Challenges in storing large amounts of client-side data

Jeremy Scheff
Indie video game dev @ ZenGM.com

IndexedDB

Good stuff!

- Stores a lot of data
- Indexes, transactions
- Cross browser support
- idb and Dexie.js are 🔥

Limitations

- Performance
- Data persistence
- “Complex” queries

Complex queries in IndexedDB

JOINS, AGGREGATION, ETC

"

There's room for SQL-based APIs on top of IndexedDB.

- Arun Ranganathan
Mozilla Hacks blog 2010

IndexedDB as a building block

2010-2020

Dexie.js does a few fancy things

...not really much else out there

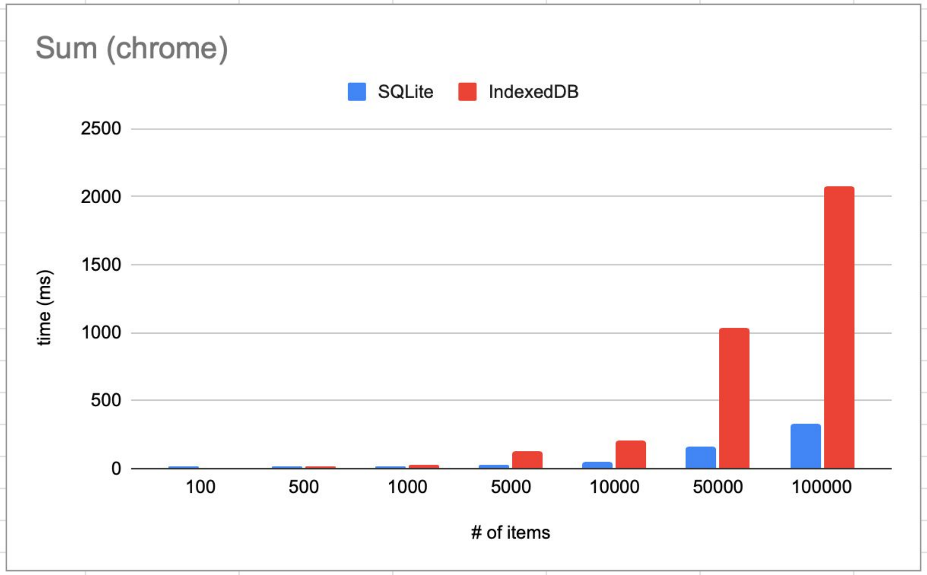
2021-present

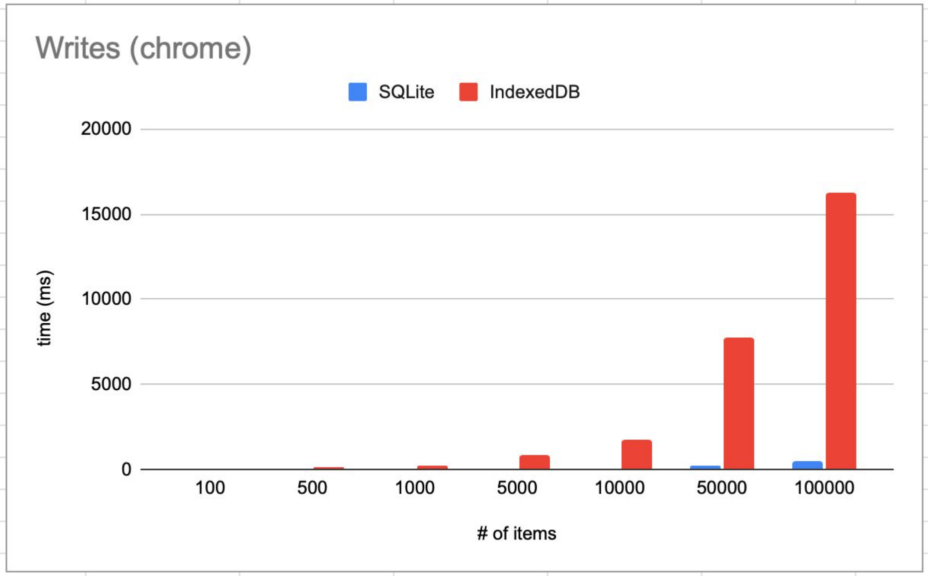
absurd-sql and wa-sqlite

SQL on top of IndexedDB!

