# Schema Evolution and Foreign Keys: Birth, Eviction, Change and Absence

Panos Vassiliadis, Michail-Romanos Kolozoff*,

Maria Zerva, Apostolos V. Zarras

Department of Computer Science and Engineering
University of Ioannina, Hellas

* Currently @ Upcom, Hellas

**http://www.cs.uoi.gr/~pvassil/publications/2017_ER/**
**http://www.cs.uoi.gr/~pvassil/projects/schemaBiographies/**

# Research Question

In the context of schema evolution,

how do foreign keys evolve over time?

# Why is schema evolution so important?

- Software and DB **maintenance** makes up for **at least 50% of all resources spent in a project**.
- **Databases are rarely stand-alone: typically, an entire ecosystem of applications is structured around them =>**
- **Changes in the schema can impact a large** (typically, not traced) **number of surrounding app's**, without explicit identification of the impact.

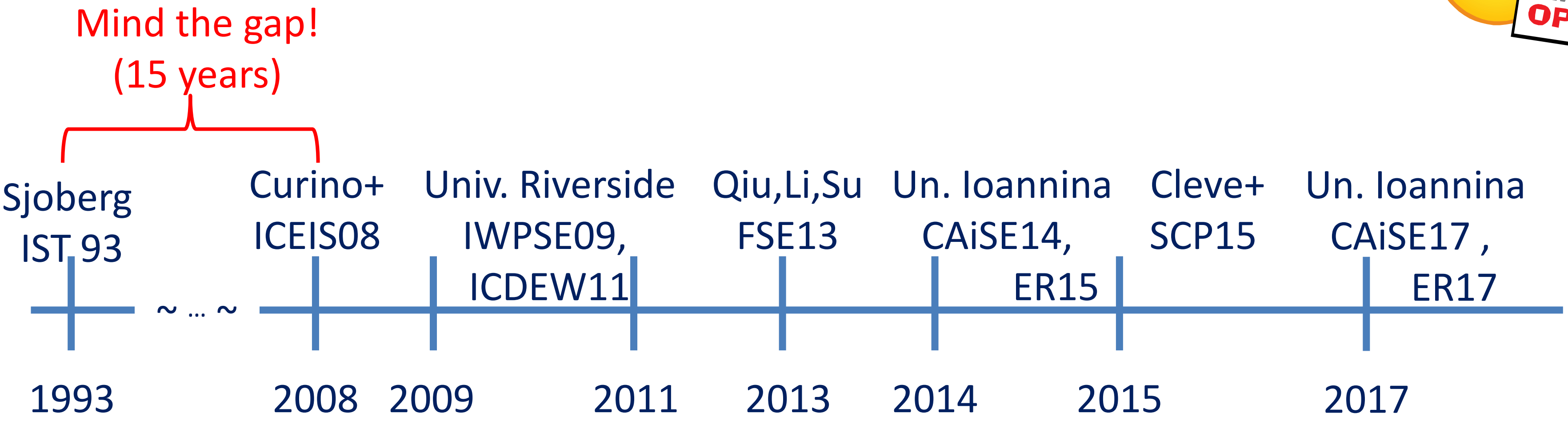
Is it possible to **"design for evolution"** and **minimize the impact of evolution** to the surrounding applications?

**… But first, we need to know the "patterns of evolution" of relational schemata! …**

# Why aren’t we there yet?

- Historically, nobody from the research community had access + the right to publish to version histories of database schemata

- Open source tools internally hosting databases have changed this landscape &
- We are now presented with the opportunity to study the version histories of such “open source databases”



Mind the gap!
(15 years)

Sjoberg IST 93 — 1993

~ … ~

Curino+ ICEIS08 — 2008

Univ. Riverside IWPSE09, ICDEW11 — 2009, 2011

Qiu,Li,Su FSE13 — 2013

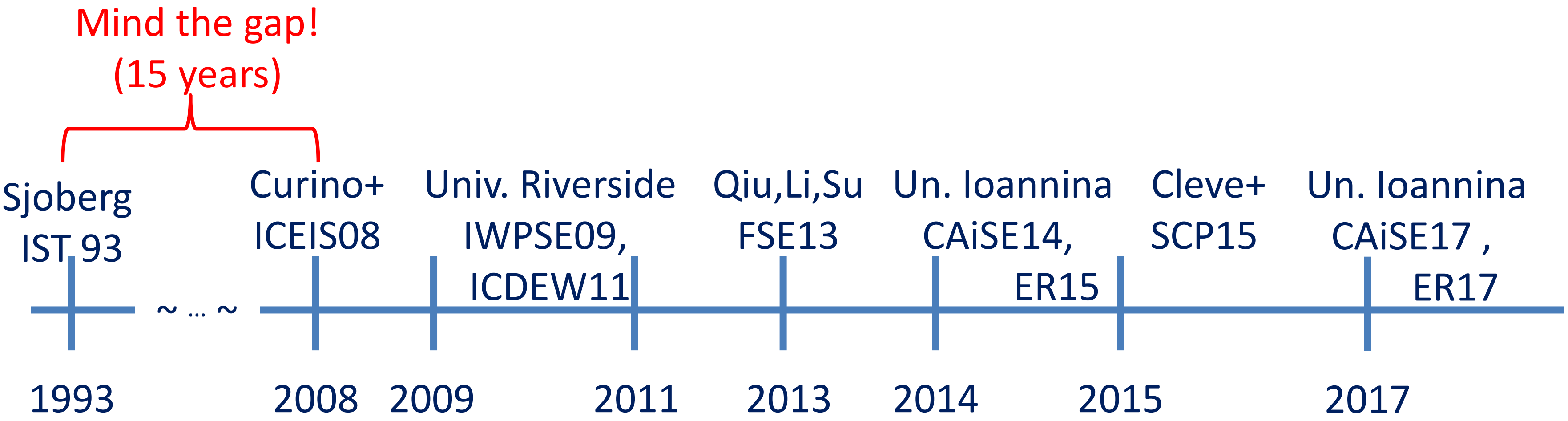
Un. Ioannina CAiSE14, ER15 — 2014

Cleve+ SCP15 — 2015

Un. Ioannina CAiSE17 , ER17 — 2017

# Why aren’t we there yet?

- In all previous attempts, the object of study was the schema size as well as the heartbeat of change,
- Patterns on table behaviors have been studied only lately.
- To the best of our knowledge, the current paper is the first comprehensive effort in the literature to study the evolution of foreign keys.



Mind the gap!
(15 years)

Sjoberg
IST 93

Curino+
ICEIS08

Univ. Riverside
IWPSE09,
ICDEW11

Qiu,Li,Su
FSE13

Un. Ioannina
CAiSE14,
ER15

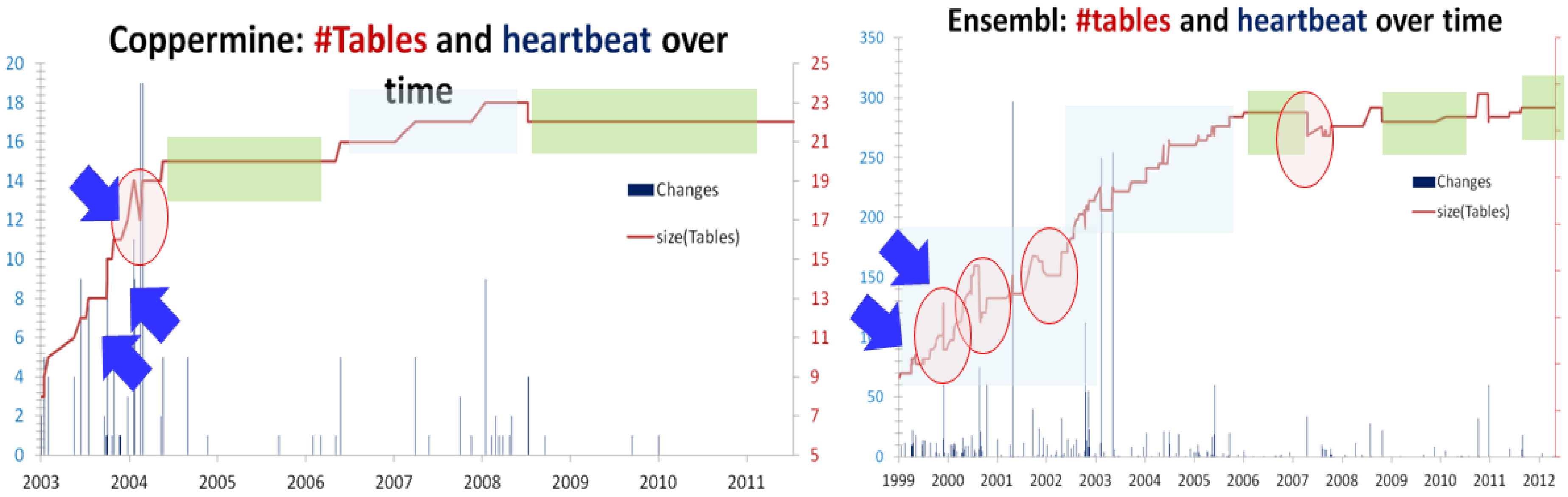
Cleve+
SCP15

Un. Ioannina
CAiSE17 ,
ER17

~ … ~

1993 2008 2009 2011 2013 2014 2015 2017

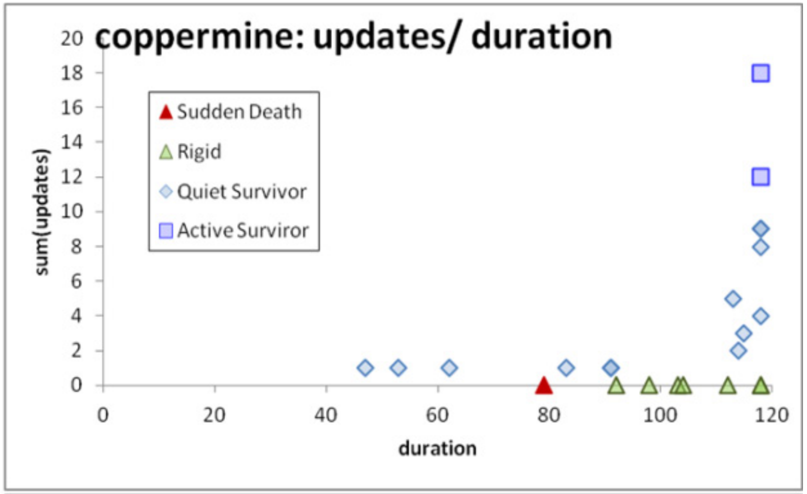
# What we have found for schema evolution [CAiSE 14, IS 15]

