Can Surrogate Keys Negatively Impact Data Quality?

Mathilde Marcy
Jean-Marc Petit
Vasile-Marian Scuturici
firstname.lastname@insa-lyon.fr
INSA Lyon, CNRS, UCBL, LIRIS, UMR5205
Villeurbanne, France

Jocelyn Bonjour
firstname.lastname@insa-lyon.fr
INSA Lyon, CNRS, CETHIL, UMR5008
Villeurbanne, France

Camille Fertel
Gerald Cavalier
firstname.lastname@cemafroid.fr
CEMAFROID
Fresnes, France

ABSTRACT

Surrogate keys are now extensively utilized by database designers to implement keys in SQL tables. They are straightforward, easy to understand, enable efficient access, and are often considered a sufficient guarantee of data integrity despite lacking any real-world semantic meaning. In spite of all their benefits, one might wonder whether surrogate keys can negatively impact data quality. IT developers who rely exclusively on surrogate keys when designing database schemas may be tempted to not encode natural keys, as they are perceived as complex to manage at the application level. In such settings, surrogate keys allow the presence of so-called *artificial unicity*, a complex form of redundancy that can be propagated through foreign keys, and other underlying data-quality issues. In the presence of artificial unicity, most data cleaning techniques, especially unsupervised, are likely to fail, making data preparation and analytics very challenging.

For relational databases implemented with surrogate keys but no natural keys, we developed RED2Hunt (RElational Databases REDundancy Hunting), a human-in-the-loop framework for identifying hidden redundancy and, if problems occur, clean the database. The framework was implemented on top of PostgreSQL within an eponym web-based platform to guide the expert through its application. In this paper, we present a demonstration of the RED2Hunt tool through three interactive scenarios on a polluted instance of the publicly available *Perfect Pet* database. During the demonstration, the visitor can take on one of two roles in the Perfect Pet database: a domain expert or a data scientist. As a domain expert, she will interact with RED2Hunt, for example to elicit natural keys, from simple yet very intuitive visualizations of tables' attributes. As a data scientist, she will explore two simple scenarios—executing SQL queries or applying learning models—on both the initial and cleaned databases to grasp the benefits of the approach.

PVLDB Reference Format:
Mathilde Marcy, Jean-Marc Petit, Vasile-Marian Scuturici, Jocelyn Bonjour, Camille Fertel, and Gerald Cavalier. Can Surrogate Keys Negatively Impact Data Quality?. PVLDB, 18(12): 5279 - 5282, 2025.
doi:10.14778/3750601.3750651

PVLDB Artifact Availability:

Proceedings of the VLDB Endowment, Vol. 18, No. 12 ISSN 2150-8097.
doi:10.14778/3750601.3750651

```
Microchip(id_microchip (SK), number, implant_date)
Doctor(id_doctor (SK), first_name, last_name, license_number,
start_date, end_date)
Animal(id_animal (SK), species, breed, name, id_microchip (SFK),
gender, dob, weight, food, hash_id, id_client (SFK))
Appointment(id_appointment (SK), id_animal (SFK), date, time,
main_reason, id_doctor (SFK))
Client(id_client (SK), id_animal (SFK), first_name, last_name, city,
phone_number)

                        FKs:
    Animal[id_microchip] ⊆ Michochip[id_microchip]
        Animal[id_client] ⊆ Client[id_client]
      Appointment[id_animal] ⊆ Animal[id_animal]
      Appointment[id_doctor] ⊆ Doctor[id_doctor]
        Client[id_animal] ⊆ Animal[id_animal]
```

Figure 1: *Perfect Pet*'s database

The source code, data, and/or other artifacts have been made available at https://github.com/mathildemarcy/perfect_pet.

1 INTRODUCTION

More than ever, small and medium enterprises (SMEs) are interested in valuing their data. However, 64% of them face challenges in doing so, and 74% have difficulty leveraging their investments in data [11]. Data quality remains a major obstacle to the spread of data analytics within these companies [2], as the exploitation of bad data carries a strategic and financial risk [10]. Redundancy in particular is a very common and significant issue found in operational data, causing underlying data quality issues such as inconsistency and inaccuracy.

Most data owned by such companies is stored into relational databases management systems (RDBMS) used to support the digitalization of their operations. The design of these databases' schemas is often entrusted to the IT professionals in charge of developing the operational applications, who commonly favor performance and gain over analytical constraints. They tend to make intensive use of surrogate keys and often do not encode natural keys, as they are perceived as restrictive and complex to manage, opening the door to redundancy. Consequently, the unicity enforced by surrogate keys might be artificial, referred to as *artificial unicity* [8].

