SAS Applications Programming with DB2/SQL

David J. Adamson

INTRODUCTION

A relational data base management system (DBMS) can be thought of as a set of sequential or flat files (VSAM), called tables. These tables consist of records and fields and SAS data sets.

Each table in the relational database is based on an entity. An entity is any object upon which other data elements are based. The columns in each of these tables relate directly to the corresponding entity. The tables themselves relate to each other through common columns, thus providing the ability to join multiple tables in order to extract the required information needed for a particular application.

SAS\DB2 is an interface between the SAS system (a 4th G\l) and the DB2 environment (a relational DBMS). The relational database model has a universal language that is used to define, extract, update, and control the data structures, this language is referred to as Structured Query Language (SQL). SQL is used to access the data stored in the DB2 environment (tables). SAS is used for the data manipulation, data analysis, and complex reporting capabilities which it provides. DB2 runs in the MVS\XA operation environment (Multiple Virtual Storage\Extended Architecture) DB2 can be accessed concurrently by:

1. TSO\SPF: TSO foreground is where SAS Display Manager runs. DB2 can be accessed through Display Manager.
2. Batch processing: Sometimes referred to as TSO background. DB2 can be accessed by SAS batch programs.

The SAS\DB2 environment contains the following components:

1. The ability to migrate from SAS data sets (fields\variables) to DB2 base tables (rows\columns).
2. Translating from SAS statements to SQL statements on relational database model. For example the SAS "If" and "Delete" statement can be compared to the SQL "Select" statement. The SAS "Merge" statement can be compared to the SQL "Join" statement.

The SAS\DB2-SQL environment consists of three procedures which can be executed interactively in Display Manager or in line mode (TSO\SPF), and in batch programs. These are as follows:

1. Proc DB2EXT, which extracts the data stored in DB2 tables and places them in a SAS data set.
2. Proc DB2LOAD, creates and loads DB2 tables with data stored in a SAS data set.
3. Proc DB2UTIL, is used to add rows to a DB2 table, delete rows from a DB2 table, update column values in a DB2 table, or add new columns to a DB2 table.

Each of these will be covered in greater detail in the paragraphs to follow.

THE DB2 ENVIRONMENT AND SUBSYSTEM OVERVIEW

The DB2 software is divided into three major components as follows:

1. The Resource Lock Manager (IRLM).
2. Database Services.

A DB2 table(s) may be in the process of being updated by a SAS application at the same time a user is trying to access those tables. The "IRLM" protects the integrity of the data and users will be temporarily "locked out" while tables are being updated. DB2 base tables and indexes are stored in 4K logical pages. A lock can be placed on a page or subpage.
at the lowest level and on a table space at the highest level. Locking cannot be set at the row level in current releases of DB2. A SAS data set will be created containing those rows the program had already extracted when the lock occurred. A SQL error will appear on the SAS Log (err-911), but the program will end with a condition code of 0 (zero).

The Database Services component is responsible for managing the creation and access to DB2 objects such as tables, table spaces, and indexes. Also keeping the DB2 system catalog current.

The System Services component manages the DB2 system as it interacts with other vendor products such as TSO/ISPF, SAS, CICS, BATCH, and IMS/VS.

The DB2 objects include base tables, storage groups, table spaces, views, indexes, index spaces, data bases, buffer pools and synonyms. A DB2 object is created and manipulated using the "SQL" language.

1. Tables, the rows correspond to records in a file or observations in a SAS data set. The columns correspond to fields in a file or variables in a SAS data set. As a rule, all columns within a table provide data relating to the same subject matter, called an entity.

2. Storage Group is a collection of DASD volumes assigned to the DB2 environment. DB2 automatically allocates space for tables on an as needed basis from the assigned DASD volumes.

3. Table space, is the physical space where tables are stored. A table space can contain more than one table and can be divided into partitions. A single table may be stored in more than one partition on the basis of a range of data values. Table spaces can be reorganized or recovered independently by DB2.

4. View, a view is comprised of one or more base tables. A view is a logical representation of these tables. It can be a subset of a single table through limiting access to specific rows and or columns. When a view is requested, operations take place according to the definition of the view, and the result is returned to the user as if it were an actual physical table. Views can be created through the Query Management Facility (QMF) support product.

