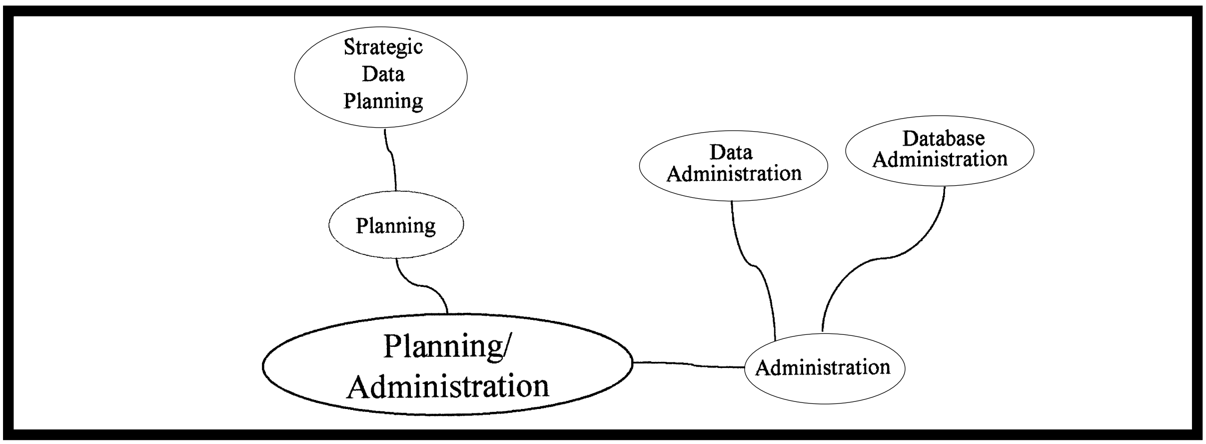
PART 5

PLANNING AND ADMINISTRATION OF DATABASE SYSTEMS

Our memories are card indexes consulted, and then put back in disorder by authorities whom we do not control.

Cyril Connolly (1903–74)



Database systems have become so important to modern organisations that significant activity is now devoted to planning for, monitoring and administering such systems. In this part we discuss three important planning and managerial activities relevant to database systems: strategic data planning, data administration and database administration.

- *Strategic data planning* is the process of developing corporate data models and using such models for the planning and management of application development (Chapter 21)
- *Data administration* is that function concerned with the management, planning and documentation of the data resource of some organisation (Chapter 22)
- *Database administration* involves the technical implementation of database systems, managing the database systems currently in use, and setting and enforcing policies for their use (Chapter 23)

CHAPTER 21

STRATEGIC DATA PLANNING

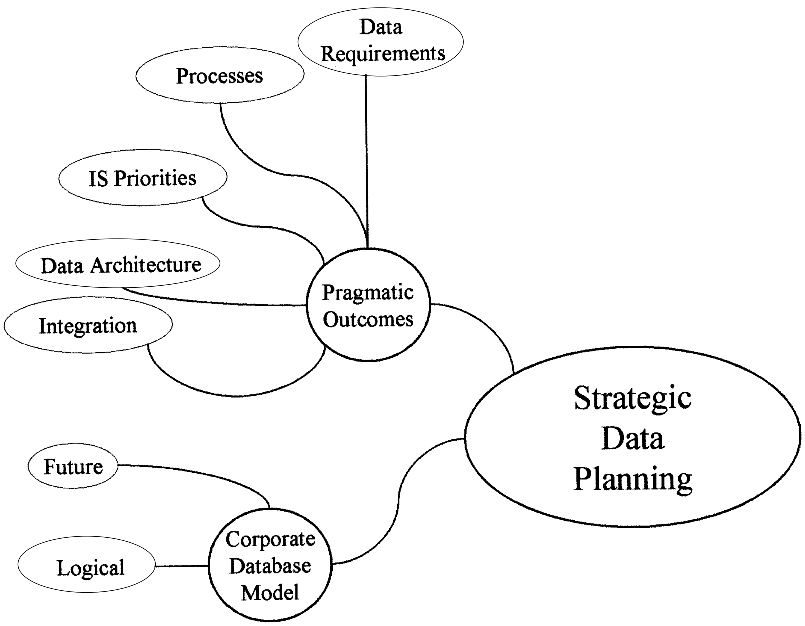
When we are planning for posterity, we ought to remember that virtue is not hereditary.

Thomas Paine (1737–1809)

LEARNING OUTCOMES

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

- Distinguish between corporate data modelling and application data modelling
- Discuss the pragmatics of strategic data planning



21.1 INTRODUCTION

To review, database development, the process of representing some universe of discourse (UoD) as a database system, can generally be said to consist of the following phases: requirements elicitation, conceptual modelling, logical modelling and physical modelling. Conceptual modelling is normally further divided into two substages: view modelling, a process that transforms user requirements into a conceptual model; and view integration, a process that combines individual views into a global conceptual model.

It thus seems to have become established practice that the appropriate way to develop database systems is to engage in a prior exercise in application data modelling. Data modelling is the process of developing an implementation-independent map of database requirements. Most data modelling is therefore conducted to aid in the development of some application system. In this sense, the term data model is used to describe a blueprint for the data requirements of a particular application.

Having said this, Hudson (1991) usefully classifies data models on two dimensions: a logical–physical dimension and a current–future dimension. Most application data models attempt to document data structures to support new systems – future data models. Data models can also be used to document existing 'legacy' systems – current data models. Logical models are designed to support the organisational view of data; physical models are designed to support the technical view of data (Figure 21.1). The activity devoted to the construction of future/logical data models is generally referred to as strategic data planning. This activity is the topic of the current chapter.

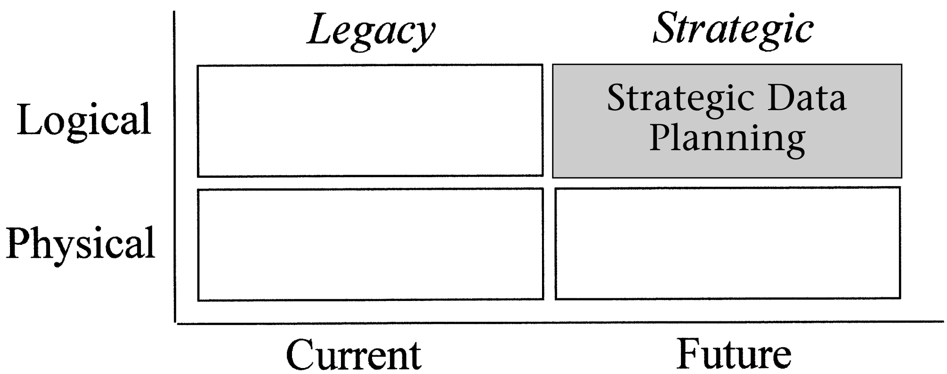


Figure 21.1 Data model types.

21.2 CORPORATE DATA MODELLING

Prefixing the term data modelling with the word corporate would tend to suggest an elevation of this practice to the level of the whole organisation. A corporate data model (CDM), using this perspective, should form a map of the data requirements of the whole or a substantial part of an organisation. This differs from an application data model which is produced to support a specific information systems project (Beynon-Davies, 1997).

