A study on the effect of a table's involvement in foreign keys to its schema evolution

Konstantinos Dimolikas, Apostolos V. Zarras, Panos Vassiliadis

Department of Computer Science and Engineering
University of Ioannina, Hellas

39th International Conference on Conceptual Modeling
(ER 2020), 3-6 Nov. 2020

http://www.cs.uoi.gr/~pvassil/publications/2020_ER_FK/

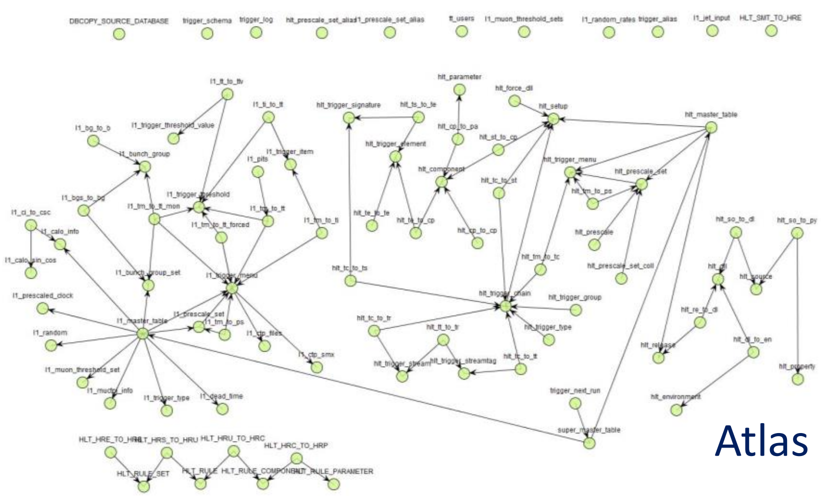
http://www.cs.uoi.gr/~pvassil/projects/schemaBiographies/

Body of Knowledge on Schema Evolution

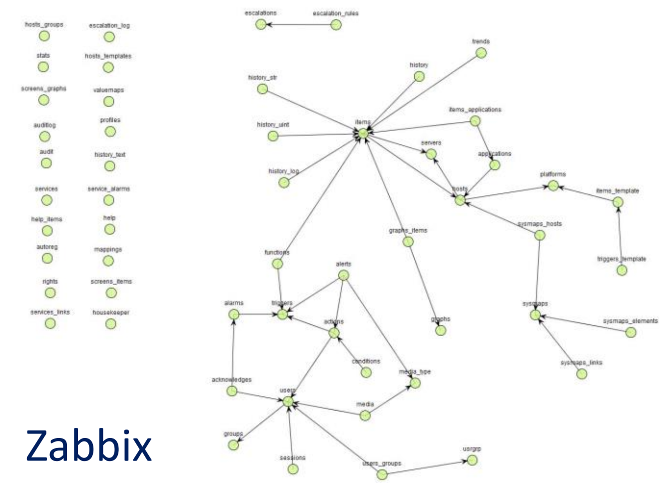
- The literature on Schema Evolution for the relational realm is limited, and mainly focused on the evolution of the entire schema in FoSS systems: schemata grow slowly over time, and, in fact with decreasing rate and alterations of change periods (mostly table insertions and updates) with long periods of calmness
- For Foreign Keys and Evolution: we presented the first paper ever, in ER'17, showing that FK's are not always welcome (frequently absent, even removed); there is also a relationship of higher activity for tables involved with many FK's

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

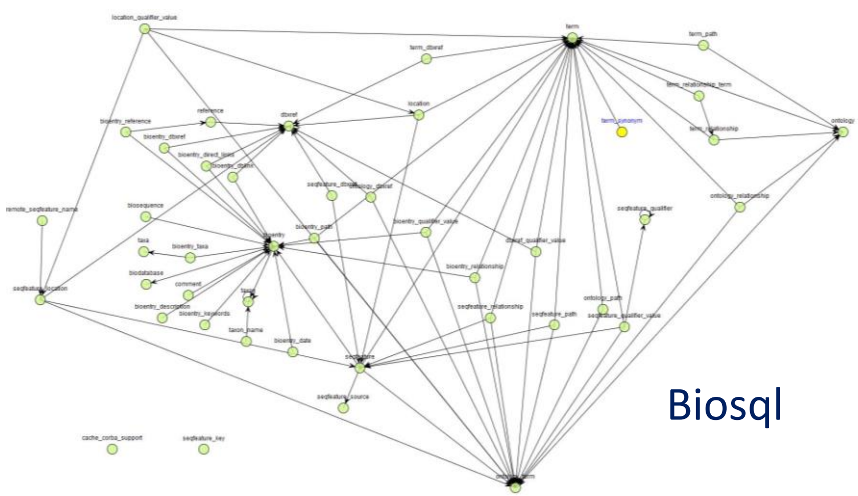
How is the structure of the foreign keys to which a table is related affecting its behavior during schema evolution?



