Chapter 9
Political Enterprise Database Management System (PDBHS or PEDBHS)

9.1- Prologue - Let us consider a polit-computer-centre, such as a polit-object, that has a large amount of polit-data kept for long periods of time *in* a computer. This data might include information about:

- Foreign-, and Intern-policy
- Polit-Organisations, Parliament's process, Government's process, etc.
- Polit-conflicts, National-conflicts, etc.
- Polit-economic, -social, -defence, -environment, -justice and law, etc.
-

Data, such as the above, that is stored more-or-less permanently in a computer we term a polit-database.

With the advent of computerize, polit-database management systems have been developed to allow data to be stored electronically, and to manage the editing, updating, accessing, and modification of both the data and the file structure itself. Database management systems are one of the most important applications of computer systems, and they are now used extensively on mainframes, minicomputers, and microcomputers.

A polit-database consists of a collection of related files that are stored on disk. Each file is composed of records that contain individual data items, called fields.

The PEDBMS manages the polit-database and allows you to access it in a wide variety of ways. On microcomputers, databases are managed with one of two kinds of software packages: File management systems or relational database management systems.

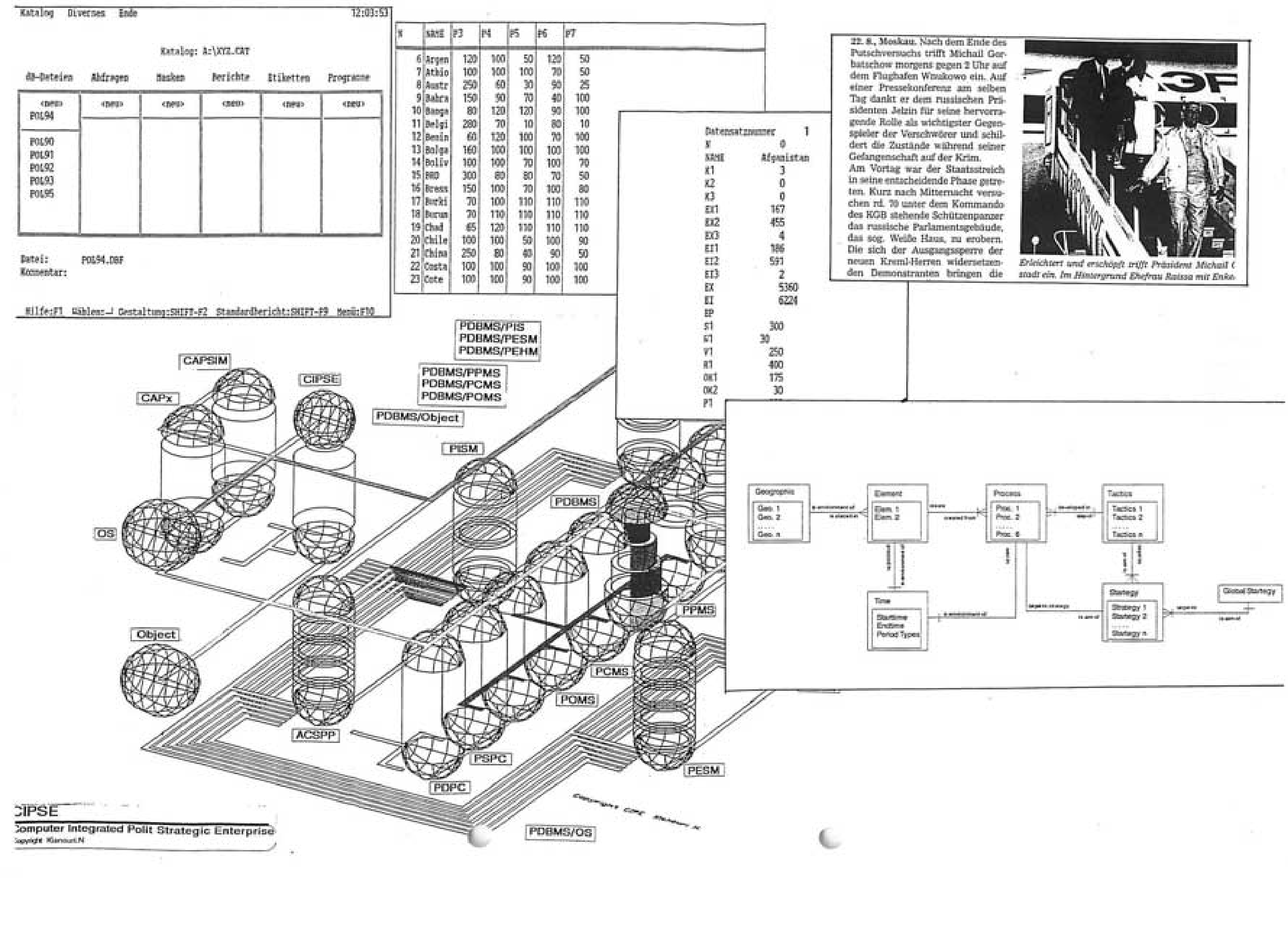


FIG.9.1. IS A REPRESENTATION OF PDBMS STATION AND INTERFACES BETWEEN CIPSE STATIONS.
THE STATION FUNCTIONS LIST AS:

1- TEXT RETRIEVAL
2- PHOTO AND GRAPHIC RETRIEVAL
3- OBJECT ORIENTED MENU FOR POLIT DATABASE
4- REPORT OF POLIT DATA
5- MENU OF POLIT DATABASE
6 - DATA MODEL.

6.2- List of Functions' of Polit-Enterprise Data base Management System (PEDBMS)

6.2.1- Polit-Data Elements:

- Polit-data, Records, files, Database
- Polit-data Hierarchy (foreign-polit-hierarchy and intern-polit-hierarchy)
- Database processing, Relationship among files
- Design database with optimal hierarchy.

6.2.2- Political Relational Database:

- The role of polit-data model
- Using the polit-data model
- Design of polit-database
- The building blocks of information engineering
- The diagrams of the documentation
- The data administrator
- Reporting at the high level
- Making object-oriented polit-database
- The polit-database dictionary
- Define The network database
- Report between database and polit-organization functions
- Define Data protect mechanism.

6.2.3- Polit-Object-Oriented Database:

- Define architectural of polit-object-oriented Database
- Programming object oriented polit-database.

6.2.3- Basic Definitions of PEDBMS:

Polit-Entity - A polit-entity is an individual polit-object / concept, or -event about which the polit-organization chooses to collect and store polit-data.

Polit-Entity Class - Is a collection of polit-entities that have similar characteristics, such as Parties, Governments, and States. Entity classes are sometimes called entity sets or entity types.

Polit-Information - Is a major corporate resource and must be managed using the same basic principles used to manage other assets, such as employees, polit-facts, equipment, and political resources. Information is an asset only if it is accurate and available needed; this can occur only if an polit-organization purposely and manages its data.

Data base - A collection of online data. The term database is more precise than data bank. Data bank refers to any collection of data, whether in the form of files, database, or an information retrieval system.
A database is an integrated collection of data, organized to meet the needs of multiple users in an organization.

A Database Application System - Includes the data definitions / stored data, transactions, and reports needed to capture, maintain, and present data from databases.

- Data Capture Applications; Capture transaction data, populate databases, and maintain the currency of data.
- Data Transfer Applications; Move data from one database to another (for example, from operational to managerial).
- Data Distribution Applications; Resulting from data analysis, essentially convert data into useful information and present them to the manager (or other users) in a readily understandable form.

