Database Management: Introduction

Luis Aguilar
University of California, Berkeley
School of Information

Info 257: Database Management

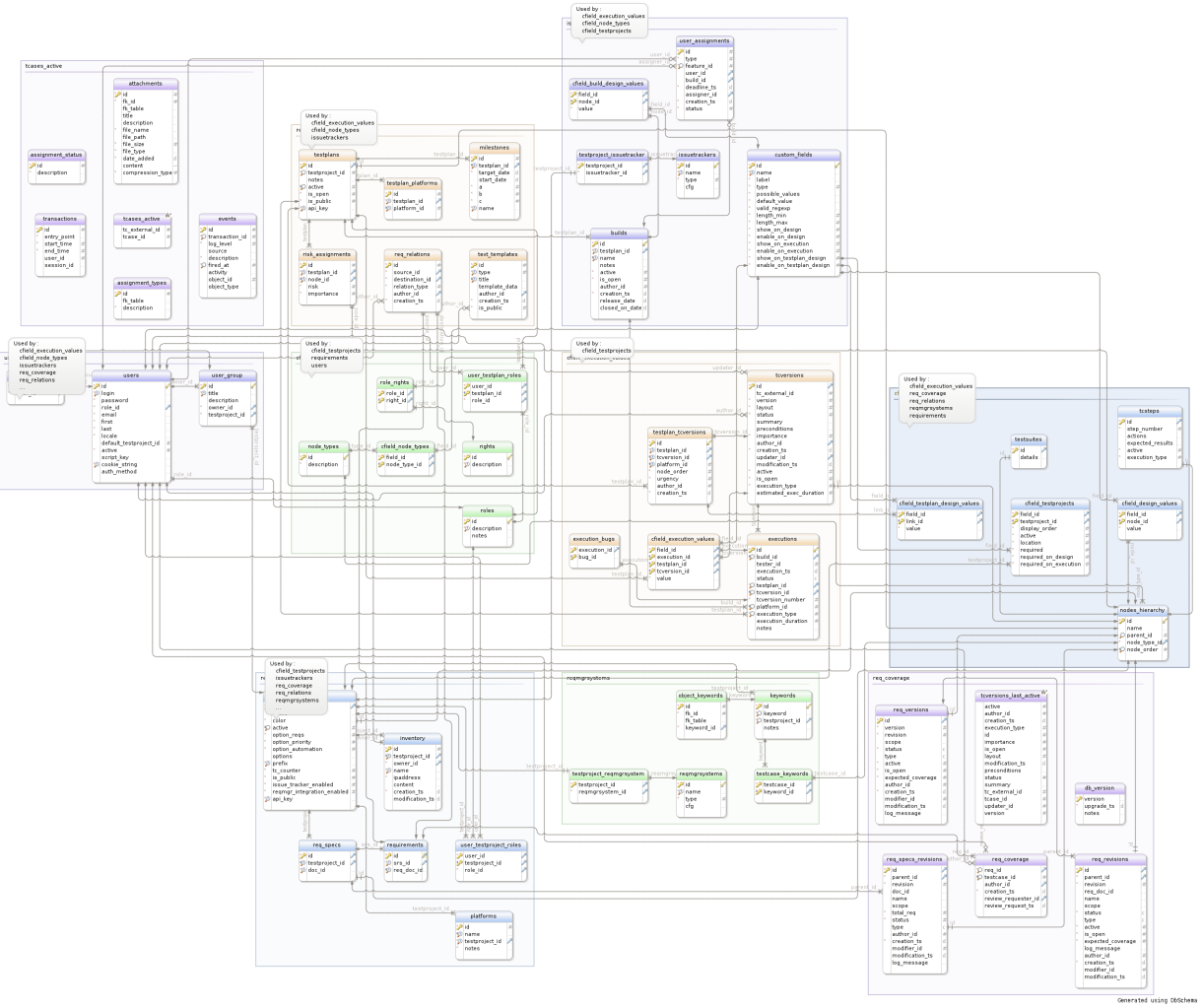
Overview

- 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.

Goal



TA Introduction

- Monik Pamecha
  - monik@berkeley.edu

Announcements

- My first time teaching this particular course, questions and feedback are welcome.

Overview

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

Course Overview

- Description of the Course
- Assignments
- Readings
- Grading
- Schedule
- Web site: http://courses.ischool.berkeley.edu/i257/s19

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

- Two kinds of assignments
  - Using a pre-built database for search and retrieval and database modification queries
  - 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…

Readings

- Textbook is:
  - Jeffrey A. Hoffer, Ramesh Venkataraman and Heikki Topi. ***Modern Database Management (12th Edition).*** Pearson : , 2015.
    - ISBN 978-0-13-354461-9 or 0-13-354461-3
  - 11th, 10th, Ninth (or even earlier) editions are OK too
  - Please note, the International Editions that you see on Amazon are good enough as well so you don't have to pay $$$

Grading

- Grades will be based on:
  - Assignments (35%)
  - Group Database project (65%)
  - (No midterm or final)

Schedule

- On website:
- http://courses.ischool.berkeley.edu/i257/s19

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?

- Broadly speaking – any collection of data might be called a database
- Normally, though, certain characteristics of the data and how it is stored distinguish a database from any random or adhoc collection of data

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."

