Data

Databases

Univ.AI

Regular Heading

Founded by Harvard and UCLA faculty, a true online alternative to the world's top institutions for AI & ML. Meet the founding professors.

Founded by Harvard and UCLA faculty, a true online alternative to the world's top institutions for AI & ML. Meet the founding professors. founding professors

Tabular Data And Datatypes

Tables

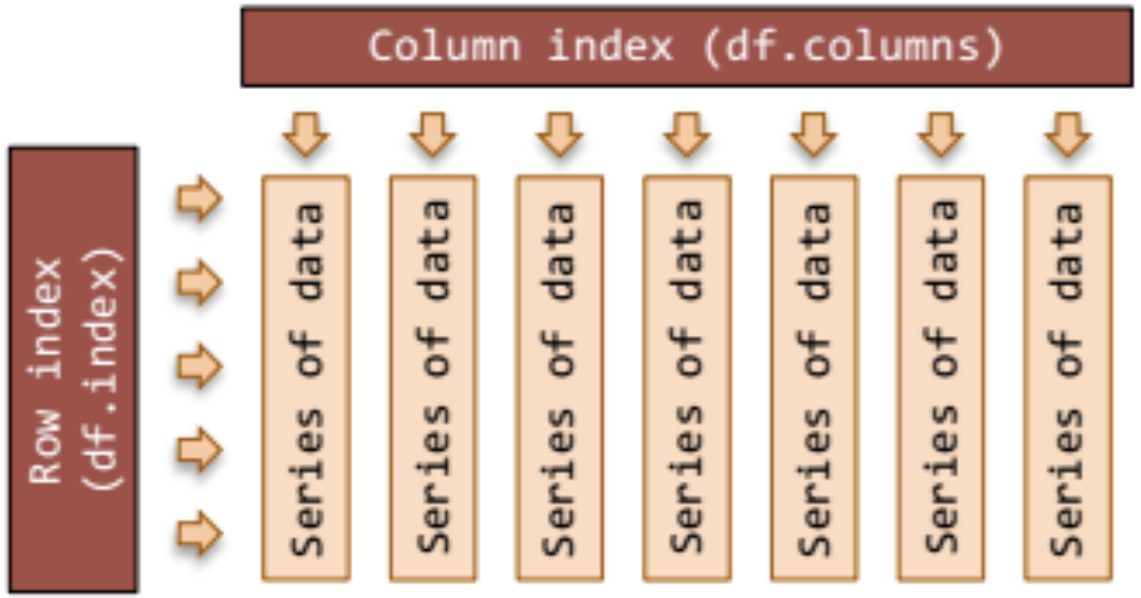
Items

item index	quality	type	price
0	High	Toy	20
1	High	Book	5
2	Medium	Craft	12
3	Low	Craft	10

Customers

customer index	item index	quantity	total
0	0	5	100
1	0	3	60
2	2	4	48
3	3	10	100

DataFrame object: a two-dimensional table of data with column and row indexes. The columns are made up of pandas Series objects.

