CS 143 Intro to Database Management Systems

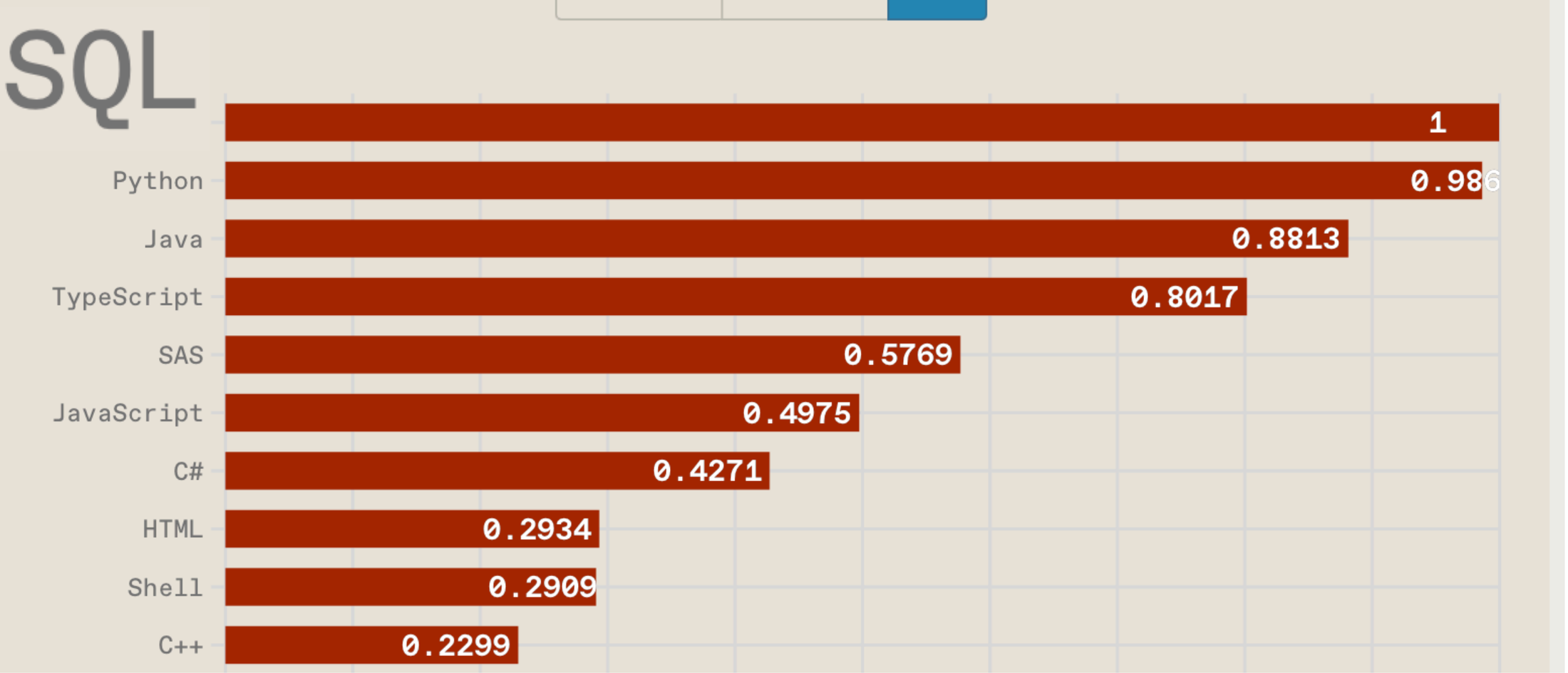
Remy Wang, 2025-04-01

Why are you here?

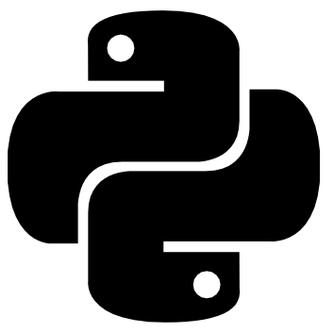
Top Programming Languages 2024

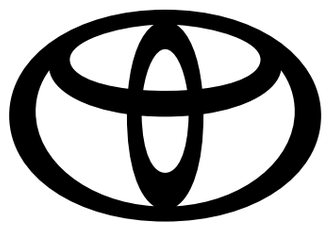
Click a button to see a differently weighted ranking

