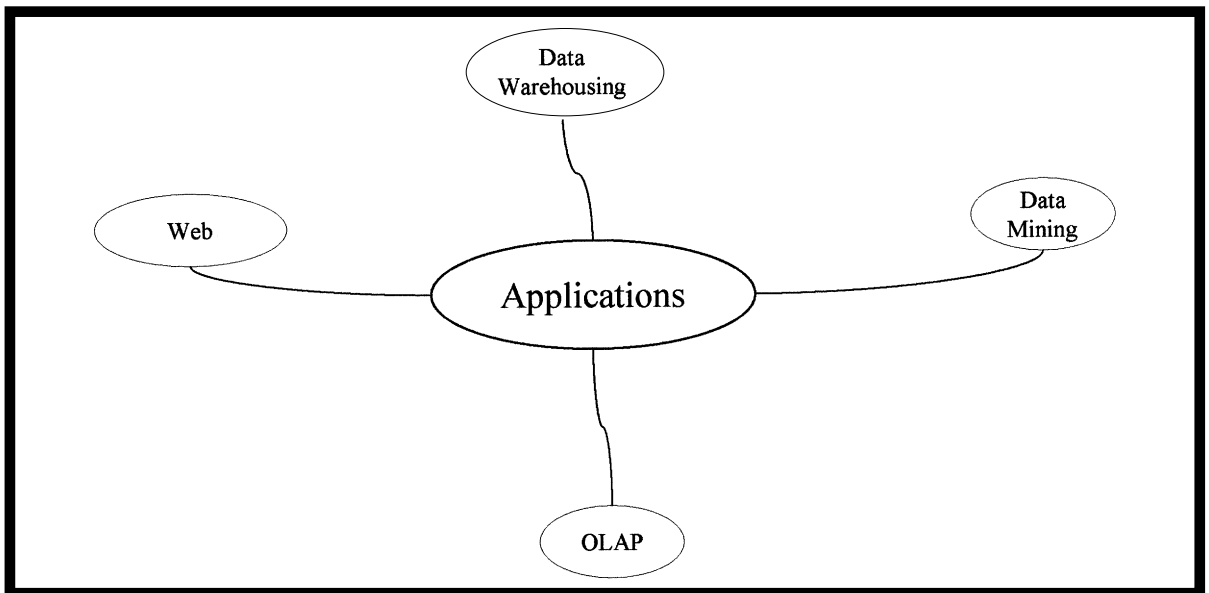




APPLICATIONS OF DATABASE SYSTEMS

There are no such things as applied sciences, only applications of science.

Louis Pasteur (1822–95)



As we mentioned in Part 1, databases have now become the common core of most information systems applications within organisations. The traditional uses for databases have been to store operational data relevant to day-to-day organisational processes such as order-entry, stock control, accounting etc. From such operational data a vast range of what we might call management information can be gleaned. In the past, the approach taken to providing management information was to build specialised systems for capturing and presenting this information. More recently, interest has grown in using sophisticated software tools to meld information from a range of diverse systems and to analyse this data to identify patterns. This is where the interrelated technologies of data warehousing, on-line analytical processing (OLAP) and data mining have a part to play. The common thread through each of these technologies is the central place of the database system.

Data warehousing is the process of integrating operational data from diverse data sources together for the support of decision-support applications such as market analysis and financial forecasting. This is the topic of Chapter 40. On-line analytical processing comprises the support of queries which can rapidly produce aggregate data from the large volumes of data typical of data warehousing applications. This is the topic of Chapter 41. Data mining involves the use of automatic algorithms to extract patterns of data. This is the topic of Chapter 42.

The Internet and its associated technologies have begun to dominate the architecture of ICT systems. Organisations now tend to construct their front-end ICT systems using Internet and Web-based standards that interface to database systems. This is the issue of Chapter 43.

CHAPTER 40

DATA WAREHOUSING

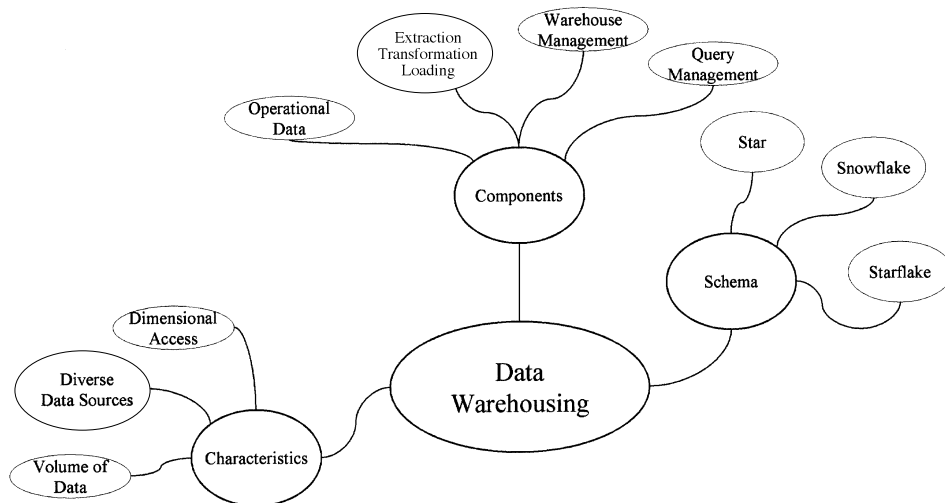
We can never tell what is in store for us.

Harry S. Truman (1884–1972)

LEARNING OUTCOMES

At the end of this chapter the reader will be able to:

- Define the concept of a data warehouse and that of a data mart
- Discuss some of the benefits and challenges of data warehousing
- Describe the steps needed to be taken in a data warehousing project
- Discuss the key components of a data warehouse
- Consider some of the issues involved in designing warehouse schemas



40.1 INTRODUCTION

Conventional database applications have been designed to handle high transaction throughput. Such applications are frequently called on-line transaction processing (OLTP) applications. The data available in such applications is important for running the day-to-day operations of some organisation. The data is also likely to be managed by relational or post-relational DBMS.

Contemporary organisations also need access to historical, summary data and access to data from other sources than that available through DBMS. For this purpose, the concept of a data warehouse has been created (Marakas, 2003). The data warehouse requires extensions to conventional database technology and also a range of application tools for on-line analytical processing (OLAP) (Chapter 41) and data mining (Chapter 42).

In this chapter we provide an overview of the concept of data warehousing. We describe some of the benefits and pitfalls associated with this construct. We also describe some of the necessary components of an architecture for data warehousing and some of the tools needed to construct data warehouses. Included in the discussion is also a brief examination of the issue of database design for data warehousing. Finally we consider the distinction between data warehouses and data marts.

40.2 DEFINITION

A data warehouse is a type of contemporary database system designed to fulfil decision-support needs (Chapter 4). However, a data warehouse differs from a conventional decision-support database in a number of ways:

- *Volume of data.* A data warehouse is likely to hold far more data than a decision-support database. Volumes of the order of over 400 gigabytes of data are commonplace
- *Diverse data sources.* The data stored in a warehouse is likely to have been extracted from a diverse range of application systems, only some of which may be database systems. These systems are described as data sources
- *Dimensional access.* A warehouse is designed to fulfil a number of distinct ways (dimensions) in which users may wish to retrieve data. This is sometimes referred to as the need to facilitate *ad-hoc* query

Inmon (2000) defines a data warehouse as being '*a subject-oriented, integrated, time-variant, and non-volatile collection of data used in support of management decision-making*':

- *Subject-oriented.* A data warehouse is structured in terms of the major subject areas of the organisation such as, in the case of a university, students, lecturers and modules, rather than in terms of application areas such as enrolment, payroll and timetabling


- *Integrated.* A data warehouse provides a data repository which integrates data from different systems with data frequently in different formats. The objective is to provide a unified view of data for users
- *Time-variant.* A data warehouse explicitly associates time with data. Data in a warehouse is only valid for some point or period in time
- *Non-volatile.* The data in a data warehouse is not updated in real-time. Instead, it is refreshed from data in operational systems on a regular basis. A consequence of this is that the management of data integrity is not a critical issue for data warehouses

40.3 THE BENEFITS OF DATA WAREHOUSING

A data warehouse is seen to deliver three major benefits for organisations:

- A data warehouse provides a single manageable structure for decision-support data
- A data warehouse enables organisational users to run complex queries on data that traverses a number of business areas
- A data warehouse enables a number of business intelligence applications such as on-line analytical processing and data mining

The overall objective for a data warehouse is to increase the productivity and effectiveness of decision-making in organisations. This, in turn, is expected to deliver competitive advantage to organisations.

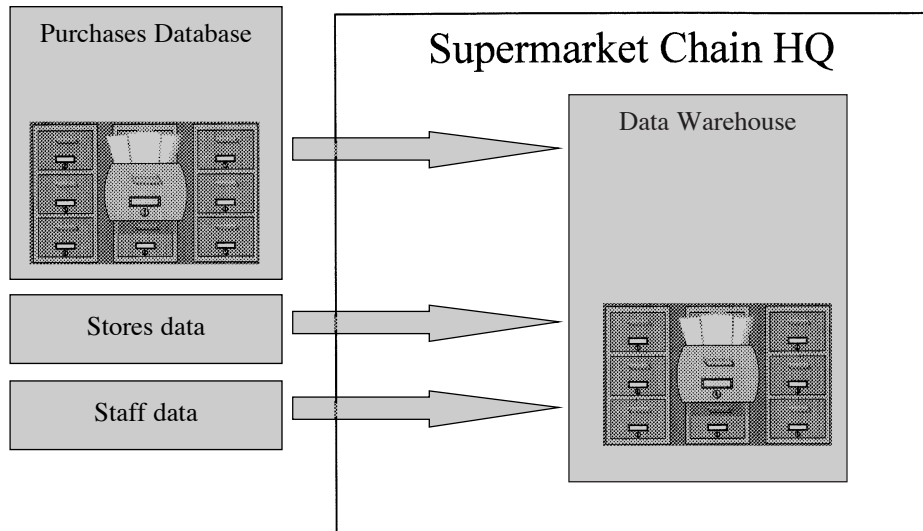
Example  Consider the case of a supermarket chain. At the operational level, sales data is recorded in each supermarket in the chain at checkouts using electronic point-of-sale devices. This data allows administrative staff to record the amount of each type of product sold and, in turn, triggers decisions as to the amount of stock to reorder.

