Challenges and Opportunities in Automating DBMS: A Qualitative Study

Yifan Wang, Pierre Bourhis, Romain Rouvoy, Patrick Royer

▶ **To cite this version:**

Yifan Wang, Pierre Bourhis, Romain Rouvoy, Patrick Royer. Challenges and Opportunities in Automating DBMS: A Qualitative Study. ASE '24: 39th IEEE/ACM International Conference on Automated Software Engineering, Oct 2024, Sacramento - Californie, United States. pp.2013-2023, 10.1145/3691620.3695264 . hal-04771192

HAL Id: hal-04771192
https://hal.science/hal-04771192v1

Submitted on 7 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Challenges & Opportunities in Automating DBMS: A Qualitative Study

Yifan Wang
Orange / Inria / Univ. Lille,
UMR CRIStAL 9189, France
yifan.wang@orange.fr

Pierre Bourhis
CNRS / Univ. Lille / Inria,
UMR CRIStAL 9189, France
pierre.bourhis@univ-lille.fr

Romain Rouvoy
Univ. Lille / CNRS / Inria
UMR CRIStAL 9189, France
romain.rouvoy@inria.fr

Patrick Royer
Orange
France
patrick.royer@orange.com

ABSTRACT

Background. In recent years, the volume and complexity of data handled by *Database Management Systems* (DBMS) have surged, necessitating greater efforts and resources for efficient administration. In response, numerous automation tools for DBMS administration have emerged, particularly with the progression of AI and machine learning technologies. However, despite these advancements, the industry-wide adoption of such tools remains limited.

Aims. This qualitative research aims to delve into the practices of DBMS users, identifying their difficulties around DBMS administration. By doing so, we intend to uncover key challenges and prospects for DBMS administration automation, thereby promoting its development and adoption.

Method. This paper presents the findings of a qualitative study we conducted in an industrial setting to explore this particular issue. The study involved conducting in-depth interviews with 11 DBMS experts, and we analyzed the data to derive a set of implications.

Results. We argue that our study offers two important contributions: firstly, it provides valuable insights into the challenges and opportunities of DBMS administration automation through interviewees' perceptions, routines, and experiences. Secondly, it presents a set of findings that can be derived to useful implications and promote DBMS administration automation.

Conclusions. This paper presents an empirical study conducted in an industrial context that examines the challenges and opportunities of DBMS administration automation within a particular company. Although the study's findings may not apply to all companies, we believe the results provide a valuable body of knowledge with implications that can be useful for future research endeavors.

CCS CONCEPTS

• **General and reference** → **Empirical studies**; • **Social and professional topics** → **Automation**.

KEYWORDS

Automation, DBMS, Empirical research, Qualitative methods

1 INTRODUCTION

With the volume of data needing storage continuously rising, accompanied by its growing complexity [21], the task of administering DBMS for data storage has become increasingly challenging. Fortunately, substantial progress has been made in the field of DBMS administration automation, especially with the incorporation of *Artificial Intelligence* (AI) and *Machine Learning* (ML) technologies.

These innovations have led to the introduction of increasingly sophisticated tools and solutions. Noteworthy progress in DBMS administration automation has been achieved by both commercial products and academic projects, offering varying levels of automation. For instance, some products aim for complete automation of DBMS operations, such as the Oracle Autonomous Database [13], NoisePage [35, 44], and CrimsonDB [1]. Others, like IBM DB2 [48, 49] and Microsoft SQL Server [27, 40], incorporate multiple intelligent features to enhance user experience with DBMS. Additionally, academic initiatives, such as [52] and [54], are concentrating on the fine-tuning of DBMS configurations, further contributing to the field's development.

Nonetheless, based on prior discussions with DBMS experts within a leading European telecommunications company, it was found that despite the promising advancements made in DBMS administration automation by AI/ML-based tools, both from industrial publishers and academic sources, the adoption and intention to adapt these tools remain very limited at the company. This observation was later corroborated by our qualitative study.

Given the unexpected limited usage and intended adoption of DBMS automation tools, despite their potential to cut costs and ease the burdens of *Database Administrators* (DBA), the primary objective of this paper is to explore the reasons for their sparse adoption within the company. This investigation aims to inform strategies that promote the application of DBMS administration automation tools, enhance DBMS automation overall, and optimize human resource allocation in these projects.

To thoroughly investigate this issue and encourage the adoption of the newly introduced automation tools for DBMS, it is essential to understand users' perceptions of DBMS automation, identify the actual challenges DBMS users face in industrial settings, and explore the obstacles that limit the application of automation tools, taking into account both human and technical factors.

To achieve this, we used qualitative methods and conducted in-depth, semi-structured interviews, and utilized grounded theory to collect and analyze data from experts with various profiles. The goal of this study is to promote DBMS automation by understanding its challenges and opportunities. This entails exploring users'

perceptions of DBMS automation, and their challenges with DBMS in industrial settings, and summarizing a set of implications for both DBMS users and researchers to more effectively advocate for DBMS administration automation.

Previous studies, such as [44, 52, 54], primarily focused on technical difficulties when developing automation functionalities for DBMS, with little attention given to human factors. In the meanwhile, [31] highlights the importance of considering the human dimension as a major obstacle to DBMS automation. However, to the best of our knowledge, existing studies do not thoroughly cover three critical aspects necessary to promote DBMS automation: i) users' perceptions of DBMS automation, ii) users' difficulties with DBMS, and iii) users' concerns regarding the adoption of automation tools in an industrial context.

Our study considers DBMS administration automation as the automation of routine tasks across the life-cycle of DBMS, encompassing deployment, monitoring, tuning, maintenance, backups, updates, and beyond.

Contribution. This paper reports on a qualitative study of challenges and opportunities in DBMS automation among experienced DBMS users of a large company, to identify the reasons behind the limited adoption of automation tools. Concretely, we conducted interviews with 11 DBMS experts with diverse profiles including DBAs, managers, product owners, architects, and developers, with the ambition to cover their opinions, routines, difficulties, and requirements for DBMS automation under an industrial context. The key contributions of this paper are:

(1) Providing a detailed understanding and examples of the interviewed DBMS users' awareness and knowledge about DBMS administration automation.
(2) Identifying the main difficulties and challenges that DBMS users encounter in their daily activity with concrete cases.
(3) Investigating the challenges and constraints when applying automation tools into production under an industrial context
(4) Deriving a set of implications to better promote DBMS administration automation.

It is crucial to recognize that this study, despite being limited to a specific company and involving a relatively small sample size, was conducted within a large international company. This setting enabled us to engage interviewees with various profiles from different departments and countries, who brought a variety of technical backgrounds to the research. Although the generalizability and transferability of our findings might still be constrained, we are confident that our insights provide valuable knowledge for researchers, DBMS users, and tool developers, offering the potential to advance the field of DBMS administration automation.

Outline. The structure of this paper is as follows. Section 2 discusses the related works in the area of DBMS administration automation and highlights our contribution to those studies. In Section 3, we formalize our research methodology and experimental protocol. Section 4 analyzes and discusses the observations and findings derived from the interview responses. Section 5 reports on the implications of our findings. Finally, Sections 6 and 7 cover the validity threats and our conclusion, respectively.

2 RELATED WORK

As mentioned in Section 1, academic projects such as [23, 35, 44, 52, 54] discussed the constraints and challenges of the corresponding automation tool from a research and technical (architect, algorithmic) perspective.

