Noname manuscript No.
(will be inserted by the editor)

Schema Evolution Survival Guide for Tables: Avoid Rigid Childhood and You 're En Route to a Quiet Life

Panos Vassiliadis · Apostolos V. Zarras

This is a post-peer-review, pre-copyedit version of an article published in Journal of Data Semantics (JODS), December 2017, Volume 6, Issue 4, pp 221-241. The final authenticated version is available online at: http://dx.doi.org/10.1007/s13740-017-0083-x

November 16, 2017

Abstract In this paper, we study the factors that relate to the survival of a table in the context of schema evolution in open-source software. We study the history of the schema of eight open source software projects that include relational databases and extract patterns related to the survival or death of their tables. Our study shows that the probability of a table with a wide schema (i.e., a large number of attributes) being removed is systematically lower than average. Activity and duration are related to survival too. Rigid tables, without any change to their schema, are more likely to be removed than tables that sustain changes. Durations of dead and survival tables demonstrate a mirror image: dead tables' durations are mostly short, whereas survivor tables gravitate towards higher durations. Our findings are mostly summarized by a pattern, which we call *electrolysis pattern*, due to its diagrammatic representation, stating that dead and survivor tables live quite different lives: tables typically die shortly after birth, with short durations and mostly no updates, whereas survivors mostly live quiet lives with few updates – except for a small group of tables with high update ratios that are characterized by high durations and survival. Equally important is the evidence that schema evolution suffers from the antagonism of *gravitation to rigidity*, i.e., the tendency to minimize evolution as much as possible in order to minimize the resulting impact to the surrounding code. Several factors contribute to this observation: the absence of long durations in removed tables, the low percentage of tables whose schema size is scaled up or down, and the low numbers of tables with a high rate of updates, contrasted to the high numbers of tables with zero or few updates. We complement our findings with explanations and recommendations to developers.

1 Introduction

It is well known that software evolution and maintenance takes up more than half the resources of a software project. Understanding the mechanics, patterns and

Dept. of Computer Science
University of Ioannina (Hellas)
E-mail: {pvassil, zarras}@cs.uoi.gr

laws of software evolution allows us to proactively plan software development and design in order to save time and resources. When it comes to the case of data-intensive information systems, a large part of the evolution of software is related to the evolution of the database schema.

Background. Studying schema evolution to dig out patterns is thus important in our attempt to understand its mechanics and plan software design and development on top of databases. However, this problem has attracted little attention by the research community so far. To a large extent, the possibility of actually studying schema evolution emerged from the availability of schema histories embedded in open source software projects, publicly available via Github. So far, research efforts [10], [2], [7], [16], [9] – see Sec. 2– have demonstrated that schemata grow over time, mostly with insertions and updates, and are frequently out of synch with their surrounding code. However, we are still far from a detailed understanding of how individual tables evolve and what factors affect their evolution. In our latest work [14, 15], we have performed a first study towards charting the relationship of factors like schema size and version of birth to the duration and the amount of change a table undergoes. In this paper, we continue along this line of work by answering two fundamental questions around the survival of a table that have not been answered so far: (a) "what factors affect survival after all?", and, in particular, (b) "how are survival, activity behavior and duration of a table interrelated?". To the best of our knowledge, the problem was only initially touched in [14, 15] and the insights of this paper are completely novel in the related literature.

Importance. Why is the knowledge of patterns in life and death of tables so important? As already mentioned, table removal or update requires maintenance of the application code. Therefore, *understanding the probability of update or removal of a table can aid the development team in avoiding to invest too much effort and code to high-risk parts of the database schema.* Equally importantly, we believe that our study gives solid evidence on a phenomenon that we call *gravitation to rigidity*[1] stating that despite some valiant efforts, relational schemata suffer from the tendency of developers to minimize evolution as much as possible in order to minimize the resulting impact to the surrounding code. *Understanding the mechanics of the gravitation to rigidity is imperative in allowing us to fight it and provide more flexible ways of developing code over evolving schemata.*

Experimental setup. Following the research method of our previous work, we have performed a large study of eight data sets with the schema history of databases included in open source projects (see Sec. 3 for our experimental setup). We have extracted the changes that occurred between subsequent versions of these histories and, for each table of each data set, we have computed statistics that highlight its evolutionary profile. The measures that we have assessed *per table* include (a) information on the version/date of its birth and death (if applicable), and duration, (b) information on its initial, ending and average schema size (in number of attributes), along with its resizing ratio between first and last version, and (c) information on its update profile, including the total number of changes it went through, the change rate, etc. We have classified tables in profiles concerning (a) their *survival* (i.e., their presence in the last version of the schema history or not), characterizing them as *survivors* or *dead*, (b) their *activity behavior*, charac-

[1] *Rigidity* is used in its software engineering meaning, referring to software that is hard to evolve and maintain.

terizing them as *rigid* (if they go through zero updates), *active* (if their rate of change is higher than 0.1 changes per transition) and *quiet* otherwise, and (c) by the combination of the above via their Cartesian product, which we call *LifeAnd-Death* profile. Once all these data were available, we studied how different factors belonging to the above list correlate to the survival and duration of a table.

Findings. Our results are detailed in Sec. 4 and Sec. 5; here, we can give a concise summary of our findings as follows. Starting with Sec. 4, the study of schema size reveals that the thinner the schema of a table is, the higher the chances of the table being removed; oppositely, the few wide tables, are survivors in their vast majority. Changes in the size of the schema are largely pertaining to survivor tables; however, both schema scale up and down are infrequent, with one in three survivors going through a scale up in its number of attributes and one in twenty through a scale down. Coming to the role of activity and duration, explored in Sec. 5, our exploration of activity indicates that rigid tables are more likely to be removed than tables that have gone through changes. Also, the antithesis of the durations between dead and survivor tables is striking: table deletions take place shortly after birth, resulting in short durations for the dead tables; this is to be contrasted with the large number of survivors with high (and frequently, maximum) durations. When activity profile, duration and survival are studied together, we observe the *electrolysis pattern*, named after the paradigm of positive and negative ions in electrolysis moving towards opposite directions: Not only dead tables cluster in short or medium durations, and practically never at high durations, but also, with few exceptions, the less active dead tables are, the higher the chance to reach shorter durations. In contrast, survivors are mostly located at medium and high durations and the more active survivors are, the stronger they are attracted towards high durations, with a significant such inclination for the few active survivors, that cluster in very high durations.

Finally, in Sec. 6, we provide detailed discussion on (a) our explanations to our observations, (b) the threats to validity of our experimental setup, (c) the usefulness of this research along with recommendations to developers, and finally, (d) roads for future work.

2 Background and Related Work

Schema evolution as an area spans a wide range of topics, in both the database and the software engineering engineering disciplines. We refer the reader who wishes to explore different topics under the umbrella of schema evolution, and especially techniques on how to practically handle schema evolution, to a comprehensive survey on the management of evolution [4] on the topics of handling evolution of relational and XML data as well as of ontologies as well as to a more recent and detailed survey [8] also covering the state of practice and the area of data warehousing. Here, we avoid extending the discussion to the broader field of works that lie beyond the scope of the paper and strictly focus on the area dedicated to the study of schema evolution in order to discover patterns and regularities. It is noteworthy, that the related work on the study of regularities of schema evolution is not abundant and, in effect quite recent.

The first known study, published in 1993 [10], monitored the database of a health management system for 18 months, to report the overall increase of schema

size over time and the percentage breakdown for different types of changes. After this study, it was only 15 years later that research was revived on the problem. The key to this revival was the existence of open source software repositories exposing all the code of a software project in all its history. Software projects based on relational databases, thus, would expose the entire history of their schema. There is a handful of works since the late '00s [2] [7] [16] [9] that have assessed the evolution of databases involved in open source software projects.

In [2], the authors report findings on the evolution of Mediawiki, the software that supports Wikipedia. The authors of this work, and also in the followup work on "algebrizing" schema modification operations [3] should be accredited for the public release of schema histories that they collected. The algebra of schema modification operations has been reshaped with a relationally complete set at [5]. Several works followed, where the authors have primarily worked on (a) the schema size, which grows over time but with progressively less rate [9], (b) the absence of total synchronization between source code and database schema, as schemata evolve [7] [16], and, (c) the impact of schema change to the surrounding code [9], which requires a significant amount of code modifications. [9] is also presenting preliminary results on the timing of the schema modifications, reporting that the early versions of the database included a large part of the schema growth. A study presented in [1] verifies the observations of other works concerning the trend of increase in schema size and the reluctance in the deletion of tables.

Recently, we have being involved in the study of schema evolution for databases that are embedded in open-source software projects. We have used as input the information extracted from open source code repositories like github and sourceforge for 8 projects with a long history (on average 6-8 years) and from different domains (specifically, physics, biomedical, and CMS's).

In [11], also presented in full length in [12], we have worked at the macroscopic level to study how the schema of a database evolves in terms of its size (Fig. 1). Our main vehicle for generating research questions has been the set of laws for software evolution, or else Lehman's laws [6] adapted for the case of schema evolution. We have measured schema size both for its evolution over time and for its incremental growth. We have found evidence that *schemata grow over time in order to satisfy new requirements, albeit not in a continuous or linear fashion, but rather, with bursts of concentrated effort interrupting longer periods of calmness*. Overall, *growth is small*, with average growth being close to zero. It is noteworthy that growth also comes with drops in schema size that signify the existence of perfective maintenance.

In a subsequent study, [14], also presented in full length in [15], we have worked on the identification of frequently encountered patterns on table properties (e.g., birth, duration, amount of change). So, whereas previous work had mostly focused on the macroscopic study of the entire database schema, in this line of work, we zoomed into the lives of tables. We identified four major patterns on the relationship of such properties (Fig. 2). Specifically, these patterns are as follows:

- The *Γ pattern* studies the interrelationship of the schema size of a table at its birth with its overall duration and indicates that tables with large schemata tend to have long durations and avoid removal.
- The *Comet pattern* studies the interrelationship of the schema size of a table at its birth with its total amount of updates and indicates that the tables with most updates are frequently the ones with medium schema size.

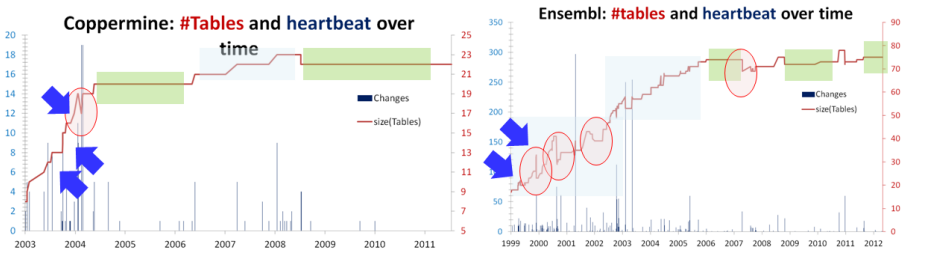

Fig. 1 Summary of [12] with schema growth over time (red continuous line) along with the heartbeat of changes (spikes) for two of our datasets. Overlayed darker green rectangles highlight the calmness periods, and lighter blue rectangles highlight smooth expansions. Arrows point at periods of abrupt expansion and circles highlight drops in size.

