



Fast and simple way to replicate data from Postgres

Emrah Erdoğan

Database Architect @LCW

2025

What is PeerDB?

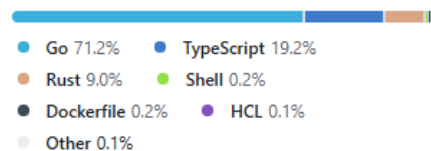
- Fast, simple and the most cost effective Postgres to Data Warehouses, Queues and Storage engines way to stream data from
- Founded by Kaushik Iska and Sai Srirampur in 2023 (San Francisco)
- Leading provider of CDC solutions for PostgreSQL
- ClickHouse acquires PeerDB in Jul 2024

Contributors 34



[+ 20 contributors](#)

Languages



Releases 141



[+ 140 releases](#)

☆ 2.7k stars

👁 20 watching

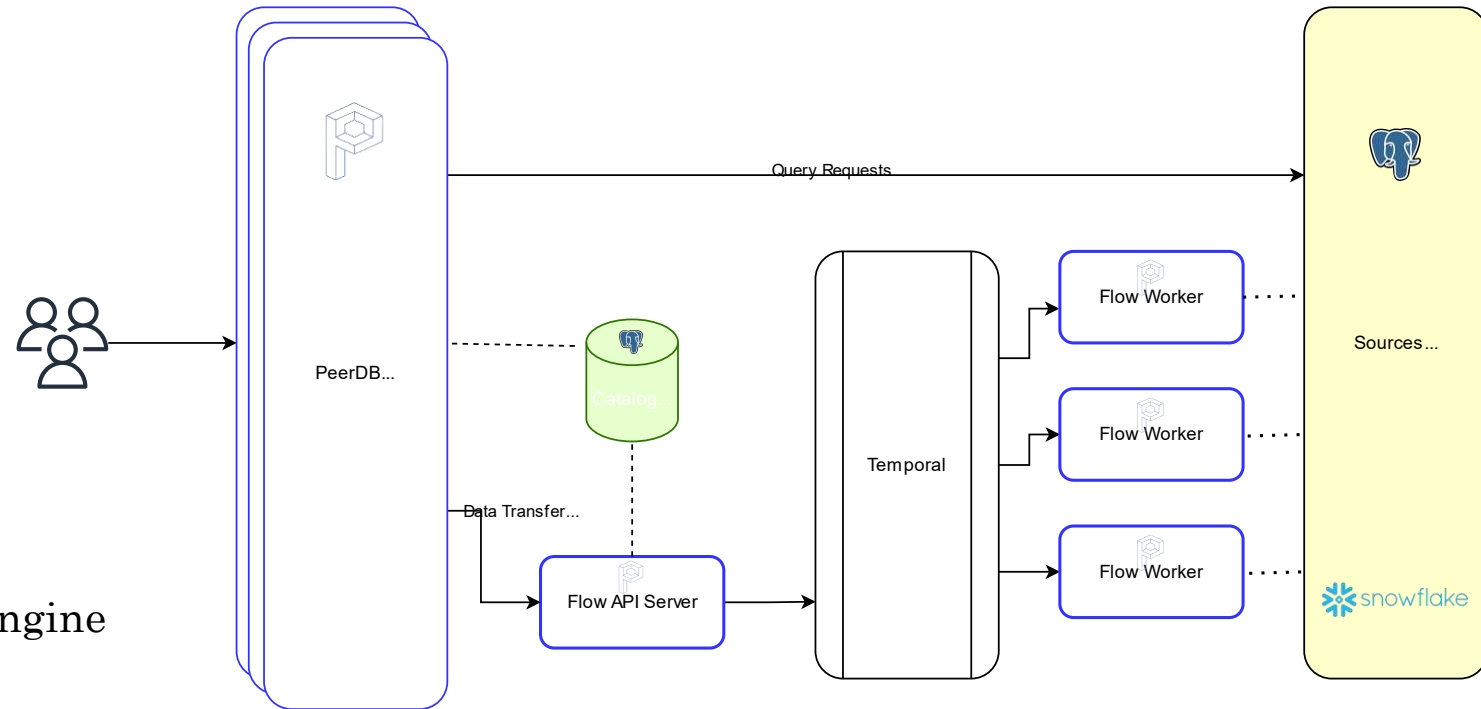
🔗 134 forks

Why PeerDB?

- Support different modes of streaming
 - Log based (CDC)
 - Cursor based (timestamp or integer)
 - XMIN
- 2x to 16x faster than other tools*
- Support native Postgres features
- Simple UI or a Powerful SQL Layer
- SQL / API / Lua (Queue) Scripting Support
- Free and Open
 - Elastic License 2.0 (ELv2)
- Postgres-compatible SQL interface to do ETL

Architecture

- Nexus Query Layer
- Flow Data Transfer
- Sources
- Sinks
- Internal Stage (MinIO:S3)
- Temporal.io Orchestration Engine
- Catalog Database



Text is not SVG - cannot display

Reference:

<https://docs.peerdb.io/architecture>

Postgres Native Features

- Support of advanced data types
 - ARRAYs, JSON/JSONB, HSTORE, ENUMs, Geospatial
- Comprehensive support of Partitioned Tables
 - Adding new partitions
 - Dropping partitions
 - Dropping partitions doesn't delete data on the target
 - Adding or dropping columns
 - Compatibility with different Postgres versions
- Efficient replication TOAST (large) columns
 - REPLICA IDENTITY FULL

Datatype Matrix - Primitive

Source	Destinations				
PostgreSQL	PostgreSQL	BigQuery	Snowflake	Clickhouse	ElasticSearch
smallint	smallint	INTEGER	INTEGER	Int16	long
integer	integer	INTEGER	INTEGER	Int32	long
bigint	bigint	INTEGER	INTEGER	Int64	long
float4	float4	FLOAT	FLOAT	Float32	float
double precision	double precision	FLOAT	FLOAT	Float64	float
boolean	bool	BOOLEAN	BOOLEAN	Bool	boolean
"char"	CHAR	STRING	STRING	FixedString(1)	text
varchar	varchar	STRING	STRING	String	text
date	date	DATE	DATE	Date	date
json	json	JSON	VARIANT	String	unnested subdocument
jsonb	jsonb	JSON	VARIANT	String	unnested subdocument
numeric	numeric	BIGNUMERIC	NUMBER	Decimal	text
text	text	STRING	STRING	String	text
timestamp	timestamp	TIMESTAMP	TIMESTAMP_NTZ	DateTime64(6)	date
timestamp with time zone	timestamp with time zone	TIMESTAMP	TIMESTAMP_TZ	DateTime64(6)	date

