LINKS. A System for Historical Family Reconstruction in the Netherlands

By Kees Mandemakers, Gerrit Bloothooft, Fons Laan, Joe Raad, Rick J. Mourits and Richard L. Zijdeman

To cite this article: Mandemakers, K., Bloothooft, G., Laan, F., Raad, J., Mourits, R. J., & Zijdeman, R. L. (2023). LINKS. A System for Historical Family Reconstruction in the Netherlands. *Historical Life Course Studies, 13*, 148–185. https://doi.org/10.51964/hlcs14685

HISTORICAL LIFE COURSE STUDIES

Content, Design and Structure of Major Databases with Historical Longitudinal Population Data

VOLUME 13, SPECIAL ISSUE 5

GUEST EDITORS
George Alter
Kees Mandemakers
Hélène Vézina

EHPS NETWORK

MISSION STATEMENT

HISTORICAL LIFE COURSE STUDIES

Historical Life Course Studies is the electronic journal of the *European Historical Population Samples Network* (EHPS-Net). The journal is the primary publishing outlet for research involved in the conversion of existing European and non-European large historical demographic databases into a common format, the Intermediate Data Structure, and for studies based on these databases. The journal publishes both methodological and substantive research articles.

Methodological Articles
This section includes methodological articles that describe all forms of data handling involving large historical databases, including extensive descriptions of new or existing databases, syntax, algorithms and extraction programs. Authors are encouraged to share their syntaxes, applications and other forms of software presented in their article, if pertinent, on the EHPS-Net website.

Research articles
This section includes substantive articles reporting the results of comparative longitudinal studies that are demographic and historical in nature, and that are based on micro-data from large historical databases.

Historical Life Course Studies is a no-fee double-blind, peer-reviewed open-access journal supported by the European Science Foundation (ESF, http://www.esf.org), the Scientific Research Network of Historical Demography (FWO Flanders, http://www.historicaldemography.be) and the International Institute of Social History Amsterdam (IISH, http://socialhistory.org/). Manuscripts are reviewed by the editors, members of the editorial and scientific boards, and by external reviewers. All journal content is freely available on the internet at hlcs.nl.

hislives@kuleuven.be

The European Science Foundation (ESF) provides a platform for its Member Organisations to advance science and explore new directions for research at the European level. Established in 1974 as an independent non-governmental organisation, the ESF currently serves 78 Member Organisations across 30 countries. EHPS-Net is an ESF Research Networking Programme.

The European Historical Population Samples Network (EHPS-net) brings together scholars to create a common format for databases containing non-aggregated information on persons, families and households. The aim is to form an integrated and joint interface between many European and non-European databases to stimulate comparative research on the micro-level.
Visit: http://www.ehps-net.eu.

HISTORICAL LIFE COURSE STUDIES
VOLUME 13 (2023), published 01-06-2023

LINKS

A System for Historical Family Reconstruction in the Netherlands

Kees Mandemakers	International Institute of Social History, Amsterdam & Erasmus University Rotterdam
Gerrit Bloothooft	Utrecht University & Meertens Institute, Amsterdam
Fons Laan	International Institute of Social History, Amsterdam
Joe Raad	LISN, CNRS (UMR 9015), University of Paris-Saclay
Rick J. Mourits	International Institute of Social History, Amsterdam
Richard L. Zijdeman	International Institute of Social History, Amsterdam & University of Stirling

ABSTRACT

LINKS stands for 'LINKing System for historical family reconstruction' and is a software system to link nominal data from the Dutch archives and ultimately reconstruct historical individuals and families. We present the background and philosophy of this matching system and explain its data structure and functioning. Currently the core data of the LINKS system consists of indexed civil certificates. These certificates are available from 1812 — the start of the Dutch Vital Registration — until the year they are confidential based on privacy laws. For more than 20 years, thousands of volunteers have been working to build this index, which contains not only the names of newborn, married and deceased persons, but also the names of their parents, places of birth, ages and sometimes their occupational titles. The software system LINKS includes the standardization of all input before linking, nominal record linkage procedures and identification of all unique persons involved in the system. All processes are repeatable and a strict distinction is maintained between source data, standardized, linked and enriched data and released data. Moreover, LINKS also informs archives about all kinds of errors and inconsistencies found during the cleaning and matching process. We will discuss two matching systems, the first is the original querying system that runs within a MySQL database environment and the second is a newly developed system, called burgerLinker, which is based on knowledge graphs and which is designed as a system that can be used independently from LINKS and is made available as open source software. Finally, we present the most important releases of LINKS data so far: two national releases that link birth and parental marriage certificates, creating families and pedigrees and an integrated dataset of persons, families and family trees in four provinces.

Keywords: Nominal record linkage, Historical population data, Civil certificates, Historical demography, Family reconstitution, Genealogical data

e-ISSN: 2352-6343
DOI article: https://doi.org/10.51964/hlcs14685

1 INTRODUCTION

LINKS stands for 'LINKing System for historical family reconstruction' and is a software system to link nominal data from the Dutch archives and ultimately reconstruct historical individuals and families. Such a reconstruction is indispensable for all scientific work dealing with people in the past, and also facilitates the work of genealogists enormously. Given privacy constraints, the reconstruction of the Dutch family network population is possible until about 100 years ago, while it could go back in time as far as the 17th century, depending on locally available sources. In this paper, we present the background and philosophy of this rule-based matching system and explain its data structure and functioning.

Currently the core data of the LINKS system consists of indexed civil certificates. These certificates are available from 1812 — the start of the Dutch Vital Registration — until the year they are confidential based on privacy laws. This limitation depends on the type of the certificate and is respectively 100, 75, and 50 years for birth, marriage, and death certificates.[1] For more than 20 years, thousands of volunteers have been working to build this index, which contains not only the names of newborn, married and deceased persons, but also the names of their parents, places of birth, ages and sometimes their occupational titles. In 2022 the index contained over 125 million person names and is continuously growing not only because new civil certificates become public each year but also because existing gaps are filled. All digitized data are publicly accessible through WieWasWie ('WhoWasWho', see https://www.wiewaswie.nl), based at the CBG Center for Family History (https://www.cbg.nl).

When designing LINKS three requirements were formulated to ensure both a successful reconstruction of historical persons and dissemination of releases with linked data: a) standardization of all input before linking, b) development of nominal record linkage procedures and c) identification of all unique persons involved in the system. All processes are repeatable and the database maintains a strict distinction between A) source data, B) standardized, linked and enriched data and C) released data (Mandemakers & Dillon, 2004). Moreover, LINKS also informs archives about all kinds of errors and inconsistencies found during the cleaning and matching process.

LINKS is based at the International Institute of Social History (IISG) as part of the HSN databases (HSNDB). Beginning in 2006, releases were disseminated from the indices of the civil certificates, mainly matched marriage records. These releases were initially known as the GENLIAS datasets, before the name LINKS was adopted. LINKS started in 2010 as a spin-off of the Historical Sample of the Netherlands (HSN) (Mandemakers & Kok, 2020), financed by the NWO CATCH program.[2] The project was a cooperation between the IISG, Utrecht University, the Meertens Institute and the Leiden Institute of Advanced Computing. For more information about the LINKS project, see https://iisg.amsterdam/en/hsn/projects/links. Nowadays, LINKS is part of HSNDB, the IISG system of databases for historical and contemporary research (see https://iisg.amsterdam/en/hsndb).

In the next three sections of this paper we explain the workflow and processes of the LINKS system; in the last two sections we describe the construction of the major releases. Section 2 concentrates on the sources that form the basis of LINKS. Section 3 focuses on the cleaning of the imported data from WieWasWie and on the feedback given to the archives that provide the WieWasWie data. In Section 4 we discuss two matching systems, the first is the original querying system that runs within a MySQL database environment and the second is a newly developed system, called burgerLinker, which is based on knowledge graphs. BurgerLinker is designed as a system that can be used independently from the LINKS system and made available as open source software, so that it can be used freely for all kinds of nominal data.[3] In Section 5 we evaluate the outcomes of different matching strategies applied on the Zeeland marriage certificates. In Section 6 we present the most important releases of LINKS data so far: two national releases that link birth and parental marriage certificates, creating families and

1 Burgerlijk Wetboek ('Dutch civil code'), article 1:17A. Retrieved 5 January 2023 from https://wetten.overheid.nl/BWBR0002656/

2 CATCH stands for Continuous Access to Cultural Heritage and is a program of the Dutch Research Council (NWO). In this program researchers and heritage managers worked together to make heritage data more accessible and develop instruments to enable heritage managers to work more efficiently. LINKS was one the 12 projects that were granted. The programme started in 2004 and ran till 2014, for more information see https://wwwnwo.nl/en/researchprogrammes/continuous-access-cultural-heritage-catch

3 See https://www.github.com/clariah/burgerlinker

pedigrees and an integrated dataset of persons, families and family trees in four provinces. The paper ends with a summary and conclusion.

2 DATA FROM THE DUTCH CIVIL REGISTERS AND THE LINKS WORKFLOW

First attempts with record linkage in historical demography were done with data from church records and civil registers. Louis Henry is the well-known founder of a methodologically grounded way of linking this kind of records. Together with Michel Fleury he developed a form to create and record family reconstitutions (Henry & Fleury, 1956; Séguy, 2016). The first datasets of this kind were limited to the parish area. Examples are the reconstitution of 34,812 families in 39 French parishes from the period 1640–1829 (Séguy, 2001) and the database constructed by the Cambridge Group for the History of Population and Social Structure for 26 parishes in England and Wales over de period 1580–1837 (Wrigley, Davies, Oeppen, & Schofield, 1997). With the growth of computing power and expertise within the field, historical reconstructions are now available for a myriad of countries and the scope is only increasing (for an overview, see Mandemakers, 2023; Song & Campbell, 2017). After the parish level whole nations came into view. In Québec, projects started with the aim to reconstruct the whole population from 1621 onwards (Dillon et al., 2018; Nault & Desjardins, 1989; Vézina & Bournival, 2020). In France, Dupâquier and Kessler (1992) collected a sample of 40,000 marriage certificates from all over France based on the letter combination TRA and linked them into pedigrees. Subsequently other researchers added other data to this basic construction such as birth and death certificates, military registers and data from hereditary tax and military registers (Bourdieu, Kesztenbaum, Postel-Vinay, & Tovey, 2014). Based on the Dutch civil records, LINKS is a continuation along the path set out especially by these French historical demographers.

Civil registration was introduced in the Netherlands in 1810, as a consequence of the annexation by the French Empire. The Code Napoléon provided for the compulsory, standardized recording of vital events in certificates. The certificates had to be drawn up in the municipality where the vital event occurred. Most Dutch municipalities introduced civil registration over the course of 1811. However, since the Dutch province of Limburg and the south of Zeeland (Zeeuws-Vlaanderen) were annexed by France in 1796, civil registration for these provinces was introduced in that year (Vulsma, 1988).

All certificates of birth, marriage, or death ever made in the Netherlands are still available, as each certificate was made in duplicate and stored in books for safekeeping. At the end of each year, one civil registry book remained in the municipality and the other was sent to the provincial courts. The registrars had to note the name, age, occupation and municipality of residence of the informants and witnesses. This information assured the correct identification of these individuals. In Dutch birth certificates we find the names, address, ages and occupations of the parents in addition to data on the newborn. Death certificates provide last residence, age and final occupation of the deceased and data on the spouse(s) and parents, including occupational titles if they were still alive. The information concerning the parents was officially less detailed, as age was not required, but nevertheless its registration was widespread. The marriage certificates give information on the occupations, illiteracy (absence of signature) and places of residence of the bride, the groom, their parents and the (usually four) witnesses, who were relatives or friends of the marrying couple about half the time (Mandemakers, 2000; Vulsma, 1988; for an exhaustive list of all information found in the certificates, see Mourits, van Dijk and Mandemakers (2020), p. 44, table 1). Data from civil certificates are non-dynamic, which implies that they are only valid at the date of the events of birth, marriage and death. This is fundamentally different from sources that offer a more continuous stream of data like the population register which is used by the Historical Sample of the Netherlands (HSN). For a systematic comparison of the value and use of both sources, see van den Berg, van Dijk, Mourits, Slagboom, Janssens and Mandemakers (2021).

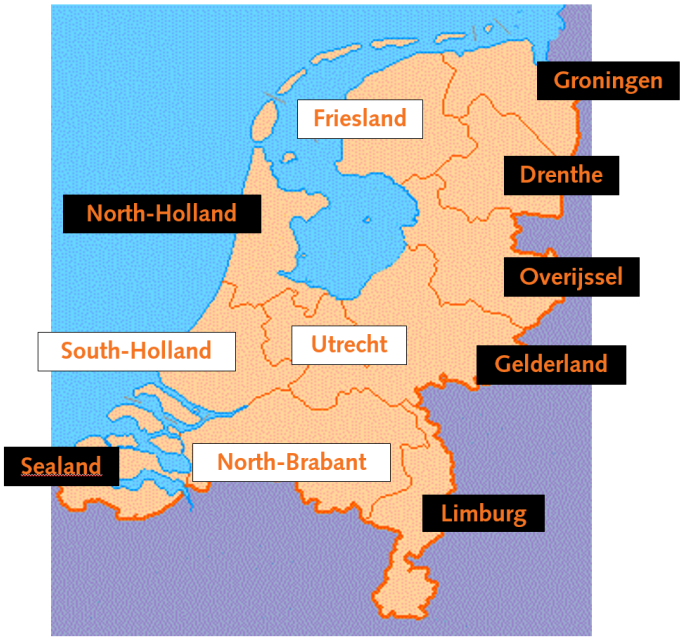
Since the early nineties of the previous century, hundreds of volunteers have been working on the indexation of all names and dates from these certificates. This indexing was embedded in the already existing practice in which local and regional archives organized volunteers to help disclosing archival collections. Identifying persons and buildings from old photographs and films was a favorite exercise. It was probably the city archive of Amsterdam that started the first volunteering project with population data by starting the data entry of the population register 1850–1853. Soon after, an initiative by the National Archive called GENLIAS started to index all marriage certificates in a consistent way. Around

1995, provincial archives were firmly encouraged to cooperate and recruit the volunteers required for indexing. In the beginning indexing was done using the original sources, but from the year 2005 onwards data entry from scans became the norm and later data entry also became web based. Volunteers could work at home, which extended enormously the base of volunteers. The organization also professionalized in the sense that data entry was done more and more by private companies offering data entry programs embedded within platforms. This also created a kind of community to advise the volunteers on problematic issues and to monitor the progress of a specific job. Important companies are Vele Handen ('Many Hands') and Het Volk ('The Crowd') that presently organize about 35 different projects, ranging from the indexing of notarial deeds, population registers to indexing photographs.[4]

A new platform called WieWasWie was constituted to present the information from the Dutch local and regional archives and deal with the technical challenges in the presentation and search possibilities of the indexed data and linked scans. WieWasWie is maintained by the Dutch Family Center and offers a central point for all archives to present their indexes and to make searches for persons on a national scale possible (https://www.wiewaswie.nl/en/). Besides names and dates, indexed information differs between archives, as WieWasWie is designed as a decentralized system. The participating archives can make their own decisions as to which data are entered into the index, but always included are the type, date and municipality of the event as well as the first names and family names of the persons involved (child/parents, bride/groom/parents or deceased/parents/partner) and usually age at the event for the deceased, bride and groom. Witnesses are seldom included. Occupational titles were systematically entered for the marriage certificates in seven out of eleven provinces, see Figure 1, adding up to about 60% of all certificates. This percentage is much lower in the case of death and especially birth certificates.

For privacy reasons, certificates are made public with a delay of 100 years (birth certificates), 75 years (marriage certificates) or 50 years (death certificates), so in 2018 certificates were available until 1918, 1943 and 1968, respectively. In practice, the delay is up to 5 years longer as most archives do not update their indexes annually. Table 1 presents the level of indexation in September 2018. It shows that the marriage certificates are almost completed. Lagging behind are the birth and to a smaller degree, the death certificates. Currently about 85% of the indexing has been done. In all, 27 million civil certificates were digitized, containing information on about 120 million person mentions. However, archives are quickly catching up, and we expect countrywide coverage in about 5 years.

Figure 1 *Provinces of the Netherlands between 1811 and 1940*



Explanation: Provinces for which occupational titles are available have a black frame; in the province of South-Holland the cities The Hague and Leiden are a positive exception since they also included the occupational titles in the index. In the case of brides and parents one often finds the term "without occupation". Occupations of deceased parents are not mentioned at all.

4 Another article in this special issue concerns the slavery registers of Suriname of which the data entry was also done by volunteers (van Galen, 2019; van Galen et al., forthcoming).