Perhaps part of the reason for the scarcity of material on the topic of CDM may be because of a commonly held assumption that developing a corporate data model is no different an activity from that of developing an applications data model. As we shall see, however, whereas the techniques applied may be similar, the specific problems involved in corporate data modelling relate to the much higher level of abstraction demanded by this activity as compared to developing an applications data model.

The idea of a CDM as a model of large scope is clearly the emphasis in methodologies such as Information Engineering, in which corporate data modelling forms a key activity within an information strategy planning phase of development (Braithwaite, 1992). The idea of CDM in association with modelling corporate processes also seems to underlie the idea of enterprise modelling and business objects. CDM is also significant in the idea of information (resource) management and the discipline of data administration which has been floated since the early 1980s.

However, there appear to be some differences in the way in which particular organisations construct such a map. One approach is to build what amounts to a map of very large scale. On such a map only the key features of the data terrain are delineated. The intention of such a map seems to be to act as an aid in controlling the systems development process in such organisations. This we shall call a type 1 CDM. Another approach is to build a map to a smaller scale but in a schematic manner. A minimum data-set is provided for key organisational functions. The key rationale for this type of corporate data model seems to be to act as a means of standardising data capture and usage across what may be an ill-structured environment in information terms. This we shall call a type 2 CDM.

21.3 EXISTING LITERATURE ON CDM

Hudson (1991) briefly describes an attempt to develop both current and future corporate database models at Sara Lee Corp, a large clothing and food consortium based in Chicago, USA. She particularly documents the importance of CASE technology to her CDM project, permitting the analysis of some 135 entities in three months. She states that ‘a future logical corporate data model can play a significant role in effectively guiding long-range applications development and information use.’ She adds however that the creation of such a model is a labour-intensive process that needs the support of CASE tools.

Sussman (1993) documents a Canadian case study in CDM – the process of building a CDM for the municipal activities of the city of Scarborough, Ontario. Although the paper is particularly concerned with the problem of integrating spatial data for Geographical Information Systems work into a

CDM (Chapter 39), Sussman makes some interesting comments about the process of building a CDM in general. The Scarborough CDM seems to have been developed as part of a larger information systems planning effort. Part of the initial analysis found that at least 125 data-sets were employed in the municipality, which, in total, required 28 man years per annum to maintain. In this context, the main aim of the CDM was to integrate diverse data-sets, and as a consequence produce a municipal data model independent of activities, computer technology or the organisational structures of departments (the idea that data is 'owned' by departments). Sussman sees a CDM as encouraging the perception of data as a city-wide resource that helps promote the sharing of data. The key use of the CDM seems to be in the planning of application development.

The UK government's centre for information systems, the CCTA, published a handbook on corporate database modelling in the mid-1990s (CCTA, 1994). It defines a corporate database model as being in very broad terms: 'a set of definitions for things of significance to the organisation, about which information needs to be held, and the relationships between them.' It describes a corporate data model as being a progression through three levels of detail:

- *Strategic tier* – a model of major business concepts
- *Architectural tier* – more detailed model including additional attributes and volume information
- *Application tier* – an aggregate of more detailed models from which applications are built

The CCTA sees the main reason for building a CDM to be to harmonise the data requirements across a number of diverse application systems. The idea is to increase the integration and interoperability of information systems through data sharing.

The CCTA maintains that corporate data modelling should be accomplished by a small team of analysts and identifies two critical success factors for corporate data modelling:

- High level of support within the organisation for what must be seen to be a valuable activity
- Strong end-user involvement with the modelling

Because of the importance of these factors it suggests that corporate data modelling should first be conducted as a pilot study on a subject area that has significant potential for data sharing. It also sees it as extremely important that, because of their overarching scope, the management and control of a CDM should be outside the framework of normal application system-development projects. Some formal structure should be set in place for delimiting ownership, responsibility and change control. A meta-model has been constructed by the CCTA which serves as a framework for determining the minimal component parts of a CDM.

21.4 CASE STUDY: STRATEGIC DATA PLANNING IN THE NATIONAL HEALTH SERVICE

The UK National Health Service (NHS) is one example of an organisation that has engaged in a number of strategic data planning initiatives. In this section we present a brief review of some of these initiatives and come to some conclusions as to the pragmatics of these exercises in the NHS (Beynon-Davies, 1994).

The idea of corporate data modelling in the NHS dates back to the late 1970s. In 1980, a body known as the NHS/DHSS (Department of Health and Social Security) Steering Group on Health Services Information was set up under the chairmanship of Mrs Edith Korner. The remit of this body was to identify a minimum data-set which could be used routinely for management purposes in every district health authority (DHA). The eventual aim was to produce management information from data usefully collected at the operational level. A number of working groups were set up, each with a substantial membership of NHS personnel. Each group was given the job of building a data model for a specific area of the health service: hospital clinical activity, patient transport services, health services manpower, activity in hospitals and the community, services in the community and health services finance.

All the working groups conducted their data modelling in a similar fashion:

- The data requirements of a DHA and its management team were identified
- The additional data needs of regional authorities and central government departments were identified
- Definitions for the data-items were agreed
- Initial recommendations were published and field-tested in four DHAs
- Consultation was encouraged through formal publications and informal seminars
- Recommendations were finalised, likely resource consequences were discussed and a timetable for implementation was generated

Over the following five years the Steering Group published six reports and one supplement which collectively became incorporated within a document known as the Korner Data Model Report (KDMR).

The KDMR contained entity–relationship (E–R) diagrams of each of the areas covered by the reports, together with detailed definitions of entities and attributes. Various changes to the KDMR have occurred since 1985, and new data-sets have been added. In 1989, the post-Korner data model was published by a newly formed body known as the NHS Information Management Centre, and renamed the Minimum Data Set Model (MDSM).

The purpose of the MDSM was 'to define for health authorities those data which they have all agreed to collect consistently. The model provided a tool for assessing ready-made systems, for creating new ones and for evaluating the impact of proposals for change.'

The MDSM documentation constituted two large volumes. Volume 1 included fifty or so detailed E–R diagrams divided into the following groups: general, hospital, community, manpower, finance, patient transport and estates. Associated with each uniquely coded diagram there was an overview which primarily described the attributes associated with each entity on a diagram. A brief activity analysis was also included against each entity model, detailing how some of the standard reports of the health service may be produced.

Volume 2 included a series of organised entity and attribute lists. This volume effectively constituted a data dictionary made up of the following components:

- An alphabetic entity list of some 500 entities in which each entity was referenced against its occurrence in each of the entity models
- An alphabetic attribute list of some 1,000 attributes in which attributes were referenced against their occurrence in entities
- An alphabetic entity list in which detailed textual definitions were given for each entity
- An alphabetic attribute list in which detailed formats were provided for each attribute

In 1991 version 2 of the MDSM was released. Whereas version 1 of the model constituted the minimum data-set required to be captured by a district health authority (a structural unit within the NHS), version 2 constituted the minimum data required to be captured by a health care provider, including the data to be passed on to purchasing organisations. These changes were clearly directed at the creation of an internal market within the NHS. Within a year of its publication the MDSM was described as being absorbed into, and superseded by, the NHS data dictionary. The data dictionary was described as recording the results of agreements and draft agreements about sets of data. It is meant to include data requirements from many more business areas than those included in the MDSM.

An E–R diagram simplified from one actually produced as part of the MDSM is illustrated in Figure 21.2.

An extract from the associated data dictionary entry is illustrated in Figure 21.3.

