AN EVALUATION OF IMS/VS

by

JOSE P. MIGUEL

B.Sc Univ. de Ingenieria
Lima, Peru 1975

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES
Department of Computer Science

We accept this thesis as conforming to the
required standard

THE UNIVERSITY OF BRITISH COLUMBIA
December 1982

© Jose P. Miguel, 1982
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Computer Science

The University of British Columbia
1956 Main Mall
Vancouver, Canada
V6T 1Y3

Date January 14, 1983
ABSTRACT

This thesis presents an evaluation of the facilities available under IBM's IMS/VS Data Base Management Systems.

Chapter 1 describes the IMS/VS main facilities, and is an introduction to chapter 2. Chapter 2 presents summaries of evaluation of IMS/VS characteristics. These characteristics are grouped in categories as suggested by the Data Base Discussion Group (DBDG) of UBC in setting guidelines for the evaluation of general Data Base Systems. Chapter 3 presents an evaluation of the method of analysis used in this study and the general conclusions drawn from the study.

Appendix 1 contains detailed answers to most of the questions proposed by the DBDG regarding the evaluation of the IMS/VS system. Appendix 2 contains the unanswered questions. Appendix 3 contains a glossary of terms used by the DBDG.
TABLE OF CONTENTS

Chapter 1: IMS/VS Main Features ........................................ 1
1.1 Introduction .................................................. 1
1.2 Storage Organization ...................................... 3
1.3 Database Design ............................................. 5

Chapter 2: IMS/VS: An evaluation ..................................... 13
2.1. Application Programming Facilities ...................... 13
2.2. Architecture, Computing environment,
      Parameter Limits and Multithreading ...................... 18
2.3. Backup / Recovery / Restart ................................ 23
2.4. Communication Facilities .................................. 26
2.5. Concurrency ................................................ 27
2.6. Data Administration and Data Dictionary .................. 28
2.7. Data independence and user views ......................... 31
2.8. Data Model ................................................ 32
2.9 Data Storage ................................................ 34
2.10. Distributed Database Facilities ............................ 39
2.11. Efficiency ................................................ 39
2.12. Human Interface ............................................ 42
2.13. Local Support Facilities ................................... 43
2.14. Portability ................................................ 44
2.15. Reliability ................................................. 45
2.16. Security ................................................ 47
2.17. Software Quality ......................................... 49
2.18. Utilities ................................................ 49
2.19 IMS/VS Uses and costs ......................... 50
2.20 IMS/VS Accomplished Objectives ............ 51
Chapter 3: Method Evaluation ..................... 54
  3.1 Evaluation Criteria ............................. 54
  3.2 Some Recommendations .......................... 56
Appendix 1: Detailed Evaluation ..................... 58
Appendix 2: Unanswered Evaluation Criteria ........ 103
Appendix 3: Glossary of terms ........................ 105
References ............................................. 106
ACKNOWLEDGMENT

I wish to express my thanks to Dr. Paul Gilmore for his helpful and patient assistance during the preparation of this thesis. I also wish to thank the Department of Computer Science, University of British Columbia, for providing financial assistance throughout the course of my studies.
CHAPTER 1: IMS/VS MAIN FEATURES

This chapter provides introductory information on the main IMS/VS facilities. These features are used to create and manipulate Data Bases in an IMS/VS environment.

The chapter has been divided in three separate sections. The first section contains an overview of the system and a description of its main objectives. Section two introduces the different storage organizations and the Operating Systems access methods. Section three describes the Database design technique available. Emphasis is given to the existing logical relationships which can be defined between hierarchies.

1.1 Introduction

IMS/VS is an IBM Database Management System. It was introduced in the early 60's as DL/I. This first system was considered a pseudo language designed to extend the capabilities of the COBOL language. Later several versions and releases were produced and DL/I evolved into IMS/360. The IMS/360 System eventually became the current IMS/VS. The IMS/VS version has incorporated many of the optional features offered by earlier releases. In addition, many of the routines inside the system have been modified and are in constant review to meet the changing needs of the users. This evolution and the process of adding functions has made of IMS/VS a very mature product. The current IMS/VS shows many of the technical advances made in the Database field, and uses the hierarchical model for the implementation of all physical Data Bases. The IMS/VS has been
successful in using the hierarchical model and has therefore popularized it. The IMS/VS software is designed as an open ended system that users tailor to suit their particular environments.

The current version of IMS/VS runs in several IBM/370 machines and compatible models. It uses many of the facilities of the operating systems OS/VS1, OS/VS2 and MVS, especially for telecommunications and terminal support. Additional products which can enhance the IMS/VS performance, are offered by IBM and other software companies.

The IMS/VS logical parent and logical child ideas are a new feature which makes partial networking facilities and many-to-many relations, possible.

IMS/VS is divided into two major components: Database and Data Communication.

The Database component (DB) is a prerequisite for the Data Communication part and is the core of the system. The DB component contains all the routines and facilities necessary for the creation and maintenance of Data Bases. This component can run on-its-own restricted to a batch-only environment.

The Data Communications component provides online support and a reasonable degree of concurrency. This component handles all message processing transactions and controls the normal operation of terminals. In addition it is a prerequisite for the installation of optional features like the Fast Path Feature and the Multiple Coupling System. The Fast Path Feature provides faster-than-normal access to selected data. The Multiple Coupling System Feature provides access to Database Distribution
IBM is constantly updating the IMS/VS system. This is reflected in the numerous additions, new options and changes from one IMS/VS version to another. The IMS/VS system has evolved under the following permanent basic objectives.

1) Independence of the application program from the underlying formats and access methods used for data input and output. Program compatibility from version to version. New versions to be easily handled by old version users and to which application programs can be readily adapted. The introduction of the additional product IMS Application Development Facility (IMSADF) simplified program writing.

2) Possible failure conditions such as system crash, hardware failure and storage media failures should not affect Database and message integrity. These conditions can include system, hardware, and storage media failures. So far, software products have been enhanced with hardware devices such as Non-interruptible power supply and Warning devices.

3) An integrated Database and Data Communication system should allow concurrency access to a Database while maintaining its integrity. The early versions of IMS/360 supported concurrency restricted to batch applications. In response to the demand for a more flexible system which would allow online concurrent updates, the Control Program was modified and several lock protocol policies were implemented with limited success. The search for an ideal concurrent environment ended with the incorporation of the Program Isolation Feature (PIF), as a built-in part of IMS/VS. The PIF allows the execution of concurrent programs as if they were running alone. Concurrent programs can access and update the same Database, with locking mechanism preserving the integrity of the Database. Deadlock situations are detected and solved by terminating one of the programs involved and releasing its resources.

1.2 Storage Organization

The IMS/VS system provides several types of storage organization addressed by a respective set of routines, within the IMS/VS Control Program. Each set of routines has the same
name as the type of storage organization it addresses. The basic
types of storage organization are: HSAM for Hierarchical
Sequential Access Method, HISAM for Hierarchical Indexed
Sequential Access Method, HIDAM for Hierarchical Indexed Direct
Access Method and HDAM for Hierarchical Direct Access Method.
The Databases are named after the type of storage organization
they use. The storage organization types are not Operating
System (OS) access methods, but they interface with truly OS
methods to retrieve the stored data. A more complete description
of these storage organizations is given in section 2.9.

In addition to the above mentioned storage organizations,
IMS/VS offers alternative storage organizations which are used
as migration tools between an IMS/VS and a non-IMS/VS
environment. These optional storage organizations are known as
Simple HSAM, Simple HISAM, and Generalized Sequential Access
Method (GSAM). They make it possible to use non-IMS/VS
application programs with IMS/VS data files. Similarly, through
these optional storage organizations it is possible to use
IMS/VS application programs with non-IMS/VS data files. IMS/VS
provides utilities for the conversion (reorganization) of Data
Bases which use the optional storage organization into one of
the four basic storage organization types.

Each storage organization type uses different Operating
System access methods. The most common methods are: OS/VS
Sequential Access Method (SAM), Indexed Sequential Access Method
(ISAM), Several versions of OS/VS Sequential Access Method
(VSAM) and an special IMS/VS method called the Overflow
Sequential Access Method (OSAM).

The different storage organizations were designed as extensions of an Operating System access method already in existence. This is not only true of the storage organizations but it is a characteristic that applies to many components of IMS/VS.

1.3 Database Design

This section will introduce the main facilities provided by IMS/VS for the design and implementation of Databases. It introduces terms and notations exclusively of IMS/VS.

The IMS/VS data sublanguage (or Data Management Facility) is called DL/1. DL/1 provides a facility to adapt IMS/VS Databases to the user applications requirements. Application programs which use DL/1 deal with the so called application data structures. These data structures define the way in which an application program will "see" the data; thus the DL/1 interface will never accesses the physical data structure. The application data structures used by DL/1 consist of one or more hierarchical data structures. The application programs use one of the three host languages: Cobol, PL/1 or Assembler to describe each of the application data structures. Ordinary subroutine calls provide access to the data.

The following is an example of DL/1 interface effects on data representation. The traditional record layout to represent data about a department, employee and project can be seen in figure 1.1.
Each of these sections (Dept, Employee, Project) may have more than one field. For example, the employee section could consist of employee name, employee address, employee salary. A traditional application program will retrieve the data in exactly the same way as it appears in secondary storage.

The same data appears recorded in figure 1.2. This time, however, the record has been laid out following a DL/1 application data structure. The Dept, Employee and Project sections are separate segments of data. In contrast to most other the name "segment" instead of "record" or "section" appears in IMS/VS terminology.

The structure in figure 1.2 is called a hierarchical data structure. Each occurrence of this structure is called a Physical Database Record. One or more Physical Database Records constitute a Physical Data Base.

Application programs are specified to show only the data indicated by the Data Base Administrator, to the users. The data to be viewed by the users is known by the name of Logical Databases. A Logical Database can be considered as a sub-hierarchy of a Physical Data Base. In all cases the root is part of the Logical Database.
It is clear, from figure 1.2, that the hierarchical data model may present some problems. The one-to-many relations impose several restrictions. If, for example, an employee works for more than one department we will need to have more than one physical Database segment with the same information about the employee. The same problem arises when new Physical Data Bases (such as the one shown in figure 1.3) are defined. These new Data Bases need to use one or more segments ("employee" in the example of figure 1.3) already defined in other Physical Data Bases.

This problem is solved through the introduction of additional pointers between segments within the same or different Physical Data Bases. These pointers provide a logical relationship among segments from the same or different hierarchies. This logical relationship be defined only with HIDAM and/or HDAM Physical Data Bases. Being able to define new hierarchical structures allows the user to avoid redundancy and duplication of information.
An example of a logical relationship is shown in figure 1.4 which shows two Physical Data Bases: The "Office-DB" and the "Dept-DB". In this case there is a unidirectional logical relationship: the relationship between the "Employee" and "Office" segments.

Definition of a logical relationship involves the specification of a logical child (in this example, the employee) and a logical parent segment (in this example, Office). The
child segment will have two access paths: one called the physical access path which goes through the physical parent and another called the logical access path which goes through the logical parent. Both access paths can concurrently be used by a program.

IMS/VS also allows for the definition of bidirectional logical relationships between segments of one or more Physical Data Bases. If a bidirectional logical relationship is defined between the segments "office" and "employee" of figure 1.4, hierarchical structures such as the ones shown in figure 1.5 and 1.6 can be obtained.

![Diagram](image)

**Figure 1.5**

The logical relationship pointers are located in segment prefixes maintained and controlled by IMS/VS-DB software. These pointers are specified through parameter entries made during the
Data Base Definition process.

Figure 1.6

The definition of each Physical Data Base is made in two levels. Both levels are controlled by the Data Base Administrator and are maintained in separate libraries.

The first level of Physical Data Base definition is known as Database Description (DBD). It consists of statements describing the overall Physical Data Base hierarchical structure and each of its segments. These description statements are made through macro-assembler instructions and used as input for the utility program DBDGEN in producing an object code as output. This code is the actual Data Description that will be used whenever a Data Base transaction takes place. The object code is stored in the DBD library. This actual Data description will allow DL/1 to use the physical data storage in accordance with
the requirements of an application data structure.

The Physical Data Base segments can be linked to generate logical Databases through the same utility program DBDGEN. The specification of logical child and logical parent segments can be done through macro assembler instructions. Prior to the definition of a Logical Database the Physical Data Base(s) have to be defined.

IMS/VS provides facilities for accessing data using secondary indexes. The use of secondary indexes requires the generation of a logical Data Base: One of the fields in a logical Database segment is specified as the secondary index or alternative key. Search operations use this field as if it were the only "key segment field".

The second level of Database definition is the generation of Program Specification Blocks (PSB). Each PSB consists of one or more Program Communication Blocks (PCBs). Each PCB defines a partial (or total) hierarchy of a Physical or Logical Database.

Every application program needs to have a PSB before it can run. The Program Specification Blocks define users' views and contain all authorizations to access a particular Data Base. All access authorizations are implemented inside a particular PCB. Segments and fields to which users have access are named "sensitive". Those segments and fields not named this way will be skipped during the user view.

The PSBGEN facility generates Program Specification Blocks through macro assembler instructions. These instructions specify the Data Bases to be used and the authorized segments and fields
within each Data Base. The output is stored in the PSB library and is retrieved when the application program is scheduled to run.

The Data Base Description and the Program Specification Block are used to generate the Application Control Block (ACB) which is stored in the ACB library. Before a program is scheduled to run, the ACB can either be retrieved from the ACB library and loaded to main memory or constructed from the DBD's and the PSB. The access authorizations are first checked against the ACB and the PCB masks specified in the program.

Changes in the specifications contained in the Data Base Description may produce some inconsistencies in the user views already in existence, i.e, the data segments used to produce the user views are dependent on Data Base Description changes. Although it is true that adding segments in certain parts of the Data Description will not need changing PSB's or application programs, more severe changes may produce other effects. For example, deleting a segment or restructuring it, may require the re-generation of some PSB's and/or the recompilation of some application programs. In addition the deletion of a segment by an authorized application program will result in the deletion of all the children segments regardless of whether they were declared sensitive or not in their respective PSB.
CHAPTER 2: IMS/VS: AN EVALUATION

This chapter proposes to identify the strongest and weakest features of the IMS/VS system.

The IMS/VS functions and/or characteristics are here evaluated in accordance with what was suggested by the Data Base Discussion Group (DBDG) of UBC. The DBDG's questions and their specific answers are presented in Appendix 1.

2.1. Application Programming Facilities

The data manipulation language used with IMS/VS is called Data Language One (DL/1). DL/1 provides the interface between a user program and IMS/VS. It can be invoked from one of the three host languages: Cobol, PL/1, or Assembler by means of ordinary subroutine calls.

There are very few differences between an IMS/VS and a non-IMS/VS environment and they imply an extra overhead. When there are differences they are reflected in the Input/Output operations. The Input/Output operations are more sophisticated in an IMS/VS system, since the data is transferred in segment form. These data segments are part of records within hierarchical structures. The differences between an IMS/VS and a non-IMS/VS environment are almost transparent to the user since the IMS/VS system handles the additional overhead through its access methods. The programmer need not be concerned with the mentioned differences.