Datatype Matrix - Array

Source	Destinations			
PostgreSQL Type	PostgreSQL	BigQuery	Snowflake	Clickhouse
ARRAY<INT2>	ARRAY<INT2>	ARRAY<INT>	VARIANT	Array<Int16>
ARRAY<INT4>	ARRAY<INT4>	ARRAY<INT>	VARIANT	Array<Int32>
ARRAY<INT8>	ARRAY<INT8>	ARRAY<INT>	VARIANT	Array<Int64>
ARRAY<FLOAT4>	ARRAY<FLOAT4>	ARRAY<FLOAT>	VARIANT	Array<Float32>
ARRAY<DOUBLE PRECISION>	ARRAY<DOUBLE PRECISION>	ARRAY<DOUBLE PRECISION>	VARIANT	Array<Float64>
ARRAY<BOOL>	ARRAY<BOOL>	ARRAY<BOOL>	VARIANT	Array<Bool>
ARRAY<VARCHAR>	ARRAY<VARCHAR>	ARRAY<STRING>	VARIANT	Array<String>
ARRAY<TEXT>	ARRAY<TEXT>	ARRAY<STRING>	VARIANT	Array<String>
ARRAY<DATE>	ARRAY<DATE>	ARRAY<DATE>	VARIANT	String
ARRAY<TIMESTAMP>	ARRAY<TIMESTAMP>	ARRAY<TIMESTAMP>	VARIANT	String
ARRAY<TIMESTAMPZ>	ARRAY<TIMESTAMPZ>	ARRAY<TIMESTAMP>	VARIANT	String

Feature Matrix - Warehouses

Warehouses

Feature	Snowflake	BigQuery	Clickhouse	Postgres	S3/GCS
<u>Initial Snapshot - Parallelizable</u>	✓	✓	✓	✓	✓
Incremental Sync (CDC)	✓	✓	✓	✓	✓
DML Support - INSERT, UPDATE and DELETE	✓	✓	✓	✓	✓
<u>Schema Changes</u>	✓	✓	✓	✓	✓
Soft Delete	✓	✓	✖	✓	✓
<u>Partitioned Tables</u>	✓	✓	✓	✓	✓
Advanced Types - Arrays, JSONB, HSTORE, Geospatial etc.	✓	✓	✓	✓	✓
Advanced Monitoring - Lag, Throughput, Postgres Wait Events etc	✓	✓	✓	✓	✓
Resync Mirrors	✓	✓	✓	✓	✓
Column Exclusion	✓	✓	✓	✓	✓
Add Tables	✓	✓	✓	✓	✓

Feature Matrix - Queues

Queues

Feature	EventHubs	Kafka	Redpanda	PubSub
<u>Initial Snapshot - Parallelizable</u>				
Incremental Sync (CDC)				
DML Support - INSERT, UPDATE and DELETE		N/A	N/A	N/A
<u>Schema Changes</u>				
<u>Partitioned Tables</u>				
Advanced Types - Arrays, JSONB, HSTORE, Geospatial etc.				
Advanced Monitoring - Lag, Throughput, Postgres Wait Events etc				
Resync Mirrors		N/A	N/A	N/A
Column Exclusion				

Feature Matrix – Query/Watermark Based Replication

Feature	Snowflake	BigQuery	Postgres	S3/GCS
Initial Snapshot	✓	✓	✓	✓
Incremental Sync Based on Watermark	✓	✓	✓	✓
DML Support - INSERT, UPDATE (if updated_at & primary key are present)	✓	✓	✓	✓
Schema Changes - ADD COLUMN	✓	✗	✗	✗
Resync Mirrors	✓	✗	✗	✗
Partitioned Tables	✓	✓	✓	✓
Advanced Types - Arrays, JSONB, HSTORE, Geospatial etc.	✓	✓	✓	✓
Advanced Monitoring - Throughput, Postgres Wait Events etc	✓	✓	✓	✓
Column Exclusion	✓	✓	✓	✓

Feature Matrix - Transformations

Target	Status
EventHubs	✓
Kafka / Redpanda	✓
PubSub	✓
Snowflake	Private Preview
BigQuery	Private Preview
Clickhouse	Private Preview
Postgres	Private Preview
S3/GCS	Private Preview

Connectors/PostgreSQL

Source	Target	Real-time Change Data Capture (CDC)	Streaming Query or Watermark Based Replication	Guides
PostgreSQL	Snowflake	✓	✓	Change Data Capture, Streaming Query Replication
PostgreSQL	BigQuery	✓	✓	Change Data Capture, Streaming Query Replication
PostgreSQL	PostgreSQL	✓	✓	Change Data Capture, Streaming Query Replication
PostgreSQL	ClickHouse	✓	✓	Change Data Capture
PostgreSQL	Kafka	✓	N/A	Setup Kafka Peer
PostgreSQL	Redpanda	✓	N/A	Setup Redpanda Peer
PostgreSQL	Google PubSub	✓	N/A	Setup PubSub Peer
PostgreSQL	Azure EventHubs	✓	N/A	Change Data Capture
PostgreSQL	S3	✓	✓	Change Data Capture, Streaming Query Replication
PostgreSQL	GCS	✓	✓	Change Data Capture, Streaming Query Replication

Connectors – MySQL/MongoDB

MySQL	Snowflake	✓	✓
MySQL	BigQuery	✓	✓
MySQL	PostgreSQL	✓	✓
MySQL	ClickHouse	✓	✓
MySQL	Kafka	✓	N/A
MySQL	Redpanda	✓	N/A
MySQL	Google PubSub	✓	N/A
MySQL	Azure EventHubs	✓	N/A
MySQL	S3	✓	✓
MySQL	GCS	✓	✓
MongoDB	ClickHouse	✓	✓

Supported Postgres Providers

- RDS Postgres
- Supabase Postgres
- Google CloudSQL Postgres (Postgres 12+)
- Azure Flexible Server Postgres
- Crunchy Bridge Postgres
- Neon Postgres
- *Support Postgres versions 12 to 16*
- *Unsupported PostgreSQL Providers*
 - Heroku
 - DigitalOcean Managed Databases

