

GH20-1626-1

Program Product

**Data Base
Design Aid
General Information Manual**

IBM

Program Product

Data Base Design Aid General Information Manual

Program Number 5748-XX4

This book, a general description of the Data Base Design Aid, is intended primarily to help evaluate the product and may also serve as a guide in planning the implementation of DBDA.

The benefits and features of the product are presented, followed by highlights of the data base design process, some problems and how DBDA can solve them, a description of the product, its environment, and related IBM productivity aids.

IBM

PREFACE

This book gives a general description of the Data Base Design Aid system. It is intended primarily to help customer executives and system analysts evaluate the system, and it may also serve as a guide in planning the implementation of the Data Base Design Aid system.

- Section 1. Scope and Benefits of Data Base Design Aid

defines DBDA and the user environment, and presents the benefits and features of DBDA.

- Section 2. The Data Base Design Process

describes six steps in the process of designing a data base, and discusses the criteria to be considered.

- Section 3. Solving Design Problems with DBDA

explains some of the problems encountered in designing data bases, and DBDA's solution to them.

- Section 4. DBDA Program Description

describes the DBDA phases, gathering and recording input, and the design, editing, and diagnostic reports. This section shows the relationship of an application's data requirements to the DBDA Requirement Specification Form.

- Section 5. Program Environment

contains information about operating systems and hardware systems used by DBDA.

- Section 6. Related IMS Productivity Aids

describes additional Information Management System (IMS) productivity aids that can be helpful in the design and evaluation phases of data base design.

Second Edition (May 1975)

This edition applies to Version 1, Modification Level 0, of the program product Data Base Design Aid (S748-XX4) and to all subsequent versions and modifications until otherwise indicated in new editions or Technical Newsletters.

Changes are continually made to the information herein. Therefore, before using this publication, consult your System/360 and /370 Bibliography (GA22-6822) for the editions that are applicable and current.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments has been provided at the back of this publication. If this form has been removed, address comments to: IBM Corporation, Technical Publications/Industry, Dept. 825, 1133 Westchester Avenue, White Plains, New York 10604. Comments become the property of IBM.

CONTENTS

SECTION 1. SCOPE AND BENEFITS OF DATA BASE DESIGN AID	1
SECTION 2. THE DATA BASE DESIGN PROCESS	3
Gathering Requirements	3
Generating a Structural Model.	3
Constructing a Physical Model.	4
Design Evaluation.	4
Physical Implementation.	4
Performance Evaluation	5
SECTION 3. SOLVING DESIGN PROBLEMS WITH DBDA.	7
The Problem of Data Base Design.	7
DBDA's Solution of the Problem	8
SECTION 4. DBDA PROGRAM DESCRIPTION	9
DBDA Input	9
Gathering and Recording Input.	12
A Data Requirements Example.	12
DBDA Output.	17
DBDA Design Reports	17
DBDA Edit/Diagnostic Reports	21
Optional Interface to IMS Dictionary System.	22
Iterative Use of DBDA.	22
SECTION 5. PROGRAM ENVIRONMENT.	23
Operating Systems.	23
Hardware Systems	23
SECTION 6. RELATED IMS PRODUCTIVITY AIDS	25
IMS DICTIONARY SYSTEM.	25
DBPROTOTYPE.	25
DBPROTOTYPE/VIS	25
DCANALYZER	27
DC MONITOR	27
IMSMAP and IMSMAP/VIS	28

FIGURES

Figure 4-1.	DBDA Phases	10
Figure 4-2.	Relationship of Data Requirements, Application Programs, and Integrated Data Base.	11
Figure 4-3.	DBDA Requirement Specification Form	13
Figure 4-4.	Language Lab Result Report.	14
Figure 4-5.	Language Lab Schedule List.	16
Figure 4-6.	Association Weights Report.	18
Figure 4-7.	Suggested Segments Report	19
Figure 4-8.	Structural Model Report	19
Figure 4-9.	Candidates for Secondary Indexes.	20
Figure 4-10.	Parent/Child Graph.	20
Figure 6-1.	Relationship of Productivity Aids to Some Steps in Data Base Design	26

SECTION 1. SCOPE AND BENEFITS OF DATA BASE DESIGN AID

The Data Base Design Aid (DBDA) is a productivity tool that automates a significant part of the data base design process. In addition, a DBDA option for IMS users allows interim or final design information to be stored in the IMS Dictionary System (Field Developed Program 5798-BBA).

DBDA is aimed at current or potential users of IBM data base products that include IMS/360, IMS/VS, DOS DL/I, and DOS DL/I entry level. Although DBDA is valuable especially as a design aid for DL/I hierarchical data structures, it can also help non-DL/I users with their data base design by analyzing data requirements for completeness, redundancy, and consistency, and by presenting a structural model that shows the relationships between the data elements.

The benefits of using DBDA are:

- Faster Data Base Design

By automating analysis of the data requirements, the time required for generating the structural model can be reduced significantly.

- Better Design Quality

DBDA performs a more thorough analysis of the data requirements than is normally possible with manual methods, and this results in the increased likelihood of attaining a consistent and effective design.

The benefits incorporate these features:

- Standardized Gathering of Data Requirements

DBDA provides a simple format for recording data base requirements. Requirements from the various application areas within a company can be recorded with an increased probability of completeness and consistency of content and meaning.

- Automated Analysis of Data Requirements

DBDA performs a comprehensive and thorough analysis of the data elements and their associations. This task is frequently neglected in data base design because the manual process is too laborious and time-consuming.

- Emphasized Designer Control

DBDA analyzes design choices in structuring the data base and detects omissions, inconsistencies, and redundancies in the data requirements. The results of this analysis are reported to the designer, who can then modify the data requirements and easily communicate his decisions to DBDA. The analysis can then continue or be repeated as often as necessary.

- Creation of Comprehensive Reports

DBDA produces design reports that define the structure of the data base and suggest a possible physical organization. Diagnostic/edit reports are also produced that help the designer use DBDA in an iterative manner. They give him information on which to base design decisions and error corrections when he specifies data requirements. Both categories of reports are valuable because they document design decisions and data requirements.

- Optional Interface to IMS Dictionary System

DBDA contains an optional feature that permits the flow of design information into the IMS Dictionary System. For IMS users, the Dictionary System becomes the central repository for a satisfactory design (or version of a design) that the designer chooses to store.

DBDA is useful in a variety of design situations. For customers who currently use or plan to use data base systems such as IMS or DOS DL/I, DBDA can help the data base designer in these areas:

- Designing new data bases
- Redesigning and integrating existing data bases
- Adding new applications to an existing data base
- Adding new elements or associations to an existing data base

DBDA complements rather than replaces the data base designer by automating parts of the design process. DBDA thoroughly analyzes the data base requirements, and generates a structural model of the data base. During this process, it identifies human decision points and prints reports that contain information the designer needs for making decisions. It also identifies and reports errors detected in the data base requirements. The designer can modify the data requirements, then easily communicate his decisions and corrections to DBDA, and the analysis can continue or be repeated.

The design reports that DBDA produces serve two purposes. They define the structural model and they suggest a hierarchy and segmentation of the desired data base as it relates to DL/I. The designer can then refine these suggestions into the design that is needed.

