

DATA ANALYSIS

Commercial systems are information systems. Information is composed of data and without data the organisation cannot function. Therefore, a thorough understanding of the data requirements will be a pre-requisite for good systems design.

Data exists in a system in the form of Data Structures or Data Groups. A data flow may consist of a single data element or more likely a group of related data elements, a Data Structure.

A data element is a named item of data which cannot be decomposed or broken down further.

eg DATA ELEMENTS

EMPLOYEE-NUMBER
EMPLOYEE-SURNAME
ADDRESS-STREET-NAME
HOUSE-NUMBER

These may make up the DATA STRUCTURE "EMPLOYEE".

In order to fully understand how a system uses data, various techniques can be used to provide us with a Logical Data Structure (LDS) of the systems data.

NORMALISATION

The first view of data that an Analyst or Database Designer may have will be a combination of both the logical and physical data. In this form the data will include duplications, redundancies and badly organised structures. Some structures may be large and consist of various logical entities or structures.

Normalisation (a methodology developed by E.F. Codd in 1972) of these data structures will:

- * Remove repeating groups from structures, thus making them into new independent structures.
- * Examine the keys which are used to identify and access the groups, and ensure that every structure has a unique key.
- * Remove any items that can be derived or calculated.

Any data structure or group of data elements that regularly appear together can be normalised, whether it is a paper form, an input or output screen, a filing cabinet or magnetic store. To ensure a true logical view of the system is produced all data structures should be normalised.

Normalisation is performed in three stages, after which the data is said to be in '3NF' (Third Normal Form). Boyce-Codd extended this methodology and included procedures to continue to Fourth (4NF) and Fifth (5NF) normal forms. Here we are only concerned with normalising to the 3rd Normal form (3NF).

When the 3NF structures are produced they are then 'consolidated', a process of comparing key sets for an exact match and combining the data groups.

The end product is a streamlined set of structures with keys. They form the basis of the physical data structures in the new system, and if the process has been carried out correctly, the system using the database or files will have the optimum organisation of data, which best meets the requirements of the system, resulting in fast response times, ease of use etc. The final 3NF structure form the basis for these file designs.

THE THREE STAGES OF NORMALISATION

Whilst the process of normalisation is referred to as having three stages, there are in fact several additional steps required to carry out each stage. The following example is based on a hospital patients appointment card. All data elements are shown in the Un-normalised form.

I) *UN-NORMALISED FORM* (UNF)

List structure tabular format, identify structure key (with an underline) and repeating items (with a brace around the first and last item).

PATIENT RECORD =

Patient Number

Patient Name

GP Number

GP Name

{Appointment date

Consultant Number

Consultant Name

Sample Required}

UN-NORMALISED FORM

- ii) *FIRST NORMAL FORM* (1NF): Separate repeating items and place under a new key. (This new key set will contain the main key **and** another key that uniquely identifies the repeating items). Show the remaining data elements under the main key.

Patient Number

Patient Name

GP Number

GP Name

FIRST NORMAL FORM

Patient Number

Appointment date

Consultant Number

Consultant Name

Sample Required

- iii) **SECOND NORMAL FORM (2NF)**: For structures with a key which is more than one element, separate items not dependent on the whole key. ie Part-key dependency. List remaining elements in 1NF groups.

	<u>Patient Number</u>
	Patient Name
	GP Number
	GP Name
SECOND NORMAL FORM	<u>Patient Number</u>
	<u>Appointment date</u>
	<u>Consultant Number</u>
	Sample Required
	<u>Consultant Number</u>
	Consultant Name

- iv) **THIRD NORMAL FORM (3NF)**: Separate out those items which depend more upon non-key items than the key items. The new keys are replaced back in their original structures as 'Foreign' keys and denoted with an asterisk (*).

Identify derivable items by either removing them altogether or noting their existence at the bottom of the TNF table, showing how they are derived. Alternatively, if the derivable items have a real use, they may be left in the structure to save re-construction at a later date.

Finally test the TNF structures to ensure that:

- All non-key items only appear once.
- The correct relationships exist between items and structures.
- All items from the UNF have been included.

	<u>Patient Number</u>
	Patient Name
	GP Number *
	<u>GP Number</u>
	GP Name
THIRD NORMAL FORM	<u>Patient Number</u>
	<u>Appointment date</u>
	<u>Consultant Number</u>
	Sample Required
	<u>Consultant Number</u>
	Consultant Name

Normalisation is a process which requires practice and patience, and with experience, is a straightforward process.

The resulting data groups or structures are more manageable and the process itself ensures that consideration is given to all the relationships in the structures.

