Ducks Love Running in Circles

DuckDB Meetup on Science and Education

London, September 2025

Torsten Grust
Universität Tübingen, Germany

1 ┆ Recursive CTEs in DuckDB

Hannes	*Today we [...] are immensely proud to release a first preview of* DuckDB.
TG	Yay! \o/ Congratulations! But no WITH RECURSIVE?
Hannes	[CTEs] ... *Not recursive for now or never.*
TG	What? NOOOOOO!
Hannes	*Happy to review a PR* ^^

Back in June 2019

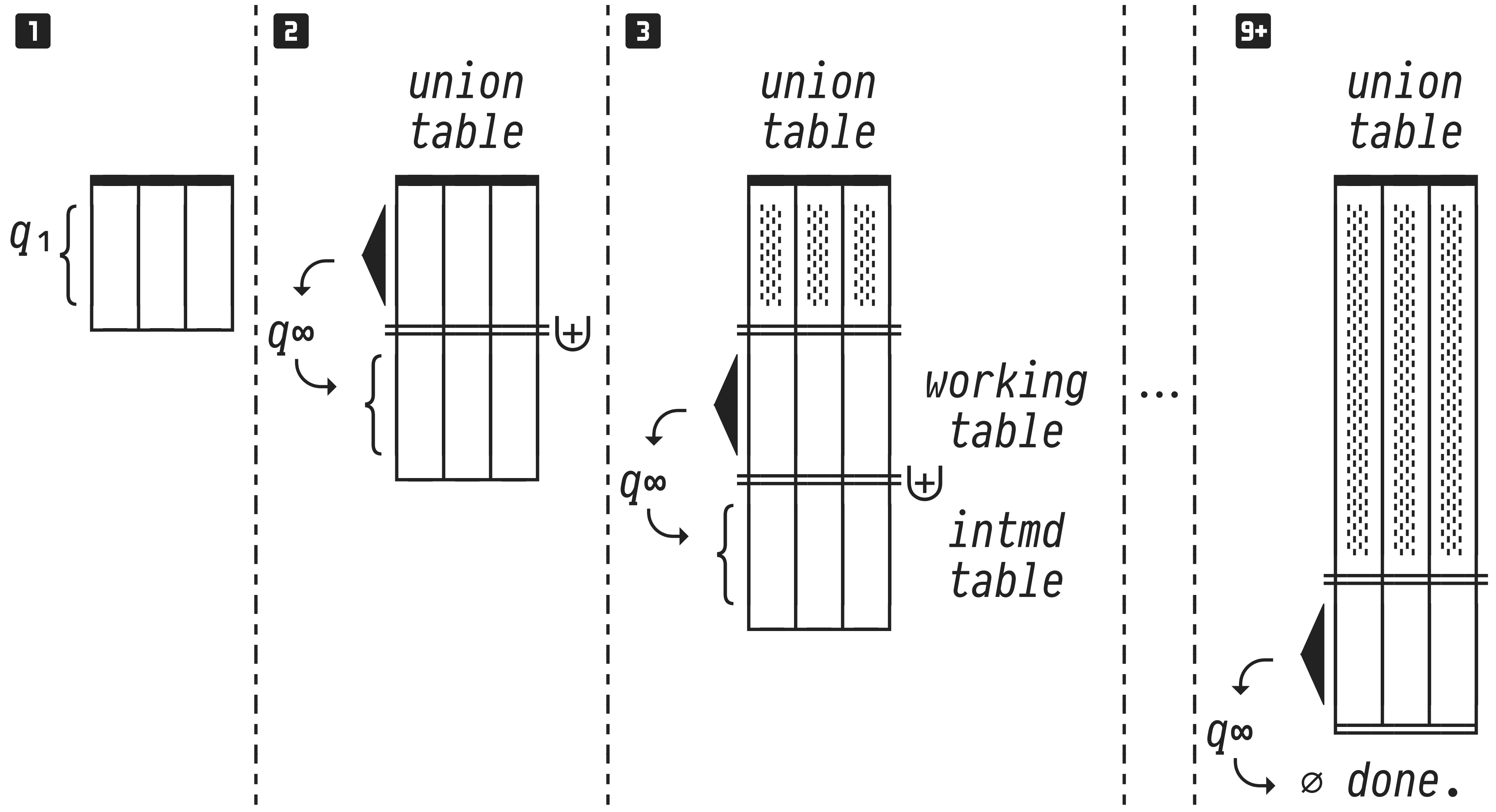
That's history. 😀

DuckDB now has become a versatile playground for **recursive CTEs:**

DuckDB version	has added ...
0.1.5 (early 2020)	**WITH RECURSIVE** (true to SQL:1999)
1.3.0 (May 2025)	**WITH RECURSIVE** ... **USING KEY**