The basic I/O operations codes of IMS/VS are GET (six variations), REPLACE, INSERT and DELETE. Their uses are
described below:

<table>
<thead>
<tr>
<th>Operation Code</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>GU</td>
<td>GET UNIQUE: Provides direct access to a Database or terminal for the retrieval of a specific segment from.</td>
</tr>
<tr>
<td>GN</td>
<td>GET NEXT: Provides sequential access GET to a Database or terminal for the retrieval of an specific segment.</td>
</tr>
<tr>
<td>GNP</td>
<td>GET NEXT WITHIN PARENT: Another sequential access to a Database or terminal. Its range is limited to a specific parent.</td>
</tr>
<tr>
<td>GHU, GHN, GHNP:</td>
<td>Corresponding to the GU, GN and GNP operations with the H signifying HOLD. They operations are used with the REPLACE and/or DELETE operations to provide a lock.</td>
</tr>
<tr>
<td>ISTR</td>
<td>INSERT: This is a PUT operation used for initial loading or for adding data to a Database.</td>
</tr>
<tr>
<td>REPL, DLET</td>
<td>REPLACE and DELETE: These two operations work in conjunction with the three GET HOLD operations. After retrieval, segments can be written back (REPL) or deleted (DLET). The REPL operation facilitates updating while DLET accomplishes purging.</td>
</tr>
</tbody>
</table>

A single program can invoke several views of a single Database, or it can refer to several Databases. The Databases structures needed by a program are contained in its Program Specification Block (PSB). A PSB consists of one or more Program Communication Blocks (PCB). A PCB is required for each view of a Database that a program intends to use. The Data Base Administrator prepares and maintains all existing PSBs. A program is restricted to the access authorizations contained in its PSB. For online Application programs the PSB and the Data Base Description (DBD) are used to generate an Application
Control Block (ACB). The ACB is loaded into the main memory before the application program takes control.

**DL/1** is a very easy and powerful Data Manipulation Language. A DL/1 call may deal with one or more segments in the hierarchical path, and one or more Segment Search Arguments (SSA). A SSA is a segment name that may be followed by one or more command codes and one or more qualification statements. The qualification statements consist of a set of comparison expressions connected by the boolean operators 'AND' or 'OR'. The command codes modify the way in which a search will be performed. As a result of these control options a DL/1 call may get very complicated, and when executed it may produce unexpected results. For this reason DL/1 is consider to be a poor language for structural manipulation.

User views are known as Logical Databases, and they represent a hierarchical and structural relationship between data segments. The overall quality of user diagnostics is good. Facilities for encryption/decription are available through user written routines and the segment edit/compression exit facilities. There is no built-in report generator and there are no sorting facilities inside the system.

As with many parts of the system, the application environment is in constant revision and improvement. New facilities are always offered as separate products. The more important products are the IMS Application Development Facility (IMSADF), the Interactive Query Facility (IQF) and the Fast Path Feature (FPF).
IMSADF is offered as a tool for a more efficient development of application programs. Included are program modules that are able to directly execute DL/1 calls, edit input data, format displays for certain IBM devices, provide a certain degree of security, and route messages between application programs and system users.

One of the primary objectives of IMSADF is the reduction to a minimum of programming needs. The package is composed of a certain number of functional program modules. These modules are able to perform the basic tasks found in many of the most common programming applications. Other routines contain the logic required to combine existing functional program modules into executable IMS/VS application programs. For applications whose requirements exceed the current capabilities of IMSADF, exits are provided so that the user can attach modules and subroutines of his own to enhance or to add functions not contained within the facility.

IQF is a query language for online data retrieval. Its structure is very simple and consists of commands. Predefined screens are provided for easy display of data. Sorting facilities are built-in and are invoked with the sort command. IQF does not provide any type of update capabilities.

The Fast Path Feature (FPF) is a product which provides rapid access to certain Databases. Because Databases using the Fast Path Feature require special data organization, it is mainly used with heavily used data. In the Efficiency section there is a more complete description of FPF.
Strengths

1) Application programs can invoke several views of a single database or of several databases.
2) DL/1 is a very easy and powerful data manipulation language, especially for search operations.
3) There is a good support for common data processing environments through the host languages PL/1 and COBOL.

Weaknesses

1) DL/1 is not a modern language for structural manipulation.
2) The semantics of DL/1 is not well defined. Control options sometimes can produce unexpected results.
3) Every application program must be explicitly configured to IMS/VS by a generation process done by the DBA. This is inconveniently static.
4) The system has so many and different options in the areas of application configuration and data definition that it can be difficult to use.
5) Online application programs can not directly reference non IMS/VS data sets.
6) There is no comprehensive vendor-supplied report generator.
7) There is too much difference between online and batch application processing. Batch application and online application programs run according to different rules.
2.2. Architecture, Computing environment, Parameter Limits and Multithreading

As mentioned in Chapter 1, there is no clear distinction between the Operating System (OS) and the IMS/VS functions. Many Database operations or features use the facilities already provided by the OS or are an extension of these facilities. IMS/VS has been designed as an extension of the Operating System, and it runs as an authorized program with high priority. The designers, instead of circumventing the OS facilities, have tried to extend them.

The two major division within IMS/VS are: the Database (DB) part and the Data Communications (DC) part. The DB is a prerequisite for the DC and can be run on a stand alone basis limiting the system to a batch-only environment.

The DB is the core of the system and provides support for the description, creation, maintenance, reorganization, access, reconstruction and sharing of all Databases in the system. The DC handles all services related to terminal management and input/output messages and provides the necessary interface with the OS to support messages passing between terminals and the system.

The separation between the DB and the DC is not totally defined. Some functions have been implemented in both places at the same time, and it is not always clear to which major division they belong.

IMS/VS uses extensively the concept of Operating System Region. The IMS/VS Control Program, which resides in the IMS/VS
Control Region, oversees and manages most of the major online activities. Among those facilities are the DL/1 interface, telecommunications, and the control of all access to online Databases.

In addition to the Control Region, a typical online IMS/VS installation will consists of a mixture of online processing regions known as Message Processing Regions (MPR) and Batch Message Processing Regions (BMPR). The MPR are the OS Regions where online application programs are executed. The BMPR provides facilities for the online execution of Batch programs.

The IMS/VS Batch environment is structured as a separate OS region known as Batch Region. This region executes independently from the Control Region. The execution of a Batch Region is considered to be a separate OS job running concurrently with the IMS/VS Control Region. Batch Regions are restricted to non-online Batch programs. Figure 2.1 shows a pictorial representation of these architectural facts.

The use of separate regions in IMS/VS provides for some of its architectural strengths. The online Control Program is isolated and this provides a natural protection against unauthorized use. In the same way, as users programs run in separate regions they are protected from each other. The system does not necessarily become unavailable because of user's errors. In fact, an application program can, at most, abnormally end the region where it is operating.
IMS/VS makes extensive use of OS/VS load libraries. The most important are the DBD library, containing all complete Database descriptions; the PSB library, containing the compiled Program Specification Blocks (PSB's); and the ACB library containing the Application Control Blocks. Before an application program is scheduled for execution, IMS/VS must have available the DBDs and the PSB control blocks. These control blocks are merged and expanded into an IMS/VS internal format called "Application Control Blocks" (ACB) which can be saved in the ACB library. A facility called the Application Control Block Maintenance can be used for pre-building, in an offline mode, the ACBs and for storing them into the ACB library. The pre-building of ACB is required for a DB/DC system, while for a DB it is optional. Before control is passed to a program, IMS/VS gets the ACB from the ACB library, or it constructs it from the
PSB and the DBD. The maintenance of these libraries is the responsibility of the Data Base Administrator.

IMS/VS offers different options for implementing data storage. The most important are: HSAM, HISAM, HIDAM and HDAM. HSAM and HISAM are used when sequential processing will be the main activity, while HIDAM and HDAM are used when the reference is at random and direct access to segments is required. HSAM offers high storage efficiency with poor functionality (no deletions, no updates, and insertions may be made only with initial loading). Both hierarchical methods HIDAM and HDAM offer great flexibility due to direct access to root segments, but the data tends to be fragmented. See Data Storage section for more details.

A certain degree of automatic reorganization is offered for databases of the hierarchical direct type (HIDAM and/or HDAM). IMS/VS allocates and reclaims storage space as segments are inserted and/or deleted.

When the DC feature is installed, a Master Terminal Operator (MTO) is required to monitor the activity of the system, make online operational adjustments, and react to any unusual conditions in the system. The MTO acts as a system resource controller for communication, scheduling and maintenance of the system. For most of his functions the MTO uses the Master Terminal (MT). The MT provides facilities similar to the console for the OS. There is one MT per IMS/VS installation; however in case of failure the system console can be used as MT.
In addition to the usual physical terminals the DC part supports logical terminals. A logical terminal is a name which is related to a physical terminal. More than one logical terminal can be assigned to an specific physical terminal, and the MTO can reassign logical terminals to other physical terminals.

Parameter limits for most facilities of IMS/VS have a good range. A Database can occupy as much storage as it needs. A maximum of 5000 Databases are allowed and the number of physical and logical terminals should not be more than 5000. There is a maximum of 15 segment types in any hierarchical directional path, a maximum of 255 different segment types per hierarchical structure, and no more than 255 fields per segment.

Strengths

1) The online control region isolates the DBMS control program, allowing a natural protection from the users.
2) The use of separate regions for separate applications gives good isolation between programs and the system.
3) The host-language interface (the DL/1 interface) is well localized.

Weaknesses

1) There is no dynamic configuration definition of Databases, user views or application programs.
2) The operating system interface is complex. It is difficult to distinguish between the IMS/VS system and the OS
functions.

3) The separation between DB and DC is not well defined. Some facilities are part of the DB and of the DC at the same time.

2.3. Backup / Recovery / Restart

Most of the Backup/ Recovery/ Restart operations are accomplished with the Data Base Reorganization and Recovery utilities. As most of the utilities can be developed semi-independently from the rest of the system, new utilities and revisions are always included in new releases and versions. At the same time new hardware devices, like non-interruptible power devices, are offered as enhancements to the software. The flexibility, overall ease of use and the wide variety of situations these utilities cover make the Backup / Recovery / Restart one of the strongest points of IMS/VS.

The main function of the Recovery utilities is the restoration of data sets that have been physical damaged and the production of backup copies. To such effect, IMS/VS provides a system log, which is used in all the recovery functions. The Data Base recovery utilities usually consist of copying the physical Database from time to time. The system log maintains all changes made to the Database in the intervening time between copying. In the event that the Database becomes damaged, it can be restored by restoring the Database from the copy and then applying all logged changes (since that copy was made) to the restored Database.
The system log is an output of the IMS/VS Control Program or the Batch Region execution. This output can be used for different aspects like statistics, accounting, and all recovery and restart operations of the system. Within the Control Program there is a routine known as the system recorder. This routine is used for storing most of the data. IMS/VS provides several utilities that can be used when the system log has to be recovered or is not complete, or when log data is lost as a result of system failure. Other utilities are provided to analyse and produce the statistical reports.

The Reorganization utilities deal primarily with the initial loading of the data and with physical and logical reorganization of existing Databases. These utilities allow the insertions of new segments in certain parts of the hierarchy, modification of existing segments, and change of access methods. There are facilities for partial reorganization of Databases.

The many utilities available can create some confusion about their use. On some occasions the use of one utility is not enough. The output of a utility can be used as input to other(s) utilities to get the desired effects. Automatic storage reorganization is offered for HIDAM and HDAM Databases at the cost of possible data fragmentation and the introduction of additional pointers.

The Utility Control Facility (UCF), a separate product, is offered as a solution to some of the problems specified in the last paragraph. UCF provides procedures for the semi-automatic combination and execution of multiple utilities included in the
system. The use of UCF presupposes a previous knowledge of the involved utilities.

In general, all utilities in IMS/VS run as standard OS/VS jobs and can be initiated and controlled from the Master Terminal. The Master Terminal provides capabilities for continuous monitoring of the system and for detecting damaged databases. The Master Terminal can dynamically initiate backup and recovery functions.

Restarting of the system depends on the conditions of the last termination. The console initiates the MT which in turn initiates all major IMS/VS operations.

Rollback and synchronization points are provided for easy recovery and can be invoked semiautomatically. Application programs can request rollback and/or synchronization points easily and fairly freely.

Strengths

1) Great flexibility in the use of rollback and synchronization points. These can be specified at almost any point inside an application program.

2) Backup / Recovery / Restart features are comprehensive, allowing recovery from almost every problem.
Weaknesses

1) Backup / Recovery / Restart requirements constrain the application programming environment in some awkward ways.

2) There are no automatic recovery functions.

2.4. Communication Facilities

IMS/VS DB is a batch environment using standard Operating Systems communication facilities. These facilities are used to provided communication between the Databases and the application program and for the generation of all messages.

For IMS/VS DB/DC the DC part of the system provides for most communication facilities. These facilities are implemented as extension to the existing Operating Systems features, and it is difficult to isolate them. The Operating System provides two communications facilities: the Basic Telecommunication Access Method (BTAM) or the Virtual Telecommunication Access Method (VTAM). However, the use of one excludes the other. If VTAM is used the installation can use the System Network Facilities (SNA) to provide with more advanced telecommunication type functions. The Multiple Systems Coupling (MSC), another optional product, provides for Distributed Database Facilities and for the sharing of Databases between two or more IMS/VS installations.
2.5. Concurrency

The Program Isolation Feature (PIF) provides most of the support for the execution of concurrent application programs. This feature isolates the activity of an application program from other concurrent application program(s), which can be working with the same Database. In this way segments compromised, because of update operations are not available to other programs. When a segment is locked, the program indicates, by reaching a synchronization point, that the data it has modified or created is consistent, complete and available to other programs. A synchronization point is reached when the next DL/1 call is executed or when a program terminates. In addition, synchronization points can be requested at several points in a program.

The PIF uses enqueue/dequeue routines to enqueue affected database elements required by more than one application program. These routines are able to detect deadlock situations. In the case of a deadlock, one of the programs involved is terminated with a special status code and its resources are released.

Granularity of locks is down to segment level, providing a reasonable amount of concurrency. Locks can be exclusive or shared, and can be set at any time. Exclusive locks remain in effect until the next synchronization point. Shared locks remain in effect until the locked segment ceases to be current. Segments may only be updated when they are exclusively locked.

Transactions are supported and, because they can be written using one of the host languages, a good amount of complexity is
allowed.

The concurrent environment is slightly restricted to the Databases attached to either the Control Region or the Batch Region. Programs running in a region dependent on the Control Region cannot access Databases attached to the Batch Region and vice versa.

**Strengths**

1) The lock support guarantees consistency of the data maintained in a concurrent environment.

2) There is good support for transactions. Transactions can be written using the host language allowing a great deal of complexity.

**Weaknesses**

1) The lock protocol is not deadlock free. Furthermore, installations have to write their own routines to handle those exceptions.

2) Databases attached to the batch environment cannot be accessed concurrently from the online environment and vice versa.

2.6. **Data Administration and Data Dictionary**

IMS/VS has been designed for a centralized administration, with the major facilities under the control of one Data Base Administrator (DBA). However, with the influx of more utilities
and the system becoming more complex the DBA job can not be done
by one person but by a group of people who work in a coordinated
way. Skilled technicians are required for the creation and
maintenance of the IMS/VS libraries. A Data Communication
Administrator will be required to handle all policies and
formats regarding telecommunications and terminals support.

The DBA defines both the global perspective (DBD) of a Data
Base and the user's view (PCB's and PSB's). It is the function
of the DBA to establish an interface between programs and the
physical characteristics of the Database. The DBA decides the
type of access storage and physical organization of the
Database. He is responsible for running most of the utilities
concerned with Backup/ Recovery/ Restart operations.

IMS/VS provides the DL/1 Test Program that is used for the
creation of test Databases. The DBA can compare performance
related aspects before deciding the final configuration of a
production Database. If IMSADF is available, it can be used in
conjunction with this program for adding and deleting segments
in certain parts of the hierarchy.

As mentioned before, the DBA administration is a complex
job that usually is performed by a group of persons. The Data
Dictionary (DD), a separate product, alleviates many of these
complexities. The DD uses the facilities of IMS/VS, and it
consists of several IMS/VS Databases. These Databases are used
to document non-IMS and IMS Data Base Systems. The DD provides
commands for the production and displaying of printed reports
containing information about the data in the Databases. It can
produce several types of structure statements about data files, Databases, segments, fields, etc which can be used for different purposes. For example, it can produce DBDs and PSBs in its source form. It can also generate Cobol, Assembler and PL/1 data structures which are stored in special OS libraries for future retrieval by application programs.