5. Index, an index in DB2 is created based on one or more columns in a table. The index is used to find (locate) the rows in the table that contain the required values of the index. More than one index can be created for a single table, however too many indexes will increase the maintenance requirements of the table(s). There are two types of indexes. A unique index forces the primary key of a table to be unique. For example, if an application tries to update the table so that the primary key has more than one row with the same the update will fail. The second is a cluster index which dictates the physical order of the rows in a DB2 table. A cluster index defines how the rows are physically routed.

6. Index space, indexes are stored in index space just as tables are stored in table spaces. DB2 controls the creation of index spaces internally.

7. Buffer Pools, are temporary storage areas used to hold data in the process of being manipulated. Buffer Pools are used during the sorts, joins and merges of DB2 tables as well as during the updating process associated with each table.

8. Synonyms, are created by the user (programmer) to assign a short easy name to a table or view that has a long name.

9. System Catalog, the DB2 catalog is a collection of DB2 tables that contain
information about those tables. In a SAS environment for example, if you had a SAS data library and could not remember the names of the SAS files in the library you would use the Proc Datasets or Proc Contents to produce a list of the SAS files. In the DB2 environment, you may need to know the names of all tables with a certain value of the first level table name. The following example illustrates the SQL code and the report containing this information:

--- SQL Code ---
Select Name, Example
from SYSIBM, systables
where creator = 'TSID123';

--- Report ---
Name    Example
Emp1    Masterfile
Salary  1991
Tax     1990

The DB2 objects must be created in a top down order. The database must be created first, followed by the tablespace, the tables, and then the indexes.

SAS DATA ACCESS TO DB2 TABLES

Accessing data stored in DB2 using SAS is relatively straightforward by comparison with other languages. When using SAS, the bind process is performed automatically, as is the compile process. Thus, submitting a SAS\DB2 program is operationally no different from submitting any other SAS program.

The DB2 Optimizer determines the data access path based on the current access paths available, the composition of the SQL code and statistics stored in the DB2 system catalog. The choices for access paths include accessing only an index, accessing an index and using this to access columns from its corresponding table(s). The processing can be quite complex with some joins and may involve sorting and the creation of intermediate result tables, although this is transparent to the SAS user, the consumption of computer resources can be quite large.

The DB2 Optimizer does not always select the most efficient access path. For complex SQL statements, those involving large tables, the SAS user should verify the access path prior to submitting the SAS program. Examination of the Plan table will give the user this ability. The Plan table is a DB2 system table containing information about the access path such as the use of an index versus a tablespace scan, names of indexes used, sort requirements, and table access sequence in the case of a join command. The Plan table is created by the SQL: Explain statement and can be executed outside the SAS environment using JCL tailored to your company or can be executed inside the SAS environment using the SAS\DB2 procedures, PROC DB2LOAD or PROC DB2UTIL.

THE SAS\DB2 ENVIRONMENT

Most file structures can be accessed directly by Base SAS in a data step, however Base SAS cannot access a relational DBMS directly SAS\DB2 is the interface between the SAS system and the DB2 environment consisting of three major procedures which can be executed interactively in Display Manager or in line mode and in batch programs.

1. PROC DB2EXT, converts rows and columns from DB2 tables into observations and variables in a SAS data set. Once PROC DB2EXT creates a SAS data set all the capabilities of the SAS system are at the programmers disposal. PROC DB2EXT first builds a map which "maps" DB2 columns to SAS variables. The map includes default SAS variable names consisting of the first eight (8) digit positions of the eighteen (18) position DB2 column name. The maps also assigns standard character and numeric formats to the SAS variables, thus constructing the appropriate SQL select statements. The select statement extracts the
required data from the DB2 table(s) and
the SAS data set is created. The map
is stored as a SAS data set and is usually
a permanent SAS data set for use by
subsequent programs.

MAP SAS
DATA SET (PDS) FILE

DB2 TABLES --> MAP --> SAS SELECT

Once a SAS data set is created by PROC
DB2EXT, other SAS products can be
used in conjunction with SAS\DB2
such as, SAS\GRAPH, SAS\FSP, and
SAS\AF.

2. PROC DB2LOAD, creates and loads a
DB2 table using the data stored in a SAS
data set. It can also be used to execute
utility type SQL statements such as
dropping a DB2 table or creating an index.
Proc DB2LOAD first builds a map which
"maps" SAS variables to DB2 columns.
The map will include default DB2 column
names and data types which must be
modified to your company's environment.
The default column names are SAS
variable names. The default DB2 data
types are CHAR for SAS character
variables and FLOAT for numeric
variables.

