Chapter 1

QUESTIONS

1. Describe a database that you have used on a job or as a consumer. List the entities and relationships that the database contains. If you are not sure, imagine the entities and relationships that are contained in the database.

Ans: Students' responses will vary. However, in the example of the University Database, the entities are students, faculty, courses, offerings, and enrollments. The relationships are faculty teaches offerings, students enroll in offerings, offerings made of courses and faculty supervises faculties.

2. For the database in question 1, list different user groups that can use the database.

Ans: Students' responses will vary. However, some examples of user groups for the University database are students, faculty and university's staff.

3. For one of the groups in question 2, describe an application (form or report) that the group uses.

Ans: Students' responses will vary. However, some examples of applications for the University database are that the user group of student can use the registration form, transcript (the report of grade). The user group of university's staff can use the assignment form. The user group of faculty can use the report of faculty workload.

4. Explain the persistent property for databases.

Ans: Persistent means that data resides on stable storage such as a magnetic disk. For example, organizations need to retain data about customers, suppliers, and inventory on stable storage because these data are repetitively used. A variable in a computer program is not persistent because it resides in main memory and disappears after the program terminates. Persistency does not mean that data lasts forever. When data are no longer relevant (such as a supplier going out of business), they are removed or archived.

5. Explain the inter-related property for databases.

Ans: Inter-related means that data stored as separate units can be connected to provide a whole picture. For example, a customer database relates customer data (name, address, …) to order data (order number, order date, …) to facilitate order processing. Databases contain both entities and relationships among entities. An entity is a cluster of data usually about a single topic that can be accessed together. An entity may denote a person, place, thing, or event.

6. Explain the shared property for databases.

Ans: Shared means that a database can have multiple uses and users. A database provides a common memory for multiple functions in an organization. For example, a personnel database can support payroll calculations, performance evaluations, government reporting requirements, and so on. Many users can use a database at the same time. For example, many customers can simultaneously make airline reservations. Unless two users are trying to change the same part of the database at the same time, they can proceed without waiting.

7. What is a DBMS?

Ans: A database management system (DBMS) is a collection of software that supports the creation, use, and maintenance of databases. Initially, DBMSs provided efficient storage and retrieval of data. Due to marketplace demands and product innovation, DBMSs have evolved to provide a broad range of features for data acquisition, storage, dissemination, maintenance, retrieval, and formatting. The evolution of these features has made DBMSs rather complex.

8. What is SQL?

Ans: The Structured Query Language (SQL) is an industry standard language supported by most DBMSs. SQL contains statements for data definition, data manipulation, and data control.

9. Describe the difference between a procedural and a non-procedural language. What statements belong in a procedural language but not in a nonprocedural language?

Ans: A nonprocedural language allows users with limited computing skills to submit queries instead of coding complex procedures. Nonprocedural languages specify what parts of a database to retrieve, not the detail of how retrieval occurs as compared to procedural language.

Statements that belong in a procedural language but not in a nonprocedural language are looping statements (for, while, and so on).

10. Why is non-procedural access an important feature of DBMS?

Ans: Non-procedural access can reduce the number of lines of code by a factor of 100 as compared to procedural access. Because a large part of business software involves data access, non-procedural access can provide a dramatic improvement in software productivity.

11. What is the connection between nonprocedural access and application (form or report) development? Can nonprocedural access be used in application development?

Ans: The connection between nonprocedural access and application (form or report) development is that non-procedural access is used in application development to indicate data requirements. Non-procedural access makes form and report creation possible without extensive coding. As part of creating a form or report, the user indicates the data requirements using a non-procedural language (SQL) or graphical tool.

12. What is difference between a form and a report?

Ans: Data entry forms provide a convenient way to enter and edit data, while reports enhance the appearance of data that is displayed or printed.

13. What is a procedural language interface?

Ans: A procedural language interface adds the full capabilities of a computer programming language. Non-procedural access and application development tools, though convenient and powerful, are sometimes not efficient enough or do not provide the level of control necessary for application development. When these tools are not adequate, DBMSs provide the full capabilities of a programming language. A procedural language interface combines a non-procedural language such as SQL with a programming language such as COBOL or Visual Basic.

