

Table of contents

- [Unit 1: Introduction to Database Systems](#)
 - [Section 1—File Processing Systems](#)
 - [Section 2—Database Management Systems](#)
 - [Section 3—Database Development Life Cycle](#)
- [Unit 2: Data Modeling: The Entity-Relationship Diagram](#)
 - [Section 1—The Entity-Relationship Model](#)
 - [Section 2—The Enhanced Entity-Relationship Model](#)
- [Unit 3: The Relational Model and Normalization](#)
 - [Section 1—The Relational Data Model](#)
 - [Section 2—Normalization](#)
 - [Notes on Normalization](#)
- [Unit 4: Physical Database Design](#)
 - [Section 1—Steps in Physical Database Design](#)
 - [Section 2—File Organization and Database Architectures](#)
- [Unit 5: SQL—A Standard Navigation Language for Relational Databases](#)
 - [Section 1—SQL as a Data Definition Language \(DDL\)](#)
 - [Section 2—SQL as a Data Manipulation Language \(DML\)](#)
 - [Section 3—Advanced SQL and Data Control Commands](#)
- [Unit 6: Introduction to Two-Tier and Three-Tier Architectures, and the Internet Database Environment](#)
 - [Section 1—Client/Server Architectures and Database Connection to Web Applications](#)
 - [Section 2—XML and Data Exchange on the Internet](#)
- [Unit 7: Introduction to Data Warehousing](#)
 - [Section 1—Basic Concepts of Data Warehousing](#)
 - [Section 2—Data Warehouse Architectures and OLAP Tools](#)
- [Unit 8: Data Quality and Database Administration](#)
 - [Section 1—Data Quality and Integration](#)
 - [Section 2—Data Administration and Database Administration](#)
- [Unit 9: Overview of Object-Oriented Databases](#)
 - [Section 1—Object-Oriented Data Modeling](#)
 - [Section 2—Implementing Object Persistence using Relational Databases](#)

Unit 1 Introduction to Database Systems

Commentary

In this unit, you will learn about databases: a convenient resource for information management. The advantages and disadvantages of databases will be outlined, and compared to those of traditional file processing systems for information management. In this unit, we will also describe the steps involved in the process of database development, the purpose of each step, and the database development activities taken at each step. A description of the team of specialists involved in the development process is given, as well as a description of each specialist's area. Finally, related views/schemas of a database are described.

Activities

- View the [Camtasia Video Tutorial](#) for *Chapter 1: Introduction to Databases* from the textbook's Web site.

Unit 1 Introduction to Database Systems

Section 1 File Processing Systems

This section introduces you to the basic concepts related to data management, and the advantages and disadvantages of the traditional file processing approach to information management.

Learning Outcomes

When you have completed this section, you should be able to

- discuss the difference between data, information, and metadata;
- outline the characteristics of file processing systems, including their advantages and limitations.

Readings

- Read the Introduction, including the Learning Objectives and Data Matters, and the sections “Basic concepts and definitions” and “Traditional file processing” from Chapter 1: “The Database Environment and Development Process” in the textbook.

Note: All readings in the *Study Guide* refer to the textbook: Hoffer, J. A., Ramesh, V., &

i, H. (2011). *Modern database management* (10th ed.) [eText version]. Prentice Hall.

Tasks

Practice the following questions and exercises from Chapter 1 of the textbook. You may want to use the Unit 1 Workspace for the tasks in this and the following sections.

- Review Questions: 4 and 5
- Problems and Exercises: 3 and 6

Key Terms

You may want to use the student-created Glossary of Key Terms on the course home page to define the following:

information, metadata, database management systems

Unit 1 Introduction to Database Systems

Section 2 Database Management Systems

Database management systems, developed to overcome the limitations of traditional file systems, have become the standard approach to handling data for large organizations. This section introduces the concept of databases, and the types, benefits, and components of the database environment.

Learning Outcomes

When you have completed this section, you should be able to

- differentiate between data and information;
- explain the database approach to information management;
- discuss the five categories of database applications;
- give an overview of a database environment;
- list some of the key decisions that must be made before developing a database in a particular category of database application;
- compare databases and file processing systems;
- discuss the database approach to information management, recognizing that it can be implemented using both traditional programming languages and database products;
- list the goals that drove the development and evolution of database systems.

Readings

- Read “The database approach,” “Components of the database environment,” and “The range of database applications process” from Chapter 1: “The Database Environment and Development Process.”

Tasks

Practice the following questions and exercises from Chapter 1.

- Review Questions: 7, 9, 10, and 11
- Problems and Exercises: 7 and 9

Key Terms

You may want to use the student—created Glossary of Key Terms on the course home page to define the following:

database application, metadata, user view, data independence, systems development life cycle, prototyping, database management system

Unit 1 Introduction to Database Systems

Section 3 Database Development Life Cycle

This section provides a broad overview of the steps involved in the process of developing databases, beginning with a description of the systems development life cycle. A description of the team of specialists responsible for developing a database will follow, as will an illustration of the three-schema architecture for database development.

Learning Outcomes

When you have completed this section, you should be able to

- identify the steps involved in developing a database, and the activities in each step;
- explain the prototyping process;
- describe the roles of the specialists involved in the process of database development;
- list the three different schemas of database development, and explain the differences between them;
- present a case study of database development.

Readings

- Read “The database development process” and “Developing a database application for Pine Valley Furniture” from Chapter 1: “The Database Environment and Development Process.”

Tasks

Practice the following questions and exercises from Chapter 1.

- Review Questions: 13, 15, and 16
- Problems and Exercises: 13, 14, and 16

Key Terms

You may want to use the student-created Glossary of Key Terms on the course home page to define the following:

systems development life cycle, prototyping, repository, project, incremental commitment, external schema (user view), conceptual schema, physical schema

Unit 2 Data Modeling: The Entity-Relationship Diagram

Commentary

This unit explains and illustrates the use of the entity-relationship (E-R) model, and introduces a set of widely accepted notations and constructs from which most E-R variants are derived. Although no unique E-R model has been generally accepted, it is a standard tool in the process of conceptual data modeling.

Activities

View the Camtasia Video Tutorials for *Chapter 2: Data Describe the Organization: Part 1—The Basics*, and *Chapter 3: Data Describe the Organization: Part 2—Advanced E-R Modeling* from the textbook's Web site.

