

# Relational Algebra

# Motivation

---

We learnt how to use relations to model data

How can we retrieve (interesting) data?

We need a query language

- declarative (to allow for abstraction)
- optimisable ( ➔ less expressive than a programming language)
- relations as input and output

E.F. Codd (1970): Relational Algebra

# Characteristics of an Algebra

---

## Expressions

- are constructed with operators from atomic operands (constants, variables, ....)
- can be evaluated
- expressions can be equivalent
  - ...if they return the same result for all values of the variables

Equivalence gives rise to identities between (schemas of) expressions

The value of an expression is independent of its context

- e.g.,  $5 + 3$  has the same value, no matter whether it occurs as

$$10 - (5 + 3) \quad \text{or} \quad 4 \cdot (5 + 3)$$

*Atomic expressions:*

*numbers and variables*

*Operators:*  $+$ ,  $-$ ,  $\cdot$ ,  $:$

*Identities:*

$$x + y = y + x$$

$$x \cdot (y + z) = x \cdot y + x \cdot z$$

*... and so on*

Consequence: subexpressions can be replaced by equivalent expressions without changing the meaning of the entire expression

# Relational Algebra: Principles

---

Atoms are relations

Operators are defined for arbitrary instances of a relation

Two results have to be defined for each operator

- result schema (depending on the schemas of the argument relations)
- result instance (depending on the instances of the arguments)

Set theoretic operators

- union “ $\cup$ ”, intersection “ $\cap$ ”, difference “ $\setminus$ ”

Renaming operator  $\rho$

Removal operators

- projection  $\pi$ , selection  $\sigma$

Combination operators

- Cartesian product “ $\times$ ”, joins “ $\bowtie$ ”

# Relational Algebra

---

*Relational Algebra* is a procedural data manipulation language (**DML**).

It specifies operations on relations to define new relations:

**Unary Relational Operations:** Select, Project

**Operations from Set Theory:** Union, Intersection, Difference,  
Cartesian Product

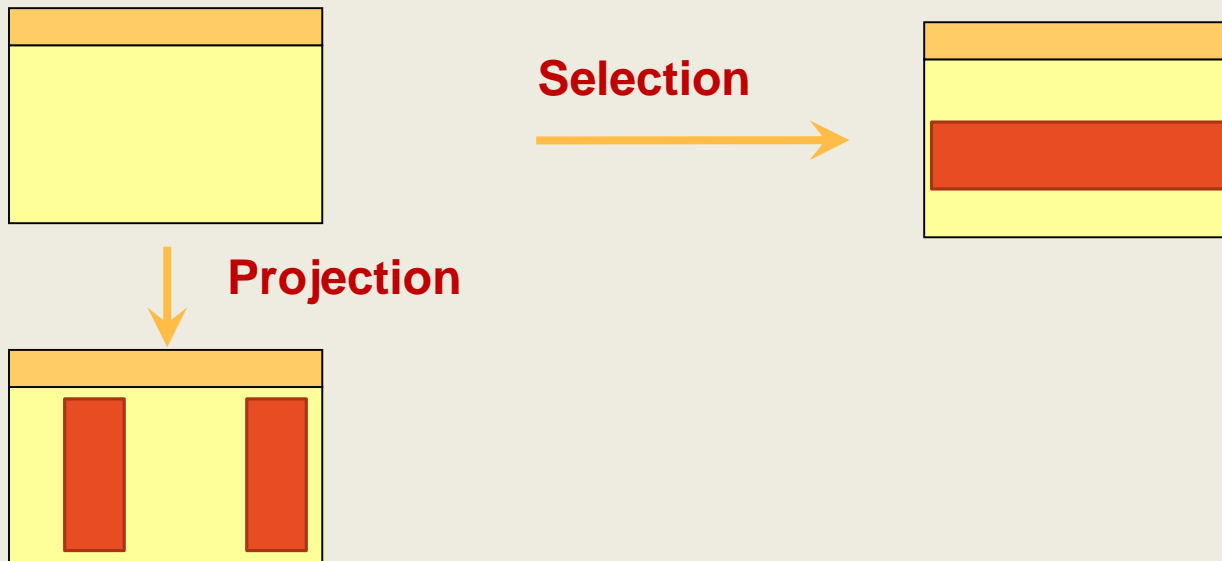
**Binary Relational Operations:** Join, Divide.

