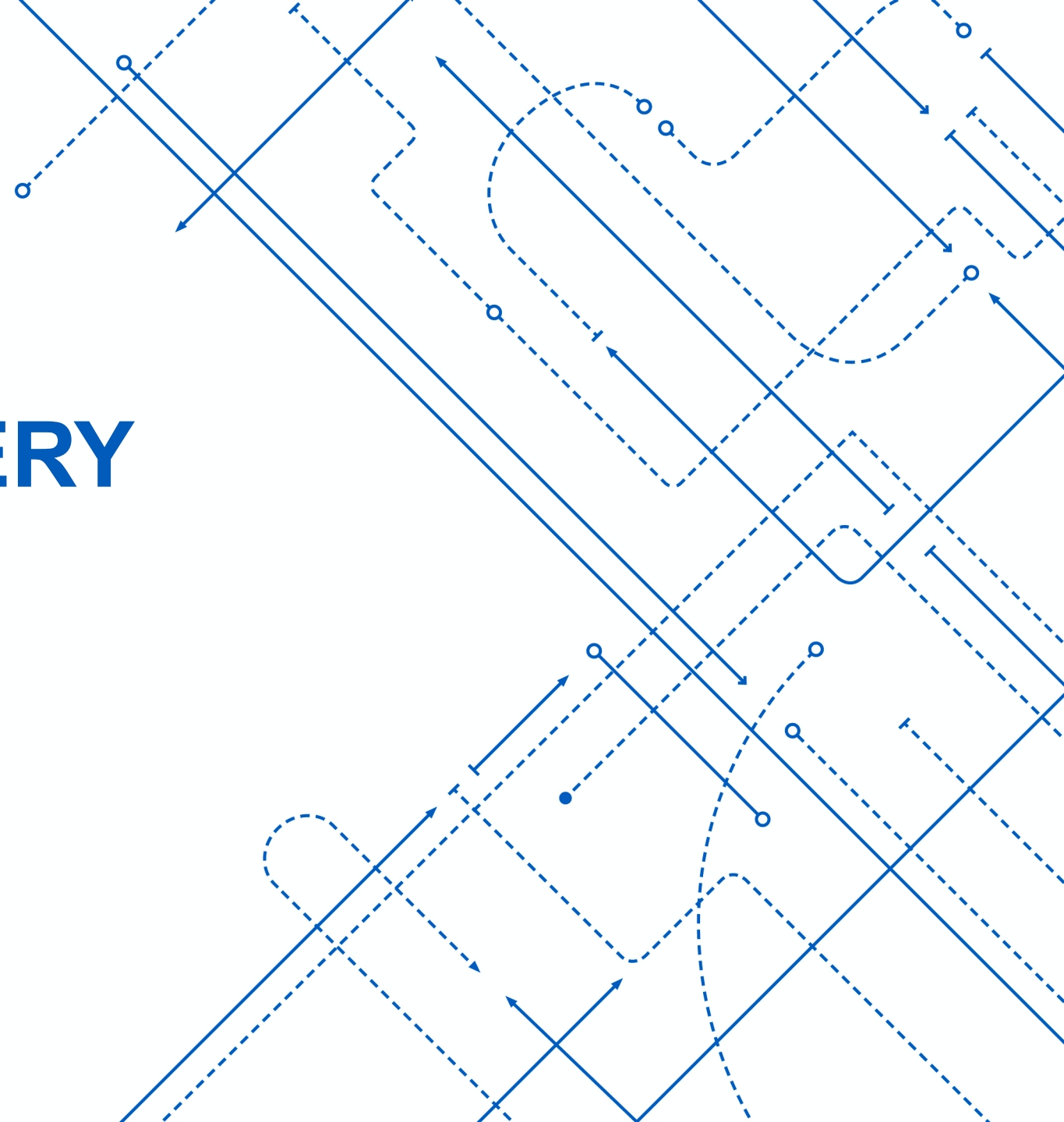


# CSE460/560 DATA MODELS AND QUERY LANGUAGES

## Relational Algebra - 1


Cheng-En Chuang

(Slides Adopted from Jan Chomicki and Ning Deng)