21.5 THE PRAGMATICS OF STRATEGIC DATA PLANNING

Goodhue *et al.* (1992a) cite previous studies as questioning the value of strategic data planning (SDP) efforts, despite strong conceptual grounds for its use. They conducted an empirical study of strategic data planning in nine US companies. The authors identify five potential outcomes of SDP exercises from the literature:

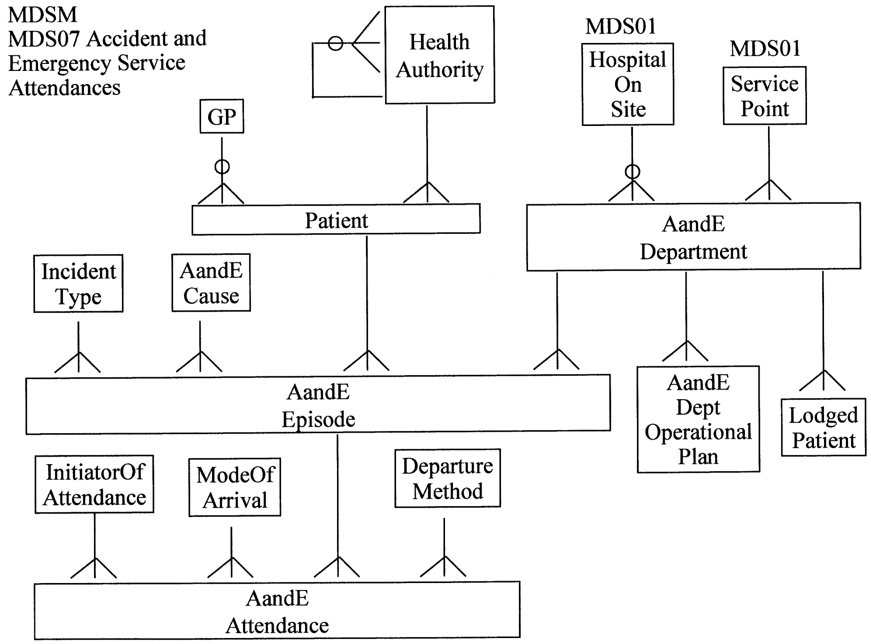


Figure 21.2 A data model from MDSM.

AandEAttendance

(Accident and Emergency Attendance)

K–Primary Key
F–Foreign Key

Health Authority Code
Service Point Code
District Patient Number
Arrival Date
Arrival Time
First Attendance
Initiator of Attendance
F INITIATOR OF ATTENDANCE
Method of Departure
F DEPARTURE METHOD
Mode of Arrival
F MODE OF ARRIVAL
Start Date
F AandE Episode

Figure 21.3 A dictionary entry from MDSM.

- The implementation of integrated information systems
- The development of a corporate-wide data architecture
- The clear identification of information systems priorities
- The rethinking of an organisation's key processes
- The education and communication of data requirements throughout the organisation

Example The strategic data planning conducted by the NHS is a good example of an organisation attempting to meet many of these objectives. The development of the MDSM was an initial attempt to standardise on an organisation-wide data architecture. The primary aim seems to have been to disseminate common data requirements throughout the organisation with the aim of at least unifying the processes of data capture throughout the NHS. As longer-term aims the MDSM was intended to facilitate better integration of information systems and to at least influence investment in information systems technology in the NHS:

Goodhue *et al.* (1992a) derive a number of lessons from their study of SDP efforts in the USA.

21.5.1 THE IMPLEMENTATION OF INTEGRATED INFORMATION SYSTEMS

Strategic data planning should be treated as a process of design as well as planning. For an SDP effort to be effective, data integration must be critical to strategic goals of the organisation as perceived by top management. Also, efforts to implement data integration need to balance global integration and local flexibility.

21.5.2 THE DEVELOPMENT OF A CORPORATE-WIDE DATA ARCHITECTURE

Goodhue *et al.* (1992a) come to the conclusion that it is not clear what the most appropriate form for a data architecture is. It is also not clear that an SDP is the most effective means to produce such architectures. An SDP project of too large a scope spells trouble. Also, to have an effect on data integration, architectures must be enforced and should be 'stolen', not invented.

21.5.3 THE CLEAR IDENTIFICATION OF INFORMATION SYSTEMS PRIORITIES

For a systems priorities goal, SDP does not narrow its scope fast enough before it begins the time-consuming process of modelling business functions and entities. In such a situation, creativity is swamped by the volume of detail.

21.5.4 THE EDUCATION AND COMMUNICATION OF DATA REQUIREMENTS THROUGHOUT THE ORGANISATION

In the projects Goodhue *et al*. (1992a) studied, new understanding seems difficult to communicate. They conclude that the cost of an SDP can probably not be justified by education and communication alone.

In a related research project, Goodhue *et al*. (1992b) considered the issue of data integration. Data integration is defined as the standardisation of data definitions and structures through the use of a common conceptual schema across a collection of data sources. This bears a resemblance to the type 2 CDM discussed in section 21.2. Data integration is implicitly assumed always to result in net benefits to the organisation. The authors question this view by using organisational information processing theory to construct a model of the costs as well as the benefits of data integration. This model is based on the central importance of mechanisms for handling uncertainty to organisations. Data integration is clearly one such mechanism for reducing uncertainty. However, a distinction is made between three major types of uncertainty:

- Uncertainty resulting from complex or non-routine subunit tasks
- Uncertainty resulting from unstable subunit task environments
- Uncertainty resulting from interdependence among subunits

They conclude that data integration can only satisfactorily address the last type of uncertainty. A distinction is also made between uncertainty and equivocality. Uncertainty is the absence of specific, needed information. Equivocality means that there are multiple, conflicting interpretations of a situation. Data integration can address aspects of uncertainty but not equivocality. Uncertainty may be reduced by a certain amount of information, equivocality can only be reduced by sufficiently rich information.

On the basis of these distinctions, the authors propose that the importance of data integration on the costs and benefits of information systems are mediated through three factors:

- *Increased ability to share information to address subunit interdependence*. All other things being equal, as the interdependence between subunits increases, the benefits of data integration will increase, and the amount of data integration in rational organisations should also increase
- *Reduced ability to meet unique subunit information requirements*. All other things being equal, as the differentiation between subunits increases, data integration will impose more and more compromise costs on local units; therefore the amount of data integration in rational firms should decrease. Also, all other things being equal, firms with increased data integration will experience greater bureaucratic delay in getting approval for changes affecting the data models used by individual subunits
- *Changes in the costs of information systems design and implementation*. All other things being equal, as the number and heterogeneity of subunit information

needs increase, the difficulty of arriving at acceptable design compromises increases, and the cost of the resulting design will increase more than linearly. Thus, rational firms will integrate less extensively when there are many heterogeneous subunits involved. As organisations face greater instability in their environments and their information requirements, the importance of this proposition will increase. In turbulent environments, firms with many heterogeneous subunits will be even less likely to integrate extensively, and firms with homogeneous subunits will be more likely to integrate extensively.

