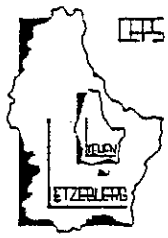
PANEL SOCIO-ECONOMIQUE

"LIEWEN ZU LETZEBUERG"

Document PSELL N° 17a

ORGANIZATION OF THE DATABASE FOR THE LUXEMBOURGER HOUSEHOLD PANEL

[Input, Storage und Analysis]



G. Schmaus

Document produit par le

CENTRE D'ETUDES DE POPULATIONS, DE PAUVRETE
ET DE POLITIQUES SOCIO-ECONOMIQUES

C.E.P.S./INSTEAD

B.P. 65 L-7201 Walferdange
Tél. (352) 33 32 33 - 1

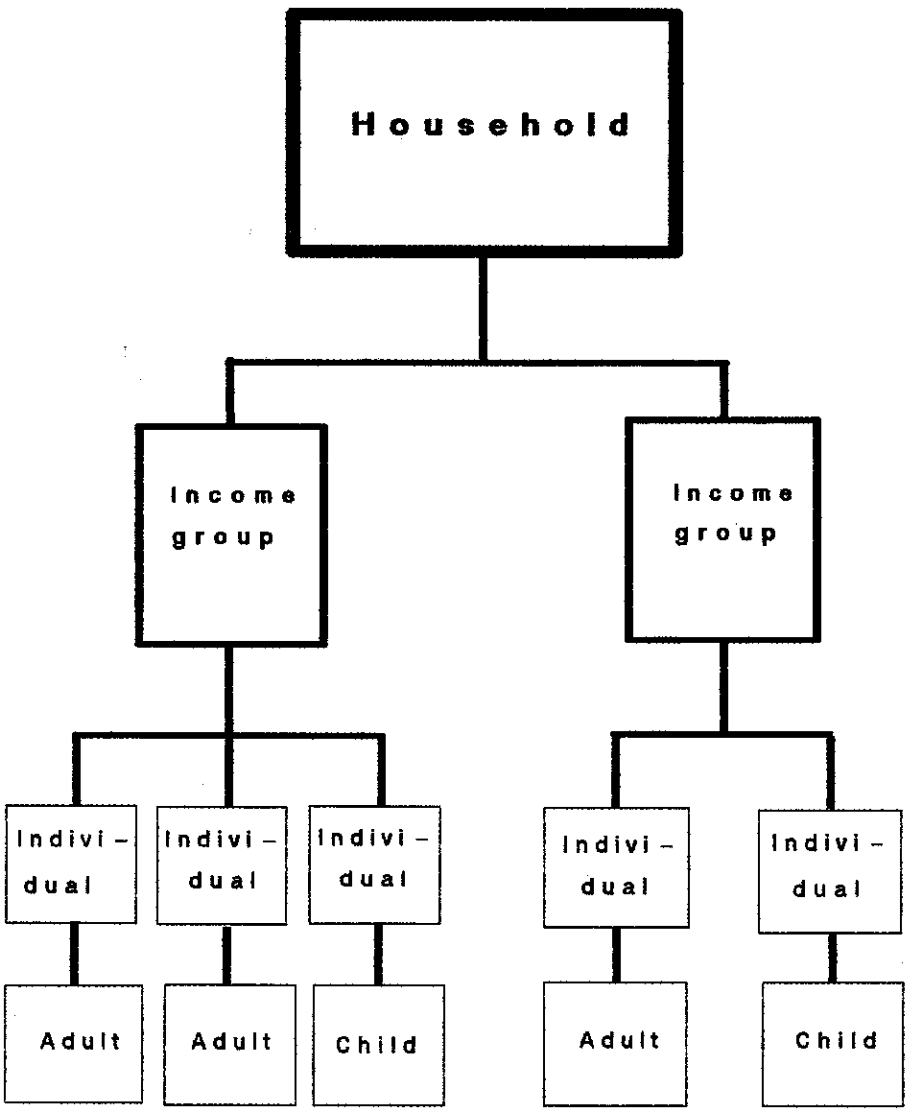
Président: Gaston Schaber

---

1 9 9 0

Table of Contents

Figure 1 : THE PANEL DATA HIERARCHY



INTRODUCTION

The socio-economic panel "Living in Luxembourg" consists of a household sample with a number of measures of demographics, housing, health, education and income. The questionnaire takes into account the fact that several people may live together in one household, and to some extent share their income as well. It includes questions at the household level, on income groups and on individual members of the household. In those sections of the questionnaire dealing with individual members, separate questions are asked about children and adults; some of the questions are asked of every member of the household.

In computer science terminology this sort of data structure is called a hierarchical file. The panel's data include three hierarchical levels: the household, income group and individual. The individual level is further broken down into children and adults (Figure 1). Owing to these multiple hierarchies, the panel represents a complex file. Furthermore, it is very large in terms of the number of variables, since in some areas quite detailed questions are asked at each level (household/income group/individual). In the case of income, information is gathered for each month and broken down according to up to 24 different types of income.

