Chapter 1

Introduction to Databases

Chapter 1 - Objectives

- **Some common uses of database systems.**
- **Characteristics of file-based systems.**
- **Problems with file-based approach.**
- **Meaning of the term database.**
- **Meaning of the term Database Management System (DBMS).**

Chapter 1 - Objectives

- **Typical functions of a DBMS.**
- **Major components of the DBMS environment.**
- **Personnel involved in the DBMS environment.**
- **History of the development of DBMSs.**
- **Advantages and disadvantages of DBMSs.**

Examples of Database Applications

- **Purchases from the supermarket**
- **Purchases using your credit card**
- **Booking a holiday at the travel agents**
- **Using the local library**
- **Using the Internet**
- **Renting a video**
- **Studying at university**

File-Based Systems

- **File-based systems were an early attempt to computerize the manual filing system that we are all familiar with. By having files stored on computers, the data could be accessed more efficiently.**

- **File-based system is a collection of application programs that perform services for the end-users, such as the production of reports. Each program defines and manages its own data.**

- **It is the predecessor of DBMS.**

File-Based Systems

- An organization might have physical files set up to hold all correspondence relating to a project, product, task, client, or employee.
- Typically, there are many such files that are stored in one or more cabinets.
- When it is necessary to look something up, we go to the filing system and search through the system, starting at the first entry, until we find what we want.
- The manual filing system works well as long as the number of items to be stored is small. It even works adequately when there are large numbers of items and we only have to store and retrieve them.

File-Based Systems

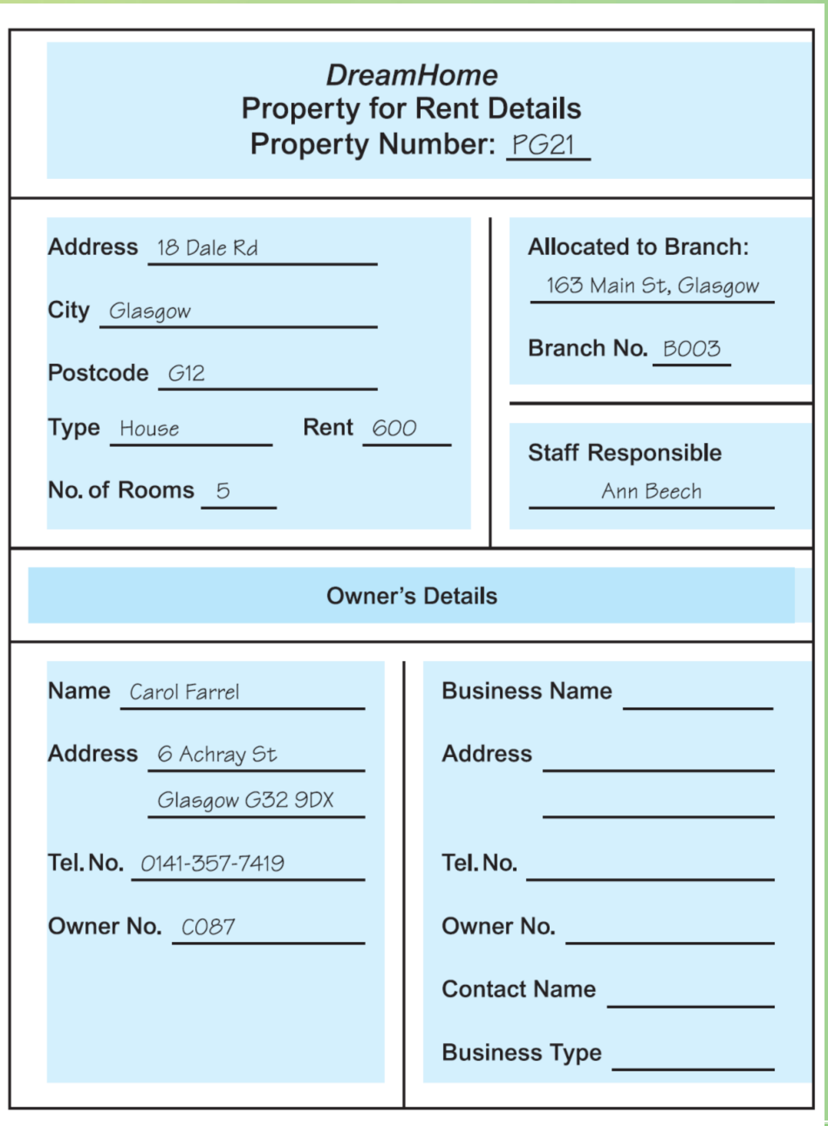
- **However, the manual filing system breaks down when we have to cross-reference or process the information in the files.**
- **For example, a typical real estate agent's office might have a separate file for each property for sale or rent, each potential buyer and renter, and each member of staff.**
- **Consider the effort that would be required to answer the following questions:**
  - What three-bedroom properties do you have for sale with an acre of land and a garage?
  - What apartments do you have for rent within three miles of downtown?
  - What is the average rent for a two-bedroom apartment?
  - What is the annual total for staff salaries?
  - What is the expected monthly net income for the next financial year?

File-Based Systems

- Increasingly nowadays, clients, senior managers, and staff want more and more information. In some business sectors, there is a legal requirement to produce detailed monthly, quarterly, and annual reports.
- Clearly, the manual system is inadequate for this type of work. The file-based system was developed in response to the needs of industry for more efficient data access.
- However, rather than establish a centralized store for the organization's operational data, a decentralized approach was taken, where each department, with the assistance of Data Processing (DP) staff, stored and controlled its own data.
- To understand what this means, consider the *DreamHome* example.