Performance related utilities are offered. The DB monitor and the DC monitor gather relevant information about the system. This information is used as input to the chosen performance utility(ies). These reports give indications about the overall performance of the system and the structure of the Databases, and can used to decide when a Database needs to be reorganized.

The MTO provides for a dynamic facility for continuous monitoring of the system. It can give a display of the current status of all resources, and it can change configuration attributes. The MTO can detect damaged Databases and initiate the proper recovery procedures.

**Strengths**

1) The Data Dictionary package provides comprehensive reporting facilities.

2) Many facilities and utilities are available for continuous monitoring of the activities inside the system.
Weaknesses

1) No built-in Data Dictionary.
2) No automatic storage reorganization. A new Data Base Description is needed when the access method changes.
3) Data Definitions are inconveniently static.
4) There can be only one DBA function. This may lead to excessive centralization of database control.
5) Macros that implement DBDs and PSBs do not allow validation features. The data contents depends on the good faith of application programmers.

2.7. Data independence and user views

The Data independence at the storage level is fairly high. Changes in the access method or the physical organization of a Database require minimum changes in the respective DBDs. If the access method changes then a new DBD generation is needed, but these changes do not affect the PSBs or the application programs. Facilities are provided for changing from one access method to another with no effect on the application programs. As a result there is complete independence from the underlying operating system access method used by the system to reach the data.

Changes in the conceptual view of a Database may require some change in application programs or in their PSBs. Application programs are restricted to whatever is specified in its PSB. Some changes in the Database may affect the PSB of an
application program and the user view. Changes to a DBD, like
insertions and/or deletions of segments or fields in certain
parts of the hierarchy, may require the recompilation and
generation of new PSB's. Application programs can not directly
access a Database.

User views are implemented using logical Databases. These
descriptions are contained in the PSBs of an application
program. User views can relate to one or more physical or
logical Database using the concept of logical child and logical
parent. A Database can only be updated using these user views.

Strengths

1) High degree of independence from the access methods and
storage organization.

2) User views allow access to one or more different
databases.

Weaknesses

1) Low degree of data independence for user views.

2) The static configuration restricts the application program
to whatever is specified in its PSB.

2.8. Data Model

The hierarchical model is used for the implementation of
all Physical Databases. A physical Database consists of one or
more occurrences of physical database records, each consisting
of one root segment, and zero or more dependent segments. Each segment consists of one or more fields.

Using the concept of logical child and logical parent extensions are provided for relating two segments in the same or different Physical Data Bases. In this way a limited network facility is provided. The different types of relations and views that can be specified are described in section 1.3.

The DBDs and the PSBs are implemented using standard macro assembler instructions and the DBDGEN and the PSBGEN utility programs. These macros are fixed and do not allow any type of validation procedures or predefined types. The contents of a Database are the responsibility of the Application Programmers.

Index facilities are provided with a full range of capabilities. For HISAM and HIDAM Databases the index provides direct access to the root segment. Secondary indexes are provided for accessing a segment within a physical or logical Database. One of the fields inside the root segment in HISAM and HIDAM Databases is known as the key which is unique for each occurrence of the physical Database Record. The index is constructed from that key. In the case of a secondary index the key is constructed from the field which define that index.

**Strengths**

1) Good hierarchical extensions with the use of Logical Databases.

2) Good index features.
Weaknesses

1) Hierarchical structure.
2) Variable length segment is not powerful enough to allow variable length fields.

2.9 Data Storage

There exist four major data organizations in IMS/VS. These are the Hierarchical Sequential Access Method (HSAM), Hierarchical Indexed Sequential Access Method (HISAM), Hierarchical Direct Access Method (HIDAM) and Hierarchical Direct Access Method (HIDAM). HSAM, HISAM, HIDAM and HDAM are not access methods in the usual sense. Operating System access methods like SAM, ISAM, VSAM, etc. (mentioned in 1.2) are software mechanisms that actually cause the transfer of data between peripheral storage devices and computer memory. The letters "AM" at the end of these acronyms have traditionally signified "Access Method". In the case of IMS/VS these letters describe data organizations, and they do not transfer data. They interface with the true Operating Systems access methods. Figure 2.2 and 2.3 shows the interface between the "access methods" of IMS/VS and the OS access methods. Figure 2.2 shows the interface with non-VSAM access methods and Fig 2.3 with VSAM access methods.

The four basic IMS/VS access methods are described:
A) Hierarchical Sequential Access Method (HSAM): a physical database is implemented using a single data set with fixed length and unblocked segments. The block size is established with the DBD generation. The possible operating system access methods that are used are SAM, BSAM and QSAM. The hierarchical sequence is represented by physical contiguity, and each stored record is immediately followed by its successor in hierarchical sequence. Insertions are allowed only with the first loading and it is not possible to perform updates and/or deletions. This organization is used by OS access methods not supported by VSAM.
and is the only access method which uses tapes. Its primary importance and use is for maintaining backup copies or when the process is completely sequential.

B) Hierarchical Indexed Sequential Access Method (HISAM) In this method of operation the actual data need not necessarily be in absolute physical sequence, but an index to the data is arranged in sequence. Access to a segment is through a root segment key via an index. Storage of deleted segments are not recovered unless there is a reorganization. The physical database can be implemented using either non-VSAM (ISAM/OSAM) or VSAM (KSDS/ESDS) data sets. A database record is implemented
as a physically contiguous string of segments in hierarchical sequence. Each sequence of segments is divided in one or more subsequences. The data is normally stored in an ISAM data set (alternatively a VSAM Key Sequenced Data Set, KSDS). If overflow space is needed one OSAM (or VSAM Entry Sequence Data Set ESDS) is allocated and the next dependent is placed at the front. The main use is for applications that process data sequentially.

C) Hierarchical Indexed Direct Access Method (HIDAM). The purpose of this implementation is for processing data both in sequential and direct ways. A physical data base is implemented as one or more OSAM or VSAM data sets. The segments of a HIDAM database can be stored in different locations relative to one another. The structure is preserved through physical pointers contained in the segment prefixes. IMS/VS manages the space according to insertions and/or deletions. A primary index database is implemented to provide index access to root segments. Pointers provide direct access to dependent segments. The primary use is for applications which require a processing mixture of sequential and direct access. Because storage occupied by deleted segments is reusable, it is adequate for Data Bases with large volumes of insert/delete activity.

D) Hierarchical Direct Access Method HDAM. This implementation is similar to HIDAM and it provides direct access to root segments. The difference is that access to the root segment is done using a hashing routine and no index data base is needed. As with HIDAM Databases this organization provides capabilities for the reuse of deleted segments. It is used for
applications processing data randomly and on line.

In addition to these basic methods IMS/VS can use other access methods to provide an interface between non-IMS/VS application programs and data files of IMS/VS and vice versa. These access methods allow the easy transfer of data from a non IMS/VS environment to an IMS/VS system and vice versa. The Generalized Sequential Access Method (GSAM) provides access support for simple sequential data sets, such as tape files, specific data sets and others which are non-hierarchical. The Simple Hierarchical Indexed Sequential Access Method (SHISAM) supports root segments only. It is used to process non-DL/1 data sets as DL/1 data sets during migration to an IMS/VS environment.

Strengths

1) Several access methods are available.
2) A full range of devices is available for storing data.
3) There is a complete set of utilities for storage reorganization. Partial reorganization is possible.

Weaknesses

1) The large number of access method choices may tend to complicate the DBA's job.
2) Databases become unavailable when their storage is being reorganized.
2.10. **Distributed Database Facilities**

The Multiple System Coupling (MSC) feature, a separate product, provides distributed database facilities between two or more separate IMS/VS installations. One database cannot be distributed into several systems. Failure of one system does not compromise or affect the execution of the other systems.

The MSC feature uses the facilities provided by the SNA (System Network Architecture) for most of its operations. The installation of the SNA requires the availability of VTAM for most of the telecommunication facilities.

**Strengths**

1) MSC is well integrated and does not affect performance of systems involved.

2.11. **Efficiency**

IMS/VS is the overall result of a system in continuous development. Many of its actual efficiency related features were designed as optional in its early releases and/or different versions.

IMS/VS provides a wide range of tools and utilities for continuous monitoring and reorganization of the system. The DB monitor collects data which is used by the Data Base Monitor Report Print Program to produce several efficiency related reports. These reports contain categories and summaries of different IMS/VS calls traced at various levels of detail. In a
similar way, the DC monitor collects information which is used as input to the Data Communication Monitor Report Print Program which in turn produces reports related to the performance of the DC part.

The system contains several utilities for reorganizing a Database and partial reorganization is available. Some utilities are available for detecting when the performance of a Database is below an acceptable level. If the Utility Control Facility is available, most reorganization and recovery functions can be performed in a semi-automatic form. A Database is not available while its storage is being reorganized.

There exist other efficiency related features, either as part of IMS/VS or as optional products. The objective of these products is the alleviation of the Data Administration job. Among these products and not discussed before is the Database Design Aid, a product which produces several reports according to input and processing requirements. These reports contain information about elements in the suggested Database, grouping of elements into suggested segments, a suggested hierarchical organization of these segments, and suggested secondary index segments.

Fast Path is another separate product designed to enable the user to select either improved performance for simply structured transactions or full function for complex transactions, depending on the requirements of particular applications. The Fast Path feature uses selected data in one of two organizations. One, a Main Storage Database (MSDB) which
consists of a root-only fixed length segment Database that resides in main storage. Two, Data Entry Data Base (DEDB) which contains records with root segment types for holding fixed and summary data, and sequential dependent segment types of the roots, which contain detail information. The objective is to provide fast access to selected data by bypassing normal procedures. Fast Path Databases are not available to IMS/VS-DL/1 applications and vice versa.

As a wide variety of storage organization is available, there is a good degree of freedom for choosing an access method. The selection is specified by the DBA at the time of DBD generation. Usually the access method depends on the type of operations that are supposed to be performed with that Database. An access method can provide good performance for certain types of operations and bad for others. For example HISAM Databases offer high efficiency for sequential retrieval, but storage utilization is poor because deleted segments still occupy storage.

Tools for improving programming are provided by the IMSADF, which consists of a wide range of routines. A discussion of IMSADF is offered in section 2.1.

**Strengths**

1) The reorganization utilities run in semiautomatic way, reducing the JCL generation to a minimum.
Weaknesses

1) Some of the efficiency related facilities, like IMSADF and UCF are separate products.

2.12. Human Interface

IMS/VS itself contains no human interfaces for applications use. IQF (Interactive Query Facility) is a separate product that can query IMS/VS Databases.

IQF supports a nonprocedural language and provides a friendly interface for non-programmer users. It was designed as a query language for making inqueries and to provide on-line retrieval of data. It provides automatic formatting of output results. IQF offers capabilities for sorting according to a designated field inside a segment. IQF does not offer update or replace capabilities.

IMSADF (IMS Application Development Facility), another separate product provides a set of routines for immediate use or for the developing of more complex programs. IMSADF has been discussed in section 2.1.

Most of the Human Interface products are optional features. These enhancement products usually require interfacing with other IBM products, which are not within the sphere of Data Base Systems Technology. Several software companies offer additional products with easy interface to IMS/VS. These additional packages become obsolete, rather quickly because as new levels of IMS/VS are released, some of the functions are absorbed into
the IMS/VS utilities, while others see new versions.

**Strengths**

1) A complete set of error messages and codes covering almost every situation.
2) IMS/VS is in constant development and new products can easily be adapted to the system.

**Weaknesses**

1) IMSADF requires additional personnel for maintenance.

**2.13. Local Support Facilities**

IMS/VS is not an easy system to install. The installation is carried out as a considerable size project. The Operating System has to be subject to some modifications and additions before IMS/VS can be executed. Small modifications are needed when a new release of either the OS or IMS/VS is installed. The installation of new features may require hardware and software products that have no relation with IMS/VS to be installed.

The IMS/VS system definition consists of the specification items such as IMS/VS libraries, data sets, buffer pool size, Database Definitions, Programs Control Blocks, screen formats and terminal device and network characteristics. Partial generation and testing products are offered to assist in this process. The manual Installation Guide provides a one step at a time process for the generation of the system. This installation
process requires highly skilled software technicians. In general each major application will require additional personnel for maintenance.

The master terminal facility provides capabilities for displaying operational status of IMS/VS and for monitoring the present configuration. The master terminal can dynamically detect and initiate any activity inside IMS/VS. In this way damaged Databases can be detected and the recovery operations can be dynamically initiated.

**Strengths**

1) The centralized control from the master terminal allows an easy control of the system.

2.14. **Portability**

IMS/VS runs on IBM/370 machines under operating systems OS/VS1 or OS/VS2. The Operating System has to undergo special preparation before the DBMS can run. IMS/VS depends heavily on many of the OS functions for correct execution, and is very difficult to isolate the IMS/VS functions from the OS functions. However the System allows certain degree of portability. Different versions of the same operating system can run IMS/VS with very small modifications.

There exists a certain degree of data portability between IMS/VS and non IMS/VS data sets. In addition to the four basic types of storage implementation (see Data Storage), IMS/VS allows the following types: SHSAM (Simple HSAM), SHISAM (Simple
HISAM) and GSAM (Generalized Sequential Access Method). These three types of Databases can be accessed by both IMS/VS and non-IMS/VS application programs. These implementations use standard OS/VS access methods. They are used when IMS/VS application programs want to pass some Database data to non-IMS/VS application programs or vice versa. Utilities are available to put data (IMS/VS or not) into these formats.

Strengths

1) Data portability facilitates the transfer of data among both IMS/VS and non-IMS/VS installations.

Weaknesses

1) The System is not portable to other Operating Systems outside the specific line offered by IBM.
2) The Operating System has to undergo some modifications before the DBMS can run.
3) The installation process is difficult.
4) The DBMS is written in Assembler language.

2.15. Reliability

The online system is monitored by the Control Program executing in the Control Region, which is an OS Region designated for that purpose. Application programs run in different OS Regions and they do not influence each other. An application program at most will be able to ABEND the region
where it is operating. The Master Terminal is able to monitor the execution of all application programs and to detect damaged databases.

IMS/VS has a close dependence on the performance of the Operating System for overall reliability. Errors inside the OS tend to make the DBMS unavailable. Several utilities are provided for recovery and restart operations and they cover a wide variety of situations.

The macros that implement the DBDs and the PSBs (See Data Administration and Data Dictionary), do not provide built-in data validation features, nor do they allow the inclusion of user-written routines in their generation. The contents of a Database will depend on the application programs used for their updating. All validation procedures have to be included into the programs.

**Strengths**

1) Region isolation protects the system from application program errors.
2) Vendor is very reliable and stable.

**Weaknesses**

1) Overall reliability is influenced by the Operating System.
2) Validation routines can not be included in the Data definition.
2.16. Security

There exist two levels of security in IMS/VS: Data Communication and Database.

Data Communication (DC) security is implemented using operating system facilities. IMS/VS interfaces with the Resource Access Control Facility (RACF) of OS/VS. RACF is used as a basic tool for defining and enforcing user authorization and resource protection. VSAM password protection can also be used as a security tool to protect data sets containing IMS/VS data.

Initial security rules are set up at IMS/VS generation time. Those rules may subsequently be modified by the System Maintenance Utility (SMU) which is part of the IMS/VS-DC product.

Resources that can be controlled include physical and logical terminals, IMS/VS libraries, database files, IMS/VS Control Region services and specific transaction types.

Database (DB) security is integrated into the Program Communication Block (PCB). It is supported using the sensitive segment and sensitive field features, which define the segments and fields the program can access. The set of PCBs an application program requires constitute its Program Specification Block (PSB). Program specification blocks reside in the PSB library and are under the control of the DBA.