This sales data may be a major data source for a data warehouse perhaps sited at the headquarters of the supermarket chain. This enables headquarters' staff to monitor nationally or perhaps internationally its sales performance in certain areas, which helps it to make decisions as to what it should be selling and for what price.

However, the sales data is likely to be combined with data from other sources such as customer data, collected perhaps through the supermarket chain's loyalty card scheme. Associating sales data with customer data in its data warehouse may provide the chain with important information about the purchasing patterns of its customers. This may enable the chain to proactively plan activities in relation to

particular customer groups or in terms of new business opportunities such as financial services.

The components of this retail application are illustrated in Figure 40.1.



Data Mart = smaller-scale data warehouse

Figure 40.1 Data warehouses/data marts.

40.4 CHALLENGES OF DATA WAREHOUSING

Data warehousing projects are large-scale development projects. Typically a data warehousing project may take of the order of three years. Some of the challenges experienced in such projects are indicated below:

- Knowing in advance what the data users require, and determining the ownership and responsibilities in terms of data sources
- Selecting, installing and integrating different hardware and software required to set up the warehouse. The large volume of data needed in terms of a data warehouse requires large amounts of disk space. This means that estimation of storage volume is a significant activity
- Identifying, reconciling and cleaning existing production data and loading it into the warehouse. The diverse sources of data feeding a data warehouse introduces problems of design in terms of creating a homogeneous data store. Problems are also introduced in terms of the effort required to extract, clean and load data into the warehouse

- Managing the refresh or update process. Once a warehouse is established, a clear scheme of effectively managing the high volume of complex, *ad-hoc* queries needs to be produced

40.5 STEPS IN BUILDING A DATA WAREHOUSE

The key steps involved in a data warehousing project are outlined below (Inmon, 2000):

- Users specify information needs
- Analysts and users create a logical and physical design
- Sources of data are identified in operational systems, external sources etc.
- Source data is scrubbed, extracted and transformed
- Data is transferred and loaded into the warehouse periodically
- Users are given access to the warehouse data
- The warehouse is maintained in terms of changing requirements

Note that the first two stages are the same as those for conventional database development (Chapter 14). Also, identifying and managing data sources may be a key activity of the data administration function (Chapter 22).

40.6 COMPONENTS OF A DATA WAREHOUSE

Figure 40.2 illustrates some of the major components of a data warehouse:

- *Operational data.* Data for the warehouse may be sourced in a number of ways, e.g. from mainframe-based hierarchical or network databases, from relational databases and from data in proprietary file systems
- *Extraction, transformation and loading functions.* These ETL operations or functions are concerned with extracting data from source systems, transforming it into a suitable form and loading the transformed data into the data warehouse
- *Warehouse management.* A series of functions must be provided to manage the warehouse: consistency analysis, indexing, denormalisation, aggregation, backup and archiving
- *Query management.* The warehouse must perform a series of operations concerned with the management of queries for use by a variety of actors: reporting and query tools, OLAP tools or tools for data mining

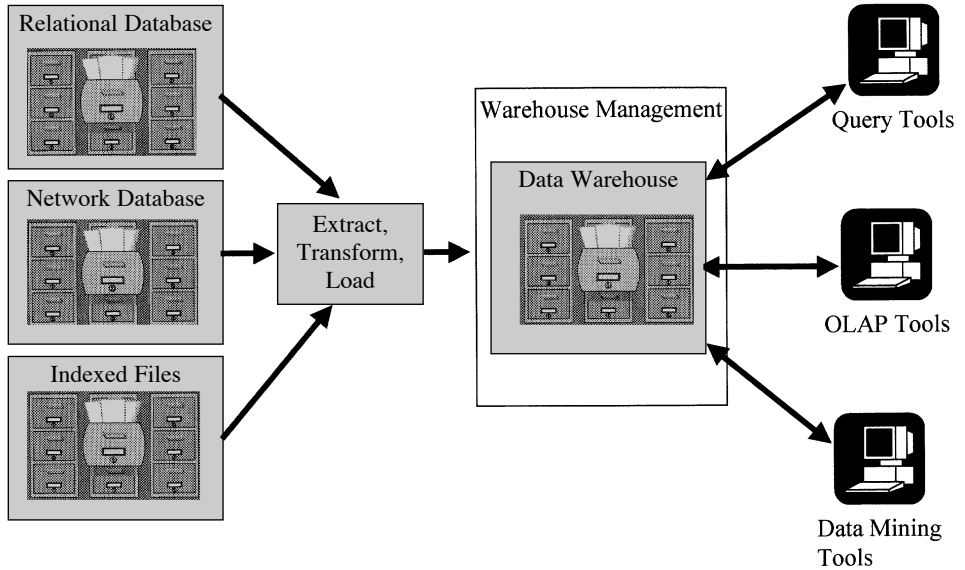




Figure 40.2 Components of a data warehouse.

Example  Clickstream records are records of the behaviour of users of Web-sites. Capture and analysis of such data is seen to be critical to e-Commerce activity (Chapter 5) (Kahavi *et al.*, 2002). Such data needs to be integrated with demographic and other behavioural data in large data warehouses. OLAP and data mining applications (Chapters 41 and 42) may then be run to support strategic decision-making in companies.

40.7 EXTRACTION, TRANSFORMATION AND LOADING

As mentioned in the previous section, a number of tools exist to:

- Extract data from source systems such as database and file systems
- Perform necessary transformation of such data
- Load the transformed data into the data warehouse

Example  Transformation of data may include cleansing data of rudimentary errors, summarising data in various ways, and converting and unifying various ways of coding data. For instance, source system 1 may use the codes 'M' for male and 'F' for female, while source system 2 may use the codes 1 and 2 for male and female. Transformation functions will involve converting both forms of coding into a common form acceptable to the data warehouse.

Such ETL tools effectively constitute middleware between source systems and the data warehouse, and may be built in a number of ways which include:

- *Code generators.* These create transformation programs based on data definitions provided for both source and target systems. The main problem is the large number of programs needed for all transformations required by a particular data warehousing application
- *Data replication tools.* These employ database triggers or a recovery log to capture transformations on a single data source from one system. Such changes are then applied to a copy of such data stored on a target system. Typically, the complexity of data transformations possible is limited
- *Dynamic transformation engines.* These are rule-driven engines that capture data from source systems at regular, predefined intervals. Transformations are applied and the data is loaded into a target environment

40.8 FORMS OF DATA IN A DATA WAREHOUSE

We may distinguish a number of different forms of data in a data warehouse:

- *Detailed data.* This comprises the detailed production data. Usually, detailed data is not stored on-line but is aggregated on a periodic basis. Sales data from the supermarket application described above is an example of detailed data
- *Summarised data.* Data in the warehouse is normally summarised or aggregated to speed up the performance of queries. The data may be lightly summarised or highly summarised. Summarised data needs to be updated periodically when the detailed data is refreshed. As an example, sales data may be summarised in terms of particular geographical areas, time-periods and/or product lines
- *Meta-data.* Data about data is needed to enable the extraction, transformation and loading processes by mapping data sources to the warehouse schema. Meta-data is also used to automate the production of summary data and to facilitate query management
- *Archive data.* Data needs to be periodically archived from the warehouse to prevent the database growing too large for its platform. Normally, this is done on the basis of some retention period established for data
- *Backup data.* Just as in conventional databases, detailed, summary and meta-data need to be backed up regularly in order to recover from failure

40.9 DATA MARTS

A data mart is a restricted data warehouse. It may be restricted in a number of ways:

- *Type of data.* A data mart may be limited to a particular type of input data such as that available from particular DBMS
- *Business area.* A data mart may be designed to store data representing a particular business area rather than representing data applicable to the entire organisation
- *Geographic area.* A data mart may be set up for one specific geographic area in terms of the activities of some organisation

Maintaining restrictions such as these means that whereas a data warehouse may store of the order of hundreds of gigabytes of data, a data mart may typically store something of the order of tens of gigabytes of data. This means that a data mart can usually be built more rapidly than a data warehouse and is easier to manage.

40.10 DESIGNING THE DATA WAREHOUSE SCHEMA

Designing a schema for a data warehousing application is a specialist case of database design. The methodology and techniques discussed in Part 4 are as relevant to data warehousing applications as they are to conventional database systems. However, two issues assume particular prominence in a data warehouse: the large volume of data and the issue of achieving satisfactory levels of retrieval performance. To provide satisfactory levels of performance frequently means designing specialised physical schemas for warehousing applications. In this section we consider three generic designs for warehousing schemas proposed by Anahory and Murray (1997).

40.10.1 STAR SCHEMAS

Star schemas are physical schemas that store what Anahory and Murray term 'factual data' in a central table surrounded by reference tables storing data relevant to particular dimensions needed for decision-support. The central factual data should be designed to store data generated by past events in relation to some organisational activity. Hence, it constitutes read-only data and means that the size of the fact table can be extremely large in comparison to that of the reference tables. Figure 40.3 illustrates a possible star schema relevant to the academic domain. At the centre we store 'facts' about the assessment of students. Three reference tables are associated with this central table, allowing us to analyse the factual data in terms of student characteristics, lecturer characteristics or assessment characteristics.

Each reference table may be a denormalised version of a number of other tables in order to improve retrieval performance. For instance, in the schema illustrated in Figure 40.3, the table assessment contains module information and the table staff contains school information.

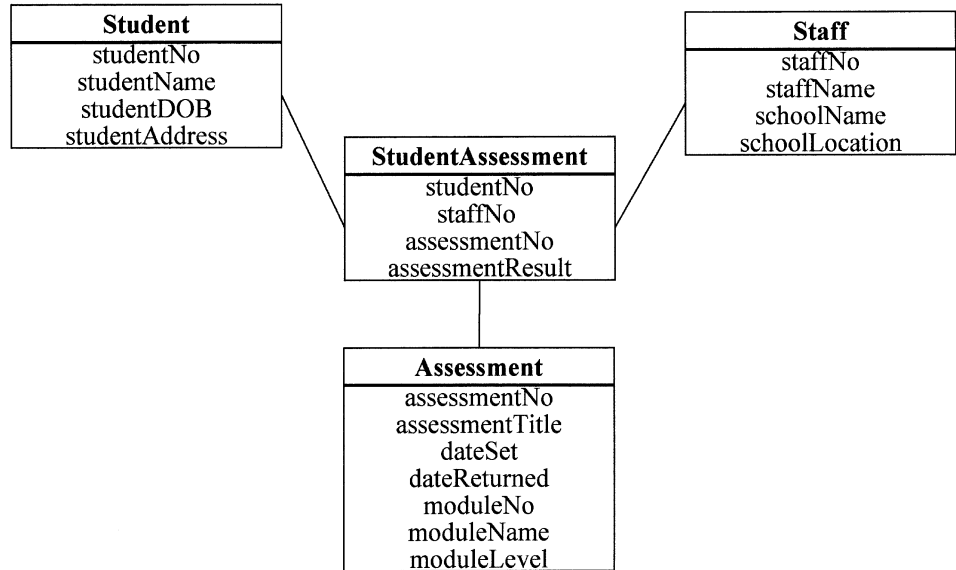


Figure 40.3 A star schema.

