BACS3183
Advanced Database Management

Chapter 2
Database Systems

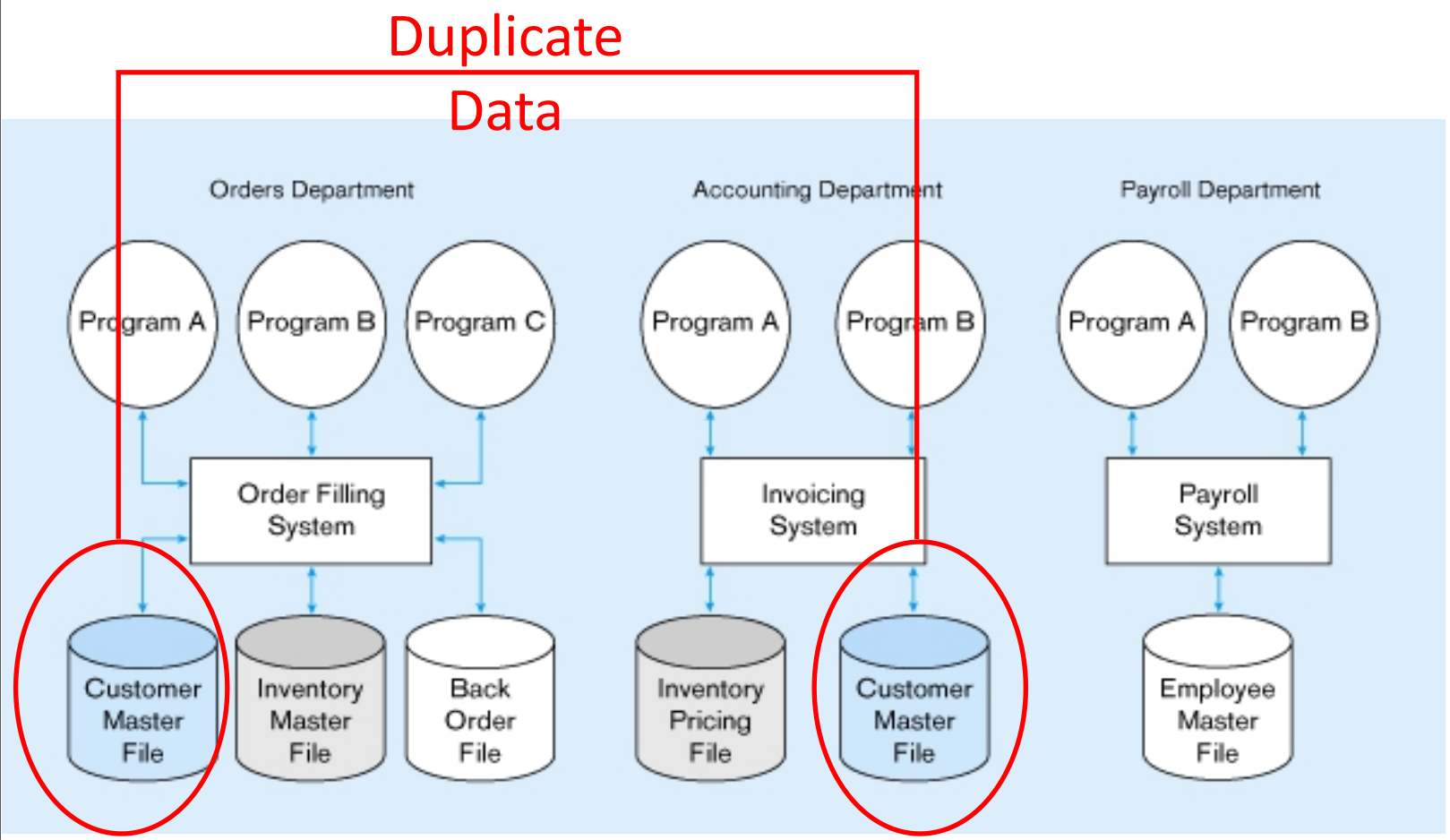
Learning Outcomes

- Explain the characteristics that distinguish the database approach from the traditional approach of programming with data files.
- Describe the basic functions and components of a DBMS
- Discuss the ANSC-SPARC Three-level architecture
- Describe the meaning of logical and physical data independence
- Differentiate among the various DBMS architectures

1. Disadvantages of File Processing (Traditional Approach)

- **Program-Data Dependence (See slide 5)**
  - All programs maintain metadata for each file they use
- **Data Redundancy (Duplication of data) (See slide 6)**
  - Different systems/programs have separate copies of the same data
- **Limited Data Sharing**
  - No centralized control of data
- **Lengthy Development Times**
  - Programmers must design their own file formats
- **Excessive Program Maintenance**
  - 80% of the information systems budget

Example : Three file processing systems



Program-Data Dependence

Data Redundancy (duplication of data)

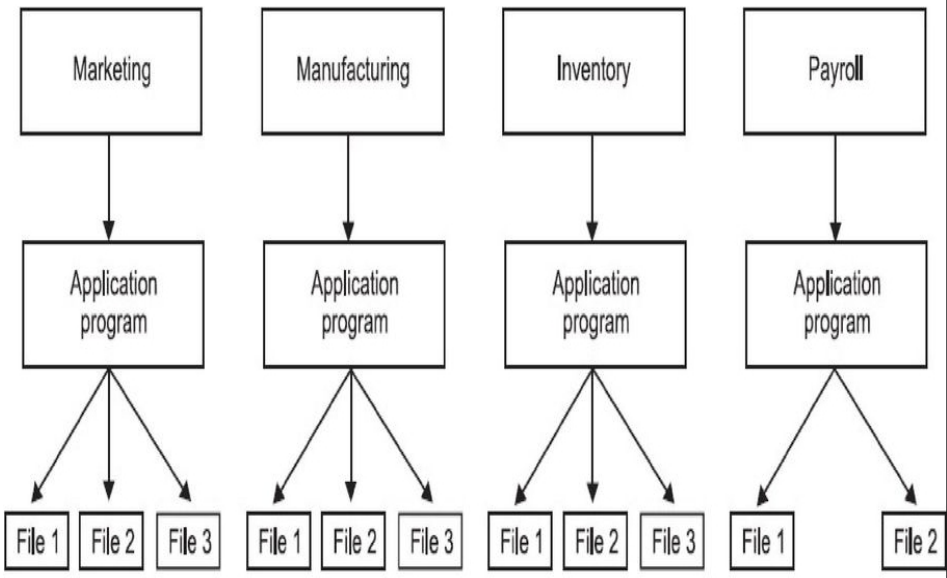
Limited Data Sharing

Lengthy Development Times

Excessive Program Maintenance

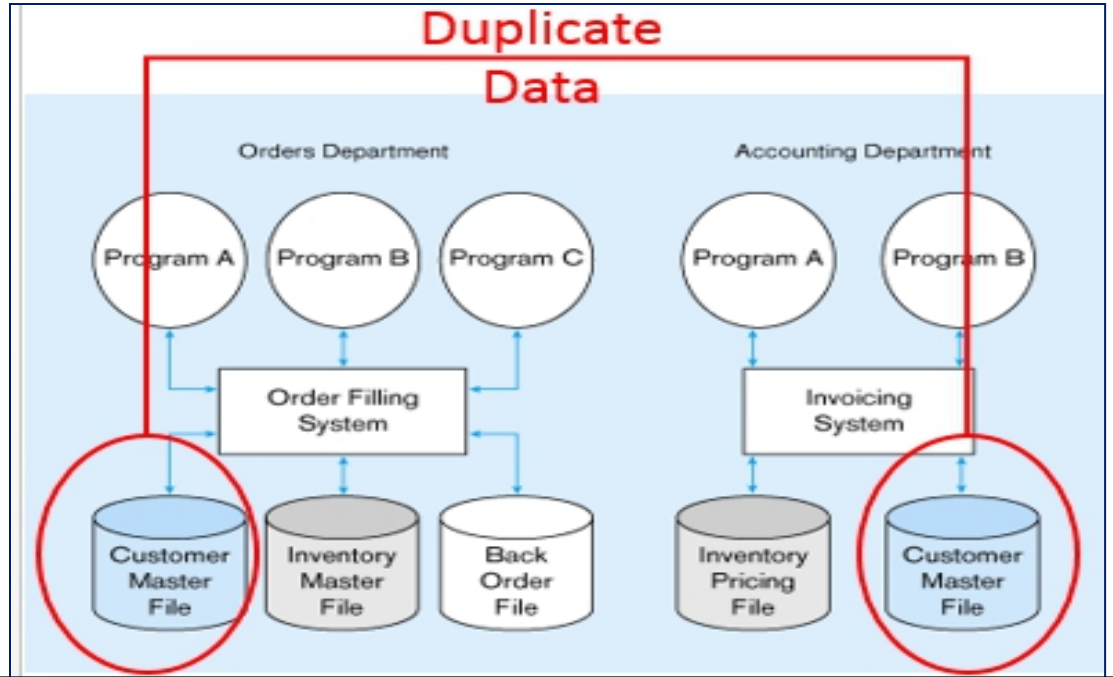
Problems with Data Dependency

- Each application programmer must maintain their own data
- Each application program needs to include code for the metadata of each file
- Each application program must have its own processing routines for reading, inserting, updating and deleting data
- Lack of coordination and central control
- Non-standard file formats



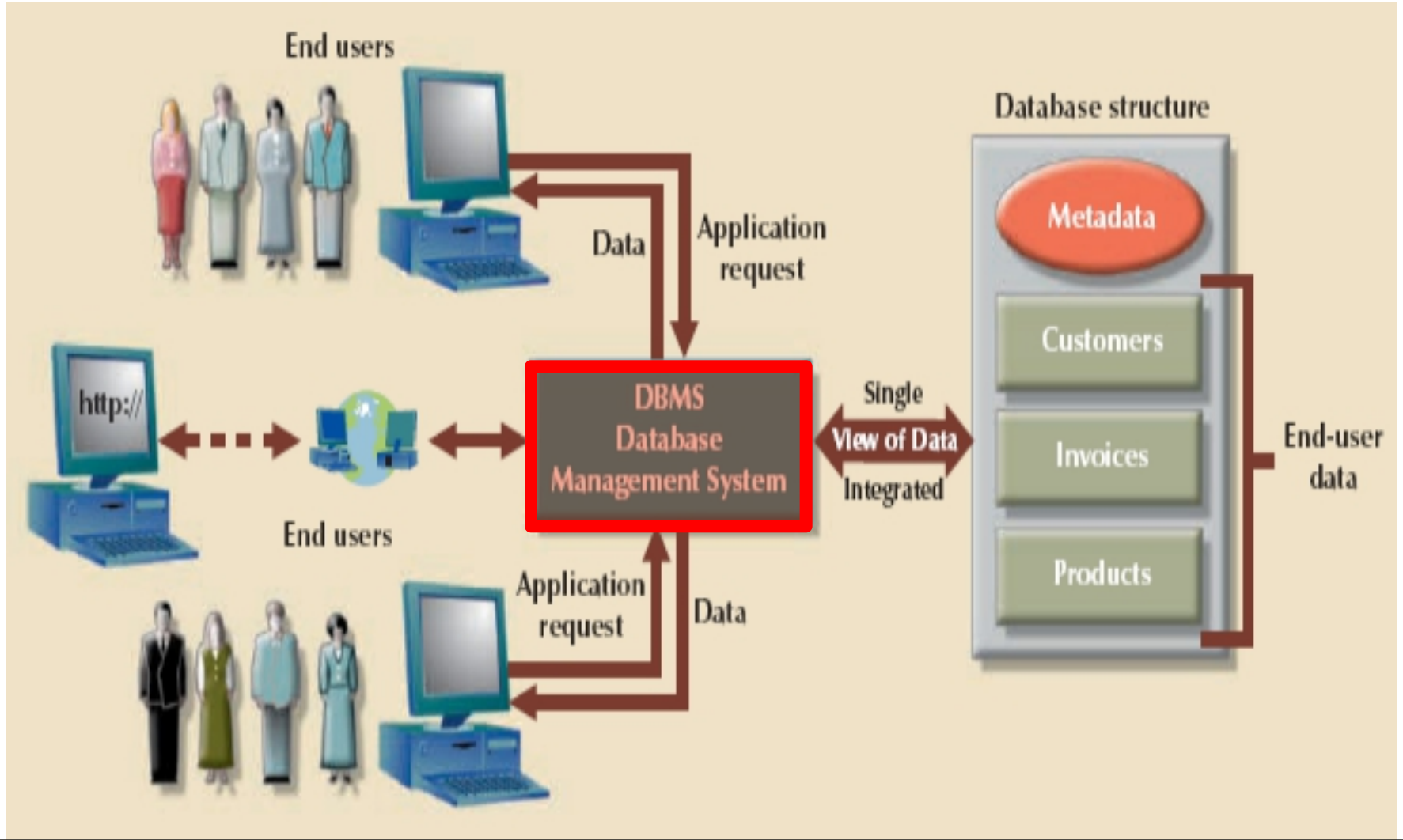
Problems with Data Redundancy

- **Waste of space to have duplicate data**
- **Causes more maintenance headaches**
- **The biggest Problem:**
  - **When data changes in one file, could cause *inconsistencies***
  - **Compromises *data integrity***



2. Database Management System (DBMS)

- A software system that enables users to define, create, maintain, and control access to the database.