A socio-economic panel differs from a cross-sectional survey in that the basic questions are asked several times of identical study units. In the Luxembourg panel, there is one wave of data each year. By the end of 1988 four waves had been completed. In contrast to one-time statistical surveys, in panels certain special cases need to be taken into consideration. First, individuals may decline to

participate in subsequent waves of interviewing or leave Luxembourg. Second, individuals may die or births may occur. Third, people may leave a household and form a new one (so called split-offs). The survey must provide for such cases, and measures must be taken to assure the feasibility of a comparative temporal analysis that includes, for example, persons who have changed households. Further difficulties arise when one attempts to study the life course of an individual for whom a children's questionnaire was filled out initially, but an adult's later on.

Each wave of the panel can be analyzed in the same way as a cross-sectional sample. However, the real value of panel data becomes evident only when waves are linked with each other, then analyzed. To do this, one must carry out a so-called "match/merge/link". Owing to the above-mentioned special cases (split-offs, deaths and births) this is a complicated process and requires more steps than a simple merging of computer files.

This paper will show how - after the interviews are conducted, the manual checking phase is completed, and coding is done - the panel data are organized in the mainframe of the Centre Informatique de l'Etat (CIE). It will explain how the data are cleaned, merged and made available for statistical analysis.

Particularly as regards the analysis of panel data, we can do no more than make recommendations here. Practical analysis will show to what extent modifications and improvements will be necessary in the organization and retrieval of data.

1 Data management

The panel data must be organized in such a way as to facilitate their entry, storage and analysis. The organizational scheme also has to fulfill the following basic conditions:

- the hierarchies of the panel data must be depictable;
- the waves must be capable of being merged
- special cases (split-offs, births, deaths) must be readily integratable.

Further, the data should be available on-line. So that ad-hoc error checks and corrections can be made, one should be able to access individual records interactively. The analysis of data should not be limited by the type of storage. User interfaces with statistical packages such as SPSSX, LIMDEP, etc. should be possible. Data retrieval should be efficient. It is also desirable to have a user-friendly system with a powerful retrieval language.

Many of these requirements can be met by relational data bases. The panel data have therefore been stored in IBM's SQL data base (SQL/DS). Analysis is currently being performed primarily through the SPSSX statistical package. Storing the data in SPSSX alone is not adequate if several panel waves are to be merged and aggregated.

1.1 Organization of the data in the database

Relational data bases organize their data files internally in what are referred to as tables. Each table is made up of lines and columns. Each column has a name and represents a variable in sequential files. There is a line in each table for each unit. If required, each line and column (or several columns simultaneously) can be retrieved separately. In order to do so, one must use the retrieval language "Structured Query Language" (SQL).

The data from each panel wave are stored in five different tables. Each hierarchy or the relevant questionnaire section is assigned to a certain table. The SQL tables are designated in this system by the letters M, G, I, C and D. The year is added following these letters to distinguish individual waves from one another.

Questionnaire sections as they appear in the SQL tables:

Household questions	--> Table M
Family table Arrivals, departures	--> Table I
Income group questions	--> Table G
Children's question	--> Table C
Adults' questions	--> Table D

Example:

Household questions for 1985 --> Table M85
Household questions for 1986 --> Table M86

SQL Table I (Family table) is of central importance in each wave. The variables in this table enable one to match individuals with income groups. Furthermore, this table shows whether for each individual a children's or adults' questionnaire has been filled out. The Family table also includes diagnostic variables, which provide information about the history of individuals who have been eliminated or added between one wave and the next (arrivals, departures). In such cases reasons are always given for their elimination (e.g., death, departure from the household (splitting off)) or arrival (e.g. moving into the household, birth).

All five tables for each wave contain at least one identification variable - the household number. In addition, with the exception of Table M, at least one other identification variable is included. In the case of Tables I, C and D(Family table/children/adults) this is the individual's identification number; for Table G it is the income group number. These identifying variables are required if one is to retrieve information from more than one table in one wave. The SQL data base refers in this case to "joining" tables. The tables' identifiers are known as "join conditions".

Household numbers and the identification numbers of individuals ensure that the proper individuals and households are matched. Various controls must be employed during the data-entry and data-cleaning phases to make sure that households and persons are identical with their numbers.

Indices have been set up to facilitate data retrieval from SQL. In general, these are the identifiers in the so-called "join conditions".

1.2 Variable names

Names are assigned to the panel variables, and thus to all of the variables in the corresponding SQL table, according to a certain system. Variable names indicate the hierarchy and wave to which each variable belongs. In addition, each variable's name shows whether it is a link variable or a regular variable as well as whether or not it is a so-called vector variable.

The first character of the variable name gives the level of that variable in the hierarchy.

M --> household (menage)
G --> group
I --> individual

The next two characters show the year of the wave.

85 --> 1985
86 --> 1986
87 --> 1987

Subsequent characters are used to distinguish variables within their hierarchy. As a rule numbers are used. A small number of variables, used for linking and identification, have an "L" (L = link) before the number.

Examples of variable names:

M87057 -->	number of cars in household in 1987
M86L01 -->	household's number in 1986 (link variable)
I85L09 -->	individual's sex, 1986 (link variable)
G87227 -->	amount of loan that must be repaid by the group in 1987.

In certain cases another letter is added at the end of the variable name to show that the definition of that variable has been changed.

