INTRODUCTION TO ADVANCED DATABASE SYSTEMS

Topic 1 Notes

1.1 Introduction

A database is a model of some aspect of the reality of an organization.

It is conventional to call this reality a universe of discourse (UoD), or sometimes a domain of discourse. A UoD will be made up of classes and relationships between classes. The classes in a UoD will be defined in terms of their properties or attributes.

A database can be considered as an organized collection of data which is meant to represent some UoD.

Data are facts. A datum, a unit of data, is one symbol or a collection of symbols that is used to represent something.

Facts by themselves are meaningless. To prove useful they must be interpreted.

Information is interpreted data. Information is data placed within a meaningful context.

Information is data with an assigned semantics or meaning.

A database can be considered as a collection of facts or positive assertions about a UoD, such as relational database design is a module and John Davies takes relational database design.

Usually negative facts, such as what modules are not taken by a student, are not stored. Hence, databases constitute ‘closed worlds’ in which only what is explicitly represented is regarded as being true.

A database is said to be in a given state at a given time. A state denotes the entire collection of facts that are true at a given instant in time. A database system can therefore be considered as a factbase which changes through time.

Data in a database is described as being persistent. By persistent we mean that the data is held for some duration. The duration may not actually be very long.

The term persistence is used to distinguish more permanent data from data which is more transient in nature.

Hence, product data, account data, patient data and student data would all normally be regarded as examples of persistent data. In contrast, data input at a personal computer, held for manipulation within a program, or printed out on a report, would not be regarded as persistent, as once it has been used it is no longer required.

A database is made up of two parts: an intentional part and an extensional part. The intension of a database is a set of definitions which describe the structure or organization of a given database. The extension of a database is the total set of data in the database. The intension of a database is also referred to as its schema. The activity of developing a schema for a database system is referred to as database design.

1.2 Properties of a Database

Data sharing

Data stored in a database is not usually held solely for the use of one person. A database is normally expected to be accessible by more than one person, perhaps at the same time. Hence a students' database might be accessible by members of not only academic but also administrative staff.

Data integration

One major responsibility of database usage is to ensure that the data is integrated. This implies that a database should be a collection of data, which, at least ideally, has no redundant data.

In the past, for instance separate files of student information might have been maintained by different academic and administrative departments of a university with many Fields in common. The aim of a database system would be to store one logical item of data in one place only. Hence, one student record would be accessible to a range of information systems.

Data integrity
Another responsibility arising as a consequence of shared data is that a database should display integrity. In other words, that the database should accurately reflect the universe of discourse that it is attempting to model.

