

# Credit Distribution through Data Provenance in Relational Scientific Databases

Dennis Dosso<sup>a</sup>, Susan B. Davidson<sup>b</sup>, Gianmaria Silvello<sup>a</sup>

<sup>a</sup>*Department of Information Engineering, University of Padua, Italy*

<sup>b</sup>*Department of Computer and Information Science, University of Pennsylvania, USA*

---

## Abstract

Digital data is an important form of research product for which citation, and the generation of credit or recognition for authors, is still not well understood. The notion of *data credit* has therefore recently emerged as a new metric, defined and based on data citation theory.

Data credit is a real value that represents the importance of data cited by a paper or by another research entity. Credit can be used to annotate data contained in a curated scientific database, and used as a measure for the importance and impact of that data in the research world. As such, it is a new method that, together with traditional citations, helps recognize the value of data and its creators.

In this paper we explore the problem of Data Credit Distribution, the process by which credit is distributed to the database parts responsible for the production of data being cited by a research entity.

We adopt as use case the IUPHAR/BPS Guide to Pharmacology (GtoPdb), a widely-used curated scientific relational database. We define three new distribution strategies, the first two based on two forms of data provenance, why-provenance and how-provenance, and the third based on the concept of responsibility.

Using these distribution strategies we show how credit can highlight frequently used database areas and how it can be used as a new bibliometric measure for data and their corresponding curators. In particular, credit rewards data and authors based on their research impact, not merely on the number of citations. We also show how these distribution strategies vary in their sensitivity to the role of an input tuple in the generation of the output data, and reward input tuples differently.

*Keywords:* Data Citation, Data Credit

---

## 1. Introduction

Citations are an essential component of scientific research, enabling research products to be found as well as the relationships between them to be created and understood. They form a basis on which to give credit to authors, papers, and venues [21, 22, 60]. Citations are used, among other things, to decide on tenure, promotion, hiring, and funding of grants for researchers [23, 36, 41, 44].

Science and research are increasingly digital, and there are numerous curated databases that are at the core of scientific research efforts [13]. It is therefore generally accepted that data must be cited and citable [16, 42], and that data citations should contribute to the scientific reputation of researchers, scientists, data curators, and creators [4, 55]. It is also accepted that data citations should be counted alongside of traditional citations, and contribute to bibliometrics indicators [7, 49].

Many initiatives, at different levels, have been promoted to make data citation a reality. Scientific publishers, such as Elsevier, Springer and Nature, have been defining data policies and author guidelines to include data citations in the reference lists of published papers [21]. The European Commission has introduced the Open Research Data Pilot (ODP), whose aim is to improve and maximize the access and re-use of research data, together with an increase to the credit given to data creators and curators [53]. Initiatives such as FORCE11 and ESIP (Earth Science Information Partners) have collaborated on data and software citation principles and guidelines [29]. Other examples are the National Science Foundation (NSF), and the National Institute of Health (NIH) in the US [53].

Moreover, there are activities to promote and specify guidelines for data citations. A significant activity getting a broad adoption, is the Research Data Alliance (RDA), that produced a recommendation on citing specific subsets of dynamic data [52]. While this approach provides reference and access to a precise subset of data, it does not address specific credit concerns for that subset, such as when different authors contribute to a larger collection [48].

A central problem in the data citation process is how to attribute credit to data creators and curators [12]. How to handle and count the credit generated by data citation, and how it contributes to traditional and new

bibliometrics, are long-standing research issues [10, 31]. However, even when correctly applied, data citations and the bibliometrics computed using them do not always correctly or completely reward the creators of data used in a database. Data, in fact, is often cited at the “database level” or the “webpage level”. In the first case, the whole database is cited and therefore all credit goes to the key personnel of the database. In the second case, the database has a website with webpages that can be individually cited. The webpages use data extracted from the database, which is aggregated by topic and built to resemble a traditional research paper. Often the creators and curators of the webpage’s data are not credited or only marginally credited for their work [3].

Recently, the concepts of *data credit* and *Data Credit Distribution* (DCD) [30, 40, 59] have emerged, built on top of methodologies for data citation. Data credit is a value that is computed based on the importance of the data being cited in a paper, and represents the impact of the data on the citing paper. The DCD problem consists of distributing this credit to elements in the databases in the citation graph that are responsible for the generation of the data being cited. The goal of DCD is to improve and expand the reach of data citation, rather than being an alternative to it. This means that to employ DCD techniques, we need data citations in some form.

In this paper, we consider data credit as a measure of value for data in a (curated) scientific database. Credit is a real value that can be assigned to data of any kind and at any level of granularity. Therefore the concept of “data” is left intentionally vague, although in this paper we focus on relational databases. Credit is a positive *real* value, acting as a proxy for the value of data based on the measure of citations, accesses, clicks, downloads, or other surrogates for data use. We call DCD the process, method, or algorithm used to assign credit to a given datum or dataset.

The DCD problem differs from the traditional citation setting since:

1. When a paper  $p_1$  cites another paper  $p_2$ , a +1 citation “credit” is given to  $p_2$ , and to all its authors. It does not matter why or how paper  $p_1$  cites paper  $p_2$ <sup>1</sup>, the result is always +1 to the citation count of  $p_2$  and of its authors. A different credit distribution strategy can assign a quantity of credit to  $p_2$ , and its authors, that is *proportional* to the

---

<sup>1</sup>Note that there is vast research on this topic and many alternative proposals, but none of them currently work at a large scale.

- 70 role played by  $p_2$  in  $p_1$ . Hence, we can weight the importance of the  
 71 cited entities and assign credit according to their role.
- 72 2. Traditional citations are *atomic*: a citation from  $p_1$  to  $p_2$  can never  
 73 be broken into pieces and assigned in part to  $p_2$  and in part to other  
 74 papers or data that contributed to  $p_2$ . In contrast, with data credit,  
 75 we use a *non-atomic* real value, which can be divided and distributed  
 76 to multiple components of a database.
- 77 3. Credit can be *transitive*, that is, it can be propagated through one  
 78 cited entity to other entities cited by it that contributed to its content.  
 79 Citations, traditionally, are not.

80 We study the DCD problem in the context of relational databases (RDBs)  
 81 since they are widely used<sup>2</sup> and are the main focus of current work in data  
 82 citation methods [13, 15, 50]. RDBs are also frequently a test-bed for new  
 83 methods that can be adapted to other databases, e.g., graphs or document  
 84 databases. The “portions” of data in an RDB that can be credited can be  
 85 defined at different levels of granularity, in particular: (i) the whole database,  
 86 (ii) tables, (iii) tuples, and (iv) attributes. The ability to specify different  
 87 levels of granularity in a relational database allows us to define the DCD  
 88 problem at a particular level of granularity. In this paper, we focus on DCD  
 89 at the tuple level.

90 The DCD process is summarized in Figure 1:

91 **Step 1** Scientists and experts contribute the curated information contained  
 92 in a scientific database. These are called the “Data Curators”.

93 **Step 2** Other researchers use the data in their research, and when possible,  
 94 cite them.

95 **Step 3** The citation to the data generates credit, that can be used as a  
 96 proxy for the impact of the data on the citing paper. This credit is  
 97 represented as a real value  $k \in \mathbb{R}_{>0}$ .

98 **Step 4** Given the database instance  $I$  and the query  $Q$ , it is possible to  
 99 compute the *data provenance* of  $Q(I)$ . The provenance of  $Q(I)$  is a

---

<sup>2</sup>The “relational database market alone has revenue upwards of \$50B” [1].

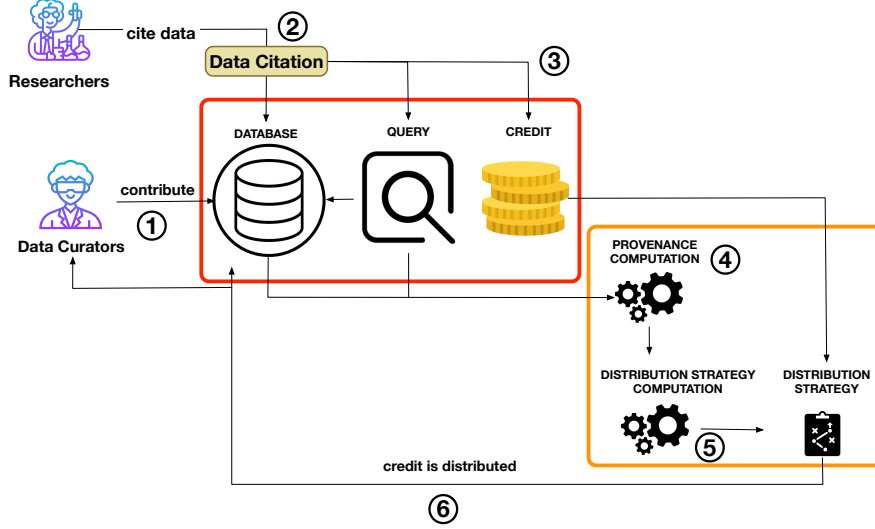


Figure 1: Overview of the credit distribution pipeline.

form of metadata that describes the generation process undertaken by  $Q$ , and the data used in  $I$  to generate the output [18]. Many different notions of provenance have been proposed in the literature for data in database management systems [14, 24, 33], describing different kinds of relationships between data in the input and the output of a query. As reported in [18], these provenances have been used in several applications beyond giving information on how queries work, for example, annotation propagation and the view update problem. In this paper, we consider three types of provenance: lineage, why-provenance, and how-provenance. Also, we consider the notions of causality and responsibility, that enrich the information provided by provenance. In the following, for simplicity of exposition, when we refer to provenance, we are also including responsibility with abuse of language.

**Step 5** Provenance is input to the DCD problem, whose aim is to compute the *Credit Distribution Strategy* (CDS, also referred only as Distribution Strategy, DS). The CDS is a function  $f$  that takes as input the credit value  $k$ , divides it and distributes it to the data in the input database  $I$ , and is defined on the basis of citation policies decided at the database administration level or at the domain community level. In this paper, since we base CDS on data provenance, we describe four

120 CDS, each one based on a different form of provenance.

121 **Step 6** Once the CDS is computed, it is used to distribute the given credit  
122  $k$  to the parts of the database that are responsible for the generation  
123 of  $Q(I)$ . Transitively, this credit is also divided and given to the corre-  
124 sponding authors of those data.

125 This paper expands our recent work in [26], which addressed the problem  
126 of how to reward data and data curators who are typically overlooked in  
127 current citation systems. In that work, we first defined the problem of DCD  
128 in relational databases, and proposed a viable Distribution Strategy (DS)  
129 based on *lineage*, which is the simplest form of *data provenance*. The lineage  
130 of a tuple  $t$  in the output  $Q(I)$  is defined as the set of all and only the tuples  
131 in the database instance  $I$  that are “relevant” to the production of  $t$ . The  
132 corresponding strategy equally redistributes the credit  $k$  to the tuples in the  
133 lineage set, thus each tuple receives credit  $k/|L_t|$ , where  $L_t$  is the lineage set  
134 of  $t$ .

135 One may argue that this DS is too simplistic, since lineage does not convey  
136 any information about the role or importance of input tuples in the query.  
137 Therefore, one may desire to give more credit to the tuples that are more  
138 *essential* to the production of the output, i.e. those tuples that, if removed,  
139 would prevent the output tuple from appearing in the final result, or those  
140 tuples used more than once by the query.

141 Therefore, in this paper, we expand the ideas in [26] by proposing three  
142 new DSs based on other forms of data provenance: why-provenance [14], how-  
143 provenance [33], and *responsibility* [45]. We use these form of provenances  
144 and the information that they carry to define new Distribution Strategies  
145 that have a different behavior with respect to the one of the lineage-based  
146 DS. We will show in the paper the formulas defined taking inspiration from  
147 the provenances and discuss their characteristics with experiments based on  
148 a real database, using both real and synthetic queries.

149 We compare these new DSs with the lineage-based solution, and discuss  
150 why one may be preferred to another depending on the application and its  
151 goals. In particular, we show that why-provenance, *responsibility* and how-  
152 provenance are more sensitive to the *role* of a tuple in a query, i.e. how many  
153 times the tuple is used and how it is used. The DSs based on why-provenance  
154 and *responsibility* reward more tuples that are essential to the production

155 of the result set, whereas the DS based on how-provenance also takes into  
156 consideration the different ways that a tuple is used.

157 For evaluation, we use a well-known curated database, the IUPHAR/BPS<sup>3</sup>  
158 Guide to Pharmacology [35], also known as GtoPdb<sup>4</sup>, which contains ex-  
159 pertly curated information about diseases, drugs, cellular drug targets, and  
160 their mechanisms of action. We chose GtoPdb for two main reasons: (i) it  
161 is a widely-used and valuable curated relational database, (ii) many papers  
162 in the literature use, and cite, its data (i.e., families, ligands, and receptors).  
163 Real queries used in papers can therefore be seen as data citations which, in  
164 turn, can be used to assign data credit.

165 We perform four sets of experiments. In the first one, real queries are ex-  
166 tracted from papers published in the British Journal of Pharmacology (BJP),  
167 that represent data citations to GtoPdb, and are used to distribute credit in  
168 the database using the three different provenance-based DSs. In the second  
169 and third experiment we analyze the behavior of the different DS when com-  
170 plex citation queries are employed. In the fourth set of experiments we use  
171 both real and synthetic queries to assess the difference between traditional  
172 citation and the notion of credit distribution in terms of rewarding those  
173 responsible for the data, e.g. data curators.

174 **Contributions** of this work include:

- 175 • Three new Distribution Strategies based on why-provenance, how-provenance,  
176 and responsibility.
- 177 • An in-depth analysis of the effects of credit distribution on real-world  
178 curated data and of the differences between the three proposed Distri-  
179 bution Strategies.
- 180 • A comparison between the behavior of traditional citations and data  
181 credit in rewarding data curators.

182 **Outline.** The rest of the paper is organized as follows: Section 2 presents  
183 the background and related work. Section 3 describes the GtoPdb use case  
184 we adopted. Section 4 briefly presents the forms of provenance used in the  
185 paper. Section 5 describes the credit distribution problem and the proposed

---

<sup>3</sup>International Union of Basic and Clinical Pharmacology/British Pharmacology Soci-  
ety

<sup>4</sup><https://www.guidetopharmacology.org/>

186 distribution strategies. In Section 6 we present the experimental evaluation.  
187 Finally, Section 8 draws some conclusions and outlines future work.

## 188 2. Background

189 *Data in Research.* The world of research is rapidly transitioning towards the  
190 *fourth paradigm of science* [37], that is, data-intensive scientific discovery,  
191 where data are important for scientific advances as well as for traditional  
192 publications [6].

193 The scientific community is promoting an *open research culture* [47],  
194 founded on methods and tools to share, discover, and access experimental  
195 data. The community has identified the FAIR principles (Findable, Acces-  
196 sible, Interoperable, and Reusable) [57], that should be enforced by every  
197 database. In particular, data should be accessible from the articles, journals,  
198 and papers that cite or use them [21]. Aspects such as the need for the *repro-*  
199 *ducibility* of experiments through the used data; the *availability* of scientific  
200 data; the *connections* between data and the scientific results are all needed  
201 aspects for the fourth paradigm, and are all relevant to the domain of *data*  
202 *citation* [38].

