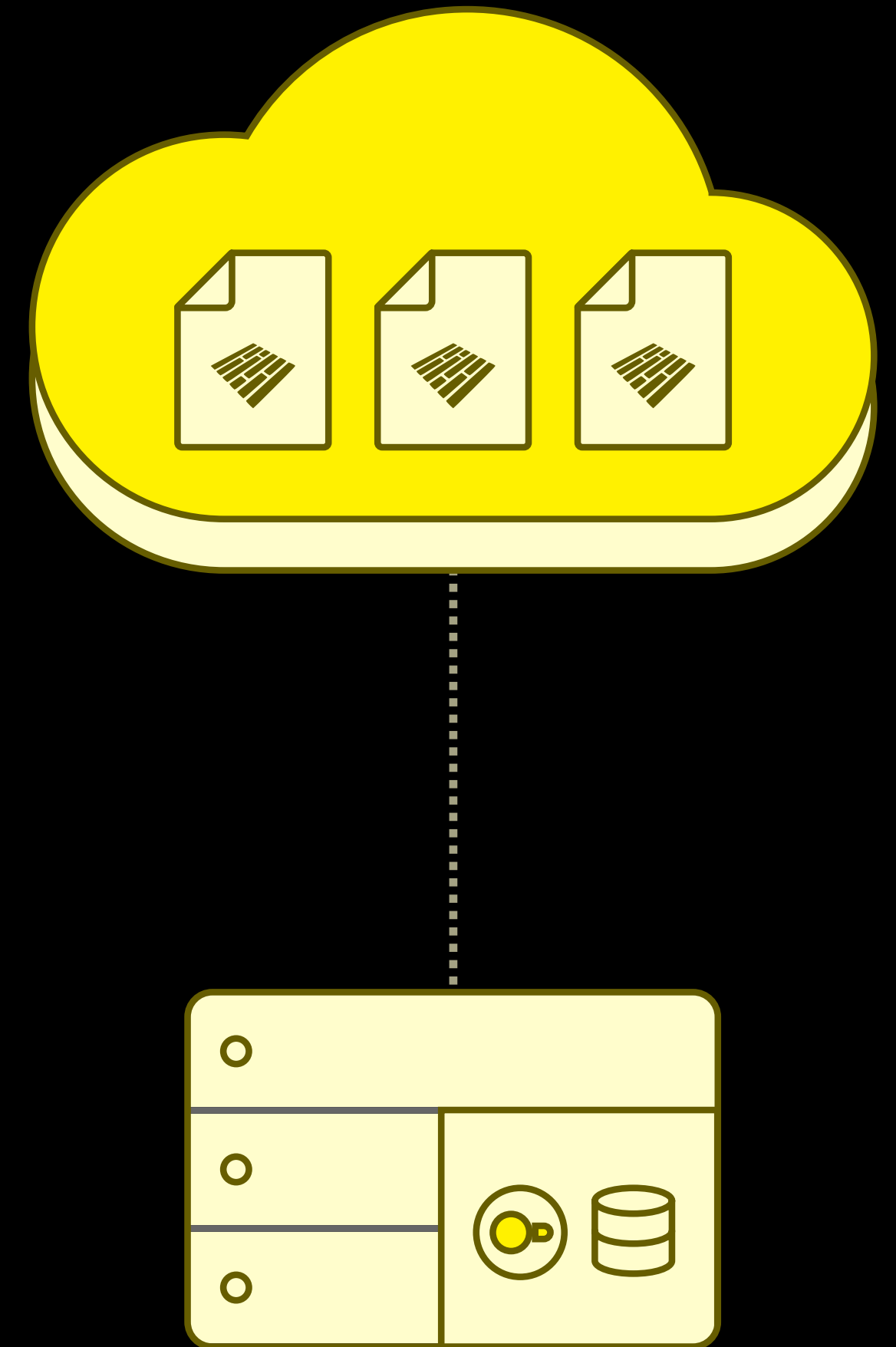
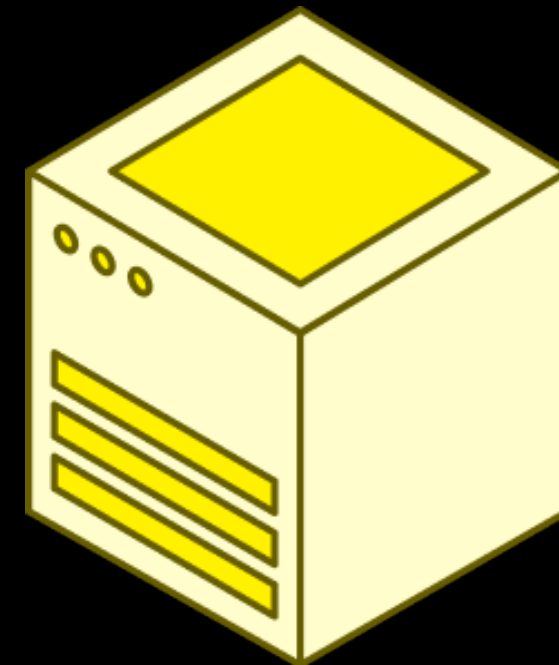


