

# Flexible Materials Properties Management System as a Basis for Data-centric Systems in Inorganic Materials Science

Victor Dudarev<sup>1</sup>[0000-0001-7243-9096], Nadezhda Kiselyova<sup>2</sup>[0000-0002-3583-0704] and Alfred Ludwig<sup>1</sup>[0000-0003-2802-6774]

<sup>1</sup> Ruhr-University Bochum, Bochum, 44801, Germany

<sup>2</sup>Baikov Institute of Metallurgy and Materials Science the Russian Academy of Sciences, Moscow, 119334, Russia

victor.dudarev@rub.de

kis@imet.ac.ru

alfred.ludwig@rub.de

**Abstract.** Efficient harvesting of research data with proper data management and data utilization is essential for scientific progress in materials science. On the one hand, the continuous production of huge amounts of data from measuring devices and the lack of industry standards for the data format for storage and processing necessitates the development of reliable information systems capable of organizing, analyzing, and using diverse materials data. This article focuses on the development of Internet-accessible data management systems for inorganic materials science, addressing typical domain requirements. These systems enable researchers to store diverse tabular data and different property types to support compatibility with wide a variety of research data. At the same time, typical data structures to address chemical systems, compositions, and their modifications are available out of the box together with a flexible graphical user interface inspired by the periodic table. The key emphasis is on data integrity, ensuring accurate tracking and preservation of materials data, and interoperability by providing data type templates and rich import/export capabilities. A case study with bandgap data highlights the system's effectiveness in materials properties storage, effective data modification, and fast user-friendly search to support data-driven decision-making. Considering the significance of a strong data infrastructure for flexible materials properties management we offer an open-source solution to facilitate and accelerate similar data-centric projects in inorganic materials science by encouraging you to deploy our solution in building your own customizable materials research infrastructure you could rely on.

**Keywords:** materials science data infrastructure, materials data management system, inorganic materials properties, materials information search.

## 1 Introduction

In modern materials science we could observe intensive data grow which brings us to the problem of efficient management and utilization of research data. Solution of this task is crucial not only for scientific progress in this domain, but also in many industrial applications for effective data sharing and information exchange for data-driven decision-making tasks related to modern materials [1].

With the continuous generation and accumulation of big data volumes in materials science, the need for robust information systems that can effectively organize, analyze, and leverage this valuable information has become paramount [2, 3]. Our focus lies specifically in the development of applicable in inorganic materials science data management systems, which are accessible via the Internet. Our extensive experience, accumulated over the years, has been demonstrated in the successful development and utilization of several information systems on inorganic materials properties [4]. With a **deep understanding** of the complexities of data in this field, especially regarding modern high-throughput techniques in materials research [5] we have been developing an open-source pioneering solution that encompasses the essential functionality required for modern materials science data management.

Since materials science core data must be strictly defined using strong types and all the problem domain **constraints** preserved, we rely on classical **relational** database structures, known for their sustainability and maturity (e.g., see ACID principles) [6]. Therefore, all general top-level chemical objects (systems, compositions) are described in terms of relations to ensure the utmost data integrity available at data level in computer science, which is a crucial aspect in accuracy of tracking and preserving materials science data. We hope that our commitment to development of transparent open-source solutions together with data integrity for core chemistry data structures will enable researchers to confidently rely on systems based on our solution for their critical data management needs, thereby enabling accurate tracking and preservation of materials data.

Despite that this article aims to highlight the significance of a strong-types data at the top level of materials infrastructure, we certainly need **flexibility** for defining materials properties data types as the foundation for flexible materials properties management in materials science. And here once again we are capitalizing on the power of relational databases to decompose arbitrary tabular research data into a set of predefined type-aware properties tables, defining algorithms for data transformations to preserve flexibility, required for diverse materials property types, including combined from tables with different column types: integers, floats, and strings. These three type primitives for table columns allow us to take care of data structures unknown at system design time and then reconstruct them on-line and facilitate search on different data types of columns comprising a property value.