14. What is a transaction?

Ans: A transaction is a unit of work that should be processed reliably without interference from other users and without loss of data due to failures. Examples of transactions are withdrawing cash at an ATM, making an airline reservation, and registering for a course.

15. What features does a DBMS provide to support transaction processing?

Ans: A DBMS ensures that transactions are free of interference from other users, parts of a transaction are not lost due to a failure, and transactions do not make the database inconsistent. Transaction processing is largely a "behind the scenes" affair. The user does not know the details about transaction processing other than the assurances about reliability.

16. For the database in question 1, describe a transaction that uses the database. How often do you think the transaction is submitted to the database? How many users submit transactions at the same time? Make guesses for the last two parts if you are unsure.

Ans: The example of transaction that uses the University Database is registering for a course. The University Database ensures that the registering for a course is free of interference from other students (or users). Parts of course registration are not lost due to a failure, and the registering for a course does not make the university database inconsistent. The transaction is often submitted to the database. Hundreds of students can submit transactions at the same time.

17. What is an enterprise DBMS?

Ans: An enterprise DBMS supports databases that are often critical to the functioning of an organization. Enterprise DBMSs usually run on powerful servers and have a high cost.

18. What is a desktop DBMS?

Ans: A desktop DBMS runs on personal computers and small servers. It supports limited transaction processing features but has a much lower cost than an enterprise DBMS. Desktop DBMSs support databases used by work teams and small businesses.

19. What is an embedded DBMS?

Ans: Embedded DBMSs are an emerging category of database software. As the name implies, an embedded DBMS resides in a larger system, either an application or a device such as a Personal Digital Assistant or smart card. Embedded DBMSs provide limited transaction processing features but have low memory, processing, and storage requirements.

20. What were the prominent features of first generation DBMSs?

Ans: File structures and proprietary program interfaces were the prominent features of first generation database software.

21. What were the prominent features of second generation DBMSs?

Ans: Networks and hierarchies of related records along with standard program interfaces were the prominent features of second generation database software.

22. What were the prominent features of third generation DBMSs?

Ans: Non-procedural languages, optimization, and transaction processing were the prominent features of third generation database software.

23. What are the prominent features of fourth generation DBMSs?

Ans: Support for multi-media data, active databases, data warehouses, and distributed processing are the prominent features of fourth generation database software.

24. For the database you described in question 1, make a table to depict differences among schema levels. Use Table 1–4 as a guide.

Ans:

Schema level	Description
External	The registration form View, the report of grade View, the faculty assignment form View, and the report of faculty workload View
Conceptual	Student, Enrollment, Course, Faculty, and Enrollment tables and relationships
Internal	Files needed to store the tables; extra files to improve performance

25. What is the purpose of the mappings in the Three Schema Architecture? Is the user or the DBMS responsible for using the mappings?

Ans: The purpose of the mappings in the Three Schema Architecture is to describe how a schema at a higher level is derived from a schema at a lower level. The DBMS, not the user, is responsible for using the mappings.

26. Explain how the Three Schema Architecture supports data independence?

Ans: The Three Schema Architecture is a standard that serves as a guideline about how data independence can be achieved. The spirit of the Three Schema Architecture is widely implemented in third- and fourth-generation DBMS. In the Three Schema Architecture, the DBMS uses schemas and mappings to ensure data independence. Typically, applications access a database using a view. The DBMS converts an application's request into a request using the conceptual schema rather than the view. The DBMS then transforms the conceptual schema request into a request using the internal schema. Most changes to the conceptual or internal schema do not affect applications because applications do not use the lower schema levels.

27. In a client–server architecture, why are processing capabilities divided between a client and a server? In other words, why not have the server do all the processing?

Ans: To improve performance and availability of data, distributed processing allows geographically dispersed computers to cooperate when providing data access. Work can be balanced between a server and a client to efficiently process data access requests.

28. In a client–server architecture, why are data sometimes stored on several computers rather than on a single computer?

Ans: Because data can be stored in different locations for management and security, data are sometimes stored on several computers rather than on a single computer.

29. For the database in question 1, describe how functional users may interact with the database. Try to identify indirect, parametric, and power uses of the database.