LOGICAL DATA STRUCTURING TECHNIQUE

The Logical Data Structuring Technique (LDST) forces the analysis and understanding of data and also provides an alternative view of the system. The DFD provides a functional view and the Logical Data Structure (LDS) a data view. Both views can be reconciled by a technique DFD/LDS Correspondence, which will be described later.

Data falls into natural groups which can be identified by an identifying data item, or data items - a unique Key - and contain related information in the form of non-key data items.

To construct the LDS it is necessary to identify the Entities and the Relationships between them.

So what is an Entity?

Something in the System
about which
Information is Kept
and can be
Identified by a Unique Key.

The first step is to identify the entities which satisfy all three of the above conditions. Draw a grid with these entities listed on the vertical and in reverse order on the horizontal axes of the grid. If there is a relationship between two entities put a cross in the square at the intersection. It might help if the relationships are expressed as simple sentences e.g.

A DEPARTMENT is composed of EMPLOYEES
A BRANCH services CLIENTS.

Note : 1. Plurals are used in this 'verbal' analysis but on the LDS the entity names must always be **singular**.

2. It is possible to have more than one different relationship between a single pair of entities. If this is found enter the number of such relationships on the grid e.g. '2' instead of a cross.

	Entity A	Entity B	Entity C	Entity D	Entity E
Entity A		X	X		
Entity B				X	
Entity C					
Entity D					2
Entity E					

Figure 10: LDS Grid

Once the grid has been completed draw a diagram with the entities depicted as named rectangles and the relationships as lines between them.

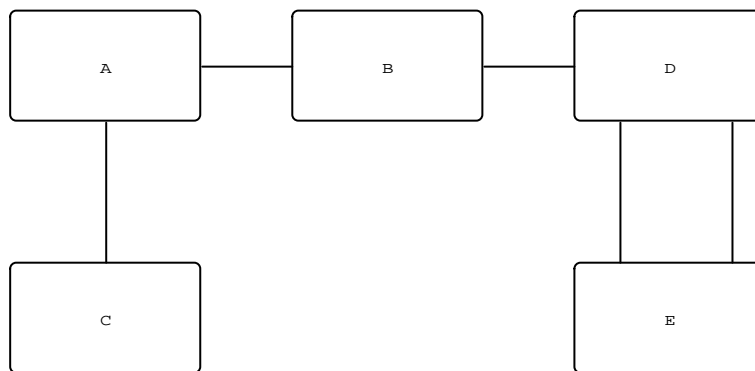


Figure 11

It is now necessary to decide the Degree of each relationship. There are three, illustrated as follows:

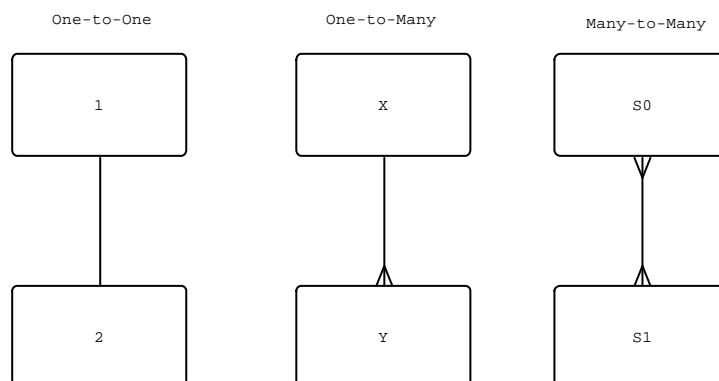


Figure 12

A simple Decision Table will illustrate the rules and actions to be taken when considering the relationship between each pair of entities - say Entity 1 and Entity 2.

	1	2	3	4
For a single occurrence of Entity 1 is there more than one occurrence of Entity 2?	Y	Y	N	N
For a single occurrence of Entity 2 is there more than one occurrence of Entity 1?	Y	N	Y	N
Degree of relationship	M:M	1:M	M:1	1:1
Add crows feet pointing to Entity	1&2	2	1	
Merge if possible				X
Add linking entity	X			
Add linking entity and check	X			
Master entity		1	2	
Detail entity		2	1	
Add crows foot pointing to Entity	1&2	2	1	
Merge if possible				X
Add linking Entity	X			
Add link to grid and check	X			

Figure 13

Let us assume that the relationships identified in Fig 10 have been checked against the decision table (Fig 14). The following degrees of relationship have been found :

- A to B - Many to Many
- A to C - One to Many
- B to D - Many to Many
- D to E - Two One-to-many.

