Meerza Ahmed - Week 1

Tuesday, August 22, 2023 1:05 PM

Lecture 1: Link | Syllabus

*Class Week 1/15

Database Systems by Coronel & Morris

Chapter 1: Database Systems

* 1.1 *Why Database?*

So, why do we need databases? In today's world, data is ubiquitous (abundant, global, everywhere) and pervasive (unescapable, prevalent, persistent). From birth to death, we generate and consume data. The trail of data starts with the birth certificate and continues all the way to a death certificate (and beyond!).

* 1.2 *Data versus Information*

Data consists of raw facts. The word raw indicates that the facts have not yet been processed to reveal their meaning

Information is the result of processing raw data to reveal its meaning. Data processing can be as simple as organizing data to reveal patterns or as complex as making forecasts or drawing inferences using statistical modeling

* Knowledge—that is, the body of information and facts about a specific subject. Knowledge implies familiarity, awareness, and understanding of information as it applies to an environment. A key characteristic of knowledge is that "new" knowledge can be derived from "old" knowledge.

* 1.3 *Introducing the Database*

A database is a shared, integrated computer structure that stores a collection of the following:

- End-user data—that is, raw facts of interest to the end user
- Metadata, or data about data, through which the end-user data is integrated and managed

A database management system (DBMS) is a collection of programs that manages the database structure and controls access to the data stored in the database. In a sense, a database resembles a very well-organized electronic filing cabinet in which powerful software (the DBMS) helps manage the cabinet's contents.

* 1.4 *Why Database Design is Important*

Database design refers to the activities that focus on the design of the database structure that will be used to store and manage end-user data

A well-designed database facilitates data management and generates accurate and valuable information.

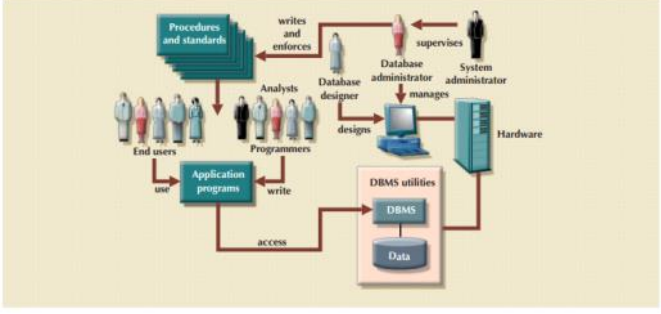
A poorly designed database is likely to become a breeding ground for difficult-to-trace errors that may lead to poor decision making—and poor decision making can lead to the failure of an organization.

* 1.7 *Database Systems*

The term Database System refers to an organization of components that define and regulate the collection, storage, management, and use of data within a database environment.

Note that a DBMS is not the same as a database system, it is however one of several crucial components of a database system.

FIGURE 1.10 THE DATABASE SYSTEM ENVIRONMENT



* *DBMS* is software that manages the database within the database system. Some examples of DBMS software include Microsoft's SQL Server, Oracle Corporation's Oracle, Oracle's MySQL, and IBM's DB2.

* *Procedures* are the instructions and rules that govern the design and use of the database system.

* *Database administrators* manages the DBMS and ensure that the database is functioning properly

* *System administrators* oversee the database system's general operations.

* *Data dictionary* management. The DBMS stores definitions of the data elements and their relationships (metadata) in a data dictionary. In turn, all programs that access the data in the database work through the DBMS. The DBMS uses the data dictionary to look up the required data component structures and relationships

Chapter 2: Data Models

* 2.1 *Data Modeling & Data Models*

Data modeling is the first step in designing a database, and refers to the process of creating a specific data model for a determined problem domain. (A problem domain is a clearly defined area within the real-world environment, with a well-defined scope and boundaries that will be systematically addressed.)
Data model is a relatively simple representation, usually graphical, of more complex real-world data structures

* 2.3 *Data Model Basic Building Blocks*

The basic building blocks of all data models are entities, attributes, relationships, and constraints. An entity is a person, place, thing, or event about which data will be collected and stored

- Entity - A person, place, thing, concept, or event for which data can be stored. See also attribute.
- Attribute - A characteristic of an entity or object. An attribute has a name and a data type
- Relationship - An association between entities.

