EMS Unit Relational Database

Academic Project Report in the field of Business IT
First year - Bachelor of Arts in Management
Faculty of Economics and Social Sciences
University of Fribourg, Switzerland

Submitted by:

Autumn Totton
BA Management Candidate
Rue de Lausanne 75
CH-1700 Fribourg
Switzerland

Under the Supervision of:

Prof. Dr. Andreas Meier
Daniel Wismer
Department of Informatics
University of Fribourg
Switzerland

EMS UNIT RELATIONAL DATABASE – PROJECT REPORT

TABLE OF CONTENTS

1. INTRODUCTION

The Universal Postal Union (UPU) is the United Nations agency in charge of the development of postal services worldwide. The EMS Cooperative is a sub-organization of the UPU in Bern, which deals with the express mail services (EMS). The EMS Unit is the department within the UPU which serves as the administrative head of the EMS Cooperative. The Cooperative is composed of 126 member postal operators that offer an international express mail service. The EMS Cooperative aims to promote cooperation among the international service providers by carrying out a wide range of targeted activities and projects, all of which are managed by the EMS Unit.

1.1 PROBLEM ASSESSMENT

As an organization grows and develops, it is imperative that the technical solutions for managing information evolve accordingly. For many organizations, however, the immediate business requirements take priority, making it difficult to direct the necessary attention towards maintaining their internal data management systems. Often times, the resulting inflation of tables containing overlapping information leads to a state defined by Meier as “data chaos”, which can be remedied with the help of a “corporate-wide” data base model (department-wide in this case). [Meier 2001 p. 47]

Over the past few years, the EMS Cooperative has grown significantly – welcoming new members and launching additional projects – which has increased the workload for the administrative team at the EMS Unit. Meanwhile, the technical applications/solutions for managing the data generated and utilized by the EMS Unit have not evolved at the same pace. The lack of an evolving department-wide solution has forced many EMS Unit team members to create ad hoc data management files of their own to meet their short-term needs.

The EMS Unit has always used a Filemaker database to manage its primary list of contact points for general correspondence. As the dispatch of correspondence via email became more prevalent, the need arose for a more extensive contact list that could be divided into categories and easily fused into email messages. To satisfy this need, a semi-duplicate list of Microsoft Outlook contacts was created and has been maintained parallel to the Filemaker list ever since.

In addition to contacts, every major project managed by the EMS Unit generates and/or employs unique information relating to postal operators. These diverse sets of information are currently maintained and analyzed using various types of data management files (MS Excel, Word and Access files) with no provision for linking the data. The maintenance of information stored in multiple data-management files creates unnecessary work and is subject to inconsistencies due to human error.

The inefficient data management also complicates the utilization of available data. One of the primary administrative tasks involves the dispatch of letter, fax, and email correspondence to specific contact points representing Cooperative members. These contact points can generally be grouped into several project-related categories. In most cases a letter/fax/email will need to be sent to a specific group of contact points which can depend on an operator's project participation and/or other criteria related to another project. As all of this data is stored independently, it cannot be dynamically queried, which means that all distribution lists must be composed manually.

The following is an example of the process that must take place, under the current conditions, in order to send an individually addressed letter to a specific group of contact points:

1. a standardized letter needs to be dispatched to EMS Cooperative members that have tracking systems;
2. the list of operators with a tracking system must be extracted from an existing Excel spreadsheet;
3. the addresses for the operational contact points of the postal operators need to be manually selected from the Filemaker database and exported so that they can be merged into individually addressed letters.

1.2 GOALS AND OBJECTIVES

The practical purpose of this project is to create an Entity-Relationship (ER) data model and corresponding database capable of eventually managing all the data generated and used by the EMS Unit. The main goals are to minimize double data entry and maximize data utility. In doing so, many of the administrative procedures for corresponding with members, such as the one described above, can be simplified and streamlined.

The following definition of a database reinforces the idea that a department-wide database is indeed the tool required to solve the problem in the EMS Unit:

> **Def.:** "A database may be defined as a collection of interrelated data stored together without harmful or unnecessary redundancy to serve multiple applications; the data are stored so that they are independent of programs which use the data; a common and controlled approach is used in adding new data and in modifying and retrieving existing data within the database. The data is structured so as to provide a foundation for future application development." [Martin 1977, p.22]

1.3 Restrictions

The EMS Unit already has three professionally developed database applications to support specific projects that the Cooperative administers:

- One database is an SQL application containing information that is used specifically for an on-line publication. Some of this information is also relevant to other projects in the department.
- The EMS Unit has also recently launched a website that contains various dynamic modules and tables. The maintenance of user profiles by means of an on-line application conflicts with the internal data-management systems because many of the website users are also contact points. Managing the tables published on the website also presents a problem, because every time a list changes it needs to be updated on the website as well.
- The third database is an MS Access application handling many of the same data objects as in this project.

As these databases are already in use, they will have to be neglected in the scope of this project. Nevertheless, it is recommended that the overlap of data between all the EMS Unit's databases be taken into account in the "user requirements" for the development of subsequent versions of these databases. At a later stage, all of these databases should be synchronized in order to completely eliminate double data-entry.

1.4 Methodology

In an initial phase, the database user-requirements will be ascertained by means of: a) an existing working document; b) a questionnaire completed by each EMS Unit team member; and c) a review of the current automated systems. In the data-analysis process the critical entities, relationships, and attributes will be identified and compiled into an ER data model. The generalization hierarchies will also be identified and illustrated. The data-model will then be transformed into a relational database schema and undergo the normalization process (to 3NF) to ensure that redundancy is eliminated. The ER model will be implemented using Microsoft Access '97. Fictitious sample data will be entered in the tables for the purpose of illustration.

2. Requirements Analysis

2.1 Review of Pre-existing Document

A rough draft document outlining the EMS Unit's basic database requirements already exists. Essentially, they need a database that can link all of the contacts they maintain with the data associated with the EMS Cooperative projects. It needs to be sophisticated enough to eliminate the need for further reliance upon other data-management files. Some of the specific requirements included in this document were: the need to accommodate the language groups assigned to each contact point, the need for the database to be accessible to multiple read-only users at one time, and the need for a database that is flexible enough to be synchronized with the Unit's pre-existing databases (as described in section 1.3).

2.2 User Questionnaire

In order to obtain an overview of what project associated data is being maintained in the department, a short questionnaire was distributed to all EMS Unit staff. The questionnaire requested users to list all of the data-management files that they use on a regular basis. It also asked them to indicate the project that each file is associated with, its current technical capabilities (query or sort functions, filters, macros, etc.), and what functions this file would ideally be capable of performing.

Most of the team members sufficiently completed the first part of the questionnaire. They listed the files they use most frequently and the projects with which the files are associated. However, there was not much response to the questions regarding the technical functions that these files perform.

2.3 Review of Pre-existing Automated Systems

The lists of files provided by the team members allowed for an adequate review of the data-management files that are being used in the EMS Unit. Many of the files relevant to this project are Excel files. Most of which contain duplicated information, such as EMS Cooperative membership, debts-in-arrears, and ISO-codes. The master membership list is currently maintained in a Word document and there is no means of linking this information to the necessary excel files where the EMS Cooperative member's attributes are stored. The following files are a few of the most frequently utilized files by the secretariat:

- Filemaker database. It contains only the contact point information (includes mainly the same attributes as shown in the CONTACT POINT table) for the Political and Operational contact points who represent the EMS Cooperative member administrations.
- Outlook contacts list. This list also contains all political and operational contact points plus other contact points. It contains several categories into which the contact points can be classified for the purpose of dispatching project related correspondence. The categories are: "Delivery Agents", "EMS Board Members", "EMS Broadcasts", "EMS Link", "EMS Pay-for-Performance", "Rugby Implementers", "Rugby Users", "Useful EMS Contacts". This creates one of the biggest sources of double data-entry because all of the Filemaker records are duplicated in the Outlook contact list.
- Master lists of EMS Cooperative membership (MS Word doc) and members with debts-in-arrears (MS Excel spreadsheet). (currently being maintained in separate files) This information is frequently duplicated in other data-management files as it is associated with almost all projects.

