Study Guide

For

Database Systems Course

Department of Computer Science

Faculty of Informatics

ቅድስት ማርያም ዩኒቨርስቲ ኮሌጅ
St. Mary’s University College

Prepared by
Worku Alemu

January 2012

Part I Course Title: **Fundamentals of Database Systems**

Course description

The course covers basic topics related to file handling mechanisms like manual, File-based and database. The various kinds of database architectures, models and issues on database design and implementation will be introduced.

Course objectives

At the end of this course students will be able to:

- Define database terminologies.
- Identify the difference between File-based approaches versus Database approach.
- Know the advantage and disadvantage of database.
- Explain the function of DBMS
- Discuss the three-level database architecture.
- Understand the purpose and importance of conceptual modeling.
- Identify the role of DDL and DML.
- Describe the relational model and properties of database relations.
- Perform conceptual, logical and physical database design.
- Write SQL commands to retrieve, insert, update and delete data and create database objects.

Chapter 1 Introduction to Databases

General objectives

At the end of this chapter students will be able to

- *Discuss the advantages and limitations of the different file handling mechanisms*
- *Explain the components of DBMS*
- *Write the roles of people in database environment*

Specific objectives

At the end of this chapter students will be able to

- Explain some common uses of database systems.
- Discuss the characteristics of file-based systems.
- Identify the problems with the file-based approach.
- Define the meaning of the term 'database'.
- Define the meaning of the term 'database management system' (DBMS).
- Explain the typical functions of a DBMS.
- Identify the major components of the DBMS environment.
- List the personnel involved in the DBMS environment.
- Explain the history of the development of DBMSs.
- Explain the advantages and disadvantages of DBMSs.

Chapter Summary

- The Database Management System (DBMS) is now the underlying framework of the information system and has fundamentally changed the way that many organizations operate. The database system remains a very active research area and many significant problems have still to be satisfactorily resolved.
- The predecessor to the DBMS was the file-based system, which is a collection of application programs that perform services for the end-users, usually the production of reports. Each program defines and manages its own data. Although the file-based system was a great

improvement on the manual filing system, it still has significant problems, mainly the amount of data redundancy present and program–data dependence.

- The database approach emerged to resolve the problems with the file-based approach. A database is a shared collection of logically related data, and a description of this data, designed to meet the information needs of an organization. A DBMS is a software system that enables users to define, create, maintain, and control access to the database. An application program is a computer program that interacts with the database by issuing an appropriate request (typically an SQL statement) to the DBMS. The more inclusive term database system is used to define a collection of application programs that interact with the database along with the DBMS and database itself.
- All access to the database is through the DBMS. The DBMS provides a Data Definition Language (DDL), which allows users to define the database, and a Data Manipulation Language (DML), which allows users to insert, update, delete, and retrieve data from the database.
- The DBMS provides controlled access to the database. It provides security, integrity, concurrency and recovery control, and a user-accessible catalog. It also provides a view mechanism to simplify the data that users have to deal with.
- The DBMS environment consists of hardware (the computer), software (the DBMS, operating system, and applications programs), data, procedures, and people. The people include data and database administrators, database designers, application developers, and end-users.
- Some advantages of the database approach include control of data redundancy, data consistency, sharing of data, and improved security and integrity. Some disadvantages include complexity, cost, reduced performance, and higher impact of a failure.

Review Questions

1.1 Discuss each of the following terms:

(a) Data

(b) Database

(c) Database management system

(d) Database application program

(e) Data independence

1.2 Describe the approach taken to the handling of data in the early file-based systems. Discuss the disadvantages of this approach.

1.3 Describe the main characteristics of the database approach and contrast it with the file-based approach.

1.4 Describe the five components of the DBMS environment and discuss how they relate to each other.

1.5 Discuss the roles of the following personnel in the database environment:

(a) Data administrator

(b) Database administrator

(c) Logical database designer

(d) Physical database designer

(e) Application developer

(f) end-users.

1.6 Discuss the advantages and disadvantages of DBMSs.

Required Readings

Database System Concepts (Silberschatz 5th Ed): Chapter 1 (1.1 to 1.6)

Database Processing (David M): Chapter 1

Database Systems for Management (James F): Chapter 1

Database Management System (Ramakrishnan): Chapter 1

Fundamental of Relational Database Management System (Sumathi): Chapter 1 (1.1)

Database System Concepts (Silberschatz 5th Ed): Chapter 1 (1.11)

Database Systems for Management (James F): Chapter 2

Fundamental of Relational Database Management System (Sumathi): Chapter 1 (1.12 to 1.14)

Modern Database Management (Jeffrey): Chapter 2

Chapter 2 Database Design

General objectives

At the end of this chapter students will be able to

- *Explain the aims of different phases database design*
- *Explain the different types of database modeling*

- *Write the advantage and limitations of each model*

Specific objectives

In this chapter you will learn:

- The purpose and origin of the three-level database architecture.
- The contents of the external, conceptual, and internal levels.
- The purpose of the external/conceptual and the conceptual/internal mappings.
- The meaning of logical and physical data independence.
- The distinction between a Data Definition Language (DDL) and a Data Manipulation Language (DML).
- A classification of data models.
- The purpose and importance of conceptual modeling.
- The typical functions and services a DBMS should provide.
- The function and importance of the system catalog.
- The software components of a DBMS.
- The meaning of the client–server architecture and the advantages of this type of architecture for a DBMS.
- Discuss types of Database Modeling i.e.
    - ✓ Hierarchical database models
    - ✓ Network database model
    - ✓ Relational model

Chapter Summary

- The ANSI-SPARC database architecture uses three levels of abstraction: external, conceptual, and internal. The external level consists of the users' views of the database. The conceptual level is the community view of the database. It specifies the information content of the entire database, independent of storage considerations.

  The conceptual level represents all entities, their attributes, and their relationships, as well as the constraints on the data, and security and integrity information. The internal level is the computer's view of the database. It specifies how data is represented, how records are sequenced, what indexes and pointers exist, and so on.

- The external/conceptual mapping transforms requests and results between the external and conceptual levels. The conceptual/internal mapping transforms requests and results between the conceptual and internal levels.
- A database schema is a description of the database structure. Data independence makes each level immune to changes to lower levels. Logical data independence refers to the immunity of the external schemas to changes in the conceptual schema. Physical data independence refers to the immunity of the conceptual schema to changes in the internal schema.
- A data sublanguage consists of two parts: a Data Definition Language (DDL) and a Data Manipulation Language (DML). The DDL is used to specify the database schema and the DML is used to both read and update the database. The part of a DML that involves data retrieval is called a query language.
- A data model is a collection of concepts that can be used to describe a set of data, the operations to manipulate the data, and a set of integrity constraints for the data. They fall into three broad categories: object-based data models, record-based data models, and physical data models. The first two are used to describe data at the conceptual and external levels; the latter is used to describe data at the internal level.
- Object-based data models include the Entity–Relationship, semantic, functional, and object-oriented models. Record-based data models include the relational, network, and hierarchical models.
- Conceptual modeling is the process of constructing a detailed architecture for a database that is independent of implementation details, such as the target DBMS, application programs, programming languages, or any other physical considerations. The design of the conceptual schema is critical to the overall success of the system. It is worth spending the time and energy necessary to produce the best possible conceptual design.
- Functions and services of a multi-user DBMS include data storage, retrieval, and update; a user-accessible catalog; transaction support; concurrency control and recovery services; authorization services; support for data communication; integrity services; services to promote data independence; utility services.
- The system catalog is one of the fundamental components of a DBMS. It contains 'data about the data', or metadata. The catalog should be accessible to users. The Information Resource

Dictionary System is an ISO standard that defines a set of access methods for a data dictionary. This allows dictionaries to be shared and transferred from one system to another.