On the other hand, we have works that focused more on a higher level of DBMS automation by specifying necessary features to achieve a higher autonomy in DBMS. For example, [44] focuses on Self-Driving Databases, and proposes 3 main functionalities to achieve higher automation of DBMS: the ability to automatically select actions to improve some objective function (e.g., throughput, latency, cost), this selection also includes how many resources to use to apply an action; the ability to automatically choose when to apply an action; and the ability to automatically learn from its actions and refine its decision-making process.

In the meanwhile, [31] specified the principle features (Self-knowledge, Self-configuring, Self-optimization, Self-healing, Self-protection) of DBMS to achieve a higher level of autonomy and discussed the general technical challenges of achieving these features, and also highlights that gaining the trust of DBAs is a major obstacle for DBMS administration automation, which are following what we have discovered in our research.

Progress beyond the state of the art. In this qualitative study, we examine the challenges and opportunities of DBMS automation by covering three critical aspects necessary to promote DBMS automation: i) users' perceptions of DBMS automation, ii) users' difficulties with DBMS, and iii) users' concerns regarding the adoption of automation tools in an industrial context.

We explore the acknowledged significance of DBMS automation within the industry, presenting the typical practices in DBMS automation. Our participants have identified major challenges, including managing multiple instances, nodes, environments, and products, which significantly complicates DBMS usage. Additional challenges arise from peripheral issues in anomaly analysis extending beyond DBMS, difficulties in communication with application teams, and a lack of adequate experience among many dealing with DBMS. Despite advancements in optimization by researchers and DBMS publishers, these are still perceived as challenges, along with ensuring *Quality of Service* (QoS) and integrating application understanding into operation and capacity planning, the latter posing difficulties due to limited data but offering significant industry benefits.

Regarding automation tool implementation in production, we distinguish between human and technical constraints. Technically, the main concerns include the potential risks of automation, its high costs (computational, development, financial, and complexity), and the need for adaptation to specific applications. On the human side, we identify critical barriers: skepticism towards automation tools, the desire for complete control, lack of support, insufficient time to adopt new tools, and the risk of losing expertise. The impact of these human factors is equally crucial for enhancing DBMS administration automation.

3 METHODOLOGY

To achieve our objective, we adopted a qualitative research approach with Grounded Theory methods based on [22], including

Table 1: List of Interviewees and their Profiles

ID	DBMS Type	Current Role	Experience	Entity
I1	XtraDB[17], Redis[19], MongoDB[12]	Manager	>20 years	Cloud Platforms
I2	Oracle Database[15], MariaDB[11], PostgreSQL[18], SQLserver	Manager	>20 years	Cloud Service
I3	MongoDB	DBMS expert	>20 years	Cloud Service
I4	Oracle Database, PostgreSQL, MariaDB, MongoDB	DBMS expert	>20 years	Products Support
I5	Oracle Database, PostgreSQL, Cassandra[6], MariaDB	DBMS expert	>20 years	Cloud Service
I6	Oracle Database, MariaDB, MySQL, PostgreSQL	DBMS expert	>20 years	Cloud Service
I7	MongoDB, MariaDB	Product Owner	>10 years	Financial Service
I8	Oracle Database, PostgreSQL, MySQL, SQLserver	Manager	>20 years	Network
I9	MongoDB, Elasticsearch[7], Redis	Architect Cloud	>10 years	Cloud Infrastructure
I10	MongoDB, Oracle, Cassandra	Developer	>15 years	Financial Service
I11	MongoDB, Oracle, MySQL, MariaDB, PostgreSQL	Developer	>20 years	Cloud Service

procedures such as coding, memoing, and theoretical saturation in [47]. Despite being complex and time-consuming [25], this approach allows us to generate new insights from the data rather than testing pre-determined research questions.

3.1 Interviews

Following empirical software engineering standards, we designed semi-structured interviews with initial and adaptive follow-up questions, guided by [33] for clear objectives and open dialogue. Our approach primarily utilized open-ended questions, with further probes for detail, centered around 12 core questions, allowing exploration based on participant responses.

(1) Which databases and tools related to databases do you use?
(2) What is the automation of database administration for you?
(3) What importance do you give to database administration automation?
(4) What tasks do you perform when managing DBMS?
(5) How much time would you spend on each of the tasks?
(6) How do you prioritize your tasks based on the impact and the importance?
(7) Which tasks do you find 'painful' and why?
(8) What are the difficulties you face when maintaining and managing DBMS?
(9) Do/did you automate some of these tasks? What motivated this automation? Which tools do you use? When did it fail?
(10) What are the benefits and limits of such routines/tools? What are the limits to engage the automation of some tasks?
(11) Do/did you actively search for automation tools for database administration, why?
(12) What are the key properties and support that should be improved by the current database administration?

The questions (1) to (3) help to cover the participants' knowledge and awareness of DBMS administration automation. Then, questions (4) to (8) aim to discover the routine, daily activity, and difficulties encountered by our interviewees. Finally, questions (9) to (12) allow us to investigate further the challenges and opportunities in DBMS administration automation.

Interviews with I1 to I11 are conducted either in-person or via video conference, based on their availability and location. The interviews progressed through three phases: an initial narrative introduction detailing the study's aim and interview process, followed by a semi-structured segment, during which we posed the above-mentioned questions along with follow-up inquiries. It concluded with a segment dedicated to addressing participants' questions and sharing further information. The mean duration time of the interviews is 53 minutes, with a minimum duration of 31 minutes, and a maximum duration of 66 minutes.

3.2 Interviewees

Participants were selected based on their extensive experience with DBMS, which facilitates a deeper understanding of the technology's details, strengths, and limitations. To ensure a diverse range of perspectives, all participants have at least 10 years of experience and continue to work with DBMS daily.

The participants in our study were volunteers who responded positively to our interview invitation and were actively engaged in DBMS projects at an international telecommunications company, boasting over 100,000 employees. The rationale behind choosing participants from the same company such as in [28, 42] is to assess the role of the company in the practice of DBMS experts. However, our study specifically focuses on promoting DBMS administration automation within a company and providing detailed empirical analysis, without aiming to create a universally applicable model for companies of varying sizes, sectors, or policies. It is important to note that the company is large and comprises many different entities, each with its independent ecosystem and operational practices. This organizational structure ensures a wide range of perspectives and practices. We carefully selected participants to ensure diversity in their departments, roles, countries, and policies. This was done to avoid homogeneity in their approaches to DBMS and to enhance the generalizability and transferability of our findings.

Rather than setting a specific number of participants, we conducted interviews until we reached a level of data saturation as in [51], which occurred after 11 interviews. Despite differences in the technologies and project types mastered by the participants, we observed convergence in the collected data and thoughts, which is consistent with similar works that have studied similar populations [32, 43, 46]. To protect the privacy and confidentiality of our participants, we used code names ranging from I1 to I11 and omitted any sensitive information, such as team or project names.

Table 3 shows the characteristics of our interviewees, their familiar DBMS technologies, their functions, their experiences and their entities. The main criteria for the interviewee selection is based on their experience, and we try to have diverse profiles among our interviewees. I1 manages a *Database-as-a-Service* team. I2 transitioned from managing a database optimization team to cloud services. I3, I4, I5, I6 are senior DBAs, experts of DBMS with different types of technologies. I7 is a technical product owner that oversees DBMS deployment and administration. I8 is a manager of a team of DBAs, who is in charge of operating a set of databases. I9 is a cloud architect, who mainly works with managed solutions with cloud providers. I10, I11 are developers, who work with an application that highly relies on DBMS. It is important to note that engaging experts from various departments and countries presents challenges, primarily due to time their demanding schedules and zone differences.

