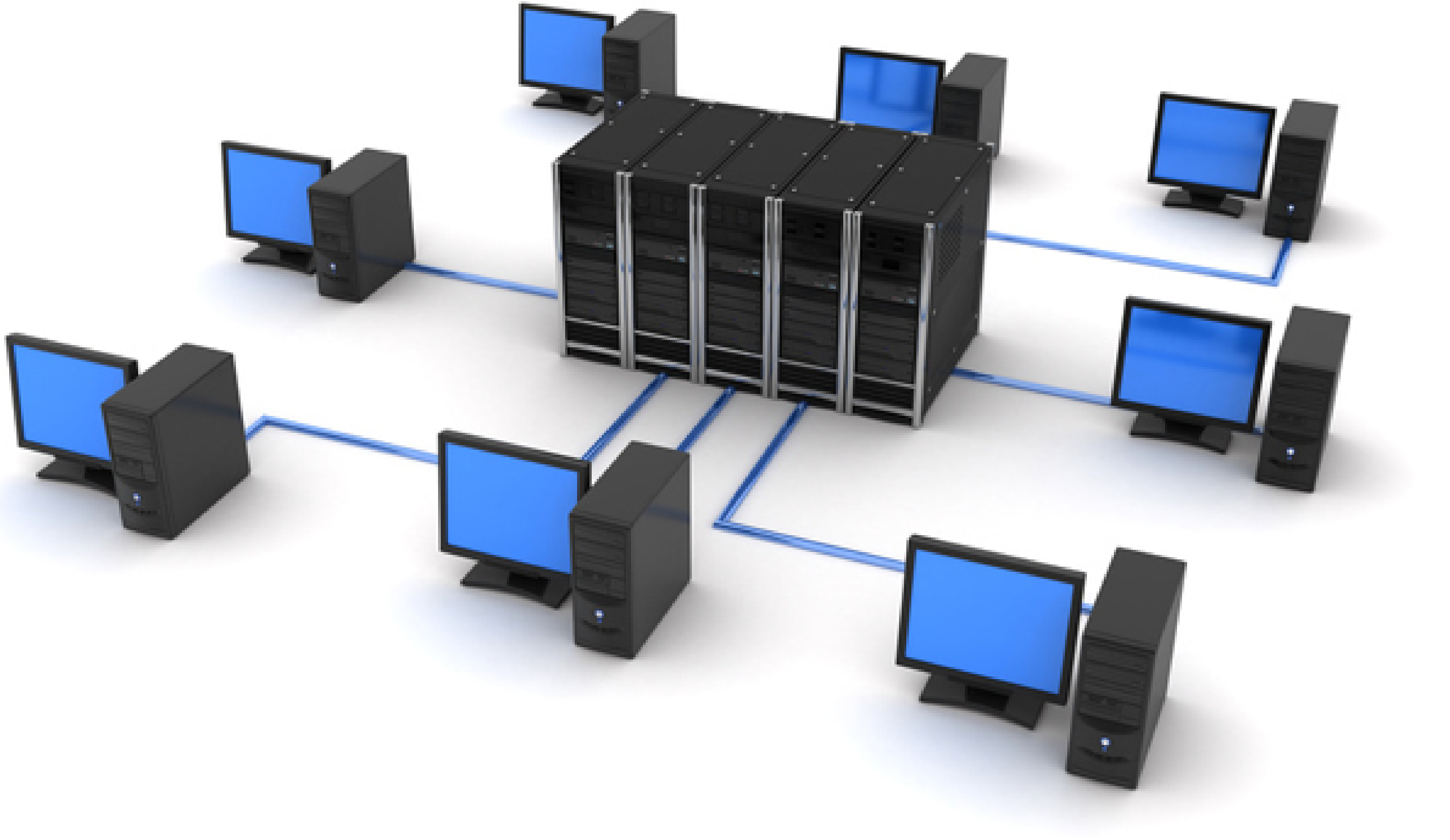
RGPV BE

Computer Science Engg. 5th Sem



Database Management System

B.E. 3rd Year (5th Semester)

CBGS Scheme

RGPV

Strictly According to the new
syllabus of RGPV

For
5th Semester
(Computer Science Engineering)

Database Management System

SYLLABUS

(Database Management System)

Unit I

DBMS Concepts and architecture Introduction, Database approach v/s Traditional file accessing approach, Advantages, of database systems, Data models, Schemas and instances, Data independence, Data Base Language and interfaces, Overall Database Structure, Functions of DBA and designer, ER data model: Entitles and attributes, Entity types, Defining the E-R diagram, Concept of Generalization, Aggregation and Specialization. transforming ER diagram into the tables. Various other data models object oriented data Model, Network data model, and Relational data model, Comparison between the three types of models.

