Database Management: Introduction

Luis Aguilar
University of California, Berkeley
School of Information

Info 257: Database Management

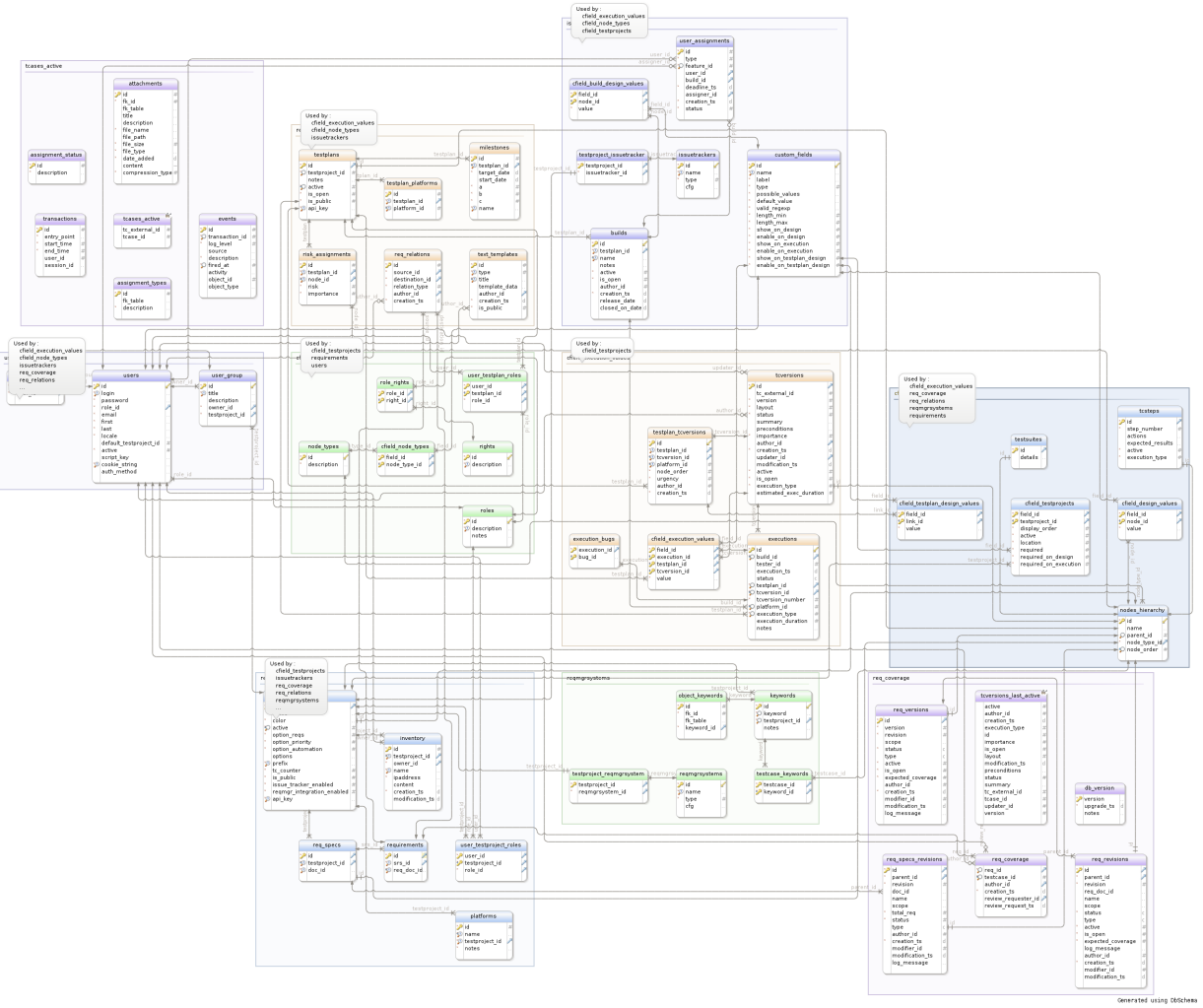
Overview

- Introduction
- Announcements
- Course Description
- Database Concepts and Terminology
- Database Models
- Database Life Cycle

Introduction

- “Lewis” or “Luis” is fine
- B.S. Computer Science from CSUS, MIMS Class of 2014
- I took INFO 257 in 2012!
- Eight years in industry as a Technical Consultant, developing custom databases, integrations and application from various data sources to Software as a Service (SaaS) providers.
- Currently a DevOps Engineer with Grove Collaborative, helping internal teams interact with our Postgres RDBMS

Goal



TA Introduction

- Eric Zan
  - eric.zan@ischool.berkeley.edu

Overview

- Announcements
- Course Description
- Database Concepts and Terminology
- Database Models
- Database Life Cycle

Course Overview

- Description of the Course
- Assignments and Labs
- Readings
- Grading
- Schedule
- Web site:
  http://courses.ischool.berkeley.edu/i257/s20

Course Description

- This course is concerned with the design of the database itself -- not with the design of database system software.
  - We will discuss DBMS internals only as they relate to the database and its design and structure
- We will spend a fair amount time on database application design, especially on options for Web application database support -- but this will not be primary focus.

Course Description

- A schedule has been created, but class is subject to change depending on class skillset.
- Not looking to duplicate the new Back-end Web Architecture class.

Assignments

- Three kinds of assignments
  - Using a pre-built database for search and retrieval and database modification queries
  - Labs where you will interact with a tool or application
  - Designing, populating, and running queries against your ***own personal (or group)*** database
    - Types of database project
      - Group (3-4)
        - » Work or Course related
        - » Final Masters' project
        - » Projects from around campus that need doing…

Periodic Labs

- Docker
  - Creating Dockerfile Images
  - DB installation
- Github
- Other Topics
  - Suggestions?

Readings

- Textbook is:
  - Jeffrey A. Hoffer, Ramesh Venkataraman and Heikki Topi. ***Modern Database Management (13th Edition).*** Pearson : , 2019.
    - ISBN 978-0-13-477365-0 or 0-13-477365-9
  - 12th, 11th editions are OK too
  - Please note, the older editions may not have up to date sections around web applications, big data, NoSQL, etc.

Grading

- Grades will be based on:
  - Assignments and Labs(50%)
  - Group Database project (50%)
  - (No midterm or final)
  - Quizzes?

Schedule

- Walkthrough of website:
  - http://courses.ischool.berkeley.edu/i257/s20

Overview

- Announcements
- Course Description
- Database Concepts and Terminology
- Database Models
- Database Life Cycle

Overview

- What's a database?
- What's a DataBase Management System?
- Why use a database & DBMS?
- How can you use a DBMS?

What is a Database?

How many have used a database today?

What is a database?

- Wikipedia tells us:
  - "A database is an organized collection of data. The data is typically organized to model relevant aspects of reality (for example, in a salesperson's database, the data would include facts such as customer name, address and telephone number."

Other Definitions

- **Data:** stored representations of meaningful objects and events
  - Structured: numbers, text, dates
  - Unstructured: images, video, documents
- **Information:** data processed to increase knowledge in the person using the data
- **Metadata:** data that describes the properties and context of user data

Figure 1-1 Converting Data to Information
(1 of 2)

(a) Data in context

Class Roster

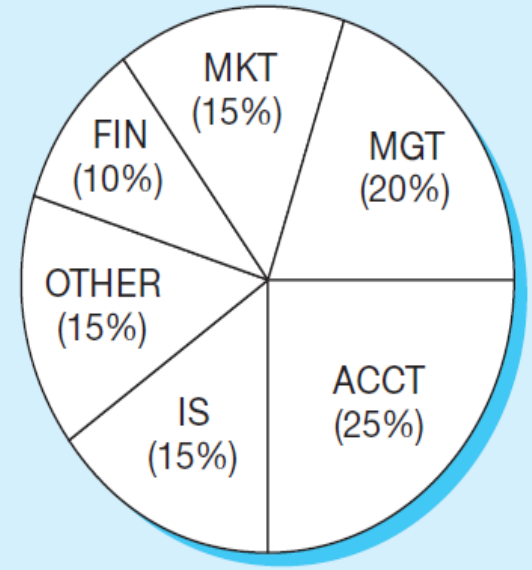
Course: MGT 500 Business Policy Semester: Spring 2018

Section: 2

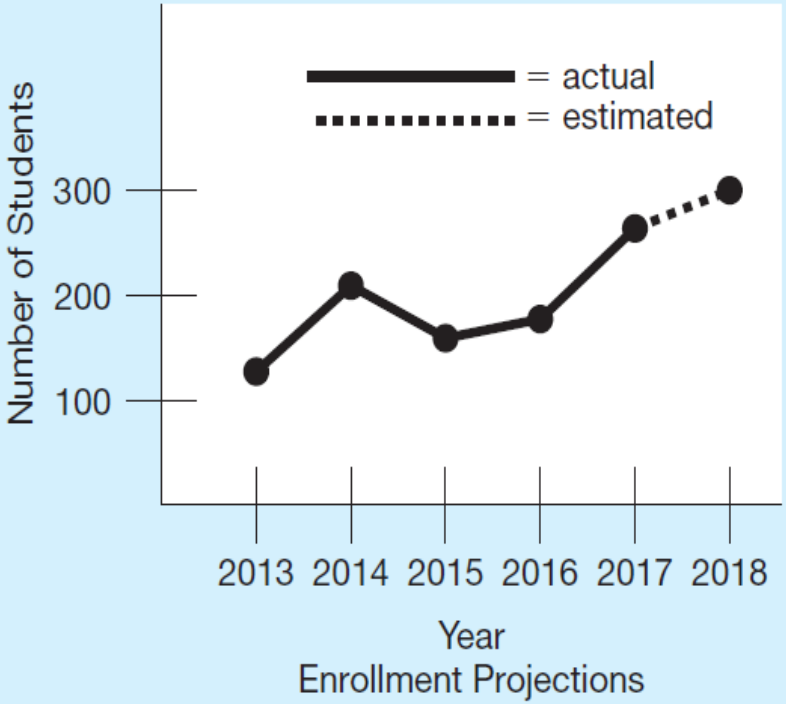
Name	ID	Major	GPA
Baker, Kenneth D.	324917628	MGT	2.9
Doyle, Joan E.	476193248	MKT	3.4
Finkle, Clive R.	548429344	PRM	2.8
Lewis, John C.	551742186	MGT	3.7
McFerran, Debra R.	409723145	IS	2.9
Sisneros, Michael	392416582	ACCT	3.3

Figure 1-1 Converting Data to Information
(2 of 2)

(b) Summarized data



Percent Enrollment by Major (2018)



Enrollment Projections

What is a Database?

