RQL: An SQL-like Query Language for Discovering Meaningful Rules

http://rql.insa-lyon.fr

Brice Chardin[1], Emmanuel Coquery[2], Marie Pailloux[3] and Jean-Marc Petit[4]

[1] LIAS, ISAE-ENSMA [2] LIRIS, Université Lyon 1
[3] LIMOS, Université Blaise Pascal [4] LIRIS, INSA Lyon

Motivation

- **Database management systems** and **data mining systems**: two separate worlds
  - data preprocessing between different systems and formats
- **Implications**: logical statements, very natural and common in everyone's terminology
- RQL inference rules require 3 properties: **reflexivity**, **augmentation** and **transitivity**, well known in databases with functional dependencies and formal concept analysis with implications

→ RQL: a declarative approach for pattern mining using database formalisms

Language definition

SafeRL: a logical query language for rules in relational databases

- compliant with the 3 aforementioned properties
- a SafeRL query Q defines a closure operator

$$Q = \{ X \rightarrow Y : \forall t_1 \ldots \forall t_n \left[\psi(t_1, \ldots, t_n) \rightarrow \left(\forall A \in X(\delta(A, t_1, \ldots, t_n)) \rightarrow \forall A \in Y(\delta(A, t_1, \ldots, t_n)) \right) \right] \}$$

  with ψ a TRC formula and δ a mining formula

RQL: its declarative counterpart

```
FINDRULES
OVER A_1, ..., A_n
SCOPE t_1(SQL_1), ..., t_n(SQL_n)
WHERE ψ(t_1, ..., t_n)
CONDITION ON A IS δ(A, t_1, ..., t_n)
```

Example: functional dependencies (1)

$$\{X \to Y : \forall t1, t2 \in R\ (\forall A \in X(t1[A] = t2[A])) \Rightarrow (\forall A \in Y(t1[A] = t2[A])\}$$

Sample relation R

R	**A**	**B**	**C**
τ_1	1	2	1
τ_2	1	2	4
τ_3	2	3	4

Functional dependencies on R

```
FINDRULES OVER A, B, C
SCOPE t1, t2 R
WHERE t1.rowid < t2.rowid
CONDITION ON $A IS t1.$A = t2.$A
```

Base of the closure system

- RQL generates an SQL query to compute the base

tuples	t1.A	t1.B	t1.C	t2.A	t2.B	t2.C	base
τ_1, τ_2	1	2	1	1	2	4	{A, B}
τ_1, τ_3	1	2	1	2	3	4	$\emptyset$
τ_2, τ_3	1	2	4	2	3	4	{C}

$$base_Q = \{\{A, B\}, \emptyset, \{C\}\}$$

Example: functional dependencies (2)

tuples	t1.A	t1.B	t1.C	t2.A	t2.B	t2.C	base
τ_1, τ_2	1	2	1	1	2	4	{A, B}
τ_1, τ_3	1	2	1	2	3	4	$\emptyset$
τ_2, τ_3	1	2	4	2	3	4	{C}

$$base_Q = \{\{A, B\}, \emptyset, \{C\}\}$$

Results computable directly from this base

- **Counter-examples**: elements of $base_Q$ contradicting the rule e.g. A $\rightarrow$ C is refuted by {A, B}, i.e. (τ_1, τ_2)
- **Rule verification**: whether a given rule has counter-examples
- **Closure** of a set of attributes e.g. $B_Q^+ = \{A, B\}$
- Canonical cover of **excluded rules**: A B $\not\rightarrow$ C, C $\not\rightarrow$ A, C $\not\rightarrow$ B
- Canonical cover of **verified rules**: A $\rightarrow$ B, B $\rightarrow$ A

Operations on the base (1)

Counter-examples

- Elements of $base_Q$ contradicting a rule $X \rightarrow Y$

$$C_{X \rightarrow Y} = \{E \in base_Q : X \subseteq E \wedge Y \nsubseteq E\}$$

Rule verification

- Check if a rule $X \rightarrow Y$ has no counter-examples

$$V_{X \rightarrow Y} = \forall\, E \in base_Q (X \nsubseteq E \vee Y \subseteq E)$$

Closure

- Closure of a set of attributes X

$$X_Q^+ = \bigcap \{E \in base_Q : X \subseteq E\}$$

Operations on the base (2)

Canonical cover of excluded rules

- Rules that become verified if any attribute is added on the left-hand part

$$Ex_Q = \{X \nrightarrow A : X \in base_Q \wedge A \notin X \wedge \nexists Y \in base_Q (A \notin Y \wedge Y \supset X)\}$$

i.e. X are the greatest (in the inclusion sense) elements in $base_Q$ that are not a superset of A

Canonical cover of verified rules

- Complements of Ex_Q define an hypergraph $H = \{(A, \overline{X})\}$
- Verified rules $\{(A, Y)\}$ are the transversal hypergraph of H

