# From SQL to RQL

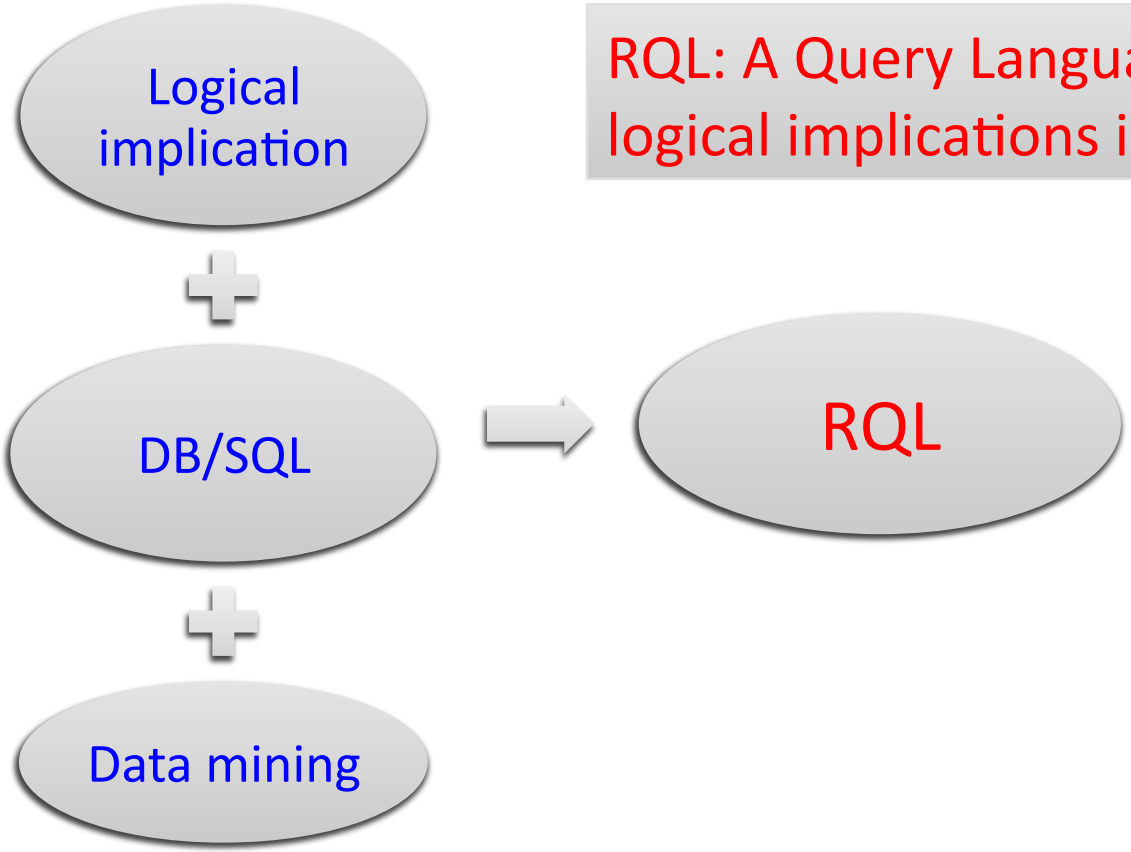
or how to re-use SQL techniques for pattern mining

**Jean-Marc Petit**
INSA/Université de Lyon, CNRS

*Joint work with* : Brice Chardin, Emmanuel Coquery, Christine Froidevaux, Marie Paillloux (Agier), Jef Wijsen … and Einoshin Suzuki !

