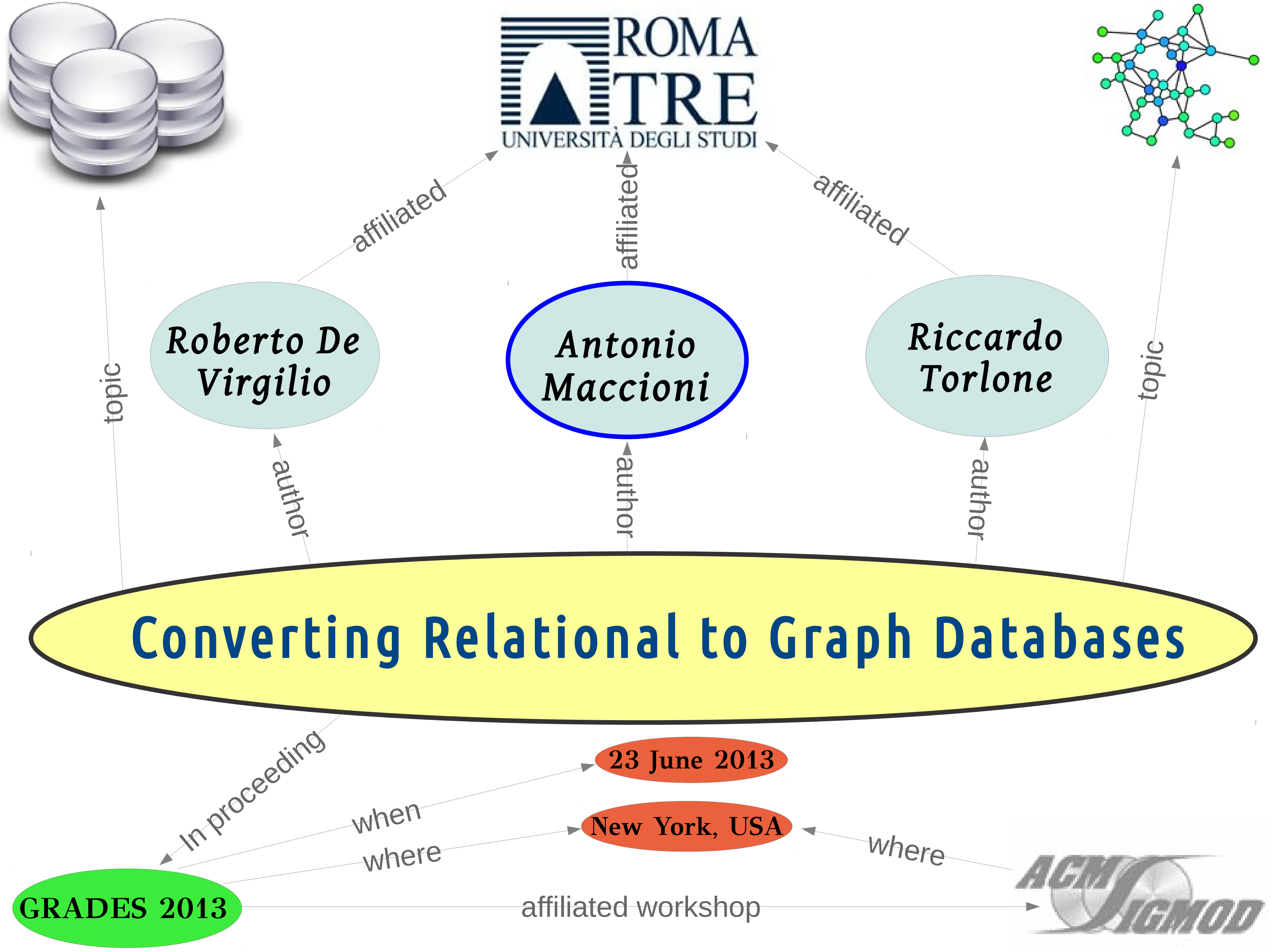


Roberto De Virgilio

Antonio Maccioni

Riccardo Torlone

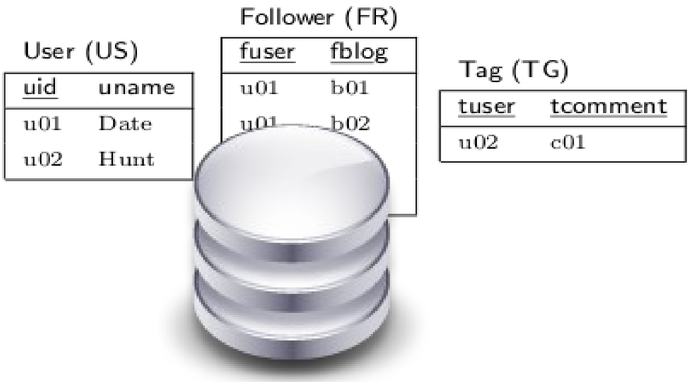
Converting Relational to Graph Databases

