

Chapter 9

The Guatemalan Commission for Historical Clarification: *Database Representation*

Humberto Sequeira

Introduction

The purpose of the database for the Guatemalan Commission for Historical Clarification (CEH) was to make it possible to process the human rights violations reports collected by the investigators who did the field work in the areas most affected by the violence in Guatemala from 1960 to 1996. The design was straightforward and simple to implement, and allowed for improvement and expansion of the database to other areas of the project.

When I joined the CEH, my experience in Information Systems was in traditional commercial areas: Point of Sales, Inventory, Export/Import, Container Control, etc. The concept of a human rights information system was new to me, and my first reaction was negative. My initial concern was that the case capturing system and database designs were almost complete. In addition, Visual FoxPro, the database application program in use, was not what I would have expected to use for a large-scale project. But I focused on the project as a whole and decided to do the best work I could with the available tools.

The choice of Visual FoxPro was a good one, despite some initial problems. Its ease of design and programming proved invaluable. In addition, we were not limited in the number of users accessing the database. Hence, in-line SQL statements proved to be excellent for testing and searching the database for errors and basic information, and for producing the tables needed for the CEH team's analyses and studies. While the selection of Windows NT Server as our server software was not optimal from the standpoint of speed, it was easy to administer. In fact, the server went down only twice in a year's work. Speed was the major problem with our server and if the CEH had been network-ready, we would have had difficulty accessing, manipulating and distributing the information. Our server was too large for the database, considering it took less than 500 MB to run the database and its software. With distributed systems we could have done much more than only serve the database team. Distributed systems would have supported more extensive collaboration on various subjects, and the ability to share resources would have been an asset.

One major disappointment was that the CEH was not network-ready in their central offices. This was the most troublesome aspect of the otherwise excellent work environment. Without a network we could not give some teams the timely information they needed to do their jobs. Also, we could not track the information the database was producing and some members lacked the technical and statistical knowledge needed to understand some of the graphs, statistics and reports they received. This created an uncomfortable relationship between the database team and other CEH workers.

In this paper, I discuss the information we used to design, implement and develop a database system that addresses human rights violations, which was our principal interest. I describe the current system with proposed improvements, its design, development, testing, training, implementing, contingency proposals and time/cost estimation, with proposed improvements. In addition, I offer recommendations for future human rights database systems in Appendix 1.

Database Representation

At first, an in-house programmer was hired to design the system to collect information from testimonies received by the CEH during their fieldwork. However, the pressure of deadlines, working with new software and intense work environment caused him to resign. Subsequently, Assist, a Guatemalan software firm was hired to carry out the design.

From a user's point of view, the current system was divided into the following parts:

1. Case Summary
2. Violation(s) Pattern
3. Victim(s) General Information

Chapter Nine: The Guatemalan Commission for Historical Clarification

4. Person(s) Denounced for the Violation(s), that is, the Perpetrator(s), the person(s) responsible for the violation(s)
5. Person(s) who Denounced the Perpetrator (Deponent)

This information comprises a *Case*. Using this structure, we obtained valuable information. I explain and discuss each of the above parts in detail in the following sections.¹

Case Summary.

The case summary is where we kept general information about the violation(s) being reported. This is general information, and was not be used for statistical analysis.

In this summary we had information such as:

- researcher name
- interview date and place
- if the interview was taped and how many cassettes were used
- documentation given to the CEH
- given case name (from the analysts' team)
- general date of the violation or the period in which the violations occurred (also from the analysts' team)
- case summary (general overview of the case)
- comments and questions about the case raised by the database analyst
- general geographic information
- key words in the violation(s) pattern
- whether this case was reported to other organizations
- number of victims mentioned in the case

We added *markers* to validate the cases and violations. A team of CEH lawyers made legal judgments about the cases and violations to provide the validations.

Some information collected in the Case Summary is redundant, but it was useful for an overall view of the case. Among the redundancies were date and place where the violation(s) took place and the number of victims mentioned in the case. This information was also stored in Violations Pattern (2) and the victims were also counted in 3. Victim(s) General Information (3), discussed below.

These redundancies were a primary cause of discrepancies in the accounts of victims, but were an aid to quickly showing the number of victims in cases. Thus, massacres were identified more easily. In my opinion, these redundancies should not be eliminated from the Case Summary. They should appear in the Case Summary and the system should provide an account of them, but this should not be a user-editable field.

Keywords gave us the ability to store information that otherwise would have been lost. This was where data such as *Modus Operandi*, Military Strategy, Violence Against Children and Women, Cruel Actions, Destruction of Goods, and Religious Violations, etc., was stored.

However, researchers should not make conclusions based on statistical analysis using the keywords for two reasons. First, the key words qualified **cases**, not victims or violations, and cases do not have clear substantive boundaries that allow them to be meaningful quantitative units. Second, the key words were not always applied with the care and precision that were customarily employed in other classifications. The purpose of the key words was to allow analysts to group cases that share some idea such that the cases could be found again and revisited for qualitative in-depth analysis.

We were able to group Cases in important categories using the Name Assigned to the Case. We used it to differentiate between *Normal Cases* and *Massacre Cases*. We were originally using the keyword Massacre to identify the Cases that had more than five Arbitrary Executions. However, problems in the definition of massacre caused some cases to be identified as massacres and others to be left out. So in the final stage of the CEH, we agreed to use the first part of the case name as Massacre to identify massacres. This was important, because we then could differentiate between Massacres and Normal Cases. A keyword validation is a more useful method, because the

¹ Editors' note. The reader of other papers in these proceedings will notice that this structuring is defined elsewhere. We retained these redundancies so that each paper is self-explanatory.

name can change and all references to a case being considered as massacre will be lost. However, with the use of a key word to identify a case called a massacre we have an historical analysis of the changing definition of massacre during the project.