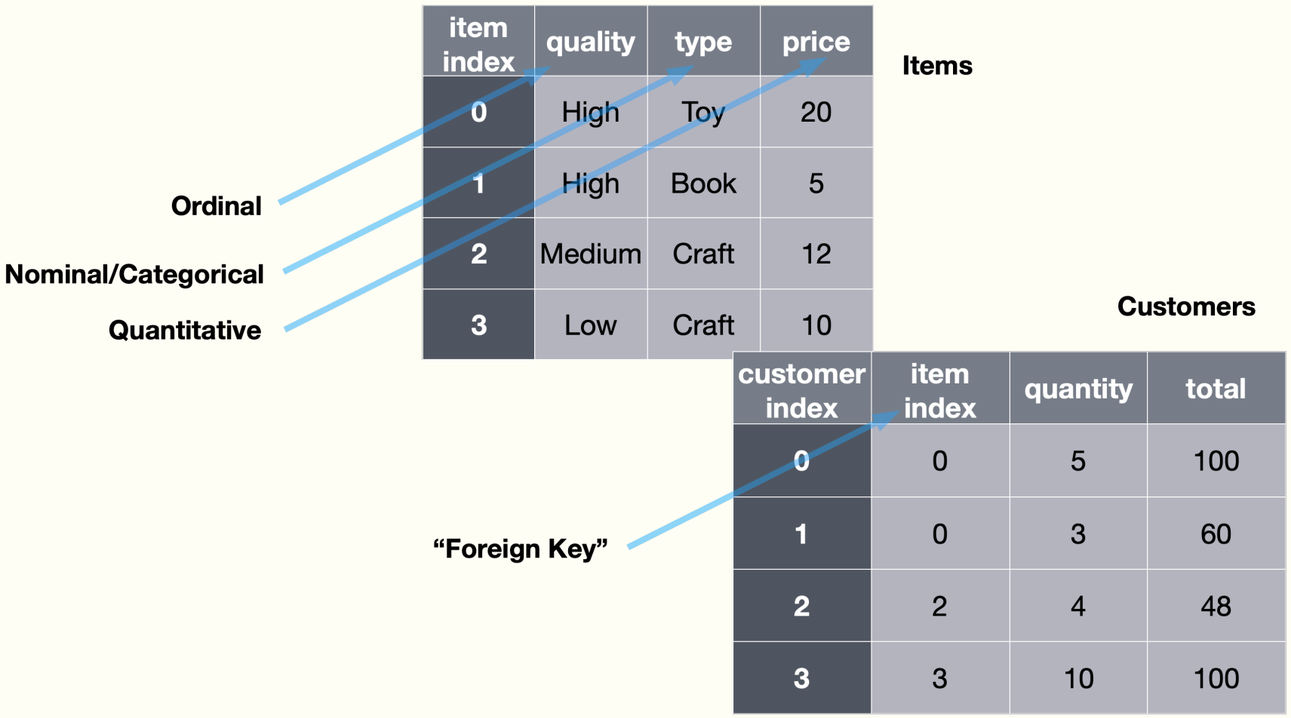


Pandas

Pandas Example

	id	first_name	last_name	middle_name	party
	Filter	Filter	Filter	Filter	Filter
1	16	Mike	Huckabee		R
2	20	Barack	Obama		D
3	22	Rudolph	Giuliani		R
4	24	Mike	Gravel		D
5	26	John	Edwards		D
6	29	Bill	Richardson		D
7	30	Duncan	Hunter		R
8	31	Dennis	Kucinich		D
9	32	Ron	Paul		R

