AMU - INFO 441 - Data Communications and Networks

Part 3 – Answers for Review Questions & Exercises

SOLUTIONS TO REVIEW QUESTIONS

AND EXERCISES

PART 3 - DATABASE ANALYSIS AND DESIGN
(CHAPTERS 10 – 15)

Solutions to Review Questions and Exercises

Chapter 10 Database System Developmemt Lifecycle

3

Chapter 11 Database Analysis and the *DreamHome* Case Study

6

Chapter 12 Entity-Relationship Modeling

10

Chapter 13 Enhanced Entity-Relationship Modeling

16

Chapter 14 Normalization

19

Chapter 15 Advanced Normalization

28

Chapter 10 Database System Development Lifecycle

Review Questions

10.1 *Describe the major components of an information system.*

Database, database software, application software, computer hardware including storage media, and personnel using and developing the system.

10.2 *Discuss the relationship between the information systems lifecycle and the database system development lifecycle.*

See Sections 9.1 and 9.2.

10.3 *Describe the main purpose(s) and activities associated with each stage of the database system development lifecycle.*

See Table 9.1 in Section 9.2.

9.4 *Discuss what a user view represents in the context of a database system.*

User view: defines what is required of a database system from the perspective of a particular job role (such as Manager or Supervisor) or enterprise application area (such as marketing, personnel, or stock control) (See Section 9.4.1).

10.5 *Discuss the main approaches for managing the design of a database system that has multiple user views.*

Three main approaches: centralized, view integration, and combination of both approaches (see Section 9.5).

10.6 *Compare and contrast the three phases of database design.*

Three phases are conceptual database design, logical database design, and physical database design.

Conceptual database design constructs a model of the information used in an enterprise, independent of all physical considerations. Logical database design is based on a specific data model, but independent of all other physical considerations. Physical database design constructs a description of the implementation of the database on secondary storage (see Section 9.6.3).

10.7 *What are the main purposes of data modeling and identify the criteria for an optimal data model.*

The two main purposes of data modeling are to assist in the understanding of the meaning (semantics) of the data and to facilitate communication about the information requirements (see Section 9.6.2).

10.8 *Identify the stage(s) where it is appropriate to select a DBMS and describe an approach to selecting the 'best' DBMS.*

Physical database design is tailored to a specific DBMS; therefore it is essential the DBMS is determined before the physical design phase can begin. Physical database design is described in Section 9.6.3.

9.9 *Application design involves transaction design and user interface design. Describe the purpose and main activities associated with each.*

See Section 9.8.

10.10 *Discuss why testing cannot show the absence of faults, only that software faults are present.*

See Section 9.12.

10.11 *Describe the main advantages of using the prototyping approach when building a database system.*

Should be relatively inexpensive to develop and quick to build (see Section 9.9).

.

Exercises

10.12 *Assume that you are responsible for selecting a new DBMS product for a group of users in your organization. To undertake this exercise, you must first establish a set of requirements for the group and then identify a set of features that a DBMS product must provide to fulfil the requirements. Describe the process of evaluating and selecting the best DBMS product.*

The student should follow the approach to DBMS selection described in Section 9.7 and produce a report that identifies a suitable DBMS product that meets the requirements of the organization. The selection should be fully justified and any assumptions made should be highlighted.

10.13 *Describe the process of evaluating and selecting a DBMS product for each of the case studies described in Appendix B.*

The student should follow the approach to DBMS selection described in Section 9.7 and produce a report that identifies a suitable DBMS product that meets the requirements of each organization described in Appendix B. The selection should be fully justified and any assumptions made about the case study should be highlighted.

10.14 Assume that you are an employee of a consultancy company that specializes in the analysis, design, and implementation of database systems. A client has recently approached your company with a view to implementing a database system but they are not familiar with the development process. You have been assigned the task to present an overview of the Database System Development Lifecycle (DSDL) to them,identifying the main stages of this lifecycle. With this task in mind, create a slide presentation and/or short report for the client. (The client for this exercise can be any one of the fictitious case studies given in Appendix or some real company identified by you or your professor).

The student should use the information presented in overview form in Section 10.2 and in more detail in sections 10.3 through 10.13 to create the presentation.

10.15 This exercise requires you to first gain permission to interview one or more people responsible for the development and/or administration of a real database system. During the interview(s), find out the following information:
(a) The approach taken to develop the database system.
(b) How the approach taken differs or is similar to the DSDL approach described in this chapter.
(c) How the requirements for different users (user views) of the database systems were managed.
(d) Whether a CASE tool was used to support the development of the database system.
(e) How the DBMS product was evaluated and then selected.
(f) How the database system is monitored and maintained.

The results of this student project will depend on the people being interviewed. Students should develop a list of questions that should lead to answers to items a-f above before beginning the interview.

Chapter 11 Database Analysis and the *DreamHome* Case Study

Review Questions

11.1 *Briefly describe what the process of fact-finding attempts to achieve for a database developer.*

Attempts to uncover facts about the business and the users of the database system including the vocabulary, problems, opportunities, constraints, requirements, and priorities.

11.2 *Describe how fact-finding is used throughout the stages of the database system development lifecycle.*

See Table 11.1 in Section 11.2.

11.3 *For each stage of the database system development lifecycle identify examples of the facts captured and the documentation produced.*

See Table 11.1 in Section 11.2.

11.4 *A database developer normally uses several fact-finding techniques during a single database project. The five most commonly used techniques are examining documentation, interviewing, observing the business in operation, conducting research, and using questionnaires. Describe each fact-finding technique and identify the advantages and disadvantages of each.*

See Section 11.3.

11.5 *Describe the purpose of defining a mission statement and mission objectives for a database system.*

Mission statement defines the major aims of the database system; the mission objectives identify the particular tasks that the database must support (see Section 11.4.2).

11.6 *What is the purpose of identifying the systems boundary for a database system?*

Ensures that all appropriate areas of the organization are supported by the database system (see Section 11.4.3).

11.7 *How does the contents of a users' requirements specification differ from a systems specification?*

Systems specification describes the any features to be included in the new database system such as networking and shared access requirements, performance requirements, and the levels of security required. On the other hand, the users' requirements specification describes in detail the data to be held in the database and how the data is to be used (see Section 10.4.4).

11.8 *Describe one method to deciding whether to use either the centralized or view integration approach, or a combination of both when developing a database system for multiple user views.*

One way is to examine the overlap in the data used between the various user views (see, for example, Table 11.7).

Exercises

11.9 *Assume that you are an employee of a consultancy company that specializes in the analysis, design, and implementation of database systems. A client has recently approached your company with a view to implementing a database system, but they are not familiar with the development process.*

Task: You are required to present an overview of the fact-finding techniques that your company intends to use to support the development of the client's database system. With this task in mind, create a slide presentation and/or report that describes each fact-finding technique and how the fact-finding techniques will be used throughout the development of the database system. The client for this exercise and those that follow can be any one of the fictitious case studies given in Appendix B or some real company identified by you or your professor.

In preparing a slide presention or report about fact-finding techniques, students should cover the five commonly used fact-finding techniques:
Examining documentation Section 11.3.1, Table 11.2);
Enterviewing (Section 11.3.2)
Observing the enterprise in operation (Section 11.3.3)
Research (Section 11.3.4)
Questionnaires (Section 11.3.5)

11.10 *Assume that you are an employee of a consultancy company that specializes in the analysis, design, and implementation of database systems. A client has recently approached your company with a view to implementing a database system.*

Task: You are required to establish the database project through the early stages of the project. With this task in mind, create a mission statement and mission objectives and high-level systems diagram for the client's database system.

Students will find it helpful to review the definition of a mission statement, mission objectives, and high-level systems diagram in Sections 11.4.2 and 11.4.3.

11.11 *Assume that you are an employee of a consultancy company that specializes in the analysis, design, and implementation of database systems. A client has recently approached your company with a view to implementing a database system. It has already been established that the client's database system will support many different groups of users (user views).*

Task: You are required to identify how to best manage the requirements for these user views. With this task in mind, create a report that identifies high-level requirements for each user view and shows the relationship between the user views. Conclude the report by identifying and justifying the best approach to managing the multi-user view requirements.

