The Relational Data Model

SWEN304 / SWEN435

Trimester 1, 2024

Lecturer: Kevin Shedlock

Engineering and Computer Science

Outline

- Basic terms and concepts:
  - Relation Schema, Attribute, Domain
  - Relation, Tuple
  - Relational database schema
  - Relational database instance
- Relational integrity constraints
  - Domain constraints, Attribute constraints
  - Key constraint, Unique constraint
  - Interrelation constraints
- Constraint violation: database updates
- Reading 3: Chapter of the Relational Data Model

History of Relational Database Systems

- Network and hierarchical database systems
  - Emerged in 1960s
  - Complex data structures (hard to comprehend & use)
  - No separation between logical and physical data description (*program data dependency*)
  - Navigational programming languages (low programming productivity)

History of Relational Database Systems

- Introduced in 1970 by E. F. Codd
- Provides a very simple way of storing, manipulating and retrieving information
- The **relational data model** (RDM) represents the database as a collection of relations
- Mathematically, relations are sets of tuples (or records)
- Well-defined concepts and easy to understand
- Clear separation of the (syntactical) schema level and the (semantic) instance level

The Relational Model of Data (RDM)

- The use of relations enables a purely logical treatment of data management tasks
- Relations enables physical data independence
  - All physical structure concepts (storage, pointers, entry point records, hashing algorithms, access tree structures, …) hidden from users/programmers
- Declarative language for DB querying & updating
- The RDM is the *de facto* standard for commercial database systems

Domains, Attributes, Tuples and Relations

The TTUPA STUDENT Domain:

Relation Name *Attribute*

Tuple

STUDENT

StudentID	FName	SName	Phone	Region	Age	GPA
1371352	Kiana	Hawkins	64297702185	Hawkes Bay	19	3.21
1371337	Awhi	Te Amo	64297213285	Nelson	18	3.53
1371240	April	Matariki	64297709291	Wellington	19	3.93
1370718	Andrew	Foster	64272016433	Otaki	22	3.21

Relation Schema: Formally!

- A Relation Schema is denoted by $N(A_1{:}D_1,\ldots, A_n{:}D_n)$
  - N is the name of the relation
  - $A_1, A_2, \ldots, A_n$ are the n attributes of the relation schema
    - Each attribute has a domain D or a set of valid values
    - And a set of constraint C defined on it (discussed later)
- The degree (or arity) of the relation is the number of attributes n of a relation schema N
- **Example**: SUPPLIER (Supplier_no: *INT*, Name: *STRING*, Address: *STRING*)
  - SUPPLIER is the relation name
  - Defined over the three attributes: Supplier_no, Name, Address. Thus, $n=3$

Relation Schema and Its Instances

- A relation is an **instance** of the relation schema $N(A_1{:}D_1,\ldots, A_n : D_n)$
  - denoted by $r(N)$ or simply r

- A relational variable $\rho(N)$ of the type N is the place holder of relation $r(N)$
  - The relational variable $\rho(N)$ (denoted in SQL simply by N) contains an instance of the relation schema N in each moment of time
  - It is the current instance of our relation schema $N(A_1{:}D_1,\ldots, A_n : D_n)$ in the database

Domain

There is a domain D associated with each attribute A, denoted by $dom\ (A) = D$

Domain D is a set of values:

- Either defined by basic data types, such as STRING, DATE:
  - e.g. dom (Lname) = *STRING*
- or defined by type specification:
  $D = \{d_i \mid i = 1, \ldots, n\}$ with D as domain name and d_i as a domain element that satisfies a constraint
  - e.g. *CourseIdDom* = {'SWEN304', 'MATH114', 'STAT193',…}
    dom (Course_id) = *CourseIdDom*
    Set of character strings: 4x capital letters + 3 digits

Relation

- Let $R = \{A_1, \ldots, A_n\}$ be a set of attributes and $dom(A_i) = D_i$ where $i = 1, \ldots, n$
- a relation r over R is a finite set of (n-)tuples t_i:

$$r = \{t_1, \ldots, t_n\}$$

- It is common to use table notation for relations
  - the attributes of R correspond to the column heads
  - the *n*-tuple correspond to the rows
  - the order of the rows in such a table is not important

Attribute

- A property of a set of similar TTUPPA objects, e.g.
  - Id, Fname, Dept, (semantically defined attributes)
  - $A, B, \ldots, X, Y$ (semantically un-interpreted attributes)

- The attribute name is used to interpret the meaning of the data for that attribute

- Some notational conventions:
  - {Fname + Lname, Fname + Major } instead of {{Fname, Lname }, {Fname, Major }}

Tuple

- A tuple t over a relation schema $N(A_1{:}D_1,\ldots, A_n{:}D_n)$ is an ordered list of values, denoted:

$$t = \langle v_1,\ldots, v_n\rangle \qquad \text{or} \qquad t = (v_1,\ldots, v_n)$$

- Each value is from a given *domain* or is a *null* value (**ω**)

- For example: $t =$ <247, ‘Feed The Crowds’, ‘Bumpytown’>
  - Is a tuple (row) in the SUPPLIER relation
  - is called a 3-tuple as it has 3 values

