COL362/632 Introduction to Database Management Systems

Relational Algebra

Kaustubh Beedkar

Department of Computer Science and Engineering
Indian Institute of Technology Delhi

Relational Algebra

- If R is a relation, then

$$R' := \langle op \rangle_{\langle param \rangle}(R) \quad \text{is also a relation}$$

- If R_1 and R_2 are relations, then

$$R' := R_1 \langle op \rangle_{\langle param \rangle} R_2 \quad \text{is also a relation}$$

- Relational algebra is a procedural query language comprising relational algebra operators
- Operates on relations and produces relations
- Basis of many data processing systems — provides a simple and effective abstraction for querying and data manipulation

Selection σ

Filter tuples based on a condition

$R' := \sigma_\theta(R) \subseteq R$

- $\theta : [R] \mapsto \{ \mathit{true}, \mathit{false} \}$

Player		
player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

$\sigma_{\text{position}>5}(\text{Player})$

player_id	**name**	**position**
123	MS Dhoni	7
142	Ashok Singh	8
163	Ashok Singh	6

$\sigma_{\text{name}=\text{'Ashok Singh'}}(\text{Player})$

player_id	**name**	**position**
142	Ashok Singh	8
163	Ashok Singh	6

$\sigma_{\text{position}\leq 7 \ \wedge \ \text{player\_id}<150}(\text{Player})$

player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1

Projection π

Retrieves specific attributes

$R' := \pi_{[R']}(R)$

- $[R'] \subseteq [R]$

Player		
player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

π_{id}(Player)
player_id
123
134
131
142
154
163
250

π_{name}(Player)
name
MS Dhoni
SK Yadav
Rohit Sharma
Ashok Singh
David Warner
Chris Gayle

$\pi_{\text{name, position}}$(Player)	
name	**position**
MS Dhoni	7
SK Yadav	5
Rohit Sharma	1
Ashok Singh	8
David Warner	3
Ashok Singh	6
Chris Gayle	1

Union $\cup$

Combines tuples from two relations

$R' := R_1 \cup R_2$

- $[R'] := [R_1] = [R_2]$

Player		
player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

MIPlayer		
player_id	**name**	**position**
134	SK Yadav	5
131	Rohit Sharma	1
176	Hardik Pandaya	5

Player $\cup$ MIPlayer		
player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1
176	Hardik Pandaya	5

Difference −

Finds tuples in one relation but not the other

$R' := R_1 - R_2$

- $[R'] := [R_1] = [R_2]$

Player		
player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

MIPlayer		
player_id	**name**	**position**
134	SK Yadav	5
131	Rohit Sharma	1
176	Hardik Pandaya	5

Player − MIPlayer		
player_id	**name**	**position**
123	MS Dhoni	7
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

Crossproduct $\times$

Combines tuples from two relations

$R' := R_1 \times R_2$

- $[R'] := [R_1] \cup [R_2]$
- $|R'| = |R_1| \cdot |R_2|$

Player		
player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

League	
id	**league_name**
11	IPL
21	CPL

Player $\times$ League				
player_id	**name**	**position**	**id**	**league_name**
123	MS Dhoni	7	11	IPL
123	MS Dhoni	7	12	CPL
134	SK Yadav	5	11	IPL
134	SK Yadav	5	12	CPL
131	Rohit Sharma	1	11	IPL
131	Rohit Sharma	1	12	CPL
142	Ashok Singh	8	11	IPL
142	Ashok Singh	8	12	CPL
154	David Warner	3	11	IPL
154	David Warner	3	12	CPL
163	Ashok Singh	6	11	IPL
163	Ashok Singh	6	12	CPL
250	Chris Gayle	1	11	IPL
250	Chris Gayle	1	12	CPL

Rename ρ

Allows renaming of a relation or its attributes

$R' := \rho'_R(R)$

- $[R'] := [R]$

Player

player_id	name	position
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

$\rho_{\text{cricketers}}(\text{Player})$

player_id	name	position
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

$R' := \rho_{A' \leftarrow A}(R)$

- $A \in R$

$\rho_{\text{position} \leftarrow \text{b\_order}}(\text{Player})$

player_id	name	b_order
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

Intersection $\cap$

$R' := R_i \cap R_2$

- $= R_1 - (R_1 - R_2)$
- $[R'] := [R_1] = [R_2]$

Player		
player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

MIPlayer		
player_id	**name**	**position**
134	SK Yadav	5
131	Rohit Sharma	1
176	Hardik Pandaya	5

Player $\cap$ MIPlayer		
player_id	**name**	**position**
134	SK Yadav	5
131	Rohit Sharma	1