3. DATA ANALYSIS

3.1 DATA MODELLING

The data modelling process consists of identifying and describing the "real-world" information requirements for the data management system, specifying the data to be maintained by the system, and specifying the data structures to be used for data storage.

A simplified version of the "real-world" information requirements can be illustrated in terms of entities and relationships in an ER model (Chen 1976). "The entity-relationship approach to data modelling has two primary objectives: first to enhance emphasis on inter-entity relationships and second to separate the database design process into two phases:

1. Record, in an ER model, the entities and inter-entity relationships required by the user. This phase and its resulting model should be independent of the database management system (DBMS) tool that is to be used for realizing the database.
2. Translate the ER model to a database schema supported by the DBMS to be used for implementation." [Nordbotten 03]

The ER diagram will be illustrated at the end of section 3.1.4., following the definition of the relevant entities, attributes, and relationships. The diagram does not include the attributes of the entities. These will be introduced in the database schema described and illustrated in section 3.2.

3.1.1 THE "REAL-WORLD" INFORMATION REQUIREMENTS

Following is a list of the relevant "real-world" descriptions of the information that will be included in the database:

1. There are 233 postal operators, from which about 172 offer an EMS Service.
2. Postal operators can be classified into regions and can be categorized as EMS Cooperative members, Non-EMS Cooperative members, and Delivery Agents.
3. Operators offering an EMS service may or may not have an entry in the EMS Operational Guide.
4. Of those operators offering an EMS service, 126 are EMS Cooperative members.
5. The contact points with which the EMS Unit corresponds can be related to one of several types of organizations—postal operators, restricted unions, and external consultants.
6. Cooperative members can be participants of the Pay-for-Performance and/or Rugby programs.
7. Some Cooperative members have validated delivery standards which are versioned.
8. Information on traffic volumes and tracking information is collected from all postal administrations offering an EMS service.

9. The contribution class of EMS Cooperative members is based on their respective traffic volumes.
10. The contribution class of a member determines their voting rights.

3.1.2 Entity and Attribute Definition

Based on the information requirements, the data objects can be identified and classified as entities or attributes. But first these terms should be defined in the context of data modelling. An entity can be defined as follows:

> **Def.:** "Any distinguishable person, place, thing, event, or concept, about which information is kept". [Bruc 1992]
>
> **Def.:** "A thing which can be distinctly identified". [Chen 1976]

An attribute can be characterized by the following definition:

> **Def.:** "*Attributes* are data objects that either identify or describe entities. Attributes that identify entities are called *key attributes*. Attributes that describe an entity are called *non-key attributes*." [This and the above definitions were taken from ITS at UTA - Identifying Data Objects and Relationships]

In the first step, the following data objects were identified and classified as entities or attributes. The key attributes are highlighted:

Entity	Attributes
Organization	***Organization_ID,*** Organization_Name_EN, Organization_Name_FR, Description, Organization_Type
Postal Operator	***ISO_Code_3***, Organization_ID, ISO_Code_2, Region, Tracking_Status, Authorization_to_IPC, Inward_Traffic_Volume, Traffic_Volume_Range Outward_Traffic_Volume, Inward_Only, Inward_and_Outward, Other_Coverage, Operator_Type
Restricted Union	***Organization_ID***
External Consultant	***Organization_ID,*** Project_Association
EMS Cooperative Member	***Member_ID,*** ISO_Code_3, Membership_Status, Debts-in-Arrears, Date_Joined, Date_Left, Pay-for-Performance_Status, Rugby_User, Validation_Status, Version, Date_Implemented
Delivery Agent	***Deliver_Agent_ID,*** ISO_Code_3, Country_Covered, Validation_Status, Version, Date_Implemented
Non-EMS Cooperative Member	***ISO_Code_3***
Contact Point	***Contact_ID***, Organization_ID, Last_Name, First_Name, Polite_Address_Lang1, Polite_Address_Lang2, Title, Address, Country, Telephone, Fax, Email, Language_1, Language_2, Operational, Political, Board, Pay-for-Performance, Rugby, EMS_Link, Operational_Guide_Username, Operational_Guide_Password
Contribution Class	***Traffic_Volume_Range,*** Contribution_Class, Voting Rights, Fixed_Fee

Entity	Attributes
EMS Operational Guide Entry	***Entry_name***, ISO_Code_3, Status, Modifications, Date_Modified, Credit_Q1Y03, Credit_Q2Y03, Credit_Q3Y03, Credit_Q4Y03
Circular	***Circular_Number***, Operator_Association, Subject

3.1.3 RELATIONSHIP DEFINITION

In the next step, the relationships between the entities will be examined. The relationship between two entity objects is described by the following definition:

> **Def.:** "A relationship is a way of tying objects together to get new information that exists apart from the particular objects". [Celko 1999, p.15]

A relationship between two entities is composed of two associations. An association can be defined as the degree of the relationship in the direction of the opposite entity. The four types of associations are shown in illustration 1. The degree of a relationship between two entities E1 and E2 can be expressed as the combination of the associations: Degree := (Type E1 towards E2 ; Type E2 towards E1). [Meier 2001, p.18-19]

Association Types :

Type 1: "exactly one"

Type c: "none or one"

Type m: "numerous"

Type mc: "none, one or numerous"

ILLUSTRATION 1: ASSOCIATION TYPES

The enitity-entity matrix in illustration 2 shows the associations between the entities identified in section 3.1.2. (The association from the entity shown on the left to the entity shown on the top is the first association when read left to right.) Here are some examples of how these associations can be expressed: "An Organization is associated with one, none, or many Contact Points" and "A Contact Point is associated with exactly one Organization"; "A Postal Operator is associated with one or no EMS Cooperative Member" and "An EMS Cooperative Member is associated with exactly one Postal Operator".

	Organization	Postal Operator	Restricted Union	External Consultant	EMS Cooperative Member	Delivery Agent	Non-EMS Cooperative Member	Contact Point	Contribution Class	EMS Operational Guide Entry	Circular
Organization		c;1	c;1	c;1				mc;1			mc;mc
Postal Operator	1;c				c;1	c;1	c;1		1;mc	c;1	
Restricted Union	1;c										
External Consultant	1;c										
EMS Cooperative Member		1;c									
Delivery Agent		1;c									
Non-EMS Cooperative Member		1;c									
Contact Point	1;mc										
Contribution Class		mc;1									
EMS Operational Guide Entry		1;c									
Circular	mc;mc										

ILLUSTRATION 2: ENTITY-ENTITY MATRIX

3.1.4 GENERALIZATION HIERARCHIES

In section 3.1.2 the entities were characterized by their similarities using attributes. The generalization process makes it possible to categorize entities by both their similarities and differences.

> **Def.:** "A generalization hierarchy is a structured grouping of entities that share common attributes. It is a powerful and widely used method for representing common characteristics among entities while preserving their differences." [ITS at UTA – Generalizations Hierarchies]

An example of a generalization hierarchy is the Organization entity, because it is a generalization of the subtypes: External Consultant, Restricted Union, and Postal Operator. The attributes that these entities have in common are the attributes of the Organization entity. The differences are the attributes of each individual subtype. This is an example of a disjoint generalization hierarchy because each Organization must belong to exactly one of these subtypes (see illustration 3). The Postal Operators entity forms a nested generalization hierarchy composed of the following subtypes: EMS Cooperative Members, Non-EMS Cooperative

Members, and Delivery Agents. This is also a disjoint generalization hierarchy as a Postal Operator must belong to at least one of these subtypes.