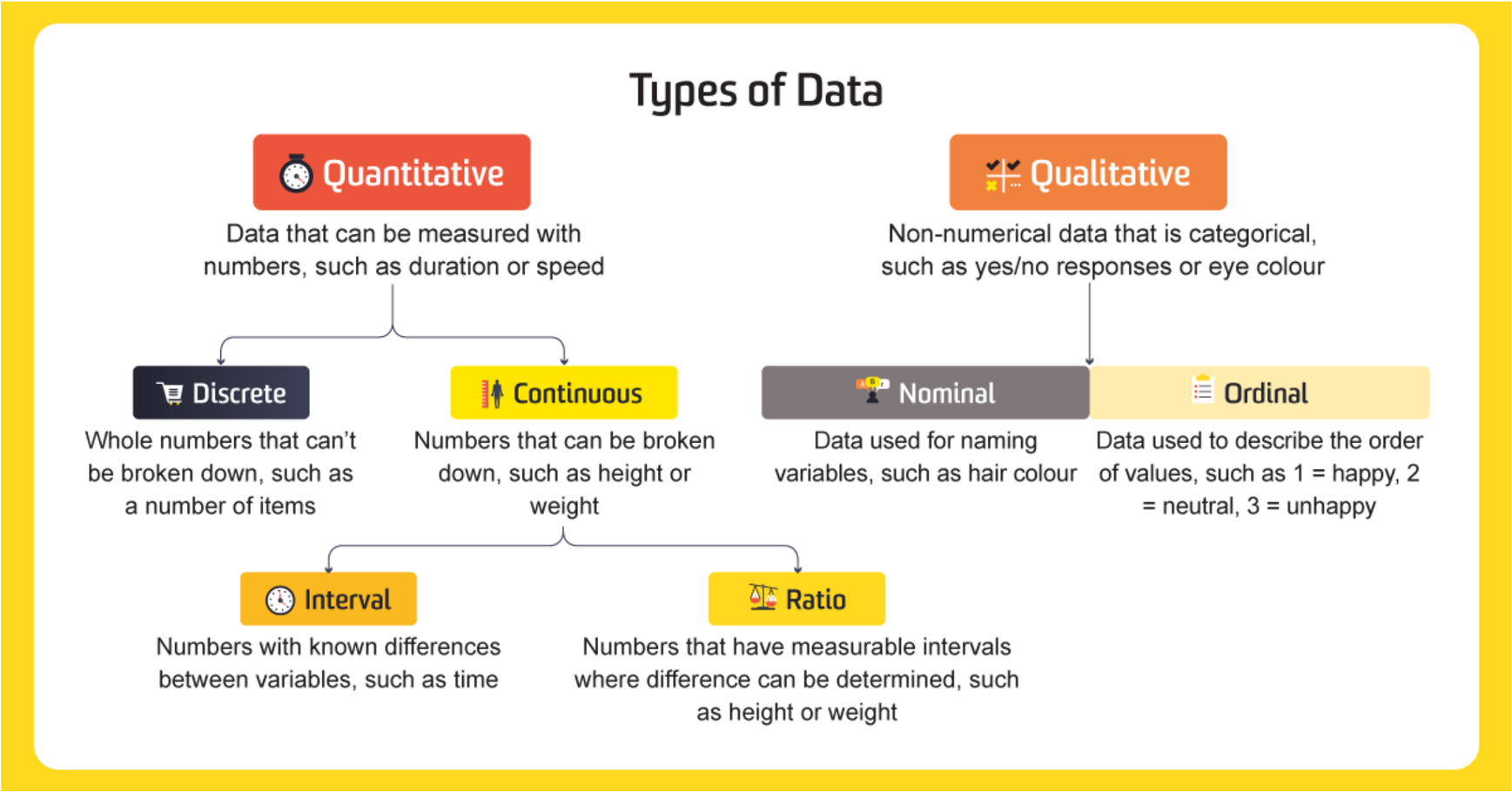
Types

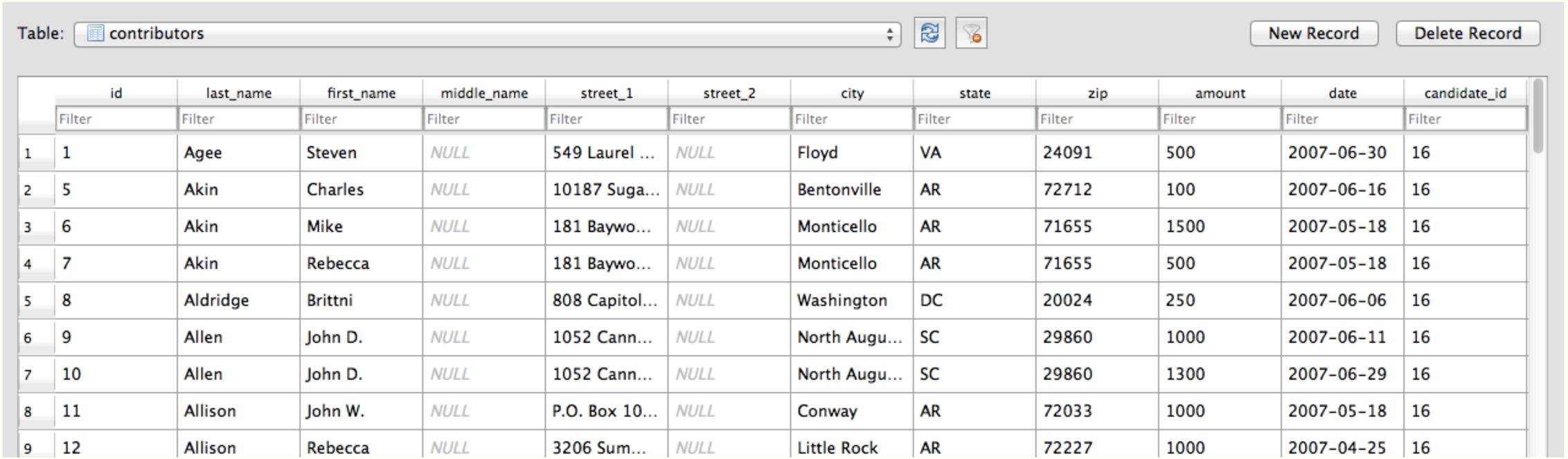


item index	quality	type	price
0	High	Toy	20
1	High	Book	5
2	Medium	Craft	12
3	Low	Craft	10

customer index	item index	quantity	total
0	0	5	100
1	0	3	60
2	2	4	48
3	3	10	100

Types: more detail





	id	last_name	first_name	middle_name	street_1	street_2	city	state	zip	amount	date	candidate_id
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Agee	Steven	NULL	549 Laurel ...	NULL	Floyd	VA	24091	500	2007-06-30	16
2	5	Akin	Charles	NULL	10187 Suga...	NULL	Bentonville	AR	72712	100	2007-06-16	16
3	6	Akin	Mike	NULL	181 Baywo...	NULL	Monticello	AR	71655	1500	2007-05-18	16
4	7	Akin	Rebecca	NULL	181 Baywo...	NULL	Monticello	AR	71655	500	2007-05-18	16
5	8	Aldridge	Brittni	NULL	808 Capitol...	NULL	Washington	DC	20024	250	2007-06-06	16