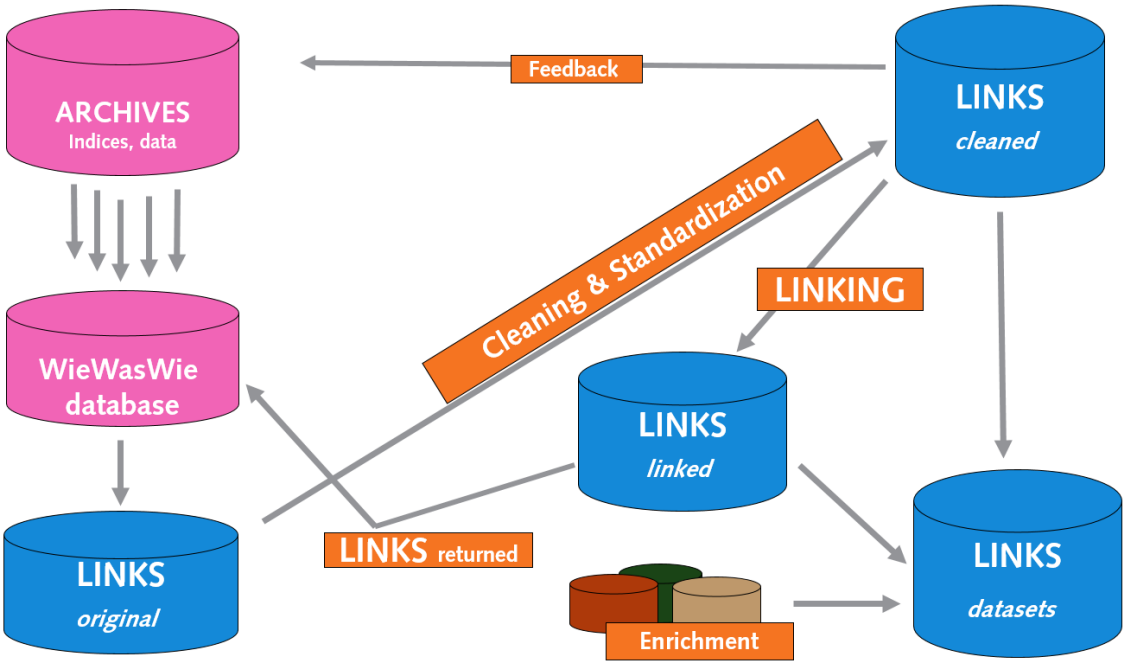
Table 1 *Number of the publicly available and indexed civil certificates (in millions), the Netherlands, September 2018, WieWasWie*

	Indexed	Public	% Indexed
Birth certificates	10.4	14.2	73.2
Death certificates	12.1	13.2	91.7
Marriage certificates	4.5	4.8	93.8
Total	27.0	32.2	83.9

Sources: The numbers of total events before 1850 were provided by van der Bie and Smits (2000), except the marriages 1812–1839 which were estimated; the number for the period from 1850 onwards were provided by the Historical Database of Dutch Municipalities (HDNG; Mourits, Boonstra, Knippenberg, Hofstee, & Zijdeman, 2016). The number of indexed certificates was calculated from the data that were gathered by LINKS in September 2018.

The LINKS system is designed in a generic way and can handle nominal data from all kinds of sources. But in the development phase, we used data from the civil registration available in WieWasWie to create datasets with linked certificates in an enriched way and made them available to the scientific community. We also created reconstructions of life courses and families for separate regions in the Netherlands. In Figure 2 the general outline of the LINKS workflow process is sketched. Data are delivered by the regional and city archives to the WieWasWie-system. The ownership and responsibility for the content rest with these archives. LINKS harvests the data from WieWasWie in the LINKS *original* database as soon as new releases of data are added to WieWasWie. From LINKS *original* the data are cleaned and standardized and subsequently added to the LINKS *cleaned* database. We give feedback to the WWW community on the errors and problems we found. These cleaned data form the basis of the different matching procedures. The resulting links are stored in LINKS *linked*, while the links between the certificates are also returned to the Dutch Family Center to be published on the website of WieWasWie. In a further stage the data from LINKS *cleaned* and LINKS *linked* are combined into LINKS datasets with, for instance, pedigrees and families, and enriched with geographical data and occupational coding. These data sets are released for scientific research.

Figure 2 *Workflow of the LINKS system*



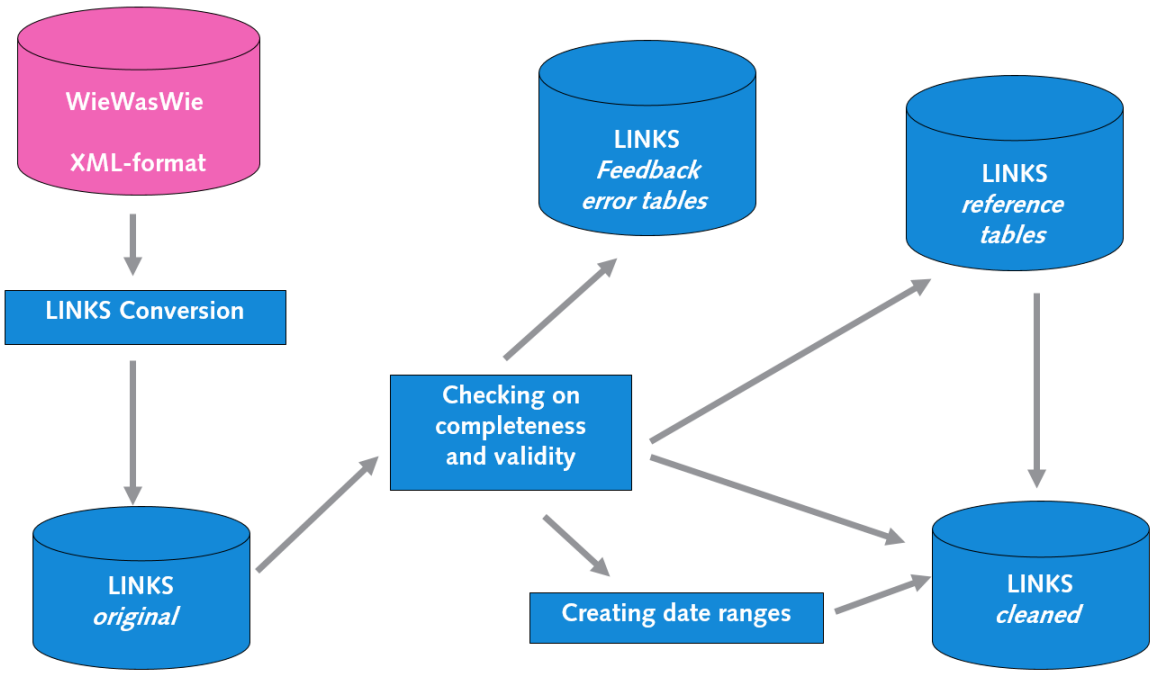
3 PREPARING THE DATA FOR MATCHING

3.1 CONVERTING, CLEANING AND STANDARDIZATION

The data from the WieWasWie database is made available in XML format. The LINKS system consists of several MySQL databases based on two tables: one table for information on the registration of events, and one table with information on all persons involved in the events.

Figure 3 focuses on the part of the workflow that processes the data from the WieWasWie dataset to the LINKS *cleaned* database. The first step is to import the data from XML into MySQL tables, and to distinguish the different types of events and define the corresponding roles of the individuals involved. This is a source-dependent operation put in place for every document type, currently birth, marriage and death certificates, but extensions of these scripts are easy to provide. In this conversion the character type is also converted to UTF-8 and all diacritics are changed into basic characters (e.g., é, ë, and è become e). The result is stored in the database LINKS *original*.

Figure 3 *LINKS workflow from original to cleaned data*



Once the data is stored in the LINKS *original* database, a cleaning process is started which results in the LINKS *cleaned* database. The cleaning process checks all data on completeness and validity. Completeness concerns the presence of compulsory information, such as including all roles belonging to a type of registration. For example, a marriage certificate should include at least a bride and a groom and not include roles exclusively belonging to a birth or death certificate. Other checks test on duplicated roles in a certificate, or whether each person has a first name and a family name which is essential for nominal linking. Validity concerns the consistency and range of dates; all dates should be in a range that is in agreement with the properties of the source type. For example, the date of a birth certificate cannot be earlier than the date of the birth itself, or a groom or bride should be at least 14 years younger than their parents. Incomplete, inconsistent or invalid data are reported in a systematic way (see Appendix A for an overview of error messages).

Data cleaning also involves standardization, for example avoiding variation that is not essential to the meaning of the information. This may concern spelling variation in place names, occupational titles, variation in the writing of dates, the writing of abbreviations in full, and so on. In some cases the spelling of first names or family names is also standardized. Standardization procedures in LINKS apply so-called reference tables. The most important ones are those for family names, first names, ages, locations, occupational titles, sex, and civil status. For every type of information, the system verifies whether the content is already present in the relevant reference table. When the original value is already included, a code is assigned describing the validity of the entry under three possible values:

One for "valid and standardized", one for "not valid, but clear enough to be standardized" and "not valid and not standardized." Corresponding standard values are written to LINKS *cleaned.* If the content is not known in the reference table, it will be written in LINKS *cleaned* and a new record will be made in the reference table where it awaits standardization. After a new round of standardization, cleaning is run again on every field. If the value is considered invalid, it is included in the error table and reported to the originating archive (see Appendix A for these messages).

Reference tables are part of the existing standardization schemes in the HSNDB domain. With each new release, all new values in the LINKS database are standardized and coded in a manual and/or semi-automatic way and added to the reference table after expert review. Table 2 gives an overview of the content of the main reference tables as of July 2021. The first column gives the number of original values, the second column the number of standardized values, the third column the number waiting to be standardized, and the fourth column the values that were considered invalid. The fifth column presents the resulting unique values. These tables are kept in one system together with other databases that form the HSNDB environment. Additionally, there are smaller reference files for data such as suffixes, aliases, sources, roles and source types. Civil status quite often also implies an indication for the sex of a person (bride, widower etc.), hence a combined table Status_sex was developed. The table Prefixes includes text that may precede a family name as part of the family name (for example "de Boer" instead of only "Boer", which is quite common in the Netherlands), or as a title. Both possibilities are separately standardized.

Table 2 *LINKS reference tables with number of original and standard values, 1–7–2021*

	Originals	Standardized	Not yet standardized	Not valid	Unique standards
Family names	846,860	797,519	49,217	124	508,877
First names	300,299	263,146	36,991	162	238,205
Prepieces (titles and prefixes)	6,264	2,201	1,140	2,923	344
Ages (days/weeks/ months/years)	75,318	60,151	0	15,167	2,576
Locations	566,068	150,182	412,148	3,738	8,736
Occupations	319,215	276,665	37,860	4,690	83,610
Status_sex (combined table)	707	674	18	15	23
Religion	2,630	2,629	0	1	117

Based on knowledge of name spelling and after experimental matching with marriage certificates of Zeeland (see Section 5.2), we decided to apply a simple initial standardization for both first and family names by replacing all "ch" by "g", "c" by "k", "ph" by "f", "z" by "s", and "ij" by "y". Furthermore, family names or first names occurring in the whole dataset less than three times were considered to be spelling errors and were standardized to the most frequent corresponding name with a Levenshtein distance of 1. We considered them by definition as spelling mistakes, for example "Gerrjt" with frequency 2 was standardized to "Gerrit". A family name as "Bakkkr" with a frequency less than two was standardized as "Bakker". Through these two operations the number of unique standard family names dropped by about 35%, and the number of unique standard first names by about 10% (see Table 2). Since adding unique names will result in more pairs being compared in an exponential way, limiting the number of spelling variants of names is quite important in reducing processing time.

The standardization of other variables is also processed in a semi-automatic way. For ages, the standard is a combination of four values for days, weeks, months and years. Currently, there is a high number of non-accepted values, as some archives entered dates of birth in the age fields rather than the age mentioned on the certificate. Almost 320,000 different original occupational titles are about 80% standardized now, resulting in about 83,000 unique standards. New versions of reference tables are released periodically, see for example Mandemakers et al. (2020) with the latest release of occupational titles (n=281,355) including standard values and corresponding HISCO, HISCAM and HISCLASS classifications (Lambert, Zijdeman, van Leeuwen, Maas, & Prandy, 2013; van Leeuwen & Maas,

2011; van Leeuwen, Maas, & Miles, 2002). Locations are standardized with respect to municipality, province/region and nation and enriched with geo-referential codes (Huijsmans, 2020). The region is a more general category used to cover regional levels above the municipality level, especially outside the Netherlands for example states in the USA, islands in the East-Indies, etc.

The output of the LINKS cleaning system will lead to feedback to the partners of WieWasWie. This may involve general suggestions to improve the quality and uniformity of the data, but also offering specific instructions to verify the data in certain certificates. Original entries which are not accepted as valid data are included in the error archive and reported to the relevant archive. In Appendix A we give an overview of all types of errors reported by LINKS and sent to the archives. Presently there are 114 different types, which resulted in a total of over 3 million messages so far (of which 1.9 million for the death certificates, for about 900,000 because of an inadequate age and 400,000 because of lacking or insufficient firstnames).

3.2 TIME RANGES FOR BIRTH, MARRIAGE AND DEATH

For all persons in each certificate, the time range in which they likely were born, married or died was calculated. These ranges limit the number of false positives and are also used to decrease processing time. The estimation of the minimum and maximum years of the range is based on six features:

1 The type of a certificate;
2 The role of a person in a certificate;
3 The age of a person at the event (if known);
4 Whether a person is alive or not at the event (if known);
5 The age of a related person in the certificate;
6 Preset ranges for certain life events.

The last feature is operationalized by the following rules:

1 A woman will give birth to children at an age between 14 and 50 years;
2 A man will father children at an age between 14 and 100 years;
3 Children are born in a legitimate state, i.e. parents are married at child birth;
4 Persons will not become older than 110 years of age;
5 Difference in age between partners will be maximally 66 years;
6 Maximal age of marriage for a woman is 90 years;
7 Maximal age of marriage for a man is 100 years.

For example, consider the mother of the groom in a marriage certificate from 1888. We know from the certificate that the groom is 25 years old. That implies that given a fertile period of the mother of 14–49 years she cannot be older than 25+50=75 years and not younger than 25+14=39 years, so the mother should be born between 1813 and 1849. We can also calculate the range of her year of marriage which is between 1813+14=1827 and 1849+14=1863, which should also be the range of the year of marriage of her husband, the father of the groom. If the mother was present at the marriage of her son, we know that she died between 1888 and 1849+110=1959 and if she was mentioned as deceased, we can expect that she died between 1888–25=1863 and 1888.

All calculation rules are defined in a specific table in which we may change our assumptions of minimum and maximum ranges. For example, we may change the range of the maximal age of a married woman as soon as we find someone who married after the age of 90. More critical is the assumption of giving birth in the age range from 14–50 years; since births below the age of 16 or above 47 are rare. We could change this into a range of 16–47 years, balancing between missing a few matches or generating false ones. Even more critical is the condition of "born in a legitimate state". A previous study on a sample of the population of the province of Noord-Holland showed that on average during the 19th century this is not the case for about 5% of all first-born children (Kok, 1991, pp. 46–48). From the birth certificates of children of all birth orders we found that in 1.5% of the cases no father was mentioned (Mandemakers & Laan, 2020a). Because these children were not recognized by their father upon birth, their biological relation is debatable. However, we might still be dealing with correct matches, when the father later appears as a legal father on the child's marriage certificate.

To explore how many links we could have missed by assuming "born in a legitimate state", we experimented with larger margins in the linking of marriage certificates, putting the potential wedding range twenty years earlier. We found for the whole of the Netherlands about 281,000 extra linked marriage certificates of which about 38,000 with a margin of one year, 62,000 with two to three years and 52,000 four to five years. Since for larger margins the percentage of exact matches dropped from 42% to 12% or lower, we decided that five years should be considered as a limit for accepting matches. In a second test we took a small sample of 24 *exact* matches from this range of maximum five years to check if these children were formally acknowledged by the father on the birth certificate and/or on the parental marriage certificate. This was true in 100% of the cases.[5] Since it is easy to implement changes in the ranges, we intend with new releases to relax this requirement into "born in or five years before a legitimate state". Although this may result in relatively more false links, the researcher can make his own decisions on the basis of information about the time lag and the quality of the matches (especially the father link since his name is not always originally included in the birth certificate).

A birth certificate includes three roles (child, mother and father), three different conditions for calculating ranges (age of the involved person known, no age known, or known to be alive or not), for three events (birth, marriage or death). This results in 3*3*3=27 cases to take into account when calculating minimum and maximum time ranges for all three roles in a birth certificate. For the marriage and death certificates there are respectively 54 and 36 different conditions. The calculation of age ranges is even more complicated since in some cases interdependencies exist between the different procedures which have been solved by developing several specific functions that overrule the outcomes of the initial calculations. In the previous example, this concerns the theoretical marriage range of the father (1792–1863) which is limited by the marriage range of the mother (1827–1863) or the minimum range of death which is always defined by an event in which a person is registered as being alive.