*Work partially funded by the French Research Agency (ANR), DAG project*

# RQL in a nutshell




RQL: A Query Language for discovering logical implications in DB

# About logical implications

- Logical consequence (or entailment)
  - One of the most fundamental concepts in logic
  - Premises, Conclusion (If … then …): Reasoning using proofs and/or models
  - Examples
    - `` ***If*** *2=3* ***then*** *I am the queen of England ''*
      - right or wrong ? Why ?
    - `` ***If*** *2=2* ***then*** *I am the queen of England ''*
      - right or wrong ? Why ?
- Focus on a specific class of logical implications
  - Three properties to be verified
    - Reflexivity, augmentation, transitivity (Armstrong axioms)

# About databases

- Relational databases systems everywhere !
  - RDBMS market expected to double by 2016 [MarketResarch.com, Aug 2012]
  - Query optimization: Awesome !
- Simple goals :
  - Query the data where they are
  - Use/extend DB languages for pattern mining problems (e.g. DMQL, MSQL, …)

# About Data Mining

- Focus on pattern mining in DB
  - patterns = **logical implication, called rules hereafter**
  - DB= **Relational DBs**
  - **Quality measure: not considered**
- Pattern mining discovery seen as query processing
  - Reusing optimization techniques from DB

# RQL: contributions

- Original ideas
  - *Marie Agier, Jean-Marc Petit, Einoshin Suzuki: Unifying Framework for Rule Semantics: Application to Gene Expression Data. Fundam. Inform. 78(4): 543-559 (2007)*
- Since then, what we have done ?
  - SafeRL: a well-founded logical query language derived from Tuple Relational Calculus (TRC)
    - **not discussed in detail here**
  - RQL: Its practical counterpart derived from SQL
  - A rewriting technique to use as much as possible the underlying DBMS
  - A web application : http://rql.insa-lyon.fr

## Outline

- Getting started with RQL through examples
- SafeRL and Query rewriting
- RQL Web Application
- Conclusions

| EMP | Empno | Lastname | Work dept | Job | Educ level | 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 | 800 | 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 |

Rules between attributes with NULL values in EMP ?

## Motivating examples

```
FINDRULES
OVER workdept, mgrno
SCOPE t1 EMP
CONDITION ON $A IS t1.$A IS NULL
```

- Mgrno -> Workdept holds
- Workdept -> Mgrno does not
  => counter-example: Empno 30 (or 50)

# Functional dependencies

- Remember the definition:
  - X->Y holds in r

    iff $\forall$ **$t_1$, $t_2$** $\in$ r,

    **if** $\forall A \in X$, $t_1[A]=t_2[A]$ **then** $\forall B \in Y$, $t_1[B]=t_2[B]$
- With RQL:

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

# Variant of FDs: Conditional FDs

- FINDRULES
  OVER Lastname, Workdept, Job, Sex, Bonus
  SCOPE t1, t2 **(select * from Emp where educlevel > 16)**
  CONDITION ON $A IS t1.$A = t2.$A

  sex -> bonus holds
  i.e. « above a certain level of qualification, the gender determines the bonus »

# Approximative FDs

- 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**

  Sal -> Comm holds
  i.e. « employees earning similar salaries receive similar commissions »

# Sequential FDs

- FINDRULES
  OVER Educlevel, Sal, Bonus, Comm
  SCOPE t1, t2 EMP
  CONDITION ON $A t1.$A >= t2.$A

  Sal -> Comm and Sal -> Comm hold

  i.e. « higher salary is equivalent to higher commission »

# Conditional sequential FDs

- FINDRULES
  OVER Educlevel, Sal, Bonus, Comm
  SCOPE t1, t2 **(select * from EMP where Sex = 'M')**
  CONDITION ON $A t1.$A >= t2.$A

  EducLevel -> Bonus holds

  i.e. « male employees with higher education levels receive higher bonus »

# Another kind of « FD »

```
FINDRULES
OVER Educlevel, Sal, Bonus, Comm
SCOPE t1, t2 EMP
WHERE t1.empno = t2.mgrno
CONDITION ON $A t1.$A >= t2.$A

{} -> Bonus holds
```

i.e. « managers always earn a bonus greater than or equal to their employees»