Students should create documentation on the required views including a table documenting view relationships such as the one shown in Figure 11.7.

Review Questions

12.1 *Describe what entity types represent in an ER model and provide examples of entities with a physical or conceptual existence.*

An entity type represents a group of 'objects' in the real world with the same properties (see Section 12.1). Examples of entities with physical and conceptual existence are shown in Figure 12.2.

12.2 *Describe what relationship types represent in an ER model and provide examples of unary, binary, ternary, and quaternary relationships.*

A relationship type is a set of associations between one or more participating entity types (see Section 12.2). Examples:

Unary:	Staff *Supervise* Staff (also called recursive relationship)
Binary:	Branch *Has* Staff
Ternary:	Staff *Registers* Client at Branch
Quaternary:	Solicitor *Arranges* Bid with a Buyer supported by a Financial Institution.

12.3 *Describe what attributes represent in an ER model and provide examples of simple, composite, single-value, multi-value, and derived attributes.*

An attribute represents a property of an entity or a relationship type (see Section 12.3). Examples:

Simple:	position or salary attribute of Staff
Composite:	address attribute composed of street, city, and postcode attributes
Single-valued:	branchNo attribute of Branch
Multi-valued:	telNo attribute of Branch
Derived:	duration attribute of Lease, calculated from rentStart and rentFinish attributes.

12.4 *Describe what the multiplicity constraint represents for a relationship type.*

Multiplicity represents the number (or range) of possible occurrences of an entity type that may relate to a single occurrence of an associated entity type through a particular relationship (see Section 12.6).

12.5 *What are enterprise constraints and how does multiplicity model these constraints?*

Enterprise constraints are rules that the data in the database must conform to as specified by users or database administrators of a database (see Section 4.3.4). Multiplicity constrains the

way that entities are related – it is a representation of the policies (or business rules) established by the user or enterprise.

12.6 *How does multiplicity represent both the cardinality and the participation constraints on a relationship type?*

Multiplicity actually consists of two separate constraints (see Section 12.6.5):

Cardinality – which describes the maximum number of possible relationship occurrences for an entity participating in a given relationship type.
Participation – which determines whether all or only some entity occurrences participate in a relationship.

12.7 *Provide an example of a relationship type with attributes.*

The relationship Newspaper *Advertises* PropertyForRent consists of two attributes: dateAdvert (representing the date the advert took place) and cost (representing the cost of the advert).

12.8 *Describe how strong and weak entity types differ and provide an example of each.*

A strong entity type is an entity type that is not existence-dependent on some other entity type (see Section 12.4). Examples of strong entity types are Branch, Staff, and PropertyForRent.

A weak entity type is an entity type that is existence-dependent on some other entity type. An example of a weak entity type is Preference.

12.9 *Describe how fan and chasm traps can occur in an ER model and how they can be resolved.*

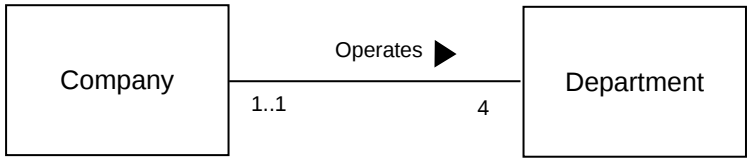
A fan trap occurs where a model represents a relationship between two entity types, but the pathway between certain entity occurrences is ambiguous. Resolve the fan trap by restructuring the original ER diagram to represent the correct association between these entities (see Section 12.7.1).

A chasm trap occurs where a model suggests the existence of a relationship between entity types, but the pathway does not exist between certain entity occurrences. A chasm trap may occur where there are one or more relationships with optional participation. Resolve the chasm trap by identifying the missing relationship (see Section 12.7.2).

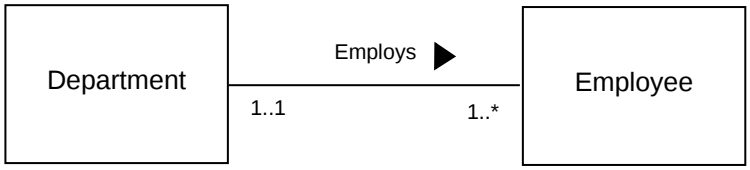
Exercises

12.10 *Create an ER diagram for each of the following descriptions:*

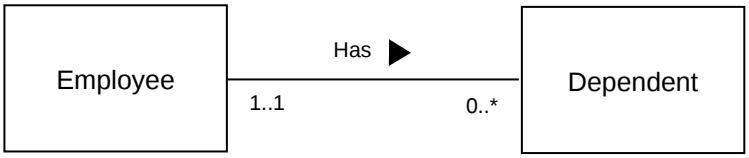
(a) *Each company operates four departments, and each department belongs to one company.*



(b) *Each department in part (a) employs one or more employees, and each employee works for one department.*

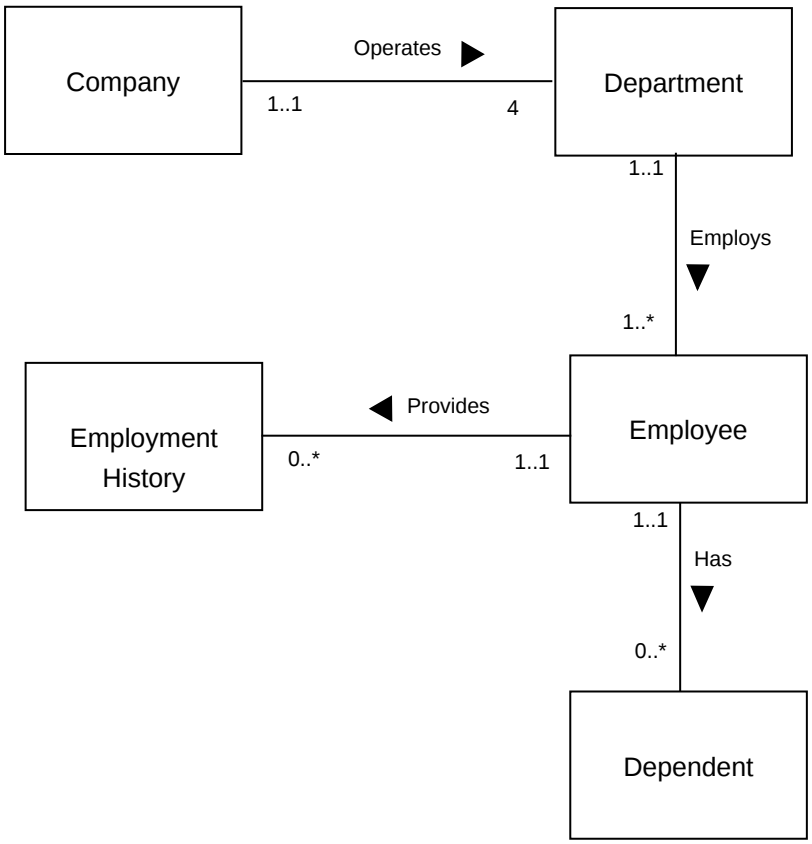


(c) *Each of the employees in part (b) may or may not have one or more dependants, and each dependant belongs to one employee.*



(d) *Each employee in part (c) may or may not have an employment history.*

(e) *Represent all the ER diagrams described in (a), (b), (c), and (d) as a single ER diagram.*