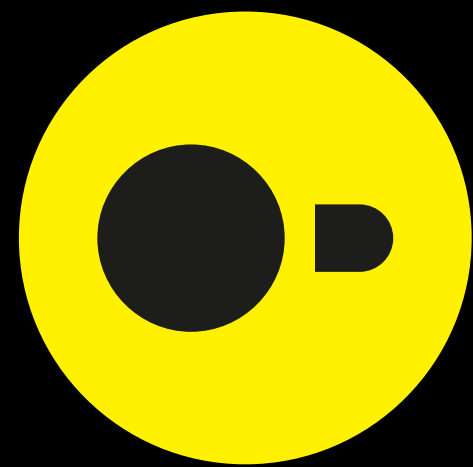
DuckDB Boston Meetup



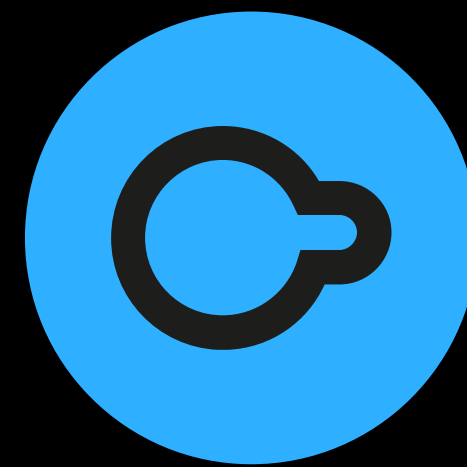
DuckDB

DuckDB: 2018–





DuckDB

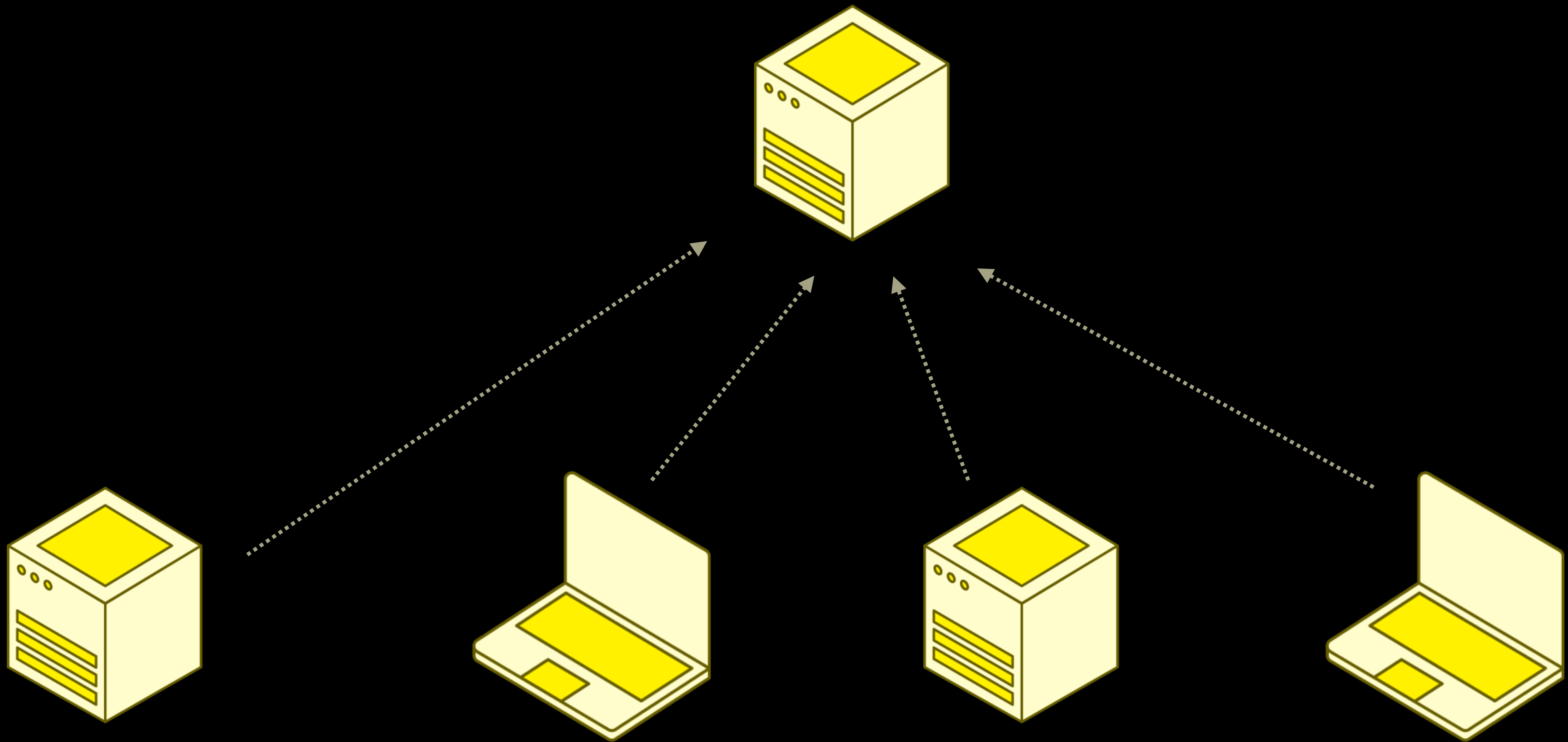


DuckLake



Quack

DuckDB: 2025-



Adoption trend in summer 2026

PyPI downloads /month

Extensions installed /month

Unique web visitors /month

JUN 2026



40M



160M