The map then constructs the appro-
priate SQL statements which include the
Create, Insert and Commit statements,
thus creating or updating the DB2 table.
The mapping SAS data set can be saved
for use in subsequent programs. PROC
DB2LOAD can also be used to execute
utility type SQL statements such as
dropping a DB2 table or creating an index.
These statements are typically performed based on data
stored in a SAS data set. For example, if rows are to be inserted in a DB2
table, the SQL INSERT statement will
be used if rows are to be deleted from
a DB2 table, the SQL DELETE state-
ment is used. PROC DB2UTIL first
builds a map which "maps" SAS
variables to DB2 columns so that the
updates are applied to the correct
columns, a SQL "where" clause is
usually used when deleting or up-
dating so the SAS observations are
matched with the proper DB2 rows.
The map then constructs the appro-
priate SQL statements to update the
selected DB2 table(s). The map or
SQL statements can then be saved
and used as input to subsequent ex-
cections of PROC DB2UTIL, and the
map is stored as a SAS data set.

MAP SAS
DATA SET (PDS) FILE

SAS DATA SET --> MAP --> SQL STATEMENTS
--> DB2 TABLE

3. PROC DB2UTIL, can be used to insert
rows, delete rows, update column
values or add new columns to a DB2
table and execute utility type SQL state-
ments such as creating an index or
dropping a table. These functions
(insert, update, delete, alter, commit)
are typically performed based on data
stored in a SAS data set. For example,
if rows are to be inserted in a DB2
table, the SQL INSERT statement will
be used if rows are to be deleted from
a DB2 table, the SQL DELETE state-
ment is used. PROC DB2UTIL first
builds a map which "maps" SAS
variables to DB2 columns so that the
updates are applied to the correct
columns, a SQL "where" clause is
usually used when deleting or up-
dating so the SAS observations are
matched with the proper DB2 rows.
The map then constructs the appro-
priate SQL statements to update the
selected DB2 table(s). The map or
SQL statements can then be saved
and used as input to subsequent ex-
cections of PROC DB2UTIL, and the
map is stored as a SAS data set.

MAP SAS
DATA SET (PDS) FILE

SAS DATA SET --> MAP --> SQL STATEMENTS
--> DB2 TABLE

ORIGINAL

DB2 TABLE

INTERACTIVE VERSUS
BATCH PROCESSING

Interactive SAS\DB2 is panel driven
through Display Manager and batch process-
ing is driven through JCL statements. Your
company standards can determine which
processing method is used.

205
In interactive mode PROC DB2EXT has three (3) primary panels. They are: the Data access (Input/Output) definition panel, the Data Extraction (Build Map) panel and the SQL Entry/Edit (Edit a select SQL statement) panel.

PROC DB2LOAD contains four (4) primary panels. They are: the data access definition panel, the table create/load (select SAS variables, name DB2 columns, specify DB2 data types) panel, the Browse (browse mapping SAS data set created) panel, and the SQL Entry/Edit (code or include SQL statements) panel.

PROC DB2UTIL contains seven (7) primary panels. They are: the data access definition panel, the table modification (add new column to DB2 table) panel, the table mapping (map SAS variables to DB2 columns) panel, the table display (list DB2 columns) panel, the where entry (map SAS observations to DB2 rows and specify update criteria) panel, the browse panel, and the SQL Entry/Edit panel.

In batch processing the primary SQL statements used by SAS users is the "select" statement, which extracts data from the DB2 tables and places them in a SAS data set(s). On PROC DB2EXT, the input JCL is a SAS data set, a map, or SOL select statements. Both the map and the select statement can be together in the same JCL as long as they are both compatible.

In PROC DB2LOAD, the DB2 table is loaded and the JCL will contain as input a SAS data set, a map, or SOL select statements. Both the map and the select statement can be together in the same JCL as long as they are both compatible.

In PROC DB2UTIL, the DB2 table is updated and the JCL will contain as input a SAS data set, a map, or SOL select statements. Both the map and the select statement can be together in the same JCL as long as they are both compatible.