SECTION 2. THE DATA BASE DESIGN PROCESS

The process of designing a data base can be generally divided into the following six steps:

- Gathering requirements
- Generating a structural model
- Constructing a physical model
- Design evaluation
- Physical implementation
- Performance evaluation

In theory (and frequently in practice), the last four steps are usually repeated until a quality design is obtained. How quickly this happens depends upon how effectively the first two steps are performed. The results of any of the steps may indicate the need to modify or clarify the data base requirements. If that is the case, the previously established requirements are modified and a new structural model is generated.

GATHERING REQUIREMENTS

The first step of data base design poses many questions: What do the applications need? What inputs are required to drive them? What data outputs will they produce? How are the data elements related to one another? Which elements are identifiers and which elements do they identify? How frequently are they used? Have input sources been specified for all data elements?

During the process of gathering requirements, these and related questions are answered, primarily during conversations between a data base designer and an application specialist from the department that requests or requires an application. In many organizations, a set of forms appropriately filled in marks the end of the requirements gathering step; in other organizations, less formality is involved. In any case, this first step in data base design ends when the designer collects the data needs of the individual applications that will use the data base being designed.

GENERATING A STRUCTURAL MODEL

A structural model of the data base is an organized collection of the data needs of all the applications using that data base; it provides required information and insight into the design process to follow. A useful structural model shows physical relationships and suggests logical relationships. It indicates which elements are keys and which are attributes of those keys; it shows how the keys are associated, which associations must be stored, and which can be implied from those stored. It can also indicate the relative effect of each association on response time.

An association is a 'from-to' relationship between two data elements, A and B, which means that A identifies B. Thus, in a network or in a tree structure one can go 'from' A 'to' B. For example, consider a parts file in which the parts are identified by part number. One can go from an identifying part number to the name, cost, or other property of the identified part.

In summary, generating a structural model combines analysis of the data requirements of the applications and synthesis of these requirements into a single network. Such a network will help the designer address the questions that arise during construction of a physical model.

CONSTRUCTING A PHYSICAL MODEL

After the structural model of the data base is generated, a physical model can be constructed. Now, additional criteria must be considered such as hardware configuration, access method characteristics, required system performance, maintenance, and backup requirements.

Constructing the physical model means organizing the data into its storage patterns, selecting the storage devices, choosing access methods, and deciding upon an indexing method. The designer is concerned with many questions. Which non-key data elements will be grouped with which keys, and how will they be organized into segments? Where should segments be placed within the physical hierarchy? Where should logical relationships and secondary indexing be established? How many physical data bases should there be? Which associations must be stored and which can be derived from those that are stored?

Answering these and related questions helps to lead the designer to a data base design aimed at maximum performance with minimum data redundancy and minimum, stored, non-essential associations.

DESIGN EVALUATION

Until very recently, a data base design was seldom evaluated before it was put into production because of the lack of effective tools available for such evaluation. Now, for IMS users, the productivity aids DBPROTOTYPE (IUP Program Number 5796-PBB), DBPROTOTYPE/VS (IUP Program Number 5796-PCX), and DCANALYZER (IUP Program Number 5796-PCA) are available to assist in the evaluation process. By simulating the functions of the applications while performing real calls against a prototype model of the data base (or a real data base), the data base design can be evaluated before it is put into production. Early and thorough analysis of data requirements by DBDA, along with early design evaluation by DBPROTOTYPE, DBPROTOTYPE/VS, or DCANALYZER can be used to obtain effective, efficient, and complete data base designs. Moreover, for IMS users, the ability to store the results of DBDA analyses in the IMS Dictionary System can allow accurate documentation of each design.

PHYSICAL IMPLEMENTATION

After the data base design has been evaluated, it can be physically implemented. This process includes generating the data base description library (DBDLIB and PSBLIB in the case of DL/I data bases), and allocating, loading, and cataloging the data base and its indexes.

PERFORMANCE EVALUATION

There are several aspects to the concept of performance evaluation. One aspect is performance measurement of throughput and response times in the environment for which the data base is designed. Another and perhaps even more important aspect of performance evaluation is the adaptability of the data base design to the addition of new or modified applications and the changing data requirements of the company. Can the new requirements be implemented at all and, if they can, what are the performance implications relative to the previously established applications?

Evaluation after physical implementation usually means running the data base in a live environment and processing real applications against it. At this stage, however, it may be too late to correct discovered design errors, so the user must accept a degree of substandard performance. But for users of IMS/360 or IMS/VS, IMS DC Monitor (Field Developed Program 5798-BDF) or IMS/VS Monitor (IMS/VS 1.0.1) are designed to handle such performance evaluations.



.

.



.

.



SECTION 3. SOLVING DESIGN PROBLEMS WITH DBDA

THE PROBLEM OF DATA BASE DESIGN

A crucial aspect of installing a data base system (or of adding new applications to an existing system) can be the design of the data base itself. Poorly designed data bases can affect the response times of the applications and, in some cases, necessitate redesign in order to bring response times within an acceptable range. Although considerable effort is often spent tuning system parameters, understanding the effect of data base design upon system performance can be neglected. Experience has shown, however, that the ineffectively designed data base can be a major reason for poor system performance.

Unacceptably long response times can be one consequence of inadequate data base design. Other problems might include longer application development time, less flexibility in the application's interface to the data base, and numerous application modifications as the data base expands to keep pace with the increasing needs of the company.

For users who store information sequentially on tape, designing a data base is a relatively simple process. However when taking advantage of the capabilities of direct access storage devices and data base systems, this process becomes complex as the following design tasks indicate:

- Gathering the data requirements for the data base
- Identifying and correcting inconsistencies, omissions, and duplications of data elements
- Selecting those elements and associations that must be included in the data base and those that can be derived from them
- Grouping attributes and keys into segments
- Determining physical and logical relationships
- Arranging segments into hierarchical structures
- Choosing the access method

While performing these tasks the designer must be aware of such factors as:

- Frequency of use and priority of use of the application programs
- Frequency of use and type of use of data elements with regard to retrieval, insert, update, and delete options
- Special requirements for updating, reorganizing, and backup

An inappropriate choice in any of these areas can have negative consequences on response times for production and/or maintenance operations. Another consequence may be that the design will not efficiently permit the addition of new applications or new data elements.

Because data base design has been largely a manual process based on a designer's insight and previous experience, there have been few

systematic procedures for gathering the requirements or for analyzing them. Thus, the magnitude of the data base design process can seem to lie beyond human capacity for thorough and effective analysis.

DBDA'S SOLUTION OF THE PROBLEM

The Data Base Design Aid is a productivity aid that organizes the large numbers of data elements and the association paths between them into a network pattern. Such a pattern, known as a structural model of the data base, is of immense value to the data base designer. It shows the association paths between keys from which physical hierarchies and logical relations can be derived. It also shows the clustering of attributes about keys from which segmentation can be determined. The structural model reveals those association paths that are required in the data base and those that can be derived from those paths.