23 June 2013

New York, USA

GRADES 2013

Relational Database Migration



User (US)

uid	uname
u01	Date
u02	Hunt

Follower (FR)

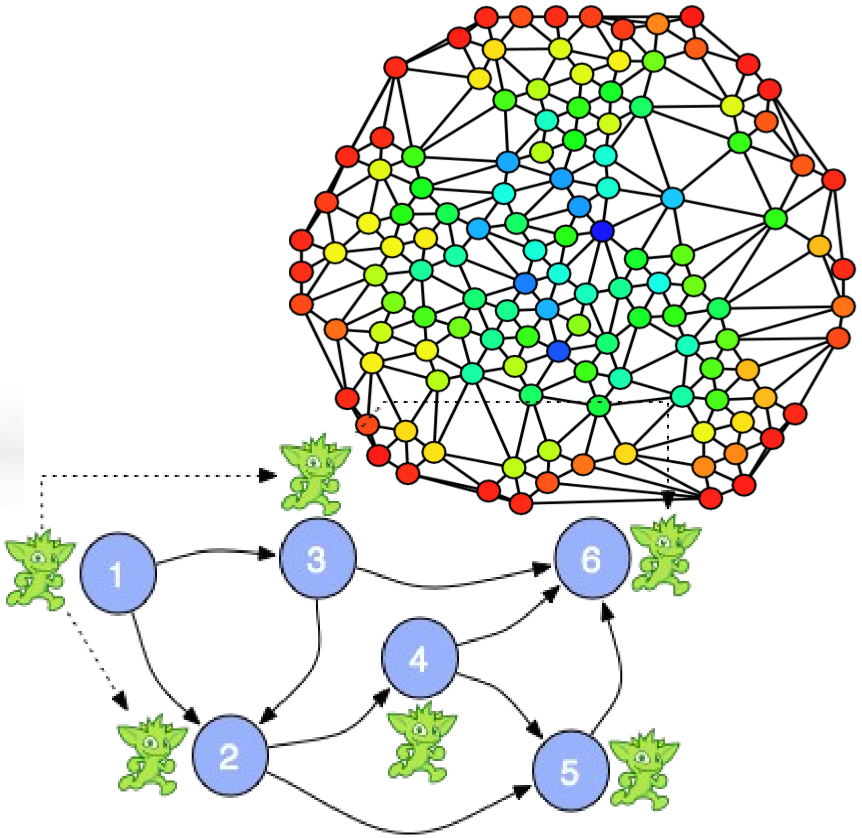
fuser	fblog
u01	b01
u01	b02

Tag (TG)