* An example of a relationship can be between a customer and an agent. The agent can serve many customers, and each customer may only be served by one agent

* Data models use three types of relationships: one-to-many, many-to-many, and one-to-one. Database designers usually use the shorthand notations 1:M or 1..*, M:N or *..*, and 1:1 or 1..1, respectively.

* The following examples illustrate the distinctions among the three relationships:

- One-to-many (1:M or 1..*) - A painter creates many different paintings, but each is painted by only one painter. Thus, the painter (the "one") is related to the paintings (the "many"). Therefore, database designers label the relationship "PAINTER paints PAINTING" as 1:M
- Many-to-many (M:N or *..*) - An employee may learn many job skills, and each job skill may be learned by many employees. Database designers label the relationship "EMPLOYEE learns SKILL" as M:N
- One-to-one (1:1 or 1..1) - A retail company's management structure may require that each of its stores be managed by a single employee. In turn, each store manager, who is an employee, manages only a single store. Therefore, the relationship "EMPLOYEE manages STORE" is labeled 1:1

* A constraint is a restriction placed on the data. Constraints are "rules" and help to ensure data integrity.
Constraints are normally expressed in the form of rules, for example:

- An employee's salary must have values that are between 6,000 and 350,000
- A student's GPA must be between 0.00 and 4.00
- Each class must have one and only one teacher.

* How do you properly identify entities, attributes, relationships, and constraints?
The first step is to clearly identify the business rules for the problem domain you are modeling.

* 2.4 *Business Rules*

A business rule is a brief, precise, and unambiguous description of a policy, procedure, or principle within any organization, large or small—a business, a government unit, a religious group, or a research laboratory—that stores and uses data to generate information. For example, a pilot cannot be on duty for more than 10 hours during a 24-hour period, or a professor may teach up to four classes during a semester.

To be effective, business rules must be easy to understand and widely disseminated to ensure that every person in the organization shares a common interpretation of the rules. Business rules describe, in simple language, the main and distinguishing characteristics of the data *as viewed by the company*.
Examples of business rules are as follows:

- A customer may generate many invoices
- An invoice is generated by only one customer
- A training session cannot be scheduled for fewer than 10 employees or for more than 30 employees

* Note that those business rules establish entities, relationships, and constraints. For example, the first two business rules establish two entities (CUSTOMER and INVOICE) and a 1:M relationship between those two entities.

* 2.4-b *Translating Business Rules into Data Model Components*

As a general rule, a noun in a business rule will translate into an entity in the model, and a verb (active or passive) that associates the nouns will translate into a relationship among the entities.

For example, the business rule "a customer may generate many invoices" contains two nouns (customer and invoices) and a verb (generate) that associates the nouns.
From this business rule, you could deduce the following:

- *Customer* and *invoice* are objects of interest for the environment and should be represented by entities.
- There is a generate relationship between *customer* and *invoice*

* 2.5 *Hierarchical and Network Models*

Hierarchical model was an early database model whose basic concepts and characteristics formed the basis for subsequent database development. This model is based on an upside-down tree structure in which each record is called a segment. The top record is the root segment. Each segment has a 1:M relationship to the segment directly below it. (A segment is the equivalent of a file systems records type.)
The Network model followed the hierarchical model it also maintained a collection of records in a 1:M relationship, however in the network model records were able to have more than one parent.
* Both models had their short comings and were eventually replaced by the relational data model in the 1980s.