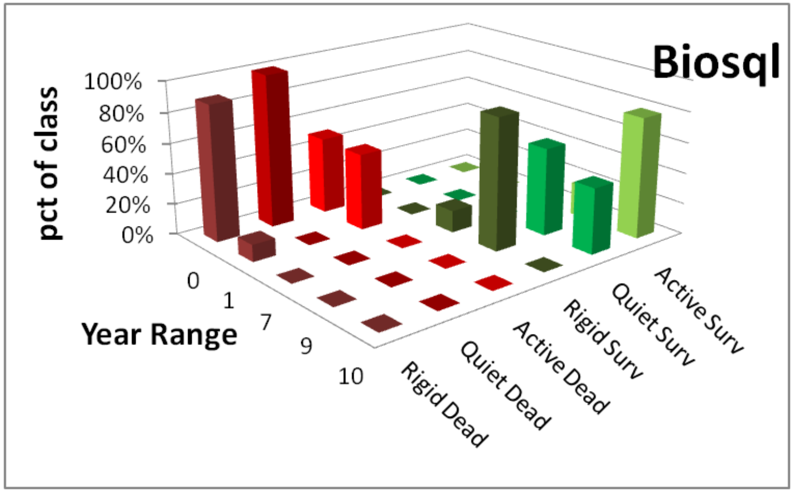

Coppermine: #Tables and heartbeat over time

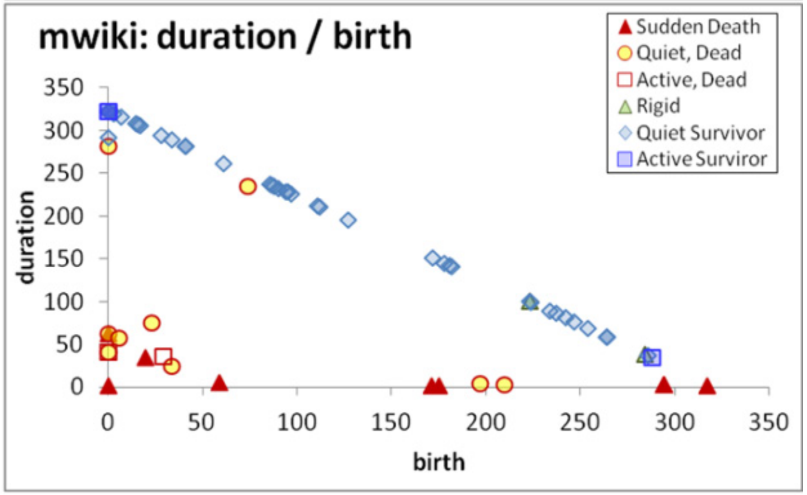
Ensembl: #tables and heartbeat over time

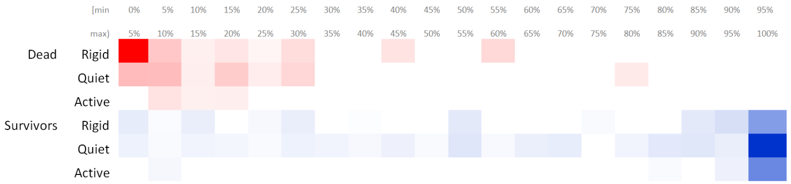
Schema growth over time (red continuous line) along with the heartbeat of changes (spikes) for two of our datasets. Overlayed darker green rectangles highlight the **calmness** versions, and lighter blue rectangles highlight **smooth expansions**. Arrows point at periods of **abrupt expansion** and circles highlight **drops in size**. [IS15]

# What we know so far for table evolution [ER 15, IS 17, CAiSE 17]









# Setup of our study

- **Scope & generalization**:
  - Collected **histories (i.e., sequence of versions) of relational schemata** being part of free open-source software (and not proprietary ones) coming with…
  - … fairly long history
  - … different domains, treatment of foreign keys, growth over time
- **Domains**
  - Science (Atlas, BioSQL)
  - Computational Resource Toolkits (Castor, Egee)
  - CMS's (Slashcode, Zabbix)

- We should be very careful to not overgeneralize findings to proprietary databases!

# Characteristics of used datasets

| Dataset | Versions | Lifetime | Tables @Start | Tables @End | Tables @ Diach. | Table Growth | FKs@ Start | FKs@ End | FKs @ Diach. | FK Growth |
|---|---|---|---|---|---|---|---|---|---|---|
| Atlas | 85 | 2 Y, 7 M | 56 | 73 | 88 | 30% | 61 | 63 | 88 | 0.03% |
| BioSQL | 47 | 6 Y, 7 M | 21 | 28 | 45 | 33% | 17 | 43 | 79 | 153% |
| Egge | 17 | 4Y | 6 | 10 | 12 | 67% | 3 | 4 | 6 | 33% |
| Castor | 194 | 3Y | 62 | 74 | 91 | 20% | 6 | 10 | 13 | 67% |
| SlashCode | 399 | 12 Y, 6 M | 42 | 87 | 126 | 108% | 0 | 0 | 47 | 0% |
| Zabbix | 160 | 10 Y, 10 M | 15 | 48 | 58 | 220% | 10 | 2 | 38 | -80% |

# Toolset

- Some preprocessing was occasionally needed to allow the parsing of schema histories
- Used out homegrown toolset to extract changes
  - **Hecate**, a tool to extract the history of changes for tables

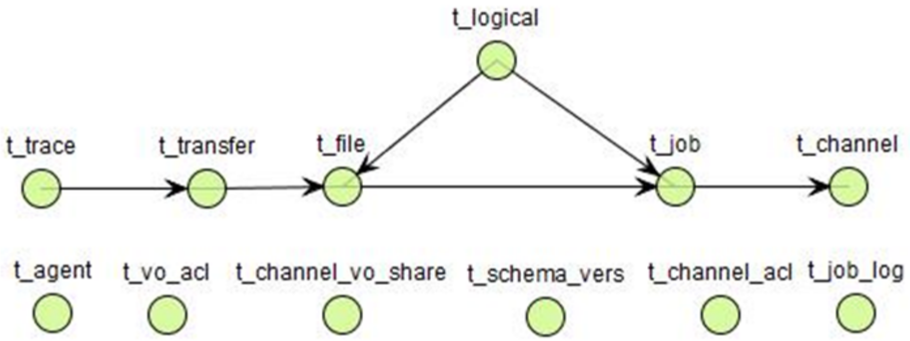
    https://github.com/DAINTINESS-Group/Hecate
  - **Parmenidian Truth**, a tool to extract the history of changes for foreign keys

    https://github.com/DAINTINESS-Group/ParmenidianTruth

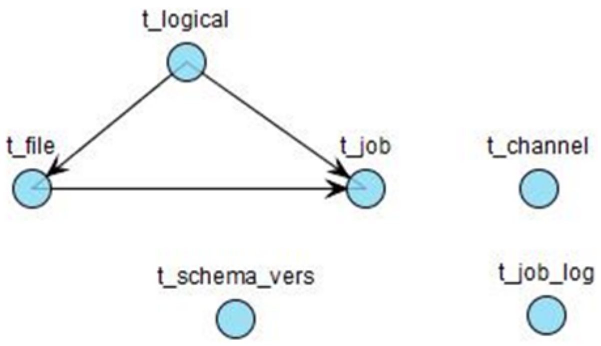
    Parmenidian Truth is also able to visualize the schema history as a PowerPoint/video file
- **All the data** are available at**:**

  https://github.com/DAINTINESS-Group/EvolutionDatasets

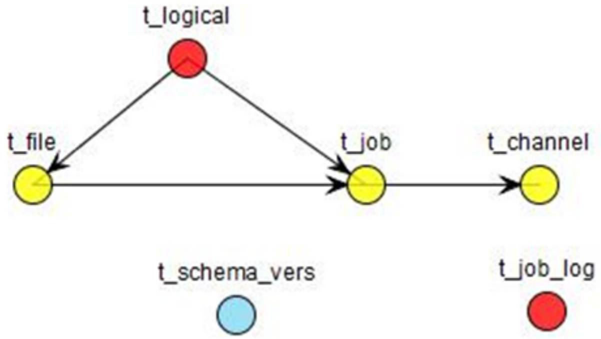
# Using a graph metaphor for evolving schemata with FK’s (bonus: the story of Egee in one slide)



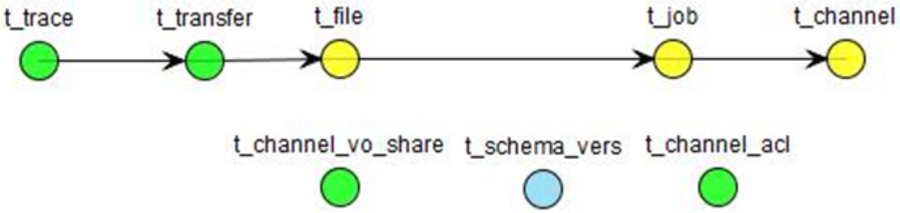
Diachronic Graph of Egee



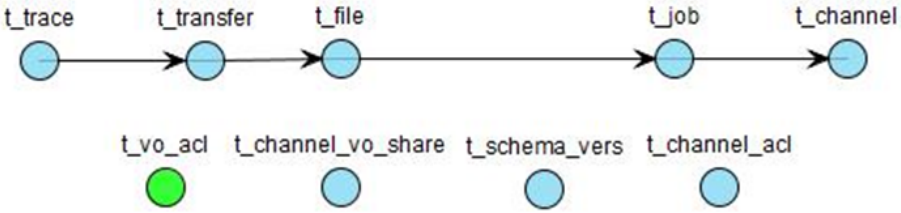
First version: v. 1.0.1



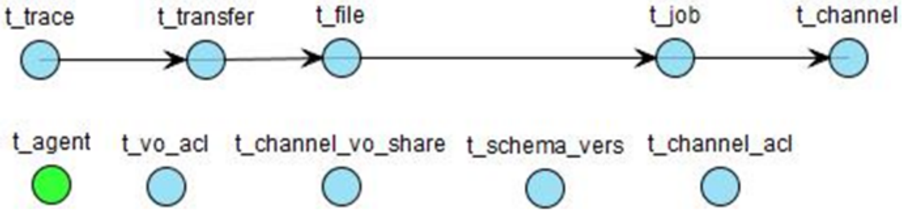
Deletions & Updates: v. 1.0.2



Additions & Updates: v. 1.0.8



Additions: v. 1.0.15



Final version: v. 1.0.17

# What we don’t know yet…

- **How do FK’s evolve?**
  - Do tables and foreign keys evolve in sync?
  - When & How do FK’s germinate & die?