Some kinds of income information collected for each month of the year are given as a table in the questionnaire. For example, a table shows the amount of income received from various sources - 24 in all. Information of this kind is stored in vector form. These vectors are marked with two letters. A third letter is inserted before the vector name, identifying the year of reference (X = preceding year, Y = current year). A number is added following the vector name to show the reference month (1 = January, 2 = February, 12 = December).

Examples of vector variable names:

I87XAB5 --> a) vector name: AB
b) information from the previous year = 1986: X
c) information for May: 5

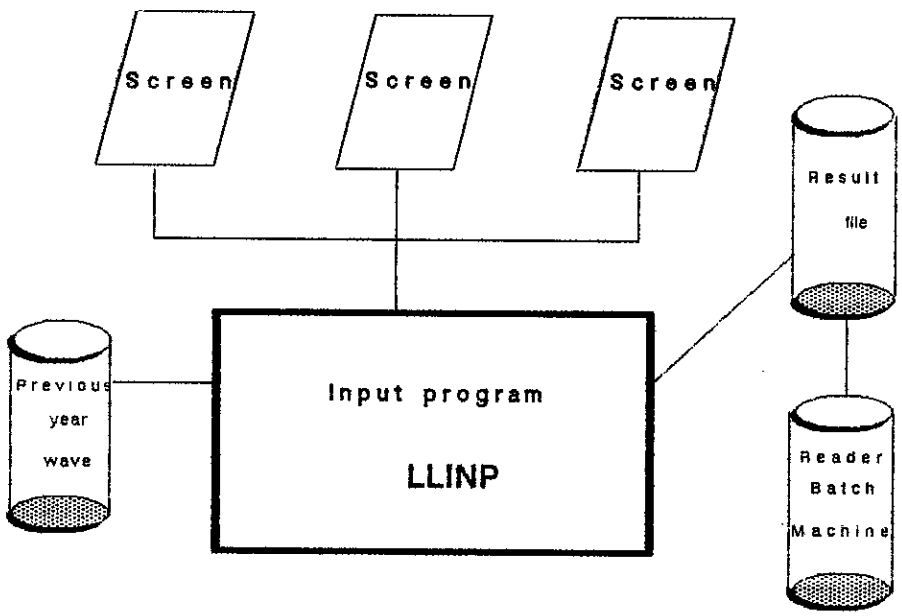
I87YAB1 --> a) vector name: AB
b) information from current year = 1987: Y
c) information for January: 1

The variable and vector names for households and income groups show immediately in which SQL tables the individual variables are stored. This is not always clear at once in the case of individuals' variables. However, they can be matched to the SQL tables since, with few exceptions, the individual panel questionnaires are assigned to strictly determined tables.

2 Panel data input program

The questionnaires from each wave of the panel are stored by means of an interactive program (Figure 2). The individual parts of the questionnaire appear on the terminal in a series of successive screens. The format is for the most part based on that of the questionnaire. The program dynamically prompts to prevent the person entering the data from forgetting to enter parts of the questionnaire; indications are given of missing or erroneous questionnaire sections.

Figure 2 : Input program



The program greatly facilitates the entry of data in the individual screens. Sections of the questionnaire are depicted graphically. The cursor moves automatically to the point where the next entry is to be made. Extensive error checks are made within each screen. For example, a check determines whether individual fields may remain blank, whether they must contain numbers and whether a number is within a given range. Moreover, the consistency of the relationship between variables on the screen is tested. Here an illustration: If the household in question owns its dwelling, then the field for rent must remain blank, while there must be an entry in the field for mortgage payments.

Only formal errors can be detected by these checks. The mistake may lie in the questionnaire itself, or it may be the result of a typing error. The program can find either type of problem.

Entry errors are pointed out immediately; the cursor highlights the erroneous entry. The program insists on a correct entry before one can proceed with entering the questionnaire.

The sections of the questionnaire are shown in the same order as in the interview as conducted in the household. First, however, in addition to the variables coded from the questionnaire, the precoded variables are entered. The order of the screens, then, is as follows:

(1) Precoded variables ("fiche-precodage")
    a) diagnostic variables
    b) other variables, such as household type

(2) General questions on the interview ("fiche-enquêteur")

(3) Family table

(4) Arrivals and departures
    a) Arrivals
    b) Departures

(5) Household questions

(6) Income group questions

(7) Individual questions
    a) Children
    b) Adults

The first data entry is that of the matrix with the diagnostic variables. This matrix lists all individuals in the wave of the previous year with their registration numbers and a note on whether persons have been added or eliminated, and, if so, which ones. The program checks this information against data from the previous wave. If it is identical, this ensures an error-free link between households and persons in the current and previous year. If it does not prove to be identical, the program will refuse further entries until identity is established.

The diagnostic variables determine which persons are to be included in the Family table. All individuals present in the previous and current wave and all those who have joined the household must be entered. The diagnostic variables also determine for which persons arrival and departure questionnaires must be entered.

After the other precoded variables are entered, the Family table variables are stored for each household member. Among other things, this means entering the income group of each individual and noting whether a children's or adults' questionnaire has been completed. These variables determine how often screens subsequently come up and are required to be completed for income groups, children and adults.

After any questionnaires for arrivals or departures have been entered, income group questions appear on the screen. When there is more than one income group in a household, further group questionnaires are entered for as many as are present. Each group is asked how many members it contains. This information is checked against the Family table and any errors are pointed out.

