1.What is information systems analysis and design?

2.What is systems thinking? How is it useful for thinking about computer-based information systems?

3.What is decomposition? Coupling? Cohesion?

4.In what way are organizations systems?

5.List and explain the different phases in the systems development life cycle.

6.What is prototyping?

7.What are CASE tools? What is a CASE repository and how is it used?

8. What is JAD? What is Participatory Design?

9. What is RAD? How does it compare to the typical SDLC?

10. What are Agile Methodologies?

UML questions

Compare and contrast the object-oriented analysis and design models with structured analysis and design models.

The biggest difference between the models is that the object-oriented models build on each other and are consistent. Models developed during structured analysis and design are weakly connected and lack a common representation.

2. Give an example of an abstract use case. Your example should involve at least two other use cases and show how they are related to the abstract use case.

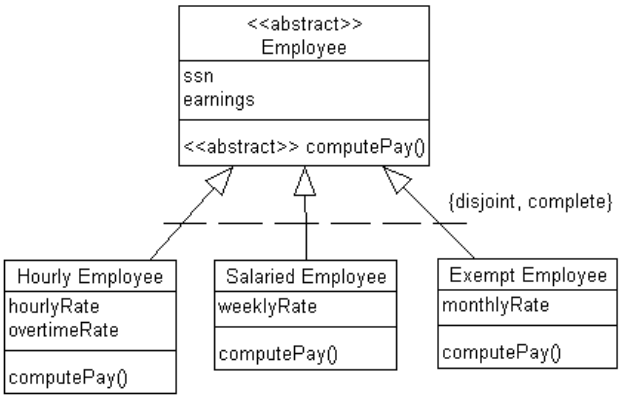
In Figure A–2, the Track Sales and Inventory Data use case is an example of an abstract use case.

3. Explain the use of association role for an association on a class diagram.

The association role is the end of an association where it connects to a class. The role can be explicitly named, indicating the role played by the class attached to the end at which the name appears. A role will have multiplicity; multiplicity gives an indication of how many objects participate in the relationship.

4. **Give an example of generalization. Your example should include at least one superclass and three subclasses, and a minimum of one attribute and one operation for each of the classes. Indicate the discriminator and specify the semantic constraints among the subclasses.**

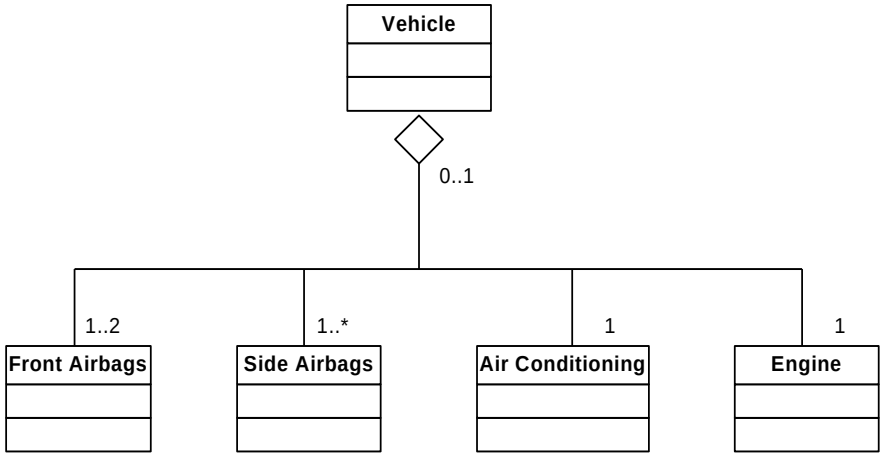
   A suggested answer is provided below



5. **Give an example of aggregation. Your example should include at least one aggregate object and three component objects. Specify the multiplicities at each end of all the aggregation relationships.**

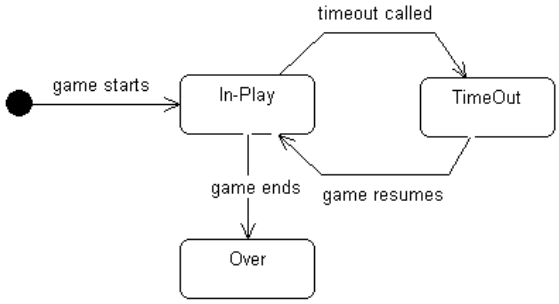
   One example of aggregation is a vehicle. A car will have front and side air bags, an air conditioner, an engine, and other components. A suggested diagram is provided below.

Problem and Exercise #5



6. **Give an example of state transition. Your example should show how the state of the object undergoes a transition based on some event.**

A suggested answer is provided below.



1. The use-case diagram shown in the figure 1 below captures the Student billing function but does not contain any function for accepting tuition payment from students. Revise the diagram to capture this functionality. Also, express some common behavior among two use cases in the revised diagram by using "include" relationships.

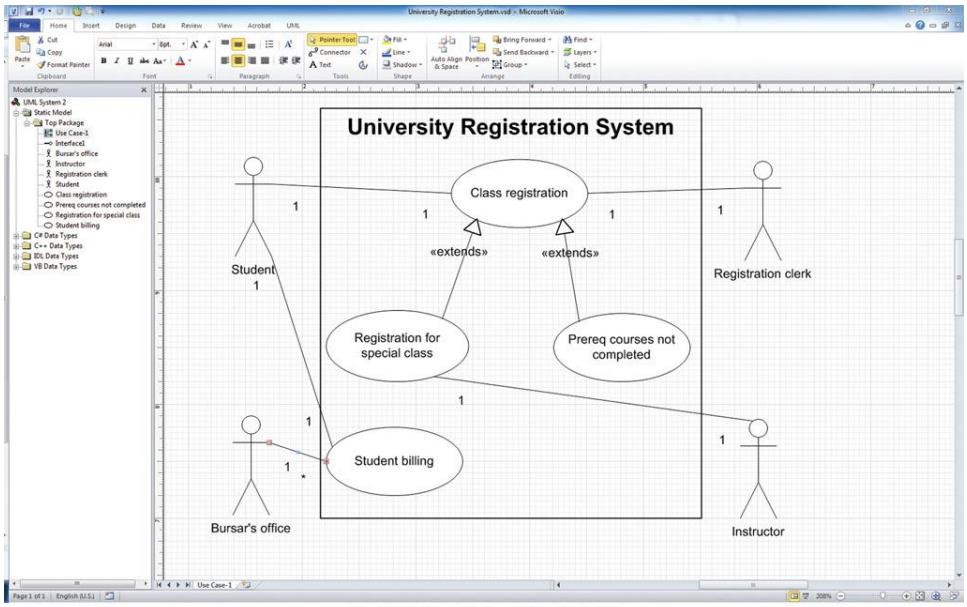
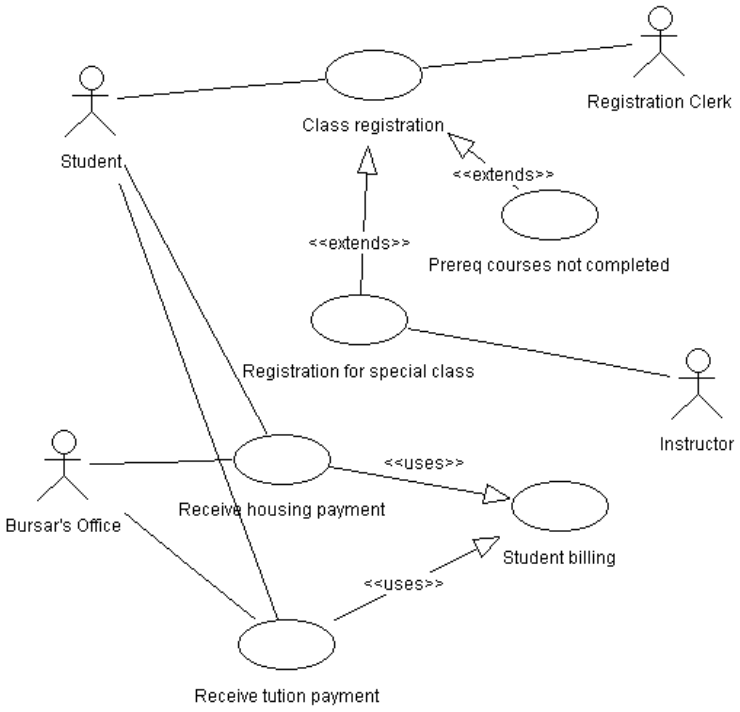


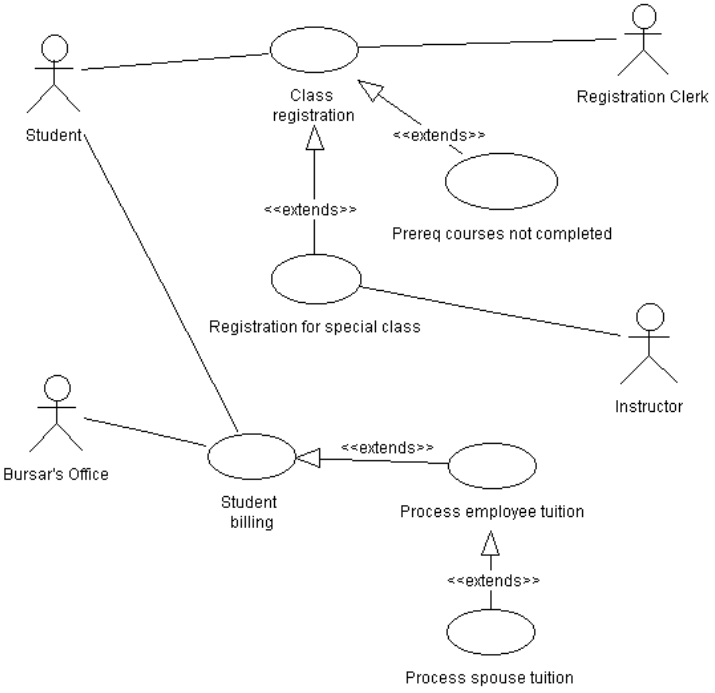
Figure 1

A suggested answer is provided below.



2. Suppose that the employees of the university are not billed for tuition. Their spouses do not get a full-tuition waiver, but pay for only 25 percent of the total tuition. Extend the use-case diagram of Figure 1 to capture these situations.

A suggested answer is provided below.



3. An organization has been entrusted with developing a Registration and Title system that maintains information about all vehicles registered in a particular state. For each vehicle that is registered with the office, the system has to store the name, address, telephone number of the owner, the start date and end date of the registration, plate information (issuer, year, type, and number), sticker (year, type, and number), and registration fee. In addition, the following information is maintained about the vehicles themselves: the number, year, make, model, body style, gross weight, number of passengers, diesel powered (yes/no), color, cost, and mileage. If the vehicle is a trailer, diesel powered and number of passengers are not relevant. For travel trailers, the body number and length must be known. The system needs to maintain information on the luggage capacity for a car, maximum cargo capacity and maximum towing capacity for a truck, and horsepower for a motorcycle. The system issues registration notices to owners of vehicles whose registrations are due to expire after two months. When the owner reviews the registration, the system updates the registration information on the vehicle.

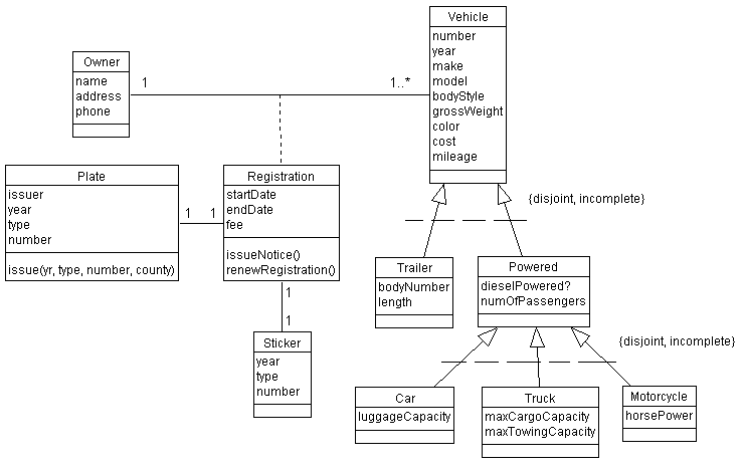
a. Develop a static object model by drawing a class diagram that shows all the object classes, attributes, operations, relationships, and multiplicities. For each operation, show its argument list.

b. Draw a state diagram that captures all the possible states of a Vehicle object, right from the time the vehicle was manufactured until it goes to the junkyard. In drawing the diagram, you may make any necessary assumptions, as long as they are realistic.

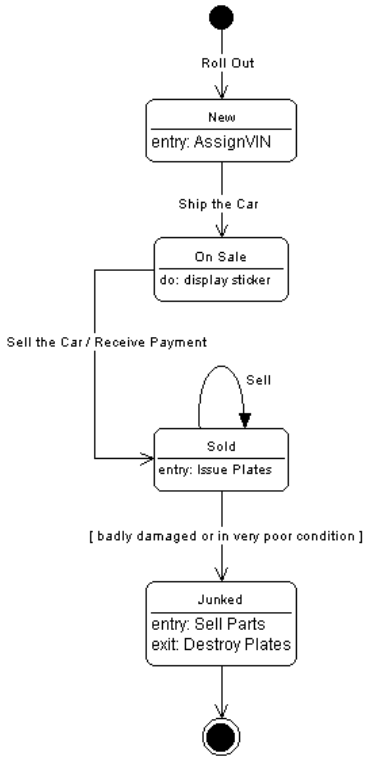
c. Select any state or event from the high-level state diagram that you have drawn and show its fine structure (substates and their transitions) in a lower-level diagram.

Suggested answers for parts a, b, c, are provided below.