7M

SEP 2026



64M



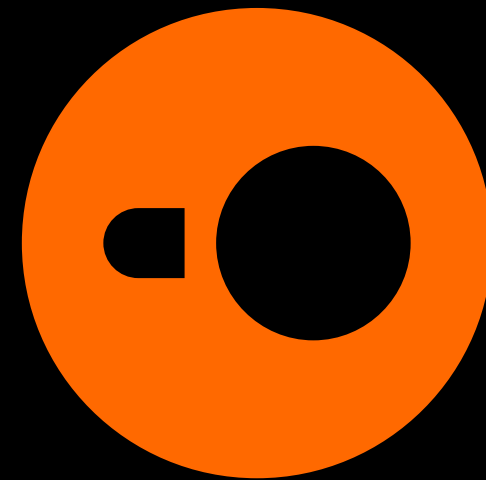
230M



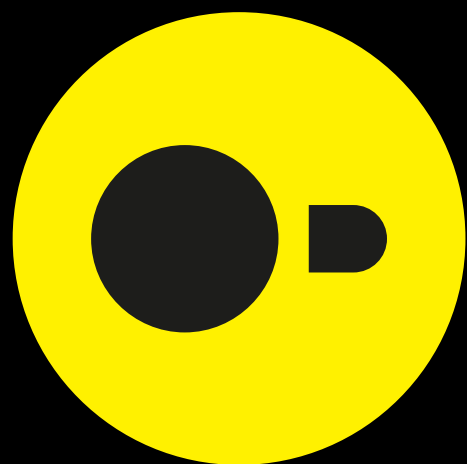
8M



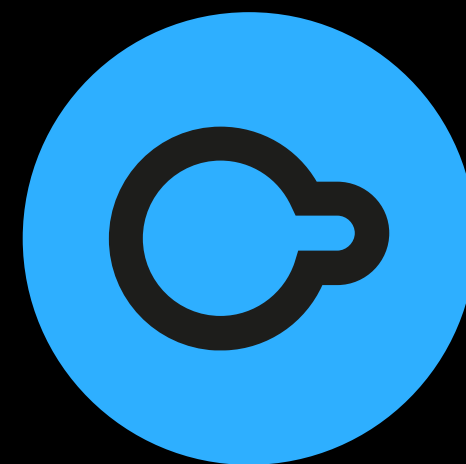
5M CE



DuckLabs



DuckDB



DuckLake



Quack



DuckLabs joins AWS

- IP stays in the DuckDB Foundation
- Projects stay open-source
- Team stays in Amsterdam
- Rapid development continues

