Chapter 1

Introduction to Databases, Environment & Architecture

Chapter 1 – Objectives

- Characteristics of file-based systems.
- Meaning of the term Database and DBMS.
- Functions & components of the DBMS.
- Advantages and disadvantages of DBMSs.
- Purpose of three-level database architecture.
- Purpose/importance of conceptual modeling.
- Meaning of client–server architecture and
- Advantages of client–server architecture for a DBMS.

File-Based Systems

- An early attempt to computerize the manual filing system.
- 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

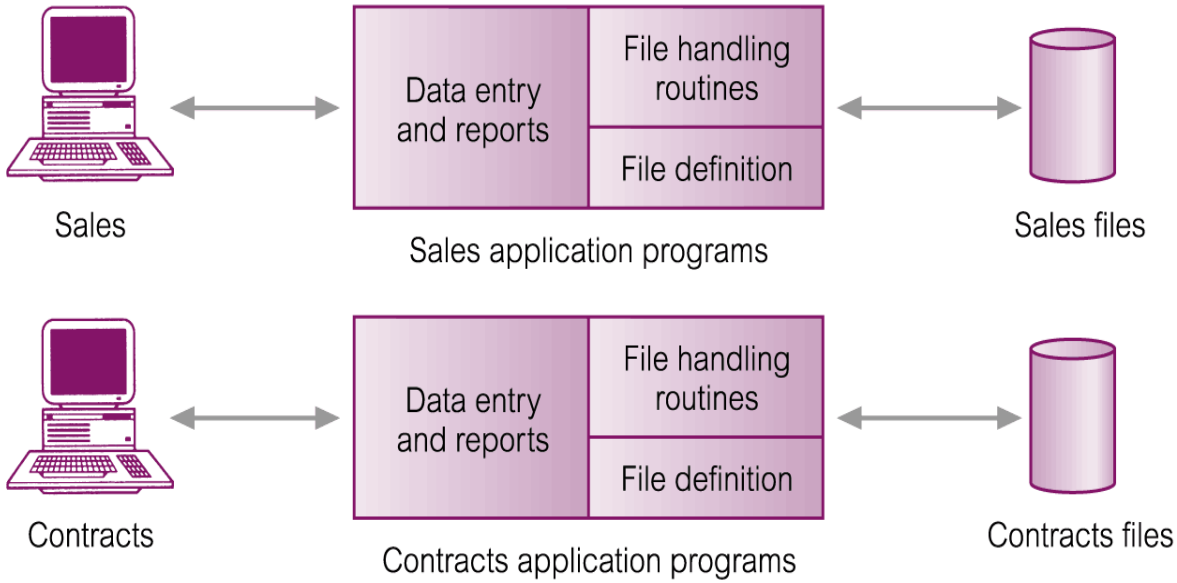


Figure 1.5 File-based processing.

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

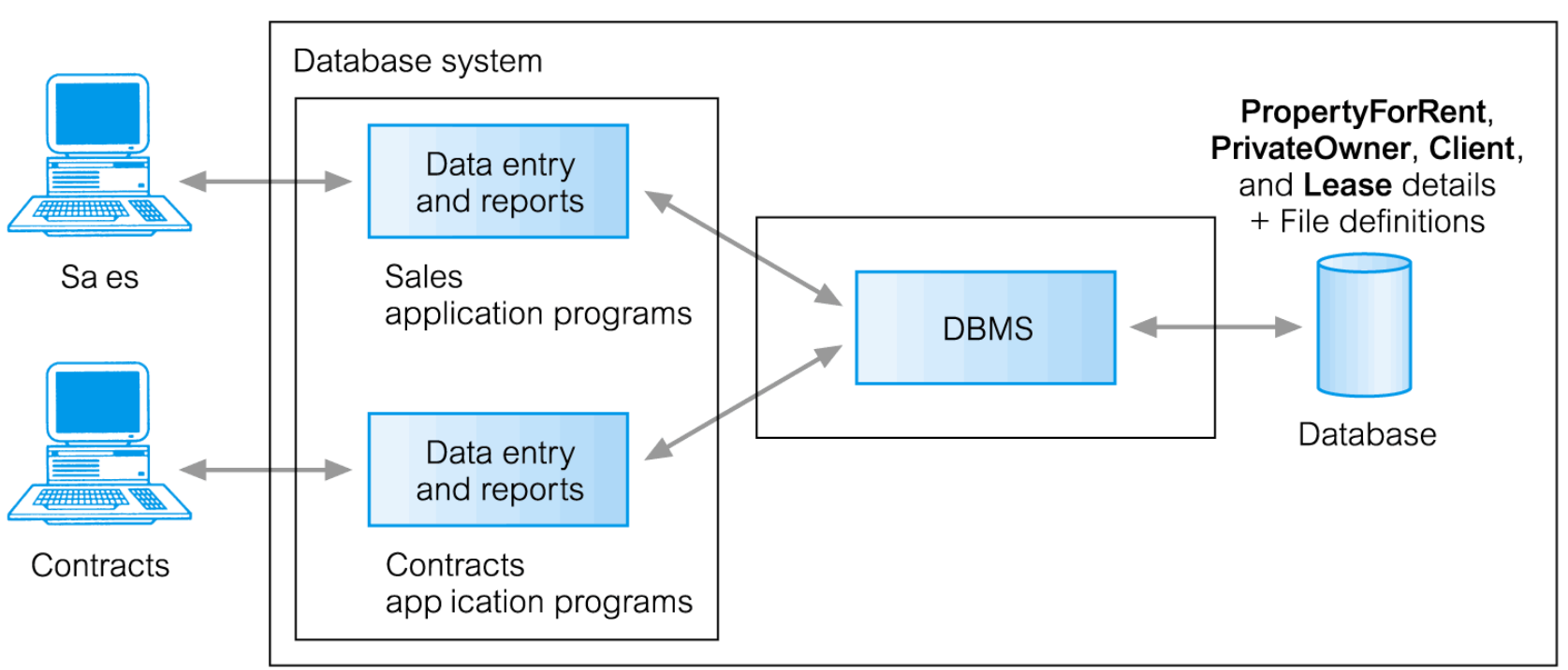
- Arose because:
  - Definition of data was embedded in application programs, rather than being stored separately and independently.
  - No control over access and manipulation of data beyond that imposed by application programs.