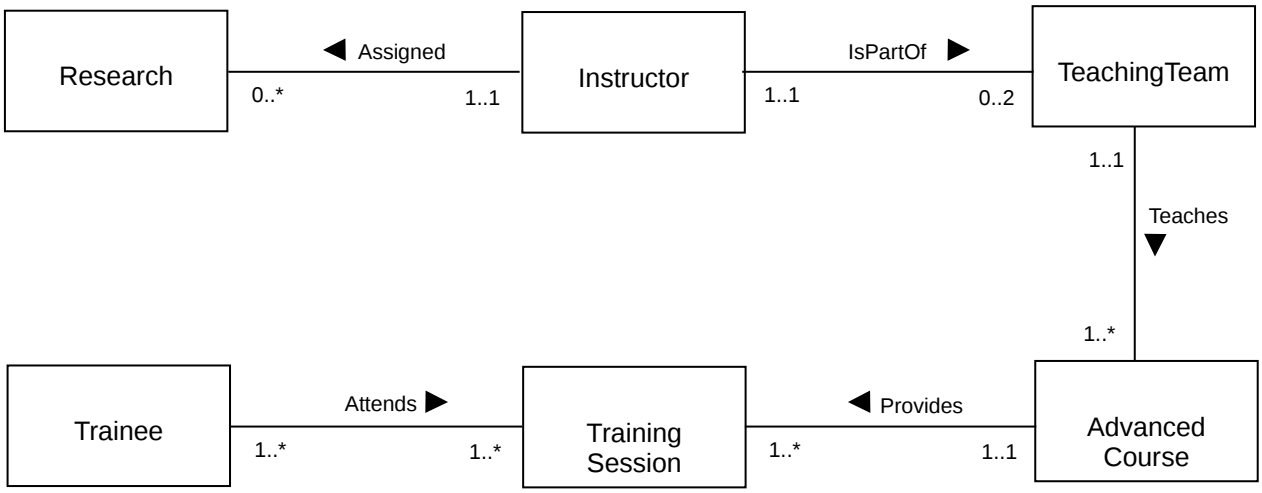


12.11 *You are required to create a conceptual data model of the data requirements for a company that specializes in IT training. The Company has 30 instructors and can handle up to 100 trainees per training session. The Company offers five advanced technology courses, each of which is taught by a teaching team of two or more instructors. Each instructor is assigned to a maximum of two teaching teams or may be assigned to do research. Each trainee undertakes one advanced technology course per training session.*

(a) Identify the main entity types for the company.

(b) Identify the main relationship types and specify the multiplicity for each relationship. State any assumptions you make about the data.

(c) Using your answers for (a) and (b), draw a single ER diagram to represent the data requirements for the company.



12.12 *Read the following case study, which describes the data requirements for a video rental company. The video rental company has several branches throughout the USA. The data held on each branch is the branch address made up of street, city, state, and zip code, and the telephone number. Each branch is given a branch number, which is unique throughout the company. Each branch is allocated staff, which includes a Manager. The Manager is responsible for the day-to-day running of a given branch. The data held on a member of staff is his or her name, position, and salary. Each member of staff is given a staff number, which is unique throughout the company. Each branch has a stock of videos. The data held on a video is the catalog number, video number, title, category, daily rental, cost, status, and the names of the main actors, and the director. The catalog number uniquely identifies each video. However, in most cases, there are several copies of each video at a branch, and the individual copies are identified using the video number. A video is given a category such as Action, Adult, Children, Drama, Horror, or Sci-Fi. The status indicates whether a specific copy of a video is available for rent. Before hiring a video from the company, a customer must first register as a member of a local branch. The data held on a member is the first and last name, address, and the date that the member registered at a branch. Each member is given a*

member number, which is unique throughout all branches of the company. Once registered, a member is free to rent videos, up to maximum of ten at any one time. The data held on each video rented is the rental number, the full name and number of the member, the video number, title, and daily rental, and the dates the video is rented out and date returned. The rental number is unique throughout the company.

(a) *Identify the main entity types of the video rental company.*

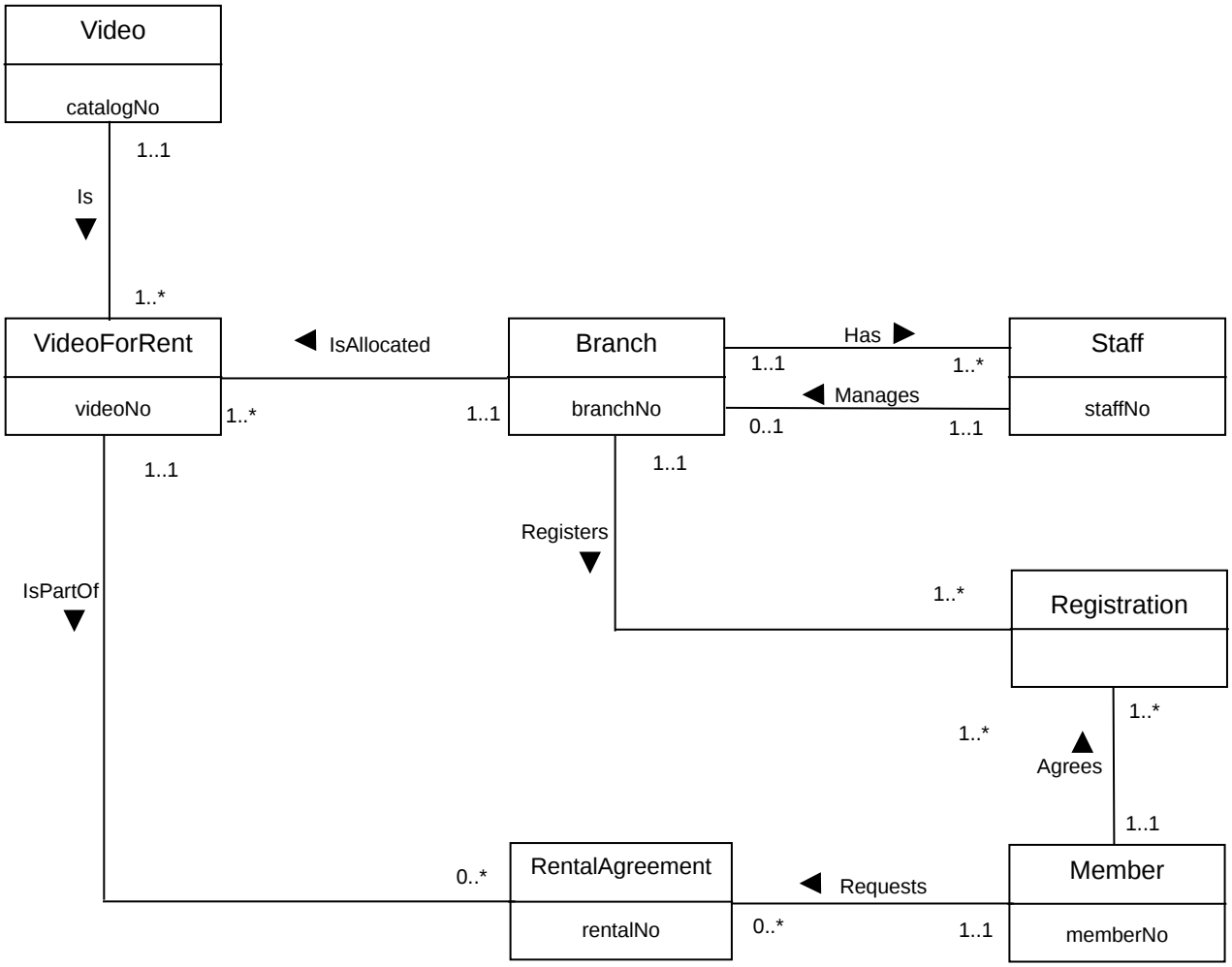
(b) *Identify the main relationship types between the entity types described in (a) and represent each relationship as an ER diagram.*

(c) *Determine the multiplicity constraints for each relationship described in (b). Represent the multiplicity for each relationship in the ER diagrams created in (b).*

