DataJoint: Bringing databases back into data science

Raphael Guzman
Dimitri Yatsenko, PhD

Intro

- developer
- technologist
- entrepreneur
- oss advocate

Raphael Guzman

https://github.com/guzman-raphael raphael.h.guzman@gmail.com https://www.linkedin.com/in/guzman-raphael/

How many of you are using relational databases?

e.g. MySQL, Postgres, SQLite

How many don't use relational databases because of SQL?

Challenge

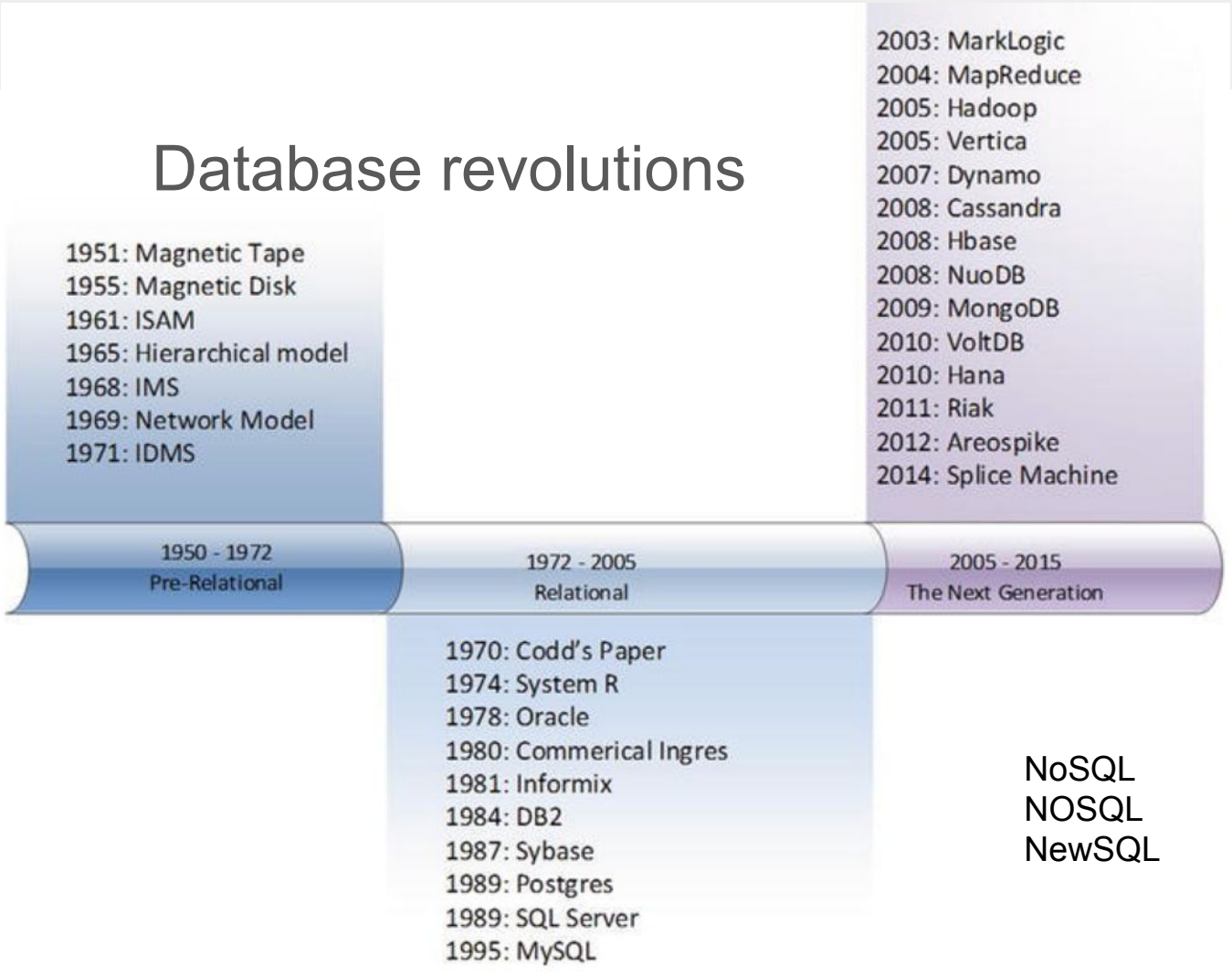
Reproducible compute pipelines in neuroscience and access to their produced results

"neuroscientist doing neuroscience when neuroscience needs to be done"

What do databases do?