- *A Database is a collection of stored operational data used by the application systems of some particular enterprise*. (C.J. Date)
  - Paper "Databases"
    - Still contain a large portion of the world's knowledge
      - Changing as, for example, book scanning projects like Google Books and the Open Content Alliance convert paper docs
  - File-Based Data Processing Systems
    - Early batch processing of (primarily) business data
      - Still with us – in fact the entire Hadoop MapReduce suite used in Big Data processing is primarily file-based
  - Database Management Systems (DBMS)
    - Some old ones still in use, but many modern DBMS are relational, object or object-relational, and extensive use of so-called "noSQL" key/object databases

What is a DataBase Management System?

- Wikipedia again:
  - "A general-purpose database management system (DBMS) is a software system designed to allow the definition, creation, querying, update, and administration of databases.
  - Well-known DBMSs include MySQL, PostgreSQL, SQLite, Microsoft SQL Server, Microsoft Access, Oracle, Sybase, dBASE, FoxPro, and IBM DB2."

What's a DBMS

- It maintains **Metadata** about the database
  - Data about data
    - In DBMS this means all of the characteristics describing the attributes of an entity, E.G.:
      - name of attributes
      - data type of attributes
      - size of the attributes
      - format or special characteristics
  - Characteristics of tables or 'relations'
    - Name, content, notes, etc.
    - Associated elements in other tables

Table 1-1 Example Metadata for Class Roster

- Descriptions of the properties or characteristics of the data, including data types, field sizes, allowable values, and data context

Name	Type	Length	Min	Max	Description	Source
Course	Alphanumeric	30			Course ID and name	Academic Unit
Section	Integer	1	1	9	Section number	Registrar
Semester	Alphanumeric	10			Semester and year	Registrar
Name	Alphanumeric	30			Student name	Student IS
ID	Integer	9			Student ID (SSN)	Student IS
Major	Alphanumeric	4			Student major	Student IS
GPA	Decimal	3	0.0	4.0	Student grade point average	Academic Unit

Why Databases and DBMS

- In programming courses, you have learned about data and file structures and how they can be used in your programs to help you accomplish various goals
- Let's say you want to create a program to keep a list of names and addresses
  - How would you write a program to do it?
  - Suppose the list got REALLY big – what kind of file structures might you use in searching it?

Why Use a DBMS?

- History
  - 50's and 60's all applications were custom built for particular needs
  - File based
  - Many similar/duplicative applications dealing with collections of business data
  - Early DBMS were extensions of programming languages
  - 1970 - E.F. Codd and the Relational Model
  - 1979 - Ashton-Tate & first Microcomputer DBMS

From File Systems to DBMS

- Problems with File Processing systems
  - Inconsistent Data (Duplication)
  - Inflexibility
  - Limited Data Sharing
  - Lengthy Development Times
  - Poor enforcement of standards
  - Excessive program maintenance

DBMS Benefits

- Minimal Data Redundancy
- Consistency of Data
- Integration of Data
- Sharing of Data
- Ease of Application Development
- Uniform Security, Privacy, and Integrity Controls
- Data Accessibility and Responsiveness
- Data Independence
- Reduced Program Maintenance
- Enforcement of Standards

DBMS (Database) Costs

- Specialized Personnel
- Installation and Management Cost
- Conversion Costs
- Need for Backup and Recovery
- Organizational Conflict

Why use a DBMS?

- You don't need to write all the code to manage your data
- It will gracefully scale to VERY large collections of data
- It will support transactions that are
  - Atomic (all or nothing)
  - Consistent (from valid state to valid state)
  - Isolated (no interference from concurrent use)
  - Durable (once committed is part of DB)
- Easy to port data to other DBMS or files

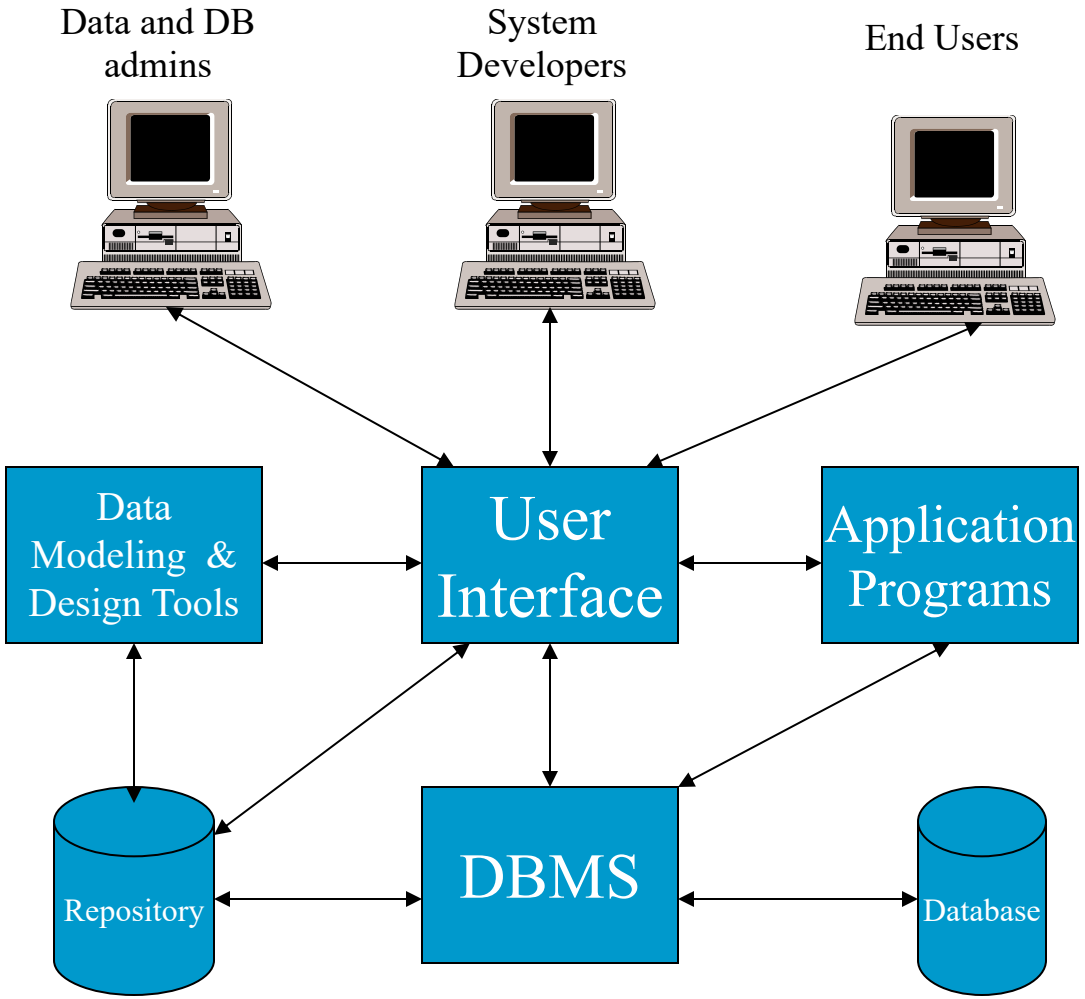
Terms and Concepts

- Database activities:
  - ***Create***
    - Add new data to the database
  - ***Read***
    - Read current data from the database
  - ***Update***
    - Update or modify current database data
  - ***Delete***
    - Remove current data from the database