Violation(s) Pattern.

The Violation(s) Pattern is where we kept track of **what, where, when** and **to whom**. As discussed in (Ball, 1996), this structure is essential when the case has more than one violation. Clarifying the structure of this information component was the most discussed aspect of how the data were represented in the database.

Here we stored the following information:

- Violation order
- Type of violation
- Certainty level for the violation
- Place of the violation
- Date of the violation
- Group responsible for the violation
- Type of responsibility for the group responsible
- Type of evidence about the responsible group
- Number of identified victims
- Number of collective and anonymous victims affected by the violation(s)
- Number of victims who died
- Disposition of body
- Location of the burial site
- Information about victim death
- Was body identified
- Was body found
- Was the victim tortured
- What kind of torture the victim received
- How the person reporting the violation knew who committed the violation.

As you can see, the Violation(s) Pattern was the category where we stored most of the information given to members of the CEH. It includes some redundant information, which concerned the victim's account according to the violation pattern. The violation pattern was the easiest way for us to represent violations to one or more persons. Some information here, like torture information, was also presented in graphical form.

Due to this design the representation of **multiple** victims receiving one or more violations was excellent, but it was not the best way to represent **one** person receiving one or more violations. The reason that this was not a good way to deal with one person receiving one or more violations was that is that the basis of our system design was to use violations as the unit of analysis rather than victims. We could give our co-workers breakdowns based on victims, but we wanted them to think in terms of violations. Some of the officials at the CEH could not grasp this approach. This is understandable, because the information they wanted to share with others were personal accounts, such as how many people were affected, not how many times they were affected.²

For example, some violations, like Arbitrary Execution (or any other in which the victim died or disappeared), were straightforward mappings of persons to violations in a one-one relation. But others like Freedom Deprivation were not so clear. In such a case, one person can be deprived of his/her liberty on more than one occasion, not only in the same case, but also if the person appeared in another case and was again deprived of liberty. Thus, we would have a one-to-many relation of one person to multiple violations.

It is important to remember not to have more than one death-causing violation in one pattern. This may seem pretty clear, but it happened to us occasionally. We took the necessary steps to fix this problem. We used a listing on the person's name, place and time of violations, along with age and parents' name. We also got some help in dealing with this problem when we stored violations. For instance, we have the Violation Order field, as we capture violations in an orderly fashion. So if

² For an explanation of the importance of this approach, see (Ball, 1996). Patrick Ball's concepts and guidance were of great assistance to me in my work on this project.

Chapter Nine: The Guatemalan Commission for Historical Clarification

a Freedom violation occurred first and then a Sexual Violation, then the Freedom violation will be given a lower order number. The order field was also useful in pinpointing the responsible forces participating in the violation, so in the same order number we could have the Army participating in the same violation with the Military Commissioners.³ The restriction for this use is that the violation must occur on the same place and date.

The following is an example in which the Army, Military Police and Civilians participate in one Sexual Violation.

ORDER	VIOLATION	GROUP RESPONSIBLE
1	Sexual Violation	Army
1		Military Police
1		Civilians

Some information was infrequently used, either for information analysis or questioning by the investigators making the interview. An example is information about the disposition of the body and whether the burial site of the victim was known. Also, in this category was information as to the death of the victim, and if the body was identified, or not found at all.

As we discussed, during the course of the project we realized that we had to update the Violation Pattern, and we came up with a solution, which will be discussed in detail in the section on database design.

Victim(s) General Information.

Here we store the information about the person suffering the violation. This “person” can be an individual or a group. If it is an individual person this part must include the minimum required information. Or this information may describe a group of persons who have some characteristics in common. In particular, we store here the following information:

- Last Name
- Second Last Name (Mother's Last Name)
- First Name
- Alias (es)
- Sex
- Documentation for the person (if given)
- Date of Birth (DOB)
- Age at the Time of the Violation (if DOB not given)
- Year of age (if DOB not given)
- Certainty on the DOB
- Nationality
- Place of Birth
- Language
- Organizational Sector of Victim
- Residence at the Time of the Violation
- If the Victim Had to Move Because of the War
- Name of the Father
- Mother
- Wife (if married)
- Sons/Daughters (all these in the form of Last Name, Second Last Name and Name and Ages at the time of the violations or in present time)
- Civil Status
- Comments

³ *Ejercito and Comisionados Militares*, respectively.

As mentioned earlier, the victims were categorized as Individual or Group. The information stored for both categories was the same except for some information that was unnecessary for Group victims such as age, date of birth, relatives' names, etc.

In the Group information we entered information regarding a group of persons, such as Sex, Birthplace, Location Where Found, Age Range, and some other general terms. We used this information in the name we gave to the group. For example, if a group of children was found dead near the river Rio Negro, we used the Last Name, Second Last Name and Name to call them "Children of Rio Negro."

We tried to determine an approximate number for how many people were affected by the violation using these classes; 2, 4, 5, 10, 11-20, 20-50, more than 50.

Sometimes it was not possible to use the classes, and terms such as "many," "a few," and "a lot" would appear, indicating that we had no approximate number to use. As time went on we needed a number to assign to these victims. We were conservative and assigned an approximate value of two. We also were faced with the anonymous victim assignment problem and gave them the same approximate number of two. The Anonymous victims were people about whom we knew almost nothing, and for that reason we stored their information in the Violation(s) Pattern.