6	9	Allen	John D.	NULL	1052 Cann...	NULL	North Augu...	SC	29860	1000	2007-06-11	16
7	10	Allen	John D.	NULL	1052 Cann...	NULL	North Augu...	SC	29860	1300	2007-06-29	16
8	11	Allison	John W.	NULL	P.O. Box 10...	NULL	Conway	AR	72033	1000	2007-05-18	16
9	12	Allison	Rebecca	NULL	3206 Sum...	NULL	Little Rock	AR	72227	1000	2007-04-25	16

SQL Typing


Univ.AI

Representation of Data

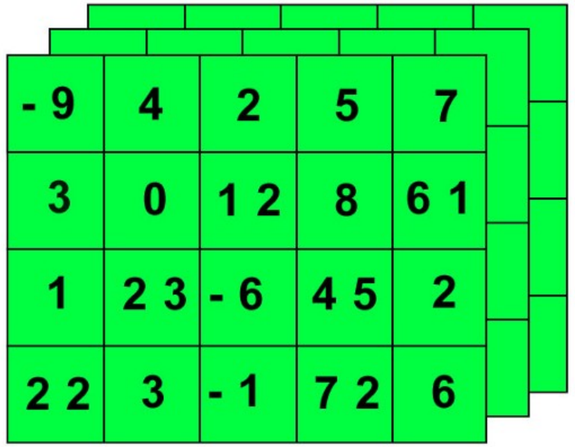
	id	first_name	last_name	middle_name	party
	Filter	Filter	Filter	Filter	Filter
1	16	Mike	Huckabee		R
2	20	Barack	Obama		D
3	22	Rudolph	Giuliani		R
4	24	Mike	Gravel		D
5	26	John	Edwards		D
6	29	Bill	Richardson		D
7	30	Duncan	Hunter		R
8	31	Dennis	Kucinich		D
9	32	Ron	Paul		R

- 9	4	2	5	7
3	0	1 2	8	6 1
1	2 3	- 6	4 5	2
2 2	3	- 1	7 2	6

Images

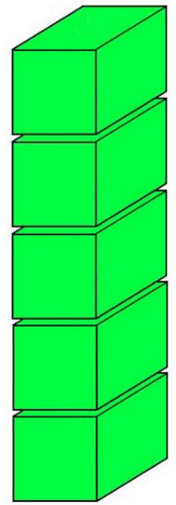
(sample_size, height, width, color_depth)

