Do You Need a Server?

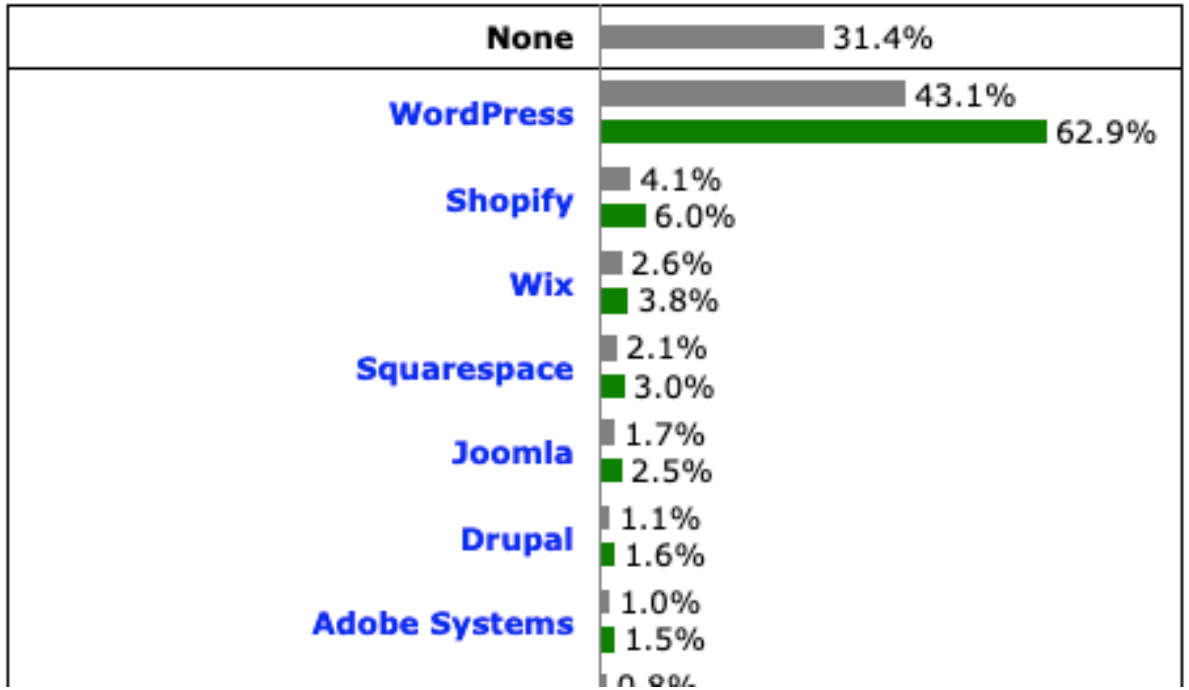
Low code, managed services, and flat files (and code too)

