SDG-KG: A Framework to Compute SDG Indicators with Open Data

Wissal Benjira
De Vinci Higher Education, De Vinci Research Center & LASTIG, IGN
Paris, France
wissal.benjira@devinci.fr

Nicolas Travers
De Vinci Higher Education
De Vinci Research Center
Paris, France
nicolas.travers@devinci.fr

Faten Atigui
CEDRIC, CNAM
Paris, France
faten.atigui@lecnam.net

Bénédicte Bucher
LASTIG, IGN
Paris, France
benedicte.bucher@ign.fr

Malika Grim-Yefsah
LASTIG, IGN
Paris, France
malika.grim-yefsah@ign.fr

ABSTRACT

Monitoring Sustainable Development Goal (SDG) indicators requires integrating heterogeneous open datasets from sources such as relational databases, NoSQL stores, and APIs. While SDG indicators follow standardized definitions, open data sources are often fragmented, schema-less, and inconsistent, making both integration and computation challenging. In this demonstration, we present **SDG-KG**, a spatio-temporal Knowledge Graph (KG) framework designed to structure metadata, guide data retrieval, and formalize indicator computation workflows. Our approach leverages graph-based modeling to construct a Metadata Graph, apply conflict resolution techniques when multiple sources provide overlapping data, and dynamically generate query-driven execution plans. Through an interactive interface, users can explore United Nations specifications, inspect data provenance and the generated KG, and visualize the computed indicators.

PVLDB Reference Format:
Wissal Benjira, Nicolas Travers, Faten Atigui, Bénédicte Bucher, and Malika Grim-Yefsah. SDG-KG: A Framework to Compute SDG Indicators with Open Data. PVLDB, 18(12): 5367 - 5370, 2025.
doi:10.14778/3750601.3750673

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/wissalbenjira/sdg-kg.

1 INTRODUCTION

The rapid growth of open data and large-scale information systems has transformed how organizations manage and analyze data. Governments, researchers, and policymakers increasingly rely on diverse and distributed datasets to measure global challenges. Among these, monitoring Sustainable Development Goals (SDGs) remains one of the decade's most demanding task [10, 14]. It is difficult to integrate and query these datasets, as they are stored in heterogeneous data sources. These sources often lack schema consistency, present varying granularity levels, and contain conflictual data [1, 2].

A promising solution to this challenge is the use of Knowledge Graphs (KGs), which provides a flexible and semantically rich data model for integrating diverse sources while preserving relationships between entities [6, 9]. KGs have been widely adopted for data integration and semantic interoperability across disparate sources [12]. Additionally, Large Language Models (LLMs) have demonstrated strong capabilities in schema matching and entity linking, making them valuable for aligning heterogeneous datasets [13].

In this paper, we present **SDG-KG**, which, as far as we know, is the first framework that enables a fully automated, query-driven computation of SDG indicators from open heterogeneous data sources. The framework builds the Metadata Graph that captures from data sources schema relationships and source properties. This structure supports cross-source query execution and provenance tracking without requiring full dataset materialization. In parallel, the framework uses the SDG Graph , a structured representation of SDG-related concepts, including goals, targets, indicators, and their attributes, introduced in our recent work [3]. Finally, LLMs are leveraged by our **SDG-KG** for automated schema alignment and for generating an execution plan.

Our contributions are as follows: (i) We introduce **SDG-KG** , a KG framework for computing SDG indicators with open data; (ii) We generate a Metadata Graph for data source abstraction, linked to the SDG Graph for indicator representation. (iii) We leverage LLMs for schema alignment; (iv) We provide an interactive demonstration for exploring the query-driven SDG indicators computation.

2 SYSTEM OVERVIEW

This section presents the **SDG-KG** workflow, which transforms open data into computed SDG indicators. Figure 1 outlines: (1) Metadata Acquisition, (2) Automated Mapping (3) Query-Driven Data Selection, (4) Conflict Resolution, and (5) Indicator Computation.

(1) Metadata Acquisition. The process begins with user's data uploads from sources such as relational databases, NoSQL stores, or APIs. Instead of directly processing raw data, **SDG-KG** extracts schema metadata and constructs the Metadata Graph (red in the figure), capturing the structure, the typing, attributes, and relationships. The metadata is then transformed into the *Open SDG*[1] format to align with global standards and facilitate queries structuring.