– The *Inverse Γ pattern* studies the interrelationship of the amount of updates in the life of a table with its duration and indicates that tables with medium or small durations produce amounts of updates lower than expected, whereas tables with long duration expose all sorts of update behavior.
– The *Empty Triangle* pattern studies the interrelationship of a table's version of birth with its overall duration and indicates a significant absence of tables of medium or long durations that were removed –thus, an empty triangle – signifying mainly short lives for deleted tables and low probability of deletion for old timers.

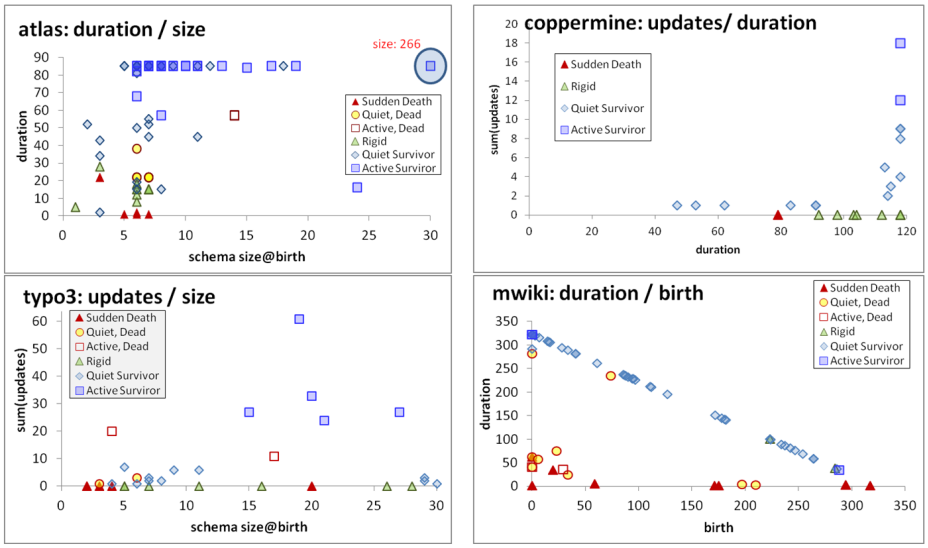

Fig. 2 The 4 patterns of [14, 15]: Gamma (top left), inverse Gamma (top right), comet (bottom left) and empty triangle (bottom right).

The observed evidence indicates that databases are rigidity-prone rather than evolution-prone. We have called the phenomenon *gravitation to rigidity* and we

attribute it to the implied impact to the surrounding code that a modification to the schema of a database has.

Although insightful, the aforementioned findings *have not exhausted the search on factors affecting survival*, and so, in this paper, we extend our knowledge by exploring *how survival is related to schema size, schema resizing (scale up or down in the number of attributes of a table), and, of course, how survival, duration and activity profile interrelate*. The current paper is a detailed version of [13], which it extends, with (a) all the results of Sec. 4 and Sec. 5.1 on how schema size, schema resizing, year-of-birth, and, activity behavior relate to survival, (b) with a deeper investigation of the combination of duration, survival, and activity and (c) with a more detailed presentation of all the material in [13]. To the best of our knowledge this is the first comprehensive study of this kind in the literature.

3 Experimental Method

In this section, we briefly present the eight datasets that we have collected and processed for our study. We have relied on the data sets used in our previous studies too [11]. Our *Schema Biographies* website[2] contains links to all our results, data, code and presentations that are made publicly available to the research community.

Dataset	Type	Versions	Lifetime	Tables @ Start	Tables @ End
ATLAS Trigger	[P]	84	**2 Y**, 7 M, 2 D	56	73
BioSQL	[B]	46	**10 Y**, 6 M, 19 D	21	28
Coppermine	[C]	117	**8 Y**, 6 M, 2 D	8	22
Ensembl	[B]	528	**13 Y**, 3 M, 15 D	17	75
MediaWiki	[C]	322	**8 Y**, 10 M, 6 D	17	50
OpenCart	[C]	164	**4 Y**, 4 M, 3 D	46	114
phpBB	[C]	133	**6 Y**, 7 M, 10 D	61	65
Typo3	[C]	97	**8 Y**, 11 M, 0 D	10	23

Fig. 3 Datasets used in our study

Data sets. We have collected the version histories of 8 data sets that support open source software projects. The criteria that we used to pick these projects also specify the scope of our study. Specifically, we picked a set of projects that (a) are part of open-source software (and not proprietary ones), (b) have a long history, preferably both in terms of time and commits (third and fourth column in Fig. 3) and (c) cover different domains, and specifically, Content Management

[2] http://www.cs.uoi.gr/∼pvassil/projects/schemaBiographies/

Systems (denoted by [C] in Fig. 3), biomedicine (denoted by [B] in Fig. 3) and physics (denoted by [P]).

Concerning the scope of the study, we would like to clarify that we work only with changes at the logical schema level (and ignore physical-level changes like index creation or change of storage engine). Also, the reader is advised to avoid generalizing our findings to proprietary databases, outside the realm of open source software.

Experimental protocol. For each dataset we gathered as many schema *versions* (DDL files) as we could from their public source code repositories (cvs, svn, git). We have targeted only changes at the database part of the project as they were integrated in the trunk of the project. The files were collected during June 2013. For all of the projects, we focused on their release for MySQL (except ATLAS Trigger, available only for Oracle). The files were then processed by our tool, Hecate, that detected, in a fully automated way, (a) changes at the table-level, i.e., which tables were inserted and deleted, and (b) updates at the attribute-level, and specifically, attributes inserted, deleted, having a changed data type, or participation in a changed primary key.

Reported Measures. Hecate pair-wise compared subsequent files and reported the changes for each *transition* between subsequent versions. The details of each particular change along with collective statistics per table, as well as for the entire schema were also reported.

A. Measures on Life and Death. These include the identification of the version id (also: the date) of *Birth*, and, if applicable, *Death* of a table. *Birth* holds the transition id that marks the first occurrence of the table in the schema (0 included for tables that came with the first creation of the schema). As version id's are related to specific dates, we can also compute time from them. *Death* holds the transition id that marks the last occurrence of the table in the schema, in case this is earlier than the last known version of the schema history. The *Survival Class* measure classifies a table as a *survivor*, if it was present in the last known version of their schema history, or *dead* otherwise. The *Duration* of the table is computed accordingly, and is measured in both version id's and time.

B. Measures on Schema size. Schema size information is reported as (a) *SchemaSizeBirth*: the number of attributes of a table at its birth, (b) *SchemaSizeEnd*: the number of attributes of a table at its last occurrence in the schema, (c) *AvgSchemaSize*: the average number of attributes of a table throughout its life. We also define *SchemaResize* as the ratio *SchemaSizeEnd / SchemaSizeBirth*.

C. Measures of Update Activity. We distinguish two measures that are the basis for our analyses in this paper:

- the *sum of updates(SumUpd)* which is the total number of updates the table has gone through throughout its life and its normalized version, and
- *ATU*: Average Transitional amount of Updates, which is the ratio *SumUpd / Duration*.

To aid our analyses, we also use two classification measures, specifically:

- *Activity Class* is a characterization of activity behavior and classifies tables in three classes, specifically: (a) *rigid*, without any change (*SumUpd* is zero) throughout their lives, (b) *active*, if they have an *ATU* rate of more than 0.1 and more than 5 changes in their life, and (c) *quiet* otherwise.

Distribution of tables per schema size range (first column) and Survival Class (D:dead, S: survivors)...
...as pct over their category (attn: not overall; each column sums to 100%)

	atlas		biosql		ensembl		mwiki		ocart*		phpBB		typo3	
	D	S	D	S	D	S	D	S	D	S	D	S	D	S
1-5	27%	11%	**94%**	75%	**70%**	52%	**62%**	62%	**93%**	65%	40%	57%	**78%**	13%
6-10	**60%**	66%	6%	25%	25%	33%	24%	16%	7%	24%	**60%**	26%	0%	30%
>10	13%	23%	0%	0%	5%	15%	14%	22%	0%	11%	0%	17%	22%	57%

Fig. 4 Distribution of tables over schema size range × survival.

– *LifeAndDeath Class* is the Cartesian product of the measures *SurvivalClass* and *ActivityClass*. The *LifeAndDeath Class* characterizes a table both with respect to its survival and to its update profile during its lifetime. The measure's domain includes six values produced by the combination of {dead, survivor} × {rigid, quiet, active}.

4 How are dead and survivor tables differnt with respect to static properties?

In this section we address the problem of how survivor tables differ from the removed ones concerning static properties like the time of their birth or their schema size at their last known version. First, we present our results on how schema size relates to survival. Second, we expand our exploration to the resizing of tables' schemata during the lifetime of a table. Third, we explore the effect of the year of birth of a table to its survival.

4.1 Relationship of Schema Information to Survival

4.1.1 Schema Size At End and Survival

Does schema size make any difference concerning death and survival? We will work with the *schema size at end*, as we are specifically interested in what happens with the possibility of removal of a table. Interestingly, in all our data sets, the level of correlation of the *average schema size* with the *schema size at end* is very strong (on average, the Kendall correlation is 0.95). It is noteworthy that, as schemata do not change much, the correlations of different measures of schema are strong: *schema size at birth* has a Kendall correlation of 0.89 with the *average schema size* and 0.86 with the *schema size at end* All these information means that *schema size at end* is also representing the other schema size measures of a table (esp., average schema size) very well. In terms of descriptive statistics, in 6 out of 8 data sets, the tables whose schema size at end is within 1 – 10 attributes is 80% or more of the total population (exceptions: atlas reaches 78% and typo 57%). The data of Fig. 4 allow us to observe several interesting properties (note that Coppermine, with a single dead table, is excluded from the figures).

Thin tables. For the range of 1-5, the percentage of dead tables is typically larger for dead than for surviving tables. In 5 out of 7 cases, the percentage of this range for dead tables is more than 60%. This means that *dead tables have a strong*

tendency to be thin in their schema. Survivors also cluster with large percentages in this range, albeit lower than the dead ones, in all but one occasion. *In 5 out of 7 cases, tables in the range of 1-5 attributes have a higher chance of being removed than the average removal probability of their data set* (see the table in Fig. 5; contrast the first row to the row with the probability of death for the entire data set). However, this involves a significant difference in only 2 occasions (atlas and typo), thus, we cannot overemphasize this finding.

Wide tables. In 6 out of 7 studied data sets, the percentage of tables with more than 10 attributes is approximately 11% higher for survivors than for dead tables (see Fig. 4); in typo3 this difference rises to 35%. In absolute numbers, the number of wide tables that eventually died is not higher than 4 tables in any data set (including the deletion-prone ensemble). *In all occasions, the probability of a wide table being removed is quite lower than the average removal probability of the data set* (see the last 2 rows of the table in Fig. 5). In other words, wide tables constitute a much more significant part of the survivor group. This is in absolute consistency with the Gamma pattern of [14]: wide tables are strongly inclined to survive; the percentages of survivor wide tables lies above 73% for all data sets.

