Answers to Review Questions

1. **What is an information system? What is its purpose?**

   An information system is a system that
   - provides the conditions for data collection, storage, and retrieval
   - facilitates the transformation of data into information
   - provides management of both data and information.

   An information system is composed of hardware, software (DBMS and applications), the database(s), procedures, and people.

   Good decisions are generally based on good information. Ultimately, the purpose of an information system is to facilitate good decision making by making relevant and timely information available to the decision makers.

2. **How do systems analysis and systems development fit into a discussion about information systems?**

   Both systems analysis and systems development constitute part of the Systems Development Life Cycle, or SDLC. Systems analysis, phase II of the SDLC, establishes the need for and the extent of an information system by
   - Establishing end-user requirements.
   - Evaluating the existing system.
   - Developing a logical systems design.

   Systems development, based on the detailed systems design found in phase III of the SDLC, yields the information system. The detailed system specifications are established during the systems design phase, in which the designer completes the design of all required system processes.
3. **Discuss the distinction between data and information**

Data is raw facts – facts that have not been yet been processed to reveal their meaning. Information represents data which has been placed in a meaningful context so as to reveal its meaning.

4. **What does the acronym SDLC mean, and what does an SDLC portray?**

SDLC is the acronym that is used to label the System Development Life Cycle. The SDLC traces the history of a information system from its inception to its obsolescence. The SDLC is composed of six phases: planning, analysis, detailed system, design, implementation and maintenance.

5. **What does the acronym DBLC mean, and what does a DBLC portray?**

DBLC is the acronym that is used to label the Database Life Cycle. The DBLC traces the history of a database system from its inception to its obsolescence. Since the database constitutes the core of an information system, the DBLC is concurrent to the SDLC. The DBLC is composed of six phases: initial study, design, implementation and loading, testing and evaluation, operation, and maintenance and evolution.

6. **Discuss the distinction between centralized and decentralized conceptual database design.**

Centralized and decentralized design constitute variations on the bottom-up and top-down approaches we discussed in the third question presented in the discussion focus. Basically, the centralized approach is best suited to relatively small and simple databases that lend themselves well to a bird's-eye view of the entire database. Such databases may be designed by a single person or by a small and informally constituted design team. The company operations and the scope of its problems are sufficiently limited to enable the designer(s) to perform all of the necessary database design tasks:

1. Define the problem(s).
2. Create the conceptual design.
3. Verify the conceptual design with all user views.
4. Define all system processes and data constraints.
5. Assure that the database design will comply with all achievable end user requirements.
The centralized design procedure thus yields the design summary shown in Figure Q9.5A.

**Figure Q9.5A The Centralized Design Procedure**

Note that the centralized design approach requires the completion and validation of a single conceptual design.

In contrast, when company operations are spread across multiple operational sites or when the database has multiple entities that are subject to complex relations, the best approach is often based on the decentralized design.

Typically, a decentralized design requires that the design task be divided into multiple modules, each one of which is assigned to a design team. The design team activities are coordinated by the lead designer, who must aggregate the design teams' efforts.

Since each team focuses on modeling a subset of the system, the definition of boundaries and the interrelation between data subsets must be very precise. Each team creates a conceptual data model corresponding to the subset being modeled. Each conceptual model is then verified individually against the user views, processes, and constraints for each of the modules. After the verification process has been completed, all modules are integrated in one conceptual model.

Since the data dictionary describes the characteristics of all the objects within the conceptual data model, it plays a vital role in the integration process. Naturally, after the subsets have been aggregated into a larger conceptual model, the lead designer must verify that the combined conceptual model is still able to support all the required transactions. Thus the decentralized design activities may be summarized as shown in Figure Q8.6B.
Keep in mind that the aggregation process requires the lead designer to assemble a single model in which various aggregation problems must be addressed:

- **synonyms and homonyms**. Different departments may know the *same object by different names* (synonyms), or they may use the *same name to address different objects* (homonyms). The object may be an entity, an attribute, or a relationship.
- **entity and entity subclasses**. An entity subset may be viewed as a separate entity by one or more departments. The designer must integrate such subclasses into a higher-level entity.
- **Conflicting object definitions**. Attributes may be recorded as different types (character, numeric), or different domains may be defined for the same attribute. Constraint definitions, too, may vary. The designer must remove such conflicts from the model.

