Structured Data Integration

Reference Mechanisms

Enamul Haque (Moni)

Grant Weddell, David Toman

We are extending and hence replacing RM/SQL with GRM/SQLP as the integrating data model and query language for structured data integration. GRM/SQLP is more succinct and separates the task of resolving reference issues from the task of designing integrating schemata. GRM/SQLP semantics is also more flexible in accommodating graph-based data sources.

⟨*table-defn*⟩ ::= `create table` ⟨*table-name*⟩ "(" {⟨*table-item*⟩} ")" [⟨*materialization*⟩]

⟨*primary-key*⟩ ::= `primary key` "(" {⟨*attribute-name*⟩} ")"

⟨*foreign-key*⟩ ::= [`constraint` ⟨*constraint-name*⟩] `foreign key` "(" {⟨*attribute-name*⟩} ")" `references` ⟨*table-name*⟩ ["(" {⟨*attribute-name*⟩} ")"]

⟨*preference*⟩ ::= `preference` "(" {⟨*table-name*⟩} ")"

⟨*inclusion-dependency*⟩ ::= (`inclusion dependency` | `inc`) "(" {⟨*attribute-name*⟩} ")" (`references` | `ref`) ⟨*table-name*⟩ ["(" {⟨*attribute-name*⟩} ")"]

⟨*isa*⟩ ::= `isa` "(" {⟨*table-name*⟩} ")"

⟨*disjoint-from*⟩ ::= (`disjoint from` | `disj`) "(" {⟨*table-name*⟩} ")"

⟨*cover-by*⟩ ::= (`cover by` | `cov`) "(" {[`not`] ⟨*table-name*⟩} ")"

⟨*path-functional-dependency*⟩ ::= (`path functional dependency` | `pfd`) [`with` ⟨*table-name*⟩] "(" {⟨*attribute-path*⟩} ")" (`determines` | `->`) ⟨*attribute-path*⟩

⟨*nominal*⟩ ::= `nominal`

⟨*constraint-name*⟩ ::= (syntax for identifiers in C)



⟨*query*⟩ ::= `select` [`distinct`] {⟨*term*⟩ [`as` ⟨*attribute-name*⟩]} [⟨*body*⟩]
| ⟨*query*⟩ `union` [`all`] ⟨*query*⟩
| `with` ⟨*table-name*⟩ `as` "(" ⟨*query*⟩ ")" ⟨*query*⟩
| "(" ⟨*query*⟩ ")"

⟨*body*⟩ ::= `from` {⟨*correlation*⟩} [`where` ⟨*predicate*⟩]

⟨*correlation*⟩ ::= ⟨*table-name*⟩ [⟨*covariable*⟩]
| "(" ⟨*query*⟩ ")" ⟨*covariable*⟩

⟨*predicate*⟩ ::= `true`
| ⟨*term*⟩ ⟨*comparison*⟩ ⟨*term*⟩
| ⟨*predicate*⟩ `and` ⟨*predicate*⟩
| ⟨*predicate*⟩ `or` ⟨*predicate*⟩
| `not` ⟨*predicate*⟩
| `exists (` `select *` ⟨*body*⟩ `)`
| "(" ⟨*predicate*⟩ ")"

⟨*term*⟩ ::= ⟨*constant*⟩
| ":"⟨*host-variable*⟩
| [⟨*covariable*⟩].⟨*attribute-path*⟩
| ⟨*unary-operator*⟩ ⟨*term*⟩
| ⟨*term*⟩ ⟨*binary-operator*⟩ ⟨*term*⟩
| "(" ⟨*term*⟩ ")"