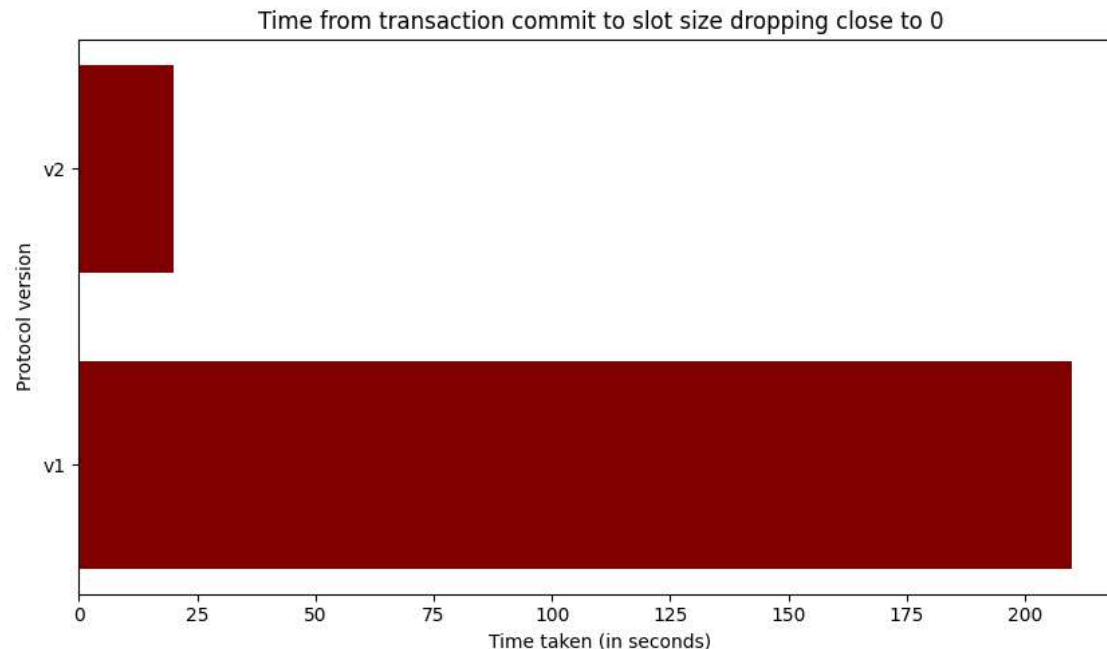
Important Notes

- Supporting DDL Changes
- TOAST columns need `REPLICA IDENTITY FULL`
 - large sized >8KB
- Logical decoding doesn't support generated columns
 - In-flight transformations
 - Extract, Load, and Transform (ELT)
- Logical replication slots don't persist on Postgres upgrades
 - Starting Postgres 17 replication slots are persisted on upgrades
- Logical replication slot don't persist failovers
 - Postgres 17 will support failover slots

Important Notes

- PostgreSQL 13 allows you to logically replicate partitioned tables
- Dropping partitions doesn't delete data on the target
- Useful Postgres configs
 - `max_slot_wal_keep_size`
 - `logical_decoding_work_mem`

proto_version



Protocol version. Currently versions 1, 2, 3, and 4 are supported.

Version 2 is supported only for server version 14 and above, and it allows streaming of large in-progress transactions.

Version 3 is supported only for server version 15 and above, and it allows streaming of two-phase commits.

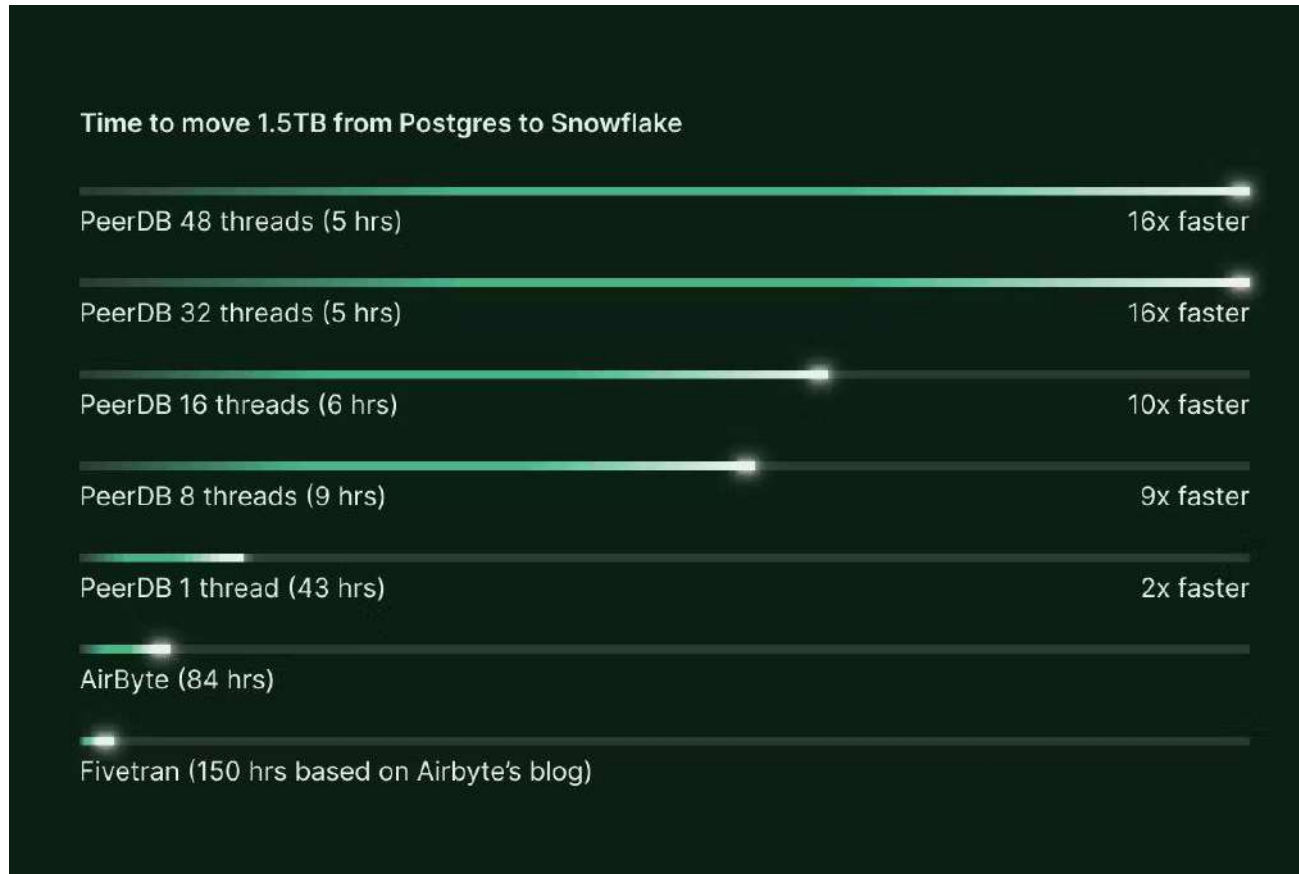
Version 4 is supported only for server version 16 and above, and it allows streams of large in-progress transactions to be applied in parallel.