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

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

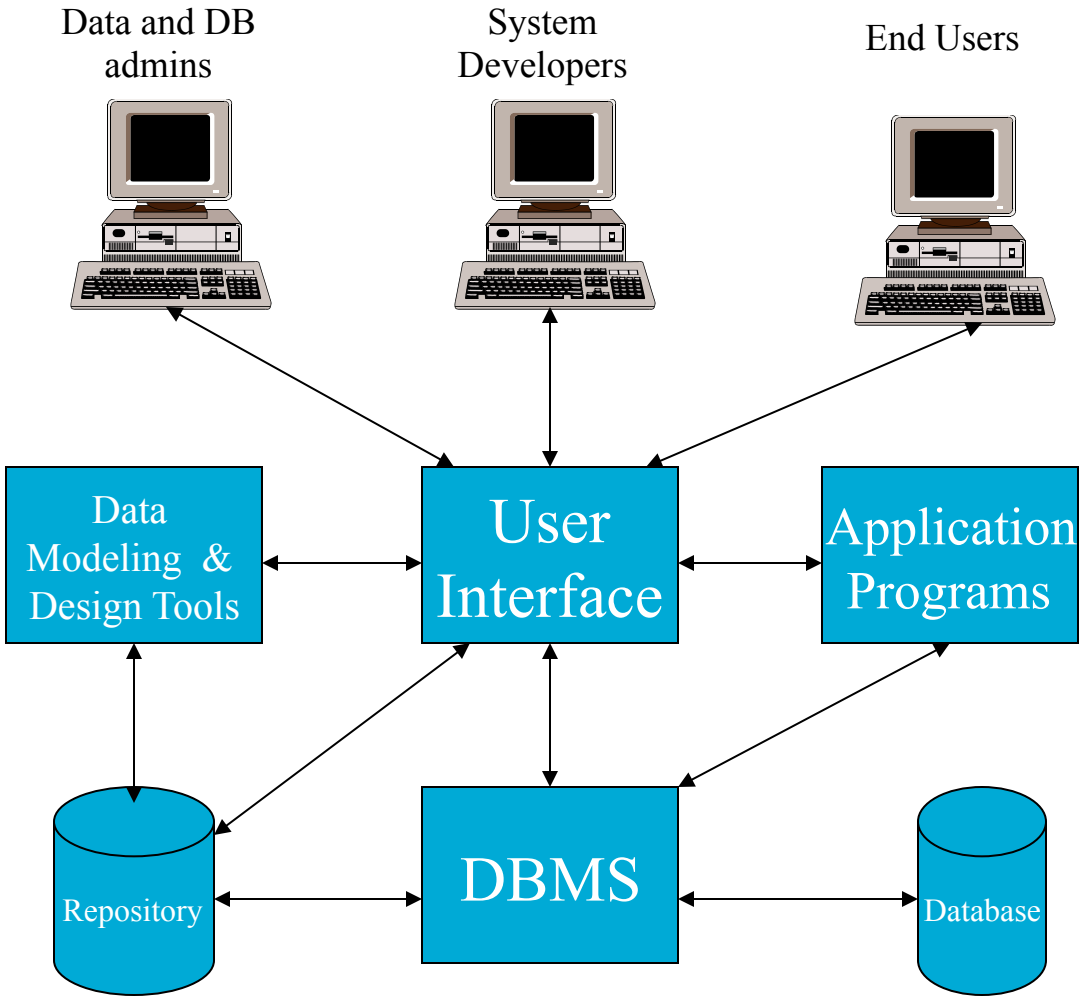
Terms and Concepts

- 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

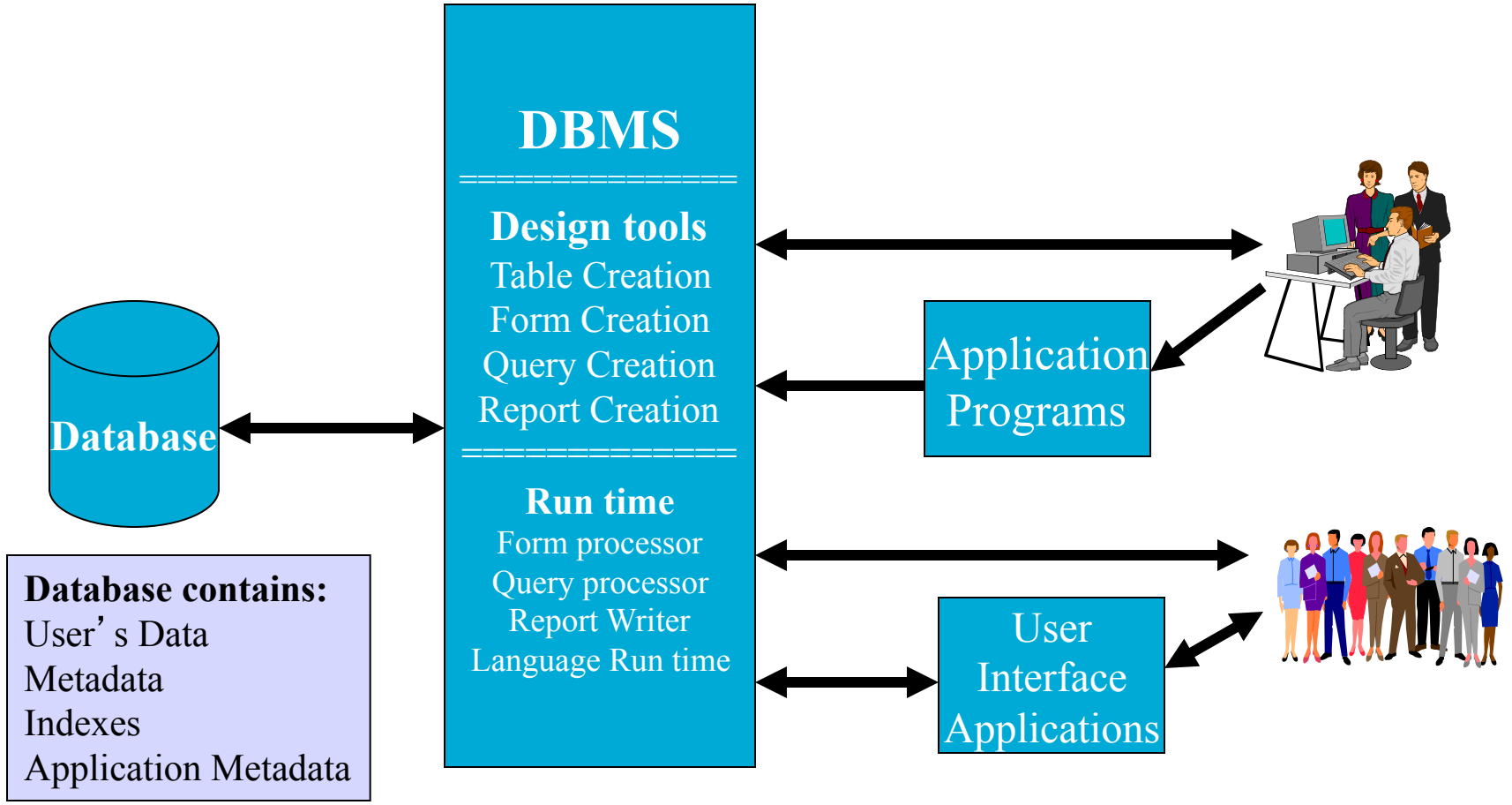
Terms and Concepts