40.10.2 SNOWFLAKE SCHEMA

Snowflake schemas are a variation on the star schema. In a snowflake schema, each dimension can have a number of its own dimensions. This means that reference tables are not denormalised in a snowflake schema. Figure 40.4 illustrates a snowflake schema for the university application.

40.10.3 STARFLAKE SCHEMA

Starflake schemas occupy the middle ground between star and snowflake schemas. In a starflake schema, some of the reference tables will be denormalised and some will be normalised. Figure 40.5 provides an example of a starflake schema. In this schema the assessment information is now modelled on the lines of a normalised snowflake pattern, while the staff table is still denormalised.

40.11 CASE STUDY: ORACLE 9i

Oracle 9i Enterprise Edition describes itself as a DBMS specifically designed for data warehousing applications. The key features offered in this version, specifically pinpointed at data warehousing, include:

- *Summary management.* The DBMS is able to automatically maintain stored aggregates of data from base tables. These can be used to improve the performance of queries which attempt to summarise data on the basis of common

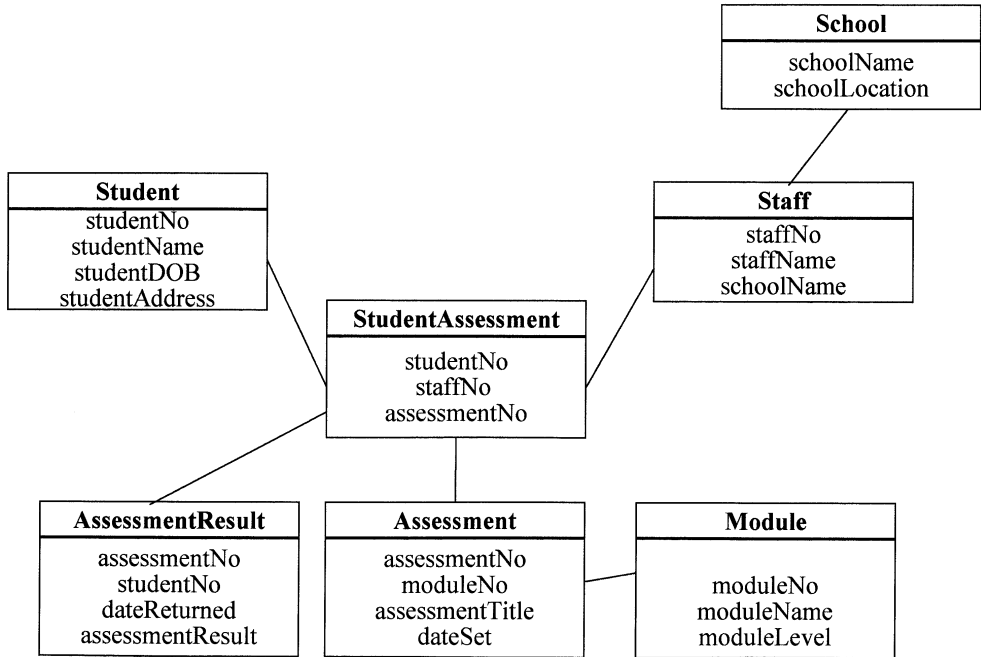


Figure 40.4 A snowflake schema.

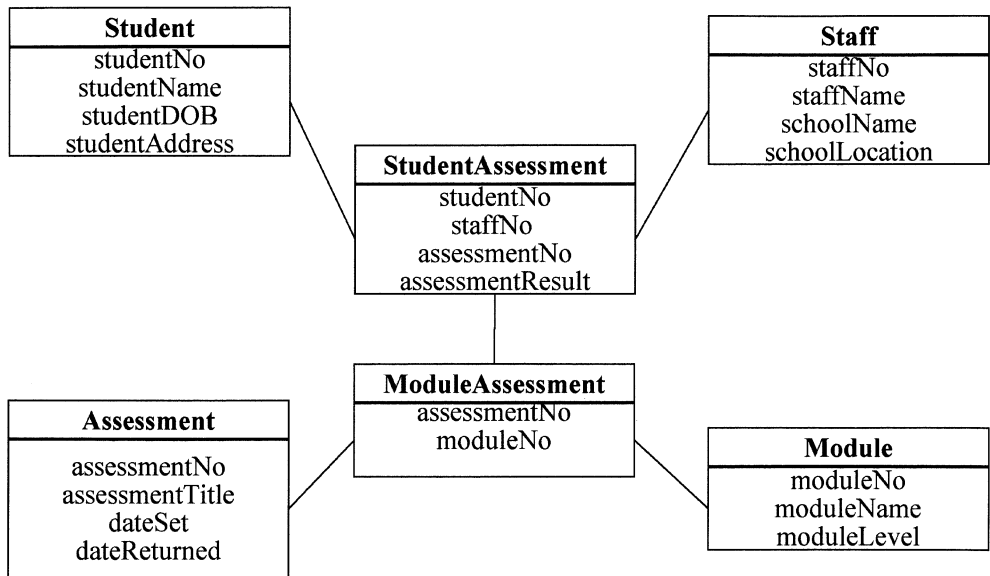


Figure 40.5 A starflake schema.

dimensions. The DBMS includes an advisor facility for the DBA which helps choose summary tables on the basis of database workload and usage statistics

- *Analytical functions.* The DBMS provides a number of added SQL functions for data warehousing work, including the ability to rank data and move aggregates across time dimensions
- *Bitmapped indexes.* Bitmapped indexes are particularly suitable for improving the performance of queries which include a wide range of criteria. The DBMS uses data compression technology to store such indexes efficiently
- *Advanced joins.* The DBMS supports advanced forms of joins such as partition joins and hash joins. Partition joins efficiently execute on tables that have been partitioned on the basis of a join key. Hash joins efficiently execute for operations that demand sorted data or parallel execution (Chapter 38)
- *Cost-based optimiser.* ORACLE's cost-based optimiser (Chapter 30) chooses the most effective execution strategy for a query based on detailed statistics held about the database. This can prove effective for queries against star schema
- *Resource management.* Effective management of resources is critical to data warehousing applications. The ORACLE DBMS uses the idea of a resource class to which can be assigned a defined set of resources and a defined group of users

40.12 SUMMARY

- A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data used in support of management decision-making
- A data mart is a small data warehouse, designed usually for a particular business area
- Data warehouses and data marts deliver decision-support applications in organisations
- Data warehousing projects are large-scale development projects
- The components of a data warehouse include: production data, extraction, transformation and loading functions, warehouse management functions and query management functions
- Three specialist forms of schema are relevant to data warehousing applications: star, snowflake and starflake schemas

40.13 ACTIVITIES

1. Consider the use of data warehousing in a university setting. What would constitute the operational data in this case?

2. For what purposes would a data warehouse be established in this domain?
3. What sort of data marts might be appropriate?

40.14 REFERENCES

- Anahory, S. and D. Murray (1997). *Data Warehousing in the Real World: A Practical Guide for Building Decision-Support Systems*. Harlow, UK, Addison-Wesley.
- Inmon, W.H. (2000). *Building the Data Warehouse*. New York, John Wiley.
- Kahavi, R., N.J. Rothleder and E. Simoudis (2002). Emerging trends in business analytics. *Communications of the ACM* **45**(8): 45–53.
- Marakas, G.M. (2003). *Modern Data Warehousing, Mining, and Visualization: Core Concepts*. Upper Saddle River, NJ, Prentice-Hall.

CHAPTER 41

ON-LINE ANALYTICAL PROCESSING

Half of analysis is anal.

Marty Indik

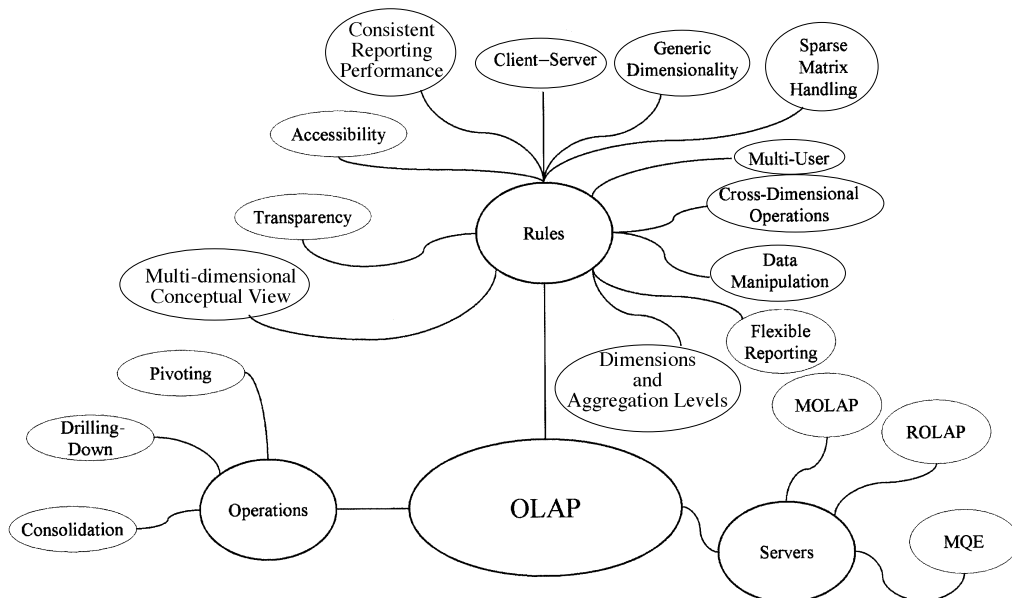
It requires a very unusual mind to undertake the analysis of the obvious.

