Where we are

- Overview of Data Science
  - We found that and important aspect is Data "Munging" / Manipulation / Cleaning / Restructuring / …
- Overview of Relational Databases
  - The original problem being addressed:
  - *physical data independence*
- Secret sauce: an *algebra* of *tables*
- This will come up over and over and over….

Most slides adapted from those by

Dan Suciu and Magda Balazinska for
Introduction to Data Management (CSE 344)

at the University of Washington

Relational Algebra Operators

- Union ∪, intersection ∩, difference -
- Selection s
- Projection Π
- Join ⋈

RA

- Duplicate elimination d
- Grouping and aggregation g
- Sorting t

Extended RA

Sets v.s. Bags

- Sets: {a,b,c}, {a,d,e,f}, { }, . . .
- Bags: {a, a, b, c}, {b, b, b, b, b}, . . .

- Relational Algebra has two semantics:
- Set semantics = standard Relational Algebra
- Bag semantics = extended Relational Algebra

- Rule of thumb:
  - Every paper will assume set semantics
  - Every implementation will assume bag semantics

Union

$R1 \cup R2$

```
SELECT * FROM R1
UNION ALL
SELECT * FROM R2
```

R1

A	B
a1	b1
a2	b1

$\cup$

R2

A	B
a1	b1
a3	b4

=

$R1 \cup R2$

A	B
a1	b1
a2	b1
a3	b4
a1	b1

Difference

R1 – R2

```
SELECT * FROM R1
EXCEPT
SELECT * FROM R2
```

R1

A	B
a1	b1
a2	b1

–

R2

A	B
a1	b1
a3	b4

=

R1 – R2

A	B
a2	b1

What about Intersection ?

- Derived operator using minus

$$\boxed{\underline{R1} \cap \underline{R2} = \underline{R1} - \underline{(\underline{R1} - \underline{R2})}}$$

- Derived using join (will explain later)

$$\boxed{R1 \underline{\cap} R2 = \underline{R1 \bowtie R2}}$$

Selection

- Returns all tuples which satisfy a condition

$$\boxed{\sigma_c(R)}$$

NOT

AND

OR

- Examples
  - sSalary > 40000 (Employee)
  - sname = "Smith" (Employee)
- The condition c can be =, <, ≤, >, ≥, <>

Employee

SSN	Name	Salary
1234545	John	200000
5423341	Smith	600000
4352342	Fred	500000

$\boxed{\sigma_{Salary > \underline{40000}}\text{(Employee)}}$

SSN	Name	Salary
5423341	Smith	600000
4352342	Fred	500000