- ***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



Types of Database Systems

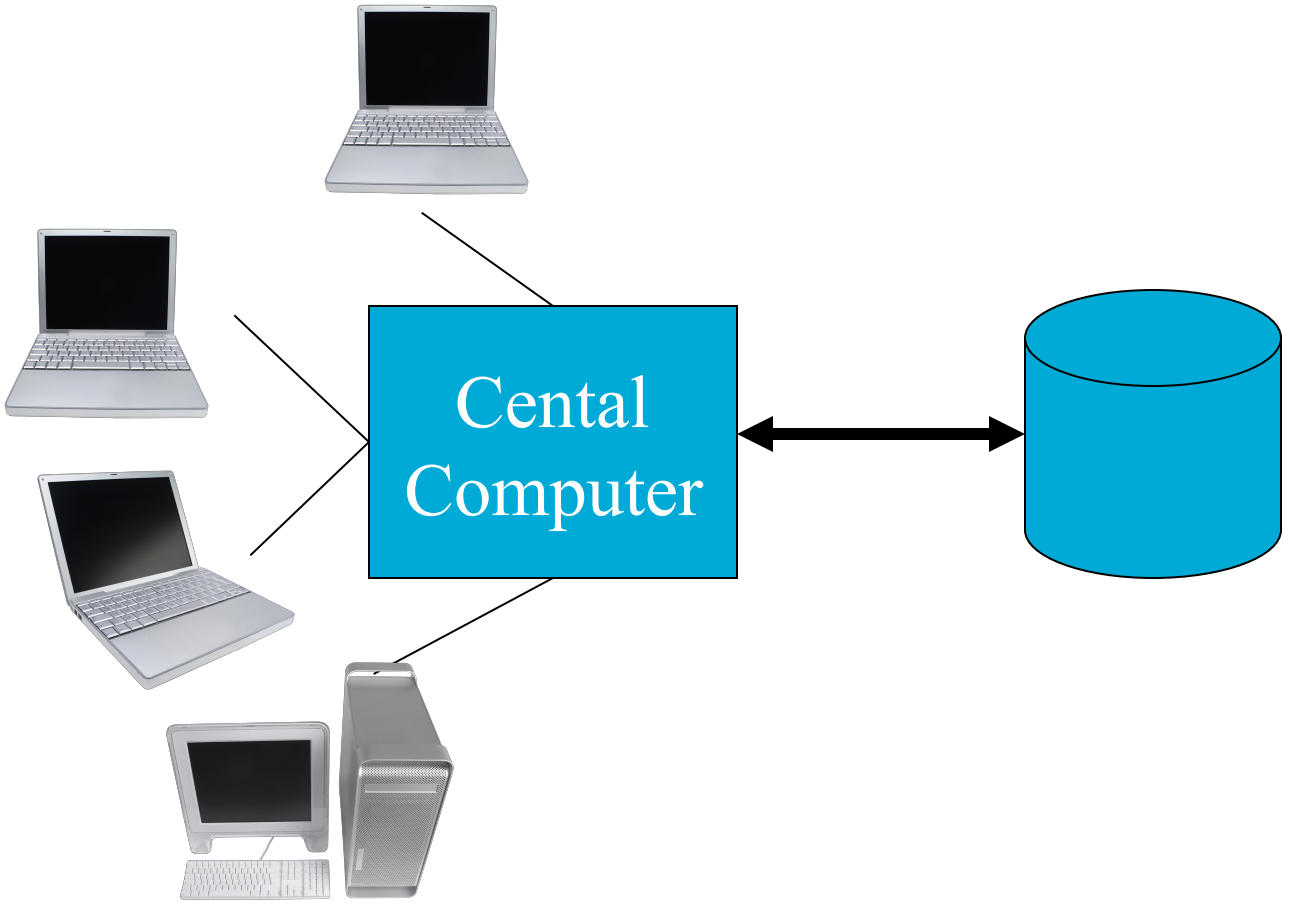
- Local Databases
- Centralized Database
- Client/Server Databases
- Distributed Databases
- Cloud-Based Databases

- Database Models

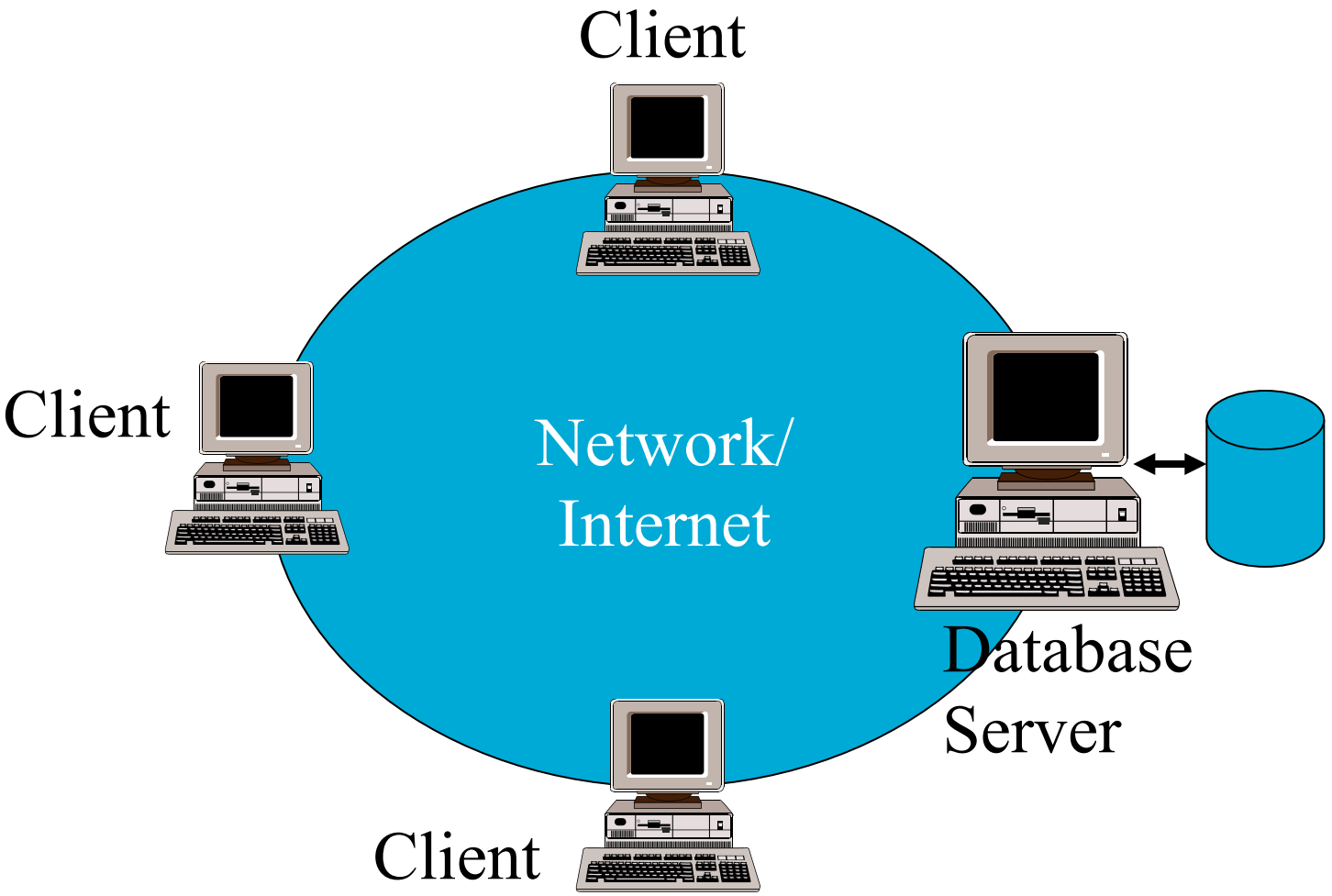
Local Databases

E.G.,
Access
SQLite
Etc.