The processing required to generate the structural model can reveal duplications, omissions, and inconsistencies that may be present in the collected data requirements. In addition, the processing can reveal situations in which human decisions need to be made about the elements and their associations. One example of such a situation is the following. When a key is found that has more than one parent, a dominant parent is selected for the physical hierarchy; the remaining parents are candidates for selection as logical parents or secondary indexes. The processing can be organized to help the designer make this selection by printing a report that shows all parents and how they fit into the overall network. In addition, the processing can suggest the dominant parent by identifying the one most frequently accessed.

By automating this process, structural models can be generated with a speed, thoroughness, and consistency not usually possible with manual techniques. Thus, a data base designer will be able to formulate his design based on a complete and comprehensive representation of the data base requirements. Another benefit of an automated design aid is the formality and precision that is imposed on the initial gathering and recording of the data base requirements. Entering them in a format acceptable to a computer program greatly increases the likelihood of attaining completeness and consistency in the resulting data base.

An important supporting function to the data base design process is the ability to store one or many versions of the design in a central, machine readable form. DBDA provides the ability to transfer design data from the Data Base Design Aid to the IMS Dictionary System. This function will be of special interest to IMS users; however, it does not prevent DBDA from being used by non-IMS data base users.

SECTION 4. DBDA PROGRAM DESCRIPTION

DBDA accepts as input the collected data base requirements, and as output it produces a series of design reports that define the structural model and suggest a possible hierarchical structure and segmentation of the desired physical model. The design reports produced by DBDA show:

- The data elements of the desired data base
- The associations defined using those data elements
- A relative measure of the frequency of use of these associations
- A grouping of the data elements into suggested segments
- A suggested hierarchical organization of these segments
- A list of candidate keys for secondary indexing

DBDA is organized into six phases, as shown in Figure 4-1, with each phase consisting of one or more job steps. Phase 1 edits the input requirements and provides a special editing report; Phases 2 through 5 perform the analysis and produce diagnostic reports for monitoring the analysis as it proceeds; Phase 6 produces the design reports. The designer can suspend processing at the end of any phase, make appropriate corrections to the input (data requirements), and restart the process at the beginning of certain phases. The designer can suppress the printing of any diagnostic report he does not wish to see by his use of job control language.

The interface to the IMS Dictionary System will be used only by IMS users, and then only when a satisfactory design (or version of a design) is to be stored in the central Dictionary repository.

DBDA INPUT

The primary source of input to DBDA is a set of descriptions of the input and output requirements, and processing functions of applications that will be using the data base (Figure 4-2). A data base designer, in consultation with an application specialist from each department that requests use of the data base, writes these requirements on DBDA Requirement Specification Forms. This information can then be put into card image form. For OS/VS users, it can be entered into an input data set using TSO; for VM/370 users, the information can be entered using CMS. Further explanation of the primary input and the use of the DBDA Requirement Specification Form is presented in the next section, Gathering and Recording Input.

An additional source of input is a table of control parameters that specifies output format instructions, analysis options, and weighting factors. In addition, this input can also contain special instructions for DBDA to follow when the structural model is generated. For example, control parameters can be used to instruct DBDA to retain certain associations it has discovered to be non-essential (or redundant) and include them in the final results.

