CSCI 305 – F13, Lecture 2

[Slides originally assembled by Prof. King]

The DBMS

DBMS

- **Database Management System**
  - DBMS
  - Specialized software design for the sole purpose of providing services to store, manage, and query data in an efficient manner

Task #1 – Create new DB

- A DBMS allows a DB administrator to specify the DB **schema**
  - Schema is a DB blueprint
  - Uses some type of **data definition language (DDL)**
  - Is usually a part of the **query language** (e.g., SQL)
  - Example:

```
CREATE TABLE employees (
    id              INTEGER     PRIMARY KEY,
    first_name      CHAR(50)    NULL,
    last_name       CHAR(75)    NOT NULL,
    dateofbirth     DATE        NULL
);
```

Task #2 – Allow querying data

- **Query languages** are used to request specific data from databases
  - Sometimes called a Data-Manipulation Language (DML)
- Examples:
  - **SQL** – very common; especially for relational databases
  - **XQuery** – used for XML repositories

```
SELECT college, region, seed FROM tournament
  ORDER BY region, seed;
```

Task #3 – Large amounts of data

- A modern DBMS must handle an enormous amount of data efficiently
- It must be **scalable**
- It must be reliable and offer **persistence**
  - (i.e. keep the data around for a LONG time)

Task #4 - Durability

