network or **CODASYL model** by IDS (Integrated Data Store), both developed in the mid-1960s. The **relational model**, proposed by E. F. Codd in 1970, represents the second generation of DBMSs. It has had a fundamental effect on the DBMS community and there are now over one hundred relational DBMSs. The third generation of DBMSs are represented by the **Object-Relational DBMS** and the **Object-Oriented DBMS**.

- 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 List four government sectors in your country that use database systems..

1.2 Discuss each of the following terms:
(a) data
(b) database
(c) database management system
(d) database application program
(e) data independence
(f) security
(g) integrity
(h) views

1.3 Describe the role of database management systems (DBMS) in the database approach. Discuss why knowledge of DBMS is important for database administrators.

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

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

1.6 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

1.7 Discuss the three generations of DBMSs.

1.8 Why are views an important aspect of database management systems?

Exercises

1.9 Interview some users of database systems. Which DBMS features do they find most useful and why? Which DBMS facilities do they find least useful and why? What do these users perceive to be the advantages and disadvantages of the DBMS?

1.10 A database approach addresses several of the problems and challenges associated with the traditional file-based approach. Using a DBMS to control how data is shared with different applications and users, through applications such as views, has a number of advantages. However, the implementation of a database approach has its own

challenges, such as expense. Discuss the various costs associated with the implementation of a database approach.

1.11 Study the *DreamHome* case study presented in Section 11.4 and Appendix A.
- **(a)** In what ways would a DBMS help this organization?
- **(b)** What do you think are the main objects that need to be represented in the database?
- **(c)** What relationships do you think exist between these main objects?
- **(d)** For each of the objects, what details do you think need to be stored in the database?
- **(e)** What queries do you think are required?

1.12 Study the *Wellmeadows Hospital* case study presented in Appendix B.3.
- **(a)** In what ways would a DBMS help this organization?
- **(b)** What do you think are the main objects that need to be represented in the database?
- **(c)** What relationships do you think exist between these main objects?
- **(d)** For each of the objects, what details do you think need to be stored in the database?
- **(e)** What queries do you think are required?

1.13 Discuss what you consider to be the three most important advantages for the use of a DBMS for a company like *DreamHome* and provide a justification for your selection. Discuss what you consider to be the three most important disadvantages for the use of a DBMS for a company like *DreamHome* and provide a justification for your selection.

1.14 Organizations have a vital need for quality information. Discuss how the following database roles relate to each other.
- **(a)** Data Administrator
- **(b)** Database Administrator
- **(c)** Database Designer
- **(d)** Application Developer
- **(e)** End-Users

- 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 hysical considerations. The design of the conceptual schema is critical to the overall success of the system. It is worth spending the time and effort 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; and 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 standard allows dictionaries to be shared and transferred from one system to another.

Review Questions

2.1 Explain the concept of database schema and discuss the three types of schema in a database.

2.2 What are data sublanguages? Why are they important?

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 that you would expect to be provided in a multi-user DBMS.

2.6 Of the facilities described in your answer to Question 2.5, which ones do you think would *not* be needed in a standalone PC DBMS? Provide justification for your answer.

2.7 Discuss the function and importance of the system catalog.

2.8 Discuss the differences between DDL and DML. What operations would you typically expect to be available in each language?

2.9 Discuss the differences between procedural DMLs and nonprocedural DMLs.

2.10 Name four object-based data models.

2.11 Name three record-based data models. Discuss the main differences between these data models.

2.12 What is a transaction? Give an example of a transaction.

2.13 What is concurrency control and why does a DBMS need a concurrency control facility?

2.14 Define the term "database integrity". How does database integrity differ from database security?

Exercises

2.15 Analyze the DBMSs that you are currently using. Determine each system's compliance with the functions that we would expect to be provided by a DBMS. What type of language does each system provide? What type of architecture does each DBMS use? Check the accessibility and extensibility of the system catalog. Is it possible to export the system catalog to another system?

2.16 Write a program that stores names and telephone numbers in a database. Write another program that stores names and addresses in a database. Modify the programs to use external, conceptual, and internal schemas. What are the advantages and disadvantages of this modification?