From One Image



To Many Images

DOVE + NOT DOVE SET

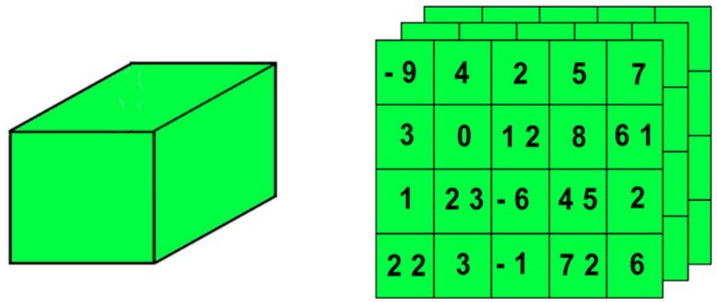


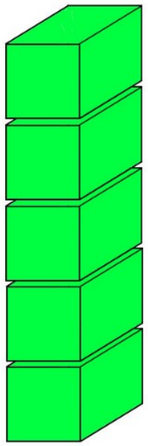
1D TENSOR/
VECTOR

2D TENSOR /
MATRIX

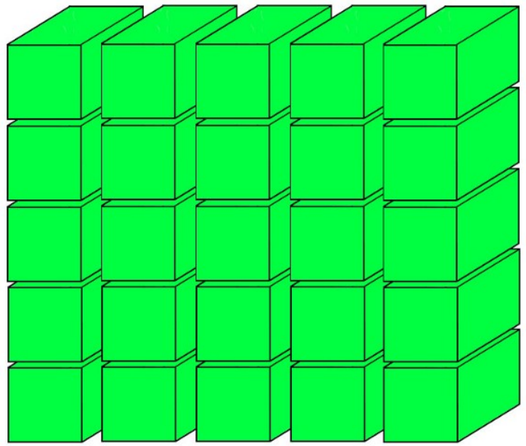
- 9	4	2	5	7
3	0	1 2	8	6 1
1	2 3	- 6	4 5	2
2 2	3	- 1	7 2	6

3D TENSOR/
CUBE





4D TENSOR
VECTOR OF CUBES



5D TENSOR
MATRIX OF CUBES

Text Data

We can store text data in a 3D tensor too. Let's take a look at tweets.

Tweets are 140 characters. Twitter uses the UTF-8 standard, which allows for millions of types of characters, but we are realistically only interested in the first 128 characters, as they are the same as basic ASCII. A single tweet could be encapsulated as a 2D vector of shape (140,128).

If we downloaded 1 million Donald Trump tweets (I think he tweeted that much last week alone) we would store that as 3D tensor of shape:

```
(number_of_tweets_captured, tweet, character)
```

That means our Donald Trump tweet collection would look like this:

```
(1000000,140,128)
```

Video Data

A 5D tensor can store video data. In TensorFlow video data is encoded as:

```
(sample_size, frames, width, height, color_depth)
```

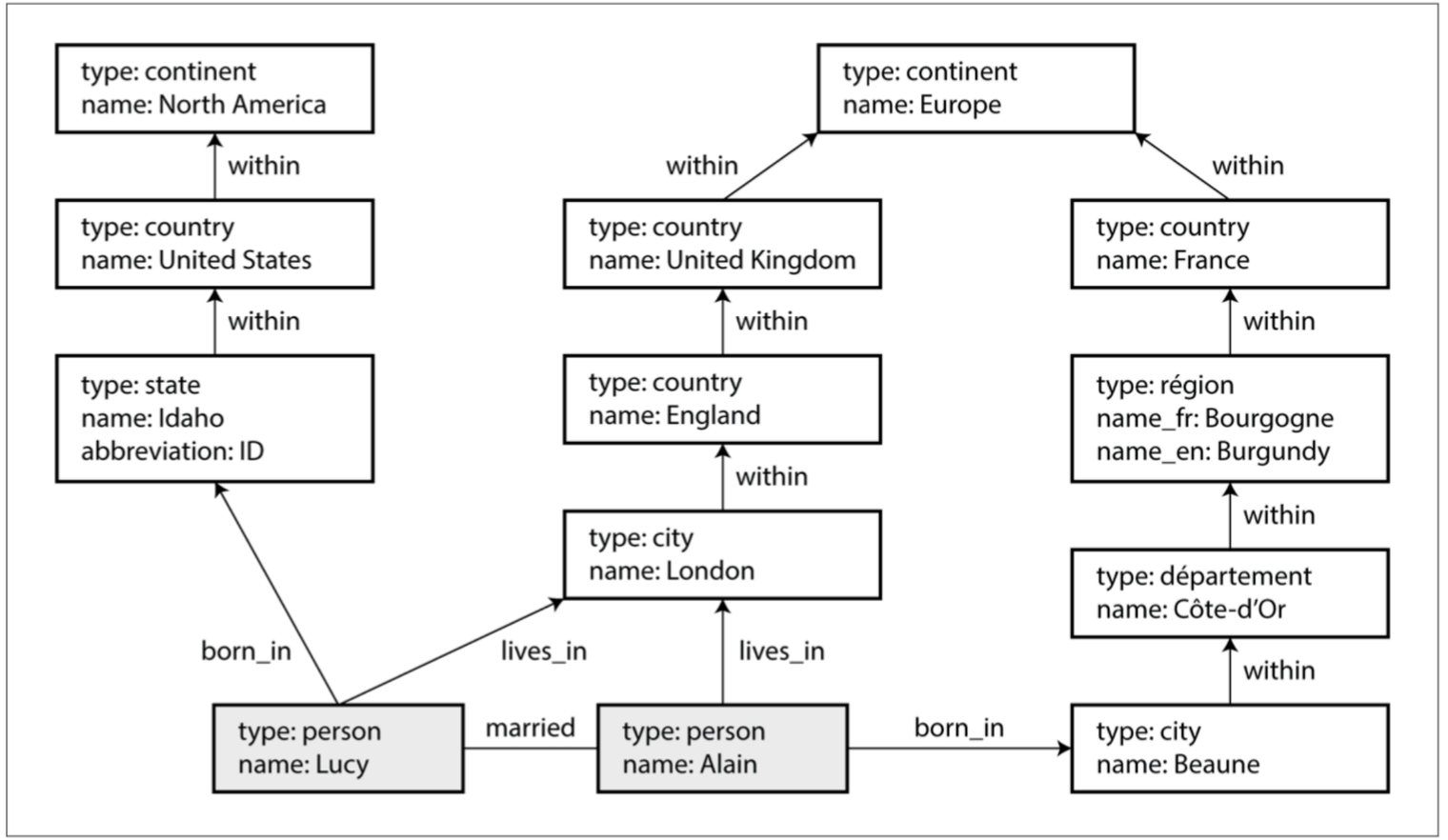
If we took a five minute video (60 seconds x 5 = 300), at 1080p HD, which is 1920 pixels x 1080 pixels, at 15 sampled frames per second (which gives us 300 seconds x 15 = 4500), with a color depth of 3, we would store that a 4D tensor that looks like this:

```
(4500,1920,1080,3)
```

