Homework 5
Rob Schwartz

Exercise 10.7.3

a) Write the natural joins that produce squares satisfying each of the three conditions above. You can use four different attributes W, X, Y, and Z, and assume that there are four copies of relation E with different schemas, so the joins can each be expressed as natural joins.

i) *c* is also lower than *b* and *d*

E(W, X) ⋪⋫ E(Y, X) ⋪⋫ E(Y, Z) ⋪⋫ E(W, Z)

```
SELECT e1.A, e1.B, e3.A, e3.B
FROM E e1, E e2, E e3, E e4
WHERE e1.A = e4.A AND e1.B = e2.B AND e2.A = e3.A AND e3.B = e4.B
```

ii) *c* is between *b* and *d*

E(W, X) ⋪⋫ E(X, Y) ⋪⋫ E(Y, Z) ⋪⋫ E(W, Z)

```
SELECT e1.A, e1.B, e3.A, e3.B
FROM E e1, E e2, E e3, E e4
WHERE e1.A = e4.A AND e1.B = e2.A AND e2.B = e3.A AND e3.B = e4.B
```

iii) *c* is higher than both *b* and *d*

E(W, X) ⋪⋫ E(X, Y) ⋪⋫ E(Z, Y) ⋪⋫ E(W, Z)

```
SELECT e1.A, e1.B, e3.A, e3.B
FROM E e1, E e2, E e3, E e4
WHERE e1.A = e4.A AND e1.B = e2.A AND e2.B = e3.B AND e3.A = e4.B
```

b) For which of these joins do we need a selection to assure that opposite corners are really different nodes?

We need a selection only for the joins that have opposite-corner nodes that are in the same comparison "bucket" (so to speak) - so opposite-corner nodes either both above, both below, or both between the other two nodes. In this case, those joins are *i* and *iii*, as *a* and *c* are both lower than *b* and *d* in *i*, and *b* and *d* are both between *a* and *c* in *iii*.

(d) Unlike the case of triangles, it is not guaranteed that each square is produced only once, although we can be sure that each square is produced by only one of the three joins. For example, a square in which the two nodes at opposite corners are each lower numerically than each of the other two nodes will only be produced by the join (i). For each of the three joins, how many times does it produce any square that it produces at all?

Because there are 4 nodes, there are 8 possible ways in which those nodes can be arranged where *a* is smaller than both *b* and *d*. They are as follows. The order listed below is in W-X-Y-Z order. The Roman numeral following each possible order represents which of the three joins could create that square.

1. 1-2-3-4 (ii)
2. 1-2-4-3 (iii)
3. 1-3-2-4 (i)
4. 1-3-4-2 (iii)
5. 1-4-2-3 (i)
6. 1-4-3-2 (ii)
7. 2-3-1-4 (i)
8. 2-4-1-3 (i)

The join *i* produces 4 squares, while the other joins produce 2 squares each. If we were to extrapolate and allow for the earlier condition (*a* being smaller than both *b* and *d*) to be ignored, each join would produce two squares given each starting node. So we can multiply out by four, and each join would, if *a* did not have to be smaller than both *b* and *d*, produce 8 squares.