- **Data structure** reflects the logic of scientific study
- **Data integrity**: entity integrity, completeness, data types, referential integrity, group integrity, transactions
- **Consistency**: everybody sees the same — concurrent access, transactions
- **Precise queries**: get exact slice of data required for analysis
- **Performance**: indexed lookup
- *Support computational dependencies!*

Database revolutions

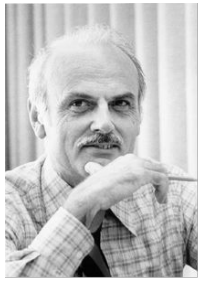


NoSQL
NOSQL
NewSQL

from "Next generation databases: NoSQL and Big Data" by Guy Harrison

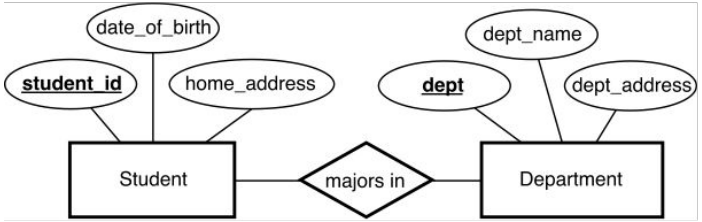
https://arxiv.org/abs/1807.11104

- Data are in flat simple tables (relations)
- Domain constraints
- Unique constraints
- Referential constraints
- Declarative queries

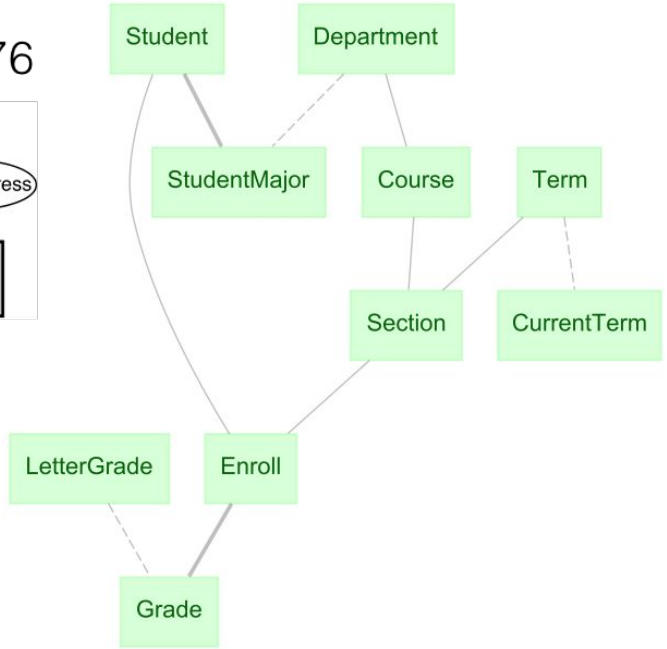


Ted F. Codd
IBM, 1969

Peter Chen, 1976



- Conceptual clarification
- Diagramming notation



- Conceptual refinement
- Definition language
- Query language
- Diagramming notation
- Integrated computation

What is DataJoint?

