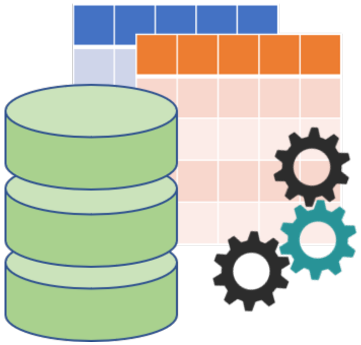
Modern Database Management

Lecture 5a – *Relational Algebra and Calculus*



Relational Algebra and Calculus

Relational Algebra and Calculus

- Relational Algebra and Relational Calculus are the formal query languages for a relational model.

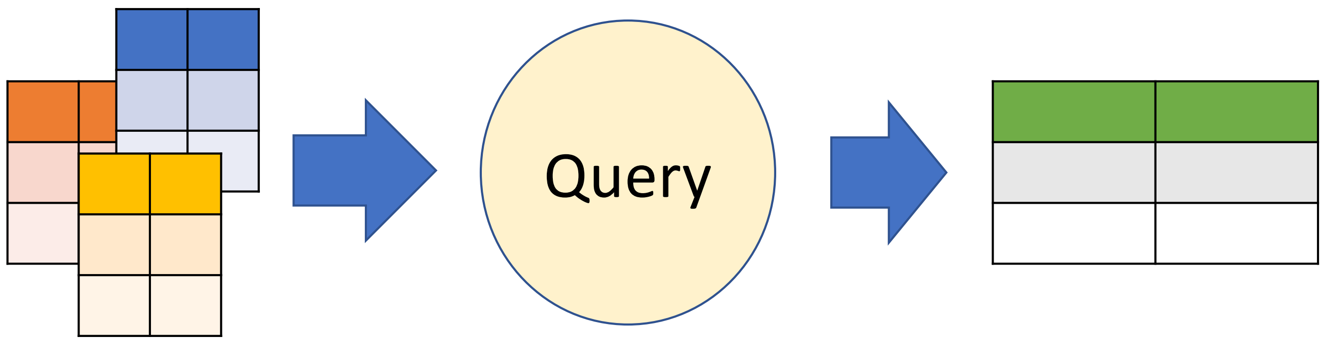
- Query languages are specialized languages for asking questions (or queries) that involve the data in a database.

- Both form the base for the SQL language which is used in most of the relational DBMSs.

Relational Queries

- Before we start, we need to clarify important points about the relational queries:

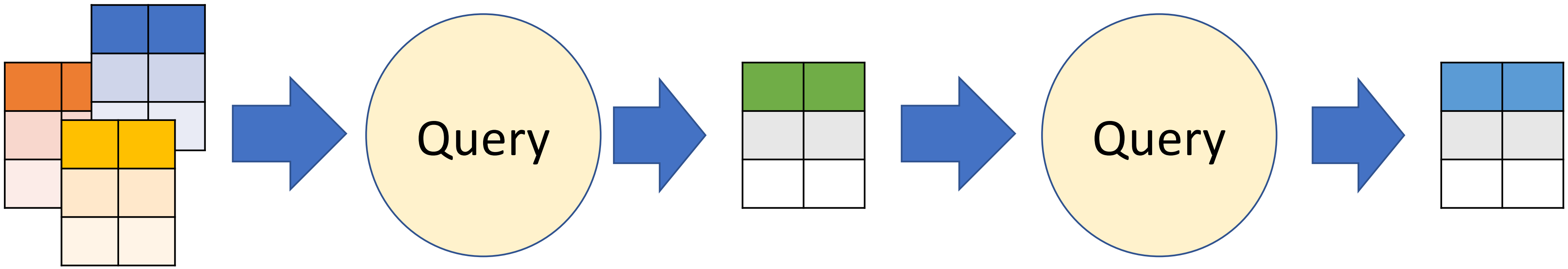
The inputs and output of a query are relations



Relational Queries

- Before we start, we need to clarify important points about the relational queries:

Queries involve the computation of intermediate results which are themselves ***relation instances***



Relational Algebra vs Relational Calculus

Relational Algebra

- ***Procedural language*** that describes the procedure to obtain the result.
- It describes the ***order of operations in the query*** that specifies how to retrieve the result.