- … as we will see, these questions led to unexpected results and more insights on how developers deal with foreign keys…

- Also studied [not part of the paper]: graph properties of tables and their relationship to evolution

Context and background

Setup of our study

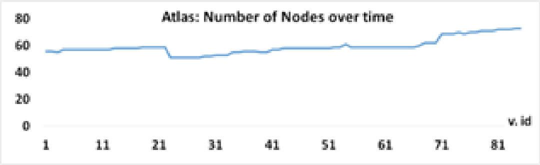
Main findings

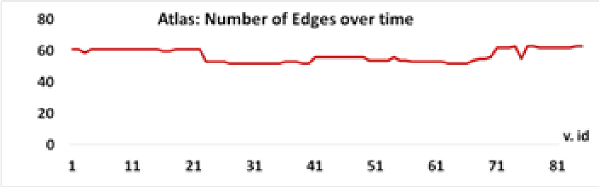
Strange things happening with FK’s

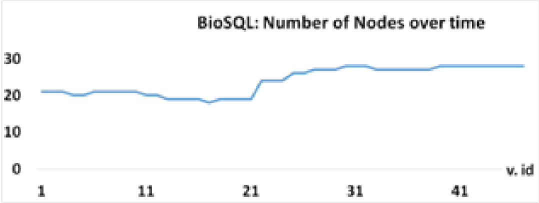
Lessons learned, open issues & why bother

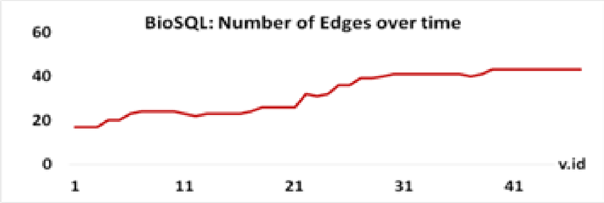
# MAIN FINDINGS

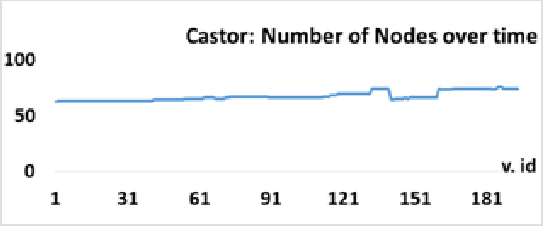
# Evolution of Tables & FK's

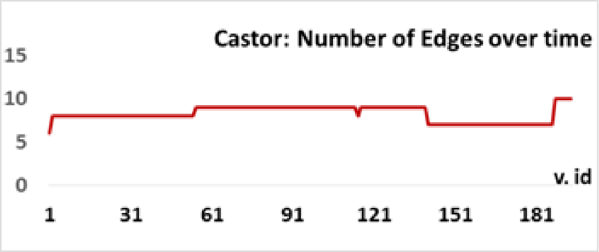


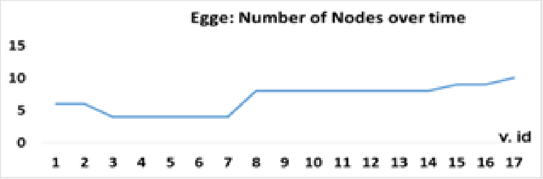


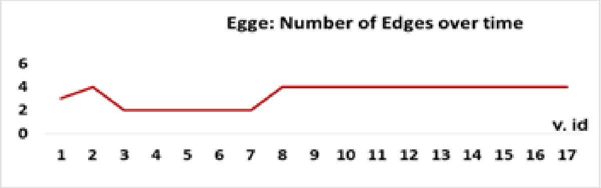


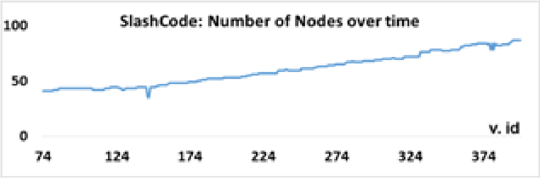


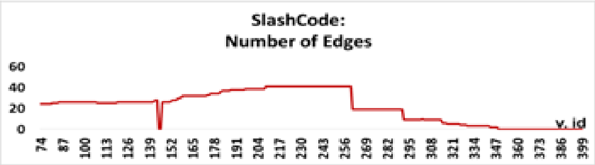


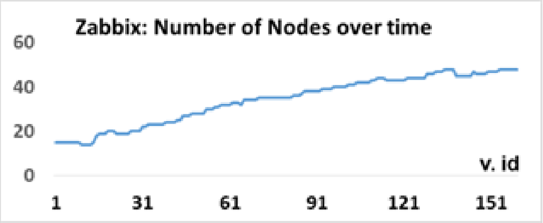


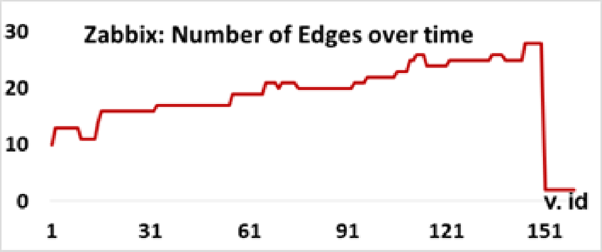






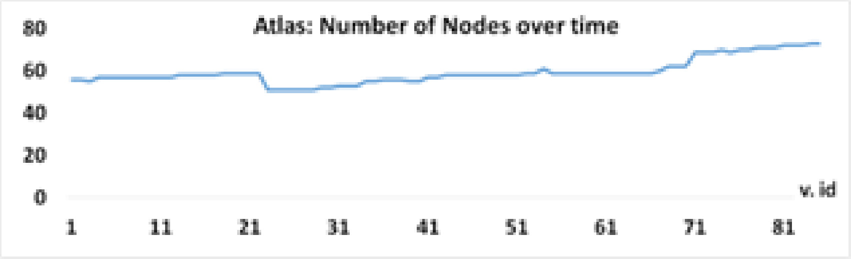


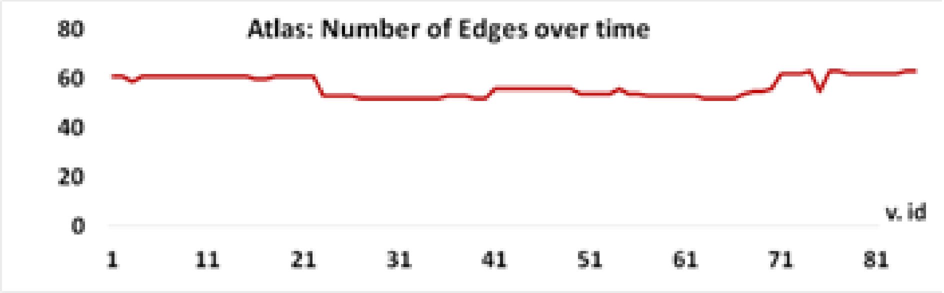


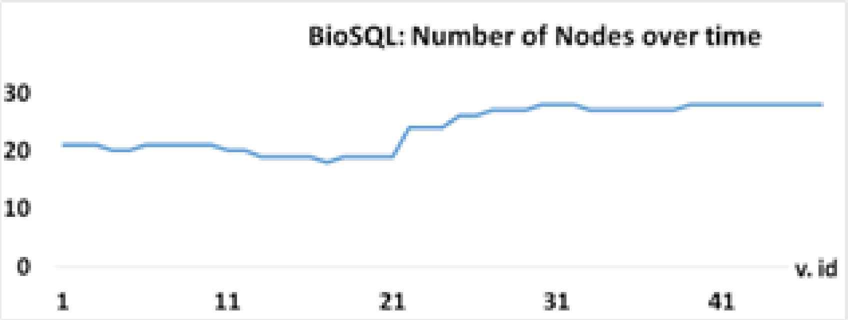


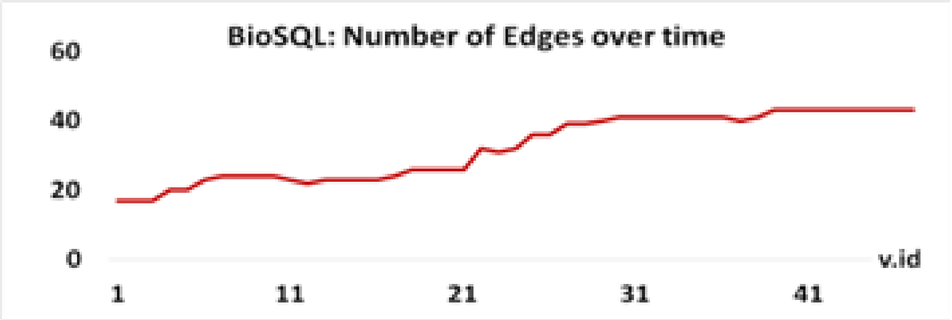
- Tables grow in all cases (known from previous research) with periods of slow growth, calmness, spikes of extension, and occasional cleanups
- Foreign Keys are treated with different mentalities. 3 families:
  - Scientific
  - Comp. Toolkits
  - CMS's

# Evolution of Tables & FK's: Scientific projects



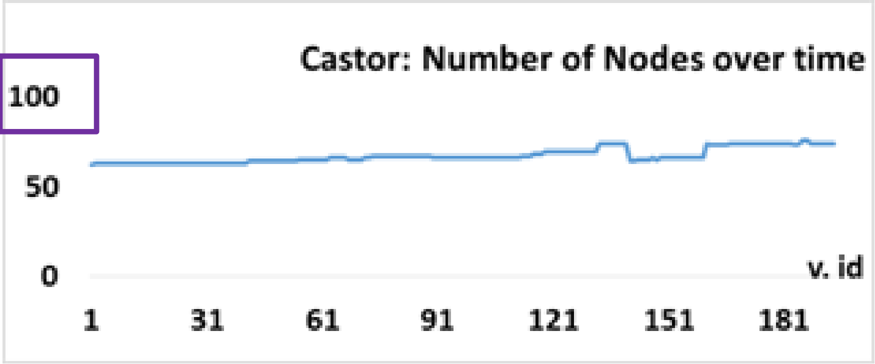


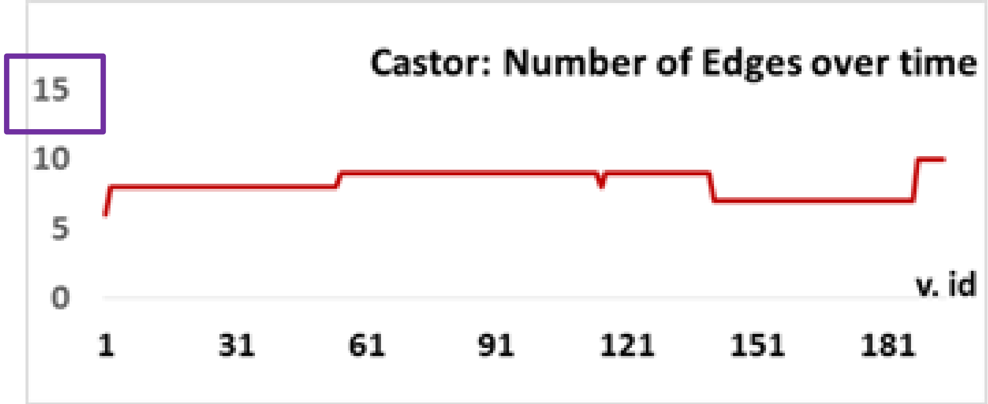


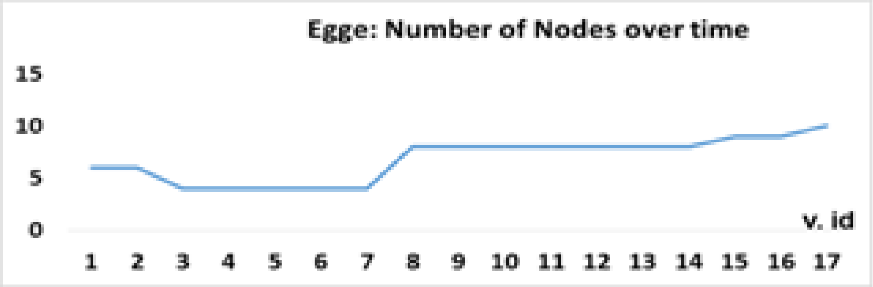


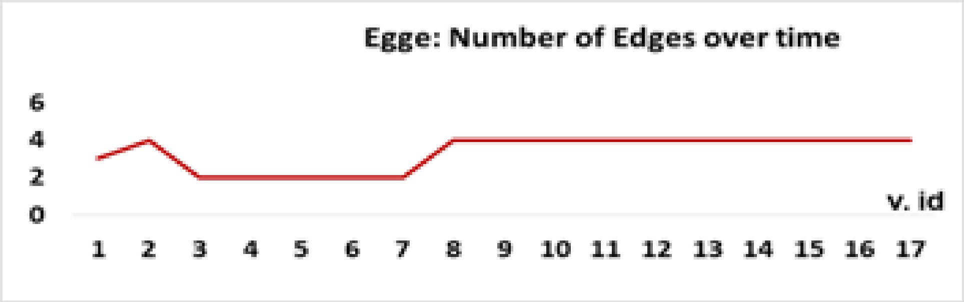
- Tables and FKS grow in synch, in both cases
- Growth comes with expansion periods, shrinkage actions, and periods of calmness in terms of both tables and foreign keys.