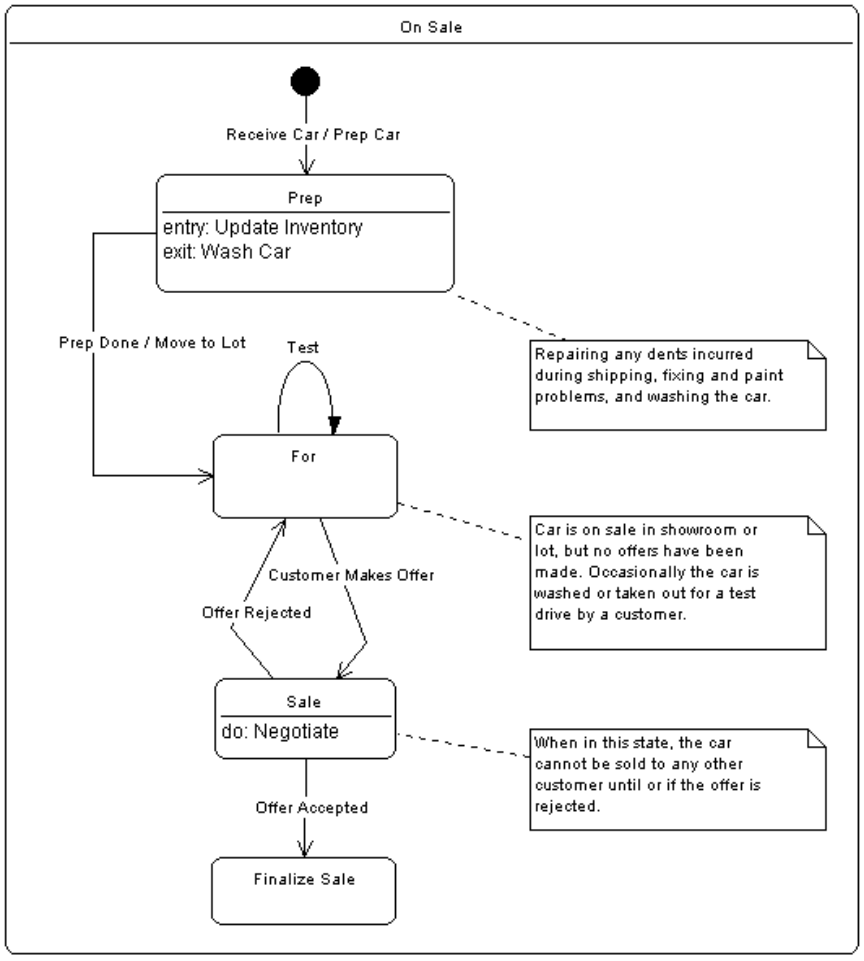
Part a


Vehicle
number
year
make
model
bodyStyle
grossWeight
color
cost
mileage
Owner
name
address
phone
1
1..*
Plate
issuer
year
type
number
issue(yr, type, number, county)
1
1
Registration
startDate
endDate
fee
issueNotice()
renewRegistration()
1
1
Sticker
year
type
number
{disjoint, incomplete}
Trailer
bodyNumber
length
Powered
dieselPowered?
numOfPassengers
{disjoint, incomplete}
Car
luggageCapacity
Truck
maxCargoCapacity
maxTowingCapacity
Motorcycle
horsePower

Part b


Roll Out
New
entry: AssignVIN
Ship the Car
On Sale
do: display sticker
Sell the Car / Receive Payment
Sell
Sold
entry: Issue Plates
[badly damaged or in very poor condition]
Junked
entry: Sell Parts
exit: Destroy Plates

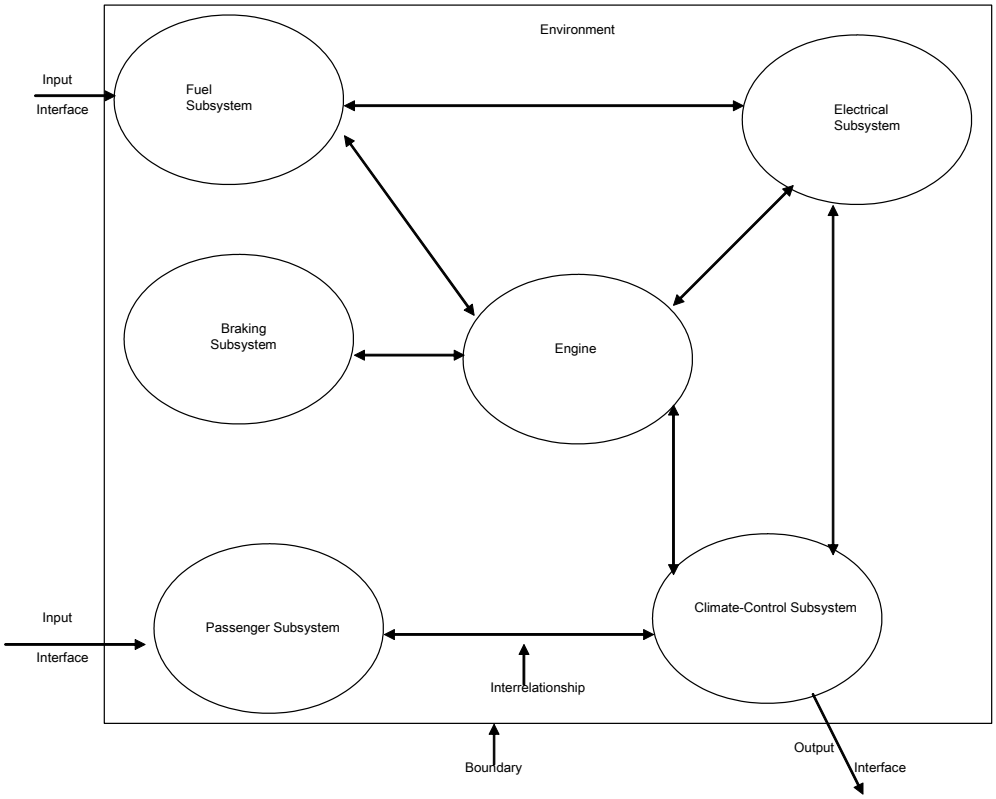
Part c



Think about small cases

A car is a system with several subsystems, including the braking subsystem, the electrical subsystem, the engine, the fuel subsystem, the climate-control subsystem, and the passenger subsystem. Draw a diagram of a car as a system and label all of its system characteristics.

The purpose of this question is to encourage your students to begin viewing various entities as systems, and decompose these systems into their components. A simple diagram, showing examples of the car system's characteristics, is provided below. The car's purpose is to provide safe transportation to wherever the driver needs to go. A constraint is the number of passengers that can ride in the vehicle.



Explain how a computer-based information system designed to process payroll is a specific example of a system. Be sure to account for all nine components of any system in your explanation.

A payroll system can be thought of as a system because it is composed of interrelated subsystems that work together to accomplish a purpose. A payroll system has many components, including components that generate paychecks, make direct deposits, generate various internal reports, process time cards, and process various forms and tax returns. Many of these subsystems or components are related. For instance, the reporting component interacts with the tax, paycheck, and direct deposit components. A direct deposit made to an employee's bank account is an example of an interface. The payroll system's boundary encompasses all of the payroll system's activities. Employees, banks, and the local, state, and federal governments are part of the payroll system's environment. Hours worked, number of dependents, and number of sick days are examples of input to the payroll system. The various reports, paychecks, and tax forms are types of output. Federal and state tax regulations are examples of system constraints.

Thinking about sources of software and project development

1. **Describe and compare the six sources of software.**

The six sources of software identified in the textbook are: (1) information technology services firms, (2) packaged software providers, (3) vendors of enterprise solution

software, (4) application service providers and managed service providers, (5) open-source software, and (6) in-house development. IT services firms help companies develop custom information systems for internal use; they develop, host, and run applications for customers; or they provide other services. An IT services firm may be chosen if the system can't be developed internally or requires customer support. Packaged software providers are companies that produce software exclusively, like Microsoft or Intuit, and are preferable if the task needing the system is generic. Vendors of enterprise solution software create a system that is composed of a series of integrated modules. Each module supports a business function, such as accounting, or human resources. ERP systems may be appropriate if a complete system is required that can cross functional boundaries. A more intense option for larger, more customizable solutions are Managed service providers who can provide more services than application service providers. ASPs and MSPs may be appropriate when instant access to an application is desired, and in the case of ASPs, when the task is generic. Open-source software is a type of software that is developed by a community of interested people and it is freely available. Open-source software is preferable when the task is generic and cost is an issue. Finally, in-house development is still an option. In-house development refers to the organization developing software with its own IT assets, rather than purchasing software elsewhere. In-house development may be appropriate when a system must be built from scratch and as long as the personnel and resources are available to the organization.

2. What are the differences between ASPs and MSPs?

ASPs host and run computer applications for other companies and usually purchase or license applications from other software vendors. MSPs provide similar services, but can also provide more customization and may also include business processes, engineering, security, and maintenance. MSPs can also provide network-based series, customized applications, and even equipment and may cost more than ASPs.

3. How can you decide among various off-the-shelf software options? What criteria do you use?

To decide what off-the-shelf software to buy, compare products and vendors. Use the following criteria (among others that may be more situation-specific): cost, functionality, vendor support, viability of vendor, flexibility, documentation, response time, and ease of installation.

4. What is an RFP and how do analysts use one to gather information on hardware and system software?

An RFP is a formal document that provides detailed specifications about a target information system and asks vendors for information on how they would develop the system. Analysts use RFPs as a way to get vendors to perform the necessary research into specific design strategies and the hardware and system software vendors believe are necessary for developing the new system.

5. What methods can a systems analyst employ to verify vendor claims about a software package?

To verify vendor claims about a software package, an analyst can ask for a software demonstration, use the software (and its documentation and training materials), personally talk with other users of the software, ask specific questions via a questionnaire, and consult independent software testing and abstracting services.

6. What are ERP systems? What are the benefits and disadvantages of such systems as a design strategy?

ERP is a system that is composed of a series of integrated modules. Each module supports a business function, such as accounting, or human resources. An advantage of ERP systems is that the modules are integrated across business functional areas, so that, for instance, all aspects of a single transaction can occur seamlessly within one information system, where it would have had to have been processed multiple times across several systems in a more traditional environment focused on functional areas. Other benefits include having a single repository of data for all aspects of a business process and the flexibility of the modules; other modules can be added as needed and are immediately integrated into the existing system. Disadvantages include complexity, lack of knowledge among organization's personnel and reliance on consultants, and expense. In some cases, an organization may be required to change how it does business to more efficiently utilize ERP systems.

7. Explain reuse and its advantages and disadvantages.

Reuse refers to using previously written software resources, especially objects and components, in new applications. It can be more efficient to reuse objects or components, and research has shown that it can increase productivity, reduce defects, and reduce the amount of rework needed. Disadvantages include organizational lack of commitment, lack of proper training and rewards needed to promote reuse, and it is difficult to measure the economic gains from reuse.

8. Compare and contrast the four approaches to reuse.

"Ad Hoc" reuse refers to individuals finding ways to reuse objects and components on their own. Facilitated reuse refers to reuse that is encouraged by the organization, but with limited resources, infrastructure, and policies that would make reuse easier. Managed reuse refers to reuse that occurs because the organization enforces reuse through resources, infrastructure, policies, and people. Designed reuse refers to an organization carefully designing assets for reuse.

9. Research how to prepare a request for proposal.

The request for proposal (RFP) is used when the organization wants to solicit proposals from several competing vendors. RFPs usually first give some background information on the company and the business units involved in the request, an explanation of the information system needs, a description of what is wanted from the vendors (i.e., what information they must provide or other actions they must take), and an explanation of any rules or procedures for the RFP and system development process. The bulk of the document then describes the

mandatory, essential, and desirable requirements in the areas of need (e.g., functionality, hardware, software, and service).

10. Review the criteria for selecting off-the-shelf software presented in this chapter. Use your experience and imagination and describe other criteria that are or might be used to select off-the-shelf software in the real world. For each new criterion, explain how its use might be functional (i.e., it is useful to use this criterion), dysfunctional, or both.

In addition to cost, functionality, vendor support, viability of vendor, flexibility, documentation, response time, and ease of installation, there are a number of other real world criteria that might be included. People often choose application packages, such as word processors and spreadsheets, based solely on their familiarity with the packages and/or their bias toward one hardware platform or operating system over another. To a certain extent this can be functional. However, this could also be dysfunctional. For example, it would be useful to consider the current staff's familiarity with the new application software and the resulting need for retraining. However, it would be dysfunctional if a company did not ever choose new software because of the employees' lack of familiarity with it. Eventually, software will evolve and the market will change, and the employees (and the company) will be left behind using antiquated technology.

Some other criteria include compatibility with currently used application software (so, for example, data can be shared), compatibility with existing hardware and system software, ability to support a range from novice to experienced (or power) users, and appeal of the user interface (ease of use).

11. In the section on choosing off-the-shelf software, eight criteria are proposed for evaluating alternative packages. Suppose the choice was between alternative custom software developers rather than prewritten packages. What criteria would be appropriate to select and compare among competing bidders for custom development of an application? Define each of these criteria.

The list for evaluating alternative custom software developers would be very similar to that for selecting off-the-shelf application software or for computer hardware and system software. In addition to cost, functionality, vendor support, viability of vendor, flexibility, documentation, response time, and ease of installation, we might include the current staff's familiarity with the software, need for retraining, compatibility and connectivity with current systems, and the track record of the vendor in successfully implementing similar software in other organizations. Such vendors should have an established track record of developing similar software in other organizations. Their references should be checked thoroughly, including visits to these other sites. If the developer's role will end after the application is accepted, then the reputation of the vendor for handling this transition from external development to internal maintenance is important. From a legal point of view, you may want to select a custom developer based on the willingness to sign a non-disclosure agreement, so that they are not allowed to develop a similar system for one of your competitors, at least for some amount of time.

12. How might the project team recommending an ERP strategy justify its recommendation as compared with other types of design strategies?

The project team would need to justify an ERP strategy by highlighting the strategic advantage to reengineering business processes to integrate with ERP modules. Doing so will provide a single repository for data, which can streamline all IT within the organization and ensure that data is consistent and accurate. Furthermore, if expansion is required in the future, additional modules are able to be integrated immediately. Finally, considering the regulatory climate, the ERP vendor (and consultants) will likely be better at ensuring the organization's compliance with regulations, such as Sarbanes-Oxley, HIPAA, etc.

13. Discuss the reasons why organizations undertake information system projects.

Information system projects are undertaken for two primary reasons: to take advantage of business opportunities and to solve business problems. Providing an innovative service to customers through the creation of a new system is an example of an opportunity. Modifying the way in which an existing system processes data so that more accurate or timely information is provided to users is an example of solving a business problem.

14. Describe the activities performed by the project manager during project initiation.

Project initiation has six major activities: (1) establishing the project initiation team; (2) establishing a relationship with the customer; (3) establishing a project initiation plan; (4) establishing management procedures; (5) establishing the project management environment and project workbook; and (6) developing the project charter. When establishing the project initiation team, the project manager organizes an initial core of project team members to assist in accomplishing the project initiation activities. Establishing a relationship with the customer builds a cooperative and trusting partnership with the customer. The project initiation plan defines the necessary activities required to organize the initiation team while they are working to define the scope of the project. Management procedures establish team communication and reporting procedures, job assignments and roles, project change procedures, and determine how project funding and billing will be handled. Establishing the project management environment and project workbook creates the repository for all project correspondence, inputs, outputs, deliverables, procedures, and standards of the project team. The project charter is a short, high-level document prepared for both internal and external stakeholders; this document ensures that the analyst and customer have a common understanding of the project.

15. Describe the activities performed by the project manager during project planning.

The activities performed by the project manager during project planning include:

(1) describing project scope, alternatives, and feasibility;
(2) dividing the project into manageable tasks;
(3) estimating resources and creating a resource plan;
(4) developing a preliminary schedule;

(5) developing a communication plan;
(6) determining project standards and procedures;
(7) identifying and assessing risk;
(8) creating a preliminary budget;
(9) developing a Project Scope Statement; and
(10) setting a Baseline Project Plan.