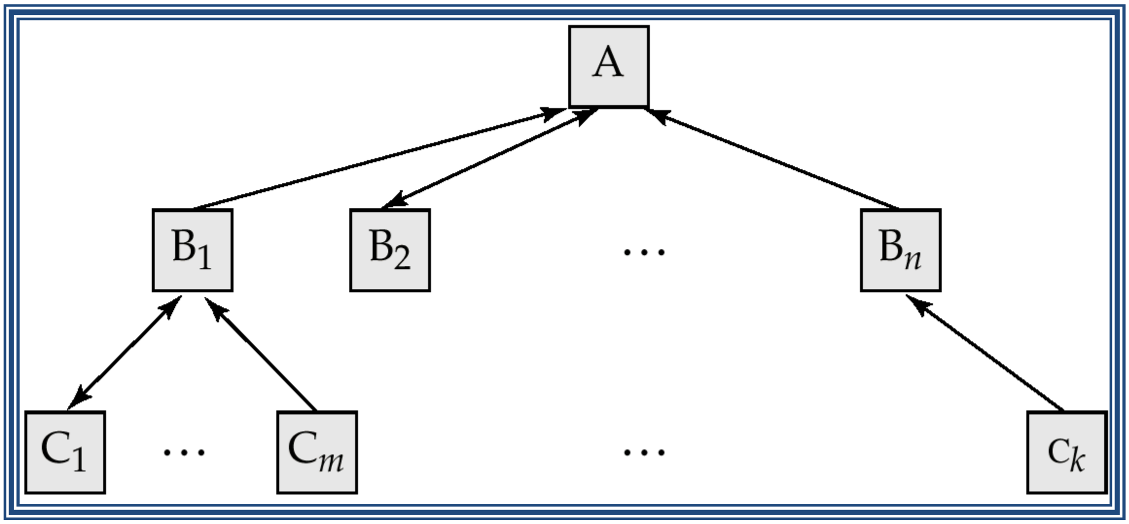
- Client–server architecture refers to the way in which software components interact. There is a client process that requires some resource, and a server that provides the resource. In the two-tier model, the client handles the user interface and business processing logic and the server handles the database functionality. In the Web environment, the traditional two-tier model has been replaced by a three-tier model, consisting of a user interface layer (the client), a business logic and data processing layer (the application server), and a DBMS (the database server), distributed over different machines.
- A data model is a description of the way that data is stored in a database. Data model helps to understand the relationship between entities and to create the most effective structure to hold data.
- The main *purpose* of Data Model is to represent the data in an understandable way Categories of data models include: Object-based, Record-based, and Physical

Record-based Data Models: Consist of a number of fixed format records. Each record type defines a fixed number of fields; each field is typically of a fixed length. These are Hierarchical Data Model, Network Data Model, Relational Data Model

a. Hierarchical Model

- The simplest data model
- Record type is referred to as node or segment
- The top node is the root node
- Nodes are arranged in a hierarchical structure as sort of upside-down tree
- A parent node can have more than one child node
- A child node can only have one parent node
- The relationship between parent and child is one-to-many
- Relation is established by creating physical link between stored records (each is stored with a predefined access path to other records)
- To add new record type or relationship, the database must be redefined and then stored in a new form.

General Structure



- A parent *may* have an arrow pointing to a child, but a child *must* have an arrow pointing to its parent

b. Network Model

- Allows record types to have more than one parent unlike hierarchical model
- A network data models sees records as set members
- Each set has an owner and one or more members
- Allow no many to many relationship between entities
- Like hierarchical model network model is a collection of physically linked records.
- Allow member records to have more than one owner

c. Relational Data Model

  - Relation: Two dimensional table
  - Stores information or data in the form of tables (rows and columns)
  - A row of the table is called tuple equivalent to record
  - A column of a table is called attribute equivalent to fields
  - Data value is the value of the Attribute
- Records are related by the data stored jointly in the fields of records in two tables or files. The related tables contain information that creates the relation
- The tables seem to be independent but are related somehow.
- No physical consideration of the storage is required by the user

Review questions

2.1 Discuss the concept of data independence and explain its importance in a database environment.

2.2 To address the issue of data independence, the ANSI-SPARC three-level architecture was proposed. Compare and contrast the three levels of this model.

2.3 What is a data model? Discuss the main types of data model.

2.4 Discuss the function and importance of conceptual modeling.

2.5 Describe the types of facility you would expect to be provided in a multi-user DBMS.

2.6 Discuss the function and importance of the system catalog.

2.7 Describe the main components in a DBMS and suggest which components are responsible for each facility.

2.8 What is meant by the term 'client–server architecture' and what are the advantages of this approach?

2.9 Compare the client–server architecture with two other architectures.

Required Readings

Database System Concepts (Silberschatz 5th Ed): Chapters 2 (2.1), 7

Database Processing (David M): Chapters 3,4,5,6

Database Systems for Management (James F): Chapters 3, 4

Data Modeling Essentials (Graemec): Chapter 2

Database Modeling and Design (Resource): Chapters 2,4,5,6

Fundamental of Relational Database Management System (Sumathi): Chapters 2, 3, 6

Chapter 3. The Relational Model

General objectives: At the end of this chapter students will be able to

- *Explain the aims of selecting relational database model*
- *Describe the different types of relational keys and Relational constraints*
- *Discuss about Relational languages and views*

Specific Objectives

In this chapter you will learn:

- The origins of the relational model.

- The terminology of the relational model.
- How tables are used to represent data.
- The connection between mathematical relations and relations in the relational model.
- Properties of database relations.
- How to identify candidate, primary, alternate, and foreign keys.
- The meaning of entity integrity and referential integrity.
- The purpose and advantages of views in relational systems.

Chapter Summary

- The Relational Database Management System (RDBMS) has become the dominant data-processing software in use today. This software represents the second generation of DBMSs and is based on

  the relational data model proposed by E. F. Codd.
- A mathematical relation is a subset of the Cartesian product of two or more sets. In database terms, a relation is any subset of the Cartesian product of the domains of the attributes. A relation is normally written as a set of *n*-tuples, in which each element is chosen from the appropriate domain.
- Relations are physically represented as tables, with the rows corresponding to individual tuples and the columns to attributes.
- Properties of database relations are: each cell contains exactly one atomic value, attribute names are distinct, attribute values come from the same domain, attribute order is immaterial, tuple order is immaterial, and there are no duplicate tuples.
- The degree of a relation is the number of attributes, while the cardinality is the number of tuples. A unary relation has one attribute, a binary relation has two, a ternary relation has three, and an *n*-ary relation has *n* attributes.
- A superkey is an attribute, or set of attributes, that identifies tuples of a relation uniquely, while a candidate key is a minimal superkey. A primary key is the candidate key chosen for use in identification of tuples.A relation must always have a primary key. A foreign key is an attribute, or set of attributes, within one relation that is the candidate key of another relation.
- A null represents a value for an attribute that is unknown at the present time or is not applicable for this tuple.

- Entity integrity is a constraint that states that in a base relation no attribute of a primary key can be null. Referential integrity states that foreign key values must match a candidate key value of some tuple in the home relation or be wholly null. Apart from relational integrity, integrity constraints include, required data, domain, and multiplicity constraints; other integrity constraints are called general constraints.
- A view in the relational model is a virtual or derived relation that is dynamically created from the underlying base relation(s) when required. Views provide security and allow the designer to customize a user's model. Not all views are updatable.

Review Questions

3.1 Discuss each of the following concepts in the context of the relational data model:

(a) Relation, (b) Attribute, (c) Domain (d) Tuple, (e) Degree and cardinality.

3.2 Describe the relationship between mathematical relations and relations in the relational data model.

3.3 Discuss the properties of a relation.

3.4 Discuss the differences between the candidate keys and the primary key of a relation. Explain what is meant by a foreign key. How do foreign keys of relations relate to candidate keys? Give examples to illustrate your answer.

