# Assignment 6
# Computer Science
# Fall 2017
# B461

Steven Myers

November 2, 2017

## Answers

1. (a) i. Original Query:

```
SELECT s.sid, b1.bookno
FROM   student s, buys b1, buys b2
WHERE  s.sid = b1.sid AND
s.sid= b2.sid AND
b1.bookno <> b2.bookno AND
s.sname = 'Eric' and b1.bookno <> '2010';
```

ii. Translated into RA:

$$\pi_{sid,bookno}(\sigma_{B.bookno \neq B2.bookno \wedge S.name='Eric' \wedge B.bookno \neq '2010' \wedge s.sid=b2.sid}(S \times B \times B2))$$

(b) i. Original Query

```
SELECT DISTINCT b.bookno, b.title
FROM book b, student s
WHERE b.price = SOME(select b1.price
                     from buys t, book b1
                     where b1.price > 50 and
                     s.sid = t.sid and
                     t.bookno = b1.bookno);
```

ii. Convert from using SOME to an exists statement

```
SELECT DISTINCT b.bookno, b.title
FROM book b, student s
WHERE EXISTS(select 1
             from buys t, book b1
             where b1.price > 50 and
             s.sid = t.sid and
             t.bookno = b1.bookno);
```

iii. Push down student $s$ relation into subquery as $s1$.

```
SELECT DISTINCT b.bookno, b.title
FROM book b, student s
WHERE EXISTS(select 1
             from buys t, book b1, student s1
             where b1.price > 50 and
```

```
s1.sid = t.sid and
t.bookno = b1.bookno);
```

iv. The inner query can be translated as:

$$\pi_{b1.price}(\sigma_{b1.price>50 \wedge s1.sid=t.sid \wedge t.bookno=b1.bookno}(T \times B1 \times S1))$$

v. Finally, we perform the semi-join with the inner query and the student and book relation:

$$\pi_{b.bookno,b.title}(S \times B \ltimes \pi_{b1.price}(\sigma_{b1.price>50 \wedge s1.sid=t.sid \wedge t.bookno=b1.bookno}(T \times B1 \times S1)))$$

(c) i. Original Query

```
SELECT b.bookno
FROM book b
WHERE b.bookno IN
(SELECT b1.bookno FROM book b1 WHERE b1.price > 50)
UNION
(SELECT c.bookno FROM cites c);
```

ii. RA representation of inner query (simple translation)

$$\pi_{bookno}(\sigma_{price>50}(book)) \cup \pi_{bookno}(cites)$$

iii. Since the IN predicate is equivalent to saying that there exists one bookno for which a bookno in the inner query matches, we can use the semi join

$$\pi_{bookno}(book) \ltimes \pi_{bookno}(\sigma_{price>50}(book)) \cup \pi_{bookno}(cites)$$

(d) i. Original Query

```
SELECT b.bookno FROM book b
WHERE b.price >= 80 and
NOT EXISTS(SELECT b1.bookno
           FROM book b1
           WHERE b1.Price > b.Price);
```

ii. First, push down book b relation into subquery as book b2.

```
SELECT b.bookno FROM book b
WHERE b.price >= 80 and
NOT EXISTS(SELECT b1.bookno
           FROM book b1, book b2
           WHERE b1.Price > b2.Price);
```

iii. Now, we can properly translate the inner query as:

$$\pi_{b1.bookno}(\sigma_{b1.price>b2.price}(B1 \times B2))$$

iv. To preserve the semantics of the original outer query, we need to perform an anti-semi join:

$$\pi_{b.bookno}(\sigma_{b.price>=80}(B \ \overline{\ltimes}\ \pi_{b1.bookno}(\sigma_{b1.price>b2.price}(B1 \times B2))))$$