Example 1.1. Let us consider a subset of a polluted instance of the publicly available Perfect Pet database which schema is described in Figure 1 [7]. All keys declared in the data dictionary are surrogate primary keys (suffixed with SK). Consequently, the five foreign keys (suffixed by SFK) are also surrogates. Table 1 includes an extract from two of its relations. *Perfect Pet*'s database suffers from artificial unicity, which can be quickly observed by looking at relations `Microchip` and `Animal` (tuples sharing the same colors

Table 1: Extract of the clinic's database

	id_animal	species	breed	name	id_microchip	gender	dob	weight	hash_id	id_client
1	2346	canine	saluki	Eleonor	2329	F	2018-12-15	19.49	m3skkj7273	2308
2	2496	canine	cocker	Coco	2479	M	-	11.37	z88he4r6b5	2458
3	3090	canine	saluki	eleonor	3073	F	2018-12-15	20.61	7frhju7qaa	3052
4	3343	canine	cocker	Coco	3326	M	-	10.34	59fr4ge5rm	3305
5	3870	canine	saluki	Eleonor	3853	F	2018-12-15	18.57	yvvtcimqbo	3832
6	5457	feline	siamese	Noora	5440	F	2022-02-01	4.08	pmugang8j8	5419
7	5482	feline	siamese	noura	5465	F	2022-02-24	3.81	z2mxqxg9j7	5444

(a) Animal

	id_microchip	number	implant_date
1	2329	832235208	2019-04-27
2	2479	470218622	2018-08-01
3	3073	832235208	2019-04-27
4	3326	470218622	2018-08-01
5	3853	832235208	2019-04-27
6	5440	481908508	2022-05-10
7	5465	481908508	2022-05-10

(b) `Microchip`

are in fact redundant). The artificial unicity contained in the surrogate key of `Microchip`, `id_microchip`, has been propagated to `Animal` through it's foreign key `id_microchip`, which should be the natural key of the relation.

Artificial unicity and other underlying data quality issues could accumulate over the years and go unnoticed as long as they do not impact the digital applications or the operations of the SME, until its analytical exploitation begins. Moreover, data quality is known to impact analytical products' accuracy and machine learning (ML) models' performance [3, 6], especially redundancy by altering the accuracy of simple statistics such as median, average, distribution, or frequency.

Example 1.2. Assume that the data scientist *Perfect Pet* has hired writes the following SQL query to get a dataset in order to learn a model predicting the number of appointments for animals:

```
SELECT id_microchip, species, breed, gender, COUNT(*) as nb_apt
FROM Animal an
JOIN Appointment ap ON an.id_animal = ap.id_animal
GROUP BY ap.id_animal
```

She will get 7 tuples due to the artificial unicity in the database, instead of 3, and the number of appointments will be inaccurate for each animal. Any model learned from this data would be biased. The data should be thoroughly cleaned before any analytical use.

However, when artificial unicity is spread through the keys-foreign keys join paths to the whole database, it prevents a straightforward detection and resolution of duplicates. As far as we know, this problem turns out to be new and extremely difficult to fix. Any attempt to remove artificial unicity on the answer set of a SQL query is likely to fail. We argue that artificial unicity must be removed at the source before any regular cleaning technique can be applied.

For relational databases implemented with surrogate keys but no natural keys, we developed RED2Hunt (RElational Databases REDundancy Hunting), a human-in-the-loop framework for identifying hidden redundancy and, if problems occur, generate clean, redundancy-free versions of these databases for analytical use [8]. The proposed framework was implemented on top of PostgreSQL within an eponym web-based platform designed to guide the expert through its entire workflow. Suppression of artificial unicity contained in a database, which is an essential prerequisite for its cleaning and the cornerstone of the framework, is achieved by leveraging specific information about the data and its structure, to be discovered by the domain expert with the help of visual tools provided within RED2Hunt.

In this paper, we present a demonstration of the RED2Hunt tool through three interactive scenarios on a polluted instance of the publicly available Perfect Pet database [7]. During the demonstration, the visitor can take on one of two roles in the *Perfect Pet* database: a domain expert or a data scientist. As a domain expert, she will interact with RED2Hunt, for example to elicit natural keys, from simple yet very intuitive visualizations of tables' attributes. As a data scientist, she will explore two simple scenarios—executing SQL queries or applying learning models—on both the initial and cleaned databases to grasp the benefits of the approach.