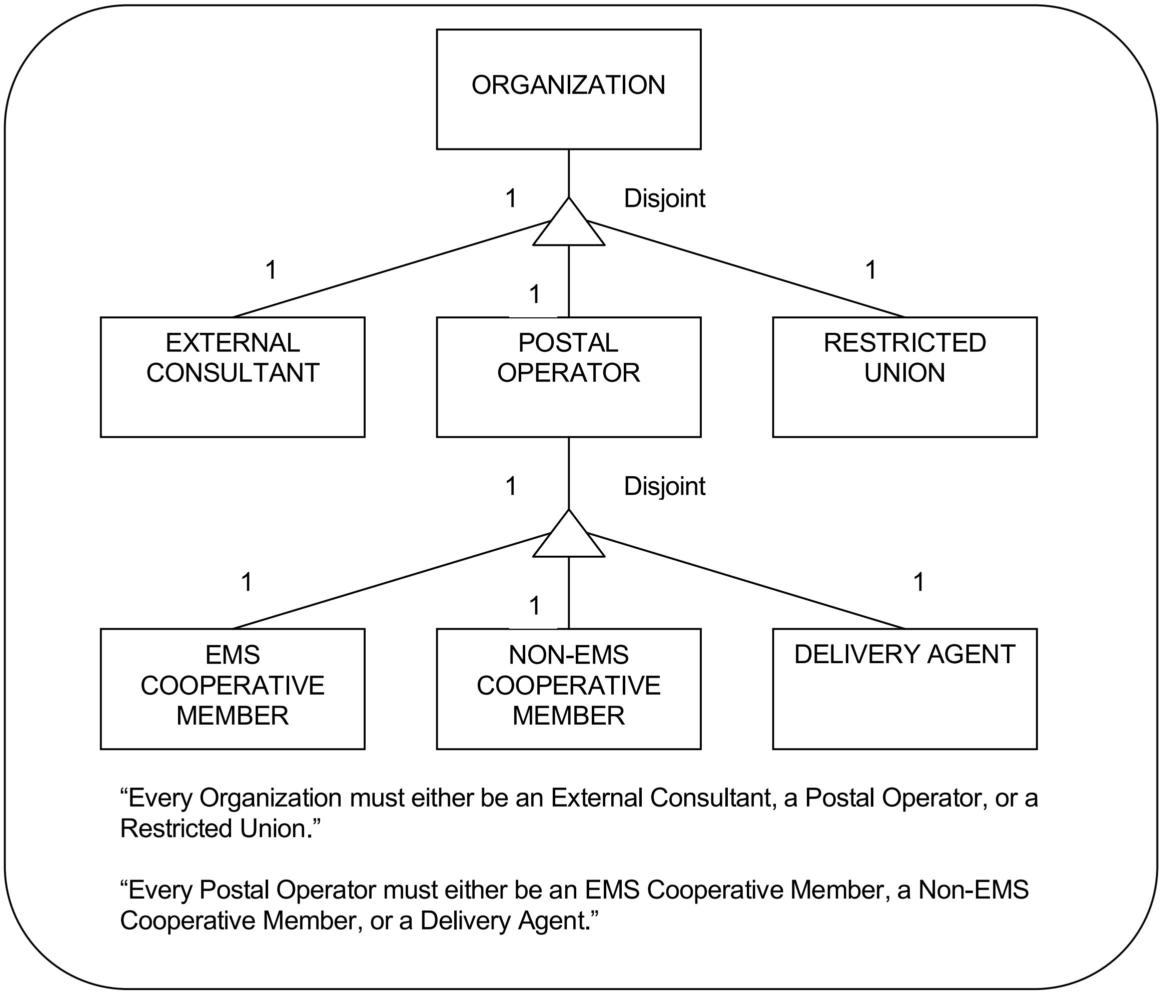


ILLUSTRATION 3: ORGANIZATION AND POSTAL OPERATOR GENERALIZATION HIERARCHIES

An example of another generalization hierarchy which might be useful in the future is the contact point entity. It could be considered a generalization of the various contact point lists that the EMS Unit maintains, i.e. Operational, Political, Pay-for-Performance, Rugby Implementers, Board, and Link. At this point, these subtypes cannot be distinguished because their differences are not recorded. For that reason, the generalization hierarchy has not been included here.

3.1.5 ENTITY-RELATIONSHIP MODEL

The ER model is a diagram that depicts the entities and the relationships they have to one another. The entities are represented by rectangular boxes and the relationships are symbolized by the diamond shapes.

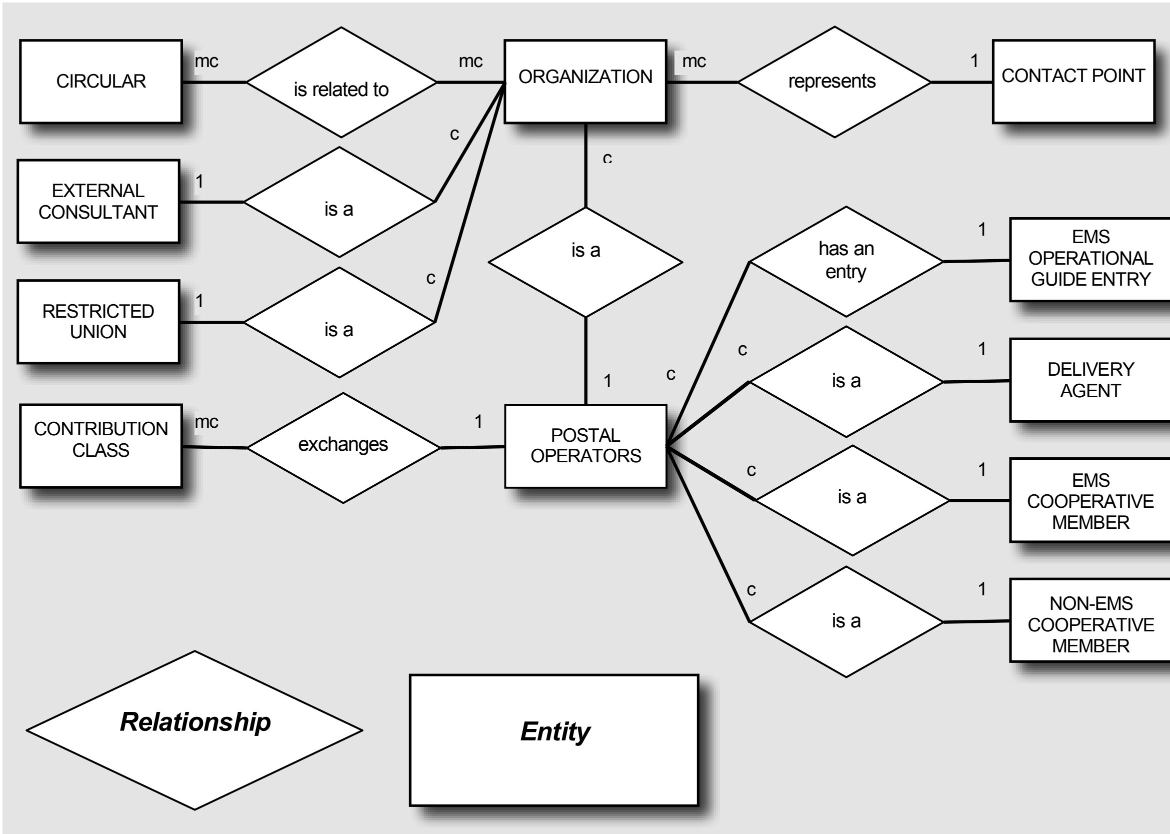


ILLUSTRATION 4: ENTITY-RELATIONSHIP MODEL

3.2 RELATIONAL DATABASE SCHEMA

The database schema essentially describes the database, i.e. the specification of the data-structure and integrity conditions. It includes the definitions for all the tables, attributes, primary keys and foreign keys which will play a fundamental role in enforcing data integrity. [Meier 2001, p.23]