Polit-Data Architecture - a structure that models the data of the polit-enterprise.

Polit-Information Retrieval - Methods or procedures used for recovering specific polit-information from stored data. Information retrieval is often seen as the core activity of all information services, and this type of software is therefore of particular importance.

There are two main types of software relevant to information retrieval: database management systems (DBMSs) and text retrieval systems.

Physical Database - A database is called a physical database if all its segments are physically present on the storage media with all their fields.

Logical Database - A database called logical database if, the segments and fields associated with it either form a subset of some physical database, or are not all physically present, with those not physically present being constructed from physically present ones.

Logical and Physical Design - In logical database design, we take the user requirements, analyze them, and produce a data model. The constructs of the relevant DBMS are then superimposed on the data model to from the input to the physical database design process.

The physical database design process is primarily concerned with storing the data as defined in the logical data model and defining access paths to the stored data.

6/.4- Political Enterprise Database Management System (PEDBMS) - A database system (DBS) provides for the secure and reliable long-term management of data and for their retrieval according to individually chosen criteria in a multi-user environment. From a structural point of view, a database system is a collection of stored data together with their description (the database) and a hardware/software system for their management, modification and retrieval (the database management system or DBMS).

Political Database management system consist of a collection of interrelated polit-data a set of programs to access that data. The collection of data, usually referred to as the database, contains information about one particular polit-enterprise. The primary goal of a PDBMS is to provide environment database information.

Polit-database systems are designed to manage large bodies of polit-information. The management of data involves both the definition of structures for the storage of information. In addition, the database system must provide for the safety of the information stored, despite system crashes or attempts at unauthorized access.

The importance of information in most polit-organizations, and hence the value of the database, has led to the development of a large body of concepts and techniques for the efficient management of data.

The database is also introduced new tools and management issues into the data processing environment. There are eight key features that are commonly considered to make up the function and quality of a database system:

- Data integration
- Application-oriented data structuring
- Consistency
- Multi-user operation
- Recovery
- Transactions
- Protection
- Data independence.

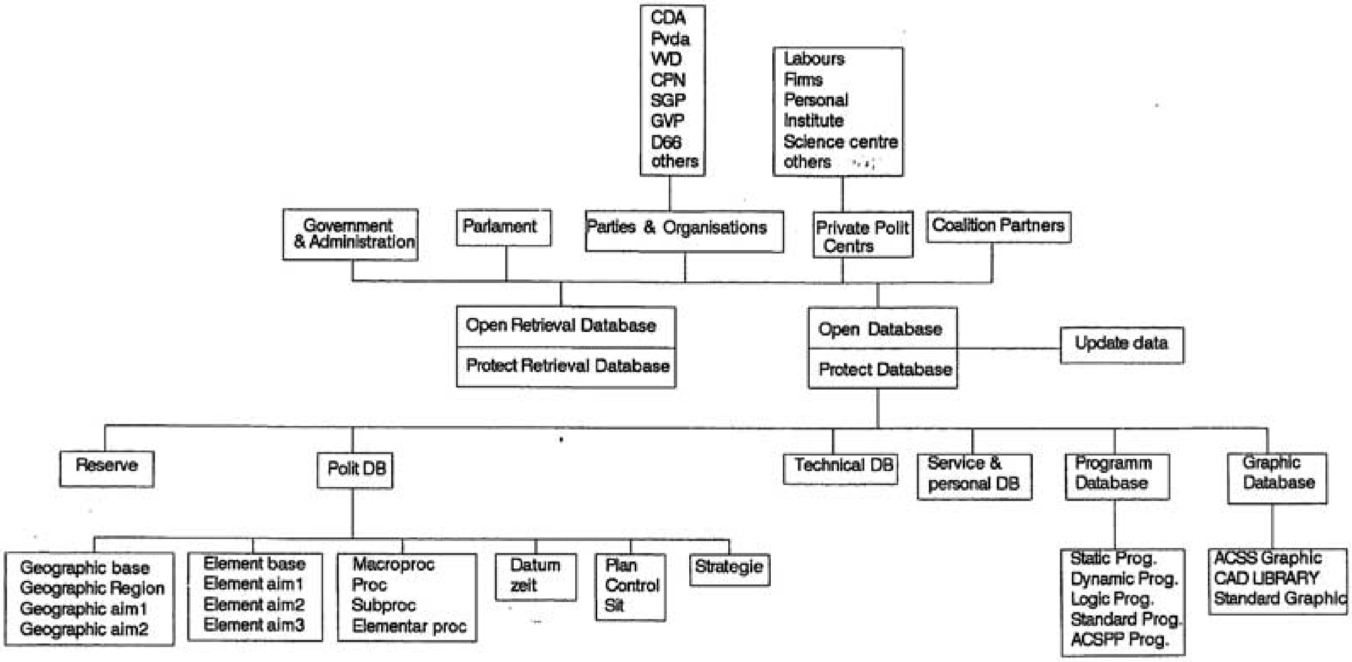


FIG. 9-2

ACSPP - Holland's Plan

FIG.9.2. IS AN EXAMPLE FOR A POLIT OBJECT ORIENTED DATABASE. THE DATABASE STRUCTURE AS OBJECT ORIENTED FORM OF POLITICAL STRUCTURE OF HOLLAND AND HER INTERNATIONAL PROCESSES HAD DESIGN.