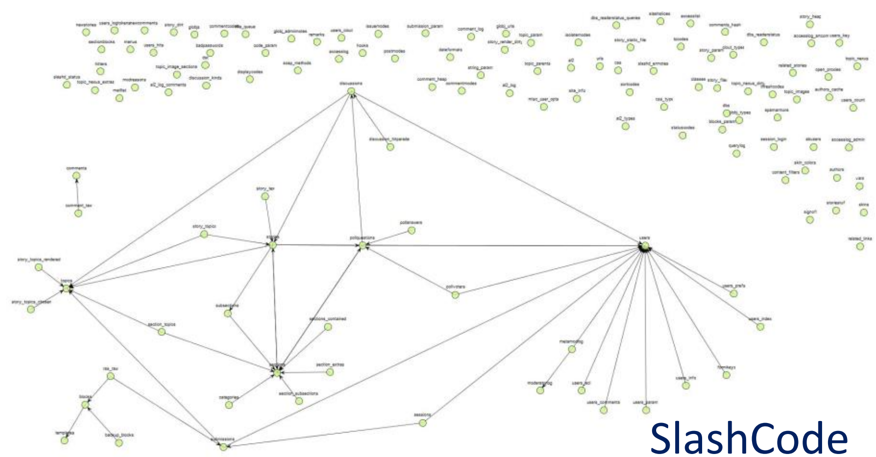
Atlas



Zabbix



Biosql



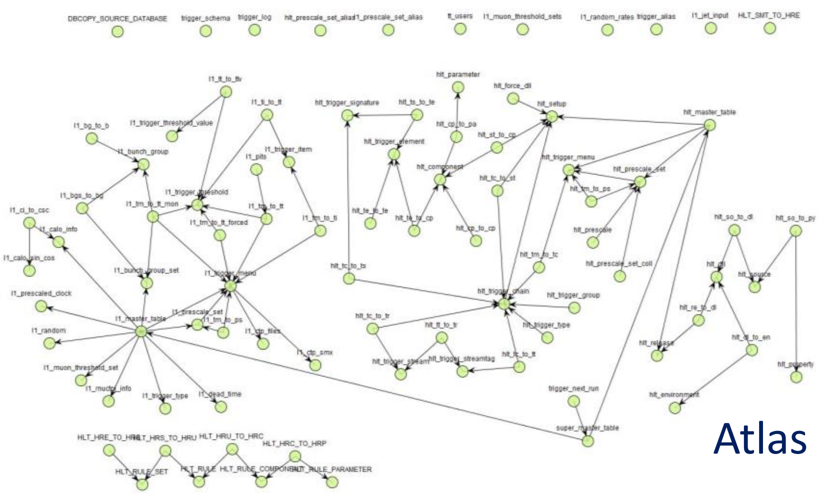
SlashCode

Method overview

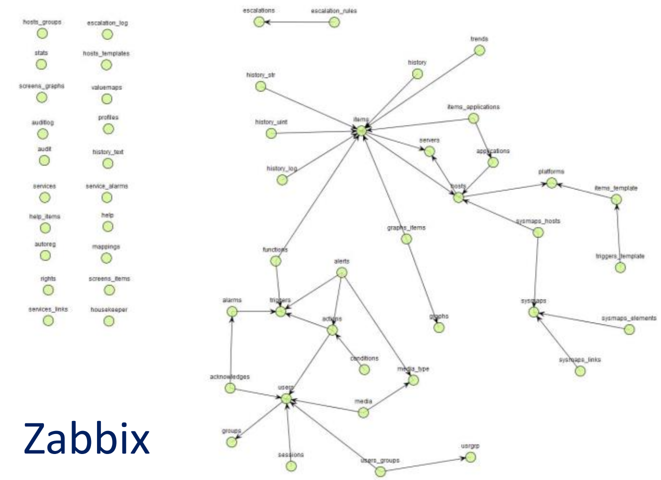
- We study the histories of 6 relational schemata of significant durations and variable characteristics
- We extract
  - births and deaths of the tables,
  - intra-table updates (attribute additions, deletions, data type and primary key updates)
  - foreign keys and their changes
- We model tables and foreign keys as nodes and edges on a graph
- We relate the position of the tables in the graph to evolutionary characteristics

Results

- We introduce a **taxonomy of topological patterns** with respect to how a table is positioned on the schema graph, with a direct relationship to how tables evolve
- Our taxonomy practically introduces a spectrum of topological complexity & we show that **evolutionary behavior is correlated with a hierarchy of topological complexity**:
  - Topologically complex tables appear to be fewer, active and born only early
  - Tables with a simpler topology are more in numbers, less active and with higher chances to be born later in the life of the db

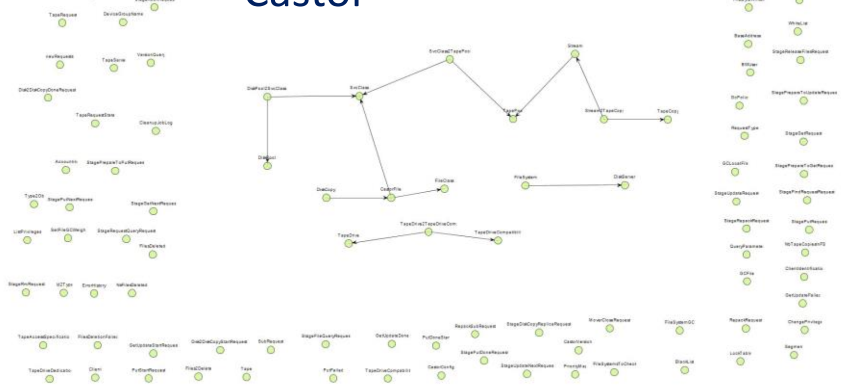


Atlas



Zabbix

Castor



Study Setup

Context and background

Study Setup

Topological Patterns

Topology and Evolution

Conclusions

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)
- **Changes extracted**
  - births and deaths of the tables,
  - intra-table updates (attribute additions, deletions, data type and primary key updates)
  - foreign keys and their changes
- We should be very careful to not overgeneralize findings to proprietary databases!

Datasets

- Cover from 17 to 399 schema versions
- Growth in number of tables from 19% to 220%
- Growth in number of foreign keys with 2 exceptions

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

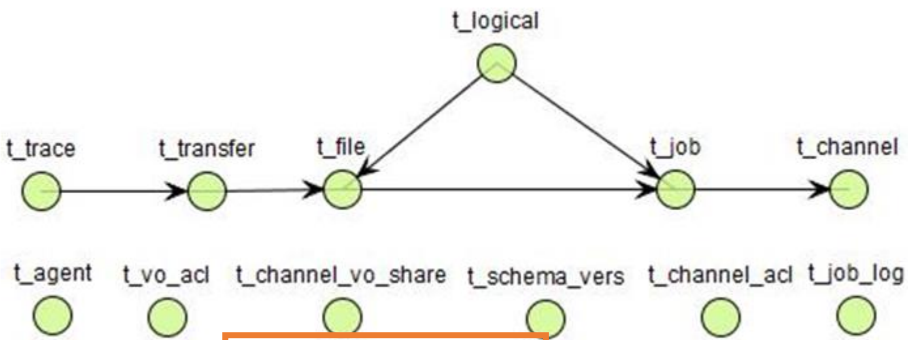
    https://github.com/DAINTINESS-Group/Hecate
  - **Parmenidian Truth**, a tool to extract the history of changes for foreign keys

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

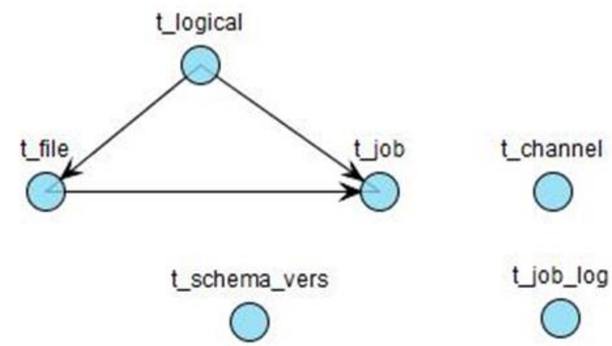
    Parmenidian Truth is also able to visualize the schema history as a PowerPoint/video file
- **All the data** are available at**:**

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

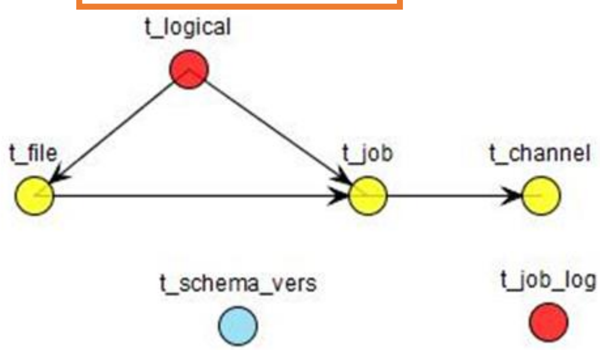
Graph modeling for evolving schemata with FK's (bonus: the story of Egee in one slide)