Alfred North Whitehead (1861–1947)

LEARNING OUTCOMES

At the end of this chapter the reader will be able to:

- Define the concept of OLAP and the operations it supports
- Relate Codd's rules for OLAP tools
- Discuss the essential features of OLAP tools
- Describe different types of OLAP servers



41.1 INTRODUCTION

Modern database applications such as market analysis and financial forecasting require access to large databases for the support of queries which can rapidly produce aggregate data. Such applications are frequently called analytical on-line processing applications, or OLAP for short. OLAP is contrasted with on-line transaction processing, or OLTP.

The increasing need for supporting such applications has led to the development of specialised tools, used normally in association with data warehouses or data marts. This chapter provides a brief review of the area.

41.2 DEFINITION


OLAP is a term coined by Tedd Codd (Codd *et al.*, 1993) to define an architecture that supports complex analytical operations such as consolidation, drilling-down and pivoting:

- *Consolidation*. Also referred to as rolling-up, consolidation comprises the aggregation of data, such as modules data, being aggregated into courses data, and courses data being aggregated into schools or departments data
- *Drilling-down*. This is the opposite of consolidation and involves revealing the detail or disaggregating data, such as breaking down school-based data into that appropriate to particular courses or modules
- *Pivoting*. This operation, sometimes referred to as ‘slicing and dicing’, comprises the ability to analyse the same data from different viewpoints, frequently along a time axis. For example in the university domain, one slice may be the average degree grade per course within a school; another slice may be the average degree grade per age band within a school

To support these applications, OLAP applications tend to utilise special-purpose, multi-dimensional data structures.

41.3 MULTI-DIMENSIONAL DATA STRUCTURES

OLAP systems use multi-dimensional data structures to store data and relationships. Such data structures are frequently visualised as data cubes or data cubes of data cubes. Each side of a cube represents a data dimension of relevance to some organisation. The intersection of each dimension forms a cell. Within each cell a data value is stored. Each such data value may be derived from some other cube representing usually some lower levels of aggregation.

Example  Figure 41.1 illustrates the organisation of data in terms of part of a data cube for an academic application. Here, three dimensions are relevant: the degree level (MSc, BSc), the year and semester, and the course. In each cell of the cube the average examination mark is stored. Such a structure enables us to obtain answers to the following queries:

- The average examination mark per course (*pivot* on course), or the average examination mark per year/semester (*pivot* on year/semester), or the average examination mark per level of a course (*pivot* on level)
- The six distinct courses represented consist of all those offered by the School of Computing. Hence we might *consolidate* to the school level to answer a question about the average examination mark per school
- A given cube may consist of a hierarchical pre-aggregation of data. For instance, each examination average may be produced from a hierarchy of the module examination grades relevant to each course. This would enable us to *drill-down* to obtain the specific examination grades per module

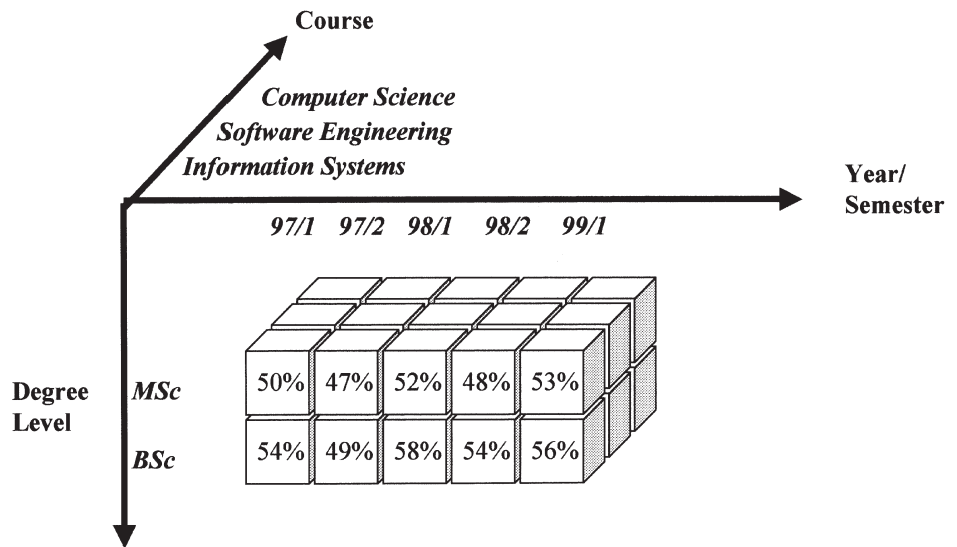


Figure 41.1 Multi-dimensional data.

41.4  SPARSITY

In terms of many applications, not all the cells in a data cube will contain data. In fact, in large applications with many dimensions, the vast majority of cells will contain no data. This is referred to as sparsity and means that a data cube is normally a sparse data structure.



Suppose we have a sales application in which the relevant dimensions are product type, customer, time of sale, location of sale, value of sale and quantity sold. In such a data cube, most of the cells will be empty because in any one period each customer will buy only a small proportion of the product range.

Because of this, OLAP servers will normally have the ability to store such data in compressed form that maximises space utilisation. Data that exists in a high percentage of the cells in a cube – so-called dense data – can be stored separately from sparse data – data in which a significant percentage of the cells of a cube is empty. This ability means that OLAP servers can both minimise physical storage capacity for data as well as making it possible to load more data into computer memory at any one time. This latter capability improves processing performance.

41.5



CODD'S RULES FOR OLAP TOOLS

In a similar manner to his twelve rules for relational DBMS, Codd has formulated twelve rules for selecting tools for OLAP:

- *Multi-dimensional conceptual view.* OLAP tools should provide a multi-dimensional model which corresponds with the user's views of the organisation. Such a model should also be easy-to-use
- *Transparency.* The technology used, the underlying database architecture and the various data sources should be transparent to the user
- *Accessibility.* The OLAP tool should be able to access data from sources in different formats – relational, non-relational and in terms of legacy systems
- *Consistent reporting performance.* The user should not perceive any degradation in performance when the number of dimensions, aggregations or the size of the database increases
- *Client-server architecture.* The OLAP tool should be able to work effectively in a client-server environment
- *Generic dimensionality.* Every dimension in the database must be equivalent in both structure and operational capabilities
- *Dynamic sparse matrix handling.* The OLAP tool should be able to adapt its physical organisation to optimise the handling of sparse matrix handling
- *Multi-user support.* The OLAP tool should be able to handle concurrent access to data
- *Unrestricted cross-dimensional operations.* The OLAP tool must be able to support dimensional hierarchies and automatically perform consolidation calculations within and across dimensions

- *Intuitive data manipulation.* Pivoting, drill-down and consolidation should be accomplished using classic graphical user interface operations such as point-and-click and drag-and-drop
- *Flexible reporting.* It must be possible to arrange rows, columns and cells in a fashion that meets the needs of particular users
- *Unlimited dimensions and aggregation levels.* The OLAP tool should not impose any restriction on the number of dimensions or aggregation levels in an analytical model

41.6 OLAP SERVERS

An OLAP application will normally reside on a specialised OLAP server, and the OLAP server will exist in the application layer of a three-tier client-server environment. The OLAP server uses multi-dimensional structures to store data and relationships. Three different types of OLAP tool may run on the OLAP server (Berson and Smith, 1997):

41.6.1 MULTI-DIMENSIONAL OLAP (MOLAP)

MOLAP tools use data structures founded in array technology and utilise efficient storage techniques to optimise disk management through sparse data management. Typically such tools demand a close coupling between the presentation and application layer of the three-tier architecture. Requests are issued from access tools in the presentation layer to the MOLAP server. The MOLAP server periodically refreshes its data from base relational and/or legacy systems (Figure 41.2).

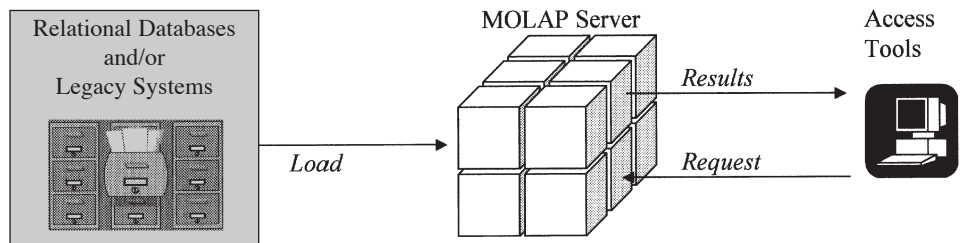


Figure 41.2 MOLAP server.

41.6.2 RELATIONAL OLAP (ROLAP)

Sometimes called multi-relational OLAP, this has been the fastest growing area of OLAP tools. ROLAP utilises connection to existing RDBMS products through the maintenance of a meta-data layer. This meta-data layer allows the creation of multiple multi-dimensional views of two-dimensional relations. Some

ROLAP products maintain enhanced query managers to enable multi-dimensional query, others emphasise the use of highly-denormalised database designs (typically a 'star' schema) to facilitate multi-dimensional access (Figure 41.3).

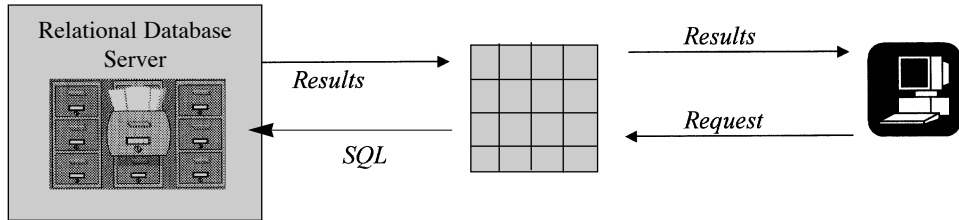


Figure 41.3 ROLAP server.

41.6.3 MANAGED QUERY ENVIRONMENT (MQE)

MQE systems are an intermediate form between ROLAP and OLAP servers. They provide limited capability to access either RDBMS directly or through using an intermediate MOLAP server. When interacting with the MOLAP server, data will be delivered in the form of a data cube to the desktop where it is stored and can be accessed locally (Figure 41.4).