Prior to transforming the Entity-Relationship Model into a relational database schema, it must be determined which entity and relationship objects need to have their own tables. This can be done by applying the following set of rules. [Meier 2001, p.23-33]

As rules 2-5 indicate, relationship objects can be depicted in the relational database scheme in two ways. In some cases they must have their own table, in other cases the relationship can be represented with use of the primary key from one table as the foreign key in the other table. The following definitions of primary key and foreign key are an expansion of the attribute definition [ITS at UTA – Primary and Foreign Keys page]:

Def: "The *primary key* is an attribute or a set of attributes that uniquely identify a specific instance of an entity."

Def: “A *foreign key* is an attribute that completes a relationship by identifying the parent entity.”

There are two possibilities for assigning keys. The first option is to elect an “intelligent” key which has some meaning in itself and occurs naturally. For an attribute (or combination of attributes) to qualify as the primary key for an entity, it must have the following properties:

- it must have a non-null value for each instance of the entity;
- the value must be unique for each instance of an entity;
- the values must not change or become null during the life of each entity instance.

The second option is to create a “surrogate” key that fulfills the primary key requirements. A surrogate key is an artificial key created for the purpose of uniquely identifying a record. There are arguments for and against both intelligent and surrogate keys. [Celko 1999, p.247-267]

Because of the nature of the data in this case, the use of surrogate keys has been adopted in cases where an appropriate natural key is not available. For example, the primary key “Organization_ID” is an artificial key because there is no natural occuring attribute that meets the primary key requirements listed above. An example of an “intelligent” key is ISO_Code_3 which appears as the primary key of the entities Postal Operator, Delivery Agent, and Non-Cooperative Member. Each Postal Operator can be identified by this 3-character code assigned by the International Standardization Organization (ISO).

Rule 1 (entity objects)

“Every entity object must be defined as its own table with a unique primary key”. This means that all of the entities identified in the table in section 3.1.2 will need to have their own tables with a unique identifier as the primary key.

Rule 2 (relationship objects)

“Every relationship can be defined as its own table, with the primary keys of the related entities being the foreign keys in this table”. According to this rule, it is optional to express every relationship with an independent table. The following 2 rules will provide more specific instructions on how to represent relationship objects in tables, based on the degree of the relationship between two entities.

Rule 3 (complex-complex relationships)

“Every complex-complex relationship [(m,m); (m,mc); (mc,m); (mc,mc)] must be defined as its own table, with at least the primary keys of the related entities being the foreign keys in this table”. There is one complex-complex relationship that needs to be represented in a separate table; that being the relationship between ORGANIZATION and CIRCULARS. As shown in illustration 5 at the end of this section, Circular_Number and Organization_ID are the foreign keys that combine to form the primary key of the CIRCULAR_ORGANIZATION table.

Rule 4 (simple-complex relationships)

"A simple-complex relationship [association type "1" or "c" with type "m" or "mc"] can be expressed through the tables of the related entities, without its own table". It is optional to express simple-complex relationships in their own tables. As an alternative, one of the primary keys needs to be contained in the other related table as a foreign key. The simple-complex relationships in this data model will be represented using the foreign key alternative, as shown in illustration 5. In general, the primary key from the table with the lower degree association will be included in the other table as a foreign key. [Meier 2001, p.30] For example, the relationship between CONTACT POINT and ORGANIZATION will be depicted by using the primary key from ORGANIZATION (Organization_ID) as a foreign key in the CONTACT POINT table. As the association from CONTACT POINT to ORGANIZATION is type "1", each contact point will be assigned to "exactly one" organization.

Rule 5 (simple-simple relationships)

"A simple-simple relationship can be expressed through the tables of the related entities, without its own table". This is similar to rule 4 for simple-complex relationships. The use of a foreign key relationship is also the alternative to creating a separate table in this case. The simple-simple relationship between POSTAL OPERATOR and EMS OPERATIONAL GUIDE ENTRY will be created by using the primary key from POSTAL OPERATOR (ISO_Code_3) as a foreign key in the EMS OPERATIONAL GUIDE table.

Rule 6 (Generalizations)

"Every subtype of a generalization hierarchy must have its own table with the primary key from the supertype as the primary key of the subtype table". This means that EXTERNAL CONSULTANT, RESTRICTED UNION, and POSTAL OPERATOR (subtypes of ORGANIZATION) have their own tables. The EXTERNAL CONSULTANT and RESTRICTED UNION tables have the same primary key as the ORGANIZATION table. The POSTAL OPERATOR has been assigned a different primary key (ISO_Code_3) in order to allow for the creation of a one-to-many relationship between ORGANIZATION and POSTAL OPERATOR in the future. Organization_ID as a foreign key in the POSTAL OPERATOR table sustains the relationship between these two entities.

The same applies to the nested POSTAL OPERATOR hierarchy. The EMS COOPERATIVE MEMBER subtype has a different primary key than the supertype. The ISO_Code_3 key is a foreign key in the EMS COOPERATIVE MEMBER table to maintain the relationship between the two entites.

As discussed in section 3.1.4, there are other generalization hierarchies that could be created in the future. Any new generalization hierarchies should be added to the database schema as prescribed in rule 6.

Illustration 5 contains all of the tables that need to be included in the database and their respective attributes (according to the rules in section 3.2).