Relational Algebra vs Relational Calculus

Relational Calculus

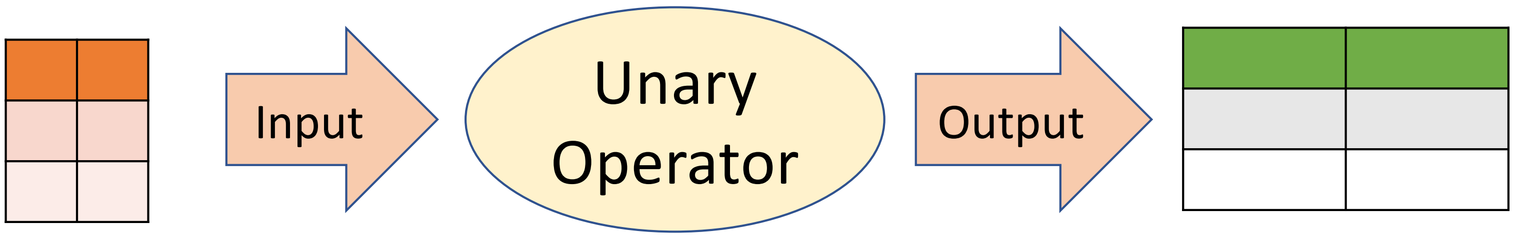
- ***Declarative language*** that defines *what* result is to be obtained.
- It ***does not specify the sequence of operations*** in which query will be evaluated.

Relational Algebra

- Relational algebra expression is a sequence of ***operations*** to build a query, through a collection of ***operators***.
- In a normal algebra we operate on number, but in relational algebra we operate on ***relations*** instead.
- The operators in any expression are either ***unary or binary operators***.

Relational Algebra

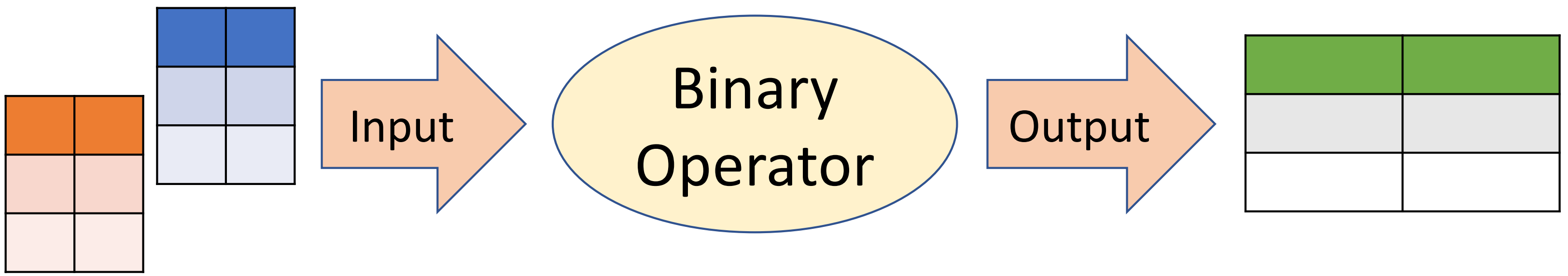
- The operators in any expression are either unary or binary operators.
- The ***unary operator*** accepts one relation as an input and produces a new relation as a result.



