

**Database Systems  
Management Design  
(or just Databases)**

Alex Kuhl

# Fine Lines

- A database at the simplest level is a data storage device
- You could achieve the same storage through spreadsheets, files, or even cleverly created text files
- Where databases get their power is not how things are stored but how they are accessed and interpreted

# Data v. Information

- **Data** is raw facts
  - Another way to think about it is anything that hasn't been processed or interpreted in any way
  - It could mean anything!
- **Information** is data that has been interpreted
  - Data that has been assigned meaning
  - Usually requires some sort of context
- Example: Listing the responses collected from a survey would be giving someone data, giving them the totals and other statistics would be providing them information

# Looking a little closer

If I were to say "average speed is 40" how would you interpret it?

# One more, just to confuse you

- On top of data and information we can also have *knowledge*
- Knowledge is accumulated information about a specific subject
  - Implies familiarity, awareness, and understanding
- The key to knowledge is that you can derive new knowledge from old
  - Think of knowledge like math rules: you can prove new laws using old ones

# Enough already!

- Think of these three in terms of building
  - Data would be the raw materials that someone would need, say stones out of a quarry
  - Information would be the stones cut to appropriate blocks and sizes based on their type, strength, etc.
  - Knowledge would be using your stone slabs and bricks to actually raise your building

# Why should I care?

- To make good decisions we need quality information
  - Accurate, timely, and relevant
- You could use a survey to easily identify the parts of your business that need improvement
- Recognizing trends is the key to staying ahead of competition

# What about databases?

- It is a little fuzzy whether databases store data or information (or both?)
  - In a table of figures do they mean something?
  - Depends on how you read them right? Are there headings?

# Storage Concerns

- There are many storage concerns
  - Formatting
  - Security
  - Physical
- **Data management** is the discipline of acquiring, storing, and accessing data (and information)

# Physical Storage

- Old school: card catalogs and records
  - Can be logical or systematic arrangement
  - How do we really aggregate or report anything?
- Not quite as old: computer files (spreadsheets for example)
  - Data processing specialist (?)
- Databases have their roots in file systems
  - Book claims managing data through a file system is obsolete... is this true?

# DB Data Details

- Databases store two types of information
  - Data is stored in the "cells" of the tables
  - Metadata is also stored on a table-by-table basis
- Metadata is data/information about the data in the table (wrap your head around that!)
  - Concretely, metadata can be headers/names, data types, relationships, etc. that aid us in turning the raw data into information

# Computer File Example

**FIGURE 1.3** Contents of the CUSTOMER file

C_NAME	C_PHONE	C_ADDRESS	C_ZIP	A_NAME	A_PHONE	TP	AMT	REN
Alfred A. Ramas	615-844-2573	218 Fork Rd., Babs, TN	36123	Leah F. Hahn	615-882-1244	T1	100.00	05-Apr-2008
Leona K. Dunne	713-894-1238	Box 12A, Fox, KY	25246	Alex B. Alby	713-228-1249	T1	250.00	16-Jun-2008
Kathy W. Smith	615-894-2285	125 Oak Ln, Babs, TN	36123	Leah F. Hahn	615-882-2144	S2	150.00	29-Jan-2009
Paul F. Olowski	615-894-2180	217 Lee Ln., Babs, TN	36123	Leah F. Hahn	615-882-1244	S1	300.00	14-Oct-2008
Myron Orlando	615-222-1672	Box 111, New, TN	36155	Alex B. Alby	713-228-1249	T1	100.00	28-Dec-2008
Amy B. O'Brian	713-442-3381	387 Troll Dr., Fox, KY	25246	John T. Okon	615-123-5589	T2	850.00	22-Sep-2008
James G. Brown	615-297-1228	21 Tye Rd., Nash, TN	37118	Leah F. Hahn	615-882-1244	S1	120.00	25-Mar-2009
George Williams	615-290-2556	155 Maple, Nash, TN	37119	John T. Okon	615-123-5589	S1	250.00	17-Jul-2008
Anne G. Farriss	713-382-7185	2119 Elm, Crew, KY	25432	Alex B. Alby	713-228-1249	T2	100.00	03-Dec-2008
Olette K. Smith	615-297-3809	2782 Main, Nash, TN	37118	John T. Okon	615-123-5589	S2	500.00	14-Mar-2009