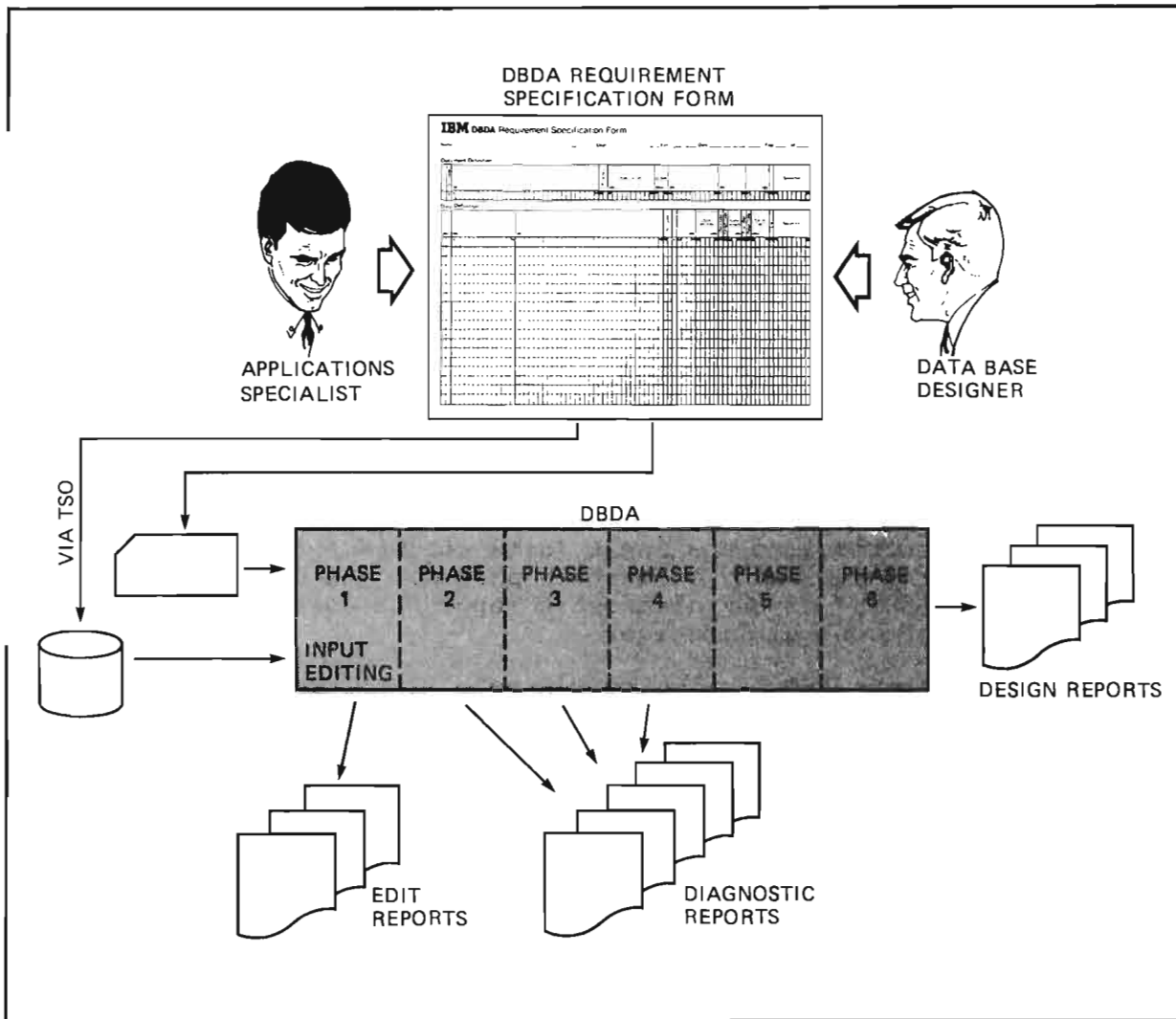


Figure 4-1. DBDA Phases

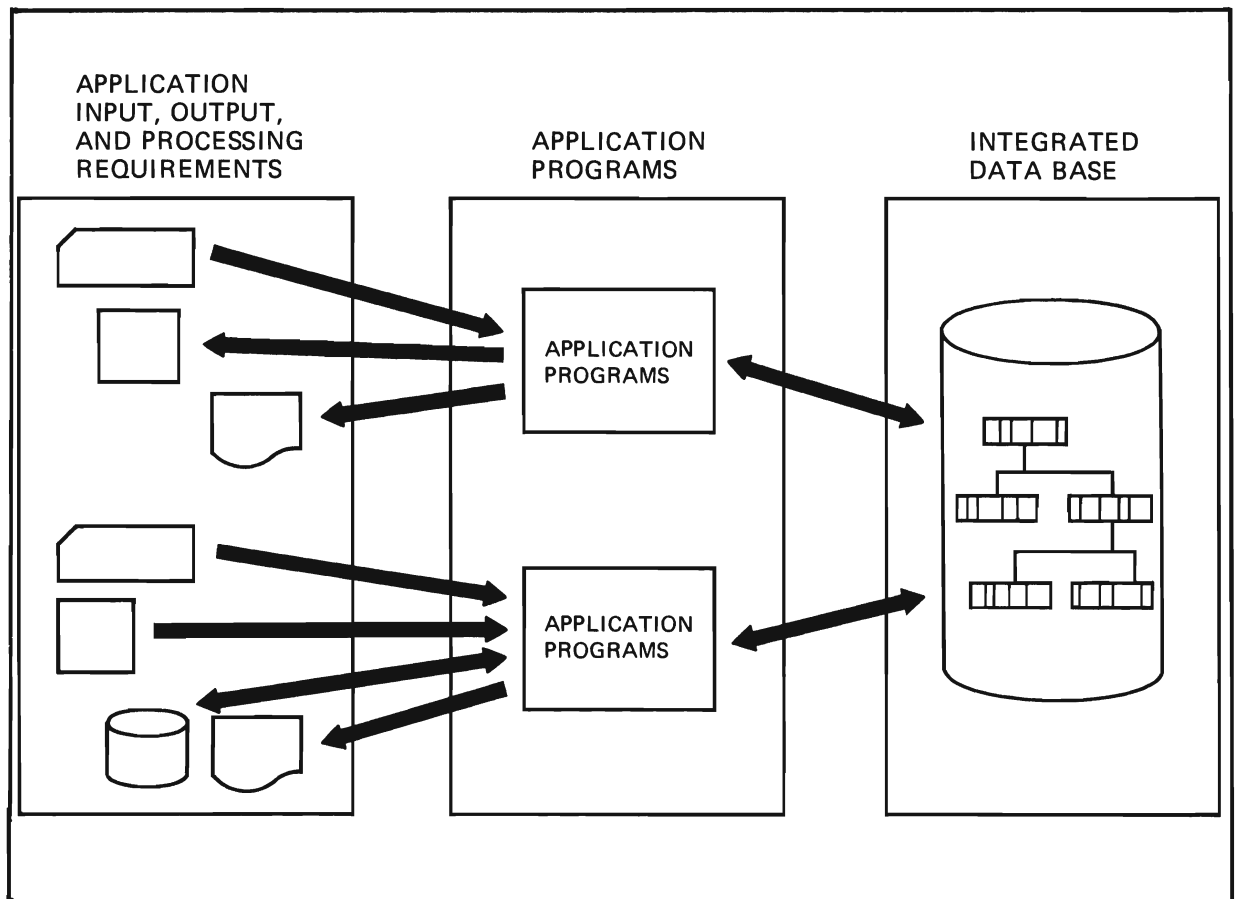


Figure 4-2. Relationship of Data Requirements, Application Programs, and Integrated Data Base

GATHERING AND RECORDING INPUT

An application program receives input, processes it (usually against existing information in a data base), and produces output. For convenience, the term 'document' is used to refer to sets of related items of data used by an application. For example, input documents can be time cards, purchase orders, or insurance claims. Processing documents refer to data items used in identifiable units of internal processing such as subroutines or internal procedures. Output documents can be checks, billing statements, or reports. Each document consists of related data items with some items serving as keys and other items serving as attributes of those keys. The set of documents about a given application make up the data requirements for that application.

The DBDA Requirement Specification Form (Figure 4-3) provides a simple format for recording the data requirements of an input or output document, or a processing function. It has two parts, a Document Definition section used to record information about the document itself, and a Data Definition section used to record the data base requirements derived from the document. The example shown below illustrates the concepts involved in identifying and recording data requirements.

A DATA REQUIREMENTS EXAMPLE

Overview

A data base system is to be designed for processing and recording administrative data (grades, attendance, and class enrollments) for a school. There will be many input sources that yield several output reports, one of which will be a Language Lab Result Report for the school's language laboratories (Figure 4-4). An input source will be a Language Lab Schedule List (Figure 4-5). Each of these documents is shown with the completed DBDA Requirement Specification Form from which it is derived. A keyed explanation of the Language Lab Result Report shows how the information recorded on the specification form was derived from the report.

Language Lab Result Report

The data base designer and an application specialist review the proposed Language Lab Result Report (Figure 4-4, Part A) to agree on the data requirements that are to be recorded on the DBDA Requirement Specification Form (Figure 4-4, Part C). They begin by filling out the Document Definition portion of the form. They record information about an output document (5a) named Language Lab Result Report (5b). The document will be produced daily (5c) from a batch processing environment (5d), and since the school has ten language labs, ten versions (5e) of the document will be produced in each batch run.

Before filling out the Data Definition portion of the form, they draw a diagram (Figure 4-4, Part B) of the required data elements and their associations. The names in the ovals represent the data elements; the arrows connecting the ovals represent the associations to be defined between pairs of elements. The circled numbers outside some of the ovals are identifying numbers assigned to elements that will serve as keys for identifying other elements. The association types simple (Type 1), complex (Type M), and conditional (Type C) are recorded beside the arrows. Diagrams of this type are invaluable aids in interpreting and understanding the data requirements of an application.

IBM DBDA Requirement Specification Form

GX20-1975-0 U/M 050*
Printed in U.S.A.

Name _____ Dept. _____ Ext. _____ Date _____ Page _____ of _____

Document Definition

Doc type Not used	1	2	3	Document name	Priority	Application	Group	Period	Batch/Online	Quantity	Not used	Sequence

