Introduction to Databases

Examples of Database Applications

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

Database Terminology

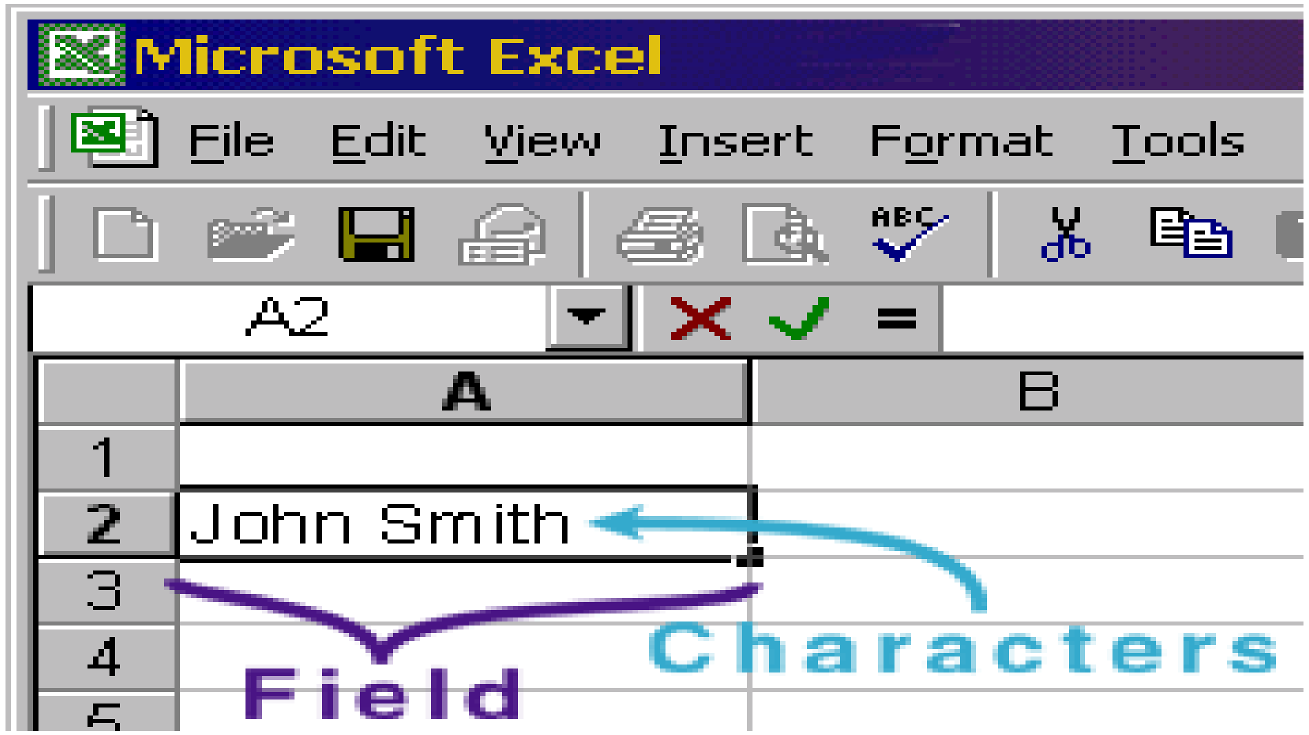
- **Character**
  - A character is the most basic element of data that can be observed and manipulated. Behind it are the invisible data elements we call bits and bytes, referring to physical storage elements used by the computer hardware. A character is a single symbol such as a digit, letter, or other special character (e.g., $, #, and ?).

Database Terminology

- **Field**
  - A field contains an item of data; that is, a character, or group of characters that are related. For instance, a grouping of related text characters such as "John Smith" makes up a name in the name field.