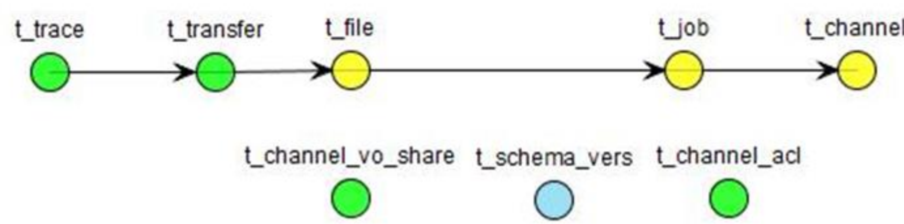
Diachronic Graph of Egee



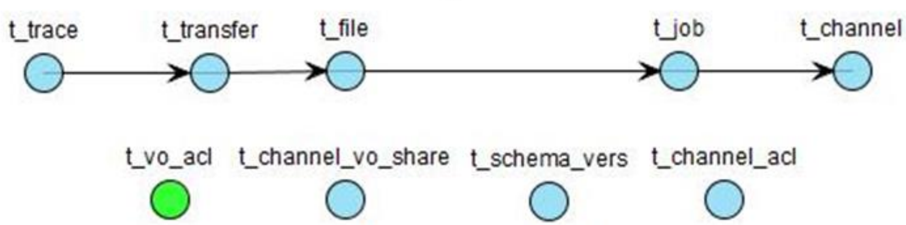
First version: v. 1.0.1



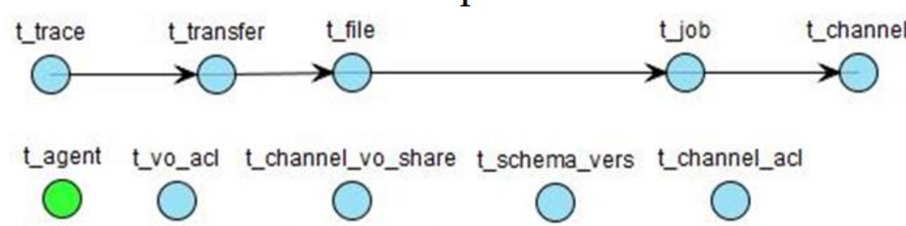
Deletions & Updates: v. 1.0.2



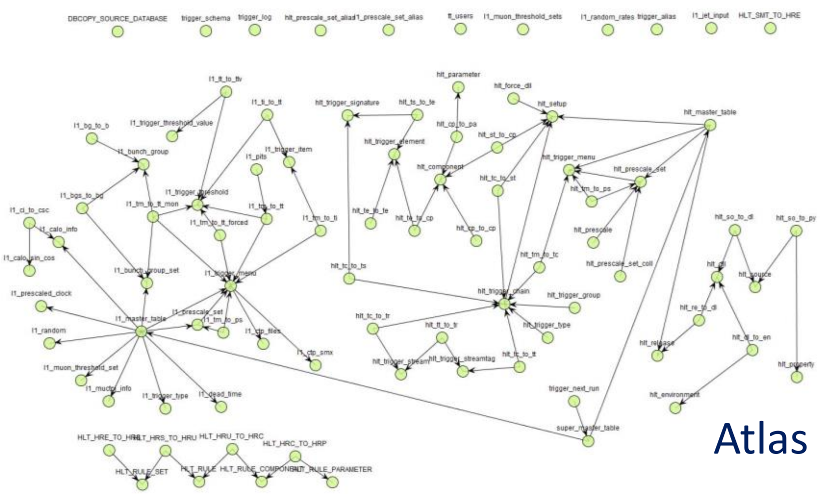
Additions & Updates: v. 1.0.8

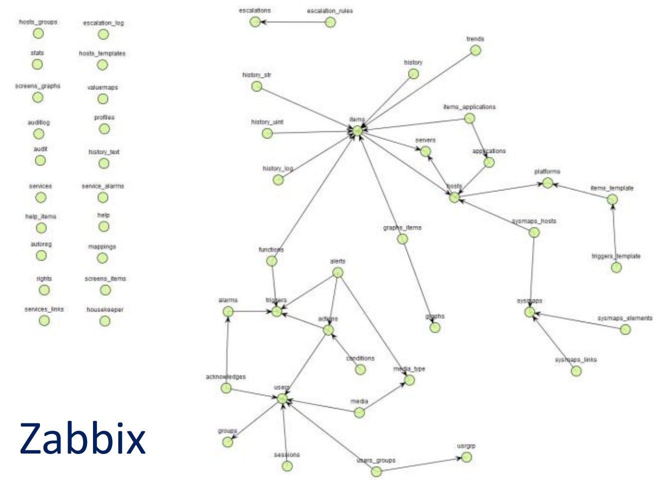


Additions: v. 1.0.15

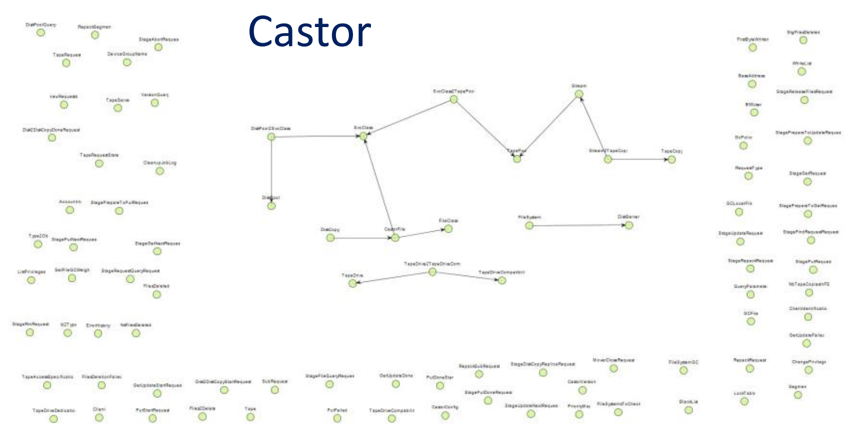


Final version: v. 1.0.17





Topological Patterns



Context and background

Study Setup

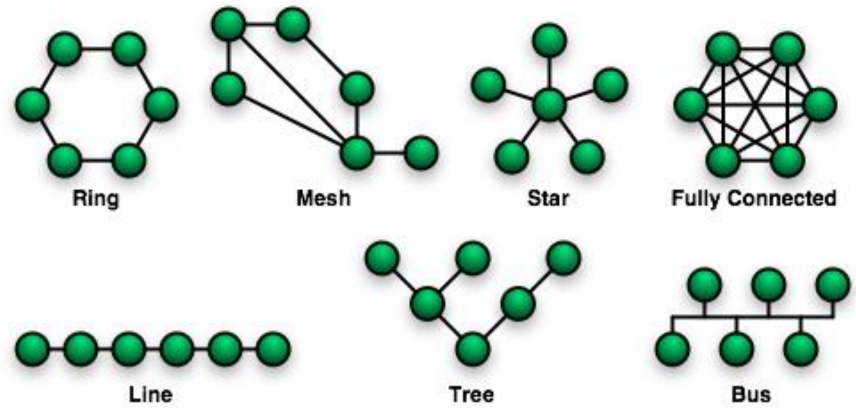
Topological Patterns

Topology and Evolution

Conclusions

Definition of Topology

- In network theory, topology is defined as the arrangement of a network's nodes and links
- In our work, a table's (node's) topology describes the pattern of edges (FKs) surrounding it