4 DESIGN OF THE MATCHING SYSTEMS

4.1 MATCHING APPROACHES

The linkage problem posed by the LINKS database is the identification of individuals and their family relations on the basis of multiple mentions in historical civil records. This process is complicated because names are seldomly unique identifiers for persons, even though in the Dutch civil administration everyone (also women) keeps the first name and family name given at birth during life time. Still, the same person may occur with different (spellings of) names, and a single name may refer to multiple persons. Therefore, for the identification of an individual (ego), related actors are needed, notably the parents and partner(s). The combination of multiple names and time ranges for birth, marriage, and death has a high probability of leading to a unique identification. Functional relational combinations are ego and partner, ego and mother (mentioned at birth, marriage and death of ego), ego, father and mother (mentioned at birth, marriage and death of ego), and ego, parents and partner (mentioned at marriage and death of ego).

The combination of the ego and partner forms the backbone of our family reconstruction as they are mentioned as bride or groom in their own marriage certificate, and can be linked to the marriage of their children where they are mentioned as parents. Links to their mentions as parents in the birth certificates of their children makes it possible to form families, while links to their mentions in death certificates complete their life history of vital events. In several matching operations combinations of three or four persons could be used to match certificates. In principle, by using more than two persons these matches show less ambiguous results than matching on only two persons (ego and mother). Once these relationships have been established the full family reconstruction is realized in a post-matching stage (see Section 6.3).

In the first instance a matching program was developed using SQL queries in a MySQL database. In Section 4.2 we will explain this SQL-based system. However, this system was relatively slow.

5 We sampled 12 cases for the father line and 12 for the mother line, each divided in three groups of four, one for 0–1 years before the marriage, one for 2–3 years and one for 4–5 years. We also checked eight parental marriage certificates of children who were born 6–10 years before their mother married. These children had in less than 50% of all cases, the same person as father as the one appearing on the marriage certificate of their mother (and appearing as (a false) father on their own certificate).

Therefore, in order to deal with the ever-increasing scale on which civil certificates were matched, a much faster system was developed by Joe Raad. This system called "burgerLinker" (*burger* meaning 'citizen') mostly applies the same matching rules, but matches compressed knowledge graphs rather than MySQL data. In Section 4.3 we will explain this graph-based system.

4.2 LINKING WITH SQL-QUERIES

4.2.1 PREMATCH TABLES

For the matching of family names and first names we use Levenshtein edit distances. We also tested the Jaro-Winkler algorithm which gives a stronger weight to the first characters of a name (Schraagen, 2014). However, this algorithm did not provide better results. This is probably the effect of the very good quality of the names in the certificates, indicating that the last half of a string is not much more vulnerable to spelling mistakes than the first half.

To speed up the matching process, we make use of prematch tables, one for first names and one for family names. These tables include all combinations of family names or first names within a certain maximum Levenshtein distance. Matching two names is thus simplified since it is possible to find relevant combinations in look-up tables. Because the Levenshtein distance is influenced by the length of both names, the maximum distance was made dependent on the length of the shortest name of a pair. This relation is given in Table 3, which presents the number of matched pairs of names for both first names and family names. We created two prematch tables with a distinction on the way Levenshtein distances and lengths are combined: one with relatively free requirements and a stricter one. In order to save processing time in the case of the freer one, we blocked on the first character. As one can see in Table 3, this has the effect that for a minimal length four or shorter, the freer variant has less matched pairs than the strict one. However, for name lengths of five and higher, the freer one results for the first names in a total of 15.22 million pairs to be looked up, whereas the strict one ends with 7.12 million. We see a similar mechanism with the family names.

Working with prematch tables has the big advantage that during the matching process it is not necessary to calculate Levenshtein again and again for the same pair of names. By excluding pairs with high Levenshtein values from these tables we make the matching process even simpler. So, we limit the pre-match tables to relatively low Levenshtein values that could lead to matchable results. Before the start of the matching process the user needs to put a limit on the accepted Levenshtein variance and to choose which table must be used by the system.

Besides the reference tables with spelling variants of names, we also developed so-called "Root name" tables, both for first names and family names. The idea is that two names could refer to the same individual while they have a large Levenshtein distance. Language is an issue, where "William" "Guillaume" or "Willem" may denote the same person (Bloothooft et al., 2020; Oosten, 2008).[6] The same occurs with abbreviations or short forms such as "Jan" which originates from "Johannes" with a Levenshtein distance of 6. We make use of existing tables with root names (Bloothooft & Schraagen, 2015). We also intend to develop new variants by checking combinations of non-matching names in situations where all name elements (minimal two first names and two family names) have a match except one element.

However, we are not sure how to use root name matching. First experiments show that when we use root matching above the existing matching with spelling variants, we get about 5% more matches but at least half of them proved to be false. So, this requires additional decision rules to make distinctions between true and false positives.

It is also possible to include the third element in Dutch name structure: the prefix such as "de" in "de Boer". So far, we have ignored the prefix in the linkage process, since it adds little value to the uniqueness of a name.

First names may also contain more than one element, so-called multiple names, e.g., "Cornelia Theresa Antonia Maria." In the look up tables we handle each name separately. This makes it possible to match only the first or the first two names of a multiple name or matching one part of a multiple name with any part of the other name or more variations.

6 See the CLARIAH financed NAMES project, https://taalmaterialen.ivdnt.org/download/names-corpus/

Table 3 *Number of pairs of names, with required Levenshtein distance in relation to the length of the shortest name*

	More free application with first character blocked			**More strict application**		
Maximum accepted Levenshtein distance	Minimal length of the shortest name	Frequency in millions		Minimal length of the shortest name	Frequency in millions	
		First names	Family names		First names	Family names
0	1	0.23	0.53	1	0.23	0.53
1	2–4	0.77	1.36	2–4	1.00	1.83
2	5–7	6.12	12.23	5 or longer	7.12	14.46
3	8	4.82	7.21			
4	9 or longer	4.28	8.36			

Explanation: Frequency numbers are reciprocal.

First names may also contain more than one element, so-called multiple names, e.g., "Cornelia Theresa Antonia Maria." In the look up tables we handle each name separately. This makes it possible to match only the first or the first two names of a multiple name or matching one part of a multiple name with any part of the other name or more variations.

To speed up the matching process, all (standardized) family names and first names were replaced by numbers (one unique number for each name). In another step these tables with names and numbers were enriched with the frequency of each name in the whole dataset. By way of these frequencies, it was possible to direct the matching algorithm in such a way that names with the lowest frequencies were compared first. In this way the selections of potential matches were kept as minimal as possible to save computing time.

4.2.2 MATCH INSTRUCTIONS BY WAY OF A TABLE

The various choices that can be made in the matching procedure are included in the LINKS system by way of a table, called Match_Process, which includes the settings of all parameters that govern the matching process. See the scheme in Table 4 summarizing all parameters that can be set for each linking process.

Each record in the Match_Process table defines a specific matching procedure. To control the number of comparisons for matching, and by this the processing time, it is also possible to limit the time window for comparisons in a dynamic way.

The Match_Process table first defines the two sources to be matched and a time window within which the matching should occur. For example, in marriage to marriage matching the time window could be set in such a way that the parents found in a marriage certificate of a child only match with their own certificate 15 to 75 years before. Secondly, the period of matching can be divided into subperiods to reduce the number of comparisons, and by this processor time, for example to a time range of 10 or 20 years. In case of a time range of twenty years the first 'window' of matching includes the period 1811–1830, in which certificates are matched with parental certificates from the period 1736–1815. In the second window the certificates from 1831–1850 are matched with those from 1756–1835, etc.[7] By constructing different time windows and matching criteria, relatively small batches are created which are processed in a simultaneous way with 20 to 30 processors.

Another parameter controls the way multiple first names are handled. Multiple first names are not always complete or written in the same order, which especially affects the names of parents. For this reason, there are three options in linking multiple first names, a) on the basis of the first two names, b) only the first name or c) only one of all names (which is a very free method, for example "Johannes Christiaan" will match with "Christiaan Arnoldus Petrus Maria").

7 There are no civil certificates before 1811 except in two regions, but the system works with vast ranges that also must cover later periods, for example 1931–1950 compared to1860–1935.

Table 4 *Parameters to be set in the matching process of two sources*

	Settings source 1	**Settings source 2**
Type or sources (in all combinations)	Birth, Marriage or Death	Birth, Marriage or Death
Type of archive	All archives or a specific selection	All archives or a specific selection
Definition of ego	Role name of ego	Role name of ego
Combination of roles to be matched (couples, triples or quadruples)	Bride and groom (M) Child and mother (B, M, D)	Mother and father (B) Child and mother (B, M, D)
	Child, mother and father (B, M, D) Child, mother and partner (M, D)	Child, mother and father (B, M, D) Child, mother and partner (M, D)
	Child, mother, father and partner (M, D)	Child, mother, father and partner (M, D)
Use time range	Per combination of type of role and source to be set on/off	Per combination of type of role and source to be set on/off
Window of matching	Defining the start and end year of the matching window in a sequential way	Defining the start and end year of the matching window in a sequential way
	Settings of source 1 and source 2 or the same	
Familyname	Type look up table	
Familyname	Maximum Levenshtein level	
First name	Type look up table	
First name	Maximum Levenshtein level	
First name	Coding how different components of multiple first names are to be handled	

Explanation: Possible combinations of roles are dependent on the sources and are indicated with B (Birth certificate), M (Marriage certificate) and D (Death certificate).

Cases matching pairs of two persons involve four name elements to be compared. The order of matching is done in such a way that the first comparison is made for the element with the lowest frequency. Secondly the other name elements are taken into consideration. E.g., in case one family name is "Bakker" with a frequency of 186,231 and the other one is "Zeldenrust" with a frequency of 1,059 the comparison is limited to the selection including "Zeldenrust". The last step is that the outcome is compared with the time ranges as defined for each person (which will always be a subperiod from the set time window for the overall matching, see Section 3.2). So, the date ranges are used as the last step of the process for accepting a match between two certificates or not. The outcome of each comparison is no match, one match or multiple matches. The looser the matching criteria the more matches and multiple matches will be created. To further speed-up the process, a subsequent comparison step is only performed if its predecessor succeeded.

4.3 LINKING WITH BURGERLINKER

4.3.1 BACKGROUND

The matching software within the MySQL environment was used to match records from the Dutch province of Zeeland and other relatively small areas. On the basis of the matching results, family reconstitutions were created (see Section 6). But on a national level, where tens of millions of certificates needed to be matched, the SQL environment proved to be relatively slow, which led to run-time problems and compelled splitting the job into sub-jobs. An alternative to the LINKS software was needed to speed up matching and to create a more general system that could be used outside the LINKS environment. Hence, burgerLinker was developed in a collaboration between the IISH, Utrecht University and Vrije Universiteit Amsterdam (for a comparison of record linkage techniques, see Christen, Vatsalan & Fu, 2014; Raad et al., 2020).

BurgerLinker is a graph-based record linkage program that uses the existing matching rules from the LINKS SQL-query environment, but retrieves candidate matches between certificates more efficiently. The program is designed as a stand-alone, scalable, and flexible software that allows the matching of other types of historical demographic records. Just as in LINKS, users can change the default settings for Levenshtein distance, ignoring filtering based on the dates of the certificates, avoiding matches without an identical first letter on the family name, ignoring parental names, or a mixture of the above. This leaves the desired precision and recall to the user, allowing users to see which candidate matches are filtered out and why. This contrasts with the MySQL environment, which was designed to deliver an optimal number of matches based on the discussions and decisions of the experts associated with the LINKS database. Users of previous releases could decide not to use certain matches which were flagged as weak, but were unable to create new matches based on their own criteria.

In the following, we will go into the processing aspect of the data and the data model, the working of the Levenshtein algorithm, the flexible recall and the increased transparency of the system.

4.3.2 BURGERLINKER PIPELINE AND DATA MODEL

BurgerLinker is graph-based and expects an HDT file (Header, Dictionary, Triples) as input, which is a standard format used within the Resource Description Framework (RDF). HDT compresses datasets in a significant way while maintaining efficient search and browse operations without prior decompression. The RDF format is a W3C standard (along with HTML, CSS, and XML) that models data through so-called triples, describing an entity with attributes and the value of attributes. In RDF-terminology, the entity is called the subject, the attribute is called the predicate, and the value is called the object. For example, "Nicholas de Vries" can be described as a person with the first name Nicholas. RDF writes this down in two triples:

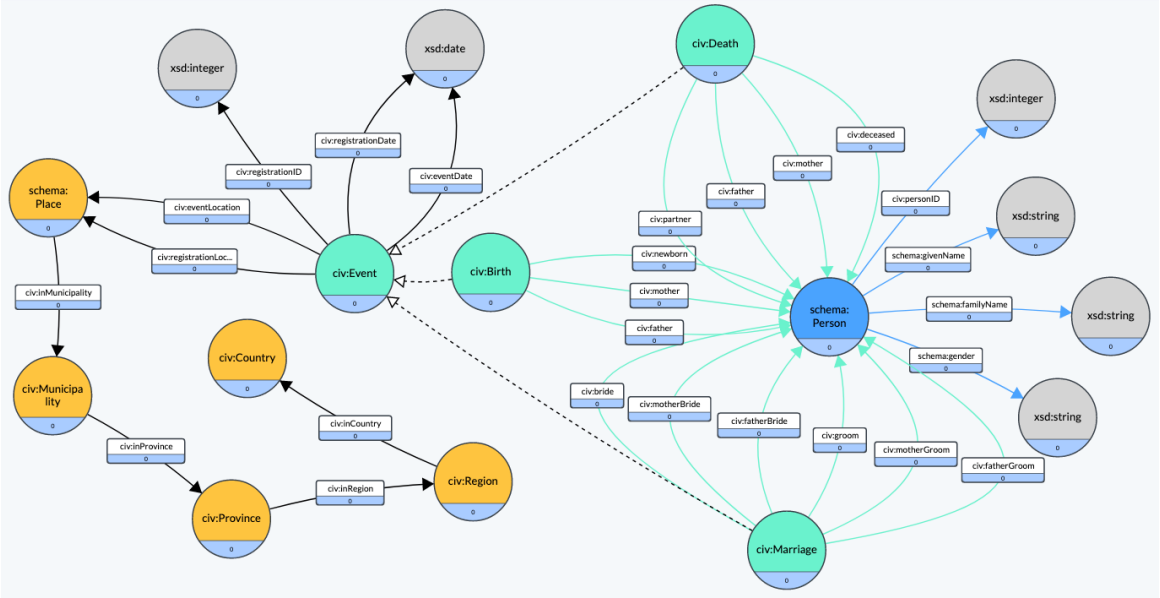
```
civ:personID_9999 rdf:type          schema:Person
civ:personID_9999 schema:givenName  "Nicholas"
```

Each part of a triple is always represented by a so-called Uniform Resource Identifier (URI), which is similar to a Uniform Resource Locator (URL), except that they are global identifiers rather than only locators or addresses. The prepositions "civ", "rdf", and "schema" are abbreviations for so-called namespaces referring to existing vocabularies in which the content is defined. The local identifier and namespace ensure that all properties in the graph are not only unique in the LINKS environment but are also globally unique on the World Wide Web. In this example, "civ" refers to the IISH namespace, "rdf" to the w3c namespace, and "schema" to schema.org. The first triple says that the personID_9999, that was assigned a global identifier using the IISH namespace, is an instance of the class Person. This triple will be interpreted similarly across all RDF applications since it has been declared using the rdf:type property standardised by the W3C, and using the class Person defined by the schema.org vocabulary, commonly used in different applications and webpages. When using standardised vocabularies such as RDF and RDFS (RDF Schema), users can directly benefit from certain reasoning capabilities supported in most RDF platforms. For instance, given the following two triples in an RDF graph: civ:personID_9999 rdf:type civ:Male; civ:Male rdfs:subClassOf schema:Person. An RDFS reasoner allows us to infer on demand a third triple indicating that personID_9999 is also an instance of the class schema:Person, as the second triple declares that all instances of civ:Male are also instances of schema:Person.

Data for burgerLinker is exported from the LINKS MySQL database of standardized and cleaned data as a Comma Separated File (CSV). By way of the IISH standard tool COW, a script is run to convert the data from CSV to RDF.[8] The required HDT input for burgerLinker is created by a) importing the data in RDF format and b) using an embedded tool to convert the RDF dataset into HDT. Within burgerLinker we can then match the civil certificates, and export the results as RDF or CSV.

8 The Python library COW (CSV On the Web; https://csvw-converter.readthedocs.io/en/latest/) allows flexible conversion of CSV datasets to RDF by relying on a JSON schema. For an example JSON schema that converts LINKS datasets from CSV to RDF according to the CIV model on the burgerLinker GitHub (https://github.com/CLARIAH/burgerLinker/blob/main/assets/examples/births_example.csv-metadata.json).

Figure 4 *Civil Registries schema for burgerLinker*



Explanation: BurgerLinker retrieves information on persons (blue), events (in green), and locations (in yellow) using the schemas Civil Registries schema (civ) and schema.org (schema). The gray indicates the expected type of literal values. Extra attributes can be added to the model, for instance, occupation, address, or age for persons, the name of the clerk for registrations, or the names of witnesses for events. Note that some of these variables are already defined in the Civil Registries Schema, such as age or occupation. See also Appendix B.

