Midterm RA Notes

Table of Contents

"Every" or "All" Without Using Division

Algorithm:

1. Get or create all necessary tables needed. (R_1)
2. Use Cartesian Product or join to create a table that contains all possible permutations. (R_2)
3. Take R_2 and subtract it from a table that contains all current permutations. The result is a table that does not satisfy the "every" or "all" condition. (R_3)
4. Take all possible values and subtract it from R_3 to get the answer.

E.g. 1 Given the below schemas, find the students who have taken every CSC course.

Schemas:
Student (<u>sid</u>, name)
Course (<u>dept</u>, <u>cnum</u>)
Taken (<u>sid</u>, <u>dept</u>, <u>cnum</u>)

Solution:
1. Filter out CSC courses.
$R_1 = \sigma_{dept = 'csc'}(Course)$

2. Cross Join Student with R_1
$R_2 = \pi_{sid}(Student) \times R_1$

Now, we have a table of all student - $\overset{CSC}{\wedge}$ course permutations.

3. Subtract Taken from R_2.
$R_3 = R_2 - Taken$

Now, we have a table of the students and the CSC courses they didn't take.

4. Subtract R_3 from Student
$R_4 = \Pi_{sid}(Student) - \Pi_{sid}(R_3)$

Now, we have our answer.

Fig. 2 Given the below schemas, find the students who have taken every course in their program.

Schemas:
Student (sid, name, program)
Course (dept, cnum)
Taken (sid, dept, cnum)

Solution:
We can't use cartesian product here by itself as we would get student-course pairs that are invalid. For example, we would get a student in the math program paired with a French course.

Instead, we have to use a theta join to make sure that the student's program and course dept match up.

1. Do a theta join between Student and Course.

$R_1 = \Pi_{sid,\ dept,\ cnum}(Student \bowtie_{program = dept} Course)$

2. Do R_1 - Taken
$R_2 = R_1$ - Taken

R_1 is a table that contains all student — course permutations such that the student's program and course deptartment match up.

So, R_2 is a table of all student - course permutations such that the student didn't take the course.

3. Do Student - R_2
$R_3 = \Pi_{sid}(Student) - \Pi_{sid}(R_2)$
↑ Final answer

At Least N

Algorithm:
1. Find or create the necessary tables. (R_1)
2. Do N-1 self joins with R_1, and filter rows.
3. Get all necessary columns.

E.g. 1 Given the schema, find the SIDs of the students who have a grade of 100 at least twice.

Schema

Student (sid, name)

Course (oid, dept, cnum)

Took (sid, oid, grade)

Solution:

1. $R_1 = (\rho_{T_1}\ \text{Took}) \times (\rho_{T_2}\ \text{Took})$
2. $R_2 = \Pi_{T_1.sid}\left(\sigma_{T_1.sid = T_2.sid \wedge T_1.oid \neq T_2.oid \wedge T_1.grade = 100 \wedge T_2.grade = 100}(R_1)\right)$

Same Student

Diff Course

Both course marks are 100.

Note: This question is from week 3 tutorial, schema 2, Q13. I modified the schemas a bit.

At Most N

Algorithm:

1. Find "At least N+1" (R_1)
2. Total − R_1
3. Project necessary columns

E.g. 1 Given the schemas, find the sids of the students who have a grade of 100 at most twice.

Schema:

Student (<u>sid</u>, name)
Course (<u>oid</u>, dept, cnum)
Took (<u>sid</u>, <u>oid</u>, grade)

Solution:

1. $R_1 = (\rho_{T_1}\ \text{Took}) \times (\rho_{T_2}\ \text{Took}) \times (\rho_{T_3}\ \text{Took})$

2. $R_2 = \Pi_{T_1.sid}\left(\sigma_{T_1.sid = T_2.sid = T_3.sid\ (R_1)} \wedge T_1.oid \neq T_2.oid \wedge T_1.oid \neq T_3.oid \wedge T_2.oid \neq T_3.oid \wedge T_1.grade = T_2.grade = T_3.grade = 100\right)$

3. $R_3 = \Pi_{sid}(Student) - R_2$

In steps 1 and 2, I was finding "At Least 3" to get the students who got at least 3 100's.

