Eco-friendly Database Space Saving using Proxy Attributes

Ahorro Sostenible de Espacio de Almacenamiento de Bases de Datos usando Atributos de Proxy

Nurul A. Emran[1], Noraswaliza Abdullah[1], Norharyati Harum [1], Amelia R. Ismail[2] , Azlin Nordin[2] and Ismael Caballero[3]

[1] *Department of Software Engineering, Fakulti Teknologi Maklumat dan Komunikasi,*
Universiti Teknikal Malaysia Melaka (UTeM), Malaysia
{nurulakmar, noraswaliza, norharyati}@utem.edu.my
[2] *Department of Computer Science, Kulliyyah of Information and Communication Technology,*
International Islamic University Malaysia
{amelia,azlinnordin}@iium.edu.my
[3]*Department of Software and Information Technologies, University of Castilla-La Mancha, Spain*
{Ismael.Caballero@uclm.es}

Abstract

Rapid data growth and inefficient data storage are two concerning issues that are becoming more and more important in green computing. The decision on the eco-friendly technology to use often relies on the amount of carbon footprint produced. Thus, it would be valuable to avoid inefficient electric power utilization by minimizing physical data storages to store large data volumes. This paper reported the implementation of proxy attributes to reduce space by optimizing the available database space through attributes substitution. We examine a set of proxies retrieved from the public databases regarding their space-saving and accuracy properties. The results indicated that useful proxies that can offer space-saving while maintaining accuracy are available. The findings contribute in understanding the practicality of proxies and their potential in database space-saving.

Keywords: eco-friendly, proxy, green computing, green data center, space-saving

Resumen

El rápido crecimiento de la cantidad de datos usados en diferentes aplicaciones, y el almacenamiento ineficiente de los datos en dichas aplicaciones están siendo dos factores cada vez más preocupantes para el Green Computing. Las decisiones sobre las tecnologías sostenibles a usarse se deberían tomar en la cantidad de la huella de carbono generada durante la ejecución de las aplicaciones. Por esta razón, es deseable optimizar minimizar el consumo energético de las operaciones de almacenamiento de datos, sobre todo en casos de grandes volúmenes de datos. En este artículo se presenta un informe del desempeño de atributos proxy para reducir el espacio necesario de almacenamiento de datos mediante la sustitución de atributos. Examinamos un conjunto de proxies recuperados de las bases de datos públicas con respecto a sus propiedades de precisión y ahorro de espacio. Los resultados indicaron que hay disponibles proxies útiles que pueden ahorrar espacio y mantener la precisión. Los hallazgos contribuyen a comprender la practicidad de los proxies y su potencial para ahorrar espacio en la base de datos.

Palabras claves: Sostenibilidad, proxy, Green Computing, centro de datos ecológico, ahorro de espacio.

1. Introduction

Green computing practice aims to optimize the utilization of Information Technology (IT), ensuring every activity involving IT deployment has environmental concerns in mind [1]. According to the Energy Star program launched by the Environmental Protection Agency (EPA), one of the initiatives of green computing is to minimize the electricity consumption of computers and their peripheral devices and to ensure their usage is eco-friendly [2].

Data centers in many organizations are the infrastructure that consists of networked computers and storage established for organizing, processing, storing, and disseminating massive data. As an infrastructure that deals with large amounts of data,

data centers consume a lot of electrical energy to ensure all the processes running smoothly. Consequently, it is undeniable that the data center contributes to carbon footprint and global warming [3],[4] which leads to proposals to measure its efficiency [5][6]. In dealing with the environmental issue, the green data centers (or green cloud) are proposed to decrease carbon footprint and operating costs (e.g., for cooling systems), where the efforts usually involve optimization of some parameters of the physical configuration of the data centers [6][7].

The rapid growth of data demands more storage allocation within the data centers. IDC (International Data Corporation) stated that within the next three years from 2020, there would be more than the data created over the past 30 years [8]. One of the data growth drivers is the COVID-19 pandemic, which caused a global increase in work-from-home staff and live video streaming.

It is also possible to witness an increasing rate of analytics (i.e big data) investments in labor and IT architecture with a total productivity growth effect of about 5.9% [9]. With this scenario, many organizations need to cope with the increasing storage demand paying special attention to the optimization of carbon footprints for data operations, specially storage. Even though the price for persistent data storage has dropped, and cloud technology has offered improved storage solutions, at the organizational level, the ability to manage massive data is crucial. Cloud storage might not be an attractive solution for cases where data are sensitive and confidential [2]. In this case, organizations need to manage their data within their own data centers. In this situation, adding more data servers is not a strategic option in dealing with the storage space issue. This last option leads to an undesirable increase in power consumption and CO2 emissions. In addition, as pointed in [10], organizations can gain financial advantage by their commitment to green efforts.

One characteristic of establishing a green data center is the efficient use of its data storage. In fact, optimizing the available storage space is one of the twelve strategies outlined by the EPA Energy Star for green data centers [11]. Nevertheless, in addition to the Energy Star's guideline, we can observe that other green guidelines such as [12] and [13] mainly cover hardware/physical modifications of the data center rather than the data management solution. Due to this limitation, little can we understand the potential of database space optimization in green computing.

Thus, the research questions of RQ.1 "How can database space storage be achieved through schema modification?" and, RQ.2 "How to evaluate the method if there is one?" motivate us to explore the topic further. Particularly, in this paper, we set to examine the use of proxy attributes in database space-saving.

In the next section, related work on the existing space-saving works will be presented. Section 3 consists of the methods used to evaluate the proxies; Section 4 covers the results and discussion. Finally, Section 5 concludes the research.

2. Related work

Research on database space optimization commonly focused on improving database performance. As a result, we can see the availability of tools and techniques provided by commercial data storage vendors of data compression tools (such as Oracle, Microsoft, SQL Server, and IBM DB2 [14],[15]) that have been a while in the market for quite some time.

