

Is the SAS® System a Database Management System?

William D. Clifford, SAS Institute Inc., Austin, TX

ABSTRACT

Commercial Database Management Systems (DBMSs) provide applications with fast access to large quantities of data. In addition, many have other capabilities such as data integrity services, data sharing, application-creation tools, and report writing. Version 6 of the SAS® System also contains a number of similar features.

This paper examines the database features of the Version 6 SAS System and compares them to the services offered by several popular DBMSs. The conclusion is that the SAS System can provide a cost-effective alternative to a commercial DBMS for the storage of data.

INTRODUCTION

Database Management Systems have been available for more than two decades and are frequently used as a repository for data. The applications that use this data are often not part of the DBMS and are either purchased from another vendor or developed by the user.

The SAS System is widely used as an application for data analysis. The data may come from a variety of repositories, including a number of DBMSs.

A definition of a DBMS is offered to use as the basis for answering the question posed in the paper's title. An inventory of features found in current DBMSs is provided and this inventory is compared to the DBMS features found in the SAS System.

With this background, an answer to the question of whether or not the SAS System is a DBMS is given. More relevant, however, than the name you call your data repository are the features you really require from it.

An argument is made that the data management facilities in the Version 6 SAS System have matured sufficiently so that it is a viable candidate for your data repository.

Finally some of the DBMS features planned for future releases of the SAS System are identified.

WHAT IS A DATABASE MANAGEMENT SYSTEM ?

A DBMS is a software package that provides a repository for computerized data. The DBMS is responsible for storing the user's data in the repository and making it available upon demand. Users of the data are shielded from the details and peculiarities of the computer software and hardware by the DBMS. That is, a DBMS separates the application from the data. This separation is a key point and will be discussed in more detail.

A database is the term used in this paper for a logical collection of data managed by a DBMS. The terms record, row, and observation are synonyms as are column, field, and variable.

Data Separation

The objective is to separate the application from the data so that the application can focus on the external or logical aspects of the data such as analysis and presentation. The DBMS focuses on managing the internal or physical aspects of the data such as the type and quantity of storage devices and the bookkeeping necessary to support the data model.

As an example (in a relational data model), the application sees the data as rows and columns. The DBMS translates its internal storage structures into these rows and columns.

The fundamental responsibility of the DBMS, once the data are in the database, is to deliver the data back to an application. Query, selection, and update facilities are manifestations of this responsibility.

Another benefit of data separation is data sharing. Once a database is created, its data can be accessed by multiple applications.

Data Model

The data model defines the relationships that exist among the various data items in the database. Some examples of relationships are:

- field owned by a record

- child record owned by parent record
- physical order of records.

The Database Management System is responsible for supporting the relationships specified by the data model. Prior to DBMSs, this was the application's responsibility.

- Earlier DBMSs made the relationships static when the database was created. The specific relationship was the main focus of these DBMSs as evidenced by the data model they supported. Examples are hierarchies and networks.
- Newer DBMSs allow some of the relationships to be specified dynamically. Their focus is also on the relationships, but in a general, flexible sense instead of a specific, rigid sense. A DBMS that supports the relational data model is an example.

Beyond the Basics

Advancements in computer technology (e.g., more power, lower cost) placed additional burdens on DBMSs (e.g., user-friendly interfaces, improved performance). This brought demand for additional features from the DBMS.

As keepers of the data, DBMSs were required to solve these problems. Automatic query optimization, integrity constraints, high speed transactions, and point-and-click interfaces are a partial list of solutions provided by the DBMS vendors.

Although most DBMSs today have a variety of data presentation and analysis services, such features are not relevant to this discussion. Our focus here is on the storage and management of data.

FEATURES FOUND IN CURRENT DBMSs

In this section, features found in present-day DBMSs are identified. There may not be industry-wide agreement on the categories or definitions used here. This section is intended to serve as a general overview of the facilities available, not a comprehensive survey.

The features are divided into two general categories, basic and advanced. The basic features reflect the core functionality of a DBMS: data separation and data relationships. The more

advanced features are built upon the basic ones and reflect additions required by users to keep up with advancements in computer technology. There is no significance to the order of presentation.

Examples of components in Release 6.08 of the SAS System are included with the description of each DBMS feature. The examples used here are not intended to be an exhaustive list of such components of the SAS System.

Basic

file management

To create, populate, delete, and backup databases.

Examples of file management services in the SAS System are the DATA step and the COPY, CIMPORT, CPORT, and SQL procedures.

data inventory services

To list and display information about the existing databases.