tuser	tcomment
u02	c01

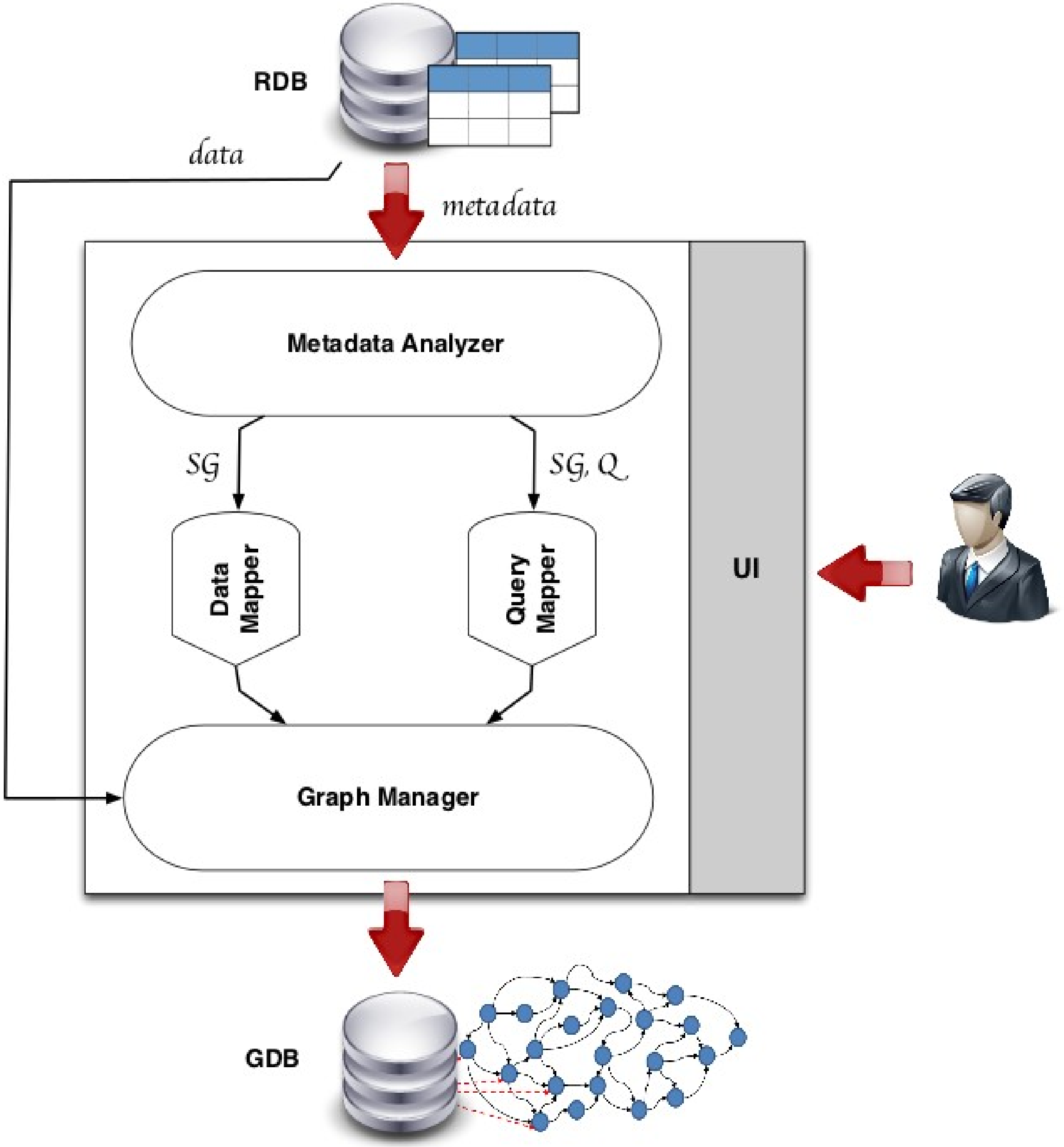
SQL

```
select *
from T
where T.A1 = v1
```

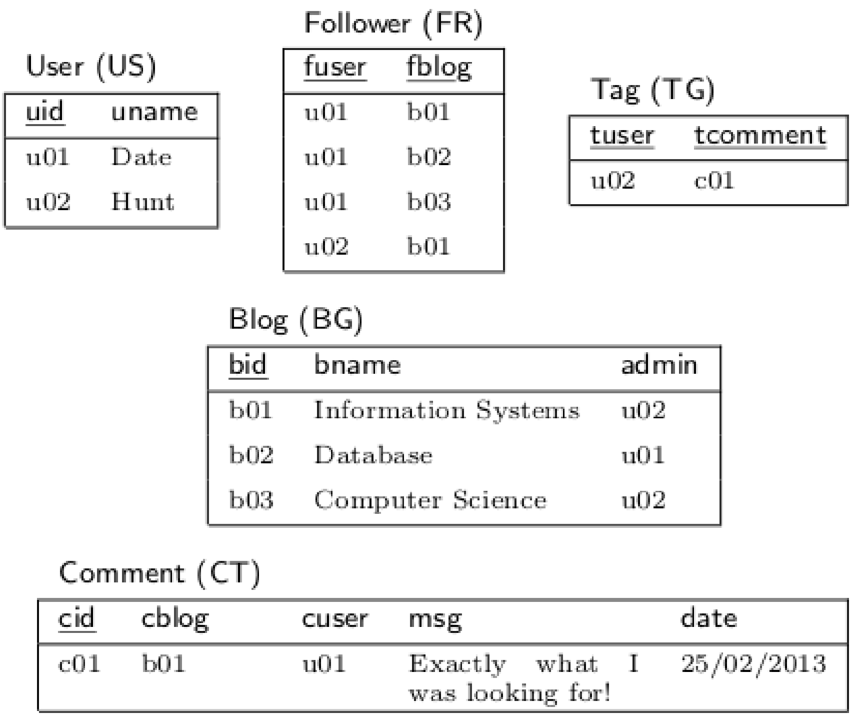


R2G: Features

- Data migration
- Query translation
- Automatic non-naïve approach
- Try to minimize the memory accesses



