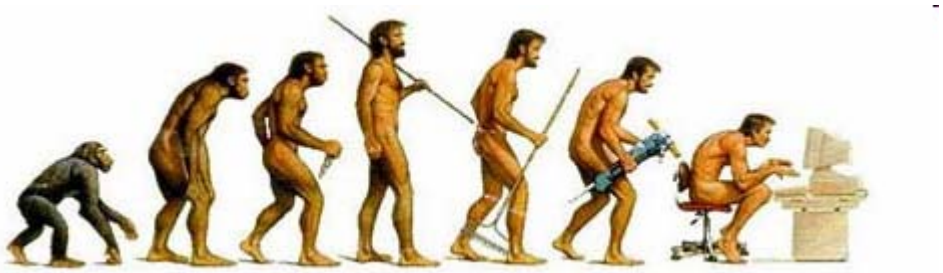
1. Database Fundamentals

Date: 07.10.2009

Instructor: SL. Dr. Ing. Ciprian Dobre

ciprian.dobre@cs.pub.ro

Welcome !



- "Let's face it, if you work in Oracle and haven't heard about Don Burleson*, you must either be very new or not interested in learning about your profession" – DBAZine

- *http://www.dba-oracle.com/resume_don.htm

Reguli de Notare

- Nota = MIN(10, NotaParcurs + NotaExamen)
- NotaParcurs (6p):
  - NotaCurs – prezenta si teste (neanuntate) (2p)
  - NotaActivitati – activitati stiintifice stabilite de comun acord cu titularul de curs, ce se finalizeaza cu un refererat si o prezentare in fata colectivului (2p)
  - Laborator (2p)
- NotaExamen (4p)

Course Topics

1. Database Fundamentals
   - Database, DBMS and Data Model
   - The Relational Data Model
   - Database Development and Planning
   - The Entity-Relationship Data Model
   - Relational Databases
2. Database Architecture
   - Logical and Physical Storage Structures
   - Application Architecture
   - Memory Architecture
   - Process Architecture
   - Database Resource Management
3. Database Structures
   - Database and Data Dictionary
   - Control and Redo Log Files
   - Tablespaces and Data Files
   - Segments and Storage Structures
   - Storage Structure and Relationships
   - Data Blocks
   - Extents
   - Segments
   - Managing Undo Segments

Course Topics

4. Managing Tables, Indexes, and Constraints
   - Managing Space for Schema Objects
   - Storing Data
   - Managing Indexes
   - Managing Constraints
   - Managing Views, Sequences, and Synonyms
5. Data Protection
   - Data Concurrency and Consistency
   - Data Integrity
   - Controlling Database Access
   - Privileges, Roles, and Security Policies
   - Auditing
   - Using Globalization Support
6. Backup and Recovery Fundamentals
   - Backup fundamentals
   - Disaster planning and recovery strategies
   - High availability overview
   - High availability architectures
   - Operational policies
   - Instance and Media Recovery Structures
   - Configuring the Database Archiving Mode

Laboratory topics

1. Installing and Managing Oracle
   - The Oracle Universal Installer
   - Oracle Enterprise Manager
   - Optimal Flexible Architecture
   - Oracle Managed Files (OMF)
2. The Oracle Instance
   - Administrator Authentication Methods
   - Connecting to an Oracle Instance
   - Database and Instance Startup and Shutdown
   - Managing Sessions
3. Creating a Database and Data Dictionary
   - Considerations Before Creating a Database
   - The Oracle Database Configuration Assistant
   - Creating a Database
   - Troubleshooting Database Creation
   - Using Data Dictionary and Dynamic Performance Views

Laboratory topics

4. Control and Redo Log Files
   - Maintaining the Control File
   - Maintaining Online Redo Log Files
   - Managing Archived Redo Logs
   - Using LogMiner to Analyze Redo Log Files
5. Managing Tablespaces and Data Files
6. Managing Undo Space
7. Managing Tables, Indexes, and Constraints
   - Managing Tables
   - Managing Indexes
   - Maintaining Data Integrity
8. Managing Users, Security, and Globalization Support
   - Managing Password Security and Resources
   - Managing Users
   - Managing Privileges
   - Managing Roles
   - Using Globalization Support

Laboratory topics

9. Oracle Net Server
   - Networking Overview
   - Oracle Net Architecture
   - Basic Oracle Net Server Side Configuration
   - Naming Method Configuration
   - Usage and Configuration of the Oracle Shared Server
10. Backup and Recovery
   - Oracle Recovery Manager Overview and Configuration
   - User-Managed Backups
   - RMAN Backups
   - User-Managed Complete Recovery
   - RMAN Complete Recovery
   - User-Managed Incomplete Recovery
   - RMAN Incomplete Recovery
   - RMAN Maintenance
   - Recovery Catalog Creation and Maintenance
11. The Database Administrator
   - Tasks of a Database Administrator
   - Managing Job Queues
   - Detecting and Repairing Data Block Corruption
   - Loading Data into a Database
   - Export and Import Utilities

Text Books

- Lecture notes will be made available online
- Recommended books
  - Kevin Loney, George Koch, "*Oracle9i*: *The Complete Reference*", Oracle Press, McGraw-Hill/Osborne
  - Oracle, "*Oracle Database. Concepts*".
  - OCP Study Books: Doug Stuns, Matthew Weishan, *"Oracle9i DBA Fundamentals I" & "Oracle9i DBA Fundamentals II*", SYBEX
- Research papers will be made available online

- Important: http://tahiti.oracle.com/

Overview

- Database, DBMS and Data Model
- Relational Databases
- Database Administration

1. Database, DBMS and Data Model

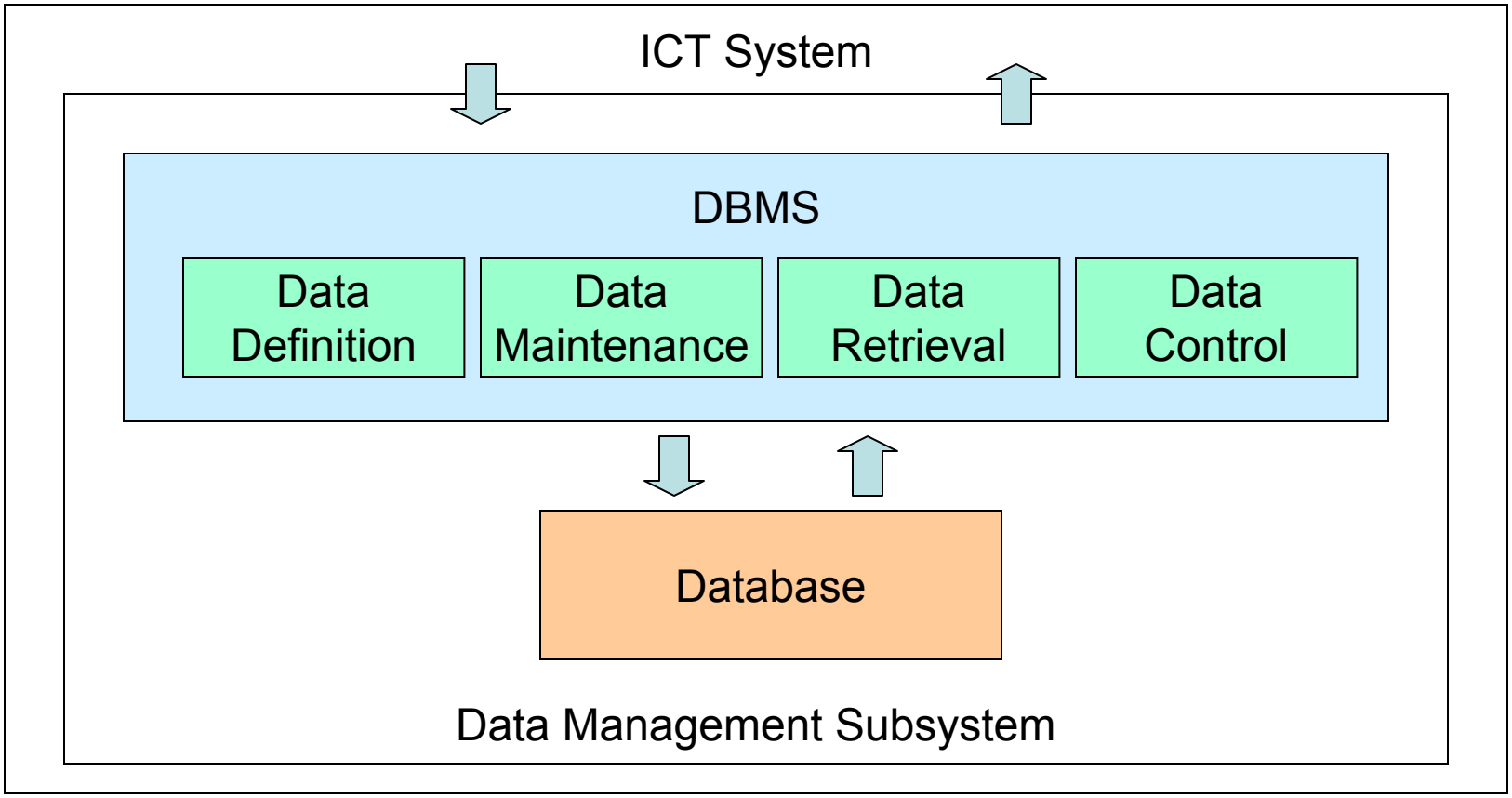
Key Concepts

- The database = shared collection of data
- The DBMS = shared collection of facilities
- Both the DBMS and database adhere to the tenets of some data model
- The DBMS is used to manage all interactions with the database by end-users or application programs
- The components of database, DBMS and data model all contribute to the definition of a database system
- Database development is the process of representing the data needed to support some organizational activity in some chosen DBMS