As an overall assessment, we can comment that the presented differences, and specifically, *the higher probability of thin tables to die, and the lower probability of wide tables to survive are consistent, although not grave (approximately in the area of 10%)*.

4.1.2 Relationship of Schema Resize and Survival

In this subsection, we discuss how survival relates to the resizing of a table's schema. Our first task is to assess how tables are distributed to categories according to their schema resizing behavior. We classify tables in three categories, specifically, (i) *no scale (NS)* or *flat* tables, who have a *SchemaResize ratio* of exactly 1.0, (ii) *ScaledUp* (S-U) tables with a *SchemaResize ratio* strictly higher than 1.0 and (iii) *ScaledDown* ($S-D$) tables, with a *SchemaResize ratio* strictly lower than 1.0. Note that no scale tables are not necessarily rigid: it is possible that they have sustained changes of key alterations, data type changes, etc.

The first observation that comes from Fig. 6 is that tables, in their broad majorities are no scale tables. The percentage of this category ranges between 50% and 77%, with an average of 63%: *almost 2 out of 3 tables do not have a modification of their schema size at the end of their observation*. Second largest category is the one of scaled up tables, ranging from 20% to 43% with an average of 32%. Scaled down tables are a small category, ranging from 1% to 9%.

As the number of dead scaled up tables is typically too small, both in absolute values and percentages (with the exception of the deletion-prone data set

Pct of dead tables over the population of ...	atlas	biosql	ensembl	mwiki	ocart*	phpBB	typo3
... thin tables (1-5 attr.)	33.33%	43.24%	58.95%	29.55%	14.94%	5.13%	70.00%
... tables with 6-10 attr.	15.79%	12.50%	44.44%	38.46%	3.57%	15.00%	0.00%
... wide tables (>10 attr.)	10.53%	-	26.67%	21.43%	0.00%	0.00%	13.33%
.. the entire data set	**17.05%**	**37.78%**	**51.61%**	**29.58%**	**10.94%**	**7.14%**	**28.13%**

Fig. 5 Percentage of dead tables over the population of tables within each schema size range.

		Pct over total #tables of:			Pct over total #tables of ...		Pct of **S-U** & **Survivor** over ...			Pct **S-D** & **Survivor** over ...		
	Total	NS	S-U	S-D	Surv.	Dead	...total	...S-UP	...Surv	...total	...S-D	...Surv
atlas	88	**69%**	25%	6%	83%	17%	25%	100%	30%	6%	100%	7%
biosql	45	**53%**	40%	7%	62%	38%	29%	72%	46%	2%	33%	4%
coppermine	23	**52%**	43%	4%	96%	4%	43%	100%	45%	4%	100%	5%
ensembl	155	**62%**	30%	8%	48%	52%	19%	64%	40%	5%	58%	9%
mwiki	71	**66%**	32%	1%	70%	30%	28%	87%	40%	0%	0%	0%
ocart*	128	**74%**	22%	4%	89%	11%	22%	100%	25%	4%	100%	4%
phpBB	70	**77%**	20%	3%	93%	7%	19%	93%	20%	1%	50%	2%
typo3	32	**50%**	41%	9%	72%	28%	34%	85%	48%	6%	67%	9%
	max	**77%**	**43%**	**9%**	96%	52%	43%	**100%**	**48%**	6%	100%	9%
	min	**50%**	**20%**	**1%**	48%	4%	19%	**64%**	**20%**	0%	0%	0%
	avg	**63%**	**32%**	**5%**	**77%**	**23%**	27%	**88%**	**37%**	4%	64%	5%
	stdev	9.7%	8.4%	2.5%	15.4%	15.4%	7.8%	12.7%	9.9%	2.0%	33.8%	3.1%

Fig. 6 Relationship of schema resize and survival as percentages.

Ensembl), we turn our interest to tables that are survivors and have a difference in their schema between their birth and the last known version of the database. The respective numbers are depicted in Fig. 6. A first observation is that *in their very large majority, tables that scale up are survivors*. The column of Fig. 6 labeled "...S-UP" assesses the percentage of survivors for the scaled up (middle of the figure) and the scaled down (right of the figure) categories. *Survivors constitute between 64% and 100% of scaled up tables, with an average of 88%*. If one poses the question in its inverse form, i.e., what percentage of survivors scale up, one can see that the average value is 37% with quite a few data sets reaching or exceeding 40%, but with *no data set exceeding 48% for the percentage of scaled up tables among the population of survivors*. Another 5% on average of the survivor tables are scaled down. Both percentages are practically in synch with the respective percentages of the overall population (i.e., being a survivor does not significantly alter a table's chances of experiencing a schema resize).

Overall, we can summarize by stating that *schema resizing is rare; the few tables that scale up, however, are survivors*.

4.2 Relationship of Year-Of-Birth to Survival

Is it possible that the version at which the table was born is related to its survival? This is a really unlikely pattern to pursue, as there is no particular evidence so far that birth affects the evolution of tables (of course, this remains to be quantified by a dedicated study). However, there is clear evidence that the database is altered in specific periods of maintenance, interrupting the otherwise calm life of the schema, including the starting period of the database where the activity is more intense [12, 15].

To this end, we quantify the number of births per year of life of the database. We count years starting from the first version of the database schema and not in terms of calendar years (thus resulting in measuring years as 1, 2, ...). We pay particular care to discriminate the first version of the database, which we treat as *Year 0*. As our inquiry is concerned with the possible differences between dead and survivor tables, we study each of these two classes separately. Fig. 7 depicts our results.

Fig. 7 Histograms of the distribution of (a) dead, vs., (b) survivor tables with respect to their year of birth

There are no significant differences between dead and survivor tables with respect to their year of birth. Small exceptions do exist (e.g., in the case of typo3) but the overall configuration clearly suggests that dead and survivor tables share the same percentage breakdown per year of birth.

5 Relationship of Duration and Activity to Survival

Already from [14, 15], we have observed 'anomalies' in the behavior of duration. When duration was related to the amount of updates undergone by each table, we observed a heavy concentration of top changers in top durations and a lack of medium sized number of updates in medium durations. This was summarized as the *inverse Gamma* pattern. A second related observation was demonstrated in the *empty triangle* pattern that related duration to birth. Again, instead of a uniform distribution of durations for dead tables, we observed the absence of dead tables at medium or high durations (which means they were removed shortly after their

	%dead	Prob. of death if a table is...			%survivors	Prob. of survival if a table is...		
	overall	Rigid	Quiet	Active	overall	Rigid	Quiet	Active
ensembl	52%	**79%**	36%	52%	48%	**21%**	64%	48%
biosql	38%	56%	46%	**13%**	62%	44%	54%	**88%**
mwiki	30%	**83%**	17%	40%	70%	**17%**	83%	60%
typo3	28%	42%	15%	29%	72%	58%	85%	71%
atlas	17%	**39%**	14%	7%	83%	**61%**	86%	93%
opencart*	11%	17%	5%	0%	89%	83%	95%	100%
phpbb	7%	0%	8%	**27%**	93%	100%	92%	**73%**
coppermine	4%	13%	0%	0%	96%	88%	100%	100%

Fig. 8 Probability of death (left) or survival (right) per activity class. The color code: blue (red) whenever the value of a cell is below (higher) 10% than the respective value for the entire data set; bold if the difference reaches or exceeds 20%. 0% and 100% are also colored anyway.

birth) and mostly towards early births. Still, these findings do not explain precisely how activity behavior, duration and survival are interrelated. In this section, we formally address the issue and quantitatively resolve it.

5.1 Relationship of Activity Behavior to Survival

Studying the activity behavior is not straightforward. Based on our investigations, we will focus our discussion on the *Activity Class* measure that classifies tables in three classes, specifically: (a) *rigid*, without any change throughout their lives, (b) *active*, if they have an *ATU* rate of more than 0.1 and more than 5 changes in their life, and (c) *quiet* otherwise.

5.1.1 Activity and Survival

To discover *how the activity behavior of a table is related to its survival*, we have computed the probability of death/survival for each of the three activity classes (rigid, quiet and active). To assess the essence of the research question, we contrast the respective probability to the overall probability of death /survival in the respective data set.

The results of our assessment are depicted in Fig. 8. The datasets are listed in decreasing order of probability of death for the entire data set (column right next to the data set name). We observe the following:

- *Rigid tables exhibit (frequently: significantly) higher chances of being removed compared to the rest of the tables*. Dead tables are typically removed shortly after their introduction, without any changes to their structure. So, not only rigid tables are a larger part of the dead population (see [14], specifically, its Fig. 5 for details) but also, rigid tables stand higher chances of being removed than the average table does. All the above observations are symmetrical for survivors: whereas there is higher chance than average for a rigid table to die, there is lower chance than average for a rigid table to survive.
- Quiet tables are the majority of the entire population in 6 out of 8 data sets, and the majority of the survivors. The difference between dead and survivors is

	DEAD					SURV.				
	#tables	Rigid	Quiet	Active	Total	#tables	Rigid	Quiet	Active	Total
coppermine	1	100%	0%	0%	100%	22	32%	59%	9%	100%
opencart*	14	79%	21%	0%	100%	114	47%	49%	4%	100%
typo3	9	56%	22%	22%	100%	23	30%	48%	22%	100%
mwiki	21	48%	43%	10%	100%	50	4%	90%	6%	100%
atlas	15	47%	40%	13%	100%	73	15%	51%	34%	100%
ensembl	80	46%	39%	15%	100%	75	13%	72%	15%	100%
biosql	17	53%	35%	12%	100%	28	25%	25%	50%	100%
phpBB	5	0%	40%	60%	100%	65	54%	34%	12%	100%

Fig. 9 Percentage of tables over their Survival Class (dead or survivor), each such class subdivided with respect to the Activity Class. All percentages refer to their Survival Class, thus the respective column of dead or survivor sums up to 100%.

not significant, albeit pretty much consistent. *When quiet tables are concerned, in 6 out of 8 data sets there is a decrease of the probability of dying (respectively: increase of the probability of surviving) compared to the average probability of dying of the data set and this difference exceeds 10% in 4 cases.*

– *Active tables have a mixed behavior*: in 4 cases, active tables are less likely to be removed than average. In 2 of them, the percentage of removals drops to 0% and in the other two, the decrease is 10% and 25% respectively. In one case, in the deletion-prone data set ensemble, there is no difference than the average of the entire table population. In three cases there is an increase in the probability of an active table being removed, with the increase reaching 10% for mwiki and 20% for phpBB (which is a peculiar data set with respect to deletions, with most tables being rigid and all its 5 deletions taking place at the same time, 2 of them being quiet and 3 of them being active). So, overall, the results for active tables are inconclusive, although a qualitative sense is that they tend to survive more than average.