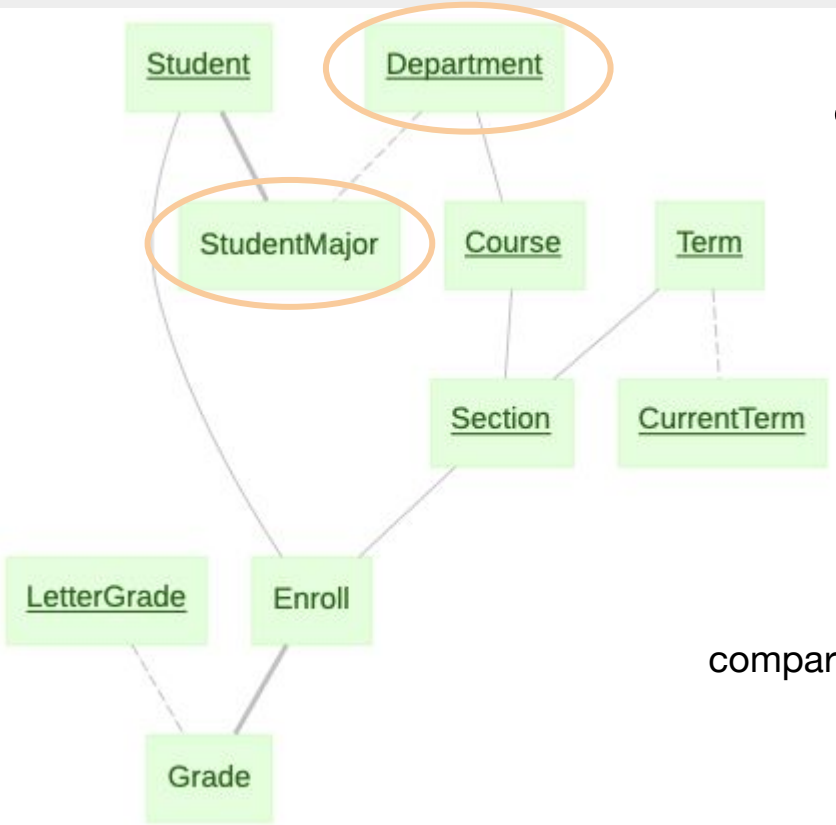
An API to help scientists with:

- Defining computational dependencies
- Querying
- Data lineage

"A researcher unwrangling the mysteries of the brain"

Yatsenko et al., DataJoint: a simpler relational data model, 2018

Data Definition Language
and diagramming notation



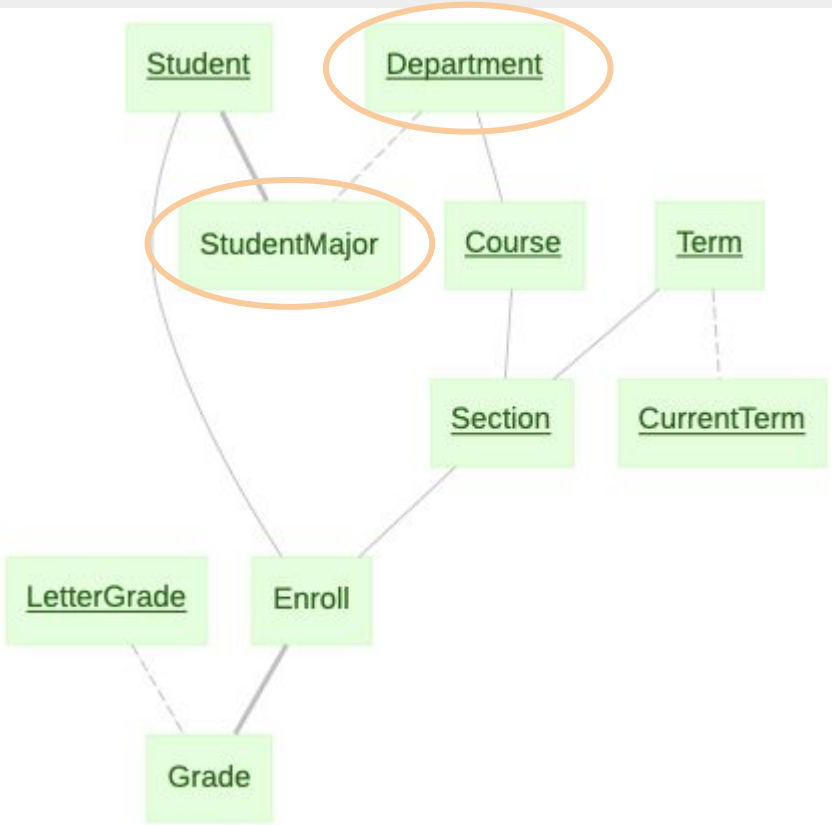
compare to SQL

```
CREATE TABLE Department(
    dept char(6) NOT NULL COMMENT "
        abbreviated department name, e.g.
        BIOL",
    dept_name varchar(200) NOT NULL COMMENT "
        full department name",
    dept_address varchar(200) NOT NULL
        COMMENT "mailing address",
    dept_phone varchar(14),
    PRIMARY KEY(dept))

CREATE TABLE StudentMajor(
    student_id int unsigned NOT NULL COMMENT
        "university ID",
    dept char(6) NOT NULL COMMENT "
        abbreviated department name, e.g.
        BIOL",
    declare_date date NOT NULL COMMENT "when
        student declared her major",
    PRIMARY KEY (student_id),
    FOREIGN KEY (student_id) REFERENCES
        Student(student_id),
    FOREIGN KEY (dept) REFERENCES Department(
        dept))
```