7. **What is the minimal data rule in conceptual design? Why is it important?**

The minimal data rule specifies that all the data defined in the data model are actually required to fit present and expected future data requirements. This rule may be phrased as *All that is needed is there, and all that is there is needed.*
8. Discuss the distinction between top-down and bottom-up approaches to database design.

We have addressed this question in detail in the discussion focus segment.

9. What are business rules? Why are they important to a database designer?

Business rules are narrative descriptions of the business policies, procedures, or principles that are derived from a detailed description of operations. Business rules are particularly valuable to database designers, because they help define:

- Entities
- Attributes
- Relationships (1:1, 1:M, M:N, expressed through connectivities and cardinalities)
- Constraints

To develop an accurate data model, the database designer must have a thorough and complete understanding of the organization's data requirements. The business rules are very important to the designer because they enable the designer to fully understand how the business works and what role is played by data within company operations.

**NOTE**

Do keep in mind that an ERD cannot always include all the applicable business rules. For example, although constraints are often crucial, it is often not possible to model them. For instance, there is no way to model a constraint such as “no pilot may be assigned to flight duties more than ten hours during any 24-hour period.”

It is also worth emphasizing that the description of (company) operations must be done in almost excruciating detail and it must be verified and re-verified. An inaccurate description of operations yields inaccurate business rules that lead to database designs that are destined to fail.

10. What is the data dictionary's function in database design?

A good data dictionary provides a precise description of the characteristics of all the entities and attributes found within the database. The data dictionary thus makes it easier to check for the existence of synonyms and homonyms, to check whether all attributes exist to support required reports, to verify appropriate relationship representations, and so on. The data dictionary's contents are both developed and used during the six DBLC phases:

**DATABASE INITIAL STUDY**

The basic data dictionary components are developed as the entities and attributes are defined during this phase.

**DATABASE DESIGN**

The data dictionary contents are used to verify the database design components: entities, attributes, and
their relationships. The designer also uses the data dictionary to check the database design for homonyms and synonyms and verifies that the entities and attributes will support all required query and report requirements.

IMPLEMENTATION AND LOADING
The DBMS's data dictionary helps to resolve any remaining attribute definition inconsistencies.

TESTING AND EVALUATION
If problems develop during this phase, the data dictionary contents may be used to help restructure the basic design components to make sure that they support all required operations.

OPERATION
If the database design still yields (the almost inevitable) operational glitches, the data dictionary may be used as a quality control device to ensure that operational modifications to the database do not conflict with existing components.

MAINTENANCE AND EVOLUTION
As users face inevitable changes in information needs, the database may be modified to support those needs. Perhaps entities, attributes, and relationships must be added, or relationships must be changed. If new database components are fit into the design, their introduction may produce conflict with existing components. The data dictionary turns out to be a very useful tool to check whether a suggested change invites conflicts within the database design and, if so, how such conflicts may be resolved.

11. What steps are required in the development of an ER diagram? (Hint: See Table 9.1.)

Table 9.1 is reproduced for your convenience.

**TABLE 9.1 Developing the Conceptual Model, Using ER Diagrams**

<table>
<thead>
<tr>
<th>STEP</th>
<th>ACTIVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Identify, analyze, and refine the business rules.</td>
</tr>
<tr>
<td>2</td>
<td>Identify the main entities, using the results of Step 1.</td>
</tr>
<tr>
<td>3</td>
<td>Define the relationships among the entities, using the results of Steps 1 and 2.</td>
</tr>
<tr>
<td>4</td>
<td>Define the attributes, primary keys, and foreign keys for each of the entities.</td>
</tr>
<tr>
<td>5</td>
<td>Normalize the entities. (Remember that entities are implemented as tables in an RDBMS.)</td>
</tr>
<tr>
<td>6</td>
<td>Complete the initial ER diagram.</td>
</tr>
<tr>
<td>7</td>
<td>Have the main end users verify the model in Step 6 against the data, information, and processing requirements.</td>
</tr>
<tr>
<td>8</td>
<td>Modify the ER diagram, using the results of Step 7.</td>
</tr>
</tbody>
</table>

