Antropol. Port., 4-5, 1986-87: 109-124

Using the data base management system SIR to link demographic data from the basque community of Sainte-Engrace, 1800-1980

A. RODRIGUEZ HERNANDORENA [1]

RESUMO

O sistema de base de dados SIR (Scientific Information Retrieval) tem sido utilizado para relacionar registos civis (nascimentos, casamentos e mortes) e registos cadastrais (posse de terra) no período de 1800-1980.

Discutem-se no artigo algumas das vantagens do SIR em relação a outros métodos de relacionamento de registos utilizados no passado: 1) a sua capacidade para tratar qualquer tipo de dados requerendo tratamento hierárquico complexo, 2) a possibilidade de transformar a estrutura de base de dados em qualquer momento, 3) a sua capacidade em gerar ficheiros formatados para SPSS e SAS, 4) o facto de SIR existir disponível como «pacote» de software para IBM XT e AT ou qualquer micro IBM compatível (requerendo um mínimo de 10 Mb em disco rígido).

Palavras-chave: Reconstituição de famílias; Base de dados; SIR.

ABSTRACT

The Data Base System SIR (Scientific Information Retrieval) has been used to link civil records (births, marriages and deaths) and cadastrial records (land ownership) for the period 1800-1980. The paper discusses some of the advantages of SIR in relation to other record linkage methods used in the past: 1) its ability to handle any set of data requiring complex hierarchical treatment, 2) the possibility of modifying the data base's structure at any required moment, 3) its ability to generate SPSS and SAS formatted files, 4) SIR is available as a software package on IBM XT and AT or any IBM compatible micro (minimum requirement 10Mb hard disk).

Key-words: Record linkage; Data base; SIR.

INTRODUCTION

This paper summarizes the strategy followed during the 10 months of record linkage spent reconstituting individual and family life events in the

[1] Department of Biological Anthropology, Oxford University.

© 1987, Instituto de Antropologia, Univ. Coimbra.

Basque Pyreneean Community of Sainte-Engrace for the period 1800-1980. The Data consists 4538 birth records (each birth record is 173 characters long and 16 variables); 1026 marriage records (each marriage record is 376 characters long and has 32 fields), and 3327 death records (each death record is 184 characters long and has 16 fields). The software used to perform the record linkage was the Data Base Managemant System SIR [2]. The strategy employed in this study was designed by D. Doulton *et al.* (1986). The method described in this paper is suitable for the linkage of any set of events.

Data entry and storage

Births, marriages and deaths were entered on a BBC micro computer. No input programs were used, because the data were entered as text. The VIEW word processing package, available on the BBC, was used to correct mistakes while typing. At the end of the data entry stage, the BBC files were transfered to the Oxford University Computing Centre's mainframe, where the rest of the work has was carried out. Figure A. 1 shows the variables included in each birth, marriage and death record. To allow any individual record to be traced back to its source document (birth, marriage or death) each set of information is given a unique identifier (NEWID); this identifier is a 5 digit integer where the first digit for the births is always a 1, the first digit for a marriage a 2, and the first digit for a death is always a 3. So, for example the three first birth records in the births file will have NEWID's 10001, 10002, and 10003. The three first marriages NEWID's will be 20001, 20002, and 20003.

The variables describing the names and surnames of individuals as well as the names of the houses were not codified; their original alphabetic nature was kept. Figure A. 2 shows three samples extracted from the master files.

Record linkage using SIR

The three principal steps involved in any linking operation are the searching step, the matching step and the storing step. The three steps operate in very different ways but share in common an important factor: the basic structure of the material they work on, that is the three master files, is always the same. Bearing this principle in mind, it was decided to load the three files into a data base where the three steps could be carried out benefiting from the advantages provided by the structure of the data base.

The data base system used is SIR (Scientific Information Retrieval). SIR is a hierarchical data base system in its physical form: each record is stored following the record to which it relates. Thus the unique identifier (NEWID) for one individual entry (a case in SIR) will point to several RECORDS concerning that entry, each of which, in turn, may contain several variables.