3.3 Transcription

For each conducted interview, we transcribed the recording using a denaturalism approach, which is a method used in similar studies,

such as [32, 43]. This approach emphasizes the interview content and is less exhaustive than other methods, such as verbatim in [41], while still being trustworthy. The transcription was done in the same language as the interview, but we translated some parts into English to include participant opinions in Section 4.

3.4 Analysis

We employed the Straussian grounded theory coding procedure [30, 47] for data analysis. We start with the open coding phase where we carefully read our transcripts multiple times to condense each chunk of data into a label, based on the inferred meaning of the text. These labels were referred to as "open codes". Subsequently, we conducted axial coding to establish the connections among the previously extracted open codes. We then utilized selective coding to pinpoint the central ideas that encompassed the collected data. Lastly, we reviewed the transcripts again and assigned all content segments to a core idea that was relevant to the selected data.

4 OBSERVATION & FINDINGS

Table 2 summarizes the key insights from our qualitative study, with the core ideas that fit our objectives. Here, we use 'DBMS users' as a general term to refer to the interviewees, their role and their focus vary. In each cell of Table 2, a reference mentioned by the participant regarding every idea that falls under the core idea is indicated by the cross mark (✓). The following section delves into our observations, with each subsection focusing on a reported idea. Each idea is then given its dedicated paragraph for discussion. Similar ideas that convey related meanings and objectives are clustered together under a single category. To summarize the observations and findings, and to offer our insights and recommendations, a discussion is provided at the end of each category.

4.1 DBMS users' awareness & knowledge of DBMS automation

4.1.1 Current routines in DBMS administration automation. Our research indicates that all interviewees are knowledgeable about automating DBMS administration, with many having experience in this area. However, only a small number of them are familiar with AI/ML-based automation tools. The primary approaches to automation mentioned in our study include Kubernetes operators, schedulers, scripts, such as shell and infrastructure-as-code-based software, monitoring tools, and DBMS internal tools.

I1 reported that "*We use Kubernetes operator to automate the majority of the tasks*". I1's team provides managed database-as-a-service using mainly containers with the help of Kubernetes[10], which is a container orchestration system for automating software deployment, scaling, and management. Kubernetes Operators extend Kubernetes' capabilities for managing complex, stateful workloads, allowing for automated rolling updates, failure recovery, and many other functionalities.

The scheduler is raised to be an important part of DBMS administration automation. "*We use the scheduler to make maintenance operations such as rebuild index, reorganization when the workload is light*", reported I6. Recurrent operations, such as backup, are also programmed with the scheduler.

Additionally, the use of external and DBMS native tools is mentioned to be an important part of DBMS automation. Especially for database monitoring, which can be heavy work without proper tools if visualization of the databases' status is required. Software, such as Grafana[8], Monolog[3] and native tools for different DBMS technologies, are essential for the diagnostic and to ensure the databases' status.

"*We develop and use scripts to automate tasks*" reported I3, I5, I6, I8, I9. Scripts are widely applied in DBMS administration automation, I5, I6, and I8 reported using scripts of infrastructure-as-code software, such as Terraform[20] and Ansible[5] to automate the databases deployment process. Such software allows users to define infrastructure using a declarative configuration language, such as HashiCorp[9], therefore saving the manual process of database deployment.

Scripts are used in many cases, especially regarding repetitive tasks. "*We create our homemade scripts every time we see repetitive tasks*", reported I8. As for concrete examples, I6 reported using scripts to automate the statistic calculation process in Oracle databases, which is a crucial operation to keep Oracle databases efficient. DBMS updates and anomaly detection are also reported to be automated by using scripts by I9. Despite widespread script use, automation potential is constrained by the script developer's expertise. As I9 notes, automating administrative tasks demands a thorough understanding of the DBMS and significant time investment. "*It's always a human who writes the scripts and programs the Scheduler. It's always the human who decides afterward what to do with the program and its frequency to be executed*" reported I5.

Discussion. We observe that in DBMS administration, there are numerous recurrent and repetitive tasks, and the participants demonstrated a keen interest in automating these tasks using various methods like custom scripts and schedulers. However, it appears that the level of automation achieved by these routines is limited and constrained by the developer's expertise, and the utilization of AI/ML-based approaches for DBMS administration automation remains quite limited among the participants.

4.1.2 DBMS automation evolution. "*DBMS needs less automation than before*" is a mutual feeling shared among I1, I2, I3, I5, I6, I10, implying that the DBMS is getting easier to access and use compared to older versions of DBMS.

Firstly, one of the factors contributing to the increased accessibility of DBMS is the efforts made by DBMS vendors to enhance system stability and robustness. "*Investigating why the database failed is probably the most annoying task. Understanding why their nodes fall and restarting them, but these are things that we see less and less because little by little, the DBMS products have increased the robustness and eliminated the factors that caused the nodes to fall*" is witnessed by I1.

By enhancing the robustness of the DBMS product, the DBMS is considered more automated by our interviewees, since fewer DBMS incidents need to be managed. I1 reported a case where they provide a 3-site cluster, which is very sensitive to the latency of different sites. During slowdowns between sites, nodes of a site could become isolated or crash. XtraDB later added the timeout parameters at the communication level between the nodes, the robustness is enhanced and they have fewer problems.

Table 2: Summary of interview content analysis

Topics	Ideas		I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11
DBMS users' awareness & knowledge of DBMS automation	I am doing DBMS automation		✓		✓		✓	✓	✓	✓	✓		
	DBMS needs less automation than before		✓	✓	✓	✓	✓	✓				✓	
	DBMS automation can reduce repetitive work and save time		✓	✓		✓	✓	✓	✓	✓		✓	✓
	DBMS automation can reduce resource consumption				✓	✓	✓			✓	✓	✓	✓
Difficulties encountered & Automation opportunities	Anomaly Detection	Scale	✓		✓	✓	✓			✓	✓		
		Peripheral		✓				✓	✓				
		Application's Behavior	✓	✓	✓	✓			✓				
	Users have no adequate experience		✓	✓	✓			✓	✓		✓		
	Query, execution plan optimisation			✓				✓				✓	✓
	Operation planning		✓	✓	✓	✓		✓		✓			
	Capacity planning				✓	✓	✓		✓				
Challenges & Constraints in DBMS automation application	Technical Factors	Automation can be dangerous	✓	✓		✓	✓			✓			
		The cost automation can be high				✓			✓		✓		
		Automation need to adapt to application	✓		✓	✓							
	Human Factors	Lack of confidence for automation tools		✓		✓		✓			✓		
		Users want absolute control over the product				✓			✓		✓		
		Users may lack support					✓	✓		✓			
		Users lack time to apply new tools			✓			✓			✓		
		Automation could cause lose of expertise						✓	✓				

I6 also observed that earlier versions of the Oracle database experienced multiple optimizer changes, impacting the performance of existing queries: "*In the past, for a slight update of Oracle database, we were still working 2 months after the update. Because there were treatments that we hadn't detected. But since v11, v12, and so on, we no longer have these problems.*" As a result, close monitoring of database performance was necessary to prevent any production errors during updates and upgrades. However, this is no longer the case with the latest versions of Oracle databases, even though there are still many modifications made to the optimizer by Oracle according to their white book[2]. Although the improved compatibility among different versions is not tailored to enhance the automation of the DBMS, it eliminates the necessity for extensive human intervention which once demanded automation and significant effort.