Note that some of the steps listed in Table 9.1 take place concurrently. And some, such as the normalization process, can generate a demand for additional entities and/or attributes, thereby causing the designer to revise the ER model. For example, while identifying two main entities, the designer might also identify the composite bridge entity that represents the many-to-many relationship between those two main entities.
12. List and briefly explain the activities involved in the verification of an ER model.

Section 9.3.2, “Database Design,” includes a discussion on verification. In addition, Appendix C, “The University Lab: Conceptual Design Verification, Logical Design, and Implementation,” covers the verification process in detail. The verification process is detailed in the text’s Table 9.3, reproduced here for your convenience.

**TABLE 9.3 The ER Model Verification Process**

<table>
<thead>
<tr>
<th>STEP</th>
<th>ACTIVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Identify the ER model’s central entity.</td>
</tr>
<tr>
<td>2</td>
<td>Identify each module and its components.</td>
</tr>
</tbody>
</table>
| 3    | Identify each module’s transaction requirements:  
|      | Internal: Updates/Inserts/Deletes/Queries/Reports  
|      | External: Module interfaces |
| 4    | Verify all processes against the ER model. |
| 5    | Make all necessary changes suggested in Step 4. |
| 6    | Repeat Steps 2–5 for all modules. |

Keep in mind that the verification process requires the continuous verification of business transactions as well as system and user requirements. The verification sequence must be repeated for each of the system’s modules.

13. What factors are important in a DBMS software selection?

The selection of DBMS software is critical to the information system’s smooth operation. Consequently, the advantages and disadvantages of the proposed DBMS software should be carefully studied. To avoid false expectations, the end user must be made aware of the limitations of both the DBMS and the database.

Although the factors affecting the purchasing decision vary from company to company, some of the most common are:

- **Cost.** Purchase, maintenance, operational, license, installation, training, and conversion costs.
- **DBMS features and tools.** Some database software includes a variety of tools that facilitate the application development task. For example, the availability of query by example (QBE), screen painters, report generators, application generators, data dictionaries, and so on, helps to create a more pleasant work environment for both the end user and the application programmer. Database administrator facilities, query facilities, ease of use, performance, security, concurrency control, transaction processing, and third-party support also influence DBMS software selection.
- **Underlying model.** Hierarchical, network, relational, object/relational, or object.
- **Portability.** Across platforms, systems, and languages.
- **DBMS hardware requirements.** Processor(s), RAM, disk space, and so on.

Problem Solutions
1. The ABC Car Service & Repair Centers are owned by the SILENT car dealer; ABC services and repairs only SILENT cars. Three ABC Car Service & Repair Centers provide service and repair for the entire state.

Each of the three centers is independently managed and operated by a shop manager, a receptionist, and at least eight mechanics. Each center maintains a fully stocked parts inventory. Each center also maintains a manual file system in which each car’s maintenance history is kept: repairs made, parts used, costs, service dates, owner, and so on. Files are also kept to track inventory, purchasing, billing, employees’ hours, and payroll.

You have been contacted by the manager of one of the centers to design and implement a computerized system. Given the preceding information, do the following:

a. Indicate the most appropriate sequence of activities by labeling each of the following steps in the correct order. (For example, if you think that “Load the database.” is the appropriate first step, label it “1.”)

   ____ Normalize the conceptual model.
   ____ Obtain a general description of company operations.
   ____ Load the database.
   ____ Create a description of each system process.
   ____ Test the system.
   ____ Draw a data flow diagram and system flowcharts.
   ____ Create a conceptual model, using E-R diagrams.
   ____ Create the application programs.
   ____ Interview the mechanics.
   ____ Create the file (table) structures.
   ____ Interview the shop manager.