- A DBMS must be resilient in the face of:
  - Hardware failure
  - Periods of heavy use
    - Must still be responsive
    - Think – Amazon in December!
  - Attempts to breach security

Task #5 - concurrency

- Provide **concurrent** access with **isolation** and **atomicity**:
  - Ensure isolation among concurrent users
  - Ensure individual DB actions to be atomic

Evolution of the database

The data model

- Every database system has an underlying **model** that determines
  - **Structure** placed on the data
  - **Operations** on the data
  - **Constraints** on the data

- Let's explore the history of these models, specifically focusing on the structure for now…

Flat File

- aka - **table** model
- Entire database consists of a single table
  - spreadsheet
- Often stored in a standard text file, CSV or tab-delimited
- Example...

- Simple user database for a system admin

UserName	Password	LastName	FirstName
Mingda	*******	Pan	Mingda
Kim	*******	Abercrombie	Kim
Junmin	*******	Hao	Junmin
David	*******	Pelton	David
Greg	*******	Winston	Greg
Frank	*******	Lee	Frank
Steve	*******	Wilson	Steve

Flat file

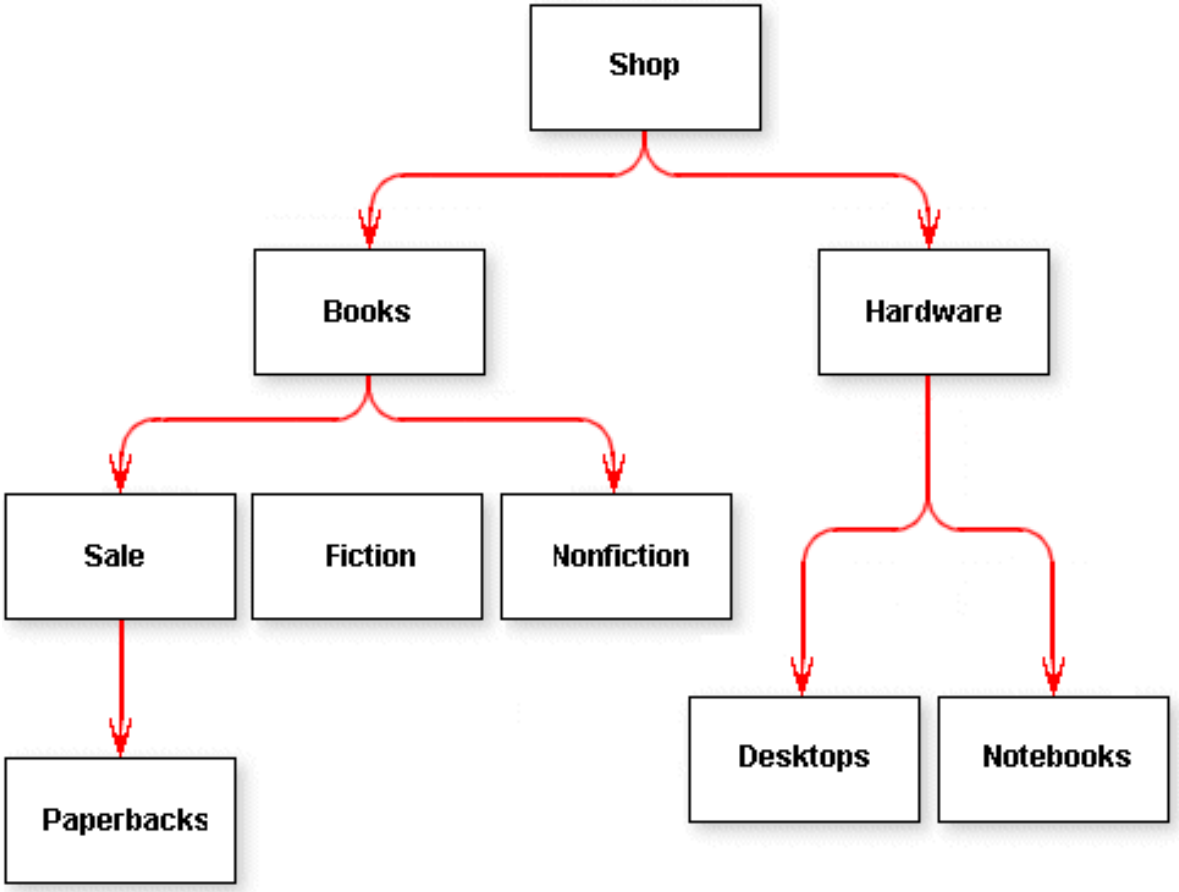
- Discuss advantages / disadvantages
  - HINT – consider an order management system. How would you have a single table containing everything you need with orders?