BurgerLinker requires that the data are modelled according to a so-called schema. A schema consists of classes (and instances of classes), which are the main entities of the data structure. The classes are associated with (sub) schemas describing specific cases of persons or locations. The schema designed for civil certificates is included in Figure 4 and is named "Civil Registries schema" (CIV). Figure 4 describes the core parts of this data model. The main entities or classes are presented as nodes. We see four (green) nodes for events: three classes for each type of civil certificate: birth, marriage and death and a more general class for the information about the event itself, heading schemas for dates and places both for the event itself and the registration of the event. Each event has a defined set of persons, for example, child, father and mother in the birth certificate of which the attributes are defined in the schema Person. The schemas are defined in our own CIV model or derived from existing schemas, in this case "xsd" for the data type (date, integer or string) and "place" for locations linked with our own more general schemas for the municipality, province, region and country. Each arrow in Figure 4 represents a triple pattern which defines the roles of the persons involved in an event and the attributes that are included in the model, for example, the name with the gray node indicating the type of literal values that are expected. In Appendix B we have included a more formal and complete description of the CIV model.

After matching, the results need to be combined with other retrieved data such as professional titles and combined into event histories and family reconstitutions (Mourits et al., 2020).

4.3.3 OPTIMIZED LEVENSHTEIN ALGORITHM

Just like the SQL-query environment, burgerLinker uses a Levenshtein algorithm to match cases. Calculating Levenshtein distances is a time-costly process, as the number of possible matches that a Levenshtein algorithm needs to consider grows exponentially with each new name that is added to the database, thus increasing the required run-time exponentially. In the MySQL database this problem was solved by making prematch tables that store the Levenshtein distance between unique names before starting the actual matching procedure. These prematch tables made the linking within the SQL-query environment more efficient, but are an extra step in the linking process that must be reproduced when new data are added to the system. In burgerLinker, the Levenshtein distances are calculated efficiently on the fly. To speed up the computation of Levenshtein distances, burgerLinker indexes the list of target names as a Minimal Acyclic Finite-State Automaton (MA-FSA), also known as Directed Acyclic Word Graphs (DAWG). An FSA is a mathematical model or an abstract machine that operates by moving

through a series of states in response to inputs, where each state represents a particular condition or configuration of the machine. Then, a Levenshtein transducer is initialized, which is an FSA that accepts a query term (e.g., a name) and returns all terms in the index that are within n spelling errors away from it. Like the MySQL system, burgerLinker allows the user to specify the maximum accepted distance for a match, with 4 being the maximum allowed distance in the current version. This procedure, implemented in the JAVA library *liblevenshtein* based on the work of Schulz and Mihov (2002), is much more efficient than the original Levenshtein algorithm, as its runtime complexity grows linearly with the length of the query term, rather than exponentially on the size of the index (Raad et al., 2020).

4.3.4 FLEXIBLE RECALL AND INCREASED TRANSPARENCY

In its earliest stage, burgerLinker produced the same matches as the SQL-query environment, using the same matching principles. However, during the testing of burgerLinker we changed strategies and opted to aim at retrieving as many candidate matches as possible. The policy for the construction of releases in the SQL environment focused on finding as many unique matches as possible within the Dutch civil registry, prioritizing the quality of established matches and limiting the retrieval of candidate matches as soon as too many multiple matches appeared. This restrictiveness on the number of matches was advantageous for researchers as the retrieved dataset was ready for analysis. Yet, this optimum is not always the same for different datasets and there is also some variance between disciplines in what researchers deem the optimal balance between recall and precision (see Section 5.2 for an elaborate discussion of this balance). The structure and matching speed of burgerLinker make it relatively easy to match with different alternative designs and define how we filter matches to get more precision at a later stage.

Secondly, by increasing the importance of the filtering procedure after the matching, burgerLinker makes the whole matching process more flexible and transparent. Just like the LINKS query system of the MySQL database, burgerLinker provides extensive data on the background of these matches. However, since burgerLinker can be used as a stand-alone tool, users are independent from the database manager in running the matching program, and can decide the maximum Levenshtein distance per name on the spot, as well as the number of persons on a certificate that should match.[9] Just as in the SQL-environment, the Levenshtein distances are made dependent on the character length of the smallest item to be matched (see Table 5 and compare Table 4). By giving users the possibility to retrieve a larger set of candidate matches, the matching procedure becomes more transparent, as it becomes clearer which candidate matches are rejected to get a higher precision.

Table 5 *Changeable settings in the burgerLinker matching environment*

Settings	**Default**
Maximum Levenshtein distance	4
Fixed Levenshtein distance	False
Ignore date consistency check	False
Ignore blocking first letter last name	False
Match single individual	False

Max Lev Distance	**Restriction of Lev Distance based on name length**				
	0	1	2	3	4
0	1+	-	-	-	-
1	1–5	6+	-	-	-
2	1	2–8	9+	-	-
3	1	2–5	6–11	12+	-
4	1	2–5	6–8	9–11	12+

Explanation: A choice for a specific maximum Levenshtein distance automatically sets lower values in case of relatively short name lengths. So, Levenshtein 4 will not work for names smaller than 12 characters.

9 See https://www.github.com/clariah/burgerlinker

The way in which first names are matched was slightly altered to retrieve more candidate matches. Just as in the LINKS system, each first name is matched separately, rather than considering them as one string. For example, "Hendrikus Kornelis Romein" can now match to "Hendrikus Romein" and "Kornelis Romein." The difference is that in burgerLinker no choices have to be made in deciding how and which part of the multiple first names will be matched (compare Section 4.2.2, Table 4). Although this procedure can lead to some overmatching, it also reveals many potentially useful matches. However, to limit obvious mismatches, two entries with multiple first names will not match when separate elements of the shortest multiple first name does not match. For example, "Hendrikus Kornelis" can match to "Hendrik Kornelis Aloysius", but not to "Johannes Hendrikus Wilhelm", because the last one lacks "Kornelis." Systematic filtering is possible due to the detailed metadata on the number of matched first names, the Levenshtein distance per matched first name, and the total Levenshtein distance of all matched first names. As a result, we can still retrieve the same matches as the SQL-environment, provided that we use the same rules for filtering.

Like the database manager in the LINKS MySQL-system, users of the burgerLinker system may choose to ignore the date consistency check, ignore blocking of the first letter of the last name, or match only on the name of one of the indexed persons (see Table 5). In general, matching on one person is not advisable, since it will give an enormous number of false matches. By default, the system will match on the ego and the mother and additionally on the father and a partner if possible. We also made it easy for the user to decide if children should be born in wedlock or not (see discussion in Section 3.2).

The main difference between burgerLinker and the MySQL query environment is that burgerLinker is designed as a tool for general use. Our hope is that burgerLinker can serve as a tool to make matching procedures within historical demography easier for researchers and also more comparable by introducing a standard way of matching. The introduction of the Intermediate Data Structure (IDS) for life course databases (Alter & Mandemakers, 2014) and its wide acceptance in the field, have laid the basis for common software, but the system supposes that the record linkage is done by the database itself. Because each database has its own selection of sources with their own local peculiarities, database managers have developed a wide range of different matching strategies. BurgerLinker will not replace these existing matching programs, but it can easily link new data to existing datasets. It could also be used in validating the quality of existing matches and help to make alternative matchings which deviate on some aspects from the standard release. This will be facilitated by converting unlinked data into IDS. The IDS is structured according to the principle of the Entity Attribute Value model (EAV) or object-attribute-value model, which was introduced in the 1970s (Stead, Hammond, & Straube, 1982). This is exactly the same structure as the RDF triple system in which the subject is the entity, the predicate is the attribute, and the object is the value (https://graphdb.ontotext.com/documentation/9.8/enterprise/devhub/rdfs.html#what-is-rdf). This makes the creation of conversion scripts to create triple systems a relatively easy job. Hence, burgerLinker is the HSNDB's effort to share our experience in matching civil records with the community.

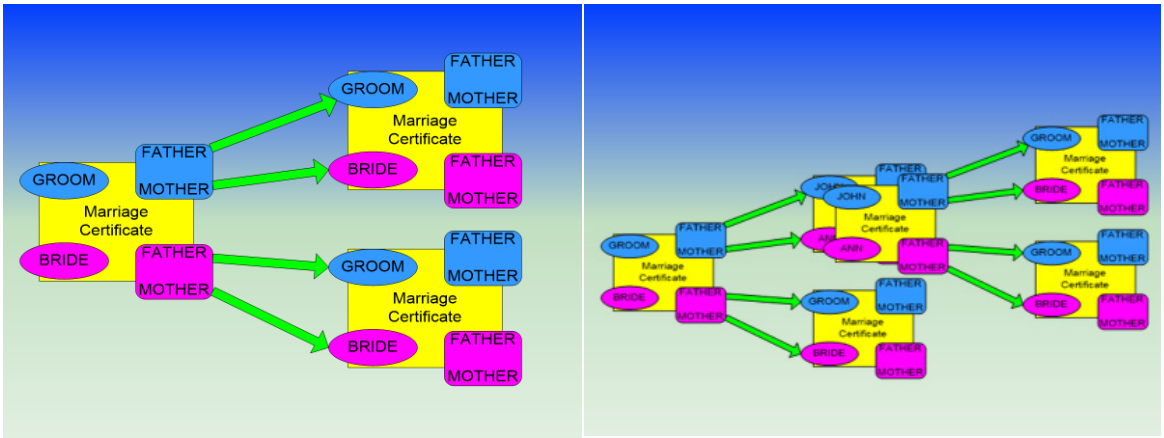
5 MATCHING EXPERIMENTS WITH MARRIAGE CERTIFICATES

5.1 THE CASE OF ZEELAND

In this section we will show the result of our experiments with the matching system, limiting our evaluation to the marriage certificates of the province of Zeeland. We choose the province of Zeeland because of the completeness of the dataset and the relatively limited number of inhabitants (on average about 5% of Dutch population).

Figure 5 presents the challenge: the father and mother of a bride or a groom are to be matched with their own marriage certificate. As soon as a match is found we have a family tie between three generations. Since both bride and groom are matchable with the marriage of their parents, we work along two lines which we call the bride and the groom line. After matching we may combine the matched pairs of certificates into lines of multiple generations. This kind of matching is relatively easy and straightforward since the *identifying* information of the marriage certificate of the second part of a pair will be identical with the first part of another pair. Essentially, this is a process of matching pairs of persons to get linked certificates.

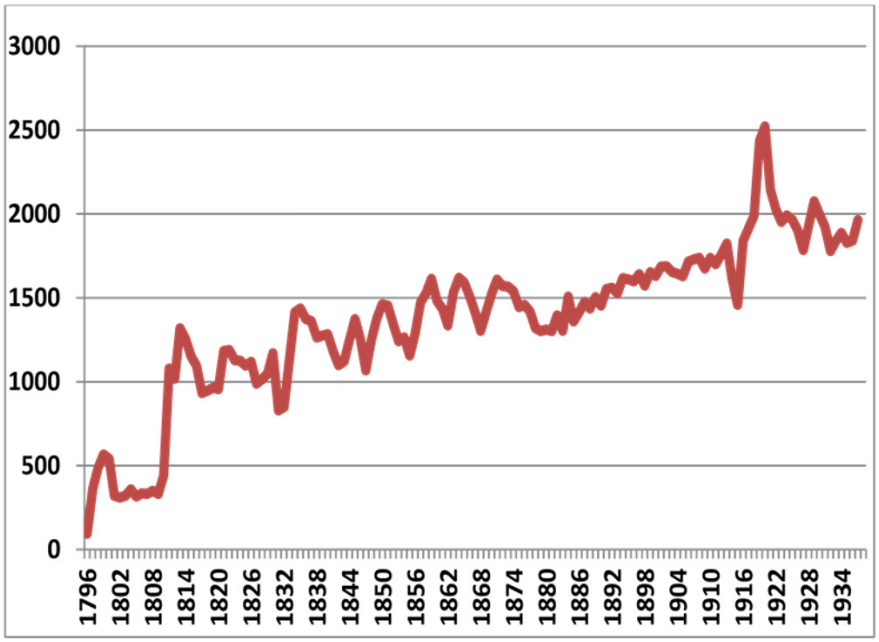
Figure 5 *Scheme of the linking process of marriage certificates into pedigrees*



For Zeeland there are indexed certificates for the period 1796–1936. See Figure 6 for an overview of the numbers per year. For most of the country, civil certificates were introduced in 1811. However, the southern part of Zeeland (Zeeuws-Vlaanderen) had started recording certificates in 1796, because they were annexed by France in 1795 (Vulsma, 1988). The rise in the numbers in 1811 is explained by the inclusion of the rest of the province Zeeland. The strong fluctuations between 1916 and 1920 were a consequence of the bad economic situation during the last two years of the First World War (although the Netherlands were not involved in the fighting) and the subsequent optimism in the three years after the end of this war (van der Bie, 1995; van Zanden & Griffiths, 1989).

All in all, the total number of included marriage certificates amounts to 191,847. This number excludes certificates before 1801, because we found a lot of double registrations of the same marriage in different municipalities, which would systematically result in double matches. Since each marriage consists of two partners, we can expect theoretically twice the number of matched parental certificates: 383,694. However, in practice not all certificates could be matched with those of the parents due to two main restrictions. The first one concerns the time window of observation: parents of persons who married before 1830 will have no chance to be matched with their own marriage certificate, since the civil registration started only in July 1811 (with the exception of some from Zeeuws-Vlaanderen). Similarly, the parents of persons who married between 1830 and 1860 are estimated to have on average only a 50% chance to be matched to their own marriage certificate. If we take this into account, there are only 339,240 potential matches left. But this number will not be reached, because — and this is the second restriction — an unknown proportion of parents who married outside Zeeland were not included in the matching operation.

Figure 6 *Number of marriage certificates per year, Zeeland, 1796–1937*



5.2 RESULTS

To measure the quality of our results, we estimate recall and precision. In record linkage the term "recall" stands for the relative share of retrieved matches compared to all potential matches, and a higher recall indicates fewer missed matches, also called false negatives. Precision is the relative share of correct matches compared to all matches made, and a higher precision indicates fewer false matches, also known as false positives. False positives are most clear when more than one match is found for the same record. This kind of multiple matching we call overlinking. In our case, overlinking exists when the parents of the groom or the bride are matched with more than one other certificate, which can only be true in the very rare case that parents remarried after having had a divorce.

The ideal in record linkage is to arrive at zero false negatives (recall) and zero false positives (precision). Since there is no information on the actual number of correct matches, we need to use other indicators to get an impression of the quality of the matching. Therefore, we use the total number of retrieved matches as an assessment of recall and the number of multiple links for the same certificate as an assessment of precision. This procedure allows us to test different matching criteria, where we go from very strict matching criteria to more free ones, while we stop when the number of matched certificates increases marginally and the number of multiple links expands exponentially (Oosten, 2008).

We combined three different methods of matching. Firstly, name matching varies depending on blocking on the first character or not in combination with the setting of the Levenshtein values. Secondly, we matched multiple first names in three variants: a) the two first names combined, b) only the first name and c) one name out of all names. And thirdly it is possible to remove checking the date range in which a parental marriage should be expected. The results of our matching operation or shown in Table 6.

We start with a baseline of very strict matching in which we a) match both for family names and first names exactly with Levenshtein=0, b) include the two first parts of a first name (if present), and c) accept no matches that are outside the estimated marriage range. This is the strictest matching set-up the system offers. The right half of Table 6 shows the result of each matching method in a) terms of the number of matches and overlinks and b) by way of an index that show the relative differences with the baseline. The right-most column shows the ratio between both indexes. If the ratio stays close to one, it indicates that the matching operation does not produce relatively more overlinking than the more secure matching operations.

The baseline shows a result of 163,237 close to matches. That is almost half of the theoretical total (n=339,240). The number of overlinks is 158 which is less than 1 in 1000. The names on the overlinked certificates are identical and all fit the time range, meaning that they cannot be distinguished by the matching software. In a second step, we raised the Levenshtein values to 2. This resulted in 10.4% extra matches and 11.4% extra overlinks, which corresponds to a ratio of 1.01. In other words, it does not make much difference when Levenshtein 0, 1 or 2 is used in the matching procedure, the quality of the matching remains the same and we have 16,955 extra matched certificates. Choosing Levenshtein 4 did not show much difference either; compared with Levenshtein 2 it returned 11.0% extra matches and 18.4% extra overlinks, which corresponds with a ratio of 1.07.

In the second group of matching exercises, we experimented with 'freeing' the first character. This was only done for a maximum Levenshtein value of 2, since earlier experiments with unblocking the first character and accepting higher Levenshtein values resulted in an explosion of false matches. We see that freeing the first character of the first name with Levenshtein 2 does add 2,381 matches (1.3%) compared to the fixed variant with a ratio of 1.02 (compared to 1.01). The other options which include freeing the first character of the family name or accepting a Levenshtein value of 4 for the first name results in ratios between 1.09 and 1.14 which do not differ much from each other nor from the baseline. And in the last option we have 20,377 (12.5%) more matches than from the baseline settings.