Spectrum | Trending | Jobs



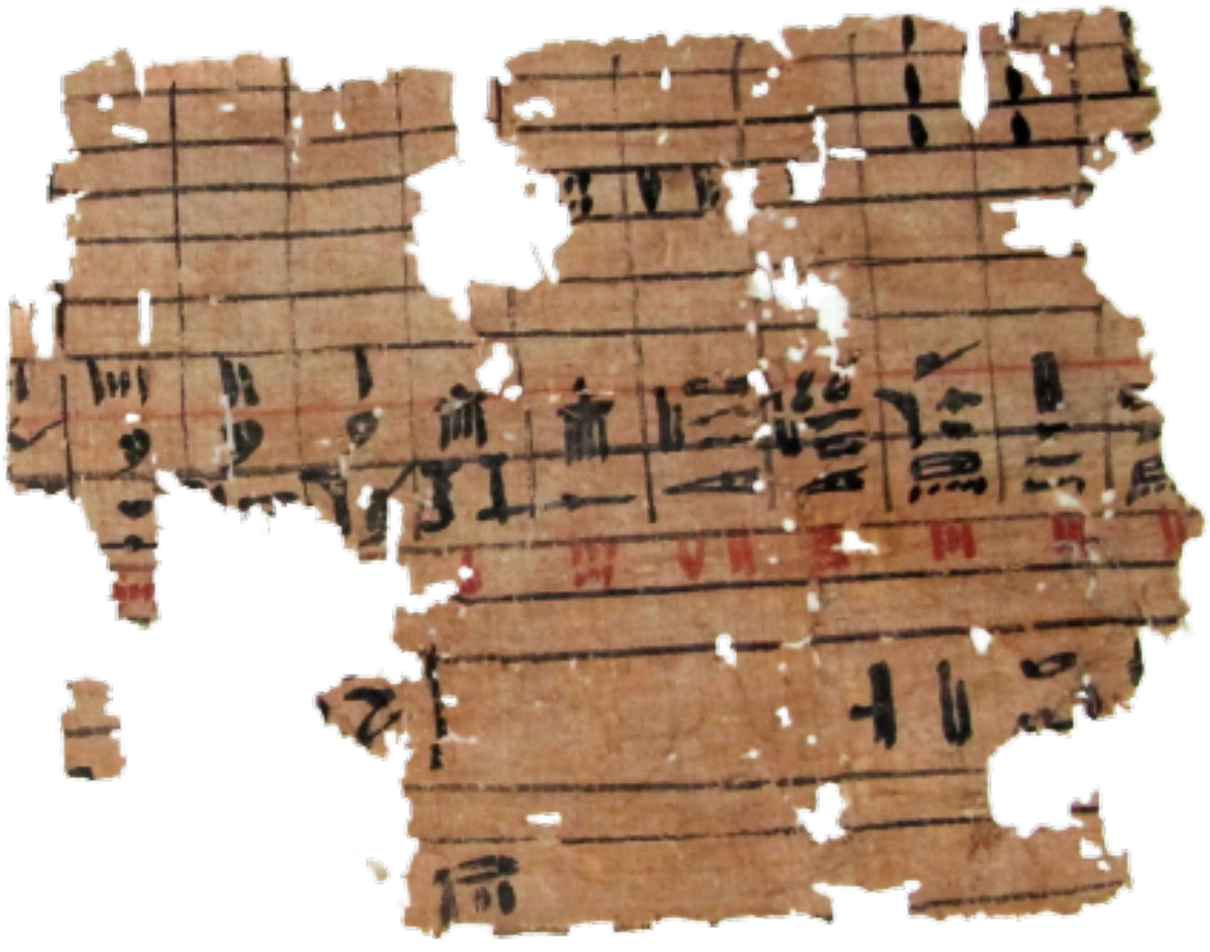
$10^{12} \times$ SQLite

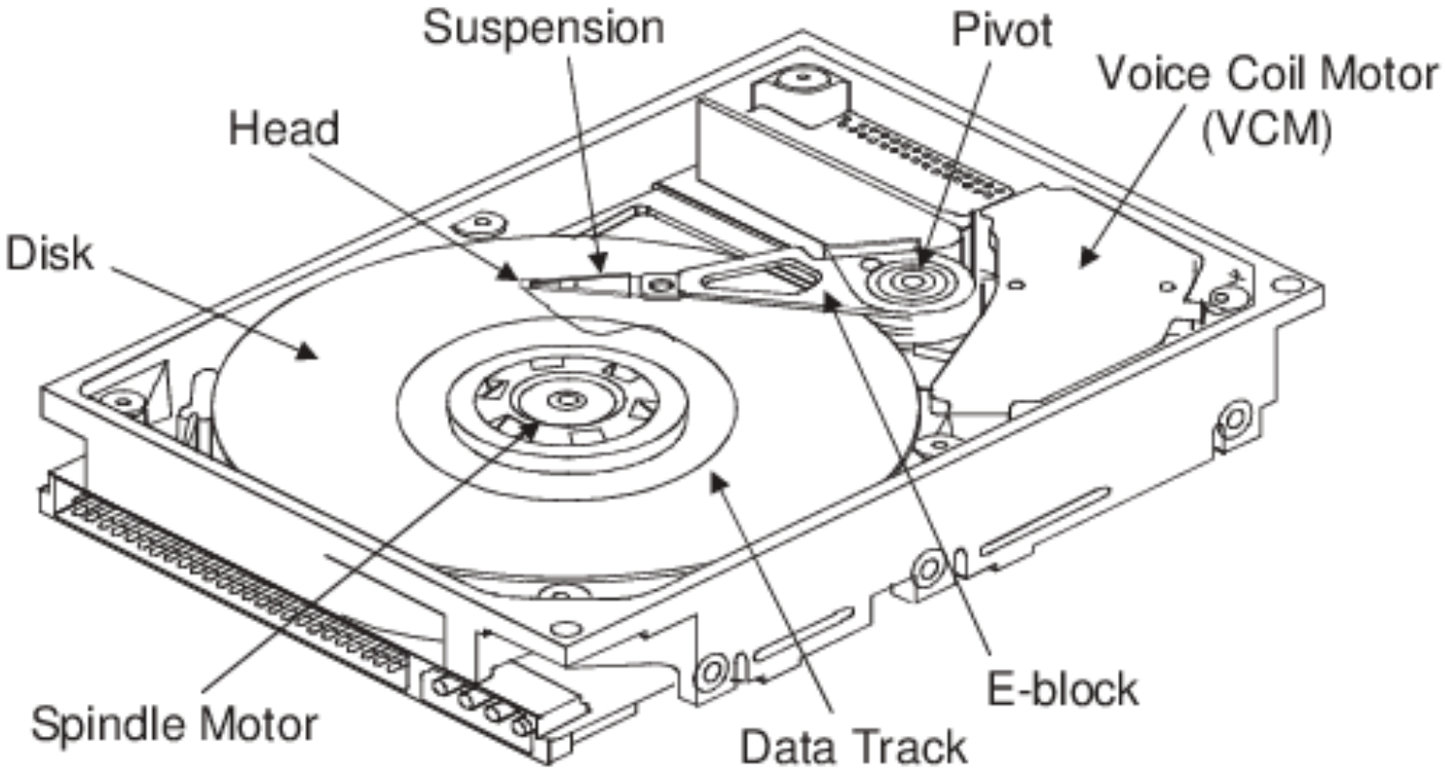






Problem	FAQ formulation	Previous Algo.	**Our Algo.**
#QCQ	$\sum_{(x_1,\ldots,x_f)} \oplus_{x_{f+1}}^{(f+1)} \cdots \oplus_{x_n}^{(n)} \prod_{S\in\mathcal{E}} \psi_S(\mathbf{x}_S)$ where $\bigoplus^{(i)} \in \{\max, \times\}$	No non-trivial algo	$\tilde{O}(N^{\mathrm{faqw}(\varphi)} + \|\varphi\|)$
QCQ	$\oplus_{x_{f+1}}^{(f+1)} \cdots \oplus_{x_n}^{(n)} \prod_{S\in\mathcal{E}} \psi_S(\mathbf{x}_S)$ where $\bigoplus^{(i)} \in \{\max, \times\}$	$\tilde{O}(N^{\mathrm{PW}(\mathcal{H})} + \|\varphi\|)$ [25]	$\tilde{O}(N^{\mathrm{faqw}(\varphi)} + \|\varphi\|)$
#CQ	$\sum_{(x_1,\ldots,x_f)} \max_{x_{f+1}} \cdots \max_{x_n} \prod_{S\in\mathcal{E}} \psi_S(\mathbf{x}_S)$	$\tilde{O}(N^{\mathrm{DM}(\mathcal{H})} + \|\varphi\|)$ [35]	$\tilde{O}(N^{\mathrm{faqw}(\varphi)} + \|\varphi\|)$
Joins	$\bigcup_{\mathbf{x}} \bigcap_{S\in\mathcal{E}} \psi_S(\mathbf{x}_S)$	$\tilde{O}\left(N^{\mathrm{fhtw}(\mathcal{H})} + \|\varphi\|\right)$ [47]	$\tilde{O}\left(N^{\mathrm{fhtw}(\mathcal{H})} + \|\varphi\|\right)$
Marginal	$\sum_{(x_{f+1},\ldots,x_n)} \prod_{S\in\mathcal{E}} \psi_S(\mathbf{x}_S)$	$\tilde{O}(N^{\mathrm{htw}(\varphi)} + \|\varphi\|)$ [55]	$\tilde{O}(N^{\mathrm{faqw}(\varphi)} + \|\varphi\|)$
MAP	$\max_{(x_{f+1},\ldots,x_n)} \prod_{S\in\mathcal{E}} \psi_S(\mathbf{x}_S)$	$\tilde{O}(N^{\mathrm{htw}(\varphi)} + \|\varphi\|)$ [55]	$\tilde{O}(N^{\mathrm{faqw}(\varphi)} + \|\varphi\|)$
MCM	$\sum_{x_2,\ldots,x_n} \prod_{i=1}^{n} \psi_{i,i+1}(x_i, x_{i+1})$	DP bound [29]	DP bound
DFT	$\sum_{\substack{(y_0,\ldots,y_{m-1})\\ \in\mathbb{Z}_p^m}} b_y \cdot \prod_{0\le j+k<m} e^{i2\pi \frac{x_j\cdot y_k}{p^{m-j-k}}}$	$O(N \log_p N)$ [28]	$O(N \log_p N)$


Suspension
Pivot
Voice Coil Motor
(VCM)
Head
Disk
Spindle Motor
Data Track
E-block

Why do we need DBs?

