Comprehensive and Comprehensible Data Catalogs: The What, Who, Where, When, Why, and How of Metadata Management

Pranav Subramaniam[1], Yintong Ma[1], Chi Li[1], Ipsita Mohanty[2], Raul Castro Fernandez[1]
[1] University of Chicago, [2] IIT Kanpur
{psubramaniam,yintongma,lichi,raulcf}@uchicago.edu,ipsita@iitk.ac.in

ABSTRACT

Scalable data science requires access to metadata, which is increasingly managed by databases called data catalogs. With today's data catalogs, users choose between designs that make it easy to store or retrieve metadata, but not both. We find this problem arises because catalogs lack an easy to understand *mental model*.

In this paper, we present a new catalog mental model called 5W1H+R. The new mental model is *comprehensive* in the metadata it represents, and *comprehensible* in that it permits users to locate metadata easily. We demonstrate these properties via a user study. We then study different schema designs for the new mental model implementation and evaluate them on different backends to understand their relative merits. We conclude mental models are important to make data catalogs more useful and to boost metadata management efforts that are crucial for data science tasks.

1 INTRODUCTION

In an increasingly data-driven world, a major data management challenge in industry, the sciences, and the civic sphere, is the lack of solutions to find relevant data (**D**iscovery), manage data usage and access (**G**overnance), combine data to multiply its value (**I**ntegration), and guarantee that data use is compliant with regulations (**C**ompliance). The common denominator to all these challenges is that addressing them requires access to *additional information about the datasets involved*: i.e., metadata.

Many initiatives have appeared over the last few years to organize and represent metadata to deal with DGIC challenges. The FAIR principles [51] in the sciences have galvanized academics to complement initiatives such as Dataverse [14], ICPSR [43]—repositories meant to share data in the sciences and social sciences—with metadata that facilitates their usage. The explosion of machine learning and its consequent pressure on data management has motivated efforts concerned with cataloging data assets in the form of datasheets [21] and enterprise metadata management systems [2, 8]. We call solutions designed to store and retrieve metadata *data catalogs*. Different applications and people define metadata differently, so catalogs must accommodate different definitions. But this multiplicity of definitions leads to disagreements on metadata descriptions and meaning. And these disagreements, in turn, make metadata hard to find, e.g., if an employee calls 'cardinality' what another calls 'distinct'. An ideal catalog accepts multiple metadata definitions while bounding disagreements on such definitions, making it easy to find metadata. Today's catalogs fall short of this ideal. Either they make it easy for people to store metadata or they make it easy to retrieve it, but not both.

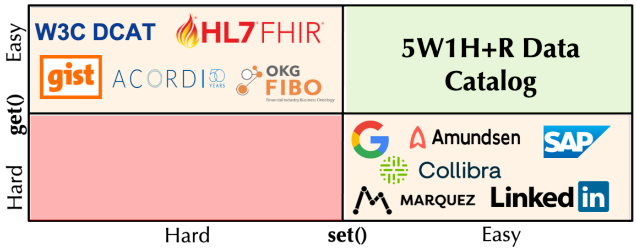

Figure 1: Design Space of Metadata Cataloging Solutions

Fig. 1 shows a mapping of catalogs in terms of how hard or easy it is to represent or query metadata. Data catalogs such as Amundsen [41], DataHub [37], Collibra [13], and the cloud vendors' make it easy to store metadata in any way users want because they offer flexible schemas. However, this flexibility leads to disagreements in the metadata definitions. As a consequence, users find it difficult to find the required metadata. Conversely, controlled vocabularies such as W3C DCAT [48] and internal data dictionaries define a precise vocabulary that, when used to annotate metadata, facilitate its retrieval. However, the annotation process is time-consuming due to the vocabulary size, which makes it tedious to find the vocabulary term that best describes the metadata. We posit that the main reason why catalog solutions do not make it both easy to store and retrieve metadata is their lack of a *mental model* that is understood by all catalog users.

In this paper we study the kinds of DGIC questions that appear in the literature as well as many data catalog solutions. We use this study to inform the main contribution of this paper, which is a mental model for metadata management. The new mental model is based on the 5W1H principles of journalism [20]. By explaining metadata in terms of these principles, technical and non-technical users alike find it easier to decide the best place to store (and hence, retrieve) metadata.

We evaluate the new mental model along two dimensions: its fit for metadata management, and its fit for practical implementation. For the former, we design a mental model that is comprehensive (allows users to represent a wide variety of metadata definitions), comprehensible (allows users to store and find metadata consistently), and easy to understand. Regarding its fit for practical implementation, we study data vault [40], a schema design framework that leads to easy-to-evolve schema designs. Evolvability is important to maintain catalogs as schemas will change over time.

We conduct an IRB-approved user study to compare the new mental model with those of existing catalogs. Our results show

that the new mental model makes it easier for users to store and retrieve metadata consistently. We compare the relative merits of different schema designs implemented on different backends to host a catalog that implements our new mental model. We compare these along metrics such as query complexity, storage footprint, query performance, and evolvability. We conclude that modern data catalogs would benefit from clear mental models, as this facilitates metadata management, and in turn, the solution to DGIC problems, which are an important bottleneck in today's applications.

Supporting Scalable Data Science. Scalability is more than performance. The lack of metadata management solutions hampers data science efforts. We have informed the motivation and contributions of this paper by consulting with a large number of industry and science stakeholders, and we make an effort in connecting our work to other initiatives with the aim of contributing to metadata management academically and practically.

Outline. Section 2 is dedicated to explain the characterization of data management questions into the DGIC categories, as well as the ecosystem of data catalog and cataloging solutions. We introduce our mental model and conceptual model in Section 3 and the catalog materializations in Section 4. Section 5 presents evaluation results, followed by a discussion in Section 6, related work (Section 7) and conclusions in Section 8.

2 LANDSCAPE OF DATA CATALOGS

We start with an example. Alice, a machine learning (ML) engineer, wants to build a model to predict how many headphones her company will sell next week, so she needs a training dataset. Alice *searches for training datasets* used by other analysts in the past. She searches for *datasets with relevant attributes* and finds four relevant datasets. She wants to choose datasets *created with the purpose of training the target ML model*, datasets with *fewest null values* and prefers *datasets created by Bob*, an engineer she trusts.

In the example, sentences in italics suggest intermediary tasks that can be assisted with relevant metadata. Without this metadata available, solving each of these tasks is usually a time-consuming, tedious, and ill-defined process that consumes much of data workers' energy. Making that metadata available requires: i) creating the metadata and, ii) representing metadata in a way that is useful to other data workers. In this paper we are concerned with the second task: how to represent metadata.

In this section, we first define metadata and mental models in Section 2.1. Then, we discuss our scope of the metadata landscape in Section 2.2. We conclude the section with a discussion of current data catalogs Section 2.3.