2.17 Write a program that stores names and dates of birth in a database. Extend the program so that it stores the format of the data in the database: in other words, create a system catalog. Provide an interface that makes this system catalog accessible to external users.

2.17 Write a program that stores names and dates of birth in a database. Extend the program so that it stores the format of the data in the database: in other words, create a system catalog. Provide an interface that makes this system catalog accessible to external users.

2.18 A database approach uses different data models. Common database models include the relational model, the network model and the hierarchical model. Which data model should be chosen under which circumstances and why?

Review Questions

3.1 What is meant by the term 'client–server architecture' and what are the advantages of this approach? Compare the client–server architecture with two other architectures.

3.2 Compare and contrast the two-tier client–server architecture for traditional DBMSs with the three-tier client–server architecture. Why is the latter architecture more appropriate for the Web?

3.3 How is an application server different from a file server?

3.4 What is a data warehouse? How is it different from an OLTP system?

3.5 What is a TP Monitor? What advantages does a TP Monitor bring to an OLTP environment?

3.6 Describe the features of a distributed database management system (DDBMS).

3.7 What technologies and standards are used to develop Web services and how do they relate to each other?

3.8 What is a service-oriented architecture?

3.9 Describe the functions of a database manager?

3.10 What is Cloud computing?

3.11 Discuss the five essential characteristics of cloud computing.

3.12 Discuss the three main service models of cloud computing.

3.13 Compare and contrast the four main deployment models for the cloud.

3.14 What is the difference between Data as a service (DaaS) and Database as a service (DBaaS)?

3.15 Discuss the different architectural models for Database as a service.

3.16 Describe the main components in a DBMS.

3.17 Describe the internal architecture of Oracle.

Exercises

3.18 Examine the documentation sets of Microsoft SQL Server, Oracle, and IBM's DB2 system to identify their support for the following:
(a) client–server architecture
(b) Web services
(c) service-oriented architecture

3.19 Search the Web for a number of Web services other than the ones discussed in Section 3.2. What do these services have in common? Identify whether the services access a database.

3.20 Based on the Oracle architecture described in section 3.7, examine the structure of two other DBMSs of your choice. Describe features common to all three DMBSs.

- **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

4.1 Discuss each of the following concepts in the context of the relational data model:
(a) relation
(b) attribute
(c) domain
(d) tuple
(e) intension and extension
(f) degree and cardinality.

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

4.3 Describe the term "normalized reaction." Why are constraints so important in a relational database?

4.4 Discuss the properties of a relation.

4.5 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.

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

4.7 Define "views." Why are they important in a database approach?

Exercises

The following tables form part of a database held in a relational DBMS:

Hotel (hotelNo, hotelName, city)
Room (roomNo, hotelNo, type, price)
Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo)
Guest (guestNo, guestName, guestAddress)

where Hotel *contains hotel details and* hotelNo *is the primary key;*
Room contains room details for each hotel and (roomNo, hotelNo) forms the primary key;
Booking contains details of bookings and (hotelNo, guestNo, dateFrom) forms the primary key;
Guest contains guest details and guestNo is the primary key.

4.8 Identify the foreign keys in this schema. Explain how the entity and referential integrity rules apply to these relations.

4.9 Produce some sample tables for these relations that observe the relational integrity rules. Suggest some general constraints that would be appropriate for this schema.

4.10 Analyze the RDBMSs that you are currently using. Determine the support the system provides for primary keys, alternate keys, foreign keys, relational integrity, and views.

4.11 Implement the above schema in one of the RDBMSs you currently use. Generate two user-views that are accessible and updatable as well as two other user-views that cannot be updated.

- 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 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 **subselect**; 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

6.1 Briefly describe the four basic SQL DML statements and explain their use.

6.2 Explain the importance and application of the WHERE clause in the UPDATE and DELETE statements.

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

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

6.5 How can results from two SQL queries be combined? Differentiate how the INTERSECT and EXCEPT commands work.

6.6 Differentiate between the three types of subqueries. Why is it important to understand the nature of subquery result before you write an SQL statement?

Exercises

For Exercises 6.7–6.28, use the Hotel schema defined at the start of the Exercises at the end of Chapter 4.

