Relational Algebra

Textbook: chapter 4 and 5

Motivation

- We’ve seen what a relational model is?
- We needed a formal language to specify data (tuples) from the relational model.
- Relational Algebra (E.F. Codd (1970))

Relational Algebra

- A procedural data manipulation language (DML).
- It specifies operations on relations to define new relations:
  - operators take one or two relations as inputs and produce a new relation as a result.

Six **basic** operators

- select: σ
- project: Π
- union: $\cup$
- set difference: –
- Cartesian product: $\times$
- rename: ρ

1 SELECT

The SELECT operation/predicate is used to select a subset of the tuples of a relation R, satisfying some condition.

Notation: $\sigma_{<selection\ condition>}(R)$

Intuition: Filters out all tuples that do not satisfy select condition; the result relation only keeps the relation that

Selection Condition

The condition is defined by a ***selection clause***:

- <attribute> operator <constant>
- < attribute > operator < attribute >

Where <op> is one of =, <, ≤, >, ≥ or ≠.

Example:

- age ≤ 24
- commission ≥ 24 000

Selection Condition

Selection clauses can also be

- <expression> operator < expression >

With this, we can now use ***Boolean connectives*** as operators

- C1 AND C2
- C1 OR C2
- NOT C

Terms equivalently expressed by ∧ (**and**), ∨ (**or**), ¬ (**not**).

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

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.

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

The output relation is

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.

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

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.

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

=

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

The output relation is

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

Selection

Properties:

- Consecutive selects ***can be combined***:

  σ <cond1> (σ <cond2> (R))

- Selection is a ***commutative*** operation:

  σ <cond1> (σ <cond2> (R)) is the same as…

  σ <cond2> (σ <cond1> (R))

2 PROJECT

The PROJECT operation is used to present a subset X of the attributes/column of a relation r.

Notation: $\mathbf{\Pi_{A1,A2,\ldots,Ak}(r)}$
where $A1, A2, \ldots, Ak$ are attribute names and r is a relation name.

The result relation will not contain duplicate tuples and produce a set of distinct tuples. Project performs duplicate elimination.

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

PROJECT

Consider $\pi_{<list1>}(\pi_{<list2>}(R))$

If <u><list2> contains all the attributes in <list1> </u>:

Then $\pi_{<list1>}(\pi_{<list2>}(R)) = \pi_{<list1>}(R)$

Else <u>the operation is *not well defined*</u>.

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

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.

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

The output relation is

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

Project Predicate

Properties:

Question: is projection operator commutative with selection?

i.e., πX (σB(R)) = σB (πX(R))?

Degree
Ph.D.

Consider the following:

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

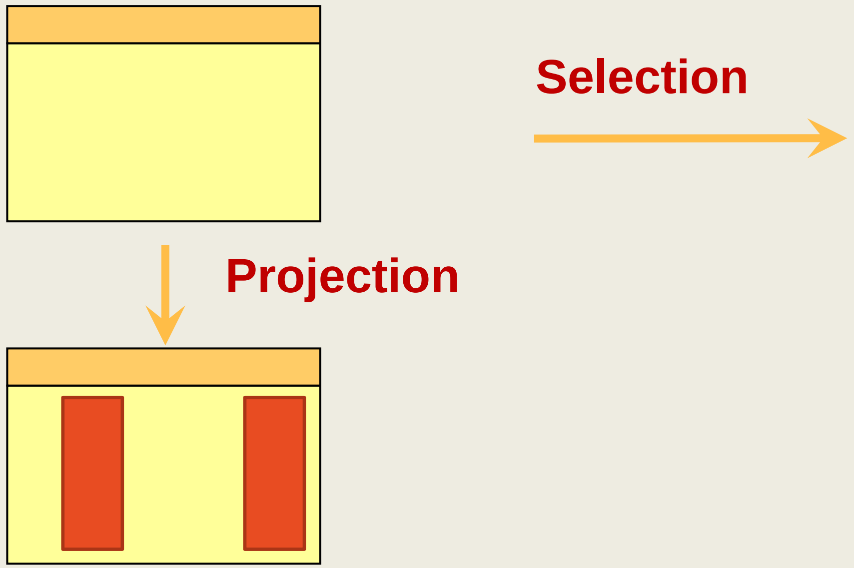
$$\sigma_{(Department='Psychology')}(\pi_{\{degree\}}(ENROLMENT))$$

Error as SELECT cannot find Department

Answer:
The attribute used in SELECT must a subset of the attribute list in PROJECT

Inutition: Projection and Selection

1. Selection performs a horizontal decomposition, and
2. projection performs a vertical decomposition



3 SET UNION

UNION Is the set theoretic union of the tuples of two relations.