Relational Algebra

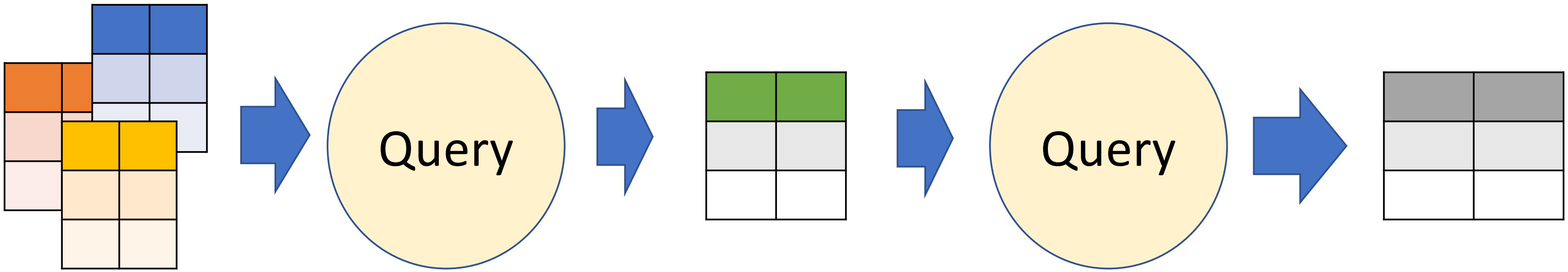
- The operators in any expression are either unary or binary operators.

- The ***binary operator*** accepts two relations as input and produces a new relation as a result.



Relational Algebra

- The result relation obtained from the expression can be further composed to other expression whose result will again be ***a new relation***.



- This property allows the ***composition of operators to form complex queries***

Selection operator - σ

- The selection operator (unary operator) returns a ***subset of tuples*** from a relation that ***satisfies certain condition***.

$$\sigma_{\text{<selection condition>}} \text{(Relation)}$$

- Think of the selection condition as the if statement in programming languages.

Selection operator - σ

$$\sigma_{\text{<selection condition>}}\text{(Relation)}$$

- The selection condition is a ***Boolean combination of terms*** with the form of:

  < Attribute > < Comparison operator > < Constant value >

  < Attribute 1 > < Comparison operator > < Attribute 2>

- The comparison operators can be: >, <, =, >=, <=, ≠

Selection operator - σ

σ _{<selection condition>} (Relation)

- The selection operator is applied independently to each ***individual tuple*** of the operand (Relation), and the tuple is selected if and only if the condition evaluates to ***TRUE***.

$$\sigma_{\text{Age} = 18}(\text{Student})$$

ID	Short name	Age	GPA
344	A J	20	3.8
342	B K	18	3.6
767	C E	20	3.2
345	D P	18	3.5
234	E U	19	3.7

- The schema of the result is the schema of the input relation instance (all the fields exist in the result, we are selecting rows/tuples)

$$\sigma_{\text{Age} = 18}\text{(Student)}$$

ID	Short name	Age	GPA
344	**A J**	**20**	**3.8**
342	**B K**	**18**	**3.6**
767	**C E**	**20**	**3.2**
345	**D P**	**18**	**3.5**
234	**E U**	**19**	**3.7**

ID	Short name	Age	GPA
342	B K	18	3.6
345	D P	18	3.5

$$\sigma_{GPA <= 3.6}(Student)$$

ID	Short name	Age	GPA
344	A J	20	3.8
342	B K	18	3.6
767	C E	20	3.2
345	D P	18	3.5
234	E U	19	3.7

ID	Short name	Age	GPA
342	B K	18	3.6
767	C E	20	3.2
345	D P	18	3.5

Selection operator - σ

$\sigma_{\text{<selection condition>}}$ (Relation)

- We can have one or more selection condition linked through Boolean operators (e.g., AND, OR, NOT)

$\sigma_{\text{< condition> Boolean operator < condition>}}$ (R)

$\sigma_{\text{GPA <= 3.6 AND Age = 20}}$ (Student)

ID	Short name	Age	GPA
344	A J	20	3.8
342	B K	18	3.6
767	C E	20	3.2
345	D P	18	3.5
234	E U	19	3.7