In the third group, we changed the way the first name is handled. At the beginning of our period, only 30% of the persons born in Zeeland had two or more names, at the end, this percentage had risen to almost 60%, of which 10% consisted of three or more names (Gerritzen, 1998). So, for roughly half our population, there might be different results depending on how the first name is handled. Restricting the first name to the first part gives an extra 9,546 matches compared with the baseline (Levenshtein 2, fixed first character, two name parts) and 26,501 more compared with the baseline with the exact match. The overlinking increases within reasonable limits from 158 to 232, which corresponds with a ratio of 1.26. The freer variants in this group result in more matches with a relatively low number of overlinks (ratio 1.33 and 1.53).

Table 6 *Results of the matching process, Zeeland, marriage certificates, 1801–1937*

Familyname		First name			Dates	Quality indicators				
First character	Maximum Levenshtein	First character	Maximum Levenshtein	Elements used	Ranges	Matches		Overlinks		
						N	Index	N	Index	Ratio
fixed	0	fixed	0	1+2	fixed	163,237	100.0	158	100.0	1.00
fixed	2	fixed	2	1+2	fixed	180,192	110.4	176	111.4	1.01
fixed	4	fixed	4	1+2	fixed	181,114	111.0	187	118.4	1.07
fixed	2	free	2	1+2	fixed	182,573	111.8	181	114.6	1.02
free	2	fixed	2	1+2	fixed	181,323	111.1	195	123.4	1.11
fixed	4	free	2	1+2	fixed	182,826	112.0	193	122.2	1.09
free	2	fixed	4	1+2	fixed	182,002	111.5	197	124.7	1.12
free	2	free	2	1+2	fixed	183,714	112.5	203	128.5	1.14
fixed	2	fixed	2	1	fixed	189,738	116.2	232	146.8	1.26
fixed	4	fixed	4	1	fixed	190,548	116.7	245	155.1	1.33
free	2	free	2	1	fixed	193,295	118.4	287	181.6	1.53
fixed	2	fixed	2	1 of all	fixed	196,062	120.1	320	202.5	1.69
fixed	4	fixed	4	1 of all	fixed	196,951	120.7	336	212.7	1.76
free	2	free	2	1 of all	fixed	199,551	122.2	411	260.1	2.13
fixed	2	fixed	2	1+2	free	183,276	112.3	559	353.8	3.15
fixed	2	fixed	2	1 of all	free	200,595	122.9	1,520	962.0	7.83
fixed	4	fixed	4	1 of all	free	201,575	123.5	1,589	1005.7	8.14

Explanation: The columns "First character" indicates if the first character is fixed (or blocked) in the matching process of free. The column "Elements used" indicates how a composed firstname is matched ("1" only the first one, "1+2" only the first two ones and "1 of all" only one random element with another one). The column "Dates" indicates if the estimated ranges within parents will marry are used to limit matching possibilities. Free means that there was no check on these ranges.

In the fourth group, we again changed the way the first name is handled. We matched in such a way that each part of the first name had an equal chance to be part of the match. A first name like "Maria Elisabeth Antonia" will match "Maria", "Elisabeth" and "Antonia." The other settings are the same as in the third group. If we compare the results, we see that for all three lines this action results in about 6,300 extra matches. Looking at the maximum number of matches compared with the baseline with exact matches, there are 36,314 extra matches while still having only a moderate increase in overlinking to a ratio of 2.13.

In the fifth and last group of Table 6, we removed the constraints on the minimum and maximum ranges of parental marriage. We see that the number of overlinks increases with a little gain in matched certificates. This results in ratios that vary from 3.15 to 8.14. In the first case, which blocked only the first character of the first name, there are 3,084 matches more than the comparable matching with fixed date ranges; while the number of overlinks increases threefold from 176 to 559. In the last two rows, we matched all parts of the first name separately. Although we got respectively 4,533 and 4,624 extra matches, compared with the first row where we matched one firstname to all other ones ("1 of all"), the number of overlinks grew almost fivefold, resulting in a ratio of respectively 7.83 and 8.14 against the exact baseline. Given this last result we conclude that 'freeing' logical date ranges is a bad strategy, unless it is done in a very limited way (compare Section 3.2).

On the basis of this experiment, we concluded that in the case of the marriage certificates, it did make a small but not unimportant difference to free the first character (given a Levenshtein level of 2). In practice, we found that most of the cases were typical features of Dutch language: mixing up of "y" and "ij", "c" and "k", "f" and "ph", "s" and "z", "ch" and "g". Because freeing the first character is very expensive in terms of computing time, we decided to stay with fixing the first letter, but standardizing the family names as described in Section 3.1.

All in all, we consider the result of our experiments as a proof of the excellent quality of the data in general, especially comparing our results with linkage between American censuses (Goeken, Huynh, Lynch, & Vick, 2011). The difference between the result of the exact matching and the most flexible alternative was only 38,338 matches (23.5%). Two factors contribute to our success. First, the high quality of the data which was due to legal requirements governing the civil registration system, especially the obligation to submit official extracts of birth certificates as part of marriage registration (Vulsma, 1988). Second, since females retained their own family name, we are usually matching pairs of people instead of individuals.

What matching strategy can be distilled from our experiments? We learned that standardization of typical Dutch spelling variances limits the advantages of freer matching, especially above the limit of Levenshtein 2. Secondly, matching with date ranges is very useful in limiting overlinking. That leaves the question of how to match with multiple first names. Should we use only the first name or the first two? Alternatively, can we use an algorithm in which one part of a first name will be sufficient, independent of its place in the sequence of names? For Zeeland, matching the first name compared with matching the first two names, resulted in more matches with a limited increase in overlinking. Comparing one part of a first name with all other parts, also seems acceptable. However, the degree of overlinking probably will be higher in other parts of the Netherlands as about half of the Zeeland population had only one first name. We may expect that the degree of overlinking grows exponentially when the entire population has multiple first names, especially higher social groups and the Roman Catholic part of the population which was very generous in giving multiple first names (Bloothooft & Onland, 2016; Gerritzen, 1998). On the other hand, less precision can be acceptable when all types of certificates are linked to produce family reconstitutions that can be used to test the integrity of the whole family (see Bloothooft, van Boheemen, & Schraagen, 2016; van Boheemen, 2016).

We can conclude from the results in Table 6 that about 199,000 matches is the maximum that we may expect from this dataset. We calculated a theoretical total of 339,000 matches. This implies that about 140,000 (42%) of the parents married outside Zeeland during the period from ca. 1810 till 1910. We may test this in future 1) by linking the certificates of Zeeland with certificates covering the rest of the Netherlands and 2) by adding matching algorithms working with root names. We expect more matches using roots for the first name, because they will take into account abbreviations, such as Jan instead of the "Johannes", and translations, such as "Guillaume" instead of "Willem" (Oosten, 2008). We will also select cases to be examined manually to find software bugs and dataset errors, such as certificates that have been entered twice in the original dataset, and to determine if some double links are really couples who married each other for a second time. Future versions of LINKS will keep using the number of matches and overlinks as proxies for recall and precision. The optimal settings will probably differ by region and time period, but the ratio will help determine the optimal settings for each context.

6 RELEASES

6.1 INTRODUCTION

In the foregoing, we explained how LINKS operates in cleaning and matching the civil certificates from the WieWasWie indices. However, matched certificates are only the basis for a research dataset. As this is being written, 40 datasets have been constructed, of which five can be downloaded directly (https://datasets.iisg.amsterdam/dataverse/hsndb-links). For a full overview of all releases, see https://iisg.amsterdam/en/hsn/projects/links/links-releases. Most of these releases covered only parts of the country, dependent on the availability of indices, the transcription of occupational titles and specific requests from researchers. Over half of these releases were 'forerunners', to be used by researchers for testing the quality of the data or developing the program to construct the dataset for analysis or doing preliminary analyses. Until 2010 only the marriage certificates were indexed completely enough to link them and to make meaningful releases (of pedigrees). It took more time to index the other certificates, and the birth certificates were lagging behind. The first more or less completely indexed provinces were Zeeland, Limburg and Groningen/Drenthe. After 2010 the index improved enormously in quantity (see Table 1) and it became possible to link the country as a whole

which resulted in two large releases at the national level: all marriage certificates linked into pedigrees (Mandemakers & Laan, 2020b) and all birth certificates linked with the marriage certificates of the parents (Mandemakers & Laan, 2020a).

Ultimately, the main goal of LINKS is to deliver complete *integrated* datasets of births, marriages and deaths, creating families and multigenerational links. So far, this kind of dataset is only realized for the provinces of Zeeland, Limburg and Groningen/Drenthe separately (Mandemakers & Laan, 2017, 2018, 2019). Researchers have used these datasets to create two types of datasets suitable for statistical analysis. The first one was a rectangular data structure constructed within the context of the project Genes, Germs and Resources (Mourits et al., 2020). The second one was a reconstruction of the Zeeland release in the format of the Intermediate Data Structure (IDS; Alter & Mandemakers, 2014).

In the following we will explain first the construction and quality of the national releases and secondly the integrated ones.

6.2 THE NATIONAL RELEASES

6.2.1 MARRIAGE CERTIFICATES

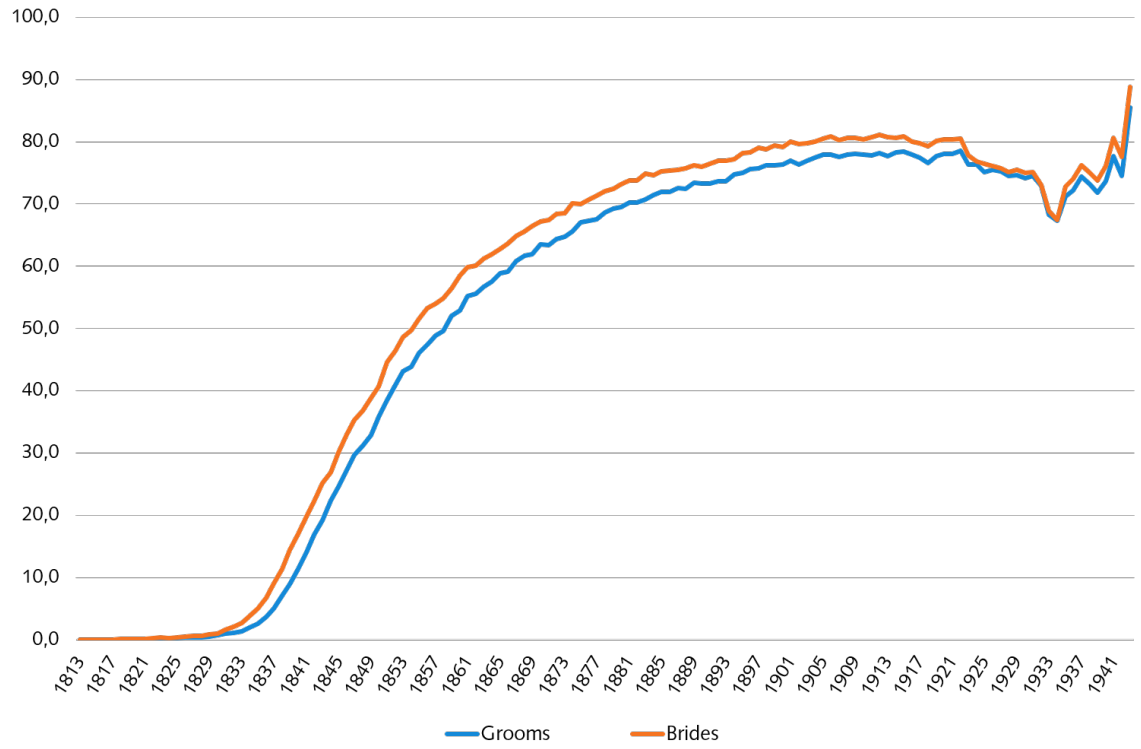
The release of the marriage certificates and their parental links (Mandemakers & Laan, 2020a) included a total of 4,158,387 certificates. The marriage certificates were linked into pedigrees (see Section 5.1 for this process). Actually, we handled 8,316,774 "marriage lines", one for the parents of the bride and one for those of the groom. The matching procedure was based on the Zeeland experiments (see Section 5.2) and consisted of four elements:

1. First names and family names were standardized as described in Section 3;
2. The first name and family name of each person were separately matched with an accepted level of variance of maximal Levenshtein 2;
3. If the first name of a person consisted of more than one part only the first part was used for matching, so a first name as "Cornelis Albert Maria" was restricted to "Cornelis";
4. The date of the parental marriage has to be 14 to 49 years before the date of the marriage certificate in which they show up as parents. This range was based on the childbearing ages of the mother and was further limited if the age of the newlyweds was indexed as well.

Matching pairs of persons means that four different strings were compared and matched: two first names and two family names which implies that Levenshtein distances could be as high as 4*2 equals 8. Some persons married more than once, which is indicated by the civil status of the bride or groom. However, this kind of information is not included in the index in a systematic way. A second matching between the parents of the marriage certificates themselves (see Section 6.3.3) could provide this information, but this operation has not been done on the national level yet. The indexed information from the certificates is limited. As mentioned in Section 2, all archives include at least the municipality and date of the event as well as the first name and family name of the bride, groom and their parents and usually the age at the event for the bride and groom. Occupational titles have been transcribed for about 60% of the marriage certificates covering seven provinces (out of a total of eleven, see Figure 1).

Of the indexed marriage certificates, 99.3% date from the period 1812–1943 when civil registration was obligatory for the whole of the Netherlands. Most of the indexed certificates are from before 1922, since indexing is lagging behind the public release of certificates. Figure 7 shows the percentage linked to the parental marriage by year of marriage. This is almost zero before 1830, because parental certificates seldom show up before the age of 18 (of the parents). From 1830 until 1860 it increased to about 60% and then further rose to about 80% in 1920. The fall and the rise after 1920 are a result of the uneven development of the index. Brides always do slightly better than grooms, because marriages tended to take place in the birthplace of the bride, resulting in a better chance to link the parental marriage certificate.

Figure 7 *Relative number of linked certificates per marriage line, Netherlands, 1812–1941*



Source: LINKS dataset linked marriages (Mandemakers & Laan, 2020b).

Table 7 presents an overview of the matching results. In total, 59.8% of the brides and grooms were matched with a parental certificate. Parental marriages before 1812, mostly explain why 28.1% of the certificates were not matched. The remaining 12.0% did not link for other reasons, such as lacking indices, ambiguous matching results or foreign marriages. Brides or grooms linking with more than one parental certificate are marked in the dataset and not linked. An exception are cases with two alternatives of which one match was almost exact and the other one had a relatively high score on Levenshtein. On this basis a meagre 0.1% could be added to the linked results. These decisions are marked in the release tables, so a user may decide not to use these ambiguous links.

Table 7 *Number of marriages lines and matching results with parental marriages*

	Number		Percentage of total
Link with parental certificate	4,975,177		59.8
No ambiguous link		4,964,157	59.7
Ambiguous but reasonable choice		11,020	0.1
No link because of technical reasons	2,337,205		28.1
Ambiguous linking result (two or more links)		217,072	2.6
Lacking identifying data of one parent		99,557	1.2
Lacking identifying data of both parents		372,206	4.5
Time range (marriage certificate before 1830)		724,783	8.7
Time range (marriage certificate 1830–1860, estimation)		923,587	11.1
No link because of other reasons	1,001,590		12.0
Total	8,313,972		100

Source: LINKS dataset linked marriages (Mandemakers & Laan, 2020b).

Table 8 *Total Levenshtein value of established links marriages with parental marriages*

	Number	Percentage of total
Exact match	3,948,798	97.4
Total Levenshtein value = 1	687,795	13.8
Total Levenshtein value = 2–3	315,738	6.3
Total Levenshtein value = 4–8	22,846	0.5
Total	4,975,177	100.0

Source: LINKS dataset linked marriages (Mandemakers & Laan, 2020b).

The sum of Levenshtein distances is also made explicit for each matched certificate (see Table 8). Of all matches 79.4% proved to be an exact match and only 0.5% were matched with a total Levenshtein value of three and more. This clearly indicates the good quality of the Dutch certificates and the matching operation. That 2.6% of the certificates with more than one match remained to unresolved (as shown in Table 7) is because most of these matches are of very good quality in terms of Levenshtein distance.

6.2.2 BIRTH TO PARENTAL MARRIAGE LINKAGE

Another release on a national scale is the linkage of births with parental marriages (Mandemakers & Laan, 2020a). This kind of matching implies that both parents must be known on the birth certificate, excluding all illegitimate children. In the future, some of these births may be linked in an indirect way (through linking the child and its mother in the death certificate or marriage certificate of the child).

We used the same matching conditions used for the marriage certificates (Section 6.2.1). We also formulated a comparable time range on the basis of the age of the bride at the event of the marriage. This means that the child's birth date should be between the date of marriage of the parents and the date of marriage plus 49 years minus either the age of the bride at marriage or 14.