5.1.2 Activity differences between Dead and Survivors

To assess whether *the activity behavior of dead tables is different than the respective behavior of survivors* we have computed the distribution of tables per Survival and Activity Class (Fig. 9). The percentages are with respect to the Survival Class (this is why the absolute value is also given for each of the two survival values and why each survival value sums up to 100%).

In Fig. 9, schemata are clustered by their predominant category. We observe 2 main clusters of schemata:

– The first cluster involves *schemata that are too rigid*, with too many rigid tables both in the *dead* class (where rigidity is more than 50% for all) and in the *survivor* class (with a percentage between 30% and 47%). Survivors also demonstrate a quiet character, with half of them been quiet, and very few of them being active.

– The second cluster includes data sets somewhat more active than the previous category. Again, the *dead* class has mainly rigid tables (half of the dead) and the *survivor* class has too many quiet tables.
– There are also two exceptions, with polarized behaviors: biosql has too many rigid dead and too many active survivors, and phpBB the inverse.

Grouping the above with respect to the *Survival Class* measure, we can summarize the situation as follows:

– Concerning dead tables, we observe a dominance of sudden deaths for rigid dead tables (top category in 7 out of 8 cases, with more than 50% of deaths in 4 of them and almost 50% in the 3 others). Quiet tables who die occupy a percentage ranging in 20%-40% and active tables range within 0%-22%, each case coming with a single exception.
– At the same time, survivors are mostly quiet (in 6 out of 8 data sets the dominating category is 'Quiet', ranging between 48%-90%), and the second place alternates between active and rigid tables, depending on the activity "profile" of the data set.

5.1.3 Statistical Significance

We have conducted both a chi-square and a Fischer test on the combination of *Activity* class and survival. We grouped tables with respect to (a) their *Activity* value (Rigid, Quiet and Active) and (b) their *Survival* value, i.e., we discriminated them between Dead and Survivors. Then we performed both a Chi-square test and a Fisher test on the dependency between the two categories. The hypotheses involved are:

H_0: the null hypothesis states that *Activity* and *Survival* are independent, i.e., there is no difference in the *Activity* profiles between dead and survivor tables

H_A: there is indeed a difference in the *Activity* profile of dead and survivor tables

The results are presented in Fig. 10.

The grouping of the schemata in Fig. 10 is consistent with the grouping of the aforementioned discussion. We observe two trends.

1. For the case of schemata with strong forces of rigidity and large percentages of rigid tables (here: Coppermine, Opencart and Typo3), rigidity is omnipresent in the lives of both dead and survivor tables, underplaying their statistical differences. In this case, the null hypothesis cannot be rejected, as the p-values are consistently above the typical significance level of α at 0.05.
2. On the other hand, the rest of the schemata, for both tests, have p-values lower than the aforementioned significance level, hence, we can accept the alternative hypothesis and suggest that *in the case where the schema is not full of rigid tables, dead and survivor tables are different with respect to their activity profile*.

5.2 Oppositely Skewed Durations

We have studied how the duration of tables is distributed in different duration ranges, thus creating a histogram of durations. We split the lifetime of each schema

P-Value for...	Chi-Square	Fisher
Coppermine	3.75E-01	4.35E-01
Ocart*	8.39E-02	1.14E-01
Typo3	3.44E-01	3.12E-01
Mediawiki	2.45E-05	1.23E-05
Atlas	1.71E-02	2.07E-02
Ensembl	2.00E-05	1.20E-05
Biosql	2.93E-02	2.80E-02
PhpBB	8.81E-03	8.92E-03

Fig. 10 Statistical evidence for the difference of dead and survivor tables with respect to their activity profile

in histogram buckets of 10 versions and counted the number of tables that pertain to each bucket. We have discriminated between dead and survivor tables, so we have a histogram for each of these two classes. Fig. 11 depicts the respective histograms (Ensembl and mediawiki are depicted with histogram buckets of 50 versions, which demonstrate the same behavior with buckets of 10 version, but, with the advantage of being fewer, and thus, presentable, as opposed to the alternative setting). We observe a phenomenon, which we call the *oppositely skewed durations* or *opposite skews* pattern.

5.2.1 The oppositely skewed durations pattern

When one constructs the histograms for the durations of dead vs survivor tables one can observe a symmetry in the histograms of the two classes. *The dead tables are strongly biased towards short durations (left-heavy), often with very large percentages of them being removed very shortly after birth. In quite the opposite manner, the survivor tables are mostly gathered at the other end of the spectrum (right-heavy), i.e., at high (frequently: max) durations.*

Exceptions to the pattern. Exceptions to the pattern do exist, albeit they do not significantly alter its validity. Coppermine's single deleted table was removed at 6 years of age. The phpBB database, which is otherwise too rigid, has 5 deleted tables that were removed at significantly larger durations than the typical in other data sets (in fact after 5 or 6 years of lifetime, all 5 being removed in the same version). The typo3 database, also has a set of 9 removed tables, again with quite high durations (7 of which had a lifetime between 4 and 8 years at the time of their removal).

Gravitation to rigidity. We attribute the tendency to short durations for the deleted tables to the cost that deletions have for the maintenance of the software that surrounds the database. The earlier a table is removed, the smaller the cost of maintaining the surrounding code is. Thus, when the table has been involved in

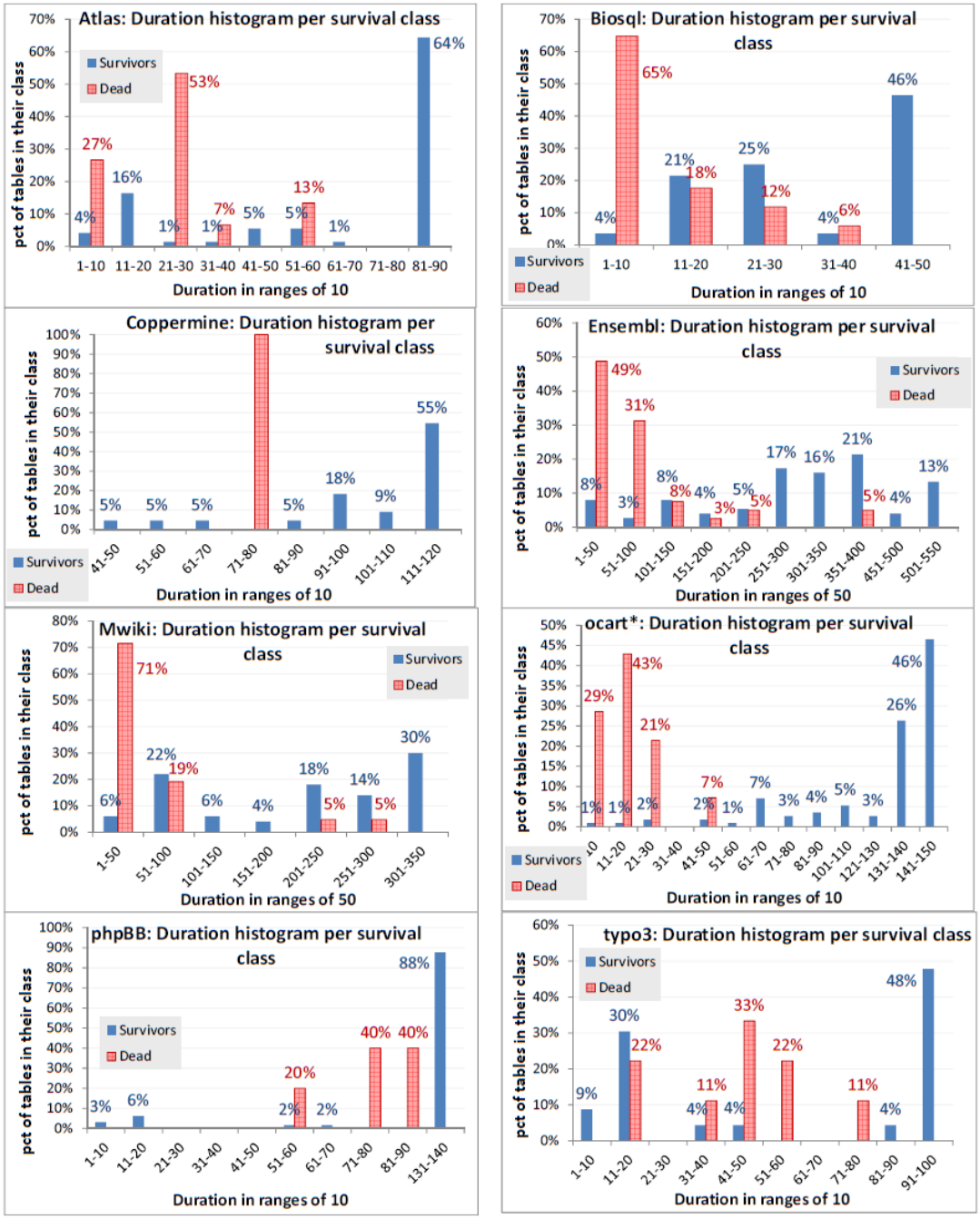

Fig. 11 Histograms of the durations of (a) dead, vs., (b) survivor tables.

several queries found in several places in the code, it is always a painstaking process

P-Value for...	Chi-Square	Fisher
Atlas	7.10E-10	1.76E-10
Biosql	8.57E-05	5.84E-06
Coppermine	1.70E-03	2.17E-01
Ensembl	2.44E-15	< 1e-07
Mediawiki	2.31E-06	4.13E-07
Ocart*	7.31E-15	6.39E-12
PhpBB	1.42E-11	3.22E-06
Typo3	1.14E-02	1.62E-03

Fig. 12 Statistical evidence for the difference of dead and survivor tables with respect to their duration profile

to locate, maintain and test the application code that uses it. At the same time, the reluctance for removals allows tables who survive the early stages to "remain safe". Thus, they grow in age without being removed. This fact, combined with the fact that the starting versions of the database already include a large percentage of the overall population of tables, results in a right-heavy, left-tailed distribution of survivor tables (for 6 out of 8 data sets, survivor durations reaching the final bucket of the respective histogram exceed 45%).

5.2.2 Statistical Significance

We have conducted both a chi-square and a Fischer test on the combination of *duration* and *survival*. We grouped tables with respect to (a) their duration histogram, and, (b) their survival value, i.e., we discriminated them between Dead and Survivors.. Then we performed both a Chi-square test and a Fisher test on the dependency between the two categories. The hypotheses involved are:

H_0: the null hypothesis states that *Duration* and *Survival* are independent, i.e., there is no difference in the *Duration* profiles between dead and survivor tables

H_A: there is indeed a difference in the *Duration* profile of dead and survivor tables

The results are presented in Fig. 12.