(e) i. Original query:

```
SELECT s.sid
FROM Student s
WHERE EXISTS(SELECT 1
             FROM Book b
             WHERE b.price > 50 AND
             b.bookno IN (SELECT t.bookno
                          FROM Buys t
                          WHERE s.sid = t.sid AND
                          s.sname = 'Eric'))
```

ii. First, push down parameterized values in the inner-most query. We will push down the student $s$ relation as $s1$ and $s2$.

```
SELECT s.sid
FROM Student s
WHERE EXISTS(SELECT 1
             FROM Book b, Student s2
             WHERE b.price > 50 AND
             b.bookno IN (SELECT t.bookno
                          FROM Buys t, Student s1
                          WHERE s1.sid = t.sid AND
                          s1.sname = 'Eric'))
```

iii. Now, we push down the book b relation into the deepest level query :

```
SELECT s.sid
FROM Student s
WHERE EXISTS(SELECT 1
             FROM Book b, Student s2
             WHERE b.price > 50 AND
             b.bookno IN (SELECT t.bookno
                          FROM Buys t, Student s1, Book b1
                          WHERE s1.sid = t.sid AND
                          s1.sname = 'Eric'))
```

iv. Next, convert the first inner subquery IN expression into an EXISTS statement.

```
SELECT s.sid
FROM Student s
WHERE EXISTS(SELECT 1
             FROM Book b, Student s2
             WHERE b.price > 50 AND
             EXISTS (SELECT 1
                          FROM Buys t, Student s1
                          WHERE s1.sid = t.sid AND
                          b.bookno = t.bookno AND
                          s1.sname = 'Eric'))
```

v. Now we can properly translate the SQL to RA:

$$\pi_{sid}(S) \ltimes \pi_{bookno}(\sigma_{price>50}(B)) \ltimes \pi_{t.bookno}(\sigma_{s1.sid=t.sid \wedge b1.book=t.bookno \wedge s1.sname='Eric'}(T \times S1 \times B1))$$

(f) i. Original query:

```
SELECT s1.sid, s2.sid
FROM student s1, student s2
WHERE s1.sid <> s2.sid AND
NOT EXISTS(SELECT 1
           FROM Buys t1
           WHERE t1.sid = s1.sid AND
           t1.bookno NOT IN (SELECT t2.bookno
                             FROM Buys t2
                             WHERE t2.sid = s2.sid));
```

ii. For the deepest query, we need to translate the NOT IN statement to NOT EXISTS.

```
SELECT s1.sid, s2.sid
FROM student s1, student s2
WHERE s1.sid <> s2.sid AND
NOT EXISTS(SELECT 1
           FROM Buys t1
           WHERE t1.sid = s1.sid AND
           NOT EXISTS (SELECT 1
                       FROM Buys t2
                       WHERE t2.sid = s2.sid AND t1.bookno <> t2.bookno));
```

iii. We now need to push down our relations recursively from the top and middle queries to the deepest query.

```
SELECT s1.sid, s2.sid
FROM student s1, student s2
WHERE s1.sid <> s2.sid AND
NOT EXISTS(SELECT 1
           FROM Buys t, Student s1, Student s2
           WHERE t.sid = s1.sid AND
           NOT EXISTS (SELECT t2.sid, t1.bookno, t2.bookno, s1.sid
                       FROM Buys t2, Buys t1, Student S1, Student S2
                       WHERE t2.sid = s2.sid AND t1.bookno <> t2.bookno));
```

iv. Now that we've pushed down the upper query's relations into deeper queries, we can translate the deepest query as:

$$\varepsilon = \pi_{t2.sid,t1.bookno,t2.bookno,s1.sid}(\sigma_{t2.sid=s2.sid \wedge t1.bookno \neq t2.bookno}(T1 \times T2 \times S1 \times S2))$$

v. Our middle query then can be translated as:

$$\tau = \pi_{s1.sid,s2.sid,t.bookno}(\sigma_{t.sid=s1.sid}(T \times S1 \times S2) \overline{\ltimes} \varepsilon)$$

vi. Finally, we can semijoin the top level query in an expression like so:

$$\pi_{s1.sid,s2.sid}(\sigma_{s1.sid \neq s2.sid}(S1 \times S2) \overline{\ltimes} \tau)$$