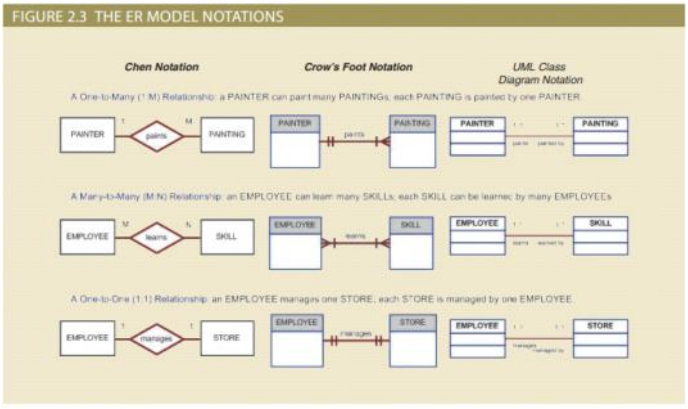
* 2.5-b *The Relational Model*

The relational model was introduced in 1970 by E. F. Codd of IBM in his landmark paper "A Relational Model of Data for Large Shared Databanks" (Communications of the ACM, June 1970, pp. 377–387). The relational model represented a major breakthrough for both users and designers

* 2.5-c *The Entity Relationship Model*

Because, it is easier to examine structures graphically than to describe them in text, database designers prefer to use a graphical tool in which entities and their relationships are pictured. Thus, the entity relationship (ER) model, or ERM, has become a widely accepted standard for data modeling.
Peter Chen first introduced the ER data model in 1976; the graphical representation of entities and their relationships in a database structure quickly became popular because it complemented the relational data model.
* UML or the Unified Modeling language shows the different types of relationships using different ER notations.

FIGURE 2.3 THE ER MODEL NOTATIONS



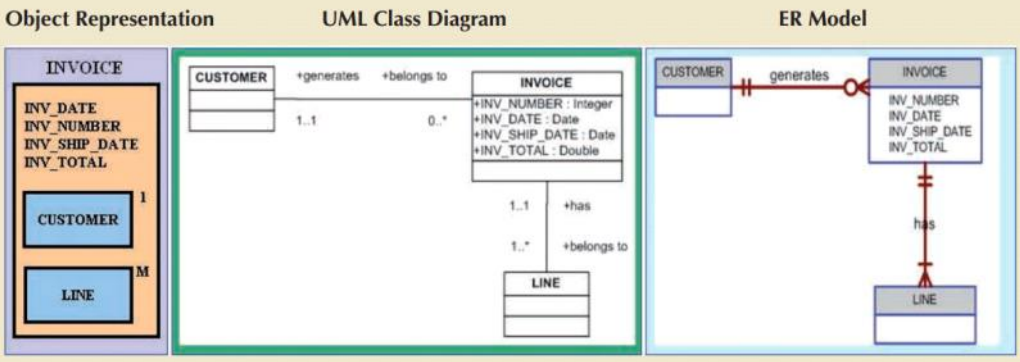
The charts shows some examples of ER notation that has been utilized in a UML
* Chen Notation was developed by Peter Chen and demonstrated relationships and cardinality
* Crows Foot Notation uses a three-pronged symbol to represent the "many" sides of the relationship
* Class Diagram Notation is the newest ER notation and uses a set of symbols and text.

* 2.5-d *The Entity Relationship Model*

In the object-oriented data model (OODM), both data and its relationships are contained in a single structure known as an object. In turn, the OODM is the basis for the object-oriented database management system (OODBMS).The OODM is said to be a semantic data model because semantic indicates meaning.
The OO data model is based on the following components:

