INFORMATION MANAGEMENT

LESSON 1: THE DATABASE ENVIRONMENT

OBJECTIVES
- ✧ Definition of terms
- ✧ Explain growth and importance of databases
- ✧ Name limitations of conventional file processing
- ✧ Identify five categories of databases
- ✧ Explain advantages of databases
- ✧ Identify costs and risks of databases
- ✧ List components of database environment
- ✧ Describe evolution of database systems

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

Figure 1-1a Data in context

Class Roster

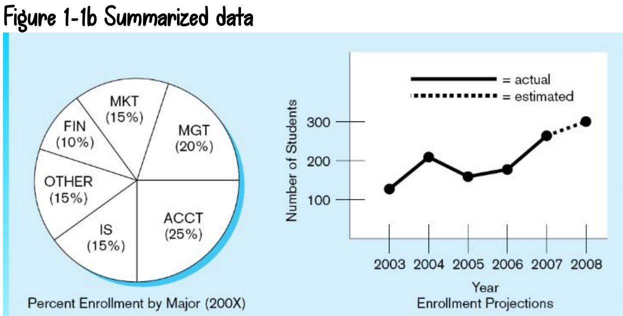
Course: MGT 500 Business Policy   Semester: Spring 200X

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

Context helps users understand data

Figure 1-1b Summarized data



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

Table 1.1 Example Metadata for Class Roster

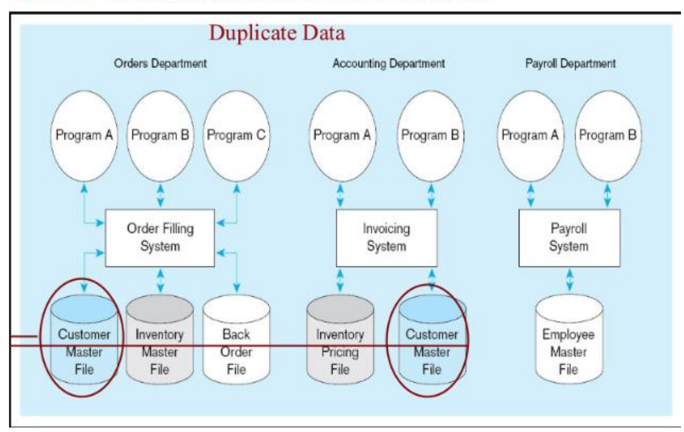
Data Item	Metadata					
Name	Type	Length	Min	Max	Description	Source
Course	Alphanumeric	30			Course ID and name	Academic Unit
Section	Integer	1	1	9	Section number	Registrar
Semester	Alphanumeric	10			Semester and year	Registrar
Name	Alphanumeric	30			Student name	Student IS
ID	Integer	9			Student ID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

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

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

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

Figure 1-3 Old file processing systems at Pine Valley Furniture Company

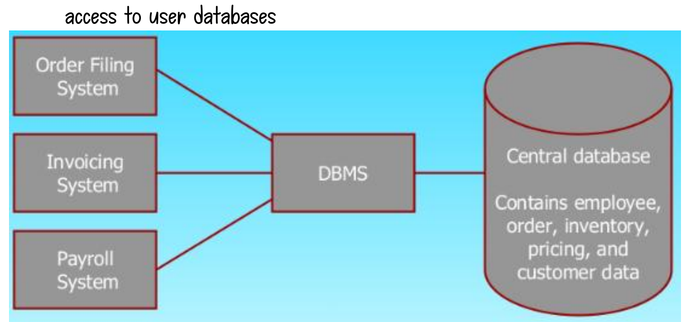


PROBLEMS WITH DATA REDUNDANCY
- ✧ Waste of space to have duplicate data
- ✧ Causes more maintenance headaches
- ✧ The biggest problem:
  - → Data changes in one file could cause Inconsistencies
  - → Compromises in data integrity

THE 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)

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

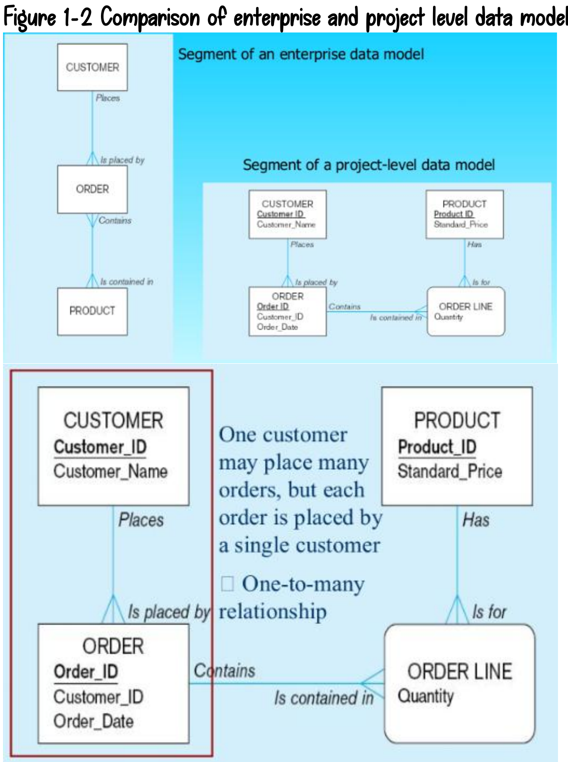
COST AND RISKS OF THE DATABASE APPROACH

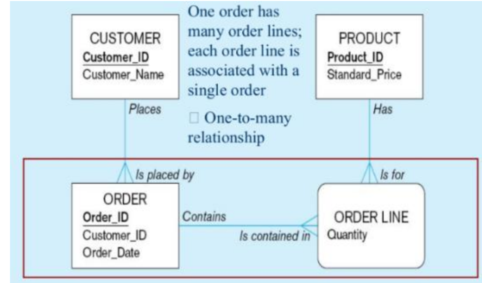
- New, specialized personnel
- Installation and management cost and complexity
- Conversion costs
- Need for explicit backup and recovery
- Organizational conflict