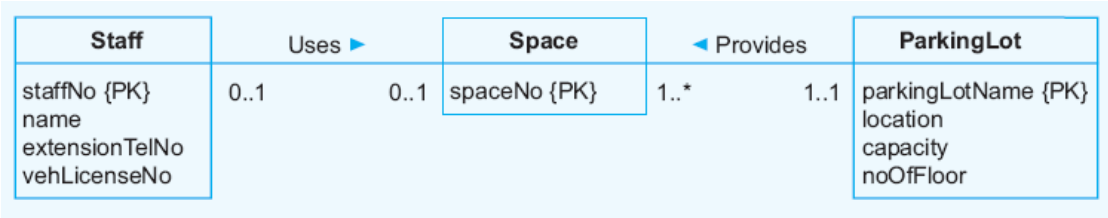
(d) *Identify attributes and associate them with entity or relationship types. Represent each attribute in the ER diagrams created in (c).*

(e) *Determine candidate and primary key attributes for each (strong) entity type.*

(f) *Using your answers (a) to (e) attempt to represent the data requirements of the video rental company as a single ER diagram. State any assumptions necessary to support your design.*



12.13 *Create an ER model for each of the following descriptions:*
(a) A large organization has several parking lots, which are used by staff.
(b) Each parking lot has a unique name, location, capacity, and number of floors (where appropriate).
(c) Each parking lot has parking spaces, which are uniquely identified using a space number.
(d) Members of staff can request the use of a parking space. Each member of staff has a unique number, name,
telephone extension number, and vehicle license number.
(e) Represent all the ER models described in parts (a), (b), (c), and (d) as a single ER model. Provide any assumptions necessary to support your model.



Chapter 13 Enhanced Entity-Relationship Modeling

Review Questions

13.1 *Describe what a superclass and a subclass represent.*

A **superclass** is an entity type that includes one or more distinct subgroupings of its occurrences, which require to be represented in a data model. A **subclass** is a distinct subgrouping of occurrences of an entity type, which require to be represented in a data model (see Section 13.1.1).

13.2 *Describe the relationship between a superclass and its subclass.*

The relationship between superclass and subclass is 1:1, called a superclass/subclass relationship (see Section 13.1.2). Each member of a subclass is also a member of the superclass.

13.3 *Describe and illustrate using an example the process of attribute inheritance.*

An entity in a subclass represents the same 'real world' object as in the superclass, and may possess subclass-specific attributes, as well as those associated with the superclass. For example, a member of the SalesPersonnel subclass *inherits* all the attributes of the Staff superclass such as staffNo, name, position, and salary together with those specifically associated with the SalesPersonnel subclass such as salesArea and carAllowance (see Section 13.1.3).

13.4 *What are the main reasons for introducing the concepts of superclasses and subclasses into an ER model?*

There are two important reasons for introducing the concepts of superclasses and subclasses into an ER model. Firstly, it avoids describing similar concepts more than once, thereby saving time for the designer and making the ER diagram more readable. Secondly, it adds more semantic information to the design in a form that is familiar to many people. For example, the assertions that 'Manager IS-A member of staff' and 'flat IS-A type of property', communicates significant semantic content in a concise form (see Section 13.1.2).

13.5 *Describe what a shared subclass represents and how does this concept relate to multiple inheritance.*

A subclass with more than one superclass is called a **shared subclass**. The subclass will inherit the attributes of all its superclasses (i.e. multiple inheritance).

13.6 *Describe and contrast the process of specialization with the process of generalization.*

Specialization is the process of maximizing the differences between members of an entity by identifying their distinguishing features. **Generalization** is the process of minimizing the differences between entities by identifying their common features. Specialization is a top-down approach whereas generalization is a bottom-up approach (see Sections 13.1.4 and 13.1.5).

13.7 *Describe the two main constraints that apply to a specialization/generalization relationship.*

Two main constraints are: participation and disjoint constraints (see Section 13.1.6).

13.8 *Describe and contrast the concepts of aggregation and composition and provide an example of each.*

Aggregation represents a 'has-a' or 'is-part-of' relationship between entity types, where one represents the 'whole' and the other 'the part'. **Composition** is a specific form of aggregation that represents an association between entities, where there is a strong ownership and coincidental lifetime between the 'whole' and the 'part' (see Sections 13.2 and 13.3).

Exercises

13.9 *Consider whether it is appropriate to introduce the enhanced concepts of specialization/generalization, aggregation, and/or composition for the case studies described in Appendix B.*

There are aspects of each case study that could be represented using specialization / generalization, aggregation, and/or composition. Some examples are given below:

The *University Accommodation Office* Case Study
Accommodation as a superclass with subclasses StudentFlat and HallofResidence.
Staff as a superclass with subclasses HallManager, AdministrativeAssistant, and Cleaner.
Aggregation relationship between StudentFlat and FlatInspection
Composition relationship between StudentFlat and FlatRoom
Composition relationship between HallofResidence and HallRoom

The *EasyDrive School of Motoring* Case Study
Test as a superclass with TheoryTest and PracticalTest as subclasses.
Staff as a superclass with subclasses SeniorInstructor, Instructor, and Administrator.
Composition relationship between Car and DrivingSchool
Composition relationship between Staff and DrivingSchool

The *Wellmeadows Hospital* Case Study
Staff as a superclass with subclasses MedicalDirector, PersonnelOfficer, ChargeNurse, SeniorNurse, JuniorNurse, Doctor, and Auxiliary.
Patient as a superclass with subclasses OutPatient and InPatient.
Supply as a superclass with subclasses Surgical, Non-surgical, and Pharmaceutical.

Aggregation relationship between Patient and PatientMedication.

13.10 *Consider whether it is appropriate to introduce the enhanced concepts of specialization/generalization, aggregation, and/or composition into the ER model for the case study described in Exercise 11.12. If appropriate, redraw the ER diagram as an EER diagram with the additional enhanced concepts.*

Could consider Manager as a specialization of the Staff entity. This would move the *Manages* relationship from Staff to the Manager subclass. However, the attributes for both entities would be the same and there would, therefore, seem to be no obvious advantage to introducing the Manager specialization.

13.11 *Introduce specialization/generalization concepts into the ER model shown in Figure 13.11 and described in Exercise 12.13 to show the following:*
(a) The majority of parking spaces are under cover and each can be allocated for use by a member of staff for a monthly rate.
(b) Parking spaces that are not under cover are free to use when available.
(c) Up to twenty covered parking spaces are available for use by visitors to the company. However, only members of staff are able to book out a space for the day of the visit. There is no charge for this type of booking, but the member of staff must provide the visitor's vehicle license number.

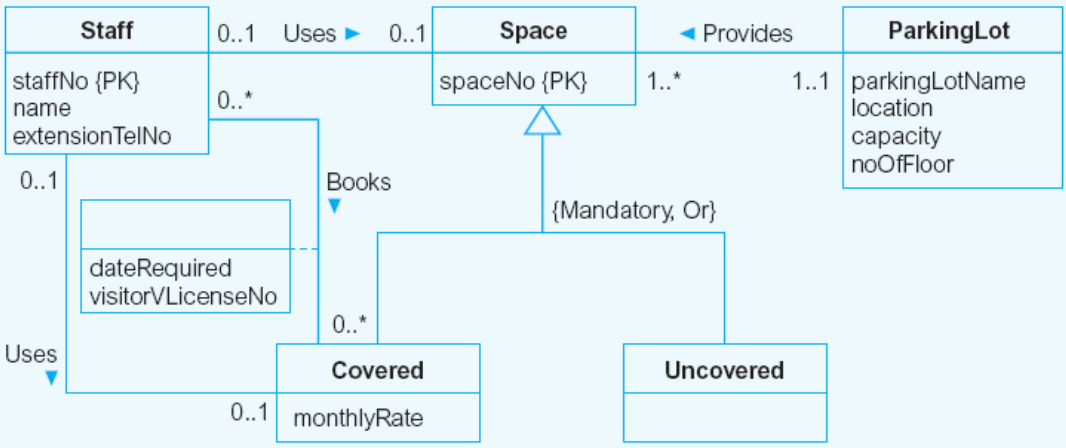


Figure 17.11 An EER model of the Parking Lot case study.