In PROC DB2LOAD and DB2UTIL release 5.16 of SAS/DB2, missing values in SAS variables can not be loaded into a DB2 column that was created as "not null". In release 5.18 of SAS/DB2, missing numeric values are loaded as zero (0) and missing character values are loaded as blanks when the DB2 column is defined as "not null". In release 5.16 or 5.18 missing values in SAS variables are loaded as "null" if the DB2 column is defined as "null".

STRUCTURED QUERY LANGUAGE
SQL AN OVERVIEW

SQL is a comprehensive language used to define, manipulate, and control DB2 and other relational DBMS. SQL statements can be executed through SAS/DB2 and batch programs using JCL tailored to your company environment. Within the DB2 environment SQL can be executed through QMF (Query Management Facility), a report writer tool and "SPUFI" (SQL Processor using File Input), a product used for testing SQL statements. SPUFI is a subset of an IBM product called DB21. DB21 can also be used to execute DB2 utilities and prepare programs, such as COBOL and SAS with embedded SQL statements for execution in batch.

SQL statements are grouped into six (6) categories according to the function they will perform in the SAS/DB2 environment.

1. The Data Definition Language (DDL) controls the creation and maintenance of DB2 objects, storage groups, tables, table spaces, indexes, views, synonyms and data bases. These SQL statements are the Create, Alter and Drop statements.

2. The Data Manipulation Language (DML) is the primary component of SQL required by the SAS users. It is comprised of four (4) SQL statements which are the Select, Insert and Delete statements.

3. The Data Control (DC) component controls the operational aspects of the DB2 updating process and these SQL statements are the Commit, Lock, and Rollback statements.
4. The Security Component controls the type of authorization (read, create, update) to DB2 objects such as tables, table spaces, storage groups, data bases, and indexes. The SQL statements are Grant and Revoke.

5. The Language Access component is used only when SQL code is embedded in SAS, COBOL or other programs for batch processing. These statements are the Open, Close, Declare, Fetch, Include, Describe, Whenever, Prepare, and Execute statements.

6. The other access component is used for single purpose SQL statements such as the Explain statement. The Explain can be executed outside the SAS environment using JCL or executed inside the SAS environment using PROC DB2LOAD or DB2UTIL. Everytime an Explain statement is executed a "plan" is added to your DB2 Plan Table. A Plan Table is a DB2 table containing information about the access path such as use of an index, sort requirements and access sequence for joins. Remember that the DB2 optimizer does not always select the most efficient access path for complex SAS\DB2-SQL statements and large tables, thus we use the Explain statement to set the proper access path for our SAS\DB2-SQL application.

THE SAS\DB2-SQL DATA MANAGEMENT TOOL

The SAS\DB2-SQL Data Management Tool is a product that develops and enforces SAS\DB2 standardization within your company environment for each application. This product is comprised of five (5) modules which can help assist both the data base administrator and the SAS\DB2 programmer in the development and maintenance of SAS\DB2-SQL applications.

1. The Repository module, allows data element standardization, enforces corporate naming standards, defines the relationship between logical and physical data structures, allows for the definition of entity and entity relationships, allows mapping of the SAS presentation of data elements on a screen or report including any validation required or special processing requirements, and provides an impact analysis showing the effect of a proposed change to a SAS map or a DB2 object.

2. The Data Definition (DDL) module, shows all references to other objects which have been validated, DB2 tables can be created from a repository or other tables, indexes built by selecting columns from the base table, space requirements automatically calculated from data entered parameters, and the alter table function is included.

3. The Data Manipulation (DML) module, builds SQL statements for all or selected columns of a table, view or map. The DCLGEN builds SAS workareas for host variables and null indicators using a standard naming convention, and the EXPLAIN has been automated.

4. The DB2\SQL\SAS Utilities module, creates JCL for DB2 and SAS utilities generated using user selected criteria and information from the DB2 catalog, the copy and merge JCL are created, all recovery options have been implemented, including automatic validation for tables, table spaces, and databases involved in referential integrity, all options of the Runstats utility are available, and load and unload JCL may be automatically built and includes an option to unload selected columns of a table.

5. The SAS\DB2 Catalog Queries, include results that may be displayed on the screen or directed to a printer or file, since the DB2 catalog is the information source, the lists are always current.

For a demo disk, please send company name, address and phone number to:

Adamson & Associates, Inc.
1 Las Brisas Boulevard,
Voorhees, New Jersey 08043
(609) 346-4084