# Another example (1/2)

- Example from gene expression data
- Assume tuples are ordered
- Idea: adapting FD for catching the evolution of attributes between two consecutive tuples

| | g1 | g2 | g3 | g4 | g5 | g6 | g7 | g8 |
|---|---|---|---|---|---|---|---|---|
| t1 | 1,9 | 0,4 | 1,4 | -1,5 | 0,3 | 1,8 | 0,8 | -1,4 |
| t2 | 1,7 | 1,5 | 1,2 | -0,3 | 1,4 | 1,6 | 0,7 | 0,0 |
| t3 | 1,8 | -0,7 | 1,3 | 0,8 | -0,1 | 1,7 | 0,9 | 0,6 |
| t4 | -1,8 | 0,4 | 1,7 | 1,8 | 0,6 | -0,4 | 1,0 | 1,5 |
| t5 | -1,7 | -1,4 | 0,9 | 0,5 | -1,8 | -0,2 | 1,2 | 0,3 |
| t6 | 0,0 | 1,9 | -1,9 | 1,7 | 1,7 | -0,5 | 1,1 | 1,3 |

$X \Rightarrow Y$ is satisfied in r iff $\forall\, t_i, t_{i+1} \in r$,

**if** $\forall g \in X,\ t_{i+1}[g] - t_i[g] \geq \varepsilon_1$ **then** $\forall g \in Y,\ t_{i+1}[g] - t_i[g] \geq \varepsilon_1$

$\varepsilon_1 = 1.0$

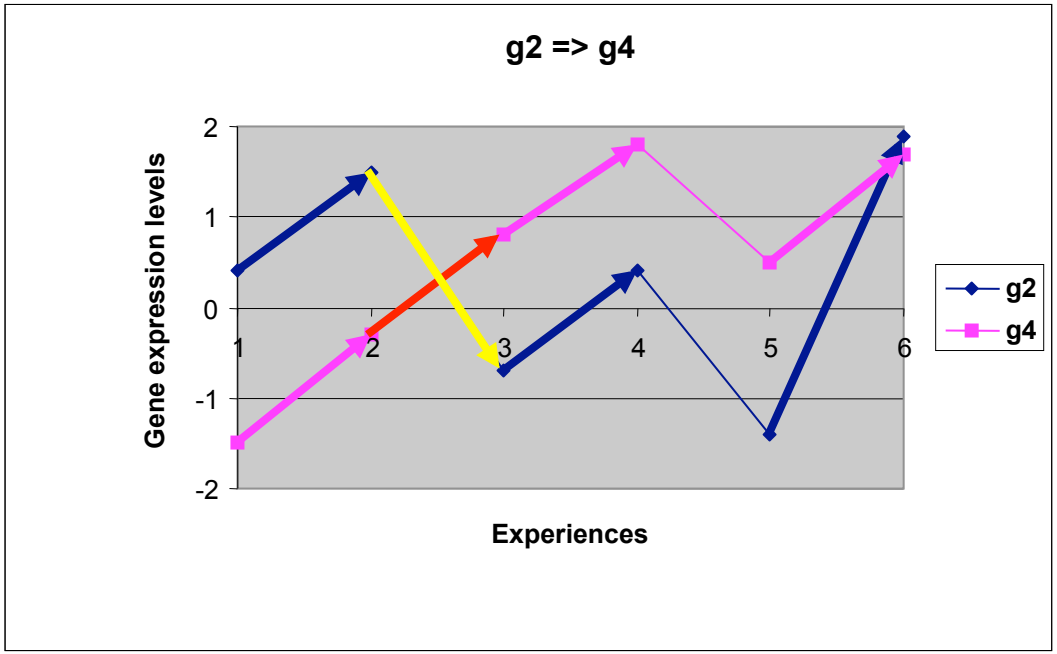
# Rules over genes

$X \Rightarrow Y$ is satisfied in r iff $\forall\ t_i, t_{i+1} \in r$,
**if** $\forall g \in X, t_{i+1}[g] - t_i[g] \geq \varepsilon_1$ **then** $\forall g \in Y, t_{i+1}[g] - t_i[g] \geq \varepsilon_1$
$\varepsilon_1 = 1.0$

```
FINDRULES
OVER g1,g2,g3,g4,g5,g6,g7,g8
SCOPE t1, t2 GENES
WHERE t2.time= t1.time+1
CONDITION ON A IS t2.A - t1.A >= 1.0
```