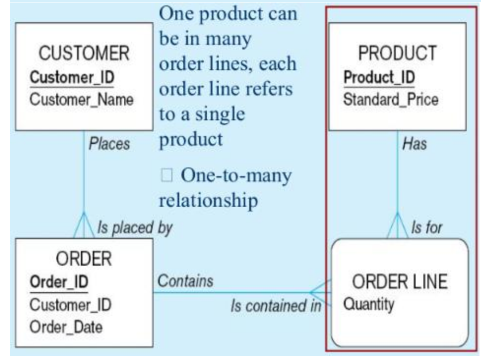
ELEMENTS OF THE DATABASE APPROACH

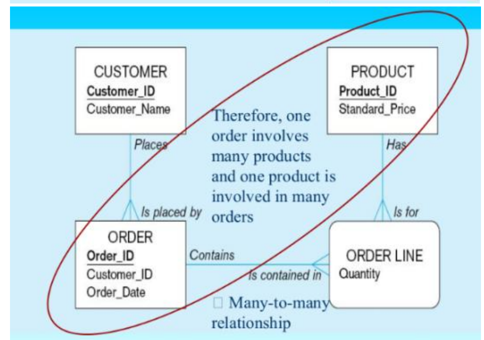
- **Data models**
  - → Graphical system capturing nature and relationship of data
  - → Enterprise Data Model-high-level entities and relationships for the organization
  - → Project Data Model-more detailed view, matching data structure in database or data warehouse
- **Relational Databases**
  - → Database technology involving tables (relations) representing entities and primary/foreign keys representing relationships
- **Use of Internet Technology**
  - → Networks and telecommunications, distributed databases, client-server, and 3-tier architectures
- **Database Applications**
  - → Application programs used to perform database activities (create, read, update, and delete) for database users

Figure 1-2 Comparison of enterprise and project level data models









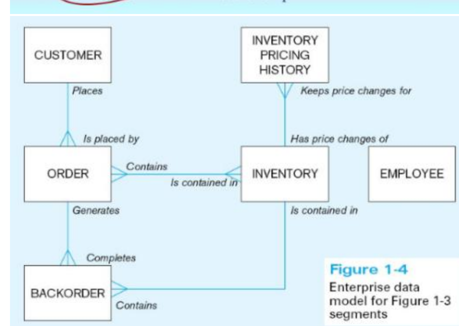
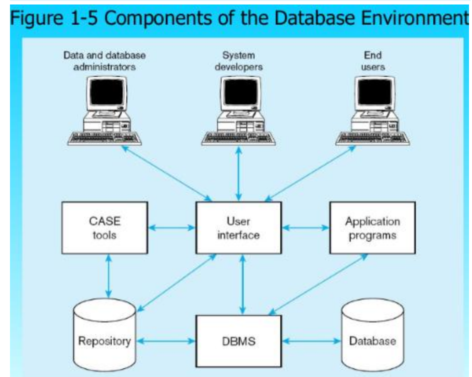


Figure 1-4 Enterprise data model for Figure 1-3 segments

Figure 1-5 Components of the Database Environment



COMPONENTS OF THE DATABASE ENVIRONMENT

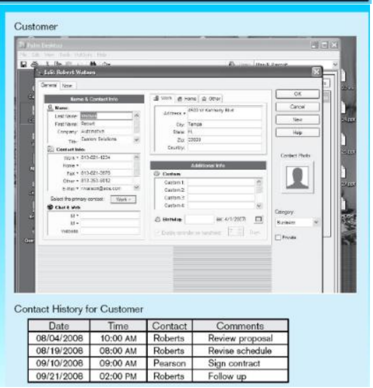
- **CASE Tools**-computer-aided software engineering
- **Repository**-centralized storehouse of metadata
- **Database Management System (DBMS)** -software for managing the database
- **Database**-storehouse of the data
- **Application Programs**-software using the data
- **User Interface**-text and graphical 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

THE RANGE OF DATABASE APPLICATION

- ✧ Personal databases
- ✧ Workgroup databases
- ✧ Departmental/divisional databases
- ✧ Enterprise database
- ✧ Enterprise resource planning (ERP) systems
- ✧ Data warehousing implementations

Table 1-6 Summary of Database Applications

Type of Database	Typical Number of Users	Typical Architecture	Typical Size of Database
Personal	1	Desktop/laptop computer, PDA	Megabytes
Workgroup	5–25	Client/server (two-tier)	Megabytes–gigabytes
Department/Division	25–100	Client/server (three-tier)	Gigabytes
Enterprise	>100	Client/server (distributed or parallel server)	Gigabytes–terabytes
Web-enabled	>1000	Web server and application servers	Megabytes–gigabytes



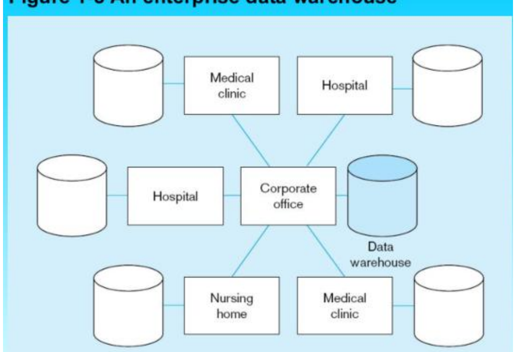
Date	Time	Contact	Comments
08/04/2008	10:00 AM	Roberts	Review proposal
08/19/2008	08:00 AM	Roberts	Revise schedule
09/10/2008	09:00 AM	Pearson	Sign contract
09/21/2008	02:00 PM	Roberts	Follow up

Figure 1-6 Typical data from a personal database

ENTERPRISE DATABASE APPLICATIONS

- ✧ **Enterprise Resource Planning (ERP)**
  - → Integrate all enterprise functions (manufacturing, finance, sales, marketing, inventory, accounting, human resources)
- ✧ **Data Warehouse**
  - → Integrated decision support system derived from various operational databases

Figure 1-8 An enterprise data warehouse



WEB-ENABLED DATABASES

Web applications requiring databases

- ✧ Customer relationship management (CRM)
- ✧ Business-to-consumer (B2C)
- ✧ Electronic data interchange (EDI)
- ✧ Private intranets
- ✧ XML-defined Web services

Issues to consider

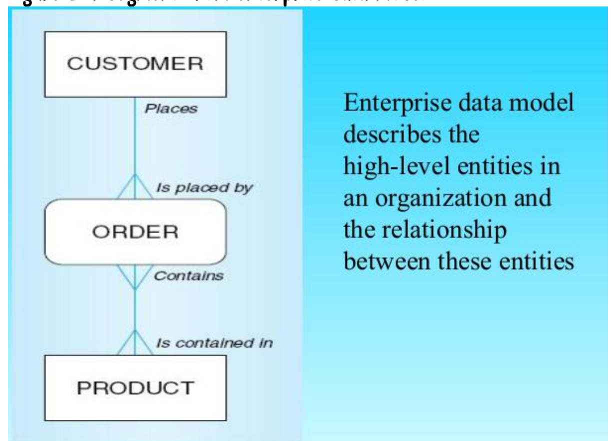
- ✧ Which technologies to use?
  - → Security/privacy protection
  - → Managing huge volumes of data from Internet transactions
  - → Maintaining data quality

