50.043 Database and Big Data Systems

Relational Algebra

Roy Ka-Wei Lee
Assistant Professor, ISTD, SUTD

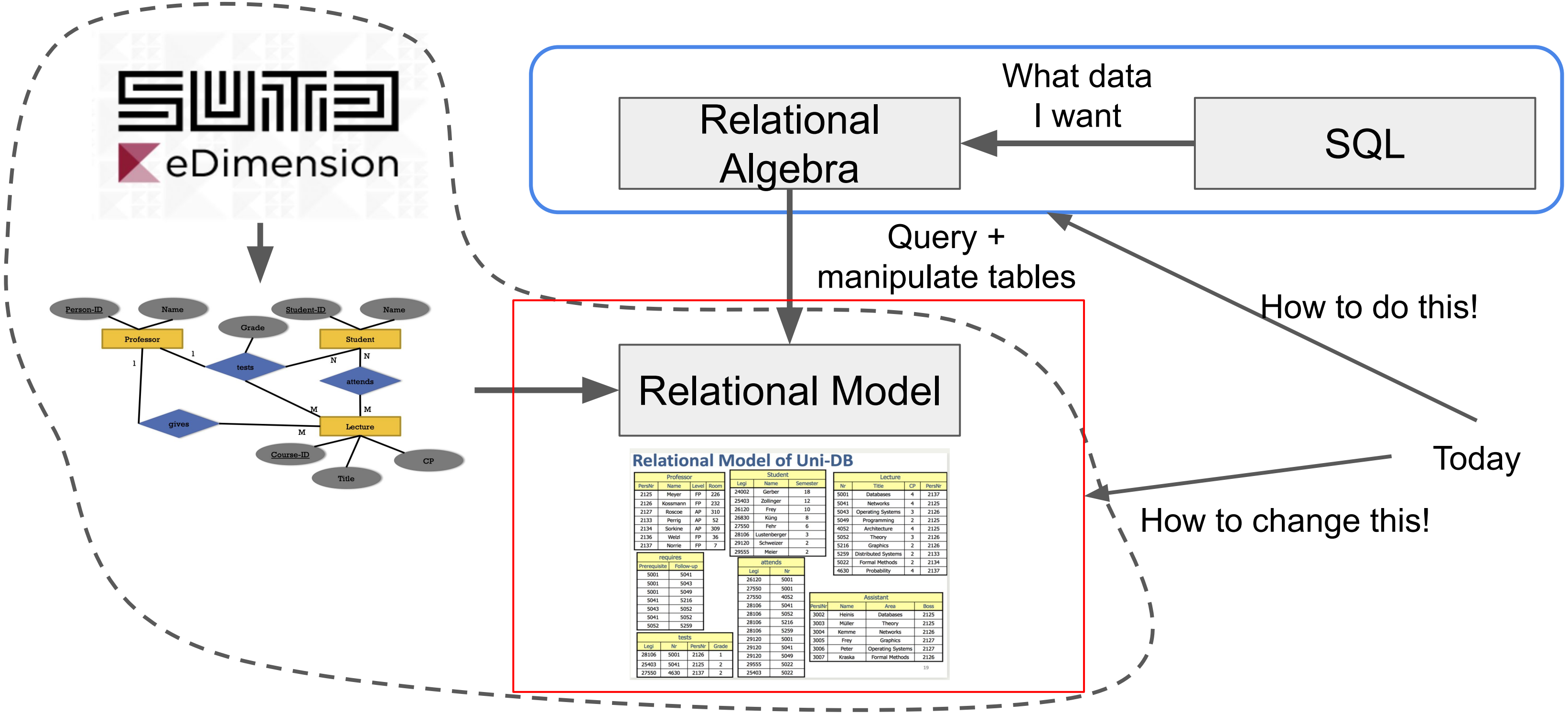
SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Learning Outcome