---

[2] The SIR software manuals can be purchased at the European SIR headquarters at: STATUS GmbH, Postfach 45 01 49, Goerzallee 5, D — 1000 BERLIN 45.

Field	Length	Cols	Label
1	4	1- 4	IDENTIFICATION NUMBER
2	11	5- 15	DATE OF BIRTH
3	20	16- 35	NAME
4	18	36- 53	FIRST SURNAME
5	16	54- 69	SECOND SURNAME
6	11	70- 80	SEX
7	16	81- 96	FATHER's NAME
8	22	97-118	FATHER's SURNAME
9	5	119-123	FATHER's OCCUPATION
10	16	124-139	MOTHER's NAME
11	23	140-162	MOTHER's SURNAME
12	3	163-165	MOTHER's OCCUPATION
13	19	166-184	HOUSE
14	11	185-195	BIRTH STATUS (legitimate/illegitimate)
			Variables in the marriage record
1	4	1- 4	IDENTIFICATION NUMBER
2	12	5- 16	DATE OF MARRIAGE
3	21	17- 37	GROOM's NAME
4	27	38- 64	GROOM's FIRST SURNAME
5	24	65- 88	GROOM's SECOND SURNAME
6	12	89-100	GROOM's AGE
7	2	101-102	GROOM's OCCUPATION
8	2	103-104	GROMM's BIRTHPLACE
9	2	105-106	GROMM's BORN legitimate/illegitimate
10	15	107-121	GROOM's FATHER's NAME
11	27	122-148	GROOM's FATHER's SURNAME
12	9	149-157	GROMM's MOTHER's KNOWN
13	13	158-170	GROOM's MOTHER's NAME
14	24	171-194	GROOM's MOTHER's SURNAME
15	12	195-206	GROOM's MARRIAGE RANK (first/second....)
.			
.			idem for the bride
.			
30			
			Variables in the death record
1	4	1- 4	IDENTIFICATION NUMBER
2	10	5- 14	DATE OF DEATH
3	4	15- 18	DATE OF BIRTH
4	2	19- 20	SEX
5	17	21- 37	NAME
6	31	38- 68	FIRST SURNAME
7	22	69- 90	SECOND SURNAME
8	3	91- 93	AGE
9	23	94-116	HOUSE
10	22	117-138	FATHER's NAME
11	13	139-151	MOTHER's NAME
12	2	152-153	DEATH STATUS (married/single etc...)
13	12	154-165	SPOUSE's NAME
14	23	166-188	SPOUSE's SURNAME

Fig. A1 — Variables in the birth record

```
100011   114.01.1793JEAN          OLHATCEBERDE     ILHARRISCAPE     1PIERRE
+        ILHARRISCAPE         1JEANNE          ILHARRISCAPE          6LOGE
+              1*
100021   218/01/1793ENGRACE       BORTHIRY         BORDALEKU        2PIERRE
+        BORTHIRY             1MARIE           BORDALEKU             0GOROST
+ IOL          1*
100031   324/01/1793JEAN          SABUQUY          ETCHEBER         1JEAN PIE
+ RRE    SABUQUY              1ANNE            ETCHEBER              0SABUQU
+ Y            1*
```

Sample of death records

```
305013622903.04.18351832 1JEAN          CHUBURU                          BORDALEK
+ U              3BORDALEKU             DOMINIQUE      ENGRACE      1
+                       *
305023623008.04.18351788 2MARIE         CHOUHOURT BOURG
+                47JAURGAIN                                          2JEAN
+    POURTAU                 *                                        BARATCE
305033623130.04.18351833 2MARIE         FUSILLES
+                2CARRIGUIRY            BERTRAND       MARIE ANNE   1
+                        *
```

