

transform by example  
extract by example  
learn by example  
fix by example  
search by example  
analyze by example  
generate by example  
debug by example  
predict by example  
query by example  
plot by example  
transform by example  
**x by example**  
extract by example  
learn by example  
fix by example  
search by example  
analyze by example  
generate by example  
debug by example  
predict by example  
query by example  
plot by example  
transform by example  
extract by example  
learn by example  
fix by example  
search by example  
analyze by example  
generate by example  
debug by example

Azza Abouzied, NYU Abu Dhabi  
[azza@nyu.edu](mailto:azza@nyu.edu)

Started  
PhD in  
Database  
research

A tutorial on Visualization  
at VLDB by Joe Hellerstein  
& Jeff Heer

Retargeting my research at the 99%

Joined NYU

HadoopDB

InvisibleLoading

Hadapt 

DataPlay

SEER

Qetch 

WhyFlow Synner

Texture

PackageBuilder 



2008

@Yale, New Haven



2009



2012

@UC Berkeley



2013

@Happy Island, Abu Dhabi



2018

About me

Projects, Places & Events

How would you describe “furniture”?

---

A thought experiment

If you thought of an example,  
you are not alone

Prototyping

Exemplar-based reasoning

Recognition-primed decision making

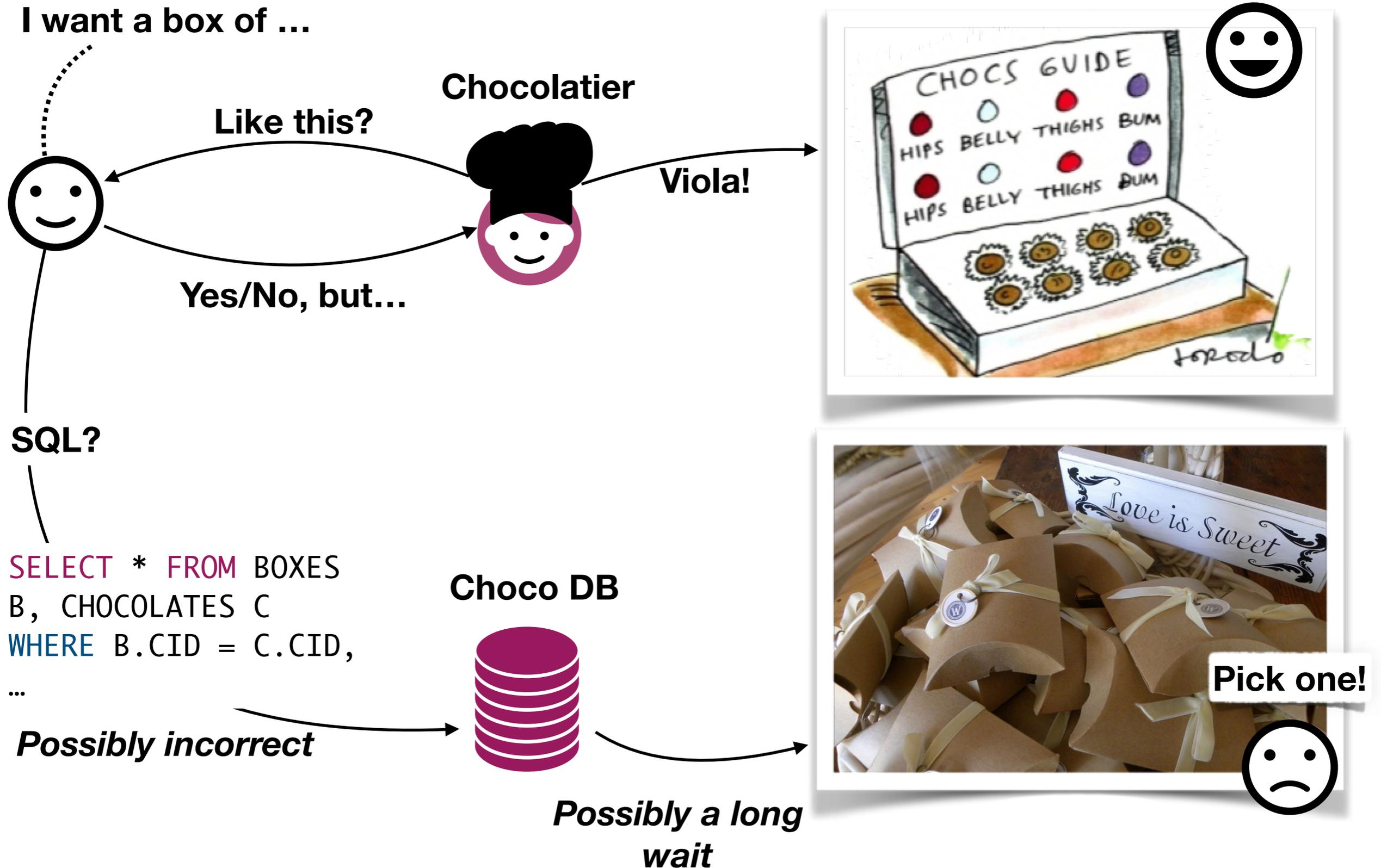


How can we improve how we communicate with  
our data tools?

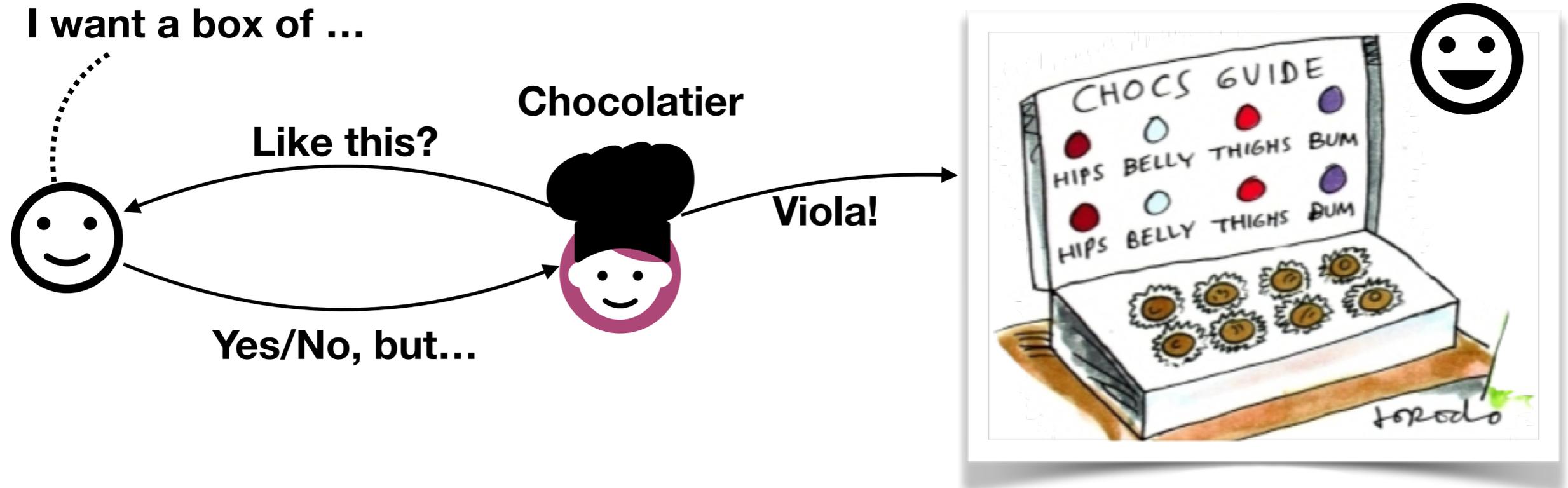
---

What are example-driven interfaces?

Suppose you want to buy a box of chocolates



Suppose you want to buy a box of chocolates



**EDIs** mimic human interactions: they allow examples of (un)expected behavior, which can be *underspecified* or *ambiguous*, and work towards a precise specification of behavior through further human interaction such as requesting more examples, counter-examples, partial specifications, constraints, etc.