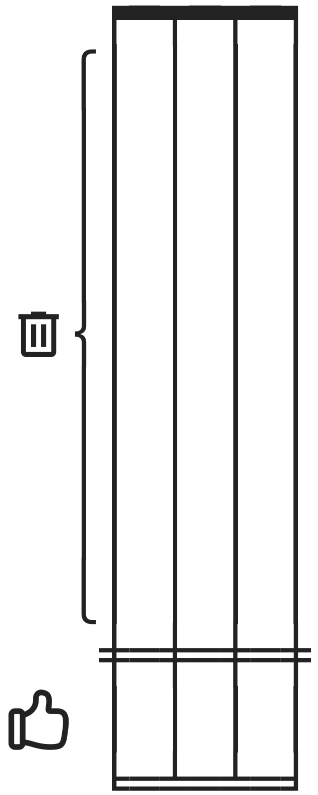
2 ┆ Recursive CTEs in SQL:1999: A Pile of Rows + Short-Term Memory

WITH RECURSIVE *t* **AS** (q_1 **UNION ALL** q_∞) **FROM** *t*



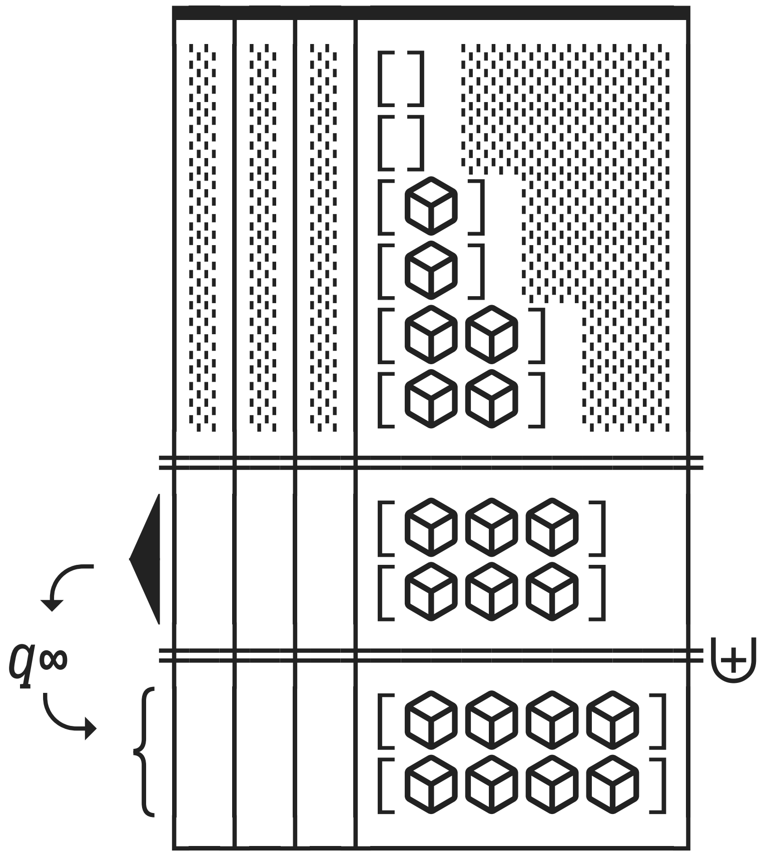
Recursive CTEs in SQL:1999: (Anti-)Idioms

final
union table



```
WITH RECURSIVE t AS (···)
FROM  t
WHERE ‹final iteration›
```

