# HISTORICAL LIFE COURSE STUDIES

VOLUME 1
2014

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 http://www.ehps-net.eu/journal.

**Editors: Koen Matthijs & Paul Puschmann**
**Family and Population Studies**
**KU Leuven, Belgium**
**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.

EUROPEAN SCIENCE FOUNDATION

**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.

EHPS NETWORK

HISTORICAL LIFE COURSE STUDIES
VOLUME 1 (2014), 1-26, published 26-05-2014

# The Intermediate Data Structure (IDS) for Longitudinal Historical Microdata, version 4

George Alter
Inter-university Consortium for Political and Social Research & University of Michigan

Kees Mandemakers
International Institute of Social History, Amsterdam & Erasmus University Rotterdam

## ABSTRACT

The Intermediate Data Structure (IDS) is a standard data format that has been adopted by several large longitudinal databases on historical populations. Since the publication of the first version in Historical Social Research in 2009, two improved and extended versions have been published in the Collaboratory Historical Life Courses. In this publication we present version 4 which is the latest 'official' standard of the IDS. Discussions with users over the last four years resulted in important changes, like the inclusion of a new table defining the hierarchical relationships among 'contexts,' decision schemes for recording relationships, additional fields in the metadata table, rules for handling stillbirths, a reciprocal model for relationships, guidance for linking IDS data with geospatial information, and the introduction of an extended IDS for computed variables.

**Keywords:** Life Courses, Demography, Historical Demography, Data Model, Entity Attribute Value Model, Intermediate Data Structure, IDS, Comparative Research, History, Social History

| | |
|---|---|
| *e*-ISSN: | 2352-6343 |
| PID article: | http://hdl.handle.net/10622/23526343-2014-0001?locatt=view:master |
| PID METADATA table: | http://hdl.handle.net/10622/23526343-2014-0002?locatt=view:master |

Both the article and METADATA table belonging to this article can be downloaded from here.

# 1 INTRODUCTION

The *Intermediate Data Structure (IDS)* is a common data format that is being adopted by longitudinal databases on historical populations around the world. Many of the important contributions from historical demographic research have been based on individual-level data describing life course transitions. Most of these studies have been based on small areas, only rarely covering an entire country (Kelly Hall et al. 2000). The value of these databases will be much greater if they can be easily compared at national and international scales, but inconsistencies in the representation of data in various databases have hampered comparison. The Intermediate Data Structure makes data comparable across databases by providing a common dissemination format. Our goal is to follow the example of the Integrated Public Use Microdata Series (IPUMS) (Ruggles et al. 2008) project, which has successfully encouraged new research with historical data. By providing data in a consistent and easy to use form, IPUMS has generated thousands of studies with data that were already available in less user-friendly versions.

The IDS is intended not only to standardize the dissemination of data, but also to encourage the development and exchange of software for data analysis. Longitudinal data is inherently complex, and the transformation of raw data into files suitable for analysis has been a difficult and costly process. Previous efforts to share data management and analysis software have been unsuccessful. The IDS provides a framework for building software that can be shared across many databases. Researchers will be able to identify common features of data from different sources without needing to learn a different data structure for each database. Database administrators will not need to build specific software for every research question, because they can draw upon modules developed by other members of the IDS community. IDS emphasizes the commonalities among databases without limiting the use of the unique features in specific sources.

In this article we present version 4 of the IDS. This version incorporates lessons learned from implementing the IDS in several databases (DDB Umea; Scania database, Historical Sample of the Netherlands and several databases kept at ICPSR) and discussions in an ongoing series of workshops.

The first 'official' version was published in *Historical Social Research* as the second part of 'Defining and Distributing Longitudinal Historical Data in a General Way through an Intermediate Structure' (Alter, Mandemakers & Gutmann 2009). We refer to this article for more about the background and motivation behind the development of the IDS. The second and third version was published at the Collaboratory *Historical Life Courses* (Alter and Mandemakers, 2011, 2012). These discussions resulted in important changes like the inclusion of a new table (CONTEXT_CONTEXT) making the handling of hierarchical contextual data more easy, the inclusion of decision schemes about the handling of relationships and new fields in the METADATA table. Rules were developed for handling stillbirths, a reciprocal model for relationships, guidance for linking IDS data with geospatial information, the introduction of an extended IDS for computed variables and the handling of start and end date of observations. For an overview of all changes since version 1, we refer to Appendix A and B.

The Intermediate Data Structure approach in the process of disseminating data has a number of important benefits:

- It is open, scalable, and extendable. Any database can transfer its data to the IDS, and the metadata registry will be extended to accommodate new types of data as they become available. New types of analysis can be introduced by adding new extraction modules.

- Since census data may be seen as a snapshot out of a life course, the IDS is always able to handle this kind of data without any restriction. The same is even more relevant for semi-longitudinal data which can be seen as a series of snapshots, for example a combination of data linked over several censuses and including civil certificates like the projected Victoria panel (Schürer 2007).

- Database managers will decide what data they provide and how their data can be used. Data producers can transfer data to the IDS in stages. Attributes that require minimal programming can be issued first, and new versions of the database can be created as more difficult data management tasks are solved. Databases that include confidential information can withhold identifiers that would disclose individual identities. For example, databases that have complex censoring structures can develop attribute types that limit the ways that their data are used. Since extraction programs will require specific attribute types, data providers can be sure that only appropriate data management procedures will be applied to their data.

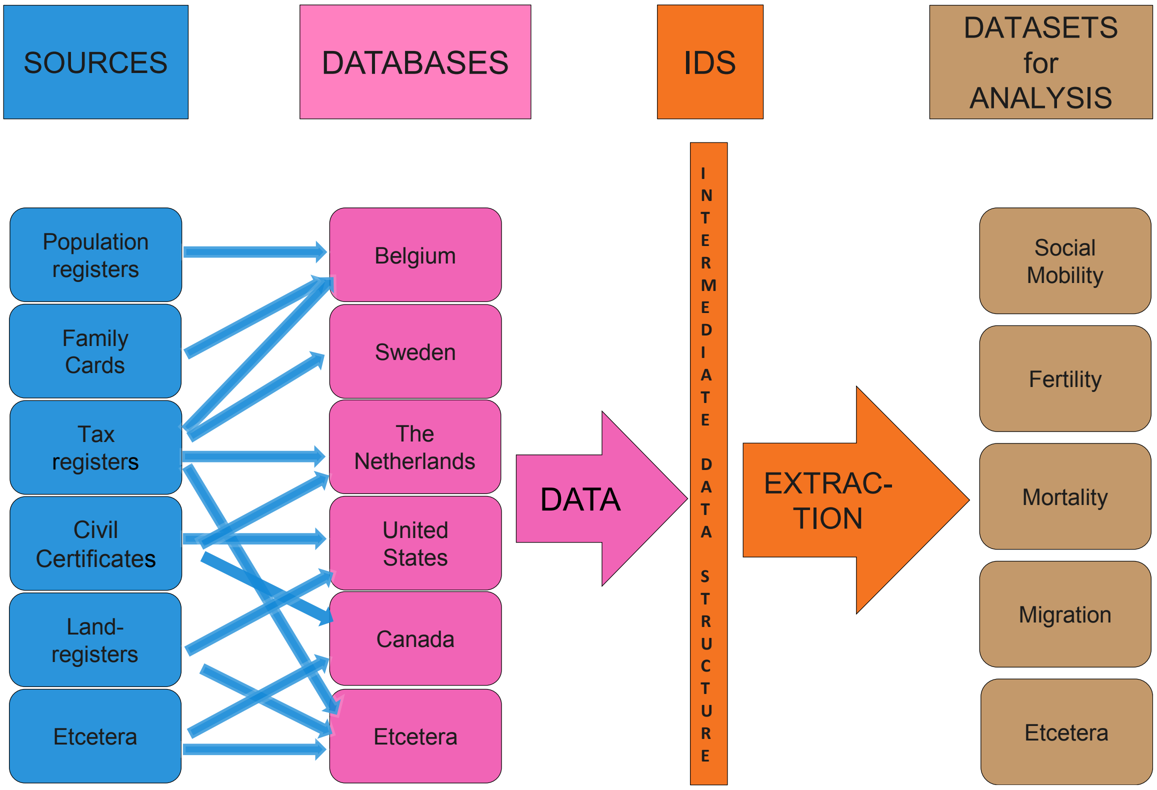
- Extraction programs will be re-usable and transparent. Anyone can contribute an extraction module, and all extraction modules will operate on every dataset with the required data. Extraction programs will also be open to scrutiny by the research community. Methodologies can be examined, discussed, and tested, and research results will be reproducible.

# 2 INTERMEDIATE DATA STRUCTURE (IDS)

## 2.1 OVERVIEW OF THE IDS

Figure 1 presents the basic idea of the *Intermediate Data Structure (IDS)* that all relevant longitudinal databases transfer their data into a simple common data format. The format of this data structure must be specified by the community of users. On the left side of the diagram are the various types of sources included in historical longitudinal databases. These sources vary widely from baptisms, marriages, and burials in parish registers to medical examinations and payment histories in pension records. Each database captures and stores data in a different way, and it is impossible to create a single data management structure that will work for every situation. On the right side of the diagram are the data files that researchers require for analysis. These files should be in a rectangular format that will be compatible with standard statistical packages (SPSS, SAS, Stata, etc.). While some statistical packages can manage hierarchical or relational file structures, these complexities impose costs on the user and limit accessibility. Between the sources and the analytical formats is an Intermediate Data Structure (IDS), which provides a standard format for all databases.