LESSON 2: THE DATABASE DEVELOPMENT PROCESS

ENTERPRISE DATA MODEL

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

Figure 2-1 Segment from enterprise data model



INFORMATION SYSTEMS ARCHITECTURE (ISA)

- ✧ Conceptual blueprint for organization's desired information systems structure
- ✧ Consists of:
  - → Data (e.g. Enterprise Data Model-simplified ER Diagram)
  - → Processes-data flow diagrams, process decomposition, etc.
  - → Data Network-topology diagram (like Fig 1-7)
  - → People-people management using project management tools (Gantt charts, etc.)
  - → Events and points in time (when processes are performed)
  - → Reasons for events and rules (e.g., decision tables)

INFORMATION ENGINEERING

- ✧ A data-oriented methodology to create and maintain information systems
- ✧ Top-down planning-a generic IS planning methodology for obtaining a broad understanding of the IS needed by the entire organization
- ✧ Four steps to Top-Down planning:
  - → Planning
  - → Analysis
  - → Design
  - → Implementation

Table 2-1 Information Engineering Planning Phase

Step	Explanation
1.	Identify strategic planning factors
	a. Goals
	b. Critical success factors
	c. Problem areas
2.	Identify corporate planning objects
	a. Organizational units
	b. Locations
	c. Business functions
	d. Entity types
3.	Develop an enterprise model
	a. Functional decomposition
	b. Entity-relationship diagram
	c. Planning matrixes

IDENTIFY STRATEGIC PLANNING FACTORS

- ✧ Organization goals-what we hope to accomplish
- ✧ Critical success factors-what MUST work in order for us to survive
- ✧ Problem areas-weaknesses we now have

IDENTIFY CORPORATE PLANNING OBJECTS

- ✧ Organizational units-departments
- ✧ Organizational locations
- ✧ Business functions-groups of business processes
- ✧ Entity types-the things we are trying to model for the database
- ✧ Information systems-application programs

DEVELOP ENTERPRISE MODEL

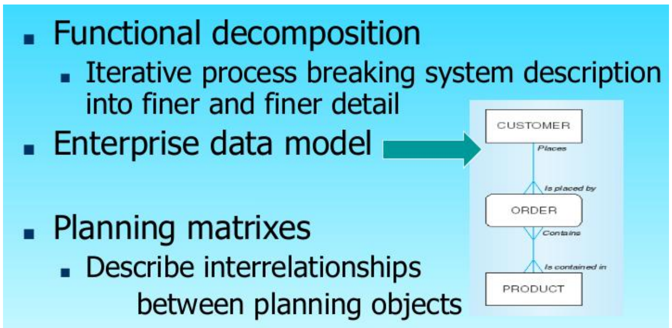
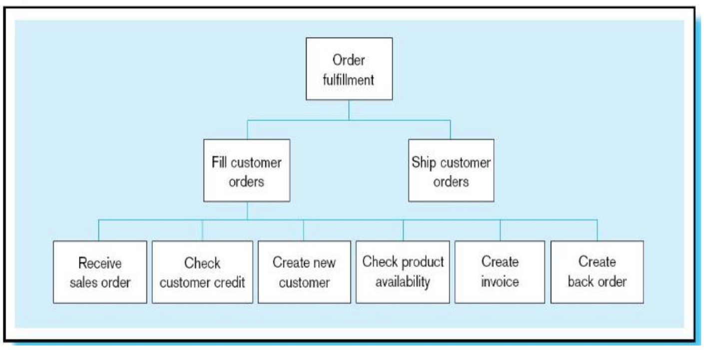


Figure 2-2 Example process decomposition of an order fulfillment function (Pine Valley Furniture Company)



PLANNING MATRIXES

- ✧ Describe relationships between planning objects in the organization
- ✧ Types of matrixes:
- ✧ Location-to-function
- ✧ Unit-to-function
- ✧ IS-to-data entity
- ✧ Supporting function-to-data entity
- ✧ IS-to-business objective

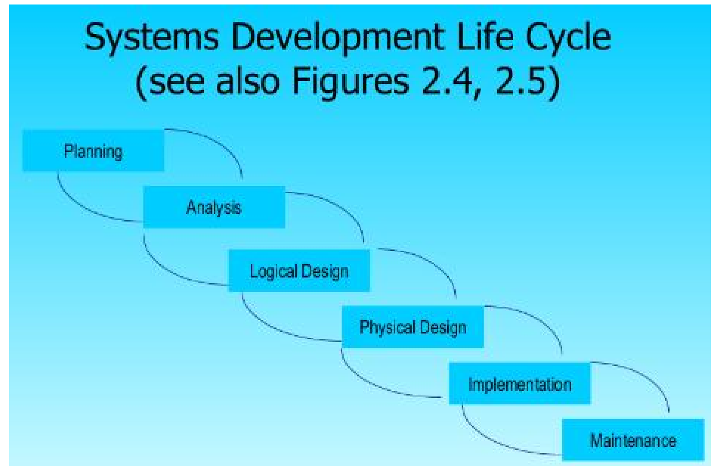
Function-to-Data Entity Matrix (Fig. 2-3)

Business Functions \ Data Entity Types	Customer	Product	Raw Material	Order	Work Center	Work Order	Invoice	Equipment	Employee
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 (column) is used within business function (row)

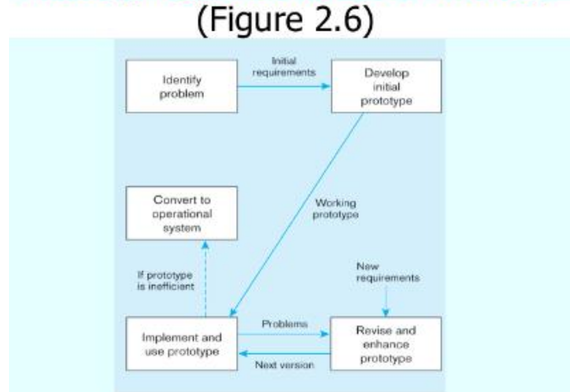
TWO APPROACHES TO DATABASE AND IS DEVELOPMENT