The fifth field in the tensor comes into play when we have multiple videos in our video set. So if we had 10 videos just like that top one, we would have a 5D tensor of shape:

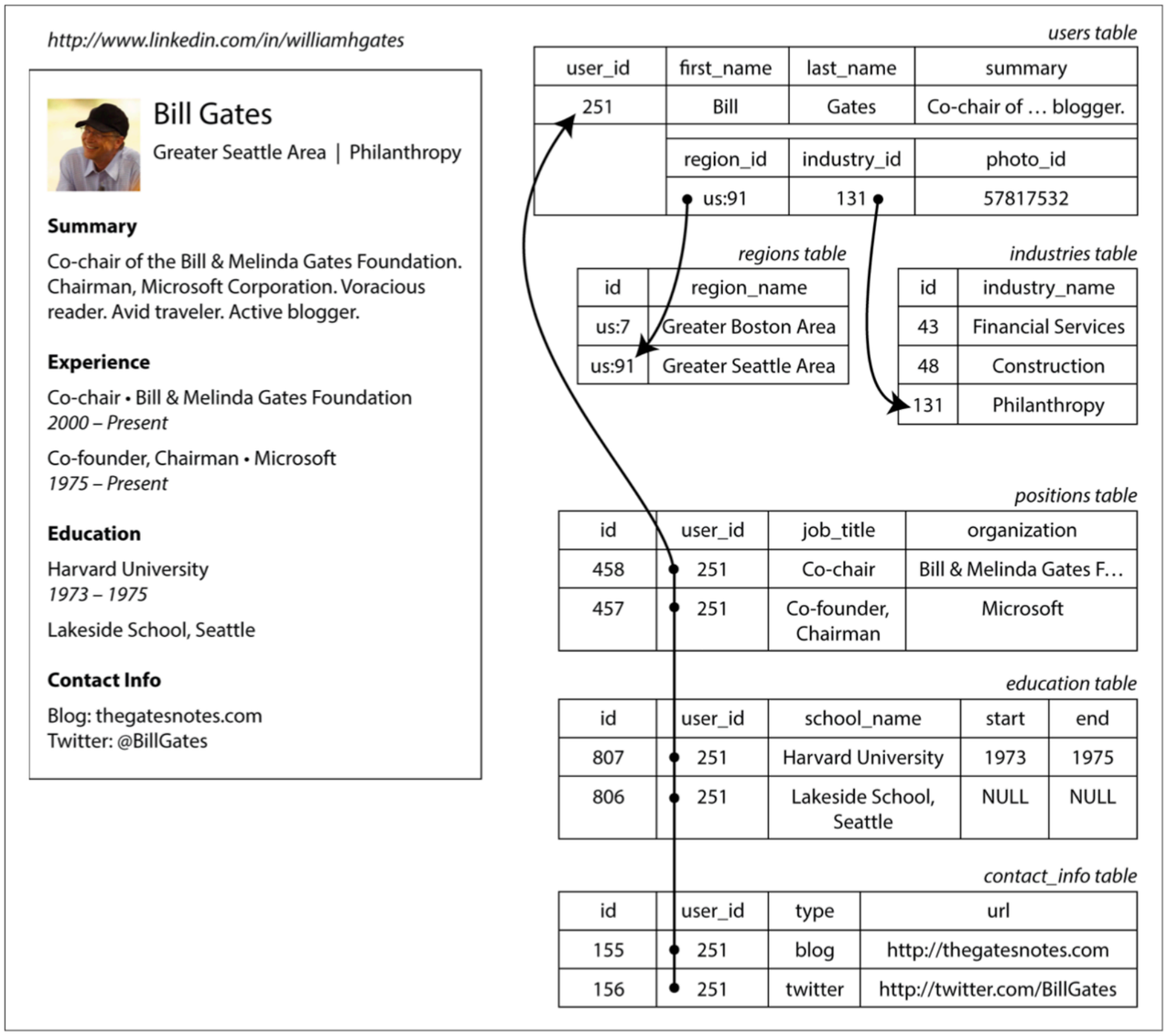
```
(10,4500,1920,1080,3)
```

Graph Data


type: continent
name: North America
within
type: country
name: United States
within
type: state
name: Idaho
abbreviation: ID
type: continent
name: Europe
within
within
type: country
name: United Kingdom
type: country
name: France
within
within
type: country
name: England
type: région
name_fr: Bourgogne
name_en: Burgundy
within
within
type: city
name: London
type: département
name: Côte-d'Or
within
born_in
lives_in
lives_in
type: person
name: Lucy
married
type: person
name: Alain
born_in
type: city
name: Beaune

