Formal Semantics of SQL
(and Cypher)

Paolo Guagliardo
**SQL**

- **Standard** query language for relational databases
- **$30B/year** business
- Implemented in all major RDBMSs (free and commercial)
- First standardized in **1986** (ANSI) and **1987** (ISO)

“The nice thing about standards is that you have so many to choose from” — Andrew S. Tanenbaum
How standard is SQL?

```
SELECT * FROM ( SELECT R.A, R.A FROM R ) S
```

PostgreSQL outputs a table with two columns named “A”

Oracle throws an ERROR: reference to column “A” is ambiguous

```
SELECT * FROM R WHERE EXISTS ( SELECT * FROM ( SELECT R.A, R.A FROM R ) S )
```

Both PostgreSQL and Oracle output R
Who is right?

Let’s have a look at the standard!

A. If the `<select list> *` is simply contained in a `<subquery>` that is immediately contained in an `<exists predicate>`, then the `<select list>` is equivalent to a `<value expression>` that is an arbitrary `<literal>`.

B. Otherwise, the `<select list> *` is equivalent to a `<value expression>` sequence in which each `<value expression>` is a column reference that references a column of T and each column of T is referenced exactly once. The columns are referenced in the ascending sequence of their ordinal position within T.
... which means

\[
\text{SELECT } * \\
\text{FROM ( SELECT R.A, R.A } \\
\text{ FROM R ) S}
\]

\[
\equiv
\]

\[
\text{SELECT S.A, S.A } \\
\text{FROM ( SELECT R.A, R.A } \\
\text{ FROM R ) S}
\]

\[
\text{SELECT } * \text{ FROM R } \\
\text{WHERE EXISTS ( \\
\text{ SELECT } * \\
\text{ FROM ( SELECT R.A, R.A } \\
\text{ FROM R ) S } \\
\text{ )}
\]

\[
\equiv
\]

\[
\text{SELECT R.A FROM R } \\
\text{WHERE EXISTS ( \\
\text{ SELECT 1 } \\
\text{ FROM ( SELECT R.A, R.A } \\
\text{ FROM R ) S } \\
\text{ )}
\]
The Need for a Formal Semantics

• Avoid ambiguity of natural language
• Clearly defined and not subject to interpretation
• Easy to understand and implement

Previous attempts

• Many simplifying assumptions: no bags, no nulls
• No justification of correctness
SELECT R.A FROM R
EXCEPT
SELECT S.A FROM S

SELECT R.A FROM R
WHERE R.A NOT IN (SELECT S.A FROM S)

SELECT R.A FROM R
WHERE NOT EXISTS (SELECT S.A FROM S WHERE S.A=R.A)
Core SQL fragment

\[ \begin{align*}
\tau &:= (T_1, \ldots, T_k), \quad \beta := (N_1, \ldots, N_k), \quad k > 0 \\
\alpha &:= (A_1, \ldots, A_m), \quad \beta' := (N'_1, \ldots, N'_m), \quad m > 0
\end{align*} \]

**Queries:**

\[ Q := \text{SELECT [DISTINCT]} (\alpha : \beta' | *) \text{ FROM } \tau : \beta \text{ WHERE } \theta \]

\[ Q (\text{UNION} | \text{INTERSECT} | \text{EXCEPT}) \text{ [ALL] } Q \]

**Conditions:**

\[ \theta := \text{TRUE} | \bar{t} (= | \neq) \bar{t} | t \text{ IS [NOT] NULL} \]

\[ | \bar{t} \text{ [NOT] IN } Q | \text{EXISTS } Q \]

\[ | \theta \text{ AND } \theta | \theta \text{ OR } \theta | \text{NOT } \theta \]

Essentially SQL without arithmetic, grouping and aggregation.
Data model

- Base relations / query outputs / intermediate results
- Primitive data manipulation operations

Attribute references

- Binding rules in subqueries
- *Environment* collects and propagates bindings
Proposed Semantics

- Fits in one page
- Non-ambiguous
- Easy to understand
- Easy to implement
- Easy to modify
Formal Semantics: Validation

• Cannot **prove** that semantics is correct
• Provide sufficient *experimental evidence*
• Implemented in Python
• Validated on 100000+ random SQL queries
Formal Semantics of Cypher

• Collaboration between Neo Technology and the University of Edinburgh

• Preliminary meeting in December

• Legal agreements finalized recently

• Neo Technology sponsors a researcher (Nadime Francis)
Challenges

• Getting the (abstract) data model right
• Intermediate representation (QUIL?)
• Identify core fragment
• Language constantly evolving
• Follow the footsteps of SQL? (nulls)