Holdea

a simple and complete interface for knowledge

Christian B. Williams, PhD

Innovation

Our proposal is a visual interface for relational databases, founded on new discoveries in the fields of category theory and logic, which supports all capabilities such as updates, queries, and migrations through interactions with relations as nodes.

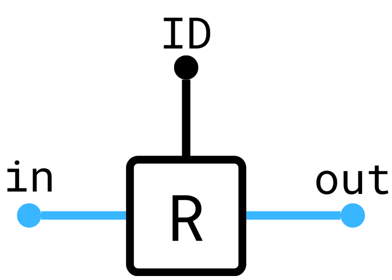
The world now runs on data. Yet exploring and manipulating data still requires a complex web of tools and syntax. For example, an analysis tool like Pandas has to import queries from SQL; and even simple queries like “friend of a friend” require multiple lines of code. Despite the billions spent on data science, we have yet to develop a tool that is both comprehensive and simple to use. This is a significant barrier to public knowledge and agency.

In fact, a unified framework is possible — a relational database is a **logic**: a table is a relation of its column datatypes, and an update is an inference rule between relations. All queries are built from logical operations on relations.

My thesis provides two significant innovations: the completed **visual language** of logic, known as the “string diagrams” of a bifibered category; and the general concept of a “migration” between logics [1], defined as a bifibered profunctor. The language defines a user interface which can support all data actions.

The visualization of logic is based on a simple idea: because a table relates its column datatypes, we can visualize a table as a **node** that connects wires, which can be colored by datatype. This idea originated in the 1800s [2], and was rediscovered in category theory [3].

Here is a small example, for a table of chemical reactions.



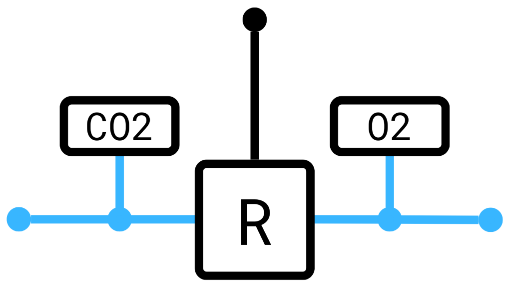
[relation as node]

in	ID	out
CH4 + 2 O2	combustion	CO2 + 2 H2O
C6H12O6 + 6 O2	respiration	6 CO2 + 6 H2O
6 CO2 + 6 H2O	photo-synthesis	C6H12O6 + 6 O2

[table display]

We can press on the node to display it as rows, or a graph, or other ways; yet in the "higher view" of the interface, a whole table is a single "building block", and all queries can be expressed by simple visual operations on nodes and wires.

For example, we can view reactions which transform carbon dioxide into oxygen. The subqueries "mix has CO2"" and "mix has O2" can be expanded and collapsed; the interface has full modularity, to grasp relations of any complexity.



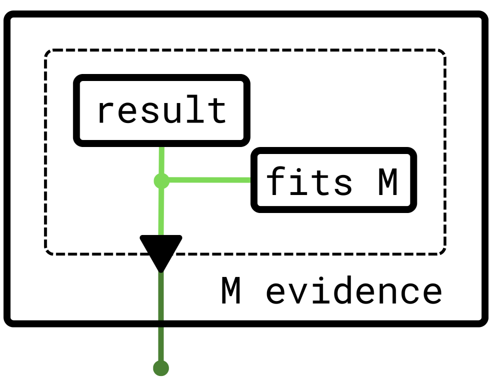
[query]

Contrast with the SQL syntax for this query — which would most people prefer?

```
SELECT * FROM R
JOIN [query "mix has CO2"]
AS CO2 ON R.Inp = CO2.Mix
JOIN [query "mix has  O2"]
AS O2 ON R.Outp =  O2.Mix;
```

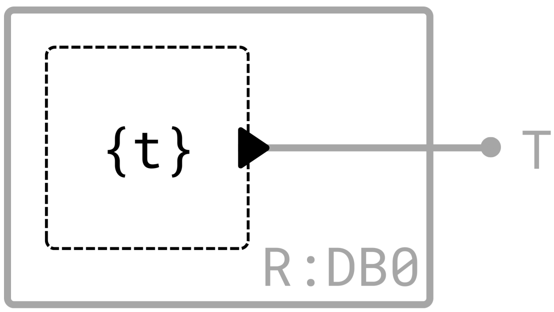
[SQL syntax]

The other basic action is to **update** a table by some rule, visualized by "nesting" an inner relation into an outer relation (the dotted line), along some program (the black triangle). For example: if the *result* of an experiment *fits a model*, then process the result data and insert into the *evidence* for that model.