Chapter 14 Normalization

Review Questions

14.1 *Describe the purpose of normalizing data.*

The purpose of normalization is to identify a suitable set of relations that support the data requirements of an enterprise. The characteristics of a suitable set of relations include the following:

- the *minimal* number of attributes necessary to support the data requirements of the enterprise;
- attributes with a close logical relationship (described as functional dependency) are found in the same relation;
- *minimal* redundancy with each attribute represented only once with the important exception of attributes that form all or part of foreign keys (see Section 4.2.5), which are essential for the joining of related relations.

The benefit of using a database that has a suitable set of relations is that the database will be easier for the user to access and maintain the data, and take up minimal storage space on the computer.

14.2 *Discuss the alternative ways that normalization can be used to support database design.*

Normalization is a formal technique that can be used at any stage of database design. However in this section we highlight two main approaches for using normalization as illustrated in Figure 14.1. Approach (1) shows how normalization can be used as a bottom-up standalone database design technique while Approach (2) shows how normalization can be used as a validation technique to check the structure of relations, which may have been created using a top-down approach such as ER modeling. No matter which approach is used the goal is the same that of creating a set of well-designed relations that meet the data requirements of the enterprise. Figure 14.1 shows examples of data sources that can be used for database design. Although, the users' requirements specification (see Section 10.5) is the preferred data source, it is possible to design a database based on the information taken directly from other data sources such as forms and reports as illustrated in this chapter and the next. Figure 14.1 also shows that the same data source can be used for both approaches, however, although this is true in principle, in practice the approach taken is likely to be determined by the size, extent, and complexity of the database being described by the data sources and by the preference and expertise of the database designer. The opportunity to use normalization as a bottom-up standalone technique (Approach 1) is often limited by the level of detail that the database

designer is reasonably expected to manage. However, this limitation is not applicable when normalization is used as a validation technique (Approach 2) as the database designer only focuses on part of the database such as a single relation at any one time. Therefore, no matter what the size or complexity of the database then normalization can be usefully applied.

See Figure 14.1

14.3 *Describe the types of update anomalies that may occur on a relation that has redundant data.*

A major aim of relational database design is to group attributes into relations so as to minimize information redundancy and thereby reduce the file storage space required by the base relations. Another serious difficulty using relations that have redundant information is the problem of update anomalies. These can be classified as insertion, deletion, or modification anomalies.

See Section 14.3

14.4 *Describe the concept of functional dependency.*

Functional dependency describes the relationship between attributes in a relation. For example, if A and B are attributes of relation R, B is functionally dependent on A (denoted A $\rightarrow$ B), if each value of A in R is associated with exactly one value of B in R.

Functional dependency is a property of the meaning or semantics of the attributes in a relation. The semantics indicate how the attributes relate to one another and specify the functional dependencies between attributes. When a functional dependency is present, the dependency is specified as a constraint between the attributes.

See also Section 14.3.

14.5 *What are the main characteristics of functional dependencies that are used when normalizing a relation?*

In summary, the functional dependencies that we use in normalization have the following characteristics:

- There is a *one-to-one* relationship between the attribute(s) on the left-hand side (determinant) and those on the right-hand side of a functional dependency. (Note that the relationship in the opposite direction; that is between the right to the left-hand side attributes can be a one-to-one relationship or one-to-many relationship).
- Holds for *all* time.
- The determinant has the *minimal* number of attributes necessary to maintain the

dependency with the attribute(s) on the right hand-side. In other words, there must be a full functional dependency between the attribute(s) on the left and right hand-side of the dependency.

14.6 *Describe how a database designer typically identifies the set of functional dependencies associated with a relation.*

Identifying all functional dependencies between a set of attributes should be relatively simple if the meaning of each attribute and the relationships between the attributes are well understood. This type of information may be provided by the enterprise in the form of discussions with users and/or appropriate documentation such as the users' requirements specification. However, if the users are unavailable for consultation and/or the documentation is incomplete then depending on the database application it may be necessary for the database designer to use their common sense and/or experience to provide the missing information.
As a contrast to this, consider the situation where functional dependencies are to be identified in the absence of appropriate information about the meaning of attributes and their relationships. In this case, it may be possible to identify functional dependencies if sample data is available that is a true representation of *all* possible data values that the database may hold.

14.7 *Describe the characteristics of a table in Unnormalized Form (UNF) and describe how such a table is converted to a First Normal Form (1NF) relation.*

A table in unnormalized form contains one or more repeating groups. To convert to first normal form (1NF) either remove the repeating group to a new relation along with a copy of the original key attribute(s), or remove the repeating group by entering appropriate data in the empty columns of rows containing the repeating data (see Section 14.5).

14.8 *What is the minimal normal form that a relation must satisfy? Provide a definition for this normal form.*

Minimal normal form is 1NF: a relation in which the intersection of each row and column contains one and only one value (see Section 14.5).

14.9 *Describe the two approaches to converting an Unnormalized Normal Form (UNF) table to First Normal Form (1NF) relation(s).*

There are two common approaches to removing repeating groups from unnormalized tables:

(a) *By entering appropriate data in the empty columns of rows containing the repeating data.* In other words, we fill in the blanks by duplicating the nonrepeating data, where required. This approach is commonly referred to as 'flattening' the table.

(b) *By placing the repeating data, along with a copy of the original key attribute(s), in a*

separate relation. Sometimes the unnormalized table may contain more than one repeating group, or repeating groups within repeating groups. In such cases, this approach is applied repeatedly until no repeating groups remain. A set of relations is in 1NF if they contain no repeating groups.

For both approaches, the resulting tables are now referred to as 1NF relations containing atomic (or single) values at the intersection of each row and column. Although both approaches are correct, while approach (a) introduces more redundancy into the original UNF table as part of the 'flattening' process, approach (b) creates two or more relations with less redundancy than in the original UNF table. In other words, approach (b) moves the original UNF table further along the normalization process than approach (a). However, no matter which initial approach is taken the original UNF table will be normalized into the same set of 3NF relations.

14.10 *Describe the concept of full functional dependency and describe how this concept relates to 2NF. Provide an example to illustrate your answer.*

Full functional dependency Indicates that if A and B are attributes of a relation, B is fully functionally dependent on A if B is functionally dependent on A, but not on any proper subset of A.

Second Normal Form (2NF) is a relation that is in first normal form and every non-primary-key attribute is fully functionally dependent on the primary key.

14.11 *Describe the concept of transitive dependency and describe how this concept relates to 3NF. Provide an example to illustrate your answer.*

Transitive dependency A condition where A, B, and C are attributes of a relation such that if A $\rightarrow$ B and B $\rightarrow$ C, then C is transitively dependent on A via B (provided that A is not functionally dependent on B or C)

Third Normal Form (3NF) is a relation that is in first and second normal form in which no non-primary-key attribute is transitively dependent on the primary key.

14.12 *Discuss how the definitions of 2NF and 3NF based on primary keys differ from the general definitions of 2NF and 3NF. Provide an example to illustrate your answer.*

The above definitions for second (2NF) and third normal form (3NF) disallow partial or transitive dependencies on the *primary key* of relations to avoid update anomalies. However, these definitions do not take into account other candidate keys of a relation, if any exist. The more general definitions for 2NF and 3NF take account of the candidate keys of a relation. Note that this requirement does not alter the definition for 1NF as this normal form is independent of keys and functional dependencies. For the general definitions, we define

that a primary-key attribute is part of any candidate key and that partial, full, and transitive dependencies are with respect to all candidate keys of a relation.

Second normal form (2NF) A relation that is in First Normal Form and every non-candidate-key attribute is fully functionally dependent on *any candidate key*.

Third normal form (3NF) A relation that is in First and Second Normal Form and in which no non-candidate-key attribute is transitively dependent on *any candidate key*.

When using the general definitions of 2NF and 3NF we must be aware of partial and transitive dependencies on all candidate keys and not just the primary key. This can make the process of normalization more complex, however the general definitions place additional constraints on the relations and may identify hidden redundancy in relations that could be missed.