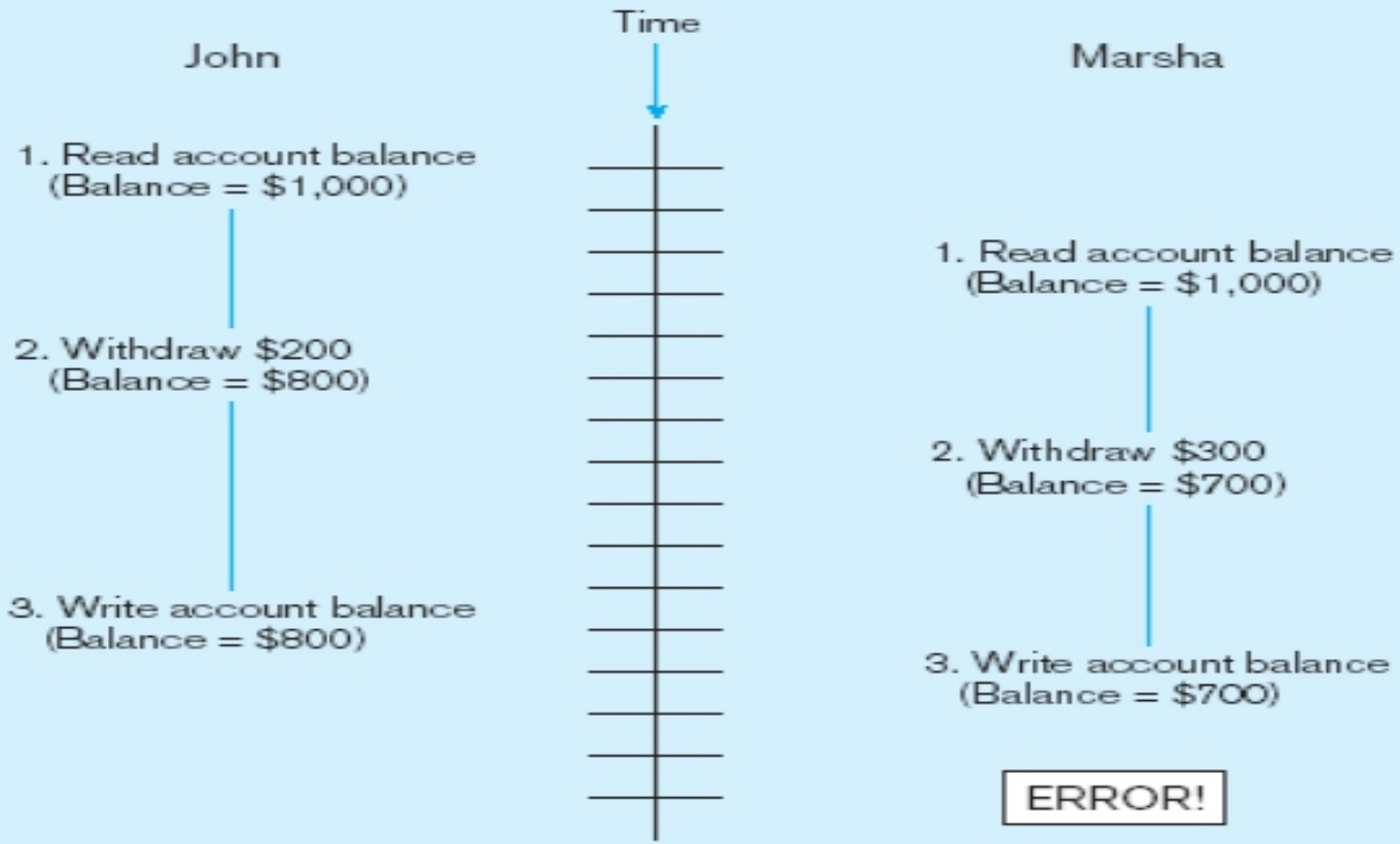
2.1 Functions of a DBMS

- **Store, Update and Retrieve Data**

- **Concurrency Control Services.**
  - **Multiple users make updates to the database simultaneously, ensure updates are made correctly and the end result is accurate.**
  - **Solution:**
    - ✓ **Locking & 2-phase locking,**
    - ✓ **Time-stamping**
    - ✓ **Versioning**

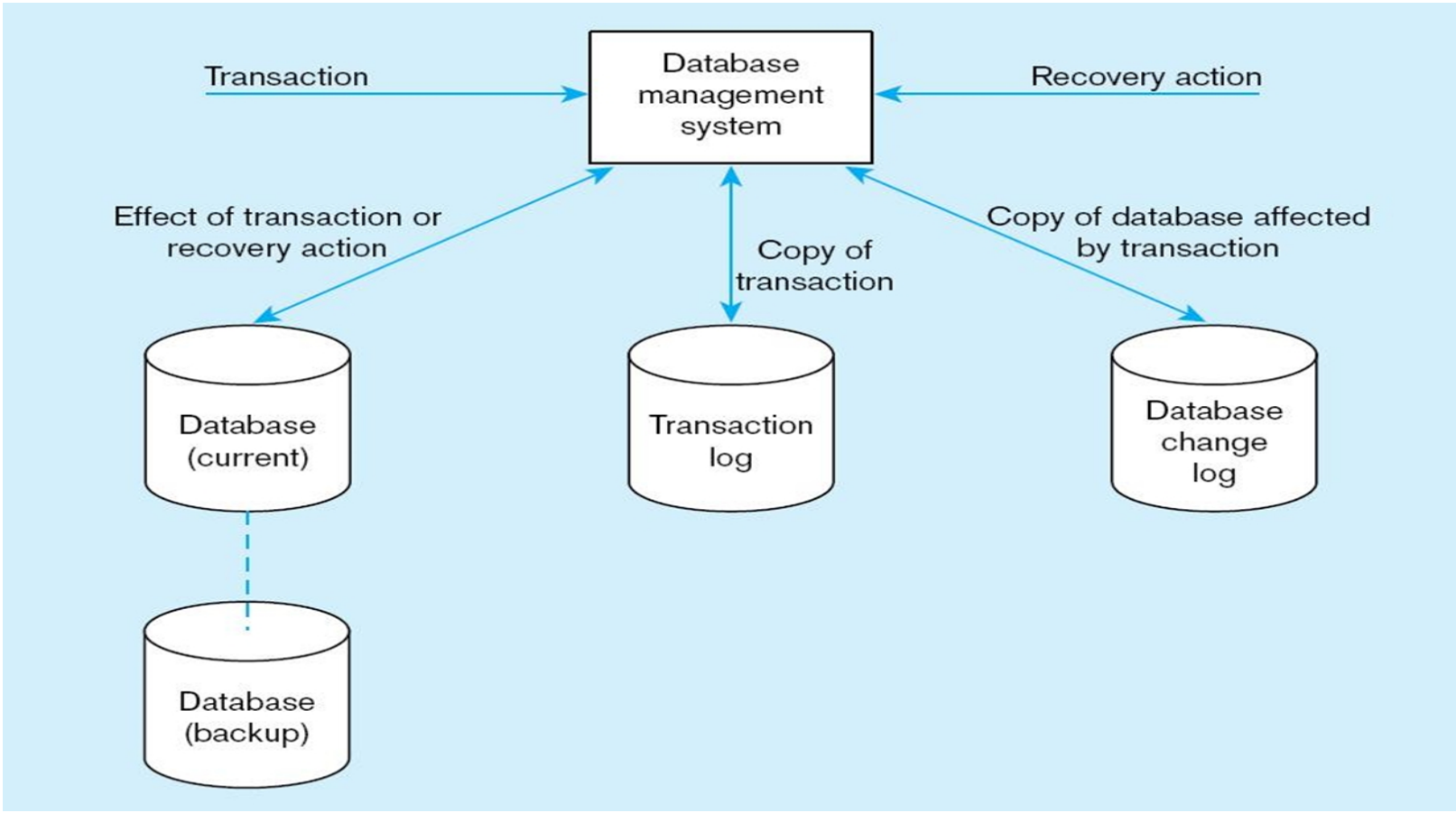
Effect of no concurrency control

E.g. Two customers withdrawing funds from the same account at the same time - account has $1000 in it, and they withdraw $200 and $300.



Functions of a DBMS

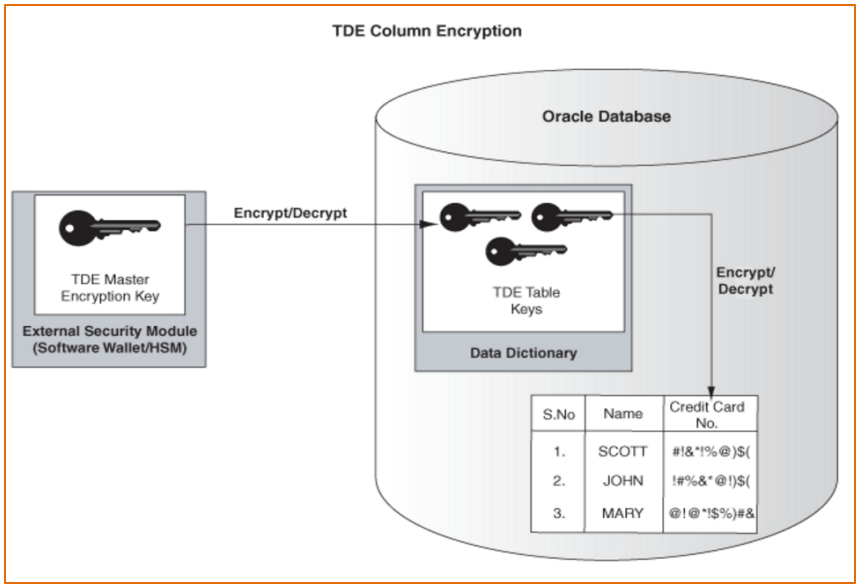
- **Backup and Recovery Services.**
  - **Crash, fire, natural disaster**
  - **So data not permanently lost**



Functions of a DBMS

Security

- Prevent unauthorized users accessing the database.
- Encryption, authentication, authorization and views



Functions of a DBMS

- **Integrity Services.**
  - Avoid incorrect or inconsistent data by complying a set of rules to ensure data integrity is enforced.

```
CREATE TABLE PropertyForRent(
  propertyNo VARCHAR(5) NOT NULL,
  rent NUMBER NOT NULL DEFAULT  600,
  staffNo VARCHAR(5)
    CONSTRAINT StaffNotHandlingTooMuch
      CHECK (NOT EXISTS (SELECT staffNo
        FROM PropertyForRent
        GROUP BY staffNO
        HAVING COUNT(*) > 100)),
  PRIMARY KEY (propertyNo),
  FOREIGN KEY (staffNo) REFERENCES Staff
```

