A Relational Database Structure

for the

March Current Population Survey

William A. Gates

The title of this paper may seem to suggest that the CPS is not in a relational form. The March CPS is in relational form, albeit a primitive one. The purpose of this paper is to illustrate by using the March 1986 CPS that a 'more' relational form has many advantages, not the least of which is reducing the sheer size of the file while providing greater clarity of content. Some may argue that new optical disk technology will soon erase the concern with size. Clarity of content remains a requirement. For example knowing in advance the actual number of persons in a sub-sample question is enough to prevent a fruitless investigation. Relational database systems together with relational query languages make discovering such facts an easy and practical matter.

This paper relies on the intuitive common-sense aspects relational database theory to illustrate one model among many for structuring the data differently. Thorough and rigorous treatment of relational database theory can be found in many sources and the reader is directed to the references. Sufficient details are provided with respect to relational concepts and to the CPS such that the reader should be able to easily grasp the results.

The resulting structure (a set of tables in relational terminology) is biased toward the objective of reducing the size of the March CPS and thus making the data 'fit' on today's PCs. The end result is still evolving but as a practical matter the March 1986 CPS can be reduced from 120 megabytes to roughly 32 megabytes and less. The person data alone can be reduced from 64 megabytes to less than 20 megabytes.

The Approach

In the initial plan the entire CPS was to be treated using a Digital Equipment Corporation VAX. After working with the household record set and in the interest of time and money this was scaled back to working with one rotation group and focusing on the person data. The results are easily extrapolated to represent the effects for the full CPS. In addition, the beneficial effects (size, clarity) are maximally expressed by reorganizing the person

data. By this approach the majority of the processing was shifted to an IBM PC/XT with a 30 megabyte hard disk. The same relational database software available on the VAX was also available on the XT. Thus the approach also tests the concept of using relational database facilities for manipulating and making data available on PC's. There are at least 15 million PC's and compatibles all processing data in the same fashion so in effect we have a standard if the data 'fits'.

Basic Steps

The characteristics pertinent to measuring our starting point are provided in Table 1. They will be repeated in Table 9-12 to compare them with the new tables and their size. Please recall that the total size of the March 1986 CPS is roughly 120 megabytes. Each MIS (month in sample) group is approximately 1/8 of the 120 megabytes.

MIS	Record Type	Fields	Size	Count	Bytes
1	Household	71	408	9,638	3,932,304
1	Family	81	408	7,934	3,237,072
1	Person	240	408	19,566	7,982,928
				Total	15,152,304

Table 1. Size characteristics of the Household, Family and Person data for MIS (month in sample) group 1 of the March 1986 CPS.

The very first observation to make about the CPS record types is that they contain a fair amount of what is called 'filler'. Traditionally filler has been used on serial media, such as magnetic tape, to pad records so that subsequently they would be easier to process by statistical packages, etc. This is much less of an issue today than when the CPS was first begun. Many statistical packages today have interfaces for relational database systems (SAS, SPSSX). Information from Table 2 contrasted with Table 1 shows the rather dramatic effect of eliminating filler from the record types (a 33% reduction in size).

Record Type	Fields	Filler Removed Bytes	Count	Bytes
Table 1 Total	-	-	-	15,152,304
Household	7	285	9,638	(2,746,830)
Family	3	199	7,934	(1,578,866)
Person	8	30	19,566	(586,980)
			SubTotal	(4,912,676)
			Net CPS Size	10,239,628

Table 2. Key characteristics of the Household, Family and Person data for MIS 1 of the March 1986 CPS with 'filler' removed.

The presence of filler in the record types could be construed to say that the CPS is not in 1NF (first normal form), since the filler could be labeled as repeating groups or null information, but the filler is of fixed length and in fixed positions. 1NF essentially requires only that the record be rectangular. The argument provides an opportunity to present the definition.

The second major observation to make about the CPS record types is that they contain fields that are used in only some of the rotation groups depending on the MIS code. The impact of eliminating these fields is presented in Table 3. Such a step implies that the data from rotation group to rotation group has records of different sizes with a majority of the fields the same but not all. The unique information could also be presented in a separate table. The focus from here until the summary is with the person record only.

Record Type	Unused Fields	Removed Bytes	Count	Bytes
Person Total	-	-	-	7,982,928
Person Filler	-	-	-	(586,980)
Person Unused	14	29	19566	(567,414)
		MIS 1 - Net CPS Size		6,828,534

Table 3. Person data for MIS 1 of the March 1986 CPS with 'filler' and unused fields removed.

