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)
- Several revision afterwards
  (SQL-89, SQL-92, SQL:1999, SQL:2003, SQL:2006, SQL:2008, SQL:2011, SQL:2016)

> *"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

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

≡

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

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

≡

```
SELECT R.A FROM R
WHERE EXISTS (
   SELECT 1
   FROM ( SELECT R.A, R.A
          FROM   R ) S
)
```

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

R

A
1
NULL

S

A
NULL

```
SELECT R.A FROM R
EXCEPT
SELECT S.A FROM S
```

Answer

A
1

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

A

```
SELECT R.A FROM R
WHERE  NOT EXISTS (
  SELECT S.A FROM S
  WHERE  S.A=R.A )
```

A
1
NULL

Core SQL fragment

$$\tau := (T_1, \ldots, T_k), \quad \beta := (N_1, \ldots, N_k), \quad k > 0$$
$$\alpha := (A_1, \ldots, A_m), \ \beta' := (N'_1, \ldots, N'_m), \ m > 0$$

Queries:

$$Q := \texttt{SELECT}\ [\texttt{DISTINCT}]\ (\alpha{:}\beta' \mid *)\ \texttt{FROM}\ \tau{:}\beta\ \texttt{WHERE}\ \theta$$
$$\mid Q\ (\texttt{UNION} \mid \texttt{INTERSECT} \mid \texttt{EXCEPT})\ [\texttt{ALL}]\ Q$$

Conditions:

$$\theta := \texttt{TRUE} \mid \bar{t}\,(= \mid \neq)\,\bar{t} \mid t\ \texttt{IS}\ [\texttt{NOT}]\ \texttt{NULL}$$
$$\mid \bar{t}\ [\texttt{NOT}]\ \texttt{IN}\ Q \mid \texttt{EXISTS}\ Q$$
$$\mid \theta\ \texttt{AND}\ \theta \mid \theta\ \texttt{OR}\ \theta \mid \texttt{NOT}\ \theta$$

Essentially SQL without arithmetic, grouping and aggregation

Formal Semantics: Challenges

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

$$[\![R]\!]_{D,\eta} = R^D$$
$$[\![\tau : \beta]\!]_{D,\eta} = [\![(T_1, \ldots, T_k) : (N_1, \ldots, N_k)]\!]_{D,\eta} = N_1 \bullet [\![T_1]\!]_{D,\eta} \times \cdots \times N_k \bullet [\![T_k]\!]_{D,\eta}$$
$$\left[\!\!\left[\begin{array}{rl} \texttt{FROM} & \tau : \beta \\ \texttt{WHERE} & \theta \end{array} \right]\!\!\right]_{D,\eta} = \left\{ \bar{a} \in [\![\tau : \beta]\!]_{D,\eta} \mid [\![\theta]\!]_{D,\eta;\eta^{\bar{a}}} = \mathbf{t} \right\}$$
$$\left[\!\!\left[\begin{array}{rl} \texttt{SELECT} & * \\ \texttt{FROM} & \tau : \beta \\ \texttt{WHERE} & \theta \end{array} \right]\!\!\right]_{D,\eta} = \left[\!\!\left[\begin{array}{rl} \texttt{FROM} & \tau : \beta \\ \texttt{WHERE} & \theta \end{array} \right]\!\!\right]_{D,\eta} : \beta^{-1}$$
$$\left[\!\!\left[\begin{array}{rl} \texttt{SELECT} & \alpha : \beta' \\ \texttt{FROM} & \tau : \beta \\ \texttt{WHERE} & \theta \end{array} \right]\!\!\right]_{D,\eta} = \pi_\alpha \left(\left[\!\!\left[\begin{array}{rl} \texttt{FROM} & \tau : \beta \\ \texttt{WHERE} & \theta \end{array} \right]\!\!\right]_{D,\eta} \right) : \beta'$$
$$\left[\!\!\left[\begin{array}{rl} \texttt{SELECT DISTINCT} & \alpha : \beta' \mid * \\ \texttt{FROM} & \tau : \beta \\ \texttt{WHERE} & \theta \end{array} \right]\!\!\right]_{D,\eta} = \varepsilon \left(\left[\!\!\left[\begin{array}{rl} \texttt{SELECT} & \alpha : \beta' \mid * \\ \texttt{FROM} & \tau : \beta \\ \texttt{WHERE} & \theta \end{array} \right]\!\!\right]_{D,\eta} \right)$$