DB security is inherent to the PSB facility. All application programs require a PSB to access a database. Unfortunately, since PSBs are static in nature, DB security can be inconveniently static as well.
One peculiar consequence of the hierarchical model is that deletion of a sensitive segment in the hierarchy will cause deletion of all its corresponding children regardless of whether the PSB declares them sensitive or not.

The segment/edit exit allows user written routines for the encryption/decription of data sets.

The Data Dictionary product can also be used for reporting actual standards and procedures, and setting policies.

**Strengths**

1) It is possible to insert some advanced security functions. Encription and decription are user written routines used with the segment/edit compression exit routine.

2) The security system does not impose any significant impact on the DBMS performance.

3) The use of OS/VS facilities allows an extra degree of protection for data sets containing the IMS/VS data.

4) There is good use of the OS/VS security facilities (RACF).

**Weaknesses**

1) There is no single security function. The DB security functions are different in style from the DC functions.

2) Database security is not dynamic. A new PSB will have to be generated when the security options for an application program require some change.

3) System is hierarchical.
2.17. Software Quality

IMS/VS is written in Assembler language. At the present there is no informations about some of the most relevant aspects, such as software modularity, standardization within the DBMS and quality of the internal control block structures.

2.18. Utilities

IMS/VS provides a full range of utilities with different purposes. The utilities can be classified in the following groups:

a) Generation utilities: They are used to assist in the creation, and maintenance of Data Bases. Among them are utilities for the generation of DBD's, PSBs and ACB's.

B) Data Base utilities: Used for reorganization, recovery, loading and reloading of data bases.

C) System Log utilities: Used for restart, recovery and analysis of the system log data.

D) Performance Reporting utilities: Used to monitor the system with data obtained from the DB and the DC monitors and to produce performance related reports.

All utilities are standard OS/VS jobs and are subject to JCL rules. Facilities in some utilities are provided to help construct JCL for running or invoking other utilities. Most of the utilities usually operate on the database.

The installation of addition features may require
additional utilities or some complementary routines to existing ones.

Strengths

1) The JCL generation job is reduced to a minimum.
2) It is possible to use some of the OS utilities that are not specific IMS/VS.

Weaknesses

1) There exist too many utilities, with not enough integration of the system.
2) Some utilities operate directly on the database files, bypassing normal security mechanisms.

2.19 IMS/VS Uses and costs

IMS/VS is a system designed to work in a wide variety of situations. The many separate products that can be added to it and are often needed have made the IMS/VS Data Base environment a very complex one. These additional products, which are sometimes not mentioned in Data Base technology, impose additional overhead on the computer environment and often require the acquisition of additional hardware components. As a consequence, the use of the IMS/VS system is recommended for companies that handle large, as opposed to small, amounts of data.

The original design of the IMS/VS system calls for only one
Data Base Administrator. However, the current complexity of the tasks performed by the system calls for a Data Administration group. This group decides what goes into the Data Bases, who accesses them and what should be the general organization of the system. The services of another Data Base Administrator are required for the Data Communication part facilities.

The training of administrative personnel, software technicians and system programmers can be expensive and in some cases it may take more than a year.

2.20 IMS/VS Accomplished Objectives

The objectives prescribed for IMS/VS as described in Chapter 1, section 1, have not been accomplished entirely. However, IMS/VS is constantly changing. New discoveries in computer technology are bound to find their way into IMS/VS and take the system closer to the prescribed goals.

The first objective, application program independence from the underlying data storage methods, has been partially attained. An application program views the data in a particular way independently from the way in which that data is stored. Application programs access the Data Base according to specific instructions provided by the Data Base Administrator. No direct access is allowed. However, the configuration of the Program Specification Block is static in that is confined to the instructions of the Data Base Administrator. Changes in the Data Base Administrator instructions may constrain and affect the current Application Program environment in different ways and
may require the recompilation of the programs.

The restrictions described above are a consequence of the IMS/VS initial design. This design is primitive by modern standards. It centralizes all administrative functions in the hands of a single Data Base Administrator. Some of these design limitations have been overcome through the use of new separate software products designed to be used in connection with the IMS/VS system.

The second objective, preservation of Database and message integrity, has been reached for a large number of situations. The Backup routines can be used to periodically produce copies of portions or whole Databases as needed. The Recovery routines can prevent the loss of data in case of a system breakdown. The restart routines can be used to start the system after a shut down. Some of the Backup/ Recovery/ Restart functions are performed by separate products designed to be used in connection with the IMS/VS system. These products constitute one of the strongest and most complete parts of IMS/VS. In addition, as they were developed almost independently from the main system, these products can be obtained from several companies. New hardware such as non-interruptible power facilities and warning devices can be obtained as well. These devices provide extra protection in the event of power failures or system crash.

The third objective, an integrated Database and Data Communication system, has been partially attained. The Data Communication part (DC) has been completely integrated within the system. However, this integration makes it sometimes
difficult to distinguish between the functions performed by each of the two parts. In the current version of the IMS/VS system the Data Communication part is heavily dependent on the Operating System and uses many of its facilities for message transaction handling and security purposes. In addition, the DC part makes the use of a good number of additional products possible. These additional products are used to obtain faster-than-normal access to data sets and to establish Database Distribution facilities.

The Data Communication part is optional in that is a product that may or may not be used in connection with the Database part of the IMS/VS system. Products similar to the Data Communication part can be obtained from software companies other than IBM. These alternate products can be integrated to the DB part in the same way that the IBM Data Communication part is integrated.
CHAPTER 3: METHOD EVALUATION

This chapter will discuss the method we have here used to assess the IMS/VS system facilities. The method was suggested by the Data Base Discussion Group (DBDG) of UBC. Suggestions to overcome some of the difficulties encountered when using the DBDG method will be presented.

3.1 Evaluation Criteria

Chapter 2 presents an assessment of the IMS/VS system according to eighteen Data Base subject categories (Sections 2.1 to 2.18) proposed by the Data Base Discussion Group (DBDG) of UBC. These subject categories offer an evaluation of the system on the basis of its main functions. The categories answer a number of questions posed by the DBDG and thus identify the major strengths and weaknesses of the Data Base system.

The IMS/VS evaluation as presented in Chapter 2 involved the application of the DBDG guidelines, and the interpretation of IMS/VS technical journals, but it lacked experimental data.

As suggested by the Data Base Discussion Group, the conclusions drawn in this study can be tested by users of the product. A verified and final version will be the result of this last effort.

In searching answers to the DBDG questions it was found that the IMS/VS manuals provide details about all the major functions of the system. A good deal of information is repeated in several IMS/VS manuals. However, these manuals lack the Operating System function descriptions to which they often
refer. The Primer manual offers a valuable description of the IMS/VSe main features.

A small percentage of DBDG questions was not answered. Unanswered questions dealt with topics such as software quality and efficiency, and were difficult to answer because they are related to the day-to-day operation of the system. Examples of topics related to software quality which were not assessed are the modularity and standardization of the routines that conform the system. Examples of topics related to efficiency which were not assessed are the functions related to system performance in a real environment, like response time, detection of bottlenecks, etc.

The method we have used to evaluate the IMS/VSe system has several strong points, among them:

1) It covers a wide variety of Data Base system functions. From those related to the simplest aspects of human interfaces to those related to the most sophisticated architectural structures

2) The interrelations between the different functions of a Data Base system can be identified and used as a guide by system designers. For example, this study concluded that the utility functions of the IMS/VSe system have a very positive impact on other functions, improving the overall system performance.

3) The grouping of the functions under study in subject categories provides for a logical distribution of study load among the members of a team project.

4) The method allows for a technical comparison of several Data Base Systems. The subject categories can be used as a basis for the design of any number of scoring methods. In this way a company trying to choose between different Data Base Systems can benefit from this method in reaching a decision about which system will suit their needs best.

In the same way, some weaknesses of the method can be
1) Researchers need to continuously verify their conclusions through data obtained by working on a day-to-day basis with the Data Base System which is being evaluated. The fulfillment of this need is not adequately provided for by the method. The method as it currently stands divides the assessment process in two parts to be carried out one after the other: an analytical part based on bibliographic research, and a practical part through which the conclusions of the first part are verified. This separation has also proven to be very time and effort consuming when dealing with a complex system such as IMS/VS.

2) The study does not provide for the complete evaluation of functions linked to or performed by the Operating System. Researchers can only adequately analyse these functions if they thoroughly understand the functioning of the Operating System itself. For example, our study of IMS/VS functions includes only partial evaluations of those functions linked or performed by the Operating System.

3) Emphasis is placed in investigating how functions are performed. There is no requirement, however, to determine the reasons for the use of a particular algorithm in implementing an specific function.

3.2 Some Recommendations

In section 3.1 we talked about some weaknesses of the Data Base Discussion Group method on which this IMS/VS assessment was based. These weaknesses can be overcome as follows:

1) The need to continuously verify conclusions being drawn during the analytical process can be fulfilled by doing the analysis and the verification of the conclusions simultaneously. The work could be carried out in constant communication with users.

2) An understanding of how the host computer (Operating System) operates can be achieved in two ways: a) making sure that the team project includes someone familiar with the
Operating System terminology as used with the Data Base System; b) working at the Data Base installation site.

3) Valuable insight into the design considerations can be obtained through an interface with the Data Base System being evaluated, and by contacting the original designers of such system.
APPENDIX 1: DETAILED EVALUATION

Application Programming Facilities

Question 40:
Availability and quality of report generator, either as part of the DBMS or adjunct to it
Assessment: Not available as part of IMS/VS. Reports can be generated using Cobol programs. GIS/VS an IBM product can be used to generate reports.

Question 45:
Availability and quality of built-in or adjunct facilities to allow convenient development of interactive applications
Assessment: Using IMSADF or Online Program and Job Development System. IMSADF is the most common used and is a separate product which provides a set of routines either for immediate execution or for use as an aid in the construction of more complex programs
Mechanisms: The Online Program and Job Development System is an IMS/VS DB/DC application which provides the ability to modify, test and execute OS/VS jobs from an IBM 3270 display station supported by the IMS/VS database and Data Communication System. IMSADF provides a general purpose application execution program which is customized and completed by the addition of specifications(rules) and, if necessary, special programming, IMSADF is basically intended for conversational applications, but it also allows the development of non-conversational and batch application programs. It includes predefined common functions(dialog management, data-access, application logic and a transaction-driver with menu-selection. Static Rules (application selection, key formation, program execution and data handling) plus Dynamic Rules (Sign-on, audit and message tables) complete the application program
References: 14(1.1), 21, 15

Question 46:
What languages are supported by the host-language interface
Assessment: Cobol, Assembler and PL/I
References: 1(1)

Question 84:
Whether the result of one database query can subsequently be queried and/or updated
Assessment: Not possible with IQF
References: 15

Question 86:
How [elaborate, complex, powerful] the selection criteria can be for search operations
Assessment: Allows a great deal of complexity
Mechanisms: A DL/I call may deal with one or more segments in the hierarchical path, and/or one or more Segment Search Arguments. A segment name optionally followed by one or more commands codes and one or more qualification statements is called a Segment Search Argument (SSA). Qualification statements consists of a set of comparison expressions connected by the boolean operators '*' or '&' for 'and' and '+' or '|' for 'or'. A comparison expression consist of <field, comparison operator, value> with the field belonging to the segment and the comparison operator being one of =, <=, >=, =,<,>. There is no limit on the number of qualification statements in an SSA, but the maximum size of an SSA is 256,000 bytes. Commands codes are specified by a single character preceded by an *. There exist 9 commands codes and they are used to modify the way of a search. For example command D allows the retrieval of the entire hierarchical path up to the lowest segment specified in the DL/I call, the F command allows the search to start with the first occurrence of the segment type indicated under its parent. Command codes can be combined to make the search more powerful. In addition IQF allows complex searches allowing logical, arithmetic and relational operators. Using this facilities information of one field can be related to the corresponding one in another field, and even arithmetic operations can be performed.

References: 29(15), 3(6), 6(2), 31

Question 150:
Availability and quality of built-in sort features
Assessment: In general there is no built-in features available. However, COBOL language facilities are available for easy sorting. In addition and for query processing IQF offers a sort-merge options for the retrieval of segments. During the reorganization of a Database and using the Data Base Reorganization utility, it is possible to reference the sort options of the OS/VS. If no options are specified OS/VS will use default options.
References: 6(5), 31

Question 154:
Degree to which the DBMS constrains the normal application programming environment, e.g. whether programs need special entry sequences, whether they must be in special libraries, whether certain language features cannot be used, whether non-database files can be accessed, ...
Assessment: Yes , IMS/VS constraints the language environment, by providing several inflexible rules, and some mandatory names.
Mechanisms: Each application program contains the so called "DL/I portion of the program". The structure of this portion depends on the language being used, and is used by IMS/VS to find the required PSB and DBD. IMS/VS
fetches them from their respective libraries and loads them into main storage. IMS/VS then fetches and loads the application program and gives it control, passing the PCB's as addresses. The PCB mask is defined according to the language in use. All features of the languages can be used.

References: 3(7), 29(15)

Question 155:
General features of the data manipulation language
Assessment: DL/I provides the following operations: retrieve unique segment (GU), retrieve the next sequential segment (GN), replace the data in an existing segment (REPL), delete the data in an existing segment (DLET), insert a new segment (ISRT). In addition DL/I provides service calls that enable an application program to: identify a point at which it may be restarted, restart from a specified checkpoint, put user information on the IMS/VS system log, put selected control blocks on the IMS/VS system log and obtain various statistics from DL/I.

References: 1(2)

Question 185
Degree to which application programs can query the data definition data
Assessment: No
Mechanisms: There are no facilities for querying the data definition data. The data definition data resides in the DBD library and is under the control of the DBA

References: 16(1)
Question 8
Operating system features required
Assessment: IMS/VS runs under the following OS/VS configurations: OS/VS1 and OS/VS2. Several optional features and libraries of the Operating System are required.
Mechanisms: Some IMS/VS modules require special macros that are resident in optional OS/VS libraries.
References: 9(4)

Question 11.1:
Nature of the process structure, both inside the DBMS and in the application environment
Assessment: IMS/VS is structured as an extension to the capabilities of OS/VS, and is considered as an OS/VS authorized program.
Mechanisms: IMS/VS uses one or more Operating Systems regions. The online IMS/VS system is a control program executing continuously in an OS/VS region called the IMS/VS Control region, which is started from the operator’s console. Other regions called the Dependent Regions can be started either from the console or from the master terminal. The control region owns all databases that can be accessed by online application programs, and is responsible for all database input/output done by application programs. All DL/I calls are serviced by the control region, and it handles all communication with the terminals. The dependent regions controlled by the online system are of two types: a) Message processing regions that are used to execute application programs in response to incoming messages and queries. They handle all normal online programs. They are under the direct control of the Control Region. This region can contain an application program in wait state until a transaction is passed to it. b) Batch message processing regions that are used for operator scheduled batch programs that require access to online databases. They are scheduled by OS/VS job management. In addition to online IMS/VS system batch executions of IMS/VS are normal operating system jobs. From 2(pp 87)

The job is initiated by the operator through a job request that specifies the IMS/VS region controller module as the program to be executed, and allocates to the job the data sets for the IMS/VS databases to be accessed. The job request also specifies, as a parameter to be passed to the region controller, the name of a user-written program to be run in this system execution. When an operating system region becomes available, the job is scheduled for execution. The region controller is fetched from the load module library and given control. The region control loads the control blocks defining the
user program, PCB's for the Databases to be accessed, loads action modules required, loads the user program and transfer control to it. After completion control is returned to the region controller, which in turn returns control to the OS, ending the job. A region where a batch execution runs is called a Batch Region. There can be several batch regions but databases can be assigned only to one region at a time.