Unit II

Relational Data models: Domains, Tuples, Attributes, Relations, Characteristics of relations, Keys, Key attributes of relation, Relational database, Schemas, Integrity constraints. Referential integrity, Intension and Extension, Relational Query languages: SQL-DDL, DML, integrity constraints, Complex queries, various joins, indexing, triggers, ssertions, Relational algebra and relational calculus, Relational algebra operations like select, Project, Join, Division, outer union. Types of relational calculus i.e. Tuple oriented and domain oriented relational calculus and its operations.

Unit III

Data Base Design: Introduction to normalization, Normal forms, Functional dependency, Decomposition, Dependency preservation and losless join, problems with null valued and dangling tuples, multivalued dependencies.Query Optimization: Introduction, steps of optimization, various algorithms to implement select, project and join operations of relational algebra, optimization methods: heuristic based, cost estimation based.

Unit IV

Transaction Processing Concepts: - Transaction System, Testing of Serilizability, Serializability of schedules, conflict & view serializable schedule, recoverability, Recovery from transaction failures. Log based recovery. Checkpoints deadlock handling. Concurrency Control Techniques: - Concurrency Control, locking Techniques for concurrency control, time stamping protocols for concurrency control, validation based protocol, multiple granularity. Multi version schemes, Recovery with concurrent transaction. Introduction to Distributed databases, data mining, data warehousing, Object Technology and DBMS, Comparative study of OODBMS Vs DBMS. Temporal, Deductive, Multimedia, Web & Mobile database.

Unit V

Study of Relational Database Management Systems through Oracle/Postgres SQL/MySQL: Architecture, physical files, memory structures, background process. Concept of table spaces, segments, extents and block. Dedicated server, multi threaded server. Distributed database, database links, and snapshot. Data dictionary, dynamic performance view.Security, role management, privilege management, profiles, invoker defined security model. SQL queries, Data extraction from single, multiple tables equi-join, non equi-join, self-join, outer join. Usage of like, any, all, exists, in Special operators. Hierarchical queries, inline queries, flashback queries. Introduction of ANSI SQL, anonymous block, nested anonymous block, branching and looping constructs in ANSI SQL. Cursor management: nested and parameterized cursors, Oracle exception handling mechanism. Stored procedures, in, out, in out type parameters, usage of parameters in procedures. User defined functions their limitations. Triggers, mutating errors, instead of triggers.

UNIT - 1

- ***What is meant by DBMS ?***

A database management system (DBMS) is a collection of interrelated data and a set of programs to access those data. The collection of data, usually referred to as the ***database,*** contains information relevant to an enterprise. The primary goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient.

Database systems are designed to manage large bodies of information. Management of data involves both defining structures for storage of information and providing mechanisms for the manipulation of information. In addition, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.

- ***Define the term data redundancy and inconsistency.***

Since different programmers create the files and application programs over a long period, the various files are likely to have different formats and the programs may be written in several programming languages. Furthermore, the same information may be duplicated in several files. This redundancy leads to higher storage and access cost. In addition, it may lead to data inconsistency, that is, the various copies of the same data may no longer agree.

- ***What are the problem caused by data redundancies ? Can data redundancy be completely eliminated when a database approach is used.***

Redundancy of data means, having same set of data many times at different locations. For example, in case of college administration system. The identity of students are stored in every department like admission department, fees department and examination conducting unit etc.

This redundancy of data leads to various problems. Firstly, different units are storing the same data thus storage spaced is wasted. Secondly, in case any manipulation has to be done, it has to be done at every place, thus leading to duplication of efforts. Third and major one, it leads to inconsistency, i.e. it may be possible that two records in the two departments reveal different facts, and we cannot judge which one is correct. Those anomalies can destroy the effectiveness of the database.

The designed goal with the database approach forces to concentrate the data into a single logical structure, that will be used by every department. And the data in design, is related with another with the help of primary fact.

