Homework Chapter1

Review Questions

1.1. Define the following terms: data, database, DBMS, database system, database catalog, program-data independence, user view, DBA, end user, canned transaction, deductive database system, persistent object, meta-data, and transaction-processing application.

Database: a collection of related data or operational data extracted from any organization.

DBMS: a collection of programs that enables users to create, maintain, and manipulate a database. The DBMS is a general purpose software system that facilitates the process of defining, constructing, and manipulating database.

Database catalog: contains complete description of the databases, database object, database structure, details of users, and constraints, and etc. that are sorted.

Program-data independence: Storing the data and program separately.

User view: database appears to a particular user.

DBA: database administrator who is responsible for authorizing access to the database, coordinating and monitoring its use, and acquiring software and hardware resources as needed.

End user: the people who want to access the database for different purpose like, querying, updating and generating reports.

Canned Transactions: Standardized queries and updates on the database using carefully programmed and tested programs.

Deductive Database System: a database system support the proof-theoretic view of a database.

Persistent object: Object-Oriented database systems are compatible with programming languages. An object that is stored in such a way that it survives that termination of the DBMS is persistent.

Meta Data: Information about the data. The information stored in the catalog is called Metadata. The schema of a table is an example of metadata.

Transaction processing application: A transaction is a logical unit of database. The processing includes one or more database operations like insertion, deletion, modification, and retrieval. SQL

1.2. What four main types of actions involve databases? Briefly discuss each.

1. Database Administration
2. Database Designing
3. Database Users
4. System Analysis and Application Programming

1.3. Discuss the main characteristics of the database approach and how it differs from traditional file systems.

Self-describing nature of a database system;

Insulation between programs and data and data abstraction;

Support multiple views of the data;

Sharing of data and multi-user transaction process

1.4. What are the responsibilities of the DBA and the database designers?

DBA: management, build the physical design

Designer: architect, work with everyone

1.5. What are the different types of database end users? Discuss the main activities of each.

Casual; Naïve or parametric; Sophisticated; Standalone

1.6. Discuss the capabilities that should be provided by a DBMS.

Controlling redundancy;

Restricting unauthorized access;

Providing persistent storage for program object;

Providing storage structures for efficient query processing;

Providing Backup and Recovery;

Proving Multiple User Interfaces;

Representing complex relationships among data;

Enforcing integrity constraints

1.7. Discuss the differences between database systems and information retrieval systems.

1.8. Identify some informal queries and update operations that you would expect to apply to the database shown in Figure 1.2.

1.9. What is the difference between controlled and uncontrolled redundancy?

Illustrate with examples.

1.10. Specify all the relationships among the records of the database shown in

Figure 1.2.

1.11. Give some additional views that may be needed by other user groups for the database shown in Figure 1.2.

1.12. Cite some examples of integrity constraints that you think can apply to the database shown in Figure 1.2.

1.13. Give examples of systems in which it may make sense to use traditional file processing instead of a database approach.

1.14. Consider Figure 1.2.

a. If the name of the ‘CS’ (Computer Science) Department changes to ‘CSSE’

(Computer Science and Software Engineering) Department and the corresponding prefix for the course number also changes, identify the columns in the database that would need to be updated.

b. Can you restructure the columns in the COURSE, SECTION, and

PREREQUISITE tables so that only one column will need to be updated?

Homework Chapter 2

Review Questions

2.1. Define the following terms: data model, database schema, database state, internal schema, conceptual schema, external schema, data independence, DDL, DML, SDL, VDL, query language, host language, data sublanguage, database utility, catalog, client/server architecture, three-tier architecture, and n-tier architecture.

Data model: a collection of concepts that can be used to described the structure of a database. Provides for the database abstractions. Data models is collections conceptual tools for describing data, data relationships, and data semantics and consistency constraints.

Database Schema: The overall design.

Database state: The data at a particular time

Internal Schema: Description of the physical stage structure

Conceptual Schema: Description of the structure of the database for a community of users.

External Schema: a particular group can see and how it sees it.

Data independence: The capacity to change the schema at one level of a database system without having to change the schema at the next higher level.

DDL: Data definition language. DBA and developers use it to create a database with specific schemas.

DML: Data manipulation language.

SDL: Storage definition language. It is used to spend the internal schema.

VDL: View definition language.

2.2. Discuss the main categories of data models. What are the basic differences among the relational model, the object model, and the XML model?

2.3. What is the difference between a database schema and a database state?

2.4. Describe the three-schema architecture. Why do we need mappings among schema levels? How do different schema definition languages support this architecture?

2.5. What is the difference between logical data independence and physical data independence? Which one is harder to achieve? Why?

