

Defining ERROS Record Formats and ERROS Exit Programs

Storing Hierarchies in the ERROS Database states "Apart from menu records and one or two other controlling records, such as operator profiles, the ERROS database handler does not need to understand the layout of any of the record formats that might be found in any standard commercial or humanities application."

All records in all data definition, application definition and user data tables are defined in the ERROS Connectionist Database as having just three fields –

key field
control data
user data

The user data field, unless it stores text, includes a link sub field that is not seen by users.

As explained in **Horizontal or Vertical? Beyond the Relational Database**, each record contains one iteration of an attribute, such as one contact name, one address, one telephone number, one order line, one employee name, etc. These records are all the same length. Where the data for one attribute record is too long to fit in the user data space, multiple records could be used. There is no effective limit on the number of attributes that can be defined for any entity type and the amount of data that could be stored about any one entity, such as a customer, might be greater than the amount of available disk space.

Because the ERROS Connectionist Database allows, by default, multiple iterations of all attributes, any attribute can be defined as repeating and this is achieved by storing multiple records where required. Where there is no value for an attribute – perhaps you don't know the name of any of the employees of the XYZ company – no record is written and no null values are required. The absence of a record shows that there is no data in the attribute. The ERROS database handler can retrieve all or a subset of the attribute values for a customer, or an order, etc. and the ERROS presentation layer can present them together to the operator as a single logical record if required. It can combine data about related entities, even if they are not of the same entity type.

Any attribute can be defined as having multiple fields, even though, with the exceptions mentioned earlier, the ERROS Database Handler will not be aware of the structure of these. This is achieved by storing a three character alphanumeric format code in the control data part of every record in the database. This causes the database handler to pass the record to an exit program in which the record layout has been defined and which is therefore able to split it into its individual fields. ERROS is object-oriented and the same format code can be used in multiple attributes and the same attribute definition can be used in many entity types. Many attribute format codes are already defined in ERROS and can be used by developers in newly created attributes and in new applications.

One of the formats already defined has the code 'NN '. This is used for contact names and would be defined as the format for the identifying name attribute for a variety of entity types such as customer, contact, person, supplier, employee, agent, etc. The exit program will split the field into several fields, such as title, last name, first name, gender, etc. The database handler will automatically call an exit program based on the first letter of the format code 'N'. There can be multiple versions of any exit program and there can also be user versions so that for a particular user group or company, the exit program might have different processing. The version to be selected is defined in the menu record that specifies the characteristics of the attribute. The menu records would normally be application specific so a different exit program could be used in different applications or in different menus in the same application.

If a new record format is created and new exit program coding is required, no changes are required to the ERROS database handler or to the presentation layer, and the changes can be made whilst ERROS is live, without affecting the response times of any users on line at the time. The new exit program will be called dynamically by the ERROS main programs, using the standard ERROS exit program interface.

This means that new attributes with new format codes can be added to entity types that are being used in live applications, but they won't be available in any application until they are put into a menu for an application. They can be fully developed in a test application before they are put into an active application.

If a record is to be added or updated, the exit program, generally in conjunction with the ERROS presentation layer and the ERROS Standard Operating Interface (SOI), will present the record field values for update to the browser. The user can type in any changes and these will be read in by the exit program which will move them back into the user data field and so back to the database handler which will then update the record as a single data field. Exit programs do not have access to the ERROS database as this is handled by the ERROS database handler.

When a record is to be displayed, the exit program will break it down into its constituent fields and pass them to the presentation layer which, using SOI, will generate the necessary HTML and Javascript for sending to the browser. SOI can allow more than one program to contribute to the HTML sent to the browser.

Relational databases only allow one record type to be present in a data table. There is no such restriction in the ERROS database.

In **Relationships and Surrogates in ERROS**, I stated that, associated with each surrogate number, is a surrogate database record which can be used to store user data. This has its own format code which would probably be different from that in the attribute data record. For a relationship, the ERROS database handler will automatically retrieve the surrogate record from either side of the relationship and the data from this can be displayed in both views of the relationship.

In a sales order, a sales order line will be a relationship between the sales order and a product as in the following example -

The screenshot shows the ERROS application interface. At the top, there is a navigation bar with buttons: Enter, JUMP BACK, Previous, Refresh, Navigate, Audit, View, Help, Print, Operate, Options, Logout. Below this, the application title is "Stock Level Adjustment (An ERROS Application)" and the user is identified as "Rob Dixon" for "Erros plc (ERROS08)". There are also buttons for "Initial Menu" and "Fast Path".

The main content area displays a sales order summary:

Sales order
date and/or number
24/01/13 17.37 Thomas Linley Baths Ltd #000000002
product ordered

Below the summary is a text input field with the label "# (Number) 0".

At the bottom of the summary area, there is a small table with buttons: Enter, Repeat Input, Add, and a small square button.

	Quan.	Price	Incl.	Quan.
	Ord.	Incl.VAT	Total	Due
3 Drawer metal filing cabinet	# 8	1	56.00	56.00 1
A4 Paper, ream, 100 gsm	# 12	24	4.00	96.00 24
Board Room table	# 7	1	1000.00	1000.00 1
3 Items EX VAT 960.00 VAT 192.00		26		1152.00 26 OVERALL TOTALS

At the bottom of the interface, there are buttons for "Earlier Records" and "More".