More at re:Invent 2026

DuckDB

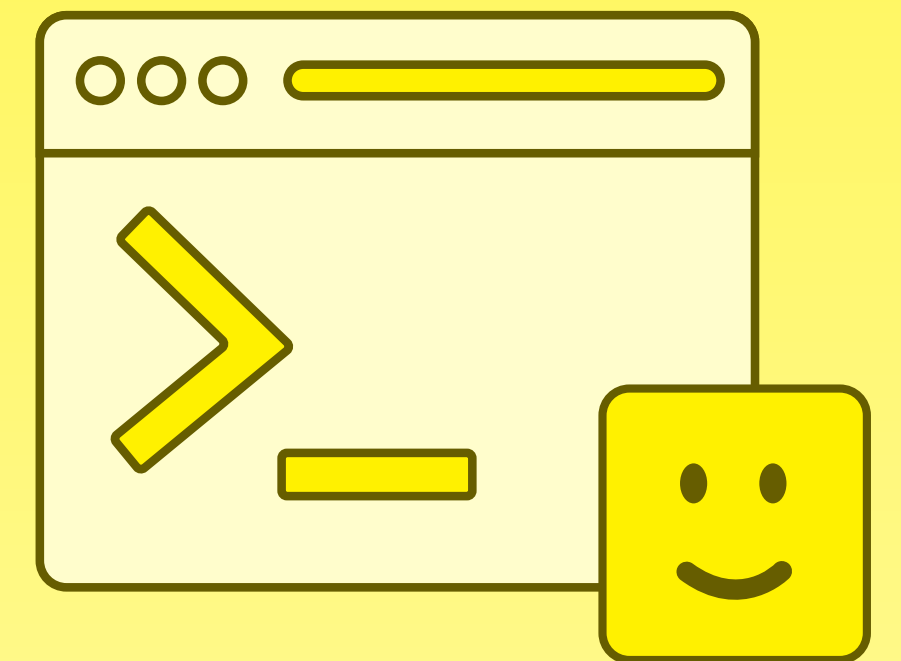
Not just a database but a ubiquitous dependency

- like glibc
- an SDK
- “connective tissue across the data estate”



01

SQL Features



VARIANT Type

```
CREATE TABLE variant_table (j VARIANT);  
INSERT INTO variant_table VALUES (JSON '{"hello":"world", "best_number": 42}');  
FROM variant_table;
```

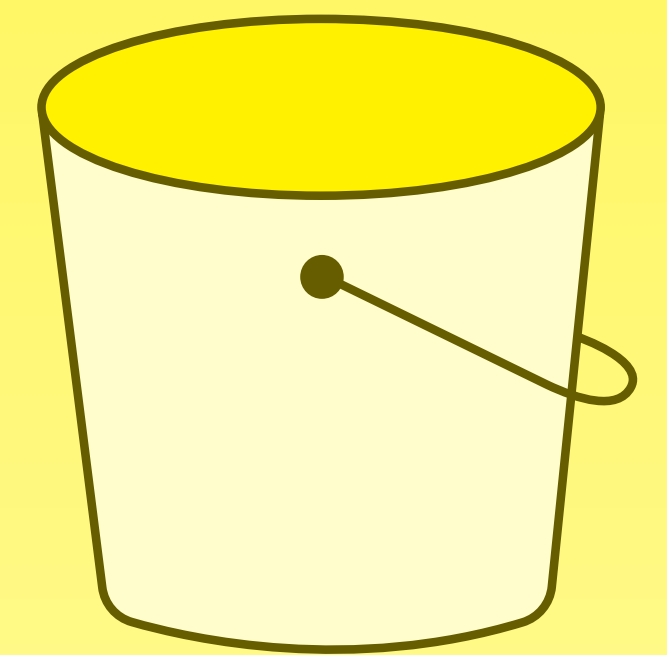
j variant
{'hello': world, 'best_number': 42}