References: 30(1), 10(1)

Question 12:
What conversational terminal access services, if any, the DBMS requires from the operating system, or whether the DBMS requires terminal access at the hardware level
Assessment: Access services provided by the Operating System are required. The more common methods are BTAM and VTAM. If the installation is planning to use the System Network Architecture then it requires VTAM, which will provide the ability to access other program products, operating systems programs, and will allow IMS/VS to enjoy its network capabilities
Mechanisms: Both BTAM and VTAM provides the basic link for data transfer between a terminal and an application program. The interface to IMS/VS is through the Message Format Service.
References: 22(1), 21(1)

Question 15:
Peripheral equipment requirements other than auxiliary storage, for instance, a journal-tape drive
Assessment: Requires at least one 2400 or 3400 9-track tape unit. Inclusion of additional features may require additional equipment
References: 1(4), 6(5)

Question 18:
Degree of multithreading allowed for DBMS applications
Assessment: The maximum number of terminal I/O users that can be using terminal transactions is 4095
Mechanisms: The SPAREA macro that is run during installation time defines the maximum number of scratchpad areas (SPA) to be maintained by the IMS/VS system. An SPA area is a work area (either in main memory or in a direct access storage device) used by IMS/VS to retain information from the application program across its execution. The size of an SPA area maintained in main memory can not exceed 32,767 bytes. The size of an SPA maintained in an auxiliary device can not exceed the track length or 32,767 bites whichever is less
References: 16(7.23), 9.69

Question 23:
Degree to which DBMS security uses and/or requires operating system security features
Assessment: IMS/VS has built-in
feature for field and segment access control, but relies heavily on the operating system for overall database security.

Mechanisms: Operating System security features used include RACF (Resource Access Control Facility) to protect system dataset and/or VSAM dataset access control mechanism to protect VSAM datasets. The master terminal can enforce passwords using OS facilities.

References: 2(6.4)

Question 28
Whether the security system is an essential design feature of the DBMS or an adjunct to it
Assessment: Data security is built-in into IMS/VS-DB. Terminal and transaction security is built into IMS/VS-DC and is controlled through facilities provided by SMU and RACF. Whether DC is or is not an essential feature of IMS/VS 'is arguable'.

Mechanisms: Data Security is limited to the facilities provided by the logical database mechanism, which is basic to the IMS/VS-DC host program interface. Terminal and transaction security is implemented via configuration options and master terminal imperatives in IMS/VS DC.

References: 2(4),3(7)

Question 38:
Existence and quality of a host-language interface
Assessment: Cobol, PL/I and Assembler available
Mechanisms: The program is compiled through the user selected language compiler and placed in an appropriate program library, after it is link-edited with the DL/I language interface modules.

References: 3(6)

Question 39:
Whether or not the query language, report generator, and utilities go through the same (or essentially the same) host language interface as user applications do
Assessment: Utilities in IMS/VS run as standard OS/VSE jobs operating directly on the database. All other IMS/VSE application functions use standard IMS/VSE resources for describing the data, accessing data, and communicating with users terminals.

References: 2, 6, 31

Question 43:
Applicability of the DBMS to both interactive and non-interactive applications, and portability of application programs between the two environments
Assessment: Both batch and online available. Batch processing is usually performed when the online system is not being used because it requires the database in an exclusive mode. See #11.1 Architecture.

References: 3(3)
Question 52:
Whether the programs that implement the data description data are DBMS applications--i.e., whether the data description data is itself a database in the DBMS. The term "integrated data dictionary" usually refers to the case in which the data description data is a database.
Assessment: Not integrated
Mechanisms: The Data Base Description is a series of macro assembler instructions describing such things as the database organization, access method, segments, fields in a database record, and relationships between different types of segments. See #77
References: 2(1)

Question 61.1
Whether multithreading is supported
Assessment: Both batch and online execution can be supported at the same time
Mechanisms: They are supported through facilities of the Operating Systems regions. One of these regions holds the IMS/VS control program while the other regions, called dependent regions, are available for online or batch execution. There exist a maximum of 31 dependent regions and they can be of 3 kinds: a) Message Processing Region used for online execution, b) Batch message processing region used for operator-scheduled program that require an on-line database, and c) batch region that is used for normal batch operations. See 11.1 Architecture
References: 30(I)

Question 79:
Format constraints on individual data objects--size, type,...
Assessment: The following value types are provided for a data field:
<table>
<thead>
<tr>
<th>type</th>
<th>Allowed length(bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hexadecimal</td>
<td>1-255</td>
</tr>
<tr>
<td>Character</td>
<td>1-255</td>
</tr>
<tr>
<td>Packed decimal</td>
<td>1-16</td>
</tr>
</tbody>
</table>
The more complex structures are formed using the segment concept or auxiliary ones through the facilities provided by the host language.
References: 30(II)

Question 80
Maximum number of databases per copy of the DBMS
Assessment: A maximum of 5000 databases can be under the control of IMS/VS online control program. The DATABASE macro statement specifies all databases to be used under the control of IMS/VS online control program.
Mechanisms: one DATABASE macro instruction must be specified for each HSAM, HISAM and HDAM databases. Two database macro statements are required for a HIDAM database: one for the index DBD and the other for the HIDAM DBD. One database macro instruction must be included for each
secondary index database that refers to any database defined to the online system
References: 9

Question 81:
Maximum storage size of a database
Assessment: Apparently no restriction

Question 82
Maximum number of users that the DBMS can record
Assessment: There is a maximum of 5000 physical terminals that can be attached to the online IMS/VS installation. A physical terminal can have attached any number of logical terminals. The maximum number of logical terminals is 5000
References: 16(7)

Question 83
Maximum complexity of a data definition, e.g. max. number of data objects per data definition, max. number of data relationships per data definition,...
Assessment: The root may contain any number of child segments types and each child any number of child segment types. There is a maximum of 15 segment types in any one directional hierarchical path and a maximum of 255 segment types for a complete PDBR. A maximum of 1000 fields can be defined for all segments in a DBD and a maximum of 255 fields can be defined for any segment type
References: 10(1), 29(13)

Question 124:
Range of applications the DBMS is designed to cover
Assessment: IMS/VS has been implemented as a general purpose system, such that users can tailor to suit their respective environments. However, it is mainly used in computer business related applications.
References: 1(1)

Question 126
What devices are supported for database storage
Assessment: Tape and disk
Mechanisms: HSAM databases usually reside in tape and are used as backout copy. HISAM, HSAM and HIDAM reside in disk for online processing. If the fast path feature is installed then databases can reside in certain designated regions of main memory.

Question 136:
Degree to which the DBMS uses operating system facilities (as opposed to degree to which the DBMS circumvents those facilities)
Assessment: IMS/VS as an authorized program uses OS/VS facilities rather than circumventing them
Question 140:
Degree to which DBMS and adjunct programs code can be shared among multiple application processes (i.e. whether the code is reentrant and, if so, whether the reentrancy can be used to make the code usable by more than one process at once)
Assessment: DBMS code is sharable
Mechanisms: Through facilities of the Operating System
References: 3(3)

Question 164:
Whether the DBMS includes features to reorganize its own storage, or whether storage reorganization is accomplished by adjunct programs
Assessment: Utility Control Facility provides a semiautomatic way of reorganization
Mechanisms: UCF is executed as a standard OS/VS job. It requires a JOB card (provided by installation), EXEC statements and DD statements that are used to define the inputs and outputs that are required
References: 6(7), 18(1)

Question 165:
Whether a database is available while its storage is being reorganized
Assessment: When a reorganization is taking effect the database is unloaded and hence not available
References: 6(5)

Question 166:
Whether the DBMS uses the host operating system's file system, or whether the DBMS has its own file system
Assessment: It uses the standard OS/VS file system

Question 168
How and when the DBMS uses intermediate, transient files
Assessment: No
Mechanisms: As specified in #18 IMS/VS uses SPA areas to do intermediate work. Typical content of an SPA are data available from the terminal and from databases, that are saved for continuing the conversation. IMS/VS automatically compress and expands scratchpad content to reduce data movement and I/O requirements
References: 1(2), 10(1), 16(1.1)

Question 172:
Degree to which the DBMS design allows a flexible boundary between hardware and software functions
Assessment: IMS/VS operates under the control and facilities of OS/VS and there is no clear distinction between IMS/VS and the operating systems
Mechanisms: The console operator communicates with the
scheduler and the supervisor. The scheduler is the portion responsible for accepting and executing OS/VS commands and JCL statements while the hardware is controlled by the supervisor. The OS/VS nucleus and its resident extensions provide the modules, resident access methods and supervisor calls when running in an IMS/VS environment. In OS/VS process gain control of the hardware only through the supervisor. The master terminal provides capabilities for IMS/VS similar to those provided to OS/VS by the system console.

References: 5(1)

Question 183:
How the DBMS function is divided internally and what each major division does

Assessment: The 2 major components of IMS/VS are the Data Base management services, which basically supports multiple applications using a common database, and the Data Communication services which support multiple terminal oriented application using a common database. The DB management are designed to provide support in: 1) Description of Data Bases, 2) Creation of Data Bases, 3) Access to and maintenance of Data, 4) Sharing of IMS/VS data Bases by multiple IMS/VS subsystems, 5) Reorganization of databases, 6) Recovery and reconstruction of data and 7) Checkpoint /restart for DB environments. The Data Communication management services of IMS/VS provide: 1) Application- independent terminal management, 2) Input/Output message traffic handling, 3) Terminal and Program message switching 4) Message format service for device independence, 5) Transaction initiated scheduling of application program, 6) Conversational /interactive application support, 7) Diagnostic aids, 8) System command and control language, 9) resource security 10) Checkpoint/restart for DB/DC environment, 11) Display terminal paging, 12) audio response.

Mechanisms: The Data Base management services are offered as basic material in a component called the Data Base System. The DB management services is prerequisite to the DC feature. When the DB system and the DC are combined they form the the IMS/VS Data Base/ Data Communication (DB/DC) system. Application programs call upon the database and the data communication through DL/1. For the most part application programming use of DB/DC management services is independent of terminal and storage media programming considerations.

References: 1(1), 10
Backup/ Recovery/ and Restart

Question 19:
OS/VS-style checkpoint/restart facilities, i.e. ability to allow orderly suspension and resumption of application processes on request
Assessment: IMS/VS supports checkpoint/restart as one of its features and is used for checkpoint, shutting down and restarting IMS/VS executions. Checkpoint and restart capabilities are provided for both batch and online systems, and IMS/VS can be stopped and restarted at certain intervals. Checkpoints can be specified in application programs almost at will.
Mechanisms: The master terminal provide a means for entering commands for checkpoint execution and for an orderly termination of the system. The restart command is used for system reconstruction after a controlled stop or an emergency stop or when integrity of the database is in question. To start IMS/VS again the operator instructs O/S to start IMS/VS. Once IMS/VS are defined, a message is transmitted to the master terminal requesting an indication of the type of restart, this answer causes the control to pass to the restart facility, which reads the restart data set or system log. It gets the input messages received but not processed or output messages generated but not transmitted during the previous execution. Any other message or job are put back into the respective queues in which they were left at the previous system stop. After that the master terminal is notified that the restart is complete and it can start sending messages to resume normal processing
References: 1(2), 18(1), 5(3)

Question 127
Ease and computing cost of performing backup operations
Assessment: It is easy using the Data Base Recovery Control Feature.
Mechanisms: Recovery and backup related information are automatically recorded in the RECOM data set
References: 18(1)

Question 148:
Whether user-initiated transaction rollback can be done
Assessment: Yes. It can be done up to the most recent synchronization point. Two types of possible rollback operations. One, which is indicated by return code to the application program and the other to terminate the program with an ABEND message
References: 3(8)

Question 160:
What transaction logging facilities are provided—whether unprocessed transaction input is logged, or before- and after-images. If the latter, whether logical or physical
images are logged

Assessment: All transactions are logged on the system log tape. Both before and after images are recorded. For recovery purposes, however, the system log tapes are not used directly. The Change Accumulation Utility is used to select changed database log records from the log tapes. An image copy of the database is also created at user specified times. When backup or recovery is required, the output of the Change Accumulation Utility and the image copy are used as input to the Database Recovery Utility.

References: 6(6)

Question 163:
What recovery features are built into the DBMS, as opposed to features that are supported by adjunct programs. If recovery features exist, whether they are automatically invoked

Assessment: Data Base Recovery Feature is included within the system

Mechanisms: IMS/VS Data Base Recovery Control Feature stores information about events that might affect recovery in two recovery control data sets. These VSAM key sequenced Data Sets are designed to contain identical information (one is the backup for the other), and each recoverable database or process must be identified in this data set, before any of the available utilities for recovery is used. The two data sets are called the RECON data set. DBRC generates most of the JCL needed to run the IMS/VS recovery utilities.

References: 18(1)
Communications Facilities

Question 44:
Availability and quality of built-in telecommunications features, or interface(s) to external telecommunication systems

Assessment: BTAM and VTAM are offered and any of them are controlled by IMS/VS through the Message Format Service and the IMS/VS conversational Processing facilities. If VTAM is used, the System Network Architecture will be available too. See #12

References: 1, 16(1,7)
Concurrency

Question 59
Granularity of the locks
Assessment: segment type, segment occurrence
Mechanisms: 1.) A PCB associated with an application program may specify an option whereby all occurrences of an entire segment type are locked in exclusive mode. IMS/VS will not load and initiate a program if its PCB entry conflicts with that of any program which is already executing. Two PCB entries conflict if either specifies the EXCLUSIVE option for the same segment type. 2.) A shared lock is automatically set on a segment occurrence when it is retrieved via a GET HOLD operation. If the user updates the occurrence, then the lock is automatically raised to an exclusive lock. IMS/VS also supports shared locks on segment occurrences which may be explicitly set and released by an application program.
References: 29(24.5)

Question 59.1:
Duration of locks
Assessment: 1.) Exclusive locks set via the PCB entry are in effect for the duration of a program. 2.) The shared lock granted in response to a GET HOLD is normally released as soon as the segment concerned ceases to be current, unless the lock has been upgraded to the exclusive level. Both shared and exclusive locks (not set via PCB) are released upon execution of a checkpoint operation or at program completion.
References: 29(24.5)

Question 59.2:
Whether both shared (read) and exclusive (write) locks are provided
Assessment: Both are provided
Mechanisms: see questions #59 and #59.1
References: 29(24.5)

Question 61:
Whether the lock protocol is deadlock-free, or, if not, whether deadlocks are detected
Assessment: Assuming that a program waits if it cannot immediately obtain a requested lock, the lock protocol imposed by IMS/VS is not deadlock free. Deadlock situations happen when a request has been tried to put into the corresponding queue for a resource that is locked by other concurrent program. IMS/VS provides dynamic enqueue/dequeue routines to intercept possible deadlock situations during enqueue processing.
References: 29(24.5), 16(3)

Question 61.5
How deadlocks are resolved
Assessment: Backout facilities are provided
Mechanisms: In a deadlock situation, one of the application programs involved in the deadlock is abnormally terminated with a special abnormal termination code. The abnormal termination causes the activity of the terminated program to be dynamically backed out to the most previous synchronization point, and as its held resources are released it allows other programs to continue its operation. DL/1 provides the ROLL command which permits a user to request backout to the last synchronization point. Start of program, end of program, and checkpoint operations are examples of synchronization points.