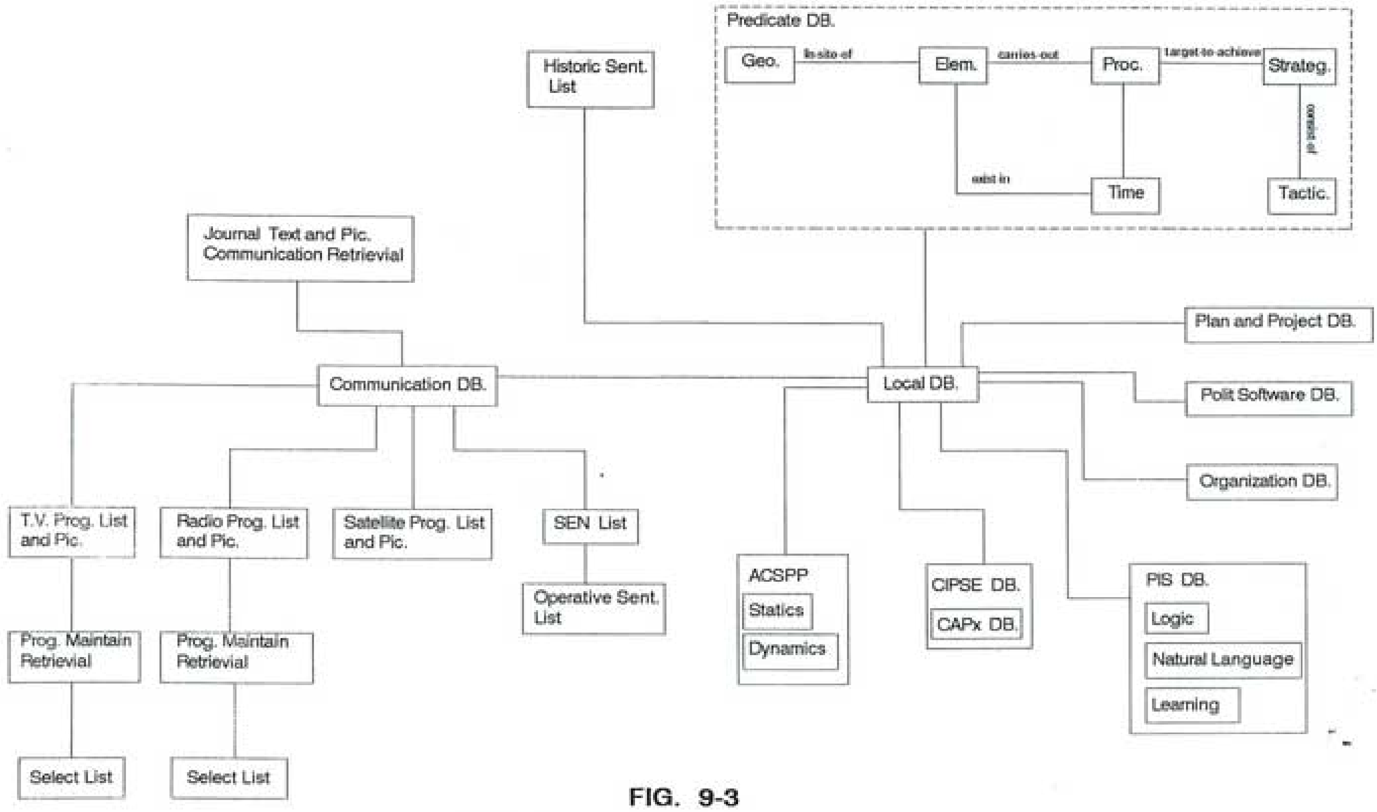


FIG. 9-3

CIPSE
Computer Integrated Polit Strategic Enterprise

FIG.9.3. IS A MORE COMPLEX POLIT-OBJECT ORIENTIED DATABASE FOR A POLITICAL STRUCTURE FOREIGN POLICY. THE MAJOR PARTS OF DATABASE ARE:

1- FIRST OPERATIVE DATABASE; THIS PART HAS RELATED WITH DAILY TRAFFIC OF POLIT DATA (FOR EXAMPLE DATA IN CODE, TEXT AND IMAGE FROM T. V. & RADIO, SATELLITE, AND OTHER RESOURCES)

2 - SECOND OPERATIVE DATABASE; THIS PART HAS RELATED WITH THE TACTICAL AND STRATEGICAL DATA (FOR EXAMPLE STATISTICAL AND DYNAMICAL DATABASE, HISTORIC AND PLAN DATABASE)

6.5- Nature of Polit-Data - Polit-data can classify as reality (the polit-real world), polit-metadata (information about data), and the actual polit-data.

Polit-Reality - Consists of the polit-organization itself, the various parts of the polit-organization, and the environment in which the polit-organization operates. Any organizations is a collection of people , facilities, and facts (or objects) that are organized to satisfy certain goals. Each organization interacts with environments and both influences and is influenced by that environment.

Polit-Metadata - Are information about the polit-data in an organization. Database administrators and other use metadata to develop logical models of an organization's entities and the associations between those entities. Metadata are stored and maintained in the organization's data dictionary/dictionary or repository or are stored with actual data to make those data more accessible.

9.6- Polit-Data **Dictionary -** A catalogue of all polit-data types, giving their names and structures, and information about data usage. Advanced data dictionaries have a dictionary function that enables them to represent and report on the cross-references between components of polit-data and polit-models.
The data dictionary is the second most important tool in the database environment; the most important is the DBMS. The data dictionary is used to record facts about objects or events in the database environment in order to facilitate communication and provide a permanent record.
In database environment, the data dictionary is based on giving information about the database itself, its contents, and its structure. The data dictionary focuses primarily on data-related components such as:

- Data elements or attributes
- Data groups, rows, or tables
- Data structures
- Databases.

The data dictionary should document the following information for the database environment:

1- Name and meaning: A unique identifier. and descriptive information that conveys the full meanings of the components. The name is used for reference and retrieval purposes, for example:

- Polit-Geographic name
- Polit-Element name
- Pol it-Process name
- Polit-Procedure
- Time name
- Quality of Tactic and strategies.

2- Physical description: The physical characteristics of the components, such as size of an attribute or the length of a table row.

3- Edit and authorization criteria: Criteria to be used to test the validity of instance of the component, such as acceptable range of values {domain} for attributes or passwords for update of a database.

4- Usage: Information about where and by whom or by what a component is used, such as the programs that can reference a given attribute.

5- Logical description: The characteristics and structure of each user view of the database, such as logical relationships among tables or table rows.
6- Procedures: Guidelines for human interaction with the database, such as for backup, recovery, and system restart.
7- responsibility: A record of the individual or organizational unit responsible for the generation and maintenance of the database component.

9.7- Polit-Data Encyclopaedia - A repository of knowledge about the polit-enterprise, its goals, entities, records, organizational units, functions, processes, procedures, and application and information system. It is populated progressively during each stage of information engineering. A dictionary contains names and descriptions of data items, processes, variable, etc. An encyclopaedia contains complete coded representations of plan, models, and designs with tools for cross-checking, correlation analysis, and validation. Graphic representations are derived from the encyclopaedia and are used to update it. The encyclopaedia contains many rules relating to the knowledge it stores, and employs rule processing, the artificial intelligence technique, to help achieve accuracy, integrity, and completeness of the plans, models, and design. The encyclopaedia is thus a knowledge base which not only stores development information but helps to control its accuracy and validity. The encyclopaedia should be designed to drive a code generator. The toolset helps the systems analyst build up in the encyclopaedia the information necessary for code generation. The encyclopaedia "understands" the modules and designs; a dictionary does not.

9.8- Types of Database - The growth of distributed processing, end user computing, and decision support systems have caused the development of several major types of databases. Five major types of databases for computer-using organizations are:

Common Operational databases; these database store detailed data needed to support the operations of the entire polit-organization. They also may be called subject area database (SADB), transaction databases, and production databases.

Common end user database; these databases store data and information extracted from selected operational and external databases. They also called polit-information databases and management databases.

Distributed database; These are databases of local work groups and departments at regional offices, branch offices, enterprising plants, and other worksites.

Personal end user database; these databases consist of a variety of data files developed by end users at their workstations.

External database; Access to large, privately owned databases or databanks is available for tie polit-end users and organizations to commercial information services networks. Data is available in the form of statistics on economic and demographic activity from statistical data banks.

The organizational data pyramid consists of operational, managerial and strategic database:

Operational Databases; Contain the political transaction history of daily political activities.

Managerial Databases; Are used primarily by middle-level managers for planning and control.

Strategic Databases; Are used by polit-managers to develop corporate strategies and seek competitive advantage.

9.9- Polit-Data Structures and Models - Data structures and data models from the underlying structures for the data base management systems (DBMSs). The data model is primarily a diagrammatical representation of entities and their association. The constraints of the DBMS are then superimposed on the data model to form the distinguishing features of each DBMS. The DBMS and its attendant software are the dominant tools in the database environment.

In political systems planning, the polit-organization seeks to determine what data architecture can developed to support its long-term data needs and assist it in achieving its goals and objectives. The data architecture is manifested as data models showing all entities that the corporation will need.

A model for polit-enterprise is a representation of real-world political-objects and -events and their associations. It is an abstraction from reality (that is, in different form) and, as such, often is simplified for ease of understanding and manipulation.