Materials management system must facilitate not only storage but efficient search for persisted materials data. To this extend we rely on periodic table-based interface that empowers researchers to interact with a system for data search. Search form is automatically adjustable depending on stored data types and their corresponding properties, which makes system flexible out of the box. This innovative approach in

materials science management systems simplifies the search process and enhances user experience, facilitating rapid access to relevant materials data, providing a robust solution for managing the ever-expanding array of materials properties.

To illustrate functional capabilities of developed system in real-world scenarios we provide case study by transferring materials properties data from existing bandgap data source and thus leveraging search facilities and delivering better user experience, that was available in initial proprietary solution.

## 2 Problem Domain and Key Requirements Definition

Materials science orientation brings to the system challenges concerned with representation of chemical entities from the problem domain. Considering vast variety of requirements gathered from diversity of problem subdomains it's clear that the general-purpose materials data management system should be capable of describing basic entities of the problem domain. Hence, we consider the very top (basic) level of materials description (chemical system, compositions, modifications) to be a part of functionality available out of the box. Any further details require developing corresponding data models and workflows to support them are subject for discussion and further implementation (system extension). Implemented three level hierarchy of chemical entities is described [7] as a model comprising:

- chemical system – a set of chemical elements that built up a material denoted by the symbol  $S$ . If each chemical element is represented by a unique symbol or identifier  $e_i$ , then mathematically, the chemical system  $S$  is a set  $\{e_1, e_2, \dots, e_n\}$ , where  $e_i$  represents a chemical element in the chemical system (qualitative composition).
- composition – a set of chemical elements with their concentration (if solid solution is considered) or particular content (in a case of single phase) that built up a material. Let  $n_i$  represents the quantity or coefficient associated with a particular element  $e_i$ , then composition, designated  $C$ , is a set of pairs:  $C = \{(e_1, n_1), (e_2, n_2), \dots, (e_n, n_n)\}$  (quantitation composition).
- modification – a particular crystal or amorphous structure realization of a composition of other subtype / subdivision, expressed as a string (further details) or in a more formal way as lattice parameters, space groups and atomic coordinates (matrices, vectors, and equations).

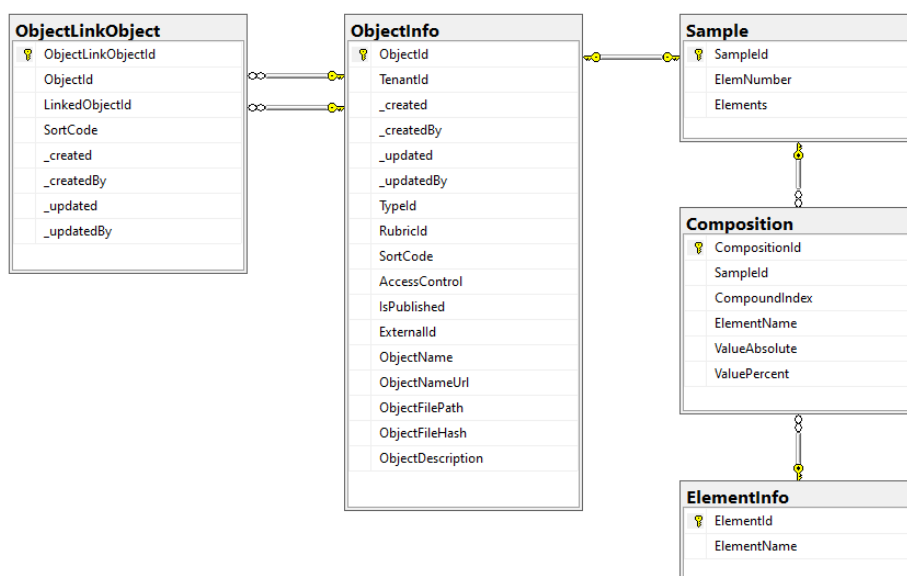
If we omit typical functional requirements (user registration and authentication, API and user interface for CRUD operations, import/export functionality, data sharing and notifications) focusing on essential data-oriented functionality for material data management we conclude with necessity to provide facilities to support for chemical entities within above-defined hierarchy and arbitrary materials data types, e.g., properties values, tables.