Trigger

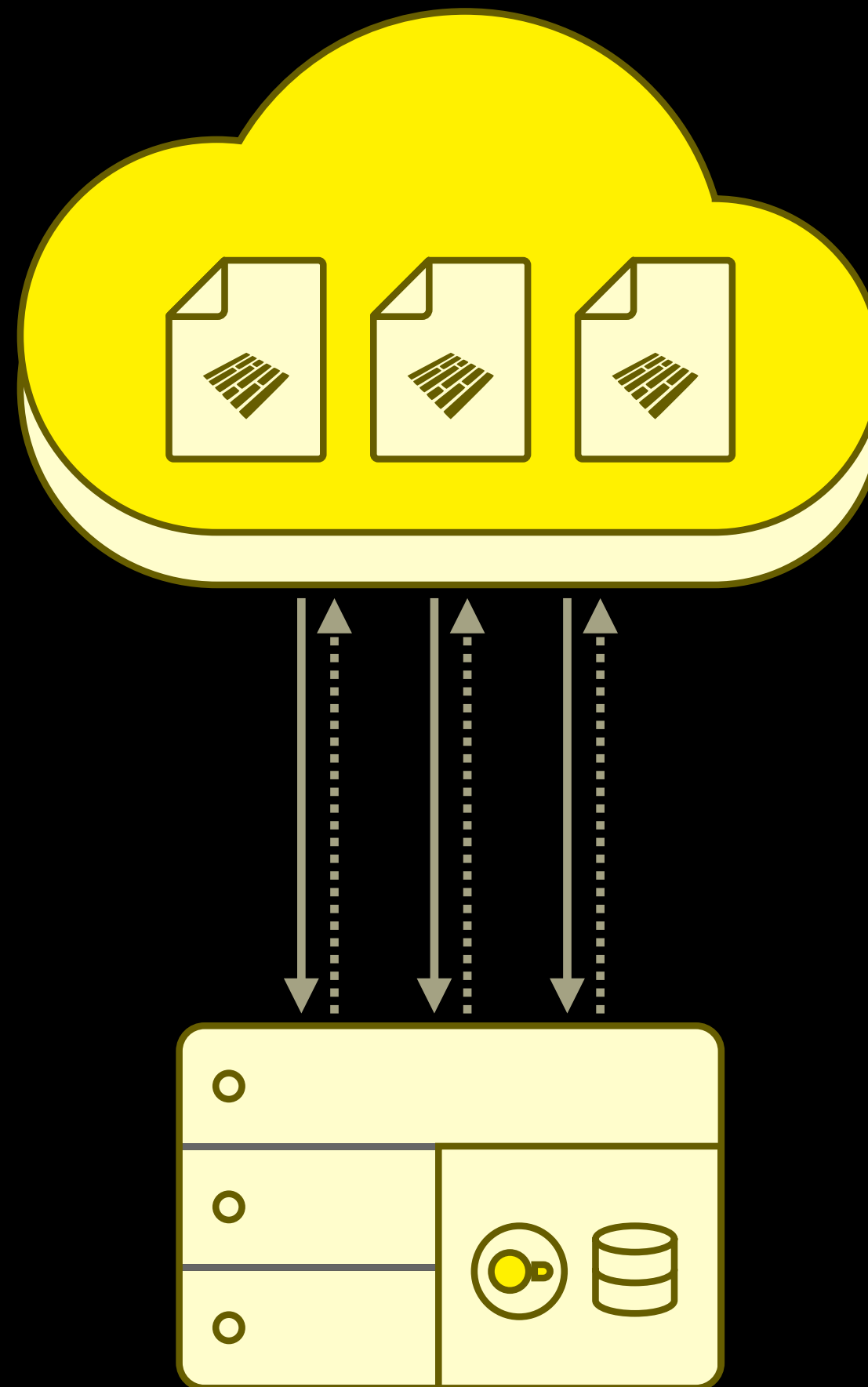
```
CREATE TABLE target (id INTEGER);  
CREATE TABLE audit  (id INTEGER);  
  
CREATE TRIGGER t  
    AFTER INSERT ON target  
    FOR EACH ROW INSERT INTO audit VALUES (NEW.id);
```


