SQL databases in your browser

Erik Söhnel (github.com/hoeck)

twenty years ago

what happened then

today

baby steps

September '22: postgres-wasm

github.com/snaplet made postgres-wasm:

Postgres running in the
browser *on top* of a
linux emulator using a
terminal emulator to interact with it

https://github.com/snaplet/postgres-wasm

just another dependency

Feb '24 - pglite

https://github.com/electric-sql/pglite

Just PostgreSQL compliled to Web Assembly!

```
$ npm install @electric-sql/pglite
```

hello world

```
import { PGlite } from "@electric-sql/pglite";

const db = new PGlite(); // a single connection
const result = await db.query("select 'Hello world' as msg;");
// {
//  "rows": [
//    {
//      "message": "Hello world"
//    }
//  ],
//  "fields": [
//    {
//      "name": "message",
//      "dataTypeID": 25
//    }
//  ],
//  "affectedRows": 0
//}
```

what cost? size!

```
$ ls -lh react/dist.tar.gz
-rw-r--r-- 1 erik erik 62K Feb  5 22:01 react/dist.tar.gz
```

vs

```
$ ls -lh react-pglite/dist.tar.gz
-rw-r--r-- 1 erik erik 4.3M Feb  5 22:01 react-pglite/dist.tar.gz
```

what cost? startup time!

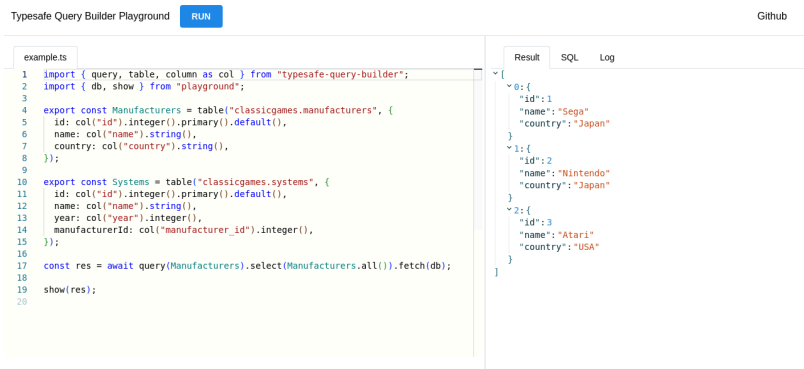
```
const start = Date.now();
const db = new PGlite();
const res = await db.query("select 'Hello world' as message");

console.log(Date.now() - start);
// ~2.5s (on my old thinkpad)
```

case study

cheap **orm sql playground**

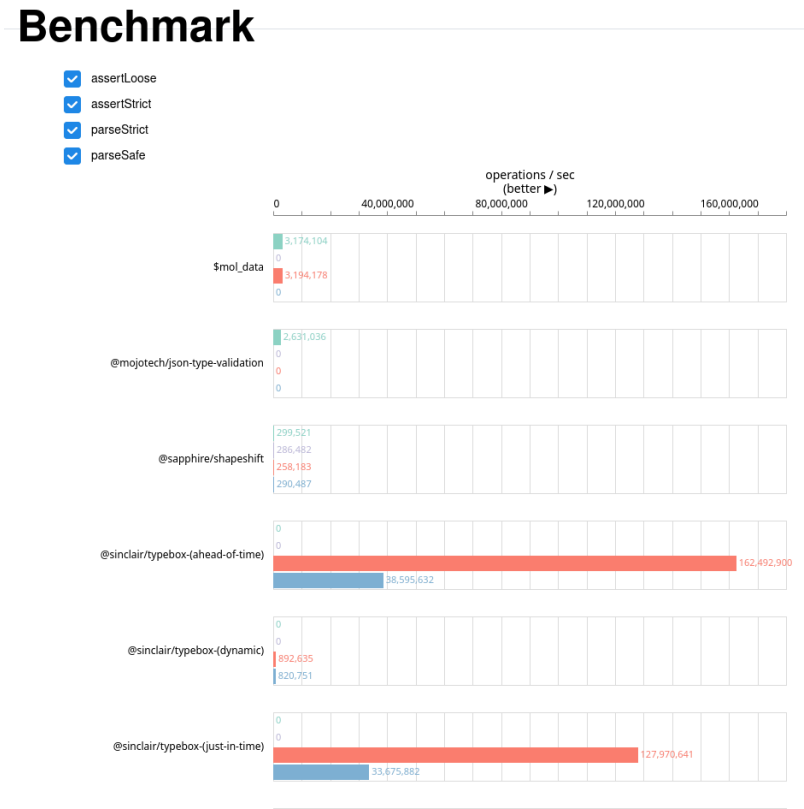
https://hoeck.github.io/typesafe-query-builder-playground/



case study

benchmark viewer prototype

uses react + only **sqlite** for state



example code (benchmark viewer)

```
const BenchmarkApp() => (
  <DatabaseContextProvider>
    <ResultsTable />
  </DatabaseContextProvider>
)
```

```
const Graph() => (
  const results = useDatabase((db) => {
    return db.findResults();
  });

  return (
    <table>
      {results?.map((r, i) => (...))}
    </table>
  )
)
```

pglite inline SQL

https://pglite.dev/docs/framework-hooks/react

```
import { useLiveQuery } from '@electric-sql/pglite-react'

const MyComponent = () => {
  const maxNumber = 100;
  const items = useLiveQuery.sql`
    SELECT *
    FROM my_table
    WHERE number <= ${maxNumber}
    ORDER BY number;
  `

  return ...
};
```

90s shit (digression)

90s shit (digression)

some learnings for me

CHORE: the database needs a schema first

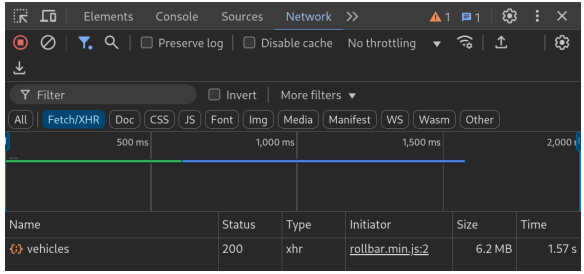
TYPES: without a good orms/query builders, its untyped and that sucks

FUN: listing / sorting stuff is so much easier in SQL

PERF: cannot memoize from SQL don’t have object identity

why do I care?

at work, users apparently like the software and use it



data always accumulates

recap

easy to install / use / operate SQL databases exist

they have downsides

they might make some things easier than before

welcome a new 🔨 in your toolbelt

SQL was already old when I started coding, 20yrs later and it looks well alive.

thanks for following along

github.com/hoeck

heavygoods.net/en/news/job-full-stack-developer

dresdenjs.io