Lab Manual

Computer Science 378 is accompanied by a set of labs to familiarize students with database techniques and languages that are being used in work environments. The labs use the Teradata University Network, which is a Web-based portal for practicing database techniques.

Note: The labs specified in the course are designed to allow you to practice the learned concepts, and are not part of the marked Assignments. Please do not submit the lab work for marking, but feel free to contact your tutor if you have any question or need help with the labs.

Lab 1: Starting up your account in the Teradata University Network (TUN)

1. Register for TUN as a student member at <http://www.teradatauniversitynetwork.com/tun/> (For the student access password please check in the Course News and Announcements or contact your tutor).
2. Create an account for the Teradata SQL Assistant (**Note:** this account is different from the one you created to be a TUN member): <http://tunweb.teradata.ws/tunstudent/>

To complete the registration form, please use the following information:

- **University:** Athabasca University
- **Instructor:** Larbi Esmahi
- **Course:** *COMP 378: Introduction to Database Management*
- **Course Password:** There is no course password, so just leave this field blank

3. Read the SQL Assistant Web Edition User Manuals, which can be found under “Help” on the main menu of <http://tunweb.teradata.ws/tunstudent/>. Note that there are a number of additional user manuals available under the Help menu, if you need them (but you are not required to read them for the course).
4. Log in to SQL Assistant, but leave the ‘Default database’ field blank.
5. Familiarize yourself with the SQL Assistant using the following queries:
 - a. Create a table in your default database.

```
Create table PERSON_t  
(person_id integer,  
address varchar(20),  
telephone varchar(15));
```

- b. Insert records in the created table.

```
Insert into PERSON_t values  
(10, '123 sesame street', '7803333222') ;  
Insert into PERSON_t values  
(11, '1 university drive', '7805555444') ;
```

- c. Select records from the table.

```
Select * from PERSON_t ;7.
```

6. Log out of the SQL Assistant and TUN.

Unit 2 Data Modeling: The Entity-Relationship Diagram

Section 1 The Entity-Relationship Model

In this section, you will learn how to use the E-R model to map entities (i.e., objects) in the real world, and to graphically represent the relationships between entities, as a step in database design. The goal of this mapping (or modeling) is to determine what data and relationships should be stored in a database.

Learning Outcomes

When you have completed this section, you should be able to

- define the key terms of the entity-relationship model;
- explain the business rules concept;
- name and define data objects;
- map multivalued attributes and repeating groups in an entity-relationship diagram;
- give examples of unary, binary, and ternary relationships, and give their cardinalities;
- model simple time-dependent data;
- draw E-R diagrams for several business environments.

Readings

- Read all sections of Chapter 2: “Modeling Data in the Organization.”
- Read Appendix A: “Data Modeling Tools and Notations.”

Tasks

Practice the following questions and exercises from Chapter 2. You may want to use the Unit 2 Workspace for the tasks in this and the following sections.

- Review Questions: 5, 6, 8, 9, 17, and 21
- Problems and Exercises: 5, 6, 21, 24, and 26

Key Terms

You may want to use the student-created Glossary of Key Terms on the course home page to define the following:

entity-relationship model, entity-relationship diagram, business rule, entity, entity type, entity instance, strong entity type, weak entity type, identifying owner,

identifying relationship, attribute, composite attribute, simple attribute, multi-valued attribute, derived attribute, identifier, composite identifier, relationship type, relationship instance, associative entity, degree of a relationship, unary relationship, binary relationship, ternary relationship, cardinality constraints, minimum cardinality, maximum cardinality, timestamp

Unit 2 Data Modeling: The Entity-Relationship Diagram

Section 2 The Enhanced Entity-Relationship Model

This section introduces some of the notations, constructs, and symbols used to model complex business rules in today's business environment. These business rules can restrict the processing of entities. This section also concentrates on the use of supertype/subtype relationships.

Learning Outcomes

When you have completed this section, you should be able to

- define the following key terms: enhanced entity-relationship model, supertype entity, subtype entity, attribute inheritance, subtype discriminator, generalization, and specialization;
- develop supertype/subtype relationships using generalization and specialization processes;
- explain how an entity cluster is used to simplify complex diagrams;
- specify constraints for supertype/subtype relationships;
- discuss packaged data models.

Readings

- Read all sections of Chapter 3: “The Enhanced E-R Model.”

Tasks

Practice the following questions and exercises from Chapter 3.

- Review Questions: 4, 5, 10,11, and 14
- Problems and Exercises: 2, 7, 14, and 15

Key Terms

You may want to use the student-created Glossary of Key Terms on the course home page to define the following:

enhanced entity-relationship model, supertype entity, subtype entity, attribute inheritance, subtype discriminator, generalization, specialization

Unit 3 The Relational Model and Normalization

Commentary

In this unit, you will learn about logical database design. This process involves transforming an entity-relationship data model, which represents an organization's requirements for data, into a database model. We will emphasize the relational model, since it is the most widely used. The steps in designing a well-structured relational model (normalization) are discussed in detail.

Activities

View the Camtasia Video Tutorial for *Chapter 4: Normalization* from the textbook's Web site.

Assignment 1

Complete [Assignment 1](#), and submit it to your tutor for evaluation and feedback.

Unit 3 The Relational Model and Normalization

Section 1 The Relational Data Model

This section discusses the relational data model in detail, beginning with some key terms and definitions. We will then discuss a few essential constraints needed to keep a database consistent, integrated, and accurate. Finally, we will explain the set of rules used to transform an enhanced entity-relationship (EER) diagram into a relational model.

Learning Outcomes

When you have completed this section, you should be able to

- discuss the properties of relations;
- discuss integrity constraints, and list the properties of well-structured relations;
- select the primary key of a relation;
- list and discuss the three types of anomaly;
- list and discuss the steps involved in transforming an EER diagram into a relational model;
- transform an EER diagram into a set of relations.

Readings

- Read the sections “The relational data model,” “Integrity constraints,” and “Transforming EER diagrams into relations” from Chapter 4: “Logical Database Design and the Relational Model.”

Tasks

Practice the following questions and exercises from Chapter 4. You may want to use the Unit 3 Workspace for the tasks in this and the following sections.