# Projection and Selection

---

Two “orthogonal” operators

- **Selection:**
  - horizontal decomposition
- **Projection:**
  - vertical decomposition



# SELECT

---

- The SELECT operation is used to choose a *subset* of the tuples (rows) from a relation that satisfies a **selection condition**, denoted by:

$$\sigma_{\langle \textit{selection condition} \rangle} (R)$$

- Result:
  - Schema: the schema of R
  - Instance: the set of all  $t \in R$  that satisfy select condition
- Intuition: Filters out all tuples that do not satisfy select condition

# Selection Conditions

---

Elementary conditions:

`<attr> op <val>      or <attr> op <attr>      or <expr> op <expr>`

where op is “=”, “<”, “≤”, (on numbers and strings)  
“LIKE” (for string comparisons),...

Example:

- `age ≤ 24`
- `phone LIKE '0039%'`
- `salary + commission ≥ 24 000`

Combined conditions (using Boolean connectives):

`C1 and C2      or   C1 or C2      or not C`



**STUDENT:**

| Person# | Name         |
|---------|--------------|
| 1       | Dr C.C.Chen  |
| 3       | Ms K.Juliff  |
| 4       | Ms J.Gledill |
| 5       | Ms B.K.Lee   |

**RESEARCHER:**

| Person# | Name             |
|---------|------------------|
| 1       | Dr C.C.Chen      |
| 2       | Dr R.G.Wilkinson |

**COURSE:**

| Department | Degree |
|------------|--------|
| Psychology | Ph.D.  |
| Comp.Sci.  | Ph.D.  |
| Comp.Sci.  | M.Sc.  |
| Psychology | M.Sc.  |

**ENROLMENT:**

| Enrolment# | Supervisee | Supervisor | Department | Degree |
|------------|------------|------------|------------|--------|
| 1          | 1          | 2          | Psychology | Ph.D.  |
| 2          | 3          | 1          | Comp.Sci.  | Ph.D.  |
| 3          | 4          | 1          | Comp.Sci.  | M.Sc.  |
| 4          | 5          | 1          | Comp.Sci.  | M.Sc.  |

## ENROLMENT:

| Enrolment# | Supervisee | Supervisor | Department | Degree |
|------------|------------|------------|------------|--------|
| 1          | 1          | 2          | Psychology | Ph.D.  |
| 2          | 3          | 1          | Comp.Sci.  | Ph.D.  |
| 3          | 4          | 1          | Comp.Sci.  | M.Sc.  |
| 4          | 5          | 1          | Comp.Sci.  | M.Sc.  |

Q: Select the enrolment records for the students whose supervisor is Person 1

$$\sigma_{(Supervisor=1)}(ENROLMENT)$$

| Enrolment# | Supervisee | Supervisor | Department | Degree |
|------------|------------|------------|------------|--------|
| 2          | 3          | 1          | Comp.Sci   | Ph.D.  |
| 3          | 4          | 1          | Comp.Sci   | M.Sc.  |
| 4          | 5          | 1          | Comp.Sci   | M.Sc.  |

## ENROLMENT:

| Enrolment# | Supervisee | Supervisor | Department | Degree |
|------------|------------|------------|------------|--------|
| 1          | 1          | 2          | Psychology | Ph.D.  |
| 2          | 3          | 1          | Comp.Sci.  | Ph.D.  |
| 3          | 4          | 1          | Comp.Sci.  | M.Sc.  |
| 4          | 5          | 1          | Comp.Sci.  | M.Sc.  |

Q: Select the enrolment records for Person 1's non-Ph.D. students

$$\sigma_{(Supervisor=1 \text{ AND } Degree \neq 'Ph.D.')} (ENROLMENT)$$

$$\sigma_{(Supervisor=1 \text{ AND } NOT Degree='Ph.D.')} (ENROLMENT)$$

| Enrolment# | Supervisee | Supervisor | Department | Degree |
|------------|------------|------------|------------|--------|
| 3          | 4          | 1          | Comp.Sci   | M.Sc.  |
| 4          | 5          | 1          | Comp.Sci   | M.Sc.  |

# Properties of SELECT

---

- Commutative:

$$\sigma_{\langle cond1 \rangle} (\sigma_{\langle cond2 \rangle} (R)) = \sigma_{\langle cond2 \rangle} (\sigma_{\langle cond1 \rangle} (R))$$

- Consecutive selects can be combined:

$$\sigma_{\langle cond1 \rangle} (\sigma_{\langle cond2 \rangle} (R)) = \sigma_{\langle cond1 \rangle \text{ AND } \langle cond2 \rangle} (R)$$

# PROJECT

---

- The PROJECT operation is used to project a subset of the attributes (column) of a relation, denoted by:
- General form:  $\pi_{\langle attribute\ list \rangle} (R)$
- Result:
  - schema: attribute list  $(A_1, \dots, A_k)$
  - instance: the set of all subtuples  $t[A_1, \dots, A_k]$  where  $t \in R$
- The PROJECT operation *removes any duplicate tuples*, so the result of the PROJECT operation is a set of distinct tuples, and this is known as **duplicate elimination**.