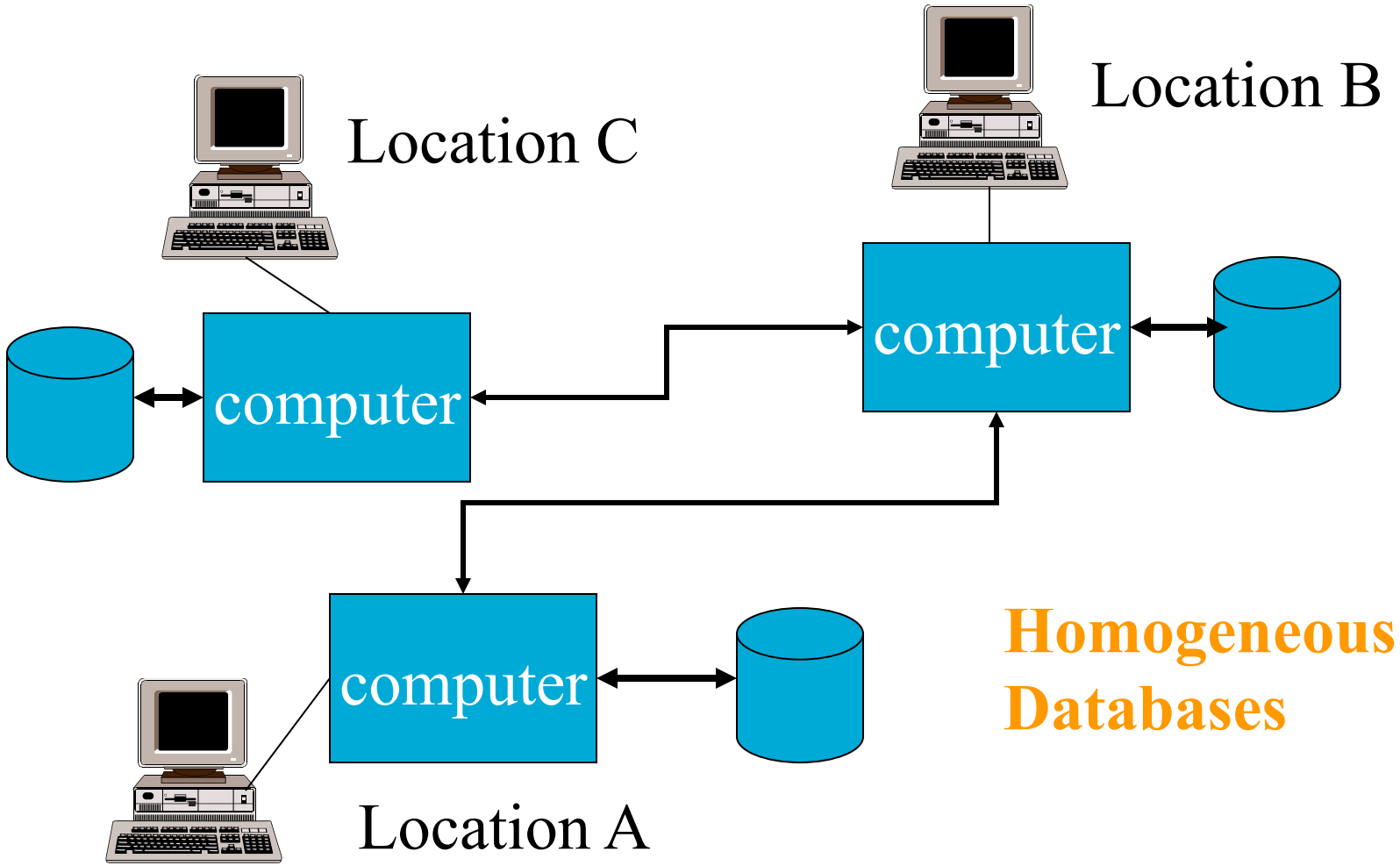
Centralized Databases



Client Server Databases

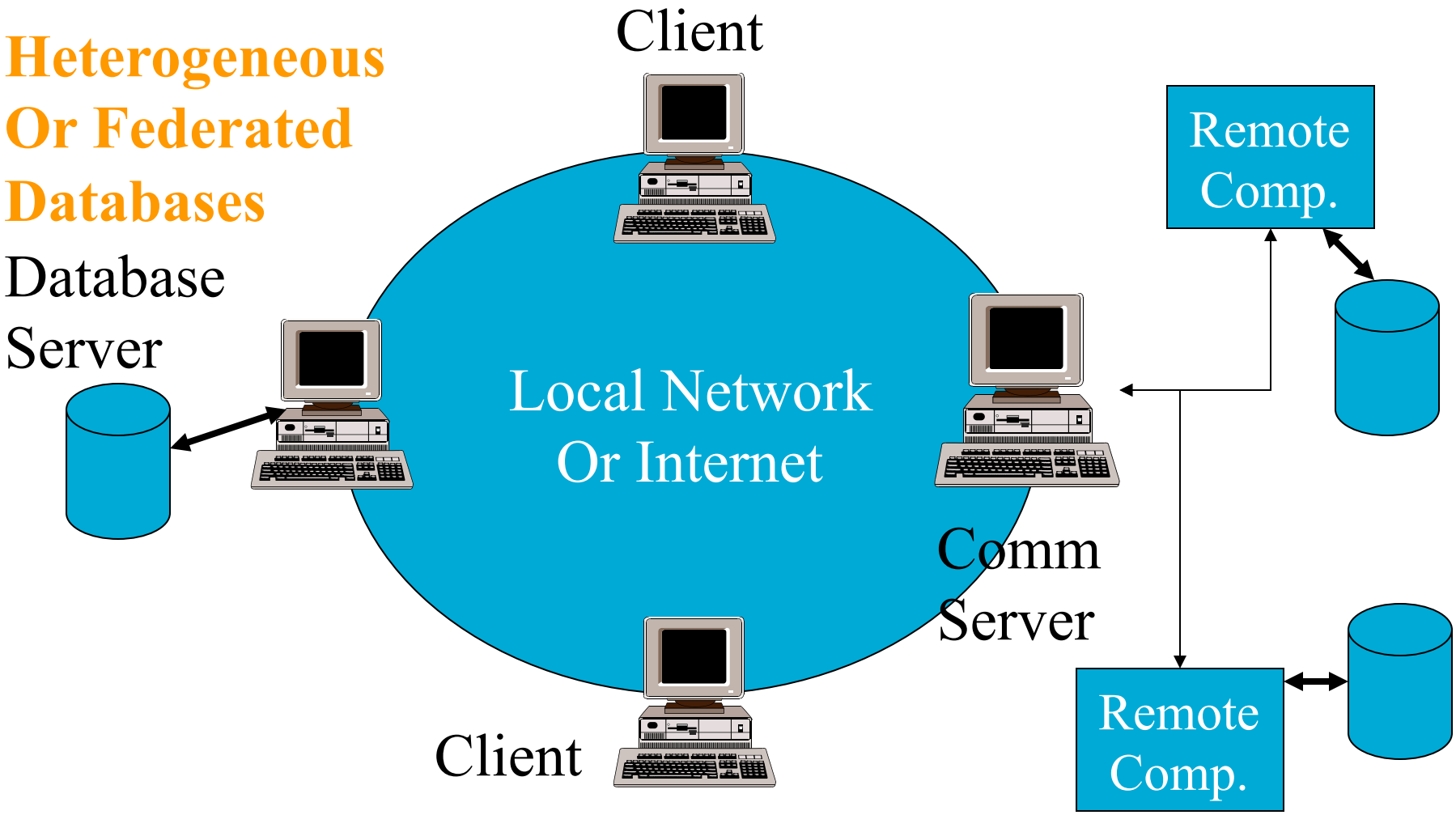


Distributed Databases (DDBMS)