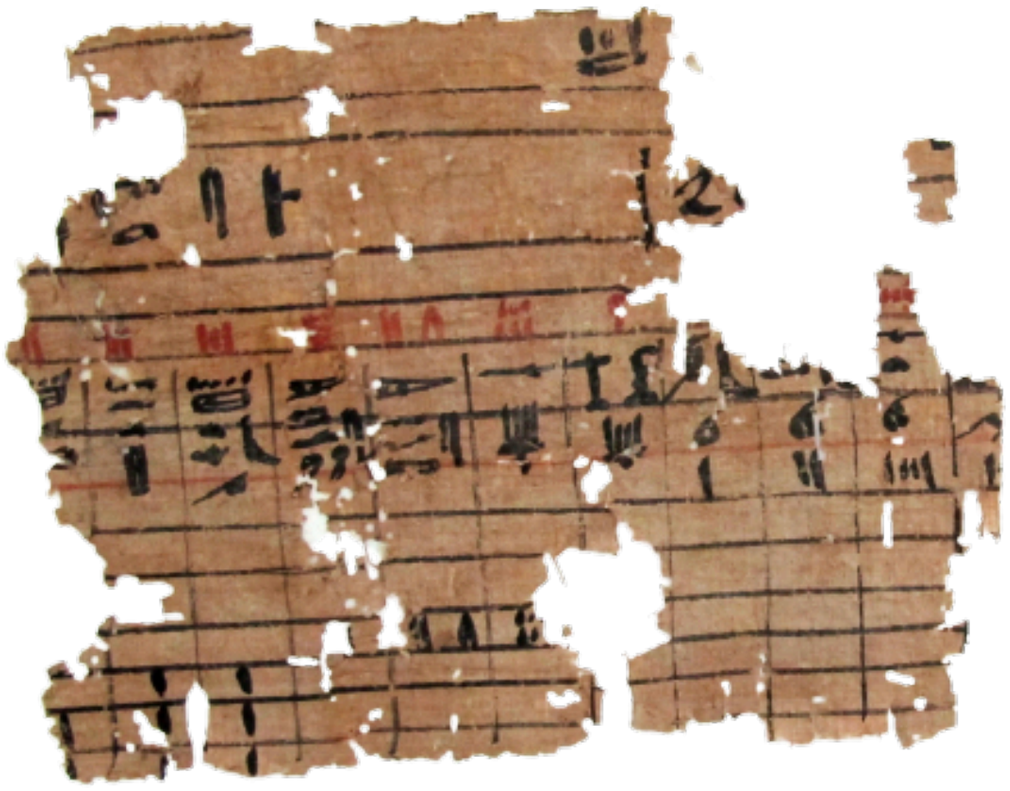
Store

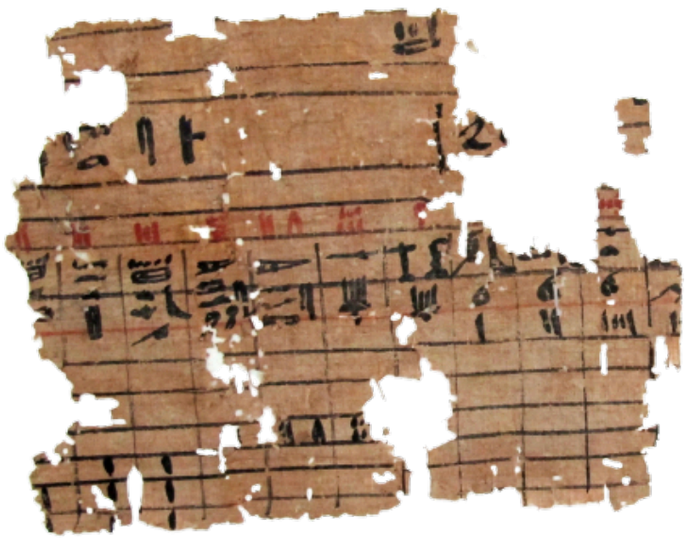
Analyze

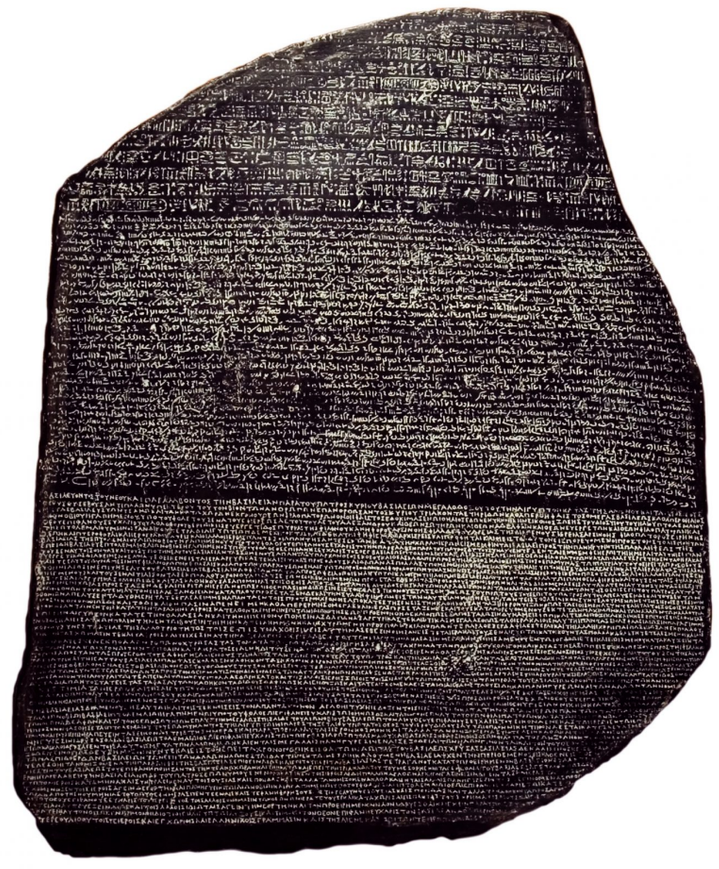
Share

} Data

What can we use as DBs?







Good properties of a DB?

Scalable

Efficient

Durable

Concurrent

n by k grid of data

table name

same column length
same type of data

pets

name	breed	age	origin	kind
casa	tabby	8	seatte	cat
kira	tuxedo	6	hawaii	cat
toby	border collie	17	seattle	dog
maya	husky	10	LA	dog

Schema

same row length

anatomy of a table (a.k.a. *relation*)

Database: many tables

pets

name	breed	age	origin	kind
casa	tabby	8	seatte	cat
kira	tuxedo	6	hawaii	cat
toby	border collie	17	seattle	dog
maya	husky	10	LA	dog

people

name	pet	addr.	phone	job
remy	casa	LA	###	UCLA

places

name	addr.	type
UCLA	LA	edu.

Database: many tables