The customer name, Thomas Linley Baths Ltd, the date and time of the order and the order number are displayed at the top of the page. The product names and numbers are stored in the attribute data records for the attribute "product ordered" whilst the numeric data is stored in the surrogate records. 24 reams of the second item, A4 paper, have been ordered. If the operator clicked on that order line, ERROS would navigate to the product and display the following screen which lists all orders for A4 paper -

Enter JUMP BACK Previous Refresh Navigate Audit View Help Print Operate Options Logout
 Stock Level Adjustment (An ERROS Application) Rob Dixon Erros plc (ERROS08)
 Initial Menu Fast Path
 product
 name and/or number
 A4 Paper, ream, 100 gsm #00000012
 orders received
 # (Number) 0 Date 0 / 0 / 0 Time 0 . 0
 Enter Repeat Input Add

			Quan.	Price	Incl.	Quan.
			Ord.	Incl.VAT	Total	Due
10/01/13 12.24	Esmond Estates Ltd	# 1	20	4.00	80.00	20
24/01/13 17.37	Thomas Linley Baths Ltd	# 2	24	4.00	96.00	24

Earlier Records More

Now it is the product name and number that the operator selected that are displayed at the top of the page. The orders are in date and time sequence and the second line is the order for 24 reams for Thomas Linley Baths. The numeric data is exactly the same data as on the previous page and in both cases has been retrieved from the same surrogate data record. This can be considered as data that is part of the relationship. The numeric part of each line is assembled by an exit program and passed to the ERROS presentation layer which assembles the left part of each line and the rest of the page and generates the necessary HTML and Javascript (to allow double clicking on the product or customer names) for sending to the browser. This may sound complex but is simple.

If the operator double clicks on the order for Esmond Estates Ltd., ERROS will navigate to the complete order. The operator could click on any product in that order and navigate to the record for that product. This ability to navigate from one record to any other related record in either direction at the same speed was a most important part of my initial requirements when I first started to think of the features that I wanted to include in ERROS as standard. This means that an operator can browse through the data and discover relationships that he or she did not know existed. In relational databases, the operator has to use an SQL query which someone has to create. By its nature, creating such a query is anticipating a result. The ERROS facility that allows operators to browse without such anticipation helps to turn raw data into a knowledge base. It is a very valuable feature in both business applications and in the humanities.

The colour selections are stored in the ERROS database and can easily be changed at any time as can the column widths.

As previously described, the characteristics of the data in an attribute are defined in menu records. These include the format code for both the attribute and the surrogate data and these codes will be used when new records are created.

ERROS Exit Programs

One standard exit program which handles records identified by name, such as contact name, company name, operator profiles, menu records, addresses is always required. This handled the customer names and the product names in the screen above. Others are created as required. They can be written in the programmer's preferred language. The surrogate records in the screens above were processed by an "Accounting" exit program. The user is totally unaware of the existence of exit programs. If a developer uses an existing format code, then he can be sure that ERROS will automatically call the correct exit program and he does not need to specify what this is unless he required a change in the way the record is processed or presented.

The exit programs all have the same, relatively simple, structure. They are called by the main ERROS programs. A large predefined number of parameters is passed to and from the exit programs. Amongst these is a "function" code that tells the exit program what it is required to do. In addition, a "view" code is passed which the exit program uses to determine how the record is to be displayed – e.g for personal names, as first name, last name or as last name, first name.

Amongst many other parameters are the primary and secondary accounting currencies of the company, the format of prices used in the company (e.g. with pence or to nearest pound or dollar), the minimum price and the maximum price (to eliminate some of the nonsensical errors that can occur as reported in the press from time to time), and other similar parameters for quantity, VAT, etc.

These values are stored in the ERROS database for each company and retrieved when the operator signs on.

An exit program can perform any standard program function – send a message or read or write an external non-ERROS file and pass data to or from ERROS, perhaps for importing data into ERROS or exporting ERROS data. Although the ERROS Standard Operating Interface is used by default, there is an option that allows an exit program to present a totally different interface if required, although that would have to be coded.

An exit program could also do printing, but this is normally done by the ERROS presentation layer which generates PCL5 printer code. Very large complex reports can be defined in the ERROS database. Text can be typeset.

Once an exit program has been coded for a record format, as in the two figures above, nothing further has to be done for printing those records as ERROS inserts PCL5 rather than HTML and Javascript into the data stream and sends it to the printer.

Once an ERROS record format code has been defined and any necessary exit program coding completed, that format code can be re-used in many different attributes in many different entity types, as ERROS is fully object-oriented, although this is quite different from existing object-oriented database and programming techniques. Developers using ERROS do not need to understand object-oriented concepts.

ERROS allows new attributes to be added to existing entity types and the creation of new entity types with already defined or new attributes, and this can be done whilst the system is live. Because the vertical structure of the ERROS database means that no existing records are changed, current applications will not be affected until the new attributes or entity types are included in them by adding them to menus. This is so unlike traditional applications based on relational databases where perhaps hundreds or even thousands of programs access, say, a customer record file and, because an extra field has been added to the record layout, they all have to be recompiled, even though for many, and perhaps the majority, the extra field is not required in the applications in which they are called.

Thus ERROS dramatically reduces the massive existing problem of system maintenance, which may consume large amounts of the time of analysts and programmers and which, however carefully planned, is liable to be error prone and may turn into a nightmare.

Rob Dixon

25th July 2017

rob.dixon@erros.co.uk
www.erros.co.uk