203 *Data Citation: Principles and Motivations.* Data Citation principles were  
204 proposed in [20], and later summarized and endorsed by the Joint Declaration  
205 of Data Citation Principles (JDDCP) [43]. The principles are divided into  
206 two groups [53]. The first one contains principles concerning the role of  
207 data citation in scholarly and research activities such as the (i) *importance*  
208 of data (why data citation is important and why data should be considered  
209 as first-class citizens); (ii) *credit* and *attribution* to the creators and curators  
210 of the data; (iii) *evidence*; (iv) *verifiability*; and *interoperability*, with these  
211 last three requiring data citation methods to be flexible enough to operate  
212 through different communities. The second group defines the main guidelines  
213 to establish a data citation systems, and contains principles such as the (i)  
214 *unique identification* of the data being cited; (ii) *(open) access* to data; (iii)  
215 guarantee of *persistence* and *availability* of citations even after the lifespan  
216 of the cited entity; the (iv) *specificity* of a citation, i.e. it must lead to the  
217 data set originally cited.

218 It is possible to outline six main motivations for data citation [53]:

- 219 • *Data attribution:* identify the individuals that should be credited for  
220 data with variable granularity.



- 221 • *Data connection*: connect papers to the data being used.
- 222 • *Data Discovery*: citations helps to find data records and subsets that
- 223 would be otherwise not findable via search engines.
- 224 • *Data Sharing*: share data obtained by researchers within the whole
- 225 community.
- 226 • *Data Impact*: highlight the results obtained in writing papers using
- 227 specific data, the frequency and modality data were used.
- 228 • *Reproducibility*: data citation greatly impacts the reproducibility of
- 229 science [5]. Many authoritative journals ask to share data and provide
- 230 valid methodologies to reproduce experiments.

## 231 2.1. Data Citation in Relational Databases

232 In this paper, we develop our methods and experiments on relational  
 233 databases. RDBs have been the main target of data citation methods since  
 234 the surge of the data-centric research paradigm. The RDA “Working Group  
 235 on Data Citation: Making Dynamic Data Citable”<sup>5</sup> [51] has been working in  
 236 the last years on large, dynamic, and changing datasets. The working group  
 237 has finished the development of its guidelines and has now moved on into an  
 238 adoption phase. The datasets considered by the Working Group are often  
 239 relational.

240 In one of its most recent sessions [52], the Working Group (WG) on  
 241 Data Citation reported that there are various implementations of its guide-  
 242 lines for Data Citation on MySQL/Postgres relational databases. Some of  
 243 these databases are: DEXHELPP<sup>6</sup> (Social Security Records); NERC (ARGO  
 244 Global Array); EODC (Earth Observation Data Centre) [32]; LNEC (River  
 245 dam monitoring); MDS (Million Song Database) [8]; CBMI<sup>7</sup> (Center for  
 246 Biomedical Informatics); VMC (Vermont Monitoring Cooperative); CCA<sup>8</sup>  
 247 (Climate Change Center Austria); VAMDC (Virtual Atomic and Molecular  
 248 Data Center) [27, 61].

---

<sup>5</sup><https://www.rd-alliance.org/groups/data-citation-wg.html>

<sup>6</sup><http://www.dexhelpp.at/>

<sup>7</sup><https://medicine.missouri.edu/centers-institutes-labs/center-for-biomedical-informatics>

<sup>8</sup><https://ccca.ac.at/startseite>

249 More examples of work on data citation in relational databases are [2, 13,  
250 25, 58]. The website <https://fairsharing.org/> keeps a long updated list  
251 of curated and scientific databases (many of which are relational or graph-  
252 based) following FAIR guidelines. These databases are citable since they are  
253 compliant with the most recent guidelines, and they are in the vast majority  
254 of cases accessible via dynamically created Webpages. In all these databases  
255 it is, therefore, possible to implement DCD on top of the existing infrastruc-  
256 tures for citing data.

257 Data citation techniques are primarily applied to relational databases  
258 because of their diffusion and also because the portions of data that are to  
259 be cited are easily identified: the whole database, a relation, a tuple, or  
260 even an attribute. Many papers [2, 11, 13] consider more complex citable  
261 units, recognizing that often the *views* of a database are the ones to be cited.  
262 Generally, a *view* is a query on the database. To this end, [58] suggested  
263 decomposing the database in a set of views, where each view is associated  
264 with its citation.

265 At present, the most common practices to cite databases include:

- 266 1. A database cited as a whole, even though only parts of the databases  
267 are used in the papers or datasets. Alternatively, the so-called “data pa-  
268 pers” are cited, being traditional papers that describe a database [17].  
269 In this case, all the credit from the citations goes to the database ad-  
270 ministrators or to the authors of the data papers.
- 271 2. Subsets of data, obtained by issuing queries to a database, are individ-  
272 ually cited. This is the solution adopted by the *Resource Data Alliance*  
273 (RDA) working group on Data Citation [51]. In this case, the credit  
274 generated from citations is distributed among the contributors of the  
275 portions of data being cited, and/or to the database administrators.
- 276 3. The database is accessible via a series of Webpages that arrange the  
277 content of the database by topic or theme. Examples in the life science  
278 domain include the Reactome Pathway database [39], the GtoPdb [35],  
279 and the VAMDC [61]. Every single Webpage is unequivocally identifi-  
280 able and can be individually cited.

## 281 2.2. Data Credit

282 Data credit is related to data citation: they both aim to recognize the  
283 work of data creators and curators. Data credit can therefore also be seen as  
284 a by-product of data citation, since credit attribution is impossible without  
285 the presence of data citations.

286 Katz [40] suggests the need for a *modified citation system* that includes  
 287 the idea of *transient* and *fractional credit*, to be used by developers of research  
 288 products as software and data. In the paper two considerations are made:  
 289 (i) research objects such as data and software are currently not formally  
 290 rewarded or recognized by the community; (ii) even in traditional papers,  
 291 the contribution of each author to the work is hard to understand, unless  
 292 explicitly specified in the paper. This is even more true for data, where  
 293 different groups of people work on the same database.

294 In [40] credit is defined as a “quantity” that describes the importance of a  
 295 research entity, such as papers, software, or data, mentioned in a citation. It  
 296 also proposed the idea of a *distribution* of credit from research entities, such  
 297 as papers or data, to other research entities through citations. *Therefore,*  
 298 *when talking about data credit, there are two main aspects to consider:*  
 299 *credit computation*, the process by which the quantity of credit generated by  
 300 the citation is computed, and *credit distribution*, the process by which credit  
 301 is distributed and assigned to the responsible entities that contributed to the  
 302 generation of the data being cited. In this paper we focus on the latter.

303 These two processes are done by exploiting the structure of the *citation*  
 304 *graph*, a directed graph whose nodes are publications and edges are citations.  
 305 This graph is the model at the core of systems such as Google Scholar and  
 306 the Web of Science. We add to this that the concept of credit can be built  
 307 on top of the existing infrastructure handling traditional and data citations.

308 Katz [40] further explores the idea of a *distribution* of credit from research  
 309 entities (i.e., papers and data) to other research entities through citations  
 310 that connect them. Thanks to traditional citations and now also to data  
 311 citations, this distribution is finally possible, at least between papers and  
 312 data. Some problems related to traditional citations can thus be solved by  
 313 citations:

- 314 1. Credit rewards research entities that to date are not (formally) recog-  
 315 nized (a goal shared with data citation).
- 316 2. Credit can reward authors *proportionally* to their role in generating the  
 317 entity. The more an author contributes to a paper, the more credit is  
 318 given to him. Zou and Peterson [60] work on something similar with  
 319 their zp-index, which includes in its formulation the position (and thus  
 320 the role) of a publication author to represent its impact in the work  
 321 itself.
- 322 3. Credit can be *transitively* channeled through a chain of papers citing

each other, thus enabling the rewarding of older papers that are no more cited, since other papers summarize or report their content but are nevertheless crucial in a research area for the influence of their content.

Fang [30] presents a framework to distribute the credit generated by a paper to its authors and to the papers in its reference list in a transitive way. Let us consider the *citation graph* as the graph where the nodes are papers and the links are the citations among them. In this graph, every paper is a source of credit, which is then transferred to the neighboring nodes. The quantity of credit received by each cited paper depends on its impact/role in the citing paper. So far, this theoretical framework is limited to papers, but it can be easily extended to a citation graph including both papers and data.

Zeng et al. [59] proposes the first method to compute credit within a network of papers citing data. Adopting a network flow algorithm, they simulate a random walker to estimate a score for each dataset, leveraging real-world usage data to compute the credit. This is the first step towards an automatic credit computation procedure. This proposal is, however, limited to assigning credit to whole datasets, and it does not deal with the granularity of data. It does not work to assign credit to a single research entity within a dataset. Differently from Zeng et al. [59], we do not treat the credit computation process, but we focus on the distribution process.

### 2.3. Data Provenance

To distribute credit, we base our methods on *data provenance*. Data provenance is information that describes the origin and the process of creation of data. It can also be seen as metadata pertaining to the derivation history of the data. It is particularly useful to help users to understand where data are coming from, and the process they went through. Data citation and data provenance are closely linked [3] since both are forms of annotations on data retrieved through queries. Data provenance has been widely studied in different areas of data management. In this paper, we focus on provenance for database management systems (DBMS). For further details on data provenance, please refer to surveys like [18] and [54].

Cheney et al. [18] presents four main types of data citation for DBMS: *lineage* [24], *why-provenance* [14], *how-provenance* [33] and *where-provenance* [14].

358 Let us start with the first three provenances. Given a database instance  
359  $I$ , a query  $Q$ , and the result  $Q(D)$ , consider one tuple  $t$  of the output. Its  
360 provenance is information about its generation through the tuples of the  
361 input that are used by  $Q$ . Different types of provenance convey different  
362 levels of information. Since these three provenances are computed for each  
363 tuple of the output, they are also referred to as *tuple-based*.

364 Where-provenance, differently from the other three, is *attribute-based*, so  
365 we do not take it into account in this work since we consider the tuple as the  
366 finest citable unit.

367 We also consider the notions of causality and responsibility, as defined  
368 in [45]. Causality is an enrichment of lineage, and it is the attribution of  
369 a certain degree of importance to the tuples of the lineage based on their  
370 role in the generation of the output. Responsibility is a value given to the  
371 tuples of the lineage to rank them based on their degree of causality (the  
372 more important the role of a tuple in generating the output, the higher its  
373 responsibility).

### 374 3. Use Case: GtoPdb

375 As use case we refer to the IUPHAR/BPS Guide to Pharmacology [35]  
376 or GtoPdb<sup>9</sup>. GtoPdb is a well-known and well structured scientific relational  
377 database that contains expertly curated information about diseases, drugs  
378 in clinical use, their cellular targets, and the mechanisms of action on the  
379 human body. It is curated and maintained by the GtoPdb Committee, and  
380 by 96 subcommittees, comprising 512 scientists collaborating with in-house  
381 curators who draw the information contained in the database from high-  
382 quality pharmacological and medicinal chemistry literature. Roughly 1000  
383 researchers from all over the world have contributed to the database, and the  
384 curators wanted to give recognition to these contributors. This led to some  
385 early work on data citation [11].

386 GtoPdb is relational, but its logical structure is hierarchical as shown  
387 in Figure 2. The information contained in the database is also organized  
388 into webpages focused on specific diseases, targets or ligands, and families  
389 for easier access by users. As depicted in Figure 2, the database can be  
390 thought of as a tree where the root is the database; the first level consists

---

<sup>9</sup><https://www.guidetopharmacology.org/>

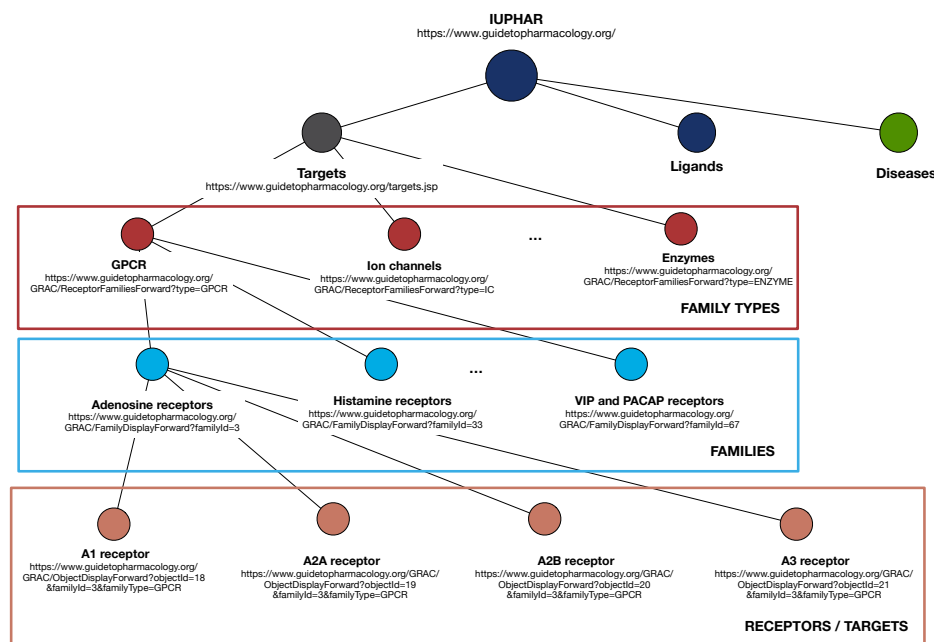


Figure 2: Partial map of the GtoPdb hierarchical structure grouping the targets into families and family types.

of all targets, ligands, and diseases; and the lower levels consists of specific targets, ligands and diseases. In this paper, we focus on targets; thus the figure at the third level shows examples of family types, at the fourth level of specific families of targets (a finer level of granularity), and finally, at the last level, the single targets (also known as receptors).

GtoPdb provides access to the webpages corresponding to all these nodes through URLs. The webpages corresponding to target families all present a similar structure, as shown in Figure 3 for the “Adenosine receptors” family. Each page has an *Overview*, a brief text describing the content of the page; a list of *Receptors* comprising the family; a section of *comments* about the family; the *References*, a list of the papers consulted by the curators of the page, similar to a reference list of a paper; the *further reading* list, reporting papers that an interested reader may want to consult to obtain more insight on the family; and a final section called *How to cite this family page*, containing text snippets useful to cite the specific page or the whole database. Figure 3 shows the SQL code that retrieves the information used to build the corresponding sections (apart from the References section). Therefore, each

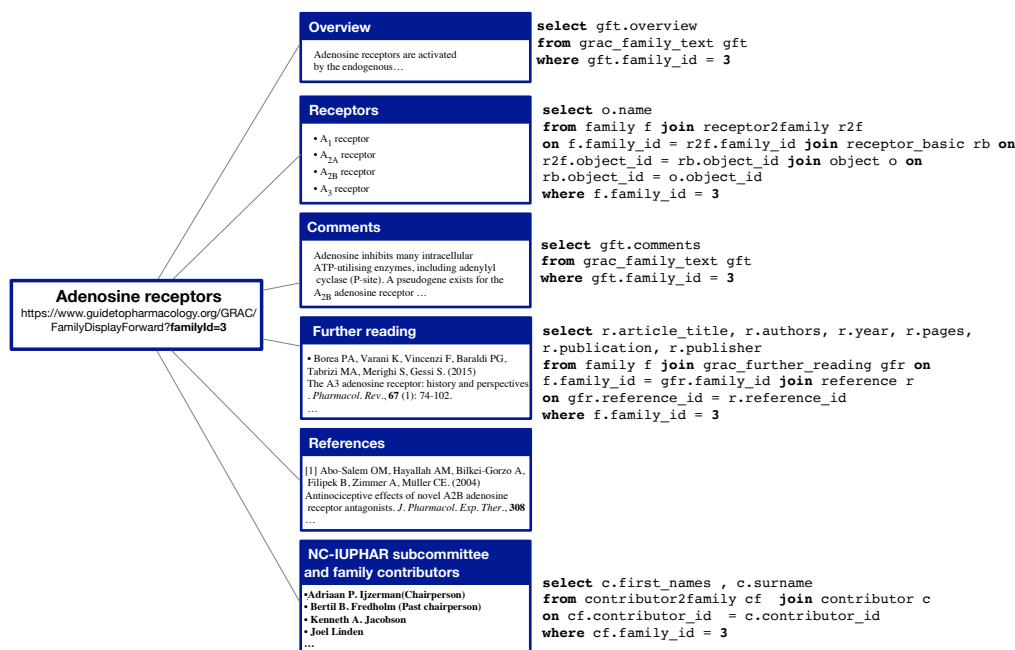


Figure 3: Basic web-page structure of “Adenosine receptors” family (ID 3), with queries used to retrieve the information contained in every section, except references.

family page can be considered a full-fledged traditional publication, consisting of title, authors, abstract (the overview), content, and references.