Types Of Integrity Constraint In Oracle

- A NOT NULL constraint prohibits a database value from being null.
- A unique constraint prohibits multiple rows from having the same value in the same column or combination of columns but allows some values to be null.
- A primary key constraint combines a NOT NULL constraint and a unique constraint in a single declaration.
- A foreign key constraint requires values in one table to match values in another table.
- A check constraint requires a value in the database to comply with a specified condition.
  - CHECK (supplier_id BETWEEN 100 and 9999)
  - CHECK (supplier_name IN ('IBM', 'Microsoft', 'NVIDIA')
  - CHECK (status NOT IN ('A','B','C','D'));

```
CREATE TABLE OrderDetail (
  OrderDetailKey  VARCHAR(10) NOT NULL,
  OrderKey        VARCHAR(10),
  ProductKey      VARCHAR(10),
  Quantity        NUMBER(3) check (Quantity >=1),
  PriceCharged    NUMBER(6,2) check (PriceCharged > 0),
  constraint OrderToOrddet unique (OrderKey, ProductKey),
  Primary key (OrderDetailKey),
  Foreign Key (OrderKey) references CustomerOrder(OrderKey),
  Foreign Key (ProductKey) references Product (ProductKey)
);
```


14

Functions of a DBMS

- **Transaction Support**
  - **Ensure that all the updates are made or that none of them is made.**

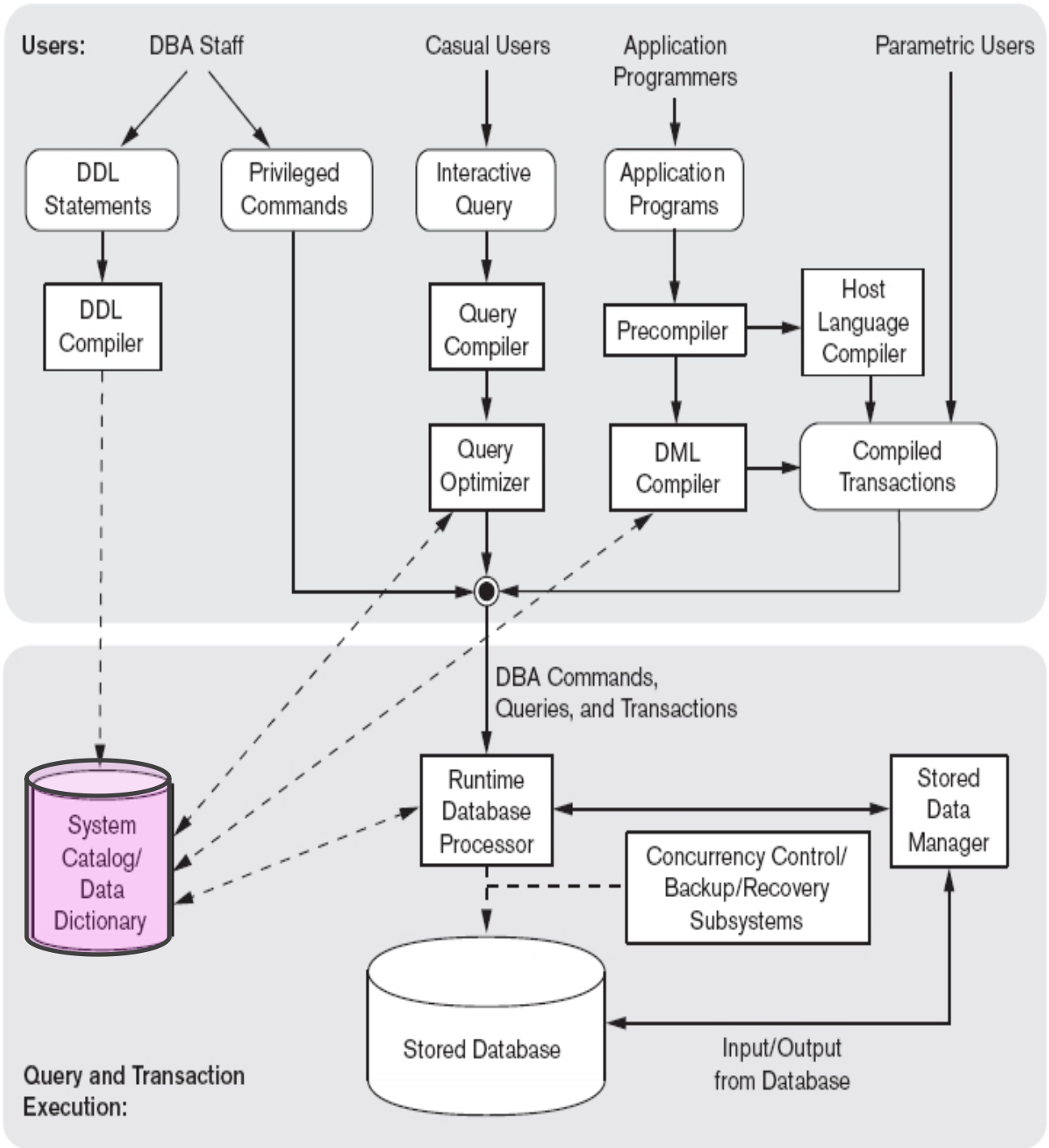
Airline Transaction Example

```
1.    BEGIN-TRANS
2.      Display greeting
3.      Get reservation preferences from user
4.      SELECT departure and return flight records
5.      If reservation is acceptable then
6.          UPDATE seats remaining of departure flight record
7.          UPDATE seats remaining of return flight record
8.          INSERT reservation record
9.          Print ticket if requested
10.     End If
11.   On Error: ROLLBACK
```

Functions of a DBMS

- **A User-Accessible System Catalogue (Data Dictionary Management).**

- **System Catalogue**
  - **Repository of information (metadata) describing the data in the database.**
  - **One of the fundamental components of DBMS.**
  - **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 (eg counts on the number of accesses made to objects in the database)**



DBMS consults the system catalog before performing almost every action:

- **Authorize users**
- **Process queries**
- **Check integrity**
- **etc**

Sample Oracle Catalog Tables

Table Name	Contents
USER_CATALOG	Contains basic data about each table and view defined by a user.
USER_OBJECTS	Contains data about each object (functions, procedures, indexes, triggers, assertions, etc.) defined by a user. This table contains the time created and the last time changed for each object.
USER_TABLES	Contains extended data about each table such as space allocation and statistical summaries.
USER_TAB_COLUMNS	Contains basic and extended data for each column such as the column name, the table reference, the data type, and a statistical summary.
USER_VIEWS	Contains the SQL statement defining each view.

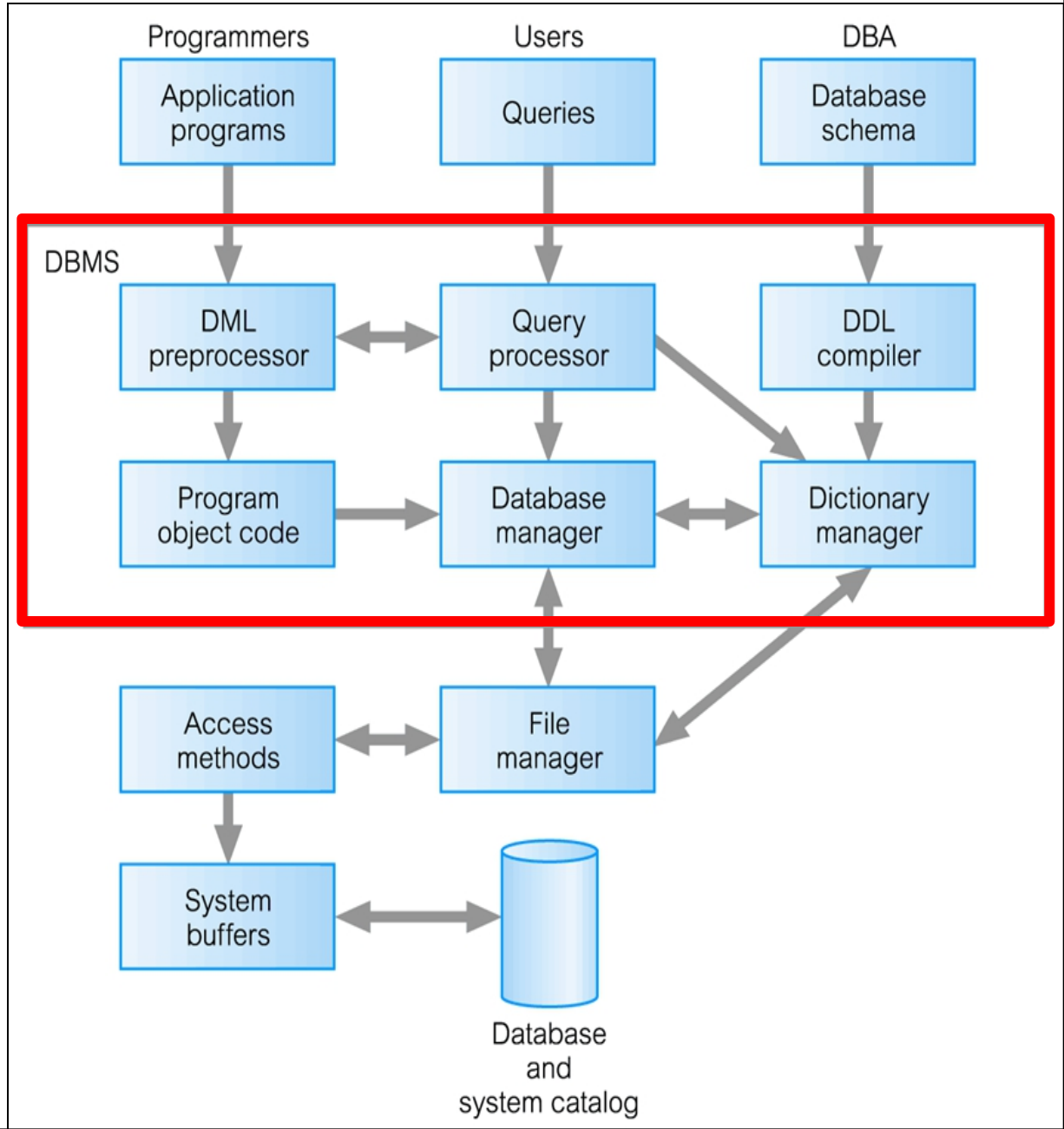
Oracle Metadata

```
SQL> desc all_users
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USERNAME                                  NOT NULL VARCHAR2(30)
 USER_ID                                   NOT NULL NUMBER
 CREATED                                   NOT NULL DATE
```

```
SQL> desc product
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 P_CODE                                    NOT NULL VARCHAR2(10)
 P_DESCRIPT                                NOT NULL VARCHAR2(35)
 P_INDATE                                  NOT NULL DATE
 P_ONHAND                                  NOT NULL NUMBER
 P_MIN                                     NOT NULL NUMBER
 P_PRICE                                   NOT NULL NUMBER(8,2)
 P_DISCOUNT                                NOT NULL NUMBER(4,2)
 V_CODE                                             NUMBER
 P_REORDER                                          NUMBER(1)
```