At the relational database table level, data compression tools, for example, apply a repeated values removal technique to gain free space and therefore gaining query processing and overall database performance advantage [16]. The application of data compression can also be seen in networking and IoT research that deals with the problem of reducing the size of data within the network to improve data transmission speed [17],[18].

Data deduplication is an area of research that aims to remove duplicate records in the table to gain storage spaces [19][20]. Data deduplication is also a technique used in data cleaning [21][22]. Research in this field also involves finding ways to detect duplicates within the database before duplicates can be removed [23][24][25].

The idea behind data compression and data deduplication is to exploit overlaps (of values or records) within tables. Both techniques are performed at the level of the entire table. An important assumption behind these optimization techniques is that all columns can be used to optimize space. Because of this assumption, knowledge about the semantics of the applications (i.e., how the columns are used) is ignored. As a result, optimization is usually performed at the level of the entire table.

The most important lesson we have learned from the optimization methods is that space optimization methods that achieve space savings at both the schema level and the entire tables are limited. Moreover, space optimization techniques that consider application semantics are also limited. Due to these limitations, the data deduplication and data compression do not fully solve the storage space problem. For example, knowledge of how datasets are used for analyses must be considered for space optimization in data-intensive applications (such as in the field of microbial genomics). Furthermore, data compression requires a de-compression process that will redeem the original space allocated. Thus, the space problem is only solved temporarily.

As proposed in [26], proxy-based space-saving attempted to address the limitations of data compression and data deduplication methods. Proxy attributes are used to minimize the need for new physical data storage, where inefficient power consumption and CO2 emission can be avoided. Accordingly, attributes substitution is applied through database schema modification.

However, the information about the implementation of proxies and their usefulness has not been reported. Thus, in this paper, we seek to evaluate this method. The findings can contribute in designing a space optimization strategy where storage space is an issue.

In the next section, we will elaborate on the experiment conducted for proxies' evaluation after providing some background on the proxy-based approach.

3. The proxy-based space saving

In designing a space-saving approach, we examine the concept of "proxy" as stated in a previous study reported in [26]. In this previous study, the implementation of proxies has not been reported. From the lesson learnt in [26] we consider that there was room enough to keep on investigating the benefits of applying proxy-based space-saving for a greener storage. The space-saving will be achieved via database schema modification and by removing redundant data from the table. In this approach, free spaces are gained by deleting selected attributes from tables. Proxies are discovered within a dataset based on the relationships that are present within the dataset, such as functional dependency.

According to Papenbrock *et al.* (2015), functional dependency (FD) is a particular kind of relationship that is present among attributes in relational tables. An FD over a relation schema R is denoted by an expression $X \rightarrow A$ where $X \subseteq R$ and $A \in R$ [27]. X in the expression is the set of attributes known as 'determinant attributes' whose values can uniquely determine the values of A. The rule regarding FDs states that $X \rightarrow A$ is valid within relation r with schema R, given that for all pairs of tuples t, $u \in r$ we have:

$$T[X]=u[X]=>t[A]=u[A] \text{ where } A, X \subseteq R$$ [27]

Proxies can be discovered based on FD rules, and thus, based on FD, a proxy is defined as a member of determinant attributes, X.

To illustrate this idea (See Fig.1) suppose there are three columns in a Customer table of a restaurant database in Malaysia. The table consists of 100,000 customer records. The columns are *name, preference* and *race*. Suppose that column *race* is selected as the droppable attribute and its proxy attribute is column *preference*. Thus, the gross amount of space-saving is 100,000 rows. As column *race* is deleted, there is information loss in the Customer table.

To compensate data loss, a proxy map table is developed to keep the mapping between the values of the deleted attribute and the proxy's values. The proxy map needs to be smaller (in size) relative to the size of the dropped attributes so that space-saving can be achieved. Using a proxy map will require additional space where space-saving can only be gained if it is in its optimal size. In this paper, we implemented the proxy map in a relational table.

Customer table

name	preference	race
Ali	Fried rice	Malay
Ahmad	Fried rice	Malay
Azizi	Fried rice	Malay
Suzy	Beef Stew	Malay
Alice	Noodles	Chinese
Aminah	Fried rice	Malay
Tan	Garlic Soups	Chinese
Chuah	Garlic Soups	Chinese
Ravi	Curry	Indian
Kim	Roasted Duck	Chinese
Sally	Curry	Indian
Isma	Spicy soup	Malay
Kaur	Chicken Masala	India
…	…	…

Fig. 1 An example of schema modification by dropping the *race* column.

A data preparation step is required to select the proxy candidates before the proxy map can be built. In this approach, the proxy and droppable attributes will be determined by discovering the relationship of the attributes in the table. The functional dependency (FD) rule is applied for this purpose. Optimization of storage space can be gained when data redundancy in the table is removed, and a small proxy map table can be generated. With the use of a proxy map table, the redundancy of column values can be removed because it will map the unique values of data.

Proxy candidates are generated among the non-key attributes. This is because, even though key attributes guarantee substitution accuracy, the proxy map table generated will consume undesirable space as every key value will be mapped to the dropped attribute's value. This will reduce the value of the proxy.

The proxy map will be used during the query transformation process. The example of mapping in the proxy map table is shown in Fig. 2, where the values of the proxy attribute's values (e.g., column *preference*) are mapped to the values of the dropped attributes (e.g., column *race*). Note that, as duplicate value pairs (i.e Fried rice, Malay) are removed from the proxy map, there is a chance to gain space through the removal of column *race* from the *Customer* table.

preference	race
Fried rice	Malay
Beef Stew	Malay
Spicy soup	Malay
Noodles	Chinese
Garlic soup	Chinese
Roasted duck	Chinese
Curry	Indian
Chicken Masala	Indian
…	…

Fig. 2 An example of mapping in a proxy map in a relational table.