File-Based Systems - the DreamHome example

- **The Sales Department is responsible for the selling and renting of properties.**
- **For example, whenever a client who wishes to offer his or her <u>property</u> as a rental, a form as shown in the figure is completed.**



DreamHome
Property for Rent Details
Property Number: PG21

Address 18 Dale Rd

City Glasgow

Postcode G12

Type House Rent 600

No. of Rooms 5

Allocated to Branch:
163 Main St, Glasgow

Branch No. B003

Staff Responsible
Ann Beech

Owner's Details

Name Carol Farrel

Address 6 Achray St
Glasgow G32 9DX

Tel. No. 0141-357-7419

Owner No. C087

Business Name ____

Address ____
____

Tel. No. ____

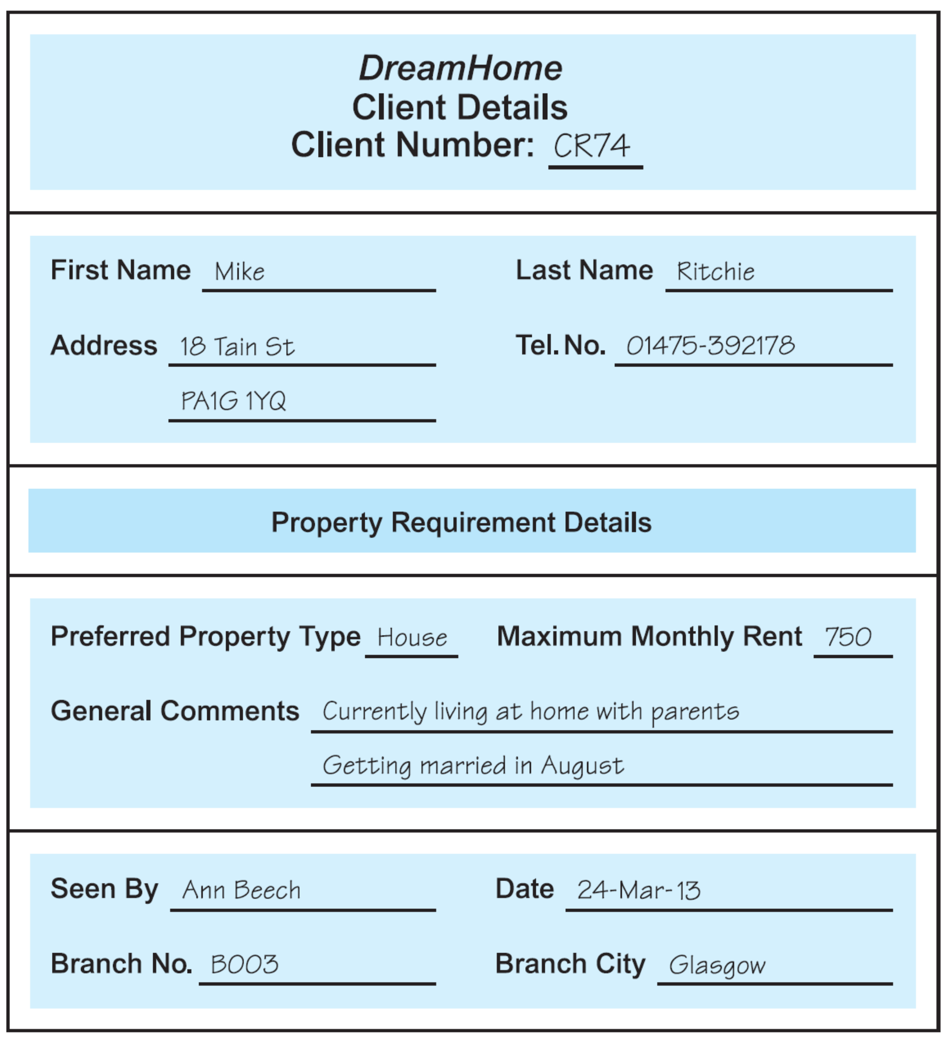
Owner No. ____

Contact Name ____

Business Type ____

File-Based Systems - the DreamHome example

- **The Sales Department also handles inquiries from <u>clients</u>, and a form similar to the one shown in the figure is completed for each one.**



DreamHome
Client Details
Client Number: CR74

First Name Mike **Last Name** Ritchie

Address 18 Tain St
PA1G 1YQ

Tel. No. 01475-392178

Property Requirement Details

Preferred Property Type House **Maximum Monthly Rent** 750

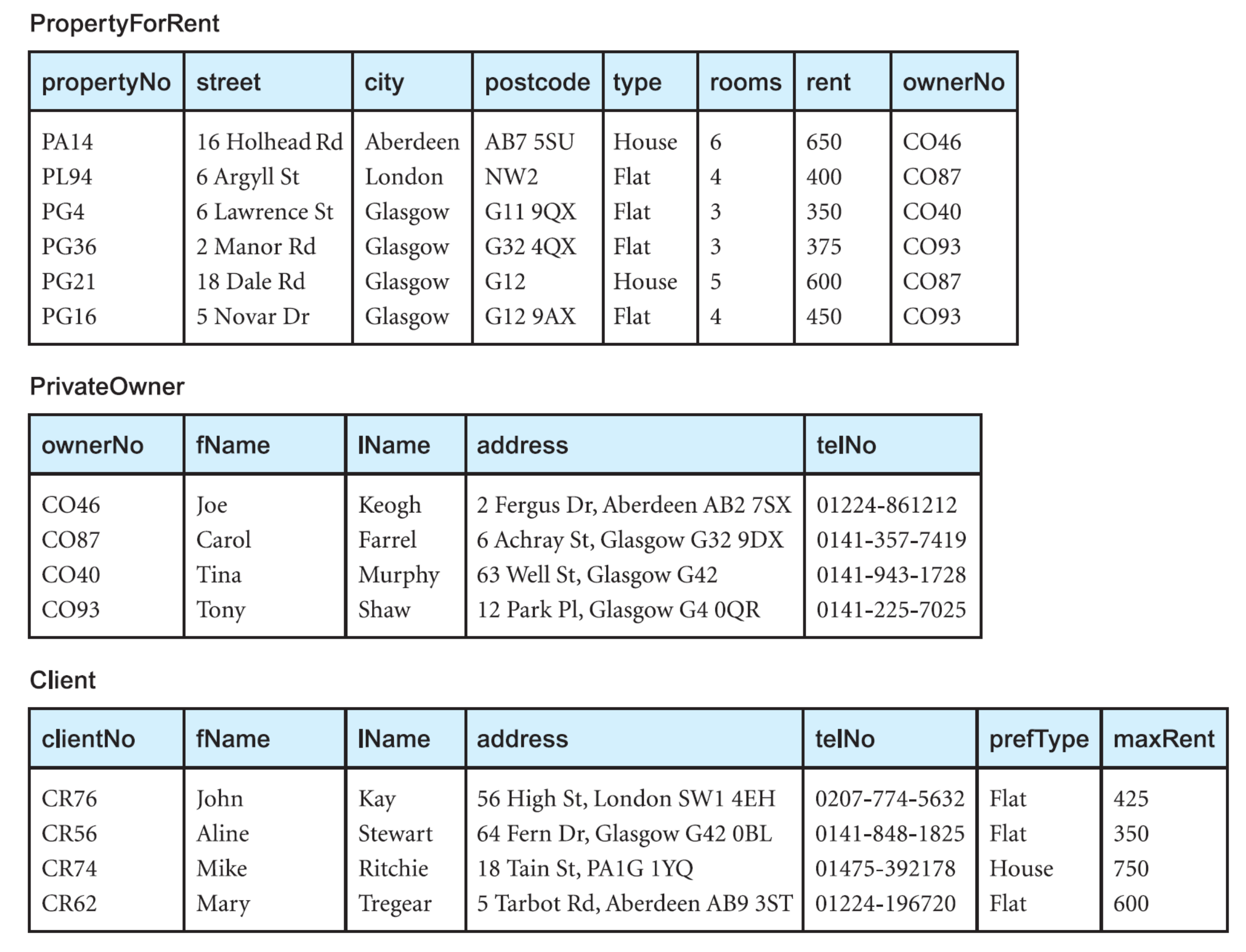
General Comments Currently living at home with parents
Getting married in August