C\_NAME = Customer name  
C\_PHONE = Customer phone  
C\_ADDRESS = Customer address  
C\_ZIP = Customer zip code

A\_NAME = Agent name  
A\_PHONE = Agent phone  
TP = Insurance type  
AMT = Insurance policy amount, in thousands of \$  
REN = Insurance renewal date

- Terms: Data, field, record, file

# Uh oh

- Writing a program to do reporting and such on one file isn't bad
- What happens when you need more than one spreadsheet?
  - Say a file for orders, a file for inventory, and a file for customers
  - These could be managed by separate departments
  - Evolution of the data processing manager

# The Trouble with Files

- Programming can become nightmare
  - "ad hoc" information nearly impossible
  - What about if a slight structural change is required to a file?
- Maintenance quickly becomes unwieldy
- Security and confidentiality (maybe)

# Dependency Problems

- Structural (in)dependence
  - Back to the "add a field" example from before
- Data (in)dependence
  - Does changing the data type of a field matter?
  - How does storage change affect access?
  - Logical v. Physical format

# I think therefore I am

The combination of data and metadata sometimes leads people to refer to databases as "a collection of ***self-describing*** data." What do you think?

# Database Management Systems

- A DBMS is the middleman between the data in the DB itself and everything else
  - A database is simply a file (or files) on a hard drive
  - The DBMS gives meaning to the contents
- Provides a way for a database administrator (DBA) to do typical system functions
- It is a way of abstracting the details of data retrieval and aggregation from end-users and programmers