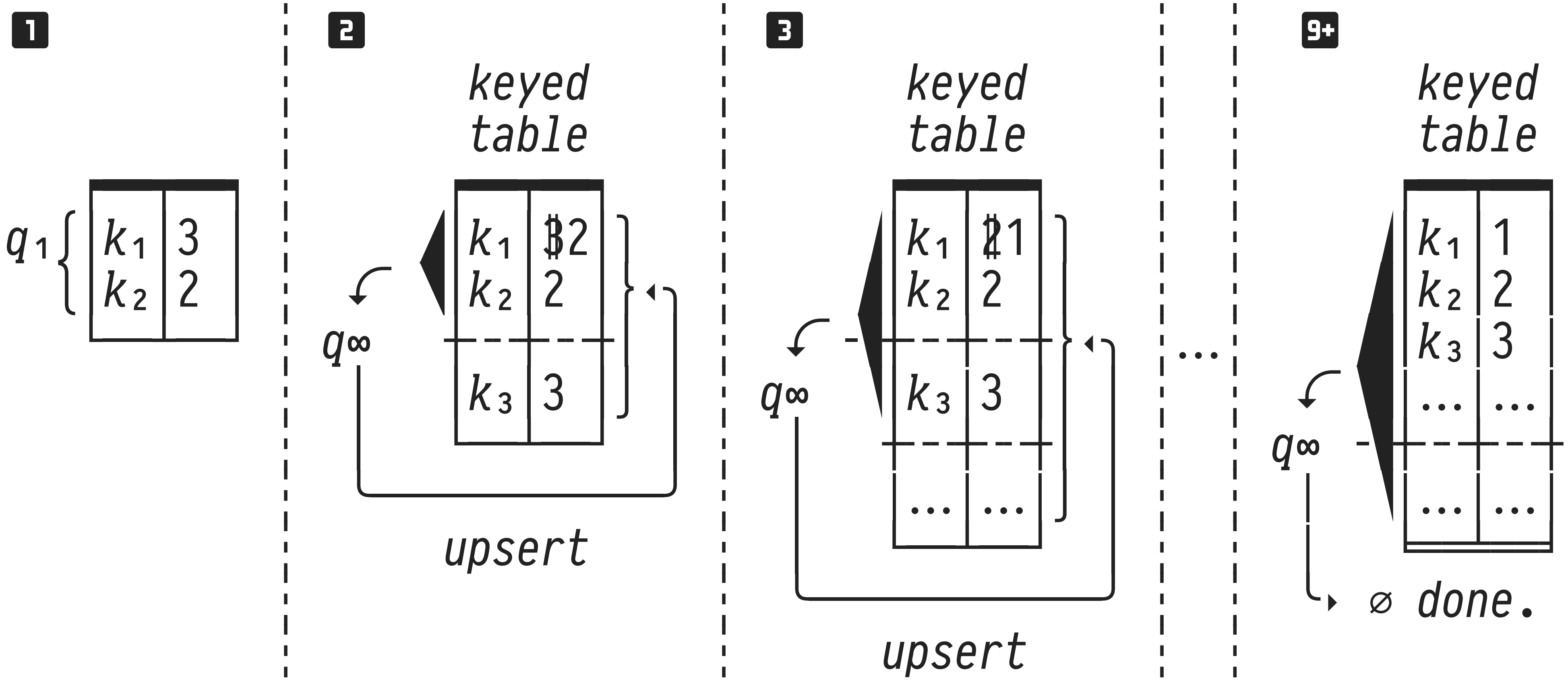
union table
+ ever-growing history



Aggregate relevant iteration history
to make up for "short-term" memory

3 ¦ Since DuckDB 1.3: USING KEY

WITH RECURSIVE *t*(*k*,…) ***USING KEY*** (*k*) **AS** (q_1 **UNION ALL** q_∞) **FROM** *t*



- Spread of keys k_i controls table size: can afford to pass entire keyed table to q_∞ (⋯ **FROM** *t*, recurring.*t* ⋯).

Since DuckDB 1.3: USING KEY

- May think of the keyed table like a dictionary or an **updatable associative array** (much like in your favourite PL).
- A wide range of iterative algorithms that **read and update intermediate state** naturally map to **USING KEY:**

Algorithm	**Key** *k*	**State**
K-Means Clustering	point ID	currently assigned cluster
Connected Components	node ID	current graph component
Distance Vector Routing	(*source*, *target*)	*via* node on currently best path