3.5 Define the two principal integrity rules for the relational model. Discuss why it is desirable to these rules.

3.6 What is a view? Discuss the difference between a view and a base relation.

Required Readings

Database System Concepts (Silberschatz 5th Ed): Chapter 2 (2.2 to 2.3)

Database Systems for Management (James F): Chapter 6

Database Management System (Ramakrishnan): Chapter 4

Chapter 4 Relational Algebra

General objectives

At the end of this chapter students will be able to

- Identify the different relational operators
- Write the results of different operations

- Write Queries in Relational Algebra

Specific Objectives

At the end of this chapter students will be able to

- Examine the two unary operations Selection and Projection.
- Examine the binary operations of the relational algebra, starting with the set operations of Union, Set difference, Intersection, and Cartesian product.
- Use various forms of Join operation, Theta join, Equijoin, Natural join, Outer join, Semi-join to write queries and write the results

Chapter Summary

- The relational algebra is a (high-level) procedural language: it can be used to tell the DBMS how to build a new relation from one or more relations in the database. The relational calculus is a non-procedural language: it can be used to formulate the definition of a relation in terms of one or more database relations. However, formally the relational algebra and relational calculus are equivalent to one another: for every expression in the algebra, there is an equivalent expression in the calculus (and vice versa).
- The relational calculus is used to measure the selective power of relational languages. A language that can be used to produce any relation that can be derived using the relational calculus is said to be relationally complete. Most relational query languages are relationally complete but have more expressive power than the relational algebra or relational calculus because of additional operations such as calculated, summary, and ordering functions.
- The five fundamental operations in relational algebra, *Selection*, *Projection*, *Cartesian product*, *Union*, and *Set difference*, perform most of the data retrieval operations that we are interested in. In addition, there are also the *Join*, *Intersection*, and *Division* operations, which can be expressed in terms of the five basic operations.

Review questions

4.1. Explain the term closure of relational operations.

4.2. Define the five basic relational algebra operations.

4.3. Discuss the differences between the five Join operations: Theta join, Equijoin, Natural join, Outer join, and Semi join. Give examples to illustrate your answer.

Required Readings

Database System Concepts (Silberschatz 5th Ed): Chapter 2

Fundamental of Relational Database Management System (Sumathi): Chapters 5

Chapter 5 SQL

***General objectives**: At the end of this chapter students will be able to*

- *Explain the aims and history and importance of SQL*
- *Explain the different types of* SQL commands

***Specific objectives**: At the end of this chapter students will be able to*

- *Write the different SQL commands to create tables, store and retrieve data*
- *Explain the objectives of creating views*
- *Create and remove views*
- *Explain the different Restrictions on views*
- *Write the different SQL commands to create tables, store and retrieve data*

Chapter Summary

- SQL is a non-procedural language, consisting of standard English words such as SELECT, INSERT,DELETE, that can be used by professionals and non-professionals alike. It is both the formal and *de facto*standard language for defining and manipulating relational databases.
- The SELECT statement is the most important statement in the language and is used to express a query.It combines the three fundamental relational algebra operations of *Selection, Projection,* and *Join*. Every SELECT statement produces a query result table consisting of one or more columns and zero or more rows.
- The SELECT clause identifies the columns and/or calculated data to appear in the result table. All column names that appear in the SELECT clause must have their corresponding tables or views listed in the FROM clause.
- The WHERE clause selects rows to be included in the result table by applying a search condition to the rows of the named table(s). The ORDER BY clause allows the result table to be sorted on the values in one or more columns. Each column can be sorted in ascending or descending order. If specified, the ORDER BY clause must be the last clause in the SELECT statement.

- SQL supports five aggregate functions (COUNT, SUM, AVG, MIN, and MAX) that take an entire column as an argument and compute a single value as the result. It is illegal to mix aggregate functions with column names in a SELECT clause, unless the GROUP BY clause is used.
- The GROUP BY clause allows summary information to be included in the result table. Rows that have the same value for one or more columns can be grouped together and treated as a unit for using the aggregate functions. In this case the aggregate functions take each group as an argument and compute a single value for each group as the result. The HAVING clause acts as a WHERE clause for groups, restricting the groups that appear in the final result table. However, unlike the WHERE clause, the HAVING clause can include aggregate functions.
- A subselect is a complete SELECT statement embedded in another query. A subselect may appear within the WHERE or HAVING clauses of an outer SELECT statement, where it is called a subquery or nested query. Conceptually, a subquery produces a temporary table whose contents can be accessed by the outer query. A subquery can be embedded in another subquery.
- There are three types of subquery: scalar, row, and table. A *scalar subquery* returns a single column and a single row; that is, a single value. In principle, a scalar subquery can be used whenever a single value is needed. A *row subquery* returns multiple columns, but again only a single row. A row subquery can be used whenever a row value constructor is needed, typically in predicates. A *table subquery* returns one or more columns and multiple rows. A table subquery can be used whenever a table is needed, for example, as an operand for the IN predicate. If the columns of the result table come from more than one table, a join must be used, by specifying more than one table in the FROM clause and typically including a WHERE clause to specify the join column(s).The ISO standard allows Outer joins to be defined. It also allows the set operations of *Union*, *Intersection*, and *Difference* to be used with the UNION, INTERSECT, and EXCEPT commands.
- As well as SELECT, the SQL DML includes the INSERT statement to insert a single row of data into a named table or to insert an arbitrary number of rows from one or more other tables using a sub-select; the UPDATE statement to update one or more values in a specified column or columns of a named table; the DELETE statement to delete one or more rows from a named table.

Review Questions

5.1 What are the two major components of SQL and what function do they serve?

5.2 What are the advantages and disadvantages of SQL?

5.3 Explain the function of each of the clauses in the SELECT statement. What restrictions are imposed on these clauses?

5.4 What restrictions apply to the use of the aggregate functions within the SELECT statement? How do nulls affect the aggregate functions?

5.5 Explain how the GROUP BY clause works. What is the difference between the WHERE and HAVING clauses?

5.6 What is the difference between a subquery and a join? Under what circumstances would you not be able to use a subquery?

Recommended readings

Database System Concepts (Silberschatz 5th Ed): Chapter 3

Database Processing (David M): Chapter 7

Database Systems for Management (James F): Chapter 5

Database Management System (Ramakrishnan): Chapter 1

Fundamental of Relational Database Management System (Sumathi): Chapter 4

Modern Database management (Jeffery): Chapter 7

Database System Concepts (Silberschatz 5th Ed): Chapter 4

Database Management System (Ramakrishnan): Chapter 5 (5.7)

Modern Database management (Jeffery): Chapter 8

Fundamental of Relational Database Management System (Sumathi): Chapter 4 (4.15)

Chapter 6 Integrity and Security

General objectives: At the end of this chapter students will be able to

- *Identify Integrity constraints that ensure that changes made to the database by authorized users do not result in a loss of data consistency.*
- *Explain integrity constraints guard against accidental damage to the database*.

Specific objectives: *At the end of this chapter students will be able to*

- *Identify the different types of entity integrities*

- *Explain the roles of integrity rules*
- *Apply integrity rules on relational database*
- *Identify the different types of database security mechanisms*
- *Explain data protection and Privacy laws*

Chapter Summary

- Database security is the mechanisms that protect the database against intentional or accidental threats.
- Database security is concerned with avoiding the following situations: theft and fraud, loss of confidentiality (secrecy), loss of privacy, loss of integrity, and loss of availability.
- A threat is any situation or event, whether intentional or accidental, that will adversely affect a system and consequently an organization.
- Computer-based security controls for the multi-user environment include: authorization, access controls, views, backup and recovery, integrity, encryption, and RAID technology.
- Authorization is the granting of a right or privilege that enables a subject to have legitimate access to a system or a system's object. Authentication is a mechanism that determines whether a user is who he or she claims to be.
- Most commercial DBMSs provide an approach called Discretionary Access Control (DAC), which manages privileges using SQL.The SQL standard supports DAC through the GRANT and REVOKE commands. Some commercial DBMSs also provide an approach to access control called Mandatory Access Control (MAC), which is based on system-wide policies that cannot be changed by individual users. In this approach each database object is assigned a *security class* and each user is assigned a *clearance* for a security class, and *rules* are imposed on reading and writing of database objects by users. The SQL standard does not include support for MAC.
- A view is the dynamic result of one or more relational operations operating on the base relations to produce another relation. A view is a virtual relation that does not actually exist in the database but is produced upon request by a particular user at the time of request. The view mechanism provides a powerful and flexible security mechanism by hiding parts of the database from certain users.

