Mining Meaningful Keys and Foreign Keys with High Precision and Recall

Henning Koehler
DataViadotto Limited
Auckland, New Zealand
h.koehler@viadotto.tech

Sebastian Link
DataViadotto Limited
Auckland, New Zealand
s.link@viadotto.tech

ABSTRACT

We demonstrate a next-generation Entity/Relationship (E/R) Profiler that mines meaningful key/foreign key relationships from a given data repository. Core novelties include a strict hierarchy of key variants ranging from candidate keys to SQL unique constraints that represent different ways to identify incomplete entities, a measure of orthogonality that separates accidental from meaningful keys, and algorithms for mining approximate keys for all these variants under different thresholds of arity, completeness, dirtiness, and orthogonality. We showcase the high precision and recall achieved by our tool and how it facilitates the users' understanding which entity and referential integrity constraints govern their data.

PVLDB Reference Format:
Henning Koehler and Sebastian Link. Mining Meaningful Keys and Foreign Keys with High Precision and Recall. PVLDB, 18(12): 5363 - 5366, 2025.
doi:10.14778/3750601.3750672

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://www.viadotto.tech.

Figure 1: Role of E/R Profiling

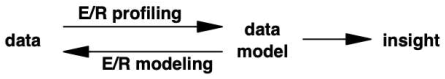
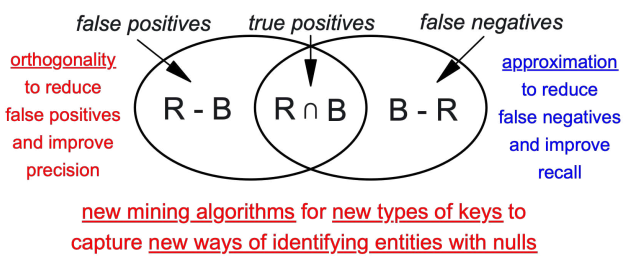


Figure 2: Novel Features and Their Impact on E/R Profiling



1 INTRODUCTION

E/R profiling is the task of identifying business rules that govern entity and referential integrity from a given data repository, primarily targeted at keys and foreign keys. As illustrated in Fig. 1, E/R profiling is the data-driven counterpart to classical E/R modeling, supporting critical tasks such data management, exchange, or integration. In short, enterprises need to know how business entities, such as products or customers, and relationships between them, such as sales or acquisition, can be identified uniquely.

E/R profiling presents a set R of uniqueness constraints (UCs) and inclusion dependencies (INDs) mined from a data repository to users who identify the set B of business rules that govern entity and referential integrity. The elusive goal is for R to be B. Fig. 2 illustrates the reality where $R \cap B$ denotes true positives of the mining process, $R - B$ contains false positives in R that hold accidentally on the repository but do not constitute a business rule, and $B - R$ features false negatives in B that are not in R after mining.

While academic prototypes such as [1, 7, 8] have excellent capabilities for mining but have not focused on identifying business rules, commercial prototypes provide excellent interfaces to validate rules that users put in but fall short in mining UCs and INDs. We will describe the next generation of the only E/R profiler [4] we are aware of. As illustrated in Fig. 2, it does not only provide first means to improve precision and recall of the original profiler by orthogonal and approximate mining, but comprises algorithms for discovering an entire hierarchy of novel key variations with new ways to identify incomplete entities. After outlining the architecture and profiling process, we highlight novel dimensions for E/R profiling, core insights for our audience, and how they experience these insights through interactions with the tool. Technical definitions, algorithms and performance results are found here [5].

2 NEXT-GEN E/R PROFILING

We describe the architecture and features of our profiler.

2.1 Architecture and Profiling Process

Fig. 3 illustrates the E/R profiling process supported by our tool.

Users connect to the data source and sample their data based on the *mod* 2^i approach [3], which is faster and more accurate than any min hashing variant [6]. *Sampling* guarantees that mining of UCs and INDs scales in the number of records. Most tables in our demo retain all records as their size permits efficient mining.