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

Database & DBMS

- A database is a shared collection of logically related data comprises entities, attributes, and relationships of an organization.
- System catalog (metadata) provides description of data to enable program–data independence.
- A DBMS is a software system that enables users to define, create, maintain, and access the database.

Database Management System (DBMS)

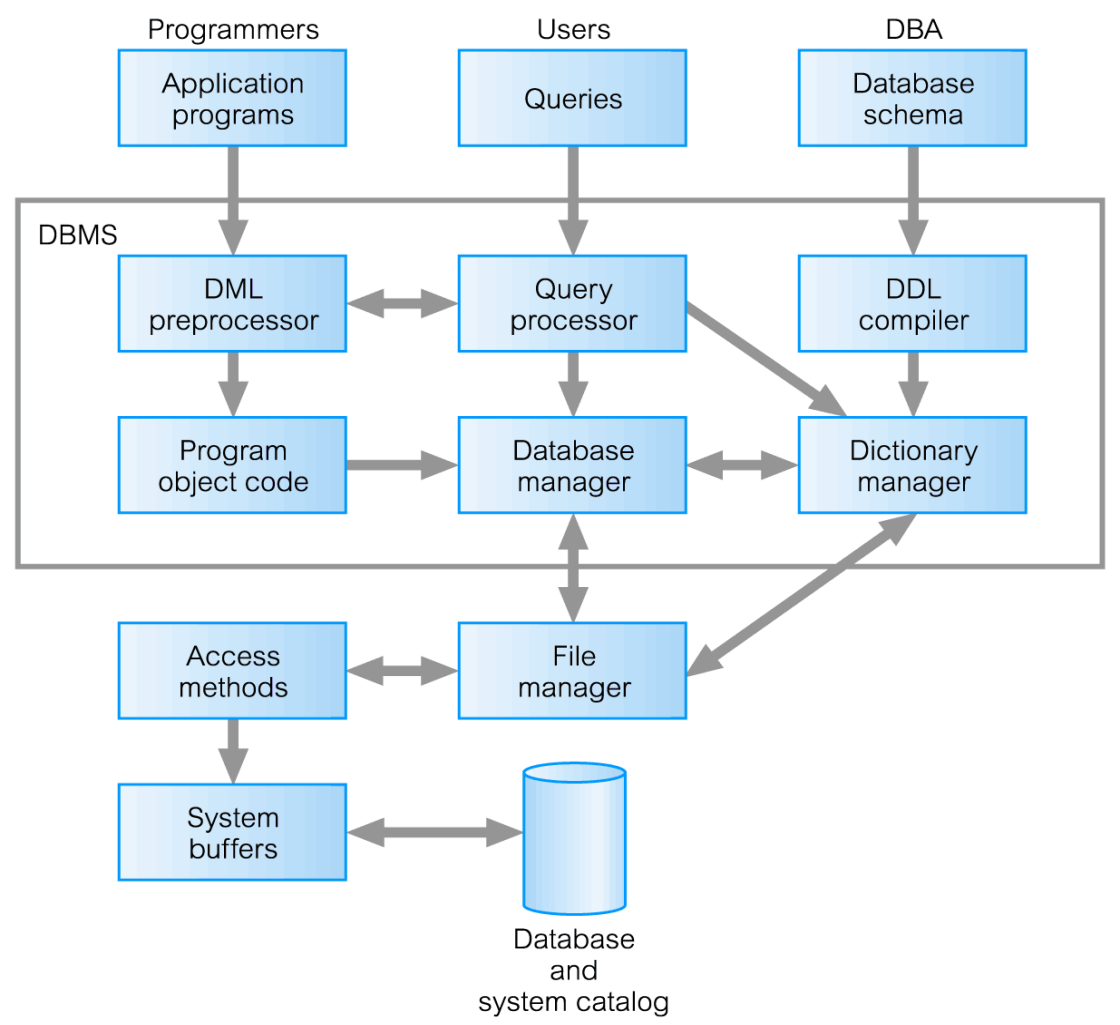


PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)
Lease (leaseNo, propertyNo, clientNo, paymentMethod, deposit, paid, rentStart, rentF nish)

Functions & Components of a DBMS

- Data Storage, Retrieval, and Update.
- A User-Accessible Catalog.
- Transaction Support.
- Concurrency Control Services.
- Recovery Services.
- Authorization Services.
- Support for Data Communication.
- Integrity Services.
- Services to Promote Data Independence.