ID	Short name	Age	GPA
767	C E	20	3.2

Selection operator - σ - equivalence

$$\sigma_{<C2>}(\sigma_{<C1>}(R))$$

=

$$\sigma_{<C1>}(R)$$ Step 1

$$\sigma_{<C2>}(S)$$ Step 2

Selection operator - σ - equivalence

$$\sigma_{<C2>}(\sigma_{<C1>}(R))$$

=

$$\sigma_{<C1>}(\sigma_{<C2>}(R))$$

=

$$\sigma_{<C1> \text{ AND } <C2>}(R)$$

Projection operator - π

- The projection operator (unary operator) returns a ***subset of fields (attributes/columns)*** from a relation.

$$\pi_{\text{<attribute 1, attribute 2 ... attribute n>}} \text{(Relation)}$$

$$\pi_{\text{ID, Short name}}\ (\text{Relation})$$

ID	Short name	Age	GPA
344	**A J**	**20**	**3.8**
342	**B K**	**18**	**3.6**
767	**C E**	**20**	**3.2**
345	**D P**	**18**	**3.5**
234	**E U**	**19**	**3.7**

ID	Short name
344	A J
342	B K
767	C E
345	D P
234	E U

$$\pi_{\text{ID}}(\pi_{\text{ID, Short name}}(\text{Relation}))$$

ID	Short name	Age	GPA
344	**A J**	**20**	**3.8**
342	**B K**	**18**	**3.6**
767	**C E**	**20**	**3.2**
345	**D P**	**18**	**3.5**
234	**E U**	**19**	**3.7**

→

ID	Short name
344	A J
342	B K
767	C E
345	D P
234	E U

→

ID
344
342
767
345
234

π_{Age} (Relation)

ID	Short name	Age	GPA
344	A J	20	3.8
342	B K	18	3.6
767	C E	20	3.2
345	D P	18	3.5
234	E U	19	3.7

→

Age
20
18
20
18
19

- The relational model is a set-based (***no duplicate tuples allowed***)

π_{Age} (Relation)

ID	Short name	Age	GPA
344	A J	20	3.8
342	B K	18	3.6
767	C E	20	3.2
345	D P	18	3.5
234	E U	19	3.7

Age
20
18
20
18
19

Age
20
18
19

- The relational model is a set-based (***no duplicate tuples allowed***)

$$\pi_{<A1>}(\sigma_{<C1>}(R))$$

$$\sigma_{<C1>}(R)$$

$$\pi_{<A1>}(S)$$

$$(\sigma_{Age<20} (\text{Student}))$$

ID	Short name	Age	GPA
344	A J	20	3.8
342	B K	18	3.6
767	C E	20	3.2
345	D P	18	3.5
234	E U	19	3.7

ID	Short name	Age	GPA
342	B K	18	3.6
345	D P	18	3.5
234	E U	19	3.7

- The relational model is a set-based (***no duplicate tuples allowed***)

$$\pi_{\text{Short name}} (\sigma_{\text{Age<20}} (\text{Student}))$$

ID	Short name	Age	GPA
344	A J	20	3.8
342	B K	18	3.6
767	C E	20	3.2
345	D P	18	3.5
234	E U	19	3.7

ID	Short name	Age	GPA
342	B K	18	3.6
345	D P	18	3.5
234	E U	19	3.7

Short name
B K
D P
E U

- The relational model is a set-based (***no duplicate tuples allowed***)

Renaming operator - ρ

- The results of relational algebra are relations ***without names***.
- The rename operation allows us to ***rename the output relation***.

Renaming operator - ρ

- Sometimes it is necessary to use the same relation or the same attribute several times in a query, so you can use the renaming operator (unary operator).

$$\rho_{S}(R)$$

rename relation R into relation S

Renaming operator - ρ

- Sometimes it is necessary to use the same relation or the same attribute several times in a query, so you can use the renaming operator (unary operator).

$$\rho_{\text{<attribute1 new name} \leftarrow \text{attribute1 old name>}}(R)$$

Rename attribute1 of R from old name to new name

$$\rho_{\text{Student id} \leftarrow \text{ID}} \text{ Student}$$

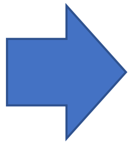
ID	Short name	Age	GPA
344	**A J**	**20**	**3.8**
342	**B K**	**18**	**3.6**
767	**C E**	**20**	**3.2**
345	**D P**	**18**	**3.5**
234	**E U**	**19**	**3.7**

