

Database Management Systems

Chapter 1: Database Environment



UNIVERSITY OF
MARYLAND

ROBERT H. SMITH
SCHOOL OF BUSINESS

DR. ADAM LEE

Objectives

- Define terms.
- Name limitations of conventional file processing.
- Explain advantages of databases.
- Identify costs and risks of databases.
- List components of database environment.
- Identify categories of database applications.
- Describe database system development life cycle.
- Explain prototyping and agile development approaches.
- Explain roles of individuals.
- Explain the three-schema architecture for databases.



Definitions

- **Database** - organized collection of logically related data
- **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



UNIVERSITY OF
MARYLAND

Figure 1-1: Data in Context

- Context helps users understand data.

Class Roster			
Course:	MGT 500 Business Policy	Semester:	Spring 2010
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: Summarized data

- Graphical displays turn data into useful information that managers can use for decision making and interpretation.

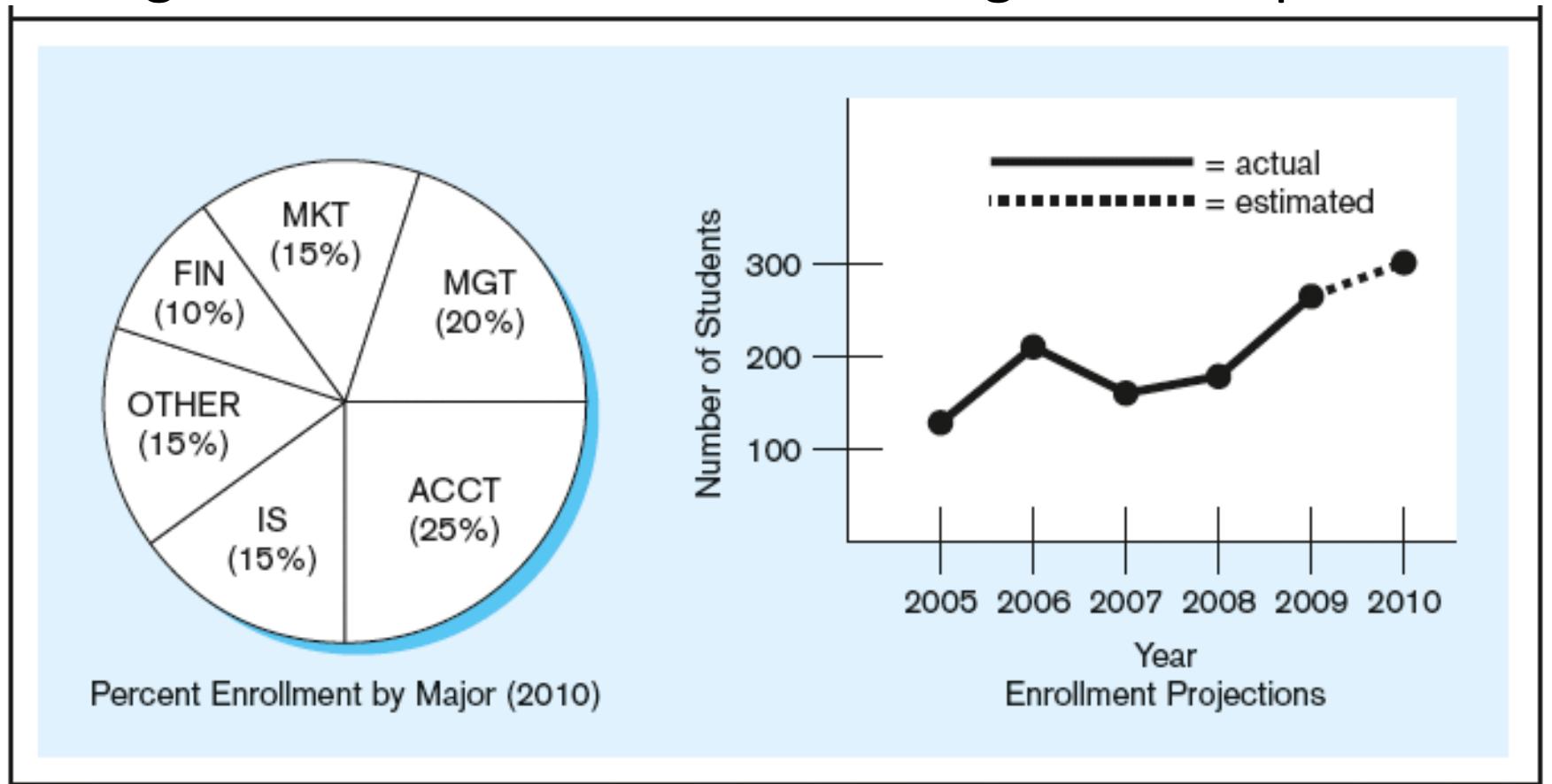


Table 1-1: Example Metadata

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

TABLE 1-1 Example Metadata for Class Roster

Data Item		Metadata				
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

Disadvantages of File Processing

■ Program-Data Dependence

- All programs maintain metadata for each file they use.

■ Duplication of Data

- Different systems/programs have separate copies of the same data.

■ Limited Data Sharing

- No centralized control of data.

■ Lengthy Development Times

- Programmers must design their own file formats.

■ Excessive Program Maintenance

- 80% of information systems budget.



UNIVERSITY OF
MARYLAND

Problems with Data Dependency

- Each application programmer must maintain his/her own data.
- Each application program needs to include code for the metadata of each file.
- Each application program must have its own processing routines for reading, inserting, updating, and deleting data.
- Lack of coordination and central control.
- Non-standard file formats.



UNIVERSITY OF
MARYLAND

Figure 1-2: Old File Processing Systems

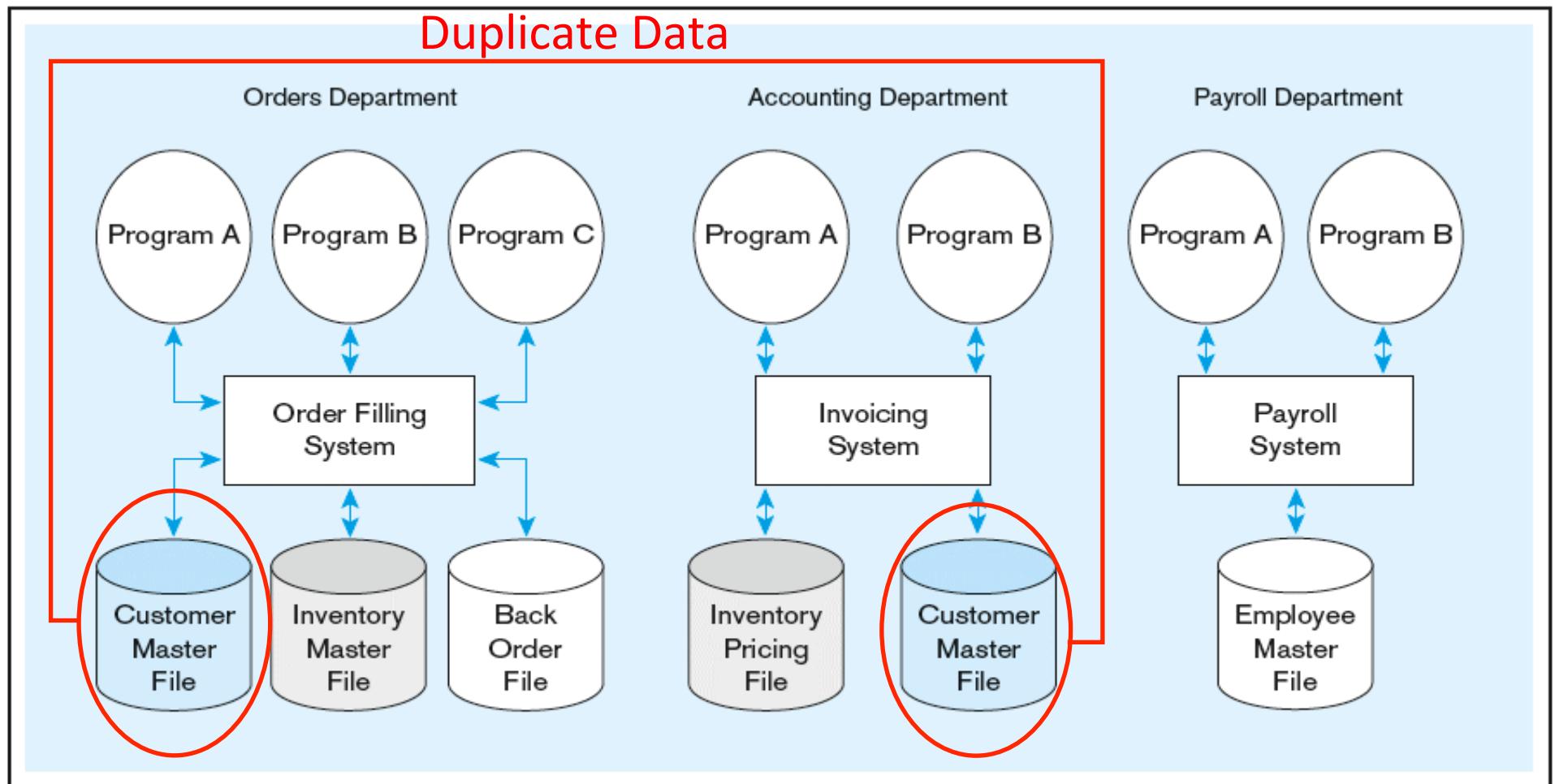


FIGURE 1-2 Old file processing systems at Pine Valley Furniture Company

MARYLAND

Problems with Data Redundancy

- Waste of space to have duplicate data.
- Causes more maintenance headaches.
- The biggest problem:
 - Data changes in one file could cause inconsistencies.
 - Compromises in **data integrity**.



UNIVERSITY OF
MARYLAND

Solution: The Database Approach

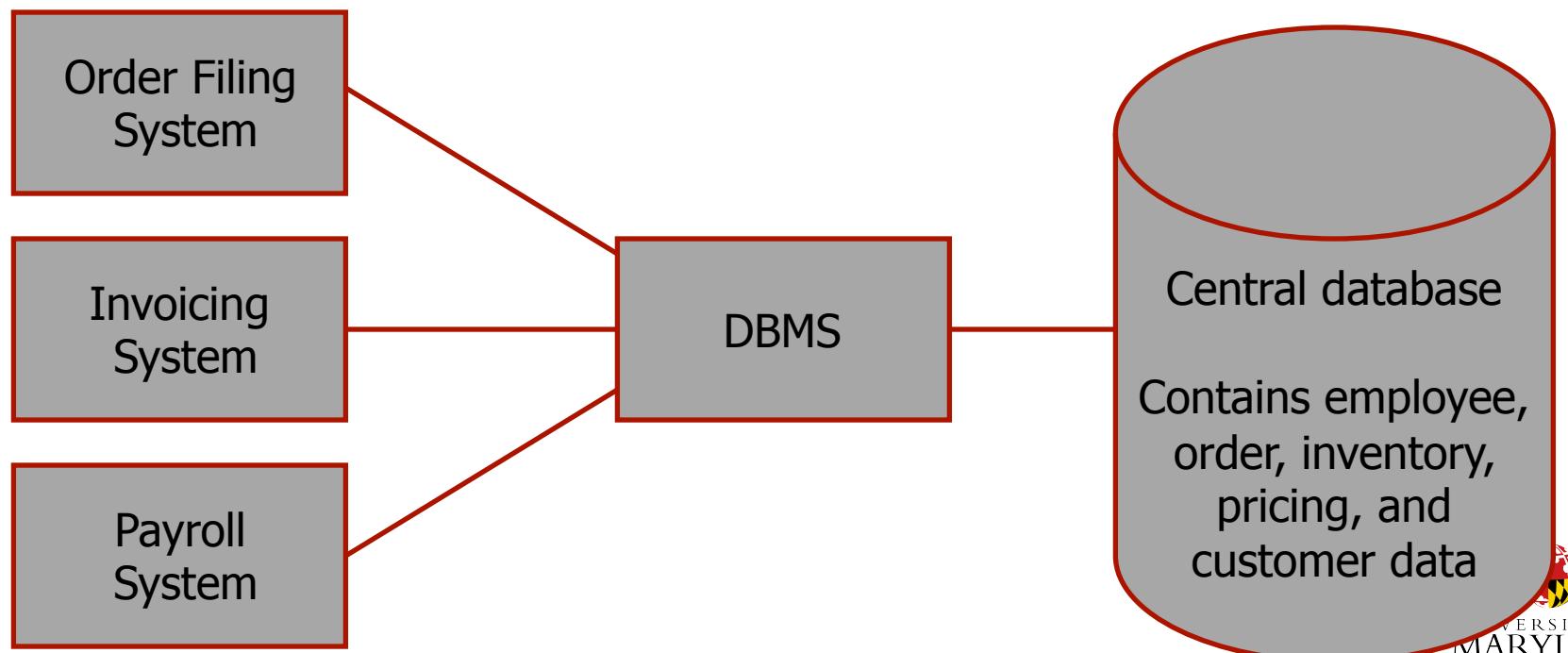
- Central repository of shared data.
- Data is managed by a controlling agent.
- Stored in a standardized, convenient form.
- Requires a Database Management System (DBMS)!



UNIVERSITY OF
MARYLAND

Database Management System (DBMS)

- A software system that is used to create, maintain, and provide controlled access to user databases.
- DBMS manages data resources like an operating system manages hardware resources.



Advantages of the Database Approach

- Program-data independence.
- Planned data redundancy.
- Improved data consistency.
- Improved data sharing.
- Increased application development productivity.
- Enforcement of standards.
- Improved data quality.
- Improved data accessibility and responsiveness.
- Reduced program maintenance.
- Improved decision support.



UNIVERSITY OF
MARYLAND

Costs and Risks of the Database Approach

- New, specialized personnel.
- Installation and management cost and complexity.
- Conversion costs.
- Need for explicit backup and recovery.
- Organizational conflict.