Seen By Ann Beech **Date** 24-Mar-13

Branch No. B003 **Branch City** Glasgow

File-Based Systems - the DreamHome example

- **With the assistance of the DP Department, the Sales Department creates an information system to handle the renting of property.**
- **The system consists of three files containing <u>property</u>, <u>owner</u>, and <u>client</u> details, as illustrated in the figure.**
- **For simplicity, we omit details relating to members of staff, branch offices, and business owners.**



PropertyForRent

propertyNo	street	city	postcode	type	rooms	rent	ownerNo
PA14	16 Holhead Rd	Aberdeen	AB7 5SU	House	6	650	CO46
PL94	6 Argyll St	London	NW2	Flat	4	400	CO87
PG4	6 Lawrence St	Glasgow	G11 9QX	Flat	3	350	CO40
PG36	2 Manor Rd	Glasgow	G32 4QX	Flat	3	375	CO93
PG21	18 Dale Rd	Glasgow	G12	House	5	600	CO87
PG16	5 Novar Dr	Glasgow	G12 9AX	Flat	4	450	CO93

PrivateOwner

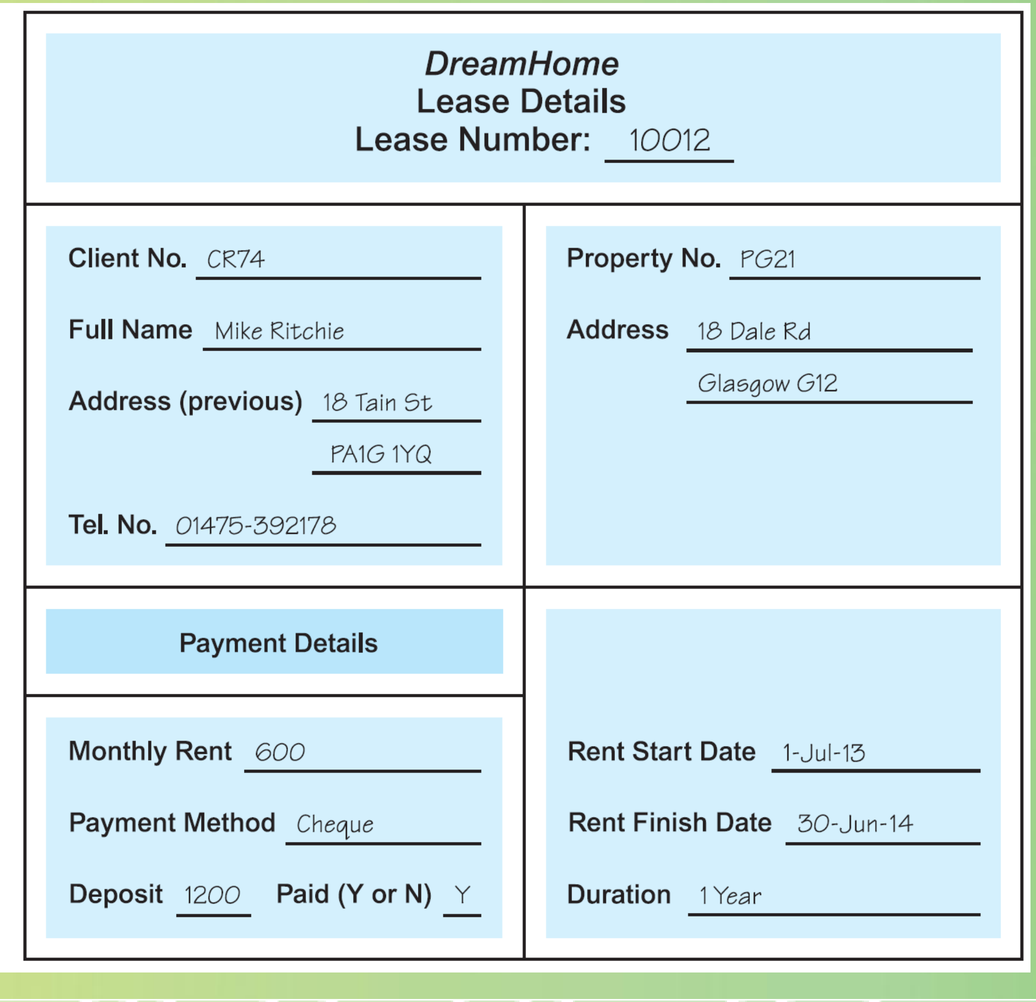
ownerNo	fName	lName	address	telNo
CO46	Joe	Keogh	2 Fergus Dr, Aberdeen AB2 7SX	01224-861212
CO87	Carol	Farrel	6 Achray St, Glasgow G32 9DX	0141-357-7419
CO40	Tina	Murphy	63 Well St, Glasgow G42	0141-943-1728
CO93	Tony	Shaw	12 Park Pl, Glasgow G4 0QR	0141-225-7025