In practice, many papers in the literature only reference GtoPdb (the root) without including a reference to the specific page being cited. That is, they only cite a paper describing GtoPdb as a whole (e.g., [35]) and refer to targets, ligands, diseases, etc. only by name. Thus, citations to specific families are *de-facto* “hidden” to citation systems such as Google Scholar, and useless for the computation of bibliometrics.

In certain “lucky” cases, as with papers available in PDF and published in the British Journal of Clinical Pharmacology<sup>10</sup> (BJCP), when a family, ligand, receptor name, etc. are used, they have a hyperlink pointing to the corresponding webpage in GtoPdb. Therefore, the citations to the families can be detected and counted using the URLs reported in the papers. However, these citations to GtoPdb webpages are not counted as such by citation

<sup>10</sup><https://bpspubs.onlinelibrary.wiley.com/journal/13652125>

family			contributor2family		
id	name	type	id	family_id	contributor_id
$f_1$	Dopamine Receptors	gpcr	$c_2f_1$	$f_1$	$c_1$
$f_2$	Bile Acid Receptor	gpcr	$c_2f_2$	$f_1$	$c_2$
$f_3$	FAK Family	enzyme	$c_2f_3$	$f_2$	$c_3$
$f_4$	YANK Family	enzyme	$c_2f_4$	$f_4$	$c_1$

contributor		
id	Name	Country
$c_1$	John Smith	UK
$c_2$	Jim Doe	UK
$c_3$	Hans Zimmerman	Germany
$c_4$	Roberta Rossi	Italy

Table 1: Example of a database consisting of three tables. **family** includes some receptor families in the database; **contributor** contains the name and country of contributors; **contributor2family** connects contributors to the families they contributed to.

422 systems, so they are not converted into credit for curators and collaborators.  
423 For our running example, consider Table 1. This simplified version of  
424 GtoPdb illustrates three tables: **family**, **contributor** and **contributor2family**.  
425 The first table, **family**, has tuples representing families with three attributes:  
426 the id of the family, its name, and type. Table **contributor** consists of peo-  
427 ple who have helped generate the data of the database. The third table,  
428 **contributor2family**, serves as a link between the families and the people  
429 who contributed to them. For instance, “John Smith” ( $c_1$ ) contributed to  
430 “Dopamine Receptors” ( $f_1$ ) as well as to the “YANK Family” ( $f_4$ ). We use  
431 this example throughout the rest of the paper. In particular, we are using  
432 the id attribute of the tables as *provenance token* of its corresponding tu-  
433 ples, that is, as a symbol that serves to identify a tuple when talking about  
434 provenance.

#### 435 4. Data Provenances

436 In this section, we present the three types of provenance used in this  
437 paper: lineage, why-provenance, and how-provenance. Also, we present the  
438 notions of Causality and Responsibility.



#### 4.1. Lineage

Lineage was first introduced by Cui et al. [24]. Here we follow its definition as given by Cheney et al. [18]. Given a database instance  $I$ , the query  $Q$ , and the result  $Q(D)$ , consider on tuple  $t$  in this output. Lineage is the simplest among the forms of provenance. It has been defined in different ways [18], but it can be thought as the set of all the tuples are used in some way by the query to produce the output tuple, i.e., the ones that are somehow *relevant* to its generation.

As an example, consider the following SQL query Q1, applied to the database described in Table 1, that asks for the names of families curated by researchers based in the United Kingdom (UK):

```
Q1: SELECT DISTINCT f.name
FROM family AS f JOIN contributor2family AS c2f
ON f.id = c2f.family_id
JOIN contributor AS c ON c2f.contributor_id = c.id
WHERE c.country = 'UK'
```

id	name	lineage
$o_1$	Dopamine Receptors	$\{f_1, c2f_1, c_1, c2f_2, c_2\}$
$o_2$	YANK Family	$\{f_4, c2f_4, c_1\}$

Table 2: Result of an SQL query applied to the database instance in Table 1, which asks for the names of families curated by a researcher based in the UK. Attribute `id` is not part of the output and was added to succinctly identify each tuple as provenance token. Each tuple is also annotated with its lineage.

Table 2 shows the query result set, which consists of two tuples. We add an extra attribute `id` so that we can easily refer to each result tuple. The lineage for tuple  $o_1$  is the set  $\{f_1, c2f_1, c_1, c2f_2, c_2\}$ , since the tuple  $f_1$  was joined with  $c2f_1$  and then with  $c_1$ , and was also joined with  $c2f_2$  and  $c_2$ . No other tuple is used in the database to produce  $o_1$ . For tuple  $o_2$  the lineage is  $\{f_4, c2f_4, c_1\}$ . Lineage is defined for each tuple of the output, and can differ between tuples.

#### 4.2. Why-Provenance

Why-Provenance was first defined in terms of a deterministic semistructured data model and query language [14]. While why-provenance can be defined in many ways, we refer to [18], where it is expressed in terms of the relational model using the relational algebra.

467 In particular, while lineage aims to find all and only the tuples in the input  
 468 relevant to the production of an output tuple, why-provenance aims to find  
 469 sub-instances of the input that “witness” a part of the output. Given a tuple  
 470  $t$  in the query’s output, a *witness* is any sub-instance of the database that  
 471 produces  $t$ , i.e., a set that guarantees the existence of  $t$  in  $Q(D)$ . In particular,  
 472 the whole database and the lineage of  $t$  are both examples of witnesses of  $t$ .  
 473 Since the definition of witness allows for the presence of “irrelevant” tuples,  
 474 the set of all witnesses is finite (since the database instance  $I$  is finite), but  
 475 it is potentially exponentially large [18].

476 Buneman et al. [14] defined the why-provenance of an output tuple  $t$  in  
 477 the result  $Q(I)$  as a special *subset* of the set of witnesses called the *witness*  
 478 *basis*. The witnesses of the basis depend on  $Q$ ; thus, each basis’s size is  
 479 bounded by the size of  $Q$ . The witnesses of the basis exclude tuples that  
 480 are irrelevant to  $t$  being produced by  $Q$ , and thus the basis tends to be very  
 481 small compared to the set of all possible witnesses [18].

id	name	why-provenance
$o_1$	Dopamine Receptors	$\{\{f_1, c2f_1, c_1\}, \{f_1, c2f_2, c_2\}\}$
$o_2$	YANK Family	$\{\{f_4, c2f_4, c_1\}\}$

Table 3: Result of a SQL query applied on the database of Table 1 with the why-provenance of the corresponding results.

482 In a sense, each witness in the witness basis captures one possible way  
 483 in which the query can generate the output. To better understand this,  
 484 consider the example in Table 3, where each tuple in the result of query Q1  
 485 is annotated with its why-provenance.

486 The why-provenance of output tuple  $o_2$  has only one witness, which co-  
 487 incides with its lineage. This happens because there is only one way this  
 488 output tuple can be produced, i.e., for tuple  $f_4$  to be joined with  $c2f_4$  and  $c_1$ .  
 489 On the other hand,  $o_1$  has a witness basis of two witnesses, since there are  
 490 two possible ways in which the query can generate  $o_1$ . One possibility is that  
 491  $f_1$  is joined with  $c2f_1$  and  $c_1$  (the first witness), and the second possibility  
 492 is that  $f_1$  is joined with  $c2f_2$  and  $c_2$  (the second witness). This means that  
 493 to generate  $o_1$ , it is sufficient that only one of the two witnesses is present in  
 494 the input database.

id	name	how-provenance
$o_1$	Dopamine Receptors	$f_1 \cdot c_2 f_1 \cdot c_1 + f_1 \cdot c_2 f_2 \cdot c_2$
$o_2$	YANK Family	$f_4 \cdot c_2 f_4 \cdot c_1$

Table 4: Result of the example SQL query **Q1** with the corresponding how-provenances of the output tuples annotated.

### 4.3. How-Provenance

While why-provenance describes the source tuples that witness an output tuple in the result of the query, it leaves out information about how the source tuples are used. How-provenance was therefore defined in [33] to capture this information using a *semiring* algebraic structure. It takes the form of a polynomial, called *provenance polynomial*, where the variables are taken from the set  $X$  of identifiers of the tuples (provided that each tuple in  $I$  has an identifier) and the coefficients are drawn from the set of natural numbers  $\mathbb{N}$ . This semiring therefore is commonly referred as  $\mathbb{N}[X]$  in the literature.

The key idea in Green et al. [33] is to use the two operators  $+$  and  $\cdot$  to represent two basic transformations that source tuples undergo as a result of applying a relational query to a database [18]. Two tuples may either be joined together, as an effect of a join (represented with the  $\cdot$  operator) or merged via union or projection (represented with the  $+$  operator).

Table 4 shows the two output tuples of our running example annotated with their respective how-provenances. Tuple  $o_2$  was produced through the join among the input tuples  $f_4$ ,  $c_2 f_4$ , and  $c_1$ . The three provenance tokens are, therefore “multiplied” together. The case of  $o_1$  is slightly more complex. This tuple, as already discussed, can be obtained through two different joins. The two monomials composing the polynomial represent these two alternatives. They correspond, in a way, to the witnesses of the why-provenance of  $o_1$ . The  $+$  operator represents the fact that the two monomials describe alternative derivations. The output tuple is the result of a merge of two distinct tuples after the projection on the attribute **name**. This merge is due to the fact that the result of a relational algebra expression is always a *set* of tuples, which corresponds to the presence of the **DISTINCT** operator in an SQL query. This simple example gives the basic idea behind how-provenance and how it allows us to track the operations that produced an output tuple.

Provenance polynomials may also have monomials whose exponents and/or coefficients are greater than one, for example,  $3f_1 \cdot c_2 f_1 \cdot c_1 + f_1 \cdot c_2 f_2^3 \cdot c_2^3$ . This is a polynomial of a tuple produced by a query where the result of the

join between the tuples  $f_1$ ,  $c2f_1$ , and  $c_1$  is produced three times and then merged (e.g. as the result of a union), and the tuples  $c2f_2$  and  $c_2$  are used three times in the operation described by the second monomial (e.g., with nested queries).

#### 4.4. Causality and Responsibility

A formal study of causality was initiated in [19, 34] and later expanded by Meliou et al. [45] to define the causes of answers and non-answers to queries. Causality is, more precisely, related to the provenance of a query result such as lineage and adds information to the one already provided by the provenance.

In the following we define causality and responsibility as done in [45]. Differently from [45], we only focus on answers of a query, and not on non-answers, since they are not relevant in the context of this paper. Let  $D$  be a database instance and  $q$  a conjunctive query, let  $D^n \subseteq D$  be the set of *endogenous tuples*, i.e. the tuples being actually considered to be possible causes of a query output; while  $D^x = D - D^n$  is the set of *exogenous tuples*, the tuples being considered external, unconcerned factors, thus deemed not to be possible causes. This distinction between endogenous and exogenous tuple is application dependent, and it can be done by the user at query time.

Given a tuple  $\bar{a}$  with the same arity as the query's answer, we write  $D \models q(\bar{a})$  when  $\bar{a}$  is an answer to  $q$  on  $D$ , and write  $D \not\models q(\bar{a})$  when  $\bar{a}$  is a non-answer to  $q$  on  $D$ . Causality is defined as follows:

**Definition 4.1.** *Causality [45]*

*Let  $t \in D^n$  be an endogenous tuple, and  $\bar{a}$  a possible answer for  $q$ . Then:*

1.  *$t$  is called a counterfactual cause for  $\bar{a}$  in  $D$  if  $D \models q(\bar{a})$  and  $D - \{t\} \not\models q(\bar{a})$*
2.  *$t \in D$  is called an actual cause for  $\bar{a}$  if there exists a set  $\Gamma \subseteq D^n$ , called contingency for  $t$ , such that  $t$  is a counterfactual cause for  $\bar{a}$  in  $D - \Gamma$ .*

$t$  is a *counterfactual cause* if, by removing it from the database, we remove  $\bar{a}$  from the answer. Therefore, it can be thought as a tuple of the lineage which is fundamental for the presence of  $\bar{a}$  in the answer. Vice-versa,  $t$  is an *actual cause* if it is possible to find a contingency set of tuples such that, if that set is removed, only then  $t$  becomes counterfactual. In other words, when  $t$  is an actual cause, even if it was removed from the database,  $\bar{a}$  would still be present in the result set thanks to the contingency set. Computing the

id	name	responsibility
$o_1$	Dopamine Receptors	$f_1 = 1, c_2f_1 = 0.5, c_2f_2 = 0.5, c_1 = 0.5, c_2 = 0.5$
$o_2$	YANK Family	$f_4 = 1, c_2f_4 = 1, c_1 = 1$

Table 5: Result of the example SQL query **Q1** with the corresponding responsibilities of the lineage tuples.

causality of tuples is NP-complete in general [28], but Meliou et al. [45] proved that the causality of conjunctive queries may be determined in PTIME.

The notion of *responsibility* was first defined in [19], and it measure the degree of causality as a function of the size of the smallest contingency set. It allows to rank the tuples in a lineage based on their degree of causality in generating the output.

**Definition 4.2.** *Responsibility* [45]

Let  $\bar{a}$  be an answer to a query  $q$ , and let  $t$  be a cause. The responsibility of  $t$  for the answer  $\bar{a}$  is:

$$\rho_t = \frac{1}{1 + \min_{\Gamma} |\Gamma|}$$

where  $\Gamma$  ranges over all contingency sets for  $t$ .

As can be seen, a counterfactual cause will have the maximum responsibility of 1, while the bigger the minimum contingency of an actual cause, the smaller its responsibility since more tuples can still guarantee the presence of the answer  $\bar{a}$ .

While in general computing the responsibility is hard [19], Meliou et al. [45] showed that for each query without self-joins the responsibility is either computed in PTIME in the size of the database or checking if it has a responsibility below a given value is NP-hard.

As an example, consider Table 4, where we reported the result set of **Q1** and the tuples of the lineages with their responsibility values. Focusing on  $o_1$ : the tuple  $f_1$  of the lineage is a counterfactual cause, since its contingency set is empty (when removed from the database,  $o_1$  disappears from the result set). Consequently, its responsibility is 1. All the other tuples of the lineage are actual causes.  $c_1$ , for example, has as minimal contingency set  $\{c_2f_2\}$ , thus its responsibility is 0.5. For the output tuple  $o_2$ , all the tuples of the lineage are counterfactual causes, thus their responsibility is 1.

## 588 5. Credit Distribution and Distribution Strategies

589 We now give formal definitions of data credit and Data Credit Dis-  
 590 tribution (DCD), and present three different Distribution Strategies (DSs)  
 591 based on the forms of provenance discussed earlier: Lineage-based DS, Why-  
 592 Provenance-based DS, How-Provenance-based DS, and responsibility-based  
 593 DS. We also show how these strategies distribute credit in the IUPHAR  
 594 example discussed earlier.