Student id	Short name	Age	GPA
344	A J	20	3.8
342	B K	18	3.6
767	C E	20	3.2
345	D P	18	3.5
234	E U	19	3.7

- The relational model is a set-based (***no duplicate tuples allowed***)

$$\rho_{\text{FirstYear\_Students}} (\sigma_{\text{Age} = 18} (\text{Student}))$$

ID	Short name	Age	GPA
344	A J	20	3.8
342	B K	18	3.6
767	C E	20	3.2
345	D P	18	3.5
234	E U	19	3.7



Result: “FirstYear_Students”

Student id	Short name	Age	GPA
342	B K	18	3.6
345	D P	18	3.5

- The relational model is a set-based (***no duplicate tuples allowed***)

$$\rho_{\text{FirstYear\_Students}}\,(\sigma_{\text{Age = 18}}\,(\text{ Student }))$$

↓

$$(\sigma_{\text{GPA > 3.5}}\,(\text{ FirstYear\_Students }))$$

Cross-product
(Cartesian Product)

Cross-product (cartesian product)

- R x S returns a relation instance whose schema contains all the fields of R followed by all the fields of S – ***forming all possible combinations*** (fields of the same name are unnamed).

R

Rid	name
22	D W
31	L M
58	R S

X

S

Sid	Bid
20	109
39	102

=

Rid	name	Sid	Bid
22	D W	20	109
22	D W	39	102
31	L M	20	109
31	L M	39	102
58	R S	20	109
58	R S	39	102

Cross-product (cartesian product)

Employee

Name	SSN
John	9999
Tony	7777

X

Dependent

ESSN	DName
9999	Emily
7777	Joe

=

Name	SSN	ESSN	DName
John	9999	9999	Emily
John	9999	7777	Joe
Tony	7777	9999	Emily
Tony	7777	7777	Joe

Assume the following relations:

```
BOOKS(DocId, Title, Publisher, Year)
STUDENTS(StId, StName, Major, Age)
AUTHORS(AName, Address)
borrows(DocId, StId, Date)
has-written(DocId, AName)
describes(DocId, Keyword)
```

- *List the year and title of each book.*

  $\pi_{\text{Year, Title}}(\text{BOOKS})$

- *List all information about students whose major is CS.*

  $\sigma_{\text{Major} = \text{'CS'}}(\text{STUDENTS})$

- *List all students with the books they can borrow.*

  $\text{STUDENTS} \times \text{BOOKS}$

- *List all books published by McGraw-Hill before 1990.*

  $\sigma_{\text{Publisher} = \text{'McGraw-Hill'} \wedge \text{Year} < 1990}(\text{BOOKS})$

Relational Algebra operators from the Set theory

Relational Algebra operators from the Set theory

Union $\cup$

Intersection $\cap$

Difference —

- The input relations ***must be compatible*** (must have the same number and names of attributes – same schema)
- The result will follows the input schema
- Duplicate tuples are eliminated.

The union binary operator:

- R ∪ S returns a relation instance containing all tuples that occur in either relation instance R **or** relation instance S (or both)

A	B
1	2
1	3
2	2

∪

A	B
1	2
3	3
2	2

=

A	B
1	2
1	3
2	2
3	3

- $(R \cup S) = (S \cup R)$

Professors	**<u>Pid</u>**	**name**	**room**	**rank**
Students	**<u>Sid</u>**	**name**	**semester**	**gpa**

Find the names of all teachers and students

$$\pi_{name}(\text{Professors}) \cup \pi_{name}(\text{Students})$$

To union different schemas, rename fields

Professors	**<u>Pid</u>**	**name**	**room**	**rank**

Students	**<u>Sid</u>**	**name**	**semester**	**gpa**

Find the names and ids of all teachers and students

$$\rho_{\text{id} \leftarrow \text{Pid}} (\pi_{\text{Pid, name}} (\text{Prof})) \quad \cup \quad \rho_{\text{id} \leftarrow \text{Sid}} (\pi_{\text{Sid, name}} (\text{Stud}))$$