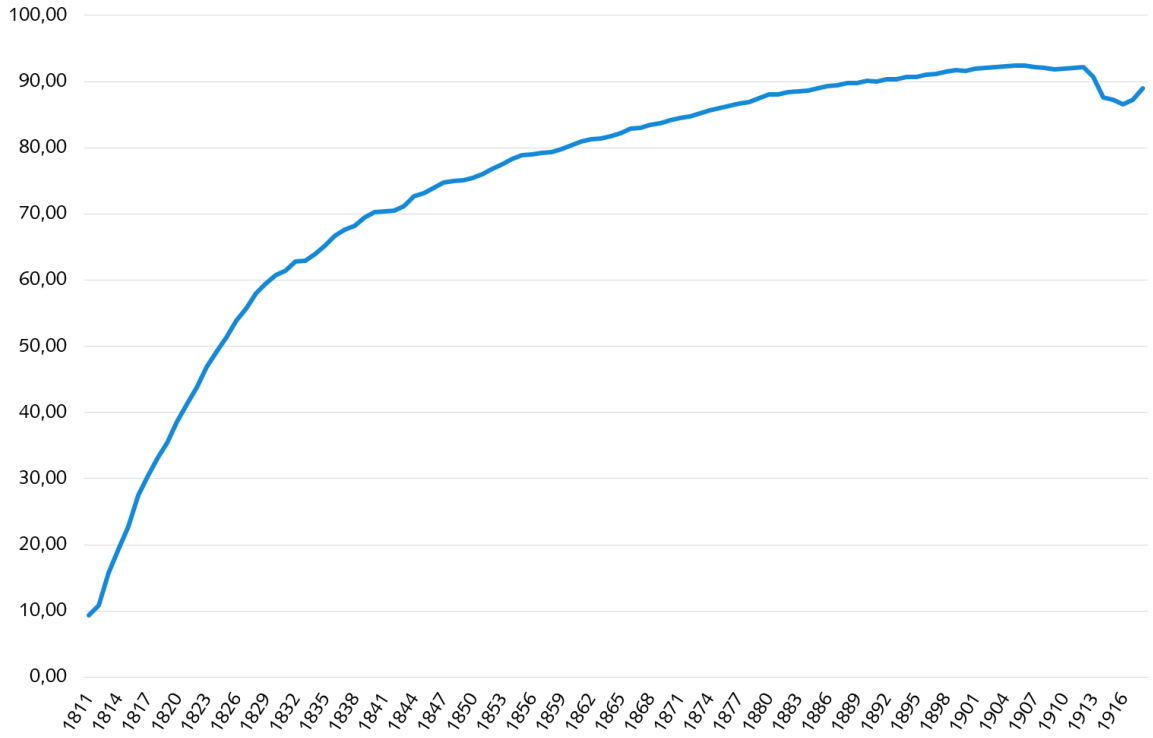
The index on birth certificates is less complete than the one for marriage certificates, e.g., Amsterdam is lacking completely. Other archives were only partially covered or had indexed only the name of the birth and no parents. To make a consistent release, we included only births from archives which had a matching rate of at least 60%, totaling 9,792,024 birth certificates which are about 2/3 of the potential number of births (compare Table 1). Of these birth certificates, 7,669,986 were linked with at least one marriage certificate (see Table 9). For 21.7% of the births no link was found. The main reasons for missing links are parental marriages before 1811 (11.3%) and incomplete data about the father (births outside a wedlock, 1.5%). For 8.5% of the missing links there is no clear reason, but some marriages are not included in the marriage index yet, and some marriages were registered outside the Netherlands. In the release we identified 25,020 births with more than one linked marriage certificate that could not be resolved. For 16,704 cases we made a choice for a specific certificate in a comparable way as we did with the marriage certificates (see Section 6.2.1).

Table 9 *Number of births and matching results with parental marriages*

	Number		Percentage of total
Link with parental certificate	7,669,986		78.3
No ambiguous link		7,653,282	78.2
Ambiguous but reasonable choice		16,704	0.2
No link because of technical reasons	1,289,625		13.2
Ambiguous linking result (two or more links)		25,020	0.3
Lacking identifying data of father		148,157	1.5
Lacking identifying data of mother		7,984	0.1
Time range (marriage could be before 1812)		1,108,464	11.3
No link because of other reasons	832,413		8.5
Total	9,792,024		100.0

Source: LINKS dataset linked births and parental marriages (Mandemakers & Laan, 2020a).

Figure 8 *Relative number of births linked with parental marriages, 1811–1918*



Source: LINKS dataset linked births and parental marriages (Mandemakers & Laan, 2020a).

Figure 8 shows the share of matched birth certificates per year. After 1850 one could expect that each birth with two parents will match with a marriage certificate. This is not always the case for reasons already mentioned. Around 1850 the percentage is about 75%, climbing to 92% for the period 1898–1912, dropping in the years of the First World War to 87%.

Looking at the Levenshtein distances, we found more or less the same results as presented in Table 8 for the linking of the marriage certificates. Of all matches, 79.6% proved to be an exact match and only 0.4% matched on the basis of a total Levenshtein value of three and more. This is another clear indication of the good quality of the Dutch certificates.

6.3 THE REGIONAL INTEGRATED RELEASES

Given the state of the indices at the end of 2017, it was possible to create integrated sets of birth, death and marriage certificates for four provinces: Zeeland, Limburg and the combination of the two bordering provinces Groningen and Drenthe (see Figure 1). For reasons of research the indices needed the inclusion of occupational titles, ruling out provinces as Utrecht and Friesland which also have high levels of indexing.

In the following we explain how we linked the different certificates, how we created uniquely identified persons out of these data and how we changed the pedigree structure into a more family tree-like structure. Firstly, we will describe the nucleus of the table system that was created out of the linked certificates. Secondly, we will elaborate on the linking process, going into the several types of matching that needed to be made. Thirdly, we will explain how the persons in all the certificates were synchronized into unique persons. In the last section we will elaborate on the outcome in terms of unique persons and families.

6.3.1 STRUCTURE OF THE DATASET

The resulting dataset consists of a system of four interlinked tables. See Figure 9 for the table structure and the relationships between the tables and the identifying keys.[10]

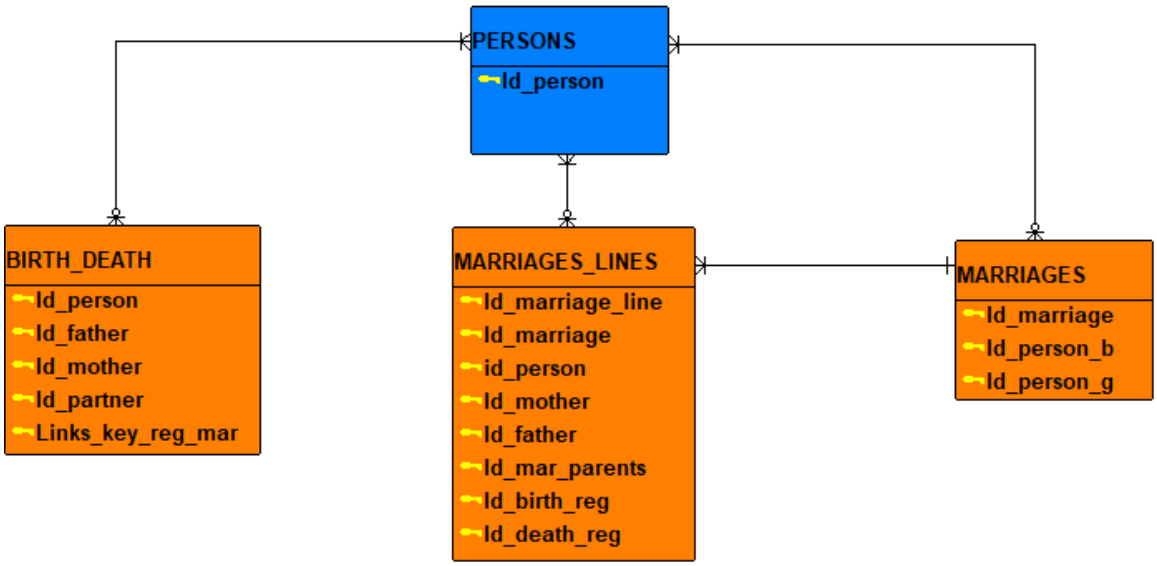
10 For the sake of clarity, the names of some keys have been changed in comparison with the published documentation (Links_key_reg_mar in BIRTH_DEATH became Id_mar_parents and Id_mar_parents in MARRIAGE_LINES became Id_mar_parents_intern).

The table BIRTH_DEATH establishes the link between a birth and a death certificate and includes all information from these certificates, such as place and time of death, occupational titles, etc. The table includes not only the linked but also the unlinked birth and death certificates. The key Id_partner refers to the (last) spouse in the death certificate.

The data of the marriage certificates is included in two tables: MARRIAGES and MARRIAGES_LINES. MARRIAGES includes all information about the marriage itself, including the identifiers of the bride and the groom. MARRIAGES_LINES contains the personal data of the bride or the groom and the data about their parents. The key Id_Marriage identifies each marriage. So, each marriage produces one record in MARRIAGES and two (bride and groom) in MARRIAGE_LINES (see also Section 6.2.1). The link with the marriage of the parents is defined through the (internal) key Id_mar_parents_intern which refers to Id_marriage.

Births and deaths belong to families, this relationship is represented through the key Id_mar_parents referring to Id_Marriage in the table MARRIAGE_LINES [and the table MARRIAGES]. Each person may marry not at all or once or more, having the corresponding number of records in MARRIAGE_LINES linked by way of the key Id_person.

Figure 9 *Table structure of interlinked civil certificates and unified personal information*



All unique persons are included in the table PERSONS, which was constructed by including appearances of persons from certificates in the following sequence:

1 All persons (birth, mother and father) from the birth certificates;
2 All persons (death, mother and father) from the death certificates that were *not* linked with a birth certificate;
3 The last partner from the death certificates;
4 All persons from the marriage certificates (bride, groom, mothers and fathers) that are not known (= are not linked) from the birth and/or death certificates.

All persons included in PERSONS are linked with the underlying tables in Figure 9 with the identifiers id_person, id_mother, id_father, id_partner, id_person_b and id_person_g.

The first three steps are rather straightforward. However, adding of the marriages is more complicated, and an update of the identifying keys in BIRTH_DEATH is needed after adding the marriage certificates. So, it is not a question of simply adding persons. In the following section we will explain the construction of unique persons out of all their appearances in the several certificates.

6.3.2 IDENTIFICATION PROCESS OF UNIQUE PERSONS

For a complete construction of unique persons, we need five types of links:

a) Marriages and parental marriages (pedigrees)

The linking process contained only one step: The marriage certificates were linked on the basis of a link between pairs: the parents of a bride or a groom with a bride/groom couple, see Section 6.2.1 for the details of this matching. Links could not be established in case the parental marriage originates from the period before 1812.

b) Shadow marriages

In case a birth certificate was not linked with a parental marriage certificate, 'shadow marriages' were created. This matching works more or less in the same way as the one creating pedigrees. But here, the parents mentioned in different birth certificates are linked to form parental environments. So, the linking is based on pairs of two persons and these marriages are also bound within an acceptable time range. Shadow marriages of parents may go back far into the 18th century.

c) Births and Deaths

The linking process connecting the birth and death certificates, forming basic lifelines, contains two different approaches: a) linking on the basis of three persons: child, mother and father and b) linking of two persons: child and mother. The second option principally implies that no father is known. Of course, there will be cases of linked certificates in which fathers show up, who did not match in the first approach. We did not use this information because including these fathers could conflict with positively matched fathers from marriage certificates. Since the data about fathers is included in BIRTH_DEATH, a user can determine whether a father is known or unknown.

d) Births/Deaths and the Marriages of the parents

The link of a birth or death with the marriage certificates of their parents was created from the point of view of birth and death. First, we tried to match each birth to a marriage certificate. Next, we repeated the same operation on all deaths that were not linked with a birth certificate, including infants recorded in the death register but not the birth register, most of whom died shortly after birth. This procedure implies that in case the link from the birth certificate would provide different results than the death certificate, the former was given automatic priority. This choice was based on the legal requirement that brides- and grooms-to be had to show a birth certificate before a marriage could take place. This means that birth certificates were used to fill in the personal information on a marriage certificate. All matching was based on linking these two pairs: bride & groom and mother & father. In this way we created families, so a family is defined as all persons linked with the same parental marriage certificate. This implies that persons whose birth and death certificates were not linked could appear as siblings in the family tree.

e) Births/Deaths and Marriages

The link of a birth certificate with his or her own marriage certificate was made from the point of view of the bride and groom lines (the table MARRIAGE_LINES). This was done, because a person is only born once, but may marry more than once. The linking proceeded in two steps: a) Linking on the basis of three persons: child (bride or groom), mother and father, b) linking on the basis of two persons: child (bride or groom) and the mother to include also brides and grooms born out of wedlock. The linking of a death certificate with marriages is comparable, except that here two additional approaches can be used: matching on the basis of four persons: the deceased, mother, father and partner, and matching on the basis of the deceased and his/her partner.

After the linkage of the births and deaths with their marriage certificates, it was possible to add more links between the birth and death certificates. Parents were often not mentioned in a death certificate, thus making it impossible to link them with a birth certificate. However, when the partner of the deceased could be used as a second person in the linkage with the marriage certificate, matches between a death and marriage certificate were made. In combination with a link between a birth and a marriage certificate, the link between a birth and death certificate could be deduced (B links M, M links D, => B links D).

All these matching operations created multiple links (so-called overlinks). Certificates with multiple links were flagged and not linked, unless the composed Levenshtein values showed significant differences (e.g., 1 compared with 7 or 8). In that case a choice was made for the match with the lowest Levenshtein value and flagged as such.

6.3.3 SYNCHRONIZATION OF THE PERSON IDENTIFIERS

All persons from the civil certificates entered the dataset with their own identifiers which are always kept in the release as well. However, it may occur that the linkage information tells us for example that the person number 1203048 in the birth certificates is the same person as the person with number 42382209 in the marriage certificates. Then, we need to synchronize the identification numbers and create a kind of global identifier, Id, which is for each release. Synchronization of the identifying keys influences all generations. A child in a birth certificate may be a bride in a marriage certificate and a parent in the next generation of birth, death and marriage certificates. So, the link between the generations is made through the marriage certificates. But that also implies that the synchronization of the identifiers must start with the marriage certificates to make sure that the same parents in the birth or death certificates end up with the same identifiers. The matching has been done in the form of pedigrees, going backwards while we need life courses and families that start at the beginning of their life cycle. This implies, that the pedigrees need to be 'toppled' into family tree systems.

Synchronizing persons from the marriage certificates

To make the synchronization process feasible two extra steps are necessary: a) the pedigree system has to be transformed into a family tree system, and b) remarried children need to be identified.

The conversion from a pedigree system into a family tree system was done by way of the following steps:

1 Define the level of the family tree as generation "1" if there is no link with a parental marriage;

2 Define the family tree as generation "2" if there is a link with a previous marriage with generation level "1";

3 Repeat step 2 up to generation level 7 or more, which is the limit and occurs only four times in the Zeeland dataset.

This procedure looks more straightforward than it is. One needs to realize that although a bride or a groom has only one parental couple, they have two grandparental couples, four great-grandparental couples etc. This implies that in numbering the generations, different levels will apply to a person depending on the path backwards. For example, in one marriage the bride may have generation level 3 linking along the father line back to the grandparents and level 2 in case the mother line shows no further links backwards. This implies that at the second level we have four marriage lines to follow: the mother line (bride -> mother), the father line (groom -> father), the diverting mother line (bride -> father) and the diverting father line (groom -> mother). In the table MARRIAGE_LINES the levels of the first two lines are represented in the field Family_tree_level; the last two in the field Family_tree_level_A. For the third level we could have doubled this system again, but we abstained from this, not wanting to make the system too complicated.

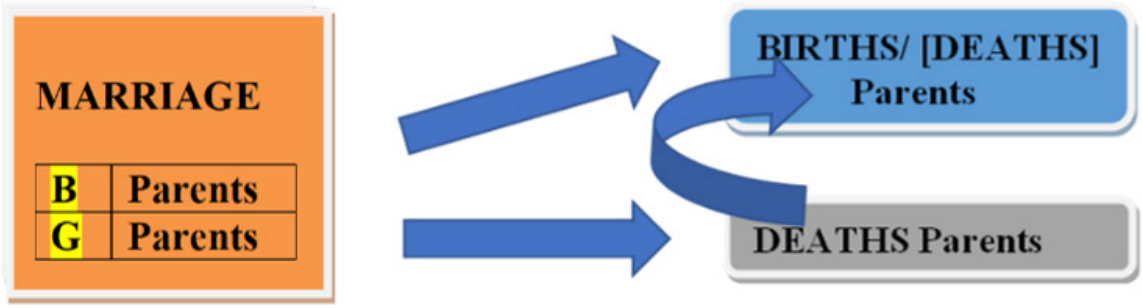
Another issue is that remarried children need to be identified. There are brides and grooms who marry more than once and initially have different identification numbers. If they are not identified as one and the same person they will be seen as siblings in the dataset. At generation level 2 and higher remarried persons are identified within the context of the parental marriage, in which they can be matched through their first name. In case of equal first names, the identification number is synchronized. At generation level 1 we created 'shadow marriages' (see Section 6.3.2, but now on the basis of marriage certificates) on which basis we could match on first names and synchronize the identifiers of remarried persons.