A data model for polit-enterprise is an abstract representation (a description) of the data about polit-entities, -events, activities, and their associations within an polit-organization. More liberally, a data model represents (describes) an polit-organization itself, since, for example, it is the association between user and the order submit that leads to associations between user records and order records.

Underlying the structure of a database is the concept of a data model, a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints. The relationships among many individual records stored in databases are based on one of several logical data structures or models. Database management system packages are designed to use a specific data structure to provide end users with quick, easy access to information stored in databases.

Three fundamental database models are the External data model, Conceptual data model, internal data model.

External Data Model; A description of a user's or application program's view of data in a database. Also called user view.

Conceptual Data Model; The overall logical model of an organization's database, which is independent of a particular DBMS.

Internal Data Model; A data model that describes an entire database using a technologically-dependent style.

9.9.1- Hierarchical Data Model - This technique arranges records in a treelike arrangement, with records related to one another in a top-to-bottom manner, much as a family tree would appear in chart form. In fact, a hierarchical structure is sometimes referred to as a tree structure. Each record is broken into subdivisions, called segments.

The basic concepts of the hierarchical model are as follows. A database consists of an arbitrary number of similarly structured hierarchies (called database records) of records in the traditional sense (that here are called segments).

The schema of a hierarchical database consists of a number of database descriptions (DBD), one for each hierarchy. In turn, a DB! consists of the following statements:

- A DBD statement specifying, among other things, the name of the database,
- A DATASET statement for each segment type in the hierarchical schema, with type name and device type,
- A SEGM statement for each segment type in the hierarchical schema, with type name and parent name,
- Following each SEGM statement, one or more FIELD statements, one per segment field.

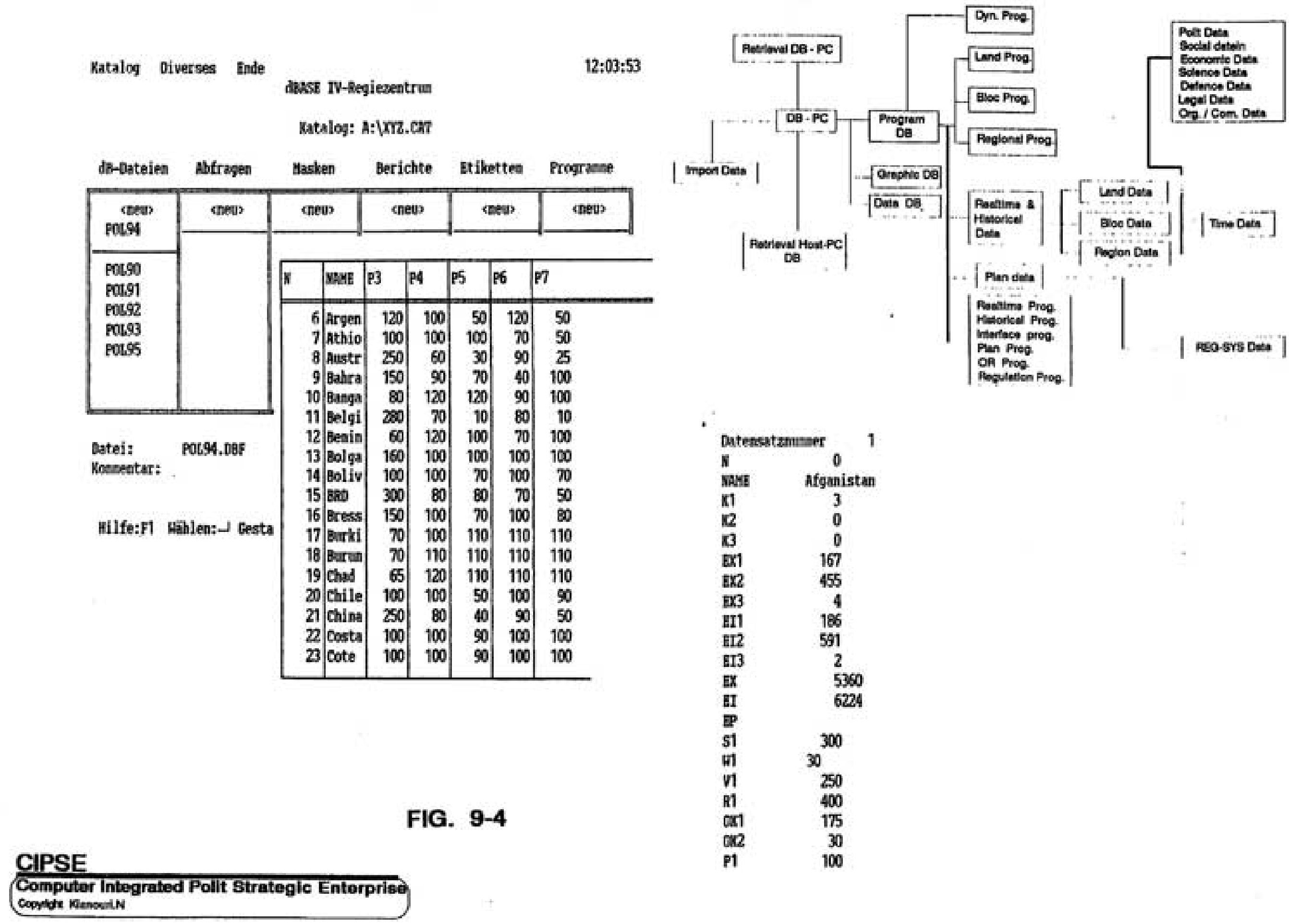


FIG. 9-4

FIG.9.4. IS AN EXAMPLE FOR A HIERARCHIAL POLIT DATA MODEL, AND RESULTAT OF DATA MODEL, WHICH WITH DBASE VI INTEGRATED PACKAGE, HAS DONE.

9.9.2- The Network Data Model - Two of the shortcomings of the hierarchical model - failure to account for a network-like topology, and a highly restricted set of entrance records - are overcome by the network data model. To start again with the structure concepts:

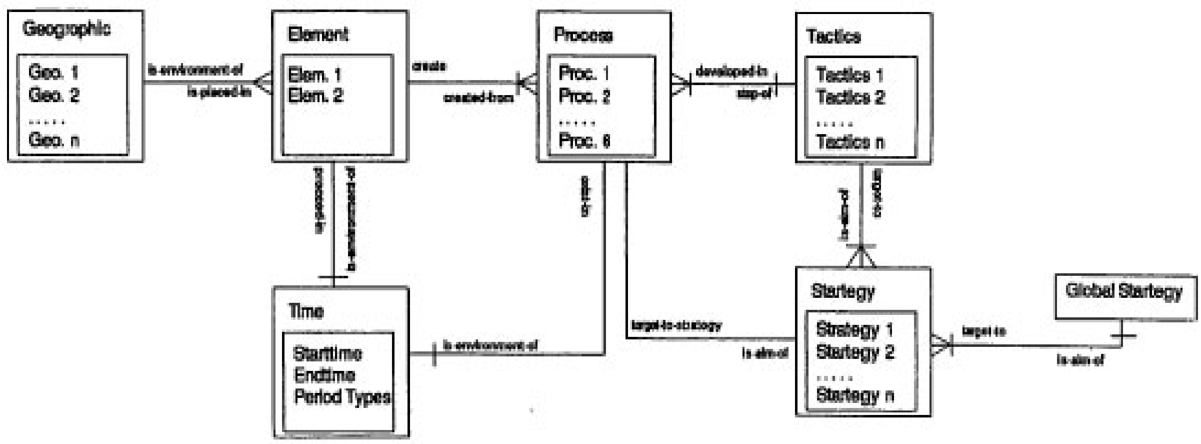
A database consists of an arbitrary number of records and interrelationships between them. Each record is of a certain record type that determines its composition from fields named by attributes. Data in the network model are collections of records, which can be viewed as pointers. The records in the database are organized as collections of arbitrary graphs. A schema of a network database consists of the following sections:

- The first is an introductory SCHEMA statement which includes a schema name equal to the database name.
- An AREA section declares the areas into which the database is to be divided, by name and access characteristics.
- Next is a Record section in which all record types are defined. In turn, the definition of a record type consist of two subentries, one describing general characteristics such as type name, associated area(s) , access authorization, identification within the area, and the other giving the record structure in terms of a hierarchical field arrangement.
- Finally, there is a SET section with a declaration for each set type. Again, such a declaration consists of two subentries. One outlines general properties such as type name, access authorization, owner type, and ordering criterion. The other subentry gives a detailed description of the properties of the member record with respect to a set instance of this type: member type, sort key, insertion and retention of this class, and rules for determining the set instance when a record of the member type is to be inserted into a set of this type.

9.9.3- Entity-Relationship Data Model - The data model that is most commonly used for enterprise, external, and logical analysis is the entity-relationship data model. Many Computer-Assisted Software Engineering (CASE) tools and databases analysis aids include this modelling convention, and much attention has been given it, with man~ extensions proposed since the original version.

The entity-relationship (E-R) data model is a simple extension of the type of graphical notations. E-R uses a few special symbols, such as the diamond, to include the relationship.

The entity-relationship data model was reviewed as the primary tool used today by professional data and systems analysts for external and conceptual data modelling. Various data semantic~ (cardinality, degree, uniqueness, time, class-subclass, aggregation, and existence) were introduced and it was illustrated if and ho," different data modelling styles deal with such characteristics of data.



CIPSE

Computer Integrated Polit Strategic Enterprise

Copyright Kianouri.N

FIG. 9-5

FIG.9.5. IS AN EXAMPLE FOR ENTITY-RELATIONSHIP DATA MODEL, ENTITY HAS SHOWS AS GEOGRAPHIC-LOCATION, POLITICAL-ELEMENT, -PROCESSES, TIME, STRATEGIC VALUE, AND TACTICAL VALUE.

9.9.4- Relational Data Model - This type of data structure has become extremely popular in recent years. The relational structure is, in one sense, a type of network structure because it provides for grouping of related data. This approach requires data to be arranged in columns and rows within two-dimensional tables. The columns correspond to fields within records and the rows to records themselves. These tables are called relations or flat files. By establishing connections between the relations, different items of data· can be interconnected rather liberally. This design gives the user the illusion that everything is stored in one large file. It is this "logical view" of data makes relational DBMSs so popular.

To the best of our knowledge, nobody mistakes a relational database for an object-oriented database.

The relational model is different from other models not only in architecture but also in the following ways:

1- Implementation independence; The relational model logically represents all relationships implicitly, and hence, one does not know what associations are or are not physically represented by an efficient method. Relational shares this property with E-R.
2- Terminology. The relational model uses its own terminology, most of which has equivalent terms in other data models.
3- Logical key pointers; The relational data model uses primary (and secondary) keys in records to represent the association between two records, whereas E-R (and network and hierarchical) use arcs between entity boxes.
4- Normalization theory; Properties of a database that make it free of certain maintenance problems have been developed within the context of the relational mode.
5- High-level programming language; Programming language have been developed specifically to access data bases defined via the relational data model; these languages permit data to be manipulated as groups or files rather than procedurally one record at a time.

9.9.5- Political Object-Oriented Databases Model - A data model is a logical organization of the real-world objects (entities), constraints on them, and relationships among objects. A data model that captures object-oriented concepts is an object-oriented data model. An object-oriented data base is a collection of objects whose behaviour and state, and the relationships are defined in accordance with an object-oriented data model. An Polit-Object-Oriented databases is based on object oriented programming languages and includes type/instance, identity, encapsulation, and inheritance all of which are characteristics of object-oriented languages. An object-oriented database system is a database system which allows the definition and manipulation of an model for the next-generation database applications, such as CAD/CAE/CASE/CAM/(CAx) systems, knowledge-based systems, multimedia information systems, and advanced human-interface systems.

An **Object** is an inseparable package or capsule of data definition and values and the procedures (sometimes called methods) that act upon the data. Because each object has its definition as part of itself, each object can be unique from objects. An object may contain any type of data: text, numbers, voice, or pictures. Further an object may contain other objects, which may contain other objects, and so forth. Finally, an object may be a super class or subclass of other objects, as discussed before under the generalization semantic.

An **Object Class** is a logical grouping of object instance that share the same attributes and operations.

An **Object instance** is one member of an object class.

An **object-oriented data model**, simply put, is a set of object-oriented concepts for modelling data. Object-oriented concepts have been embedded in various programming languages and knowledge presentation languages. A survey of object-oriented programming languages and knowledge representation languages can bring out a set of commonly accepted and fundamentally important data modelling concepts. This set of modelling concepts forms a core object-oriented-data model. The core model application, and to shed insight into changes an object-oriented-data model forces on the architecture of a database system.

An object-oriented data model then may be defined as a core mode] augmented with semantic-integrity constraints and a number of semantic relationships.

Core Model Concepts - Object and object Identifier; any real-world entity is an object, with which is associated a system-wide unique identifier.

Attributes and methods; an object has one or more attributes, and one or more methods which operate on the values of the attributes. The value of an at tribute of an object is also an object. AI attribute of an object may take on a single value or a set of values.

Encapsulation and message Passing; Messages are sent to an object to access the values of the attributes and the methods encapsulated in the object. There is no way to access an object except through the public specified for it.

Class; all objects which share the same set of attributes and methods may be grouped into a class. An object belongs to only one class as an instance of a metaclass.

Class Hierarchy and Inheritance; The classes in a system from a hierarchy or a rooted directed a cyclic graph, called a class hierarchy.

Differences - There are important differences between an object-oriented data model and relational data model. An object-oriented data model includes the object-oriented concepts of encapsulation, inheritance, and polymorphism; these concepts are not part of the conventional models of data. The difference between an object-oriented database system and non-object-oriented database system i:: that an object-oriented database system can directly support the needs of the applications that create and manage objects that have the object-oriented semantics, namely, object-oriented programming languages or applications designed in an object-oriented style.

9.9.6- Conventional Database system - Let us overview several of the well-known shortcomings of the conventional database technology.

A conventional data model, especially the relational model, is too simple for modelling complex nested entities, such as design and engineering objects, and complex documents; conventional database system do not provide mechanisms, such as configuration management. to present and manage such entities.

Conventional database systems support only a limited set 01 atomic data types, such as integer, string, etc.; they don't support general data types found in programming languages.

The conventional data model does not include a number of frequently useful semantic concepts, such as generalization and aggregation relationships; this means that application programmer: must explicitly represent and manage such relationships in their programs, since the database system dose not provide the necessary functions.

The performance of conventional database systems, especially relational database systems, is unacceptable for various types of compute-intensive applications, such as simulation programs in computer-aided design and programming language environment.

Application programs are implemented in some algorithmic programming language (such as COBOL, Fortran, or PL/l) and some database language embedded in it (such as SQL, DL/l, or CODASYI DML). Database languages are very different from programming languages, in both data model and data structure. This impedance mismatch problem motivated the development fourth-generator. languages.