2.1 Metadata and Mental Models

Data assets are artifacts describable with metadata, including rows, columns, relations, unstructured files, derived data products, or others, as well as groups of any of these. A **metadata-item (MI)** refers to a *data asset* or a group of *data assets*. A MI is a key:value pair where the key indicates a property of interest and value contains the value for the property.

MIs are stored in a **catalog** via set(key, value) operations and retrieved using a get(key) operation. A catalog stores two types of metadata: the MIs that refer to data assets, and **audit metadata** that refers to the set and get operations and includes who executes the operations, and when, among others.

Applications may describe the same property of a data asset differently, leading to different MIs. Or they may express the MI values using different representations, e.g., different units. This flexibility lets applications describe data assets without constraints but makes retrieving metadata challenging because the MI key is not known a priori.

A **mental model** (MM) is a partitioning of the metadata items C into subsets $\{p_1, \ldots, p_n \in P | p_i \subseteq C, p_i \cap p_j = \emptyset, \forall i, j\}$, where $\bigcup_{i=1}^{n} p_i = C$. Without a MM available, retrieving a MI without knowing its key requires searching through all MIs. If a MM is available, retrieving a MI requires i) finding the p_i where it belongs, and ii) searching through all MIs in that partition. Mappings between MIs and MM's partitions do not exist a priori and depend on user's and application's understanding of both MIs and the MM's partitions, i.e., users of the MM decide where to map a MI. A good MM is one that leads two users to agree, without explicit communication, on the partition where a MI belongs. Hence, sharing a mental model leads to consistent decisions when addressing a task, even when users of the MM are users in different organizations who do not communicate with each other.

MM quality. A MM is *comprehensive* in the metadata it describes if every MI can be placed in a partition of the MM. A MM is *comprehensible* if it generates agreement on the mapping between MIs and partitions among users of a catalog implementing that MM.

Search and IR. IR and MM are orthogonal methods to facilitate finding MI elements in catalogs. While MM are geared towards narrowing down the search space for MI to individual partitions, IR systems can help with addressing the remaining gap. This paper's focus is on mental models.

2.2 The Metadata Landscape

In practice there will be many valid MIs. To gain intuition on those commonly used we have surveyed the literature and consulted with data employees. We have extracted 155 DGIC questions data workers often face that would be addressed with access to the right MIs. These questions illustrate the metadata landscape we consider in this paper. We have synthesized the 155 questions into 27, shown in Table 1. Answering a question in our paper means identifying the partition where the answer MI is stored. Classifying a question means identifying a partition where this MI should be stored.

The DGIC questions. A ***D**iscovery* task involves selecting a subset of datasets from a larger set to satisfy specific criteria [9, 13, 25]. A ***G**overnance* task involves ensuring that the purpose of a data asset, and the method used to fulfill that purpose are understood by all data users. Note that our definition of governance is more inclusive than existing definitions [35], which consider governance tasks to be related to decision rights and accountabilities regarding who uses data, and for what. An ***I**ntegration* task involves combining existing data assets, which may require preparation [11, 16]. A ***C**ompliance* task is one in which a data user has to ensure that data assets with sensitive information meet regulatory requirements [5, 36]. We show in Table 1 the question classification into DGIC.

	Representatives of Common Data Questions	DGIC Category	5W1H+R Partition
Q1	For what purpose was the dataset created?	G [21, 49]	Why
Q2	Are there tasks for which the dataset should not be used?	G,C [21]	Why
Q3	Who created the dataset?	G,C [21, 26]	Who
Q4	Who was involved in the data creation process?	G,C [21]	Who
Q5	How can the owner/curator/manager of the dataset be contacted?	G [21]	Who
Q6	What are the privacy and legal constraints on the accessibility of the dataset?	C [38]	Who
Q7	Is there an access control list for the dataset?	G,D [26]	Who
Q8	What is the reputation of the creator of a dataset?	G [24]	Who
Q9	What do the instances of the dataset represent?	D,G,I [21]	What
Q10	What is the size of the dataset?	D,G,I [26]	What
Q11	Are there errors in the dataset?	D,G,I [21, 24, 38]	What
Q12	Does the dataset have missing values?	D,G,I [24]	What
Q13	What is the domain of the values in this dataset?	D,G,I [30]	What
Q14	If the dataset is a sample of a larger dataset, what was the sampling strategy?	G,I [21]	How
Q15	Does the dataset contain personally identifiable information (PII)?	G,C [4, 49]	What
Q16	What is the quality of the dataset?	G [3, 4, 13, 39]	What
Q17	Was any preprocessing/cleaning/labeling of the dataset done?	G [21]	How
Q18	Was data collection randomized? Could it be biased in any way?	G [38]	How
Q19	Is there anything about dataset preprocessing/cleaning that could impact future uses?	G [21]	How
Q20	What is the dataset's release date?	D,G,I [30]	When
Q21	Is there an expiration date for this dataset?	D,G [3]	When
Q22	How often will the dataset be updated?	G,I [21]	When
Q23	When was the data last modified?	D,G,I [26]	When
Q24	How easy is it to download and explore this dataset?	D [24]	Where
Q25	What is the format of the dataset, and what type of repository is the dataset located in?	D [38]	Where
Q26	What is the provenance of this dataset?	I [54]	Relationship
Q27	What other datasets exist in this repository that are related to this dataset?	D,G,I [52]	Relationship

Table 1: Questions, DGIC and 5W1H+R partition

	D	G	I	C
Berkeley's Ground [28]	K	P	P	P
Microsoft Azure Data Catalog [31]	W	P	P	P
Apache Atlas [3]	W	P	P	P
Denodo platform [17]	W	P	P	P
SAP Data Intelligence platform [46]	W	P	K	K
Boomi Data platform [7]	W	P	P	P
WeWork's Marquez [50]	K	P	P	P
Lyft's Amundsen [41]	W	P	P	P
Linkedin's Datahub [37]	W	P	P	P

Table 2: DGIC support in existing catalogs. *Keyword* (W), *Key* (K), *Partition* (P).

2.3 Analysis of Today's Data Catalogs

We study existing data catalogs to understand what features they provide to store and retrieve metadata. The results are shown in Table 2. *Keyword* (W) means the catalog provides a full-text search feature to search for metadata. *Key* (K) indicates the catalog explicitly provides a pre-specified collection of MI keys, e.g., when catalogs focus on a specific area of DGIC. *Partition* (P) indicates the catalog provides a partition (in the MM sense) to classify a MI.

Some catalogs were designed to address different areas of DGIC and consequently have more support for those than others. Although we are not aware if catalogs were designed around a MM explicitly, we can derive their MM from their documentation and after using them extensively. All the analyzed catalogs implement a MM consisting of: (i) partitions for one specific key:value pair of metadata; and (ii) a *catch-all partition*: a single partition for all key:value pairs the catalog does not explicitly describe.