Source: https://commons.wikimedia.org/wiki/File:NetworkTopologies.png

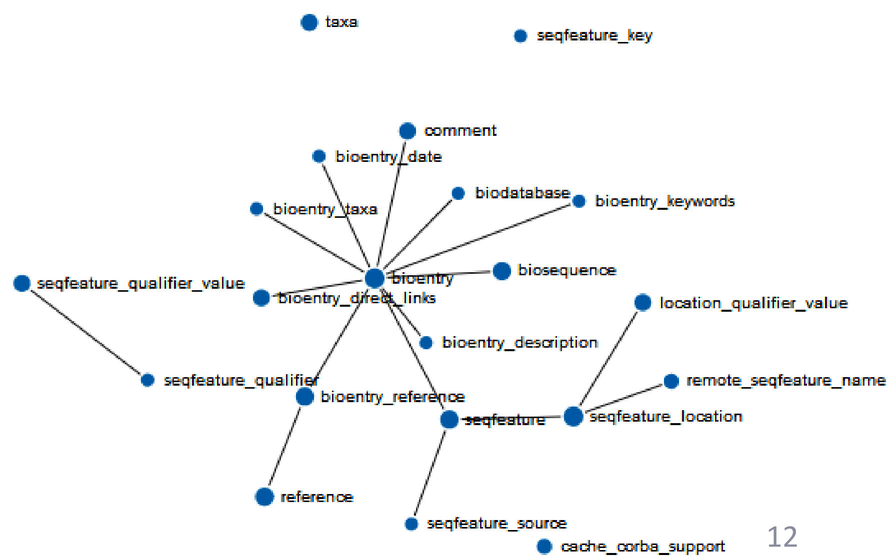
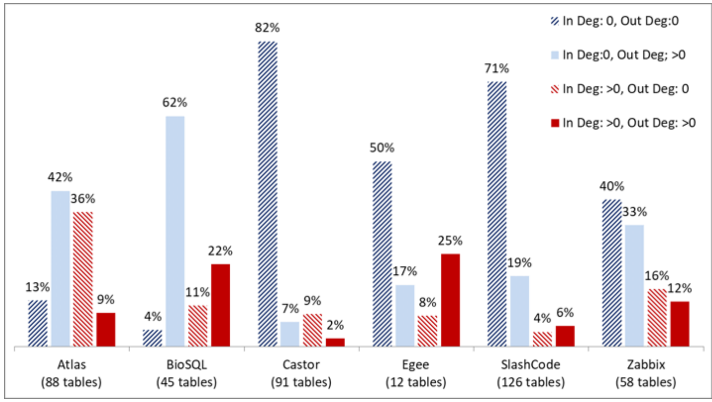
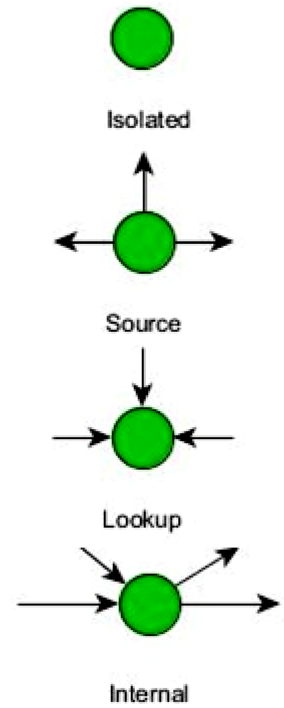


Table Topology



- Defined 4 topological categories for the tables of a schema
  - **Isolated** tables with no inciting edges
  - **Source** tables with only outgoing edges
  - **Lookup** tables with only incoming edges
  - **Internal** with at least 1 incoming and 1 outgoing edges



The problem of how to label tables with their topol. category

- Given just a single graph as input, the labeling of the tables is straightforward with a single pass over the nodes, as the categories are disjoint and independent of a node's neighborhood.
- Given a schema history, the labeling problem is different, as **a table can change labels over time**
- Unexpectedly, **this is rather rare**
- We have **manually** inspected all cases

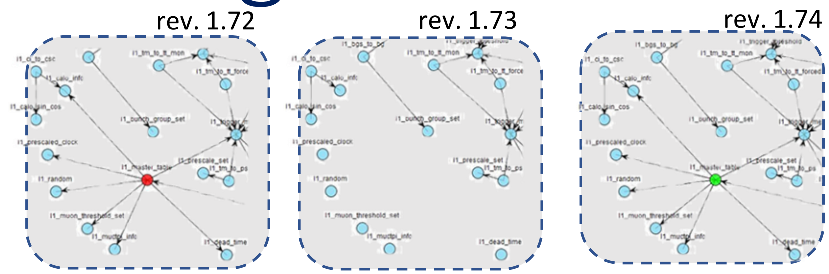
  & **assigned a single label** to each case

  by exploiting the **patterns of change**

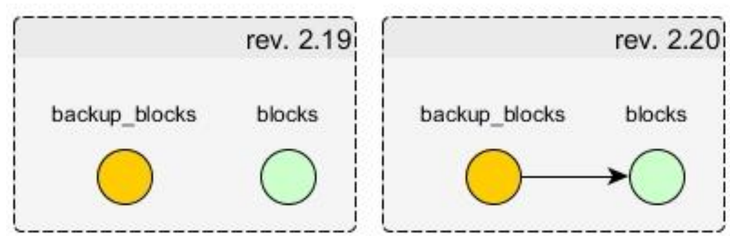
Datasets	Total #tables	#Tables with... single label	#Tables with... >1 label
Atlas	88	76	12
BioSQL	45	39	6
Castor	91	84	7
Egee	12	9	3
SlashCode	126	97	29
Zabbix	58	30	28