The intersection binary operator:

- R ∩ S returns a relation instance containing all tuples that occur in both relation instance R **and** relation instance S

A	B
1	2
1	3
2	2

∩

A	B
1	2
3	3
2	2

=

A	B
1	2
2	2

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

The Difference binary operator:

- R — S returns a relation instance containing all tuples that occur in relation instance R **but not** in relation instance S

R

A	B
1	2
1	3
2	2

—

S

A	B
1	2
3	3
2	2

=

R — S

A	B
1	3

S — R

A	B
3	3

- (R — S) $\neq$ (S — R)

The Symmetrical Difference binary operator:

$$(R \Delta S) = (R - S) \cup (S - R)$$

- **Determine all students who so far have not taken any exam**

Professors	<u>Pid</u>	name	room	rank
Students	<u>Sid</u>	name	semester	gpa
tests	<u>Sid</u>	<u>Lid</u>	<u>Pid</u>	grade

$$\pi_{\text{Sid}}\,(\text{students}) \quad - \quad \pi_{\text{Sid}}\,(\text{tests})$$

Assume the following relations:

```
BOOKS(DocId, Title, Publisher, Year)
STUDENTS(StId, StName, Major, Age)
AUTHORS(AName, Address)
borrows(DocId, StId, Date)
has-written(DocId, AName)
describes(DocId, Keyword)
```

- *List the name of students who are older than 30 and who are not studying CS.*

$$\pi_{\text{StName}}(\sigma_{\text{Age}>30}(\text{STUDENTS})) - \pi_{\text{StName}}(\sigma_{\text{Major}=\text{'CS'}}(\text{STUDENTS}))$$

Join Operator

The Join Operator

- The most used operator in the relational algebra.

- The join operator allows us to establish ***connections among data*** in different relations.

- Three main versions of the join:
  1. Natural Join
  2. Theta Join
  3. Equi Join

Natural Join Operator: ⋈

- Assume relation R has attributes A_1, … A_m, **B_1, …, B_k**
- Assume relation S has attributes **B_1, … B_k**, C_1, …, C_n

$$R \bowtie S$$

$$\pi_{A1,..,Am,\mathbf{R.B1,..,R.Bk},C1,..,Cn} (\sigma_{R.B1=S.B1 \wedge \ldots \wedge R.Bk=S.Bk} (R x S))$$

Natural Join Operator: ⋈

First step: R x S

R

A	B	C
A1	B1	C1
A2	B2	C2

X

S

C	D	E
C1	D1	E1
C3	D3	E3

=

R			**S**		
A	B	R.C	S.C	D	E
A1	B1	C1	C1	D1	E1
A1	B1	C1	C3	D3	E3
A2	B2	C2	C1	D1	E1
A2	B2	C2	C3	D3	E3

Natural Join Operator: ⋈

Second step: $\sigma_{R.C=S.C}(R x S)$

A	B	R.C	S.C	D	E
A1	**B1**	**C1**	**C1**	**D1**	**E1**
A1	B1	C1	C3	D3	E3
A2	B2	C2	C1	D1	E1
A2	B2	C2	C3	D3	E3

A	B	R.C	S.C	D	E
A1	**B1**	**C1**	**C1**	**D1**	**E1**

Natural Join Operator: ⋈

Third step: $\pi_{A,\ B,\ R.C,\ D,\ E}\ (\ \sigma_{R.C=S.C}\ (\ RxS\ )\ )$

R			S		
A	**B**	**R.C**	**S.C**	**D**	**E**
A1	**B1**	**C1**	**C1**	**D1**	**E1**

R			S	
A	**B**	**C**	**D**	**E**
A1	**B1**	**C1**	**D1**	**E1**

Natural Join Operator: ⋈

$$R \bowtie S = \pi_{A,\ B,\ R.C,\ D,\ E}\left(\sigma_{\mathbf{R.C=S.C}} \left(R x S \right) \right)$$