In a final step the identifiers of the bride and groom of generation 1 replace the identifiers of the parents of generation 2, etc. up until generation 5 replaces the identifiers of the parents of the 6th generation.

Synchronizing persons from the birth and death certificates

In first instance, the identifiers of the deceased, mother and father of the death certificate were replaced by those of the birth certificate by way of the linkage itself. All persons from the death certificates that were not linked kept their original identifiers. Eventual partners of the deceased always kept their own number because they were not included in the birth certificates.

Figure 10 *Synchronization scheme parents' birth and death certificates*



In a second step the identifiers of the bride and groom of the parental marriage certificate were used to replace the respective identifiers of the parents in the birth and death certificates (see Figure 10). On the basis of the linkage between the births/deaths and their own marriage certificates the identifiers of the births and the deaths in the table BIRTH_DEATH were replaced by the ones of the bride and the groom. In case of parents from generation level 1 (whose own marriage certificate has not been found), the identifiers of these parents were equalized with those from the birth or death certificates.

In a final step, on the basis of the links between the death and marriage certificates the partners in the death certificates were synchronized with the identifiers in the marriage certificate. In case of multiple marriages, the death certificates usually include the last partner. Since only those death certificates are used that have been linked with the marriage certificates including a link with the partner, possible false links are logically impossible.

6.3.4 RESULTS

Three regions were matched separately: the combination of the provinces of Groningen and Drenthe as well as Limburg and Zeeland. Table 10 presents the number of included certificates and the results of the matching and identifying process for each region. Almost 6 million certificates are included which stands for about 20 million person appearances. Matching was done along the lines of the releases as discussed in Sections 6.1. and 6.2 with the exception that the matching was limited to the certificates from the specific region.

All in all, we identified just under 8 million different persons in these three integrated systems. Some of these persons are combined into 407.435 families. With an average number of 4.29 children and 2 parents, which means that about 2.56 million unique persons are involved in a family structure defined as a married couple with at least one known child. Given the total of 7.99 million, it seems that 5.5 million persons are lacking. These are the persons that are included in the certificates that could not be linked. But this total of 5.5 million is seriously exaggerated because identical persons that are not linked within a family structure are not identified as such and are counted more than once.

The combination of the two provinces of Groningen and Drenthe has about 50% more indexed certificates than the other two provinces Zeeland and Limburg. In terms of linkage results between birth and death certificates there is no big difference between Groningen/Drenthe and Zeeland. Limburg has a much lower result, despite the relatively high number of included death certificates. The main reason for this result is the shape of the province having a much longer border with other provinces and Belgium and Germany than the other ones. This implies that we could expect that there was more in- and outmigration. And during the first half of the 20th century the coal area of Limburg attracted many persons from outside the province (Langeweg, 2012).

Table 10 *Integrated linking results for three areas: Groningen/Drenthe, Limburg and Zeeland*

	Groningen/ Drenthe	Zeeland	Limburg	Total
Number of included birth certificates	1,061,614	698,361	761,857	2,521,832
Number of included death certificates	1,043,926	650,728	843,413	2,538,067
Number of included marriage certificates	365,672	193,793	212,399	771,864
Number of linked birth/death	567,333	368,517	326,818	1,262,668
% of included births	53.4	52.8	42.9	50.1
% of included deaths	54.3	56.6	38.7	49.7
Number of linked brides and grooms with parental marriages	465,650	227,604	169,538	862,792
% of linked marriage lines	63.8	58.7	39.9	55.9
Number of linked births with parental marriage	835,081	511,647	402,933	1,749,661
% of included births	78.7	73.3	52.9	69.4
Number of linked deaths with parental marriage	585,120	351,058	305,468	1,241,646
% of included deaths	56.1	53.9	36.2	48.9
Number of unique persons	2,891,468	1,939,954	3,160,298	7,991,720
Number of families	201,882	106,082	99,471	407,435
Average number of children/family	4.14	4.82	4.05	4.29
Number of three generation pedigrees	246,855	109,847	83,293	439,995
Number of four or more generation pedigrees	153,555	53,442	34,170	241,167

Explanation: Number of death certificates Zeeland and Limburg include lifeless reported certificates (respectively n=40,786 and n=52,068). Lifeless reported cases are linked with marriage certificate of the parents (but are lacking a birth certificate). Families are defined as marriages with at least one identified child. Three-generation structures are pedigrees with at least two linked marriage certificates; a distinction has been made between a) the first three generations and b) 'doubling structures' in case of more than three generations (a sixth generation family structure contains four overlapping three-generation structures).

The results for links with the marriages show the same pattern of a relatively bad performance of Limburg. Especially for the pedigrees and links of the births with the parental marriages Groningen/ Drenthe also shows a better result than Zeeland with a positive difference of about 5%. The average number of children per family is in line with what one would expect for the 19th century which was on average 4.7 children per family (van den Berg et al., 2021; also in line with Dribe et al. (2017) and Engelen (2009; p. 174) who came to the same result on the basis of the census outcomes).

6.3.5 SOME REMARKS ABOUT THE RESULTS

Matching certificates to reconstruct life courses, pedigrees, family trees, families, etc. from a limited area and time period, implies several 'data leaks' and inconsistencies, mainly because of the following reasons:

1. Persons could have emigrated to another area;
2. Persons could immigrate from another area;
3. Not all certificates are matched because of insufficient identifying information;
4. Certificates are matched in an ambiguous way because the identifying information is not accurate enough;
5. Persons cannot be matched because the certificate to be matched does not exist (before 1812) or is not indexed yet;
6. Bugs in the matching software;
7. Inconsistencies in one generation may have consequences for the family trees that have been constructed.

In the releases, several fields are included that describe the way the data have been matched. These flags may be used to make selections from the dataset to test how robust the outcomes of the statistical analyses are. Ultimately, it is the researcher who is responsible for the way the data are used.

Since the marriage certificates are linked with both birth and death certificates, it is possible to check on triangle problems. It turns out that many death certificates are linked with marriage certificates and not with birth certificates, where these birth certificates are linked with the same marriage certificate. In the case of death certificates of persons born before 1850, in which names of parents are often of poor quality if mentioned at all, they could easily be linked to their own marriage certificates on the basis of the names of partners.

Reshaping the pedigrees into family trees, is a kind of toppling of the pedigree system. It is an essential step because most of the generational analysis should be done from the perspective of the beginning of a family line (not of the end), especially when the system is to be extended with other certificates (e.g., of the children of the couples). There is also a practical problem: because each line in the pedigree will result in a different generation level for the last generation, one cannot simply fix the generation level for one marriage line in a marriage certificate. The more generations are involved in such a system the more complicated this will become.

6.3.6 DATASETS FOR ANALYSIS

The *integrated* datasets of births, marriages and deaths with created families and multigenerational links are not sufficient to be immediately usable for research. For the Zeeland release two types of datasets suitable for statistical analysis were created. The first one was a rectangular-type structure that was constructed within the context of the project Genes, Germs and Resources (https://www.nwo.nl/en/projects/360-53-180-0; Mourits et al., 2020). The second was a conversion of the format of the Zeeland release into the format of the Intermediate Data Structure (Alter & Mandemakers, 2014).

The database which was constructed for the project Genes, Germs and Resources (LINKS-gen; see Mourits et al., 2020), served several goals. The first one was to create more explicit family links than were provided in the Zeeland release and to improve and extend dates of birth, last observation and other variables. The second one was to reformat the design into a so-called pedigree format.

Through better integration of unlinked newborn and deceased persons that were linked to the same parents, a more consistent dataset could be created. Also, several data improvements were applied. For example, the conversion of ages at a specific moment into birth ranges, fields were created for up to five marriages and newborns who were reported dead on registration lacked a date of birth which was included as the date of death. Other improvements made twins explicit, added dates of last observation and flagged complete cases. To retain all data and relational information of a person on one record the database was restructured into a so-called pedigree structure. This format structures the data in such a way that each record includes the identification number of a person, the identification numbers of his or her parents (if known), the sex and all other variables. Families and familial relationships are defined through the father and mother. Through restructuring the dataset, hidden links between persons were also made explicit. By this operation a new structure was created, making it also much easier for researchers to select their case for a specific analysis (Mourits et al., 2020).

The conversion of the Zeeland release into the IDS-format was relatively easy, mainly because the number of variables, or types in IDS-grammar, is quite limited (Mandemakers & Laan, 2017, IDS version). In the INDIVIDUAL table we have, sex, occupations, and the date and location of birth, marriage and death. Relations that are established in the INDIV_INDIV table are those between children and parents (including in-law relationships) and marriage couples. The nature and location of the certificates were used as the lowest level in the contextual system. An alternative could have been the use of the "Union" concept as lowest level as Klancher Merchant and Alter (2017) have done, but this approach was not necessary given the nature of the research for which the IDS dataset was developed. It concerned research into intergenerational effects of infant mortality in which four other databases were involved. All were structured into the IDS and reshaped in datasets ready for statistical analysis by a common Stata script (Quaranta, 2018; van Dijk & Mandemakers, 2018).

7 SUMMARY AND CONCLUSIONS

In this paper we explained the construction of the LINKS database using the indices of the civil certificates as collected by the Dutch Family Center and published on the website WieWasWie. Presently, over 40 million certificates and 120 million appearances of persons have been included in this index by hundreds of volunteers working over the last twenty years to create an electronic index of the civil certificates as soon as they become public.

Two matching systems have been developed within the HSNDB environment. The first one is a query system based on SQL queries selecting the data from the MySQL-database in which the matching queries are only part of a wider environment directed at the standardizing, cleaning, enriching and outputting of the LINKS data. Using the Zeeland marriage certificates as an example, we showed the excellent quality of the data material in general. The difference between the result of the exact matching and the least restricted, yet acceptable alternative in matching was only 38,338 matches (23.5%). Moreover, 80% of all matches were exact matches, thanks to the legal structure of civil registration that standardized the names and a legal and administrative structure that kept the birth name of the females alive with and after marriage.

However, on large datasets these matching queries are slow and the end user has no direct influence on the matching alternatives, which are set by the database manager unless special requests are made. For these reasons a second matching system, burgerLinker, was developed, based on knowledge graphs which can be run independently of the LINKS environment.

In both cases, matching is a first step for the creation of a linked dataset that can be used for research. We explained the construction of several relatively recent data releases. On a national scale, we released a pedigree system based on all marriage certificates and a dataset in which the births are linked with the marriages of their parents, forming families. On a regional level we created three separate releases for the provinces of Zeeland, Limburg and Groningen/Drenthe. Here, we combined birth, death and marriage certificates to create three-generation families. One of the issues for which we found a solution was the identification of unique persons in this three-generation system.

The LINKS project started in 2010. Since then, over 40 releases have been produced resulting in over 50 publications including several dissertations (Mandemakers & Kok, 2020). We expect that in the future, more releases will be made by HSNDB or by individual users of burgerLinker. Increasingly, researchers are using burgerLinker to link large collections of individual-level data. Military, inheritance tax and income tax registers have all proven to be important sources for future research. LINKS continues on the path set out by previous generations of historical demographers, creating new options for generations to come.

REFERENCES

Alter, G., & Mandemakers, K. (2014). The Intermediate Data Structure (IDS) for longitudinal historical microdata, version 4. *Historical Life Course Studies, 1*, 1–26. doi: 10.51964/hlcs9290

Bloothooft, G., van Boheemen, J., & Schraagen, M. (2016). Historical life cycle reconstruction by indexing. *Workshop Data Linkage: Techniques, Challenges and Applications at Isaac Newton Institute for Mathematics*. Cambridge UK. Retrieved from https://www.gerritbloothooft.nl/Publications/Cambridge_Bloothooft_etal.pdf

Bloothooft, G., & Onland, D. (2016). Multiple first names in the Netherlands (1760–2014). *Names, 64*(1), 3–18. doi: 10.1080/00277738.2016.1118860

Bloothooft, G., Onland, D., Reynaert, M., Depuydt, K., Schoonheim, T., Fannee, M., & Noordzij, J. (2020). *NAMES Corpus*. Retrieved from https://taalmaterialen.ivdnt.org/download/names-corpus/

Bloothooft, G., & Schraagen, M. (2015). Learning name variants from inexact high-confidence matches. In: G. Bloothooft, P. Christen, K. Mandemakers, & M. Schraagen (Eds.). *Population Reconstruction* (pp. 87–110). Cham: Springer. doi: 10.1007/978-3-319-19884-2_4

Bourdieu, J., Kesztenbaum, L., Postel-Vinay, G., & Tovey, J. (2014). The TRA project, a historical matrix. *Population (English Edition)*, *69*(2), 191–220. doi: 10.3917/popu.1402.0217

Christen, P., Vatsalan, D., & Fu, Z. (2015). Advanced record linkage methods and privacy aspects for population reconstruction — A survey and case studies. In: G. Bloothooft, P. Christen, K. Mandemakers, & M. Schraagen (Eds.). *Population Reconstruction* (pp. 87–110). Cham: Springer. doi: 10.1007/978-3-319-19884-2_5

Dillon, L., Amorevieta-Gentil, M., Caron, M., Lewis, C., Guay-Giroux, A., Desjardins, B., & Gagnon, A. (2018). The *Programme de recherche en démographie historique*: Past, present and future developments in family reconstitution. *The History of the Family, 23*(1), 20–53. doi: 10.1080/1081602X.2016.1222501

Dribe, M., Breschi, M., Gagnon, A., Gauvreau, D., Hanson, H. A., Maloney, Th. N., Mazzoni, S., Molitoris, J., Pozzi, L., Smith, K. R., & Vézina, H. (2017). Socio-economic status and fertility decline: Insights from historical transitions in Europe and North America. *Population Studies, 71*(1), 3–21. doi: 10.1080/00324728.2016.1253857

Dupâquier, J., Kessler, D. (Eds.). (1992). *La société française au XIXe siècle. Tradition, transition, transformations* [French society in the XIX century. Tradition, transition, transformation]. Paris: Fayard.

Engelen, Th. (2009). *Van 2 miljoen naar 16 miljoen mensen. Demografie van Nederland, 1800–nu* [From 2 million to 16 million people. Demography of the Netherlands, 1800–present]. Amsterdam: Boom. Retrieved from http://hdl.handle.net/2066/78820

Gerritzen, D. (1998). Voornamen in Zeeland [Firstnames in Sealand]. In: K. Mandemakers, O. Hoogerhuis, & A. de Klerk (Eds.). Over Zeeuwse mensen. Demografische en sociale ontwikkelingen in Zeeland in de negentiende en twintigste eeuw [Special issue]. *Zeeland, 7*(3), 104–115.

Goeken, R., Huynh, L., Lynch, T.A., & Vick, R. (2011). New methods of census record linking. *Historical Methods, 44*(1), 7–14. doi: 10.1080/01615440.2010.517152

Henry, L., & Fleury, M. (1956). *Manuel de dépouillement et d'exploitation de l'état civil ancien* [Manual to analyse and exploit the ancient civil registration]. Paris: INED.

Huijsmans, D. P. (2020). *HSN Gazetteer* [Data set]. Retrieved from https://hdl.handle.net/10622/ZDT2DJ

Klancher Merchant, E., & Alter, G. (2017). IDS Transposer: A users guide. *Historical Life Course Studies*, *4*, 59–96. doi: 10.51964/hlcs9339

Kok, J. (1991). *Langs verboden wegen. De achtergronden van buitenechtelijke geboorten in Noord-Holland 1812–1914* [Along forbidden roads. Background of illegitimate births in North-Holland 1812–1914]. Hilversum: Verloren.

Lambert, P. S., Zijdeman, R. L., van Leeuwen, M. H. D., Maas, I., & Prandy, K. (2013). The construction of HISCAM: A stratification scale based on social interactions for historical comparative research. *Historical Methods, 46*(2), 77–89. doi: 10.1080/01615440.2012.715569

Langeweg, S. (2012). Werving, herkomst en binding van mijnwerkers [Recruitment, origin and binding of miners]. In: Knotter, A. (Ed.), *Mijnwerkers in Limburg. Een sociale geschiedenis* (pp. 100–138). Nijmegen: Vantilt.