Andy Boughton

Abecasis Group Meeting

Jan 4, 2024

Motivation

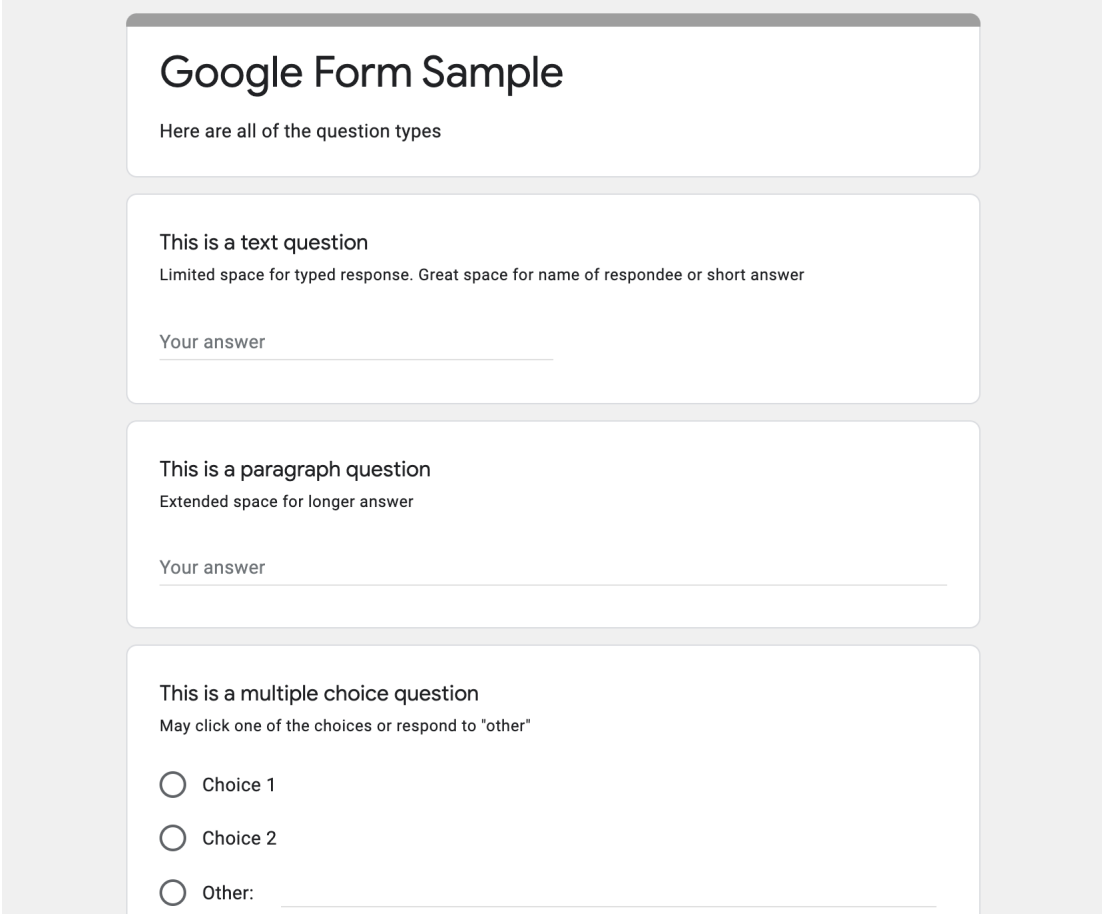


- In early days of computing, almost every web site had its own developer or infra
  - *webmaster@thing.com*
- Modern tools allow people to help themselves...
- How do we stay relevant?

https://docs.github.com/en/pages/setting-up-a-github-pages-site-with-jekyll
https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/getting-started-cloudfront-overview.html
https://w3techs.com/technologies/overview/content_management