While the catch-all partition allows these MM to be comprehensive in the MI they describe, they lead to lack of comprehensibility because they accept any MI. For that reason, these catalogs sit in the bottom right corner of the quadrant in Fig. 1: it is easy to store MI but it is hard to find later.

On the other hand, controlled vocabularies such as W3C DCAT consists of many partitions, each with a single key:value pair. This makes it easy to query data catalogs implementing this ontology as a MM as long as the terms in the vocabulary are known by the user who submits the query. However, it is difficult to represent metadata with these catalogs because each MI needs to be annotated with one of the many partitions. As a consequence, we classify controlled vocabularies in the top left corner of the Fig. 1 quadrant.

3 MENTAL MODEL AND CATALOG DESIGN

In this section, we introduce the 5W1H+R MM in Section 3.1 and explain how to use it as part of a catalog in Section 3.2.

3.1 The Mental Model

The 5W1H component of the MM applies to MIs of individual data assets (Section 3.1.1), and the +R component applies to relationships between more than one data asset (Section 3.1.2). We conclude explaining why the MM is comprehensive, comprehensible and easy to understand in Section 3.1.3.

3.1.1 5W1H-Profiles. Given a MI that describes 1 data asset, the MI fits into one of the 5W1H partitions of the MM. We have included a column with the 5W1H partition in Table 1 as an example. Note that we call each 5W1H partition of a data asset a *profile*.

A mnemonic for the 5W1H MM is the following: '**Why**, **When**, and **How** does **Who** use a **Data Asset** (located **Where**) whose contents are described by **What**?'

Who-profile. Items identifying persons or software that created, modified, or can access the data asset, and/or explaining their relationship with the data asset (e.g. role information, access privileges to the data asset). A data question falls in this partition if: i) the question can be answered by information about who has used the data asset before; ii) the question can be answered by information about who can access the data asset.

What-profile. Information that requires looking at the data. A data question falls in this partition if: i) the question can be answered by reading the data directly and/or performing computations on the data. ii) the question can be answered through (existing) semantic information about the data, such as schema annotations or descriptions.

When-profile. Temporal information about the data asset lifecycle: when it was created, modified, when does it expire, etc. A data question falls in this partition if: i) the question can be answered by information about data asset usage during a particular time period; ii) the question can be answered by information about when the data asset is available.

Why-profile. Explains why the data asset exists, its purpose, and intended use. A data question falls in this partition if: i) the question can be answered by information about why the data asset was

used a certain way, including why the data asset was created and deleted; ii) the question can be answered by information related to the intended uses of the data asset.

Where-profile. Physical location of the data asset and information about how it is accessed. A data question falls in this partition if: i) the question can be answered by information about how to access this data asset; ii) the question can be answered by information about the format of the data asset being stored (such as CSV file vs. postgres table vs. mysql table). iii) the question can be answered by information about the source where this data asset can be located.

How-profile. Information about the processes that produced, modified, or read the data asset. This includes data collection and preparation methods, as well as programs, queries, or artifacts that were run on the data asset. A data question falls in this partition if this question be answered by information about how a data asset has been used (read, modified, or created) for a task/application.

3.1.2 Relationships Profile. A MI may refer to more than 1 data asset, in which case we cannot classify into any of the 6 above partitions. Our MM includes a *relationship* partition to describe these MIs. As with individual items, it is useful to think about the types of relationships our MM should support in terms of the 5W1H partitions. For example, consider 2 columns as data assets. Their Jaccard similarity and containment (inclusion dependency) is a relationship on their What-Profiles because both can be computed by looking at the data. A PK/FK relationship can be represented as a Why-Profile if the ids were produced with the purpose of indicating a join relationship between two relations. Consider relations A and B, where B is a version of A. We can express their temporal relationship using a When-Profile, their provenance using a How-Profile, etc. We can relate data assets according to who created them, or who accessed them using relationships on Who-Profiles. We can store relationship MI as combinations of profiles.

3.1.3 The 5W1H+R MM Rationale. The 5W1H+R MM *bounds disagreements* when setting/getting MIs by presenting users with a MM with high cognitive fit, i.e., one that leads users to implicitly agree on the mapping of a MI to a MM's partition. On choosing the MM we strived to make it comprehensive, comprehensible, and easy-to-understand:

- **Comprehensive.** Journalists use 5W1H to cover news because it leads to a broad coverage of the event. We could represent all 155 DGIC questions extracted from the literature in the MM. We include the 5W1H+R partition along with the 27 questions that synthesize the corpus in Table 1.
- **Comprehensible.** The 5W1H+R MM has high cognitive fit [47] with any task involving the storage or retrieval of metadata. A MM has cognitive fit with a task if there is a common mapping among users performing the same task between a MM's *internal representation* (the way the MM partitions metadata) and the *external task representation* (a user's specification of the metadata required for the task).
- **Easy-to-understand.** The 5W1H+R MM is explained in natural language. Most natural languages describe objects and events using 5W1H, making it a familiar set of terms to most people, technical and non-technical alike.

The above are qualitative reasons. We provide quantitative data, obtained through a user study, in Section 5.

3.2 Catalog Design

Catalogs are more than MMs. They need to store metadata about the set and get operations for the reasons we explain next.

Consider a user who is extracting a MI manually from a set of high-value data assets, and a software profiler that is doing the same. Even if the profiler's developer and the user agreed on the partition where to store the MI—an indication of having a good MM—the actual MI may be different. For example, the user may choose the word 'uniques' to describe the number of unique values in a relation column and a ratio to express the quantity, while a software profiler may choose 'unique_number_values' and store the absolute number instead.

A good catalog design must accommodate all versions of MIs referring to the same data asset, and let the consumer reconcile or decide which one to trust. To facilitate this, a catalog design must include *audit metadata* that describe each set and get operation. Audit metadata includes who (what user) is setting a MI, the version of the MI (to differentiate among existing ones), the time at which the MI was stored, etc. We enumerate the following requirements for a catalog design:

- MIs should be represented as key:value pairs, so the catalog does not restrict different definitions. For example, in schema designs this can be achieved using a JSON data type for the MI fields in each profile.
- The catalog should represent explicitly the 5W1H+R in the schema so it maintains high cognitive fit. For example, in a schema design there should be a table for each 5W1H profile and relationships, each holding the MIs associated with that profile.
- The catalog should represent data assets of multiple granularities and refer to them with unique identifiers, so they can be associated with profiles.
- Each MI in a profile should include audit metadata information describing the set operation used to store such a MI event.

Armed with the MM and the catalog requirements above, one can design a schema to materialize the catalog. We discuss next some practical considerations that we deem important when materializing the design.

