# COSMA: Constraint-based Schema Matching

Keywords: integrity constraints, schema matching, data integration

## Matching of Database Schemas

Schema matching is an important step in a typical **data integration** process. Its goal is to identify correspondences between the individual schema elements (i.e., tables and attributes in the case of relational data). An example is depicted in Figure 1 where we have the two databases D1 and D2. While the schema of D1 consists of a single table, the attributes of D2 are distributed across two tables. An easy to identify correspondence is the one between the two attributes D1.Course and D2.Class. In contrast, it is more difficult to determine the correspondence between D1.Module and D2.M, since their values are encoded differently. As we can also see in this example, correspondences do not need to be one-to-one, but can also be one-to-many or many-to-many, such as $\{Name\} \leftrightarrow \{FName, LName\}$.

**Database D1**

Lectures

| SID | Name | Sem | Course | Module | Lecturer |
|---|---|---|---|---|---|
| A3 | Jane Smith | 3 | AI Intro | A | Asimov |
| C2 | Ben Moore | 4 | Time Series | B | Doc Brown |
| A3 | Jane Smith | 3 | Logic | C | Spock |

**Correspondenes:**

- $\{SID\} \leftrightarrow \{SID\}$
- $\{Course\} \leftrightarrow \{Class\}$
- $\{Lecturer\} \leftrightarrow \{Teacher\}$
- $\{Module\} \leftrightarrow \{M\}$
- $\{Name\} \leftrightarrow \{FName, LName\}$

**Database D2**

Enrollment

| ID | Class | Teacher | Year | M | Student |
|---|---|---|---|---|---|
| 1 | Time Series | Dr. Brown | 2023 | 1 | B3 |
| 2 | Time Series | Dr. Brown | 2022 | 1 | C2 |
| 3 | Logic | Mr. Spock | 2023 | 2 | A3 |

Students

| SID | FName | LName |
|---|---|---|
| A3 | Jane | Smith |
| C2 | Ben | Moore |
| B3 | Marty | McFly |

*Figure 1: Schema Matching Example*

Existing schema matching algorithms [1] detect correspondences using the **similarity** between labels (e.g., attribute names), predefined attribute-specific **constraints** (e.g., data types, check constraints), **structures** within the schemas (e.g., neighbored attributes and foreign keys), and **instance data** (e.g., the values of two attributes). However, as we know from data profiling [2], the instance of a database contains much more information than just the actual data values. This includes *functional dependencies*, *unique constraints*, *inclusion dependencies*, *order dependencies*, or *matching dependencies*. In this master project, we explore the potential of using integrity constraints to match two or more relational schemas. For instance, if we know the functional dependencies $FD1\colon Course \rightarrow Module$ and $FD2\colon Module \rightarrow Lecturer$ for database D1 as well as $FD3\colon Class \rightarrow M$ and $FD4\colon M \rightarrow Teacher$ for database D2, the correspondence $\{Module\} \leftrightarrow \{M\}$ becomes much more evident to us.

## Project Goal

In this master project, we will follow an **entire research cycle** from problem inception and literature research to algorithm development and, finally, to evaluation on real-world datasets. We will prepare a research article and submit it to an **international conference**.

Our goal is to develop an algorithm that can leverage different types of integrity constraints to detect correspondences between the elements of two or more relational schemas. We start

the project with a literature search phase. Afterwards, we decide on and implement (or re-use) baseline approaches to schema matching. Thereafter, we will design and develop our novel COSMA algorithm, which we will evaluate against baseline approaches. We will regard both quality and efficiency of the approaches and deduct additional experiments showing strengths and weaknesses of the developed algorithm.

The basic idea for our COSMA approach consists of the following steps (but of course we are also open for other ideas):

- Discovery of constraints within the individual databases using the data profiling tool **Metanome**
- Building a **hypergraph** per database where every node corresponds to an attribute and multiple attributes are connected by an hyperedge if they coexist in a constraint. Additionally, the hyperedges are labeled based on their corresponding type of constraint.
- **Matching** the hypergraphs using an existing algorithm, such as HNN-HM [3], or development of a new solution.

We start with a simple scenario, where the schemas to be compared consist of the exact same set of tables and attributes and we thus focus on the detection of one-to-one correspondences between attributes. However, even such a scenario can be quite challenging if the attribute names are cryptic (e.g., "A1" or "xyz") or missing. Moreover, if the values of the matching attributes are encoded using different units of measurement (e.g., metric vs. imperial system) or vocabularies (e.g., {'bachelor', 'master', 'phd'} vs. {0, 1, 2}), they cannot be directly compared using traditional instance-based schema matchers.
Depending on the project's progress, we can extend the scope step by step. Potential extensions include:

- Integration of similarities between attributes from **different schemas**, calculated using traditional schema matching approaches.
- Integration of constraints across different schemas (e.g., inclusion dependencies).
- Detection of one-to-many and **many-to-many** correspondences (e.g., by merging nodes within the hypergraph).
- Consideration of **approximate** constraints (i.e., constraints that are not always valid).

## Experience in the following is beneficial:

- Data profiling and schema matching
- Proficiency in Java or Python

## Contact Details

This project will be supervised by Dr. Fabian Panse, Dr. Matteo Paganelli, and Prof. Dr. Felix Naumann at the Information Systems chair. If you have any questions, please do not hesitate to contact us.

## References

[1] E.Rahm, PA.Bernstein: *A Survey of Approaches to Automatic Schema Matching*, VLDB Journal 10(4), 2001.

[2] Z.Abedjan, L.Golab, F.Naumann: *Profiling Relational Data: A Survey*, VLDB Journal 24(4), 2015.

[3] X.Liao, Y.Xu, H.Ling: *Hypergraph Neural Networks for Hypergraph Matching*, International Conference on Computer Vision (ICCV), 2021.