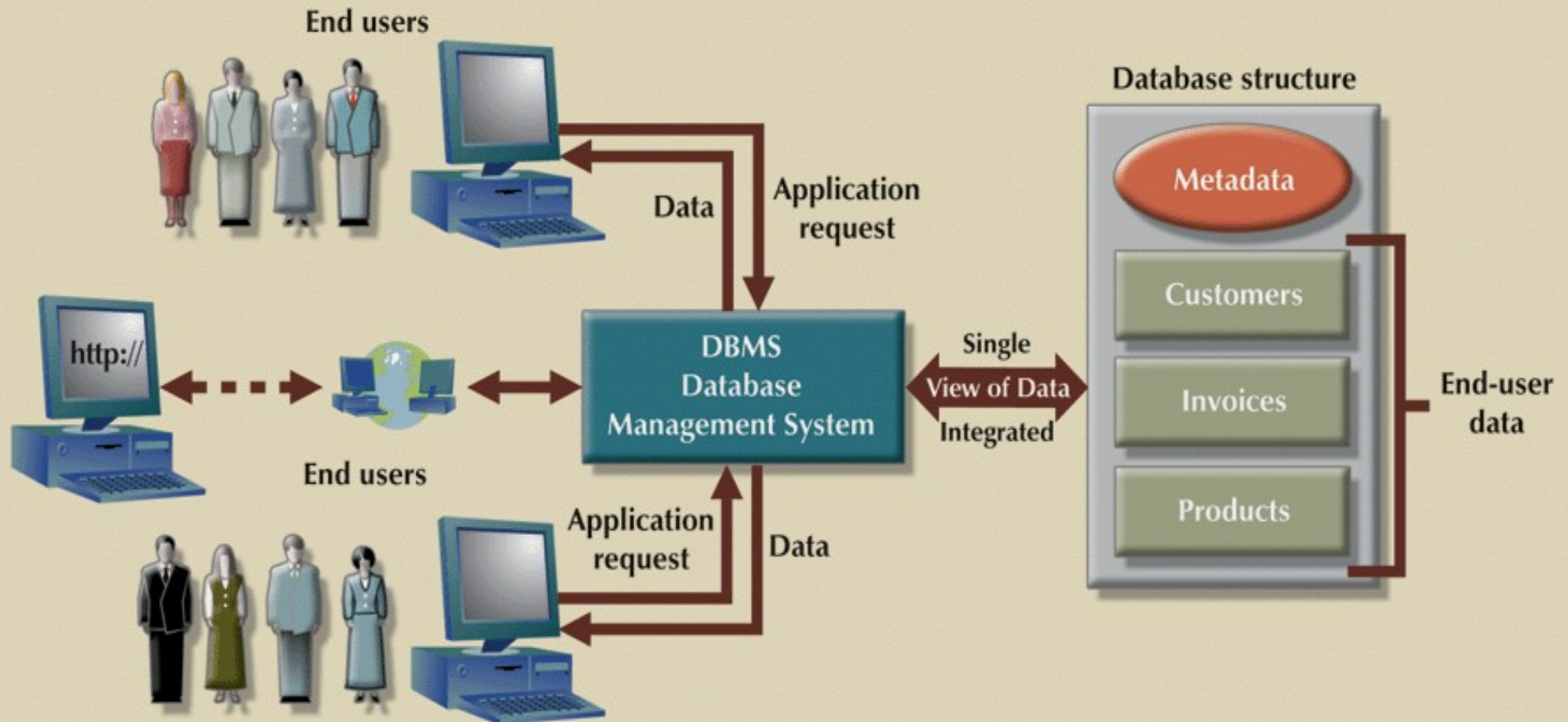
# More Benefits

- A DBMS provides a layer of security for concurrency features
  - Access/sharing
  - Security
  - Integrity
- Allows for queries (in SQL) to be run against the DB data
- Decision making and end-user productivity

# The Visual

FIGURE 1.2

The DBMS manages the interaction between the end user and the database



# Types of Databases

- Single-user or desktop
- Multiuser – workgroup or enterprise
- Centralized or Distributed
- Operational/transactional/production
- Data warehouses
- In reality, these terms probably aren't valid or are a little grayer than the book makes it out to be
- A few types: Access, MS SQL, MySQL, Oracle

# Design Preview

**FIGURE 1.3** Contents of the CUSTOMER file

C_NAME	C_PHONE	C_ADDRESS	C_ZIP	A_NAME	A_PHONE	TP	AMT	REN
Alfred A. Ramas	615-844-2573	218 Fork Rd., Babs, TN	36123	Leah F. Hahn	615-882-1244	T1	100.00	05-Apr-2008
Leona K. Dunne	713-894-1238	Box 12A, Fox, KY	25246	Alex B. Alby	713-228-1249	T1	250.00	16-Jun-2008
Kathy W. Smith	615-894-2285	125 Oak Ln, Babs, TN	36123	Leah F. Hahn	615-882-2144	S2	150.00	29-Jan-2009
Paul F. Olowski	615-894-2180	217 Lee Ln., Babs, TN	36123	Leah F. Hahn	615-882-1244	S1	300.00	14-Oct-2008
Myron Orlando	615-222-1672	Box 111, New, TN	36155	Alex B. Alby	713-228-1249	T1	100.00	28-Dec-2008
Amy B. O'Brian	713-442-3381	387 Troll Dr., Fox, KY	25246	John T. Okon	615-123-5589	T2	850.00	22-Sep-2008
James G. Brown	615-297-1228	21 Tye Rd., Nash, TN	37118	Leah F. Hahn	615-882-1244	S1	120.00	25-Mar-2009
George Williams	615-290-2556	155 Maple, Nash, TN	37119	John T. Okon	615-123-5589	S1	250.00	17-Jul-2008
Anne G. Farriss	713-382-7185	2119 Elm, Crew, KY	25432	Alex B. Alby	713-228-1249	T2	100.00	03-Dec-2008
Olette K. Smith	615-297-3809	2782 Main, Nash, TN	37118	John T. Okon	615-123-5589	S2	500.00	14-Mar-2009

**C\_NAME** = Customer name  
**C\_PHONE** = Customer phone  
**C\_ADDRESS** = Customer address  
**C\_ZIP** = Customer zip code

**A\_NAME** = Agent name  
**A\_PHONE** = Agent phone  
**TP** = Insurance type  
**AMT** = Insurance policy amount, in thousands of \$  
**REN** = Insurance renewal date

# Things to Think About

- Descriptive names
  - But not *too* descriptive
- Appropriate data separation
- Unique identifiers (primary key)
- "Islands of information" (?)