Conventional database systems do not support facilities for representing and managing the temporal dimension in databases, including the notion of time and versions of objects and schema, and change notification.

9.7- Data Base Languages - Conventional database languages to allow application programmers or end-users to define and manipulate the database. Database languages can often be embedded in other programming languages for greater flexibility. an effort is also afoot to make them easy to use, so that nonprogrammers can manage the normal data of the world of policy.

A conventional database language consists of three components (or sublanguages): a data definition language, a data manipulation language, and a data control language.

9.7.1- Data Manipulation Language (DHL) - The language that the programmer uses to cause data to be transferred between his program and the database. The data manipulation language is not a complete language by itself. It relies on a host programming language to provide a framework for it and provide the procedural capabilities required to manipulate data.

The data manipulation language (DML) includes facilities to express queries and updates (replace, insert, delete) against the database. In relational databases, a query is an expression of a subset of the database to be retrieved which satisfies user specified search conditions.

9.7.2- Data Definition Language (DDL) - Allows the specification of the database schema. In relational databases, the schema of a database is the specification of a set of relations: the name of each relation, the name of each column of a relation, the domain (data type) of each column, and integrity constraints on the domain.

The result of compilation of DDL statements is a set of tables which are stored in a special file called data dictionary.

9.7.3- The Data Control Language (DCL) - Includes facilities for protecting the integrity of the database and for managing resources of the system. The integrity feature of a DCL includes transactions; a limited from of semantic integrity constraints, for most systems; and. authorization. A transaction is a sequence of reads and writes against a database, such that the sequence is treated as an atomic action against the database.

9.8- PEDBMS Interfaces with other CIPSE Stations:

9.8.1- PEDBMS/PISE (System Engineering of PDBMS or Political Database Development) - Developing small personal database is relatively easy using microcomputer database management packages. However developing a large database can be a complex task. In man; companies, developing and managing large corporate databases is the primary responsibility of the database administrator and database design analysts. They work with end users and systems analysts to determine (1) what data definitions should be included in the database and (2) what structure or relationship should exist among the data elements.

Designing and creating a database are analogous to building house. The form, function, and cost should judiciously optimize. Ir. a database system, the data should be organized in such a way that it meets the user's needs and can be economically stored· and retrieved with ease. The major steps can list as following components;

- Analyzing Data and Polit-User Needs; the first step in the database development process is analysis: analysis of available data and our requirements leading to identification of basic data items in the database that will meet our objectives as users. Since this analysis is based on our logical data manipulation and retrieval needs and not on any specific computer hardware and software, this phase is also known as the logical or conceptual design phase.
- Implementing the Database Design; The next step is to implement the conceptual database design in a specific computer hardware-software configuration.
- Entering Data; once the database structure or framework is created, the next step is to enter data in appropriate fields. This process of creating database records is known as data entry.
- Saving the Database; therefore, it is good practice, as with any data entry operation, to save entered data periodically on a hard or floppy disk.
- Editing the Database; Data change over time. Consequently, the databases reflect current status and meet our needs. This is achieved through the database editing process. All activities for example adding, deleting, updating, and correcting records are carried out as part of the database editing process.
- Utilizing the Database; The last and perhaps most useful step from the user's perspective is utilization of data entered in the database. The entire database can be displayed on screen or a hard copy generated for record keeping.
- Data security and Integrity; One of the primary objectives of creating databases is to improve the accessibility and dissemination of information that will be useful to a large number of people in an organization. With this, however come the issues of data security and integrity. These issues are especially critical since databases are increasingly accessible over computer networks.
- Characteristic of a Database Management Program. Typical database development project can be build in four steps, polit-enterprise modelling, logical database design, physical database design and creation, and programming.

1- Polit-Enterprise Modelling - List as;

- Analyze current data processing
- Analyze the general political database needs
- Plan database development project
- Develop preliminary conceptual data model.

A database development project begins with enterprise modelling which is where the scope and the general contents of the database are sketched. This step involves reviewing current systems, analyzing~ the nature of the policy area to be supported, describing the date needed at a very high level of abstraction, and planning the rest of the project.

2- Logical Database Design - List as;

- Normalize transactions and reports
- Integrate views into conceptual data model
- Design screens, reports, and applications
- Identify data integrity and security requirements.

Logical design performs the detailed review of individual] reports, transactions, screens, and so on the determine exactly what data are to be maintained in the database.

Logical Data Elements, consist of five elements character, field, record, file, and database;

- Character; A single element of a group of data, such as a letter, number, or any special character.
- Field; One or more characters representing a subdivision of a record.
- Record; A collection of data fields treated as a unit.
- File; A collection of related records, arranged in sequence according to a key value contained within each record.

3- Physical Database Design and Creation - List as;

- Define database to DBMS
- Decide on physical organization of data
- Design programs.

In physical design and creation, the database developer decides on the organization of the database on computer storage (usually disk) and defines the physical structures to the database software (the database management system). Programs to process transaction and to generate anticipated management information and decision support reports are mapped out.

4- Programming - list as;

- Code and test programs
- Complete database documentation
- Install database and convert from prior systems

The final step is programming in which the programs are written, tested, and implemented. Programming may occur in standard programming languages (like COBOL or C), in special database processing languages (like SQL or the DBASE IV query languages), or via special purpose fourth-generation languages to produce stylized reports and screens, possibly including graphs.

Database Design - Is the process of developing databases structures from user requirements for data. It starts with requirements definition, which identifies user needs (present and future) for data. It then proceeds by translating these user requirements first into a logical, then a physical database design. The resulting design must satisfy user needs in terms of completeness, integrity. Performance constraints and other factors.

The database design process is closely related to the three schema architecture. A simplified version of the three-schema-architecture is the following:

1- External level; At this level we identify user views such as reports, displays, and transactions. These views represent use] requirements in a technology-independence manner.

2- Conceptual level; At this level all user requirements are merge' into a single logical database description that is technology independent.

3- Internal level; At this level we develop one or more internal data models (or schemas) that define the database in a technology dependent manner.

The four steps in database design are: requirements definition, conceptual design, implementation design, and physical design.

1- Requirements design - The purpose of requirements definition is to identify and describe the data that are required by users in the organization. This step is related to the external level since the identify and model users views.

2- Conceptual Design - The purpose of conceptual design is to develop a conceptual data model (or information architecture) that will support the diverse information needs of users throughout the organization. Conceptual database design is a data-driven process that is, it is completely independent of hardware and software implementation details. Conceptual design corresponds to the conceptual level of the there schema architecture. Conceptual database design is the process of constructing a detailed architecture for a database that is independent of implementation details, including the DBMS that is to be used. Conceptual design (which also called logical design) is the second phase of the database design process. The objectives of conceptual database design are structural validity, no redundancy, simplicity, share ability, extensibility, and integrity.

Five-step processes for conceptual database design are following:

- **Develop Conceptual Data Model;** during this step, E-R diagrams are converted to relations. This step might also be considered part of implementation design.
- **Transform Data Structures to Relations;** During this step, kkkkkkE-}i diagrams (or other logical data models) are converted to relations. This step might be considered part of implementation design.
- **Normalize the Relations;** during this step, the relations that were derived in the previous step are normalized.
- **Integrate the Views;** View integration is the process of merging individual user views (in the form of E-R diagrams or 3NF relations) into an integrated data structure (or conceptual schema).
- **Develop action Diagrams;** Action diagrams are high-level definitions of data operations that maintenance a database in a current and consistent state. Typical database operations add and delete records, modify records, and produce output in the form of reports and displays.