# Evolution of Tables & FK’s: Computational Resource Toolkits



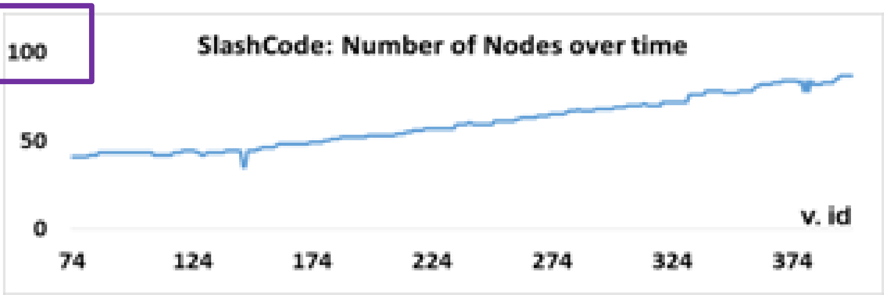


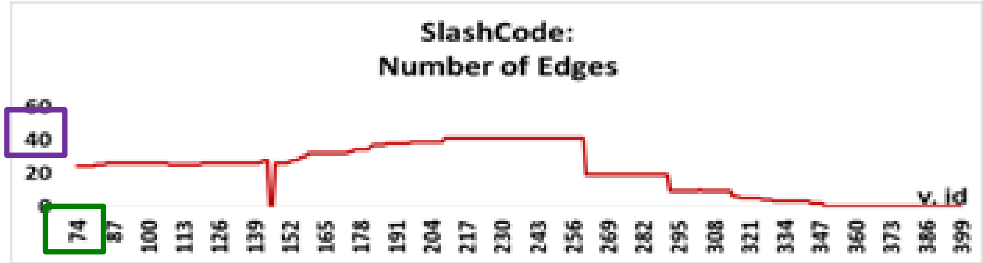


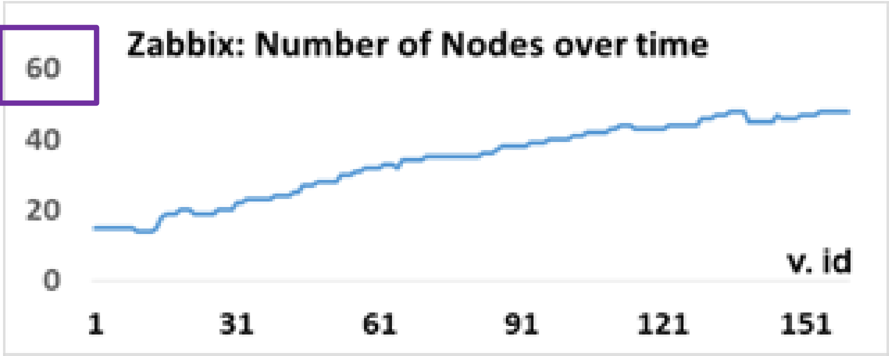


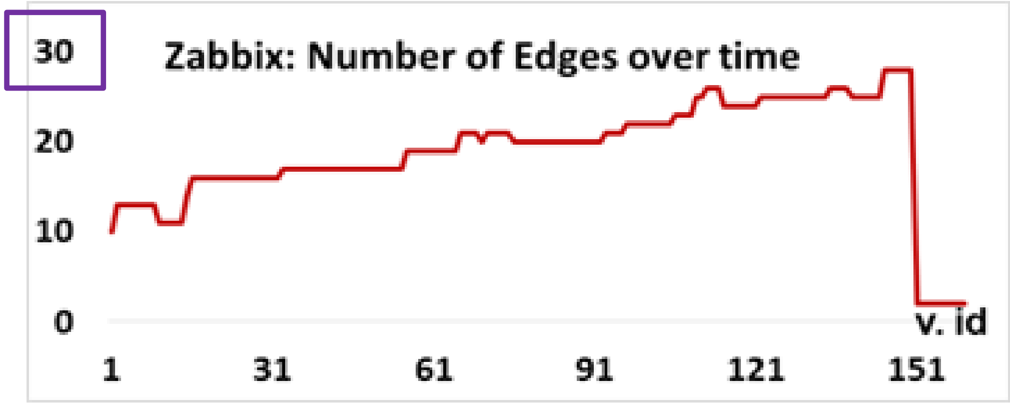
- Tables and FKS grow little and slowly; for Castor, not exactly in sync
- Castor: observe **how scarce FK’s are** (too few tables come with FK’s, see vertical axis)

# Evolution of Tables & FK’s: Content Management Systems (CMS’s)



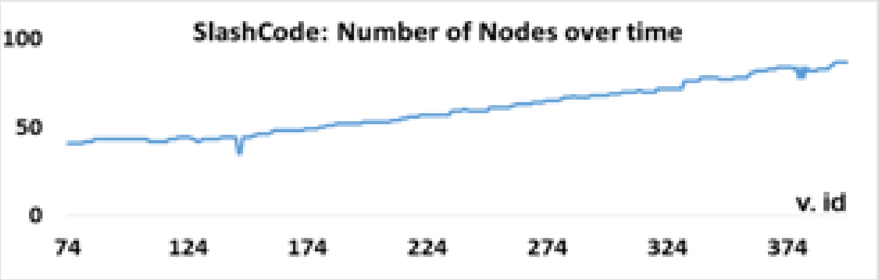


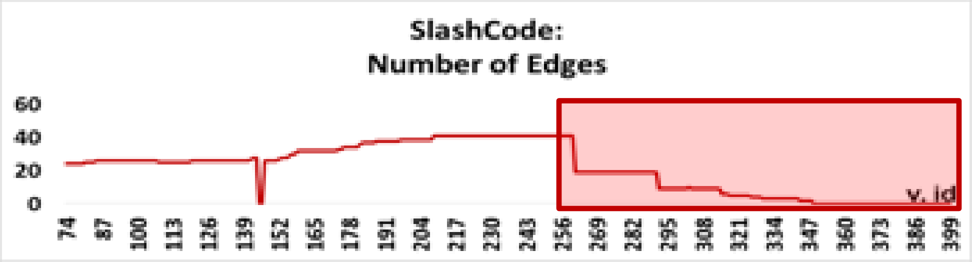


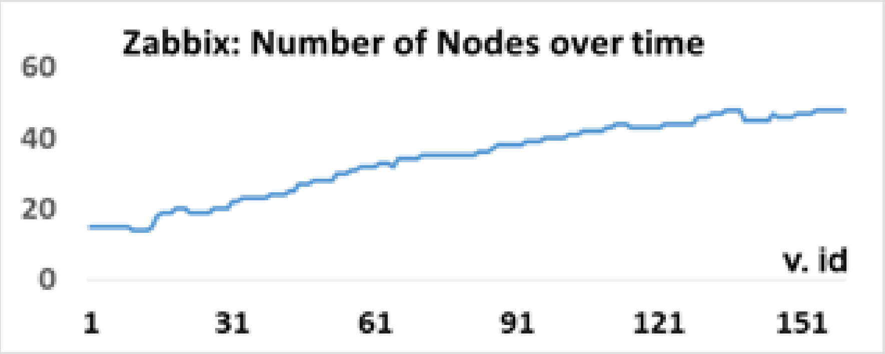


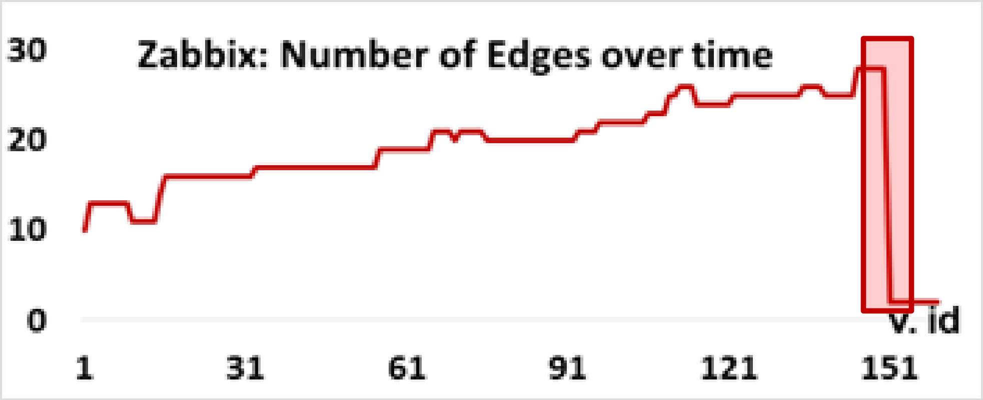
- **FK scarcity**: really big at Slashcode, moderate at Zabbix
- Slashcode started without foreign keys at all; 1st set of FK’s in v. 74. Zabbix seems to show a certain degree of syncronized growth
- **Yet, … both CMS's end up with no FK’s!!** *-> see next*

# What an unpleasant surprise: developers can resort in full removal of foreign keys!









- Slashcode: there is a clear phase of **progressive removal**
- Zabbix: **abrupt removal** of almost the entire set of foreign keys in a single transition (unexpected based on how FK's had been treated till then)
- We dedicate some explanations in the sequel…

# How do FK’s germinate and die?

- We classified FK’s births and deaths in 4 categories
- Births
  - **Born with table**: when either the source or the target table is born along with the foreign key,
  - **Explicit addition**: when a foreign key is added to two existing tables.
- Deletions
  - **Died with table**: when either the source or the target table is removed along with the foreign key,
  - **Explicit deletion**: when neither of the source or target tables gets deleted and only the foreign key is removed.

