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
- ၂<mark>s </mark> dinux emulator using a
- JS 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/typesafequery-builder-playground/

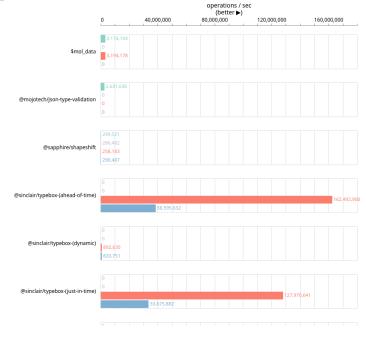
```
Typesafe Query Builder Playground
                                                                                                                                           Github
  example.ts
                                                                                             Result SQL Log
      import { guery, table, column as col } from "typesafe-guery-builder";
      import { db, show } from "playground";
                                                                                              "id":1
      export const Manufacturers = table("classicgames.manufacturers", {
                                                                                              "name": "Sega"
        id: col("id").integer().primary().default(),
                                                                                              "country": "Japan"
        name: col("name").string(),
       country: col("country").string(),
                                                                                              "id":2
                                                                                              "name": "Nintendo"
      export const Systems = table("classicgames.systems", {
                                                                                              "country": "Japan"
       id: col("id").integer().primary().default(),
        name: col("name").string(),
       year: col("year").integer(),
                                                                                              "id":3
       manufacturerId: col("manufacturer id").integer(),
                                                                                              "name": "Atari"
 15
                                                                                               "country": "USA"
     const res = await query(Manufacturers).select(Manufacturers.all()).fetch(db);
 19
     show(res);
```

case study

benchmark viewer prototype
uses react + only sqlite for state

Benchmark

- assertLoose
- assertStrict
- parseStrict
- parseSafe



example code (benchmark viewer)

```
const BenchmarkApp() \Rightarrow (
  <DatabaseContextProvider>
    <ResultsTable />
  </DatabaseContextProvider>
const Graph() \Rightarrow (
  const results = useDatabase((db) ⇒ {
    return db.findResults();
  });
  return (
    {results?.map((\mathbf{r}, i) \Rightarrow (...))}
```

pglite inline SQL

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

```
import { useLiveQuery } from '@electric-sql/pglite-react'
const MyComponent = () \Rightarrow \{
  const maxNumber = 100;
  const items = useLiveQuery.sql`
    SELECT *
   FROM my_table
    WHERE number ≤ ${maxNumber}
    ORDER BY number;
 return ...
```

some learnings for me

CHORE: the database needs a schema first

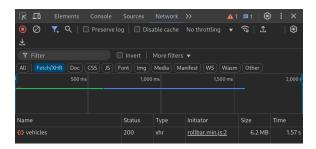
TYPES: without a good orms/query builders, its untyped and that sucks

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 🛢 🏺





















A services					
2015 Tarties E. 1915 Tarties E. 1915 Tarties E.		lar-stale Lastes	11.700 mm	00 0000	
7712 - 42751/970884.6 (1 - 4),b) 46. 3 FFM-4 (2750/97084.6 (1 - 4),b)		loroinio Endo) 20105 mm		
OTT JETTINES		laminale Fashes) 18300mm		
Section 2		territoire Zaales	: 18300mm		
505. 150 150 150 150 150 150 150 150 150 150		laminio Engles	1 16 Millione	000-0000	
200 275 m 1956		Teninalis Tenina	126.000 mm		
003 1982 Seed and Schiller 2010 87 (2004)		lar-dado Fasina	13400 cm		
(5) 10 July 2 Series Sept & 2007 (17 20%) (Report) 10 July 2 Series 10 July 2 Ser		Sectodo Saales	13600 em.		
1900 - 19		Tanks	11300cm	000-00000	
199 Timbe 6 RR FFINes on Selection		laminale Taubu	11300mm	70000-0000	
20 Thinks 154 Philips 154 Philips		Strategy ST miles	i Tillidam	0000-00000	
500. 150. Fairer s 150. Fairer s the fairer		Strategie 33 miles	: 1630cm	00000000	
200 3 Přířánsk 11,1,3 Přířánsk		teratula 11 miles	1 21 360 mm	00000-00000	
STATE OF THE STATE		Trades	125800 mm		
1900 - STEMAN ST		Services Si selen	12700 mm	00000-000000	
TO THE PARTY OF TH		Sentente Si salen	1 24 200 mm		
SEASON SE		Section Section 1	1 100 mm	0	
(1815) - Mariner John (177 MMCLE)		lar-state Easter	17800 mm		
CHES And Anthony countries or Estatem (spt. Staff and WEVA 200000). And the first Residency as and the or Estatem (spt. Staff and WEVA 200000). And Conference or Confer		lor-stude Fashes	i 11.600 mm		
een. Af daar wat de gegen An daar wat de gegen		landes	17900mm		
CETE CONTRACTOR AND ADDRESS OF THE CONTRACTOR AND ADDRESS OF THE CONTRACTOR ADDRESS OF THE CONTR		Sandra Sandra	11280 mm		
COST STATE OF STATE O		factors	1 Wildren		
CPU		Santos	- William	00000	
THE STATE OF THE S		Territolin Zasko	11 170 mm		
(SEA) (Read of March Resident lawy 1 (1970s - Seatronia (B audies) (Read of March (March (B audies)) (Read of March (B audies))		Tenden Teacher	1180cm	000	
periodical desiration of the second s		lander Easter	1 100 mm	-L ₀₀₀₀	
COSC THE MILE OF THE MANUAL STATES AND THE		Section States	: 18300 nan	00000000	
200 August 15 Augus 15 Augus 15 Augus 16 August 16 Augus		landale II siles	1 20100 mm.	00000000	
555. Section 17 Selection 17 Se		laminale Marke	21 William	000000000	
1993. When CH Aline When CH Al		Service of the section of the sectio	1 24000 mm	00000000000	
COST AND		Section 10	1 27900 mm	000000000000	
False S Asia		Trades	1 2000 cm.	00000000000000	
loss Bead Julein (et Bell) austins "SE (EE) aktickels SE Julein (eg Del panier (TS EE)) in all reliefs		lucitale Inde	: 11 Milesen		
Colonial Edition	0	Territorio Tendro	: Million		/ 0 1

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

neavygoods.net/en/news/job-rutt-stack-devetoper

dresdenjs.io