2. (a) i. Original RA Expression:

$$\pi_{sid,bookno}(\sigma_{T.bookno \neq T2.bookno \wedge S.name='Eric' \wedge T.bookno \neq '2010' \wedge S.Sid=T.sid}(S \times T \times T2))$$

ii. Rule of Cascading Selections and pushing selections over cartesian products. Forming natural join between Student and Buys:

$$\pi_{sid,bookno}(\sigma_{T.bookno \neq T2.bookno}(\sigma_{sname='Eric'}(S) \bowtie \sigma_{bookno \neq '2010'}(T) \times T2))$$

iii. Now we can only project the required attributes from each relation in the cartesian product:

$$\pi_{sid,bookno}( \sigma_{T.bookno \neq T2.bookno}( \pi_{sid}((\sigma_{sname='Eric'}(S)) \bowtie \pi_{bookno,sid}(\sigma_{bookno \neq '2010'}(T)) \times \pi_{bookno}(T2)))$$

(b) i. Original RA Expression:

$$\pi_{b.bookno,b.title}(S \times B \ltimes \pi_{b1.price}(\sigma_{b1.price>50 \wedge s1.sid=t.sid \wedge t.bookno=b1.bookno}(T \times B1 \times S1))$$

ii. Cascade selections and pushing over cartesian products:

$$\pi_{b.bookno,b.title}(S \times B \ltimes \pi_{b1.price}(S1 \bowtie (T \bowtie B1) \times \sigma_{price>50}(B1)))$$

iii. Now we only project the pieces we need. We can just grab the sid from the Student relation, since no pieces of the Student relation is used.

$$\pi_{b.bookno,b.title}(\pi_{sid}(S) \times B \ltimes \pi_{b1.price}(\pi_{sid}(S1) \bowtie (T \bowtie B1) \times \sigma_{price>50}(\pi_{price}(B1))))$$

(c) i. Original RA Expression. Already optimized!

$$\pi_{bookno}(book) \ltimes \pi_{bookno}(\sigma_{price>50}(book)) \cup \pi_{bookno}(cites)$$

(d) i. Original RA Expression. Already optimized!

$$\pi_{b.bookno}(\sigma_{b.price>=80}(B \ \overline{\ltimes}\ \pi_{b1.bookno}(\sigma_{b1.price>b2.price}(B1 \times B2))))$$

(e) i. Original RA Expression:

$$\pi_{sid}(S) \ltimes \pi_{bookno}(\sigma_{price>50}(B)) \ltimes \pi_{t.bookno}(\sigma_{s1.sid=t.sid \wedge b1.book=t.bookno \wedge s1.sname='Eric'}(T \times S1 \times B1))$$

ii. Move selections inwards and form natural joins:

$$\pi_{sid}(S) \ltimes \pi_{bookno}(\sigma_{price>50}(B)) \ltimes \pi_{t.bookno}(\pi_{bookno}(\pi_{sname='Eric'}(S1) \bowtie (T1 \bowtie \pi_{bookno}(B1))))$$

(f) i. Original RA Expression:

$$\varepsilon = \pi_{t2.sid,t1.bookno,t2.bookno,s1.sid}(\sigma_{t2.sid=s2.sid \wedge t1.bookno \neq t2.bookno}(T1 \times T2 \times S1 \times S2))$$

$$\tau = \pi_{s1.sid,s2.sid,t.bookno}(\sigma_{t.sid=s1.sid}(T \times S1 \times S2) \overline{\ltimes} \varepsilon)$$

$$\pi_{s1.sid,s2.sid}(\sigma_{s1.sid \neq s2.sid}(S1 \times S2) \overline{\ltimes} \tau)$$