Perpetrator (s) General Information

The information stored here closely resembled that stored in the individual victim component. This included the following information:

- Last Name
- Second Last Name (Mother's Last Name)
- Name
- Alias (es)
- Sex
- Documentation for the person (if given)
- Date of Birth
- Age at the Time of the Violation (if DOB was not given)
- Year of age (if DOB was not given)
- Certainty of the DOB
- Nationality
- Place of Birth
- Language
- To which group the person belonged and his role in that organization
- With what violations and cases this person was charged
- Type of Responsibility and Evidence for this Accusation
- If the person participated in other violations at any time or any place
- If the person reporting the violation knew where this person is now going and where he might be located
- Comments.

The most secret materials kept at the CEH were the perpetrator's name along with the information about the person reporting the violation, because it was essential that the information given to the CEH should not be used for revenge. **In fact, the most important aspect of the Database is to produce information that is without bias and not manipulated in any way to benefit any group or person in particular.**

The data stored here was information about the individual perpetrators identified in a violation or violations. We soon found out that an individual perpetrator could appear in more than one case. This is the same as for victims or persons reporting the violation to the CEH, and we made the necessary adjustment to reflect this relation. A victim can also be a person denounced for a violation or/and a person reporting a violation in some other case. The lack of accurate information was responsible for the problem that a person might have been mistakenly referenced as a perpetrator; however the analysis team made its best effort to avoid this situation.

There can be many persons responsible for a case, but only one can be represented as the Person Denounced for the Violation(s), the Perpetrator who is the person responsible for the violation(s). Also, we did not accept a general description such as "Person living in the village of Baraza in 1991." We demanded a name or at least, an alias.

Chapter Nine: The Guatemalan Commission for Historical Clarification

Person(s) Who Denounced the Perpetrator

Here we stored the information for the persons. (*deponents*) reporting the violations to the CEH. As with the victim(s) general information part, it can represent an individual or a group of persons. We stored here the following information:

- Last Name
- Second Last Name (Mother's Last Name)
- Name
- Sex
- Documentation for the person (if given)
- Date of birth (DOB)
- Actual age (if DOB not given or known)
- Year of age
- (If DOB not given)
- Certainty of the DOB
- Nationality
- Place of birth
- Language
- If the person making the report is an individual accused of a violation (from an analyst's point of view) and if that individual is accused
- The perpetrator case and identification number
- If the person making the report is a victim
- The victim's case and identification number.
- If the person making the report was a refugee
- Displaced person
- Resettled person
- Returnee
- Survivor of a massacre
- Victim of some other violation
- Witness or violence survivor
- The relationship of this person to the victims of this case. The information required here is the Case number, victim's identification number and relation to the victim

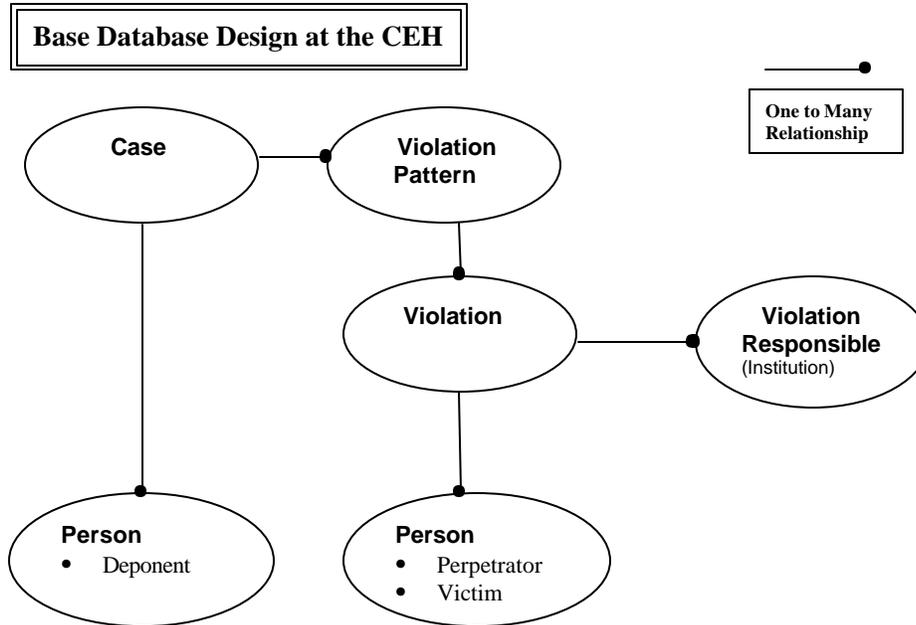
Once again it is important to note that we should always protect the identity of the persons who have trusted the CEH with the information stored by this screen.⁴ It should remain only in the archives of the CEH.

It is essential not to change the stored information in some tables. For example, David Sequeira is a victim (victim #2), his mother is Magda Sequeira and his father is Walter Sequeira. We save this information on the victims screen. In this case Magda is also a victim (victim #6). Since David is also the person reporting the violation, some users entered incorrect information stating that the relationship was son/daughter, thus changing the relationship between Magda and David.

Database Design

The following flow chart is an overview of database design at the CEH.

⁴ When we refer to a screen, we are referring both to the screen as shown by the computer for data entry and the physical form that may have been used to record the information.



As mentioned previously, Patrick Ball designed the CEH relational database so that we could use SQL syntax to code a wide range of information. The design not only counted violations as the primary unit of analysis, but was flexible enough to support other units of analysis.

The first table used is the CASE table,⁵ which is the main table in the database. In this table, we stored the basic case information, such as the information stored in the Case Summary. The field investigators assigned most of the case numbers according to a limited set of numbers given for each area, so that an area such as Guatemala City could range from 1 to 1000, for Zacapa, 1001 to 2000, for Quiché, 2001-3000. In a similar fashion, ranges of numbers covered all the sites where CEH offices operated.