3.1. Experiment design

An experiment was set to test the hypothesis that proxies that contribute to space-saving are also accurate. Thus, the amount of space-saving and accuracy (of the transformed queries) are chosen as performance indicators of the proxies. The flow of the experiment is as shown in Fig. 3. The flow starts by setting the experiment hypothesis. Once the datasets are indentified, the process of downloading and cleaning the data sets begin. Details on the subsequent processes will be given in the next sub-sections.

3.1.1. Datasets description

We retrieved the Comprehensive Microbial Resource (CMR) datasets that consist of microbial bacterial genome types provided by the National Genomics Data Center[1]. The raw dataset consists of genome collections with annotations downloadable from the website[2]. Fig. 4 depicts the raw dataset downloaded from the database. A data cleaning process was performed to remove missing values and inconsistent formatting. As shown in Fig. 3, the raw dataset was also converted into a format compatible with the proxy discovery step. After this process, CMR datasets are stored as relational tables in the proxy database. These tables are regarded as original tables, as no attributes are being dropped. The size of these tables will be needed to measure space-saving.

CMR datasets consist of microbial bacterial genome types of data. Researchers use it to investigate the types of bacteria and species that come along with their different genome collections and annotations [28]. CMR datasets are scientific datasets that offer interesting relationships among attributes. This makes CMR datasets fit for our purpose.

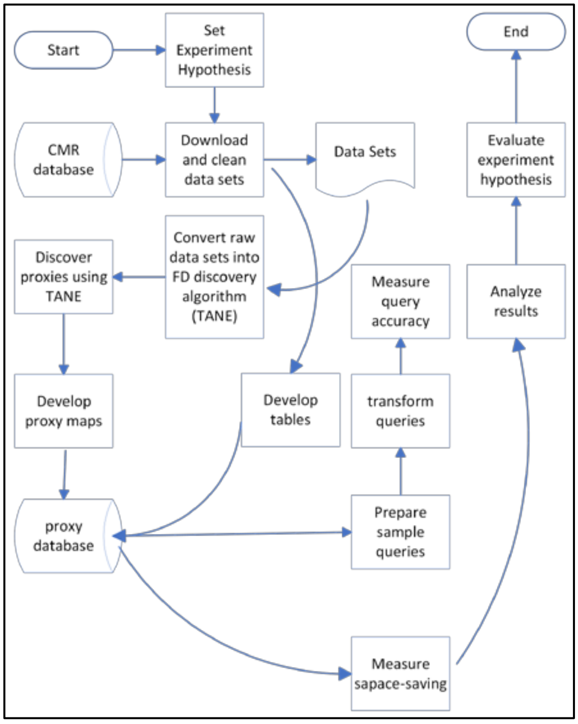
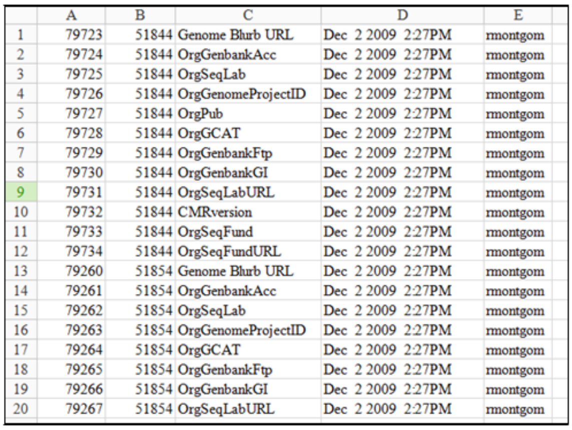


Fig.3: Experiment flow for proxy evaluation



	A	B	C	D	E
1	79723	51844	Genome Blurb URL	Dec 2 2009 2:27PM	rmontgom
2	79724	51844	OrgGenbankAcc	Dec 2 2009 2:27PM	rmontgom
3	79725	51844	OrgSeqLab	Dec 2 2009 2:27PM	rmontgom
4	79726	51844	OrgGenomeProjectID	Dec 2 2009 2:27PM	rmontgom
5	79727	51844	OrgPub	Dec 2 2009 2:27PM	rmontgom
6	79728	51844	OrgGCAT	Dec 2 2009 2:27PM	rmontgom
7	79729	51844	OrgGenbankFtp	Dec 2 2009 2:27PM	rmontgom
8	79730	51844	OrgGenbankGI	Dec 2 2009 2:27PM	rmontgom
9	79731	51844	OrgSeqLabURL	Dec 2 2009 2:27PM	rmontgom
10	79732	51844	CMRversion	Dec 2 2009 2:27PM	rmontgom
11	79733	51844	OrgSeqFund	Dec 2 2009 2:27PM	rmontgom
12	79734	51844	OrgSeqFundURL	Dec 2 2009 2:27PM	rmontgom
13	79260	51854	Genome Blurb URL	Dec 2 2009 2:27PM	rmontgom
14	79261	51854	OrgGenbankAcc	Dec 2 2009 2:27PM	rmontgom
15	79262	51854	OrgSeqLab	Dec 2 2009 2:27PM	rmontgom
16	79263	51854	OrgGenomeProjectID	Dec 2 2009 2:27PM	rmontgom
17	79264	51854	OrgGCAT	Dec 2 2009 2:27PM	rmontgom
18	79265	51854	OrgGenbankFtp	Dec 2 2009 2:27PM	rmontgom
19	79266	51854	OrgGenbankGI	Dec 2 2009 2:27PM	rmontgom
20	79267	51854	OrgSeqLabURL	Dec 2 2009 2:27PM	rmontgom

Fig. 4: A sample of a raw dataset from CMR database in Excel spreadsheet

CMR datasets consist of 24 tables of data. In the experiment, three datasets with varying sizes were used as samples which are:

i. Taxon:

The taxon dataset initially consists of 14 attributes. After inspection, we removed six attributes that consist of more than 50% missing values. The total rows of this table are about 723 rows. The Taxon table is the smallest table in this study. The schema of taxon table is:

Taxon(id, taxon_id, kingdom, genus, species, i_rank_1, i_rank_2, short_name)

ii. Bug_attribute:

[1] https://ngdc.cncb.ac.cn/

[2] https://download.cncb.ac.cn/Genome/RefSeq/

The Bug_attribute dataset initially consists of six attributes. We applied the same rule for Taxon dataset where an attribute with more than 50% missing values is removed. The total rows of this table are about 10,165 rows. This table is of medium size.