Mandemakers, K. (2000). The Netherlands. Historical Sample of the Netherlands. In: P. Kelly Hall, R. McCaa, & G. Thorvaldsen (Eds.), *Handbook of international historical microdata for population research* (pp. 149–178). Minneapolis: Minnesota Population Center. Retrieved from https://international.ipums.org/international/resources/microdata_handbook/1_10_netherlands_ch11.pdf

Mandemakers, K. (2023, January 20). *"You really got me". Ontwikkeling en toekomst van historische databestanden met microdata* [Development and future of historical databases with microdata] (Valedictory speech]. Erasmus University, Rotterdam, the Netherlands. doi: 10.25397/eur.23256467

Mandemakers, K., & Dillon, L. (2004). Best practices with large databases on historical populations. *Historical Methods, 37*(1), 34–38. doi: 10.3200/HMTS.37.1.34-38

Mandemakers, K., Hornix, J., Mourits, R. J., Muurling, S., Boter, C., van Dijk, I. K., Maas, I., Van de Putte, B., Zijdeman, R. L., Lambert, P., van Leeuwen, M. H. D., van Poppel, F., & Miles, A. (2020). *HSN standardized, HISCO-coded and classified occupational titles, HSN release 2020.02* [Data set]. Retrieved from https://hdl.handle.net/10622/88ZXD8

Mandemakers, K., & Kok, J. (2020). Dutch lives. The Historical Sample of the Netherlands (1987–): Development and research. *Historical Life Course Studies, 9*, 69–113. doi: 10.51964/hlcs9298

Mandemakers, K., & Laan, F. (2017). *LINKS Zeeland linked dataset (Marriages, births and deaths), province of Zeeland, Release 2017_02, including IDS format* [Data set].

Mandemakers, K., & Laan, F. (2018). *LINKS Groningen-Drenthe linked sataset (Marriages, births and deaths), Release 2018_01* [Data set].

Mandemakers, K., & Laan, F. (2019). *LINKS dataset WieWasWie Limburg, linked civil certificates (Births, deaths and marriages), Release 2019.02* [Data set].

Mandemakers, K., & Laan, F. (2020a). *LINKS dataset linked births and marriage certificates parents, the Netherlands, Release 2020.01* [Data set].

Mandemakers, K., & Laan, F. (2020b). *LINKS dataset linked marriages, the Netherlands, 1796–1943, Release 2020.03 (n=4,158,388), Also a version including first names of bride/groom and parents, Release 2020.03_f)* [Data set].

Mourits, R. J., Boonstra, O., Knippenberg, H.,Hofstee, E. W., & Zijdeman, R. L. (2016). *Historische Database Nederlandse Gemeenten* [Data set]. Retrieved from https://hdl.handle.net/10622/RPBVK4

Mourits, R. J., van Dijk, I. K., & Mandemakers, K. (2020). From matched certificates to related persons. *Historical Life Course Studies*, *9*, 49–68. doi: 10.51964/hlcs9310

Nault, F., & Desjardins, B. (1989). Computers and historical demography: The reconstitution of the early Québec population. In: P. Denley, S. Fogelvik, & Ch. Harvey. *History and computing, II*. (pp. 143–148). Manchester: Manchester University Press.

Quaranta, L. (2018). Program for studying intergenerational transmissions in infant mortality using the Intermediate Data Structure (IDS). *Historical Life Course Studies*, *7*, 11–27. doi: 10.51964/hlcs9287

Oosten, M. (2008). *Verleden namen. Familieverbanden uit Genlias-data* [Names from the past. Family structures from GENLIAS data] (Unpublished master's thesis). LIACS and IISG, Leiden.

Raad, J., Mourits, R. J., Rijpma, A., Schalk, R., Zijdeman, R. L., Mandemakers, K., & Meroño-Peñuela, A. (2020). Linking Dutch civil certificates. *3rd Workshop on Humanities in the Semantic Web (WHiSe) conference proceedings*. Heraklion, Greece. Retrieved from https://ceur-ws.org/Vol-2695/paper6.pdf

Schraagen, M. (2014). *Aspects of record linkage (*PhD thesis). Leiden University. Retrieved from http://hdl.handle.net/1887/29716

Schulz, K. U., & Mihov, S. (2002). Fast string correction with Levenshtein automata. *International Journal on Document Analysis and Recognition, 5*(1), 67–85. doi: 10.1007/s10032-002-0082-8

Séguy, I. (2001). *La population de la France de 1670 à 1829: l'Enquête Louis Henry et ses données* [The population of France from 1670 to 1829: The Louis Henry survey and its data]. Paris: INED.

Séguy, I. (2016). The French school of historical demography (1950–2000). In: A. Fauve-Chamoux, I. Bolovan, & S. Sogner (Eds.). *A global history of historical demography. Half a century of interdisciplinarity* (pp. 257–276). Bern: Peter Lang.

Song, X., & Campbell, C. D. (2017). Genealogical microdata and their significance for social science, *Annual Review of Sociology, 43,* 75–99. doi: 10.1146%2Fannurev-soc-073014-112157

Stead, W. W., Hammond, W. E., & Straube, M. J. (1982, November). A chartless record — Is it adequate? *Proceedings of the Annual Symposium on Computer Application in Medical Care*, 89–94. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2580254/

van Boheemen, J. (2016). *Assembling the pages. A sorting-based approach to historical record linkage.* (Bachelor's thesis). Universiteit Utrecht. Retrieved from https://studenttheses.uu.nl/handle/20.500.12932/23905

van den Berg, N., van Dijk, I. K., Mourits, R. J., Slagboom, P. E., Janssens, A. A. P. O., & Mandemakers, K. (2021). Families in comparison: An individual-level comparison of life-course and family reconstructions between population and vital event registers. *Population Studies, 75*(1), 91–110. doi: 10.1080/00324728.2020.1718186

van der Bie, R. J.(1995). "*Een doorlopende groote roes*". *De economische ontwikkeling van Nederland 1913–1921* ["A continuous big Rush". The economic development of the Netherlands, 1913–1921]. Amsterdam: Tinbergen Institute research Series.

van der Bie, R. J., & Smits, J. P. (Eds.). (2000). *Tweehonderd jaar statistiek in tijdreeksen, 1800–1999* [Two hundred year statistics in time series, 1800–1999]. Amsterdam: Stichting Beheer IISG.

van Dijk, I. K., & Mandemakers, K. (2018). Like mother, like daughter. Intergenerational transmission of infant mortality clustering in Zeeland, the Netherlands, 1833–1912. *Historical Life Course Studies*, *7*, 28–46. doi: 10.51964/hlcs9286

van Galen, C. W. (2019). Creating an audience: Experiences from the Surinamese slave registers crowdsourcing project. *Historical Methods, 52*(3), 178–194. doi: 10.1080/01615440.2019.1590268

van Galen, C. W., Mourits. R. J., Rosenbaum-Feldbrügge, M., A.B., M., Janssen, J., Quanjer, B., van Oort, Th., & Kok, J. (forthcoming). Slavery in Suriname: A reconstruction of life courses, 1830–1863. *Historical Life Course Studies.*

van Leeuwen, M. H. D., & Maas, I. (2011). *HISCLASS. A Historical International Social Class Scheme*. Leuven: Leuven University Press.

van Leeuwen, M. H. D., Maas, I., & Miles, A. (2002). *HISCO. Historical International Standard Classification of Occupations.* Leuven: Leuven University Press.

van Zanden, J. L., & Griffiths, R. T. (1989). *Economische geschiedenis van Nederland in de 20e eeuw* [Economic history of the Netherlands in the 20th century]. Utrecht: Uitgeverij Het Spectrum.

Vézina, H., & Bournival, J.-S. (2020). An overview of the BALSAC population batabase. Past developments, current state and future prospects. *Historical Life Course Studies*, *9*, 114–129. doi: 10.51964/hlcs9299

Vulsma, R. F. (1988). *Burgerlijke stand en bevolkingsregister* [Civil registration and population register]. 's-Gravenhage: Centraal Bureau voor de Genealogie.

Wrigley, E. A., Davies, R. S., Oeppen, J. E., & Schofield, R.S. (1997). *English population history from family reconstitution 1580–1837.* Cambridge: Cambridge University Press. doi: 10.1017/CBO9780511660344

APPENDIX A OVERVIEW OF ERROR REPORTS

This appendix is an overview of all types of error reporting (logic, completeness and errors). The field Type provides a reference to the "error message". The field Class consists of three values: "FT" for Fout ('error'), which means that some error has taken place; "WA" for Waarschuwing ('warning') which means that a value could be wrong but there is a chance that it is not a problem, which is typically for values not yet standardized; "NB" stands for a value that needs to be checked, but which is not necessarily a mistake (depends on the archive). The field Content provides the error message that is reported for the error type, e.g., type 41 returns a message like "Non authorized occupation: shoematter", "No standard; standard_code= x". The messages are delivered with information identifying the original source.

Type	Class	Content
1	FT	Double entry of the original registration
2	FT	Registration without a registration date
3	FT	Registration without defining one or more roles
4	FT	One of more than two entries with same Registration Details (Type, Location, Year and Sequence)
21	WA	Non authorized religion: No standard; standard_code= "x"
23	WA	Invalid religion: No standard; standard_code= "n"
25	WA	Invalid religion: Standard present; standard_code= "u"
29	FT	Standard_code not valid
31	WA	Non authorized gender: No standard; standard_code= "x"
33	WA	Invalid gender: No standard; standard_code= "n"
35	WA	Invalid gender: Standard present; standard_code= "u"
39	FT	Standard_code not valid
41	WA	Non authorized occupation: No standard; standard_code= "x"
43	WA	Invalid occupation: No standard; standard_code= "n"
45	WA	Invalid occupation: Standard present; standard_code= "u"
49	FT	Standard_code not valid
51	WA	Non authorized registratietype: No standard; standard_code= "x"
53	WA	Invalid registration type: No standard; standard_code= "n"
55	WA	Invalid registration type: Standard present; standard_code= "u"
59	FT	Standard_code not valid
61	WA	Non authorized gender or status: No standard; standard_code= "x"
63	WA	Invalid gender or status: No standard; standard_code= "n"
65	WA	Invalid gender or status: Standard present; standard_code= "u"
68	FT	Civil Status: suggests a gender wich is inconsistent with the gender of this person
69	FT	Standard_code not valid
71	WA	Non authorized suffix: No standard; standard_code= "x"
73	WA	Invalid suffix: No standard; standard_code= "n"
75	WA	Invalid suffix: Standard present; standard_code= "u"
79	WA	Standard_code not valid
81	WA	Non authorized title or prefix: No standard; standard_code= "x"
83	WA	Invalid title or prefix: No standard; standard_code= "n"
85	WA	Invalid title or prefix: standaard aanwezig; standard_code= "u"
89	FT	Standard_code not valid
91	WA	Non authorized location: No standard; standard_code= "x"
93	WA	Invalid location: No standard; standard_code= "n"

Type	Class	Content
95	WA	Invalid location: Standard present; standard_code= "u"
99	FT	Standard_code not valid
102	NB	Restant opmerking:
103	NB	Not valid combination of role: [rol] and date: [date]
104	NB	Invalid function code: from table ref_date_minmax
105	FT	Could not find all info in reference table "ref_date_minmax" to calculate minmax: <>
106	FT	Function minMaxMainAge cannot find record in ref_date_minmax
107	FT	Duplicate role within one registration
111	NB	Sequence number not present
112	NB	Sequence number is not numeric:
113	FT	Sequence number occured twice:
114	FT	Missing Sequence number (previous number is lacking):
115	FT	There are more than 100 records for a specific source per year per municipality but december is missing; 100:12 rule
141	WA	Non authorized role: No standard; standard_code= "x"
142	FT	Invalid role: In combination with registration type
143	WA	Invalid role: No standard; standard_code= "n"
145	WA	Invalid role: Standard present; standard_code= "u"
149	FT	Standard_code not valid
201	FT	Constructed (from events) registration date is invalid:
202	WA	Date of registration based only on the year of the registration
203	WA	Invalid registration_date, but reconstructable
204	FT	Components registration date are are invalid
205	FT	No Registration date and registration_date is not constructable
206	WA	Registration date and registration elements unequal
211	FT	Invalid Birth date:
221	FT	Invalid Marriage date:
231	FT	Invalid Death date:
241	FT	Age in days is out of range (0-99):
242	FT	Age in weeks is out of range (0-49):
243	FT	Age in months is out of range (0-49):
244	FT	Age in years is out of range (0-114):
251	WA	Non authorized literal age: No standard; standard_code= "x"
253	WA	Invalid literal age: No standard; standard_code= "n"
255	WA	Invalid literal_age: Standard present; standard_code= "u"
259	FT	Standard_code not valid
261	WA	Content Age_literal: conflicts with Age_year:
262	WA	Content Age_literal: conflicts with Age_month:
263	WA	Content Age_literal: conflicts with Age_week:
264	WA	Content Age_literal: conflicts with Age_day
265	FT	Content Age_year: where role is parent or partner
266	FT	Minimum Age: larger than Maximum Age:
267	FT	Content Age_literal: conflicts with Role is "Kind"
271	FT	Missing newborn in birth registration

Type	Class	Content
272	FT	Missing bride in marriage registration
273	FT	Missing groom in marriage registration
274	FT	Missing deceased in death registration
281	WA	More than one newborn in birth registration
282	WA	More than one bride in marriage registration
283	WA	More than one groom in marriage registration
284	WA	More than one deceased in birth registration
1000	FT	Invalid familie name: standard present; standard_code= "u"
1001	FT	Person has no family name
1002	WA	Family name: uncleaned familyname does not exists in ref_file
1003	FT	Family name: contains two or more serried spaces (automatically corrected)
1004	FT	Family name: contains invalid character (automatically corrected)
1005	FT	Invalid familie name: No standard; standard_code= "n"
1006	FT	Invalid family name: contains suffix
1007	FT	Invalid family name: contains an alias
1008	FT	Invalid family name: contains prefix/title
1009	WA	Non authorized family name: No standard; standard_code= "x"
1010	FT	Standard_code not valid
1011	WA	Famillyname includes string without spaces
1012	FT	Famillyname includes prefix as a suffix
1100	FT	Invalid first name: Standard present; standard_code= "u"
1101	FT	Person has no first name
1104	FT	First name: contains invalid character (automatically corrected)
1105	FT	Invalid first name: No standard; standard_code= "n"
1106	FT	Invalid first name: contains suffix
1107	FT	Invalid first name: contains alias
1108	FT	Invalid first name: contains prefix/title
1109	WA	Non authorized first name: No standard; standard_code= "x"
1110	FT	Standard_code not valid
1111	WA	Firstname includes string without spaces
1112	WA	Firstname includes embedded capital:
1113	WA	Firstname includes embedded slash:
1114	WA	Firstname includes embedded HTML break:
1203	WA	Prefix, postfix or alias: contains two or more serried spaces (automatically corrected)
1204	WA	Prefix, postfix or alias: contains invalid character (automatically corrected)
1211	WA	Prefix, postfix or alias: includes string without spaces

APPENDIX B THE CIVIC REGISTRY MODEL

In this appendix the classes and schemas of the Civic Registry Model (CIV) showed in Figure 4 are described in a more formal way.

The CIV-model is composed of three parts:

1. Person (blue)

This part is only composed of the class schema:Person, representing the individuals described in the civil registries. An instance of this class must have a unique identifier (civ:personID), a first name (schema:givenName), and a last name (schema:familyName). All these properties are required for linking persons. In addition, for improving the accuracy and the speed of linking, adding the gender (schema:gender) of every individual is recommended.

2. Events (green)

We make a distinction between three different types of events: civ:Birth, civ:Marriage, and civ:Death. These three types of events are all sub-types of the general class civ:Event. Being sub-type of civ:Event means that these three classes inherit the properties of their general class, i.e., each instance of the class civ:Birth, civ:Marriage, and civ:Death can have the five relations that are associated with civ:Event. Out of these five relations, only two are required for linking: a unique event/registration identifier (civ:registrationID) and the date of an event (civ:eventDate). The remaining three optional relations are used for indicating the date of registration (civ:registrationDate), its location (civ:registrationLocation) and the event location (civ:eventLocation). In this model, a distinction is made between the date/location of an event and the date/location of its registration in the civil registries, as certain civil registrations can be produced in different dates and locations from where the life event happened.

In addition, each of these three types of event has different relations associated to it:

civ:Birth

An instance of this class can have the three properties: civ:newborn, civ:mother, and civ:father. For linking, all information regarding the newborn must be present in a birth event, in addition to at least one of their parents.

civ:Marriage

An instance of this class can also have the six properties: civ:bride, civ:motherBride, civ:fatherBride, civ:groom, civ:motherGroom, civ:fatherGroom. For linking, all information regarding the bride and groom must be present in a marriage event, in addition to at least one parent for each of the bride and groom.

civ:Death

An instance of this class can also have the four properties: civ:deceased, civ:partner, civ:mother civ:father. For linking, all information regarding the deceased must be present in a death event, in addition to at least one of their parents.

3. Location (yellow)

The final part describes the location where each life event has happened and the location where it was registered. In this part, information regarding the municipality, the province, the region, and the country can be available. This part is completely optional, as none of the information regarding the locations of the events and their registrations are used for linking.