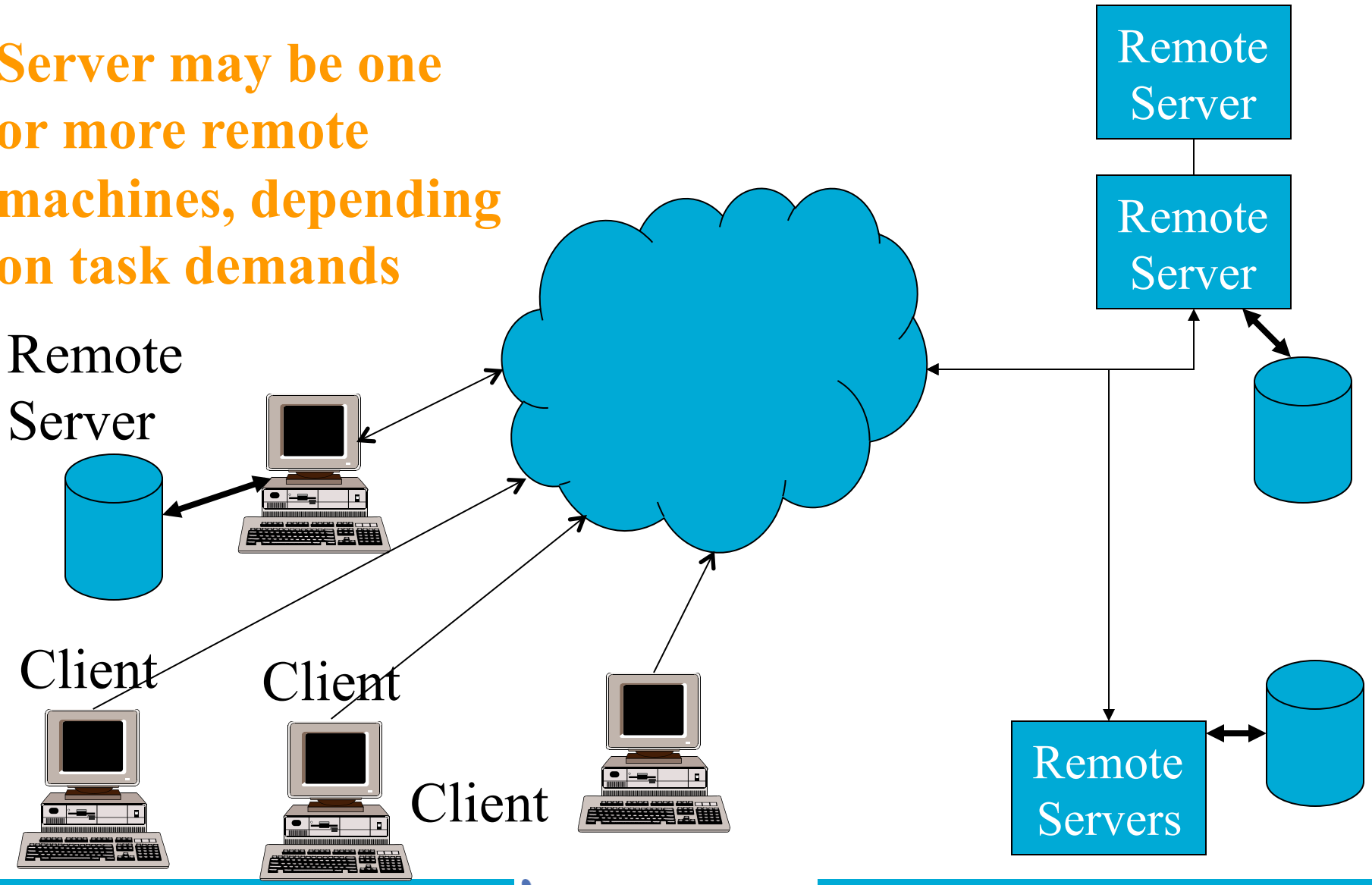
Distributed Databases (DDBMS)

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 database users

Range of Database Applications

- Local databases
  - Usually for individual user applications
    - E.g. SQLite is used by many iPhone Apps including IOS itself
- WorkGroup databases
  - Small group use where everyone has access to the database over a LAN (or internet)
- Departmental databases
  - Larger than a workgroup – but similar
- Enterprise databases
  - For the entire organization over an intranet or the internet

