Instructions





Relational Algebra

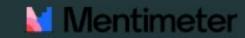
Matthew Pike & Yuan Yao

Univeristy of Nottingham Ningbo China (UNNC)



Overview

Content



- Selection
- Projection
- Product
- Join
- · Union, Intersection, Difference
- Rename

Learning Outcomes

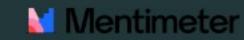


- · Understand the meaning of each operator.
- · Write relational algebra to query data in given relations.
- Calculate the data specified by a particular relational algebra.



Relational Algebra

What is the Relational Algebra?



Data Manipulation:

English ⇔ Relational Algebra ⇔ SQL queries

University

uID	uName	Country	Enrollment
	•••		•••

- English: "Find all universities with more than 20000 students."
- Relational Algebra: $\pi_{uName}(\sigma_{Enrollment} > 20000(University))$
- SQL: SELECT uName FROM University WHERE Enrollment > 20000

Relational Model and Relational Algebra



- Relational Model: Data → Relations
- Data Manipulation → operations on relations
- · Relational Algebra:
 - · A theoretical language with operations that work on relations.
 - Takes relations as input and produce new relations.
 - Operations won't affect the original relations!
 - Theoretical foundation for SQL.
- Operators: +,-, \times ,÷ for numbers, &, |, \neg for boolean
 - Common to Set-theoretic one
 - Specific to relations



Unary Operations

· What are the primary keys?

Apply			
SID	uName	Subj	Dec
0135	CAM	CS	'A'
0135	UON	CS	'A'
0423	UON	ENG	'R'

University				
uName	County	Enrollment		
UON	Nott/shire	18000		
CAM	Cam/shire	22000		
UCL	Great/Lon	20000		

Student				
SID	sName	GPA	HS	
0135	John	18.5	100	
0025	Mary	19.3	1000	
0423	Mary	17.5	300	

Example: University Applications

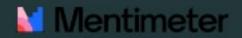
- Primary keys are:
 - uName for University.
 - SID for Student.
 - · (SID, uName, Subj) for Apply.
- I want to know the informtion of the students who has a GPA less then
 19.
 - What operator should I use?
 - How to write a query?

Apply			
SID	uName	Subj	Dec
0135	CAM	CS	'A'
0135	UON	CS	'A'
0423	UON	ENG	'R'

University				
uName	County	Enrollment		
UON	Nott/shire	18000		
CAM	Cam/shire	22000		
UCL	Great/Lon	20000		

Student				
SID	sName	GPA	HS	
0135	John	18.5	100	
0025	Mary	19.3	1000	
0423	Mary	17.5	300	

Relations: Mathematical definition



• A relation \mathcal{R} of degree n, where the domains for each attributes are D_1, \ldots, D_n , is a subset of the Cartesian product of the domains:

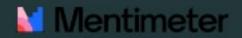
$$\mathcal{R} \subseteq D_1 \times \cdots \times D_n$$

Cartesian product:

$$D_1 \times \cdots \times D_n = \{(v_1, \dots, v_n) | v_i \in D_i\}$$

- Example:
 - if $A_1 = \{1, 2\}$ and $A_2 = \{3, 4\}$, then
 - $A_1 \times A_2 = \{(1,3), (1,4), (2,3), (2,4)\}$

Selection



- Selection works as filters.
- Let \mathcal{R} be a relation with n columns and α is some properties of tuples.
- Selection from \mathcal{R} subject to condition α is defined as:

$$\sigma_{\alpha}(\mathcal{R}) = \{(a_1, \ldots, a_n) | (a_1, \ldots, a_n) \in \mathcal{R}, \alpha(a_1, \ldots, a_n) \}$$

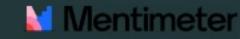
- σ : the selection operator.
- $(a_1, \ldots, a_n) \in \mathcal{R}$ means (a_1, \ldots, a_n) is a tuple in relation \mathcal{R} .

What are properties?



- Properties are expressions connected with logical symbols, i.e., and, or, not.
- Each expression is either:
 - Attributes comparisons $(=, \neq, >, <, \geq, \leq)$
 - e.g., Enrollment > HS
 - Or comparision between an attribute and a value
 - e.g., GPA > 15
- $\sigma_{GPA<19}(Student)$

Student				
SID	sName	GPA	HS	
0135	John	18.5	100	
0025	Mary	19.3	1000	
0423	Mary	17.5	300	



- Find out all students with GPA more than 19.
- Find out all students with GPA more than 18 and high school size less than 1000.
- Find out all applications to
 University of Nottingham (UoN)
 with subject CS.