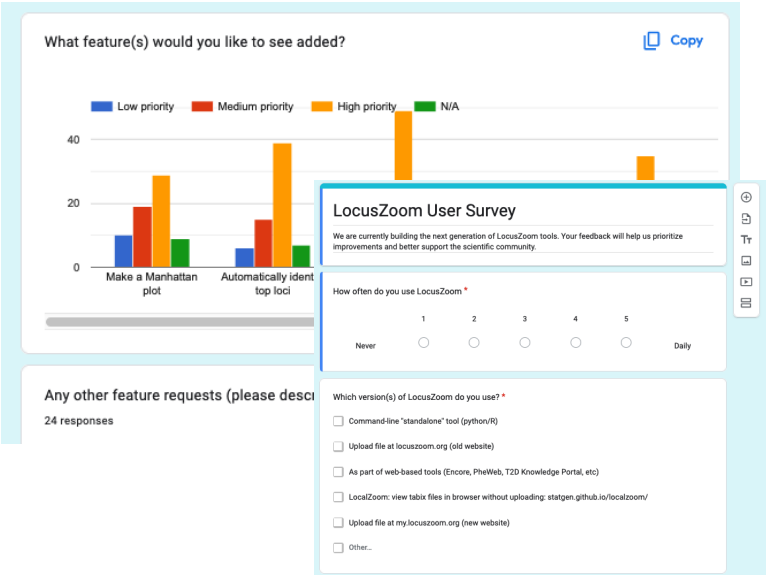
Low / No code: Platforms and tools

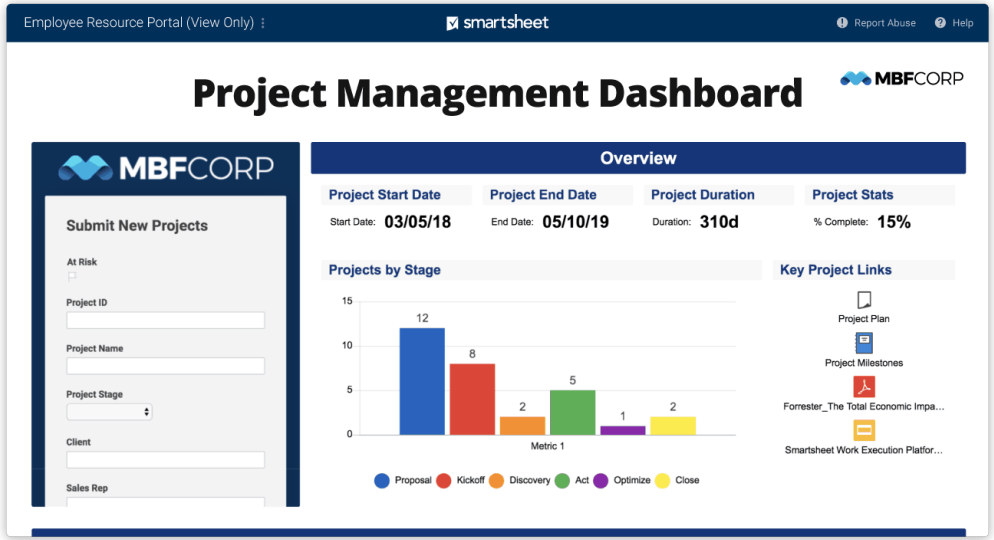
- Solve common types of tasks with a little customization
  - Wordpress, Smartsheet, Google Forms...
- Sometimes a very simple tool solves many common needs!
- Must understand pricing, customization, and requirements to use well
  - SaaS can be $$$ if you have many people or lots of data
  - Some tools offer choice of SaaS or on prem options. (Warning: they know what pays the bills)
  - May offer limited control over UI, permissions, etc.
- Beware of plugins
  - Wordpress has a plugin for anything, but most features you add this way will have security issues
  - Plugins often stop receiving updates, leaving migration path unclear



Spreadsheets all the way down

- People use spreadsheets for a lot of weird stuff- some tools build on what business people already know!
- Good for single-purpose projects with limited scope and one clear owner: “event RSVP” or “Project management dashboard”
- Beware of data corruption or “copy paste” errors as your project grows
  - A bad choice when you need “one clear source of truth”

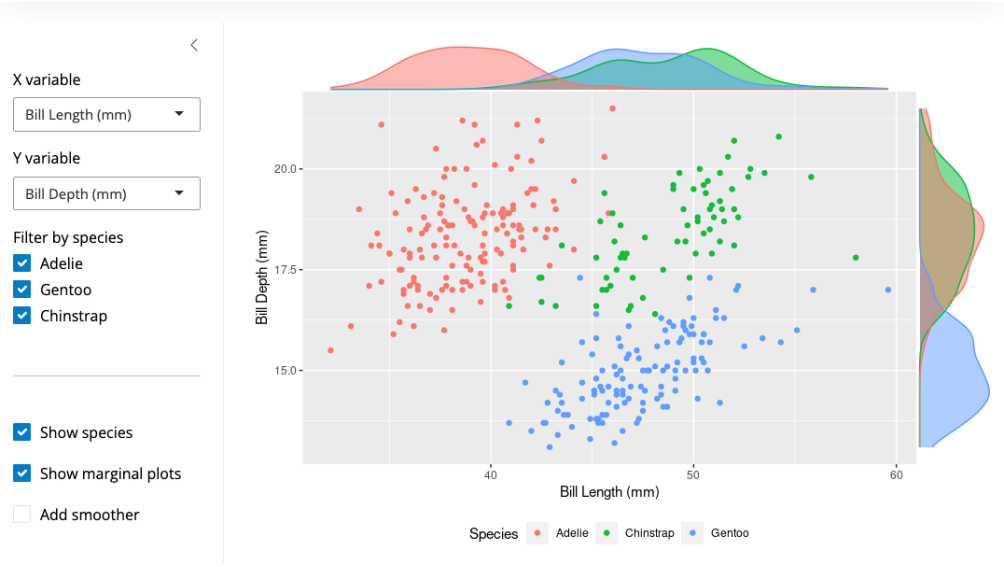




https://www.theverge.com/2023/10/21/23926585/microsoft-excel-misreading-dates-human-genes-conversion-fixed
https://www.smartsheet.com/content-center/product-news/reports-dashboards/smartsheet-dashboard-gallery