pets

name	breed	age	origin	kind
casa	tabby	8	seatte	cat
kira	tuxedo	6	hawaii	cat
toby	border collie	17	seattle	dog
maya	husky	10	LA	dog

people

name	pet	addr.	phone	job
remy	casa	LA	###	UCLA

places

name	addr.	type
UCLA	LA	edu.

Database: many tables

pets

name	breed	age	origin	kind
casa	tabby	8	seatte	cat
kira	tuxedo	6	hawaii	cat
toby	border collie	17	seattle	dog
maya	husky	10	LA	dog

people

name	pet	addr.	phone	job
remy	casa	LA	###	UCLA

places

name	addr.	type
UCLA	LA	edu.

Why this design?

name	breed	age	origin	kind
casa	tabby	8	seatte	cat
kira	tuxedo	6	hawaii	cat
toby	border collie	17	seattle	dog
maya	husky	10	LA	dog

Simple!

Intuitive for humans

Easy for machines

Scalable: distributed storage

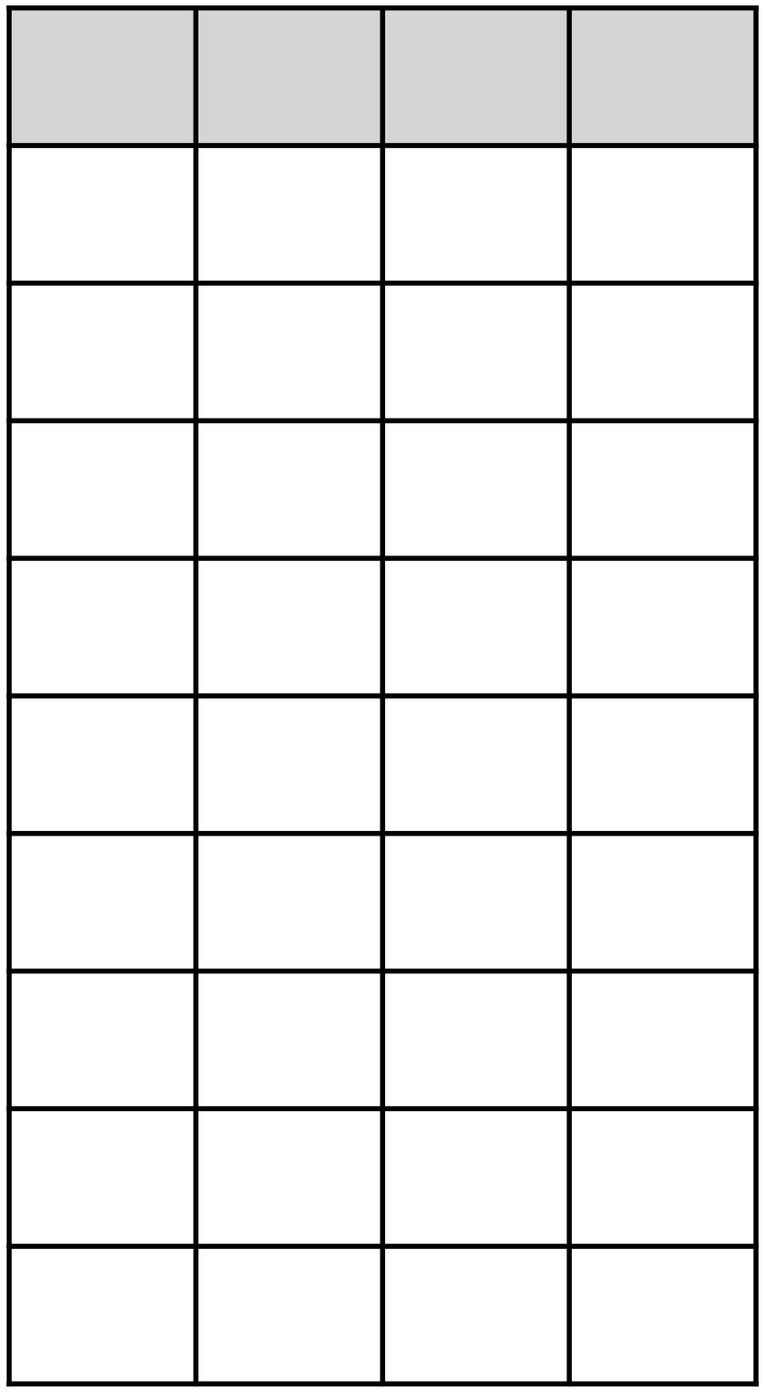


Table too large?

Split it over machines!