Database Terminology

- **Field**

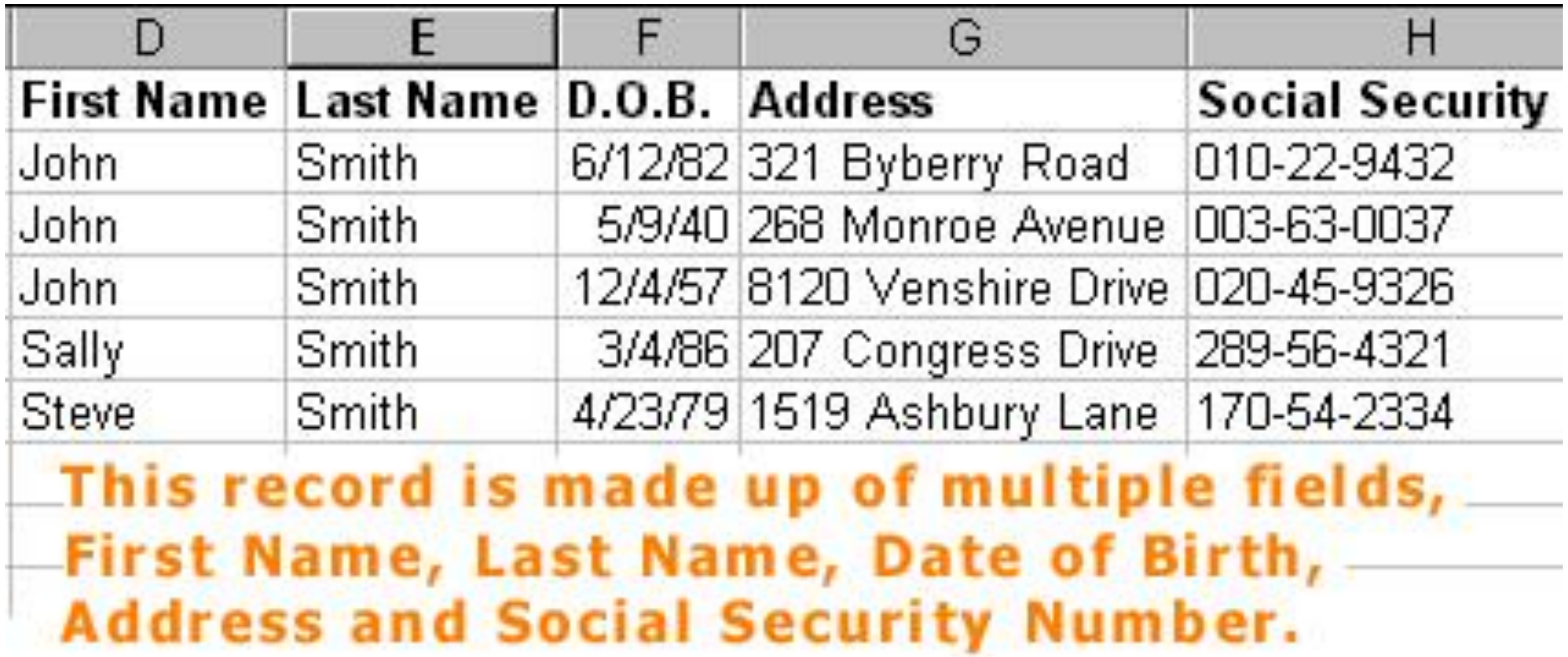


Database Terminology

- **Record**
  - A record is composed of a group of related fields. As another way of saying it, a record contains a collection of attributes related to an entity such as a person or product. A payroll record would contain the name, address, social security number, and title of each employee.

Database Terminology

- **Record**



D	E	F	G	H
First Name	**Last Name**	**D.O.B.**	**Address**	**Social Security**
John	Smith	6/12/82	321 Byberry Road	010-22-9432
John	Smith	5/9/40	268 Monroe Avenue	003-63-0037
John	Smith	12/4/57	8120 Venshire Drive	020-45-9326
Sally	Smith	3/4/86	207 Congress Drive	289-56-4321
Steve	Smith	4/23/79	1519 Ashbury Lane	170-54-2334

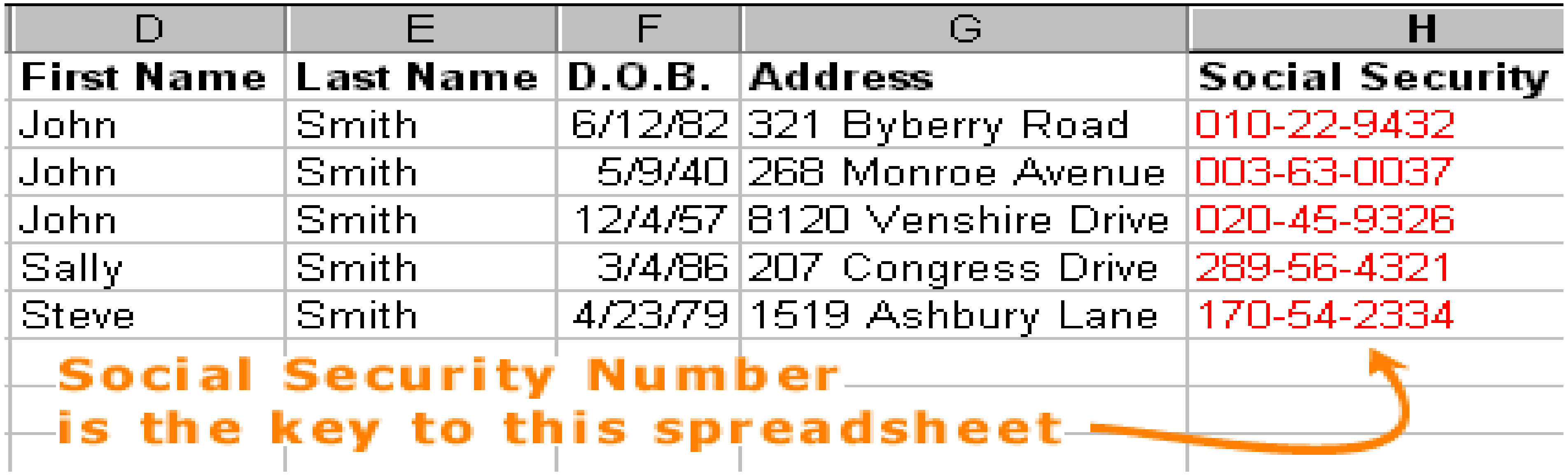
This record is made up of multiple fields, First Name, Last Name, Date of Birth, Address and Social Security Number.

Database Terminology

- **Key**
  - In order to track and analyse data effectively, each record requires a unique identifier or what is called a key. The key must be completely unique to a particular record just as each individual has a unique social security number assigned to them.

Database Terminology

- **Key**



D	E	F	G	H
First Name	**Last Name**	**D.O.B.**	**Address**	**Social Security**
John	Smith	6/12/82	321 Byberry Road	010-22-9432
John	Smith	5/9/40	268 Monroe Avenue	003-63-0037
John	Smith	12/4/57	8120 Venshire Drive	020-45-9326
Sally	Smith	3/4/86	207 Congress Drive	289-56-4321
Steve	Smith	4/23/79	1519 Ashbury Lane	170-54-2334

Social Security Number is the key to this spreadsheet

Database Terminology

- **Database File**
  - As we move up the ladder, a database file is defined as a collection of related records. A database file is sometimes called a *table*. A file may be composed of a complete list of individuals on a mailing list, including their addresses and telephone numbers.

