

# Portable In-Browser Data Cube Exploration

Kareem El Gebaly, Lukasz Golab, and Jimmy Lin

# Data exploration for everyone

From data democratization to analytics democratization

## Who?

- Data scientists
- Data analysts
- Data journalists
- And may be their audience!

## How?

- Easy to use
- Easy to interpret
- Does not require specialized infrastructure
- Does not require specialized pre-configurations

Plugged a full fledged SQL engine and a data exploration tool inside the browser.. so data exploration tasks can be easily shared with everyone without any external dependencies or pre-configurations.



**Afterburner –  
Explore the data cube in  
the browser**

**Explanation tables –  
Highlight the most  
informative  
parts of the cube**



id	item	season	location	expires?
1	Cheese	Winter	Kitchen	No
2	Cherries	Summer	Summer house	Yes
3	Chocolate	Summer	Summer house	No
4	Chocolate	Spring	Bedroom	No
5	Chocolate	Winter	Office	No
6	Chocolate	Summer	Basement	No
7	Chocolate	Fall	Winter house	No
8	Eggs	Fall	Kitchen	Yes
9	Eggs	Winter	Winter house	Yes
10	Juice	Spring	Office	No
11	Milk	Spring	Office	Yes
12	Milk	Summer	Winter house	Yes
13	Veggies	Spring	Summer house	Yes
14	Veggies	Winter	Winter house	Yes





**item season location count expires?**

*	*	*	14	7/14
---	---	---	----	------



**item season location count expires?**

Cheese	*	*	1	0/1
Cherries	*	*	1	1/1
Chocolate	*	*	5	<b>0/5</b>



**item season location count expires?**

*	Winter	*	4	2/2
*	Summer	*	4	2/2
*	Spring	*	4	2/2



**item season location count expires?**

*	*	Kitchen	2	1/2
*	*	Bedroom	1	0/1
*	*	Office	3	1/3

**Potentially |items| \* |seasons| \* |locations| patterns!**

<b>item</b>	<b>season</b>	<b>location</b>	<b>count</b>	<b>expires?</b>
*	*	*	14	7/14
Chocolate	*	*	5	0/5
*	*	Winter House	4	3/4
		Summer House	3	2/3

# Explanation tables:

- 1. Information theoretic approach to highlight the most important parts of the cube**
- 2. Iterative scaling finds maximum entropy estimates**
- 3. Sample based approach for pruning the datacube**



**Afterburner is an in browser SQL engine that uses Code Generation that almost matches the state of the art SQL engines running native on the same machine.**

**Afterburner exploits two JavaScript features:**

**JavaScript typed arrays:**

- Contiguous in memory storage
- Predefined types using typed views
- Similar storage efficiency to C arrays

**Asm.js:**

- Statically-typed subset of JavaScript
- Amenable to AOT optimization
- On average  $\sim 1.5\times$  slower than native code





# Demo scenario

- Live demo (ALT-TAB)

# Conclusion

- Easy to interpret summaries
- Intuitive starting point for data exploration
- In browser implementation requires no configuration and easy sharing
  
- Please check out our live demo at:
  - <https://afterburnerdb.github.io/afterburner/explore.html>
- Find our open source code:
  - <https://github.com/afterburnerdb/afterburner>