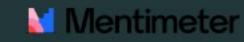
Apply			
SID	uName	Subj	Dec
0135	CAM	CS	'A'
0135	UON	CS	'A'
0423	UON	ENG	'R'

University			
uName	County	Enrollment	
UON	Nott/shire	18000	
CAM	Cam/shire	22000	
UCL	Great/Lon	20000	

Student				
SID	sName	GPA	HS	
0135	John	18.5	100	
0025	Mary	19.3	1000	
0423	Mary	17.5	300	



Excercise: Selection



Find out all students with GPA more than 19.

$$\sigma_{GPA>19}(Student)$$

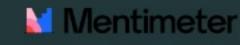
 Find out all students with GPA more than 18 and high school size less than 1000.

$$\sigma_{GPA>18}$$
 and $HS<1000$ (Student)

Find out all applications to University of Nottingham (UoN) with subject CS.

$$\sigma_{uName='UON'}$$
 and $Subj='CS'$ (Apply)

$$\sigma_{uName='UON'}(\sigma_{Subj='CS'}(Apply))$$



- Projection works as slicing.
- Let \mathcal{R} be a relation with n columns and \mathcal{X} is a set of attributes.
- Projection of \mathcal{R} on \mathcal{X} is defined as:

$$\pi_{\mathcal{X}}(R)$$

- π : the projection operator.
- $\pi_{\mathcal{X}}(R)$ generate a new relation which only contains attributes from X.

Student				
SID	sName	GPA	HS	
0135	John	18.5	100	
0025	Mary	19.3	1000	
	Mary	17.5	300	

Excercise: Projection

- Get IDs and decisions from all applications.
- Get IDs and names of students with GPA greater than 19.

Apply			
SID	uName	Subj	Dec
0135	CAM	CS	'A'
0135	UON	CS	'A'
0423	UON	ENG	'R'

University		
uName	County	Enrollment
UON	Nott/shire	18000
CAM	Cam/shire	22000
UCL	Great/Lon	20000

Student			
SID	sName	GPA	HS
0135	John	18.5	100
0025	Mary	19.3	1000
0423	Mary	17.5	300

· Get IDs and decisions from all applications.

$$\pi_{SID,Dec}(Apply)$$

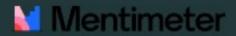
· Get IDs and names of students with GPA greater than 19.

$$\pi_{SID,SName}(\sigma_{GPA>19}(Student))$$

Can we change the order?

$$\sigma_{GPA>19}(\pi_{SID,SName}(Student))$$

Union



The definition from standard set-theory:

$$A \cup B = \{x | x \in A \text{ or } x \in B\}$$

• E.g., $\{a, b, c\} \cup \{a, d, e\} = \{a, b, c, d, e\}$

ID	Name
M139	John Smith
A368	Jane Brown
A367	David Jones

ID	Name
M140	Mary Jones
A222	Mark Brown
A367	David Jones

ID	Name
M139	John Smith
M140	Mary Jones
A368	Jane Brown
A222	Mark Brown
A367	David Jones

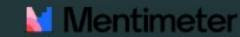
Union-compatible

• Two relations \mathcal{R}_1 and \mathcal{R}_2 are union-compatible if and only if they have the same number of attributes, and corresponding attributes have the same domain.

ID	Name
M139	John Smith
A368	Jane Brown
A367	David Jones

ID	Age
M140	23
A222	31
A367	28

Set Difference



The definition from standard set-theory:

$$A - B = \{x | x \in A \text{ and } x \notin B\}$$

- E.g., $\{a, b, c\} \{a, d, e\} = \{b, c\}$
- · Require union-compatible.

ID	Name
M139	John Smith
A368	Jane Brown
A367	David Jones

ID	Name
M140	Mary Jones
A222	Mark Brown
A367	David Jones

ID	Name
M139	John Smith
A368	Jane Brown

Intersection

The definition from standard set-theory:

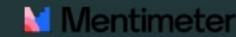
$$A \cap B = \{x | x \in A \text{ and } x \in B\}$$

- E.g., $\{a, b, c\} \cap \{a, d, e\} = \{a\}$
- · Require union-compatible.