Database Terminology

- **Database File**

Related records combine to create a database file

Database Terminology

- **Database**
  - Organisations and individuals use databases to bring independent sources of data together and store them electronically. Thus, a database is composed of related files that are consolidated, organised and stored together. One collection of related files might pertain to employee information.

Database Terminology

- **Database Management System**
  - Database management systems are used to access and manipulate data in a database. A database management system is a software package that enables users to edit, link, and update files as needs dictate.

File-Based Systems

- Collection of application programs that perform services for the end users (e.g. reports).

- Each program defines and manages its own data.

File-Based Processing

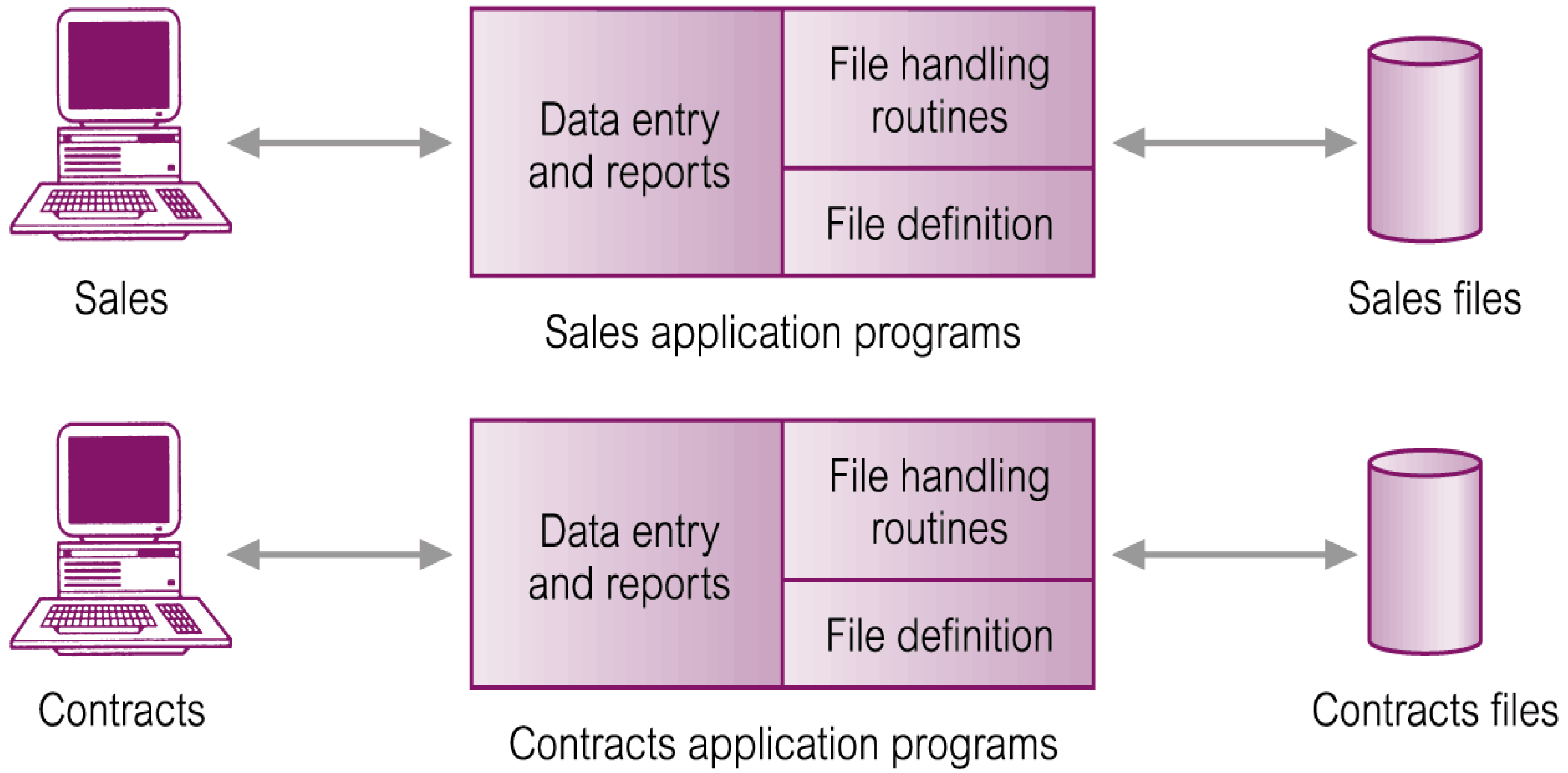


Figure 1.5 File-based processing.

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)

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 space and potentially different values and/or different formats for the same item.

Limitations of File-Based Approach

- **Data dependence**
  - File structure is defined in the program code.
- **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

- Shared collection of logically related data (and a description of this data), designed to meet the information needs of an organisation.

- System catalog (metadata) provides description of data to enable program–data independence.

- Logically related data comprises entities, attributes, and relationships of an organisation’s information.

Database Management System (DBMS)

- A software system that enables users to define, create, maintain, and control access to the database.
- (Database) application program: a computer program that interacts with database by issuing an appropriate request (SQL statement) to the DBMS.