Graph Modeling of Relational DB



User (US)

uid	uname
u01	Date
u02	Hunt

Follower (FR)

fuser	fblog
u01	b01
u01	b02
u01	b03
u02	b01

Tag (TG)

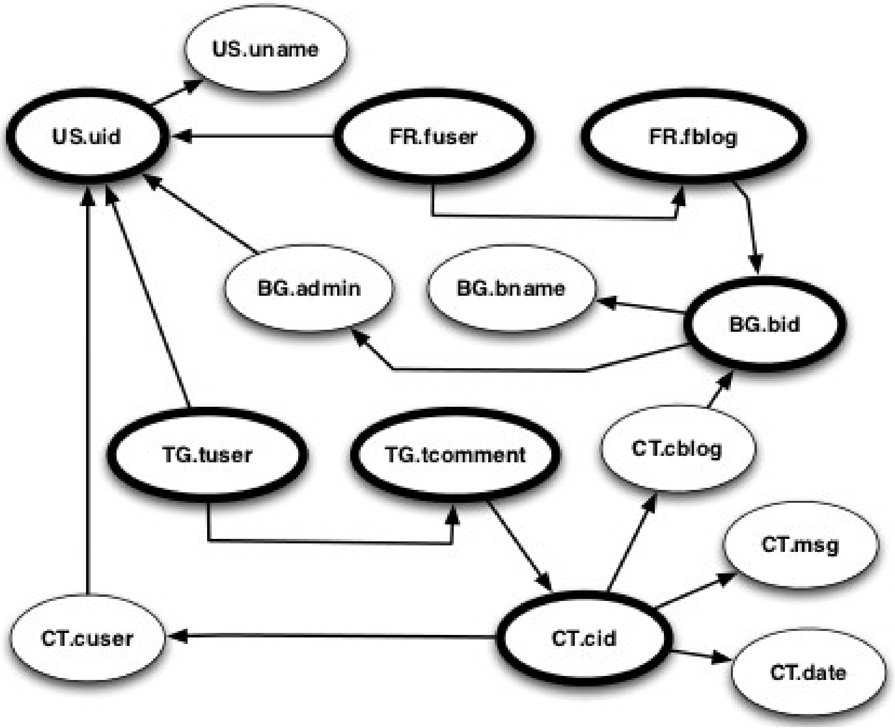
tuser	tcomment
u02	c01

Blog (BG)

bid	bname	admin
b01	Information Systems	u02
b02	Database	u01
b03	Computer Science	u02

Comment (CT)

cid	cblog	cuser	msg	date
c01	b01	u01	Exactly what I was looking for!	25/02/2013



- Full Schema Paths:

  FR.fuser → US.uid → US.uname

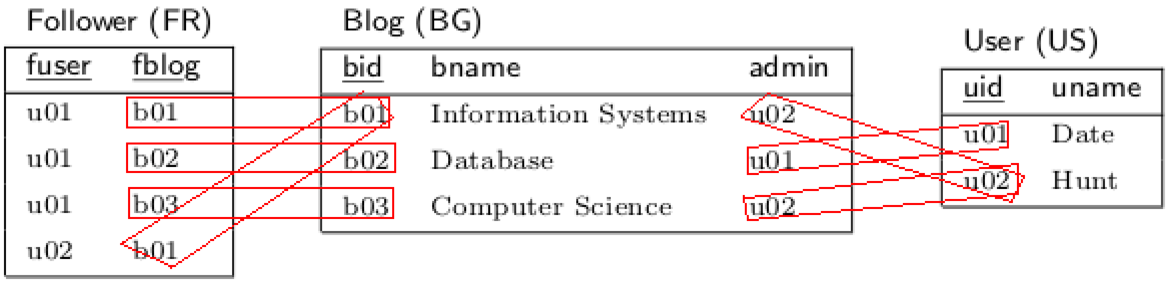
  FR.fuser → FR.fblog → BG.bid → BG.bname

  FR.fuser → FR.fblog → BG.bid → BG.admin → US.uid → US.uname

  ...

Basic Concepts

- Joinable tuples $t_1 \in R_1$ and $t_2 \in R_2$:
  - there is a foreign key constraint between R1.A and R2.B and t1[A] = t2[B].
- Unifiability of data values $t_1[A]$ and $t_2[B]$:
  - (i) $t_1=t_2$ and both A and B do not belong to a multi-attribute key;
  - (ii) t_1 and t_2 are joinable and A belongs to a multi-attribute key;
  - (iii) t_1 and t_2 are joinable, A and B do not belong to a multi-attribute key and there is no other tuple t_3 that is joinable with t_2.