Figure 1 *Strategy with intermediate structure collecting data for scientific research from historical longitudinal databases.*



The IDS requires two kinds of computer programs:

*Data transfer.* Data must be reformatted for transfer from the database to the IDS. This includes original data as well as enhancements and standardizations, such as recoding occupations into the HISCO system. Transferring information from the source database into the IDS format also implies the generation of descriptive metadata to document the source and construction of all data. Since each source database is unique, this process will vary in many details. This approach gives each database administrator control over what and how data are disseminated.

*Extraction.* The extraction process moves data from the IDS into file formats designed for analysis. This process may include steps to construct new variables in IDS format and steps to convert data from the IDS into other formats more convenient for programming. Since the requirements of every type of analysis differ (fertility, mortality, social mobility, etc.), we expect to have many specialized extraction programs. However, all extraction programs will start with the IDS, and they will work on any dataset that includes the necessary attribute types. Extraction programs will be modular, and some types of analysis will require 'workflows' that link together several extraction services. This process creates standardized information for all databases.[1]

This approach separates the programs that transfer data from the original database into the IDS from the programs that create datasets in the rectangular format used by statistical packages. All databases will have the same structure, which will be independent of the form in which they were originally captured or stored. Researchers will not need to learn a new set of formats and relational structures for every database. Consequently, data extraction programs can be re-used and adapted to other purposes, and the steps involved in preparing data for analysis will be more open and transparent. Each database providing data will be responsible for transferring their data into the IDS, and databases will be able to choose how their data are represented in the IDS to control how it can be used.

## 2.2 PRINCIPLES

1. The database consists of two kinds of entities, persons and contexts, and the relations among persons, contexts and between persons and contexts.

2. Identifying unique persons from multiple appearances in the sources (record linkage) must be done by the data producer.

3. Contexts locate individuals in physical and social space. Contexts are multidimensional and may be nested.

4. The links between individuals and contexts tell us who lived together and who shared the same environments and experiences.

5. All entities in the IDS can be located in time. A *Time Stamp* is used to date all attributes of persons and contexts. Time stamps must be constructed by the database provider and should include information about how estimates have been made.

6. Individuals and contexts are described by attributes. Each database can choose which attributes to provide.

7. Attribute definitions are embedded in the IDS by the attribute *Type*. A Metadata Registry will be maintained so that common attribute types can be re-used by various archives, but each data provider can define (and register) new attribute types as necessary.

8. Each record entails only one attribute. This approach is known as the Entity Attribute Value model (EAV) or object-attribute-value model and was introduced in the 1970s (Stead et al. 1982).

[1] Luciana Quaranta (2013) developed a model for extraction software, combining both the addition of new variables in IDS format, so-called extended IDS, and the conversion of the IDS into other formats.

# 3 DATA MODEL

## 3.1 TABLES

The IDS consists of six files (or 'tables' in database terminology):

INDIVIDUAL consists of attributes belonging to a person (name, sex, wealth, literacy, etc.) and events (birth, marriage, migration, death, etc.). Every item of information about an individual is recorded as a separate row in this table. Each row has an attribute type, keys linking to an individual, and a timestamp. Rows in this table may be time-constant attributes (sex, date of birth), time-varying attributes (marital status, occupation), or events that mark changes in attributes (marriage, retirement). The attribute type will distinguish between a marriage certificate (which records the date that a subject's marital status changed from 'single' to 'married') from the marital status 'married' recorded in a census (which means that the subject became married some time before the date of the census).

INDIV_INDIV characterizes relationships between persons. This table will record relationships between two individuals. These relationships may be biological (parent-child), social (husband-wife, godparent-godchild), or economic (master-apprentice, owner-renter). Relationships will be timestamped, when appropriate (e.g. date of marriage).

CONTEXT describes places or environments that affect one or many persons, such as a household, house, geographic location, school, business firm, or organization. Contexts are sets of characteristics shared by groups. Household, for example, implies that a group of individuals shares a common living area, eats together, and pools resources. Contexts may also be places (buildings, geographic coordinates, villages, districts), organizations (business firms), or kinship groups (clans). Like the individual attribute table, contexts are described by attribute types and timestamps. Contexts may also be layered, and each context may include a link to a higher level of context in which it is nested.

INDIV_CONTEXT associates an individual with a context at a moment or during a period of time. Date stamped links between individuals and contexts are recorded in this table.

CONTEXT_CONTEXT defines the relations between different layers in a hierarchy of contexts. Layers are often specific to a country or region, and they may change over time.

METADATA Attribute types will be recorded in a central metadata registry. This will encourage standardization, but it also allows databases to add attribute types that are tailored to their needs. For example, 'marriage' will be used by many databases, but some databases will have 'publication of marriage banns' or 'marriage contract signed.'

## 3.2 INDIVIDUAL DATA

### 3.2.1 TABLE INDIVIDUAL

The table INDIVIDUAL contains all attributes that characterize an individual. This table has the following (basic) structure (see also table 1 with some examples of records):

Id Primary key.

Id_D Identifier of the database or parts of the database from which the data are extracted. This code is especially needed to differentiate between databases when tables from different databases are merged. More generally it is the way the version of the release has to be documented.

Id_I Identifying number of each individual in the database. *This presupposes that the work of linking individuals has been done by the original database.*

Source Specification of the source. We include a field for the source, because an attribute may be reported more than once in different documents within a single database.

Type Type of attribute (including events that are a subcategory of attributes).

Attribute types are explained in the METADATA table. The following examples illustrate attribute types, starting with common ones and ending with more specific attributes belonging to only one database:

Last name
Date of Birth
Location of Birth
Date of Baptism
Date of Death
Date of Marriage
Location of Marriage
*If the sequence of marriages can be distinguished:*
Date of First Marriage
Date of Second Marriage, etc.

Start Observation
End Observation
Location of Origin
Location of Departure
Reason for Sampling

Dutch Personal Income Tax (period 1860-1880)
Number of Food Distribution Card during First World War

Value The value of the attribute. Many attributes have values, such as 'male' and 'female' for the attribute 'sex'. For events (e.g. birth, death), this value usually will be left empty, because the time stamp shows when the event occurred.

Many of these values are of a contextual nature like location of birth. To facilitate a direct connection with the CONTEXT-table the following field *Value_Id_C* may be used. In that case the attribute *Value* remains empty.

Value_Id_C Identifier to the CONTEXT-table for values of a contextual nature.

Timestamp A time stamp for the moment or period in time that the attribute is valid (see section 4).

A special note concerns the handling of stillbirths. For the stillbirths itself we have defined specific types (date and location of stillbirth). Stillbirths are defined as ‘persons’ that have never been alive. However, in the sources quite often infants that died within a couple of hours are included as stillbirths. If this situation is recognized these ‘stillbirths’ must be handled as normal births who were born and died on the same day. This is in accordance with the definition of ‘live births’ from the World Health Organisation (WHO) that defines a live birth as a ‘complete expulsion or extraction from the mother of a baby, irrespective of the duration of the pregnancy, which, after such separation, breathes or shows any other evidence of life [..]’. [2]

For the dating and the location of main events like BIRTH, DEATH, MARRIAGE and STILL_BIRTH we use two records to distinguish between location and date. At first sight this seems a little bit redundant. However, in principle they are two different attributes of which location could be time stamped by the date of birth.This is in accordance with principle 8 (see chapter 2.2). It is also practical, since in many cases we know the location but not the date (for example if we have age instead of the date of birth). Since the time stamp has an excellent structure to handle dates we use the timestamp to fill in values with a dating character and not the value field itself.

[2] http://www.who.int/healthinfo/statistics/indmaternalmortality/en/index.html

Table 1 *Records in the table INDIVIDUAL (excluding timestamp variables)*

| Id | Id_D | Id_I | Source | Type | Value | Value_Id_C |
|---|---|---|---|---|---|---|
| 1 | DDB_release_2012.01 | 1 | Population register | Last_Name | Johansson | |
| 2 | DDB_release_2012.01 | 1 | Population register | First_Name | Christiaan | |
| 3 | DDB_release_2012.01 | 1 | Population register | Birth_Date | <time stamp> | |
| 4 | DDB_release_2012.01 | 1 | Population register | Birth_Location | | 1029 |
| 5 | DDB_release_2012.01 | 1 | Population register | Death_Date | <time stamp> | |
| 6 | DDB_release_2012.01 | 1 | Marriage certificate | Marriage_Date | <time stamp> | |
| 7 | DDB_release_2012.01 | 1 | Population register | Observation | <time stamp> | |
| 8 | DDB_release_2012.01 | 1 | Income tax register | Occupation | Timmerman | |
| 9 | DDB_release_2012.01 | 1 | Income tax register | Occupation_Eng | Carpenter | |
| 10 | DDB_release_2012.01 | 1 | Income tax register | Occupation_ HISCO | 95410 | |
| 11 | DDB_release_2012.01 | 1 | Population register | Civil Status | Married | |
| 12 | DDB_release_2012.01 | 1 | Population register | Sex | Male | |
| 13 | DDB_release_2012.01 | 1 | Vaccination register | Vaccination | Vaccinated | |

## 3.2.2 START AND END OF OBSERVATIONS

The start and end of an observation is an important topic in the case of population registers. While in the case of events the date of the event is the start and end of an observation in one; in the case of population registers we have to do with one or more periods of observation. We define an observation as the period a person is registered in subsequent sources without leaving a time gap.