4 MATERIALIZING THE MENTAL MODEL

In this section, we explain the need for schema evolution in data catalogs and their consequences for catalog users and applications in Section 4.1, followed by a primer on data vault (Section 4.2) and ending with a comparison of the relative merits of data vault design vs traditional database normalization from the perspective of schema evolution in Section 4.3.

4.1 Schema Evolution Consequences

Every MI is a key:value pair, and all MIs that pertain to the same partition will be stored together (for example, as a JSON object containing a list of key:value pairs in a column of the profile table). In practice, some MIs will be queried more often than others, and some MIs may be amenable to specific indexing techniques.

Consequently, database administrators may prefer to isolate these often-queried MIs together and create an index to speed up queries. This introduces a schema evolution problem: how to change the schema without breaking existing applications.

In particular, users and applications that use catalogs implemented on 3NF schema designs will have problems identifying and correctly using these indexes. In a normalized schema, extracting MIs from a profile requires creating a new *key table* whose column names are MI key names, and whose values are the MI values. This key table is required, as opposed to adding new columns to the profile table, in order to maintain the schema in 3NF: adding new columns would create NULL values in profile records that do not have the often-queried MIs, denormalizing the resulting schema. This means entirely new entities and/or relationships are required to describe all such key tables to catalog users, placing the burden of identifying and correctly using these key tables on them.

We consider an alternative schema design that is robust to schema changes: data vault.

4.2 A Primer on Data Vault Schema Design

The data vault schema design was created for the purpose of tracking data stored in a data warehouse [32]. It was designed to support auditability and historical tracking, while separating business keys, relationships and attributes in a fashion similar to a snowflake schema. A data vault schema has three types of tables: *hubs*, *links*, and *satellites*. Hubs store the IDs of entities, links correspond to relationship tables between two entities, and satellites of a hub hold the attributes of the entity corresponding to the hub. Data vault readily includes tracking attributes such as load timestamp and record source (where the record was loaded from). In the context of a catalog, we replace this information with the audit metadata (see Section 2.1) associated with set and get operations.

This results in schema evolution with easier-to-identify key tables than with a 3NF design. Isolating a set of MIs in their own key table requires creating a new satellite table whose attributes include the key names of the MIs. Applications that benefit from these new indexes can use the new satellite table, entirely aware of which profile this satellite table is describing, and not requiring any further changes.

Data vault and snowflake. The data vault schema is essentially a snowflake schema: the asset hub table is a fact table after adding the profile IDs from the link tables, and all other tables are dimension tables. Although data vault specifies the inclusion of tracking information, i.e., that we call audit metadata, the same information could be included in a snowflake design.

4.3 Evolving the Catalog

Evolving data vault results in key tables that are easier to identify and correctly use compared to 3NF. Unlike 3NF, which requires changing existing queries to join with a new table or use a new table that is completely isolated from the profile table, in the case of data vault we only need to modify queries to use a new satellite table of an existing profile table. This is because adding this key table to 3NF requires including a new entity and/or relationship, while in data vault, because satellite tables are linked to hub tables by design, adding a new key table only requires adding new attributes to an existing entity set.

Evolvability is the main reason why we explore the data vault design. It comes with some costs in the form of increased storage footprint, more complex queries, and worse query performance, due to duplication of data across hubs, links, and satellites and joining across links. We study these properties in detail in Section 5.

5 EVALUATION

In this section, we present the results of an IRB-approved user study we conducted to answer the main research question of our work: *is the 5W1H+R MM more comprehensible and easier to understand than other MMs?* (Section 5.1). Then we explore the relative merits of different catalog implementations in Section 5.2.

5.1 Evaluating Catalog Mental Models

The 5W1H+R is comprehensive: we could represent metadata items (MIs) answering all 27 questions (see Table 1). We design a user study to measure the two other metrics of interest: *consistency* and *ease of understanding*. Consistency is our measure of comprehensibility. It measures the degree to which there exists a common mapping among users between the MIs required for a task and the MM partitions. That is, it measures the degree of agreement on MI-MM partition mappings. Ease of understanding measures how easy users find it to map a MI to the MM. We want MMs with high consistency and high ease of understanding.

5.1.1 Target Catalogs. We compare the 5W1H+R MM with LinkedIn's Datahub and Google Data Catalog MMs because their MMs are good representatives of open-source catalogs, (i.e., Amundsen, Marquez, and Apache Atlas), and cloud service data catalogs, (i.e., Microsoft Azure Data Catalog, Denodo Cloud Platform, and SAP Data Intelligence), respectively. Although new catalogs appear often[1] we believe these collections represent existing offerings well. To extract the MM from each catalog, we applied a rubric that consisted of using the catalogs in practice to insert metadata and consulting their documentation. We only use terminology the catalogs explicitly describe and define.

5.1.2 Between-Subjects Study Design. We implement the user study as a survey which we distributed online via Prolific [45] to 160 participants. We used several pilot studies with database researchers and non-technical people to hone the language of our survey to avoid threats to validity and allow both technical and non-technical data users to participate in our study.

We ask each participant to rate their familiarity with the questions from Table 1. We then show them the questions they rated high and present them one MM:

Consistency: Each participant classifies the question to a MM's partition (as defined in Section 2.2). Participants' answers for a question provide a frequency distribution over the partitions of a MM. Because different MMs contain different partitions, we cannot compare distributions directly. However, we know that the further from a uniform distribution the more agreement exists on a MM, so we use entropy to determine a MM's consistency. A consistent

[1] an industrial collaborator tells us they have tested 40 different catalogs

	Better															Inconclusive											Worse		Average
MM	Q1	Q2	Q3	Q5	Q6	Q8	Q9	Q12	Q14	Q16	Q17	Q19	Q21	Q22	Q27	Q4	Q7	Q10	Q11	Q13	Q15	Q18	Q20	Q23	Q24	Q25	Q26	
5W1H	**1.06**	**1.83**	**0.81**	**2**	**0.61**	**0.99**	**1.73**	**1.34**	**1.9**	**1.83**	**0.93**	**2.37**	**1.94**	**1.6**	**1.58**	2.42	1.97	1.81	2.25	2.06	2.09	1.46	1.13	1.84	1.52	2.07	2.72	**1.698**
GCS	2.4	2.68	2.6	2.49	1.01	2.66	2.48	2.72	2.67	2.4	2.52	2.61	2.32	2.05	2.28	2.44	**1.78**	2.35	2.36	2.49	2.48	**1.44**	**0.68**	2.02	2.12	1.81	1.88	2.212
Datahub	2.27	2.36	1.65	2.37	1.23	1.8	2.18	2.15	2.52	2.25	1.34	2.38	2.01	1.91	1.66	**2.02**	1.92	**1.73**	**2.03**	**1.63**	**1.87**	2.06	1.64	**1.65**	**1.41**	**1.67**	**1.81**	1.908