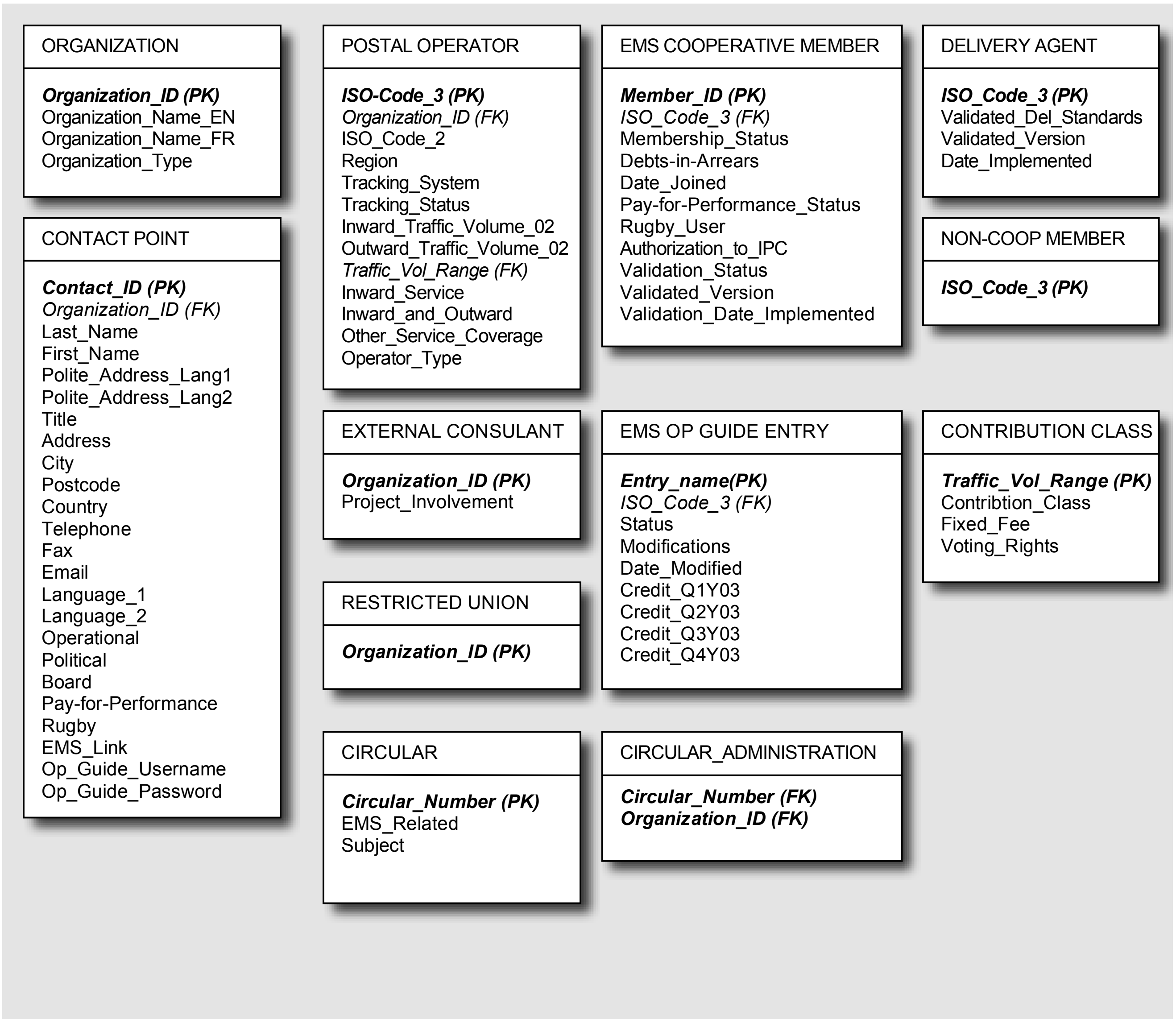


ILLUSTRATION 5 : RELATIONAL DATABASE SCHEMA

3.3 NORMALIZATION

The goal of the normalization process is to eliminate redundant data within the tables and to ensure that related sets of data can be maintained efficiently throughout the database.

Def.: "An attribute of a table is *redundant* if individual values of this attribute can be excluded without losing any information." (Meier 2001, p.34)

Redundant data in tables leads to the occurrence of update anomalies. Which means to change, add, or delete a value of a redundant attribute, it must be done everywhere in the database where that attribute is repeated.

An acceptable state of normalization can be attained with the help of normal forms 1 through 3. (Meier 2001, p.36-40) These normal forms, defined by E.F. Codd, set specific criteria which a table must meet in order to be considered normalized in that form. In the next section, these forms will be described and applied to the tables in the database schema.

First Normal Form (1NF)

"A table is in the first normal form if all values of the attributes are atomic". In other words, the columns contain no repeating values. By definition, all relational tables are in first normal form, which means that the tables defined in the database schema above are already in first normal form.

Second Normal Form (2NF)

"A table is in the second normal form if it is in the first normal form and if all non-key attributes are functionally dependant upon the primary key."

Def.: "An attribute B is *functional dependent* upon attribute A, if for every value of A exactly one value of B exists".

For a table to be in second normal form, every value of the primary key corresponds to exactly one value of the non-key attributes. Tables with a composite primary key, the same applies, except all non-key attributes must be functionally dependent upon the entire primary key, i.e. upon the combination of the keys.

Examination of the tables reveals that all of the tables are in the second normal form. The CIRCULAR_ADMINISTRATION table does not contain any non-key attributes at this point. Should it be necessary to add attributes in the future, it is important that they are functionally dependant upon the composite primary key.

Third Normal Form (3NF)

"A table is in the third normal form if it is the second normal form and if none of the non-key attributes are transitively dependant upon one of the key attributes".

Def.: "An attribute C is *transitively dependent* upon A, in the case that B is functionally dependent upon A, C is functionally dependent upon B, and at the same time A is not functionally dependent upon B".

All of the tables in the database schema are also in third normal form as there is no transitive dependency in the tables.

The fact that none of the tables needed further adjustment in the normalization process confirms the assertion made by Meier: If the ER model is clearly defined and the rules for creating the database schema are strictly followed, the normal forms will be achieved. [Meier 2001, p.36]

4 Implementation as an Access Database

The chosen application for implementing this relational database is Microsoft Access. The EMS Unit currently uses the Windows '98 operating system and carries out most of its administrative tasks using Microsoft Office '97 applications. The migration of the primary "contact points" database from Filemaker to a Microsoft application will improve compatibility with the other applications utilized in the department. Letter and fax correspondence created in MS Word can be merged directly from Access, as opposed to having to use a merge file exported from Filemaker. The need for an additional list of contacts in Outlook will be eliminated. It will also be possible to easily analyze Access data using Excel. Furthermore, an Access database can be accessed (read-only) by multiple users at the same time and it is potentially compatible with 2 of the EMS Unit's pre-existing databases described in section 1.3.

The implementation in Access is relatively straightforward, because the database schema has already been defined and can basically be converted into a series of access tables with corresponding relationships. The challenge lies in adding the necessary elements to make the database user-friendly and comprehensible for the users. If one of the users were to open the POSTAL OPERATOR table, for example, they would be shocked to find that the data they use exclusively for their project would be mixed in amongst 12 other columns of irrelevant data. In the future, to avoid shock and confusion, it will be necessary to create various queries and forms that will allow the users to view only the information that they need.