Properties of a database

- Data Sharing — More users
- Data Integration — Collection of data
- Data Integrity — Related objects should reflect in the database structure
- Data Security — Securing the database
- Data Abstraction — Abstraction of the real-world
- Data Independence — Independence between how the data is stored vs. what the processes need

Functions of a DBMS



Database System

- DBMS:
  - Define suitable data structures (the schema)
  - Define integrity constraints
  - Includes manipulative facilities
  - Enforce integrity
- Result = database
- The DBMS and database must adhere to a data model

Data models

- DBMS must adhere to the principles of *data model*
  - *General principles for handling data*
    - *Relational data model*
    - *Hierarchical data model*
    - *Object-oriented data model*

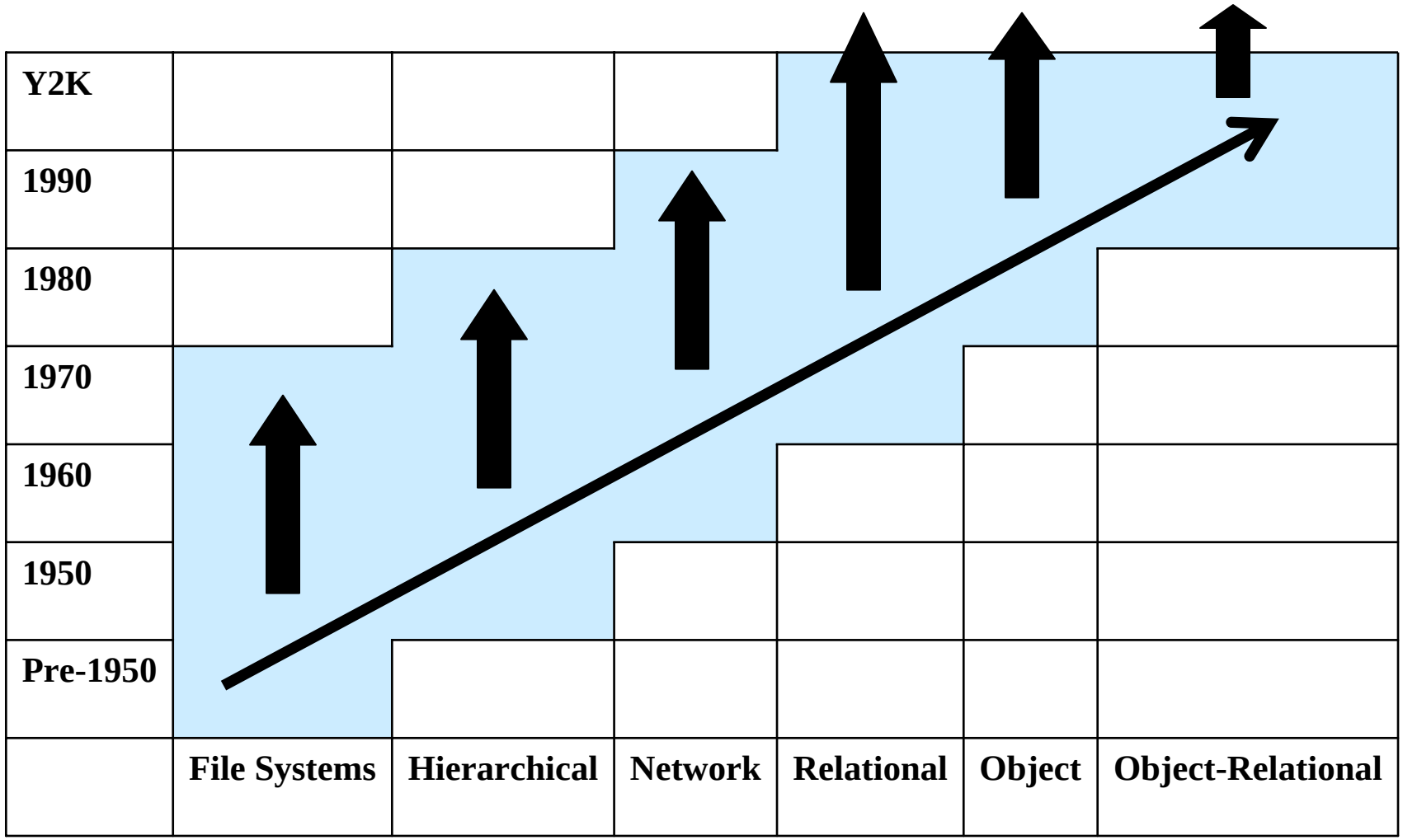
Data models

- *Set of principles* that define a data model:
  - Data definition
    - How data is structured
    - Set of templates
  - Data manipulation
    - How data is operated upon
    - Access and change over data
  - Data integrity
    - Which states are valid for a database
    - Valid and invalid changes of data

Generations of architectural data models

- Primitive data models
  - Record structures grouped in file-structures
- Classic data models
  - Hierarchical → network → relational data models
- Semantic data models
  - SDMs provide a more expressive means of representing the meaning of data
  - Post-relational and object-oriented data models

The Evolution of Database Modeling Techniques



A walk down the history of data management…

The Paleolithic Period …

- No general purpose tools for managing large volumes of data…
  - OS provided resource management
  - Data was stored in files
  - Applications performed data management functionalities
    - Fault-tolerance
    - Concurrency control
    - Reliability
    - Optimizations
    - …
  - Such functionalities had to be re-implemented for each application

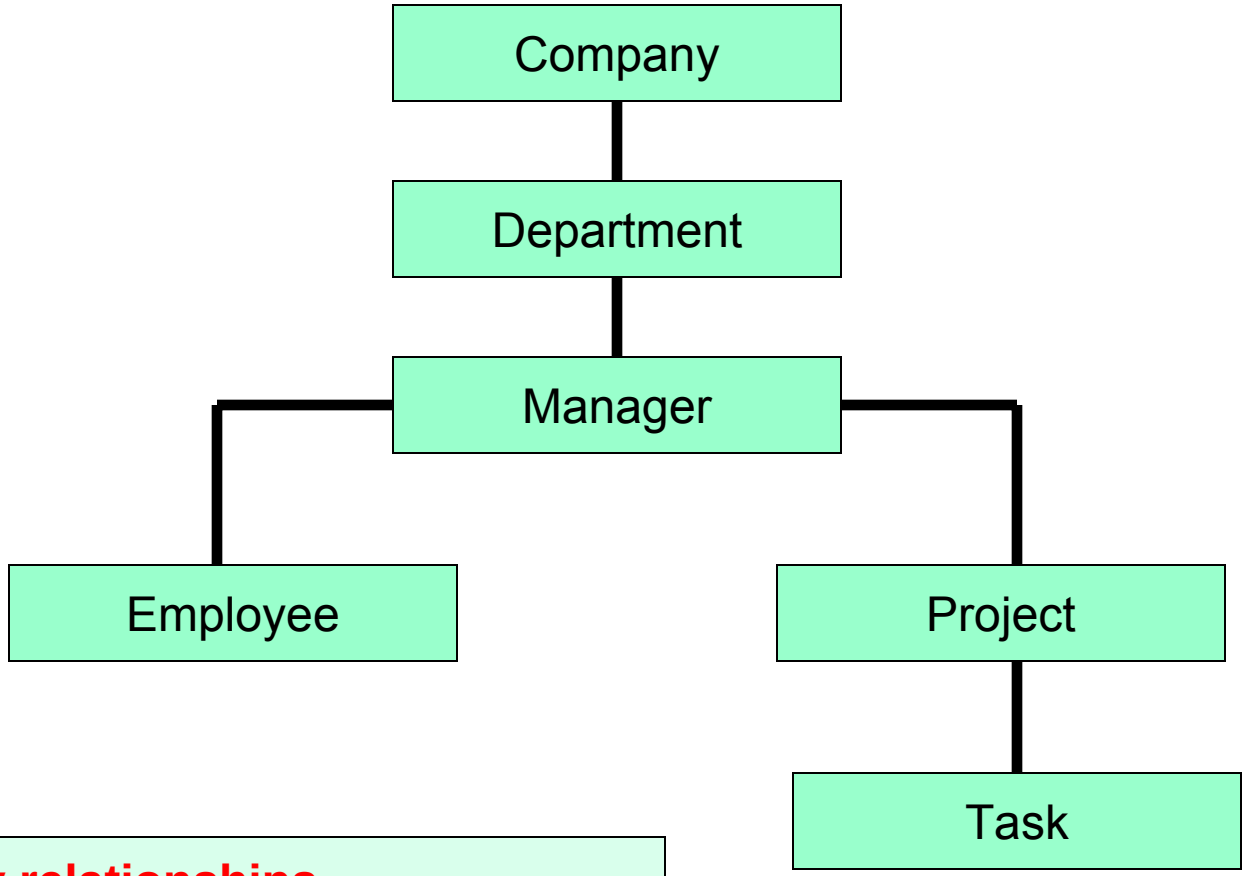
The Neolithic Period…

- Early file systems evolve into general-purpose data management tools.
- DBMS Goals:
  - Efficiency and scalability (faster than files)
  - Management of large heterogeneous types of structured data
  - High reliability
  - Information sharing (multiple users)
- DBMS Users:
  - E-commerce companies, banks, airlines, transportation companies, corporate databases, government agencies, …
  - Anyone you can think of!