Database Management System (DBMS)

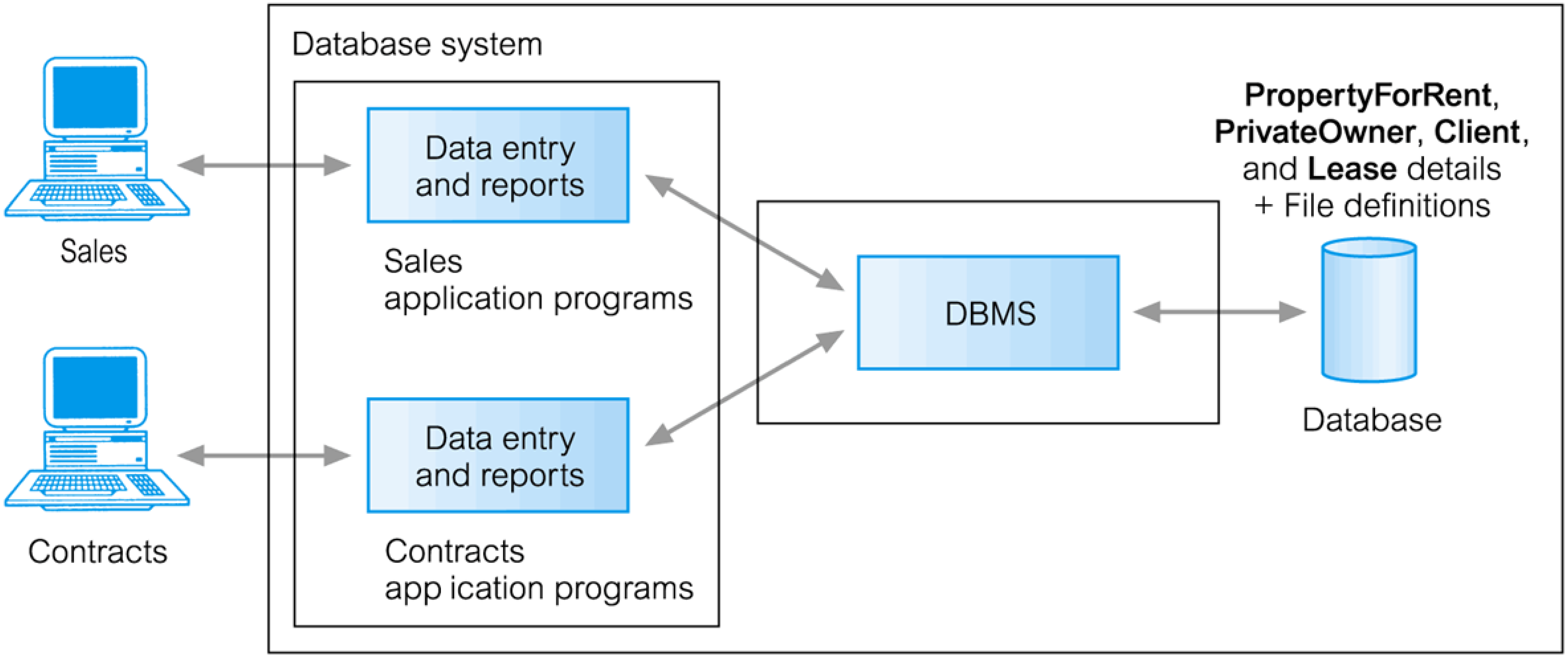
- **Data definition language (DDL).**
  - Permits specification of data types, structures and any data constraints.
  - All specifications are stored in the database.

- **Data manipulation language (DML).**
  - General enquiry facility (query language) of the data.

Database Management System (DBMS)

- **Controlled access to database may include:**
  - a security system;
  - an integrity system;
  - a concurrency control system;
  - a recovery control system;
  - a user-accessible catalog.

Database Management System (DBMS)


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

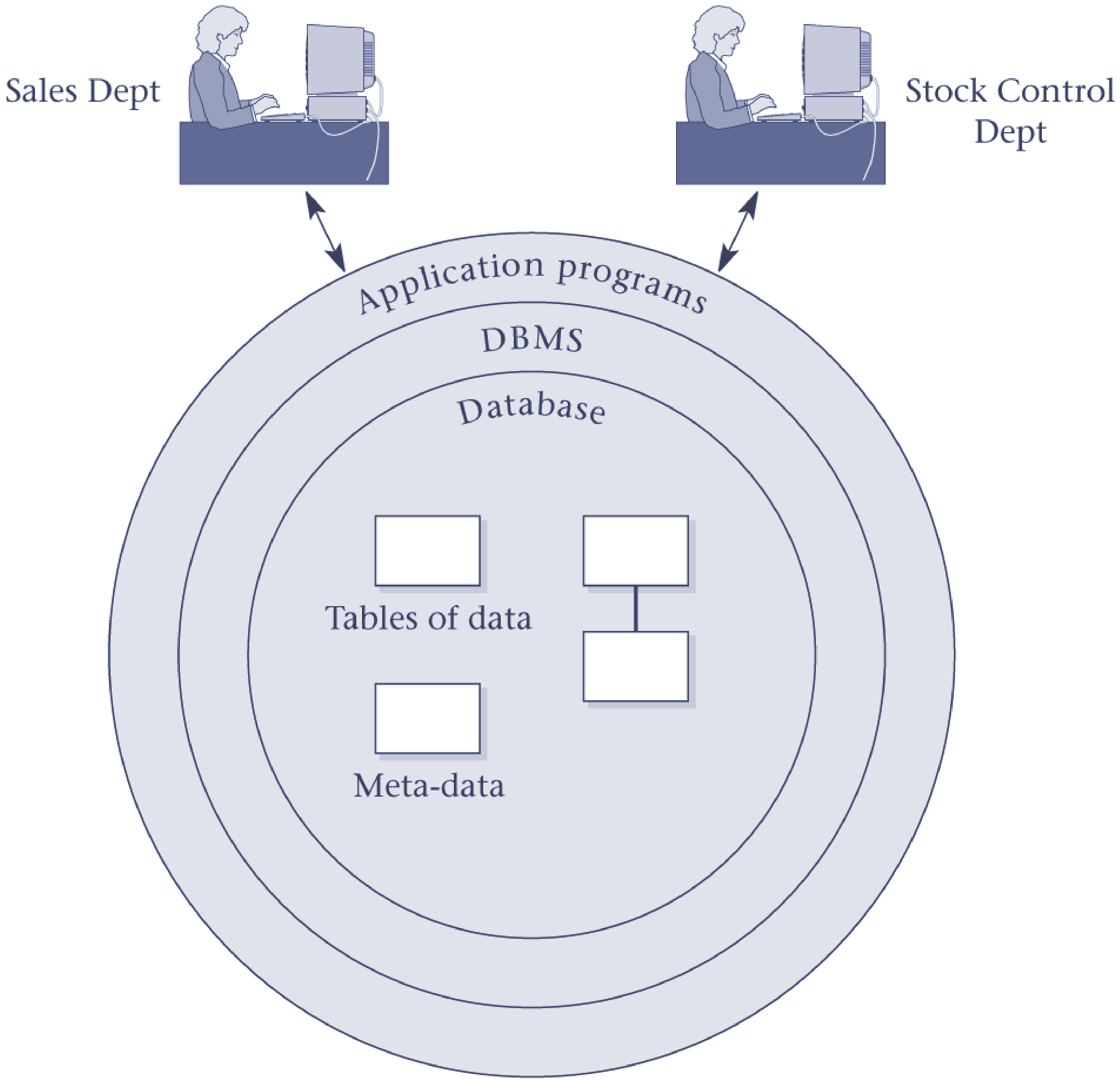
PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)
Lease (leaseNo, propertyNo, clientNo, paymentMethod, deposit, paid, rentStart, rentF nish)

