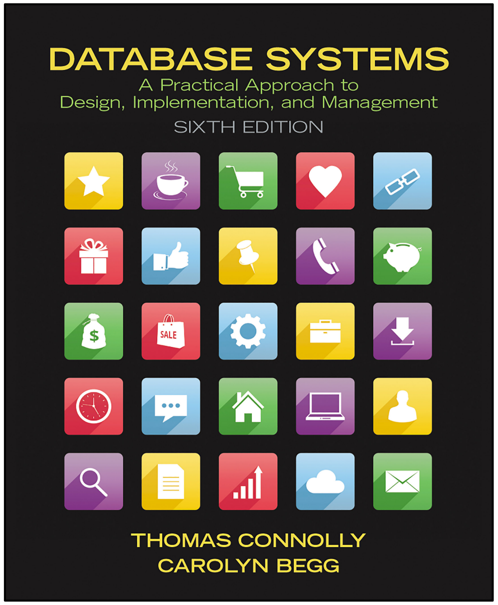
Relational Data Model

Topic 2

Lesson 1 – Relational Data Model

Chapter 4 Connolly and Begg



Logical representation of a relation

Structural representation of a relation

Degree: number of columns (attributes)

Unique_Relational_Name such as student

id	Name	school	major
1	Li, Alex	1	CS
2	Snow, Abigail	3	Accounting
3	Fix, Madison	1	DS

Cardinality: number of rows (tuples)

Domain: allowed values for an attribute

Mathematical Definition of a Relation (1)

- Consider two sets, D_1 & D_2, where $D_1 = \{2, 4\}$ and $D_2 = \{1, 3, 5\}$.
- Cartesian product, $D_1 \times D_2$, is the set of all ordered pairs, where first element is member of D_1 and second element is member of D_2.

$$D_1 \times D_2 = \{(2,1), (2,3), (2,5), (4,1), (4,3), (4,5)\}$$

- Alternative way is to find all combinations of elements with first from D_1 and second from D_2.

Mathematical Definition of a Relation (2)

- Any subset of the Cartesian product is a relation; e.g.

$$R = \{(2,1),(4,1)\}$$

- May specify which pairs are in a relation using some condition for selection; e.g.
  - second element is 1:

$$R = \{(x,y) \mid x \in D_1, y \in D_2, \text{ and } y = 1\}$$

  - first element is always twice the second:

$$S = \{(x,y) \mid x \in D_1, y \in D_2, \text{ and } x = 2y\}$$

Now consider 3 sets

- Consider three sets D_1, D_2, D_3 with Cartesian Product $D_1 \times D_2 \times D_3$; e.g. $D_1 = \{1,3\}$, $D_2 = \{2,4\}$, $D_3 = \{5,6\}$

$$D_1 = \{1, 3\} \quad D_2 = \{2, 4\} \quad D_3 = \{5, 6\}$$

$$D_1 \times D_2 \times D_3 = \{(1,2,5), (1,2,6), (1,4,5), (1,4,6), (3,2,5), (3,2,6), (3,4,5), (3,4,6)\}$$

General definition for a relation

- Cartesian product of *n* sets $\left(D_1, D_2, \ldots, D_n\right)$ is:

  $$D_1 \times D_2 \times \ldots \times D_n = \{\left(d_1, d_2, \ldots, d_n\right) \mid d_1 \in D_1,\ d_2 \in D_2, \ldots, d_n \in D_n\}$$

  usually written as:

  $$\mathop{\mathsf{X}}_{i=1}^{n} D_i$$

- **Any set of *n*-tuples from this Cartesian product is a relation on the *n* sets.**

Given Codd’s definition of a relation

Can we determine the properties of a relation given Codd’s mathematical definition of a relation?

Let’s start with column and row order..

Does order matter? Is Student_instance1 the same as Student_instance2?