Ans: In the university database, a student can be an indirect user when student requests and receives the transcript. A student can be a parametric user when student registers for a course. An academic advisor can be a power user when she/he queries reports of certain students that meet specific criteria.

30. Explain the differences in responsibilities between an active functional user of a database and an analyst. What schema level is used by both kinds of users?

Ans: A functional user may participate in designing and using databases, while an analyst is responsible for collecting requirements, designing applications, and implementing information system. External level is used by both kinds of users.

31. Which role, database administrator or data administrator, is more appealing to you as a long-term career goal? Briefly explain your preference.

Ans: Students' responses will vary. However, an example response might be: Data administrator is more appealing to me as a long-term career goal because the Data administrator has broader responsibilities than the database administrator. Data administrator is primarily a planning role, while database administrator is a more technical role focused on individual databases and DBMS. Data administrators also view the information resource in a broader context and consider all kinds of data, both computerized and noncomputerized.

32. What market niche is occupied by open source DBMS products.?

Ans: Many organizations have reported cost savings using open source DBMS products, mostly for non mission critical systems.

Chapter 2

QUESTIONS

1. What is the relationship between a system and an information system?

Ans: A system is a set of related components that work together to accomplish some objectives. An information system is a system; however, an information system manipulates data rather than a physical object.

2. Provide an example of a system that is not an information system.

Ans: Students' answers may vary. An example might be: An ecology system.

3. For an information system of which you are aware, describe some of the components (input data, output data, people, software, hardware, and procedures).

Ans: Students' responses will vary. However, the question is looking for the following type of response: Input data is used in the processes of the system. Output data is produced by the system. People interact with the system. Software implements and supports the processes. Hardware is the equipment used in the system. Procedures are steps in the process. An example system is a hospital billing system. Important inputs to the system are patient admittances, patient resources consumed during a hospital stay, third party (insurance and government) policies, patient checkouts, and payments. Important outputs are bills to consumers and third party payers.

4. Briefly describe some of the kinds of data in the database for the information system in question 3.

Ans: Students' responses will vary. However, the following is an example: Input data is used in the database to produce output. For example, the database contains data about patients, hospital resources and services, policies of third party providers, bills prepared, payments received, and hospital stays.

5. Describe the phases of the waterfall model.

Ans: The phases of the waterfall model are the traditional systems development life cycle. The result of each phase flows to the next phase. The following items describe the activities in each phase:

- Preliminary Investigation phase: Produces a problem statement and feasibility study.
- Systems analysis phase: Produces requirements describing processes, data, and environment interactions.
- Systems design phase: Produces a plan to efficiently Implement the requirements.
- Systems Implementation phase: Produces executable code, databases, and user documentation.
- Maintenance phase: Produces corrections, changes, and enhancements to an operating information system.

6. Why is the waterfall model considered only a reference framework?

Ans: Because the particular phases are not standard. For most systems, the boundary between phases is blurred and there is considerable backtracking between phases.

7. What are the shortcomings in the waterfall model?

Ans: The traditional life cycle has been criticized for several reasons. First, an operating system is not produced until late in the process. By the time a system is operational, the requirements already have changed. Second, there is often a rush to begin implementation so that a product is visible. In this rush, appropriate time may not be devoted to analysis and design.

8. What alternative methodologies have been proposed to alleviate the difficulties of the waterfall model?

Ans: A number of alternative methodologies have been proposed to alleviate these difficulties. In spiral development methodologies, the life-cycle phases are performed for subsets of a system, progressively producing a larger system until the complete system emerges. Rapid application development methodologies delay producing design documents until requirements are clear. Scaled-down versions of a system known as prototypes are used to clarify requirements.

9. What is the relationship of the database development process to the information systems development process?

Ans: The database development process is an important part of the information systems development process. The database development process produces the data model that specifies the operational database for the information system. The data model is usually the most important artifact for an information system.

10. What is a data model? Process model? Environment interaction model?

Ans: The data model describes the kinds of data and relationships. The process model describes relationships among processes. The environment interaction model describes relationships between events and processes. In many information systems development efforts, the data model is the most important. For business information systems, the process and environment interaction models are usually produced after the data model.