### 3 Key data structures to address materials data management tasks

Since data integrity is our primary concern we rely on relational databases, designed with intention to keep data in as strict a format as possible, but at the same time ensuring capabilities for flexible extension in terms of materials properties description.

#### 3.1 Chemical entities

Every object that is stored within materials management system holds its record within **ObjectInfo** table, that encompasses all mandatory metadata for every object type such as creation and modification datetime stamps and corresponding user references (`_*` columns), reference to a particular object type (`TypeId`), defined in an external table, reference to a container for object (`RubricId`), that allows to structure all objects within a project tree, sort code to define a certain order in list of objects (`SortCode`), object's accessibility: public/protected/private (`AccessControl`); obligatory object name (`ObjectName`) (see fig. 1).



**Fig. 1.** SQL Server diagram of objects storage structure within materials data management system.

In terms of data every chemical entity is an object and thus it must have a record in **ObjectInfo**, but apart from that it is at least a chemical system, implying that we know chemical elements set, which is to be stored in **Elements** column of a **Sample** table as an alphabetically ordered sequence of elements separated with hyphen sign and starting and ending with hyphen also, e.g., “-Li-Nb-O-” or “-As-Ga-” (additionally number of

chemical elements in a system is stored (ElemNumber) to optimize queries for chemical systems). Denormalization for chemical system is used here to optimize search speed [8].

According to definition, compositions additionally contain quantity for every element either in terms of absolute (ValueAbsolute) or relative (ValuePercent) values. So, to describe chemical composition in addition to ObjectInfo and Sample tables one should feed several rows to **Composition** table, according to number of unique chemical elements. It's worth mentioning that ValueAbsolute is always calculated to enable that sum of elements within a composition is 100% and to make search more flexible. **ElementInfo** on the scheme is an auxiliary table to ensure element designations correctness (unknown ElementName is rejected).

ObjectLinkObject table is worth mentioning since it serves as linking opportunity between objects enabling to establish a dependency graph if one is required. By default, link is interpreted as a directed relation (arc, directed edge), implying that reverse relation should have opposite meaning, but in general link semantic could be defined on linked object types basis.

### 3.2 Object's properties database structure

As it was shown in previous section with chemical compositions or systems, additional information on object can be obtained from other tables, but it requires additional programming to support corresponding data type. In most of the cases it's enough to have an opportunity to add several additional values for object as a set of key-value pairs, respecting corresponding value's type (float, integer, string).

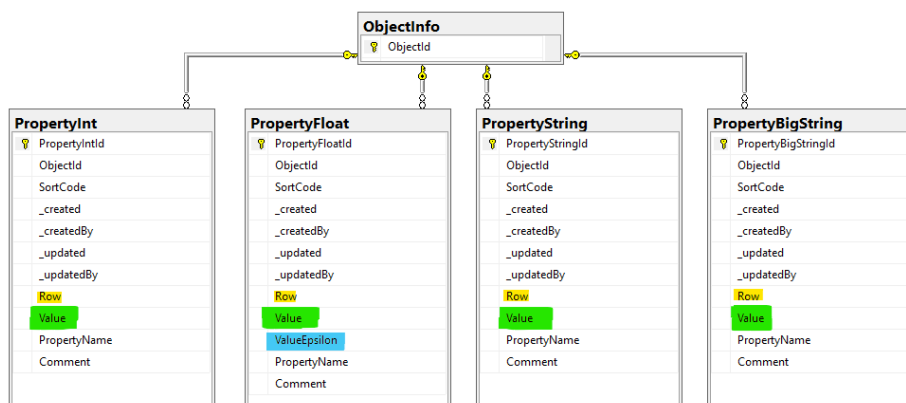


Fig. 2. Extended properties to describe objects.

To provide a proper storage for different types of values and considering specifics of storage in SQL Server for string of different size we designed four tables **Property\*** with obvious designation, see Fig. 2. Every table has similar structure, the only difference is in **Value** column data type and, in addition, **PropertyFloat** table contains optional **ValueEpsilon** field to specify measurement error, if possible.