- Backup is the process of periodically taking a copy of the database and log file (and possibly programs) on to offline storage media. Journaling is the process of keeping and maintaining a log file (or journal) of all changes made to the database to enable recovery to be undertaken effectively in the event of a failure.
- Integrity constraints also contribute to maintaining a secure database system by preventing data from becoming invalid, and hence giving misleading or incorrect results.
- Encryption is the encoding of the data by a special algorithm that renders the data unreadable by any program without the decryption key.

Review Questions

6.1 Explain the purpose and scope of database security.

6.2 List the main types of threat that could affect a database system and for each describe the controls that you would use to counteract each of them.

6.3 Explain the following in terms of providing security for a database:

(a) Authorization; (b). Access controls; (c) . Views; (d). Backup and recovery; (e). integrity;

(f) Encryption;

Chapter 7 Transaction Management

General objectives: At the end of this chapter students will be able to

- *Identify the different types of recovery control*

Specific objectives: At the end of this chapter students will be able to

- *Explain the need for Recovery Control*
- *Explain the need for concurrency Control*

Chapter Summary

- Concurrency control is the process of managing simultaneous operations on the database without having them interfere with one another. Database recovery is the process of restoring the database to a correct state after a failure. Both protect the database from inconsistencies and data loss.

  A transaction is an action, or series of actions, carried out by a single user or application program, which accesses or changes the contents of the database. A transaction is a logical *unit* *of work* that takes the database from one consistent state to another. Transactions can terminate successfully (commit) or unsuccessfully (abort). Aborted transactions must be undone or rolled back. The transaction is also the *unit of concurrency* and the *unit of recovery*.

- A transaction should possess the four basic or so-called ACID, properties: atomicity, consistency, isolation, and durability. Atomicity and durability are the responsibility of the recovery subsystem; isolation and, to some extent, consistency are the responsibility of the concurrency control subsystem.

Concurrency control is needed when multiple users are allowed to access the database simultaneously. Without it, problems of *lost update*, *uncommitted dependency*, and *inconsistent analysis* can arise.

  - Serial execution means executing one transaction at a time, with no interleaving of operations. A schedule shows the sequence of the operations of transactions. A schedule is serializable if it produces the same results as some serial schedule.
  - Two methods that guarantee serializability are two-phase locking (2PL) and time stamping. Locks may be shared (read) or exclusive (write). In two-phase locking, a transaction acquires all its locks before releasing any. With timestamping, transactions are ordered in such a way that older transactions get priority in the event of conflict.

- Deadlock occurs when two or more transactions are waiting to access data the other transaction has locked. The only way to break deadlock once it has occurred is to abort one or more of the transactions.
- A tree may be used to represent the granularity of locks in a system that allows locking of data items of different sizes. When an item is locked, all its descendants are also locked. When a new transaction requests a lock, it is easy to check all the ancestors of the object to determine whether they are already locked. To show whether any of the node's descendants are locked, an intention lock is placed on all the ancestors of any node being locked.
- Some causes of failure are system crashes, media failures, application software errors, carelessness, natural physical disasters, and sabotage. These failures can result in the loss of main memory and/or the disk copy of the database. Recovery techniques minimize these effects.
- To facilitate recovery, one method is for the system to maintain a log file containing transaction records that identify the start/end of transactions and the before- and after-images of the write

operations. Using deferred updates, writes are done initially to the log only and the log records are used to perform actual updates to the database. If the system fails, it examines the log to determine which transactions it needs to redo, but there is no need to undo any writes. Using immediate updates, an update may be made to the database itself any time after a log record is written. The log can be used to undo and redo transactions in the event of failure.

- Checkpoints are used to improve database recovery. At a checkpoint, all modified buffer blocks, all log records, and a checkpoint record identifying all active transactions are written to disk. If a failure occurs, the checkpoint record identifies which transactions need to be redone.
- Advanced transaction models include nested transactions, sagas, multilevel transactions, dynamically restructuring transactions, and workflow models.

Review Questions

7.1 Explain what is meant by a transaction.

7.2 The consistency and reliability aspects of transactions are due to the 'ACIDity' properties of transactions. Discuss each of these properties and how they relate to the concurrency control and recovery mechanisms. Give examples to illustrate your answer.

7.3 Describe, with examples, the types of problem that can occur in a multi-user environment when concurrent access to the database is allowed.

7.4. Discuss how the concurrency control mechanism interacts with the transaction mechanism.

7.5 Explain the concepts of serial, non-serial, and serializable schedules. State the rules for equivalence of schedules.

7.6 Discuss the difference between conflict serializability and view serializability.

7.7 Discuss the types of problem that can occur with locking-based mechanisms for concurrency control and the actions that can be taken by a DBMS to prevent them.

7.8 Why would two-phase locking not be an appropriate concurrency control scheme for indexes?

7.9 What is a timestamp? How do time stamp based protocols for concurrency control differ from locking based protocols?

7.10 Describe the basic timestamp ordering protocol for concurrency control. What is Thomas's write rule and how does this affect the basic timestamp ordering protocol?

7.11 Describe how versions can be used to increase concurrency.

7.12 Discuss the difference between pessimistic and optimistic concurrency control.

7.13 Discuss the types of failure that may occur in a database environment. Explain why it is important for a multi-user DBMS to provide a recovery mechanism.

7.14 Discuss how the log file (or journal) is a fundamental feature in any recovery mechanism. Explain what is meant by forward and backward recovery and describe how the log file is used in forward and backward recovery. What is the significance of the write-ahead log protocol? How do checkpoints affect the recovery protocol?

7.15 Compare and contrast the deferred update and immediate update recovery protocols.

.

Required Readings

Database System Concepts (Silberschatz 5th Ed): Chapters 15, 16

Database Management System (Ramakrishnan): Chapters 16, 17

Fundamental of Relational Database Management System (Sumathi): Chapter 7

Part II Course Title: Advanced Database Systems

Chapter 8: Relational Database Design

Enhanced Entity–Relationship Modeling

General objectives:

At the end of this unit you will be able to: Explain the limitations of the basic concepts of the Entity–Relationship (ER) model and the requirements to represent more complex applications using additional data modeling concepts.

Specific objectives

At the end of this unit you will be able to:

- Identify the most useful additional data modeling concepts of the Enhanced Entity–Relationship (EER) model called specialization/generalization, aggregation, and composition.
- Identify a diagrammatic technique for displaying specialization/generalization, aggregation, and composition in an EER diagram using the Unified Modeling Language (UML).

Chapter Summary

- A super class is an entity type that includes one or more distinct sub groupings of its occurrences, which require to be represented in a data model. A subclass is a distinct sub grouping of occurrences of an entity type, which require to be represented in a data model.
- 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.
- There are two constraints that may apply to a specialization/generalization called participation constraints and disjoint constraints.
- A participation constraint determines whether every member in the super class must participate as a member of a subclass.
- A disjoint constraint describes the relationship between members of the subclasses and indicates whether it is possible for a member of a super class to be a member of one, or more than one, subclass.
- 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'.

Review Questions

8.1 Describe what a superclass and a subclass represent.

8.2 Describe the relationship between a super class and its subclass.

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

8.4 What are the main reasons for introducing the concepts of super classes and subclasses into an ER model?

8.5 Describe what a shared subclass represents and how this concept relates to multiple inheritances.

8.6 Describe and contrast the process of specialization with the process of generalization.

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

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

Required Readings

Database Systems Thomas Connoly Carolyn Besg Fourth edition Chapter 12

Chapter 9: Database implementation and Tools

General objective:

At the end of this chapter you will be able to : *E*xplain Data Design and Implementation in an Organization

Specific objectives:

At the end of this unit you will be able to:

- Describe the use of UML and its support for database design specifications
- Describe representing specialization and generalization in UML Class diagram.
- Describe UML based design tools
- Identify automated database design tools.

Chapter Summary

- Database design mainly involves the design of the database schema. The entity relationship (E-R) data model is a widely used data model for database design. It provides a convenient graphical representation to view data, relationships, and constraints.
- The model is intended primarily for the database-design process. It was developed to facilitate database design by allowing the specification of an enterprise schema. Such a schema represents the overall logical structure of the database. This overall structure can be expressed graphically by an E-R diagram.
- An entity is an object that exists in the real world and is distinguishable from other objects. We express the distinction by associating with each entity a set of attributes that describes the object.