Exercises

14.13 Continue the process of normalizing the Client and PropertyRentalOwner 1NF relations shown in Figure 14.13 to 3NF relations. At the end of this process check that the resultant 3NF relations are the same as those produce from the alternative ClientRental 1NF relation shown in Figure 14.16.

The benefits of using the approach that creates two or more relations from the UNF table is that some or all of the resulting tables may already be in 3NF. This is true for the Client table, which is in 3NF. However, the PropertyRentalOwner table is only in 1NF due to the presence of a partial dependency and a transitive dependency.

The PropertyRentalOwner relation is converted to 2NF with the removal of a partial dependency. The result is creation of a new relation called PropertyOwner (2NF) and a relation called Rental, which is in 3NF.

The PropertyOwner (2NF) relation is converted to 3NF with the removal of a transitive dependency. The result is the creation of a new relation called PropertyForRent (3NF) and Owner (3NF).

In summary, the process of normalizing the PropertyRentalOwner (1NF) relation resulted in the creation of the Rental (3NF), PropertyForRent (3NF), and Owner (3NF) relations, which together with the Client relation is the same four relations created from the ClientRental 1NF relation shown in Figure 13.16.

14.14 *Examine the Patient Medication Form for the Wellmeadows Hospital case study shown in Figure 14.18.*

(a) Identify the functional dependencies represented by the data shown in the form in Figure 14.18.

patientNo → fullName

wardNo → wardName
wardName → wardNo
drugNo → name, description, dosage, methodOfAdmin
patientNo, drugNo, startDate → unitsPerDay, finishDate

The functional dependencies for bedNo are unclear. If bedNo was a unique number for the entire hospital, then could say that bedNo → wardNo. However, from further examination of the requirements specification, we can observe that bedNo is to do with the allocation of patients on the waiting list to beds.

(b) *Describe and illustrate the process of normalizing the data shown in Figure 14.19 to First (1NF), Second (2NF), and Third (3NF).*

First Normal Form

patientNo, drugNo, startDate, fullName, wardNo, wardName, bedNo, name, description, dosage, methodOfAdmin, unitsPerDay, finishDate

Second Normal Form

patientNo, drugNo, startDate, wardNo, wardName, bedNo, unitsPerDay, finish Date
drugNo, name, description, dosage, methodOfAdmin
patientNo, fullName

Third Normal Form

patientNo, drugNo, startDate, wardNo, bedNo, unitsPerDay, finish Date
drugNo, name, description, dosage, methodOfAdmin
patientNo, fullName
wardNo, wardName

(c) *Identify the primary, alternate, and foreign keys in your 3NF relations.*

patientNo (FK), drugNo(FK), startDate, wardNo(FK), bedNo, unitsPerDay, finish Date
drugNo, name, description, dosage, methodOfAdmin
patientNo, fullName
wardNo, wardName (AK)

(Primary keys underlined.)

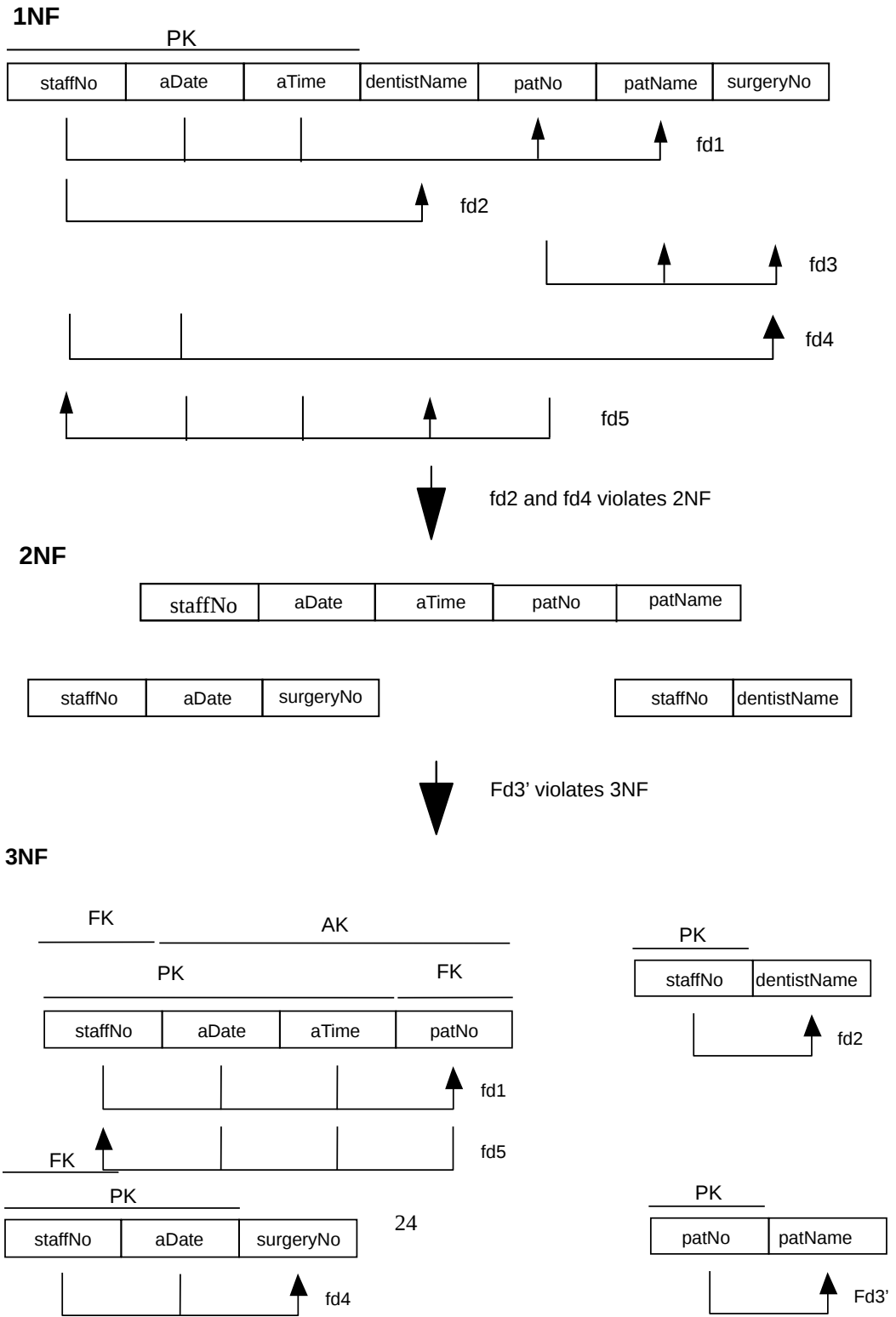
14.15 *The table shown in Figure 14.19 lists dentist/patient appointment data. A patient is given an appointment at a specific time and date with a dentist located at a particular surgery. On each day of patient appointments, a dentist is allocated to a specific surgery for that day.*

(a) *The table shown in Figure 14.19 is susceptible to update anomalies. Provide examples of insertion, deletion, and update anomalies.*

The student should provide examples of insertion, deletion and update anomalies using the data shown in the table. An example of a deletion anomaly is if we delete the details of the dentist called 'Helen Pearson', we also loose the appointment details of the patient called 'Ian MacKay'.

(b) *Describe and illustrate the process of normalizing the table shown in Figure 14.19 to 3NF. State any assumptions you make about the data shown in this table.*

The student should state any assumptions made about the data shown in the table. For example, we may assume that a patient is registered at only one surgery. Also, a patient may have more than one appointment on a given day.