11. What is the purpose of prototyping in the information systems development process?

Ans: To clarify requirements by using the scaled-down versions of a system. Prototyping can reduce the risk of developing an information system because it allows earlier and more direct feedback about the system.

12. How is a database designer like a politician in establishing a common vocabulary?

Ans: A good politician often finds solutions with which everyone finds something to agree or disagree. A good database designer also finds similar imperfect solutions.

13. Why should a database designer establish the meaning of data?

Ans: Because a database contains business rules to support organizational policies. Defining business rules is the essence of defining the semantics or meaning of the database.

14. What factors should a database designer consider when choosing database constraints?

Ans: Selecting appropriate constraint levels may require compromise to balance the needs of different groups. Constraints that are too strict may force work-around solutions to handle exceptions. In contrast, constraints that are too loose may allow incorrect data in a database.

15. Why is data quality important?

Ans: The importance of data quality is analogous to the importance of product quality in manufacturing. Because data are the product of an information system, data quality is also important. Poor data quality can lead to poor decision making about communicating with customers, identifying repeat customers, tracking sales, and resolving customer problems.

16. Provide examples of data quality problems according to two characteristics mentioned in Section 2.2.3.

Ans: Students' responses may vary. However, an example might be: The data quality problem on correctness could be that the employee mistypes the address of a customer. The timeliness problem could be that the store has announced sales in an advertisement, but the store database has not updated the price yet.

17. How does a database designer decide on the appropriate level of data quality?

Ans: Achieving adequate data quality may require a cost-benefit trade-off. For example, in a grocery store database, the benefits of timely price updates are reduced consumer complaints and less loss in fines from government agencies. Achieving data quality can be costly both in preventative and monitoring activities. For example, to improve the timeliness and accuracy of price updates, automated data entry may be used (preventative activity) as well as sampling the accuracy of the prices charged to consumers (monitoring activity). The database designer should consider both short-term and long-term effects of data quality.

18. Why is it important to find an efficient implementation?

Ans: Because even if the other design goals are met, a slow-performing database will not be used.

19. What are the inputs and the outputs of the conceptual data modeling phase?

Ans: The inputs of the conceptual data modeling phase are data requirements. The outputs of the conceptual data modeling phase are entity relationship diagrams.

20. What are the inputs and the outputs of the logical database design phase?

Ans: The inputs of the logical database design phase are entity relationship diagrams. The outputs of the logical database design phase are relational database tables.

21. What are the inputs and the outputs of the distributed database design phase?

Ans: The inputs of the distributed database design phase are relational database tables. The outputs of the distributed database design phase are the distribution schema indicating the location of data and processes.

22. What are the inputs and the outputs of the physical database design phase?

Ans: The inputs of the physical database design phase are the distribution schema for each site. The outputs of the physical database design phase are the internal schema and the populated database. The internal schema indicates index choices, data placement, and other decisions.

23. What does it mean to say that the conceptual data modeling phase and the logical database design phase are concerned with the information content of the database?

Ans: The conceptual data modeling phase and the logical database design phase emphasize the meaning of the database, not an efficient representation. The information content involves the entities and relationships in the database, not efficiency concerns. The information content also involves the representation of the information content on a commercial DBMS.

24. Why are there two phases (conceptual data modeling and logical database design) that involve the information content of the database?

Ans: Because of the database development process, the first two phases are concerned with the information content of the database while the last two phases are concerned with efficient implementation. The conceptual data modeling phase uses data requirements and produces entity relationship diagrams for the conceptual schema and for each external schema. The logical database design phase is concerned with refinements to the conceptual data model. The refinements preserve the information content of the conceptual data model while enabling representation on a commercial DBMS.

25. What is the relationship of view design and view integration to conceptual data modeling?

Ans: View design and integration is an approach to managing the complexity of large database development efforts. For large databases, the conceptual modeling phase is usually modified. Designing large databases is a time-consuming and labor-intensive process often involving a team of designers. In view design, an ERD is constructed for each group of users. A view is typically small enough for a single person to design. The view integration process merges the views into a complete, conceptual schema.

26. What is a soft skill?

Ans: A soft skill is a qualitative, subjective, and people-oriented skill used for conceptual data modeling and logical database design phases.

