

**“What info do we need? How does it need to be formatted?”**

**Really, it’s all sort of problematic without knowing what the purpose of the database is.”**

# The Deep End of the Pool

***“Nihil est in intellectu quod non prius in sensu.”***

(Nothing is in the intellect that was not first in the senses.)

Good database design is an art. We are servants to users and Codd both. We must learn to anticipate that which others have never even conceived.

“Don’t just do something... stand there!”

# User Requirements and Specifications

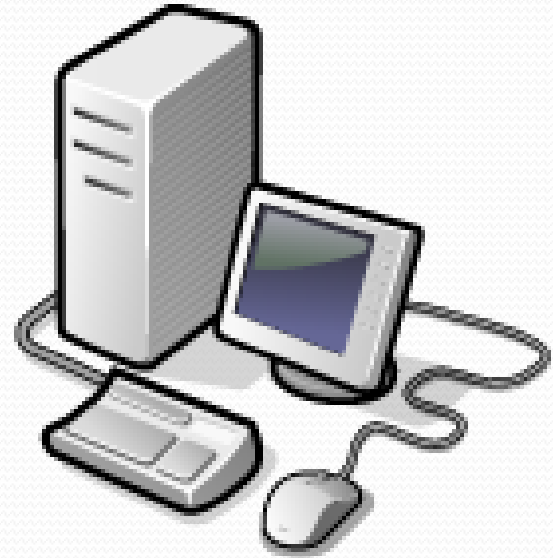
- Who are the users?
  - Who will input data?
  - Who will view data?
  - How will the data be used?
  - Who needs to know what? When? Where?
- The database designer needs to anticipate current and future needs.
- The users themselves oftentimes do not understand the complex issues involved.

# Our Progress So Far...

- **Free form data**
  - Difficult to sort, filter, find
  - Hard to share
  - Hard to establish standards
  - No problem adding as many values in whatever quantity
- **Table in a spreadsheet**
  - Easier to sort, filter, find
  - Hard to share
  - Hard to establish standards
  - Multi-valued items prove to be a pain in the fanny