21.6 SUMMARY

- Strategic data planning is the activity devoted to the construction of future/logical data models
- A corporate data model is a map of the data requirements of the whole or a substantial part of an organisation
- There are two main types of corporate data model: a type 1 CDM maps key features of the data terrain; a type 2 CDM provides a minimum data-set for key organisational functions
- Organisations generally conduct strategic data planning with one or more of the following objectives in mind: the implementation of integrated information systems; the development of a corporate-wide data architecture; the clear identification of information systems priorities; the rethinking of an organisation's key processes; and the education and communication of data requirements throughout the organisation
- Strategic data planning may be relavant only for certain forms of organisation. Generally speaking those organisations with homogeneous and interdependent subunits will benefit most from the data integration arising from strategic data planning

21.7 ACTIVITIES

1. Examine an organisation known to you, such as a university. Determine whether it has a corporate data model. If it does, is it a type 1 or type 2 corporate data model?
2. If the organisation does not have a corporate data model, attempt to determine three or four of the most important entities to be included in such a model.
3. In terms of some organisation known to you, such as a university, use the data integration model produced by Goodhue *et al*. (1992b) to determine the importance of strategic data planning for the organisation.

21.8 REFERENCES

Beynon-Davies, P. (1994). Information management in the British National Health Service: the pragmatics of strategic data planning. *International Journal of Information Management* **14**(2 (April)), 84–94.

Beynon-Davies, P. (1997). The corporate data model: a study of organisational practice. *Journal of Systems and Information Technology* **1**(1 (March)), 47–63.

Braithwaite, K.S. (1992). *Information Engineering: Analysis and Administration*. London, CRC Press.

CCTA (1994). *Corporate Data Modelling*. London, HMSO.

Goodhue, D.L., L.J. Kirsch, J.A. Quilliard and M.D. Wybo (1992a). Strategic data planning: lessons from the field. *MIS Quarterly* March: 11–34.

Goodhue, D.L., M.D. Wybo and L.J. Kirsch (1992b). The impact of data integration on the costs and benefits of information systems. *MIS Quarterly* September: 293–311.

Hudson, D.L. (1991). Approaches to corporate data model development at Sara Lee Corp. *Data Resource Management* **2**(3): 49–55.

Sussman, R. (1993). Municipal GIS and the enterprise data model. *Int. Journal of Geographical Information Systems* **7**(4): 367–77.

CHAPTER 22

DATA ADMINISTRATION

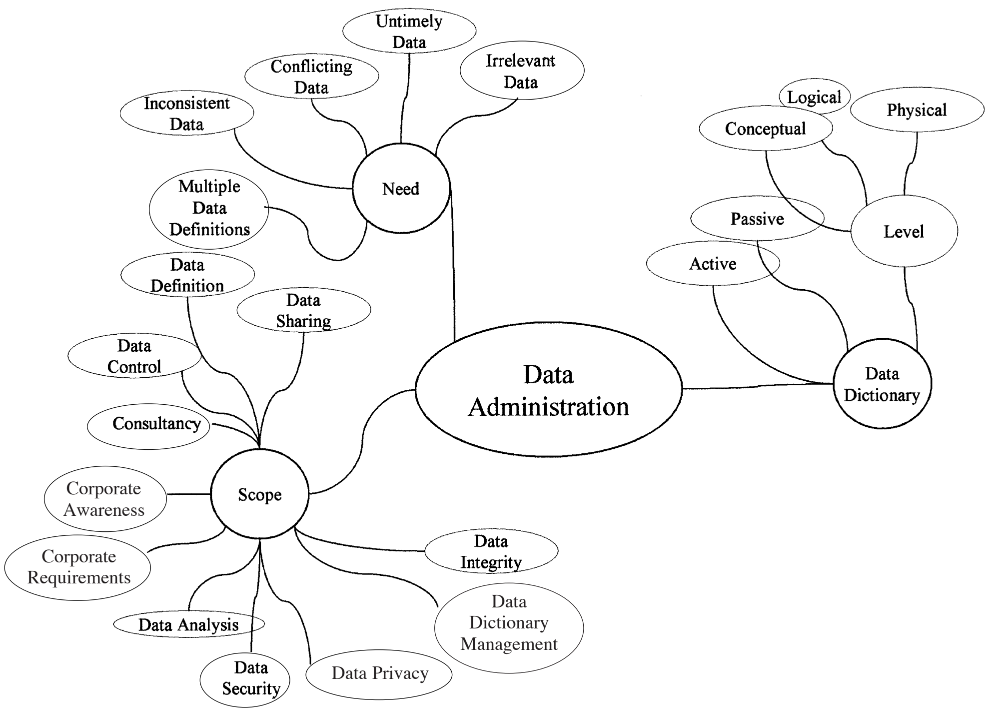
Tis hard to fight with anger, but the prudent man keeps it under control.

Democritus (460–370 BC)

LEARNING OUTCOMES

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

- Define the concept of data administration
- Discuss the scope of the data administration function
- Relate the costs and benefits of having a data administration function
- Define the concept of a data dictionary
- Consider the issue of database security



22.1 INTRODUCTION

Database technology, because of its centrality in the modern informatics infrastructure, has stimulated the development of a large range of roles for servicing this technology. Two of the most commonplace roles are the data administrator and the database administrator.

Data administration is that function concerned with the management, planning and documentation of the data resource of some organisation (Gillenson, 1987). Data administration is concerned with the management of an organisation's meta-data, that is, data about data. It is a function which deals with the conceptual or business view of an organisation's data resource.

The key concept in the move towards data administration has been that data, like capital, personnel etc., should be treated as a manageable resource. In other words, data is a critical commodity in an organisation's attempt to compete in the open marketplace. In this sense, the data administration function is seen as a key part of the data and information management strategy of organisations (Beynon-Davies, 2002). The data administrator is also likely to be a key person in any technology concerned with data management such as data warehousing (Chapter 40).

Example Why is data such a critical resource for organisations? Consider the case of a university. Without data such as what students it has, what students are taking which modules and what grades have been achieved by students, a university is unable to operate effectively. However, consider the case in which:

- Different university departments or schools maintain their own distinctive collections of data with their own distinctive definitions for data-items
- Data is frequently missing or incomplete
- Data is frequently out-of-date
- There is incomplete knowledge among staff as to what data is collected, and where it is kept

In such situations staff may spend a substantial amount of its time resolving data problems. Such situations demonstrate the key need for an organisational function tasked with managing the data resource.

22.2 THE SCOPE OF DATA ADMINISTRATION

The scope of the data administration function will vary according to the organisation. There are however a core set of activities for which a data administration function is typically responsible:

- *Consultancy* – offering consultancy on all aspects related to an organisation's meta-data, particularly expertise in data analysis (Part 4)
- *Corporate awareness* – educating to increase awareness of the importance of data; also disseminating information on what data exists and for what purpose
- *Corporate requirements* – identifying corporate data requirements, particularly building a corporate data architecture which incorporates strategic planning
- *Data analysis* – coordinating the use of a standard data analysis methodology, and using such a methodology to develop business data models
- *Data control* – implementing standards for ensuring that access to data is controlled; also ensuring that suitable recovery procedures are in place
- *Data security* – ensuring that both technical and administrative controls are in place to protect against threats
- *Data definition* – implementing standards for the definition of data and controlling the medium for the recording and communication of such definitions
- *Data integrity* – implementing standard mechanisms for ensuring the integrity of an organisation's data; also documenting the rules for ensuring integrity
- *Data dictionary management* – promoting the use of a logical data dictionary (see below) and implementing standards for its control; also monitoring the use of and content of the data dictionary
- *Data privacy* – implementing procedures to ensure that the organisation complies with any legislation concerning national data regulation
- *Data sharing* – encouraging sharing of data across applications and promoting the idea of data that is independent of applications

22.3 THE NEED FOR DATA ADMINISTRATION

The need for a data administration function arises from the problems of managing data within an organisation. A list of such problems is detailed below:

- A number of applications are developing within some organisations which use different definitions for the same data
- Data held by a number of diverse applications is inconsistent
- Decision-makers within an organisation receive conflicting data from different sources within that organisation

- Decision-makers receive data too late for it to be useful
- Decision-makers receive too much irrelevant data
- There are notable gaps in the data collected by an organisation
- Departments within an organisation have no clear idea why they collect certain data

In this view, data administration may be seen as an attempt to develop some order from the chaos of corporate information systems. However, data administration also involves planning the data required for future information systems. Data administrators would be key personnel in any strategic data planning effort (Chapter 21).

