Chapter 2

Database Environment

Chapter 2 - Objectives

- Purpose of three-level database architecture.
- Contents of external, conceptual, and internal levels.
- Purpose of external/conceptual and conceptual/internal mappings.
- Meaning of logical and physical data independence.
- Distinction between DDL and DML.
- A classification of data models.

Chapter 2 - Objectives

- Purpose/importance of conceptual modeling.
- Typical functions and services a DBMS should provide.
- Function and importance of system catalog.
- Software components of a DBMS.
- Meaning of client–server architecture and advantages of this type of architecture for a DBMS.
- Function and uses of Transaction Processing Monitors.

Objectives of Three-Level Architecture

- **All users should be able to access same data, but have a different customized view of data.**

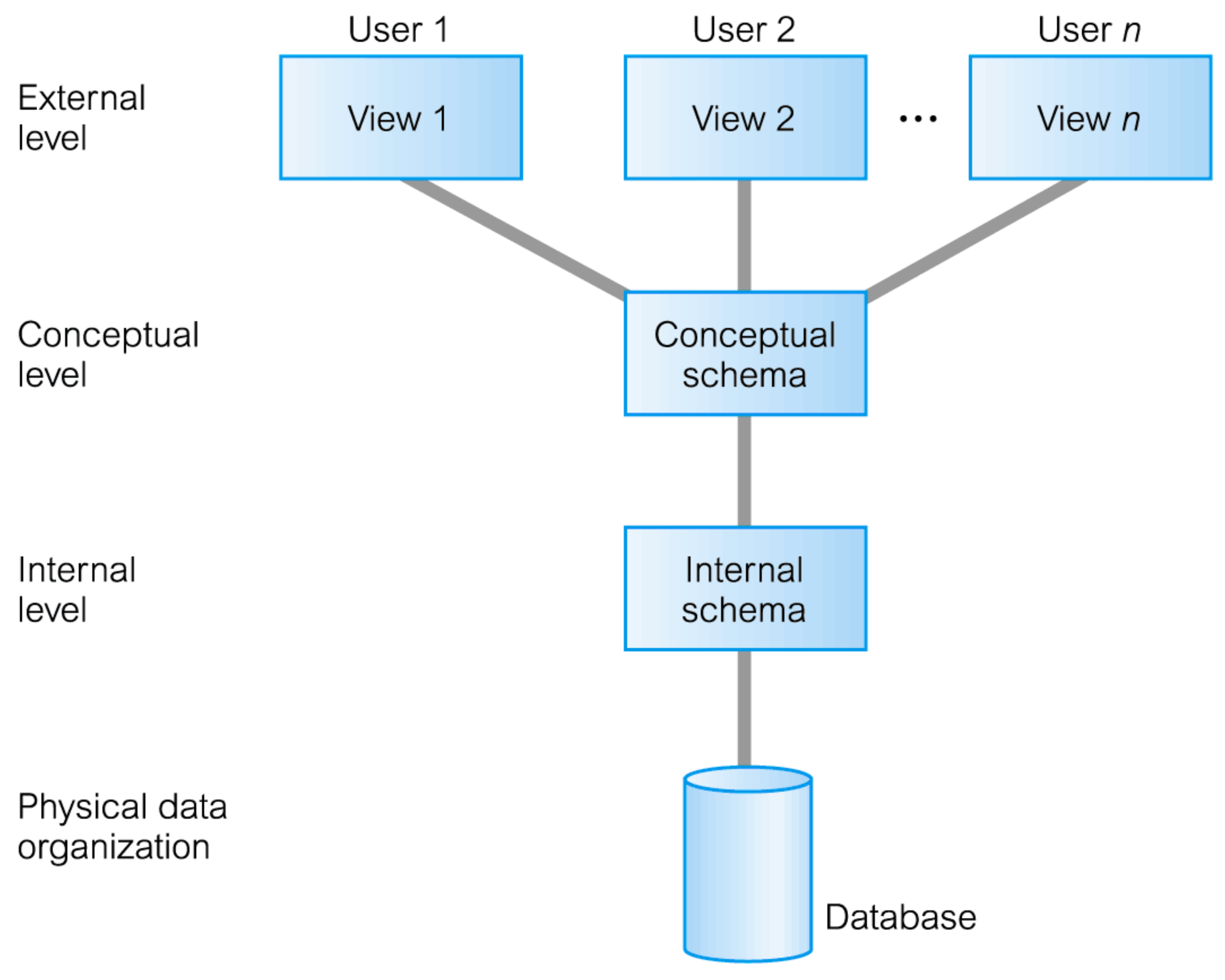
- **Each user should be able to change the way he or she views the data, and this change should not affect other users.**

- **Users should not need to know physical database storage details.**

Objectives of Three-Level Architecture Cont.

- DBA should be able to change database storage structures without affecting the users' views.
- DBA should be able to change conceptual structure of database without affecting all users.
- The internal structure of the database should be unaffected by the changes to the physical aspects of storage, such as change over to a new storage devices.

ANSI-SPARC Three-Level Architecture



Pearson Education © 2014

ANSI-SPARC Three-Level Architecture Cont.

- External Level
  - Users' view of the database.
  - Describes that part of database that is relevant to a particular user. It excludes irrelevant data as well as data which the user in not authorized to access.
- Conceptual Level
  - Describes what data is stored within the whole database and how the data is inter-related.
  - The conceptual level does not specify how the data is physically stored.

ANSI-SPARC Three-Level Architecture Cont.