# Another kind of rules (2/2)

| | g1 | g2 | g3 | g4 | g5 | g6 | g7 | g8 |
|---|---|---|---|---|---|---|---|---|
| t1 | 1,9 | 0,4 | 1,4 | -1,5 | 0,3 | 1,8 | 0,8 | -1,4 |
| t2 | 1,7 | 1,5 | 1,2 | -0,3 | 1,4 | 1,6 | 0,7 | 0,0 |
| t3 | 1,8 | -0,7 | 1,3 | 0,8 | -0,1 | 1,7 | 0,9 | 0,6 |
| t4 | -1,8 | 0,4 | 1,7 | 1,8 | 0,6 | -0,4 | 1,0 | 1,5 |
| t5 | -1,7 | -1,4 | 0,9 | 0,5 | -1,8 | -0,2 | 1,2 | 0,3 |
| t6 | 0,0 | 1,9 | -1,9 | 1,7 | 1,7 | -0,5 | 1,1 | 1,3 |




$r \models g_2 \Rightarrow g_4$

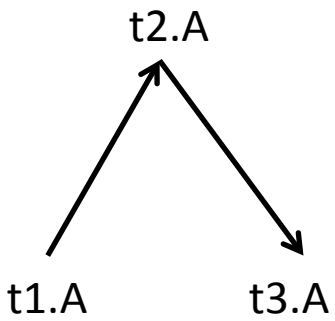
**If** expression level of $g_2$ grows between $t_i$ and $t_{i+1}$,

**then** expression level of $g_4$ also grows

But $r \not\models g_4 \Rightarrow g_2$

$t_2, t_3$ is a couter-example

# Local maximum




```
FINDRULES
OVER g1,g2,g3,g4,g5,g6,g7,g8
SCOPE t1, t2, t3 Genes
WHERE t2.time=t1.time+1 AND t3.time=t2.time
   +1
CONDITION ON $A IS t1.$A < t2.$A AND t3.$A<
   t2.$A
```

=> three tuples variables needed to express a local maximum

# Synthesis of RQL

- RQL : « look & feel » of SQL
  - Very simple and easy to use by SQL analysts
- Powerful query language
  - Allow interactions with data analysts
  - Powerful tool, need some practice to get fluent with …
- Can be used
  - To generate the rules (if schema permits)
  - To test whether or not a given rule holds
    - If yes, just say « Yes » ☺
    - Otherwise, find a counter-examples in the data and refine your query

- Motivating examples
- SafeRL and Query rewriting
- RQL Web Application
- Conclusions

## Outline

# SafeRL: a query language for rules

- SafeRL: a well-founded logical query language

$$Q=\{X\rightarrow Y \mid \forall t1 \ldots \forall tn(\psi(t1,\ldots,tn) \wedge (\forall A \in X(\delta(A,t1,\ldots,tn)) \rightarrow \forall A \in Y(\delta(A, t1, \ldots, tn))))\}$$

  - Syntax + semantics **not detailed here**: cf papers

- Every SafeRL query Q defines rules "equivalent to" FD or implications

  - Result: There is a closure system C(Q) associated to Q

# Contribution: Query rewriting

- A base B of a closure system C is such that
  - Irreducible(C) $\subseteq$ B $\subseteq$ C
- Main result (cf LML 2013):
  Let Q be a SafeRL query over a DB d

  THM : There exists a *SQL query* Q' over d such that Q' computes a base B of C(Q), the closure system associated to Q

# Base of a query: the data-centric step

From the base B of Q in d, we can get:

- The **closure** of an attribute set
- The **canonical cover** of satisfied rules
- The cover of Gotlob&Libkin of **approximate rules**