There are three ways to arrive in a context: by birth, by arrival from another context and by way of the fact that a person was already included at the start of the register. And the three counterparts define the way to depart from a context: by death, departure to another context and being present at the closing of the register.

To capture start and end of observation of an individual we use the types *Start_Observation* and *End_Observation*. Both types have only three values; in the case of the start of the observation: 'Source_Start', 'Birth', 'Arrival' and in the case of the end of the observation: 'Source_End', 'Death' and 'Departure'.

To specify the context where persons come from or where persons go to, we have two types in the INDIVIDUAL table: *Arrival_From* with the location of origin and *Departure_To* with the location of destination. The timestamp of these two types will be equal with the timestamp in the corresponding values in *Start_Observation* and *End_Observation*.

In quite a lot of cases persons are only changing sources without changing contexts, for example when a population is reentered in a new opened register, replacing the old one. Or persons are moving from one context to another without leaving observation. This happens for example when persons are moving from one address to another or when persons are migrating to another parish or municipality, while both sources are covered by the database in question. These changes are recorded in the INDIV_CONTEXT table in which all context changes are covered (see section 3.3.2 and 3.3.6). All source changes are covered in this system as well.

NB There is also the type *Observation*; this type covers the period or periods when a person is observed. In principle this type is redundant, but it could be used to get a quick answer on questions of total observation time, or total time of gaps in observation.

### 3.2.3 TABLE INDIV_INDIV

The table INDIV_INDIV shows how individuals are related to each other. See figure 2 for a presentation of how the INDIVIDUAL and INDIV_INDIV tables are used, see table 2 for an example of records.

This table has the following structure:

Id Primary key.

Id_D Identifier of the database or parts of the database from which the data are extracted. This field is to be used for versioning as well.

Id_I_1 Identifying number of the first individual in the relationship, referring to *Id_I* in the INDIVIDUAL table.

Id_I_2 Identifying number of the second individual in the relationship, referring to *Id_I* in the INDIVIDUAL table.

Source Specification of the source.

Relation Type of relationship of the first person to the second person. For example, person 1 is the 'father' of person 2. This implies that all relationships are reciprocal.

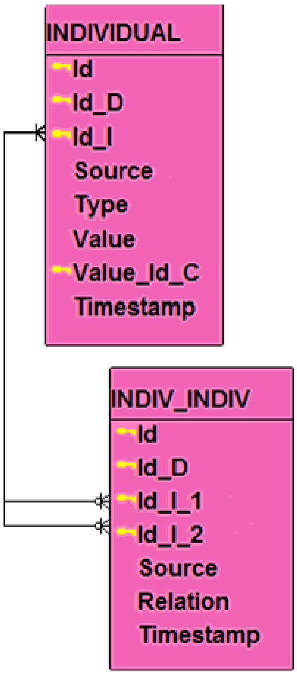
Values like:
Father [which means that the person from *Id_I_1* is the father of the one from *Id_I_2*]
Child
Husband
Wife
Householder
Maid
Stillbirth
etc

A list of all valid values for Relation will be maintained in the metadata-table (see section 3.4). In ambiguous situations, multiple relations may be included in the METADATA table with the operator 'or', such as 'sibling or half-sibling' (when one parent is not known) and 'cousin or nephew' (when the sources do not distinguish between these relationships).

Discussion may be necessary when variants describe the same relationship. For example, 'Groom' and 'Husband' describe the same relationship in different situations. 'Groom' is only used during an event (marriage), while 'Husband' is a status that does not identify the timing of the event. In general, one variant should be used for each relationship to avoid duplication and reduce the need for programming.

Timestamp A time stamp for the moment or period in time that the relationship is valid (see section 4). Biological relationships are independent of time like 'father', 'child' or 'brother". The timestamp may be left empty in those cases. The data producer will be responsible for resolving inconsistencies in relationships before the data is transferred into the IDS, but standard programs for detecting inconsistencies may be developed.

Figure 2 *ERD-diagram tables of individual data*



*Explanation: The relations are described by way of so-called Entity Relationship Diagramming. Here: every individual may have one or more relationships with other individuals, but every relationship must refer to two individuals in the INDIV_INDIV table (see Beaumont 2007, for more information about Entity Relationship Diagramming).*

Table 2 *Records in the table INDIV_INDIV (excluding timestamp variable)*

| Id | Id_D | Id_I_1 | Id_I_2 | Source | Relation |
|---|---|---|---|---|---|
| 1 | HSN_release_2010.02 | 1 | 2 | Birth certificate | Wife |
| 2 | HSN_release_2010.02 | 2 | 1 | Population register | Husband |
| 3 | HSN_release_2010.02 | 1 | 22 | Birth certificate | Mother |
| 4 | HSN_release_2010.02 | 22 | 1 | Birth certificate | Child |
| 5 | HSN_release_2010.02 | 2 | 22 | Population register | Father |
| 6 | HSN_release_2010.02 | 22 | 2 | Marriage certificate | Child |
| 7 | HSN_release_2010.02 | 2 | 23 | Population register | Householder |
| 8 | HSN_release_2010.02 | 23 | 2 | Population register | Maid |
| 9 | HSN_release_2010.02 | 2 | 8493 | Population register | Master |
| 10 | HSN_release_2010.02 | 8493 | 2 | Population register | Servant |
| 11 | HSN_release_2010.02 | 823 | 824 | Population register | Sibling |
| 12 | HSN_release_2010.02 | 824 | 823 | Population register | Sibling |

## 3.3 CONTEXT DATA

### 3.3.1 INTRODUCTION

Contextual information can be described as information about shared environments, such as households and regions. An individual lives in several contexts at the same time. Contexts are of a hierarchical nature, but different types of contextual divisions may exist at the same level and at the same time. A municipality, for example, may be part of a judicial district with different geographic borders than the school district to which the municipality also belongs.

Contextual levels are important because they define the living environment of individuals, but also because at the level of context we can connect more information to the individuals included in our research. Contextual information may include the amount of tax paid by the household, the quality of the soil in the locality, the number of inhabitants of a municipality, the level of school enrolment in the school district, etc.

We may also use the concept of context to capture administrative or technical aspects of source documents. In Swedish population registers for example, it is not unusual for servants to be registered at the end of a page without making clear to which household they belong among several households listed on the same page. We may use the page as the contextual framework to which all individuals of that page belong to connect households and servants. More generally, a page in a population register often corresponds to an address, and a complete register may correspond to a specific locality within a municipality or quarter within a city.

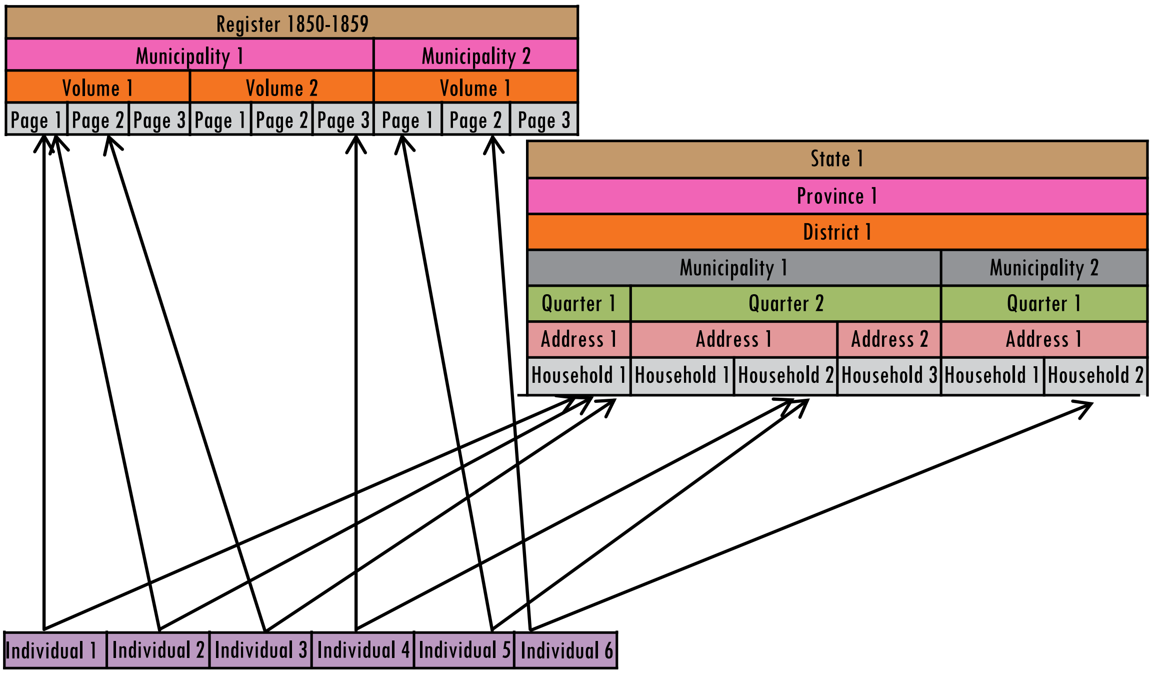
Each database will define its own contextual system, which may differ according to period and region. Contextual information may be linked to geographical identifiers like centroids (coordinates defining a central point on the map, e.g. the geographical middle of a locality) or polygons (combining coordinates of line segments into an area). These geographical identifiers enable the creation of maps as a way of analysing our data in a more descriptive and intuitive way (Gregory and Ell, 2007). In some cases, the CONTEXT table may refer to geographical information stored outside of the IDS structure (see section 3.3.3). Contextual information may also link to so-called gazetteers (reference tables with names of locations and coordinates which may include other information concerning locations like standardized spelling, higher contextual levels and various geographic characteristics) or comparable databases (Southall et al. 2011).