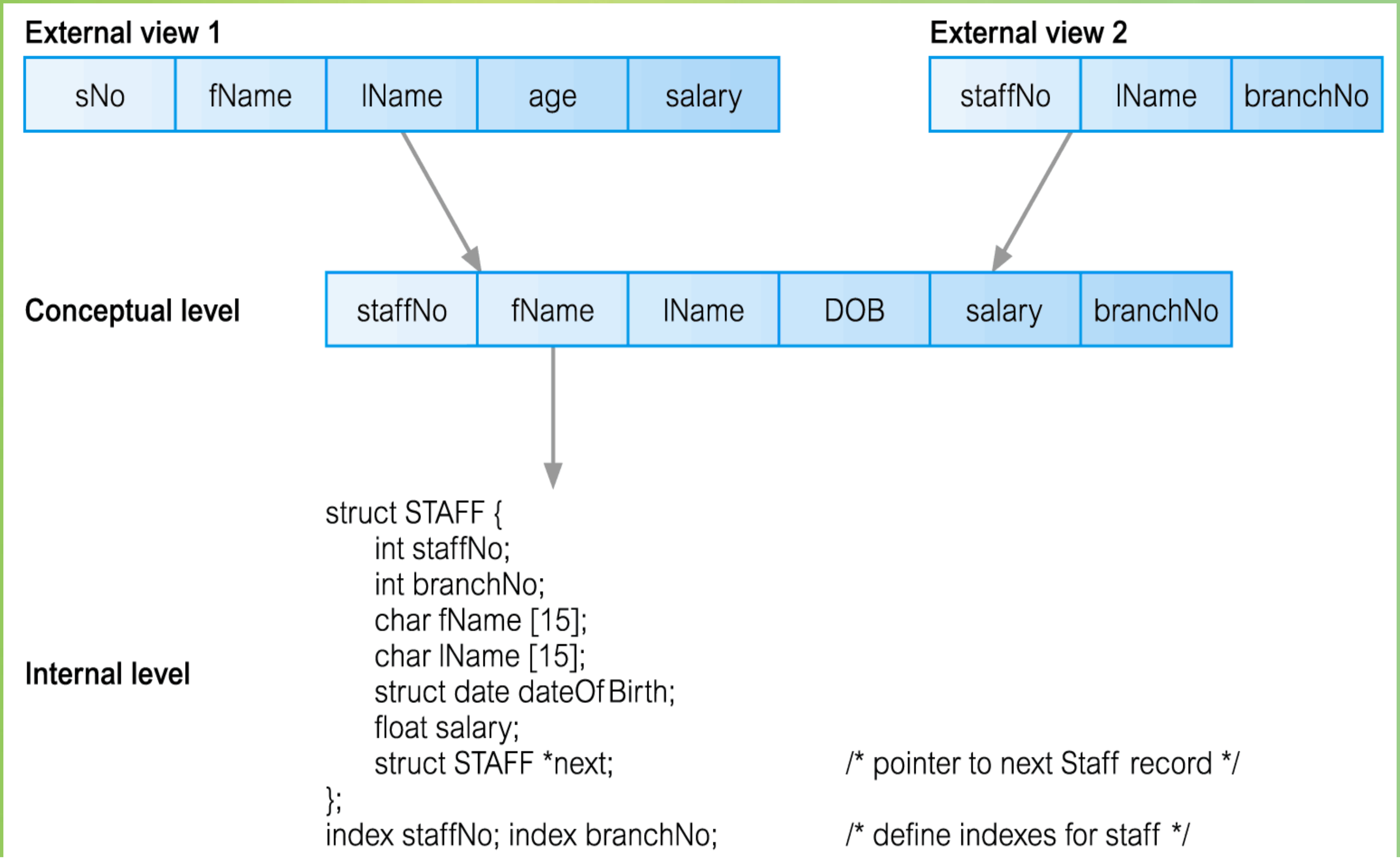
Some impotant facts about conceptual level are:

- DBA works at this level.
- Describes the structure of all users.
- Only DBA can define this level.
- Global view of database.
- Independent of hardware and software.

ANSI-SPARC Three-Level Architecture Cont.

- Internal Level
  - Involves how the database is physically represented on the computer system.
  - It describes how the data is actually stored in the database and on the computer hardware.

Differences between Three Levels of ANSI-SPARC Architecture



External view 1

sNo	fName	lName	age	salary

External view 2

staffNo	lName	branchNo

Conceptual level

staffNo	fName	lName	DOB	salary	branchNo

Internal level

```
struct STAFF {
      int staffNo;
      int branchNo;
      char fName [15];
      char lName [15];
      struct date dateOfBirth;
      float salary;
      struct STAFF *next;                /* pointer to next Staff record */
};
index staffNo; index branchNo;           /* define indexes for staff */
```

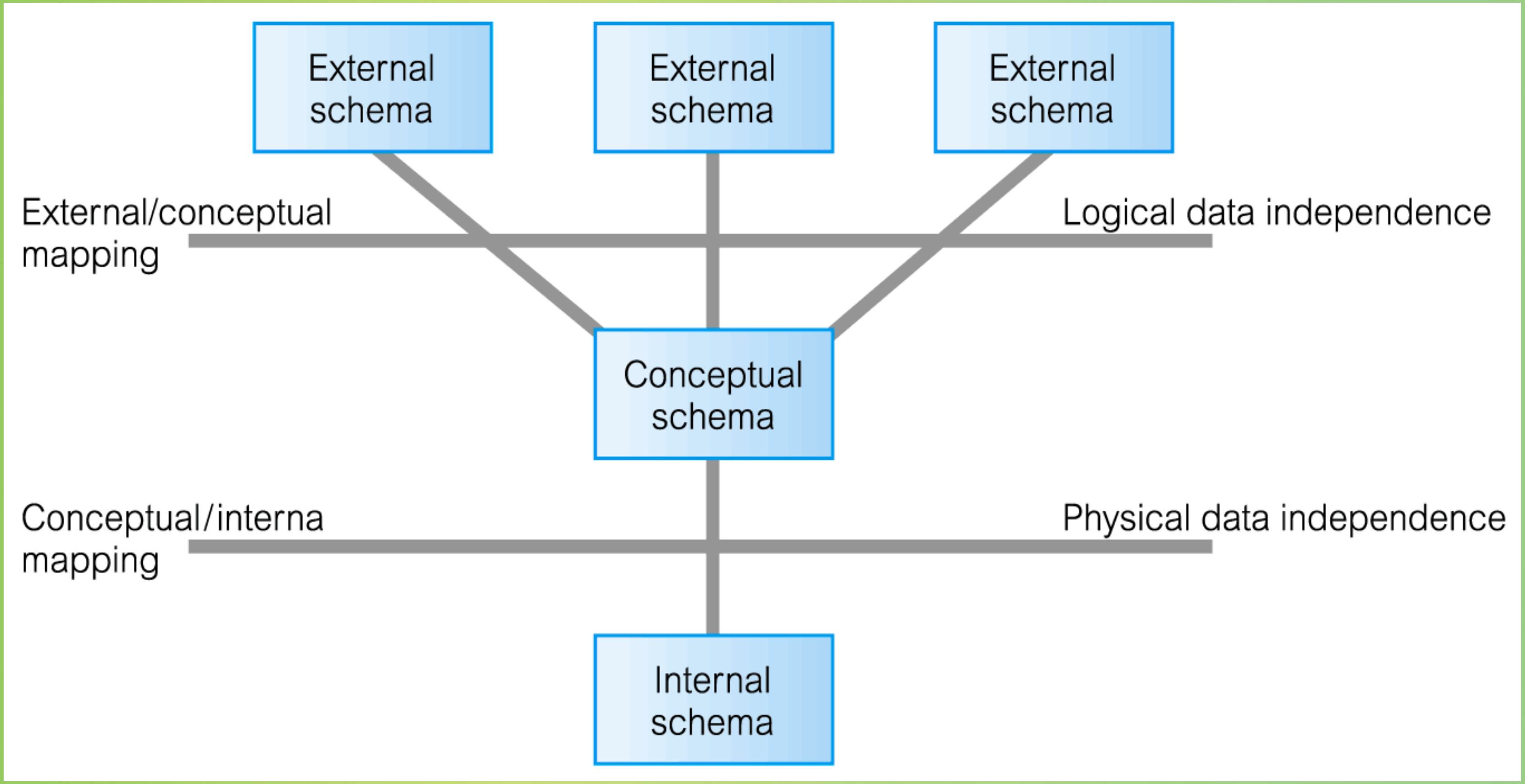
Data Independence

- Logical Data Independence
  - Refers to immunity of external schemas to changes in conceptual schema.
  - Conceptual schema changes (e.g. addition/removal of entities).
  - Should not require changes to external schema or rewrites of application programs.
  - We can simply say, external schema is independent to conceptual schema.

