

The DBA and Application Selection/Development

This chapter provides an overview of application selection and planning for your database. It covers the following topics:

- Configuration and Applications Planning
 - Database Organization
 - Understanding Current and Future User Requirements
 - Coordinating Database Activities
 - Analyzing Access Requirements
 - Establishing Data Availability
 - Performance Versus Flexibility
 - Advising on Application/Program/Database Design
 - Determining Physical Storage Requirements
 - The Test Database and Testing Strategy
-

Configuration and Applications Planning

From the DBA's knowledge of the use of the database and the monitoring of its performance, the DBA can contribute valuable data-processing expertise for making management decisions in the area of configuration and applications planning. The DBA is aware of the user's short- and long-term needs, as well as day-to-day problems and difficulties. The DBA's contacts with Software AG enable the DBA to keep abreast of the developments intended for the DBMS. The DBA should, therefore, be brought into this type of discussion.

The DBA should be involved in any application development project from the beginning. The DBA will be able to help in the initial survey in order to decide whether a database approach is justified in view of the organization's planned data processing developments.

The DBA will continue to be involved in project development after the initial implementation of the database project(s). The addition of new projects creates special problems which the DBA must resolve carefully.

The addition of new data and a changing use of existing data may change the performance characteristics of an existing system. Careful redesign of the physical structure and placement of data may be needed in order to give a reasonable service to all users.

Database Organization

As mentioned elsewhere, the DBA is responsible for the formulation and definition of the data relationships for the purpose of defining logical data structures. These data structures should reflect the DBA's knowledge of foreseeable developments and include the needs of other related users.

There are two major aspects:

- the definition and organizing of existing data; and
- the addition of new data.

Efficient physical structuring demands considerable expertise in translating and implementing logical relationships. Space, performance and cost must be balanced, taking into account

- data structure (logical data formats and relationships);
- storage structure (physical data formats and relationships);
- access methods (available and to be used);
- frequency of access;
- physical storage media requirements;
- timing considerations; and
- search strategies.

The solution (and the reasoning behind it) should be fully documented.

Understanding Current and Future User Requirements

The DBA is in an ideal position to help the members of a project team appreciate and be aware of the user's current and future requirements. In this instance, *user* is to be interpreted in its broadest sense. It does not only mean the section or department for which the application system is being designed and developed. User also means other potential users, not forgetting the organization as a whole. Known future developments must be taken into account. At times, this may mean that the DBA will need to exercise control over the development of a new application, in order that these developments may be readily included in the database operation when completed.

Coordinating Database Activities

Providing that the contents of this documentation are put into practice, the DBA will be able to coordinate all database activities. The DBA's advice should be sought on all developments planned for the database and the DBA should aim to ensure a steady, controlled progression to an integrated information system, which will serve the organization as a whole.

In general, the DBA should be involved in all stages of a new project from feasibility study onwards, both in order to advise on the practical uses of the database and also to carry out the DBA's quality control function.

The DBA will, therefore, provide lines of communication between different project teams, as well as with present and future users. The aim should be to cultivate the attitude of designing the database for the greatest benefit of all users.

Analyzing Access Requirements

This is an important part of the design of the database. When new projects reach the data analysis and file design stage, it is important for the DBA to ensure that the project team does not take too parochial a view of the requirements for access to the data to be used by the new application system.

The analysis of access requirements is also an ongoing task. As the requirements of the organization change (as they are bound to do with time), the DBA will be receiving feedback on these changing requirements. However, the DBA should be careful not to overreact to a new requirement; it may only be required this one time. Rather, the DBA should respond to gradual and perceptible changes of emphasis in the access and/or processing requirements of the organization and even then, only after full discussion with all the affected parties.

Establishing Data Availability

The DBA should assist the project team (possibly by using the data dictionary) to plan a suitable data acquisition program, ensuring that the following aspects are taken into account:

- The present form and location of the data;
- How it is to be collected;
- How accurate and complete it is at present;
- What modifications are required to be made to the data before its inclusion in the database;
- At what time should the data be collected in relation to the implementation of the application system. How is the intervening period to be handled?

This process will result in a data collection program, which will include the necessary specifications for any special editing or validation that may be required, as well as providing information to the DBA for recording in the data dictionary.