# Redundancy

- Good redundancy → backups!
- Bad redundancy
  - Inconsistency – data between tables doesn't match  
→ Lack of *integrity* = accuracy and verifiability
  - Anomalies – duplicated data makes it easy for updates to not be made fully  
→ can be update, insertion, or deletion types

# Database System

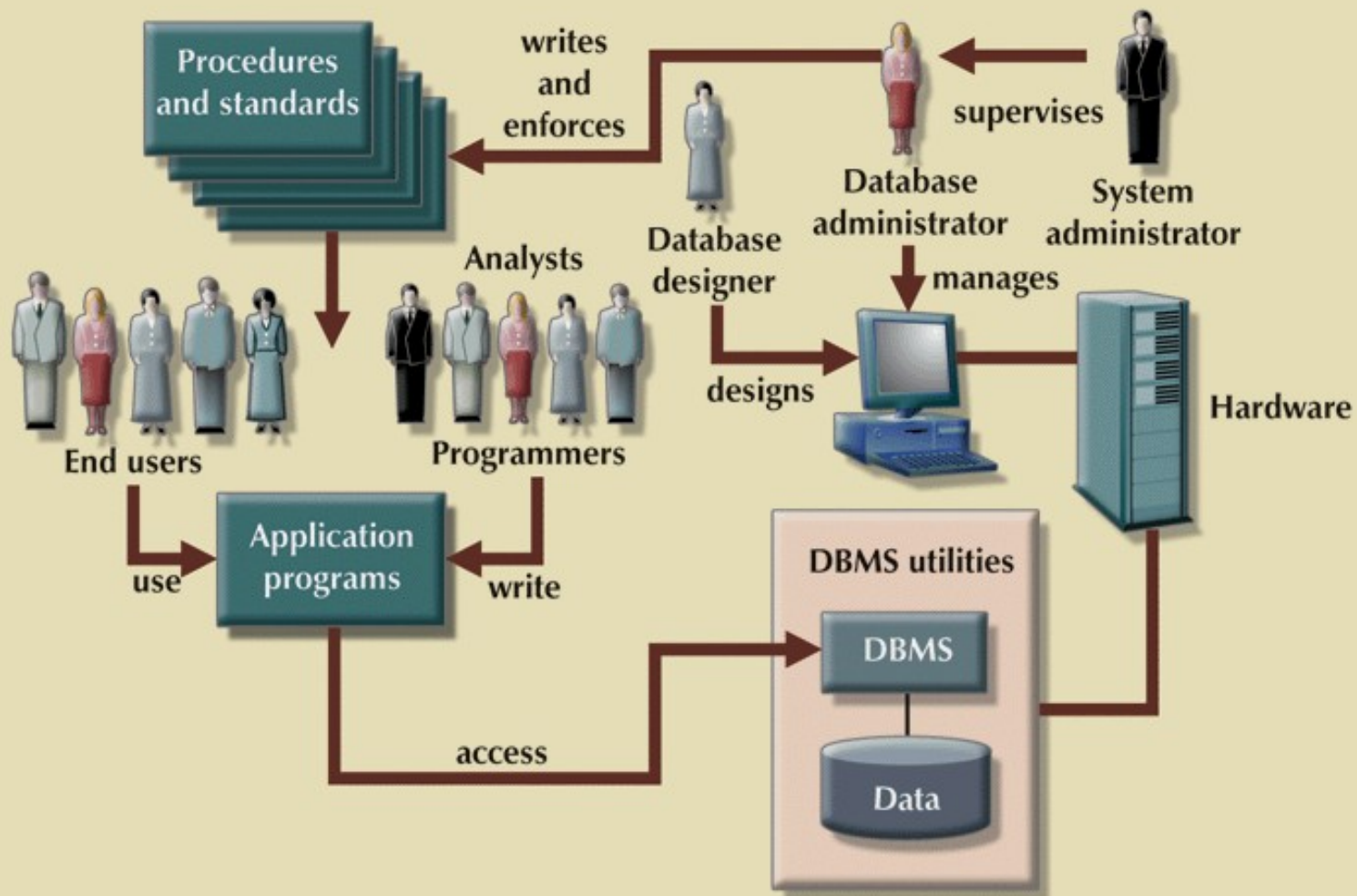
- Basically, sensibly related data stored in a single *logical* location
- Manages storage, structure, relationships, and access
- The DBMS is able to manage the system and alleviate or eliminate the problems with file systems
- DBMS is only part of the Database System though...