- ✧ 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

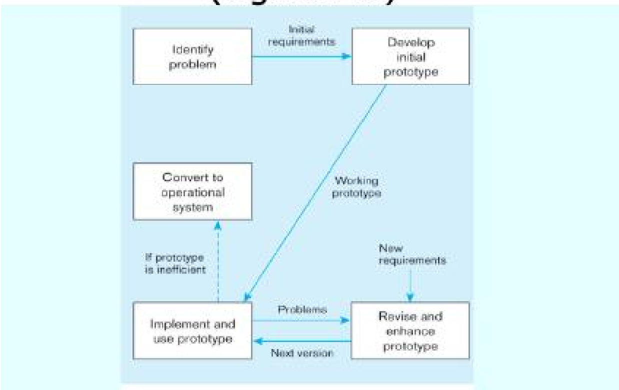


- ✧ PLANNING
  - → Purpose - preliminary understanding
  - → Deliverable - request for study
  - → Database activity - enterprise modeling and early conceptual data modeling
- ✧ ANALYSIS
  - → Purpose-thorough requirements analysis and structuring
  - → Deliverable-functional system specifications
  - → Database activity-thorough and integrated conceptual data modeling
- ✧ LOGICAL DESIGN
  - → Purpose-information requirements elicitation and structure
  - → Deliverable-detailed design specifications
  - → Database activity- logical database design (transactions, forms, displays, views, data integrity and security)
- ✧ PHYSICAL DESIGN
  - → Purpose-develop technology and organizational specifications
  - → Deliverable-program/data structures, technology purchases, organization redesigns
  - → Database activity-physical database design (define database to DBMS, physical data organization, database processing programs)
- ✧ IMPLEMENTATION
  - → Purpose-programming, testing, training, installation, documenting
  - → Deliverable-operational programs, documentation, training materials
  - → Database activity-database implementation, including coded programs, documentation, installation and conversion
- ✧ MAINTENANCE
  - → Purpose-monitor, repair, enhance
  - → Deliverable-periodic audits
  - → Database activity-database maintenance, performance analysis and tuning, error corrections

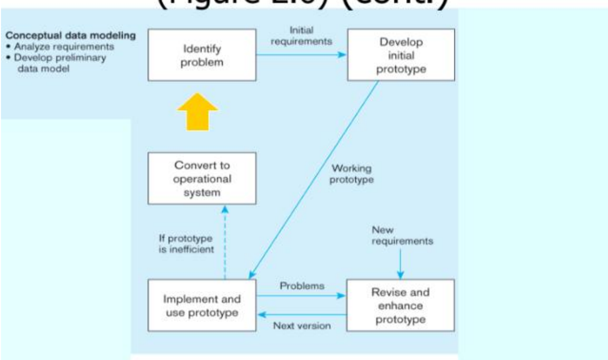
Prototyping Database Methodology (Figure 2.6)



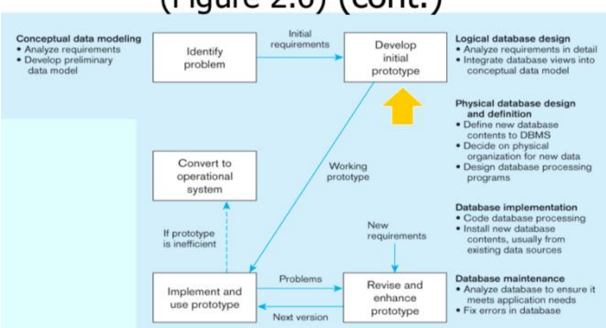
Prototyping Database Methodology (Figure 2.6)



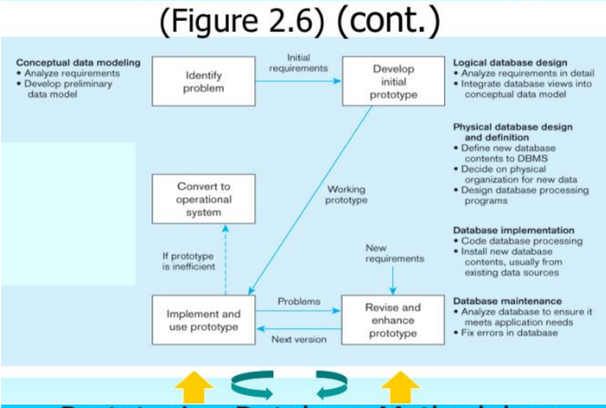
Prototyping Database Methodology (Figure 2.6) (cont.)



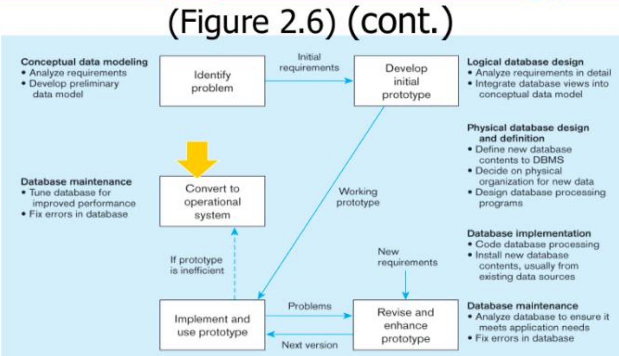
Prototyping Database Methodology (Figure 2.6) (cont.)



Prototyping Database Methodology (Figure 2.6) (cont.)



Prototyping Database Methodology (Figure 2.6) (cont.)



CASE

- Computer-Aided Software Engineering (CASE)-software tools providing automated support for systems development

Three database features:

- Data modeling-drawing entity-relationship diagrams
- Code generation-SQL code for table creation
- Repositories-knowledge base of enterprise information

PACKAGED DATA MODELS

- Model components that can be purchased, customized, and assembled into full-scale data models

Advantages

- Reduced development time
- Higher model quality and reliability

Two types:

- Universal data models
- Industry-specific data models

MANAGING PROJECTS

- Project-a planned undertaking of related activities to reach an objective that has a beginning and an end
- Involves use of review points for:
  - → Validation of satisfactory progress
  - → Step back from detail to overall view
  - → Renew commitment of stakeholders
  - → Incremental commitment-review of systems development project after each development phase with rejustification after each phase

MANAGING PROJECTS: PEOPLE INVOLVED

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

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
  - → E-R models-covered in Chapters 3 and 4
- Internal Schema
  - → Logical structures-covered in Chapter 5
  - → Physical structures-covered in Chapter 6