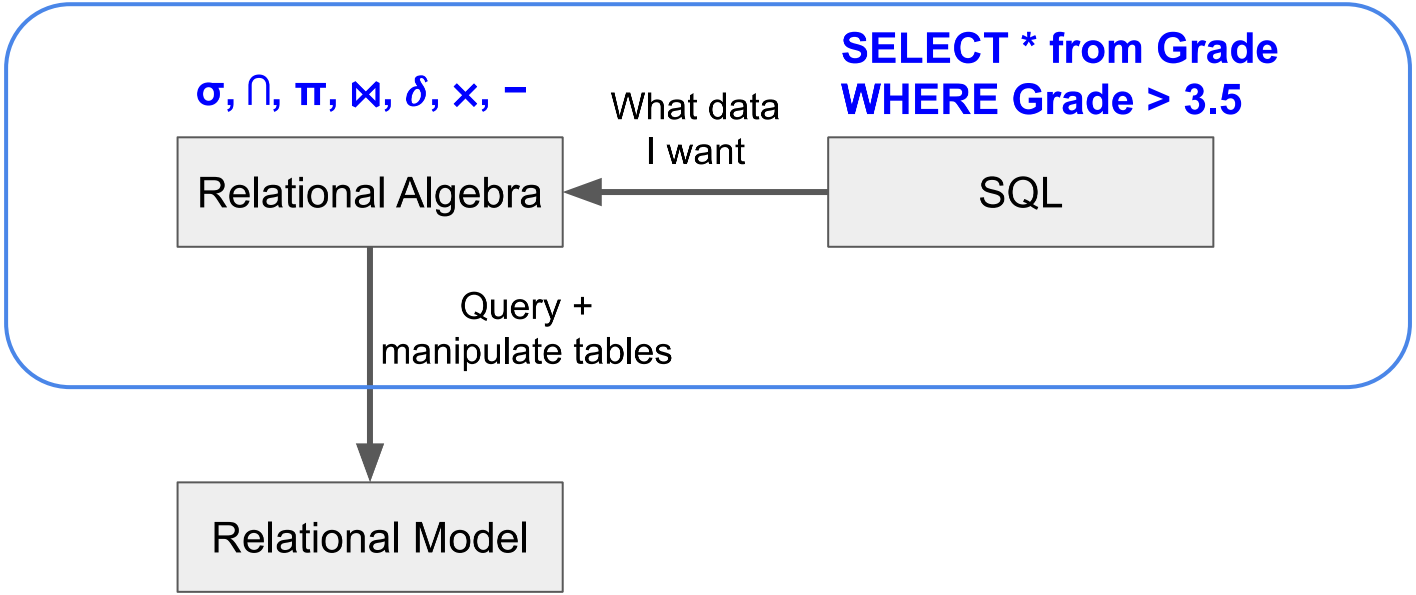
By the end of this unit, you should be able to

- Interpret relational algebra terms (queries)
- Define relational algebra terms to query a relational model

The Big Picture



Example…



σ, ∩, π, ⋈, δ, ×, −
Relational Algebra
What data I want
SELECT * from Grade WHERE Grade > 3.5
SQL
Query + manipulate tables
Relational Model

Example…

Payroll

UserID	Name	Job	Salary
123	Alice	Asst. Prof	100
456	Bob	TA	80
789	Carol	Asst. Prof	120
101	David	Prof	150