Reference:

<https://blog.peerddb.io/exploring-versions-of-the-postgres-logical-replication-protocol>

<https://blog.peerddb.io/overcoming-pitfalls-of-postgres-logical-decoding>

Benchmark - PeerDB vs Airbyte



Reference:

<https://docs.peerdb.io/benchmarks/benchmarking-Postgres-Replication:-PeerDB-vs-Airbyte>

Cloud

- PeerDB Cloud
 - <https://app.peerdb.cloud>
- ClickHouse Cloud (ClickPipe)
 - <https://console.clickhouse.cloud>

Demo

- ClickHouse
- S3
- Kafka

```
git clone --recursive https://github.com/PeerDB-io/peerdb.git
```

```
cd peerdb
```

```
./run-peerdb.sh
```

PG2PG Demo | Prepare



Demo - Source

Select source

Start by selecting the data source for your new peer.

[? Learn about peers](#)

Data source

Select a source



Cancel

Continue

 BIGQUERY

 SNOWFLAKE

 POSTGRES

 S3

 CLICKHOUSE

Demo - Source

Setup a new Postgres peer

Name *	source	
Configuration		
Host *	catalog	
Port *	5432	
User *	postgres	
Password *		
Database *	source	
Transaction Snapshot		

SSH Configuration

You may provide SSH configuration to connect to your PostgreSQL database through SSH tunnel.

Configure SSH Tunnel ☐

[Back](#) [Validate](#) [Create](#)

Peer is valid

Demo - Target

Setup a new Postgres peer

Name *	target	
Configuration		
Host *	catalog	
Port *	5432	
User *	postgres	
Password *		
Database *	target	
Transaction Snapshot		

SSH Configuration

You may provide SSH configuration to connect to your PostgreSQL database through SSH tunnel.

Configure SSH Tunnel ☐

[Back](#) [Validate](#) [Create](#)

Peer is valid

Demo - Peers

← → ↻ 🏠 ⓘ localhost:3000/peers

 PeerDB

👤 Peers

↔ Mirrors

🔔 Alert Configuration

Peers

All peers

Peer Name	Peer Type
 source	 PostgreSQL
 target	 PostgreSQL

Demo - Mirrors



Demo - Mirrors

Create a new mirror

Set up a new mirror in a few easy steps.

Mirror type

☒ CDC

Change-data Capture or CDC refers to replication of changes on the source table to the target table with initial load. This is recommended.

[Learn more](#)

☐ Query Replication

Query Replication allows you to specify a set of rows to be synced via a SELECT query.

[Learn more](#)

☐ XMIN

XMIN mode uses the xmin system column of PostgreSQL as a watermark column for replication.

[Learn more](#)

Mirror Name

cdc_mirror

Source Peer *

source

Destination Peer *

target

Configuration

Initial Copy

☒

Sync Interval (Seconds)

60

Publication Name

Snapshot Number of Rows Per Partition

500000

Snapshot Maximum Parallel Workers

1

Snapshot Number of Tables In Parallel

4

Soft Delete

☐

Demo - Mirrors

Select tables to sync

Before selecting tables, please make sure that [these permissions](#) have been granted for your tables.

🔍 Search for schemas

▶ public

⚙️ Validate

+ Create Mirror

Demo - Mirrors

Select tables to sync

Before selecting tables, please make sure that [these permissions](#) have been granted for your tables.

Search for schemas

▼ public ☐ Select All Search for tables

☒ public.test

Columns

- ☒ id integer
- ☒ c1 integer
- ☒ c2 integer
- ☒ t text

Target Table:
public.test

☒ Validate



Demo - Mirrors

Select tables to sync

Before selecting tables, please make sure that [these permissions](#) have been granted for your tables.

Search for schemas

public ☐ Select All Search for tables

☒ public.test

Columns

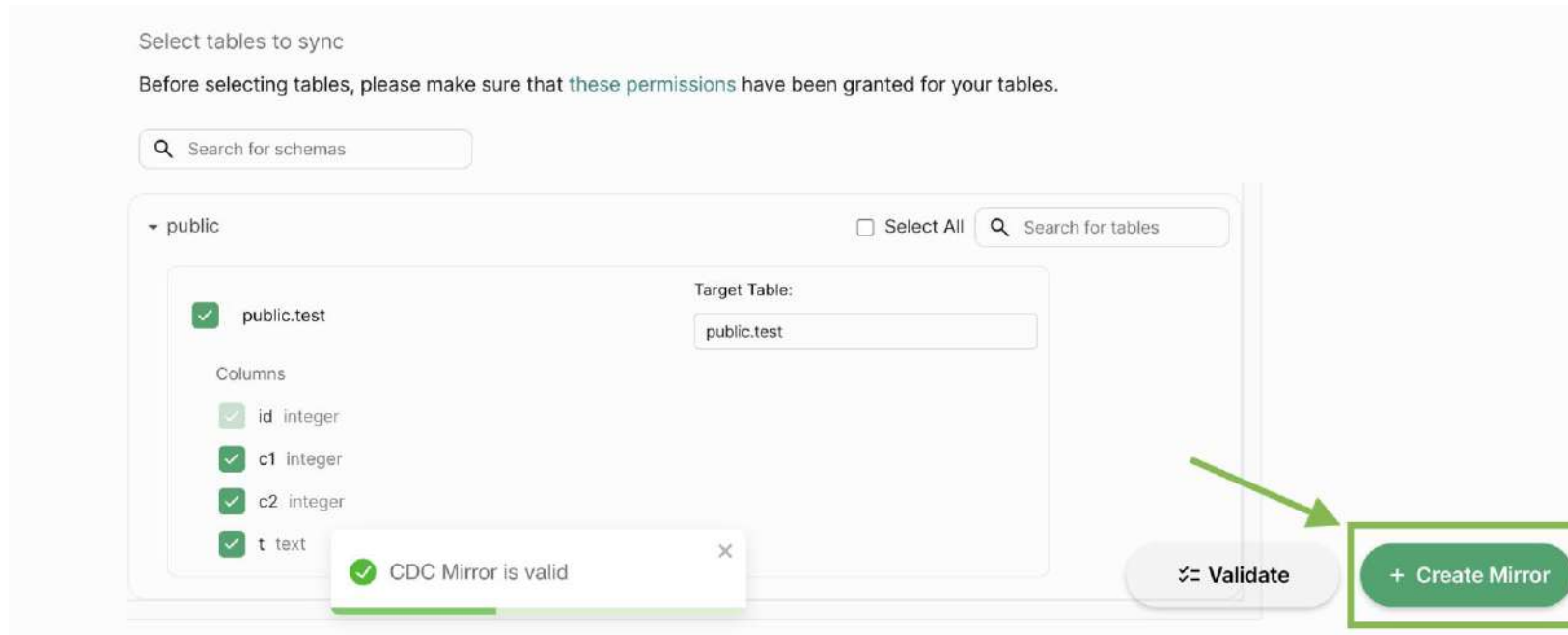
- ☒ id integer
- ☒ c1 integer
- ☒ c2 integer
- ☒ t text