- Review Questions: 4, 5, 6, and 9
- Problems and Exercises: 1.I and 2.I

Key Terms

You may want to use the student-created Glossary of Key Terms on the course home page to define the following:

relation, primary key, composite key, foreign key, null, entity integrity rule, referential integrity constraint, well-structured relation, anomalies, recursive foreign key

Unit 3 The Relational Model and Normalization

Section 2 Normalization

This section discusses the concepts of normalization. Normalization is the process of converting a set of relations with anomalies (i.e., problems) to a set of relations free of anomalies.

Learning Outcomes

When you have completed this section, you should be able to

- describe the steps in normalization;
- determine the candidate key of a relation;
- define the determinants of a relation;
- transform an E-R diagram into a set of normalized relations;
- merge relations to remove redundancy;
- identify synonyms and homonyms.

Readings

- Read the sections “Introduction to normalization,” “Normalization example,” “Merging relations,” and “A final step for defining relational keys” from Chapter 4: “Logical Database Design and the Relational Model.”
- Read Appendix B: “Advanced Normal Forms.”
- Read “Notes on Normalization” below.

Tasks

Practice the following questions and exercises from Chapter 4.

- Review Questions: 8, 13, 16, and 22
- Problems and Exercises: 3, 4, 8, 13, and 21

Key Terms

You may want to use the student-created Glossary of Key Terms on the course home page to define the following:

normalization, normal forms, functional dependency, determinant, candidate key, first normal form, second normal form, partial functional dependency, third normal form, transitive dependency, synonyms, alias, homonym

Unit 3 The Relational Model and Normalization

Section 2 Normalization

Notes on Normalization

The aim of these notes is to explain normalization and normal forms in simple terms so you can better understand the concepts involved.

The first phase in database development involves mapping into relations the entity-relationship diagrams that represent a business organization. If this mapping is done poorly, then the resulting relational model will suffer from modification anomalies (problems) when data is inserted or deleted. Normalization is the process of applying a set of rules, called normal forms, to the relational model in order to obtain a well-designed relational model. Theoretically, there are six normal forms that can be applied to relations: 1NF, 2NF, 3NF, BCNF, 4NF, and 5NF; however in practice, you will usually need only the first three.

Each normal form builds on the one before it, such that a relation can only be in 2NF if it is first in 1NF, and conforms to the 2NF rules. For a relation to be in 4NF, it must go through 1NF, 2NF, 3NF, BCNF, and conform to the 4NF rules. For practical purposes, putting relations in BCNF will result in a well-designed relational model, so we will mainly focus on the first four normal forms.

Just for your information, there is also a stand-alone normal form called the Domain-Key Normal Form (DKNF). If a relation can be placed in the DKNF, then it is guaranteed to be free of modification anomalies. Unfortunately, it is not known whether it is possible to put any given relation in DKNF. Additionally, this rule is complex, and is not required to obtain a well-designed relational model. To avoid confusion, we will not discuss this rule here.

Since normal forms are concerned with the relational model, we have to make sure that the tables we have are truly relations. This is accomplished by applying the first normal form (1NF).

- I. A relation is in 1NF if any repeating groups have been removed. This condition applies to any well-structured relation. Consider the following example.

A university department offers courses to students who pay tuition fees. The following relation represents this situation.

DEPARTMENT (student-no, course-no, fee)

DEPARTMENT (student-no, course-no, fee)

student-no **course-no** **fee**

| | | |
|------|------|-----|
| S101 | C213 | 500 |
| S201 | C213 | 500 |
| S301 | C413 | 550 |

What will happen if S301 decides to drop course C413, and s/he is the only student enrolled in the course? If we delete the third row (tuple), then not only do we delete the piece of information saying that S301 is enrolled in C413, but we also delete the rest of the information about C413. We will have removed the fact that C413 is being offered by the department, and that it costs \$550 to enroll. This is an example of a deletion anomaly, because more information was removed than was intended.

Now consider another situation. If the department decides to offer the course C513, that piece of information cannot be entered until a student enrolls in the course because, according to the definition of a relation, an attribute cannot have a null value. This is an example of an insertion anomaly.

These anomalies occur because the key is compound, and the non-key attribute (i.e., the fee) depends on only part of the key. The second normal form (2NF) resolves this problem.

- II. A relation is in 2NF if it is in 1NF, and every non-key attribute depends on the primary key. If the key is a single attribute, then it will be in 2NF. If the key is compound (i.e., has more than one attribute), then all non-key attributes should depend on the whole key.

In our example above, fee depends only on course (i.e., only part of the key), and not on the whole compound key. This relation can therefore be split into two relations, each of which is in 2NF.

| STUDENT (student-no, course-no) | | COURSE (course-no, fee) | |
|---------------------------------|-----------|-------------------------|-----|
| student-no | course-no | course-no | fee |
| S101 | C213 | C213 | 500 |
| S201 | C213 | C413 | 550 |
| S301 | C413 | | |

Examine each of these two relations for the modification anomalies mentioned above, and assure yourself that each is in 2NF.

- III. A relation is in 3NF if the relation is in 2NF, and there are no transitive dependencies. Consider the following situation.

Young children ride buses between home and school. Each student pays a rate according to bus type: basic (B), loaded (L), or fully loaded (FL). We can represent this situation with the following relation.

TRANSPORTATION (student-no, bus-type, rate)

| student-no | bus-type | rate |
|-------------------|-----------------|-------------|
| S101 | B | 100 |
| S201 | B | 100 |
| S301 | L | 120 |

This table is a relation, so it is in 1NF by definition. The key is a single attribute (i.e., there is no compound key), so it is in 2NF. However, this table still suffers from modification anomalies. If student S301 moves from bus type L to B, or changes schools, and s/he is the only student using that bus, then we will have to delete that row (tuple). In this case, we will also delete more information than we want: we will delete the piece of information saying that there is a bus of type L, and that it costs \$120 (a deletion anomaly). Also, since no student is currently riding a bus of type FL, we cannot insert that piece of information in the relation because we cannot enter a null value in the student-no attribute (an insertion anomaly). These anomalies occur as a result of transitive dependency. Student-no determines bus-type, and bus-type determines rate; therefore, student-no determines rate because rate is mainly associated with bus-type, but the key is student-no. To put this relation in 3NF, we must split it into:

STUDENT (student-no, bus-type) BUS (bus-type, rate)

| student-no | bus-type | bus-type | rate |
|-------------------|-----------------|-----------------|-------------|
| S101 | B | B | 100 |
| S201 | B | L | 120 |
| S301 | L | FL | 150 |

Examine these two relations, and assure yourself that they are in 3NF by making sure that they are in 1NF and 2NF, and that neither has a transitive dependency.

