

Referential Integrity



Jackie Jansen

I have recently had the opportunity of talking to many System i customers about query and reporting tools and our database. This includes over 150 enthusiastic members in Toronto at TUG's May meeting.

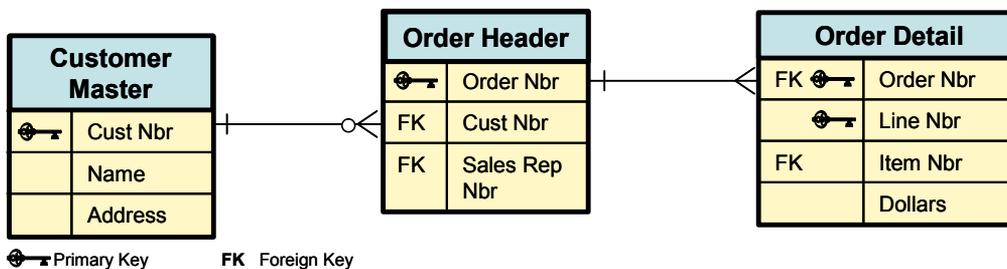
One thing that has become very clear to me is the number of customers that have not implemented referential integrity. Even more interesting, when I described foreign key constraints in my various presentations, I was asked by many of you if this functionality came in V5R3 or V5R4? Now, I have been checking into old manuals and I haven't been able to confirm the actual date and release that this support was added to the AS/400. To the best of my recollection, I believe that I went across Canada talking to customers about this support when we announced V3R4 which was well over ten years ago. It surprises me how few of you, in over a decade, have taken advantage of this basic relational functionality in the database. What I would like to do in this column is describe referential integrity at a fairly high level and then drill down into specifics and implementation in future columns.

Header table is called a foreign key. When you define a referential constraint (primary key constraint or unique constraint) you are telling DB2 that when a record is inserted into the Order Header file the system must check that the customer number points to a valid customer in the customer master file. If the customer number is invalid the database will prevent you from adding the record and will send an error condition to your program. Without referential constraints you must do this checking yourself. One benefit of referential integrity therefore is the reduction in programming requirements and as a result a reduction in programmer time and effort. A second and probably much more important justification is that the database will enforce this existence check for a valid customer regardless of the interface used to modify the table. Although programs written in RPG and COBOL usually check for a valid customer, this doesn't prevent someone using DFU or interactive SQL or even a PC ODBC utility from adding a record with an invalid value. When this rule or constraint is defined to the database, a record cannot be added via any interface at all if it will cause the rule to be broken.

not to allow a customer to be deleted that has orders associated with it. On the other hand you might specify that when someone tries to delete an order header record any dependant detail records should automatically be deleted. This cascades the delete from the order header through to the order detail records.

Referential constraints can be defined through iSeries Navigator or through the CL command ADDPFCST (add physical file constraint) or by using the SQL CONSTRAINT clause in the CREATE and ALTER table statements.

Some query and reporting tools such as IBM's DB2 Web Query for System i take advantage of foreign key definitions and referential constraints. With DB2 Web Query, when an end user asks to work with our Order Header file they will automatically see the customer's name and address associated with the order as if these fields actually were in the Order Header file itself. The end user never needs to reference the customer master file or create a join to the customer master file. This greatly contributes to ease of use.



When you define referential integrity constraints to DB2 for i5/OS you are defining the relationship between two tables, such as a parent and a dependant table. In essence you are moving some of your business logic down to the database to administer.

In the example shown here, you can see that the Order Header file contains a customer number. Customer number is a field in our customer master file that uniquely identifies a customer. Customer number in the Order

In addition to checking for a valid customer at insert time you can also use referential constraints to tell the database the conditions that need to occur for a delete to be valid. For example, what should happen if a program tries to delete a customer record when there are currently orders that point to that customer? You can tell the database to reject the delete, or to allow it but change all the affected orders to have a null or default customer number or you can tell the database to cascade the delete. In our example, you might tell the database

Modern programming design dictates moving these types of business rules out of the programmer's hands and into the database for enforcement. 

Jackie Jansen is a Senior Consulting IT Specialist. She currently works in the IBM Americas Advanced Technical Support Solutions Centre. Jackie is a frequent speaker at iSeries Technical Conferences and User Group meetings. Contact her at jjansen@ca.ibm.com.