UNIVERSITY OF
MARYLAND

Elements of the Database Approach

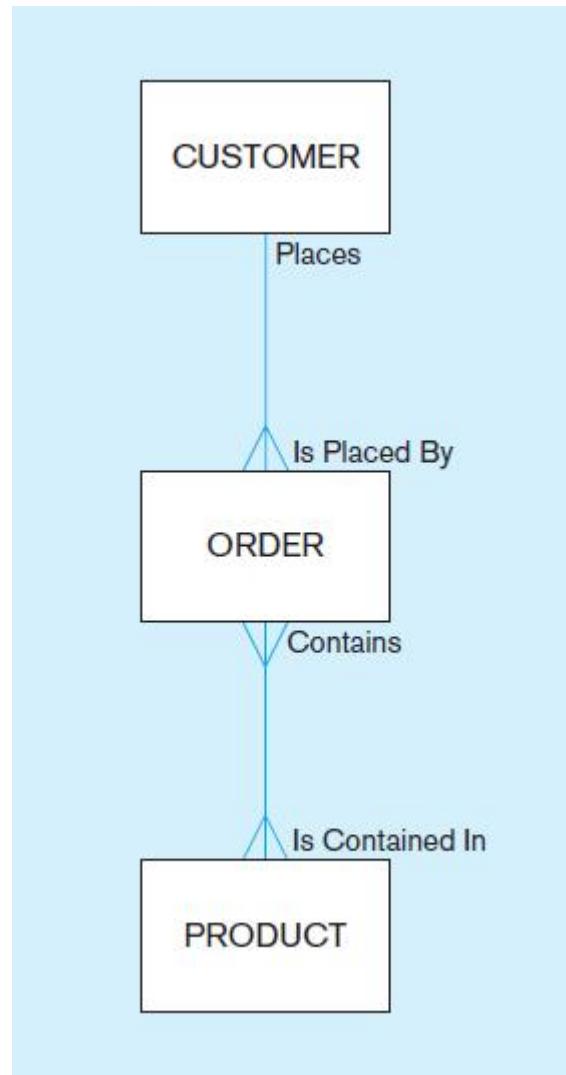
- Data models:

- Graphical system capturing nature and relationship of data.
- **Enterprise Data Model** – high-level entities and relationships for the organization.
- **Project Data Model** – more detailed view, matching data structure in database or data warehouse.