- Still occasionally used for ***simple*** designs
  - Even here it is usually discouraged. Why?
  - Always need to consider scalability!

Hierarchical Database Model

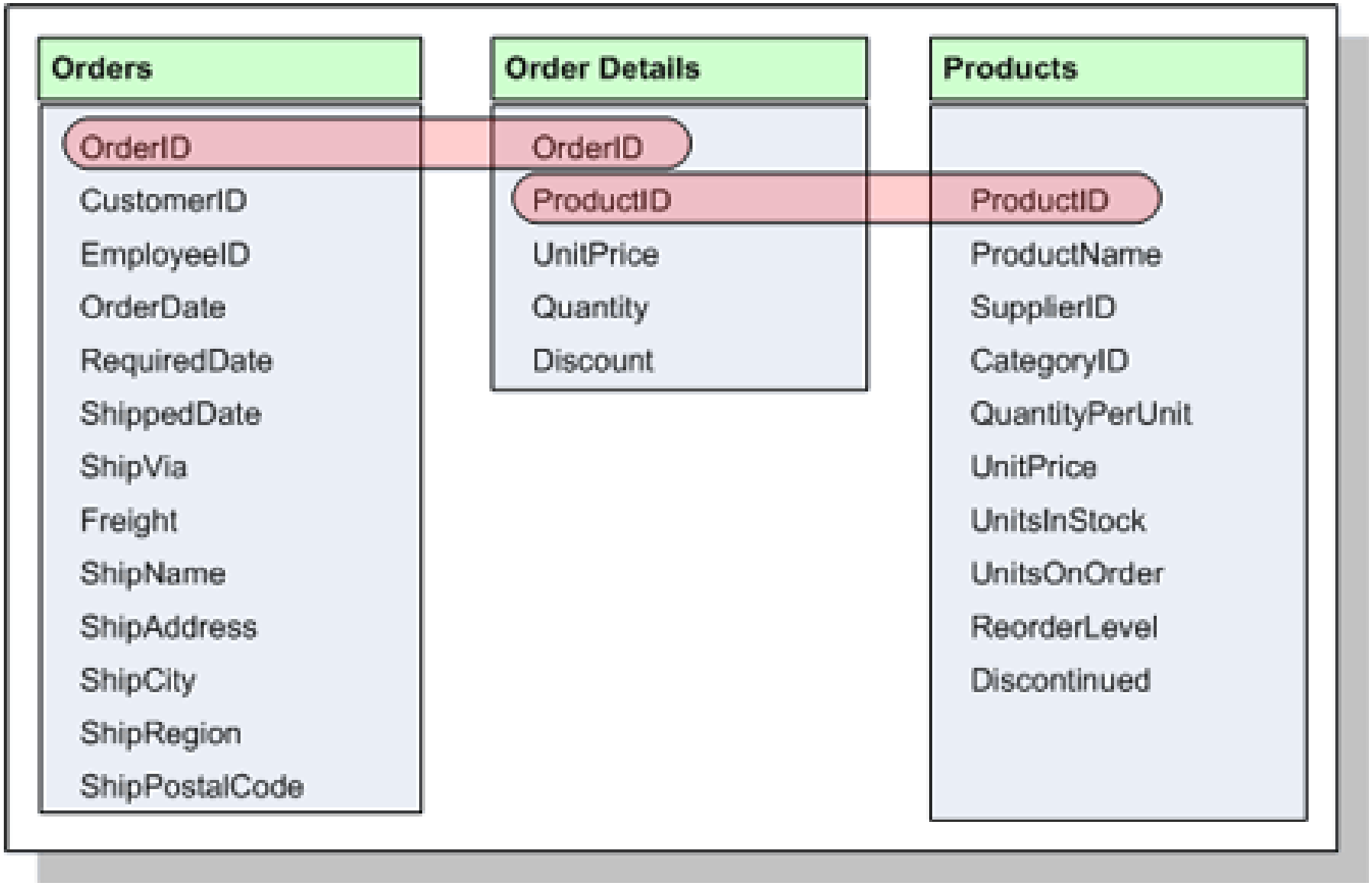
- 1960s – 1970s
- Data is organized into a tree-like structure
- Restriction – **must** maintain tree structure
  - Parents have many children
  - Children have only one parent
- Introduced by IBM with IMS – Information Management System
  - Also used by the Windows Registry!