For example, in student database, the primary fact is roll number, the various data like name, address, and phone along with roll number is stored by admission unit, the other unit uses the roll no., for distinguishing every student from another.

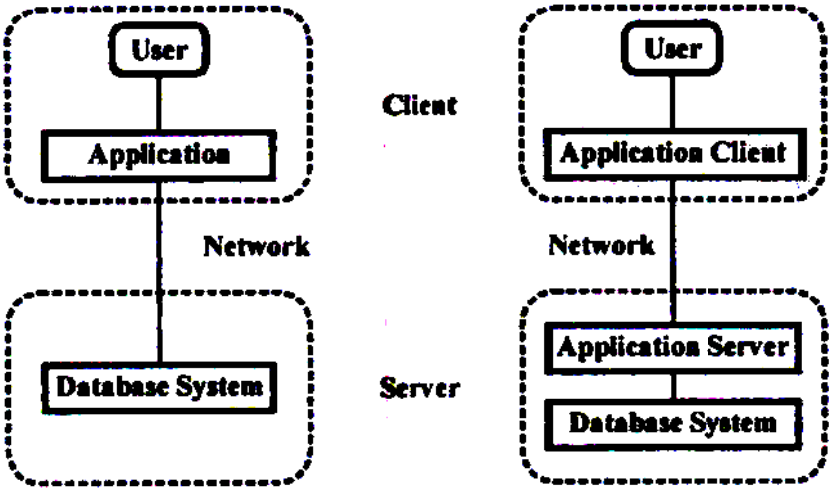
The relational database controls data redundancy by using common attributes that are shared by tables, called foreign keys. The proper use of foreign keys is crucial to controlling data redundancy. Although the use of foreign keys does not totally eliminate data redundancies, because the foreign key values can be repeated many times, the proper use of foreign keys minimizes data redundancies, thus minimizing the chance that destructive data anomalies will develop.

- ***Explain data dictionary.***

The DBMS must provide a data dictionary function. The data dictionary is a database which contains "data about the data" (called metadata or descriptor) – that is, definitions of other objects in the system, instead of "raw data". In paniicular, all of the various schemas and mappings and all of the various security and integrity constraints will be stored, in both source and object form, in the dictionary.

- ***Explain the difference between the two-tier and three-tier client/ server architecture.***

Database applications are partitioned into two or three parts as shown in fig. In a ***two-tier architecture***, the application is partitioned into a component that resides at the client machine, which invokes database system functionality at the server machine through query language statements. Application program interface standards like ODBC and JDBC are used for interaction between the client and the server.



(a) Two-tier Architecture *(b) Three-tier Architecture*

Fig. Two-tier and Three-tier Architectures

In contrast, in a three-tier architecture, the client machine acts as merely a front end and does not contain any direct database calls. Instead, the client end communicates with an application server, through a forms interface. The application server in turn communicates with a database system to access data. The ***business logic*** of the application, which says what actions to carry out under what conditions, is embedded in the application server, instead of being distributed across multiple clients. Three-tier applications are more appropriate for large applications, and for applications that run on the World Wide Web.

- ***Describe the generalized architecture of a database system.***

A database system is partitioned into module that deal with each of the responsibilities of the overall system. The functional components of a database system can be divided into the storage manager and the query processor components. Fig. shows the structure of DBMS.

In fig. the functional components of DBMS have query components and storage components as follows –

