

## COMP9120 Relational Database Systems

**Tutorial Week 4: SQL and Relational Algebra****Exercise 1. SQL**

Consider the following query:

```
SELECT S.Name
FROM Student S, Transcript T
WHERE S.studId = T.studId
AND T.uosCode IN ('INFO2005', 'INFO2120')
```

a) What does this query mean (express the meaning in one short English sentence)?

The meaning of the query is: list the names of all students who took units INFO2005 or INFO2120.

b) Write an equivalent SQL query without using the IN operator and the set construct.

Without using IN operator or a set construct:

```
SELECT S.name
FROM Student S, Transcript T
WHERE S.studId = T.studId
AND ( T.uosCode = 'INFO2005' OR T.uosCode = 'INFO2120' )
```

There are alternatives, e.g., by using an explicit JOIN of the two relations.

c) Write the query in relational algebra.

$$\pi_{Name}(\sigma_{uosCode='INFO2005' \vee uosCode='INFO2120'}(Student \bowtie Transcript))$$
**Exercise 2. Reading Relational Algebra**

Consider the following schema:

```
Book (isbn, title, publisher, publicationYear)
Author (aname, birthdate)
Publisher (pname, address)
Wrote (isbn, aname) // which author wrote which book
```

What is the English explanation of the following Relational Algebra expressions?

a)  $\pi_{title, publicationYear}(Book)$

List the title and publication years of all books in the database.

b)  $\pi_{pname}(\sigma_{address='New York'}(Publisher))$

List the names of all publishers in New York.

c)  $\pi_{aname}(\sigma_{title='A First Course in Database Systems'}(Book \bowtie Wrote))$  *Natural join*

List the author(s) of the book 'A First Course in Database Systems'.

d)  $\pi_{address}(\sigma_{title='Databases' \vee title='Data Management'}(Publisher \bowtie_{pname=publisher} Book))$  *Theta join*

Show the address of the publisher(s) of books titled 'Databases' or 'Data Management'.

Why do the two previous queries ((c) and (d)) have to formulate their joins differently?

Because 'Book' and 'Wrote' have one common attribute ('isbn'), we can use the natural join. In contrast, 'Book' and 'Publisher' do not have an attribute in common – the foreign key in Book to Publisher is called 'publisher', not 'pname' – hence we need to explicitly formulate an equi-join condition.

### Exercise 3. Writing Relational Algebra

For the same schema as above, use relational algebra to express the following queries:

a) Find all book titles published by Acme Publishers

$$\pi_{title}(\sigma_{publisher='Acme'}(Book))$$

b) Find all authors (just by name) of the book with ISBN 0444455551

$$\pi_{aname}(\sigma_{isbn=0444455551}(Wrote))$$

c) Find all authors (name) who published at least one book with Acme Publishers

$$\pi_{aname}(\sigma_{publisher='Acme'}(Book \bowtie Wrote))$$

d) Find all authors (name) who never published a book with Acme Publishers.

$$\pi_{aname}(Author) - \pi_{aname}(\sigma_{publisher='Acme'}(Book \bowtie Wrote))$$

*Most efficient answer*