Every property record (PropertyName, Value) references appropriate object by specifying its ObjectId identifier and some common data regarding creation and modification timestamps (underscore columns) and provides, by manipulating SortCode value, showing properties in desired order by SortCode ascending. Comment field is an optional one and could be used to attach some remarks to the value of a property.

However, the way more interesting use case we have when it's required to represent not just a set of properties, but a table with rows and columns. To support this use case in Properties\* tables it is required to use **Row** column, which is *null* by default, for non-tabular properties. If it's required to represent tabular data, then table is decomposed into a number of cells and every cell is represented as a property with name of the column and particular row value. This enables us to decompose tables of any structure (sparse tables are packed reasonably well, since empty cells could be omitted) and provide strong type check for every column according to selected property type (see Fig. 3).

Modification	CrystalSystem	SpaceGroup	Temperature	E	IsCalculated	Referenceld	Comment
α	Monoclinic	P2 <sub>1</sub> /c	300	0.87	0	581	Optical absorption spectroscopy. Direct transition. Thin film, single crystal
Row=1							
β	Cubic	Im3(-)m	77	1.23	0	581	Photoconduction. Thin film, polycrystalline
Row=2							
β	Cubic	Im3(-)m	296	1.03	0	581	Photoconduction. Thin film, polycrystalline
Row=3							
String x3			Float x2		Int x2		String

Fig. 3. Extended properties for description of tabular data (Ag<sub>2</sub>S bandgap).

### 3.3 Properties Templates and Import/Export features

Although materials management system offers user interface to perform CRUD-operations on object's properties on individual property value basis and from formal perspective functionality is covered, but if you have to deal with a plenty of object properties, then this could be inefficient and time-consuming task and therefore could cause problems. To overcome this, we introduced properties templates and properties import/export from/to Excel.

Properties templates is a mechanism of arbitrary table definition that could be applied to a particular object data type. Using template user should define structure of a table as an ordered (SortCode is used) sequence of column names and types. Once property template is defined for a type, user could download Excel table template to work with properties. Thus, it's easy to fill desired table data in Excel document and import them to the system for selected object. Or if you need some data modifications in table properties you could export properties in Excel and upload adjusted version

after modification. We extensively tested this option even for relatively large tables, e.g. properties from table with 16 columns and 468 rows (about 7.5K properties) were updated in several seconds.

#### 4 Use case: data import from bandgap database

Described data structures are to be proved by real life data to make sure that they are flexible to deal with real materials data coming from existing data source in relatively significant volumes and provide enough expressiveness to tailor all data details from external sources. To test developed materials data management system, we decided to import data from previously developed information system on bandgap of inorganic compounds <https://bg.imet-db.ru> [9]. Basically, we required to transfer data on inorganic compounds, corresponding bandgap values (which depend on compound crystal structure, temperature, measurements method, etc.) that were extracted from literature sources, that also good to have imported, since these are essential references to discover details.

Since literature reference is a common data type for data management system, we also introduced additional data table, called **Reference**, that extends standard object by meaningful publication's data (title, journal, year, volume, pages, DOI, etc.). So, the task was to map bandgap database schema to the schema of materials management system, and it was solved with the following key points: compounds were mapped to Composition data type (ObjectInfo+Sample+Composition tables); literature references – to Reference data type (ObjectInfo+Reference tables); table data on bandgap values for a particular composition – to properties (Property\* tables) and connections between literature reference and composition were established through ObjectLinkObject table. All data (about 5K compounds, 2K literature references and 10K bandgap values) were transferred by running developed SQL-scripts.

#### 5 Search Features in Material Data Management System

Data representation is important, but it makes no sense if data are not searchable and easily accessible. This means both that stored chemical entities are to be easily searchable by means of intuitive user interface, based on the periodic table and other common input forms, and all custom-defined properties should be also available for search with the help of dynamically adjustable search form (see Fig. 4 – search for compositions with bandgap in range [2; 2.1] eV in Ga-As system with 50% Ga contents).