Multi-label Table-Changes

- **Ephemeral** transition to a different category and return to the former one

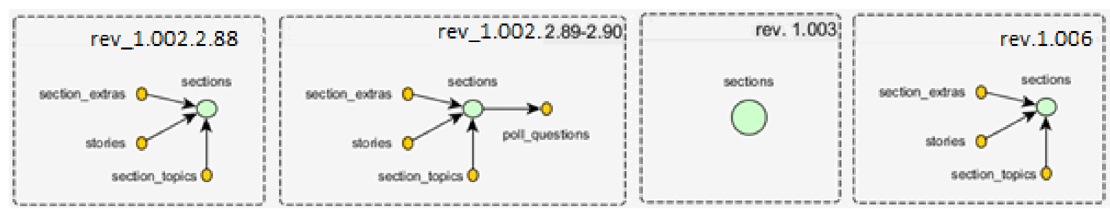


- **Change** from the **isolated** category to **another** category

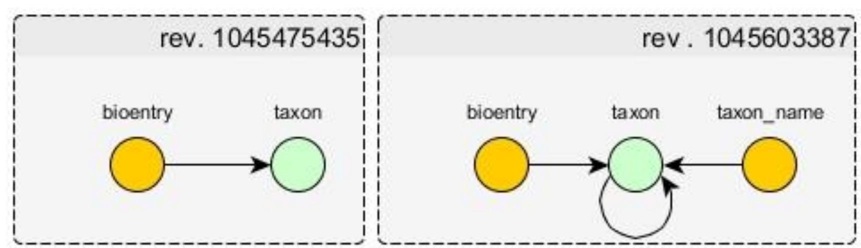


- **Change** of category **soon after** table's **birth**

- **Changes** leading to labels assigned for a short period



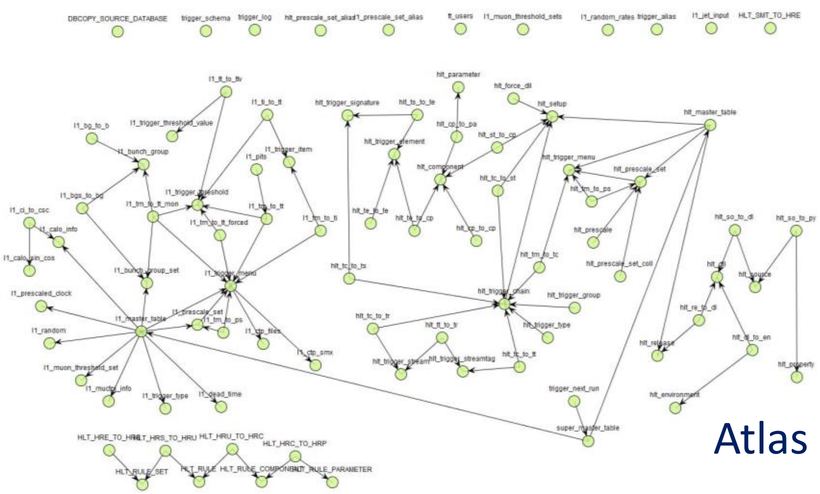
- **Changes** caused by the presence of self-references



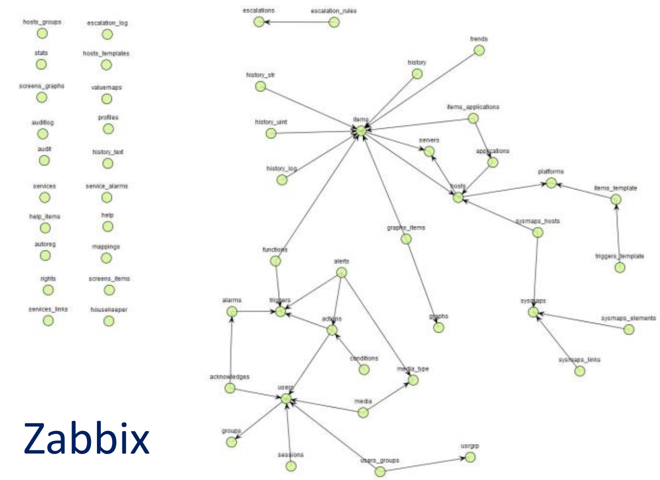
Simple resolution rule-set for assigning single-labels

Datasets	Type of Change: Ephemeral (DO-UNDO)	ISOLATED -> new category	Soon after birth	Short - lasting labels	Self-references	Other
Atlas	6	0	0	1	0	7
BioSQL	0	1	0	3	5	0
Castor	2	6	0	3	0	0
Egee	0	1	1	2	0	1
SlashCode	20	3	1	0	0	5
Zabbix	0	4	2	3	0	4

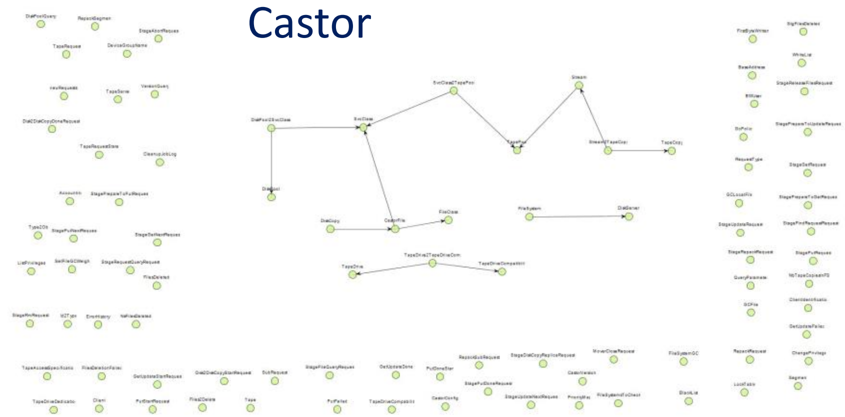
Rule	Description of Changes	Specific Criteria	Rule Decision
R0	No category change	$label_i = label_{i-1}$	The respective category
R1	Ephemeral category changes (DO-UNDO) in successive versions	$label_{i-1} = label_{i+1} \neq label_i$	Remove ephemeral $label_i$ and keep the remaining category $label_{i+1}$
R2	Changing from ISOLATED to another category	$label_{i-1} \neq label_i$, $label_{i-1} = ISO$	Remove *ISO* and keep the post-change label $label_i$
R3	Changing category in less than 10 versions after the First Known Version (FKV)	$label_{i-1} \neq label_i,\ i<10$	Remove the first labels and keep the post-change label $label_i$
R4	Changing to a category for a short period of less than 10 versions	$label_i = \ldots = label_{i+k}$, $k<10$, $label_{i-1} \neq \{label_i, \ldots, label_{i+k-1}\}$	Remove the period's labels & keep the pre-change label, $label_{i-1}$
R5	Changing category due to the presence of self-references	*An FK is added from the table to itself* $\Rightarrow label_i = INTERNAL$	Remove $label_i$ & keep the pre-change label, $label_{i-1}$
R6	Changes not abiding by any of the previous rules	-	Return the Most Frequent Label in the table's history



Atlas



Zabbix



Castor

Topology & Evolution

Context and background

Study Setup

Topological Patterns

Topology and Evolution

Conclusions

Is the evolution of a table related to its topology?

- We present results on birth and activity related to topological patterns

- … before proceeding, see also the stats
  - … when isolated are included, and,
  - … when excluded

%Table Population per Topological Pattern

Topological Category	Atlas	BioSQL	Castor	Egee	SlashCode	Zabbix
ISOLATED	13%	4%	**82%**	**50%**	**51%**	**39%**
SOURCE	**43%**	**64%**	7%	17%	32%	36%
LOOKUP	36%	18%	10%	8%	10%	20%
INTERNAL	8%	13%	1%	25%	6%	6%
Total	**88**	**45**	**91**	**12**	**68**	**56**

%Table Population (without ISOLATED)

Topological Category	Atlas	BioSQL	Castor	Egee	SlashCode	Zabbix
SOURCE	**49%**	**67%**	38%	33%	**67%**	**59%**
LOOKUP	42%	19%	**56%**	17%	21%	32%
INTERNAL	9%	14%	6%	**50%**	12%	9%
Total	**77**	**43**	**16**	**6**	**33**	**34**

Research Question: how is the topological category of a table related to the probability of being born in the originating version of the schema history?

- **Internal** & **lookup:** birth very likely at 1st version
  - Internal: 100% for 3 of 5 data sets; lookup: in majority of cases for 4 of 5 data sets
  - => UNLIKELY to be born later!!