Next, the children's and adults' questionnaires are displayed on the screen in the order occurring in the Family table. Some of the questions in the adults' questionnaire are further broken down according to whether the adult in question is employed, seeking work, part-time employed or not employed.

Another important task of the program is to ensure that information is consistent with regard to the household hierarchy and the household's subdivision into groups, individuals, children and adults. This is done through the dynamic screens. The organizational structure of the dynamic screens guarantees a maximum of error-free entries, as illustrated by the following two examples:

a) If the diagnostic variables have been entered incorrectly, e.g., if an individual has mistakenly been identified as an 'old' person (person existing in the new and the previous wave) rather than as a departure, then the program requests an entry for an individual in the Family table. Information on this individual cannot be found in the

questionnaire, so the operator recognizes and corrects the error.

b) If there is an erroneous entry in the Family table, e.g., an individual has mistakenly been identified as a child, then there is a prompt for a children's questionnaire. Since the operator has only an adults' questionnaire, however, the error is apparent.

Some of the income-group and individual variables contain information for twelve months out of a year. These twelve consecutive months are combined into one variable using a compression algorithm, an efficient procedure for the following reasons:

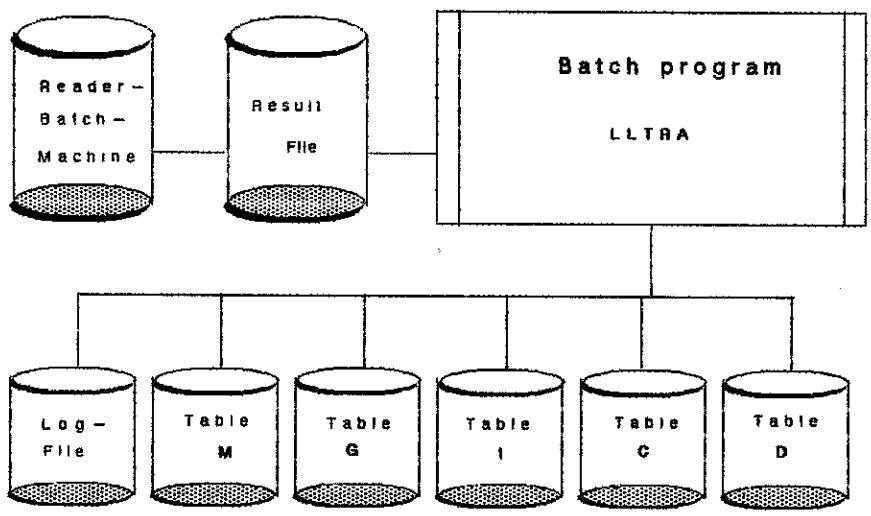
a) compression saves disk space;

b) the number of variables stored in the SQL is considerably reduced (savings per wave: approximately 500 variables).

The panel entry program produces a separate file for each household. This file consists of several parts separated by identification markers. It contains a household record, one or several records for the income groups as well as one or several for the Family table, children and adults.

Upon completion of the entry process, this file is sent from each input machine to the batch machine. A program is run in the batch machine to collect the individual questionnaires that have been entered (Figure 3). Using its virtual reader, the batch machine reads in a questionnaire and splits it into five sections according to the five SQL tables. These sections are added to the five intermediate files used to load the tables in SQL. As a check the program produces a log file in the batch machine. The log file records which questionnaires have been entered and determines whether any of them contain errors.

Figure 3 : Batch program LLTRA



3 Intermediate data check

During the data entry phase and before the data have finally been stored in the SQL data base, further checks are run:

a) The five intermediate files are read into SPSSX. There the data can be listed case by case and compared manually with the original data in the questionnaires. Simple statistical analyses are carried out in SPSSX as a further check. The minimum/maximum and averages are found for each variable and the minimum and maximum values are compared with the acceptable value ranges. In this way systematic errors in the entry process can easily be found and corrected.

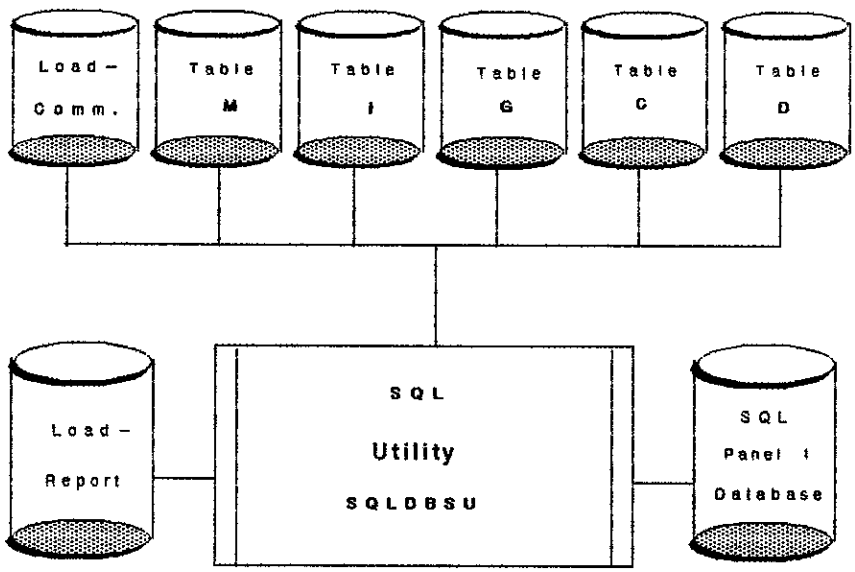
b) A check is also made to see whether duplicate sets (households, income groups, individuals) are contained in the intermediate files. If so, the appropriate sets are deleted before loading is done in SQL.