Describing project scope, alternatives, and feasibility develops an understanding of the content and complexity of the project by gaining answers to and agreement on the following types of questions:

- What problem or opportunity does the project address?
- What are the quantifiable results to be achieved?
- What needs to be done?
- How will success be measured?
- How will we know when we are finished?

After defining the scope of the project, the next objective is to identify and document general solutions for the current business problem or opportunity and to assess each solution for feasibility so that a choice can be made as to which to consider during subsequent SDLC phases.

Dividing the project into manageable tasks identifies specific tasks and then logically orders them to ensure a smooth evolution between tasks.

Estimating resources and creating a resource plan approximates resource requirements for each project activity and uses this information to create a project resource plan.

A **preliminary schedule is developed** using information regarding tasks and resource availability to assign time estimates to each activity in the work breakdown structure. This assignment will allow for the creation of target starting and ending dates for the project.

The **communication plan** outlines the communication procedures between management, project team members, and the customer.

Project standards and procedures specify how various deliverables are produced and tested.

The activity of **identifying and assessing risk** examines sources of project risk and estimates the consequences of those risks. Risks might arise from the use of new technology, resistance to change, availability of critical resources, competitive and regulatory actions, and team member inexperience with technology or the business area.

Creating a preliminary budget is the process of outlining the planned expenses and revenues associated with the project.

The focus of developing a Project Scope Statement is to create a document that outlines all work that will be done and makes clear what the project will deliver.

The purpose of **setting a Baseline Project Plan** is to develop an initial plan that reflects the best estimate of the project's tasks and resource requirements and is used to guide the next project phase, execution.

16. Describe the activities performed by the project manager during project execution.

Project execution activities include:
(1) executing the Baseline Project Plan;
(2) monitoring project progress against the Baseline Project Plan;
(3) managing changes to the Baseline Project Plan;
(4) maintaining the project workbook; and
(5) communicating the project status.

The **execution of the Baseline Project Plan** puts the Baseline Project Plan (e.g., the execution of project activities, acquire and assign resources, orient and train new team members, keep the project on schedule, and assure the quality of project deliverables) into action.

The purpose of **monitoring project progress against the Baseline Project Plan** is to compare the actual progress of the project to the Baseline Project Plan. If the project gets ahead of (or behind) schedule, adjustments to resources, activities, and budgets can be made.
When **managing changes to the Baseline Project Plan**, alterations to the Baseline Project Plan are made as events (e.g., a slipped completion date of an activity) occur.

The **project workbook is maintained** through routine updating of project-related information documented within the workbook.

The project **status is communicated** to inform all interested parties—systems developers, managers, and customers—about how the project is progressing.

17. List various project team communication methods, and describe an example of the type of information that might be shared among team members using each method.

.

Team Communication Methods and

Corresponding Examples

Procedure	Formality	Use
Project workbook	High	Inform; permanent record
Meetings	Medium to high	Resolve issues
Seminars and workshops	Low to medium	Inform
Project newsletters	Medium to high	Inform
Status reports	High	Inform
Specification documents	High	Inform; permanent record
Minutes of meetings	High	Inform; permanent record
Bulletin boards	Low	Inform
Memos	Medium to high	Inform
Brown bag lunches	Low	Inform
Hallway discussions	Low	Inform; resolve issues

18. **Describe the activities performed by the project manager during project closedown.**

    The activities performed by the project manager during project closedown are: (1) closing down the project, (2) conducting postproject reviews, and (3) closing the customer contract. The focus of closing down the project is to conclude the project. The objective of conducting post-project reviews is to assess the strengths and weaknesses of project deliverables, the processes used to create them, and the project management process. The purpose of closing the customer contract is to ensure that all contractual terms of the project have been met.

19. **What characteristics must a project have in order for critical path scheduling to be applicable?**

    Critical path scheduling is a scheduling technique where the order and duration of the sequence of project activities directly affect the completion date of a project.

    Applicable project characteristics include:
    (1) well-defined activities that have a clear beginning and end point;
    (2) activities that can be worked on independently of other activities;
    (3) activities that are ordered (or can be ordered); and
    (4) activities that when completed serve the purpose of the project.

20. Describe the steps involved in making a Gantt chart.

The steps involved in making a Gantt chart are:
(1) identify each activity to be completed in the project;
(2) determine time estimates and calculate the expected completion time for each activity;
(3) determine the sequence of the activities and precedence relationships among all activities; and
(4) construct the Gantt chart.

21. Describe the steps involved in making a Network diagram.

The steps involved in making a Network diagram are:
(1) identify each activity to be completed in the project;
(2) determine time estimates and calculate the expected completion time for each activity;
(3) determine the sequence of the activities and precedence relationships among all activities; and
(4) construct the Network diagram.

NOTE: The steps involved in making Gantt charts and Network diagrams are identical. However, they differ in the way the information is presented. A Gantt chart is presented with bars that represent the tasks, whereas a Network diagram is presented with a rectangle to represent the tasks, with task information included in the rectangle. See the answers below in numbers 7 and 8 in the Problems and Exercises Solutions.

22. In which phase of the systems development life cycle does project planning typically occur? In which phase does project management occur?

Project planning typically occurs during the planning and analysis phases of the SDLC. Project management occurs during all phases of the SDLC; yet, different project management activities occur during different SDLC phases.

23. What are some reasons why one activity may have to precede another activity before the second activity can begin? In other words, what causes precedence relationships between project activities?

Task sequence will depend on which tasks produce deliverables needed in other tasks, when critical resources are available, constraints placed on the project by the client, and the process outlined in the SDLC.

Somesthing about project control

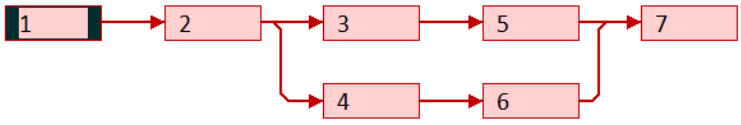
Can a project have two critical paths? Why or Why not? Give a brief example to illustrate your point.

Yes. A critical path is, by definition, the shortest time in which a project can be completed. If a project has two sub-paths that take the same amount of time, then both are on a critical path. The example is illustrated below.

Both sub-paths (through C & E, and through D & F) have identical total duration, and thus both satisfy the critical path criterion.

Activity	Time (weeks)	Immediate Predecessors
A	1	
B	1	A
C	2	B
D	3	B
E	3	C
F	2	D
G	1	E,F

This PERT Chart shows that there are 2 critical paths.



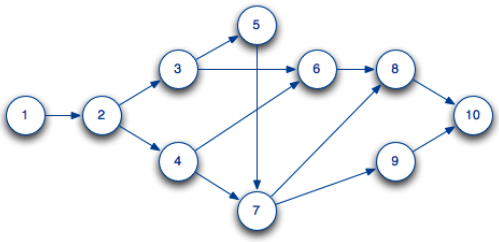
A project has been defined to contain the following list of activities along with their required times for completion.

Activity No.	Activity	Time (weeks)	Immediate Predecessors
1	Collect requirements	3	
2	Analyze processes	2	1
3	Analyze data	2	2
4	Design processes	6	2

5	Design data	3	3
6	Design screens	2	3,4
7	Design reports	4	4,5
8	Program	5	6,7
9	Test and Document	7	7
10	Install	2	8,9

---

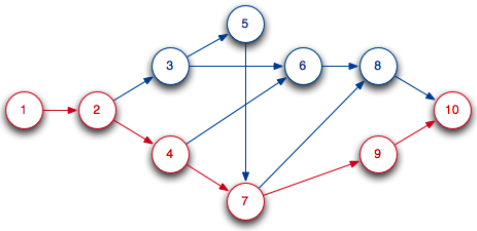
a. Draw a Network diagram for the activities.



b. Calculate the earliest expected completion time.

24 weeks.

c. Show the critical path



d. What would happen if activity 6 were revised to take 6 weeks instead of 2 weeks?

The critical path would change to include activity 6 instead of activity 7, and the overall duration of the project would increase by 2 weeks.

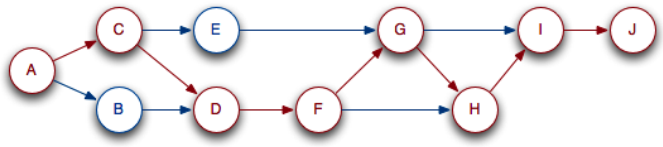
Assume you have a project with 10 activities labeled A-J. Derive the earliest completion time (or early finish—EF), latest completion time (or late finish—LF), and slack for each of the following tasks (begin at time = 0). Which tasks are on the

critical path? Draw both Gantt chart and Network diagram for these tasks, and make sure you highlight the critical path on your Network diagram.

Activity	Preceding Event	Expected Duration	EF	LF	Slack	Critical Path?
A	—	3				
B	A	1				
C	A	2				
D	B,C	5				
E	C	3				
F	D	2				
G	E,F	3				
H	F,G	5				
I	G,H	5				
J	I	2				

Activity	Preceding Event	Expected Duration	EF	LF	Slack	Critical Path?
A	—	3	3	3	0	Yes
B	A	1	4	5	1	No
C	A	2	5	5	0	Yes
D	B,C	5	10	10	0	Yes
E	C	1	6	12	6	No
F	D	2	12	12	0	Yes
G	E,F	3	15	15	0	Yes
H	F,G	5	20	20	0	Yes

I	G,H	5	25	25	0	Yes
J	I	2	27	27	0	Yes



Analyzing the alternatives

Imagine you are a member of the project approval committee. An ambitious young manager in the marketing department is well-connected with the top management team in your company. He catches you in the hall and mentions that he is frustrated with how long it takes to get a simple system enhancement through the "bureaucratic" approval process. He wonders whether you could sign off on a small enhancement request for his team's reporting application. With a wink, he promises to "owe you one." What would you say to him and why?

Answers will vary. Most should include a discussion of the necessity of project planning and selection in the context of all the potential projects in an organization. Giving preferential treatment to one project will weaken any project approval process. All projects should be considered by the approval committee and subjected to the same review and approval process.

Of the six methods for assessing project feasibility, which is the most important? In which situation is each method more or less important?

The specifics of the project will determine which feasibility criterion is most important. Also, each method helps identify potential benefits and risks associated with the project.

Assume you are put in charge of launching a new website for a local nonprofit organization. What costs would you need to account for? Make a list of expected costs and benefits for the project. You don't need to list values, just sources of expense. Consider both one-time and recurring costs.

One-time Costs: server purchase, internet service setup, initial website development, etc.

Recurring Costs: internet connectivity fees, hardware upgrades, website maintenance, etc.

Recurring Benefits: reduced administrative costs, etc.

Now the analysis

1. **Describe systems analysis and the major activities that occur during this phase of the systems development life cycle.**

   During systems analysis, you determine how the current information system functions and assess what users would like to see in a new system. Requirements determination, requirements structuring, and alternative generation and selection are the three primary systems analysis activities.

2. **What are some useful character traits for an analyst involved in requirements determination?**

   Requirements determination is a crucial part of the systems development life cycle; therefore impertinence, impartiality, relaxing constraints, attention to details, and reframing are important characteristics.

3. **Describe three traditional techniques for collecting information during analysis. When might one be better than another?**

   Traditional techniques for collecting requirements include interviewing and listening, observing users, and analyzing procedures and other documents. Interviewing and listening involve talking with users individually or as a group to discover their views about the current and target systems; it also involves carefully preparing an interview outline and guide before conducting the interview. Interviews are best done when only a few people are involved, when you need open-ended questions or the questions vary from individual to individual, or when a more personal method is needed.

   Directly observing users involves watching how people work in order to uncover information users may not be consciously aware of. Direct observation is best when detailed or complicated procedures must be documented, when you do not want people to know they are giving you information you need, when only a few people are involved, and when observational data are representative of all situations. Analyzing procedures and other documents involves identifying and collecting written procedures, forms, reports, and other relevant documents in order to better identify data and processes that would be part of the current and target systems. Analyzing documents is the best technique when documents are complete and unbiased, when other forms of requirements determination are too obtrusive, and when history must be studied and people do not have first-hand data about history.

4. **What are the general guidelines for conducting interviews?**

5. **What are the general guidelines for collecting data through observing workers?**

   When observing workers, it is important that the analyst select typical and atypical people and sites; he/she should also observe during normal and abnormal conditions.

6. **What are the general guidelines for collecting data through analyzing documents?**

   Document analysis is an effective way to learn about the current system and future system's requirements. It is important for the analyst to collect and review as many documents as he/she can. He/she should collect organizational mission statements, business plans, organization charts, business policy manuals, job descriptions, internal and external correspondence, and reports from prior organizational studies. Specifically, he/she should review business forms, reports, written work procedures, and existing system documentation.

7. **Compare collecting information through observation and through document analysis. Describe a hypothetical situation in which each of these methods would be an effective way to collect information system requirements.**

   The observation method is high in information richness; requires significant time, can be expensive, and provides the chance for follow-up and probing. The observee is known to the analyst and the potential audience is small. In contrast, the information richness of document analysis is low and old; the time required is low to moderate; its expense is low to moderate, and the chance for follow-up is limited. The confidentiality of the document depends on the nature of the document; there is no clear commitment on the subject's part, and there is potentially bias by which documents were retained.

8. **What is JAD? How is it better than traditional information-gathering techniques? What are its weaknesses?**

   JAD is a structured process in which users, managers, and analysts work together for several days in a series of intensive meetings to specify or review system requirements. It is better than traditional techniques because you have key personnel in one place at one time, saving everyone time and resulting in high levels of system ownership as more people have more of a role in the development process. Weaknesses include the level of commitment necessary to make the JAD work, the high degree of required planning, and the typical lack of computer support.

9. **How has computing been used to support requirements determination?**

   Computing supports requirements determination in the form of CASE tools, the administration of questionnaires, group support systems, and prototyping. CASE tools, especially upper CASE tools, can be of assistance during JAD and prototyping.

10. **Describe how prototyping can be used during requirements determination. How is it better or worse than traditional methods?**

    Prototyping can be used during requirements determination to collect user requirements and present them in the form of a working system prototype. Users can look at, play with, and compare the prototype to their system requirements. Analysts can then adjust the prototype to better fit what the users have in mind. Prototyping is better than traditional methods where system requirements are not well understood, where few users that are stakeholders are involved, where designs may be complex, where there have been past communication problems, and where the necessary tools are readily available. Prototyping may be worse than traditional methods where formal requirements are not documented, where prototypes become

idiosyncratic to the initial user, where issues of data sharing and integration with other systems are ignored, and where SDLC checks are bypassed.

11. When conducting a business process reengineering study, what should you look for when trying to identify business processes to change? Why?

As part of the BPR effort, key business processes and important, changeable, or dysfunctional activities are identified. Key business processes are the structured, measured set of activities designed to produce a specific output for a particular customer or market. Benefits of BPR include radical improvements in speed, quality, and customer satisfaction.