Secondly, new algorithms and tools have been introduced by the vendor to help with DBMS administration automation. "*Oracle database have implemented many algorithms to automate tasks such as the optimization of queries and database resizing*" reported I4. "*You have paid options in Oracle database, such as Advanced Security, to ensure the security of your databases and encryption option to do TLS*", stated I6.

Thirdly, more recent non-relational databases (NoSQL) are considered to be lighter than traditional relational DBMS. "*For PostgreSQL database, you need to do the memory tuning. But MongoDB will manage its memory in relation to the RAM that you have allocated to the OS and it will manage its own memory of its WiredTiger engine. It does it rather well*", as reported by I7. The feeling is shared with I3: "*Apart from the backup and the deployment, in terms of administration and operation, reorganization of data is the only task that I feel like necessary to be automated around MongoDB.*" I10 also thinks highly of the flexibility and self-sufficiency offered by MongoDB. As a developer, I10 occasionally relies on the assistance of DBAs when dealing with relational DBMS like Oracle. However, due to MongoDB's comparative ease of management, he can handle tasks independently.

Discussion. DBMS publishers have significantly improved accessibility and user-friendliness, this evolution towards greater robustness and stability has reduced the need for extensive human intervention, once a critical necessity for automation and effort. Interviews reveal that many believe modern DBMS demands less automation compared to their predecessors.

4.1.3 Importance of DBMS administration automation. During our study, every interviewee expressed the belief that the automation of DBMS administration is crucial. "*Automation can reduce repetitive work and save time*", mentioned by the majority of our interviewees. According to I1: "*We try to eliminate repetitive work as much as possible [...]. Every time we see a task that needs to be done more than one time, we try to automate that immediately.*"

"*Automation saves time and allows us to do other more interesting tasks*", stated I7. For I9, the benefit of DBMS administration automation is making complicated tasks straightforward, which allows people with less experience to solve problems in production. I10 also highlights the fact that not only DBMS administration automation can reduce repetitive work, the automation of certain manual tasks also makes the process more reliable.

DBMS administration automation is also mentioned as being able to reduce computational resource consumption. "*For example, the benefit (automation with intelligent scripts instead of a fixed scheduler) is to avoid rebuilding an index when it is unnecessary and saves the CPU and RAM consumption*" stated I5.

Discussion. Automation of DBMS administration brings many benefits, the main benefits mentioned by our interviewees are: saving time, saving computational resources, reduce repetitive work. These benefits are the key factors that drive the automation of DBMS administration.

4.2 Difficulties encountered & automation opportunities

Although our participants find that the DBMS are getting increasingly automated and robust, and they have approaches such as scripts and schedulers to automate certain tasks, there remain some difficulties and challenges when automating DBMS administration. These challenges offer opportunities for researchers working in the field of DBMS.

4.2.1 Anomaly Investigation. The investigation and resolution of the alarm, incidents, and possible failure of the databases in the production environment and repair the system is considered to be one of the top priorities (I1, I5, I7, I8, I9). "*The priority is to ensure the quality of service*", reported I1. Being one of the main difficulties encountered by DBMS users, the anomaly investigation remains hard to automate for the following three reasons:

Scale. In large applications, we face multiple environments (different operating system releases, different database technologies, and versions) and multiple databases (up to 2,000 instances reported by I1), and this complicates the investigation of the problem and other administration tasks. "*It's okay when you have 2 environments, we can afford to verify them daily, but it's different when you have 200 different environments*", reported I4. And it is often difficult to make tests on different environments, "*You see problems in the production environment that you do not see in the test environment*", reported I9 when trying to write a script to hot update the databases.

"*What takes us most time is the fact that we are managing thousands of databases*" reported I1. Managing databases on a large scale is a widespread issue, according to I5, "*An average DBA in teams that manage large environments, has to manage several hundred environments*".

Cluster deployment, aimed at ensuring high availability and fail-safe protection, complicates DBMS management and anomaly resolution. "*More complex the architecture of the clusters is, more nodes we have, and we have more things to manage, we are going to have additional problems related to latency, anomalies and organization of data within the shard clusters*", reported I3.

I7 reported a case with a MariaDB cluster, "*we have several databases in a cluster, if you do a physical backup you won't be able to segregate the databases inside your backup, which means if you can't restore a single database, you have to restore all the databases. But it is not the case if you switch to logical backup.*"

Peripheral. Secondly, the problem can come from the peripheral devices and not the DBMS. "*I see in alert in production due to the loss of user connections. And it was not related to the base, it was a network failure, the problem was peripheral*", reported I2. I7 also reported backup failure only because there was not enough space on the disk allocated to the backup operations.

"*I have a request that takes too long. So we have to find if it is the query that is badly written if it is incoming data that is not going well, if it is the settings that are not going well, if it is the disks, because sometimes the disks are slow*", witnessed I6. When investigating a problem, we must consider all the possible failures, and peripheral issues are not negligible.

Application's Behavior. Thirdly, databases are attached to applications. Unpredicted behaviors in the application can cause the database to malfunction, "*When we observe incidents, for me this is more related to the current activity of the application than operational tasks*", reported I3. I3 witnessed a case when an application did many inserts and deletes, which caused the database to disorganize and occupy very large disk space while only containing a small amount of data. In addition, I4 reported to have seen projects launched multiple times with unnecessary batch treatments on tables, causing the database size to grow unexpectedly. I7 reported a case of a wrong database address used in the application causing the database to fail. I2 stated that "*what we often see in backup and restore, are application errors, data corruption, and human errors.*"

I1 witnessed a case of database failure where an extreme and not previewed workload is applied to their databases (insert and delete of big images in short periods), "*The DBA needs to understand the databases' application's behavior to understand the potential problems. When investigating a problem, the fact that we need to have interactions with clients to understand what they are doing to understand where went wrong is where we have most trouble with*".

Yet, despite the fact the anomaly of databases has various and complex causes, some anomalies are considered as problems that can be anticipated. "*You see certain errors in certain files, such as log files, that arrive from time to time, it makes the problem detectable in advance and can be dealt with proactively*", stated I7. Many basic anomalies can be detected using monitoring software and scripts, reported I9.

Discussion. It is hard to uncover the root causes of the databases' incidents, especially under the context of multi-environments, multi-instances, and multi-nodes in clusters. Databases' behavior is also related to the applications' behavior and peripheral devices that they are attached to. These factors make the investigation of the databases' problem difficult to automate.

Earlier research on anomaly detection, such as [34, 38, 45], primarily concentrated on identifying malicious activities. Other studies, like [36], were beneficial in determining the timing of anomalies, but had limitations when it came to pinpointing their root causes. Although these works hold great significance, they do not inherently address the challenges we uncovered in our study, including issues of scale, application behavior, and peripheral concerns.

4.2.2 Users miss adequate experience. The lack of experience and expertise of users is another difficulty reported by I1, I2, I3, I6, I7, I9. Depending on the profile of the interviewee, the 'User' that they referred to varies from application development team to database administrators.

"*Users and database administrators don't know the product*", stated I9. I9 reported a case where a DBA received an alarm indicating the lack of disk space, to solve this, the DBA removed an unnecessary index, which was needed, and led to the failure of the system. "*We can add CPU and RAM to meet performance requirements, but we are more constrained when we have an extreme workload due to a lack of knowledge or expertise of the developers*", reported I1. The lack of experience of the developers and DBA from the application team can influence the operation of the DBMS, and we can only acquire experience with time. "*I have people on the production side*

that I have known for several years, they are starting to be good but they knew nothing when they started", witnessed I3.

