CHAPTER 2

ADABAS Essentials

Introduction

This chapter introduces SAS System users to ADABAS, Software AG's database management system (DBMS). The chapter focuses on the terms and concepts that will enable you to use the SAS/ACCESS interface to ADABAS, which include:

- the ADABAS DBMS and ADABAS databases
- ADABAS files, NATURAL Data Definition Modules, and ADABAS descriptors (indexes)
- ADABAS data fields and ADABAS and NATURAL data formats and lengths

- null (missing) values
- ADABAS Security and NATURAL SECURITY System options.

If you want more information on an ADABAS concept or term than this chapter provides, see the ADABAS information on your system.

ADABAS DBMS

ADABAS is Software AG's *database management system* (DBMS). ADABAS organizes and accesses data according to relationships among *data fields*. The relationships among data fields are expressed by *ADABAS files*, which consist of *data fields* and *logical records*.

With the ADABAS DBMS, you can also use the high-level language NATURAL to operate on data that is managed by the DBMS. NATURAL is Software AG's fourth generation application development system that allows you to create, modify, read, and protect data that the DBMS manages. All ADABAS files and data fields referenced in a NATURAL program must be defined to NATURAL through a Data Definition Module (DDM).

ADABAS has single-user and multi-user execution environments, both of which are supported by the SAS/ACCESS interface to ADABAS.

ADABAS Databases

An ADABAS database is a collection of data organized into ADABAS files. Each database has an associated *database identifier*, which is a numerical value in the range 1 to 255, and a *database name*, which is a character value with a maximum of 16 characters. Each database can consist of up to 255 ADABAS files.

An ADABAS database consists of three system files: Data Storage, Associator, and Work Storage.

- The Data Storage system file contains the actual data records for all ADABAS files in a database, in compressed form.
- The Associator system file contains internal storage information that manages the data for the entire database.
- The Work Storage system file contains temporary work files.

To use the SAS/ACCESS interface to ADABAS, you need to be familiar with three ADABAS components: ADABAS files, NATURAL DDMs, and ADABAS descriptors (which is an ADABAS data field that provides an index of its values). ADABAS files and NATURAL DDMs are the components from which you create SAS/ACCESS access descriptor and view descriptor files. Knowing about ADABAS descriptors can help you minimize ADABAS's processing time for your SAS/ACCESS view descriptors.

Note: To avoid confusion, keep in mind the two usages of the term *descriptor* throughout this book:

- An *ADABAS descriptor* is an ADABAS data field that provides an index of the data field's values.
- *SAS/ACCESS descriptor files*, on the other hand, are the files used to establish a connection between the SAS System and ADABAS.

△

The following sections describe ADABAS files, NATURAL DDMs, and ADABAS descriptors.

ADABAS Files

An ADABAS file is a collection of logically related data, organized by data fields and logical records. ADABAS permits maximums of 926 data fields and 16,777,215 logical records in each ADABAS file.

Output 2.1 on page 9 illustrates four data fields and seven logical records from an ADABAS file containing data about customers. The data fields are the vertical columns of data. The logical records are the horizontal rows of data.

Output 2.1 Sample ADABAS File

```
CU          CI              ST        CO

14324742    San Jose        CA        USA
14569877    Memphis         TN        USA
14898029    Rockville       MD        USA
24589689    Belgrade                  Yugoslavia
26422096    La Rochelle               France
38763919    Buenos Aires              Argentina
46783280    Singapore                 Singapore
```

ADABAS files are created with the ADABAS utility ADACMP. (To see the ADABAS data definition statements that created the ADABAS files used in this book, refer to Appendix 3, “Example Data,” on page 137.)

ADABAS File Number

When you create an ADABAS file, you assign a file number using the FILE= statement of the ADACMP utility. Each database can consist of up to 255 ADABAS files.

Level Number

A data field *level number* is a one- or two-digit number, from 01 to 07, used in conjunction with data field grouping. (Grouping is discussed in “ADABAS Data Fields” on page 11.) Data fields with a level of 2 or greater are considered to be a part of the immediately preceding group, which has a lower level number.

Data Field Names

ADABAS data fields are identified by a two-character name. Each data field name in an ADABAS file must be unique. The first character must be alphabetic, and the second character can be either alphabetic or numeric. For example, AA and B4 are valid data field names.

