CSC 365

Introduction to Database Systems

Relational Model

Three important components of a data model:

✓ Structure

✓ Integrity Constraints

Manipulation

What is an Algebra?

CAL POLY

An algebra consists of **operators** and **operands** that can be combined into **expressions**

In familiar elementary algebra:

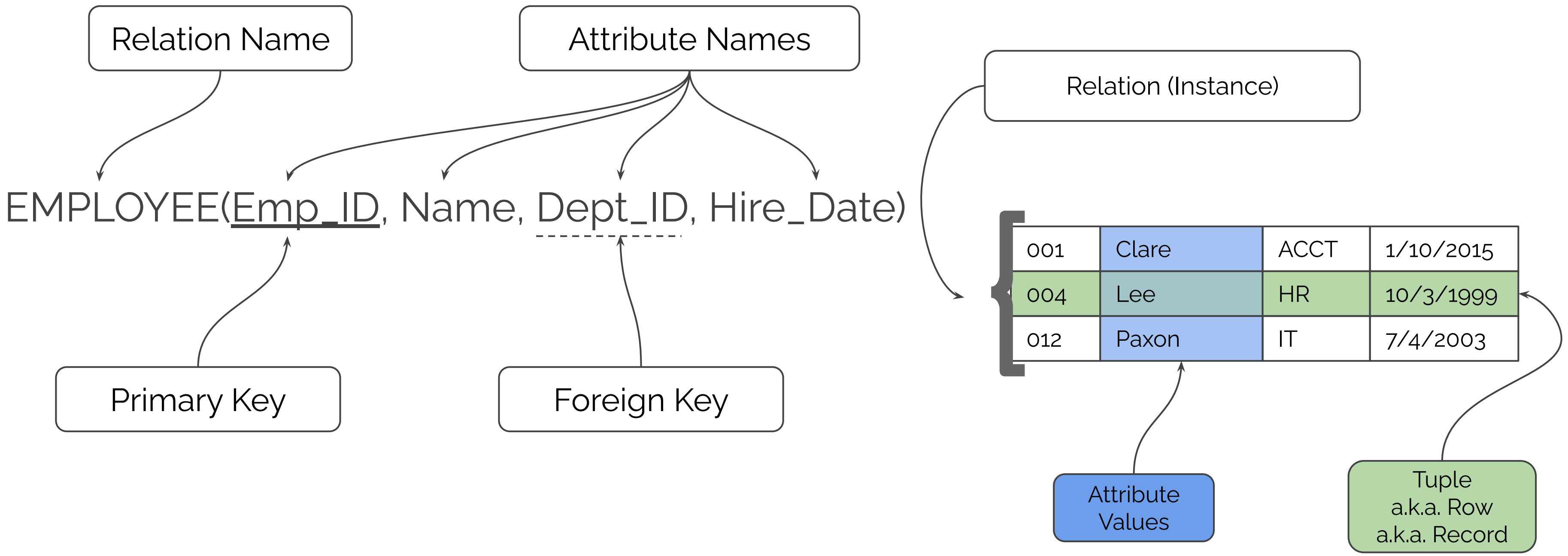
- Operands are variables (*x*) and constants (42)
- Familiar operators: +, -, *, /

Relational Model - Definition Review

- **Tuple**: List of attribute values (synonyms: row, record)
- **Relation**: A set of tuples, or informally, a named two-dimensional table of data
- **Attribute**: A named column of a relation
- **Domain**: Data type of an attribute, *must be atomic* (integer, string, date)
    - Special value (*null*) is a member of every domain

Relational Model - Notation & Structure

CAL POLY



Relational Algebra

CAL POLY

- Operands in Relational Algebra are relations (or expressions that yield relations)
- We will also cover a few derived operators that are not listed here: Division, additional join variations (Semijoin, Antijoin, Outer Join)

(*derived* operators are marked with ✱)

Operator Name	Symbol
Selection	σ
Projection	π
Cartesian Product	$\times$
Union	$\cup$
Difference	$-$
Rename	ρ
Intersection ✱	$\cap$
Natural Join ✱	$\bowtie$
Theta Join ✱	$\bowtie_{\theta}$