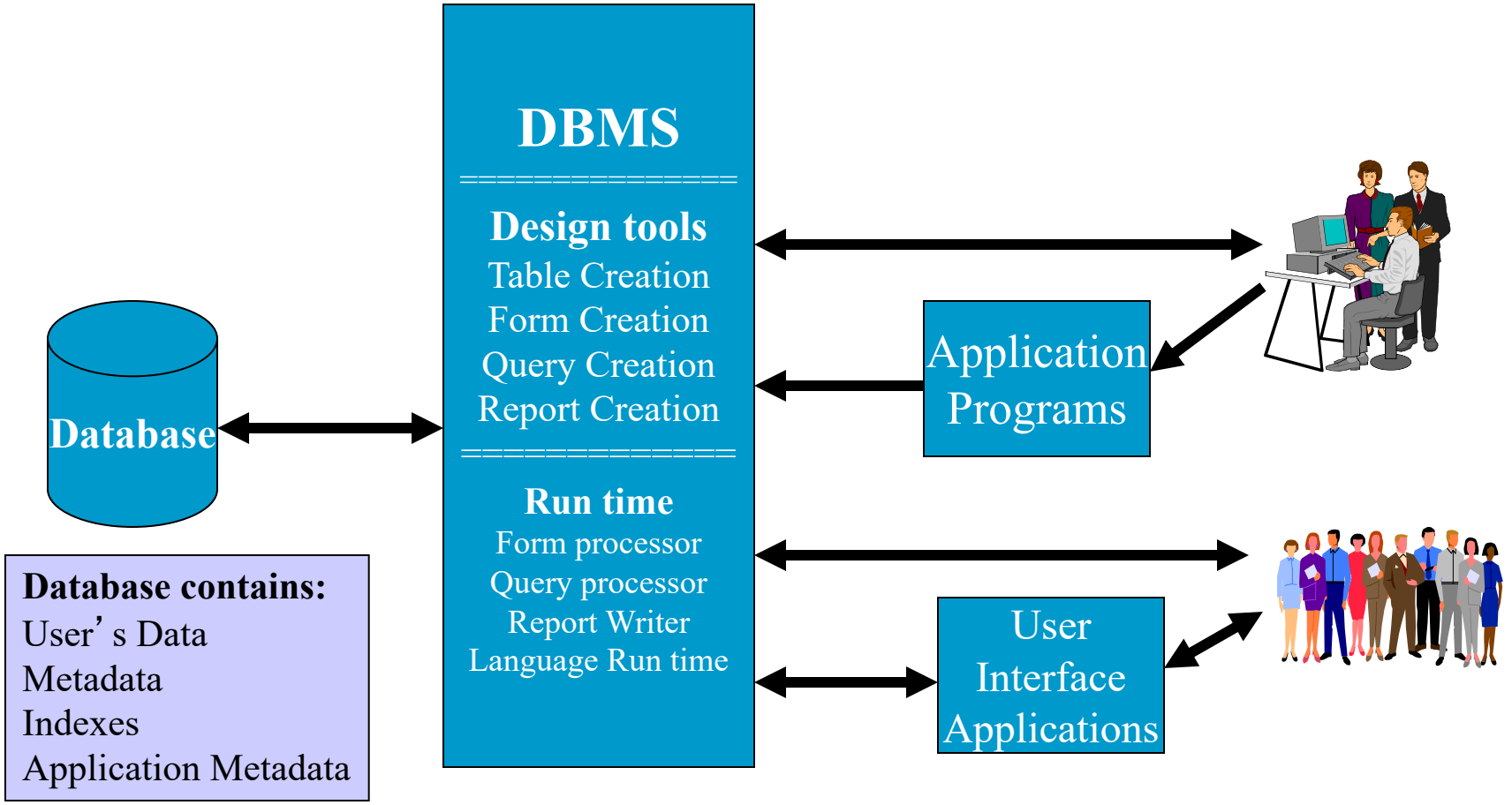
Terms and Concepts

- ***Data Independence***
  - Physical representation and location of data and the use of that data are separated
    - The application doesn't need to know how or where the database has stored the data, but just how to ask for it.
    - Moving a database from one DBMS to another should not have a material effect on application programs
    - Recoding, adding fields, etc. in the database should not affect applications

Database Environment



Database Components



Integrated Data Management Framework

	Operational	Informational	
	Transactional	Analytical– Data Warehousing	Analytical– Big Data
Technology	Relational	Relational	Non-relational
Modeling	Conceptual data modeling with (E)ER (Chapters 2 and 3)		
Design	Logical data modeling with the relational model; Normalization (Chapter 4)	Data warehousing and data integration (Chapter 9)	Big data technologies, including Hadoop & NoSQL (Chapter 10)
Infrastructure	Physical design of relational databases; Security; Cloud computing (Chapter 8)		
Access	SQL (Chapters 5 and 6) Applications with SQL (Chapter 7)		
Data analysis	Analytics and its implications (Chapter 11)		
Governance and data management	Lifecycle (Chapter 1) Governance, data quality, and master data management (Chapter 12)		

Types of Database Systems

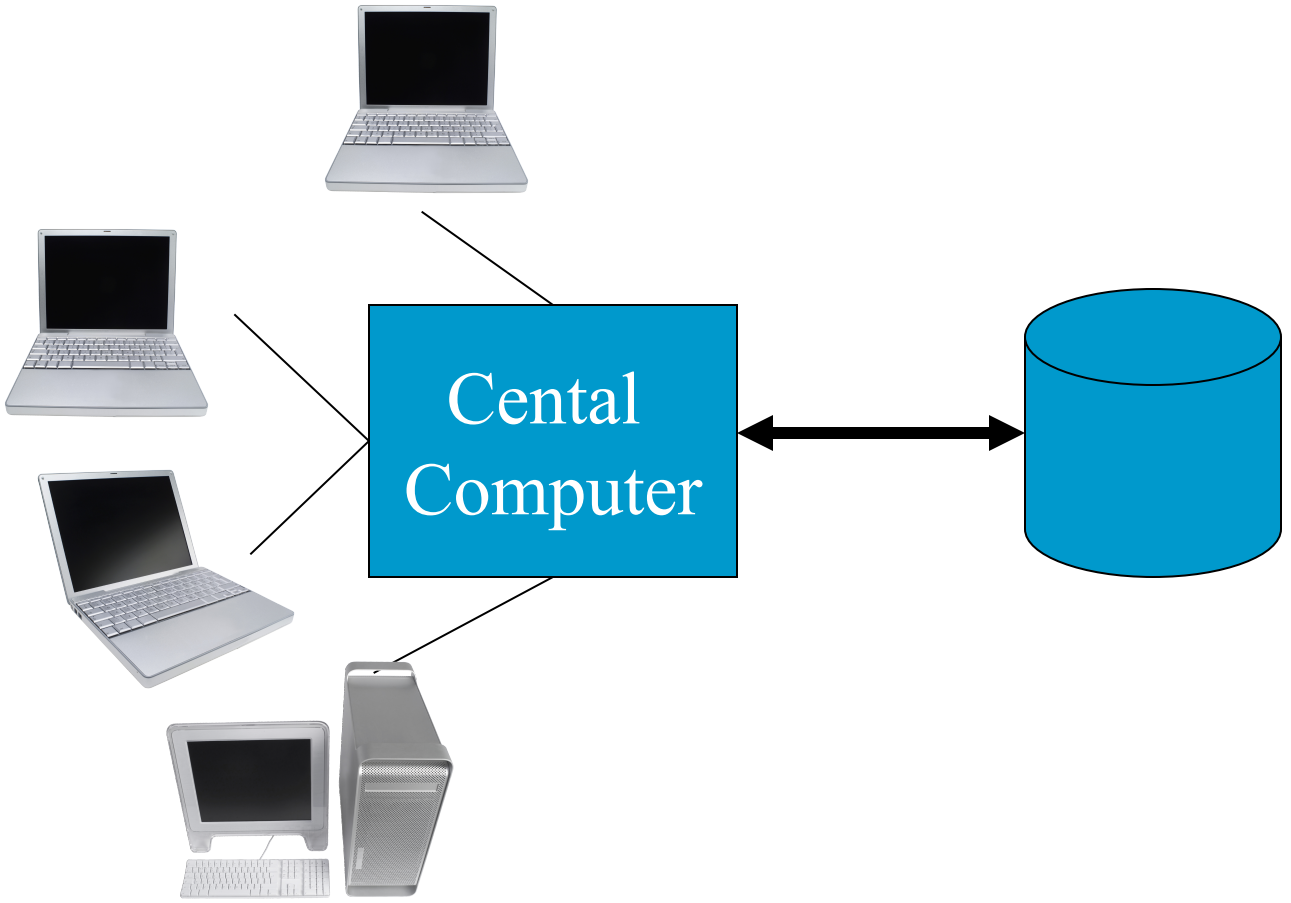
- Local Databases
- Centralized Database
- Client/Server Databases
- Distributed Databases
- Cloud-Based Databases