The Dark Ages ….

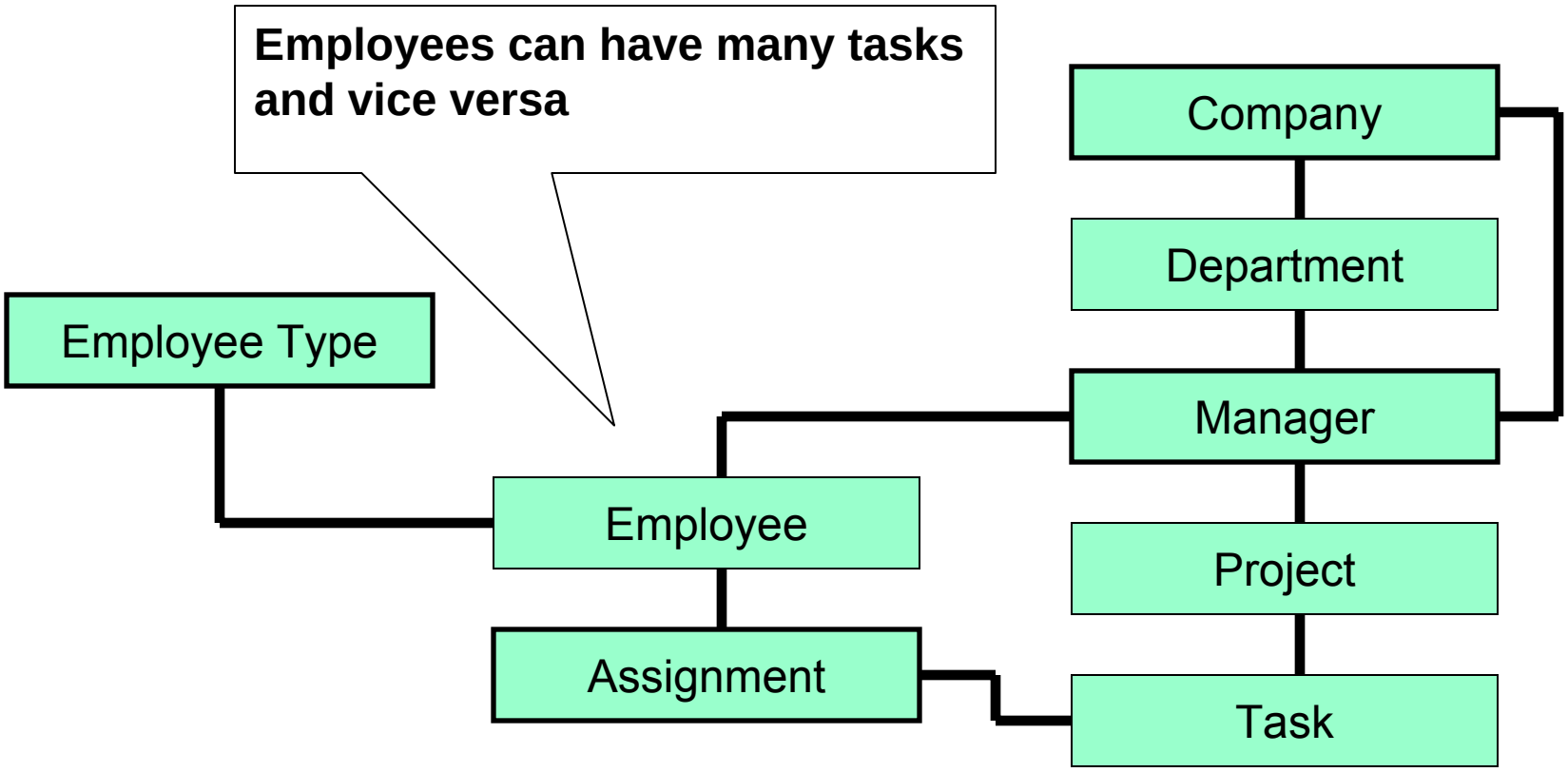
- Network & hierarchical data models
  - Resulted in data spaghetti
  - Applications needed to chase pointers
  - There was little data abstraction or separation of concerns
    - little difference between physical data representation and logical data representation
  - Optimization was entirely left to application writers
  - There were no clean data management languages
    - Cobol anyone?

Hierarchical Database Model



- **One-to-many relationships**
- **Every access must originate at the root**

Network Database Model



- Child tables can have more parents
- Many-to-many relationships

The Relational Era..

- Relational model proposed by Codd (Codd, 1970)
  - Everything is a relation
  - Query consists of algebraic composition of a few powerful operators
  - Equivalent to a first-order relational calculus

- Primary features
  - Simple clean data representation
    - solid mathematical basis
  - Data abstraction
    - Users did not need to be concerned about how data is stored physically
  - Simple declarative query language
    - User's specify what to compute not how to do it.
  - Optimization by the system

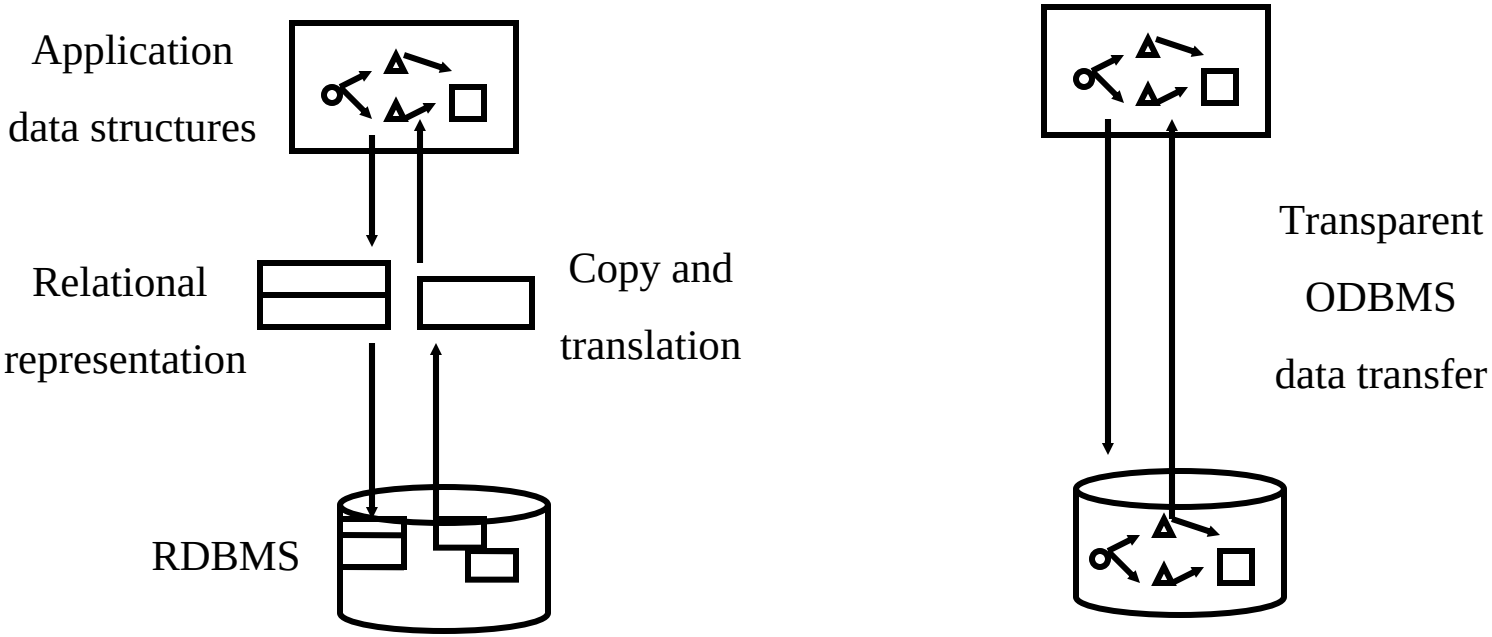
Data Wars (1)

- Codasyl versus relational debates began…
  - **Codasyl:** relational model is too simple, applications built using it will never scale in performance.
  - **Relational:** network/hierarchical models have no formal basis, are too complex, and unmanageable as application complexity increases.
- Relational model found many supporters
  - Specially at universities
  - Its simplicity was enticing

Data Wars (2)

- Many projects started off trying to implement a relational DBMS
  - System R @ IBM Almaden
  - Ingres @ Berkeley
  - These early systems led to the technologies that drive modern data management
- Early prototypes became products
  - DB2 & Ingres
- Principle designers from both the System R teams & Ingres left to start companies
  - Oracle, Sybase
- Early relational companies went door to door converting industry to the relational model
  - Industry got hooked on to the simplicity of writing complex applications in relational model
  - Boeing among the first converts

Pointer's Strike Back...