- A relationship is an association among several entities. A relationship set is a collection of relationships of the same type, and an entity set is a collection of entities of the same type.
- A superkey of an entity set is a set of one or more attributes that, taken collectively, allows us to identify uniquely an entity in the entity set. We choose a minimal superkey for each entity set from among its superkeys; the minimal superkey is termed the entity set's primary key. Similarly, a relationship set is a set of one or more attributes that, taken collectively, allows us to identify uniquely a relationship in the relationship set. Likewise, we choose a minimal superkey for each relationship set from among its superkeys; this is the relationship set's primary key.
- Mapping cardinalities express the number of entities to which another entity can be associated via a relationship set.
- An entity set that does not is termed a weak entity set, strong entity set.
- Specialization and generalization define a containment relationship between a higher-Level entity set and one or more lower-level entity sets. Specialization is the result of taking a subset of a higher-level entity set to form a lower level entity set. Generalization is the result of taking the union of two or more have sufficient attributes to form a primary key
- Aggregation is an abstraction in which relationship sets (along with their associated entity sets) are treated as higher-level entity sets, and can participate in relationships.
- The various features of the E-R model offer the database designer numerous choices in how to best represent the enterprise being modeled. Concepts and objects may, in certain cases, be represented by entities, relationships, or attributes.
- A database design specified by an E,R diagram can be represented by a collection of relation schemas. For each entity set and for each relationship set in the database there is a unique relation schema hat is assigned the name of the corresponding entity set or relationship set. This forms the basis for deriving a relational database design from an E-R diagram.
- The Unified Modeling Language (UML) provides a graphical means of modeling various components of a software system. The class diagram component of UML is based on E-R diagrams. However, there are some differences between the two that one must beware of.

Review questions

9.1 Explain the distinctions among the terms primary key, candidate key, and superkey.

9.2 Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted.

9.3 Explain the difference between a weak and a strong entity set.

9.4 Define the concept of aggregation. Give two examples of where this concept is useful.

9.5 Design a generalization-specialization hierarchy for a motor vehicle sales company. The company sells motorcycles, passenger cars, vans, and buses. Justify your placement of attributes at each level of the hierarchy. Explain why they should not be placed at a higher or lower level.

9.6 Explain the distinction between total and partial constraints

Chapter 10: Advanced SQL in Oracle

General objectives:

At the end of this chapter you will be able to: Discuss Features and basic architecture, Database Design and Querying Tools, SQL Variations and extensions

Specific objectives: **At the end of this chapter you will be able to**

- Explain Storage and Indexing, Query Processing, evaluation and Optimization, Assertion and views, Cursors, triggers and stored procedures
- Identify Embedded SQL, dynamic SQL, and Advanced Features of SQL, System Catalog in Oracle

  Identify SQL Variations and Extensions, Transaction Management, Storage and Indexing ,Query Processing and evaluation and optimization

Chapter Summary

Database Structure and Space Management Overview

- An Oracle **database** is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information. A database server is the key to solving the problems of information management. In general, a **server** reliably manages a large amount of data in a multiuser environment so that many users can concurrently access the same data. All this is accomplished while delivering high performance. A database server also prevents unauthorized access and provides efficient solutions for failure recovery

Database Design and Querying Tools

- Oracle provides a variety of tools for database design, querying, report generation, and data analysis, including OLAP.

Database Design Tools

Most of Oracle's design tools are included in the Oracle Developer Suite. This is a suite of tools for various aspects of application development, including tools for forms development, data modeling, reporting, and querying. The suite supports the UML standard for development modeling.

- The major database design tool in the suite is Oracle Designer, which translates business logic and data flows into schema definitions and procedural scripts for application logic. It supports such modeling techniques as E-R diagrams, information engineering, and object analysis and design. Oracle Designer stores the design in Oracle Repository, which serves as a single point of metadata for the application.

Querying Tools

- Oracle provides tools for ad-hoc querying, report generation and data analysis, including OLAP.

Oracle Application Server Discoverer is a Web-based, ad-hoc query, reporting, analysis, and Web publishing tool for end-users and data analysts. It allows users to drill up and down on result sets, pivot data, and store calculations as reports that can be published in a variety of formats such as spreadsheets or HTML.

Variations and Extensions

Oracle supports all core SQL: 1999 features fully or partially, with some minor exceptions such as distinct data types. In addition, Oracle supports a large number of other language constructs, some of which conform with sel:1999, while others are oracle-specific in syntax or functionality.

Object-Relational Features

Oracle has extensive support for object-relational constructs, including:

- Object types single-inheritance model is supported for type hierarchies.
- Collection types. Oracle supports arrays, which are variable-length arrays, and nested tables.
- Object tables. These are used to store objects while providing a relational view of the attributes of the objects.
- Object views. These provide a virtual object table view of data stored in a regular relational table. They allow data to be accessed or viewed in an object oriented style even if the data are really stored in a traditional relational format.

- User-defined aggregate functions. These can be used in SQL statements in the same way as built-in functions such as sum and count.

Triggers

- Oracle provides several types of triggers and several options for when and how they are invoked. For triggers that execute on DML statements such as insert, update, and delete,
- Oracle supports row triggers and statement triggers. Row triggers execute once for every row that is affected (updated or deleted, for example) by the DML operation.
- Oracle also has triggers that execute on a variety of other events, like database start-up or shut down/ server error messages/ user logon or logoff, and DDL statements such as create, alter, and drop statements.

Storage and Indexing

- In Oracle parlance, a database consists of information stored in files and is accessed through an instance, which is a shared-memory area and a set of processes that interact with the data in the files.

Tables

- A standard table in Oracle is heap organized; that is, the storage location of a row in a table is not based on the values contained in the row, and is fixed when the row is inserted.
- Oracle supports nested tables; that is, a table can have a column whose data type is another table. The nested table is not stored in line in the parent table, but is stored in a separate table.
- Oracle supports temporary tables where the duration of the data is either the transaction in which the data are inserted or the user session. The data are private to the session and are automatically removed at the end of its duration.

Indices

Oracle supports several different types of indices. The most commonly used type is what Oracle (and several other vendors) call a B-tree index created on one or multiple columns.

Partitioning

Oracle supports various kinds of horizontal partitioning of tables and indices, and this feature plays a major role in Oracle's ability to support very large databases. The ability to partition a table or index has advantages in many areas

- Backup and recovery are easier and faster, since they can be done on individual partitions rather than on the table as a whole.
- Loading operations in a data warehousing environment are less intrusive:

Data can be added to a partition, and then the partition added to a table, which is an instantaneous operation. Likewise, dropping a partition with obsolete data from a table is very easy in a data warehouse that maintains a rolling window of historical data.

Query Processing and Optimization

- Oracle supports a large variety of processing techniques in its query-processing engine.

Some of the more important ones are described here briefly.

Execution Methods

Data can be accessed through a variety of access methods:

- Full table scan. The query processor scans the entire table by getting information about the blocks that make up the table from the extent map and scanning those blocks.
- Index fast full scan. The processor scans the extents in the same way as the table extent in a full table scan.

Parallel Execution

- Oracle allows the execution of a single SQL statement to be parallelized by dividing the work between multiple processes on a multiprocessor computer. This feature is especially useful for computationally intensive operations that would otherwise take an unacceptably long time to perform.

Concurrency control and Recovery

- Oracle supports concurrency-control and recovery techniques that provide a number of useful features.

Concurrency Control

Oracle's multi version concurrency control differs from the concurrency mechanisms used by most other database vendors. Read-only queries are given a read-consistent snapshot, which is a view of the database as it existed at a specific point in time, containing all updates that were committed by that point in time, and not containing any updates that were not committed at that point in time. Thus, read locks are not used and read-only queries do not interfere with other database activity in terms of locking.

Database Administration Tools

- Oracle provides users a number of tools and features for system management and application development. In the release Oracle10g, much emphasis was put on the concept of manageability that is, reducing the complexity of all aspects of creating and administering an Oracle database. This effort covered a wide variety of areas, including database creation, tuning, space management, storage management, backup and recovery, memory management, performance diagnostics, and workload management.