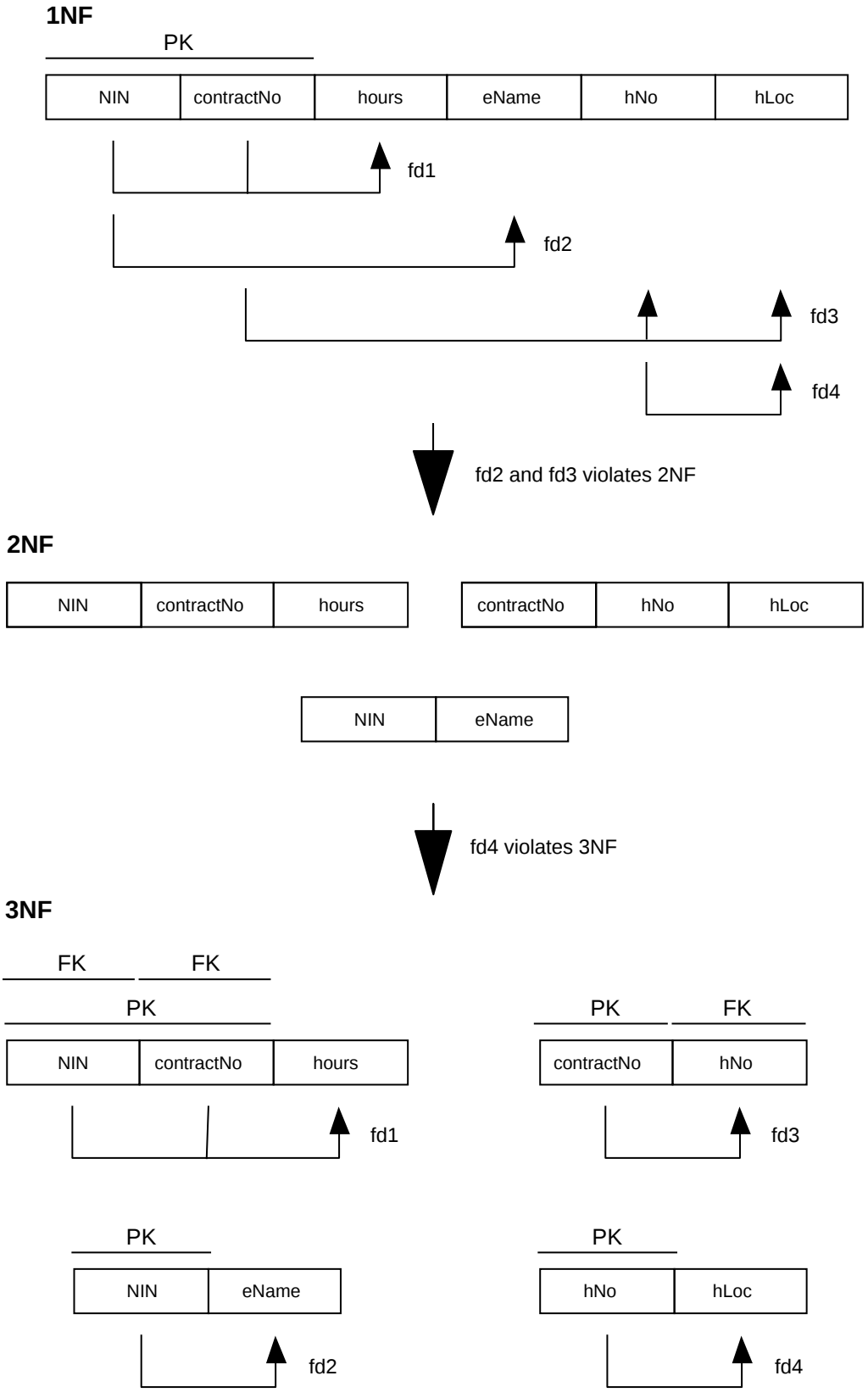
14.16 *An agency called Instant Cover supplies part-time/temporary staff to hotels within Scotland. The table shown in Figure 14.20 lists the time spent by agency staff working at various hotels. The National Insurance Number (NIN) is unique for every member of staff.*

(a) *The table shown in Figure 14.20 is susceptible to update anomalies. Provide examples of insertion, deletion, and update anomalies.*

The student should provide examples of insertion, deletion and update anomalies using the data shown in the table. An example of an update anomaly is if we wish to change the name of the employee called 'Smith J', we may only change the entry in the first row and not the last with the result that the database becomes inconsistent.

(b) *Describe and illustrate the process of normalizing the table shown in Figure 14.20 to 3NF. State any assumptions you make about the data shown in this table.*

The student should state any assumptions made about the data shown in the table. For example, we may assume that a hotel may be associated with one or more contracts.


1NF
PK
NIN
contractNo
hours
eName
hNo
hLoc
fd1
fd2
fd3
fd4
fd2 and fd3 violates 2NF
2NF
NIN
contractNo
hours
contractNo
hNo
hLoc
NIN
eName
fd4 violates 3NF
3NF
FK
FK
PK
NIN
contractNo
hours
fd1
PK
FK
contractNo
hNo
fd3
PK
NIN
eName
fd2
PK
hNo
hLoc
fd4

Chapter 15 Advanced Normalization

Review Questions

15.1 *Describe the purpose of using inference rules to identify functional dependencies for a given relation.*

Even if we restrict our attention to nontrivial functional dependencies with one-to-one (1:1) relationships that hold for all time, the complete set of functional dependencies for a given relation can still be very large. It is important to find an approach that can reduce that set to a manageable size. Ideally, we want to identify a set of functional dependencies (represented as X) for a relation that is smaller than the complete set of functional dependencies (represented as Y) for that relation and has the property that every functional dependency in Y is implied by the functional dependencies in X. Hence, if we enforce the integrity constraints defined by the functional dependencies in X, we automatically enforce the integrity constraints defined in the larger set of functional dependencies in Y. This requirement suggests that there must be functional dependencies that can be inferred from other functional dependencies. For example, functional dependencies $A \rightarrow B$ and $B \rightarrow C$ in a relation implies that the functional dependency $A \rightarrow C$ also holds in that relation. $A \rightarrow C$ is an example of a **transitive** functional dependency.

How do we begin to identify useful functional dependencies on a relation? Normally, the database designer starts by specifying functional dependencies that are semantically obvious; however, there are usually numerous other functional dependencies. In fact, the task of specifying all possible functional dependencies for 'real' database projects is more often than not, impractical. However, in this section we do consider an approach that helps identify the complete set of functional dependencies for a relation and then discuss how to achieve a minimal set of functional dependencies that can represent the complete set.

15.2 *Discuss the purpose of Armstrong's axioms.*

The set of all functional dependencies that are implied by a given set of functional dependencies X is called the **closure** of X, written X^+. We clearly need a set of rules to help compute X^+ from X. A set of inference rules, called **Armstrong's axioms**, specifies how new functional dependencies can be inferred from given ones (Armstrong, 1974). For our discussion, let A, B, and C be subsets of the attributes of the relation R. Armstrong's axioms are as follows:

(1) **Reflexivity:** If B is a subset of A, then $A \rightarrow B$

(2) **Augmentation:** If $A \rightarrow B$, then $A,C \rightarrow B,C$

(3) **Transitivity:** If $A \rightarrow B$ and $B \rightarrow C$, then $A \rightarrow C$

Note that each of these three rules can be directly proved from the definition of functional dependency. The rules are **complete** in that given a set X of functional dependencies, all functional dependencies implied by X can be derived from X using these rules. The rules are also **sound** in that no additional functional dependencies can be derived that are not implied by X. In other words, the rules can be used to derive the closure of X^+.
Several further rules can be derived from the three given above that simplify the practical task of computing X^+.

(See Section 15.1)

To begin to identify the set of functional dependencies F for a relation, typically we first identify the dependencies that are determined from the semantics of the attributes of the relation. Then, we apply Armstrong's axioms (Rules 1 to 3) to infer additional functional dependencies that are also true for that relation. A systematic way to determine these additional functional dependencies is to first determine each set of attributes A that appears on the left-hand side of some functional dependencies and then to determine the set of *all* attributes that are dependent on A. Thus, for each set of attributes A we can determine the set A^+ of attributes that are functionally determined by A based on F; (A^+ is called the **closure of A under F**).