University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

# Outline

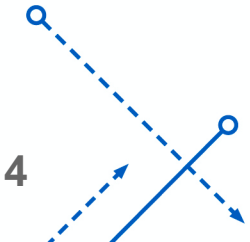
1. Imperative Languages and Declarative Languages
2. Overview of Relational Query Languages
3. Introduction to Relational Algebra
4. Minimal Set of Relational Algebra Operators

# Outline

1. **Imperative Languages and Declarative Languages**
2. Overview of Relational Query Languages
3. Introduction to Relational Algebra
4. Minimal Set of Relational Algebra Operators

# Imperative Languages and Declarative Laguages

- Declarative Language: say WHAT you want
  - Give me a red pen
- Imperative Languages: say HOW you do it
  - Check every pen in the drawer
  - If the pen is red then give it to me and stop looking up



# Why declarative language?

- Make it easier to explore **equivalent** computations to find the best one
  - Replace [thing] with better, but equivalent [thing]
    - How can we tell if one [thing] is equivalent?
    - How can we tell if one [thing] is better than the other?
- Example
  - Give me all red pens
    - Check every pen in the drawer, if the pen is red, then give it to me
    - Order the pens by color, then give me all pens in the red colo bucket
    - O(n) vs. O(n logn)

# Outline

1. Imperative Languages and Declarative Languages
2. **Overview of Relational Query Languages**
3. Introduction to Relational Algebra
4. Minimal Set of Relational Algebra Operators

# Relational Algebra and Relational Calculus

- Relational Algebra
  - A set of algebraic operators
  - Each operator take one or two **relations** as arguments
    - Returns a **relation** as the result
  - Operators can be nested to form expressions
  - Procedural query language
    - Expressions describe how the query can be evaluated
- Relational Calculus
  - A logic language: expressions invlude Boolean operators and quantifiers
  - Declarative query language
    - expressions do not describe how to evaluate the query $\{ t \mid emp(t) \wedge t.salary > 5000\}$
  - We will not talk about it

# SQL

- RA and RC from the basis for relational query languages
  - SQL
    - A mix of relational algebra and relational calculus (logic)
    - The stand query language of the exising DBMS
- Preliminaries
  - Queries are applied to Relations
  - A query works on fixed relation schemas
    - But runs on any relation instance
  - The result of a query is also a **relation**!

# Subtle Issues

- Nulls
  - RA does NOT allow nulls
  - SQL does
- Duplicates
  - By default, RA operates on sets and does not allow duplicates
  - SQL allow duplicates and operates on bags
  - Duplicates irrelevant for most queries
- Order
  - Neither RA nor SQL can specify order within sets of tuples
  - In SQL top-level query results can be ordered (but not in sub-queries)

# Outline

1. Imperative Languages and Declarative Languages
2. Overview of Relational Query Languages
3. **Introduction to Relational Algebra**
4. Minimal Set of Relational Algebra Operators

# What is an “algebra”

- A mathematical system consisting of
  - Operands: variables or values from which new values can be constructed
  - Operators: symbol denoting procedures that construct new values from given values
- Relational Algebra
  - An algebra that can be used as a query languages for relations
    - The operands are relations or variable that represent relations
    - The operators are designed to do the most common things that we need to do with the relations in a database

