SESSION 8 : DATABASES MODELS

INTRODUCTION
Database management systems are usually categorized according to the database model that they support, such as the network, relational or object model.

OBJECTIVES
• Identify the various components of a DBMS.
• Describe the different database model
• Outline the basic features of each model
DBMS Components

Source Schemas and mappings → DDL Processors

Planned DML requests → DML Processors

Unplanned (adhoc)DML → Query Language Processors

Compiled requests → Optimizer

Optimized requests → Run Time Manager

Database

Data

Metadata (Data dictionary)

Metadata

Source and object schemas and mapping

-Source : An Into to database system by C.J. Date
Now we will discuss about the how DBMS functions and its various components.

- **Data Definition**, the DBMS must be able to accept data definitions in source form and convert them to the appropriate object form. In other words it must include DDL processor component for each of the various data definition languages.

- **Data Manipulation** helps user to add, change, and delete information in a database and query it for valuable information. Software tools within the data manipulation subsystem are most often the primary interface between user and the information contained in a database. DML processor allows user to specify its logical information requirements. Typically DML requests can be “Planned” or “unplanned”.
  - Planned requests is one for which the need was foreseen well in advance of the time at which the request is executed.
  - Unplanned request, on the other hand, is an *ad hoc* query, i.e, a request for which the need was not seen in advance.

- **Optimization and execution**, DML requests both planned and unplanned, are processed by optimizer, who determines an effective way to implement requests. The optimized requests are executed under the control of the run-time manager. If there are frequently occurring usage patterns or requests, some DBMS can adjust themselves to improve the speed of those interactions. In some cases the DBMS will merely provide tools to monitor performance, allowing a human expert to make the necessary adjustments after reviewing the statistics collected.

- **Data Security and Integrity**, Often it is desirable to limit who can see or change which attributes or groups of attributes of database. DBMS must monitor user requests to tackle unauthorized requests. It also ensures the data integrity.

- **Data Recovery and Concurrency**, talks about the Transaction manager.

- **Data Dictionary**, contains data about data often called as Metadata-that is, definitions of other objects in the system, instead of just “raw data”. In particular,
all of the various schemas and mapping and all of the security and integrity constraints will be stored, in both source and object form, in data dictionary.

**DATA MODELS**

According to the DBMS database model. The four most common types of models are the:

- hierarchical model,
- network model,
- relational model, and
- object model.

**Hierarchical Model**

The hierarchical data model organizes data in a tree structure. There is a hierarchy of parent and child data segments. This structure implies that a record can have repeating information, generally in the child data segments. Data in a series of records, which have a set of field values attached to it. It collects all the instances of a specific record together as a record type. These record types are the equivalent of tables in the relational model, and with the individual records being the equivalent of rows. To create links between these record types, the hierarchical model uses Parent Child Relationships. These are a 1:N mapping between record types. This is done by using trees, like set theory used in the relational model, "borrowed" from maths. For example, an organization might store information about an employee, such as name, employee number, department, salary. The organization might also store information about an employee's children, such as name and date of birth. The employee and children data forms a hierarchy, where the employee data represents the parent segment and the children data represents the child segment. If an employee has three children, then there would be three child segments associated with one employee.
In a hierarchical database the parent-child relationship is one to many. This restricts a child segment to having only one parent segment.

Another example we could have of a tree representing a university department, with subtrees representing staff members, students, facilities and courses. We could have several of these department trees, however we could not specify one employee working for more than one department. To do this we would have to create two instances of the employee. As you can imagine, this can cause concurrency inaccuracies when updating information e.g. the information may not be exactly the same for all records, leading to confusion.

Hierarchical DBMSs were popular from the late 1960s, with the introduction of IBM's Information Management System (IMS) DBMS, through the 1970s. The hierarchical model is no longer used as the basis for current commercially produced systems, however, there are a large number of legacy installations. These legacy systems are likely to be phased out over time, as the number of qualified staff declines due to retirement and retraining.

**Network Model**

The popularity of the network data model coincided with the popularity of the hierarchical data model. Some data were more naturally modeled with more than one parent per child. So, the network model permitted the modeling of many-to-many relationships in data. In 1971, the Conference on Data Systems Languages (CODASYL) formally defined the network model. The basic data modeling construct in the network model is the set construct. A set consists of an owner record type, a set name, and a member record type. A member record type can have that role in more than one set, hence the multiparent concept is supported. An owner record type can also be a member or owner in another set. The data model is a simple network, and link and intersection record types (called junction records by IDMS) may exist, as well as sets between them. Thus, the complete
network of relationships is represented by several pairwise sets; in each set some (one) record type is owner (at the tail of the network arrow) and one or more record types are members (at the head of the relationship arrow). Usually, a set defines a 1:M relationship, although 1:1 is permitted. The CODASYL network model is based on mathematical set theory.

