Aiden Pratt
6016 Lab 3

Part 1 - Joins

1.

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

2.

A1	Q	R	A2	B	C
20	b	8	20	b	6
20	b	8	20	b	5

3.

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

4.

A	Q	R	B	C
20	a	5	b	5

Aiden Pratt
6016 Lab 3

Part 2 - Chess Queries

1. π_{Name} ($_{\sigma\ Elo\ >=\ 2850}$(Players))

2. π_{Name}(Players ⋈(σ wpID > 0 (Games))

3. π_{Name} (Players ⋈ $_{Players.pID\ =\ Games.wpID}$)^$_{(Games.Result\ ==\ ‘1\text{-}0’)}$ (Games))

4. π_{Name} (Players ⋈$_{((Players.pID\ =\ wpID\ v\ (Players.pID\ =\ bpID))\wedge(Games.Year\ ==\ ‘2018)}$(Games))

5. *p*(MCID (π pID (σ $_{Name\ ==\ ‘Magnus\ Carlsen’}$ (Players)))

   π Name, Year (Event ⋈(π eID (σ (wpID = MCID ^ Result == ‘0-1’(Games))) ∪ π eID (σ (bpID = MCID ^ Result == ‘0-1’ (Games)))

find the name and year of the event. natural combine event with games. select the wpID that is equal to the projected pID of Magnus Carlson , saved in ‘MCID’. Look at the results in the Games that is ‘0-1’

6. *p*(MCID,(π pID (σ $_{Name\ ==\ ‘Magnus\ Carlsen’}$ (Players)))

   p(OpponentPlayedBlack(π $_{bpID(\sigma\ wpID\ =\ MCID}$ (Games))

   p(OpponentPlayedWhite(π $_{wpID(\sigma\ bpID\ =\ MCID}$(Games))

   p(AllOpponentIDs ($_{OpponentPlayedBlack\ \cup\ OpponentPlayedWhite}$))

   π Name(Players ⋈ (π AllOpponentIDs))

Aiden Pratt
6016 Lab 3

Part 3 - LMS Queries

3.1

- $\rho(C, \pi_{sid}(\sigma_{Grd=C}(Enrolled)))$
- $\pi_{Name}((\pi_{sid}(Enrolled) - C) \bowtie Students)$

Name
Hermoine
Harry

pull out the student IDs of the students who got a C. Using the rename symbol to call this C. Do the difference operator with the set of data in Enrolled. This removes the student IDs which have a Grd of C. Then naturally join with Students to display the names of students who have not received a C Grd.

3.2

- $\rho(S1, Students)$
- $\rho(S2, Students)$
- $\pi_{S2.Name}(\sigma_{S1.Name==Ron \wedge S1.DOB==S2.DOB \wedge S2.name!=Ron}(S1 \times S2))$

Name
Hermoine

rename Students to S1 , then rename Students to S2, creating two copies of students. Filter the names in S1 where the name is ‘Ron’. Logical and this S1 copy with the DOB where someone in the S2 dataset that has a name that is not ‘Ron’ shares the same DOB. Cross product the two copies of the Students table. Project the name of S2 that meets these conditions.

3.3

- $\pi_{cName}((\pi_{cID,sID}(Enroll)/\pi_{sID}(Students)) \bowtie Courses)$

This results in an empty table.

extract the cID and sID from the Enroll table. Extract the sID from the students table. the ‘/’ operand subtracts the sID set from the set of sID in the enrolled table. This is joined with

Courses. The result of the equation will return a table of any course name that every student is enrolled in.

Part 4

$p(\text{ThreeKLevels},(\pi_{cID}(_{\sigma\ (cID > 3000)\wedge(cID < 4000)}(\text{Courses})))$

$p(\text{studentsEnrolled}, (\pi_{sID}(\text{Enrolled} / \text{ThreeKLevels})))$

$\pi_{Name}(\text{Students} \bowtie_{Students.sID=studentsEnrolled.sID}(\text{studentsEnrolled}))$