Give me salaries of all Asst. Prof

Here

UserID	Salary
123	100
789	120

Example…

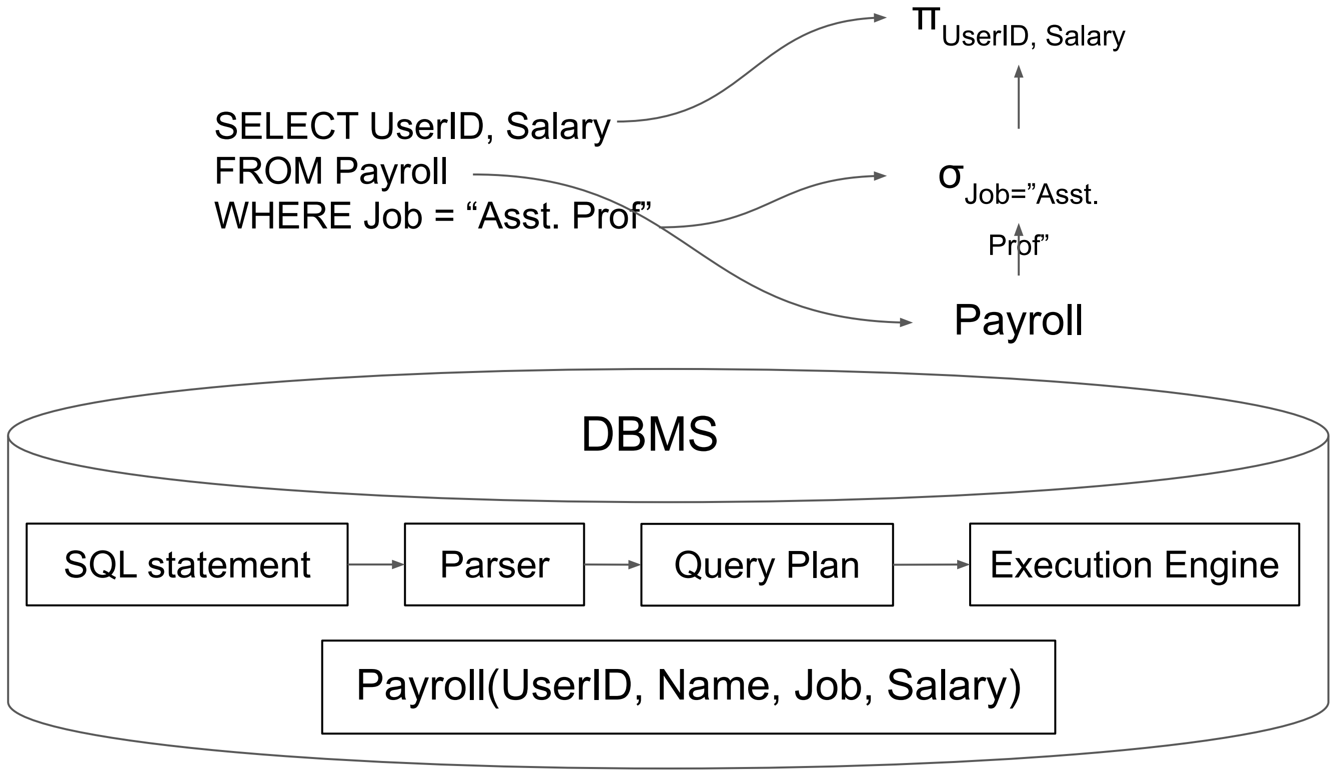
SQL query

DBMS

Payroll(**<u>UserID</u>**, Name, Job, Salary)

SELECT UserID, Salary
FROM Payroll
WHERE Job = “Asst. Prof”

Example…


SELECT UserID, Salary
FROM Payroll
WHERE Job = "Asst. Prof"
$\pi_{\text{UserID, Salary}}$
$\sigma_{\text{Job="Asst. Prof"}}$
Payroll
DBMS
SQL statement
Parser
Query Plan
Execution Engine
Payroll(UserID, Name, Job, Salary)

Example…

UserID	Name	Job	Salary
123	Alice	Asst. Prof	100
456	Bob	TA	80
789	Carol	Asst. Prof	120
101	David	Prof	150

$\pi_{\text{UserID, Salary}}$

$\uparrow$

$\sigma_{\text{Job="Asst. Prof"}}$

$\uparrow$

Payroll

Execution Engine

```
foreach row in Payroll:
  if (row.Job == "Asst. Prof")
    output (row.UseID, row.Salary)
```