Theta join $\bowtie_\theta$

$R' := R_1 \bowtie_\theta R_2$

- $= \sigma_\theta(R_1 \times R_2) \subseteq R_1 \times R_2$
- $[R'] := [R_1] \cup [R_2]$
- $\theta : [R_1] \cup [R_2] \mapsto \{\text{true}, \text{false}\}$

Player		
player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

League	
id	**league_name**
150	IPL
210	CPL

Player $\bowtie_{\theta:=\text{true}}$ League				
player_id	**name**	**position**	**id**	**league_name**
123	MS Dhoni	7	150	IPL
123	MS Dhoni	7	210	CPL
134	SK Yadav	5	150	IPL
134	SK Yadav	5	210	CPL
131	Rohit Sharma	1	150	IPL
131	Rohit Sharma	1	210	CPL
142	Ashok Singh	8	150	IPL
142	Ashok Singh	8	210	CPL
154	David Warner	3	150	IPL
154	David Warner	3	210	CPL
163	Ashok Singh	6	150	IPL
163	Ashok Singh	6	210	CPL
250	Chris Gayle	1	150	IPL
250	Chris Gayle	1	210	CPL

Theta join $\bowtie_\theta$ with selective predicate

$R' := R_1 \bowtie_\theta R_2$

- $= \sigma_\theta(R_1 \times R_2) \subseteq R_1 \times R_2$
- $[R'] := [R_1] \cup [R_2]$
- $\theta : [R_1] \cup [R_2] \mapsto \{\text{true}, \text{false}\}$

Player		
player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

League	
id	**league_name**
150	IPL
210	CPL

Player $\bowtie_{\theta=\text{player\_id}>\text{id}}$ League				
player_id	**name**	**position**	**id**	**league_name**
154	David Warner	3	150	IPL
163	Ashok Singh	6	150	IPL
250	Chris Gayle	1	150	IPL
250	Chris Gayle	1	210	CPL

Equi join $\bowtie$

$R' := R_1 \bowtie_{[L][R]} R_2$

$:= R_1 \bowtie_\theta R_2$

- $\theta := R_1.[L] == R_2.[R]$ and $[L] \subseteq [R_1], [R] \subseteq [R_2]$

Player		
player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

Batter	
id	**bt_type**
123	finisher
131	opener
250	opener
272	lower order

Player $\bowtie_{\text{player\_id, id}}$ Batter				
player_id	**name**	**position**	**id**	**bt_type**
123	MS Dhoni	7	123	finisher
131	Rohit Sharma	1	131	opener
250	Chris Gayle	1	250	opener

Natural join $\bowtie$

$R' := R_1 \bowtie_{[J][J]} R_2$

$:= \pi_{[R']}(R_1 \bowtie_{[J][J]} R_2)$

- $= [J] := [R_1] \cap [R_2]$ and $[R'] := [R_1] \cup ([R_2] - [J])$

Player		
player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

Batter	
player_id	**bt_type**
123	finisher
131	opener
250	opener

Player $\bowtie$ Batter				
player_id	**name**	**position**	**player_id**	**bt_type**
123	MS Dhoni	7	123	finisher
131	Rohit Sharma	1	131	opener
250	Chris Gayle	1	259	opener

Left semi-join $\ltimes$

$$R' := R_1 \ltimes R_2$$
$$:= \pi_{[R_1]}(R_1 \bowtie R_2) \subseteq R_1$$

- $[R'] := [R_1]$

Player		
player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

Batter	
player_id	**bt_type**
123	finisher
131	opener
250	opener

Player $\ltimes$ Batter		
player_id	**name**	**position**
123	MS Dhoni	7
131	Rohit Sharma	1
250	Chris Gayle	1

Right semi-join $\rtimes$

$$R' := R_1 \rtimes R_2$$
$$:= \pi_{[R_2]}(R_1 \bowtie R_2) \subseteq R_2$$

- $[R'] := [R_2]$

Player		
player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

Batter	
player_id	**bt_type**
123	finisher
131	opener
267	middle order

Player $\rtimes$ Batter	
player_id	**bt_type**
123	finisher
131	opener

Left anti semi-join $\triangleright$

$$R' := R_1 \triangleright R_2$$
$$= R_1 - (R_1 \ltimes R_2) \subseteq R_1$$

- $[R'] := [R_1]$

Player		
player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

Batter	
player_id	**bt_type**
123	finisher
131	opener
267	middle order

Player $\triangleright$ Batter		
player_id	**name**	**position**
134	SK Yadav	5
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