Components of a DBMS



System Catalog

- One of the fundamental components of DBMS.
- Repository of information (metadata) describing the data in the 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.

Advantages of DBMSs

- Control of data redundancy
- Data consistency
- Sharing of data
- Improved data integrity
- Improved security
- Enforcement of standards
- Economy of scale

Advantages of DBMSs

- Improved data accessibility and responsiveness
- Improved maintenance through data independence
- Increased concurrency
- Improved backup and recovery services
- Increased productivity

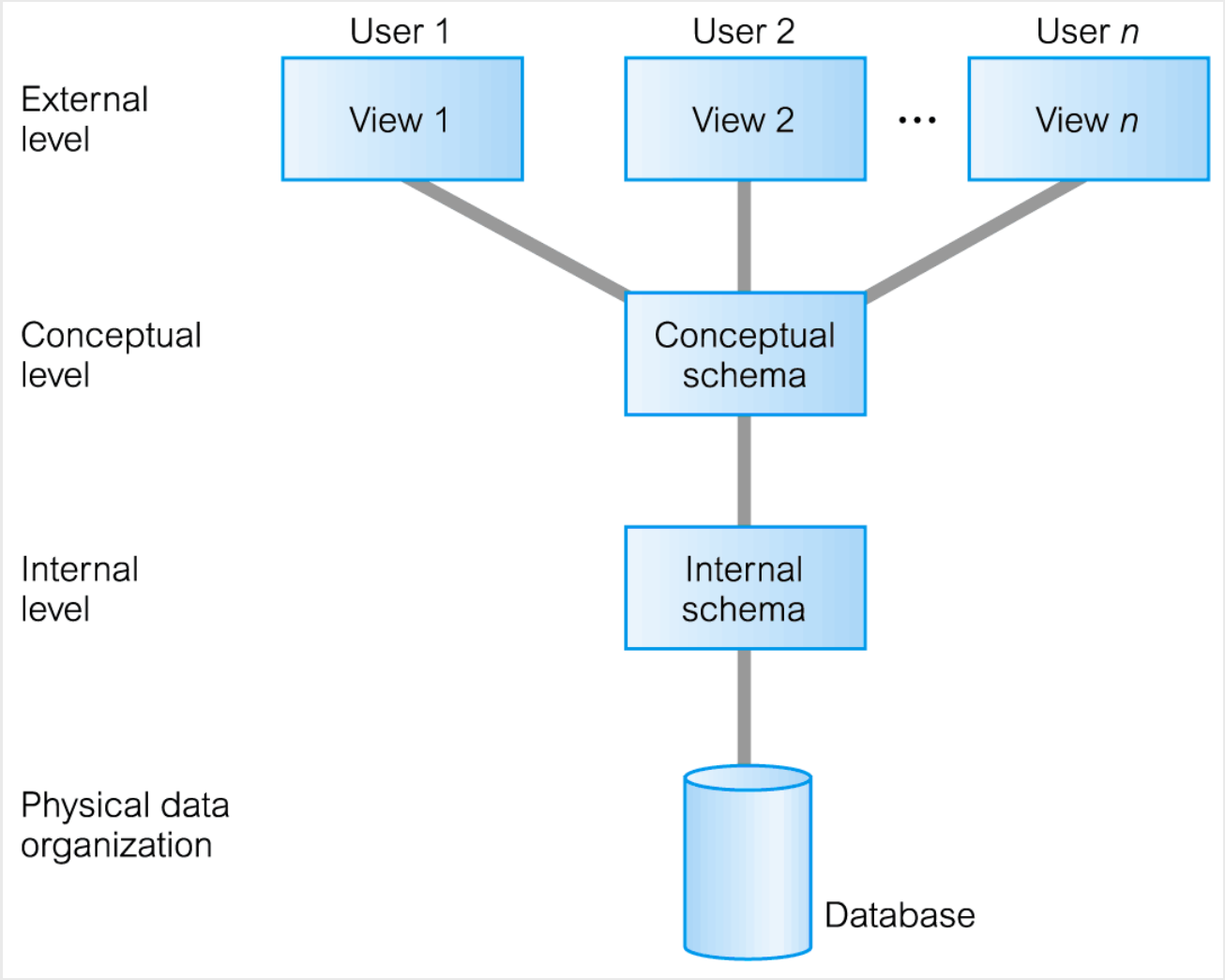
Disadvantages of DBMSs

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

ANSI-SPARC Three-Level Architecture

- External Level
  - Users' view of the database.
  - Describes that part of database that is relevant to a particular user.
- Conceptual Level
  - Community view of the database.
  - Describes what data is stored in database and relationships among the data.
- Internal Level
  - Describes how the data is physically stored in the database.