UserID	Salary
123	100
789	120

Relational Algebra

- Algebra:
  - Study of symbols
    - Their meanings
    - Their relationships
- Relational Algebra
  - Study of symbols that manipulates relations
  - Symbols = **operators**

Algebra (from Arabic *"al-jabr"*, literally meaning "reunion of broken parts"[1]) is one of the broad parts of mathematics, together with number theory, geometry and analysis. In its most general form, algebra is the study of mathematical symbols and the rules for manipulating these symbols;[2] it is a unifying thread of almost all of mathematics.[3] It includes everything from elementary equation solving to the study of abstractions such as groups, rings, and

$$x^2 - 2x - 4 = 0$$

$$\sigma_{A > 3}(\pi_{X,Y}(R) \bowtie_X \pi_{X,Z}(T)) = S$$

Relational Algebra

- Contain operators to:
  - Retrieve, manipulate relations
  - Each operator:
    - Take one or more relations as input
    - Output a new relation
- Meaning of each operation (semantics)
  - Based on set
- They can be chained
  - To express complex operation

σ	Selection
π	Projection
$-$	Difference
$\cup$	Union
$\cap$	Intersection
$\bowtie$	Join
$\times$	Product
ρ	Rename
γ	Aggregation

Selection

- Syntax: $\sigma_{predicate}(R)$
- Return tuples from R satisfying the predicate
  - Like a filter
  - Predicate can be complex, with conjunction (AND) and disjunction (OR)

A	B
a1	101
a2	102
a2	103
a3	104

```
Select * from R
Where A='a2' and B > 102;
```

$\sigma_{A=\text{"a2" AND } B>102}(R)$

A	B
a2	103

Note: Selection does not correspond to SELECT in SQL, but to WHERE, Why?

Projection

- Syntax: $\pi_{A1,A2,..}(R)$
- Return tuples with the specified attributes
  - Can rearrange attribute order
  - Can transform values

```
Select B-100, A from R
Where A='a2';
```

$\pi_{B\text{-}100, A}(\sigma_{A=\text{"a2"}}(R))$

A	B
a1	101
a2	102
a2	103
a3	104

B-100	A
2	a2
3	a2

Union

- Syntax: **(R ∪ S)**
- Return tuples appearing in **any** of the two relations
  - Exactly like set union

```
(Select * from R)
UNION
(Select * from S);
```

R ∪ S

R(A,B)

A	B
a1	101
a2	102
a3	103

S(A,B)

A	B
a3	103
a4	104
a5	105

R ∪ S

A	B
a1	101
a2	102
a3	103
a4	104
a5	105

Intersection

- Syntax: **(R ∩ S)**
- Return tuples appearing in **both** relations
  - Exactly like set intersection

```
(Select * from R)
INTERSECT
(Select * from S);
```

R ∩ S

R(A,B)

A	B
a1	101
a2	102
a3	103

S(A,B)

A	B
a3	103
a4	104
a5	105

R ∩ S

A	B
a3	103

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Difference

- Syntax: **(R − S)**
- Return tuples appearing in R but not in S
  - Exactly like set difference

R(A,B)

A	B
a1	101
a2	102
a3	103

S(A,B)

A	B
a3	103
a4	104
a5	105

R − S

A	B
a1	101
a2	102

Product

- Syntax: **(R × S)**
- Return all possible combination of tuples from R and S
  - Exactly like set Cartesian product
  - R, S can have different schema

```
Select * from R cross join S;
```

R(A,B)

A	B
a1	101
a2	102

S(C,D)

C	D
a3	103
a4	104

R × S

R.A	R.B	S.C	S.D
a1	101	a3	103
a1	101	a4	104
a2	102	a3	103
a2	102	a4	104

Product

- Syntax: **(R × S)**
- Return all possible combination of tuples from R and S
  - Exactly like set Cartesian product
  - R, S can have different schema

```
Select * from R cross join S;
```

When there is no ambiguity, we drop the Relation name prefix.