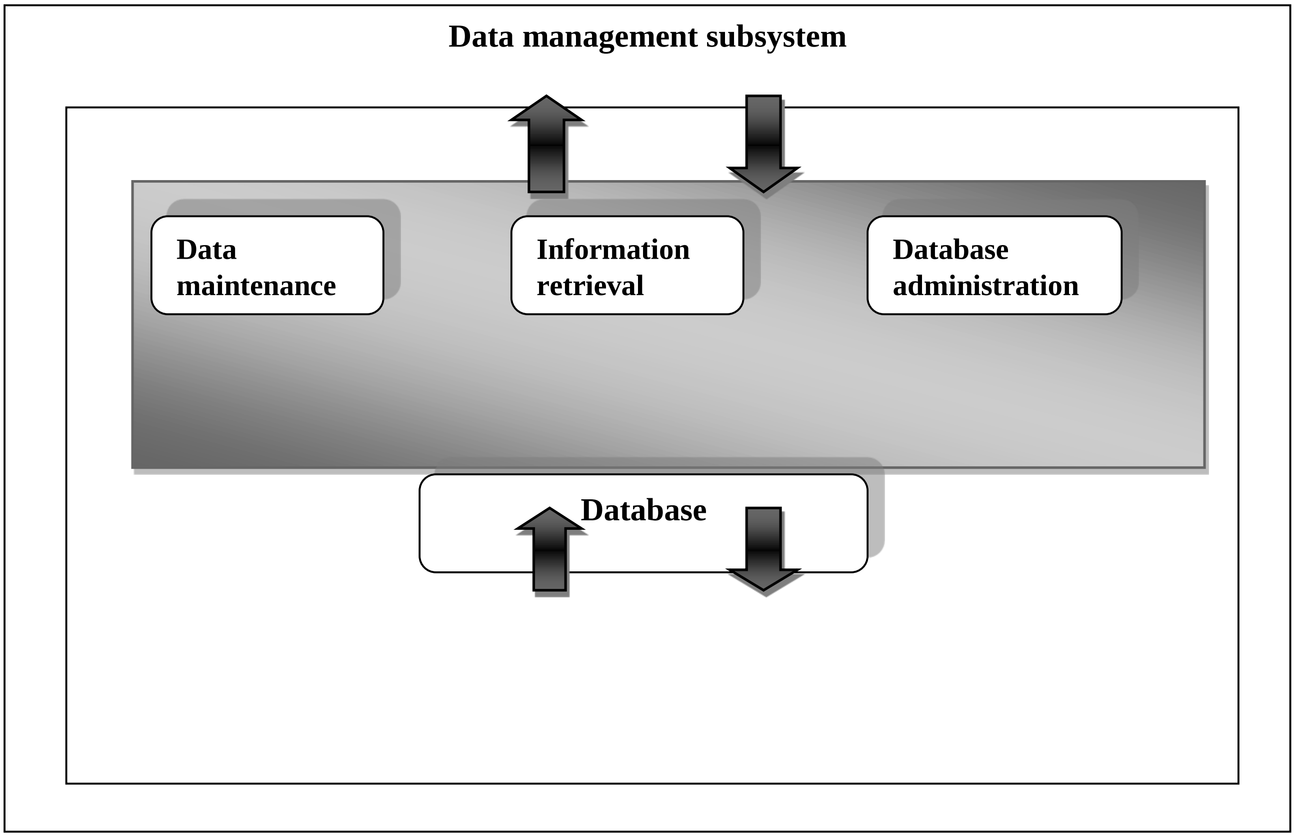
Data security
One of the major ways of ensuring the integrity of a database is by restricting access. In other words, securing the database. The main way that this is done in contemporary database systems is by defining in some detail a set of authorized users of the whole, or more usually parts of the database. For instance, a secure system would be one where the finance department has access to information used for the collection of student fees but is prohibited from changing the fee levels of given students.

Data abstraction
A database can be viewed as a model of reality. The information stored in a database is usually an attempt to represent the properties of some objects in the real world.

Data independence
One immediate consequence of abstraction is the idea of buffering data from the processes that use such data. The idea is to achieve a situation where data organization is transparent to the users or application programs which feed off data. If for instance, a change is made to some part of the underlying database no application programs using affected data should need to be changed. Also, if a change is made to some part of an application system then this should not affect the structure of the underlying data used by the application.

1.3 Database Management System

A database management system (DBMS) is an organized set of facilities for accessing and maintaining one or more databases. A DBMS is a shell which surrounds a database or series of databases and through which all interactions take place with the database.
The interactions catered for by most existing DBMS fall into three main groups
1. Data maintenance - Adding new data structures to the database, removing data structures from the database, modifying the structure of existing data, inserting new data into existing data structures, updating data in existing data structures, deleting data from existing data structures.
2. Data Retrieval - Querying existing data by end -users and extracting data for use by application programs.
3. Data control -data in the database and monitoring the performance of databases. Data Maintenance Information Retrieval Database Administration Database Data Management Subsystem



1.3.1 DBMS Components

Data: Data stored in a database include numerical data which may be integers (whole numbers only) or floating point numbers (decimal), and non-numerical data such as characters (alphabetic and numeric characters), date or logical (true or false). More advanced systems may include more complicated data entities such as pictures and images as data types.

Standard operations: Standard operations are provided by most DBMS. These operations provide the user basic capabilities for data manipulation. Examples of these standard operations are sorting, deleting and selecting records.