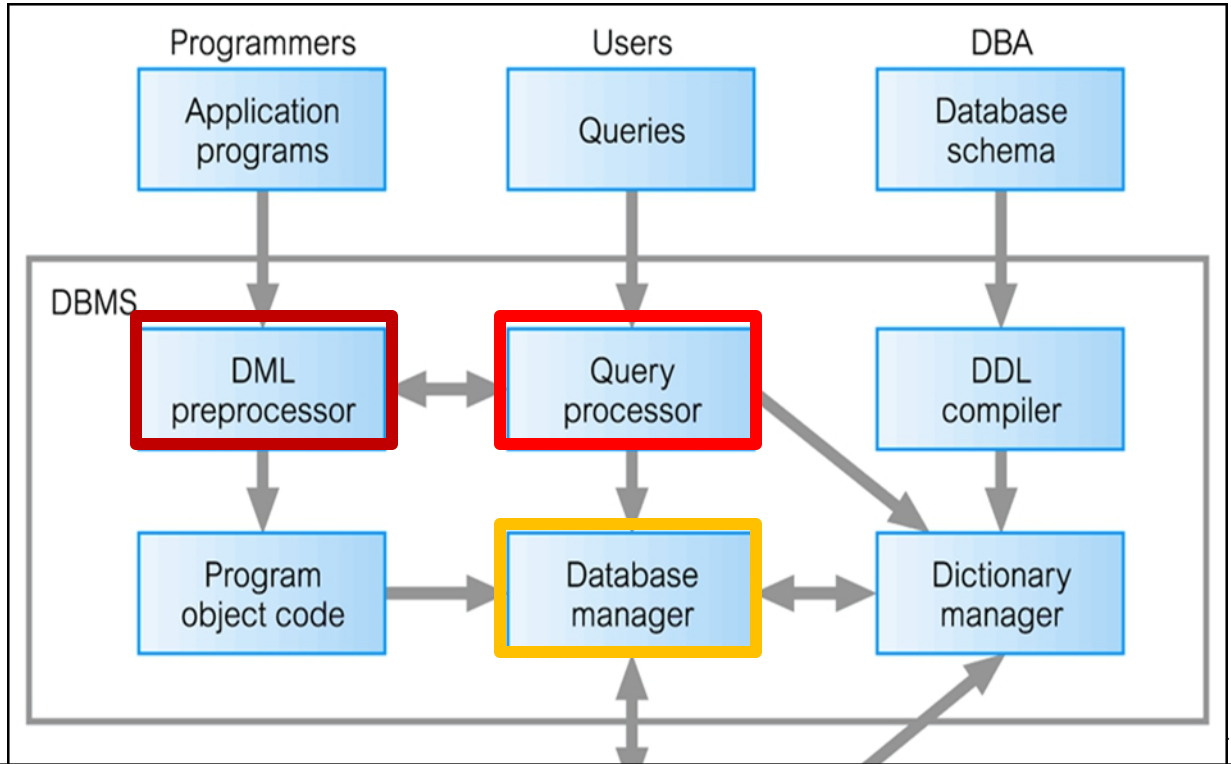
2.2 Components of a DBMS

- **A DBMS is partitioned into several software components (or *modules),* each of which is assigned a specific operation.**
- **The DBMS interfaces with other software components, such as user queries and access methods.**



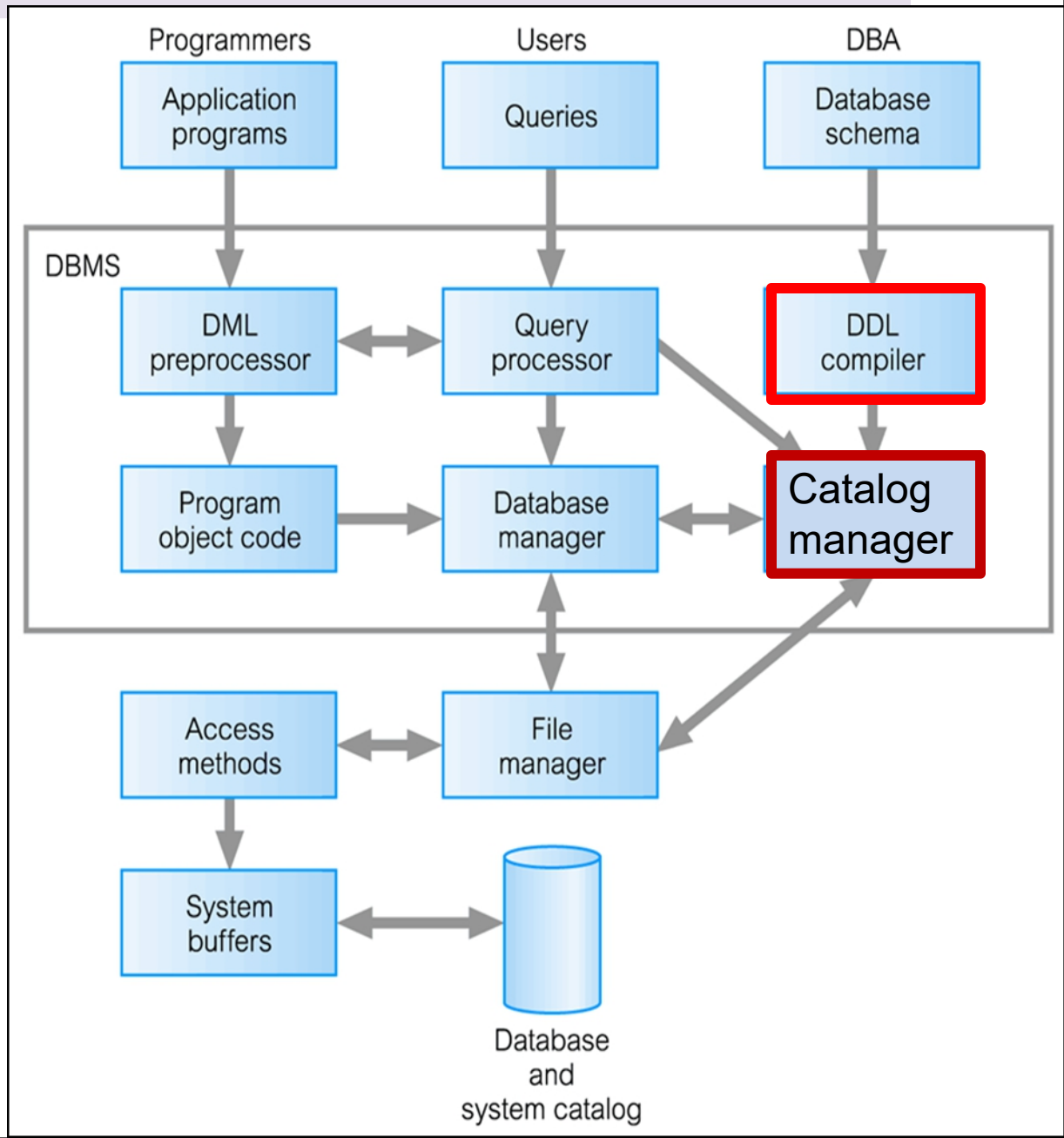
Components of a DBMS

- *Query processor* - transforms queries into low-level instructions directed to the database manager.
- *Database manager*
- *DML preprocessor* converts DML statements embedded in an application program into standard function calls in the host language.

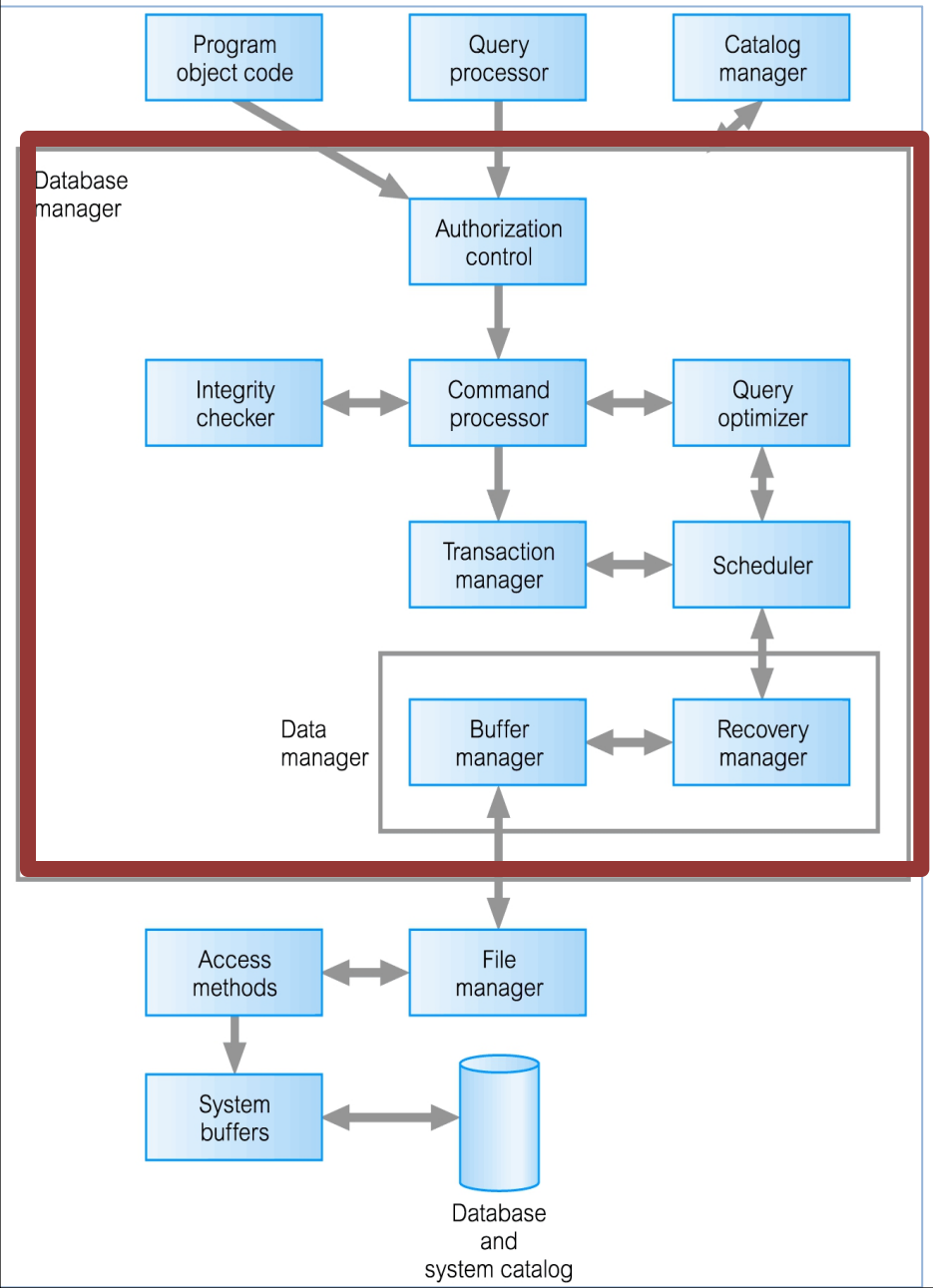


Components of a DBMS

- *DDL compiler* converts DDL statements into a set of tables containing metadata. These tables are then stored in the system catalog
- *Catalog manager* manages access to and maintains the system catalog.



Components of Database Manager



- ***Authorization control*.**
- ***Command processor***
- ***Integrity checker***
- ***Query optimizer*** **determines an optimal strategy for the query execution.**
- ***Transaction manager***
- ***Scheduler*** **ensures concurrent operations proceed without conflicting with one another.**
  Recovery manager
- ***Buffer manager*** **responsible for the transfer of data between main memory and secondary storage, such as disk and tape.**

3. ANSI-SPARC Architecture

The **ANSI-SPARC Architecture (**ANSI-SPARC stands for *American National Standards Institute, Standards Planning And Requirements Committee)* is an abstract **design standard** for a Database Management System (DBMS), first proposed in 1975.

Most modern commercial DBMS are based on this standard.