Client

clientNo	fName	lName	address	telNo	prefType	maxRent
CR76	John	Kay	56 High St, London SW1 4EH	0207-774-5632	Flat	425
CR56	Aline	Stewart	64 Fern Dr, Glasgow G42 0BL	0141-848-1825	Flat	350
CR74	Mike	Ritchie	18 Tain St, PA1G 1YQ	01475-392178	House	750
CR62	Mary	Tregear	5 Tarbot Rd, Aberdeen AB9 3ST	01224-196720	Flat	600

File-Based Systems - the DreamHome example

- **The Contracts Department is responsible for handling the lease agreements associated with properties for rent.**
- **Whenever a client agrees to rent a property, a form with the client and property details is filled in by one of the sales staff, as shown in the figure.**
- **This form is passed to the Contracts Department, which allocates a lease number and completes the payment and rental period details.**



DreamHome
Lease Details
Lease Number: 10012

Client No. CR74
Full Name Mike Ritchie
Address (previous) 18 Tain St
PA1G 1YQ
Tel. No. 01475-392178

Property No. PG21
Address 18 Dale Rd
Glasgow G12

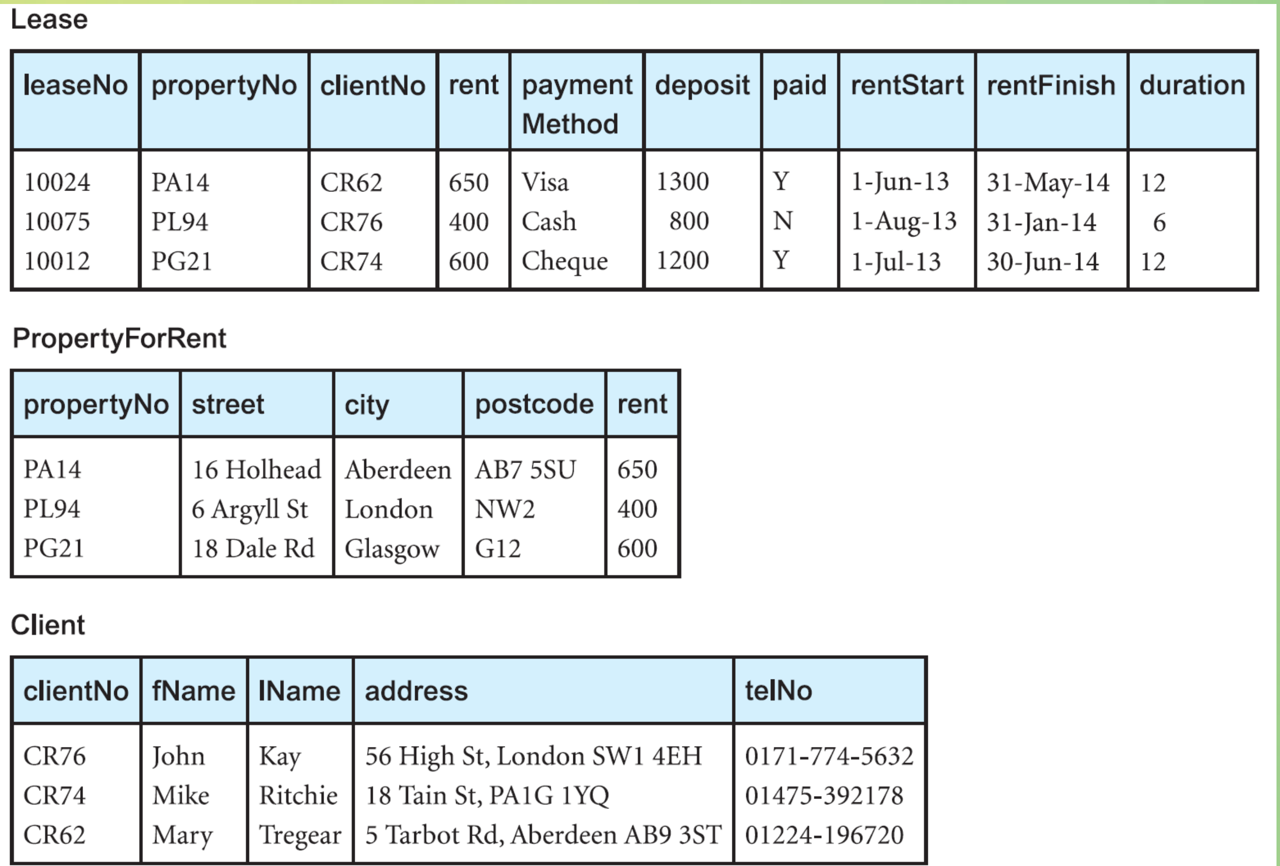
Payment Details

Monthly Rent 600
Payment Method Cheque
Deposit 1200 Paid (Y or N) Y

Rent Start Date 1-Jul-13
Rent Finish Date 30-Jun-14
Duration 1 Year

File-Based Systems - the DreamHome example

- **Again, with the assistance of the DP Department, the Contracts Department creates an information system to handle lease agreements.**
- **The system consists of three files that store lease, property, and client details, and that contain similar data to that held by the Sales Department, as illustrated in the figure.**