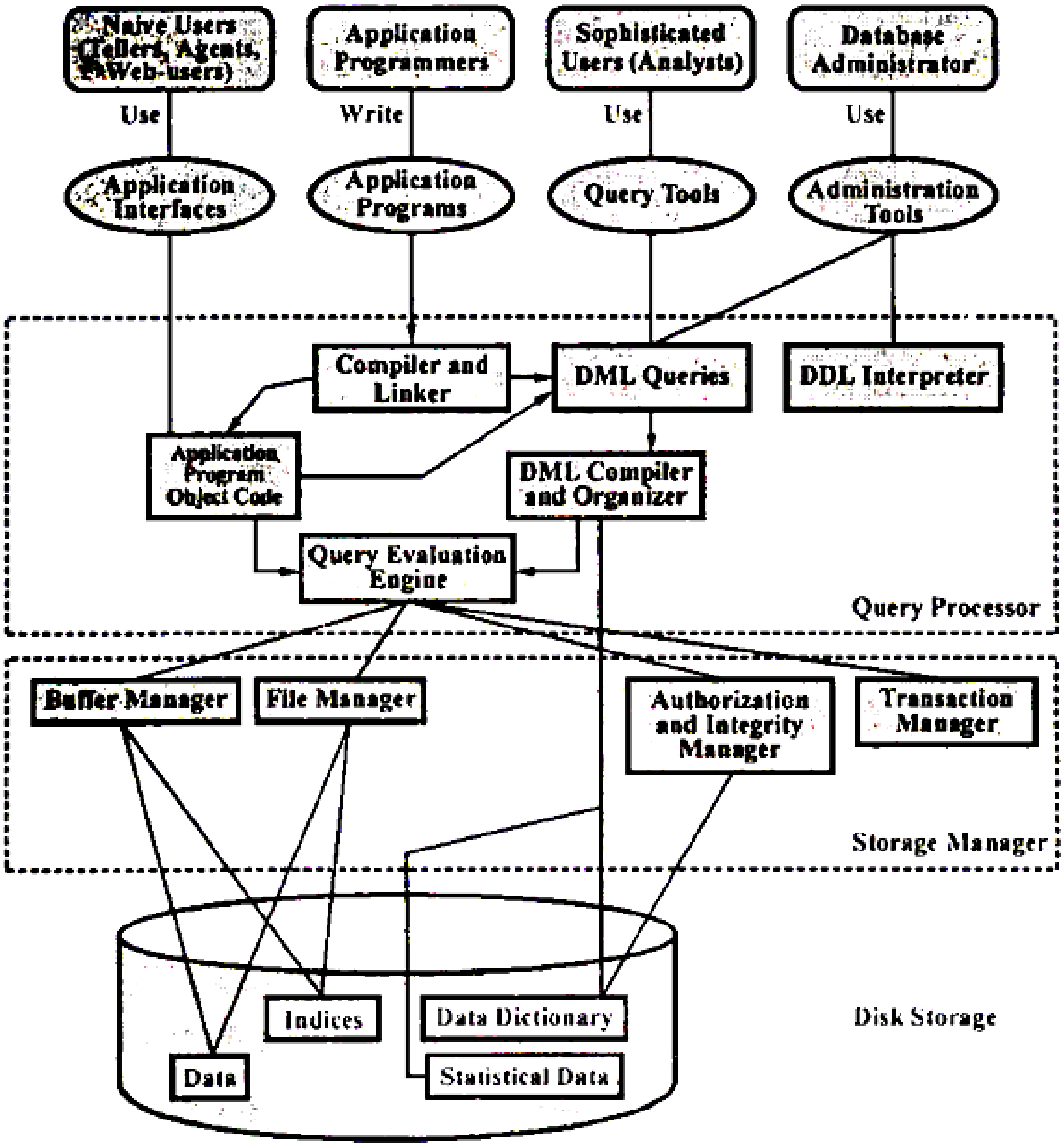


Fig. System Structure

The Query Processor – The query processor components include –

(i) DDL Interpreter – Its function is to execute the low-level statements and records them in a set of tables that having metadata.

(ii) DML Compiler – Its function is to convert DML statements in source form of query language into necessary object form (i.e., low-level instructions) that query evaluation engine understands.

(iii) DML Precompiler – Its function is to convert DML statements embedded in application program to normal procedure calls in host language. To generate appropriate code, the precompiler must interact with DML compiler.

(iv) Query Evaluation Engine – Its function is to execute the low-level instruction generated by DML compiler.

Storage Manager – A storage manager is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.

The storage manager components include –

(i) Authorization and Integrity Manager – Tests for the satisfaction at integrity constraints and checks the authority of users to access data.

(ii) Transaction Manager – Ensures that the database remains in consistent state despite system failure and concurrent transaction executions proceed without conflicting.

(iii) Buffer Manager – Responsible for data fetching from disk storage into main memory and deciding what data to cache in memory.

(iv) File Manager – Manages allocation of space on disk storage and data structures used to represent information that stored on disk.

The storage manager implements several other data structures are needed as the part of physical system for implementation –

(i) Indices – These provide fast access to the data items that hold particular values.

(ii) Statistical Data – This store statistical information about data in database. To execute a query, this information is used by query processor.