In our *Finder* menu, users set parameters that control the efficiency and outcome of mining. An overview is given in Table 1. Firstly, the *key variant* provides different ways to identify records

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit https://creativecommons.org/licenses/by-nc-nd/4.0/ to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment, Vol. 18, No. 12 ISSN 2150-8097.
doi:10.14778/3750601.3750672

Figure 3: Architecture of the DataViadotto Profiler

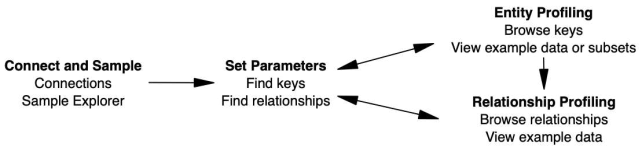


Table 1: New parameters to control E/R profiling

Parameter	*Profiling dimension controlled*
key variant	how to identify entities with nulls
arity	column scalability
sampling	record scalability
completeness	scope of entity integrity
dirtiness	recall
orthogonality	precision

with nulls, leading to algorithms that mine possible (= SQL `UNIQUE`), domain, partial, certain, and candidates for primary keys. Since UC mining is $W[2]$- and IND mining is $W[3]$-complete in the arity [2], there are limitations to column scalability. Users can specify the maximum *arity* as parameter for mining UCs and INDs to control scale in columns. The *completeness* ratio says how many percent of records do not have nulls in any UC field to be mined. All key variants coincide when completeness is set to 100%. Otherwise the ratio says how close a key variation is to a candidate key. The *dirtiness* ratio says up to what percentage of records need removal to obtain a valid key. Non-zero dirtiness triggers approximate mining, returning keys despite duplicate entities up to the threshold. This controls the recall of business rules. In practice, small increases in recall typically incur larger decreases in precision. *Orthogonality* sets a level of anti-correlation required between subsets of a key, with level zero representing that orthogonality is not tested. Intuitively, adding sufficiently many columns leads to a key as any two records will likely differ on some column. As major novelty we propose that accidental keys can be separated from meaningful keys because their data distributions on individual columns make it likely to form a key, in contrast to meaningful keys. While this is not always entirely accurate, it is still vital for dealing with overwhelmingly many key candidates, especially huge numbers of false positives during approximate key mining. Hence, we can control precision. For IND mining, users can set the arity, restrict search to foreign keys, and specify thresholds for the uniqueness of referenced columns (100% for foreign keys), under simple, partial, and full null semantics, respectively. Users can upload a list of UCs to restrict INDs mining to foreign keys that reference some UC in the list. As Fig. 3 illustrates by the edge from entity to relationship profiling, we recommend using results of entity profiling for relationship profiling.

The *Browser* menu gives access to mining results, including measures that say how well constraints hold on the data, and to example data users inspect to understand whether constraints form business rules. The example data provide evidence for the minimality of UCs, and show examples of any constraint violations. This interaction between User and Profiler enhances the users' understanding how data and business rules interact in the given application. Users are guided towards understanding which business rules govern the data, a main purpose of E/R Profiling. Once a UC or IND has been identified as meaningful, any violations constitute dirty data. Likewise, identifying records as sensible will highlight to users that a constraint is not sensible. Data examples also feature records with nulls to further understanding of the underlying semantics, and highlight opportunities for potential imputation of missing data. Users can also see the uniqueness and completeness ratios for all subsets of a UC returned, and also inspect example data for any subset. The ability to set thresholds for simple, partial and full semantics by which INDs should hold, empowers users with the discovery of approximate referential constraints that suffer from violations. Hence, data quality challenges for the discovery of sensible constraints are turned into opportunities by the tool.

Under *Analysis*, users directly enter a UC or IND of interest. The tool returns all measures for the constraint, and users inspect example data to increase their understanding of the constraint.

Fig. 3 encourages an iterative process where users explore results of mining with different input parameters. This process supports the ultimate goal of identifying which constraints are meaningful.

2.2 Novelty and Impact

There are great surveys on mining constraints [1]. As described in [4], our initial profiler brought together the best from academic prototypes and commercial tools. This demo will illustrate how several novel features elevate the efficacy of E/R profiling, as illustrated in Fig. 2. Foremost, these include i) a strict hierarchy of keys, including new notions, ii) orthogonality to control precision of profiling, iii) approximation to control recall of profiling, and iv) algorithms for all key variants with orthogonality and approximation.

Actioning insight from data are narratives assembled from business entities and relationships that govern the data. As illustrated in Fig. 1, E/R profiling facilitates that data and data models match to maximize actionable insight in our high-demand data era.