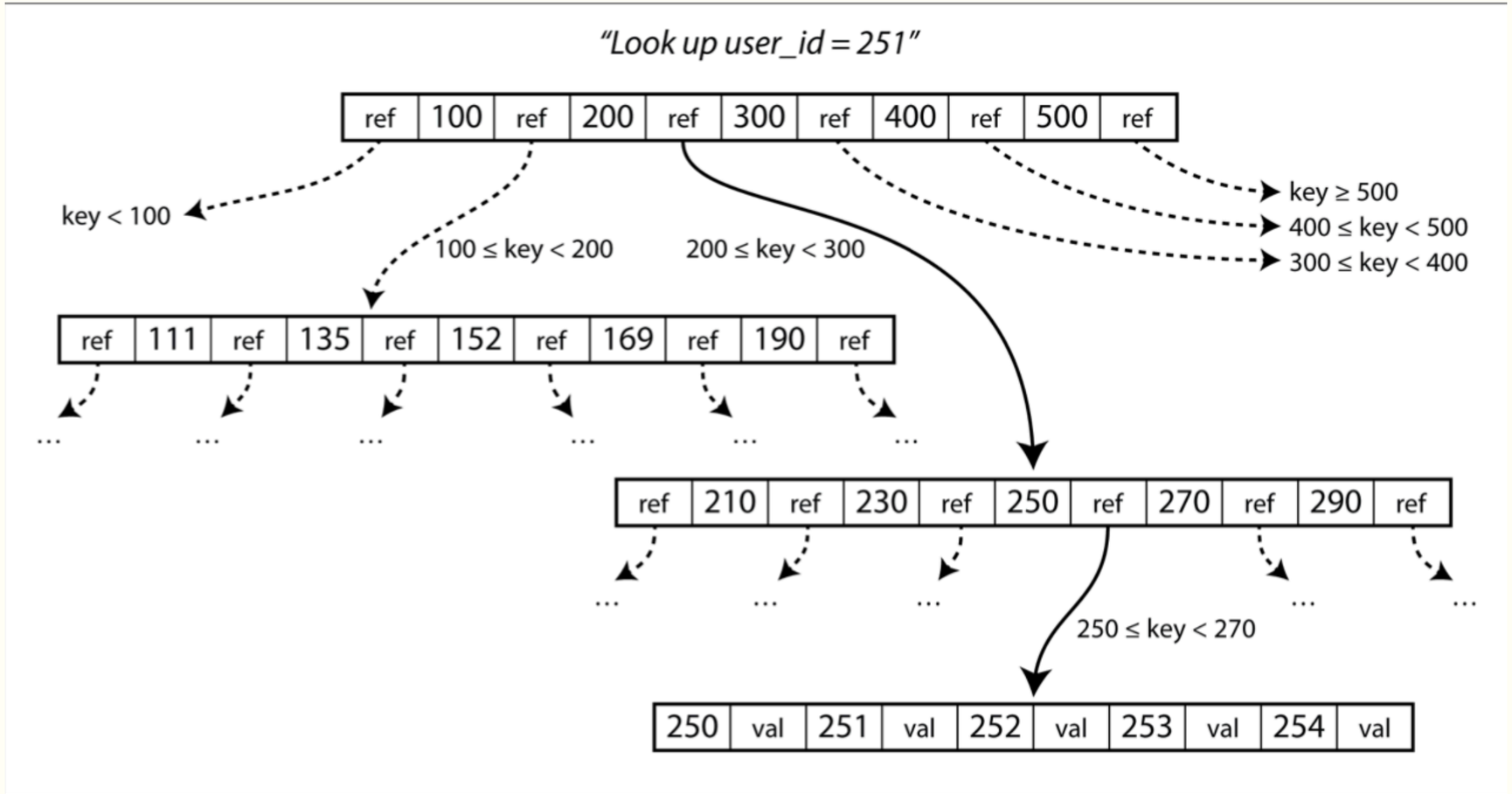

Univ.AI

Databases and Indexes



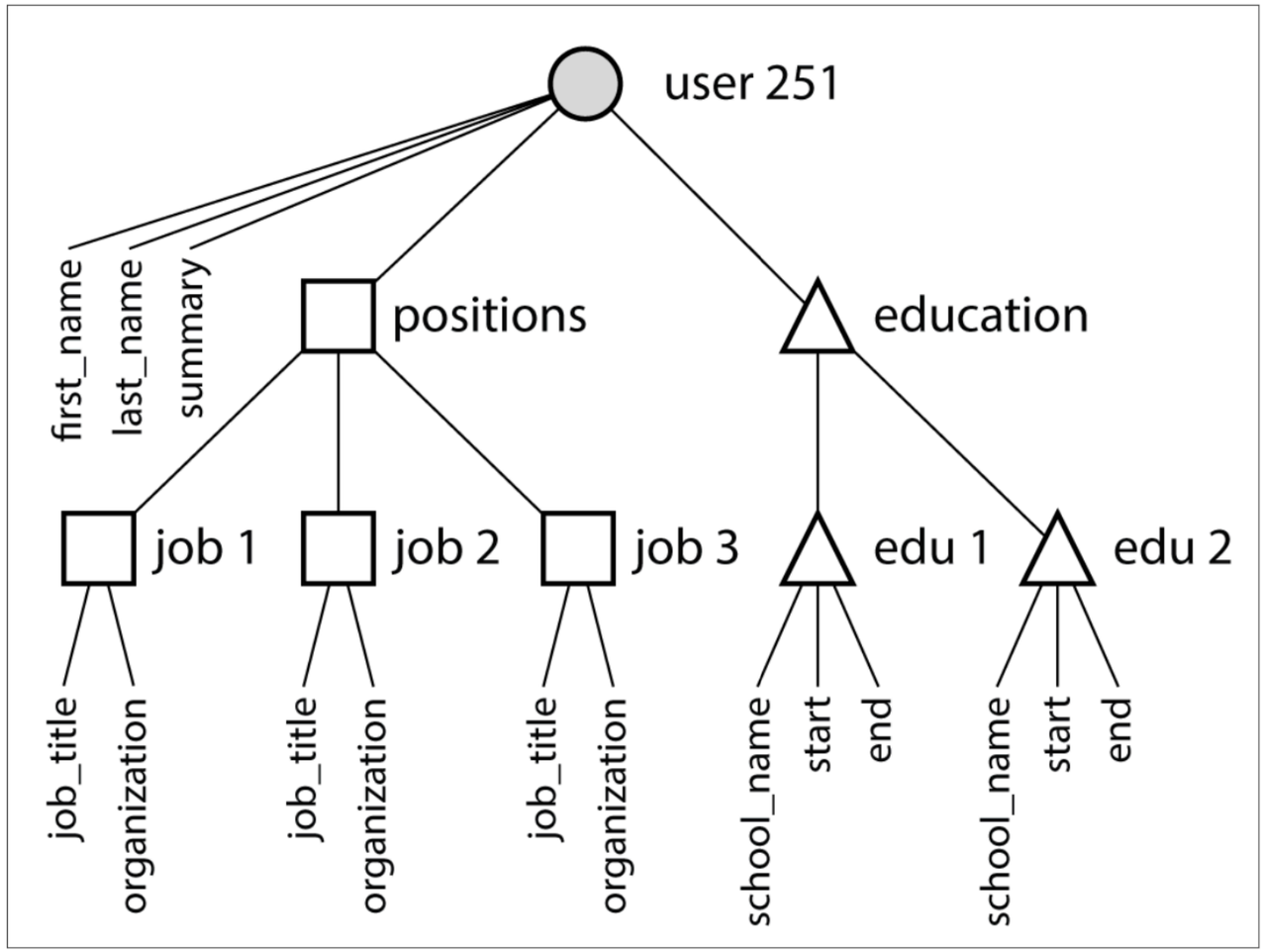
Relational Database

How to get stuff?