Sample of marriage records

```
200012100008.01.1793PIERRE                ASTARINHANDY                ELGUEBARNE
+             30 0 1 1GRATIEN         ASTARINHANDY                0MARIE        E
+ LGUEBARNE                 1BRIGITTE        BAGOLLE                      ACCOCE
+                27 0 1 1JEAN             BAGOLLE                     0ANNE
+       ACCOCE                   1ASTARINHANDY   ACCOCE          *
200022100105.02.1793JEAN                  BORDALEKU                   HUSTU
+             19 0 1 1PIERRE          BORDALEKU                   0ANNE        H
+ USTU                    1ENGRACE         DRONDE                       AGARAS
+                27 0 1 1JOSEPH           DRONDE                      0CATHERIN
+ E     AGARAS                   1HUSTU           AGARAS          *
200032100205.02.1793PIERRE                BIDEGARAY                OLHATCEBERDE
+             31 0 1 1PIERRE          BIDEGARAY                   0MARIE        O
+ LHATCEBERDE                1MARIE           ELGART                       ESPEL
+                27 0 1 1DIEGO            ELGART                      0ENGRACE
+       ESPEL                    1OLHATCEBERDE   ESPEL           *
```

The stars (*) indicate the end of each record.

Fig. A2 — A sample of birth records

A major advantage of SIR is that, althought it has a hierarchical physical model, one is not restricted to hierarchical access. The logical model of the data base allows hierarchical, relational and network access. It is the network facility that is most useful for record linkage; since from any position in the hierarchy it is possible to re-enter the hierarchy at any other point whilst retainning the original position for subsequent continued search.

SIR is available on most mainframe computers. It is also available on IBM and IBM compatible micros. The requirement for SIR to be run on an IBM micro is having a minimum of 10Mbytes of hard disk.

In summary, SIR was selected for the following reasons:

— Very direct and friendly interaction with the data base. The variety of means for inputting, modifying, deleting and in general controlling the contents of the data file are very straightfoward.

— Possibility of modifying the data base's structure at any point during the study.

— No limitations on the complexity of the database's structure. Althought SIR has a hierarchical physical model, one is not restricted to hierarchical access, the logical model of the data base allowing also relational and network access.

— The data retrieval capabilities of SIR allows performing both simple and highly complex retrieval in a straightfoward manner.

— The data description in SIR has been patterned after syntax of SPSS, and the SIR programming language shares many similarities with it.

— SIR can interface with the most widely used statistical systems SPSS and SAS.

Although it is possible to change the structure at any point during the study, an initial structure had to be provided. The three different sources, births, marriages and deaths were stored in three separate compartments or record types. The births were allocated to record type 1, the marriages to record type 2, and the deaths to record type 3. The decision to refer to births as record type 1, marriages 2 etc..., is an entirely arbitrary one.

Before loading the data into the data base we must provide a structure for the data base to know where to alocate the variables, records etc... The case definition commands supply information about the general structure of the data base.

The record definition commands are used to describe the contents and structure of individual record types. Figure 1 shows the case definition file and figure 2 shows the record definition command file for the birth records. Once the basic structure for the record types is created, the birth, marriage, and death records are loaded into the data base.

The searching step

The general aim of the searching step is to bring potencially linkable records together, first for comparison, and secondly and eventually, for matching.

```
CASE ID          PREC
MAX REC TYPES    15
RECTYPE COLS     6
N OF CASES       10000
RECS PER CASE    20
MAX REC COUNT    10
MAX KEY SIZE     3
MAX INPUT COLS   425
COMMON VARS      PREC (I,5)
```