UNIVERSITY OF
MARYLAND

Figure 1-3: Comparison of Enterprise and Project Level Data Models



Segment of an enterprise data model

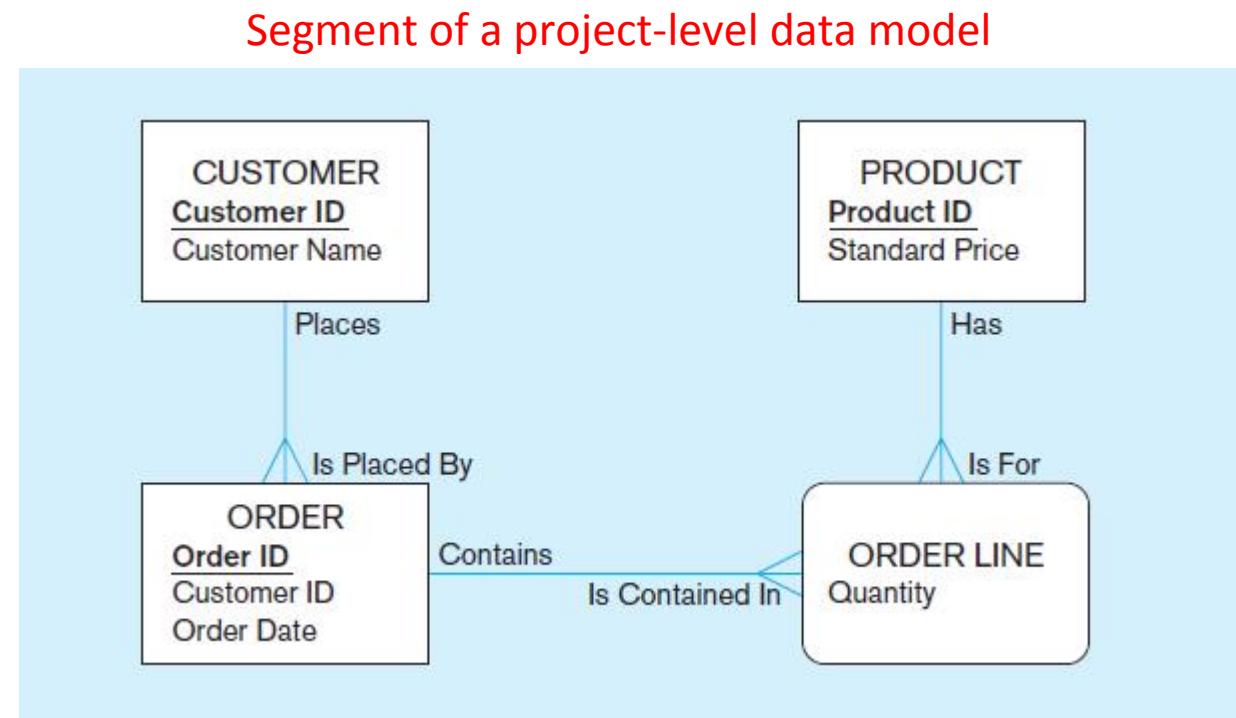


Figure 1-3: Project-Level Data Models

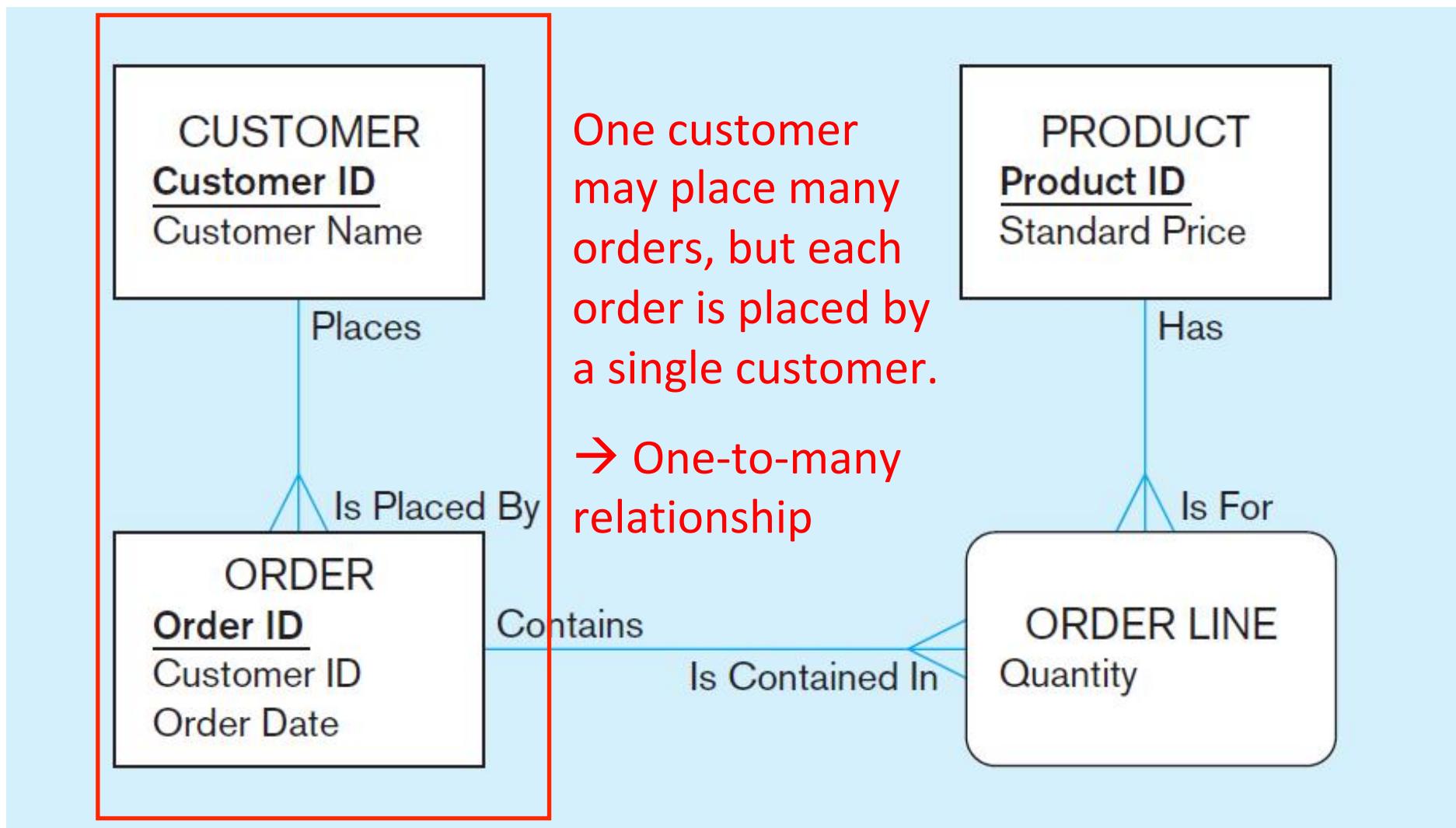


Figure 1-3: Project-Level Data Models

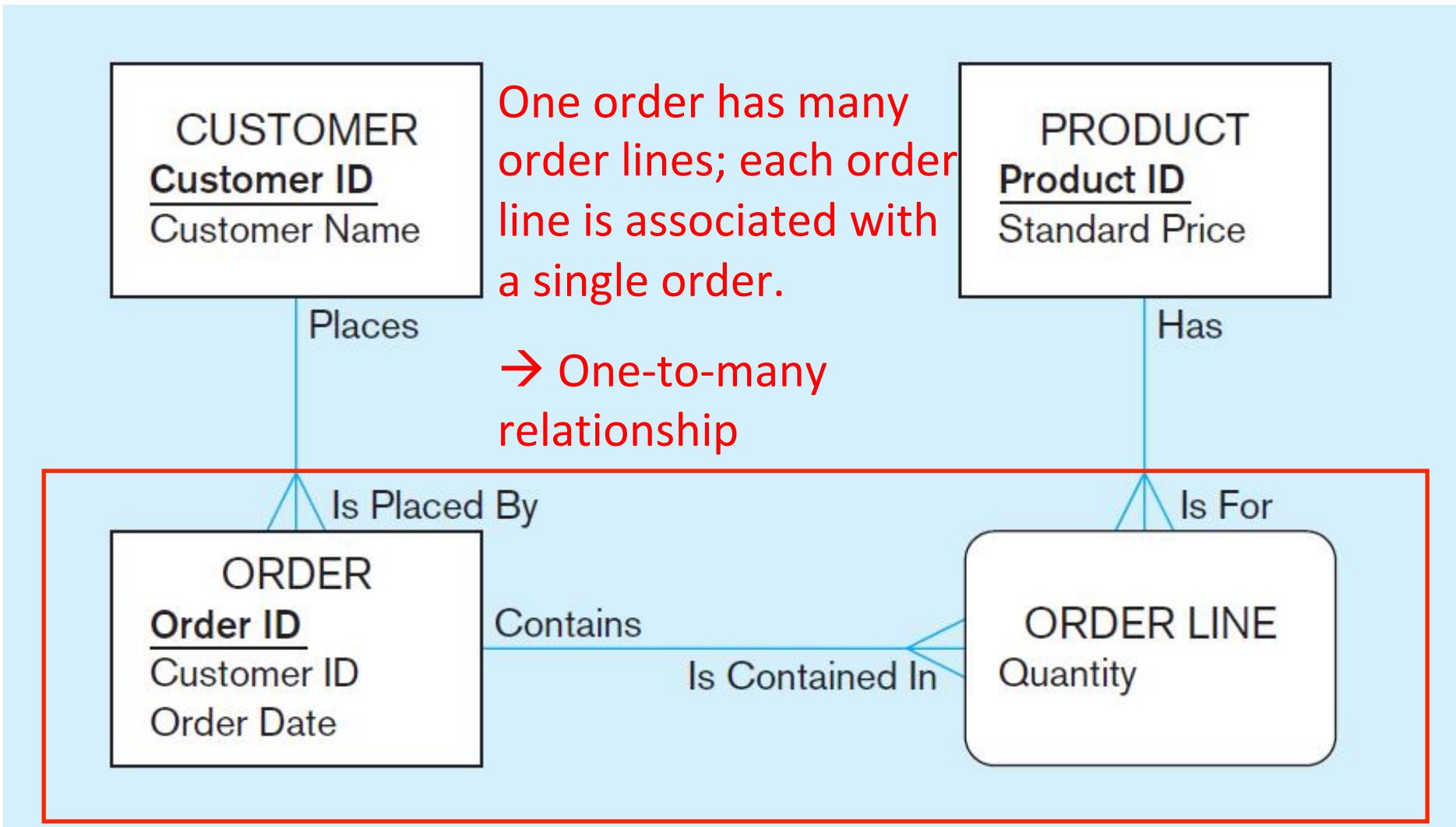


Figure 1-3: Project-Level Data Models

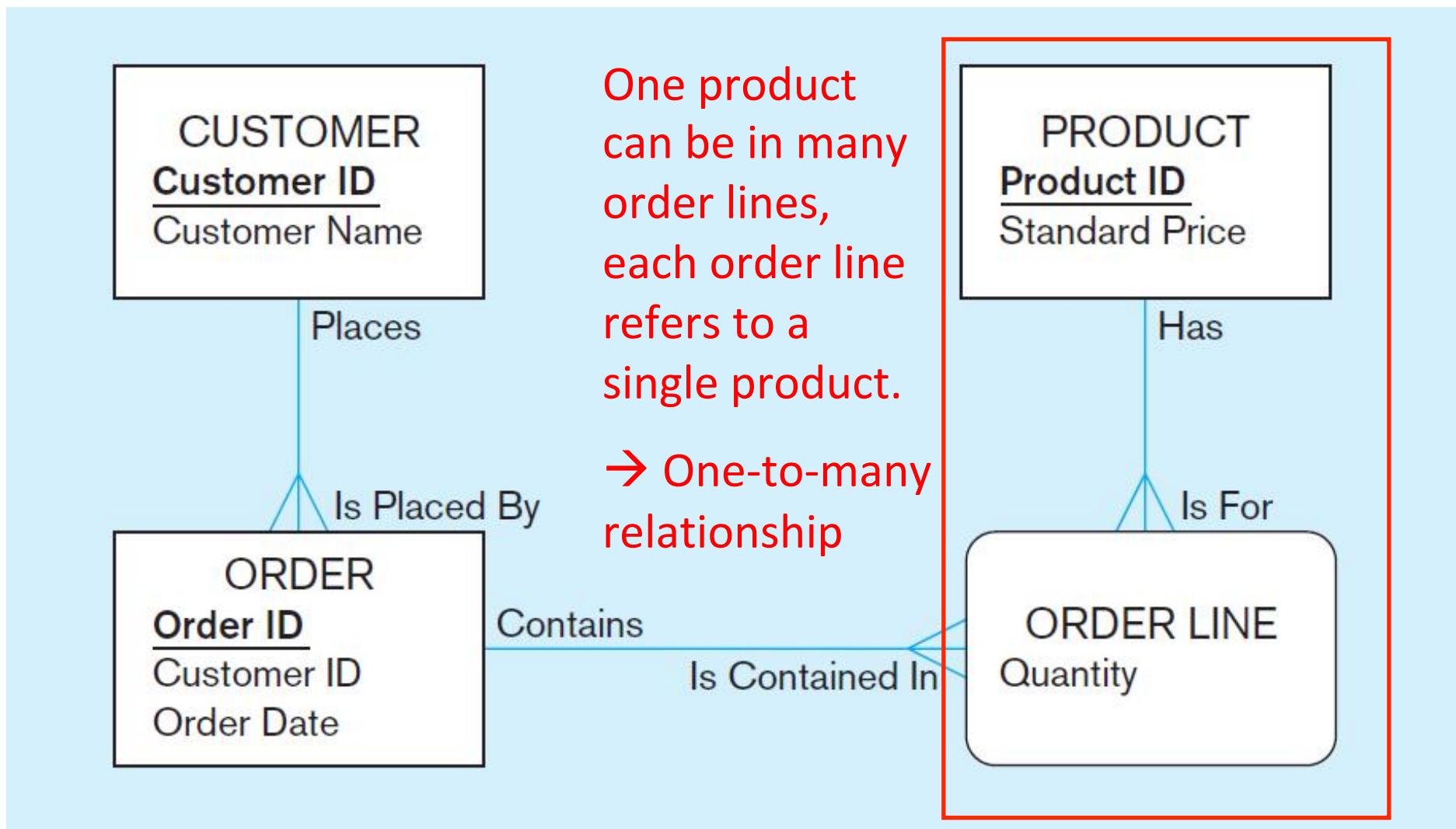


Figure 1-3: Project-Level Data Models

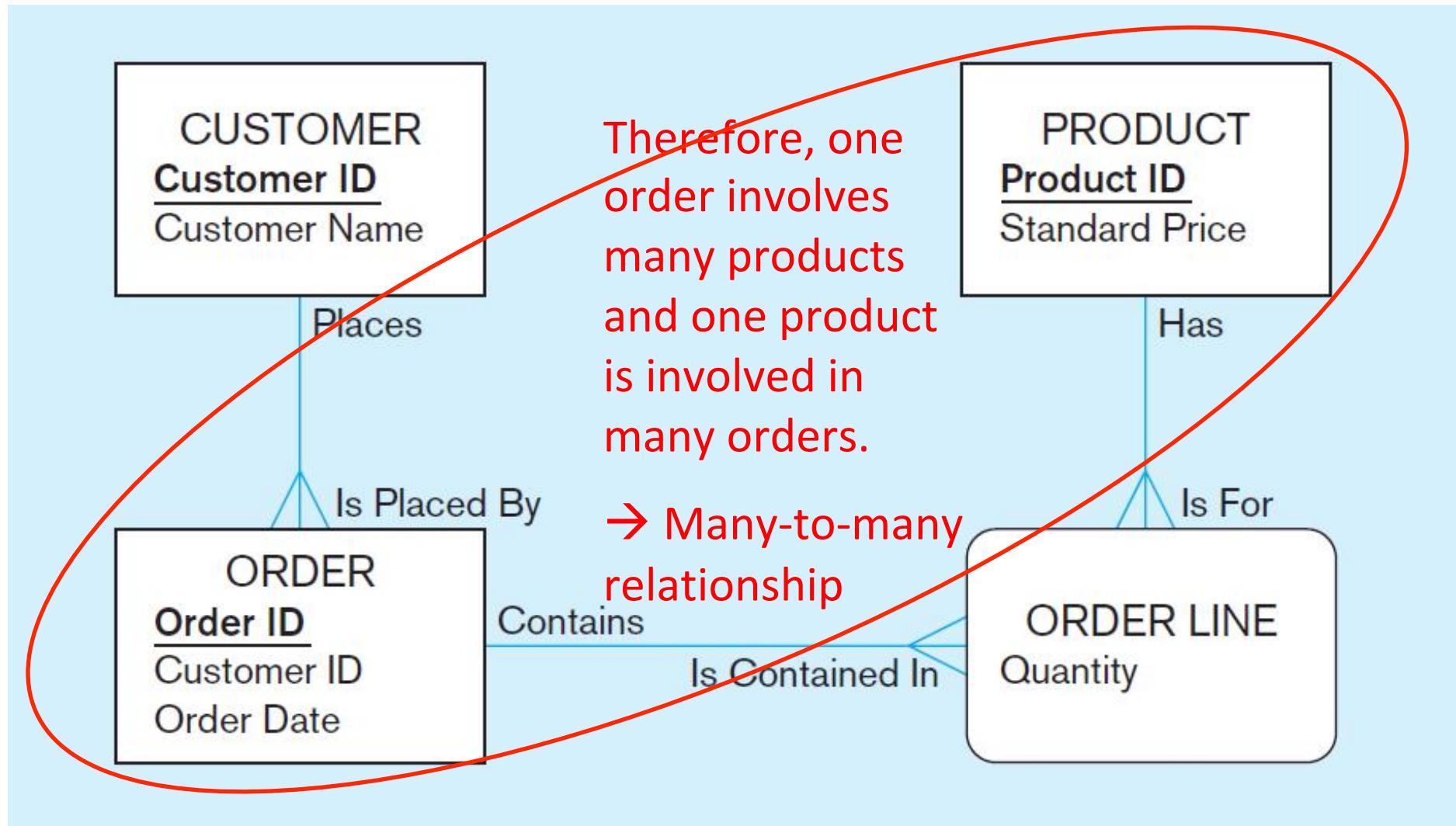


Figure 1-4: Enterprise Data Models

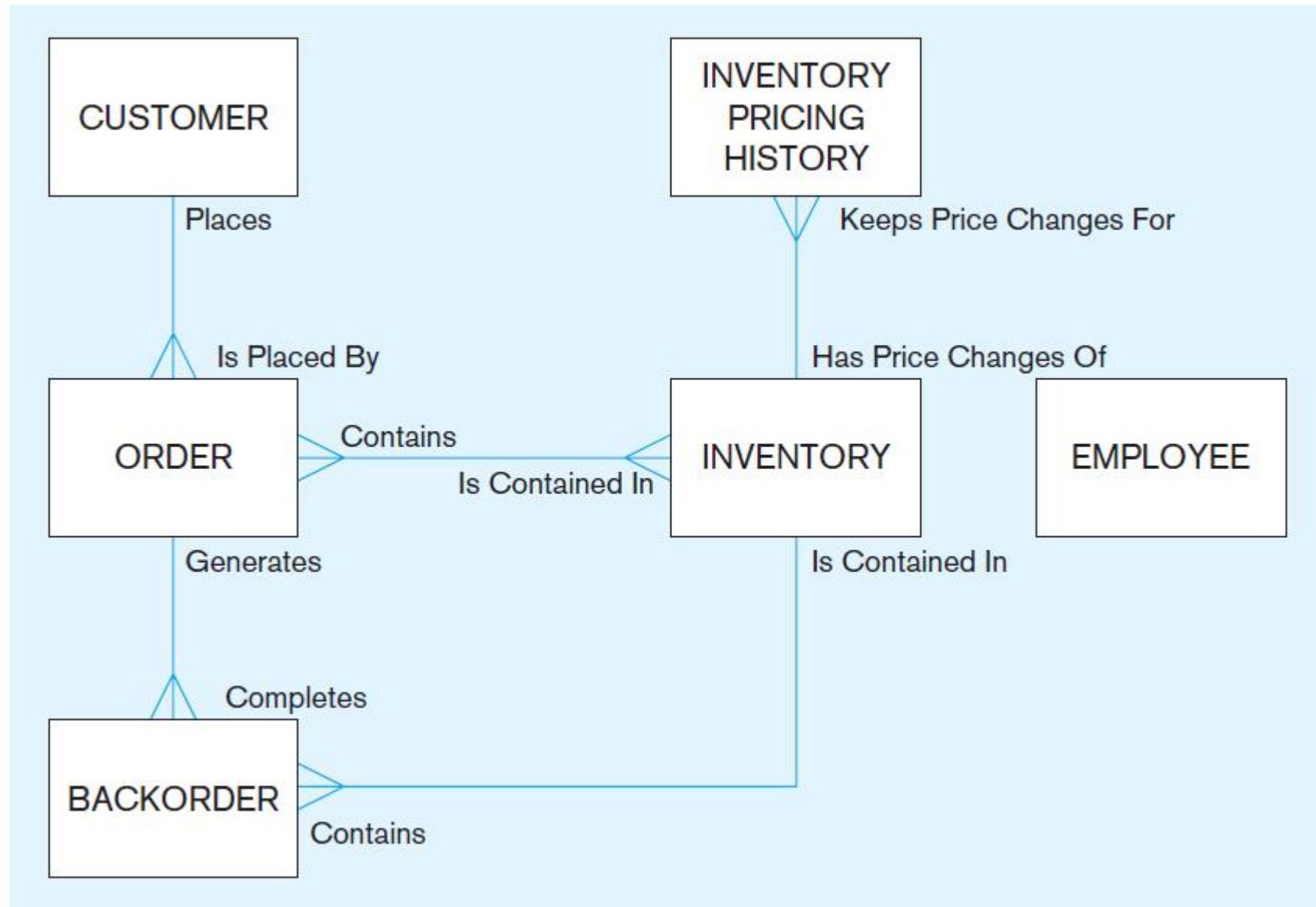
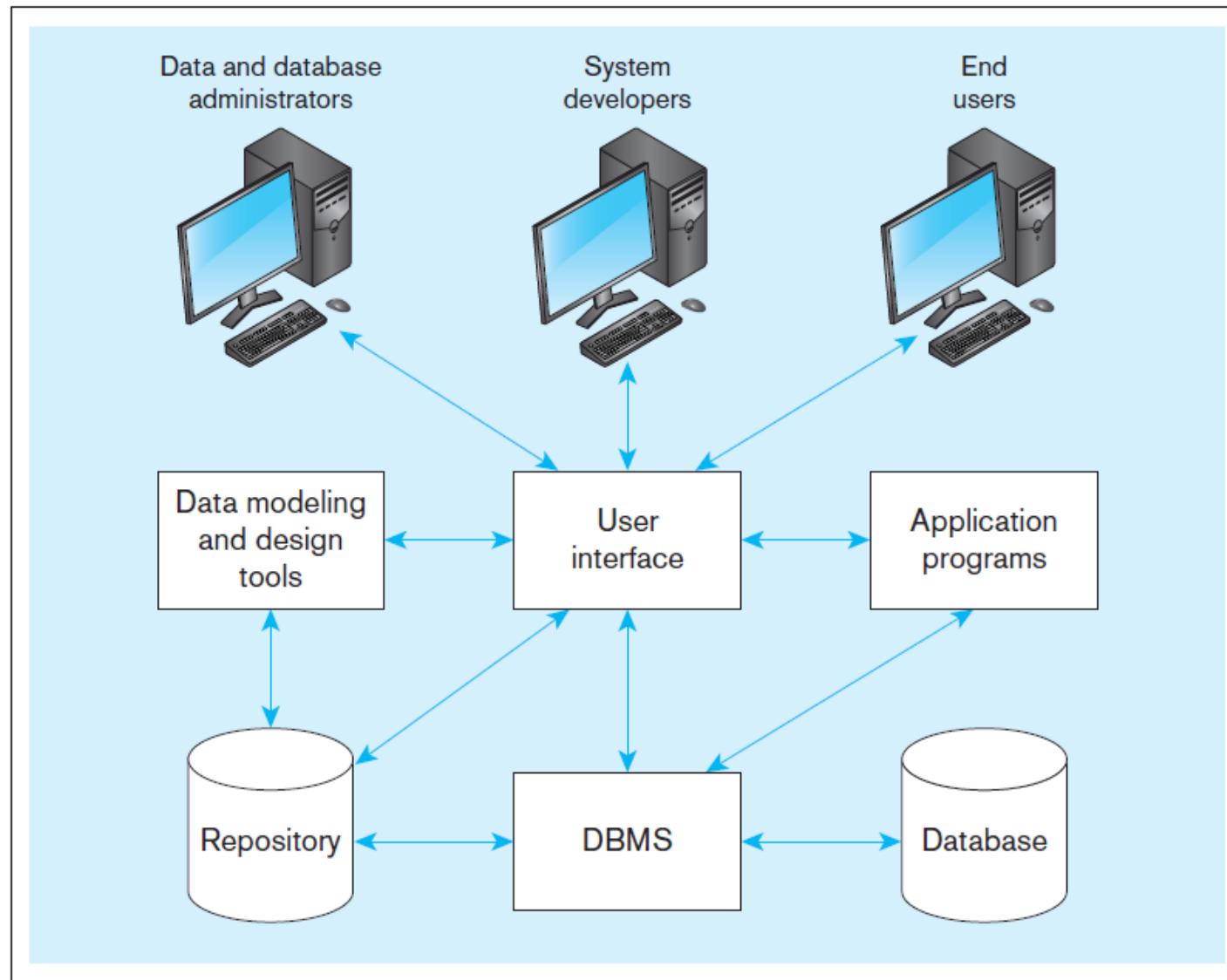


Figure 1-5: Components of DB Environment



UNIVERSITY OF
MARYLAND

Components of the Database Environment

- **Data Modeling and Design Tools** – automated tools
- **Repository** – centralized storehouse of metadata
- **Database Management System (DBMS)** – software for managing the database
- **Database** – storehouse of the data
- **Application Programs** – software using the data
- **User Interface** – text and graphical displays to users
- **Data/Database Administrators** – personnel responsible for maintaining the database
- **System Developers** – personnel responsible for designing databases and software
- **End Users** – people who use the applications and databases



Enterprise Data Model

- First step in the database development process.
- Specifies scope and general content.
- Overall picture of organizational data at high level of abstraction.
- Entity-relationship diagram (ERD).
- Descriptions of entity types.
- Relationships between entities.
- Business rules.