- Database Models

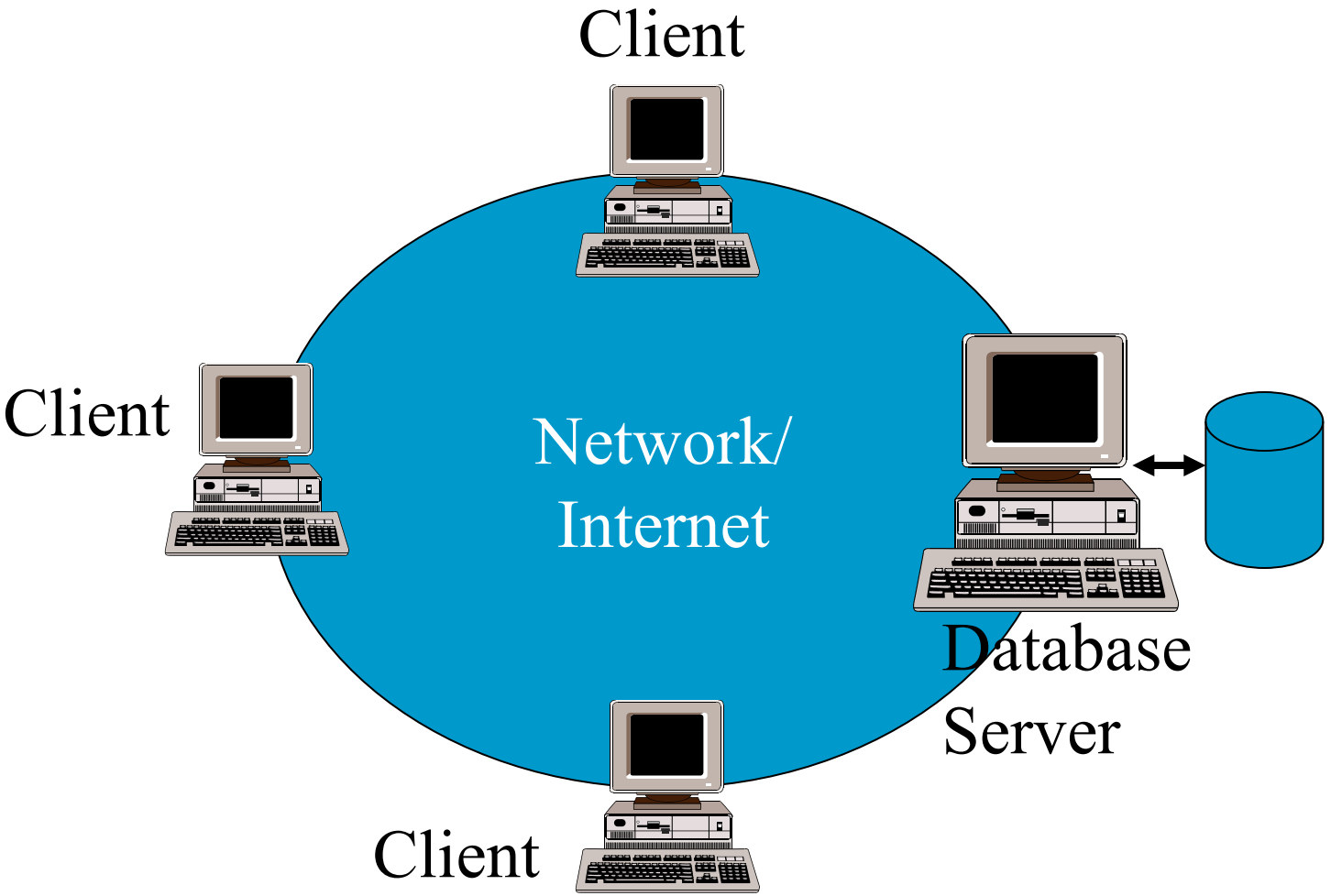
Local Databases

E.G.,
Access
SQLite
Etc.

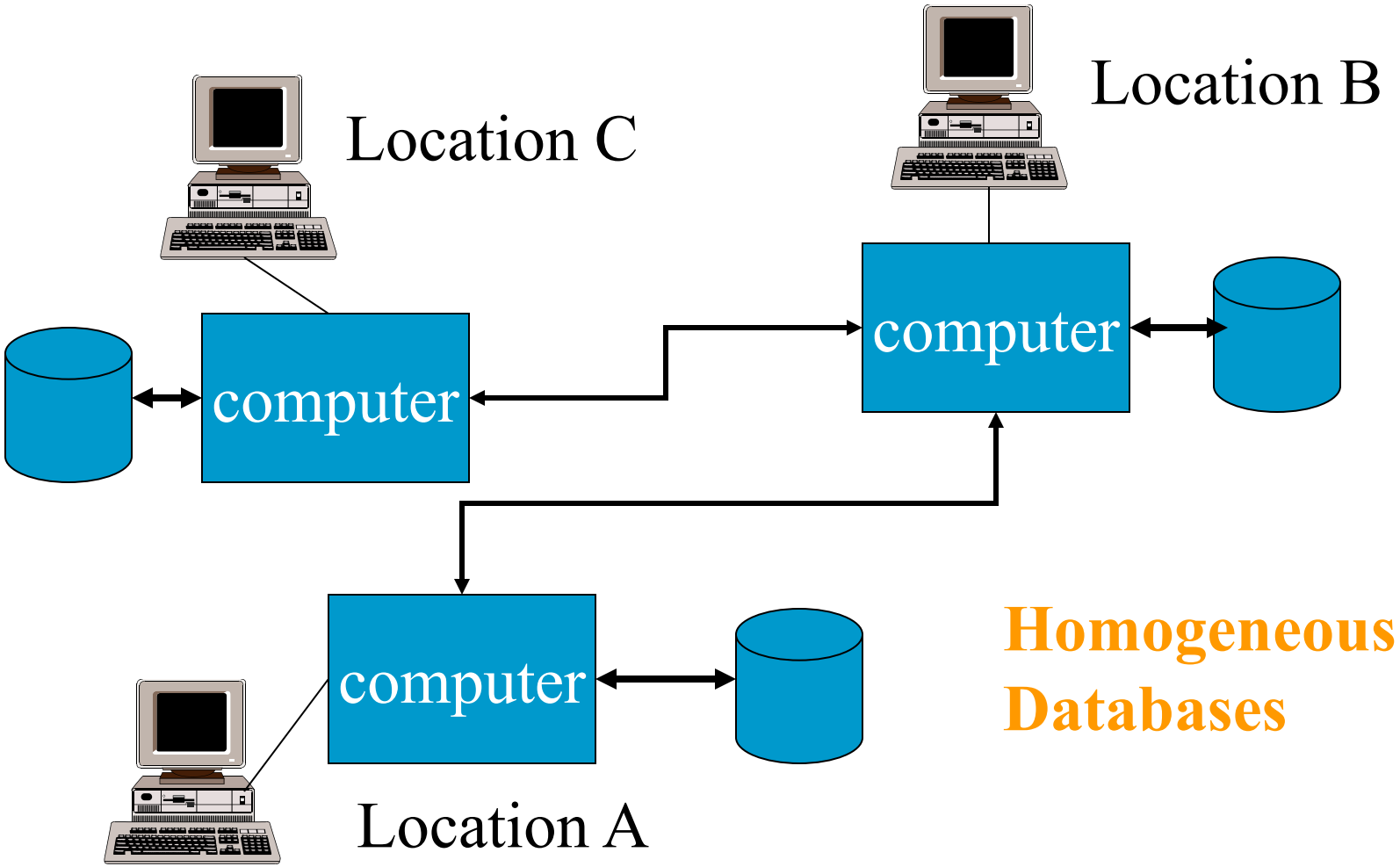
Centralized Databases



Client Server Databases

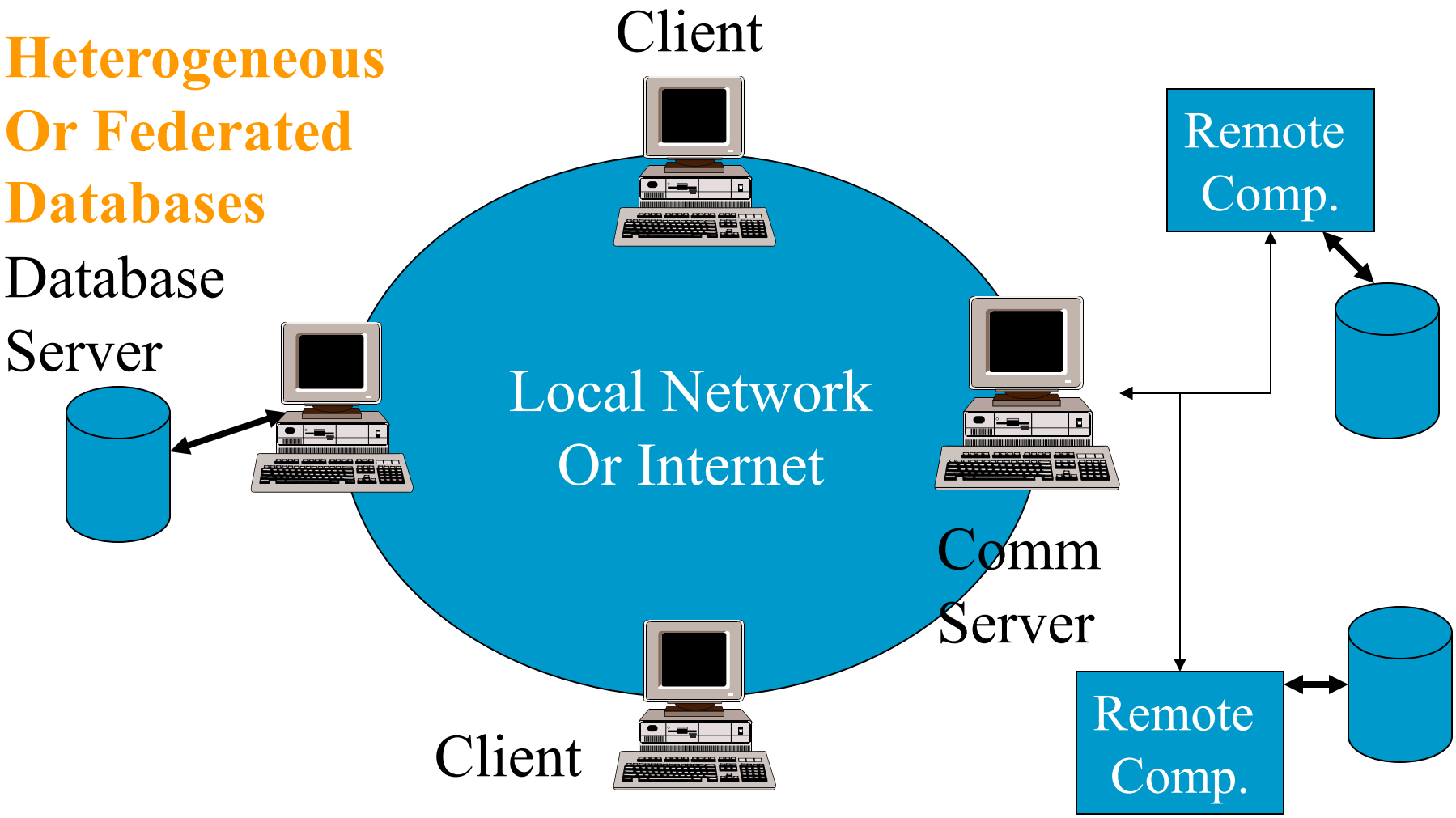


Distributed Databases (DDBMS)