Order Does Not Matter

Student_instance1

id	name	school	major
1	Li, Alex	1	CS
2	Snow, Abigail	3	Accounting
3	Fix, Madison	1	DS

Student _instance2

major	name	school	id
DS	Fix, Madison	1	3
Accounting	Snow, Abigail	3	2
CS	Li, Alex	1	1

Duplicate tuples in a relation?

Can a relation have duplicate rows? Does it make sense for the same element to be represented twice in a relation?

Student_instance1

id	name	school	major
1	Li, Alex	1	CS
2	Snow, Abigail	3	Accounting
1	Li, Alex	1	CS

Just like sets, duplicates are not allowed

Duplicate tuples in a relation?

No – a set does not have duplicate elements.

Student_instance1

id	name	school	major
1	Li, Alex	1	CS
2	Snow, Abigail	3	Accounting

How to distinguish tuples?

We need to be able to distinguish one tuple from another tuple. We use the concept of a candidate key to distinguish tuples. A **candidate key** is one or more attributes that must be unique for each tuple. Also, no proper subset of a candidate key can be a candidate key. This means all attributes in the candidate key are necessary for **uniqueness.** This is known as i**rreducibility**.

id	name	school
1	Smith	1
2	Shah	3
3	Li	1

Primary key vs. Alternative keys

Once, all candidate keys are identified for a relation the physical database designer must choose one of the candidate keys as the **primary key** for the relation. The primary key is the chosen candidate key used to represent unique tuples. The other candidate keys not chosen as the primary key are called **alternate keys.**

id	name	SSN
1	Smith	
2	Shah	
3	Li	

Uniqueness without irreducibility

A **super key** provides uniqueness but **not irreducibility**. This means a super key contains additional attributes that do not contribute to the key's uniqueness. Can you name some super keys for the student table?

id	name	school
1	Li, Alex	1
2	Snow, Abigail	3
3	Fix, Madison	1

Values for an attribute

Can an attribute contain values from different domains?
Can an element in a domain represent a set or collection of values?

student_instance3

id_name	school	major
1_Smith	1	CS Accounting
2_Shaj	3	CS
1_LI	1	DS

Values for an attribute

An element for a set must map to a single entity. We must not bury relationships in attributes.

student_instance3

id_name	school	major
1_SMith	1	CS Accounting
2_Shaj	3	CS
1_LI	1	DS

It takes both ID and Major attributes to represent the relationship of a student declaring multiple majors

id	name	school
1	Li, Alex	1
2	Snow, Abigail	3
3	Fix, Madison	1

available_major

major
CS
Accounting
DS

student_major

id	major
1	CS
1	Accounting
2	CS
3	DS

Example of a relational database schema

school

school_ID	name
1	Khoury
2	D'Amore McKim
3	CSSH

student

student_id	name	school_id
1	Li, Alex	1
2	Snow, Abigail	3
3	Fix, Madison	1

available_majors

major
CS
Accounting
DS

student_major

sid	major	sch_id
1	CS	1
2	Accounting	2
3	DS	1

course

course_id	name	school_id
1		
2		
3		

All relations should be normalized. Normalization is a process where we reduce redundancy in the database. We will cover this concept in Topic 3.

Representing a relationship

To represent a relationship between entities we create a **foreign key**. A foreign key is an attribute, or set of attributes, within one relation that matches a candidate key of some other relation. Identify the foreign key below.

school

school_id	s_name
1	Khoury
2	D'Amore McKim
3	CSSH

student

student_id	name
1	Li, Alex
2	Snow, Abigail
3	Fix, Madison

Foreign key

student_id	name	school_id
1	Li, Alex	1
2	Snow, Abigail	3
3	Fix, Madison	1

Constraints for the relational data model

So far, we have discussed the structure of a relation. A relational data model also provides constraints for the data. These constraints are rules that the data must follow to be stored in the schema. These rules ensures an accurate representation of the concept.