Probability To Be Born @V0 Per Topological Category (Percentages Over #Tables Of Each Topological Category)

	Isolated		Source		Lookup		Internal		ALL TABLES	
	#Tables	Born @v0	#Tables	Born @v0	#Tables	Born @v0	#Tables	Born @v0	#Tables	Born @v0
Atlas	11	***9%***	38	61%	32	**78%**	7	**100%**	88	64%
BioSQL	2	**100%**	29	38%	8	50%	6	**67%**	45	47%
Castor	75	64%	6	**83%**	9	**89%**	1	**100%**	91	68%
SlashCode	35	***43%***	22	**73%**	7	**86%**	4	**100%**	68	60%
Zabbix	22	***9%***	20	30%	11	**45%**	3	**67%**	56	27%

Research Question: how is the topological category of a table related to the probability of being born in the originating version of the schema history?

- **Isolated** and **source** have higher chances to appear in later versions compared to the previous two categories

Probability To Be Born @V0 Per Topological Category (Percentages Over #Tables Of Each Topological Category)

	Isolated		Source		Lookup		Internal		ALL TABLES	
	#Tables	Born @v0	#Tables	Born @v0	#Tables	Born @v0	#Tables	Born @v0	#Tables	Born @v0
Atlas	11	***9%***	38	61%	32	**78%**	7	**100%**	88	64%
BioSQL	2	**100%**	29	38%	8	50%	6	**67%**	45	47%
Castor	75	64%	6	**83%**	9	**89%**	1	**100%**	91	68%
SlashCode	35	***43%***	22	**73%**	7	**86%**	4	**100%**	68	60%
Zabbix	22	***9%***	20	30%	11	**45%**	3	**67%**	56	27%

Research Question: how is the topological category of a table related to the probability of being born in the originating version of the schema history?

- *Internal and lookup tables are more likely to be born in the originating version of their dataset's history, which, expressed in a different way, means that it is quite unlikely that they are " born" after this first version.*
- *In contrast, source tables follow the trend of the general population and isolated tables are the ones with higher chances to be born in versions succeeding the originating one.*

Probability To Be Born @V0 Per Topological Category (Percentages Over #Tables Of Each Topological Category)

	Isolated		Source		Lookup		Internal		ALL TABLES	
	#Tables	Born @v0	#Tables	Born @v0	#Tables	Born @v0	#Tables	Born @v0	#Tables	Born @v0
Atlas	11	***9%***	38	61%	32	**78%**	7	**100%**	88	64%
BioSQL	2	**100%**	29	38%	8	50%	6	**67%**	45	47%
Castor	75	64%	6	**83%**	9	**89%**	1	**100%**	91	68%
SlashCode	35	***43%***	22	**73%**	7	**86%**	4	**100%**	68	60%
Zabbix	22	***9%***	20	30%	11	**45%**	3	**67%**	56	27%

Research Question: is there a relationship between the topological category of a table and its update activity?

- 3 activity categories*
  - **Rigid**
  - **Quiet**
  - **Active**

Breakdown of Tables over their Activity Class
(Percentages over Total #Tables)

	Total #Tables	Activity Class			Activity Class (%)		
		RIGID	QUIET	ACTIVE	RIGID	QUIET	ACTIVE
Atlas	88	18	43	27	***20%***	**49%**	31%
BioSQL	45	16	13	16	**36%**	***29%***	**36%**
Castor	91	57	31	3	**63%**	34%	***3%***
SlashCode	68	15	38	15	***22%***	**56%**	***22%***
Zabbix	56	23	30	3	41%	**54%**	***5%***

Quiet tables are the largest group in 3/5 datasets; when not, **rigid** are the most populous

*: P. Vassiliadis, A. Zarras, I. Skoulis , How is Life for a Table in an Evolving Relational Schema? Birth, Death and Everything in Between, ER 2015

Research Question: is there a relationship between the topological category of a table and its update activity?

- Isolated: mostly rigid and very rarely active!
- Source: follow the overall pattern of their dataset (also due to population) => mostly quiet or rigid, and rarely active.
- Lookup: more prone to changes wrt above ones, and wrt the overall dataset.
- Internal: mostly active, with probability higher than in any other activity category!

PROBABILITY FOR A TABLE OF A TOPOLOGICAL CATEGORY TO DEVELOP A CERTAIN UPDATE ACTIVITY (PERCENTAGES OVER TOTAL #TABLES OF EACH TOPOLOGICAL CATEGORY)

TOPOLOGICAL CATEGORY

	ISOLATED				SOURCE				LOOKUP				INTERNAL				Aggregate per Activity Class			
	Total #Tables	RIGID	QUIET	ACTIVE	Total #Tables	RIGID	QUIET	ACTIVE	Total #Tables	RIGID	QUIET	ACTIVE	Total #Tables	RIGID	QUIET	ACTIVE	Total #Tables	RIGID	QUIET	ACTIVE
Atlas	11	27%	55%	18%	38	29%	58%	13%	32	13%	47%	41%	7	0%	0%	100%	88	20%	49%	31%
BioSQL	2	100%	0%	0%	29	34%	31%	34%	8	25%	38%	38%	6	33%	17%	50%	45	36%	29%	36%
Castor	75	67%	32%	1%	6	67%	17%	17%	9	33%	56%	11%	1	0%	100%	0%	91	63%	34%	3%
SlashCode	35	34%	54%	11%	22	14%	68%	18%	7	0%	43%	57%	4	0%	25%	75%	68	22%	56%	22%
Zabbix	22	55%	41%	5%	20	35%	65%	0%	11	27%	73%	0%	3	33%	0%	67%	56	41%	54%	5%

Research Question: is there a relationship between the topological category of a table and its update activity?

- Rigid tables: mostly isolated or source. The probability for a rigid table to be lookup very low and almost zero for internals.
- Quiet tables distribution ~ aggregate distribution in all datasets. Closely follows the most populous category (source, isolated) too.
- Active tables are strongly inclined towards higher topological complexity, esp. internals, much higher than their dataset's distribution.

PROBABILITY FOR A TABLE OF AN ACTIVITY CLASS TO BELONG TO A CERTAIN TOPOLOGICAL CATEGORY (PERCENTAGES OVER TOTAL #TABLES OF EACH ACTIVITY CLASS)

	ACTIVITY CLASS																			
	RIGID					QUIET					ACTIVE					Aggregate per Topological Category				
	Total #Tables	ISOLATED	SOURCE	LOOKUP	INTERNAL	Total #Tables	ISOLATED	SOURCE	LOOKUP	INTERNAL	Total #Tables	ISOLATED	SOURCE	LOOKUP	INTERNAL	Total #Tables	ISOLATED	SOURCE	LOOKUP	INTERNAL
Atlas	18	17%	61%	22%	0%	43	14%	51%	35%	0%	27	7%	19%	48%	26%	88	13%	43%	36%	8%
BioSQL	16	13%	63%	13%	13%	13	0%	69%	23%	8%	16	0%	63%	19%	19%	45	4%	64%	18%	13%
Castor	57	88%	7%	5%	0%	31	77%	3%	16%	3%	3	33%	33%	33%	0%	91	82%	7%	10%	1%
SlashCode	15	80%	20%	0%	0%	38	50%	39%	8%	3%	15	27%	27%	27%	20%	68	51%	32%	10%	6%
Zabbix	23	52%	30%	13%	4%	30	30%	43%	27%	0%	3	33%	0%	0%	67%	56	39%	36%	20%	5%

Research Question: is there a relationship between the topological category of a table and its update activity?

