CS 1555

www.cs.pitt.edu/~nlf4/cs1555/

The Relational Model

Data models

- How we represent the data stored in the database
    - And relationships between data items

The relational data model

- Proposed by E.F. “Ted” Codd in 1970
  - “A Relational Model of Data for Large Shared Data Banks.”
  - Built on the concept of the mathematical relation
  - Codd won a Turing award for this work in 1981
- First systems came about in 1977-1978
  - System-R
  - Ingres
- First commercial systems in the 1980’s
  - IBM
  - Oracle

Review: Relations

- First, we'll specifically discuss *binary* relations:
  - Definition: Let A and B be two sets. A binary relation from A to B is a subset of $A \times B$.
  - In other words, a binary relation R is a set of ordered pairs (a_i, b_i) where $a_i \in A$ and $b_i \in B$.
    - In general, entities in a relation are called *tuples*

Binary relation example

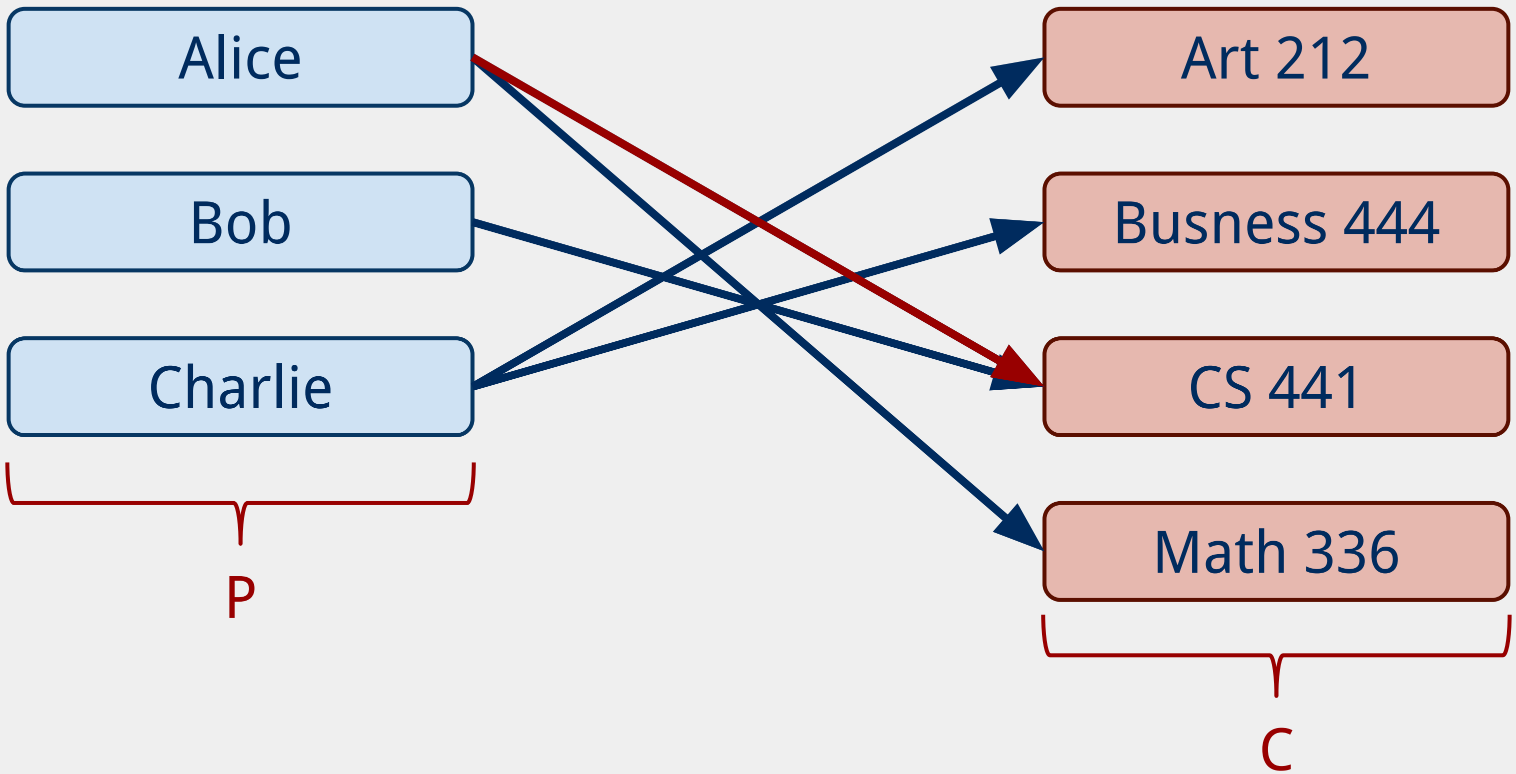
Let's say that Alice and Bob are taking CS 441. Alice is also taking Math 336. Furthermore, Charlie is taking Art 212 and Business 444. Define a relation R that represents the relationship between people and classes.