Simple queries

6.7 List full details of all hotels.

6.8 List full details of all hotels in London.

6.9 List the names and addresses of all guests living in London, alphabetically ordered by name.

6.10 List all double or family rooms with a price below £40.00 per night, in ascending order of price.

6.11 List the bookings for which no dateTo has been specified.

Aggregate functions

6.12 How many hotels are there?

6.13 What is the average price of a room?

6.14 What is the total revenue per night from all double rooms?

6.15 How many different guests have made bookings for August?

Subqueries and joins

6.16 List the price and type of all rooms at the Grosvenor Hotel.

6.17 List all guests currently staying at the Grosvenor Hotel.

6.18 List the details of all rooms at the Grosvenor Hotel, including the name of the guest staying in the room, if the room is occupied.

6.19 What is the total income from bookings for the Grosvenor Hotel today?

6.20 List the rooms that are currently unoccupied at the Grosvenor Hotel.

6.21 What is the lost income from unoccupied rooms at the Grosvenor Hotel?

Grouping

6.22 List the number of rooms in each hotel.

6.23 List the number of rooms in each hotel in London.

6.24 What is the average number of bookings for each hotel in August?

6.25 What is the most commonly booked room type for each hotel in London?

6.26 What is the lost income from unoccupied rooms at each hotel today?

Populating tables

6.27 Insert rows into each of these tables.

6.28 Update the price of all rooms by 5%.

General

6.29 Investigate the SQL dialect on any DBMS that you are currently using. Determine the system's compliance with the DML statements of the ISO standard. Investigate the functionality of any extensions that the DBMS supports. Are there any functions not supported?

6.30 Demonstrate that queries written using the UNION operator can be rewritten using the OR operator to produce the same result.

6.31 Apply the syntax for inserting data into a table.

Case Study 2

For Exercises 6.32–6.40, use the Projects schema defined in the Exercises at the end of Chapter 5.

6.32 List all employees from BRICS countries in alphabetical order of surname.

6.33 List all the details of employees born between 1980–90.

6.34 List all managers who are female in alphabetical order of surname, and then first name.

6.35 Remove all projects that are managed by the planning department.

6.36 Assume the planning department is going to be merged with the IT department. Update employee records to reflect the proposed change.

6.37 Using the UNION command, list all projects that are managed by the IT and the HR department.

6.38 Produce a report of the total hours worked by each female employee, arranged by department number and alphabetically by employee surname within each department.

6.39 Remove all project from the database which had no employees worked..

6.40 List the total number of employees in each department for those departments with more than 10 employees. Create an appropriate heading for the columns of the results table.

Case Study 3

For Exercises 6.41–6.54, use the Library schema defined in the Exercises at the end of Chapter 5.

6.41 List all book titles.

6.42 List all borrower details.

6.43 List all books titles published between 2010 and 2014.

6.44 Remove all books published before 1950 from the database.

6.45 List all book titles that have never been borrowed by any borrower.

6.46 List all book titles that contain the word 'database' and are available for loan.

6.47 List the names of borrowers with overdue books.

6.48 How many copies of each book title are there?

6.49 How many copies of ISBN "0-321-52306-7" are currently available?

6.50 How many times has the book title with ISBN "0-321-52306-7" been borrowed?

6.51 Produce a report of book titles that have been borrowed by "Peter Bloomfield."

6.52 For each book title with more than three copies, list the names of library members who have borrowed them.

6.53 Produce a report with the details of borrowers who currently have books overdue.

6.54 Produce a report detailing how many times each book title has been borrowed.

that is created in SQL has an **owner**. The owner can pass **privileges** on to other users using the GRANT statement and can revoke the privileges passed on using the REVOKE statement. The privileges that can be passed on are USAGE, SELECT, DELETE, INSERT, UPDATE, and REFERENCES; INSERT, UPDATE, and REFERENCES can be restricted to specific columns. A user can allow a receiving user to pass privileges on using the WITH GRANT OPTION clause and can revoke this privilege using the GRANT OPTION FOR clause.