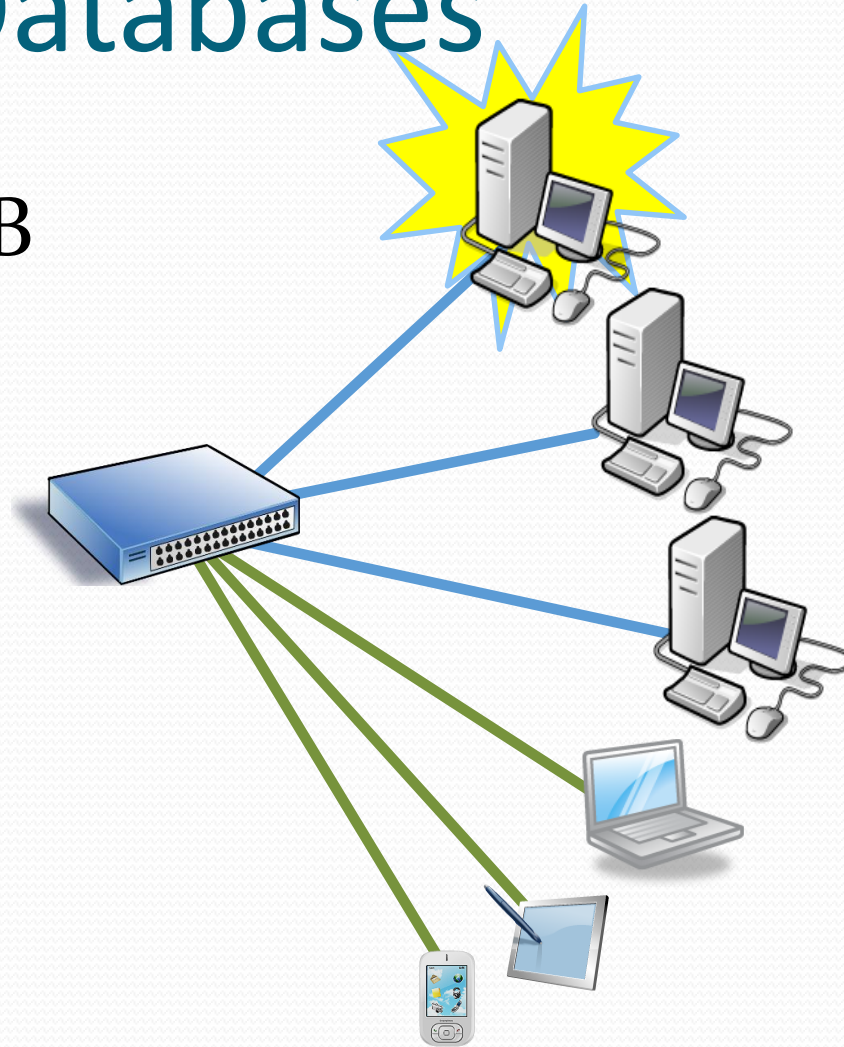
# Brands of Desktop Databases

- Microsoft Access
- dBase
- FileMaker Pro
- FoxPro
- Paradox



# Sharing Desktop Databases

- One desktop hosts the DB
- Clients run the program
- Shares data by copying over network
- Host computer is key
  - No gaming allowed!
- Limited number of users



# Downsides of Desktop DBMS

- Desktop databases - like FoxPro, Access, and Paradox - use the client's processing power, RAM, and bandwidth.
- Make it awkward to share, frequently “lock” records/files.
- Monolithic (all-in-one) infrastructure

User interface

Design

Queries

Reports

Maintenance...