In general, out of the box search works for: chemical system, composition, object type, phrase in object's name or description, properties values, person created record, creation date. For every object type user could select property value to search for (property names are dynamically fetched for autocomplete list) and adjust search form by adding more criteria for property search.

Valuable feature on search is a persistent URL, that reflects search criteria, therefore one could easily provide a link to share with collaborators (e.g. link <https://demo.mdi.ruhr-uni-bochum.de/search/?system=As->

Ga&typeid=8&Gapctmin=50&Gapctmax=50&pr0name=E%3Csub%3Eg%3C%2Fsub%3E&pr0type=Float&pr0min=2&pr0max=2.1&prcnt=1 provides permanent link to search result from Fig. 4). To test user interface, you could visit <https://demo.mdi.ruhr-uni-bochum.de> without necessity to register. In general, current graphical interface offers sufficient functionality and user experience that satisfies requirements for data search offered by specialized solutions, for example original <https://bg.imet-db.ru>.

Search system As-Ga and Ga in [50%; 50%] and Eg in [2; 2.1] and type is Composition

Chosen elements: As-Ga [clear]

Absolute  ≤ N ≤  or Percentage  ≤ N % ≤

Additional filters

Object type: Composition Search phrase: text in Name or Description

Property type: Float Name: Eg Property Value: 2 ≤ value ≤ 2.1

Created by: from: mm/dd/yyyy till: mm/dd/yyyy

Id	Type	Name / Description	Document	Created	Person	Access
2288	Composition	GaAs <sub>0.5</sub> P <sub>0.5</sub> GaAs <sub>0.3</sub> P <sub>0.7</sub>		2023-02-01	Victor Dudarev	Public

Fig. 4. User interface (search for bandgap in Ga-As in [2; 2.1] eV range).

## 6 Conclusion

In current paper we touched only the core parts of materials management system being developed as an open-source project [<https://gitlab.ruhr-uni-bochum.de/vic/infproject>] to facilitate deploying and usage of data infrastructure for inorganic materials science domain. Although the system is still under development, all core functions are already on their places. We struggle to grow project as extensible solution without plans for breaking changes in future. Through this article, we encourage researchers to establish their data-centric material infrastructure and unlock the full potential of materials science through efficient research data management. We believe that the system discussed, in its current state, is ready to support data-based materials research and can be easily adjusted to fulfill specific requirements.

A lot of work is still to be done in future. Additional efforts to support **specific** data types and use cases are to be undertaken within current INF project, so we could expect



more use cases and other useful scenarios to address materials research requirements, including API introduction to facilitate data exchange with external systems (materials repository, ML and data analysis tools) and measurement devices, establishing and monitoring materials research pipelines in collaborative research centers.

This research was financially supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Project-ID 388390466-TRR 247 (subproject INF). The study was carried out as a part of the state assignment (project nos. 075-01176-23-00).

