Solution Manual for Database Systems A Practical Approach to Design Implementation and Management 6th Edition by Connolly and Begg ISBN 0132943263 9780132943260

Full link download
Solution Manual:

Chapter 2 Database Environment

Review Questions

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

See Section 2.1.5

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.*

See Section 2.1

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

An integrated collection of concepts for describing and manipulating data, relationships between data, and constraints on the data in an organization. See Section 2.3.

Object-based data models such as the Entity-Relationship model (see Section 2.3.1). Record-based data models such as the relational data model, network data model, and hierarchical data model (see Section 2.3.2). Physical data models describe how data is stored in the computer (see Section 2.3.3).

2.4   *Discuss the function and importance of conceptual modeling.*

See Section 2.3.4.

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

Data Storage, Retrieval and Update	Authorization Services
A User-Accessible Catalog	Support for Data Communication
Transaction Support	Integrity Services
Concurrency Control Services	Services to Promote Data Independence
Recovery Services	Utility Services

See also Section 2.4

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.*

Concurrency Control Services - only single user.
Authorization Services - only single user, but may be needed if different individuals are to use the DBMS at different times.
Utility Services - limited in scope.
Support for Data Communication - only standalone system.

2.7   *Discuss the function and importance of the system catalog.*

See Section 2.4, Service (2) – User-accessible catalog.

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

DDL - A language that allows the DBA or user to describe and name the entities, attributes, and relationships required for the application, together with any associated integrity and security constraints.

DML - A language that provides a set of operations to support the basic data manipulation operations on the data held in the database.

See Section 2.2.2.

2.9   *Discuss the differences between procedural DMLs and nonprocedural DMLs?*

Procedural DML - A language that allows the user to tell the system what data is needed and exactly how to retrieve the data.

Nonprocedural DML - A language that allows the user to state what data is needed rather than how it is to be retrieved.

See Section 2.2.2.

2.10   *Name four object-based data models.*

- Entity-Relationship (ER)
- Semantic
- Functional
- Object-oriented.

See Section 2.3.1.

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

- **relational data model** - data and relationships are represented as tables, each of which has a number of columns with a unique name
- **network data model** - data is represented as collections of *records*, and relationships are represented by *sets*. Compared with the relational model, relationships are explicitly modeled by the sets, which become pointers in the implementation. The records are organized as generalized graph structures with records appearing as *nodes* (also called *segments*) and sets as *edges* in the graph

- **hierarchical data model** - restricted type of network model. Again, data is represented as collections of *records* and relationships are represented by *sets*. However, the hierarchical model allows a node to have only one parent. A hierarchical model can be represented as a tree graph, with records appearing as *nodes* (also called *segments*) and sets as *edges*.

See Section 2.3.2.

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

A transaction is a series of actions, carried out by a single user or application program, which accesses or changes the contents of the database.

See Section 2.4.

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

A mechanism to ensure that the database is updated correctly when multiple users are updating the database concurrently.

This avoids inconsistencies from arising when two or more transactions are executing and at least one is updating the database.

See Section 2.4.

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

"Database integrity" refers to the correctness and consistency of stored data: it can be considered as another type of database protection. Although integrity is related to security, it has wider implications: integrity is concerned with the quality of data itself. Integrity is usually expressed in terms of *constraints*, which are consistency rules that the database is not permitted to violate.

See Section 2.4.

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?*

To do this you will need to obtain appropriate information about each system. There should be manuals available or possibly someone in charge of each system who could supply the necessary information.

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?

The programs can be written in any suitable language and should be well structured and appropriately commented. Two distinct files result. The structures can be combined into one containing name, address, and telNo, which can be the representation of both the internal and conceptual schemas. The conceptual schema should be created separately with a routine to map the conceptual to the internal schema. The two external schemas also must be created separately with routines to map the data between the external and the conceptual schema. The two programs should then use the appropriate external schema and routines.

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.

Again, the program can be written in any suitable language. It should then be modified to add the data format to the original file. This should not be difficult, if the original program is well structured. The interface for other users operates on the data dictionary and is separate from the original program. A menu-based interface is adequate.