SQLite on IndexedDB

- Compile SQLite to run in the browser
- Write backend to store block-level data in IDB





Graphs from:

https://jlongster.com/future-sql-web

#PWASummit2022

Limitations

- 🧪 Basically **experimental** projects made by individual devs
- 🐘 Large **bundle size**
- 🪨 Doesn't say anything about **durability/persistence**

Data persistence and durability

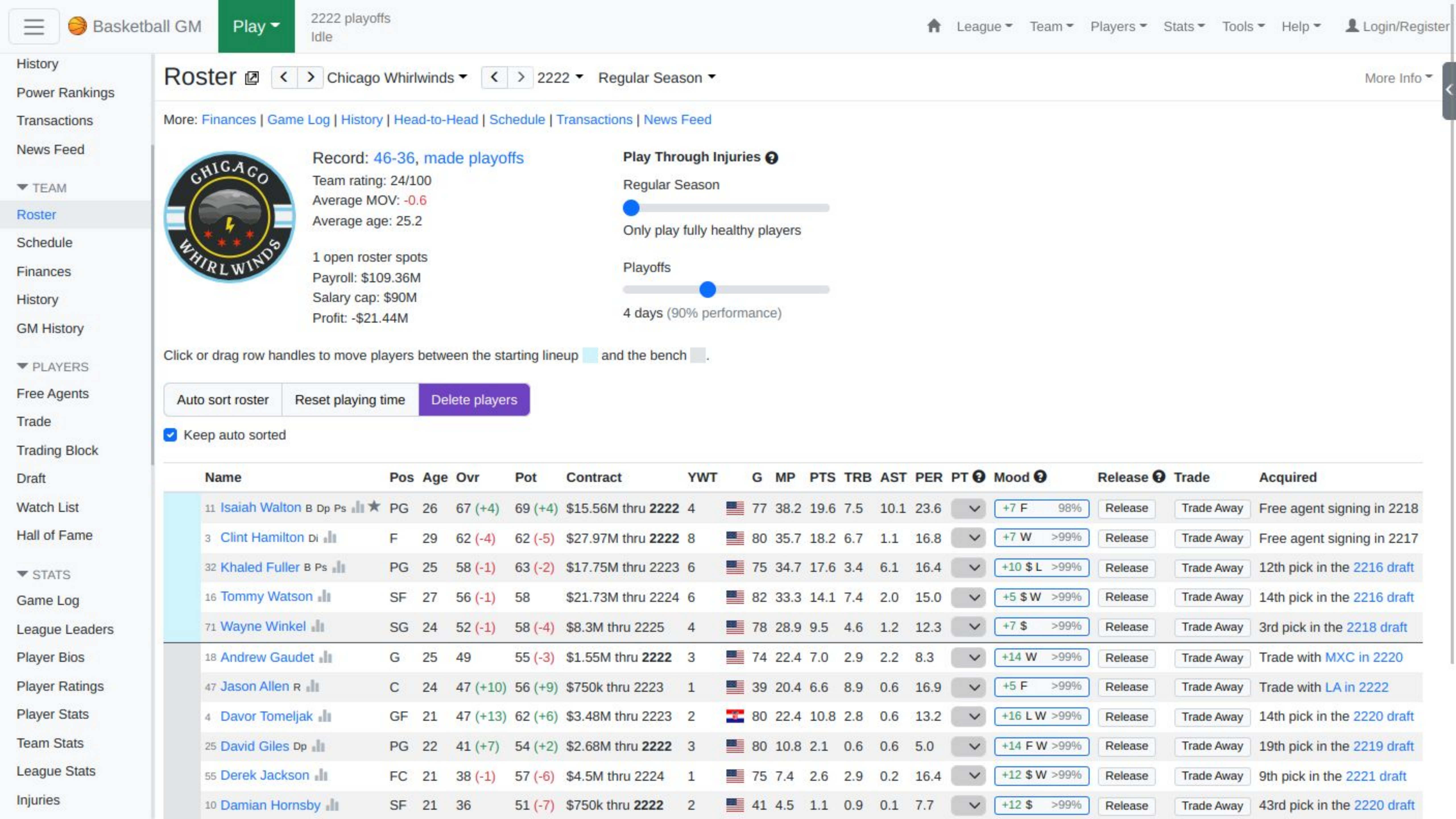
CAN YOU RELY ON YOUR DATABASE?

Offline first or offline only?

Offline first is great, but sometimes offline only is necessary:

💰 Data is large
🔃 Data changes frequently
🕵️ Private data
🖥️ Server is too complicated for some apps
💪 Desktop apps can do it, so why not PWAs?

Ultimately, **persistence matters for all PWAs**.



Roster

Chicago Whirlwinds 2222 Regular Season — More Info

More: Finances | Game Log | History | Head-to-Head | Schedule | Transactions | News Feed

Record: 46-36, made playoffs
Team rating: 24/100
Average MOV: -0.6
Average age: 25.2

1 open roster spots
Payroll: $109.36M
Salary cap: $90M
Profit: -$21.44M

Play Through Injuries

Regular Season
Only play fully healthy players

Playoffs
4 days (90% performance)

Click or drag row handles to move players between the starting lineup and the bench.