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

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

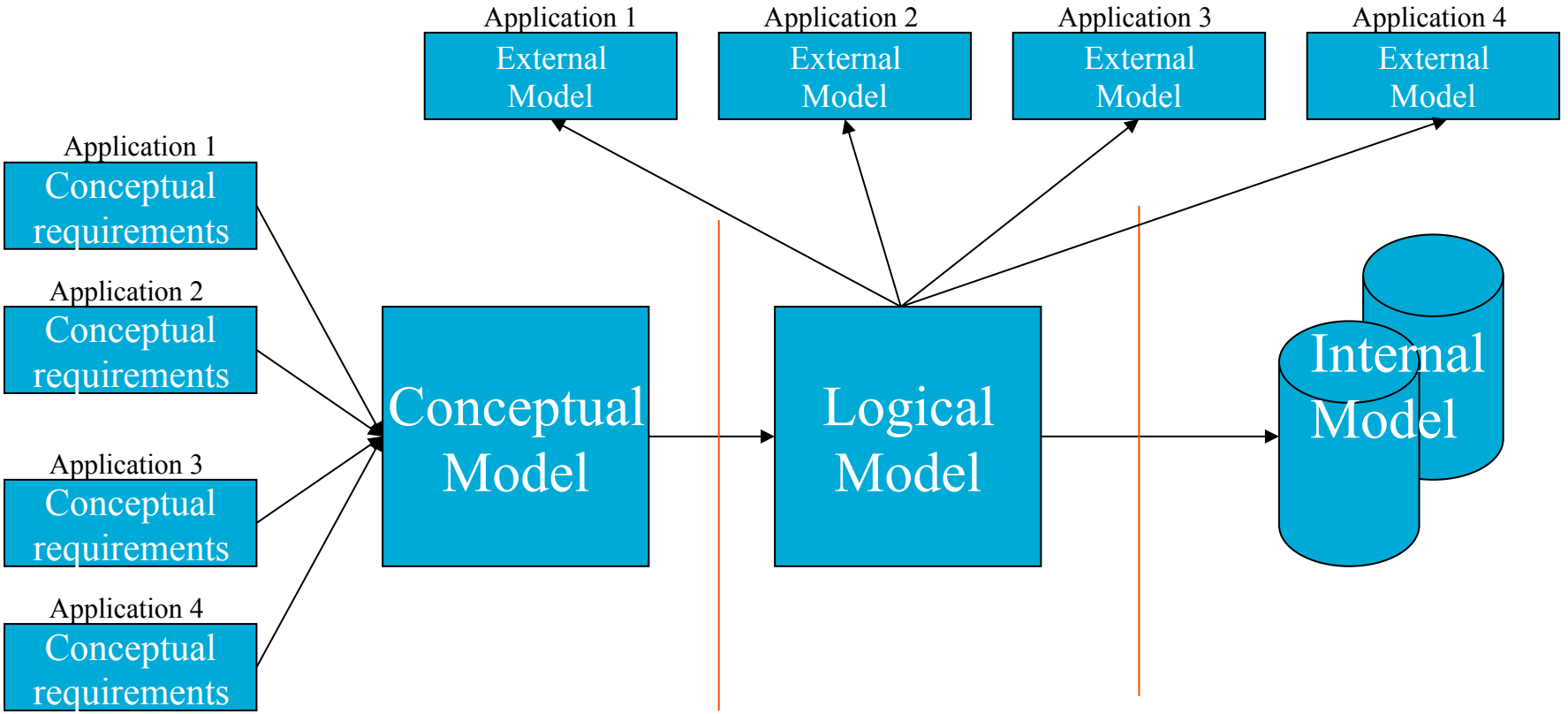
Terms and Concepts

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

Overview

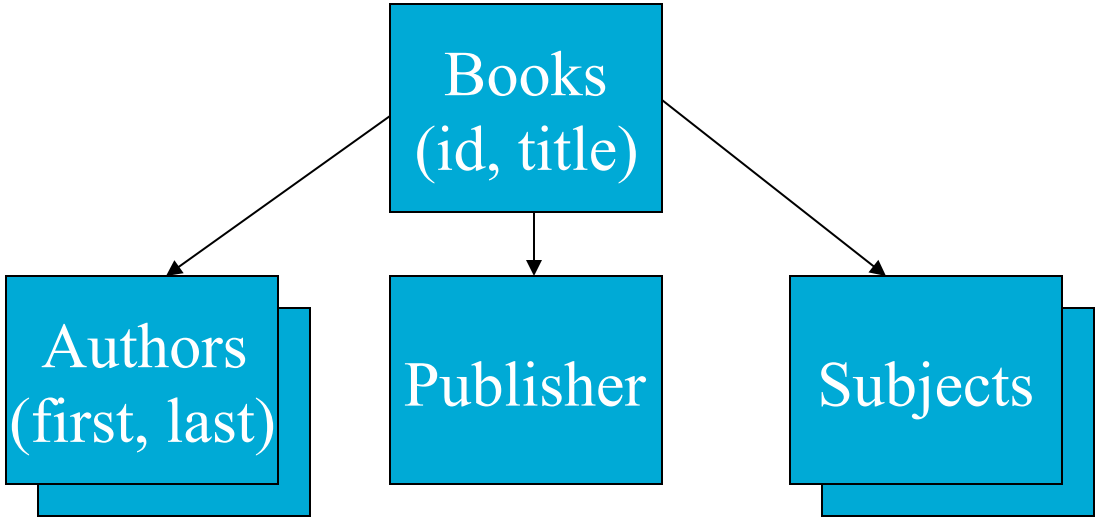
- Announcements
- Course Description
- Database Concepts and Terminology
- Database Models
- Database Life Cycle