Database Approach

- Users interact with the database through a number of application programs that are used to create and maintain the database and to generate information. These programs can be conventional batch applications or, more typically they will be online applications.

Database Approach

- The Sales and Stock Control 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.
- The physical structure and storage of the data are managed by the DBMS.



Views

- Allows each user to have his or her own view of the database.
- A view is essentially some subset of the database.

Views - Benefits

- Reduce complexity.
- Provide a level of security.
- Provide a mechanism to customise 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

- Hardware
- Software
- Data
- Procedures
- Data Access Language
- People

Components of DBMS Environment

- **Hardware**
  - The DBMS and the applications require hardware to run. The hardware can range from a single PC to a single mainframe, a network of computers, and database servers. The particular hardware depends on the organisation’s requirements and the DBMS used.
  - A DBMS requires a minimum amount of main memory and disk space to run.

Components of DBMS Environment

- **Software**
  - The main component of a DBMS is the software. It is the set of programs used to handle the database and to control and manage the overall computersed database

    1. DBMS software itself, is the most important software component in the overall system

    2. Operating system including network software being used in network, to share the data of the database among multiple users.

Components of DBMS Environment

- **Software continued…**

  3. Application programs developed in programming languages such as Java, C++, PHP that are used to access the database in a database management system. Each program contains embedded SQL statements that request the DBMS to perform operations on the database. The operations may include retrieving, updating, deleting data etc…

Components of DBMS Environment

- **Data**
  - Data is the most important component of the DBMS. The main purpose of DBMS is to process the data. In DBMS, databases are defined, constructed and then data is stored, updated and retrieved to and from the databases. The database contains both the actual (or operational) data and the metadata (data about data or description about data).
  - The structure of the database is called the **schema**.

Components of DBMS Environment

- **Procedures**
  - Procedures refer to the instructions and rules that help to design the database and to use the DBMS. The users that operate and manage the DBMS require documented procedures on how to use or run the database management system.

Components of DBMS Environment

- **Procedures continued…**
  - These may include:
    1. Procedure to install a new DBMS.
    2. To log on to the DBMS.
    3. To use the DBMS or application program.
    4. To make backup copies of database.
    5. To change the structure of database.
    6. To generate the reports of data retrieved from database.

Components of DBMS Environment

- **Database Access Language**
  - The database access language is used to access the data to and from the database.
  - The users use the database access language (through a GUI or command line) to enter new data, change the existing data in database and to retrieve required data from databases.
  - The DBMS translates the user commands and sends it to a specific part of the DBMS called the Database Jet Engine.

Components of DBMS Environment

- **Database Access Languages continued…**
    - The database engine generates a set of results according to the commands submitted by the user, converts these into a user readable form called an Inquiry Report and then displays them on the screen.
    - The administrators may also use the database access language to create and maintain the databases.
      The most popular database access language is SQL (Structured Query Language). Relational databases are required to have a database query language.

Components of DBMS Environment

- **Users**
  - The users are the people who manage the databases and perform different operations on the databases in the database system. There are three kinds of people who play different roles in database system.

Components of DBMS Environment

- **Users continued…**

  - Database Administrators
    - A person who is responsible for managing the overall database management system is called database administrator or simply DBA.

  - Application Programmers
    - The people who write application programs in programming languages (such as PHP, Java, or C++) to interact with databases are called Application Programmers. These Application Programmes will include SQL code that will interact with the database and code that will handle the results returned from the interaction.

Components of DBMS Environment

- **Users continued…**
  - End-Users
    - The end-users are the people who interact with database management system to perform different operations on database such as retrieving, updating, inserting, deleting data etc.

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.

- **Database Administrator (DBA)** is responsible for the physical realisation of the database, including:
  - physical database design and implementation,
  - security, and integrity control,
  - maintenance of the operational system,
  - ensuring satisfactory performance of the applications for end users.

Roles in the Database Environment

- Database Designers (Logical and Physical).
- Application Programmers.
- End Users (naive and sophisticated).

History of Database Systems

- **First-generation**
  - Hierarchical and Network.