Fig 11 now becomes :

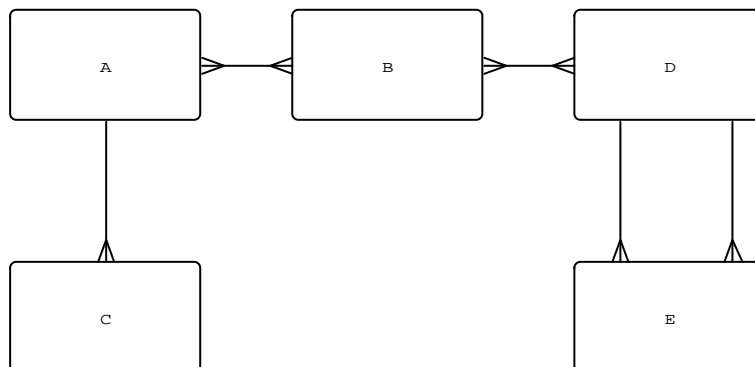


Figure 14

Rule 1 of the Decision Table states that, in the case of Many-to-many relationships 'ADD LINKING ENTITY'. There are two such relationships : A to B and B to D.

The diagram will now become :

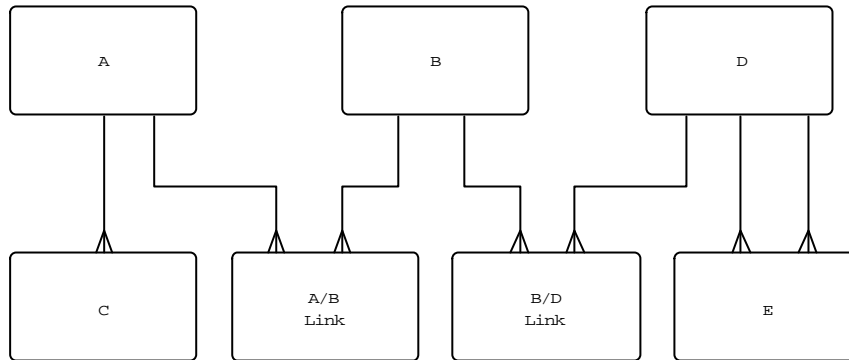


Figure 15

The naming of Link Entities is not easy but the names should be meaningful and reflect some aspect of one, or both of the entities being linked. One way is to 'promote' an attribute of one of the entities to become an entity in its own right.

For example :

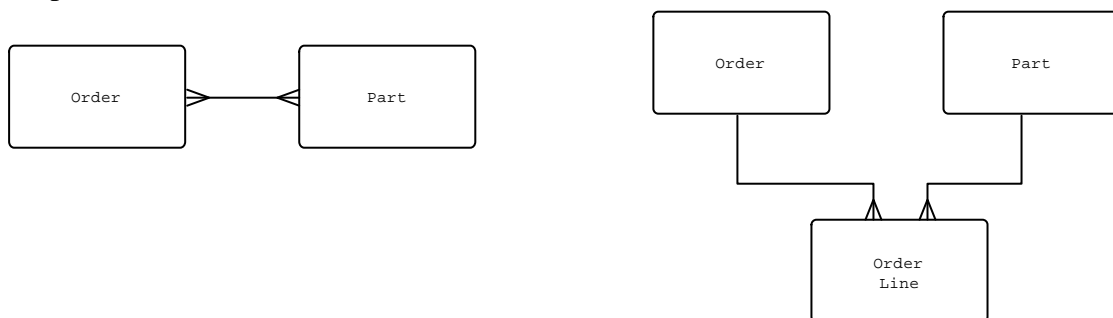


Figure 16 : Linking Entity

An order is composed of an Order Header containing details of Customer Name/address: Order No: Date:Customer Code etc and a series of Order Lines showing Item No: Part No: Description: Quantity Ordered etc. In Fig 16 the Order Header is represented by the Order entity and the attribute Order Line has been 'promoted' to a linking entity.

Note :

Any entities added to the model must be entered on to the grid to check whether there are any relationships between these additional entities and the original entities. If there are, annotate the grid and add the relationships to the model.

One-to-One Relationships

These are permissible for clarity but, where possible should be merged and renamed, e.g.

A Client has one Account

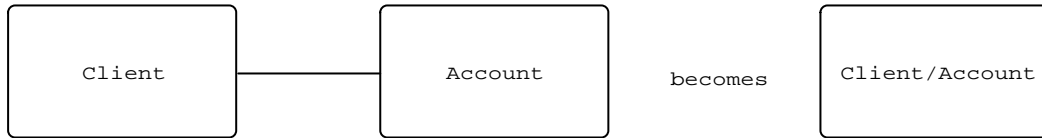


Figure 17

The merged entity Client/Account will contain all data about the client and his/her account.

