CTEC323 Lecture 2

Dan Zingaro
OISE/UT

September 11, 2008

Data Models

- Data modeling: first step in database design
- Modeling clarifies communication by reducing real-world complexities to more easily understood abstractions that define entities and their relationships
- e.g. we abstract from the complexities of a person (height, favourite food, best friend's name . . .) and only store what is important to us

Building Blocks of Data Models

- Entities: anything about which data are to be collected and stored (e.g. customers, products, concert bookings)
- Attributes: characteristics of entities (e.g. first name, last name, ticket price)
- Relationships: associations among entities (1:M, M:N, 1:1)
- Constraints: expressed in the form of rules (e.g. mark between 0 and 100 percent)

Examples of Relationships

Classify the following as 1:1, 1:M or M:N relationships:

- Each person can attend zero or more concerts; each concert can be attended by zero or more people
- Each professor teaches many classes; each class is taught by exactly one professor
- Each customer has one bank account; each bank account is owned by one customer

1:M Relationships

- For 1:1 and M:N relationships, both sides are the same so the direction of the relationship doesn't matter
- e.g. "1:1 relationship between stores and managers", and "1:1 relationship between managers and stores", mean the same thing
- But "1:M relationship between people and cars" and "1:M relationship between cars and people" are different

1:M Relationships...

“A person can own zero or more cars; a car is owned by exactly one person.”

- We have a X:Y relationship between people and cars
- Consider one person. They can own many cars, so Y is M
- Consider one car. It is associated with only one person, so X is 1
- So we have a 1:M relationship between people and cars
- What is the relationship between cars and people?

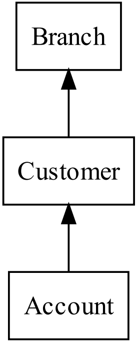
Business Rules

- Collection of data becomes meaningful only when it reflects properly defined business rules
- A business rule is a description of a policy, procedure or principle
- e.g.: “a customer is permitted to have at most one library book overdue to be allowed to check out more books”
- They help standardize the company’s view of data
- They allow designer to develop appropriate relationship participation rules and constraints
- Nouns translate into entities; verbs into relationships

Hierarchical Model

- Logical structure is a tree, consisting of records connected through links
- Example: file cabinet contains drawers, which contain assemblies, which contain smaller parts
- Depicts a set of 1:m relationships between parents and children
- Many of its features form foundation of current models
- Limitations include: complex to implement, difficult to manage, lacked structural independence, only supported 1:m relationships

Hierarchical Model Query Language



```
get first account
where customer.customer-name = "X" and account.balance > Y;
while DB-status = 0 do begin
  get first account
  where customer.customer-name = "X" and account.balance > Y;
end
```

Network Model

- Created to represent complex data relationships more effectively, improve performance, and impose database standard
- This is the first time a data management language was specified
- Records can have more than one parent
- Limitations include: still no ad hoc queries or structural independence

Relational Model

- User sees database as a collection of tables, and can query and manipulate data intuitively
- Tables are related through sharing a common entity
- Achieves data and structural independence because tables are purely logical
- Entity-relationship model is a graphical view of the entities and relationships
- Relational model is a conceptual model; network and hierarchical models are implementation models

Levels of Data Abstraction

- When designing, we start with abstract view and keep adding details
- External Model
  - End users' view of data environment
  - Each business unit uses a subset of the overall data
  - Advantages include: helps determine data used by each business unit, gives designer feedback about model's adequacy, helps ensure security constraints, makes application programming simpler
- Conceptual model (e.g. ER model)
  - Integrate external views into a single view, resulting in global view of database
  - It is independent of hardware and software
  - Gives bird's-eye view of data

Levels of Data Abstraction

- Internal Model
  - Maps conceptual model to DBMS (e.g. entities to tables in a relational database)
  - Represented by data management language (e.g. SQL for relational)
  - Software-dependent
- Physical Model
  - The specifics of how the data are actually saved on disk
  - Hardware- and software-dependent