The Record Definition Commands File.

```
RECORD SCHEMA    1,BIRTH
MAX REC COUNT    1
DATA LIST        (1) / 1
                 PREC 1-5 (I)
                 BREC 7-10 (I)
                 BDATE 11-20 (A)
                 XBDATE 11-20 (A)
                 BN 21-37 (A)
                 BS1 38-54 (A)
                 BS2 55-70 (A)
                 BSEX 71-72 (I)
                 BFN 73-87 (A)
                 BFS 88-109 (A)
                 BFOCC 110-111 (I)
                 BMN 112-126 (A)
                 BMS 127-150 (A)
                 BMOCC 151-152 (I)
                 BH 153-171 (A)
                 BSTATUS 172-173 (I)
                 BHN 174-178 (I)
DATE VARS        BDATE ('DDIMMIYYYY')
MISSING VALUES   BDATE ('          ')
END SCHEMA
```

Fig. 1 — The Case Definition Commands File

But prior to the actual searching operation decisions have to be made on the linkage criteria, or control variable or variables to be used in the operation.

Choosing a control variable

The major problem in most record linkage exercises is to determine a set of linkage criteria or control variable which will maximize the accuracy of the generated links. At least five alphabetic variables were simultaneously present in more than 60% of all births, marriages and deaths records. It was though that a gradual variable reduction approach would suit the situation best. If we take for example the case of a woman with birth record number 10344. Her name is MARIE. If we use her name to search for her corresponding and death records we will find hundreds of candidates, because there are many MARIE's in the data base. In other words if we only use the variable NAME as the linkage criteria we are going to gather too many candidates for each match. But there are other variable choice alternatives. Her name is MARIE, her mother's name is ISABEL, her father's name is PIERRE, the birth record also gives us her first surname ELGOYHEN, and her second surname AGARAS. Her birth record provides 5 alphabetic fields that can be put together in one single variable that we will call STRING and will look like:

STRING = MARIE ISABEL PIERRE ELOGOYHEN AGARAS

ANNE	ANNE ENGRACE JEAN ACCOCE ETCHECOPAR
ANNE	ANNE MADALEINE PIERRE JONNET AGARAS
ANNE	ANNE MARIE PIERRE JONNET AGARAS
ANTOINE	ANTOINE HELENE PAUL GORRY ELGOYHEN
ANTOINE	ANTOINE PIERRETE PAUL ETCHEBER ELICHALT
AMELIE	AMELIE ANNE JEAN HALCAREN SOCCOROS

Fig. 2 — Variables NAME and STRING sorted alphabetically

Figure 2 shows two columns of birth variables sorted by NAME and STRING.

Although this abstract and newly created variable STRING, does not correspond to a real name or surname, it accomplishes the funtion required in this exercice. It is behaving as a good alphabetic identifier. The number of variables used to make up the control variable varies according to the degree

of stringency applied to the searching operation. In the beginning we are very stringent, and thus use up to 5 variables. These 5 variables generally are:

EGO'S NAME EGO'S MOTHER NAME EGO'S FATHER NAME—
EGO'S SURNAME 1 EGO'S SURNAME 2

Latter in the searching process we will go down to using 4, and 3 variables.

Searching through the birth, marriage and death records

We can move foward to explain how the searching operation was performed. An hypothetic example will be used to illustrate the process. We can write a SIR procedure (see figure 3) to move and sort 5 alphabetic variables for a reduced sample of birth, marriage, and death records. The 5 variables are written out in the following order:

EGO'S NAME EGO'S FATHER NAME EGO'S MOTHER NAME—
EGO'S FATHER SURNAME EGO'S MOTHER SURNAME

If SIR sorts alphabetically the records we get the following list

3 0034	ANNE MARIE AGARAS ELICHIRY
2 0245	ANNE ANGEL ENGRACE LAXALT BORTHIRY
1 0176	ANNE ANTOINE MARIE ACCOCEBERRY TOUSTAU
3 0542	ANNE ANTOINE MARIE ACCOCEBERRY TOUSTOU
1 2389	ANNE PIERRE THERESE ELGOYHEN ELICHALT
2 0633	ANNE PIERRE THERESE ELGOYHEN ELICHALT
2 0126	ANNE PIERRE CATHERINE ELGOYHEN ELICHIRY
1 0523	ANTOINE MARCEL ENGRACE AGARAS CONSTANTIN
3 2199	ANTOINE MARCEL ENGRACE AGARAS CONSTANTIN
1 3822	ANTOINE MARCEL ENGRACE ETCHEBER UNGURATURU
2 0721	ANTOINE MARCEL ENGRACE ETCHEBER UNGURATURU
3 1335	ANTOINE MARCEL ENGRACE ETCHEBER UNGURATURU
2 0356	ANTOINE PAUL MARIE CHUBURU ETCHECOPAR