15.3 *Discuss the purpose of Boyce-Codd Normal Form (BCNF) and describe how BCNF differs from 3NF. Provide an example to illustrate your answer.*

BCNF is based on functional dependencies that take into account all candidate keys in a relation, however BCNF also has additional constraints compared with the general definition of 3NF given above.

Boyce–Codd normal form (BCNF) A relation is in BCNF if and only if every determinant is a candidate key.

To test whether a relation is in BCNF, we identify all the determinants and make sure that they are candidate keys.

See Sections 15.2 and 15.3.

15.4 *Describe the concept of multi-valued dependency and describe how this concept relates to 4NF. Provide an example to illustrate your answer.*

Multi-valued Dependency (MVD) Represents a dependency between attributes (for example, A, B, and C) in a relation, such that for each value of A there is a set of values for B and a set of values for C. However, the set of values for B and C are independent of each other.

A multi-valued dependency can be defined as being **trivial** or **nontrivial**. A MVD A ——➢➢ B in relation R is defined as being trivial if (a) B is a subset of A *or* (b) A ∪ B = R. A MVD is defined as being nontrivial if neither (a) nor (b) are satisfied. A trivial MVD does not specify a constraint on a relation, while a nontrivial MVD does specify a constraint.

Fourth Normal Form (4NF) A relation that is in Boyce-Codd Normal Form and contains no nontrivial multi-valued dependencies.

Fourth normal form (4NF) is a stronger normal form than BCNF as it prevents relations from containing nontrivial MVDs, and hence data redundancy. The normalization of BCNF relations to 4NF involves the removal of the MVD from the relation by placing the attribute(s) in a new relation along with a copy of the determinant(s).

See Section 15.4.

15.5 *Describe the concept of join dependency and describe how this concept relates to 5NF. Provide an example to illustrate your answer.*

Lossless-join dependency A property of decomposition, which ensures that no spurious tuples are generated when relations are reunited through a natural join operation.

In splitting relations by projection, we are very explicit about the method of decomposition. In particular, we are careful to use projections that can be reversed by joining the resulting relations, so that the original relation is reconstructed. Such a decomposition is called a **lossless-join** (also called a *nonloss* or *nonadditive*) decomposition, because it preserves all the data in the original relation and does not result in the creation of additional spurious tuples. However, there are cases were we require to perform a lossless-join decompose of a relation into more than two relations. These cases are the focus of the lossless-join dependency and fifth normal form (5NF).

Fifth Normal Form (5NF) A relation that has no join dependency.

Fifth normal form (5NF) (also called project-join normal form (PJNF)) specifies that a 5NF relation has no join dependency.

See Section 15.5.

Exercises

15.6 *On completion of Exercise 14.14 examine the 3NF relations created to represent the attributes shown in the Wellmeadows Hospital form shown in Figure 14.18. Determine whether these*

relations are also in BCNF. If not, transform the relations that do not conform into BCNF.

The only relations that may violate BCNF are those that have more than one candidate key. Therefore we need only re-examine the Ward relation, which has a wardNo as a PK and wardName as an alternate key. This relation contains the following functional dependencies:

wardNo → wardName (fd1)

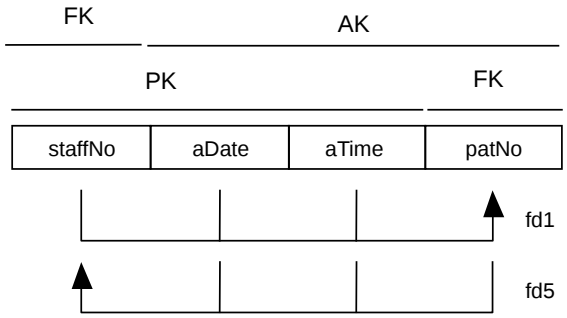
wardName → wardNo (fd2)

The presence of fd2 does not break BCNF because wardName is a candidate key for this relation. Hence the Ward relation is in BCNF.

As the other relations shown in the answer for Exercise *14.14* have only one candidate key, they must also be in BCNF.

15.7 On completion of Exercise 14.15 examine the 3NF relations created to represent the attributes shown in the relation that displays dentist/patient appointment data in Figure 14.19. Determine whether these relations are also in BCNF. If not, transform the relations that do not conform into BCNF.

The only relations that may violate BCNF are those that have more than one candidate key. Therefore we need only re-examine the Appointment relation, which has (staffNo, aDate, aTime) as a PK and (patNo, aDate, aTime) as an alternate key. This relation contains the following functional dependencies:



The presence of fd5 does not break BCNF because (patNo, aDate, aTime) is a candidate key for this relation. Hence the Appointment relation is in BCNF.

As the other relations shown in the answer for Exercise *14.15* have only one candidate key, they must also be in BCNF.

15.8 On completion of Exercise 14.16 examine the 3NF relations created to represent the attributes shown in the relation displaying employee contract data for an agency called Instant Cover in Figure 1420. Determine whether these relations are also in BCNF. If not, transform the relations

that do not conform into BCNF.

The only relations that may violate BCNF are those that have more than one candidate key.
Therefore for this exercise, as there are no relations that satify this criteria, the relations shown in the answer for Exercise 14.16 must already be in BCNF.

15.9 *The relation shown in Figure 15.11 lists members of staff (staffName) working in a given ward (wardName) and patients (patientName) allocated to a given ward. There is no relationship between members of staff and patients in each ward. In this example assume that staff name (staffName) uniquely identifies each member of staff and that the patient name (patientName) uniquely identifies each patient.*

wardName	staffName	patientName
Pediatrics	Kim Jones	Claire Johnson
Pediatrics	Kim Jones	Brian White
Pediatrics	Stephen Ball	Claire Johnson
Pediatrics	Stephen Ball	Brian White

Figure 15.11 The WardStaffPatient relation.

(a) *Describe why the relation shown in Figure 15.11 is in BCNF and not in 4NF.*

wardName ——➢➢ staffName

wardName ——➢➢ patientName

Relation is in BCNF but there is a nontrivial multi-valued dependency in the relation, so relation is not in 4NF.

(b) *The relation shown in Figure 15.11 is susceptible to update anomalies. Provide examples of insertion, deletion, and update anomalies.*

If we wanted to insert a new patient name, would have to add two records, one for each member of staff.
If we wanted to update the name of patient Claire Johnson, we would have to update two records.
If we wanted to delete the record corresponding to patient Claire Johnson, we would have to delete two records.

(c) *Describe and illustrate the process of normalizing the relation shown in Figure 15.11 to 4NF.*

To remove the MVD, we create two new relations:

WardStaff (wardName, staffName)
WardPatient(wardName, patientName)

15.10 *The relation shown in Figure 15.12 describes hospitals (hospitalName) that require certain items (itemDescription), which are supplied by suppliers (supplierNo) to the hospitals (hospitalName). Furthermore, whenever a hospital (h) requires a certain item (i) and a supplier (s) supplies that item (i) and the supplier (s) already supplies at least one item to that hospital (h), then the supplier (s) will also be supplying the required item (i) to the hospital (h). In this example, assume that a description of an item (itemDescription) uniquely identifies each type of item.*

hospitalName	itemDescription	supplierNo
Western General	Antiseptic Wipes	S1
Western General	Paper Towels	S2
Yorkhill	Antiseptic Wipes	S2
Western General	Antiseptic Wipes	S2

Figure 15.12 The HospitalItemSupplier relation.

(a) *Describe why the relation shown in Figure 15.12 is not in 5NF.*

This relation has a join dependency JD(hospitalName, itemDescription, supplierNo) among the three projections: R1(hospitalName, itemDescription), R2(hospitalName, supplierNo), and R3(itemDescription, supplierNo) of HospitalItemSupplier.

(b) Describe and illustrate the process of normalizing the relation shown in Figure 15.12 to 5NF.

To remove the join dependency, we create the following 5NF relations:

HospitalItem(hospitalName, itemDescription)
HospitalSupplier(hospitalName, supplierNo)
ItemSupplier(itemDescription, supplierNo)