

# ICS 434

## Advanced Database Systems

Dr. Abdallah Al-Sukairi  
sukairi@kfupm.edu.sa

Second Semester 2003 - 2004 (032)

King Fahd University of Petroleum & Minerals  
Information & Computer Science Department



## Outline

1. The Relational Data Model: Version 2
2. Advanced Data Modeling
3. **Client-Server Architecture**
4. Client-Server Databases & Tools
5. Databases on the Web
6. The System Catalog
7. Query Processing and Optimization
8. Transaction Processing
9. Concurrency Control
10. Recovery
11. Administration & Security
12. Distributed Databases
13. Database Replication
14. Object-Oriented Databases
15. Data Warehousing and Data Mining
16. Other Emerging Database Technologies



### 3. Client-Server Architecture



#### Classification of DBMSs

- Mainframe (host-based) DBMSs
- PC-Based DBMSs
  - Single user
  - File/Server
- Client-Server Database Systems
- Distributed DBMSs



## Mainframe (Host-Based) DBMSs

- Multi-user environment
- Information sharing
- Centralized data management
- Sophisticated administration and security features
- Advanced operating system features



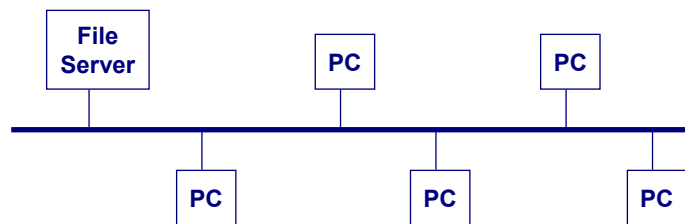
## PC-Based DBMSs

- Graphical User Interface (GUI)
- Ease of use
- Outstanding price/performance ratio
- Advanced communication
- Individual environment
- Increasingly powerful hardware and software
- RISC-based workstations
- Multiprocessing systems

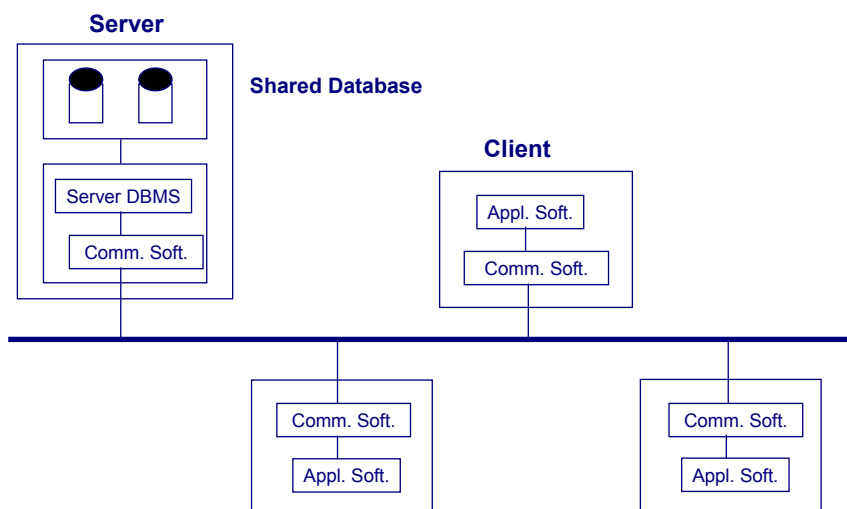


## File/Server

- Multi-User (LAN version)