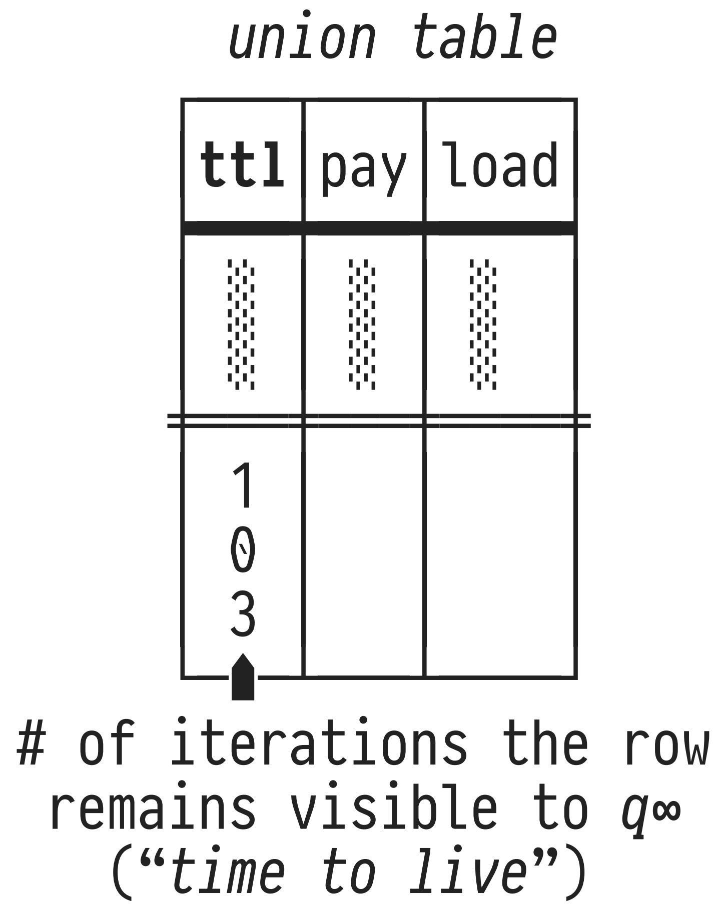
- But also …

Computation	**Key** *k*	**State**
execute PL/SQL-style UDF	variable ID	current variable contents

4 ┆ Future DuckDB: More Iterative SQL Queries

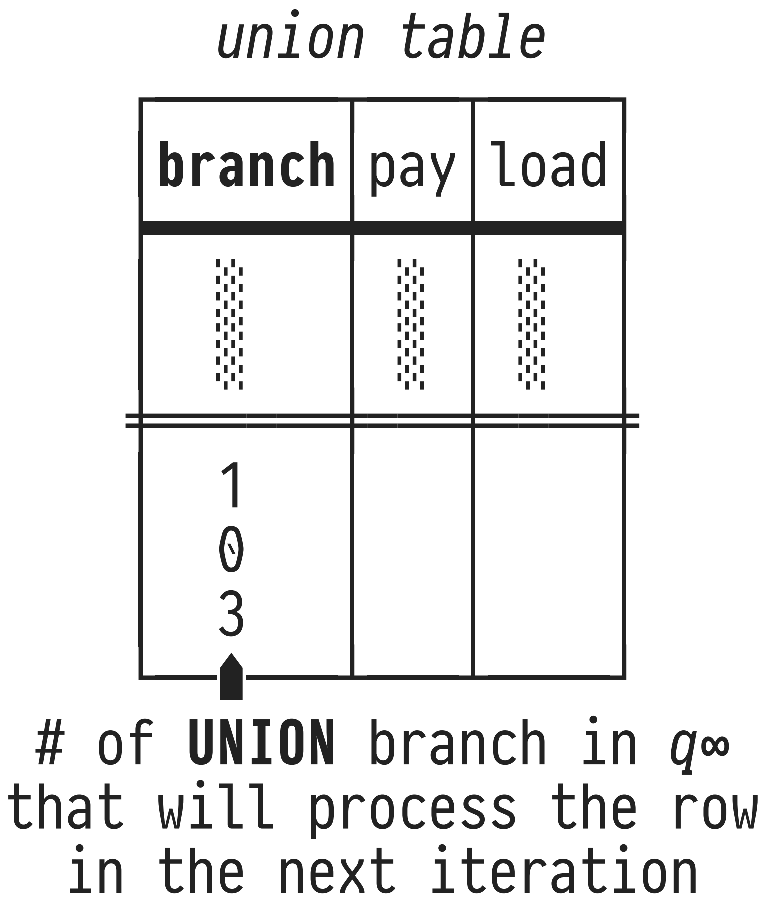
- In pipeline: generalize **USING KEY**, boost plan iteration.

Aging Row Memory



# of iterations the row remains visible to *q*∞ ("*time to live*")

Trampoline-Style Computation



of **UNION** branch in *q*∞ that will process the row in the next iteration

Ducks Love Running in Circles

Torsten Grust
Universität Tübingen, Germany

db.cs.uni-tuebingen.de
@teggy.org
Teggy