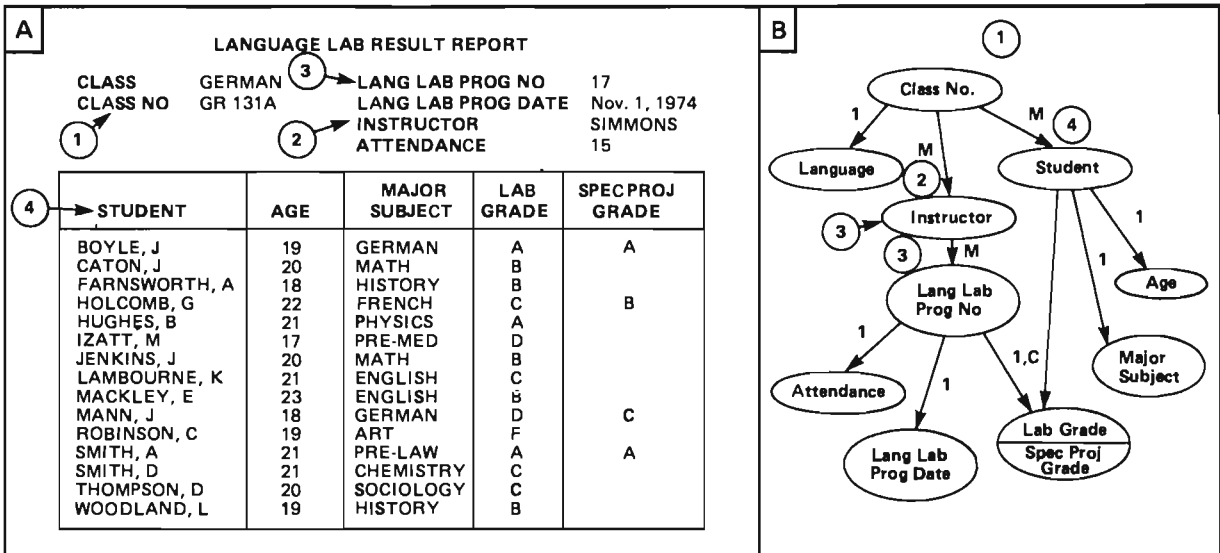
Data Definition

Key ID	1	3	Identified by key ID	1617	Data name	Not used	Assoc type	Freq. of use per key	5152	5556	Freq. of occur per key	Data type	Length type	30313233	Data length	Proc. Code	Factor code(s)	35363738	717273	Count	Sequence	
																						4849

Left justify left-pointing arrows; right justify right-pointing arrows
Coding in shaded columns is optional

* No. of forms per pad may vary slightly.

Figure 4-3. DBDA Requirement Specification Form



C

IBM DBDA Requirement Specification Form

Name A. Mandrell Dept. _____ Ext. 1604 Date Nov 1, 1974 Page 1 of 1

Document Definition

Doc type	Doc name	Priority	Application	Group	Period	Batch	Online	Quantity	Sequence
	LANGUAGE LAB RESULT REPORT				DAILY			BATCH	1

Data Definition

Key ID	Identified by Key ID	Doc name	Not used	Assoc. type	Field	Freq. of occur.	Data type	Data length	Data length	Proc. Op.	Factor	Code(s)	Cont.	Sequence
1		CLASS-NUMBER			1	1	1	1	1					
2	1	LANGUAGE			1	1	1	1	1					
3	1	INSTRUCTOR			1	1	1	1	1					
4	1	LANG-LAB-PROG-NUMBER			1	1	1	1	1					
		ATTENDANCE			1	1	1	1	1					
		LANG-LAB-PROG-DATE			1	1	1	1	1					
		STUDENT			1	1	1	1	1					
		AGE			1	1	1	1	1					
		MAJOR-SUBJECT			1	1	1	1	1					
		LAB-GRADE			1	1	1	1	1					
		SPECIAL-PROJECT-GRADE			1	1	1	1	1					

Left justify left-pointing arrows, right justify right-pointing arrows
 Coding in shaded columns is optional

Figure 4-4. Language Lab Result Report

The content of a document can be interpreted in different ways (which data elements are identifiers of which other elements) and this can lead to differing specifications of the same data requirements. This situation emphasizes the need for close cooperation between the data base designer and the application specialist. Proper interpretation of the data requirements is crucial to any data base design study.

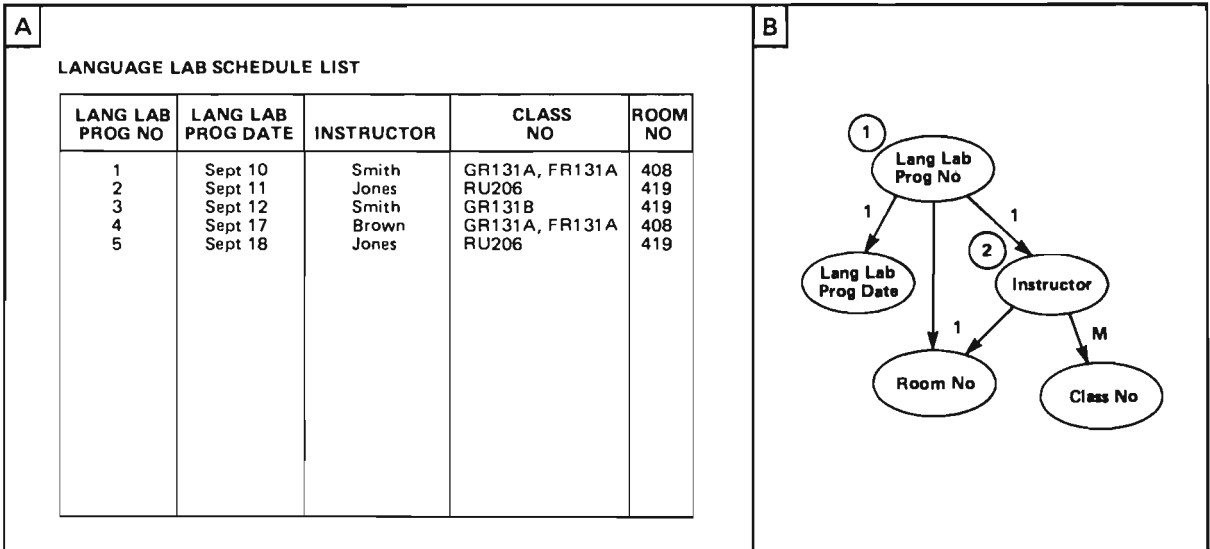
After diagramming the data elements and their associations, recording the data requirements in the Data Definition portion of the DBDA Requirement Specification Form is a straightforward process. The following discussion illustrates the recording process in detail.

The designer and the specialist begin filling out the Data Definition portion of the form, having agreed that CLASS NO (1) should be the root key through which all other data elements on the report will be identified. Knowing that all classes in the school have unique class numbers assigned to them, they record CLASS-NUMBER (1a) in the Data Name field, and they show it as the root (or 1st level) key by writing 01 in the Key ID field (1b). Knowing also that CLASS-NUMBER is used only once per report, they write a 1 in the Frequency of Use per Key field (1c).

Next they record LANGUAGE (2a) and INSTRUCTOR (3a) as elements identified by CLASS NUMBER by indicating that each is identified by the key whose Key ID is 01 (the Key ID assigned to CLASS-NUMBER) (2b) and (3b). The application specialist knows that CLASS-NUMBER uniquely identifies LANGUAGE, so the association from CLASS-NUMBER to LANGUAGE is defined as a simple (Type 1) association (2c). Further, LANGUAGE will be referenced only one time (2d) for each reference to its key, CLASS-NUMBER.

But CLASS-NUMBER does not uniquely identify INSTRUCTOR, for there can be more than one instructor for a class. Therefore, the association from CLASS-NUMBER to INSTRUCTOR is complex (Type M) (3c). In this case, there are two lab instructors, so INSTRUCTOR will be referenced two times (3d) for each reference to CLASS-NUMBER. Since INSTRUCTOR will be used as a key, it is given the next available number, 02, as its Key ID (3e).