R(A,B)

A	B
a1	101
a2	102

S(C,D)

C	D
a3	103
a4	104

R × S

A	B	C	D
a1	101	a3	103
a1	101	a4	104
a2	102	a3	103
a2	102	a4	104

Join

- Very important
- Already seen cross-join: ×
- We focus on three variants:
  - Inner Join (Equi-Join)
  - Natural Join
  - Left/Right/Full Outer Join

Inner Join

- Syntax: $(R \bowtie_{R.A = S.B, R.C=S.D,.} S)$
- Return tuples in R × S and satisfying a condition
  - Product followed by Selection
  - Can join on multiple columns

R(A,B,C)

A	B	C
a1	101	0
a2	102	1
a3	103	0

S(D,E,F)

D	E	F
a3	103	'a'
a1	107	'b'
a5	105	'c'

$R \bowtie_{R.A = S.D} S$

```
Select * from R, S
Where R.A = S.D;
```

R.A	R.B	R.C	S.D	S.E	S.F
a1	101	0	a1	107	'b'
a3	103	0	a3	103	'a'

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Natural Join

- Syntax: **(R ⋈ S)**
- Like Inner Join, but:
  - Detect **attributes with the same names**, then use them as join condition
  - Remove duplicate columns (same names)

```
Select * from R natural join S;
```

R(A,B,C)

A	B	C
a1	101	0
a2	102	1
a3	103	0

S(D,E,F)

D	E	F
a3	103	‘a’
a1	107	‘b’
a5	105	‘c’

R ⋈ S

R.A	R.B	R.C	S.E	S.F
a1	101	0	107	‘b’
a3	103	0	103	‘a’

Left Outer Join

- Syntax: **$(R ⟕_{R.A = S.B}\ S)$**
- Same as Inner Join, except:
  - All tuples of R appear in the result

R(A,B,C)

A	B	C
a1	101	0
a2	102	1
a3	103	0

S(D,E,F)

D	E	F
a3	103	'a'
a1	107	'b'
a5	105	'c'

$R ⟕_{R.A = S.D}\ S$

R.A	R.B	R.C	S.D	S.E	S.F
a1	101	0	a1	107	'b'
a3	103	0	a3	103	'a'
a2	102	1	*NULL*	*NULL*	*NULL*

```
Select * from R left outer
join S on R.A = S.D;
```

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Rename

- Syntax: $\rho_{R'(A1,...,An)}(R)$
- R' is the new relation name
  - $A_1,....,A_n$ are the new attribute names
  - We omit the field names if we rename only the relation name, vice versa.

R(A,B,C)

A	B	C
a1	101	0
a2	102	1
a3	103	0

S(D,E,F)

D	E	F
a3	103	'a'
a1	107	'b'
a5	105	'c'

$R \bowtie \rho_{S(A,E,F)}(S)$

R.A	R.B	R.C	S.E	S.F
a1	101	0	107	'b'
a3	103	0	103	'a'

```
Select * from R natural join
(Select D as A, E, F from S) as S2;
```

Aggregation

- Syntax: $_{A1,\ldots,An}\gamma_{F1(B1),F2(B2),\ldots}(R)$
- R' is the new relation name
  - $A_1,\ldots,A_n$ are the attributes on which to group
  - $F_1,\ldots,F_m$ are aggregation functions
    - SUM(), AVG(), MIN(), MAX(), COUNT()
  - $B_1,\ldots,B_m$ are the attributes served as arguments of the aggregation functions.