# Database System Definition

- An organization of components that define and regulate the collection, storage, management, and use of data within a database environment
- It is composed of five major components:
  - Hardware
  - Software
  - People
  - Procedures
  - Data

# The Scope

FIGURE 1.7 The database system environment



# The Components

- Hardware – the physical structure
- Software – making everything run
  - OS
  - DMBS
  - Other applications, tools, and utilities
- People
  - SA and DBA
  - DB designers
  - Programmers
  - Users

# The Components

- Procedures – how to design and use the database system
- Data – you should know this one by now

# A new level

- Levels of complexity for a DBS vary depending on its use
  - Does State Farm have the same needs of Amazon?  
What about the local library?
  - Human factors
  - Cost factors
  - Existing technologies

# Functions of a DBMS

- Quick review of what a DBMS does for us:
  - Data management and storage
  - Access, calculations, and presentation
  - Security
  - Concurrency and Integrity
  - Backup and Recovery
  - SQL, API, and other communication interfaces
- The goal is to provide these services as transparently as possible
  - "Just let me get my work done!"

# Data Management and Storage

- Keep track of data and metadata in the *data dictionary*
- Provide a level of abstraction for both structure and storage
- (Automatic) Performance Tuning
- Also stores secondary information such as pre-defined reports and queries

# Access, Calculation, Presentation

- Similar to the last point, abstracts the access of data
- Separates logical representation from physical
  - ex. dates have to be stored consistently but presented based on who is viewing it
- Formatting upon entering data
- Calculating data before presenting to the user
- For the types of operations that a DB is used for it is much faster than working with files and procedural languages

# Security

- Role-based access control – users have read/write permissions based on their roles in the organization
- Authentication

# Concurrency and Integrity

- Allows multiple users to be accessing the same data at the same time without them having to worry about dirty (or stale) data read/writes
- Minimize or eliminate data redundancy
- This ensures that data integrity is always maintained

# Concurrency and Integrity

- Remember the ACID test
  - **A**tomicity – each transaction must either complete or roll back
  - **C**onsistency – the DB can only move from one valid state to another, if a transaction would result in an invalid state is is not allowed or rolled back
  - **I**solation – transaction results are invisible to the rest of the system until they are complete; you can see the before or after state but not an intermediate
  - **D**urability – once a transaction is committed to the DB it becomes (semi)permanent

# Backup and Recovery

- Backup is something that should *always* be done at *regular* intervals
- The DMBS can help to automate the process
- With these you can recover data due to failures, disasters, or dumb programmers (guilty as charged!)
- It is also a good idea to **sandbox** development to separate it from a production DB

# SQL, APIs, and Communications

- SQL – a nonprocedural (or *declarative*) way of accessing data
  - Instead of step-by-step instructions you say "I need data that meet these criteria," much easier for non-programmers and eliminates the need to know about the data and its structure
- API – make DB functionality available to other languages
- DBMSs can also report directly, through systems like web pages, e-mail (or other alerts), and reports

# Preview: DB Management

- With a DBS handling a lot of the prior grunt work, focus shifts on managing the system and its data
- Problems with a DBS
  - Higher cost (maybe)
  - Management Complexity
  - Vendor Dependence
  - Maintaining Currency (updates and training)
  - Frequent upgrade/replacement cycles (ha!)