Exercise - Data

Exercises in Data Modeling (190.021)

[1]*dm@ai-lab.science* February 8, 2026

This is a cheat sheet describing the key concepts of this exercise on data storing techniques.

1 General Information

Note that this is not a full explanation of all topics that are covered in the chapter "Data". It mainly provides tips for the exercise and further explanations of the covered concepts.

2 Table-based Data Storage

To understand the need for data storage techniques, we will use a CSV table as a starting point. This table contains all the information about a university and is centered around students who registered for lectures. Every student is enrolled in different lectures. For each Student-Lecture relationship, there is an entry in the table. Further, the table includes information about the enrolled majors, teaching faculties, and departments.

In general, there is a lot of redundant and repeated information included in this table. To get rid of these repetitions, the included information will be grouped. Within the exercises, we use entity-relationship diagrams (ERDs) to optimize the data structure.

3 Entity-Relationship Diagram (ERD)

An ERD can be used to model the logical structure and content of data before dividing the data into tables. There are three different components:

- Entity: Uniquely identified real-world objects and concepts.
- Attribute: Properties of entities.
- Relationships: Connections between the entities.

3.1 Cardinality

The cardinality defines how many instances of one entity can be associated with another. There are three different cardinalities that we differentiate:

- one-to-one
- many-to-one
- many-to-many

3.2 Primary key

A primary key is an attribute or a combination of attributes that uniquely identifies each instance of an entity. Each entity needs to have exactly one primary key, having a value that occurs exactly once and does not change over time. This ensures that each record can be referenced and allows entities to be linked via relationships. In practice, these primary keys often appear as IDs, such as a matriculation number or a Lecture ID.

4 Split Tables into Smaller Tables and Store in a Database

After identifying all the downsides of storing the data in one big data and already dividing the data into logical subsections, the data will be split into smaller tables. Many-to-one relationships are directly included within the smaller tables, while

many-to-many relationships deserve separate tables.

It primarily reduces redundancy, improves data consistency, and simplifies updates. After splitting the tables, we store them in a relational database, and the relationships between the tables are maintained using primary and foreign keys. Databases allow for efficient storing, querying, and management of structured data and are often used for large and often updated datasets.

5 Memory Requirements

As the last step of the notebook, we want to measure the effect of the division and storage change of the data. By splitting the data into multiple tables and getting rid of the redundancies and repetitions, the overall memory consumption can be reduced. A well-designed data structure improves clarity and maintainability, and also reduces storage requirements.