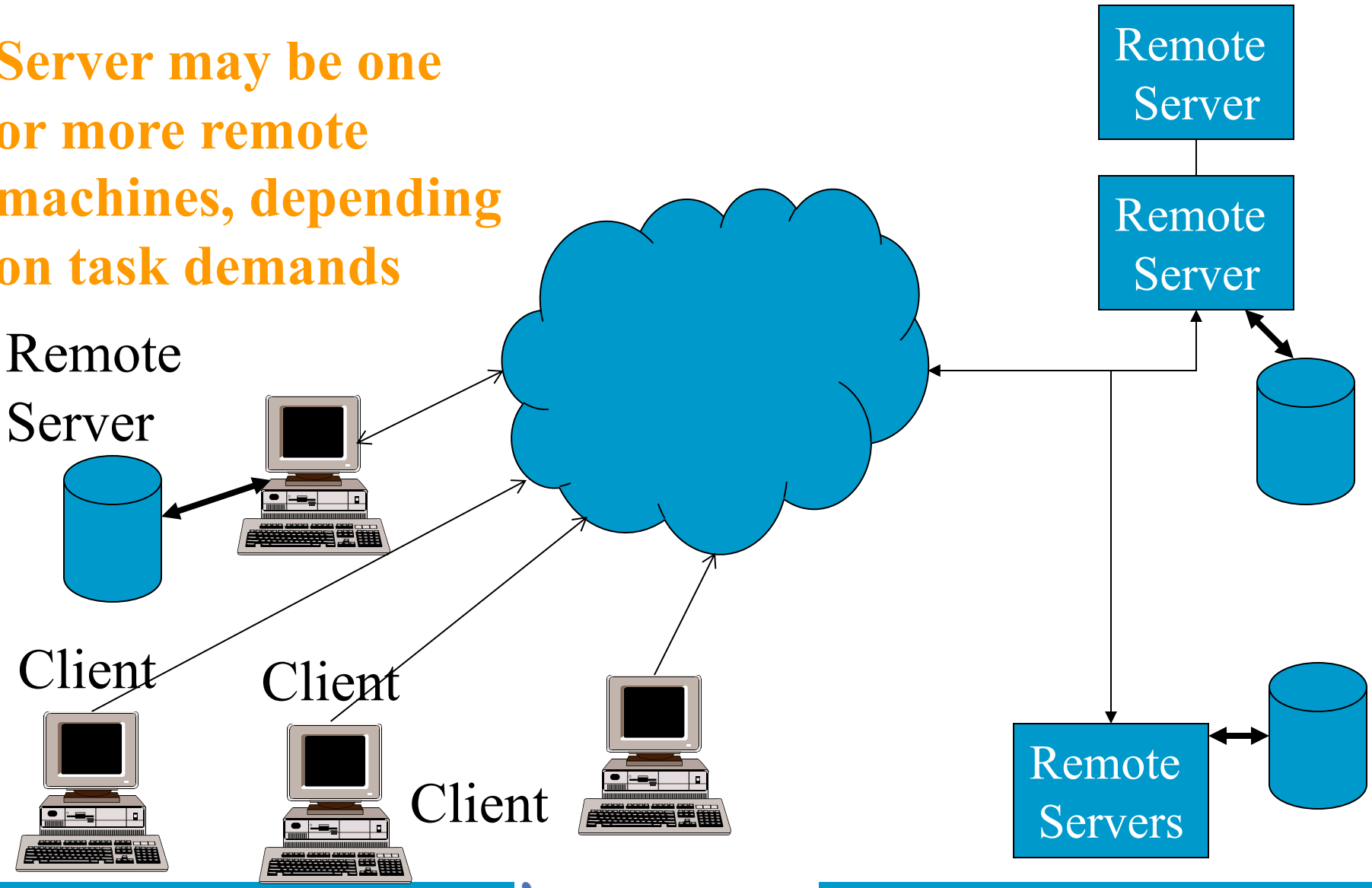
Distributed Databases (DDBMS)

Heterogeneous Or Federated Databases



Cloud-Based Databases

Server may be one or more remote machines, depending on task demands



Terms and Concepts

- ***Database Application***
  - An application program (or set of related programs) that is used to perform a series of database activities:
    - Create
    - Read
    - Update
    - Delete
  - On behalf of application users

Terms and Concepts

- ***Enterprise***
  - Organization
- ***Entity***
  - Person, Place, Thing, Event, Concept...
- ***Attributes***
  - Data elements (facts) about some entity
  - Also sometimes called fields or items or domains
- ***Data values***
  - instances of a particular attribute for a particular entity

The Range of Database Applications

- Personal Databases
  - Typical size in the megabytes
  - Intended for one user
    - E.g. SQLite is used by many iPhone Apps including IOS itself
- Departmental Multi-Tiered Client/Server Databases
  - Typical size in the gigabytes
  - Intended for several users, usually doesn't exceed 100, department-wide
- Enterprise Applications
  - Typical size in the gigabytes, terabytes, or even petabytes
  - Intended for a very large user base, company wide

Types of Enterprise Applications

- Enterprise Systems (typically involve relational databases)
  - Backbone of an organization
  - Enterprise resource planning (ERP)
  - Customer relationship management
  - Supply chain management
  - Human resource management and payroll
- Data Warehouses (typically involve relational databases)
  - Integrates data from multiple data sources
  - Maintain historical data
  - Help identify patterns and trends
- Data Lakes (often don't involve relational databases)
  - Large integrated repository for internal and external data that does not follow a predefined schema

Terms and Concepts

- ***Records***
  - The set of values for all attributes of a particular entity
  - AKA "tuples" or "rows" in relational DBMS
- ***File***
  - Collection of records
  - AKA "Relation" or "Table" in relational DBMS

Terms and Concepts

- ***Key***
  - an attribute or set of attributes used to identify or locate rows in a table
- ***Primary Key***
  - an attribute or set of attributes that ***uniquely*** identifies each row in a table

Terms and Concepts

- *DA*
  - Data adminstrator - person responsible for the Data Administration function in an organization
  - Sometimes may be the
    - CIO -- Chief Information Officer
    - CDO – Chief Data Officer
- *DBA*
  - Database Administrator - person responsible for the Database Administration Function

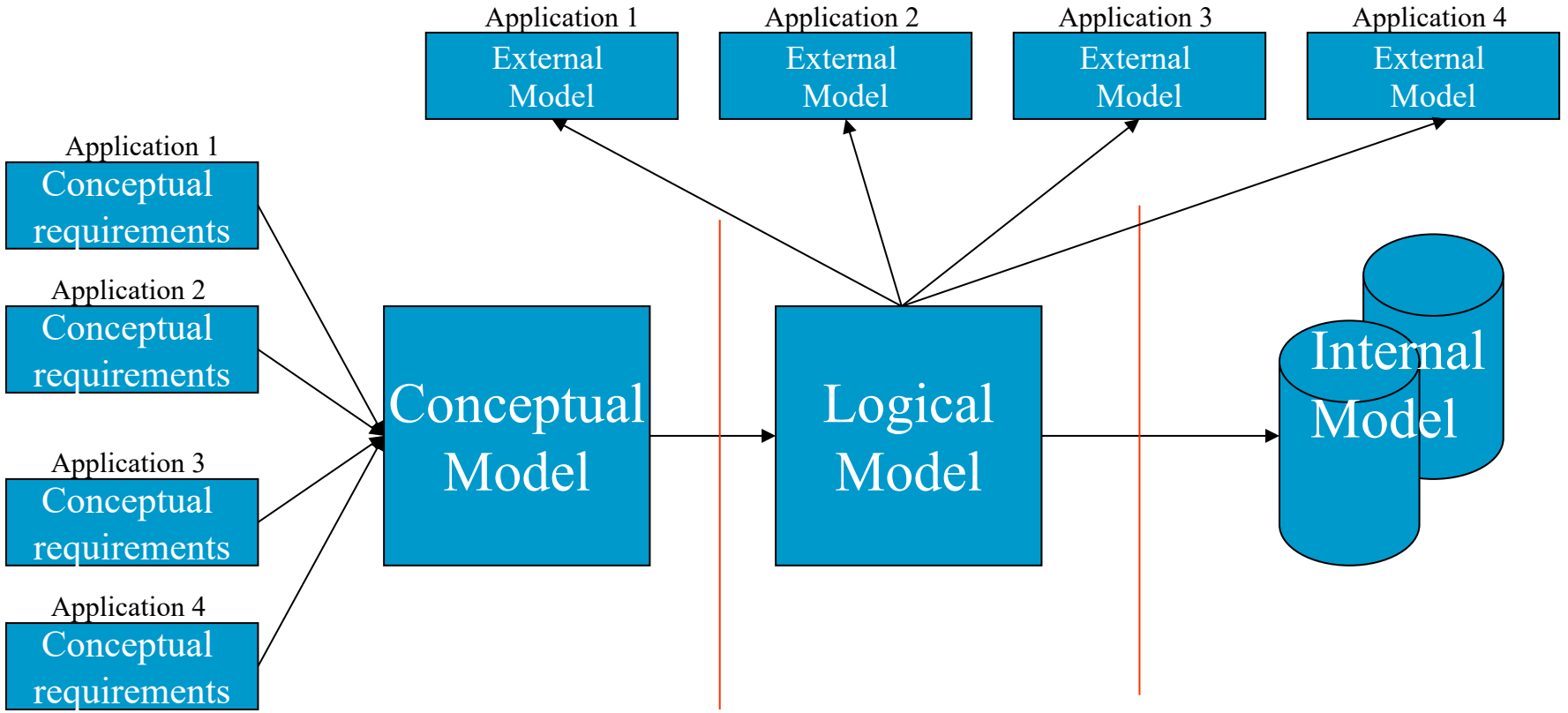
Terms and Concepts

- ***Models***
  - (1) Levels or views of the Database
    - Conceptual, logical, physical
  - (2) DBMS types
    - Relational, Hierarchic, Network, Object-Oriented, Object-Relational