DataJoint: A Simpler Relational Data Model (2018)

Data Definition Language
and diagramming notation

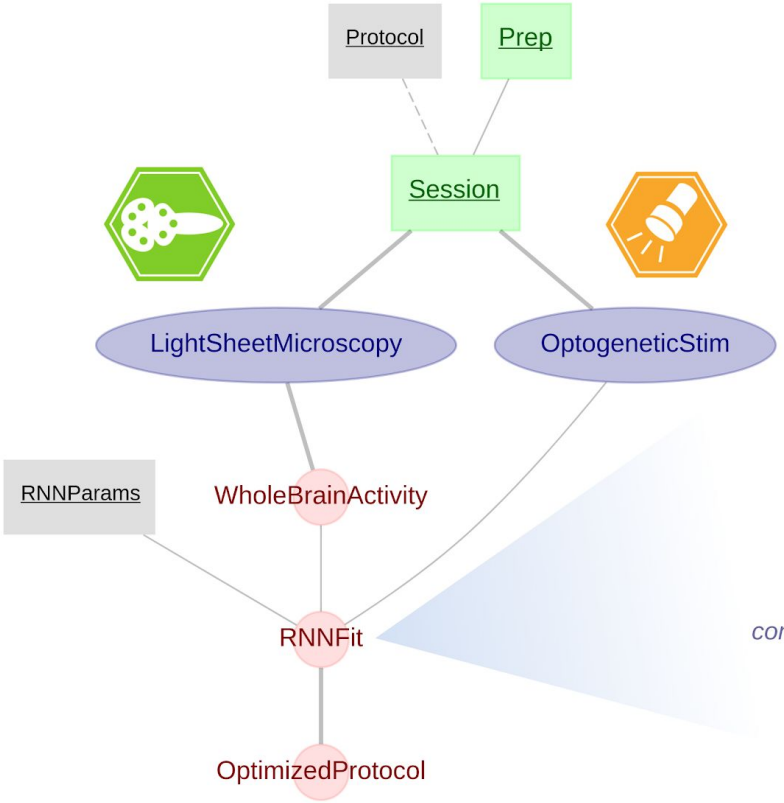



```
::Department
dept : char(6)    # abbreviated department name, e.g. BIOL
---
dept_name    : varchar(200)  # full department name
dept_address : varchar(200)  # mailing address
dept_phone   : varchar(14)

::StudentMajor
-> Student
---
-> Department
declare_date :  date  # when student declared her major
```

DataJoint: A Simpler Relational Data Model (2018)

Computation is part of the data model



database link *table name*

```
@schema
class RNNFit(dj.Computed):
    definition = """
    -> WholeBrainActivity
    -> OptogeneticStim
    -> RNNParams
    ---
    fit   : longblob
    cv    : longblob   # cross-validation
    score : double     # fit quality score
    """

    def make(self, key):

        # fetch data
        params = (RNNParams & key).fetch()
        neural_activity = (WholeBrainActivity & key).fetch()
        opto_stim = (OptogeneticStim & key).fetch()

        # compute
        fit, cv, score = RNN.fit(neural_activity, opto_stim, **params)

        # submit result
        self.insert1(dict(key, fit=fit, cv=cv, score=score))
```