We used a technique called *automatic number generation*, in which case #9 will not necessarily be #9 internally in the database. Instead, case #9 could be internally represented by CA0000257. The cases were numbered, as entered, one by one by the database and given a sequential number. Although you might normally expect that case #9 would be entered before case #1125, with automatic number generation, this is not necessarily true.

In addition to the information in the Case Summary, we stored here the Case Creation Date, Case Modification Date, User Who Entered the Case and User who Modified the Case.

The master key for this table was the CASE_ID, which was used throughout the system.

Following with the case information we also had a table to store the keywords used to qualify a case, CASE_CLV. A case can have more than one qualifying keyword. The information was CASE_ID, PALA_COD and AUTHOR. Using this table, it is possible to record some essential information that was not recorded in the VIOL table and otherwise would have been impossible to obtain. PALA_COD is the keyword code.

If a case was reported to another institution besides the CEH we used a table, CASE_DEN. Here we stored the CASE_ID, the person who reported it, the institution that received the report and the date.

The PATTERN table is the glue that holds together the information in the database, as here we store the Pattern number along with the number of identified, collective and anonymous victims, and the estimated magnitude of the number of persons mentioned (“many,” “a few,” “a lot,” etc.). This table is directly linked to the CASE table in such a way that a case can have several PATTERNS. The key for this table is CASE_ID+PATTERN_ID.

The PATTERN table is the parent table of the VIOL table. In the VIOL table, we stored the pattern number, order number, the violation, the date, the place, and some geographic description to

⁵ In the discussions that follow, I follow conventional practice in information system design and implementation of capitalizing the names of tables, keys, codes, and field names when appropriate.

Chapter Nine: The Guatemalan Commission for Historical Clarification

help us identify the place and violation certainty level. The full key for this table is PATTERN_ID+ORDEN_NUM+VIOL_ID, which we can link to the CASE table using the PATTERN table. Thus, if we had only the VIOL table, some of the information would be available, such as **what**, **when** and **where** but we would still be missing **who** were the perpetrator(s), which is the table I describe next.

We stored information about the organization responsible for the violation in the table VIOL_RSP. We stored the VIOL_ID code for the perpetrator group, type of responsibility, evidence, and place of the group assigned to the violation. In this table we can show if more than one group shared the violation, up to n groups. The key for this table is the VIOL_ID.

The table PATR_VICT stores the victim's id number and the pattern in which the victim's violation occurred, so we only need the PATR_ID, PERSON_ID and victim number. Thus, we have all the violations that a person suffered in a given case or in more than one case. It doesn't matter if the victim is individual or collective; we only need the PERSON_ID number. We also use the VICT table as a backup for PATR_VICT; the only difference here was that we also stored the CASE_ID number.

The PERS table stored all the information regarding a person, any kind of person. The key for this table is PERS_ID. Victims (Individual or Collective), Persons Accused of Violations, Persons Reporting a Violation (Individual or Collective), Brothers, Parents, Son/Daughters, Wives, Husbands, etc. Every name of a known or unknown Individual or Collective person is here. By July 25, 1998, we had about 30,000 persons in the PERS table. We had three fields to mark if a person was a Victim, a Perpetrator of Violations and a Person Reporting a Violation. We also had a field to show if the victim was an Individual or a Collective person. Important information regarding the person such as: Full Name, Alias (es), Sex, Age, DOB, Nationality, Language, Documents, Place of Birth, Civil Status, Comments, Age for Deponent.

Because we wanted full information about a person we were faced with a problem in time representation. For example, a person who doesn't remember his DOB (a frequent occurrence) reports a violation he suffered in May 1982 when he states his age as 21. He then reports a violation he suffered in July 1993, when he states his age as 34. But this is 11 years later, and if his 1982 age were correct, then he should be 33. The report was filed in 1996 and in it he says he is 36 years old. If the 1982 age were correct, he should be 35. On the other hand, if the 1993 age was correct he should be 37. Which age do you accept as the age of the person? We represented the last age entered in the database, because the person can't be entered in the database again to show a different age. If this person was reporting a violation, we stored his age in a different field, and when performing calculations we used the best age possible. Still, it is not a guarantee that we had the person's correct age.

We also had problems with the Comment field. Some comments about the person were about when he was a victim. Others were about when he was reporting a violation and others when he was accused of a violation. This was because the victims, persons reporting a violation and perpetrator all share the same information; the individual can have more than one role. This was a programming error and was fixed as soon as I found out about it.

For any dates used in the CEH database, we used the Russian date format, yyyyymmdd (year, month, day). We used this format so that we could use zero (00) in representing the month and day. We had to, because on some dates we could not get any certainty at all, for example:

- In the winter of 1986
- Some day in January 1981
- At the end of Easter 1983
- I don't remember when

The IHCH table is where we kept the information about the individual accused of the violation (the perpetrator), such as CASE_ID, PERS_ID and IHCH number. This is basic information about this person and useful if we knew that a person participated in a violation, but were not sure what the violation was. More complete information on the person's participation can be found in IHCH_VIOL which included the CASE_ID, PATR_ID, VIOL_ID and evidence and responsibility.

The PERS_CLS table stores information on the activities of the victim, or to what organizations the person belonged (such as union, religious organization, etc.). The information needed is the PERS_ID and VICTCLS_COD (Type of victim according to our catalog).

Still working with the PERS table, we used the PERS_HIJO to store relations between people. Here we show the relationships of the victims or persons reporting the violations to the CEH. This is a good approach and worked for a while, but catalog limitations (imposed by ourselves) made us misplace some information. For example, if a person only had one parent, we could not know if it was the person's father or mother, because we used Mother/Father as one category.