(iii) Data Files – These are actual files that store the data in database i.e., these are database files

(iv) Data Dictionary – This stores metadata about each and every entity of the database along with security and integrity constraints.

- ***Explain the difference between a file-oriented system and a database oriented system.***

There are a number of characteristics to distinguish the database approach from the traditional approach of programming with files. In traditional file processing, each user defines and implements the files needed for a specific application as part of programming the application. For example, one user, the ***grade reporting office***, may keep a filė on students and their grades. Programs to print a student transcript and to enter new grades into the file are implemented. A second user, the accounting office, may keep track of student's fees and their payments. Although both users are interested in data about students, each user has to maintain separate files and programs to manipulate these files because each user requires some data not available from the other user's files. This redundancy in defining and storing data causes wasted storage space and redundant efforts to maintain common data up-to-date. In the database approach, a single repository of data is maintained that ıs defined once and then is accessed by various users.

The main characteristics of the database approach versus the file-processing approach are as follows –

(i) Self-describing Nature of a Database System – The database system contains not only the database itself but also a complete definition or description of the database structure and constraints. This definition is stored in the system, ***catalog***, which contains information such as the structure of each file, the type and storage format of each data item and various constraints on the data. The information stored in the catalog is ***meta-data*** and it describes the structure of the primary database.

In traditional file processing, data definition is the part of the application programs themselves. Hence, these programs are constrained to work with only one specific database, whose structure is declared in the application programs. Whereas file-processing software can access only specific databases, DBMS software can access diverse databases by extracting the data definitions from the catalog and then using these definitions.

(ii) Insulation between Programs and Data, and Data Abstraction – In traditional file processing, the structure of data files is embedded in the access programs, so any changes to the structure of a file may require changing all programs that access this file. By contrast, DBMS access programs do not require such changes. The structure of data files is stored in the DBMS catalog separately from the access programs. This property is called ***program-data independence***.

The characteristic that allows program-data independence and program-operation independence is called ***data abstraction***.

(iii) Multiple Views of the Data – A database has many users, each of whom may require a different perspective or view of the database. A view may be a subset of the database or it may contain virtual data that is derived from the database files but is not explicitly stored. Some users do not need to be aware of whether the data they refer to is stored or derived. A multiuser DBMS having variety of applications must provide facilities for defining multiple views.

(iv) Sharing of Data and Multiuser Transaction Processing – A multiuser DBMS must allow multiple users to access the database at the same time. This is essential if data for multiple applications is to be integrated and maintained in a single database. The DBMS must include concurrency control software to ensure that several users trying to update the same data do so in a controlled manner so that the result of the updates is correct.

- ***What are the advantages of DBMS over traditional file system ?***

The advantages of DBMS are as follows –

(i) Reduction of Redundancies – Centralized control of data by the DBA avoids unnecessary duplication of data and reduces the total amount of data storage required. It also eliminates the extra processing necessary to trace the required data in a large mass of data. Another advantage of avoiding duplication is the elimination of the inconsistencies present in redundant data files.

(ii) Shared Data – A database allows the sharing of data under its control by any number of application programs or users.

(iii) Integrity – Centralized control ensures that adequate checks are incorporated in the DBMS to provide data integrity. Data integrity means that the data contained in the database is accurate and consistent.

(iv) Security – Data is of vital importance to an organization and may be confidential. This confidential data must not be accessed by unauthorised persons. The DBA ensures that proper access procedures are followed, including proper authentication schemes for access to the DBMS and additional checks before permitting access to sensitive data. Different levels of security can be implemented for various types of data and operations. The enforcement of security can be datavalue dependent as well as data-type dependent.

(v) Conflict Resolution – The DBA resolves the conflicting requirements of various users and applications. In essence, the DBA chooses the best file structure and access method to get optimal performance for the response-critical applications, while permitting less critical applications to continue to use the database, although with a relatively slower response.

(vi) Data Independence – There are two types of data independence – physical data independence and logical data independence. Data independence is advantageous in the database environment since it allows for changes at one level of the database without affecting other levels.

- ***Define the term data models.***

A model is an abstraction process that hides superfluous details while highlighting details pertinent to the applications at hand. Data model is a mechanism that provides the abstraction for database applications. Data modeling is used to represent entities of interest and their relationships in the database. It allows the conceptualization of the association between various entities and their attributes. We can say that data model is a collection of conceptual tools to describe data, data relationships, data semantics and consistency constraints.