Getting to know how entities and relationships can be identified is central to all data processing, management and analytical tasks. Direct benefits include: (1) Organize data effectively and efficiently by (a) Pathways to critical data elements, (b) Points of reference for joins, (c) Integrating data assets using matching fields across tables, (d) despite missing and inconsistent data; (2) Better data quality by (a) Enforcing entity integrity with keys, (b) Enforcing referential integrity with foreign keys, (c) Finding all ids used for the same entity, (d) Imputing missing data, (e) Removing data redundancy with foreign keys, (f) Avoiding sources of inconsistent data; and (3) Higher performance by (a) Faster access using unique indexes, (b) Optimizing join types and other queries, (c) Trust in data with business keys and foreign keys, (d) Actual insight from accurate reports, and (e) Less biased, more explainable, higher quality analytics.

The new features of our profiler maximize automation and human-computer engagement in finding business keys and foreign keys, and therefore minimize resources required while solving tasks critical for various data professionals, such as (i) amplify the accuracy and transparency of reports that analysts communicate, (ii) optimize the fit of logical data models that architects create, (iii) boost the effectiveness of data pipelines that engineers build, (iv) surge the effectiveness of feature stores that scientists manage, and (vi) magnify data linkage and insight from catalogs stewards maintain.

Figure 4: Parameter setting for key mining

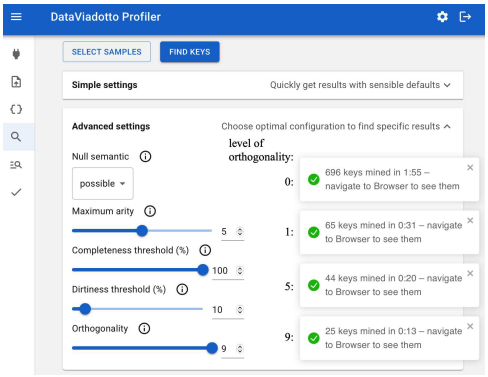
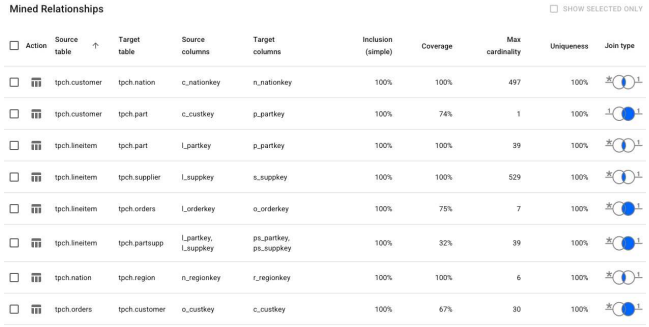


Figure 5: Analyzing sample data to validate key candidate

tpch.lineitem [l_orderkey,l_partkey]

l_shipdate	l_orderkey	l_suppkey	l_quantity	l_returnflag	l_partkey	l_linestatus	l_shipmode	l_linenumber
1998-10-25	4898374	7328	21	N	122303	O	RAIL	1
1994-04-30	1590181	7328	18	A	122303	F	AIR	2
1994-09-19	3418950	8864	36	R	151318	F	TRUCK	1
1994-07-13	3418950	4627	50	R	32123	F	REG AIR	2

Figure 6: Browsing interface for foreign key candidates



3 DEMONSTRATION

Our demonstration focuses on how the audience will experience the following *core takeaways* through interactions with our profiler. (1) Arity, completeness, and dirtiness are important parameters that control E/R profiling and its scalability, (2) Mining results are only recommendations, and a platform for interacting with data is required to identify meaningful constraints from those recommendations, (3) Orthogonality helps separate accidental from meaningful keys, thereby increasing precision of profiling, (4) Approximation helps uncover meaningful constraints violated by dirty data, thereby increasing recall of profiling, (5) Orthogonality makes approximate mining more sensible and scalable, (6) Meaningful keys provides a sound basis for sensible and scalable relationship profiling, (7) Different variants of keys are required to address different ways in which incomplete entities can be identified.

3.1 Scenario 1: TPC-H

We mine all tables of the TPC-H benchmark[1]. The audience will experience the process of E/R profiling including sampling, selecting variants of keys and parameters for mining them, witnessing the speed of mining, browsing results, validating them through our example interface, and utilizing the results for mining INDs.

Our scenario will demonstrate the first six core takeaways. Indeed, due to its novel features our profiler mines the primary keys and foreign keys from all tables with 100% recall and only few false positives (high precision) in under ten seconds, respectively.