In step 3, I subtracted the students we got from all of the students to get the students who got 100 at most twice.

Note: This question is from tutorial 3, schema 2, Q15 with the schemas modified a bit.

Exactly N

Algorithm

1. Find "At least N" (R_1)
2. Find "At least N+1" (R_2)
3. $R_1 - R_2$ and project necessary columns

E.g. 1 Given the schemas below, find the sids of the students who got a mark of 100 exactly twice.

Schemas:
Student (<u>sid</u>, name)
Course (<u>oid</u>, dept, cnum)
Took (<u>sid</u>, <u>oid</u>, grade)

Note: This question is from week 3 tutorial, schema 2, Q14 with the schemas modified a bit.

Solution:

At least 2

1. $R_1 = (\rho_{T_1}\ Took) \times (\rho_{T_2}\ Took)$
2. $R_2 = \Pi_{T_1.sid}\left(\sigma_{T_1.sid = T_2.sid \wedge T_1.oid \neq T_2.oid \wedge T_1.grade = T_2.grade = 100}\ (R_1)\right)$

At least 3

3. $R_3 = R_1 \times (\rho_{T_3}\ Took)$
4. $R_4 = \Pi_{T_1.sid}\left(\sigma_{T_1.sid = T_2.sid = T_3.sid \wedge T_1.oid \neq T_2.oid \wedge T_2.oid \neq T_3.oid \wedge T_1.oid \neq T_3.oid \wedge T_1.grade = T_2.grade = T_3.grade = 100}\ (R_3)\right)$
5. $R_2 - R_4$

Highest / Most

Algorithm:

1. Create or get the necessary table(s). (R_1)
2. Create a renamed copy of R_1. (R_2)
3. Do $R_1 \bowtie_{R1.value < R2.value} R_2$ (R_3)
4. Do $R_1.value - R_3.R_1\text{-}value$

E.g. 1 Given the schemas below, find the sids of the students with the highest mark in term 20089 for CSCC43.

Schemas:

Student (<u>sid</u>, name)
Course (<u>oid</u>, dept, cnum, term)
Took (<u>sid</u>, <u>oid</u>, grade)

Note: This question is from tutorial 3, schema 2, Q12, with the schemas modified a bit.

Solution:

1. $R_1 = \Pi_{oid}\left(\sigma_{dept = \text{'csc'} \wedge cnum = 343 \wedge term = 20089}(Course)\right)$ ← Filtering for the course and term.
2. $R_2 = R_1 \bowtie Took$
3. $R_3 = (\rho_{new\text{-}r2}{}^{(R2)}) \bowtie_{new\text{-}r2.grade < R2.grade} (R2)$
4. $R_4 = \Pi_{sid}\, Student - \Pi_{new\text{-}r2.sid}\, R_3$

In step 3, because we did new-r2.grade < R2.grade, new-r2.grade can't contain the highest mark now. Hence, when we did Π_{sid} Student − $\Pi_{new\text{-}r2.sid}$ R_3, we get the sid of the student who got the highest mark.

Smallest / Lowest

Algorithm:

1. Create or get the necessary tables. (R1)
2. Create a renamed copy of R1. (R2)
3. Do $R1 \bowtie_{R1.value < R2.value} R2$ (R3)
4. Do R1.value - R3.R2-value

E.g. 1 Given the schemas below, find the sids of the students with the lowest mark for CSCC43 in term 20089.

Schemas:

Student (<u>sid</u>, name)
Course (<u>oid</u>, dept, cnum, term)
Took (<u>sid</u>, <u>oid</u>, grade)

Solution:

1. $R_1 = \Pi_{oid}\left(\sigma_{\substack{dept = \text{'csc'} \wedge \\ cnum = 343 \wedge \\ term = 20089}} Course\right)$
2. $R_2 = R_1 \bowtie Took$
3. $R_3 = (\rho_{new\text{-}r2}(R2)) \bowtie_{new\text{-}r2.grade < r2.grade} (R2)$
4. $R_4 = \Pi_{sid}\, Student - \Pi_{r2.sid}(R3)$