**Database  
Engine**

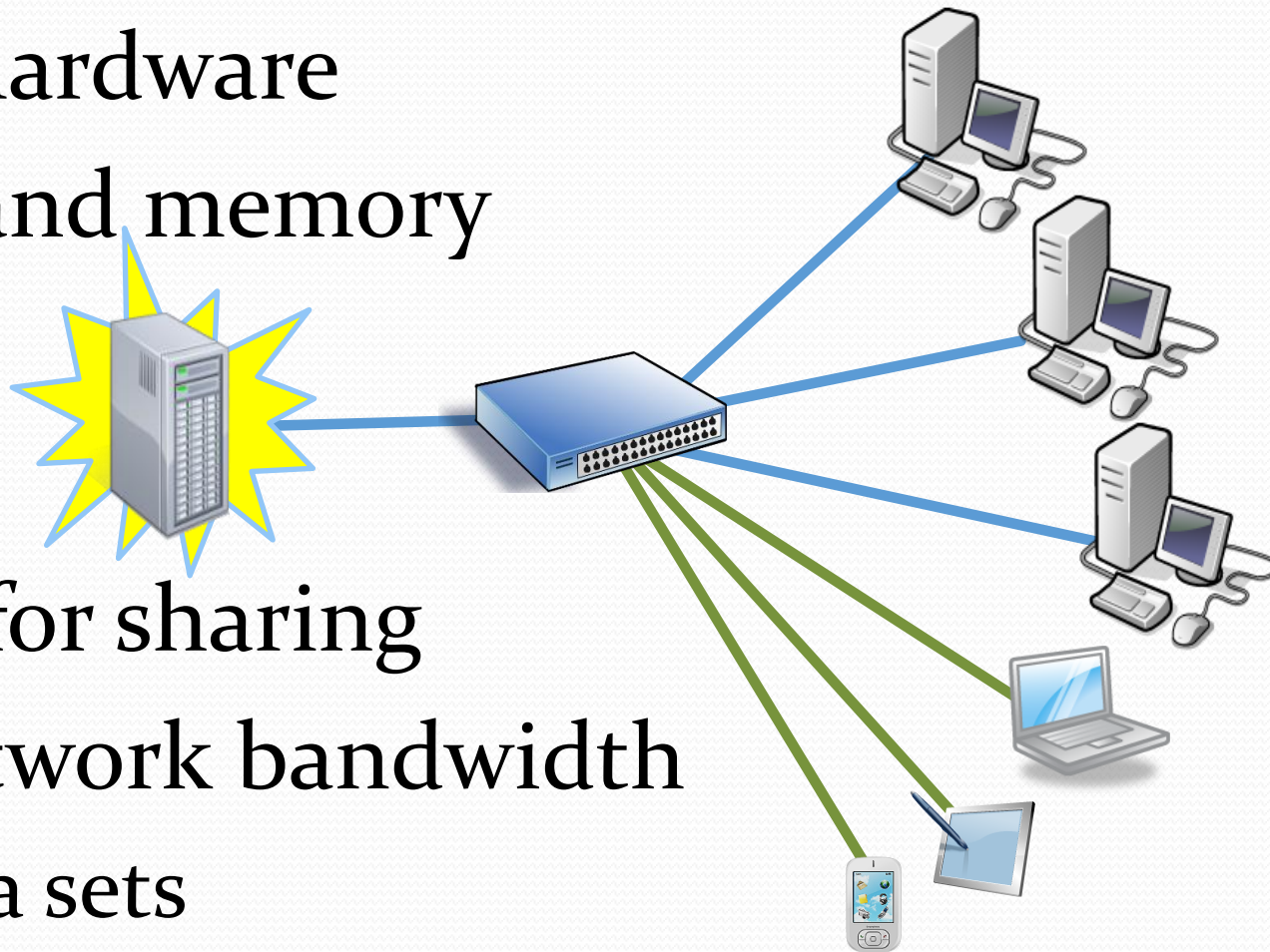
# Enterprise Databases

- INGRES
- DB<sub>2</sub>
- Microsoft SQL
- MySQL / MariaDB
- Oracle
- PL/SQL
- Postgres

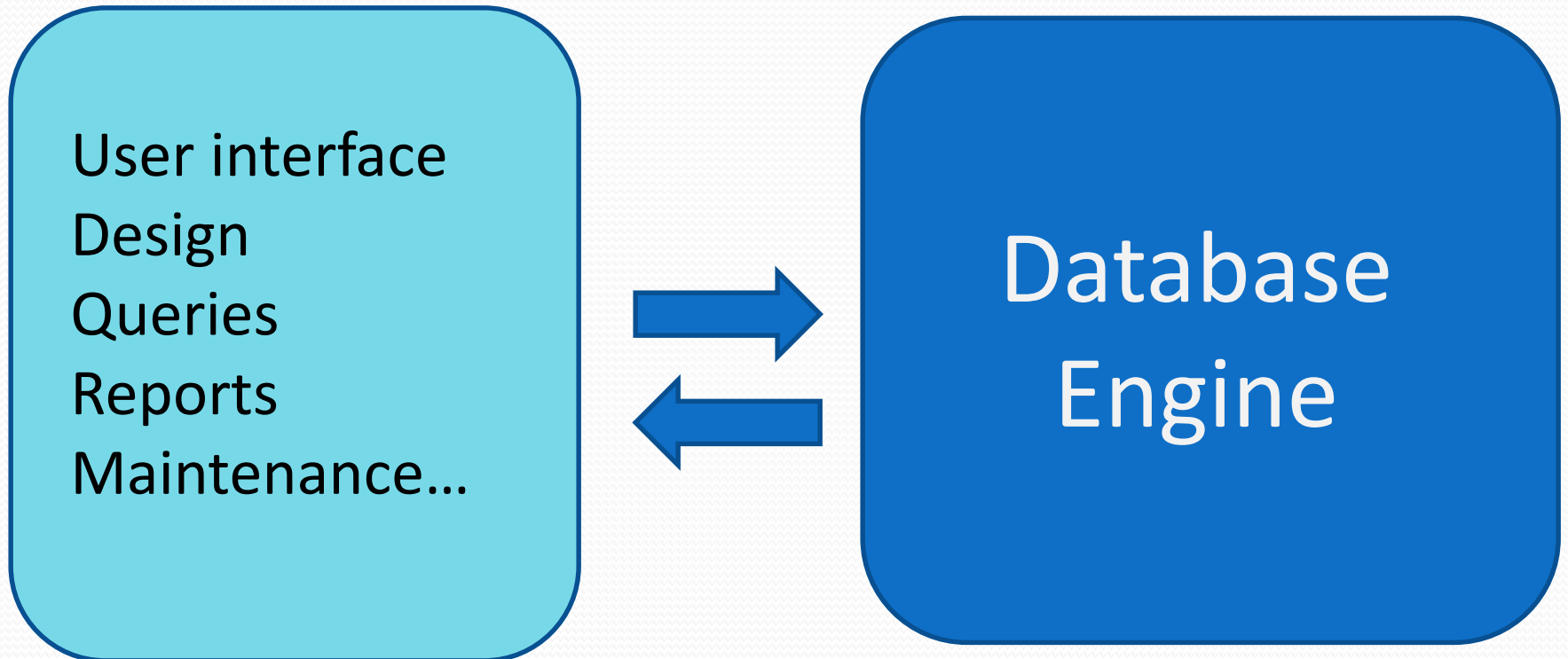


# Enterprise Databases

- Dedicated hardware
- Processors and memory
- Faster
- Secure
- Optimized for sharing
- Reduces network bandwidth
  - smaller data sets