R(A,B,C)

A	B	C
a1	101	0
a2	102	1
a3	103	0

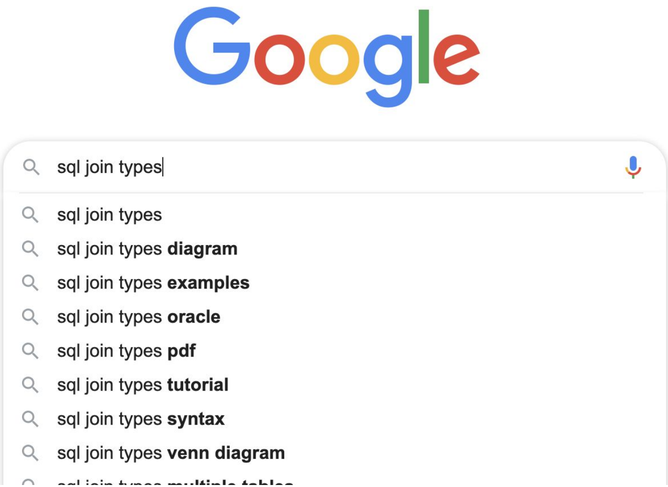
$\rho_{(C,CNT)}({}_{C}\gamma_{COUNT(B)}(R))$

C	CNT
0	2
1	1

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Other Operators

- Many other operators
- Check them out yourself



δ	Duplicate Elimination
τ	Sorting
⟖	Right Outer Join
⟗	Full Outer Join

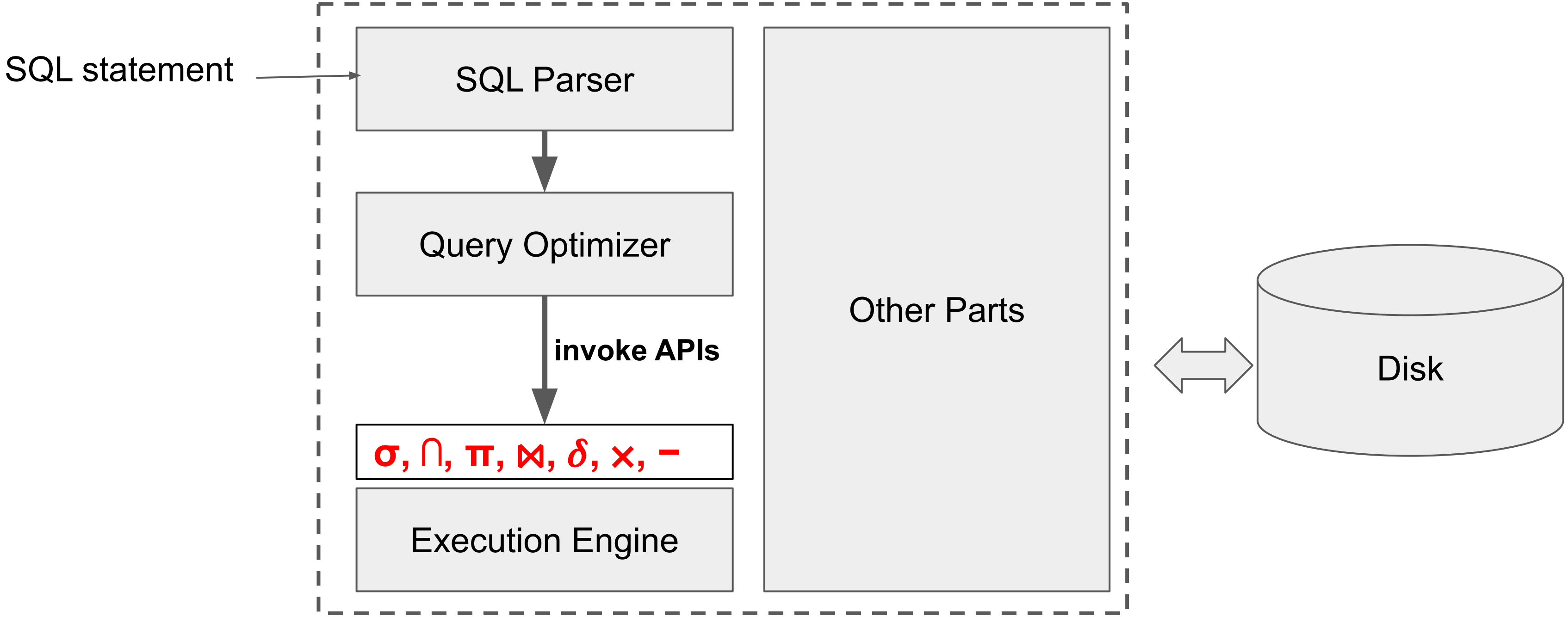
Relational Algebra

- What can you say about these two:
  - $\sigma_{S.B=102}(R \bowtie S)$
  - $R \bowtie (\sigma_{B=102}(S))$
- Given input relations, there are > 1 ways to get the desired output
- What this means?
  - Let the machine select the best way

σ	Selection
π	Projection
$-$	Difference
$\cup$	Union
$\cap$	Intersection
$\bowtie$	Join
$\times$	Product
ρ	Rename
γ	Aggregation

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Glimpse Into Database Internal



In-Class Exercise

R(A,B)

A	B
1	x
2	y
2	z
3	x
9	a

S(B,C,D)

B	C	D
x	0	3
y	2	1
y	3	3
w	3	0
y	2	0

Expression	Size of results
$R \times S$	
$R \bowtie S$	
$R \bowtie_{A=D} S$	
$\pi_B(R) - \pi_B(\sigma_{C<3}(S))$	

SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

In-Class Exercise

R(A,B)

A	B
1	x
2	y
2	z
3	x
9	a

S(B,C,D)

B	C	D
x	0	3
y	2	1
y	3	3
w	3	0
y	2	0

Expression	Size of results
$R \times S$	25
$R \bowtie S$	
$R \bowtie_{A=D} S$	
$\pi_B(R) - \pi_B(\sigma_{C<3}(S))$	

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

In-Class Exercise

R(A,B)

A	B
1	x
2	y
2	z
3	x
9	a

S(B,C,D)

