CSCC43 Tutorial #2

More Relational Algebra

Andrew Leung

Administration

A1 is out on Quercus

Due on MarkUs June 8

Group size is up to 2 people

Piazza is your friend!

Schema

Student (sID, surName, firstName, campus, email, cgpa)
Course (dept, cNum, name, breadth)
Offering (oID, dept, cNum, term, instructor)
Took (sID, oID, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

Note: “breadth” is a Boolean indicating whether or not a course satisfies the breadth requirement for degrees in the Faculty of Arts and Science.

Queries

Student (<u>sID</u>, surName, firstName, campus, email, cgpa)
Course (<u>dept</u>, <u>cNum</u>, name, breadth)
Offering (<u>oID</u>, dept, cNum, term, instructor)
Took (<u>sID</u>, <u>oID</u>, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

1. Student number of all students who have taken csc343.

$$\pi_{sID}\ \sigma_{dept='csc' \wedge cNum=343}\ (Took \bowtie Offering)$$

Queries

Student (<u>sID</u>, surName, firstName, campus, email, cgpa)
Course (<u>dept</u>, <u>cNum</u>, name, breadth)
Offering (<u>oID</u>, dept, cNum, term, instructor)
Took (<u>sID</u>, <u>oID</u>, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

2. Student number of all students who have taken csc343 and earned an A+ in it. (Assume grade A for >=90)

$$R1 := \pi_{sID}\ \sigma_{dept='CSC' \wedge cNum=343 \wedge grade>=90}\ (Took \bowtie Offering)$$

Queries

Student (<u>sID</u>, surName, firstName, campus, email, cgpa)
Course (<u>dept</u>, <u>cNum</u>, name, breadth)
Offering (<u>oID</u>, dept, cNum, term, instructor)
Took (<u>sID</u>, <u>oID</u>, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

3. The names of all such students.

$$\pi_{surName,\ firstName}(R1 \bowtie Student)$$

Queries

Student (<u>sID</u>, surName, firstName, campus, email, cgpa)
Course (<u>dept</u>, <u>cNum</u>, name, breadth)
Offering (<u>oID</u>, dept, cNum, term, instructor)
Took (<u>sID</u>, <u>oID</u>, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

4. The names of all students who have passed a breadth course with Professor Percy.

$$PercyBreadth := \pi_{oID}\ \sigma_{breadth=true \wedge instructor='Percy'}\ (Course \bowtie Offering)$$

$$Passers := \pi_{sID}\ \sigma_{grade >= 50}\ (PercyBreadth \bowtie Took)$$

$$Names := \pi_{surName, firstName}\ (Passers \bowtie Student)$$

Queries

Student (<u>sID</u>, surName, firstName, campus, email, cgpa)
Course (<u>dept</u>, <u>cNum</u>, name, breadth)
Offering (<u>oID</u>, dept, cNum, term, instructor)
Took (<u>sID</u>, <u>oID</u>, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

5. sID of all students who have earned some grade over 80 and some grade below 50.

$$(\pi_{sID}\ \sigma_{grade>80}\ Took) \cap (\pi_{sID}\ \sigma_{grade<50}\ Took)$$

Queries

Student (<u>sID</u>, surName, firstName, campus, email, cgpa)
Course (<u>dept</u>, <u>cNum</u>, name, breadth)
Offering (<u>oID</u>, dept, cNum, term, instructor)
Took (<u>sID</u>, <u>oID</u>, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

6. Terms when Horton and Heap were both teaching something.

$$(\pi_{term}\ \sigma_{instructor='Horton'}\ Offering) \cap (\pi_{term}\ \sigma_{instructor='Heap'}\ Offering)$$

Queries

Student (<u>sID</u>, surName, firstName, campus, email, cgpa)
Course (<u>dept</u>, <u>cNum</u>, name, breadth)
Offering (<u>oID</u>, dept, cNum, term, instructor)
Took (<u>sID</u>, <u>oID</u>, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

7. Terms when either of them was teaching CSC343.

$$\pi_{term}\ \sigma_{dept='CSC' \wedge cNum='343' \wedge (instructor='Horton' \vee instructor='Heap')}\ Offering$$

Queries

Student (<u>sID</u>, surName, firstName, campus, email, cgpa)
Course (<u>dept</u>, <u>cNum</u>, name, breadth)
Offering (<u>oID</u>, dept, cNum, term, instructor)
Took (<u>sID</u>, <u>oID</u>, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

8. sID of students who have earned a grade of 85 or more, or who have passed a course taught by Atwood (Assume passing at 50).

$HighGrade := \pi_{sID}\ \sigma_{grade >= 85}\ Took$

$PassedByAtwood := \pi_{sID}\ \sigma_{grade >= 50 \wedge instructor = 'Atwood'}\ (Took \bowtie Offering)$

$Students := HighGrade \cup PassedByAtwood$

Queries

Student (<u>sID</u>, surName, firstName, campus, email, cgpa)
Course (<u>dept</u>, <u>cNum</u>, name, breadth)
Offering (<u>oID</u>, dept, cNum, term, instructor)
Took (<u>sID</u>, <u>oID</u>, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

9. Terms when CSC369 was not offered.

$$\pi_{term}\ Offering - \left(\pi_{term}\ \sigma_{dept='CSC' \wedge cNum=369}\ Offering\right)$$

Queries

Student (<u>sID</u>, surName, firstName, campus, email, cgpa)
Course (<u>dept</u>, <u>cNum</u>, name, breadth)
Offering (<u>oID</u>, dept, cNum, term, instructor)
Took (<u>sID</u>, <u>oID</u>, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