Discussion. Being extremely complex software, DBMS is known to be hard to manage since they require experience and a thorough understanding of the system ([26, 50]). Misjudgment due to the lack of experience can cause the system to fail, and the fact that most users and DBAs only have limited experience is considered to be an actual difficulty in the industry.

4.2.3 Query & execution plan optimisation . Depending on the profile and the technology used by our participants, query, and plan optimization to meet performance requirements are also mentioned to be one of the difficulties that they face in the production (I2, I6).

"*One of the things we miss is to be able to have a system that verifies if our plans, our path, or our way of doing things is optimal. The DBMS calculates a scoring with statistics and paths, which is used to find the execution plan. But very often, the scoring is flawed*" stated I2 when experiencing Oracle databases.

"*The problem that we often encounter, is the request that is badly written*" reported I6, even with the query optimizer integrated with the DBMS I6 still witnesses inefficient queries from inexperienced users causing the system to stall. The fact that I6 needs to deal with queries from inexperienced users strengthens our point in Section 4.2.2.

Additionally, despite that I10 has many years of experience around MongoDB, he still struggles with how to set optimal indexes, "*We don't have performance issues (with MongoDB), but personally, I know that my configurations are not optimal, […], and I don't have a plan regarding how to create proper indexes*". I11 shares a similar experience, revealing that the most challenging task for I11 involves fine-tuning complex queries and identifying the optimal data structure to boost the database's performance.

Discussion. Despite the efforts made by DBMS publishers regarding query and plan optimization, it is still considered a difficult subject. However, I6 also indicates that only 20% of applications would require a high-level optimization, it is not a mandatory procedure in most cases. And I1 stated that there is always the option to add more RAM and CPU to meet performance requests. I10 also brings up the challenge of identifying the optimal indexes for MongoDB; however, I10 is not particularly concerned by DBMS' performance issues. In short, optimization and fine-tuning is a challenge, but it does not appear to be a major obstacle in the activities of our interviewees.

4.2.4 Operation planning. To maintain DBMS QoS, users perform maintenance tasks like disk reorganization and upgrades, impacting the production environment by using significant computational resources. Minimizing operation duration is crucial to prevent service interruptions.

Currently, maintenance scheduling is based on the application's profile and relies on human expertise."*On XtraDB in particular, we have 5 or 6 instances that need optimizations (free up disk) from time to time. For one of our customers, this is done every 6 weeks, for the others, it may be every 6 months*", reported I1. I5 stated a similar observation, "*The frequency of scheduling of these operations depends on the application. In other words, there may be applications where there are many activities and modifications such as deletes, updates, etc. that disorganize the objects, so it may be necessary to update or rebuild the indexes of certain instances every week, whereas on other applications, it will not be disorganized.*" I6 also reported to use of integrated schedulers to set a specific time to run maintenance operations while the workload on the database is light.

Maintenance needs can be identified using DBMS metrics such as fragmentation ratio. However, automating these tasks is challenging due to potential service impact. "*You have to be able to determine the most favorable moment to run the operation and you have to be able to predict the duration of the operation. This is why all these operations are manually programmed. When we have to carry out large defragmentation operations, we simply stop the service*", reported I4. The fact that system shutdown is necessary for certain operations complicates the operation planning, "*Most of the systems work 24/7. So it is not so easy to find the patching window.*" reported I8 when having to patch his DBMS on a 24/7 application. Furthermore, depending on the application and the type of the company, sometimes it is not up to DBA to decide when they can shut down the system, it is decided by the responsible from the business side or responsible with a higher hierarchy (I8). Although updates and upgrades present a challenge in 24/7 systems, the rolling update is reported to be applied to update the system while providing a stable service in certain use cases (I1).

Discussion. Maintenance planning is closely linked to application behavior, often impacting the production environment and sometimes necessitating system downtime. This planning demands expertise in understanding the operation's effect on the DBMS and the application's behavior. Predicting short-term workloads is crucial for assessing if maintenance might affect QoS. Although there have been advances in workload prediction [24, 29, 39], these methods require high-quality training data and additional computational resources for precise predictions. However, obtaining such data can be challenging due to legal and security constraints in the industry, limiting their practical use.

4.2.5 Capacity planning. Estimating a project's capacity is crucial for creating a robust system with a DBMS, involving calculations for computational resources like RAM, CPU, and disk space. Yet, accurately predicting these needs is challenging in the industry. Accurate capacity planning hinges on predicting project workloads, a detail often unavailable early in projects, as mentioned by I3.

This planning predominantly relies on limited human expertise. I4 highlighted the difficulty of achieving accurate capacity predictions, "*Projects come up with charts saying, 'Here we have these volumetric projections over a year', and we realize that either it's oversized, or it's undersized, it's rarely well sized*". When lacking an accurate capacity estimation, DBMS users tend to oversize the database to ensure that the database is operational. "*We tend to always increase resources and we rarely reduce them, and that's a big problem today with Corporate Social Responsibility, we have to think differently*", reported I4. The amount of wasted resources may be large, as I7 reported to have reduced 350 unnecessary CPUs and 400 GB of unnecessary disk space in a project.

With the capacity demands changing over time, development efforts are also increased when capacity planning is not enabled. "*At the beginning of a project, we can use a non-partitioned model with 100 GB. And let us imagine that the base grows to 10 TB. We*

can no longer remain in a non-partitioned model", stated I4. I3 also reported a case where a project forgot to delete inactive accounts, causing a waste of disk space. These problems can be avoided with accurate capacity planning.

Furthermore, accurate capacity forecasting enables DBMS users to anticipate and address issues proactively, such as determining the need for additional resources to handle growing workloads (I3, I4). According to I3, "*If we were able to predict the future derivatives of your database, you would be able to easily integrate other actions. Without the capability, you will often be on very instantaneous issues*".

Discussion. Accurate capacity planning and workload prediction enable proactive problem management, minimize unnecessary computational resource allocation, and facilitate advanced architecture design, thus conserving development efforts. However, the scarcity of detailed information from DBMS users and product owners at project initiation complicates precise planning and forecasting. Conversely, services such as Amazon Web Services[4], Microsoft Azure[16], and Oracle Cloud[14] now offer auto-scaling features that monitor applications and automatically adjust capacity, thereby reducing the complexity of capacity planning. Nevertheless, within the industrial context, numerous projects, particularly legacy ones, are unable to migrate to the cloud due to financial and legal barriers.

4.3 Challenges & constraints in DBMS automation applications

In this section, we discuss the challenges and constraints of automation tool adoption in the industry revealed in our study. The ideas that we revealed in our study are categorized into 2 categories: human factors and technical factors.

4.3.1 Technical Factors.

Automation can be dangerous. The QoS of the DBMS is very critical in production. Administration tasks, such as backup, and index rebuild, consume computational resources and, therefore, harm the database (I2, I7). The modification of important parameters and execution plans can all impact the performance (I1), not to mention that some operations require the database to shut down and restart (I4).

"*Automating everything is a bit dangerous. Many actions that can be done in a completely automated way, but it poses problems for the operation of the databases when you start to touch the volume, the data, and things like that. We need to put more safeguards in place*", reported I1. I4 reported that Oracle databases are integrating AI-based tools to automate and help users with administration tasks, even though these tools are great in theory, the fact that these tools may change execution plans and destabilize DBMS performance is dangerous.

Discussion. The fact that many administration tasks could influence the databases' performance is considered to be dangerous in production. This makes people reluctant to delegate sensitive tasks to automation tools, because they may cause unexpected consequences if not thoroughly designed.