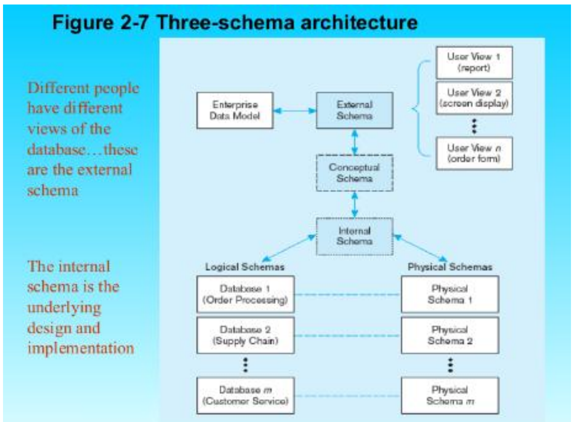


Figure 2-8 Process of developing three-schema architecture for a database project

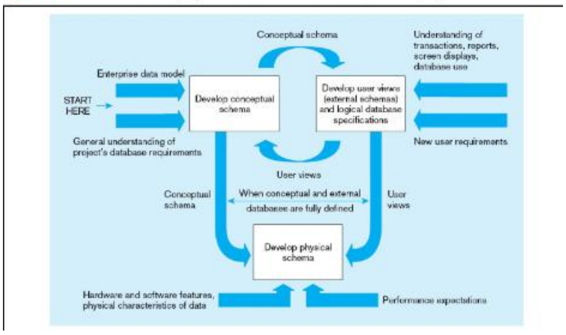
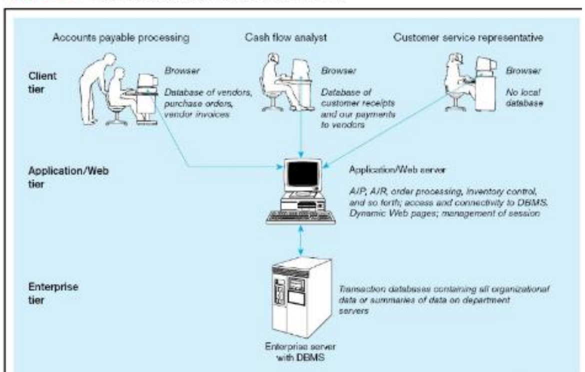
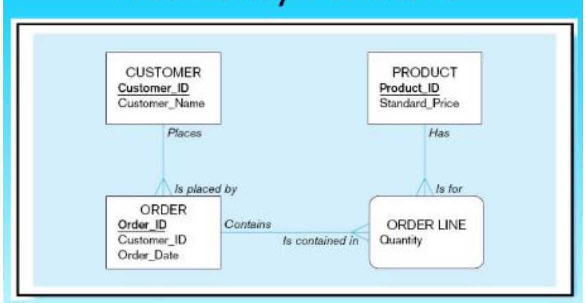


Figure 2-9 Three-tiered client/server database architecture

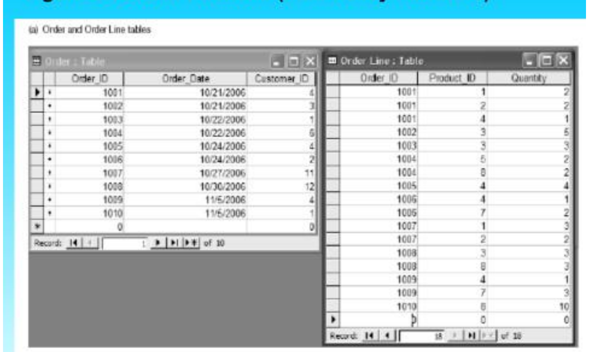


Pine Valley Furniture



Segment of project data model (Figure 2-11)

Figure 2-12 Four relations (Pine Valley Furniture)



(a) Order and Order Line tables

Order : Table

Order_ID	Order_Date	Customer_ID
1001	10/21/2006	4
1002	10/21/2006	3
1003	10/22/2006	1
1004	10/22/2006	6
1005	10/24/2006	4
1006	10/24/2006	2
1007	10/27/2006	11
1008	10/30/2006	12
1009	11/5/2006	4
1010	11/6/2006	1
0		0

Record: 1 of 10

Order Line : Table

Order_ID	Product_ID	Quantity
1001	1	2
1001	2	2
1001	4	1
1002	3	5
1003	3	3
1004	5	2
1004	8	2
1005	4	4
1006	4	1
1006	7	2
1007	1	3
1007	2	2
1008	3	3
1008	8	3
1009	4	1
1009	7	3
1010	8	10
0	0	0

Record: 18 of 18

Figure 2-12 Four relations (Pine Valley Furniture) (cont.)

(b) Customer table

Customer : Table

Customer_ID	Customer_Name
1	Contemporary Casuals
2	Value Furniture
3	Home Furnishings
4	Eastern Furniture
5	Impressions
6	Furniture Gallery
7	Period Furniture
8	California Classics
9	M and H Casual Furniture
10	Seminole Interiors
11	American Euro Lifestyles
12	Battle Creek Furniture
13	Heritage Furnishings
14	Kaneohe Homes
15	Mountain Scenes

Record: 16 of 15

(c) Product table

Product : Table

Product_ID	Standard_Price
1	$175.00
2	$200.00
3	$375.00
4	$650.00
5	$325.00
6	$750.00
7	$150.00
8	$250.00
	$0.00

Record: 9

LESSON 3:MODELING DATA IN THE ORGANIZATION

BUSINESS RULES

- Statements that define or constrain some aspect of the business
- Assert business structure
- Control/influence business behavior
- Expressed in terms familiar to end users
- Automated through DBMS software

A GOOD BUSINESS RULE IS:

- **Declarative**-what, not how
- **Precise**-clear, agreed-upon meaning
- **Atomic**-one statement
- **Consistent**-internally and externally
- **Expressible**-structured, natural language
- **Distinct**-non-redundant
- **Business-oriented**-understood by business people

A GOOD DATA NAME IS:

- Related to business, not technical, characteristics
- Meaningful and self-documenting
- Unique
- Readable
- Composed of words from an approved list
- Repeatable
- Follows standard syntax

DATA DEFINITIONS

- Explanation of a term or fact
  - → Term-word or phrase with specific meaning
  - → Fact-association between two or more terms
- Guidelines for good data definition
  - → Gathered in conjunction with systems requirements
  - → Accompanied by diagrams
  - → Concise description of essential data meaning
  - → Achieved by consensus, and iteratively refine