The third major observation to make about the CPS record types is the presence of 'redundant' information. Removing redundant information to auxilary tables is a step toward putting the table in 3NF. Information is redundant in that it can easily be reconstructed from one or more fields. Not all instances of redundancy have been removed. The effect of removing these redundant fields is of smaller consequence for the person record but still contributes to space savings. Table 4 gives the results.

Record Type	Redundant Fields	Removed Bytes	Count	Bytes
Table 3 Net	-	-	-	6,828,534
Person Redun.	8	15	19,566	(293,490)
		MIS 1 - Net CPS Person Size		6,535,044

Table 4. Person data for MIS 1 of the March 1986 CPS with 'filler', unused and redundant fields removed.

If one stopped at this point the full CPS person data has been reduced from roughly 64 megabytes to approximately 52 megabytes.

The process of removing fields provides a good opportunity for illustrating some features of relational query languages. Figures 1 through 3 provide examples of SQL (Structured Query Language). Building tables from other tables is easily accomplished. Rebuilding tables exactly reproducing the original is also simple. VIEWs (rules stored in the database) let you see the data differently from its stored form.

```
create table AGE_RECODE as
     select distinct age, rage1
          from PERSON
               order by age
```

Figure 1. Building a recode table for redundant fields with SQL.

AGE_RECODE is a table that contains only as many rows as there are unique combinations of 'age' and 'rage1.. This AGE_RECODE table will be at a maximum 6800 bytes for all MIS versus roughly 626,000 bytes for leaving it in the base record. The original person table is then copied to a new table omitting the 'rage1' field.

```
create view PERSON_AGE as
     select x.ppseqnum, x.ppos, x.age, y.rage1
          from PERSON_NEW x, AGE_RECODE y
               where x.age = y.age
```

Figure 2. Creating a VIEW (including a join) that can be used to retrieve 'rage1'.

The VIEW is a standard ANSI SQL feature. The rule can be recalled and read at any time. Figure 3 illustrates how to use and then how to write it instead as an SQL select. The select has to be written each time. The second example illustrates using the VIEW (as if it were a table), with an additional 'where qualification'. Storing the rule (creating a VIEW) assures that you get consistent results and that you always perform the same task.

```
/* Example 1 - using the VIEW within a Select */

select * from PERSON_AGE


/*
Example 2 - using the VIEW within a Select for persons > 14
*/

select * from PERSON_AGE x

     where x.rage1 = 6


/*
Example 3 - writing an equivalent SQL select for Example 1
*/

select x.ppseqnum, x.ppos, x.age, a.rage1

          from PERSON_NEW x, AGE_RECODE a

               where x.age = a.rage1
```

Figure 3. Re-creating the original information using the VIEW, selecting adults age 25-29 with the VIEW, and using just SQL.

The fourth major observation to make about the CPS record types is also illustrated by the 'person record'. All of the persons are not the same. Not all information is collected or recorded for each person. Broadly there are at least three person-types as segregated by the data collected about them 'adults', 'children', and 'military personnel'. Indeed there are, however, certain items recorded such as AGE for all persons.

Relational database theory and/or entity relationship analysis suggests that we re-label and segregate data to tables along the three person-types and then use a VIEW (rule) to see the 'person information'. Alternatively the data in common to all persons could be in one table and other tables hold the particular information about just about 'adults', etc. These tables require a common key to join them. PPSEQNUM and PPOS serve this purpose as the 'key' in each. Neither of these organizational alternatives is a requirement but each, minimally, saves space and makes the content clearer. The preferred organization depends on what questions you think will be asked more often, i.e., performance issues.

Figure 4 depicts the tables built from person. Figure 6 in the section 'Children, Adults, ...' illustrates the SQL used to perform the table building.