B	C	D
x	0	3
y	2	1
y	3	3
w	3	0
y	2	0

Expression	Size of results
$R \times S$	25
$R \bowtie S$	5
$R \bowtie_{A=D} S$	
$\pi_B(R) - \pi_B(\sigma_{C<3}(S))$	

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

In-Class Exercise

R(A,B)

A	B
1	x
2	y
2	z
3	x
9	a

S(B,C,D)

B	C	D
x	0	3
y	2	1
y	3	3
w	3	0
y	2	0

Expression	Size of results
$R \times S$	25
$R \bowtie S$	5
$R \bowtie_{A=D} S$	3
$\pi_B(R) - \pi_B(\sigma_{C<3}(S))$	

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

In-Class Exercise

R(A,B)

A	B
1	x
2	y
2	z
3	x
9	a

S(B,C,D)

B	C	D
x	0	3
y	2	1
y	3	3
w	3	0
y	2	0

Expression	Size of results
$R \times S$	25
$R \bowtie S$	5
$R \bowtie_{A=D} S$	3
$\pi_B(R) - \pi_B(\sigma_{C<3}(S))$	2

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

What you should know?

- How to interpret relational algebra terms (queries)?
- How to define relational algebra terms to query a relational model?

Reading Resources:

- https://sutd50043.github.io/notes/l2_relational_algebra/

Please work on Cohort 2!

Acknowledgement

- *The following material have been referenced or partially used:*
  - *MIT Database Systems (6.830)*
  - *University of Washington: Introduction to Data Management (CSE344)*
  - *CMU Database Systems (15-445/645)*
  - *ETH’s Data Modeling and Databases (252-0063-00L)*
  - *ETH’s Big Data For Engineers*
  - *Yale’s Database System Concepts Seventh Edition (https://codex.cs.yale.edu/avi/courses/CS-437/slides/index.html)*