More Dependencies

CS157A

Chris Pollett

Dec. 5, 2005.

Outline

- Inclusion Dependencies
- Template Dependencies
- Domain Key Normal Form
- Database Design Methodology

Introduction to Inclusion Dependencies

- Two kinds of constraints not handled by our normal forms are:
  - Foreign key constraints
  - class/subclass constraints.
- We'd like dependencies which measure such constraints so that we can come up with good normal forms for such constraints.

Inclusion Dependencies

Definition: An **inclusion dependency** R.X < S.Y between two sets of attributes -- X of schema R, and Y of schema S -- specifies the constraint that, at any specific time where r is a relation state of R and s is a relation state of S, we must have

$$\pi_X(r) \subseteq \pi_Y(s)$$

Note: X and Y must have the same number of attributes and the domain of corresponding attribute must be compatible.

Examples:

DEPARTEMENT.DMGRSSN < EMPLOYEE.SSN

EMPLOYEE.SSN < PERSON.SSN

There are inference rules for IDs. IDs cannot be expressed as JDs and vice-versa We will give a very strong kind of normal form in a minute which handle a class of dependencies containing both IDs and JDs.

Template Dependencies

- Templates can be used to provide a general framework for specifying dependencies -- especially those we know exist on the data but are hard to spell out in terms of our earlier dependencies.
- There are two kinds of templates which are used to specify dependencies: *tuple generating templates* and *constraint templates*.
- Any template consists of a list of *hypothesis tuples* followed by a *template conclusion*.
- For tuple generating templates, the conclusion is a set of tuples that must exist in the relation if the hypothesis tuples are there.
- For constraint generating templates, the template conclusion is a condition that must hold on the hypothesis tuples.

Template Examples

(a) R={A,B,C,D}

hypothesis a1 b1 c1 d1

a1 b1 c2 d2

=========

conclusion c1=c2 and d1=d2

This represents the FD AB-->CD

(b) R={A,B,C,D} S={E,F,G}

hypothesis a1 b1 c1 d1

=======================

conclusion c1 d1 g

Let X={C,D} and Y={E,F} then this represents the inclusion dependency R.X< S.Y

Domain Key Normal Form

- One can try to come up with more and more general normal forms.
- About the most general normal form that has been considered is so called Domain Key Normal Form (DKNF).
- A schema is in DKNF if all constraints and dependencies that hold on valid relation states can be enforced by enforcing just domain constraints and key constraints on the relation.
- Notice this eliminates arbitrary template dependencies and so handles IDs and JDs.
- Still, in practice some constraints are hard to represent in relations using just domain and key constraints.
- For instance, from CAR(MAKE,VIN#) and MANUFACTURE(VIN#, COUNTRY), one might have a constraint that if you know the make is Toyota and the VIN# begin with J the car was made in Japan; for Honda if the second character is a J it was made in Japan.

Database Design Methodology

- We have now completed owr discussion of how to model databases and how to tell if the representations of our models in the relational model are “good”.
- We now finish the semester with a discussion of a process to actually go about designing a database.

Context of Database in Organization

- Databases are important to the successful management of business and government agencies because:
  - Data is regarded as a corporate resource and its management and control is central to the effective working of an organization
  - As more functions of organization are computerized, the need to keep large quantities of data up-to-date increases
  - As the complexity of the data grows, complex relationships among the data need to be modeled and maintained.
  - There is a tendency to consolidate information resources in many organizations.
  - It is often cheaper to reduce personnel costs by letting end-users perform business transactions as much as possible.

Information System Lifecycle

- In a large organization, the DBMS is part of the information system of the organization. This information system includes all resources that are involved in the collection, management, use, and dissemination of the information resources in an organization.
- The life cycle of an information system might look something like:
  - Feasibility Analysis
  - Requirement collection and analysis.
  - Design (both of the DB and associated apps)
  - Implementation
  - Validation and acceptance testing.
  - Deployment, operation, and maintenance.

Database Application System Life Cycle

- Within the macro life cycle of the Information System one also has a micro life cycle for the database application system. This includes:
  - System Definition
  - Database Design
  - Database Implementation
  - Loading or data conversion
  - Application conversion
  - Testing and validation
  - Operation
  - Monitoring and maintenance

Database Design and Implementation Processes

- Just as in software engineering there are different processes one can use in designing a database application system.
- One simple process is as follows:
  - Requirement collection
  - Conceptual database design
  - Data model mapping
  - Physical database design
  - Database system implementation and tuning