CSE 414 Worksheet: RA and Subqueries

Relational algebra operators:
- Union \( \bigcup \)
- Difference \( - \)
- Select \( \sigma \)
- Project \( \pi \)
- Join \( \Join \)
- Sort \( \tau \)
- Rename \( \rho \)
- Duplicate elimination \( \delta \)
- Grouping and aggregation \( \gamma \)

**CREATE TABLE Person (**
- `pid` INT PRIMARY KEY, -- person ID
- `name` VARCHAR(100)); -- person name

**CREATE TABLE Email (**
- `eid` INT PRIMARY KEY, -- email ID
- `pidFrom` INT REFERENCES Person, -- email sender
- `length` INT); -- email char length

**CREATE TABLE EmailTo (**
- `eid` INT REFERENCES Email, -- email ID
- `pidTo` INT REFERENCES Person, -- email recipient
- PRIMARY KEY (eid, pidTo));

**A warmup:** Find the number of emails that each person has sent.

**SELECT** `P.name`, COUNT(*)
**FROM** Person `P`, Email `E`
**WHERE** `P.pid = E.pidFrom`
**GROUP BY** `P.pid`, `P.name`;

Draw the RA tree for the query.

**Three-way join:** List the pid of people who write emails to themselves only shorter than 1000 characters and the number of emails they have sent to themselves.

**SELECT** `E1.pidFrom`, COUNT(*)
**FROM** Email `E1`, EmailTo `T`, Email `E2`
**WHERE** `E1.eid = T.eid` AND
- `T.pidTo = E2.pidFrom`
**GROUP BY** `E1.pidFrom`
**HAVING** MAX(`E2.length`) < 1000;
Draw the RA tree for the query.

A subquery problem: Find all emails where all of the recipients are named Alice.

```sql
SELECT E1.eid
FROM Email E1
WHERE NOT EXISTS (SELECT *
                    FROM EmailTo E2, Person P
                    WHERE E1.eid = E2.eid AND
                          E2.pidTo = P.pid AND
                          P.name != 'Alice');
```

The above has a correlated subquery. Write a de-correlated version of the query.

```sql
SELECT E1.eid
FROM Email E1
WHERE E1.eid NOT IN (SELECT E2.eid
                      FROM EmailTo E2, Person P
                      WHERE E2.pidTo = P.pid AND
                            P.name != 'Alice');
```

Draw the RA tree for the uncorrelated version of the query.