# Stats on FK Change

| | | | Atlas | Biosql | Egee | Castor | Slashcode | Zabbix |
|---|---|---|---|---|---|---|---|---|
| | Diachronic Graph | TablesDG | 88 | 45 | 12 | 91 | 126 | 58 |
| | | FK'sDG | 88 | 79 | 6 | 13 | 47 | 38 |
| | Start/End | FKs@start | 61 | 17 | 3 | 6 | 0 | 10 |
| | | FKs@end | 65 | 52 | 5 | 10 | 0 | 2 |
| #FKs_added … | … in absolute numbers | Total | 41 | 81 | 4 | 8 | 77 | 28 |
| | | Born w/ table | 37 | 71 | 3 | 2 | 21 | 24 |
| | | Explicit addition | 4 | 10 | 1 | 6 | 56 | 4 |
| | … as pct | (%)Born w/ table | 90% | 88% | 75% | 25% | 27% | 86% |
| | | (%)Explicit addition | 10% | 12% | 25% | 75% | 73% | 14% |
| #FKs_removed … | … in absolute numbers | Total | 37 | 46 | 2 | 4 | 77 | 36 |
| | | Died w/ table | 25 | 42 | 2 | 2 | 16 | 8 |
| | | Explicit deletion | 12 | 4 | 0 | 2 | 61 | 28 |
| | … as pct | (%)Died w/ table | 68% | 91% | 100% | 50% | 21% | 22% |
| | | (%)Explicit deletion | 32% | 9% | 0% | 50% | 79% | 78% |

# Stats on FK Change

| | | | Atlas | Biosql | Egee | Castor | Slashcode | Zabbix |
|---|---|---|---|---|---|---|---|---|
| | Diachronic Graph | TablesDG | 88 | 45 | 12 | | | |
| | | FK'sDG | 88 | 79 | 6 | | | |
| | Start/End | FKs@start | 61 | 17 | 3 | | | |
| | | FKs@end | 65 | 52 | 5 | | | |
| #FKs_added … | … in absolute numbers | Total | 41 | 81 | 4 | | | |
| | | Born w/ table | 37 | 71 | 3 | | | |
| | | Explicit addition | 4 | 10 | 1 | | | |
| | … as pct | (%)Born w/ table | 90% | 88% | 75% | | | |
| | | (%)Explicit addition | 10% | 12% | 25% | | | |
| #FKs_removed … | … in absolute numbers | Total | 37 | 46 | 2 | | | |
| | | Died w/ table | 25 | 42 | 2 | | | |
| | | Explicit deletion | 12 | 4 | 0 | | | |
| | … as pct | (%)Died w/ table | 68% | 91% | 100% | | | |
| | | (%)Explicit deletion | 32% | 9% | 0% | | | |

Atlas, Biosql and Egee (less) deal with **FK's as regular part of the schema**

FK's are, to a large extent …

- Born with tables
- Removed with tables

# Stats on FK Change

| | | | Atlas | Biosql | Egee | Castor | Slashcode | Zabbix |
|---|---|---|---|---|---|---|---|---|
| | Diachronic Graph | TablesDG | | | | 91 | 126 | 58 |
| | | FK'sDG | | | | 13 | 47 | 38 |
| | Start/End | FKs@start | | | | 6 | 0 | 10 |
| | | FKs@end | | | | 10 | 0 | 2 |
| #FKs_added ... | ... in absolute numbers | Total | | | | 8 | 77 | 28 |
| | | Born w/ table | | | | 2 | 21 | 24 |
| | | Explicit addition | | | | 6 | 56 | 4 |
| | ... as pct | (%)Born w/ table | | | | 25% | 27% | 86% |
| | | (%)Explicit addition | | | | 75% | 73% | 14% |
| #FKs_removed ... | ... in absolute numbers | Total | | | | 4 | 77 | 36 |
| | | Died w/ table | | | | 2 | 16 | 8 |
| | | Explicit deletion | | | | 2 | 61 | 28 |
| | ... as pct | (%)Died w/ table | | | | 50% | 21% | 22% |
| | | (%)Explicit deletion | | | | 50% | 79% | 78% |

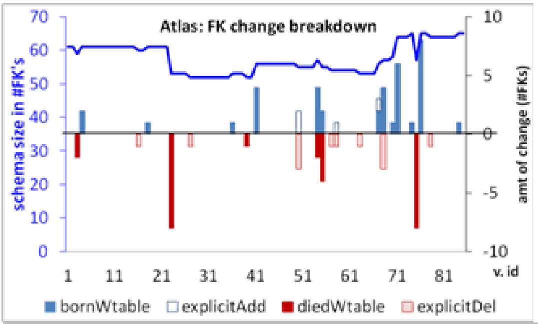
Castor & Slashcode (both with a really small minority of FK's) deal with **FK's as an ad-hoc add on**: FK's are mostly explicitly added/ removed

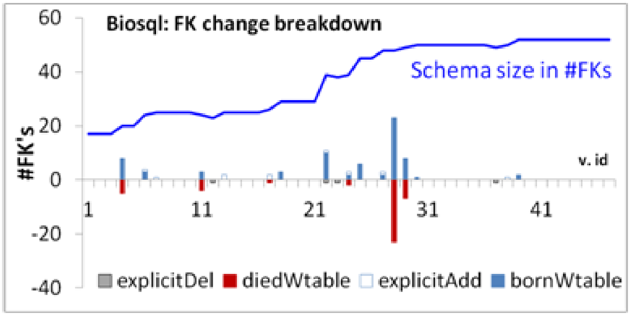
Zabbix has a mixed style: explicit del. and add. w. tables (& a sudden style change)

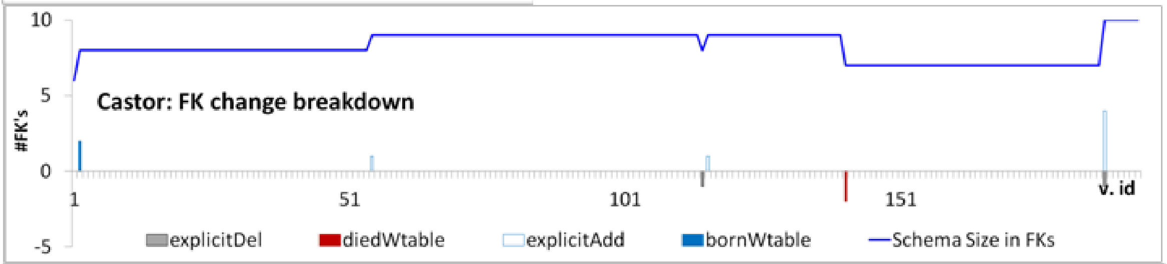
# Families of developer profiles wrt the treatment of Foreign Keys

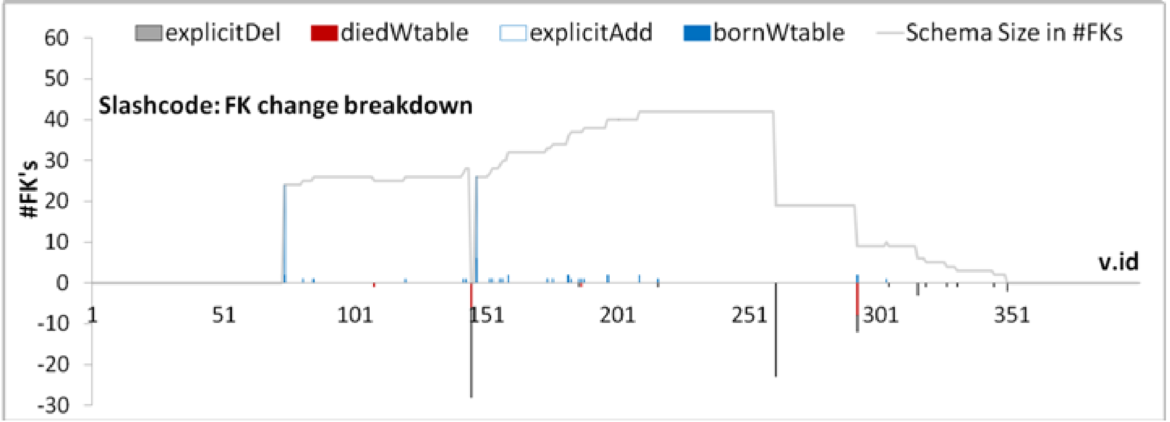
- **Integral part of schema**: fairly large pct of tables involved in FKs, grow in sync with tables, germinate and die with them
- **Disposable Add-on**: small pct of tables involved in FK's, explicit additions and deletions, easy to remove them (in some cases, entirely!)
- **Mixed**: can be with a change of style

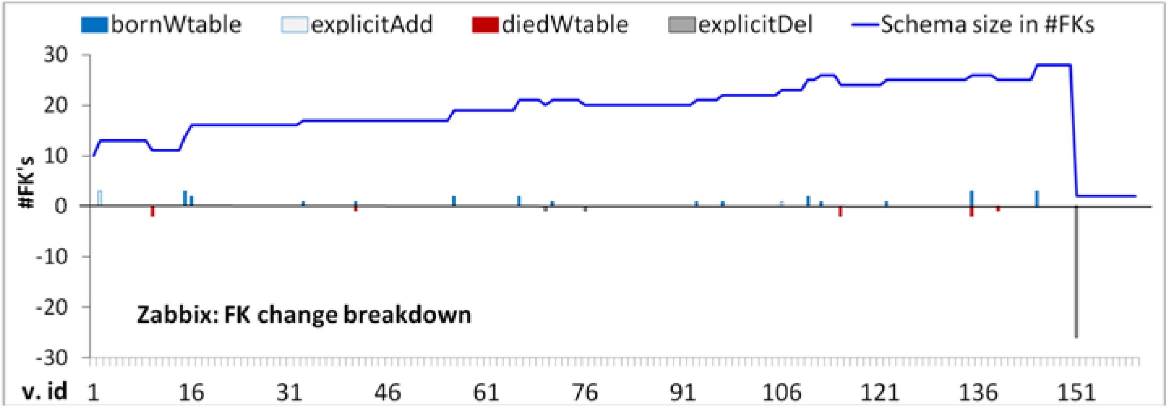
# Heartbeat of change











**Birth & deaths are proportionally spread in time** -- except Atlas.

The **volume of change is typically low**: most changes ~ 1 FK.

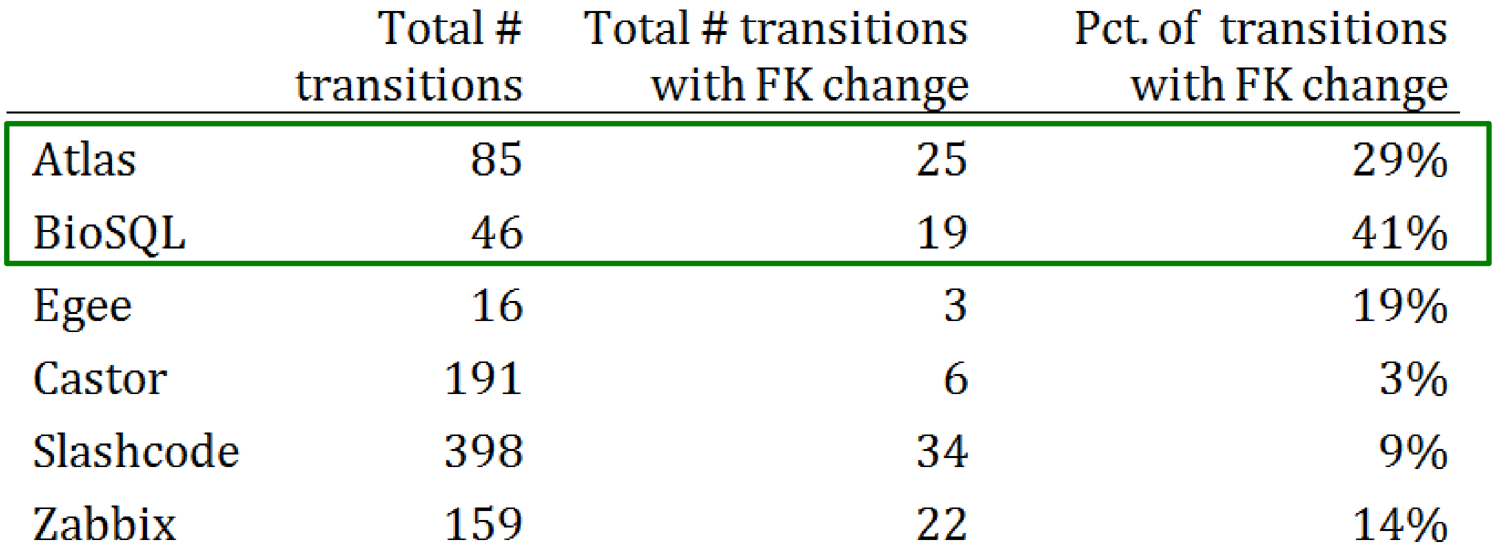
Exceptions:

(a) explicit mass add & del,

(b) do-undo actions (Atlas, Slashcode and Castor), and,

(c) restructuring due to table renamings (4 in Biosql, 2 in Zabbix).