References: 29(24.5), 10

Question 64:
Availability of predicate locking, i.e. locking on the basis of some selection condition
Assessment: no
References: 29(24.5)

Question 65:
Whether the DBMS is concurrency supporting.
Assessment: Yes
Mechanisms: The DC feature provides a so called Program Isolation facility. All activity of application programs are isolated from each other running in the system until that application program reaches a synchronization point. The dynamic enqueue/dequeue routine enqueues the affected data between synchronization points. The dynamic log modules saves the prior database record images between those synchronization points allowing to dynamically back out the effects of an application program without affecting the integrity of the database.
References: 30(V), 16(3)

Question 67:
Whether the degree of concurrency control is user-selectable, e.g. whether users can elect to risk viewing the data in an inconsistent state in exchange for quicker or cheaper access
Assessment: Yes
Mechanisms: IMS/VS permits specification of an option in the PCB (called the READ EXPRESS option) whereby the program is allowed to see uncommitted changes made by other concurrent programs. Uncommitted changes are those which have been made by a program still in progress which may be later backed out. The program is not allowed to update uncommitted segments.
References: 29(24.5)

Question 68.1:
How complex a transaction can be (i.e. no. of data objects referenced, no. and type of actions performed)
Assessment: no limit
Mechanisms: Transactions can be written in the same language as batch processing programs and they can have the same structure and characteristics as normal programs. The only difference is that the PCB's are loaded in the control region when the program is first scheduled. IMS/VS maintains a pool of PCB's retaining as many as possible in main memory in order to minimize access to the PSB library.
References: 29(24.5), 30(V)
Data Administration and Data Dictionary

Question 16:
Degree and type of manual direct-access storage administration required
Assessment: All storage administration and mainly reorganization is performed using manual methods. For the general an appropriate reorganization utility has to be run. See #164 & 167 Architecture
References: 10(9)

Question 48:
Whether there is one copy of the data definition data for both data administration and DBMS operation, or whether the DBMS operates from one copy and data administration is done using another. The term "active data dictionary" usually means that there is only one copy
Assessment: There exists 2 copies of the Data Base Description for each defined Database
Mechanisms: One of the copies is the source form, while the other is the object form residing in the DBD library. If the Data Dictionary is available the source copy can be maintained there, or the user can invoke some facilities to produce a source form
References: 2(1),27(1)

Question 50:
Whether the DBMS or the database in question must be made temporarily unavailable in order to make a data definition take effect
Assessment: IMS/VS does not need to become temporarily unavailable, but the Database in question becomes unavailable
Mechanisms: For each Database there is a unique Data Base Description. The DBD are static Assembler language macros instructions. They are subject to the rules contained in OS/VS and DOS/VS VM/370 Assembler Language. A utility called DBDGEN is usually run when a new DBD or a replacement is needed.
References: 6(1)

Question 53:
What data administration data is part of the data description data
Assessment: The source form of a Data Base is available for use by the DBA, while the object form is in use. Any changes in the Data Description will require to generate a replacement object form. Application programs and/or PSB's can be affected by such changes. If the Data Dictionary is installed and in use then the DBA can obtain any information he wants from the DBD and the PSB libraries. The dictionary can produce reports for him about PSB, DBD and logical views.
References: 10,27(1)
Question 54:
What access and update usage statistics are collected and what reporting features for them are available
Assessment: DL/I Call Summary Report available. The database DB monitor Report Print Program is an offline program that produces several reports related with activity in the system. Among these is the DL/I Call Summary Report. The DL/I Call Summary Report arrange data about PCB's used and for each PCB the DL/I call function(s) employed. For each call function it records the segment accessed and the return code obtained
Mechanisms: The DB monitor Report Print Program use data gathered by the DB monitor on either the IMS/VS system log or in a separated DB monitor log
References: 16(9), 2(8)

Question 55:
What data storage use statistics are available and what reporting facilities for them exist
Assessment: There exists several utilities available. Their use depend mainly on the type of access to the Data Base. For example the Data Base Surveyor utility scans all or a user specified part of an IMS/VS HDAM or HIDAM database providing a report describing the physical organization and the free space utilization
References: 7(5)

Question 77:
Availability and quality of special tools (if any) for maintaining and querying the data description data
Assessment: A separate IBM product, the IMS/VS Data Dictionary is available to help maintain and indirectly query the Data Description Data
Mechanisms: The IMS/VS Data Dictionary maintains an IMS/VS Database of Data Definition Data. When an application database is to be created or modified, the data Dictionary package can be used to produce the appropriate database or PSB generation macros. Also various queries and reports on the Data Dictionary's database are supported
References: 27

Question 119:
Ability to define in the data definition user-written data validation procedures
Assessment: Not during database definition. DBD and PSB generation are macro assembler instructions with fixed rules. Any validation procedure has to be written as a separate routine or as part of an application program.
Mechanisms: IMS/VS can generate some error messages if operands or parameters other than those shown for each type of database are coded, or if operands or parameters necessary are omitted or they do not correspond to specifications of DBD and PSB.
References: 2(1)

Question 120:
Whether more than one database administrator can exist, each
with his or her own set of databases
Assessment: There exists only 1 database administrator for
the total installation. The DBA controls Data Base
Creation, PSB generation, Recovery of Data Bases,
establishing security, etc. Due to the complexity of this
job it is most likely that the DBA is a group of persons
References: (16)

Question 149:
Availability and quality of built-in data validation features
Assessment: There is in general no built-in data validation
procedure
References: 6(1-63)

Question 167:
Degree of DBA control over data storage structures
Assessment: There is a fair degree of control of data storage
structures. It is possible to reorganize the data
structure and change the access method. Partial
reorganization is available for HDAM and HIDAM databases
Mechanisms: There exist 10 reorganization utilities that are
run as standard OS/VS jobs. Some of the reorganizations
are unlikely. See # 57 Architecture
References: 1(2), 6(5)

Question 180:
How complex a data integrity constraint the DBMS is capable
of enforcing, either by user-written procedure or by
built-in data validation feature, e.g. how many data
items may be referenced, whether constraints can be
specified over greater than one action

Question 181:
Availability of automatic data modification functions, either
by built-in feature or by user-written procedure, e.g.
encryption/decryption, compression, encoding/decoding,
formatting, etc.
Assessment: The segment edit/compression facility is an
IMS/VS function containing user exits that allow the user
to encode, edit, or compress the data portion of a
variable length segment. Encryption is performed using
user written procedure which is entered at the segment
edit/compression exit.
Mechanisms: Use of the segment edit/compression facility is
specified by segment type on the SEGM statement in the
DBD. The user writes an edit routine that actually
manipulates the data in the segment. The IMS/VS function
gives the edit routine information about the segment's
location and assists in moving the segment back and forth
between the buffer pool and the application program's i/o
area. When a segment is written, it is passed to the edit routine for encryption before being put in the database. The reverse happens when a segment is read, with the segment being decrypted by the edit program before being sent to the user.

References: 2(4.2,11)

Question 186:
Degree to which subschemas are validated against the database definition, and whether such validation can be done at will
Assessment: User views definitions are contained in the PSB library that is generated by the DBA. The system checks that is a "legal" view (i.e. all segments and fields are defined in the DBD and that the hierarchy is preserved). These validations can not be made at will.
References: 16(1)

Question 187:
Availability and quality of tools to help the DBA discover the impact of schema changes on (1) the validity of data in the database, and (2) the validity of existing subschemas
Assessment: The Data Dictionary (DD) provides a tool for detecting the impact of changes in the Data Base Description over views and on application programs.
Mechanisms: The DD can produce reports indicating which PCB's and which programs may be affected due to changes in the DBD. The DBA and programmers are responsible for doing the adequate changes
References: 2(9)

Question 191:
Availability and quality of support for "triggers", i.e. application programs that are automatically invoked when some prespecified data condition arises
Assessment: None available during database definition. However host languages Cobol and PL/I provide some facilities in that direction.
References: 6(1)

Question 196:
Availability of features to help the DBA predict the performance impact of proposed data definition changes
Assessment: Yes
Mechanisms: The DBA can choose to develop a test database to set up standards for the production databases. If the IMSADF is installed the DBA will be able to dynamically add segments to a test database and monitor its activity. Reports can be generated to give indications of the actual performance characteristics.
References: 2(5)
Data Independence and User Views

Question 56:
Degree to which application programs are protected from changes in the conceptual model
Assessment: There exist a certain degree
Mechanisms: Use of LDB which allows segment and field isolation through 'sensitive' declaration. New segment type may be added at the same level in the hierarchy (not between two levels) with no effect at all in application programs. However, certain modification in the original hierarchy may require certain changes to application programs or may affect its PSB. Variable length segments are allowed. Equivalent to view concept - must be careful about deletes as deletion of parent segment can wipe out children segments
References: 29, 6

Question 57:
Degree to which changes in the storage level necessitate changes in the conceptual level
Assessment: Not much
Mechanisms: The DBD is still the same with the only difference that the access method is changed. A new DBD is needed with the run of an appropriate utility. Transfer from one storage method to another is available but for some 'transfers' it is unlikely to be used.

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>HSAM</th>
<th>HISAM</th>
<th>HDAM</th>
<th>HIDAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>HSAM</td>
<td>X</td>
<td>OK</td>
<td>#3</td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>HISAM</td>
<td>#1</td>
<td>X</td>
<td>#3</td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>HDAM</td>
<td>#1/2</td>
<td>#2</td>
<td>X</td>
<td>#2</td>
<td></td>
</tr>
<tr>
<td>HIDAM</td>
<td>#1</td>
<td>OK</td>
<td>#3</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

#1 Used to provide back up copy or when a reorganization is taking effect
#2 new DB load in hierarchical sequence
#3 Problems with root segment
It would seem that the conceptual level does not change but applications may need recompilation
References: 1, 6(1,2)

Question 76:
Availability and quality of support for user views of the data. Low quality view support only allows views to be defined on restricted subsets of the data, while high quality view support allows fields to be computed from other fields and from external data, and allows subschema records to be constructed from more than one schema record's fields
Assessment: A fairly high quality of support for user view through the use of logical Databases that allows a user defined view of the database. In addition it allows the cross-referencing or interrelation of data within two or more physical databases through the concept of logical
parent-logical child

Mechanisms: If a logical relationship is going to be defined then a logical DBD has to be defined. Each logical DBD is defined in terms of one or more underlying physical PBDs, which must already exists. The macro statements are essentially the same as with the definition of the physical DBDs with the exception of the SEGM and LCHILD statements. The SEGM statement, which describes a segment -its length and position in hierarchy- is expanded to include the new types of pointers. The LCHILD statement is added to include the logical relationship between the segment types.

References: 10(4), 29(17)
Data Independence and User Views

Question 76.1:
Degree to which the database can be updated through a user view

Assessment: A user view is implemented through a logical database

Mechanisms: This view is supported by a PSB block in the PSB library. The Data Base will be update through this view, if the corresponding authorizations were defined in such PSB. Because of the hierarchy model deletion of a parent segment will cause the deletion of all its children regardless of if they were declare sensitive or not in the PSB

References: 29(14),2
Data Model

Question 70:
Most elementary element of data that the DBMS knows about, i.e. field, segment, record...
Assessment: The most elementary element of data that IMS/VS knows is the field
Mechanisms: Fields are defined during database definition and are part of a segment
References: 1(2)

Question 71:
Relationships supported (1:1, 1:n, n:m)
Assessment: The main relation supported is 1:n. Logical databases allows the definition of logical relationships between 2 physical databases and hence relations of type m:n
References: 16(1)

Question 72:
Whether or not data objects can be ordered, and what ordering criteria are provided (i.e. order by entry, order by some data value)
Assessment: There is an order provided when one of the defined segments has a field that defines a unique key. The order will be according to the value of the key field
References: 6(1)

Question 73:
Whether duplicate data objects are allowed, or whether each object requires a unique key
Assessment: Duplicates data objects are allowed and that is defined in the field parameter of a segment at DBD definition time
References: 6(4)

Question 85:
Whether the database access operations are closed, i.e. whether the output of one operation is the same type of object as the input of the operation
Assessment: Operation is closed. Application programs return a hierarchical view
Mechanisms: IMS/VS maintains a current position pointer that can point to any segment in the database. When a GN call is issued the current position pointer is advanced to point to the next segment in hierarchical sequence. GU statements produce a change in the position of the current pointer. Facilities are provided for maintaining more than one current pointer at the same time.
References: 30, 6

Question 87:
Whether non-key fields can be searched by other than...
exhaustive means
Assessment: Non key fields may be searched using secondary indexes
Mechanisms: After DL/I has been indicated that a secondary index is being used for a particular segment DL/I will process the segment as though the indexed field is the key. Secondary indexes are specified in the DBD generation of the Database by macros that immediately follow that segment.
References: 2(3)

Question 156
Whether facilities for specifying linkages between data objects exist so that update of one data value will cause automatic update of another
Assessment: No
Mechanisms: An application program can update only one segment at a time. The GH puts an exclusive lock in only one segment at a time. Before updating a segment the program has to put a GH on that segment
References: 3(6)

Question 190:
Degree of data typing/data abstraction available, i.e. whether data elements are strongly ("date", "address") or weakly ("character(8), character(60)") typed
Assessment: No degree
Mechanisms: All macros for the definition of DBD's and PSB's are subject to non-flexible rules
References: 6(1,2)
Data Storage

Question 13:
Data storage efficiency
Assessment: High efficiency with lower functionality or low efficiency with higher functionality
Mechanisms: In HSAM databases all segments within each database record are physically adjacent in storage. In this way there is virtually no added storage overhead. Inconveniences are that segments can only be updated by rewriting the whole database. Delete and replace operations are not allowed, and insertion is allowed only with the initial loading. HDAM and HIDAM databases provide a high functionality but the storage organization is not as efficient as in HSAM because of the need to maintain pointers to several segments. Because of this data tend to be fragmented. The system tries to insert segments as close as possible to their hierarchical location and during deletions storage is reclaimed. In HISAM databases deleted segments still occupy storage.
References: 2(4)

Question 117:
Whether variable-length data objects require the minimum amount of storage necessary to record their current values, or whether they always require some larger size
Assessment: When a variable length segment is initially loaded, the space used to store its data position is the length specified in the MINBYTES operand or the length specified in the size field whichever is larger. If the space in the MINBYTES is longer, more space is allocated for the segment than is required. The additional space can be used when existing data in the segment is replaced with data that is longer
References: 10 (4)

Question 118:
Availability and quality of indexing features
Assessment: A full range of index features offered. The most important is the index database that provides direct reference to physical database by way of their content
Mechanisms: From 2(pp.104) 'The index database is a degenerate case of a physical database, in which the record type consists of a single (root) segment type. The following types of index database are provided: a) Primary index databases: that provides direct access to the records of a physical database(the indexed database) by way of the key value in the record root segments. The index database contains one record for each record of the indexed database. This record contains the key value from the root segment of the corresponding record in the indexed database (indexed databases keys must be unique). Each index data base record contains a pointer to the
root segment of the corresponding record in the indexed database. b) Secondary index databases: providing means for directly accessing a segment within a physical or logical database by means of data within the segment or within some dependent of the segment. A secondary index database contains one record for each occurrence of the source segment type in the indexed database. Each index database record contains a search field value which is the concatenation of the values of up to five fields in the associated source segment. The index database may also contain a subsequence field derived from up to five source segment fields, up to five duplicate data fields taken from the source segment( segment to be accessed) and arbitrary user maintained data. The search field and subsequence field together serve as the secondary index database record key, which may be either a unique or a multiple key. The purpose of the subsequence field is to permit the generation of unique keys within the index database, even in the presence of duplicate search fields, and thus make the accessing of the index database more efficient. Index databases are automatically created and maintained by the system. In addition they may be accessed by application programs in the same way as physical databases. The provision for carrying duplicate data in secondary indexes is useful for rapidly accessing frequently used data. Index databases are defined in a manner similar to physical databases. An LCHILD statement is used to designate the target segment type, and the target segment type definition specifies, through an XDFLD statement the source segment type and search field. At most one primary index database and any number of secondary index databases may be associated with a physical database. All physical databases connected by logical relationship, together with their associated index databases, constitute a database group. The data base group is a significant construct from an operational point of view, since in general all members of a group must be present to process any one of them'.