Example Consider the case of identifiers as data. A data administrator should ensure that identifiers for objects such as products, people, invoices and orders are designed to have a number of features:

- An identifier, as a matter of definition, should be uniquely associated with one, and only one, object in a database. Hence, every object in a database, such as an instance of a product, should have an identifier
- An identifier should be assigned immediately on creation of an object
- An identifier should not contain any details about the object it identifies. It should serve the sole function of identifying an object. The reason for this is that so-called non-mnemonic identifiers maintain stability over time

Consider the case where a company uses a three-digit code to identify its products. The first digit is used to indicate the warehouse where the product is stored. Now suppose the company decides to change its warehousing practice and moves all products of a particular type from one warehouse to another. This will necessitate changing all the product codes for the products moved.

A natural consequence of the discussion above is that it is important to administer and control the assignment and use of identifiers in organisations. It is usually not good practice to rely on identifiers supplied by external agencies. For example, suppose an organisation uses the delivery advice number from its supplier to identify different deliveries. If the supplier inadvertently sends two separate deliveries with the same advice number then the organisation's internal information systems are likely to suffer.

22.4 THE DIFFICULTIES OF DATA ADMINISTRATION

It is rare for a data administration function to be set up in an environment in which no applications systems yet exist. Most data administration departments are set up as a response to some of the problems outlined above. In such a situation it is usually a significant task to reconcile data collected from diverse areas of the organisation. Many parts of the organisation will openly disagree about data requirements. Data administration is hence frequently a function which is shaped by the sociology, politics and economics of organisations.

22.5 THE PLACE OF THE DATA ADMINISTRATION FUNCTION

The concept of data administration crosses departmental and indeed organisational boundaries. The relationship between data administration and the structure of organisations is therefore significant.

In many organisations the data administration function has arisen naturally from the realms of the information systems department. Data administration then becomes another arm of information systems. However, because of its cross-departmental role, there is some argument for placing the data administration function outside the information systems department.

Many have argued that wherever it is placed, the data administration function cannot work effectively unless it has access to the management level of organisations. Some have even suggested that the data administrator should have a seat at executive board level.

22.6 DATA DICTIONARIES

The main tool of the database or data administrator is the data dictionary. In this section we first define what we mean by the term data dictionary, then we distinguish between different types of data dictionary.

22.6.1 WHAT IS A DATA DICTIONARY?

A data dictionary is a means for recording the meta-data of some organisation (Navathe and Kerschberg, 1986); that is, data about data. What constitutes meta-data is obviously determined by the stage of database development as discussed in Chapter 14. In this sense, we might usefully define three levels of data dictionary:

- Conceptual data dictionaries record meta-data at a very high level of abstraction. In Chapter 16 we discussed the use of a data dictionary to record the details of an entity–relationship model
- Logical data dictionaries are used to record data requirements independently of how these requirements are to be met. Logical meta-data is however at a slightly lower level of abstraction than conceptual meta-data
- Physical data dictionaries are used to record data structures; that is, they record meta-data relating to actual database or file structures. The system tables at the heart of a relational DBMS comprise a physical data dictionary

In terms of meta-data, therefore, we conventionally mean both data resources and data requirements. A data dictionary is a mechanism for recording the data resources and requirements of some organisation. This means that a data dictionary cannot be considered solely as a tool for the analysis and design of database systems. It is also an important implementation tool and an important function of what we might call the corporate information architecture. Conceptual and logical data dictionaries are usually the realm of the data administrator. Physical data dictionaries are the realm of the database administrator (Chapter 23).

22.6.2 ACTIVE AND PASSIVE DATA DICTIONARIES

The classic data dictionary is a passive repository for meta-data. In other words, the first data dictionaries were built as systems external to a database and DBMS. They were systems built to record the complexities of a given database structure for use by application development staff and database administrators (DBAs).

If we equate the term data dictionary with the set of system tables in a relational database, then a data dictionary in a relational system is an active repository. It is an inherent, internal part of the database system designed to buffer the user from the base tables.

As we have seen, many DBMS are beginning to extend the functionality of such active data dictionaries, particularly in the area of integrity management.

Traditionally, integrity issues have been external to the database system. Integrity has primarily been the province of application systems written in languages such as COBOL or C which interact with the database. Programs are written in such languages to validate data entered into the database and ensure that any suitable propagation of results occurs.

Many have argued however that the logical place for integrity is within the domain of the data dictionary under the control of the DBMS. The argument is that integrity cannot be divorced from the underlying database. Two or more application systems interacting with one database may enforce integrity differently or inconsistently. Hence, there should be only one central resort for monitoring integrity. Integrity should be the responsibility of the DBA. Mechanisms should be available therefore for the DBA to define and enforce integrity via the DBMS.

22.7 DATABASE SECURITY

Database security is the process of protecting a database from external threats. Database security is of primary concern to the data administrator because of the key value that data holds for modern organisations.

Threats include possibilities of theft and fraud, loss of confidentiality, loss of privacy, loss of integrity and loss of availability. Examples of each type of threat are given below:

- *Theft and fraud* – examples here include a person, not authorised to do so, updating corporate data with the aim of defrauding his or her employer. Another example is that of a hacker making an illegal entry into a database system and extracting corporate data without permission
- *Confidentiality* – an unauthorised person viewing information on confidential corporate policies and disclosing it to outside agencies
- *Privacy* – an unauthorised person viewing data on persons held by an organisation
- *Integrity* – a software virus causing corruption of data or loss of integrity through software or hardware failure
- *Availability* – a database system becoming unavailable because of natural disasters such as fire and flood, or disasters initiated by humans such as bomb attacks

The data administrator tasked with managing database security will need to plan a number of computer-based and non-computer-based measures for countering threats such as those listed above. Computer-based measures include:

- Implementing suitable authorisation in relation to an operating system on which the database system runs
- Implementing authorisation strategies that grant privileges to certain users and groups to access certain database objects via the chosen DBMS
- Setting up views into the database and granting access to such views to users and groups
- Taking regular backups of the whole of or part of a database
- Maintaining a log of all changes made to the database
- Utilising suitable strategies for encrypting sensitive data

The implementation of many of these computer-based strategies may be delegated to the DBA. Non-computer-based measures include:

- Establishing a security policy and plan
- Putting suitable personnel controls in place, such as separation of duties
- Positioning computer hardware in secure environments through physical access controls
- Securing copies of data and software in off-site, fireproof storage

The data administrator will need to liaise with other staff within the organisation, such as people managing the estate of the organisation, to execute these security strategies effectively.