# Percentage of transitions with FK change



| | Total # transitions | Total # transitions with FK change | Pct. of transitions with FK change |
|---|---|---|---|
| Atlas | 85 | 25 | 29% |
| BioSQL | 46 | 19 | 41% |
| Egee | 16 | 3 | 19% |
| Castor | 191 | 6 | 3% |
| Slashcode | 398 | 34 | 9% |
| Zabbix | 159 | 22 | 14% |

Common theme in all the data sets: the **consistent scarcity of FK changes**

- Scientific data sets: short active period + treatment of FK's as an integral part of the schema (births and deaths of tables and FK's in sync) => high pct of transitions with FK change
- The rest: FK b&d are rare and explicit (w/o mass removals, would be even less)

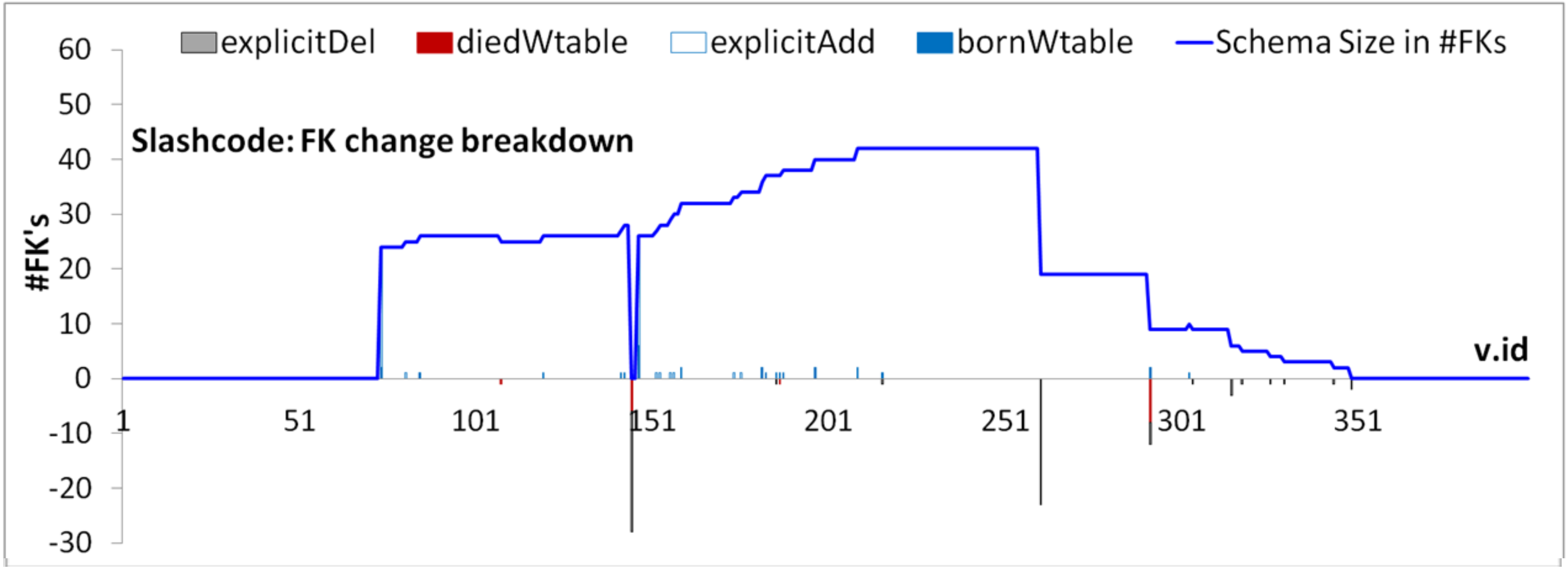
# Characteristics of the heartbeat of schemata wrt Foreign Keys

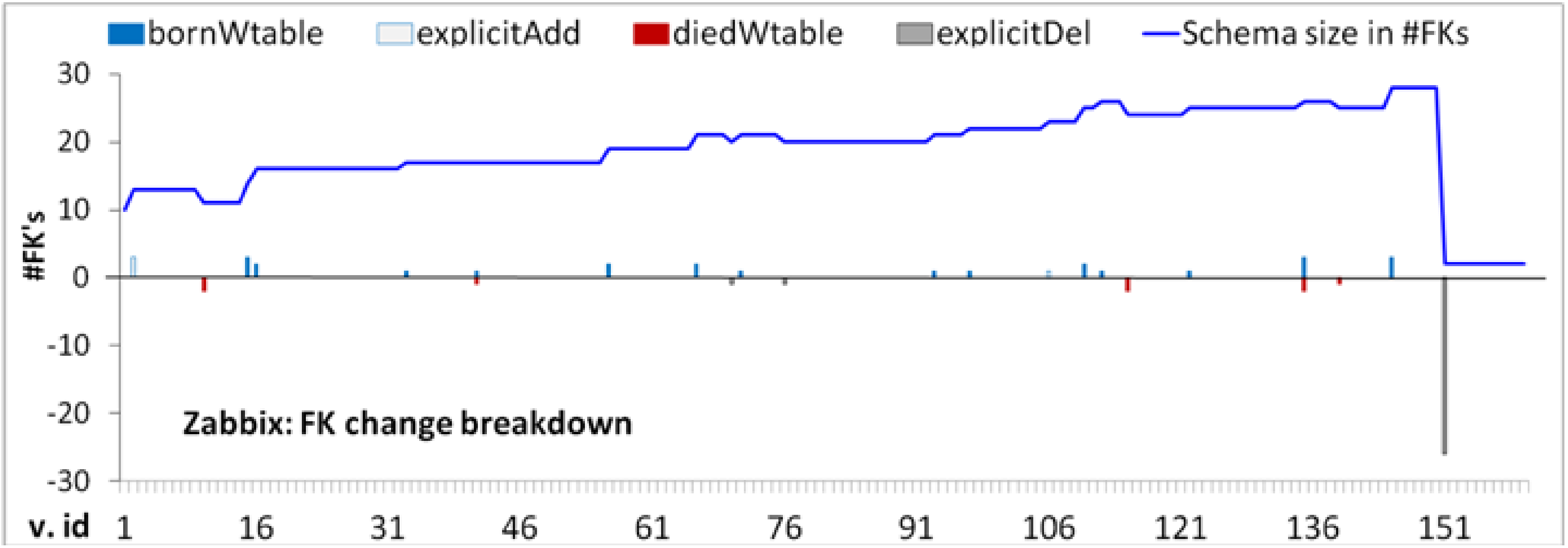
- **Scarcity of FK change**: expectedly very few transitions come with FK change, except for idiosyncratic cases
- **Low volume**: typically 1 FK change at a time, except for mass add/del
- **Birth & deaths are proportionally spread in time**
- Occasional **do-undo** and restructuring due to **table renames**

Context and background
Setup of our study
Main findings
Strange things happening with FK’s
Lessons learned, open issues & why bother

# THE MYSTERIOUS CASE OF THE DISAPPEARING FOREIGN KEYS

# Heartbeat of change: CMS's

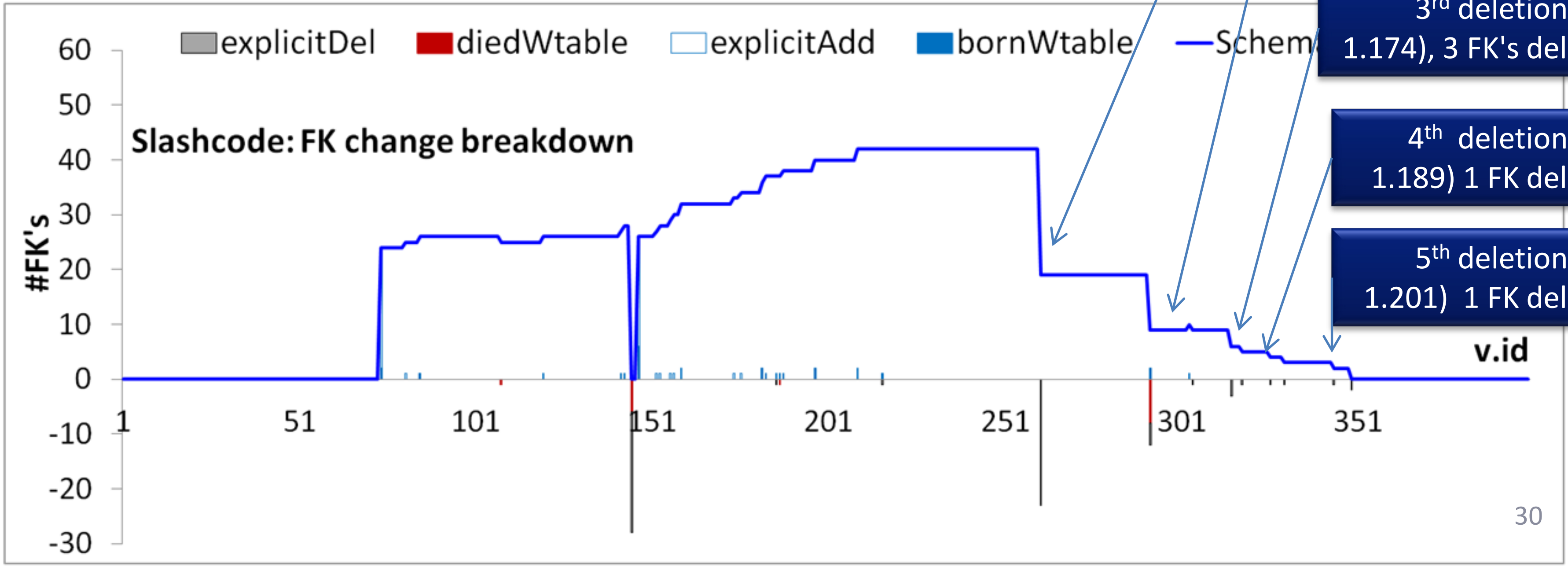




# Slashcode: the disappearing FK’s

- At the end of its studied history, and via a progressive removal period, the schema is left with zero foreign keys.
- Interestingly enough, the schema also contained zero foreign keys at its start.
- Quite importantly, Slashcode's behavior holds both foreign key additions and deletions mostly happening explicitly (i.e., without the addition or removal of the involved tables).
- In other words, it appears that **foreign keys are treated as a disposable add-on that was removed when problems occurred.**

# Slashcode: the disappearing FK’s



1st massive foreign key removal (rev 1.120), 22 FK’s deleted.

2nd massive deletion (rev 1.151), 10 FK's deleted

3rd deletion (rev 1.174), 3 FK's deleted

4th deletion (rev 1.189) 1 FK deleted

5th deletion (rev 1.201) 1 FK deleted

"Commented-out foreign keys are ones which currently cannot be used because they refer to a primary key which is NOT NULL AUTO INCREMENT and the child's key either has a default value which would be invalid for an auto increment field, typically NOT NULL DEFAULT '0'.
Or, in some cases, the primary key is e.g. VARCHAR(20) NOT NULL and the child's key will be VARCHAR(20). The possibility of NULLs negates the ability to add a foreign key. <= That's my current theory, but it doesn't explain why discussions.topic SMALLINT UNSIGNED NOT NULL DEFAULT '0' is able to be foreign-keyed to topics.tid SMALLINT UNSIGNED NOT NULL AUTO INCREMENT"

1st massive foreign key removal (rev 1.120), 22 FK's deleted.