The schema for Bug_attribute table is:

Bug_attribute(id, db_data_id, att_type, assignby, datebug).

iii. Role_link:

This dataset initially consists of five attributes with 1048576 rows. As an attribute (*role_id*) exhibits about 10% of missing values, we removed only the affected rows rather than the entire attribute. The total number of the remaining rows in this table is 934,206 rows. This table is the largest in this study. The schema of *Role_link* table is:

Role_link(id, locus, role_id, assignby,datestamp)

The second dataset was retrieved from UC Irvine Machine Learning Repository, which is also known as UCI[3]. UCI currently maintains datasets for machine learning as community services. In particular, we used an Adult dataset from UCI that initially consisted of 14 attributes and 48842 rows[4]. After deleting the column and rows that have missing values, the Adult dataset used in the study has 13 columns and 30162 rows. The dataset was retrieved from a census database which consists of many attributes and a large number of records. Thus, it is suitable to be used in the experiment. The schema of *adult* table is:

Adult(age, workclass,fnlwgt, education, educationNum,maritalStatus,occupation,relationship ,race,sex,hours,country,class)

3.1.2. Experiment flow

To determine the droppable attributes and the proxy attributes, we need to discover the relationship among the attributes in the tables. Several functional dependency discovery algorithms/tools are available, such as TANE algorithm [27], FDTool (which is written in Python) [29], FD_Mine [30] and Discovery Functional Dependencies (DFD) [31]. Thorsten et al. (2015) compared seven functional dependency algorithms, especially regarding the scalability of the algorithms [32].

As TANE provides easy installation that suits our purpose, we use TANE to search for droppable attributes and proxies in this study. TANE algorithm is implemented on partitions that represent the collection of tuple values for each specified attribute in a given relational database [27]. It separates the tuple in such a way that the redundancy of tuple values for each attribute may be easily discovered and contributed.

The value of tuple of each attribute is computed level by level, and redundancies of each item in the attributes are defined in the appropriate subsets of each characteristic through the redundancies of each item in the specific attribute. This was designed to discover the functional dependencies from the databases. TANE is useful in discovering functional and approximation dependencies from relations. The method is based on determining dependencies from partitions and searching for them on a quality basis.

TANE's source code can be downloaded from the researchers' website[5]. TANE algorithms can only work in the LINUX operating system environment. Since Windows operating system was used in the experiment, we used Kali Linux to run the LINUX command in the Windows platform. Kali Linux can be downloaded from the website[6]. Once all the setup files were downloaded, the installation of TANE was performed by following two simple steps: the compilation of TANE folder and running the 'make' command from *src* folder. Using TANE, functional dependency scores (called g3 error) were used to determine the relationship strength among the attributes. g3 is the standard error measure for functional dependency. Thus, low g3 value is desirable, and proxy candidates will undergo a filtering process where proxies were chosen among the non-key attributes with the lowest possible g3 error (less than 0.1%).

Once we finalized the proxy candidates, we develop the proxy map tables and store them in the proxy database. In calculating the final space-saving, the size of the proxy maps must be considered from the gross space-saving value (from deleting the column alone).

For example, given that column *b* is the proxy for column *c*, the following are the schema of:

- the original table, *A*: *A(a,b,c),*
- the table after column deletion *A'*: *A'(a,b)*
- the proxy map table *P_b*: *P_b(b,c)*

Suppose that the size of table *A* is $|A|$, the size of table *A'* is $|A'|$, and the size of proxy map *P_b* table is $|P\_b|$ the amount of space-saving gained (in percentage) is calculated as follows:

$$\text{space saving } (\%)=((|A|\text{-}(|A'|+|P\_b|))/|A|)*100$$

(Eq. 1)

[3] https://archive.ics.uci.edu/ml/index.php

[4] https://archive.ics.uci.edu/ml/datasets/adult

[5] http://www.cs.helsinki.fi/research/fdk/datamining/tane/

[6] https://www.kali.org/kali-on-windows-app.

The proxy map table is created by issuing the following query:

```
CREATE TABLE P_b AS SELECT DISTINCT b,c
FROM A;
```

In this paper, the space value was measured in kilobytes units. Based on this formula, we know that, as long as the size of the proxy map is smaller than the size of the deleted column, space gain can be offered at the table level.

For example, using the sysdba role, and the name of the database is *proxyDB*, the size of the tables stored in a database (i.e MySql) is retrieved by issuing the following SQL command:

```
SELECT table_name AS ‘Table’,
ROUND((data_length + index_length) / 1024) AS
‘Size (KB)’
FROM information_schema.tables
WHERE table_schema = "proxyDb";
```

3.2. Query transformation accuracy

Before we can measure the accuracy of the proxies, we need to perform the query transformation process, as shown in Fig. 3. Queries submitted against the table with the deleted column need to undergo a query transformation process. The proxy used to substitute the deleted column must be present in the transformed query predicate. The substituted values are retrieved from the proxy map. Using the same schema provided earlier, suppose that the query before transformation *q*, is defined as a simple selection query as follows:

q: SELECT *c* from *A* WHERE $c=V_1$;

With the presence of proxy, *q* will be rewritten to *q’* as:

q’: SELECT *b* from *A’* WHERE b IN

(SELECT b FROM P_b WHERE $c=V_1$);

The accuracy of a proxy is determined by the number of overlapping results between the queries. Thus, the accuracy of a transformed query is calculated as:

$$\text{accuracy }(\%)=(|Q\cap Q'|/|Q\cup Q'|)*100 \quad \text{(Eq. 2)}$$

where *Q* is the result set of *q* and *Q’* is the result set of *q’*.