Proceedings of the VLDB Endowment, Vol. 18, No. 12 ISSN 2150-8097.
doi:10.14778/3750601.3750673

[1] https://open-sdg.readthedocs.io/en/latest/data-format

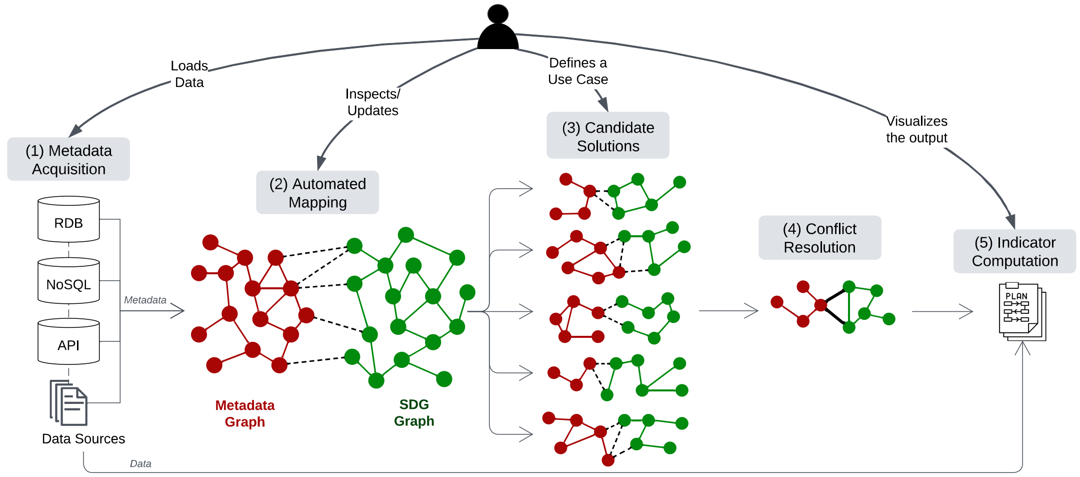


Figure 1: Architecture of the SDG-KG system.

(2) Automated Mapping. The next step is to align dataset attributes with the predefined `SDG Graph` (green in the figure). The latter encodes relationships between SDG goals, targets, indicators, and required concepts. Data sources' attributes are mapped to SDG attributes. This step has been addressed in more details in our previous work [3, 4]. For that, we combine rule-based filtering with LLM-powered schema mapping to establish semantic correspondences between diverse data sources attributes and SDG attributes. Users can review and manually refine the mappings if necessary. The output **SDG-KG** $\mathcal{G}$ is stored in the `Neo4j` graph database.

(3) Query-Driven Data Selection. By choosing an SDG indicator (*i.e.,* `SDG Graph` node) and linked parameters (*e.g.,* spatial area, time period), the user defines its own use case. Optionally, the user can request disaggregation by contextual criteria such as age group, sex, or socioeconomic status. Based on these inputs, multiple datasets may match the selection criteria, but they may vary in granularity, completeness, or reliability.

(4) Conflict Resolution. When dealing with Open Data, a given use case can lead to multiple answering datasets (here candidate solutions as subgraphs $\mathcal{G}_I \subset \mathcal{G}$). **SDG-KG** addresses the issue of overlapping or conflicting information (multiple mappings on same attribute nodes) by applying a resolution strategy [5]. Since **SDG-KG**operates at the metadata level and does not process data streams, we adopt the *Trust Your Friend* approach, introducing a quality-based criterion to prioritize one source over another.

(5) Indicator Computation and Results. Using the chosen sub-graph $G^* \subseteq \mathcal{G}_I$ without conflicts, we generate an execution plan that follows the indicator formula (given by the use case). The computation propagates from the `SDG Graph` to the `Metadata Graph` and guides the workflow process. To finish with, users visualize the spatio-temporal results, both timely evolution and maps with computed values, moreover they can inspect provenance details of the computed indicator.

3 METHODOLOGY

In this section, we build upon our `SDG Graph` model [3]. The Sustainable Development Goals (SDGs) are structured into interconnected components that define measurable targets and indicators for assessing global progress. While LinkedSDGs and SustainGraph [7] model SDG concepts and track target's evolution, `SDG Graph` focuses on modeling indicators for open data computation.