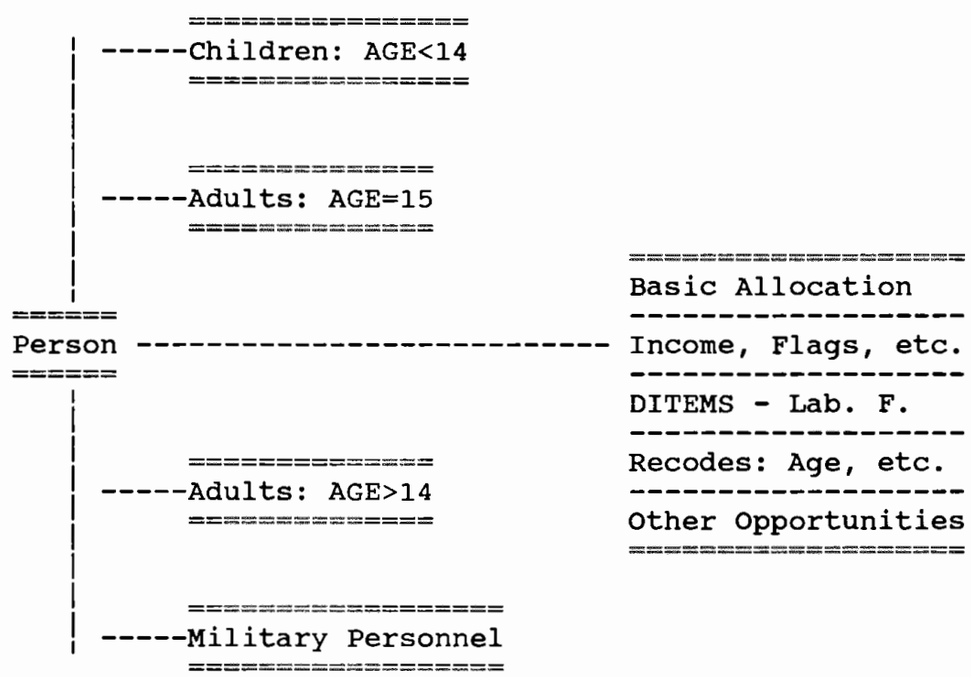


Figure 4. Basic tables built from the PERSON set.

The final observation to make about the CPS record types is really a generalization of the previous case. The survey skip patterns and the use of flags associated with fields make it possible to create more specialized tables. This also saves space and adds to the clarity of content. A corollary relates to fields that are two-valued (or binary), especially when one of the code values is NIU (not in the universe). The 'basic allocation' fields provide a good example of the impact of the two-valued case. These fields record as '0' or '1' whether or not the coded responses to 22 fields were allocated, i.e., supplied by a rule rather than respondent. The value '1' means it was allocated. The BALLOC table contains the 22 allocation flags plus the key PPSEQNUM and PPOS if any one of the allocation flags is set to '1'. The table is derived from the original PERSON table and can be completely reconstructed. Table 5 shows the storage impact of this type of operation.

BALLOC Table	Universe Size	Fields Eliminated Bytes	Net Bytes
Total	19566	-	430,452
No Allocation	19405	426,910	-
At Least One Field Allocated	161	-	3,542
	Savings		426,910

Table 5. Impact of BALLOC table on person records for MIS 1; 22 one byte fields - BAITM19 thru BAESR (repetition of the key is not included in the net calculation).

The very fact that there are only 161 allocation cases for MIS = 1 is immensely useful. It implies that there are perhaps 1200 cases for the entire person record set. By using SQL and writing a simple join the individual records can be easily examined or isolated for further study.

Interpreting the allocation fields as two-valued is not exactly correct. In fact the value '0' really has two meanings. One is 'not in the universe,' where the question was indeed not asked and thus the allocation flag has no real purpose. The second meaning is 'not allocated,' where the value of the question exists as a valid response. The two-valued representation simply reflects a determination of the utility of the information. If we are concerned with statistical treatment of the basic field then we are probably concerned with values that are allocated or we may chose to ignore the issue and rely instead on the 'robustness' of the sample.

The Process

The basic allocation field example can be carried through for several other groups of fields. Realize that the original form or information can always be reconstructed yet the unique aspects of the sub-population is now more clearly visible, espcially the size of the group. This section details the results for the person record including subdividing the person record into several mutually exclusive sub-populations. The tables were depicted earlier in Figure 4. As stated earlier the decisions about which

fields to treat and how to treat them is biased toward reducing the size of CPS while enhancing clarity of content. As such the process has not resulted in an absolutely minimal size nor in a structure that all would consider ideal.

Redundancy: Age, Occupation, Industry, Income

Although there are more candidates than age, occupation, industry and income only 8 recode-tables were built. The impact of removing these fields was reported earlier in Table 4. There are a number of other basic fields that are also recoded and then represented within the person record. All are candidates for representing with what are called called code tables and/or VIEWs. They could contribute an equal value in space savings.

The code table, AGE_RECODE from Figure 1 is used again in Figure 6 to create a list of persons. Compare this with the examples of Figure 3 (especially example 2).

```
/*
     create a list of 25 to 29 year olds
*/

create table a25_29 as

     select p.ppseqnum, p.ppos

          from person p, age_recode a

               where p.age = a.age and a.rage1 = 6
```

Figure 5. Creating a new table and using an AGE_RECODE code table.

As a practical matter it is more convenient to perform the steps described so far before creating the subsets of children, adults and military personnel. It could however have been done after the subsets were created. It would require less processing to perform the creating of occupation code tables on just the adult population than on the entire person population. Along the way a good deal of tablulating is performed to verify the legitimacy and accuracy of the step. The tradeoff is between the larger single step processing, the time that elapses and what executing a false step costs. Even after the subsets were created more unused fields were eliminated, some from all three sets.