R

A	B	C
A1	B1	C1
A2	B2	C2

⋈

S

C	D	E
C1	D1	E1
C3	D3	E3

→

R ⋈ S

A	B	C	D	E
A1	**B1**	**C1**	**D1**	**E1**

Natural Join Operator: ⋈

$$R \bowtie S = \pi_{A,\ R.B,\ C} (\sigma_{R.B=S.B} (RxS))$$

R

A	B
X	Y
X	Z
Y	Z
Z	A

⋈

S

B	C
Z	U
A	B
Z	M

Natural Join Operator: ⋈

$$R \bowtie S = \pi_{A,\ R.B,\ C}\ (\ \sigma_{R.B=S.B}\ (\ RxS\ )\ )$$

R

A	B
X	Y
X	Z
Y	Z
Z	A

⋈

S

B	C
Z	U
A	B
Z	M

=

A	B	C
X	Z	U
X	Z	M
Y	Z	U
Y	Z	M
Z	A	B

Which lectures are held by which professors?

Professors	<u>**Pid**</u>	name	room	rank
Lectures	<u>**Lid**</u>	title	credits	Pid

Professors ⋈ Lectures

Result	Pid	name	room	rank	Lid	title	credits

Which lectures are held by which professors, in terms of the lecture title and professor name?

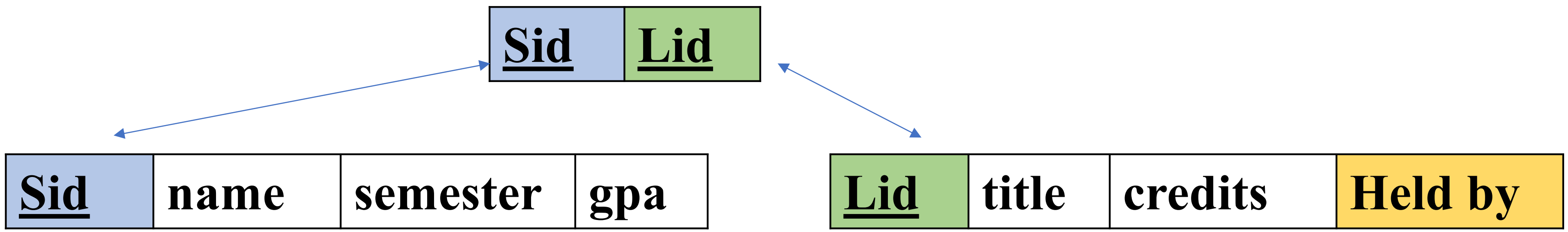
Professors	<u>Pid</u>	name	room	rank
Lectures	<u>Lid</u>	title	credits	Pid

$$\pi_{\text{name,title}} (\text{Professors} \bowtie \text{Lectures})$$

Which students attend which lectures?

Students	**<u>Sid</u>**	**name**	**semester**	**gpa**
Lectures	**<u>Lid</u>**	**title**	**credits**	**Held by**
Attends		**<u>Sid</u>**	**<u>Lid</u>**	

Which students attend which lectures?



Students ⋈ Attends ⋈ Lectures

Result

Sid	name	semester	gpa	Lid	title	credits	Held by

Theta Join Operator: $\bowtie_{condition}$

- Assume relation R has attributes $A_1, \ldots A_m$
- Assume relation S has attributes $B_1, \ldots B_k$

- The result schema has m + k attributes

$$R \bowtie_{condition} S \quad = \quad \sigma_{condition} (R \times S)$$

Theta Join Operator: $\bowtie_{condition}$

- Assume relation R has attributes $A_1, \ldots A_m$
- Assume relation S has attributes $B_1, \ldots B_k$

- The result schema has m + k attributes

$$R \bowtie_{R.Ai < S.Bj} S \quad = \quad \sigma_{R.Ai < S.Bj} (R \times S)$$

Theta Join Operator: $\bowtie_{condition}$

- Assume relation R has attributes A_1, … A_m
- Assume relation S has attributes B_1, … B_k

- The result schema has m + k attributes

$$R \bowtie_{R.Ai\ =\ S.Bj} S \quad = \quad \sigma_{R.Ai\ =\ S.Bj} (R \times S)$$

- When the condition involves equality check between certain attributes, the theta join is donated as **equi-join**

Which lectures are held by which professors?

