

Lab 3: Relational Algebra

Samantha Pope

June 6, 2024

1 Part 1

Table 1: Natural *Join* on $ColumnA(T1 \bowtie T2)$

A	Q	R	B	C
20	a	5	b	6
20	a	5	b	5

Table 2: Join T1 and T2 where $T1.Q = T2.B$

A1	Q	R	A2	B	C
20	a	5	20	b	6
20	a	5	20	b	5

Table 3: Cross Join ($T1 \times T2$)

A1	Q	R	A2	B	C
20	a	5	20	b	6
20	a	5	45	c	3
20	a	5	20	b	5
25	b	8	20	b	6
25	b	8	45	c	3
25	b	8	20	b	5
35	a	6	20	b	6
35	a	6	45	c	3
35	a	6	20	b	5

Table 4: Join T1 and T2 on A and $T1.R = T2.C$

A	Q	R	B	C
20	a	5	b	5

2 Part 2

2.1 Query 1: Players with an Elo rating of 2850 or higher

This query selects players from the **Players** relation whose Elo rating is 2850 or higher:

$$\sigma_{\text{Elo} \geq 2850}(\text{Players})$$

2.2 Query 2: Players who have played as white

This query finds all players who have played at least one game as the white player by joining the **Players** and **Games** relations on player ID:

$$\Pi_{\text{Name}}(\sigma_{\text{wpID}=\text{pID}}(\text{Players} \bowtie \text{Games}))$$

2.3 Query 3: Players who have won a game as white

This retrieves the names of players who have won a game playing as white, by checking for results where the white player was the winner ('1-0'):

$$\Pi_{\text{Name}}(\sigma_{\text{wpID}=\text{pID} \wedge \text{Result}='1-0'}(\text{Players} \bowtie \text{Games}))$$

2.4 Query 4: Players who played games in 2018

This complex query finds players who participated in any games during the year 2018 by joining the **Events**, **Games**, and **Players** relations based on event IDs and player IDs:

$$\Pi_{\text{Name}}((\sigma_{\text{Year}=2018}(\text{Events}) \bowtie_{\text{eID}=\text{eID}} \text{Games}) \bowtie_{\text{wpID}=\text{pID} \vee \text{bpID}=\text{pID}} \text{Players})$$

2.5 Query 5: Events where Magnus Carlsen lost a game

Here, the query identifies events where Magnus Carlsen lost a game playing as black, by checking the game result for '0-1':

$$\Pi_{\text{Name}, \text{Year}}((\sigma_{\text{Name}='MagnusCarlsen' \wedge \text{pID}=\text{bpID} \wedge \text{Result}='0-1'}(\text{Players} \bowtie \text{Games})) \bowtie_{\text{eID}=\text{eID}} \text{Events})$$

2.6 Query 6: Opponents of Magnus Carlsen

This query fetches names of all players who have played against Magnus Carlsen, whether he played as white or black:

$$\Pi_{\text{Name}}((\sigma_{\text{Name}='MagnusCarlsen'}(\text{Players}) \bowtie_{\text{wpID}=\text{pID} \vee \text{bpID}=\text{pID}} \text{Games}) \bowtie_{\text{wpID} \neq \text{pID} \wedge \text{bpID} \neq \text{pID}} \text{Players})$$

3 Part 3

3.1 Part 3.1

```
a) SELECT Students.SID, Students.Name
FROM Students
JOIN Enrolled ON Students.sID = Enrolled.sID
JOIN Courses ON Enrolled.cID = Courses.cID
WHERE Courses.Name = 'Databases';
```

Table 5: Part 3.1 Resulting Relation

sID	Name
1	Hermione
2	Harry

b) This query searches for the names and sID for all students enrolled in "Databases" course.

3.2 Part 3.2

a) `SELECT Courses.cID, Courses.Name`
`FROM Courses`
`JOIN Enrolled ON Courses.cID = Enrolled.cID`
`JOIN Students ON Enrolled.sID = Students.sID`
`WHERE Students.Name = 'Harry';`

Table 6: Part 3.2 Resulting Relation

cID	Name
3500	SW Practice
3810	Architecture

b) This query is having us select all the courses ID and course names that Harry is enrolled in.

3.3 Part 3.3

a)
`SELECT Students.sID, Students.Name`
`FROM Students`
`JOIN Enrolled ON Students.sID = Enrolled.sID`
`WHERE Enrolled.Grd = 'A';`

Table 7: Part 3.3 Resulting Relation

sID	Name
1	Hermione
2	Harry

b) This query searches for the names and student IDs of all students who have received a grade of 'A' in any of their courses.

4 Part 4

1. first extract all the 3000 level courses
 σ (selection) to filter 3xxx-level courses from the **Courses** table.
2. second prepare students and their courses
 π (projection) to focus on the relevant attributes.
3. third divide to find the students taking all 3xxx courses
 \div (division) to find students who are taking all specified courses.
4. fourth retrieve student names
 \bowtie (natural join) to match students' IDs with their names.