We used the group of tables PATR_DS, PATR_MU and PATR_TO to store information on whether the victim was disappeared, dead or tortured, respectively. This information is basic and the only thing required here is the victim's form or screen number along with the PATR_ID.

The USERS table stored the information about users who entered information on the database. This information included user code, name, security level and password.

Some of the other tables in the database are the catalogs, which we used to complete the information, such as the Institutions, Type of Victim, Key word, Relation, Language, Nationality and others. We hope to see them when the CEH authorizes the use of the information stored in the database.

Application Programming: Development, Testing, Training, Implementing

As mentioned previously, Assist, a Guatemalan company, did the initial programming and I did the balance. The system was network-ready and with minor adjustments after it was implemented, worked quite well. In this section, I give some details of the process.

Once the database design was provided, Assist delivered their product in about five weeks. I needed completion quickly, so we agreed that I would finish the application and they would correct some residual problems. This was of value to me and I became familiar with the final product. At first I wasn't happy with Visual FoxPro as the development tool, as I formerly worked with UNIX database systems and used Visual Basic for a visual interface. However, Visual FoxPro had an edge: no per-user cost to access the database, inline SQL access (not possible with Visual Basic), no need to back me up in UNIX administration and most important, no delay in startup. So I gracefully accepted Visual FoxPro as the tool of choice. I have not regretted that decision, and would only complain about the slow speed of operation.

The first version of the system was a FoxPro Application, which meant it ran inside the Visual FoxPro Environment and each machine had to have FoxPro installed. We had a number of problems with this version. The application kept producing Error Protection Faults (EPF) either from the Kernel or from Visual FoxPro, two or three times a day per machine. Fortunately, the tables were not damaged (as in previous versions of FoxPro) but too much time was lost, as we had six machines. So we built an executable file from FoxPro and reduced system outages to only two or three times per month resulting from problems affecting the Kernel.

The FoxPro executable is not a true Windows executable, but behaves as one. Using the FoxPro executable, we did not need to install FoxPro in the machines, only several DLL files necessary for some functions. Everything else is in the executable file.

As mentioned, we did not have the proper time to test the systems (about one half-hour per change no matter what size) but it worked out well. The system was solid, so we did not lose a lot of time. Training was also informal, but we tried to give the people working with the system at least one full day to become familiar with it.

The CEH system can still be improved. Some procedures can be automated to give the user entering the case more speed and less overhead. These are:

Problem	Solution
Beginning and Ending date of the Case	This can be taken from the Pattern form automatically (scanning all violations)
Places where the case took place	This can be taken from the Pattern form automatically (scanning all violations)
Keywords for the cases	Keywords cannot be repeated

Chapter Nine: The Guatemalan Commission for Historical Clarification

Other institutions where a violation was reported	A catalog of Human Rights or other institutions can be made, including military institutions
Only one interviewer can record the case	More than one interviewer can participate in recording the interview or case
Identified, Collective and Anonymous victims in the Case form	This should be eliminated from the case form, and can be obtained automatically from the Pattern and Victims form
Identified and Collective victims from the Pattern Form	This should be eliminated from the pattern form, and can be obtained automatically from the Victims form
Where the victim lived when the violation occurred	This should be coded from the Places Catalog
Language of the persons	This should be the ethnic group to which they belonged; their language should be stored in a separate location
Type of victim	Type of victim can't be repeated. If this happens, more violations would be counted in the type of victim analysis
Personal relations	They all should be independent. Do not use something like "Father/Mother." Instead, separate them
Comments of the victims	Should be shared in all the forms. Should be common to the person
Group to which the person belongs	Group cannot be repeated. The only case is when the charge inside the group changes; thus we can keep a history of the person (mostly related to perpetrators)

As to the application programming, it doesn't matter in what language or on what platform it is developed. It should be fast for the users, and aid them in any way possible in their daily routine. This is dull work, with almost no changes in the daily routine. Every step should be taken to make the application flawless.

As time went by, a need for more information was evident. The CEH database was network-ready, but the CEH organization was not. Some teams needed to work on the cases to make quick studies (they also needed to write a short summaries), and needed many varied and unpredictable types of information. One of these teams was the Ordinary Cases Team, which needed ALL the information for the case on one page (except for perpetrators and deponents, not given for security reasons).

Accordingly, I decided to build an off-line system with requested updates, so that when the user finished with the cases I could provide them with an update, keeping the summary they have written. This took a lot of my time, because I had to use diskettes to transfer the program, and there were eight persons working with this software.

This subsystem appeared like this to the user:

Formulario Para Equipo de Casos Ordinarios

Case Number

Case Name

Generic Place

Save

Exit

Violations (Pattern Number/Violation/Date/Place/Org.Responsible)

Analist Comments

Case Summary

Victims (Pattern Number/Victim Number/Full Name)

I created another sub-system for the Recommendations Team. In it, I showed the comments of the deponents and allowed building keywords on each comment. We could produce statistics on the most common keywords to represent the concepts appearing in the information being collected.

In the Appendixes to this paper, I give recommendations for system automation in similar projects (Appendix 1), detailed information on our SQL queries (Appendix 2), and my recommendations for information integrity and security (Appendix 3).

Chapter Nine: The Guatemalan Commission for Historical Clarification

Lessons Learned

In Appendixes 1 and 3 I give the body of my recommendations for future similar projects. In this section, I present other, more specific lessons that were learned in the course of my work on the CEH project.