Database Resource Management

A database administrator needs to be able to control how the processing power of the hardware is divided among users all groups of users. Some groups may execute interactive queries where response time is critical; others may execute long-running reports that can be run as batch jobs in the background when the system load is low.

It is also important to be able to prevent a user from inadvertently submitting an extremely expensive ad-hoc query that will unduly delay other users.

Chapter 11: Query Processing and Evaluation

General objectives

At the end of this chapter you will be able to: Identify the different Measures of Query Cost

Specific objectives

At the end of this chapter you will be able to:

- Explain the Role of Relational Algebra and Relational Calculus in query optimization
- Explain Estimating Statistics of Expression

- Choice of Evaluation Plans
- Views and query processing
- Storage and query optimization

Chapter Summary

- The aims of **query processing** are to transform a query written in a high-level language, typically SQL, into a correct and efficient execution strategy expressed in a low-level language like the relational algebra, and to execute the strategy to retrieve the required data.
- As there are many equivalent transformations of the same high-level query, the DBMS has to choose the one that minimizes resource usage. This is the aim of **query optimization**. Since the problem is computationally intractable with a large number of relations, the strategy adopted is generally reduced to finding a near-optimum solution.
- There are two main techniques for query optimization, although the two strategies are usually combined in practice. The first technique uses **heuristic rules** that order the operations in a query. The other technique compares different strategies based on their relative costs, and selects the one that minimizes resource usage.
- Query processing can be divided into four main phases: decomposition (consisting of parsing and validation), optimization, code generation, and execution. The first three can be done either at compile time or at runtime.
- **Query decomposition** transforms a high-level query into a relational algebra query, and checks that the query is syntactically and semantically correct. The typical stages of query decomposition are analysis, normalization, semantic analysis, simplification, and query restructuring. A **relational algebra tree** can be used to provide an internal representation of a transformed query.
- **Query optimization** can apply transformation rules to convert one relational algebra expression into an equivalent expression that is known to be more efficient. Transformation rules include cascade of selection, commutativity of unary operations, commutativity of Theta join (and

Cartesian product), commutativity of unary operations and Theta join (and Cartesian product), and associativity of Theta join (and Cartesian product).

- **Heuristics rules** include performing Selection and Projection operations as early as possible; combining Cartesian product with a subsequent Selection whose predicate represents a join condition into a Join operation; using associativity of binary operations to rearrange leaf nodes so that leaf nodes with the most restrictive Selections are executed first.
- **Cost estimation** depends on statistical information held in the system catalog. Typical statistics include the cardinality of each base relation, the number of blocks required to store a relation, the number of distinct values for each attribute, the selection cardinality of each attribute, and the number of levels in each multilevel index.

Review Questions

11.1 What are the objectives of query processing?

11.2 How does query processing in relational systems differ from the processing of low-level query languages for network and hierarchical systems?

11.3 What are the typical phases of query processing?

11.4 What are the typical stages of query decomposition?

11.5 What is the difference between conjunctive and disjunctive normal form?

11.6 How would you check the semantic correctness of a query?

11.7 State the transformation rules that apply to:

(a) Selection operations, (b) Projection operations (c) Theta join operations.

11.8 State the heuristics that should be applied to improve the processing of a query.

11.9 What types of statistics should a DBMS hold to be able to derive estimates of relational algebra operations?

11.10 Under what circumstances would the system have to resort to a linear search when implementing a Selection operation?

Chapter 12: Distributed Databases

General objectives

At the end of this chapter you will be able to: Explain the need for distributed databases.

Specific objectives:

At the end of this chapter you will be able to:

- The differences between distributed database systems, distributed processing, and parallel database systems.
- Explain the advantages and disadvantages of distributed DBMSs.
- Identify the problems of heterogeneity in a distributed DBMS.
- Explain the functions that should be provided by a distributed DBMS.
- Explain architecture for a distributed DBMS.
- Identify the main issues associated with distributed database design, namely fragmentation, replication, and allocation.
- Explain how fragmentation should be carried out.
- The importance of allocation and replication in distributed databases.

Chapter Summary

- A distributed database is a logically interrelated collection of shared data (and a description of this data), physically distributed over a computer network. The DDBMS is the software that transparently manages the distributed database.
- A DDBMS is distinct from distributed processing, where a centralized DBMS is accessed over a network. It is also distinct from a parallel DBMS, which is a DBMS running across multiple processors and disks and which has been designed to evaluate operations in parallel, whenever possible, in order to improve performance.
- The advantages of a DDBMS are that it reflects the organizational structure; it makes remote data more shareable, it improves reliability, availability, and performance; it may be more economical, it provides for modular growth, facilitates integration, and helps organizations remain competitive. The major disadvantages are cost, complexity, lack of standards, and experience.
- A DDBMS may be classified as homogeneous or heterogeneous. In a homogeneous system, all sites use the same DBMS product. In a heterogeneous system, sites may run different DBMS products, which need not be based on the same underlying data model, and so the system may be composed of relational, network, hierarchical, and object-oriented DBMSs.

- A multi database system (MDBS) is a distributed DBMS in which each site maintains complete autonomy. An MDBS resides transparently on top of existing database and file systems, and presents a single database to its users. It maintains a global schema against which users issue queries and updates; an MDBS maintains only the global schema and the local DBMSs themselves maintain all user data.
- Communication takes place over a network, which may be a local area network (LAN) or a wide area network (WAN). LANs are intended for short distances and provide faster communication than WANs. A special case of the WAN is a metropolitan area network (MAN), which generally covers a city or suburb.
- As well as having the standard functionality expected of a centralized DBMS, a DDBMS will need extended communication services, extended system catalog, distributed query processing, and extended security, concurrency, and recovery services.
- A relation may be divided into a number of sub relations called fragments, which are allocated to one or more sites. Fragments may be replicated to provide improved availability and performance.
- There are two main types of fragmentation: horizontal and vertical. Horizontal fragments are subsets of tuples and vertical fragments are subsets of attributes. There are also two other types of fragmentation: mixed and derived a type of horizontal fragmentation where the fragmentation of one relation is based on the fragmentation of another relation.
- The definition and allocation of fragments are carried out strategically to achieve locality of reference, improved reliability and availability, acceptable performance, balanced storage capacities and costs, and minimal communication costs. The three correctness rules of fragmentation are: completeness, reconstruction, and disjointness.
- There are four allocation strategies regarding the placement of data: centralized (a single centralized database), fragmented (fragments assigned to one site), complete replication (complete copy of the database maintained at each site), and selective replication (combination of the first three).
- The DDBMS should appear like a centralized DBMS by providing a series of transparencies. With distribution transparency, users should not know that the data has been fragmented/ replicated. With transaction transparency, the consistency of the global database should be

maintained when multiple users are accessing the database concurrently and when failures occur. With performance transparency, the system should be able to efficiently handle queries that reference data at more than one site. With DBMS transparency, it should be possible to have different DBMSs in the system.

Review question

12.1 Explain what is meant by a DDBMS and discuss the motivation in providing such a system.

12.2 Compare and contrast a DDBMS with distributed processing. Under what circumstances would you choose a DDBMS over distributed processing?

12.3 Compare and contrast a DDBMS with a parallel DBMS. Under what circumstances would you choose a DDBMS over a parallel DBMS?

12.4 Discuss the advantages and disadvantages of a DDBMS.

12.5 What is the difference between a homogeneous and a heterogeneous DDBMS? Under what circumstances would such systems generally arise?