ANSI-SPARC Three-Level Architecture

External Level

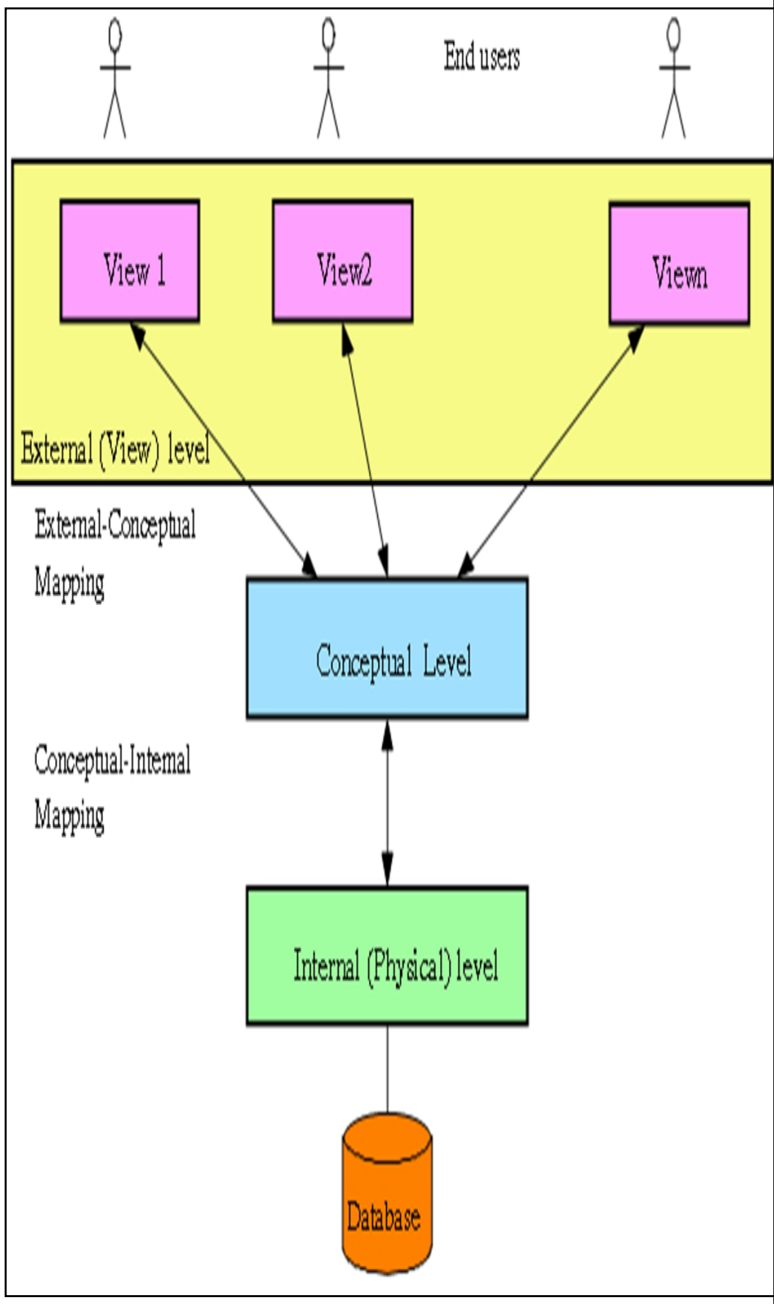
- Users' view of the database.
- Describes that part of database that is relevant to a particular user.

Conceptual Level

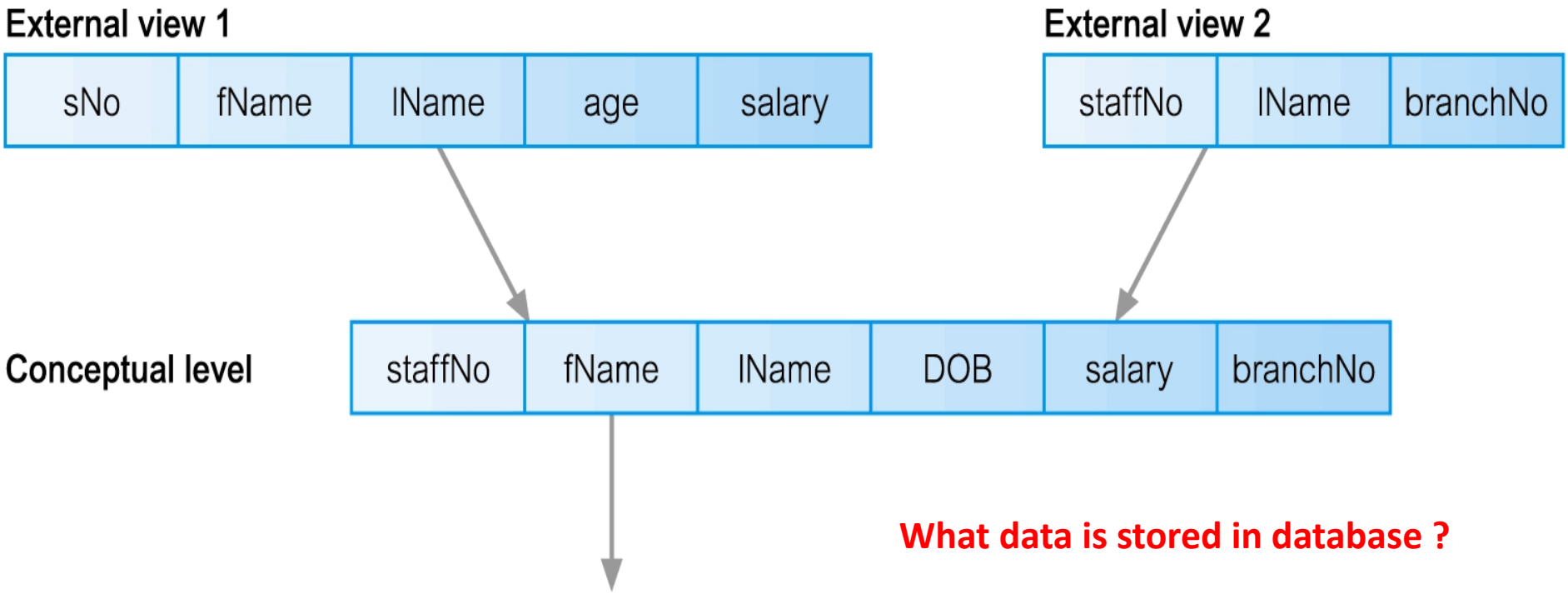
- Community view of the database.
- Describes what data is stored in database and relationships among the data.

Internal Level

- Physical representation of the database on the computer.
- Describes how the data is stored in the database.



The Three Levels of ANSI-SPARC Architecture



What data is stored in database ?

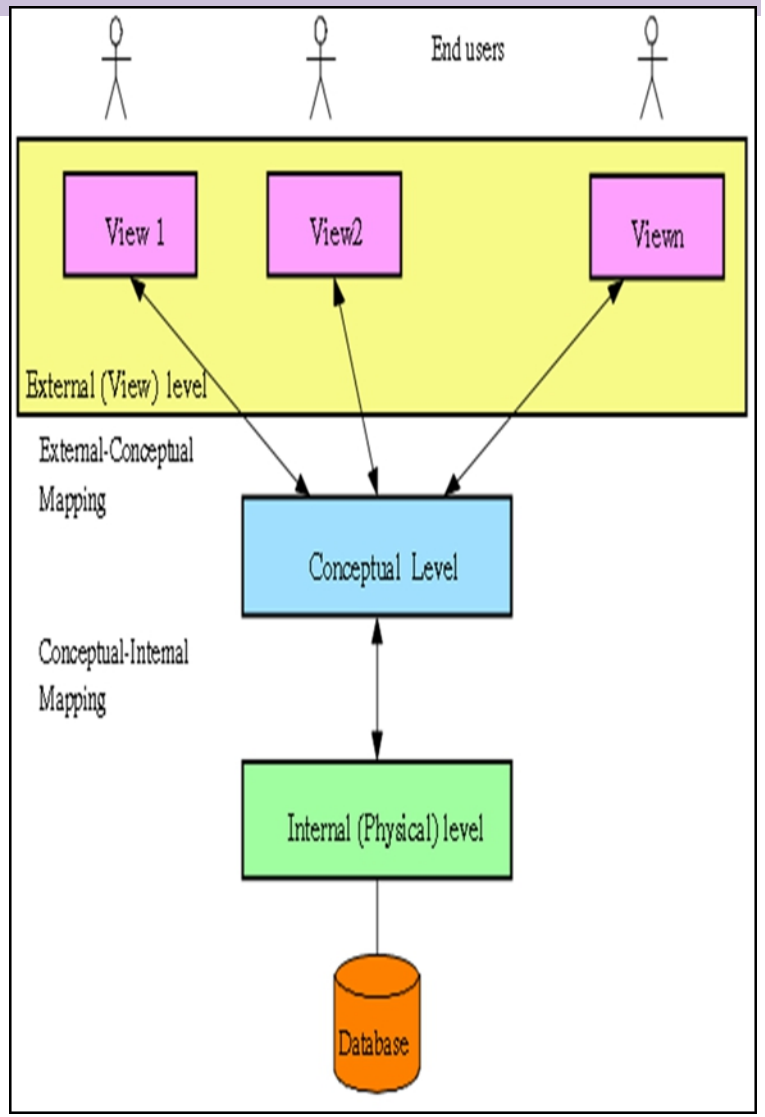
Internal level

```
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 */
```

How data is stored in database ?