c) Hierarchy consistency is checked by means of an independent Fortran program. Here consistency means that there are no contradictions between the individual sets in the hierarchy. Accordingly, the information contained in the Family table on income group membership and on whether a children's or adults' questionnaire should have been filled out should match the information from the income groups and from the children's and adults' questionnaires. Furthermore, individuals may not appear more than once. Each person can be assigned to only one group, and both a children's and an adults' questionnaire may not be filled out for any one person.

4 Data storage in SQL

Following the intermediate data checks, the new panel wave data are stored in the appropriate SQL tables (Figure 3A). Each table has a separate loading file. After the data have been loaded, for purposes of efficiency so-called indices are defined for SQL. Depending on the table, various indices are set:

Figure 3A : Loading Panel Data



Indices for SQL tables:

Table	Index
M	Household number
I	Identification number Household and identification number
G	Household and income group number
C	Identification number Household and identification number
D	Identification number Household and identification number

The indices are used in SQL primarily to make queries more efficient. In addition, they can be used to check whether duplicate sets (with the same identification number) have been loaded into the tables as a result of undetected errors.

After the indices have been set, further tests are carried out with the SQL tables. Some of the tests are cross-sectional, the rest longitudinal.

4.1 Cross-sectional tests

The cross-sectional tests check for an unambiguous hierarchy in the household. This is guaranteed if all of the units in one table can be linked to those in the corresponding table. The following tests are used:

Cross-sectional tests:

a) All income groups in Table G must be linkable to a household in Table M.

b) All individuals in Table I must be linkable to a group in Table G.

c) All individuals in Table I must be linkable to a household in Table M.

d) All children in Table C must be listed in Table I.

e) All adults in Table D must be listed in Table I.

f) No child in Table C may be included in Table D.

g) No adult in Table D may be included in Table C.

4.2 Longitudinal tests

The longitudinal comparison is aimed at determining whether each household, individual and income group in a new wave can be linked to the corresponding units in the previous wave. The method used here is based on the premise that if each new wave can be clearly linked to the previous one, this also ensures a link with all prior waves.

Longitudinal tests:

a) All households from the new wave present in Table M must be entered in the corresponding Table M of the previous wave.

b) All "old" individuals (persons existing in the new and previous wave), identified by the diagnostic variables in Table I, must be entered in Table I of the previous wave.

c) All "old" children in Table C, identified by the diagnostic variables in Table I, must be entered in Table I of the previous wave.

d) All "old" adults in Table D, identified by the diagnostic variables in Table I, must be entered in Table I of the previous wave.

e) Individuals with a children's questionnaire in the previous wave may have an adults' questionnaire in the current wave, but it is not possible to change from an adults' to a children's questionnaire between the previous and the current wave.

More complex cases are also covered by the longitudinal tests. Things become complex because in each wave new individuals are added and others leave. This also holds for households with split-offs. Thus queries and tests are substantially more extensive than in normal cases.

Particular attention should also be paid to the quality of the merge with regard to content. The tests described above can only ensure the formal quality of the merge. To deal with the problems of merge quality with regard to content, two programs are used outside SQL.

a) One program checks to see whether each household that was merged formally on the basis of the household number is also the same in terms of content. The household variables are not used for this; they may change from wave to wave. Instead, one indication of an identical household is if at least one person has belonged to the merged household in both waves. This person is not further specified for the test. Another possible test, whether the household head or wife is present in both waves, is not always helpful. From wave to wave individuals, including the household head or wife, may leave for a variety of reasons, while the household itself continues to exist.

If discrepancies show up in merged households, one can also examine the household's original questionnaires to determine whether or not the households are indeed identical.

b) In the other program a check is made to see whether each person with a given identification number is indeed the same person. Primarily sex and birth date are used for this comparison. If these two variables agree, then an erroneous merge can generally be ruled out. If they do not, the data on this person are printed out and, with the help of the questionnaire, contradictions are straightened out and corrected.

5 Data editing and analysis

After the data are stored and checked in the data base, the data editing phase begins, after which analyses can be carried out. Here data editing means that so-called "missing values" are replaced (at least in part) by estimates, errors are corrected and, to facilitate analysis, additional "new" variables are generated from the existing ones.

5.1 Problems of missing values

Missing values occur when respondents cannot answer questions or refuse to do so. Here one should distinguish between qualitative and quantitative variables. Particularly in the case of quantitative variables, missing values represent a serious problem for the analyst. In the panel, an individual's income is broken down into 24 different types of income. It is impossible to determine the total income of an individual or household, aggregated over all members, in SPSSX if even one partial component (i.e., just one variable among many) contains a missing value. It is not adequate to substitute a zero for these missing values if one knows from other variables that the value should be larger than that.