GUI-driven data exploration tools

- Popular for data exploration
  - R Shiny
  - Plot.ly / Dash
  - Jupyter notebook widgets
- Provide basic UI controls for freeform data exploration websites! Differentiators to look for:
  - How hard is this to set up / move to a new host?
  - Does it scale?
  - Access controls for public / private/ granular sharing
- UM has some tools!
  - Biostatistics offers R Shiny workshops
  - Some compute environments have JupyterLab installed for hosted notebooks



Host your own dataset

- For very well structured data, tools can automate data ingest and create a UI
  - PheWeb for genetics
  - Datasette for CSVs
  - PostgREST
  - AWS Athena, Snowflake, etc: SQL on CSV data lakes
- May automate data import, website creation, or both
  - PheWeb, Datasette
- Often designed with basic setup instructions: you need to provide your own server
- Plugins enable maps, custom templates, and other features

https://datasette.io/tutorials/explore
https://congress-legislators.datasettes.com/legislators/offices

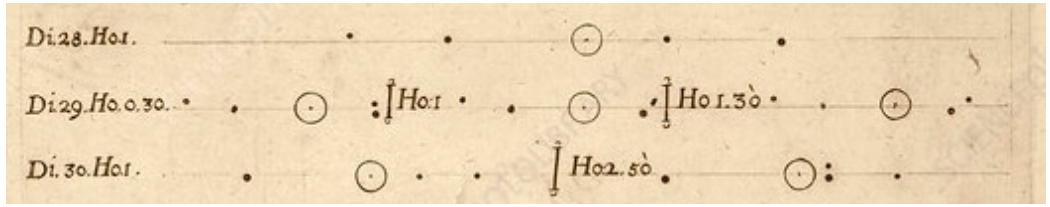
Where does this break down?

- Someone has to run the server and keep it updated
- Think about “multi user” scenarios
  - Controlling access to private data
  - “Per user” pricing is common in SaaS platforms. Can you release your prototype?
- Plan for change
  - “SQL from CSV” works great.... *if the file formats are consistent*! Research data changes schema often
  - Many “low code” tools focus on the ”baked data” pattern: user-provided or new data may be beyond the design of the tool
    - Imagine a timesheet system that didn’t let you enter data! Developers are still relevant.
- Often, low code tools are better for UI and exploring, but break down when offering complex new calculations
  - If you invite the entire planet to use one server, it may crash

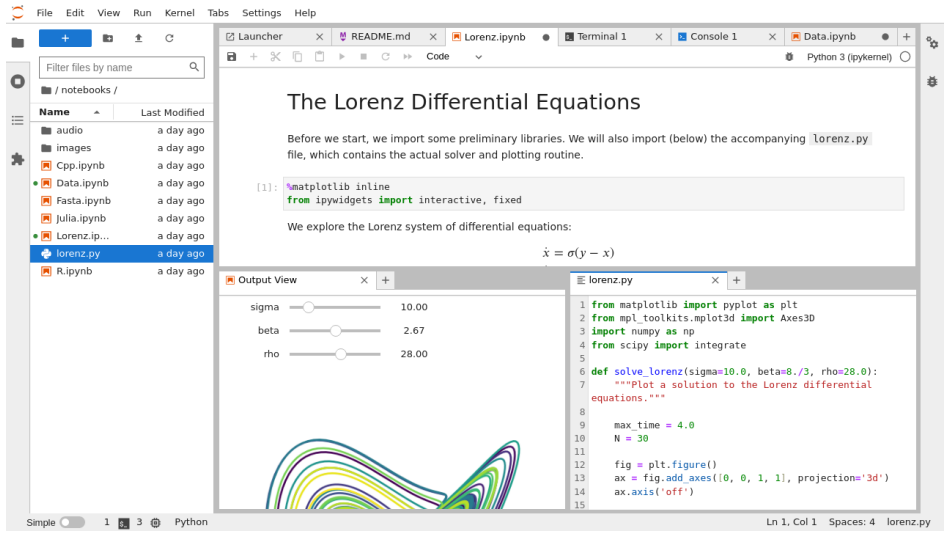
https://simonwillison.net/2021/Jul/28/baked-data/

Notebooks

- Mix code, results, and explanation
  - An old idea for the modern age: Galileo, Literate Programming, etc.