Auto sort roster | Reset playing time | Delete players

✓ Keep auto sorted

Name	Pos	Age	Ovr	Pot	Contract	YWT		G	MP	PTS	TRB	AST	PER	PT	Mood	Release	Trade	Acquired
11 Isaiah Walton B Dp Ps ★	PG	26	67 (+4)	69 (+4)	$15.56M thru **2222**	4	🇺🇸	77	38.2	19.6	7.5	10.1	23.6		+7 F 98%	Release	Trade Away	Free agent signing in 2218
3 Clint Hamilton Di	F	29	62 (-4)	62 (-5)	$27.97M thru **2222**	8	🇺🇸	80	35.7	18.2	6.7	1.1	16.8		+7 W >99%	Release	Trade Away	Free agent signing in 2217
32 Khaled Fuller B Ps	PG	25	58 (-1)	63 (-2)	$17.75M thru 2223	6	🇺🇸	75	34.7	17.6	3.4	6.1	16.4		+10 $ L >99%	Release	Trade Away	12th pick in the 2216 draft
16 Tommy Watson	SF	27	56 (-1)	58	$21.73M thru 2224	6	🇺🇸	82	33.3	14.1	7.4	2.0	15.0		+5 $ W >99%	Release	Trade Away	14th pick in the 2216 draft
71 Wayne Winkel	SG	24	52 (-1)	58 (-4)	$8.3M thru 2225	4	🇺🇸	78	28.9	9.5	4.6	1.2	12.3		+7 $ >99%	Release	Trade Away	3rd pick in the 2218 draft
18 Andrew Gaudet	G	25	49	55 (-3)	$1.55M thru **2222**	3	🇺🇸	74	22.4	7.0	2.9	2.2	8.3		+14 W >99%	Release	Trade Away	Trade with MXC in 2220
47 Jason Allen R	C	24	47 (+10)	56 (+9)	$750k thru 2223	1	🇺🇸	39	20.4	6.6	8.9	0.6	16.9		+5 F >99%	Release	Trade Away	Trade with LA in 2222
4 Davor Tomeljak	GF	21	47 (+13)	62 (+6)	$3.48M thru 2223	2	🇭🇷	80	22.4	10.8	2.8	0.6	13.2		+16 L W >99%	Release	Trade Away	14th pick in the 2220 draft
25 David Giles Dp	PG	22	41 (+7)	54 (+2)	$2.68M thru **2222**	3	🇺🇸	80	10.8	2.1	0.6	0.6	5.0		+14 F W >99%	Release	Trade Away	19th pick in the 2219 draft
55 Derek Jackson	FC	21	38 (-1)	57 (-6)	$4.5M thru 2224	1	🇺🇸	75	7.4	2.6	2.9	0.2	16.4		+12 $ W >99%	Release	Trade Away	9th pick in the 2221 draft
10 Damian Hornsby	SF	21	36	51 (-7)	$750k thru **2222**	2	🇺🇸	41	4.5	1.1	0.9	0.1	7.7		+12 $ >99%	Release	Trade Away	43rd pick in the 2220 draft

Inadvertent data loss in IndexedDB

- Browser configured to **clear data on close** (libraries, incognito mode, etc)
- User **manually clears** browser data
- **Eviction** when disk space is low, especially on mobile

Persistent storage - a solution?

“To prevent the browser from deleting your data, you can request that your entire site's storage be marked persistent.”

https://web.dev/persistent-storage/

```
const success = await navigator.storage.persist();
```

Persistent storage - challenges

- Browser may use heuristics to deny request without prompting user - **confusing!**
- Only helps with **eviction**
- May not completely help with eviction

Only addresses **part** of the problem, and does not completely solve that part.

Origin Private File System (OPFS)

- API for **high performance** file system access
- Not “real” files, private to origin
- Chrome + Safari now, Firefox in dev
- Better backend for **SQLite** than currently available
- Chrome devs and SQLite team working together 🐡

Origin Private File System (OPFS)

Persistence?

Spec: "Clearing browsing data **will not** affect those files in any way"

- Not just eviction!
- ...but deleting this data will still be tempting, right? Especially since they aren’t “real” files.

Open questions for the future

- 🌎 When will all browsers have OPFS support?
- 🐡 Will Chrome+SQLite come up with something good? Will it become popular?
- 🪨 Will persistence/durability still be an issue?
- 🐘 Are PWAs routinely going to be shipping a copy of SQLite? Will that be a sunk cost for the bundle size of a PWA with sophisticated client side data?

Thanks!

You can find me at:
@basketball_gm
jeremy@zengm.com
github.com/dumbmatter

I'll be answering questions via the PWA Summit Discord server

Play my PWA video games!

🏀 Basketball GM basketball-gm.com
🏈 Football GM football-gm.com
⚾ ZenGM Baseball zengm.com/baseball
🏒 ZenGM Hockey zengm.com/hockey