**EDIs** can support a variety of data tasks: extraction, transformation, visualization, querying, analysis, debugging, generation, etc.

**What**  
are EDIs?

**Why**  
now?

**How**  
to build them?

**When**  
does it work?

**Where**  
do we go from here?

# Why is now the right time for example-driven interfaces?

---

A confluence of many maturing research areas

# Query by Example

by MOSHÉ M. ZLOOF

IBM T. J. Watson Research Center  
Yorktown Heights, New York

## INTRODUCTION

In the last few years we have witnessed a trend to appeal to the non-professional user who has little or virtually no computer or mathematical background.

The 'Query by Example' Language is an attempt in that direction. It operates on a relational Model of data as was introduced by Codd [1-5].

In this paper we deal only with normalized relations [1]. A relation is normalized if each of its domains is simple, i.e., no domain is itself a relation.

A normalized relation can be viewed as a table of  $n$  columns and a varying number of rows as illustrated in Figure 1. Three properties of normalized relations are noteworthy to mention:

1. ALL rows of the table are distinct.
2. The ordering of the rows is immaterial.
3. The ordering of the columns is immaterial provided each has a distinct name.

| EMP | NAME     | SALARY | MANAGER | DEPARTMENT |
|-----|----------|--------|---------|------------|
|     | ANDERSON | 8K     | SMITH   | TOY        |
|     | MORGAN   | 10K    | LEE     | COSMETICS  |
|     | .        |        |         |            |
|     | .        |        |         |            |
|     | .        |        |         |            |

trations of queries and their answers, each illustration followed by a discussion to point out major features. The illustrations get progressively more complex until the whole scope of the Language is covered. In so doing, a user dealing with "simple" queries needs to study the system *only* to that point of complexity which is compatible with the level of sophistication required within the domain of those queries.

Furthermore, although the introduction of the concepts through illustrative examples reduces somewhat from the rigor of mathematical formulation through definitions, it is—in our opinion—more appealing to the casual user, which is one of the major aspects of Query by Example.

Most of the queries are drawn from the following tables (relations), which are part of a department store data base.

EMP (NAME, SAL, MGR, DEPT)

SALES (DEPT, ITEM)

SUPPLY (SUPPLIER, ITEM)

TYPE (ITEM, COLOR, SIZE)

—The EMP Table specifies the name, salary, manager and department of each employee.

—The SALES Table is a listing of the items sold by departments.

—The SUPPLY Table is a listing of the items supplied by suppliers.

—The TYPE Table describes each item by color and size.

At this point we are assuming that these tables are made available to the user upon calling them by name. In a sub-

# 1 Program Synthesis

## Circuit Synthesis

1957  
Alonzo Church. *Application of recursive arithmetic to the problem of circuit synthesis*. In Summaries of Talks Presented at the Summer Institute for Symbolic Logic, Cornell University.

## Solver-backed Synthesis

2008  
Armando Solar-Lezama. *Program Synthesis by Sketching*. PhD Thesis. UC-Berkeley

## PBE is mainstream: FlashFill in Excel

2017  
S. Gulwani, O. Polozov and R. Singh. *Program Synthesis*. Foundations and Trends in Programming Languages, vol. 4, no. 1-2

# 2 Mixed Initiative User Interfaces



1997

1999

Eric Horvitz. *Principles of mixed-initiative user interfaces*. CHI '99. ACM



2007



2015

Jeffrey Heer, Joseph M. Hellerstein, Sean Kandel. *Predictive Interaction for Data Transformation*. CIDR'15

## 3 Active Learning

Learning Theory: Membership Questions, Teaching Dimension, ...

1987  
Dana Angluin. *Learning regular sets from queries and counterexamples*. Inf. Comput. 75, 2

Crowd-sourced & Function Labeling

2017  snorkel  
Christopher Ré. *Software 2.0 and Snorkel: Beyond Hand-Labeled Data*. KDD '18.