The remainder of the specification form is filled in according to a mutual understanding of the requirements of the application. The last two lines, however, on which the elements LAB-GRADE and SPECIAL-PROJECT-GRADE are recorded (4), have a unique aspect that should be explained. Each of these grade fields requires more than one key. In each case, the application specialist wants to record the grades of a particular student in a particular language lab program. Thus, in the Identified by Key ID field, each of these grades is identified by the Key ID (04,03) of STUDENT and LANG-LAB-PROGRAM-NUMBER, meaning that both keys are required for identification. The entry for SPECIAL-PROJECT-GRADE is a Type C association because the association from its compound key, STUDENT and LANG-LAB-PROGRAM-NUMBER, is conditional. In this association, a data element identified by a key can have either 0 or 1 occurrences.



C

IBM DBDA Requirement Specification Form

Name A. Mandrell Dept _____ Ext. 1104 Date 12-17-74 Page 1 of 1

Document Definition

Doc. Type	Doc. Name	Priority	Application	Group	Period	Batch/Online	Quantity	Sequence
1	LANGUAGE LAB SCHEDULE LIST			QUARTER		BATCH	10	

Data Definition

Key ID	Field Name	Not used	Frequency	Freq. of occur. per key	Data type	Data length	Factor	Code list	Code	Sequence
*1	LANG-LAB-PROG-NUMBER		1	1						
*2	LANG-LAB-PROG-DATE		1	1						
*3	INSTRUCTOR		1	2						
*4	CLASS-NUMBER		M	1						
*5	ROOM-NUMBER		1	1						

Let left justify left pointing arrows, right justify right pointing arrows
Coding in shaded columns is optional

Figure 4-5. Language Lab Schedule List

DBDA OUTPUT

During generation of the structural model, DBDA edits and checks the data requirements and identifies errors, omissions, and inconsistencies in these requirements. As a result of editing and checking, DBDA produces a series of edit and diagnostic reports.

As the final result, DBDA produces five design reports that define the structural model of the data base and suggest a possible physical model. The reports represent a thorough analysis of the data requirements collected by the data base designer.

DBDA DESIGN REPORTS

A sample of each report (Figures 4-6 through 4-10) shows the results generated from the school data base system.

Association Weights

Association weights show the relative frequency of use of each of the associations defined to DBDA. Batch and online environments are considered separately, and provision is made for special weighting of inserts, replaces, and deletes. This report helps the designer make decisions that will affect system performance. Figure 4-6 shows a sample report.

Suggested Segments

A list of all attributes (non-key elements) clustered about a key and ordered by key is shown in this report. For each suggested segment the report shows key name, attribute names, an indication of fixed or variable lengths, and segment size. Figure 4-7 shows a sample report.

Structural Model

This report presents a hierarchy of keys showing parent-child relationships at each level of the hierarchy, and it lists the various trees of physically connected keys. In cases where a key is pointed to by more than one parent key, the additional parents are listed. A summary of the association weights is also included (see Association Weights report above). Figure 4-8 shows a sample report.

Candidates for Secondary Indexes

Any data element that appears as a root in the requirements of some application, but is not a root in the suggested hierarchical structure, is listed as a candidate for secondary indexing on this report. Figure 4-9 shows a sample report.

Parent/Child Graph

This report, a sample of which is shown in Figure 4-10, presents a graph of the suggested hierarchical structures. If a key is pointed to by more than one parent key, a dominant parent can be selected by the DBDA program or by the designer.

ASSOCIATION	DOCUMENT	PER.	SYS FLG	USES/DOC	DOCUMENT QUANTITY	PROCESSING FACTOR	NORMALIZED WEIGHT	PR
CLASS-NO ATTENDANCE	LAB PROGRAM RESULT INPUT	DAY	0	2.0	5.0	1.0 G	4.300E+02	
							ONLINE SUMMARY:	4.300E+02
CLASS-NO INSTRUCTOR	LANGUAGE LAB RESULT REPORT	DAY	B	2.0	10.0	1.0 G	4.300E+02	
							BATCH SUMMARY:	4.300E+02
CLASS-NO LANG-LAB-PROG-NO	CLASS YTD RECORD LIST	MON.	B	10.0	10.0	1.0 G	1.000E+02	
	STUDENT STATISTICS INQUIRY	DAY	0	10.0	1.0	1.0 G	4.300E+02	
							BATCH SUMMARY:	1.000E+02
							ONLINE SUMMARY:	4.300E+02
						TOTAL:	5.300E+02	
CLASS-NO LANGUAGE	LANGUAGE LAB ENROLLMENT LIST	QUA.	B	1.0	10.0	1.0 G	3.333E+00	
	LANGUAGE LAB RESULT REPORT	DAY	B	1.0	10.0	1.0 G	2.150E+02	
							BATCH SUMMARY:	2.183E+02
CLASS-NO STUDENT	CLASS YTD STUDENT STATISTICS	MON.	B	18.0	10.0	1.0 G	1.800E+02	
	LAB PROGRAM RESULT INPUT	DAY	0	36.0	5.0	1.0 G	7.740E+03	
	LANGUAGE LAB ENROLLMENT LIST	QUA.	B	18.0	10.0	1.0 G	6.000E+01	
	LANGUAGE LAB RESULT REPORT	DAY	B	18.0	10.0	1.0 G	3.870E+03	
							BATCH SUMMARY:	4.110E+03
						ONLINE SUMMARY:	7.740E+03	
						TOTAL:	1.185E+04	
INSTRUCTOR CLASS-NO	DAILY ATTENDANCE REPORT	DAY	0	5.0	1.0	1.0 G	2.150E+02	
	LAB PROGRAM RESULT INPUT	DAY	0	2.0	5.0	1.0 G	4.300E+02	
	LANGUAGE LAB SCHEDULE LIST	QUA.	B	2.0	50.0	1.0 G	3.333E+01	
						BATCH SUMMARY:	3.333E+01	
						ONLINE SUMMARY:	6.450E+02	
						TOTAL:	6.783E+02	
INSTRUCTOR LANG-LAB-PROG-NO	LANGUAGE LAB RESULT REPORT	DAY	B	20.0	10.0	1.0 G	4.300E+03	

Figure 4-6. Association Weights Report

KEY OF SEGMENT	F/V	LENGTH	DATA FIELDS (C=CONDITIONAL, I=INTERSECTION)
5 CLASS-NO	F	0+	ATTENDANCE(I), LANGUAGE
11 LANG-LAB-PROG-NO	F	0+	ATTENDANCE(I), INSTRUCTOR, LANG-LAB-PROG-DATE
19 STUDENT	F	0+	AGE, MAJOR-SUBJECT
23 (INSTRUCTOR*LANG-LAB-PROG-NO)	F	0+	ROOM-NO
24 (LANG-LAB-PROG-NO*STUDENT)	F	0+	LAB-GRADE, SPEC-PROJ-GRADE(C)

Figure 4-7. Suggested Segments Report

KEY OF SEGMENT	ADDITIONAL PARENTS	FREQUENCY OF OCCURRENCE	WEIGHT C/P	P/C
5 CLASS-NO		00001		
19 STUDENT		00001	2.230E+02	1.185E+04
24 (LANG-LAB-PROG-NO*STUDENT)	LANG-LAB-PROG-NO	00000	0.000E+00	0.000E+00
22 (CLASS-NO*LANG-LAB-PROG-NO)	LANG-LAB-PROG-NO	00000	0.000E+00	0.000E+00