Table 3: Entropies per MM per Question

MM	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26	Q27	Total
5W1H	**1**	6	**1**	**7***	**9**	**2**	4	**7**	**10**	18*	**12***	**12**	**1**	**4**	**11***	**7**	**2**	**1**	**9***	**2**	**6**	**5**	11*	**4**	3	**8***	**2**	**165**
GCS	11*	**4**	14*	18*	19*	8	5	27*	11*	27*	20*	17*	9*	7*	20*	15*	8	4	18*	5	12*	17*	**8**	13*	**1**	12*	10*	340
Datahub	13*	7	5	20*	17*	4	**3**	14*	19*	**10**	15*	25*	8	8*	14*	13*	11	17*	15*	18*	15*	20*	15	**4**	**1**	15*	5	331

Table 4: Number of None Responses Per MM Across Questions And Users: * None was the most selected answer

	Better								Inconclusive																Worse	
MM	Q1	Q2	Q3	Q5	Q12	Q17	Q18	Q27	Q4	Q6	Q7	Q8	Q9	Q11	Q13	Q14	Q15	Q16	Q19	Q20	Q21	Q22	Q23	Q24	Q26	Q10	Q25
5W1H	**2****	**3***	**2****	**3***	**2****	**2****	**2***	**2***	**3***	2	3*	3	3	**3**	**3****	3**	3	3	**3***	**2***	3	3**	3*	3	**3**	4	3
GCS	3	3.5	3	4	3	3	3	3.5	4	2	3	3	3	4	3.5	3	3	3	**3**	**2**	3	3	3	3	4	4	**2**
Datahub	3	3.5	3	3.5	3	3	3	3	**3**	2	3	3	3	**3**	**3**	3	3	3	4	3	3	3	3	3	**3**	**3**	3

Table 5: Median Difficulties per MM per Question Using Likert Scale(1-5): * $p < 0.05$ for one MM, ** $p < 0.05$ for both MMs

MM is one that produces a low entropy distribution. We analyze None values after discussing the consistency results.

Ease-of-Understanding: a MM's ease-of-understanding for a data question is measured using the standard Likert scale (a 1-5 difficulty rating, with 1 being "Very Easy", and 5 being "Very Difficult").

5.1.3 Participant Recruitment. We use a combination of Prolific filters (only participants with formal education can participate) and pre-screening process (participants only work on questions they are familiar with) to avoid spurious answers. We make sure participants are not familiar with the MM before the study to avoid confounding effects due to prior familiarity.

5.1.4 Threats To Validity. Threats to validity of our study include:
Ordering and Learning Effects: We use a between-subjects study, which complicates our statistical analysis and data collection, but avoids learning effects of answering questions across MMs.
Selection Effects: We use the prescreening questions and Prolific filters to select a homogeneous participant pool with data management familiarity. We assign MMs to participants in the pool randomly, and they only answer questions they understand.
Experimenter Bias: We mitigate the effects of experimenter bias by generating all MMs using the same procedure, and presenting these MMs to participants with the same instructions on how to use each. Additionally, because our study is an online survey, participants cannot be further exposed to our bias.
Reactivity Effects: We ask participants to rate the difficulty of categorizing data questions. Participants may feel they need to rate the difficulty lower than their actual experience if they feel our experiment is evaluating them. We mitigate these effects by wording the question carefully and by explaining in our survey introduction that our survey is intended to evaluate the MMs and not the participants' ability.

5.1.5 Results. **Consistency.** The results in Table 3 show that the 5W1H+R MM has a lower average entropy across all questions compared to other MMs, i.e., it is more consistent. Concretely, the 5W1H+R MM has lower entropy than others in 15 questions, it is second best in 10 and worse in 2/27. The 5W1H+R MM outperforms the others on questions where, either the answer cannot be found in the other MMs, or one of the 5W1H+R definitions explicitly describes the answer (for example, the answer to the question "What is the purpose of this dataset?" is Why-profile because it appears in its definition). We suspect that the main reason for the 5W1H+R MM's worse performance on the other questions is that the other MMs offer more concrete concepts for classifying the answer to a question. For example, the data question "How easy is it to download and explore this dataset?" is more concretely answered through information about the repository in which a dataset is located and its format (the "Data Source" definition from the Datahub MM) rather than "information about where a dataset is located and how it is accessed" (Where-profile). Another example is the question "What are the privacy and legal constraints on accessing this dataset?" This question is more concretely answered using information about "who currently has access, and what is their role with respect to the dataset" (Ownership from Datahub's MM) rather than "who can access this dataset, and/or explaining their relationship to the dataset" (Who-profile).

Choosing None. None answers indicate a weakness of a MM. Table 4 shows that participants using 5W1H+R chose this option the least amount of times; less than half as frequently as in other MMs. In 19 out of 27 questions, they chose it the least often. Finally, None was the most selected answer for only 7 questions when using the 5W1H+R MM, as opposed to 19 and 16 for GCS and Datahub.

Higher consistency and lower use of None demonstrate the 5W1H+R MM is comprehensible.

Difficulty. Table 5 shows that the 5W1H+R MM has the same or lower median difficulty than the others for 25 out of the 27 questions. Further, using a Mann-Whitney U Test, we find that there is a significant difference between the 5W1H+R median difficulty rating and the median difficulty rating of both other MMs for 7 of the 25 questions, and there is a significant difference between the 5W1H+R median difficulty rating and the median difficulty rating of one of the other MMs for 9 of the 25 questions. This result suggests that for most questions, **the 5W1H+R MM is easier to understand compared to other MMs**.

5.2 Evaluating Catalog Materializations

In this section, we evaluate the merits of different schema designs when implemented on different backends.

5.2.1 Catalog Evaluation Methodology and Metrics. We consider a normalized (3NF) and a data vault schema implemented on top of a RDBMS, SQLite, and on top of a graph database, Neo4j. Data vault is easier to *evolve* than the normalized schema. Here we focus on other metrics:

- Query Performance: runtimes of queries in a comprehensive workload that we develop.
- Storage Footprint: the size of the databases for each combination.
- Query Complexity: how difficult it is to write a query for a schema-backend combination, measured using Halstead complexity and a High-Level Declarative Difficulty metric.

Setup. We run all experiments in an Ubuntu 16.04 OS, Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz, 8GB DDR3 RAM, and 451 GB HDD disk space. We use SQLite 3.33.0 and Neo4j 4.2.1. We execute queries Q4 to Q7 10 times and report the average runtime and deviation. Between each execution, we clear the pagecache: we are interested in measuring the relative merits of different schema designs and not how well the backend systems manage memory.