- Complex objects in emerging DBMS applications cannot be effectively represented as records in relational model.
- Representing information in RDBMSs requires complex and inefficient conversion into and from the relational model to the application programming language
- ODBMSs provide a direct representation of objects to DBMSs overcoming the *impedance mismatch* problem

Object Model

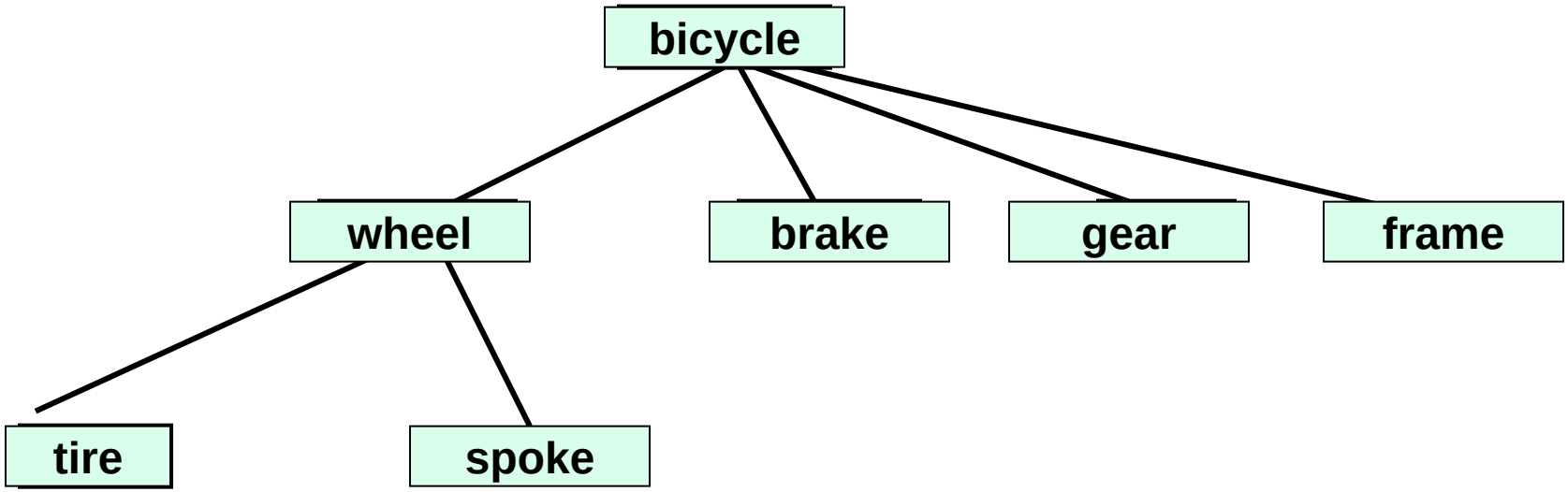
- Object:
  - observable entity in the world being modeled
  - similar to concept to entity in the E/R model
- An object consists of:
  - **attributes**: properties built in from primitive types
  - **relationships**: properties whose type is a reference to some other object or a collection of references
  - **methods**: functions that may be applied to the object.

http://en.wikipedia.org/wiki/Comparison_of_object_database_management_systems

Object Identity

- Each object has an identity which it maintains even if some or all of its attributes change.
- Object identity is a stronger notion of identity than in relational DBMSs.
- Identity in relational DBMSs is value based (primary key).
- Identity in ODBMSs built into data model
  - no user specified identifier is required
- OID is a similar notion as pointer in programming language
- Object identifier (OID) can be stored as attribute in object to refer to another object.
- References to other objects via their OIDs can result in a *containment hierarchy*
- *Note:* containment hierarchy different from class hierarchy

Containment Hierarchy



Links in containment hierarchy should be read as **is-part-of** instead of **is-a**

Object Database Management Group (ODMG)

- Special interest group to develop standards that allow ODBMS customers to write portable applications
- Standards include ("The Object Data Standard ODMG 3.0", 2001):
  - Object Model
  - Object Specification Languages
    - Object Definition Language (ODL) for schema definition
    - Object Interchange Format (OIF) to exchange objects between databases
  - Object Query Language
    - declarative language to query and update database objects
  - Language Bindings (C++, Java, Smalltalk)
    - Object manipulation language
    - Mechanisms to invoke OQL from language
    - Procedures for operation on databases and transactions
- **Next Generation Standardization Work by OMG**
  - Currently, the Object Management Group (OMG), as distinguished from ODMG, plans a "4th, next generation" object database standard to reflect changes in object database technology.

Disadvantages of ODBMS Approach

- *Low protection*
  - since persistent objects manipulated from applications directly, more changes that errors in applications can violate data integrity.
- *Non-declarative interface:*
  - difficult to optimize queries
  - difficult to express queries
- But …..
  - Most ODBMSs offer a declarative query language OQL to overcome the problem.
  - OQL is very similar to SQL and can be optimized effectively.
  - OQL can be invoked from inside ODBMS programming language.
  - Objects can be manipulated both within OQL and programming language without explicitly transferring values between the two languages.
  - OQL embedding maintains simplicity of ODBMS programming language interface and yet provides declarative access.

The Return of the Relations … POSTGRES

- Relational model evolved into ORDBMSs that include “best of” object-oriented concepts
- The very first ORDBMS prototype built @ Berkeley

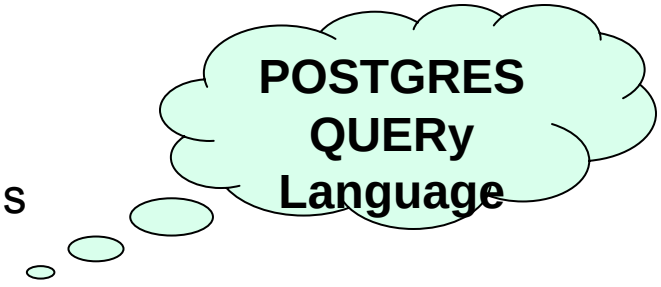
POSTGRES —commercialized→ Illustra —bought by→ Informix →
Informix Universal Server (IUS) product line

→ PostgreSQL (released under a BSD-style license)

- Has had major impact on major commercial DBMS which have all migrated to ORDBMS model.
- SQL3 supported by modern databases adapted many of the concepts developed in Postgres
  - » http://www.objs.com/x3h7/sql3.htm

POSTGRES — Combinations

- Introduced object orientation into relation DBMSs.
- Fundamental Concepts.
  - Each record has an OID
  - Access to data though:
    - query language POSTQUEL
    - navigation through OIDs
  - Classes
  - Inheritance
  - Types: rich set of types available for columns
  - Functions: can be called within POSTQUEL



Classes And Inheritance

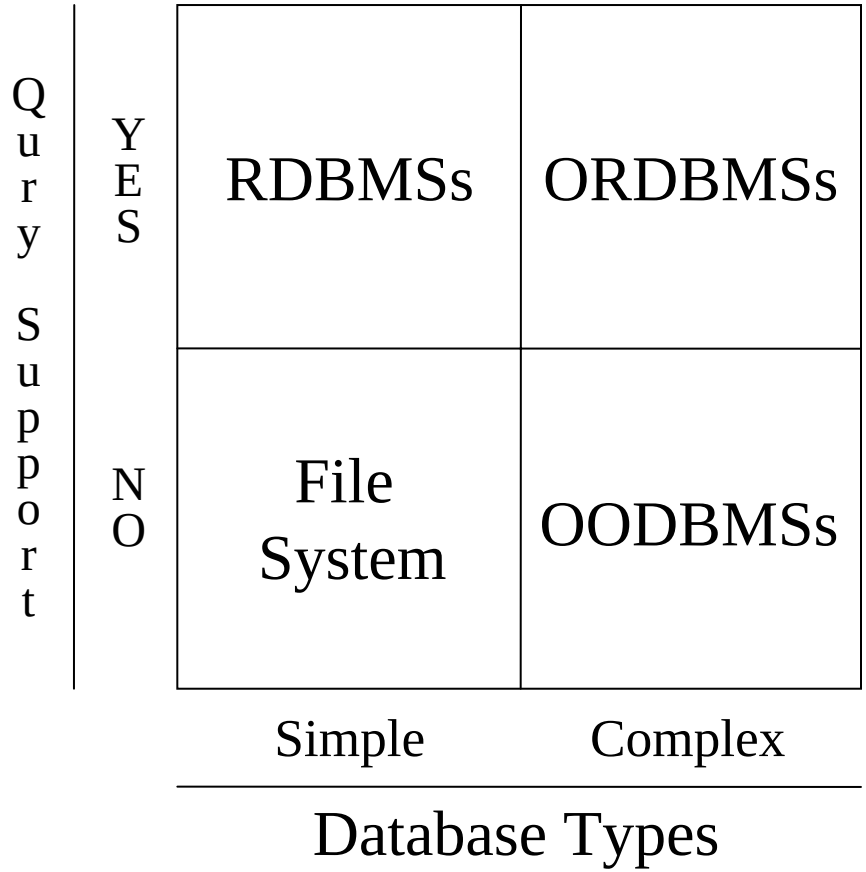
- Class analogous to relation
- User can create new class

  create Emp (name = c12, salary = float, age = int)
- Classes can inherit from others

  create Salesman (quota = float) inherits Emp
- Multiple inheritance permitted. If new class causes ambiguity it is not created.
- Classes:
  - real: base classes or relations
  - derived: views
  - version: maintained differentially compared to parent class