### 595 5.1. Data Credit and Data Credit Distribution

596 Given a database instance  $I$ , a *recipient of credit* is a unit of information  
 597 within  $I$ . In the case of relational databases, recipients may be (i) the whole  
 598 database; (ii) a table; (iii) a tuple; or (iv) an attribute.

599 *Data credit* is a value  $k \in \mathbb{R}_{>0}$ . Every recipient in a database is annotated  
 600 with a quantity of credit as a proxy for its importance. In this paper, we  
 601 focus on *tuples* as recipients of credit.

602 Given a *distribution strategy* (DS), *Data Credit Distribution* (DCD) takes  
 603 a database instance  $I$ , a quantity of credit  $k$ , and query  $Q$  over  $I$ , and it splits  
 604  $k$  among the recipients of credit in  $I$ .

605 In the following, we use the notation in Cheney et al. [18]: Given an  
 606 instance  $I$ , a *tuple location*  $(R, t)$  is a tuple  $t$  in relation  $R$ . With reference to  
 607 the running example,  $(\text{family}, \langle f_1, \text{Dopamine Receptors}, \text{gpcr} \rangle)$  is the  
 608 tuple location of the first tuple in the **family** relation. The set of all tuple  
 609 locations in  $I$  is called *TupleLoc*. We use this to formally define DCD at the  
 610 *tuple level*.

611 **Definition 5.1. Tuple Level Data Credit Distribution (DCD) [26]**  
 612 *Given a query  $Q$  over  $I$  and  $k \in \mathbb{R}_{>0}$ , DCD is defined by the function  $f_{I,Q} :$   
 613  $\text{TupleLoc} \times \mathbb{R}_{>0} \rightarrow \mathbb{R}_{\geq 0}$  such that  $f_{I,Q}(t, k) = h$  where  $0 \leq h \leq k$  and  
 614  $\sum_{t \in \text{TupleLoc}} f_{I,Q}(t, k) = k$ . The function  $f_{I,Q}$  is the distribution strategy (DS).*

615 As we can see, the DS is a function that annotates each tuple in the  
 616 database with a real value, which is a fraction of the given quantity  $k$ . The  
 617 only constraint is that the sum of the credit annotations on tuples must be  
 618  $k$ , i.e. that no credit is generated or destroyed during the distribution. Given  
 619  $I$  and  $Q$ , many different DSs may be defined as long as they sum up to  $k$ .

620 In what follows, we use information provided by data provenance to de-  
 621 fine distribution functions. For simplicity, we assume that the credit  $k$  is  
 622 distributed equally across the set of output tuples (i.e. the result of a query),

623 and discuss how the credit of one output tuple  $o$ ,  $k_o$ , is distributed across the  
 624 instance  $I$ .

## 625 5.2. A Lineage-based Distribution Strategy

626 In the lineage-based distribution strategy, each tuple in the output of  
 627 a query distributes credit equally to each input tuple that appears in its  
 628 lineage. More formally:

**Definition 5.2.** *Lineage-based Distribution Strategy [26]*

*Let  $I$  be a database instance,  $Q$  a query over  $I$ ,  $o \in Q(I)$  an output tuple and  $k_o$  the credit associated to  $o$ . Let  $L$  be the lineage of  $o$  and  $t$  be a tuple in  $I$ , then  $t$  receives credit equal to:*

$$f_{I,Q}(t, k_o) = \begin{cases} 0 & \text{if } t \notin L \\ \frac{k_o}{|L|} & \text{if } t \in L \end{cases}$$

629 Note that lineage-based DS distributes credit only to input tuples that  
 630 have a role in creating  $o$  by the query  $Q$ , and that each receives an equal  
 631 share of credit. Thus, the more tuples in a lineage set, the less credit each  
 632 tuple receives.

633 As an example, consider the output tuples of Table 2, and assume that  
 634 each output tuple has credit  $k_o = 1$ . The lineage of the first tuple,  $o_1$ , is  
 635 the set  $\{f_1, c2f_1, c_1, c2f_2, c_2\}$ . Therefore, each tuple in this set receives credit  
 636  $1/5$ . The other tuples of the database receive zero credit. The lineage of the  
 637 second output tuple is  $\{f_4, c2f_4, c_1\}$ , therefore each of these tuples receives  
 638 credit  $1/3$ .

639 At the end of the process, tuples  $f_1$ ,  $c2f_2$  and  $c_2$  each receive credit  $1/5$ ,  
 640 tuples  $f_4$  and  $c2f_4$  receive  $1/3$ , while tuple  $c_1$  receives  $8/15$ . Note that if a  
 641 tuple appears in more than one lineage set, then it will accumulate credit  
 642 from the distribution associated with each one of these sets, implying that  
 643 it has a more significant role in the context  $Q$ , as is the case with  $c_1$  in this  
 644 example.

645 Not all of the tuples in the lineage of an output tuple are necessary to be  
 646 present at the same time for the output tuple to appear in the query results.  
 647 For example, if the database only had the set of tuples  $\{f_1, c2f_1, c_1\}$  or the set  
 648  $\{f_1, c2f_2, c_2\}$ , the existence of  $o_1$  would still be guaranteed. In other words,  
 649 while  $f_1$  is always needed for  $o_1$  to appear in the output, only one of the sets  
 650 of tuples  $\{c2f_1, c_1\}$  and  $\{c2f_2, c_2\}$  is required. One could therefore argue that

651 it would be more fair for  $f_1$  to receive more credit than the other four tuples,  
 652 given its role in producing  $o_1$ .

653 This highlights one limitation of the lineage-based DS: while able to find  
 654 all and only the relevant tuples of the output, it does not distinguish the  
 655 *importance* of tuples in the query computations. We therefore present three  
 656 other, more sophisticated, forms of distribution strategies based on why-  
 657 provenance, how-provenance, and responsibility.

### 658 5.3. A Why-Provenance-Based Distribution Strategy

659 The distribution strategy based on why-provenance first equally distributes  
 660 the credit  $k_o$  among the witnesses of the witness basis for  $o$ , and then equally  
 661 divides the credit of a witness among the tuples in the witness. Since a tuple  
 662 may appear in more than one witness, it will receive more than one portion  
 663 of credit from the same distribution. More formally:

664 **Definition 5.3.** *Why-Provenance-based Distribution Strategy*

665 *Let  $I$  be a database instance,  $Q$  a query over  $I$ ,  $o \in Q(I)$  an output tuple  
 666 and  $k_o$  the total credit associated to  $o$ . Let  $\mathcal{W} = \text{Why}(Q, I, o)$  be the witness  
 667 basis of  $o$  according to  $Q$  and  $I$ , and  $W \in \mathcal{W}$  be a witness.*

*Then tuple  $t$  in  $I$  receives credit equal to:*

$$f_{I,Q}(t, k_o) = \frac{k_o}{|\mathcal{W}|} \sum_{W \in \gamma(\mathcal{W}, t)} \frac{1}{|W|}$$

where  $\gamma$  is a function which returns all witnesses  $W$  in which  $t$  appears:

$$\gamma(\mathcal{W}, t) = \{W \in \mathcal{W} : t \in W\}$$

668 Figure 4 shows the distribution of credit with why-provenance-based DS  
 669 for tuple  $o_1$ . The credit is first equally divided between the two witnesses, so  
 670 that both receive credit  $1/2$ . The credit is then further divided among the  
 671 tuples in each witness. Since each witness has three tuples, each tuple in a  
 672 witness receives  $1/6$  of credit. At the end of the distribution,  $f_1$  receives a  
 673 total credit of  $1/3$ , and the other tuples receive  $1/6$  each. This distribution  
 674 better reflects the role of  $f_1$  in the generation of  $o_1$  since, as discussed earlier,  
 675 it is the only mandatory tuple for  $o_1$  to appear in the output; only one of the  
 676 two other pairs of tuples are necessary for  $o_1$  to appear in the result.

677 This example illustrates that why-provenance can better reward input  
 678 tuples depending on their role. Tuples that appear in more than one witness  
 679 are rewarded more than others.



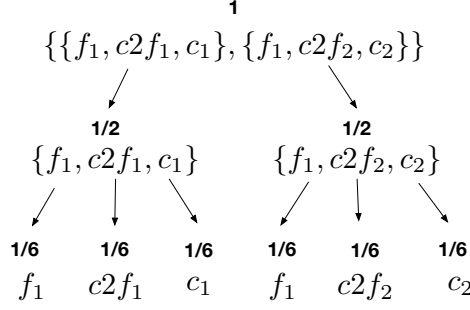


Figure 4: Distribution of credit using why-provenance-based DS for tuple  $o_1$ .

Table 6: **Notations used in Definition 5.4.**

$\mathcal{H}$	provenance polynomial
$M_i$	a monomial in $\mathcal{H}$
$t_j$	a tuple in $M_i$
$c(\mathcal{H})$	sum of $\mathcal{H}$ 's coefficients
$e(M_i)$	sum of $M_i$ 's exponents
$mc(M_i)$	$M_i$ 's coefficient
$te(t_j, M_i)$	exponent of $t_j$ in $M_i$
$\gamma(t_j, \mathcal{H})$	set of monomials in $\mathcal{H}$ containing $t_j$

#### 5.4. A How-Provenance Based Distribution Strategy

The how-provenance-based DS first distributes the credit to the monomials of the polynomial accordingly to the weight represented by their coefficients, then to the tuples of each monomial accordingly to the weights represented by their exponents.

To define the DS more formally, we introduce some notation and illustrate it using the provenance polynomial  $\mathcal{H}$  shown in Figure 5. This notation is also reported for an easy reference in Table 6.

We call  $c$  the function that, given a polynomial, returns the sum of its coefficients; thus  $c(\mathcal{H}) = 3 + 1 = 4$ . We call  $e$  the function that, given a monomial, returns the sum of its exponents, thus  $e(M_2) = 1 + 3 + 3 = 7$ .  $mc$  is the function that takes as input a monomial and returns its coefficient; thus  $mc(M_1) = 3$ .  $te$  is a function that takes as input a tuple and a monomial, and returns the exponent of the tuple in the monomial, if present; thus  $te(c_2, M_2) = 3$ . Finally,  $\gamma$  takes as input a tuple and the whole polynomial, and returns a set of monomials containing that tuple, if present in the

$$\begin{aligned}
\mathcal{H} &= \underbrace{3f_1 \cdot c2f_1 \cdot c_1}_{M_1} + \underbrace{f_1 \cdot c2f_2^3 \cdot c_2^3}_{M_2} \\
c(\mathcal{H}) &= 4 & e(M_2) &= 7 \\
mc(M_1) &= 3 & mc(M_2) &= 1 \\
te(c_2, M_2) &= 3 & \gamma(c_1, \mathcal{H}) &= \{M_1\} \\
\gamma(f_1, \mathcal{H}) &= \{M_1, M_2\}
\end{aligned}$$

Figure 5: Illustration of notation used to define the how-provenance based DS in Definition 5.4.

polynomial; thus  $\gamma(f_1, \mathcal{H}) = \{M_1, M_2\}$ ,  $\gamma(c_2, \mathcal{H}) = \{M_2\}$ .

**Definition 5.4.** *How-Provenance-Based Distribution Strategy*

Let  $I$  be a database instance,  $Q$  a query over  $I$ ,  $o \in Q(I)$  an output tuple,  $\mathcal{H}$  be the provenance polynomial for  $o$ , and  $k_o$  the credit given to  $o$ . The credit given to tuple  $t$  in  $I$  is:

$$f_{I,Q}(t, k_o) = \frac{k_o}{c(\mathcal{H})} \sum_{M \in \gamma(t, \mathcal{H})} mc(M) \frac{te(t, M)}{e(M)}$$

Going back to the example of Table 4, consider  $o_1$  with provenance polynomial  $f_1c2f_1c_1 + f_1c2f_2c_2$ . The how-provenance-based DS firstly divides the credit between the two monomials. Since the coefficients of each monomial are 1, the credit is split in half. If they were, for example, 1 and 2 respectively, 1/3 of the credit would go to the first monomial, and 2/3 to the second. Since in our example each variable has exponent 1, the credit is further divided equally among the three variables. Thus, at the end of the computation,  $f_1$  receives 1/3, and the other tuples receive 1/6.

Consider instead the example where the polynomial is  $f_1^2c2f_1c_1 + f_1^2c2f_2c_2$ ,  $k_o = 1$ , and let us focus on the first monomial. It receives 1/2 of credit, then  $f_1$  receives 1/4 of credit due to its exponent, while the other two tuples receive 1/8.

In this specific example, the how-provenance-based DS has the same outcome as the one based on why-provenance. We therefore consider another query over GtoPdb, Q2, that asks for the families of type **gpcr** that have as contributor a researcher located in the UK:

```

Q2: SELECT DISTINCT F.name
FROM family as F JOIN

```

id	name
$oxs_1$	Dopamine Receptors

lineage	why-provenance	how-provenance
$\{f_1, c2f_1, c_1, c2f_2, c_2\}$	$\{\{f_1, c2f_1, c_1\}, \{f_1, c2f_2, c_2\}\}$	$f_1^2 c2f_1 c_1 + f_1^2 c2f_2 c_2$

Table 7: Result of query Q2 applied on the database of Table 1 and its different provenances. The reported numbers are the credit distributed through the process.

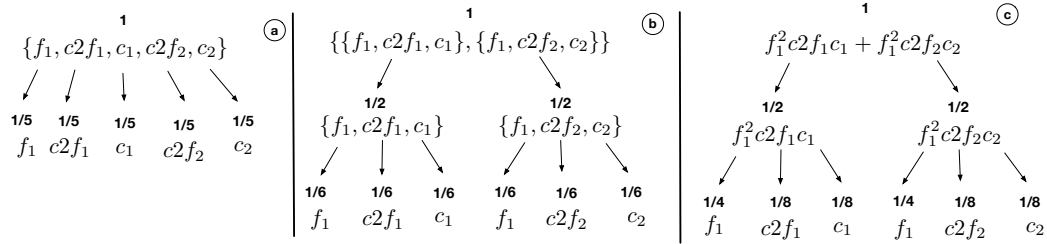


Figure 6: Comparison of different distributions strategies for tuple  $o_1$  produced by query Q2.

```

719 (SELECT DISTINCT f.name AS name
720 FROM family AS f JOIN contributor2family AS c2f ON f.id = c2f.family_id
721 JOIN contributor AS c ON c2f.contributor_id = c.id
722 WHERE c.country = "UK") AS R ON F.name = R.name
723 WHERE F.type = "gpcr"

```

724 The result of Q2 is shown in Table 7, and consists of one tuple, anno-  
725 tated with each of the three provenances. As can be seen, lineage and why-  
726 provenance are identical to those of the tuple  $o_1$  in the previous example.  
727 The how-provenance, however, is different since tuple  $f_1$  is used twice: first  
728 in the join of the inner query, and second in the join of the outer query. This  
729 information is lost in the first two forms of provenances since they are sets,  
730 but it is captured in how-provenance through the use of the operator ‘.’.

731 Figure 6 shows the differences between the three DS for the tuple  $o_1$   
732 of Table 7. Subfigure 7.a uses lineage, sub-figure 7.b uses why-provenance,  
733 and sub-figure 7.c uses how-provenance. The DS based on the provenance  
734 polynomial gives credit 1/2 to  $f_1$ , and 1/8 to the other tuples. This is  
735 reasonable since Q2 relies on  $f_1$  even more than Q1 does. The distribution  
736 based on how-provenance rewards  $f_1$  more, showing that how-provenance is  
737 even more sensitive to the tuples’ role in a query than why-provenance. This  
738 is a direct consequence of the fact that, as proven in [33], how-provenance is

more general than why-provenance and lineage, in the sense that it contains more information.