Professors	**Pid**	name	room	rank
Lectures	**Lid**	title	credits	Held by

$$\textbf{Professors} \bowtie_{Pid = Held\_by} \textbf{Lectures}$$

OR

$$\textbf{Professors} \bowtie (\rho_{Pid \leftarrow Held\_by} \textbf{Lectures})$$

Popup quiz

Sailor	<u>Sid</u>	Sname	Rating	age

Reserve	<u>Sid</u>	<u>Bid</u>	day	month

Boat	<u>Bid</u>	Bname	Color	Size

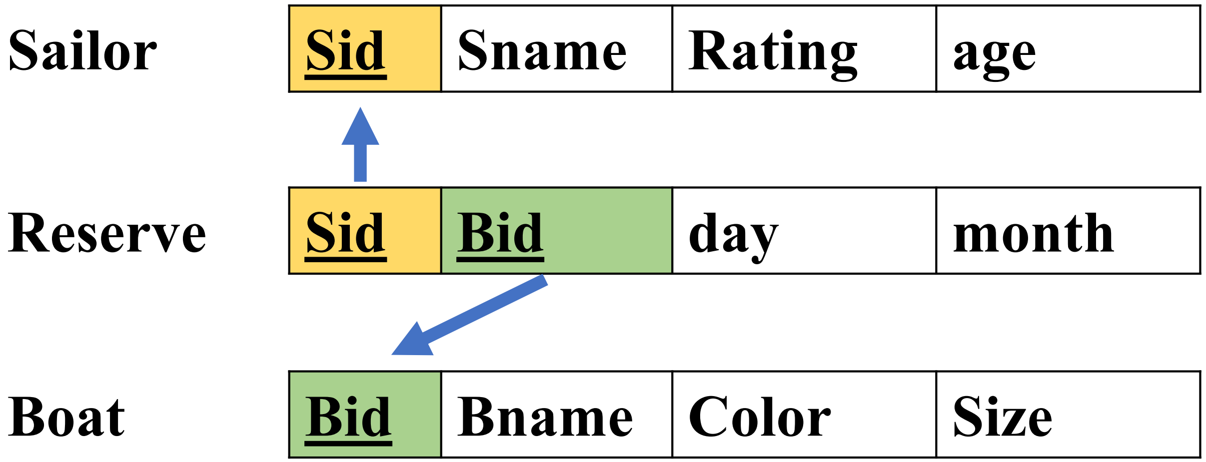
Q1: Find names of sailors who have reserved boat with id 103?

$$\pi_{Sname}(\sigma_{Bid=103}(\text{Reserve} \bowtie \text{Sailors}))$$

OR

$$\pi_{Sname}((\sigma_{Bid=103}\text{Reserve}) \bowtie \text{Sailors})$$

Popup quiz



Q2: Find names of sailors who have reserved red boat?

$$\pi_{Sname}\,(\,\sigma_{Color\,=\,\text{'red'}}\,(\,Reserve \bowtie Sailors \bowtie Boat\,)\,)$$

OR

$$\pi_{Sname}\,(\,(\sigma_{color\,=\,\text{'red'}}\,Boat) \bowtie Sailors \bowtie Reserve\,)$$

Popup quiz

Sailor	**Sid**	**Sname**	**Rating**	**age**
Reserve	**Sid**	**Bid**	**day**	**month**
Boat	**Bid**	**Bname**	**Color**	**Size**

Q3: Find the colors of boats reserved by John?

$$\pi_{Color}(\sigma_{Sname = \text{'john'}}(Sailor \bowtie Reserve \bowtie Boat))$$

OR

$$\pi_{Color}((\sigma_{Sname = \text{'john'}} Sailor) \bowtie Reserve \bowtie Boat)$$