Follower (FR)

fuser	fblog
u01	b01
u01	b02
u01	b03
u02	b01

Blog (BG)

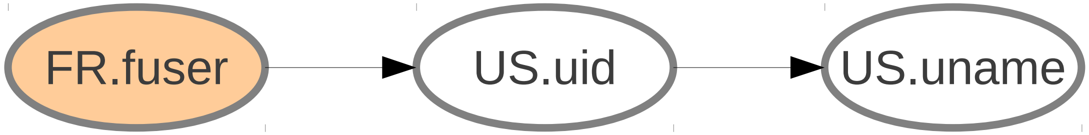
bid	bname	admin
b01	Information Systems	u02
b02	Database	u01
b03	Computer Science	u02

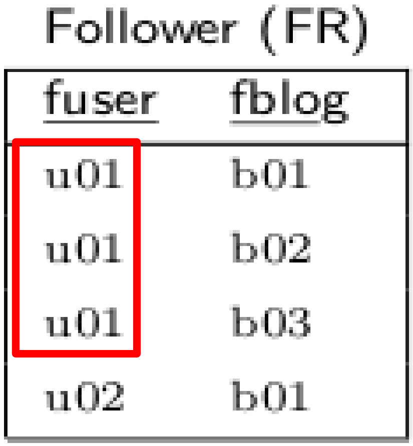
User (US)

uid	uname
u01	Date
u02	Hunt

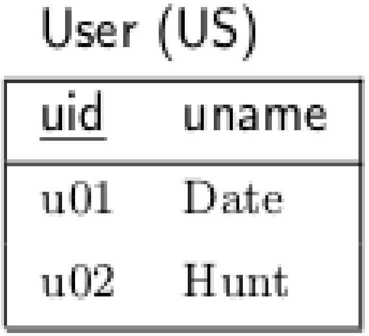
Data Migration (1)

- Identify unifiable data exploiting schema and constraints

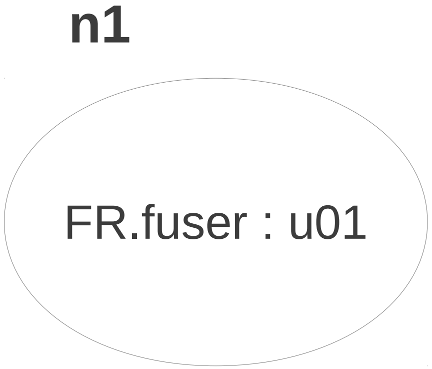

FR.fuser
US.uid
US.uname


Follower (FR)

fuser	fblog
u01	b01
u01	b02
u01	b03
u02	b01


User (US)

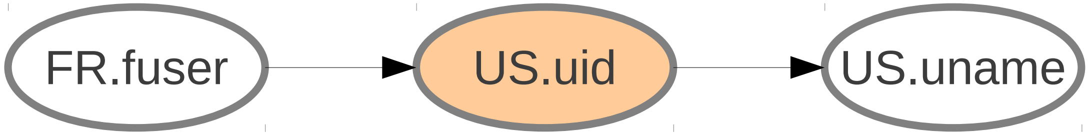
uid	uname
u01	Date
u02	Hunt

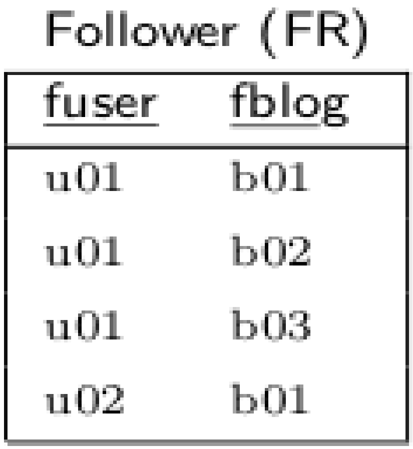

n1

FR.fuser : u01

Data Migration (2)

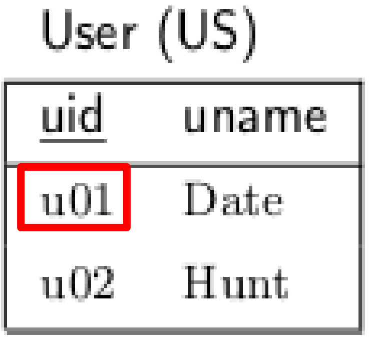
- Identify unifiable data exploiting schema and constraints