Logical Record ISN

Each logical record within an ADABAS file is assigned an *internal sequence number* (ISN). An ISN is the logical identifier for each record. ISNs are unique within each ADABAS file.

Note: When you create SAS/ACCESS descriptor files for ADABAS data, the ACCESS procedure creates a SAS variable named ISN. This variable gives you access to the ISNs for all logical records stored in the ADABAS file. △

NATURAL Data Definition Modules

To reference an ADABAS file and its data fields in NATURAL programs, you must create a NATURAL Data Definition Module (DDM) based on the ADABAS file. (Note that a DDM is often referred to as an ADABAS file, even though it is really only a *view* of an actual ADABAS file.) A DDM has an assigned name, which references the ADABAS file number that the DDM is based on. Also, more descriptive data field names can be assigned to a DDM. DDMs are stored in a system file, which is simply another ADABAS file.

DDM File Name

The filename for a NATURAL DDM can be a maximum of 32 characters.

Data Field Names

In a NATURAL DDM, data fields can be assigned a DDM external name of 3 to 32 characters. For example, in the CUSTOMERS DDM, the DDM data field name CUSTOMER corresponds to the ADABAS file two-character data field name CU.

ADABAS Descriptors

If you plan to use a data field often in selection criteria, you can designate it as a key field. You designate a key field by specifying the descriptor option in the ADACMP utility data definition statement. When a data field is a descriptor field, ADABAS maintains and stores its values in an *inverted list*. An inverted list contains the different values of a descriptor data field, along with the count and the ISNs of the logical records that contain each value. ADABAS descriptors can also be defined so that inverted lists contain unique values only.

Specifying ADABAS descriptors speeds up the selection process considerably since ADABAS is able to access key values directly. Also, specifying descriptors controls read sequence when reading ADABAS data in sequential order.

Several descriptor types can be specified for a data field. Each descriptor type is explained below.

Note: In order for you to use SAS variables corresponding to ADABAS data fields in a SAS BY statement, an SQL ORDER BY clause, or a view SORT clause, the data field must be designated as an ADABAS descriptor. Regarding a WHERE clause (both view and SAS), there are conditions when you can use a nondescriptor data field and when you must use a descriptor data field. These conditions are explained in Chapter 5, "ACCESS Procedure Reference," on page 59. △

Subdescriptor

A subdescriptor is an ADABAS descriptor that is derived from a portion of an elementary data field. For example, if ZIPCODE is a data field, a subdescriptor for it could be ZIPLAST2 defined for the last two digits of a zipcode.

You can include a subdescriptor in SAS/ACCESS descriptor files for retrieval and selection criteria, but you cannot use subdescriptors in SAS System updating procedures.

Superdescriptor

A superdescriptor is an ADABAS descriptor derived from more than one data field, portions of data fields, or combinations thereof. For example, a superdescriptor named STATE-ZIPLAST2 could be defined for the first two digits from the STATE data field and the last two digits from the ZIPCODE data field.

You can include a superdescriptor in SAS/ACCESS descriptor files for retrieval and selection criteria, but you cannot use superdescriptors in SAS System updating procedures.

Phonetic Descriptor

A phonetic descriptor is an ADABAS descriptor defined to perform searches based on phonetic values, for example, retrieval by family name.

You can include a phonetic descriptor in SAS/ACCESS descriptor files for retrieval and selection criteria, but you cannot use phonetic descriptors in SAS System updating procedures.

Note that if you use a phonetic descriptor in a SAS WHERE clause, the interface view engine must be able to process the entire SAS WHERE clause.

Note: The hyperdescriptor type is not described because hyperdescriptors are not supported by the SAS/ACCESS interface to ADABAS. Your ADABAS file can contain hyperdescriptors, but they will be ignored. △

ADABAS Data Fields

You can group logically related ADABAS data fields into one ADABAS file, which consequently can be accessed by one NATURAL DDM. Up to 926 data fields can be contained in a single logical record. Data fields have assigned types, formats, and lengths.

Data Field Types

The SAS/ACCESS interface to ADABAS supports the ADABAS data fields as described below.

Elementary Field

An elementary field is limited to one value per record. For example, LASTNAME could be an elementary field.

Multiple-value Field

A multiple-value field can have 0 to 191 values per record. For example, JOBTITLE could be a multiple-value field because each employee at a company could have multiple job titles during his or her employment.