The answer to this question may vary slightly from one designer to the next, depending on the selected design methodology and even on personal designer preferences. Yet, in spite of such differences, it is possible to develop a common design methodology to permit the development of a basic decision-making process and the analysis required in designing an information system.

Whatever the design philosophy, a good designer uses a specific and ordered set of steps through which the database design problem is approached. The steps are generally based on three phases: analysis, design, and implementation. These phases yield the following activities:

**ANALYSIS**
1. Interview the shop manager
2. Interview the mechanics
3. Obtain a general description of company operations
4. Create a description of each system process

**DESIGN**
5. Create a conceptual model, using E-R diagrams
6. Draw a data flow diagram and system flow charts
7. Normalize the conceptual model

IMPLEMENTATION
8. Create the table structures
9. Load the database
10. Create the application programs
11. Test the system.

This listing implies that, within each of the three phases, the steps are completed in a specific order. For example, it would seem reasonable to argue that we must first complete the interviews if we are to obtain a proper description of the company operations. Similarly, we may argue that a data flow diagram precedes the creation of the E-R diagram. Nevertheless, the specific tasks and the order in which they are addressed may vary. Such variations do not matter, as long as the designer bases the selected procedures on an appropriate design philosophy, such as top-down vs. bottom-up.

Given this discussion, we may present problem 1’s solution this way:

7. Normalize the conceptual model.
3. Obtain a general description of company operations.
9. Load the database.
4. Create a description of each system process.
11. Test the system.
6. Draw a data flow diagram and system flow charts.
5. Create a conceptual model, using E-R diagrams.
10. Create the application programs.
2. Interview the mechanics.
8. Create the file (table) structures.
1. Interview the shop manager.
b. Describe the different modules that you believe the system should include.

This question may be addressed in several ways. We suggest the following approach to develop a system composed of four main modules: Inventory, Payroll, Work order, and Customer.

We have illustrated the Information System's main modules in Figure P9.1B.

Figure P9.1B The ABC Company’s IS System Modules

The Inventory module will include the Parts and Purchasing sub-modules. The Payroll Module will handle all employee and payroll information. The Work order module keeps track of the car maintenance history and all work orders for maintenance done on a car. The Customer module keeps track of the billing of the work orders to the customers and of the payments received from those customers.

c. How will a data dictionary help you develop the system? Give examples.

We have addressed the role of the data dictionary within the DBLC in detail in the answer to review question 10. Remember that the data dictionary makes it easier to check for the existence of synonyms and homonyms, to check whether all attributes exist to support required reports, to verify appropriate relationship representations, and so on. Therefore, the data dictionary's contents will help us to provide consistency across modules and to evaluate the system's ability to generate the required reports. In addition, the use of the data dictionary facilitates the creation of system documentation.
d. What general (system) recommendations might you make to the shop manager? (For example, if the system will be integrated, what modules will be integrated? What benefits would be derived from such an integrated system? Include several general recommendations.)

The designer's job is to provide solutions to the main problems found during the initial study. Clearly, any system is subject to both internal and external constraints. For example, we can safely assume that the owner of the ABC Car Service and Repair Center has a timeframe in mind, not to mention a spending limitation. As is true in all design work, the designer and the business owner must prioritize the modules and develop those that yield the greatest benefit within the stated time and development budget constraints.

Keep in mind that it is always useful to develop a modular system that provides for future enhancement and expansion. Suppose, for example, that the ABC Car Service & Repair company management decides to integrate all of its service stations in the state in order to provide better statewide service. Such integration is likely to yield many benefits: The car history of each car will be available to any station for cars that have been serviced in more than one location; the inventory of parts will be online, thus allowing parts orders to be placed between service stations; mechanics can better share tips concerning the solution to car maintenance problems, and so on.

e. What is the best approach to conceptual database design? Why?

Given the nature of this business, the best way to produce this conceptual database design would be to use a centralized and top-down approach. Keep in mind that the designer must keep the design sufficiently flexible to make sure that it can accommodate any future integration of this system with the other service stations in the state.

f. Name and describe at least four reports the system should have. Explain their use. Who will use those reports?