The Chi-square test show overwhelmingly that the null hypothesis cannot be supported. The reported values are several orders of magnitude lower than the typical significance level of 0.05. For the case of Ensembl, we could not get R to successfully complete with the exact p-value, although the estimation that R provides guaranteed that it is lower than 10^{-7}. There is a single exception, in the Fisher test of Coppermine, which is attributed to the fact the Coppermine comes with just one deleted table which results in a conservative estimation that what we observe has a 21% chance of being random. However, in all other cases, the evidence is strongly suggesting that *dead and survivor tables have significantly different duration profiles*

5.3 The electrolysis pattern

What happens if we relate duration with activity? The research question that is guiding us here is to discover whether there are patterns in the way survival, duration and activity behavior relate. Our analysis is a refinement of the oppositely skewed durations pattern with activity profile information. Whereas the opposite skews pattern simply reports percentages for duration ranges, here, we refine them by *LifeAndDeath* Class too. So, in the rest of this subsection, we will group the tables according to the *LifeAndDeath* class, which expresses the profile of a table with respect to the combination of *survival* x *activity*, practically composing the two domains {*dead*, *survivor*} × {*rigid*, *quiet*, *active*} into their Cartesian product. Then, for each of the resulting six classes, we study the durations of the tables that belong to it.

5.3.1 The essence of the pattern

We formulate our observations as a new pattern, which we call the electrolysis pattern. Remember than in an electrolysis experimental device, two electrodes are inserted in water: a negative electrode, or cathode and a positive electrode, or anode. Then, negatively charged ions move towards the positive anode and positively charged ions move towards the negative cathode.

A somewhat similar phenomenon occurs for dead and survivor tables concerning the combination of duration and survival, which we call the *electrolysis pattern*.

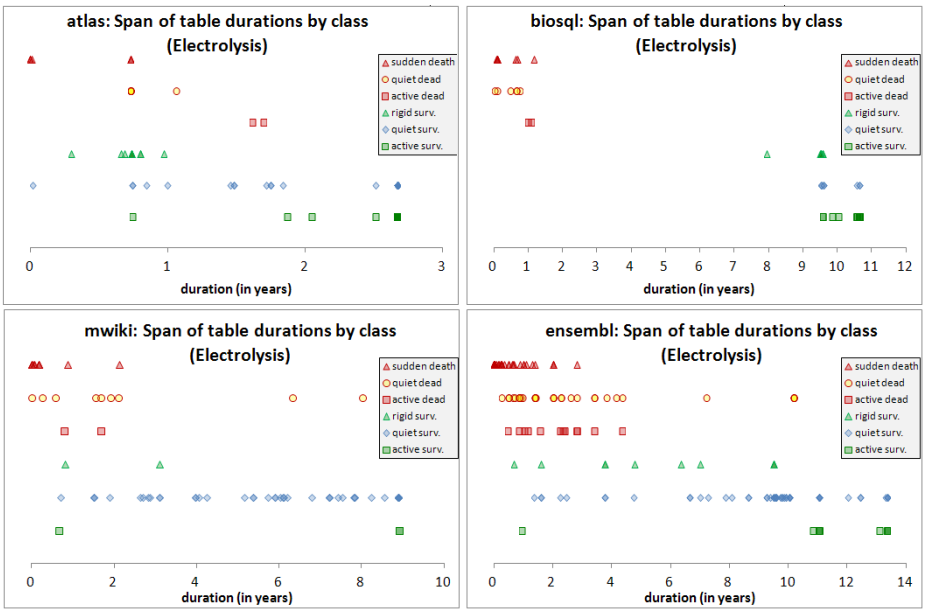

Fig. 13 The Electrolysis pattern. Each point refers to a table with (a) its duration at the x-axis and (b) its LifeAndDeath class (including both survival and activity) at the y-axis (also its symbol). Points are semi-transparent: intense color signifies large concentration of overlapping points.

In Fig. 13, we graphically depict the phenomenon via scatter-plots that demonstrate the *LifeAndDeath* × *Duration* space for all the studied data sets.

Electrolysis pattern: Dead tables demonstrate much shorter lifetimes than survivor ones and can be located at short or medium durations, and practically never at high durations. With few exceptions, the less active dead tables are the higher the chance to reach shorter durations. Survivors expose the inverse behavior i.e., mostly located at medium or high durations. The more active survivors are, the stronger they are attracted towards high durations, with a significant such inclination for the few active ones that cluster in very high durations.

Fig. 13 vividly reveals the pattern's highlights. Observe:

- The total absence of dead tables from high durations.
- The clustering of rigid dead at low durations, the spread of quiet dead tables to low or medium durations, and the occasional presence of the few active dead, that are found also at low or medium durations, but in a clustered way.
- The extreme clustering of active survivors to high durations.
- The wider spread of the (quite numerous) quiet survivors to a large span of durations with long trails of points.
- The spread of rigid survivors, albeit not just to one, but to all kinds of durations (frequently, not as high as quiet and active survivors).

One could possibly argue that the observed clusterings and time spans are simply a matter of numbers: the more populated a class is, the broader its span is. To forestall any such criticism, this is simply not the case. We give the respective numbers in the sequel of this subsection; here, we proactively mention a few examples to address this concern. Rigid dead tables are the most populated group in the dead class, yet they have the shortest span of all. The rigid survivors, who are the 2^{nd} most populated class of the entire population, exhibit all kinds of behaviors; yet, in most of the cases, they are disproportionally clustered and not spread throughout the different categories. Active survivors are also disproportionately clustered at high durations. Overall, with the exception of the quiet survivors that indeed span a large variance of durations, in the rest of the categories, the time span is disproportionate to the size of the population (number of points in the scatter plot) of the respective class.

5.3.2 In-depth study of durations

To understand how tables are distributed in different durations, we have expressed table durations as percentages over the lifetime of their schema. Then, for each *LifeAndDeath* value and for each duration range of 5% of the database lifetime, we computed the percentage of tables whose duration falls within this range.

As we have performed this computation per data set, and in order to come up with a single view for all the eight data sets, we have resorted in averaging the respective percentages of the eight data sets[3]. The data for this averaged breakdown are presented in Fig. 14.

[3] An acute reader might express the concern whether it would be better to gather all the tables in one single set and average over them. We disagree: each data set comes with its own the requirements, development style and idiosyncrasy and putting all tables in a single data set, not only scandalously favors large data sets, but integrates different things. We average the behavior of schemata, not tables here.

	0% 5%	5% 10%	10% 15%	15% 20%	20% 25%	25% 30%	30% 35%	35% 40%	40% 45%	45% 50%	50% 55%	55% 60%	60% 65%	65% 70%	70% 75%	75% 80%	80% 85%	85% 90%	90% 95%	95% 100%
DR	**6.0%**	**1.3%**	0.4%	0.6%	0.3%	0.8%	0.0%	0.0%	0.7%	0.8%	0.0%	0.9%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
DQ	**1.6%**	**1.6%**	0.4%	**1.2%**	0.4%	**1.0%**	0.2%	0.1%	0.0%	0.4%	0.1%	0.0%	0.0%	0.0%	0.2%	0.5%	0.0%	0.2%	0.2%	0.0%
DA	0.1%	0.7%	0.4%	0.4%	0.2%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	0.4%	0.3%	0.0%	0.0%	0.0%	0.4%	0.5%	0.0%	0.0%
SR	**1.7%**	0.4%	**1.5%**	0.0%	0.6%	**1.5%**	0.2%	0.2%	0.0%	0.1%	**1.9%**	0.0%	0.0%	0.0%	0.5%	0.0%	0.7%	**1.9%**	**2.8%**	**8.6%**
SQ	**1.2%**	0.5%	**1.0%**	0.8%	0.5%	**1.2%**	**1.0%**	0.6%	**1.3%**	0.5%	**2.2%**	0.5%	**1.5%**	**1.6%**	**1.4%**	**1.0%**	**2.0%**	**2.2%**	**1.5%**	**17.8%**
SA	0.4%	0.7%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.1%	0.5%	0.0%	**1.3%**	**10.4%**

Fig. 14 Average participation percentage per 5% duration range and LifeAndDeath value

In order to summarize the detailed data of Fig. 14, in Fig. 17 we regrouped the data in just 3 duration ranges: (a) durations lower that the 20% of the database lifetime (that attracts a large number of dead tables, esp., the rigid ones), to which we will refer as *low durations*, in the sequel, (b) durations higher that the 80% of the database lifetime (where too many survivors, esp., active ones are found), to which we will refer as *high durations*, and finally, (c) the rest of the durations in between, forming an intermediate category of *medium* durations.

To visually aid the reader, in Fig. 15 (top) we present a heat map for the data of Fig. 14. The intensity of each color signifies how large the respective percentage is in Fig. 14. Moreover, to avoid overemphasizing single occurrences that might expose a large percentage, we have nullified all cells that correspond to only one data set. The respective heat map (which, we believe, more accurately shows the actual situation), is depicted in the bottom of Fig. 15.

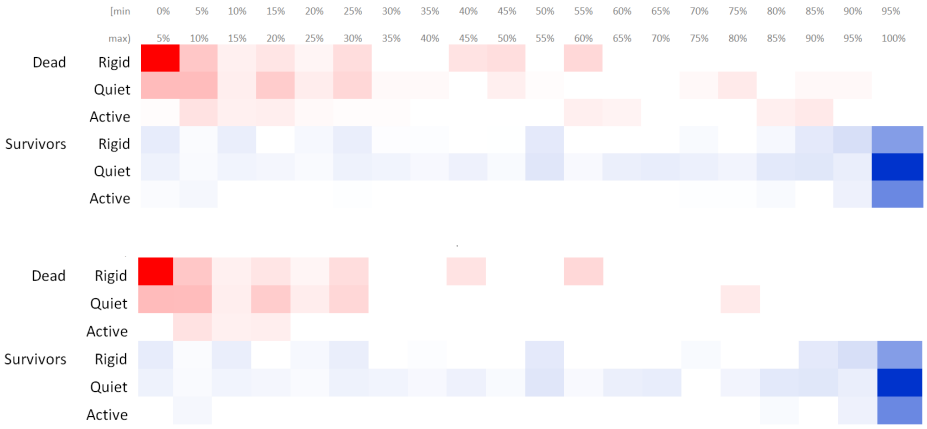

Fig. 15 Electrolysis-all-in-one. Heat-map for the participation of each LifeAndDeath value in a certain duration range, averaged for all 8 data sets (top: pure; bottom: with outliers removed)

5.3.3 Studying the critical low and top 20% of durations

The above summaries demark fairly accurately the electrolysis pattern and the mutually "repulsing" behaviors of dead and survivor classes. Still, as these representations are presenting an averaged representation, important details are hidden within the respective percentages and heat maps. To reveal these subtleties, we

Atlas	Rigid dead	Quiet Dead	Active Dead	Rigid Surv	Quiet Surv	Active Surv	
[0% -20%)	5%	0%	0%	1%	1%	0%	7%
[20%-80%)	3%	7%	2%	11%	13%	3%	40%
[80%-100%]	0%	0%	0%	0%	28%	25%	53%
	8%	7%	2%	13%	42%	28%	100%

Biosql	Rigid dead	Quiet Dead	Active Dead	Rigid Surv	Quiet Surv	Active Surv	
[0-20%)	20%	13%	4%	0%	0%	0%	38%
[20%-80)	0%	0%	0%	2%	0%	0%	2%
[80%-100%]	0%	0%	0%	13%	16%	31%	60%
	20%	13%	4%	16%	16%	31%	100%