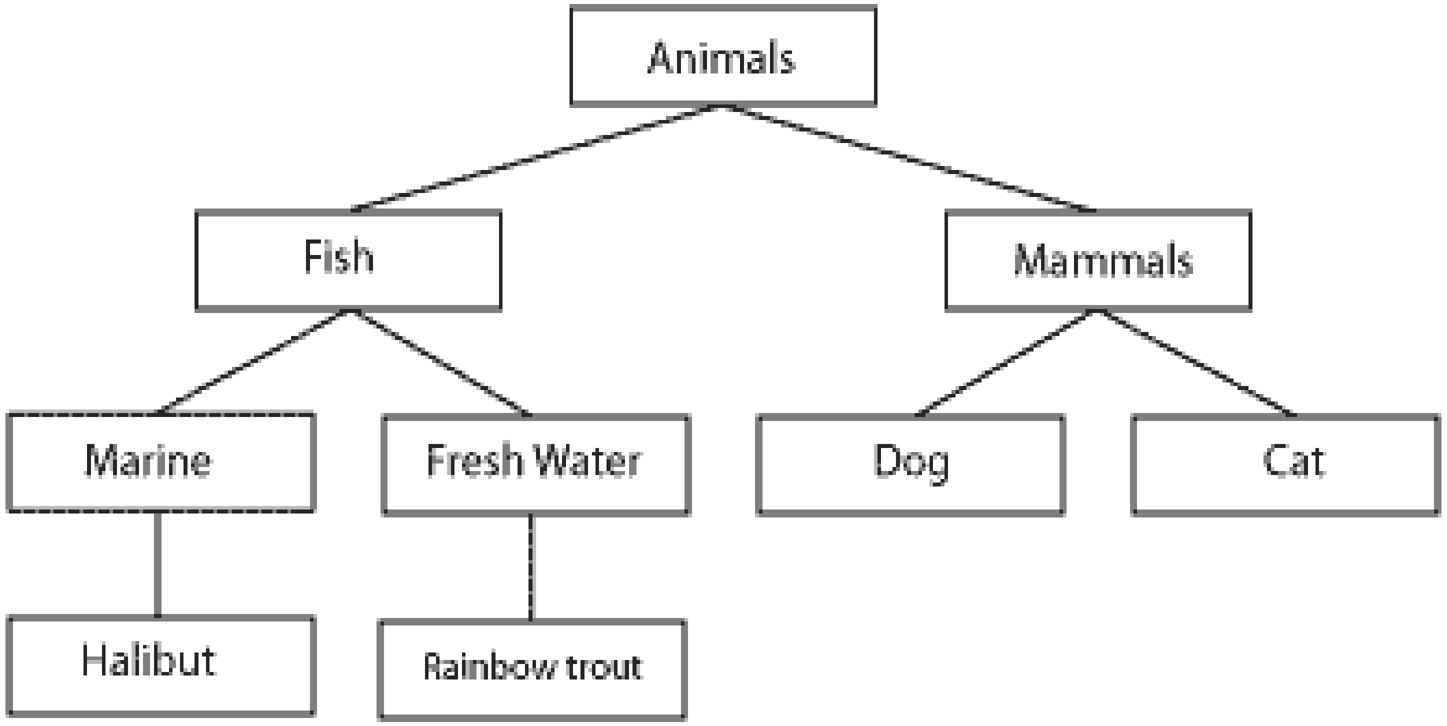
- **Second generation**
  - Relational

- **Third generation**
  - Object-Relational.
  - Object-Oriented.
  - NoSQL.
  - NewSQL.

Hierarchical DBMS

- **Hierarchical Databases (DBMS),** commonly used on mainframe computers, have been around for a long time. It is one of the oldest methods of organising and storing data. Hierarchical databases arrange data in a "tree" structure, which is similar to folders and files on a computer. Just as a file on a computer sits in one folder, every record in the database has one "parent." Hierarchically arranged data is often described as having only parent/child relationships.