DITEMS

DITEMS stands for the fields identified as DITEM21A through DITEM22F. As a series of questions about labor force participation they apply to a small portion of the total sample and even in this grouping they could be divided further into more mutually exclusive subsets.

DITEMS Table	Universe Size	Fields Eliminated Bytes	Net Bytes
Total	19566	-	547,848
No Allocation	18365	514,220	-
At Least One Field Allocated	1201	-	33,628
		Savings	514,220

Table 6. Impact of person record DITEMS fields for MIS = 1; 28 one byte fields - DITEM21A thru DITEM22F (repetition of the key is not included in the net calculation).

In this case then 1201 individuals are represented as being in this collective population. SQL can be used to examine their particular characteristics and perform some analysis. SQL joins can be used to create a new structure of just this population in conjunction with the basic data represented among the adults.

Children, Adults and Military Personnel

As stated earlier the partitions of the person data could have been done at any time but were delayed until after the 'filler', many of the unused and redundant fields were removed (code tables built). Even after this step new tables were created as it was clear additional fields could be removed. By creating these partitions many 'not in universe' fields, especially obvious in children records, become unnecessary.

```
create table  fp1_civ15 as
     select * from person p
          where p.popstat = 1      /* n = 14,897 */

create table fp1_milt as
     select * from person p
          where  p.popstat = 2     /* n = 81 */

create table  fp1_y as
     select * from person p
          where  p.popstat = 3 and p.age = 14  /* n = 284 */

create table  fp1_y14 as
     select * from person p
          where  p.popstat = 3 and p.age <14  /* n = 4304 */
```

Figure 6. SQL code for creating first versions of mutually exclusive person partitions.

Space savings results and estimates for work done after this step does not include the military population nor the 14 year olds. In the aggregate for the whole CPS this causes the space savings to be understated by a few hundred thousand bytes at most.

Income Recipiency, Income, Top Coding and Allocation Flags

This set of fields can be handled in a variety of ways all resulting in major space savings and greater clarity of content. One approach is presented and the actual results are estimated from tabulated frequencies. In the arrangement within the person record income recipiency fields are three-valued (NIU=0, Yes=1, No=2) and 23 types of recipiency are identified, however only 11 amounts with 11 corresponding top coding flags and allocation flags are presented since some recipiencies have been summed. In keeping with the practice of being able to reconstruct the original data the 'No' group can be removed and a set of tables created. All adults, age > 14 are considered to be in the universe so the earlier splitting of PERSON has isolated them.

The prefered approach is to compact the 23 recipiency codes making 23 tables recording the key for each type of 'yes' recipiency. Combine and create 11 tables with the appropriate recipiency flags and a single income.

N = 14,897

Total recipiency flags (of all types) set 'yes'...29,209
- approximately 18,000 are wages or interest

Approximate cost is 29209 * 6 175,254

Approximate net savings from flags.....315,000

Approximate net savings from amount fields.....714,000

Total Savings.....1,029,000

Table 7. Estimate of impact from removing recipiency and income to a special set of 11 tables for MIS 1 and adults (repetition of keys is not included).

Top coding occurs so seldom that it is most efficiently stored separately. There are only 48 cases of any top coding occuring for any one field in a person's record. Table 8 provides an appoximation of the space savings.

The income allocation flags record information about both the allocation of recipiency and the amount. There are 3,555 cases of at least one of the person's recipiency or amount fields being allocated. The results for 11 flags being retained as a single table are reported in Table 8. Further mutually exclusive sub-populations would result in ever greater space savings.

N = 14,897

Approximate cost 3,555 * 11.....39,105

Approximate net savings from top coding 164,000

Approximate net savings from allocation.....125,000

Table 8. Estimate of impact from removing top coding and income allocation to two separate tables for MIS 1 (repetition of keys is not included, the cost of top coding has been ignored).

Summary and Results

To assess the overall impact on the March 1986 CPS this section summarizes the results for the person record set and other work done with the household and family record sets but not discussed in this paper.

MIS	Record Type	Fields	Size	Count	Bytes
1	Household Original	71	408	9,638	3,932,304
All	Household Original	71	408	74,145	30,251,160
All	Household 1	43	68	58,719	3,992,892
All	Household 2	50	82	216	17,712
All	Household 3	37	37	3,499	129,463
All	Household 4	37	37	11,711	433,307
		All Household net size			4,573,374

Table 9. Size for CPS Household records - MIS 1, MIS ALL original record, MIS ALL by household interview type (fields have also been converted to internal representations, some table recoding and special population tables have been included for Household 1).