5.2.2 Dataset Generation and Workload. For validation we loaded a real dataset provided publicly by Google [42] into a 5W1H+R data catalog. The dataset is a CSV file of size 5.35 GB with 17 attributes describing the name of a data asset, where the asset is from, how it can be accessed, and how it should be used. We could map the attributes to 5W1H+R partitions easily, and insert them into the catalog by converting them into key:value pairs.

To conduct meaningful performance experiments we generate a 9.42GB dataset that allows us to exercise all the partitions of our MM. The dataset has an equal number of records for each of the 5W1H+R tables, and fixed numbers of records for the number of repositories the data catalog tracks data assets from (1000) and the number of relationships each data asset has with other assets (5).

We then design a workload that exercises bulk and transactional storage and retrieval in both database backends. We implement the workload using SQL for the SQLite backend and Cypher for the Neo4j backend:

- Q1 (Bulk Insert): Insert 10^5 records into the what-profile table (updating the versions and timestamps of what-profile records).
- Q2 (Single-Record Insert): Insert a new what-profile record.
- Q3 (Multi-Table Insert): Insert 10^5 who/when/why/how profiles.
- Q4 (Range Lookup): Get all information on where an asset was stored during a particular month.
- Q5 (Multi-Table Lookup): Get the how/who/when/why profiles for a data asset.
- Q6 (Limit Order by): Get the 10 latest how/who/when/why profiles on a data asset.
- Q7 (Aggregate Query): Group the recorded data asset users by the number of operations performed by each user (in the how profile), and return the top 10^4 such users.

	SQLite		Neo4j	
	Normalized	Data vault	Normalized	Data vault
Q1	**16.576**	35.762	52.286	182.949
Q2	**0.000**	0.015	0.339	0.238
Q3	**19.957**	36.253	480.392	2229.288
Q4	**0.028** ± 0.002	0.051 ± 0.024	27.966 ± 0.261	38.384 ± 3.221
Q5	0.157 ± 0.026	0.241 ± 0.051	**0.059** ± 0.006	0.112 ± 0.114
Q6	333.419 ± 0.765	623.978 ± 0.875	**0.182** ± 0.011	0.31 ± 0.255
Q7	**0.005** ± 0.01	0.014 ± 0.031	0.04 ± 0.001	0.041 ± 0.001

Table 6: SQLite and Neo4j Query Performance

5.2.3 Query Complexity Metric. We want to measure what schema design leads to more complex queries. We use two metrics to measure query complexity. We use the Halstead Difficulty measure [27], which is standard to measure program complexity. Because the Halstead metric penalizes repetition of operators, it is not well suited to measure the complexity of SQL queries. Hence, we devise a variant of Halstead, *High-level declarative difficulty metric* (HLD Difficulty) that reflects the following criteria: i) length of the query measured as number of operands; ii) number of distinct operators does not weigh much; iii) number of distinct operands does not affect query complexity: $(N_1 + N_2) \cdot (log(n_1) + 1)$, where N_2 and n_2 is the total and distinct number of operands, and N_1, n_1 indicate the same for operators.

5.2.4 Results. **Storage footprint.** Loading a 9.42GB dataset into the catalog results in a footprint of 13.30GB and 28.79GB for the normalized and data vault schemas on SQLite, respectively. These results include indexes on the primary keys of all tables. This difference exists because each of the profile tables in the normalized schema must be represented as 3 tables in the data vault table, and each of these data vault tables has replicated timestamp, user, and version information, meaning that each individual table in data vault is only a few pages smaller than each table in normalized. Further, the data vault design consists of more tables and hence more indexes. Together, this explains the higher footprint. We find similar results on Neo4j.

Runtime. We show our query performance results in Table 6. Table 6 shows that the data vault schema performs worse than the normalized schema in both backends. This is because the same query using the data vault schema requires joining more tables (or inserting into more tables), since a table in the normalized schema is represented with three tables (hub, satellite, and links) in the data vault schema. Note also that there were outliers that we excluded for the Neo4j queries (Q5-Q7) coming from the first run or first two runs of each of these queries due to Neo4j's index initialization and query caching for subsequent runs. We report them here: (i) Data vault Q5: 224.789, 261.855 (ii) Data vault Q6: 523.898, 899.198 (iii) Data vault Q7: 438.458 (iv) Normalized Q5: 62.635 (v) Normalized Q6: 51.231 (vi) Normalized Q7: 49.265.

Query complexity. Table 7 shows that all data vault relational variants of the queries are either as difficult or more difficult compared

to the normalized relational variants according to the Halstead difficulty. This is because the overall number of operands required for the data vault relational variants of the queries is higher. The HLD Difficulty metric further accentuates the difference in the length of the queries. Queries on the normalized schema are easier to write than in data vault.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7
Normalized Halstead	**0.5**	**0.5**	2	1.5	1.5	2.5	**2**
Data vault Halstead	3.56	2.85	**1.06**	1.5	1.5	2.5	2.25
Normalized HLD	**8**	**8**	**105**	**18.09**	**64.62**	**89.69**	**30**
Data vault HLD	56.87	56.87	109	31.02	121.49	139.5	42

Table 7: Summary of Query Complexity Results

In summary, our results show that, across backends, *the normalized schema design outperforms the data vault design with respect to query performance, storage footprint, and query complexity*. This is because the data vault schema requires more tables, hence introducing more duplication leading to a higher storage footprint, and more joins in queries, making them more complicated to write and more expensive to execute. **While data vault is easier to evolve compared to normalized, there is a tradeoff between evolvability and other performance metrics.**

6 CONNECTION TO METADATA EFFORTS

There are a plethora of related metadata management initiatives which we discuss and relate to our work here:

FAIR Principles. Catalogs generally, and our proposal in particular, are designed to manage metadata, which is a keystone to making datasets findable, accessible, interoperable, and reusable.

Ontologies. Although we have argued against using ontologies as a catalog's MM, they can annotate and describe contents within a MI, hence complementing metadata management solutions. Such annotations document and disambiguate the meaning of MI's keys and values. For example, one can use W3C DCAT's terms of "dataset distribution", and "temporal coverage" as What-profile key names, since these can be found by looking at the data.

ML model tracking. MLFlow [53] and ModelDB [29] track the lifecycle of machine learning model engineering and deployment. Both systems produce metadata that describes training datasets and models. Such metadata can be represented in the 5W1H+R MM. First, models and datasets are data assets in the MM. The metadata generated can be mapped to partitions in the MM. For example, the location of the code file, model parameters, and metrics become How-profiles. A provenance Relationship captures the link between model and the data used to fit the model, as well as plots and other derived data products. Model annotations and descriptions can be represented using the What-profile table of a 5W1H+R data catalog. Finally, different runs fit well with the versioning of What-Profiles along with the use of When-Profiles.

Metadata Collection. There are many methods and standards for metadata collection, including Montreal Data Licenses [6], Google Model Cards [18], Datasheets for Datasets [21], CancerGrid [15], DLHub [10], and ISO/IEC 11179 Metadata Registry [44].

