Tutorial 1 – Introduction

WITH SUGGESTED SOLUTIONS

Answers to Review Questions

1. Define each of the following key terms:
   a. *Data*. Stored representations of objects and events that have meaning and importance in the user's environment.
   b. *Information*. Data that have been processed in such a way as to increase the knowledge of the person who uses it.
   c. *Metadata*. Data that describes the properties or characteristics of end-user data and the context of that data.
   d. *Database application*. An application program (or set of related programs) that is used to perform a series of database activities (create, read, update, and delete) on behalf of database users.
   e. *Constraint.* A rule that cannot be violated by database users.
   f. *Database*. An organized collection of logically related data.
   g. *Entity* A person, place, object, event, or concept in the user environment about which the organization wishes to maintain data.
   h. *Database management system.* A software system that is used to create, maintain, and provide controlled access to user databases.
   i. *Conceptual data model (or schema).* A detailed, technology-independent specification of the overall structure of organizational data (could be represented by an entity-relationship diagram).
   j. *Logical data model (or schema).* Data model specific to a particular database approach, for example the relational data model. In the case of a relational data model, elements of the logical model include tables, columns, rows, primary and foreign keys, as well as constraints.
   k. *Physical data model (or schema).* A set of specifications that detail how data from a logical data model (or schema) are stored in a computer's secondary memory for a specific database management system. There is one physical data model (or schema) for each logical data model.

2. Contrast the following terms:
   a. *Structured data*; *unstructured data*. Structured data refers to facts related to objects and events of importance in the user's environment and represent the traditional data that is easily stored and retrieved in traditional databases and data warehouses. Unstructured data refers to multimedia data, such as images, sound and video segments that are now stored as part of the user's business environment.
   b. *Data; information*. Data consist of facts, text, and other multimedia objects, while information is data that have been processed in such a way that it can increase the knowledge of the person who uses it.

Solutions to Problems and Exercises

1. The following information relates to driver license. Differentiate between data and metadata. Identify the type of data.
   - Driver's name, address, and birth date: **structured data**
   - The fact that the driver's name is a 30-character field: **metadata**; fact describing property
   - A photo image of the driver: **unstructured data**
   - An image of the driver's fingerprint: **unstructured data**
   - The make and serial number of the scanning device that was used to scan the fingerprint: **structured data**
   - The resolution (in megapixels) of the camera that was used to photograph the driver: **metadata**; fact describing context
   - The fact that the driver's birth date must precede today's date by at least 16 years: **metadata/constraint**

2. Describe a database that you are familiar with.
   a. Briefly explain what type of system the database supports (e.g. order management, library loans, personal banking).
   b. How do you think using a database affects the quality of the service or product provided by that organization or business? Why?
   c. List the entities and relationships that you think the database contains. If you are not sure of these, first identify the kind of data items that you think would need to be stored in the database and then form entities from these data items. For example, with bookshop database system, some data one could initially list are price, title, author, stock-level. These are characteristics that need to be stored about a particular thing i.e. a book (a.k.a. an entity!).
   d. Next, try to identify the relationships that link the entities.

   Answer: Your responses will vary. For example in a university database, the entities are students, faculty (lecturers), courses, offerings (subjects), and enrollments. The relationships are faculty teaches offerings, students enroll in offerings, offerings made of courses and faculty supervises faculties.

3. Explain the difference between a procedural and a nonprocedural language. Name one procedural language and one nonprocedural language. What statements belong in a procedural language, but not in a nonprocedural language?

   Answer: A nonprocedural language enables users to submit queries without having to coding complex procedures. Nonprocedural languages specify what to retrieve, not the detail of how retrieval occurs (i.e. database navigation).

   - Procedural language: C++, Visual BASIC etc
   - Nonprocedural language: SQL

   Statements that belong in a procedural language but not in a nonprocedural language are loops (for, do-while etc), if-then-else statements, etc.

4. a. What is the purpose of the ANSI/SPARC (3-Schema) architecture?

Answer: ANSI/SPARC architecture is a 3-level database architecture consisting of the external view (user views), conceptual view (E-R diagram), and internal view (logical level – data items and their relationships and physical level – stored records, indexes, etc.). The main purpose of this architecture is to provide a framework for understanding the relationship between external applications (user views) and the logical and physical database levels, and the mapping between these levels. See pg 57 in textbook.

b. Explain the concept of data independence.

Answer: Data independence is property of DBMS that decouples applications (external views) from the database, so that logical and physical changes to the database do not impact on application programs.

c. Differentiate between logical and physical data independence.

Answer: Logical data independence enables logical changes to the database schema (e.g. adding a new column to a table) without necessitating changes to existing applications. Physical data independence enables changes to physical data organization (e.g. adding an index to a table) without impacting on the logical schema.