The topological category of a table is quite strongly related to its update activity. Isolated and source tables are inclined towards zero or few updates in their lifetime, lookup tables with few or many changes and internal tables with an inclination to active lives with many updates.

PROBABILITY FOR A TABLE OF A TOPOLOGICAL CATEGORY TO DEVELOP A CERTAIN UPDATE ACTIVITY (PERCENTAGES OVER TOTAL #TABLES OF EACH TOPOLOGICAL CATEGORY)

TOPOLOGICAL CATEGORY

	ISOLATED				SOURCE				LOOKUP				INTERNAL				Aggregate per Activity Class			
	Total #Tables	RIGID	QUIET	ACTIVE	Total #Tables	RIGID	QUIET	ACTIVE	Total #Tables	RIGID	QUIET	ACTIVE	Total #Tables	RIGID	QUIET	ACTIVE	Total #Tables	RIGID	QUIET	ACTIVE
Atlas	11	27%	55%	18%	38	29%	58%	13%	32	13%	47%	41%	7	0%	0%	100%	88	20%	49%	31%
BioSQL	2	100%	0%	0%	29	34%	31%	34%	8	25%	38%	38%	6	33%	17%	50%	45	36%	29%	36%
Castor	75	67%	32%	1%	6	67%	17%	17%	9	33%	56%	11%	1	0%	100%	0%	91	63%	34%	3%
SlashCode	35	34%	54%	11%	22	14%	68%	18%	7	0%	43%	57%	4	0%	25%	75%	68	22%	56%	22%
Zabbix	22	55%	41%	5%	20	35%	65%	0%	11	27%	73%	0%	3	33%	0%	67%	56	41%	54%	5%

PROBABILITY FOR A TABLE OF AN ACTIVITY CLASS TO BELONG TO A CERTAIN TOPOLOGICAL CATEGORY (PERCENTAGES OVER TOTAL #TABLES OF EACH ACTIVITY CLASS)

ACTIVITY CLASS

	RIGID					QUIET					ACTIVE					Aggregate per Topological Category				
	Total #Tables	ISOLATED	SOURCE	LOOKUP	INTERNAL	Total #Tables	ISOLATED	SOURCE	LOOKUP	INTERNAL	Total #Tables	ISOLATED	SOURCE	LOOKUP	INTERNAL	Total #Tables	ISOLATED	SOURCE	LOOKUP	INTERNAL
Atlas	18	17%	61%	22%	0%	43	14%	51%	35%	0%	27	7%	19%	48%	26%	88	13%	43%	36%	8%
BioSQL	16	13%	63%	13%	13%	13	0%	69%	23%	8%	16	0%	63%	19%	19%	45	4%	64%	18%	13%
Castor	57	88%	7%	5%	0%	31	77%	3%	16%	3%	3	33%	33%	33%	0%	91	82%	7%	10%	1%
SlashCode	15	80%	20%	0%	0%	38	50%	39%	8%	3%	15	27%	27%	27%	20%	68	51%	32%	10%	6%
Zabbix	23	52%	30%	13%	4%	30	30%	43%	27%	0%	3	33%	0%	0%	67%	56	39%	36%	20%	5%

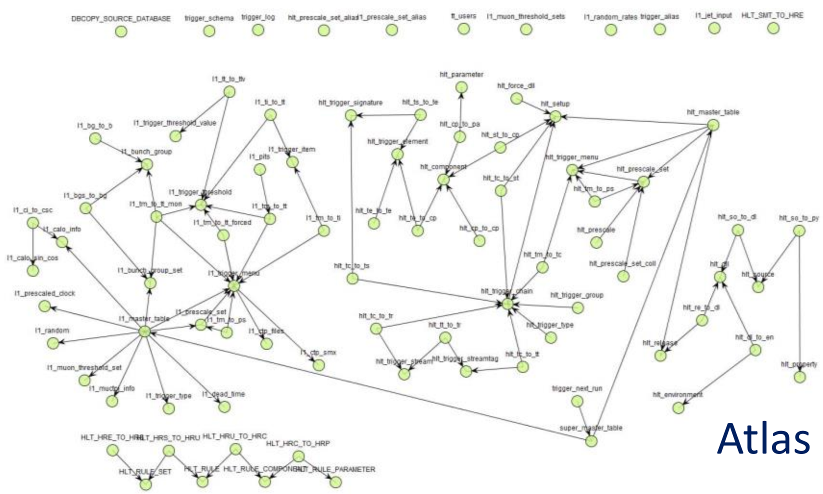
Analysis: Resize of table schemata

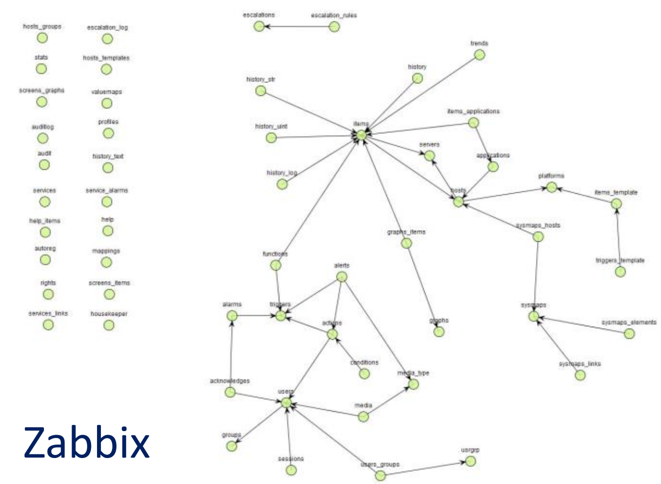
- Table schema resize: ratio |#attr@last v.|/ |#attr@first v.| of the table
- Overall, at least half the tables remain steady and 25%-47% increase their schema, 2% - 6% reduce their schema
- **Internal** and **lookup** more prone to increase their size than the dataset's avg., less prone to remain steady or reduce
- **Isolated** and **source** have higher prob. to remain steady than the dataset's avg., less prone to increase or reduce

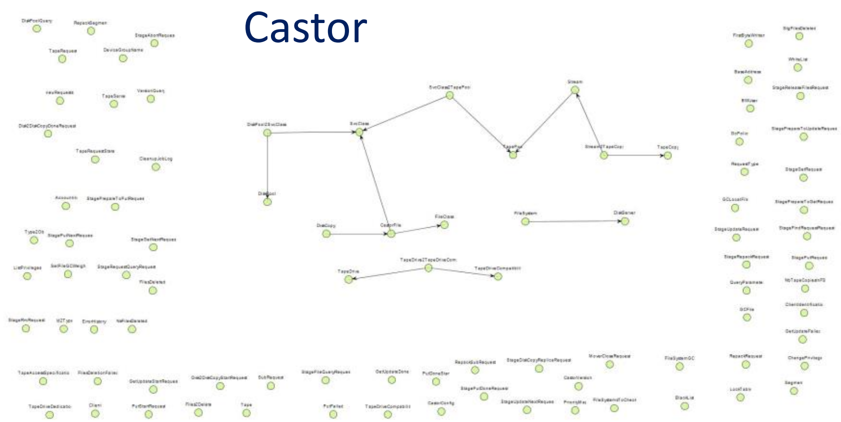
PROBABILITY FOR A TABLE OF A TOPOLOGICAL CATEGORY TO HAVE CERTAIN SIZE SCALE (PERCENTAGES OVER TOTAL #TABLES OF EACH TOPOLOGICAL CATEGORY)