- If archived properly, would let a peer reviewer check your calculations and trace how analysis was done
- Some journals are experimenting with notebooks as a first-class artifact for publication / review.
- A potential win for portability / collaboration across teams: focus on exploration and communicating results!
  - Not every data set needs to live forever



https://datasciencenotebook.org/compare/jupyter/observable

A future for publishing?

"Notebooks are not included in the paper peer-review workflow, ... evaluation by reviewers into the data processing and thus results. ... AGU's current recommendations support the more popular end-to-end workflow that involves ... Jupyter Notebooks using GitHub (which, can also be used to render/display Notebooks), and creating/**linking to a runnable version** via Binder 14"...

- Galileo's notebooks are 400 yr old, but most published links last < 10 yr
  - ....And analysis code might break by the time the paper is submitted!
- AGU is experimenting with notebooks as a submission artifact
  - Physical science data is often less sensitive than human genetics
- No one has a great plan for how to archive the code parts long term

Table 2 Overview of which application supports the corresponding criteria. (N/D = no data)

	Authorea	Binder	Code Ocean	eLife RDS	Galaxy	Gigantum	Manuscripts	o2r	REANA	Repro Zip	Whole Tale
Free self-hosting	–	+	–	+*	+	–	+	+	+*	+	+
Open license	–	+	–	+	+	+/–	+	+	+	+	+
In use	in use [40]	in use [2]	in use [41]	in use [42]	in use [43]	–	–	–	in use [44]	in use [31]	–
Grant-based	–	+	–	+	+	–	N/D	+	+	+	+
R Markdown	–	+	+	+	–	+	–	+	–	–	+
Jupyter Notebooks	+	+	+	+	+	+	–	–	+	+	+
Extensible	–	+	+	+	+	–	–	–	+	+	+
Upload	+	+	+	–	+	–	+	+	–	–	+
Copyright	+	N/D	+	N/D	+	+	N/D	+	N/D	N/D	+
Sensitive data	–	–	–	–	–	–	–	–	–	–	–
Discovery	+	–	+	+	+	–	–	+	–	–	+
Inspection	+	+	+	+	+	+	+	+	–	–	+
Execution	+	+	+	+	+	+	+	+	+	+	+
Manipulation	+	+	+	+	+	+	+	+	+	+	+
Substitution	–	–	–	–	–	–	–	+	–	+	–
Download	+	+	+	+	+	+	+	+	–	+	+
Modify/Delete after publishing	–	+	–	–	+	+	+	–	+	+	–
Shared via DOI	+	–	+	+	–	–	–	–	–	–	+
Shared via URL	+	+	+	+	+	+	+	+	–	+	–

https://data.agu.org/notebooks-now/about

https://arxiv.org/ftp/arxiv/papers/2001/2001.00484.pdf

Hennessey, J. & Ge, S. X. A cross disciplinary study of link decay and the effectiveness of mitigation techniques. *BMC Bioinformatics* **14**, S5 (2013).
Trisovic, A., Lau, M. K., Pasquier, T. & Crosas, M. A large-scale study on research code quality and execution. *Sci Data* **9**, 60 (2022).

Archiving notebooks

- Who runs the backend to show the notebook?
  - What happens when old software is unsupported?
- What if all “smart” logic could be done by the client?
- Web based notebooks are trying to bridge this gap: basic compute features using WASM in the browser
  - Observable
  - JupyterLite
- Caveats:
  - Data still has to live somewhere, and that means you still need a server to query big data. Often not a self-contained archive
  - An incomplete solution for archiving: The notebook might run in the browser, but the library you use might not
  - JS is a bad language for data science