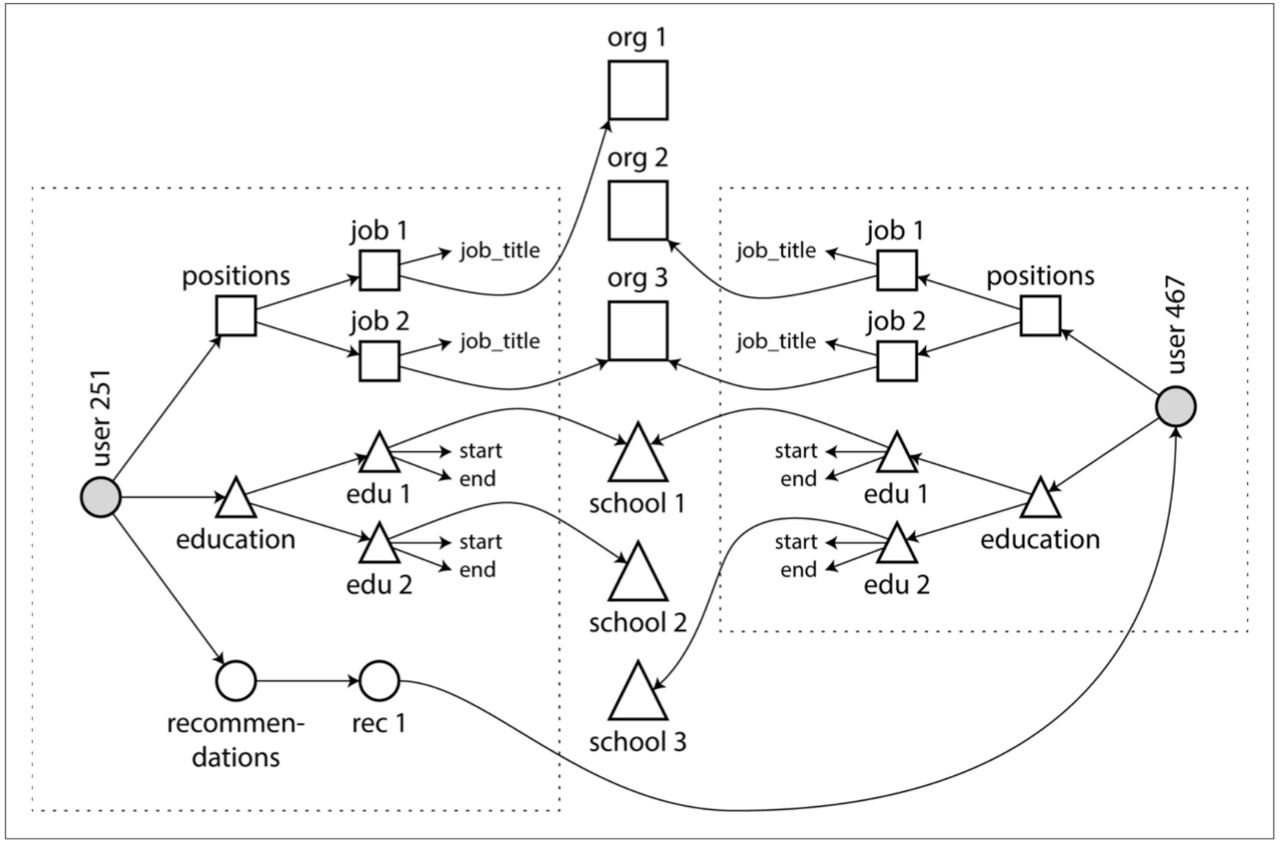


Document Database


```
{
  "user_id":     251,
  "first_name":  "Bill",
  "last_name":   "Gates",
  "summary":     "Co-chair of the Bill & Melinda Gates... Active blogger.",
  "region_id":   "us:91",
  "industry_id": 131,
  "photo_url":   "/p/7/000/253/05b/308dd6e.jpg",
  "positions": [
    {"job_title": "Co-chair", "organization": "Bill & Melinda Gates Foundation"},
    {"job_title": "Co-founder, Chairman", "organization": "Microsoft"}
  ],
  "education": [
    {"school_name": "Harvard University",       "start": 1973, "end": 1975},
    {"school_name": "Lakeside School, Seattle", "start": null, "end": null}
  ],
```

Graph or Relational?

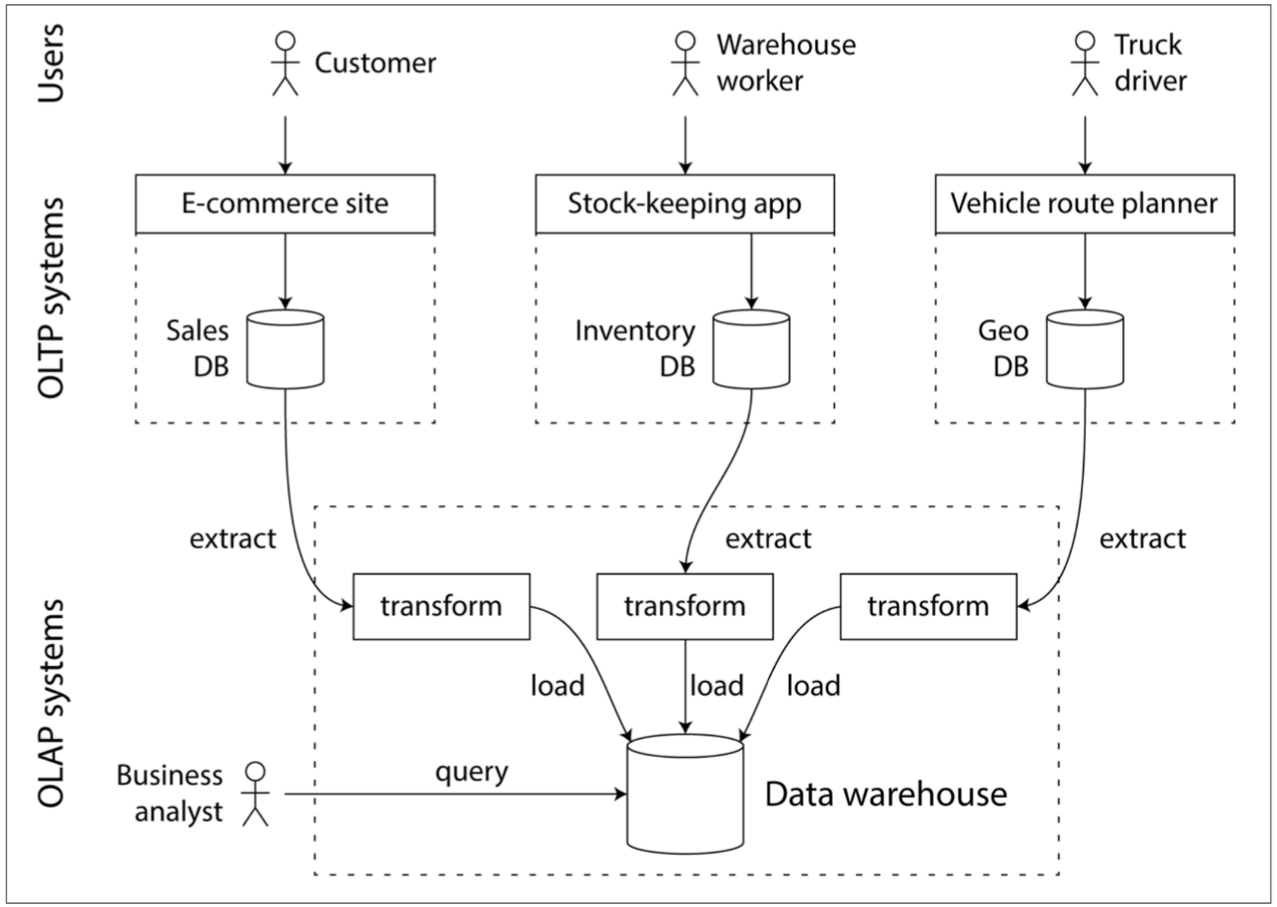


What kind of data access do you need?

- **relational**: pandas, SQL: Postgres, sqlite, Hbase, VoltDB
- **document oriented**: MongoDB, CouchDB
- **key-value**: Riak, Redis, Memcached
- **graph oriented**: Neo4J

What kind of data access do you need?

- in memory
- on disk
- on cluster


Users
Customer
Warehouse worker
Truck driver
OLTP systems
E-commerce site
Stock-keeping app
Vehicle route planner
Sales DB
Inventory DB
Geo DB
extract
extract
extract
OLAP systems
transform
transform
transform
load
load
load
Business analyst
query
Data warehouse

OLTP vs OLAP

Normalization to ....