"Stories is now InnoDB and these other tables are still MyISAM, so no foreign keys between them."

2nd massive deletion (rev 1.151), 10 FK's deleted

"This doesn't work, makes createStory die. These don't work, should check why..."

3rd deletion (rev 1.174), 3 FK's deleted

"This doesn't work, since in the install pollquestions is populated before users, alphabetically"

4th deletion (rev 1.189) 1 FK deleted

"This doesn't work, since discussion may be 0."

5th deletion (rev 1.201) 1 FK deleted

# Slashcode: what did the comments say?

- **The main problem seems to be the difficulty of developers with the tuning and handling of both foreign and primary keys.**
- Sometimes difficulties are hard -- e.g., different storage engines, typically due to performance reasons
- Some difficulties are complicated due to technicalities like autonumbering
- Sometimes fixes could be found with some effort (e.g., changing the order of table population, or using numeric data types for primary keys, or inserting some “goalkeeper” values at FK target table)

# Slashcode: what do we make out of this case?

- **The main problem seems to be the difficulty of developers with the tuning and handling of both foreign and primary keys.**
- Practically, it appears that the easiest way out of this kind of problems is to comment out the respective foreign key.
- So, **removals of foreign keys went on as a regular practice, instead of attempting to fix the problems.**
- This simply states that **the essence of the contribution of foreign keys in the consistency of the schema does not seem to outweigh the need to quickly get things done.**

# Scarcity of Foreign keys

- A 2013 collection of schema histories, lists **21 data sets**, -- some have more than one target DBMS variants.

```
$ cd RESEARCH/Github/EvolutionDatasets
$ ls -d * */*
CERN          CMS's/Coppermine  CMS's/XOOPS      Med
CERN/Atlas    CMS's/DekiWiki    CMS's/Zabbix     Med/Ensembl
CERN/CASTOR   CMS's/Joomla 1.5  CMS's/e107       Med/biosql
CERN/DQ2      CMS's/NucleusCMS  CMS's/opencart   README.md
CERN/DRAC     CMS's/SlashCode   CMS's/phpBB
CERN/EGEE     CMS's/TikiWiki    CMS's/phpwiki
CMS's         CMS's/Typo3       CMS's/wikimedia
```

- **How many data sets contain foreign keys?**
- Try this (also backed by manual sampling):

```
grep -rl "FOREIGN" . >> ALL-FKs-by-grep.ascii
awk '{split($0,a,"/"); print a[2],a[3]}' ALL-FKs-by-grep.ascii  |
uniq
```

# Scarcity of Foreign keys

- **How many data sets, out of the 21, contain foreign keys?**

CERN Atlas
CERN CASTOR
CERN EGEE
CMS's SlashC
CMS's Zabbix
Med biosql

CERN DQ2
CERN DIRAC
Med Ensembl

The **6** data sets reported here

+

**DQ2** (only in the mySQL, not in the Oracle version): FK’s in 19 versions out of the 55.
Starts with 2 FK's and ends with 1.

**DIRAC** (not in the production folder, only at python+mysql).
9 tables at first version, 15 tables at last version
Starts with 10 FK's, ends with 8

**Ensembl**: not able to link FK DDL files to table evolution, yet

- **9 out of the 21 data sets do** (including 3 that are really small for harnessing valuable results, spec., Egee, DQ2, DIRAC)

**you're not welcome here...**

**#sorrynotsorry**

http://www.boldomatic.com/view/post/G_xPI

Context and background
Setup of our study
Main findings
Strange things happening with FK’s
Lessons learned, open issues & why bother

# LESSONS LEARNED, THINGS TO DO & WHY ALL THIS MATTERS

# Main findings

- **Schemata grow in terms of tables**, as time passes
- Cases, mainly in projects of scientific nature, where **FK's are treated as an integral part of the system**, and they are born and evicted along with table birth and eviction.
- Cases where **FK's are treated as a disposable add-on**: only a small subset of the tables involved in FK's; birth and eviction of FK's rarely performed in synch with their tables.
- **The heartbeat of FK change is mostly rare and small in volume**, also with do-undo pairs of commits and occasionally massive removals).
- Within all the **CMS's** we collected, **FK's are too scarce**. For the two CMSs that we studied, **both ended-up with their complete remova**l, due to difficulty of managing technical issues related to FK's.

# Open research issues

- More studies, by other groups, if we are to establish solid patterns and (who knows?) laws
- More in-depth studies on the reasons of the observed phenomena
- Mining patterns of graph evolution

# Threats to validity

- The **scope**, **external validity & generalization** of our study is restricted to **databases that are part of FOSS projects** (and not closed ones) **and also pay the price for data consistency via foreign keys**.
- We have  data sets from **different domains** (occasionally with domain-dependent) with **adequately long stories and schema sizes**. We make clear if patterns are omni-present or strictly characteristic to a domain can indeed be generalized.
- **Measurement validity**: we have tested our tools with black box testing & fixed problems.
- As this is the first -to our knowledge- study of its kind, it is strictly of exploratory nature & **more studies are needed!.**

# Why does this matter?

- We need to understand how schemata evolve over time and do it with solid evidence, because, …
  - We are **scientifically curious** on how our discipline's artifact evolve
  - We will be able to **design databases with a view to their evolution** and **minimize the impact of evolution to the surrounding applications**
  - We can plan to **identify and avoid "design anti-patterns"** leading to cumulative complexity for both the database and the surrounding applications,
  - We can **plan administration and maintenance tasks and resources**, instead of just responding to emergencies.

# Why does this matter?

- … Yet , the study also reveals **unexpected results**: Although it is important not to over-generalize our findings outside the area of Free, Open Source Software, **we have now significant evidence that, unless specifically curated, foreign keys in a FOSS database can potentially be unwelcome (and thus, rare) or even completely removed by the developers**.

***This is a clear warning that we, as a community, need to do better (a) in terms of making systems easier at handling foreign keys and their implications, especially at the deep technical details, as well as, (b) in terms of better educating developers on the benefits and necessities behind the usage of foreign keys in their databases.***

# Moltes gràcies! Muchas gracias! Thank you!

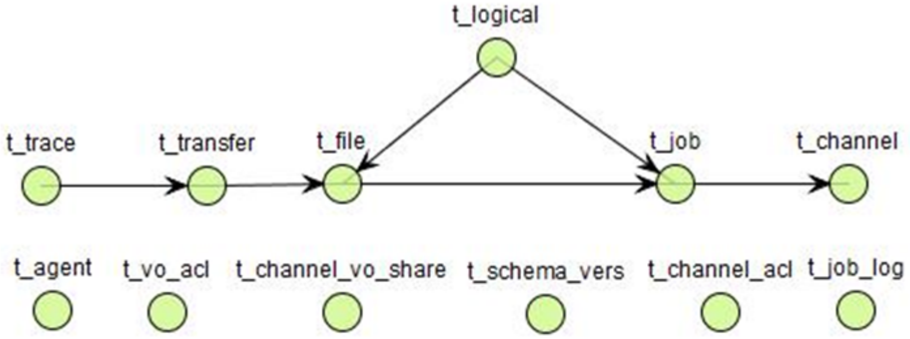
- Foreign Key Evolution comes with different treatments:
  - Sometimes, **FK's are treated as an integral part of the system**, and they are born and evicted along with table birth and eviction.
  - Other times, **FK's are treated as a disposable add-on**: only a small subset of the tables involved in FK's; birth and eviction of FK's rarely performed in synch with their tables.
- Within all the CMS' we collected, **FK's are too scarce** & we even witnessed **complete removal of FK's** from the schema --> we need to react as a community
- Treating the **evolving schema as an evolving graph** comes with particular potential for deeper study.

**To probe further (**code, data, details, presentations, …**)**

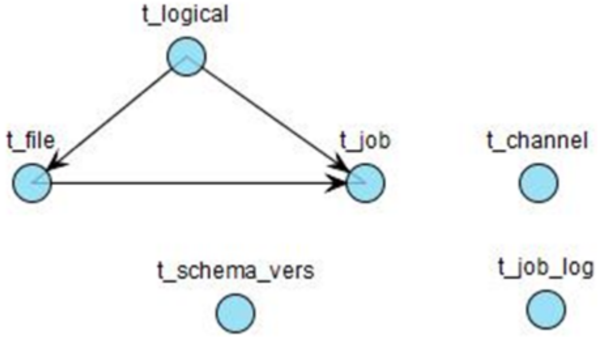
**http://www.cs.uoi.gr/~pvassil/projects/schemaBiographies**