Figure 3 shows individuals linked to two context hierarchies. One describes the structure of information in the administrative source, a population register. Individuals are listed on pages, pages are grouped in volumes, volumes are grouped by municipality, etc. The second context hierarchy describes residential locations. Individuals live in households, households are located at addresses, addresses are within quarters, etc. In general, contexts within the same layer of a hierarchy are mutually exclusive, and several lower levels may be linked to the same higher level context, such as municipality, province and state.

The system of contexts may overlap and change over time, and several systems may exist for the same period of time. For example in Dutch administration the provincial level is very important, however some divisions in socio-economic regions zigzag across provincial borders. Every new arrangement of municipalities into school districts means that we have a new context for school districts. The highest distinguished level, the state, is important because many kinds of economic data (e.g. industrial activity, inflation rates, unemployment) are often available only at this level.

The foregoing implies that several hierarchies may be constructed. In some cases, distinctions between hierarchies will be indicated by time stamps. Some hierarchies may converge, e.g. in figure 3 the division into municipalities may be the same for both hierarchies. The first one is source oriented; capturing typical information like volume and page of the original document. The second one is context oriented creating a hierarchy of contexts which may be described from sources or databases complementing the micro-data about people.

Figure 3 *Example of hierarchical layering of contextual information*



## 3.3.2 MODEL

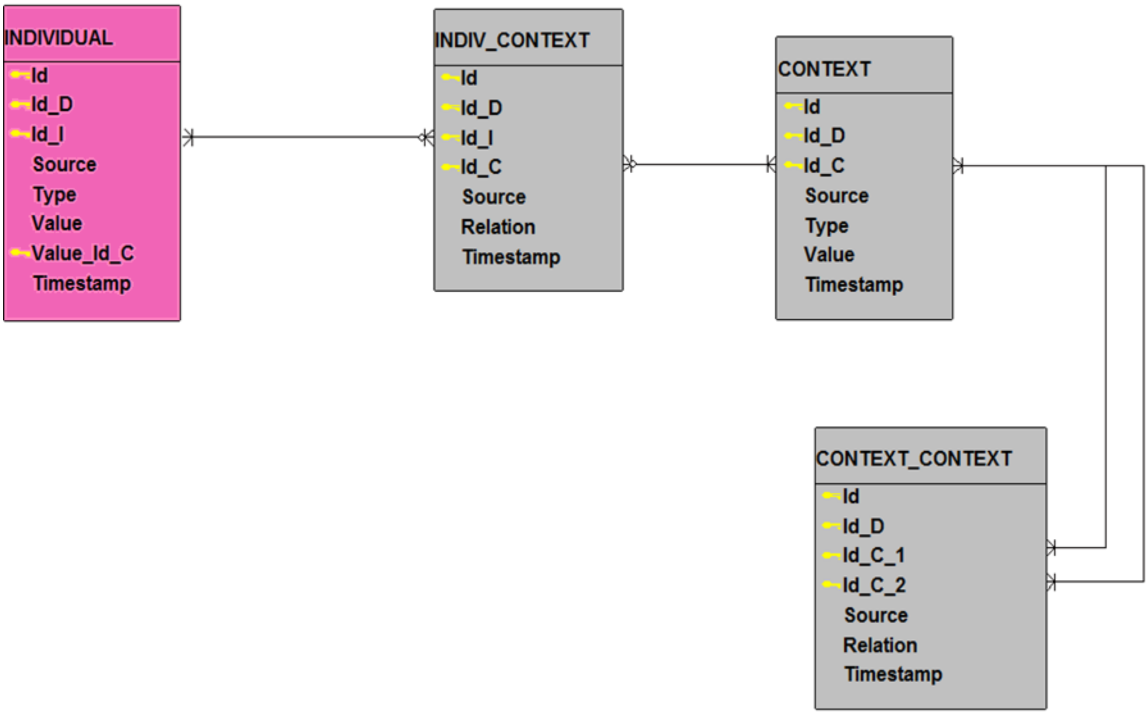
Contextual information is described by three tables. The CONTEXT table provides all attributes for each distinguished context. The INDIV_CONTEXT table connects individuals with the contextual information. The CONTEXT_CONTEXT table shows relations between contexts, and it allows characteristics of multiple layers of contextual information to be associated with individuals without repeating the information in the database. See appendix B for the differences with the way the context was handled in the first version of the IDS.

Because of the hierarchical nature of contextual information, it is not necessary to link contexts over more than one level. Elaborating on the example of figure 3, it is necessary to link each quarter with the higher level of municipality, but it is not necessary for quarters (in the residential hierarchy) to correspond to volumes (in the source document), even though both can be grouped into municipalities.

When there is more than one hierarchy as in in figure 3, it is necessary to have more records in the INDIV_CONTEXT table. Each hierarchy needs its own connections, because they form separate structures.

See figure 4 for a presentation of how the CONTEXT tables are connected with each other and the INDIVIDUAL table. Each database may decide to include extra tables in this system to define relationships with geographical identifiers, because data describing polygons or raster images are often too voluminous for the standard IDS tables.

Figure 4 *ERD-diagram of the contextual data*



*Explanation: see figure 2*

### 3.3.3 TABLE CONTEXT

The table CONTEXT lists all relevant contextual information, such as information about households and regions. Each context is assigned a unique ID_C identifier. As in the INDIVIDUAL table, each row in the CONTEXT table describes an attribute of a context. Constructed attributes (like household size or household type) may be provided by the database as a service to users, but the IDS also allows these attributes to be constructed dynamically by data extraction programs. These data may also be included in the IDS system as so called *extended IDS* (see section 3.4).

An individual can live at the same time in multiple contexts because they are layered. All connections between contexts are made explicit in the table CONTEXT_CONTEXT. For this reason it is not necessary to repeat information about attributes of higher layers in the CONTEXT table. The level or type of the context will be included as an attribute.

For a discussion of the concept CONTEXT and what a 'contextual entity' defines, see section 3.3.1. The IDS is not appropriate for storing types of geospatial data that cannot be summarized in the 250 characters allowed for the value of an attribute (e.g. polygon descriptions). In these cases, we recommend that these data be stored in an external database with a key (e.g. an ID number) linking entities in the CONTEXT table to external representations of them. That there is an external database is indicated with the value 'Id_Polygon' in the *Type* field, and the *Value* field will contain the external key. Additional information about the location of the external database should be provided when the *Type* is defined in a database specific way in the METADATA table. See Table 3 for examples of records.

The CONTEXT table consists of the following (basic) data structure:

| | |
|---|---|
| Id | Primary key |
| Id_D | Identifier of the database or parts of the database from which the data are extracted. This field is to be used for versioning as well. |
| Id_C | Identifying number of the context |

Source  Specification of the source

Type  Type of attribute of the context
Name
Level
Housenumber
Streetname
Postal code
Id_polygon [A key linking this context to data in a location outside the IDS. The external location is defined in the METADATA table.]
Etc.

Value  The value of the attribute

Timestamp  A time stamp for the moment or period in time that the attribute is valid, see section 4. If no timestamp is given in the table CONTEXT, the timestamp in the table INDIV_CONTEXT is supposed to cover fully the specific context.