## ENROLMENT:

| Enrolment# | Supervisee | Supervisor | Department | Degree |
|------------|------------|------------|------------|--------|
| 1          | 1          | 2          | Psychology | Ph.D.  |
| 2          | 3          | 1          | Comp.Sci.  | Ph.D.  |
| 3          | 4          | 1          | Comp.Sci.  | M.Sc.  |
| 4          | 5          | 1          | Comp.Sci.  | M.Sc.  |

Q: Find departments and degree requirements for the courses that students enroll.

$$\pi_{\{department, degree\}}(ENROLMENT)$$

| Department | Degree |
|------------|--------|
| Psychology | Ph.D.  |
| Comp.Sci   | Ph.D.  |
| Comp.Sci   | M.Sc.  |

# Properties of PROJECT

---

- if  $\langle \text{list2} \rangle$  contains all the attributes in  $\langle \text{list1} \rangle$  then

$$\pi_{\langle \text{list1} \rangle}(\pi_{\langle \text{list2} \rangle}(R)) = \pi_{\langle \text{list1} \rangle}(R)$$

else

The operation is not well defined.

- commutes with selection:

$$\pi_X(\sigma_B(R)) = \sigma_B(\pi_X(R)) \quad (?)$$

**Commutates follows if and only if the attribute names used in SELECT is a subset of the attribute list in PROJECT**

Check the example below:

$$\pi_{\{\text{degree}\}}(\sigma_{(\text{Department}='Psychology')}(ENROLMENT)) =$$

|        |
|--------|
| Degree |
| Ph.D.  |

$$\sigma_{(\text{Department}='Psychology')}(\pi_{\{\text{degree}\}}(ENROLMENT)) = \text{Error as SELECT cannot find Department}$$

# UNION

---

- UNION is a relation that includes all tuples that are either in the left relation or in the right relation or in both relations, denoted by

$$R \cup S = \{t : t \in R \text{ or } t \in S\}$$

- Note: Union requires R and S to be **union compatible**:  
that there is a 1-1 correspondence between their attributes,  
in which corresponding attributes are over the same domain



Example:

$R1 \leftarrow \sigma_{(Supervisor=2)}(ENROLMENT)$

$R2 \leftarrow \sigma_{(Name='M.Sc')}(ENROLMENT)$

$R1 \cup R2 =$

| Enrolment# | Supervisee | Supervisor | Department | Name  |
|------------|------------|------------|------------|-------|
| 1          | 1          | 2          | Psych.     | Ph.D. |
| 3          | 4          | 1          | Comp.Sci   | M.Sc  |
| 4          | 5          | 1          | Comp.Sci   | M.Sc  |

Example:  $STUDENT \cup RESEARCHER =$

| Person# | Name             |
|---------|------------------|
| 1       | Dr C.C.Chen      |
| 3       | Ms K.Juliff      |
| 4       | Ms J.Gledhill    |
| 5       | Ms B.K.Lee       |
| 2       | Dr R.G.Wilkinson |

# INTERSECTION

---

- INTERSECTION is a relation that includes all tuples that are in both relations, denoted by

$$R \cap S = \{t : t \in R \text{ and } t \in S\}$$

- Example:

$$R_1 \leftarrow \sigma_{(Supervisor=1)}(ENROLMENT)$$

$$R_2 \leftarrow \sigma_{(Degree='Ph.D.')} (ENROLMENT)$$

$$R_1 \cap R_2 =$$

| Enrolment# | Supervisee | Supervisor | Department | Name  |
|------------|------------|------------|------------|-------|
| 2          | 3          | 1          | Comp.Sci.  | Ph.D. |

### STUDENT:

| Person# | Name         |
|---------|--------------|
| 1       | Dr C.C.Chen  |
| 3       | Ms K.Juliff  |
| 4       | Ms J.Gledill |
| 5       | Ms B.K.Lee   |

