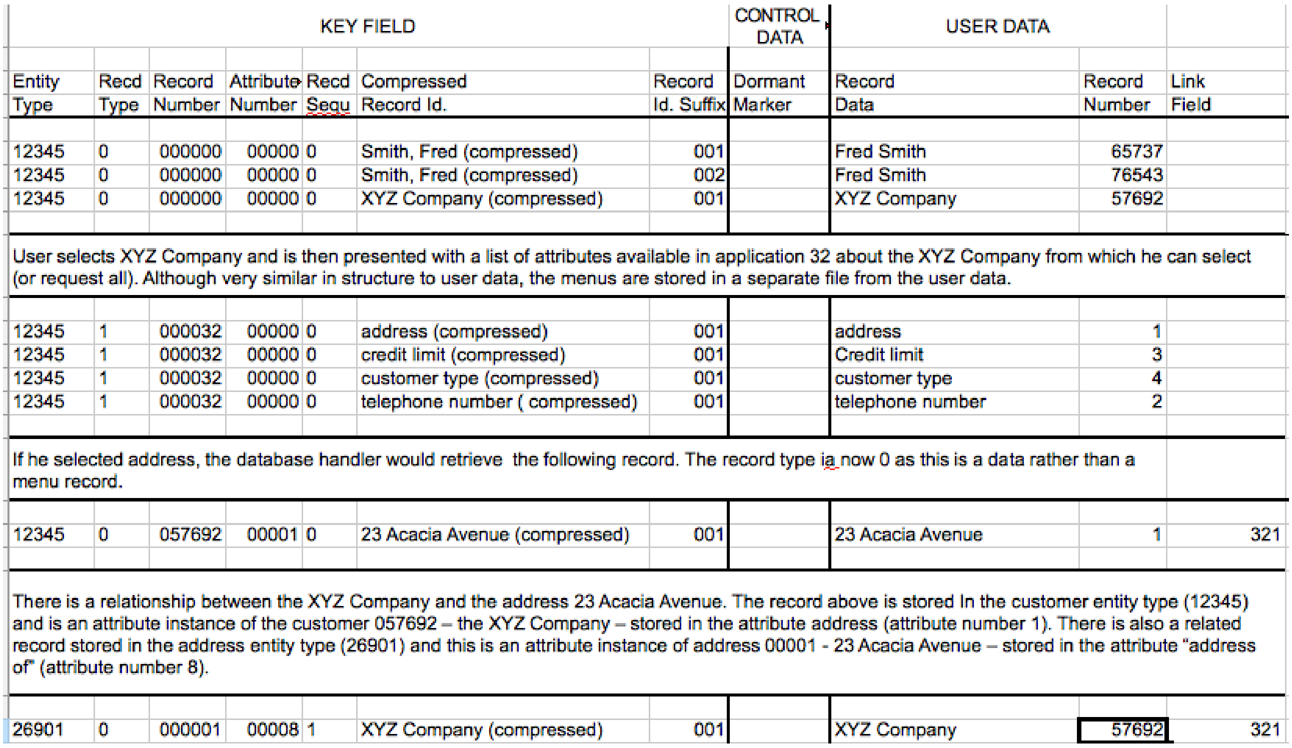
Relationships and Surrogates in ERROS

Most attributes in the ERROS database are defined as relationships. This means that, using addresses as an example, each address is stored as an individual record in a separate table of addresses, accessible by address (city before street) or by post or zip code. Relationships are then created between people or businesses and their addresses. There could be multiple address relationships – postal or billing or delivery addresses. As the ERROS database stores many-to-many relationships by default, each person or company can have many addresses of each type and each address may be related to many people or businesses. If, when putting up a new contact, the operator finds that the address already exists, he can immediately check who else is at that address. If the previous occupant has left, the relationship will have been deleted, but the address should not have been deleted as it is still a valid address. Deleted records can be retrieved by operators with adequate security so that history is not destroyed.

To find out who occupies an address, rather than using an SQL query to search the customer database, the operator can go to the address table, type in the post code or the address and immediately find the name(s) of the current occupier(s) recorded in the database. This only takes a second or two. All relationships can be navigated in either direction at the same speed, so, having started with an address, the operator can click on the name of an occupant and immediately navigate to their record and look up any information about them.

All relationships in ERROS are bi-directional and, when a relationship is created, one record with the address is put in the customer table under the customer number, and another record with the customer name is put in the address table, under the address number. The two records are connected by a link field, which is invisible to operators. This enables navigation and it also links the two records together so that they are both updated or deleted together.