02

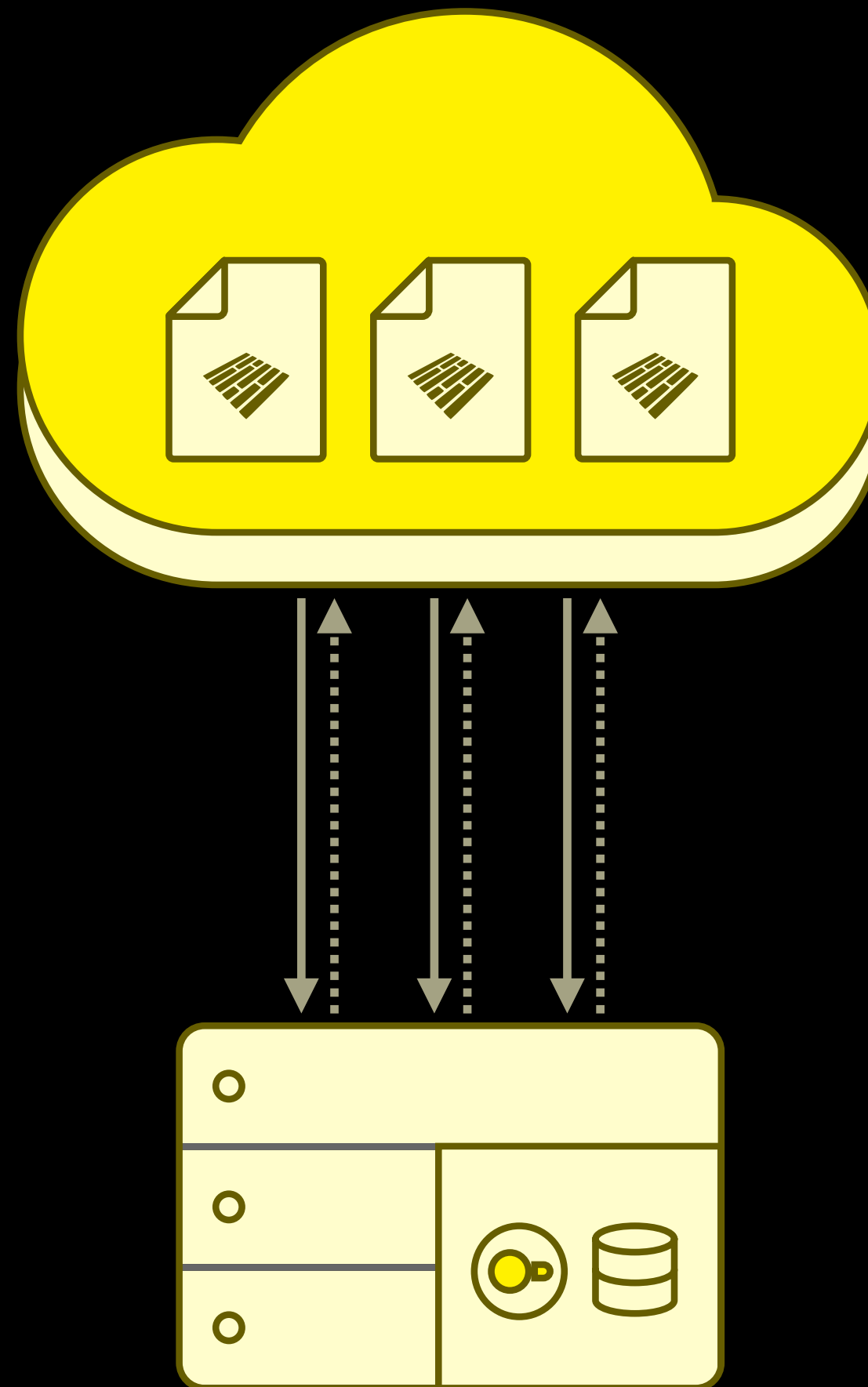
Object Stores



Sync I/O

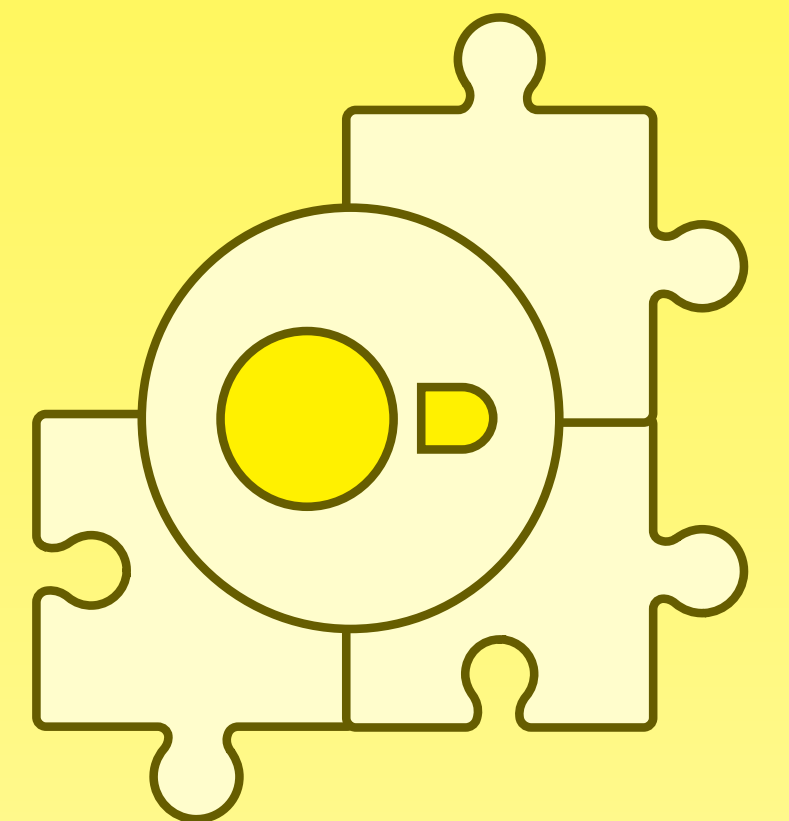


Async I/O

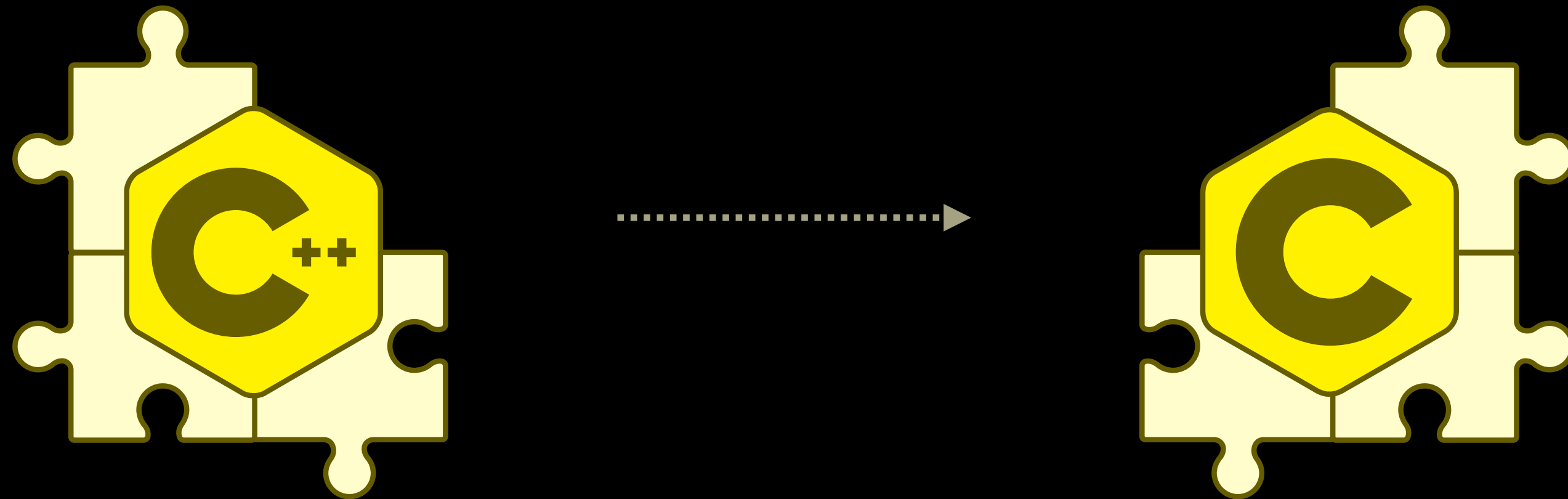


03

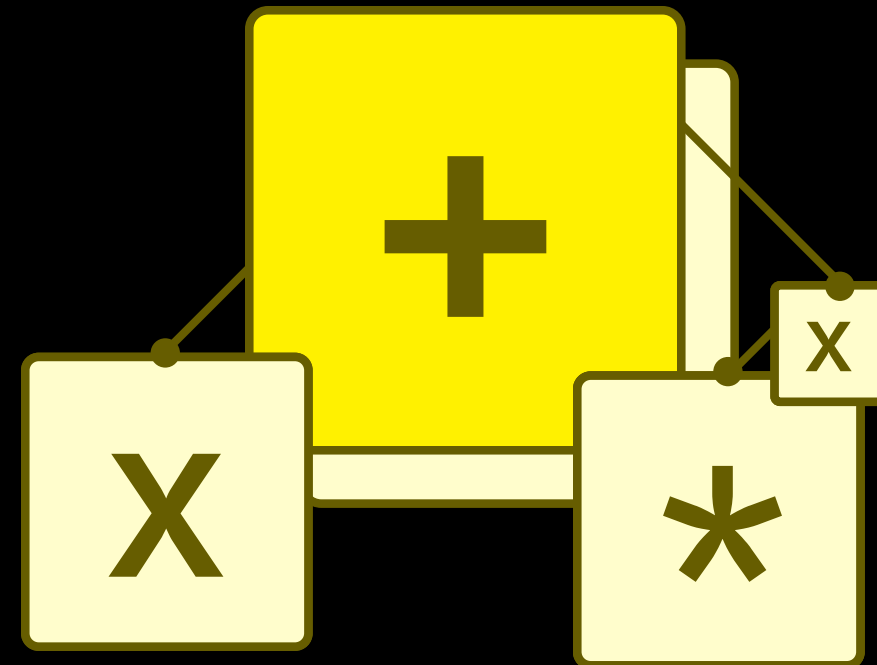
Extensions



C API v2 for Extensions



Extensible PEG parser



Learn SQL and DuckDB internals from U. Tübingen

Tabular Database Systems

07

Inner Join Follows Foreign Keys

vehicles						peeps			
vid	vehicle	kind	seats	wheels?	pid	pid	pic	name	born
v ₁		car	5	true	p ₄	p ₁		Cleo	2013
v ₂		SUV	3	true	p ₄	p ₂		Bert	1968