Functional dependencies

Sample relation EMP

Empno	Lastname	Workdept	Job	Educlevel	Sex	Sal	Bonus	Comm	Mgrno
10	SPEN	C01	FINANCE	18	F	52750	500	4220	20
20	THOMP	-	MANAGER	18	M	41250	800	3300	-
30	KWAN	-	FINANCE	20	F	38250	500	3060	10
50	GEYER	-	MANAGER	16	M	40175	700	3214	20
60	STERN	D21	SALE	14	M	32250	500	2580	30
70	PULASKI	D21	SALE	16	F	36170	700	2893	100
90	HENDER	D21	SALE	17	F	29750	500	2380	10
100	SPEN	C01	FINANCE	18	M	26150	800	2092	20

```
FINDRULES
OVER Empno, Lastname, Workdept, Job, Sex, Bonus, Mgrno
SCOPE t1, t2 Emp
CONDITION ON $A IS t1.$A = t2.$A
```

Implications (on null values)

Sample relation EMP

Empno	Lastname	Workdept	Job	Educlevel	Sex	Sal	Bonus	Comm	Mgrno
10	SPEN	C01	FINANCE	18	F	52750	500	4220	20
20	THOMP	-	MANAGER	18	M	41250	800	3300	-
30	KWAN	-	FINANCE	20	F	38250	500	3060	10
50	GEYER	-	MANAGER	16	M	40175	700	3214	20
60	STERN	D21	SALE	14	M	32250	500	2580	30
70	PULASKI	D21	SALE	16	F	36170	700	2893	100
90	HENDER	D21	SALE	17	F	29750	500	2380	10
100	SPEN	C01	FINANCE	18	M	26150	800	2092	20

```
FINDRULES
OVER Empno, Lastname, Workdept, Job, Sex, Bonus, Mgrno
SCOPE t1 Emp
CONDITION ON $A IS t1.$A IS NULL
```

Conditional functional dependencies

Sample relation EMP

Empno	Lastname	Workdept	Job	Educlevel	Sex	Sal	Bonus	Comm	Mgrno
10	SPEN	C01	FINANCE	18	F	52750	500	4220	20
20	THOMP	-	MANAGER	18	M	41250	800	3300	-
30	KWAN	-	FINANCE	20	F	38250	500	3060	10
50	GEYER	-	MANAGER	16	M	40175	700	3214	20
60	STERN	D21	SALE	14	M	32250	500	2580	30
70	PULASKI	D21	SALE	16	F	36170	700	2893	100
90	HENDER	D21	SALE	17	F	29750	500	2380	10
100	SPEN	C01	FINANCE	18	M	26150	800	2092	20

```
FINDRULES
OVER Empno, Lastname, Workdept, Job, Sex, Bonus
SCOPE t1, t2 (SELECT * FROM Emp WHERE Educlevel > 16)
CONDITION ON $A IS t1.$A = t2.$A
```

Approximate functional dependencies

Sample relation EMP

Empno	Lastname	Workdept	Job	Educlevel	Sex	Sal	Bonus	Comm	Mgrno
10	SPEN	C01	FINANCE	18	F	52750	500	4220	20
20	THOMP	-	MANAGER	18	M	41250	800	3300	-
30	KWAN	-	FINANCE	20	F	38250	500	3060	10
50	GEYER	-	MANAGER	16	M	40175	700	3214	20
60	STERN	D21	SALE	14	M	32250	500	2580	30
70	PULASKI	D21	SALE	16	F	36170	700	2893	100
90	HENDER	D21	SALE	17	F	29750	500	2380	10
100	SPEN	C01	FINANCE	18	M	26150	800	2092	20

```
FINDRULES
OVER Educlevel, Sal, Bonus, Comm
SCOPE t1, t2 Emp
CONDITION ON $A IS 2*ABS(t1.$A-t2.$A)/(t1.$A+t2.$A) < 0.1
```

Sequential dependencies

Sample relation EMP

Empno	Lastname	Workdept	Job	Educlevel	Sex	Sal	Bonus	Comm	Mgrno
10	SPEN	C01	FINANCE	18	F	52750	500	4220	20
20	THOMP	-	MANAGER	18	M	41250	800	3300	-
30	KWAN	-	FINANCE	20	F	38250	500	3060	10
50	GEYER	-	MANAGER	16	M	40175	700	3214	20
60	STERN	D21	SALE	14	M	32250	500	2580	30
70	PULASKI	D21	SALE	16	F	36170	700	2893	100
90	HENDER	D21	SALE	17	F	29750	500	2380	10
100	SPEN	C01	FINANCE	18	M	26150	800	2092	20

```
FINDRULES
OVER Educlevel, Sal, Bonus, Comm
SCOPE t1, t2 Emp
CONDITION ON $A IS t1.$A > t2.$A
```