3.1 The SDG Graph Model

The `SDG Graph` represents the hierarchical structure of the SDGs, where entities and relationships are explicitly defined following the United Nations (UN) *SDG Indicator Metadata Repository.*

3.1.1 The SDG Graph. This structure is automatically generated as part of a data-driven approach and results from an expert-validated contribution. The model consists of the following components:

- **Goals:** High-level objectives that address global issues (e.g., *Sustainable Cities and Communities* - SDG 11).
- **Targets:** Specific aims within each goal that define measurable outcomes (e.g., *Provide access to safe, affordable, accessible, and sustainable transport systems*, Target 11.2).
- **Indicators:** Quantifiable metrics for evaluating progress toward each target (e.g., *Proportion of the population that has convenient access to public transport, by sex, age, and persons with disabilities*, Indicator 11.2.1).
- **Concepts:** Domain-specific entities related to SDG measurement (e.g., *Population*, *Public Transport*, *Street Network*).
- **Attributes:** Descriptive properties of concepts used in calculations (e.g., for *Population*: *Space*, *Time*, *Age Group*, *Sex*).

Figure 2 illustrates a sample *SDG Graph*, highlighting the nodes and relationships between goals, targets, indicators, concepts, and attributes. To enable automatic computation, open dataset attributes are mapped to predefined SDG attributes.

3.1.2 SDG Indicator Formula. To compute an indicator, the corresponding formula is embedded in the `SDG Graph`'s indicator node, represented in a structured JSON format that defines how involved concepts are transformed and aggregated. The *formula* is extracted from the UN Metadata repository, where it is expressed in natural language or mathematical notation and must be encoded.

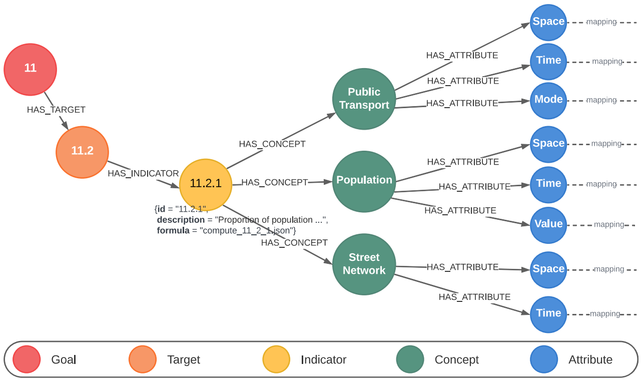


Figure 2: Sample of SDG Graph

Take the following JSON as an example (refers to the *formula "compute_11_2_1.json"* of the indicator node in Figure 2):

```
{
  "AccessPop":[
     "Population",
     {"$ComputeDistance": ["PublicTransport", "StreetNetwork"]}
  ],
  "RETURN": {"$div": ["AccessPop", "Population"]}
}
```

Here, the "AccessPop" variable represents the subset of the population with convenient access to public transport. The "$ComputeDistance" operation calculates the proximity between population clusters and public transport stops using the "StreetNetwork". Finally, the "RETURN" statement computes the population proportion with access by dividing "AccessPop" by the total "Population".

3.2 Data Source Selection Metric

The **SDG-KG** graph $\mathcal{G}$ helps to provide a set of solution sub-graphs $\mathcal{G}_I \subset \mathcal{G}$ where the indicator node I (from use case Q_I), in the SDG Graph, is linked to source nodes $D \subset \mathcal{D}$ in the Metadata Graph.

The goal of this step is to select the most relevant graph $G^* \subseteq \mathcal{G}_I$ containing the best set of sources D among the available sources $\mathcal{D} \subset \mathcal{G}_I$. To achieve this, we propose a *Trust Score Metric (TSM)* that quantifies the alignment between a dataset's metadata and the user-defined query parameters.

Definition 3.1. Let $D \subseteq \mathcal{D} \subset \mathcal{G}_I$ be a dataset with metadata $D = \{D_s, D_t, D_c, D_r, D_p\}$ respectively representing its spatial, temporal, contextual granularity, reliability and completeness information. Let $Q_I = \{q_s, q_t, q_c\}$ be a query specifying spatial, temporal, and contextual constraints. The *TSM* of D regarding Q_I is defined as:

$$TSM(D, Q_I) = \frac{\sum\limits_{i \in \{s,t,c\}} (w_i \cdot \mathrm{Sim}(D_i, q_i)) + w_r \cdot u(D_r) + w_p \cdot v(D_p)}{\sum\limits_{i \in \{s,t,c,r,p\}} w_i}$$

where w_i, w_r, w_p represent user-defined normalized weights [11] controlling the importance of each factor, $w_i + w_r + w_p = 1$. $\mathrm{Sim}(D_i, q_i)$ is the similarity function for each dimension (spatial, temporal, contextual - $q_i \in Q_I$). $u(D_r)$ is the reliability score, assessing the trustworthiness of the dataset source (e.g., governmental, NGO, crowdsourced data) - $u(D_r) \in [0, 1]$. $v(D_p)$ is the completeness score, indicating the proportion of expected records available in the dataset - $v(D_p) \in [0, 1]$.