12. What are disruptive technologies and how do they enable organizations to change their business processes radically?

Disruptive technologies enable the breaking of long-held business rules that inhibit organizations from making radical business changes. Disruptive technologies enable companies to apply information technology innovatively. As a point of discussion, ask students to discuss the concept of a virtual university. Is this an acceptable application of a disruptive technology?

13. One of the potential problems mentioned in the chapter with gathering information requirements by observing potential system users is that people may change their behavior when observed. What could you do to overcome this potential confounding factor in accurately determining information requirements?

The analyst could conduct the observations unobtrusively, so that the effect on the users' behavior is minimized. This could be done using a confederate or by a hidden camera. The analyst could also brief the users on the observation so that the users will relax and behave naturally. For example, you can make it clear to users that they are not being evaluated and that the observations collected will not be associated with anyone individually. In addition, the analysts could perform multiple observations over time. This would tend to minimize the effects of aberrant behaviors. Alternatively, the analysts could supplement their requirements determination with additional data collection methods.

14. Summarize the problems with the reliability and usefulness of analyzing business documents as a method for gathering information requirements. How could you cope with these problems to use business documents effectively as a source of insights on system requirements?

One of the primary problems with analyzing business documents is that they do not give the full picture of how work is done and why. First, business documents are often incomplete, since people have selectively retained documentation. Second, business documents often describe the formal system as opposed to the informal system, which is more often the way the work is actually completed. Whether the business documents are accurate or not, they provide useful information. If the business documents are accurate, then much of the work of gathering information requirements is nearly finished. If the business documents are inaccurate, then the analyst can use these to understand how the work processes ought to be done, or are thought to be done, or, perhaps, should not be done. In any event, analyzing business documents should be done in conjunction with other, supplemental data

collection methods. In addition, the analyst should speak to multiple people to gather their perceptions and uses of the documents.

15. How might an interview guide differ when a group interview is to be conducted?

Because the group interview might be more difficult to conduct and more time consuming, the analyst might add to the interview guide "time certain" events or "time stamped" agenda items. For example, if the meeting begins at 8:00 a.m., and the manager of the users will come in to the meeting to give a brief talk, this event might be scheduled for 8:15 a.m. until 8:30 a.m. Other processes in the meeting will be postponed during this event. With time stamping, the analyst writes definitive start times next to each of the agenda items and then uses this to keep the group on track. Alternatively, the analyst might write specific questions that should be asked of specific members of the interview group. If several analysts are involved in the group interview, the group could be broken into parallel sessions, each with its own agenda. Finally, agenda activities should be allotted for discussion and interchange between the interviewees, so that consensus and synergy can occur.

16. JADs are very powerful ways to collect system requirements, but special problems arise during group requirements collection sessions. Summarize these special interviewing and group problems, and suggest ways that you, as a group facilitator, might deal with them.

Some of the problems include difficulty in scheduling, enabling all group members to participate during the meeting, some people being afraid or not willing to speak in front of certain other people, conflicts existing among group members, keeping the group on track during the meeting, and accurately collecting all the information as multiple people speak at once. Some of the ways to deal with these problems include training in team building, group dynamics, and managing conflict; having multiple interviewers work together; and using a technological aid such as CASE or GSS.

17. Suppose you are a systems analyst charged with gathering information requirements. You decide that you want to use prototyping to gather these requirements. It provides benefits beyond interviews and observations, but also presents unique challenges. Discuss the challenges you expect to face and what processes you will put in place to prevent them from harming your information system.

This question has several possible answers.

Some challenges that can be expected are:

1. Analysts may feel that properly documenting the requirements is redundant. This could be addressed by requiring careful notes to be taken when soliciting feedback from users or by not allowing any changes to the prototype without justification in the requirements.

2. Initial users that participate in the prototyping process may have heavier influence on the resulting system. This could be mitigated by rotating the early prototype responses among various users to ensure a more comprehensive set of perspectives.
3. The prototype is non-functional and does not interoperate with other data sources. The system may perform differently when more data is involved or different data sources are accessed. Effort should be made to load the system with enough data to allow the users to see a more realistic view of the system.

18. Questionnaires can be administered both on paper and via the Internet. Online questionnaires allow for the use of complex analysis tools and real-time results. However, online questionnaires have idiosyncratic challenges. Three such challenges can be computer access concerns, getting users to participate, and employee concerns for privacy of results. Discuss when each concern is likely to impact the online questionnaire and how you would address each challenge.

This question asks students to discuss solutions to three problems with Internet-based questionnaires. These concerns and possible solutions are:

1. Computer access concerns – not all workers will have roles that provide access to computers, such as factory line workers and nurses in hospitals. However, their input can be critical to the successful creation of an information system to meet their needs. For questionnaires administered online, workers should be given some way to provide their input while at work. Either providing access to a computer or providing a paper-based form may address this issue.
2. User participation – getting workers to participate can be more difficult with online questionnaires because it requires taking time on a computer to complete, whereas paper-based questionnaires can be filled out on-the-spot. This can be addressed by providing time set aside just for the questionnaire, sending follow-up reminders to complete the questionnaire, or by providing incentives for the completion of the questionnaire.
3. Privacy concerns – employees may be concerned that their responses will not be kept confidential. This may be more of an issue when the employee has opinions which are different than those of their superiors (formal or informal). Employees should be reassured that the responses will not be linked to the individual respondents. An easy way to enhance privacy is to not collect any personal information and to have a single link to the instrument that all respondents will type in.

19. Some of the key problems with information systems that show up later in the systems development life cycle can be traced back to inadequate work during requirements determination. How might this be avoided?

It is imperative that the analyst gathers as much information about the current and new system's requirements as she can. During requirements determination, impertinence, impartiality, relaxing constraints, attention to details, and reframing are good characteristics for an analyst to exhibit. These characteristics will help the analyst better understand the existing system and the new system's requirements. Additionally following the effective guidelines for conducting interviews, performing observations, and analyzing documents are important. It is important that the analyst not hurry through this step; he/she must take his/her time and gather all the necessary information.

20. What is a data-flow diagram? Why do systems analysts use data-flow diagrams?

A data-flow diagram is a picture of the movement of data between external entities and the processes and data stores within a system. Systems analysts use data-flow diagrams to help them model the processes internal to an information system as well as how data from the system's environment enter the system, are used by the system, and are returned to the environment. DFDs help analysts understand how the organization handles information and what its information needs are or might be. Analysts also use DFDs to study alternative information handling procedures during the process of designing new information services.

21. Explain the rules for drawing good data-flow diagrams.

Processes cannot have only outputs, cannot have only inputs, and must have a verb phrase label. Data can move to a data store from only a process, not from another data store or an outside source. Similarly, data can be moved to only an outside sink or to another data store by a process. Data to and from external sources and sinks can be moved by only processes. Data flows move in one direction only. Both branches of a forked or a joined data flow must represent the same data. A data flow cannot return to the process from which it originated.

22. What is decomposition? What is balancing? How can you determine if DFDs are not balanced?

Decomposition is the iterative process by which a system description is broken down into finer and finer detail, creating a set of diagrams in which one process on a given diagram is explained in greater detail on a lower–level diagram. Balancing is the conservation of inputs and outputs to a data-flow diagram process when that process is decomposed to a lower level. You can determine if a set of DFDs are balanced or not by observing whether or not a process that appears in a level-*n* diagram has the same inputs and outputs when decomposed for a lower-level diagram.

23. Explain the convention for naming different levels of data-flow diagrams.

The highest level DFD is called a context diagram. It represents the system as a single process, with all the related entities and the data flows in and out of the system. The next level diagram, called a level-0, decomposes the one process from the context diagram into between two to nine high-level processes. Each process in

a level-0 diagram can be decomposed, if necessary. Each resulting diagram is called a level-1. Should processes in a level-1 diagram be decomposed, each resulting diagram would be called a level-2 diagram. Each of these processes would be decomposed on a level-3 diagram, and so on.

24. How can data-flow diagrams be used as analysis tools?

DFDs can be used as analysis tools to help determine the completeness of a system model and a model's internal consistency, as a way to determine when system events occur through analyzing timeliness, and, through iterative use, to develop and check models. Analysts can study DFDs to find excessive information handling, thus identifying areas for possible efficiencies.

25. Explain the guidelines for deciding when to stop decomposing DFDs.

You can stop decomposing a DFD when the following six conditions are satisfied:

(1) each process is a single decision or calculation or a single database operation, such as retrieve, update, create, delete, or read;
(2) each data store represents data about a single entity, such as a customer, employee, product, or order;
(3) the system user does not care to see any more detail, or when you and other analysts have documented sufficient detail to do subsequent systems development tasks;
(4) every data flow does not need to be split further to show that different data are handled in different ways;
(5) you believe that you have shown each business form or transaction, computer screen, and report as a single data flow; and
(6) you believe there is a separate process for each choice on all lowest–level menu options for the system.

26. How do you decide if a system component should be represented as a source/sink or as a process?

Sources and sinks are always outside of the system being considered. They are of interest to the system being considered only because they represent sources of data coming into the system and destinations for data leaving the system. If any data processing occurs inside a source or sink, it should be of no interest to the system being modeled. If the processing is of interest, however, or if the identified source/sink has several inputs and outputs to and from the rest of the system, it may be better considered as an internal process.

27. What unique rules apply to drawing context diagrams?

Context diagrams have only one process that represents the entire system being modeled and shows only the data flows into and out of the system. The diagram also includes sources and sinks, which represent the system's environmental boundaries. There are usually no data stores in a context diagram.

28. Explain what the term *DFD consistency* means and provide an example.

DFD consistency is the extent to which information contained on one level of a set of nested data-flow diagrams is also included on other levels. Balancing errors are one type of consistency violation mentioned in the textbook. For instance, a

payment data flow that appears on a level-1 diagram, but not on the level-0 diagram, is a consistency violation.

29. Explain what the term *DFD completeness* means and provide an example.

DFD completeness is the extent to which all necessary components of a data-flow diagram have been included and fully described. A data store that does not have any data flows coming into or out of it is a completeness violation.

30. How well do DFDs illustrate timing considerations for systems? Explain your answer.

Timing considerations are not noted on DFDs. For instance indications of whether a process occurs hourly, daily, weekly, monthly, or yearly are not made.

31. How can data-flow diagrams be used in business process redesign?

DFDs can graphically illustrate, at varying levels of detail, how a process or processes work. Analysts can study DFDs of the current system and identify areas of inefficiency. Analysts can prepare DFDs for the new system, identifying changes for the new system.

32. What is the purpose of logic modeling? What techniques are used to model decision logic and what techniques are used to model temporal logic?

The purpose of logic modeling is to show the rules that govern the behavior of processes represented in data-flow diagrams. Decision tables model decision logic. State diagrams model temporal logic.

33. What are the steps in creating a decision table? How do you reduce the size and complexity of a decision table?

The steps for creating a decision table are:

(1) name the conditions and the values each condition can assume;
(2) name all possible actions that can occur;
(3) list all possible rules;
(4) define the actions for each rule; and
(5) simplify the decision table.

To reduce the size and complexity of a decision table, use separate, linked decision tables, or use numbers that indicate sequence rather than Xs where rules and action stubs intersect. Also, the analyst should identify indifferent conditions and simplify the decision table.

34. What formula is used to calculate the number of rules a decision table must cover?

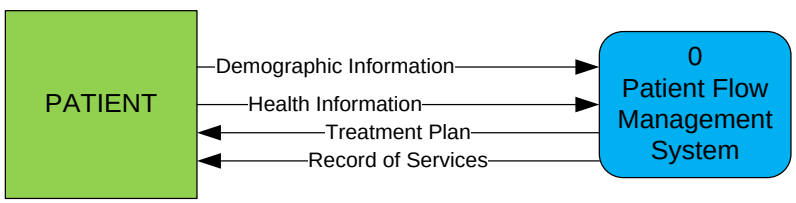
To determine the number of rules a decision table must cover, simply determine the number of values each condition may have and multiply the number of values for each condition by the number of values for every other condition.

Analyze these small cases

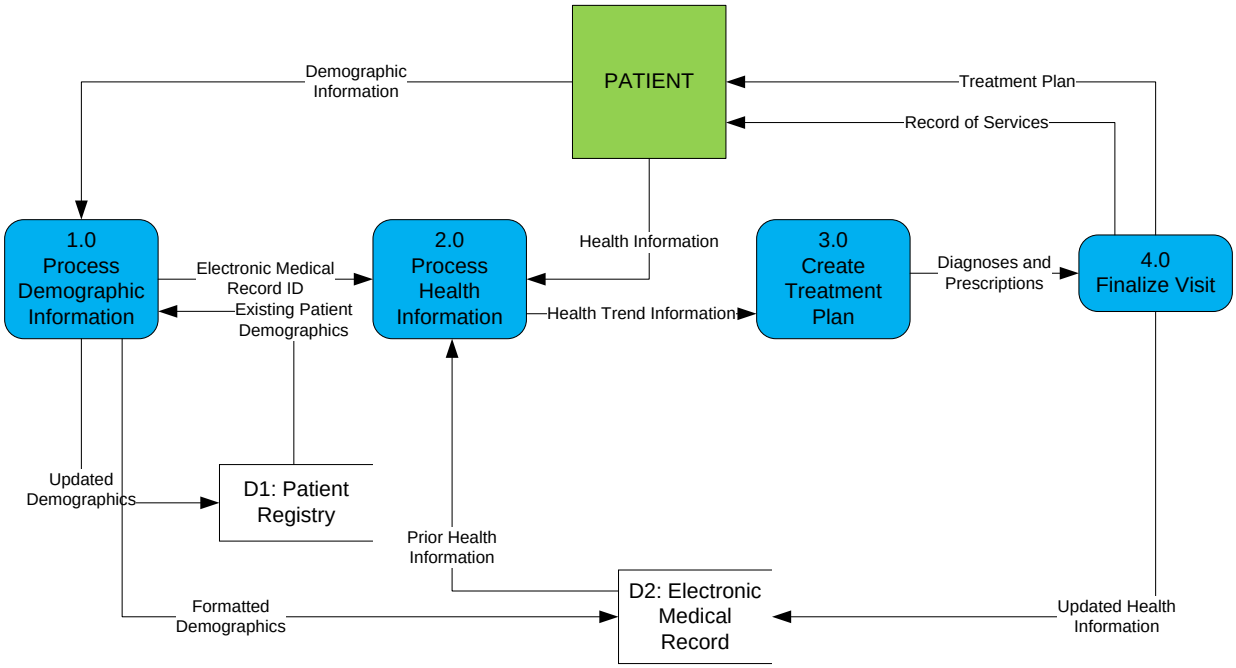
Starting with a context diagram, draw as many nested DFDs as you consider necessary to represent all of the details of the patient flow management system described in the following narrative. You must draw at least a context diagram and a level-0 diagram. In drawing these diagrams, if you discover that the narrative is incomplete, make up reasonable explanations to complete the story. Provide these extra explanations along with the diagrams. Indicate what could be errors that in DFD can occur

Dr. Frank's walk-in clinic has decided to go paperless and will use an information system to help move patient through the clinic as efficiently as possible. Patients enter the system at the front desk by providing demographic information to the personnel. If this is the first time the patient has been seen, insurance and basic demographic information is collected from the patient. If the patient has been seen previously, the patient is asked to verify the information pulled from the patient registry. The front desk person then updates the patient registry and ensures that the patient has a chart in the electronic medical records system; if not, a new medical record is started by placing formatted demographics into a blank medical record. The front desk person then enters the medical record ID into the system. Next, a medical technician collects the patient's health history, weight, height, temperature, blood pressure, and other medical information, and combines this information with any information from the patient's medical record, summarizing the information into a health trend. A doctor then sees the patient, prescribes medication or treatment where appropriate based on the medical trend, and sends the patient to checkout. The employee at checkout updates the patient's electronic medical record and provides prescriptions for medications or treatments and a printed record of the health services received.