Column one contains a mixture of 1, 2, and 3 values, because the three types of information (births, marriages, and deaths) have now been put together. In the second column we have the unique identifier for each record followed by the list of sorted STRING's. The following records seem to have found matching pairs or triplets.

Rec type 1 0176 (BIRTH) with Rec type 3 0542 (DEATH)

Rec type 1 2389 (BIRTH) with Rec type 2 0633 (MARRIAGE)

Rec type 1 0523 (BIRTH) with Rec type 3 2199 (DEATH)

Rec type 1 3822 (BIRTH) with Rec type 2 0721 (MARRIAGE)
with Rec type 3 1335 (DEATH)

The matching step

When pairs or triplets of records are brought together for comparison, decisions must be made as to whether these are to be regarded as linked, not linked, etc... It is also desirable to make a high proportion of those decisions automatically by the use of some SIR procedures. In most studies involving record linkage the matching step and the storing step are executed simultaneously, but with SIR the strategy is different. In the first part, the links are made and stored in a mainframe, external, file. In the second part the links are read back into the data base. The summary of the steps used with the SIR strategy are:

A — Processing the birth, marriage and death records and computing the required STRING variable.
B — Sorting the above records by the variable STRING.
C — Making the links.
D — Writting out the mainframe a file containing the ID's of the above links.
E — Reading back the links into the data base.

Steps A and B have been described earlier in this paper (choosing the control variable). Step E will be described in the storing step. The other two steps, C and D can be carried out using the SIR report generator. The report generator works on the records extracted from the data base (steps A and B) performing specific actions depending on whether the value of STRING remains the same or changes. The SIR report generator will use following algorithm:

a) When a new value of the control variable STRING is encountered, write the NEWID (NEWID1) and store its value.

b) For each subsequent record with the same value of the control variable, write its NEWID (NEWID2) and the above NEWID (NEWID1), followed by the NEWID1 alone on the next line.

c) When the value of the control variable changes again, write the NEWID1 twice.

Figure 4 shows the SIR report used to perform the above algorithm.

The storing step

In the matching step, the SIR report generator was used to compare the values of the alphabetic variable STRING, and to make some decisions on whether they should be linked or not. Ideally, each cain should contain a birth, a marriage, and a death record, or for those who died single, only a

birth and a death record. The following figure shows the different acceptable combinations of records:

BIRTH ------- MARRIAGE ------- DEATH		(1-2-3)
BIRTH ------- MARRIAGE		(1-2)
BIRTH ---------- DEATH		(1-3)
MARRIAGE ------- DEATH		(2-3)

For married people the ideal combination to find is (1-2-3), and for single individuals (1-3). But when we run the report generator on the data we found that some chains presented the following combinations:

1-1-2	1-3-3-3
1-2-2-3	1-2-3-3-3
2-2-3-3	

This means that there are several birth, marriage and death records with identical STRINGs, and so the computer is creating chains with more records than required. This also indicates that more knowledge on the above records is required in order to make decisons on what is to be linked to what. Here we can indicate again that the number to multi-records chains we obtain, very much depends on the degree of stringency we put on the variable STRING. In other words, if STRING is made up of five different elements (ego's name, ego's father name, ego's mother name, ego's first name, ego's second surname) the chances of getting identical STRINGs are smaller than if STRING is made up of a single variable, say ego's name. Because the report generator will not always produce perfect link-chains, a new SIR procedure must be writen that will take all these factors in account.