### RESEARCHER:

| Person# | Name                |
|---------|---------------------|
| 1       | Dr C.C.Chen         |
| 2       | Dr<br>R.G.Wilkinson |

Example: STUDENT  $\cap$  RESEARCHER =

| Person# | Name         |
|---------|--------------|
| 1       | Dr C.C. Chen |

# DIFFERENCE

---

- SET DIFFERENCE is a relation that includes all tuples that are in the left relation but not in the right relation, denoted by

$$R - S = \{t : t \in R \text{ and } t \notin S\}$$

- Example: STUDENT – RESEARCHER =

| Person# | Name           |
|---------|----------------|
| 3       | Ms K. Juliff   |
| 4       | Ms J. Gledhill |
| 5       | Ms B.K. Lee    |

Except Query



# Renaming

---

- The renaming operator  $\rho$  changes the name of one or more attributes
- It changes the **schema**, but **not the instance** of a relation

Father-Child

| Father  | Child |
|---------|-------|
| Adam    | Abel  |
| Adam    | Cain  |
| Abraham | Isaac |

$\rho_{\text{Parent}} \leftarrow \text{Father} (\text{Father-Child})$

| Parent  | Child |
|---------|-------|
| Adam    | Abel  |
| Adam    | Cain  |
| Abraham | Isaac |

# Exercise

STUDENT:

| Person# | Name         |
|---------|--------------|
| 1       | Dr C.C.Chen  |
| 3       | Ms K.Juliff  |
| 4       | Ms J.Gledill |
| 5       | Ms B.K.Lee   |

ENROLMENT:

| Enrolment# | Supervisee | Supervisor | Department | Degree |
|------------|------------|------------|------------|--------|
| 1          | 1          | 2          | Psychology | Ph.D.  |
| 2          | 3          | 1          | Comp.Sci.  | Ph.D.  |
| 3          | 4          | 1          | Comp.Sci.  | M.Sc.  |
| 4          | 5          | 1          | Comp.Sci.  | M.Sc.  |

RESEARCHER:

| Person# | Name             |
|---------|------------------|
| 1       | Dr C.C.Chen      |
| 2       | Dr R.G.Wilkinson |

COURSE:

| Department | Degree |
|------------|--------|
| Psychology | Ph.D.  |
| Comp.Sci.  | Ph.D.  |
| Comp.Sci.  | M.Sc.  |
| Psychology | M.Sc.  |

# Exercise

---

Write relational algebra that retrieve:

1. The names of persons who are either a student or a researcher
2. The names of persons who are a student and a researcher
3. The names of persons who are a student but not a researcher
4. The IDs of persons who are supervisors in the Computer Science Department
5. The departments and degrees of Courses which are not enrolled by any student

# Exercise Answer:

---

1. The names of persons who are either a student or a researcher

$\pi_{\{Name\}}(STUDENT \cup RESEARCHER)$

| Name             |
|------------------|
| Dr C.C.Chen      |
| Dr R.G.Wilkinson |
| Ms K.Juliff      |
| Ms J.Gledill     |
| Ms B.K.Lee       |



# Exercise Answer (continue):

---

2. The names of persons who are a student and a researcher

$$\pi_{\{Name\}}(STUDENT \cap RESEARCHER)$$

| Name        |
|-------------|
| Dr C.C.Chen |

3. The names of persons who are a student but not a researcher

$$\pi_{\{Name\}}(STUDENT - RESEARCHER)$$

| Name         |
|--------------|
| Ms K.Juliff  |
| Ms J.Gledill |
| Ms B.K.Lee   |

# Exercise Answer (continue):

---

4. The IDs of persons who are supervisors in the Computer Science Department

$$\pi_{\{Supervisor\}}(\sigma_{<Department='Comp.Sci.'>}(ENROLMENT))$$

| Supervisor |
|------------|
| 1          |

5. The departments and degrees of Courses which are not enrolled by any student

$$Course - \pi_{\{Department, Degree\}}(ENROLMENT)$$

| Department | Degree |
|------------|--------|
| Psychology | M.Sc.  |

# CARTESIAN PRODUCT

---

$$R \times S = \{t_1 || t_2 : t_1 \in R \text{ and } t_2 \in S\}$$

- Where  $t_1 || t_2$  indicates the concatenation of tuples.
- Intuition: to put together every tuple in  $R$  with every tuple in  $S$
- The number of tuples in  $R \times S : |R| * |S|$