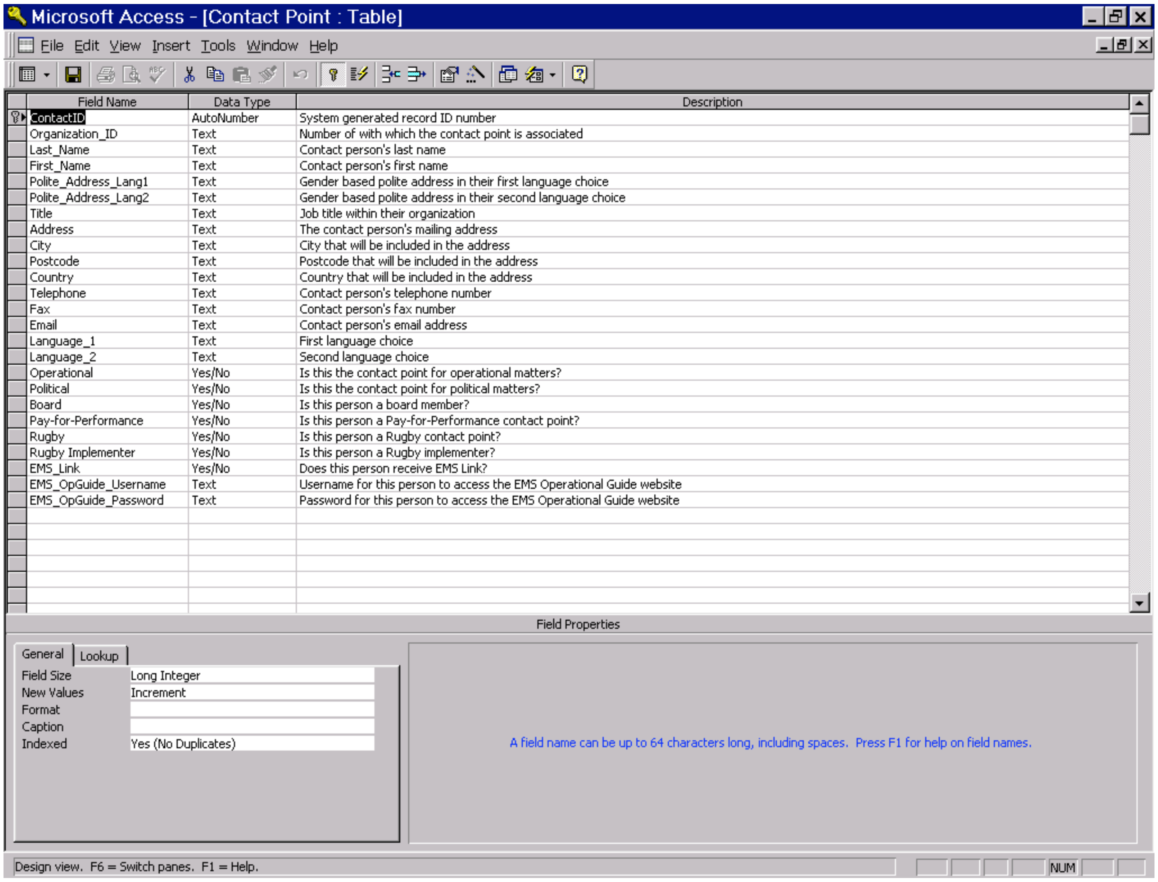
4.1 Table and Key Definition

The first step using MS Access is to create the tables and define their primary keys. Every table illustrated in the relational database schema represents an independent table. The attributes listed under each table will correspond to a field/column in the table. In design view mode of an Access table, the field properties can be set to define the characteristics of the field and determine how data is entered and displayed in this field. [Microsoft Press, p.81]

Illustration 6 is an example of what one of the Access tables looks like in design view mode. The first column is for the field name, the second column controls the data type setting, and the third column gives a brief description of the field.

- The *field names* correspond to those shown in the database schema. The spaces in the field names are filled with underscores.
- "The *data type* establishes the kind of data that the field can accept— text, numbers, dates and times, yes-or-no data, monetary amounts—or whether it can accept user-entered data at all." [Microsoft Press, p.81] The data types used in the tables throughout the database are limited to "Text", "AutoNumber", and "Yes/No". Although there are other options that would be suitable for the data types in certain fields, most fields have been left as "Text" to avoid restrictions that would not accommodate data anomalies. As the format in which data is entered becomes more standardized, the field data type settings can be adjusted to maintain that standardization.
- The *field description* is what appears in the status bar at the bottom of the Access window when that field in the table (or related query/form) is active. It gives the user a better idea of what information is being stored in that field.

The primary key of each table is indicated by the key symbol next to the field. In this example, the ContactID field is the primary key.



Microsoft Access - [Contact Point : Table]

Field Name	Data Type	Description
ContactID	AutoNumber	System generated record ID number
Organization_ID	Text	Number of with which the contact point is associated
Last_Name	Text	Contact person's last name
First_Name	Text	Contact person's first name
Polite_Address_Lang1	Text	Gender based polite address in their first language choice
Polite_Address_Lang2	Text	Gender based polite address in their second language choice
Title	Text	Job title within their organization
Address	Text	The contact person's mailing address
City	Text	City that will be included in the address
Postcode	Text	Postcode that will be included in the address
Country	Text	Country that will be included in the address
Telephone	Text	Contact person's telephone number
Fax	Text	Contact person's fax number
Email	Text	Contact person's email address
Language_1	Text	First language choice
Language_2	Text	Second language choice
Operational	Yes/No	Is this the contact point for operational matters?
Political	Yes/No	Is this the contact point for political matters?
Board	Yes/No	Is this person a board member?
Pay-for-Performance	Yes/No	Is this person a Pay-for-Performance contact point?
Rugby	Yes/No	Is this person a Rugby contact point?
Rugby Implementer	Yes/No	Is this person a Rugby implementer?
EMS_Link	Yes/No	Does this person receive EMS Link?
EMS_OpGuide_Username	Text	Username for this person to access the EMS Operational Guide website
EMS_OpGuide_Password	Text	Password for this person to access the EMS Operational Guide website

Field Properties

General | Lookup

Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Design view. F6 = Switch panes. F1 = Help.

ILLUSTRATION 6: CONTACT POINT TABLE IN DESIGN VIEW

4.2 RELATIONSHIPS

The next step is to build the relationships between the tables. This can be done with help of the Entity-Entity Matrix (illustration 2 in section 3.1.3). In Access, however, the relationships are presented in the opposite direction as they are in the Matrix. For example, relationship between CONTRIBUTION CLASS and POSTAL OPERATOR is shown as m;1 in the Matrix, but in Access the many is on the end of POSTAL OPERATOR and the 1 is on the CONTRIBUTION CLASS end (see illustration 7: Relationships View in Access).

When creating a relationship, Access allows the referential integrity between the related tables to be enforced. Referential integrity is the system of rules that Access uses to be sure that relationships between records in related tables are valid, and that changes made in one table are properly applied to a related table. Changes made to the primary table are automatically applied to the related tables through a process called cascading. [Microsoft Press 1999] For example, if a record in the ORGANIZATION table is deleted or changed all of the corresponding record(s) in the CONTACT POINT table will be updated or deleted as well.

