VAQO
Visual Algebraic Query Optimizer

Team

Christopher Broderick Computer Science Algorithm Developer

Don Nguyen Computer Science Database Developer

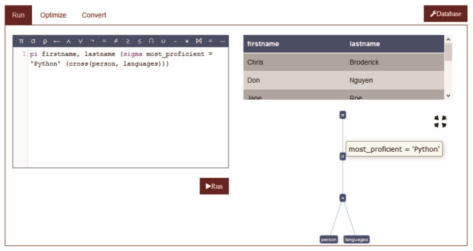
Sumeet Ranu Computer Science Web Developer

Karen Davis Professor - EECS Advisor

Tool

Live at: **vaqo.azurewebsites.net**

Run relational algebra queries on the DB

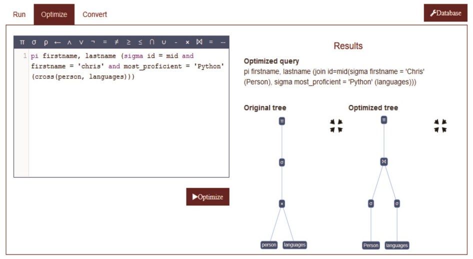


```
pi firstname, lastname (sigma most_proficient =
'Python' (cross(person, languages)))
```

firstname	lastname
Chris	Broderick
Don	Nguyen
Jane	Roe

most_proficient = 'Python'

Optimize using rules and heuristics



```
pi firstname, lastname (sigma id = mid and
firstname = 'chris' and most_proficient = 'Python'
(cross(person, languages)))
```

Results

Optimized query

pi firstname, lastname (join id=mid(sigma firstname = 'Chris' (Person), sigma most_proficient = 'Python' (languages)))

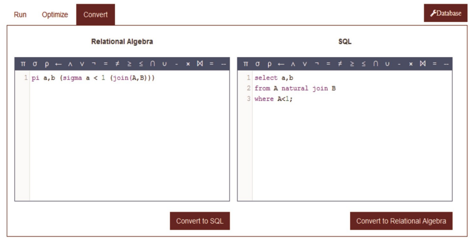
Original tree

Optimized tree

Optimization steps for the above example:

1. Sigma (σ) split into 3 parts
2. Sigmas (σ) pushed as far down as possible
3. Cross (×) combined with sigma (σ) to make a join (⋈)

Convert between SQL & relational algebra


Run
Optimize
Convert
Database
π σ ρ ← ∧ ∨ ¬ = ≠ ≥ ≤ ∩ ∪ - × ⋈ = --
Convert to SQL
Convert to Relational Algebra