Table 3  *Records in the table CONTEXT (excluding source and timestamp variables)*

| Id | Id_D | Id_C | Type | Value |
|---|---|---|---|---|
| 1 | Utah_release_2011.01 | 115023 | Street_id | 3929 |
| 2 | Utah_release_2011.01 | 115023 | Streetname | Mainstreet |
| 3 | Utah_release_2011.01 | 115023 | Streetnumber | 12 |
| 4 | Utah_release_2011.01 | 115023 | Long_Centroid | 233.838 |
| 5 | Utah_release_2011.01 | 115023 | Latit_Centroid | 193.933 |
| 6 | Utah_release_2011.01 | 115023 | Level | Address |
| 7 | Utah_release_2011.01 | 9022 | Name | Salt Lake Harbour |
| 8 | Utah_release_2011.01 | 9022 | Number_inhab | 230 |
| 9 | Utah_release_2011.01 | 9022 | Long_Centroid | 233.838 |
| 10 | Utah_release_2011.01 | 9022 | Latit_Centroid | 193.933 |
| 11 | Utah_release_2011.01 | 9022 | Level | Neighbourhood |
| 12 | Utah_release_2011.01 | 10345 | Name | Salt Lake City |
| 13 | Utah_release_2011.01 | 10345 | Number_inhab | 23455 |
| 14 | Utah_release_2011.01 | 10345 | Long_Centroid | 233.921 |
| 15 | Utah_release_2011.01 | 10345 | Latit_Centroid | 193.888 |
| 16 | Utah_release_2011.01 | 10345 | Level | Municipality |
| 17 | Utah_release_2011.01 | 115029 | Street_id | 2932 |
| 18 | Utah_release_2011.01 | 115029 | Streetname | Smallstreet |
| 19 | Utah_release_2011.01 | 115029 | Streetnumber | 212 |
| 20 | Utah_release_2011.01 | 115029 | Longitude | 233.847 |
| 21 | Utah_release_2011.01 | 115029 | Latitude | 193.899 |
| 22 | Utah_release_2011.01 | 115029 | Level | Address |

### 3.3.4 TABLE CONTEXT_CONTEXT

The CONTEXT_CONTEXT table defines connections between different layers in a hierarchy. Contexts are hierarchical, but several contexts may be defined for the same layer. The system of contexts may change over time and several systems may exist for the same period of time. A database may include multiple context hierarchies, such as address-neighborhood-municipality and page-volume-district.

When the context hierarchy has been specified, attributes of more inclusive layers can be linked directly or indirectly to lower layers. This implies that only one record is necessary in the INDIV_CONTEXT table to grasp all contextual situations for an individual in a particular context hierarchy. For example, linking an individual to an address may imply links to a neighborhood, municipality, and province (all within the time frame for each context to context relationship defined by way of the time stamps).

The CONTEXT_CONTEXT table consists of the following (basic) data structure:

Id Primary key

Id_D Identifier of the database or parts of the database from which the data are extracted. This field is to be used for versioning as well.

Id_C_1 Identifying number of the less inclusive context in the relationship, referring to a value on the field *ID_C* in the CONTEXT table. This layer is always included within the context related to by *Id_C_2.*

Id_C_2 Identifying number of the more inclusive context in the relationship, referring to a value on the field *ID_C* in the CONTEXT table.

Source Specification of the source

Relation Description of the relationship between the context layers, like:

Address and neighborhood
Neighborhood and municipality
Municipality and school district system A
Municipality and census districts 1850-1880

Timestamp A time stamp for the moment or period in time that the relationship is valid (see section 4).

Table 4 *Records in the table CONTEXT_CONTEXT (without the Source field and part of the timestamp)*

| Id | Id_D | Id_C_1 | Id_C_2 | Relation | Time Stamp (period) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Utah_ release_2011.01 | 115023 | 10345 | Address and Municipality | 21 | 2 | 1879 | 2 | 6 | 1882 |
| 2 | Utah_ release_2011.01 | 115029 | 9022 | Address and Neighborhood | 21 | 2 | 1879 | 2 | 6 | 1882 |
| 3 | Utah_ release_2011.01 | 9022 | 10345 | Neighborhood and Municipality | 21 | 2 | 1879 | 2 | 6 | 1882 |

## 3.3.5 HOUSEHOLDS AND INSTITUTIONS

The concept of ‘household’ is often problematic. ‘Household’ usually refers to a group who pool income and share consumption (Hammel and Laslett, 1974; Brettell 2003). In some cultures, households have continuity over time that is independent of the people who inhabit them. In other cultures, households are simply the group that lives together at a moment in time. In these cases, it is often useful to define households by associating each household with a single reference person, who may or may not be the ‘head,’ such that everyone who lives with the reference person is in the same household. When a source, such as a census, specifies relationships among people in a household, those relationships may be captured in the INDIV_INDIV table. When these relationships can be redefined in terms of biological relationships, like ‘Son of householder’ they should always be included in the INDIV_INDIV table.

Institutional environments like boarding schools, old people’s homes, hospitals or military barracks are to be considered as contextual environments comparable with households. In some sources they are often headed by a more or less artificial householder like headmaster, ‘father’ of an orphanage, director or captain.

### 3.3.6 TABLE INDIV_CONTEXT

The table INDIV_CONTEXT places individuals into contexts. When the CONTEXT_CONTEXT table is fully specified, it is only necessary to link a record to the lowest level of each context hierarchy.

INDIV_CONTEXT includes a field named *Relation* to specify the relationship of a person to the specific context. Relationships between individuals should be defined in the INDIV_INDIV table, but some relationships may be recorded in both places. For example, ‘servant’ usually implies both a relationship to the householder as an individual (employer/servant) and to the entire household for which services are provided. 'Apprentice shoemaker' may describe only an individual relationship (master/apprentice), but it may also represent a relationship to the household as a unit of production (e.g. a shop).

Scheme 1 presents a decision scheme for determining where relationships should be defined. Figure 5 shows the graphical presentation of this scheme. Table 5 gives a few examples of records in the INDIV_CONTEXT table.

Id Primary key.

Id_D Identifier of the database or parts of the database from which the data are extracted. This field is to be used for versioning as well.

Id_I Identifying number of an individual.

Id_C Identifying number of a context.

Source Specification of the source.

Relation The type of the relationship between individual and context (a value will not always be needed). In cases with multiple values include extra records with the same timestamp.

Householder
Co-resident
Lodger
Boarder
Servant
Maid
Student
Monk
Nun
Abbot
Prioress
etc.

Timestamp A time stamp for the moment or period in time that the attribute is valid, see section 4.

Scheme 1 *Guidelines for defining relationships of persons in the INDIV_CONTEXT table and including records in other tables*

| | | |
|---|---|---|
| 0 | Every individual has at least one record in the INDIV_CONTEXT table for each context in which the individual has been recorded. Do we need records in other tables as well? | |
| 1 | Is the relationship independent of the context, such as a biological or marital relationship (parent/child, husband/wife)? | |
| | Yes | There must be a record in the INDIV_INDIV table. A definition of the relationship on the record in the INDIV_CONTEXT is usually not necessary.<br>If additional information about a specific context is present, (e.g. a niece is reported with a relationship as 'lodger') go to step 3. |
| | No | Go to step 2 |
| 2 | Has the person a specific relationship with one or more individuals in the specific context? | |
| | Yes | There must be records in the INDIV_INDIV table for these relationships. For example, when the source explicitly lists a relation to the head of household for each person, each of those relationships should be recorded in the INDIV_INDIV table.<br>However, when only the head is designated in the source, he/she can be identified in the INDIV_CONTEXT table. In other words, it is not necessary to make records with unspecified relationships to the head in the INDIV_INDIV table.<br>If additional information about the context is present, go to step 3. |
| | No | Go to step 3 |
| 3 | Does the relationship include an occupational title (like servant, maid)? This includes titles describing a status, like 'gentleman', 'student' or 'orphan'. | |
| | Yes | Occupations are recorded in the INDIVIDUAL table.<br>If additional information about the context is present, go to step 4. |
| | No | Go to step 4 |
| 4 | Does the relationship have a meaning that is tied to the context in some way? Examples are: servant, lodger, boarder, boarding house keeper. | |
| | YES | Include the value in the field *Relation* |
| | NO | Keep the *field Relation* empty |

Figure 5 *Defining relations between individuals in INDIV_INDIV or INDIV_CONTEXT tables*

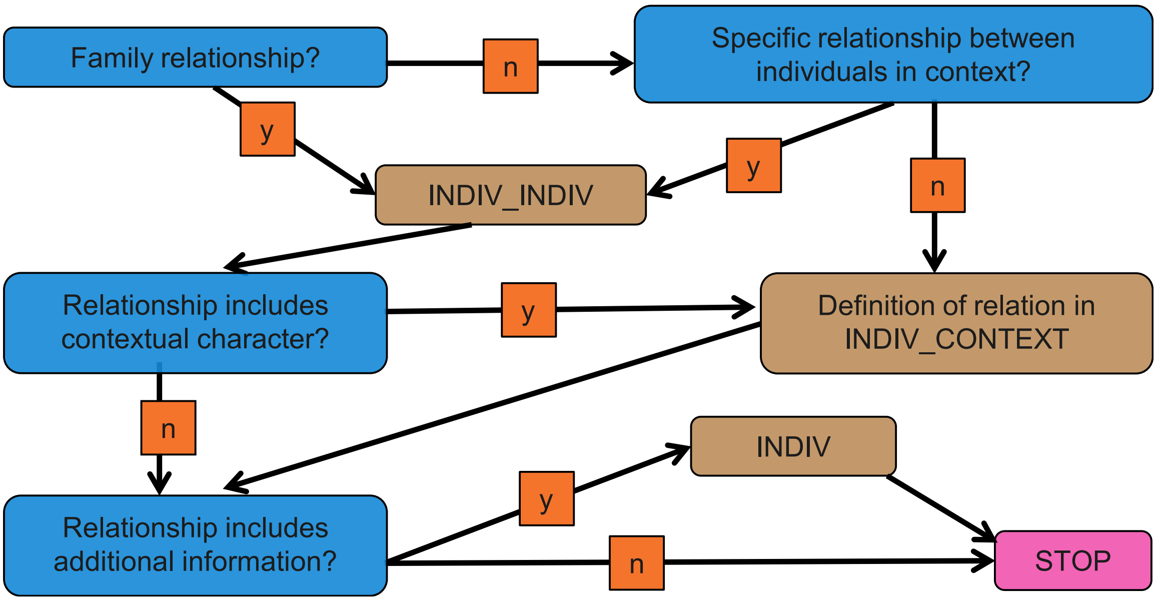

Family relationship?
n
Specific relationship between individuals in context?
y
INDIV_INDIV
y
n
Relationship includes contextual character?
y
Definition of relation in INDIV_CONTEXT
n
INDIV
y
Relationship includes additional information?
n
STOP

