however, migrations at 12.0 beta sites have been both fast and smooth. It seems reasonable that a migration to 12.0, even assuming some glitches, should take less than a week. Of course, we must bear in mind that nothing just happens. While CA has set things up to proceed cleanly, it obviously won't happen that way without planning. As with change control, the essence of success lies with careful planning.

Interestingly, perhaps the most important of all aspects of the migration is something we will not have to include in the plan. A migration to 12.0 does not affect the data in the data base. The current data and data structures remain intact and unchanged. What's more, application programs are upwardly compatible and will run without change in the new 12.0 environment.

Information will not be altered, the corporate asset will be protected, and CA will have to its credit both a giant leap and a small step. Not only will CA have painlessly migrated its CA-IDMS users to a whole new technological world, but it will have made Bhaskara Chary happy.

This column shares information and ideas among those who are responsible for IDMS data base environments. What data base struggles have you encountered recently? What problems have you recently solved in the IDMS arena? Feel free to submit your concerns, thoughts and ideas that focus on IDMS issues to "IDMS Concepts" in care of 370/390 ***Data Base Management.***

■ *Was this article of value to you? If so, please let us know by circling Reader Service No. 88.*

SQL/DS

by R.G. Eaton

New Horizons

The SQL/DS EXPLAIN Command

Editor's Note: This month's column deals with the EXPLAIN command for SQL/DS. Our thanks to R.G. Eaton for contributing this material.

R.G. Eaton is a development team leader at Twentieth Century Services, Inc. He has worked with SQL/DS in varying roles, including data base administrator, since 1984.

Description

One of the virtues of SQL/DS is an optimizer to perform data access path selection. Because SQL/DS's optimizer determines data access paths, SQL/DS provides the EXPLAIN command to retrieve information about optimizer decisions. You may consider it a window to the optimizer. Users can review this information to:

- ensure that indices are being utilized;
- identify additional index requirements; and
- performance tune SQL code.

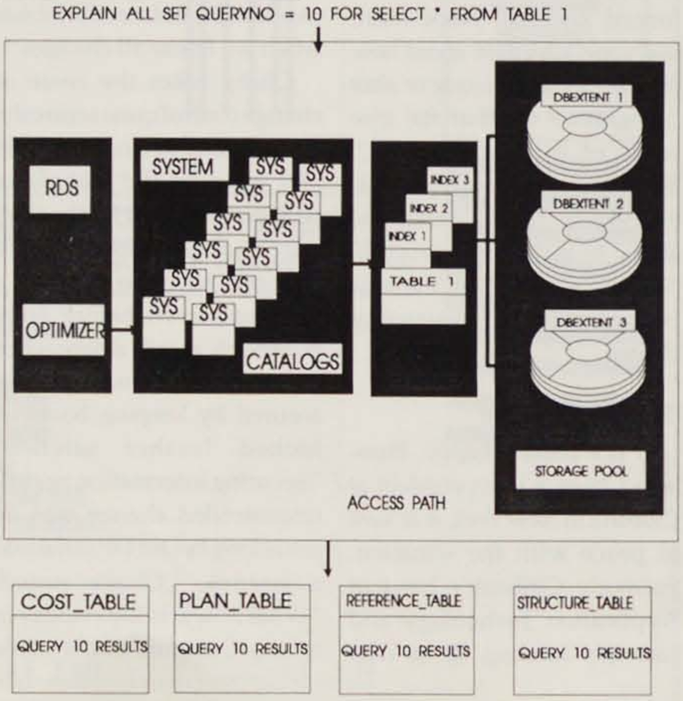
The EXPLAIN command analyzes an SQL instruction and inserts the results into EXPLAINation tables. However, the SQL instruction is never actually performed. SQL commands that can be explained include: DELETE, INSERT, SELECT and UPDATE.

The most common use of the EXPLAIN command is to analyze SELECT commands or queries. Optimizer information is EXPLAINed into one or more of the following tables:

REFERENCE_TABLE: This table contains the columns affected by the SQL code. It also provides a filter factor that indicates the selectivity of the "where" predicate.

STRUCTURE_TABLE: This table contains an estimate of the number of rows the SQL

Figure 1: Tracking EXPLAIN Results


EXPLAIN ALL SET QUERYNO = 10 FOR SELECT * FROM TABLE 1
RDS
OPTIMIZER
SYSTEM
SYS
CATALOGS
INDEX 3
INDEX 2
INDEX 1
TABLE 1
DBEXTENT 1
DBEXTENT 2
DBEXTENT 3
STORAGE POOL
ACCESS PATH
COST_TABLE
QUERY 10 RESULTS
PLAN_TABLE
QUERY 10 RESULTS
REFERENCE_TABLE
QUERY 10 RESULTS
STRUCTURE_TABLE
QUERY 10 RESULTS