Relational Data Model

DSC 301: Lecture 2

January 27, 2021

Learning Objectives

In this lecture, we introduce the notion of the **relational data model** and the associated terminology. In particular, we will cover:

- What is a data model?
- List types of data models
- Identify problems with some data models
- Define Relational data model
- Define key terms
- Data Keys

1 What are data models?

A model represents real-world objects and events, and their relationships/associations. A model is an abstraction that concentrates on the essential, inherent aspects of an idea (or notion) and ignores incidental or irrelevant properties.

Definition 1. *A* ***data model*** *is a mechanism to describe, organize, and represent data consisting of the following components*

1. Structure - physical representation of the data such as a two-dimensional array.
2. Operations - defines the types of operations that are allowed on the data (include data retrieval and modification).
3. Constraints - specify the data type, whether null values are allowed, and identify keys used in relationships between tables (e.g., common attributes).

Model used in databases

- Network (or Graph-based) model
- Hierarchy model - (restricted graph model) a tree-based model where each node has only one parent. Records = nodes
    - Primary problem is data dependence. The program is depended on the data model. Any changes to the program (or data model) will break the system.
    - Data redundancy is an issue
        - Same information in multiple files
            - Larger memory footprint
            - Difficult to manage
    - Inconsistency is an issue - different programmers create different file structures.
        - Same user, different input
- Object-based model - analogous to Object-Oriented Programming where a class object has properties and methods. There is also an ORDBMS.
- **Relational model** - based on the mathematical relation (covered next time), refers to relations as data structures and includes an algebra to specify queries.

2 Relational Model

E.F. Codd of IBM proposed the **relational data model** in 1970 which limited some data redundancy and integrity problems in the flat file design (see [1]).

The relational model is the primary model used in modern database management systems.

Definitions

- A **relation** is a two-dimensional array, called a **table**, consisting of **rows** and **columns** and form the basis of the relational database model. Note: relations are typically time-varying (living, breathing)
    - Relation = Table (which has rows and columns
    - Set of tuples (see below)
- An **attribute** is a column of a relation. Attributes can be sorted in any order and produces an equivalent relation.
    - Each attribute has a domain, denoted $\mathsf{dom}(A_i)$.
- The **degree** (or **dimension**) of a relation is the number of attributes.

- Each attribute is defined on a **domain** which represents the possible values it can attain. Domains are a “pool of values”, some or all of which may be represented in the database at any instant. Relations should be domain-unordered. If a given relation is domain-ordered, we can transform a domain-ordered relation to a domain-unorderded relation by supplying a unique name for the column.

  - The set of values represeted at some instant is called the **active domain**.

- The elements of a relation are called **tuples**, (also called **records**) and are the **rows** of the table.

- The number of tuples of a relation is the **cardinality**.

  - Symbolically: $|R|$

- **Relational schema** : Name and attributes of a relation (analogous to variable type definition in programming), e.g., The relation “is of type double.”

  Examples of relational schema:
  Courses(department, number, credits)
  Sections(course, room, time, instructor)
  Classrooms(building, room, capacity)
  Instructors(name, email, department,salary)
  Students(name, major, hometown, email)

  Note: Naming conventions - Upper case, plural. This may NOT be standard naming convention.

Table 1: Course Relation

Title	Room	Time	Instructor
Calculus 1	Jones Hall 110	1:00	Dr. Smith
Calculus 1	Ramsey Hall 236	2:00	Dr. Adams
Calculus 2	Jones Hall 120	3:00	Dr. Williams
History 111	Lambert Hall 325	1:00	Dr. Roberts

1. What is the dimension of the relation above?

2. What is the cardinality of the relation above?

3. What is the domain of the attribute Time?

2.1 Primary Keys

- A **superkey** is a set of one or more attributes whose values are used to uniquely identify tuples.
  - A **candidate key** is a minimal superkey, i.e., no subset of is a superkey. The least number of attributes needed to produce a unique identifier.
  - **Primary key** is one of the candidate key. each element (n-tuple) of a relation is called a **primary key**.
    - A relation may contain more than one *nonredundant* primary key.
      - i.e., Candidate keys
  - A combination of domains used as a primary key is called a **composite key**.
    Example: A *Classroom* relation has building and room number as a composite key as room number alone would not uniquely identify a particular classroom
  - Natural vs. Surrogate Keys
    - A **natural key** is a primary key made up of real data.
      Examples: Social Security Number, ISBN, Email Address
      - Pros: Easier to search (key makes sense), Fewer joins (discussed later)
      - Cons: Larger memory, may change
    - **Surrogate keys** that do not have a natural relationship with the rest of the columns in a table. Typically *auto-incremented*.
      - Pros: Small memory footprint, no meaning, no updating, sequential
      - Cons: No useful in searches since there is no meaning
        Example: searching joe@gmail.com
- A **foreign key** (used to cross-reference relations) is a domain of a relation whose elements are values of a primary key in some other (possibly same) relation.

References

[1] Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377387.