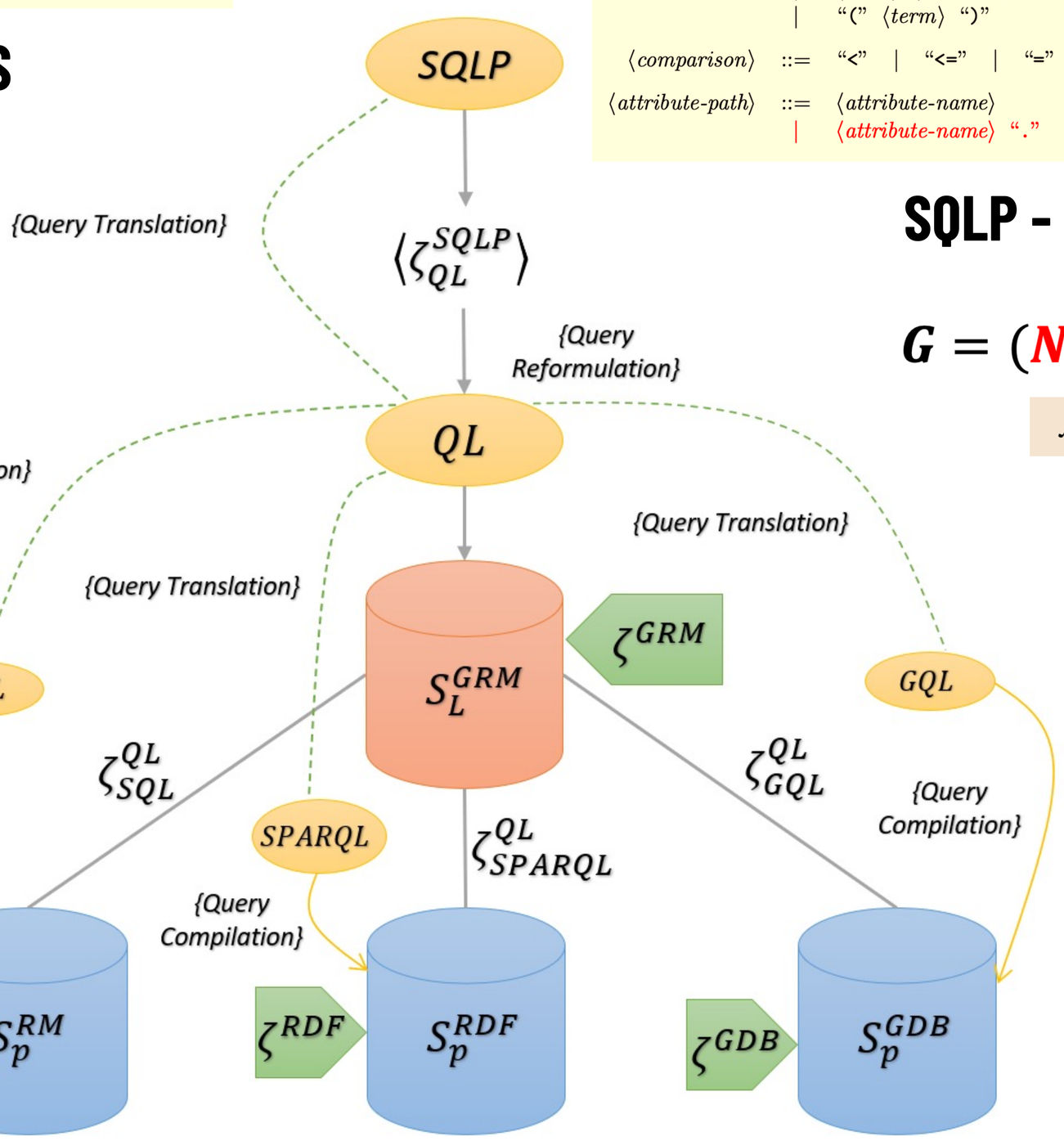
⟨*comparison*⟩ ::= "<" | "<=" | "=" | ">" | ">=" | "<>"

⟨*attribute-path*⟩ ::= ⟨*attribute-name*⟩
| ⟨*attribute-name*⟩ "." ⟨*attribute-path*⟩