$$[\![\texttt{TRUE}]\!]_{D,\eta} = \mathbf{t}$$
$$[\![t]\!]_{D,\eta} = \begin{cases} \eta(A) & \text{if } t = A \\ t & \text{if } t \in \mathsf{C} \text{ or } t = \texttt{NULL} \end{cases}$$
$$[\![t_1 = t_2]\!]_{D,\eta} = \begin{cases} \mathbf{t} & \text{if } [\![t_1]\!]_{D,\eta} = [\![t_2]\!]_{D,\eta} \text{ and } [\![t_1]\!]_{D,\eta} \neq \texttt{NULL} \text{ and } [\![t_2]\!]_{D,\eta} \neq \texttt{NULL} \\ \mathbf{f} & \text{if } [\![t_1]\!]_{D,\eta} \neq [\![t_2]\!]_{D,\eta} \text{ and } [\![t_1]\!]_{D,\eta} \neq \texttt{NULL} \text{ and } [\![t_2]\!]_{D,\eta} \neq \texttt{NULL} \\ \mathbf{u} & \text{if } [\![t_1]\!]_{D,\eta} = \texttt{NULL} \text{ or } [\![t_2]\!]_{D,\eta} = \texttt{NULL} \end{cases}$$
$$[\![t \texttt{ IS NULL}]\!]_{D,\eta} = \begin{cases} \mathbf{t} & \text{if } [\![t]\!]_{D,\eta} = \texttt{NULL} \\ \mathbf{f} & \text{if } [\![t]\!]_{D,\eta} \neq \texttt{NULL} \end{cases}$$
$$[\![t \texttt{ IS NOT NULL}]\!]_{D,\eta} = \neg [\![t \texttt{ IS NULL}]\!]_{D,\eta}$$
$$[\![(t_1, \ldots t_n) = (t'_1, \ldots, t'_n)]\!]_{D,\eta} = \bigwedge_{i=1}^{n} [\![t_i = t'_i]\!]_{D,\eta} \qquad [\![(t_1, \ldots t_n) \neq (t'_1, \ldots, t'_n)]\!]_{D,\eta} = \bigvee_{i=1}^{n} [\![t_i \neq t'_i]\!]_{D,\eta}$$
$$[\![\bar{t} \texttt{ IN } Q]\!]_{D,\eta} = \begin{cases} \mathbf{t} & \text{if } \exists \bar{r} \in [\![Q]\!]_{D,\eta} \colon [\![\bar{t} = \nu(\bar{r})]\!]_{D,\eta} = \mathbf{t} \\ \mathbf{f} & \text{if } \forall \bar{r} \in [\![Q]\!]_{D,\eta} \colon [\![\bar{t} = \nu(\bar{r})]\!]_{D,\eta} = \mathbf{f} \\ \mathbf{u} & \text{if } \nexists \bar{r} \in [\![Q]\!]_{D,\eta} \colon [\![\bar{t} = \nu(\bar{r})]\!]_{D,\eta} = \mathbf{t} \text{ and } \exists \bar{r} \in [\![Q]\!]_{D,\eta} \colon [\![\bar{t} = \nu(\bar{r})]\!]_{D,\eta} \neq \mathbf{f} \end{cases}$$
$$[\![\bar{t} \texttt{ NOT IN } Q]\!]_{D,\eta} = \neg [\![\bar{t} \texttt{ IN } Q]\!]_{D,\eta}$$
$$[\![\texttt{EXISTS } Q]\!]_{D,\eta} = \begin{cases} \mathbf{t} & \text{if } [\![Q]\!]_{D,\eta} \neq \varnothing \\ \mathbf{f} & \text{if } [\![Q]\!]_{D,\eta} = \varnothing \end{cases}$$

$$[\![\theta_1 \texttt{ AND } \theta_2]\!]_{D,\eta} = [\![\theta_1]\!]_{D,\eta} \wedge [\![\theta_2]\!]_{D,\eta} \qquad [\![\theta_1 \texttt{ OR } \theta_2]\!]_{D,\eta} = [\![\theta_1]\!]_{D,\eta} \vee [\![\theta_2]\!]_{D,\eta} \qquad [\![\texttt{NOT } \theta]\!]_{D,\eta} = \neg [\![\theta]\!]_{D,\eta}$$

$$[\![Q_1 \texttt{ UNION ALL } Q_2]\!]_{D,\eta} = [\![Q_1]\!]_{D,\eta} \cup [\![Q_2]\!]_{D,\eta} : \ell([\![Q_1]\!])$$
$$[\![Q_1 \texttt{ INTERSECT ALL } Q_2]\!]_{D,\eta} = [\![Q_1]\!]_{D,\eta} \cap [\![Q_2]\!]_{D,\eta} : \ell([\![Q_1]\!])$$
$$[\![Q_1 \texttt{ EXCEPT ALL } Q_2]\!]_{D,\eta} = [\![Q_1]\!]_{D,\eta} - [\![Q_2]\!]_{D,\eta} : \ell([\![Q_1]\!])$$
$$[\![Q_1 \star Q_2]\!]_{D,\eta} = \varepsilon\big([\![Q_1 \star \texttt{ ALL } Q_2]\!]_{D,\eta}\big), \qquad \star \in \{\texttt{UNION}, \texttt{INTERSECT}\}$$
$$[\![Q_1 \texttt{ EXCEPT } Q_2]\!]_{D,\eta} = \varepsilon([\![Q_1]\!]_{D,\eta}) - [\![Q_2]\!]_{D,\eta} : \ell([\![Q_1]\!])$$

- 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)