MIS	Record Type	Fields	Size	Count	Bytes
1	Family Original	81	408	7,934	3,237,072
All	Family Original	81	408	64,450	26,295,600
All	Family Compressed	56	115	64,450	7,411,750
		All Household net size			7,411,750

Table 10. Size for CPS Family records - MIS 1, MIS ALL original record, MIS ALL with redundant fields and'filler' removed (fields have also been converted to internal representations, other structuring is also implied).

MIS	Record Type	Fields	Size	Count	Bytes
All	Pers. Original	240	408	157,661	64,325,688
1	Pers. Original	240	408	19,566	7,982,928
1	Children<14	30	38	4,304	129,120
1	Adults>14	103	142	14,897	2,115,374
1	Adults=15	*		284	*
1	Military	*		81	*
1	Others Combined				251,000
				Total	2,495,494
		Estimate for all 8 MIS			19,963,952

Table 11. Size for CPS Person record - MIS All Original,MIS 1 Original, MIS 1 Sub-Populations and Other special tables, extrapolation for full CPS Person record (fields have also been converted to internal representations).

MIS	Record Type	Original Size in Bytes	New Estimated Size in Bytes
All	Household	30,251,160	4,573,374
All	Family	26,295,600	7,411,750
All	Person	64,325,688	19,963,952
	Totals	120,872,448	31,949,076

Table 12. Size characteristics of the Household, Family and Person data extrapolated from MIS 1 to the all of the March 1986 CPS.

Thus the March 1986 CPS can be reduced from 120 megabytes to appoximately 32 megabytes or less. The strategy taken has only explored some of the obvious alternatives.

The task for MIS 1 was accomplished 1) on an IBM XT, 2) with a standard relational query language and 3) a relational database system. The resulting structure is certainly more compact. It is just as compelling that the new tables (structure) make it possible to focus and address questions of concern more immediately.

References

Date, C.J. An Introduction to Database Systems. 3rd Ed. Addison-Wesley, Reading, MA, 1981.

Relational Technology, Inc. INGRES/SQL Reference Manual. Alameda, CA, 1986.

U.S. Department of Commerce. Current Population Survey, March 1986 Technical Documentation. Washington, D.C.

U.S. Department of Commerce. The Current Population Survey, Desgin and Methodology. Washington, D.C., 1978.

A Relational Database Structure for the March Current Population Survey

William A. Gates

Purpose of this Study

- Make CPS data more available ... especially on PCs!

- Facilitate research using the CPS.

- Demonstrate the effectiveness of relational data modeling and relational database software.

- Illustrate advantages of a standard for data operations - ANSI SQL.

Approach

- Characterize the March CPS data file and opportunities.
- Introduce "Normal Forms" defined in relational database theory.
- Describe the basic pragmatics in reducing the size of the March CPS.
- Illustrate putting it back together, generating simple statistics and the use of "Views".

How big is the March CPS?

- **March 1986 CPS** ... 296,256 records

  ... 408 bytes each

  ... requires 120,872,448 bytes!

CPS Basic Structure

- Three record types

Household records	...	74,145 records.
Family records	...	64,450 records.
Person records	...	157,661 records.

- All related records have same identifier ... HHSEQNUM, (FFSEQNUM, PPSEQNUM)

- Household records identified by according to the month in the sample (MIS) code.

- Skip patterns vary by MIS code, question and sub-population.

- Data on tape is sorted by household, family and then person (with multiple families and person groups within households).

Computing opportunities?

- At least 15 million PCs.
- 20 megabyte disk drive and larger is commonplace and optical drives offer even larger potentials.
- Processing speed has gone from a standard of 4 megahertz to 20 megahertz and soon 33 megahertz.
- ANSI standard SQL (Structured Query Language) available in software from PCs to mainframes.
- SQL access is becoming a standard in statistical package interfaces.
- ASCII data representation can be replaced and/or complimented by more compact computational forms.

CPS Data Expectations!

- Roughly 33% of the March CPS data file is filler

  ... so 120 megabytes becomes 80 megabytes!

  "Filler" makes distribution and traditional processing simpler.

- Table-ing "recodes" provides disk storage savings.

- Representation of survey skip patterns are natural in relational databases.

- Decomposition of data by appropriate sub-populations implies great disk storage savings.

- Similarly for variants by rotation groups.

- Relational representation brings clarity to data.

Example: The Household Record

- Four types designated by the HHTYPE field ...

	Fields
HHTYPE 1 - Interview	58
HHTYPE 2 - Group Quarters	50
HHTYPE 3 - Type A Non-interview	22
HHTYPE 4 - Type B/C Non-interview	22

- Total bytes for storage 30,251,160

- Total after "filler" removed 9,119,835

Households Divided

- Four different types of records.