- Image from oracle.com



Relational Model

- Used by the majority of modern database management systems (DBMS)
- Database = set of **relations** (or tables)
  - Each relation in the database has a distinct name
  - Example: Student, Campus
- Each relation has a set of **attributes**
  - Each attribute has a distinct name, used to describe the **entities** in the relation
  - Example: Students have FirstName, LastName, ID, SS#, etc.
- **Tuple** has a value for each attribute, representing a single **entity**
  - Example: Each student in the class is a tuple
  - An specific entity, Brian
- Each attribute has a **type** (or **domain**)
  - ID:char(9), name:char(25), GPA:float
- One attribute is used to represent something unique about the record
  - Called a **key**
  - Other tables reference this key instead of containing duplication information
    - This is what makes it relational

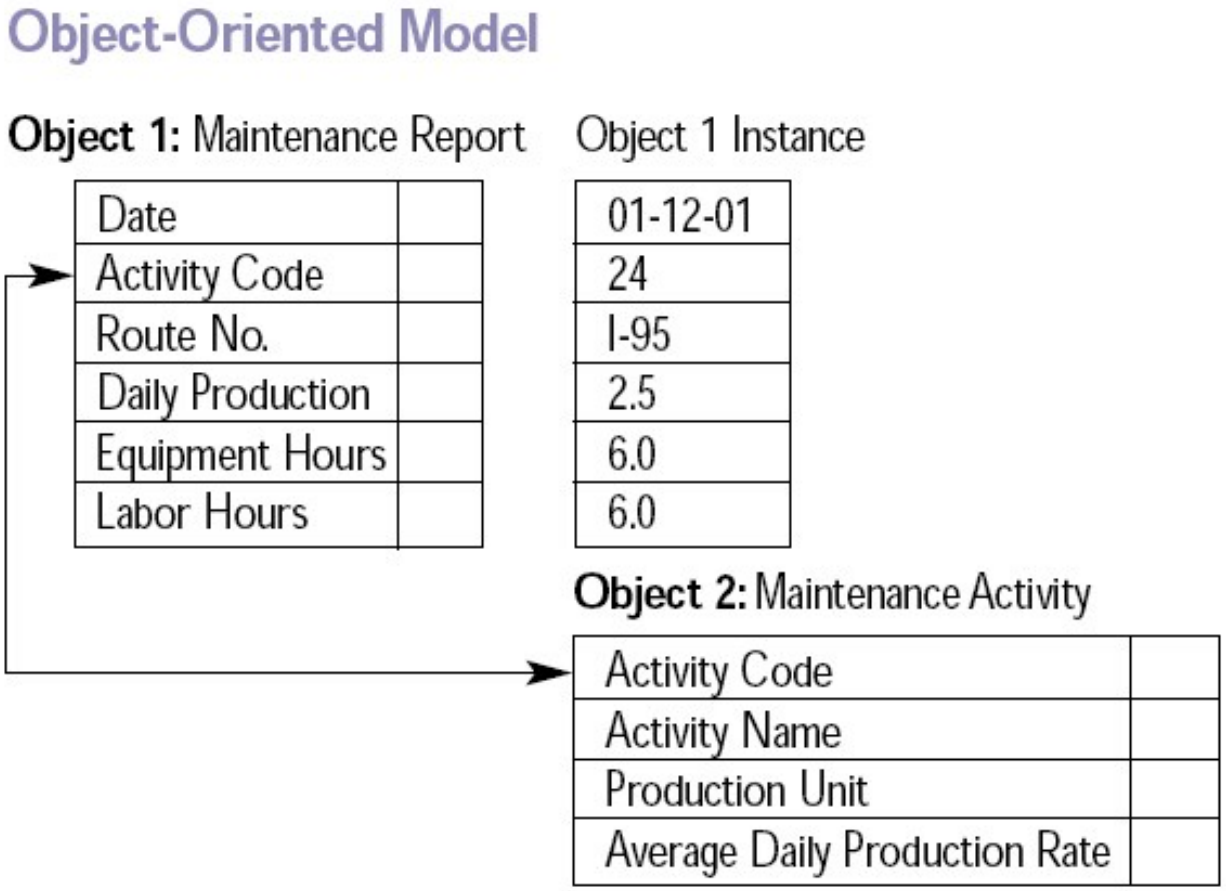

Orders
OrderID
CustomerID
EmployeeID
OrderDate
RequiredDate
ShippedDate
ShipVia
Freight
ShipName
ShipAddress
ShipCity
ShipRegion
ShipPostalCode
Order Details
OrderID
ProductID
UnitPrice
Quantity
Discount
Products
ProductID
ProductName
SupplierID
CategoryID
QuantityPerUnit
UnitPrice
UnitsInStock
UnitsOnOrder
ReorderLevel
Discontinued

- Widespread use today!
- Yet, it has its challenges
  - Type system
  - Implementation of a table
    - Are you going to keep the entire facebook database of users in memory at all times? (Probably not!)
    - Rarely implemented as a 2D table…

Object-oriented model

- You'll also see object-relational model
- A relational model with the restriction that the underlying database system has the same type system as the application program
  - What benefits does this offer the application developer?
  - JDBC