E-R MODEL CONSTRUCTS

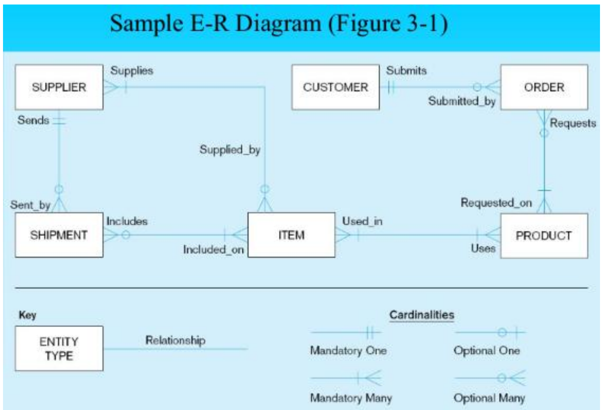
ENTITIES:

- **Entity instance**-person, place, object, event, concept (often corresponds to a row in a table)
- **Entity Type**-collection of entities (often corresponds to a table)

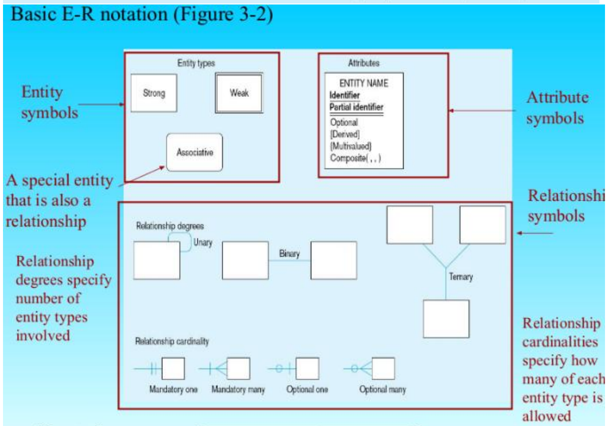
RELATIONSHIPS:

- **Relationship instance**-link between entities (corresponds to primary key-foreign key equivalencies in related tables)
- **Relationship type**-category of relationship…link between entity types

ATTRIBUTE-property or characteristic of an entity or relationship type (often corresponds to a field in a table)



Sample E-R Diagram (Figure 3-1)



Basic E-R notation (Figure 3-2)

WHAT SHOULD AN ENTITY BE?

SHOULD BE:

- ✧ An object that will have many instances in the database
- ✧ An object that will be composed of multiple attributes
- ✧ An object that we are trying to model

SHOULD NOT BE:

- ✧ A user of the database system
- ✧ An output of the database system (e.g., a report)

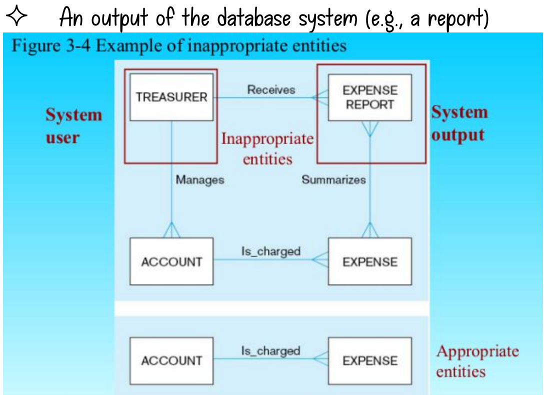


Figure 3-4 Example of inappropriate entities

ATTRIBUTES

- ✧ **Attribute** – property or characteristic of an entity or relationship type

CLASSIFICATIONS OF ATTRIBUTES:

- ✧ Required versus Optional Attributes
- ✧ Simple versus Composite Attribute
- ✧ Single-Valued versus Multivalued Attribute
- ✧ Stored versus Derived Attributes
- ✧ Identifier Attributes

IDENTIFIER (KEY)

- ✧ Identifier (Key) – an attribute (or combination of attributes) that uniquely identifies individual instances of an entity type
- ✧ Simple versus Composite Identifier
- ✧ Candidate Identifier-an attribute that could be a key…satisfies the requirements for being an identifier

CHARACTERISTICS OF IDENTIFIERS

- ✧ Will not change in value
- ✧ Will not be null
- ✧ No intelligent identifiers (e.g., containing locations or people that might change)
- ✧ Substitute new, simple keys for long, composite keys

Figure 3-7 A **composite** attribute

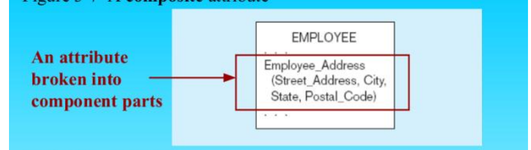


Figure 3-8 Entity with **multivalued** attribute (Skill) and **derived** attribute (Years_Employed)

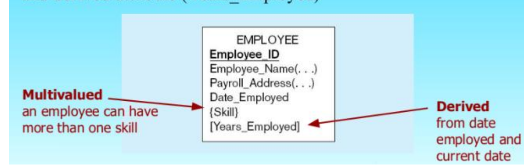


Figure 3-9 Simple and composite identifier attributes

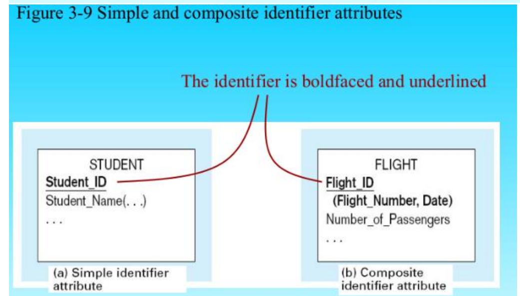


Figure 3-19 Simple example of time-stamping

MORE ON RELATIONSHIPS

RELATIONSHIP TYPES VS. RELATIONSHIP INSTANCES

- ✧ The relationship type is modeled as lines between entity types…the instance is between specific entity instances

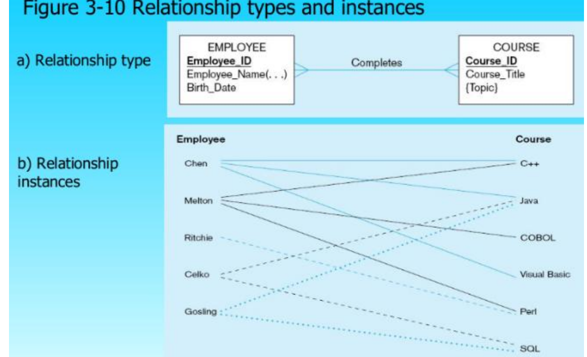
RELATIONSHIPS CAN HAVE ATTRIBUTES

- ✧ These describe features pertaining to the association between the entities in the relationship