Copperm.	Rigid dead	Quiet Dead	Active Dead	Rigid Surv	Quiet Surv	Active Surv	
[0-20%)	0%	0%	0%	0%	0%	0%	0%
[20%-80)	4%	0%	0%	0%	13%	0%	17%
[80%-100%]	0%	0%	0%	30%	43%	9%	83%
	4%	0%	0%	30%	57%	9%	100%

Ensembl	Rigid dead	Quiet Dead	Active Dead	Rigid Surv	Quiet Surv	Active Surv	
[0-20%)	23%	13%	5%	1%	3%	1%	37%
[20%-80)	1%	7%	3%	5%	23%	0%	54%
[80%-100%]	0%	0%	0%	0%	9%	6%	8%
	24%	20%	8%	6%	35%	7%	100%

Mwiki	Rigid dead	Quiet Dead	Active Dead	Rigid Surv	Quiet Surv	Active Surv	
[0-20%)	13%	7%	3%	1%	6%	1%	23%
[20%-80)	1%	4%	0%	1%	31%	0%	58%
[80%-100%]	0%	1%	0%	0%	27%	3%	20%
	14%	13%	3%	3%	63%	4%	100%

Ocart*	Rigid dead	Quiet Dead	Active Dead	Rigid Surv	Quiet Surv	Active Surv	
[0-20%)	3%	2%	0%	8%	5%	1%	23%
[20%-80)	5%	0%	0%	17%	16%	0%	43%
[80%-100%]	0%	0%	0%	17%	22%	2%	34%
	9%	2%	0%	42%	44%	3%	100%

phpBB	Rigid dead	Quiet Dead	Active Dead	Rigid Surv	Quiet Surv	Active Surv	
[0-20%)	0%	0%	0%	1%	3%	3%	7%
[20%-80)	0%	1%	0%	0%	4%	0%	11%
[80%-100%]	0%	1%	4%	49%	24%	9%	81%
	0%	3%	4%	50%	31%	11%	100%

typo3	Rigid dead	Quiet Dead	Active Dead	Rigid Surv	Quiet Surv	Active Surv	
[0-20%)	3%	3%	0%	16%	9%	3%	34%
[20%-80)	13%	3%	3%	3%	6%	0%	31%
[80%-100%]	0%	0%	3%	3%	19%	13%	34%
	16%	6%	6%	22%	34%	16%	100%

Fig. 16 For each data set, for each LifeAndDeath class, percentage of tables per duration range over the total of the data set (for each data set, the sum of all cells adds up to 100%).

	Rigid Dead	Quiet Dead	Active Dead	Rigid Surv	Quiet Surv	Active Surv	
[0-20%)	8%	5%	2%	4%	3%	1%	23%
[20%-80%)	3%	3%	1%	5%	13%	0%	26%
[80%-100%]	0%	0%	1%	14%	24%	12%	51%
	12%	8%	3%	23%	40%	14%	100%

Fig. 17 Indicative, average values over all datasets: for each LifeAndDeath class, percentage of tables per duration range over the total of the entire data set.

perform a targeted study of the two duration ranges of concern. Based on the analysis already performed, we regrouped the data in just 3 duration ranges: (a) durations lower that the 20% of the database lifetime (that attracts a large number of dead tables, esp., the rigid ones), to which we will refer as *low durations*, in the sequel, (b) durations higher that the 80% of the database lifetime (where too many survivors, esp., active ones are found), to which we will refer as *high durations*, and finally, (c) the rest of the durations in between, forming an intermediate category of *medium* durations.

The histograms of all the data sets for the distribution of tables to the space *Duration* × *LifeAndDeath Class* is depicted in Fig. 16 and their summary is shown in Fig. 17. The lattest figure reports on the average value over the eight data sets that we have studied. In other words, for each of the reported cells, we have averaged the respective cells of the eight data sets.

5.3.4 Breakdown per LifeAndDeath *class*

Another research question concerns the breakdown of the distribution of tables within each *LifeAndDeath* class. In other words, we ask: *do certain LifeAndDeath classes have high concentrations in particular duration ranges*?

If one wants to measure the percentage of tables for each value of the *LifeAndDeath Class* over each duration range, we need to calculate each cell as the percentage of the specific class (e.g., each cell in the rigid dead column should

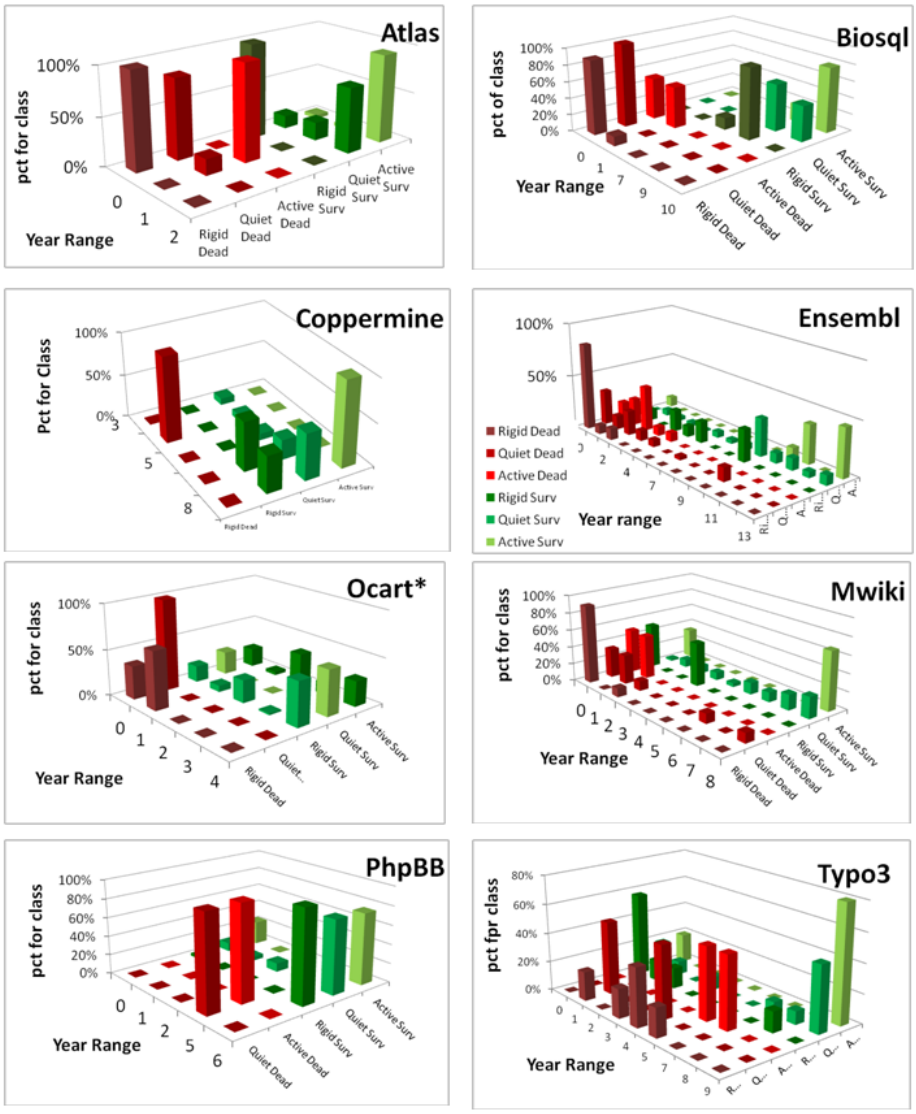

Fig. 18 For each data set, for each LifeAndDeath class, a 3D visualization of percentage of tables per duration range over the total of the LifeAndDeath class.

measure the fraction of rigid dead tables that belong to its particular duration range). The detailed data per data set, where the value of each cell is presented as percentage over its *LifeAndDeath* class are depicted in Fig. 18. Fig. 18 is probably the modst vivid depiction of the electrolysis pattern. For the readers interested in the numerical values, it is best to resort to the 20%-80%-100% tabular presentation format of Fig. 19. If we average the percentages for the respective cells of all the data sets, we end up with the distribution of values depicted in Fig. 20.

Atlas	Rigid dead	Quiet Dead	Active Dead	Rigid Surv	Quiet Surv	Active Surv
[0% -20%)	57%	0%	0%	9%	3%	0%
[20%-80%)	43%	100%	100%	91%	30%	12%
[80%-100%]	0%	0%	0%	0%	68%	88%
	100%	100%	100%	100%	100%	100%

Biosql	Rigid dead	Quiet Dead	Active Dead	Rigid Surv	Quiet Surv	Active Surv
[0-20%)	100%	100%	100%	0%	0%	0%
[20%-80%)	0%	0%	0%	14%	0%	0%
[80%-100%]	0%	0%	0%	86%	100%	100%
	100%	100%	100%	100%	100%	100%

Copperm.	Rigid dead	Quiet Dead	Active Dead	Rigid Surv	Quiet Surv	Active Surv
[0-20%)	0%			0%	0%	0%
[20%-80%)	100%			0%	23%	0%
[80%-100%]	0%			100%	77%	100%
	100%			100%	100%	100%

Ensembl	Rigid dead	Quiet Dead	Active Dead	Rigid Surv	Quiet Surv	Active Surv
[0-20%)	97%	65%	67%	20%	9%	9%
[20%-80%)	3%	35%	33%	80%	65%	0%
[80%-100%]	0%	0%	0%	0%	26%	91%
	100%	100%	100%	100%	100%	100%

Mwiki	Rigid dead	Quiet Dead	Active Dead	Rigid Surv	Quiet Surv	Active Surv
[0-20%)	90%	56%	100%	50%	9%	33%
[20%-80%)	10%	33%	0%	50%	49%	0%
[80%-100%]	0%	11%	0%	0%	42%	67%
	100%	100%	100%	100%	100%	100%

Ocart*	Rigid dead	Quiet Dead	Active Dead	Rigid Surv	Quiet Surv	Active Surv
[0-20%)	36%	100%		19%	13%	25%
[20%-80%)	64%	0%		41%	38%	0%
[80%-100%]	0%	0%		41%	50%	75%
	100%	100%		100%	100%	100%

phpBB	Rigid dead	Quiet Dead	Active Dead	Rigid Surv	Quiet Surv	Active Surv
[0-20%)		0%	0%	3%	9%	25%
[20%-80%)		50%	0%	0%	14%	0%
[80%-100%]		50%	100%	97%	77%	75%
		100%	100%	100%	100%	100%

typo3	Rigid dead	Quiet Dead	Active Dead	Rigid Surv	Quiet Surv	Active Surv
[0-20%)	20%	50%	0%	71%	27%	20%
[20%-80%)	80%	50%	50%	14%	18%	0%
[80%-100%]	0%	0%	50%	14%	55%	80%
	100%	100%	100%	100%	100%	100%

Fig. 19 For each data set, for each LifeAndDeath class, percentage of tables per duration range over the total of the LifeAndDeath class (for each data set, for each column, percentages add up to 100%).