- IV. A relation is in BCNF if the relation is in 3NF, and every determinant is a candidate key. Consider the following situation.

Students in a school are asked to join activities where they will be supervised and trained by coaches. Each student may join more than one activity. Each coach may train students in only one activity, but each activity may have more than one coach. If many students choose an activity with more than one coach, they will be divided into groups, and each group will be supervised by one coach. The following relation represents this situation.

ST-AC-CO (student-no, activity, coach)

| student-no | activity | coach |
|-------------------|-----------------|--------------|
| S101 | A1 | C1 |
| S101 | A2 | C2 |

| | | |
|------|----|----|
| S202 | A1 | C1 |
| S303 | A1 | C3 |
| S303 | A2 | C2 |
| S303 | A3 | C4 |
| S404 | A3 | C5 |

ST-AC-CO is in 3NF because it is a relation (1NF), every non-key attribute depends on the whole key (2NF), and it has no transitive dependency (3NF); however, it still has modification anomalies. If S303 no longer participates in A1, and we delete the fourth row (tuple), then we will also delete the information that tells us C3 is ready to train students who enroll in A1 (deletion anomaly). Also, if C6 is ready to train students in A4, but no student is enrolled in A4, then we cannot insert the piece of information about C4 (insertion anomalies).

The 'coach' attribute is a determinant because it determines 'activity' (not the opposite). To put the above relation in BCNF, we have to split ST-AC-CO into:

ST-CO (student-no, coach) CO-AC (coach, activity)

| student-no | coach | coach | activity |
|-------------------|--------------|--------------|-----------------|
| S101 | C1 | C1 | A1 |
| S101 | C2 | C2 | A2 |
| S202 | C1 | C3 | A1 |
| S303 | C3 | C4 | A3 |
| S303 | C2 | C5 | A3 |
| S303 | C4 | | |
| S404 | C5 | | |

Examine ST-CO and CO-AC to make sure they are in 1NF, 2NF, 3NF, and BCNF, and free of the modification anomalies that exist in ST-AC-CO. Check to see if dividing ST-AC-CO into ST-AC and AC-CO is an acceptable alternative to ST-CO and CO-AC.

- V. A relation is in 4NF if it is in BCNF, and has no multi-valued independent attributes (i.e., there are no multi-valued attributes; or if multi-valued attributes exist, then they are functionally dependent). Consider the following situation.

Students are encouraged to participate in extracurricular activities (or hobbies), and we want to record the students' courses and hobbies. Each student may have more than one course and more than one hobby. This situation is represented by the relation ST-CO-HO-A.

ST-CO-HO-A (student-no, course-no, hobby-name)

| student-no | course-no | hobby-name |
|-------------------|------------------|-------------------|
| S101 | C213 | billiards |
| S101 | C413 | billiards |
| S201 | C213 | football |
| S301 | C413 | chess |
| S301 | C213 | basketball |
| S301 | C213 | swimming |

Consider the following anomalies in this relation.

1. Redundancy: S101 has two courses and one hobby, forcing us to repeat the value of his/her hobby because the hobby-name attribute in the second row cannot have a null value.
2. Deletion anomaly: If S210 drops C213, and we delete that row, then we will also delete the fact that he plays football.
3. Insertion anomaly: If S401 is enrolled in course C213, but does not have a hobby, this information cannot be recorded because the hobby-name attribute cannot have a null value.
4. False impressions: From S301, one may have the impression that C413 encourages intellectual activities, since chess is in the same row; whereas C213 encourages physical activities, such as basketball and swimming. In fact, there is no relationship between courses and hobbies. To remove this false impression, we would have to repeat each course with each hobby, giving us the relation ST-CO-HO-B.

ST-CO-HO-B (student-no, course-no, hobby-name)

| student-no | course-no | hobby-name |
|-------------------|------------------|-------------------|
| S101 | C213 | billiards |
| S101 | C413 | billiards |
| S201 | C213 | football |
| S301 | C413 | chess |
| S301 | C213 | basketball |
| S301 | C213 | swimming |
| S301 | C413 | swimming |
| S301 | C413 | basketball |
| S301 | C213 | chess |

Note the redundancy in ST-CO-HO-B. This redundancy occurs because the two attributes (course-no and hobby-name) can have multiple values for each student, and there is no relationship between the values of the attributes. ST-CO-HO-B is in 1NF, 2NF, 3NF, and BCNF,

but it still has the same anomalies mentioned in points 1, 2, and 3 (above). To resolve these anomalies by placing the relation in 4NF, we must split ST-CO-HO-B into:

STU-CO (student-no, course-no) STU-HO (student-no, hobby-name)

| student-no | course-no | student-no | hobby-name |
|-------------------|------------------|-------------------|-------------------|
| S101 | C213 | S101 | billiards |
| S101 | C413 | S201 | football |
| S201 | C213 | S301 | chess |
| S301 | C413 | S301 | basketball |
| S301 | C213 | S301 | swimming |

Examine STU-CO and STU-HO, assure yourself that they are free of the modification anomalies present in ST-CO-HO-B, and that they are both in 4NF.

Unit 4 Physical Database Design

Commentary

In this unit, you will learn how to transform a logical database model into a physical database model. The physical model is the second and last stage of database design. It is a detailed specification of the database structure that involves the definition of physical names, formats, and locations of physical records and their fields; the physical representation and organization of records and the relationships among them; and the recognition of constraints and keys.

Activities

Project

Start working on the Project, and communicate with your tutor regarding the software you will use to implement it.

Online Quiz

Complete the Online Quiz on the course home page under Unit 4.

Unit 4 Physical Database Design

Section 1 Steps in Physical Database Design

This section discusses the physical database design process used to produce a single, centralized, and efficient database. Physical database design is concerned with the internal level of the database, which is the responsibility of physical database designers. Designers work with disk drives, pointers, physical addresses, and indexes to select the most efficient data access methods.

Learning Outcomes

When you have completed this section, you should be able to

- list the information required for physical file and database design;
- describe the important decisions to be made in the physical database design process;
- explain why data volume and usage analysis are important inputs to physical database design;
- discuss the criteria for selecting a data type;
- explain the importance and use of coding and compression techniques;
- outline the integrity controls supported by the database management system (DBMS);
- identify and discuss methods of handling missing data;
- outline situations where denormalization would be preferred over normalization;
- discuss the advantages and disadvantages of partitioning;
- compare vertical and horizontal partitioning.