Lease

leaseNo	propertyNo	clientNo	rent	paymentMethod	deposit	paid	rentStart	rentFinish	duration
10024	PA14	CR62	650	Visa	1300	Y	1-Jun-13	31-May-14	12
10075	PL94	CR76	400	Cash	800	N	1-Aug-13	31-Jan-14	6
10012	PG21	CR74	600	Cheque	1200	Y	1-Jul-13	30-Jun-14	12

PropertyForRent

propertyNo	street	city	postcode	rent
PA14	16 Holhead	Aberdeen	AB7 5SU	650
PL94	6 Argyll St	London	NW2	400
PG21	18 Dale Rd	Glasgow	G12	600

Client

clientNo	fName	lName	address	telNo
CR76	John	Kay	56 High St, London SW1 4EH	0171-774-5632
CR74	Mike	Ritchie	18 Tain St, PA1G 1YQ	01475-392178
CR62	Mary	Tregear	5 Tarbot Rd, Aberdeen AB9 3ST	01224-196720

File-Based Systems - the DreamHome example

- **Clearly, each department accesses his own files through application programs written especially for it.**
- **Each set of departmental application programs handles data entry, file maintenance, and the generation of a fixed set of specific reports.**
- **More important, the physical structure and storage of the data files and records are defined in the application code.**

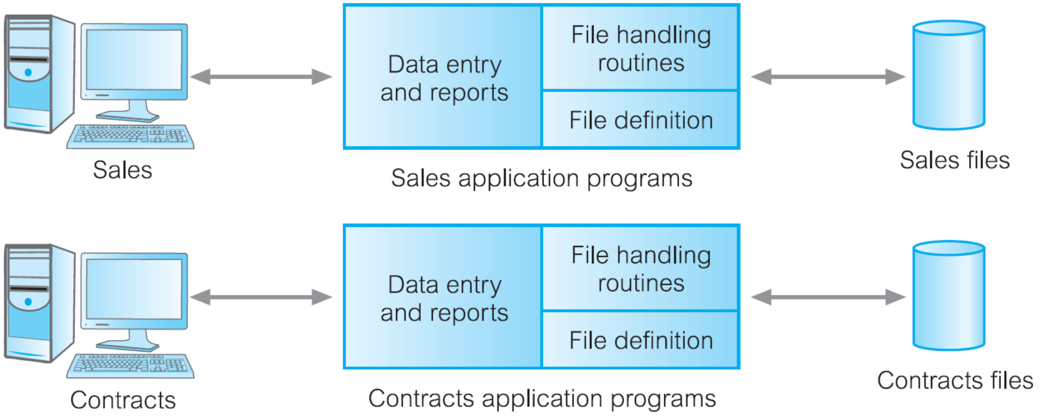


Figure 1.5 File-based processing.

Sales Files

PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)

Contracts Files

Lease (leaseNo, propertyNo, clientNo, rent, paymentMethod, deposit, paid, rentStart, rentFinish, duration)
PropertyForRent (propertyNo, street, city, postcode, rent)
Client (clientNo, fName, lName, address, telNo)

File-Based Systems - the DreamHome example

- **It can be seen quite clearly that there is a significant amount of duplication of data in these departments, and this is generally true of file-based systems.**

- **Before we discuss the limitations of this approach, it may be useful to understand the terminology used in file-based systems.**

- **A file is simply a collection of records, which contains logically related data. For example, the PropertyForRent (in the Contracts Department) file contains three records: *PA14*, *PL94*, and *PG21* (one for each property).**

- **Each record contains a logically connected set of one or more fields (attributes), where each field represents some characteristic of the real-world object that is being modeled. For example, the fields of the PropertyForRent file are *street*, *city*, *postcode*, and *rent*.**

Limitations of File-Based Approach

- **Separation and isolation of data: Each program maintains its own set of data.**
  - **Users of one program may be unaware of potentially useful data held by other programs.**

- **Duplication of data: Same data is held by different programs.**
  - **Wasted time**
  - **Wasted space.**
  - **Potential different values and/or different formats for the same item.**

Limitations of File-Based Approach

- **Data dependence: The physical structure and storage of the data files and records are defined in the application code, and so any changes to an existing structure are difficult to make.**

- **Incompatible file formats: Programs are written in different languages, and so cannot easily access each other's files.**

- **Fixed Queries/Proliferation of application programs**
  - **Programs are written to satisfy particular functions.**
  - **Any new requirement needs a new program.**

Database Approach

- Arose because:
  - Definition of data was embedded in application programs, rather than being stored separately and independently.
  - No control over access and manipulation of data beyond that imposed by application programs.

- Result:
  - the database and Database Management System (DBMS).

Database

- Database is a shared collection of logically related data and its description, designed to meet the information needs of an organization.
- The database is a single, possibly large repository of data that can be used *simultaneously* by many departments and users.
- Instead of disconnected files with redundant data, *all data items are integrated* with a minimum amount of duplication.
- The database is no longer owned by one department but is *a shared corporate resource*.
- The database holds not only the organization’s operational data, but also a description of this data. For this reason, a database is also defined as a self-describing collection of integrated records.
- The description of the data is known as the *system catalog* (or *data dictionary* or *metadata*—the “data about data”).

Database Management System (DBMS)

- **DBMS is a software system that enables users to define, create, maintain, and control access to the database.**
  - **It interacts with the users' application programs and the database.**
  - **It processes the requests and returns the results to the application program.**
- **(Database) application program is a computer program that interacts with database by issuing an appropriate request (SQL statement) to the DBMS.**
  - **It provides the user-interface to send requests to database management system and to receive processed results from database management system.**
  - **It can be conventional batch applications or, more typically nowadays, online applications.**
  - **The application programs may be written in a programming language or in higher-level fourth-generation language.**

Database Management System (DBMS)

- **Typically, a DBMS provides the following facilities:**
  - It allows to define the database through a **Data Definition Language (DDL)** that allows users to specify the data types and structures and constraints.
  - It allows to insert, update, delete, and retrieve data from the database through a **Data Manipulation Language (DML)**.
  - It provides controlled access to the database. For example, it may provide:
    - – a security system, which prevents unauthorized access;
    - – an integrity system, which maintains the consistency of stored data;
    - – a concurrency control system, which allows shared access of the database;
    - – a recovery control system, which restores the database to a previous consistent state following a hardware or software failure;
    - – a user-accessible catalog, which contains descriptions of the data in the database.