The resulting *TSM* score is normalized between 0 and 1, where higher values indicate a closer match to the query. This formulation ensures that selected datasets for indicator computation best align with user-defined spatial, temporal, and contextual constraints with respect to the source reliability and completeness score.

The problem of finding the best candidate graph G^* can be formalized as the following:

$$G^* = \arg\max_{G \subset \mathcal{G}_I} \prod_{D \subseteq \mathcal{D} \cap G} \mathrm{TSM}(D, Q_I) \qquad (1)$$

3.3 Indicator Computation Algorithm

Once the indicator is chosen with a formula and a use case Q_I, as well as the best sub-graph solution $G^* \subseteq \mathcal{G}_I$, it requires to translate this sub-graph into an execution plan that links open data sources D to the formula computation $f(I)$. From the structured JSON (cf. Section 3.1.2), we produce a sequence of operations $\theta \in \Theta$ that retrieve, filter, and aggregate data to compute the indicator value.

Each concept from the SDG Graph is linked to at least space s, time t attributes and value nodes (v). Thus, the use case Q_I is applied on each linked source to get corresponding data $R = [s, t, (v)]$. Thus, any operator $\theta \in \Theta$ is defined as: $\theta : R \longrightarrow R$, keeping a closed form.

From the indicator formula, we can apply operations on source nodes linked to the corresponding concept nodes. The combination generates the source code which computes the indicator itself.

Algorithm 1 illustrates this process for indicator 11.2.1, computing the proportion of the population with convenient access to public transport. The algorithm extracts the concepts from the SDG Graph, applies spatial and temporal filters on corresponding sources $D \subset G^*$ from the Metadata Graph, computes service areas, and aggregates population counts to derive the indicator value.

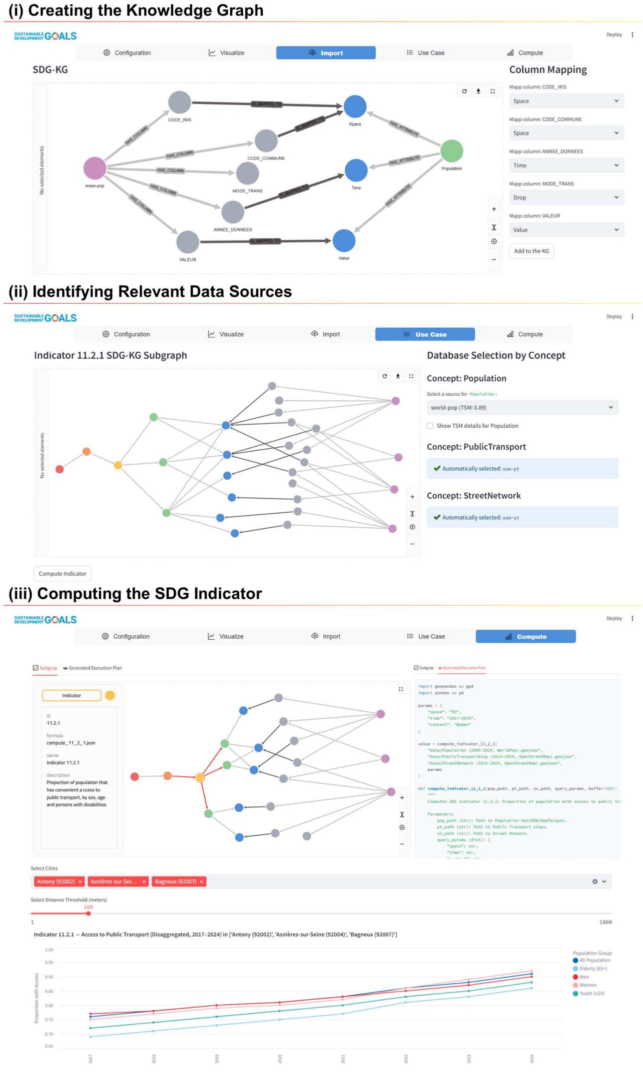
Each algorithm function has an equivalent Cypher query that is applied in the graph database. Outputs from GetConcept functions (*PT, Pop, SN*) on G^* follow the R data model. More details are available in Github.