# Cartesian Product

---

- Example: ENROLMENT X RESEARCHER =

| E'ment# | S'ee | S'or | D'ment   | Degree | Person# | Name             |
|---------|------|------|----------|--------|---------|------------------|
| 1       | 1    | 2    | Psych.   | Ph.D.  | 1       | Dr C.C. Chen     |
| 1       | 1    | 2    | Psych.   | Ph.D.  | 2       | Dr R.G.Wilkinson |
| 2       | 3    | 1    | Comp.Sci | Ph.D.  | 1       | Dr C.C. Chen     |
| 2       | 3    | 1    | Comp.Sci | Ph.D.  | 2       | Dr R.G.Wilkinson |
| 3       | 4    | 1    | Comp.Sci | M.Sc.  | 1       | Dr C.C. Chen     |
| 3       | 4    | 1    | Comp.Sci | M.Sc.  | 2       | Dr R.G.Wilkinson |
| 4       | 5    | 1    | Comp.Sci | M.Sc.  | 1       | Dr C.C. Chen     |
| 4       | 5    | 1    | Comp.Sci | M.Sc.  | 2       | Dr R.G.Wilkinson |

There are 4 tuples in ENROLMENT, 2 tuples in RESEARCHER. In the result, there are 8 tuples.

More useful is: specify the condition

$R_1 \leftarrow ENROLMENT \times RESEARCHER$

$\sigma_{(Supervisor=Person\#)}(R_1) =$

| E'ment# | S'ee | S'or | D'ment    | E'ment.<br>Name | Person# | R'cher. Name     |
|---------|------|------|-----------|-----------------|---------|------------------|
| 1       | 1    | 2    | Psych.    | Ph.D.           | 2       | Dr R.G.Wilkinson |
| 2       | 3    | 1    | Comp.Sci. | Ph.D.           | 1       | Dr C.C. Chen     |
| 3       | 4    | 1    | Comp.Sci. | M.Sc.           | 1       | Dr C.C. Chen     |
| 4       | 5    | 1    | Comp.Sci. | M.Sc.           | 1       | Dr C.C. Chen     |

Or even better: record equal attributes only once

$$R_1 \leftarrow ENROLMENT \times RESEARCHER$$

$$R_2 \leftarrow \sigma_{(Supervisor=Person\#)}(R_1)$$

$$\pi_{\{E'ment\#,S'ee,S'or,Name,D'ment,Degree\}}(R_2) =$$

| E'ment# | S'ee | S'or | Name             | D'ment    | Degree |
|---------|------|------|------------------|-----------|--------|
| 1       | 1    | 2    | Dr R.G.Wilkinson | Psych.    | Ph.D.  |
| 2       | 3    | 1    | Dr C.C. Chen     | Comp.Sci. | Ph.D.  |
| 3       | 4    | 1    | Dr C.C. Chen     | Comp.Sci. | M.Sc.  |
| 4       | 5    | 1    | Dr C.C. Chen     | Comp.Sci. | M.Sc.  |

The last of these is also known as natural join, the next to last is equi-join.

# JOIN

---

- JOIN is used to combine related tuples from two relations into single "longer" tuples.
- **Theta-join**

$$R \bowtie_{\langle \text{join condition} \rangle} S = \{t_1 || t_2 : t_1 \in R \text{ and } t_2 \in S \text{ and } \langle \text{join condition} \rangle\}$$

- A general join condition is of the form:

$\langle \text{condition} \rangle$  **AND**  $\langle \text{condition} \rangle$  **AND** ... **AND**  $\langle \text{condition} \rangle$

- where each condition is of the form  $A_i \theta B_j$ , in which  $A_i$  is an attribute of R,  $B_j$  is an attribute of S,  $A_i$  and  $B_j$  have the same domain, and  $\theta$  is a comparison operator. A JOIN operation with such a general join condition is called a **THETA JOIN**.

# JOIN: Equi-join

---

**EQUI-JOIN** is a theta-join where the only comparison operator used is “=”.

*Example:*

*ENROLMENT*  $\bowtie_{(Supervisor=Person\#)}$  *RESEARCHER*



# JOIN: Natural join

---

**NATURAL JOIN** is an equi-join which requires that the two join attributes (or each pair of join attributes) have the same name in both relations.