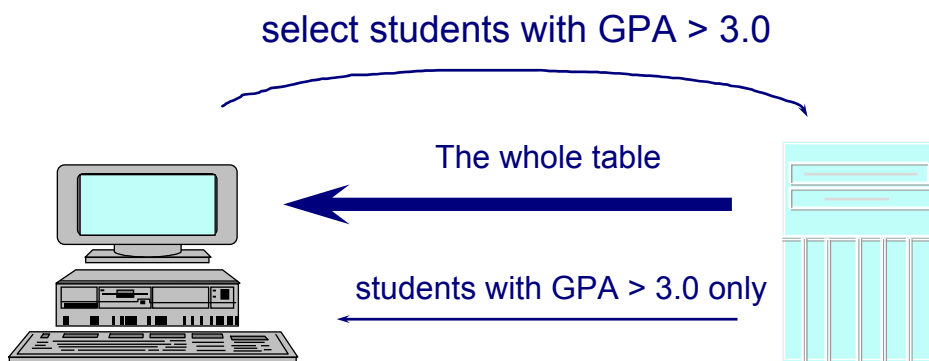
## Client-Server Database Systems



## File/server vs. Client-Server

User station

Server



## Why Client-Server ?

- Mainframe Computing
- Desktop Computing
- Client-Server computing combines the benefits of both worlds
- Downsizing trend
- Scalability
- Openness
- Enterprise-wide computing model

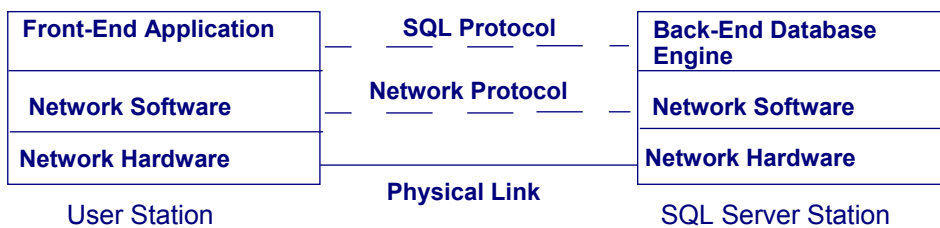


## ... Why Client-Server ?

- A form of distributed processing
- Hardware
  - LAN
  - Back-end Server
  - Front-end station
- Software
  - Communication software
  - Back-end software
  - Front-end tool
- Applications
  - Client-Server Databases
  - E-mail software
  - GroupWare



## Client-Server Database Model





## Components of Data-Intensive Systems

- Three separate types of functionality:
  - Data management
  - Application logic
  - Presentation
- The system architecture determines whether these three components reside on a single system ("tier") or are distributed across several tiers



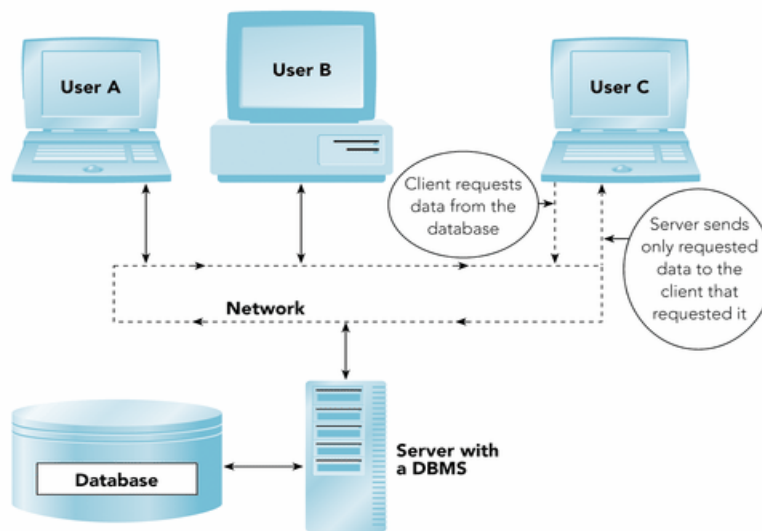
## Single-Tier Architectures

- All functionality combined into a single tier, usually on a mainframe
  - User access through dumb terminals
- Advantages:
  - Easy maintenance and administration
- Disadvantages:
  - Today, users expect graphical user interfaces.
  - Centralized computation of all of them is too much for a central system

## Client-Server Architectures

- Work division: Thin client
  - Client implements only the graphical user interface
  - Server implements business logic and data management
  
- Work division: Thick client
  - Client implements both the graphical user interface and the business logic
  - Server implements data management