Database Management System (DBMS)

- The database approach is illustrated in the figure. It shows the Sales and Contracts Departments using their application programs to access the database through the DBMS.
- Each set of departmental application programs handles data entry, data maintenance, and the generation of reports.
- However, as opposed to the file-based approach, the physical structure and storage of the data are now managed by the DBMS.

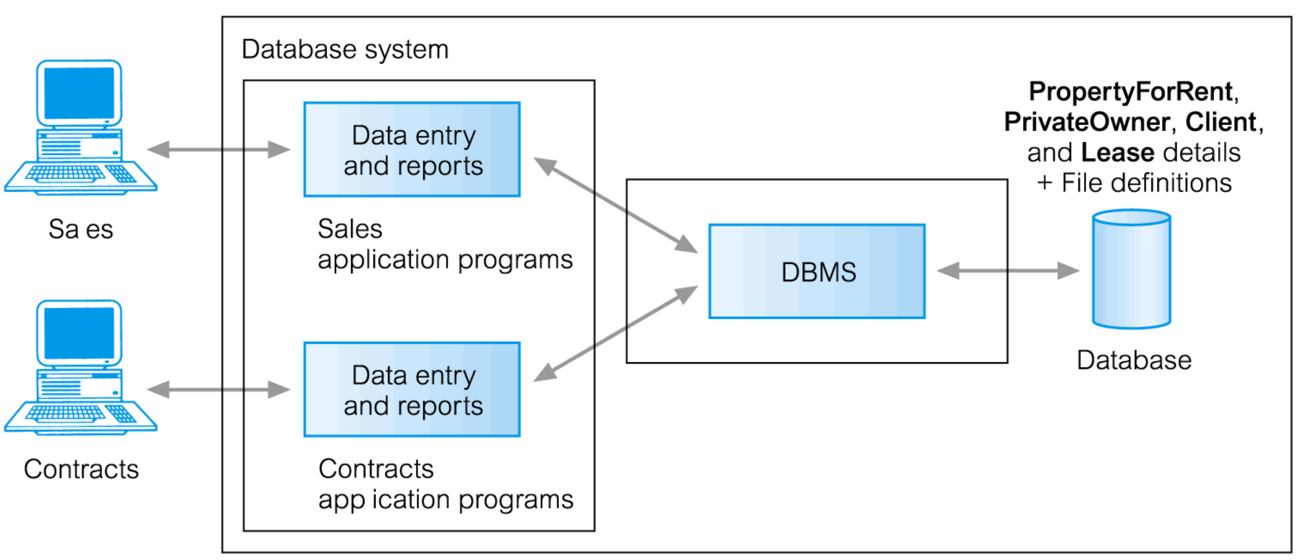

Database system
Sa es
Contracts
Data entry and reports
Sales application programs
Data entry and reports
Contracts app ication programs
DBMS
PropertyForRent, **PrivateOwner**, **Client**, and **Lease** details + File definitions
Database

Figure 1.7 Database processing

PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)
Lease (leaseNo, propertyNo, clientNo, paymentMethod, deposit, paid, rentStart, rentF nish)

Views

- Allows each user to have his or her own view of the database.
- A view is essentially some subset of the database.
- For example, we could set up a view that allows the Contracts Department to see only the data that they want to see for rental properties.

Views - Benefits

- Reduce complexity
- Provide a level of security
- Provide a mechanism to customize the appearance of the database
- Present a consistent, unchanging picture of the structure of the database, even if the underlying database is changed.

Components of DBMS Environment

We can identify five major components in the DBMS environment: hardware, software, data, procedures, and people, as illustrated in Figure 1.8.

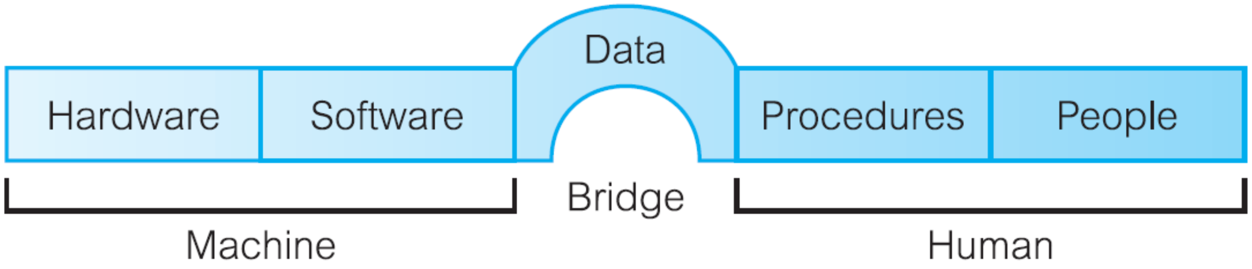


Figure 1.8 The DBMS environment.

Components of DBMS Environment

- **Hardware**
  - **Can range from a PC to a network of computers.**
  - **A simplified hardware configuration for DreamHome is illustrated in Figure 1.9.**