Data Independence Cont.

- **Physical Data Independence**
  - **Refers to immunity of conceptual schema to changes in the internal schema.**
  - **Internal schema changes (e.g. using different file organizations, storage structures/devices).**
  - **Should not require change to conceptual or external schemas.**
  - **We can simply say, external schema is independent to internal schema.**

Data Independence and the ANSI-SPARC Three-Level Architecture


External schema
External schema
External schema
External/conceptual mapping
Logical data independence
Conceptual schema
Conceptual/interna mapping
Physical data independence
Internal schema

Database Languages

- Data Definition Language (DDL)
  - DDL is a standard for commands that define the different in a database.
  - Allows the DBA or user to create, modify, and remove database objects such as tables, indexes, and users.
  - Plus any associated integrity and security constraints.

Database Languages Cont.

- **Data Manipulation Language (DML)**
  - **DML is a family of syntax elements similar to a computer programming language used for selecting, inserting, deleting and updating data in database. There are two type of DML:**

Database Languages Cont.

- Procedural DML (3GL):
  - This languages tell the computer what to do and how to do.
  - It is very difficult to learn as compared to non-procedural.
  - It is difficult to debug.
  - It requires large number of procedural instruction.
  - It is normally used by processional programmer.
  - It is typically file-oriented.
  - Procedural languages provides many programming capabilities.

Database Languages Cont.

- Non-Procedural DML (4GL):
  - This languages tell the computer what to do, not how to do.
  - It is very easy learn.
  - It is very easy to debug.
  - It requires few non-procedural instruction.
  - It can be used professional and technical users.
  - It is typically database oriented.
  - Non-Procedural languages provides less programming capabilities.

Data Model

- Database model shows the logical structure of a database including the relationship and constraints that determine how data can be stored and accessed.
- Individual database models are designed on the rules and concepts of whichever broader data model the designers adopt.
- Most data models can be represented by an accompanying database diagram.

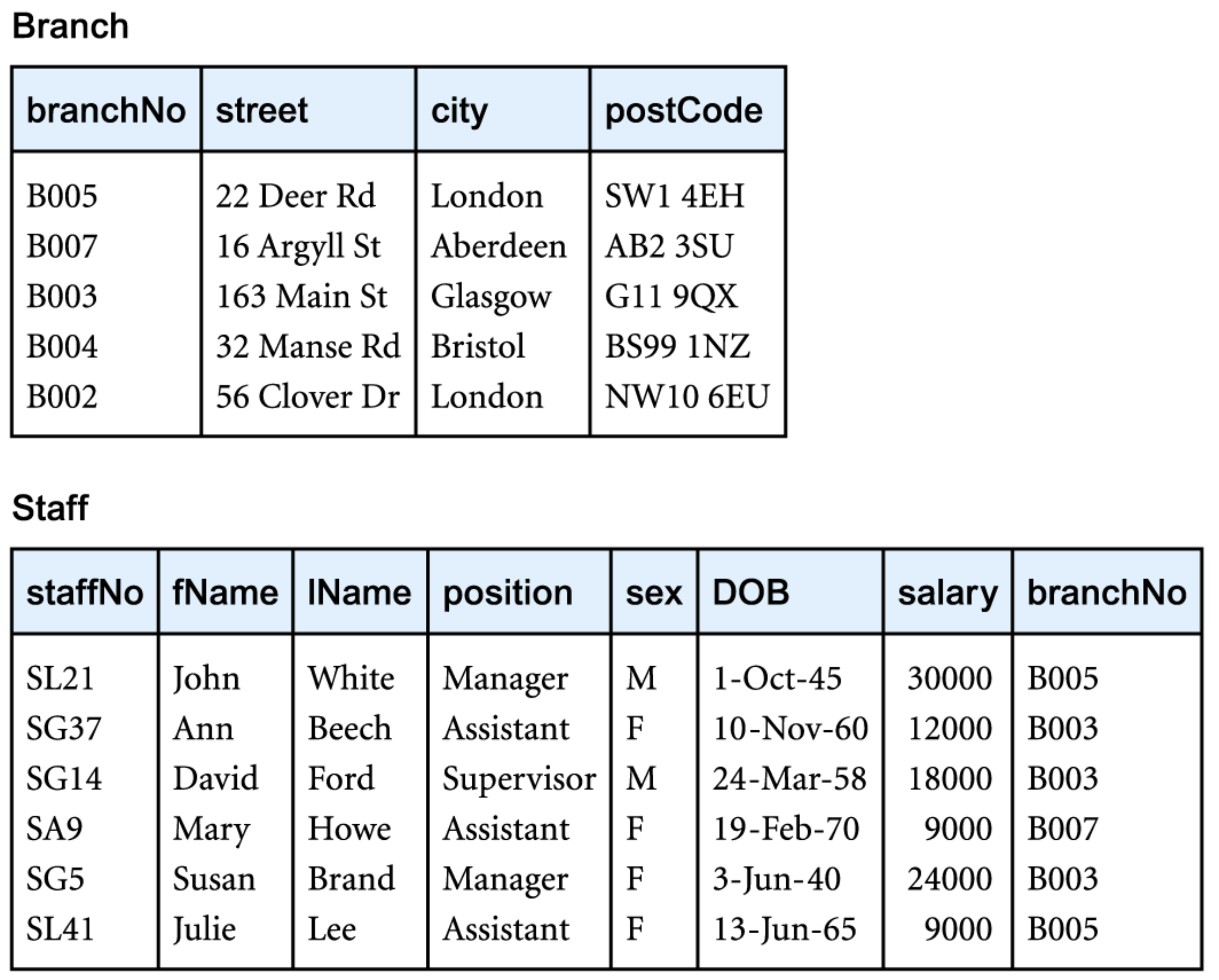
Data Model Cont.

- Categories of data models include:
  - Relational Model
  - Hierarchical Model
  - Network Model
  - Object-oriented database model.
  - Object-relational model.
  - NoSQL database models.
- Data Model comprises:
  - a structural part;
  - a manipulative part;
  - possibly a set of integrity rules.

Relational Data Models

- The mot common model, the relational model sorts data into tables, also known as relations, each of which consists columns and rows.
- Each column lists an attribute of the entity in question, such as price, zip code, etc. Together the attributes in a relation are called a domain.
- A particular attribute or combination of attributes is chosen as a primary key that can be referred to in other tables, when it is called a foreign key.

Relational Data Model Cont.