### 5.5. Responsibility-based Distribution Strategy

As we described in Section 4.3, causality and responsibility are not new forms of data provenance, but rather new information that is added to the already available lineage. Given the lineage of an output tuple  $o$ , it is first possible to compute the type of causality of each of its tuples, distinguishing between counterfactual and actual causes, by testing what happens by removing single tuples and contingency sets of other tuples of the lineage. Successively, it is possible to compute their responsibility through the minimal contingency sets found in this way.

One first option to define a distribution strategy using responsibility is to simply assign the responsibility of a tuple as its credit. In this way, responsibility is both a way to compute credit and to distribute it. Using the example of Table 5, in the case of output tuple  $o_1$ ,  $f_1$  receives credit 1, the other tuples credit 0.5.

However, we want a DS that is also a function of the input credit value  $k$  in order to be comparable with the other three strategies proposed so far. We define a new DS based on responsibility that is a function of the quantity of credit  $k_o$  that assigns to each tuple of the lineage a portion of this credit weighted by its normalized quantity of responsibility. This function will give a bigger portion of credit to tuples that are higher in the responsibility ranking. Formally:

**Definition 5.5.** *Responsibility-based Distribution Strategy*

Let  $I$  be a database instance,  $Q$  a query over  $I$ ,  $o \in Q(I)$  an output tuple,  $L$  the lineage of  $o$ , and  $k_o$  the credit given to  $o$ . The credit given to tuple  $t$  in  $I$  is:

$$f_{I,Q}(t, k_o) = k_o \frac{\rho_t}{\sum_{t' \in L} \rho_{t'}}$$

where  $\rho_j$  is the responsibility of tuple  $j$  as in Definition 4.2.

Note that only the tuples that belong to the lineage will receive a quantity of credit  $> 0$ . The more important the tuple, i.e., the higher its responsibility, the bigger the quantity of credit received.

Figure 7 shows the responsibility and the credit assigned to the tuples of the lineage of the output tuple  $o_1$  of Table 5. Assuming that  $k_{o_1} = 1$ ,  $f_1$

	counterfactual cause		actual causes		
$k_{o_1} = 1$	$f_1$	$c_2 f_1$	$c_2 f_2$	$c_1$	$c_2$
	↓	↓	↓	↓	↓
responsibility	1	0.5	0.5	0.5	0.5
responsibility-based DS	1/3	1/6	1/6	1/6	1/6

Figure 7: Example of distribution of credit using responsibility and normalized responsibility and the responsibility-based DS, assuming  $k_o = 1$ .

receives credit 1/3, while the others receive credit 1/6. As we see, the DS in this case returns the same distribution obtained with why-provenance that was shown in Figure 6. This is not always the case though, as we show in the example of Section 6.2.

## 6. Experimental Evaluation

To understand the trade-offs between these Distribution Strategies (DSs), we perform four sets of experiments using queries over target families presented on the GtoPdb website. The first set of experiments use real queries extracted from citations to GtoPdb published in the British Journal of Pharmacology. The second set uses synthetically produced provenance polynomials, corresponding to more complex queries, in order to better highlight the differences between the DSs. The third set of experiments considers the accrual of credit over time by the three strategies, again using synthetic queries. The fourth set of experiments shows how the DSs compare to traditional citations in giving credit to data curators using both real and synthetic queries.

All experiments were carried out on a MacBook Pro with a 2.4 GHz processor Intel Core i5 quad-core and 8 GB of memory at 2133 MHz. Code was written in Java, supported by a PostgreSQL database.<sup>11</sup>

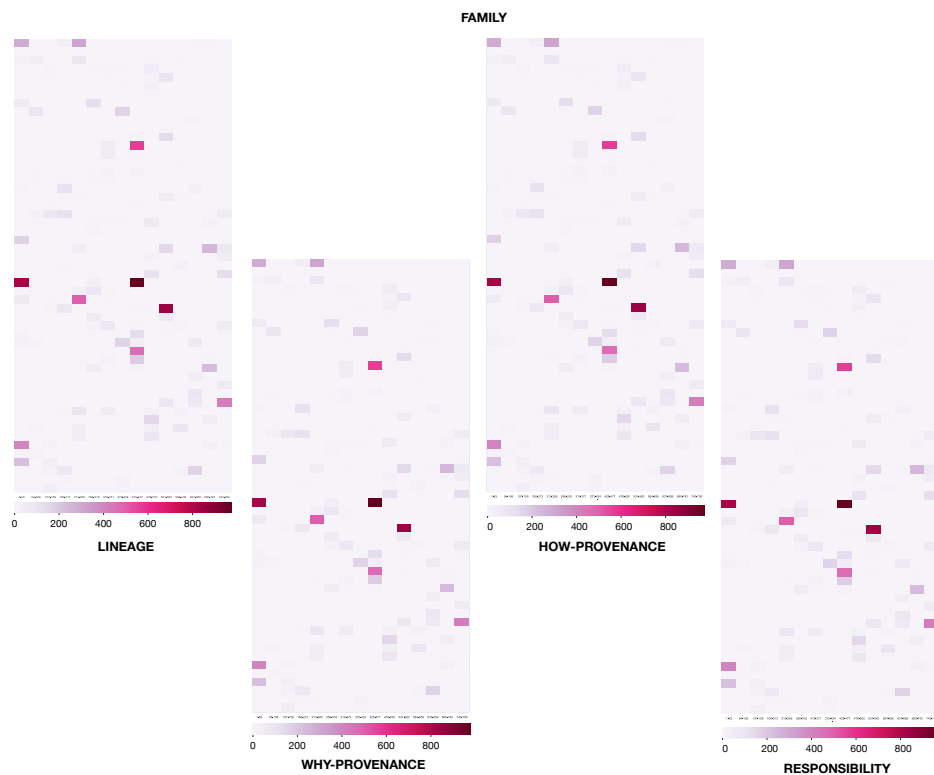


Figure 8: Comparison of four DS on the same table **family** using the distribution given by the queries retrieved from papers. Each cell is a tuple.

### 6.1. Real-world queries

Examples of real queries are drawn from papers published in the British Journal of Pharmacology (BJP) <sup>12</sup>. Each time a paper in this journal cites a webpage from GtoPdb, it reports the URL of the page. From this URL, the query used to obtain the webpage data can be determined. We considered all 889 papers in BJCP citing the IUPHAR/BPS Guide to pharmacology [35] as of October 2020, and extracted all webpage URLs to GtoPdb contained

<sup>11</sup>For purposes of reproducibility, the code we used for our experiments and all queries are available here: [https://bitbucket.org/dennis\\_dosso/credit\\_distribution\\_project](https://bitbucket.org/dennis_dosso/credit_distribution_project).

<sup>12</sup><https://bpspubs.onlinelibrary.wiley.com>

800 within the paper.<sup>13</sup>

801 The queries that we inferred are those used to build target family web-  
802 pages within GtoPdb. An example was given in Figure 3, where we show  
803 how the structure of the “Adenosine receptors” family can be mapped into  
804 queries over the underlying database. In GtoPdb, all target family pages  
805 share a similar structure; the only difference is that individual sections, such  
806 as “contributors” or “further readings”, may be absent. Therefore, the same  
807 queries can be used to build all of the target family pages by changing the  
808 family id used in the query (for example, in Figure 3, it is 3). Note that  
809 the queries are fairly simple SQL queries, and fall into a class called “select-  
810 project-join” or “SPJ” queries. A total of more than 12K different queries  
811 were built in this way. Without loss of generality, we give each tuple in the  
812 output of a query a credit of 1.

813 *Results.* Figure 8 shows the heat-maps obtained by the distribution of credit  
814 according to the **four** different DS on one of the tables in the underlying  
815 database, **family**, which is often joined with other tables in the database to  
816 build the webpages. Each cell in a heat-map represents a tuple of the **family**  
817 table and the color indicates the amount of credit attributed to such tuple.  
818 It can be seen that the result of credit distribution over **family** is the same  
819 for all **four** strategies. The same result is also obtained with the other tables  
820 of the database used by the queries shown in Figure 3.

821 The reason why credit distribution is the same for all **four** strategies is  
822 that the queries are all simple SPJ queries, which use each table only once and  
823 do joins on key attributes. Under these conditions, each tuple of the output  
824 presents: (i) a how-provenance that is a single monomial with coefficient 1  
825 and exponent 1 in each variable; (ii) a why-provenance with only one witness;  
826 (iii) a lineage that coincides with the witness in the basis, and (iv) all tuples  
827 are counterfactual causes. Hence, for these queries, the **four** DSs behave in  
828 the same way: credit is uniformly distributed among the tuples present in  
829 each provenance.

830 To illustrate this, consider one of the queries in Figure 3 which is used to  
831 build the output webpage:

---

<sup>13</sup>The IUPHAR/BPS Guide is a journal that describes the structure and evolution of GtoPdb. At the time of writing, it had received more than 1200 citations on Google Scholar.

```

832     Q3: SELECT c.first_names, c.surname
833     FROM contributor2family AS cf JOIN contributor AS c ON
834     cf.contributor_id = c.contributor_id
835     WHERE f.family_id = 3

```

836 Q3 returned 10 tuples from the version of GtoPdb used. The first tu-  
837 ple, <Bertil B., Fredholm>, has  $c_{939} \cdot c_{2f_{496}}$  as its provenance polynomial.  
838  $c_{939}$  represents the provenance token of a tuple in **contributor**, and  $c_{2f_{496}}$   
839 the provenance token of a tuple in table **contributor2family**. The why-  
840 provenance of this tuple is  $\{\{c_{939}, c_{2f_{496}}\}\}$ , its lineage is  $\{c_{939}, c_{2f_{496}}\}$ , **both**  
841 **these tuples are counterfactual causes and have responsibility 1**. Therefore,  
842 the credit assigned to these tuples is 1/2 using all four DS. This happens  
843 for all the tuples in the output of each query of GtoPdb, thus making the  
844 distributions equivalent over all outputs.

845 However, this is not the case with more complex queries. As we showed  
846 in the previous section, when two or more tuples are merged as a result of a  
847 projection or union, the credit distributions will differ between the strategies.

## 848 6.2. Synthetic queries

849 To simulate synthetic queries, we randomly generated provenance poly-  
850 nomials in which the coefficients and exponents could be greater than 1.  
851 The queries involve three GtoPdb tables: **family**, **contributor2family**,  
852 and **contributor**. The polynomials were generated as follows (every time  
853 we use the word “randomly”, we mean that we used a uniform distribution):  
854 first, the number of monomials composing the polynomial is decided choos-  
855 ing randomly a number between 1 and 6. Then, we randomly choose a tuple  
856 from the tables **family**, one from the table **contributor2family** and one  
857 from table **contributor** that are used as the monomial’s variables. We then  
858 choose a coefficient for this monomial (between 1 and 3) and an exponent for  
859 each tuple (between 1 and 4). For the next monomial, then, we decide if we  
860 want to keep the same tuple from the table **family** as first tuple of the new  
861 monomial. To do so, we generate a random number between 0 and 1. If the  
862 number is above 0.2, we change the family tuple.

863 An example can be found in Figure 9, which shows a sample synthetic  
864 provenance polynomial (the how-provenance), the corresponding why-provenance,  
865 lineage, and the causality of the tuples of the lineage, together with their re-  
866 sponsibility. The resulting credit distribution for each DS is also shown.

867 As an example of how the distribution strategies behave with these syn-  
868 thetic queries, consider tuple  $f_5$  in Figure 9. **This tuple receives the high-**



**How-provenance:**  $3f_1^3c_2f_1^2c_1^2 + 2f_1c_2f_2^3c_2^3 + 4f_5c_2f_{17}^4c_{18}^3$

**Credit distribution:**

$$f_1 = \frac{59}{315}, f_5 = \frac{1}{18}, c_2f_1 = \frac{2}{21}, c_2f_2 = \frac{2}{15}, c_2f_{17} = \frac{2}{9}, c_1 = \frac{2}{21}, c_2 = \frac{2}{15}, c_{18} = \frac{1}{6}$$

**Why-provenance:**  $\{\{f_1, c_2f_1, c_1\}, \{f_1, c_2f_2, c_2\}, \{f_5, c_2f_{17}, c_{18}\}\}$

**Credit distribution:**

$$f_1 = \frac{2}{9}, f_5 = \frac{1}{9}, c_2f_1 = \frac{1}{9}, c_2f_2 = \frac{1}{9}, c_2f_{17} = \frac{1}{9}, c_1 = \frac{1}{9}, c_2 = \frac{1}{9}, c_{18} = \frac{1}{9}$$

**Lineage:**  $\{f_1, f_5, c_2f_1, c_1, c_2f_1, c_2f_2, c_2f_{17}, c_1, c_2, c_{18}\}$

**Credit distribution:**

$$f_1 = \frac{1}{8}, f_5 = \frac{1}{8}, c_2f_1 = \frac{1}{8}, c_2f_2 = \frac{1}{8}, c_2f_{17} = \frac{1}{8}, c_1 = \frac{1}{8}, c_2 = \frac{1}{8}, c_{18} = \frac{1}{8}$$

**Causality:** counterfactual causes:  $\emptyset$ ,

actual causes:  $\{f_1, f_5, c_2f_1, c_1, c_2f_1, c_2f_2, c_2f_{17}, c_1, c_2, c_{18}\}$

**Responsibility:**

$$f_1 = \frac{1}{2}, f_5 = \frac{1}{2}, c_2f_1 = \frac{1}{3}, c_2f_2 = \frac{1}{3}, c_2f_{17} = \frac{1}{2}, c_1 = \frac{1}{3}, c_2 = \frac{1}{3}, c_{18} = \frac{1}{2}$$

**Credit distribution:**

$$f_1 = \frac{3}{20}, f_5 = \frac{3}{20}, c_2f_1 = \frac{1}{10}, c_2f_2 = \frac{1}{10}, c_2f_{17} = \frac{3}{20}, c_1 = \frac{1}{10}, c_2 = \frac{1}{10}, c_{18} = \frac{3}{20}$$

Figure 9: Sample synthetic provenance polynomial (how-provenance) and corresponding why-provenance, lineage, causality and responsibility values, together with the corresponding credit distributions.