Right anti semi-join $\triangleleft$

$$R' := R_1 \triangleleft R_2$$
$$= R_2 - (R_1 \rtimes R_2) \subseteq R_2$$

- $[R'] := [R_2]$

Player		
player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

Batter	
player_id	**bt_type**
123	finisher
131	opener
267	middle order

Player $\triangleleft$ Batter	
player_id	**bt_type**
267	middle order

Left outer join $\rtimes$

$$R' := R_1 \rtimes R_2$$
$$= (R_1 \bowtie R_2) \cup ((R_1 \triangleright R_2) \times R_3)$$

- where $R_3 = \{ (\omega, \ldots, \omega) \}$ and $|(\omega, \ldots, \omega)| = |[R'] - [R_1]|$
- $[R'] := [R_1 \bowtie R_2]$

Player		
player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

Batter	
player_id	**bt_type**
123	finisher
131	opener
267	middle order

Player $\rtimes$ Batter			
player_id	**name**	**position**	**bt_type**
123	MS Dhoni	7	finisher
134	SK Yadav	5	ω
131	Rohit Sharma	1	opener
142	Ashok Singh	8	ω
154	David Warner	3	ω
163	Ashok Singh	6	ω
250	Chris Gayle	1	ω

Right outer join $\bowtie\!\!\!\!\subset$

$$R' := R_1 \bowtie\!\!\!\!\subset R_2$$
$$= (R_1 \bowtie R_2) \cup (R_3 \times (R_1 \lhd R_2))$$

- where $R_3 = \{ (\omega, \ldots, \omega) \}$ and $|(\omega, \ldots, \omega)| = |[R'] - [R_2]|$
- $[R'] := [R_1 \bowtie R_2]$

Player		
player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

Batter	
player_id	**bt_type**
123	finisher
131	opener
267	middle order

Player $\bowtie\!\!\!\!\subset$ Batter			
player_id	**name**	**position**	**bt_type**
123	MS Dhoni	7	finisher
131	Rohit Sharma	1	opener
267	ω	ω	middle order

Outer join $⟗$

$$R' := R_1 ⟗ R_2$$
$$= R_1 ⟕ R_2 \cup R_1 ⟖ R_2$$

- $[R'] := [R1 ⋈ R_2]$

Player		
player_id	**name**	**position**
123	MS Dhoni	7
134	SK Yadav	5
131	Rohit Sharma	1
142	Ashok Singh	8
154	David Warner	3
163	Ashok Singh	6
250	Chris Gayle	1

Batter	
player_id	**bt_type**
123	finisher
131	opener
267	middle order

Player $⟗$ Batter			
player_id	**name**	**position**	**bt_type**
123	MS Dhoni	7	finisher
134	SK Yadav	5	ω
131	Rohit Sharma	1	opener
142	Ashok Singh	8	ω
154	David Warner	3	ω
163	Ashok Singh	6	ω
250	Chris Gayle	1	ω
267	ω	ω	middle order

Grouping and aggregation Γ

$R' := \Gamma_{[G],f_1(G_1),\ldots,f_n(G_n)}(R)$

- $[G] \subseteq [R]$ and $[G_i] \subset [R]$
- $f_i : [G_i] \mapsto D_i$

Common aggregate functions

- sum
- min
- max
- avg
- count
- count-distinct*

Player			
player_id	**name**	**team**	**runs**
123	MS Dhoni	CSK	600
134	SK Yadav	MI	400
131	Rohit Sharma	MI	500
142	Ashok Singh	CSK	100
154	David Warner	DC	150
163	Ashok Singh	CSK	250
250	Chris Gayle	RCB	300

$\Gamma_{team,\text{sum(runs)}}$(Player)	
team	**sum(runs)**
CSK	950
MI	900
DC	150
RCB	300

Generalized projections

$R' := \pi_{E_1,\ldots E_n}(R)$

- E_i is arithmetic expression involving constants and attributes of R
- Same as projection, but some operation can be "directly" expressed

Player			
player_id	**name**	**bat_first**	**bat_second**
123	MS Dhoni	1000	600
134	SK Yadav	900	400
131	Rohit Sharma	800	500
142	Ashok Singh	300	100
154	David Warner	800	150
163	Ashok Singh	100	250
250	Chris Gayle	1200	300

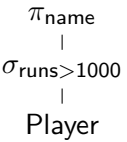
$\pi_{\text{name,bat\_first+bat\_second}}$(Player)	
name	**bat_first + bat_second**
MS Dhoni	1600
SK Yadav	1300
Rohit Sharma	1300
Ashok Singh	400
David Warner	950
Ashok Singh	350
Chris Gayle	1500