Target Table:
public.test

☒ CDC Mirror is valid

Validate

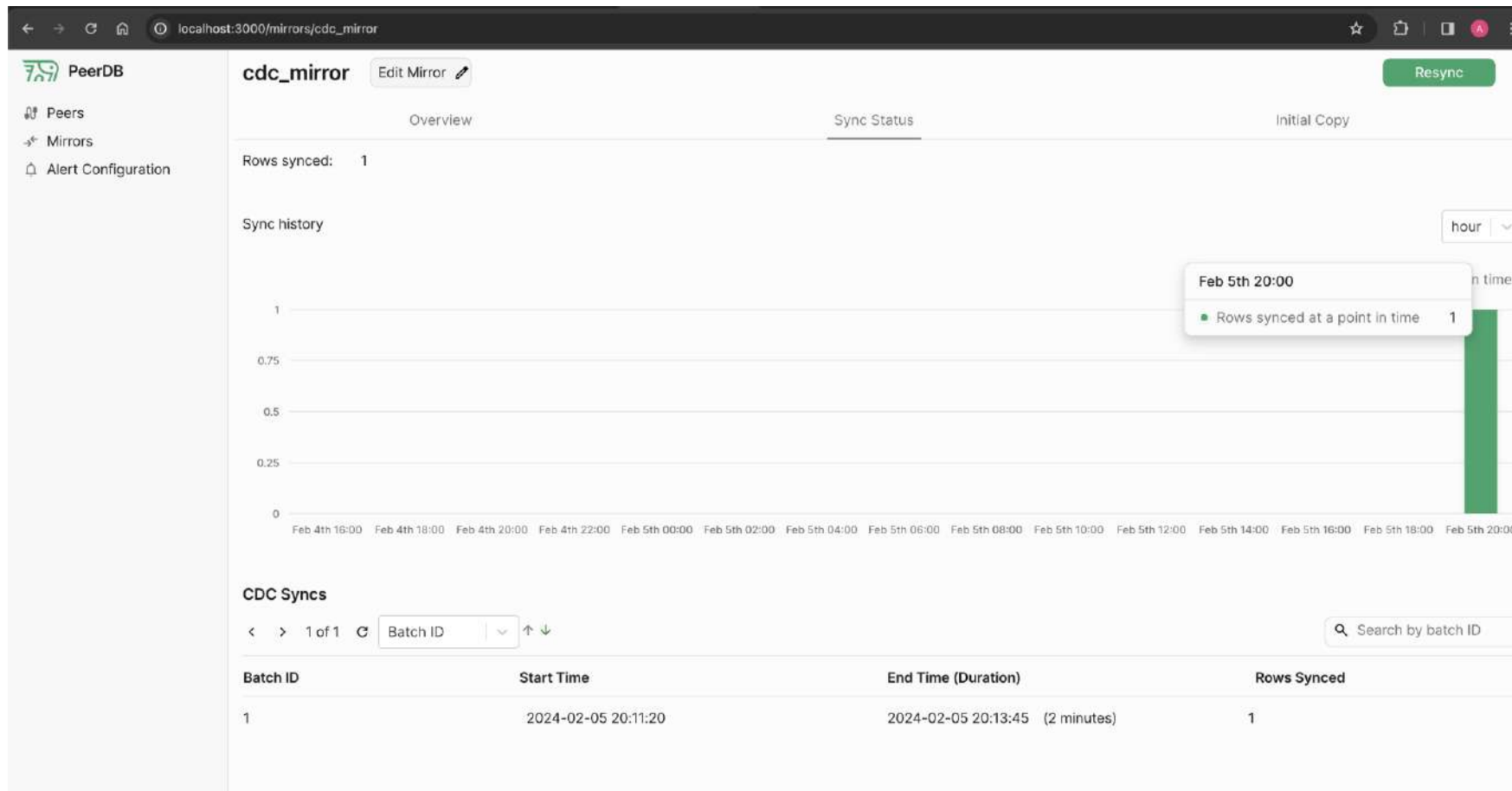
+ Create Mirror



Demo - Mirrors

Mirrors					
Change-data capture					
+ New mirror					
Search by flow name					
Name	Source	Destination	Start Time	Logs	
<u>cdc_mirror</u>	 source	 target	2024-02-05 20:11:20		

Demo - Mirrors



Demo – Mirrors ClickHouse

Setup a Clickhouse peer

We recommend going through our [setup guide for Clickhouse destinations](#).

Name *	clickhouse_devcls_n1	
Configuration		
Host *	chdev1	
Port *	9000	
User *	default	
Password		
Database *	uk	
Cluster		
Replicated?	☐	
Disable TLS?	☒	
Certificate	Choose File No file chosen	
Private Key	Choose File No file chosen	
Root Certificate	Choose File No file chosen	

Transient S3 Stage (Optional)

PeerDB loads rows as files in an internal staging environment during CDC, under the covers.
If you want this stage to belong to you, you can configure a bucket below.

[Setting up an S3 bucket](#)

Configure a stage

[Back](#) [Validate](#) [Create peer](#)

✓ Peer is valid

Demo – Mirrors ClickHouse

Create a new mirror

Set up a new mirror in a few easy steps.

Mirror type



CDC

Change-data Capture or CDC refers to replication of changes on the source table to the target table with initial load. This is recommended.

[Learn more](#)



Query Replication

Query Replication allows you to specify a set of rows to be synced via a SELECT query. Useful for replicating views and tables without primary keys.

[Learn more](#)



XMin

XMIN mode uses the xmin system column of PostgreSQL as a watermark column for replication.