Relational Algebra

```
pi a,b (sigma a < 1 (join(A,B)))
```

SQL

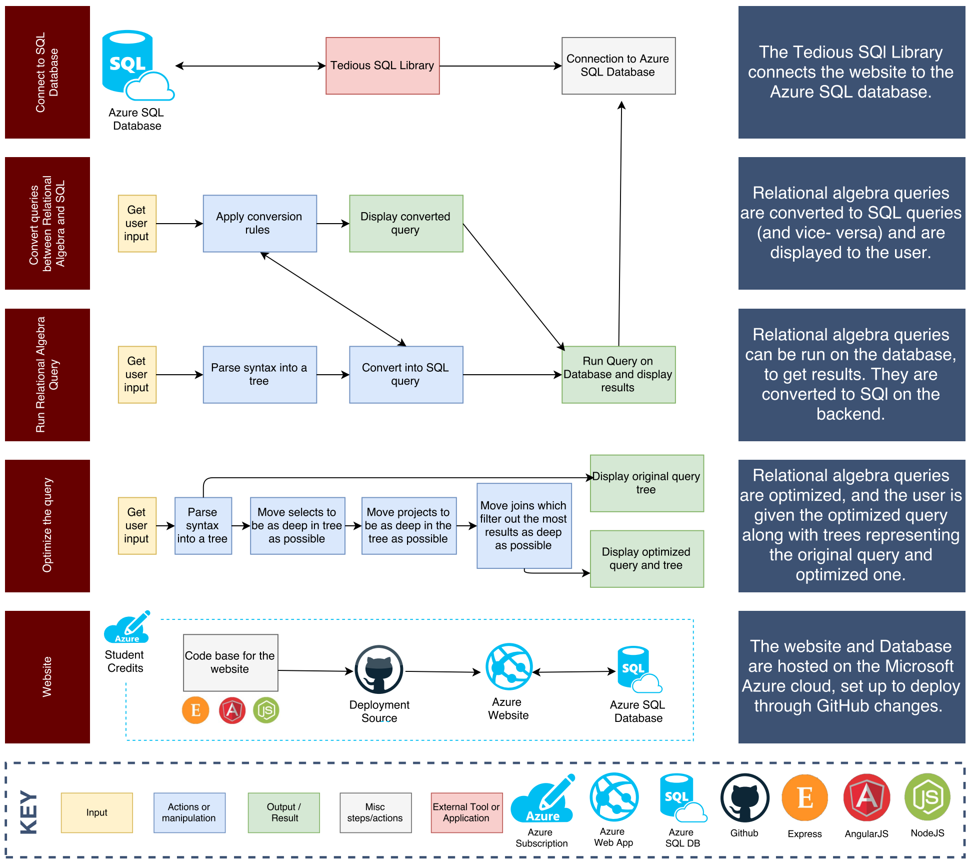
```
select a,b
from A natural join B
where A<1;
```

Problem

- There are many separate tools that can run, convert and optimize relational algebra queries, but there isn't a single central tool.
- Optimizing queries is a complex task that is difficult to visualize. In addition to that, relational algebra syntax is counter intuitive and hard to break down.
- Relational algebra is an abstract concept. Our main stakeholder Dr. Davis observed in her classroom that it can be difficult for students to understand without application.

Solution

- VAQO is an online tool to run queries, convert between relational algebra and SQL and optimize relational algebra queries, on 1 page.
- The tool creates interactive query trees that break down the relational algebra syntax into steps. The trees are much easier to read than queries, and give a better understanding of changes due to the optimization.
- Queries are run in real-time on a database, and the results are displayed to the user. This helps users understand what the query does by applying the concepts to real data.



The Tedious SQL Library connects the website to the Azure SQL database.

Relational algebra queries are converted to SQL queries (and vice- versa) and are displayed to the user.

Relational algebra queries can be run on the database, to get results. They are converted to SQL on the backend.

Relational algebra queries are optimized, and the user is given the optimized query along with trees representing the original query and optimized one.

The website and Database are hosted on the Microsoft Azure cloud, set up to deploy through GitHub changes.

Obstacles

Database connectivity roadblocks - The SQL server connection had roadblocks that made us consider different routes to take.

- Many SQL connection frameworks work locally but were difficult to implement on the cloud due to latency, firewalls, etc. The solution was the "Tedious" library that connects NodeJS to Azure SQL databases.
- The plan was to let each user upload a SQL dump to have a separate database to work with. Our Azure credits only supported a single database, so we did not go that route.

Converting from relational algebra to SQL

- The order of operations in relational algebra is different from the order in SQL, making it difficult to write the optimization algorithms.
- Due to this, the conversion algorithm took longer to write than the others.

Future

User database connections - Allow users to connect to a personal database server or import a SQL dump, and be able to run queries and optimizations on their own database.

Add learning tools - Add guides and other tools to further enhance the learning experience.

- A tutorial on how to use the website.
- Training modules that go through query optimization heuristics.
- Enhanced query trees that display the size of data and complexity reduced by each step.

Technologies

Web: node Express

DB: SQL, tedious, Microsoft SQL Server

UI: Bootstrap, Font Awesome