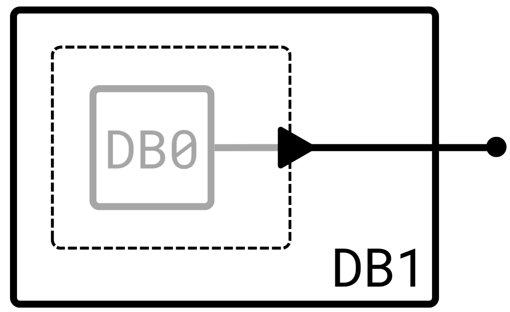


[update rule]

More generally, nesting can represent a **migration** between databases: a whole system of programs between datatypes, and rules between tables; this encompasses the general concept of an "extract, transform, load" workflow. An **instance** of a database is a migration from the trivial schema, and composition of migrations provides traversable histories of database changes.



[instance]



[migration]

In this way, visual logic forms the basis for a complete database tool: schemas, updates, queries, and migrations can all be expressed in one simple interface, with no code separating us from ideas.

References

[1] Williams, C. The Metalanguage of Category Theory. PhD thesis, University of California, Riverside, 2023. Available at https://escholarship.org/uc/item/84j4z67h.

[2] Roberts, D. The Existential Graphs of Charles S. Peirce. De Gruyter, Dec. 1973. Available at https://core.ac.uk/display/82124291.

[3] Joyal, A., and Street, R. The Geometry of Tensor Calculus, I. Advances in Mathematics 88 (1991), 55–112. At https://api.semanticscholar.org/CorpusID:120026295.

Objectives

The object of Phase 1 is the design specification of a visual interface for relational data management, commercially viable as a superior alternative to SQL.

The design should meet the following criteria.

1. Visualize tables and queries as nodes, pressed to display data in various modes;

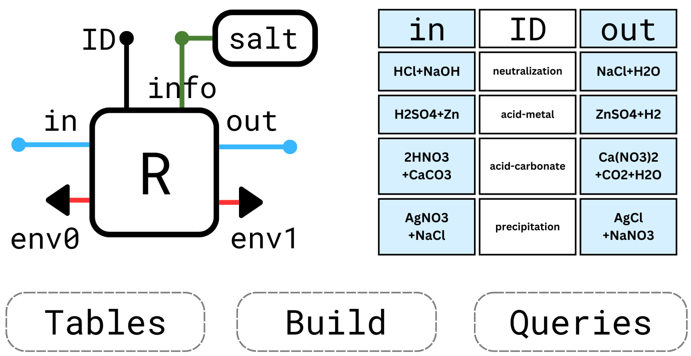
2. Support interactive operations on nodes and wires (both touch and keyboard), for all SQL commands: querying and updating, and developing the schema;

3. Execute these commands as ACID transactions (by default option), and store all updates of the data and the schema in traversable histories;

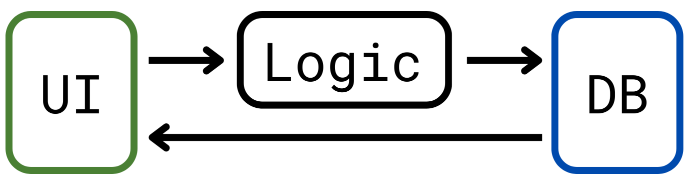
4. Preserve future compatibility with other data tools, and design toward the possibility of a global network of interactive knowledge.

The interface will have three basic “frames”: build, view, and library. The build frame is a canvas for making and editing queries and updates, the view frame provides various modes of displaying relations, and the library frame organizes tables, operations, queries, and rules.

Loading a relational database, the UI could appear as follows: a visual query, here “salt reactions in any environment” in row view mode; plus a “home bar” with the current libraries.



The basic architecture is simple: an interactive UI system, such as React JS, displays operations and sends commands to the Logic backend, such as TypeScript, which then converts and runs SQL commands, returned and displayed to the UI.

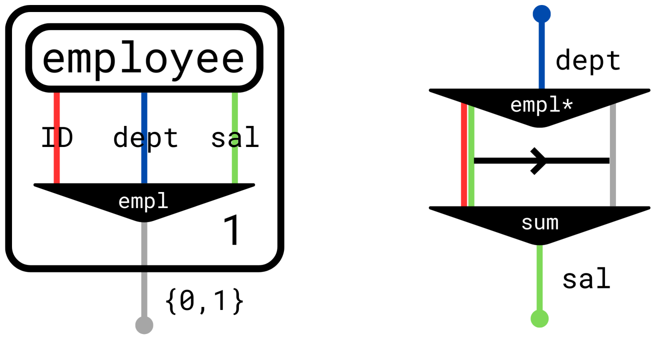


Before development, the foundation needs to be established in a **whitepaper**, of which there are three main parts.