Entity, Function	Problem	Lesson	Issues
Case reports	Started giving general date and place, Members of the CEH increasingly wanted more exact information in their reports	Show all the places and dates of the violations in a concise manner, take information from the violation (i.e., where it occurred and when). Team must decide how to handle use of information in case reports.	Must not be a user-editable field. Only so much information can be shown in a listing.
Violation pattern screen	Sometimes users would open screen and then stop.	Know who opens the violation pattern and then stops	
Personal information	Inconsistent ages reported by individuals in different cases	Make table with PATR_ID, PERS_ID and AGE, to store the information of the person's age in more than one case.	This table is a modification of the PATR_VICT table in use at CEH, without the AGE field.
Screens	Inconsistent information	Make all the information on the form or screen consistent	Discipline in the design process
Y2K problems	Unpredictable	Store dates in the Russian format using 00 for month and days	Easy to do, was done for CEH
Personal information	Confusion about whether the parent was mother or father.	Use two separate categories	
Hardware & software	Some machines had problems with certain installed virtual drivers	Remove the offending drivers in advance.	Knowing in advance which drivers are likely to cause problems
Workplace	Pressure on users resulting from daily quotas and other factors, limited individual capabilities for pressure and for work	Database administrators should review with data entry personnel to get speed without compromise of accuracy, individual progress reports rather than competition Select and hire users with necessary knowledge and ability to work under pressure.	May call for change in management style of database administrators Verification of experience Getting users with similar capabilities

Appendix 1

Recommendations on System Automation.⁶

Introduction

In this Appendix, I make recommendations for system automation in any similar project. In my work at the CEH, I became concerned about the CEH structure's weakness with relation to the group work that was about to be performed for the CEH report. Also, I make observations on the CEH computing structure and compare its structure with other more general projects and databases. I believe that if followed on similar human rights projects, the recommendations in this Appendix will enable future systems to avoid these problems.

Today's technology makes our work easy: it helps us to plan, organize and administer work. In collaborative interfaces such as that found in the CEH, the lack of planning in topology for computer networks and data servers for document administration had negative effects, causing setbacks and problems in the administrative process.

I document the need for software (locally/internally produced programs or packages from established companies) and hardware (equipment and accessories). Without these two components, it would be impossible to achieve these tasks. Today, the need for access to, and administration of, information mandates these programs communicate with each other. Hence, the organization can be seen as a sensitive, responsive, and interconnected system. Some of the necessary programs for working on all aspects of the documents are so simple and standardized that they can be internally or locally produced. However, others are so complex that the best investment might be to buy an external package that produces the desired results and meets our information and communication needs. The decision-makers in future projects should make these decisions explicitly to fit their particular needs.

All of the programs should be multipurpose and have the same user-friendly and intuitive interface. And of course, in the final analysis, they should meet both the needs of the user team and the organization.

As to hardware, the shared use of available resources (laser printers, color printers, modems, scanners, and hard discs) is one of the most pressing issues. By sharing resources, the organization can focus on acquiring the correct peripheral equipment and reduce costs. For example, a printer for every computer is no longer necessary. Information could be stored in a centralized manner in archive servers, and these servers could be divided into work groups.

Our purposes are special and human rights organizations are almost invariably under-funded. One solution is to contact the vendors of software in the beginning of a project, and possibly the Business Software Alliance (BSA), to obtain licenses for each package. By explaining the purposes and objectives of the organization, it is likely that the organization could obtain savings or discounts in acquiring software. The software vendors may be responsive to knowing that the use of their product will become public knowledge. This has in fact occurred frequently in similar situations. Often the vendors are willing to donate or provide at a large discount the version prior to the current version they are marketing.

Systems Automation for the Human Rights Project

I recommend four major approaches for the system automation plan:

1. Magnetic identification system for team members.
2. Bar code identification system for documents, supplies, and office furniture.
3. System physical facility including wiring, equipment, networking, communications, etc., for the project systems.
4. Workflow system for automating the office.

Magnetic identification system for project members

With the use of a unique identity card for all project members the physical and logical administration cost could be reduced. Personnel could become members of one or more different work

⁶ I thank Walter Sequeira and Eduardo Meyer for their participation in this paper.

Chapter Nine: The Guatemalan Commission for Historical Clarification

groups in which they would be allowed access to different documents generated and/or used in the project. This card would be valid only during the employee's physical participation in the project.

Physical security could be centralized with the personal identification number on the user's card; both the personnel and security offices would administer access to offices.

This magnetic code is generally the same as that seen on credit cards. The following benefits would occur with the use of the magnetic identification system for project members:

- Any team member would be easily located.
- There would be oversight of the project team members' time.
- Access to documents would appear more dynamic and beneficial (in combination with point three) since every document would have an owner at all times (whether its place of origin or the person who requested it) and would be easily located, ready for its return or subsequent processing.

The project member's photograph should be included on the identity card in order to make visual confirmation simpler and faster. The identity card should be non-transferable and should be destroyed once the employee terminates participation in the project.

Bar code identification system for documents, supplies, and office equipment

The bar code is a simple and economical way to label multiple classes of physical objects. In a project of this magnitude, management of documents is a priority. Documents should be maintained in good condition while still being easy to access. Their security and sensitivity should also be taken into account. Documents can be one page, lists, graphics or complete books; this makes their complete identification necessary.

The use of the bar code is a viable and, at the same time, economical option. All of the documents that are accessible to project members should have a unique code created locally in the project.

Frequent problems that occur in locating, receiving and delivering documents would be drastically reduced. With the combined use of project member identity cards, the following information on documents could be known:

- to whom it was delivered
- who has it
- where it is
- how many times it has been requested and by whom
- what time it has been used and by whom

The use of bar codes for the office team cuts administrative costs and time. Thus, office team control will be transparent and, above all, organized. The organization will have knowledge of the location of teams and to whom they were assigned.

The administration of supplies in a project of this magnitude is of vital importance since it offers:

- Projections that can be made monthly, tri-annually, biannually and annually.
- Reports on different teams' and departments' demands.
- Help in estimating the project budget using different available projections.