Figure 4-8. Structural Model Report

INDEXING SEGMENT	INDEXED SEGMENT
LANG-LAB-PROG-DATE	LANG-LAB-PROG-NO
STUDENT	CLASS-NO

Figure 4-9. Candidates for Secondary Indexes

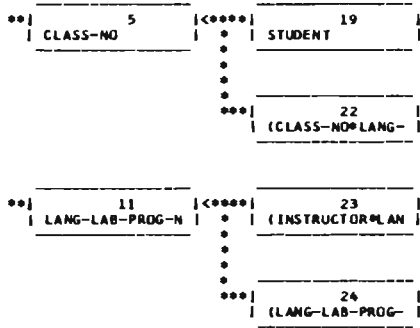


Figure 4-10. Parent/Child Graph

DBDA EDIT/DIAGNOSTIC REPORTS

Each DBDA phase (except the last) produces edit/diagnostic reports that show the analysis to that point. The designer uses the information on these reports to modify and correct the input requirements and to make decisions when choices are reported.

Edit Listing of Input

An identical reproduction of the input records with line numbers added, and input diagnostics by line number.

Name/Line Number Cross Reference

An alphabetic list of all data element names and their input line numbers which helps locate the names on the Edit Listing of Input.

KWIC Listing of Data Names

A further aid in listing and locating data names, application names, and document names.

Inconsistent Associations

A list of user-supplied associations entered with inconsistent characteristics.

Attribute Analysis

A list of multi-keyed attributes with the names of their keys.

Association Path Exceptions

A list of data names in connected associations that loop back onto themselves or that exceed 15 names.

Forced Essential Diagnostics

A list of implied associations specified by the FORCE or DOMINANT keywords that could not be included in the model as essential associations because of DBDA-detected errors.

Complex Associations

A list of all complex (M:M) mappings and whether a connecting path of simple associations exists in each case.

Implied Associations

A list of each user-specified association that can be implied from connected paths of other simple associations.

New Associations

A list of implied associations not specified in the input requirements, but deduced in the analysis.

OPTIONAL INTERFACE TO IMS DICTIONARY SYSTEM

IMS/360 or IMS/VS users can optionally call the Dictionary Command Generator (a part of DBDA) to generate a set of IMS Dictionary control statements and data records that represent the result of a DBDA process or run. This result will normally be an interim or a final data base design and can be stored in the IMS Dictionary System.

ITERATIVE USE OF DBDA

It is expected that the designer will use DBDA in an iterative manner to finally converge on the structural model. Before beginning the analytical phases, the designer may want to run Phase 1 several times until all detectable errors have been removed from the input requirements. While in the analytical phases, he can start another iterative process by stopping at the end of a phase, making further corrections, and beginning the phase again. In some cases he will alter processing control information (contained in tables of control parameters) and repeat only the analytical phases; in other cases he will alter the input requirements and start again at the beginning. In either case, iterative use of DBDA is easy and effective. By closely monitoring the diagnostic reports, the designer can produce the structural model quickly and with minimal reworking.

SECTION 5. PROGRAM ENVIRONMENT

OPERATING SYSTEMS

DBDA operates under the OS/VS2, OS/VS1 and DOS/VS operating systems and uses the VSAM access method. Any release of OS/VS1 or OS/VS2 that supports VSAM (Release 1.5) with multi-positioning and generic key searches is suitable. DOS/VS systems beginning with Release 29 fulfill this requirement.

The host system should include a sort/merge routine designed for VSAM files and the standard OS/DOS interface with E15 and E35 exits. OS/VS Sort/Merge (5740-SM1) and DOS/VS Sort/Merge (5746-SM1) meet these requirements. However, the user can modify the standard DBDA JCL in order to use the standard OS Sort routine (5734-SM1). DBDA performance may be degraded in this case.

The IMS or DOS DL/I systems are not required for DBDA and need not be installed on the system processing DBDA. For IMS users who wish to store the results of a design study in the IMS Dictionary System, the Dictionary (and IMS) must be installed, but this can take place on a different system after the completion of DBDA processing.

DBDA source code is distributed in assembler language which can be assembled with either OS/VS or DOS/VS assemblers.

HARDWARE SYSTEMS

DBDA requires a System/370 CPU (Model 145 or larger for OS/VS users; Model 115 or larger for DOS/VS users) with direct access storage devices supported by VSAM. Storage requirements depend on the operating system used, but the following guidelines apply. DBDA is designed to function in an OS/VS virtual region of 300K, or in a DOS/VS partition of 200K. A tape drive is required for installation only.

For OS/VS users, DBDA input data can be entered into an input data set from a remote terminal using TSO; for VM/370 users, the information can be entered using CMS. Otherwise, a card reader is required for card input. In all cases, a printer is required for the DBDA reports.



.

.

.



.

.



SECTION 6. RELATED IMS PRODUCTIVITY AIDS

A number of IBM Field Developed Programs and Installed User Programs are available to assist in the design of data base/data communication systems. A few of the products that can be helpful in the design, design evaluation, and performance evaluation phases of data base design are described below. Their relationship to the steps in data base design is shown in Figure 6-1.

IMS DICTIONARY SYSTEM (5798-EBA)

- A central repository of information about the structure and relationship of data in an organization's data base, both IMS and non-IMS.
- Provides the ability to administer and control an IMS integrated data base.
- Requires IMS (Version 2.3.1 or 2.4) or IMS/VS (Version 1.0).

Availability Notice: GB21-1255
Program Description/Operations Manual: SB21-1256
Systems Guide: LB21-1257

DBPROTOTYPE (5796-PBB)

- Allows the user to build a prototype of his actual or intended IMS data base.
- Accepts descriptions of application programs to be processed against the prototype data base.
- Executes concurrently with IMS and derives data base call timings and other performance data to allow the user to evaluate the data base design.
- Requires IMS (Version 2.3 or 2.4).

Availability Notice: G320-1523
General Information Manual: GH20-1272
Program Description/Operations Manual: SH20-1303
Systems Guide: LY20-0771

DBPROTOTYPE/VS (5796-PCX)

- Adds full support of unique features of IMS/VS.
- Allows the processing of application models against user-constructed data bases and prototype data bases.
- Adds new features to the design evaluation process such as selective dynamic dumps and CPU time per data base call.
- Requires IMS/VS (Version 1.0) and DBPROTOTYPE.