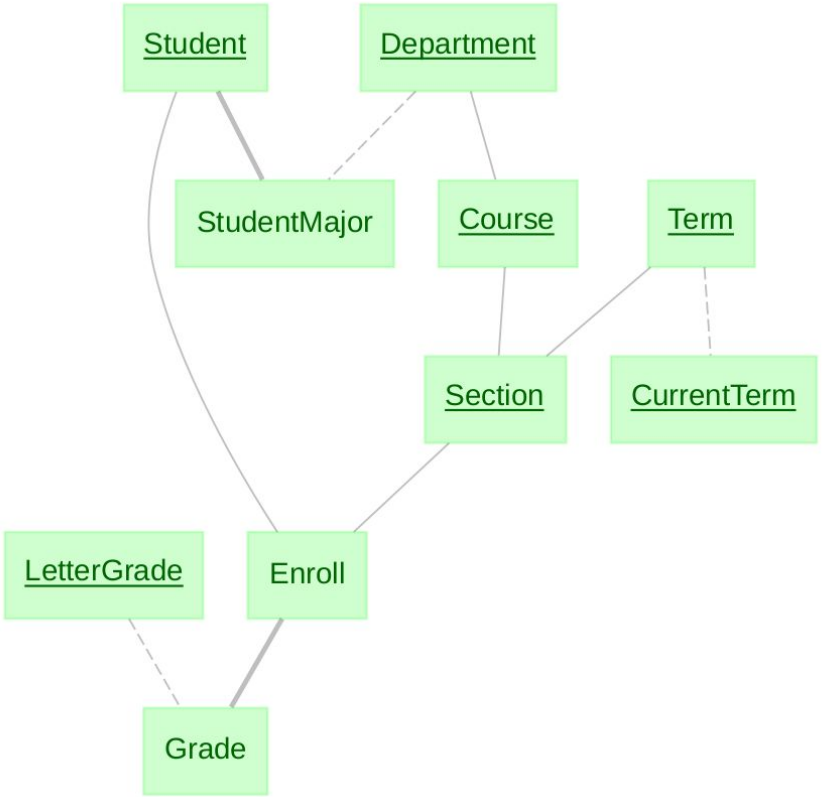
dependency

computed attributes

computation

DataJoint: A Simpler Relational Data Model (2018)

Query Language



Query: "Currently enrolled students"

DataJoint:

```
Student & (Enroll * CurrentTerm)
```

SQL:

```
SELECT * FROM Student
WHERE student_id IN (
  SELECT student_id FROM Enroll
  WHERE (term_year, term) IN (
     SELECT term_year, term
     FROM CurrentTerm))
```

DataJoint: A Simpler Relational Data Model (2018)


Live Demo?

Community of 60+ Research Labs

https://datajoint.com/docs/projects

Key Contributors

Dimitri Yatsenko

Edgar Y. Walker

Fabian Sinz

Jacob Reimer

Andreas Hoenselaar

Alex Ecker

Philipp Berens

Manolis Froudarakis

Raphael Guzman

Kabilar Gunalan

Thinh Nguyen

Chris Turner

Adib Baji

Jeroen Verswijver

Shan Shen

"superheroes that improve science through code"

Any many more…

Thank you!

🙏 Sponsored by NIH Grant U24 NS116470 🙏

Goals

- We seek to elevate DataJoint as an integral part of the SciPy data science stack
- a general tool for scientific relational database programming
- .. with support for computational dependencies
- solid theoretical foundations
- community-driven open-source project
- bolster open-source governance, community feedback, contribution guidelines
- engage NumFocus and other open-source advocates

"fostering a community of respect and acceptance"

Raphael Guzman

Community

- https://datajoint.com/docs
- https://github.com/datajoint/datajoint-tutorials
- https://stackoverflow.com/questions/tagged/datajoint