## 4 Causality & Explanations

Lineage & Provenance in Databases

1748  
**David Hume:** *We may define a cause to be an object followed by another, and where all the objects, similar to the first, are followed by objects similar to the second. Or, in other words, where, if the first object had not been, the second never had existed.*

2000

2005  
Joseph Y. Halpern & Judea Pearl. *Causes and explanations: A structural-model approach. Part I: Causes*. British Journal for the Philosophy of Science 56 (4)

Causality & Explanations

2010  
Meliou et al. *Why so? or Why no? Functional Causality for Explaining Query Answers*. MUD

5

# Automatic, Declarative, Data Visualization



2003

2011

1987  
Jock Mackinlay. *Automating the Design of Graphical Presentations of Relational Information*. ACM Transactions on Graphics 5(2).

2010  
Heer & Bostock. *Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design*. CHI

2018  
Moritz et al. *Formalizing Visualization Design Knowledge as Constraints: Actionable and Extensible Models in Draco*. InfoVis

1984  
Cleveland & McGill. *Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods*. Journal of the American Statistical Association 79(387)

2016  
Satyanarayan et al. *Vega-Lite: A Grammar of Interactive Graphics*. InfoVis

# How to build example-driven data tools?

---

An example recipe

# The Dimensions of Example-Driven Interfaces

*From Sumit Gulwani's Cookbook - Dimensions of Program Synthesis*

## Intent Specification

Inputs, Outputs  
Positive & Negative  
Program Sketch

## Search Space

Scope & define your tasks  
Create a syntactic bias  
DSL  
(Invertible) Operators  
Templates

## Search Techniques

Version Space Algebras  
SMT-guided search  
Brute-force search

## Ambiguity Resolution

Ranking  
Distinguishing Inputs  
Exposing Semantics

# When do example-driven data tools work?

---

A few illustrative examples from my research

visualize

some in student.takes.course.area == "Systems"

QUERY TREE

Toggle Quantifier Toggle Coverage Delete

VISUALIZATIONS

Clear In/Out Fix my query

ANSWERS

| want out?                | keep in?                 | student.id | student.name      | student.dept | student.year | student.takes.grade | student.takes.course.id | student.takes.c |
|--------------------------|--------------------------|------------|-------------------|--------------|--------------|---------------------|-------------------------|-----------------|
| <input type="checkbox"/> | <input type="checkbox"/> | ST89       | Darren Preston    | MATH         | 1            | A                   | CS11                    | Systems         |
|                          |                          |            |                   |              |              | A                   | CS12                    | Systems         |
|                          |                          |            |                   |              |              | A                   | CS16                    | Programming     |
|                          |                          |            |                   |              |              | A                   | CS18                    | Seminar         |
| <input type="checkbox"/> | <input type="checkbox"/> | ST142      | Tiffany D'Ascenzo | CS           | 1            | A                   | CS11                    | Systems         |
|                          |                          |            |                   |              |              | A                   | CS12                    | Systems         |
|                          |                          |            |                   |              |              | A                   | CS14                    | Theory          |

NON-ANSWERS

| want in?                 | keep out?                | student.id | student.name  | student.dept | student.year | student.takes.grade | student.takes.course.id | student.takes.c |
|--------------------------|--------------------------|------------|---------------|--------------|--------------|---------------------|-------------------------|-----------------|
| <input type="checkbox"/> | <input type="checkbox"/> | ST116      | John Gross    | ECON         | 1            | A                   | CS11                    | Systems         |
|                          |                          |            |               |              |              | A                   | CS14                    | Theory          |
|                          |                          |            |               |              |              | B                   | CS16                    | Programming     |
|                          |                          |            |               |              |              | A                   | CS18                    | Seminar         |
| <input type="checkbox"/> | <input type="checkbox"/> | ST298      | Brenda DeMuth | CS           | 2            | A                   | CS11                    | Systems         |
|                          |                          |            |               |              |              | B                   | CS15                    | Programming     |
|                          |                          |            |               |              |              | A                   | CS16                    | Programming     |

β- preview

# DataPlay

Example-driven database querying

Abouzied et al. *DataPlay: Interactive Tweaking and Example-driven Correction of Graphical Database Queries*. UIST 2012

Abouzied et al. *Learning and verifying quantified boolean queries by example*. PODS 2013

Dataset: data/crimeStatements/ Search: Search here

**FBI Announces Executive Appointments**  
 Washington, D.C.  
 July 30, 2015

Positive Example? Negative Example?