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

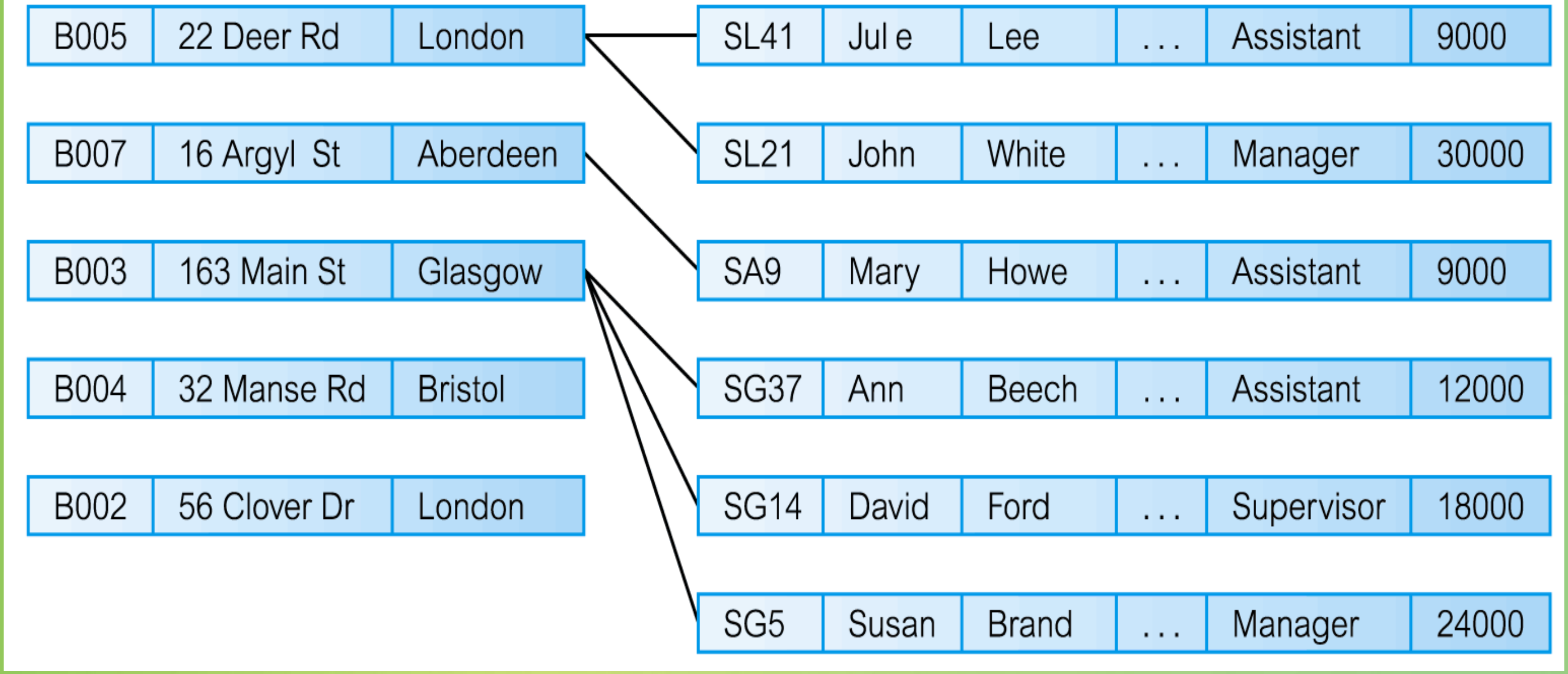
Staff

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

Network Data Models

- The network model builds the hierarchical model by allowing many to many relationships between linked record, implying multiple parent records.
- Based on mathematical set theory, the model is constructed with sets of related records, each set consists of one owner or parent record and one or more member or child records.
- A record can be a member or child in multiple sets, allowing this model to convey complex relationship.

Network Data Model Cont.



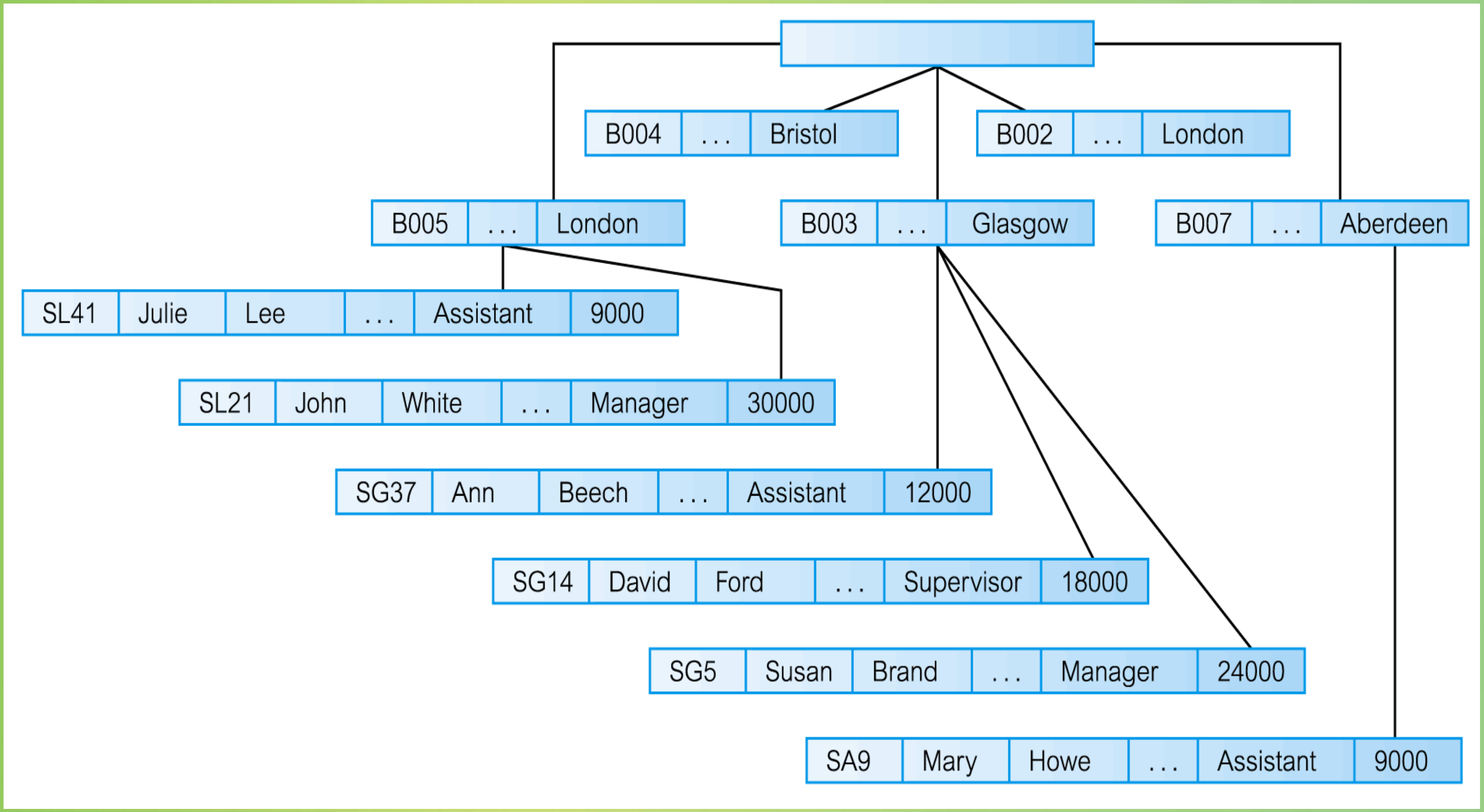
B005	22 Deer Rd	London
B007	16 Argyl St	Aberdeen
B003	163 Main St	Glasgow
B004	32 Manse Rd	Bristol
B002	56 Clover Dr	London