# Outline

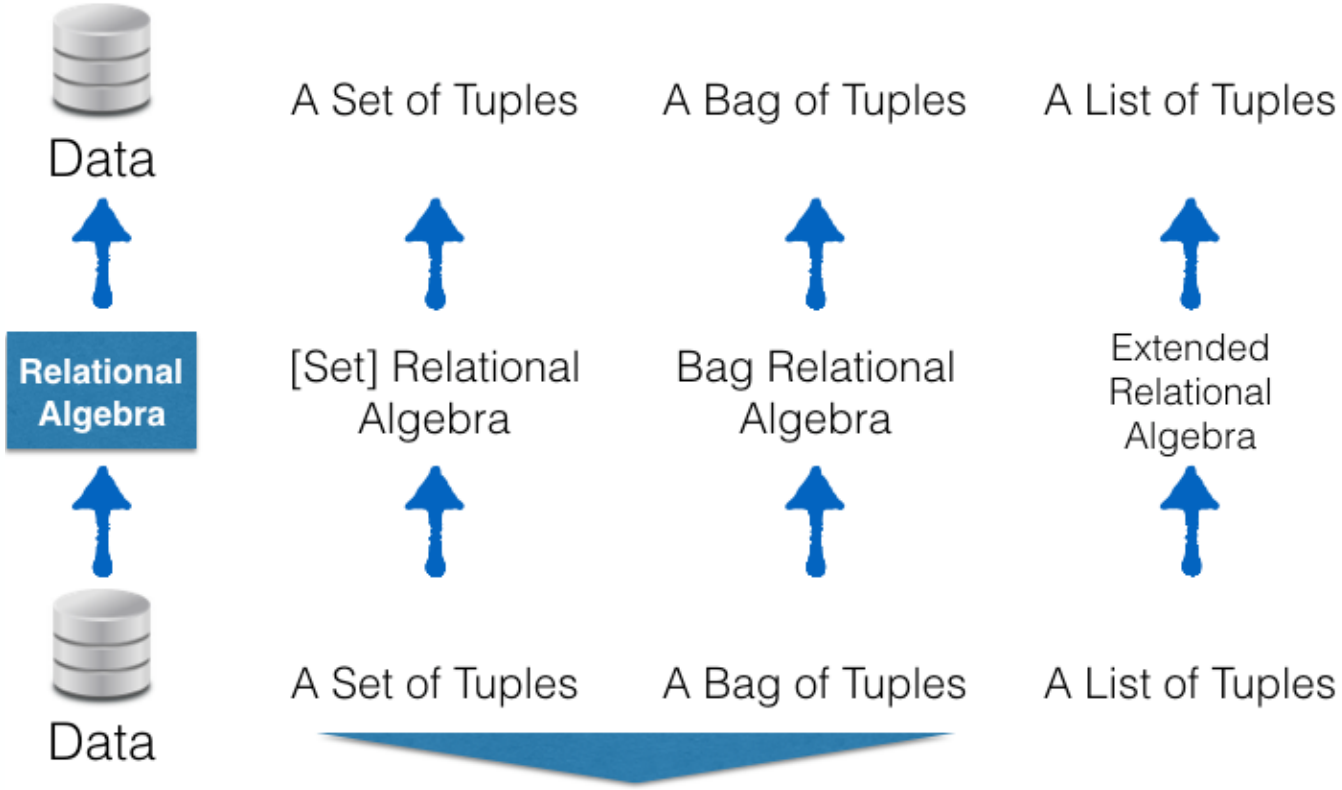
1. Imperative Languages and Declarative Languages
2. Overview of Relational Query Languages
3. Introduction to Relational Algebra
4. **Minimal Set of Relational Algebra Operators**

# Relational Algebra Operators

| Operation | Symbol | Meaning |
|---|---|---|
| Selection | σ | Selection a subset of the input tuples in a given relation |
| Projection | π | Pick wanted attributes (Delete unwanted attributes) |
| Cross-Product | x | Combine two relations |
| Renaming | ρ | Rename attribute names |
| Set-difference | - | Tuples in R1 but not in R2 |
| Union | U | Tuples either in R1 or in R2 |

- Also: Intersection ∩, **Join** ⋈, Division /
  - Not essential, but very useful
- Each operation returns a relation
- Operations are composable (RA operators are **closed**)

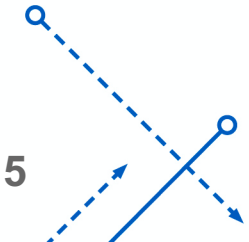
# Relational Algebra




[Graph: Dr. Oliver Kennedy CSE 562]

# Sets vs. Bags

- Rule of thumb
  - Papers: assume set semantics, i.e. Set RA
  - Implementation: assume bag semantics, i.e. Bag RA



# Recommended Reading

Database Systems: The Complete Book
Chapter 2.4, 5.2