A number of models for data representation have been developed. Most data representation models, provide mechanisms to structure data for the entities being modeled and allow a set of operations to be defined on them. The models also enforce a set of constraints to maintain the integrity of the data. These models differ in their method of representing the associations amongst entities and attributes.

- ***Explain the various data models briefly with an example.***

One data model can be distinguished from other on the basis of the way relationship among data, that is defined and the way the data is conceptually defined. There are many data models, chosen as per need of the application. These are fallen in following categories –

(i) Object-based logical models

(ii) Record-based logical models

(iii) Physical data models.

(i) Object-based Logical Models – These models are used in describing data at logical and view levels. They are characterized by the fact that they provide flexible structuring capabilities and allow data constraints to specify explicitly. This model emphasizes that, everything is object having a set of attributes. There are many data models in this category –

(a) Entity-relationship model (b) Object-oriented model

(c) Semantic data model (d) Functional ~~data~~ model.

(a) Entity-relationship Model – This model moves around three things – entity, attribute and relationship. This model is based on perception that consists of collection of objects called entities and every two entities are distinguished from other through their own set of properties. The relationship exists between these entities. E-R diagram graphically expresses the logical structure of database (schema) and it uses –

(1) Rectangles, to represent entity set

(2) Ellipses, to represent attributes

(3) Diamonds, to represent relationships among entity sets

(4) Lines, to show the links between entities and relationships.

(b) Object-oriented Model – This model is based on a collection of objects. Object has values stored in instance variable within the object. An object also contains bodies of code that operate on the object. These bodies of code are called ***methods***. This model introduces the concept of classes that contains objects having same type of values i.e., same set of attribute names and same methods. Values of attributes determine the object along with the methods i.e., set of instructions, that are used to modify those values. At any particular moment of time, object can be said to be the instance of the class. Two or more objects communicate each other by passing messages. The only way in which one object can access the data of another object is by invoking a method of that object. This action is called ***sending a message*** to the object. Thus, the call interface of the methods of an object defines that objects are externally visible parts. The internal parts of the object, the instance variables and methods code are not visible externally. The result is two levels of data abstraction.

(ii) Record-based Logical Models – These models are also used in describing data at logical and view levels. But in contrast to the object-based data models, they revolve around the records of the database and specify the overall structure of database, with the help of values of records. Record-based models are so named, since the database is structured in fixed-format records of several types.

Each record type defines a fixed number of fields or attributes and each field is usually of a fixed length. This simplifies the physical-level implementation of the database. The three most widely used record-based models are –

(a) Relational model (b) Network model (c) Hierarchical model.

(a) Relational Model – This is most popular among the various record-based models. This model uses a collection of tables to represent both data and the relationships among those data. Each table has multiple columns and each column has unique name. Table is given the name relations, rows represent the records and the columns represent the attributes or properties.

In this model, data is handled on a conceptual rather than physical basis. This mechanism helps in processing entire files of data with single statement. The logical manipulation of data also makes feasible the creation of query languages more accessible to non-technical users.

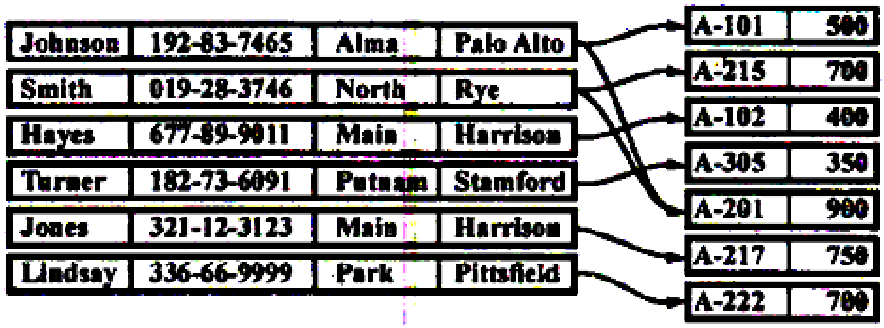
For example, fig. represents a sample of relational database comprising of two tables. One table shows bank customers and other shows the account that belong to those customers. It shows, for example, that customer Johnson with *social_security* number 192-83-7465, lives on Alma in Palo Alto and has two accounts, A-101 with a *balance* of $500 and A-201 with a *balance* of $900. Also, customers Johnson and Smith share account number A-201.

