Projection

- Eliminates columns

$$\Pi_{A1,\ldots,An}(R)$$

- Example: project social-security number and names:
  - P SSN, Name (Employee)
  - Answer(SSN, Name)

Employee

SSN	Name	Salary
1234545	John	20000
5423341	John	60000
4352342	John	20000

$\Pi_{\text{Name,Salary}}$ (Employee)

Name	Salary
John	20000
John	60000
John	20000

Name	Salary
John	20000
John	60000

Set semantics

Which is more efficient?

Cross Product

- Each tuple in R1 with each tuple in R2

$$|R1 \times R2| = |R \times S|$$

- Traditionally rare in practice, but can come up in analytics
- "Find all pairs of similar images/tweets/songs"
  - Compute the cross product, then compute a similarity function $f(x_1,x_2)$ for every possible pair

Employee

Name	SSN
John	999999999
Tony	777777777

Dependent

EmpSSN	DepName
999999999	Emily
777777777	Joe

Employee ✕ Dependent

Name	SSN	EmpSSN	DepName
John	999999999	999999999	Emily
John	999999999	777777777	Joe
Tony	777777777	999999999	Emily
Tony	777777777	777777777	Joe

Equi-join

$$R1 \bowtie_{A=B} R2 = \sigma_{A=B}(R1 \times R2)$$

```
SELECT *
FROM R1, R2
WHERE R1.A = R2.B
```

```
SELECT *
FROM R1 JOIN R2
ON R1.A = R2.B
```

- Two ways to "spell" the same query
- The optimizer doesn't care about the syntax you use; it's going to work on the algebraic representation anyway.
- Sometimes one syntax or the other is more convenient.