## Two-Tier Client/Server Architecture





## Components of Client/Server Architecture

- Client
  - Front-end application
- Server
  - Back-end application



## ... Client-Server Architectures

- Disadvantages of thick clients
  - No central place to update the business logic
  - Security issues: Server needs to trust clients
    - Access control and authentication needs to be managed at the server
    - Clients need to leave server database in consistent state
    - One possibility: Encapsulate all database access into stored procedures
  - Does not scale to more than several 100s of clients
    - Large data transfer between server and client
    - More than one server creates a problem:  $x$  clients,  $y$  servers:  $x*y$  connections



## The Three Layers

- **Presentation tier**
  - Primary interface to the user
  - Needs to adapt to different display devices (PC, PDA, cell phone, voice access?)
- **Middle tier**
  - Implements business logic (implements complex actions, maintains state between different steps of a workflow)
  - Accesses different data management systems
- **Data management tier**
  - One or more standard database management systems



## The Three-Tier Architecture

Presentation tier

Client Program (Web Browser)

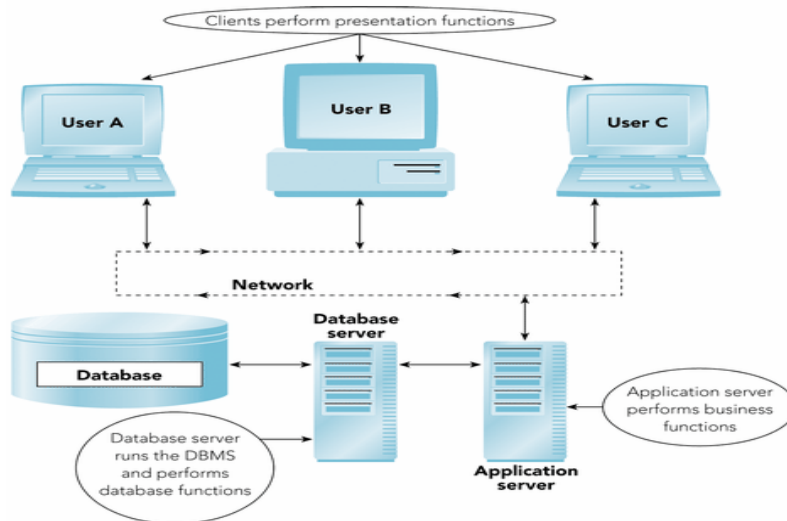
Middle tier

Application Server

Data management  
tier

Database System

## Three-Tier Client/Server Architecture



## Advantages of the Three-Tier Architecture

- Heterogeneous systems
  - Tiers can be independently maintained, modified, and replaced
- Thin clients
  - Only presentation layer at clients (web browsers)
- Integrated data access
  - Several database systems can be handled transparently at the middle tier
  - Central management of connections
- Scalability
  - Replication at middle tier permits scalability of business logic
- Software development
  - Code for business logic is centralized
  - Interaction between tiers through well-defined APIs: Can reuse standard components at each tier



## Example 1: Airline reservations

- Build a system for making airline reservations
- What is done in the different tiers?
- Database System
  - Airline info, available seats, customer info, etc.
- Application Server
  - Logic to make reservations, cancel reservations, add new airlines, etc.
- Client Program
  - Log in different users, display forms and human-readable output



## Example 2: Course Enrollment

- Build a system using which students can enroll in courses
- Database System
  - Student info, course info, instructor info, course availability, pre-requisites, etc.
- Application Server
  - Logic to add a course, drop a course, create a new course, etc.
- Client Program
  - Log in different users (students, staff, faculty), display forms and human-readable output



## Client/Server DBMS Functions

- Transparent data access to multiple, heterogeneous clients
- Allow client requests to the database server over network
- Process client data requests at local server
- Send only SQL results to clients over network



## Summary: Advantages of Client/Server Systems

- Lower network traffic
- Improved processing distribution
- Thinner clients
- Greater processing transparency
- Increased network, hardware, and software transparency
- Improved security
- Decreased costs
- Increased scalability