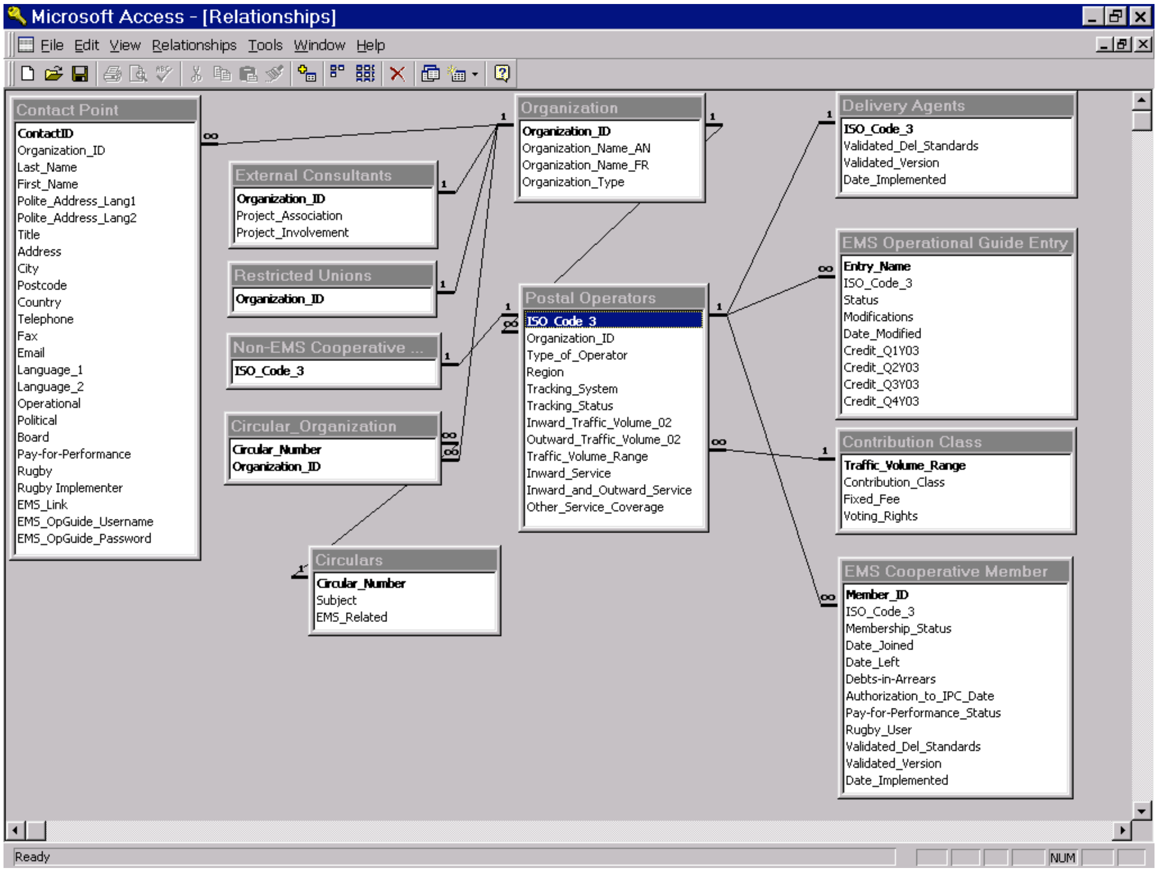


ILLUSTRATION 7: RELATIONSHIPS VIEW IN ACCESS

4.3 QUERIES

Queries allow the user to call up specific information of interest. They can draw the information from one table or a combination of related tables and preserve that view permanently. With the help of a query, the process for sending an individually addressed letter to a specific group of contact points as described in section 1.1 can be simplified. In this case, the indirect relationship between the CONTACT POINT and the POSTAL OPERATOR tables makes it possible to query the database for the Cooperative member operators that have tracking systems. Simultaneously, the operational contact point information for those operators can also be returned. The Access window in query design mode for this particular query is shown in illustration 8. The results of the query are shown in datasheet view mode in illustration 9.

Queries only need to be created and saved once. Every time the query is recalled it is run on the up-to-date data stored in the tables upon which it is based. After the query is created, the process of sending out a letter only requires using the MS Office Link to merge the data from the query with the MS Word document where the letter is stored. This process is certainly much more efficient than that described in section 1.1.

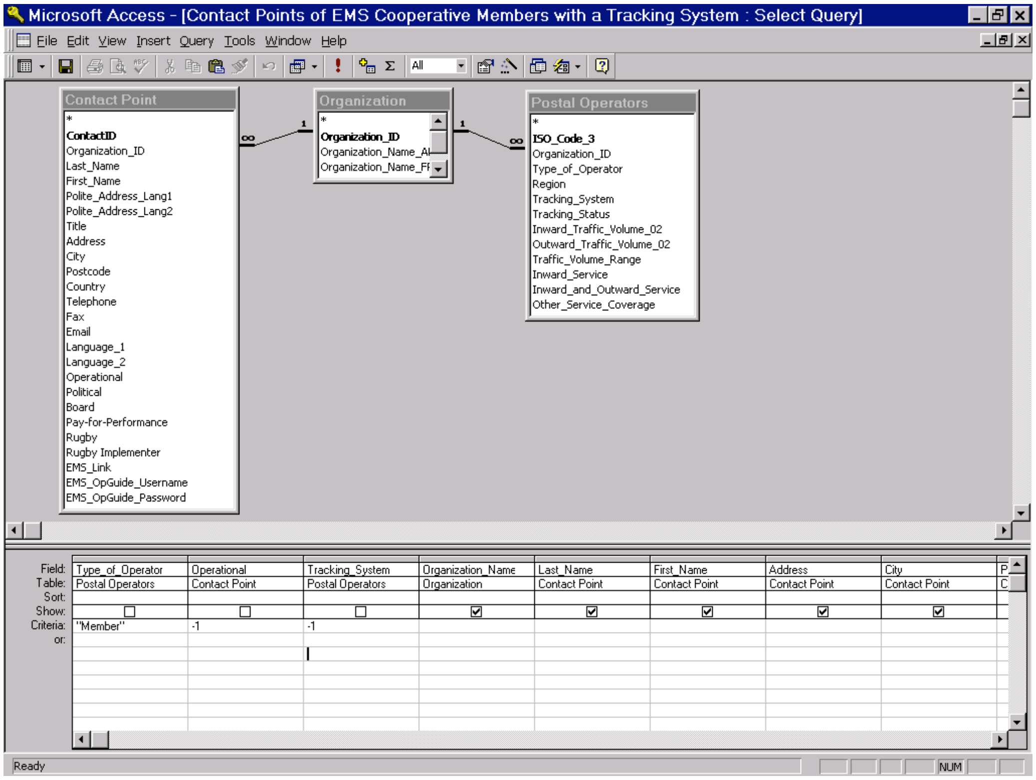


ILLUSTRATION 8: QUERY IN DESIGN VIEW

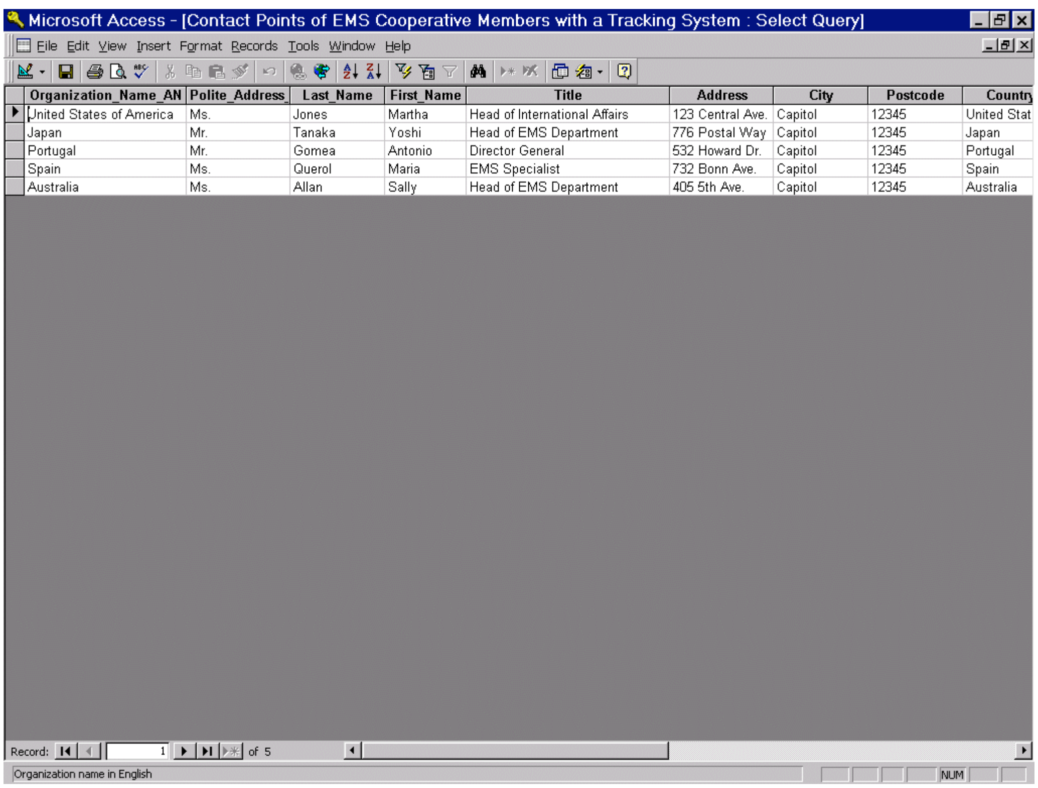


ILLUSTRATION 9: QUERY RESULTS IN DATASHEET VIEW

4.4 FORMS

Access forms can be used to input data and to enhance the visual impression of the input interface. Given that groups of related data are stored in different tables, it is difficult to enter data directly into the tables. It would require having to go into each table where the particular attributes are stored and enter the data individually. This kind of inconvenience can be eliminated with the help of a form which is based on a query that combines the information from multiple tables. Illustration 10 shows a form based on a query that combines the CONTACT POINT and ORGANIZATION tables.