Sample Relation Instances

CAL POLY

FLIGHT

TailNum	**PilotID**	**CopilotID**	**Runway**	**Date**
R53Q	K407	D342	S-2	9/1/17
T80H	K407	null	W-2	9/21/17
C97W	D342	*null*	W-2	8/9/21
T80H	D342	K407	W-3	9/9/17

AIRPLANE

TailNum	**Make**	**Model**	**MaxSpeed**
C97W	Boeing	797	*null*
R53Q	Cessna	FG	220
T80H	Airbus	A380	634
G59K	Airbus	A320	450
P88T	Piper	Arrow	180
K30W	Boeing	707	450

PILOT

PilotID	**Name**
D342	Charlie
K407	Juliett
H452	Piper

Sample Relation Instances

CAL POLY

BOOK

ISBN	Title	PubYear	Author
1234	DBMS: The Complete Book	2015	Jennifer Widom
4567	The Art of Computer Programming	1990	Donald Knuth

PATRON

PatronId	Name	SignUpDate
1	Jennifer Widom	7/7/1977
2	Donald Knuth	9/21/1955
3	Grace Hopper	8/9/1942
4	E.F. Codd	7/7/1977

BORROW

Book	PatronId	DateOut	DateDue	DateIn
1234	2	1/1/2009	1/15/2009	1/5/2009
1234	3	1/17/1985	2/3/1985	2/1/1985
4567	1	3/4/2005	4/4/2005	4/15/2005
4567	2	7/1/2018	7/15/2018	*null*

Relational Algebra - Selection - (σ)

The selection operator, applied to relation *R*, produces a new relation with a subset of *R*'s tuples. A condition *C*, *involving the attributes of R*, may be applied.

$$\sigma_C(R) = \{\, t \mid t \in R \text{ and } t \text{ satisfies } C \,\}$$

Examples:

$$\sigma_{MaxSpeed > 500}(\text{AIRPLANE})$$

$$\sigma_{Date <= 9/1/2017 \wedge Runway != 'W\text{-}3'}(\text{FLIGHT})$$

AND	∧	&&
OR	∨	||
NOT	¬	!

Relational Algebra - Projection (π)

The projection operator is used to produce from a relation *R* a new relation that has only some of *R*'s columns, possibly rearranged.

$$\pi_{A1,\ A2,\ \ldots,\ An}(R)$$

Examples:

$\pi_{Make,\ Model,\ TailNum}$ (AIRPLANE)

π_{Name} (PILOT)

Relational Algebra - Rename (ρ)

"Utility" operator that does not affect tuples of relation, but changes that relation *schema* (names of attributes and/or name of the relation itself) Three variations:

$\rho_{S(B1,\ B2,\ \dots,\ Bn)}(R)$ $\rho_{S}(R)$ $\rho_{(B1,\ B2,\ \dots,\ Bn)}(R)$

S is the new relation name, *B1, B2, ..., Bn* are the new attribute names.

Relational Algebra - Union (∪)

Set of elements that are in *R* or *S* or both. (Normal set operation, applied to relations.) Relations *R* and *S* must be ***union compatible***: same number of attributes and matching data types.

$$R \cup S = \{\, t \mid t \in R \text{ or } t \in S \,\}$$

Examples:

$\sigma_{MaxSpeed > 500}$(AIRPLANE) $\cup$ $\sigma_{MaxSpeed < 250}$(AIRPLANE)

π_{Name} (PILOT) $\cup$ π_{Make} (AIRPLANE)

Relational Algebra - Difference (–)

CAL POLY

Set of elements that are in *R* but not in *S*. As with union and intersect, relations *R* and *S* must have the same number of attributes and matching data types.

$$R - S = \{\ t \mid t \in R\ and\ t \notin S\ \}$$

Example:

π_{Make} (AIRPLANE) $-$ π_{Name} (PILOT)

Relational Algebra - Intersection ($\cap$)

CAL POLY

Set of elements that are in both *R* and *S*. As with union, relations *R* and *S* must have the same number of attributes and matching data types.

$$R \cap S = \{\, t \mid t \in R \textit{ and } t \in S \,\}$$

Example:

π_{Name} (PILOT) $\cap$ π_{Make} (AIRPLANE)

Relational Algebra - Intersection ($\cap$)

CAL POLY

Intersection is a derived relational algebra operator.