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

Condition: R and S must be **union compatible!**
Union compatibility: there is a 1-1 correspondence between their attributes: the same name and the same ***domain***

Example: to find all courses taught in the Fall 2009 semester, or in the Spring 2010 semester, or in both

$$\Pi_{course\_id} (\sigma_{semester=\text{“Fall”} \;\wedge\; year=2009} (section)) \;\cup$$
$$\Pi_{course\_id} (\sigma_{semester=\text{“Spring”} \;\wedge\; year=2010} (section))$$

Example

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

Example: *STUDENT* ∪ *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

4 SET INTERSECTION

- *INTERSECTION* is an operation that includes all tuples that are in present both relations, denoted by

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

- Condition: R and S must also be **union compatible!**
- 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.

Example of Intersection

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

Example: STUDENT ∩ RESEARCHER =

Person#	Name
1	Dr C.C. Chen

5 SET DIFFERENCE

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

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

Condition: R and S must also be **union compatible!**

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

- Example: STUDENT – RESEARCHRER =

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

Summary

Operations on Relations

- *UNION*
- *INTERSECTION*
- *DIFFERENCE*

1. Binary (SET THEORETIC) operators
2. Relations must be union compatible

Express: The names of persons who are either a student or a researcher

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

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

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

Express: The names of persons who are a student and a researcher

STUDENT:

<u>Person#</u>	Name
1	Dr C.C.Chen
3	Ms K.Juliff
4	Ms J.Gledill
5	Ms B.K.Lee

RESEARCHER:

<u>Person#</u>	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

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

Name
Dr C.C.Chen

Express: The names of persons who are a student but not a researcher

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

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

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

Express: The IDs of persons who are supervisors in the Computer Science Department

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.

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

Supervisor
1

Express: The departments and degrees of Courses which are not enrolled by any student

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.

COURSE:

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

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

Department	Degree
Psychology	M.Sc.

6 JOIN

- Joins related tuples from two relations into single "longer" tuples.

$$R \bowtie_{<join\ condition>} S = \{t_1 || t_2 : t_1 \in R\ and\ t_2 \in S\ and\ < join\ condition >\}$$

- 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 θ is a ***comparison operator***.
- A JOIN operation with a join condition is called a **THETA JOIN**.

6.1 Equi-join

A type of theta-join where the only comparison operator used is "=" is called an Equi-join

Example:

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

6.2 Natural join

A type of equi-join that requires each pair of join attributes to have the same name and domain in both relations

Notes: In a natural join, there may be several valid pairs of join attributes.

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

If there are pairs of joining attributes identically named, we can write ENROLMENT⋈COURSE

JOINS

Remember the differences between the types of joins:

1. Theta JOIN
2. Equi JOIN
3. Natural JOIN

Note: all denoted with $\bowtie$

Express: The name of supervisor who supervises student with ID 3

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.

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

Name
Dr C.C.Chen

Express: What are the names of students who are studying MSc in computer science

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.

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

Name
Ms J.Gledill
Ms B.K.Lee

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

STUDENT:

<u>Person#</u>	Name
1	Dr C.C.Chen
3	Ms K.Juliff
4	Ms J.Gledill
5	Ms B.K.Lee

RESEARCHER:

<u>Person#</u>	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

COURSE:

<u>Department</u>	<u>Degree</u>
Psychology	Ph.D.
Comp.Sci.	Ph.D.
Comp.Sci.	M.Sc.
Psychology	M.Sc.

ENROLMENT:

<u>Enrolment#</u>	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.

Supervisee
3
4
5

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

Express: The ID of supervisor who supervises both MSc and PhD students

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.

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

Name
Dr C.C.Chen

CARTESIAN PRODUCT

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

- Intuition: every combination of tuples in *R* with tuples in *S*
- $t_1||t_2$ indicates the **concatenation** of tuples.
- R and S not required to be union compatible, but
- The number of tuples in the output relation is always <u>|R| * |S|</u>

Usually assumes that attributes of $r(R)$ and $s(S)$ are disjoint. (That is, $R \cap S = \emptyset$). If not, you must devise a naming schema to distinguish between the attribute names if they are the same in $r(A, B)$ and $s(A, C)$, by attaching the relation's name, $r.A$ and $s.A$ (known as dot-notation).

Example of cartesian product

- ENROLMENT X RESEARCHRER =

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	Cmp.Sci	Ph.D.	1	Dr C.C. Chen
2	3	1	Cmp.Sci	Ph.D.	2	Dr R.G.Wilkinson
3	4	1	Cmp.Sci	M.Sc.	1	Dr C.C. Chen
3	4	1	Cmp.Sci	M.Sc.	2	Dr R.G.Wilkinson
4	5	1	Cmp.Sci	M.Sc.	1	Dr C.C. Chen
4	5	1	Cmp.Sci	M.Sc.	2	Dr R.G.Wilkinson

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

Useful if we add a condition

$$R_1 \leftarrow ENROLMENT \times RESEARCHER$$

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	Cmp.Sci	Ph.D.	1	Dr C.C. Chen
2	3	1	Cmp.Sci	Ph.D.	2	Dr R.G.Wilkinson
3	4	1	Cmp.Sci	M.Sc.	1	Dr C.C. Chen
3	4	1	Cmp.Sci	M.Sc.	2	Dr R.G.Wilkinson
4	5	1	Cmp.Sci	M.Sc.	1	Dr C.C. Chen
4	5	1	Cmp.Sci	M.Sc.	2	Dr R.G.Wilkinson

In practice it's useful if we give a cartesian product specified condition

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

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

More useful if we add a projection

$$R_1 \leftarrow ENROLMENT \times RESEARCHER$$

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

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

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

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 two equal attributes occur only once

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

DIVIDE

Typical use: which courses are offered by all departments?