1. Fully specify the visual language of first-order logic: the string diagrams of a self-dual cartesian bifibered category, which models first-order logic as *regular logic* [4] plus *negation* [5]. This defines the interface components of most query operations, and the rules which define update operations.

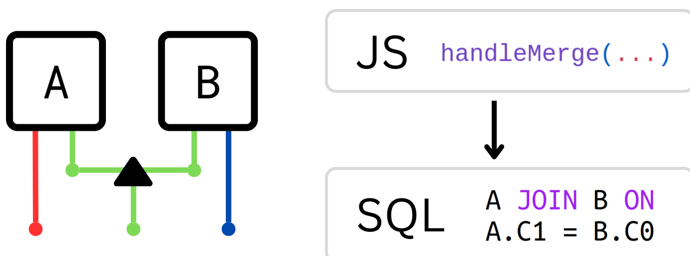
2. Specify the “higher-order” aspect of databases, *aggregate functions*: operations that input a whole table rather than a single row, such as summing many values, or returning the highest value, or counting rows.

Modern research has theorized a visualization: see a table as its truth function, and “bend wires backward” [6] to form a family of tables. For example, below is the query: “for each department, show the total salary of its employees”.



3. Determine the programming languages for the front-end and the back-end; and specify the basic components and actions of the build, view, and library frames.

Then, define the correspondence between user actions and SQL commands. For example: when a user places two tables (nodes) which share a column (wire), and then drags those wires to merge them together — this forms a *join* in SQL.



In general, most core aspects of the UI are already determined by visual logic, and the primary technical risk is suboptimal design. To address this, I will hire a software engineer that is highly experienced in both interfaces and databases.

A Haskell prototype I made to visualize first-order logic is on GitHub [7].

The result of Phase 1 is a full specification of both the visual language and the interface design for ”version alpha”, the SQL alternative, ready for development.

References

[4] Fong, B., and Spivak, D. I. Graphical regular logic, 2019. Available at https://arxiv.org/abs/1812.05765.

[5] Bonchi, F., Giorgio, A. D., Haydon, N., and Sobocinski, P. Diagrammatic algebra of first order logic, 2024. Available at https://arxiv.org/abs/2401.07055.

[6] Baez, J., and Stay, M. Physics, Topology, Logic and Computation: A Rosetta Stone. Available at https://arxiv.org/abs/0903.0340.

[7] Williams, C. Visualization of First-Order Logic. Available at https://github.com/cb-wells/vis_fol.

Market Opportunity

While our long-term vision is for public science, our initial target market is based on the immense need and funding for data management in the private sector.

The data science market size is over 100 billion USD, and expanding rapidly. A company spends millions each year on specialists and tools, just to utilize its databases. What if employees could easily navigate and update and analyze company data, with minimal training? Save money, improve communication and workflow — and have greater insight, by making *everyone* a data scientist.

Our customer is a company which values relational data, but is weighed down by too much code and too many tools. Holdea can provide a complete data interface, by visualizing all your information as connections. The near-term value proposition is a significantly better alternative to SQL; and the long-term value proposition is an all-in-one database framework.

The expected pain point is *interoperability*, so customers will be determined by tool integration. We plan to focus first on MySQL, used by more than 100,000 companies worldwide, to establish the user base. Offer the query tool as a free plug-in, and then develop capabilities as features that are free for individuals and paid for companies. Once integrated with the ETL platform Airflow, and eventually the whole suite of Apache data tools, Holdea will support most database needs.

Company

The company is now incorporated by the name Holdea, with domain [holdea.co]. The author is presently the sole founder, and principal investigator.

Last year, I completed a PhD in Mathematics at the University of California [1], with guidance of leading experts John Baez and Mike Shulman, in category theory — the general language founding the idea that “knowledge is connection”.

The community of Applied Category Theory intersects mathematics, science, and programming; and many of its members have pertinent skills and connections. There are currently several who are able and willing to support this development.

For Phase 1, I will hire a software engineer, and a business person who is experienced in the database sector; for software I am in contact with a strong candidate, and for business I am connected to someone who knows strong candidates. Because the initial design is simple, we can make a superior alternative to SQL within two years; and then we will have significant momentum.

The vision is simple: empower people to see, hold, create, and share knowledge. I have already seen how this message resonates with many different kinds of people, and I am confident that I can assemble the team to bring this vision into reality.