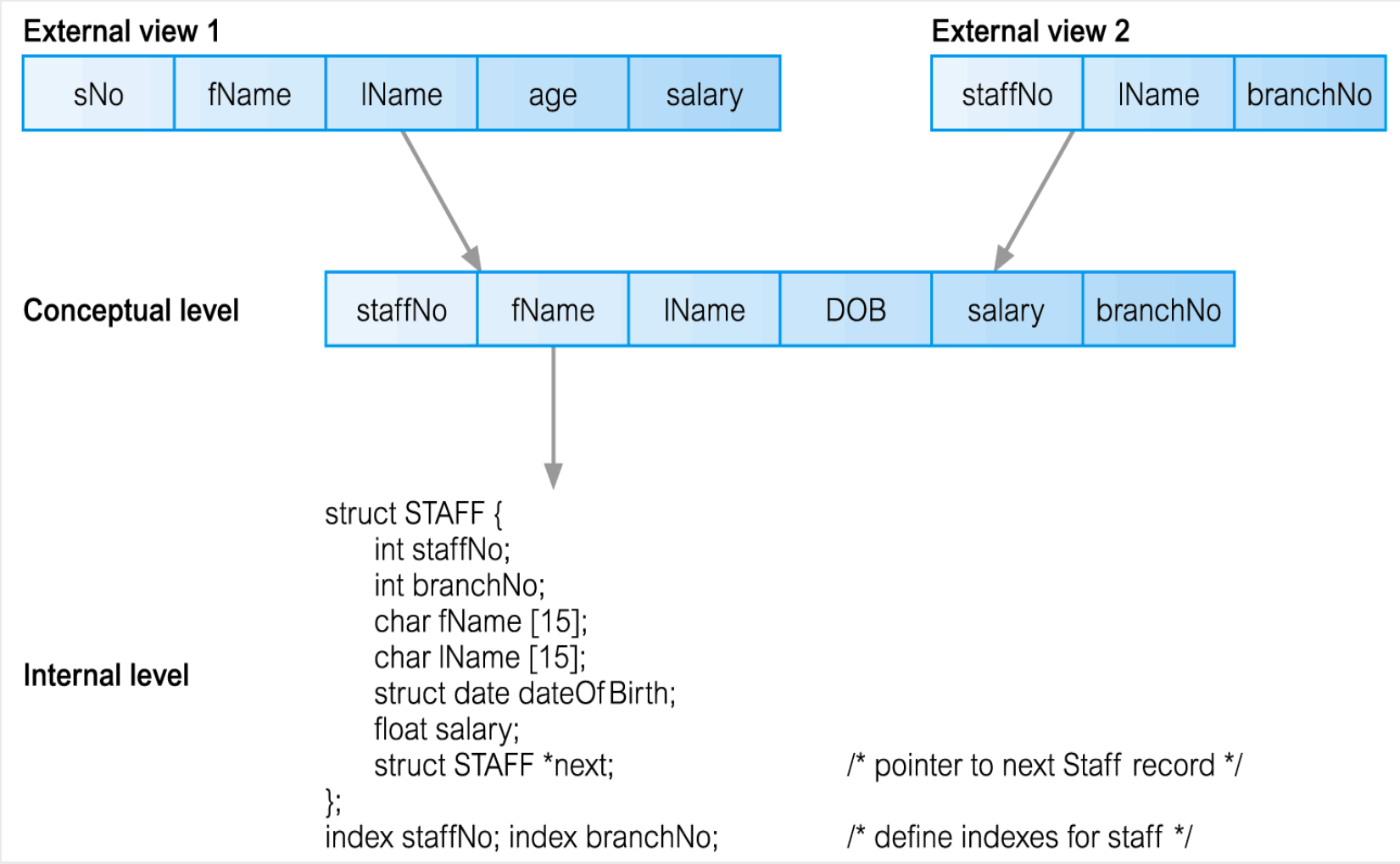
ANSI-SPARC Three-Level Architecture



Objectives of Three-Level Architecture

- All users should be able to access same data.
- A user's view is immune to changes made in other views.
- Users should not need to know physical database storage details.