Scalable: distributed storage

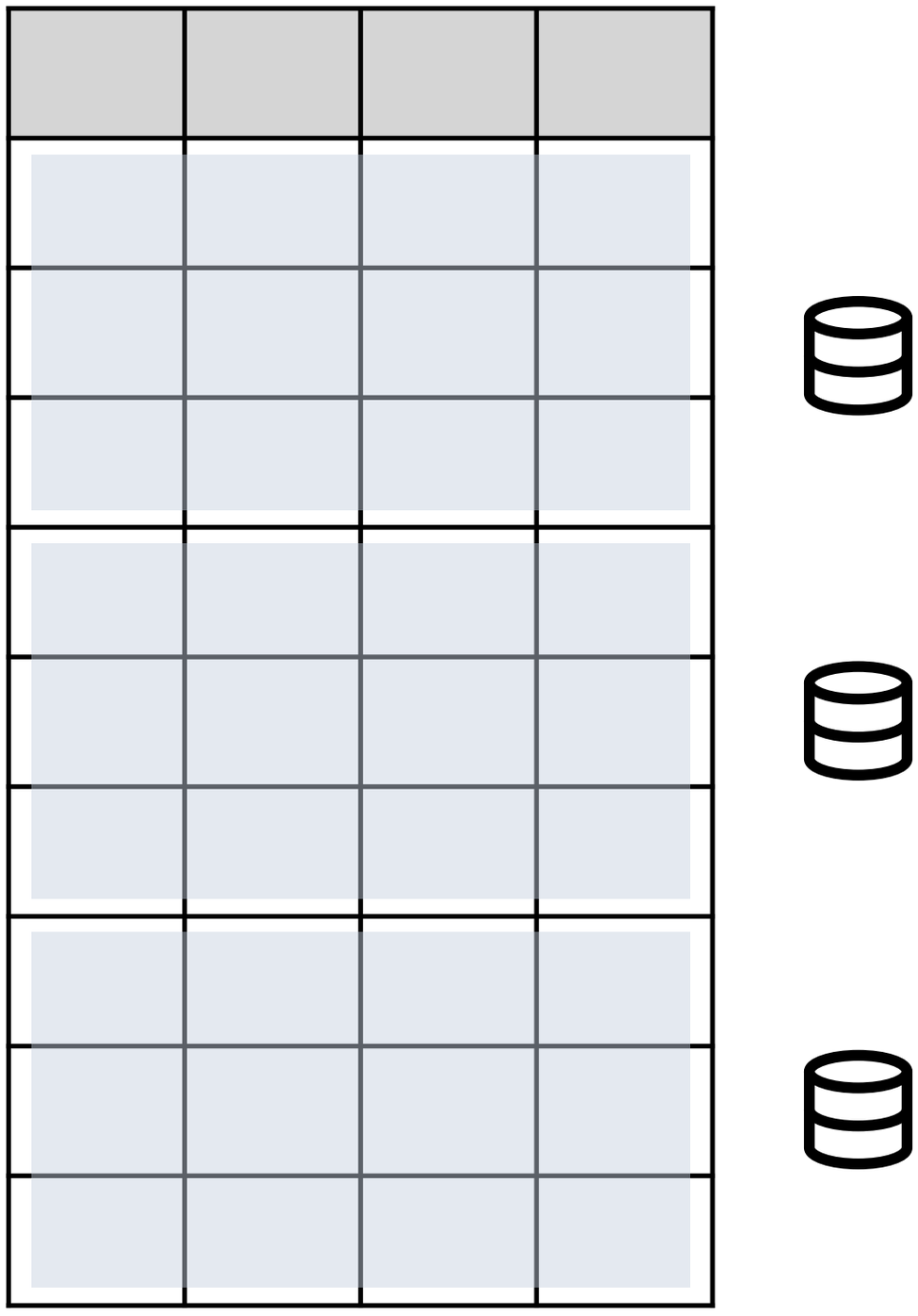


Table too large?

Split it over machines!

Scalable: distributed compute

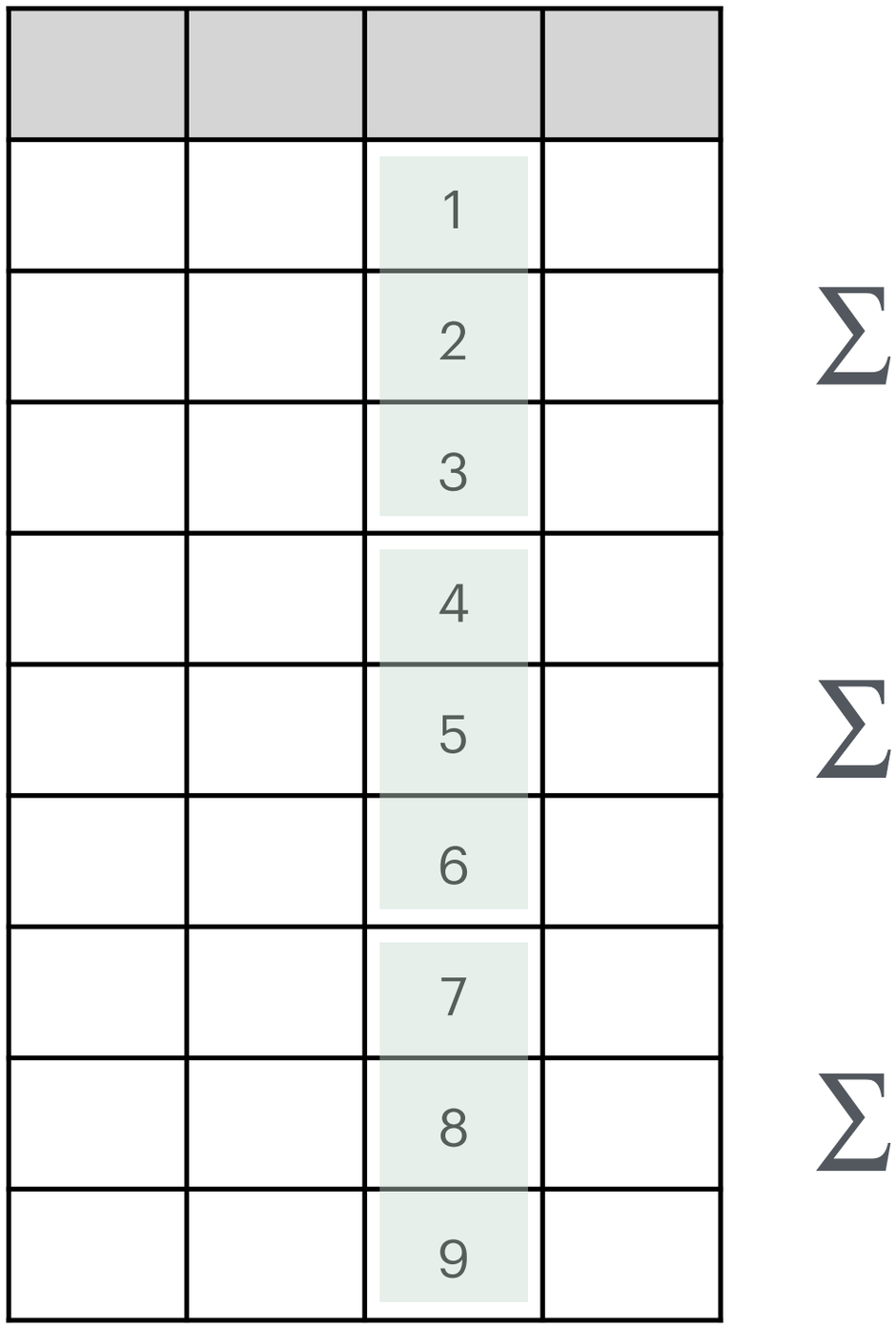
		1	
		2	
		3	
		4	
		5	
		6	
		7	
		8	
		9	