Overview

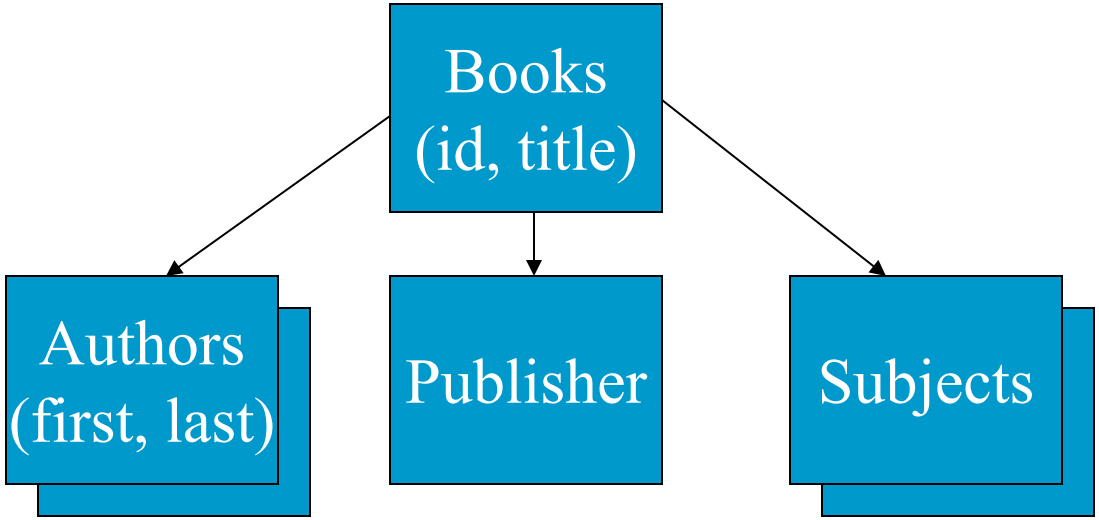
- Announcements
- Course Description
- Database Concepts and Terminology
- Database Models
- Database Life Cycle

Models (1)


Application 1
External Model
Application 2
External Model
Application 3
External Model
Application 4
External Model
Application 1
Conceptual requirements
Application 2
Conceptual requirements
Application 3
Conceptual requirements
Application 4
Conceptual requirements
Conceptual Model
Logical Model
Internal Model

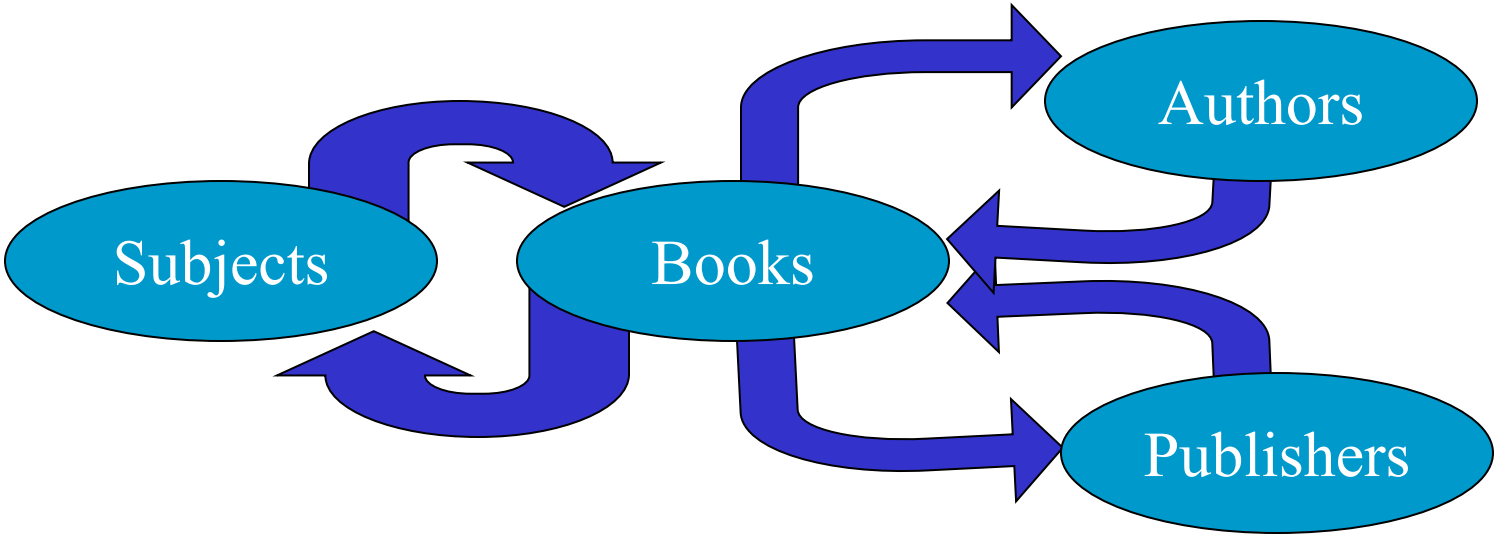
Data Models(2): History

- Hierarchical Model (1960's and 1970's)
  - Similar to data structures in programming languages.



Data Models(2): History

- Network Model (1970's)
  - Provides for single entries of data and navigational "links" through chains of data.



Data Models(2): History

- Relational Model (1980's)
  - Provides a conceptually simple model for data as relations (typically considered "tables") with all data visible.

Book ID	Title	pubid	Author id
1	Introductio	2	1
2	The history	4	2
3	New stuff a	3	3
4	Another tit	2	4
5	And yet m	1	5

pubid	pubname
1	Harper
2	Addison
3	Oxford
4	Que

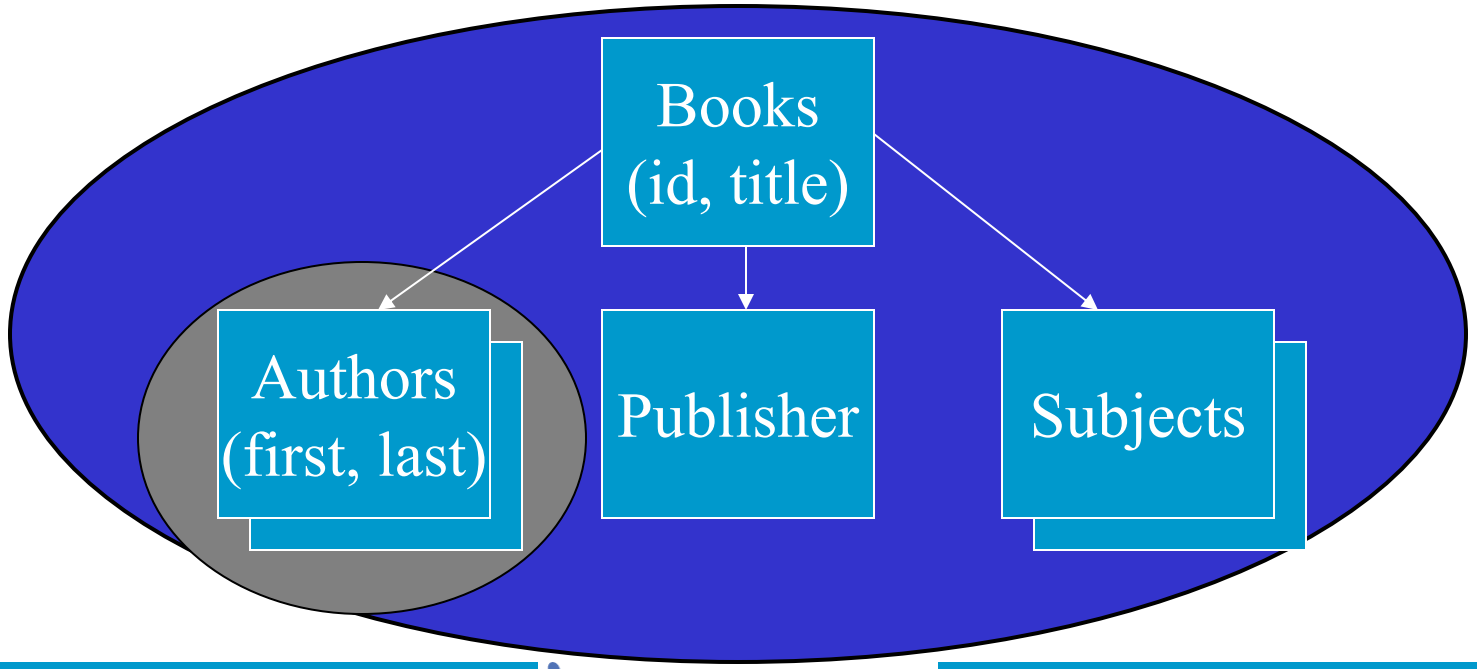
Authorid	Author nan
1	Smith
2	Wynar
3	Jones
4	Duncan
5	Applegate

Book ID	Subid
1	2
2	1
3	3
4	2
4	3

Subid	Subject
1	cataloging
2	history
3	stuff

Data Models(2): History

- Object Oriented Data Model (1990's)
  - Encapsulates data and operations as "Objects"