FBI National Press Office  
 (202) 324-3891  
 Director James B. Comey announced today the following leadership appointments:

Kevin Perkins, Special Agent in Charge, FBI Baltimore Division

After three years of dedicated service as the associate deputy director, Kevin Perkins will become the special agent in charge of the Baltimore Division and succeed Stephen Vogt, who is retiring following a 25-year career with the FBI.

Mr. Perkins entered on duty as a special agent in January 1986. He previously served in the Kansas City, Philadelphia, and Baltimore Divisions in a variety of investigative and leadership positions. Mr. Perkins previously served as the special agent in charge in Baltimore from January 2004 to February 2006.

Mr. Perkins' executive leadership positions included serving as assistant director for the Criminal Investigative Division, the Inspection Division, and the Finance Division, where he also served as chief financial officer of the FBI.

As associate deputy director, Mr. Perkins is responsible for all aspects of the FBI's budget, human resources, information systems, and administrative functions.

Suggest me some rules!

| Position | Positive Example  | Position | Negative Example |
|----------|-------------------|----------|------------------|
| 127      | February 29, 2016 | 201      | James B. Comey   |
| 61857    | January 1986      |          |                  |

To extract: January 1986, ...

Positive Examples: January 1986 February 29, 2016

To further refine the rules we suggest, select whether the following examples should be extracted or not? As you decide the fate of some examples, we automatically disable other examples that could potentially conflict with your selections so far.

Reset

| Extract the following: | Yes?                             | No?                   |
|------------------------|----------------------------------|-----------------------|
| June 16, 2015          | <input checked="" type="radio"/> | <input type="radio"/> |
| May 2011               | <input type="radio"/>            | <input type="radio"/> |
| February 21, 2016      | <input type="radio"/>            | <input type="radio"/> |
| August 2011            | <input type="radio"/>            | <input type="radio"/> |

Extraction Rules:

- DateTime prebuilt
- "January Feb..." dictionary 0-2 token\_gsp\_range IntegerNumber prebuilt
- "January Feb..." dictionary 0-2 token\_gsp\_range "[0-9]+" regex

Results:

|                         |                     |
|-------------------------|---------------------|
| /FBIPressReleaseAug.txt | "February 29, 2016" |
| /FBIPressReleaseAug.txt | "February 24, 2016" |
| /FBIPressReleaseAug.txt | "February 21, 2016" |
| /FBIPressReleaseAug.txt | "February 21, 2016" |

# SEER

Example-driven data extraction from text

Maeda Hanafi, Azza Abouzied, Laura Chiticariu, and Yunyao Li. *SEER: Learning Information Extraction Rules from User-Specified Examples*. CHI 2017