The SIR procedure to read the «good» linsk back into the date base

This procedure performs simultaneously the following tasks:

A — Reads through the links file produced by the SIR report generator (this is an external or mainframe file, so at this point we are not within the data base yet).

B — Skips all the chains where the NEWID is linked to itself.

C — If the chain involves more than one NEWID there is potencial linkage and so, the procedure must carry out further exploration:

1. — It must make sure the records involved in the chain present one of the following combinations:

BIRTH — MARRIAGE — DEATH	(1-2-3)
BIRTH — MARRIAGE	(1-2)
BIRTH — DEATH	(1-3)
MARRIAGE — DEATH	(2-3)

Any chain containing a different combination of records is not accepted. For example 1-1-2, or 2-2-3-3 etc...

D — If the chain belongs to any of the acceptable combinations, the dates of the records involved in the chain are looked at to make sure last occuring event took place after the first occuring event. In other words, the death should always take place after marriage, and the marriage always after the birth. If the chain meets the record combination requirements but, the dates are impossible, the chain is skipped.

E — Finnally, the SIR procedure the «good» links into the data base.

Bringing the links back into the data base

Once the SIR procedure selects «good» links, it has to find a place in the data base to store them. It was decided to create two different compartments, one for female individuals, and another one for male individuals. It was arbitrarily decided to allocate female individuals to record type 8, and male individuals to record type 9. Records types 8 and 9 have the same structure; each record is made of one line containing the three ID numbers included in each link: the record's own ID (NEWID), the linked ID (LID), and the first ID (FID). Figure 5 shows how the NEWID, LID and FID guide us through the linked events.

Explaining the contents of record types 8 and 9

If for example we look at any woman in compartiment or record type 8, we can find:

a) A woman's birth record, linked to her death record. Because each link needs a triplet of ID's this woman's links look like (her birth records NEWID = 1 0254, her death record = 3 2080):

B	D	B				
10254	32080	10254	BIRTH	DEATH	BIRTH	1-3-1
32080	10254	10254	DEATH	BIRTH	BIRTH	3-1-1
D	B	B				

The first ID in the chain is her own birth record (FID) 10254. In the second line her death record (NEWID) 32080 is linked back to her own birth record (LID) which is also the first ID in the chain, 10254.

b) A woman's birth record linked to her marriage record.

B	M	B				
13721	20412	13721	BIRTH	MARRIAGE	BIRTH	1-2-1
20412	13721	13721	MARRIAGE	BIRTH	BIRTH	2-1-1
M	B	B				

This time the structure of the chain is identical, the only difference being the nature of the records

c) A woman's marriage linked to her death record.

M	D	M				
20622	31633	20622	MARRIAGE	DEATH	MARRIAGE	2-3-2
31633	20622	20622	DEATH	MARRIAGE	MARRIAGE	3-2-2
D	M	M				

d) Finally we can find a woman that has the three events linked within the same chain.

B	M	B				
14176	20522	14176	BIRTH	MARRIAGE	BIRTH	1-2-1
M	D	B				
20522	31421	14176	MARRIAGE	DEATH	BIRTH	2-3-1
D	B	B				
30421	14176	14176	DEATH	BIRTH	BIRTH	3-1-1

This last chain represents one of the most complicated triplet structure that any male or female individual can have in the data base. The strategy used to link the mother's marriage record to her children birth records is identical to the one used to link birth, marriage and death for one person. A STRING variable made up with the husband's name surname and the wife's name and surname is computed for each marriage record. The variable STRING computed for each birth record includes ego's father's name and surname and ego's mother's name and surname (four variables as in above marriage). All the marriages and births are moved on to the same file where the records are sorted by STRING. As a result each record clusters arround the parents marriage record (see Figure 6). These NEWID'S are brought back into the data base and sorted at record type 7.

«The pointers». What are they and what do they do?