In such a case one can only make a reasonable estimate of the missing value. To do this, various procedures may be used. The regression approach estimates regression coefficients using the data from units without missing values. These coefficients are used to replace missing values. Depending on how differentiated the approach is, the estimates can be either very rough or very precise. The table approach is the second alternative. This approach can, first, use as estimates more or less differentiated averages

based on the data from units without missing values. Second, external information in table form can serve as estimated values.

Whichever method is used, problems crop up when one needs to estimate a variable's value in one wave, while information from the household is available for another wave. In the case of income variables, this can lead to implausible income jumps when longitudinal analysis is performed. So that such cases can be readily identified, estimated variables are marked by a flag. In general, any estimate in a wave should be corrected if the variables in later waves indicate that it is in error.

Missing values do not pose as much of a problem for analysis in the case of qualitative as in that of quantitative characteristics. In the latter case, these should be replaced wherever possible. Here panels often have better means of correcting errors than do cross-sectional surveys. If information is missing in one wave, it can be gleaned from another wave.

5.2 Problems with error-prone variables

Some of the variables are gathered in every wave. For them, repetition provides an opportunity to identify errors that may have occurred earlier on. Here the difficulty lies in finding out whether a change has actually taken place between one wave and the next, or whether an apparent change is really an error that must be corrected. With a small number of variables (e.g., sex, marital status and birth date, household composition) it is possible from wave to wave to correct earlier errors and thus to render the data more coherent.

5.3 Problems with new variables

For a variety of reasons, "new" (additional) variables are formed out of existing ones to be used in analysis:

a) Recoding of variables (e.g., age groups)

b) Formation of classification variables with the help of other variables

c) Addition and subtraction of variables to form income concepts

d) Aggregation of variables on a higher level of the hierarchy.

Extensive analyses can produce large numbers of added variables. The question arises as to whether all of these newly formed variables -and if not all, then

which ones- should be stored in the data base. A fresh calculation of variables can be seen as an alternative to storage. Storing new variables requires additional storage space, while calculation takes more computer time. There can be no one single answer to the question of storage versus renewed calculation. A decision needs to be made in each individual case as to which variables will be stored and which will be recalculated.

The new release (3.0) of SPSSX now has two commands that can be used to handle on an organizational level the renewed computation of variables. These are the option of dynamically inserting SPSSX commands from an external file into an SPSSX program (Includes) and the possibility of working with Macros. With these instruments, external Includes/Macros can be made available for groups of variables during analysis. These Includes/Macros are available on a minidisk for every user. Since they exist in only one form for all users, they can be handled centrally and corrected whenever necessary.

5.4 Update possibilities

The panel data are, for a number of reasons, subject to continual change and expansion. This is due primarily to individual corrections of existing variables and the generating of "new" variables. Individual corrections can be carried out ad hoc and interactively at the terminal screen (ISQL). This procedure is not advisable if the changes to be made need to be documented. Individual corrections should be entered through an editor into a command file for the SQLDBSU Utility, then carried out. This command file should be kept permanently; it documents the changes made and makes it possible when reloading a table to replicate those changes. New variables can

be brought into the data base by expanding existing tables or establishing new ones. Which of these two methods is preferable can only be determined on a pragmatic basis. If there are relatively few new variables, one might expand the existing tables. If the number of new variables is quite large, then one should set up a new table. An advantage to the latter procedure is that there is a clear separation of "new" variables from those taken from the questionnaires. Should some of the new variables need to be corrected, the entire table can first be removed and all new variables can be reloaded. Updating the new variables would be considerably more cumbersome if they were stored in the table with the old ones.

6 Analyses

After data entry, loading and correction comes the analysis phase. With hierarchical data sets and panel data, there are a number of different analysis options:

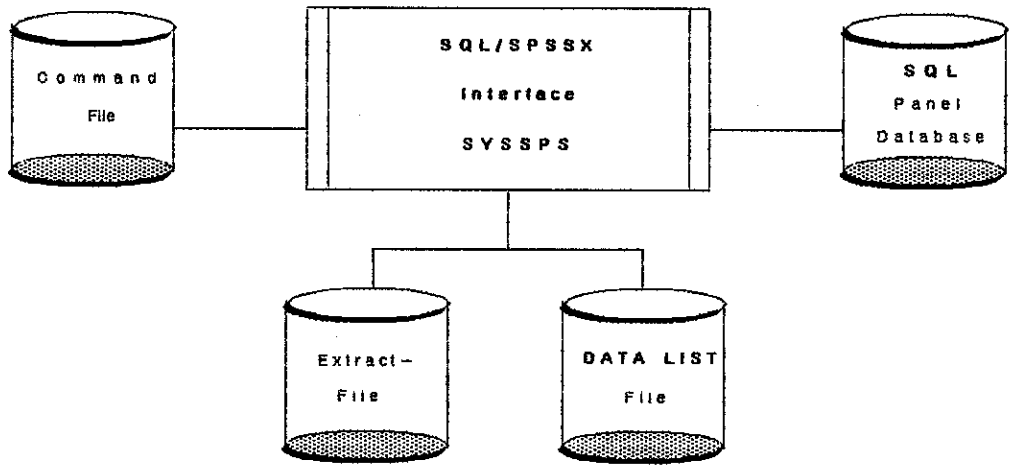
Analyses can be carried out separately for households, income groups and individuals. If information from several hierarchical levels is required, the data from lower levels must be aggregated to a higher level or the data from a higher level must be transformed into a lower one. Furthermore, analyses can be performed with a cross-sectional or longitudinal orientation.