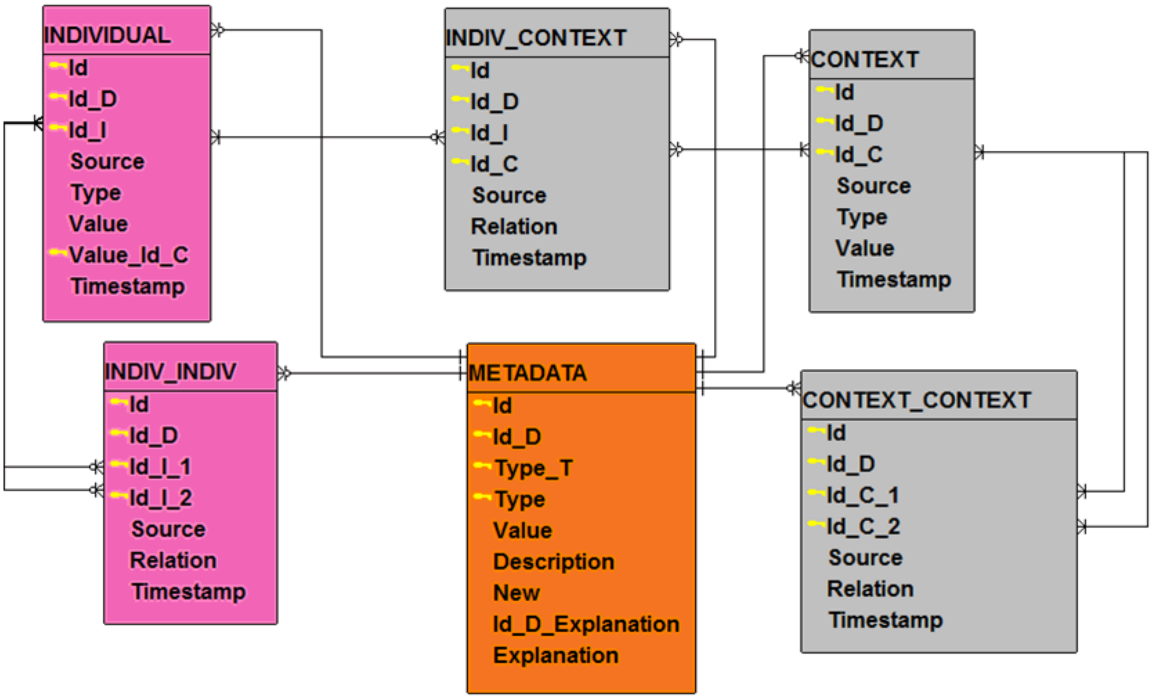
Table 5 *Example of records in the table INDIV_CONTEXT (without the field Source and part of the timestamp)*

| Id | Id_D | Id_I | Id_C | Relation | Time Stamp (period) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Utah_release_2011.01 | 1001 | 115023 | Householder | 21 | 2 | 1879 | 2 | 6 | 1880 |
| 2 | Utah_release_2011.01 | 1001 | 115029 | Householder | 3 | 6 | 1880 | 30 | 11 | 1882 |
| 3 | Utah_release_2011.01 | 2009 | 115029 | Servant | 15 | 8 | 1879 | 5 | 8 | 1882 |

## 3.4 METADATA TABLE

The METADATA table provides a complete explanation of all data in each database system. It is important to notice that the variables *Type* and *Value* already include a brief description of the meaning of the attribute. See figure 6 for the structure of the IDS, including the METADATA table.

Figure 6 *ERD-diagram of the Intermediate Data Structure including the METADATA table*



*Explanation: see figure 2*

The METADATA table consists of eight fields; the fields *Id_D*, *Type_T* and *Type* form the key to the other tables.

Id Primary key

Id_D Identifier of the database or parts of it from which the data are extracted. The name 'STANDARD' is reserved for metadata accepted by the community of researchers for general use, see below. [In the METADATA table versioning is handled by way of the field *New*.]

Type_T Identifier of the table or timestamp concerning the specific metadata. All five data tables include a column identifying a type of attribute or relation, and there are three are three kinds of information about dates on each timestamp, see section 4. Above this there are identifiers for the METADATA table as such and the types belonging to extended IDS.

| | |
|---|---|
| INDIVIDUAL | [Type] |
| INDIV_INDIV | [Relation] |
| CONTEXT | [Type] |
| INDIV_CONTEXT | [Relation] |
| CONTEXT_CONTEXT | [Relation] |
| TIMESTAMP | [Date_type] |
| TIMESTAMP | [Estimation] |
| TIMESTAMP | [Missing] |
| EXTENDED_IDS | [Type (only) made by extraction software] |
| METADATA | This field is used to provide information about the dataset as a whole, specifically the IDS version and the version of the specific dataset. |

Type Type of attribute, relation or timestamp.

In case of *Type_T* equals 'METADATA' only two types are possible: 'Version_IDS' and 'Version_Release'; the content is specified in the next field *Value.*

Value Type of values belonging to the specific attribute, relation or timestamp; in case only the variable itself is defined, the value will always be 'DEFINITION'.

New Field to keep track of all changes in the METADATA table. The value will be given by the release version of the IDS (like 3.0).

Extract Name of the extraction software in case the variable has been constructed by extraction software.

Id_D_Explanation Identifier of the database for which the explanation was made about the way the variable or value has been constructed.

Explanation Memo-field with an explanation of the meaning and use of this type of data (including for example a further explanation of the relevant sources).

The value 'STANDARD' in the fields *Id_D* and *Id_D_Explanation* is reserved to distinguish standard definitions of variables from database specific ones. The standard meaning of an attribute will be specified by the community of researchers, and database-administrators must follow those guidelines, if they use a standard TYPE and/or VALUE. Databases will add rows with their own *ID_D_Explanation* for each standard TYPE, which they may use to describe how an attribute is derived from the sources available to them. Thus, a TYPE or specific VALUE will have only one row where *Id_D* is 'STANDARD', showing the community's specification of this attribute, but it may have many rows explaining how various databases implemented that type.

Since we use one table to describe *both* variables and values, it is necessary to make this distinction explicit. This is realized by putting the term 'DEFINITION' in the value-field for a variable explanation.

As already mentioned in the description of the CONTEXT-file, it is possible to construct attributes and add them to the CONTEXT or the INDIVIDUAL table. Variables like household-size, number of children, time since last birth may be computed from attributes found in the original data.

Constructed variables may be created by the database itself or by 'extraction' software designed for this purpose. When extraction programs are offered to others as Open Access software and deposited online for the IDS community, the variables constructed by this software may be included in the METADATA table as a special content of the field *Type_T*: 'EXTENDED_IDS'. To have a link with the software an extra field has been added in the Metadata table, called *Extract* to archive the name of the program that creates the specific attribute or variable. This implies that the same attribute (e.g. household size) may be derived directly from a source or computed by software. Here the field *Type_T* will have the name of the specific table, while the existence of an extended version is made known by having filled in the field *Extract*.

Table 6 gives an example of nine records in the metadata-registry. Scheme 2 presents a decision scheme for distinguishing between the three options for defining variables in the metadata table.