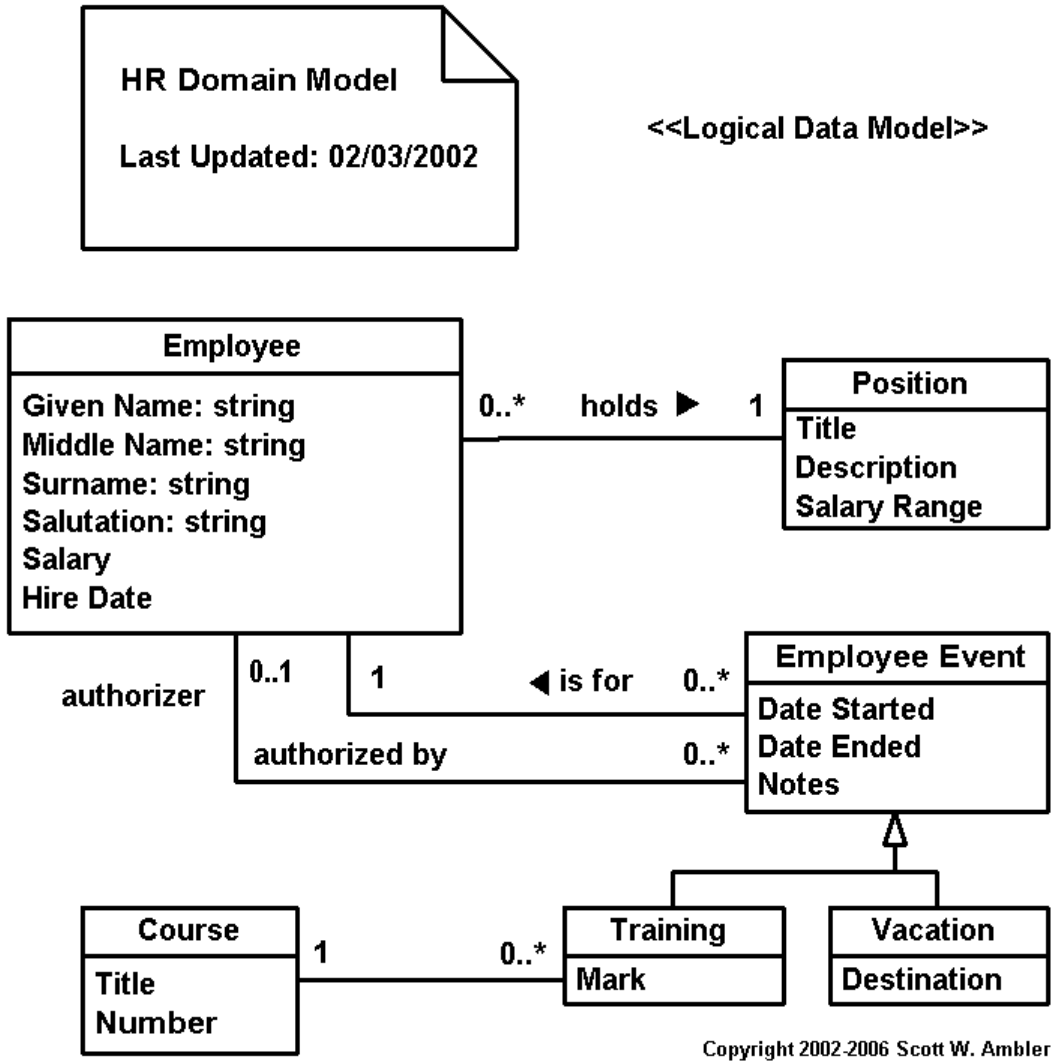
Example


Object-Oriented Model
Object 1: Maintenance Report
Object 1 Instance
Date
Activity Code
Route No.
Daily Production
Equipment Hours
Labor Hours
01-12-01
24
I-95
2.5
6.0
6.0
Object 2: Maintenance Activity
Activity Code
Activity Name
Production Unit
Average Daily Production Rate

Better example

Look familiar?

UML Class diagram!



Semi-structured model

- Data is stored in tree or graph form, rather than tables or arrays
- XML – predominant manifestation of this model
  - Organized as hierarchical nested tags
  - Similar in appearance to HTML

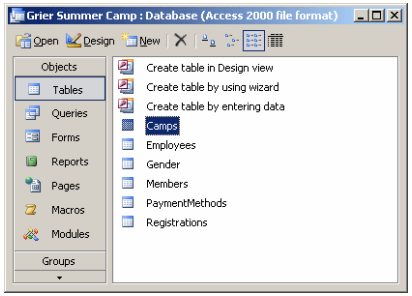
```
<Bookstore>
   <Book ISBN="ISBN-0-13-713526-2" Price="85" Edition="3rd">
      <Title>A First Course in Database Systems</Title>
      <Authors>
         <Author>
            <First_Name>Jeffrey</First_Name>
            <Last_Name>Ullman</Last_Name>
         </Author>
         <Author>
            <First_Name>Jennifer</First_Name>
            <Last_Name>Widom</Last_Name>
         </Author>
      </Authors>
   </Book>
   <Book ISBN="ISBN-0-13-815504-6" Price="100">
      <Title>Database Systems: The Complete Book</Title>
      <Authors>
         <Author>
            <First_Name>Hector</First_Name>
            <Last_Name>Garcia-Molina</Last_Name>
         </Author>
         <Author>
            <First_Name>Jeffrey</First_Name>
            <Last_Name>Ullman</Last_Name>
         </Author>
         <Author>
            <First_Name>Jennifer</First_Name>
            <Last_Name>Widom</Last_Name>
         </Author>
      </Authors>
      <Remark>
         Amazon.com says: Buy this book bundled with "A First Course,"
         it's a great deal!
      </Remark>
   </Book>
</Bookstore>
```

Evolution of the DBMS

… another perspective

- Consider any standard database *transaction*. Who/what is involved?