UNIVERSITY OF
MARYLAND

Figure 1-6: Example Business Function-to-Data Entity Matrix

		Data Entity Types								
		Customer	Product	Raw Material	Order	Work Center	Work Order	Invoice	Equipment	Employee
Business Functions	Business Planning	X	X						X	X
	Product Development		X	X		X			X	
	Materials Management		X	X	X	X	X		X	
	Order Fulfillment	X	X	X	X	X	X	X	X	X
	Order Shipment	X	X		X	X		X		X
	Sales Summarization	X	X		X			X		X
	Production Operations		X	X	X	X	X		X	X
	Finance and Accounting	X	X	X	X	X		X	X	X

X = data entity is used within business function



UNIVERSITY OF
MARYLAND

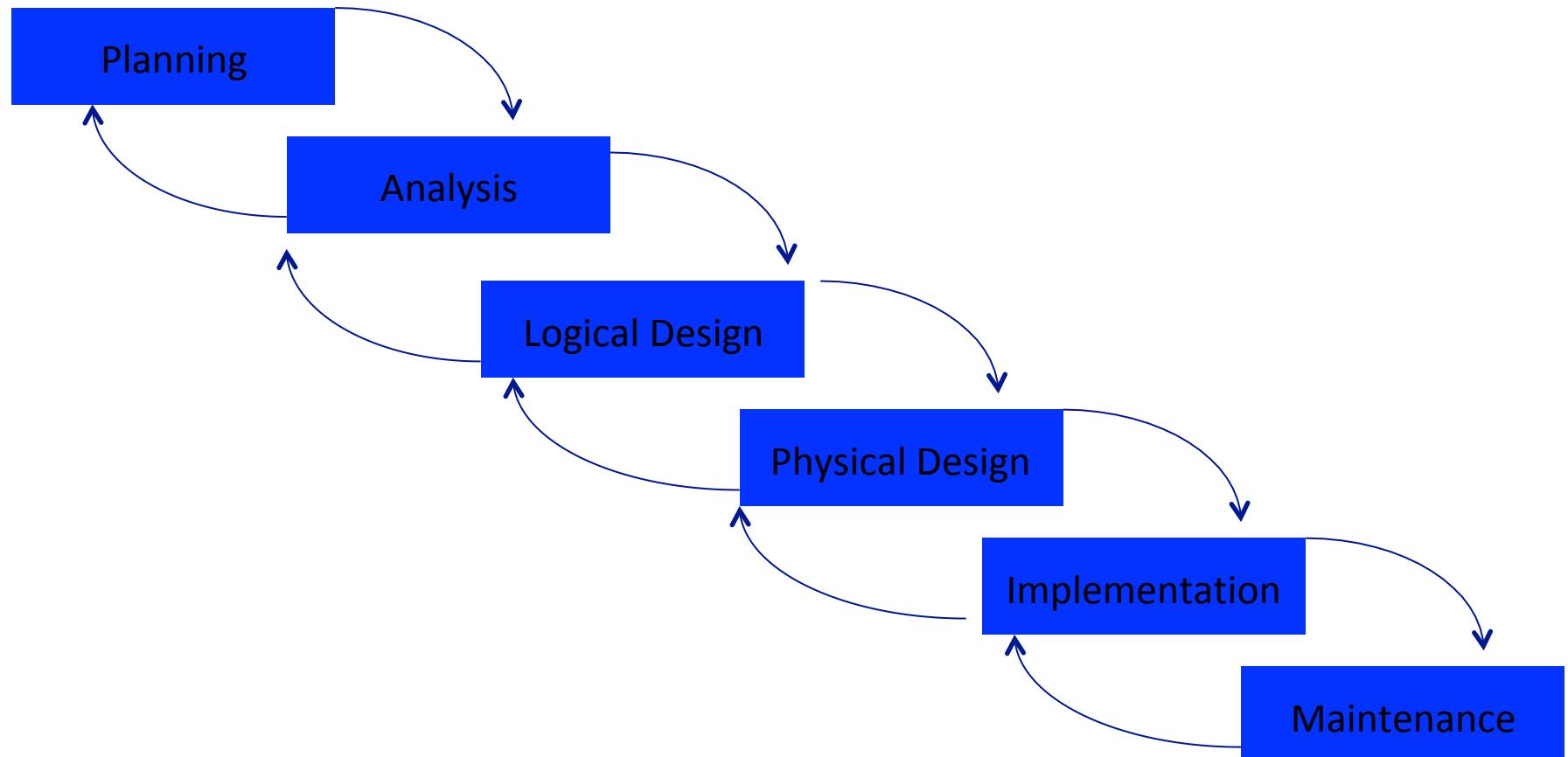
Two Approaches to Database and IS Development

- 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.