	Rigid dead	Quiet Dead	Active Deac	Rigid Surv	Quiet Surv	Active Surv
[0-20%)	57%	53%	44%	21%	9%	14%
[20%-80)	43%	38%	31%	36%	29%	2%
[80%-100%]	0%	9%	25%	42%	62%	84%
	100%	100%	100%	100%	100%	100%

Fig. 20 Average values over all datasets: for each LifeAndDeath class, percentage of tables per duration range over the total of the LifeAndDeath class (for each data set, for each column, percentages add up to 100%).

Finally, in Fig. 21 we zoom only in (a) dead tables at the lowest 20% and (b) survivors at the highest 20% of durations. We count the number of tables, per *LifeAndDeath* class, for the respective critical duration range, and we compute the fraction of this value over the total number of tables pertaining to this *LifeAndDeath* class (columns *Rigid*, *Quiet*, *Active*). For the *Dead* and *Surv* columns, we divide the total number of dead/survivor tables belonging to the respective critical duration over the total number of dead/survivor tables overall.

We observe that in more than half of the cells of the table in Fig. 21, the percentage reaches or exceeds 50%. This clearly demarcates the high concentrations of dead tables in low durations and of survivor tables in high durations.

5.3.5 Statistical Significance

We have conducted both a chi-square and a Fischer test on the combination of *LifeAndDeath* class and survival. We grouped tables with respect to their combination of (a) *LifeAndDeath* class (i.e., the respective six values of the domain of *LifeAndDeath* class) and (b) in three intervals with respect to their duration, specifically, those belonging to [0-20%), [20%-80%), and [80%-100%] of the data

	Pct of durations shorter than 20% of db life for Dead tables over the …				*Pct of durations longer than 80% of db life for Survivor tables over the …*			
	… Dead	… Rigid	… Quiet	…Active	… Surv	… Rigid	… Quiet	…Active
atlas	27%	57%	0%	0%	64%	0%	68%	**88%**
biosql	**100%**	**100%**	**100%**	**100%**	**96%**	**86%**	**100%**	**100%**
coppermine	0%	0%	-	-	**86%**	**100%**	**77%**	**100%**
ensembl	**80%**	**97%**	65%	67%	32%	0%	26%	**91%**
mediawiki	**76%**	**90%**	56%	**100%**	42%	0%	42%	67%
opencart*	50%	36%	**100%**	-	46%	41%	50%	**75%**
phpBB	0%	-	0%	0%	**88%**	**97%**	**77%**	**75%**
typo3	22%	20%	50%	0%	48%	14%	55%	**80%**

Fig. 21 Percentages of dead tables with too short durations and survivor tables with too long durations (red: above 50%, bold: above 75%, blue: below 20%, dash: no such tables).

P-value for…	*Chi-square*	*Fischer*
Atlas	3.61E-11	1.04E-12
Biosql	2.67E-07	1.04E-12
Coppermine	NaN	1.46E-01
Ensembl	6.76E-24	2.96E-08
Mediawiki	4.75E-05	3.95E-07
Ocart*	NaN	9.15E-04
PhpBB	NaN	3.31E-02
Typo3	3.90E-02	2.77E-02

Fig. 22 Statistical evidence for the difference of different *LifeAndDeath* values with respect to their duration profile

sets lifetime. Then we performed both a Chi-square test and a Fisher test on the dependency between the two categories. The hypotheses involved are:

H_0: the null hypothesis states that *Duration* and *LifeAndDeath* profile are independent, i.e., there is no difference in the *LifeAndDeath* profile between the durations of the different categories

H_A: there is indeed a difference in the *LifeAndDeath* profile of the three duration categories

The results are presented in Fig. 22.

The results are overwhelmingly in favor of the rejection of the null hypothesis and the adoption of the alternative one stating that LifeAndDeath profiles present different different durations. Almost all p-values are several orders of magnitude below what would be an acceptable 0.05 significance level α The only exception is Coppermine, with its single deleted table over its 23 tables that appeared in its lifetime. We believe that this idiosyncrasy of the data set results in the 14% probability of observing these results by chance. The NaN results is due to *LifeAndDeath* values having all zeros.

5.3.6 Observations and Findings

Now, we are ready to quantitatively support the wording of the electrolysis pattern. We organize our discussion by *LifeAndDeath* class. Our quantitative findings for the electrolysis pattern are delineated in the rest of this subsection.

Dead tables. We already knew from [15] that almost half the dead tables are rigid. Here, we have a clear testimony, however, that *not only are dead tables inclined to rigidity, but they are also strongly attracted to small durations*. The less active tables are the more they are attracted to short durations. *The attraction of dead tables, especially rigid ones, to (primarily) low or (secondarily) medium durations is significant and only few tables in the class of dead tables escape this rule*. Interestingly, in all our datasets, the only dead tables that escape the barrier of low and medium durations are a single table in mediawiki, another one in typo3 and the 4 of the 5 tables that are simultaneously deleted in phpBB.

- *Rigid dead tables, which is the most populated category of dead tables, strongly cluster in the area of low durations (lower than the 20% of the database lifetime) with percentages of 90% – 100% in 3 of the 6 data sets* (Fig. 21). Atlas follows with a large percentage of 57% in this range. Two exceptions exist: opencart and typo3, having most of their dead tables in the medium range. There are also two exceptions of minor importance: coppermine with a single deleted table and phpBB with a focused deletion of 5 tables at a single time point.
- *Quiet dead tables, which is a category including few tables, are mostly oriented towards low durations*. Specifically, there are 5 data sets with a high concentration of tables in the area of low durations (Fig. 21); for the rest the majority of quiet dead tables lie elsewhere: atlas has 100% in the medium range and phpBB is split in half between medium and large durations.
- Finally, for the very few active dead, which is a category where only six of the eight data sets have even a single table, there are two of them with 100% concentration and another one in 67% of its population in the low durations (Fig. 21). For the rest, atlas has 100% of its active dead in the medium range, phpBB 100% of the active dead in the long range (remember that phpBB has an exceptional behavior) and typo3 is split in half between low and medium durations (Fig. 19).

Survivors. *Survivors have the opposite tendency of clustering compared to the dead ones*. So, there are quite a few cases where survivor tables reach very high concentrations in high durations, and, interestingly, the more active the tables are, the higher their clustering in high durations.

- *Rigid survivors demonstrate a large variety of behaviors*. Rigid survivors are the second most populated category of tables after quiet survivors and demonstrate too many profiles of clustering (Fig. 19): one data set comes with a low-heavy profile, another 3 with a high-heavy profile, another two with a medium-heavy profile, and there is one data set split in half between early and medium durations and another one with an orientation of medium-to-high durations.
- *Quiet survivors, being the (sometimes vast) majority of survivor tables, are mostly gravitated towards large durations, and secondarily to medium ones*. In 6 out of 8 data sets, the percentage of quiet survivors that exceed 80% of db lifetime surpasses 50% (Fig. 19). In the two exceptions, medium durations is the largest

subgroup of quiet survivors. Still, quiet survivors also demonstrate short durations too (Fig. 19), so overall, their span of possible durations is large. Notably, in all data sets, there are quiet survivors reaching maximum duration.
– It is extremely surprising that the vast majority of active survivors exceed 80% of the database lifetime in all datasets (Fig. 21). With the exception of three data sets in the range of 67%-75%, *the percentage of active survivors that exceed 80% of the db lifetime exceeds 80% and even attains totality in 2 cases*. Active survivor tables are not too many; however, it is their clustering to high durations (implying early birth) that is amazing. If one looks into the detailed data and in synch with the empty triangle pattern of [15], *the top changers are very often of maximum duration, i.e., early born and survivors* (Fig. 13).

Absence of evolution. Although the majority of survivor tables are in the quiet class, we can quite emphatically say that *it is the absence of evolution that dominates*. Survivors vastly outnumber removed tables. Similarly, rigid tables outnumber the active ones, both in the survival and, in particular, in the dead class. Schema size is rarely resized, and only in survivors. Active tables are few and are mainly born early phases of the database lifetime.

6 Discussion, Take up and Future Work

Carl Sagan is quoted to have said: "extinction is the rule, survival is the exception". This is definitely not the case for tables in evolving open source software. Evidently, not only survival is stronger than removal, but rigidity is also stronger a force than variability and the combination of these two forces further lowers the amount of change in the life of a database schema.

In the rest of this section, we summarize our findings, concerns on their validity, our opinion on the mechanics behind them and, at the same time, expand the discussion on the usefulness of our approach and roads for future research.

6.1 Summary of findings

In a nutshell, our findings are as follows:

1. Concerning *schema size at their end*: dead tables are mostly thin (in accordance to the general pattern of the entire data set), and, in all occasions, the probability of a wide table being removed is quite lower than the average removal probability of the data set.
2. Concerning *schema resizing*: Schema resizing is infrequent: almost 2 out of 3 tables do not have a modification of their schema size at the end of their observation. As most of the dead tables are rigid, i.e., they die without going through any modifications, the tables with observed resizings are mostly survivors. On average, 37% of survivors go through a scale up and 5% through a scale down of the schema size, meaning that the majority of survivors is also without schema resizing. At the same time, tables that scale up are mostly survivors: specifically, survivors constitute between 64% and 100% of scaled up tables.

3. Concerning *activity profile*: rigid tables exhibit (frequently: significantly) higher chances of being removed compared to the rest of the tables; quiet tables, on the other hand, have a slightly decreased probability of dying, compared to the average probability.
4. Concerning *durations*, we observe the *oppositely skewed durations pattern*: The dead tables are strongly biased towards short durations, often with very large percentages of them being removed very shortly after birth. In quite the opposite manner, the survivor tables are mostly heavy-tailed at the other end of the spectrum, i.e., at high (frequently: max) durations.
5. When the *activity profile is combined with the duration*, we observe the *electrolysis pattern*: Dead tables demonstrate much shorter lifetimes than survivor ones and can be located at short or medium durations, and practically never at high durations. With few exceptions, the less active dead tables are, the higher the chance to reach shorter durations. Survivors expose the inverse behavior i.e., mostly located at medium or high durations. The more active survivors are, the stronger they are attracted towards high durations, with a significant such inclination for the few active ones that cluster in very high durations.