Data definition language (DDL): DDL is the language used to describe the contents of the database. It is used to describe, for example, attribute names (field names), data types, location in the database, etc.

Data manipulation and query language: Normally a query language is supported by a DBMS to form commands for input, edit, analysis, output, reformatting, etc. Some degree of standardization has been achieved with SQL (Structured Query Language).

Programming tools: Besides commands and queries, the database should be accessible directly from application programs through function calls (subroutine calls) in conventional programming languages.

File structures: Every DBMS has its own internal structures used to organize the data although some common data models used by most DBMS.

Data Manager:The Data Manager is the central software component of the DBMS also knows as Database Control System.

The Main Functions of Data Manager Are: –

i. Convert operations in user's Queries coming from the application programs or combination of DML Compiler and Query optimizer which is known as Query Processor from user's logical view to physical file system.
ii. Controls DBMS information access that is stored on disk.
iii. It also controls handling buffers in main memory.
iv. It also enforces constraints to maintain consistency and integrity of the data.
v. It also synchronizes the simultaneous operations performed by the concurrent users.
vi. It also controls the backup and recovery operations.

Data Dictionary: Data Dictionary is a repository of description of data in the database. It contains information about

i. Data - names of the tables, names of attributes of each table, length of attributes, and number of rows in each table.
ii. Relationships between database transactions and data items referenced by them which is useful in determining which transactions are affected when certain data definitions are changed.
iii. Constraints on data i.e. range of values permitted.
iv. Detailed information on physical database design such as storage structure, access paths, files and record sizes.
v. Access Authorization - is the Description of database users their responsibilities and their access rights.
vi. Usage statistics such as frequency of query and transactions.
vii. Data dictionary is used to actually control the data integrity, database operation and accuracy. It may be used as a important part of the DBMS.

Importance of Data Dictionary

Data Dictionary is necessary in the databases due to following reasons:

i. It improves the control of DBA over the information system and user's understanding of use of the system.
ii. It helps in documenting the database design process by storing documentation of the result of every design phase and design decisions.
iii. It helps in searching the views on the database definitions of those views.
iv. It provides great assistance in producing a report of which data elements (i.e. data values) are used in all the programs.
v. It promotes data independence i.e. by addition or modifications of structures in the database application program are not affected.