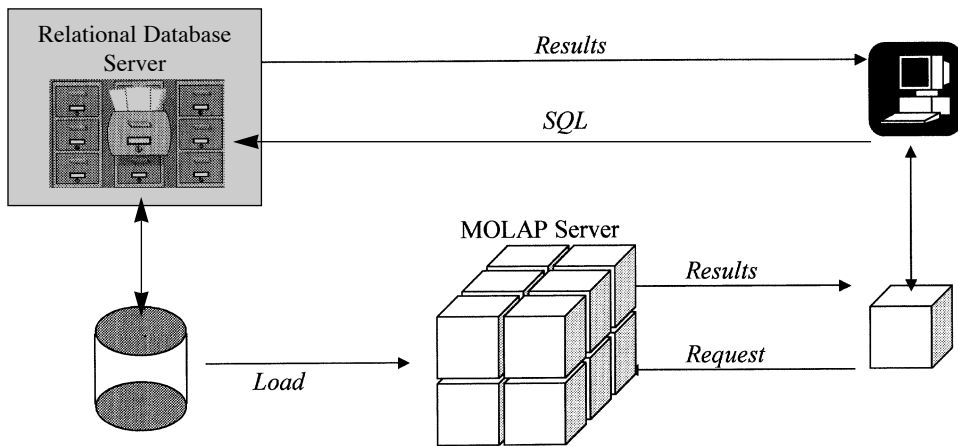


Figure 41.4 Managed query environment.

41.7 © CASE STUDY: OLAP EXTENSIONS TO SQL

One of the key problems of a number of the standards for SQL discussed in Chapter 11 is that the language does not contain sufficient functionality for business analysts wishing to compute moving averages, cumulative sums and other statistical functions. For this reason, a number of additions to the



standard in this area have been proposed to support OLAP applications. A number of vendors of database technology, such as IBM and ORACLE, have been instrumental in this movement. To illustrate some of the principles of using SQL in this manner, consider the case in which we wish to show the monthly sales figures for branch office 001 along with monthly year-to-date figures for this branch. Assume that we have an appropriate database with a table called BranchQuarterlySales. In this table there are three columns – branchNo, quarter and quarterlySales. Further assume that one of the OLAP extensions to SQL is a CUME function which enables us to compute a running or cumulative total of a specified column's value. A relevant SQL query might then be:

```
SELECT quarter, quarterlySales CUME(quarterlySales) AS yearToDate
FROM BranchQuarterlySales
WHERE branchNo = '001'
```

The resulting output from this query might then look like:

quarter	quarterlySales	yearToDate
1	20000	20000
2	30000	50000
3	25000	75000
4	35000	110000

41.8 SUMMARY

- OLAP involves the multi-dimensional analysis of large volumes of data
- OLAP applications tend to utilise special-purpose multi-dimensional data structures. Such data structures are frequently visualised as cubes or cubes of cubes with each side of a cube being a dimension
- OLAP systems support complex analytical operations such as consolidation, drilling-down and pivoting
- Codd formulated twelve rules for selecting tools for OLAP
- There are three main categories of OLAP tool: multi-dimensional OLAP, relational OLAP and managed query environment

41.9 ACTIVITIES

1. Write a report analysing the applicability of OLAP technology to an organisation such as a university.
2. Identify a given OLAP product and determine its type.
3. Determine the key uses of OLAP technology in modern business.

41.10  REFERENCES

Berson, A. and S.J. Smith (1997). *Data Warehousing, Data Mining and OLAP*. New York, McGraw-Hill.

Codd, E.F., S.B. Codd and C.T. Salley (1993). *Providing OLAP (On-Line Analytical Processing) to User-Analysts: An IT Mandate*. Ann Arbor, MI, Arbor Software Corporation.

CHAPTER 42

DATA MINING

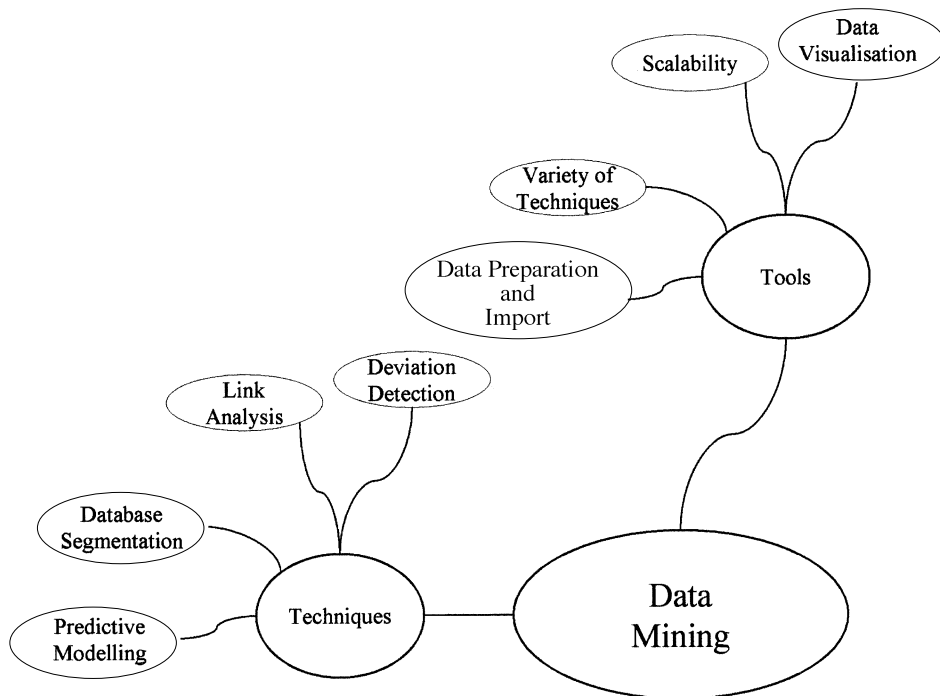
Aristotle maintained that women have fewer teeth than men; although he was twice married, it never occurred to him to verify this statement by examining his wife's mouths.

Bertrand Russell (1872–1970)

LEARNING OUTCOMES

At the end of this chapter the reader will be able to:

- Define the concept of data mining
- Discuss the essential features of data mining techniques



42.1 INTRODUCTION


Data mining is normally used in association with data warehouses and data marts. To gain benefit from a data warehouse or mart, the data patterns resident in the large data-sets characteristic of such applications need to be extracted. As the size of a data warehouse grows, the more difficult it is to extract such data using the conventional means of query and analysis. Data mining involves the use of automatic algorithms to extract this data.

42.2 DEFINITION

Data mining is the process of extracting previously unknown data from large databases and using it to make organisational decisions (Kantardzic, 2002). There are a number of features to this definition:

- Data mining is concerned with the discovery of hidden, unexpected patterns of data
- Data mining usually works on large volumes of data. Frequently, large volumes are needed to produce reliable conclusions in relation to data patterns
- Data mining is useful in making critical organisational decisions, particularly those of a strategic nature

Data mining began its life in specialist applications such as geological research (searching for natural resources) and meteorological research (weather forecasting). More recently it has been applied in a number of areas of industry and commerce.

Examples  Examples of such business applications are listed below:

- In retail chains, data mining has been used to identify the purchasing patterns of customers and associating this data with demographic characteristics such as the age and class profile of customers. Such patterns are useful in making decisions such as what products to sell in which retail stores and when
- In the insurance industry, data mining has been used to analyse the claims made against insurance policies and hence feed into actuarial decisions such as the pricing of particular policies
- In the finance industry, banks have used data mining techniques to identify fraudulent credit card use among its transaction data
- In the medical domain, data mining may be applied to identify successful medical treatments for particular medical complaints
- In direct-mail targeted marketing, retailers must be able to identify subsets of the population likely to respond to their promotions



42.3 DATA MINING OPERATIONS OR TECHNIQUES

Data mining is implemented in terms of a number of operations or techniques (Cabena *et al.*, 1997). There are four main operations associated with data mining: predictive modelling, database segmentation, link analysis and deviation detection. Particular operations are usually associated with particular applications. For instance, credit card fraud detection is likely to be implemented by predictive modelling, while profiling of customers is likely to be implemented by database segmentation.

Each operation may be implemented using a number of different techniques or algorithms. Since each particular technique has its own strengths and weaknesses, data mining tools may offer a choice of techniques to implement an operation:

42.3.1 PREDICTIVE MODELLING


Predictive modelling is an attempt to model the way in which humans learn. An existing data-set is analysed to form a model of its essential characteristics. This model is developed using a supervised learning approach consisting of two phases. The first, training, phase involves building a model using a large sample of data known as the training set. The second, testing, phase involves trying out the model on new data. Predictive modelling is associated with the techniques of classification and value prediction.

Classification involves determining a class for each row in a database. This may be done using decision trees or neural networks.

A decision tree (Beynon-Davies, 1998) represents a classification problem as a series of conditions. Each node represents a condition and each branch a specific response to a condition. The leaf nodes of the tree represent the range of classes into which a row might be classified. This tree will be induced from an analysis of the input data-set.

A neural network comprises a network of nodes. The nodes are divided up into three layers: an input layer, a processing layer and an output layer. Nodes in the input layer are activated depending on the characteristics of the data input into the network. The input nodes activate in turn the processing nodes, the 'strength' of the activation being dependent on weights assigned to the relationships between nodes. The processing nodes determine the relevance of applicable output nodes (the classes in the problem) based on these weights. The weights assigned to the relationships between nodes will be constructed in a phase in which the network 'learns' from a sample data-set.

Value prediction is a technique used to estimate a value associated with a database row. The technique generally uses the well-established statistical algorithms associated with linear regression. Linear regression attempts to build the best fit of a straight line through a plot of data.

Example  A fictitious example of a classification approach to data mining is illustrated in Figure 42.1. Here a pattern identified in a data-set of student applications for university courses is represented as a simple decision tree. The pattern induced from the existing data on student applications may be used in a number of ways. For example, it might be used to target marketing material at particular populations or it might be used to construct campaigns to address an obvious gender imbalance in applications for university courses.