References: 30(2)

Question 171
Types of storage structures used for database data
Assessment: There exist several types of data storage structure.
Mechanisms: There are four principal database access methods, each with its own storage structure or structures. HSAM databases uses a single SAM data set with fixed length unblocked records, and the segments are stored in hierarchical sequence, in one or more consecutive data sets. HISAM databases may be implemented on one of two OS/VS access method: ISAM/OSAM or VSAM. With the ISAM/OSAM base, a physical database can be implemented as two data sets: an ISAM data set and an OSAM data set. With the VSAM data set two data sets are also used, a Key
Sequenced Data Set (KSDS) and an Entry Sequenced Data Set (ESDS). In HIDAM databases, a physical database is implemented as one or more OSAM or ESDS data sets, each of which holds all occurrences of a given set of segment types (Without regard to record structure). The data set that holds root segment occurrences is called the primary data set and the remainder (optional) are called secondary data sets, the segments of a record are interconnected through physical pointers. An associated index database provides access to root segments in primary data set. Structure of HDAM database is similar to HIDAM database. The only difference is that a hash function is provided instead of an index database to access root segments or in terms of HIDAM the primary data set.

References: 30(11)

Question 193:
How frequently data storage reorganization is required, as a function of data definition complexity and update frequency.
Assessment: It depends on the particular type of databases and the most application programming environment.
Mechanisms: For HISAN databases reorganization is required when there were too many insertions and/or deletions such that the hierarchical structure is no longer maintained or deleted segments still occupy a good percentage of storage. For HD databases reorganization is required when the segments are too much fragmented or when the quantity of pointers is affecting the performance. Several utilities and facilities are available to help decide when to make a reorganization.
References: 16(6)

Question 194:
Granularity of storage reorganization functions, i.e. degree to which database storage can be reorganized in small independent parts.
Assessment: Yes
Mechanisms: IMS/VS has incorporated the Partial Reorganization Utility facility that allows the user to perform reorganization in specified sets of data. Facilities are provided to help decide the range of data to be reorganized.
References: 6(5)
Distributed Database Facilities

Question 173:
Whether different invocations of the DBMS, running in separate, interconnected computer systems, can communicate with each other.
Assessment: In addition to communication with other programs and terminals in an IMS/VS system, a program can communicate with terminals and programs in other IMS/VS installations.
Mechanisms: Multiple System Coupling (MSC) is used. MSC establish links between 2 or more separate IMS/VS systems. The terminals and transaction codes are defined as belonging to that system.
References: 3(6)

Question 174:
Degree of data sharing possible among separate, communicating DBMS invocations—whether whole databases can be shared, or just directories of some kind.
Assessment: Whole databases may be shared.
References: 29(24.5)

Question 175:
Whether each single database must be stored entirely under one DBMS invocation, or whether a database can be distributed piecewise under a set of communicating DBMS invocations.
Assessment: Each database must be stored under the direct control of a single IMS/VS invocation.

Question 176:
Degree to which having multiple interconnected DBMS invocations compromises, confounds, or restricts concurrency control, compared to the case in which there is only one DBMS invocation.
Assessment: No, each system acts in a separate way. The communication is through a logical link.
Mechanisms: When two configurations are connected by a MSC link, a definition of the link must be included in the definition of both configurations. Physical link parameters include name, type and address of the channel or line that implement the link.
References: 30(IV)

Question 178:
In a set of intercommunicating DBMS invocations, degree to which catastrophic failure of one invocation affects availability of the others.
Assessment: Each system in a multiple system configuration maintains its own recoverability and uses the full recovery features of a single system. Facilities are provided to back up unavailable multisystems connections.
and systems
Mechanisms: When one of the interconnected systems fails the logical link to other systems is lost and all communications have to wait until it is successfully restarted on the same or an alternate physical link.
References: 5(3)

Question 179:
Degree to which having multiple interconnected DBMS invocations compromises, confounds, or restricts backup and recovery, compared to the case for which there is only one DBMS invocation
Assessment: No, having multiple interconnected DBMSs does not compromise, confound, or restrict backup or recovery.
Mechanisms: Each system in the multisystem configuration still use the full recovery capabilities of IMS/VS. Full recovery in the multisystem configuration is ensured as long as no log tapes are lost and no cold start of the system is attempted.
References: 10(12), 5(3)
**Efficiency**

**Question 2:**
CPU load imposed as a function of database size and transaction rate

**Assessment:** The effect of the size of the Database on CPU load will mainly depend on the kind of organization used at the storage level. For example in HISAM databases, deleted segments still occupy physical storage until a reorganization is performed. For HIDAM and HDAM databases it will depend on the actual quantity of fragmentation already imposed.

**References:** 6(5)

**Question 4:**
Amount of processor memory required to store the DBMS when no database activity is in progress (residency requirement)

**Assessment:** For a basic support of DC at least 500K

**Mechanisms:** The total amount of storage will depend on the total number of features required to install. IBM through chapter 5 of the Systems Programming Manual provides several formulas and guidelines to calculate the storage required for each additional feature.

**References:** 4(5)

**Question 90:**
Availability and quality of performance tuning mechanisms

**Assessment:** Yes. Most of the tuning tools are based in detecting processor resource problems, I/O resource contention, communication performance. Other tuning tools consist of utilities used for making a reorganization of the database.

**Mechanisms:** IMS/VS provides 2 separate monitors: The DB monitor and the DC monitor. Each monitor gather and format performance related data and record this data on a statistical report, these statistics are used or can be used as input to related report programs, that produces outputs summarizing and categorizing IMS/VS activities.

**References:** 5(3),2(8),10(10)

**Question 114:**
Availability and quality of efficiency-improving features, e.g. ability to precompile queries, automatic storage reorganization, automatic performance monitoring and tuning, etc.

**Assessment:** IMSADF allows many IMS/VS applications to be implemented without conventional programming. IMSADF provides facilities for implementing test databases with dynamic insertion and deletion of segments. In this way the DBA can have some guidelines about performance and reports to help him decide the best option for a production database. Reorganization or changing the structure of storage can be accomplish using some of the
available utilities or writing a program to such effect. The Utility Control Facility can be used to perform most of the operations and maintainance for both recovery and reorganization in a semiautomatic way.

References: 14(1)

Question 157
How efficient the DBMS is for sequential retrieval
Assessment: Several degrees of efficiency
Mechanisms: The most efficient are HSAM databases because all segments are stored in hierarchical sequence, and sequential retrieval of segments will be easy. The most complicated are HDAM and HIDAM. In the case of HDAM databases sequential access gets complicated because of hashing access to roots and in the case of HIDAM because of fragmentation and traversing 2 databases.

References: 29(18)

Question 170:
Availability and quality of query optimization features (i.e. access strategy computation methods)
Assessment: None integrated. Another IBM product: fast path feature is offered to produce faster response time and it allows terminals to have rapid access to main storage databases (up to field level) and to direct access data entry databases. The user can choose use this feature but he has no control on it.
Mechanisms: Fast path bypasses the normal IMS/VS transaction message queueing and application scheduling

References: 11
Human Interfaces

Question 37:
Availability and quality of naive-user-friendly query facilities, either as part of the DBMS or adjunct to it
Assessment: Yes
Mechanisms: The Interactive Query Facility (IQF), a separate product, is available for use in IMS/VS installations
References: 31

Question 41:
Availability and quality of advanced-user query and report-generator facilities, e.g. facilities with concise error messages, less prompting, and more powerful primitives.
Assessment: not available, but IQF can be considered in this line
References: 31

Question 42.1:
Whether the query and report generator facilities are procedural or non-procedural
Assessment: IQF is a nonprocedural facility
References: 31

Question 121:
Appropriateness of DBMS for small problems
Assessment: IMS/VS is a large and complex system that is not recommended for small problems.

Question 122:
General quality of user diagnostic features (e.g. error messages)
Assessment: Good
Mechanisms: There exists a manual on error codes. Each error contains a short description. The user is referred to the manual for a more extensive explanation and possible action
References: 13

Question 146:
Whether the query language(s) include update features
Assessment: No
Mechanisms: IQF is a query facility and it does not contain any update features. IMSADF can be used as a query language facility and as such include update facilities
References: 15(2.9 2.13), 31

Question 147:
Whether the report generator(s) include update features
Assessment: not available

Question 153:
Availability and quality of built-in or adjunct (batch) data-
entry facilities

Assessment: Yes
Mechanisms: Usually accomplished by user written application programs that are stored in the IMS/VS application program library and are used during batch executions of the system. Batch programs can run either in the online execution in a Batch Message Processing Region or in a Batch Region. In both cases a Database can be accessed for update by only one application program at a time. See #11.1 Applic Prog fac.

References: 30(II), 16(1-17)
Local Support Facilities

Question 14:
Sensitivity of the DBMS to new releases of the operating system
Assessment: IMS/VS as an extension of the capabilities of the OS is sensible to changes. It is recommended that for proper execution of IMS/VS, the system definition and system execution be performed under the same Operating System release. For example, unless it is a requirement in the OS, a new IMS/VS system definition using the same IMS/VS release does not make it necessary to recompile user application programs
References: 9(4)

Question 17:
Sensitivity of DBMS applications to releases of the DBMS. Ability of old applications to use new versions of the DBMS and of new applications to use old versions of the DBMS
Assessment: A fair degree of compatibility. Application programs, in IMS/VS version 1, using the Message Format Service and input cursor field may require a minor change. Releases 1.4, 1.5 and 1.6 that uses Multiple Coupling Feature will have to undertake certain modification before release 2 is installed
References: 1(2)

Question 36:
Existence of tools for programming experts to use in repairing DBMS bugs, and in recovering the DBMS from circumstances in which normal backup and recovery facilities have failed
Assessment: An emergency restart is provided by IMS/VS that handles the condition caused by an ABEND of IMS/VS or a machine error
Mechanisms: A message is send to the OS/VS console. To recover the system log the OS/VS DUMP command can be used to create a dump data set that can be used when an emergency restart is begun. The system log is closed online by IMS/VS during an emergency restart
References: 5(3)

Question 69:
Availability and quality of DBMS-maintained performance statistics and reporting facilities thereon
Assessment: IMS/VS Statistical Analysis Utility, File Selecting Formatting Print Program and the Log Transaction Analysis Utility are the main facilities to produce reports about performance
Mechanisms: All of these utilities run with the System Log data set. The System Log data set contains all inputs output messages, all kinds of references made to a Data
Base, all uses of DL/I, etc. The IMS/VS Log Tape Merge Utility is provided for merging up to nine IMS/VS input log tapes, which can be used as input to the other utilities.

References: 6(9)

Question 98
Relative complexity of procedures for installing new versions of the DBMS and adjunct programs.
Assessment: Not very easy. Usually a new system generation will be recommended. Some new features, from release to release do not offer any difficulty for installation in old versions, while others may require additional modifications in the Operating Systems. In addition a new release can differ a lot from previous releases. For example, the Program Isolation Feature for concurrency supporting was an optional feature in earlier releases. With the increase in concurrency supporting and facilities for database distribution it became part of the DBMS package after release 1.5.

References: 9

Question 141:
Ease of maintaining configuration information—what storage devices or areas are in use, what application programs use the DBMS, what user IDs there are, what terminals use the DBMS, and so on.
Assessment: It is easy either using the master terminal or by running some of the utilities such as SMU or the Resource Access Control Facility. The main control is from the Master Terminal.
Mechanisms: The master terminal can receive system messages and is allowed to start, stop and restart IMS/VS. Master terminal can display the status of the system and its resources. It can display number of programs and databases that are active, number of transactions to be processed, number of active terminals, number of logical terminals. The command used is /DISPLAY with several options.

References: 10(1), 23, 27

Question 143:
Availability and quality of built-in diagnostic features—error logs, event traces, selectable operation traps, etc.
Assessment: Diagnostics features are mainly in the way of messages and codes. Installations are free to define certain error routines to handle some specific codes.

References: 6

Question 158:
Availability and quality of database dump/reload functions—whether partial dumps and reloads are possible, whether dump and reload operates on logical or physical portions.
of the database, whether a database can be dumped or reloaded while it is being used, ...

Assessment: Several options for dump reload databases
Mechanisms: The Image Copy Utility permits getting a backup copy of a Database. It can operate on either physical or logical database. Partial dumps are allowed too, and in both cases are OS/VS jobs. A database can not be used while it is being dumped or reloaded

References: 6

Question 159:
How costly database dumps and reloads are—in particular, whether dumps and reloads are practical at all for very large databases
Assessment: Not known about cost. They can be practical for producing back-up copies or when a reorganization is going to take effect
Mechanisms: Dumps and reloads are standard OS/VS jobs. The cost is mainly in the time the database will not be available. They can be practical when performance of the response time for a particular database has degenerate to an unacceptable level. For example in HISAM database deleted segments still occupy storage that is traversed each time a transaction is processed. If a database is too large, partial reorganization is available.

References: 6

Question 184:
Ease of displaying operational status of the DBMS (i.e. programs running, terminals enabled)
Assessment: The master terminal provides for means to display operational status of the system. It provides for a way to detect programs running, databases actually in used, logical and physical terminals connected to the databases, etc. See #141 Architecture
Mechanisms: Master terminal provides facilities similar to the monitor for the Operating System

References: 10
Portability

Question 105:
Compatibility of DBMS with usual data processing application environments
Assessment: Compatible
Mechanisms: Cobol and PL/1 are among the most widely used languages for data processing

Question 111:
What computing environments are capable of running the DBMS (i.e., how portable the DBMS is)
Assessment: IMS/VS runs under the virtual storages operating systems OS/VS1 and OS/VS2 in systems IBM/370 models 138, 145, 155II, 158,165II and 168, on the 303X and 4300 processors, and on any other compatible machine.
References: 1(4),9(3)

Question 151:
How portable the DBMS data is
Assessment: Certain degree of portability
Mechanisms: IMS/VS, in addition to the basic types of Data Bases, provides for SHSAM, SHISAM, and GSAM databases for converting from one non database system to IMS/VS and viceversa. They allow IMS/VS application programs (using OS/VS access methods) to pass data to non-database application programs and viceversa. The differences between SHSAM, SHISAM and GSAM is in the kind of Operating System access method used, if variable records are allowed, if they can use segment edit/compression facility, and others.
References: 2(4)

Question 188:
Availability and quality of tools to aid in migrating application programs and data to or from other DBMS environments
Assessment: Application programs depend on their associate PSB making difficult to migrate them to other DBMS installations. Similarly an application programs coming from other installation needs a PSB before it can be executed. IMS/VS provides, in addition to the four basic methods of access storage, a Simple HSAM, Simple HISAM and Generalized Sequential Access Method to allow non-IMS/VS application programs to access data sets containing database data. These same facilities provide for IMS/VS application programs for accessing non-IMS/VS data.
References: 10
Reliability

Question 66.1:
Degree to which the DBMS guarantees that no transaction will leave the database in an inconsistent state even if system crashes, storage device failures or other environmental errors occur.
Assessment: All transactions, messages, etc. are recorded in the system transaction log. If a RESTART command is issued all suspended messages are taken back into the respective queues. See #19 ON BACKUP ...
References: 5(1)

Question 107:
Degree to which errors in the operating system tend to make the DBMS unavailable (fault intolerance)
Assessment: If OS/VS fails the DBMS tends to be unavailable
Mechanisms: If OS/VS fails virtual storage is copied to a dump data set, that later is used to start IMS/VS. IBM offers a power warning feature that is under control of IMS/VS, that works with a noninterrumpible power supply. This equipment is available only for systems 370 models 158 or 168. See 66.1 backup too.
References: 5(3)

Question 108:
Degree to which application program errors tend to make the DBMS unavailable—particularly, whether one program's errors tend to make the DBMS or the database unavailable to other, innocent programs, e.g. the case in which DBMS common memory is in application program address space.
Assessment: No
Mechanisms: See #65

Question 128:
Availability and quality of features for automatically detecting damaged databases
Assessment: Damaged Databases can not be detected in any automatic way. Programs trying to access a damaged database will produce and ABEND message reported to the master terminal. The master terminal, through the use of the command /DISPLAY DATABASE <options> can find the status or detect if it is damaged and initialize recovery options.
References: 22