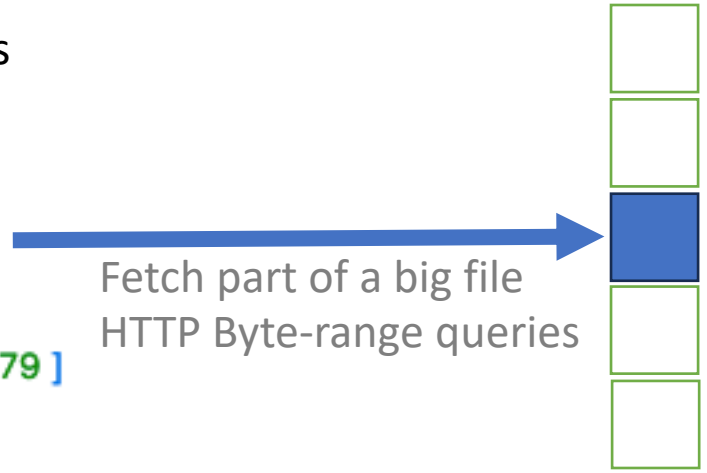
https://observablehq.com/@jkeefe/electoral-turnout
https://www.destroyallsoftware.com/talks/wat

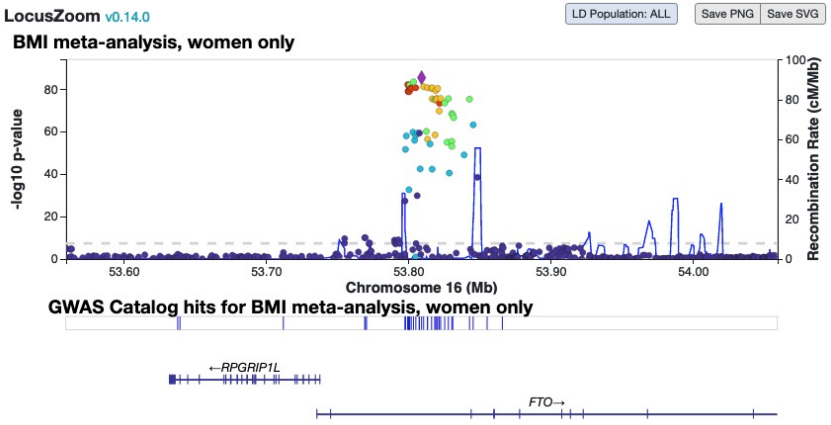
Tabix: Sometimes all you need is disk

- Popular genomics library for querying arbitrary parts of compressed data. Tiny index can be read client side and queried over HTTP
- Can be used to query big datasets client-side
- The basis of our LocalZoom visualization tool

Small index of “start / stop” ranges

- BRONX: [190, 173830]
- BROOKLYN: [173831, 537778]
- MANHATTAN: [537779, 815399]
- QUEENS: [815400, 1048598]
- STATEN ISLAND: [1048599, 1088079]

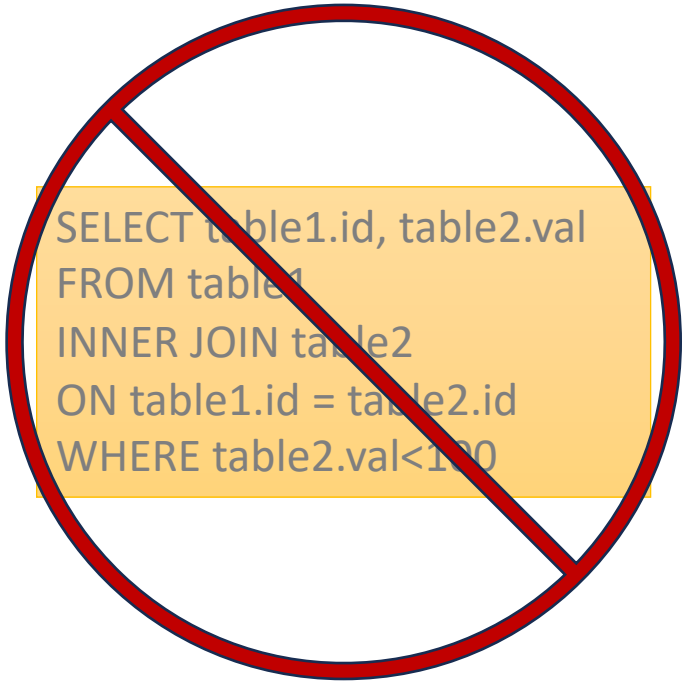




https://samtools.github.io/hts-specs/tabix.pdf
https://abought.github.io/weetabix/ ← Please don’t use this library for anything real
https://statgen.github.io/locuszoom/examples/ext/tabix_tracks.html