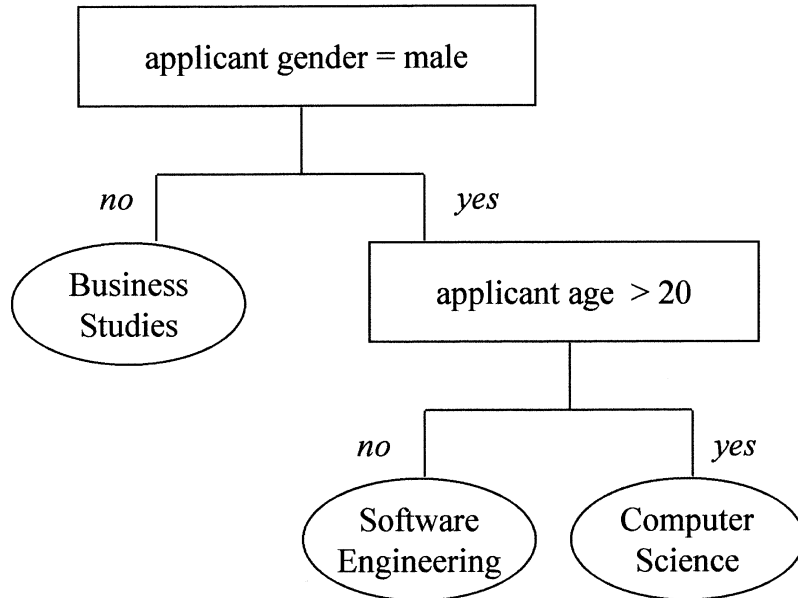



Figure 42.1 A tree induction example.

42.3.2 DATABASE SEGMENTATION

Database segmentation involves partitioning the database into a number of segments or clusters. Each segment comprises a set of rows having a number of properties in common. Segments are built and refined using a process of unsupervised learning.

Examples  An example of the application of this technique is in the area of bank note forgeries. A database might have been constructed in terms of observations recorded against a population of banknotes. Performing segmentation on this data-set allows us to cluster the banknotes into distinct sets in terms of dimensions such as size, colour etc. This may allow us not only to distinguish forgeries from legal tender, it may also allow us to distinguish particular types of forgery, perhaps associated with particular criminal groups.



Database segmentation may also be applied in market basket analysis (Witten and Frank, 2000). This is the use of association techniques to find groups of items associated with consumer transactions. For example, in a food supermarket a company gathers data about purchasing at checkouts. From conducting data mining on such data we determine that customers often purchase beer and disposable nappies at the same time. This may be because parents with young children typically stock up with both for the weekend. Such associations can be used for a number of purposes, such as planning store layouts and discounting of products sold together.

42.3.3 LINK ANALYSIS

Link analysis attempts to establish associations between individual records in the data-set. Such associations are frequently represented as rules in a technique known as associations discovery.

Example ■■■▶ For instance, an analysis of data collected on student progression may identify a rule such as *if a computing student passes a first year programming module then in 40% of cases he or she will gain an upper second class degree.*

In a technique known as sequential pattern discovery, the associations are represented in terms of the presence of one set of data-items being followed by another set of data-items.

Example ■■■▶ An example here might be the patterns identified in student module enrolments, allowing us to predict the likely take-up of final-year module options. Similarly, time sequence discovery identifies patterns between data which are time-dependent. For instance, a link analysis of customer purchasing patterns using this technique might find that within a certain time period after making a house purchase, buyers purchase items such as washing machines, refrigerators etc.

42.3.4 DEVIATION DETECTION

Deviation detection involves identifying so-called outliers in the population of data – i.e. those that deviate from some norm. Deviation detection can be detected by statistical and visualisation techniques such as linear regression. A classic example of deviation detection is that applicable to the quality control associated with manufacturing. Analysis of defect data associated with the production process may allow a company to identify reasons for the introduction of faults and may lead to improvements in production processes.

42.4 DATA MINING TOOLS

Data mining tools are likely to incorporate the following facilities:

- *Data preparation and import facilities.* The data mining tool should be capable of importing data from the target environment
- *Selection of data mining operations and techniques.* The data mining tool should provide a range of algorithms for pattern detection
- *Product scalability and performance.* The tool should be capable of handling sufficient volumes of data and should offer satisfactory levels of performance in terms of data manipulation operations
- *Facilities for data visualisation.* The tool should be capable of presenting a number of different graphical representations of patterns detected

42.5 CASE STUDY: DATA MINING OF MEDICAL DATA

In the 1990s, approximately 10% of the population of Singapore were diabetic. This is a medical condition with many side-effects such as increased risk of eye disease and kidney failure. Early detection of the condition and proper care management can significantly improve the health and longevity of sufferers.

To combat the disease, the government of Singapore introduced in 1992 a regular screening programme for diabetic patients in public hospitals. This programme generated a vast amount of data about patients, including patient details, clinical symptoms, eye-disease diagnoses and treatments. Over ten years of data has now been collected and made available for data mining. The aim is to use data mining to discover rules to enable physicians to better understand the association of the disease with particular population segments.

Before the data could be used it had to be cleaned of typographical errors, missing values and incorrect values. Rule generation algorithms applied to the data also produced too many rules to be practically used. Therefore, data pruning techniques had to be used to remove insignificant rules. Data visualisation software was also used to present the rules for easy manipulation by physicians.

42.6 SUMMARY

- Data mining is the process of extracting previously unknown data from large databases and using it to make organisational decisions
- There are four main operations associated with data mining: predictive modelling, database segmentation, link analysis and deviation detection
- Each operation may be implemented using different algorithms
- A particular software vendor may offer a number of algorithms to implement an operation



42.7 ACTIVITIES

1. Write a brief report identifying possible applications of data mining in an organisation such as a university.
2. For the business applications listed in section 42.2, identify the most appropriate data mining technique.

42.8 REFERENCES

- Beynon-Davies, P. (1998). *Information Systems Development: An Introduction to Systems Engineering*. Basingstoke, Macmillan (now Palgrave).
- Cabena, P., P. Hadjinian, R. Stadler, J. Verhees and A. Zanasi (1997). *Discovering Data Mining from Concept to Implementation*. Englewood Cliffs, NJ, Prentice-Hall.
- Kantardzic, M. (2002). *Data Mining: Concepts, Models, Methods and Algorithms*. Chichester, John Wiley.
- Witten, I.H. and E. Frank (2000). *Data Mining*. San Francisco, CA, Morgan Kaufman.

CHAPTER 43

DATABASES AND THE WEB

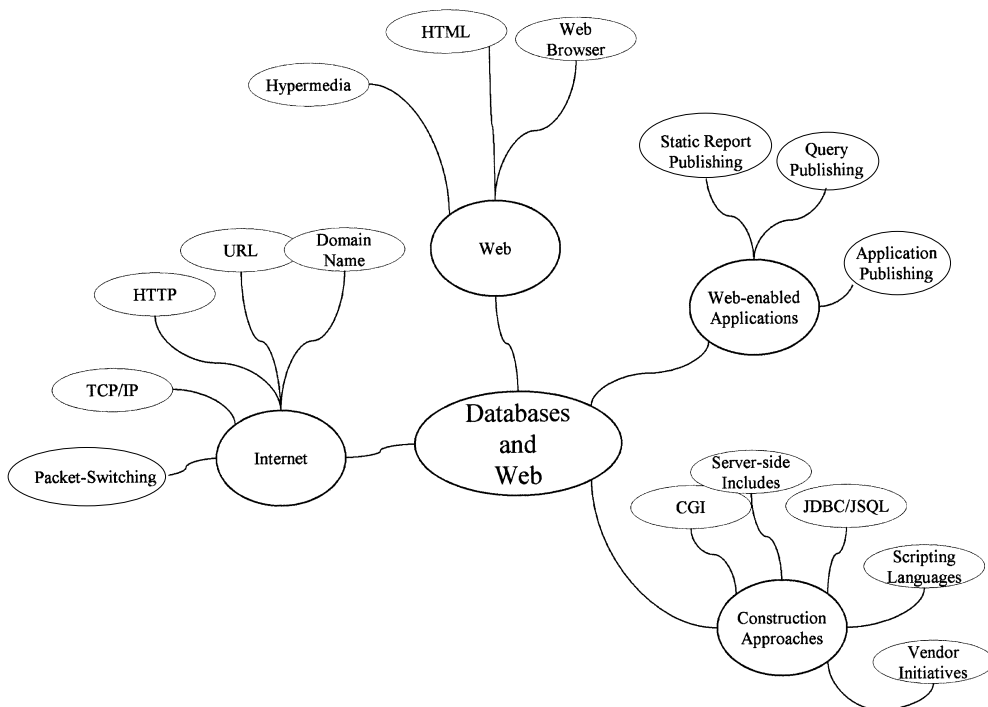
All men are caught in an inescapable network of mutuality.

Martin Luther King Jr (1929–68)

LEARNING OUTCOMES

At the end of this chapter the reader will be able to:

- Define the concept of the Internet
- Describe the key components of the Web
- Relate the place of database systems in Internet and Web applications
- Provide an overview of a number of different ways of building Web-enabled database applications



43.1 INTRODUCTION

The Internet – short for inter-network – began as a Wide Area Network in the USA, funded by its Department of Defense to link scientists and researchers around the world. It was initially designed primarily as a medium to exchange research data.

Currently the Internet is a set of interconnected computer networks distributed around the globe. The Internet can be considered on a number of levels. At the base level we have the technical infrastructure of the Internet which is composed of packet-switched networks and a series of communication protocols. On this layer runs a series of applications such as electronic mail (e-mail) and more recently the World-Wide-Web.

In this chapter we provide an overview of the Internet and the Web. We describe some of the ways in which these technologies are affecting the construction of ICT system applications. We also consider some of the reasons for the use of database systems within Internet/Web applications. Finally we conclude with a consideration of a number of ways in which Web-enabled database applications can be built.

43.2 INTERNET INFRASTRUCTURE

The technical infrastructure of the Internet consists of a number of components. These include:

- Packet-switched networks
- TCP/IP
- HTTP
- IP addresses
- URLs
- Domain names

43.3 PACKET-SWITCHED NETWORKS

The early computer networks were modelled on the local and long-distance telephone networks that dated back to the early 1950s. Computer networks tended to be composed of leased telephone lines. In these traditional telephone networks, a connection between a caller and the receiver was established through telephone switching equipment (both mechanical and computerised) selecting specific electrical circuits to form a single path. Once the connection was established, data travelled along the path. This is known as a circuit-switching network.

Circuit-switching works well for telephone communication but proves

expensive for data communication because of the need to establish a point-to-point connection for each pair of senders/receivers. Most computer networks therefore use a form of network technology known as packet-switching. In such a network the data in a message or file is broken up into chunks known as packets. Each packet is electronically labelled with codes that indicate the sender (origin) and receiver (destination) of the packet. Data travels along the network from computer to computer until it reaches its destination. Each computer in the network determines the best route forward for the packets it receives and must transmit. Computers that make these decisions are known as routers. The destination computer reassembles the packets into the original message.