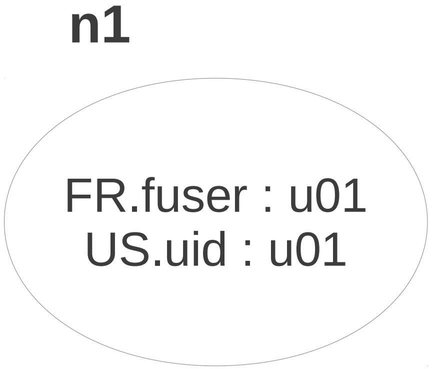
Follower (FR)

fuser	fblog
u01	b01
u01	b02
u01	b03
u02	b01



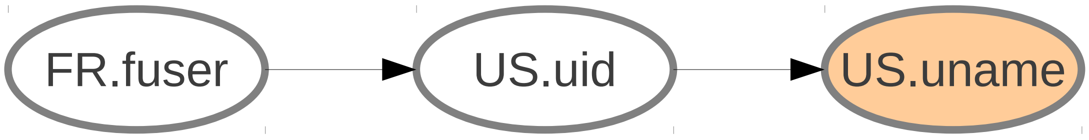
User (US)

uid	uname
u01	Date
u02	Hunt



Data Migration (3)

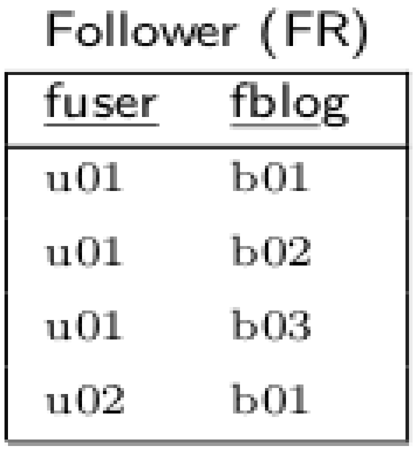
- Identify unifiable data exploiting schema and constraints



FR.fuser

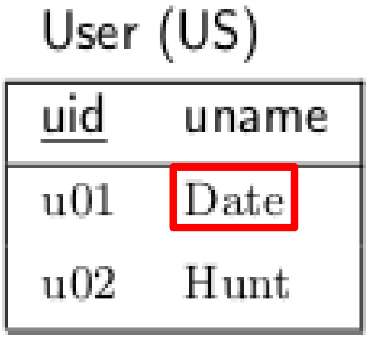
US.uid

US.uname



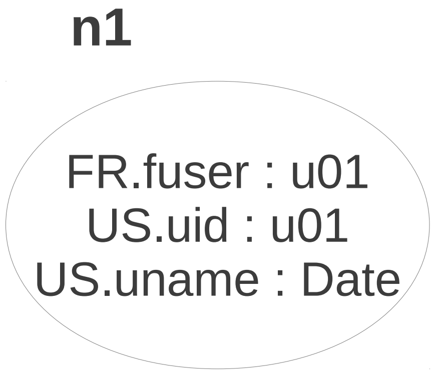
Follower (FR)

fuser	fblog
u01	b01
u01	b02
u01	b03
u02	b01



User (US)

uid	uname
u01	Date
u02	Hunt

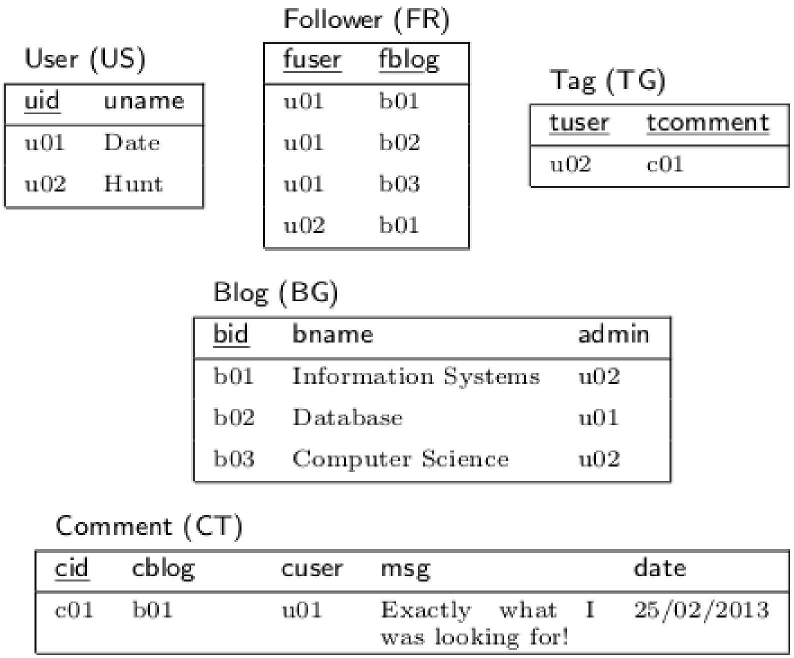


n1

FR.fuser : u01
US.uid : u01
US.uname : Date

Data Migration (4)