Table 6 *Records in the table METADATA*

| Id | Id_D | Type_T | Type | Value | New | Extract | Id_D_ Explanation | Explanation |
|---|---|---|---|---|---|---|---|---|
| 1 | STANDARD | INDIVIDUAL | DEATH | DEFINITION | 1.0 | | STANDARD | Date of occurence of death. |
| 2 | STANDARD | INDIVIDUAL | DEATH | DEFINITION | 1.0 | | HSN | Date of occurence of death; three sources which we used in the following preference: 1 civil certificate, 2 population register, 3 Red Cross. We use 'Red Cross' as source when dates are estimated on the basis of circumstantial information but must be considered quite accurate, e.g. the date of death in German termination camps like Sobibor which was estimated on the base of date of deportation from The Netherlands. |
| 3 | DDB | INDIVIDUAL | CHILDBIRTH ASSISTANT | DEFINITION | 3.0 | | DDB | Indicates whether the child is delivered by a trained midwife. |
| 4 | DDB | INDIVIDUAL | CHILDBIRTH ASSISTANT | Delivery with an unexamined assistant | 3.0 | | DDB | The child was delivered with help from an untrained assistant. |
| 5 | DDB | INDIVIDUAL | CHILDBIRTH ASSISTANT | Midwife delivery | 3.0 | | DDB | The child was delivered with help from a trained midwife. |
| 6 | DDB | INDIVIDUAL | CHILDBIRTH ASSISTANT | Midwife delivery with instruments | 3.0 | | DDB | The child was delivered with help from a trained midwife and instruments were used. |
| 7 | DDB | INDIVIDUAL | CHILDBIRTH ASSISTANT | Unknown | 3.0 | | DDB | The way the child was delivered is unknown. |
| 8 | STANDARD | CONTEXT | HOUSEHOLD SIZE | DEFINITION | 4.0 | H_Size__ SEDD_ 2013_01 | STANDARD | Total number of household membership. |
| 9 | HSN | INDIVIDUAL | MUNICIPAL_ INCOME_TAX | DEFINITION | X | | HSN | Value municipal income tax, year of the tax defined by the timestamp and name of the municipality by way of the context. |

Scheme 2 *Practical guidelines for defining variables in the Metadata table*

| | | | |
|---|---|---|---|
| 1 | Is your variable/value completely described in the field *Explanation* in the STANDARD scheme? | | |
| | Yes | You are using the standard explanation and you have nothing to add to the content of the *Explanation* field (example record 1 in table 6). | |
| | | *Id_D* | STANDARD |
| | | *Id_D_Explanation* | STANDARD |
| | No | Go to step 2 | |
| 2 | Is the STANDARD explanation applicable but incomplete? For example, do you need more explanation about the construction of the variable? | | |
| | Yes | Make a new record with your own explanation in the *Explanation* field while copying the content of the fields *Type_t, Type* and *Value* (example record 2 in table 6) | |
| | | *Id_D* | STANDARD |
| | | *Id_D_Explanation* | Acronym of your database |
| | No | Go to step 3 | |
| 3 | Your variable does not fit in the existing STANDARD scheme, and you think it is a good candidate for a new STANDARD variable. | | |
| | Yes | Make a record with the explanation of your proposal for the new STANDARD variable or values and send the proposal to the Clearing Committee; while waiting for approval, go further with step 4 as a temporary solution. | |
| | No | Go to step 4 | |
| 4 | Your variable does not fit in the existing STANDARD scheme, and you must make metadata that will function within the IDS of your own database (example record 9 in table 6). | | |
| | YES | *Id_D* | Acronym of your database |
| | | *Id_D_Explanation* | Acronym of your database |
| | | *New* | Fill with an 'X', in case of clearance the version number of the IDS will replace the 'X.' |

## 4 TIME STAMP

Time is defined by way of the Gregorian calendar.

We make a distinction between dates and periods. If the reference is an exact date (e.g. a birth date), it is not necessary to define a period. When there is some degree of fuzziness about a date, we include the period in which the date is situated.

In principle databases will provide estimates of dates in case of missing values. Databases must describe how they have estimated their dates in the METADATA table by providing an explanation for all their values used with the field *Estimation*.

Each *Time Stamp* consists of the following elements (or fields):

Date_type Type of each date

Event Date of an event, observed at the moment of the event itself
Reported Date of an event reported in a later source
Declared Date at which point in time or period a certain attribute is valid (like 'married' or some occupational title)
Assigned Date or period assigned by the database administrator, for example to make explicit the period(s) that a certain person could be traced down in the archives.

Estimation Type of the estimation of the date or period, except for the first three values these values are defined in the METADATA table by each database itself.

| | |
|---|---|
| Exact | Exact date |
| Month_year | Month and year |
| Year | Only the year |

Database specific values may be:

| | |
|---|---|
| Middling | Middling of a period |
| Age_based | Period of birth based on age and date of source<br>For example: If you know an age of 25 at the 28th of February 1860,<br>then you know that the person is born between 1st of March 1834 and the 28th of February 1835. |
| Etcetera | |

*An exact date consists of five fields*:

| | |
|---|---|
| Day | Day number |
| Month | Month number |
| Year | Year number |
| Hour | Hour (0 to 23 hours) |
| Minute | Minutes (0 to 59 minutes) |

*A period is defined by six fields:*

| | |
|---|---|
| Start_day | Start day number |
| Start_month | Start month number |
| Start_year | Start year number |
| End_day | End day number |
| End_month | End month number |
| End_year | End year number |

Missing This field explains why a date or part of a date is missing (Mandemakers and Dillon 2004), and eventually had to be estimated.

| | |
|---|---|
| Unavailable | No data available (in the source) |
| Unreadable | Data is not readable (in the source) |
| Anonymized | The date is anonymized (by the database) |
| Private | The  date is not available for reasons of privacy, not included in the database |
| Time invariant | It is not necessary to have a date (e.g. sex) |
| Unknown | Unknown (in the database) why a value is unknown |

Day, month, and year are included as separate columns, rather than relying on the built-in date formats used by various software packages, to avoid incompatibilities between systems.

The values of *Date_Type*, *Estimation* and *Missing* may be further explained in the metadata registry.

Scheme 3 presents a decision scheme for distinguishing between the several options for defining the values on the variables *Date_type* and *Estimation* of the timestamp.

A time stamp can be developed much further, including atomic precision but for historical databases a precision in minutes seems to be sufficient (compare J. Benzler & S. Clark 2005).

Scheme 3 *Practical guidelines for defining dates and periods in the Timestamp*

| | | | | |
|---|---|---|---|---|
| 0 | Is the date or period assigned by the database administrator or derived from the sources? For example, periods of observation which are not directly given in the sources are 'assigned' dates. | | | |
| | Yes | *Date_Type* | Assigned | |
| | | *Estimation* | All possible values; in case of estimation of a date, the range in which the date is estimated may be given in the period fields. | |
| | No | Go to step 1 | | |
| 1 | Does the date describe an event, observed at the moment of the event itself (like the date of a divorce in a divorce certificate or the date of a birth in a birth certificate)? | | | |
| | Yes | In this case no estimation of a date is allowed: | | |
| | | *Date_Type* | Event | |
| | | *Estimation* | Exact | |
| | | Go to step 4 | | |
| | No | Go to step 2 | | |
| 2 | Does the date describe an event from an earlier time, reported in a source compiled after the event occurred (like the date of a marriage in a certificate of divorce or a marriage date in a population register)? | | | |
| | Yes | Is the date an exact date? | | |
| | | Yes | *Date_Type* | Reported |
| | | | *Estimation* | Exact |
| | | No | *Date_Type* | Reported |
| | | | *Estimation* | All possible values; in the period fields provide the range in which the date is estimated. |
| | | Go to step 4 | | |
| | No | Go to step 3 | | |
| 3 | Is the date a moment in time at which a certain attribute is valid (like the status 'married' or some occupational title)? In this case, you do not know when the attribute took this value. | | | |
| | Yes | Is the date an exact date? | | |
| | | Yes | *Date_Type* | Declared |
| | | | *Estimation* | Exact |
| | | No | *Date_Type* | Declared |
| | | | *Estimation* | All possible values; in the period fields you have to include the range in which the date is estimated. |
| 4 | Does the source (or combination of sources) implicitly or explicitly include a second date for the same attribute? Implicit dates may be related to the end of observation in a source, like a population register which is valid for a period of time. | | | |
| | Yes | Follow steps 1-3 above to create a second record in which the timestamp includes the beginning/end of the period. Choose the appropriate *Date_type* (Event, Reported, or Declared). If a combination of sources is used to determine a date, mention them all in the *Source* field. | | |
| | No | End | | |

## ACKNOWLEDGEMENTS

This article is a result of several workshops aimed at solving the problem of sharing historical life history data for demographic research. The problematic character of longitudinal historical data was raised in November 2003 at a meeting in Montréal (Dillon and Roberts 2006). The second workshop, titled 'Disseminating and Analyzing Longitudinal Historical Data', took place in Amsterdam, February 2006. On May 1-2, 2008, the Inter-university Consortium for Political and Social Research (ICPSR) hosted a planning group to continue working on the results gained in Amsterdam. This resulted in a model for data sharing, which was presented to an open meeting of historical databases at the Social Science History Association (SSHA) meeting in Miami, October 22, 2008, which became version 1 of the IDS (Alter, Mandemakers and Gutmann 2009). This version was discussed in workshops held at the Social Science History Association conferences in Long Beach (2009), Chicago (2010), Boston (2011) and Vancouver (2012). Since 2011 the enterprise has become part of the European Historical Population Samples network, resulting in workshops in Umea (2012) and Lund (2013).

Many people have contributed to discussions concerning the IDS. Participants of the initiating Amsterdam workshop were George Alter, Tommy Bengtsson, Onno Boonstra, Jerome Bourdieu, Bertrand Desjardins, Lisa Dillon, Peter Doorn, Sören Edvinsson, Joseph Ferrie, Eilidh Garret, Myron Gutmann, Lex Heerma van Voss, René van Horik, Alice Kasakoff, Hiroshi Kawaguchi, Lionel Kesztenbaum, Jan Kok, Marco van Leeuwen, Kate Lynch, Kees Mandemakers, Janet McCalman, Géri Mineau, Michel Oris, Frans van Poppel, Alice Reid, Steven Ruggles, Gunnar Thorvaldsen, Marc Tremblay, Hanne Van Baelen and Hélène Vézina.