	Bytes		Records	Total Bytes
Interview	87	x	58,719	5,108,553
Group Quarters	82	x	216	17,712
Non-interview Type A	37	x	3,499	129,463
Non-interview Type B/C	37	x	11,711	433,307
			74,145	5,689,035

Relational Database Systems
Common Aspects

- Normal forms.
- Views.
- Query languages.
- Entity integrity and referential integrity.
- Statistical package interfaces.
- Few purists.

The Role of Normal Forms

- Guidelines for data record design ... full implementation is not a requirement.
- Meaningful for relational or non-relational worlds.
- Intended to prevent update anomalies, minimize redundancy and inconsistency.
- Biased for updates.
- Clarifies melding of universes.
- Eliminates the need for reverse engineering of secondary data sources when documentation is lacking.

Examples: MARCH CPS Opportunities

- Is a null value (or NIU) a fact about the key ... whatever the answer ... you do not need them and can easily create them when you must.

- Household records contains a count of persons in the household ... it is a "fact" about the household but it depends on the person records.

- Person record contains spouse's line number rather than a relation ...

  SPOUSE (PPSEQNUM, PPOS_a, PPOS_b)

Relational Theory - Briefly

- Relation ... given a collection of sets $S_1, S_2, \ldots S_n$, then R is a relation on these n sets comprised of a set of ordered n-tuples $\{s_1, \ldots, s_n\}$.

- The S_i are called domains.

- Binary, ternary to nth degree.

- Relation ... like a file only better behaved ... where each record has a unique identifier (a key) and is normalized.

- Tuple ... like a record only better behaved.

- Attribute ... one of the s_i in a tuple.

Other Concepts

- Semantics and normal forms ... you have to know the meaning of data to assess the degree of normalizing that has taken place.

- Functional dependence ... for each x there is precisely one y.

- Mutually independent ... attributes are not functionally dependent.

- Multi-valued facts.

- Transitively dependent ... functional dependence via an intermediate attribute.

- Determinant - an attribute on which some other attribute in functionally dependent.

- Projection ... what you do to get normal forms.

1st Normal Form

- All records must have the same number of fields.
- Variable repeating fields and repeating groups are excluded.
- All records have a field(s) designated as a "key".

2nd Normal Form

- A non-key field must be a fact about all of the key.

- A key may span or include several fields ... a composite key.

- On the person record PPSEQNUM and PPOS are a composite key.

More Normal Forms

- 3NF - a non-key field must be a fact about the key not any other non-key field.

  Age, industry and occupation recodes.

- 4th ... does not contain two or more independent multi-valued facts!

- 5th ... record type cannot be reconstructed from several smaller record types.

- 5th normal form may imply several more record types but there may be many fewer record occurrences.

- Remember - few purists!

SQL - Structured Query Language

- **Select -**

  select distinct age, count(age) from PERSON

  where age > 14

  group by age

  ... a frequency table by age for persons over 14

- **Insert - for adding new rows**

- **Update - for changing values in rows**

- **Delete - for removing rows**

Recode Tables and VIEWS

- Recoding

```
create table AGE_RECODE as
        select distinct age, rage1
            from PERSON
                order by age
```

- Creating a VIEW

```
create view PERSON_AGE as
    select x.ppseqnum, x.ppos, x.age, y.rage1
        from PERSON_NEW x, AGE_RECODE y
            where x.age = y.age
```

Build a Statistical Abstract

```
create table AGE_RFREQ as
        select distinct age, rage1, count (age)
                from PERSON
                        group by age
```

Another example - SELECT

```
/*
     create a list of 25 to 29 year olds
*/

create table a25_29 as

     select p.ppseqnum, p.ppos

          from person p, age_recode a

               where p.age = a.age and a.rage1 = 6
```

Some VIEWs and SELECT Examples

```
/*   Example 1 - using the VIEW within a Select */

     select * from PERSON_AGE

/*
     Example 2 - using the VIEW within a Select
     for  persons > 14
*/

     select * from PERSON_AGE x

               where x.rage1 = 6

/*
     Example 3 -
     writing an equivalent SQL select for Example 1
*/

select x.ppseqnum, x.ppos, x.age, a.rage1

          from PERSON_NEW x, AGE_RECODE a

               where x.age = a.rage1
```

Process Overview

- Build database dictionary.
- Remove filler.
- Choose datatypes.
- Investigate and learn.
- Start modeling.
- Investigate and learn ... keep modeling!

Ideals

- Recodes ... all redundant fields would be removed and a supporting recode table provided ... e.g., AGE.