## References

1. Klingner, C., Denker, M. et al. (2023) Research Data Management and Data Sharing for Reproducible Research—Results of a Community Survey of the German National Research Data Infrastructure Initiative Neuroscience. *eNeuro* 7 February 2023, 10 (2) ENEURO.0215-22.2023; DOI: <https://doi.org/10.1523/ENEURO.0215-22.2023>.
2. Herres-Pawlis, S., Liermann, J.C. and Koepler, O. (2020). Research Data in Chemistry – Results of the first NFDI4Chem Community Survey. *Z. Anorg. Allg. Chem.*, 646: 1748-1757. DOI: <https://doi.org/10.1002/zaac.202000339>.
3. Junkes, H., Oppermann, P., Schlögl, R., Trunschke, A., Krieger, M., & Weber, H. (2021). FAIRmat-a Consortium of the German Research-Data Infrastructure (NFDI). ICALEPCS2021. DOI:10.18429/JACoW-ICALEPCS2021-WEBL01
4. Kiselyova, N., Dudarev, V. (2005). The Distributed System of Databases on Properties of Inorganic Substances and Materials. *Int. Journal “Information Theories & Applications”*, vol. 12, number 3, 2005. pp. 219-224.
5. Banko, L., Ludwig, A. (2020) Fast-Track to Research Data Management in Experimental Material Science—Setting the Ground for Research Group Level Materials Digitalization. *ACS Comb. Sci.* 2020, 22, 8, 401–409. DOI: <https://doi.org/10.1021/acscombsci.0c00057>.
6. Nance, C., Losser, T., Iype, R., Harmon, G. (2013). NoSQL vs RDBMS – Why there is room for both. *SAIS 2013 Proceedings*. 27. <https://aisel.aisnet.org/sais2013/27>.
7. Kiselyova, N., Dudarev, V. (2017). Creating Inorganic Chemistry Data Infrastructure for Materials Science Specialists. *Communications in Computer and Information Science*. v.706. p. 222-236. ISSN 1865-0929, DOI: <https://doi.org/10.1007/978-3-319-57135-5>.
8. Sanders, G., Shin, S. Denormalization effects on performance of RDBMS. *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, Maui, HI, USA, 2001, pp. 9 pp.-, DOI: <https://doi.org/10.1109/HICSS.2001.926306>.
9. Kiselyova, N., Dudarev, V., Korzhuev, M. (2016). Database on bandgaps of inorganic substances and materials. *Inorganic Materials: Applied Research*. 2016. V. 7. №1. P.34-39. DOI: <https://doi.org/10.1134/S2075113316010093>.

## Responses to reviewers' comments

### Reviewer 1

**Remark 1:** The notations of links between tables in the database schema (Fig. 1) are not standard and should be explained.

**Answer:** On Fig.1 Microsoft SQL Server diagram was used (which is now added to the figure caption). All the links depicted are one-to-many relations.

**Remark 2:** It seems that the terms 'compound' and 'composition' are related to the same entity. It is better to use a single term throughout the paper.

**Answer:** Corrected (in section 2 now we use only all "composition" term, which is more relevant).

**Remark 3:** It is not clear why 'ElementName' is used in the 'Composition' table for indexing rather than 'ElementId'.

**Answer:** In general, all identifiers (\*Id attributes) are integers (4 bytes), at the same time 'ElementName' is varchar(2) (1-2 bytes for 1-2 characters + 2 bytes for string length). This means that effectively they are the same from storage perspective, but from the performance one using 'ElementName' in Composition table eliminates one extra table join, which makes the system more efficient (no join with ElementInfo table required). In other words, ElementInfo table serves as a constraint for element names in the Composition table.

**Remark 4:** In fact, the 'ValuePercent' value in the 'Composition' table may denote mass, volume, or mole fraction. It should be specified.

**Answer:** It can be anything from above mentioned (from the storage perspective they are the same), depending on the particular domain / tenant / object type.

**Remark 5:** The main question is the ambiguous choice of the quantity and dimension names in the 'PropertyName' field. As shown in Fig. 4, the user must enter the correct property name for searching the database, which is tricky without a catalog of the quantities and their dimensions. It is still not clear if this catalog is available to the user.

**Answer:** Correct point. Search form after selecting a particular object type provides drop-down list with available properties and autocomplete experience for user (that simplifies the search process). A number of properties and their types designed for a particular object type can be provided as a template withing RDMS for any object type (this functionality is briefly mentioned in section 3.3).

### Reviewer 2

**Remark 1:** It would be good to know how the TRR 247 infrastructure might connect to work done within the NFDI (e.g. NFDI4Chem, FAIRmat, NFDIMatWerk).

**Answer:** Can't comment on that since we are not aware of any standardized infrastructure within any of three above-mentioned consortia (but we'd love to know about them if they are).

**Remark 2:** Another discussion that could be of interest would be how the three-level hierarchy would relate to the description of chemical entities by other approaches, such as TUCAN (doi:10.1186/s13321-022-00640-5) or the EMMO.

**Answer:** We believe that currently three-level hierarchy can't describe organic compounds well enough to make it practically applicable. Anyway, any interschema relation should be meaningful to solve some goal. Unless the purpose is exactly specified there is no subject for consideration (any model is just an abstraction to solve a particular goal, information model as well).