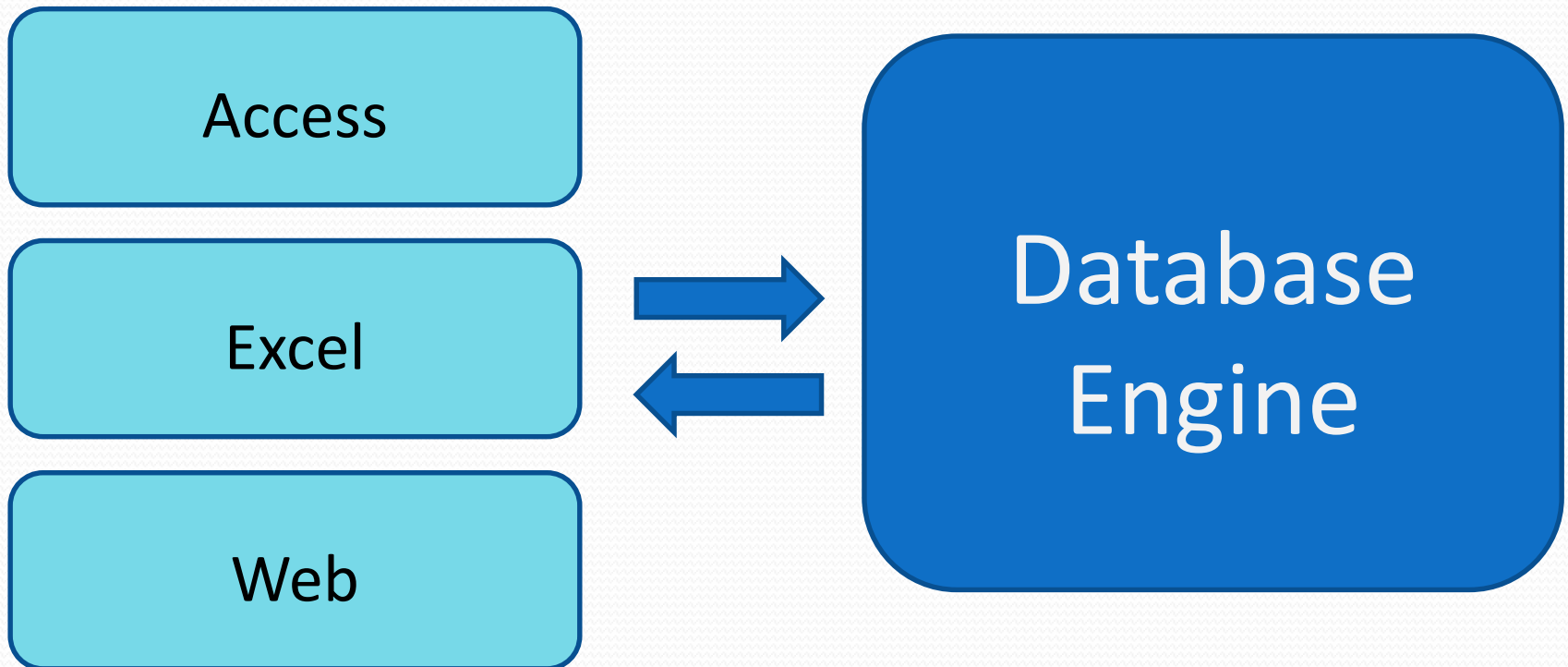


# Server-Based DBMS



# Server-Based DBMS

- A server-based database management system allows for many users (**clients**) to interact simultaneously, increases processing power, and reduces bandwidth.



# Popular Programming Languages

- Active Server Pages (ASP or ASPx)
- Cold Fusion
- PHP
- Java and Javascript
- Application Programmer Interface (API)
- Delphi
- Paradox
- Visual Basic
- Perl

# Why SQL?

- A common query language makes it possible for manifold types of clients to interact simultaneously with the database server.
  - CREATE
  - ALTER
  - INSERT
  - UPDATE
  - DELETE
  - SELECT

# First Command Line Query

```
CREATE TABLE ClassMember1 (FirstName VARCHAR(50), LastName VARCHAR(50));
```

**Good Luck!**