End Users

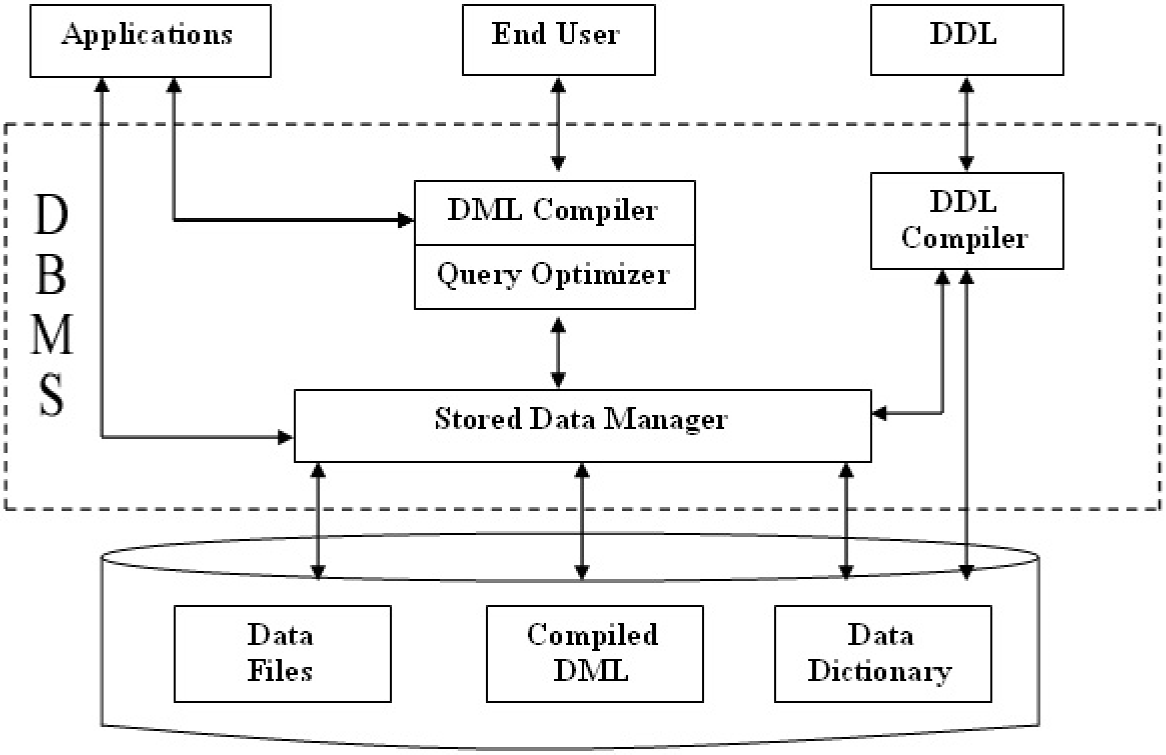


Fig 1.2: DBMS components

1.4 Database Concepts

Integrity

When we say that a database displays integrity we mean that it is an accurate reflection of its UOD. The process of ensuring integrity is a major feature of modern information systems. The process of designing for integrity is a much-neglected aspect of database development.

Integrity is an important issue because most databases are designed, once in use, to change. In other words, the data in a database will change over a period of time. If a database does not change, i.e. it is only used for reference purposes, then integrity is not an issue of concern.

It is useful to think of database change as occurring in discrete rather than continuous time. In this sense, we may conceive of a database as undergoing a number of state changes, caused by external and internal events. Of the set of possible future states feasible for a database some of these states are valid and some are invalid. Each valid state forms the extension of the database at that moment in time. Integrity is the process of ensuring that a database travels through a space defined by valid states.

Integrity involves determining whether a transition to the state below is a valid one. That is, integrity involves answering questions such as: is it valid to add another data recode university database, which relates Miller L.H to Computer Science?

Replication

When we design the database should design to minimize replication of data. In a database we attempt to store only one item of data about one object or relationships between objects in our UOD. Ideally, a database should be a repository with no replicated facts.

Transaction

The events that cause a change of state of the database is characterized in database terms as transaction. A transaction changes a database from one state to another.

A transaction type that might be relevant to the university database system might be:

Enroll Student in Module

Integrity Constraints

Database integrity is ensured through integrity constraints. An integrity constraint is a rule, which establishes how a database is to remain an accurate reflection of its UOD.

Constraints may be divided into two major types: static constraints and transition constraints.

A static constraint or “state invariant” is used to check that an incoming transaction will not change a database into an invalid state. A static constraint is a restriction defined on states of the database. An example of a static constraint relevant to our University database might be: students can only take currently offered modules only. This static constraint would prevent us from entering the following fact into our current database:

Miller L.H takes Deductive Database Systems

Since Deductive Database Systems is not a currently offered module by the university.

Database Functions

Most data held in a database is there to fulfill some organizational need. To perform useful activity with a database we need two types of function: Update and Query functions. Update functions cause changes to data. Query functions will extract data from the database.

Update Functions

A transaction is an update function. It changes a database from one state to another.

Update functions that might be relevant to the university database might be:

Initiate Semester

Offer Module

Cancel Module

Enroll student on course

Enroll student in module

Transfer Student between modules

Query Functions

This does not modify database in any way, but is used primarily to check whether a fact or group of facts holds in a given database. Query functions can use to retrieve the data from the database. Is course X being offered?

Is Student Y takes Course X?

Above Query functions will relevant to our OUD.

Multi-User Access

By definition some way must exist in a multiuser database of handling situations where a number of persons or application systems want to access the same data at effectively the same time.