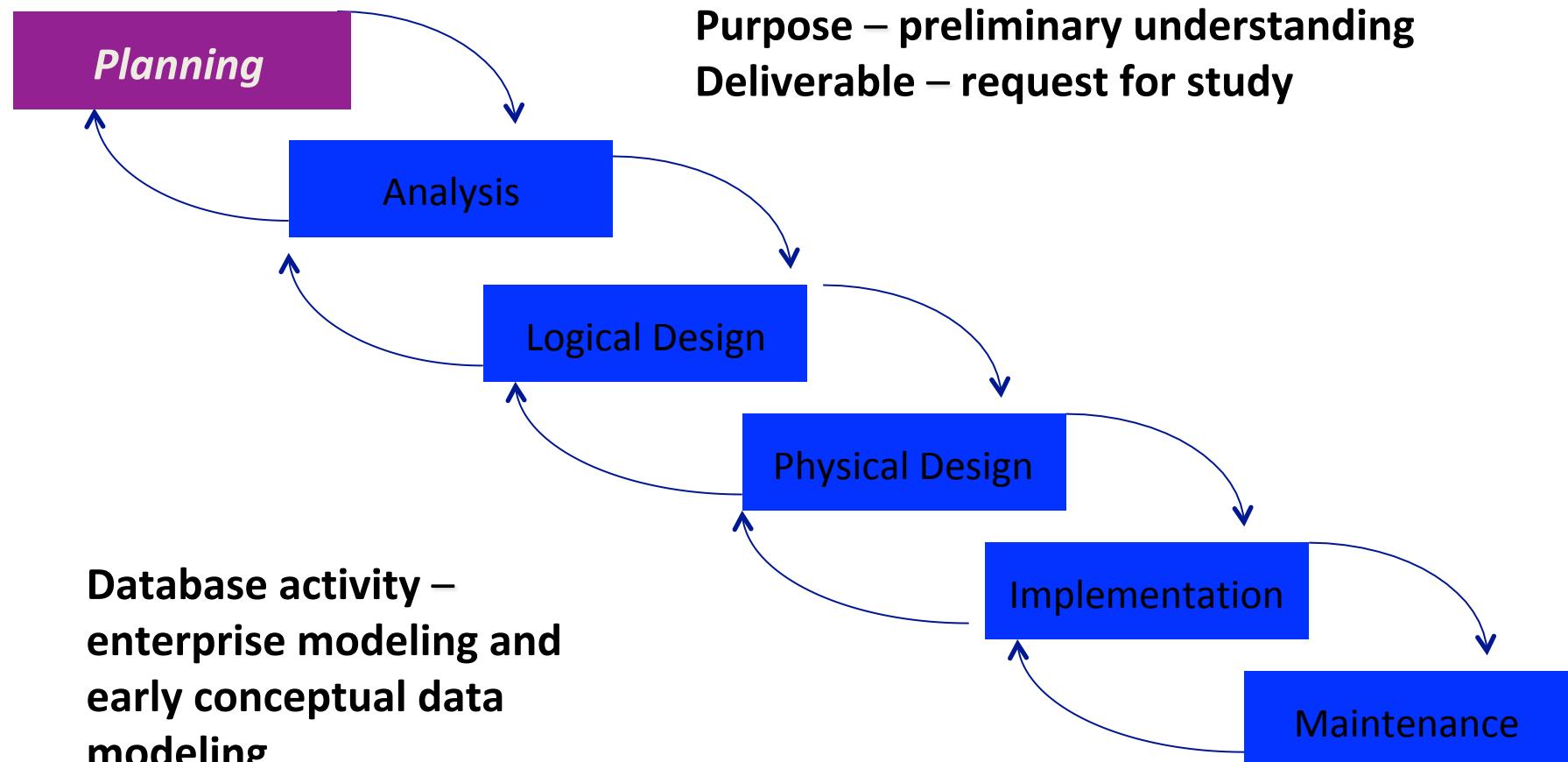
UNIVERSITY OF
MARYLAND

Figure 1-7: Systems Development Life Cycle



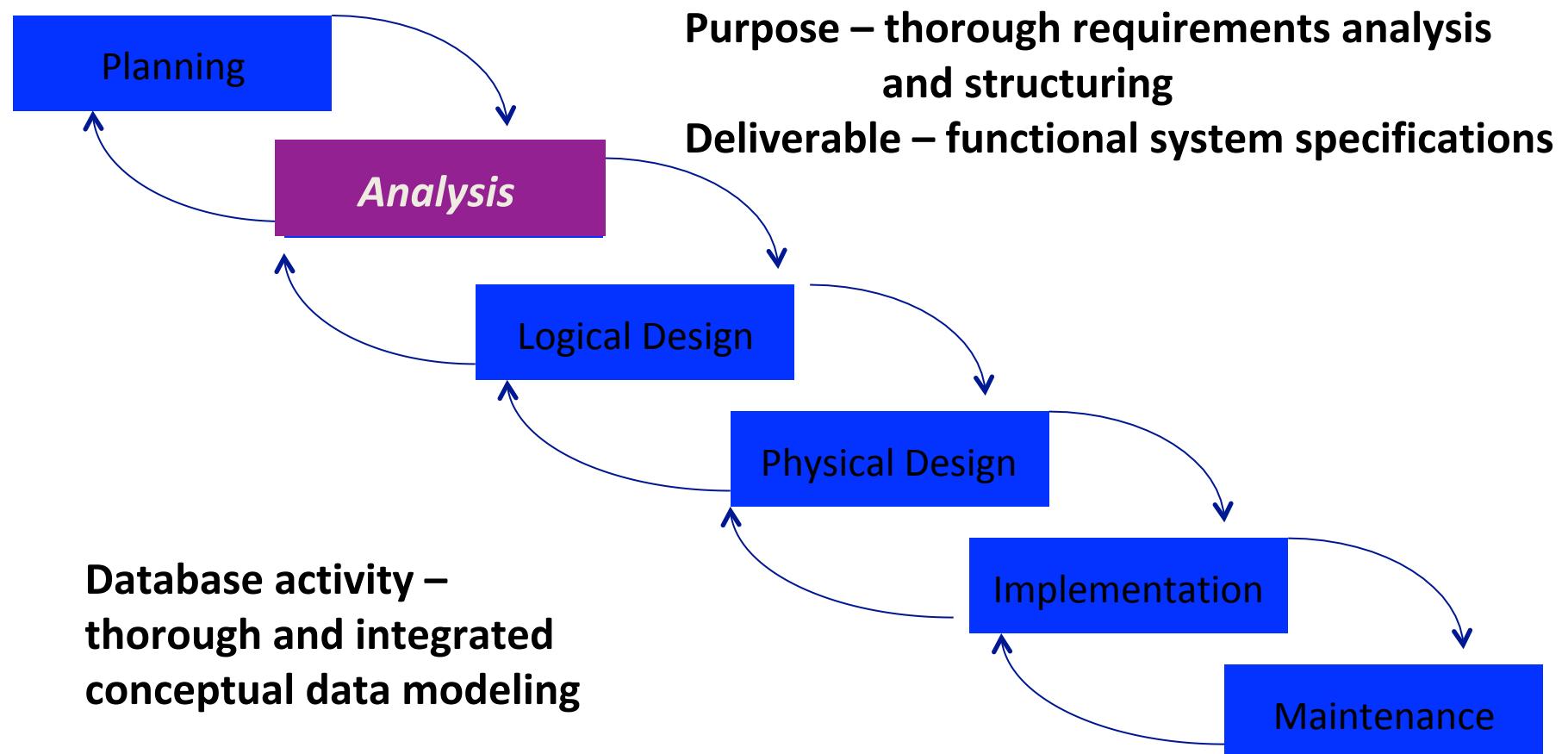
UNIVERSITY OF
MARYLAND

Figure 1-7: Systems Development Life Cycle



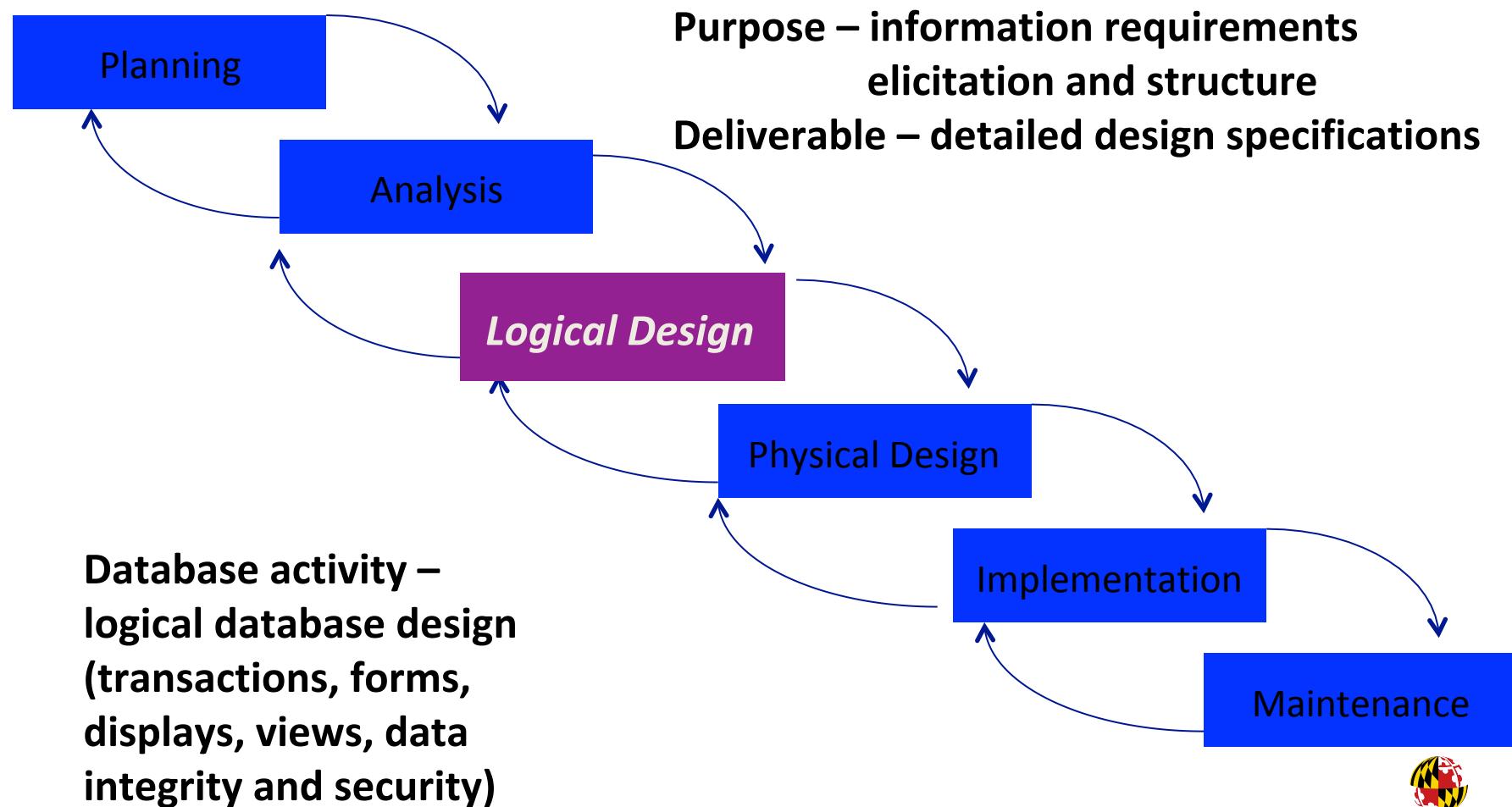
UNIVERSITY OF
MARYLAND

Figure 1-7: Systems Development Life Cycle



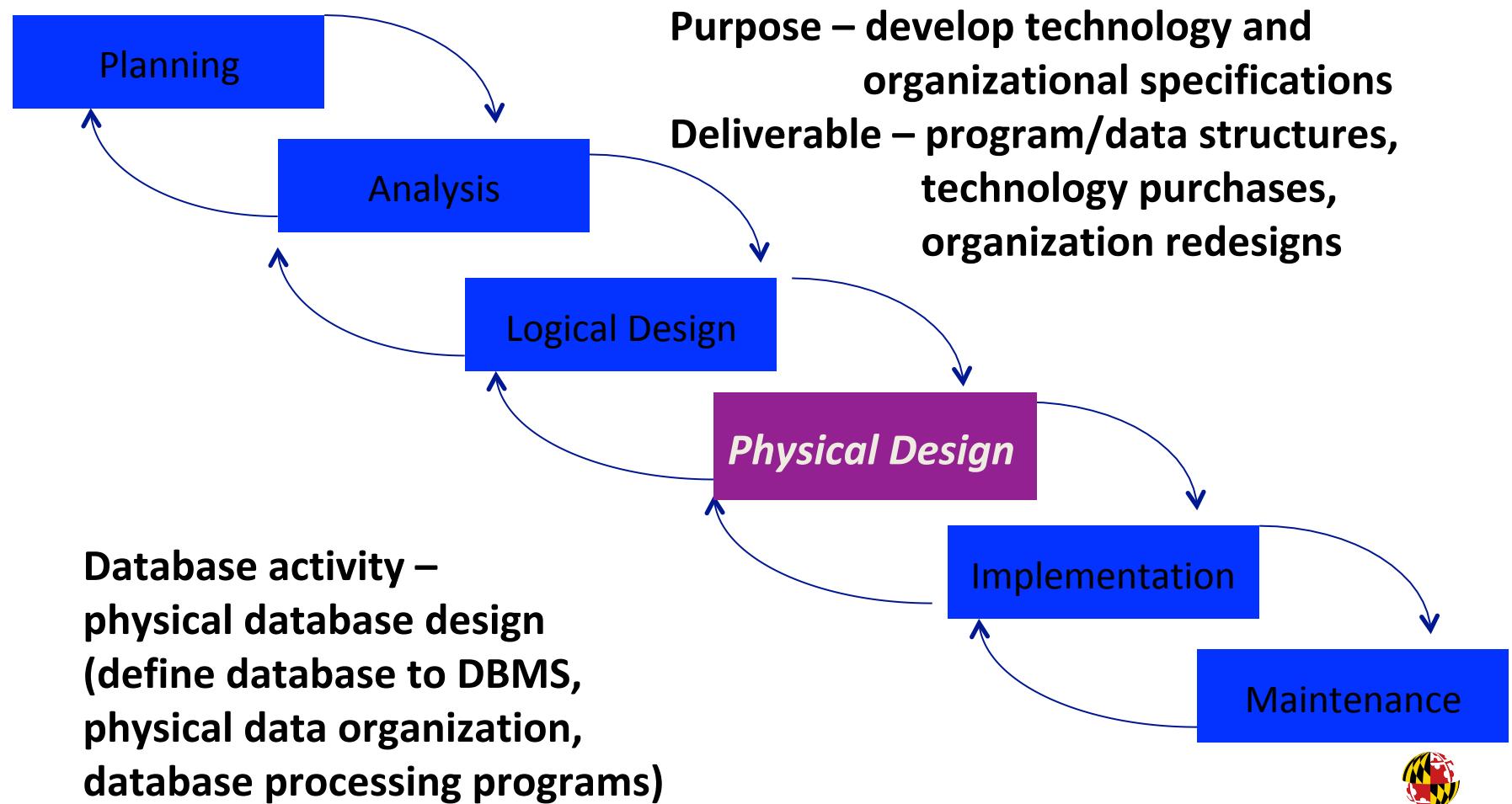
UNIVERSITY OF
MARYLAND