CONTEXT DIAGRAM:



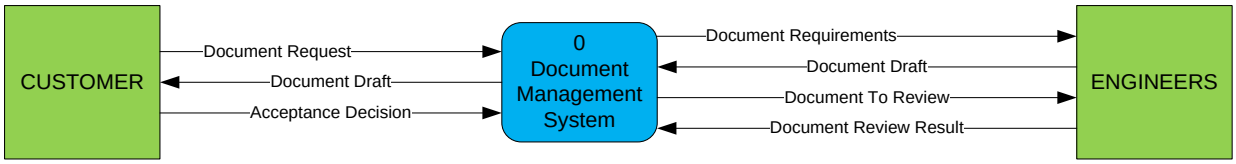
Level-0 Diagram:



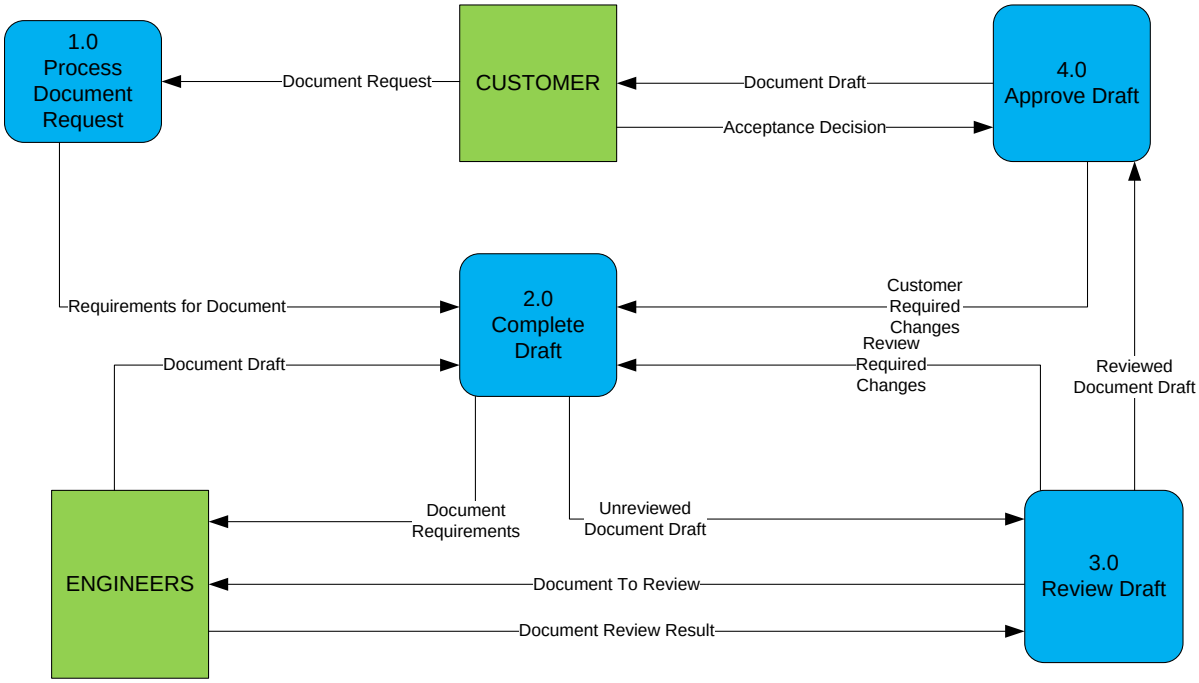
Starting with a context diagram, draw as many nested DFDs as you consider necessary to represent all of the details of the engineering document management system described in the following narrative. You must draw at least a context diagram and a level-0 diagram. In drawing these diagrams, if you discover that the narrative is incomplete, make up reasonable explanations to complete the story. Provide these extra explanations along with the diagrams.

Projects, Inc. is an engineering firm with approximately 500 engineers that provide mechanical engineering assistance to organizations, which requires managing many documents. Projects, Inc. is known for its strong emphasis on change management and quality assurance procedures. The customer provides detailed information when requesting a document through a web portal. An engineer is assigned to write the first draft of the requested document. Upon completion, two peer engineers review the document to ensure that it is correct and meets the requirements. These reviewers may require changes or may approve the document as-is. The document is updated until the reviewers are satisfied with the quality of the document. The document is then sent to the customer for approval. The customer can require changes or accept the document. When the customer requires changes, an engineer is assigned to make the changes to the document. When those changes are made, two other engineers must review those changes. When those reviewers are satisfied with the changes, the document is sent back to the customer. This may happen through several iterations until the customer is satisfied with the document.

Context:



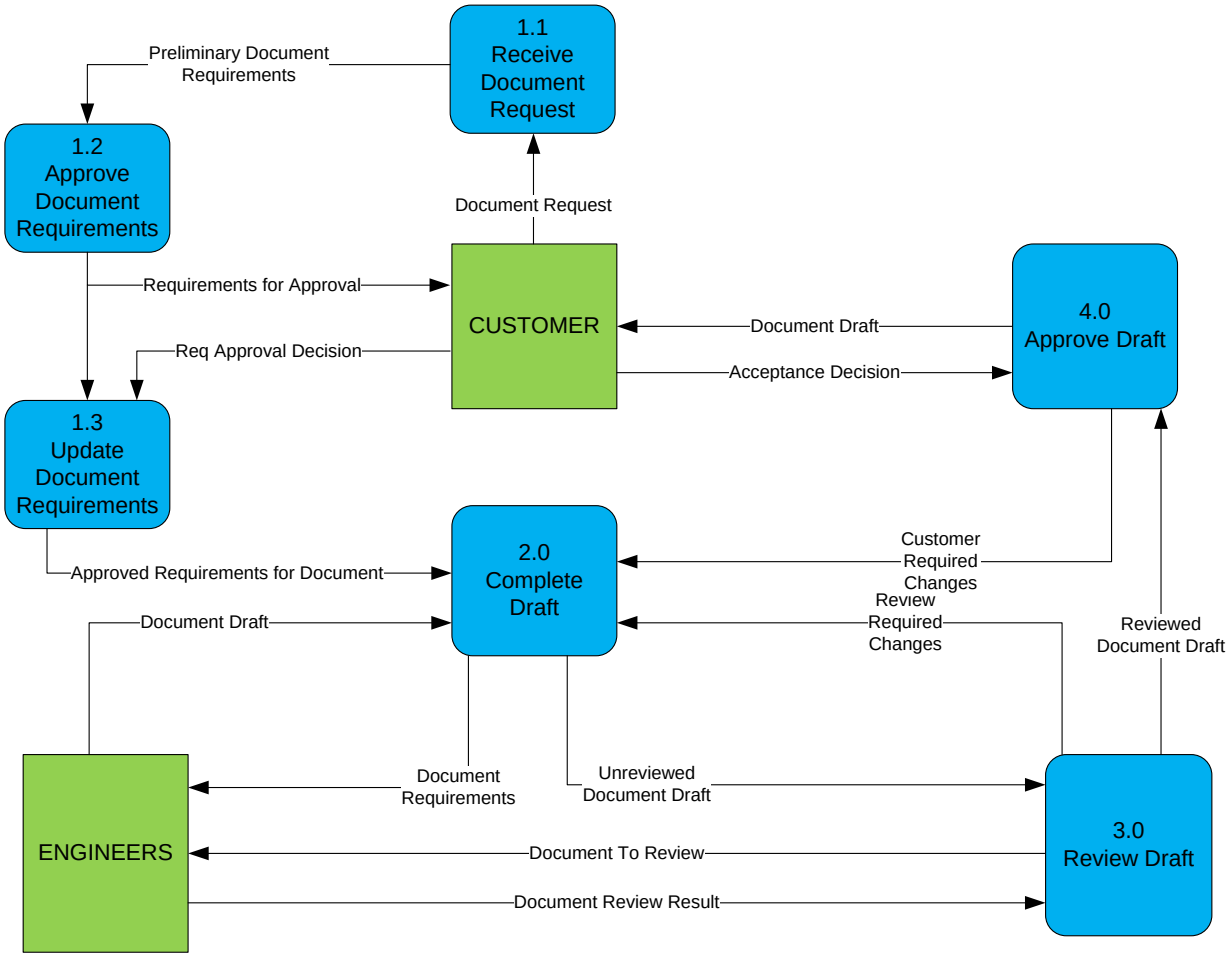
Level-0:



Analyze the DFDs you created above. What recommendations for improvements can you make based on this analysis? Draw new logical DFDs that represent the requirements you would suggest for an improved document management system. Remember, these are to be logical DFDs, so consider improvements independent of technology that can be used to support the management of these documents.

The DFDs show an inherent weakness in the processes used for the data to flow. The document may go through several revisions before the customer has seen any information from the document, thereby wasting a massive amount of effort. A better process would seek customer input on the requirements that are used as input into the 2.0 Complete Draft process. This could be done by extending the process using this as the Level-1 diagram. Note that this is not balanced with the context diagram above.

Level-1:



35. What characteristics of data are represented in an E-R diagram?

An E-R diagram shows many characteristics of data, including the definition, structure, and relationships within data. Additionally, this diagram shows cardinalities, relationship degrees, and business rules.

36. What elements of a data flow diagram should be analyzed as part of data modeling?

Data stores, data flows, and even processes all provide information for data modeling. A data store often represents one or more data entities and their associated attributes. All data in data flows must either be stored in some entity, be computed from data in entities, or in rare circumstances pass through the system. The description of a process can shed light on business rules that must be represented in the data model.

37. Explain why a ternary relationship is not the same as three binary relationships.

A ternary relationship represents the *simultaneous* association of three entities (such as a selling relationship links a customer with a product and salesperson), not three binary relationships (between a sale entity/associative entity and customer, sale and product, and sale and salesperson).

38. When must a many-to-many relationship be modeled as an associative entity?

A many-to-many relationship must be modeled as an associative entity when there are attributes associated with the relationship.

39. Which of the following types of relationships can have attributes associated with them: one-to-one, one-to-many, many-to-many?

One-to-one and many-to-many relationships (associative entities) may have attributes. For example, a one-to-one unary relationship between employees, Married to, may have a Date Married attribute, and a many-to-many binary relationship between students and courses, Takes, may have a Grade attribute.

40. What is the degree of a relationship? Give an example of each of the relationship degrees illustrated in this chapter.

The degree of a relationship indicates the number of entity types participating in a relationship. The three most common relationships are unary, binary, and ternary. An employee working for a department is an example of a binary relationship. A part composed of other parts is an example of a unary relationship. A customer placing an order with a salesperson is an example of a ternary relationship.

41. Give an example of a ternary relationship (different from any example in this chapter.)

An example of a ternary relationship might be that of a car service. A particular driver and car might be assigned to a particular client.

42. List the deliverables from conceptual data modeling.

The primary deliverable for the conceptual modeling part of analysis is an E-R diagram, showing the major categories of data and the business relationships between them. A full set of entries about data objects to be stored in the project repository is also produced.

43. Explain the relationship between minimum cardinality and optional and mandatory participation.

Minimum cardinality refers to the minimum number of instances of entity B that can be associated with entity A. If the minimum cardinality of B is one, then entity B is a mandatory participant in the relationship. However, if the minimum cardinality for entity B can be zero, then entity B can be thought of as an optional participant in the relationship.

44. List the ideal characteristics of an entity identifier attribute.

An identifier that meets the criteria set forth in the chapter would be an ideal choice. The criteria include: (1) choosing an attribute that will not change its value over the life of each entity type; (2) choosing an attribute that for each instance of the entity will have valid values and will not be null; (3) avoiding intelligent key usage; and (4) substituting surrogate keys for large composite keys.

45. List the four types of E-R diagrams produced and analyzed during conceptual data modeling.

E-R diagrams are produced: (1) to cover just the data needed in the project's application; (2) for the application system being replaced; (3) to document the entire database from which the new application's data are extracted; and (4) for the whole database from which data for the application being replaced are drawn.

46. What notation is used on an E-R diagram to show a lower-bound or upper-bound limit on the "many" side of a one-to-many relationship?

A fixed number, such as the number 6, is placed above or below the crow's foot notation next to the entity.

47. Explain the difference between a candidate key and the identifier of an entity type.

A candidate key is an attribute(s) that uniquely identifies each instance of an entity type. An identifier is a candidate key that has been *chosen* as the unique, identifying characteristic for that entity type.

48. What distinguishes a repeating group from a simple multivalued attribute?

A multivalued attribute is a single attribute that may legitimately assume more than one value for each entity instance. A repeating group is a set of two or more multivalued attributes that are logically related.

49. How do analysts generate alternative solutions to information systems problems?

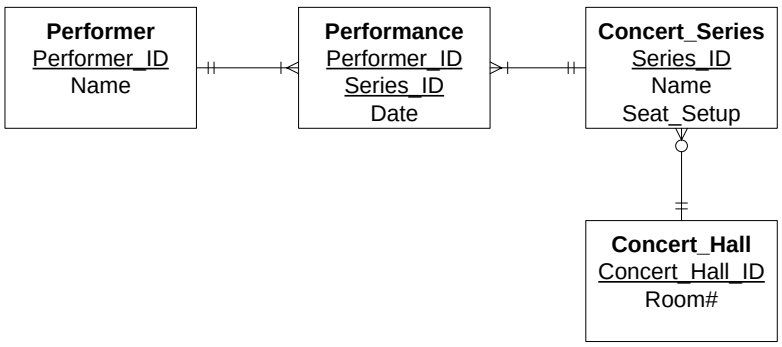
Analysts consider many issues in developing alternative solutions to information system problems. Of particular interest are the system owner's and users' prioritized system objectives and system (and development) constraints. Analysts consider which design strategies would minimally satisfy objectives and not violate constraints, on the one hand, as well as which design strategies would meet or exceed objectives with minimal violation of constraints on the other hand. There are many possible design strategies between these two extreme positions.