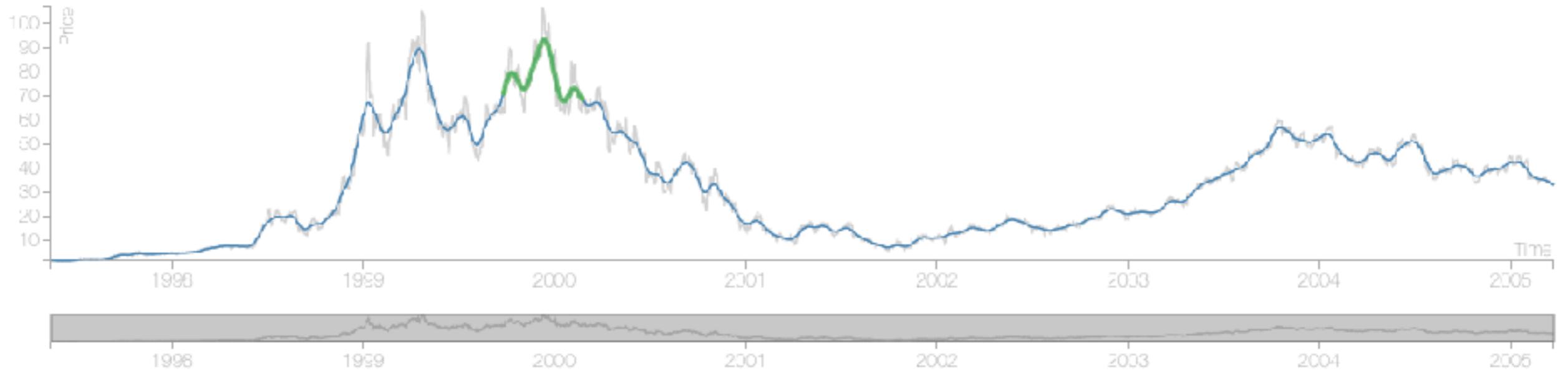
# Qetch

Load new data ▾ ⚙ Settings

Dataset

Stock Prices: AMZN ▾

Smooth iteration:



Query H I H I C C^n ⊙ Clear

Predefined queries ▲ History ▲

Results

| Distance ↓ | Smooth iteration | Time span | Show all ▾ |
|------------|------------------|-----------|------------|
| 0.96       | 8                | 149 days  | Show       |
| 1.41       | 9                | 201 days  | Show       |
| 1.60       | 9                | 220 days  | Show       |
| 1.78       | 6                | 131 days  | Show       |
| 2.14       | 2                | 60 days   | Show       |

## Qetch

Time series querying with hand-drawn sketches

Miro Mannino, Azza Abouzied. *Expressive Time Series Querying with Hand-Drawn Scale-Free Sketches*. CHI 2018 - **Best Paper Award**

# When do example-driven data tools fail?

---

A case-study on debugging data processing pipelines by example

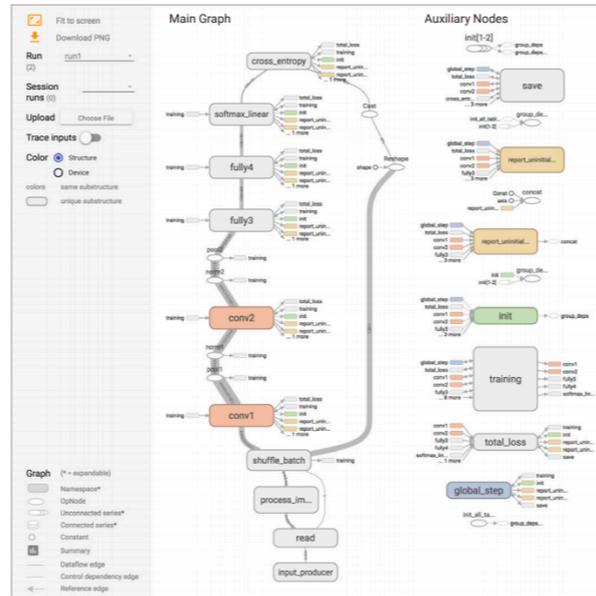
# Where do we go from here?