Strengths and limitations

- A notebook could use this to provide interactive exploration of big files, without running a server!
  - Drop files in S3 bucket and you have a website that “just works”
- BUT:
  - Usually requires data to be sorted (contiguous) for index to be useful
    - Fetching “every other row” would be messy, and the index would become bloated
  - Only ~1 kind of query per file: Since the dataset is queried as a block of bytes, much less flexible than SQL
    - “best of all time” are probably not “the first 10 sorted by year”
  - Bad for “JOIN logic”: you might fetch 200 records and end up using only 2
    - Network operations have way more overhead than disk I/O

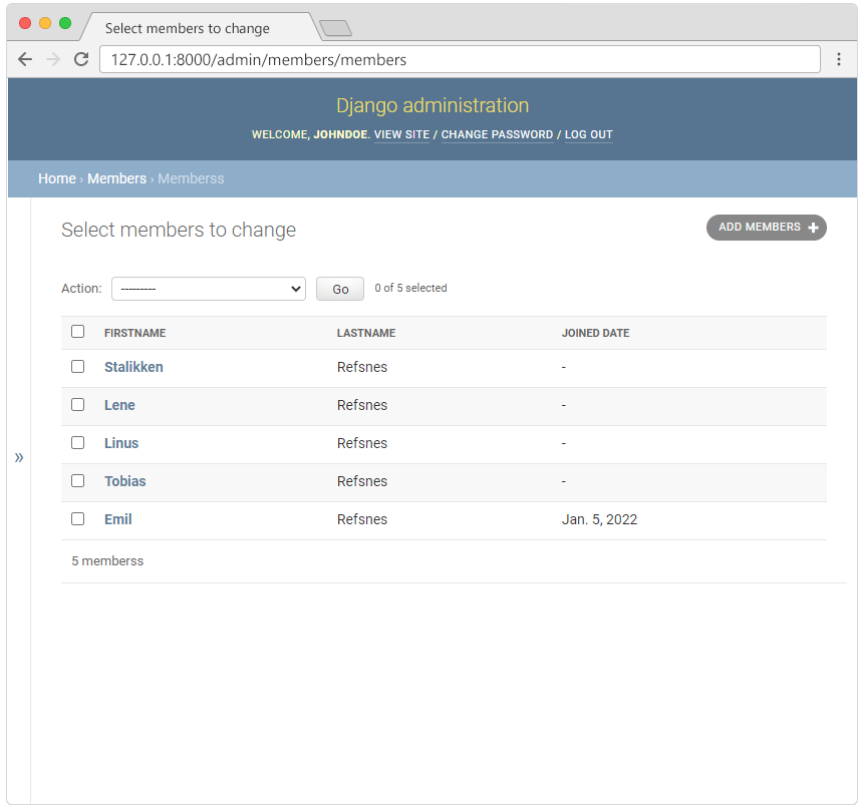


SQLite in the browser. Not great for “big data”- but possible! https://sql.js.org/#/

The future for specialized developers

Our skills are still needed for:

1. Granular access/ security requirements
2. Complex, long-lived datasets with one source of truth
3. Unique needs, especially across big data or large amounts of compute
4. We must focus on enabling productivity- don’t be a gatekeeper!
   - Let users handle common tasks themselves via admin interfaces and data validation



“

...historically our teams have been pretty independent from Enterprise IT... HPC operations ... [are] different technologies, “weird” operationally, and has very different quality of service requirements ...

There’s been six shifts in the last 15 years which have brought us closer to their orbit:

- Enterprise adoption of cloud
- Everyone wants Agile
- The explosion of data, funder requirements for data management
- The emergence of teaching data science
- Expansion of computing beyond RCD’s traditional fields
- Cybersecurity concerns

”

https://www.researchcomputingteams.org/newsletter_issues/0171