$$R \cap S \quad = \quad R - (R - S)$$

Relational Algebra - Cartesian Product (×)

The **Cartesian Product** (or *cross-product*, or just *product*) of two relations *R* and *S* pairs the tuples of two relations in all possible ways.

$$R \times S$$

$$= \{ t \mid t = (a_1, \ldots, a_n, b_1, \ldots, b_m) \wedge (a_1, \ldots, a_n) \in R \wedge (b_1, \ldots, b_m) \in S \}$$

Note that result tuples are "flattened"

An example (deck of playing cards):

SUIT × RANK

Relational Model - A Few More Terms

CAL POLY

Degree: number of attributes in a relation

| Schema(R) |

Cardinality: number of tuples in a relation

| R |

Relational Algebra - Cartesian Product ($\times$)

$$R \times S$$

Degree?

Cardinality?

Relational Algebra - Cartesian Product (×)

CAL POLY

Consider:

EMPLOYEE(<u>EmployeeID</u>, Name, HireDate, DeptID)

DEPARTMENT(<u>DeptID</u>, DeptName, Location)

EMPLOYEE × DEPARTMENT

Attribute names?

Relational Algebra - Joins

CAL POLY

- Several types of join operations
  - **Theta join**
  - **Natural join**
  - Equijoin
  - Semijoin
  - Antijoin

- Selectively pair tuples from two relations.

- Joins can be expressed in terms of other basic operators we have seen (cartesian product, selection, projection)

Relational Algebra - Theta Join ($\bowtie_\theta$)

Theta Join pairs tuples from two relations based on an arbitrary condition.

$$R \bowtie_\theta S$$

The result could be constructed as follows:

1. Take the cartesian product (×) of *R* and *S*
2. Select from the product only those tuples that satisfy condition θ (the condition is sometimes represented using the letter *C*)

Relational Algebra - Theta Join ($\bowtie_\theta$)

CAL POLY

Theta Join may be expressed as selection combined with cartesian product

$$R \bowtie_\theta S \quad = \quad \sigma_\theta(R \times S)$$

Relational Algebra - Natural Join ($\bowtie$)

Generally, a ***join*** is the product of two relations *R* and *S*, with *matching* tuples paired. ***Natural Join*** pairs only those tuples from *R* and *S* that agree in whatever attribute names are *common to the schemas of both R* and *S*.

$$R \bowtie S$$

Examples:

FLIGHT $\bowtie$ PILOT

PILOT $\bowtie$ AIRPLANE

Relational Algebra

Relation *R*

A	B
1	2
3	4

Relation *S*

B	C	D
2	5	6
4	7	8
9	10	11

Relational Algebra - Example

Relation *R*

A	B
1	2
3	4

Relation *S*

B	C	D
2	5	6
4	7	8
9	10	11

$R \times S$ Degree? Cardinality? Result? What about column B?

$R \bowtie S$?

Relational Algebra - Example

Relation U

A	B	C
1	2	3
6	7	8
9	2	8

Relation V

B	C	D
2	3	4
2	3	5
7	8	10

$U \bowtie V$ Degree? Cardinality? Result?

$U \bowtie_{A < D} V$

$U \bowtie_{A < D \ \wedge \ U.B \,!= V.B} V$

Relational Algebra - Tuple Counts

Given information about relation structure and constraints, along with a relational algebra expression, it is often useful to estimate (or determine exactly) the cardinality of the result.

We can do so using knowledge of the algebraic operators.

Relational Algebra - Tuple Counts

CAL POLY

Given two relations R1 and R2, where R1 has cardinality N1, R2 has cardinality N2 (N2>N1>0)

RA Expression	Assumptions?	Min Cardinality	Max Cardinality
R1 ∪ R2			
R1 ∩ R2			
R2 − R1			
R1 x R2			
R2 x R1			
$\sigma_{A>3}(R1)$			
$\pi_{A,B}(R2)$			
R1 ⋈ R2			

Relational Algebra - Summary

CAL POLY

- Five core relational algebra operators can be combined to express interesting and complex queries
    - Selection
    - Projection
    - Union
    - Difference
    - Cartesian Product
    - *(and rename, a "utility operator")*

- Derived Operators
    - Intersection
    - Natural Join
    - Theta Join