

Part 1 - Joins

1. $T1 \bowtie_{T1.A=T2.A} T2$

Schema(A, Q, R, A', B, C)

T1.A	Q	R	T2.A	B	C
20	a	5	20	b	6
20	a	5	20	b	5

2. $T1 \bowtie_{T1.Q=T2.B} T2$ // combine rows from T1 and T2 where values of Q in T1 match the values of B
 T2. Schema(A, Q, R, A', B, C)

T1.A	Q	R	T2.A	B	C
25	b	8	20	b	6
25	b	8	20	b	5

3. $T1 \bowtie T2$

Schema(A, Q, R, A', B, C)

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

4. $T1 \bowtie_{T1.A=T2.A \wedge T1.R=T2.C} T2$

Schema(A, Q, R, B, C)

T1.A	Q	R	T2.A	B	C
20	a	5	20	b	5

Part 2 - Chess Queries

- $\Pi_{Name} (\sigma_{Elo > 2849}(Players))$
- $\Pi_{Name} (\sigma_{pID \in (\Pi_{wpID}(Games))}(Players))$

3. $\Pi_{\text{Name}} (\sigma_{\text{pID} \in (\Pi_{\text{wpID}} (\sigma_{\text{Result}='1-0'}(\text{Games})))}(\text{Players}))$
4. $\Pi_{\text{Name}} (\sigma_{\text{pID} \in (\Pi_{\text{wpID}} ((\sigma_{\text{eID} \in (\Pi_{\text{eID}} (\sigma_{\text{Year}=2018}(\text{Events}))) (\text{Games})) \cup \Pi_{\text{bpID}} (\sigma_{\text{eID} \in (\Pi_{\text{eID}} (\sigma_{\text{Year}=2018}(\text{Events}))) (\text{Games})))}(\text{Players}))$
5. $\rho_{\text{Magnus}}(\sigma_{\text{Name}='MagnusCarlsen'}(\text{Players}))$
 $\rho_{\text{GamesWithLosses}}(\sigma_{(\text{wpID}=\text{Magnus.pID} \wedge \text{Result}='0-1') \vee (\text{bpID}=\text{Magnus.pID} \wedge \text{Result}='1-0')}(\text{Games}))$
 $\Pi_{\text{Events.Name, Event, Year}} (\sigma_{\text{Events.eID}=\text{GamesWithLosses.eID}}(\text{Events} \times \text{GamesWithLosses}))$
6. $\rho_{\text{Magnus}}(\sigma_{\text{Name}='MagnusCarlsen'}(\text{Players}))$
 $\rho_{\text{WhiteOpponents}}(\Pi_{\text{bpID}} (\sigma_{\text{wpID}=\text{Magnus.pID}}(\text{Games}))$
 $\rho_{\text{BlackOpponents}}(\Pi_{\text{wpID}} (\sigma_{\text{bpID}=\text{Magnus.pID}}(\text{Games}))$
 $\rho_{\text{Opponents}}(\Pi_{\text{pID}}(\text{WhiteOpponents}) \cup \Pi_{\text{pID}}(\text{BlackOpponents}))$
 $\Pi_{\text{Name}} (\sigma_{\text{pID} \in (\Pi_{\text{pID}}(\text{Opponents}))}(\text{Players}))$

Part 3 - LMS Queries

- 3.1

a) Schema (Name String)

Project name ... project student id from enrolled where their grades are not C natural join of all students —

Name
Hermione
Harry

b) The entry is filtering out all students that have a grade of C in the enrolled in table.

- 3.2

a) Schema(Name String)

Name
Hermione

b) The query is searching for the names of all students who have the same date of birth as Ron but are not named Ron.

- 3.3

a) Projects the name of result – projection of cID and sID from enrolled / projection of all students ID where result is empty. No courses have all 4 students enrolled in it. No table as a result. Schema(name String)

Searches for the names of courses in which every student is enrolled

b) The query searches for the names of courses where every student is enrolled.

Part 4

- Relational algebra that uses the divide operator to find the names of all students who are taking all of the 3xxx-level classes

$\Pi_{\text{Name}}(\sigma_{\text{sID}} \in (\Pi_{\text{sID}}(\Pi_{\text{sID, clD}}(\text{Enrolled}) / \Pi_{\text{clD}}(\sigma_{\text{clD}} > 2999 \wedge \text{clD} < 4000(\text{Courses}))))(\text{Students}))$