2.6. What is the difference between procedural and nonprocedural DMLs?

2.7. Discuss the different types of user-friendly interfaces and the types of users who typically use each.

2.8. With what other computer system software does a DBMS interact?

2.9. What is the difference between the two-tier and three-tier client/server architectures?

2.10. Discuss some types of database utilities and tools and their functions.

2.11. What is the additional functionality incorporated in n-tier architecture (n .3)?

Exercises

2.12. Think of different users for the database shown in Figure 1.2. What types of applications would each user need? To which user category would each belong, and what type of interface would each need?

2.13. Choose a database application with which you are familiar. Design a schema and show a sample database for that application, using the notation of Figures 1.2 and 2.1. What types of additional information and constraints would you like to represent in the schema? Think of several users of your database, and design a view for each.

2.14. If you were designing a Web-based system to make airline reservations and sell airline tickets, which DBMS architecture would you choose from Section 2.5?

Why? Why would the other architectures not be a good choice?

2.15. Consider Figure 2.1. In addition to constraints relating the values of columns in one table to columns in another table, there are also constraints that impose restrictions on values in a column or a combination of columns within a table. One such constraint dictates that a column or a group of columns must be unique across all rows in the table. For example, in the STUDENT table, the Student number column must be unique (to prevent two different students from having the same Student number). Identify the column or the group of columns in the other tables that must be unique across all rows in the table.

Homework Chapter 3

Review Questions

3.1. Discuss the role of a high-level data model in the database design process.

3.2. List the various cases where use of a NULL value would be appropriate.

3.3. Define the following terms: entity, attribute, attribute value, relationship instance, composite attribute, multivalued attribute, derived attribute, complex attribute, key attribute, and value set (domain).

3.4. What is an entity type? What is an entity set? Explain the differences among an entity, an entity type, and an entity set.

**3.5. Explain the difference between an attribute and a value set.**

3.6. What is a relationship type? Explain the differences among a relationship instance, a relationship type, and a relationship set.

3.7. What is a participation role? When is it necessary to use role names in the description of relationship types?

3.8. Describe the two alternatives for specifying structural constraints on relationship types. What are the advantages and disadvantages of each?

3.9. Under what conditions can an attribute of a binary relationship type be migrated to become an attribute of one of the participating entity types? 3.10. When we think of relationships as attributes, what are the value sets of these attributes? What class of data models is based on this concept?

**3.11. What is meant by a recursive relationship type? Give some examples of recursive relationship types.**

**3.12. When is the concept of a weak entity used in data modeling? Define the terms owner entity type, weak entity type, identifying relationship type, and partial key.**

3.13. Can an identifying relationship of a weak entity type be of a degree greater than two? Give examples to illustrate your answer.

3.14. Discuss the conventions for displaying an ER schema as an ER diagram.

3.15. Discuss the naming conventions used for ER schema diagrams.

Exercises

3.16. Which combinations of attributes have to be unique for each individual SECTION entity in the UNIVERSITY database shown in Figure 3.20 to enforce each of the following miniworld constraints:

a. During a particular semester and year, only one section can use a particular classroom at a particular DaysTime value.

b. During a particular semester and year, an instructor can teach only one section at a particular DaysTime value.

c. During a particular semester and year, the section numbers for sections offered for the same course must all be different.

Can you think of any other similar constraints?

**3.17.** Composite and multivalued attributes can be nested to any number of levels.

Suppose we want to design an attribute for a STUDENT entity type to keep track of previous college education. Such an attribute will have one entry for each college previously attended, and each such entry will be composed of college name, start and end dates, degree entries (degrees awarded at that college, if any), and transcript entries (courses completed at that college, if any). Each degree entry contains the degree name and the month and year the degree was awarded, and each transcript entry contains a course name, semester, year, and grade. Design an attribute to hold this information.

Use the conventions in Figure 3.5.

3.18. Show an alternative design for the attribute described in Exercise 3.17 that uses only entity types (including weak entity types, if needed) and relationship types.

3.19. Consider the ER diagram in Figure 3.21, which shows a simplified schema for an airline reservations system. Extract from the ER diagram the requirements and constraints that produced this schema. Try to be as precise as possible in your requirements and constraints specification.

3.20. In Chapters 1 and 2, we discussed the database environment and database users. We can consider many entity types to describe such an environment, such as DBMS, stored database, DBA, and catalog/data dictionary. Try to specify all the entity types that can fully describe a database system and its environment; then specify the relationship types among them, and draw an ER diagram to describe such a general database environment.