Figure 1-7: Systems Development Life Cycle



UNIVERSITY OF
MARYLAND

Figure 1-7: Systems Development Life Cycle



UNIVERSITY OF
MARYLAND

Figure 1-7: Systems Development Life Cycle

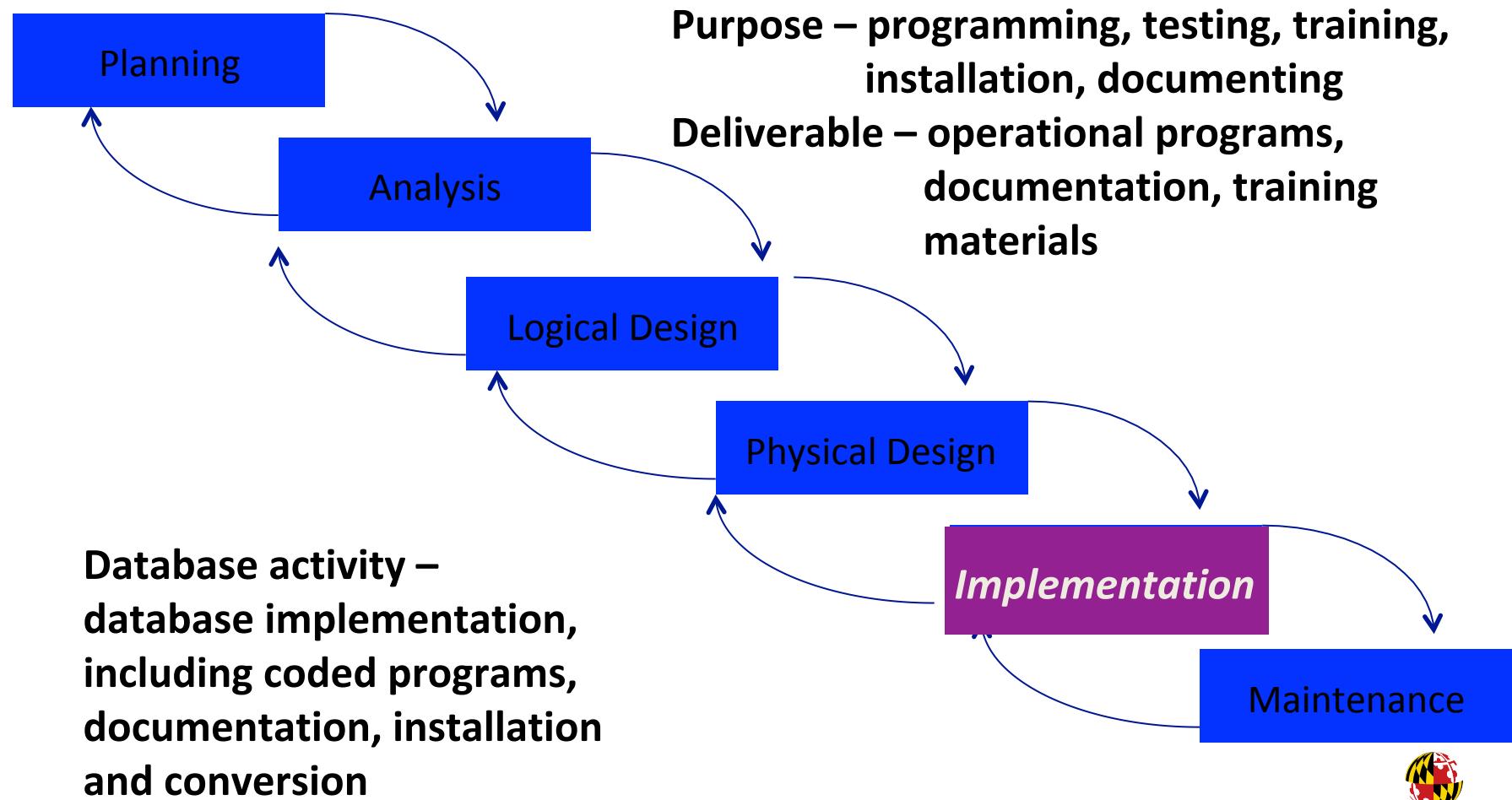


Figure 1-7: Systems Development Life Cycle

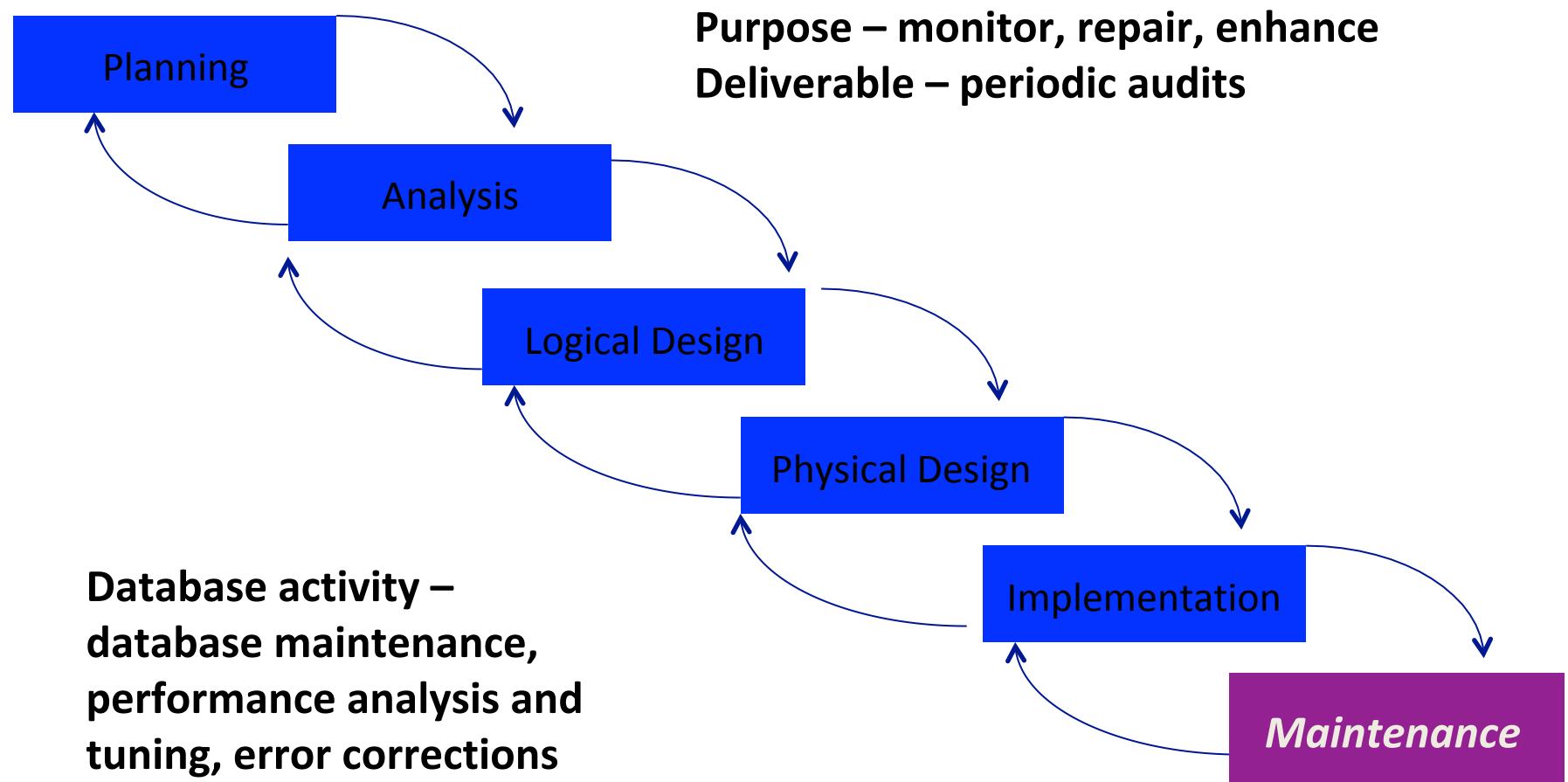
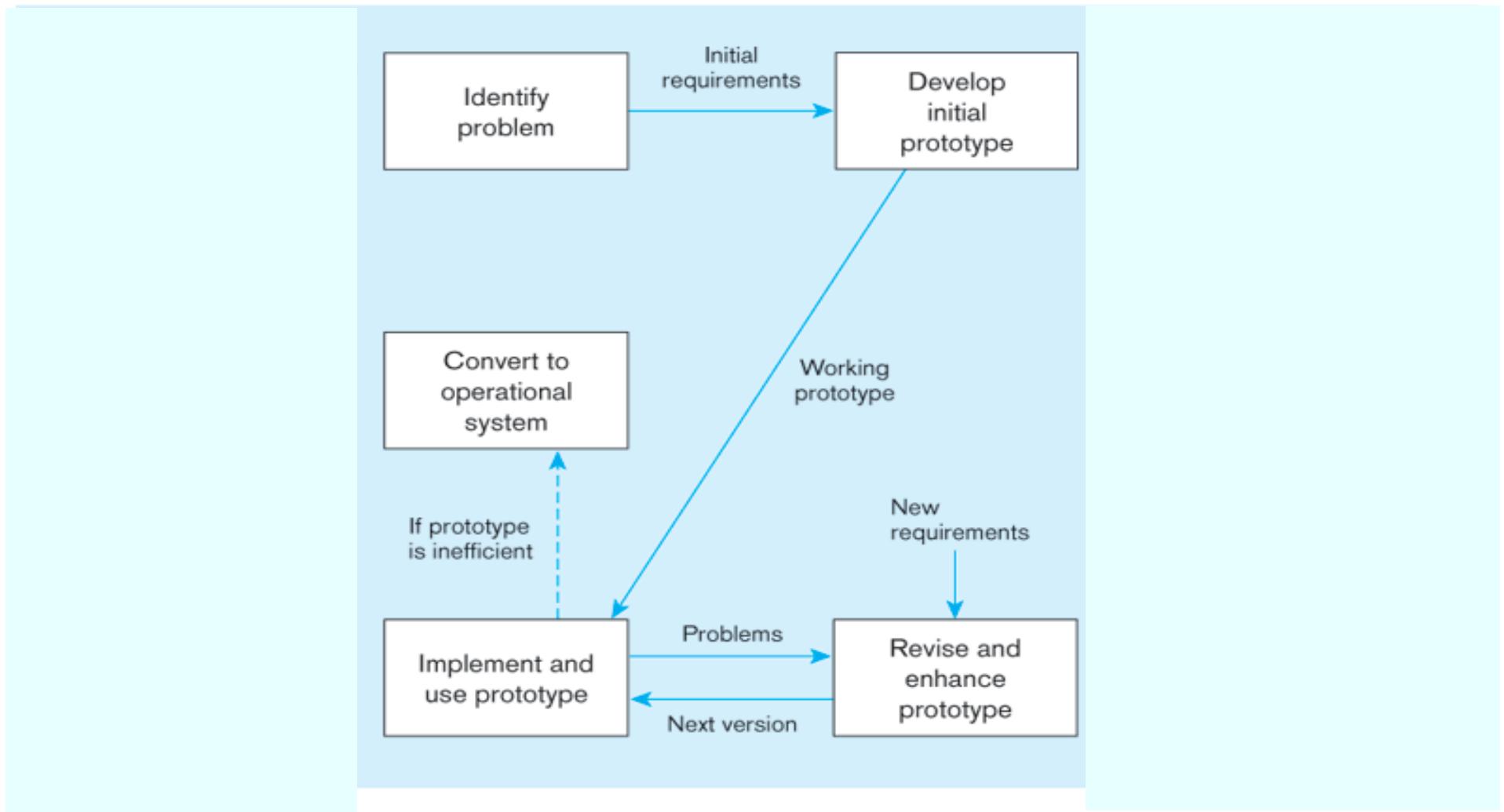
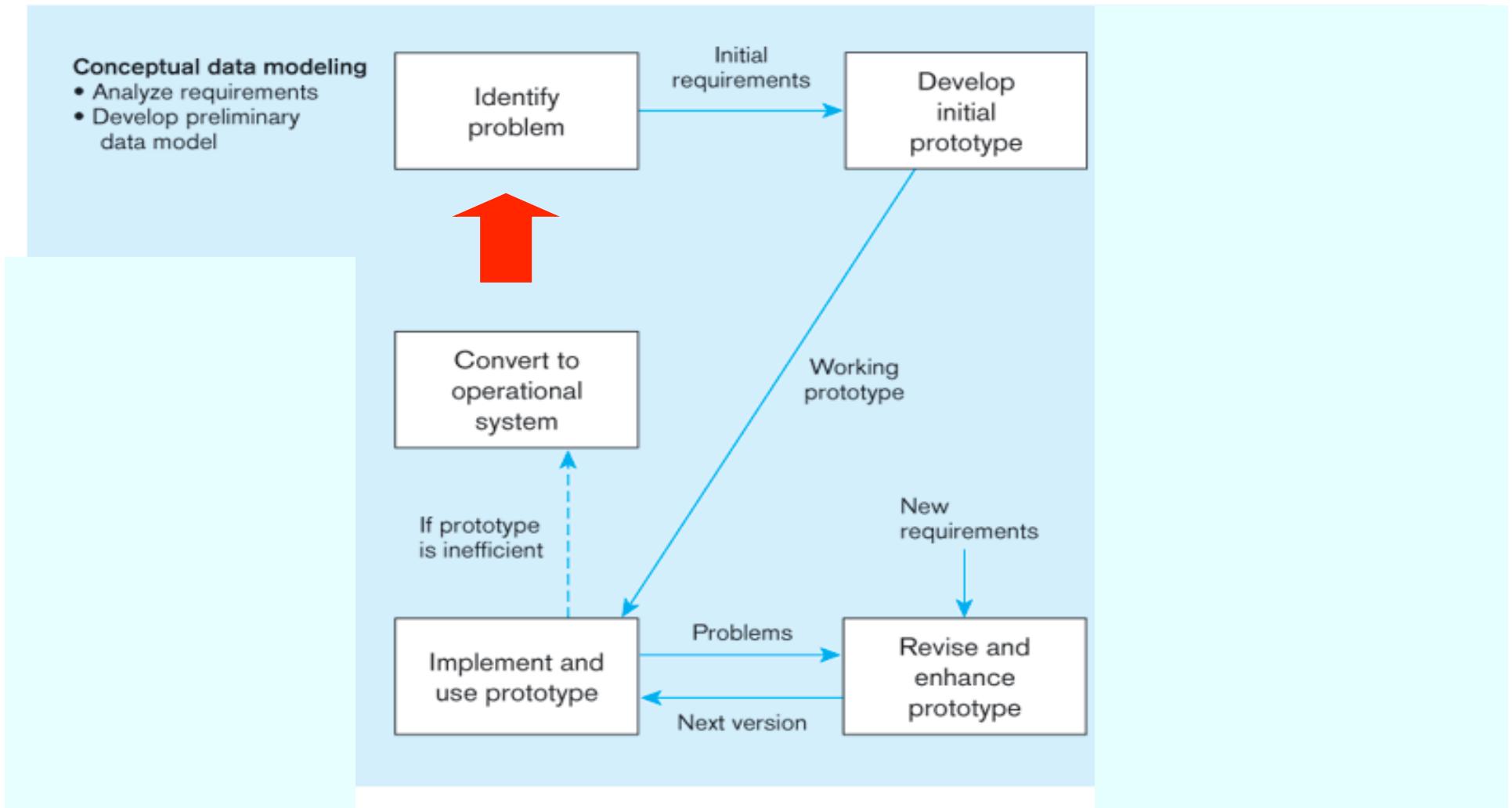