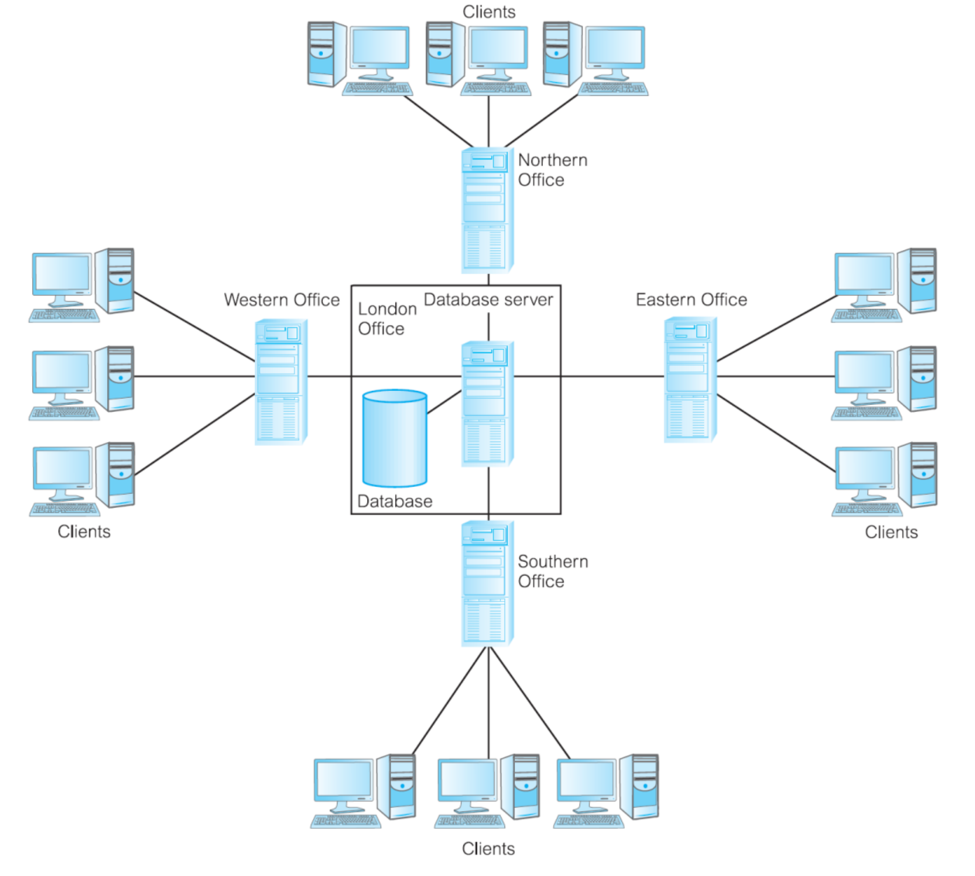


Figure 1.9 *DreamHome* hardware configuration.

Components of DBMS Environment

- **Software**
  - **DBMS, application programs, operating system, and network software (if necessary).**
  - **Application programs are written in a third-generation programming language (3GL), such as C, C++, C#, Java, Visual Basic, COBOL, Fortran, Ada, or Pascal, or a fourth-generation language (4GL), such as SQL, embedded in a third-generation language.**

Components of DBMS Environment

- **Data**
  - **Used by the organization and a description of this data called the *schema*.**
  - **In Figure 1.7, the *schema* consists of four files, or tables, namely: PropertyForRent, PrivateOwner, Client, and Lease.**
  - **The PropertyForRent table, for example, has eight fields, or attributes, namely: propertyNo, street, city, zipCode, type (the property type), rooms (the number of rooms), rent (the monthly rent), and ownerNo.**

Components of DBMS Environment

- **Procedures**
  - **Procedures refer to the instructions and rules that govern the design and use of the database.**
  - **The users the staff who need documented procedures on how to use or run the system.**
  - **These may consist of instructions on how to:**
    - Log on to the DBMS.
    - Use a particular DBMS facility or application program.
    - Start and stop the DBMS.
    - Make backup copies of the database.
    - Handle hardware or software failures.
- **People.**

Roles in the Database Environment

- **We can identify five types of people who participate in the DBMS environment:**
  - **Data Administrator (DA)**
  - **Database Administrator (DBA)**
  - **Database Designers (Logical and Physical)**
  - **Application Programmers**
  - **End Users (naive and sophisticated)**

Roles in the Database Environment

- **Data Administrator (DA)**
  - Is responsible for the management of the data resource, including database planning; development and maintenance of standards, policies and procedures; and conceptual/logical database design.
  - The DA consults with and advises senior managers, ensuring that the direction of database development will ultimately support corporate objectives.

Roles in the Database Environment

- **Database Administrator (DBA)**
  - Is responsible for the physical realization of the database, including physical database design and implementation, security and integrity control, maintenance of the operational system, and ensuring satisfactory performance of the applications for users.
  - In some organizations there is no distinction between DA and DBA.

Roles in the Database Environment

- **Database Designers (Logical and Physical)**
  - The **logical database designer** is concerned with identifying the data (that is, the entities and attributes), the relationships between the data, and the constraints on the data.
  - The **physical database designer** decides how the logical database design is to be physically realized.

    This involves:
    - Mapping the logical database design into a set of tables and integrity constraints.
    - Selecting specific storage structures and access methods for the data to achieve good performance;
    - Designing any security measures required on the data.