Question: If two relations have no join attributes, how do you define the join result? Why?

$$R(A, B) \bowtie S(B, C) \bowtie T(C, D)$$

## Notes:

1. In a natural join, there may be several pairs of join attributes.

Example:

| COURSE     |       |            |
|------------|-------|------------|
| Department | Name  | By         |
| Comp.Sci   | Ph.D. | Research   |
| Comp.Sci.  | M.Sc. | Research   |
| Psychology | M.Sc. | Coursework |

Calculate

$$ENROLMENT \bowtie_{(Department, Name), (Department, Name)} COURSE$$

2. If the pairs of joining attributes are exactly those that are identically named, we can write

$$ENROLMENT \bowtie COURSE$$

# Exercise

STUDENT:

| Person# | Name         |
|---------|--------------|
| 1       | Dr C.C.Chen  |
| 3       | Ms K.Juliff  |
| 4       | Ms J.Gledill |
| 5       | Ms B.K.Lee   |

ENROLMENT:

| Enrolment# | Supervisee | Supervisor | Department | Degree |
|------------|------------|------------|------------|--------|
| 1          | 1          | 2          | Psychology | Ph.D.  |
| 2          | 3          | 1          | Comp.Sci.  | Ph.D.  |
| 3          | 4          | 1          | Comp.Sci.  | M.Sc.  |
| 4          | 5          | 1          | Comp.Sci.  | M.Sc.  |

RESEARCHER:

| Person# | Name             |
|---------|------------------|
| 1       | Dr C.C.Chen      |
| 2       | Dr R.G.Wilkinson |

COURSE:

| Department | Degree |
|------------|--------|
| Psychology | Ph.D.  |
| Comp.Sci.  | Ph.D.  |
| Comp.Sci.  | M.Sc.  |
| Psychology | M.Sc.  |

# Exercise

---

Write relational algebra that retrieve:

1. The name of supervisor who supervises student with ID 3
2. The names of students who are studying MSc in computer science
3. The IDs of students who are supervised by Dr C.C.Chen
4. The ID of supervisor who supervises both MSc and PhD students

# Exercise Answers:

---

1. The name of supervisor who supervises student with ID 3

$$\pi_{\{Name\}}(\sigma_{<Supervisee=3>}(ENROLMENT) \bowtie_{(Supervisor),(Person\#)} RESEARCHER)$$

| Name        |
|-------------|
| Dr C.C.Chen |

2. The names of students who are studying MSc in computer science

$$\pi_{\{Name\}}(\sigma_{<Degree='M.Sc.' \text{ and } Department='Comp.Sci.'>}(ENROLMENT) \bowtie_{(Supervisee),(Person\#)} STUDENT)$$

| Name         |
|--------------|
| Ms J.Gledill |
| Ms B.K.Lee   |

# Exercise Answers (continue):

3. The IDs of students who are supervised by Dr C.C.Chen

$$\pi_{\{Supervisee\}}(ENROLMENT \bowtie_{(Supervisor),(Person\#)} \sigma_{<Name='Dr\ C.C.Chen'>(RESEARCHER))$$

| Supervisee |
|------------|
| 3          |
| 4          |
| 5          |

4. The name of supervisor who supervises both MSc and PhD students

$$\pi_{\{Name\}}(\sigma_{<Degree='M.Sc.'>(ENROLMENT) \bowtie_{(Supervisor),(Person\#)} RESEARCHER) \cap \pi_{\{Name\}}(\sigma_{<Degree='Ph.D.'>(ENROLMENT) \bowtie_{(Supervisor),(Person\#)} RESEARCHER)$$

| Name        |
|-------------|
| Dr C.C.Chen |

# DIVIDE

The DIVISION operation is applied to two Relations

$$R(Z) \div S(X)$$

Where the attributes of S are a subset of the attributes of R.

Let Y be the set of attributes of R that are not attributes of S

| R              |                |
|----------------|----------------|
| A              | B              |
| a <sub>1</sub> | b <sub>1</sub> |
| a <sub>1</sub> | b <sub>2</sub> |
| a <sub>2</sub> | b <sub>1</sub> |
| a <sub>3</sub> | b <sub>2</sub> |
| a <sub>4</sub> | b <sub>1</sub> |
| a <sub>5</sub> | b <sub>1</sub> |
| a <sub>5</sub> | b <sub>2</sub> |

| S              |
|----------------|
| B              |
| b <sub>1</sub> |
| b <sub>2</sub> |