$\sum$

Computer too slow?

Add more computers!

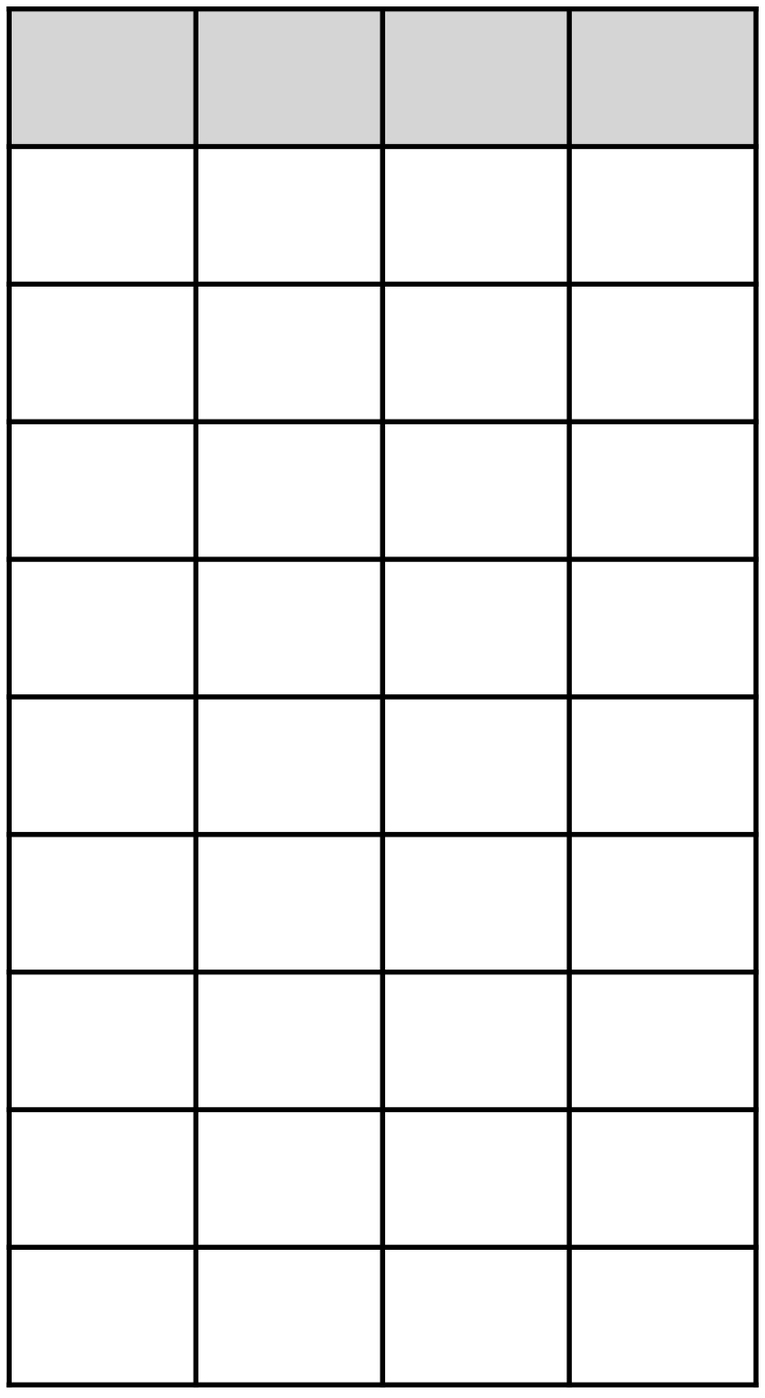
Scalable: distributed compute



Computer too slow?

Add more computers!