See “Data mining si data warehousing” or “Modelarea datelor”…

Database Technology Matrix

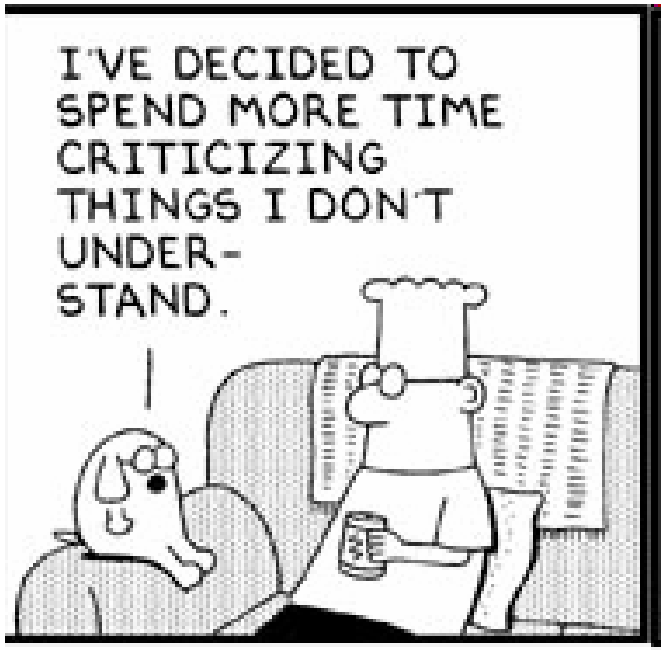


XML the new revolution?

- Just when relational model had driven out object-oriented database technology, WWW led to the proliferation of semi-structured data.

- 2 approaches to supporting XML
  - Extend relational technology to support XML
  - Native XML databases

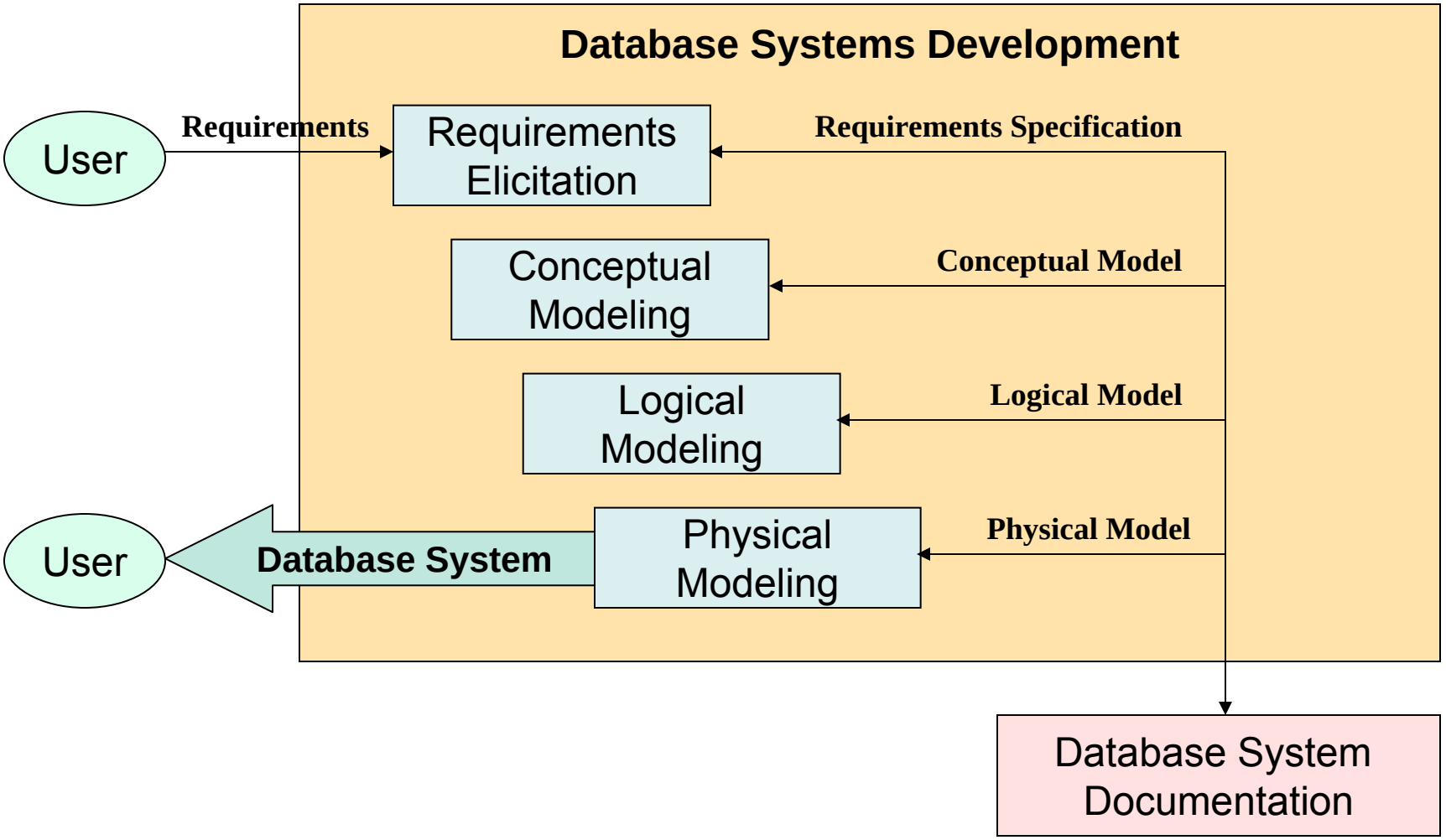
And we're back…



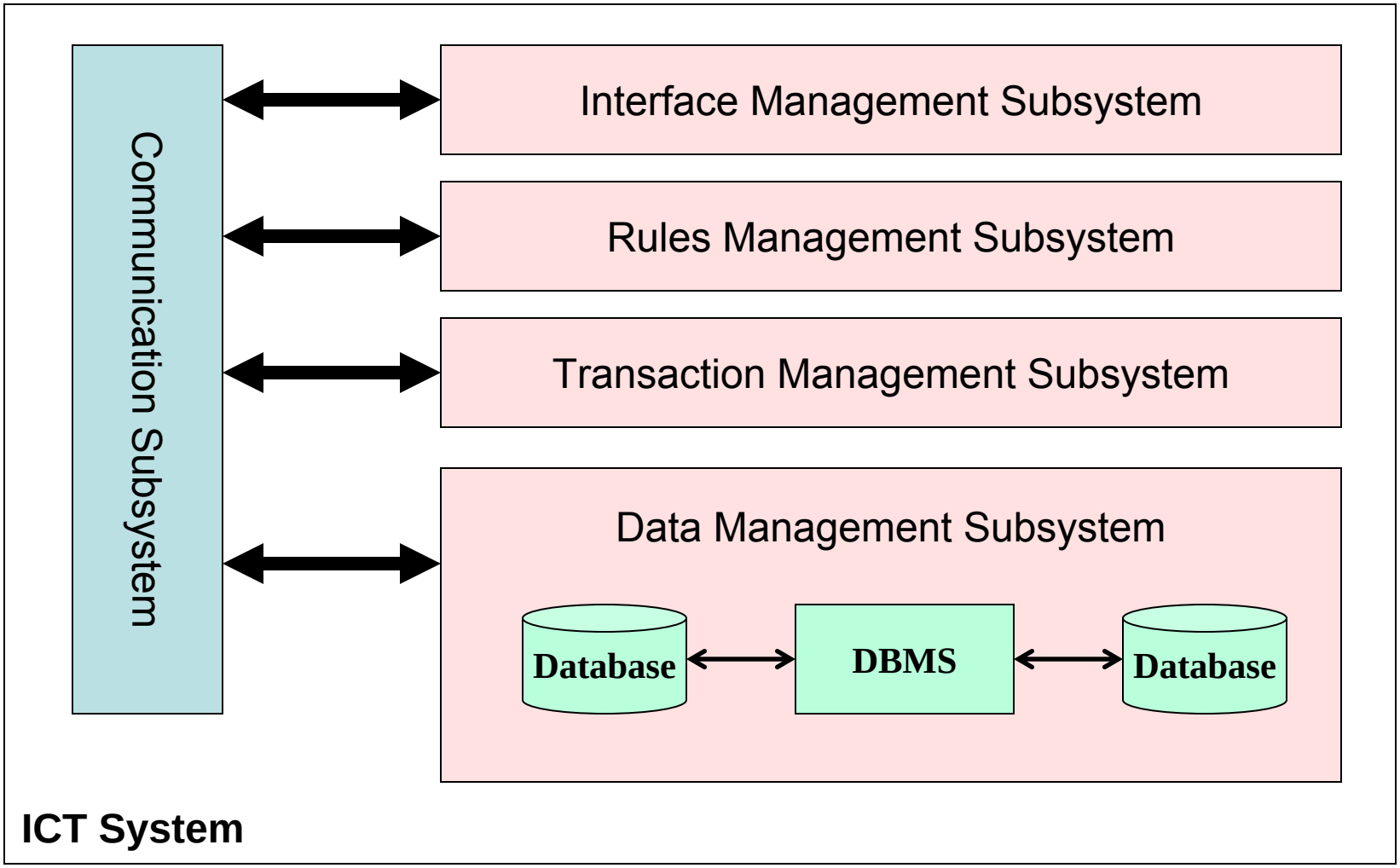
Database Development

- Conceptual modeling
  - The real-world is expressed in terms of data requirements
- Logical modeling
  - The real-world is expressed in terms of the principle of the data model
- Physical modeling
  - The real-world is expressed in terms of the constructs of the DMBS (tables, access structures such as indexes)