12.6 What is the main differences between LAN and WAN?

12.7 What functionality do you expect in a DDBMS?

12.8 What is a multi-database system? Describe reference architecture for such a system.

Chapter 13: Object Oriented Database

General objectives:

In this chapter you will be able to: Explain the limitations of Relational databases and the need of Object oriented databases

Specific objectives:

- Identify Complex Data Types
- Explain Structured Types and Inheritance in SQL
- Data types (arrays, multi-set etc) and structure in Object oriented databases using SQL
- Object-Identity and Reference Types in SQL
- Explain ODL and OQL
- Compare Object-Oriented versus Object-Relational

- Give examples of Object oriented and object relational database implementation

Chapter Summery

- The object-relational data model extends the relational data model by providing a richer type system including collection types and object orientation.
- Collection tuples include nested relations, sets, multi-sets, and arrays, and the object-relational model permits attributes of a table to be collections.
- Object orientation provides inheritance with subtypes and sub-tables, as well as object (tuple) references
- Object-relational database systems (that is, database systems based on the object-relation model) provide a convenient migration path for users of relational databases who wish to use object-oriented features
- The relational model, and relational systems in particular, have weaknesses such as poor representation of 'real world' entities, semantic overloading, poor support for integrity and enterprise constraints, limited operations, and impedance mismatch. The limited modeling capabilities of relational DBMSs have made them unsuitable for advanced database applications.
- The concept of encapsulation means that an object contains both a data structure and the set of operations that can be used to manipulate it. The concept of information hiding means that the external aspects of an object are separated from its internal details, which are hidden from the outside world.
- An object is a uniquely identifiable entity that contains both the attributes that describe the state of a 'real world' object and the actions (behavior) that are associated with it. Objects can contain other objects. A key part of the definition of an object is unique identity. In an object-oriented system, each object has a unique system-wide identifier (the OID) that is independent of the values of its attributes and, ideally, invisible to the user.
- Methods define the behavior of the object. They can be used to change the object's state by modifying its attribute values or to query the value of selected attributes. Messages are the means by which objects communicate. A message is simply a request from one object (the

sender) to another object (the receiver) asking the second object to execute one of its methods. The sender and receiver may be the same object.

- Objects that have the same attributes and respond to the same messages can be grouped together to form a class. The attributes and associated methods can then be defined once for the class rather than separately for each object. A class is also an object and has its own attributes and methods, referred to as class attributes and class methods, respectively. Class attributes describe the general characteristics of the class, such as totals or averages.
- Inheritance allows one class to be defined as a special case of a more general class. These special cases are known as subclasses and the more general cases are known as superclasses. The process of forming a superclass is referred to as generalization; forming a subclass is specialization. A subclass inherits all the properties of its superclass and additionally defines its own unique properties (attributes and methods).

All instances of the subclass are also instances of the superclass. The principle of substitutability states that an instance of the subclass can be used whenever a method or a construct expects an instance of the superclass.

- Overloading allows the name of a method to be reused within a class definition or across definitions. Overriding, a special case of overloading, allows the name of a property to be redefined in a subclass. Dynamic binding allows the determination of an object's type and methods to be deferred until runtime.
- In response to the increasing complexity of database applications, two 'new' data models have emerged: the Object-Oriented Data Model (OODM) and the Object-Relational Data Model (ORDM). However, unlike previous models, the actual composition of these models is not clear. This evolution represents the third generation of DBMSs.

Review questions

13.1 Discuss the general characteristics of advanced database applications.

13.2 Discuss why the weaknesses of the relational data model and relational DBMSs may make them unsuitable for advanced database applications.

13.3 Define each of the following concepts in the context of an object-oriented data model:

(a) Abstraction, encapsulation, and information hiding;

(b) Objects and attributes;

(c) Object identity;

(d) Methods and messages;

(e) Classes, subclasses, superclasses, and inheritance;

(f) Overriding and overloading;

(g) Polymorphism and dynamic binding.

13.4 Discuss the difficulties involved in mapping objects created in an object-oriented programming language to a relational database.

13.5 Describe the three generations of DBMSs.

13.6 Describe how relationships can be modeled in an OODBMS.

13.7 Describe the different modeling notations in the UML.

Chapter 14: Introduction to data warehousing

General Objectives

At the end of this chapter you will be able to: Explain How data warehousing evolved and the main concepts and benefits associated with data warehousing.

Specific objectives:

At the end of this chapter you will be able to:

- Explain how Online Transaction Processing (OLTP) systems differ from data warehousing.
- Identify the problems associated with data warehousing.
- Discuss the architecture and main components of a data warehouse.
- Explain the important data flows or processes of a data warehouse.
- Discuss the main tools and technologies associated with data warehousing.
- Explain the issues associated with the integration of a data warehouse and the importance of managing metadata.
- Explain the concept of a data mart and the main reasons for implementing a data mart.
- Explain the main issues associated with the development and management of data marts.

Chapter Summary

- Data warehousing is subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision-making process. A data warehouse is data management and data analysis technology.
- Data Web house is a distributed data warehouse that is implemented over the Web with no central data repository. The potential benefits of data warehousing are high returns on investment, substantial competitive advantage, and increased productivity of corporate decision-makers.
- A DBMS built for Online Transaction Processing (OLTP) is generally regarded as unsuitable for data warehousing because each system is designed with a differing set of requirements in mind. For example, OLTP systems are design to maximize the transaction processing capacity, while data warehouses are designed to support *ad hoc* query processing.
- The major components of a data warehouse include the operational data sources, operational data store, load manager, warehouse manager, query manager, detailed, lightly and highly summarized data, archive/backup data, metadata, and end-user access tools.
- The operational data source for the data warehouse is supplied from mainframe operational data held in first generation hierarchical and network databases, departmental data held in proprietary file systems, private data held on workstations and private servers and external systems such as the Internet, commercially available databases, or databases associated with an organization's suppliers or customers.
- The operational data store (ODS) is a repository of current and integrated operational data used for analysis. It is often structured and supplied with data in the same way as the data warehouse, but may in fact simply act as a staging area for data to be moved into the warehouse. The load manager (also called the *frontend* component) performs all the operations associated with the extraction and loading of data into the warehouse. These operations include simple transformations of the data to prepare the data for entry into the warehouse.
- The warehouse manager performs all the operations associated with the management of the data in the warehouse. The operations performed by this component include analysis of data to ensure consistency, transformation and merging of source data, creation of indexes and views, generation of denormalizations and aggregations, and archiving and backing-up data.

- The query manager (also called the *backend* component) performs all the operations associated with the management of user queries. The operations performed by this component include directing queries to the
  appropriate tables and scheduling the execution of queries.
- End-user access tools can be categorized into five main groups: data reporting and query tools, application development tools, executive information system (EIS) tools, Online Analytical Processing (OLAP) tools, and data mining tools.
- Data warehousing focuses on the management of five primary data flows, namely the inflow, up-flow, down-flow, outflow, and meta-flow.
- Inflow is the processes associated with the extraction, cleansing, and loading of the data from the source systems into the data warehouse.
- Up-flow is the processes associated with adding value to the data in the warehouse through summarizing, packaging, and distribution of the data.
- Down-flow is the processes associated with archiving and backing-up of data in the warehouse.
- Outflow is the processes associated with making the data available to the end-users.
- Meta flow is the processes associated with the management of the metadata (data about data).
- Data mart is a subset of a data warehouse that supports the requirements of a particular department or business function. The issues associated with data marts include functionality, size, load performance, users 'access to data in multiple data marts, Internet/intranet access, administration, and installation.

Review Questions

14.1 Define the terms data warehousing and data mining.

14.2 What is data warehousing and why do we need it?

14.3 What are the rules for data warehouses?

Chapter 15 Introduction to Data Mining

General objectives

At the end of this chapter you will be able:

Explain the concepts associated with data mining.

Specific objectives

- Explain the data mining process and its operational techniques
- Identify the main features of data mining operations, including predictive modeling, database segmentation, link analysis, and deviation detection.
- Discuss the techniques associated with the data mining operations.
- Identify important characteristics of data mining tools.