3- Implementation Design - The purpose of implementation design is to map the logical data model into a schema that can be processed be a particular DBMS. First, the conceptual data model is mapped into a hierarchical, network, or relational data model. Then DBMS process able schemas and sub schemas are developed using the date description language for the DBMS to be used.

Implementation design is concerned with mapping the conceptual data model (which may be expressed as objects, E-R diagrams, 3NI relations, and so on) into logical database structures~ (hierarchical, network, or relational data models). Physical database design is concerned with mapping the logical database structures into an internal model consisting of stored records files, and other physical structures. The objective of implementation design is to develop logical database structures that accurately reflect user needs for information and that can grow and evolve as information needs change. The objective of physical design is to implement the database to meet user needs in terms of performance, security, integrity, and related factors.

Implementation design is required especially when the target DBME support a hierarchical or network data model. In this case, the logical data model must be mapped to either hierarchical or networ1 structures. Mapping to a hierarchical data model involves identifying the root node types, identifying dependent node types, and resolving multiple parentages. Mapping to a network data mode] includes defining records types, defining sets, eliminating redundant keys, and defining record access strategies. Although implementation design is relatively straightforward, it must be done carefully since the resulting data models must accurately represent information requirements and provide a stable basis for future represent.

4- Physical Design - Physical design is the last stage of database design. In this stage the logical database structures (normalized relations, trees, networks, and so on) are mapped to physical storage structures such as files and tables. Indexes are specified, as well as access methods, record blocking, and other physical factors.

Many of the decisions in physical database design depend on estimates of the size and probable usage patterns of the data. Data volume analysis is the process of estimating database size by estimating the number and size of each database entity or record type. Data usage analysis identifies the major database transactions and the access paths required for each transaction.

One of the most interesting problems in physical database design is deciding on the distribution of data among the nodes of a network. There are three basic data distribution strategies: centralized, partitioned, and replicated. With the partitioned approach, the central database is partition is assigned to a particular site. With a replicated approach, a full copy of the database is assigned to each site in the network. A hybrid approach combines the elements of both the partitioned and replicated approaches. The data distribution decision can be analyzed using information describing data usage and the characteristics of the network.

9.8.2- PEDBHS/PPHS (Managing Project of PEDBMS) - In this section we describe a methodology (or process) for Database planning. Although there is no standard methodology for IS planning, most methodologies have certain techniques in common:

1- A Top-Down approach - Planning begins with the policy and the data it uses and leads to ideas for specific projects.

2- Modelling - Planners define models of political-functions, date use, and related factors, often in matrix form.

3- Involvement and accountability - A planning team is established and planning involves all levels of management.

4- The limits - Because information technology changes so rapidly planning should be completed quickly- in three to six months possible. '

Before the planning process commences, three important preliminary steps must be accomplished:

1- Top management commitment - Top management must be full1 committed and prepared to become actively involved in the planning process.

2- Project team - A project team comprised of user-managers and information systems specialists should be appointed to perform strategic information systems planning.

3- Planning methodology - The team should select a planning methodology that is consistent with corporate needs and that is supported by comprehensive CASE tools.

Most of planning steps require extensive interviews with all managers in organizations.

1- Understanding policy direction - Define and/or understand policy goals, strategies, and critical success factors.

2- Policy modelling - Develop formal models for data, functions, and users.

- **Polit-User modelling:** Develop models of the polit-organization showing the organizational components, decision made at each level, the flow of information, and the need for information by each manager.
- **Function modelling:** Define polit-entities and model the relationships among entities.
- **Data modelling:** Define polit-data model.

3- Model integration - Combine the models from the previous step through a series of matrices into an integrated polit-model.

4- Design technological architecture - Design architecture (hardware, data communications, database, and organization, systems software) to support the strategic direction of the organization over the planning horizon (at least three to five years).

5- Strategy formulation - Combine the results of the previous steps with. an analysis of existing systems to identify gaps to be filled with new systems projects.

6- Recommendations and priority setting of projects - Identify new systems projects to fill the identified in the previous step, and establish priorities for the projects.

7- Tactical planning projects - Develop project schedules and resource allocations for new projects.

When all of the planning data had been entered into CAS] databases, the planning team proceeded to analyze the data anI integrate the various models. This step included constructing aJ entity-relationship diagram and analyzing the planning matrices.

9.8.3- PDBMS/PECHS (Managing Communication of PDBMS) - A date communication system transmits data from one location (called the source) to another location (called the receiver). To illustrate, remote terminal transmits data to a centralized computer for processing. The data to be transmitted, referred to as the message, are entered into the sending terminal and stored. When the terminal is ready to transmit the data, a communications interface device: such as a modern, converts the input data to signals that can be transmitted over a communications channel.

This communication can process with help of different communication or direct after database packages.

Communication systems include databases which must be managed ant manipulated, often dynamically in real time. This data may include tables defining objects of interest, their characteristics, locations, and the heading and speed of moving objects. Seldom are commercially available database-management systems adequate for real-time operation, in part because data input, reorganization, and output functions may not operate concurrently. In a typical Communication system, parts of the data may be rapidly time variable, requiring data-base-management processes which involve a minimum of the comparison, manipulation, and sorting characteristics of most data-base software. Data may need be available for display instantly its arrival, even while it is being compared with previously stored data. For the most part, such software is ad hoc to the application.

9.8.4- PDBHS/PEOHS (Managing Organization of PDBMS) - There are two distinct aspects of organizing the corporate database. The first is the policy aspect, identifying which data is relevant to the polit-computer-centre, its source and method of capture, and the interrelationships among the data items. The second is the technical aspect, storing data on computer-readable media in a form readily accessible by the corporate decision-makers. The policy tasks of organizing the corporate database required the creation of the relatively new position, chief information officer (CIO), and the more traditional data administration function.

The CIO is the executive in charge of the information systems department and is responsible for formulating an information strategy that includes all systems development, computer operations, and communications planning and operation.

Database Administrator is a person with an overview understanding of an organization's data. The function is responsible for the most cost-effective organization and use of an enterprise's data resources. The data administrator is ultimately responsible for designing the data model and obtaining agreement about the definitions of data which are maintained in the data dictionary.

A database administrator has control on one or more databases, which controls the physical and use of these databases. It is often better to use two people: A data administrator, who manages. The architecture and logical model, and a database administrator, who design the physical aspects of the database.

The data administration function links computer systems and the policy functions they are designed to serve. The group responsible for data administration builds and maintains the corporate data model. A properly constructed data model places the system to be developed into a proper policy perspective. This model is instrumental in the preparation of the information system department's strategic plan.

Database Security - Is concerned with protecting a database agains1 accidental or intentional loss, destruction, or misuse. DBM~ software provides security control through facilities such as use1 views, authorization rules, encryption, and authentication schemes.

9.8.5- PDBHS /PIS (Intelligence PEDBMS) - Large operations driven by large amounts of data. To run a factor efficiently requires that the data be readily accessible to those who need it to perform their jobs. AI techniques and relational databases in particular, can provide many of the features and functions that are desirable in managing large databases. These include:

- Natural language interface which permit access by no technical users.
- The ability to support frequent changes in the database and in user needs.
- Expert aids for designing and building a database application
- Efficient management of computer resources in the acquisition, preparation, manipulation, and storage of enterprising data

There are many common applications for database management systems in polit-enterprising today which should only expand in the future as intelligent systems become more readily available. Some applications are:

- Solution schedules and status
- Process description and routings
- Technical references (e.g., manuals and procedures)
- Management report.