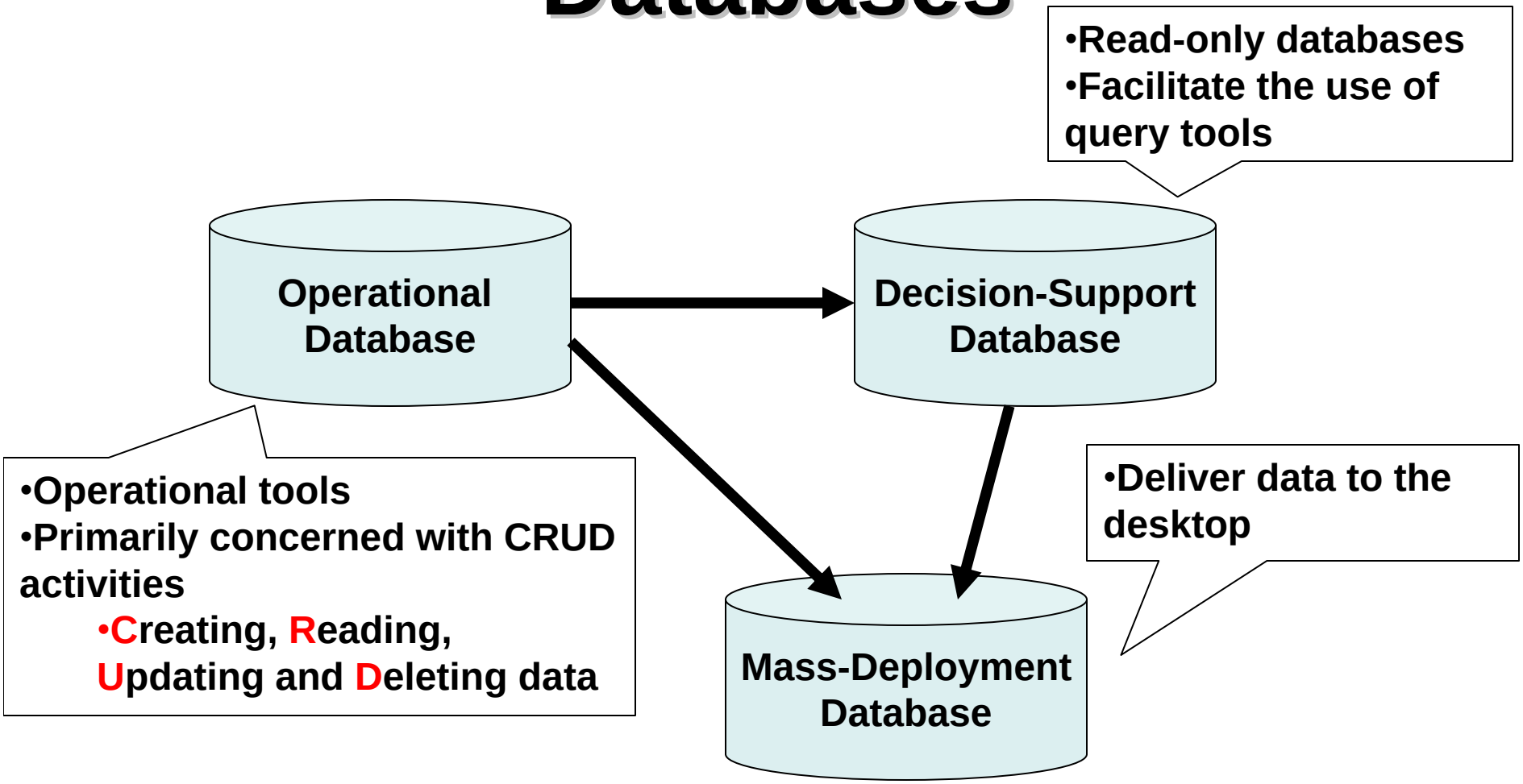
Database Systems Development



Layers of an ICT system



Contemporary Types of Databases



Data management layer

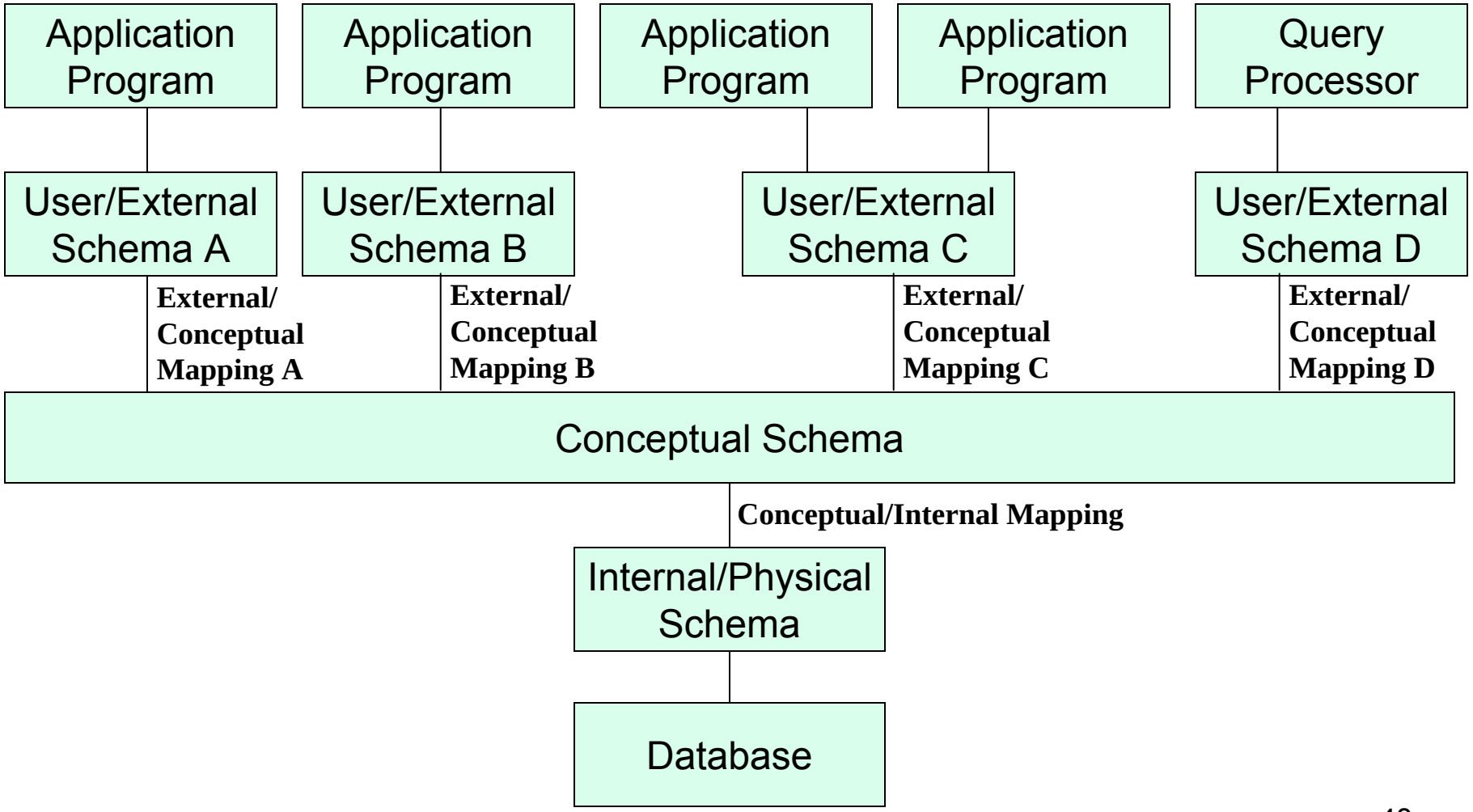
- Facilities provided by DBMS
  - Buffer between applications, end-users and a database
- ANSI-SPARC*, 1975
  - Proposed a three-layer architecture for the data management
    - Three schemas/level of abstraction/views