Example:

$$X \subseteq Z$$

$$X = \{B\}, Z = \{A, B\}$$

$$\text{and } Y = Z - X = \{A\}$$

# DIVIDE

**DIVISION** is a relation  $T(Y)$  that includes a tuple  $t$  if tuples  $t_R$  appear in  $R$  with  $t_R[Y] = t$ , and with  $t_R[X] = t_S$  for every tuple  $t_S$  in  $S$ .

$$R \div S = \{t : t \times S \subseteq R\}$$

Example:

$$X = \{B\}, Z = \{A, B\}, Y = \{A\}$$

$$t_R[X] = t_S = \{b_1, b_2\}$$

In  $R$ , there are two satisfied  $t_R$  pairs:

$$\{a_1b_1, a_1b_2\} \text{ and } \{a_5b_1, a_5b_2\}$$

$$\text{So } t = t_R[Y] = \{a_1, a_5\}$$

| R              |                |
|----------------|----------------|
| A              | B              |
| a <sub>1</sub> | b <sub>1</sub> |
| a <sub>1</sub> | b <sub>2</sub> |
| a <sub>2</sub> | b <sub>1</sub> |
| a <sub>3</sub> | b <sub>2</sub> |
| a <sub>4</sub> | b <sub>1</sub> |
| a <sub>5</sub> | b <sub>1</sub> |
| a <sub>5</sub> | b <sub>2</sub> |

| S              |
|----------------|
| B              |
| b <sub>1</sub> |
| b <sub>2</sub> |

| T              |
|----------------|
| A              |
| a <sub>1</sub> |
| a <sub>5</sub> |



# DIVIDE

---

| R |                |                |
|---|----------------|----------------|
|   | A              | B              |
|   | a <sub>1</sub> | b <sub>1</sub> |
|   | a <sub>1</sub> | b <sub>2</sub> |
|   | a <sub>2</sub> | b <sub>1</sub> |
|   | a <sub>3</sub> | b <sub>2</sub> |
|   | a <sub>4</sub> | b <sub>1</sub> |
|   | a <sub>5</sub> | b <sub>1</sub> |
|   | a <sub>5</sub> | b <sub>2</sub> |

| S              |
|----------------|
| B              |
| b <sub>1</sub> |
| b <sub>2</sub> |

$$R(Z) \div S(X) =$$

| T              |
|----------------|
| A              |
| a <sub>1</sub> |
| a <sub>5</sub> |

# DIVIDE

---

**Typical use:** which courses are offered by all departments?

$$COURSE \div (\pi_{Department} COURSE)$$

# Exercise

STUDENT:

| Person# | Name         |
|---------|--------------|
| 1       | Dr C.C.Chen  |
| 3       | Ms K.Juliff  |
| 4       | Ms J.Gledill |
| 5       | Ms B.K.Lee   |

ENROLMENT:

| Enrolment# | Supervisee | Supervisor | Department | Degree |
|------------|------------|------------|------------|--------|
| 1          | 1          | 2          | Psychology | Ph.D.  |
| 2          | 3          | 1          | Comp.Sci.  | Ph.D.  |
| 3          | 4          | 1          | Comp.Sci.  | M.Sc.  |
| 4          | 5          | 1          | Comp.Sci.  | M.Sc.  |

RESEARCHER:

| Person# | Name             |
|---------|------------------|
| 1       | Dr C.C.Chen      |
| 2       | Dr R.G.Wilkinson |

COURSE:

| Department | Degree |
|------------|--------|
| Psychology | Ph.D.  |
| Comp.Sci.  | Ph.D.  |
| Comp.Sci.  | M.Sc.  |
| Psychology | M.Sc.  |

# Exercise:

---

Write relational algebra that retrieve:

1. The departments which offer all degrees

# Exercise Answers:

---

1. The departments which offer all degrees

$Course \div \pi_{\{Degree\}}(Course)$

| Department |
|------------|
| Psychology |
| Comp.Sci.  |

# Exercise

---

R:

| A              | B              | C              |
|----------------|----------------|----------------|
| a <sub>1</sub> | b <sub>1</sub> | c <sub>1</sub> |
| a <sub>1</sub> | b <sub>1</sub> | c <sub>2</sub> |
| a <sub>1</sub> | b <sub>1</sub> | c <sub>3</sub> |
| a <sub>1</sub> | b <sub>2</sub> | c <sub>2</sub> |
| a <sub>2</sub> | b <sub>1</sub> | c <sub>1</sub> |
| a <sub>2</sub> | b <sub>2</sub> | c <sub>2</sub> |
| a <sub>3</sub> | b <sub>1</sub> | c <sub>1</sub> |
| a <sub>3</sub> | b <sub>2</sub> | c <sub>1</sub> |
| a <sub>3</sub> | b <sub>2</sub> | c <sub>2</sub> |