The use of bar codes together with the fourth theme of workflow would form a formidable combination in relation to the access and classification of documents when the option of data entry is used.

Structured wiring and necessary equipment for the project systems

None of the proposed systems in this paper would be feasible if the project did not have an adequate infrastructure. The network is the spinal cord of the whole system. The design should plan for such extra capacity in relation to the physical capacity of the location at the time of planning, to maximize workstations, and to estimate the different types of users that can occupy these places and their volume of demands and production.

Different types of project teams and their members generate different demands, such as:

- Groups that work exclusively as a team with multiple documents.
- Groups of members modifying the same documents.
- Team members with pressing demands for manipulating graphs.
- Team members with high levels of demand for printing, some in color.
- Specific employees from the different databases used in the project.
- Mobile uses, such as portable or cellular PC's.

All of these demands can be centralized in equipment (computers). The number of users will determine these demands and their specific needs.

Once a locale is chosen, requests for networking should be solicited, always keeping in mind an available route in case of growth.

One important consideration is the physical security of the wiring, since it cannot in any way jeopardize the information that will flow in it. This is the main purpose the wiring fills for a company or professional person, since the wiring necessary for linking more than one floor in a building is laborious and the limitations of distance and security require different hardware and software selections and communication protocols.

Many kinds of available networks must be taken into consideration, and the types, models, and features are constantly changing. At the current time (mid-1999), for an organization with fewer than 200 members and individual teams with fewer than 30 members, I recommend a local area Ethernet type network, the most popular, easiest to administer and most economical in a PC interface.

Workflow system for automating the office

The organization of information, both electronic information and traditional, is a project of vital importance. A new term in this realm is the *workflow*. With the combined use of hardware and software in the form of computer programs, scanners, faxes, modems and others, documents can be administered in an orderly manner.

With these new technologies documents can be:

- Indexed with a unique key
- Stored in a specific location
- Consulted on screen
- Printed on some device
- Sent by fax
- Accessed by different search methods
- Modified for various users

Often, a document needs to be revised or modified by more than one person until it arrives at its final destination. Here is a typical workflow for working with such a document; it goes from Workstation 0 (origination) to Workstation 5 (delivery of final document) after having been revised by employees 1, 2, 3 and 4. The workflow is defined in such a way that the document can be returned to the previous employee if it contains errors, or passed to the next employee if the previous revision is satisfactory. The document can be accessed for consultation at any point in the flow and other users can make modifications on it (if that is how it was defined in the flow). Maximum time periods that an employee can have the document can also be established. If the system detects that the maximum time has passed, administrative alerts are sent to the predefined users.

One must take into account that at no moment has the document been printed; all of the revisions and modifications have been done electronically. This is a small example of a document. The flow can be increased in complexity, demands, users, functions, etc. All of these functions define an electronic office. Imagine how at the moment they are generated, all of our memoranda, faxes, lists, graphs, etc., would have the ease of being sent to the necessary work posts.

One of the disadvantages of these systems is that the people who design the workflow must have a clear idea of the needs and demands of every flow, since not all require the same restrictions nor do they go to the same users. Another disadvantage is that it may be difficult initially for people to work from the screen, rather than with hard copy. However, by providing them an adequate monitor, an optimal resolution screen and an infrastructure of fast networks, these obstacles can be overcome. The workflow will not completely replace paper since some lists, reports, and other types of documents that call for revision will be printed.

There are different kinds of workflow software on the market today. Strategic project planning is what determines if only one will be used or if the configuration will be a combination in relation to the size of the work groups. That decision will determine the type of software to be used.

Server/Workstation Hardware & Software

This is an open topic. It primarily depends on the economic resources available for the project and the persons involved in the planning and their preferences. Some projects will be small;

Chapter Nine: The Guatemalan Commission for Historical Clarification

others will have almost all the correct technology available. Availability of product and support in a given geographical location can also influence the choice of hardware and software. My choice for software is to go with Microsoft® products, except for the database, where you can choose from many vendors with more robust products.

Based on my CEH experience, I recommend consideration of the following:

Feature	Server Side	WorkStation Side
Operating System	WINDOWS NT SERVER 4.0 or better Any UNIX (Aix, Digital, SCO) or LINUX flavor	WINDOWS NT WORKSTATION 4.0 or better Windows 95/98 (If necessary access through emulation, ODBC or Middleware)
Software Network Protocol	Will depend on OS, but TCP/IP for large projects. NetBEUI for small projects.	Will depend on OS, but TCP/IP for large projects. NetBEUI for small projects.
Hardware Network Protocol	Ethernet/Fast Ethernet Fiber Optics	Ethernet/Fast Ethernet Fiber Optics
Database	Will depend on the Server platform Oracle from Oracle Corp. DB2 from IBM Corp. Informix from Informix SQL Server from Microsoft Access from Microsoft Visual FoxPro from Microsoft	N/A

Appendix 2

Information Requests and Structure Query Language

Structured Query Language (SQL) was not an area of expertise for me at the start of the CEH project, but it ultimately became of great importance to me and to the project as a whole. By the end of the project, programmers were producing up to 20 SQL queries a day, most of which were quite complex.

SQL is a specialized language by which a programmer or user can query a database that is using to the Open Database Connectivity (ODBC) standard. It uses drivers to access databases in many different formats. In our case, we used EXCEL for working with the tables produced by an SQL query, and FoxPro as the data source.

Our work with SQL took place in three relatively distinct phases in chronological sequence.

In Phase 1 we were making simple queries because the information requests we received were simple. Most of them were queries at the case level. For example, listings such as in Figure 1.⁷

Figure 1. Listings I.