---

Some parting thoughts on the future of this research

# Visualization Research

Exciting opportunities at the interplay of visualizing program artifacts & data



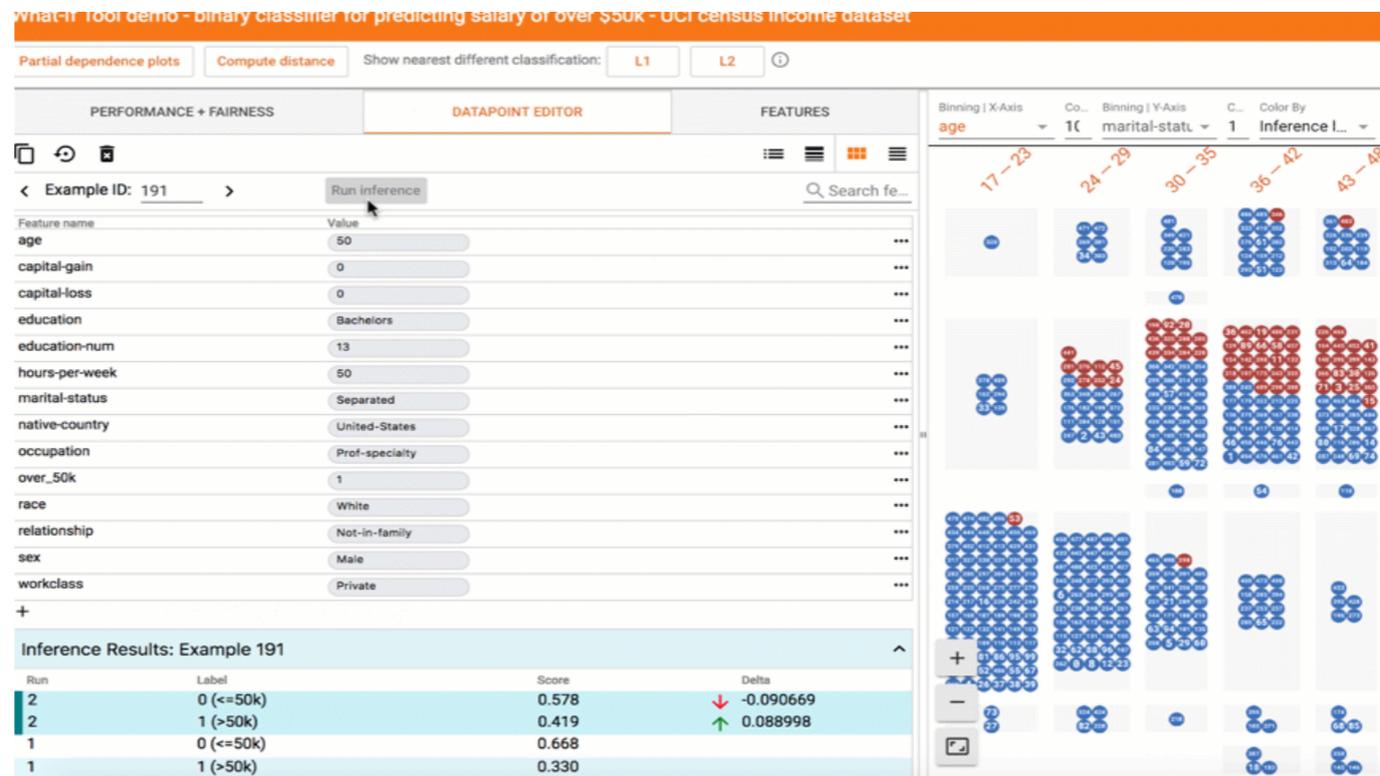
```

33 {
34   "name": "indexed_stocks",
35   "source": "stocks",
36   "transform": [
37     {
38       "type": "lookup",
39       "on": "index", "onKey": "symbol",
40       "keys": ["symbol"], "as": ["index_term"],
41       "default": {"price": 0}
42     }, {
43       "type": "formula",
44       "field": "indexed_price",
45       "expr": "datum.index_term.price > 0 ? (datum.price - datum.index_term.price) : datum.index_term.price"
46     }
47   ]
48 }

```

Wongsuphasawat, Kanit, Daniel Smilkov, James Wexler, Jimbo Wilson, Dandelion Mané, Doug Fritz, Dilip Krishnan, Fernanda B. Viégas, and Martin Wattenberg. *Visualizing dataflow graphs of deep learning models in TensorFlow*. IEEE transactions on visualization and computer graphics 24, no. 1 (2018): 1-12.