Differences between Three Levels of ANSI-SPARC Architecture



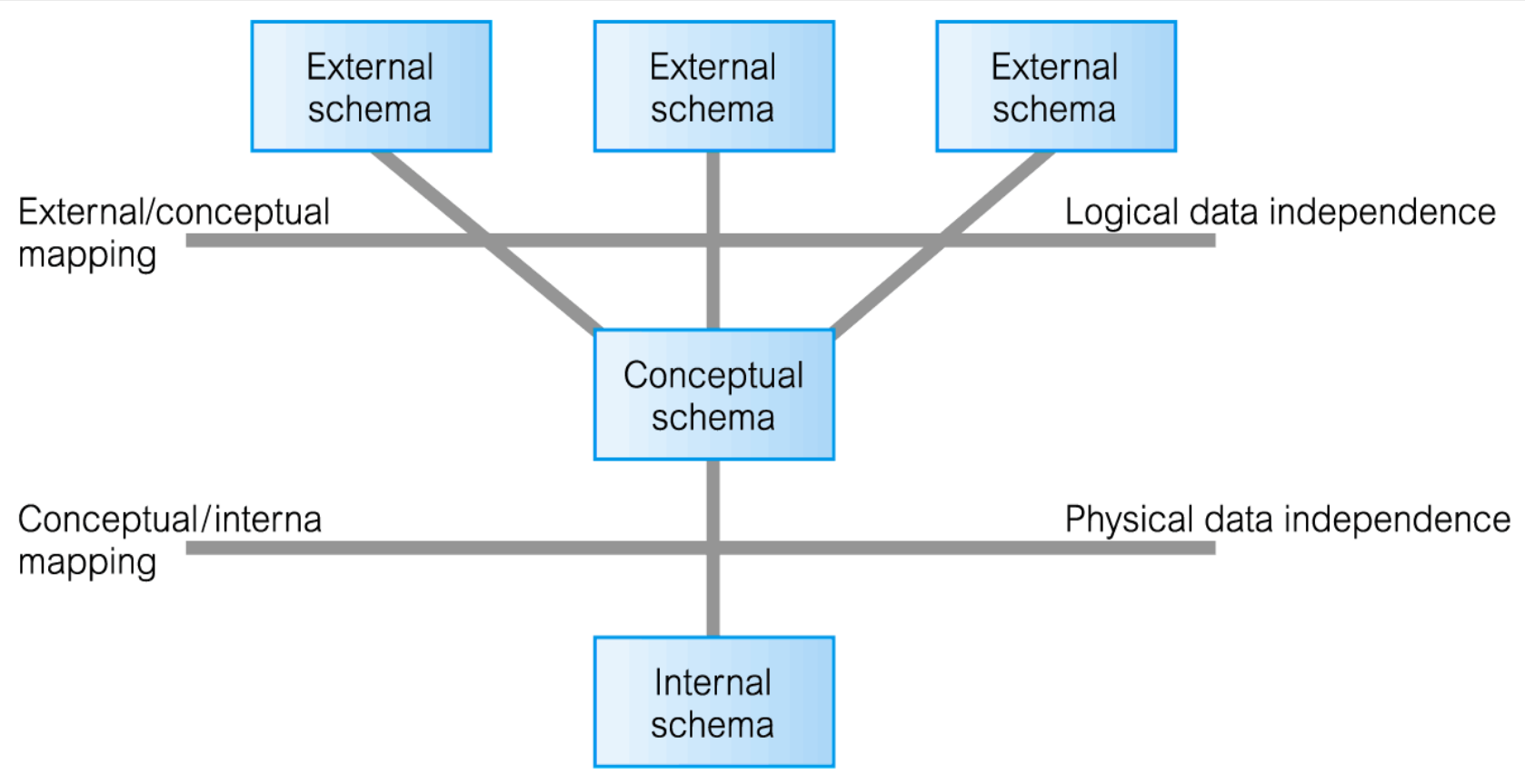
Data Independence

- Logical Data Independence
  - Refers to immunity of external schemas to changes in conceptual schema.
  - Conceptual schema changes (e.g. addition/removal of entities).

Data Independence

- 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.

Data Independence and the ANSI-SPARC Three-Level Architecture



Data Model

Integrated collection of concepts for describing data, relationships between data, and constraints on the data in an organization.

- Data Model comprises:
  - a structural part;
  - a manipulative part;
  - possibly a set of integrity rules.

Data Model

- Purpose
  - To represent data in an understandable way.

- Categories of data models include:
  - Object-based (ERD & Object Oriented)
  - Record-based (Relational , Network , Hierarchical)
  - Physical.

