CS34800 Information Systems

Data Models: Overview

Prof. Chris Clifton

24 August, 2016

21

EMERGENCY PREPAREDNESS – A MESSAGE FROM PURDUE

To report an emergency, call 911. To obtain updates regarding an ongoing emergency, sign up for Purdue Alert text messages, view www.purdue.edu/ea.

There are nearly 300 Emergency Telephones outdoors across campus and in parking garages that connect directly to the PUPD. If you feel threatened or need help, push the button and you will be connected immediately.

If we hear a fire alarm during class we will immediately suspend class, evacuate the building, and proceed outdoors. Do not use the elevator.

If we are notified during class of a Shelter in Place requirement for a tornado warning, we will suspend class and shelter in [the basement].

If we are notified during class of a Shelter in Place requirement for a hazardous materials release, or a civil disturbance, including a shooting or other use of weapons, we will suspend class and shelter in the classroom, shutting the door and turning off the lights.

Please review the Emergency Preparedness website for additional information. http://www.purdue.edu/ehps/emergency_preparedness/index.html

Goals of a DBMS

Data Integration	→	Enhances the accessibility of data, reduces redundancies and inconsistencies
Data Independency	→	Simplifies the development of new applications, and the maintainance of existing applications
Centralized Data Control	→	Assures data quality, confidentiality, and integrity

What are we studying?

- Methods to build databases
  - Data modeling
  - Query languages
  - Data integrity, confidentiality, consistency

  While we emphasize the relational model, we will look at other models/approaches as well
- ~~Methods to build DBMSs~~
  - Take CS44800

Motivations for DB Technology
Information and Data

- Data are elementary facts that need to be interpreted in order to convey information
- Example:
  - Consider the byte 00000011. What does this tell you?
  - 3?
  - <Control-C>?
  - End of Text?
- By contrast, saying that 3 is the number of credits of CS348 provides some information

Motivations for DB Technology
Information and Data

- Computer systems represent information through data
- *Data* represents information. *Information* is the (subjective) interpretation of data
- ***Data* -** Physical phenomena chosen by convention to represent certain aspects of our conceptual and real world. The meaning we assign to data are called information. Data is used to transmit and store information and to derive new information by manipulating the data according to formal rules.

  from:

  P.Brinch Hansen. *Operating Systems Principles*. Prentice-Hall, 1973.

Motivations for DB Technology

Information and Data

- One of the fundamental goals of a database management system (DBMS) is to provide an interpretation context to a collection of data, so that users can effectively access information encoded by this collection of data

Goals of a DBMS

- The basic mechanism that makes possible to integrate data is a logical and centralized definition of the data, referred to as database **schema**

- A schema specifies, through a high level formalism referred to as **data model**, the contents of the database

Data Models

- A data model is a "conceptual tool", or *formalism*, that includes three fundamental components:
  - One or more data structures.
  - A notation to specify the data through the data structures of the model.
  - A set of operations for managing data; these operations are defined in terms of the data structures of the model.

Data Models

- A data model allows one to represent real-world entities of interest to a given set of applications
- It is thus useful to identity the basic concepts of such representation; relevant concepts include:
  - *Entity*: an "object" of the application domain
  - *Attribute*: a property of a given entity which meaningful, for the description of the application domain

    Each entity is thus characterized by one or more attributes; an attribute takes one or more values, referred to as *attribute values*, from a set of possible values; such set if referred to as *attribute domain*

Data Models

- A collection of tools for describing
  - Data
  - Data relationships
  - Data semantics
  - Data constraints
- Relational model
- Entity-Relationship data model (mainly for database design)
- Object-based data models (Object-oriented and Object-relational)
- Semistructured data model (XML)
- Other older models:
  - Network model
  - Hierarchical model

Data Models

- *Entity set*: it groups together a set of "objects" characterized by the same features; that is, it groups “similar” entities having the same attributes, even though these attributes do not necessarily have the same values
- A set of attributes the values of which uniquely identify an entity in a given entity set is a **key** for the entity set
- *Relationship*: a correspondence between the elements of two (or more) entity sets

Data Models: Example

- Consider the example of employees and departments:
  - **Entity**: John Smith, the department #30
  - **Entity set**: the set of all employees, the set of all departments
  - **Attribute**: employee name, salary, job, department number, department name
  - **Relationship**: the fact that John Smith works in the department #30

Data Models

- Each data model must answer two fundamental questions:
  - (a) how to represent the entities and their attributes
  - (b) how to represent the relationships

(a) Almost all models use structures such as the record; each component in a record represents an attribute

(b) Data models widely in this respect; relationships can be represented as:

- specific structures, values, pointers (logical or physical)

Levels of Abstraction

- **Physical level:** describes how a record (e.g., instructor) is stored.
- **Logical level:** describes data stored in database, and the relationships among the data.

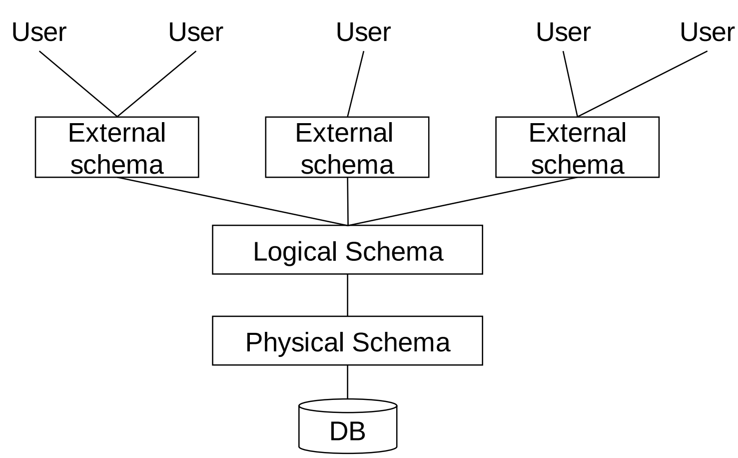
  type *instructor* = **record**

  ID : string;
  name : string;
  dept_name : string;
  salary : integer;

  end;

- **View level:** application programs hide details of data types. Views can also hide information (such as an employee's salary) for security purposes.

Architecture ANSI/SPARC
Three representation levels



Logical Models: Evolution

- First generation (60):
  - Network data model – Codasyl
  - Hierarchical data model
- Relational data model (70)
- Post-relational data models:
  - Nested relational data models
  - Object-oriented data models
  - Deductive data models (e.g. Datalog and its extensions)
  - Object-relational data models
  - XML

Instances and Schemas

- Similar to types and variables in programming languages
- **Logical Schema** – the overall logical structure of the database
  - Example: The database consists of information about a set of customers and accounts in a bank and the relationship between them
    - Analogous to type information of a variable in a program
- **Physical schema**– the overall physical structure of the database
- **Instance** – the actual content of the database at a particular point in time
  - Analogous to the value of a variable
- **Physical Data Independence** – the ability to modify the physical schema without changing the logical schema
  - Applications depend on the logical schema
  - In general, the interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others.

Relational Model

- All the data is stored in various tables.
- Example of tabular data in the relational model

Columns

Rows

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

A Sample Relational Database

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

Database Design

The process of designing the general structure of the database:

- Logical Design – Deciding on the database schema. Database design requires that we find a "good" collection of relation schemas.
  - Business decision – What attributes should we record in the database?
  - Computer Science decision – What relation schemas should we have and how should the attributes be distributed among the various relation schemas?
- Physical Design – Deciding on the physical layout of the database

Database Design (Cont.)

- Is there any problem with this relation?

ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000