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

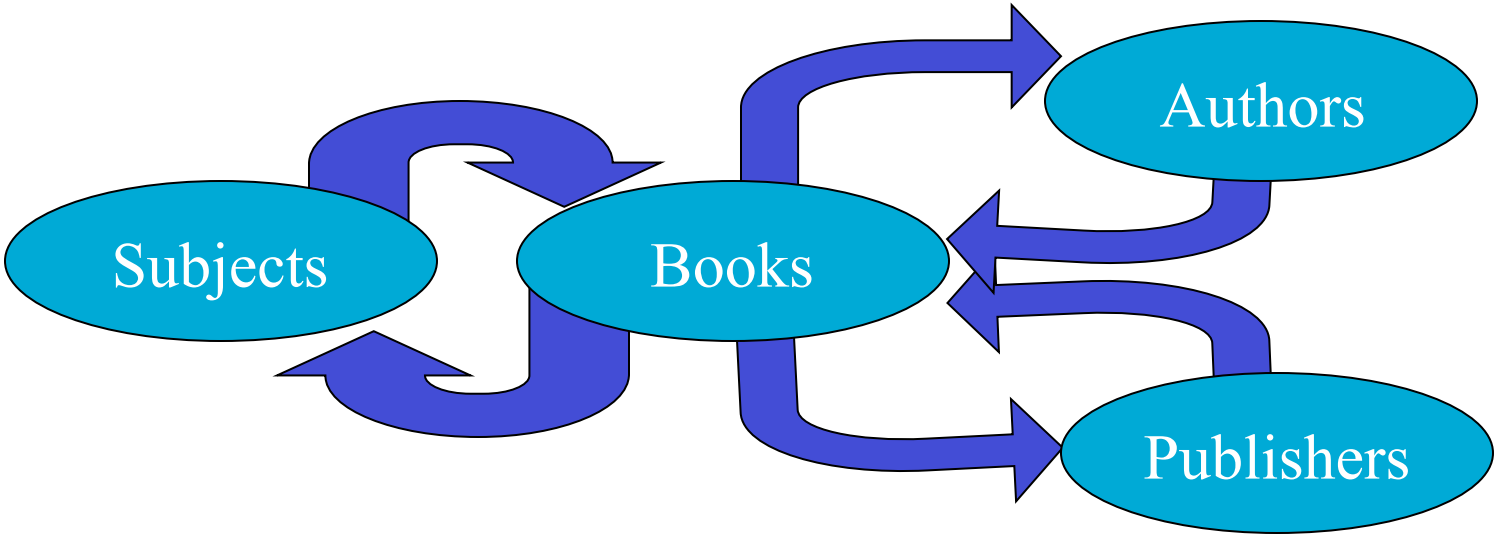
Data Models(2): History

- 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

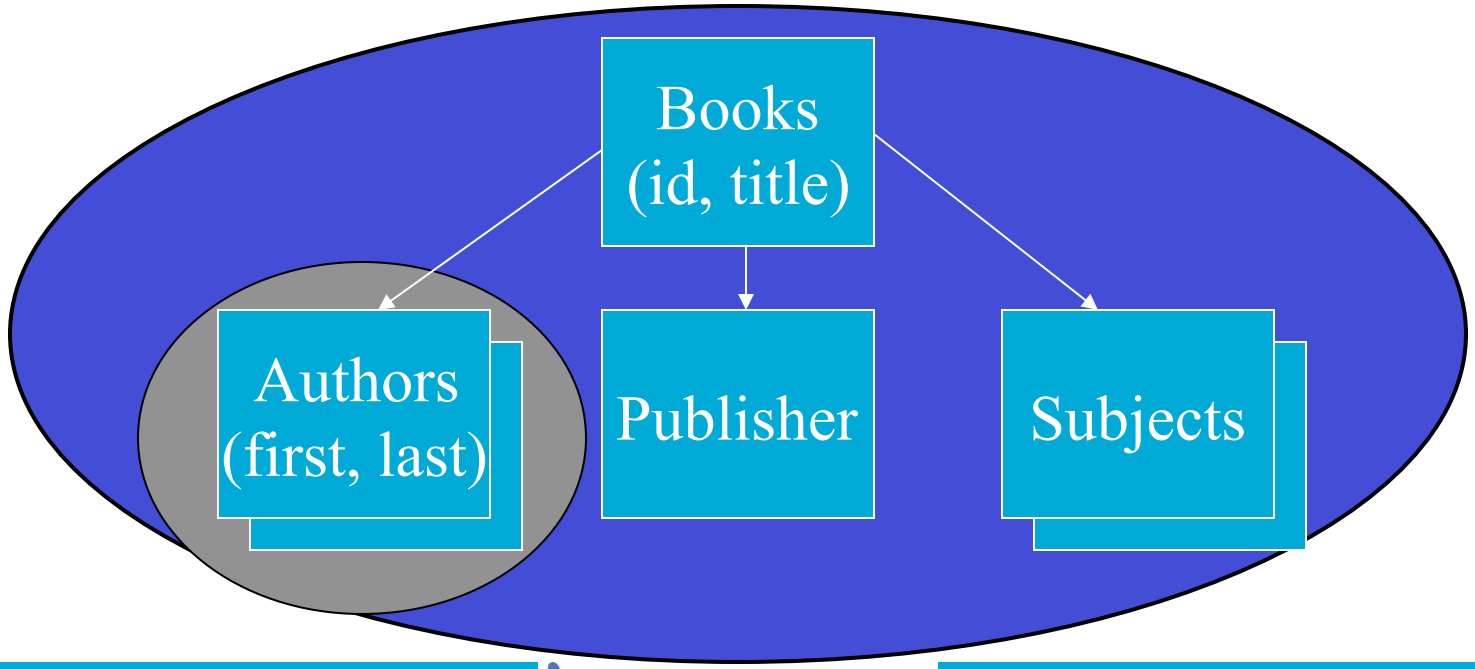
Authorid	Author nan
1	Smith
2	Wynar
3	Jones
4	Duncan
5	Applegate