Algorithm 1 Generated execution plan for Indicator 11.2.1

```
Input: A Solution Graph: $G^*$, Query Parameters $Q_I = \{q_s, q_t, q_c\}$
    Step 1: Retrieve concepts from the graph
    PT ← GetConcept($G^*$, "PublicTransport", $q_s$, $q_t$)
    Pop ← GetConcept($G^*$,"Population", $q_s$, $q_t$)
    SN ← GetConcept($G^*$, "StreetNetwork", $q_s$, $q_t$)
    Step 2: Filter Population
    FilteredPop ← Filter($Pop$, $q_c$)
    Step 3: Compute Service Area
    ServiceArea ← ComputeDistance(PT, SN, 500)
    Step 4: Identify Population within the service area
    PopWithAccess ← Filter(FilteredPop, {$q_s$: In(ServiceArea)})
    Step 5: Compute Indicator Value
    Population ← Sum(GetAttribute(FilteredPop, "Value"))
    AccessPop ← Sum(GetAttribute(PopWithAccess, "Value"))
    RETURN (AccessPop / Population)
```

4 DEMONSTRATION SCENARIOS

In this demonstration, we rely on GPT-3.5 for the LLM-based mapping, following our previous work [3] and Neo4j for graph storage.


(i) Creating the Knowledge Graph

Column Mapping

Mapp column: CODE_IRIS
Space

Mapp column: CODE_COMMUNE
Space

Mapp column: ANNEE_DONNEES
Time

Mapp column: MODE_TRANS
Drop

Mapp column: VALEUR
Value

Add to the KG

(ii) Identifying Relevant Data Sources

Database Selection by Concept

Concept: Population

Select a source for `Population` :
world-pop (TSM: 0.89)

□ Show TSM details for Population

Concept: PublicTransport

✓ Automatically selected: `osm-pt`

Concept: StreetNetwork

✓ Automatically selected: `osm-st`

(iii) Computing the SDG Indicator

id
11.2.1

formula
compute_11_2_1.json

name
Indicator 11.2.1

description
Proportion of population that has convenient a ccess to public transport, by sex, age and persons with disabilities

```
import geopandas as gpd
import pandas as pd

params = {
    "space": "92",
    "time": "2017-2024",
    "context": "Women"
}

value = compute_indicator_11_2_1(
    "data/Population (2000-2024, WorldPop).geojson",
    "data/PublicTransportStop (2014-2024, OpenStreetMap).geojson",
    "data/StreetNetwork (2014-2024, OpenStreetMap).geojson",
    params
)

def compute_indicator_11_2_1(pop_path, pt_path, sn_path, query_params, buffer=500):
    """
    Computes SDG Indicator 11.2.1: Proportion of population with access to public tra

    Parameters:
        pop_path (str): Path to Population GeoJSON/GeoParquet.
        pt_path (str): Path to Public Transport stops.
        sn_path (str): Path to Street Network.
        query_params (dict): {
            "space": str,
            "time": str,
```

Select Cities
Antony (92002) × Asnières-sur-Sei... × Bagneux (92007) ×

Select Distance Threshold (meters)
100
1 1000

Indicator 11.2.1 — Access to Public Transport (Disaggregated, 2017–2024) in ['Antony (92002)', 'Asnières-sur-Seine (92004)', 'Bagneux (92007)']

Figure 3: Visualization Interface

Besides, we focus on indicator 11.2.1 (another can be tested), which addresses access to public transport in urban areas, as it is a key metric for evaluating sustainable urban mobility [8].

We use the following inputs:

- The United Nations' SDG Indicator Metadata Repository
- Three Open data sources censing Population datasets from INSEE, WorldPop, OpenDataParis
- The Open Street Map API for getting Street Networks and Public Transport Stops

The steps correspond to those shown in Figure 3 (demo video).

Step (i). The user uploads one or more datasets. The framework automatically extracts metadata (columns, types) and generates a metadata graph. Next, we use an LLM-assisted schema alignment method to map Metadata attributes to the SDG attributes. Each matching column is mapped to an SDG concept (e.g., Space, Time, Value, Population) using the right-hand panel. The Knowledge Graph is therefore created in which the user can navigate to check the quality. This scenario includes phase (1) and (2) in Figure 1.