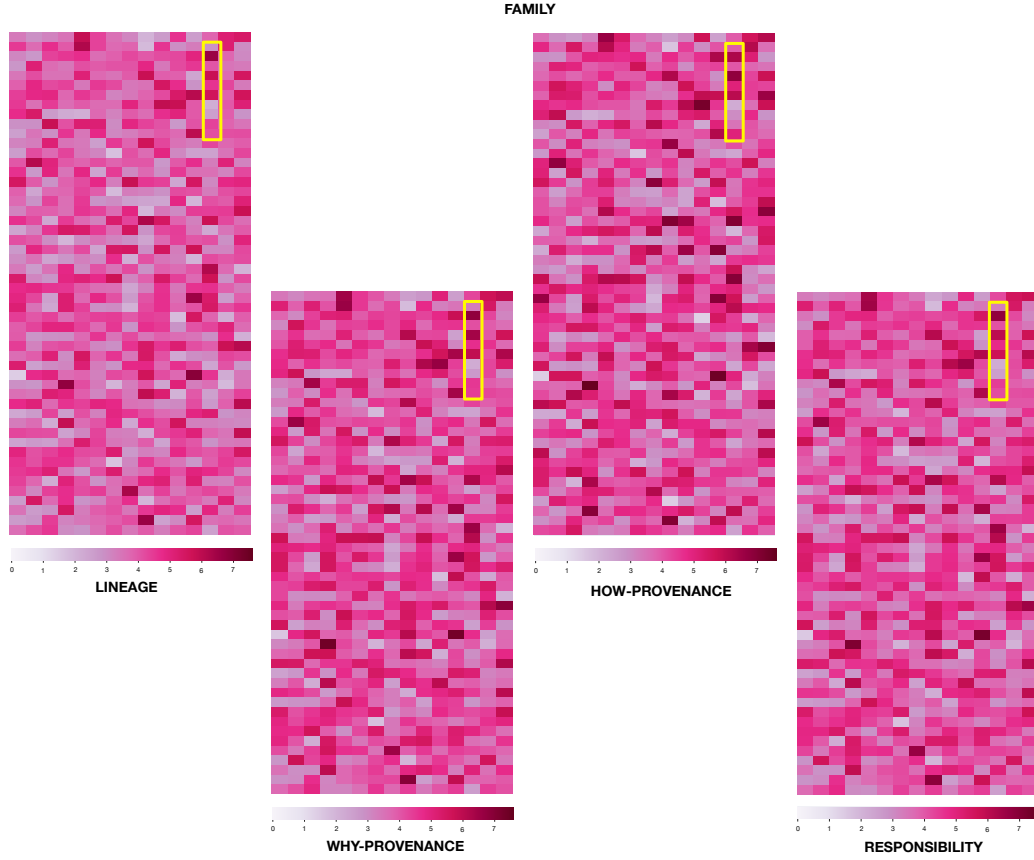


Figure 10: Comparison of three DS on the same table `family` after the distribution computed using 10K synthetic and randomly generated provenance polynomials. The tuples in the blue rectangles are used as example in the discussion connected to Figure 11.

est quantity of credit using responsibility-based distribution, and less credit  
 using, in order, lineage, why- and how-provenance. This is because more  
 information is available about the role of the tuple in the overall compu-  
 tation. Generally speaking, the more complex the distribution (the most  
 complex being how-provenance), the more credit is given to tuples which are  
 more frequently used, and thus have a higher impact in producing the out-  
 put tuple. Responsibility, on its part, can be seen as an enrichment of the  
 information brought by lineage. It creates a ranking among lineage’s tuples  
 describing the importance of their role in generating the output. As such,  
 the responsibility-based DS gives more credit to  $f_1, f_5, c2f_{17}$  and  $c_{18}$  due to  
 their higher responsibility values. This notion of “importance” is connected

880 to their corresponding minimal contingency sets. For example,  $f_1$  has as  
 881 minimal contingency set (one of the many)  $\{f_5\}$ , with cardinality 1. On the  
 882 other hand,  $c_1$  has, as minimal contingency set (one of the many)  $\{f_5, c_2\}$ ,  
 883 with cardinality 2. This means that  $c_1$  is “less important” of the tuples with  
 884 minimal contingency sets of lower cardinality, and this is reflected on the  
 885 different quantity of credit being distributed.

886 Despite being synthetic, these provenance polynomials may actually be  
 887 obtained from queries. The polynomials can be obtained by any nested query  
 888 with join and union operations that use the same tuple multiple times (in  
 889 which case the exponents are bigger than 1), and the same combination of  
 890 operations more than once (in which case the coefficients of monomials are  
 891 bigger than 1).

892 *Results.* The results of credit distribution on the **family** table using 10K  
 893 randomly generated synthetic provenance polynomials are shown in Figure  
 894 10. We set the maximum value in the heat maps to the highest value reached  
 895 by a tuple in all three distributions (i.e., 7.5).

896 As can be seen, the four strategies generate different credit distributions,  
 897 indicated by the varying hues. However, there is a certain amount of consis-  
 898 tency between them in that tuples which are highly rewarded by one strategy  
 899 are also highly rewarded by the others. This shows that the four DSs consis-  
 900 tently reward certain tuples more than others.

901 Note that lineage-based DS gives the least credit to tuples in the **family**  
 902 table, indicated by an overall lighter hue. This is because the DS distributes  
 903 credit equally to all tuples appearing in the lineage. Since these queries also  
 904 use two other tables, credit is distributed to tuples in those tables.

905 Moving to why-provenance-based DS, we see that more credit is given to  
 906 tuples in the **family** table than with the previous strategy. This is because  
 907 the DS considers the different ways that a tuple is used, e.g. in joins with  
 908 other tuples. If the same tuple is present in more than one witness, it will  
 909 draw more credit and take it from other tuples in the witness basis. In  
 910 this case, tuples in **family** drew more credit, taking it from tuples in the  
 911 other two tables, due to the role that **family** tuples played in the queries  
 912 that were executed. We also notice that the responsibility-based distribution  
 913 strategy has a distribution that is quite similar to the one provided by why-  
 914 provenance. It is often the case, for example when the witnesses of the  
 915 why provenance share one common tuple, that the two distributions behave  
 916 similarly. As a consequence, the synthetic polynomials are at times such that

917 the two distributions behave in the same way, or very similarly.

918 We note that the lineage-based DS gives an average credit of 3.92 to each  
919 tuple in the table, while the DS based on why-provenance assigns 4.18, how-  
920 provenance 4.18, and the one based on responsibility 4.13. Moreover, lineage  
921 distributed a total of about 3121 units of credit to the **family** table, why-  
922 provenance 3333, how-provenance 3331, while responsibility assigned 3290.

923 Finally, consider the how-provenance-based DS heat-map. As with why-  
924 provenance, more credit is typically given to tuples in **family** compared to  
925 lineage-based DS, since it recognizes the role of these tuples in the queries,  
926 and the overall hue is deeper. The two distributions appear similar, although  
927 on closer inspection, slight differences can be seen. This is because how-  
928 provenance also considers the frequency with which tuples are used, not only  
929 the ways in which they are used. Therefore, although the overall distribution  
930 is similar, there are small differences due to the presence of exponents and  
931 coefficients in the provenance polynomials, influencing the distribution of  
932 credit.

933 To better understand this difference, in the next subsection we consider  
934 the accrual of credit over time. In doing so, we will focus on the ten tuples  
935 shown within the large yellow rectangles in Figure 11. Each small rectangle  
936 within a large blue rectangle is a tuple, and we number them from 1 (top) to  
937 10 (bottom). These ten tuples were selected specifically because they allow  
938 us to see the evolution of the distribution of credit through time.

### 939 6.3. Credit accrual over time

940 Since credit accrues over time, we simulate the passage of time by varying  
941 the number of queries executed, and look at the “snapshots” of credit for each  
942 of the strategies using synthetic queries. The results are shown in Figure 11.

943 In this figure, four groups of heat-maps are shown. Each group represents  
944 a “snapshot” taken after 1K, 2K, 5K and 10K provenance polynomials have  
945 been considered for credit distribution. The ten tuples in each heat-map are  
946 those highlighted in the yellow boxes of Figure 10 from the **family** table.

947 The polynomials used are the same as the experiment of the previous  
948 section. The range of credit in each map goes from 0 (no credit) to 7 (the  
949 maximum quantity of credit reached – using how-provenance – on one of the  
950 tuples of the considered window at the “snapshot” with 10K queries). The  
951 color hue of the legend, as can be seen, still ranges from 0 to 7.5.

952 By the end of 1K queries, credit differentials between tuples as well as  
953 between strategies can be seen. For example, tuple 3 is usually rewarded the

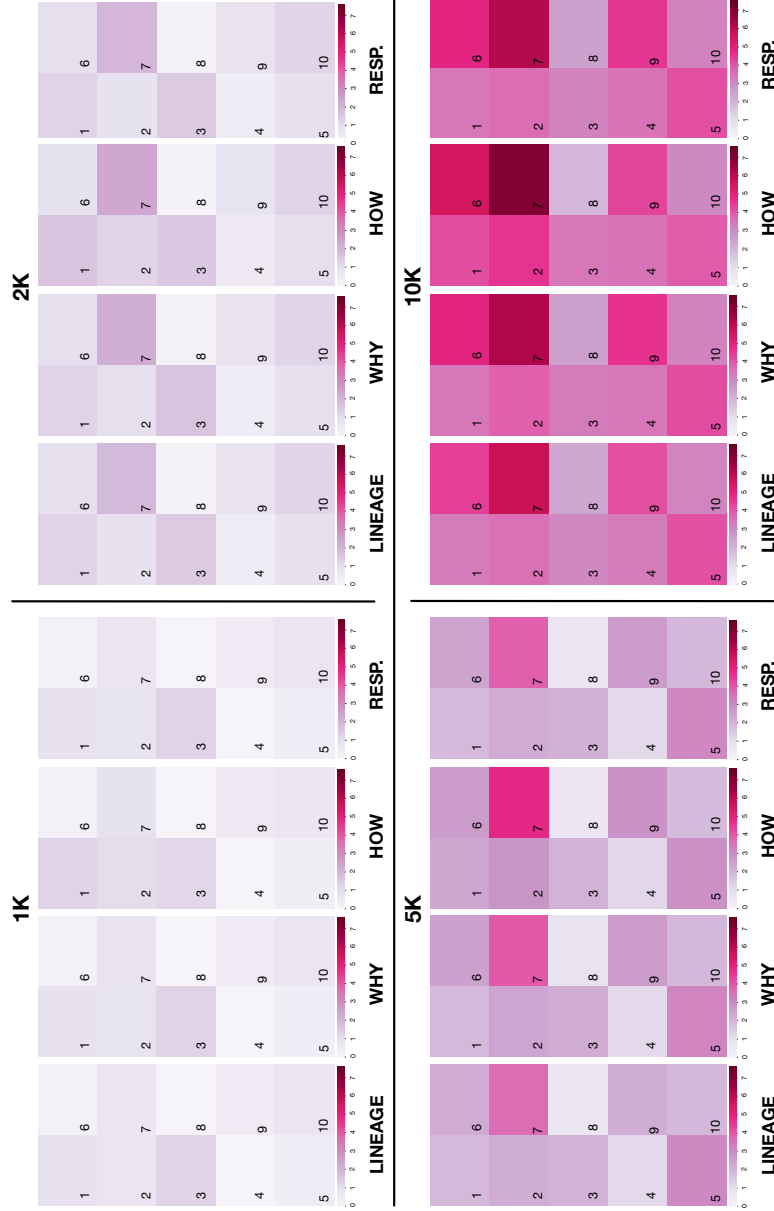


Figure 11: Comparison of the distribution of credit performed by the four DSs on a subset of 10 tuples taken from the `family` table, simulating the passing of time. The number at the top of each group of heat-maps represents the number of polynomials whose credit has been distributed.

954 most credit by all four strategies. However, it receives the highest quantity  
 955 of credit from the why-provenance- and responsibility-based strategy (1.33).  
 956 Moreover, it can be seen that tuples 1 and 7 receive a higher quantity of credit  
 957 when how-provenance is exploited with respect to the overs, showing how this  
 958 provenance behaves differently from the others in this context. Moving to  
 959 2K queries, it is possible to see that tuple 3 and 7 are still the most rewarded  
 960 by the strategies. This trend continues to the end of 2k queries.

961 By the end of 5k queries, tuple 7 emerges with the highest value of credit  
 962 with all four DSs, a position which is strengthened by the end of 10k queries.  
 963 Moreover, with the passing of time, tuple 3 ceases to be one of the most  
 964 rewarded ones and new tuples, such as 6 and 9, emerge as being particularly  
 965 rewarded at 5K, while at 10K tuples 6 and 7 are the most rewarded from  
 966 the distributions. This is because tuple 7 is used several times within queries  
 967 being executed, which is rewarded strongly by why- and how-provenance.  
 968 We also note that the responsibility-based distribution confirms its trend of  
 969 being similar to why-provenance, although not completely identical. This  
 970 is more evident at step 10K, where tuple 7 is slightly less rewarded using  
 971 responsibility (6.12) with respect to why-provenance (6.24). This is due to  
 972 the fact that tuple 7 had, among some of the polynomials being used for the  
 973 experiments, a high responsibility but it did not appear in all witnesses. This  
 974 changed slightly the distribution.

975 While the relative value of credit “positions” of tuples within a DS strat-  
 976 egy depends on what queries are being executed, the important thing to  
 977 notice is the difference between the DSs over time: overall, lineage gives less  
 978 credit to tuples in the `family` table than the other two strategies since credit  
 979 is shared with tuples in other tables. However, the why-, `responsibility`- and  
 980 how-provenance-based strategies recognize the more important role being  
 981 played by the `family` tuples than those in the other tables. The differences  
 982 between why- and responsibility-based DS are, for the most times, negligible.  
 983 The differences between the why- and how-provenance-based DSs are also rel-  
 984 atively minor in most cases. However, there are certain situations in which  
 985 the role of a tuple is particularly critical in a query, and in this case the dif-  
 986 ference in the value of credit assigned is notably higher for how-provenance,  
 987 as we saw with tuple 7 in the example of Figure 11.

988 To sum up, the DS based on lineage is sufficient to highlight which tuples  
 989 in the database are used by a query, and distributes credit equally to these  
 990 tuples. The resulting distribution rewards tuples that are used by more  
 991 queries, but does not reward how many times tuples are used in the same

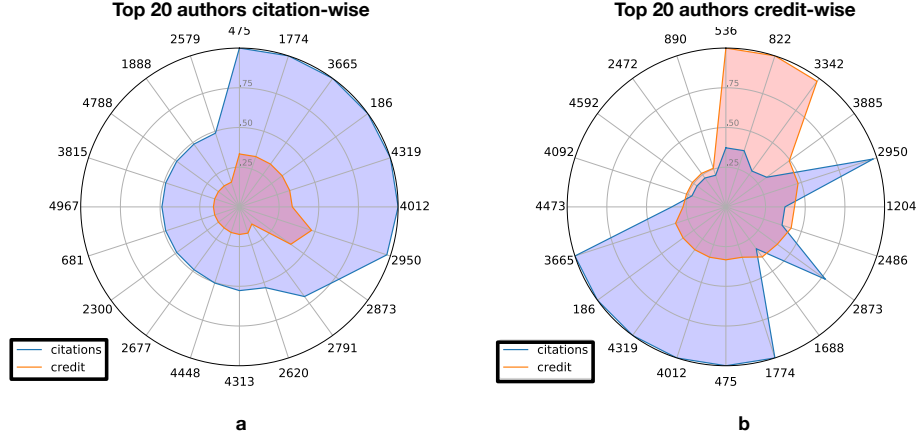


Figure 12: Radars presenting the top 20 authors citation-wise and credit wise, together with their (normalized between 0 and 1) values of citations and credit.

query. However, a DS based on why-, responsibility- or how-provenance may be better if the queries are complex, since they reward more tuples that have a critical role in generating the output. In particular, these three DSs may be useful for finding “hotspots” in the database based on the role of tuples, with the how-provenance-based DS being preferable if a higher sensitivity to the role of a tuple in queries is required.

#### 6.4. Credit vs Citations

In the last set of experiments, we compare traditional citations to the proposed credit distribution strategies to see the difference in reward for data authors and curators. Using both real-world and synthetic queries, we distribute credit to the authors responsible for the data under the different strategies. Our results show that credit rewards authors of data that is cited fewer times, but that has a higher impact on the query results.

To do so, we need to identify a set of authors and queries that cite data curated by them. Considering GtoPdb, each target family page has a list of curators, representing the people who are co-creators and curators of the data comprising the page. This list can be obtained using the last query shown in Figure 3. Each time a target family page is cited, we assign one *citation* to each author associated with the page. The authors also receive *credit* in the amount assigned to the data used by the query to construct the webpage, equally divided between the authors of the webpage.

1013 *Results: Real-world queries.* As described in Section 6.1, we consider real-  
1014 world queries taken from papers published in the BJP which reference web-  
1015 pages in GtoPdb. Since for these queries there is no difference in the distri-  
1016 bution of credit between the DSs, only one value for credit is used.