- P = {Alice, Bob, Charlie}
- C = {CS 441, Math 336, Art 212, Business 444}
- By definition $R \subseteq P \times C$, and we specifically know:
  - (Alice, CS 441) $\in$ R
  - (Bob, CS 441) $\in$ R
  - (Alice, Math 336) $\in$ R
  - (Charlie, Art 212) $\in$ R
  - (Charlie, Business 444) $\in$ R
- R = {(Alice, CS 441), (Bob, CS 441), (Alice, Math 336), (Charlie, Art 212), (Charlie, Business 444)}

Binary relation example

Let's say that Alice and Bob are taking CS 441. Alice is also taking Math 336. Furthermore, Charlie is taking Art 212 and Business 444. Define a relation R that represents the relationship between people and classes.

(Alice, CS 441) ∈ R



How do binary relations compare to functions?

- Recall the definition of a function:
  - Let A and B be nonempty sets. A function, f, is an assignment of exactly one element of set B to each element of set A.
- What does this mean for binary relations?
  - All functions are also relations
  - Not all relations are functions

We can use set operations on relations

Let R be the relation that pairs students with courses that they have taken. Let S be the relation that pairs students with courses that they need to graduate. What do the relations $R \cup S$, $R \cap S$, and $S - R$ represent?

- $R \cup S$ = All pairs (a,b) where
  - student a has taken course b OR
  - student a needs to take course b to graduate
- $R \cap S$ = All pairs (a,b) where
  - Student a has taken course b AND
  - Student a needs course b to graduate
- $S - R$ = All pairs (a,b) where
  - Student a needs to take course b to graduate BUT
  - Student a has not yet taken course b

Relating more than two sets

- Binary sets are rather limited
  - To solve our data management woes, we will need to express much more complex relations!
- Let D_1, D_2, …, D_n be sets. An *n-ary relation* on these sets is a subset of $D_1 \times D_2 \times \ldots \times D_n$. The sets D_1, D_2, …, D_n are called the *domains* of the relation, and n is its *degree* (or *arity*).
  - Made up of n-tuples
- Let R be the relation on Z × Z × Z+ consisting of triples (a, b, m) where $a \equiv b \pmod{m}$
  - What is R's degree?
  - What are R's domains?
  - Is $(8, 2, 3) \in R$?
  - Is $(-1,9,5) \in R$?
  - Is $(11,0,6) \in R$?

Using relations to build databases

- A relational *schema* defines the name of a relation, the names of the attributes of that relation, and domains of those attributes
- $R = \{A_1{:}D_1, A_2{:}D_2, \ldots A_n{:}D_n\}$
  - or simply $R = \{A_1, A_2, \ldots A_n\}$ with domains specified elsewhere
- Domains specify both the *data type* and *format* of attributes
  - Data types must be *atomic*
    - No attribute is composite
  - Formats specify representations of data values

Specifying tuples

- Two approaches:
  - Set of attributes:
    - $t = \{A_1{:}v_1, A_2{:}v_2, \ldots, A_n{:}v_n\}, v_i \in D_i, 1 <= i <= n$
  - List of attributes
    - $t = \{v_1, v_2, \ldots, v_n\}, v_i \in D_i, 1 <= i <= n$
- Clearly:
  - Order is important for the list of attributes approach
  - Not important for the set of attributes approach

Further properties of relations

- The number of tuples in a relation is its *cardinality*
- No duplicate tuples in a relation
  - It is a **set** of tuples
- The order of tuples within a relation is not important
- A value may appear multiple times within a column

n-ary relation example

Students			
Name	ID	Major	GPA
Alice	334322	CS	3.45
Bob	546346	Math	3.23
Charlie	045628	CS	2.75
Denise	964389	Art	4.0

Enrollment	
Stud_ID	Course
334322	CS 441
334322	Math 336
546346	Math 422
964389	Art 707

A note on notation

- For simplicity, we've referred to relations with single capital letters
  - R, S, R ∪ S, etc.
- But, we've also referred to schema names as single capital letters...
  - $R = \{A_1{:}D_1, A_2{:}D_2, \dots A_n{:}D_n\}$
- A bit ambiguous…
- So! Notation to use going forward (and used in the book…)
  - R: a relational schema
    - |R| = arity of R
  - r(R): a relation of schema R
    - |r(R)| = cardinality of r(R)

E.g.

- Say different schools use S as the schema for their students relation
  - |S| is the arity of S
- p(S) could be Pitt's student relation
  - |p(S)| is the number of Pitt students
- d(S) could be Duquesne's students relation
  - |d(S)| would be the number of Duquesne students