Data Models(2): History

- Object-Relational Model (1990's)
  - Combines the well-known properties of the Relational Model with such OO features as:
    - User-defined datatypes
    - User-defined functions
    - Inheritance and sub-classing

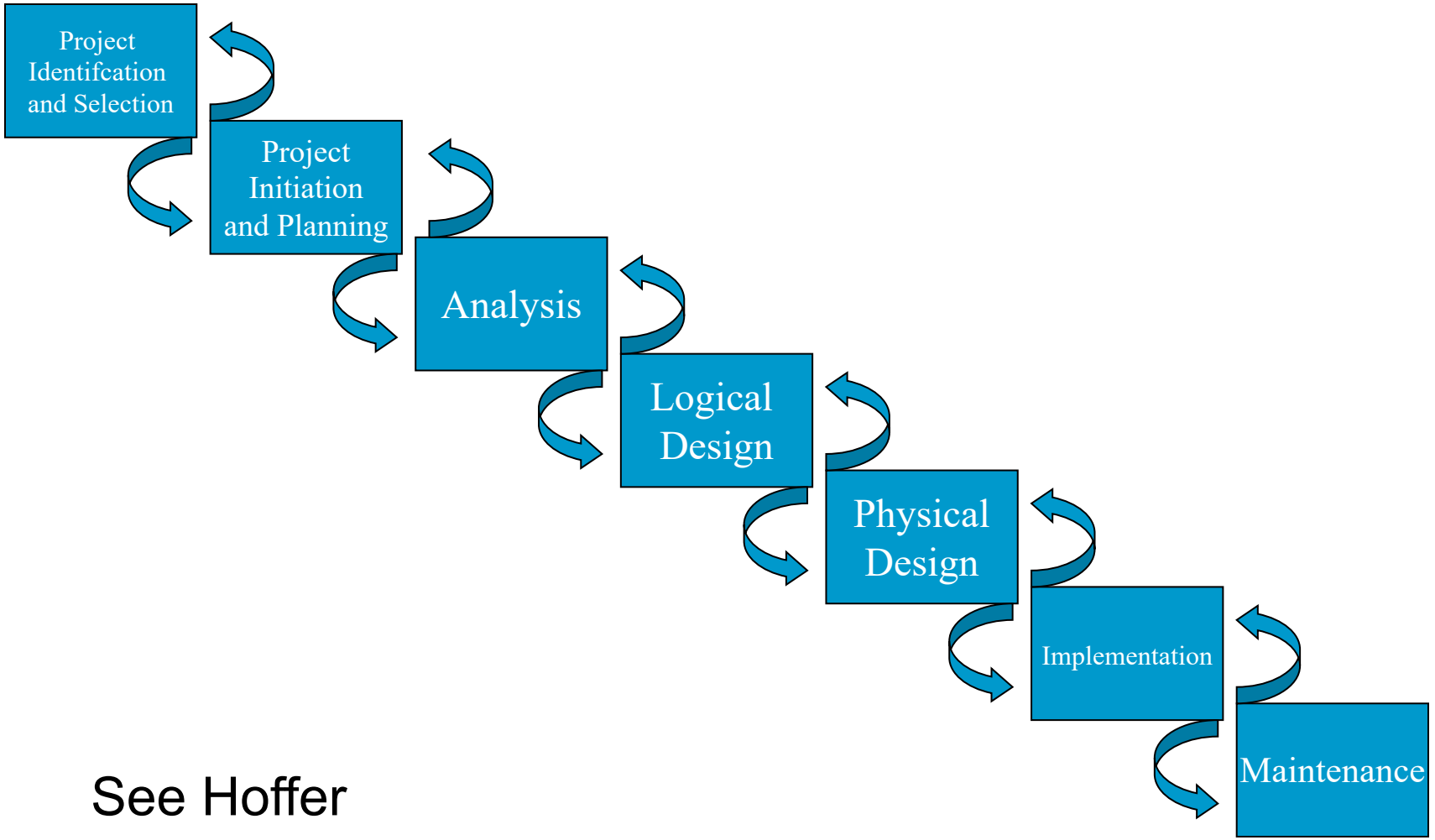
NoSQL Databases

- Started as a reaction to the overhead in more conventional SQL DBMS
- Usually very simple key/value search operations
- Usually very fast, with low storage overhead, but at the expense of consistency and other features of RDBMS
- May use distributed parallel processing (grid/cloud, e.g. MongoDB + Hadoop)

Overview

- Announcements
- Course Description
- Database Concepts and Terminology
- Database Models
- Database Life Cycle

The "Cascade" View



See Hoffer

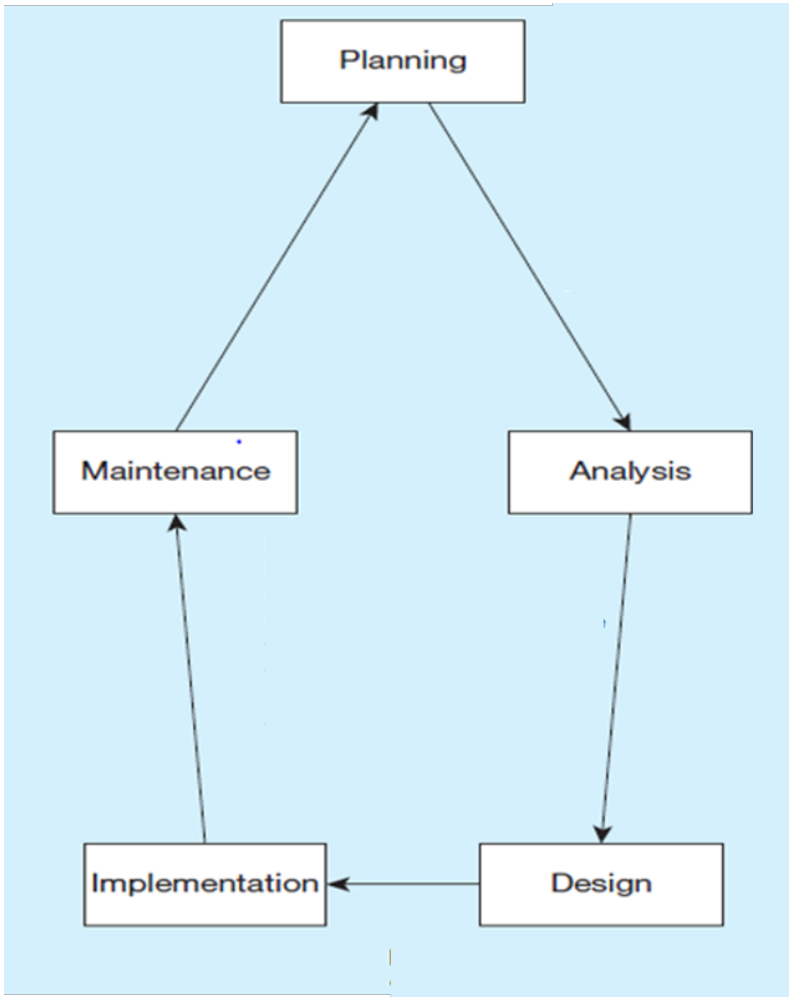
The Database Development Process

- SDLC
  - System Development Life Cycle
  - Detailed, well-planned development process
  - Time-consuming, but comprehensive
  - Long development cycle
- Prototyping
  - Rapid application development (RAD)
  - Cursory attempt at conceptual data modeling
  - Define database during development of initial prototype
  - Repeat implementation and maintenance activities with new prototype versions

Systems Development Life Cycle (SDLC)

- The traditional methodology used to develop, maintain, and replace information systems
- Five main steps:
  1. **Planning** – preliminary understanding of business situation. Enterprise model and conceptual data modeling.
  2. **Analysis** – thorough analysis of business situation, leading to functional requirements. Detailed conceptual data modeling.
  3. **Design** – logical and physical database design, to develop technology and organization.
  4. **Implementation** – writing programs, building databases, testing, installing, training, and documenting.
  5. **Maintenance** – monitoring, repairing, and enhancing.