There are a number of advantages to packet-switching networks for data communications. Long streams of data can be broken up into small, manageable chunks. This means that the packets can be distributed efficiently to balance the traffic across a wide range of possible transmission paths in a network.

43.4 TCP/IP

One of the key objectives of most computer networks is to achieve high levels of connectivity. Connectivity is the ability of computer systems to communicate with each other and to share data. For connectivity, standards must be defined to enable communication between sender and receiver. Such standards are embodied in communication software.

One approach to developing higher connectivity among systems is by using the idea of open systems. Open systems are built on public domain operating systems, user interfaces, application standards and networking standards. One of the oldest examples of an open systems model for communications is TCP/IP. This was developed by the US Department of Defense in 1972. The Transmission Control Protocol/Internet Protocol is the communications software model underlying the Internet. A protocol is a statement that explains how a specific networking task should be performed. TCP/IP divides the communication process into five layers of network tasks:

- *Application*. The application layer is that closest to the network user. The application layer provides data entry and presentation functionality to the end-user of the network
- *Transport/TCP*. This layer breaks application data up into TCP packets known as datagrams. Each packet consists of a header comprising the address of the sending computer, data for reassembling the original data and error-checking data
- *Internet protocol*. This layer receives datagrams from the TCP layer and breaks the packets down further. An IP packet contains a header with an address and carries TCP information and data in the body of the packet. The IP layer routes the individual packets from the sender to the receiver

- *Network*. This handles addressing issues, usually within the operating system, as well as providing an interface between the computer and the network. Each device on a network will normally have a unique ID (an IP number) assigned to it – represented in the network interface of each device
- *Physical*. This defines the basic characteristics of signal transmission along communication networks

Two different computer systems using TCP/IP are able to communicate with each other even though they may be based on different hardware and software platforms. Data sent from one computer passes down through the five layers. Once the data reaches the receiving computer it travels up through the layers. If the receiving computer finds a damaged packet it requests the sending computer to send again. This process is illustrated in Figure 43.1.

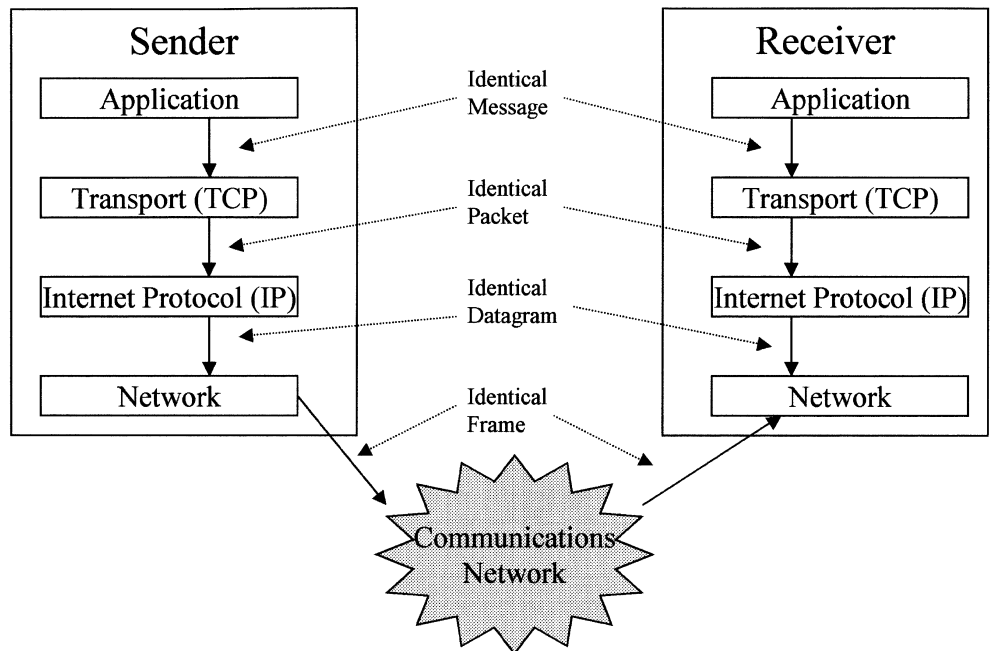


Figure 43.1 TCP/IP layers.

43.5  HYPERTEXT TRANSFER PROTOCOL (HTTP)

HTTP is an object-oriented protocol that defines how information can be transmitted between clients and servers. A HTTP transaction consists of the following phases:

- *Connection*. The client establishes a connection with a Web server
- *Request*. The client sends a request message to a Web server


- *Response.* The Web server sends a response to the client
- *Close.* The connection is closed by the Web server

HTTP is said to be a stateless protocol. This means that when a server provides a response and the connection is closed, the server has no memory of any previous transactions. This has the advantage of simplicity in that clients and servers can run with simple logic and there is little need for extra memory.

43.6 IP ADDRESSES

An IP address is the fundamental way of identifying uniquely a computer system on the Internet. An IP address is constructed as a series of up to four numbers, each delimited by a period. It is hence called a dotted quad. In a 32-bit IP address, each of the four numbers can range from 0 to 255. Generally the first of the four numbers identifies a computer network. The remaining numbers usually identify a node on this network.

Because of the explosion in Internet usage, more computers are being added to the global network. Hence, the 32-bit IP address will eventually run out of unique addresses. To offset this, a 128-bit IP address will be introduced globally.

Example  126.203.97.54 may be an IP address for a computer on the local area network at my university.

43.7 UNIVERSAL RESOURCE LOCATORS (URLs)


Internet users generally find IP addresses difficult to remember. Hence more mnemonic identifiers have been introduced which map to IP addresses.

A computer attached to the Internet and the HTML (hypertext markup language) documents resident on these computers are identified by universal resource locators (URLs). URLs can thus be used to provide a unique address for each document on the Web. Links between documents are activated by 'hotspots' in the document: a word, phrase or image used to reference a link to another document.

The syntax of a URL consists of at least two and as many as four parts. A simple two-part URL consists of:

- The protocol used for the connection (such as HTTP)
- The address at which a resource may be located on the host



Example  In the URL below, the protocol – HTTP – is placed before the symbols `://`. The address after these symbols identifies a specific Web page on the host computer, in this case the home page of the University of Wales, Swansea:

`HTTP://www.swansea.ac.uk`

43.8 DOMAIN NAMES

The ‘swansea’ in the URL in the previous example is short for ‘The University of Wales, Swansea’, the ‘ac’ for academic and the ‘uk’ for United Kingdom. This constitutes a so-called domain name, an agreed string of characters that may be used to provide some greater meaning to a URL. In practice, a domain name identifies and locates a host computer or service on the Internet. It often relates to the name of a business, organisation or service and must be registered in the same way as a company name.

A domain name is actually made up of three parts:

- *Subdomain*. This constitutes a provider of an Internet service. In this case it is University of Wales, Swansea
- *Domain type*. This suggests the type of provider. In this case it is ‘ac’ – indicating an academic institution based in the UK. The string ‘edu’ (short for education) is used more generally for an educational institution internationally
- *Country code*. Every country has its own specific code. For instance, ‘au’ is the code for Australia. If no country code is specified then the organisation is more than likely based in the USA

43.9 COMPONENTS OF THE WWW

The World-Wide-Web, or Web for short (Berners-Lee, 1999) is the application (or series of applications) which is associated most readily with the Internet at the current time.

Figure 43.2 illustrates the primary components of the Web:

- Hypertext/Hypermedia
- HTML
- Web browsers

43.10 HYPERTEXT/HYPERMEDIA

Vannevar Bush (1945) envisaged hypertext/hypermedia systems in the 1940s. In a ground-breaking paper Bush discussed the concept of a memex (memory

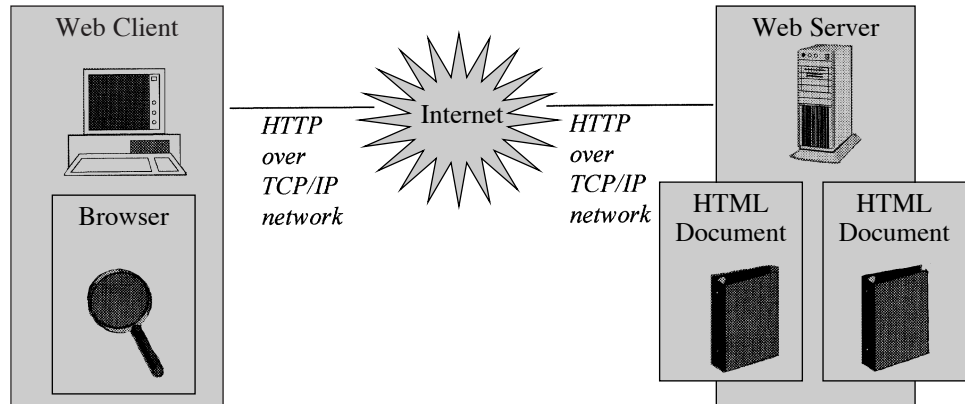


Figure 43.2 Components of the World-Wide-Web.

extender), a device capable of storing and retrieving information on the basis of content. In 1968, Douglas Englebart demonstrated the Augment system. Augment was an on-line working environment designed to augment the human intellect. It could be used to store and retrieve memos, research notes and other forms of documentation. Ted Nelson extended Bush's original idea in his Xanadu environment. Xanadu was designed to be an ever-expanding work-space that could be used to create and interconnect documents containing text, video, audio and graphics. Nelson actually coined the term hypertext and described it as being 'non-linear reading or writing.' A number of prominent hypermedia prototypes were developed during the 1980s. However, a software tool bundled with the Apple Macintosh computer did most to popularise the concept. In recent times hypertext and hypermedia form the bedrock for the Web.

Text can normally be organised in three major ways:

- *Linear text.* Linear text is exemplified in the format of the conventional novel. The reader is expected to start at the beginning and progress to the conclusion via a sequential reading of chapters
- *Hierarchical text.* Most textbooks and reports are organised hierarchically in terms of chapters, sections, subsections etc. The reader is able to access the text at various points in the hierarchy
- *Network text.* The dictionary or the encyclopaedia exemplifies network text. In a dictionary, each entry has an independent existence but is linked to a number of other entries via references

Hypertext is an electronic or on-line version of network text. A hypertext document is made up of a number of textual chunks connected together with associative links called hyperlinks. Hypermedia is a superset of hypertext. Here the nodes of the network may be various media, including text, graphics, audio and video.