9.8.6- PDBHS/PES (Expert PEDBHS) - It is logical to define an Expert Database System (EDS) as a tool for developing application~ requiring both a DBMS and one or more ES's. There are clearly many applications which require such as a tool. An EDS is defined here as a system for developing applications requiring knowledge-directed processing of shared information.

An expert data-base model, as a specialized expert system, involves the use of artificial intelligence technology. The goal in this is to provide for data-base functionality in more complex environments that require at least some limited form of intelligent capability. To allow this may require adaptive identification of system characteristics, or learning over times as experience with an issue and the environment into which it is embedded increases. The principal approach that has been developed to date comprises an object-oriented data base (OODB) design with any of the several knowledge representation approaches useful in expert system development.

9.8.7- PDBMS/PESM (Managing Software of PEDBMS) - The examples for databank packages that can use in CIPSE are:

DBASE 4; Integrated rational data management system that combines a data management system an application development languages & 4GL development tools. A DBMS, or database management system, such as dBase 4 is a set of programs that allows you to manage data using a computer. You may add, delete, change, sort, search for or calculate information from a database using a DBMS.

As the DOS tutorial indicated, it is possible for you to use an application effectively without total proficiency in DOS commands. Some important characters of DBASE 4:

Source code Language	C
Number of fields per record	256
Number of characters per text fields	255
Number of character per record	4000 bytes
Number of records per fiels	1 billion
SQL compatibility	IBM SAA
Number of sort fields	1024 bytes

Database 2 (DB2); Database 2 (DB2) is a commercial and relational database system available from IBM for use on large mainframe~ running the MVS operating system. DB2 interacts with three different subsystems running under which provides a form of transaction management. These there system - MVS, TSO, and CICS (Customer Information Control System) _ predate DB2.

DB2 provides a modern relational database with the Structure Query Language (SQL) interface. The product offers many functions: for both transaction processing and decision support applications. DB2 is IBM's strategic database product for its largest systems, which support IBM's biggest and most influential customers.

A CIMS environment enhances and simultaneously constrains the development of systems. It constrains development because it provides the structures and architectures that are vital to a highly productive environment. At the same time, it enhances development by allowing designers and implementers to concentrate their efforts on creative and responsive solution to policy problems rather than reinvent ting interfaces to the technical environment.

True integration in CIMS environment means that systems will be built in new more rigorous ways.

SQL/DS; SQL/data System is an intermediate-size commercial database system available from IBM. Its scope is between that of mainframe database systems (for example, DB2 and IMS) and personal computer database systems (for example, dBase). SQL/DS runs under either of two IBM operating systems: DOS/VSE or VM/SP.

An important problem for SQL/DS is in its ability to extract data from IMS databases using the DL/I query language of IMS. The DL/I extract feature allows data from a (hierarchical) IMS database to be copied into part of a (relational) SQL/DS database. Once the data is stored in the SQL/DS system, it can be queried using SQL. It is not possible, however, to modify the IMS database using SQL.

The data definition of SQL/DS includes the usual SQL statements: create table, create index, create view, and so on. In addition, SQL/DS has statement that allow database administrator to control the physical database structure.

IMS; Is an IBM program product for the MVS environment (like DB2, though of course unlike DB2 it is hierarchic, not relational).

The full name of the current version of IMS is "information Management System/Virtual Storage" (IMS/VE). The IMS Database Manager contains tools for users to:

- Define hierarchical database structures
- Create database
- Access and maintain databases
- Reorganize database
- Recover and reconstruct data.

Within a database, IMS defines a hierarchical relationship among segments. Database administrators can establish logical:

Relationships between segments in different hierarchical structures. IMS offers a powerful and complete set of utilities to recognize and recover physical database. The IMS Database Manager's checkpoin1 and restart functions integrate well with the transaction Manage} and provide much higher integrity than CICS. However, the Database Manager is very different from IBM's relational database, DB2, and thus offers limited integration.

POSTGRES; Is the best-known next-generation database system which is based on the extended relational approach. The POSTGRES data mode] includes some basic object-oriented concepts. It includes a, relation' hierarchy with multiple inheritance. Further, it introduces a ' procedure' as a valid type for an attribute of 8 relation; a procedure is a method. The value returned from a procedure may be the result of evaluating queries against other relations. An attribute to type procedure allows POSTGRES to support the aggregation concept (nested relations).

One of the noteworthy aspects of POSTGRES is the proposed for historical data, besides versions. POSTGRES is one of only a few systems, along with Vision, designed to support the temporal dimension of data.

Another novel aspect of the POSTGRES is the incorporation of advantages in hardware technology into database system architecture.

CPISE/SAA Database Management System - SAA includes a relational database that has a structured Query Language (SQL) interface. Such databases are rapidly growing in popularity in the System/370 environment, but are not dominant today. Many databases used for commercial applications are based on 1970’s technology, including the hierarchical, network, and inverted-list technology found in IBM's IMS, Cullinet's IDMS, and software AG's Adabas.

Typical applications run in a transaction processing subsystem and interact with a database system. Parts the applications may also run in a time sharing or batch mode. In theory, any database that provides an SAA SQL interface is considered to be compliant with the architecture. Third-party products, such as Oracle, may become prominent in SAA-compatible systems.

The IBM SAA database products for system/370 are DB2 (DataBase 2) for MVS and SQL/DS (Structured Query Language/Data System) for VM.

SAA includes a relational database that has a Structured Query Language (SQL) interface. DBMPS is a layer of software between the physical database and the users of the polit-system. All requests from users for access to the database are handled by the DBMPS. Retrieving data from updating data in such file or table, are facilities by the DBMPS. In the other words, the DBMPS provides users with a view of the database that is elevated somewhat above the hardware level, and supports user expressed in terms of the higher-level view. DBMPS, software, such as DB2 or IMS, that manages a data base.

9.8.8- PEDBMS/PEHM (Managing Hardware of PEDBMS) - Turning now to the issue of technical resources, those that have the biggest effect on the performance of a DBMS other than the DBMS implementation itself are the peripheral storage media used, and the operating system supporting it. Technical characteristics of typical peripheral storage media can list as, with storage capacity, storage cost, access time, and write ability the decisive factors. As far as operating systems are concerned even the more modern systems are not optimally disposed towards DBMS: As a consequence, most DBMS bypass many of the high-level services that an operating system provides, and re- implement some of these in accordance with their own needs on the basis of lower-level services. Unfortunately, this~ makes a DBMS especially vulnerable to revisions in the underlying operating system. Other services such as virtual memory management cannot be bypassed at all although they deter mental to DBMS performance.
Since the issues involved are highly technical we shall not go into details here, expect for two aspects that may at times become visible to the user:

File management; Ultimately *I* the database consisting of raw data, data schema, and auxiliary to support associative access must be mapped to number of files. When a DBMS is installed, and a database is created, the files needed must be storage space organization, record formats and lengths, and buffering techniques, to name just a few.

Task Management; Today's operating systems are based on the principle of concurrent execution of processes. Consequently, a straightforward organization of transaction of processes. Consequently, a straightforward organization of transactions would associate a process with each transaction.