SL41	Jul e	Lee	. . .	Assistant	9000
SL21	John	White	. . .	Manager	30000
SA9	Mary	Howe	. . .	Assistant	9000
SG37	Ann	Beech	. . .	Assistant	12000
SG14	David	Ford	. . .	Supervisor	18000
SG5	Susan	Brand	. . .	Manager	24000

Hierarchical Data Models

- The hierarchical model organizes data into a tree-like structure, where each record has a single parent or root.
- Sibling records are sorted in particular order.
- The order is used as the physical order for storing the database, this model is good for describing many real-world relationships.

Hierarchical Data Model Cont.



B004 . . . Bristol

B002 . . . London

B005 . . . London

B003 . . . Glasgow

B007 . . . Aberdeen

SL41 Julie Lee . . . Assistant 9000

SL21 John White . . . Manager 30000

SG37 Ann Beech . . . Assistant 12000

SG14 David Ford . . . Supervisor 18000

SG5 Susan Brand . . . Manager 24000

SA9 Mary Howe . . . Assistant 9000

Conceptual Modelling

- Conceptual schema is the core of a system supporting all user views.
- Should be complete and accurate representation of an organization's data requirements.
- Conceptual modelling is process of developing a model of information use that is independent of implementation details.
- Result is a conceptual data model.

Functions of a DBMS

There are several functions that DBMS performs to ensure data integrity and consistency data in the database.

- Data Storage, Retrieval, and Update.
- A User-Accessible Catalog.
- Transaction Support.
- Concurrency Control Services.
- Recovery Services.

Functions of a DBMS Cont.

- Authorization Services.
- Support for Data Communication.
- Integrity Services.
- Services to Promote Data Independence.
- Utility Services.

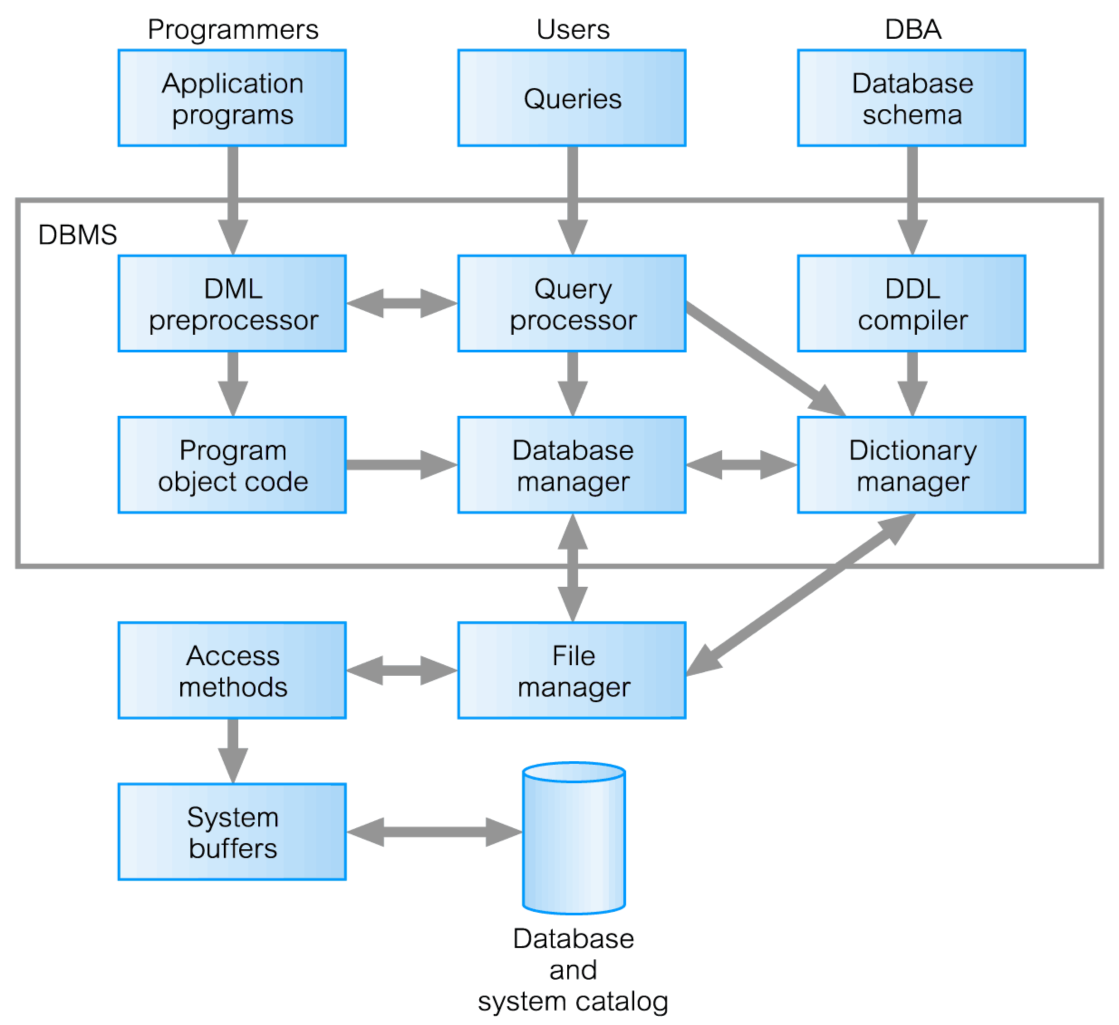
System Catalog

- System catalog is a collection of tables in a database which are used by the DBMS for describing the structure of the database, it is created automatically whenever a database is created, also it can be accessed by the users of the database, a user can access (for read only) the system catalog to get more information about the structure of database,
- Typically stores:
  - names, types, and sizes of data items;
  - constraints on the data;
  - names of authorized users;
  - data items accessible by a user and the type of access;
  - usage statistics.