Review Questions

7.1 What are the main SQL DDL statements?

7.2 Discuss the functionality and importance of the Integrity Enhancement Feature (IFF).

7.3 What are the privileges commonly granted to database users?

7.4 Discuss the advantages and disadvantages of views.

7.5 Discuss the ways by which a transaction can complete.

7.6 What restrictions are necessary to ensure that a view is updatable?

7.7 What is a materialized view and what are the advantages of a maintaining a materialized view rather than using the view resolution process?

7.8 Describe the difference between discretionary and mandatory access control. What type of control mechanism does SQL support?

7.9 Describe how the access control mechanisms of SQL work.

Exercises

Answer the following questions using the relational schema from the Exercises at the end of Chapter 4:

7.10 Create the **Hotel** table using the integrity enhancement features of SQL.

7.11 Now create the **Room**, **Booking**, and **Guest** tables using the integrity enhancement features of SQL with the following constraints:
- **(a)** **type** must be one of Single, Double, or Family.
- **(b)** **price** must be between £10 and £100.
- **(c)** **roomNo** must be between 1 and 100.
- **(d)** **dateFrom** and **dateTo** must be greater than today's date.
- **(e)** The same room cannot be double-booked.
- **(f)** The same guest cannot have overlapping bookings.

7.12 Create a separate table with the same structure as the **Booking** table to hold archive records. Using the INSERT statement, copy the records from the **Booking** table to the archive table relating to bookings before 1 January 2013. Delete all bookings before 1 January 2013 from the **Booking** table.

7.13 Assume that all hotels are loaded. Create a view containing the cheapest hotels in the world.

7.14 Create a view containing the guests who are from BRICS countries.

7.15 Give the users **Manager** and **Director** full access to these views, with the privilege to pass the access on to other users.

7.16 Give the user **Accounts** SELECT access to these views. Now revoke the access from this user.

7.17 Consider the following view defined on the Hotel schema:

```
CREATE VIEW HotelBookingCount (hotelNo, bookingCount)
AS SELECT h.hotelNo, COUNT(*)
    FROM Hotel h, Room r, Booking b
    WHERE h.hotelNo = r.hotelNo AND r.roomNo = b.roomNo
    GROUP BY h.hotelNo;
```

For each of the following queries, state whether the query is valid, and for the valid ones, show how each of the queries would be mapped on to a query on the underlying base tables.

(a)
```
SELECT *
FROM HotelBookingCount;
```
(b)
```
SELECT hotelNo
FROM HotelBookingCount
WHERE hotelNo ='H001';
```
(c)
```
SELECT MIN(bookingCount)
FROM HotelBookingCount;
```
(d)
```
SELECT COUNT(*)
FROM HotelBookingCount;
```
(e)
```
SELECT hotelNo
FROM HotelBookingCount
WHERE bookingCount > 1000;
```
(f)
```
SELECT hotelNo
FROM HotelBookingCount
ORDER BY bookingCount;
```

7.19 Assume that we also have a table for suppliers:

```
Supplier (supplierNo. partNo, price)
```

and a view SupplierParts, which contains the distinct part numbers that are supplied by at least one supplier:

```
CREATE VIEW SupplierParts (partNo)
AS SELECT DISTINCT partNo
    FROM Supplier s, Part p
    WHERE s.partNo = p.partNo;
```

Discuss how you would maintain this as a materialized view and under what circumstances you would be able to maintain the view without having to access the underlying base tables Part and Supplier.

7.20 Analyze three different DBMSs of your choice. Identify objects that are available in the system catalog. Compare and contrast the object organization, name scheme, and the ways used to retrieve object description.

7.21 Create the *DreamHome* rental database schema defined in Section 4.2.6 and insert the tuples shown in Figure 4.3.

7.22 Use the view you created in exercise 7.13 to discuss how you would improve the performance of the SQL command.

7.23 You are contracted to investigate queries with degraded performance to improve them. Based on the schemas created in previous exercises, discuss the criteria to decide for or against indexing.

Case Study 2

For Exercises 7.24 to 7.40, use the Projects schema defined in the Exercises at the end of Chapter 5.