As in Fig. 4, the audience mines possible keys up to arity five, with 100% completeness (no nulls), and up to 10% dirtiness. Without orthogonality (level 0) the miner needs 115 seconds to mine 696 constraints that satisfy these parameters. With the highest level of orthogonality (9) most false positives are removed and 25 candidate keys are returned in 13 seconds. All candidates hold exactly (with dirtiness 0), contain the primary key for each table, ten meaningful alternative keys, and seven with column `comment` that should not be part of any meaningful key. Specifying the alternative keys will lead to more efficacious data management. We will also illustrate how increasing arity and dirtiness or decreasing completeness increase mining time and introduce more false positives.

[1]https://www.tpc.org/tpch/

We will invite the audience to use our browsing interface for analyzing the composite key {*l_orderkey*,*l_partkey*} that was returned as a candidate for the largest table `lineitem`. A brief analysis of our sample data in Fig. 5 suggests that this is meaningful composite key because every part should only be listed once on every order. Indeed, column *l_quantity* lists how many times the same part is included in the same order. Since the primary key on *lineitem* is {*l_orderkey*,*l_lineitem*}, this illustrates how E/R profiling leads to the discovery of additional meaningful keys invaluable for managing, processing, and analyzing the data.

Similarly, we highlight the impact of our features in foreign key mining by prompting mining with different parameter choices. Mining exact foreign keys up to arity three with coverage 0% returns 34 candidates in 4 seconds, while coverage 30% returns 13 candidates in the same time. Allowing arbitrary INDs returns 83 candidates in 6 seconds. Fig. 6 shows the browsing interface for the mined foreign key candidates, including measures and a summary of the join types they represent. The false positive candidate from *customer* to *part* table illustrates that results need to be carefully analyzed.

3.2 Scenario 2: Mining Variants of Keys

The main purpose of Scenario 2 is to illustrate the different ways by which incomplete entities can be identified (Takeaway 6), leading to a strict hierarchy between possible, domain, partial, certain and candidate keys. Using a toy data set, users can mine these different notions of keys and analyze the results using our example interface.

Figure 7: Partial keys

(a) Mining results

Columns	Dirtiness	Completeness
Salary, Job	0%	25%
Name, Job	0%	50%
Job	50%	50%
Salary	25%	75%
DoB, Job	25%	25%
DoB	50%	75%
Name, Salary	25%	75%
Name, DoB	25%	75%
Name	25%	100%

(b) Partial key that is not certain sample [Salary,Job]

Name	DoB	Salary	Job
John	null	null	Analyst
Susan	01/03/1985	100k	null
Dave	06/06/1991	75k	null
John	01/03/1985	90k	Developer

Figure 8: Mining possible keys on *hockey* database

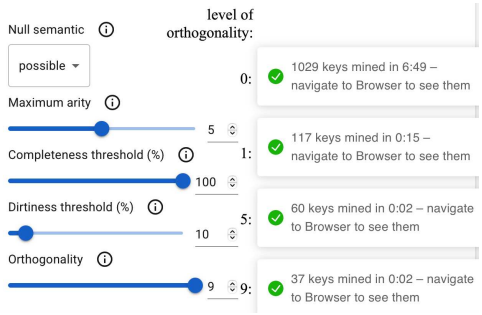


Fig. 8a lists all partial keys and their degrees of dirtiness and completeness on the sample table. Fig. 8b shows why {*Job*,*Salary*} is a partial key that is not certain. Indeed, no two different records match partially, but the `null` for `John` can be replaced by `100k`, and the `null` for `Susan` can be replaced by `Analyst`, leading to a possible world where different entities have the same job and salary. We will illustrate all key variants and their importance for mining.

3.3 Scenario 3: High Precision and Recall

The demo introduces the *Hockey* data set[2], which has 22 tables, 100k records, 300 columns, and a size of 15.6MB. Nine tables do not have a primary key, and no table has a business key. There is lots of missing data, which makes the data set a representative use case.