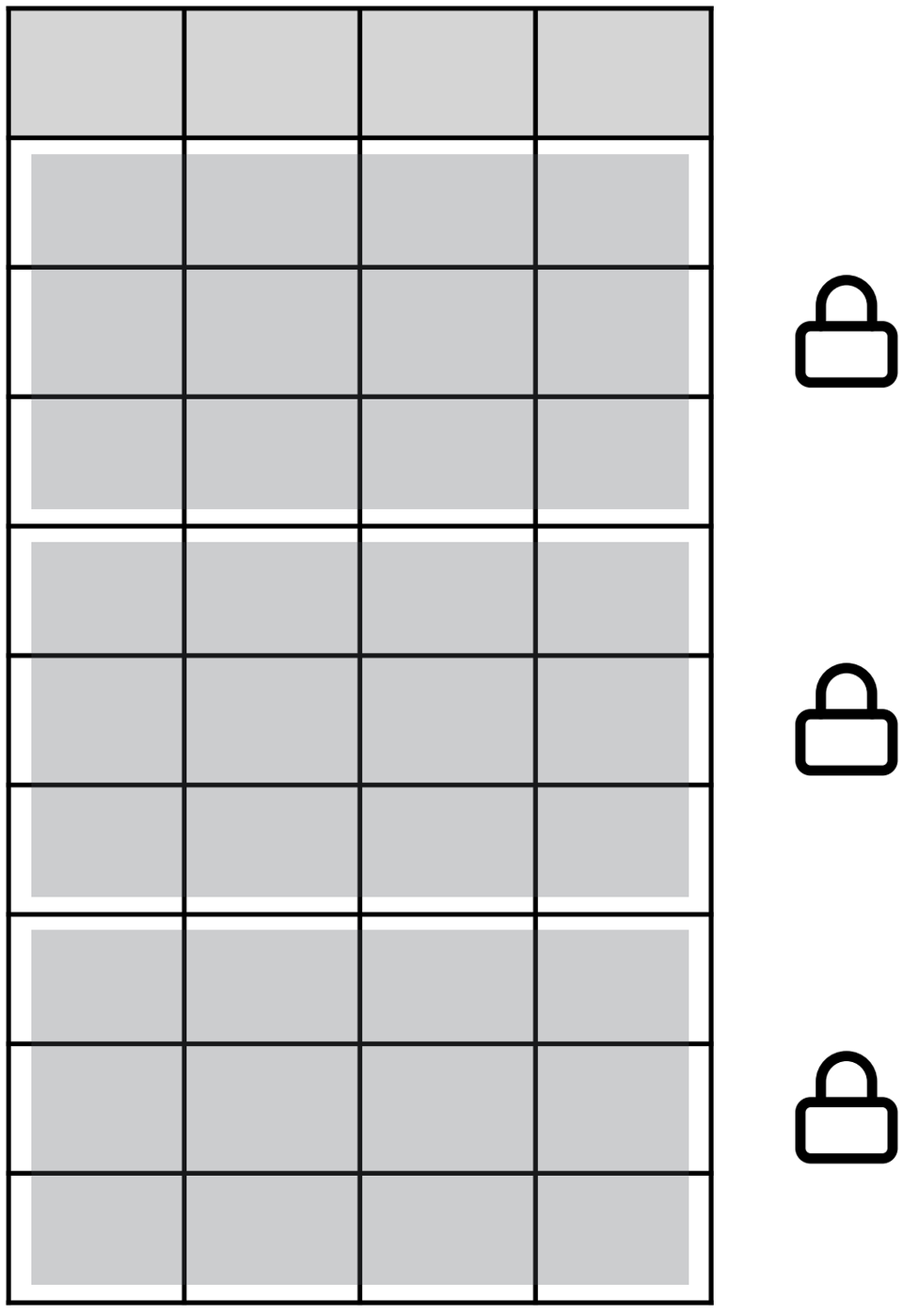
Concurrent access & update



Avoid conflicting

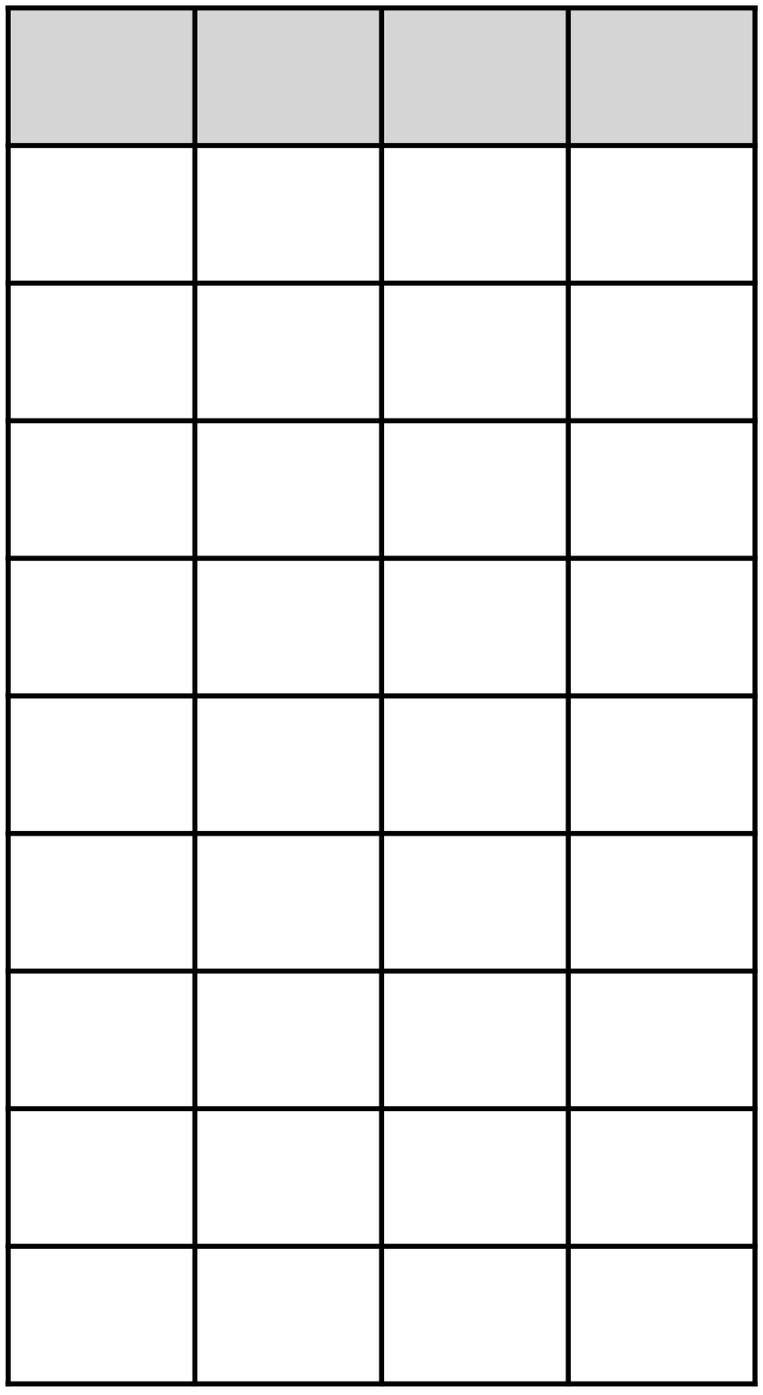
reads & writes

Concurrent access & update



Avoid conflicting reads & writes

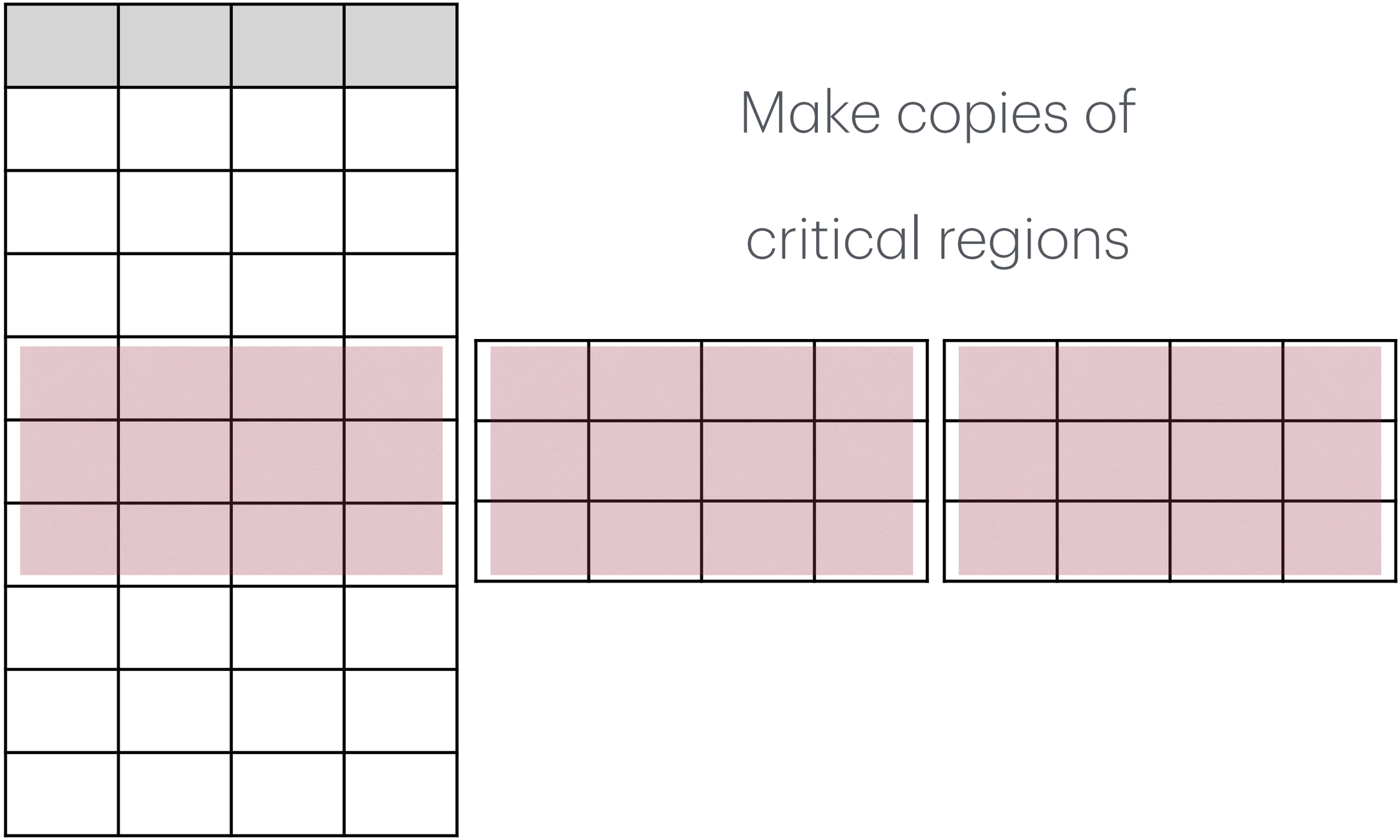
Durable: use replication



Make copies of

critical regions

Durable: use replication



Make copies of

critical regions

SQL: the DB language

pets

name	age	bday	kind
casa	8	2017-01-01	cat
kira	6	2019-09-16	cat
toby	17	2008-02-03	dog
maya	10	2015-11-21	dog

```
CREATE TABLE pets (
  name TEXT,
  age INT,
  bday DATE,
  kind TEXT,
);
```

```
INSERT INTO pets values ("casa", 8, 2017-1-1, "cat");

.tables

.schema pets

DROP TABLE pets;
```

```
.mode box

SELECT name FROM pets;

SELECT age FROM pets;

SELECT * FROM pets;
```

```
SELECT e(columns)
  FROM table
 WHERE condition;
```

```sql
SELECT age, name
  FROM pets
 WHERE 7*age < 70;
```

```
for p in pets:
    if 7*p.age < 70:
        print(p.age, p.name)
```

name	age	bday	kind
casa	8	2017-01-01	cat
kira	6	2019-09-16	cat
toby	17	2008-02-03	dog
maya	10	2015-11-21	dog

```
for p in pets:
    if 7*p.age < 70:
        print(p.age, p.name)
```

```
SELECT age, name
  FROM pets
 WHERE 7*age < 70;
```

```
(age, name) in output <=>
  exists p in pets :
    p.age=age & p.name=name
    & 7*age < 70
```

name	age	bday	kind
casa	8	2017-01-01	cat
kira	6	2019-09-16	cat