S:

| B              | C              |
|----------------|----------------|
| b <sub>1</sub> | c <sub>1</sub> |
| b <sub>1</sub> | c <sub>2</sub> |

# Exercise:

---

Write relational algebra that retrieve:

2. Find  $A$  of  $R$  that contains all  $S$ .
3. Find  $(A, B)$  of  $R$  that contains all  $C$  of  $S$ .

# Exercise Answers:

---

2.  $R \div S$

| A              |
|----------------|
| a <sub>1</sub> |

3.  $R \div \pi_{\{c_j\}}(S)$

| A              | B              |
|----------------|----------------|
| a <sub>1</sub> | b <sub>1</sub> |
| a <sub>3</sub> | b <sub>2</sub> |



| OPERATION         | PURPOSE  | NOTATION   |
|-------------------|--|--|
| SELECT            | Selects all tuples that satisfy the selection condition from a relation R  | $\sigma_{\langle selection\ condition \rangle}(R)$ |
| PROJECT           | Produces a new relation with only some of the attributes of R, and removes duplicate tuples.   | $\pi_{\langle attribute\ list \rangle}(R)$         |
| THETA-JOIN        | Produces all combinations of tuples from R and S that satisfy the join condition.  | $R \bowtie_{\langle join\ condition \rangle} S$    |
| EQUI-JOIN         | Produces all the combinations of tuples from R and S that satisfy a join condition with only equality comparisons.   | $R \bowtie_{\langle join\ condition \rangle} S$    |
| NATURAL-JOIN      | Same as EQUIJOIN except that the join attributes of S are not included in the resulting relation; if the join attributes have the same names, they do not have to be specified at all. | $R \bowtie_{\langle join\ condition \rangle} S$    |
| UNION             | Produces a relation that includes all the tuples in R or S or both R and S; R and S must be union compatible.  | $R \cup S$   |
| INTERSECTION      | Produces a relation that includes all the tuples in both R and S; R and S must be union compatible.  | $R \cap S$   |
| DIFFERENCE        | Produces a relation that includes all the tuples in R that are not in S; R and S must be union compatible.   | $R - S$  |
| CARTESIAN PRODUCT | Produces a relation that has the attributes of R and S and includes as tuples all possible combinations of tuples from R and S.  | $R \times S$                                       |
| DIVISION          | Produces a relation T(X) that includes all tuples t[X] in R(Z) that appear in R in combination with every tuple from S(Y), where $Z = X \cup Y$ .                                      | $R(Z) \div S(Y)$                                   |

# Aggregation

---

Often, we want to retrieve aggregate values, like the “sum of salaries” of employees, or the “average age” of students.

This is achieved using aggregation functions, such as SUM, AVG, MIN, MAX, or COUNT.

Such functions are applied by the grouping and aggregation operator  $\gamma$ .

|          |   |   |                                      |        |
|----------|---|---|--------------------------------------|--------|
| If $R =$ | A | B | , then $\gamma_{\text{SUM(A)}}(R) =$ | SUM(A) |
|          | 1 | 2 |                                      | 8      |
|          | 3 | 4 |                                      |        |
|          | 3 | 5 |                                      |        |
|          | 1 | 1 |                                      |        |
|          |   |   | and $\gamma_{\text{AVG(B)}}(R) =$    | AVG(B) |
|          |   |   |                                      | 3      |

# Grouping and Aggregation

---

More often, we want to retrieve aggregate values for groups, like the “sum of employee salaries” per department, or the “average student age” per faculty.

As additional parameters, we give  $\gamma$  attributes that specify the criteria according to which the tuples of the argument are grouped.

E.g., the operator  $\gamma_{A,SUM(B)}(R)$

- partitions the tuples of  $R$  in groups that agree on  $A$ ,
- returns the sum of all  $B$  values for each group.

If  $R =$

| A | B |
|---|---|
| 1 | 2 |
| 3 | 4 |
| 3 | 5 |
| 1 | 3 |

, then  $\gamma_{A,SUM(B)}(R) =$

| A | SUM(B) |
|---|--------|
| 1 | 5      |
| 3 | 9      |

# Learning Outcome

---

- Write relational algebra expressions for given queries