In the experiment, 30 random queries were used for each proxy under study. Table 1 shows an example query for the tables.

The average query accuracy was used as a measure of a proxy’s accuracy. The results of space-saving and accuracy of the proxies were analyzed before we could evaluate the experiment hypothesis.

Table 1: Example of query transformations

Proxy	**Original query, *q***	**Transformed query, *q’***
Table: *Taxon* *genus*→ *i_rank_2*	SELECT id, kingdom,genus, species FROM taxon WHERE i_rank_1 = 'Acaryochloris';	SELECT id, kingdom,genus,species FROM taxon’ WHERE genus IN (SELECT GENUS FROM proxy_genus WHERE i_rank_1 ='Acaryochloris');
Table: *Bug_attribute* *assignby*→ *datebug*	SELECT att_type, datebug FROM bug_attribute WHERE assignby = 'rmontgom';	SELECT att_type, datebug FROM bug_attribute’ WHERE datebug IN (SELECT datebug FROM proxy_assignby where assignby = 'rmontgom');
Table: *Role_link* *locus*→ *assignby*	SELECT idrole, locus, role_id FROM role_link WHERE assignby = 'autoannotate';	SELECT idrole, locus, role_id FROM role_link’ WHERE locus IN (SELECT locus FROM proxy_locus WHERE assignby = 'autoAnnotate');
Table: *Adult* *education*→ *educationNum*	SELECT age,flnwgt FROM adult WHERE educationNum = 7;	SELECT age,flnwgt FROM adult’ WHERE education IN(SELECT education from proxy_education WHERE educationNum = 7);

4. Results and discussion

4.1. Results for CMR datasets

The results of the proxy discovery step using TANE algorithm show that several proxy candidates are present, with varying g3 values for CMR dataset as shown in Table 2. The Taxon table has ten proxy candidates, while *Bug_attribute* and *Role_link* tables both with eight proxy candidates.

As a part of the filtering process, all key attributes are excluded from the final proxy candidates list. As described earlier, the decision is made due to the foreseen cost of the proxy map table size imposed by these attributes. Attributes with g3 value of more than 0.10% are also excluded from the candidate list as we chose to be less tolerant with weak proxies. We also excluded composite attributes from the list as we can also foresee the potential cost these attributes will cause on the size of the proxy map tables.

Table 2: The initial list of proxy candidates for CMR datasets

Table	Proxy	g3 error (%)
Taxon	id→ taxon_id	0.00 (key)
	id→ kingdom	0.00 (key)
	id→ genus	0.00 (key)
	id→ species	0.00 (key)
	id→ i_rank_1	0.00 (key)
	id→ i_rank_2	0.00 (key)
	id→ short_name	0.00 (key)
	Species, i_rank_1→ genus	0.04
	species→ i_rank_1	0.07
	genus→ i_rank_2	0.01
Bug_attribute	id→ db_data_id	0.00 (key)
	id→ att_type	0.00 (key)
	id→ assignby	0.00 (key)
	id→ datebug	0.00 (key)
	assignby→ datebug	0.00
	db_data_id, att_type→ id	0.06
	db_data_id, att_type→ assignby	0.06
	db_data_id, att_type→ datebug	0.06
Role_link	id→ locus	0.00 (key)
	id→ role_id	0.00 (key)
	id→ assignby	0.00 (key)
	id→ datestamp	0.00 (key)
	locus→ assignby	0.00
	locus, role_id →id	0.00
	locus, role_id →assignby	0.00
	locus, role_id →datestamp	0.00

Table 3 shows the final four proxy candidates (labeled as P1 to P4) for CMR dataset. *assignby* attribute (P3) from the *Bug_attribute* table exhibits the lowest g3 value with 0.00%.

Table 3: The final proxy candidates for CMR datasets

Table	Proxy	g3 error (%)
Taxon	P1:species→ i_rank_1	0.07
Taxon	P2:genus→ i_rank_2	0.01
Bug_attribute	P3:assignby→ datebug	0.00
Role_link	P4:locus→ assignby	0.02

The final list of proxy candidates in CMR dataset led to the creation of proxy map tables with the following schema:

- *Proxy_species(species, i_rank_1,)*
- *Proxy_genus(genus, i_rank_2)*
- *Proxy_assignby(assignby, datebug)*
- *Proxy_locus(locus, assignby)*

The following are the schema of the tables with the deleted columns for CMR datasets:

- *Taxon1(id, taxon_id, kingdom, genus, species, i_rank_2,short_name)*
- *Taxon2(id, taxon_id, kingdom, genus, species, i_rank_1,short_name)*
- *Bug_attribute(id, db_data_id, att_type, assignby)*
- *Role_link(id, locus, role_id, datestamp)*

Table 4 shows the scores of space savings and average accuracy by proxies of CMR datasets. P4 exhibits the highest amount of space-saving. This is followed by P3 with 13% space-saving.

As observed, the average error is low (less than 6%), with P3 offers highly accurate queries with 13% of space-saving. The only proxy that does not offer space-saving is P1, with -0.12% space-saving and 2.46 average error. Thus, P1 is not valuable for space-saving as the negative scores indicated more space will be acquired instead of space-saving.

Table 4: Space-saving and average accuracy results for CMR datasets

Table Name	Proxy	Space-saving (%)	Average Error (%)
Taxon	P1:species→ i_rank_1	-0.12	2.46
Taxon	P2:genus→ i_rank_2	8	5.18
Bug_attribute	P3:assignby→ datebug	13	0.00
Role_link	P4:locus→ assignby	21	2.46

4.2. Results for UCI dataset

Several proxy candidates are also present, with varying g3 for UCI dataset as shown in Table 5. For the UCI dataset, the Adult table has 46 proxy candidates. For simplicity, composite proxy candidates in the UCI dataset with more than three attributes are excluded from the table.