Readings

- Read the sections “The physical database design process,” “Designing fields,” and “Denormalizing and partitioning data” from Chapter 5: “Physical Database Design and Performance.”

Tasks

Practice the following questions and exercises from Chapter 5. You may want to use the Unit 4 Workspace for the tasks in this and the following sections.

- Review Questions: 4, 5, 12, 13, and 14
- Problems and Exercises: 7, 8, and 9

Key Terms

You may want to use the student-created Glossary of Key Terms on the course home page to define the following:

field, data type, data integrity, denormalization, horizontal partitioning, vertical partitioning

Unit 4 Physical Database Design

Section 2 File Organization and Database Architectures

This section describes data storage and indexing options, and discusses criteria for choosing appropriate solutions. It also discusses query optimization, and the use of parallel processing to improve file access performance.

Learning Outcomes

When you have completed this section, you should be able to

- list and discuss the criteria for selecting a file organization;
- compare the three basic file organization methods;
- organize files according to the three basic file organization methods;
- cluster records to reduce data access time;
- determine when to use indexes;
- describe parallel processing, and explain how it is used to improve performance.

Readings

- Read the sections “Designing physical database files,” “Using and selecting indexes,” and “Designing a database for optimal query performance” from Chapter 5: “Physical Database Design and Performance.”

Tasks

Practice the following questions and exercises from Chapter 5.

- Review Questions: 17, 19, 21, and 22
- Problems and Exercises: 12, 13, 16, and 19

Key Terms

You may want to use the student-created Glossary of Key Terms on the course home page to define the following:

physical file, tablespace, extent, file organization, sequential file organization, indexed file organization, index, secondary key, join index, pointer, hashed file organization, hashing algorithm, hash index table

Commentary

This unit describes a standard data definition and manipulation language used in the creation and processing of relational databases. SQL is used to define the logical/conceptual schema of relational databases. It is also used to express queries, create views, and retrieve, insert, delete, and update the values of relations in a relational database.

Activities

View the Camtasia Video Tutorials for *Chapter 6: Writing SQL Queries: Part 1—Single Table Queries*, and *Chapter 7: Writing SQL Queries: Part 2—Multiple Table Queries* from the textbook's Web site.

Lab 2: Tables, Views, SQL Queries, and Triggers

Use the Teradata University Network to implement the following tables.

Table 1: 'Employees' includes the first name, middle initial, and last name of every worker in the company, as well as their SIN, date of birth, address, sex, salary, and the number of the department with which they are associated.

Table 2: 'Departments' indicates the name (Consumer Products, Industrial Products, and Research) and number of each department in the company, and the SIN and start date of the manager of each department.

Table 3: 'Projects' includes the project name and number, the number of the department in charge of the project, and the location of the office working on the project.

Table 4: 'Locations' lists the locations of all the offices of each department.

Table 1: Employees

| <u>FNAME</u> | <u>MINIT</u> | <u>LNAME</u> | <u>SIN</u> | <u>BDATE</u> | <u>ADDRESS</u> | <u>GENDER</u> | <u>SALARY</u> | <u>DNO</u> |
|--------------|--------------|--------------|------------|--------------|----------------|---------------|---------------|------------|
| Harris | T | Chomsky | 123 | 1965-12-10 | Edmonton | M | 50 000 | 12 |
| Kristian | C | Bohr | 456 | 1975-10-05 | Ottawa | M | 28 000 | 11 |
| Charlotte | F | Bouchard | 789 | 1985-08-06 | Montreal | F | 40 000 | 11 |
| Said | J | Ahmad | 111 | 1970-09-07 | Toronto | M | 30 000 | 12 |
| Andrew | U | Brahe | 222 | 1970-04-02 | Winnipeg | M | 20 000 | 10 |

| | | | | | | | | |
|-------|---|---------|-----|------------|-----------|---|--------|----|
| Nadia | O | Mamary | 333 | 1960-01-08 | Saskatoon | F | 35 000 | 10 |
| Yuan | P | Nielsen | 987 | 1983-02-27 | Moncton | F | 32 000 | 11 |
| Neil | A | Dion | 654 | 1953-02-27 | Moncton | M | 32 000 | 11 |
| Karen | C | Ming | 321 | 1963-11-16 | Victoria | F | 26 000 | 12 |

Table 2: Departments

| <u>DNAME</u> | <u>DNUMBER</u> | <u>MGRSIN</u> | <u>MGRSTARTDATE</u> |
|--------------|----------------|---------------|---------------------|
| ConsProd | 10 | 333 | 2004-10-01 |
| InduProd | 11 | 654 | 2005-05-01 |
| Research | 12 | 111 | 2000-06-15 |

Table 3: Projects

| <u>PNAME</u> | <u>PNUMBER</u> | <u>PLOCATION</u> | <u>DNUM</u> |
|-------------------|----------------|------------------|-------------|
| Mobile University | 1 | Ottawa | 10 |
| Interactive TV | 2 | Ottawa | 12 |
| Intelligent Agent | 3 | Athabasca | 11 |
| E-Commerce | 4 | Edmonton | 10 |
| Mobile Office | 5 | Athabasca | 11 |

Table 4: Locations

| <u>DNBR</u> | <u>DLOCATION</u> |
|-------------|------------------|
| 10 | Edmonton |
| 10 | Ottawa |
| 11 | Athabasca |
| 12 | Ottawa |
| 12 | Montreal |

- Create these MIS COMPANY tables in your home database.
- Use SQL queries to populate the tables with the specified data.
- It is preferable to use triggers to enforce an “on updates cascade” policy for foreign keys. In other words, if X is an attribute in Table 1, and is a foreign key in Table 2 and Table 3, then any change to an X value in Table 1 will result in all X values equal to the old value being updated accordingly in Table 2 and Table 3. Write a trigger to handle this “on updates cascade” for your tables. Your solution should be simple and correct.
- The company administration has decided to change the research department number from “12” to “14”. Write an SQL query that performs this update. Your query should make use of your trigger (from c) to ensure database integrity.
- Write an SQL statement to retrieve the number of people working in each of the departments responsible for mobile technology projects.