Chapter Summary

- **Data mining** is the process of extracting valid, previously unknown, comprehensible, and actionable information from large databases and using it to make crucial business decisions.
- There are four main operations associated with data mining techniques: predictive modeling, database segmentation, link analysis, and deviation detection.
- Techniques are specific implementations of the operations (algorithms) that are used to carry out the data mining operations. Each operation has its own strengths and weaknesses.
- **Predictive modeling** can be used to analyze an existing database to determine some essential characteristics (model) about the data set. The model is developed using a supervised learning approach, which has two phases: training and testing. Applications of predictive modeling include customer retention management, credit approval, cross-selling, and direct marketing. There are two associated techniques: classification and value prediction.
- **Database segmentation** partitions a database into an unknown number of segments, or clusters, of similar records. This approach uses unsupervised learning to discover homogeneous sub-populations in a database to improve the accuracy of the profiles.
- **Link analysis** aims to establish links, called associations, between the individual records, or sets of records, in a database. There are three specializations of link analysis: associations discovery, sequential pattern discovery, and similar time sequence discovery. Associations discovery finds items that imply the presence of other items in the same event. Sequential pattern discovery finds patterns between events such that the presence of one set of items is followed by another set of items in a database of events over a period of time.

  Similar time sequence discovery is used, for example, in the discovery of links between two sets of data that are time-dependent, and is based on the degree of similarity between the patterns that both time series demonstrate.

- **Deviation detection** is often a source of true discovery because it identifies outliers, which express deviation from some previously known expectation and norm. This operation can be performed using *statistics* and *visualization* techniques or as a by-product of data mining.
- The Cross Industry Standard Process for Data Mining (**CRISP-DM**) specification describes a data mining process model that is not specific to any particular industry or tool.
- The important characteristics of data mining tools include: data preparation facilities; selection of data mining operations (algorithms); scalability and performance; and facilities for understanding results.
- A data warehouse is well equipped for providing data for mining as a warehouse not only holds data of high quality and consistency, and from multiple sources, but is also capable of providing subsets (views) of the data for analysis and lower level details of the source data, when required.

Review Questions

15.1Discuss what data mining represents.

15.2 What Can Data Mining Do?

15.3 Describe how the following data mining operations are applied and provide typical examples for each:

(a) Predictive modeling, (b) Database segmentation, (c) Link analysis, (d) Deviation detection

15.4 Describe the main aims and phases of the CRISP-DM model.

15.5 Discuss the relationship between data warehousing and data mining.

Answers to selected questions

Chapter 1

1.1 (b) Database is a shared collection of logically related data designed to meet the information needs of an organization. Since it is a shared corporate resource, the database is integrated with minimum amount of or no duplication.

1.2 (c) A database-management system (DBMS) is a collection of interrelated data and a set of programs to access those data. The collection of data, usually referred to as the database, contains information relevant to an enterprise.

Chapter 2

2.2 The three levels of ANSI-SPARC model are

i. Internal level The physical representation of the database on the computer. This level describes *how* the data is stored in the database.

ii. Logical data independence Logical data independence refers to the immunity of the external schemas to changes in the conceptual schema.

iii. Physical data independence Physical data independence refers to the immunity of the conceptual schema to changes in the internal schema.

2.3 Data model is an integrated collection of concepts for describing and manipulating data, relationships between data, and constraints on the data in an organization.

2.7 The major components of DBMS are

- Query processor ,Database manager (DM) , File manager ,DML preprocessor ,DDL compiler
- Catalog manager

Chapter 3

3.1 **(**a) A relation is a table with columns and rows.

(b) An attribute is a named column of a relation.

(c) A domain is the set of allowable values for one or more attributes

(e)The degree of a relation is the number of attributes it contains. The cardinality of a relation is the number of tuples it contains.

3.3 A relation has the following properties:

- the relation has a name that is distinct from all other relation names in the relational schema;
- each cell of the relation contains exactly one atomic (single) value;
- each attribute has a distinct name;
- the values of an attribute are all from the same domain;
- each tuple is distinct; there are no duplicate tuples;
- the order of attributes has no significance;
- the order of tuples has no significance, theoretically. (However, in practice, the order

may affect the efficiency of accessing tuples.)

Chapter 5

5.1 The two major components are:

- Data Definition Language (DDL) for defining the database structure and controlling access to the data;
- Data Manipulation Language (DML) for retrieving and updating data.

Chapter 6

6.1 Database security is The mechanisms that protect the database against intentional or accidental threats.

6.2 The main the types of threat (i) Theft and fraud (ii). Loss of confidentiality (iii). Loss of privacy (iv). Loss of integrity (v). Loss of availability

Chapter 7

7.1 Transaction is An action, or series of actions, carried out by a single user or application program, which reads or updates the contents of the database.

7.9 **Timestamp** is a unique identifier created by the DBMS that indicates the relative starting time of a transaction.

Chapter 8

8.1 Inheritance allows one class to be defined as a special case of a more general class. These special cases are known as **subclasses** and the more general cases are known as **superclasses**.

8.2 The process of forming a superclass is referred to as **generalization** and the process of forming a subclass is **specialization**

Chapter 9

9.1 Superkey: an attribute or set of attributes that uniquely identifies a tuple within a relation.

Candidate key: A superkey such that no proper subset is a superkey within the relation

Primary key : The candidate key that is selected to identify tuples uniquely within the relation

Foreign key :an attribute, or set of attributes, within one relation that matches the candidate key of some (possibly the same) relation.

9.3 An entity type is referred to as being **strong** if its existence does not depend upon the existence of another entity type.

Weak entity type An entity type that is existence-dependent on some other entity type.

Chapter 11

11.1 **Query processing the** activities involved in parsing, validating, optimizing, and executing a query.

11.4 The typical stages of **query decomposition** are analysis, normalization, semantic analysis, simplification, and query restructuring.

Chapter 12

12.1 Distributed database A logically interrelated collection of shared data (and a description of this data) physically distributed over a computer network.

Distributed DBMS (DDBMS) :The software system that permits the management of the distributed database and makes the distribution transparent to users.

12.6 Communication networks may be classified in several ways. One classification is according to whether the distance separating the computers is short (local area network) or long (wide area network).

A **local area network** (LAN) is intended for connecting computers over a relatively short distance, for example, within an office building, a school or college, or home. Sometimes one building will contain several small LANs and sometimes one LAN will span several nearby buildings. LANs are typically owned, controlled, and managed by a single organization or individual.

A **wide area network** (WAN) is used when computers or LANs need to be connected over long distances. The largest WAN in existence is the Internet.

Chapter 13

13.2 The weaknesses of relational DBMSs are

i. Poor representation of 'real world' entities

ii. Semantic overloading

iii. Poor support for integrity and enterprise constraints

iv. Homogeneous data structure

v. Limited operations

vi. Difficulty handling recursive queries

vii. Impedance mismatch

viii. Other problems with RDBMSs associated with concurrency,

ix. schema changes, and poor navigational access

Chapter 14

14.1 Data warehousing A subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision-making process.

Data mining The process of extracting valid, previously unknown, comprehensible, and actionable information from large databases and using it to make crucial business decisions.

Data mining discovers information within data warehouses that queries and reports cannot effectively reveal.

Chapter 15

15.1 Data mining, or knowledge discovery, is the computer-assisted process of digging through and analyzing

enormous sets of data and then extracting the meaning of the data

15.2Although data mining is still in its infancy, companies in a wide range of industries – including finance,

health care, manufacturing, transportation, are already using data mining tools and techniques to take advantage of historical data

Suggested Readings

1. Database systems Thomas M. Connolly • Carolyn E. Begg, University of Paisley A Practical Approach to Design, Implementation, and Management (4th Edition)
2. Database Management Systems, Ramakrishnan. 2003 (3rd edition)
3. Database Processing, David M. Western Washington University, 2012 (12th edition)
4. Modern Database management Jeffery A. 2007 (8th edition)
5. Database Systems for Management, James F. 2010 (3rd edition)
6. Fundamental of Relational Database Management System Sumathi S. volume 47, 2007
7. The Relational Model for Database Management, E.F. Codd, Version 2, 1999
8. Fundamentals of Database System, Ramez Elmasri, Shamkant B. Navathe. 1994. (2nd edition)
9. Databases: Design, development, and deployment McGraw-Hill, N.Delhi,2001

10. An Introduction to Database System, Bipin C. Desai, Galgotia Publication 2002

11. Distributed Database Management Systems, Saeed K. IEEE Computer Society 2010.

12. Grid and Cloud Database Management, Sandro F. 2011

13. Multimedia Database Management Systems, Guojun Lu. 1999