Availability Notice: G320-1535
Program Description/Operations Manual: SH20-1391
Systems Guide: LY20-0947

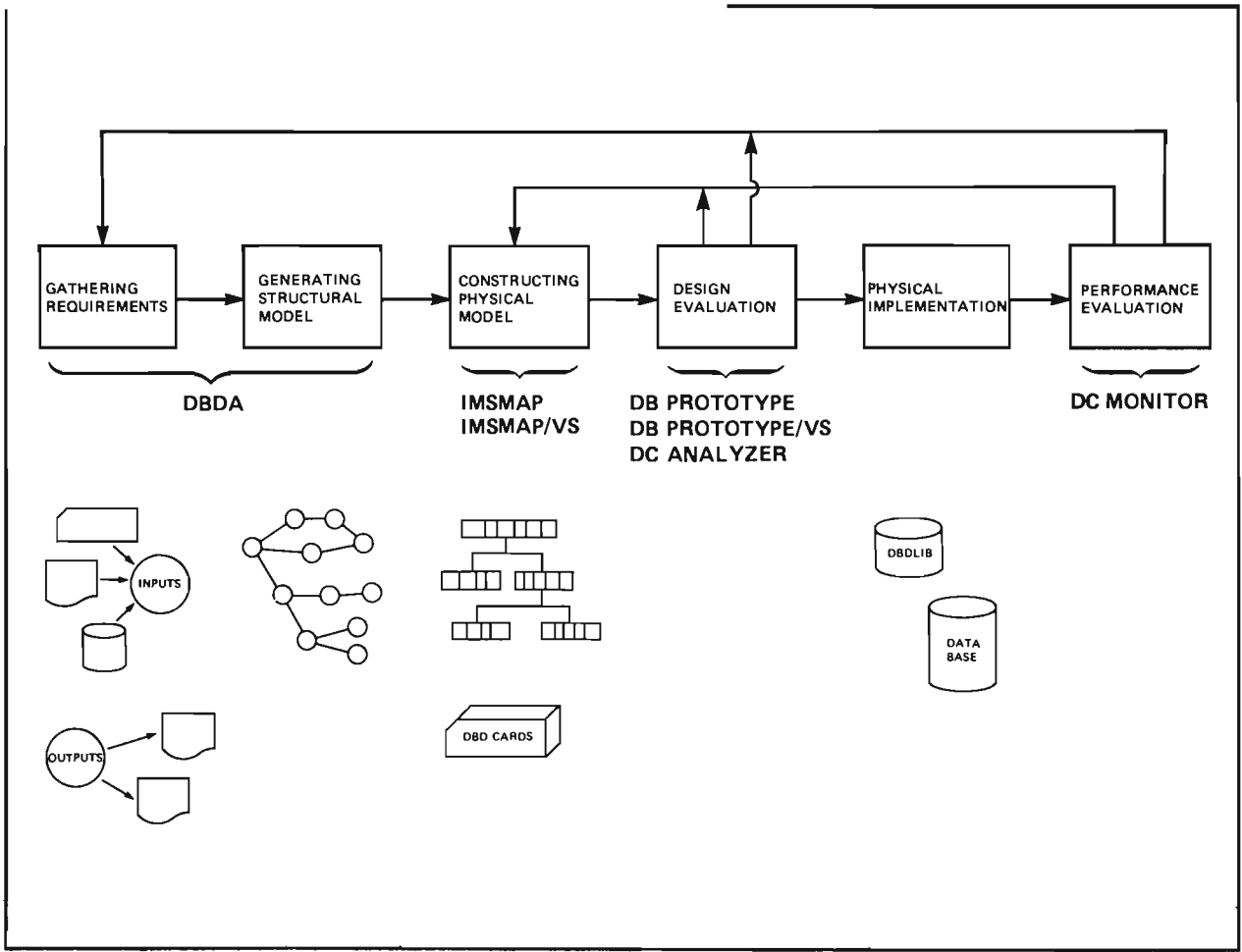


Figure 6-1. Relationship of Productivity Aids to Some Steps in Data Base Design

DCANALYZER (5796-PCA)

- Aimed at assisting users of the data communications part of IMS/360 or IMS/VS.
- Provides the ability to evaluate a DB/DC design under varying transaction loads.
- Does not require or use physical communications lines, terminals, or teleprocessing control units, but drives by feeding transactions to the IMS/360 or IMS/VS message queues.
- Provides reports on response and service times per transaction type under varying load conditions.
- Operates on prototype data bases as generated by DBPROTOTYPE.
- Requires IMS (Version 2.3 or 2.4) or IMS/VS (Version 1.0).

Availability Notice: G320-1532
Program Description/Operations Manual: SH20-1368
Systems Guide: LY20-0937

DC MONITOR (5798-BDF)

- Collects performance data and produces reports on application design, data base design, and resource allocation during test, implementation, and operation of IMS DB/DC systems.
- Indicates effect of data base changes during monitoring of the IMS online system.
- Helps in tuning IMS parameters controlled by the user.
- Requires IMS (Version 2.3 or 2.4).

Availability Notice: GB21-1336
Program Description/Operations Manual: SB21-1337
Systems Guide: LB21-1338

IMSMAP (5796-PBC)
IMSMAP/VS (5796-PCY)

- Prints maps of data base structures which represent the characteristics of an IMS or an IMS/VS data base.
- Prints a detailed report that shows the characteristics of each DBD (data base description).
- Requires IMS (Version 2.3 or 2.4) or IMS/VS (Version 1.0).

Availability Notice: G320-1538
Program Description/Operations Manual: SH20-1539
Systems Guide: LY20-2050

These Field Developed Programs and Installed User Programs are distributed on an "as is" basis, without warranty either express or implied. Successful implementation of Field Developed Programs and Installed User Programs depends solely on the customer's ability to integrate each program into his total inventory of "in-house" produced programs, including his acceptance of full maintenance responsibility. While each offering has been reviewed by IBM for its transferability and maintainability, no assurance of successful installation can be given.

READER'S COMMENT FORM

Data Base Design Aid
General Information Manual
Program Number 5748-XX4

GH20-1626-1

Please comment on the usefulness and readability of this publication, suggest additions and deletions, and list specific errors and omissions (give page numbers). All comments and suggestions become the property of IBM. If you wish a reply, be sure to include your name and address.

COMMENTS

—
fold

—
fold

—
fold

—
fold

Your comments, please . . .

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. Your comments on the other side of this form will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Fold

Fold

First Class
Permit 40
Armonk
New York

Business Reply Mail

No postage stamp necessary if mailed in the U.S.A.

Postage will be paid by:

International Business Machines Corporation
1133 Westchester Avenue
White Plains, New York 10604



Att: Technical Publications/Industry – Dept. 825

Fold

Fold



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

READER'S COMMENT FORM

Data Base Design Aid
General Information Manual
Program Number 5748-XX4

GH20-1626-1

Please comment on the usefulness and readability of this publication, suggest additions and deletions, and list specific errors and omissions (give page numbers). All comments and suggestions become the property of IBM. If you wish a reply, be sure to include your name and address.

COMMENTS

—
fold

—
fold

—
fold

—
fold

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.
FOLD ON TWO LINES, STAPLE AND MAIL.

Your comments, please . . .

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. Your comments on the other side of this form will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Fold

Fold

First Class
Permit 40
Armonk
New York

Business Reply Mail
No postage stamp necessary if mailed in the U.S.A.



Postage will be paid by:

International Business Machines Corporation
1133 Westchester Avenue
White Plains, New York 10604

Att: Technical Publications/Industry – Dept. 825

Fold

Fold



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

GH20-1626-1

Data Base Design Aid General Information Manual Printed in U.S.A. GH20-1626-1

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)