- An object is an abstraction of a real-world entity. In general terms, an object may be considered equivalent to an ER model's entity.
- Attributes describe the properties of an object. For example, a PERSON object includes the attributes Name, Social Security Number, and Date of Birth
- Objects that share similar characteristics are grouped in classes. A class is a collection of similar objects with shared structure (attributes) and behavior (methods)
- A class's method represents a real-world action such as finding a selected PERSON's name, changing a PERSON's name, or printing a PERSON's address.
- Classes are organized in a class hierarchy. The class hierarchy resembles an upside-down tree in which each class has only one parent.
- Inheritance is the ability of an object within the class hierarchy to inherit the attributes and methods of the classes above it. For example, two classes, CUSTOMER and EMPLOYEE, can be created as subclasses from the class PERSON. In this case, CUSTOMER and EMPLOYEE will inherit all attributes and methods from PERSON.
- Object-oriented data models are typically depicted using Unified Modeling Language (UML) class diagrams (see below)

FIGURE 2.4 A COMPARISON OF OO, UML AND ER MODELS



* 2.5-f *Emerging Data Model: Big Data and NoSQL*

The need to manage and leverage all converging trends (rapid data growth, performance, scalability, and lower costs) has triggered a phenomenon called "Big Data." Big Data refers to a movement to find new and better ways to manage large amounts of web and sensor-generated data and derive business insight from it, while simultaneously providing high performance and scalability at a reasonable cost

The basic characteristics of Big Data databases are volume, velocity, and variety, or the 3 Vs

- *Volume* refers to the amounts of data being stored.
- *Velocity* refers not only to the speed with which data grows but also to the need to process this data quickly in order to generate information and insight
- *Variety* refers to the fact that the data being collected comes in multiple different data formats

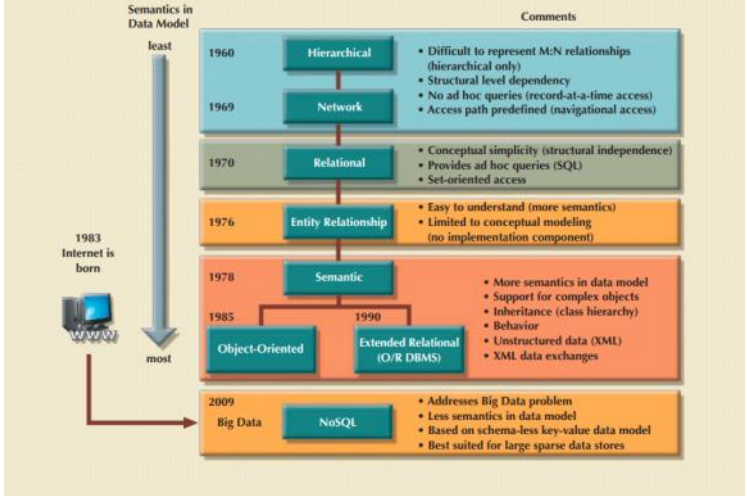
* Every time you search for a product on Amazon, send messages to friends in Facebook, watch a video on YouTube, or search for directions in Google Maps, you are using a NoSQL database. NoSQL databases are not based on the relational model and there is no standard NoSQL data model. Many different data models are grouped under the NoSQL umbrella, from document databases to graph stores, column stores, and key-value stores.

* The key-value data model is based on a structure composed of two data elements: a key and a value, in which every key has a corresponding value or set of values

NoSQL databases use their own native application programming interface (API) with simple data access commands, such as put, read, and delete. Because there is no declarative SQL-like syntax to retrieve data, the program code must take care of retrieving related data in the correct way

* 2.5-g *Data Models: A Summery*

FIGURE 2.6 THE EVOLUTION OF DATA MODELS



* Each new data model addresses the shortcomings of previous models. The network model replaced the hierarchical model because the former made it much easier to represent complex (many-to-many) relationships. In turn, the relational model offers several advantages over the hierarchical and network models through its simpler data representation, superior data independence, and easy-to-use query language; these features made it the preferred data model for business applications.

The OO data model introduced support for complex data within a rich semantic framework. The ERDM added many OO features to the relational model and allowed it to maintain strong market share within the business environment.

In recent years, the Big Data phenomenon has stimulated the development of alternative ways to model, store, and manage data that represents a break with traditional data management