**Remark 3:** Furthermore, if the authors foresee that every time they exchange data with other systems, a new 1:1 mapping needs to be created, or whether there are some plans and ideas for semantic interoperability.

**Answer:** Not yet. Data integration, especially in case of semantic heterogeneity of materials data sources to be consolidated is a totally separate challenge (that, we believe, has no general solution).

### Reviewer 3

**Remark 1:** On the other hand, results obtained from numerical simulations are now very important as a source of material's information, and repositories such as the NOMAD project have been created; what are your thoughts on the linkage with such computational resources?

**Answer:** Fully agreed. Every data source should be accessible (FAIR) and preferably expose its data via API. Linkage of some common data with heterogeneous Web Applications is totally possible and first successful attempts to implement it were undertaken 15 years ago in IMET RAS consolidating materials databases with NIMS (Dudarev V.A., Kiselyova N.N., Xu Y., Yamazaki M. Virtual integration of the Russian and Japanese databases on properties of inorganic substances and materials // MITS 2009. Symposium on Materials Database. National Institute for Materials Science (NIMS). Materials Database Station (MDBS). 2009. p. 37-48).

**Remark 2:** Also, regarding the interface with Excel, there are some problems with Excel in handling numerical data, such as the number of significant digits, how you can solve this fault?

**Answer:** We have not faced this problem. We could imagine it with some long numbers, but in principle any number can be represented as a text string while import / export operations, then data should not suffer from changes and roundings.

### Reviewer 4

**Remark 1:** The authors should compare their system with competitors, especially using periodic table-based interface (I can mention a couple from the first page of Google search: [Zakutayev, A., Wunder, N., Schwarting, M. et al. An open experimental database for exploring inorganic materials. *Sci Data* 5, 180053 (2018). <https://doi.org/10.1038/sdata.2018.53>] [Inorganic Materials Database for Exploring the Nature of Material. Yibin Xu, Masayoshi Yamazaki and Pierre Villars. 2011 The Japan

Society of Applied Physics. Japanese Journal of Applied Physics, Volume 50, Number 11S)

**Answer:** Periodic table is a natural interface for chemists, that was used for many years in this particular domain. In the above-mentioned articles deal with a specific domain-oriented databases (year 2018 - HTEM by NREL; year 2011 - AtomWork by NIMS, 2011) rather than proposing an approach to develop extensible materials framework in terms of enabling storage and search for diverse chemical object types. To that extend these papers are not competitors but excellent examples of particular materials databases creation.

**Remark 2:** Previous works of the authors concerning database and information system development in material science should be also highlighted in related work section.

**Answer:** We have 4 references to our papers from 2005, 2016, 2017 and 2020, which we believe fine, since we rely on our experience and lessons learned from these systems to overcome the limitations in the new RDMS.

**Remark 3:** Internet accessibility is posed as one of the key requirements, but software architecture of the system is not considered.

**Answer:** That is true, but since the RDMS being developed relies on the classical 3-tier architecture (database server – application server – client/browser), there is nothing special to mention on this point, but e.g. digging to specifics of tenant implementation would require far more technical details that could not fit to the regular article format, but could be answered during Q&A session (we provide open-source code to clarify all the questions that could come as well).

**Remark 4:** Algorithms for data transformations are just mentioned in the introduction, but not discussed in the paper. These are properties import/export scripts, ad-hoc data import sql-scripts mentioned in section 4, or something more? Probably some examples of properties import/export scripts should be provided.

**Answer:** Shortly, algorithms for data transformations are type specific and in general are defined through the `IDatabaseValuesGetter` interface (<https://gitlab.ruhr-uni-bochum.de/vic/infproject/-/blob/master/TypeValidationLibrary/IDatabaseValuesGetter.cs>) that is to be implemented for a specific object type either internally (“type:” prefix - initialization via Reflection mechanism) or externally (“https://” prefix - call to an external webservice, standardized by OpenAPI). You could see an implementation for EDX object type ([https://gitlab.ruhr-uni-bochum.de/vic/infproject/-/blob/master/TypeValidationLibrary/TypeValidator\\_EDX\\_CSV.cs](https://gitlab.ruhr-uni-bochum.de/vic/infproject/-/blob/master/TypeValidationLibrary/TypeValidator_EDX_CSV.cs)) that is accessible in both ways via `TypeInfo.DataSchema=”type:TypeValidationLibrary.TypeValidator_EDX_CSV”` or `TypeInfo.DataSchema=”https://validation.matinf.pro/edx/data”` (see `POST /edx/data/databasevaluesbody` method at <https://validation.matinf.pro/index.html>).