A 5W1H+R data catalog complements these efforts. Metadata collected following those standards fits well into the 5W1H+R. For example, the Montreal Data License information could be stored in an asset's Who-Profile and Why-profiles, as answers to who is allowed to access the data asset, and for what purposes. Datasheets [21] and other metadata representation formats fit directly into the MM.

7 RELATED WORK

Existing Metadata Definitions. Metadata definitions aim to clarify what metadata means. One approach to defining metadata is to differentiate 'business' from 'technical' metadata [22, 23]. Ground proposes thinking of metadata as Application, Behavior, Context [28]. These definitions do not conform a MM, so they are orthogonal to our definition.

Cognitive Fit Theory. The theory of cognitive fit originates in information systems (IS) [47], where it has been used to inform conceptual schemas [33, 34], such as ER diagrams. To our knowledge the 5W1H+R MM is the first to use this theory to propose a MM for metadata management.

Schema Designs And Evolvability. We considered data vault as an evolvable schema but there are other alternatives. In fact, the main difference between alternatives such as star and snowflake schemas and data vault, is data vault explicitly captures who wrote what and when. This metadata about operations is precisely the information we argued in Section 3.2 and the reason for our choice.

Metadata Extraction. is complementary to storage. Aurum ingests and processes structured data to discover relationships between similar datasets, which it models as an enterprise knowledge graph [19]. Juneau [54] discovers related tables using workflows provided by Jupyter notebooks. Pytheas [12] discovers tables from CSV files using a flexible rule set. Survey work [1] has explained some useful dataset profiles, which we call metadata.

8 CONCLUSION

We proposed a new MM for metadata and discussed practical implementations of its materialization in a catalog. We designed the new MM informed by an in-depth study of existing catalog technology. We justified the use of the MM based on the cognitive fit theory and discuss data vault as a schema design amenable for data catalogs. We evaluated the new MM with a user study, and considered the implications of different schema designs and backend implementations. All in all, we consider our work to help shed some light in the vast area of metadata management, which is growing in importance and poses a big scalability challenge in organizations today.

REFERENCES

[1] Ziawasch Abedjan, Lukasz Golab, and Felix Naumann. 2015. Profiling Relational Data: A Survey. *The VLDB Journal* 24, 4 (Aug. 2015), 557–581. https://doi.org/10.1007/s00778-015-0389-y

[2] Assaf Araki and Ben Lorica. 2021. The Growing Importance of Metadata Management Systems. https://gradientflow.com/the-growing-importance-of-metadata-management-systems/?utm_campaign=DC_Thurs&utm_medium=email&_hsmi=111418530&_hsenc=p2ANqtz-9z0vq2HiKarPuqhQ2HZ47ftKjSkvTyQqBa4PhQUA3GGASaEIllFbSQDCoV9wMsENFILo9g43LGlO2h_7OE7PWm_1sy1A&utm_content=111418602&utm_source=hs_email

[3] Apache Atlas. 2020. Apache Atlas: Overview. https://atlas.apache.org/#/

[4] Shekhar Bapat. 2020. Discover, understand and manage your data with Data Catalog, now GA. https://cloud.google.com/blog/products/data-analytics/data-catalog-metadata-management-now-generally-available

[5] Xavier Becerra. 2021. California Consumer Privacy Act (CCPA). https://oag.ca.gov/privacy/ccpa

[6] Misha Benjamin, Paul Gagnon, Negar Rostamzadeh, Chris Pal, Yoshua Bengio, and Alex Shee. 2019. Towards Standardization of Data Licenses: The Montreal Data License. arXiv:1903.12262 [cs.CY]

[7] boomi. 2021. Data Catalog and Preparation. https://boomi.com/platform/data-catalog-and-preparation/#/706e6e/home

[8] John Carter. 2020. Data Vault Automation with erwin and Snowflake: Building and Automating a Scalable Data Warehouse Based on Data Vault 2.0. https://www.snowflake.com/blog/data-vault-automation-with-erwin-and-snowflake-building-and-automating-a-scalable-data-warehouse-based-on-data-vault-2-0/

[9] R. Castro Fernandez, Z. Abedjan, F. Koko, G. Yuan, S. Madden, and M. Stonebraker. 2018. Aurum: A Data Discovery System. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. 1001–1012. https://doi.org/10.1109/ICDE.2018.00094

[10] Ryan Chard, Zhuozhao Li, Kyle Chard, Logan T. Ward, Yadu N. Babuji, Anna Woodard, Steven Tuecke, Ben Blaiszik, Michael J. Franklin, and Ian T. Foster. 2018. DLHub: Model and Data Serving for Science. *CoRR* abs/1811.11213 (2018). arXiv:1811.11213 http://arxiv.org/abs/1811.11213

[11] Chen Chen, Behzad Golshan, Alon Y. Halevy, Wang Chiew Tan, and AnHai Doan. 2018. BigGorilla: An Open-Source Ecosystem for Data Preparation and Integration. *IEEE Data Eng. Bull.* 41 (2018), 10–22.

[12] Christina Christodoulakis, Eric B. Munson, Moshe Gabel, Angela Demke Brown, and Renée J. Miller. 2020. Pytheas: Pattern-Based Table Discovery in CSV Files. *Proc. VLDB Endow.* 13, 12 (July 2020), 2075–2089. https://doi.org/10.14778/3407790.3407810

[13] Collibra. 2020. What is a data catalog? https://www.collibra.com/data-catalog

[14] Mercé Crosas. 2011. The Dataverse Network®: An Open-Source Application for Sharing, Discovering and Preserving Data. 17, 1 (2011). https://doi.org/10.1045/january2011-crosas

[15] Jim Davies, Jeremy Gibbons, Steve Harris, and Charles Crichton. 2014. The CancerGrid Experience: Metadata-Based Model-Driven Engineering for Clinical Trials. *Science of Computer Programming* 89B (September 2014), 126–143. https://doi.org/10.1016/j.scico.2013.02.010

[16] Dong Deng, Raul Castro Fernandez, Ziawasch Abedjan, Sibo Wang, Michael Stonebraker, Ahmed K Elmagarmid, Ihab F Ilyas, Samuel Madden, Mourad Ouzzani, and Nan Tang. 2017. The Data Civilizer System.. In *CIDR*.

[17] denodo. 2021. Denodo Platform Overview: Denodo Platform goes beyond every other data virtualization solution. https://www.denodo.com/en/denodo-platform/overview

[18] Huanming Fang and Hui Miao. 2020. Introducing the Model Card Toolkit for Easier Model Transparency Reporting. https://ai.googleblog.com/2020/07/introducing-model-card-toolkit-for.html

[19] Raul Castro Fernandez, Ziawasch Abedjan, Famien Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. 2018. Aurum: A data discovery system. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 1001–1012.