Group Field

A group field is several consecutive data fields combined into one for efficient access and ease of reference. Defining a group field allows you to reference a series of data fields by using a group name. For example, a group field named EDUCATION could consist of these data fields: COLLEGE, DEGREE, and YEAR.

A group field can also consist of other groups. In conjunction with grouping, you can assign level numbers 01 to 07 to define a group.

Periodic Group Field

A periodic group field is a group of data fields that repeat. A periodic group can be repeated up to 99 times and can contain one or more elementary fields and multiple-value fields. Groups can be nested, but periodic groups cannot. One periodic group cannot contain another. However, a record can have several different periodic groups.

Subfield

A subfield is a data field defined from a portion of another data field. For example, a subfield named AREA-CODE could be defined for the first three digits from the PHONE data field.

You use subfields for read operations only; they cannot be used for updating directly.

Superfield

A superfield is a data field composed of several data fields, portions of fields, or combinations thereof. For example, a superfield could be STATE-AREA-CODE accessing such values as TX512, NM505, and CA213.

You use superfields for read operations only; they cannot be used for updating directly.

Mapping Data between the SAS System and ADABAS

When you access ADABAS data through the SAS/ACCESS interface, the interface view engine maps the ADABAS data into SAS observations. You need to be aware of how the interface view engine maps multiple-value fields and periodic groups. That is, *multiple-value field occurrences are mapped to multiple SAS variables, and periodic group occurrences are mapped to multiple SAS observations*.

For example, suppose an ADABAS file has the data fields and values shown in Output 2.2 on page 12. LASTNAME is an elementary field, JOBTITLE is a multiple-value field, and EDUCATION is a periodic group consisting of the data fields COLLEGE, DEGREE, and YEAR.

Output 2.2 ADABAS Data

LASTNAME	JOBTITLE	EDUCATION		
Reid	Systems Analyst	Purdue	BA	1973
	DBA	Harvard	MBA	1975

The interface view engine would map the ADABAS data into two SAS observations, as shown in Output 2.3 on page 13.

Output 2.3 ADABAS Data Mapped into SAS Observations

```
LASTNAME   JOBTITL1          JOBTITL2   COLLEGE   DEGREE   YEAR
Reid       Systems Analyst   DBA        Purdue    BA       1973
Reid       Systems Analyst   DBA        Harvard   MBA      1975
```

If you were browsing the ADABAS data, such as with the FSVIEW procedure, the results would be similar to Output 2.3 on page 13, with LASTNAME, JOBTITL1, and JOBTITL2 repeated for each set of COLLEGE, DEGREE, and YEAR values. Actually though, the value Reid is stored in the ADABAS file only once. For retrievals, the results are straightforward. When updating, however, you need to keep in mind how the interface view engine maps multiple-value fields and periodic groups.

Using the FSVIEW procedure, suppose you want to change the spelling of a last name, for example, from Reid to Reed. All you need to do is type REED over one of the REID values, and, with a single update operation, the last names are all changed. On the other hand, using the FSEDIT procedure, suppose you want to delete an observation for Reid. Each observation for his job titles and education data would display his last name. If you deleted an observation, for example, the one for Purdue, the deletion would not affect his last name or his job title data, but the Purdue observation would be gone. For more information and an example of deleting an observation from ADABAS data, see Chapter 4, “Browsing and Updating ADABAS Data,” on page 39.

Data Field Formats and Lengths

Data definition statements allow you to define data field formats and lengths for both ADABAS files and NATURAL DDMs. The standard format of a data field is specified with a one-character code shown next in Table 2.1 on page 13. The standard length of a data field is specified in bytes; the maximum length is also given.

Table 2.1 ADABAS Standard Data Field Formats and Lengths

Data Type	Standard Format	Standard Length	Description
Alphanumeric	A (ADABAS) A (DDM)	253 byte maximum	Left-justified, with trailing blanks removed
Binary	B (ADABAS) B (DDM)	126 byte maximum	Right-justified, unsigned, with leading zeros removed
Fixed Point	F (ADABAS) B (DDM)	Must be 4 bytes	Right-justified, signed, with twos complement notation
Floating Point	G (ADABAS) F (DDM)	Must be 4 or 8 bytes	In normalized form and signed

Data Type	Standard Format	Standard Length	Description
Packed Decimal	P (ADABAS) P (DDM)	15 byte maximum	Right-justified and signed
Unpacked Decimal (Zoned)	U (ADABAS) N (DDM)	29 byte maximum	Right-justified and signed

If the standard length of a data field is specified as zero, the data field is a *variable length field*, which has no maximum or required length.