Figure 1-8: Prototyping Database Methodology



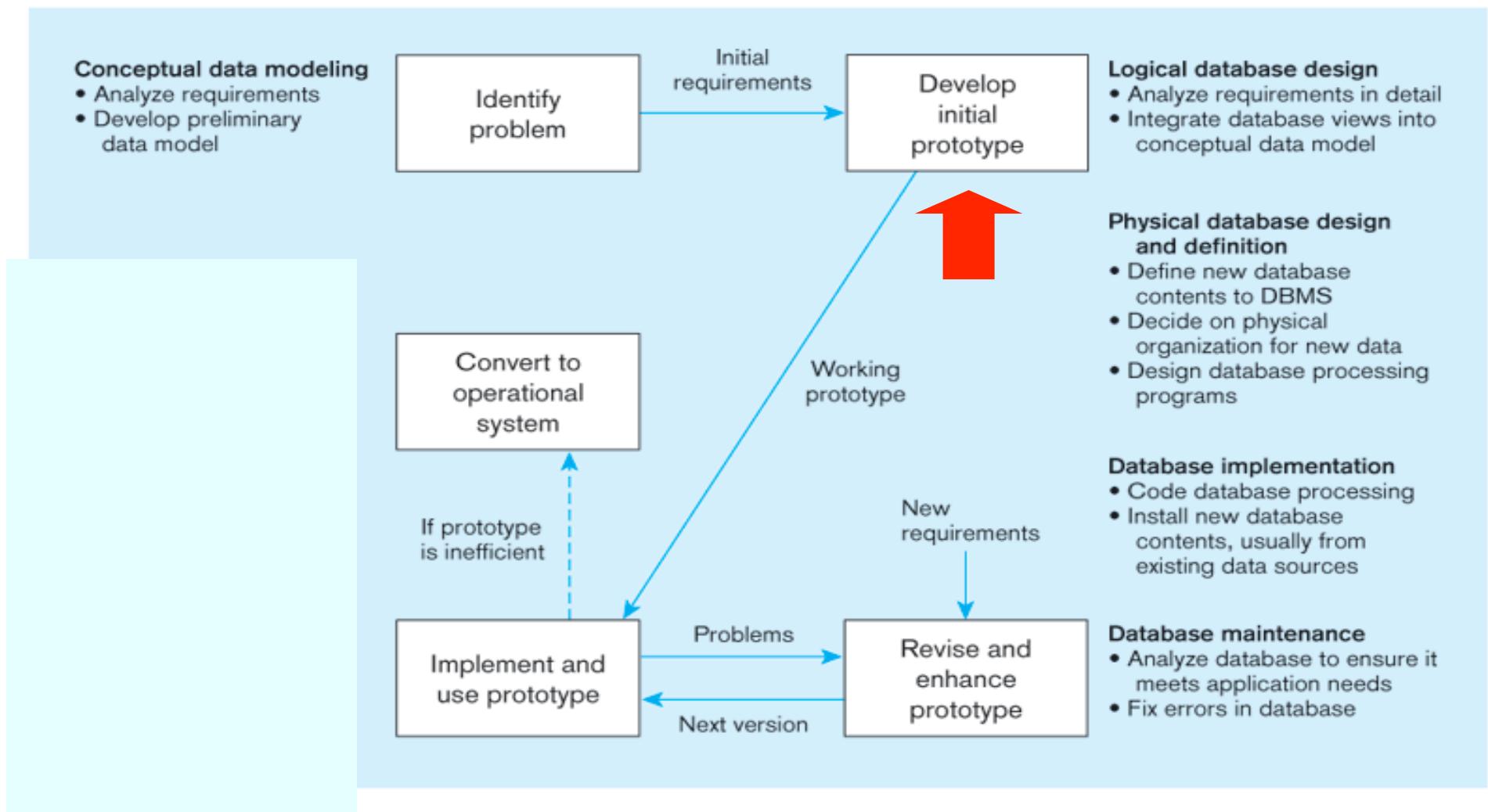
UNIVERSITY OF
MARYLAND

Figure 1-8: Prototyping Database Methodology



UNIVERSITY OF
MARYLAND

Figure 1-8: Prototyping Database Methodology



UNIVERSITY OF
MARYLAND

Figure 1-8: Prototyping Database Methodology

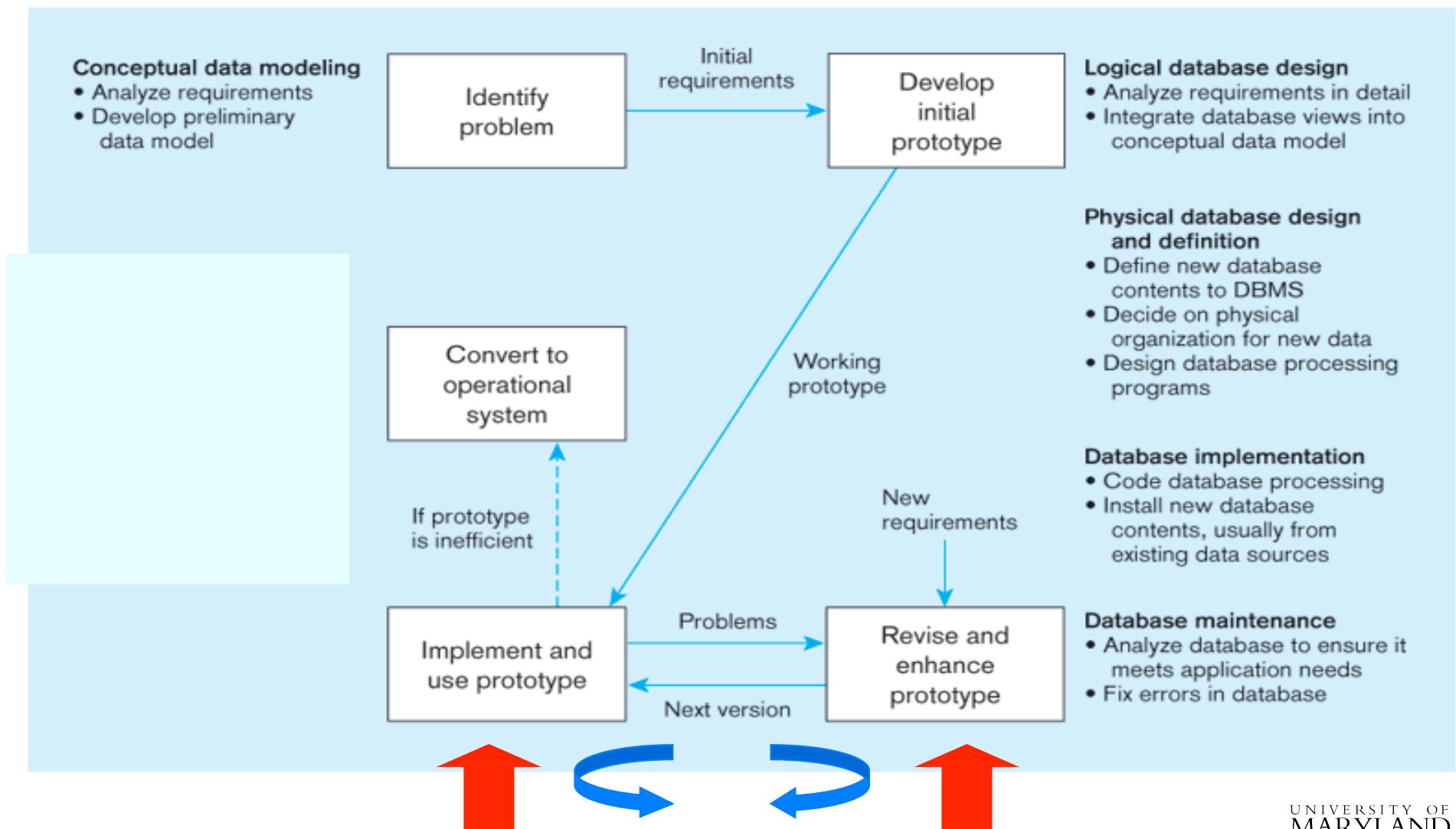
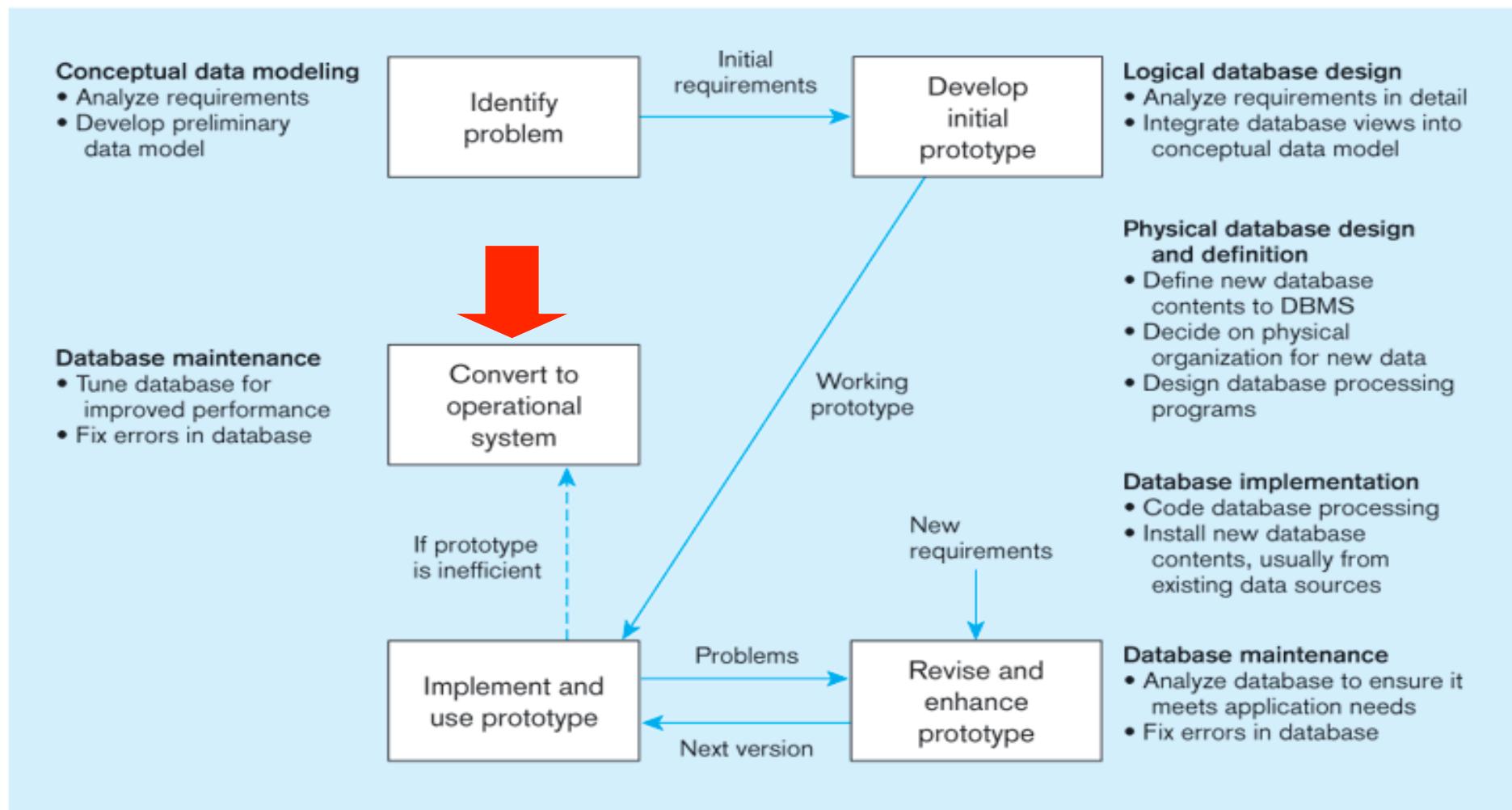


Figure 1-8: Prototyping Database Methodology



Database Schema

■ External Schema:

- User Views
- Subsets of Conceptual Schema
- Can be determined from business-function/data entity matrices
- DBA determines schema for different users

■ Conceptual Schema:

- ER models – *covered in Chapters 2 and 3*

■ Internal Schema:

- Logical structures – *covered in Chapter 4*
- Physical structures – *covered in Chapters 5, 6 and 7*

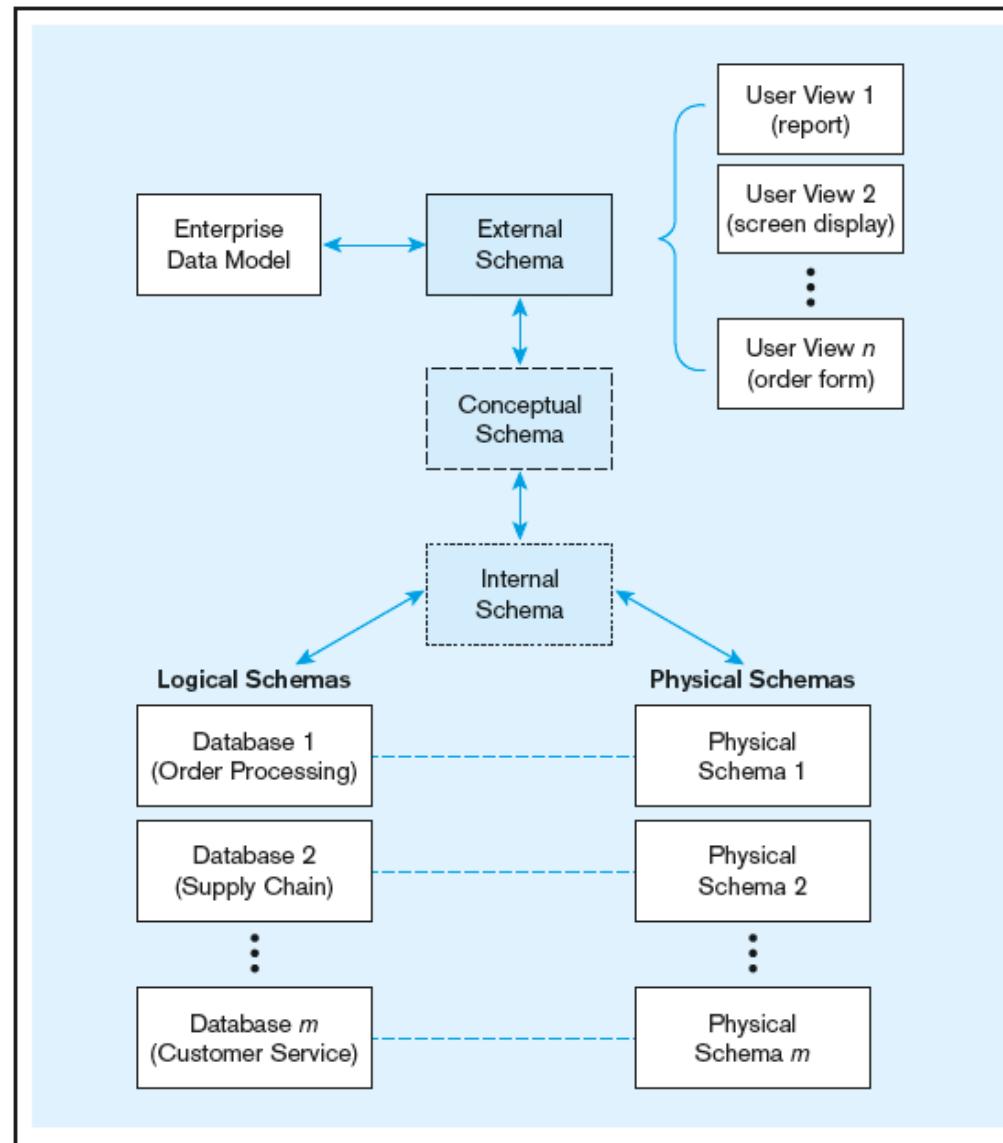


UNIVERSITY OF
MARYLAND

Figure 1-9: Three-Schema Architecture

Different people
have different views
of the database ...
these are the
external schema.

The internal schema
is the underlying
design and
implementation.



Managing Projects

- **Project** – a planned undertaking of related activities to reach an objective that has a beginning and an end.
- Initiated and planned in planning stage of SDLC.
- Executed during analysis, design, and implementation.
- Closed at the end of implementation.



UNIVERSITY OF
MARYLAND

Managing Projects: People Involved

- Business analysts
- Systems analysts
- Database analysts and data modelers
- Users
- Programmers
- Database architects
- Data administrators
- Project managers
- Other technical experts



UNIVERSITY OF
MARYLAND

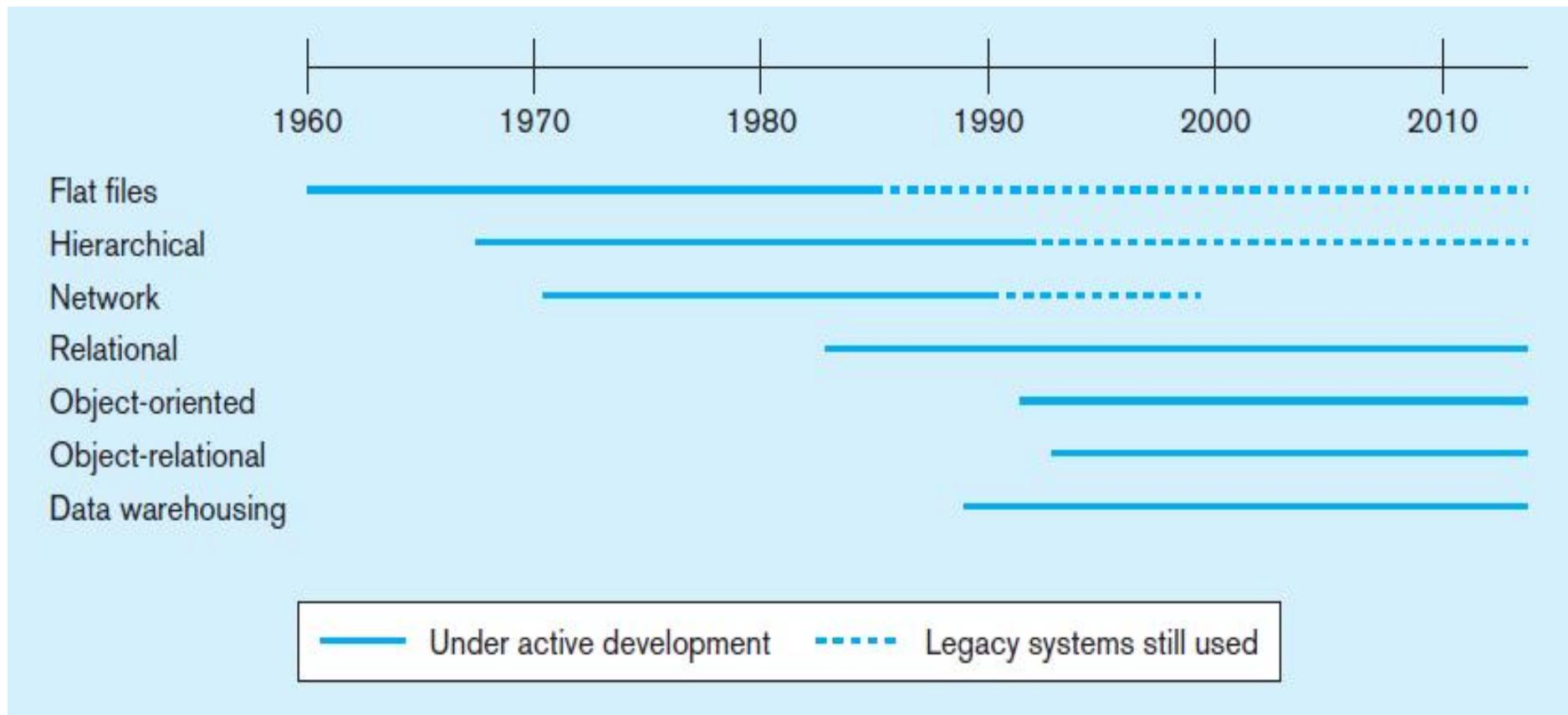
Evolution of Database Systems

- Driven by four main objectives:
 - Need for program-data independence → reduced maintenance.
 - Desire to manage more complex data types and structures.
 - Ease of data access for less technical personnel.
 - Need for more powerful decision support platforms.