Consider a situation where one user is enrolling student Anne John on module Relational Database Systems. At the same time another user is removing the module Relational Database Systems from the current offering. Clearly, in the time it takes the first user to enter an enrolment fact, the second user could have denied the module fact.

The database is left in an inconsistent state.

In any multi user database system some system must therefore be provided to resolve such conflicts of concurrency.

Database Views

Part of the reason that data in databases is shared is that a database may be used for different purposes within one organization. For instance, the academic database described in this chapter might be used for various purposes in a university such as recording student grades or timetabling classes. Each distinct user group may demand a particular subset of the database in terms of the data it needs to perform its work. Hence administrators in academic departments will be interested in items such as student names and grades while a timetabler will be interested in rooms and times. This subset of data is known as a view.

In practice a view is merely a query function that is packaged for use by a particular user group or program.

1.5 Features and Capabilities of DBMS

Features commonly offered by database management systems include:

Query Ability

Querying is the process of requesting attribute information from various perspectives and combinations of factors. Example: "How many 2-door cars in Texas are green?"

A database query language and report writer allow users to interactively interrogate the database, analyze its data and update it according to the user’s privileges on data. It also controls the security of the database.

If the DBMS provides a way to interactively enter and update the database, as well as interrogate it, this capability allows for managing personal databases. However, it may not leave an audit trail of actions or provide the kinds of controls necessary in a multi-user organization. These controls are only available when a set of application programs are customized for each data entry and updating function.

Backup and Replication

Copies of attributes need to be made regularly in case primary disks or other equipment fails. A periodic copy of attributes may also be created for a distant organization that cannot readily access the original. DBMS usually provide utilities to facilitate the process of extracting and disseminating attribute sets.

When data is replicated between database servers, so that the information remains consistent throughout the database system and users cannot tell or even know which server in the DBMS they are using, the system is said to exhibit replication transparency.

Rule Enforcement

Often one wants to apply rules to attributes so that the attributes are clean and reliable. For example, we may have a rule that says each car can have only one engine associated with it (identified by Engine Number). If somebody tries to associate a second engine with a given car, we want the DBMS to deny such a request and display an error message. However, with changes in the model specification such as, in this example, hybrid gas-electric cars, rules may need to change. Ideally such rules should be able to be added and removed as needed without significant data layout redesign.

Security
Often it is desirable to limit who can see or change a given attributes or groups of attributes. This may be managed directly by individual, or by the assignment of individuals and privileges to groups, or (in the most elaborate models) through the assignment of individuals and groups to roles which are then granted entitlements.

Computation
There are common computations requested on attributes such as counting, summing, averaging, sorting, grouping, cross-referencing, etc. Rather than have each computer application implement these from scratch, they can rely on the DBMS to supply such calculations. All arithmetical work to perform by computer is called a computation.

Change and Access Logging
Often one wants to know who accessed what attributes, what was changed, and when it was changed. Logging services allow this by keeping a record of access occurrences and changes.

Automated Optimization
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.

1.6 Uses of Database Management Systems

The four major uses of database management systems are:

Database Development
Database packages like Microsoft Access, Lotus Approach allow end users to develop the database they need. However, large organizations with client/server or mainframe-based system usually place control of enterprise-wide database development in the hands ent in the hands of database administrators and other database specialists. This improves the integrity and security of organizational database. Database developers use the data definition languages (DDL) in database management systems like oracle 9i or IBM's BD2 to develop and specify the data contents, relationships and structure each databases, and to modify these database specifications called a data dictionary.

Database interrogation
This is a major use of the database management system where end users can interrogate a

database management system by asking for information from a database using a *query language* or a *report generator*. They can receive an immediate response in the form of video displays or printed reports. No difficult programming ideas are required.

Database Maintenance

The databases of organizations need to be updated continually to reflect new business transactions and other events. Other miscellaneous changes must also be made to ensure accuracy of the data in the database. This database maintenance process is accomplished by transaction processing programs and other end-user application packages within the support of the database management system. Endusers and information specialists can also employ various utilities provided by a DBMS for database maintenance.