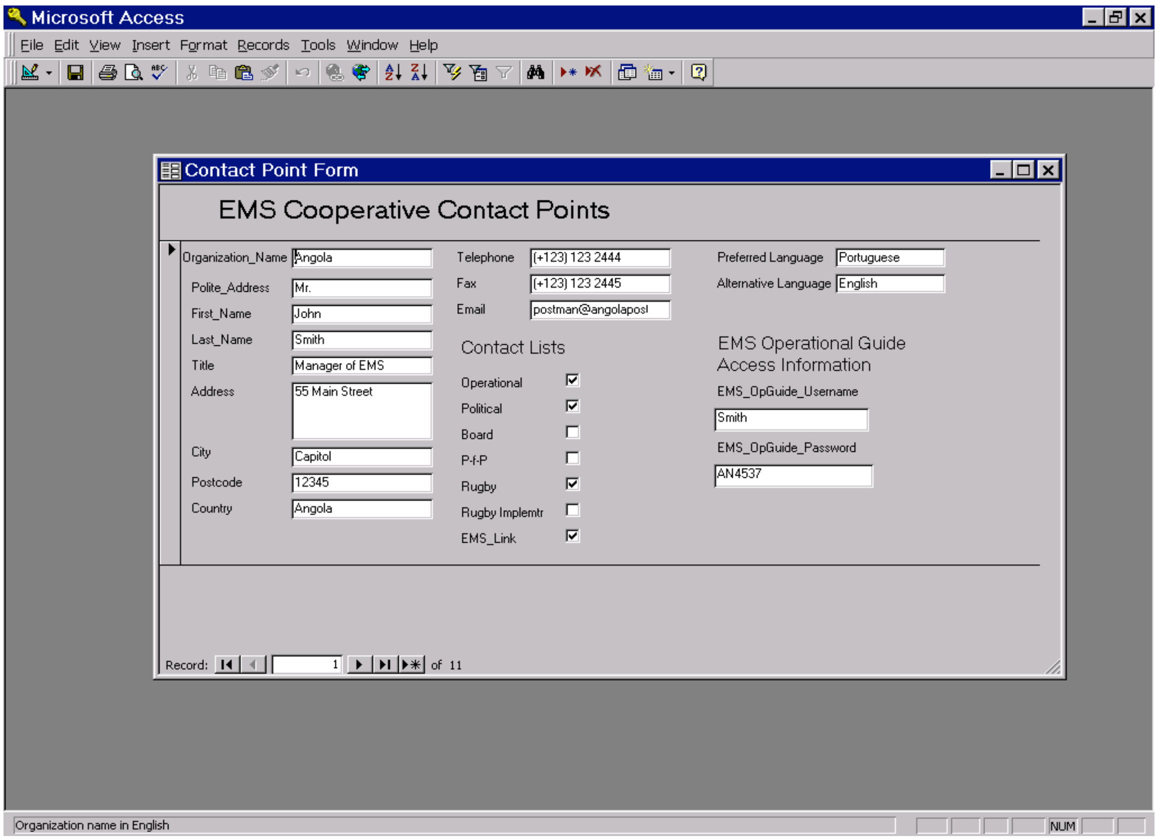


ILLUSTRATION 10: CONTACT POINT FORM IN ACCESS

This database really requires a user interface that would cover all aspect of the information it contains. An idea would be to have one Access form separated into tabs where the different project-associated groups of information are presented. This is an issue to be considered for future developments.

5 Conclusion

5.1 Recommendations

The solution to the EMS Unit's data management problem that I have presented in this document is one of many potential solutions. Although this may not be the optimal solution, this database model could serve as the basis for the future development of an optimized solution.

In my opinion, there are three alternatives as to how to implement a department-wide data management system in the EMS Unit. The first one would be to actually implement what I've proposed here and deal with the deficiencies as they present themselves along the way. The second alternative would be take what I've proposed here and pass it on to the organization's internal database specialists (professionals) for revision (or scrapping) and implementation. The third option would be to pass this on to the database specialists and form a team of users in the EMS Unit who participate in the implementation. Let's look at the advantages and drawbacks of each of these alternatives.

1. If the data model that I've proposed here were to be slowly implemented without involving any professionals, the worst that could happen is that the current situation wouldn't change. In the best case, it would partially solve the problem and the EMS Unit team members, who have a knowledge of Access, could modify the database as necessary.
2. The second option is to turn it over to the professionals. Let's say they decide to implement it using a different relational database application and design us a snazzy user-interface. Suddenly, the solution becomes too high-tech for the EMS Unit to maintain independently, which would make us reliant upon the professional developers to maintain the system as our requirements change in the future. This also means that the implementation would depend on the availability of the professionals and, as the EMS Unit is not the only "client", they may have other priorities.
3. The third option would be to have the professionals revise the data model and then to establish a group of EMS Unit team members who participate in the implementation. After a professional review, we could be confident that there are no grave oversights or exclusions in what I've proposed. And at the same time, the group could ensure that the solution is still something that could be managed and developed with the technical knowledge available in the EMS Unit.

In my opinion, the third option would be the most effective in the long-run. Although it would require the time commitment of selected EMS Unit team members during the implementation phase, it would allow for the automation of many of the administrative tasks that are currently very time-consuming and inefficient. Furthermore, it would keep the database on a technical level that would allow it to be maintained by EMS Unit staff in the future. Introducing the system in the department in collaboration with the professionals, would give the users enough confidence in the system to give up their current solutions.

5.2 Final Remarks

It is very rewarding to apply the theory behind business information technology (IT) to improve data management. Relational databases have tremendous potential for simplifying procedures, minimizing redundancy and maximizing efficiency. My attempt at creating a database of this magnitude was definitely challenging. The fact that I have a very good knowledge of how the information in the EMS Unit is used helped me a great deal in defining and transforming the “real-world” situation into a data model. At the same time, this made me very aware of the anomalies that lurk in the background, but remain out of my control on account of my limited knowledge of database design. As I progressed, I realized that the scope of this project is virtually endless. There is so much information that could still be added to the database, but for the purpose of my academic project I was forced to draw the line at some point. I tried to include all of the essentials and neglect only the parts that could easily be added at a later stage as needed. If nothing else, I hope my work will serve as a foundation for future development of a department-wide database in the EMS Unit. In conclusion, I’m including an appropriate and reassuring quote that I came across in Celko’s book:

> *“Perfection is finally attained not when there is no longer anything to add but when there is no l longer anything to take away.” —Antoine de Saint Exupery*

6 ILLUSTRATION INDEX

7 SOURCE INDEX

- [Meier 2001] Meier, Andreas: Relationale Datenbanken – Leitfaden für die Praxis. Springer-Verlag, 4. Auflage, Berlin 2001.
- [Martin 1977] Martin, James: Computer Data-Base Organization. Prentice-Hall, 2nd Edition, New Jersey 1997.
- [Celko 1999] Celko, Joe: Data & Databases: Concepts in Practice. Morgan Kaufmann, San Francisco 1999.
- [Microsoft Press 1999] Catapult, Inc: Step-by-Step Microsoft Access 2000. Microsoft Press, Redmond, Washington 1999.
- [Nordbotten 03] Nordbotten: Modelling Complex and Multimedia Data. http://nordbotten.ifi.uib.no/ADM/ADM_text/ADM_data_modelling.htm, accessed 21st Oct., 2003.
- [ITS at UTA –…] Information Technology Services at The University of Texas at Austin – Introduction to Data Modelling. http://www.utexas.edu/its/windows/database/datamodeling/…, accessed 21st Oct. 2003.