The cost of automation can be high. Even though DBMS administration automation brings many benefits, there is still a cost behind the automation. "*When the version of DBMS changes, I may have to change my scripts. The maintenance of the final scripts is expensive too*", reported I9. Updates and upgrades of the DBMS bring changes, human efforts are needed to maintain the automation tools.

Automation also has a computational cost. "*If we let Oracle database auto-tune. It is devastating at the level of memory resources consumed by the Oracle database as it does a lot of operations. All these operations are also stored in their dictionary, so all this has a cost at the CPU level and the disk level. Is it worth setting up this system? I am not sure*", reported I4.

Many commercial cloud providers deliver managed database service and strong automation tools, even though they reduce largely the DBMS administration workload, their price is still blocking user adoption (I7).

Adopting an automation tool also adds an extra layer of complexity. In case of anomaly and problem, it increases the difficulty to resolve the problem. "*Behind it, there are events that appear in the alert file, so afterward we have to filter what can be problematic from what is not*", reported I4 when using an automation tool to resize the database. "*If I have an error, is it an error of my script or it's the DBMS underneath?*", I9 shared similar concerns.

Discussion. The automation of DBMS administration has a human cost, computational cost, and even financial cost, and it could potentially have a cost of increasing the complexity of the system. It is important to control trade-offs between the cost of automation and the benefits that automation can bring.

Automation needs to adapt to application. The DBMS is attached to an application and must adapt to the requirements of the application and the human operation behind the application. "*For me, it cannot be all automated, there are times when human interaction is needed to understand the customer's use*", reported I1. This adds extra complexity to the DBMS automation. For example, I4 reported a case if an erroneous query is sent to the Oracle database with the auto-tuning option on, then the DBMS can consume a lot of resources only to make bad tuning and optimization decisions based on the erroneous query and degrade the DBMS's performance.

Discussion. The application must be taken into account when doing DBMS administration automation, the changing application and possible human errors could cause the automation to fail. This complicates the adoption of automation tools.

4.3.2 Human Factors.

Lack of confidence in automation tools. Given that automation of administration tasks can be dangerous, DBMS users are more careful when adopting automation tools. "*We are always afraid that the automation tool does not do it as well as we do. [...] We are always worried because, in the end, it's still us who will be bothered somewhere*", reported I4. I2's team already has a well-done automation tool for backup and restore, but since it is a very sensitive task, they still demand a DBA to make sure the operation is successful. "*The automation tool must be perfect, otherwise it does not make any sense*", reported I8 and I9. Because of the sensitivity of DBMS regarding data integrity and data security, people often lack confidence in automation tools.

Discussion. DBMS are very sensitive, all unexpected and incorrect operations could jeopardize the entire system. For an automation tool to be adopted, the tool must be reliable, for DBMS users to have a deep confidence. In the end, it is not only about the robustness of the automation under all conditions but rather whether the tool appears to be trustworthy to the decision-makers. [31] also highlights that gaining the trust of DBAs is a major obstacle for DBMS administration automation.

User wants absolute control over the product. "*We want to control everything from A to Z*", reported I4. As highlighted in Sections 4.3.1 and 4.3.2, automation poses risks and lacks full user trust, especially given the sensitivity of DBMS. Users frequently demand total control over the system.

"*Oracle database tried to integrate several automatic tools to do auto-tuning. But it is dangerous to leave them active. The request execution plans are critical to the business if we want to guarantee the stability of the performances, we prefer to fix the plans*", stated I4. Meanwhile, I4 also highlights that once the data model evolves, the fixed plan could become no longer valid. Delegating the tuning task to the Oracle database allows the system to adapt to evolving data models, but I4 made the compromise and chose the stability and total control of the execution plan. For I7, it is good to have an automation tool, but it is also essential for him to know how the tool functions and how the decisions are made by the tool.

Another critical aspect of having absolute control is that DBMS users must be able to access the product to resolve issues when problems arise. I9 uses managed database service provided by the DBMS publishers, while this reduces the workload of the DBMS administration, I9 stated that "*when you're in trouble, you can't get your hands on it*".

Discussion. When delegating administration tasks to automation tools, DBMS users could lose control and the track of states of the DBMS. This is blocking the adoption of automation tools. To avoid this, automation tools must reveal complete information about the changes that it has made and allow users to have global control.

User lacks support with open-source, academic project. Due to the increasing license fee of commercial DBMS products, our participants tend to adopt open-source and community solutions (I1, I4, I5, I6, I7, I8). As we have more control over the open-source DBMS, this is favorable for the academic community to research and develop automation tools for open-source DBMS.

However, academia and the open-source community have fewer resources compared to commercial products. Therefore, they only provide limited support. "*With open-source projects, the DBA is, unfortunately, less confident, sometimes he is a bit on his own [...] I would say that the DBA is worried when there are breakdowns, bugs*", reported I5.

I6 reported a case of trying to adopt an open-source automation tool for DBMS migration, but it only solves a partial problem and has limited support. I6 then abandoned this tool and decided to develop it on his own. "*open-source solutions cost us more, more human power on the maintenance because open-source needs more work, not everything is done ideally*", reported I8. The lack of support increases DBMS user's development costs and limits the adoption of open-source solutions.

Discussion. Many DBMS automation tools have been introduced in the open-source community by the researchers [37, 39, 53, 54]. In contrast to the comprehensive support offered by major corporations to their commercial automation tools, academic projects tend to provide limited support. The authors and researchers involved in these academic projects often have various commitments, making it challenging to establish contact with them.

User lacks time to apply new tools. "*We do not have the time to do it*", reported I3, I7, I9. Our participants are overloaded with their daily activities, making it impossible for them to be up to date about the state-of-art of DBMS administration automation tools and research topics. Nevertheless, they keep showing interest and have an open mind about ongoing research, "*I do not have that much time, it is a bit of a shame, but if I had more time I think I would keep myself a bit more up to date*", reported I7.

Discussion. It requires time and effort for DBMS users to be up to date about the novel and research solutions. The fact that DBMS users lack time to make such inquiries creates a gap between the industrial and research communities, therefore restraining further adoption of newly introduced automation tools.

Automation could cause loss of expertise. Loss of expertise caused by automation is another challenge that was revealed in our study. "*What customers really want is managed databases. They want to install, make queries and that the database lives on its own. They do not intend to calculate statistics or any other kind of thing*", reported I2. This is very common as DBMS automation and managed solutions can reduce largely the workload, and require less expertise. "*It will also diminish his knowledge and I have problems with all these black boxes, people do not know why they are in trouble because they do not know what they do. We have pseudo DBAs who do not know much and who have no idea what they are doing*" reported I7.

Discussion. While automation tools enable DBMS users to delegate administration tasks, this could cause a loss of expertise. On the other hand, "pseudo DBA" being able to manage databases also means that less expertise is required to operate complex DBMS.

5 IMPLICATIONS

Our study yielded several sets of implications for DBMS users, notably DBAs, researchers, and tool creators. Although the results may not be generalizable to all organizations due to our focus on one particular company, we believe that these findings provide a valuable knowledge base. They can enhance the understanding of DBMS user activities and the routines of DBMS administration automation, thereby facilitating the adoption of automation tools.

5.1 For DBMS users

DBMS users including product owners, managers of teams that provide database service, DBMS experts, developers, DBAs are the information source of our study, here are a couple of implications retrieved for them:

- Many of our participants showed a deep understanding of DBMS administration automation, but only a small number

of them are actively keeping up with the latest automation tools and instead sticking to their established routines. We believe that DBMS users should prioritize this activity by allocating more time to explore novel automation projects in the field.

