# Examples are All You Need: Iterative Data Discovery by Example in Data Lakes

El Kindi Rezig[1], Anshul Bhandari[2], Anna Fariha[3], Benjamin Price[4], Allan Vanterpool[5], Andrew Bowne[5], Lindsey McEvoy[5], Vijay Gadepally[4]

[1]MIT CSAIL, [2]NIT Hamirpur, [3]Microsoft, [4]MIT Lincoln Laboratory, [5]United States Air Force
elkindi@csail.mit.edu, 185529@nith.ac.in, annafariha@microsoft.com, {ben.price, vijayg}@ll.mit.edu, {allan.vanterpool, lindsey.mcevoy.1}@us.af.mil, andrew.bowne.2@spaceforce.mil

Data-science development is highly experimental and incremental, i.e., it takes dozens of iterations to produce a satisfactory pipeline. At the core of any data-science pipeline, lies the data. As a result, building effective data-science pipelines often hinges on finding the right data to be consumed by the downstream operators. While the database community has successfully developed sophisticated query languages for data retrieval, those only work when data resides in a traditional relational database, conforming to a predefined schema. Even then, the users are required to master the query languages, have a good understanding of the schema, and, finally, they must know *what* they are looking for (e.g., records) and *how* to find it (e.g., which tables to join). However, in the era of data science, data scientists often need to extract data from data lakes, which typically lack meta-data (from which the schema can be inferred) and are too large to sift through manually. As a result, there is a renewed interest in designing novel interfaces to discover relevant data from data lakes effectively and efficiently. Yet, current systems pose a high entry barrier for the users, i.e., users must have expertise in query languages and knowledge of the underlying structure of the lake.

In the course of our collaboration with the U.S. Air Force, we realize that there is a pressing need to develop a data-discovery system that (1) liberates the users from the requirement of learning and composing complex queries, and, thus, makes data discovery easy for them; (2) returns effective discovery results that satisfy the user's intent; and (3) can return results efficiently. To this end, we introduce *DICE*—a *Data dIsCovery by Example* [2] system that (1) does not require any query-language expertise and only needs a few *exemplar* records to find relevant data from data lakes; (2) involves users in the discovery process (since an example-driven search often suffers from intent ambiguity, it is important to allow users to steer the discovery process iteratively); and (3) can return "quick-and-dirty" results interactively.

Data lakes contain heterogeneous data sources with limited meta-data (e.g., schema, integrity constraints). As a result, finding data of interest from data lakes is often time-consuming and labor-intensive, e.g., one would need to find joinable tables to produce the desired result. Conceptually, data scientists employ one of the following two approaches to find data of interest from data lakes: (1) Sift through the data lake manually (e.g., through a series of SQL queries) and find relevant tables and joins. While this method does not require leanring special tools, it is too primitive and time-consuming. (2) Use a data-discovery system that facilitates finding joinable tables [1]. While existing data-discovery systems [1] mitigate some of the challenges mentioned above, they still require users to formulate queries or have knowledge of the structure of the lake. Since users are often aware of a few examples of what they are looking for, *DICE* follows a by-example paradigm for data discovery, which addresses the limitations of the two aforementioned approaches: (1) it alleviates the need to write queries (which requires knowledge of the data lake structure) and (2) it does not require users to manually inspect the lake to find their data of interest.

The talk will feature a walkthrough of the current prototype of *DICE* and its key contributions including: (1) mapping of user-provided examples to data-lake tables; (2) interactive discovery of Primary Key - Foreign Key relationships from tables; (3) feedback loop to disambiguate the user's intent; and (4) efficient similarity-based lookups to generate results.

## Acknowledgements

Research was sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

## References

[1] R. C. Fernandez, Z. Abedjan, F. Koko, G. Yuan, S. Madden, and M. Stonebraker. Aurum: A data discovery system. In *ICDE*, pages 1001–1012, 2018.

[2] E. K. Rezig, A. Bhandari, A. Fariha, B. Price, A. Vanterpool, V. Gadepally, and M. Stonebraker. DICE: data discovery by example. *PVLDB*, 14(12):2819–2822, 2021.


*12th Annual Conference on Innovative Data Systems Research (CIDR '22)* January 9-12, 2022, Chaminade, USA.