Question 129:
Whether error handling tends to be localized, or whether errors tend to make the DBMS globally unavailable.
Assessment: Errors tend to be localized
Mechanisms: It uses checkpoints in several stages of the operation and send messages about error conditions. The
master terminal is able to terminate the session of a
terminal where the error was originated
References: 8(4)

Question 161:
Degree to which the data is protected from physical damage,
e.g. a disk head crash
Assessment: All transactions are recorded in the System Log
that is used for recovery purposes
Mechanisms: The log tape must be closed before using it with
the recovery utilities. If it is not closed it must be
used with the dump data sets when IMS/VS is emergency
restarted. If the system log can not be recovery in its
totality there exist some recovery utilities provided by
IMS/VS to such effect. The routines are used to check
status of the log tape and detect possible errors.
References: 5(1)

Question 162:
Degree to which the data is protected from effects of
hardware or software failures during updating
Assessment: When a power failure occurs the contents of real
storage is written to a data set designated specifically
for this purpose. When power is supplied and OS/VS is
active the Log Tape Recovery Utility is run for all
online Data Bases that were being updated when system
some input transactions may be lost and some duplicate
output messages may be generated after running this
procedure. IMS/VS provides with a Power Warning Feature,
that is optional.
References: 5(3)


Security

Question 22:
"Granularity" of access authorization system—whether individual access authorization specifications can be restricted to subsets of the data, and, if so, what kinds and sizes of subsets can exist
Assessment: Down to the field level
Mechanisms: The PCB, using the SENSEG macro statement can be used by the DBA to define which segments the user program can access. Within a segment the SENFLD macro statement controls field access. The PCB's required by a program constitute the PSB for that program and they are loaded first.
References: 6(2)

Question 24:
When access authorization can and cannot be specified—during database definition, when the data is entered, or any time
Assessment: Access authorization are specified as part of Program Specification Block generation which can be run during Data Base Generation or afterwards.
Mechanisms: A program Specification Block (PSB) is created to define the data storage structure required by an application program. A PSB contains one or more PCB (Program Communication Blocks), one for each hierarchical data structure the program intends to use. Each PCB defines the hierarchical (sub)structure the program sees from the physical or logical database. It specifies for each segment the kinds of access allowed by the program, that is, read only, update, insert and/or delete. The PSB, in a similar way to the DBD,is created by the running of the macro utility PSBGEN and the object code is stored in a PSB library. This PSB library is accessed by IMS/VSe at load time and before the program is given control. See # 27 this section
References: 10(5),16(1)

Question 25
Whether the authority to control access can be delegated
Assessment: Control access can not be delegated
Mechanisms: Each program has its own Program Specification Block defined by the DBA and residing in the PSB library.
References: 2(6.5)

Question 26:
Impact of security system on DBMS performance
Assessment: Security features of IMS/VSe are built-in and there is no measurement of their impact on system performance
Mechanisms: IMS/VSe data security is achieved using definition of logical databases (user views) and generation of the
PCB's for a program, which is a basic feature of IMS/VS. All programs have to have a PSB which is loaded before the program takes control. Hence this type of security can not be considered to have a separate overhead. IMS/VS terminal and transaction type security is relative simple and is invoked infrequently compared to the normal frequencies of data reference. Hence, no security feature could be said to affect performance significantly.

References: 5(4.2), 10(6.3)

Question 27:
Ease of use of security system
Assessment: Data security is easy to understand but inconveniently static. Terminal and transaction type security is easy to understand and somehow more dynamic.
Mechanisms: Data security is specified in logical database options given as part of the SENSEG macro statement of the PCB definitions. Therefore knowing how to define logical databases is sufficient to define data security. However, changing access authorizations to an existing program may require the regeneration of its PSB, which is inconvenient. Terminal and transaction-type security is specified via the SMU (Security Maintenance Utility) which can be run at any time. The master terminal provides for a dynamic control of some of the configuration details set up by SMU.
References: 2(6.5)

Question 29:
How late access authorizations are bound to an application program--at compile time, load time, or when the data access is made
Assessment: At load time
Mechanisms: When a program is scheduled for execution it specifies its PSB by means of a JCL statement. IMS/VS fetches and loads into storage the DBD's and the PCB's required from their respective libraries. IMS/VS checks for security options and determine security options for the program. It, then loads the application program and gives it control, passing the PCB's addresses as parameters.
References: 2(4)

Question 30.1:
How complex an access constraint may be--what criteria (data object name, data object value, user-supplied data, user ID, terminal ID, time of day, etc.) are available and what relationships between them can be defined
Assessment: Constraints that define access to programs include terminal ID, user ID, and data object name. The constraints, when combined, can be quite complex. For example, a specified user may only be allowed to use a particular terminal to process certain transactions using a few allowed commands. Criteria such as time of day,
data object value, and user-supplied data are not readily available. Constraints that define program access to the data are not very complex. Access at this level is controlled by user views defined by the DBA. The views can be defined down to the field level.

**Mechanisms:** Access controls on programs are set up by executing Security Maintainence Utility (SMU) after system definition. The master terminal operator (MTO), however, can modify all security profiles during normal operations, or during normal restart, subject to normal limitations imposed by system definition. Access controls on the program-data interface are defined using SENSEG and SENSFLD statements in the PCB's a program will be authorized to access.

**References:** 3(13), 4(6), 10(11)

**Question 31:**
Availability and quality of security monitoring tools, e.g. audit trail

**Assessment:** The logging facility provides a good audit trail for the monitoring of security.

**Mechanisms:** All messages entering and leaving the system, and all changes to the database are recorded on log tapes. The log tape reflects the state of the system at that point in time. The log tape is also used to restart the system after it was stopped for any reason.

**References:** 23(1.2)

**Question 32:**
How security exceptions are handled

**Assessment:** Security violations are recorded on the IMS/VS system log tape. This provides an audit trail. An error message is sent to the terminal entering the message. The master terminal is also notified when a threshold error limit is reached.

**Mechanisms:** The threshold limit can be set by specifying the number of violations before notification. (i.e. 1 to 3 invalid entries)

**References:** 10(6)

**Question 34**
How secure the security functions are

**Assessment:** The security functions are quite secure. The critical control point to the system is the master terminal operator (MTO), because the MTO can modify all security profiles during normal operations. The MTO can be controlled using LTERM security, signon verification, and by enforcing security options. This assessment reflects only the security of IMS/VS. It does not reflect the security of the operating system on which it is dependent upon.

**Mechanisms:** LTERM and signon verification security features are part of SMU. The signon verification will identify the MTO while the LTERM security allows you to specify
the commands the MTO is allowed to use. At cold or warm start time, the MTO can control many security features: signon verification security, transaction authorization, terminal security, transaction command security, and password security. This can be prohibited using SECURITY macro keywords.

References: 10(6)

Question 182:
The ability to define user-written security-control procedures, e.g. access-checking routines, special logging functions, etc.
Assessment: Yes, the user is able to write his own transaction accessing maintainence utilities, and also security functions at the program-data interface.
Mechanisms: Transaction access control involves a table which can be maintained by normal application programs. Program-data security functions can be implemented by subroutines called by the segment edit/compression exit.
References: 10(6)

Question 197:
Whether access control facilities make it possible to exclude accessors as easily as to include them—e.g. whether access can be granted to all employees in department X EXCEPT employee A
Assessment: None available
Software Quality

Question 113:
Quality of the DBMS interface to the operating system
Assessment: OS/VS has to undertake certain modifications before IMS/VS can be executed. Certain standard OS/VS system generation options and some non standard options are required before IMS/VS can be executed
References: 9(4)

Question 131:
Whether the DBMS is written in a modern programming language
Assessment: No, IMS/VS is written in Assembler Language
References: 1(3)

Question 132:
Quality of the DBMS operating system interface—in particular, degree to which operating system functions are invoked from one (or a small number) of places in the DBMS code, and the degree to which the repertoire of operating system services used is well-defined
Assessment: These services are very complicated and difficult to understand. At times is almost impossible to separate service of the OS and IMS/VS itself
Mechanisms: IMS/VS uses the OS/VS to perform such activities as Input/Output, storage management and task management.
References: 8(1)
APPENDIX 2: UNANSWERED EVALUATION CRITERIA

Application Programming Facilities

192 Availability and quality of compilers or precompilers for languages that include data manipulation features, and degree to which those compilers have knowledge of the data definition

Architecture, Computing Environment, Parameter Limits, and Multithreading

10 Memory-management services required from the operating system (e.g. dynamic memory allocation, shared memory among multiple tasks, etc.)

Data Model

116 Whether repeating fields (i.e., fields which are arrays) are allowed

Data Storage

195 Availability, quality, and efficiency of features for migrating less frequently used data to cheaper, slower storage. Also, degree of granularity and degree of transparency of such migrations, and degree to which they are performed automatically

198 Degree to which data can be placed in physical storage so as to take advantage of clustering of data values for improving speed of access

Distributed Database Facilities

177 Communications bandwidth required to run multiple, interconnected invocations of the DBMS in separate computer systems, as a function of transaction rate and database size

Efficiency

3 Difference in processing costs between inquiry and update transactions

5 Amount of dynamically-acquired processor memory required as a function of DBMS service demand

6 Degree to which the DBMS imposes paging load on a virtual-memory environment as a function of service demand (e.g. working set size)

Human Interfaces
General amount of training required to use DBMS

Local Support Facilities

Whether the source code for the DBMS is available to customers

Portability

Availability and quality of features to allow accessing other database management systems' data

Reliability

Availability and quality of built-in features to detect and recover from internal DBMS software errors

Software Quality

Bug-freeness of the DBMS software

Amount of DBMS and adjunct programs code

Degree of standardization with the DBMS—whether the code is written according to explicit conventions, degree to which internal interfaces are defined, etc.

Degree to which the DBMS and adjunct programs are composed of cleanly separated modules and/or architectural layers, with each module/layer having well-defined functions and external interfaces (modularity)

Given a relatively modular DBMS, whether the choices of which modules there are and what each module does are reasonable

Quality of the DBMS's internal control block structures

Degree to which the DBMS's design tends to allow easy growth of usage parameters and limits (e.g. number and maximum number of databases, number and maximum number of users, ...) as opposed to the degree to which the DBMS design tends to limit expansion

Whether the DBMS and adjunct programs' code modules have tractably small sizes

Quality of the DBMS and adjunct programs code—types of coding structures, use of identifiers, quantity and quality of source comments, etc.
APPENDIX 3: GLOSSARY OF TERMS

Subject Categories:

Multithreading. the running of more than one application process at the same time

Parameter limits.
  1.) restrictions on complexity of data definition or amount of data in database
  2.) other operating parameters such as maximum number of concurrent users, etc.

Architecture. What is the DBMS design like and how good is it?

Software quality. quality of code, usefulness of error diagnostics, etc.

Communications facilities. telecommunications, networking, terminal support

Computing environment. operating system and hardware

Terms:

Data definition data. data that the DBMS needs to run

Data Administration data. data that the DBA needs to do his job

Data description data. union of the above two

Transaction. A group of actions by the same user which is consistency preserving. That is, a transaction when executed alone transforms a consistent database into a new consistent state (assuming no hardware or software errors). Note that the database may be temporarily inconsistent during execution of a transaction.

Concurrency supporting. A DBMS is concurrency supporting if it allows simultaneously executing transactions to behave as if they are being executed one at a time.

Atomic. indivisible
  (a) apparently indivisible - behaves as one
  (b) actually indivisible - is one. For example, an atomic transaction behaves as if it is one action and therefore, is apparently indivisible. An atomic action, on the other hand, is actually indivisible.

Multiprocessing.
  1.) within one session ability to have greater than one simultaneously executing process owned by the same session.
  2.) within one operating system ability to have greater than one simultaneously executing process, whether or not owned by the same user.
REFERENCES

1. IMS/VS GENERAL INFORMATION MANUAL (GIM), GH20-1260 IBM Corporation 1980 Sept

2. IMS/VS DATA BASE ADMINISTRATION GUIDE (DBAG), SH20-9025 IBM Corporation 1981 March

3. IMS/VS APPLICATION PROGRAMMING: DESIGN AND CODING (APDC), SH20-9026 IBM Corporation 1981 March

4. IMS/VS SYSTEM PROGRAMMING REFERENCE MANUAL (SPRM), SH20-9027 IBM Corporation July 1980

5. IMS/VS OPERATORS REFERENCE MANUAL (OPRM), SH20-9028 IBM Corporation 1980 July

6. IMS/VS UTILITIES REFERENCE MANUAL (UTRM), SH20-9029 IBM Corporation 1980 July

7. MESSAGE FORMAT SERVICE USER'S GUIDE (MFS), SH20-9053 IBM Corporation 1981 March

8. IMS/VS PROGRAMMING GUIDE FOR REMOTE SNA SYSTEMS (SNAP), SH20-9054 IBM Corporation 1980 July

9. IMS/VS INSTALLATION GUIDE (IG), SH20-9081 IBM Corporation 1981 March

10. IMS/VS SYSTEM ADMINISTRATION GUIDE (SAG), SH20-9078 IBM Corporation 1981 March

11. IMS/VS FAST PATH GENERAL INFORMATION MANUAL (FPGI), GH20-9069 IBM Corporation 1978 April

12. IMS/VS MASTER INDEX AND GLOSSARY (MIG), SH20-9085 IBM Corporation (March 1981)

14. IMS/VS APLICATION DEVELOPMENT FACILITY - GENERAL Information, GB21-9326 IBM Corporation 1980 July

15. IMS/VS APLICATION DEVELOPMENT FACILITY - SH20-9031 IBM Corporation 1981 April

16. IMS/VS PRIMER, S320-5767 IBM Corporation 1977 January

17. IMS/VS PRIMER SAMPLE LISTING, SH20-9149 IBM Corporation 1978 September

18. IMS/VS DATA BASE RECOVERY CONTROL FEATURE: General Information GH35-0010 IBM Corporation 1981 April


20. IMS/VS ADVANCED FUNCTIONS FOR COMMUNICATIONS, SH20-9054 IBM Corporation 1977 July

21. IMS/VS PRIMER REMOTE TERMINAL OPERATOR'S GUIDE, SH20-9148 IBM Corporation 1978 October

22. IMS/VS PRIMER MASTER TERMINAL OPERATOR'S GUIDE - VTAM, SH20-9147 IBM Corporation 1978 September

23. IMS/VS PRIMER MASTER TERMINAL OPERATOR'S GUIDE - BTAM, SH20-9146 IBM Corporation 1978 September

24. IMS/VS PROGRAM PRODUCT - SPECIFICATIONS, 5740-XX2 IBM Corporation

25. OS/VS DB/DC DATA DICTIONARY - APPLICATION GUIDE - SH20-9190 IBM Corporation 1980 Sept

27. OS/VS DB/DC DATA DICTIONARY - GENERAL INFORMATION MANUAL GH20-9104 IBM Corporation 1978 Oct

28. OS/VS DB/DC DATA DICTIONARY - INSTALLATION GUIDE, SH20-9191 IBM Corporation 1980 Sept

29. AN INTRODUCTION TO DATABASE SYSTEMS, SECOND EDITION C.J Date Addison Wesley 1977

30. INFORMATION MANAGEMENT SYSTEM IBM Systems Journal Volume 16, number 2, 1977

31. INTERACTIVE QUERY FACILITY(IQF) FOR IMS/360, H20-1223, INTRODUCTION TO IQF LANGUAGE GUIDE, H20-1222 IBM Corporation

32. AN INTRODUCTION TO DATABASE SYSTEMS, SECOND EDITION C.J Date Addison Wesley 1977

32. INFORMATION MANAGEMENT SYSTEM IBM Systems Journal Volume 16, number 2, 1977

33. INTERACTIVE QUERY FACILITY(IQF) FOR IMS/360, H20-1223, INTRODUCTION TO IQF LANGUAGE GUIDE, H20-1222 IBM Corporation 1974