Hierarchical DBMS



Hierarchical DBMS

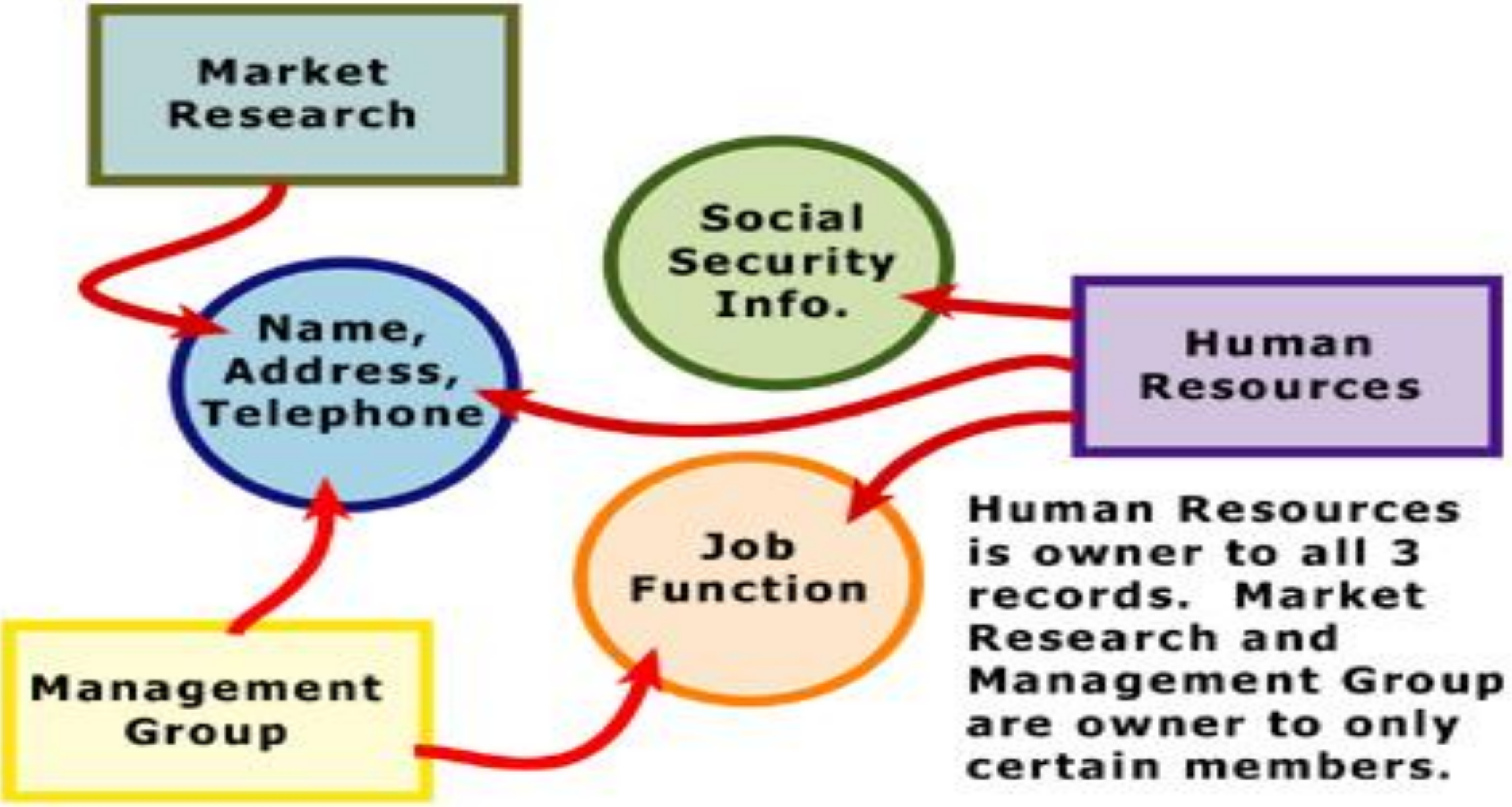
- Based on this analogy, the parent record at the top of the pyramid is called the **root** record. A child record always has only one parent record to which it is linked. In contrast, a parent record may have more than one child record linked to it. Hierarchical databases work by moving from the top down. A record search is conducted by starting at the top of the pyramid and working down through the tree from parent to child until the appropriate child record is found. Furthermore, each child can also be a parent with children underneath it.

- Example: IBM's IMS system.

Network DBMS

- **Network databases** are similar to hierarchical databases by also having a hierarchical structure. There are a few key differences, however. Instead of looking like an upside-down tree, a network database looks more like a cobweb or interconnected network of records. In network databases, children are called **members** and parents are called **owners**. The most important difference is that each child or member can have more than one parent (or owner).

Network DBMS



Human Resources is owner to all 3 records. Market Research and Management Group are owner to only certain members.

Network DBMS

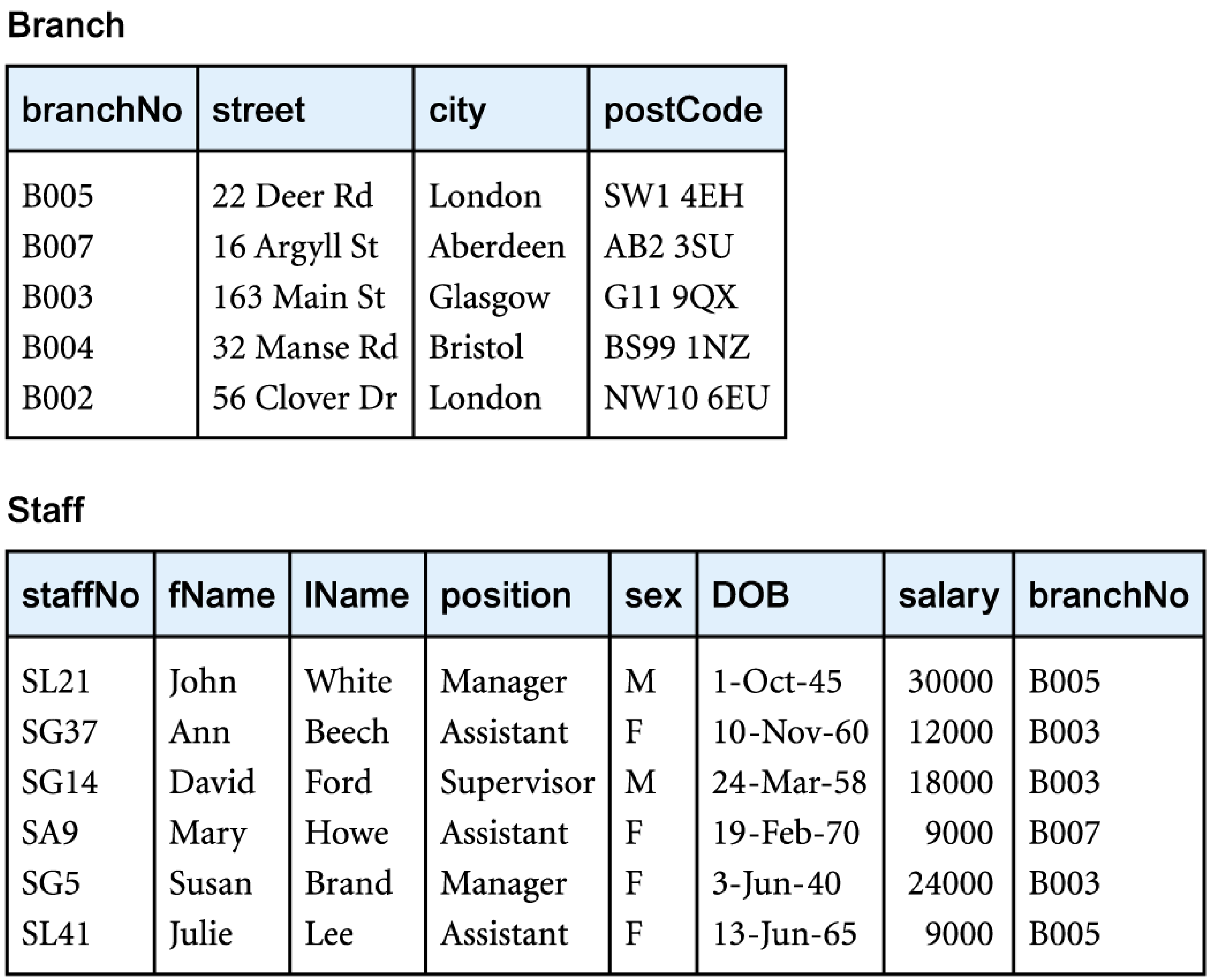
- Like hierarchical databases, network databases are principally used on mainframe computers. Since more connections can be made between different types of data, network databases are considered more flexible.