UNIVERSITY OF
MARYLAND

Figure 1-10: Evolution of Database Technologies



UNIVERSITY OF
MARYLAND

Figure 1-10: Database Architectures

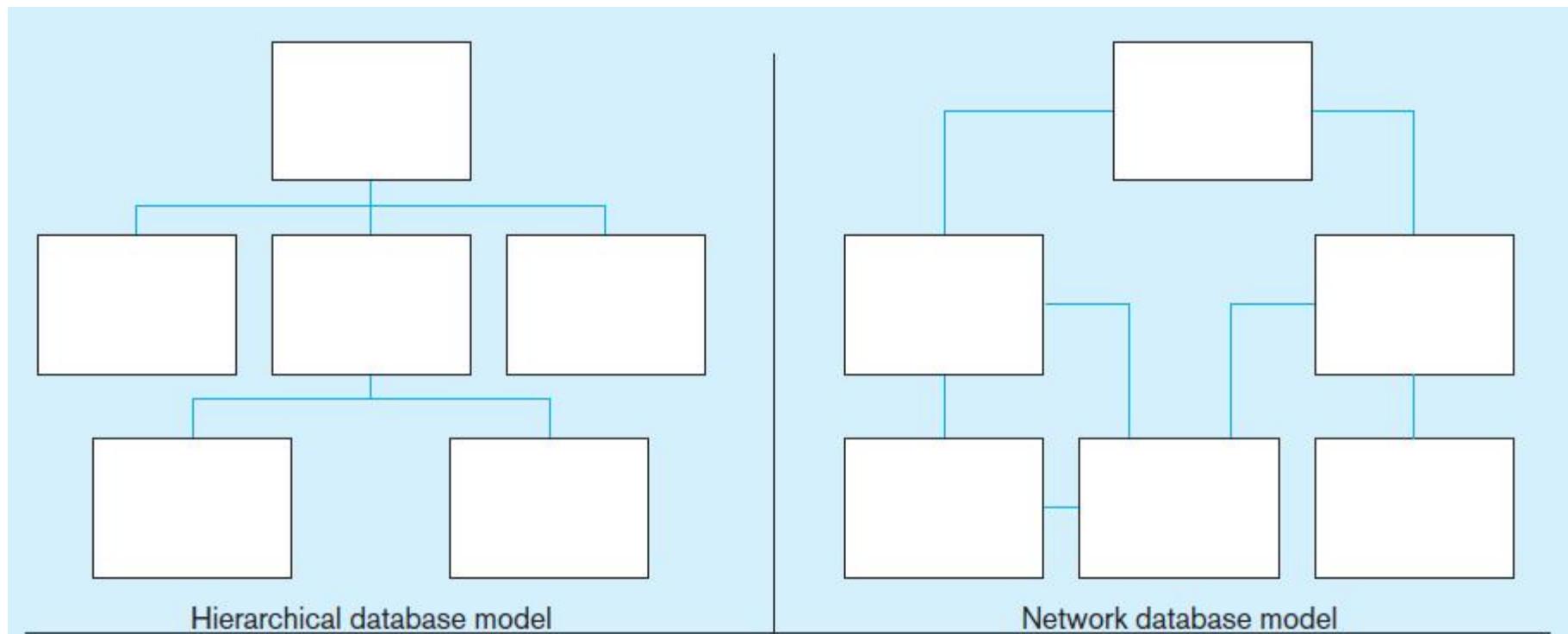


Figure 1-10: Database Architectures

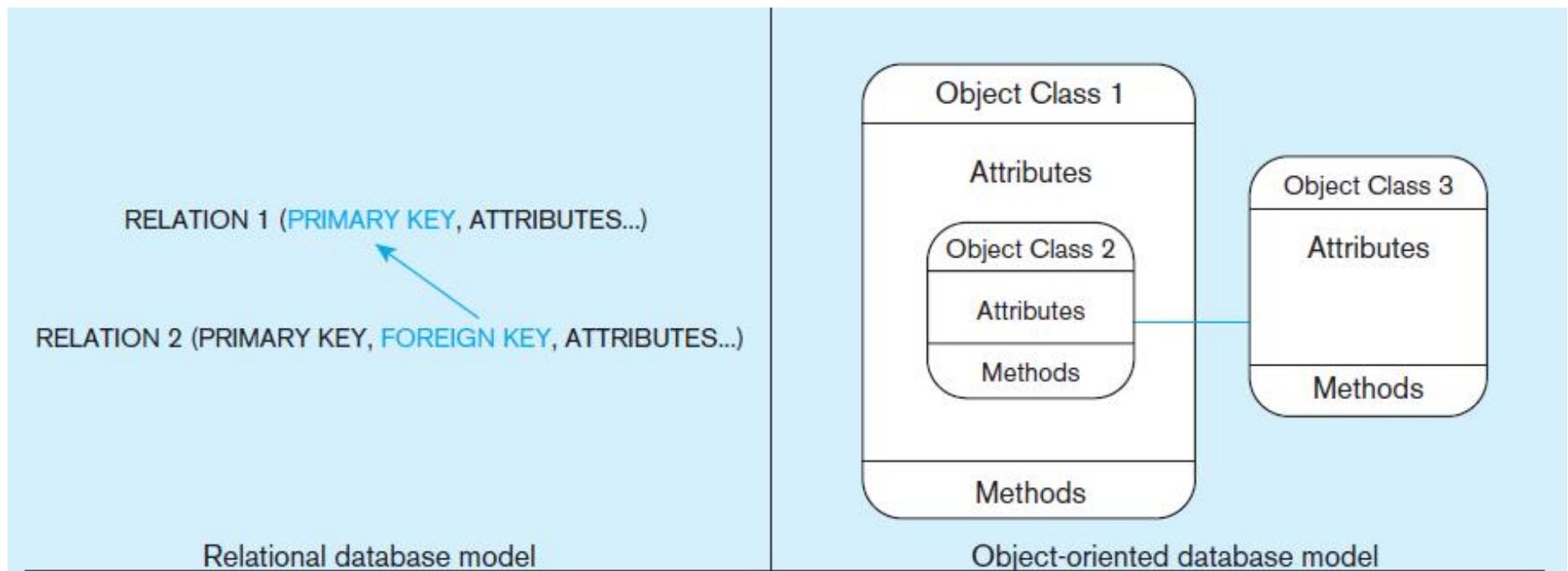


Figure 1-10: Database Architectures

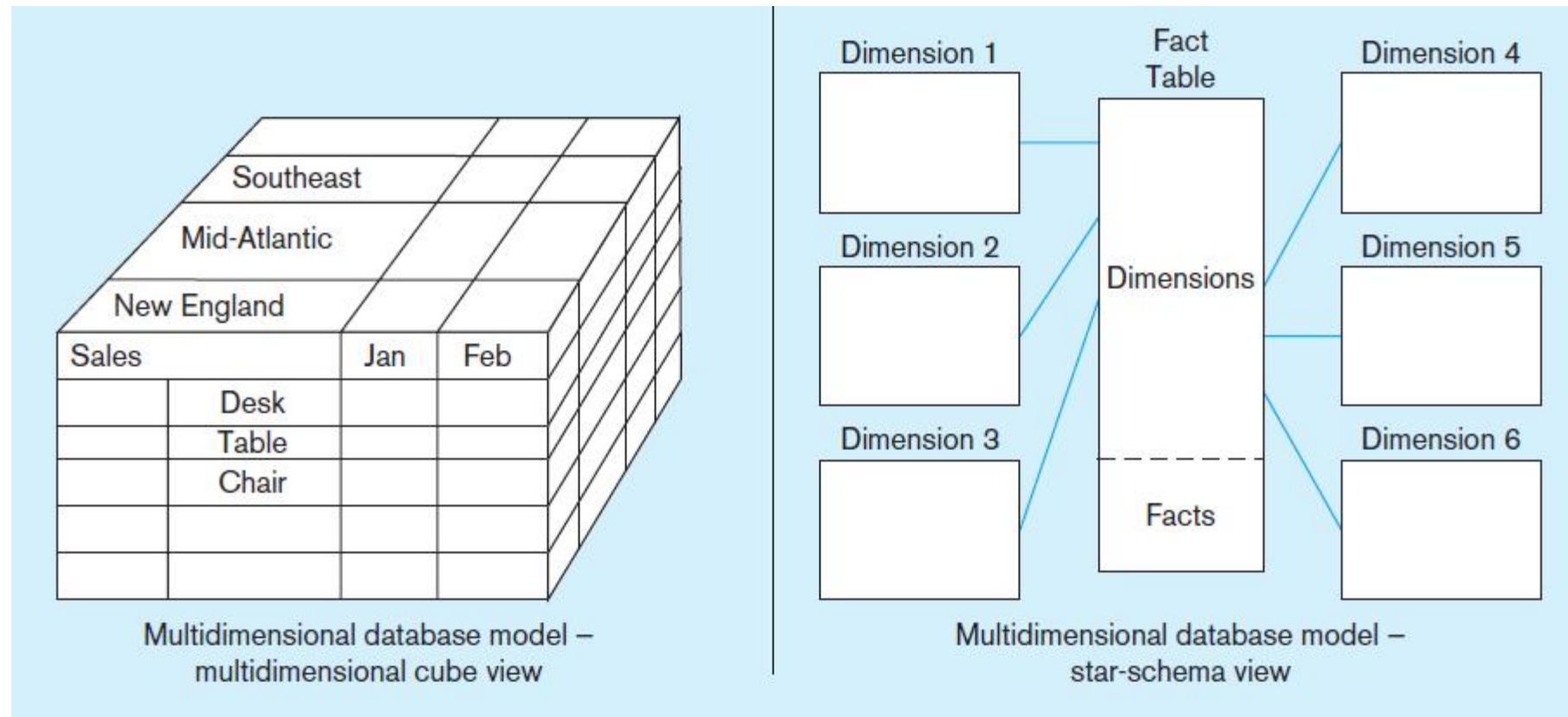


Table 1-5: The Range of Database Applications

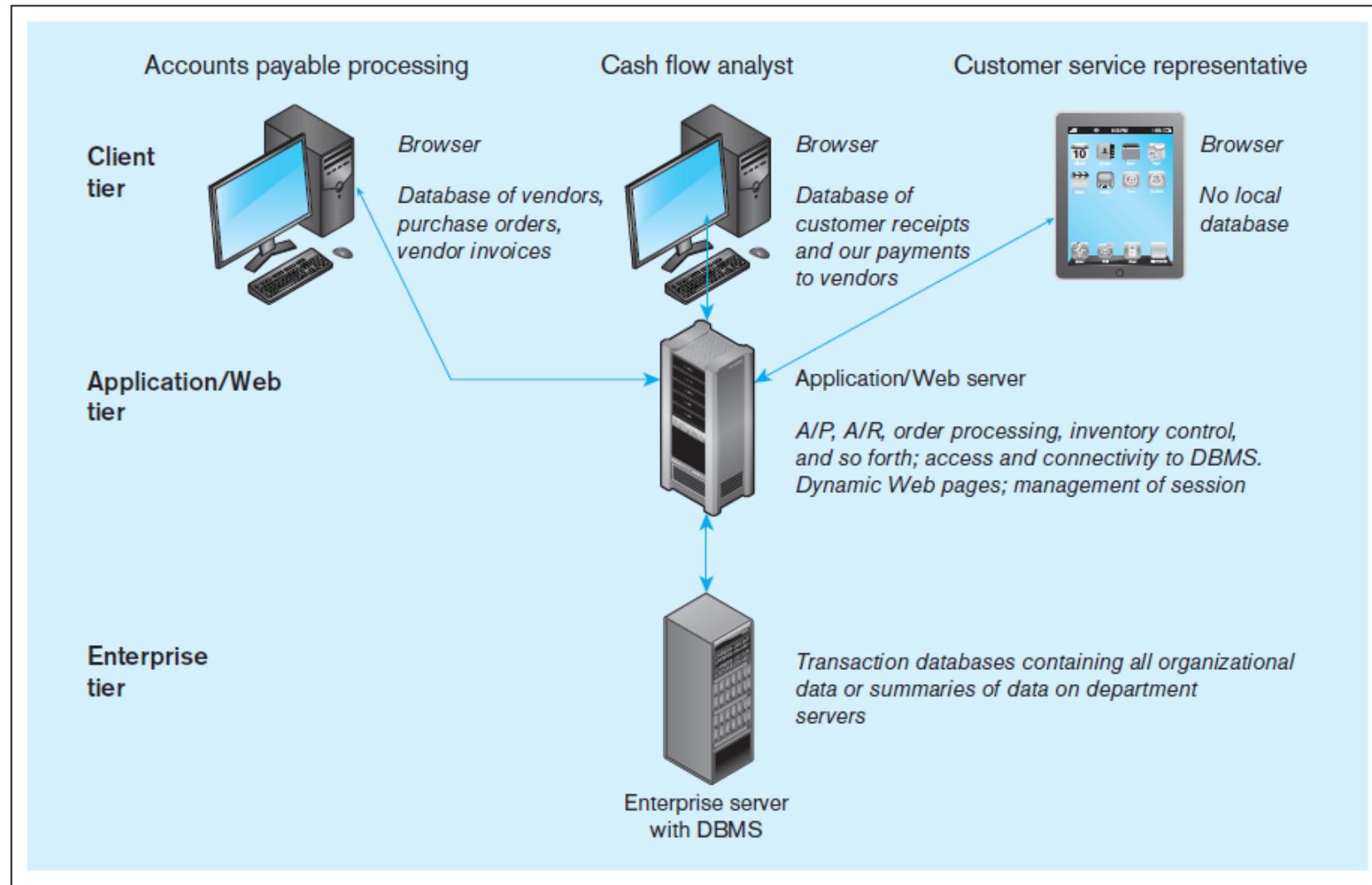
- Personal databases.
- Two-tier and n -tier client/server databases.
- Enterprise applications:
 - Enterprise resource planning (ERP) systems.
 - Data warehousing implementations.

TABLE 1-5 Summary of Database Applications

Type of Database / Application	Typical Number of Users	Typical Size of Database
Personal	1	Megabytes
Two-tier	5–100	Megabytes–gigabytes
Three-tier	100–1000	Gigabytes
Enterprise resource planning	>100	Gigabytes–terabytes
Data warehousing	>100	Terabytes–petabytes



Figure 1-11: Multi-Tiered Client/Server Database Architecture



Enterprise Database Applications

- **Enterprise Resource Planning (ERP):**

- Integrate all enterprise functions (manufacturing, finance, sales, marketing, inventory, accounting, human resources).

- **Data Warehouse (DW):**

- Integrated decision support system derived from various operational databases.



UNIVERSITY OF
MARYLAND

Figure 1-13: Computer System for Pine Valley Furniture Company