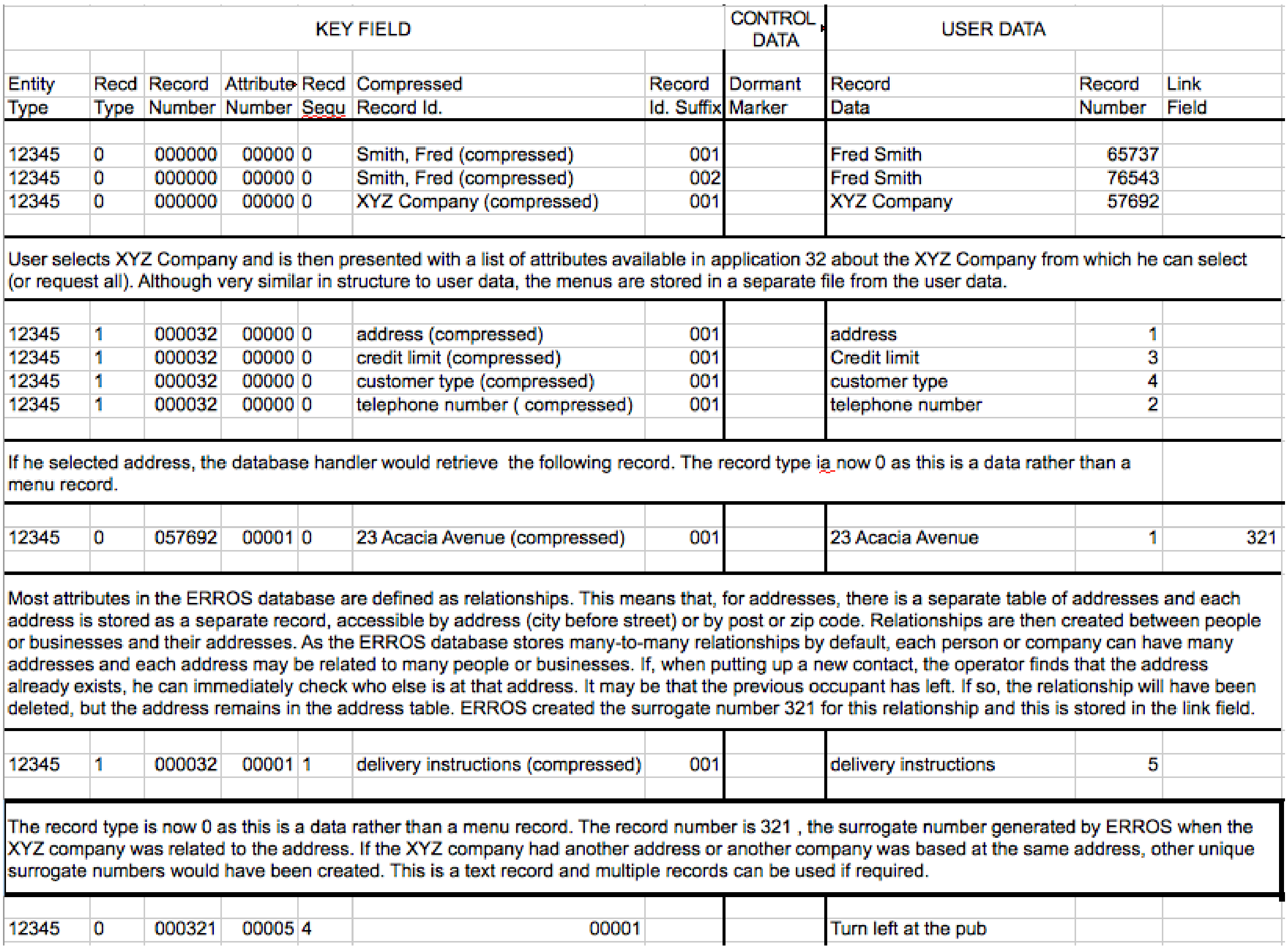


KEY FIELD							CONTROL DATA	USER DATA		
Entity Type	Recd Type	Record Number	Attribute Number	Recd Sequ	Compressed Record Id.	Record Id. Suffix	Dormant Marker	Record Data	Record Number	Link Field
12345	0	000000	00000	0	Smith, Fred (compressed)	001		Fred Smith	65737	
12345	0	000000	00000	0	Smith, Fred (compressed)	002		Fred Smith	76543	
12345	0	000000	00000	0	XYZ Company (compressed)	001		XYZ Company	57692	
User selects XYZ Company and is then presented with a list of attributes available in application 32 about the XYZ Company from which he can select (or request all). Although very similar in structure to user data, the menus are stored in a separate file from the user data.										
12345	1	000032	00000	0	address (compressed)	001		address	1	
12345	1	000032	00000	0	credit limit (compressed)	001		Credit limit	3	
12345	1	000032	00000	0	customer type (compressed)	001		customer type	4	
12345	1	000032	00000	0	telephone number (compressed)	001		telephone number	2	
If he selected address, the database handler would retrieve the following record. The record type ia now 0 as this is a data rather than a menu record.										
12345	0	057692	00001	0	23 Acacia Avenue (compressed)	001		23 Acacia Avenue	1	321
There is a relationship between the XYZ Company and the address 23 Acacia Avenue. The record above is stored In the customer entity type (12345) and is an attribute instance of the customer 057692 – the XYZ Company – stored in the attribute address (attribute number 1). There is also a related record stored in the address entity type (26901) and this is an attribute instance of address 00001 - 23 Acacia Avenue – stored in the attribute "address of" (attribute number 8).										
26901	0	000001	00008	1	XYZ Company (compressed)	001		XYZ Company	57692	321

The number 321, which is also part of the link field, is the surrogate number of both the records. Surrogate numbers are explained below.