* American National Standards Institute Standard Planning and Requirements Committee

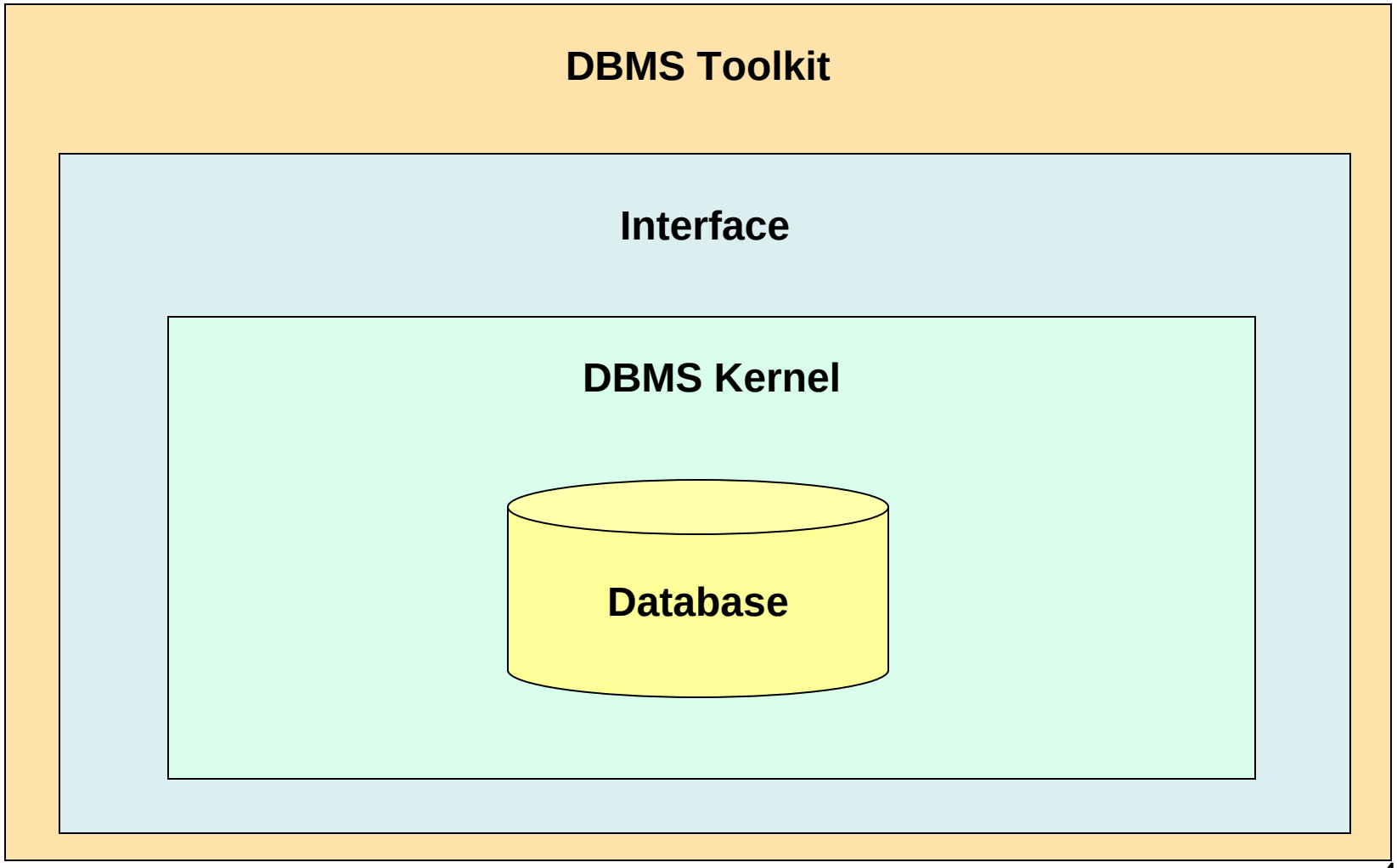
ANSI/SPARC Architecture

- The external or user level
  - How the user or application view the database
  - Several programs or users can share the same view
- The conceptual level
  - The organization view of the data
  - Relationship between the database and applicable constraints
  - Logical view of the database
- The internal or physical level
  - How data is physically stored