TOPOLOGICAL CATEGORY

	ISOLATED				**SOURCE**				**LOOKUP**				**INTERNAL**				**Aggregate per Size Scale Category**			
	Total #Tables	<=0,99	1	>1	Total #Tables	<=0,99	1	>1	Total #Tables	<=0,99	1	>1	Total #Tables	<=0,99	1	>1	Total #Tables	<=0,99	1	>1
Atlas	11	**9%**	**73%**	18%	38	**5%**	**82%**	13%	32	**3%**	**59%**	38%	7	**14%**	**43%**	**43%**	88	6%	69%	25%
BioSQL	2	0%	**100%**	0%	29	**10%**	**62%**	28%	8	0%	**25%**	**75%**	6	0%	**33%**	**67%**	45	7%	53%	40%
Castor	75	**1%**	**72%**	27%	6	0%	**67%**	**33%**	9	**22%**	33%	**44%**	1	0%	0%	**100%**	91	3%	67%	30%
SlashCode	35	**3%**	**69%**	29%	22	**5%**	41%	**55%**	7	0%	**14%**	**86%**	4	0%	0%	**100%**	68	3%	50%	47%
Zabbix	22	**5%**	**68%**	27%	20	0%	**55%**	**45%**	11	0%	**36%**	**64%**	3	0%	**33%**	**67%**	56	2%	55%	43%







Conclusions

Context and background

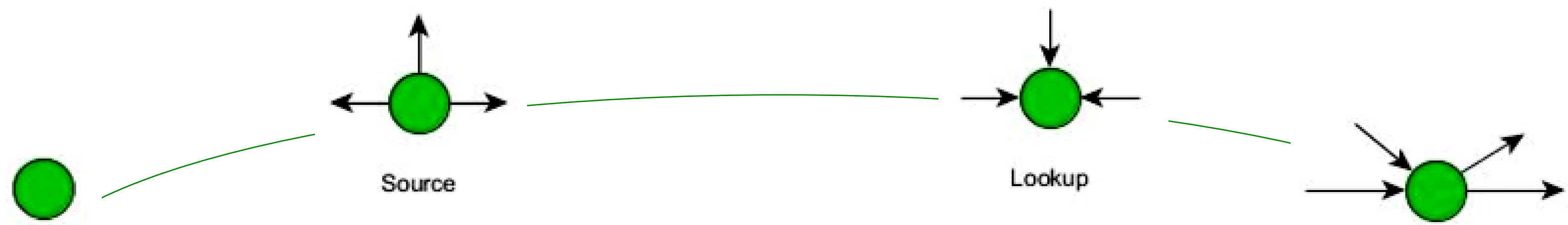
Study Setup

Topological Patterns

Topology and Evolution

Conclusions

A spectrum of (increasing) complexity



Isolated
- Almost rigid
- Frequently born after @V0

Source
- Similar to ISO
- Typically resist change
- Not unlikely to be born after @V0

Lookup
- Similar to INT
- Prone to demonstrate change
- Rarely born after @V0

Internal
- fewest of all
- high activity
- mostly born @V0

Isolated / Source	Lookup / Internal
Most populous	Less populous
Less active	Most active
Easier to add later	Harder to add later

As time passes, people
- are disinclined to add more complex structures to their database;
- are more comfortable with adding new simple structures;
- update complex structures with attribute injection when necessary.

Gravitation to rigidity explains why!

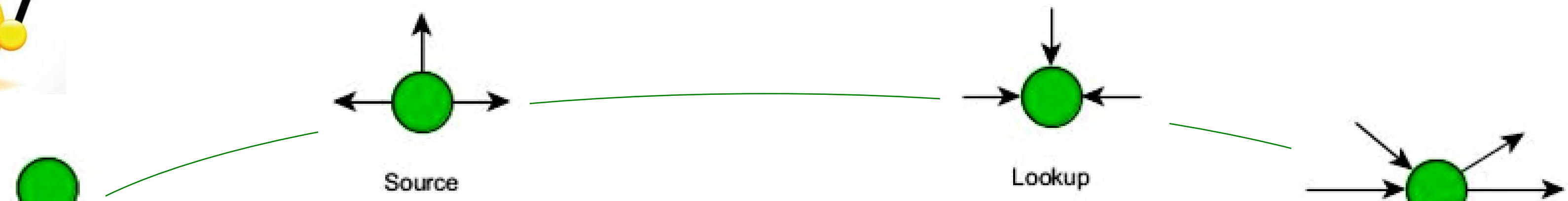
- Gravitation to rigidity refers to the difficulty of altering the schema of a database when surrounding code is built upon it. See it working here:
  - Topologically simple tables are much more populous and easy to create than complex and active ones;
  - Very few tables change topological category, with most changes in the ephemeral or short-lasting categories of label-changes;
  - Most of the activity of the high-end of the complexity spectrum is due to the addition of attributes to the existing tables, …
  - … quite differently from the lower end of the spectrum, where administrators are more inclined towards building new tables.
- Maintenance-by-addition, equiv., avoid-to-break-the-code principle: adding new info via (expendable) new tables, does not result in the necessity to update the surrounding code.
- FOSS projects are built to be selected by other organizations. Upgrading the schema in the presence of existing data is a painful experience, and simple structures and maintenance-by-addition reduce this pain

Why bother?

- Our empirical study on how schema evolution relates to foreign keys in FOSS projects …
  - advances our knowledge with solid evidence,
  - provides both maintenance clues to curators and evaluators of FOSS projects, and,
  - provides insights to the research community on practical problems.

- **Project curators** can expect that the schema in the future will expand in terms of (a) topologically simple structures and (b) complex topological structures. Enforcing maintenance-by-addition will allow lower impact to the surrounding code.
- **FOSS Evaluators**, when selecting a software projects for adoption, will need to also assess the threats posed by the absence of (a) foreign keys and (b) maintenance actions from the side of the curators.
- **Researchers** must understand that the nature of the situation boils down to the fundamentals of the relational model and how relational databases can be coupled to surrounding applications. Must also go for
  - More flexible ways of building applications on top of databases
  - Tools that accurately highlight the points of maintenance in the surrounding code, in the event of schema evolution.

The more topologically complex tables are, the fewer, the more active (via attribute injection), and the harder to add as the schema ages!



Isolated

- Almost rigid
- Frequently born after @V0

Source

- Similar to ISO
- Typically resist change
- Not unlikely to be born after @V0

Lookup

- Similar to INT
- Prone to demonstrate change
- Rarely born after @V0

Internal

- fewest of all
- high activity
- mostly born @V0

Most populous
Less active
Easier to add later

Less populous
Most active
Harder to add later

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

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