The network model is not commonly used today to design database systems, however, there are a few instances of it being used by companies as part of a legacy system. Because it represents data in a format closer to the way in which it stores it than the other models, it can be faster than relational systems.

**Relational Model**

(RDBMS - relational database management system) A database based on the relational model developed by E.F. Codd. A relational database allows the definition of data structures, storage and retrieval operations and integrity constraints. In such a database the data and relations between them are organised in tables. A table is a collection of records and each record in a table contains the same fields.

Properties of Relational Tables:

- Values Are Atomic
- Each Row is Unique
- Column Values Are of the Same Kind
- The Sequence of Columns is Insignificant
- The Sequence of Rows is Insignificant
- Each Column Has a Unique Name

Certain fields may be designated as keys, which means that searches for specific values of that field will use indexing to speed them up. Where fields in two different tables take values from the same set, a join operation can be performed to select related records in the two tables by matching values in those fields.
Often, but not always, the fields will have the same name in both tables. For example, an "orders" table might contain (customer-ID, product-code) pairs and a "products" table might contain (product-code, price) pairs so to calculate a given customer's bill you would sum the prices of all products ordered by that customer by joining on the product-code fields of the two tables. This can be extended to joining multiple tables on multiple fields. Because these relationships are only specified at retrieval time, relational databases are classed as dynamic database management system.

**Advantages of the relational model over the hierarchical and network models.**

1. The relational model is more flexible than other models. The way the data values exist in the relational tables does not in any way restrict the kinds of processing that can be done. In the hierarchical and network models, manipulation of the data is restricted by the structure built into the data model.
2. The relational model has a sound theoretical base in mathematical theory. You can use the mathematics of relations as the basis for data processing procedures.
3. The organization of the relational model is simple to understand and, therefore, a good vehicle to communicate database ideas.
4. The same database can generally be represented with less redundancy using the relational model than the other two models.

**Object Model**

Object DBMSs add database functionality to object programming languages. They bring much more than persistent storage of programming language objects. Object DBMSs extend the semantics of the C++, Smalltalk and Java object
programming languages to provide full-featured database programming capability, while retaining native language compatibility. A major benefit of this approach is the unification of the application and database development into a seamless data model and language environment. As a result, applications require less code, use more natural data modeling, and code bases are easier to maintain. Object developers can write complete database applications with a modest amount of additional effort.

According to Rao (1994), "The object-oriented database (OODB) paradigm is the combination of object-oriented programming language (OOPL) systems and persistent systems. The power of the OODB comes from the seamless treatment of both persistent data, as found in databases, and transient data, as found in executing programs."

In contrast to a relational DBMS where a complex data structure must be flattened out to fit into tables or joined together from those tables to form the in-memory structure, object DBMSs have no performance overhead to store or retrieve a web or hierarchy of interrelated objects. This one-to-one mapping of object programming language objects to database objects has two benefits over other storage approaches: it provides higher performance management of objects, and it enables better management of the complex interrelationships between objects. This makes object DBMSs better suited to support applications such as financial portfolio risk analysis systems, telecommunications service applications, worldwide web document structures, design and manufacturing systems, and hospital patient record systems, which have complex relationships between data.
SUMMARY:

The hierarchical and network database models preceded the relational model; today very few commercial databases use either of these models in hierarchical and network database models. Data is represented as a series of parent/child relationships which is fundamentally different from relational model where data resides in a collection of tables without any hierarchy and that are physically independent of each other. The hard-coding of links makes the hierarchical and network model very inflexible; the relational database provides flexibility because the logical design is independent of the physical design. The object-oriented database, also referred to as the ‘post-relational’ database model, addresses the limitations of the relational model.

SELF TEST:

Q1. List the various components of DBMS.

Q2. Differentiate between hierarchical and network database models.

Q3. Why the relational databases are more flexible than the hierarchical and network database models.

Q4. Discuss the advantages of object oriented databases.

Q5. Surf the net and find out the names of popular RDBMS and OODBMS available in the market today.