Exam 2022 - question 1

samedi 13 janvier 2024 15:19

Question 1. (25 pts)

Consider the following relations for a database that keeps track of student enrollement in courses. Also consider that the relations are bags that allow duplicates:

- student<id, name, age, deptid> storing 100000 tuples
- department<id, name> storing 600 tuples
- course<id, name, semester> strong 3000 tuples
- enrollment<student_id, course_id, year, grade> storing 4000000 tuples

(a) For the following query:

List id and name of all students who didn't take courses in the first semester of 2021

give two RA expressions for this query (for example, by applying equivalence rules), one of which is optimized in terms of performance, and briefly discuss the estimated performance of the two expressions

```
Select s.id, s.name
From students, enrollment e
Where s.id = e.enrolmentid AND NOT EXISTS
        (SELECT *
         FROM course c
         where  c.course_ID = e.courseID
           AND  c.semester = 1  AND e.year = 2021)
```

⇒ Not optimised as there is a big subquery

(A) $\Pi_{s.id,\ s.name}\left(\sigma_{c.semester=1 \wedge e.year=2021}\left(\rho_c(course) \bowtie \rho_e(enrollment) \bowtie \rho_s(student)\right)\right)$

↳ Not optimised as join THEN filtering, it is better to do the inverse for memory

(B) $\Pi_{s.id,\ s.name}\left(\rho_s(student) \bowtie \left(\sigma_{e.year=2021}(\rho_e(enrollment)) \bowtie \sigma_{c.semester=1}(\rho_c(course))\right)\right)$

↳ optimised as the filtering is made BEFORE the joins

(A) Filter 100.000 x 3000 x 4000.000 tuples

(B) Filter on 100.000 + 3000 + 4000.000 tuples ⇒ this is non negligible!!

(b) Write an RA expression to express this query:

List all department id and name of the students who attend the Database course ⇒ set-based query

Make sure not to show duplicate results

```
SELECT s.name, s.deptid DISTINCT  FROM student s, course c, enrollment e
WHERE c.name='Database' AND c.id = e.course_id AND e.studentid = s.id
```

$\Pi_{s.name,\ s.dept\_id}\left(\rho_c\left(\sigma_{name='Database'}(course)\right) \bowtie_{c.id=e.course\_id} \rho_e(enrollment) \bowtie_{e.student\_id=s.id} \rho_s(student)\right)$

$\pi_{s.name, s.dept\_id}(\rho_c(\sigma_{name='Database'}(course)) \bowtie_{c.id=e.course\_id} \rho_e(enrollment) \bowtie_{e.student\_id=s.id} \rho_s(student))$

↳ There will be no duplicates as π is a projection and eradicates them :)

(c) Normalize/flatten the following query and give its equivalent RA expression

```
SELECT course.name
FROM course
WHERE course.id in (
      SELECT enrollment.course_id
      FROM student, enrollment
      WHERE student.id = enrollment.student_id AND
      student.name LIKE '%son');
```

Normalizing → transforming to a EXIST subquery

```
SELECT course.name
FROM course
WHERE EXISTS ( SELECT 1 FROM enrollment, student
               WHERE enrollment.student_id = student.id AND student.name like '%son')
```

Flattening is done in 6 steps

1. Translate the subquery

$$\rho_s(\sigma_{name\ like\ '\%son'}(student) \bowtie_{e.student\_id=s.id} \rho_e(enrollment)$$

2. Add context relation and parameters

$$\pi_{e.course\_id}(\rho_s(\sigma_{name\ like\ '\%son'}(student) \bowtie_{e.student\_id=s.id} \rho_e(enrollment)))$$

3. Translate the FROM of the outer query

$$\rho_c(course)$$

4. Synthesize both expression

$$\rho_c(course) \bowtie \pi_{e.course\_id}(\rho_s(\sigma_{name\ like\ '\%son'}(student) \bowtie_{e.student\_id=s.id} \rho_e(enrollment)))$$

5. simplify

6. Complete the expression

$$\pi_{c.name}(\rho_c(course) \bowtie \pi_{e.course\_id}(\rho_s(\sigma_{name\ like\ '\%son'}(student) \bowtie_{e.student\_id=s.id} \rho_e(enrollment))))$$

(d) Suppose relation R<A,B,C, D> has the tuples

(1,2,3,4)
(1,2,3,5)
(3,2,1,0)

(A) Using **bag** projection and theta-join, how many tuples appear in the result of:

$$\pi_{A,B}(R) \bowtie_{R.B<S.B} \rho_S(\pi_{B,C}(R))$$

(B) Using **set** projection and theta-join, how many tuples appear in the result?

A	B	C	D
1	2	3	4

(A) $\pi_{A,B}(R)$

R

A	B
1	2

$\rho_S(\pi_{B,C}(R))$

S

B	C
2	3

A	B	C	D
1	2	3	4
1	2	3	5
3	2	1	0

(A) $\Pi_{A,B}(R)$

A	B
1	2
1	2
3	2

$\rho_S(\Pi_{B,C}(R))$

B	C
2	3
2	3
2	1

$\Rightarrow \Pi_{A,B}(R) \bowtie_{R.B < S.B} \rho_S(\Pi_{B,C}(R))$ outputs 0 results as . 2 is < 2

(B) If it is set-based $\Pi_{A,B}(R)$

R

A	B
1	2
3	2

$\rho_S(\Pi_{B,C}(R))$

S

B	C
2	3
2	1

$\Rightarrow$ 0 output