Relational Model

CSE462 Database Concepts

Demian Lessa and Jan Chomicki

Department of Computer Science and Engineering
State University of New York, Buffalo

Fall 2013

Data Model

- A data model is a set of formal concepts used to describe the structure of the data and the constraints the data should obey. Often, it also defines operations for data retrieval and modification.
- In database design, we typically use three different kinds of data models.
  - Conceptual: concepts are close to user perceptions.
  - Physical: details of data storage, organization, access methods, etc.
  - Logical: falls between the other two, balancing user views with some representation details.

Data Model

Why different data models?

- Requirements Analysis
  - Determines data, applications, critical operations, etc.
- Conceptual Design
  - High-level description of data and constraints (e.g., ERM).
- Logical Design
  - Conversion of the conceptual design into a database schema.
- Schema Refinement
  - Redundancy elimination through a process called normalization.
- Physical Design
  - Considers workloads, indexes, clustering/partitioning, etc.
- Application and Security Design. . .

Overview

Relational Model (1970)

- Originally proposed by Ted Codd.
- Separates physical implementation from conceptual view.
- Models data independently from its intended or actual use.
  - The main data structure is a relation.
  - A relation consists of a set of attributes and an unordered set of tuples, each containing one value for each attribute of the relation.
  - The concept of a relation comes from logic and mathematics.
  - The model uses standard mathematical operations over the data– relational algebra or relational calculus.

Definitions

The relational model represents relations as two-dimensional tables.

title	year	length	genre
Gone With the Wind	1939	231	drama
Star Wars	1977	124	scifi
Wayne's World	1992	95	comedy

Relation : `Movies`.

- Each row in the `Movies` relation represents a movie and each column a movie property. Every column header is an attribute.
- A relation schema consists of a relation name and a *set of attributes*. For presentation purposes, a standard order is used for the attributes.

  Notation: `Movies(title, year, length, genre)`.
- A database schema is the set of all relation schemas in the database.

Definitions (cont.)

- The arity of a relation is the number of attributes in its schema. Every attribute is associated with a domain, which is an elementary type such as `int` or `string`. Domains may be optionally included in relation schemas:

  `Movies(title:string,year:int,length:int,genre:string)`

- Rows of a relation are called tuples. A tuple is a mapping from the attributes of the relation to tuple components. A tuple component is either a value belonging to the corresponding attribute's domain or is `NULL` (`NULL` is not a value!). The arity of a tuple is the same as the arity of its containing relation.
- A tuple is presented as a list of its components, in the same order as the attributes in the relation schema declaration.

  Notation: ('Gone With the Wind', 1939, 231, 'drama').

Definitions (cont.)

- Relations are sets of tuples, so presentation order is irrelevant.
- Attributes may be reordered without changing the relation, but tuple components must be reordered accordingly.
- In how many different ways can the `Movies` relation be presented?

year	genre	title	length
1939	drama	Gone With the Wind	231
1977	scifi	Star Wars	124
1992	comedy	Wayne's World	95

Relation : `Movies`, alternate presentation.

Schemas and Instances

- A relation instance is the set of tuples of a given relation. Instances normally change over time as data is inserted, deleted, and updated.
- A database instance is the set of all relation instances in the database. Conventional databases maintain one version of each relation: the current instance.
- Relation schemas change much less frequently than instances. When modifying a relation schema, all tuples in the relation must be rewritten to accomodate the changes. It may be difficult or impossible to generate appropriate values for new components in existing tuples.
- Users formulate queries against a database schema. Queries are validated against the database schema and evaluated against the database instance.

Key Constraint

- A set of attributes forms a key for a relation if no two tuples in a relation instance are allowed to have the same values in all key attributes.

  Notation: `Movies(title, year, length, genre).`
- A key is minimal if the set obtained by removing any attribute from the key is no longer a key. Non-minimal keys are called superkeys.
- A relation may have multiple keys. It may also have no natural key, in which case an artificial (synthetic) one may be defined.
- How would you identify:
  - a university student?
  - a company employee?
  - a driver?
  - an automobile?

Exercise 2.2.1

The relations below constitute part of a banking database.

acctNo	type	balance
12345	savings	12000
23456	checking	1000
34567	savings	25

Relation : `Accounts`.

firstName	lastName	idNo	account
Robbie	Banks	901-222	12345
Lena	Hand	805-333	12345
Lena	Hand	805-333	23456

Relation : `Customers`.

Indicate:

- The attributes of each relation.
- The tuples of each relation.
- The components of the first tuple of each relation.
- The relation schema for each relation.
- The database schema.
- A suitable domain for each attribute.
- Another equivalent way to present each relation.

Exercise 2.2.3:

Considering orders of tuples and attributes, how many different ways are there to represent a relation instance if the instance has:

- Three attributes and three tuples?
- Four attributes and five tuples?
- n attributes and m tuples?