Application Development

Database management system packages play major roles in application development. End-users, systems analysts and other application developers can use the fourth generational languages (4GL) programming languages and built-in software development tools provided by many DBMS packages to develop custom application programs. For example you can use a DBMS to easily develop the data entry screens, forms, reports, or web pages by a business application. A database management system also makes the job of application programmers easier, since they do not have to develop detailed data handling procedures using a conventional programming language every time they write a program.

1.7 Advantages and Disadvantages of a DBMS

1.7.1 Advantages

Using a DBMS to manage data has many advantages:

Data independence: Application programs should be as independent as possible from details of data representation and storage. The DBMS can provide an abstract view of the data to insulate application code from such details.

Efficient data access: A DBMS utilizes a variety of sophisticated techniques to store and retrieve data efficiently. This feature is especially important if the data is stored on external storage devices.

Data integrity and security: If data is always accessed through the DBMS, the DBMS can enforce integrity constraints on the data. For example, before inserting salary information for an employee, the DBMS can check that the department budget is not exceeded. Also, the DBMS can enforce *access controls* that govern what data is visible to different classes of users.

Data administration: When several users share the data, centralizing the administration of data can offer significant improvements. Experienced professionals who understand the nature of the data being managed, and how different groups of users use it, can be responsible for organizing the data representation to minimize redundancy and for fine-tuning the storage of the data to make retrieval efficient.

Concurrent access and crash recovery: A DBMS schedules concurrent accesses to the data in such a manner that users can think of the data as being accessed by only one user at a time. Further, the DBMS protects users from the effects of system failures.

Reduced application development time: Clearly, the DBMS supports many important functions that are common to many applications accessing data stored in the DBMS. This, in conjunction with the high-level interface to the data, facilitates quick development of applications. Such applications are also likely to be more robust than applications developed from scratch because many important tasks are handled by the DBMS instead of being implemented by the application.

1.7.2Disadvantages

A database system generally provides on-line access to the database for many users. In contrast, a conventional system is often designed to meet a specific need and therefore generally provides access to only a small number of users. Because of the larger number of users accessing the data when a database is used, the enterprise may involve additional risks as compared to a conventional data processing system in the following areas.

Confidentiality, Privacy and Security When information is centralized and is made available to users from remote locations, the possibilities of abuse are often more than in a conventional system. To reduce the chances of unauthorized users accessing sensitive information, it is necessary to take technical, administrative and, possibly, legal measures. Most databases store valuable information that must be protected against deliberate trespass and destruction.

Data Quality Since the database is accessible to users remotely, adequate controls are needed to control users updating data and to control data quality. With increased number of users accessing data directly, there are enormous opportunities for users to damage the data. Unless there are suitable controls, the data quality may be compromised.

Data Integrity Since a large number of users could be using a database concurrently, technical safeguards are necessary to ensure that the data remain correct during operation. The main threat to data integrity comes from several different users attempting to update the same data at the same time. The database therefore needs to be protected against inadvertent changes by the users.

Enterprise Vulnerability Centralizing all data of an enterprise in one database may mean that the database becomes an indispensable resource. The survival of the enterprise may depend on reliable information being available from its database. The enterprise therefore becomes vulnerable to the destruction of the database or to unauthorized modification of the database.

The Cost of using a DBMS. Conventional data processing systems are typically designed to run a number of well-defined, preplanned processes. Such systems are often "tuned" to run efficiently for the processes that they were designed for. Although the conventional systems are usually fairly inflexible in that new applications may be difficult to implement and/or expensive to run, they are usually very efficient for the applications they are designed for.

The database approach on the other hand provides a flexible alternative where new applications can be developed relatively inexpensively. The flexible approach is not without its costs and one of these costs is the additional cost of running applications that the conventional system was designed for. Using standardized software is almost always less machine efficient than specialized software.

Revision questions

1. What is a database?
2. What properties should a database have to ensure effectiveness?

3. What is the difference between a database and a database management system?
4. What are the features of a database management system?

What advantages does an organization stand to gain in implementing a DBMS?