Note that when creating SAS/ACCESS descriptor files, you can specify SAS formats for ADABAS data to change the way the data appears, for example, to add decimal points. Also, you can specify a SAS date format in your SAS/ACCESS descriptor files to designate a date representation.

Null Values

ADABAS has a special value called a *null* value, which means an absence of information. A null value is analogous to the SAS System's missing value.

You can define data fields to not store null data by specifying the NU option in data definition statements. In normal data storage (that is, NU not specified), a null value is represented by two bytes (one for the value length and one for the null value). Suppressing null values results in a null value being represented by a one-byte empty field indicator. The null value itself is not stored.

Knowing whether a data field allows null values assists you in writing selection criteria and in entering values to update ADABAS data. For example, if the NU option is specified for an ADABAS descriptor data field, null values for the data field are not stored in the inverted list. Therefore, a search using this data field and for which a null value is the search value, would result in no records selected, even though there may be records that contain a null value for the data field.

For more information on null values, see “Missing Values (Nulls)” on page 126.

Security Options

The ADABAS DBMS offers security options through both ADABAS and NATURAL. To protect your ADABAS data, you can use either form of security, or you can have both work together.

ADABAS Security Options

ADABAS provides a security facility to prevent unauthorized access to data stored in ADABAS files. Security is available through password protection and by maintaining data in enciphered form.

passwords — provide protection at the ADABAS file level, data field level, and data value level. These security options are defined with the SECURITY utility ADASCR and are stored in the ADABAS SECURITY system file.

To access an ADABAS file protected by a password, you must provide the valid password. Each data field in an ADABAS file can

be assigned up to fifteen levels of read and update security. A user password specifies the authority for the data field, and ADABAS automatically determines whether the user is authorized to perform the requested operation. If the permission level of a user's password is equal to or greater than the permission level for the file the user is trying to access, access is granted. Any ADABAS file can be protected on individual data field values. In this case, the password specifies value restrictions on logical records to be selected, read, and updated.

cipher codes — are simple numeric codes that you can assign using the ADACMP utility when creating an ADABAS file. Ciphering renders data records unreadable when they are displayed with a non-ADABAS program or utility. You must supply this cipher code in order to access the enciphered data.

Note: System information such as DDM and NATURAL SECURITY information is also stored in ADABAS files; they too can be password-protected or enciphered. △

NATURAL SECURITY

NATURAL provides an optional security system that controls the access and use of the NATURAL environment. You can restrict the use of whole application systems, individual programs and functions, and the access to DDMs.

Security is accomplished by defining objects and the relationships among these objects. There are three objects that you need to be familiar with when accessing data through NATURAL DDMs with the SAS/ACCESS interface: users, libraries, and files.

users — can be people, terminals, or groups of either, with assigned identifiers. The user identifier identifies the user to NATURAL SECURITY and controls user activity during a NATURAL session. The identifier is unique to NATURAL and can be up to eight characters long. Each user identifier can have an associated eight-character password.

libraries — contain sets of NATURAL source programs and/or object modules that perform a particular function, with assigned identifiers. Stored in the library data are the ADABAS passwords or cipher codes to allow NATURAL programs to work with ADABAS Security. The library identifier identifies the library and the ADABAS file it is authorized to access to NATURAL SECURITY. The identifier is unique to NATURAL and can be up to eight characters long.

files — are the NATURAL DDMs based on ADABAS files.

Relationships, called *LINKS*, are defined among these objects. These links define which users are allowed to use a library and which files a library is allowed to access. The users, libraries, files, and links are all stored in the NATURAL SECURITY system file, which can also be protected with an ADABAS password or cipher code since it is an ADABAS file. For example, one user identifier and library may be able to access a DDM for read only, while another user identifier and library may be able to read and update the same DDM.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/ACCESS® Interface to ADABAS Software: Reference, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS/ACCESS® Interface to ADABAS Software: Reference, Version 8

ISBN 1–58025–546–9

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999