6.2 Threats to validity

It is always necessary to approach one's study with a critical eye for its validity. With respect to the *measurement validity* of our work, we have tested (i) our automatic extraction tool, Hecate, for the accuracy of its automatic extraction of delta's and measures, and (ii) our human-made calculations. With respect to the *scope* of the study, as already mentioned, we frame our investigation to schemata that belong to open-source projects. This has to do with the decentralized nature of the development process in an open source environment. Databases in closed organizational environments have different administration protocols, their surrounding applications are possibly developed under strict software house regulations and also suffer from the inertia that their evolution might incur, due to the need to migrate large data volumes. So, we warn the reader not to overgeneralize our results to this area. Another warning to the reader is that we have worked only with changes at the logical and not the physical layer. Having said that, we should mention, however, that the *external validity* of our study is supported by several strong statements: we have chosen data sets with (a) fairly long histories of versions, (b) a variety of domains (CMS's and scientific systems), (c) a variety in the number of their commits (from 46 to 528), and, (d) a variety of schema sizes (from 23 to 114 at the end of the study); kindly refer to Fig. 3 for all these properties. We have also been steadily attentive to work only with phenomena that are common to all the data sets. We warn the reader not to interpret our findings as laws (that would need confirmation of our results by other research groups), but rather as patterns. Favorably, some very recent anecdotal evidence, in fact coming from the industrial world, is corroborating in favor of our gravitation to rigidity theory (see the blog entry by Stonebraker et al., at `http://cacm.acm.org/blogs/blog-cacm/208958-database-decay-and-what-to-do-about-it/fulltext`). Based on the above, we are confident for the validity of our findings within the aforementioned scope.

6.3 Why do we see what we see

We believe that this study strengthens our theory that schema evolution antagonizes a powerful gravitation to rigidity. The "dependency magnet" nature of databases, where all the application code relies on them but not vice versa, leads to this phenomenon, as avoiding the adaptation and maintenance of application code is a strong driver towards avoiding the frequent evolution of the database. Some explanations around the individual phenomena that we have observed can be attributed to the gravitation to rigidity:

- Survival of wide tables can be attributed to the fact, that the wider a table is, the more it is relied upon (both due to the number of involved attributes and the fact that it is probably a information carrying table).
- Scale up and down of a schema is also affected by the gravitation to rigidity. Tables mostly live their lives without schema resizing, esp., the dead ones, which means that developers are not particularly enthusiastic towards altering the schema of a table. For the rare tables who actually go through resizing, most of them are survivors (i.e., the investment of effort seems to keep them alive).
- Dead tables die shortly after their birth and quite often, rigid: this setting provides as little as possible exposure to application development for tables to be removed.
- As dead tables do not attain high durations, it appears that after a certain period, practically within 10%-20% of the databases' lifetime, tables begin to be "safe". The significant amount of tables that stand the chance to attain maximum durations can be explained if we combine this observation with the fact that large percentages of tables are created at the first version of the database.
- Rigid tables find it hard to attain high durations (unless found in an environment of low change activity). This difficulty can be explained by two reasons. First, shortly after they are born, rigid tables are in the high-risk group of being removed. Second, rigid tables are also a class of tables with the highest migration probability. Even if their duration surpasses the critical 10% of databases lifetime where the mass of the deleted tables lies, they are candidates for being updated and migrating to the quiet class.
- Tables with high durations (i.e., early born) that survive spend their lives mostly quietly (i.e., with the few occasional maintenance changes) – again minimizing the impact to the surrounding code.
- The high concentration of the few active tables to very high durations and survival (which is of course related to early births) is also related to the gravitation to rigidity: the early phases of the database lifetime typically include more table births and, at the same time, gravitation to rigidity says that after the development of a substantial amount of code, too high rate of updates becomes harder; this results in very low numbers of active tables being born later. So, the pattern should not be read so much as "active tables are born early", but rather as "we do not see so many active tables being born in late phases of the database life".

6.4 Practical Recommendations and elaboration on the Usefulness of this research

What have we gained from this paper, at the end of the day? What do we gain from this kind of research? We keep encountering these questions time and again, and we find it appropriate to devote some space to clarify our viewpoint on them.

The entire paper so far has been a contribution to expansion of our knowledge on the mechanics of schema evolution. Thus, its *usefulness from the standpoint of knowledge expansion* has been already demonstrated (and summarized in Sections 6.1 and 6.2). In this subsection, we expand the discussion on both the immediate practical take-away's of our findings and the general benefits from a scientific point of view.

6.4.1 Usefulness from the practical standpoint: recommendations for developers

Assume you are a developer of an application built on top of, and depending upon, an evolving data base. Equivalently, assume you have deployed a Free Open Source system, like the ones discussed in this paper, that contains a database, and you are expecting its next release. Knowing which tables are likely to change or die, or even deeper, knowing how is a table going to live its life, is useful knowledge on how to schedule application development, determine the exposure to the underlying database, organize the usage of views as wrappers for the schema elements that can be renamed, merged, split, or relocated, as well as, for planning database administration and migration tasks.

We have established that the thinner a table is, the higher the chances of being removed; oppositely, the few wide tables, with schema size larger than 10 attributes are survivors in their vast majority, with lower probabilities of being removed. Tables typically die shortly after their birth and quite often, rigid, i.e., without having experienced any update before. So, young rigid tables are the high risk group for being removed.

Typically, if a table surpasses infant mortality, it will likely survive to live a rigid or, more commonly, a quiet live. Schema scale up or down are infrequent, with one in three survivors going through a scale up in its number of attributes and one in twenty through a scale down. There is a small group of active tables, going through significant updates. Look for them in the early born survivors, as later phases of the database life do not seem to generate tables that are too active.

Overall, after a table is born, the development of code that depends on it should be kept as restrained as possible – preferably encapsulated via views that will hide the changes from the application code. After the period of infant mortality, it is fairly safe to say that unless the table shows signs of significant update activity, gravitation to rigidity enters the stage and the table's evolution will be low.

6.4.2 Usefulness from the scientific standpoint

Computer scientists, as a hybrid of scientists and engineers have been building models on the one hand, and, systems and applications based on them, on the other. As the related work section has demonstrated, the issue of studying how our artefacts have been used (or maybe misused), has largely been neglected. We use the relational model and build databases; we use embedded SQL or ORM's to

link databases to applications built on top of them; we evolve both the databases and the applications – however, we, as a research community, do not seem to take the time to look back and assess whether this usage and activity is really done properly and in an orderly fashion.

So, why have we embarked on this research program on extracting patterns of evolution and what do our efforts tell us? On the one hand, we have only started understanding the patterns of evolution and, if the research community continues, we might be lucky enough to discover more. At the same time, there is a serious concern uncovered, and it is gravitation to rigidity. The essence of the relational model usage has been that the schema of the data determine their semantics; thus table and attribute names are the source of semantics for the stored data. As application code and its embedded queries (in any form they might come) depend on these names, schema evolution affects both the integrity of the code, which can crash, but also the semantics of the information stored or delivered to the end user. Thus, changes necessarily require the maintenance of the code, and this is a force hindering evolution.

So, one of the concerns raised by our research program is that it is quite possible we need to devise new ways of relating applications to databases, minimizing the impact of change and the gravitation to rigidity. Replacing the relational model with new options is one of the possible ways; using ontologies on top of it and relating application queries to these ontologies is another possibility. The more we learn on the maintenance patterns of tables and schemata, however, the easier it will be for us to focus our attention to what developers face and how to handle it.

6.5 Future work

In the context of this study we have seen that birth does not really affect survival. Still, why do active survivors cluster in early births is still unknown. Related literature suggests that database evolution cools down after the first versions. This has been studied for the slowdown of the birth rate, however a precise, deep investigation of the timing of the heartbeat of the database schema with all its births, deaths and updates is still pending. To our view, this practically marks the limits of analyses based on descriptive statistics. The next challenge for the research community lies in going all the way down to the posted comments and the expressed user requirements at the public repositories and try to figure out why change is happening the way it does. Automating this effort is a very ambitious goal in this context. Finally, the validation of existing research results with more studies from other groups, different software tools, hopefully extending the set of studied data sets, is imperative to allow us progressively to move towards ‘laws’ rather than ‘patterns’ of change in the field of understanding schema evolution.

References

1. Cleve A, Gobert M, Meurice L, Maes J, Weber JH (2015) Understanding database schema evolution: A case study. Sci Comput Program 97:113–121

2. Curino C, Moon HJ, Tanca L, Zaniolo C (2008) Schema evolution in wikipedia: toward a web information system benchmark. In: Proceedings of ICEIS 2008, Citeseer
3. Curino C, Moon HJ, Deutsch A, Zaniolo C (2013) Automating the database schema evolution process. VLDB J 22(1):73–98
4. Hartung M, Terwilliger JF, Rahm E (2011) Schema Matching and Mapping, Springer, chap Recent Advances in Schema and Ontology Evolution, pp 149–190
5. Herrmann K, Voigt H, Behrend A, Lehner W (2015) Codel - A relationally complete language for database evolution. In: Proceedings of 19th East European Conference on Advances in Databases and Information Systems (ADBIS 2015), Poitiers, France, September 8-11, 2015, pp 63–76
6. Lehman MM, Fernandez-Ramil JC (2006) Software Evolution and Feedback: Theory and Practice, John Wiley and Sons Ltd, chap Rules and Tools for Software Evolution Planning and Management. ISBN-13: 978-0-470-87180-5
7. Lin DY, Neamtiu I (2009) Collateral evolution of applications and databases. In: Proceedings of the Joint International and Annual ERCIM Workshops on Principles of Software Evolution (IWPSE) and Software Evolution (Evol) Workshops, IWPSE-Evol '09, pp 31–40
8. Manousis P, Vassiliadis P, Zarras AV, Papastefanatos G (2015) Schema evolution for databases and data warehouses. In: 5th European Summer School on Business Intelligence (eBISS 2015), Barcelona, Spain, July 5-10, 2015, Lecture Notes in Business Information Processing (LNBIP) v. 253, pp 1–31
9. Qiu D, Li B, Su Z (2013) An empirical analysis of the co-evolution of schema and code in database applications. In: Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering, ESEC/FSE 2013, pp 125–135
10. Sjøberg D (1993) Quantifying schema evolution. Information and Software Technology 35(1):35–44
11. Skoulis I, Vassiliadis P, Zarras A (2014) Open-source databases: Within, outside, or beyond Lehman's laws of software evolution? In: Proceedings of 26th International Conference on Advanced Information Systems Engineering - CAiSE 2014, pp 379–393
12. Skoulis I, Vassiliadis P, Zarras AV (2015) Growing up with stability: How open-source relational databases evolve. Information Systems 53:363–385
13. Vassiliadis P, Zarras AV (2017) Survival in schema evolution: Putting the lives of survivor and dead tables in counterpoint. In: Proceedings of 29th International Conference on Advanced Information Systems Engineering(CAiSE 2017), Essen, Germany, June 12-16, 2017, pp 333–347
14. Vassiliadis P, Zarras AV, Skoulis I (2015) How is Life for a Table in an Evolving Relational Schema? Birth, Death and Everything in Between. In: Proceedings of 34th International Conference on Conceptual Modeling (ER 2015), Stockholm, Sweden, October 19-22, 2015, pp 453–466
15. Vassiliadis P, Zarras A, Skoulis I (2017) Gravitating to Rigidity: Patterns of Schema Evolution -and its Absence- in the Lives of Tables. Information Systems 63:24 – 46
16. Wu S, Neamtiu I (2011) Schema evolution analysis for embedded databases. In: Proceedings of the 2011 IEEE 27th International Conference on Data Engineering Workshops, ICDEW '11, pp 151–156