Univ.-Prof. Dr.-Ing. Matthias Boehm
Graz University of Technology
Computer Science and Biomedical Engineering
Institute of Interactive Systems and Data Science
BMK endowed chair for Data Management

1 Data Management WS21/22: Exercise 01 – Data Modeling

Published: October 08, 2021 (last update: Oct 08)
Deadline: November 02, 2021, 11.59pm

This exercise on data modeling aims to provide practical experience in Entity-Relationship (ER) modeling, ER-relational mapping, and relational normalization. The expected result is a zip archive named `DBExercise01_<student_ID>.zip`, containing the partial results of the individual sub-tasks, submitted in TeachCenter.

1.1 ER Modeling (15/25 points)

Create an ER diagram in Modified Chen (MC) notation—including entity types, relationship types, attributes, cardinalities, and keys (create surrogate keys if natural unique identifiers are missing)—for managing the Austrian National Elections (1920-2019)[1]. It is up to you if you use existing tools for data modeling, or draw this via presentation software or hand. There are multiple correct ways of modeling (alternatives, and level of details), however the diagram should capture the essential information of the following discourse:

- The Austrian National Council is elected every 5 years (previously 4 years), or earlier if needed. A single *election* is described by a unique short name (e.g., NRW 2019), a unique sequence number (e.g., 27 for NRW 2019), and an election date.
- A *person* can be a voter, a candidate, or both. Each person is described by a unique person identifier (PID), a first name, a last name, a date of birth, a gender (female, male, diverse), and exactly one living *location*. A location is in turn described by a street name and number, a postal code, a city, and a country.
- Multiple political *parties* compete in the elections. Each party has a short name (e.g., ÖVP), a long name (e.g., Österreichische Volkspartei), and a head quarters location (e.g., Lichtenfelsgasse 7, 1010 Vienna). Each party nominates a ranked list of candidates (i.e., persons) for each election (e.g., ÖVP top-4 at NRW 2019: 1 Kurz, 2 Köstinger, 3 Blümel, 4 Schramböck). A person cannot be a candidate for multiple parties at a single election.
- Persons can vote at most once at a specific election—either in person (in an assigned polling place) or via ballot-by-mail—and are registered accordingly. Both polling places and ballot-by-mail belong to a hierarchy of *electoral authorities* (each with a name, and location), that count and aggregate votes per election and party.

Partial Result: `ERDiagram.pdf`

[1] Dataset link: `https://github.com/tugraz-isds/datasets/tree/master/elections_at/raw`. A cleaned version of the dataset will be made available for Exercise 2 after additional data preparation and cleaning.

1.2 Mapping ER Diagrams into the Relational Model (10/25 points)

Create a relational schema for the ER diagram designed in Task 1.1 and bring it into third normal form, assuming the following additional functional dependencies (besides dependencies from unique attributes to other attributes):

$$\text{Location.City} \rightarrow \text{Location.Country}$$
$$\text{Party.ShortName} \rightarrow \text{Party.LongName}$$

This schema should include the relations and typed attributes, as well as all primary and foreign keys. Furthermore, the schema should also ensure that no person can vote multiple times at a single election (e.g., via in-person and mail-by-ballot voting). It is up to you if you provide either a SQL script (`CreateSchema.sql`) with `CREATE TABLE` statements, or provide a text schema (`Schema.txt`) in the following text notation (one line per table):

```
<Table>(<Attribute 1>:<type>(PK), <Attribute 2>:<type>, ..., <Attribute n>(FK))
```

Here, PK and FK indicate primary and foreign keys, where multiple attributes with (PK) represent a composite primary key. If an attribute is both foreign key and (part of) a primary key, use `<Attribute m>(PK,FK)`. Please, adhere to this notation with unchanged parentheses and delimiters, and limit yourself to common data types (i.e., `int`, `numeric(p,s)`, `char(n)`, `varchar(n)`, or `date`) because this sub-task is subject to automated grading.

Partial Result: `Schema.txt`, or `CreateSchema.sql`