7.24 Create the Projects schema using the integrity enhancement features of SQL with the following constraints:
(a) sex must be one of the single characters 'M' or 'F'.
(b) position must be one of 'Manager', 'Team Leader', 'Analyst', or 'Software Developer'.
(c) hoursWorked must be an integer value between 0 and 40.

7.25 Create a view consisting of projects managed by female managers and ordered by project number.

7.26 Create a view consisting of the attributes empNo, fName, lName, projName, and hoursWorked attributes.

7.27 Consider the following view defined on the Projects schema:

```
CREATE VIEW EmpProject(empNo, projNo, totalHours)
AS SELECT w.empNo, w.projNo, SUM(hoursWorked)
FROM Employee e, Project p, WorksOn w
WHERE e.empNo = w.empNo AND p.projNo = w.projNo
GROUP BY w.empNo, w.projNo;
```

(a)
```
SELECT*
FROM EmpProject;
```
(b)
```
SELECT projNo
FROM EmpProject
WHERE projNo = 'SCCS';
```
(c)
```
SELECT COUNT(projNo)
FROM EmpProject
WHERE empNo = 'E1';
```
(d)
```
SELECT empNo, totalHours
FROM EmpProject
GROUP BY empNo;
```

General

7.28 Consider the following table:

Part (partNo, contract, partCost)

which represents the cost negotiated under each contract for a part (a part may have a different price under each contract). Now consider the following view ExpensiveParts, which contains the distinct part numbers for parts that cost more than £1000:

```
CREATE VIEW ExpensiveParts (partNo)
AS SELECT DISTINCT partNo
    FROM Part
    WHERE partCost > 1000;
```

Discuss how you would maintain this as a materialized view and under what circumstances you would be able to maintain the view without having to access the underlying base table Part.

- A **trigger** defines an action that the database should take when some event occurs in the application. A trigger may be used to enforce some referential integrity constraints, to enforce complex integrity constraints, or to audit changes to data. Triggers are based on the Event-Condition-Action (ECA) model: the *event* (or *events*) that trigger the rule, the *condition* that determines whether the action should be executed, and the *action* to be taken.
- Advantages of triggers include: eliminates redundant code, simplifies modifications, increases security, improves integrity, improves processing power, and fits well with the client-server architecture. Disadvantages of triggers include: performance overhead, cascading effects, inability to be scheduled, and less portable.

Review Questions

8.1 Advanced SQL deals with SQL/PSM and PL/SQL. What led to the introduction of SQL/PSM?

8.2 Describe the general structure of a PL/SQL block.

8.3 Describe the control statements in PL/SQL. Give examples to illustrate your answers.

8.4 Describe how the PL/SQL statements differ from the SQL standard. Give examples to illustrate your answers.

8.5 What are SQL cursors? Give an example of the use of an SQL cursor.

8.6 How is a procedure different from a function?

8.7 Discuss the differences between BEFORE, AFTER, and INSTEAD OF triggers. Give examples to illustrate your answers.

8.8 Rows can be changed after they have been fetched through a cursor. How can this event be stopped?

8.9 Discuss the advantages and disadvantages of database triggers.

Exercises

For the following questions, use the Hotel schema from the Exercises at the end of Chapter 4.

8.10 List all hotels in the capital cities of BRICS countries. .

8.11 Create a database trigger for the following situations:
- **(a)** The price of all double rooms must be greater than £100.
- **(b)** The price of double rooms must be greater than the price of the highest single room.
- **(c)** A booking cannot be for a hotel room that is already booked for any of the specified dates.
- **(d)** A guest cannot make two bookings with overlapping dates.
- **(e)** Maintain an audit table with the names and addresses of all guests who make bookings for hotels in London (do not store duplicate guest details).

8.12 Create an INSTEAD OF database trigger that will allow data to be inserted into the following view:

```
CREATE VIEW LondonHotelRoom AS
    SELECT h.hotelNo, hotelName, city, roomNo, type, price
    FROM Hotel h, Room r
    WHERE h.hotelNo = r.hotelNo AND city = 'London'
```

8.13 Not all modern DBMSs are embedded with SQL/PSM features. Investigate a DBMS of your choice to determine if it is SQL/PSM compliant. Discuss situations in the *DreamHome* project that require trigger creation.