22.8 CASE STUDY: DATA ADMINISTRATION IN A UNIVERSITY SETTING

Clearly, a university can be considered to be made up of a number of interdependent human activity systems (Chapter 4). A systems model of a university is likely to consist of at least three core processes. These are all organisational processes in that they are cross-organisational, have a defined set of customers, and link to the core competencies of the organisation.

- *Teaching*. It is important for a university to achieve suitable levels of enrolment by students on courses and to progress students through various undergraduate and postgraduate programmes
- *Research*. It is important for a university to achieve satisfactory levels of research funding from government and industry
- *Consultancy*. It is important for universities to achieve satisfactory levels of income from consultancy projects

Universities need to maintain accurate data about students, staff, courses, modules and finance in order to work effectively and efficiently. Such data supports the key organisational processes of teaching, research and consultancy:

- The teaching process needs accurate data about students, courses, modules, assessments and fees
- The research process needs accurate data about research outputs such as publications and research inputs such as grant income
- The consultancy process needs accurate data about projects and consultancy income

Data administration in such a university will be concerned with ensuring that data is not collected, stored and manipulated in a piecemeal manner by administrative and academic departments; in order to ensure this it is likely to enforce a series of common data definitions for such data across the organisation. This will enable sharing of data across the organisation.

The data administrator will also ensure that access to such data is regulated and that the privacy of sensitive data (such as that held about students) is preserved. He or she will also be critically involved in monitoring current data requirements across the organisation and constructing plans for data to fulfil future organisational strategy.

22.9 SUMMARY

- Database technology, because of its centrality in the modern IS architecture, has stimulated the development of a large range of roles for servicing this technology. Two of the most commonplace roles are the data administrator and the database administrator

- The data administrator is a high-level, corporate function
- As far as development is concerned, the data administrator will be involved in the analysis and design of a database system. Data administrators are also particularly involved in the management of the organisational data resource
- The main tool of the data administrator is the data dictionary, particularly conceptual and logical data dictionaries. A data dictionary is a means for recording the meta-data of some organisation
- Database security is a concern of the data administrator in association with the database administrator. Database security involves protecting a database from external threat by using computer-based and non-computer-based controls

22.10 ACTIVITIES

1. In terms of some organisation known to you, examine whether it has a data administration function. If it does, identify the functions undertaken by the data administration function. If it does not, examine the case for data administration.
2. Examine a DBMS known to you. See if you can determine the structure of the physical data dictionary used by the DBMS.

22.11 REFERENCES

Beynon-Davies, P. (2002). *Information Systems: An Introduction to Informatics in Organisations*. Basingstoke, Palgrave (formerly Macmillan).

Gillenson, M.L. (1987). The duality of database structures and design techniques. *Comm. of ACM* **30**(12): 1056–65.

Navathe, S.B. and L. Kerschberg (1986). Role of data dictionaries in information resource management. *Information and Management* **10**: 21–46.

CHAPTER 23

DATABASE ADMINISTRATION

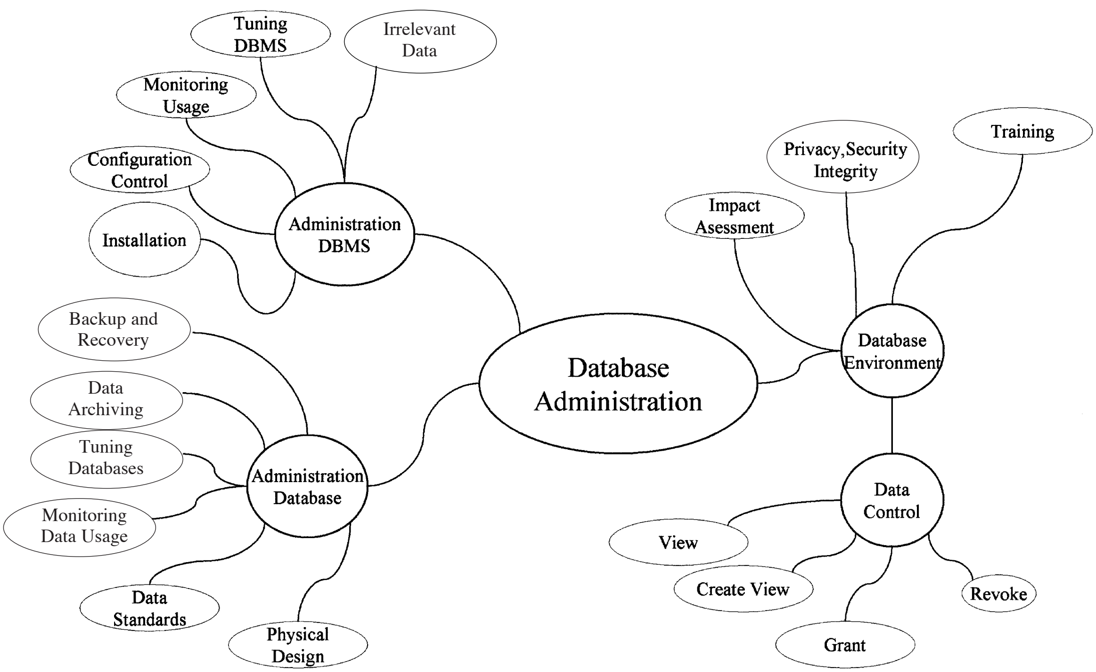
Control thy passions, lest they take vengeance on thee.

Epictetus (AD 55–AD 135)

LEARNING OUTCOMES

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

- Define the concept of database administration
- Discuss the key features of the database administrator
- Discuss the key administration function of data control



23.1 INTRODUCTION

The database administrator (DBA) is responsible for the technical implementation of database systems, managing the database systems currently in use, and setting and enforcing policies for their use. Whereas the data administrator (Chapter 22) works primarily at the conceptual level of business data, the DBA works primarily at the physical level. The place where the data administrator and the DBA meet is at the logical level. Both the data administrator and DBA must be involved in the system-independent specification and management of data.

The need for a specialist DBA function varies depending on the size of the database system being administered. In terms of a small desktop database system, the main user of the database will probably perform all DBA tasks such as taking regular backups of data. However, when a database is being used by many people and the volume of data is significant, the need for a person or persons who specialise in administration functions becomes apparent.

23.2 KEY FUNCTIONS OF DATABASE ADMINISTRATION

In many organisations, particularly small organisations, the DBA will be expected to undertake many data administration tasks. However, in general, the DBA can be said to have the following core responsibilities: administration of the database, administration of the DBMS and administration of the database environment.