In ***Storing Hierarchies in the ERROS Database***, I stated that the key field structure of the ERROS database allowed the storing of up to 999,999,998 levels of hierarchy with records at each level indexed in alphabetic sequence with a 64 character name, yet the key field length of ERROS remains constant at 28 bytes at all levels. Even though the patented 2 stage ERROS compression algorithm reduces the 64 character name to 4 bytes, that only copes with a single level of hierarchy before you would need to expand the key field at lower levels. When I was designing ERROS, evolving a solution to this problem perplexed me for some time. What I seemed to need was a key field that could grow by four bytes at each lower level, but database systems could not cope then with variable length key fields. My assumption was that, even if they had had this capability, performance would be poor. There had to be another solution.

I realised that if I could create a unique, perhaps hidden, identifier, just 4 bytes long, each time I added a record to the database, this would seem to do the trick. If I created a unique series of numbers for every context (i.e. each entity type and attribute combination), and used those numbers as the record number for the records at the lower levels, then the problem was solved. Of course, there might be an enormous number of contexts, so I had to design an automatic, generic solution that did not require any manual intervention. I was able to include this in my successful patent application. This works very well and I call them surrogate numbers. Users are totally unaware of their existence. Associated with each surrogate number is a surrogate database record which can be used to store user data. I do not call these surrogate keys as the numbers generated are only one constituent of a multi-part key field.



KEY FIELD							CONTROL DATA	USER DATA		
Entity Type	Recd Type	Record Number	Attribute Number	Recd Sequ	Compressed Record Id.	Record Id. Suffix	Dormant Marker	Record Data	Record Number	Link Field
12345	0	000000	00000	0	Smith, Fred (compressed)	001		Fred Smith	65737	
12345	0	000000	00000	0	Smith, Fred (compressed)	002		Fred Smith	76543	
12345	0	000000	00000	0	XYZ Company (compressed)	001		XYZ Company	57692	
User selects XYZ Company and is then presented with a list of attributes available in application 32 about the XYZ Company from which he can select (or request all). Although very similar in structure to user data, the menus are stored in a separate file from the user data.										
12345	1	000032	00000	0	address (compressed)	001		address	1	
12345	1	000032	00000	0	credit limit (compressed)	001		Credit limit	3	
12345	1	000032	00000	0	customer type (compressed)	001		customer type	4	
12345	1	000032	00000	0	telephone number (compressed)	001		telephone number	2	
If he selected address, the database handler would retrieve the following record. The record type ia now 0 as this is a data rather than a menu record.										
12345	0	057692	00001	0	23 Acacia Avenue (compressed)	001		23 Acacia Avenue	1	321
Most attributes in the ERROS database are defined as relationships. This means that, for addresses, there is a separate table of addresses and each address is stored as a separate record, accessible by address (city before street) or by post or zip code. Relationships are then created between people or businesses and their addresses. As the ERROS database stores many-to-many relationships by default, each person or company can have many addresses and each address may be related to many people or businesses. If, when putting up a new contact, the operator finds that the address already exists, he can immediately check who else is at that address. It may be that the previous occupant has left. If so, the relationship will have been deleted, but the address remains in the address table. ERROS created the surrogate number 321 for this relationship and this is stored in the link field.										
12345	1	000032	00001	1	delivery instructions (compressed)	001		delivery instructions	5	
The record type is now 0 as this is a data rather than a menu record. The record number is 321 , the surrogate number generated by ERROS when the XYZ company was related to the address. If the XYZ company had another address or another company was based at the same address, other unique surrogate numbers would have been created. This is a text record and multiple records can be used if required.										
12345	0	000321	00005	4	00001			Turn left at the pub		

If another customer was at the same address, a different surrogate number would be created. By using the surrogate number rather than the address number as the record number, a different delivery instruction could be entered for each occupant at the same address. Another person might say "opposite the church – 9.00am-1.00pm only". There might well be an address number 321 but the ERROS database handler can distinguish between address number 321 and surrogate 321. Surrogates are a vital part of the ERROS database structure.

This document describes some of the principles on which ERRROS works, but the reality is more complex. All of the above and so much more can be achieved without any programming.

Rob Dixon – 29th June 2017

rob.dixon@erros.co.uk