**3.21.** Design an ER schema for keeping track of information about votes taken in the U.S. House of Representatives during the current two-year congressional session. The database needs to keep track of each U.S. STATE’s Name (e.g., ‘Texas’, ‘New York’, ‘California’) and include the Region of the state (whose domain is {‘Northeast’, ‘Midwest’, ‘Southeast’, ‘Southwest’, ‘West’}). Each CONGRESS\_PERSON in the House of Representatives is described by his or her Name, plus the District represented, the Start\_date when the congressperson was first elected, and the political Party to which he or she belongs (whose domain is {‘Republican’, ‘Democrat’, ‘Independent’,‘Other’}). The database keeps track of each BILL (i.e., proposed law), including the Bill\_name, the Date\_of\_vote on the bill, whether the bill Passed\_or\_failed (whose domain is {‘Yes’, ‘No’}), and the Sponsor (the congressperson(s) who sponsored—that is, proposed—the bill). The database also keeps track of how each congressperson voted on each bill (domain of Vote attribute is {‘Yes’, ‘No’, ‘Abstain’, ‘Absent’}). Draw an ER schema diagram for this application. State clearly any assumptions you make.

3.22. A database is being constructed to keep track of the teams and games of a sports league. A team has a number of players, not all of whom participate in each game. It is desired to keep track of the players participating in each game for each team, the positions they played in that game, and the result of

3.23. Consider the ER diagram shown in Figure 3.22 for part of a BANK database.

Each bank can have multiple branches, and each branch can have multiple accounts and loans.

a. List the strong (nonweak) entity types in the ER diagram.

b. Is there a weak entity type? If so, give its name, partial key, and identifying relationship.

c. What constraints do the partial key and the identifying relationship of the weak entity type specify in this diagram?

d. List the names of all relationship types, and specify the (min, max) constraint on each participation of an entity type in a relationship type. Justify your choices.

3.24. Consider the ER diagram in Figure 3.23. Assume that an employee may work in up to two departments or may not be assigned to any department.

Assume that each department must have one and may have up to three phone numbers. Supply (min, max) constraints on this diagram. State clearly any additional assumptions you make. Under what conditions would the relationship HAS\_PHONE be redundant in this example?

3.25. Consider the ER diagram in Figure 3.24. Assume that a course may or may not use a textbook, but that a text by definition is a book that is used in some course. A course may not use more than five books. Instructors teach from two to four courses. Supply (min, max) constraints on this diagram. State clearly any additional assumptions you make. If we add the relationship

ADOPTS, to indicate the textbook(s) that an instructor uses for a course, should it be a binary relationship between INSTRUCTOR and TEXT, or a ternary relationship among all three entity types? What (min, max) constraints would you put on the relationship? Why?

**3.26.** Consider an entity type SECTION in a UNIVERSITY database, which describes the section offerings of courses. The attributes of SECTION are Section\_number, Semester, Year, Course\_number, Instructor, Room\_no (where section is taught), Building (where section is taught), Weekdays (domain is the possible combinations of weekdays in which a section can be offered {‘MWF’, ‘MW’, ‘TT’, and so on}), and Hours (domain is all possible time periods during which sections are offered {‘9–9:50 a.m.’, ‘10–10:50 a.m.’, . . . , ‘3:30–4:50 p.m.’, ‘5:30–6:20 p.m.’, and so on}). Assume that Section\_number is unique for each course within a particular semester/ year combination (that is, if a course is offered multiple times during a particular semester, its section offerings are numbered 1, 2, 3, and so on). There are several composite keys for section, and some attributes are components of more than one key. Identify three composite keys, and show how they can be represented in an ER schema diagram.

**3.27.** Cardinality ratios often dictate the detailed design of a database. The cardinality ratio depends on the real-world meaning of the entity types involved and is defined by the specific application. For the following binary relationships, suggest cardinality ratios based on the common-sense meaning of the entity types. Clearly state any assumptions you make.

Entity 1 Cardinality Ratio Entity 2

1. STUDENT \_\_\_\_\_\_\_\_\_\_\_\_\_\_ SOCIAL\_SECURITY\_CARD

2. STUDENT \_\_\_\_\_\_\_\_\_\_\_\_\_\_ TEACHER

3. CLASSROOM \_\_\_\_\_\_\_\_\_\_\_\_\_\_ WALL

4. COUNTRY \_\_\_\_\_\_\_\_\_\_\_\_\_\_ CURRENT\_PRESIDENT

5. COURSE \_\_\_\_\_\_\_\_\_\_\_\_\_\_ TEXTBOOK

6. ITEM (that can be found in an order) \_\_\_\_\_\_\_\_\_\_\_\_\_\_ ORDER

7. STUDENT \_\_\_\_\_\_\_\_\_\_\_\_\_\_ CLASS

8. CLASS \_\_\_\_\_\_\_\_\_\_\_\_\_\_ INSTRUCTOR

9. INSTRUCTOR \_\_\_\_\_\_\_\_\_\_\_\_\_\_ OFFICE

10. EBAY\_AUCTION\_ITEM \_\_\_\_\_\_\_\_\_\_\_\_\_\_ EBAY\_BID