Relational Data Model

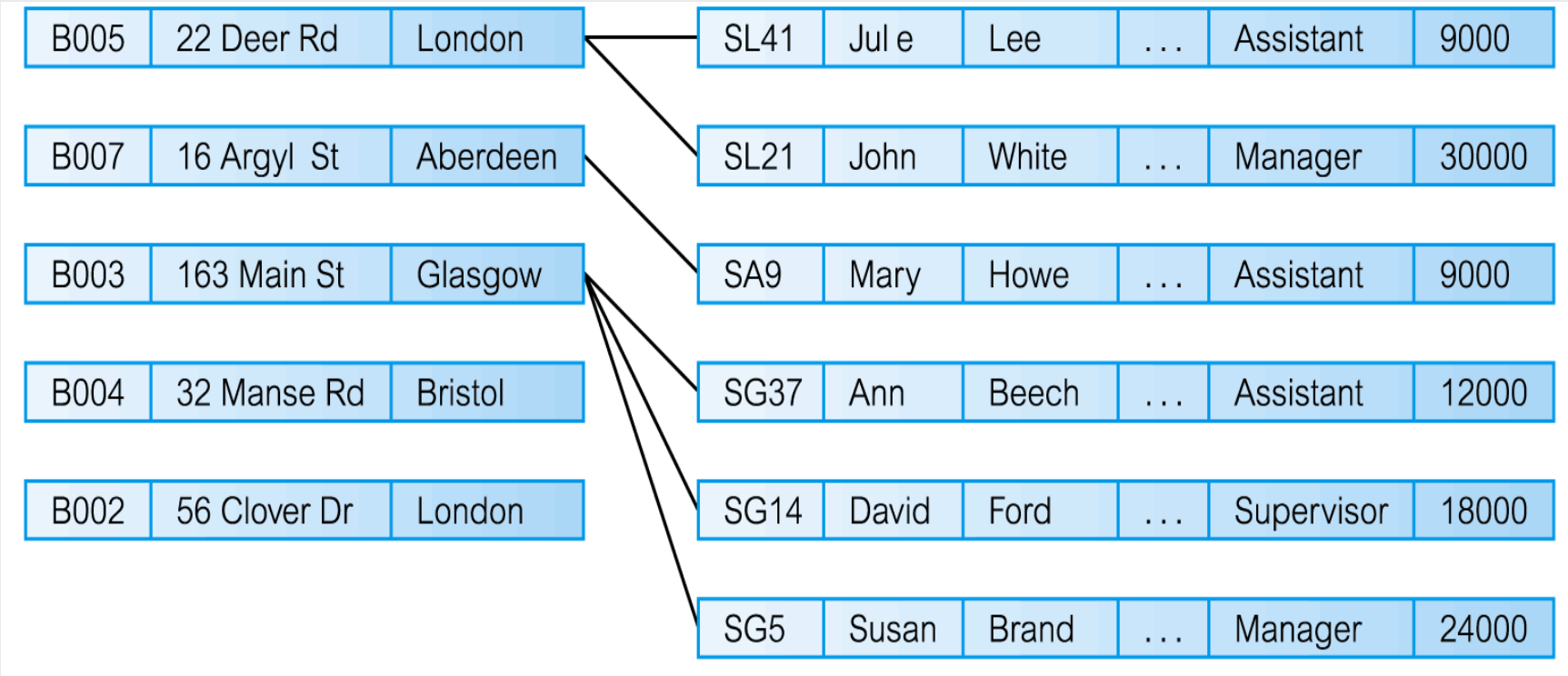
Branch

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

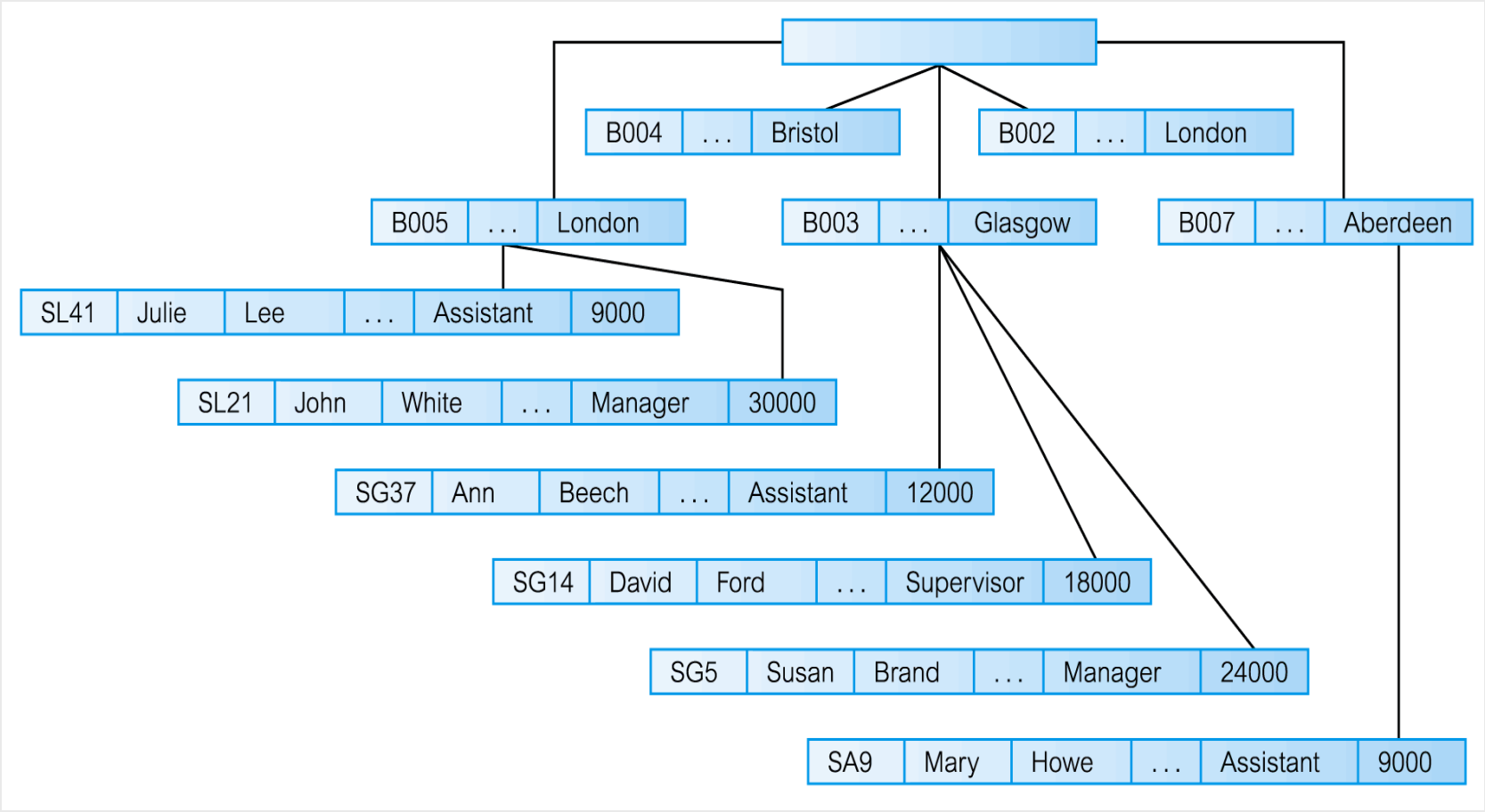
Staff

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 Model



Hierarchical Data Model

