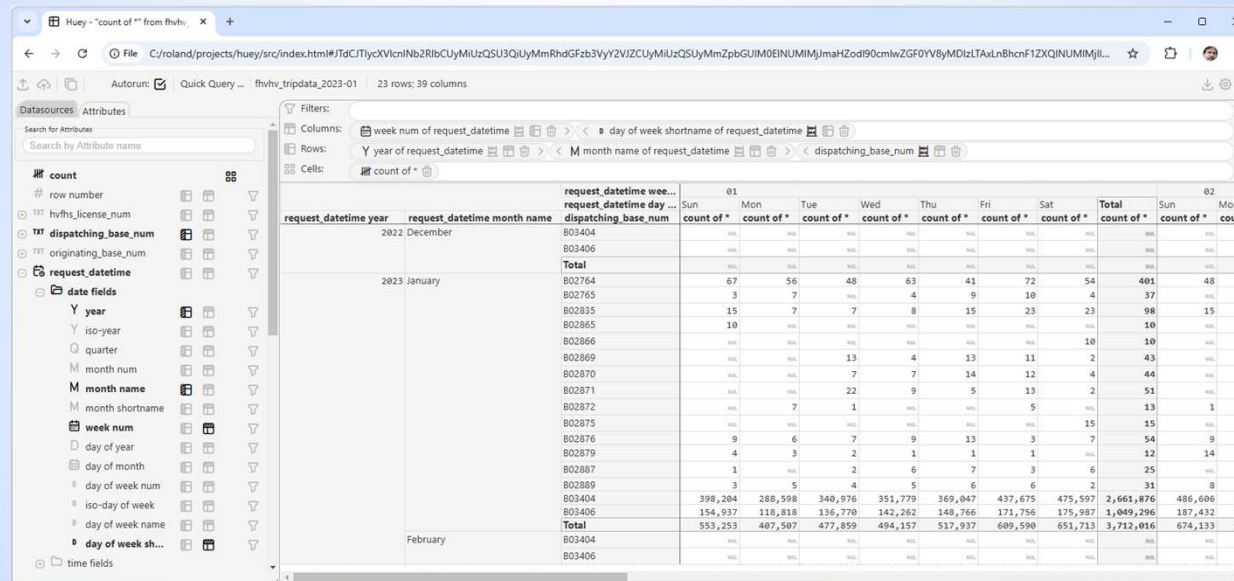


Huey



The screenshot displays the Huey web interface, a tool for querying data. The main area shows a table with 23 rows and 39 columns. The table is filtered by 'week num of request_datetime' and 'day of week shortname of request_datetime'. The interface includes a sidebar with 'Databases' and 'Attributes' sections, and a top navigation bar with 'Quick Query' and '23 rows, 39 columns'.

request_datetime year	request_datetime month name	request_datetime wee...	request_datetime day ...	dispatching_base_num	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Total	Sun	Mon
2022	December	B03404												
		B03406												
		Total												
2023	January	B02764			67	56	48	63	41	72	54	401	48	
		B02765			3	7		4	9	10	4	37		
		B02835			15	7	7	8	15	23	23	98	15	
		B02865			10							10		
		B02866									10	10		
		B02869					13	4	13	11	2	43		
		B02870					7	7	14	12	4	44		
		B02871					22	9	5	13	2	51		
		B02872				7	1				5	13	1	
		B02875									15	15		
		B02876			9	6	7	9	13	3	7	54	9	
		B02879			4	3	2	1	1	1		12	14	
		B02887			1		2	6	7	3	6	25		
		B02889			3	5	4	5	6	6	2	31	8	
		B03404			398,204	288,998	340,876	351,779	369,047	437,675	475,597	2,661,876	486,606	3
		B03406			154,937	118,818	136,770	142,262	148,766	171,756	175,987	1,049,296	187,432	1
		Total			553,253	407,507	477,859	494,157	517,937	609,590	651,713	3,712,016	674,133	4
	February	B03404												
		B03406												

Blazing fast Analytics on Parquet, CSV and JSON - right in your Browser

Hi! Thanks for joining me!

Roland Bouman

Chief Software Engineer
Data and Analytics branch
EPAM Systems B.V.

- linkedin.com/in/rpbouman
- github.com/rpbouman
- rpbouman.blogspot.com
- Roland.Bouman@gmail.com



What is Huey?

- Browser-based app
- Explore and analyze data
- Intuitive and powerful query builder
- Lazy Pivot table
- Export results
- Save/Load/Share queries



What's special about Huey?