Table 6 shows the final six proxy candidates (labeled as P5 to P10) for the UCI dataset. Similar to P3, proxies *education* (P5) and *educationNum* (P6) from *Adult* table exhibit the lowest g3 value with 0.00%. These attributes are the non-key attributes that behave like a key attribute with no g3 error. The highest g3 value considered in this study 0.10% that belongs to P7, P8 and P9.

The following are the proxy map tables schema for the UCI dataset:

- *Proxy_education1(education,educationNum)*
- *Proxy_educationNum1(educationNum,education)*
- *Proxy_education2(education,country)*
- *Proxy_educationNum2(educationNum,country)*
- *Proxy_maritalStatus(maritalStatus,country)*
- *Proxy_maritalStatus(race,country)*

The following are the schema of the tables with the deleted columns:

- *Adult1(id, workclass,fnlwgt, education,maritalStatus,occupation,relationship,race,sex,hours,country,class)*
- *Adult2(id, workclass,fnlwgt, educationNum,maritalStatus,occupation,relationship,race,sex,hours,country,class)*
- *Adult3(id, workclass,fnlwgt, education, educationNum,maritalStatus,occupation,relationship,race,sex,hours,class)*

Table 5: The initial list of proxy candidates for UCI dataset

Table	Proxy	g3 error(%)
Adult	fnlwgt→ age	0.00 (key)
	fnlwgt→ workclass	0.00 (key)
	fnlwgt→ education	0.00 (key)
	fnlwgt→ educationNum	0.00 (key)
	fnlwgt→ maritalStatus	0.00 (key)
	fnlwgt→ occupation	0.00 (key)
	fnlwgt→ relationship	0.00 (key)
	fnlwgt→ race	0.00 (key)
	fnlwgt→ sex	0.00 (key)
	fnlwgt→ hours	0.00 (key)
	fnlwgt→ country	0.00 (key)
	fnlwgt→ class	0.00 (key)
	education→ educationNum	0.00
	educationNum→ education	0.00
	education→ country	0.10
	educationNum→ country	0.10
	maritalStatus→ country	0.10
	race→ country	0.09
	hours→ country	0.10
	age,occupation→ workclass	0.07
	age,occupation→ fnlwgt	0.07
	age,occupation→	0.07
	age,occupation→ education	0.07
	age,occupation→ educationNum	0.07
	age,occupation→ relationship	0.07
	age,occupation→ race	0.07
	age,occupation→ hours	0.07
	age,occupation→ country	0.07
	age,occupation→ class	0.07
	age,workclass→ hours	0.07
	age,education→ race	0.08
	age,education→class	0.08
	age,educationNum→race	0.08
	age,relationship→marital Status	0.09
	age, maritalStatus →sex	0.09
	age, relationship →sex	0.06
	age, relationship →country	0.09
	age,hours→race	0.10
	age,hours→sex	0.08
	age,class→country	0.08
	workclass,occupation→country	0.10
	workclass,class→country	0.10
	education,hours→race	0.09
	educationNum,hours→race	0.09
	occupation,relationship→sex	0.09
	relationship,hours→race	0.10

Table 6: The final proxy candidates for UCI datasets

Table	Proxy	g3 error (%)
Adult	P5:education→ educationNum	0.00
	P6:educationNum→ education	0.00
	P7:education→ country	0.10
	P8:educationNum→ country	0.10
	P9:maritalStatus→ country	0.10
	P10:race→ country	0.09

Note that, table *Adult3* can be used by proxies P7, P8, P9 and P10 as these proxies share the same dropped attribute, which is *country*.

Table 7 shows the scores of space savings and average accuracy by the proxies from the UCI dataset. The highest space-saving for Adult table is 8.3% by P9 and P10.

Table 7: Space-saving and average accuracy results of UCI dataset

Table Name	Proxy	Space-saving (%)	Average Error (%)
Adult	P5:education→ educationNum	0.6	0.00
Adult	P6:educationNum→ education	5.7	0.00
Adult	P7:education→ country	7.6	6.92
Adult	P8:educationNum→ country	7.6	6.92
Adult	P9:maritalStatus→ country	8.3	14.12
Adult	P10:race→ country	8.3	26.97

P5 and P6 from Adult dataset are the highly accurate proxies with 0.6 and 5.7 space-saving, respectively. P10 is the proxy with the highest average error, with 26.97%.

4.3. Discussion of results

The findings revealed that there is no solid indicator to say if the g3 error relates to the query errors. The situation can be explained by the coverage of the query predicates on the values that might (or might not) violate the functional dependency rule between the attributes. Thus, we can have a situation where proxies (notably P1 and P4) whose average error is the same do not have the same g3 values; proxy with the highest average error is not with the highest g3 value (see P2). Nevertheless, a consistent behavior can be observed for P3, P5 and P7 (that have no g3 error) with their query accuracy. Thus, one must be cautious to solely relies on g3 values (other than 0%) to judge a proxy's accuracy.

Fig. 5 further illustrates the results as a selection space within a Scatter plot graph for the proxies under consideration. The most desirable proxies are those in the top-left quadrant of the graph that offers high space-saving and low average error. Given this visual, one can decide on the adoption of proxies at a database table level. For example, given the results, as shown in Fig. 5, proxies with less than 10% average error that offer space-saving might be accepted. In this case, space-saving benefits might outweigh the slight errors that the proxies have.

The acceptability of proxies can be determined by whether a stringent threshold is set on accuracy and space-saving characteristics. The most useful proxies are certainly P3, P5 and P6 which does not only offer space-saving but are also accurate. This desirable property of proxy can be observed in an attribute that behaves like a key attribute (in terms of uniqueness), but there is no one-to-one mapping between its value and the deleted column. With one-to-many value mapping, the size of the proxy map can be small enough to allow space contribution. This means, a situation where each proxy value maps to exactly one deleted column's value, and each deleted column's value maps to many proxy values will always be beneficial in terms of accuracy and space-saving.