Analyses can be carried out using various statistical packages. Currently available are the software options SPSSX, LIMDEP, MDSX and SPADNA. The statistical packages differ from one another, among other things, in the way they handle data. Three of these packages (LIMDEP, MDSX, SPADNA) do not offer actual database functions such as merging or aggregating of data sets. SPSSX, however, offers some very finely-tuned ways of dealing with complex structures. In this context the functions "match files" and "aggregate files" are of particular importance. As a rule, all of these packages require as system input formated rectangular files ("flat files").

These various methods of analysis with regard to data structure and statistical packages require a flexible link between SQL and the application package. An interface is available for this purpose (Figure 4). The interface makes available the data from the SQL data base in a formated rectangular extract file ("flat file"). The extract file can be read into SPSSX. The necessary SPSSX commands for reading in the data are produced by the interface and are

available in a separate file. With this extract file, statistical analyses can be carried out immediately in SPSSX, or an SPSSX system file can be produced. In addition, the extract file can serve as data input for the other analysis packages.

Figure 4 : SQL/SPSSX Interface



The interface must be told which kind of extract file is desired and which variables are to be selected. This is done through a command file with the necessary SQL commands. The command file must contain the following information:

a) Select clause: which variables (columns) are to be extracted

b) From clause: which tables are to be used

c) Where clause: how these tables are to be linked

d) Which records (lines) are to be retrieved (optional)

e) Order by clause: whether the file is to be sorted (optional)

With the help of these commands, extract files can be set up for purposes of analysis. In the simplest case, the table or parts of it are unloaded from the panel database. The following command, for example, copies all variables from the Family table 1986 into the extract file:

```
SELECT * FROM I86
```

More complex extract files, files containing information from two and more SQL tables, are needed for extensive analyses and in particular for longitudinal analyses.

The contents of the interface extract file may be of very different kinds:

1) Data from one table (simple cross-section)

2) Data from more than one table in the same wave (hierarchical cross-section)

3) Data from more than one table of different waves (simple longitudinal sample)

4) Data from several tables of different waves (hierarchical longitudinal sample)

The structures of the individual extract files differ only in their variables.

Examples of structures of extract files:

1) File with individuals' variables from wave 85:

   I85001, I85002

2) File with variables for individuals from wave 85 for which household variables have been expanded:

   I85001, I85002, M85010, M85011

3) File with variables for individuals from waves 85 and 86:

   I85001, I85002, I86001, I86002

4) File with variables for individuals from waves 85 and 86 for which household variables have been expanded:

   I85001, I85002, I86001, I86002, M85010, M85011, M86010, M86011

To produce such complex extract files, the SQL tables must be joined or the SPSSX files must be matched. Joining tables or files offers several possibilities:

a) Parallel match (e.g., linking new and old variables from one wave)

b) Non-parallel match (e.g., information on individuals from two or more waves)

c) Expanding variables (e.g., matching household variables to persons)

d) Aggregating sets (e.g., adding income at the group level to arrive at the household level)

Here the question arises as to which file operations should be carried out in SQL and which in SPSSX. For connecting tables there are two possibilities: With the first method, a separate extract file is produced from each SQL table. If information is to be used from two or more tables, the corresponding files must be transformed in SPSSX. With the second method, only one intermediate file is produced in each case and the appropriate tables are joined in SQL. Empirical tests have shown that both methods have advantages and disadvantages. SPSSX offers numerous different transformation procedures. Within SPSSX it is possible to create very complexe files without any serious restriction on the number of variables. Particularly in the case of longitudinal analyses with hierarchical data, the necessary transformations in SPSSX are quite complicated, requiring a number of auxiliary files and multi-stage procedures.

The second method (SQL) makes it possible to carry out this kind of transformation with greater ease. In SQL one can produce an extract file with information from several tables under relatively few, albeit complex, From and Where conditions. The appropriate SQL commands transform the tables in such a way that no additional auxiliary files are needed. In SQL the extract file is set up in a one-step procedure. However, it should be noted that with this method the number of selected variables must not be too large. Depending on their complexity, only approximately 300 variables can be unloaded into the extract file. Furthermore, it is very difficult to unload and to join data from many tables at the same time. For this reason, it may be necessary to use the first method (transformation in SPSSX) for extract files with a very large number of variables.