and we can decide whether or not a given rule is satisfied

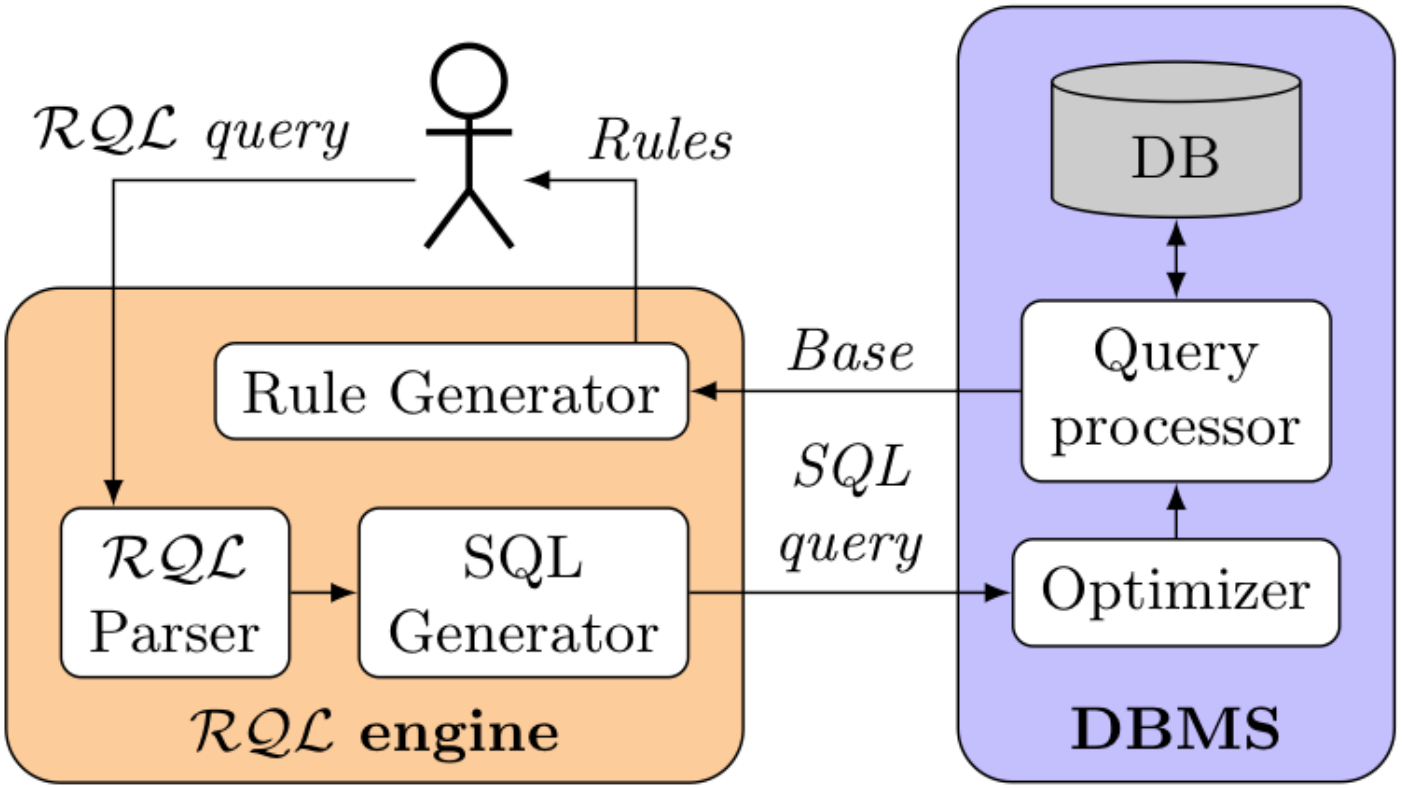
- If not, a **counter example** from d can be provided

- *Nothing new here*, cf related works

- Motivating examples
- SafeRL and Query rewriting
- RQL Web Application
- Conclusions