Tuple

- Tuple t is also sometime represented as:

$$t = \{(A_1,v_1),\ldots, (A_n,v_n)\}$$

with (A_n,v_n) as (attribute, value) pairs

- For example: $t =$ { (‘Supplier_No’, 247),
(‘Name’, ‘Feed The Crowds’),
(‘Address’, ‘Bumpytown’)}

Relation Schema, Variable, and Instances

- **Relation Schema*:***

  STUDENT(Lname: *STRING*, Fname: *STRING*,
  Id: *STRING*, Major: *STRING*)

The TTUPA STUDENT Domain instances:

Relation Name *Attribute*

Tuple

STUDENT

StudentID	FName	SName	Phone	Region	Age	GPA
1371352	Kiana	Hawkins	64297702185	Hawkes Bay	19	3.21
1371337	Awhi	Te Amo	64297213285	Nelson	18	3.53
1371240	April	Matariki	64297709291	Wellington	19	3.93
1370718	Andrew	Foster	64272016433	Otaki	22	3.21

Questions

1. Suppose you have a set of tuples, $\{t_1, t_2, t_3\}$, where each t_i is a tuple over attributes R.
   - How many different relations over R can be built by using subsets of this set of tuples?

2^3=8

2. Suppose you are given a set of 100 tuples over the same set of attributes R
   - How many different relations over R can be built by using subsets of this set of tuples?

2^100= a lot

Restrictions

- Let $R = \{A_1, \ldots, A_n\}$ be the set of attributes of a relation schema N and $r(N) = \{t_1, \ldots, t_n\}$
- Restriction of a tuple t onto $\{A_k, \ldots, A_m\} \subseteq \{A_1, \ldots, A_n\}$, denoted as $t\,[A_k, \ldots, A_m]$, refers to a **sublist** of values $(v_k, \ldots, v_m)$ in $t = (v_1, \ldots, v_n)$, for $1 \leq k$ and $m \leq n$

- Example: STUDENT = {Id, Lname, Fname, Major}

  t = (300121, Bond, James, MATH)

  t [*Lname*] = <Bond>,

  t [Fname, Major] = <James, Math>
- Restriction of a relation r onto a set of attributes $\{A_k, \ldots, A_m\}$, is denoted by:

  $$r(N)[A_k, \ldots, A_m] = \{t\,[A_k, \ldots, A_m] \mid t \in r\}$$

A Question

- Given a relation

STUDENT			
Id	**Lname**	**Fname**	**Major**
300111	Smith	Susan	COMP
300121	Bond	James	MATH
300132	Smith	Susan	COMP
300135	John	Cecil	MATH

- What is r(STUDENT)[Lname, Major]?

a)

Lname	Major
Smith	COMP
Bond	MATH
Smith	COMP
John	MATH

b)

Lname	Major
Bond	MATH
Smith	COMP
John	MATH

c)

Lname	Major
Smith	COMP
Bond	MATH
John	MATH

Definitions: In Summary!

- Given a relation schema $N(A_1:D_1, A_2: D_2, \ldots\ldots\ldots, A_n: D_n)$
  - N is the **name** of the relation
  - $A_1, A_2, \ldots\ldots\ldots, A_n$ are the **attributes** of the relation
  - D_i is the **domain** of attribute A_i: $dom(A_i) = D_i$
- For convenience we sometimes omit the domain assignment from a relation schema
- Relation $r(N)$: a specific **state** (or "value" or "population") of N as a *set of tuples* (rows)
  - $r(N) = \{t_1, t_2, \ldots, t_n\}$ where each t_i is an n-tuple
  - $t_i = <v_1, v_2, \ldots, v_n>$ where each v_j is an element of $dom(A_j)$
- $r(R) \subset dom\ (A_1)\ X\ dom\ (A_2)\ X \ldots. X\ dom(A_n)$

***Proper* Subset** ($\subset$) — **Cartesian Product** (X)

Example

- Let $N(A_1, A_2)$ be a relation schema:
- Let $dom(A_1) = \{0,1\}$, $dom(A_2)$ = {a,b,c}
- Then: $dom(A_1)$ X $dom(A_2)$ is all possible combinations:

  {<0,a> , <0,b> , <0,c>, <1,a>, <1,b>, <1,c> }
- The relation state $r(N) \subset dom(A_1)$ X $dom(A_2)$
- Example: $r(N)$ could be {<0,a> , <0,b> , <1,c> }
  - this is one possible state (or "population" or "extension") r of the relation N, defined over A_1 and A_2
  - It has three 2-tuples: <0,a> , <0,b> , <1,c>
  - How many different states (instances) can there be?

Exercises

Consider schema STUDENT(Id, Lname, Fname, Major)

1. Suppose each attribute (e.g. Lname) can have 100 different values
   a) How many different individual records of the STUDENT schema construct can be made?

      e.g. (007007, Bond, James, Comp), or

      (010101, Wong, Sue, Math),

      100^4

   b) How many different student records can be created if we create a constraint that each record must have a unique *Id* value?

2. Suppose, instead, each attribute (e.g. Lname) can have only 2 different values, and there is no restriction on *Id* values
   - How many different sets of records (instances) can be made?

2^(2^4)=2^16
2^16=65536