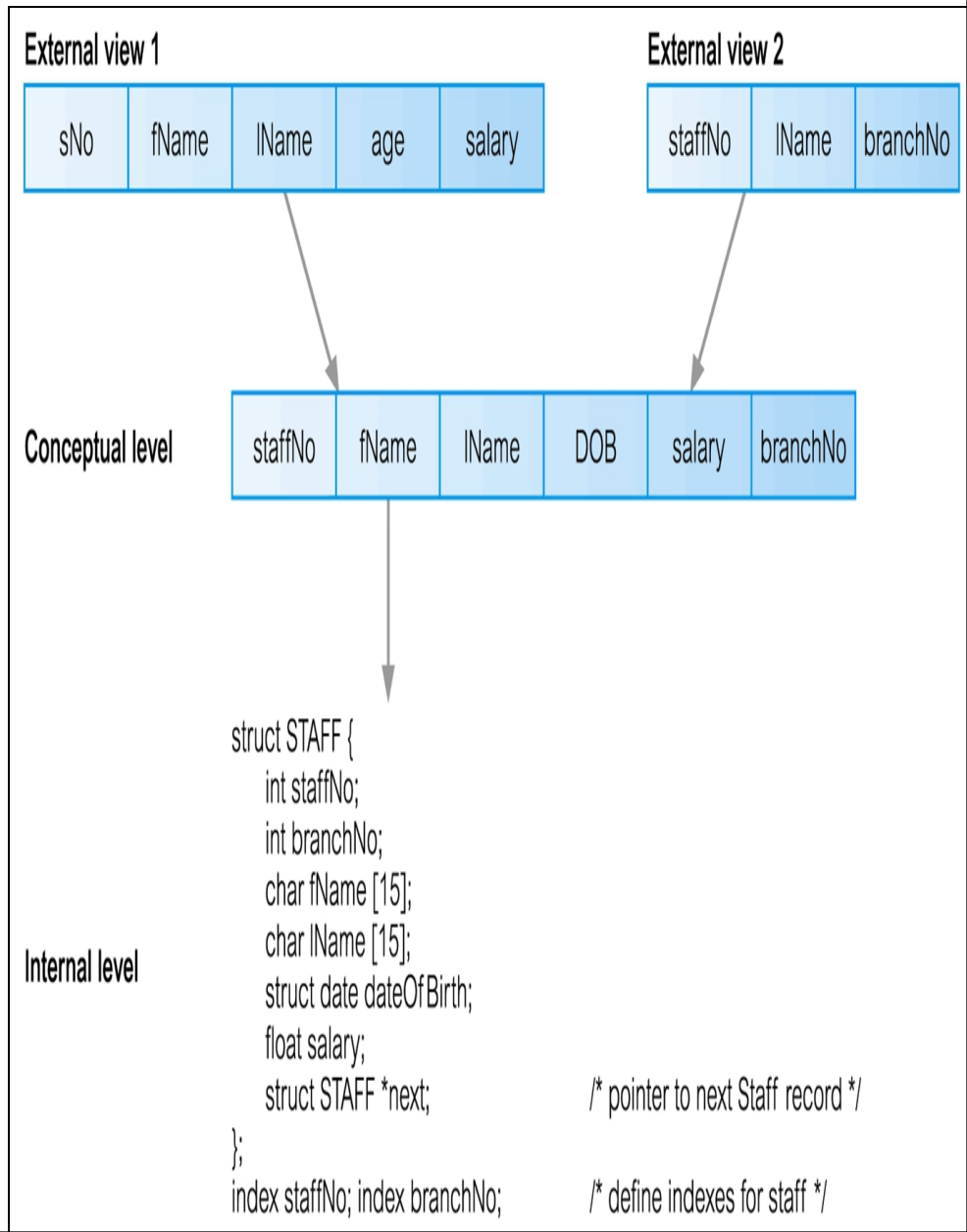
Objectives of Three-Level Architecture

- **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.**



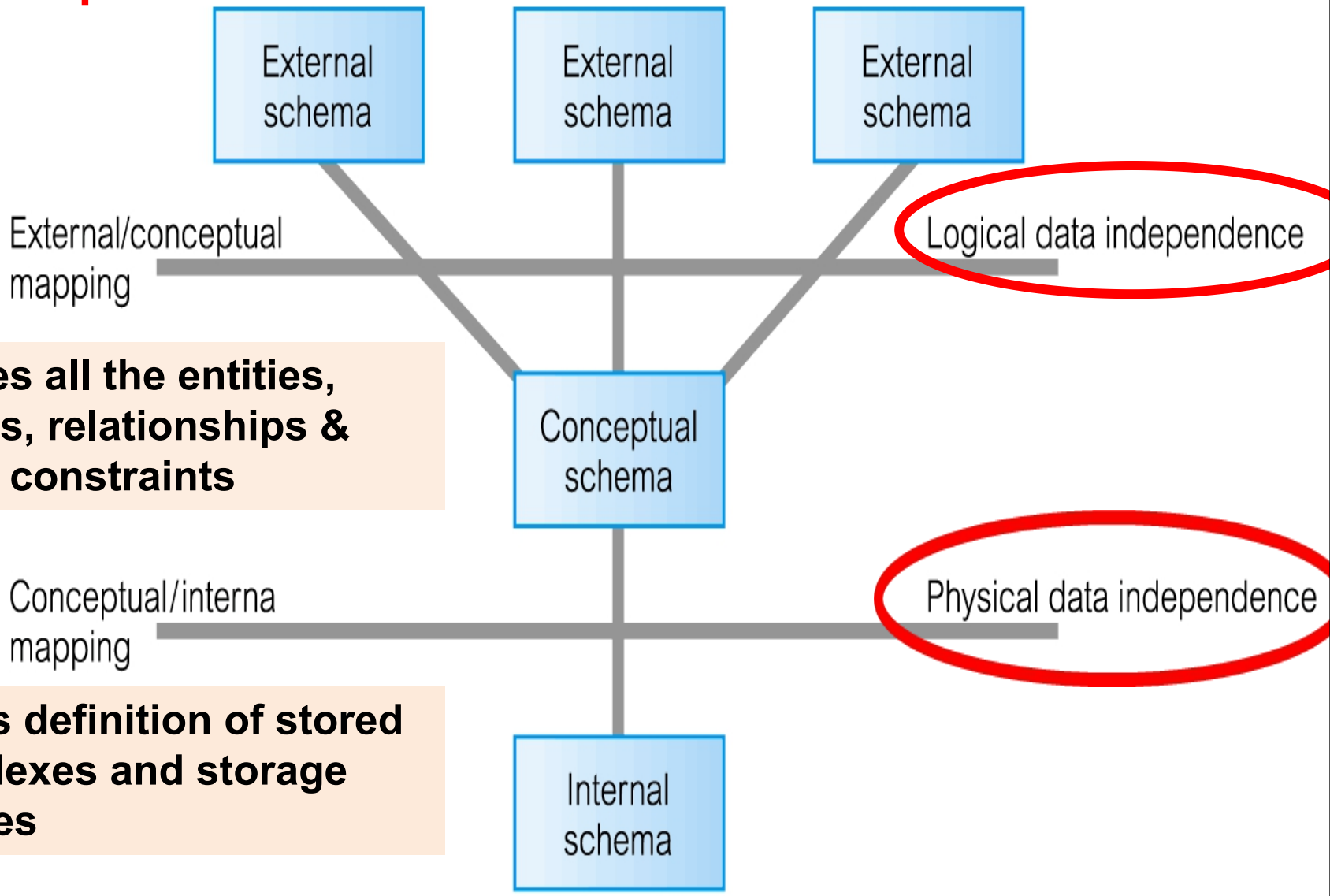
Objectives of Three-Level Architecture

- **DBA should be able to change database storage structures without affecting the users' views.**
- **Internal structure of database should be unaffected by changes to physical aspects of storage.**
- **DBA should be able to change conceptual structure of database without affecting all users.**



Data Independence and the ANSI-SPARC Three-Level Architecture

A major objective for the three-level architecture is to provide data independence



Describes all the entities, attributes, relationships & integrity constraints

Contains definition of stored data, indexes and storage structures

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/attributes) should not require changes to external schema or rewrites of application programs.

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.

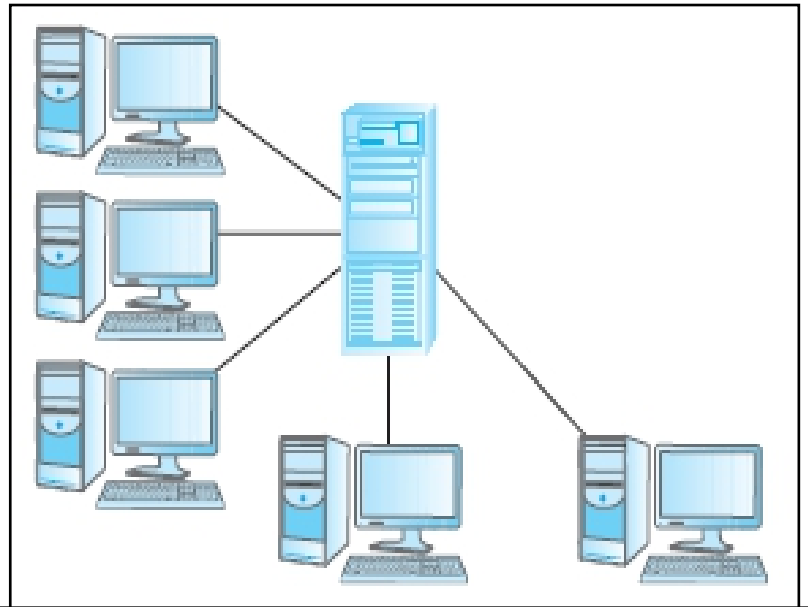
4. Multi-user DBMS Architectures

- The common architectures that are used to implement multi-user database management systems:
  - Teleprocessing
  - File-Server
  - Client-Server

31

Teleprocessing

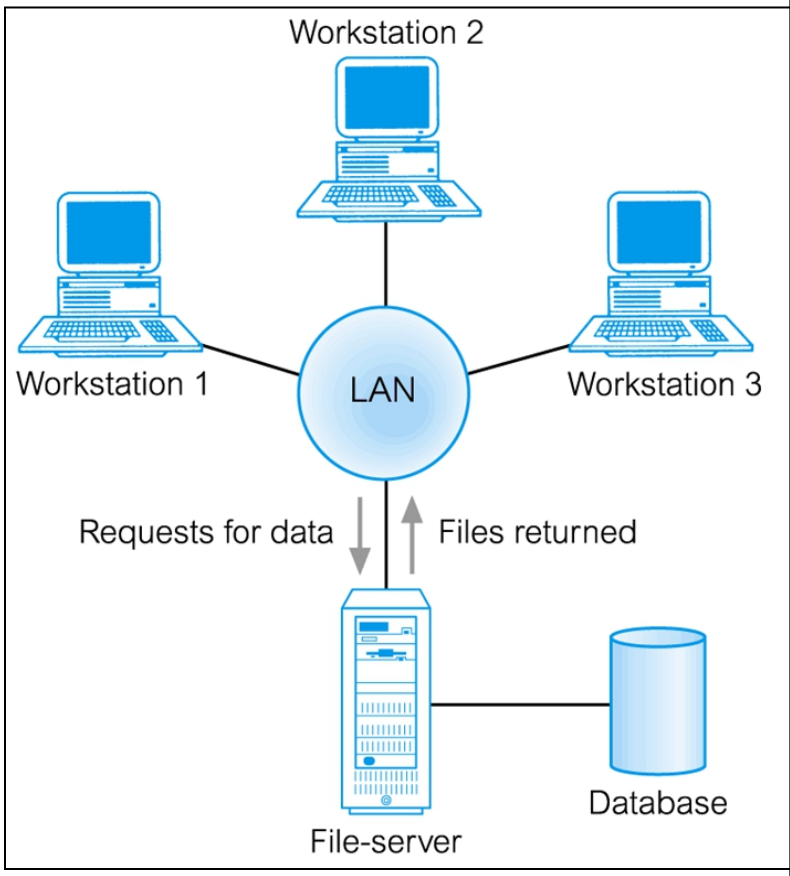
- One computer with a single CPU and a number of terminals.
- Processing performed within the same physical computer. User terminals are typically "dumb", incapable of functioning on their own, and cabled to the central computer.



32

File-Server Architecture

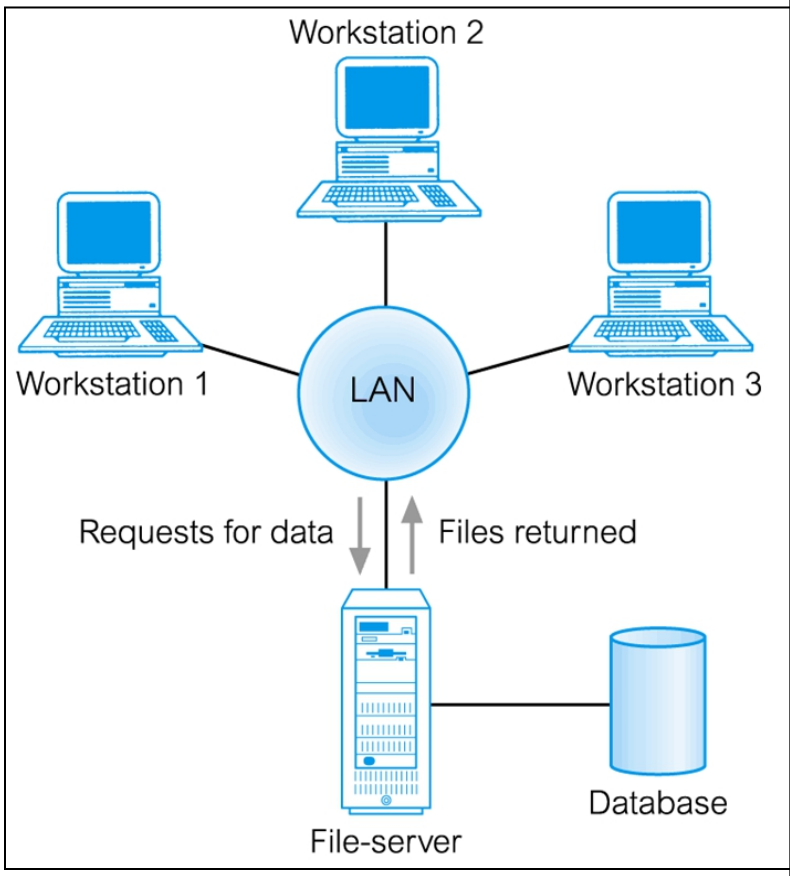
- File-server is connected to several workstations across a network.
- Database resides on file-server.
- DBMS and applications run on each workstation.