From the selection space, one can easily rule out proxies that come from the bottom-right quadrant, such as P9 and P10, and the one with no space-saving such as P1.

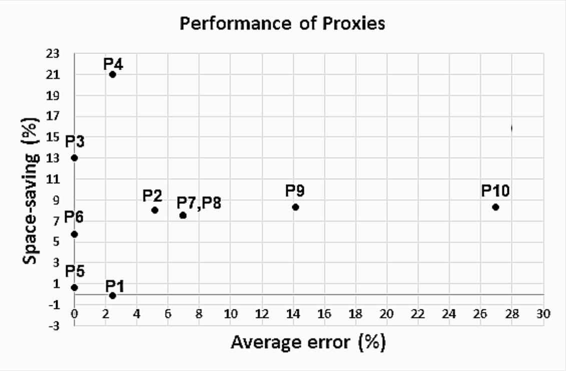


Fig. 5 Space-saving and average error of proxies

The findings also indicated that good proxies come from big tables with more than 10,000 rows. Proxies (P1 and P2) from a small table like *Taxon* (with only 723 rows) can offer small space-saving (or none), and their average errors are also relatively high.

The implementation of proxies in this study leads to the following observations:

i. Proxy-based space-saving is a lossy method provided that the proxies under consideration are from those that are only able to comply with the approximate functional dependency (g3 with more than zero). In this situation, one might tolerate a small amount of error in the queries. Nevertheless, this is not the case for proxies that fully comply with the functional dependency rule, as no information loss will occur during the query transformation.

ii. A pre-processing step is required to determine the feasibility of proxies for space-saving solutions. Extra space to store the temporary tables and proxy maps is also needed. This is the up-front cost that must be considered by the practitioners before implementing space-saving through proxies. Nevertheless, one can estimate the size of the proxy maps by issuing a query to get the amount of distinct proxy-deleted column value pairs. The accuracy of a proxy can also be estimated by determining whether it behaves like a key attribute that can uniquely identifies the deleted column's value.

iii. Query transformation is additional processing required in the proxy-based approach. Whether the cost of query processing will outweigh the space-saving benefits to reduce the carbon footprints is an open problem. In the best case scenario where the proxy map only consists of one row of the proxy-deleted column pair, query transformation is very minimal. This is because the subquery in *q'* only returns a value.

iv. The amount of duplicate pairs in the proxy tables is the key for space-saving. Datasets with such characteristics will get the benefits of proxies. In the best case scenario, we only need to store one row in the proxy map table, as the duplicate pair occupies the entire original table. For example, (*Fried rice, Malay)* is the only pair of values that occupies *preference* and *race* columns in the Customer table.

v. Maintenance of proxies is required to reflect the changes in the original tables. Frequent updates will cause a maintenance burden. Unless the value pairs inside the proxy map are less likely to change frequently, one must reconsider the adoption of proxies. Historical or archival datasets are the cases where value changes are minimal and thus low proxy maintenance is needed.

5. Conclusions

In conclusion, the results presented in this paper show the implementation of proxy-based storage-saving. The proxies have been evaluated in terms of space-saving and query accuracy within the scope datasets under study. The findings support the hypothesis of proxies that contribute to space-saving are also accurate. Nevertheless, caution must be given on selecting proxy candidates, especially on the g3, before further evaluation. In addition, the up-front and maintenance cost of proxies are among the issues that one needs to consider in proxy's adoption. The findings presented in this paper are limited to the datasets under study. However, the proxies will still be useful to meet the space-saving need with similar dataset characteristics. The findings contribute towards understanding the potential of proxies and their usefulness in database space optimization strategy where storage space is a constraint. To enhance the eco-friendly characteristic of this approach, the cost of query transformation will be considered in the cost-benefit analysis of proxies in our future work. Finally, the adoption of proxies depends on one's acceptability of the space-saving and the accuracy trade-offs. The findings contribute to an understanding of an alternative for existing storage space optimization methods.

Competing interests

The authors have declared that no competing interests exist.

Authors' contribution

All the authors in this document participated in the development of and successful completion of the research. All authors read and approved the final manuscript.

Acknowledgements

We would like to thank the Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka (UTeM). This work is under the UTeM research grant PJP/2020/FTMK/PP/S01771.

References

[1] B. Anthony, M. Abdul Majid, and A. Romli, "A Descriptive Study towards Green Computing Practice Application for Data Centers in IT Based Industries," *MATEC Web Conf.*, vol. 150, pp. 1–8, 2018.

[2] A. Sabban, "Introductory Chapter: Green Computing Technologies and Industry in 2021," in *Green Computing Technologies and Computing Industry in 2021*, 2021, pp. 1–16.

[3] R. R. Schmidt, E. E. Cruz, and M. K. Iyengar, "Challenges of data center thermal management," *IBM J. Res. Dev.*, vol. 49, no. 4–5, pp. 709–723, 2005.

[4] N. A. Ali and M. Abu-Elkheir, "Data management for the Internet of Things: Green directions," *2012 IEEE Globecom Work. GC Wkshps 2012*, pp. 386–390, 2012.

[5] J. Yuventi and R. Mehdizadeh, "A critical analysis of Power Usage Effectiveness and its use in communicating data center energy consumption," *Energy Build.*, vol. 64, pp. 90–94, Sep. 2013.

[6] D. Mukherjee, S. Roy, R. Bose, and D. Ghosh, "A Practical Approach to Measure Data Centre Efficiency Usage Effectiveness," 2022.

[7] R. Rahmani, I. Moser, and A. L. Cricenti, "Modelling and optimisation of microgrid configuration for green data centres: A metaheuristic approach," *Futur. Gener. Comput. Syst.*, vol. 108, pp. 742–750, 2020.