1017 The results are shown in the radar plots of Figure 12, in which each  
1018 number on the outer circle (e.g. 475, 1774 and 3665) represents an author  
1019 (id) and the blue (red) line represents the normalized value of credit generated  
1020 by citations (credit), respectively. The first radar plot, Figure 12.a, shows the  
1021 top 20 authors in terms of *citations*, ordered in a clockwise direction, whereas  
1022 Figure 12.b orders the authors based on *credit*. Comparing the author ids  
1023 used in the outer circles of these two plots, it can immediately be seen that  
1024 the “top authors” are very different using these two metrics, although there  
1025 is some overlap (for example, authors 1774, 475, and 4012).

1026 Diving a bit deeper to focus on the red and blue areas in each of the plots  
1027 reveals that there is a significance difference between citations and credit:  
1028 The top 20 authors in terms of citations do not have the highest values  
1029 of credit (Figure 12.a). Conversely, the authors with the highest values of  
1030 credit do not necessarily have a large number of citations (Figure 12.b). For  
1031 example, author 536 has the highest value of credit, but is not even in the  
1032 top 20 authors in terms of citations. This means that authors like 536, 822,  
1033 and 3342 in Figure 12.b receive much more credit from their relatively few  
1034 citations than authors like 475, who receives the largest number of citations.  
1035 That is, the data underlying certain webpages is more “valuable” in terms  
1036 of credit than a citation to the webpage.

1037 The reason for the difference between citation and credit is partly due to  
1038 the experimental setup: each output tuple carries a credit of 1, and there can  
1039 be many tuples used to generate a webpage. Thus a webpage that is created  
1040 from more tuples will have a higher credit value than one created from fewer  
1041 tuples. Furthermore, authors who collaborated with fewer people will receive  
1042 a biggest share of the equally divided credit. However, all authors will receive  
1043 a citation of one.

1044 Credit distribution therefore rewards authors differently than traditional  
1045 citations: an author who has curated larger quantities of cited data and  
1046 collaborated with fewer co-authors, will receive larger quantities of credit.  
1047 Thus, credit rewards them for their larger contribution to the database.

1048 *Results: Synthetic queries.* We used the same synthetic polynomials de-  
1049 scribed in Section 6.2, and we distributed credit with the first 100,  $1K$ , and



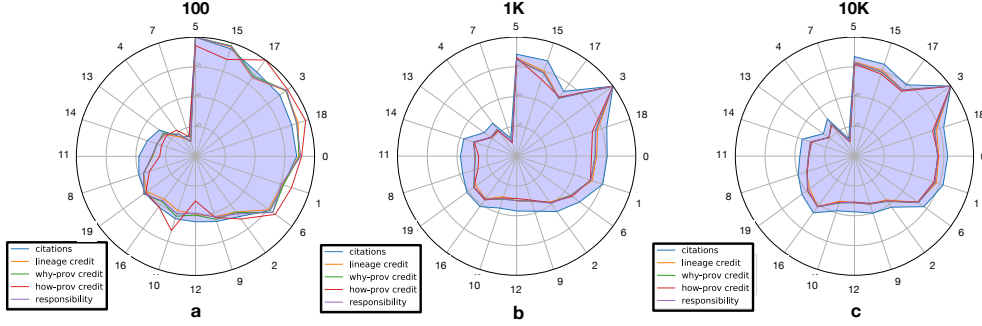


Figure 13: Radars presenting the 20 synthetic authors with corresponding citation and quantities of credit distributed through the 4 DSs (all values normalized between 0 and 1) through different numbers of polynomials (respectively, 100, 1K and 10K). The order is the one defined by figure a, i.e. descending order of citations obtained from 100 polynomials.

10K of them. Since these polynomials are created by randomly selecting tuples from three tables, they usually correspond to a set of data curated by authors who, in reality, did not collaborate. To make the size of the author set more realistic, we therefore created 20 synthetic authors, and randomly assigned one author to blocks of consecutive tuples in the database, with the size of each block varying between 10 and 40, to simulate different quantities of work performed by an author. Every time an author appears as curator of one or more tuples used in a polynomial, we assign them one citation. They also receive four kinds of credit, each one using a different DS.

Figure 13 shows three radar plots, one for the results obtained with 100 polynomials, one with 1K polynomials, one with 10K polynomials. Each plot shows the top 20 authors in terms of citations (hence the authors and clockwise ordering is the same in each of the plots), and additionally shows the the normalized values of citation (blue line), lineage-based credit (yellow line), why-provenance-based credit (green line), how-provenance-based credit (red line), and responsibility-based credit (violet line). As can be seen, given the synthetic nature of the queries, the correlation between the number of citations and the quantity of credit assigned to the authors appears to be a much stronger than with the real-world queries of Figure 12. In fact, for Figure 13.a the linear correlation between the citation number and all four types of credit is always above 0.94 with p values in the order of  $3e-8$ . The credit distributed via lineage is closest to the number of citations (a linear correlation of 0.99, p value of  $2e-16$  in Figure 13.a), while the other three

types of credit behave slightly differently (a linear correlation of around 0.95 in all other three cases in Figure 13.a). Similar observations can be made for Figure 13.b and 13.c.

What these figures show is that, in certain cases, authors who do not have a large number of citations receive more credit than others, as for example authors 17 and 10 in Figure 13.a, and especially when credit is distributed using how-provenance. This again shows how credit gives a different perspective on the role of data and authors by going beyond the limitations of traditional citations.

It is worth noting that, when scaling up to  $1K$  and  $10K$  polynomials, the credit distributions become almost identical (the linear correlation for the values of Figure 13.c is more than 0.99 with a p-value of  $1.32e-32$ ). This is consistent with what we observed in Figure 10.

## 7. Discussion

*Credit Generation.* In this paper we focused on Credit Distribution, the problem of distributing credit generated by a citation to the parts of the database being used by the query subsumed by that citation. A different problem is credit generation, the task of generating credit *before* its distribution. Credit generation presents, in itself, a series of new problems. Among them, we count:

1. *The correct generation of credit* Different types of citations may generate different quantities of credit. Data being cited in the related work of a paper may generate less credit than a result set of data that are extensively used throughout the paper. Different techniques may be employed to compute credit in a way that reflects the impact of the data being cited. Examples may be the manual annotation by the authors of the data that are more relevant in their own assessment to the economy of the paper, or computations performed through NLP techniques to infer the importance of a citation based on the context of the text where it is cited.
2. *Credit produced by self-citations* Data credit, being built on top of traditional citations, inherits some of its problems. Authors, using self-citations, may generate and distribute credit to themselves, making

1107 their work appear much more impactful than it really is in reality. Dif-  
1108 ferent strategies may be exploited in this scenario, ranging from ignor-  
1109 ing completely the credit generated from self-citations to applying a  
1110 discount factor to control it.

1111 3. *Generic citations* As we mentioned, citations may go to the whole  
1112 database, or to views of the database computed using a big portion  
1113 of its data. In this case, credit may be assigned indiscriminately to  
1114 large portions of data, losing the ability to accurately identify parts of  
1115 the database that have high impact. In this case it is also possible to  
1116 ignore queries that are too “general” and considering only queries that  
1117 are discriminative enough.

1118 4. *Different types of credit* In the real world, there are different types of  
1119 research communities interested in information in a database. Doctors’  
1120 interests and queries may differ from the interests and queries of oph-  
1121 thalmologists or pharmacists. For this reason, only distributing one  
1122 generic credit generated from all possible queries coming from this het-  
1123 erogeneous set of users may simply highlight data that are important in  
1124 general, without taking into consideration the specific their specific and  
1125 different needs. One possibility is to keep separated the credit gener-  
1126 ated by different types of users, e.g., have one type of credit generated  
1127 from queries coming from doctors, another type of credit generated  
1128 from queries submitted by ophthalmologists, etc. In this way, it will be  
1129 possible to accurately tailor the process of credit distribution around  
1130 the information need of different categories of users.

1131

1132 *Credit Generation vs Credit Distribution.* We note that, in our experiments,  
1133 we always assumed that the credit carried by an output tuple is 1. Thus, each  
1134 tuple in the output has equal importance. This in general may not be true,  
1135 since different tuples in the output may have different weight, depending on  
1136 the context of the citation. For example, data that is fundamental for the  
1137 results of a paper may have more credit than data being cited as a reference.  
1138 *Credit generation*, i.e. the process by which the credit of the output tuples  
1139 is decided, is a research problem with its own dignity and complexities, and  
1140 we did not face it in this paper.

1141 From the point of view of the model, even when the credit of the output  
1142 tuples is different than 1, nothing needs to change in the models presented

here, since they were defined for a generic value  $k$ . We note that, if the quantity of credit carried by an output tuple changes, as a consequence the final distribution will change too, since certain tuples will be more “impactful” (i.e., distribute more credit) than others.

*On the choice of the DS.* Depending on the type of task at hand, one DS may be preferred on the other. When the user only wants to highlight the tuples being used in the database by a workload, the lineage-based DS is sufficient. When the user also wants to know the relative impact of tuples in the context of the query, the other DSs may be used since they allow to better understand the actual importance of data.

In the real-world example that we used showed that the four DSs behave the same, this was due to the specific nature of the data and the queries being used. However, the why-provenance of a query for example differs from the lineage of the same query whenever the output tuples can be computed in more than one way by the query, i.e., if there is more than one witness. While at the best of our knowledge there isn’t any work that explores SQL query logs to validate the presence of this diversity, we still think that it is safe to assume that in reality there actually are user-submitted queries that present different distribution of credit. To support this opinion, the work by Bonifati et al. [9] showed that in the context of SPARQL query logs submitted to various databases such as DBpedia and Wikidata, more than 90% of these queries are of type select, and more than 30% perform join operations through the `and` operator. These queries contain triple patterns with cardinalities that range from 1 to 11 triples, thus showing their big complexity in certain cases. These queries, that many times are converted in their SQL versions, are composed by join operations that may result in why-provenances with cardinality bigger than 1.

Other works, such as [56], showed that operations such as inner joins can be found in at least 4.5% of queries in the considered workload, with a maximum number of times that operator is used in the same query equal to 164. Outer joins were found in 1% of the queries, and used up to 247 times in the same query. This is another evidence of the potentiality of the fact the why-provenances may become quite complex.

From a complexity standpoint, all four DS present a similar complexity since we focused on SPJ queries. Although responsibility has been found to be hard to compute in certain cases, those are not the situations of the queries

1180 considered in this paper. Speaking in terms of complexity of implementation,  
1181 lineage can be thought as the easiest form of provenance, since it only cares  
1182 about a tuple being used, while the other provenances also need additional  
1183 information to be taken into consideration.

## 1184 8. Conclusions and Future Work

1185 This paper defines three new distribution strategies based on why-provenance,  
1186 how-provenance, and responsibility, and it compares them against the lineage-  
1187 based distribution strategy defined in [26]. The first, why-provenance-based  
1188 DS, uses the concept of a witness, and gives more credit to tuples that ap-  
1189 pear in more than one witness. In this way, tuples that are more important  
1190 to the query and are used in different ways are rewarded more. The sec-  
1191 ond, how-provenance-based DS, considers the frequency with which a tuple  
1192 or combination of tuples is used in the query through the information con-  
1193 tained in a provenance polynomial. In this case, the how-provenance-based  
1194 DS is more sensitive than the why-provenance-based DS to the role and im-  
1195 portance of tuples. The third one exploits the notion of responsibility, a  
1196 real value ranking the tuples of the lineage based on their degree of causal-  
1197 ity in generating the output. The responsibility-based DS showed to behave  
1198 similarly to the why-provenance based one.

1199 To show the differences between the four DSs, we performed extensive  
1200 experiments based on GtoPdb, a curated scientific relational database, using  
1201 both real and synthetic queries. In the first set of experiments, we used select-  
1202 project-join (SPJ) queries extracted from citations to webpages in GtoPdb  
1203 found in papers published in the British Journal of Pharmacology. Using  
1204 these “real” queries, we distributed credit to tuples in different tables of the  
1205 database, highlighting tuples that were more frequently used. We showed  
1206 that, with these queries, the four strategies produce the same distribution.  
1207 This is because the SPJ queries were fairly simple, and did not use self-joins.  
1208 Therefore the formulas underlying the different DSs had the same output.

1209 In the second set of experiments, we synthetically produced more com-  
1210 plex provenance polynomials, corresponding to more complex queries, that  
1211 resulted in exponents and coefficients in the provenance polynomials that  
1212 were greater than (or equal to) 1. These experiments highlighted the differ-  
1213 ences between the four DSs. While the DS based on lineage rewards all the  
1214 tuples used by a query equally, the strategies based on why-provenance and  
1215 responsibility give more credit to tuples that are more critical to the query.

In particular, why-provenance consider the different ways in which a tuple is used in a query, while responsibility considers the relative importance of a tuple in the generation of the output. How-provenance is even more sensitive to the tuple’s role: it also considers the frequency with which a tuple or a set of tuples is used.

In the third set of experiments, we showed how the differences between the DS are compounded over time, i.e. when more and more queries are processed by the system.

In the fourth set of experiments we compared traditional citations to authors to the credit accrued to them via the DSs. We showed how, in both real-world and synthetic scenarios, credit rewards authors who contribute/curate data that has the highest impact, and therefore receives the biggest quantity of credit, and not necessarily the data with the highest citation count. In this sense, credit appears to be an useful new measure to discover data and their corresponding curators that have a high impact in the research world, even when they are cited few times or do not appear at all in the data that are cited (i.e. the case of data used to build the output of a query but that is not visualized in the output itself).

In future work, we plan to explore different strategies to generate and distribute credit. In this paper we assumed that each output tuple carries credit 1. In more sophisticated scenarios we can employ different strategies to compute credit, that reflect the importance of cited data. Also, other, and more sophisticated strategies could also be used to decide how credit is distributed between the authors, beyond the uniform distribution used here, in a way to reflect the work performed by them on the cited data.

We will also explore new applications for credit over relational databases. One example is *data pricing*, which gives a price to a query submitted by a user who wants to buy the produced information. Currently, a commonly strategy used for data pricing is based on query rewriting: A database stores a set of views with their price. When a new query arrives, the system rewrites it using the stored views to obtain a query price, a process that can be computationally expensive. We plan to distribute credit through carefully planned and representative queries, and use credit information to define a new, faster, and potentially more flexible pricing function.

Another application is *data reduction* [46], which addresses the problem of reducing the vast – and rapidly expanding – amount of data that is being produced.

Data credit can also address this problem, by helping find “hotspots”

1254 and “coldspots” of data. A hotspot is data in a database (e.g. a tuple) with  
1255 a high quantity of credit, which is therefore valuable for the set of queries  
1256 that execute frequently over the data and distribute the credit. On the other  
1257 hand, a coldspot is data with a low quantity of credit, which is therefore  
1258 considered less important and could be deleted or moved to cheaper and/or  
1259 less efficient memory.

## 1260 Acknowledgement

1261 The work was partially supported by the ExaMode project, as part of the  
1262 European Union H2020 program under Grant Agreement no. 825292.