- Example: IDMS.

Relational DBMS

- In **relational databases**, the relationship between data files is relational, not hierarchical. Relational databases connect data in different files by using common data elements or a key field. Data in relational databases is stored in different tables, each having a key field that uniquely identifies each row. Relational databases are more flexible than either the hierarchical or network database structures. In relational databases, tables or files filled with data are called **relations**, **tuples** designates a row or record, and columns are referred to as **attributes** or fields. Tables are linked through **foreign keys**.

Relational DBMS



Branch

branchNo	street	city	postCode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

Object-oriented DBMS and object-relational DBMS

- Traditional database models, including the relational model, have worked well in traditional business situations. However, they fall short in areas such as engineering design and manufacturing, scientific experiments, telecommunications, geographic information systems, and multimedia.

- The object-oriented database model was developed to meet the needs of these applications.

- An object-oriented database uses “objects,” software written in small, reusable chunks, as elements within database files.

Object-oriented DBMS and object-relational DBMS

- An object consists of :
  - data in any form, including graphics, audio, and video, and
  - instructions on the action to be taken on the data.
- Examples of object-oriented databases are FastObjects, Objectivity DB and KE Texpress.

NoSQL

- **NoSQL** (Not only SQL) encompasses a wide variety of different database technologies that were developed in response to a rise in the volume of data stored about users, objects and products, the frequency in which this data is accessed, and performance and processing needs.

- NoSQL are next generation databases mostly addressing some of the points:
  being **non-relational, distributed,**
  open-source and **horizontally scalable**.

NoSQL

- NoSQL database management systems store data in a variety of formats, chief among them being document store, graph store, and key-value store.
- NoSQL databases are increasingly used in big data and real-time web applications.

NewSql

- **NewSQL** is a class of modern relational database management system that seeks to provide the same scalable performance of NoSQL systems for online transaction processing (read-write) workloads while still maintaining the ACID (Atomicity, Consistency, Isolation, Durability) guarantees of a traditional single-node database system.

Advantages of DBMSs

- **Control of data redundancy:** The database approach attempts to eliminate redundancy by integrating the files so that multiple copies of the same data are not stored.

- **Data consistency:** By eliminating or controlling redundancy, we reduce the risk of inconsistencies occurring.

- **More information from the same amount of data:** With the integration of the operational data, it may be possible for the organisation to derive additional information from the same data.

Advantages of DBMSs

- **Sharing of data:** The database is owned by the entire organisation and can be shared by authorised users.
- **Improved data integrity:** Database integrity is enforced using constraints.
- **Improved security:** The DBA defines the database security and the DBMS enforces it.
- **Enforcement of standards:** The DBA defines and the DBMS enforces the necessary standards.

Advantages of DBMSs

- **Economy of scale:** Combining all the organisation’s operational data into one database and creating a set of applications that work on this one source of data can result in cost savings.

- **Balance conflicting requirements:** The database is under the control of the DBA, and he/she can make decisions about the design and operational use of the database that provide the best use of resources for the organisation as a whole.

Advantages of DBMSs

- **Improved data accessibility and responsiveness:** Data is directly accessible to the end users.

- **Increased productivity:** The DBMS provides all the low-level file-handling routines that are typical in application programs.

- **Improved maintenance through data independence:** A DBMS separates the data descriptions from the applications thereby making applications immune to changes in the data descriptions.

Advantages of DBMSs

- **Increased concurrency:** Many DBMSs manage concurrent database access and ensure concurrency problems do not occur.

- **Improved backup and recovery services:** Modern DBMSs provide facilities for backup and recovery services.

Disadvantages of DBMSs

- **Complexity:** The provision of the functionality makes the DBMS a complex piece of software.

- **Size:** The complexity and breadth of functionality makes the DBMS an extremely large piece of software occupying many megabytes of disk space and requiring substantial amounts of memory to run efficiently.

- **Cost of DBMS:** The total cost should be calculated as a combination of the license cost of the DBMS, the license cost of any required supporting software, the cost of database professionals to program, support and administer the DBMS, and the cost of the computing resources required to operate the DBMS.

Disadvantages of DBMSs

- **Additional hardware costs:** The disk storage requirements for the DBMS and the database may necessitate the purchase of additional storage space.

- **Cost of conversion:** As well as the DBMS, and storage costs, there is development costs, and staff training.

- **Performance:** Some applications may not run as fast as on a dedicated file system.

- **Higher impact of a failure:** The centralisation of resources increases the vulnerability of the system.