The following persons took part during the follow-up conferences: George Alter, Trygve Andersen, David Barss, Peter Baskerville, Tommy Bengtsson, Anders Brändström, Jerome Bourdieu, Joey Burton, Antonio Camara, Marianne Caron, Nanne Floor Clausen, Bertrand Desjardins, Chris Dibben, Lisa Dillon, Stefan Dormans, Martin Dribe, Sören Edvinsson, Marius Eppel, Marianne Erikstad, Myron Gutmann, Mats Hayen, Maria Hiltonen, Kris Inwood, Alice Kasakoff, Lionel Kesztenbaum, Rebecca Kippen, Jan Kok, Satomi Koruso, Dahlia Leinaike, Susan Leonard, Roger Lund, Joe Malletores, Kees Mandemakers, Koen Matthijs, Hamish Maxwell Stuart, Janet McCalman, Emily Merchant, Géri Mineau, Stefan Oberg, Luciana Quaranta, Paul Rotering, Steven Ruggles, Rembrandt Scholz, Wen Shan Yang, Sandra Silcot, Ken Smith, Patrick Svensson, Ken Sylvester, Gunnar Thorvaldsen, Matthijs Vandezande, Hélène Vézina, Annika Westberg and Leigh Williamson.

A special thanks to Emily Merchant who made wonderful accurate minutes of the often complicated discussions at these workshops. The IDS clearing committee consists of George Alter, Kees Mandemakers and Annika Westberg.

The work was made possible by grants from the Netherlands Organization for Scientific Research (NWO), Humanities (Internationalisation, 236-053-004), the Inter-university Consortium for Political and Social Research (ICPSR) in Ann Arbor, the Demographic Data Base, Umeå University, Sweden (DDB), the International Institute of Social History (IISH), Amsterdam and since 2011 the European Science Foundation by way of the EHPS-network (see http://www.ehps-net.eu/).

## REFERENCES

Alter, G., Mandemakers, K. & Gutmann, M. (2009). Defining and Distributing Longitudinal Historical Data in a General Way Through an Intermediate Structure. *Historical Social Research,* 34 (3), 78-114.

Alter, G. & Mandemakers, K. (2011). The Intermediate Data Structure (IDS) for Longitudinal Historical Microdata, version 2, dated 13 march 2011. *Working paper published on the EHPS collaboratory.*

Alter, G. & Mandemakers, K. (2012). The Intermediate Data Structure (IDS) for Longitudinal Historical Microdata, version 3, dated 12 July 2012. *Working paper published on the EHPS collaboratory.*

Beaumont, R. (2007). An Introduction to Entity Relationship Diagrams (ERDs), version 5.

Benzler, J. & Clark, S. (2005). Toward a Unified Timestamp with Explicit Precision. *Demographic Research,* 12, 107-140.
DOI: 10.4054/DemRes.2005.12.6

Brettell, C. (2003). *Anthropology and Migration. Essays on Transnationalism. Ethnicity and Identity.* Walnut Creek: Altamira Press.

Dillon, L. & Roberts, E. (Eds.) (2006). Special Issue on Longitudinal and Cross-Sectional Historical Data: Intersections and Opportunities. *History & Computing,* 14 n1/2.

Gregory, I. N. & Ell, P. (2007). *Historical GIS: Technologies, Methodologies and Scholarship*. Cambridge & New York: Cambridge University Press.

Hammel, E. A. & Laslett, P. (1974). Comparing Household Structure over Time and Between Cultures. *Comparative Studies in Society and History,* 16 (1), 73-109.
DOI: 10.1017/S0010417500007362

Kelly Hall, P., McCaa, R. & Thorvaldsen, G. (Eds.) (2000). *Handbook of International Historical Microdata for Population Research*. Minneapolis: Minnesota Population Center.

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

Quaranta, L. (2013). Making an Extraction from the Scanian Economic Demographic Database, *PowerPoint Presentation on the EHPS collaboratory*.

Ruggles, S., Sobek, M., Alexander, T., Fitch, C. A., Goeken, R., Kelly Hall, P., King, M. & Ronnander, C. (2008). *Integrated Public Use Microdata Series: Version 4.0 [Machine-readable database]*. Minneapolis: Minnesota Population Center.

Schürer, K. (2007). Creating a Nationally Representative Individual and Household Sample for Great Britain, 1851 to 1901 – The Victorian Panel Study (VPS). *Historical Social Research,* 32 (2), 211-331.

Southall, H., Mostern, R., Berman, M. L. (2011). On Historical Gazetteers. *International Journal of Humanities and Arts Computing,* 5 (2), 127-145.
DOI: 10.3366/ijhac.2011.0028

Stead, W. W., Hammond, W. E. & Straube, M. J. (1982). A Chartless Record – Is it Adequate? *Proceedings of the Annual Symposium on Computer Application in Medical Care,* Nov 2, 89–94.

# Appendix A Track of all changes between the several versions of the IDS

Smaller improvements like improving or correcting the text are not included unless they imply a change of strategy at some point in the IDS.

| Version | Change |
|---|---|
| 2.0 | New approach of the CONTEXT, introduction of CONTEX_CONTEXT table, see Appendix B for a complete explanation of the change and reasons for the change; this implied also rewriting of the chapters relating to CONTEXT. |
| 2.0 | Making the field *Relation* in the table INDIV_INDIV reciprocal. |
| 2.0 | Including all examples of records in the tables in the text itself (instead of the Appendix). |
| 2.0 | Removing the first more theoretically part of the original IDS-article. |
| 2.0 | Including decision schemes on relations (what to put in which table); metadata and the timestamp. |
| 2.0 | Introduction of the field *Value_Id_C* in the INDIV table to create a direct relationship with the CONTEXT table. |
| 2.0 | Introduction of the field *Id_D_Description* in the METADATA table. However this was never implemented in the table itself, the whole was reconsidered in 3.0. |
| | |
| 3.0 | Including two new fields in the METADATA table (*Id_D_Explanation* and *New)* and checking and improving the text on metadata (including the removal of some inconsistencies). |
| 3.0 | Including a graphic form of the decision scheme on INDIV_CONTEXT. |
| 3.0 | Introducing a new value for the field *Data_Type* of the TIMESTAMP: Assigned and adjusting the scheme on the TIMESTAMP for this new value. |
| 3.0 | Including a solution for geometric data of which the strings are too long to be included in the IDS. |
| 3.0 | The database identifying field *Id_D* is destined to handle the versioning per record of the several IDS databases a data archive may disseminate. |
| 3.0 | Checking and improving the text on inconsistencies (especially the tables with examples of records). |
| 3.0 | The description of the METADATA table (par. 3.4) has been improved and the format of the table itself has been brought in correspondence with the description. This includes two new fields: *Id_D_Explanation* and *New*. The last field is created to document the changes in the METADATA table itself. |
| | |
| 4.0 | Inclusion of a type for stillbirths (and discussion of the peculiar nature of this type). |
| 4.0 | Including a new section 3.2.2 for the handling of the start date and end date of observations. |
| 4.0 | Explanation why locations and dates have two records in the case of events (instead of one). |
| 4.0 | Introduction of extended IDS and a new field in the Metadata table, Extract, for the name of extraction software (extension of section on metadata). |

## Appendix B Handling of the context in version 1 of the IDS

As described in section 3.3.2 contexts are often hierarchical or nested. There are several ways to represent context hierarchies in the IDS.

Version 1 of the IDS described three approaches for building the CONTEXT and INDIV_CONTEXT tables:

1 Each individual is linked with records for all characteristics of all levels of which a context exist.
2 Each individual is linked with each layer of the context he occupies and all characteristics of that layer are described only once in the CONTEXT table (the layering is documented in the METADATA table).
3 Each individual is linked with only the lowest layer of the context he occupies and all characteristics of that layer are described only once in the CONTEXT table including records to define the layering (by way of defining the nearest higher level).

Option 1 involved repetition in the database, because all individuals need the same attributes for all parts of the context again and again. Also when contextual attributes change the whole has to be repeated. This results in an enormous amount of records and therefore this approach is more a theoretical one and not useful in the practice of large databases.

By using a CONTEXT_CONTEXT table a choice has been made for a strategy in which option 2 and 3 are combined. Since each address, neighbourhood, municipality etc. will be identified by the value on the field *ID_C*, their attributes will be described only once, and information will not be repeated in the CONTEXT table. And the layering of the attributes which is problematic in both options has been made explicit in the CONTEXT_CONTEXT table.

All records in the CONTEXT table need a time stamp otherwise the timestamp of the record in INDIV_CONTEXT will define the period. A change in an attribute of e.g. a municipality would result in only one new time stamped attribute, which is associated with the field *ID_C* of the municipality.

Note that in the end, we want the attributes of all levels of the hierarchy (e.g. address, neighbourhood, municipality) to appear in separate columns on every individual record in the rectangular dataset that is used for analysis. Descriptions of higher level contexts will always be repeated in the rectangularized file, even if they are not repeated in the IDS.