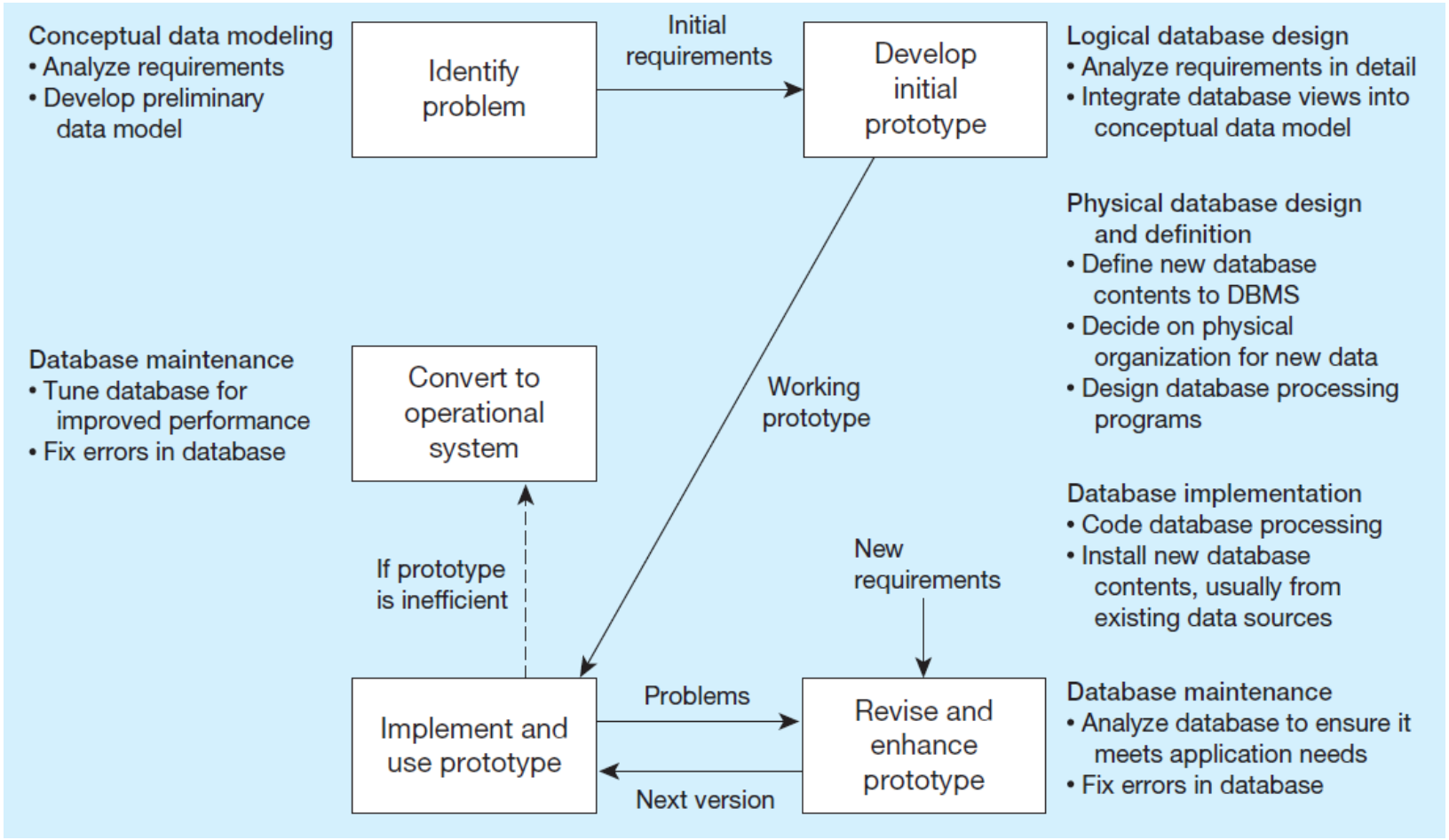
Database Development Activities During the SDLC



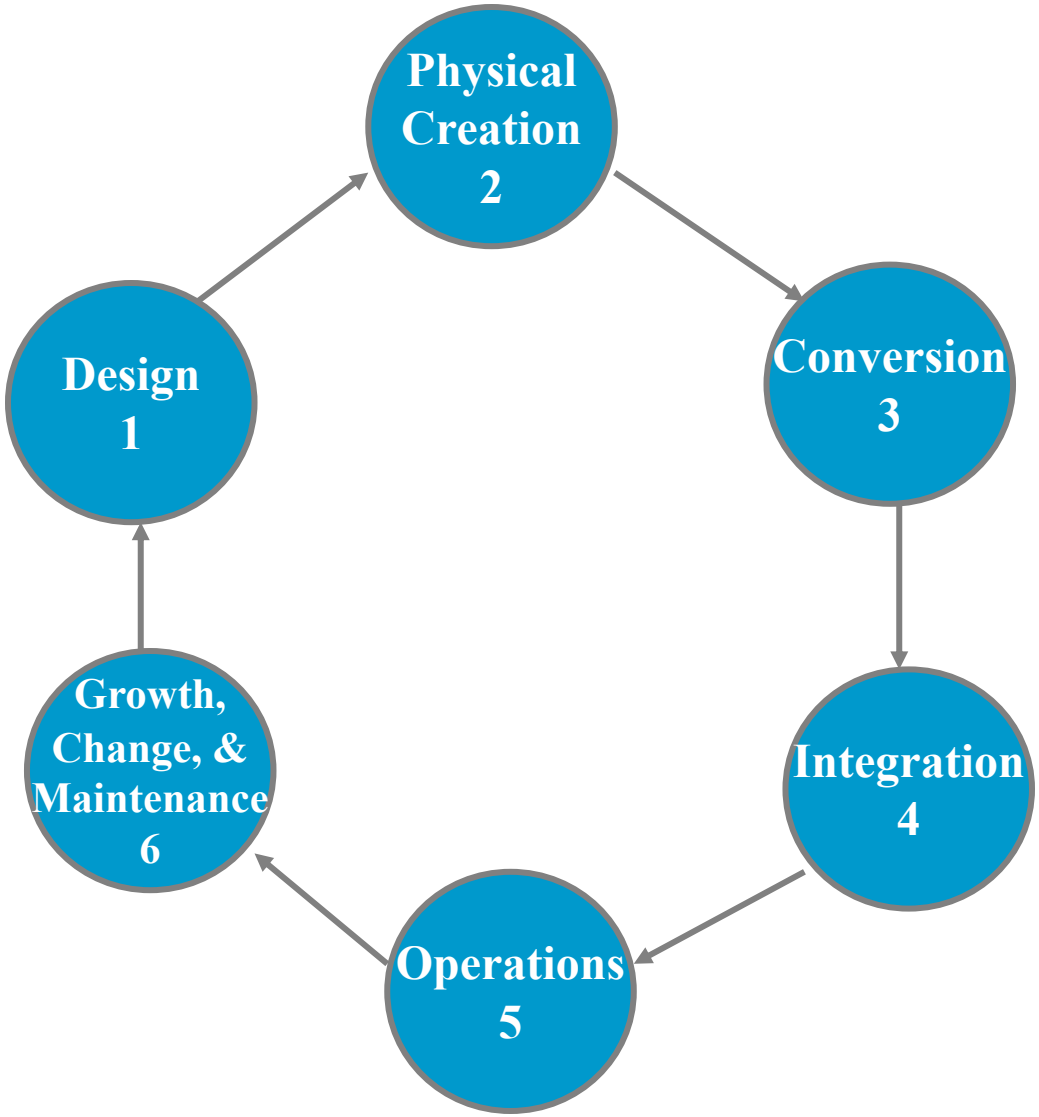
Alternative Information Systems Development Approaches

- Traditional SDLC: methodical, structured, and time consuming
- Rapid Application Development (RAD): faster and more adaptive, especially when a database is already in place
- Several flavors:
  - Prototyping
  - Agile methodologies
  - eXtreme programming
  - Scrum
  - DSDM (dynamic system development methodologies)

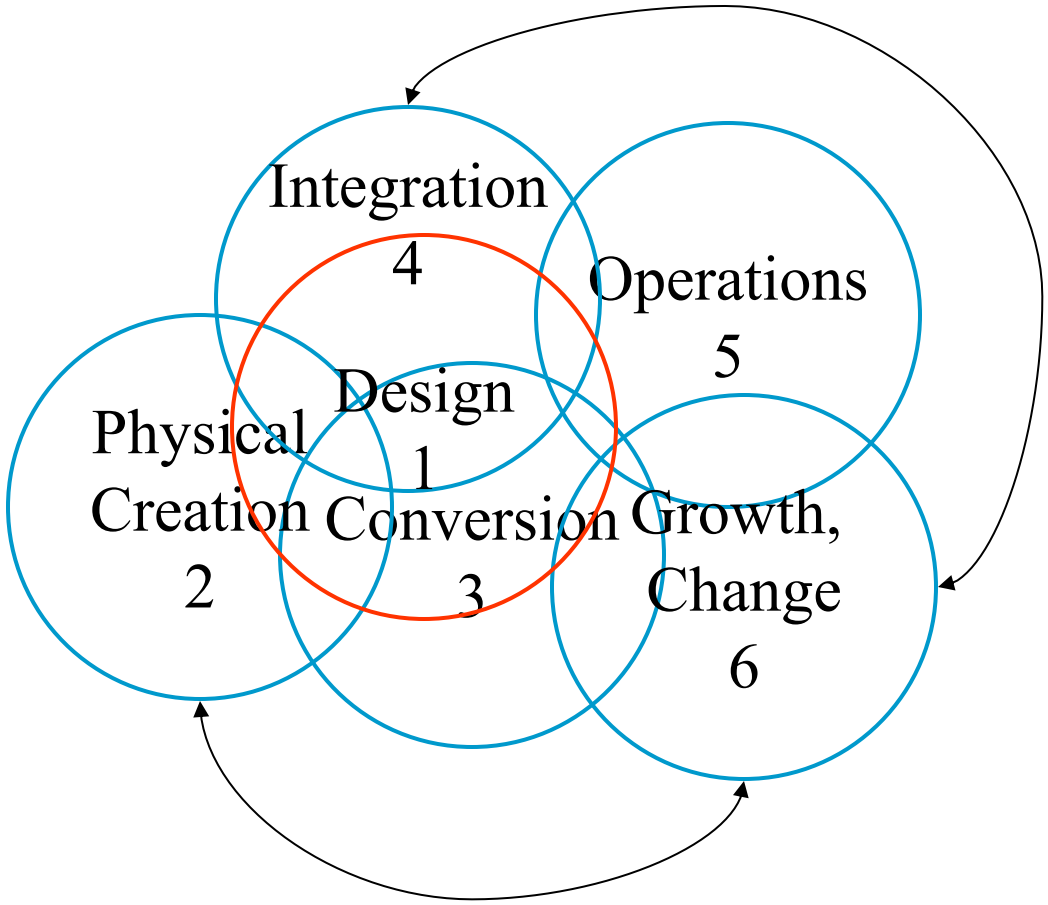
The Prototyping Methodology and Database Development Process



Database System Life Cycle



Another View of the Life Cycle



Next Time

- Introduction to the Diveshop database
- Introduction to Database Design