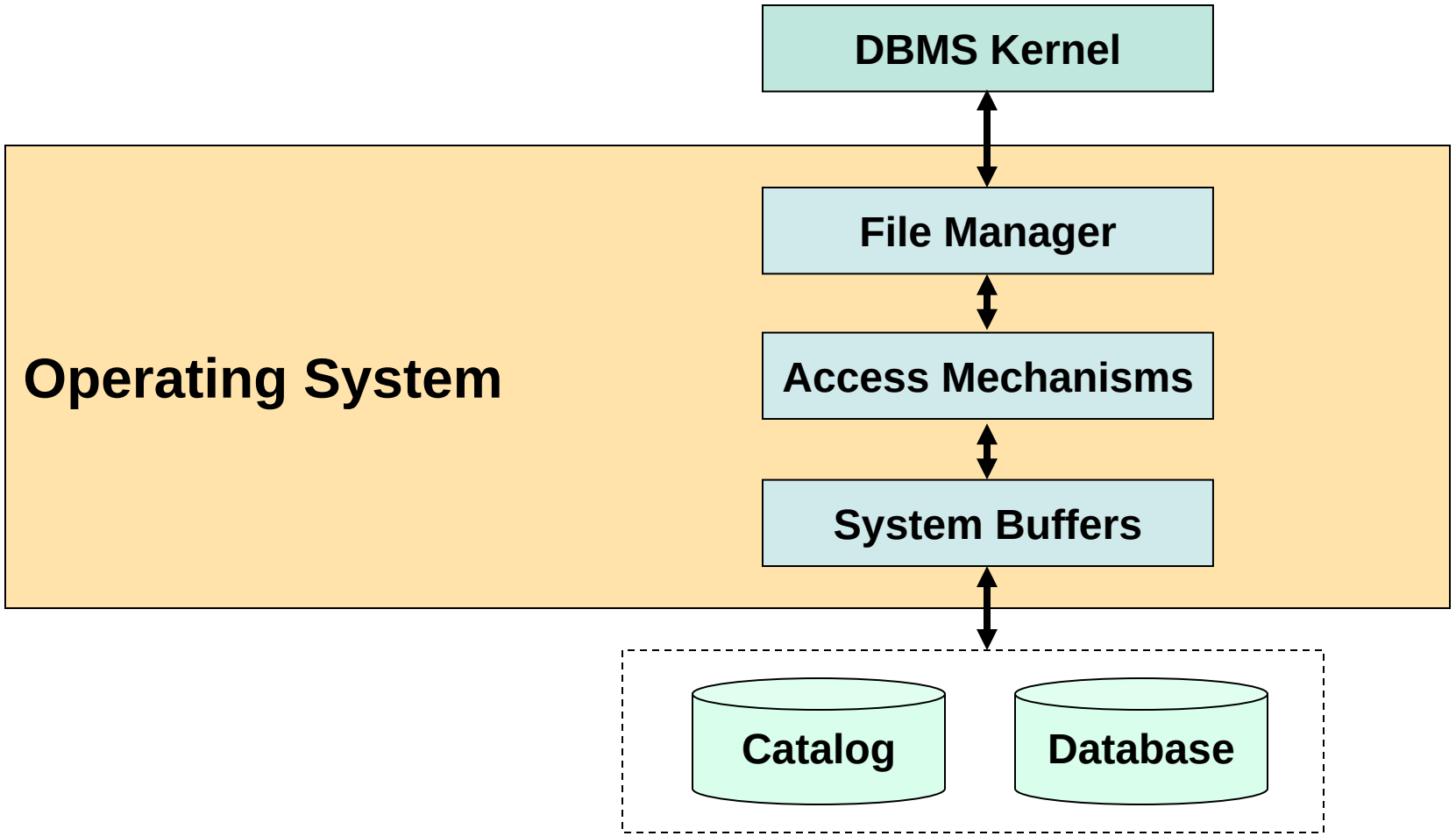
ANSI/SPARC Architecture



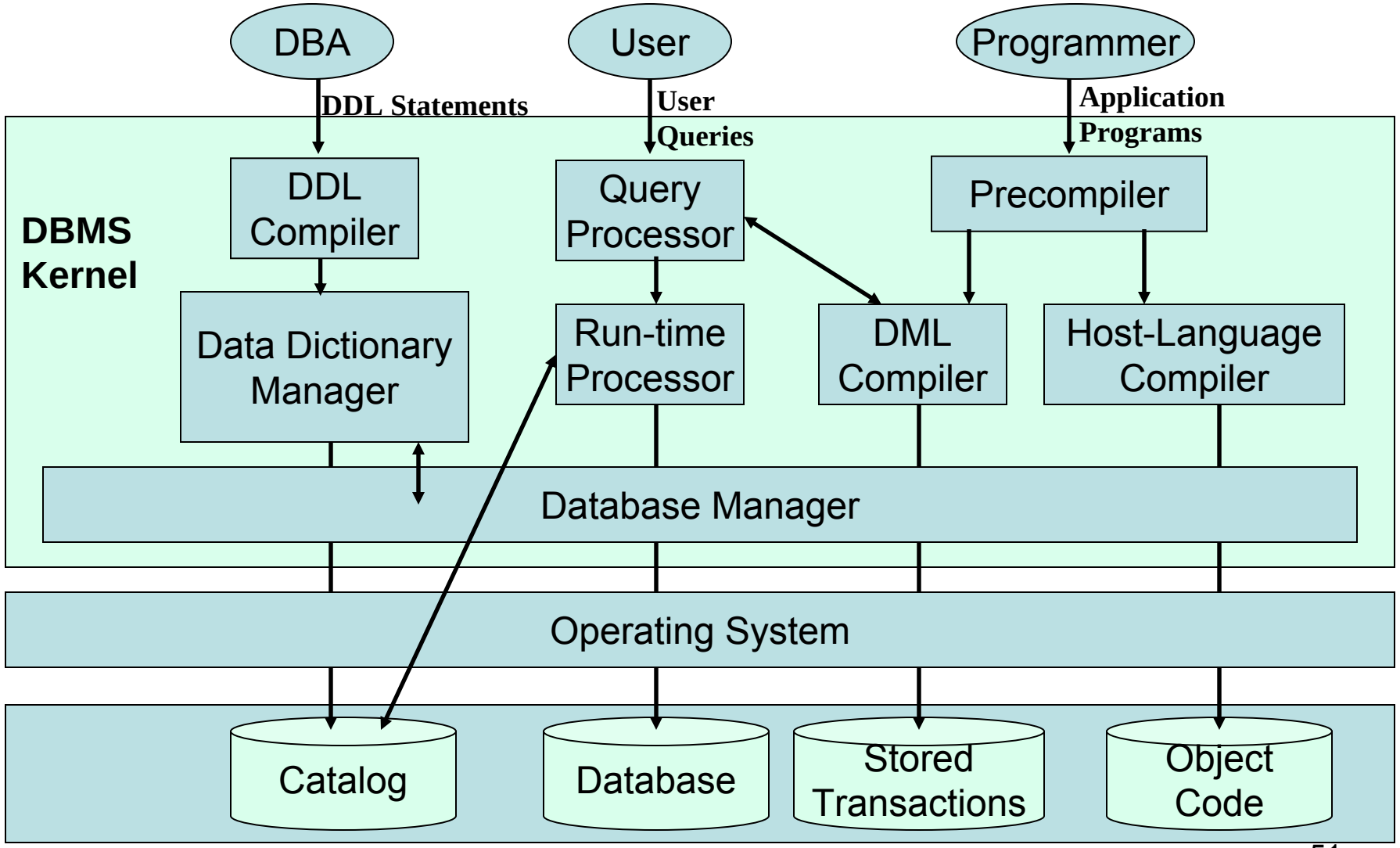
DBMS



Interaction between DBMS and operating system



DBMS Kernel

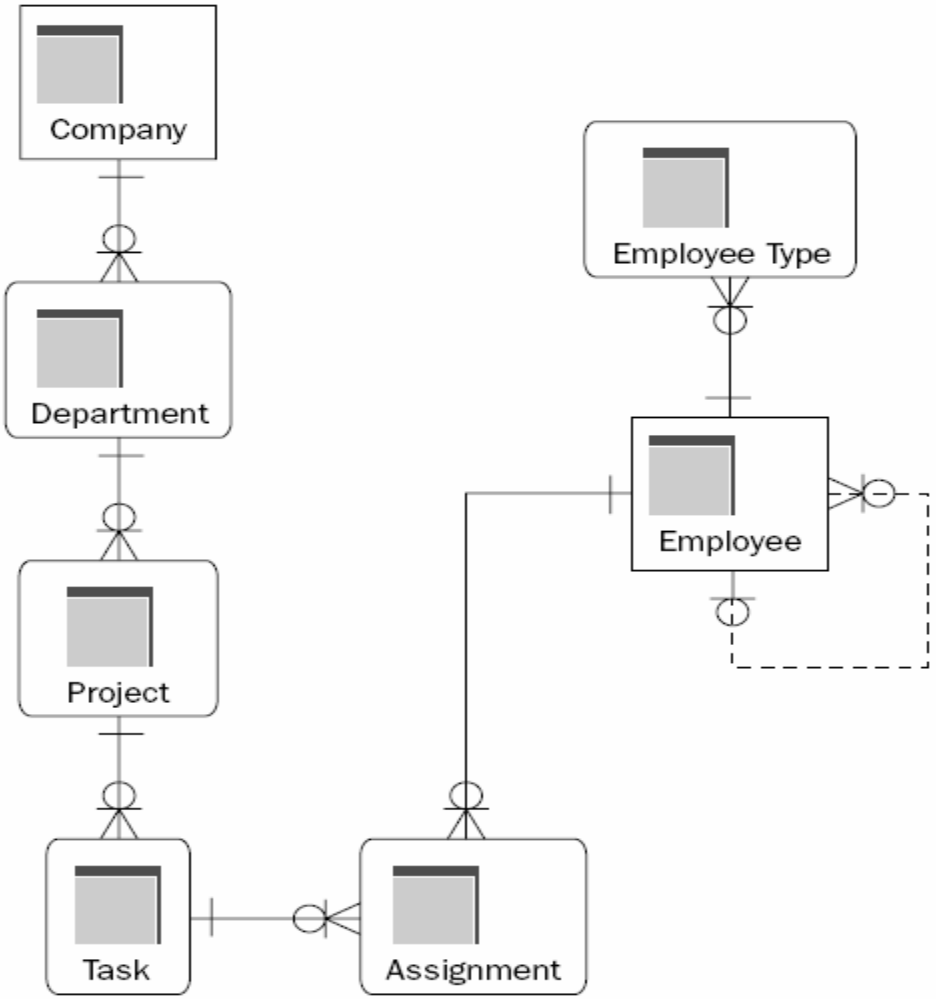


Functions of a DBMS

- CRUD functions
- Data dictionary
- Transaction management
- Concurrency control
- Recovery
- Authorization
- Data communication
- Data integrity
- Administration utilities

2. Relational Databases

The Relational Database Model



Original of the Relational Model

- Developed by Dr. E. F. Codd in 1969
- Relies on mathematical relations
- Relations are defined that store data
- Most databases today are based on the Relational Model
- Operators are used to produce and manage relations
- Constraints are used for integrity enforcement
- Utilizes a standard access language (SQL)
- Tools are provided to manage the Relational Database

Relational Database Model – a picture of the data

Project

PROJECT_ID	DEPARTMENT_ID	PROJECT	COMPLETION	BUDGET
1	1	Software sales data mart	4-Apr-05	35,000
2	1	Software development costing application	24-Apr-05	50,000
3	2	Easy Street construction project	15-Dec-08	25,000,000
4	1	Company data warehouse	31-Dec-06	250,000

Task

TASK_ID	PROJECT_ID	TASK
1	1	Acquire data from outside vendors
2	1	Build transformation code
3	1	Test all ETL process
4	2	Assess vendor costing applications
5	3	Hire an architect
6	3	Hire an engineer
7	3	Buy lots of bricks
8	3	Buy lots of concrete
9	3	Find someone to do this because we don't know how

3. Database Administration

Database Administration

- ***Database administration***
  - The function of managing and maintaining database management systems (DBMS) software.

- DBMS software comes with tools to help DBAs manage the DBMS
  - e. g. Microsoft SQL Server Enterprise Manager, Oracle SQL*Plus and Oracle Enterprise Manager

Database Administration

- Three types of DBAs:
  - System DBA
    - ≈ Physical DBA, Operations DBA, Production Support DBA
    - Focus on physical aspects:
      - DBMS installation
      - Configuration
      - Patching
      - Upgrades
      - Backups
      - Restores
      - Refreshes
      - Performance optimization
      - Maintenance and disaster recovery

Database Administration

- Development DBA
  - Focus on
    - data model design and maintenance
    - DDL (data definition language) generation
    - SQL writing and tuning
    - Coding stored procedures
    - Collaborating with developers to help choose the most appropriate DBMS feature/functionality
- Application DBA
  - 3rd party applications

Database Administrator

- The **database administrator** (DBA)
  Responsible for:
  - the technical implementation of database systems,
  - managing the database systems currently in use,
  - setting and enforcing policies for their use.
- The DBA has as core responsibilities:
  - administration of the database,
  - administration of the DBMS,
  - administration of the database environment.

Administration of the database

- Physical design
  - Data administrator (logical design) vs. database administrator (physical design)
- Data standards and documentation
- Monitoring data usage and tuning database structures
- Data archiving
- Data backup and recovery

Administration of the DBMS

- Installation
- Configuration control
  - Enforcing policies and procedures for managing updates and changes to the software
- Monitoring DBMS usage and tuning DBMS

Administration of the database environment

- Data control
  - Establishing user groups
  - Assigning passwords
  - Granting access to DBMS facilities
  - Granting access to databases
- Impact assessment
  - Assessing the impact of any data changes
- Privacy, security and integrity
  - Strategies for data integrity, security and privacy
- Training
  - Education of users regarding policies and principles of database use

Next time…