Classwork #1

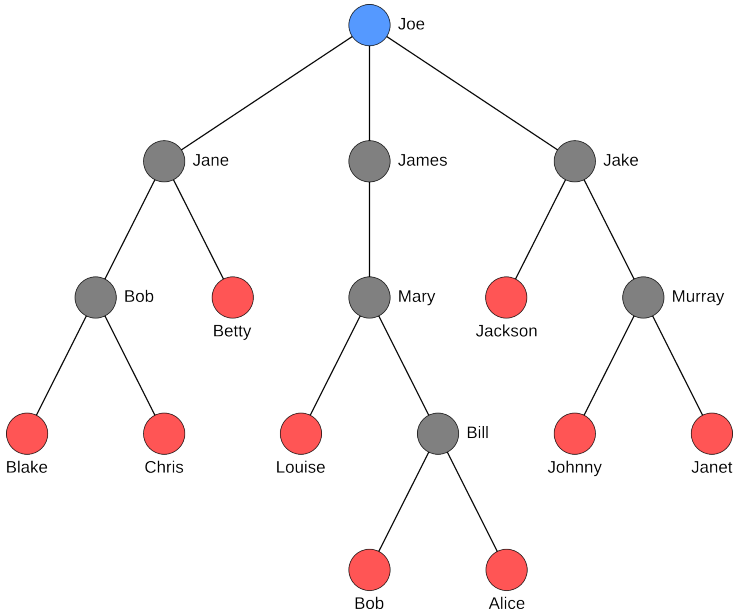
Page	Day	Hits
index.html	2011-01-21	18
schedule.html	2011-01-21	12
syllabus.html	2011-01-21	11
index.html	2011-01-22	18
schedule.html	2011-01-22	9
syllabus.html	2011-01-22	6

Relation : `WebStats`, snapshot of our course's web site statistics.

1. Specify a schema for `WebStats`.
   - Include attribute names, their domains, and a minimal key.
2. Can ('index.html', 2011-01-22, 15) be inserted into `WebStats`?
   - Justify based on your answer above.

Classwork #2

Consider an application that organizes its data as a tree. Each tree node has at most one parent and arbitrary string data. Define a database schema for the application and show an instance encoding the tree below.



Classwork #2 (cont.)

Option #1: each node has any number of data values.

- `Node(nid:int, parentid:int).`
- `Data(nid:int, value:string).`
- Problem: querying ancestor/descendant relationship.

Classwork #2 (cont.)

Option #2: each node has any number of key/value pairs.

- `Node(nid:int, parentid:int).`
- `Data(nid:int, key:string, value:string).`
- Problem: querying ancestor/descendant relationship.

Classwork #2 (cont.)

Option #3: ancestry information implicitly encoded.

- `Node(nid:int, parentid:int, ibegin:real, iend:real).`
- `Data(nid:int, key:string, data:string).`
- `Node.ibegin` $<$ `Node.iend`.
- Every child interval is contained in the parent's interval.
- Every child interval is disjoint from other siblings' intervals.
- Problem: maintaining interval relationships after CRUD operations.
- Also, less declarative– ancestor/descendant relationship relies on explicitly testing interval containment.

Classwork #3

- Define a database schema that encodes a relational database.
- How many relation schemas do we need? Which ones? Why?
- Tip #1: Consider only two relations– one that encodes relation schemas and one that encodes relation instances.
- Tip #2: Assume that schemas consist of relations names with their attributes names and type names.
- Tip #3: Assume that tuple components can be encoded as strings.

Classwork #3 (cont.)

Proposed solution:

- `RelSchema(rel_name, attr_name, attr_type)`
- `RelInstance(rel_name, tid, attr_name, value)`

Explain the encoding of `RelSchema`.

- A relation schema consists of a name and a set of attributes. Each attribute has a name and a type. Thus, `rel_name` is a relation name, `attr_name` is an attribute name, and `attr_type` is the attribute type.
- The pair (`rel_name`, `attr_name`) must be unique: for a fixed relation name, the associated attribute names must form a set and, thus, be unique. Note that the same attribute name may still appear in different relations. Finally, the attribute type depends on the attribute and relation, therefore, should not be part of the key.

Classwork #3 (cont.)

Proposed solution:

- `RelSchema(rel_name, attr_name, attr_type)`
- `RelInstance(rel_name, tid, attr_name, value)`

Explain the encoding of `RelInstance`.

- A relation instance is a set of tuples. Each tuple is a mapping from the relation's attributes to the tuple's components (values). `rel_name` identifies the instance (i.e., the set of tuples), `tid` one tuple in the tuple set, `attr_name` a particular component of the tuple, and `value` is the component's value.
- The triple (`rel_name`, `tid`, `attr_name`) must be unique since it identifies the particular component of every tuple of each relation instance. The value of a tuple component depends is determined by this triple, therefore, should not be part of the key.

Required

- Read sections 2.1 and 2.2 of chapter #2.
- Review the movies database schema of section 2.2.8.