23.2.1 ADMINISTRATION OF THE DATABASE

The DBA would normally be expected to engage in the following key activities in relation to administering a database or a series of databases:

- *Physical design* – whereas the data administrator will be concerned with the conceptual and logical design of database systems, the DBA will be concerned with the physical design and implementation of databases
- *Data standards and documentation* – ensuring that physical data is documented in a standard way such that multiple applications and end-users can access the data effectively
- *Monitoring data usage and tuning database structures* – monitoring live running against a database and modifying the schema or access mechanisms to increase the performance of such systems
- *Data archiving* – establishing a strategy for archiving of ‘dead’ data
- *Data backup and recovery* – establishing a procedure for backing-up data and recovering data in the event of hardware or software failure

23.2.2 ADMINISTRATION OF THE DBMS

The DBA would normally be expected to engage in the following key activities in relation to administering a DBMS:

- *Installation* – taking key responsibility for installing DBMS or DBMS components
- *Configuration control* – enforcing policies and procedures for managing updates and changes to the software of the database system
- *Monitoring DBMS usage and tuning DBMS* – monitoring live running of database systems and tailoring elements of the DBMS structure to ensure the effective performance of such systems

23.2.3 ADMINISTRATION OF THE DATABASE ENVIRONMENT

By administering the database environment, we mean monitoring and controlling the access to the database and DBMS by users and application systems. Activities in this area include:

- *Data control* – establishing user groups, assigning passwords, granting access to DBMS facilities, granting access to databases
- *Impact assessment* – assessing the impact of any changes in the use of data held within database systems
- *Privacy, security and integrity* – ensuring that the strategies laid down by data administration for data integrity, security and privacy are adhered to at the physical level
- *Training* – holding responsibility for the education and training of users in the principles and policies of database use

23.3 DATA CONTROL

One of the primary functions of the DBA is data control.

In any multi-user system, some person or persons have to be given responsibility for allocating resources to the community of users and monitoring the usage of such resources. In database systems, two resources are of preeminent importance: data and DBMS functions. Data control is the activity which concerns itself with allocating access to data and allocating access to facilities for manipulating data. Data control is normally the responsibility of the DBA.

In this section we shall examine some of the fundamental aspects of data control in relational database systems. First, we shall examine the concept of a relational view in SQL. Second, we shall discuss how DBAs manipulate the view concept in order to allocate access to data. Third, we shall examine the DBA's ability to allocate access to DBMS functions such as data definition.

23.4 VIEWS

A view is a virtual table. It has no concrete reality in the sense that no data is actually stored in a view. A view is simply a window into a database in that it presents to the user a particular perspective on the world represented by the database.

Example For instance, suppose we have a base table of student data, a sample of which is given below:

Students

student No	***student Name***	***sex***	***Date of Birth***	***studentTerm Addr***	***student Tel***	***course Code***
1234	Paul S	M	01/03/1980	14, Hillside Tce	432568	CSD
1235	Patel J	M	01/05/1981	28, Park Place	235634	CSD
2367	Singh B	F	01/01/1974	10a, Pleasant Flats	345623	BSD
3245	Conran J	M	01/08/1980	Rm 123, University Towers	NULL	CSD
2333	Jones L	F	01/02/1981	9 Redhill St	222333	BSD

We can identify three main interdependent uses for views in a database system: to simplify, to perform some regular function and to provide security.

23.4.1 SIMPLIFICATION

Suppose the students table above is defined for an entire university. Each department or school within the university however only wishes to see data relevant to its own courses. A number of views might then be created to simplify the data that each department or school sees.

Example For instance, we might have two views established on the table above, one for the students of the school of computing and one for the students of the business school.

Computing Students

student No	**student Name**	**sex**	**studentTermAddr**	**student Tel**
1234	Paul S	M	14, Hillside Tce	432568
1235	Patel J	M	28, Park Place	235634
3245	Conran J	M	Rm 123, University Towers	NULL

			Business Students	
student No	student Name	sex	studentTermAddr	studentTel
2367	Singh B	F	10a, Pleasant Flats	345623
2333	Jones L	F	9 Redhill St	222333

23.4.2 FUNCTIONALITY

There is an implicit edict in the relational model: store as little as possible. One consequence of this is that any data that can be derived from other data in the database is not normally stored.

Example If, for instance, there is a requirement to regularly display the ages of students, we would not store both a student's age and his or her birthdate. A student's age can be derived from a person's birthdate. The view mechanism is the conventional means for performing such computations and displaying them to the user. For example, in the view below we display the sex and age profile of students. Age is a derived attribute in this view.

Students			
student No	student Name	sex	age
1234	Paul S	M	19
1235	Patel J	M	18
2367	Singh B	F	25
3245	Conran J	M	18
2333	Jones L	F	19

23.4.3 SECURITY

In a university we might want to restrict each department or school only to be able to update the data relating to its own students. A view declared on each department or school and attached to a range of user privileges can ensure this. Hence the two views above, *Computing Students* and *Business Students*, might be secured for access only by administrative staff of the school of computing and the business school respectively.

23.5 UPDATING THROUGH VIEWS

Retrieving information through views is a straightforward process. It involves merging the original query on the view with the view definition itself. File maintenance operations such as insertion, deletion and update are however more complex to accomplish via views, and hence deserve more consideration.

The main point is that not all views are updatable.

Example Consider for instance, two views declared on the students table, S1 and S2. S1 produces a view of all the columns in the base table but only for computing students; S2 is a vertical fragment of the base table:

S1

student No	student Name	sex	Date of Birth	studentTerm Addr	student Tel	course Code
1234	Paul S	M	01/03/1980	14, Hillside Tce	432568	CSD
1235	Patel J	M	01/05/1981	28, Park Place	235634	CSD
3245	Conran J	M	01/08/1980	Rm 123, University Towers		CSD

S2

studentName	Sex	Course Code
Paul S	M	CSD
Patel J	M	CSD
Singh B	F	BSD
Conran J	M	CSD
Jones L	F	BSD

In the case of S1, for instance, we can insert a new row into the view, delete an existing record from a view or update an existing field in the view. In the case of S2, problems arise. For example, suppose we wished to insert a record for a new student, Balshi J, M, BSD. We would have to insert the record *null, Balshi J, M, null, null, BSD* into the underlying students table. This attempt will fail since studentNo (the table's primary key) must not be null.

On closer examination, the main difference between S1 and S2 lies in the fact that S1 contains the primary key column of the base table and S2 does not.

This means that updates to S1 unambiguously update rows in a view. Updates to S2 are frequently ambiguous. S1 is hence said to be an updatable view whereas S2 is a non-updatable view.

23.6 THE *CREATE VIEW* STATEMENT

As stated previously, a view is a virtual table. It has no concrete reality in the sense that no data is actually stored in a view. A view is simply a window into a database in that it presents to the user a particular perspective on the world represented by the database. A view in SQL is a packaged *select* statement. The syntax is as follows:

```
CREATE VIEW <view name>
AS <select statement>
```

Example The following two examples create the S1 and S2 views discussed in section 23.4.1:

```
CREATE VIEW S1 AS
SELECT *
FROM Students
WHERE courseCode = 'CSD'

CREATE VIEW S2 AS
SELECT studentName, sex, courseCode
FROM Students
```

When a view has been created, it can be queried in the same manner as a base table. For instance:

```
SELECT *
FROM S1
WHERE sex = 'F'
```

23.6.1 UPDATING THROUGH VIEWS IN SQL

In section 23.5 we discussed how certain views are not updatable and how certain other views are updatable. In this context, C. Date has made the distinction between three types of views (Date, 2000):

- Views which are updatable in theory and in practice
- Views which are updatable in theory, but not yet in practice
- Views which do not appear to be updatable in theory and therefore cannot be updatable in practice

What then defines the first type of view? As far as most SQL-based systems are concerned, the following conditions have to be satisfied:

- The *select* clause in a view cannot use a distinct qualifier or any function definition
- The *from* clause must specify just one table. No joins are allowed
- The *where* clause cannot contain a correlated subquery
- The *group by* and *order by* clauses are not allowed
- No *union* operator is permitted

23.7 DATA CONTROL COMMANDS FOR DBAs

In any database system, the DBA needs to be able to do three main things:

- Prevent would-be users from logging-on to the database
- Allocate access to specific parts of the database to specific users
- Allocate access to specific operations to specific users

In this section we shall examine SQL's capability for handling the second of these needs. Because the first and third functions are non-standard across DBMS we shall resort to examining the facilities available under the ORACLE RDBMS.