- Identify unifiable data exploiting schema and constraints



User (US)

uid	uname
u01	Date
u02	Hunt

Follower (FR)

fuser	fblog
u01	b01
u01	b02
u01	b03
u02	b01

Tag (TG)

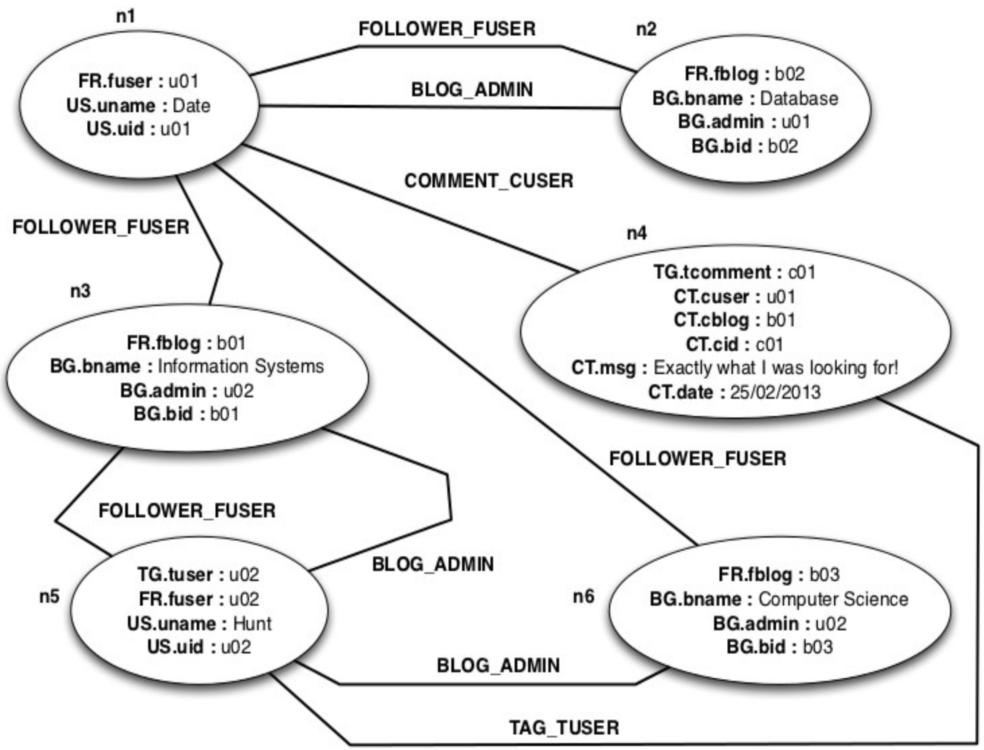
tuser	tcomment
u02	c01

Blog (BG)

bid	bname	admin
b01	Information Systems	u02
b02	Database	u01
b03	Computer Science	u02

Comment (CT)

cid	cblog	cuser	msg	date
c01	b01	u01	Exactly what I was looking for!	25/02/2013



Query Translation

```
select  US.uname
from    User US, Blog BG
where   (BG.admin = US.uid) and
        (BG.bname = 'Inf. Systems')
```

User (US)

uid	uname
...	...

Follower (FR)

fuser	fblog
...	...

Tag (TG)

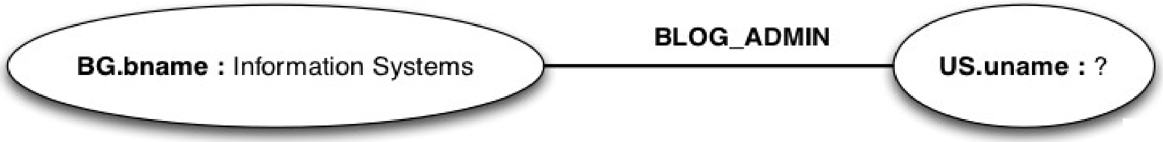
tuser	tcomment
...	...

Blog (BG)

bid	bname	admin
...	...	...

Comment (CT)

cid	cblog	cuser	msg	date
...	...	...	...	...