50. How do managers decide which alternative design strategy to develop?

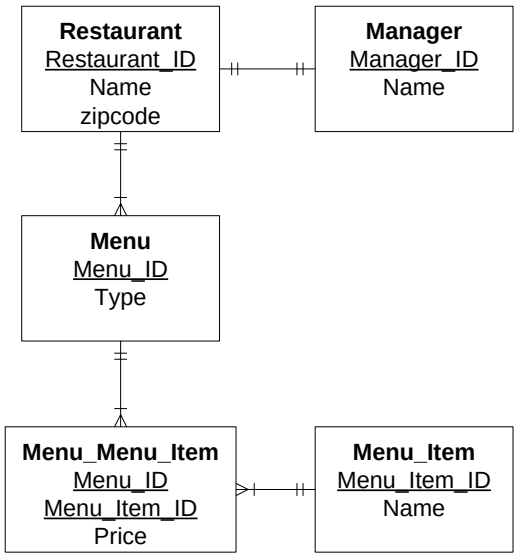
While alternative design strategies may be compared in many different objective ways, the actual design strategy chosen by management will depend on what management's true objectives are for a particular development project. Management may ignore constraints, or alternatively, choose the least expensive system to develop, regardless of which design strategy appeared to be the best in the objective comparison.

Some examples to analyze

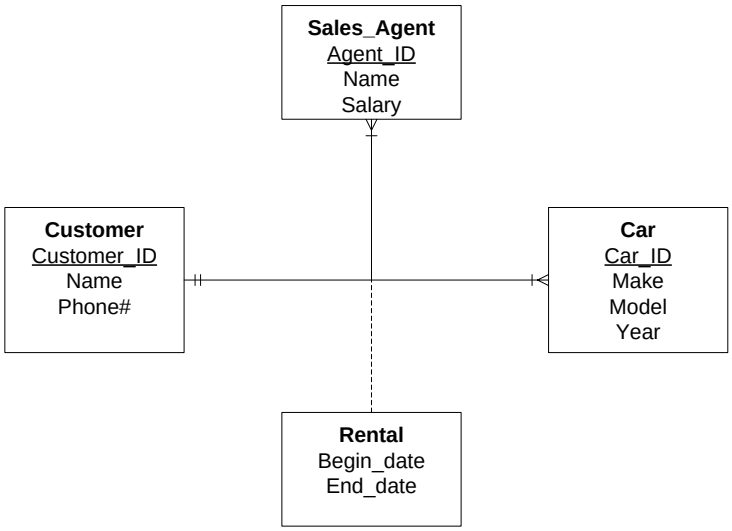
A performance venue hosts many concert series a year. Performers have a name and perform several times in a concert series (each constituting a performance with a different date). Concert series have one or more performers and have a name and a specified seating arrangement. A concert series is held in one (and only one) of several concert halls, each of which has a room number. Represent this situation of concerts and performers with an E-R diagram.



A restaurant chain has several store locations in a city (with a name and zipcode stored for each), and each is managed by one manager. Managers manage only one store. Each restaurant location has its own unique set of menus. Most have more than one menu (e.g., lunch and dinner menus). Each menu has many menu items, and items can appear on multiple menus, and with different prices on different menus. Represent this situation of restaurants with an E-R diagram.

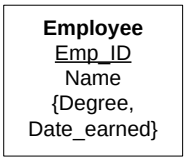


A car rental is an association between a customer, sales agent, and a car. Select a few pertinent attributes for each of these entity types and represent a rental in an E-R diagram.

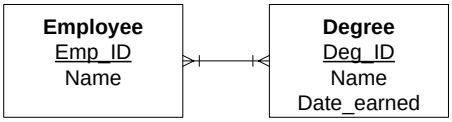


A company database contains an entity called EMPLOYEE. Among other information, the company records information about any degrees each employee has earned, along with the graduation date for the degree.

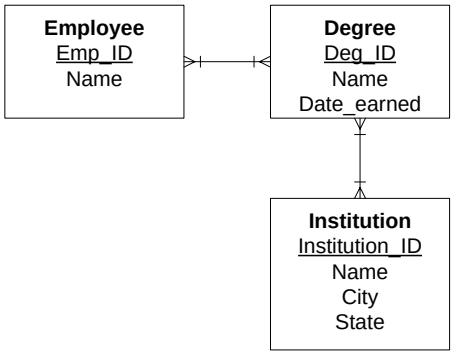
a. Represent the EMPLOYEE entity and its degree attributes using the notation for multivalued attributes.



b. Represent the EMPLOYEE entity and its degree attributes using two entity types.



c. Finally, assume the company decides to also keep data about the institution from which the employees' degrees were earned, including name of institution, city, and state where the institution is located. Augment you answer to part b to accommodate this new entity type.



Discuss why some systems developers believe that a data model is one of the most important parts of the statement of information system requirements.

Four reasons were provided in the textbook: (1) the characteristics of data captured during data modeling are crucial in the design of databases, programs, computer screens, and printed reports; (2) data rather than processes are the most complex aspects of many modern information systems; (3) the characteristics about data are permanent; and (4) structural information about data is essential to generate programs automatically

Now to communicate with the user

Describe the prototyping process of designing forms and reports. What deliverables are produced from this process? Are these deliverables the same for all types of system projects? Why or why not?

Designing forms and reports is a user-focused activity that typically follows a prototyping approach. First, you must gain an understanding of the intended user and task objectives by collecting initial requirements during requirements determination. After collecting the initial requirements, you structure and refine this information into an initial prototype. Structuring and refining the requirements are completed independent from the users, although you may need to occasionally contact users to clarify some issue overlooked during analysis. Finally, you ask users to review and evaluate the prototype. After reviewing the prototype, users may accept the design or request that changes be made. If changes are needed, you will repeat the construction-evaluate-refinement cycle until the design is accepted. Usually, several iterations of this cycle occur during the design of a single form or report. The major deliverables produced from the process of designing forms and reports are the creation of design specifications. A design specification includes a narrative overview, a sample design, and a testing and usability assessment section.

Some specification information may be irrelevant when designing some forms and reports. For example, the design of a simple "Yes/No" selection form may be so straightforward that no usability assessment would be needed. Also, much of the narrative overview may be unneeded, unless it was used to highlight some exception that must be considered during implementation.

2. **To which initial questions must the analyst gain answers to in order to build an initial prototype of a system output?**

   (1) who will use this form or report;
   (2) what is the purpose of the form or report;
   (3) when is the form or report needed and used;
   (4) where does the form or report need to be delivered and used; and
   (5) how many people need to use or view the form or report?

3. **How should textual information be formatted on a help screen?**

4. **What type of labeling can you use in a table or list to improve its usability?**

   The following labeling guidelines can improve the usability of a table or list: (1) all columns and rows should have meaningful labels; (2) labels should be separated from other information by using highlighting; and (3) redisplay labels when the data extends beyond a single screen or page.

5. **What column, row, and text formatting issues are important when designing tables and lists?**

   The guidelines for formatting columns, rows, and text in a table or list include: (1) sorting in a meaningful order (e.g., ascending, descending, or alphabetic); (2) placing a blank line between every five rows in long columns; (3) sorting vertically similar information displayed in multiple columns (that is, read from top to bottom,

not left to right); (4) placing at least two spaces between columns; (5) allowing white space on printed reports for the user to write notes; (6) using a single typeface, except for emphasis; (7) using the same family of typefaces within and across displays and reports; and (8) avoiding overly fancy fonts.

6. **Describe how numeric, textual, and alphanumeric data should be formatted in a table or list.**

   The guidelines for formatting numeric, textual, and alphanumeric data in a table or list include: (1) right justifying numeric data and aligning columns by decimal points or some other delimiter; (2) left justifying textual data and using a short line length, usually 30–40 characters per line (this is what newspapers use, and it is easier to speed read); and (3) breaking long sequences of alphanumeric data into small groups of three to four characters each.

7. **Provide some examples where variations in user, task, system, and environmental characteristics might impact the design of system forms and reports.**

   User-related concerns include experience, skills, motivation, education, and personality. Task-related concerns include time pressure, cost of errors, and work duration. The system platform is a system-related concern, and lighting, sound, and task interruptions are environment-related concerns.

8. **Describe the process of designing interfaces and dialogues. What deliverables are produced from this process? Are these deliverables the same for all types of system projects? Why or why not?**

   The design process is a user-focused activity that typically follows a prototyping approach. First, you must gain an understanding of the intended user and task objectives by collecting initial requirements during requirements determination. After collecting the initial requirements, you structure and refine this information into an initial prototype. Structuring and refining the requirements are completed independent from the users, although you may need to occasionally contact users to clarify some issue overlooked during analysis. Finally, you ask users to review and evaluate the prototype. After reviewing the prototype, users either accept the design or request changes. If changes are needed, you will repeat the construction-evaluate-refinement cycle until the design is accepted. Usually, several iterations of this cycle occur during the design of a single interface or dialogue. Design specifications are the primary deliverables; these documents include a narrative overview, sample design, testing and usability assessment, and dialogue sequence outline.

9. **List and describe the functional capabilities needed in an interface for effective entry and navigation. Which capabilities are most important? Why? Will this be the same for all systems? Why or why not?**

   The functional capabilities include cursor control capabilities, editing capabilities, exit capabilities, and help capabilities When designing the navigation procedures within your system, flexibility and consistency are primary concerns. Users should be able to freely move forward and backward or to any desired data entry fields. Consistency extends to the selection of keys and commands. Each key or command should have only one function, and this function should be consistent

throughout the entire system and across systems if possible. The most important capabilities will vary from system to system. For a given system, consistency is the most important capability. System capabilities will, unfortunately, vary from system to system.

10. Describe the general guidelines for structuring data-entry fields. Can you think of any instances when it would be appropriate to violate these guidelines?

Some of the design guidelines may be violated in certain circumstances. For example, a help system might be omitted from a very simple system or a system designed to be used by a single user or small group.

11. Describe four types of data errors.

Sources of data errors include appending, truncating, transcripting, and transposing. Appending adds additional characters to a field; truncating loses characters from a field; transcripting errors enter invalid data into a field; transposing reverses the sequence of one or more characters in a field.

12. Describe the types of system feedback. Is any form of feedback more important than the others? Why or why not?

Three types of system feedback are discussed in the chapter; these include status information, prompting cues, and error or warning messages. No single method can be deemed “most” important because each is critical to an effective design.

13. Describe the general guidelines for designing usable help. Can you think of any instances when it would be appropriate to violate these guidelines?

Some of the design guidelines may be violated in certain circumstances. For example, a help system might be omitted from a very simple system or a system designed to be used by a single user or small group.

14. What steps do you need to follow when designing a dialogue? Of the guidelines for designing a dialogue, which is most important? Why?

The three major steps in dialogue design are: (1) design the dialogue sequence; (2) build a prototype; and (3) assess usability. A case could be made for the importance of all steps. Yet, designing the dialogue (with the input of the user) is likely the most important step because this is where you collect and structure the users' requirements.

15. Describe what is meant by a *cookie crumb*. How do these help prevent users from getting lost?

A cookie crumb is a technique for showing users where they are in a Web site; a series of tabs are placed on a Web page showing the users where they are and where they have been. By showing users where they are in a Web site, cookie crumbs keep users from getting lost; users can just click on a hyperlink to return home.

16. Describe why you might want to use lightweight graphics on some Web pages and large detailed graphics on others.

Often users do not want to wait for large, color images to download, especially if they have a slow modem. If the time is excessive, the users will often leave the site. Lightweight graphics download much faster and give the user a thumbnail sketch of the product. If the user wishes to view a larger image, she can click on the thumbnail sketch to see a larger graphic.

17. Why is it especially important to eliminate data-entry errors on an electronic commerce Web site?

If data errors are caught quickly, this helps keep erroneous data from being permanently stored in the company's database.

18. How can template-based HTML help to make a large electronic commerce site more maintainable?

Template-based HTML processes and displays common attributes of higher-level, more abstract items. When a common attribute changes, this change requires updating only the template, as opposed to making many changes. For instance, if your company manufactures handbags, the styles of these handbags will differ but the available colors are probably the same. If a new color, suppose chocolate, is added then only the available colors require updating.

Consider a system that produces inventory reports at a local retailer. Alternatively, consider a system that produces student academic records for the records office at a university. For whichever system you choose, answer the following design questions: Who will use the output? What is the purpose of the output? When is the output needed, and when is the information that will be used within the output available? Where does the output need to be delivered? How many people need to view the output?

Inventory reports are used by managers and stocking employees, and perhaps store owners, depending on how large the retailer is. The output's purpose is to give an accurate picture of the levels of various inventory items, for purchasing or other decisions. The output would likely be needed at regular intervals (e.g., weekly), but would only be available after ordering and sales information is updated (so that updated inventory levels would display). Assuming this is a smaller retailer, the output would only be needed either on the computer screen, or perhaps as a printed report for reconciliation purposes. In a smaller retailer, the information is really only applicable to managers and a few employees in charge of stocking shelves.

Imagine the worst possible reports from a system. What is wrong with them? List as many problems as you can. What are the consequences of such reports? What could go wrong as a result? How does the prototyping process help guard against each problem?

The worst possible reports come so late that the information is be useless, the reports contain information that the user does not need, and they do not contain the information that the user really needs. The information presented has errors and is inconsistent from page to page and from report to report. The layout is difficult to read and understand, and the reports are inflexible, prohibiting the user from changing the information to his liking. As farfetched as this sounds, these types of

problems happen frequently with poorly designed reports.

The consequences are that the users of these reports will not be able to use the reports in any meaningful way. They ultimately become frustrated with the reports and, eventually, stop looking at them. If they are unable to find the information they need in some other way, their decision making and performance suffers. Ultimately, the organization's performance suffers. The prototyping process helps guard against these problems because users and system personnel can work together to ensure that the reports are useful; they present useful information in an effective format, and they are delivered in a timely manner.

Given the guidelines presented in this chapter, identify flaws in the design of the Report of Employees shown. What assumptions about users and tasks did you make in order to assess this design? Redesign this report to correct these flaws.

```
Report of Employees-1-2-08

Em_ID       Name, Title

0124543      John Smith, VP
               Marketing

2345645      Jared Wright,
            Project Manager

2342456     Jennifer Chang,
            Systems Analyst

4564234      Mark Walters,
               Software
               Engineer

7875468     Nick Shelley, BI
               Analyst

4446789      Kim Eagar, HR
               Manager

4678899      Emily Graham,
             Receptionist

4452378      Matt Hoffman,
               Network
              Operations
              Specialist
```

The report lacks a clear title, meaningful column labels consistent spacing of rows, logical

sequence of information, proper segregation of column data, and left-justification of alphabetic text. The long ID numbers are hard to read. The data is also ambiguous (it could be M-D-YY or D-M-YY).