Step (ii). The **SDG-KG** identifies matching concepts for each required input of the selected indicator. The user defines query parameters such as geographic area and temporal range. The system then uses the metadata graph and the mappings to identify matching data sources. The user will witness conflicting or overlaping sources in the graph. In that case, the best candidate source is determined. This scenario corresponds to phase (3) and (4) in Figure 1.

Step (iii). The system generates an execution plan based on the selected indicator's formula. It is represented as a JSON structure that defines the sequence of operations. It identifies the required concepts (e.g., Population, PublicTransport) and maps them to the aligned dataset attributes selected in the previous step. The execution plan is then translated into executable code using Python and libraries such as Pandas and GeoPandas. The code can be exported for further executions.

In our scenario, the code handles data loading, preprocessing, spatial joins, distance filtering, and final aggregation. The result is visualized with a provenance path showing which sources and attributes were used. A map is showed to compare computed indicator values in various areas. This scenario corresponds to step (5) in Figure 1. In this scenario, we measure changes in accessibility over time. Such results can often be linked to real-world events. For example, the increase in 2022 matches the preparation for the Paris 2024 Olympic Games, during which new transport stations were built or extended, impacting the indicator.

REFERENCES

[1] R. Alotaibi, B. Cautis, A. Deutsch, M. Latrache, I. Manolescu, and Y. Yang. 2020. ESTOCADA: Towards Scalable Polystore Systems. *PVLDB* 13, 12 (2020).

[2] N. Barret, I. Manolescu, and P. Upadhyay. 2024. Computing Generic Abstractions from Application Datasets. In *EDBT'24*, Vol. 27. Paestum, Italy, 94–107.

[3] W. Benjira, F. Atigui, B. Bucher, M. Grim-Yefsah, and N. Travers. 2025. Automated mapping between SDG indicators and open data: An LLM-augmented knowledge graph approach. *Data & Knowledge Engineering* 156 (2025), 102405.

[4] W. Benjira, F. Atigui, B. Bucher, M. Grim-Yefsah, and N. Travers. 2025. Web Open Data to SDG Indicators: Towards an LLM-Augmented Knowledge Graph Solution. In *Web Information Systems Engineering – WISE'24*.

[5] J. Bleiholder and F. Naumann. 2006. *Conflict Handling Strategies in an Integrated Information System.* Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche. 286–297 pages.

[6] X. Dong, E. Gabrilovich, G. Heitz, et al. 2014. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. *KDD* (2014), 601–610.

[7] E. Fotopoulou, I. Mandilara, A. Zafeiropoulos, C. Laspidou, Gi. Adamos, P. Koundouri, and S. Papavassiliou. 2022. SustainGraph: A knowledge graph for tracking the progress and the interlinking among the sustainable development goals' targets. *Frontiers in Environmental Science* 10 (2022).

[8] F. Hanani and S. Aziz. 2021. Improving traffic congestion assessment by using fuzzy logic approach. *Journal of Theoretical and Applied Information Technology* Vol.99. No 3 (2021), 625–638.

[9] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. De Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann. 2021. Knowledge Graphs. *ACM Comput. Surv.* 54, 4 (2021).

[10] Y. Jiang and D. Johnson. 2023. Data Discovery for the SDGs: A Systematic Rule-based Approach. In *GoodIT'23* (Lisbon, Portugal). 384–391.

[11] R. L. Keeney, H. Raiffa, and David W. Rajala. 1979. *Decisions with Multiple Objectives: Preferences and Value Trade-Offs.* Vol. 9. 403–403 pages.

[12] N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, and J. Taylor. 2019. Industry-scale knowledge graphs: lessons and challenges. *Commun. ACM* 62, 8 (2019), 36–43. https://doi.org/10.1145/3331166

[13] M. Parciak, B. Vandevoort, F. Neven, L. M. Peeters, and S. Vansummeren. 2024. Schema Matching with Large Language Models: an Experimental Study. In *TaDA Workshop at VLDB'24*. VLDB.org.

[14] R. van Loenhout, C. Ranasinghe, A. Degbelo, and N. Bouali. 2022. Physicalizing Sustainable Development Goals Data: An Example with SDG 7. In *CHI EA'22*.