- Constructed record type redundancy would be eliminated and VIEWS used to create the value ... e.g., PINCTOT.

- Basic records would be completely decomposed by appropriate universes ... children, adults, etc.

Pragmatics

- Incomplete information or some degree of uncertainty ... effort of discovery.
- Advantageous impact on storage savings.
- Estimate of usefulness or cost of reconstruction.
- Can it be done on a PC.

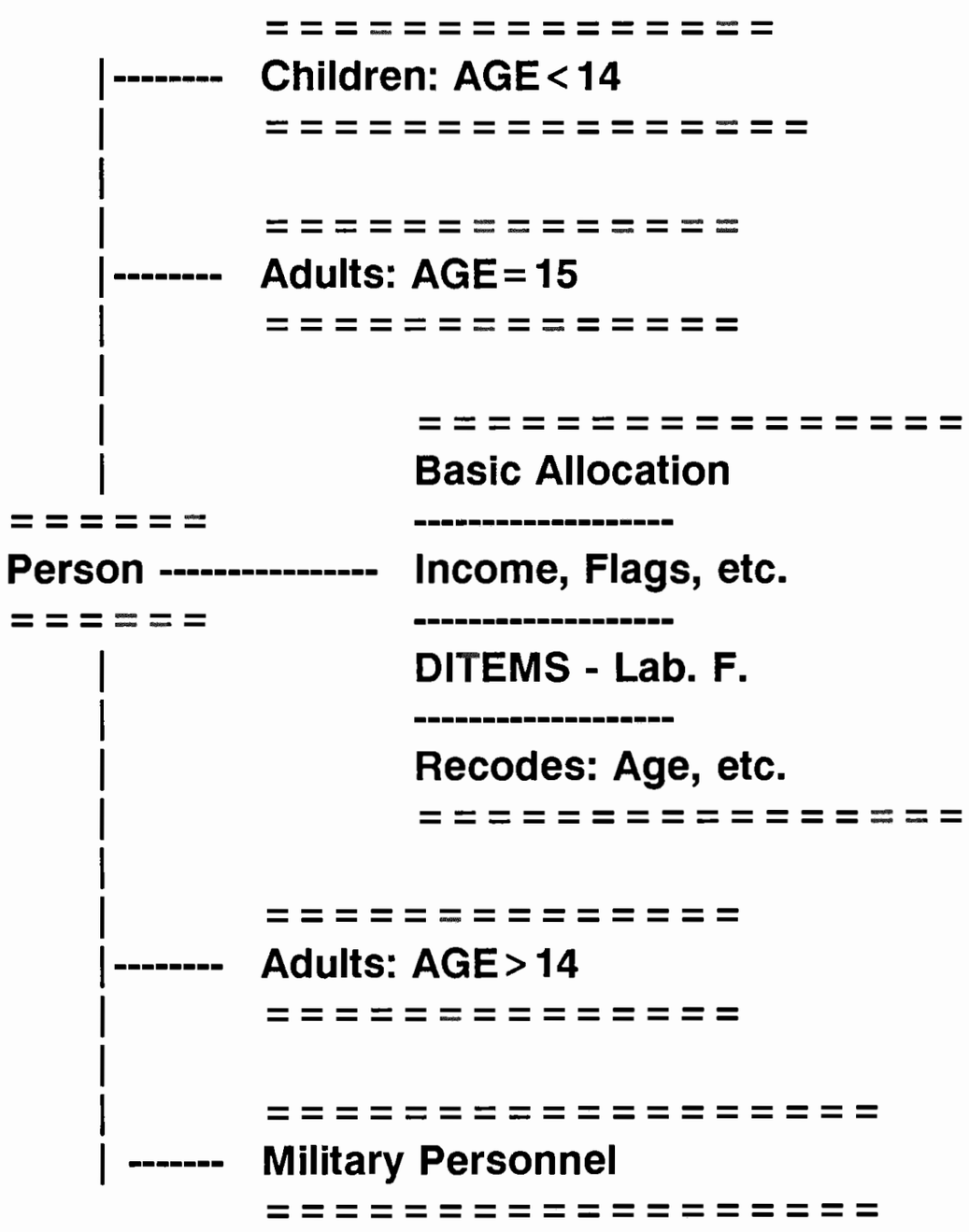
March 1986 CPS - MIS 1

Record Type	Fields	Size	Count	Bytes
Household	71	408	9,638	3,932,304
Family	81	408	7,934	3,237,072
Person	240	408	19,566	7,982,928
			Total	15,152,304

Person Tables...Some Universes

- Filler.
- Never used fields.
- Children ... under the age of 14
- Persons ... age 14
- Adults ... 15 and over
- Military Personnel
- Labor Force.
- Attending School.
- Income Questions.
- Allocation Universes.
- Health Care.
- Auxiliary Tables.
- Making the implicit explicit.