Problems may arise when data sets are to be aggregated with the second method. SQL contains only four aggregation functions, while there are about 19 different functions in SPSSX. For aggregation operations it is therefore better to use a mixed procedure:

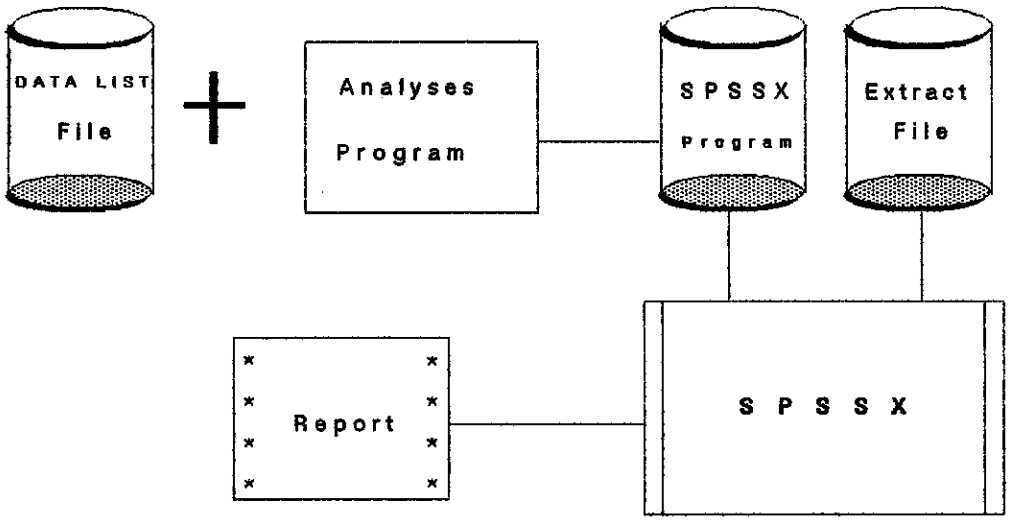
If, for example, information from the household and individual level is needed at the income group level, then an individual file should be produced in SQL in which the household and income group variables are expanded. This individual file is aggregated in SPSSX to the group level.

These variables to be extracted must be listed in the command file for the interface. If more than one table is involved, SQL must be informed as to how these tables are to be linked. Precise knowledge of how the tables are organized and which identifiers must be used is required in order to be able to link the tables in SQL. Inexperienced or new users do not have such knowledge; for them there is a small program library with the appropriate SQL commands. The individual submodules contain the From and Where clauses for standard analyses of cross-sectional or longitudinal nature. A user interested in using the SQL panel data can put these commands into the command file for the interface. As a rule, the only other thing necessary is to specify the Select variables.

SQL offers the possibility of using so-called views. Views are imaginary tables resulting from combining existing ones. Thus views could be an alternative to using the program library. The number of variables in the views is limited by SQL to 140. But since most analyses require more variables than that at one time, views cannot be used for specific panel analyses.

The interface produces extract files that can be directly read into SPSSX. There statistical analyses can be carried out (Figure 5). In some cases aggregation procedures should be performed first.

Figure 5 : SPSSX Analyses program



For strategic reasons, it is best to produce only small intermediate files which, in general, are kept only temporarily. In the interest of efficiency (computing time and storage space), it is not always a good idea to unload all variables in one or several tables at a time through the interface. It is wise to extract only those variables that will be required later for analysis. Depending on the type of analysis, the interface files can be kept temporarily

or permanently. Temporary files are adequate for ad hoc tabulations. For the duration of a large-scale study it may be sensible to produce files with subset of variables that are stored as SPSSX system files. This is particularly advantageous if matching or aggregation were necessary for their production. These files can be deleted after the study has been completed. In contrast to normal cross-sectional studies, the data-cleaning phase continues throughout the panel surveys. Thus the data are continually updated in SQL. Analyses should be conducted with the most recently updated data. This is possible only if the data or portions of them are not kept parallel to one another in SQL and as SPSSX files.

Each additional wave can offer the opportunity to repeat analyses conducted on the previous waves. For this purpose the analysis program must be kept general. Here "general" means that the year of the wave must not be explicitly included in variable names. Each variable name should contain the wave year as a parameter. Depending on the wave (cross-section) or study period (longitudinal), actual values can be assigned to this parameter. In SPSSX one can do this by using Macros. A general Macro can be written for each analysis. Only the year of the wave and/or the study period serves as an input parameter.

Computer programs :

1) **Input program:**

   a) Setting up control files from the previous wave: LLPREP (Fortran)
   b) Data entry program: LLINP(REXX, ISPF)
   c) Batch program: LLTRA (REXX, Fortran)

2) **Program for intermediate data checking:**

   a) Checking household files (M): LLPRA (Fortran)
   b) Checking individual files (I): LLPRB (Fortran)
   c) Checking children's and adults' files (C,D): LLPRC
   d) Checking income group files (G): LLPRD (Fortran)
   e) Checking the hierarchy: LLPRE (Fortran)

3) **Loading data into SQL (SQLDBSU)**

Households:	CREATM	LOADM
Individuals:	CREATI	LOADI
Income groups:	CREATG	LOADG
Children:	CREATC	LOADC
Adults:	CREATD	LOADD

4) **Checking data in SQL:**

   a) Cross-sectional data tests: ISQL commands
   b) Longitudinal tests: ISQL commands
   c) Test for identical households: CONS2 (Fortran)
   d) Test for identical individuals: CONTEST1 (SPSSX)

5) **SQL interface:** SYSSPS (SQLDBSU/REXX)

Explanations of abbreviations:

REXX: "Restructured Executer Language" is an interpretative language from IBM at the command level

ISPF: "Interactive System Productivity Facility" is an IBM software program that makes possible the flexible programming of screens

ISQL: "Interactive Structured Query Language" is the dialogue-oriented query language of the SQL data base

SQLDBSU: "SQL Database Service Utility" is the batch-oriented query and manipulation option of the SQL data base