We are now in a position to say that the links are brought back into the data base, and are stored there as «pointers». In a situation like the present one, where a record «owns» several other records, and at the same it is «owned» by other records itself, the use of pointers is almost inevitable. Imagine the situation where a woman's marriage record is «owned» by:

1 — Herself when refering to her sequential event record.
2 — Her offspring when making the family links.
3 — Her own parents when making her links as an offspring.

```
RETRIEVAL
.PROCESS CASES

   PROCESS REC 1 BIRTHS
      MOVE VARS NEWID BN BS1 BS2 BFN BMN
      COMPUTE STRING=  (BN) + (BFN) + (BMN) + (BS1)
      PERFORM PROCS
   END PROCESS REC

   PROCESS REC 2 MARRIAGES
      MOVE VARS NEWID MHN MHFN MHS1 MHMN MHS2
      COMPUTE STRING=  (MHN) + (MHFN) + (MHMN) + (MHS1)
      PERFORM PROCS
   END PROCESS REC

   PROCESS REC 3 DEATHS
      MOVE VARS NEWID DN DS1 DS2 DFN DMN
      COMPUTE STRING=(DN)+(DFN)+(DMN)+(DS1)
       PERFORM PROCS
   END PROCESS REC

.END PROCESS CASES

WRITE RECORDS FILENAME='EXAMPLE1'/
     VARIABLES=NEWID STRING/
     FORMAT=(I5,3X,A97)/
     SORT=STRING/
END RETRIEVAL
```

Fig. 3 — SIR retrieval to write out and sort the NEWID or birth, marriage and deaths

```
RETRIEVAL
CALL FINDLINK.FULLBODY(<1>)
REPORT FILENAME='LINKS<1>'/SORT=NAM,PREC/NOPAGING/
BEFORE REPORT
SET LAST(0)
BREAK LEVEL 1 ,NAM
INITIAL BLOCK
COMPUTE LAST=PREC
WRITE   PREC ('ZZZZZZ')
DETAIL BLOCK
IFTHEN(LAST NE PREC)
.  WRITE   PREC ('ZZZZZZ') 4X LAST ('ZZZZZZ')
.  WRITE   PREC ('ZZZZZZ')
ENDIF
AT END BLOCK
WRITE   LAST ('ZZZZZZ') 4X LAST ('ZZZZZZ')
END BREAK LEVEL
```

Fig. 4 — SIR report to produce the links file

```
10055 (B) 20247 (M) 10055 (B)

20247 (M) 31042 (D) 10055 (B)

31042 (D) 10055 (B) 10055 (B)


10055  04.02.1813 ANNE, BAGOLLE, varB3, varB4,......varBn _______> 20247

20247  21.06.1838 ANNE, BAGOLLE, varM3, varM4,......varMn _______> 31042

31042  10.05.1860 ANNE, BAGOLLE, varD3, varD4,......varDn _______> 10055
```

B= birth; M= marriage; D= death; varM=a marriage variable; varB= a birth variable; varD= a death variable.

Fig. 5 — The «chain» of NEWID's linking the birth, marriage and death of an individual in the data base

```
20145 10.05.1824 ACCOCE JEAN MARIE
21548 02.08.1796 ACCOCE PIERRE ANNE
20124 13.07.1821 ACCOCEBERRY ANTOINE ENGRACE
10245 22.09.1823 ACCOCEBERRY ANTOINE ENGRACE
10278 12.10.1825 ACCOCEBERRY ANTOINE ENGRACE
20541 22.07.1924 ARHANCET PAUL HELENE

20247 21.06.1838 BAGOLLE LAURENT ANNE
10845 14.03.1839 BAGOLLE LAURENT ANNE      (1)
10962 22.09.1841 BAGOLLE LAURENT ANNE
10991 01.02.1843 BAGOLLE LAURENT ANNE

20874 02.08.1853 BARANTHOL BERNARD MARIE
21275 13.01.1953 BURGUBURU ANATOLE ENGRACE
```

(1) The marriage record with NEWID=20247 appears close to the NEWID's of it's three offspring: 10845, 10962 and 10991.