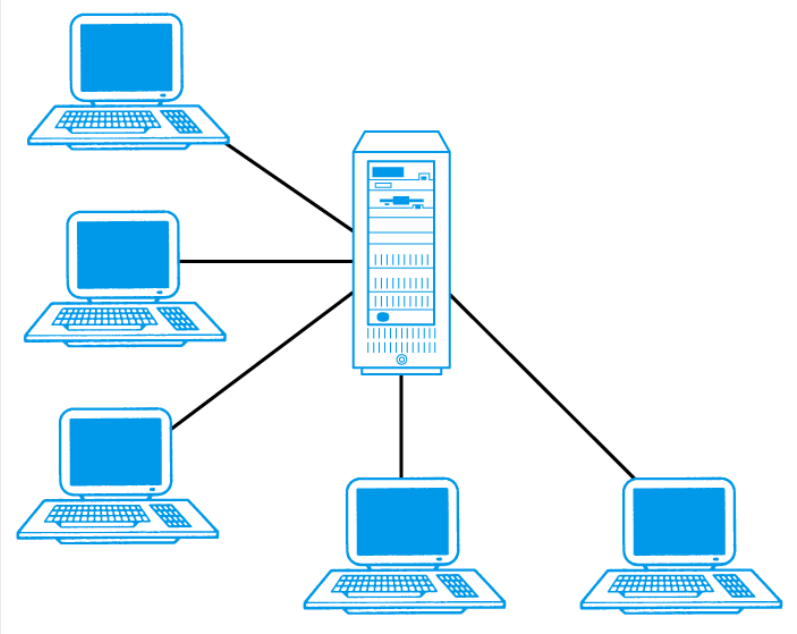


Multi-User DBMS Architectures

- Teleprocessing
- File-server
- Client-server

Teleprocessing

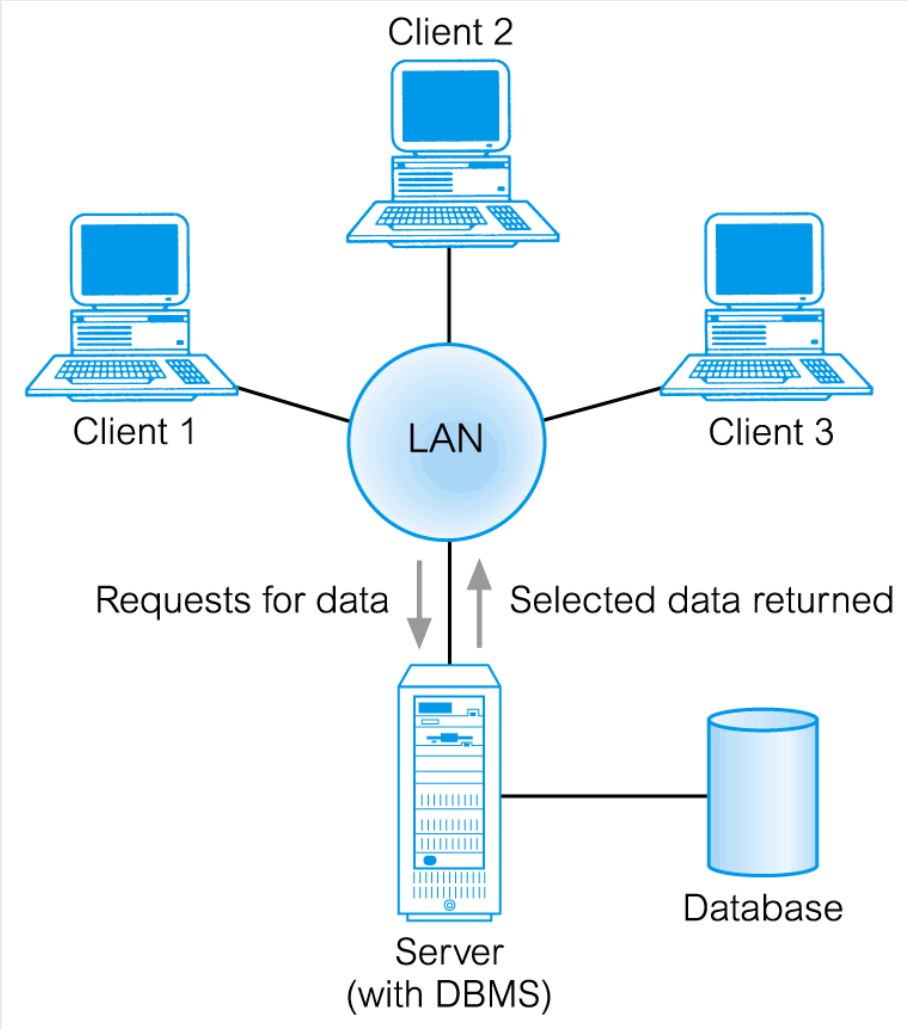
- Traditional architecture.
- Single mainframe with a number of terminals attached.
- Trend is now towards downsizing.



