<u>Database Management System</u>

Praveen Kumar, *Dept. of Geography*

Reasons for using a DBMS

There are various reasons for using a DBMS for data storage and processing.

1. **A DBMS supports the storage and manipulation of very large data sets**

Some data sets are so big that storing them in text files or spreadsheet files becomes too awkward for use in practice. The result may be that finding simple facts takes minutes, and performing simple calculations perhaps even hours. A DBMS is specifically designed for doing this efficiently.

2. **A DBMS supports the concurrent use of the same data set by many users**

Many people are usually involved in the data collection, maintenance and processing. These data sets are often considered to be of a high strategic value to the owner(s). Moreover, for different users of the database, different views on the data can be defined. In this way, users will be under the impression that they operate on their personal database, and not on one shared by many people. They may all be using the database at the same time, without affecting each other's activities. This DBMS function is called concurrency control.

3. **A DBMS supports the use of a data model**

A data model is a language with which one can define a database structure and manipulate the data stored in it. The most prominent data model is the relational data model. Its primitives are tuples 1 A DBMS provides a high-level, declarative query language¬ (also known as records, or rows) with attribute values, and relations, being sets of similarly formed tuples.

In a database, a query is a computer program that extracts from the database that data that meets some specified conditions. A query language can be used to define queries and updates.

4. **A DBMS can be instructed to guard over data correctness**

We can ensure that the data that is entered into the database does not contain errors. This is generally known as integrity constraints. More complex integrity constraints are certainly possible, and their definition is part of the design of a database.

5. **A DBMS includes data backup and recovery functions and the control of data redundancy**

Regular back-ups of the dataset and automatic recovery schemes provide an insurance against loss of data. Storing a fact multiple times, a phenomenon known as data redundancy, can lead to situations in which stored facts may contradict each other, causing reduced usefulness of the data. A well-designed database takes care to store single facts only once. Redundancy, however, is not necessarily always problematic, as long as we specify where it occurs so that it can be controlled.

Alternatives for data management

The decision to use a DBMS will depend, among other things, on how much data there is, what type of use will be made of it, and how many users might be involved. Let's look at an example. Amina is a statistician working for an international NGO. She keeps all useful addresses in a personal address book on her computer. This list contains all her contacts. When the data set is small, its use relatively simple, and with just one user, simple text files, and a text processor may be used. Text files offer no support for data analysis whatsoever, except perhaps in alphabetical sorting.

Amina has recently collected a number of field observations with measurements that she wants to prepare for statistical analysis In this case, since Amina's data set is still small and numeric by nature, and she has a single type of use in mind, a spreadsheet program will suffice. Spreadsheets do support some data analysis, like averages, sums, minimum and maximum values. All such computations are usually restricted to just a single table of data. However, when data become a large number, spreadsheets may not be enough. Amina has carried out a nationwide census, with many observation stations and/or field observers and all sorts of different measurements. She needs to analyse, manipulate and prepare these data for visualization on a map. In this case, spreadsheets do not accommodate concurrent use of the data set well, although they do support some data analysis, especially when it comes to calculations over a single table, like averages, sums, minimum and maximum values. When one wants to relate the values in the table with values of another nature in some other table, some expertise and significant amounts of time are usually required to make this happen inside the spreadsheet, whereas a database enables to quickly keep track of all the data.

Strengths and weaknesses of GIS and DBMS

How do GIS and databases interact? Let's consider their different characteristics.

GIS	Database
GIS software packages provide support for both spatial and attribute data. They accommodate spatial data storage using the vector and raster approaches and attribute data using tables.	Database management systems (DBMS) have been based on the notion of tables for data storage.

Currently, all major GIS packages provide links to a database that stores attribute data with and make use of its superior data management functions. Spatial (vector or raster) and attribute data are still sometimes stored in separate structures, although they can now be stored directly in a spatial database.

GIS and database

GIS and databases interact to enhance individual strenghts and minimise their weaknesses.

GIS	Database
The strength of GIS lies in its built-in 'understanding' of geographic space and all functions that derive from this, for purposes such as storage, analysis, and map production. GIS packages themselves can	DBMSs have a long tradition in handling attribute (i.e., administrative, non-spatial, tabular, thematic) data in a secure way, for multiple users at the same time. Arguably, DBMSs offer much better table functionality,

store tabular data. However, they do not always provide a fullfledged query language to operate on the tables.	since they are specifically designed for this purpose. A lot of the data in GIS applications is attribute data, so it made sense to use a DBMS for it.

A GIS has spatial data represented with rasters or vectors, and the attribute data stored in an external DBMS. The DBMS serves as a centralized data repository for all users, while each user runs her/his own GIS software, obtaining its data from the DBMS. With raster representations, each raster cell stores a characteristic value. This value can be used to look up attribute data in an accompanying database table. With vector representations, our spatial objects - whether they are points, lines or polygons, or multipoints, multilines, or multipolygons - are automatically given a unique identifier by the system. This identifier is usually just called the 'objectID' or 'featureID' and is used to link the spatial object (as represented in vectors) with its attribute data. The principle applied here is similar to that in raster settings, but in this case each object has its own identifier.