Fig. 6 — Marriage and birth records sorted by STRING

```
$
M/20456/15-02-1857
H/ASCARATEIL /JEAN PIERRE /04-06-1821 /15-02-1857 /25-01-1900
W/AMESTOY ARRACO /ELISABETH /09-12-1834 /15-02-1857 /01-02-1900
C/MARIE /2/22-12-1857 /* /08-02-1887
C/MADELEINE /2/28-02-1862 /* /01-12-1940
$
M/20459/22-11-1857
H/ASCONCILLO BASSAHON /DOMINIQUE /12-08-1819 /22-11-1857 /13-02-1872
W/BARANTHOL /MONIQUE /04-05-1820 /22-11-1857 /13-11-1902
C/JEAN /1/22-08-1859 /* /27-10-1859
C/ANNE /2/06-12-1862 /22-01-1889 /05-08-1940
C/ /1/29-05-1866 /* /*
$
M/20462/22-11-1857
H/ETCHECOPAR BORTHIRY /GUILLAUME /13-12-1831 /22-11-1857 /08-01-1892
W/BARANTHOL /ENGRACE /22-02-1838 /22-11-1857 /16-09-1907
C/ENGRACE /2/30-03-1859 /27-10-1885 /26-12-1948
C/BARTHELEMY /1/18-12-1860 /* /*
C/GABRIELLE /2/29-12-1862 /* /*
C/CATHERINE /2/29-04-1866 /* /09-03-1885
C/MARIE ANNE /2/24-07-1870 /* /*
C/JEAN /1/16-03-1872 /* /*
C/ENGRACE /2/04-07-1876 /* /*
C/GABRIELLE /2/18-06-1879 /* /22-02-1893
```

M=marriage; H=husband; W=wife; C=child; *=missing date; the dates between the slashes are the birth date, marriage date and death date in that order.
$= marks the end of one family and the begining of the next family.

Fig. 7 — An extract of the reconstituted family file

This list could become infinetely long if we had to move upwards or downwards through several generations. Copying again and again the entire marriage record of this woman for each link would require an enormous amount of machine space, and would make any searching operation immensely slow. The alternative of using pointers represents the solution to this type of problem. In this method, the use of pointers is combined with a clever links network where the reocrds can be followed very efficiently. The way the links have been stored in the data base enables a chain of links to be followed from its beginning to its end, and even more important, it enables the user to enter the chains at any point.

CONCLUSIONS

What amount of information does an individual own in the data base once all the links have been made? Lets take for example the ideal situation, where the individual has the three vital events linked and stored at record type 8 (she is a woman), and also has the chain of links to her offspring stored at record type 7. This individual can have access to the following information:

— Via her record type 8 has the three pointer pointing to her birth, marriage, and death records.

— Via her record type 7 she has the chain of pointers leading to her offsprings's birth records.

— Via her own birth record she can get into her siblings chain, or her mother's record type 7 information.

— Once she is within her record type 7 dealing to her offspring birth records she could get into each child and follow their links too.

In other words, this woman could travel as far as she wanted through her relatives records picking up any variable, as long as the chain of links exist. The reason why creating such a spectacular network of relations is possible without increasing the size of the data base, is that we use the pointers to guide us through the data. For example, it would only take 45 k to store all the information about somebody's birth, marriage and death instead of the 753 required if the three entire records had to be stored. Once the chain of links with the pointers or NEWID's are established it is only a matter of going to appropriate record type and extracting the information we want from there. Figure 7 shows an extract of the reconstituted family file retrieved from the data base.

REFERENCES

DAULTON D., 1986 — Using the Data Base System, SIR, to link Political Data from Viana do Castelo, Minho, Portugal 1828-1895. Paper given at the Inaugural Conference of the Association for History and Computing. Westfield College, University of London 21 — 23 March 1986 (unpublished).