## 1263 References

- 1264 [1] Abadi, D., Ailamaki, A., Andersen, D., Bailis, P., Balazinska, M., Bern-  
1265 stein, P., Boncz, P., Chaudhuri, S., Cheung, A., Doan, A., Dong, L.,  
1266 Franklin, M. J., Freire, J., Halevy, A., Hellerstein, J. M., Idreos, S., Koss-  
1267 mann, D., Kraska, T., Krishnamurthy, S., Markl, V., Melnik, S., Milo,  
1268 T., Mohan, C., Neumann, T., Chin Ooi, B., Ozcan, F., Patel, J., Pavlo,  
1269 A., Popa, R., Ramakrishnan, R., Ré, C., Stonebraker, M., and Suci, D.  
1270 (2020). The seattle report on database research. *SIGMOD Rec.*, 48(4):44–  
1271 53.
- 1272 [2] Alawini, A., Davidson, S. B., Hu, W., and Wu, Y. (2017). Automating  
1273 data citation in citedb. *PVLDB*, 10(12):1881–1884.
- 1274 [3] Alawini, A., Davidson, S. B., Silvello, G., Tannen, V., and Wu, Y.  
1275 (2018). Data citation: A new provenance challenge. *IEEE Data Eng.*  
1276 *Bull.*, 41(1):27–38.
- 1277 [4] Altman, M., Borgman, C. L., Crosas, M., and Martone, M. (2015). An  
1278 Introduction to the Joint Principles for Data Citation. *Bulletin of the*  
1279 *Association for Information Science and Technology*, 41(3):43–45.
- 1280 [5] Baggerly, K. (2010). Disclose all data in publications. *Nature*,  
1281 467(7314):401–401.
- 1282 [6] Bechhofer, S., Buchan, I. E., De Roure, D., Missier, P., Ainsworth, J. D.,  
1283 Bhagat, J., Couch, P. A., Cruickshank, D., Delderfield, M., Dunlop, I.,  
1284 Gamble, M., Michaelides, D. T., Owen, S., Newman, D. R., Sufi, S., and

- 1285 Goble, C. A. (2013). Why linked data is not enough for scientists. *Future*  
1286 *Gener. Comput. Syst.*, 29(2):599–611.
- 1287 [7] Belter, C. W. (2014). Measuring the Value of Research Data: A Citation  
1288 Analysis of Oceanographic Data Sets. *PLoS ONE*, 9(3):e92590.
- 1289 [8] Bertin-Mahieux, T., Ellis, D., Whitman, B., and Lamere, P. (2011). The  
1290 million song dataset. In *Proceedings of the 12th International Conference*  
1291 *on Music Information Retrieval (ISMIR 2011)*, pages 591–596.
- 1292 [9] Bonifati, A., Martens, W., and Timm, T. (2017). An analytical study of  
1293 large SPARQL query logs. *PVLDB*, 11(2):149–161.
- 1294 [10] Borgman, C. L. (2016). Data Citation as a Bibliometric Oxymoron. In  
1295 Sugimoto, C. R., editor, *Theories of Informetrics and Scholarly Commu-*  
1296 *nication*, pages 93–116. De Gruyter Mouton.
- 1297 [11] Buneman, P. (2006). How to cite curated databases and how to make  
1298 them citable. In *18th International Conference on Scientific and Statistical*  
1299 *Database Management, SSDBM*, pages 195–203. IEEE Computer Society.
- 1300 [12] Buneman, P., Christie, G., Davies, J. A., Dimitrellou, R., Harding, S. D.,  
1301 Pawson, A. J., Sharman, J. L., and Wu, Y. (2020). Why data citation isn’t  
1302 working, and what to do about it. *Database J. Biol. Databases Curation*,  
1303 2020.
- 1304 [13] Buneman, P., Davidson, S. B., and Frew, J. (2016). Why data citation  
1305 is a computational problem. *Commun. ACM*, 59(9):50–57.
- 1306 [14] Buneman, P., Khanna, S., and Tan, W. C. (2001). Why and where: A  
1307 characterization of data provenance. In *Database Theory - ICDT 2001,*  
1308 *8th International Conference*, pages 316–330.
- 1309 [15] Buneman, P. and Silvello, G. (2010). A rule-based citation system for  
1310 structured and evolving datasets. *IEEE Data Eng. Bull.*, 33(3):33–41.
- 1311 [16] Callaghan, S., Donegan, S., Pepler, S., Thorley, M., Cunningham, N.,  
1312 Kirsch, P., Ault, L., Bell, P., Bowie, R., Leadbetter, A. M., Lowry,  
1313 R. K., Moncoiffé, G., Harrison, K., Smith-Haddon, B., Weatherby, a.,  
1314 and Wright, D. (2012). Making Data a First Class Scientific Output:  
1315 Data Citation and Publication by NERC’s Environmental Data Centres.  
1316 *International Journal of Digital Curation*, 7(1):107–113.



- 1317 [17] Candela, L., Castelli, D., Manghi, P., and Tani, A. (2015). Data Jour-  
 1318 nals: A Survey. *Journal of the Association for Information Science and*  
 1319 *Technology*, 66(9):1747–1762.
- 1320 [18] Cheney, J., Chiticariu, L., and Tan, W. (2009). Provenance in databases:  
 1321 Why, how, and where. *Foundations and Trends in Databases*, 1(4):379–  
 1322 474.
- 1323 [19] Chockler, H. and Halpern, J. Y. (2004). Responsibility and blame: A  
 1324 structural-model approach. *J. Artif. Intell. Res.*, 22:93–115.
- 1325 [20] CODATA-ICSTI Task Group on Data Citation Standards and Practices  
 1326 (2013). *Out of Cite, Out of Mind: The Current State of Practice, Policy,*  
 1327 *and Technology for the Citation of Data*, volume 12.
- 1328 [21] Cousijn, H., Feeney, P., Lowenberg, D., Presani, E., and Simons, N.  
 1329 (2019). Bringing citations and usage metrics together to make data count.  
 1330 *Data Science Journal*, 18(1).
- 1331 [22] Cronin, B. (1984). *The Citation Process. The Role and Significance of*  
 1332 *Citations in Scientific Communication*. London: Taylor Graham.
- 1333 [23] Cronin, B. (2001). Hyperauthorship: A postmodern perversion or evi-  
 1334 dence of a structural shift in scholarly communication practices? *JASIST*,  
 1335 52(7):558–569.
- 1336 [24] Cui, Y., Widom, J., and Wiener, J. L. (2000). Tracing the lineage of  
 1337 view data in a warehousing environment. *ACM Trans. Database Syst.*,  
 1338 25(2):179–227.
- 1339 [25] Davidson, S. B., Deutch, D., Milo, T., and Silvello, G. (2017). A model  
 1340 for fine-grained data citation. In *CIDR 2017, 8th Biennial Conference on*  
 1341 *Innovative Data Systems Research*. [www.cidrdb.org](http://www.cidrdb.org).
- 1342 [26] Dosso, D. and Silvello, G. (2020). Data credit distribution: A  
 1343 new method to estimate databases impact. *Journal of Informetrics*,  
 1344 14(4):101080.
- 1345 [27] Dubernet, M. L., Antony, B. K., Ba, Y. A., et al. (2016). The vir-  
 1346 tual atomic and molecular data centre (VAMDC) consortium. *Journal of*  
 1347 *Physics B: Atomic, Molecular and Optical Physics*, 49(7):074003.

- 1348 [28] Eiter, T. and Lukasiewicz, T. (2002). Complexity results for structure-  
1349 based causality. *Artif. Intell.*, 142(1):53–89.
- 1350 [29] ESIP Data Preservation and Stewardship Committee (EDPSC) (2019).  
1351 Data citation guidelines for earth science data, version 2. Version 2, Earth  
1352 Science Information Partners.
- 1353 [30] Fang, H. (2018). A discussion of citations from the perspective of the  
1354 contribution of the cited paper to the citing paper. *JASIST*, 69(12):1513–  
1355 1520.
- 1356 [31] Garfield, E. (1999). Journal impact factor: a brief review. *Can. Med.*  
1357 *Assoc.*, 979-980.
- 1358 [32] Gößwein, B., Miksa, T., Rauber, A., and Wagner, W. (2019). Data  
1359 identification and process monitoring for reproducible earth observation  
1360 research. In *2019 15th International Conference on eScience (eScience)*,  
1361 pages 28–38. IEEE.
- 1362 [33] Green, T. J., Karvounarakis, G., and Tannen, V. (2007). Provenance  
1363 semirings. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-*  
1364 *SIGART symposium on Principles of database systems*, pages 31–40. ACM.
- 1365 [34] Halpern, J. Y. and Pearl, J. (2013). Causes and explanations: A  
1366 structural-model approach — part 1: Causes. *CoRR*, abs/1301.2275.
- 1367 [35] Harding, S. D., Sharman, J. L., Faccenda, E., Southan, C., Pawson,  
1368 A. J., Ireland, S., Gray, A. J. G., Bruce, L., Alexander, S. P. H., Anderton,  
1369 S., Bryant, C., Davenport, A. P., Doerig, C., Fabbro, D., Levi-Schaffer, F.,  
1370 Spedding, M., Davies, J. A., and Nc-Iuphar (2018). The IUPHAR/BPS  
1371 guide to PHARMACOLOGY in 2018: updates and expansion to encom-  
1372 pass the new guide to IMMUNOPHARMACOLOGY. *Nucleic Acids Re-*  
1373 *search*, 46(Database-Issue):D1091–D1106.
- 1374 [36] Hartley, J. (2017). Authors and their citations: a point of view. *Scien-*  
1375 *tometrics*, 110(2):1081–1084.
- 1376 [37] Hey, T., Tansley, S., and Tolle, K. M. (2009). Jim Gray on eScience: a  
1377 transformed scientific method.

- [38] Honor, L. B., Haselgrove, C., Frazier, J. A., and Kennedy, D. N. (2016). Data citation in neuroimaging: proposed best practices for data identification and attribution. *Frontiers in neuroinformatics*, 10:34.
- [39] Joshi-Tope, G., Gillespie, M., Vastrik, I., D'Eustachio, P., Schmidt, E., de Bono, B., Jassal, B., Gopinath, G. R., Wu, G. R., Matthews, L., Lewis, S., Birney, E., and Stein, L. (2005). Reactome: a knowledgebase of biological pathways. *Nucleic Acids Research*, 33(Database-Issue):428–432.
- [40] Katz, D. (2014). Transitive credit as a means to address social and technological concerns stemming from citation and attribution of digital products. *Journal of Open Research Software*, 2(1).
- [41] Kosten, J. (2016). A classification of the use of research indicators. *Scientometrics*, 108(1):457–464.
- [42] Lawrence, B., Jones, C., Matthews, B., Pepler, S., and Callaghan, S. (2011). Citation and Peer Review of Data: Moving Towards Formal Data Publication. *International Journal of Digital Curation*, 6(2):4–37.
- [43] Martone, M. (2014). Joint declaration of data citation principles. *FORCE11. San Diego CA. Data Citation Synthesis Group*. <https://www.force11.org/datacitationprinciples>, online September 2020.
- [44] Meho, L. I. and Yang, K. (2007). Impact of data sources on citation counts and rankings of LIS faculty: Web of science versus scopus and google scholar. *Journal of the american society for information science and technology*, 58(13):2105–2125.
- [45] Meliou, A., Gatterbauer, W., Moore, K. F., and Suciu, D. (2010). The complexity of causality and responsibility for query answers and non-answers. *Proc. VLDB Endow.*, 4(1):34–45.
- [46] Milo, T. (2019). Getting rid of data. *Journal of Data and Information Quality (JDIQ)*, 12(1):1–7.
- [47] Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D., Breckler, S. J., Buck, S., Chambers, C. D., Chin, G., Christensen, G., Contestabile, M., Dafoe, A., Eich, E., Freese, J., Glennerster, R., Goroff, D., Green, D. P., Hesse, B., Humphreys, M., Ishiyama, J., Karlan, D., Kraut, A., Lupia, A., Mabry, P., Madon, T., Malhotra, N., Mayo-Wilson,

- 1410 E., McNutt, M., Miguel, M., Paluck, E. L., Simonsohn, U., Soderberg, C.,  
 1411 Spellman, B. A., Turitto, J., VandenBos, G., Vazire, S., Wagenmakers,  
 1412 E. J., Wilson, R., and Yarkoni, T. (2015). Promoting an open research  
 1413 culture. *Science*, 348(6242):1422–1425.
- 1414 [48] Parsons, M. A., Duerr, R. E., and Jones, M. B. (2019). The history and  
 1415 future of data citation in practice. *Data Science Journal*, 18(1).
- 1416 [49] Peters, I., Kraker, P., Lex, E., Gumpenberger, C., and Gorraiz, J.  
 1417 (2016). Research data explored: An extended analysis of citations and  
 1418 altmetrics. *Scientometrics*, 107(2):723–744.
- 1419 [50] Pröll, S. and Rauber, A. (2013). Scalable data citation in dynamic,  
 1420 large databases: Model and reference implementation. In *Proceedings of*  
 1421 *the 2013 IEEE International Conference on Big Data, 6-9 October 2013,*  
 1422 *Santa Clara, CA, USA*, pages 307–312.
- 1423 [51] Rauber, A., Ari, A., van Uytvanck, D., and Pröll, S. (2016). Identifi-  
 1424 cation of Reproducible Subsets for Data Citation, Sharing and Re-Use.  
 1425 *Bulletin of IEEE Technical Committee on Digital Libraries, Special Issue*  
 1426 *on Data Citation*, 12(1):6–15.
- 1427 [52] Rauber, A., Asmi, A., van Uytvanck, D., and Proell, S. (2015). Data  
 1428 citation of evolving data: Recommendations of the working group on data  
 1429 citation (wgdc). *Result of the RDA Data Citation WG*, 20.
- 1430 [53] Silvello, G. (2018). Theory and practice of data citation. *J. Assoc. Inf.*  
 1431 *Sci. Technol.*, 69(1):6–20.
- 1432 [54] Simmhan, Y., Plale, B., and Gannon, D. (2005). A survey of data  
 1433 provenance in e-science. *SIGMOD Record*, 34(3):31–36.
- 1434 [55] Spengler, S. (2012). Data Citation and Attribution: A Funder’s Per-  
 1435 spective. In of Sciences’ Board on Research Data, N. A. and Information,  
 1436 editors, *Report from Developing Data Attribution and Citation Practices*  
 1437 *and Standards: An International Symposium and Workshop*, pages 177–  
 1438 178. National Academies Press: Washington DC.
- 1439 [56] Vogelsgesang, A., Haubenschild, M., Finis, J., Kemper, A., Leis, V.,  
 1440 Mühlbauer, T., Neumann, T., and Then, M. (2018). Get real: How bench-  
 1441 marks fail to represent the real world. In *Proceedings of the Workshop on*  
 1442 *Testing Database Systems*, pages 1–6.

- 1443 [57] Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G.,  
 1444 Axton, M., Baak, A., Blomberg, N., Boiten, J., da Silva Santos, L. B.,  
 1445 Bourne, P. E., et al. (2016). The fair guiding principles for scientific data  
 1446 management and stewardship. *Scientific data*, 3.
- 1447 [58] Wu, Y., Alawini, A., Davidson, S. B., and Silvello, G. (2018). Data  
 1448 citation: Giving credit where credit is due. In *Proceedings of the 2018*  
 1449 *International Conference on Management of Data, SIGMOD*, pages 99–  
 1450 114.
- 1451 [59] Zeng, T., Wu, L., Bratt, S., and Acuna, D. E. (2020). Assigning credit to  
 1452 scientific datasets using article citation networks. *Journal of Informetrics*,  
 1453 14(2).
- 1454 [60] Zou, C. and Peterson, J. B. (2016). Quantifying the scientific output of  
 1455 new researchers using the zp-index. *Scientometrics*, 106(3):901–916.
- 1456 [61] Zwölf, C. M., Moreau, N., and Dubernet, M.-L. (2016). New Model for  
 1457 Datasets Citation and Extraction Reproducibility in VADMC. *Journal of*  
 1458 *Molecular Spectroscopy*, 327:122–137.