GRM – TABLE & CLAUSES

Identity Resolution

Required to resolve identification issues in Query Translations



RM GRM

RDF GRM

P.Graph GRM

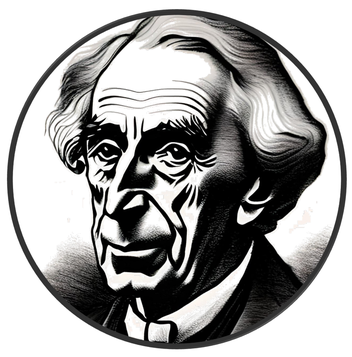
GRM FunDL

SQLP - QUERY LANGUAGE

$$G = (N_R \uplus N_G \uplus N_B, E)$$

$$\mathcal{M}([\![Q]\!](G)) = [\![Q]\!](\mathcal{M}(G))$$





Existential & Uniqueness Constraints

Existential constraints ensure such an entity exists and uniqueness constraints ensure there is one and only one such entity that exists.

$$Rt \quad ::= \quad \{?\} \mid T \mid \mathsf{Pf} = Rt \mid Rt_1, Rt_2 \mid Rt_1; Rt_2$$

RGB Graph as Semantics

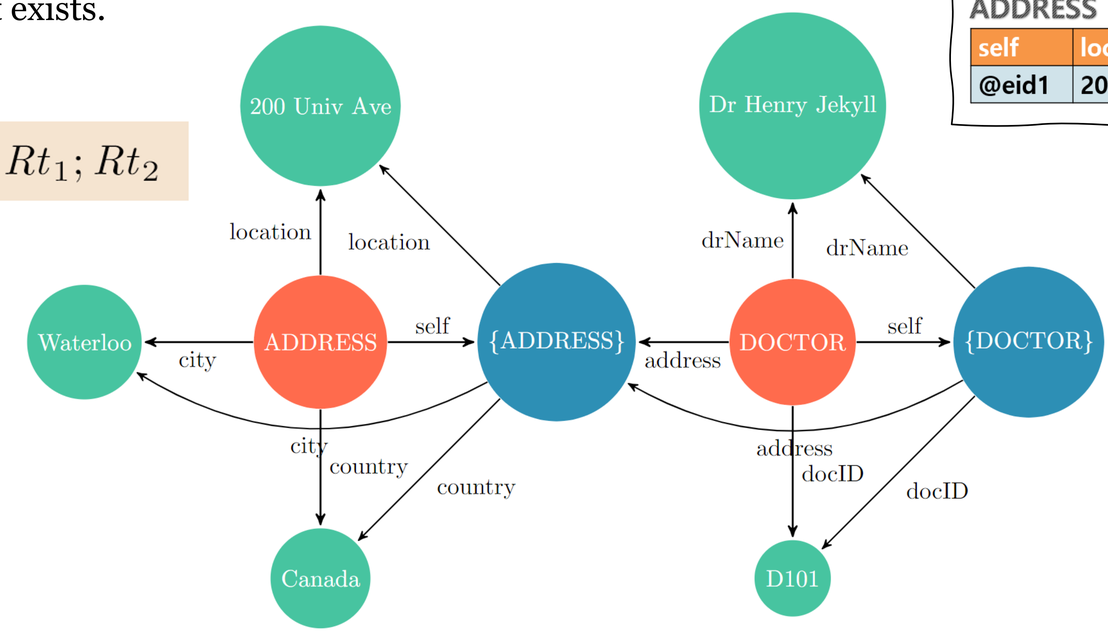
Example ...

DOCTOR

self	docID	drName	address
@eid2	D101	Dr Henry Jekyll	@eid1

ADDRESS

self	location	city	country
@eid1	200 Univ Ave	Waterloo	Canada



UNIVERSITY OF WATERLOO
FACULTY OF MATHEMATICS
David R. Cheriton School of Computer Science

WATERLOO DSG Data Systems Group