- f. Retrieve the department name and the number of female employees working for each department whose average salary is more than 27K.
- g. For some strategic decisions, the president of the company needs summary data about the departments. For each department, he needs to know the number of employees working on mobile technology projects, as well as their total and average salaries. Does the following view answer the president's request? If not, write the correct view that will satisfy the president's request.

```
CREATE VIEW DEPT_SUMMARY (D, C, TOTAL_S, AVERAGE_S)
AS SELECT DNO, COUNT (*), SUM (SALARY), AVG (SALARY)
FROM EMPLOYEE, PROJECT
WHERE DNO = DNUM and PNAME like 'Mobile%'
GROUP BY DNO;
```

- h. Which of the following queries and updates would be allowed by this view (from g)? If a query or update is allowed, what would the corresponding query or update on the base relations look like? Give its result when applied to the database.

```
Q1. SELECT *
FROM DEPT_SUMMARY;
Q2. SELECT D, C
FROM DEPT_SUMMARY
WHERE TOTAL_S > 100000;
Q3. SELECT D, AVERAGE_S
FROM DEPT_SUMMARY
WHERE C > (SELECT C FROM DEPT_SUMMARY WHERE D=4);
Q4. UPDATE DEPT_SUMMARY
SET D=3
WHERE D=4;
Q5. DELETE FROM DEPT_SUMMARY
WHERE C > 4;
```

Section 1 SQL as a Data Definition Language (DDL)

This section introduces the SQL commands for creating tables or indexes, for altering tables, or for deleting tables in a database.

Learning Outcomes

When you have completed this section, you should be able to

- describe the SQL environment, and define certain concepts related to SQL, such as catalog, schema, DDL, DML, and DCL;
- use SQL commands to define a database;
- list and explain the seven steps to be followed when creating tables in databases;
- use SQL commands to create, alter, or delete tables in a database.

Readings

- Read the sections “Origins of the SQL standard,” “The SQL environment,” and “Defining a database in SQL” from Chapter 6: “Introduction to SQL.”

Tasks

Practice the following questions and exercises from Chapter 6. You may want to use the Unit 5 Workspace for the tasks in this and the following sections.

- Review Questions: 3, 7, and 8
- Problems and Exercises: 1, 3, and 4

Key Terms

You may want to use the student-created Glossary of Key Terms on the course home page to define the following:

data definition language, data manipulation language, data control language, referential integrity

Section 2 SQL as a Data Manipulation Language (DML)

This section introduces the SQL commands for inserting new data in tables, changing existing tables, retrieving data from tables, and defining views.

Learning Outcomes

When you have completed this section, you should be able to

- apply SQL commands to insert one or more rows into an existing table;
- apply SQL commands to change data in one or more rows of an existing table;
- apply SQL commands to delete one or more rows from an existing table;
- apply SQL commands to retrieve data from tables;
- practice the use of distinct values, ORDER BY, GROUP BY, and HAVING;
- use and define views.

Readings

- Read the sections “Inserting, updating, and deleting data,” “Internal schema definition in RDBMSs,” and “Processing single tables” from Chapter 6: “Introduction to SQL.”

Tasks

Practice the following questions and exercises from Chapter 6.

- Review Questions: 13, 16, 17, 19, and 24
- Problems and Exercises: 5, 7, 8, 9, 12, 13, and 15

Key Terms

You may want to use the student-created Glossary of Key Terms on the course home page to define the following:

scalar aggregate, vector aggregate, virtual table, dynamic view, materialized view

Section 3 Advanced SQL and Data Control Commands

This section examines multiple table queries and data dictionary facilities. We will also discuss the commands used to control the transactions that maintain data integrity and data consistency in a database.

Learning Outcomes

When you have completed this section, you should be able to

- describe the process of querying data from multiple tables;
- illustrate the use of equi-join, natural-join, outer-join, union-join, subqueries, and derived tables;
- explain how data integrity and data consistency are maintained in a database;
- explain the programming extensions of SQL, such as UDP, OLAP, SQL/PSM, Triggers and Routines, Embedded SQL, and Dynamic SQL.

Readings

- Read all sections of Chapter 7: “Advanced SQL.”

Tasks

Practice the following questions and exercises from Chapter 7.

- Review Questions: 3, 4, 7, 10, and 15
- Problems and Exercises: 1, 3, 5, 6, and 13

Key Terms

You may want to use the student-created Glossary of Key Terms on the course home page to define the following:

subqueries, correlated subquery, derived tables, data dictionary, user-defined data type, persistent stored modules, triggers, embedded SQL, dynamic SQL

Unit 6 Introduction to Two-Tier and Three-Tier Architectures, and the Internet Database Environment

Commentary

This unit discusses client/server architectures, applications, and middleware, and describes open database connectivity (ODBC) and Java database connectivity (JDBC). This unit also provides an overview of the main concepts related to Internet databases, and describes the key components of Web applications. We will also discuss XQuery, and explain how XML has contributed to the spread of Web services and service-oriented architecture.

Activities

Assignment 2

Complete [Assignment 2](#), and submit it to your tutor for evaluation and feedback.

Unit 6 Introduction to Two-Tier and Three-Tier Architectures, and the Internet Database Environment

Section 1 Client/Server Architectures and Database Connection to Web Applications

This section explains the client/server model, and identifies the different architectures used to organize data storage and manage application processing. This section also examines the problems related to the distribution of data and processing, and describes how to connect databases to Web applications in two-tier and three-tier architectures.

Learning Outcomes

When you have completed this section, you should be able to

- describe the different client/server architectures: file server architecture, database server (two-tier) architecture, and three-tier/n-tier architecture;
- introduce different models used to implement client/server applications;
- discuss the issues related to the distribution of processing and data;
- describe how to connect to databases in two-tier and three-tier applications.

Readings

- Read the sections “Client/server architectures,” “Databases in a two-tier architecture,” “Three-tier architectures,” “Web application components,” “Databases in three-tier applications,” and “Key considerations in three-tier applications” from Chapter 8: “Database Application Development.”

Tasks

Practice the following questions and exercises from Chapter 8. You may want to use the Unit 6 Workspace for the tasks in this and the following sections.