DIVISION outputs 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.

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

The DIVISION operation takes two Relations

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

R	
A	B
a_1	b_1
a_1	b_2
a_2	b_1
a_3	b_2
a_4	b_1
a_5	b_1
a_5	b_2

S
B
b_1
b_2

$P \div S =$

A
a_1
a_5

Exercise: The departments which offer all degrees?

COURSE:

<u>Department</u>	<u>Degree</u>
Psychology	Ph.D.
Comp.Sci.	Ph.D.
Comp.Sci.	M.Sc.
Psychology	M.Sc.

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

Department
Psychology
Comp.Sci.

Exercise:

Write **relational algebra** that retrieve:

1. Find (A, B) of *R* that has a tuple with every C in *S*.

R:

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_1	c_3
a_1	b_2	c_2
a_2	b_1	c_1
a_2	b_2	c_2
a_3	b_1	c_1
a_3	b_2	c_1
a_3	b_2	c_2

S:

B	C
b_1	c_1
b_1	c_2

$$R \div \pi_{\{c\}}(S)$$

A	B
a_1	b_1
a_3	b_2

Basic vs Extended Operators

Note: $\{\sigma, \pi, \cup, -, \times\}$ (*and rename*) are sufficient to define all these operations: this is a relationally complete set of operators. These are the ***basic operators*** of the Relational Algebra.

What about *JOIN*, *INTERSECTION* and *DIVIDE*?
They are ***extended operators*** because they can be derived by the basic operators.

E.g., We can write $r \div s$ as

$$\boxed{\begin{array}{l} temp1 \leftarrow \prod_{R-S} (r) \\ temp2 \leftarrow \prod_{R-S} ((temp1 \times s) - \prod_{R-S,S} (r)) \\ result = temp1 - temp2 \end{array}}$$

- The result to the right of $\leftarrow$ is assigned to the relation variable on the left of $\leftarrow$.
- May use variable in subsequent expressions.

Aggregate Operators

What if we want a relation with information about "sum of salaries" of employees, or the "average age" of students.

We need more expressive power, we can use ***aggregation functions*** that to summarize information from multiple tuples into ***aggregate values.***

We can use an **aggregation operator** γ and an function such as *SUM, AVG, MIN, MAX, or COUNT*. What if NULL?

If R =

A	B
1	2
3	4
3	5
1	1

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

SUM(A)
8

and $\gamma_{AVG(B)}(R)$ =

AVG(B)
3

Aggregate Operators

We can also retrieve aggregate values for groups, like the "sum of employee salaries" *per department*, or the "average student age" *per faculty*.

We give γ additional arguments to specify that the aggregation behaviour should be based on groups (defined by a set of attributes).

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

Formal Definition

A basic relational algebra expression is one of the following:

- A relation in the database
- (could also be a) constant relation

A general relational algebra expression is constructed out of smaller subexpressions. Let E_1 and E_2 be relational algebra expressions; the following are all relational-algebra expressions:

- $E_1 \cup E_2$
- $E_1 - E_2$
- $E_1 \times E_2$
- $\sigma_P(E_1)$ *where* P is a predicate on attributes in E_1
- $\Pi_S(E_1)$ where S is a set of attributes in E_1
- $\rho_X(E_1)$ *where* X is the new name for the result of E_1

OPERATION	PURPOSE	NOTATION
SELECT	Selects all tuples that satisfy the selection condition from a relation R	$\sigma_{<selection\ condition>}(R)$
PROJECT	Produces a new relation with only some of the attributes of R and removes duplicate tuples.	$\pi_{<attribute\ list>}(R)$
THETA-JOIN	Produces all combinations of tuples from R and S that satisfy the join condition.	$R \bowtie_{<join\ condition>} S$
EQUI-JOIN	Produces all the combinations of tuples from R and S that satisfy a join condition with only equality comparisons.	$R \bowtie_{<join\ condition>} S$
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.	$R \bowtie_{<join\ condition>} S$
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 ∪ Y.	$R(Z) \div S(Y)$

Rename Operator

- The ***rename*** operator ρ changes the name of one or more attributes
- Change the names in a schema
- Does not affect instance of the target relation

Family

Father	Child
Adam	Abel
Adam	Cain
Abraham	Isaac

$\rho_{\text{(Parent, Child)}}$ (Family)

Parent	Child
Adam	Abel
Adam	Cain
Abraham	Isaac

- Why might this be useful? To be included in relational algebra?