Domain constraints on field in a database

A domain constraint limits the values that can be stored in a field. If we specify that school_id is an INTEGER, then if we attempt to store a tuple with a non-integer value then the tuple will not be stored in the relation.

school

school_id	name
1	Khoury
2	D’Amore McKim
3	CSSH

student

student_id	name	school_id
1	Li, Alex	1
2	Snow, Abigail	3
3	Fix, Madison	1

NULL Constraint on field in a database

NULL constraint allows the creator of a database to specify which attributes may be missing. The value may be missing because it is unknown or is not applicable to the tuple.

If we declare that a field value cannot be NULL then if you attempt to store a tuple with that field missing, then the tuple will not be stored in the relation.

school

school_ID	name
1	Khoury
2	D'Amore McKim
3	COE

student

student_id	name	school_id
1	Li, Alex	1
2	Snow, Abigail	3
3	Fix, Madison	1

Integrity Constraints

Entity integrity each tuple must be unique from the other tuples. A primary key provides the uniqueness. In a base relation, no attribute of a primary key can be NULL

Referential integrity states that if a foreign key exists in a relation, either the foreign key value must match a candidate key value of some tuple in its home relation or the foreign key value must be wholly null.

school

school_id	name
1	Khoury
2	D'Amore McKim
3	CSSH

student

student_id	name	school_id
1	Li, Alex	1
2	Snow, Abigail	3
3	Fix, Madison	1

General constraints

school

school_id	name
1	Khoury
2	D'Amore McKim
3	CSSH

student

student_id	name	school_id
1	Li, Alex	1
2	Snow, Abigail	3
3	Fix, Madison	1

available_major

major
CS
Accounting
DS

student_major

sid	major	sch_id
1	CS	1
2	Accounting	2
3	DS	1

course

course_id	name	school_id
1		
2		
3		

Additional rules specified by users or database administrators that define or constrain some aspect of the enterprise. Given your knowledge of Northeastern's schools, majors and students what are some general constraints?

Representing concepts

Base Relation

- Named relation corresponding to an entity in the conceptual schema, whose tuples are physically stored in database.

id	name	school	address
1	Li, Alex	1	131 Main
2	Snow, Abigail	3	81 Broad
3	Fix, Madison	1	411 Hue

View

- Dynamic result of one or more relational operations operating on base relations to produce another relation.

student_ID	name	school_id
1	Li, Alex	1
2	Snow, Abigail	3
3	Fix, Madison	1

Table updates are reflected in the view

Views are dynamic, meaning that changes made to base relations that affect views are immediately reflected in the view. For example, if we add Student 4 to the Student table, the Student view will contain the new tuple.

Student table

student_id	name	school_id
1	Li, Alex	Khoury
2	Snow, Abigail	CSSH
3	Fix, Madison	Khoury
4	Shah, Bill	Science

Student view for a user

student_id	name
1	Li, Alex
2	Snow, Abigail
3	Fix, Madison
4	Shah, Bill

Should data be updated via a view?

If a user has access to the Student view, should a user of the view always be able to add tuples to the relation? If yes, what values should the attributes receive if they are not part of the view?

Student table

student_id	name	school_id
1	Li, Alex	Khoury
2	Snow, Abigail	CSSH
3	Fix, Madison	Khoury
4	Shah, Bill	Science
…		
2200	Last, Joe	CSSH

Student view 1

student_id	name
1	Li, Alex
2	Snow, Abigail
3	Fix, Madison
4	Shah, Bill
…	
2200	Last, Joe

Should data be updated via a view (2) ?

What about this view of the student table ?

Student table

student_id	name	school_id
1	Li, Alex	Khoury
2	Snow, Abigail	CSSH
3	Fix, Madison	Khoury
4	Shah, Bill	Science
…		
2200	Last, Joe	CSSH

Student view 2

student_count	school_id
800	Khoury
700	D'Amore McKim
300	Science
400	CSSH

Restrictions to updating a view

- Updates are allowed if query involves a single base relation and contains a candidate key of the base relation.
- Updates are not allowed involving multiple base relations.
- Updates are not allowed involving aggregation or grouping operations.

Summary

In this module you learned:

- The Mathematical definition of a relation
- The properties of a relation
- The definition of entity and referential constraint
- Views and the allowed operations.