Query	From Table	Comments
Case Number	CASO	
Case Name	CASO	Given by the analyst
Certainty for the Case		Based on the violations on the case
Generic Date	CASO	A generic date for when the violations occurred. The Analyst chooses the date.
Violations	VIOL	All violations in the case.
Generic Place	CASO	A generic place where the violations occurred. The Analyst chooses the place.

In Phase 2 we started producing violation-based information, such as Figures 2-4, below.

Figure 2. Violations, I.

Query	From Table	Comments
Specific Violation	VIOL	Violation Name from The Violations Catalog
Identified Victims Count **	PATR_VICT	
Collective Victims Count **	PATR_VICT	Identified + Collective
Anonymous Victims Count **	PATR	Identified + Collective + Anonymous

** This means we had three columns in the table to count the victims.

With the availability of the results from these queries, the number of requests from members of the CEH team increased rapidly. Due to the organization of our Catalog, some of the killed were

⁷ Figure and table numbers in this appendix are sequential within the appendix and do not relate to the captions in the full paper.

Chapter Nine: The Guatemalan Commission for Historical Clarification

adding to the violation count (for example, Arbitrary Execution, Death as a Result of Violence). For that reason we made the grouping shown in Figure 3, below.

Figure 3. Violations, II.

Query	From Table	Comments
Generic Violation	VIOL	Violation Name from The Violations Catalog
Specific Violation	VIOL	Violation Name from The Violations Catalog
Identified Victims Count	PATR_VICT	
Collective Victims Count	PATR_VICT	Identified + Collective
Anonymous Victims Count	PATR	Identified + Collective + Anonymous

Note: By using Generic Violation, you can obtain all the dead.

The analyses and investigations of the CEH team came to the point where they needed to include particular keywords. A typical such request would be expressed in narrative form as “a listing of cases that include the keywords ‘Violence Against Children,’ ‘Territorial Movements and/or Religious Attacks’ in the year 1982 in the province of Huehuetenango, including ‘Responsible Groups.’” The response would be a level case listing such as in Figure 4, below.

Figure 4. Listings, II.

Query	From Table	Comments
Case Number	CASO	
Case Name	CASO	
Certainty for the Case		Based on the violations on the case
Generic Date	CASO	
Violations	VIOL	
Responsible Groups	VIOL_RSP	All the groups that participated in all the violations in the case.
Generic Place	CASO	

In Phase 3, we added additional information on violations. First we included information that is in a one-to-one relationship to the violations, such as the place where it happened. Thus, we added Department/Province to the table, and then information that was in more complex relationships, as shown in Figure 5.

Table 5. Additions to the tables

Option	Comments
Age (Categorized/Grouped)	Initially for only the Identified Victims, but later for Collective and Anonymous victims as well. We used the number -1 to identify the age for collective and anonymous victims, and mapped the violations to the general violation table.
Date (Month-Year/Trimester/Semester/Year)	This was to make it possible to divide the year in any way users wanted.
Place (Department and Cities)	At first only Departments, afterwards including cities where desired by the CEH teams.
Sex	Initially for only the Identified Victims, but later for Collective and Anonymous victims as well.
Forces Responsible (Institutional level Perpetrator)	This is where the violations count will be higher than that from the general (base) violations table because more than one perpetrator can participate in a violation. The first production represented all the forces responsible. The next version would group them to represent groups of interest to the CEH (i.e. URNG, ORPA, and all the other guerrilla groups will be grouped in the "Guerrilla Group." The same was done with the Government Institutions.

Of course the options were mixed, and we ended up with table analyses for many topics. These included: Age/Sex, Place/Language/Sex/Age, Place/Language, Place/Type of Victim, Place/Date, Place/Force Responsible, Place/Date/Force Responsible, and analyses for specific purposes: Massacres, Non-Massacre Analysis, Non-Guerrilla Analysis, Government Forces Analysis, Range of Victims Analysis, Non-Massacre Analysis, and so forth.

When half of the work was done, we decided to create table structures that would produce consistent information. We built structures for Identified, Collective and Anonymous Victims. These structures will lead to tables with information that will satisfy all the future requests we could conceive at the time. If we had done this earlier, it would have simplified our work. These structures were a success from both the user and programmer standpoints. With time, I became quite glad that I had learned how to use SQL, which enabled us to easily and quickly program queries to facilitate the work of the CEH teams.

Appendix 3

Security and Contingency Recommendations

In any field, at any time, we have problems of security and the need for contingency plans. Thus, we must be prepared to deal with these problems in a timely and correct manner. I make the following recommendations to achieve those results:

1. Avert data loss problems by having a tape backup on the server machine, and do automatic backups daily at midnight
2. Check backup tape every morning to see if everything was backed up properly
3. Keep a biweekly backup tape in a secure location outside the database building.
4. Do a backup and fire drill every month
5. Change users' logins and database passwords at least once every 45 days or when a database member leaves
6. Physically check the database area for unknown cables on the roof and floor
7. Close all the diskette operations on the client's machine except one, to be chosen by the DBA
8. Keep Internet access on a separate machine
9. In case of an Intranet, divide users by groups and give specific access to those groups
10. Remember that attacks to the database occur. Most of them will come from within the organization, so be selective in the personnel you choose and the members of the project you physically allow to enter the database
11. Use strong data encryption when moving information outside the database
12. Try to establish an internal audit trail data system within the database, to keep a log of updates and modifications
13. All systems should have a functional UPS, with sufficient uptime to finish the current operation and shut down the computer

References

Ball, Patrick. 1996. *Who Did What to Whom? Planning and implementing a Large Scale Human Rights Data Project*. Washington, DC: American Association for the Advancement of Science.