**43.11**  **HYPertext MARKUP LANGUAGE (HTML)**

The WWW can be thought of as a collection of documents residing on thousands of servers or Web-sites around the world. Each such document contains content such as text and a set of embedded tags that indicate how the content is to be presented. This process of tagging text with extra presentational information is known as marking-up, and the set of tags for doing this a markup language. In the 1960s work began on developing a generalised markup language for describing the formatting of electronic documents. This work became established in a standard known as the standard generalised markup language, or SGML.

SGML in fact constitutes a meta-language – a language for defining other languages. Hence, SGML can be used to define a large set of markup languages. Tim Berners-Lee used SGML to define a specific language for hypertext documents known as hypertext markup language (HTML). HTML is a standard for marking-up or tagging documents that can be published on the Web, and can be made up of text, graphics, images, audio-clips and video-clips. Documents also include links to other documents either stored on the local HTML server or on remote HTML servers. HTML documents are also referred to as 'pages'.

Example  Below we include a very simple document expressed in HTML:

```
<HTML>
<TITLE>Information Systems</TITLE>
<H1>Information Systems</H1>
<H2>Paul Beynon-Davies</H2>
</HTML>
```

The text between angled brackets constitutes tags. Each piece of text is preceded by a start-tag and succeeded by an end-tag. A forward slash precedes an end-tag. The *HTML* tags indicate the start and end of the document. The tag *Title* provides a name for the page. The tags *H1* and *H2* indicate that a first-level and second-level heading should be displayed respectively.


A HTML document contains both content and tags. The document content consists of what is displayed on the computer screen. The tags constitute codes that tell the browser how to format and present the content on the screen.

The general form of this relationship between tags and content is

```
<tagname properties> content </tagname>
```

The tagname is taken from an established set of keywords established in the particular version of HTML. Certain tagnames have associated with them a

number of properties which serve to refine the meaning of a tag to a browser.

Examples  In the tag `<P align="right">`, *P* is the tagname and acts as an abbreviation for the word paragraph. Consequently this tag is designed to be placed at the start of a chunk or paragraph of text. The word *align* is a property which can be assigned a number of values from a limited list. Here the value *right* specifies that the paragraph in question should be right-justified on the screen. An end-tag `</P>` will be placed at the end of the chunk of text.

The hyperlinks between pieces of text are established using anchor tags. The link can be to a textual element in the same document or to another document. This anchor tag has the form:

```
<A HREF="address">visible link text</A>
```

The letter A stands for anchor. HREF is a property which is used to specify the address of the document or piece of text to be linked to. The visible link text establishes what is displayed on the screen as a so-called hotspot. When you move the cursor over a hotspot, the cursor changes from an arrow to a pointing hand. This indicates that a click on the hotspot will transfer you to the text specified in the address.

The following tag embedded in a HTML document will establish a hyperlink to the author's current place of work:

```
<A HREF="http://www.swan.ac.uk">The University of Wales, Swansea</A>
```

43.12 WEB BROWSER

To access the Web one needs a browser. This is essentially a program that lets the user read Web documents, view any in-built images or activate other media and hotspots. After the invention of the concept of the Web, the idea became established quickly in the scientific community. However, few people outside this community had software capable of reading HTML documents. In 1993 the first program that could read HTML documents and display them on a graphical user interface was written at the University of Illinois. It became known as Mosaic.

The commercial opportunities afforded by browsers soon became apparent and members of the Illinois team joined with one other to form the company Netscape Communications. Their key product, Netscape Navigator, became an immediate success. Microsoft soon entered the market with its Internet Explorer product and these two browsers still dominate this niche in the software market.



43.13 THE PLACE OF DATABASE SYSTEMS IN INTERNET APPLICATIONS

Many contemporary Web-sites are repositories for files. In this case each Web document is stored in its own file. This is perfectly satisfactory for small Web-sites. As the amount of information provided on a Web-site grows, this file-based model proves difficult to manage.

For example, it is difficult to keep track of information contained in hundreds or thousands of separate Web documents. If changes are made to the structure of information contained in documents, such changes may affect hundreds of links from other documents.

Many organisations are now looking to Web-sites to make connections to their customers and suppliers. In terms of customers, the organisation may wish to provide a complete list of its products. In terms of suppliers, the organisation may wish to provide access to its production data. To support such applications, connections need to be made from Web servers to data contained in corporate databases.

The use of database systems to store both content and link information of relevance to Web information has therefore become a significant force over the last few years. We generally refer to the use of databases in this context as networked database applications.

43.14 TYPES OF WEB-ENABLED DATABASE APPLICATIONS

A Web-enabled database application is a database application in which clients use a HTML-based browser and the server includes both an Internet server, a DBMS and a database. Several different types of such applications are emerging. Many of these are referred to as publishing applications because of the way in which Internet applications are organised around requests to an information source. We can distinguish between three main ways in which a database system may interact with Web-based services:

43.14.1 STATIC REPORT PUBLISHING

In this type of Web-enabled database application the DBMS generates a report, static form (a display only form), or response to a query in HTML format, and posts this onto the Web-site. The traffic in this type of application is one-way, from the server to the client. The user provides no data to the application.

43.14.2 QUERY PUBLISHING

In this type of Web-enabled database application a HTML form is generated containing text boxes in which the user may enter criteria for a query. Once

the form is submitted, a request is made to the DBMS which returns matching data or an error.

43.14.3 APPLICATION PUBLISHING

In this type of Web-enabled database application the interfaces (both data entry and reports) are all Web based. This is clearly a Web-based emulation of the traditional information technology system architecture. However, such applications currently suffer from three major problems:

- Internet applications usually suffer from low data transmission rates. This means that only applications having limited demands on data transmission can generally be shared
- Internet security currently suffers from limited security. Developments are occurring in areas like encryption but as yet many organisations are unhappy in using Internet technology to handle commercially sensitive data
- The stateless nature of Internet transactions means that the client needs to maintain some record of the status of a database session. This means that client applications generally have to be large and quite complex

Each of the networked database applications described above demand the use of dynamic Web pages. Traditionally, a HTML file stored as a document is an example of a static Web page. The content of the page only changes if a change is made to the document. In a dynamic Web page, the content is dynamically generated each time the page is accessed.

43.15 APPROACHES TO BUILDING WEB-ENABLED DATABASE APPLICATIONS

There are a number of approaches to building Web-enabled database applications including:

- *Common gateway interface (CGI)*. This constitutes a specification for how scripts communicate with Web servers. Scripts are programs stored on the Web server which are executed by the server, and the results of the execution are returned to the browser. Parameters can be passed to the script using the query string part of a URL
- *Server-side includes*. Normally Web servers do not examine the files that they send to browsers. Some servers are able to examine the files for so-called server-side includes. Such includes may command the server to execute some program and include the results within the document before returning it to the browser
- *JDBC, JSQL*. Java can be used in association with a number of defined application programming interfaces (APIs). For instance, Java Database

Connectivity (JDBC) is modelled on the Open Database Connectivity (ODBC) API. JSQL is an extension to the embedded SQL standard proposed for use with Java

- *Scripting languages*, such as Javascript and Vbscript. Scripting languages enable the specification of functions within HTML documents. Javascript is an object-based scripting language originally developed by Netscape and Sun. Vbscript is a Microsoft scripting language whose functionality is similar to Javascript but which more closely resembles Microsoft's Visual Basic in terms of syntax
- *Vendor initiatives*, such as Microsoft's Active Server Pages. Active Server Pages (ASP) is a Microsoft-specific model for building Web pages on Web servers. An ASP can contain a combination of text, HTML tags, and script commands and output expressions. ASP files are requested using the '.asp' extension from the server. The Web server reads through the requested file, executes any commands and returns the dynamic HTML page to the browser

43.16  CASE STUDY: USING CGI

As indicated above, a Web server is a program running on a server machine that accepts requests from a Web browser and returns the results in the form of HTML documents. The browser and server communicate using the HTTP protocol. HTTP provides a number of features over and above the simple transfer of documents. One of the most important of these is the ability to execute programs with arguments supplied by the user, and deliver the results back in the form of HTML documents. When a Web server recognises a URL as pointing to a file, it returns the contents of the file. In contrast, when a URL points to a program, it executes the script and returns the output of the program back to the browser as a document.

This enables a Web server to act as an intermediary in an *N*-tier architecture (Chapter 36) between Web browser and application programs. The common gateway interface (CGI) standard defines how a Web server may communicate with application programs. In turn, the application programs will communicate with a database server through ODBC (Open Database Connectivity), JDBC or other protocols.

Consider the case of a customer wishing to order an item on an e-Commerce site. The customer navigates through the Web-site to an on-line order form. After completing the form, the customer clicks on a submit button. This initiates a request to the Web server which it interprets as a request to run an application program. This program (or programs) is likely to conduct a series of actions such as:

- Checking that the data on the order is correct
- Initiating a request to a database to check stock
- Initiating a request to another database to check customer details

Depending on the result of each of these actions, various responses are returned to the browser. For instance, if some data is missing, an error message is likely to be returned and the customer requested to correct the details.

43.17 SUMMARY

- The Internet is a set of interconnected computer networks distributed around the globe
- The infrastructure of the Internet is built upon packet-switched networks, TCP/IP, HTTP, IP addresses, URLs and domain names
- The facilities provided by the World-Wide-Web are the ones most readily associated with the concept of the Internet at the current time
- The primary components of the Web comprise: hypertext transfer protocol (HTTP), hypertext markup language (HTML), universal resource locators and browsers
- Database systems are important in the context of managing large-scale Web applications through the use of dynamic pages
- Three types of applications for the use of database systems within Web applications are report publishing, query publishing and application publishing
- Different approaches for building networked database applications include: CGI, JDBC and Active Server Pages

43.18 ACTIVITIES

1. Investigate the process by which domain names are assigned globally.
2. Investigate some of the other common tags used in a HTML document.
3. Identify three applications in a university setting corresponding to the three types of application for database systems within Web applications.
4. Determine the most popular approach for connecting Web servers to database servers.

43.19 REFERENCES

- Berners-Lee, T. (1999). *Weaving the Web: The Past, Present and Future of the World Wide Web by its Inventor*. London, Orion Business Publishing.
- Bush, V. (1945). As we may think. *Atlantic Monthly* 176: 101–3.