Everything here is quite complex to describe within article format, probably dealing with EDX data would require the whole article (we consider writing manual after all).

SQL-script, mentioned in Section 4, was used to transfer data from Bandgap database (`bg.imet-db.ru`) to one of the tenants of developed system to demonstrate functionality. This is not “documented way” of transferring the data, but a good one for quick data import from existing relational data source to demonstrate the system. We don’t consider this particular script to be of any interest since it hardly relies on the Bandgap

database structure, but for the sake of truth it's available "as is" at <https://drive.google.com/file/d/1q5bVGsVE257aUswzvJHuWpiIZXD-cRgyi/view?usp=sharing>

**Remark 5:** Extensibility supported by "extended properties" is fine. But what about query performance with denormalized schema? More tables require more joins in queries. 7500 properties mentioned in 3.3 do not seem as a large table for the modern databases.

**Answer:** Extended properties is a mechanism that out of the box enables users to store user-defined table structures and search them. An example provided with 7.5K properties for the single object is just a test case to store and search many properties for a single object, whereas a database could exist a plenty of them. The only limitation is a database size which is not a constraint in case with SQL Server and materials data. Of course, performance on denormalized data in that way would be never efficient, which is fine if you don't deal with big data. In case of really huge data we still have opportunities to tune up the performance in two ways: 1) simple - tune up indices and materialized views; 2) implementing required table type in the system as an extension of ObjectInfo table (see dbo.Reference table in database and corresponding <https://gitlab.ruhr-uni-bochum.de/vic/infproject/-/blob/master/InfProject/Models/Reference.cs>), this introduces outstanding native support with maximal performance. Which way to choose depends on data volumes and performance requirements.

**Remark 5:** Is it possible to provide some use case for "data-driven decision-making" using the system?

**Answer:** The system belongs to data management class. Responsibility of the system ends when providing data to external software via API (under development). We have plans of automatic pairing with our ML-tools to enable users make data-experiments easy, but it's not yet done and according to our opinion definitely out of the scope regarding the paper title.

**Remark 6:** As stated in abstract, "facilitate and accelerate similar data-centric projects in inorganic materials science by encouraging you to deploy our solution in building your own customizable materials research infrastructure". Is it possible to describe a potential use case?

**Answer:** We claim that due to extensible user-defined types and extended properties any of the known database systems in inorganic materials science could be implemented on the base of developed system (bg.imet-db.ru is already demonstrated, other IMET RAS systems could be potentially transferred to the new platform, e.g. crystal.imet-db.ru, phase.imet-db.ru; MDI chair at Ruhr-Uni Bochum also uses the system for High-Entropy Alloys thin films, etc...)

**Remark 7:** For the main software components, database schema, import-export scripts, use case scripts the respective links in gitlab should be provided.

**Answer:** The link is provided in conclusion section: <https://gitlab.ruhr-uni-bochum.de/vic/infproject> (a couple of adjacent components could be also found at <https://gitlab.ruhr-uni-bochum.de/vic>)

**Remark 8:** URL link at the end of page 8 should be shortened using a URL-shortener or/and moved footnote

**Answer:** Actually, the link specifies all the search criteria that makes this link meaningful to understand search results persistence. Shortened link would hide the details preventing readers from persistence understanding in terms of search parameters snapshot. We are sorry for long link, but it provides search on objects of specific user type belonging to specific chemical system and having specified property in a desired range (to this extend we would like to keep the link in the main text).

**We cordially thank all the reviewers for their useful comments and looking forward to having fruitful after talk discussion.**