Employee Report			**Jan 02, 2008**
Employee ID	**Last Name**	**First Name**	**Title**
0-124-543	Chang	Jennifer	VP Marketing
2-345-645	Eagar	Kim	Project Manager
2-342-456	Graham	Emily	Systems Analyst
4-564-234	Hoffman	Matt	Software Engineer
7-875-468	Shelley	Nick	BI Analyst
4-446-789	Smith	John	HR Manager
4-678-899	Walters	Mark	Receptionist
4-452-378	Wright	Jared	Network Operations Engineer

Discuss the differences between a form and a report. What characteristics make a form or report good (bad) and effective (ineffective)?

A form is a business document that contains some predefined data and may include some areas where additional data are to be filled in. In contrast, a report is a business document that contains only predefined data. A properly designed form or report conforms to the guidelines presented in Table 8–2, meaning the form or report uses meaningful titles, includes meaningful information, balances the layout, and is easy to navigate. In contrast, a poorly designed form or report violates one or more of these guidelines.

Data bases

1. **What is the purpose of normalization?**

   The purpose of normalization is to rid relations of anomalies. The goal is to form well-structured relations that are simple and stable when data values change or data are added or deleted.

2. **List five properties of relations.**

   The five properties of relations are: (1) entries in cells are simple, (2) entries in columns are from the same set of values, (3) each row is unique, (4) the sequence of columns is insignificant, and (5) the sequence of rows is insignificant.

3. **What problems can arise during view integration or merging relations?**

   Synonyms, homonyms, and dependencies between nonkeys can arise during view integration. Synonyms occur when two or more different names are used for the same attribute from different user views. Homonyms occur when two or more attributes from different user views have the same name. Functional dependencies between nonkey attributes arise when functionally dependent nonkeys come from different user views.

4. **How are relationships between entities represented in the relational data model?**

   Relationships between entities are represented in several ways in the relational data model. A binary 1:M relationship is represented by placing a foreign key (the primary key of the entity on the one-side of the relationship) in the relation for the entity on the many-side of the relationship. In a binary 1:1 relationship, a foreign key is placed in the relation on either side of the relationship or on both sides. For a binary and higher degree M:N relationship, a relation is created with a primary key which is the concatenation of the primary keys from the related entities. In a unary relationship, a recursive foreign key is added to the relation.

5. **What is the relationship between the primary key of a relation and the functional dependencies among all attributes within that relation?**

   The fundamental rule of normalization is that each non-key attribute must be a fully functionally dependent on the whole primary key attribute (a nonkey is dependent on the whole key and nothing but the key). Thus, there can be no functional dependencies between nonkeys.

6. **How is a foreign key represented in relational notation?**

   A foreign key is identified by using a dashed underline.

7. **Can instances of a relation (sample data) prove the existence of a functional dependency? Why or why not?**

   Instances in a relation *cannot* prove that a functional dependency exists; however, you can use sample data to demonstrate that a functional dependency *does not* exist. The sample data does not show you every possible instance, only a sampling. Knowledge of the problem domain is a reliable method for identifying functional dependencies.

8. **In what way does the choice of a data type for a field help to control the integrity of that field?**

   The choice of data type often limits the possible values that may be stored for a field. For example, a numeric data type forbids alphabetic characters. Some data types have an assumed length, which places an implicit range control on values. Data type may also limit the kinds of data manipulations possible, thus further controlling the integrity of the data or results from manipulating the data. For example, a DATE data type causes addition and subtraction to be limited by rules about dates.

9. **What is the difference between how a range control statement and a**

referential integrity control statement are handled by a file management system?

Range controls identify a set of permissible values, such as college ranking (senior, junior, sophomore, and freshman). Referential integrity requires a foreign key to match a primary key in another relation. Alternatively, referential integrity would require a recursive foreign key to match the primary key of that same relation. A referential integrity control requires that the data management software access other data records to determine if the value is permitted, whereas a range control is checked by looking up values outside the files and database, in a repository or other source of metadata.

10. What is the purpose of denormalization? Why might you not want to create one physical table or file for each relation in a logical data model?

Denormalization provides for more efficient data processing. Denormalization enables data to be stored in physical tables based on affinity of use; this feature reduces the number of I/O operations.

11. What factors influence the decision to create an index on a field?

The factors that influence the decision to create an index are the data retrieval, insertion, deletion, and updating costs with and without the index. Indexes allow for rapid random retrieval and sorting of data, but indexes create additional storage and maintenance costs. The guidelines presented in the chapter offer suggestions for when to use an index; these guidelines include specifying a unique index for the primary key of each table; specifying an index for foreign keys, and specifying an index for nonkey fields that are referenced in qualification, sorting, and grouping commands for the purpose of retrieving data.

12. Explain the purpose of data compression techniques.

Coding and compression techniques reduce storage space and can increase data integrity. Data compression techniques are pattern matching and other methods that replace repeating strings of characters with codes of shorter lengths, thus reducing data storage requirements.

13. What are the goals of designing physical tables?

The two goals mentioned in the chapter are efficient use of secondary storage space and data processing speed.

14. What are the seven factors that should be considered in selecting a file organization?

When selecting a file organization, the analyst should consider fast data retrieval, high throughput for processing transactions, efficient use of storage space, protection from failures or data loss, minimizing need for reorganization, accommodating growth, and security from unauthorized use.

15. What are the four steps in logical database modeling and design?

The four steps in logical database modeling and design are: (1) develop a logical data model for each known user interface for the application using normalization principles; (2) combine normalized data requirements from all user interfaces into

one consolidated logical database model; (3) translate the conceptual E-R data model for the application into normalized data requirements, and (4) compare the consolidated logical database design with the translated E-R model and produce one final logical database model for the application.

16. **What are the four steps in transforming an E-R diagram into normalized relations?**

    The four steps are (1) represent entities; (2) represent relationships; (3) normalize relations; and (4) merge the relations.

Mini cases

Consider the list of individual 3NF relations that follow. These relations were developed from several separate normalization activities.

PATIENT(Patient_ID, Room_Number, Admit_Date, Address)
ROOM(Room_Number, Phone, Daily_Rate)
PATIENT(Patient_Number, Treatment_Description, Address)
TREATMENT(Treatment_ID, Description, Cost)
PHYSICIAN(Physician_ID, Name, Department)
PHYSICIAN(Physician_ID, Name, Supervisor_ID)

a. **Merge these relations into a consolidated set of 3NF relations. Make and state whatever assumptions you consider necessary to resolve any potential problems you identify in the merging process.**

b. **Draw an E-R diagram for your answer to part a.**

Listed below are merged 3NF relations for this hospital example. This is an interesting exercise because it points out how semantically lacking the relational data model is, since questions arise about functional dependencies across separately developed relations. One observation is clear: The second 3NF PATIENT relation in the exercise has only one value of TREATMENT DESCRIPTION, so each patient must be associated with only one treatment, otherwise this relation would not be in 3NF. But, we must also assume that each department has only one supervisor and each supervisor can supervise only one department. This last assumption means that we could create only a supervisor or a department relation, but not both. This is sufficient because, if the original set of six 3NF relations is comprehensive, there are no nonkey attributes dependent on either DEPARTMENT or SUPERVISOR ID. We, however, create both supervisor and department relations, with a one-to-one relationship between them, to allow for some evolution of the data model. One additional assumption about supervisors: SUPERVISOR is a separate entity from PHYSICIAN. We also assume that the attribute ADDRESS means the same address in both PATIENT relations, and further there are no other synonyms or homonyms across the relations. Interestingly, there is no relationship between patient and physician implied in the

original 3NF relations, and we assume none exists. With these assumptions, the merged relations are (foreign keys are shown with a dashed underline):

PATIENT (Patient Number, Address, Room No, Admit Date, Treatment ID)
ROOM (Room No, Phone, Daily Rate)
PHYSICIAN (Physician ID, Name, Department ID)
TREATMENT (Treatment ID, Description, Cost)
SUPERVISOR (Supervisor ID, Department ID)
DEPARTMENT (Department ID, Supervisor ID)

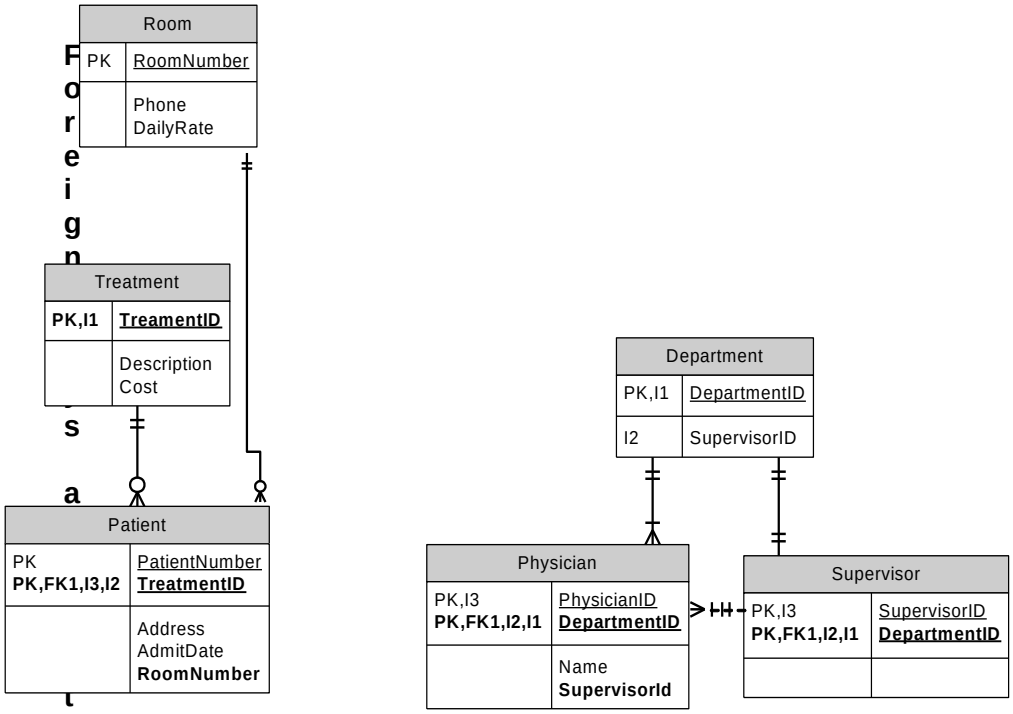
To create the E-R diagram from these 3NF relations, we have to make additional assumptions about minimum cardinalities. We assume that every patient is assigned a room, but a room may be empty; not all treatments have to be associated with a patient, but a patient has to have a treatment; and that each department has one supervisor and each supervisor has one department. We show relationships from both a department and a supervisor to a physician, but only one is necessary; we also assume that a physician must be associated with both a department and a supervisor. This is an interesting E-R diagram since it contains two, disconnected parts. This is possible, although rare in actual organizations. The following is an E-R diagram, including attributes, for this situation. Microsoft Visio was used to create the E-R diagram.

5. Consider the following 3NF relations about a sorority or fraternity:

MEMBER(Member_ID, Name, Address, Dues_Owed)
OFFICE(Office_Name, Officer_ID, Term_Start_Date, Budget)
EXPENSE(Ledger_Number, Office_Name, Expense_Date, Amt_Owed)
PAYMENT(Check_Number, Expense_Ledger_Number, Amt_Paid)
RECEIPT(Member_ID, Receipt_Date, Dues_Received)
COMMITTEE(Committee_ID, Officer_in_Charge)
WORKERS(Committee_ID, Member_ID)

Problem and Exercise #4

a. For each foreign key indicated in these relations. Decide which attributes are foreign keys and justify your decisions.



b. Draw an E-R diagram for these relations, using your answer to part a.

c. Explain the assumptions you made about cardinalities in your answer to part b. Explain why it is said that the E-R data model is more expressive or more semantically rich than the relational data model.

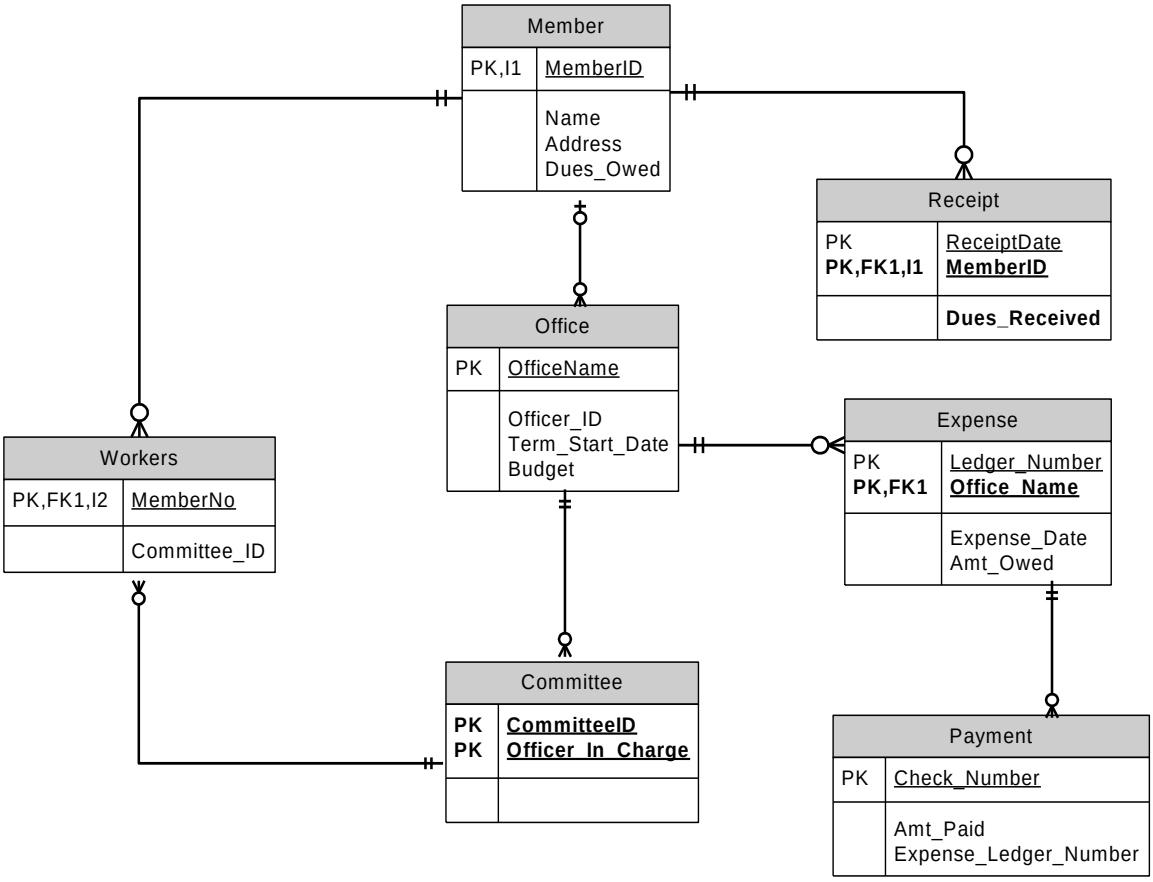
There are several foreign keys in these relations. OFFICER_ID is a foreign key in OFFICE referencing MEMBER_ID from the MEMBER relation. OFFICE_NAME is a foreign key in EXPENSE referencing OFFICE_NAME in the OFFICE relation. OFFICER_IN_CHARGE is a foreign key in COMMITTEE referring to OFFICER_ID or OFFICE_NAME (which is not clear from simply the relations) in the OFFICE

relation. EXPENSE_LEDGER_NUMBER is a foreign key in PAYMENT referencing LEDGER_NUMBER in the EXPENSE relation. MEMBER_ID in both RECEIPT and WORKERS cross references MEMBER_ID in the MEMBER relation. COMMITTEE_ID in WORKERS cross references COMMITTEE_ID in COMMITTEE.