Components of a DBMS

- **The software of DBMS seems very complex and confusing, but if we look DBMS internally then it is just a combination of different modules having specific tasks.**
- **Each module of DBMS performs particular task and operation.**
- **DBMS fetches query from the user, works on it using different plans an produces results as the data fed in it, there are many user interfaces which are used by end users to make queries.**

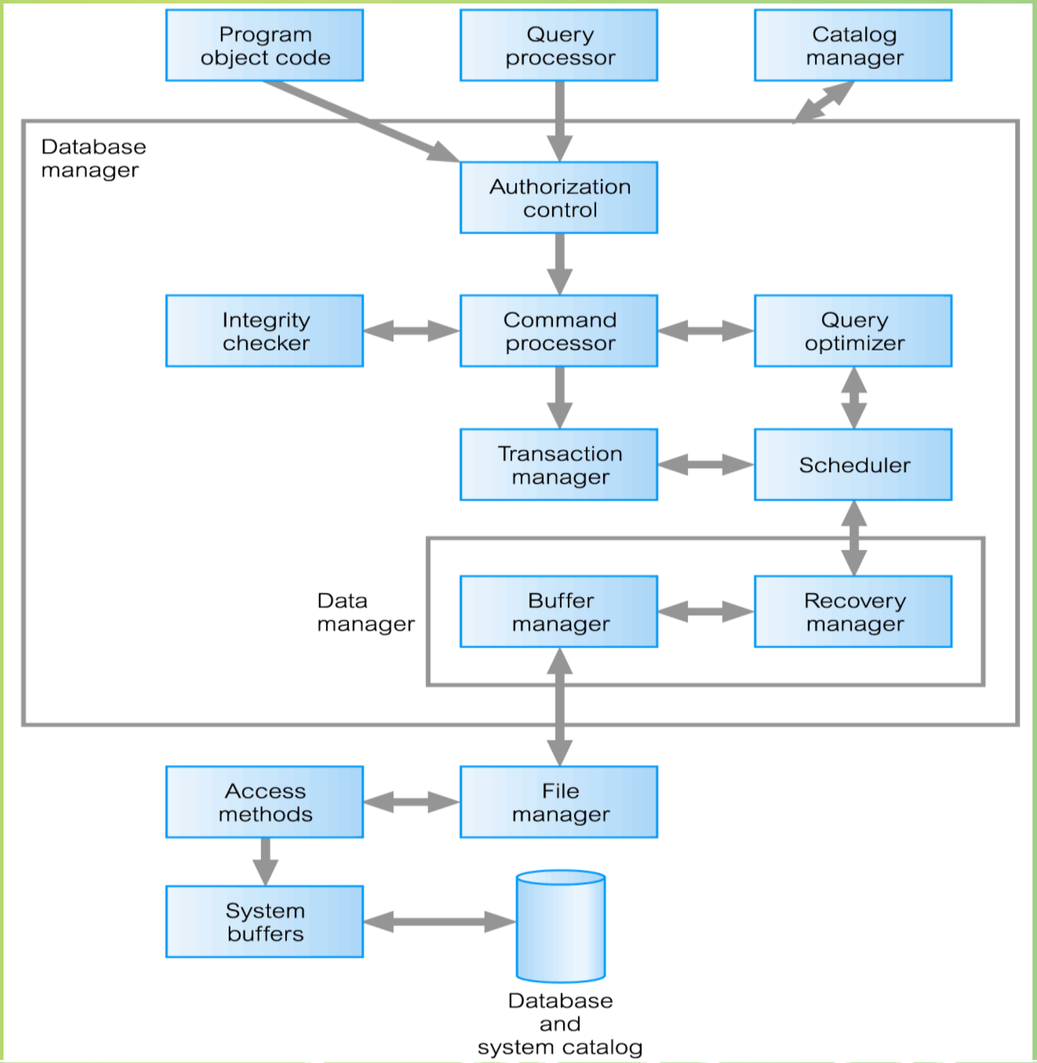
Components of a DBMS Cont.



Components of Database Manager

- **Database manager components issue the interface between low level data, application program and queries.**
- **Also it controls the consistency and integrity of data**
- **Security is also performed by the database manager.**

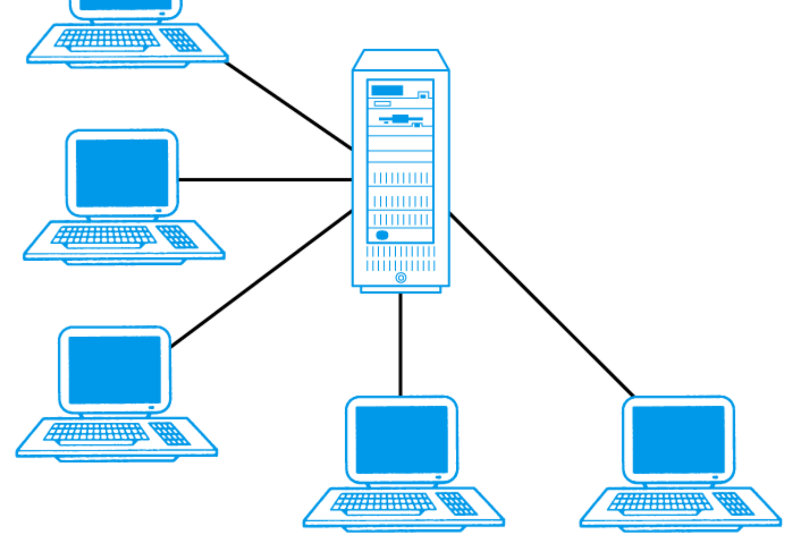
Components of Database Manager Cont.