toby	17	2008-02-03	dog
maya	10	2015-11-21	dog

```
(10, maya) in output <=>
  exists p in pets :
    p.age=10 & p.name=maya
    & 7*10 < 70
```

Keep rows: `7*age < 70`

name	age	bday	kind
casa	8	2017-01-01	cat
kira	6	2019-09-16	cat
toby	17	2008-02-03	dog
maya	10	2015-11-21	dog

name	age	bday	kind
casa	8	2017-01-01	cat
kira	6	2019-09-16	cat

Keep cols: `age, name`

name	age
casa	8
kira	6

```
SELECT age, name
  FROM pets
 WHERE 7*age < 70;
```

Keep rows: `7*age < 70`

Selection: $\sigma_p(T) = \{t \mid t \in T \wedge p(t)\}$

Keep cols: `age, name`

Projection: $\pi_{x,y,z}(T) = \{(t.x, t.y, t.z) \mid t \in T\}$

Ext. proj.: $\pi_{e(x,y,z)}(T) = \{e(t.x, t.y, t.z) \mid t \in T\}$

name	age	bday	kind
casa	8	2017-01-01	cat
kira	6	2019-09-16	cat
toby	17	2008-02-03	dog
maya	10	2015-11-21	dog

```
SELECT year(bday) + age
  FROM pets
```

Ext. proj.: $\pi_{e(x,y,z)}(T) = \{e(t.x, t.y, t.z) \mid t \in T\}$

Selection: $\sigma_p(T) = \{t \mid t \in T \land p(t)\}$

Ext. proj.: $\pi_{e(x,y,z)}(T) = \{e(t.x, t.y, t.z) \mid t \in T\}$

Union: $T_1 \cup T_2 = \{t \mid t \in T_1 \lor r \in T_2\}$

Intersect: $T_1 \cap T_2 = \{t \mid t \in T_1 \land r \in T_2\}$

Aggregate: $\gamma_{F(x)}(T) = F(T.x)$

Bag v.s. Set

name	age	bday	kind
casa	8	2017-01-01	cat
kira	6	2019-09-16	cat
toby	17	2008-02-03	dog
maya	10	2015-11-21	dog

```
SELECT count(kind) FROM pets;
```

Logistics