Two entities can have more than one type of relationship between them (multiple relationships)

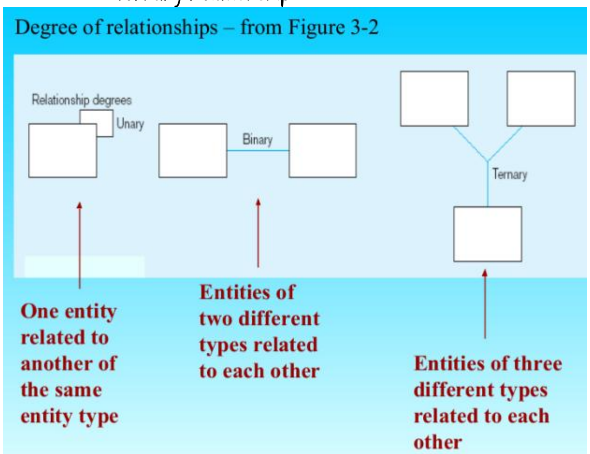
- ✧ **ASSOCIATIVE ENTITY**-combination of relationship and entity

Figure 3-10 Relationship types and instances



DEGREE OF RELATIONSHIPS

✧ Degree of a relationship is the number of entity types that participate in it
  - → Unary Relationship
  - → Binary Relationship
  - → Ternary Relationship



CARDINALITY OF RELATIONSHIPS

✧ One-to-One
  - → Each entity in the relationship will have exactly one related entity

✧ One-to-Many
  - → An entity on one side of the relationship can have many related entities, but an entity on the other side will have a maximum of one related entity

✧ Many-to-Many
  - → Entities on both sides of the relationship can have many related entities on the other side

CARDINALITY CONSTRAINTS

✧ Cardinality Constraints–the number of instances of one entity that can or must be associated with each instance of another entity