- Review Questions: 3, 7, 8, and 9
- Problems and Exercises: 2, 4, 5, and 6

Unit 6 Introduction to Two-Tier and Three-Tier Architectures, and the Internet Database Environment

Section 2 XML and Data Exchange on the Internet

This section describes the role of XML as a data exchange standard, and the storage of XML documents in databases. This section also examines the use of XQuery and various XML-based technologies, such as UDDI, WSDL, and SOAP.

Learning Outcomes

When you have completed this section, you should be able to

- explain how to use XML to standardize data exchanges in Internet applications;
- explain how to use XQuery to retrieve XML documents in databases;
- discuss service-oriented architecture (SOA) and Web services.

Readings

- Read the section “Extensible Markup Language (XML)” from Chapter 8: “Database Application Development.”

Tasks

Practice the following questions and exercises from Chapter 8.

- Review Questions: 12, 13, 15, and 16
- Problems and Exercises: 8, 9, and 10

Unit 7 Introduction to Data Warehousing

Commentary

This unit provides an overview of data warehousing, and introduces data warehousing design, management, and architectures. This unit will also identify the problems related to extracting data from operational systems, and adding it to a data warehouse. The star schema for data marts is described, and some OLAP tools are presented.

Unit 7 Introduction to Data Warehousing

Section 1 Basic Concepts of Data Warehousing

This section describes the basic concepts of data warehousing, and defines related key terms.

Learning Outcomes

When you have completed this section, you should be able to

- define the basic concept of data warehousing;
- explain the main reasons behind the need for data warehousing.

Readings

- Read the section “Basic concepts of data warehousing” from Chapter 9: “Data Warehousing.”

Tasks

Practice the following questions from Chapter 9. You may want to use the Unit 7 Workspace for the tasks in this and the following sections.

- Review Questions: 3 and 4

Key Terms

You may want to use the student-created Glossary of Key Terms on the course home page to define the following:

data warehouse, operational system, informational system

Unit 7 Introduction to Data Warehousing

Section 2 Data Warehouse Architectures and OLAP Tools

This section describes different architectures used in data warehousing systems. This section also explains the extract/transfer/load process used to transfer operational data to the enterprise data warehouse or data store. It will also describe the design and generation of data marts, and the use of OLAP tools.

Learning Outcomes

When you have completed this section, you should be able to

- describe the main architectures used to build data warehouse systems;
- list and explain the basic characteristics and types of data in data warehousing systems;
- explain the main concepts involved in the design of data marts;
- describe the process used to generate derived data;
- describe the star schema, as used in data marts;
- present a variety of tools and techniques that are used to query, analyze, and visualize the data stored in data warehouses and data marts.

Readings

- Read the sections “Data warehouse architectures,” “Some characteristics of data warehouse data,” “The derived data layer,” “Column databases: A new alternative for data warehouses,” “The user interface,” “SQL OLAP querying,” and “Data visualization” from Chapter 9: “Data Warehousing.”

Tasks

Practice the following questions and exercises from Chapter 9.

- Review Questions: 5,6, 10, 14, 16, 17, 22, and 23
- Problems and Exercises: 3, 5, 6, and 7

Key Terms

You may want to use the student-created Glossary of Key Terms on the course home page to define the following:

data mart, enterprise data warehouse, operational data store, logical data mart, real-time data warehouse, reconciled data, derived data, transient data, periodic data, star schema, grain, snowflake schema, OLAP, ROLAP, MOLAP

Unit 8 Data Quality and Database Administration

Commentary

This unit describes the main concepts related to data quality and data integration, including the process of managing data quality in databases, and the process for data integration in data warehouses. This unit also describes the role of the data and database administrators. Data administration involves interfacing with upper management, employees, staff, users, programmers, and working with hardware and software packages. We will also discuss different techniques for data security, database backup and recovery, and database tuning.

Activities

Assignment 3

Complete [Assignment 3](#), and submit it to your tutor for evaluation and feedback.

Unit 8 Data Quality and Database Administration

Section 1 Data Quality and Integration

This section identifies the characteristics of data quality, and defines the process used to manage data quality in databases. We will also discuss general approaches to data integration, and describe the ‘extract, transform, load’ (ETL) process for data integration in data warehouses.

Learning Outcomes

When you have completed this section, you should be able to

- define the characteristics of data quality;
- identify situations of poor data quality, and measures for improving them;
- describe the following data integration approaches: data consolidation, data federation, and data propagation;
- discuss, in detail, the ETL process for data integration used in the reconciled data layer of data warehousing;
- discuss the data transformation functions.

Readings

- Read all sections of Chapter 10: “Data Quality and Integration.”

Tasks

Practice the following questions from Chapter 10. You may want to use the Unit 8 Workspace for the tasks in this and the following sections.

- Review Questions: 4, 7, 10, 15, 18, and 20

Key Terms

You may want to use the student-created Glossary of Key Terms on the course home page to define the following:

data governance, master data management, data consolidation, data federation, data propagation, data transformation

Unit 8 Data Quality and Database Administration

Section 2 Data Administration and Database Administration

This section discusses the main activities performed by database administrators. Because a database is a shared resource in an organization, users' needs, interests, and requirements may conflict. Therefore, a need arises for a person (or a group of people) to resolve these conflicts. The database administrator performs some managerial activities (e.g., setting standards) and some technical activities (e.g., developing security and recovery procedures).

Learning Outcomes

When you have completed this section, you should be able to

- discuss the need for organizations to have both data administrator(s) and database administrator(s);
- list and discuss the roles of data administrators and database administrators;
- describe the threats addressed by data security;
- discuss the characteristics of most data security plans;
- discuss the Sarbanes-Oxley Act (SOX) for the control and security of databases and applications;
- list and discuss the backup and recovery facilities provided by database management systems;
- discuss the types of database failures and the process for disaster recovery;
- outline different mechanisms used for concurrency control;
- list and discuss the core functions supported by repository engines.

Readings

- Read all sections of Chapter 11: “Data and Database Administration.”

Tasks

Practice the following questions and exercises from Chapter 11.