v ₃		bus	42	true	□	p ₃		Drew	□
v ₄		bus	7	true	□	p ₄		Alex	2002
v ₅		bike	1	true	p ₂				
v ₆		tank	□	false	p ₃				
v ₇		cabrio	2	true	p ₄				

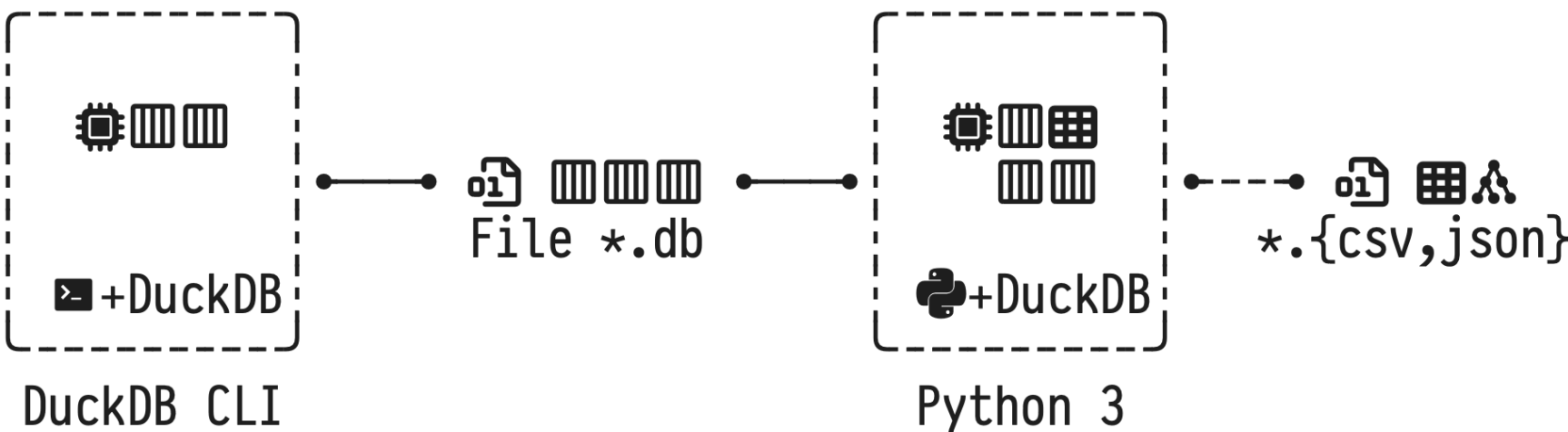
```
SELECT v.vehicle, p.name AS driver, p.pic
FROM vehicles AS v INNER JOIN peeps AS p ON (v.pid = p.pid)
                                "equi-join" ↗
```

- Some rows in table `peeps` find multiple *join partners* in `vehicles` (like `p4`) , some find none (`p1`).
- General: A join between `t1` and `t2` may yield $0 \dots |t_1| \times |t_2|$ rows.

Design and Implementation of DuckDB Internals

05

DuckDB: A Tabular DBMS Inside Your Own Process



- DuckDB kernel and client share a **single process** .
- Python: `import duckdb`, C/C++: link with `libduckdb`.
- Table data resides in a **single database file** `*.db`, native DuckDB data formats in file and in RAM are similar.
- DuckDB sees in-process client data and can **directly read/write client data structures** using SQL (“zero copy”, replacement scans).
- DuckDB allocates sizable RAM buffers (but can use disk for temporary storage if required).

Q & A