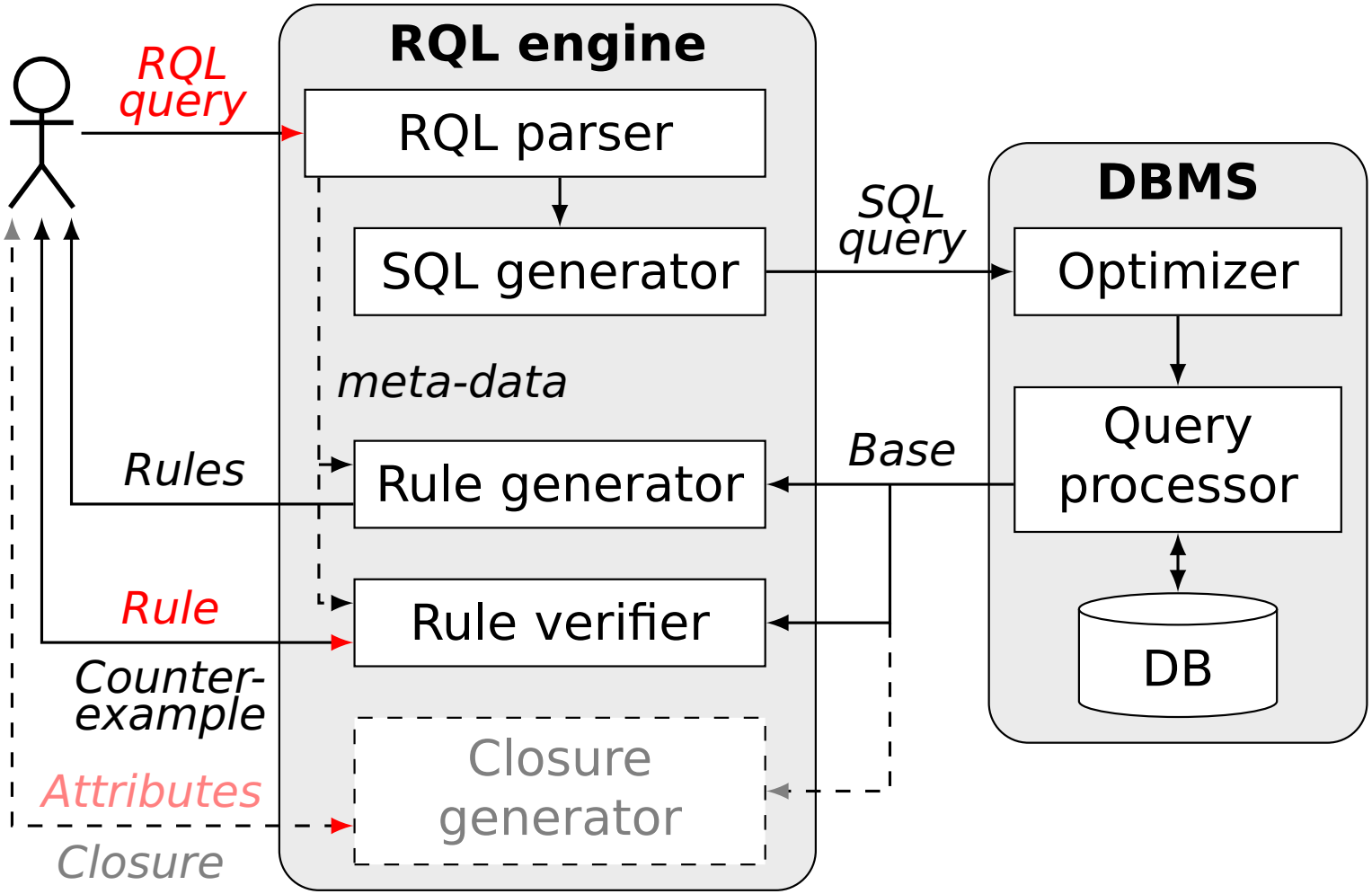
Ad-hoc rules

Sample relation EMP

Empno	Lastname	Workdept	Job	Educlevel	Sex	Sal	Bonus	Comm	Mgrno
10	SPEN	C01	FINANCE	18	F	52750	500	4220	20
20	THOMP	-	MANAGER	18	M	41250	800	3300	-
30	KWAN	-	FINANCE	20	F	38250	500	3060	10
50	GEYER	-	MANAGER	16	M	40175	700	3214	20
60	STERN	D21	SALE	14	M	32250	500	2580	30
70	PULASKI	D21	SALE	16	F	36170	700	2893	100
90	HENDER	D21	SALE	17	F	29750	500	2380	10
100	SPEN	C01	FINANCE	18	M	26150	800	2092	20

```
FINDRULES
OVER Educlevel, Sal, Bonus, Comm
SCOPE t1, t2 Emp
WHERE t1.Empno = t2.Mgrno
CONDITION ON $A IS t1.$A >= t2.$A
```

Architecture



- The RQL Engine is a Java application using the Play Framework

Web application (http://rql.insa-lyon.fr)

- **Unified interface** for data querying (SQL) and data mining (RQL)
- RQL queries allow multiple feedbacks
  - **comprehensive rule generation**: canonical cover of rules
  - **rule verification**: single rule checking with counter-examples (if any)

Conclusion

- Attempt to bridge the gap between pattern mining and databases
- Compatible with **implications**, functional dependencies (**FDs**), conditional or approximate FDs (**CFDs**, **AFDs**) and many more **custom rules**
- Applications to **data cleaning**, **database understanding**, **bioinformatics**, **education**...
- **Web application** accessible to everyone: *give it a try!*