It is inferred that a member sometimes has many receipts, but a receipt must have a member. An expense sometimes has multiple payments, but each payment must have an expense. Each office sometimes has multiple expenses, but each expense must have an office. Each office may have a member as an officer–in–charge, and each member sometimes holds many offices. An office sometimes is responsible for many committees, and each committee must have an office in charge (although that office may not have a member assigned as officer). Committees sometimes have many workers, and each worker sometimes works on many committees. The E-R diagram is more expressive in that it displays explicitly the minimum cardinalities of relationships and shows exactly which entities are related.

A suggested E-R diagram is provided below. Microsoft Visio was used to prepare this diagram.

Problem and Exercise 5



6. Consider the following functional dependencies:

Applicant_ID →Applicant_Name
Applicant_ID →Applicant_Address
Position_ID →Position_Title
Position_ID →Date_Position_Opens
Position_ID →Department
Applicant_ID + Position_ID→Date_Applied
Applicant_ID + Position_ID + Date_Interviewed →

a. Represent these attributes with 3NF relations. Provide meaningful relation names.

b. Represent these attributes using an E-R diagram. Provide meaningful entity and relationship names.

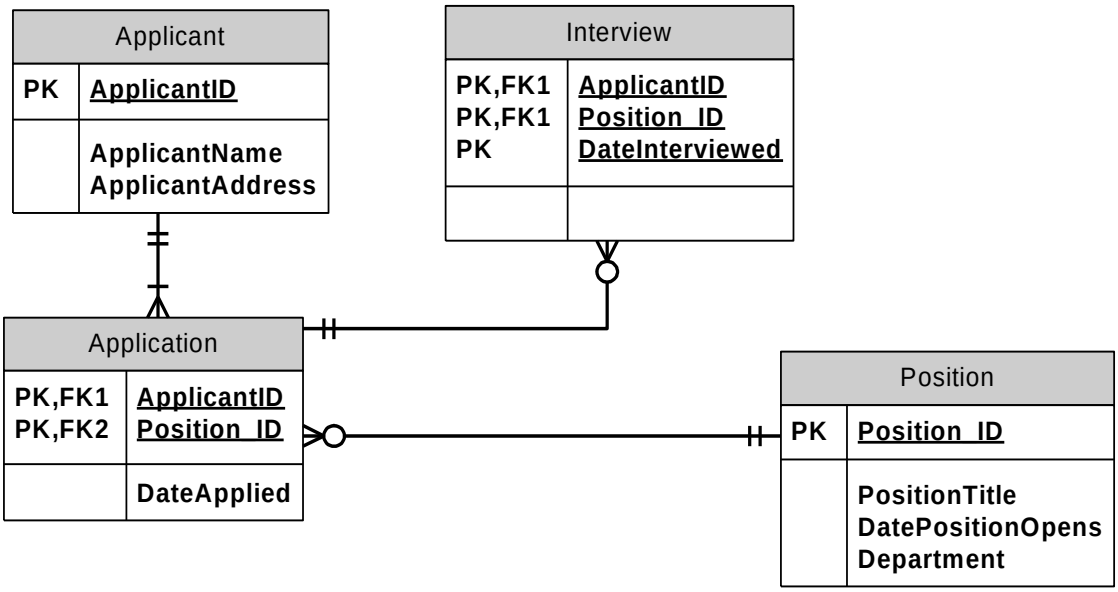
Since there are four determinants among the functional dependencies, there will be four relations. The last functional dependency, the one with only a three-key composite determinant, signifies all the dates on which a particular applicant interviewed for a particular position. This functional dependency does not signify a many-to-many relationship, like many composite keys do, since date interviewed is

itself not a determinant. It signifies an entity with a three-component composite key. The four 3NF relations are:

APPLICANT (Applicant ID, Applicant Name, Applicant Address)
POSITION (Position ID, Position Title, Date Position Opens, Department)
APPLICATION (Applicant ID, Position ID, Date Applied)
INTERVIEW (Applicant ID, Position ID, Date Interviewed)

See the accompanying E-R diagram corresponding to these four relations. Microsoft Visio was used to prepare the following E-R diagram.

Problem and Exercise #6



Many database management systems offer the ability to enforce referential integrity. Why would using such a feature be a good idea? Are there any situations in which referential integrity might not be important?

If the DBMS offers this ability, it should always be used. The possibility that referential integrity will never be violated as engineers interact with a database is very low, and having the system enforce the principle will reduce human error. There should be no situations in which referential integrity is not important.

Assume you are part of the systems development team at a medium-sized organization. You have just completed the database design portion of the systems design phase, and the project sponsor would like a status update. Assuming the project sponsor is a VP in the marketing department, with only a high-level understanding of technical subjects, how would you go about presenting the

database design you have just completed? How would your presentation approach change if the project sponsor were the manager of the database team?

Answers will vary. Presenting to a non-technical manager should be much more high-level, with few if any technical details. The focus should be on how the database design will benefit the business (for example, with faster report generation). Presenting to a technical manager would be focused much more on the technical aspects of the database, with less of a focus on the business benefits of the design.

At the end the problem should be solved

1.What are the deliverables from coding, testing, and installation?

The deliverables from coding, testing, and installation are: (1) from coding, code and program documentation; (2) from testing, test scenarios and test data and the results of program and system testing; and (3) from installation, user guides, user training plans, and installation plans.

2. Explain the testing process for code.

The testing process involves testing code for errors and functionality. The testing process, guided by a detailed testing plan, can begin as soon as modules are coded and proceeds in parallel with the rest of the coding process. Modules are tested individually and then as parts of larger programs and parts of larger systems.

3. What are the four approaches to installation? Which is the most expensive? Which is the most risky? How does an organization decide which approach to use?

The four approaches to installation are direct, parallel, single location, and phased. With the direct approach, the old system is turned off, and the new one is turned on. In the parallel approach, both systems are run until management decides the old system can be turned off. In single location, the system is tried out in a pilot project at one location and then implemented elsewhere if the pilot is successful. With the phased approach, the new system is brought online gradually, usually function by function. Parallel is usually the most expensive due to the redundant costs, and direct is usually the most risky due to the potential hazards if the new system fails. An organization decides which approach to use depending on the scope and complexity of the change and the organization's risk aversion.

4. List and define the factors that are important to successful implementation efforts.

Several factors are important to the successful implementation of a system, including management support, the involvement of users, and user expectations. Additionally, commitment to the project, commitment to change, and extent of project definition and planning are important. Commitment to the project refers to managing the project so that the problem being solved is well understood and the target system actually solves the problem. Commitment to change means being

willing to change behaviors and procedures. The premise of project definition and planning is that more extensive planning is more effective than less extensive planning.

5. What is the difference between system documentation and user documentation?

System documentation is the detailed information about a system's design specifications, its internal workings, and its functionality, whereas user documentation consists of written or other visual information about an application system, how it works, and how to use it.

6. List and define the various methods of user training.

Methods of training include resident experts, computer-aided instruction, formal courses, software help components, tutorials, interactive training manuals, and external sources. Descriptions of these methods are provided in the chapter.

7. Describe the delivery methods many vendors employ for providing support.

Vendors provide support through the following mechanisms: automated support, such as online support forums; bulletin board systems; on-demand fax; voice-response systems; help desk support; and technical support 800 telephone units, all or some of which can be either external or internal to the user organization.

8. List the steps in the maintenance process and contrast them with the phases of the systems development life cycle.

Four major activities occur within maintenance; these are obtaining maintenance requests, transforming requests into changes, designing changes, and implementing changes. The first phase of the SDLC, systems planning and selection, is analogous to the maintenance process of obtaining a maintenance request. The systems analysis phase is analogous to the maintenance process of transforming requests into a specific system change. Systems design is similar to the designing changes process. Finally, the systems implementation and operations phase is similar to implementing changes. This similarity between the maintenance process and the SDLC is no accident. The concepts and techniques used to initially develop a system are also used to maintain it.

9. What are the different types of maintenance and how do they differ?

Corrective, adaptive, perfective, and preventive are the four types of maintenance. Corrective maintenance is concerned with repairing design and programming errors; adaptive maintenance modifies the system to reflect environmental changes; perfective maintenance evolves the system to solve new problems or take advantage of new opportunities; preventive maintenance safeguards the system from future problems.

10. Describe the factors that influence the cost of maintenance. Are any factors more important? Why?

Latent defects, number of customers for a given system, quality of system documentation, maintenance personnel, tools, and well-structured programs are factors that influence the cost of maintenance. Of these factors, three were

described as being very important: defects, customers, and documentation. The number of latent defects refers to the number of unknown errors existing in the system after it is installed. Because corrective maintenance accounts for most maintenance activity, the number of latent defects in a system influences most of the costs associated with maintaining a system. If there are no errors in the system after it is installed, then maintenance costs will be relatively low. If there are a large number of defects in the system when it is installed, maintenance costs will likely be high. A second factor influencing maintenance costs is the number of customers for a given system. In general, the greater the number of customers, the greater the maintenance costs. A third major contributing factor to maintenance costs is the quality of system documentation. Without quality documentation, maintenance efforts can increase exponentially.

11. What types of measurements must be taken to gain an understanding of the effectiveness of maintenance? Why is tracking mean time between failures an important measurement?

To measure effectiveness, you must measure the number of failures, time between each failure, and type of failure. Measuring the number and time between failures will provide you with the basis to calculate a widely used measure of system quality. This metric is referred to as the mean time between failures (MTBF). As its name implies, the MTBF measure shows the average length of time between the identification of one system failure to the next. Over time, you should expect the MTBF value to rapidly increase after a few months of use (and corrective maintenance) of the system. If the MTBF does not rapidly increase over time, it will be a signal to management that major problems exist within the system that are not being adequately resolved through the maintenance process.

12. Describe the process for controlling maintenance requests. Should all requests be handled in the same way or are there situations when you should be able to circumvent the process? If so, when and why?

The chapter presents a flowchart that suggests one possible method for dealing with maintenance change requests. This chart suggests that you must determine the type of request. If, for example, the request is an error—that is, a corrective maintenance request—then the flowchart shows that a question must be asked related to the error's severity. If the error is “very” severe, then the request has top priority and is placed at the top of a queue of tasks waiting to be performed on the system. If, however, the error is considered not very severe, then the change request can be categorized and prioritized based upon its type and relative importance. If the change request does not concern an error, then you must determine if the request is to adapt the system to technology changes and/or business requirements or to enhance the system so it will provide new business functionality. For adaptation requests, they too will need to be evaluated, categorized, prioritized, and placed in the queue. For enhancement type requests, they must first be evaluated to see if they are aligned with future business and information systems plans. If not, the request will be rejected and the requester will be informed. If the enhancement appears to align with business and information systems plans, it is then prioritized and placed into the queue of future tasks.

The method for reviewing maintenance requests strongly suggests that all requests should not be handled in the same way. The process also suggests what should be

done with different types of requests. In situations where there is a catastrophic failure of the system, time may be of the essence, and a formal review of a request will take too long. In such a situation, the formal evaluation process is circumvented.

13. What is meant by *configuration management*? Why do you think organizations have adopted the approach of using a systems librarian?

Configuration management is the process of assuring that only authorized changes are made to a system. A system librarian controls the baseline source code modules. If maintenance personnel are assigned to make changes to a system, they must first check-out a copy of the baseline system modules because no one can directly modify the baseline modules. This ensures that only those modules that have been checked-out and then formally checked-in can reside in the library. Organizations have adopted this approach so that before any code can be checked back into the librarian, they must pass the quality control procedures, testing, and documentation standards established by the organization.

Consider the reasons implementations fail. For at least three of these reasons, explain why this happens, if there is one (or more) type of implementation likely to minimize the occurrence, and if there is one (or more) type of installation more likely to induce failure for this reason.

There are several reasons for failed implementation, some of which can be exacerbated by the methods chosen for installation. Some of the reasons include:

- Lack of commitment to the project – Management and employee commitment are important to the successful implementation of an information system. Parallel installation can signal a lack of commitment by management. Direct installation, on the other hand, shows a complete confidence in making the system a success. Phased and single location installations can demonstrate successes, increasing commitment to the project so that the benefits of the new information system can be extended, either across all locations (in a single location installation) or across all functions (in a phased installation).
- Lack of commitment to change – Much like commitment to the project, users must also be willing to change how they accomplish their work to include the new information system. Parallel implementation is more likely to leave users wishing to use the old system which could eventually lead to a failed implementation. Direct installation makes it more difficult to choose not to change, as restoring the old system can be difficult, time-consuming, and costly. Single location installation and phased installation can increase commitment by showing early successes with the system as a motivation for change in the other users.
- Poorly conceptualized system – Systems may fail because they were not properly conceptualized or developed. This indicates a serious lack of planning and project definition. The form of installation would not be likely to impact this cause for failure.
- Unrealistic user expectations – Users may not completely understand what the information system can and cannot do. When the users form unrealistic

expectations, they are guaranteed to be disappointed by the system. Both direct and parallel installations are likely to enhance these feelings of disappointment; parallel installations let users see how the new system doesn't solve all of the problems of the old system while direct installation may lead users to a feeling of nostalgia for the old system. Phased and pilot installations can be used to educate users as to what the new system will (and will not) do.

- Users refuse to use the system – Users may simply refuse to use the system. This however usually happens for one of the other reasons listed. It can also happen because of characteristics of the users themselves. For instance, workers who have done their job the same way for 20 years may be resistant to the change irrespective of the information system involved. The installation method is not likely to influence these user-centric characteristics.

  Additionally, characteristics of the information system may make use of the system unpalatable. Cumbersome interfaces and error-riddled software can lead to low satisfaction. Direct installation is more likely to cause problems as phased, parallel, and pilot installations are more likely to discover issues before the product is rolled out to all users or when there is a backup readily at-hand (in the case of parallel).

- Political reasons – Implementation may fail because of the misalignment of personal goals of politically powerful people. In such cases, the method of installation is unlikely to impact the outcome.