XQuery

```
for     $x in /[BG.bname='Inf. Systems'],
        $y in $x/BLOG_ADMIN/*
return  $y/US.uname
```

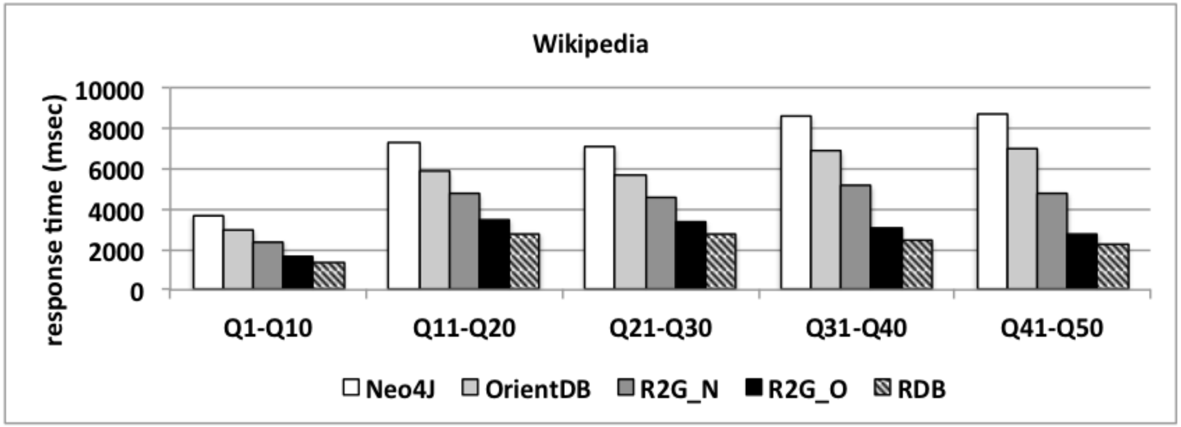
Gremlin

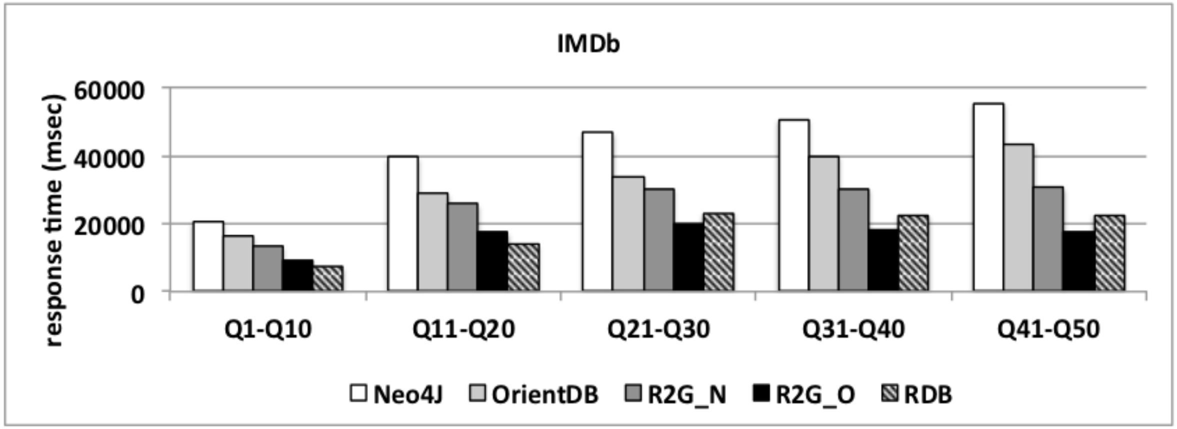
```
g.V.filter{it.BGbname=='Inf. Systems'}.
 .outE.filter{it.label=='BLOG_ADMIN'}.
.inV.USname
```

Experimental Results

Dataset	Neo4J	OrientDB	R2G_N	R2G_O
MONDIAL	7.4 sec	5.3 sec	13.9 sec	9.3 sec
WIKIPEDIA	70.7 sec	66.5 sec	161.5 sec	148.7 sec
IMDB	8.1 min	10.2 min	16.2 min	22.1 min

The MONDIAL Database





Conclusion

- Automatic data mapping
- Conjunctive query translation into a path traversal query
- Independent from a specific GDBMS
- Efficient exploitation of Graph Database Features

Future Work

- Consider frequent queries to migrate data
- Consider wider range of queries than CQ
- Improve compactness of the graph database

Thanks For The Attention

... demo presentation during the following interactive session!