Person Tables



SQL to Create Person Subsets

```
create table  fp1_civ15 as select * from person p
     where p.popstat = 1       /* n = 14,897 */

create table fp1_milt as select * from person p
     where  p.popstat = 2           /* n = 81 */

create table  fp1_y as select * from person p
     where  p.popstat = 3 and p.age = 14
               /* n = 284 */

create table  fp1_y14 as select * from person p
     where  p.popstat = 3 and p.age <14
               /* n = 4304 */
```

FILLER Removed MIS 1

Record Type	Fields	Filler Removed Bytes	Count	Bytes
Previous Total	-	-	-	15,152,304
Household	7	285	9,638	(2,746,830)
Family	3	199	7,934	(1,578,866)
Person	8	30	19,566	(586,980)
		SubTotal		(4,912,676)
		Net CPS Size		10,239,628

Person: Unused Removed - MIS 1

Record Type	Unused Fields	Unused Bytes	Removed Count	Removed Bytes
Person Total	-	-	-	7,982,928
Person Filler	-	-	-	(586,980)
Person Unused	14	29	19566	(567,414)
MIS 1 - Net CPS Size				6,828,534

Person: Redundant Fields - MIS 1

Record Type	Redundant Fields	Removed Bytes	Count	Bytes
Table 3 Net	-	-	-	6,828,534
Person Redun.	8	15	19,566	(293,490)
MIS 1 - Net CPS Person Size				6,535,044

Person: Basic Allocation - MIS 1

BALLOC Table	Universe Size	Fields Eliminated Bytes	Net Bytes
Total	19566	-	430,452
No Allocation	19405	426,910	-
At Least One Field Allocated	161	-	3,542
	Savings		426,910

Person: Labor Force - MIS 1

DITEMS Table	Universe Size	Fields Eliminated Bytes	Net Bytes
Total	19566	-	547,848
No Allocation	18365	514,220	-
At Least One Field Allocated	1201	-	33,628
		Savings	514,220

Person: MIS 1
Recipiency and Income - MIS 1

N = 14,897

Total recipiency flags (of all types) set 'yes'...29,209
- approximately 18,000 are wages or interest

Approximate cost is 29209 * 6 175,254

Approximate net savings from flags 315,000

Approximate net savings from amount fields 714,000

Total Savings.....1,029,000

Person: MIS 1
Top Coding and Allocation

N = 14,897

Approximate cost 3,555 * 11 39,105

Approximate net savings from top coding 164,000

Approximate net savings from allocation 125,000

CPS Household Size

Record Type	Fields	Size	Count	Bytes
Household Original MIS 1	71	408	9,638	3,932,304
Household Original All	71	408	74,145	30,251,160
Household 1	43	68	58,719	3,992,892
Household 2	50	82	216	17,712
Household 3	37	37	3,499	129,463
Household 4	37	37	11,711	433,307
All Household net size				4,573,374

CPS Family Size

Record Type	Fields	Size	Count	Bytes
Family Original MIS 1	81	408	7,934	3,237,072
Family Original All MIS	81	408	64,450	26,295,600
Family Compressed All MIS	56	115	64,450	7,411,750
All Household net size				7,411,750

CPS Person Size

Record Type	Fields	Size	Count	Bytes
Person Original MIS 1	240	408	19,566	7,982,928
MIS 1				
Children < 14	30	38	4,304	129,120
Adults > 14	103	142	14,897	2,115,374
Adults = 15	*		284	*
Military	*		81	*
Others Combined				251,000
			Total	2,495,494

Estimate for all 8 MIS 19,963,952

Summing It All Up

Record Type	Original Size in Bytes	New Estimated Size in Bytes
Household	30,251,160	4,573,374
Family	26,295,600	7,411,750
Person	64,325,688	19,963,952
Totals	120,872,448	31,949,076

Opportunities?

- You can do it on a PC/XT.
- At least 15 million PCs.
- 20 megabyte disk drive and larger is commonplace and optical drives offer even larger potentials.
- Processing speed has gone from a standard of 4 megahertz to 20 megahertz and soon 33 megahertz.
- ANSI standard SQL (Structured Query Language) available in software from PCs to mainframes.
- SQL access is becoming a standard in statistical package interfaces.

Purpose

- Make CPS data more available ... especially on PCs!
- Facilitate research using the CPS.
- Demonstrate the effectiveness of relational data modeling and relational database software.
- Illustrate advantages of a standard for data operations - ANSI SQL.
- Relational representation brings clarity to data.