2 RED2HUNT

The RED2Hunt framework, illustrated in figure 2, is composed of three blocks described in detail in [8]. Each block was implemented within the eponym platform to offer a simple and intuitive interface to facilitate its application. The platform was developed in Python on top of PostgreSQL using the flask framework and relies on several packages such as pandas, psycopg2, matplotlib, networkx.

The implementation of each of these three blocks is presented in this section, highlighting the interaction with the domain expert. The platform includes an additional upstream block allowing the user to connect to their database, and to filter the relations that are of most interest for analytical purposes.

2.1 Block 1 - Elicitation of keys

When it exists, artificial unicity has to be removed from the database before any data cleaning can take place. Artificial unicity is defined by the interaction between surrogate and potential keys. Consequently, their identification is a prerequisite to its detection in a relation. The first block of RED2Hunt is dedicated to this task, for which we favored an expert-based approach to avoid the known pitfalls of profiling methods [9].

Relations are presented to the domain expert one by one in a specific order allowing a seamless familiarization with the process. For each relation examined, the expert is offered several built-in visualization options to facilitate discovery, among which: data samples, counterexamples of a potential natural key, and visualization of its Relation Redundancy Profile (RRP). RRP visualizations provide a comprehensive view of all attributes in a relation, each one characterized by different metrics, such as the numbers of unique, null, and repeated values, name, and category (key, foreign-key, other). Figure 3a presents the RRP of `Animal` before key elicitation.

Based on the RRP, the expert can easily identify a non-declared key (such as `hash_id` in `Animal`), and the entity represented by the relation, with its natural key (the relation's potential key). After identifying them, the expert has to declare the surrogate and natural

Figure 2: Overview of the RED2Hunt framework

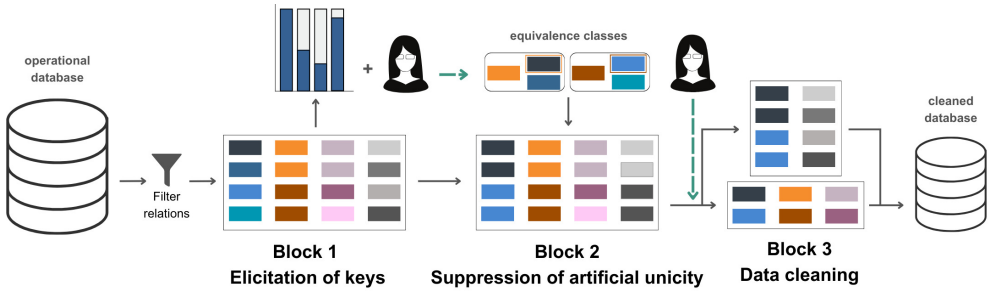
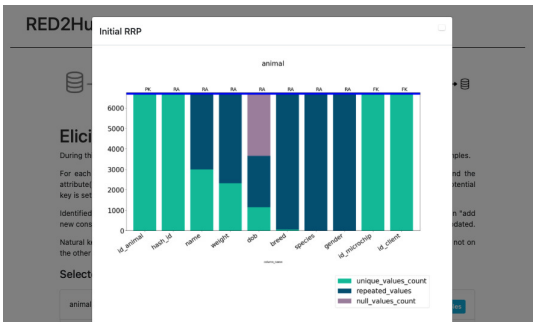
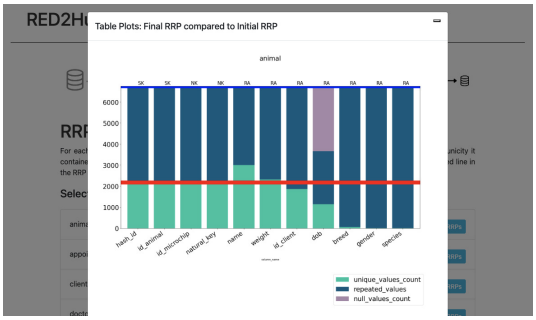


Figure 3: Visualization of the relation `Animal`'s RRP



(a) during block 1



(b) after block 2

keys in RED2Hunt to update the relations' RRPs and their visualizations. A clear correspondence between a relation and an entity type does not always exist in operational databases. In such relations, the potential key is defined as the collection of all attributes except the surrogate keys, instead of the natural key.