- There are DBMS users who are reluctant to trust AI and ML-based tools, due to concerns about their reliability and stability. While it is true that these tools are still limited in their use cases, we encourage DBMS users to try them out to not miss out on potential automation opportunities.
- Based on our interviews, it appears that our participants are dissatisfied with the lack of expertise and experience in DBMS among many other DBAs and developers, which results in wasted time and effort in dealing with human errors. As a solution, we suggest that they invest more time in training those with less experience.

5.2 For researchers & tool creators

Our study revealed some of the difficulties encountered by DBMS users and they present implications of interesting and challenging research opportunities, they give numerous guides and specifications for tool creators:

- Our participants have shown a preference for adopting open-source DBMS products. As a result, we suggest that researchers and tool creators focus more on studying these types of products, which would provide researchers with better access.
- Our study has uncovered a range of intriguing industrial challenges, such as capacity planning with limited data, and operation planning with business constraints. Therefore, we encourage researchers to view these challenges as valuable opportunities for further investigation, taking into account the limitations under the industrial context.
- It is essential to strike a balance between the advantages and costs of automating DBMS administration. While automation can be computationally resource-intensive and require high-quality data, the costs of automation should not exceed the benefits it provides.
- Gaining the trust of DBMS users is a significant hurdle for DBMS administration automation tools. To overcome this obstacle, we recommend investing more effort into improving the transparency, applicability, and safety measures of the tools, to enhance their stability and reliability and, ultimately, earn the trust of DBMS users.
- Considering the sensitivity of DBMS, it is important to establish a suitable human-machine interface that grants DBMS users greater control over the process when required. This interface should also provide detailed information, to avoid appearing like a black box and undermining users' expertise.
- DBMS are closely related to applications, and the behavior of these applications is a significant contributor to DBMS anomalies. To address this issue, we recommend investing more effort into developing automation tools that capable of handling the dynamic behavior of applications.

6 THREATS TO VALIDITY

Generalizability. The interviewees may not be representative of all populations. The sample size and the particularity within a specific company, though providing a certain level of data saturation [47, 51], may not be large enough to ensure better generalizability to other populations. Additionally, selecting only experts and experienced DBMS users who are more interested in the topic than average may not be the best representation of all scenarios.
Credibility. A potential threat to our study's validity is the accuracy of participants' responses. Although we interviewed experts, their answers might still be influenced by external sources. To address this concern, we emphasized that the interview process was not intended to be judgmental and made a conscious effort to ask for specific details related to the participants' projects and profiles during the interviews.
Confirmability. A potential issue that may affect the accuracy of our study is the analysis, particularly the coding step of the interviews. To mitigate this concern and enhance the reliability of our conclusions, all the obtained results are carefully discussed by the authors and an extra expert in this domain.

7 CONCLUSION

In this paper, we document a qualitative study that involved conducting in-depth interviews with experienced DBMS experts employed at a large company. The insights and discoveries that we present offer: i) providing a detailed understanding and use-cases of the interviewed DBMS users' perspective on DBMS administration automation, ii) identifying the main challenges and difficulties that DBMS users encounter with concrete cases, and iii) investigating the challenges and constraints when applying automation tools to production under an industrial context. Our study has numerous implications for DBAs, DBMS users, tools creators, and researchers working with prioritizing DBMS administration automation.

REFERENCES

[1] 2017. CrimsonDB: A Self-Designing Key-Value Store. https://demosubmitter.github.io/. [Accessed 12-01-2024].

[2] 2017. Optimizer with Oracle Database 12c Release 2. https://www.oracle.com/technetwork/database/bi-datawarehousing/twp-optimizer-with-oracledb-12c-1963236.pdf. [Accessed 8-8-2023].

[3] 2020. Monolog. https://github.com/waza-ari/monolog-mysql. [Accessed 12-6-2024].

[4] 2023. AWS Amazon Autoscaling. https://aws.amazon.com/en/autoscaling/. [Accessed 8-5-2024].

[5] 2024. Ansible. https://www.ansible.com/. [Accessed 8-4-2024].

[6] 2024. Cassandra. https://cassandra.apache.org/. [Accessed 13-02-2024].

[7] 2024. Elasticsearch. https://www.elastic.co/. [Accessed 13-02-2024].

[8] 2024. Grafana. https://grafana.com/. [Accessed 8-4-2024].

[9] 2024. Hashicorp. https://www.hashicorp.com/. [Accessed 8-4-2024].

[10] 2024. Kubernetes. https://kubernetes.io/. [Accessed 12-11-2023].

[11] 2024. MariaDB. https://mariadb.org/. [Accessed 13-02-2024].

[12] 2024. Mongodb. https://www.mongodb.com/. [Accessed 13-02-2024].

[13] 2024. Oracle Autonomous Database. https://www.oracle.com/autonomous-database/. [Accessed 12-03-2024].

[14] 2024. Oracle Cloud Infrastructure Documentation Autoscaling. https://docs.oracle.com/en-us/iaas/Content/Compute/Tasks/autoscalinginstancepools.htm. [Accessed 8-5-2024].

[15] 2024. Oracle Database. https://www.oracle.com. [Accessed 13-02-2024].

[16] 2024. Overview of autoscale with Azure Virtual Machine Scale Sets. https://learn.microsoft.com/en-us/azure/virtual-machine-scale-sets/virtual-machine-scale-sets-autoscale-overview. [Accessed 8-5-2024].

[17] 2024. Percona XtraDB Cluster. https://www.percona.com/software/mysql-database/percona-xtradb-cluster. [Accessed 13-02-2024].

[18] 2024. PostgreSQL. https://www.postgresql.org/. [Accessed 13-02-2024].

[19] 2024. Redis. https://redis.io/. [Accessed 13-02-2024].

[20] 2024. Terraform. https://www.terraform.io/. [Accessed 8-4-2024].

[21] 2024. Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025. https://www.statista.com/statistics/871513/worldwide-data-created/. Accessed: 10-06-2024.

[22] Steve Adolph, Wendy Hall, and Philippe Kruchten. 2011. Using grounded theory to study the experience of software development. *Empirical Software Engineering* 16 (08 2011), 487–513. https://doi.org/10.1007/s10664-010-9152-6

[23] Mert Akdere, Ugur Cetintemel, Matteo Riondato, Eli Upfal, and Stan Zdonik. 2011. The Case for Predictive Database Systems: Opportunities and Challenges. 167–174.

[24] Francisco J. Baldan, Sergio Ramirez-Gallego, Christoph Bergmeir, Francisco Herrera, and Jose M. Benitez. 2018. A Forecasting Methodology for Workload Forecasting in Cloud Systems. *IEEE Transactions on Cloud Computing* 6, 4 (2018), 929–941. https://doi.org/10.1109/TCC.2016.2586064

[25] Titus Barik, Brittany Johnson, and Emerson Murphy-Hill. 2015. I heart hacker news: expanding qualitative research findings by analyzing social news websites. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering* (Bergamo, Italy) *(ESEC/FSE 2015)*. Association for Computing Machinery, New York, NY, USA, 882–885. https://doi.org/10.1145/2786805.2803200

[26] Surajit Chaudhuri and Vivek Narasayya. 1998. AutoAdmin "What-If" Index Analysis Utility. *SIGMOD Rec.* 27, 2 (jun 1998), 367–378. https://doi.org/10.1145/276305.276337

[27] Surajit Chaudhuri and Vivek Narasayya. 2007. Self-Tuning Database Systems: A Decade of Progress. *VLDB*, 3–14.