Users that need access to data



The UI

DBMS

Database

Application needing data – contains business logic

The early days

Users that need access to data

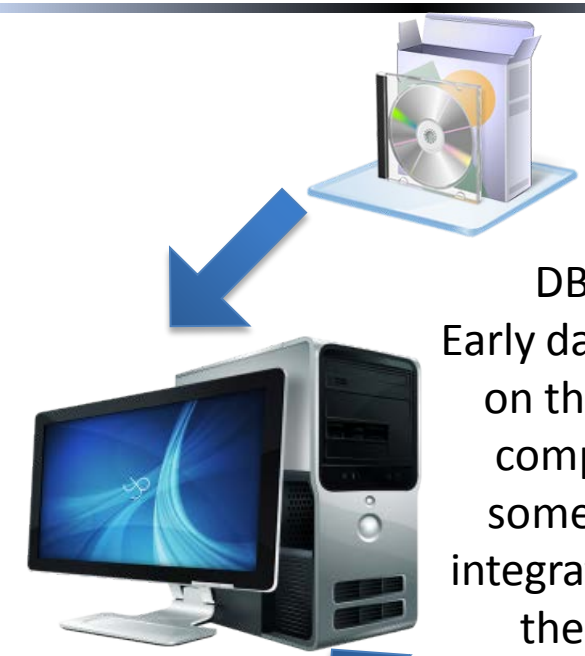
ONE computer that supplied ALL functionality.

Data AND Software that managed data AND interface to data on a local system!

Problems?

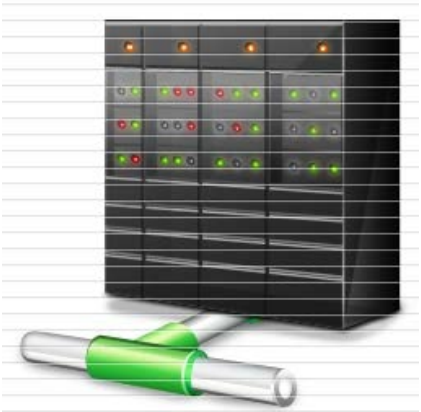
Networked systems - I

Users that need access to data



Data-centric app on computer

DBMS
Early days, lived on the user computer, sometimes integrated into the app

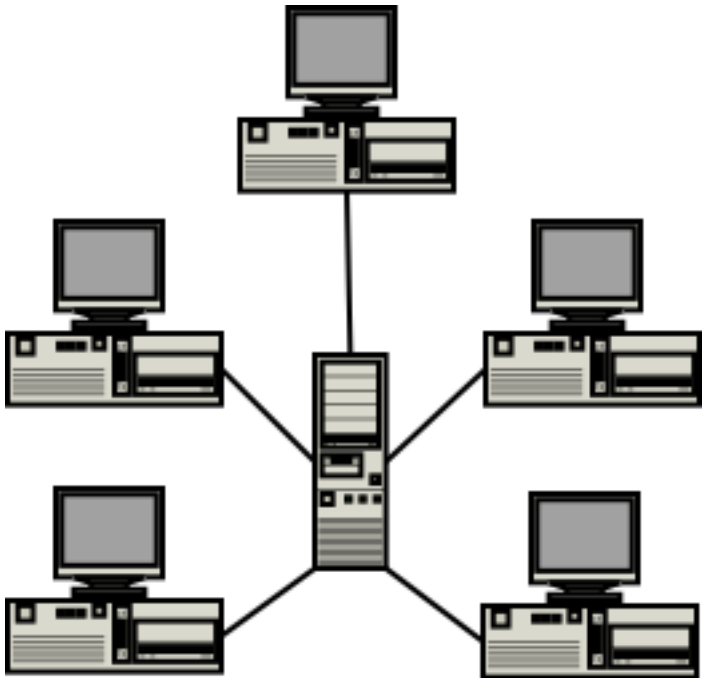


Database

Problems?