More operators

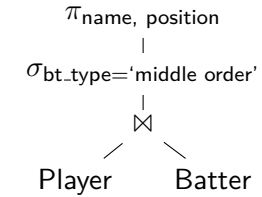
- duplicate elimination ($\delta(R)$)
- Sorting τ
- More on this later

Composing Operators

- Names of players with more than 1000 runs
  $\pi_{\text{name}}(\sigma_{\text{runs}>1000}(\text{Player}))$



- Batters and their position who play in the middle order
  $\pi_{\text{name, position}}(\sigma_{\text{bt\_type}=\text{'middle order'}}(\text{Player} \bowtie \text{Batter}))$



Examples

Students		
StudentID	**Name**	**Major**
1	Akash	CS
2	Binoy	Math
3	Charu	CS

Courses		
CourseID	**CourseName**	**InstructorID**
101	Databases	201
102	Calculus	202
103	Algorithms	201

Instructors		
InstructorID	**Name**	**Department**
201	Prof. Neetu	CS
202	Prof. Ram	Math

- Retrieve names of all students majoring in "CS"

  $\pi_{name}(\sigma_{major='CS'}(Students))$

- Find instructors who teach courses in both CS and Math departments

  $R = \pi_{\text{InstructorID}}(\sigma_{\text{Department='CS'}}(\text{Instructors}))$
  $S = \pi_{\text{InstructorID}}(\sigma_{\text{Department='Math'}}(\text{Instructors}))$
  $R \cap S$

Examples

Students		
StudentID	**Name**	**Major**
1	Akash	CS
2	Binoy	Math
3	Charu	CS

Courses		
CourseID	**CourseName**	**InstructorID**
101	Databases	201
102	Calculus	202
103	Algorithms	201

Instructors		
InstructorID	**Name**	**Department**
201	Prof. Neetu	CS
202	Prof. Ram	Math

Enrollments		
EnrollmentID	**StudentID**	**CourseID**
1	1	101
2	2	102
3	3	101
4	1	103
5	3	103

- Find all students enrolled in courses taught by "Prof. Neetu."
  1. Find the 'InstructorID' of "Prof. Neetu":
     $R_1 = \sigma_{\text{Name='Prof. Neetu'}}(\text{Instructors})$
  2. Join this with the 'Courses' table to find the courses taught by "Prof. Neetu":
     $R_2 = R_1 \bowtie_{\text{Instructors.InstructorID=Courses.InstructorID}} \text{Courses}$
  3. Join the result with the 'Enrollments' table:
     $R_3 = R_2 \bowtie_{\text{Courses.CourseID=Enrollments.CourseID}} \text{Enrollments}$
  4. Retrieve the names of the students:
     $\pi_{\text{Name}}(R_3 \bowtie_{\text{Students.StudentID=Enrollments.StudentID}} \text{Students})$

Examples

Students		
StudentID	**Name**	**Major**
1	Akash	CS
2	Binoy	Math
3	Charu	CS

Courses		
CourseID	**CourseName**	**InstructorID**
101	Databases	201
102	Calculus	202
103	Algorithms	201

Instructors		
InstructorID	**Name**	**Department**
201	Prof. Neetu	CS
202	Prof. Ram	Math

Enrollments		
EnrollmentID	**StudentID**	**CourseID**
1	1	101
2	2	102
3	3	101
4	1	103
5	3	103

- List students who are not taking any courses.
  1. Find all ‘StudentID‘ who are enrolled in courses:
     $R_1 = \pi_{\text{StudentID}}(\text{Enrollments})$
  2. Find all ‘StudentID‘ in the ‘Students‘ table:
     $R_2 = \pi_{\text{StudentID}}(\text{Students})$
  3. Use set difference to find students not enrolled in any course:
     $R_2 - R_1$

Homework

Students		
StudentID	**Name**	**Major**
1	Akash	CS
2	Binoy	Math
3	Charu	CS

Courses		
CourseID	**CourseName**	**InstructorID**
101	Databases	201
102	Calculus	202
103	Algorithms	201

Instructors		
InstructorID	**Name**	**Department**
201	Prof. Neetu	CS
202	Prof. Ram	Math

Enrollments		
EnrollmentID	**StudentID**	**CourseID**
1	1	101
2	2	102
3	3	101
4	1	103
5	3	103

- Find all courses taken by every CS major.
- Find instructors who teach at least two different courses.
- Find pairs of students who are enrolled in the same course.

Summary

- Relational algebra is a foundational tool for understanding and building queries
  - Basis of SQL
  - Basis of modern data processing systems
- Key operations
  - Selection, projection, union, set difference, cross product, joins