Multi-User DBMS Architectures

- Teleprocessing
- File-server
- Client-server

Teleprocessing

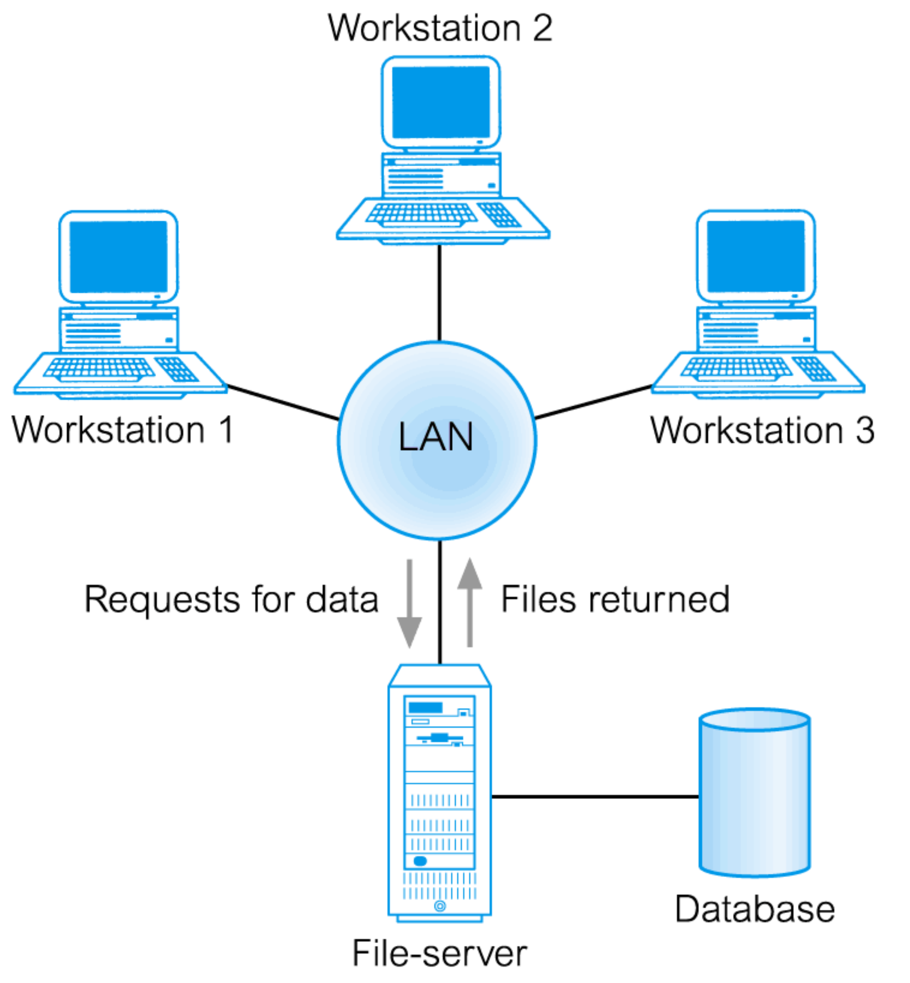


- Traditional architecture.
- Single mainframe with a number of terminals attached.
- This system can enables terminal to access DBMS and database directly, but it also makes very heavy load to the mainframe because it must runs applications and DBMS, formats data to be presented on terminals.

File-Server

- File-server is connected to several workstations across a network.
- Database resides on file-server.
- DBMS and applications run on each workstation.
- Disadvantages include:
  - Significant network traffic.
  - Copy of DBMS on each workstation.
  - Concurrency, recovery and integrity control more complex.