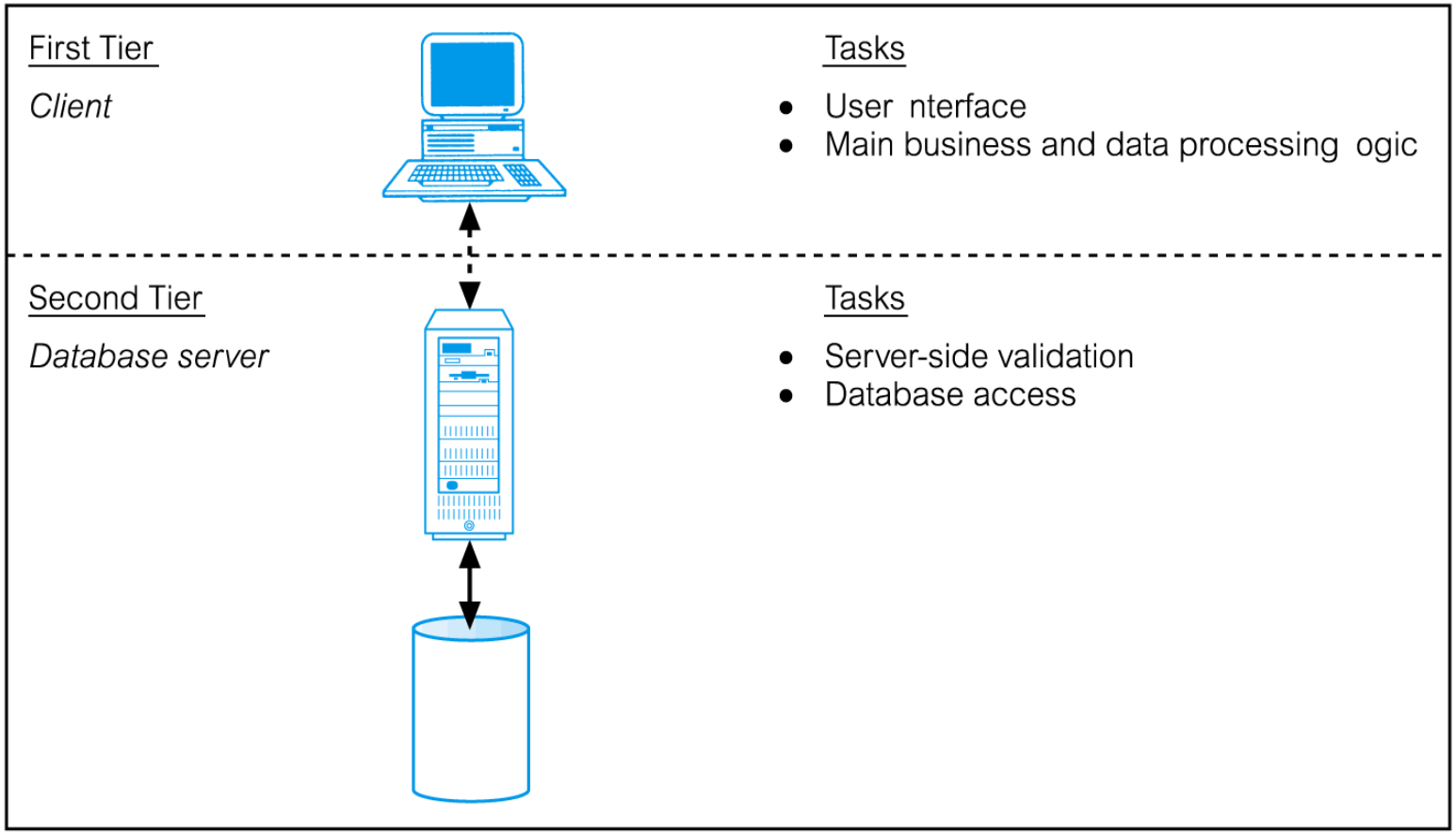
Traditional Two-Tier Client-Server

- 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



Summary of client–server functions

CLIENT	SERVER
Manages the user interface	Accepts and processes database requests from clients
Accepts and checks syntax of user input	Checks authorization
Processes application logic	Ensures integrity constraints not violated
Generates database requests and transmits to server	Performs query/update processing and transmits response to client
Passes response back to user	Maintains system catalog Provides concurrent database access Provides recovery control

Three-Tier Client-Server

- **Client side presented two problems preventing true scalability:**
  - **'Fat' client, requiring considerable resources on client's computer to run effectively.**
  - **Significant client side administration overhead.**
- **By 1995, three layers proposed, each potentially running on a different platform.**

Three-Tier Client-Server

- **Advantages:**
  - **'Thin' client, requiring less expensive hardware.**
  - **Application maintenance centralized.**
  - **Easier to modify or replace one tier without affecting others.**
  - **Separating business logic from database functions makes it easier to implement load balancing.**
  - **Maps quite naturally to Web environment.**

Three-Tier Client-Server