customer_name	social_security	customer_street	customer_city	account_number
Johnson	192-83-7465	Alma	Palo Alto	A-101
Smith	019-28-3746	North	Rye	A-215
Hayes	677-89-9011	Main	Harrison	A-102
Turner	182-73-6091	Putnam	Stamford	A-305
Johnson	192-83-7465	Alma	Palo Alto	A-201
Jones	321-12-3123	Main	Harrison	A-217
Lindsay	336-66-9999	Park	Pittsfield	A-222
Smith	019-28-3746	North	Rye	A-201

account_number	balance
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

Fig. Relational Database Sample

(b) Network Model – Data in network model are represented by collection of records and relationships among data are represented by links, that can be viewed as pointers. A pointer is a physical address which identifies where next record can be found on the disk. Fig. shows a network database using the same information as shown in fig.



Johnson	192-83-7465	Alma	Palo Alto
Smith	019-28-3746	North	Rye
Hayes	677-89-9011	Main	Harrison
Turner	182-73-6091	Putnam	Stamford
Jones	321-12-3123	Main	Harrison
Lindsay	336-66-9999	Park	Pittsfield

A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

Fig. Network Database Sample

(c) Hierarchical Model – It is very similar to network model, as in both of the models, data and relationships among data are represented by records and links respectively. In this model, records are organized as collection of trees rather than arbitrary graphs. Fig. shows hierarchical database system.

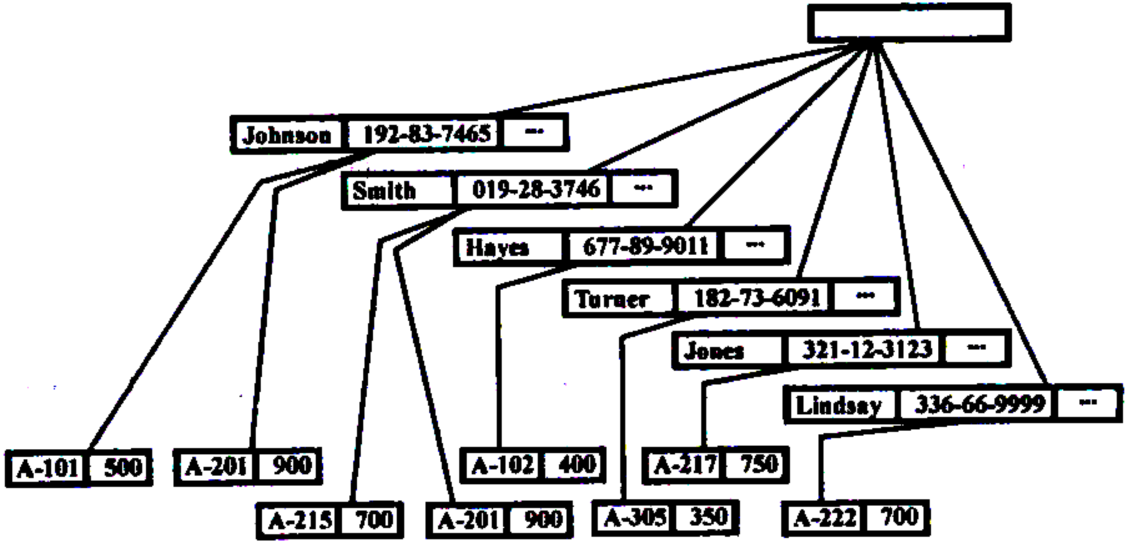


Fig. Hierarchical Database Sample

(iii) Physical Data Models – This model is used to describe data at the lowest level i.e., to describe the behaviour of data at the disk level i.e., the way the data and data relationships are maintained while storing them on the disk. This decided the way the DBMS is going to use secondary storage devices, for storing and accessing database. The widely used these data models are –

- ***What is schema and subschema ? Explain these two concepts through examples.***

Schema is the logical description of entire database. That is, the overall design of the database is called ***database schema*** and it is changed rarely. The arrangement of all the files for an entire organization is an example of the schema.

Database systems have several schemas, partitioned according to the levels of abstraction. At lowest level is physical schema, at intermediate level is logical schema and at highest level is subschema or external schema. Generally, database systems support one physical schema, one logical schema and several subschemas.