File-Server Architecture

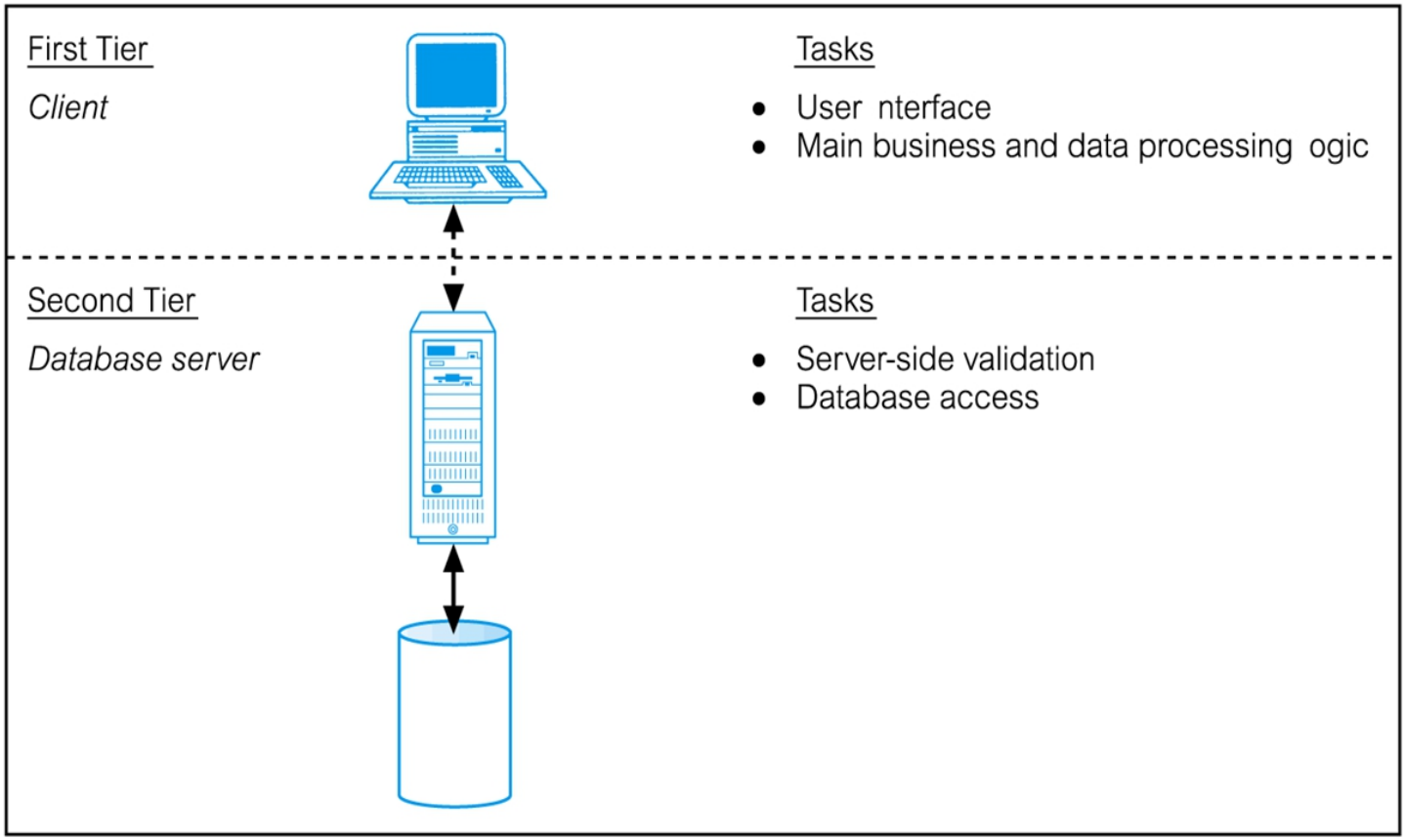
Disadvantages :

- **Significant network traffic.**
- **Copy of DBMS on each workstation.**
- **Concurrency, recovery and integrity control more complex (multiple DBMSs accessing the same files.)**

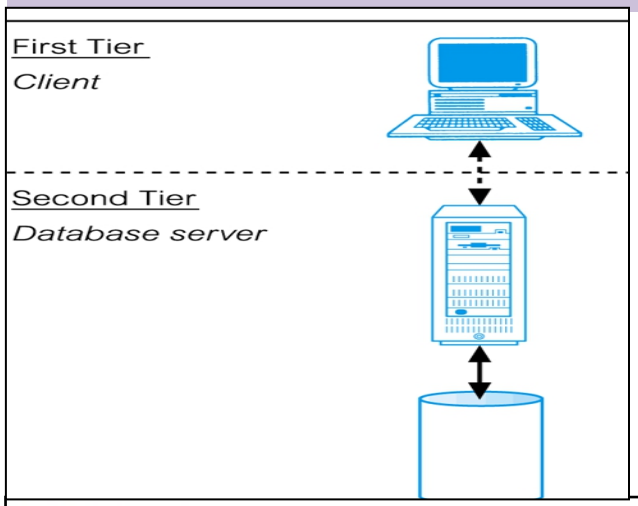


Traditional Two-Tier Client-Server

- **Client (tier 1)** manages user interface and runs applications.
- **Server (tier 2)** holds database and DBMS.



Summary of Client-Server Functions

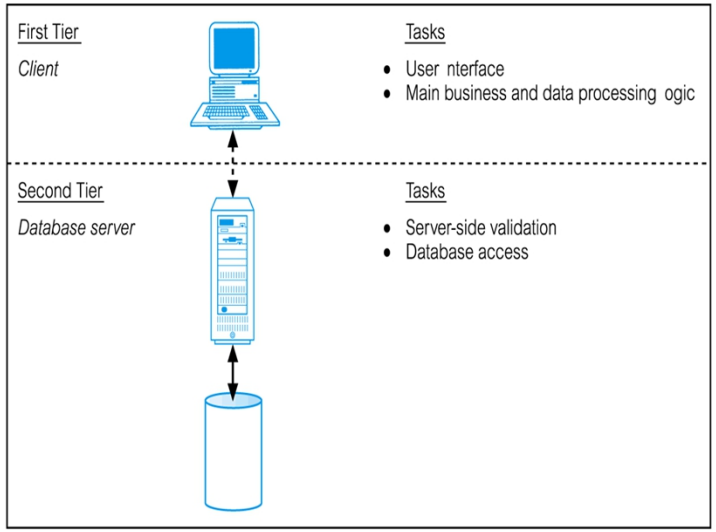


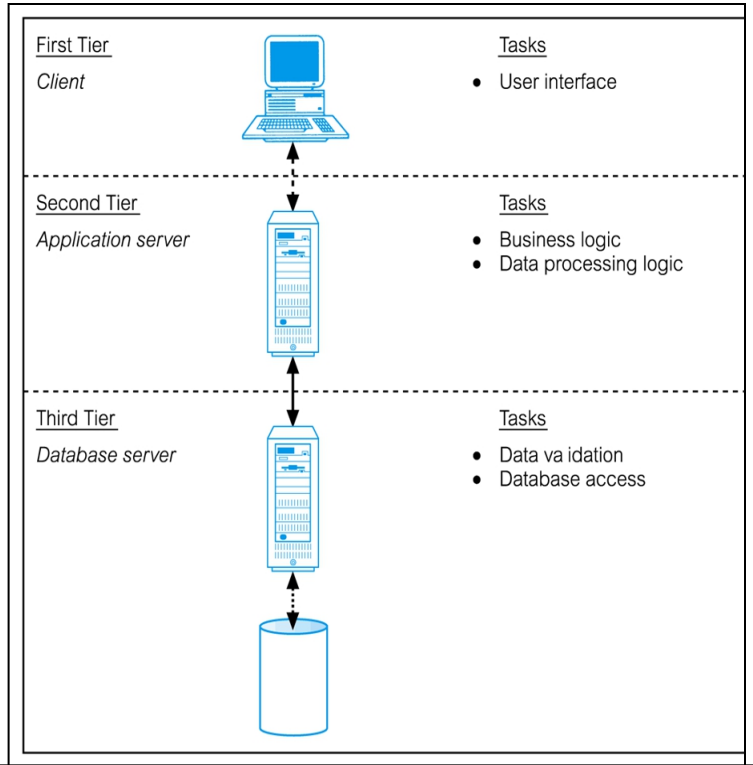
CLIENT	SERVER
Manages the user interface	Accepts and processes database requests from clients
Accepts and checks syntax of user input	Checks authorization
Processes application logic	Ensures integrity constraints not violated
Generates database requests and transmits to server	Performs query/update processing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

36

Three-Tier Client-Server

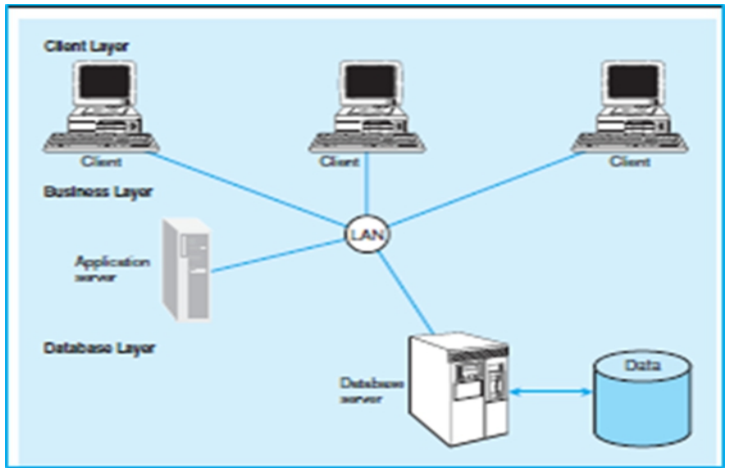
- 2-tier architecture presented two problems preventing true scalability:
  - ‘Fat’ client, requiring considerable resources on client’s computer to run effectively.
  - Significant client side administration overhead.
- By 1995, three layers proposed, each potentially running on a different platform.