ID	Name
M139	John Smith
A368	Jane Brown
A367	David Jones

ID	Name	
M140	Mary Jones	
A222	Mark Brown	
A367	David Jones	

ID	Name
A367	David Jones



The definition from standard set-theory:

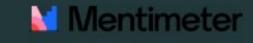
$$A \times B = \{(x, y) | x \in A, x \in B\}$$

- E.g., $\{a,b\} \times \{d,e\} = \{(a,d),(a,e),(b,d),(b,e)\}$
- Does not require union-compatible.
- Extended Cartesian product:

$$A \times B = \{(c_1, \ldots, c_n, d_1, \ldots, d_m) | (c_1, \ldots, c_n) \in A, (d_1, \ldots, d_m) \in B\}$$

Student				
SID	sName	GPA	HS	
0135	John	18.5	100	
0025	Mary	19.3	1000	
0423	Mary	17.5	300	

Apply			
SID	uName	Subj	Dec
0135	CAM	CS	'A'
0135	UON	CS	'A'
0423	UON	ENG	'R'



Student x Apply							
S.SID	sName	GPA	HS	A.SID	uName	Subj	Dec
0135	John	18.5	100	0135	CAM	CS	'A'
0135	John	18.5	100	0135	UON	CS	'A'
0135	John	18.5	100	0423	UON	ENG	'R'
0025	Mary	19.3	1000	0135	CAM	CS	'A'
0025	Mary	19.3	1000	0135	UON	CS	'A'
0025	Mary	19.3	1000	0423	UON	ENG	'R'
0423	Mary	17.5	300	0135	CAM	CS	'A'
0423	Mary	17.5	300	0135	UON	CS	'A'
0423	Mary	17.5	300	0423	UON	ENG	'R'

- What do the tuples in red mean?
- · How to solve the problem?





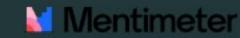
Join Operators



- Student

 Apply (bowtie)
 - Similar to Cartesian Product but enforce equality on all attributes with the same name (SID in the previous case).
 - Automatically sets values equal when attribute names are the same.
 - Get rid of multiple copies of the attributes with the same name.

Student x Apply						
SID	sName	GPA	HS	uName	Subj	Dec
0135	John	18.5	100	CAM	CS	'A'
0135	John	18.5	100	UON	CS	'A'
0423	Mary	17.5	300	UON	ENG	'R'



- Get names and GPAs of the students with high school size greater than 1000 who applied to CS and were rejected.
- Get names and GPAs of the students with high school size greater than 1000 who applied to CS at Universities with enrollment greater than 20000 and were rejected.

Apply			
SID	uName	Subj	Dec
0135	CAM	CS	'A'
0135	UON	CS	'A'
0423	UON	ENG	'R'

University				
uName	County	Enrollment		
UON	Nott/shire	18000		
CAM	Cam/shire	22000		
UCL	Great/Lon	20000		

Student				
SID	sName	GPA	HS	
0135	John	18.5	100	
0025	Mary	19.3	1000	
0423	Mary	17.5	300	

 Get names and GPAs of the students with high school size greater than 1000 who applied to CS and were rejected.

$$\pi_{GPA,sName}(\sigma_{HS>1000 \text{ and } Subj='CS' \text{ and } Dec='R'}(Student \bowtie Apply))$$

• Get names and IDs of the students who applied to CS at Universities with enrollment greater than 20000.

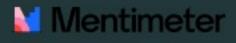
$$\pi_{SID,SName}(\sigma_{Subj='CS'})$$
 and $Enrollment>20000$ (Student \bowtie Apply

- · Cartesian Product satisfying certain properties.
- · Can be implemented via Cartesian Product and Select operator.
- If \mathcal{R}_1 and \mathcal{R}_2 are two relations, θ is a property or properties, then the Theta Join Operator is defined as:

$$\mathcal{R}_1 \bowtie_{\theta} \mathcal{R}_2 = \sigma_{\theta}(\mathcal{R}_1 \times \mathcal{R}_2)$$

- The results consists of all combinations of tuples in \mathcal{R}_1 and \mathcal{R}_2 that satisfy properties θ .
- · Can we define Natural Join based on Theta Join?

Rename Operator



- The rename operator has 3 forms. Suppose E is a relational algebra that generates a new relation, $S(T_1, \ldots, T_n)$.
 - $\rho_{R(A_1,...,A_n)}(E)$: return a relation $R(A_1,...,A_n)$
 - $\rho_R(E)$: returns a relation $R(T_1, \ldots, T_n)$
 - $\rho_{(A_1,\ldots,A_n)}(E)$: returns a relation $S(A_1,\ldots,A_n)$
- · R is the new name of the relation generated by E
- A_1, \ldots, A_n are the new names for the attributes in the new relation.

Rename Operator



Why we want a rename operator?

- Unifies schemas for Union, Difference and Intersection
 - List all Student and University names

$$\rho_{(name)}(\pi_{sName}(Student)) \cup \rho_{(name)}(\pi_{uName}(University))$$

- Help to disambiguation in self joins
 - Pairs of Universities in the same County.
 - We want to use rename and natural join

$$\sigma_{n1>n2}(\rho_{U_1(n_1,c,e_1)}(University) \bowtie \rho_{U_2(n_2,c,e_2)}(University))$$