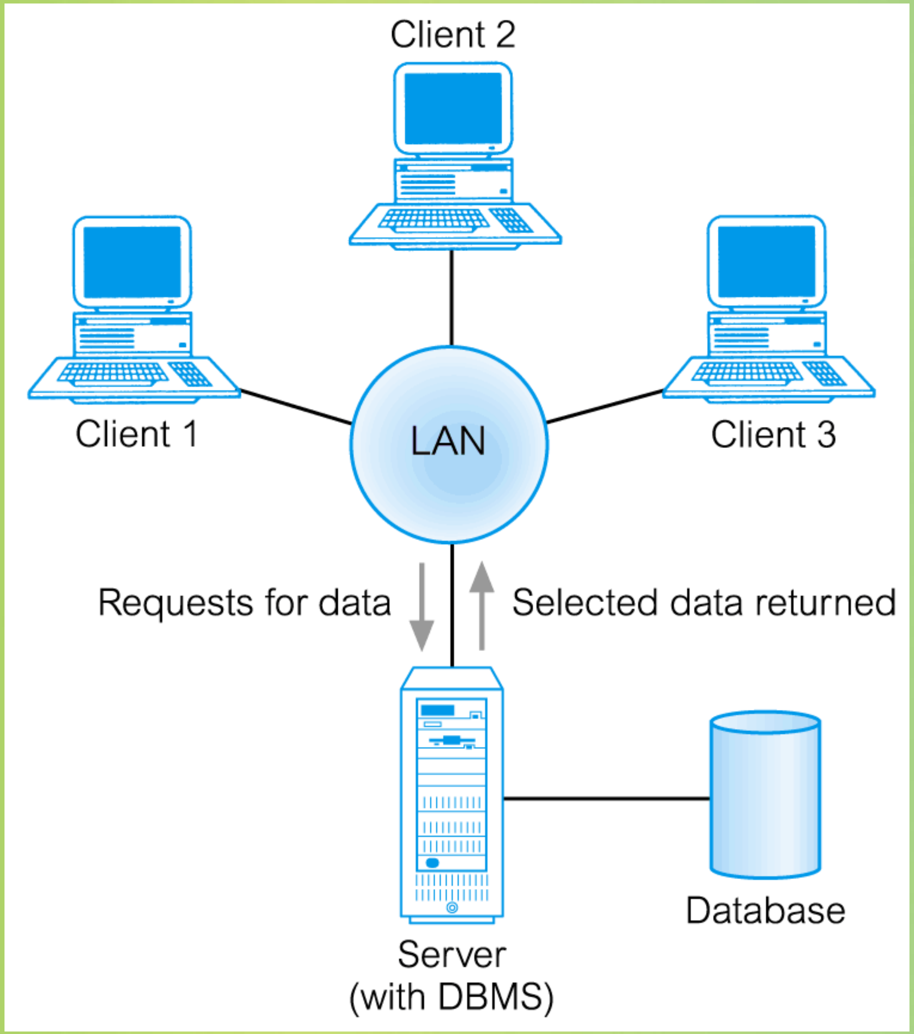
File-Server Architecture



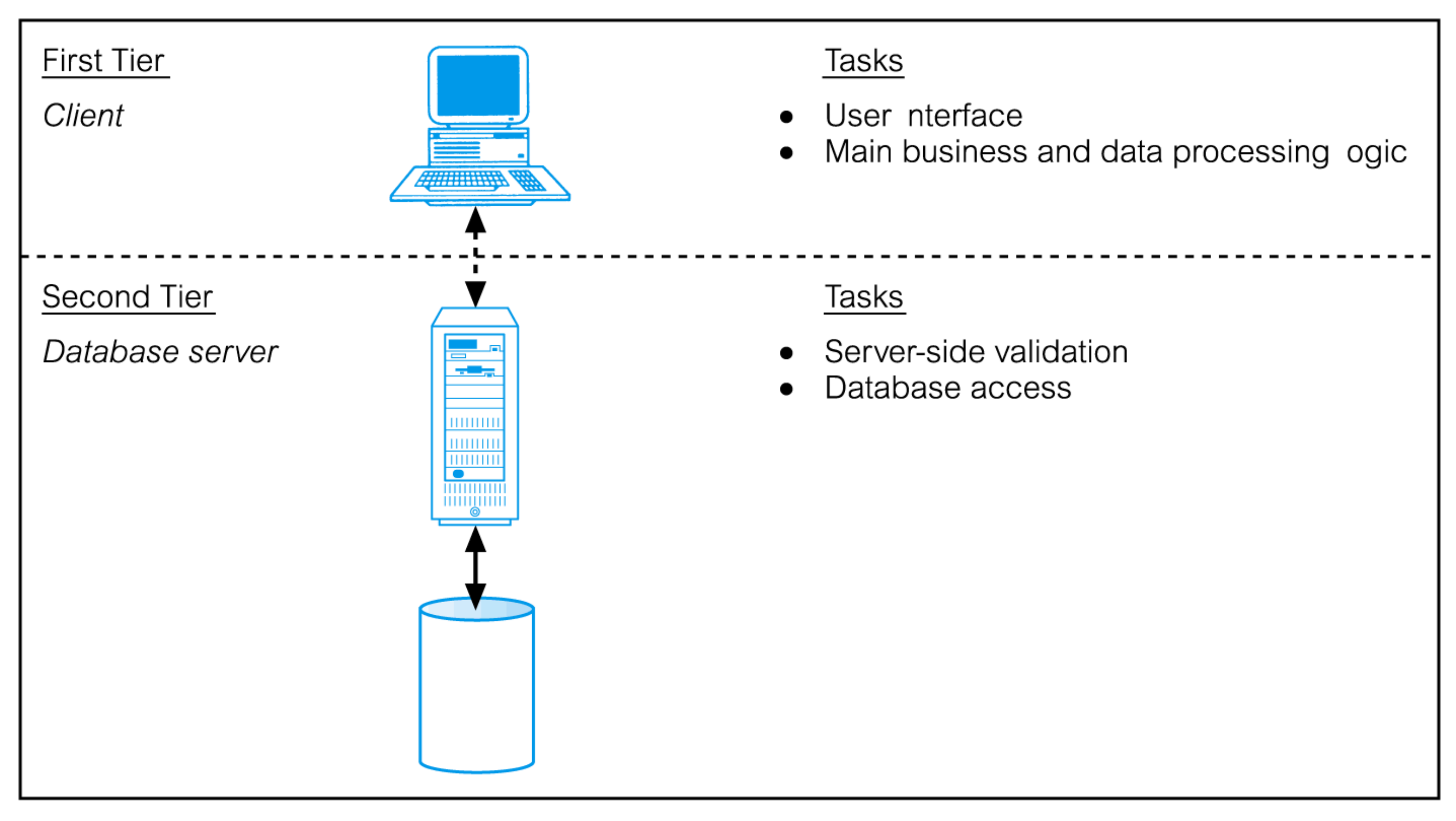
Traditional Two-Tier Client-Server

- Client (tier 1) manages user interface and runs applications.
- Server (tier 2) holds database and DBMS.
- Advantages include:
  - wider access to existing databases;
  - increased performance;
  - possible reduction in hardware costs;
  - reduction in communication costs;
  - increased consistency.

Traditional Two-Tier Client-Server



Traditional Two-Tier Client-Server



First Tier

Client

Tasks

- User nterface
- Main business and data processing ogic

Second Tier

Database server

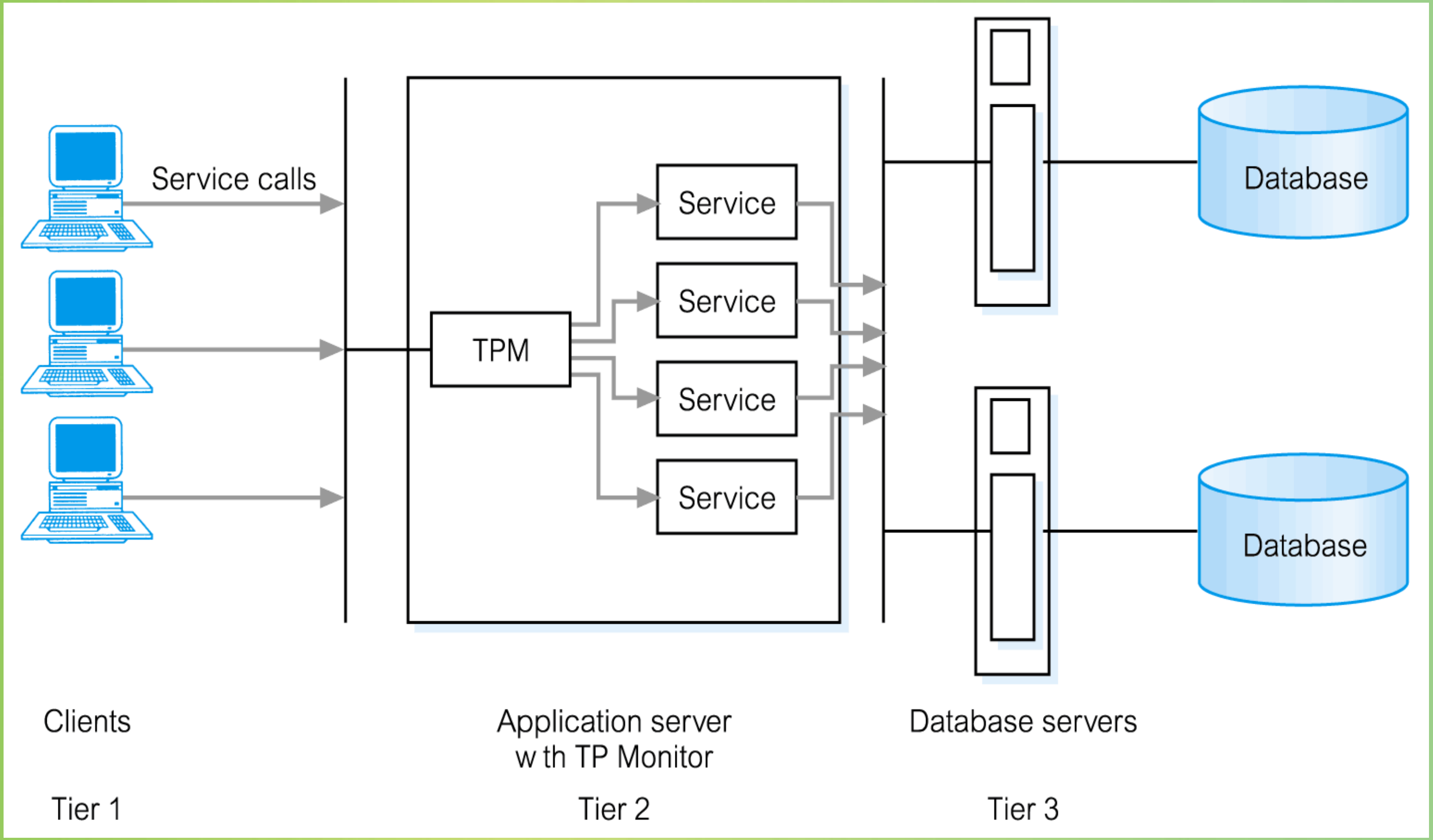
Tasks

- Server-side validation
- Database access

Transaction Processing Monitors

- Program that controls data transfer between clients and servers in order to provide a consistent environment, particularly for Online Transaction Processing (OLTP).

TPM as middle tier of 3-tier client-server


Service calls
TPM
Service
Service
Service
Service
Database
Database
Clients
Application server
w th TP Monitor
Database servers
Tier 1
Tier 2
Tier 3