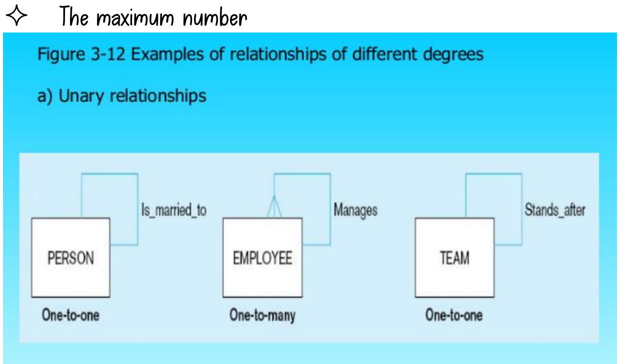
MINIMUM CARDINALITY

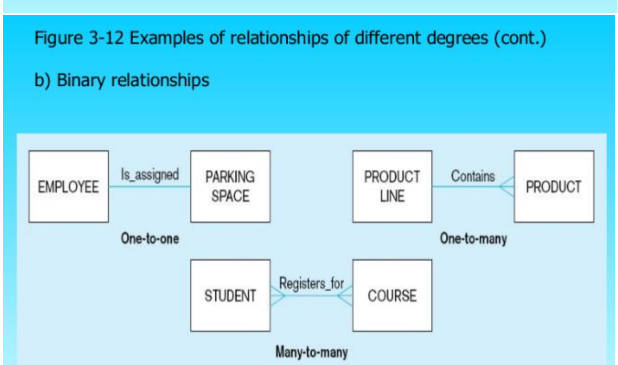
✧ If zero, then optional

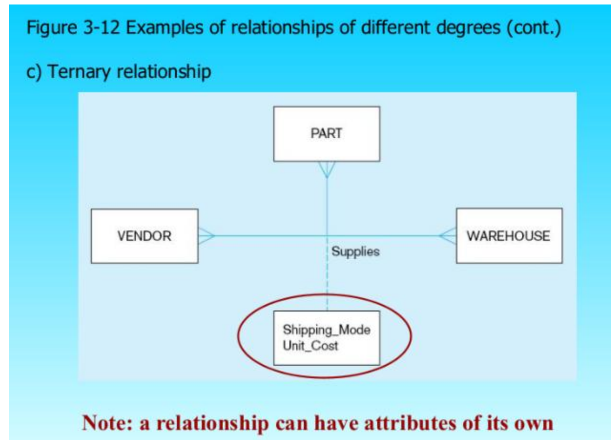
✧ If one or more, then mandatory

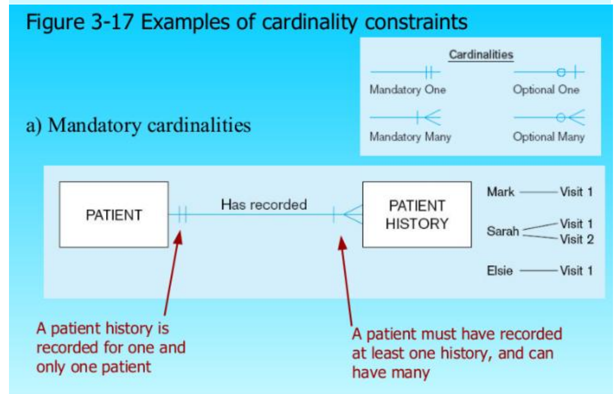
MAXIMUM CARDINALITY

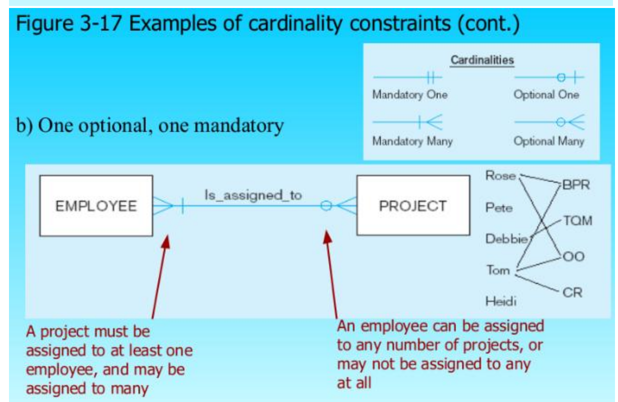
✧ The maximum number

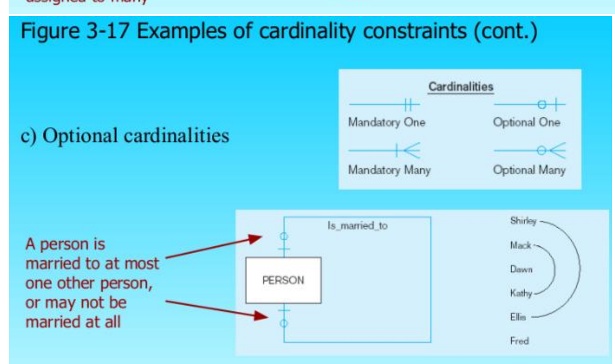












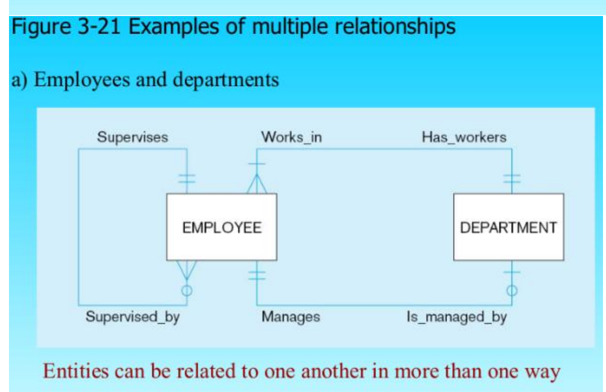
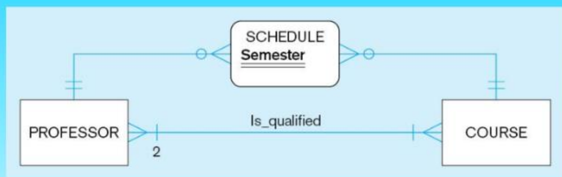


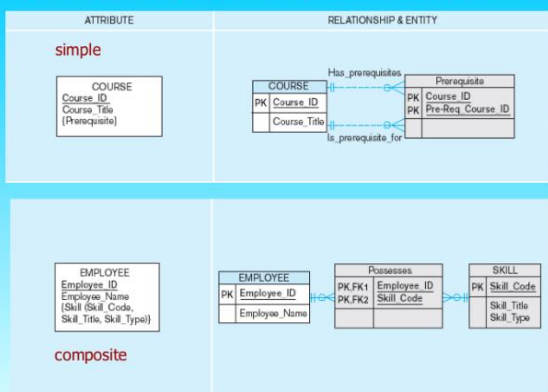
Figure 3-21 Examples of multiple relationships (cont.)

b) Professors and courses (fixed lower limit constraint)



Here, min cardinality constraint is 2

Figure 3-15a and 3-15b Multivalued attributes can be represented as relationships



STRONG VS. WEAK ENTITIES, AND IDENTIFYING RELATIONSHIPS

- ✧ **Strong entities**
  - → exist independently of other types of entities
  - → has its own unique identifier
  - → identifier underlined with single line
- ✧ **Weak entity**
  - → dependent on a strong entity (identifying owner)…cannot exist on its own
  - → does not have a unique identifier (only a partial identifier)
  - → partial identifier underlined with double line
  - → entity box has double line
- ✧ **Identifying relationship**
  - → links strong entities to weak entities

Identifying relationship (Figure 3-5)

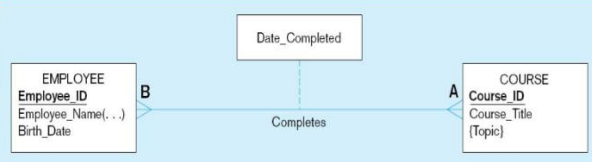
Strong entity

Weak entity

ASSOCIATIVE ENTITIES

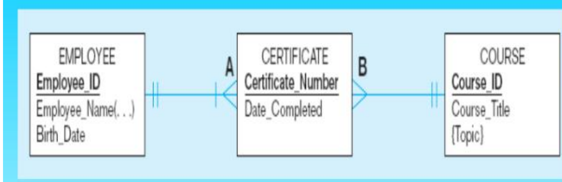
- ✧ An entity–has attributes
- ✧ A relationship–links entities together
- ✧ When should a relationship with attributes instead be an associative entity?
- ✧ All relationships for the associative entity should be many
- ✧ The associative entity could have meaning independent of the other entities
- ✧ The associative entity preferably has a unique identifier, and should also have other attributes
- ✧ The associative entity may participate in other relationships other than the entities of the associated relationship
- ✧ Ternary relationships should be converted to associative entities

Figure 3-11a A binary relationship with an attribute



Here, the date completed attribute pertains specifically to the employee's completion of a course…it is an attribute of the *relationship*

Figure 3-11b An associative entity (CERTIFICATE)



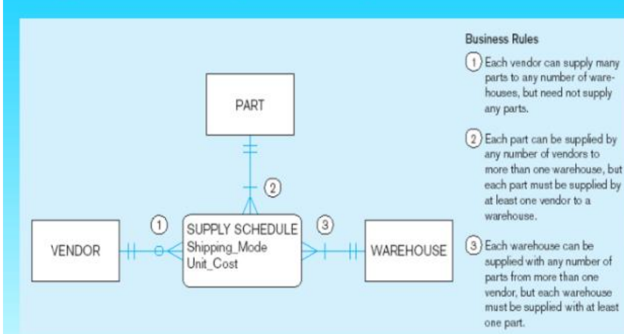
Associative entity is like a relationship with an attribute, but it is also considered to be an entity in its own right

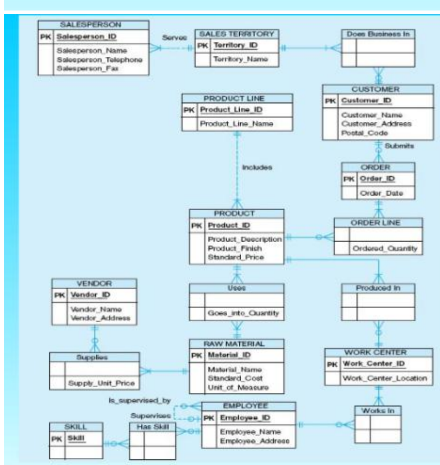
Note that the many-to-many cardinality between entities in Figure 3-11a has been replaced by two one-to-many relationships with the associative entity

Figure 3-13c An associative entity – bill of materials structure

This could just be a relationship with attributes…it's a judgment call

Figure 3-18 Ternary relationship as an associative entity





Microsoft Visio Notation for Pine Valley Furniture E-R diagram

(Figure 3-22)

Different modeling software tools may have different notation for the same constructs