[Learn more](#)

Mirror Name

mirror_pg2ch_house_sales

Source Peer *



pgdev1



Destination Peer *



chdev1



Configuration

Initial Copy *



Sync Interval (Seconds) *

10



Publication Name

peerdb_publication



Parallelism for Initial Load *

4



Demo – Mirrors ClickHouse

Advanced Settings ⌄

Pull Batch Size	250000	
Snapshot Number of Rows Per Partition	250000	
Snapshot Number of Partitions Override	0	
Snapshot Number of Tables In Parallel	1	
Initial Copy Only	☒	
Script		
Disable all PeerDB columns (overrides any other setting)	☒	
Settings override		

Demo – Mirrors ClickHouse

Select tables to sync

Before selecting tables, please make sure that [these permissions](#) have been granted for your tables.

🔍 Search for schemas

▼ public

☐ Select All

🔍 Search for

☒ public.house_sales 4203 MB

Target Table:

ch_house_sales

Custom Partitioning Key:

Enter optional custom partition key

Engine:

ReplacingMergeTree

Sharding Key:

Sharding key expression (optional)

Policy Name:

Policy name (optional)

Partition By Expr:

Partition By expression (optional)

Columns

- ☒ addr1 character varying(100)
- ☒ addr2 character varying(100)
- ☒ county character varying(35)
- ☒ date date
- ☒ district character varying(40)
- ☒ duration character varying(9)
- ☒ id Integer
- ☒ is_new smallint
- ☒ locality character varying(35)
- ☒ postcode1 character varying(8)
- ☒ postcode2 character varying(3)
- ☒ price Integer
- ☒ street character varying(60)
- ☒ town character varying(35)
- ☒ type character varying(13)

☐ Use a custom sorting key

☐ Partition ClickHouse table

Demo – Mirrors ClickHouse

mirror_pg2ch_housesales

Actions

Overview

Sync Status

Initial Copy

In progress

< > 1 of 1

Start Time

↑ ↓

Search by table name

Table Identifier	Status	Sync Start	Progress Partitions	Num Rows Processed	Avg Time Per Partition
house_sales_resync	⬆️ Syncing	2025-10-20 19:08:29	16 / 111	3997836	7 seconds

Completed tables

< > 1 of 0

Start Time

↑ ↓

Search by table name

Table Identifier	Status	Sync Start	Progress Partitions	Num Rows Processed	Avg Time Per Partition
------------------	--------	------------	---------------------	--------------------	------------------------

Demo – Mirrors ClickHouse

mirror_pg2ch_housesales

Actions

Overview

Sync Status

Initial Copy

In progress

< > 1 of 0

Start Time

↑ ↓

Search by table name

Table Identifier	Status	Sync Start	Progress Partitions	Num Rows Processed	Avg Time Per Partition
------------------	--------	------------	---------------------	--------------------	------------------------

Completed tables

< > 1 of 1

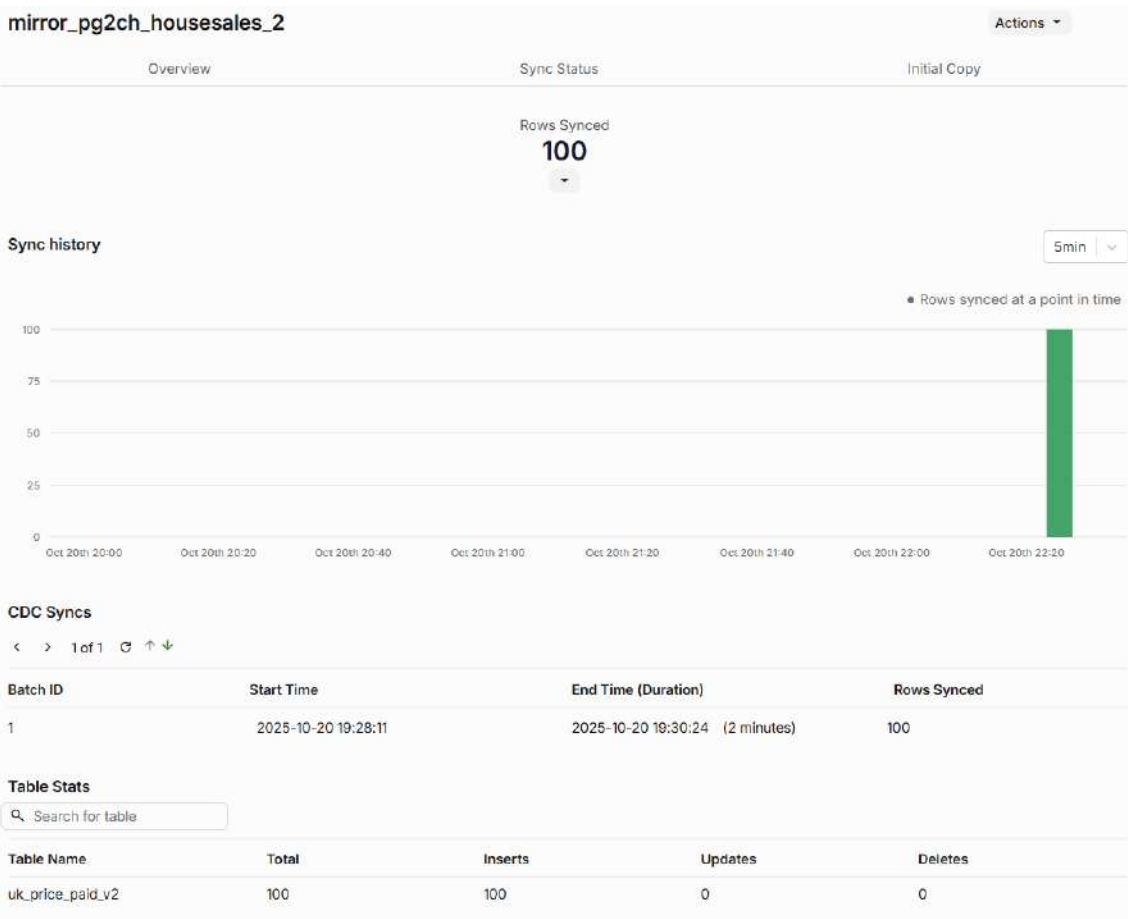
Start Time

↑ ↓

Search by table name

Table Identifier	Status	Sync Start	Progress Partitions	Num Rows Processed	Avg Time Per Partition
house_sales_resync	✓ Done	2025-10-20 19:08:29	111 / 111	27734966	8 seconds