23.7.1 *GRANT* AND *REVOKE*

SQL provides two commands for controlling access to data. One command gives access privileges to users. The other command takes such privileges away.

To give a user access to a view or a table we use the grant command:

```
GRANT [ALL : SELECT : INSERT : UPDATE : DELETE ]
ON [<table name> : <view name>]
TO <user name>
```

Example For instance:

```
GRANT INSERT ON Modules TO pbd
GRANT SELECT ON Lecturers TO pbd
```

To take existing privileges away from a user we use the revoke command.

```
REVOKE [ALL : SELECT : INSERT : UPDATE : DELETE ]
ON [<table name> : <view name>]
FROM <user name>
```

Example For instance:

```
REVOKE INSERT ON Modules FROM pbd
REVOKE SELECT ON Lecturers FROM pbd
```

23.7.2 GRANTING ACCESS VIA VIEWS

One of the ways of enforcing control in relational database systems is via the view concept.

Example Let us suppose, for instance, that we wish to give R. Evans the ability to look at and update the records of Lecturers in his department – Computer Studies. We do not however wish to give R. Evans the ability to delete Lecturer records or insert new Lecturer records. To enact these organisation rules we create a view:

```
CREATE VIEW Evans AS
    SELECT *
    FROM Lecturers
    WHERE deptName =
    (SELECT deptName
    FROM Lecturers
    WHERE staffName = 'Evans R')
```

Note that the employee record for R. Evans would be included in this view. If we now provide select and update access for R. Evans as below:

```
GRANT SELECT, UPDATE
ON research
TO REvans
```

this particular person will be able to change his own salary! There are a number of ways we can prevent this from happening by modifying the view definition. One possible solution is given below:

```
CREATE VIEW Evans AS
    SELECT *
    FROM Lecturers
    WHERE deptName =
    (SELECT deptName
    FROM Lecturers
    WHERE staffName = 'Evans R')
    AND staffName <> 'Evans R'
```

23.7.3 DBA PRIVILEGES IN ORACLE

In the above discussion we have considered how SQL permits the DBA to specify access to parts of the database and how it permits the DBA to associate these allocations with particular users. However, at present SQL does not contain a definition of how users are declared to the database system in the first place, and also how such users are given access to various DBMS facilities. Assigning users and giving such users access privileges are two DBA functions which vary from product to product. To give the reader a flavour of the type of mechanisms available, we discuss here those available under the ORACLE RDBMS (Versions 6 through 8).

23.7.4 CONNECT, RESOURCE AND DBA

Most RDBMS differ in the levels of facilities or system privileges offered and what they are called. In this section we illustrate what is available under the ORACLE DBMS.

When a company receives a new ORACLE system it comes automatically installed with two 'super' users. The first thing the DBA does is change the default passwords for these user names. The second thing he or she does is start enrolling users into the system.

To declare an ORACLE user the DBA must normally supply three pieces of information: a distinct user name, a password and the level of privilege granted to the user.

There are three classes of ORACLE user: *connect*, *resource* and *dba*. A connect user is able to look at other users' data only if allowed by other users, perform data manipulation tasks specified by the DBA, and create views. Resource privilege allows the user to create database tables and indexes and grant other users access to these tables and indexes. Dba privilege is normally given to a chosen few. The superusers discussed above have dba privileges. Such privileges permit access to any user's data, and allow the granting and revoking of access privileges to any user in the database.

When a new user is enrolled into the system the database administrator grants the user one or more of these privileges using the command below:

```
GRANT {CONNECT : RESOURCE : DBA}
TO <username>
[IDENTIFIED BY <password>]
```

Example For instance:

```
GRANT CONNECT, RESOURCE
TO pbd
IDENTIFIED BY crs
```

In a similar manner to granting database access, DBMS privileges can be revoked as below:

```
REVOKE {CONNECT : RESOURCE : DBA}
FROM <username>
```

Example For instance:

```
REVOKE CONNECT, RESOURCE
FROM pbd
```

23.8 CASE STUDY: DATA CONTROL OF THE ACADEMIC DATABASE

Assume we appoint a DBA to control the academic database below:

Courses(<u>courseCode</u>, courseName, deptName, courseLeader)
CourseEnrolment(<u>courseNo</u>, <u>studentNo</u>, enrolmentDate, feePaid)
Modules(<u>moduleNo</u>, moduleName, level, courseCode)
ModuleEnrolment(<u>moduleNo</u>, <u>studentNo</u>, enrolmentDate)
ModuleAssessment(<u>moduleNo</u>, <u>studentNo</u>, assessmentNo, grade)
Lecturers(<u>staffNo</u>, staffName, status, salary)
Students(studentNo, studentName, gender, studentTermAddress, studentTermTelNo, studentHomeAddress, studentHomeTelNo)
Teaches(<u>lecturerNo</u>, <u>moduleNo</u>)
Prerequisites(<u>moduleNo</u>, <u>moduleNo</u>)

The DBA first has to establish the major user groups for this database. Five user groups are established – personnel administrators, teaching administrators, registry staff, academic staff and university management. The DBA establishes the following data control requirements for each such group:

- *Personnel administrators*. Lecturers [ALL]
- *Teaching administrators*. Courses, Modules, Prerequisites, Teaches, ModuleAssessment [ALL]
- *Registry staff*. Students, CourseEnrolment, ModuleEnrolment [ALL]
- *Academic staff*. View of Lecturer's Modules [SELECT]; View of Students on Lecturer's Module [SELECT]
- *Managers*. Entire database [SELECT]

23.9 SUMMARY

- The database administrator is a low-level, technical function
- In terms of development, the database administrator will be involved in the physical design of database systems. On the data management side, the database administrator will be particularly concerned with the issue of data control

- Data control is the activity devoted to controlling access to data and DBMS functions
- In relational database systems, the data aspect of data control revolves around the concept of a view. A view is a virtual table. Views can be used to create 'windows' into a database which can be allocated to specific users of the database
- The DBMS-side of data control varies from product to product. Most RDBMS however have facilities for assigning user names and passwords to users, and defining the overall privileges available to users
- ISO SQL has a limited range of DBA functions, particularly focused on granting access to data and revoking access to data. Most DBMS also have a range of non-standard functions for declaring users and passwords, fine-tuning database sizing, monitoring the performance of a database and fragmenting databases

23.10 ACTIVITIES

1. Examine an organisation known to you. Determine whether it employs database administrators. Examine the activities conducted by the DBAs in this organisation and compare them against the activities identified in this chapter.
2. Examine a DBMS known to you. Determine the mechanisms for enforcing data and system privileges in this DBMS.
3. Create the views discussed in section 23.4.2 using SQL.
4. Grant appropriate data access on these views using SQL.

23.11 REFERENCE

Date, C.J. (2000). *An Introduction to Database Systems*. Reading, MA, Addison-Wesley.