# Challenge #3
# Anonymize Data

## Anchoring and overview

| | | |
|---|---|---|
| Modul: | 683 | Datenbestände analysieren und interpretieren |
| Handlungsziele: | HZ 2 | Maskiert, pseudonymisiert oder anonymisiert sensitive Rohdaten bei Bedarf und unter Berücksichtigung des Datenschutzes. |
| Leistungskriterien PO: | LK-B-7 | Datenbestände inhaltlich zu analysieren und/oder zu vergleichen und die gewonnenen Informationen zu verdichten und darzustellen. |
| | [LK-B-6] | [Funktionen mittels Skriptsprachen für die Auswertung von Daten zu programmieren] |

# Formulation of task (candidate view)

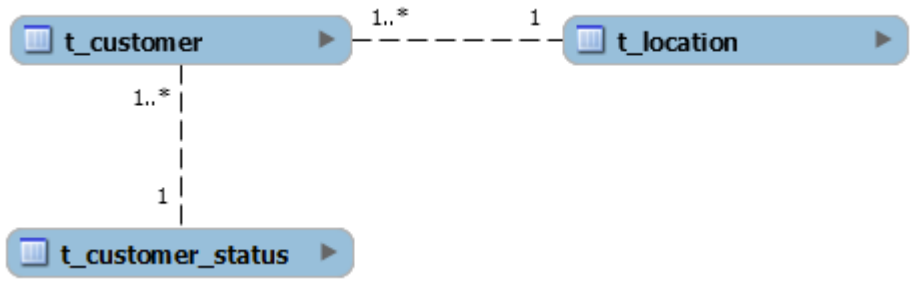
# Anonymize Data

## Resources

[1] Database file with data to be anonymized
`c3-database-initial.db` - MD5-Checksum: xxx

## Introduction

The figure shows the SQLite database model of an application with 3 tables and 2 relationships:




Data must be anonymized before the data can be passed on to a third-party provider for statistical analysis. Before you start, verify the integrity of the downloaded database file in the resource section by calculating and comparing the MD5 checksum.

## Goal and tasks

Create a CSV file with the anonymized data of all customers from the database. The CSV file must contain the following columns as header and the processed data in its rows.

| **Column** | **Customer data** |
|---|---|
| gender | Gender from table `t_customer`. |
| lastname | Masked last name form table `t_customer`. Keep the first 2 characters of the last name and replace any further characters by exactly 8 hyphens (-). Example: "Miller" is changed to "`Mi--------`" |
| birthyear | Birthdate from table `t_customer` reduced to the year only. Example: 01.01.1999 is changed to 1999 |
| zip | Related zip code from table `t_location` with reduced precision. Reduce precision by keeping the first 2 digits of the zip and replacing the last 2 digits with zeros. Example: 4934 is changed to 4900 |
| status | Related status from table `t_customer_status`. |

## Submission

Submit a ZIP archive containing the CSV file with the anonymized data and a written report as PDF document with a brief technical description of your approach (including information about methods, tools, commands, scripts etc. used).

# Specifications for correction

## Evaluation criteria

| | | |
|---|---|---|
| C1: | Written report delivered (regardless of the correctness) | = 1 pt. |
| C2: | Descriptions in report are understandable and comprehensible | = 1 pt. |
| C3: | The candidate's approach is simple and efficient (see instructions below) | = 1 pt. |
| C4: | CSV file with the 5 required and correctly labeled columns delivered (regardless of the correctness and completeness of the data) | = 2 pt. |
| C5: | The CSV file contains unchanged data on gender for all 2000 customers | = 2 pt. |
| C6: | The CSV file contains correctly masked last names for all 2000 customers | = 2 pt. |
| C7: | The CSV file contains correctly reduced birth years for all 2000 customers | = 2 pt. |
| C8: | The CSV file contains correctly processed zip codes for all 2000 customers | = 2 pt. |
| C9: | The CSV file contains unchanged status for all 2000 customers | = 2 pt. |

## Correction instructions

- This challenge gives a maximum of **15 points**.
- Given scores on the criteria may not be further subdivided.
- The candidate's solution may differ from the sample solution in terms of approach (method and / or tool).
- C3: Approaches with SQL manipulations in a GUI tool or a script (e.g. Python) are considered as simple and efficient. Processing 2000 datasets with formulas or macros "semi-manually" in a spreadsheet is considered as inefficient and error prone.

## Sample solution

No specific approach or tool is specified to solve this task. However, a solution with SQL manipulations seems the simplest and most efficient approach. SQLite databases can easily be manipulated with tools such as DB Browser for SQLite or SQLiteStudio. These tools also provide the functionality to export data as CSV file (or other formats).

```
-- creating new table t_statistics

CREATE TABLE "t_statistics" (
  "gender"    TEXT,
  "lastname"  TEXT,
  "birthyear" TEXT,
  "zip"       TEXT,
  "status"    TEXT
);

-- copy initial values into table t_statistics

-- hint: joining foreign keys is required due to relationsships

INSERT INTO t_statistics(gender, lastname, birthyear, zip, status)
SELECT tc.gender, tc.lastname, tc.birthdate, tl.zip, ts.status
FROM t_customer tc, t_location tl, t_customer_status ts
WHERE tc.fk_location = tl.pk_location
AND tc.fk_customer_status = ts.pk_customer_status

-- anonymize last name

UPDATE t_statistics SET lastname = substr(lastname, 1, 2) || '--------';

-- reduce birthdate to year only

UPDATE t_statistics SET birthyear = substr(birthyear, 7, 4);

-- reduce birthdate to year only

UPDATE t_statistics SET zip = substr(zip, 1, 2) || '00';

-- export t_statistics to CSV in the selected tool
```