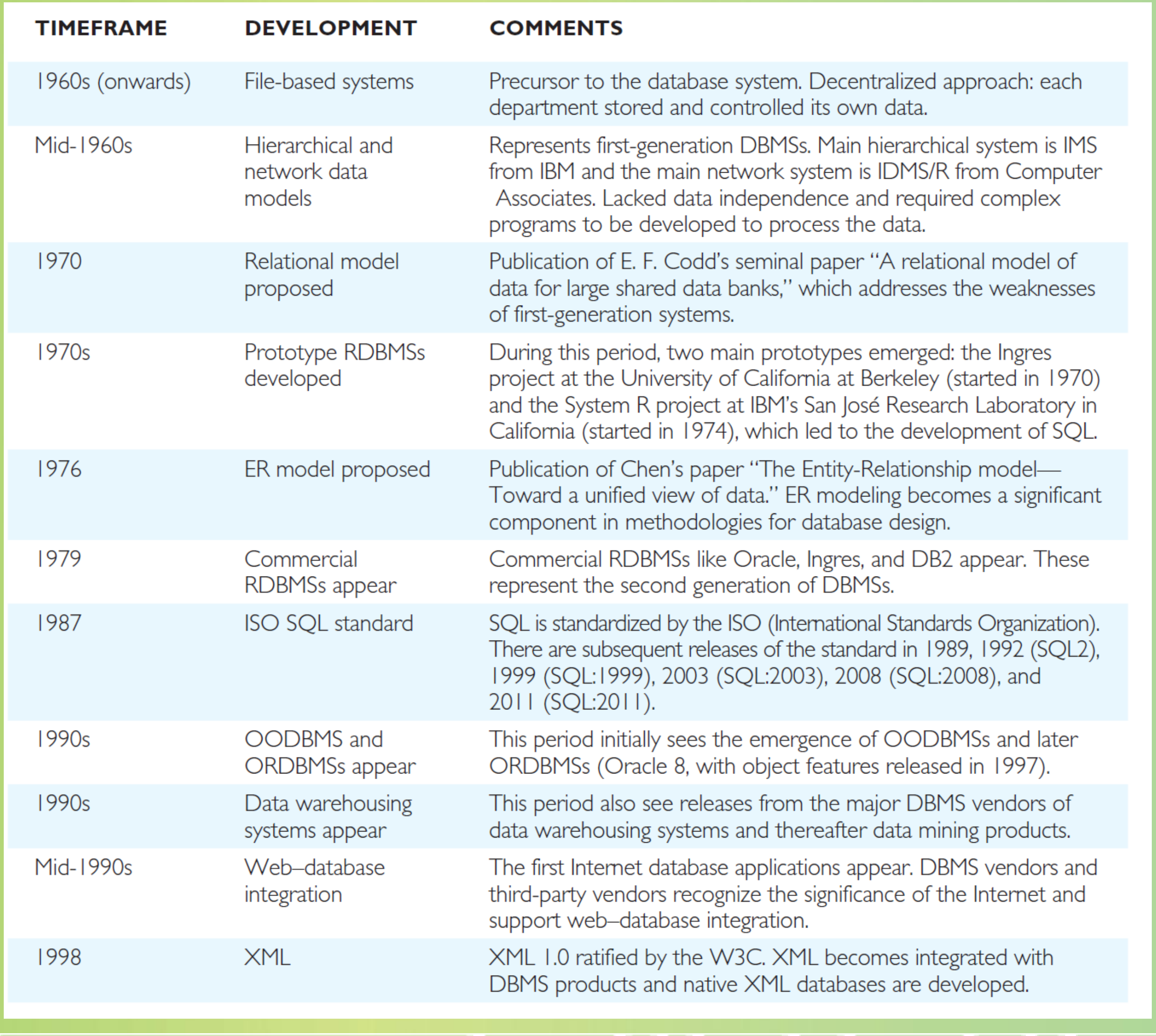
Roles in the Database Environment

- **Application Programmers**
  - Once the database has been implemented, the application programs that provide the required functionality for the end-users must be implemented.
  - This is the responsibility of the **application developers**.
  - Each program contains statements that request the DBMS to perform some operation on the database which includes retrieving data, inserting, updating, and deleting data.
  - The programs may be written in a third-generation or fourth-generation programming language, as discussed previously.

Roles in the Database Environment

- **End Users (naive and sophisticated)**
  - **Naïve users** are unaware of the DBMS.
    - They access the database through application programs.
    - For example, the checkout assistant at a supermarket uses a bar code reader to find out the price of the item. However, there is an application program that reads the bar code, looks up the price of the item, reduces the database field containing the number of such items in stock, and displays the price.
  - **Sophisticated users** are familiar with the structure of the database and the facilities offered by the DBMS.
    - They may use a high-level query language such as SQL to perform the required operations.
    - They even can write application programs for their own use.

History of Database Systems



TIMEFRAME	DEVELOPMENT	COMMENTS
1960s (onwards)	File-based systems	Precursor to the database system. Decentralized approach: each department stored and controlled its own data.
Mid-1960s	Hierarchical and network data models	Represents first-generation DBMSs. Main hierarchical system is IMS from IBM and the main network system is IDMS/R from Computer Associates. Lacked data independence and required complex programs to be developed to process the data.
1970	Relational model proposed	Publication of E. F. Codd's seminal paper "A relational model of data for large shared data banks," which addresses the weaknesses of first-generation systems.
1970s	Prototype RDBMSs developed	During this period, two main prototypes emerged: the Ingres project at the University of California at Berkeley (started in 1970) and the System R project at IBM's San José Research Laboratory in California (started in 1974), which led to the development of SQL.
1976	ER model proposed	Publication of Chen's paper "The Entity-Relationship model—Toward a unified view of data." ER modeling becomes a significant component in methodologies for database design.
1979	Commercial RDBMSs appear	Commercial RDBMSs like Oracle, Ingres, and DB2 appear. These represent the second generation of DBMSs.
1987	ISO SQL standard	SQL is standardized by the ISO (International Standards Organization). There are subsequent releases of the standard in 1989, 1992 (SQL2), 1999 (SQL:1999), 2003 (SQL:2003), 2008 (SQL:2008), and 2011 (SQL:2011).
1990s	OODBMS and ORDBMSs appear	This period initially sees the emergence of OODBMSs and later ORDBMSs (Oracle 8, with object features released in 1997).
1990s	Data warehousing systems appear	This period also see releases from the major DBMS vendors of data warehousing systems and thereafter data mining products.
Mid-1990s	Web–database integration	The first Internet database applications appear. DBMS vendors and third-party vendors recognize the significance of the Internet and support web–database integration.
1998	XML	XML 1.0 ratified by the W3C. XML becomes integrated with DBMS products and native XML databases are developed.

Advantages of DBMSs

- **Control of data redundancy**
- **Data consistency**
- **More information from the same amount of data**
- **Sharing of data**
- **Improved data integrity**
- **Improved security**
- **Enforcement of standards**
- **Economy of scale**

Advantages of DBMSs

- **Balance conflicting requirements**
- **Improved data accessibility and responsiveness**
- **Increased productivity**
- **Improved maintenance through data independence**
- **Increased concurrency**
- **Improved backup and recovery services**

Disadvantages of DBMSs

- **Complexity**
- **Size**
- **Cost of DBMS**
- **Additional hardware costs**
- **Cost of conversion**
- **Performance**
- **Higher impact of a failure**