[20] Leon Nelson Flint. 1917. *Newspaper writing in high schools, containing an outline for the use of teachers.* University of Kansas.

[21] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumeé III, and Kate Crawford. 2018. Datasheets for datasets. *arXiv preprint arXiv:1803.09010* (2018).

[22] Lars George. 2020. Technical vs. Business Metadata Management. https://www.okera.com/blogs/technical-vs-business-metadata-management/

[23] Google. 2020. Data Catalog. https://cloud.google.com/data-catalog#all-features

[24] Kathleen Gregory, Paul Groth, Andrea Scharnhorst, and Sally Wyatt. 2019. Lost or found? Discovering data needed for research. *arXiv preprint arXiv:1909.00464* (2019).

[25] Alon Halevy et al. 2016. Goods: Organizing Google's Datasets. In *SIGMOD*.

[26] Alon Halevy, Flip Korn, Natalya F Noy, Christopher Olston, Neoklis Polyzotis, Sudip Roy, and Steven Euijong Whang. 2016. Goods: Organizing google's datasets. In *Proceedings of the 2016 International Conference on Management of Data*. 795–806.

[27] Maurice H. Halstead. 1977. *Elements of Software Science (Operating and Programming Systems Series).* Elsevier Science Inc., USA.

[28] Joseph M Hellerstein, Vikram Sreekanti, Joseph E Gonzalez, James Dalton, Akon Dey, Sreyashi Nag, Krishna Ramachandran, Sudhanshu Arora, Arka Bhattacharyya, Shirshanka Das, et al. 2017. Ground: A Data Context Service.. In *CIDR*.

[29] Michael L. Hines, Thomas Morse, Michele Migliore, Nicholas T. Carnevale, and Gordon M. Shepherd. 2004. ModelDB: A Database to Support Computational Neuroscience. *Journal of computational neuroscience* 17, 1 (2004), 7–11. https://doi.org/10.1023/B:JCNS.0000023869.22017.2e 15218350[pmid].

[30] Sarah Holland, Ahmed Hosny, Sarah Newman, Joshua Joseph, and Kasia Chmielinski. 2018. The dataset nutrition label: A framework to drive higher data quality standards. *arXiv preprint arXiv:1805.03677* (2018).

[31] Jason Howell. 2019. Azure Data Catalog common scenarios. https://docs.microsoft.com/en-us/azure/data-catalog/data-catalog-common-scenarios

[32] Hans Hultgren. 2012. DATA VAULT MODELING GUIDE: Introductory Guide to Data Vault Modeling. https://hanshultgren.files.wordpress.com/2012/09/data-vault-modeling-guide.pdf

[33] V. Khatri, I. Vessey, S. Ram, and V. Ramesh. 2006. Cognitive fit between conceptual schemas and internal problem representations: the case of geospatio-temporal conceptual schema comprehension. *IEEE Transactions on Professional Communication* 49, 2 (2006), 109–127. https://doi.org/10.1109/TPC.2006.875091

[34] Vijay Khatri, Iris Vessey, V. Ramesh, Paul Clay, and Sung-Jin Park. 2006. Understanding Conceptual Schemas: Exploring the Role of Application and IS Domain Knowledge. *Information Systems Research* 17, 1 (2006), 81–99. http://www.jstor.org/stable/23015782

[35] Michelle Knight. 2017. What is Data Governance? https://www.dataversity.net/what-is-data-governance

[36] Richie Koch. 2020. Everything you need to know about GDPR compliance. https://gdpr.eu/compliance/

[37] Mars Lan. 2019. DataHub: A generalized metadata search and discovery tool. https://engineering.linkedin.com/blog/2019/data-hub

[38] Neil D. Lawrence. 2017. Data Readiness Levels. arXiv:1705.02245 [cs.DB]

[39] Sebastian Lawrenz, Priyanka Sharma, and Andreas Rausch. 2020. The significant role of metadata for data marketplaces. In *International Conference on Dublin Core and Metadata Applications*. 95–101.

[40] Dan Linstedt. 2015. Data Vault Basics. http://danlinstedt.com/solutions-2/data-vault-basics/

[41] Lyft. 2020. Open source data discovery and metadata engine. https://www.amundsen.io/

[42] Natasha Noy and Omar Benjelloun. 2020. An Analysis of Online Datasets Using Dataset Search (Published, in Part, as a Dataset). https://ai.googleblog.com/2020/08/an-analysis-of-online-datasets-using.html

[43] University of Michigan. 2021. ICPSR: Sharing data to advance science. https://www.icpsr.umich.edu/web/pages/

[44] Raymond K. Pon and David J. Buttler. 2009. *Metadata Registry, ISO/IEC 11179.* Springer US, Boston, MA, 1724–1727. https://doi.org/10.1007/978-0-387-39940-9_907

[45] Prolific. 2021. Prolific: Quickly find research participants you can trust. https://www.prolific.co

[46] SAP. 2021. SAP Business Technology Platform: Turn data chaos into data value with data intelligence. https://www.sap.com/products/data-intelligence.html

[47] Iris Vessey. 1991. Cognitive Fit: A Theory-Based Analysis of the Graphs Versus Tables Literature*. *Decision Sciences* 22, 2 (1991), 219–240. https://doi.org/10.1111/j.1540-5915.1991.tb00344.x arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-5915.1991.tb00344.x

[48] W3C. 2020. Data Catalog Vocabulary (DCAT) - Version 3. https://www.w3.org/TR/vocab-dcat-3/

[49] Dave Wells. 2020. A Data Architect's Guide to the Data Catalog. https://www.alation.com/blog/a-data-architects-guide-to-the-data-catalog/

[50] WeWork. 2021. Marquez: Collect, aggregate, and visualize a data ecosystem's metadata. https://marquezproject.github.io/marquez/

[51] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J.G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A.C 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3, 1 (15 Mar 2016), 160018. https://doi.org/10.1038/sdata.2016.18

[52] Mohamed Yakout, Kris Ganjam, Kaushik Chakrabarti, and Surajit Chaudhuri. 2012. Infogather: entity augmentation and attribute discovery by holistic matching with web tables. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. 97–108.

[53] Matei Zaharia, Andrew Chen, Aaron Davidson, Ali Ghodsi, Sue Ann Hong, Andy Konwinski, Siddharth Murching, Tomas Nykodym, Paul Ogilvie, Mani Parkhe, Fen Xie, and Corey Zumar. 2020. Accelerating the Machine Learning Lifecycle with MLflow. https://databricks.com/research/accelerating-the-machine-learning-lifecycle-with-mlflow

[54] Yi Zhang and Zachary G Ives. 2020. Finding Related Tables in Data Lakes for Interactive Data Science. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1951–1966.