- Review Questions: 7, 8, 12, 14, 16, 19, 25, and 27
- Problems and Exercises: 2, 3, 4, 9, 11, and 12

Key Terms

You may want to use the student-created Glossary of Key Terms on the course home page to define the following:

data administration, database administration, database security, authorization rules, encryption, database recovery, backup facilities, journalizing facilities, backward recovery, forward recovery, concurrency control, locking mechanism, versioning, data dictionary, information repository, data archiving

Unit 9 Overview of Object-Oriented Databases

Commentary

The object-oriented model (OOM), like the E-R model discussed in Unit 2, is used to represent real-life situations and user requirements. The OOM can also be transformed and used in database design. In the OOM, we first recognize the objects of concern to the situation, and then look at their characteristics and behaviours. There are two main solutions for implementing Object databases: either using pure object-oriented database systems, such as object store, db4o, and OrientDB; or using relational database systems to provide object persistence. In this unit, we will review the latter solution as it is widely used in the database development market.

Activities

Project

Complete the Project, and submit it to your tutor for evaluation and feedback.

Practice Final Examination

Complete the Practice Final Examination.

Unit 9 Overview of Object-Oriented Databases

Section 1 Object-Oriented Data Modeling

This section provides an overview of the main concepts used to create object data models. As with the entities in the EER model discussed in Unit 2, the object-oriented model (OOM) is composed of objects that encapsulate data and their behaviours. These objects may be sub-classed or super-classed. The OOM has many advantages over the E-R and EER models, including its ability to represent complex data structures in multimedia applications. This section will also review the unified modeling language (UML) class diagram notation; and discuss related concepts, such as generalization, aggregation, and composition.

Learning Outcomes

When you have completed this section, you should be able to

- list the advantages of the object-oriented model (OOM);
- list the different phases of the development cycle of object-oriented systems;
- develop class diagrams using the unified modeling language (UML);
- represent association relationships of different degrees;
- explain when it is useful to model associations as association classes;
- discuss when and how to use and model generalization, inheritance, and aggregation;
- graphically represent object-oriented data models in a variety of cases.

Readings

- Read all sections of Chapter 13: “Overview: Object-Oriented Data Modeling.”

Supplemental reading

- Download and read the full version of Chapter 13: “Object-Oriented Data Modeling” from the textbook Web site.

Tasks

Practice the following questions and exercises. You may want to use the Unit 9 Workspace for the tasks in this and the following sections.

Review Questions

1. Compare and contrast the object-oriented model with the enhanced E-R (EER) model.

2. Give an example of generalization. Your example should include at least one superclass, three subclasses, and one attribute and one operation for each class. Indicate the discriminator, and specify the semantic constraints among the subclasses.
3. Give an example of aggregation. Your example should include at least one aggregate object and three component objects. Specify the multiplicities at each end of the aggregation relationships.

Problems and Exercises

1. A student, whose attributes include student_Name, address, phone number, and age, may engage in multiple activities. The university keeps track of the number of years a given student has participated in a specific activity. At the end of each academic year, the university mails an activity report to the student showing her/his participation in various activities. Draw a class diagram for this situation.
2. Draw a class diagram showing the relevant classes, attributes, operations, and relationships for the following situation. The company Emerging Energy wishes to create a database with the following classes and attributes: **Customer**, with attributes customer_ID, name, address (street, city, province, postal code), and phone number; **Location**, with attributes location_ID, address (street, city, province, postal code), and Type (business, residential); and **Rate**, with attributes rate_class and rate_per_kWh.

After interviewing the owners, you have come up with the following business rules. Customers can have one or more locations, and each location can have one or more rate—depending on the time of day.

3. You've been hired to do the initial design of an object-oriented database for a local medical doctor's office. Assume you have conducted the initial interview with office personnel, and you have gathered the data given below. You may need to make some logical assumptions to clarify this information. Specify key (and basic) attributes of each entity type, and mapping/structural constraints on each side of the relationship type, such as cardinality and participation. Be particularly careful to list any attributes that are mentioned in these notes. In your design, note any unspecified requirements and the appropriate assumptions used to make the specification complete.

This office is a partnership between two doctors. The office maintains patient information: SIN, name, address, home phone number, work phone number, employer name, employer address, pertinent medical history (illnesses, allergies, medical allergies to drugs, blood type, etc.), current medical history with this office (i.e., a record of all previous visits to this office). The office filing system is based on patient appointments, and therefore a schedule of appointments is maintained. The office files medical insurance claims for patients, and has contracts with several insurance companies. The office processes accounts receivable and payable. For each patient, the office maintains information about the party responsible for the patient with respect to their medical bills and medical permission for procedures. Each medical procedure has a unique medical code, which is used to file insurance claims, and to determine the amount of the bill. Patients may have multiple procedures performed in one visit. Each of the two doctors has hospital privileges at (possibly different and multiple) hospitals. Each doctor has a state medical

license and a local business license. The office files monthly reports with the state concerning the diseases that were treated. The office has an answering service that provides information on all messages after hours. Your database will also need to maintain information on the office staff and professionals. Draw a class diagram for the above situation.

Key Terms

You may want to use the student-created Glossary of Key Terms on the course home page to define the following:

object, state, behaviour, object class, class diagram, object diagram, operation, encapsulation, constructor operation, query operation, update operation, association, association role, multiplicity, association class, abstract class, concrete class, class-scope attribute, abstract operation, method, polymorphism, overriding, multiple classification, aggregation, composition

Section 2 Implementing Object Persistence using Relational Databases

This section discusses the integration between the object-oriented application development model and databases. It will also discuss the common approaches and mechanisms used to provide object persistence. This section will explore the Hibernate framework in greater detail through examples, and will present the Hibernate Query Language (HQL).

Learning Outcomes

When you have completed this section, you should be able to

- explain object-relational mapping, and distinguish between an object-oriented and a relational representation of data;
- explain the fundamental mismatch between the object-oriented paradigm and the relational model;
- discuss the different approaches used to address the object-relational mismatch;
- apply the mapping of object-oriented structures to a relational database;
- use the Hibernate Query Language (HQL) to formulate various types of queries.

Readings

- Read all sections of Chapter 14: “Overview: Using Relational Databases to Provide Object Persistence.”

Supplemental reading

- Download and read the full version of Chapter 14: “Using Relational Databases to Provide Object Persistence“ from the textbook Web site.

Tasks

Practice the following questions and exercises from the supplemental reading (i.e., the online version of Chapter 14).

- Review Questions: 4, 5, 6, 7, 11, 18, and 23
- Problems and Exercises: 1, 2, 3, and 4

Key Terms

You may want to use the student-created Glossary of Key Terms on the course home page to define the following:

persistence, serialization, object-relational mapping, object identity, assessor method, entity class, value type, HQL