Types of Relationship - Special Cases

Involuted Relationship

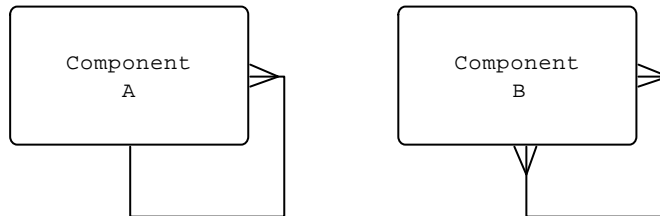


Figure 18

This will occur when an entity has a relationship with itself.

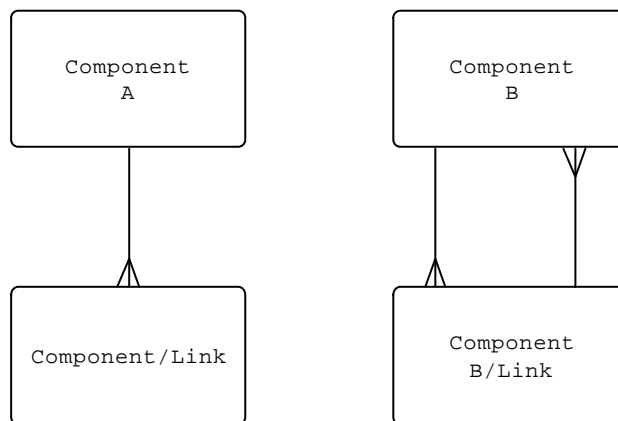


Figure 19

These involuted relationships are resolved by the use of linking entities :

Optional Relationship

Some relationships may, or may not exist at any one point in time, therefore the value of the occurrence is Zero; One Or More. In this case the relationship will be Optional and represented as:

Exclusive Relationship

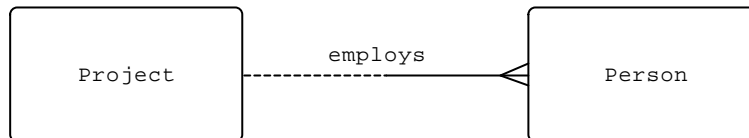


Figure 20

This could occur when there are two or more relationships with the same Detail entity. One of the relationships must exist, but more than one cannot exist at the same time. In the example below an AGENT can place an SALES ORDER and so can a SALESPERSON, but they cannot place the same order at the same time. The arc denotes that the relationships are mutually exclusive.

Validation and Documentation

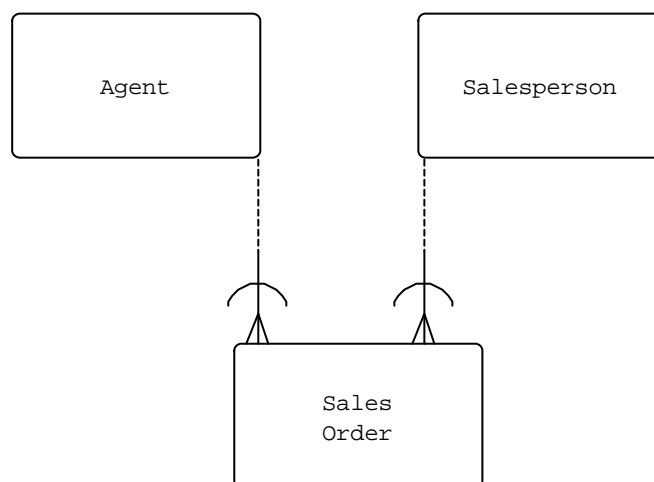


Figure 21: Mutually Exclusive

All entities should be described and their Keys and major Data Items listed i.e., Keys and Attributes.

Volumetric data should also be recorded.

Using the processes from the DFDs, the access paths through the LDS should be tested to check for either missing entities and relationships.

General

It should be noted that experienced analysts can combine many of the steps described above.

Key Set Definitions

Prime or Simple Key

is a single element key;

eg: INVOICE NO.

An alternative prime key is known as a Candidate Key.

Compound Key

is where each element of the key is the key to another data structure or group;

eg: INVOICE NO.
PRODUCT NO.

Composite Key

is where the 1st element (qualifying) is a key to another group, BUT, other elements (qualified) are meaningless on their own and are NOT keys to other data groups.

eg: INVOICE NO.
%ITEM NO. .

Compound and Composite Keys are also known as CONCATENATED KEYS because they are made from two or more data elements.

Foreign Key

is a Non-Key data element which is the key to another data groups or structure. It is usually identified by and asterisk (*).

UNNORMALISED FORM <i>(List all items and underline the form's key item, identify repeating items)</i>	1st NORMAL FORM <i>(Separate into groups of items which repeat and identify key)</i>	2nd NORMAL FORM <i>(Separate out items of part-key dependancy)</i>	3rd NORMAL FORM <i>(Separate out items which depend on a non-Key item, deal with derivable & test)</i>