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

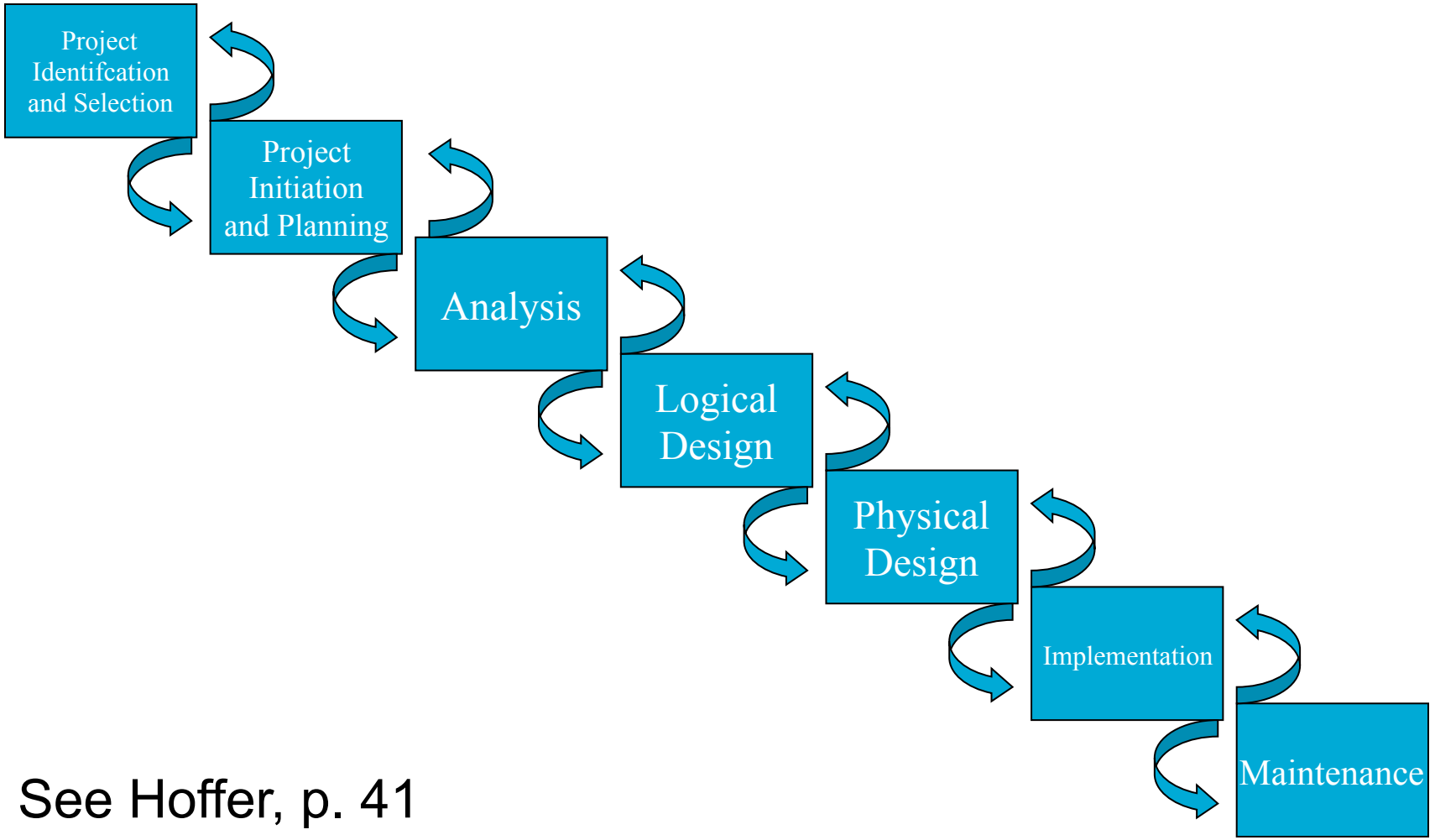
NoSQL Databases

- 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

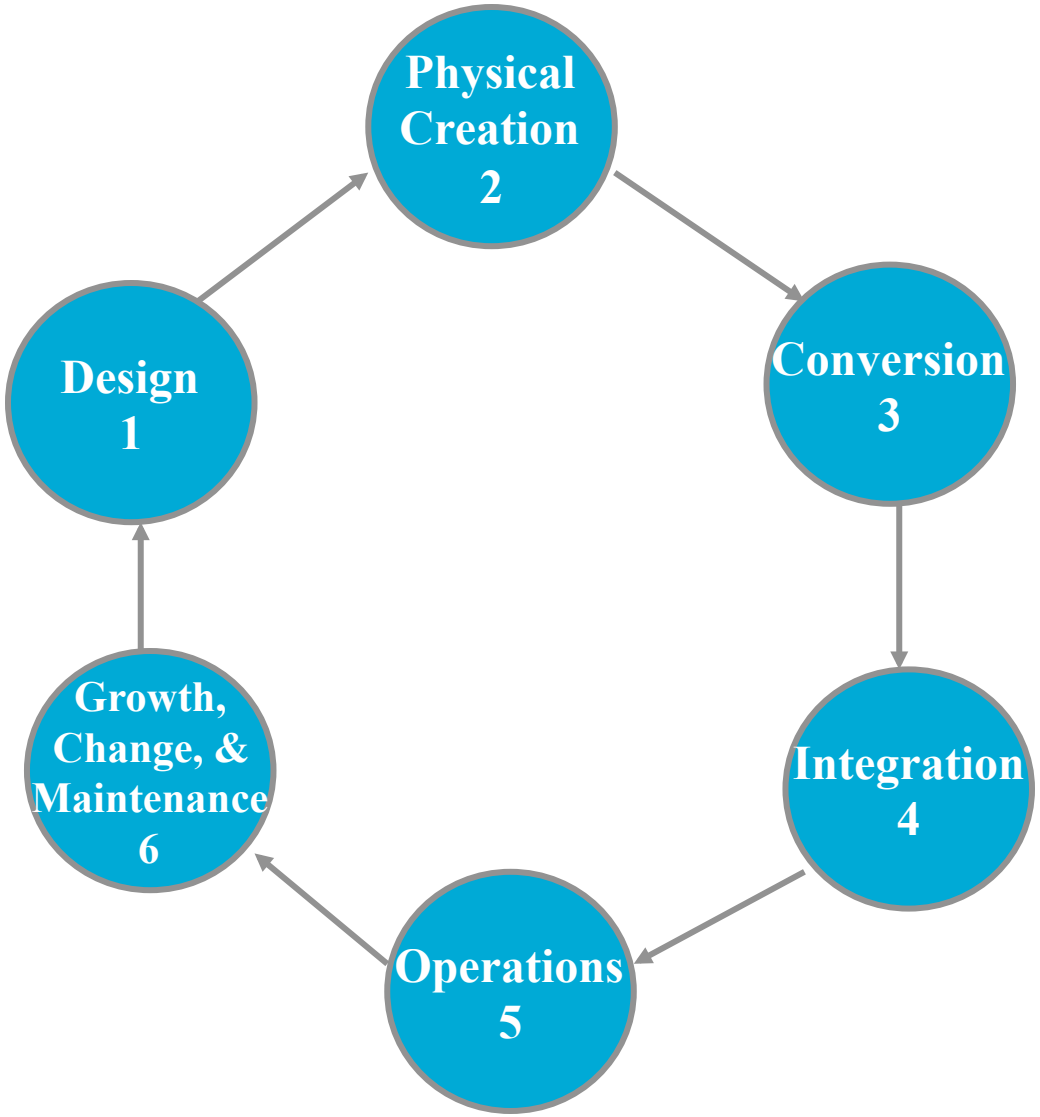
- Announcements
- Course Description
- Database Concepts and Terminology
- Database Models
- Database Life Cycle

The "Cascade" View



See Hoffer, p. 41

Database System Life Cycle



Design

- Determination of the needs of the organization
- Development of the Conceptual Model of the database
  - Typically using Entity-Relationship diagramming techniques
- Construction of a Data Dictionary
- Development of the Logical Model

Physical Creation

- Development of the Physical Model of the Database
  - data formats and types
  - determination of indexes, etc.
- Load a prototype database and test
- Determine and implement security, privacy and access controls
- Determine and implement integrity constraints

Conversion

- Convert existing data sets and applications to use the new database
  - May need programs, conversion utilities to convert old data to new formats.

Integration

- Overlaps with Phase 3
- Integration of converted applications and new applications into the new database

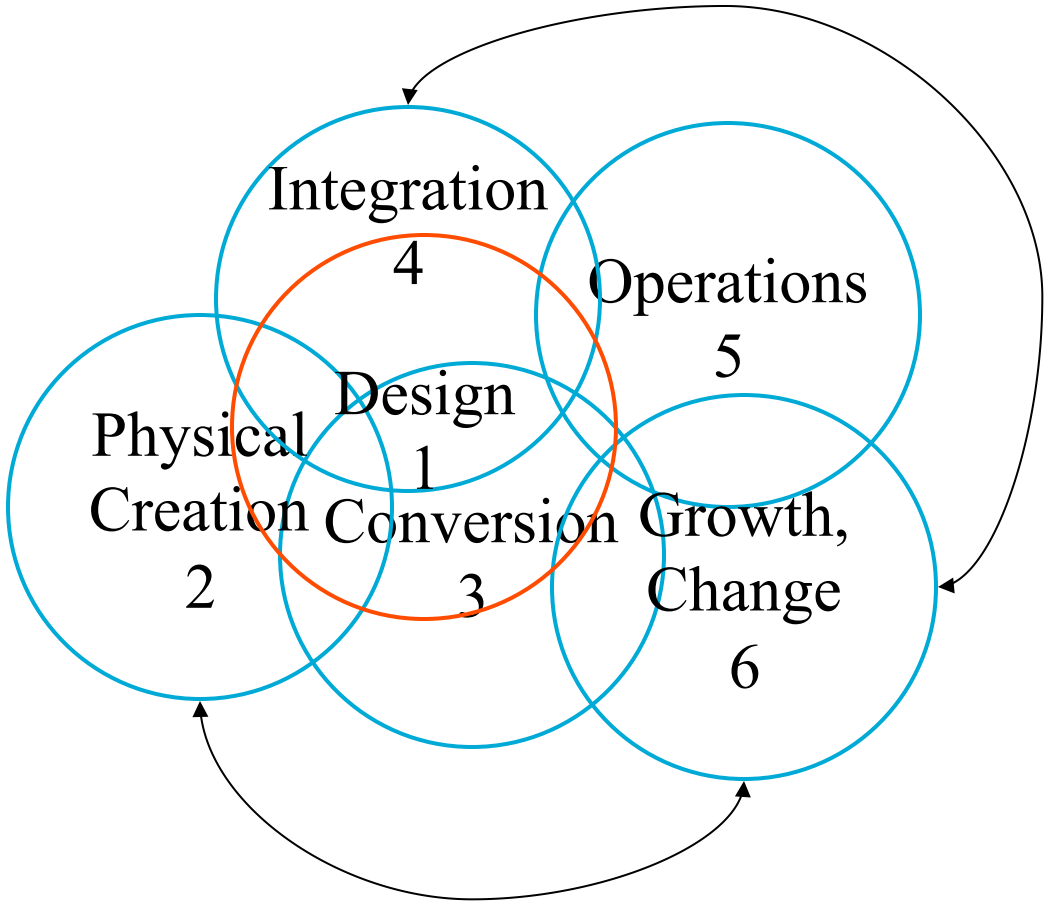
Operations

- All applications run full-scale
- Privacy, security, access control must be in place.
- Recovery and Backup procedures must be established and used

Growth, Change & Maintenance

- Change is a way of life
  - Applications, data requirements, reports, etc. will all change as new needs and requirements are found
  - The Database and applications and will need to be modified to meet the needs of changes

Another View of the Life Cycle



Next Time

- Introduction to the Diveshop database
- Introduction to Database Design