The DATASETS and CONTENTS procedures provide data inventory services in the SAS System.

query processing

To retrieve the stored data, including data filtering, that is, selection and projection.

The DATA step, SCL, the WHERE clause, and the PRINT, SQL, REPORT, and FSBROWSE procedures provide query processing in the SAS System.

update processing

To change existing data in a database and add new data.

The DATA step, SCL, the SQL, APPEND, and FSEDIT procedures can be used for update processing in the SAS System.

relational data model

To provide support for the data model that is most popular for new applications. (However, this is not a requirement for a system to be a DBMS.)

SAS data sets are composed of rows (observations) and columns (variables), and thus are relational tables. The SQL procedure implements the de facto industry

standard data manipulation language for the relational model.

file-level security

To grant or deny a user's access to an entire data file.

All host-level file security features are honored by the SAS System. In addition, data set passwords to control read, write, and utility access can be defined.

provide data in sorted order

To physically store the data in sorted order, or to sort data temporarily before they are returned to the application.

The SORT procedure and BY processing can be used to return data to the application in sorted order.

Advanced

row-level security

To grant or deny a user's access to a single row.

The SQL procedure can be used to define views with a WHERE clause to restrict a user's access to certain rows.

portability of applications

To facilitate the movement of applications and data to different platforms.

The MultiVendor Architecture™ of the SAS System is designed to provide portability of applications across heterogeneous platforms.

automatic query optimization

To allow the DBMS to determine the most efficient method of obtaining the requested data. This may include the use of auxiliary data structures such as indexes and hash tables.

Applications can create indexes for SAS data sets that will automatically be considered for WHERE clause optimization. The SQL procedure will also use appropriate indexes for join optimization.

multiple users access to data

To permit multiple users to query and update the same database concurrently. The DBMS,

not the application, is responsible for preventing data corruption by coordinating access to the data.

The SAS/SHARE® software product is designed to permit multiple users to read and update the same data set concurrently. The data sharing is transparent to the application.

row-level locking

To allow data sharing by row. This means multiple users can query and update a given database concurrently as long as they do not request the same row. File-level locking, by contrast, permits only one user access to the file at a time.

The SAS System supports row-level locking of a single row in a data set within SAS/SHARE software and for multiple opens of the same data set in a standalone environment.

Integrated data dictionary

To provide a database of information, maintained and used by the DBMS, containing data (meta data) about all the databases managed by the DBMS.

Currently the SAS System does not have an integrated data dictionary. SAS/EIS® software supports a non-integrated metabase.

non-integrated integrity constraints

To support data validation checks performed by the application.

The SAS applications programmer can use Informats and write validation code in the DATA step, SCL, and the AF and FSP procedures.

Integrated integrity constraints

To support data validation checks in a multiple user/application environment. These checks are performed automatically by the DBMS for all applications. Non-integrated data validation techniques can be applied to this environment.

Currently the SAS System does not support integrated integrity constraints

audit trail

To maintain a time-stamped log of what user made a given update, including the new data values.

No integrated audit trail currently exists for the SAS System. For a given application, the DATA step and SCL support user-written schemes for collecting such data.

rollforward

To permit the recovery of a lost or damaged data set by the application of updates from an audit trail to an archived copy of the database.

The SAS System currently does not support a rollforward mechanism. For a given application, the DATA step and SCL support user-written schemes for collecting such data.

transactions with rollback

To logically bind multiple updates into a single atomic update. That is, either all the updates are successfully applied to the database or none of them are applied. Rollback initiates the removal of pending updates in the atomic unit.

Currently there is no support for transactions in the SAS System

high volume transactions

To provide very fast response time to a large number of requests, also known as On-Line Transaction Processing (OLTP). Here performance is of key importance. The environment is usually highly interactive with many users. An example is an airline reservation system.

The SAS System has been tailored for fast sequential processing, and therefore is not well-suited to this type of application.

distributed data/distributed processing

To support an environment with applications and data on separate platforms. A given database will reside entirely on a single platform.

SAS/CONNECT® software allows an application to access data from a different platform, and it permits the application to execute on another platform. SAS/ACCESS® software supports access to data on other platforms in some environments.

distributed databases

To store parts of the same database on different platforms.

There is no support in the SAS System for distribution of a single data set across different platforms.

IS THE SAS SYSTEM A DBMS?

If you use the historical definition of a DBMS as a data repository that provides separation of data and applications, then the SAS System is clearly a DBMS.