REPORT 1
Monthly Activity contains a summary of service categories by branch and by month. Such reports may become the basis for forecasting personnel and stock requirements for each branch and for each period.

REPORT 2
Mechanic Summary Sheet contains a summary of work hours clocked by each mechanic. This report would be generated weekly and would be useful for payroll and maintenance personnel scheduling purposes.

REPORT 3
Monthly Inventory contains a summary of parts in inventory, inventory draw-down, parts reorder points, and information about the vendors who will provide the parts to be reordered. This report will be especially useful for inventory management purposes.

REPORT 4
Customer Activity contains a breakdown of customers by location, maintenance activity, current balances, available credit, and so on. This report would be useful to forecast various service demand
factors, to mail promotional materials, to send maintenance reminders, to keep track of special customer requirements, and so on.

2. **Suppose you have been asked to create an information system for a manufacturing plant that produces nuts and bolts of many shapes, sizes, and functions. What questions would you ask, and how would the answers to those questions affect the database design?**

Basically, all answers to all (relevant) questions help shape the database design. In fact, all information collected during the initial study and all subsequent phases will have an impact on the database design. Keep in mind that the information is collected to establish the entities, attributes, and the relationships among the entities. Specifically, the relationships, connectivities, and cardinalities are shaped by the business rules that are derived from the information collected by the designer.

Sample questions and their likely impact on the design might be:

- **Do you want to develop the database for all departments at once, or do you want to design and implement the database for one department at a time?**

- **How will the design approach affect the design process?** (In other words, assess top-down vs. bottom-up, centralized or decentralized, system scope and boundaries.)

- **Do you want to develop one module at a time, or do you want an integrated system?** (Inventory, production, shipping, billing, etc.)

- **Do you want to keep track of the nuts and bolts by lot number, production shift, type, and department?** Impact: conceptual and logical database design.

- **Do you want to keep track of the suppliers of each batch of raw material used in the production of the nuts and bolts?** Impact: conceptual and logical database design. E-R model.

- **Do you want to keep track of the customers who received the batches of nuts and bolts?** Impact: conceptual and logical database design. ER model.

- **What reports will you require, what will be the specific reporting requirements, and to whom will these reports be distributed?**

The answers to such questions affect the conceptual and logical database design, the database’s implementation, its testing, and its subsequent operation.

a. **What do you envision the SDLC to be?**

The SDLC is not a function of the information collected. Regardless of the extent of the design or its specific implementation, the SDLC phases remain:

**PLANNING**

- Initial assessment
- Feasibility study

**ANALYSIS**

- User requirements
- Study of existing systems
- Logical system design
b. **What do you envision the DBLC to be?**

As is true for the SDLC, the DBLC is not a function of the kind and extent of the collected information. Thus, the DBLC phases and their activities remain as shown:

**DATABASE INITIAL STUDY**
- Analyze the company situation
- Define problems and constraints
- Define objectives
- Define scope and boundaries

**DATABASE DESIGN**
- Create the conceptual design
- Create the logical design
- Create the physical design

**IMPLEMENTATION AND LOADING**
- Install the DBMS
- Create the database(s)
- Load or convert the data

**TESTING AND EVALUATION**
- Test the database
- Fine-tune the database
- Evaluate the database and its application programs

**OPERATION**
- Produce the required information flow

**MAINTENANCE AND EVOLUTION**
- Introduce changes
- Make enhancements

3. **Suppose you perform the same functions noted in Problem 2 for a larger warehousing operation. How are the two sets of procedures similar? How and why are they different?**
The development of an information system will differ in the approach and philosophy used. More precisely, the designer team will probably be formed by a group of system analysts and may decide to use a decentralized approach to database design.

Also, as is true for any organization, the system scope and constraints may be very different for different systems. Therefore, designers may opt to use different techniques at different stages. For example, the database initial study phase may include separate studies carried out by separate design teams at several geographically distant locations. Each of the findings of the design teams will later be integrated to identify the main problems, solutions, and opportunities that will guide the design and development of the system.