Performance Versus Flexibility

The design of a (part of the) database will naturally involve consideration of performance (in the sense of disk space utilization and computer processing time), as opposed to flexibility (the ease of adapting to future unknown needs).

The DBA should ensure that the project team does not opt for performance at the expense of flexibility, and vice versa. The DBA is in a good position to advise the project team on which areas of the application need to be flexible (i.e., a planned system will also use this data) and which should be designed for good performance. The ultimate aim of such considerations of performance versus flexibility is to avoid making decisions where only one of these aspects is considered at the possible expense of the other.

Advising on Application/Program/Database Design

From the DBA's contact with project teams, with other Adabas users, and with Software AG, the DBA gradually acquires considerable knowledge about application program and data structure design. The DBA circulates appropriate information throughout the organization. In addition, the DBA must be available to advise on application design.

Although the DBA may not actually design the database, he/she must be able to advise team members on file/record design, descriptor selection, and other matters. This provides the DBA an opportunity to represent other users in the database design.

The DBA must ensure that the design of the physical database will efficiently support the logical requirements of the first application without prejudicing the success of later projects. The DBA will advise on how Adabas should be used in order to fulfill the security, integrity and recovery needs of the application system, design rules and procedures for these. In some cases, additional software may be needed and the DBA will help to design this. The DBA will consider whether additional utilities are needed for saving/restoring the database, measuring performance, analyzing the actual content of the database, and so on.

During this phase, the DBA may also be advising application project teams on the best approach to data analysis, the use of Adabas, how to design the logical data structure and which design options are likely to prove to be the most efficient.

Determining Physical Storage Requirements

It is the DBA's responsibility to provide assistance to the project team in determining the physical storage requirements for a particular application system.

The following parameters should be considered:

- The volume of data to be stored;
- The anticipated growth of data;
- The average size of records;
- The number of additions and deletions of records over a given time period;
- Data relationships (data structure);
- Data representation (internal formats);
- The effect of compression;
- The access methods which are to be used on the data.

Evaluate carefully the tradeoffs between minimizing the use of storage media and processing costs while maximizing service (measured in terms of speed or throughput). Also consider the need for flexibility in the implementation of the application system; requirements may be subject to change, and other application systems may need access to this application's data. If, minimizing physical storage requirements means a loss of flexibility, it should not be done without careful consideration of the problems that may arise in the future. The DBA is, of course, ideally placed to provide the project team

with this type of information.

The Test Database and Testing Strategy

The DBA should advise the project team on the type of test database to be used for the new application. Assist the team by setting up the test database. During system testing, test with your own monitoring, audit, error correction, and control procedures before the system goes live. These procedures should not be designed after the system has gone into production; they should be developed by the DBA in parallel with the development of the application system.

It is best to keep test databases separate from the production database by loading them onto separate disk packs, or even databases. This, in itself, poses two problems:

- Tests cannot take place in parallel with production work in multiuser mode (in single-user mode, this problem does not exist);
- Production data (or some fields in a production file) may be required to test the new system.

The first of these problems could be solved (if storage permits) by running two copies of the Adabas nucleus in parallel—one for production work, one for test work.

The second can be solved by using the Adabas ADASAV utility to copy across the required data from the production database to the test packs. In this case, the access authorization for the test data will have to be agreed before testing begins.

The main advantages of having a separate test database are that files can be loaded with the file numbers they will have when the system goes live and testing can in no way corrupt the production database. This is a particularly important consideration when fields are to be added to an existing file or new descriptors are to be established for the new application system.

Before systems testing starts, the DBA should decide how file conversion and database initialization is to be accomplished and ensure the preparation of any necessary special conversion or set-up programs. The strategy for parallel running will need careful consideration and here, too, special programs may be needed to assist in the comparison of outputs from the existing system and the new system or to carry out validity checks. Special inquiry facilities (e.g., Natural) may be needed to help testing and parallel running (these have sometimes also been found useful in subsequent live running).

Before the new database is finally implemented, acceptance tests should be run to demonstrate that all aspects of the system, including performance and resilience, are satisfactory. These may or may not be additional to the parallel runs.

Close control of the way in which the new project accesses data will be necessary in order to ensure that there is not loss of data integrity for existing users of the database. For systems testing, a special testing mode may be needed in order to ensure that test changes to the database do not actually affect the operational database.