27. What phases of database development primarily involve soft skills?

Ans: The conceptual data modeling and logical database design phases involve mostly soft skills. As a database designer, you want to generate a range of feasible alternatives. The choice among feasible alternatives can be subjective. You should note the assumptions in which each feasible alternative is preferred. The alternative chosen is often subjective based on the designer's assessment of the most reasonable assumptions. Conceptual data modeling is especially people-oriented. In the role of data modeling, you need to obtain requirements from diverse groups of users.

28. What is a hard skill?

Ans: A hard skill is a quantitative, objective, and data intensive skill used for distributed database design and physical database design.

29. What phases of database development primarily involve hard skills?

Ans: Distributed database design and physical database design involve mostly hard skills. Background in quantitative disciplines such as statistics and operations management can be useful to understand mathematical models used in these phases. Many decisions cannot be based on objective criteria alone because of uncertainty about database usage. To resolve uncertainty, intensive data analysis can be useful. The database designer should collect and analyze data to understand patterns of database usage and database performance.

30. What kind of background is appropriate for hard skills?

Ans: Background in statistics and operations management.

31. Why do large organizations sometimes have different people performing design phases dealing with information content and efficient implementation?

Ans: Because of the diverse skills and background knowledge required in different phases of database development, the same people will not perform both roles in large organizations.

32. Why are CASE tools useful in the database development process?

Ans: CASE tools can help improve the productivity of information systems professionals working on large projects as well as end users working on small projects. A number of studies suggest that CASE tools facilitate improvements in the early phases of systems development leading to lower cost, higher quality, and faster implementations.

33. What is the difference between front-end and back-end CASE tools?

Ans: CASE tools often are classified as front-end and back-end tools. Front-end CASE tools can help designers diagram, analyze, and document models used in the database development process. Back-end CASE tools create prototypes and generate code that can be used to cross check a database with other components of an information system.

34. What kinds of support can a CASE tool provide for drawing a database diagram?

Ans: Most CASE tools provide predefined shapes and connections among the shapes. The connection tools typically allow the shapes to be moved but stay connected as though "glued." This glue feature provides important flexibility because symbols on a diagram typically are rearranged many times. For large drawings, CASE tools provide several features. Most CASE tools allow diagrams to span multiple pages. Multiple-page drawings can be printed so that the pages can be pasted together to make a wall display. Layout can be difficult for large drawings. Some CASE tools try to improve the visual appeal of a diagram by performing automatic layout.

35. What kinds of support can a CASE tool provide for documenting a database design?

Ans: Storing various properties of a data model and linking the properties to symbols on the diagram, storing text describing assumptions, alternatives, and notes, and documenting versions.

36. What kinds of support can a CASE tool provide for analyzing a database design?

Ans: Active assistance to database designers through analysis functions. Analysis functions can be provided in each phase of database development. In the conceptual data modeling phase, analysis functions can reveal conflicts in an ERD. In the logical database design phase, conversion and normalization are common analysis functions. Conversion produces a table design from an ERD. Normalization removes redundancy in a table design. In the distributed database design and physical database design phases, analysis functions can suggest decisions about data location and index selection. In addition, analysis functions for version control can cross database development phases. Analysis functions can convert between versions and show a list of differences between versions.

37. What kinds of support can a CASE tool provide for prototyping?

Ans: Prototyping tools provide a link between database development and application development. Prototyping tools can be used to create forms and reports that use a database. Because prototyping tools may generate code (SQL statements and programming language code), they are sometimes known as code generation tools. Prototyping tools are often provided as part of a DBMS. The prototyping tools may feature wizards to aid a developer in quickly creating applications that can be tested by users.

38. Should you expect to find one software vendor providing a full range of functions (drawing, documenting, analyzing, and prototyping) for the database development process? Why or why not?

Ans: Students' responses may vary. There are a number of CASE tools that provide extensive functionality for database development. Each of the products in Table 2-2 is a complex product that supports the full life-cycle of information systems development. Although the features of the products look similar, the quality, the depth, and the breadth of the features may vary across products. In addition, most of the products in Table 2-2 have several different versions that vary in price and features. Most major products are relatively neutral to a particular DBMS even if a product is offered by a DBMS vendor.