# AUXILIARY MATERIAL

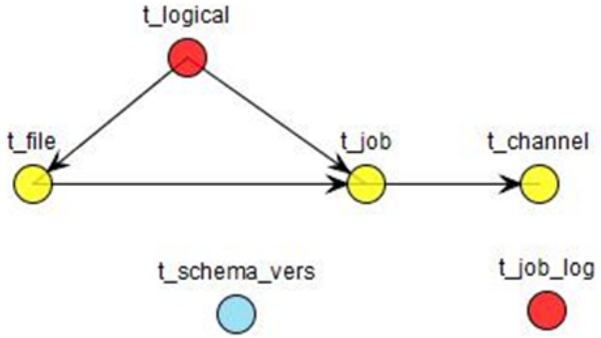
# The story of Egee in one slide



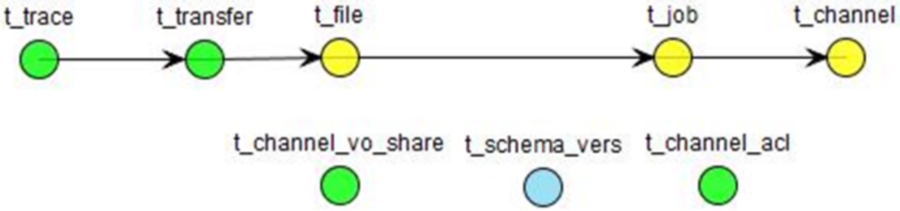
Diachronic Graph of Egee



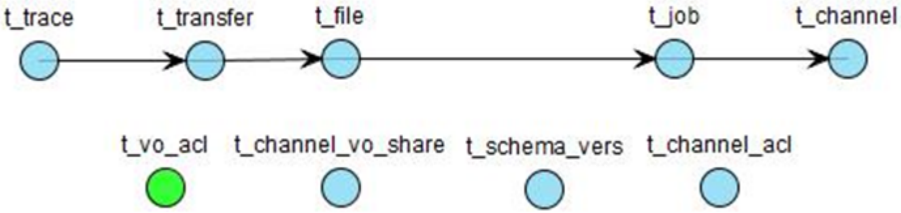
First version: v. 1.0.1



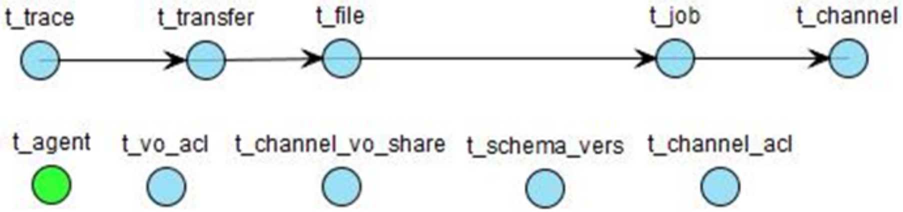
Deletions & Updates: v. 1.0.2



Additions & Updates: v. 1.0.8

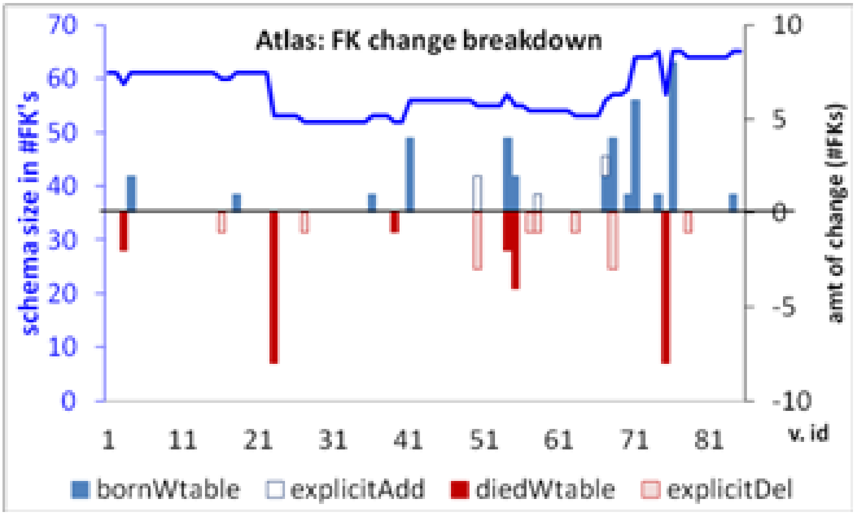


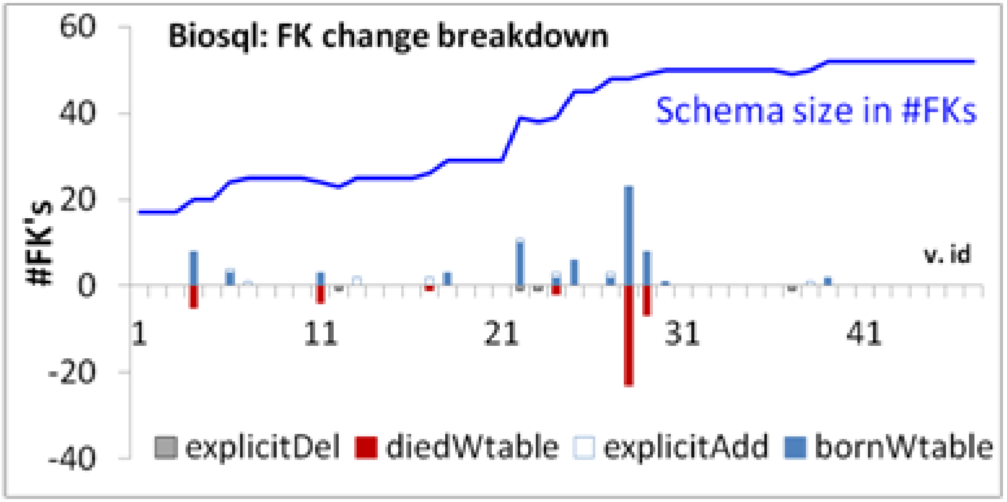
Additions: v. 1.0.15



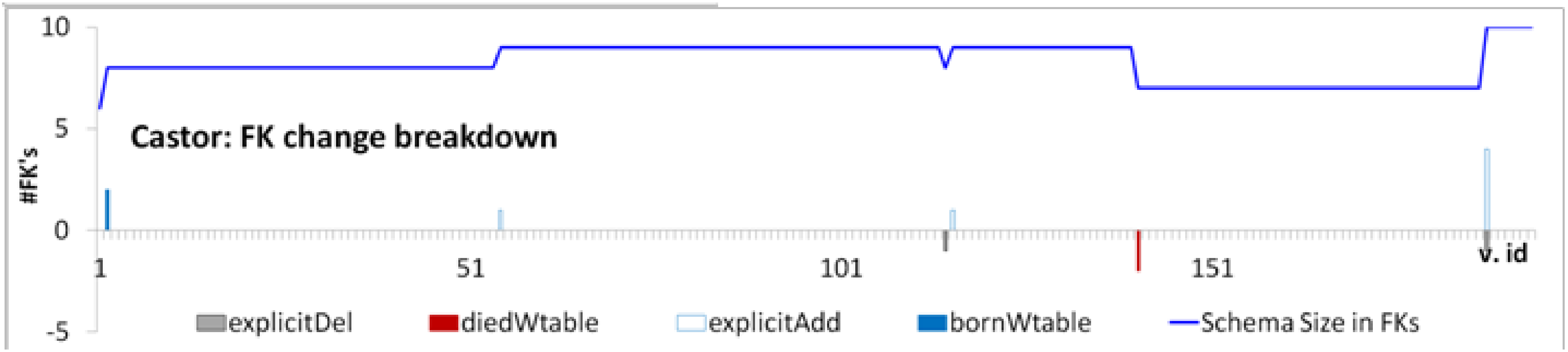
Final version: v. 1.0.17

# Heartbeat of change: Scientific projects

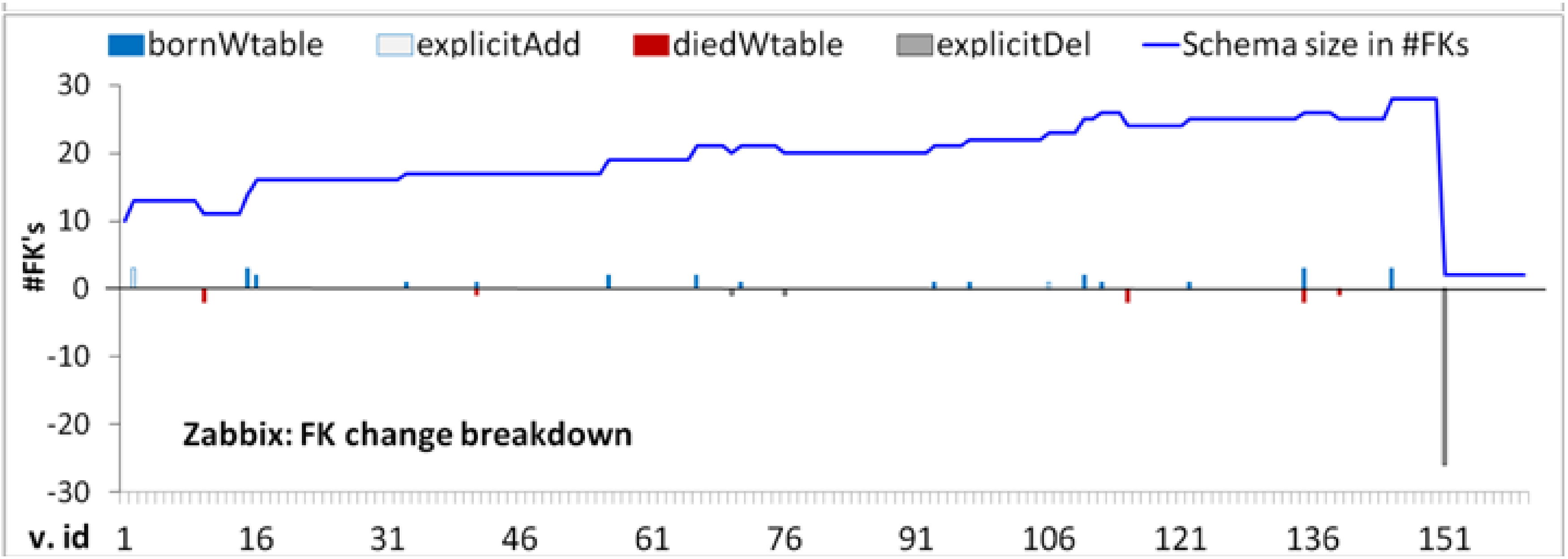




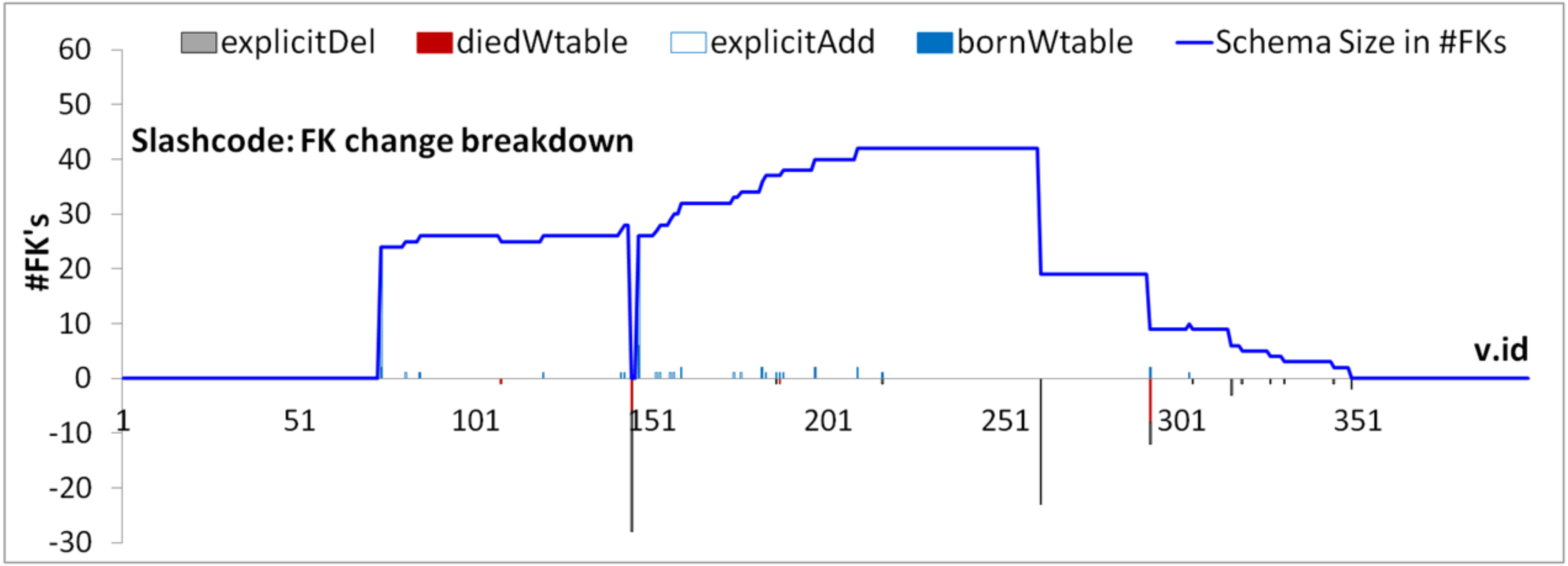
# Heartbeat of change: Computational Resource Toolkits



# Heartbeat of change: Zabbix CMS



# Heartbeat of change: Slashcode CMS



# Einstein on curiosity

From: The ultimate quotable Einstein. Collected and edited by Alice Calaprice, Princeton Univ. Press

The important thing is not to stop questioning. Curiosity has its own reason for existing. One cannot help but be in awe when he contemplates the mysteries of eternity, of life, of the marvelous structure of reality. It is enough if one tries merely to comprehend a little of this mystery every day.

Memoirs of William Miller, editor, quoted in Life magazine, May 2, 1955

The main source of all technological achievements is the divine curiosity and playful drive of the tinkering and thoughtful researcher, as much as it is the creative imagination of the inventor

Speech on the occasion of the opening of the 7th German Radio and Audio Show in Berlin on August 22 in 1930