2.2 Block 2 - Artificial unicity suppression

This block is entirely executed in the background and does not require any interaction with the expert. In order not to modify the operational database and its schema, a copy is created on which key constraints are deactivated.

Artificial unicity is removed from a relation by unifying the equivalent values of its surrogate keys, and correcting data-quality issues in its potential key, wherever they exist. In practice, classes of equivalent potential keys are first formed through the application of any entity matching (EM) method [1], where the potential key serves as matching key. RED2Hunt currently offers three built-in EM methods: exact matching, clustering, and ML model training [4].

However, the presence of artificial unicity within a surrogate foreign key contained in a potential key will bias the formation of equivalence classes, no matter which EM method is used. It should therefore be removed beforehand by propagation of the referenced surrogate key's equivalence. Relations are thus processed in a specific order, based on the database's propagation graph, to guarantee the accuracy of the classes and efficiency of the process. The order is presented to the domain expert, along with the propagation graph, as illustrated in figure 4.

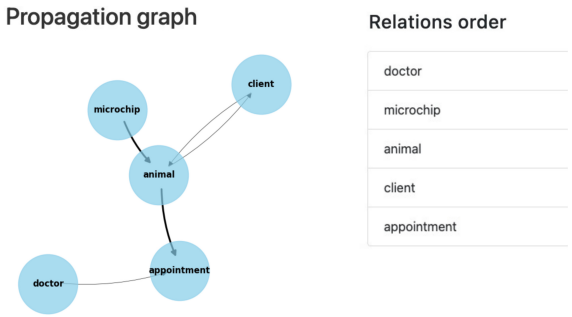


Figure 4: Propagation graph and relations order in RED2Hunt

In our Perfect Pet example, we first extract the equivalence classes of `Microchip` and suppress its artificial unicity, then propagate its equivalence values to `Animal`, as presented in table 2. Then, the same process is applied in cascade to other relations. RRPs are updated after complete removal of artificial unicity within the database. Comparison between original and updated RRPs allows a quick visual assessment of the artificial unicity levels the operational database is suffering from.

Figure 3b presents the updated RRP of `Animal`. The reduction in every surrogate key and surrogate foreign key carrying artificial unicity following its removal can easily be assessed by comparing this RRP with the final one. The red line represents the updated number of distinct values on its natural key `id_microchip` (which originally was equal to the size of the relation). The difference

Table 2: Extract of the clinic's database after removing artificial unicity from `Microchip`

	id_animal	species	breed	name	id_microchip	gender	dob	weight	hash_id	id_client
1	2346	canine	saluki	Eleonor	2329	F	2018-12-15	19.49	m3skkj7273	2308
2	2496	canine	cocker	Coco	2479	M	-	11.37	z88he4r6b5	2458
3	2346	canine	saluki	Eleonor	2329	F	2018-12-15	20.61	m3skkj7273	2308
4	2496	canine	cocker	Coco	2479	M	-	10.34	z88he4r6b5	2458
5	2346	canine	saluki	Eleonor	2329	F	2018-12-15	18.57	m3skkj7273	2308
6	5457	feline	siamese	Noora	5440	F	2022-02-01	4.08	pmugang8j8	5419
7	5457	feline	siamese	Noora	5440	F	2022-02-24	3.81	pmugang8j8	5419

(a) Animal

	id_microchip	number	implant_date
1	2329	832235208	2019-04-27
2	2479	470218622	2018-08-01
3	2329	832235208	2019-04-27
4	2479	470218622	2018-08-01
5	2329	832235208	2019-04-27
6	5440	481908508	2022-05-10
7	5440	481908508	2022-05-10

(b) `Microchip`

between the blue and red lines illustrates the true rate of artificial unicity (i.e. redundancy) contained in the relation.

2.3 Block 3 - Data cleaning

This block is devoted to cleaning the artificial unicity-free database by leveraging well-known techniques, relying on both automated methods and the involvement of a domain expert.

First, we verify for each non surrogate-key, non natural-key attribute A of every relation for which a natural key NK was identified if it induces some denormalization within the relation, relying both on the $g3$ value associated to functional dependency $NK \rightarrow A$ [5], and the domain expert's answer to the question: "Could this attribute, in real life, potentially accept several values and still characterize the same entity?". RED2Hunt offers the visualization of some counterexamples to support her answer. Relations are then normalized based on these verifications, remaining data inconsistencies are resolved by applying the well-known CHASE procedure, and finally remaining exact duplicates are suppressed. Following this step, we have a new normalized, cleaned database, on which integrity constraints are re-encoded.