Client-Server

- Sometimes referred to as the two-tier model



Server holds database, often applications as well

Clients all run identical applications locally, make requests to DB

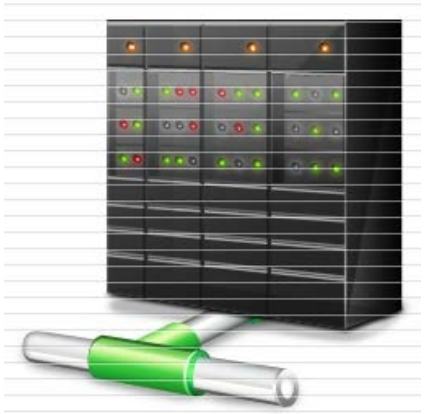
Networked systems - II

Users that need access to data

Application on computer, generates commands to remote DBMS

SQL

DATA

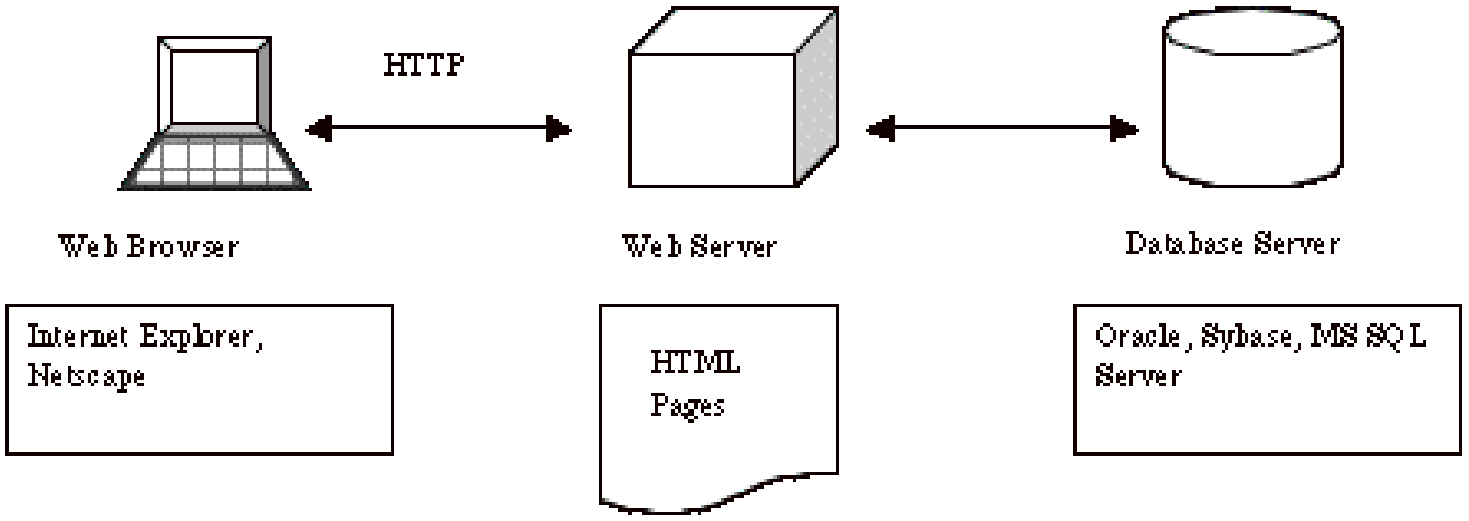


DBMS - Modern designs lives on server

Database

Three-tier System

- Large movement to replace client-server



Networked systems - III

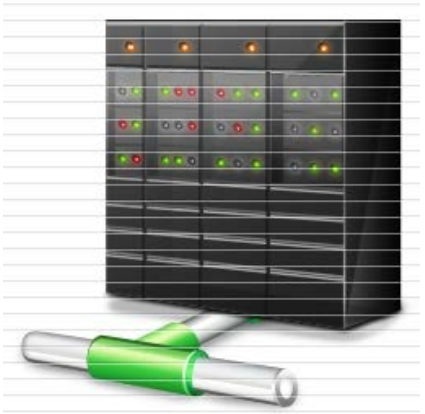
Users that need access to data

Browser – The UI

Web server (e.g. Apache) XHTML, CSS, PHP

Contains business logic

SQL

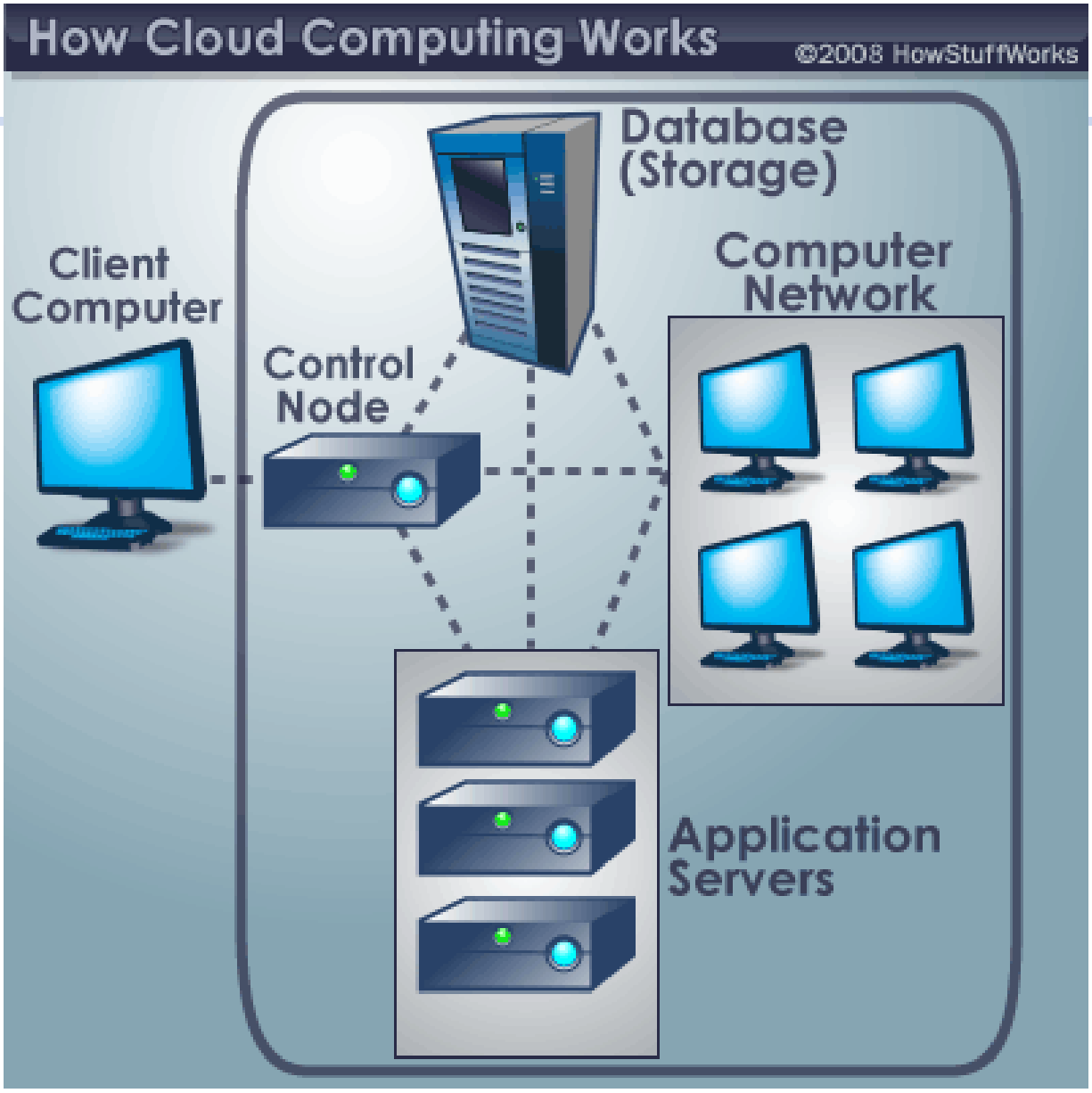


DBMS - Modern designs lives on server

Database

Cloud Computing

- Cloud computing
  - Internet-based computing
  - Computational resources are provided over the Internet, on-demand
    - Applications
    - Data
    - Essentially, the network becomes the classic notion of a computational resource – TIME and SPACE! Both are provided via the Internet
  - Knowledge of physical location of resources is irrelevant
- Benefits?
- Problems?
- Examples:
  - Google Apps
  - Amazon Web Services
  - Microsoft Azure


How Cloud Computing Works
©2008 HowStuffWorks
Database (Storage)
Client Computer
Computer Network
Control Node
Application Servers

References

- http://databases.about.com/od/specificproducts/a/architectures.htm
- http://en.wikipedia.org/wiki/Multitier_architecture
- http://en.wikipedia.org/wiki/Cloud_computing