If you choose a more contemporary definition of a DBMS, then the SAS System falls somewhat short of being a DBMS. It has a number of features found in many commercial DBMSs, but it does not have all of them.

However, this question is really academic. A better question is "What specific requirements do you have for your data repository?" If you have an OLTP environment, the SAS System will probably not satisfy your performance requirements. An Information Database environment that depends upon lots of rapid sequential access to the databases, is likely to find the SAS System's performance very good.

WHERE SHOULD YOU STORE YOUR DATA?

DBMS vendors position their product as a data repository. The applications that use the data are usually not provided by the DBMS vendor. The SAS System is positioned as a data analysis and information delivery system. That is, the SAS System is the application that uses the data.

The SAS System has facilities to access data in many different formats and repositories as has been mentioned earlier. Given that you want to process/analyze your data with the SAS System, then the question here is not access to the data but where the data are to be permanently stored.

There are three basic choices for the data repository: flat/unstructured files, a commercial DBMS, or the SAS System. And there are SAS applications and non-SAS applications. With these variables, let's define six simple models:

model	primary application	data repository
1	non-SAS	flat file
2	SAS	flat file
3	non-SAS	DBMS
4	SAS	DBMS
5	non-SAS	SAS System
6	SAS	SAS System

The first two models are quite reasonable and common uses of flat files as data repositories. The SAS System, via the DATA step, has powerful facilities for accessing a wide variety of flat file formats.

Models 3 and 4 are the traditional ones with a DBMS as the data repository and non-DBMS applications as consumers of the data.

In a model 5 environment, the DATA step can provide the data to applications in a wide variety of flat file formats when the original data cannot be read by the applications. The DATA step can produce multiple different flat files, one for each of the different applications. While stored in SAS data sets, the data can be edited (to repair invalid values) and subtyped prior to delivery to the applications.

The main premise of this paper is that model 6 is a viable model and should be carefully considered when deciding upon a data repository for SAS applications. The choice between model 4 and model 6 should be based upon the features you require from your data repository.

Version 6 of the SAS System lacks some features found in commercial DBMSs as has been described previously. If you do not have any of these requirements for your data repository, then you should seriously consider using the SAS System.

The benefits of using the SAS System for the storage of your data include:

- faster access to the data for SAS applications. The SAS System is optimized to deliver data to its own procedures.
- more cost-effective solution. You don't have the added expense of a DBMS.
- a reduction in the number of vendors involved. Using the SAS System for both

data analysis and data storage will eliminate the need for maintenance and system upgrades to another product (the DBMS), and it will provide a single source for problem resolution. Compatibility issues between different versions of the application software and the DBMS software will not exist.

- product consistency across many platforms. The MultiVendor Architecture (MVA)[™] of the SAS System provides a portable applications environment independent of the host computer system. There is only one language to learn. SAS applications developed on one platform will run on other platforms. Data can be shared across different platforms. Your data and applications are not tied to a particular computer system.
- the ease of transferring data to non-SAS applications. In many cases, the flexibility of the SAS System for this purpose exceeds that of a traditional DBMS. While most DBMSs do have an export feature, the length and data types of the exported data are often fixed. The DATA step allows you to output flat files exactly as you want them, or as the next application needs them. In fact, the SAS System data management capabilities are often used just to massage data between applications.

FUTURE DIRECTIONS FOR DBMS FEATURES OF THE SAS SYSTEM

The features listed below are under consideration for some future release of the SAS System. No details are given as the research and development is in progress and numerous issues remain to be resolved.

- audit trail, with optional rollforward
- integrated integrity constraints, including referential integrity
- integrated data dictionary
- rollback, multiple record locking, and transactions
- improved distributed data access (libname on different host)

The goal of these efforts is to expand the DBMS services the SAS System offers you, not to displace current DBMS products in the marketplace. There is considerable use of the DBMS features that have already been implemented and strong interest in those that are on the drawing board.

CONCLUSION

It may be difficult to agree on the exact definition of a DBMS and whether or not the SAS System satisfies that definition. However, it should be clear that the SAS System does support many features found in current DBMS products, and in some cases provides more functionality. In future releases, additional DBMS functionality will be added to the SAS System.

When you are making a decision about what repository to use for your data, the SAS System is a serious candidate. It's the functionality that comes with the product, not the product's classification, that's important.

SAS, SAS/ACCESS, SAS/CONNECT, SAS/EIS, SAS/SHARE, MultiVendor Architecture, and MVA are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.