3 DEMONSTRATION

RED2Hunt has been tested on real-life operational databases, its end-target. Because none of these databases could be used due to privacy constraints, and none of the publicly available databases disclosed for data quality suffer from artificial unicity, we created an open-source software to generate and pollute instances of the synthetic Perfect Pet database, and used it to generate a database instance for this demonstration [7]. The instance's generalized artificial unicity makes its cleaning and exploitation extremely challenging.

Three demonstration scenarios will be offered to the audience. In the first one, the audience will play the role of the domain expert during the cleaning process. They will use the RED2Hunt platform to navigate through the three blocks and will be encouraged to explore all the functionalities the tool offers. This demonstration's main goal is to display the facility with which domain experts are engaged throughout the workflow. Data visualization options throughout the process will also allow participants to gauge the transformation applied to the data.

In the other scenarios, the audience will act as the data scientist entrusted by Perfect Pet with the mission of utilizing their data. In the second one, the audience will have to answer very simple analytical questions from Perfect Pet's owner. They will be provided with predefined SQL queries that are easy to understand, and result sets on both the original data and the cleaned data. Since the two answer sets are quite different, we will have the opportunity to explain why and how RED2Hunt proceeds to get such results. In the last scenario, the audience will be asked to train a very basic ML model on two datasets: one extracted from the original database and the second from the cleaned database. The two models' performance will be assessed and compared based on well-known indicators. A platform offering simple options for data cleaning, exploration, visualization, and modelling will be made available to the visitors for these tasks. The goal of these demonstrations is to illustrate the benefits of using RED2Hunt to clean relational data prior to their extraction for analytical purposes.

ACKNOWLEDGMENTS

We thank our industrial partners for their involvement in developing and testing RED2Hunt, and CEMAFROID and ANRT for funding part of this research.

REFERENCES

[1] Peter Christen. 2012. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection.* Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-642-31164-2.

[2] Lisa Ehrlinger and Wolfram Wöß. 2022. A Survey of Data Quality Measurement and Monitoring Tools. *Frontiers in Big Data* 5 (2022), 850611.

[3] Bakhtiyar Doskenov Ga Young Lee, Lubna Alzamil and Arash Termehchy. 2021. A Survey on Data Cleaning Methods for Improved Machine Learning Model Performance. *arXiv preprint* arXiv:2109.07127 (2021).

[4] Forest Gregg and Derek Eder. 2025. Dedupe. https://github.com/dedupeio/dedupe. open-source software, GitHub repository. Accessed: June 25, 2025.

[5] Jyrki Kivinen and Heikki Mannila. 1995. Approximate Inference of Functional Dependencies from Relations. *Theoretical Computer Science* 149, 1 (1995), 129–149.

[6] Nina Ihde Andrea Nathansen Nele Noack Hendrik Patzlaff Felix Naumann Lukas Budach, Moritz Feuerpfeil and Hazar Harmouch. 2022. The Effects of Data Quality on Machine Learning Performance. *arXiv preprint* arXiv:2207.14529 (2022).

[7] Jean-Marc Petit Mathilde Marcy and Vasile-Marian Scuturici. 2025. Perfect Pet: Synthetic Database Generation with Artificial Unicity. https://github.com/mathildemarcy/perfect_pet. open-source software, GitHub repository. Accessed: July 14, 2025.

[8] Vasile-Marian Scuturici Jocelyn Bonjour Camille Fertel Mathilde Marcy, Jean-Marc Petit and Gerald Cavalier. 2025. RED2Hunt: an Actionable Framework for Cleaning Operational Databases with Surrogate Keys. *arXiv preprint* arXiv:2503.20593 (2025).

[9] Tova Milo and Dan Suciu. 1999. Efficient Discovery of Functional Dependencies and Armstrong Relations. *SIAM J. Comput.* 28, 4 (1999), 1237–1257.

[10] Thomas C. Redman. 2017. Seizing Opportunity in Data Quality. https://sloanreview.mit.edu/article/seizing-opportunity-in-data-quality/. (2017). Accessed: July 15, 2025.

[11] World Economic Forum. 2023. Data Unleashed: Empowering Small and Medium Enterprises (SMEs) for Innovation and Success. https://www3.weforum.org/docs/WEF_Data_Unleashed_Empowering_Small_and_Medium_Enterprises_(SMEs)_for_Innovation_and_Success_2023.pdf. Accessed: July 15, 2025.