## Outline

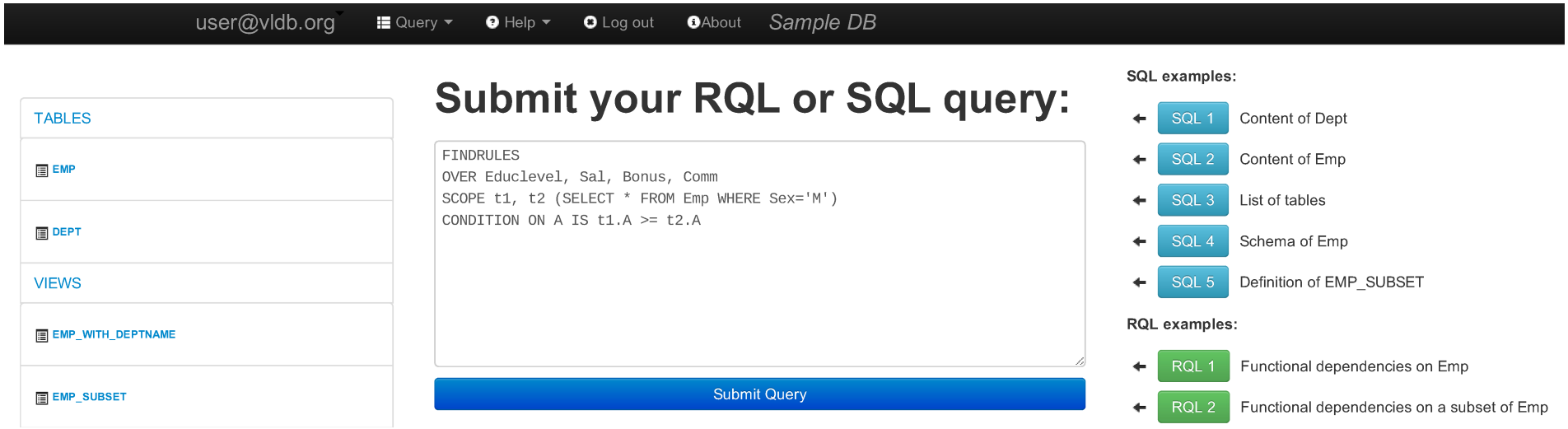
# Architecture



# RQL Web application

- Open to registered users (simple application form)
  - dedicated Oracle user, 200Ko quota
- Web Framework For Java
  - Play Framework (http://www.playframework.com/)
  - DBMS: Oracle v11 (+ MySQL)
  - + specific development in C++, Java, C (Uno's code)
- Two modes: Sample (predefined schema) and SandBox (user schema)
- Try it out ! http://rql.insa-lyon.fr

# Snapshot: sample mode



# Counter example

## Rule verification:

The rule **Sal Educlevel ➔ Bonus** is **false**

**Counter-example:**

| EMPNO | LASTNAME | WORKDEPT | JOB | EDUCLEVEL | SEX | SAL | BONUS | COMM | MGRNO |
|---|---|---|---|---|---|---|---|---|---|
| 10 | SPEN | C01 | FINANCE | 18 | F | 52750 | 500 | 4220 | 20 |
| 20 | THOMP | null | MANAGER | 18 | M | 41250 | 800 | 3300 | null |

**Generated query:**

```
1.  SELECT t1.*, t2.*
2.  FROM Emp t1, Emp t2
3.  WHERE (t1.Sal >= t2.Sal AND t1.Educlevel >= t2.Educlevel)
4.  AND CASE WHEN (t1.Bonus >= t2.Bonus) THEN 1 ELSE 0 END = 0
5.  AND rownum <= 1
```

- Motivating examples
- RQL Web Application
- SafeRL and Query rewriting
- Conclusions

## Outline

# Conclusion

- From a logical query language for rules to the practical language RQL
  - Easy to use by SQL-aware analysts
  - No discretization
  - Promoting query processing techniques in pattern mining
- RQL: a practical Web application
  - For teaching
  - For research
- Future works: **Data exploration with RQL through counter-examples**

- ***Query Rewriting for Rule Mining in Databases***. *B. Chardin, E Coquery, B. Gouriou, M. Pailloux, J-M Petit. In Languages for Data Mining and Machine Learning (LML) Workshop@ECML/PKDD 2013, Bruno Crémilleux, Luc De Raedt, Paolo Frasconi, Tias Guns ed. Prague. pp. 1-16. 2013*

- ***On Armstrong-compliant Logical Query Languages***. *M. Agier, C. Froidevaux, J-M Petit, Y. Renaud, J. Wijsen. Dans 4th International Workshop on Logic in Databases (LID 2011) collocated with the 2011 EDBT/ICDT conference, Sweeden. pp. 33-40. ACM*

- ***Unifying Framework for Rule Semantics: Application to Gene Expression Data***. *Marie Agier, Jean-Marc Petit, Einoshin Suzuki. Fundam. Inform. 78(4): 543-559 (2007)*

**Merci !**
**Questions ?**

Work partially supported by the French ANR project DAG (ANR DEFIS 2009)