[28] Thomas Fritz and Gail Murphy. 2011. Determining Relevancy: How Software Developers Determine Relevant Information in Feeds. 1827–1830. https://doi.org/10.1145/1978942.1979206

[29] Archana Ganapathi, Harumi Kuno, Umeshwar Dayal, Janet L. Wiener, Armando Fox, Michael Jordan, and David Patterson. 2009. Predicting Multiple Metrics for Queries: Better Decisions Enabled by Machine Learning. In *2009 IEEE 25th International Conference on Data Engineering*. 592–603. https://doi.org/10.1109/ICDE.2009.130

[30] B.G. Glaser and A.L. Strauss. 1967. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine Transaction. https://books.google.fr/books?id=oUxEAQAAIAAJ

[31] Katarina Grolinger and Miriam Capretz. 2011. Autonomic Database Management: State of the Art and Future Trends. *Proceedings of the ISCA 27th International Conference on Computers and Their Applications, CATA 2012*.

[32] Sarra Habchi, Xavier Blanc, and Romain Rouvoy. 2018. On adopting linters to deal with performance concerns in android apps. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*. 6–16.

[33] S.E. Hove and Bente Anda. 2005. Experiences from Conducting Semi-structured Interviews in Empirical Software Engineering Research. *Proceedings - International Software Metrics Symposium* 2005, 10 pp.–. https://doi.org/10.1109/METRICS.2005.24

[34] Yi Hu and B. Panda. 2003. Identification of malicious transactions in database systems. In *Seventh International Database Engineering and Applications Symposium, 2003. Proceedings.* 329–335. https://doi.org/10.1109/IDEAS.2003.1214946

[35] Tim Kraska, Mohammad Alizadeh, Alex Beutel, Ed H. Chi, Jialin Ding, Ani Kristo, Guillaume Leclerc, Samuel Madden, Hongzi Mao, and Vikram Nathan. 2019. SageDB: A Learned Database System.

[36] Doyup Lee. 2017. Anomaly Detection in Multivariate Non-stationary Time Series for Automatic DBMS Diagnosis. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 412–419. https://doi.org/10.1109/ICMLA.2017.0-126

[37] Guoliang Li, Xuanhe Zhou, Shifu Li, and Bo Gao. 2019. QTune: A Query-Aware Database Tuning System with Deep Reinforcement Learning. *Proc. VLDB Endow.* 12, 12 (aug 2019), 2118–2130. https://doi.org/10.14778/3352063.3352129

[38] Sainan Li, Qilei Yin, Guoliang Li, Qi Li, Zhuotao Liu, and Jinwei Zhu. 2022. Unsupervised Contextual Anomaly Detection for Database Systems. In *Proceedings of the 2022 International Conference on Management of Data* (Philadelphia, PA, USA) *(SIGMOD '22)*. Association for Computing Machinery, New York, NY, USA, 788–802. https://doi.org/10.1145/3514221.3517861

[39] Lin Ma, Dana Van Aken, Ahmed Hefny, Gustavo Mezerhane, Andrew Pavlo, and Geoffrey Gordon. 2018. Query-based Workload Forecasting for Self-Driving Database Management Systems. *SIGMOD '18: Proceedings of the 2018 International Conference on Management of Data*, 631–645. https://doi.org/10.1145/3183713.3196908

[40] D. Narayanan, E. Thereska, and Anastasia Ailamaki. 2005. Continuous resource monitoring for self-predicting DBMS, Vol. 2005. 239– 248. https://doi.org/10.1109/MASCOTS.2005.21

[41] Daniel Oliver, Julianne Serovich, and Tina Mason. 2006. Constraints and Opportunities with Interview Transcription: Towards Reflection in Qualitative Research. *Social forces; a scientific medium of social study and interpretation* 84 (01 2006), 1273–1289. https://doi.org/10.1353/sof.2006.0023

[42] Zakaria Ournani, Romain Rouvoy, Pierre Rust, and Joel Penhoat. 2020. On Reducing the Energy Consumption of Software: From Hurdles to Requirements. In *Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)* (Bari, Italy) *(ESEM '20)*. Association for Computing Machinery, New York, NY, USA, Article 14, 12 pages. https://doi.org/10.1145/3382494.3410678

[43] Zakaria Ournani, Romain Rouvoy, Pierre Rust, and Joel Penhoat. 2020. On reducing the energy consumption of software: From hurdles to requirements. In *Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 1–12.

[44] Andrew Pavlo, Gustavo Angulo, Joy Arulraj, Haibin Lin, Jiexi Lin, Lin Ma, Prashanth Menon, Todd C Mowry, Matthew Perron, Ian Quah, et al. 2017. Self-Driving Database Management Systems.. In *CIDR*, Vol. 4. 1.

[45] Asmaa Sallam, Elisa Bertino, Syed Rafiul Hussain, David Landers, R. Michael Lefler, and Donald Steiner. 2017. DBSAFE—An Anomaly Detection System to Protect Databases From Exfiltration Attempts. *IEEE Systems Journal* 11, 2 (2017), 483–493. https://doi.org/10.1109/JSYST.2015.2487221

[46] Justin Smith, Brittany Johnson, Emerson Murphy-Hill, Bill Chu, and Heather Richter Lipford. 2019. How Developers Diagnose Potential Security Vulnerabilities with a Static Analysis Tool. *IEEE Transactions on Software Engineering* 45, 9 (2019), 877–897. https://doi.org/10.1109/TSE.2018.2810116

[47] Klaas-Jan Stol, Paul Ralph, and Brian Fitzgerald. 2016. Grounded Theory in Software Engineering Research: A Critical Review and Guidelines. https://doi.org/10.1145/2884781.2884833

[48] Adam Storm, Christian Garcia-Arellano, Sam Lightstone, Yixin Diao, and Maheswaran Surendra. 2006. Adaptive Self-tuning Memory in DB2. 1081–1092.

[49] Wenhu Tian, Patrick Martin, and Wendy Powley. 2003. Techniques for automatically sizing multiple buffer pools in DB2. 294–302. https://doi.org/10.1145/961322.961367

[50] Dana Van Aken, Andrew Pavlo, Geoffrey Gordon, and Bohan Zhang. 2017. Automatic Database Management System Tuning Through Large-scale Machine Learning. 1009–1024. https://doi.org/10.1145/3035918.3064029

[51] Konstantina Vasileiou, Julie Barnett, Susan Thorpe, and Terry Young. 2018. Characterising and justifying sample size sufficiency in interview-based studies: systematic analysis of qualitative health research over a 15-year period. *BMC medical research methodology* 18 (2018), 1–18.

[52] Junxiong Wang, Immanuel Trummer, and Debabrota Basu. 2021. UDO: Universal Database Optimization using Reinforcement Learning. *CoRR* abs/2104.01744 (2021). arXiv:2104.01744 https://arxiv.org/abs/2104.01744

[53] Ji Zhang, Yu Liu, Ke Zhou, Guoliang Li, Zhili Xiao, Bin Cheng, Jiashu Xing, Yangtao Wang, Tianheng Cheng, Li Liu, et al. 2019. An end-to-end automatic cloud database tuning system using deep reinforcement learning. In *Proceedings of the 2019 International Conference on Management of Data*. 415–432.

[54] Ji Zhang, Ke Zhou, Guoliang Li, Yu Liu, Ming Xie, Bin Cheng, and Jiashu Xing. 2021. CDBTune+: An efficient deep reinforcement learning-based automatic cloud database tuning system. *The VLDB Journal* 30 (11 2021). https://doi.org/10.1007/s00778-021-00670-9