10. Department and course number of courses that have never been offered.

$$\pi_{dept, cNum}\ Course - \pi_{dept, cNum}\ Offering$$

Queries

Student (<u>sID</u>, surName, firstName, campus, email, cgpa)
Course (<u>dept</u>, <u>cNum</u>, name, breadth)
Offering (<u>oID</u>, dept, cNum, term, instructor)
Took (<u>sID</u>, <u>oID</u>, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

11. SIDs and surnames of all pairs of students who have taken a course together.

$Pairs(sID1, sID2) := \pi_{T1.sID,\ T2.sID}\ \sigma_{T1.oID = T2.oID\ \wedge\ T1.sID < T2.sID}\ (\rho_{T1}\ Took \times \rho_{T2}\ Took)$

$OneName(sID1, sID2, name1) := \pi_{sID1, sID2, surName}\ \sigma_{sID1 = sID}\ Pairs \times Student$

$Answer(sID1, sID2, name1, name2) := \pi_{sID1, sID2, surName}\ \sigma_{sID2 = sID}\ (OneName \times Student)$

Queries

Student (<u>sID</u>, surName, firstName, campus, email, cgpa)
Course (<u>dept</u>, <u>cNum</u>, name, breadth)
Offering (<u>oID</u>, dept, cNum, term, instructor)
Took (<u>sID</u>, <u>oID</u>, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

12. sID of student(s) with the highest grade in CSC343, in term 20089.

$$Takers := \pi_{sID, oID, grade}\left(\left(\sigma_{\substack{dept='CSC' \wedge \\ cNum=343 \wedge \\ term=20089}} Offering\right) \bowtie Took\right)$$

$$NotTop := \pi_{\substack{T1.sID, \\ T1.oID \\ T1.grade}} \sigma_{T1.grade < T2.grade}\left(\rho_{T1}\, Takers \times \rho_{T2}\, Takers\right)$$

$$Answer := \pi_{sID}(Takers - NotTop)$$

Queries

Student (<u>sID</u>, surName, firstName, campus, email, cgpa)
Course (<u>dept</u>, <u>cNum</u>, name, breadth)
Offering (<u>oID</u>, dept, cNum, term, instructor)
Took (<u>sID</u>, <u>oID</u>, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

13. sID of students who have a grade of 100 at least twice.

$$\text{AtLeastTwice} := \Pi_{T1.sID}\ \sigma_{T1.grade = T2.grade = 100 \wedge T1.sID = T2.sID \wedge T1.oID < T2.oID}\ (\rho_{T1}\,\text{Took} \times \rho_{T2}\,\text{Took})$$

Queries

Student (<u>sID</u>, surName, firstName, campus, email, cgpa)
Course (<u>dept</u>, <u>cNum</u>, name, breadth)
Offering (<u>oID</u>, dept, cNum, term, instructor)
Took (<u>sID</u>, <u>oID</u>, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

14. sID of students who have a grade of 100 exactly twice.

AtLeastThrice := $\Pi_{T1.sID}$ $\sigma_{T1.oID<T2.oID<T3.oID \wedge T1.grade=T2.grade=T3.grade=100 \wedge T1.sID=T2.sID=T3.sID}$ (ρ_{T1} Took × ρ_{T2} Took × ρ_{T3} Took)

ExactlyTwice := AtLeastTwice − AtLeastThrice

Queries

Student (<u>sID</u>, surName, firstName, campus, email, cgpa)
Course (<u>dept</u>, <u>cNum</u>, name, breadth)
Offering (<u>oID</u>, dept, cNum, term, instructor)
Took (<u>sID</u>, <u>oID</u>, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

15. sID of students who have a grade of 100 at most twice.

π_{sID} Student − AtLeastThrice

Queries

Student (<u>sID</u>, surName, firstName, campus, email, cgpa)
Course (<u>dept</u>, <u>cNum</u>, name, breadth)
Offering (<u>oID</u>, dept, cNum, term, instructor)
Took (<u>sID</u>, <u>oID</u>, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

16. Department and cNum of all courses that have been taught in every term when CSC320 was taught.

320Terms := π_{term} $\sigma_{dept='CSC' \wedge cNum=320}$ Offering

CourseTerms := $\pi_{dept, cNum, term}$ Offering

Answer := CourseTerms $\div$ 320Terms

Exercise: Find equivalent query without using $\div$

Integrity Constraints

Student (sID, surName, firstName, campus, email, cgpa)
Course (dept, cNum, name, breadth)
Offering (oID, dept, cNum, term, instructor)
Took (sID, oID, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

Use the notation.

$<relational\ algebra\ expression> = \emptyset$

to write an integrity constraint for each of the following.

Integrity Constraints

Student (sID, surName, firstName, campus, email, cgpa)
Course (dept, cNum, name, breadth)
Offering (oID, dept, cNum, term, instructor)
Took (sID, oID, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

1. Courses at the 400-level cannot count for breadth.

$\sigma_{400 \leq cNum < 500 \wedge breadth=true}\ Course = \emptyset$

Integrity Constraints

Student (<u>sID</u>, surName, firstName, campus, email, cgpa)
Course (<u>dept</u>, <u>cNum</u>, name, breadth)
Offering (<u>oID</u>, dept, cNum, term, instructor)
Took (<u>sID</u>, <u>oID</u>, grade)

Offering [dept, cNum] ⊆ Course [dept, cNum]
Took [sID] ⊆ Student [sID]
Took [oID] ⊆ Offering [oID]

2. CSC490 can only be offered at the same time as CSC454.

$$490Terms := \pi_{term}\ \sigma_{dept='CSC' \wedge cNum=490}\ Offering$$

$$454Terms := \pi_{term}\ \sigma_{dept='CSC' \wedge cNum=454}\ Offering$$

$$490Terms - 454Terms = \emptyset$$