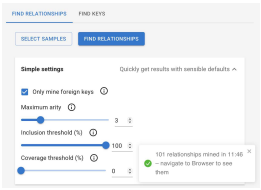
As in Fig. 8, the audience will mine candidate keys without orthogonality (level 0) taking 6:55 minutes for 1,029 candidates, while level 1 finds 117 keys in 15 seconds, level 5 finds 60 keys in 2 seconds, and level 9 returns 37 keys in 2 seconds. The latter include meaningful keys for 14 tables, level 1 generates meaningful keys for 5 out of the remaining 8 tables. Finally, level 1 finds some meaningful keys for 2 out of the remaining 3 tables. Hence, the audience will experience how meaningful keys are discovered gradually.

Table *Master* manages people with roles such as players (playerID), coaches (coachID), and hall of famers (hofID). Nulls occur whenever a person never assumes the corresponding role. Hence, the table does not exhibit any approximate candidate key. This is a common design flaw, but our novel notion of partial keys can reveal them. Mining partial keys up to arity seven with 10% dirtiness, 0% completeness, and level 5 orthogonality returns 12 keys in 4 seconds. Two keys are exact and sensible, such as {*playerID*,*coachID*,*hofID*}.

[2]relational.fit.cvut.cz/dataset/Hockey

Figure 9: Foreign key mining with and without set keys

(a) Mining from scratch



(b) Using keys as references

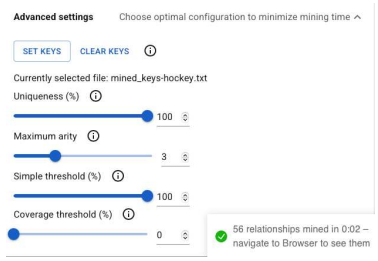


Figure 10: Meaningful foreign keys that are violated

Source table	Target table	Source columns	Target columns	Inclusion (simple)	Inclusion (partial)	Inclusion (full)	Uniqueness	Join type
Hockey.SeriesPost	Hockey.TeamsPost	year, tmIDWinner	year, tmID	99%	99%	99%	100%	
Hockey.SeriesPost	Hockey.TeamsPost	year, tmIDLoser	year, tmID	99%	99%	99%	100%	

Mining foreign keys from all 22 tables requires 11:46 minutes to produce 101 candidates, while mining only foreign keys that reference previously identified business keys produces 56 candidates within two seconds. The latter include all foreign keys enforced by the schema, and several new ones. Lower inclusion thresholds reveal further meaningful foreign keys that violate referential integrity due to not being enforced. Fig. 10 shows two examples.

4 CONCLUSION

The demo pinpoints how the challenge of identifying meaningful constraints from incomplete and inconsistent data is overcome by combining novel notions of keys, approximation, and orthogonality with mining algorithms whose scalability, precision, and recall are controlled by parameters such as arity, completeness, and dirtiness.

REFERENCES

[1] Ziawasch Abedjan, Lukasz Golab, Felix Naumann, and Thorsten Papenbrock. 2018. *Data Profiling.* Morgan & Claypool Publishers.

[2] Thomas Bläsius, Tobias Friedrich, and Martin Schirneck. 2022. The complexity of dependency detection and discovery in relational databases. *Theor. Comput. Sci.* 900 (2022), 79–96.

[3] A. Broder. 1997. On the Resemblance and Containment of Documents *(SEQUENCES).* IEEE Computer Society.

[4] Henning Koehler and Sebastian Link. 2024. Entity/Relationship Profiling. In *IEEE ICDE.* 5393–5396.

[5] Henning Koehler and Sebastian Link. 2025. Orthogonal Key Mining for Incomplete and Inconsistent Relations. https://www.dropbox.com/scl/fi/hjuvkli9y39u2el6xbacn/tech-report_mining.pdf?rlkey=va7r3cnndd0k83dlumrxhvp0m&st=d4rk71c5&dl=0. 14.07.2025.

[6] Henning Koehler, Xiaofang Zhou, Shazia Wasim Sadiq, Yanfeng Shu, and Kerry L. Taylor. 2010. Sampling dirty data for matching attributes. In *SIGMOD.* 63–74.

[7] Thorsten Papenbrock, Tanja Bergmann, Moritz Finke, Jakob Zwiener, and Felix Naumann. 2015. Data Profiling with Metanome. *Proc. VLDB Endow.* 8, 12 (2015), 1860–1863.

[8] Fabian Tschirschnitz, Thorsten Papenbrock, and Felix Naumann. 2017. Detecting Inclusion Dependencies on Very Many Tables. *ACM Trans. Database Syst.* 42, 3 (2017), 18:1–18:29.