- EASY!
 - Zero install
- SAFE!
 - Almost zero dependencies
 - Can run online or offline
- SECURE!
 - Local data processing
 - Data stays on your device
- FAST!
 - Powered by DuckDb / WASM
- FREE!
 - Open Source under a permissive MIT license

Using Huey

1. Load data from file or URL
 - ▀ CSV
 - ▀ Parquet
 - ▀ JSON
 - ▀ MS Excel
 - ▀ DuckDb database
2. Inspect the datamodel
 - ▀ Attributes
 - ▀ Derivations
 - ▀ Aggregators
 - ▀ Members
3. Place Items on Query Axes
 - ▀ Point-and-click
 - ▀ Drag-and-drop
4. Add Filters
 - ▀ IN, BETWEEN, LIKE
 - ▀ Array: ANY, ALL
5. Add totals
6. Export Results
 - ▀ Clipboard
 - ▀ File
7. Share query
 - ▀ Url hash

Live Demo

- Source
 - <https://github.com/rpbouman/huey/archive/refs/heads/dev.zip>
- Live Demo
 - <https://rpbouman.github.io/huey/src/index.html>
- Dataset
 - <https://data.lacity.org/api/views/ajiv-uc63/rows.csv?accessType=DOWNLOAD>



Download Source



Live Demo Site



Dataset

Using DuckDB WASM

```
<script type="module">
  // Boilerplate code, copy paste from https://duckdb.org/docs/api/wasm/instantiation
  import * as duckdb from 'https://cdn.jsdelivr.net/npm/@duckdb/duckdb-wasm@1.30.1-dev2.0/+esm';

  const JSDELIVR_BUNDLES = duckdb.getJsDelivrBundles();
  const bundle = await duckdb.selectBundle(JSDELIVR_BUNDLES);

  const worker_url = URL.createObjectURL(
    new Blob(['importScripts("${bundle.mainWorker}");'], {type: 'text/javascript'}))
  );
  const worker = new Worker(worker_url);
  const logger = new duckdb.ConsoleLogger();
  const db = new duckdb.AsyncDuckDB(logger, worker);
  URL.revokeObjectURL(worker_url);

  await db.instantiate(bundle.mainModule, bundle.pthreadWorker);
  const connection = await db.connect();

  // hoist DuckDB as a global
  window.hueyDb = {
    duckdb: duckdb,
    instance: db,
    connection: connection
  };
</script>
```

Registering Files

```
<script>
async function registerFile(file) {

    var duckdbInstance = window.hueyDb.instance;
    var protocol = window.hueyDb.duckDb.DuckDBDataProtocol.BROWSER_FILEREADER;
    var directIO = true;

    duckdbInstance.registerFileHandle(
        file.name,
        file,
        protocol,
        directIO
    ).then(function() {
        // we can now reference the file in our SQL statements
        var sql = `SELECT * FROM "${file.name}"`;
        ...
    }).catch(function(error) {
        // handle error
        ...
    });
}
</script>
```

Running Queries

```
<script>
async function doQuery(sql) {
  var connection = window.hueyDb.connection;
  var result = await connection.query(sql);
  return result;
}

var sql = `
  SELECT "dispatching_base_num" AS "dispatching_base_num"
  , "hvfhs_license_num" AS "hvfhs_license_num"
  , COUNT( * ) AS "* count"
  FROM read_parquet( "fhvhv_tripdata_2023-01.parquet" )
  GROUP BY "dispatching_base_num"
  , "hvfhs_license_num"
  ORDER BY "dispatching_base_num"
  , "hvfhs_license_num"
`;

doQuery(sql)
  .then( handleResult )
  .catch( handleError )
;
</script>
```

Consuming Results

```
<script>
function handleResult(resultSet) {
  for (var i = 0; i < resultSet.numRows; i++) {
    var row = resultSet.get(i);
    for (var j = 0; j < resultSet.schema.fields.length; j++){
      var field = resultSet.schema.fields[j];
      var value = row[field.name];
      var type = field.type;
      //
      // see: https://arrow.apache.org/docs/6.0/js/enums/Arrow\_dom.Type.html
      //
      switch (field.typeId) {
        case ...
          break;
        default:
          value = String(value);
      }
      // do something with the value
    }
  }
}
</script>
```

Downloading Results

```
<script>
async function downloadResult(sql, fileExtension) {

  var tmpFileName = `${crypto.randomUUID()}.${fileExtension}`;
  var copyStatement = `COPY ( ${sql} ) TO ${tmpFileName}`;

  var connection = window.hueyDb.connection;

  await connection.query( copyStatement );
  var data = await connection.copyFileToBuffer(tmpFileName);

  var mimeType = getMimeType(fileExtension);
  var blob = new Blob([data], {type: mimeType});
  var url = URL.createObjectURL(blob);

  var link = document.createElement('A');
  link.href = url;
  document.body.appendChild(link);

  link.click();
}
</script>
```

Roadmap

- Custom Attributes, Derivations, Aggregators
- SQL data sources
- Filter on aggregates
- Support for Secrets (to access URLs)
- Support for DuckDB Encryption
- Better Excel Experience
- Sorting
- Multi-data source Models (Datamarts / Star Schemas)

Links & Resources

- Github Project: <https://github.com/rpbouman/huey>
- Live Demo: <https://rpbouman.github.io/huey/src/index.html>
- Bugs & Feature requests: <https://github.com/rpbouman/huey/issues>
- Contact: Roland.Bouman@gmail.com
- Become a sponsor: <https://github.com/sponsors/rpbouman?o=nsm&sc=t>
- Donate: https://www.paypal.com/donate/?hosted_button_id=776A6UNZ35M84

Acknowledgements:

- DuckDB: <https://duckdb.org/>
- Tabler Icons: <https://tabler.io/icons>