

ERROS Triggers and Procedures

With relational databases, it is always possible to update a file outside an application, using SQL or another utility or a program that is not part of an application. In traditional systems, trigger programs can be used to track record changes or to create an audit trail. Trigger programs can also be used to add, change or delete records. They may not be complex but they still have to be specified, written and tested. In addition, they can significantly affect performance. Hundreds or even thousands of trigger programs may be required in large complex systems.

Because of the unique vertical structure of the ERROS Connectionist Database, and its complex, binary multi-part key fields, the ERROS database cannot be updated by SQL or any other utility as it is not possible to specify to a utility which records are to be updated or which fields are to be changed. Only the ERROS database handler program can update the ERROS database and this can only be called in an ERROS application. This is one of several factors that make the ERROS database much more secure than relational databases. The ERROS files are all journalled with before and after images being recorded. The journal records the name of the program that changed any file. In ERROS, this will always be the same program. If journaling is switched off, ERROS will issue an error message and stop working. ERROS has its own audit trail that records the date and time of every change to the database and who made it and in which ERROS application. This applies to all data and application definitions as well as to all user data.

ERROS has its own form of trigger mechanism but these can cause the automatic addition, update, deletion (in ERROS delete means mark as deleted but not removed) or removal of records from the system. For example, if a record is required in an attribute but none are present, the menu record that has led to that attribute can indicate that ERROS is to add one. The definition in the menu record for the attribute will specify that the operator is to be prompted to enter values for the new record or that it is to be added without the operator's knowledge. This latter option is necessary for transaction processing (as are update, delete or remove) when multiple records need to be changed as a group, using commitment control. The boundaries to start and end commitment control are defined in the menu records.

Complex procedures can be defined in menu records. They can be used for defining the steps in transaction processing, for defining the layout of ERROS web pages and for defining complex printed reports. All of these can combine records from multiple entity types for updating and also for display and print.

Where there is more than one menu record in a menu, the previously menu record may determine that the records should be displayed to the user so that he can select one or it may determine that the records are to be read one at a time, as in a procedure, going on to the next data or menu record and so on until there are no more records to be found. In such a procedure, ERROS might then go backwards, looking for further records at each level and moving forwards and then backwards until it returns to the starting menu. It will then read the next record in that and move forward again and then backwards, only stopping when it has finished processing all the records in the starting menu. This might be used for transaction processing or for creating complex reports. There can be tests in a procedure, that, if valid, instruct ERROS to exit the procedure at other points. Menus in a procedure will not generally be visible to a user.

ERROS procedures can be used in a great variety of ways. For instance, when a customer name is selected from a list, it might be displayed with address, telephone number(s), contact name and date and value of last 3 orders but omitting any other attributes. A procedure will retrieve the necessary data, which is stored in multiple records and display (or print) them as a single logical record. Alternatively, when entering sales orders, ERROS can look for the best price for the customer, searching first for prices or discounts for a product or product group for that particular customer, or, if those are not found, perhaps for discounts for the geographical area of the customer, etc. If no special price or discount is found, then ERROS would retrieve and display the standard price. Prices can be date and time related and have an end date and

time, so that a special price for a limited time can be entered in advance. ERROS will automatically apply this when appropriate and then revert to the standard price. This can all be achieved by selecting the various options during development and without any programming.

Rob Dixon

2nd July 2017

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