Demo – Mirrors ClickHouse



Demo – Temporal Workflow

default

19 Workflows

7 Running

7 Completed

4 Continued as New

1 Canceled

Filter

	Status	Workflow ID	Run ID	Type
☐	Running	mirror_pg2ch_housesales-peerflow-3f5d3ffd-11f6-497d-b1db-8c9f040a558c	0b51ccdd-68a3-4cb9-ae07-2351142ae23f	CDCFlowWorkflow
☐	Running	mirror_pg2s_test-peerflow-1e736112-31b7-4444-a791-bf6eed5fc4ca	b715b236-911e-4464-9f04-857774587ce6	CDCFlowWorkflow
☐	Running	pg2clickhouse-peerflow-05fe65fa-c143-45e2-a410-de8541ea121c	fd7da2f1-2377-4f35-8a0e-6ca98e52d775	CDCFlowWorkflow
☐	Running	scheduler-1dfd4f63-c02c-4ade-9137-08e98bce8355	019a02c9-868a-7335-89cc-811e911ad66d	ScheduledTasksWorkflow
☐	Running	mng2ch_reviews-peerflow-213e7e1a-9386-4be5-929d-8c521f7b10b5	416983f4-eb1a-44b5-b681-3c652f367c59	CDCFlowWorkflow
☐	Running	mng2ch_calendar-peerflow-43092749-2695-4917-aa96-5a7bb673f1c7	ce6c617e-e4ec-4459-bc49-b5edc1b74db9	CDCFlowWorkflow
☐	Running	mirror_mngtest-peerflow-bd47dfa0-d44e-4720-abf5-ab7497ff9a27	b9de9df6-7989-4e8f-9ac2-44b8c5aa1ee3	CDCFlowWorkflow
☐	Continued as New	mirror_pg2ch_housesales-peerflow-3f5d3ffd-11f6-497d-b1db-8c9f040a558c	019a02db-1959-76dd-833b-7a4343a356af	CDCFlowWorkflow
☐	Completed	snapshot-flow-mirror_pg2ch_housesales-019a02db-1959-76dd-833b-7a4343a356af	019a02db-1bd8-7512-8df0-f9bde142f8f9	SnapshotFlowWorkflow
☐	Completed	clone_mirror_pg2ch_housesales_public_house_sales-019a02db-1bd8-7512-8df0-f9bde142f8f9	7f9b3c64-b407-4dd9-b333-59a38d4bf1d6	QRepFlowWorkflow
☐	Continued as New	clone_mirror_pg2ch_housesales_public_house_sales-019a02db-1bd8-7512-8df0-f9bde142f8f9	019a02db-1d83-714d-a45a-a36670887adf	QRepFlowWorkflow
☐	Completed	qrep-part-clone_mirror_pg2ch_housesales_public_house_sales-019a02db-1bd8-7512-8df0-f9bde142f8f9-b4a96019-8643-4386-8275-c3e950726c21	019a02db-5afc-7a0d-82af-307b41fac91a	QRepPartitionWorkflow
☐	Completed	qrep-part-clone_mirror_pg2ch_housesales_public_house_sales-019a02db-1bd8-7512-8df0-f9bde142f8f9-3f8ac53a-5012-46ff-93e9-dbf9270cdc05	019a02db-5b15-7682-8e1d-2b7b5e4a26f6	QRepPartitionWorkflow
☐	Completed	qrep-part-clone_mirror_pg2ch_housesales_public_house_sales-019a02db-1bd8-7512-8df0-f9bde142f8f9-579ae36d-787a-4834-800e-191417abb971	019a02db-5b2b-7481-860e-1564f2188cfe	QRepPartitionWorkflow
☐	Completed	qrep-part-clone_mirror_pg2ch_housesales_public_house_sales-019a02db-1bd8-7512-8df0-f9bde142f8f9-5e0075ca-9f78-4722-aa54-7c1614ef768a	019a02db-5ae0-7675-88ae-505dc2e2bde8	QRepPartitionWorkflow
☐	Completed	setup-flow-mirror_pg2ch_housesales-019a02db-1959-76dd-833b-7a4343a356af	019a02db-19ec-7c87-b0e5-91a224e8274a	SetupFlowWorkflow
☐	Continued as New	mirror_pg2s_test-peerflow-1e736112-31b7-4444-a791-bf6eed5fc4ca	1dab0b1a-ed31-41fa-b6d4-62d0ccb4d16e	CDCFlowWorkflow
☐	Continued as New	pg2clickhouse-peerflow-05fe65fa-c143-45e2-a410-de8541ea121c	ca4c449d-18f6-4fe0-afac-4b726d68a218	CDCFlowWorkflow
☐	Canceled	scheduler-0a26b8a6-6c1a-4628-9b0d-005e589c5f81	0199f37e-ea0e-79a5-9973-7c53eeec3827	ScheduledTasksWorkflow

100

Demo – Temporal Workflow



Event ID	Timestamp	Event Type	Details
23	2025-10-20 UTC 18:50:31.43	Workflow Execution Continued As New	Workflow Type Name: CDCFlowWorkflow
22	2025-10-20 UTC 18:50:31.43	Workflow Task Completed	Identity: 1@1e567e1a2426@
21	2025-10-20 UTC 18:50:31.43	Workflow Task Started	History Size Bytes: 5974
20	2025-10-20 UTC 18:50:31.42	Workflow Task Scheduled	Task Queue Name: 1e567e1a2426:45f984a1-4e31-4a7a-8828-4d10d07f5492
19	2025-10-20 UTC 18:50:31.42	Timer Fired	Timer ID: 18
18	2025-10-20 UTC 18:49:31.42	Timer Started	Start To Fire Timeout: 1 minute Summary: Sleep
17	2025-10-20 UTC 18:49:31.42	Workflow Task Completed	Identity: 1@1e567e1a2426@
16	2025-10-20 UTC 18:49:31.41	Workflow Task Started	History Size Bytes: 5443
15	2025-10-20 UTC 18:49:31.40	Workflow Task Scheduled	Task Queue Name: 1e567e1a2426:45f984a1-4e31-4a7a-8828-4d10d07f5492
14	2025-10-20 UTC 18:49:31.40	Activity Task Failed	Failure Message: failed in pull records when: ReceiveMessage failed: receive message failed: ...
13	2025-10-20 UTC 18:23:58.07	Activity Task Started	Attempt: 1
12	2025-10-20 UTC 18:23:58.05	Local Activity	Activity Type: updateFlowStatusInCatalogActivity
11	2025-10-20 UTC 18:23:58.05	Activity Task Scheduled	Activity Type: SyncFlow
10	2025-10-20 UTC 18:23:58.05	Workflow Task Completed	Identity: 1@1e567e1a2426@
9	2025-10-20 UTC 18:23:58.03	Workflow Task Started	History Size Bytes: 3289
8	2025-10-20 UTC 18:23:58.02	Workflow Task Scheduled	Task Queue Name: 1e567e1a2426:45f984a1-4e31-4a7a-8828-4d10d07f5492
7	2025-10-20 UTC 18:23:58.02	Activity Task Completed	Result: {"flowName": "mizroz_pg2ch_housesales", "source": {"n...
6	2025-10-20 UTC 18:23:58.02	Activity Task Started	Attempt: 1
5	2025-10-20 UTC 18:23:58.01	Activity Task Scheduled	Activity Type: GetFlowMetadata
4	2025-10-20 UTC 18:23:58.01	Workflow Task Completed	Identity: 1@1e567e1a2426@