[8] M. Shirer and J. Rydning, "IDC's Global DataSphere Forecast Shows Continued Steady Growth in the Creation and Consumption of Data," *International Data Corporation (IDC),* 2020. https://www.idc.com/getdoc.jsp?containerId=prUS46286020 (accessed Aug. 13, 2021).

[9] J. Bughin, "Big data, Big bang?," *J. Big Data*, vol. 3, no. 1, p. 2, Dec. 2016, doi: 10.1186/s40537-015-0014-3.

[10] J. F. Molina-Azorín, E. Claver-Cortés, M. D. López-Gamero, and J. J. Tarí, "Green management and financial performance: A literature review," *Manag. Decis.*, vol. 47, no. 7, pp. 1080–1100, 2009.

[11] EPA Energy Star, "Top 12 Ways to Decrease the Energy Consumption of Your Data Centre," 2021. https://www.energystar.gov/buildings/tools-and-resources/top-12-ways-decrease-energy-consumption-your-data-center (accessed Aug. 13, 2021).

[12] S. Greenberg and M. Herrlin, "Small Data Centers, Big Energy Savings: An Introduction for Owners and Operators, FINAL REPORT," 2017.

[13] E. Ayanoglu, "Energy Efficiency in Data Centers | IEEE Communications Society," *IEEE ComSoc Technical Committees Newsletter*, 2019. https://www.comsoc.org/publications/tcn/2019-nov/energy-efficiency-data-centers (accessed Aug. 13, 2021).

[14] Oracle Corporation, "Oracle Advanced Compression Proof-of-Concept (POC) Insights and Best Practices," 2018. http://www.oracle.com/technetwork/databas (accessed Aug. 14, 2021).

[15] S. Alen, "Comparison on DB2 10.1 Vs SQL Server 2012 Vs Oracle 11g R2 latest features to suite SAP Products," 2013. http://scn.sap.com/docs/DOC-45542 (accessed Aug. 14, 2021).

[16] S. Aghav, "Database compression techniques for performance optimization," in *ICCET 2010 - 2010 International Conference on Computer Engineering and Technology, Proceedings*, 2010, vol. 6.

[17] T. Kim, N. S. Artan, J. Viventi, and H. J. Chao, "Spatiotemporal compression for efficient storage and transmission of high-resolution electrocorticography data," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2012.

[18] S. Nosratian, M. Moradkhani, and M. B. Tavakoli, "Hybrid data compression using fuzzy logic and huffman coding in secure iot," *Iran. J. Fuzzy Syst.*, vol. 18, no. 1, pp. 101–116, 2021.

[19] Y. Wang, M. Miao, J. Wang, and X. Zhang, "Secure deduplication with efficient user revocation in cloud storage," *Comput. Stand. Interfaces*, vol. 78, 2021.

[20] W. Tian, R. Li, C. Z. Xu, and Z. Xu, "Sed-Dedup: An efficient secure deduplication system with data modifications," *Concurr. Comput. Pract. Exp.*, vol. 33, no. 15, 2021.

[21] W. Lup Low, M. Li Lee, and T. Wang Ling, "A knowledge-based approach for duplicate elimination in data cleaning," *Inf. Syst.*, 2001.

[22] S. M. Randall, A. M. Ferrante, J. H. Boyd, and J. B. Semmens, "The effect of data cleaning on record linkage quality," *BMC Med. Informatics Decis. Mak. 2013 131*, vol. 13, no. 1, pp. 1–10, Jun. 2013.

[23] A. Ali, N. A. Emran, and S. A. Asmai, "Missing Values Compensation in Duplicates Detection Using Hot Deck," *J. Big Data*, vol. 8, no. 112, pp. 1–19, 2021.

[24] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: a survey," *{IEEE} Trans. Knowl. Data Eng.*, vol. 19, no. 1, pp. 1–16, 2007.

[25] G. Beskales, M. A. Soliman, I. F. Ilyas, and S. Ben-David, "Modeling and Querying Possible Repairs in Duplicate Detection," *Publ. Very Large Database Endow.*, vol. 2, no. 1, pp. 598–609, 2009.

[26] N. A. Emran, N. Abdullah, and M. N. M. Isa, "Storage Space Optimisation for Green Data Center," in *Procedia Engineering*, 2013, vol. 53, pp. 483–490.

[27] Y. Huhtala, J. Kärkkäinen, P. Porkka, and H. Toivonen, "TANE: An Efficient Algorithm for Discovering Functional and Approximate Dependencies," *Comput. J.*, vol. 42, no. 2, pp. 100–111, 1999.

[28] V. M. Markowitz, "Microbial Genome Data Resources.," *Curr. Opin. Biotechnol.*, vol. 18, no. 3, pp. 267–72, Jun. 2007.

[29] M. Buranosky, E. Stellnberger, E. Pfaff, D. Diaz-Sanchez, and C. Ward-Caviness, "FDTool: a Python application to mine for functional dependencies and candidate keys in tabular data," *F1000Research*, 2019.

[30] H. Yao, H. J. Hamilton, and C. J. Butz, "FD mine: Discovering functional dependencies in a database using equivalences," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2002.

[31] Z. Abedjan, P. Schulze, and F. Naumann, “DFD: Efficient functional dependency discovery,” in *CIKM 2014 - Proceedings of the 2014 ACM International Conference on Information and Knowledge Management*, 2014.

[32] T. Papenbrock *et al.*, “Functional dependency discovery: An experimental evaluation of seven algorithms,” in *Proceedings of the VLDB Endowment*, vol. 8, no. 10, 2015, pp. 1082–1093.

Citation: N. A. Emran, N. Abdullah, N. Harum, A. R. Ismail, A. Nordin, I. Caballero. *Eco-friendly Database Space Saving using Proxy Attributes.* Journal of Computer Science & Technology, vol. 22, no. 1, pp. 58-68, 2022.

DOI: 10.24215/16666038.22.e04

Received: September 22, 2021 **Accepted:** January 24, 2022.