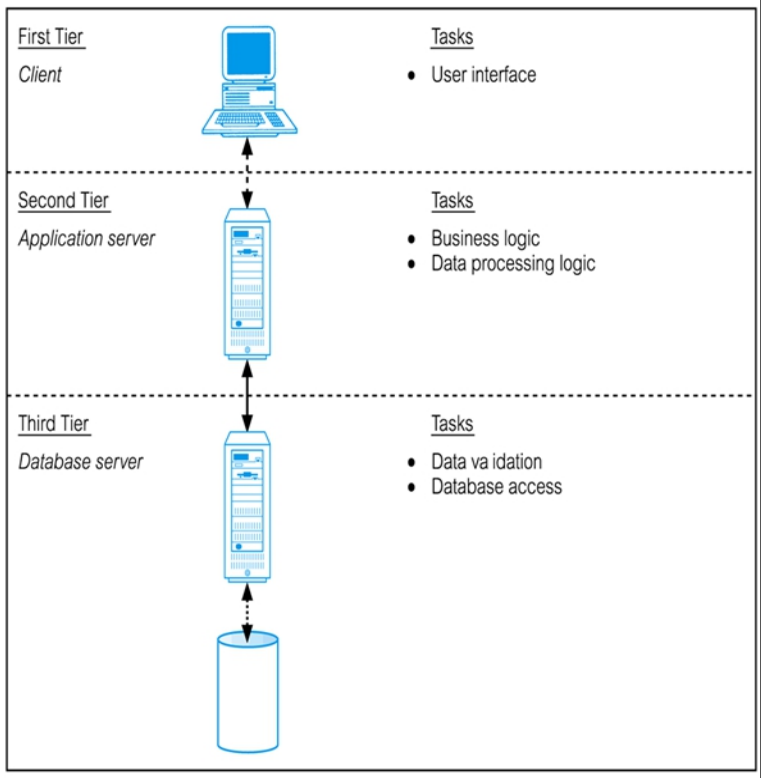




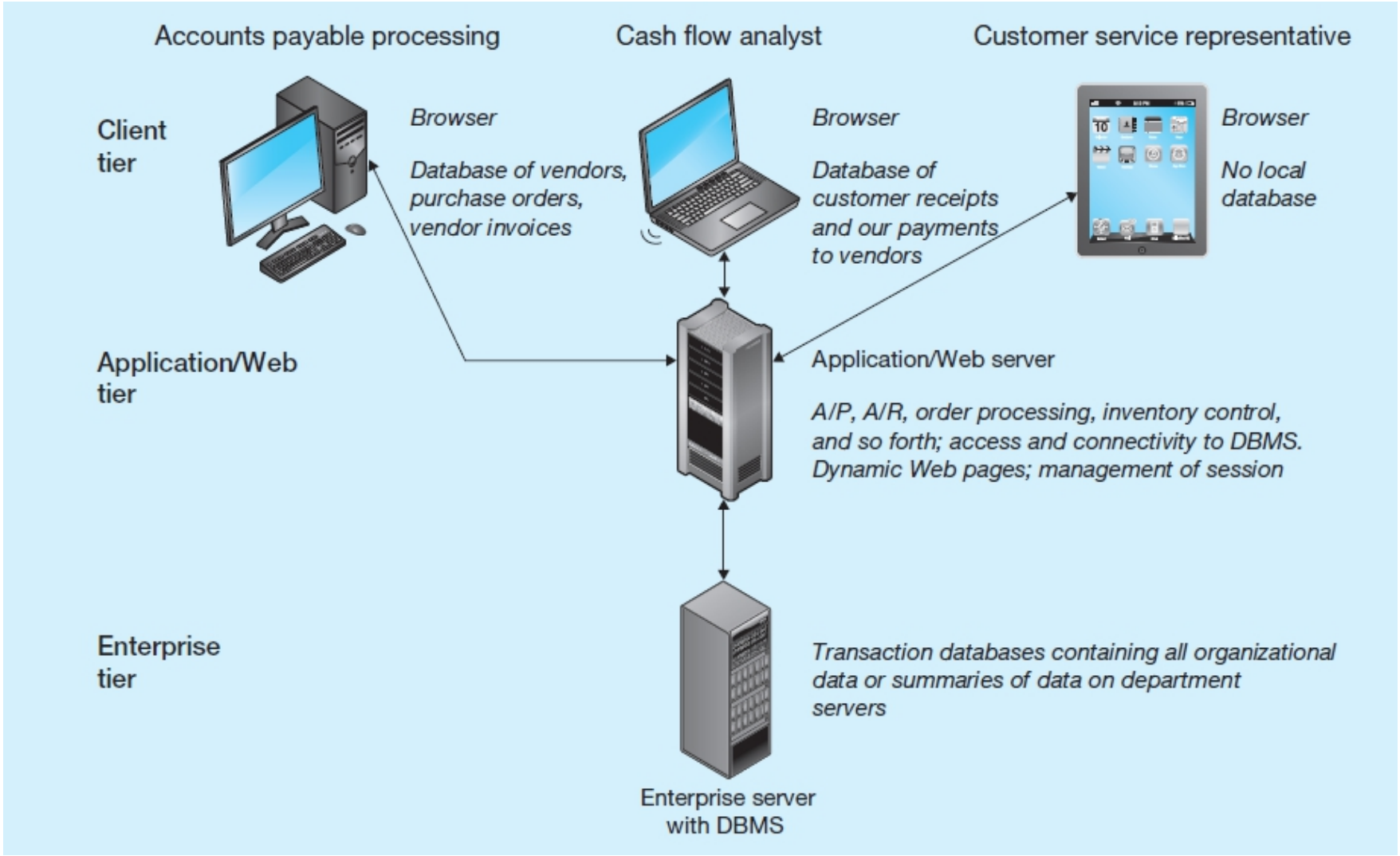
Three-Tier Client-Server

- **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.





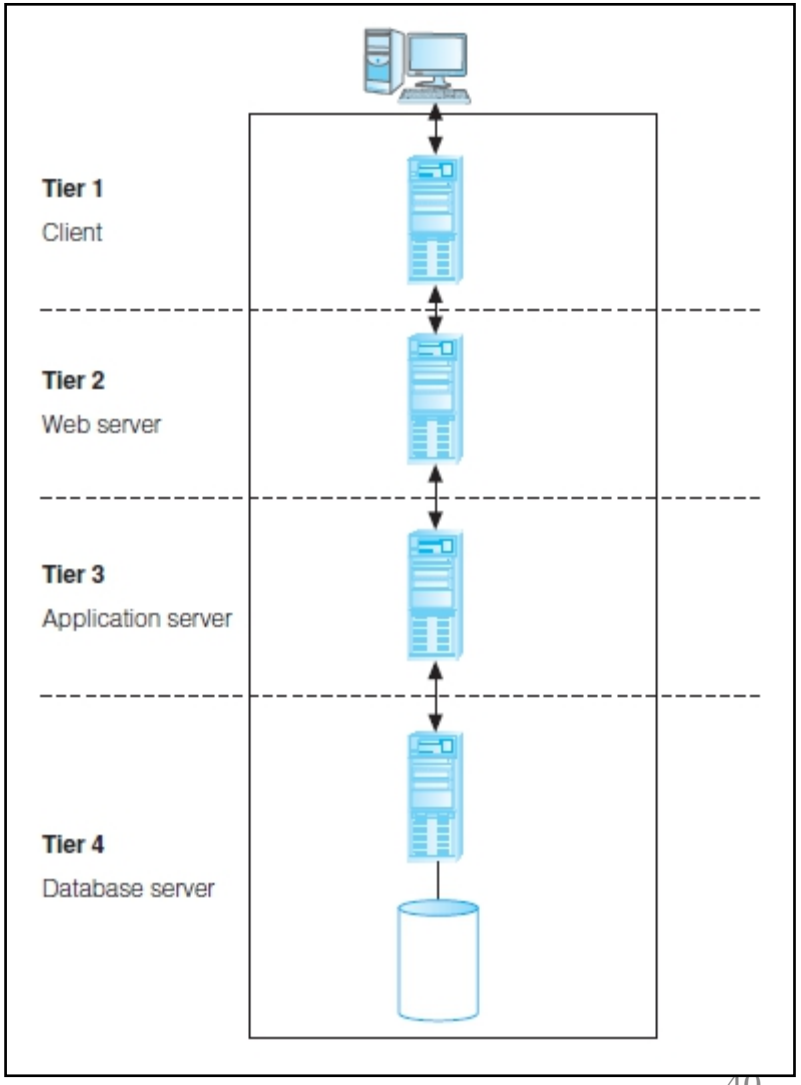
Three-Tier Client-Server



Modern Database Management, Hoffer

N-Tier Client-Server Architectures

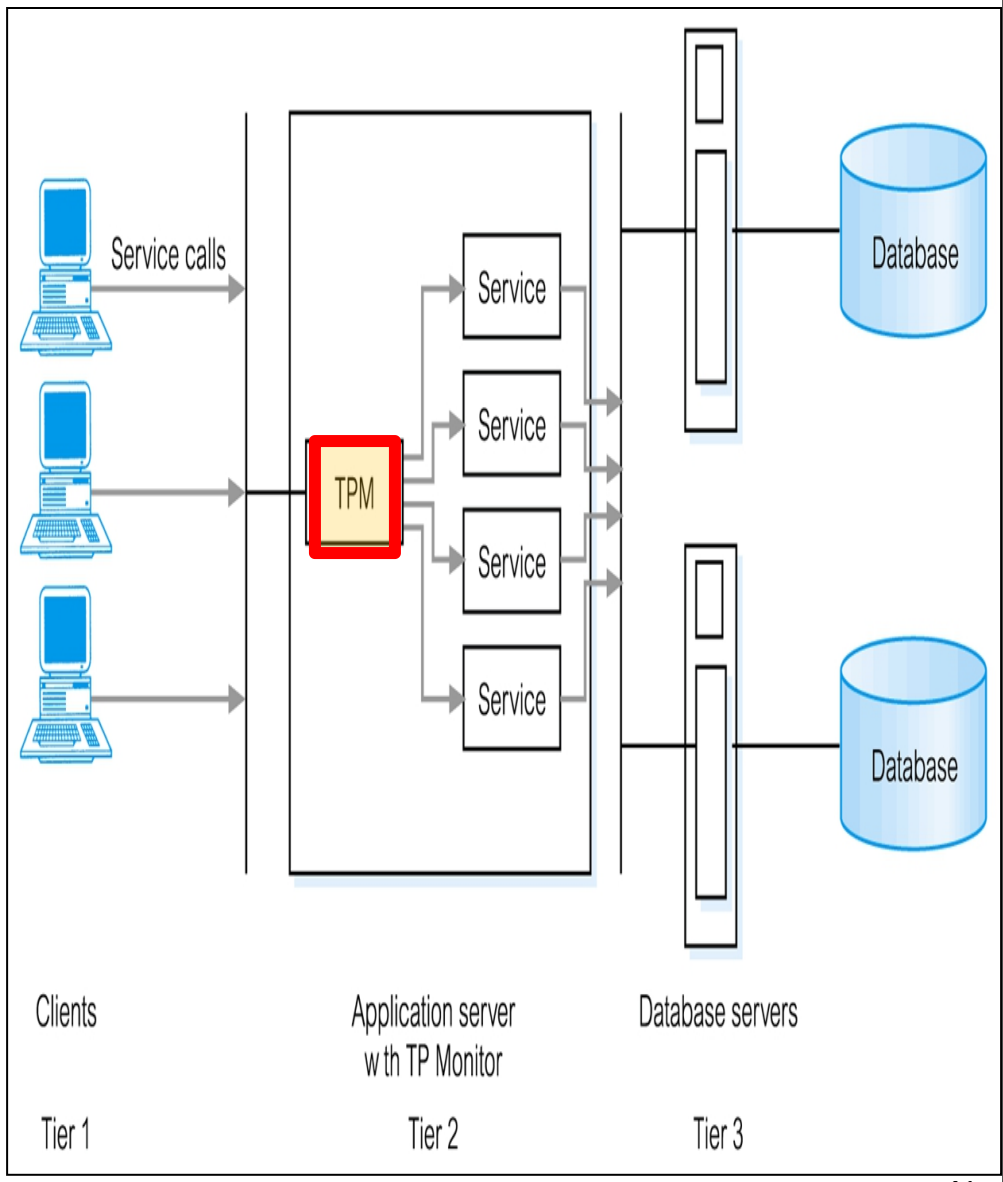
- The three-tier architecture can be expanded to *n* tiers, with additional tiers providing more flexibility and scalability.
- Applications servers host API to expose business logic and business processes for use by other applications.



40

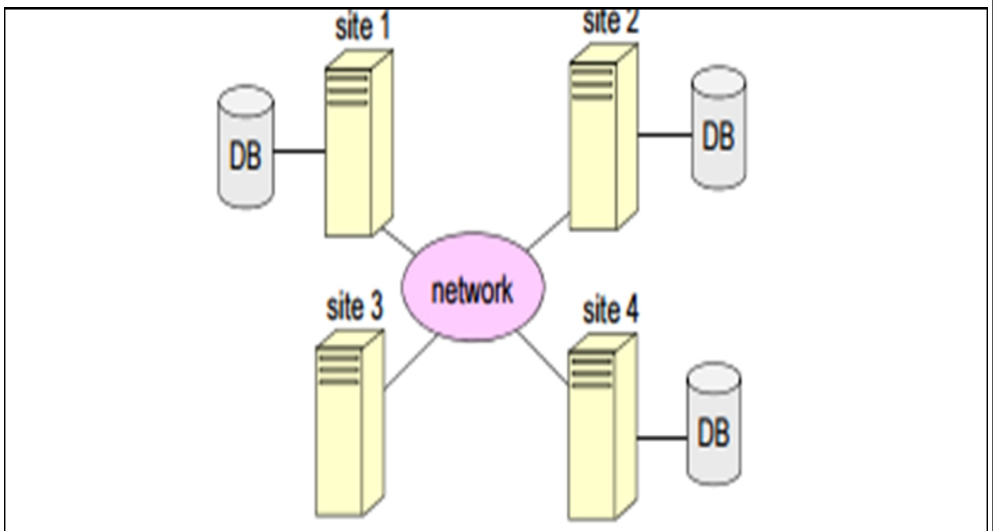
Transaction Processing Monitors

- **Forms the middle tier of a three-tier architecture**
- **TP monitor is a program that controls data transfer between clients and servers in order to provide a consistent environment, particularly for Online Transaction Processing (OLTP).**



Distributed DBMSs

- **A distributed database is a logically interrelated collection of shared data (and a description of this data), physically distributed over a computer network.**
- **Distributed DBMS permits the management of the distributed database and makes the distribution transparent to users.**



- A given site may have a local copy of all, part, or none of a particular database.

42

References

- ***Database Systems: A Practical Approach to Design, Implementation and Management*.** **Connolly, T. M. and Begg, C. E. .**
- ***Modern Database Management.*** **Hoffer, J.A., Prescott, M., and McFadden,F.**
- ***Database System Concepts*** **. Silberschatz, A., Korth, H,. and Sudarshan, S.**
- ***Fundamentals of Database Systems.*** **Elmasri, R. and Navathe, S.B.**

43