Demo – PG Table Size

```
demo=# select count(*) as row_cnt, pg_size_pretty(pg_total_relation_size('house_sales')) as total_size from public.house_sales;
 row_cnt | total_size
```

```
-----+-----
 27734966 | 4203 MB
```

```
demo=# \d house_sales
```

Table "public.house_sales"				
Column	Type	Collation	Nullable	Default
id	integer		not null	
price	integer			
date	date			
postcode1	character varying(8)			
postcode2	character varying(3)			
type	character varying(13)			
is_new	smallint			
duration	character varying(9)			
addr1	character varying(100)			
addr2	character varying(100)			
street	character varying(60)			
locality	character varying(35)			
town	character varying(35)			
district	character varying(40)			
county	character varying(35)			

Indexes:

"house_sales_pkey" PRIMARY KEY, btree (id)

Publications:

"peerdb_publication"

Demo – CH Table Size

```
SELECT
  formatReadableSize(sum(data_compressed_bytes)) AS cmprs_size,
  formatReadableSize(sum(data_uncompressed_bytes)) AS raw_size,
  formatReadableQuantity(sum(rows)) AS row_cnt
FROM system.parts
WHERE ('table' = 'house_sales') AND active
```

Query id: ec7c3ba3-90c1-4b89-a831-b440a6f4898a

	cmprs_size	raw_size	row_cnt
1.	440.12 MiB	2.87 GiB	27.73 million

```
statement
CREATE TABLE uk.house_sales
(
  `id` Int32,
  `price` Int32,
  `date` Date32,
  `postcode1` String,
  `postcode2` String,
  `type` String,
  `is_new` Int16,
  `duration` String,
  `addr1` String,
  `addr2` String,
  `street` String,
  `locality` String,
  `town` String,
  `district` String,
  `county` String,
  `_peerdb_synced_at` DateTime64(9) DEFAULT now64(),
  `_peerdb_is_deleted` Int8,
  `_peerdb_version` Int64
)
ENGINE = ReplacingMergeTree(_peerdb_version)
PRIMARY KEY id
ORDER BY id
SETTINGS index_granularity = 8192
```

Demo – Power BI / Superset

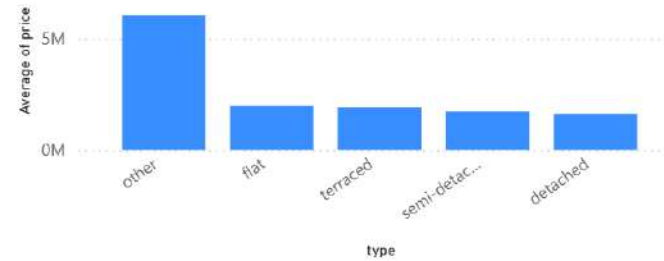
2.72M

Average of price

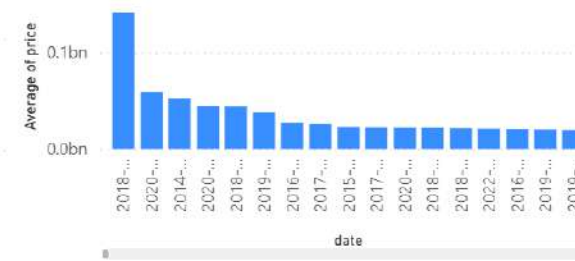
309.75K

Count of price

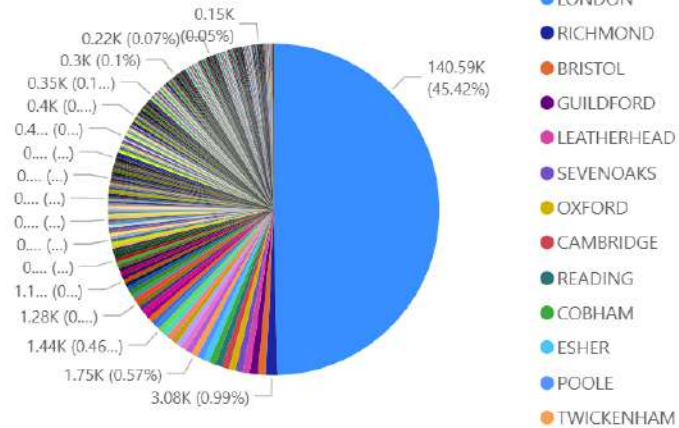
Average of price by type



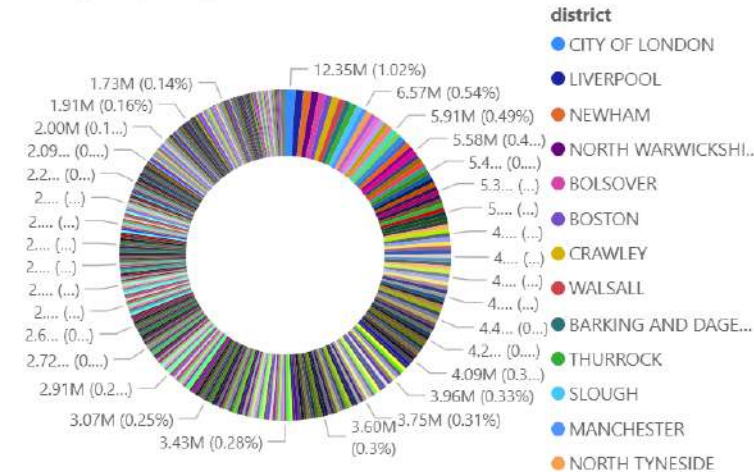
Average of price by date



Count of town by town



Average of price by district



THANK YOU!

QUESTIONS?