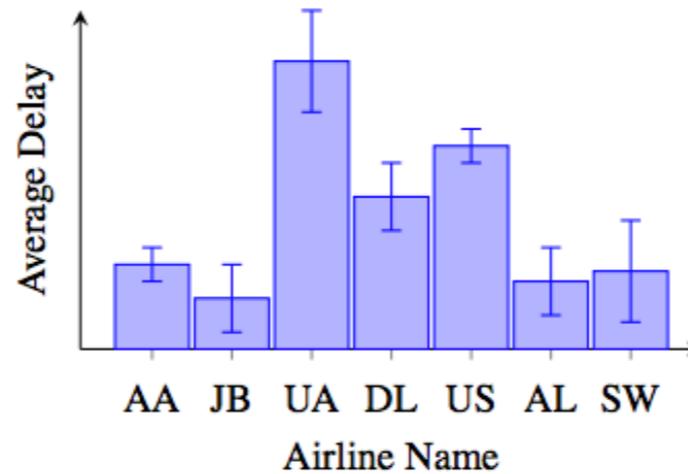
Jane Hoffswell, Arvind Satyanarayan, Jeffrey Heer. *Augmenting Code with In Situ Visualizations to Aid Program Understanding*. CHI '18.



*The What-If Tool: Code-Free Probing of Machine Learning Models*. Google AI

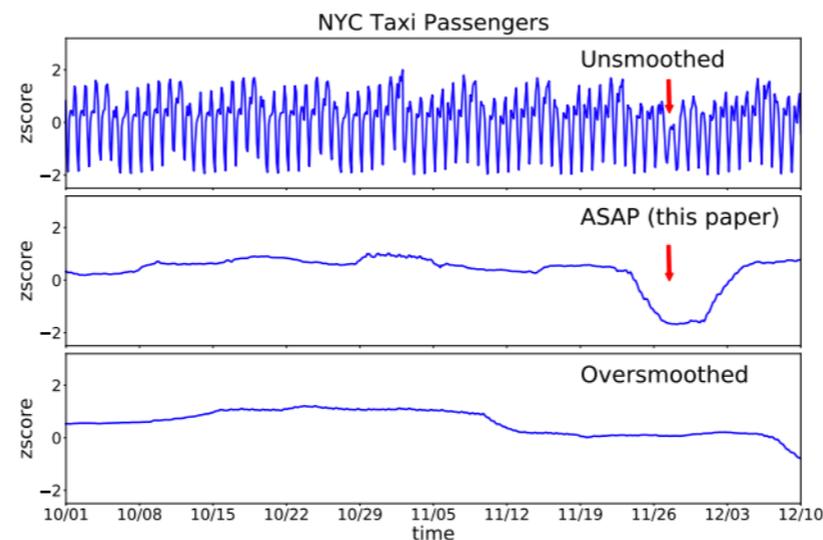
## Data Research

Open up the database:  
The human interactions should drive the (re)design of abstractions and operations. Exciting opportunities in incremental & interactive querying and beyond relational data & queries



Kim et al. *Rapid sampling for visualizations with ordering guarantees*. VLDB '15

Siddiqui et al. *Effortless Visual Data Exploration with Zenvisage: An Interactive and Expressive Visual Analytics System*. VLDB '17



Kexin Rong, Peter Bailis. *ASAP: Prioritizing Attention via Time Series Smoothing*, VLDB '17

Bailis et al. *MacroBase: Prioritizing Attention in Fast Data*, SIGMOD 2017.

### PaQL syntax specification

```
SELECT PACKAGE (*|column_name [...]) [AS] package_name
FROM relation_name [AS] relation_alias
    [REPEAT repeat] [...]
[WHERE w_expression]
[SUCH THAT st_expression]
[(MINIMIZE|MAXIMIZE) obj_expression]
```

### PaQL query for Example 1

```
Q: SELECT PACKAGE (*) AS P
FROM Recipes R REPEAT 0
WHERE R.gluten = 'free'
SUCH THAT COUNT (P.*) = 3 AND
SUM(P.kcal) BETWEEN 2.0 AND 2.5
MINIMIZE SUM(P.sat_fat)
```

Brucato, Matteo, Azza Abouzied, and Alexandra Meliou. *Package queries: efficient and scalable computation of high-order constraints*. The VLDB Journal 2018

# Thank you

---

Can't wait to hear your thoughts, comments or questions.